Universidade Federal de Juiz de Fora
Instituto de Ciências Exatas
Bacharelado em Ciência da Computação

Interoperabilidade Omnichain em Blockchain: uma análise do protocolo LayerZero

Maria Cecília Romão Santos

JUIZ DE FORA
JANEIRO, 2025

Interoperabilidade Omnichain em Blockchain: uma análise do protocolo LayerZero

Maria Cecília Romão Santos

Universidade Federal de Juiz de Fora

Instituto de Ciências Exatas

Departamento de Ciência da Computação

Bacharelado em Ciência da Computação

Orientador: Alex Borges Vieira

JUIZ DE FORA

JANEIRO, 2025

Interoperabilidade Omnichain em Blockchain: uma análise do protocolo LayerZero

Maria Cecília Romão Santos

MONOGRAFIA SUBMETIDA AO CORPO DOCENTE DO INTITUTO DE CIÊNCIAS EXATAS DA UNIVERSIDADE FEDERAL DE JUIZ DE FORA, COMO PARTE INTEGRANTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE BACHAREL EM CIÊNCIA DA COMPUTAÇÃO.

Aprovada por:

Alex Borges Vieira
D.Sc. em Ciências da Computação - UFMG

Ronan Dutra Mendonça
D.Sc. em Ciência da Computação - UFV

Luciana Conceição Dias Campos
D.Sc. em Engenharia Elétrica - PUCRio

JUIZ DE FORA
20 DE JANEIRO, 2025

À minha família, pelo apoio e sustento.

Resumo

A interoperabilidade entre *blockchains* tornou-se um desafio relevante diante da fragmentação do ecossistema distribuído e da crescente demanda por aplicações descentralizadas que operem de forma integrada entre diferentes redes. A ausência de mecanismos amplamente padronizados e seguros para comunicação *cross-chain* levanta questões técnicas relacionadas a modelos de confiança, segurança e desenho de protocolos. Este trabalho tem como objetivo analisar a interoperabilidade *omnichain* sob uma perspectiva conceitual e arquitetural, com foco no protocolo *LayerZero*. A metodologia adotada combina uma revisão bibliográfica da literatura recente sobre interoperabilidade *blockchain* com uma análise crítica das principais abordagens existentes, incluindo *bridges* tradicionais, protocolos *multi-chain* e soluções *omnichain*. Como complemento, realiza-se uma experimentação prática por meio da implementação do padrão *Omnichain Fungible Token* (*OFT*), permitindo observar o funcionamento do protocolo *LayerZero* em um cenário aplicado. Como resultado, o trabalho organiza e discute pressupostos, vantagens e limitações do modelo *omnichain* proposto pelo *LayerZero*, destacando *trade-offs* em termos de segurança, dependências externas e impactos no desenvolvimento de aplicações *cross-chain*. A análise contribui para uma compreensão mais precisa das implicações técnicas da interoperabilidade *blockchain*, oferecendo subsídios tanto para pesquisadores quanto para desenvolvedores interessados na adoção de soluções mais seguras e previsíveis.

Palavras-chave: Interoperabilidade *Blockchain*; *Omnichain*; *Cross-chain*; *LayerZero*; Sistemas Distribuídos.

Abstract

Blockchain interoperability has become a significant challenge in light of the fragmentation of the distributed ecosystem and the growing demand for decentralized applications that operate in an integrated manner across different networks. The lack of widely standardized and secure mechanisms for cross-chain communication raises technical issues related to trust models, security, and protocol design. This work aims to analyze omnichain interoperability from a conceptual and architectural perspective, focusing on the LayerZero protocol. The adopted methodology combines a literature review of recent literature on blockchain interoperability with a critical analysis of the main existing approaches, including traditional bridges, multi-chain protocols, and omnichain solutions. Additionally, a practical experiment is conducted through the implementation of the Omnichain Fungible Token (OFT) standard, allowing the observation of LayerZero protocol in an applied scenario. As a result, this paper organizes and discusses the main assumptions, advantages, and limitations of the omnichain model proposed by LayerZero, highlighting its trade-offs in terms of security, external dependencies, and impacts on cross-chain application development. The analysis contributes to a more accurate understanding of the technical implications of blockchain interoperability, providing insights for both researchers and developers interested in adopting more secure and predictable cross-chain solutions.

Keywords: Blockchain Interoperability; Omnichain; Cross-chain; LayerZero; Distributed Systems.

Agradecimentos

Primeiramente a Deus, pois sem Ele nada seria possível na minha vida.

À minha mãe e à minha avó (in memoriam), meu maior alicerce. Por apostarem todas as fichas em mim, pelas orações que sempre me guardaram e pela fé inesgotável que sempre depositaram em mim. Nada disso teria sido possível sem o amor, o apoio e a força que recebi de vocês.

Estendo esse agradecimento a todas as mulheres que vieram antes de mim e que, cada uma à sua maneira, abriram caminhos, ocuparam espaços e me inspiraram pela coragem, pela resistência e pela forma singular com que escreveram suas próprias histórias.

Aos meus padrinhos (in memoriam), cuja presença, apoio e generosidade marcaram de forma definitiva a minha trajetória. Em momentos difíceis, foram fonte de amparo, cuidado e incentivo. Levo comigo os valores que me transmitiram e a confiança em mim depositada, que se refletem em cada conquista alcançada.

Ao Lucas, meu companheiro de jornada, pela paciência nos momentos mais atarefados, pelo apoio constante e por acreditar em mim mesmo quando eu mesma duvido. Sua presença foi essencial para que eu seguisse em frente.

Ao professor Alex Borges, meu orientador, pelo incentivo desde as primeiras ideias deste trabalho e pela orientação atenta, sem os quais este trabalho não seria concluído

Aos meus amigos da Cheesecake Labs, pelo ambiente de trabalho inspirador, pelas trocas constantes de conhecimento e pela oportunidade de atuar em grandes projetos na área de blockchain. O apoio, a parceria e a compreensão tornaram a dupla jornada de trabalho e estudo mais leve e enriquecedora, contribuindo de forma significativa para o meu crescimento profissional.

Aos professores do Departamento de Ciência da Computação pelos seus ensinamentos e aos funcionários do curso, que durante esses anos, contribuíram de algum modo para o nosso enriquecimento pessoal e profissional.

"A mente que se abre a uma nova ideia jamais retorna ao seu tamanho original".

Albert Einstein

Conteúdo

Lista de Figuras

Lista de Tabelas

Lista de Abreviações e Glossário

Blockchain	Livro-razão distribuído baseado em consenso descentralizado
Bridge	Mecanismo de transferência de ativos ou mensagens entre *blockchains*
Cross-chain	Comunicação ou interação entre diferentes *blockchains*
DVN	*Decentralized Verification Networks*
DeFi	Finanças Descentralizadas
EVM	*Ethereum Virtual Machine*
IBC	*Inter-Blockchain Communication Protocol*
MEV	*Maximal* (ou *Miner*) *Extractable Value*
Multi-chain	Arquitetura com múltiplas *blockchains* operando em paralelo
NFT	*Non-Fungible Token*
OFT	*Omnichain Fungible Token*
Omnichain	Abordagem que abstrai a comunicação entre múltiplas *blockchains*
UA	*User Application* (aplicação usuária no *LayerZero*)
UX	*User Experience* (Experiência do usuário)
Oracle	Componente que fornece dados externos ao blockchain
Relayer	Componente responsável por encaminhar mensagens entre redes

1 Introdução

Nos últimos anos, a tecnologia *blockchain* consolidou-se como uma infraestrutura relevante para além do uso em criptomoedas, passando a sustentar aplicações em áreas como finanças descentralizadas (DeFi), identidade digital, governança e tokenização de ativos (DENG et al., 2025; LAWRENCE, 2025b). Esse avanço, entretanto, ocorreu de forma descentralizada e heterogênea, resultando no surgimento de múltiplas redes *blockchain*, cada uma com objetivos específicos, modelos de consenso distintos (como *Proof of Work* e *Byzantine Fault Tolerance*), linguagens próprias e ecossistemas independentes (YEDDOU; BELOUCHRANI; MOKRANE, 2024; VONEITZEN et al., 2024). Como consequência, o ambiente atual caracteriza-se por um cenário *multi-chain* fragmentado, no qual redes operam de maneira isolada, formando "silos de dados e valor"que dificultam a circulação de ativos, dados e aplicações entre diferentes domínios (BELCHIOR et al., 2021; LAWRENCE, 2025a; ZHAO et al., 2023).

Esse cenário de fragmentação pode ser visualizado na Figura 1.1, que ilustra diferentes redes blockchain operando de forma independente, cada uma com seus próprios nós, protocolos e ecossistemas. A ausência de mecanismos nativos de interoperabilidade entre essas redes evidencia a formação de silos, nos quais ativos e informações permanecem restritos ao domínio de uma única blockchain, reforçando os desafios para integração e comunicação entre sistemas distribuídos heterogêneos.

Nesse contexto, a interoperabilidade *blockchain* emerge como um problema central. De forma geral, interoperar significa permitir que *blockchains* distintas troquem informações e ativos de maneira segura, consistente e previsível, apesar de suas diferenças estruturais (LU; JAJOO; NAMJOSHI, 2024a). A literatura aponta que esse desafio envolve mais do que simples conectividade: trata-se de estabelecer mecanismos de comunicação entre sistemas autônomos, sem um domínio de confiança compartilhado, preservando propriedades como integridade das mensagens, autenticidade, ordenação de eventos e consistência entre estados (GAUTHIER et al., 2023). Assim, a interoperabilidade pode ser entendida como uma camada adicional de comunicação sobre infraestrutu-

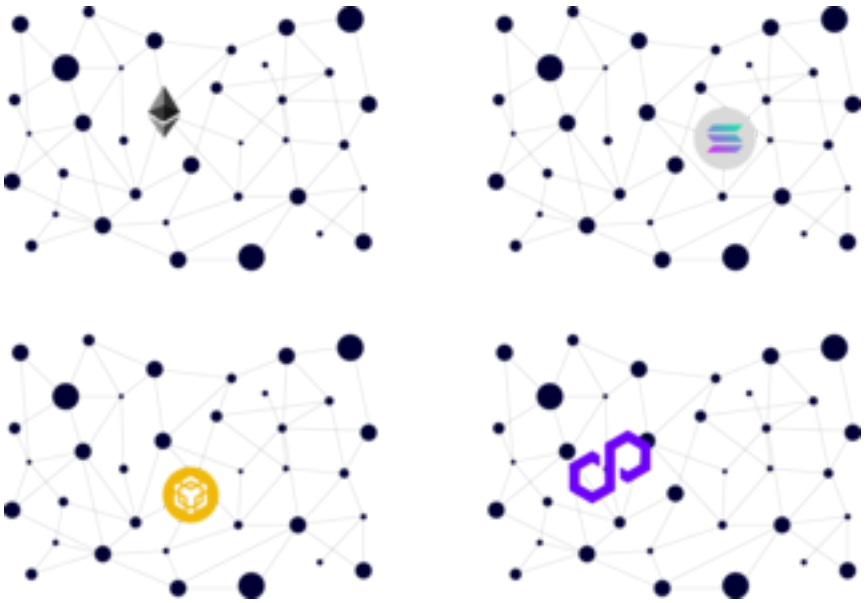

Figura 1.1: Fragmentação do ecossistema blockchain, com múltiplas redes operando de forma isolada.

ras distribuídas que não foram originalmente projetadas para operar de forma integrada (DAVIDSON, 2024).

O problema se aproxima do desenho de protocolos de comunicação entre domínios independentes ou sistemas autonomos, levantando questões clássicas relacionadas a garantias de entrega, latência, tolerância a falhas e definição explícita de hipóteses de confiança (DE; AGRAWAL; ABBADI, 2024). Além disso, a interoperabilidade envolve coordenação entre sistemas com diferentes modelos de execução e finalização de transações, variando entre finalidade probabilística e instantânea, nos quais não existe um consenso global nem mecanismos nativos de sincronização entre estados (ANIDO-RIFÓN, 2025; BELCHIOR et al., 2024; LESAVRE; VARIN; YAGA, 2020). Essas diferenças tornam evidente que soluções de interoperabilidade não são neutras: cada abordagem carrega pressupostos e *trade-offs*, frequentemente analisados sob a ótica do "trilema da interoperabilidade"(segurança, extensibilidade e generalidade), que impactam diretamente a segurança e o comportamento das aplicações (BELCHIOR et al., 2024; ZHAO et al., 2023).

Historicamente, grande parte das soluções de interoperabilidade adotadas na prática baseia-se em *bridges* tradicionais (BELCHIOR et al., 2021). Essas *bridges* atuam como intermediárias entre redes, permitindo a transferência de ativos ou mensagens

por meio de contratos inteligentes, *relayers* e mecanismos externos de verificação (LAWRENCE, 2025a). Embora sejam amplamente utilizadas por sua simplicidade e viabilidade operacional, diversos estudos apontam que tais soluções introduzem fragilidades significativas. Taxonomias de ataques e análises de incidentes mostram que *bridges* tendem a concentrar grandes volumes de valor e dependem de componentes externos, como chaves, validadores ou federações, tornando-se alvos frequentes de falhas de segurança (ZHANG et al., 2024). Além disso, erros de verificação, inconsistências entre estados de origem e destino e dependências operacionais aparecem como padrões recorrentes, reforçando a percepção de que *bridges* representam um ponto crítico na arquitetura de sistemas *cross-chain* (BELCHIOR et al., 2024; ZHAO et al., 2023).

Diante dessas limitações, a literatura mais recente passa a explorar arquiteturas alternativas, incluindo protocolos *multi-chain* e abordagens denominadas *omnichain* (Banaeian Far; Hosseini Bamakan, 2025; ZARICK et al., 2024). Essas propostas buscam ir além da simples transferência de ativos, oferecendo mecanismos de mensageria genérica e abstrações que permitam a composição de aplicações distribuídas entre múltiplas redes (BELCHIOR et al., 2024; LU; JAJOO; NAMJOSHI, 2024a; GAJERA; REDDY; REDDY, 2025). A ideia central é reduzir o acoplamento entre aplicações e mecanismos específicos de interoperabilidade, ao mesmo tempo em que se explicitam as hipóteses de confiança envolvidas (ZARICK et al., 2024; GAUTHIER et al., 2023).

É nesse contexto que surge o *LayerZero*, apresentado como um protocolo de comunicação *omnichain* que separa responsabilidades de envio, verificação e execução de mensagens (ZARICK et al., 2024; ZARICK; PELLEGRINO; BANISTER, 2023a). Diferentemente de *bridges* tradicionais, o *LayerZero* propõe um modelo no qual aplicações podem configurar seus próprios componentes de verificação, ajustando o equilíbrio entre segurança, custo e complexidade (ZARICK et al., 2024; **??**). O interesse pelo protocolo não se limita ao plano conceitual: estudos recentes analisam sua adoção em cenários aplicados, como o funcionamento do *Stargate Bridge* (HUANG; YAN; TESSONE, 2024) e a implementação de mercados e padrões *cross-chain*, incluindo o *Omnichain Fungible Token (OFT)* (ÖZ et al., 2025; ZARICK et al., 2024), evidenciando sua relevância prática no ecossistema atual (ARULKUMARAN et al., 2024; HAN et al., 2024).

Apesar desse crescimento, ainda são limitadas as análises que investigam o *LayerZero* de forma crítica, conectando seu modelo de comunicação às propriedades clássicas estudadas em redes de computadores e sistemas distribuídos. Em particular, permanece a necessidade de compreender de forma mais clara quais garantias o protocolo efetivamente oferece, quais dependências externas são introduzidas e como seus *trade-offs* impactam o desenvolvimento de aplicações cross-chain seguras e previsíveis.

Objetivos

O objetivo geral desta monografia é analisar a interoperabilidade *omnichain* proposta pelo protocolo *LayerZero*, considerando seus aspectos conceituais, arquiteturais e implicações práticas.

Como objetivos específicos, pretende-se:

- contextualizar o problema da interoperabilidade no ecossistema *blockchain*, com base na literatura fundacional sobre o tema;
- discutir limitações e riscos associados às bridges tradicionais, à luz de estudos recentes em segurança;
- descrever a arquitetura e o modelo de comunicação do *LayerZero*, destacando seus principais componentes e pressupostos;
- realizar uma experimentação prática por meio da implementação do padrão OFT, observando o funcionamento do protocolo em um cenário aplicado;
- analisar os *trade-offs* do modelo *omnichain* em termos de segurança, dependências externas e impacto no desenvolvimento de aplicações *cross-chain*.

Organização do trabalho

Esta monografia está organizada da seguinte forma. O Capítulo 2 apresenta a fundamentação teórica, abordando conceitos de interoperabilidade *blockchain*, bridges

tradicionais, questões de segurança e a evolução para arquiteturas *multi-chain* e *omni-chain*. O Capítulo 3 descreve o protocolo *LayerZero*, detalhando sua arquitetura e seu modelo de comunicação *cross-chain*. O Capítulo 4 apresenta a implementação experimental com o padrão OFT e discute observações práticas obtidas a partir desse experimento. O Capítulo 5 reúne a análise e discussão dos resultados, relacionando-os à literatura estudada. Por fim, o Capítulo 6 apresenta as conclusões e aponta possíveis direções para trabalhos futuros.

2 Fundamentação Teórica

Este capítulo consolida os principais conceitos e resultados da literatura relacionados à interoperabilidade entre *blockchains*, com foco em comunicação *cross-chain*, modelos de confiança e segurança. A revisão não se limita a descrever propostas, mas busca confrontar pressupostos, *trade-offs* e implicações práticas sob a ótica de redes de computadores e sistemas distribuídos: garantias de entrega, tolerância a falhas, latência, ordenação de eventos e pontos de centralização. Panoramas e surveys ajudam a organizar o campo e a esclarecer que "interoperabilidade" não é um único problema; trata-se de uma família de objetivos e mecanismos, frequentemente com tensões entre segurança, desempenho e custo operacional (BELCHIOR et al., 2021; BELCHIOR et al., 2024; DENG et al., 2025).

2.1 Interoperabilidade Blockchain

2.1.1 Conceitos básicos e escopo do problema

Em um ecossistema *multi-chain*, interoperabilidade pode ser entendida como a capacidade de dois ou mais sistema distintos cooperarem e trocarem dados de forma legível, apesar das diferenças de linguagens, interfaces e plataformas de execução, (BELCHIOR et al., 2021; GAUTHIER et al., 2023; LAWRENCE, 2025a). Mais do que uma simples conexão, esta propriedade estabelece uma dependência semântica entre registos distintos para transferir valor ou informação, garantindo que as propriedades de integridade, autenticidade e validade sejam preservadas sem a necessidade de intermediários centralizados (BELCHIOR et al., 2021; LESAVRE; VARIN; YAGA, 2020). Na prática, essa definição se desdobra em diferentes objetivos: há soluções voltadas a *transferência de ativos* (focadas na liquidez e migração de valor entre cadeias), outras focadas em *mensageria genérica* (*General Message Passing* - GMP) que permitem a circulação de dados arbitrários e estados remotos, e outras que tentam viabilizar *interações mais ricas*, permitindo a invocação

de contratos inteligentes (*cross-chain calls*) e a execução de aplicações verdadeiramente distribuídas entre redes heterogêneas (DENG et al., 2025; LAWRENCE, 2025a).

Uma leitura crítica recorrente na literatura é que muitos sistemas tratam a comunicação *cross-chain* como mera "conectividade" entre redes (BELCHIOR et al., 2021; DAVIDSON, 2024). Essa visão é insuficiente porque *blockchains* não são apenas canais de mensagens: elas impõem políticas próprias de validação, finalização e execução, o que introduz diferenças de semântica e de superfície de ataque (ANIDO-RIFÓN, 2025; BELCHIOR et al., 2024; GAUTHIER et al., 2023). Assim, o desafio não é apenas "enviar dados"; é tornar *crível* (e verificável) que esses dados têm origem legítima, correspondem a um estado finalizado e serão interpretados corretamente no destino (BELCHIOR et al., 2021; LESAVRE; VARIN; YAGA, 2020).

2.1.2 Tipos de interoperabilidade

A literatura costuma organizar interoperabilidade em, pelo menos, três categorias práticas (SCHULTE et al., 2019; BELCHIOR et al., 2021; ZHENG; LEE; QIAN, 2023; DENG et al., 2025):

- **Transferência de ativos (*asset transfer*):** foco em mover *tokens* e valores entre redes, frequentemente por mecanismos de bloqueio/emissão (lock-and-mint) ou queima/emissão (burn-and-mint) (BELCHIOR et al., 2021; ZHENG; LEE; QIAN, 2023).

- **Mensageria (*messaging*):** foco em transmitir mensagens genéricas, permitindo que contratos em uma rede acionem lógica em outra (*General Message Passing*), com validação do conteúdo (ZARICK et al., 2024; DENG et al., 2025).

- **Interoperabilidade de estado (*state/interactions*):** foco em compor interações *cross-chain* preservando propriedades mais fortes, como atomicidade, coordenação e abstração para aplicações (LU; JAJOO; NAMJOSHI, 2024a; BELCHIOR et al., 2021).

Embora essas categorias ajudem a estruturar o campo, elas também escondem uma tensão: soluções de transferência de ativos podem ser eficazes na movimentação

de liquidez e tokens sem, contudo, oferecer garantias robustas de mensageria ou maturidade para a transmissão de dados arbitrários ou chamadas complexas de funções (ANIDO-RIFÓN, 2025; BELCHIOR et al., 2021; DENG et al., 2025). Em contra partida, as soluções de mensageria podem ser generalistas mas frequentemente introduzem dependências *off-chain* q(como oráculos e relayers) que alteram o modelo de confiança, exigindo que o sistema dependa da honestidade ou da não colusão desses intermediários externos (ANIDO-RIFÓN, 2025; KATE et al., 2025).

2.1.3 Desafios técnicos: finalidade, latência, confiança e heterogeneidade

Quatro desafios aparecem de forma quase universal na literatura:

(i) Finalidade e Reorganização da cadeia de blocos. Redes possuem diferentes mecanismos de consenso e diferentes noções de finalização (ANIDO-RIFÓN, 2025; BELCHIOR et al., 2021; DENG et al., 2025). Para a comunicação *cross-chain*, isso significa que uma prova válida no momento da observação pode ser revertida por reorganizações, dependendo da janela de confirmação adotada e se o modelo é de finalidade determinística ou probabilística (ZHANG et al., 2024).

(ii) Latência e assimetria temporal. Em blockchains, latência não é apenas tempo de rede: ela inclui tempo de confirmação, finalização e propagação de estado (CHEN et al., 2024; KHAN et al., 2023). Em cenários *cross-chain*, a assimetria de tempos entre redes (onde uma direção de transferência pode ser drasticamente mais lenta que a oposta) amplifica riscos econômicos, de segurança e de ordenação, além de afetar a experiência do utilizador e o custo (YAN; HUANG; TESSONE, 2025; EPURE, 2024; CHEN et al., 2024).

(iii) Confiança e verificação. Sem um domínio de confiança comum, o destino precisa de um mecanismo verificável para aceitar (ou rejeitar) mensagens da origem, o que leva a modelos híbridos que combinam provas on-chain com componentes off-chain, como relayers, validadores ou oráculos (SEVIM, 2022; ZHENG; LEE; QIAN, 2023; SCHULTE et al., 2019).

(iv) Heterogeneidade. Diferenças de VM, formatos de prova, modelos de conta

e semânticas de execução afetam o que pode ser verificado e como, dificultando a interoperabilidade nativa (DENG et al., 2025; BELCHIOR et al., 2021). Parte relevante do esforço de interoperabilidade é engenharia de compatibilidade: traduzir eventos e preservar significado entre domínios heterogeneos (LU; JAJOO; NAMJOSHI, 2024a; DENG et al., 2025; YOUNG, 2024).

Esta perspetiva reforça que a interoperabilidade transcende a mera definição de regras de conexão, consolidando-se como um desafio estrutural que integra protocolos de mensageria e arquiteturas modulares (BELCHIOR et al., 2021; BELCHIOR et al., 2024). Esta visão decomposta em camadas — comunicação (transporte), verificação (segurança) e execução (lógica) — permite que a complexidade técnica e a lógica de negócio sejam geridas de forma independente (ZARICK et al., 2024). A seleção dos mecanismos de verificação, sejam eles baseados em validadores externos, redes de oráculos ou provas matemáticas (light clients), altera fundamentalmente o modelo de ameaça ao qual o sistema está exposto (ZHANG et al., 2024; YEDDOU; BELOUCHRANI; MOKRANE, 2024). Consequentemente, os trade-offs operacionais entre latência, custo e descentralização definem as garantias de atomicidade, integridade e finalidade que as aplicações herdam diretamente da infraestrutura subjacente (LU; JAJOO; NAMJOSHI, 2024a; BELCHIOR et al., 2024).

2.2 Bridges Tradicionais

2.2.1 Modelos clássicos e pressupostos operacionais

Bridges tradicionais surgiram como resposta pragmática à demanda por movimentação de ativos e integração rápida entre ecossistemas. Na literatura, três famílias são frequentemente discutidas (BELCHIOR et al., 2021; ZHANG et al., 2024; HUANG; YAN; TESSONE, 2024):

- **Lock–mint:** ativos são bloqueados na origem e “representações” (ativos wrapped ou sintéticos) são emitidas no destino (BELCHIOR et al., 2021; ZHANG et al., 2024; HUANG; YAN; TESSONE, 2024).

- **Burn–mint:** ativos são queimados na origem e reemitidos no destino, exigindo prova de queima para validar a nova cunhagem (HUANG; YAN; TESSONE, 2024; YAN; HUANG; TESSONE, 2025).

- **Redes de liquidez (liquidity networks):** transferências são atendidas por *pools* de liquidez preexistentes em ambas as cadeias, reduzindo o tempo de espera, mas introduzindo riscos de inventário e incentivos específicos para os provedores (ZHANG et al., 2024; ZHENG; LEE; QIAN, 2023; HUANG; YAN; TESSONE, 2024).

Esses modelos compartilham uma característica estrutural comum: a necessidade de componentes intermediários responsáveis por coordenar ou validar eventos entre cadeias distintas. A Figura 2.1 ilustra um modelo clássico de bridge, no qual a comunicação entre redes ocorre por meio de entidades externas ao consenso nativo das blockchains envolvidas, evidenciando a concentração de confiança nesses componentes e seus impactos sobre a segurança e a descentralização do sistema.

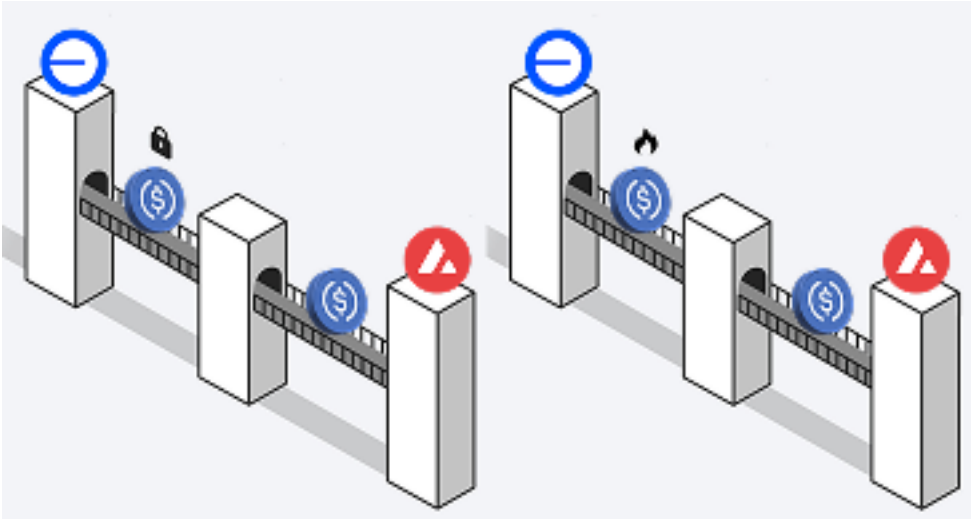

Figura 2.1: Modelo clássico de bridge com concentração de confiança em componentes externos.

O ponto crítico é que, em maior ou menor grau, esses modelos dependem de algum mecanismo de verificação para atestar a veracidade de eventos na origem, tais como depósitos, queimas ou mensagens (DENG et al., 2025; HUANG; YAN; TESSONE, 2024). Este processo frequentemente envolve participantes ou infraestrutura adicional, como redes de oráculos, comitês de validadores externos ou sistemas de computação multipartidária (MPC), que operam fora do consenso nativo de ambas as redes (ZHANG et al.,

2024; ZHAO et al., 2023). Na literatura, esta dependência é apontada como um desafio fundamental, uma vez que a comunicação entre redes isoladas não é tecnicamente viável sem um âncora de confiança (centralizado ou descentralizado) que atue como intermediário (ANIDO-RIFÓN, 2025; BELCHIOR et al., 2021).

2.2.2 O custo da confiança e efeitos arquiteturais

A literatura de segurança mostra que *bridges* tendem a concentrar risco e valor, criando alvos atrativos e sistêmicos: um erro de verificação ou falha operacional pode comprometer grandes volumes de ativos e afetar múltiplas redes conectadas (ZHANG et al., 2024; ZHAO et al., 2023). Em termos de arquitetura de redes, isso equivale a introduzir um *ponto de fragilidade* no canal de comunicação: um componente que se torna "autoridade de fato" para validação *cross-chain*, mesmo quando o sistema se propõe descentralizado (DAVIDSON, 2024; VONEITZEN et al., 2024).

Esse custo aparece de três formas recorrentes: (1) *custódia e centralização*, definindo quem controla as chaves ou assinaturas e como a gestão é processada (LESAVRE; VARIN; YAGA, 2020; ZHANG et al., 2024; VONEITZEN et al., 2024); (2) *superfície de verificação*, que determina o que exatamente é provado (se por observação, certificação ou prova simplificada) e a complexidade do espaço de ação dos validadores (DAVIDSON, 2024; HAN et al., 2024; ILISEI, 2024); e (3) *incentivos econômicos*, avaliando como os participantes são remunerados para alinhar comportamentos racionais e como o custo do ataque é balanceado pelo valor em risco e pelas garantias de estacamento (*staking*)(BELCHIOR et al., 2021; KATE et al., 2025; ÖZ et al., 2025).

2.2.3 Superfícies de ataque recorrentes e problemas práticos

Taxonomias recentes descrevem padrões de falhas em *bridges*, reforçando que incidentes graves raramente são "bugs aleatórios": eles se repetem em superfícies bem definidas, como validação incompleta, assinaturas comprometidas, inconsistência entre estados e manipulação econômica (ZHANG et al., 2024; ZHAO et al., 2023). Além disso, análises empíricas de uso e fluxo de *bridges* ajudam a enxergar como esses mecanismos são utilizados na prática e onde ocorrem gargalos e riscos de rastreabilidade e confiabilidade

(YAN; HUANG; TESSONE, 2025; KHAN et al., 2023).

Também é relevante observar que, mesmo quando a ponte em si é funcional, seu uso pode abrir espaço para exploração de *ordering* e *MEV* (*Maximal Extractable Value*) em cenários *cross-chain*, principalmente quando o sistema depende de *relayers* ou sequenciadores que influenciam a ordem de entrega (ILISEI, 2024; ÖZ et al., 2025).

2.3 Segurança em Comunicação Cross-Chain

2.3.1 Taxonomias e padrões de falha

A pesquisa em segurança *cross-chain* fornece um arcabouço essencial para avaliar propostas além do "*happy path*". Estudos recentes organizam ataques e vulnerabilidades por camadas e por dependências, destacando que o elo fraco costuma estar na fronteira entre validação *on-chain* e componentes *off-chain* (ZHANG et al., 2024; ZHAO et al., 2023). Essas taxonomias são particularmente úteis porque conectam *mecanismos* (provas, relayers, assinaturas, oráculos) a *efeitos* (roubo, duplicação, censura, inconsistência), oferecendo uma linguagem comum para comparar soluções (ZHANG et al., 2024; ZHAO et al., 2023; DENG et al., 2025).

Zhang et al. (2024) identifica 12 vetores de ataque e classifica as falhas em quatro categorias fundamentais: **problemas de permissão** (PI), **erros de lógica** (LI), **anomalias em eventos** (EI) e **falhas de front-end**. Complementarmente, Zhao et al. (2023) organiza a exploração de vulnerabilidades em quatro grandes grupos: **bypass de verificação** (ex: funções fantasmagóricas), **mecanismos de processamento assimétricos** (ex: ataques de re-execução ou *replay*), **roubo de chaves** e vulnerabilidades diversas como a manipulação de *logs*.

Para além das falhas de código, a literatura introduz a análise do **modelo racional-malicioso**. Kate et al. (2025) argumenta que assumir a honestidade de um subconjunto de nós é irrealista face aos incentivos econômicos; em vez disso, os nós devem ser vistos como agentes racionais que podem **coludir para fraudar o sistema** se o lucro da fraude exceder o valor do seu *stake* depositado. Este risco é amplificado pela opacidade das operações, onde a extração de **MEV (*Maximal Extractable Value*)** e arbitragens

não-atómicas introduzem novas superfícies de exploração que comprometem a integridade e a descentralização dos ecossistemas.

2.3.2 Ataques por validação externa, colusão e falhas de verificação

Os três vetores de ataque apresentados refletem as vulnerabilidades mais críticas documentadas na literatura sobre a segurança de pontes e protocolos de interoperabilidade (KATE et al., 2025; ZHANG et al., 2024).

1. **Validação Externa e Dependência de Intermediários.** A validação externa é o mecanismo mais comum em *bridges*, mas introduz pressupostos de confiança significativos ao depender de um conjunto de validadores, redes de oráculos ou computação multi-partidária (MPC) que são externos às redes de origem e destino (ZHANG et al., 2024). Nesses sistemas, a segurança é limitada pela honestidade e pela não colusão desses intermediários (KATE et al., 2025; VONEITZEN et al., 2024). Se o conjunto de validadores for pequeno ou centralizado, o sistema torna-se vulnerável à captura, onde uma maioria maliciosa pode assinar mensagens fraudulentas e drenar ativos. Além disso, falhas operacionais ou indisponibilidade desses nós podem levar à perda de vivacidade da ponte, impedindo transações legítimas de serem concluídas (ZARICK et al., 2024).

2. **Colusão Racional e Comprometimento de Chaves.** A literatura destaca que ataques de colusão e comprometimento são frequentemente motivados por incentivos econômicos diretos. Num modelo de agentes racionais, os participantes podem coludir se o lucro esperado de uma fraude, como mintar ativos sem colateral, superar o valor do seu *stake* ou o custo esperado da penalização (*slashing*) (KATE et al., 2025). Casos reais confirmam esta ameaça:

    - Ronin Network: Um ataque de $625 milhões ocorreu após o comprometimento de chaves privadas em esquema multi-assinatura (5 de 9 validadores);
    - Horizon Bridge: Perdas de $100 milhões foram causadas pelo roubo de duas chaves privadas num sistema de 2-de-4 assinaturas (ZHANG et al., 2024;

ZHENG; LEE; QIAN, 2023).

A centralização da governação e a má gestão de chaves privadas permanecem como os principais vetores para estas explorações sistêmicas (ÖZ et al., 2025; YAN; HUANG; TESSONE, 2025; ZARICK et al., 2024).

3. **Falhas Técnicas de Verificação e Lógica de Contrato.** As falhas de verificação demonstram que a presença de uma prova criptográfica não é garantia absoluta de segurança; a robustez depende da implementação da lógica no destino (ZHAO et al., 2023). Vulnerabilidades recorrentes incluem:

   - Validação incompleta: Falhas em verificar a unicidade de provas permitem ataques de re-execução (*replay attacks*);
   - Emissão incorreta de eventos: Erros onde depósitos de *tokens* sem valor são interpretados como depósitos de ativos valiosos (ex: ataques às pontes Qubit e Meter.io);
   - Funções Fantasma (*Phantom Functions*): Chamadas a funções inexistentes que ativam o *fallback* do contrato, contornando verificações de segurança cruciais (ZHANG et al., 2024; ZHAO et al., 2023);
   - Inconsistência de Estado: Se uma rede intermédia for comprometida num canal multi-hop, ela pode fabricar provas de estado que o destino não consegue validar de forma independente sem correr um nó completo da origem (DAVIDSON, 2024).

   Aqui, a crítica importante é que "ter uma prova" não garante segurança: depende do que é provado e de como o destino interpreta essa prova.

2.3.3 MEV, ordering e efeitos econômicos

Além de vulnerabilidades técnicas, há ataques e extrações de valor baseadas na assimetria de informação, latência e controle de ordenação, um fenomeno conhecido com MEV (*Maximal Extractable Value*). Em ambientes *cross-chain*, a janela temporal e os estados assíncronos entre origem e destino, enquanto tempos de bloco distintos, podem

facilitar estratégias de arbitragem não-atômica e exploração econômica que não existem, ou aparecem com menor intensidade, em contexto single-chain (ÖZ et al., 2025). Estudos empíricos revelam que a proliferação de *bridges* expôs valor extraível adicional, onde arbitragens cíclicas entre redes como Ethereum e Polygon ocorrem diariamente, muitas vezes dominadas por um pequeno grupo de atores especializados (ILISEI, 2024).

Trabalhos que analisam *bridges* em relação a MEV reforçam que o canal de comunicação pode tornar-se um meio de extração ativa. Estratégias como a Arbitragem Dependente de Sequência (SDA) dependem diretamente da infraestrutura das *bridges* para transferir o colateral entre pernas da transação, tornando o tempo de transporte um fator crítico de risco e lucro (ÖZ et al., 2025). Nesses cenários, os validadores da *bridge* detêm um poder de autoridade comparável ao dos validadores da rede nativa, podendo decidir quando e o que é incluído no processo de transferência, o que abre espaço para explorações de sequenciamento e censura (ILISEI, 2024; KATE et al., 2025).

Uma limitação recorrente na literatura é a discussão isolada de ataques, sem uma conexão sistemática entre o modelo de ameaça e as garantias de entrega. Para o desenvolvimento de redes, este vínculo é central: a adoção de um *modular stack design* permite que a aplicação herde a segurança de camadas de mensagens generalistas (GMP), mas exige que o desenvolvedor selecione conscientemente entre modelos de segurança (como multi-assinaturas externas, modelos otimistas ou provas de conhecimento zero), equilibrando os trade-offs entre custo, latência e confiança (BELCHIOR et al., 2024; ZARICK et al., 2024). Sem esta clareza, a aplicação assume riscos implícitos na camada de comunicação que podem comprometer a sua integridade econômica (ZARICK et al., 2024; ZHANG et al., 2024; ZHENG; LEE; QIAN, 2023).

2.4 Protocolos Multi-chain e Omnichain

2.4.1 Arquiteturas *multi-chain*: visão conceitual

Do ponto de vista arquitetural, é fundamental distinguir abordagens multi-chain tradicionais de propostas omnichain, uma vez que essas diferem tanto na forma de interação do usuário com múltiplas redes quanto na abstração oferecida pelas camadas de

interoperabilidade. Enquanto arquiteturas multi-chain tendem a exigir que aplicações e usuários gerenciem explicitamente múltiplas blockchains e seus respectivos estados, abordagens omnichain buscam ocultar essa heterogeneidade por meio de uma camada unificada de comunicação. Essa diferença conceitual é ilustrada na Figura 2.2.

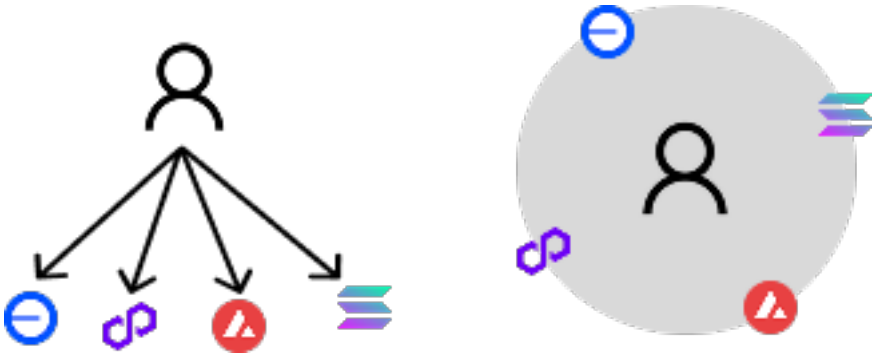

Figura 2.2: Diferença conceitual entre arquiteturas multi-chain e omnichain.

As abordagens *multi-chain* representam uma evolução estrutural no campo, transicionando de conexões ponto a ponto isoladas para ecossistemas integrados que fornecem camadas reutilizáveis de dados, consenso e segurança. Diferente das *bridges* convencionais, estas arquiteturas baseiam-se em frameworks de "blockchains de blockchains"(como Cosmos, Polkadot ou Avalanche), onde cada instância comunica através de um âncora de confiança ou protocolo nativo comum (BELCHIOR et al., 2024). O objetivo central é criar uma *Internet das Blockchains (IoB)*, permitindo que redes independentes colaborem de forma sistemática sem sacrificar a sua soberania (BELCHIOR et al., 2021). Busca-se estruturar a interoperabilidade com componentes e padrões mais claros, como hubs, padrões de mensageria e protocolos dedicados. A motivação é reduzir dependência em pontes ad-hoc e oferecer um caminho mais sistemático para comunicação e composição entre redes (SCHULTE et al., 2019; ZHENG; LEE; QIAN, 2023).

A literatura destaca que este movimento reflete uma mudança de paradigma: a interoperabilidade deixa de ser vista como uma "ponte"temporária e passa a ser tratada como uma infraestrutura de mensageria modular (BELCHIOR et al., 2024). Nesta visão, as responsabilidades são decompostas em camadas funcionais:

1. Roteamento e Transporte: Responsável pela entrega e ordenação de pacotes de

dados opacos entre módulos em diferentes registos (DAVIDSON, 2024; DOTY et al., 2023)

2. Verificação (Segurança): Onde se garante a integridade dos dados através de mecanismos como provas de validade (*zk-SNARKs*) ou clientes leves (*light clients*) que processam provas criptográficas (ZARICK et al., 2024)

3. Execução (Lógica): Onde o destino interpreta a mensagem e executa a tarefa específica da aplicação, separada do contexto de verificação para aumentar a flexibilidade

No entanto, a implementação prática destas arquiteturas costuma envolver custos de coordenação, requisitos de participação e, em alguns casos, a necessidade de infraestrutura adicional que nem sempre se aplica a todas as redes (KHAN et al., 2023; LAWRENCE, 2025a; LAWRENCE, 2025b). Além disso, enfrenta desafios de centralização, pois muitos modelos atuais dependem de uma arquitetura hub-and-spoke. Redes como o *Cosmos Hub* ou a *Axelar* atuam como eixos centrais que, embora facilitem a conectividade em larga escala, podem tornar-se pontos únicos de falha e gargalos de desempenho quando a carga de transações aumenta. Além disso, a heterogeneidade entre diferentes ecossistemas (ex: conectar Polkadot ao Cosmos) ainda exige o desenvolvimento de protocolos de mensageria generalizados (GMP) que consigam traduzir semânticas entre domínios distintos (DAVIDSON, 2024).

Nesta monografia, o interesse por *multi-chain* não é catalogar plataformas, mas explicitar o movimento do campo: da interoperabilidade como "ponte" para interoperabilidade como "infraestrutura de mensageria" com responsabilidades separadas (roteamento, verificação, execução).

2.4.2 Omnichain: abstração e modularidade

Mais recentemente, surgem propostas que enfatizam a *abstração de cadeia* e a *modularidade*, fundamentadas na premissa de que as aplicações (*OApps*) devem interagir com a infraestrutura como se existisse uma camada de comunicação mais uniforme entre redes. Esta visão *omnichain* procura resolver a fragmentação de liquidez e de da-

dos através de *frameworks* que ocultam as nuances técnicas de cada registo distribuído, permitindo que a lógica de negócio seja executada sem modificações em múltiplos ecossistemas. Segundo a literatura, a modularidade é alcançada ao isolar a lógica de execução da camada de verificação, o que concede aos desenvolvedores a liberdade de atualizar as características de segurança sem comprometer a integridade da aplicação (GAJERA; REDDY; REDDY, 2025; ZARICK et al., 2024).

Apesar desta abstração reduzir acoplamento no nível de aplicação, introduz o desafio de uma dependência crítica de infraestruturas externas, como Redes de Verificadores Descentralizadas (DVNs), oráculos e *relayers*, podendo aumentar complexidade e pontos de confiança (ZARICK et al., 2024; VONEITZEN et al., 2024). Embora o protocolo facilite a vida do desenvolvedor ao unificar a semântica da rede, ele exige a aceitação de hipóteses de confiança implícitas, nomeadamente a de que os agentes responsáveis pela verificação (como oráculos e *relayers*) não coludem para comprometer a entrega de mensagens (DE; AGRAWAL; ABBADI, 2024). Além disso, a literatura aponta que o uso de provas criptográficas avançadas, como as de conhecimento zero (ZKPs), embora aumente a segurança e a privacidade, pode introduzir latência e custos computacionais significativos que afetam o desempenho em ambientes de alta cadência (ANIDO-RIFÓN, 2025).

Assim, a abstração não elimina o risco, mas desloca-o da aplicação para a infraestrutura, exigindo que o desenvolvedor avalie se as garantias herdadas são adequadas ao modelo de ameaça da sua aplicação (GAJERA; REDDY; REDDY, 2025). Portanto, a pergunta correta não é apenas se o protocolo facilita a vida do desenvolvedor, mas quais hipóteses precisam ser aceitas para que essa facilidade seja segura e previsível (ZHENG; LEE; QIAN, 2023; SEVIM, 2022).

2.4.3 O protocolo LayerZero no contexto omnichain

Dentro do cenário *omnichain*, o *LayerZero* se destaca por se apresentar como um protocolo de mensageria que separa rigorosamente a camada de verificação da camada de execução e permite configurações de segurança por aplicação. Esta arquitetura permite que as aplicações (OApps) detenham a propriedade exclusiva das configurações de segu-

rança e de custo, utilizando uma infraestrutura de verificação totalmente configurável. Na prática, a integridade do canal depende de componentes que podem ser escolhidos pelo desenvolvedor, tais como Redes de Verificadores Descentralizadas (DVNs) e executores independentes, o que desloca a soberania das decisões de segurança para o nível da aplicação. Ou seja, a proposta envolve um canal de mensagens cuja verificação e entrega dependem de componentes que podem ser escolhidos/configurados (por exemplo, combinações envolvendo oráculos e *relayers*), o que desloca parte das decisões de segurança para o nível da aplicação (ZARICK; PELLEGRINO; BANISTER, 2021; ZARICK; PELLEGRINO; BANISTER, 2023a; ZARICK et al., 2024; ZARICK; PELLEGRINO; BANISTER, 2023b).

Do ponto de vista analítico, o LayerZero é inovador por não impor um único modelo de confiança fixo, permitindo que o sistema admita variações adaptadas a diferentes casos de uso e redes, o que torna a avaliação mais sutil. Esta flexibilidade pode ser uma vantagem para diferentes perfis de aplicação ao permitir que as aplicações escolham entre modelos baseados em provas de conhecimento zero, redes de oráculos ou pontes nativas. No entanto, esta abstração introduz riscos de configurações inadequadas e de dependencias pouco compreendidas. A segurança da aplicação torna-se dependente da honestidade dos validadores externos e da configuração correta do "Security Stack", sendo que falhas de configuração ou colusão entre agentes podem comprometer a validade dos dados (ZARICK et al., 2024; VONEITZEN et al., 2024).

Além disso, a viabilidade do protocolo é sustentada pela sua aplicação em ecossistemas reais, como o *Stargate*, um projeto de liquidez que utiliza o *LayerZero* para permitir transferências de ativos nativos com finalidade garantida instantânea e liquidez unificada entre múltiplas redes EVM (HUANG; YAN; TESSONE, 2024; ZHENG; LEE; QIAN, 2023). O protocolo também tem sido fundamental no desenvolvimento de marketplaces de *NFTs cross-chain*, facilitando o movimento fluido de ativos digitais e dados sem a necessidade de intermediários centralizados (ARULKUMARAN et al., 2024).

Na prática, estes casos de uso reforçam a importância de uma abordagem modular, onde o desenvolvedor deve equilibrar conscientemente os *trade-offs* entre custo, latência e confiança inerentes ao desenho de redes interoperáveis. Além disso, esses tra-

balhos também reforçam a motivação desta monografia: compreender o protocolo não apenas pelo discurso de "interoperabilidade", mas pelas garantias que ele oferece, pelas dependências que introduz e pelos *trade-offs* que impõe ao desenho de aplicações distribuídas entre cadeias (ZARICK et al., 2024; ZHANG et al., 2024; ZHENG; LEE; QIAN, 2023).

3 O Protocolo LayerZero

Este capítulo apresenta uma análise conceitual e arquitetural do protocolo *LayerZero*, com o objetivo de compreender sua proposta de interoperabilidade omnichain à luz das soluções já consolidadas na literatura. A discussão é conduzida sob a perspectiva de redes de computadores e sistemas distribuídos, enfatizando aspectos como desenho de protocolos, separação de responsabilidades, modelos de confiança, garantias de entrega e dependência de componentes externos.

Diferentemente de abordagens que tratam a interoperabilidade apenas como transferência de ativos, o *LayerZero* propõe uma camada genérica de mensageria *cross-chain*, o que amplia significativamente o espaço de aplicações possíveis, mas também introduz novos *trade-offs* de segurança e engenharia.

O *LayerZero* é definido na literatura como um protocolo de interoperabilidade *omnichain* concebido para conectar redes *blockchain* de forma descentralizada, eliminando a necessidade de intermediários centrais (ANIDO-RIFÓN, 2025; SEVIM, 2022). Ao contrário das abordagens de "*middle chain*", que introduzem um novo consenso para validar mensagens, o *LayerZero* utiliza *Ultra-Light Nodes* (ULNs), contratos inteligentes que funcionam como nós leves, para verificar a validade das transações entre cadeias de forma eficiente e com baixo custo de armazenamento (HAN et al., 2024). A sua arquitetura modular separa rigorosamente a camada de verificação da camada de execução, permitindo que as aplicações *omnichain* (OApps) detenham a propriedade exclusiva das suas configurações de segurança e custos (ZARICK et al., 2024).

O modelo de confiança do protocolo assenta na independência entre dois atores principais: o Oráculo, responsável por transportar o cabeçalho do bloco da rede de origem, e o *Relayer* (ou executor), que se encarrega de fornecer a prova de transação correspondente (HAN et al., 2024; DE; AGRAWAL; ABBADI, 2024; ZHANG et al., 2024). A integridade do sistema é garantida sob a premissa de que não ocorre colusão entre estas entidades, uma vez que é matematicamente inviável validar uma prova de transação contra um cabeçalho de bloco incorreto, e vice-versa (DENG et al., 2025; ZARICK; PEL-

LEGRINO; BANISTER, 2023a). Esta abordagem permite que o protocolo apresente semântica de rede universal, sendo agnóstico à infraestrutura ou à *blockchain* subjacente e facilitando a integração de novas redes de forma "*plug-and-play*"(ZARICK et al., 2024; ZARICK; ZHANG, 2025).

No entanto, a proposta de uma camada genérica de mensageria introduz novos *trade-offs* de engenharia e segurança. A flexibilidade do protocolo permite que os desenvolvedores escolham entre diferentes Redes de Verificadores Descentralizadas (DVNs) e métodos de prova, como ZKPs ou oráculos tradicionais. Mas esta escolha desloca a responsabilidade do risco para o nível da aplicação (ZARICK et al., 2024). Depender de validadores externos pode introduzir vulnerabilidades sistémicas se os agentes de verificação forem comprometidos ou se a configuração da "*Security Stack*"for inadequada (ANIDO-RIFÓN, 2025; VONEITZEN et al., 2024). Além disso, enquanto as pontes tradicionais se limitam a transferências de ativos via *lock-and-mint*, o *LayerZero* funciona como uma primitiva de comunicação de baixo nível, possibilitando aplicações complexas como governação *cross-chain*, empréstimos multi-rede e agregação de rendimentos em tempo real (ZARICK et al., 2024; SEVIM, 2022).

3.1 Visão Geral do LayerZero

Diferentes abordagens para a troca de ativos e informações entre blockchains refletem distintas escolhas arquiteturais e pressupostos de confiança. Desde modelos centralizados, nos quais uma entidade intermediária coordena operações fora da cadeia, até soluções descentralizadas baseadas em contratos inteligentes ou camadas intermediárias, essas arquiteturas variam quanto à distribuição de responsabilidades, acoplamento entre redes e dependência de componentes externos. A Figura 3.1 apresenta uma comparação conceitual entre uma exchange centralizada, uma exchange descentralizada tradicional e uma solução construída sobre o protocolo LayerZero, destacando as diferenças na forma como a comunicação entre cadeias é estruturada.

O crescimento do ecossistema blockchain resultou em um ambiente fortemente fragmentado, no qual múltiplas redes coexistem com diferentes modelos de consenso, tempos de finalização, linguagens de contrato e políticas de validação (LAWRENCE,

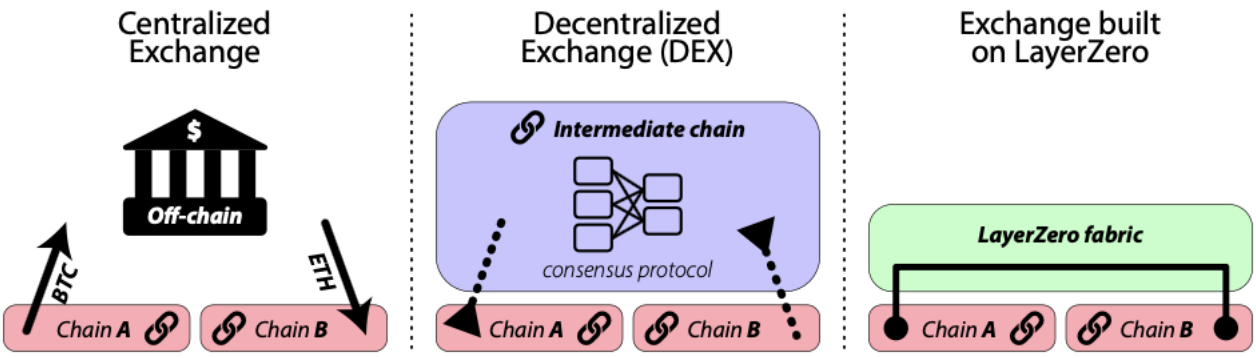

Figura 3.1: Diferenças arquiteturais entre uma exchange centralizada, uma descentralizada e uma entre cadeias utilizando(ZARICK; PELLEGRINO; BANISTER, 2021)

2025a; SCHULTE et al., 2019; ZHENG; LEE; QIAN, 2023). As redes individuais operam frequentemente como "ilhas"ou"jardins murados"que não conseguem comunicar de forma nativa (HUANG; YAN; TESSONE, 2024; LAWRENCE, 2025a; ZHAO et al., 2023). Nesse cenário, soluções de interoperabilidade surgem como tentativas de mitigar o isolamento entre redes, viabilizando a transferência de ativos, dados e chamadas de função entre blockchains heterogêneas (DAVIDSON, 2024; SEVIM, 2022).

As primeiras soluções amplamente adotadas foram as chamadas *bridges tradicionais*, que normalmente implementam padrões como *lock–mint* ou *burn–mint* (HUANG; YAN; TESSONE, 2024; ZHANG et al., 2024; ZHENG; LEE; QIAN, 2023). Apesar de sua simplicidade operacional, essas abordagens concentraram grandes volumes de valor em contratos ou federações específicas, tornando-se alvos recorrentes de ataques (KATE et al., 2025; ZHANG et al., 2024). Estudos empíricos e relatórios de segurança mostram que falhas nestas infraestruturas representam uma parcela significativa das perdas financeiras em interoperabilidade, acumulando prejuízos superiores a 3,2 bilhões de dólares (BELCHIOR et al., 2024).

O LayerZero surge nesse contexto como uma proposta alternativa, cujo objetivo principal é oferecer uma camada de comunicação *omnichain* mais flexível e modular (ARULKUMARAN et al., 2024; VONEITZEN et al., 2024; ZARICK et al., 2024). Em vez de impor um modelo único de validação ou um conjunto fixo de participantes confiáveis, o protocolo permite que aplicações (OApps) escolham explicitamente como mensagens

cross-chain serão verificadas e executadas através da configuração do seu próprio *Security Stack* (ZARICK; PELLEGRINO; BANISTER, 2021; ZARICK; PELLEGRINO; BANISTER, 2023a). Esta filosofia, que separa rigorosamente a camada de verificação da camada de execução, aproxima o *LayerZero* de um protocolo de transporte de mensagens de baixo nível, mais do que de uma bridge no sentido tradicional.

Do ponto de vista conceitual, essa abordagem diferencia o *LayerZero* de soluções como *Polkadot* e *Cosmos*, que adotam arquiteturas baseadas em *hubs* ou cadeias centrais (como a Relay Chain ou o Cosmos Hub) que podem atuar como pontos únicos de falha ou gargalos de desempenho (DAVIDSON, 2024). Diferencia-se também de soluções como *CCIP* (*Chainlink*) e *Axelar*, que fornecem serviços de mensageria e validação fortemente apoiados em redes externas de validadores ou redes de oráculos descentralizadas (DONs) que detêm autoridade sobre o estado da mensagem. No LayerZero, a confiança é reduzida à hipótese de não-colusão entre duas entidades independentes: o Oráculo (ou DVN) e o Relayer (ou Executor) (ZARICK et al., 2024; DE; AGRAWAL; ABBADI, 2024; ZHENG; LEE; QIAN, 2023).

3.2 Arquitetura do LayerZero

A arquitetura do LayerZero é composta por um conjunto de elementos que explicitam a separação entre envio, verificação e execução de mensagens. Essa separação é um aspecto central do protocolo e dialoga diretamente com princípios clássicos da engenharia de redes, como modularidade e desacoplamento funcional, permitindo que a lógica de funcionalidade seja isolada da segurança da verificação (ZARICK et al., 2024).

O principal componente do protocolo é o *Endpoint*, implantado em cada blockchain participante. O endpoint funciona como a interface entre a aplicação e o LayerZero, sendo composto por módulos internos designados como Communicator, Validator e Network. Ele é responsável por registar as mensagens de saída e receber as mensagens de entrada, mas é importante notar que o endpoint não valida, por si só, a veracidade das mensagens, delegando essa tarefa a bibliotecas de verificação (DE; AGRAWAL; ABBADI, 2024; HAN et al., 2024; ZARICK et al., 2024)

A verificação é delegada a dois tipos de entidades externas independentes: o

Oracle e o *Relayer*, que nas versões mais recentes evoluíram para o conceito de Redes de Verificação Descentralizadas (*Decentralized Verification Networks - DVNs*) (ZARICK et al., 2024; ZARICK; PELLEGRINO; BANISTER, 2023b; DAVIDSON, 2024; ZHANG et al., 2024). O oracle fornece ao destino informações sobre o estado da blockchain de origem, especificamente os cabeçalhos de bloco, enquanto o *relayer* transmite a prova de transação, geralmente uma prova de Merkle, específica da mensagem enviada pela aplicação (DE; AGRAWAL; ABBADI, 2024; HUANG; YAN; TESSONE, 2024; DENG et al., 2025). A segurança do sistema baseia-se na premissa de não-colusão entre estas duas entidades (ZARICK et al., 2024; ZARICK; ZHANG, 2025).

A interação entre esses componentes pode ser compreendida de forma mais clara por meio de uma visão estrutural do fluxo de mensagens cross-chain. A Figura 3.2 apresenta o esquema arquitetural do LayerZero, destacando a separação explícita entre os contratos endpoint implantados nas blockchains de origem e destino e os componentes off-chain responsáveis pela verificação das mensagens, evidenciando o desacoplamento entre envio, validação e execução que caracteriza o protocolo.

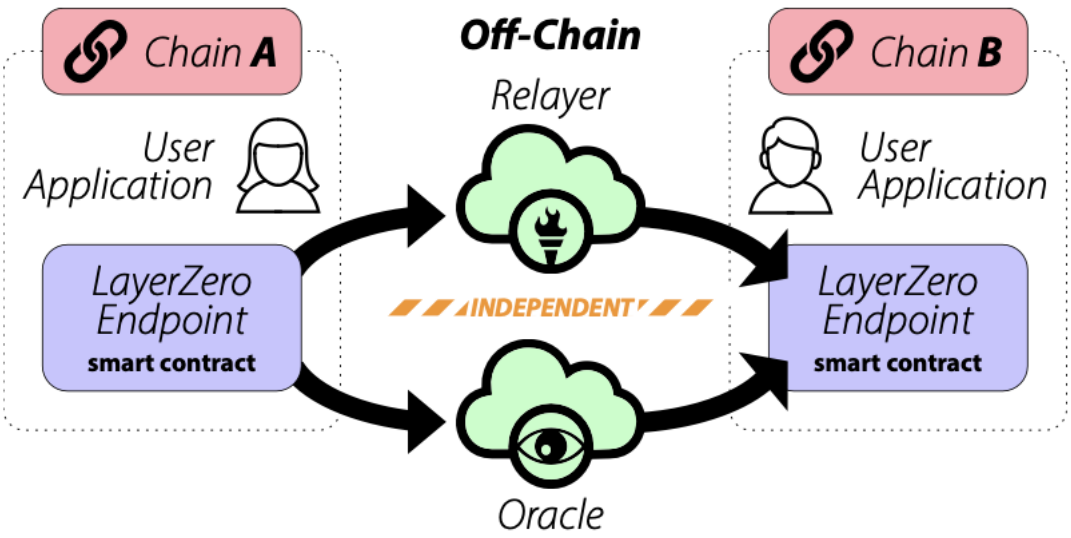

Figura 3.2: Esquema arquitetural do LayerZero.(ZARICK; PELLEGRINO; BANISTER, 2021)

Além destes componentes, o LayerZero introduz o conceito de *Ultra Light Nodes (ULNs)*, que permitem verificar mensagens através de contratos inteligentes que funcionam como nós leves eficientes, sem a necessidade de manter uma cópia completa do estado

da blockchain de origem (ANIDO-RIFÓN, 2025; Banaeian Far; Hosseini Bamakan, 2025; YEDDOU; BELOUCHRANI; MOKRANE, 2024). Ao contrário dos nós leves tradicionais que armazenam todos os cabeçalhos sequencialmente, os ULNs utilizam oráculos para realizar o streaming de cabeçalhos sob demanda, garantindo o mesmo nível de segurança sem o custo proibitivo de armazenamento (ARULKUMARAN et al., 2024; ZHENG; LEE; QIAN, 2023). Esse modelo busca um equilíbrio entre eficiência e segurança, evitando tanto o custo elevado de *full nodes* quanto a fragilidade de validações puramente *off-chain* ou de soluções centralizadas de *middle chain*.

O *Executor* é o componente responsável por efetivar a execução da lógica da aplicação no destino após a verificação bem-sucedida da mensagem pela camada de segurança (ZARICK et al., 2024; DAVIDSON, 2024). Esta arquitetura explicita que o protocolo atua essencialmente como uma camada de transporte (primitiva de comunicação de baixo nível), enquanto a semântica e a configuração de segurança da aplicação permanecem sob o controle exclusivo do desenvolvedor (CHEN et al., 2024; YAN; HUANG; TESSONE, 2025; ZARICK et al., 2024).

3.3 Modelo de Comunicação Cross-Chain

O fluxo de comunicação no *LayerZero* pode ser decomposto em três etapas principais: envio, verificação e execução. Esta arquitetura reflete princípios de sistemas distribuídos ao isolar as responsabilidades de transporte, integridade de dados e lógica aplicacional. (DE; AGRAWAL; ABBADI, 2024; ZARICK et al., 2024).

Na etapa de envio, a aplicação na *blockchain* de origem invoca o *endpoint* local, composto pelos módulos *Communicator*, *Validator* e *Network*, que registra a mensagem, atribui-lhe um identificador único global (GUID) e emite eventos que são monitorizados por entidades externas, os oracles e relayers (HAN et al., 2024; ZARICK et al., 2024; VONEITZEN et al., 2024).

Em seguida, na fase de verificação, o protocolo depende da independência entre duas entidades: o *Oracle*, que transmite o cabeçalho do bloco ou o estado da cadeia de origem; e o *Relayer*, que fornece a prova associada à mensagem específica. Apenas quando ambas as informações são recebidas e consideradas consistentes o endpoint de

destino aceita a mensagem (ZARICK et al., 2024; HAN et al., 2024; ZHANG et al., 2024). O *endpoint* de destino só aceita e considera a mensagem válida se houver uma concordância matemática entre a prova enviada pelo relator e o cabeçalho fornecido pelo oráculo, garantindo que não houve colusão (VONEITZEN et al., 2024; DE; AGRAWAL; ABBADI, 2024).

A execução ocorre quando um Executor (entidade *permissionless*) aciona a função de *callback* correspondente na aplicação de destino para efetivar a lógica *cross-chain* pretendida. Este modelo modular permite uma forma de tolerância a falhas por decomposição funcional, uma vez que erros na lógica de execução de uma aplicação não comprometem a integridade do canal de comunicação nem a segurança das outras aplicações (ZARICK et al., 2024).

O encadeamento dessas etapas pode ser melhor compreendido por meio de uma representação explícita do fluxo de mensagens entre as cadeias envolvidas. A Figura 3.3 ilustra o modelo de comunicação cross-chain do LayerZero em uma única transação, evidenciando a atuação coordenada do endpoint na cadeia de origem, das entidades externas de verificação (oracle e relayer) e do endpoint na cadeia de destino, bem como os pontos em que ocorrem o registro, a validação e a execução da mensagem.

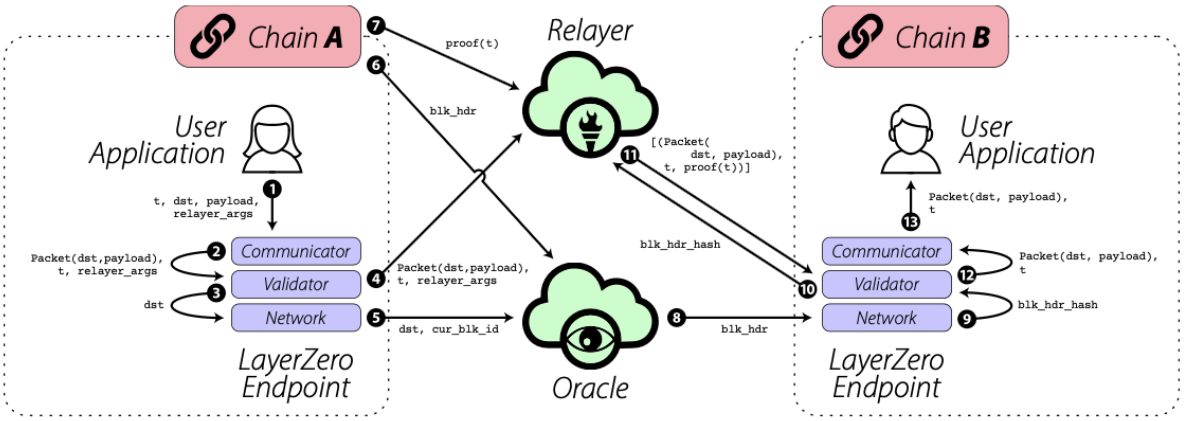

Figura 3.3: Fluxo de comunicação em uma única transação entre cadeias usando LayerZero.(ZARICK; PELLEGRINO; BANISTER, 2021)

Diferentemente de protocolos como o IBC (*Inter-Blockchain Communication*), que exigem garantias fortes de ordenação e entrega baseadas na sincronização total e na-

tiva de cabeçalhos entre cadeias determinísticas (DE; AGRAWAL; ABBADI, 2024; KHAN et al., 2023; SEVIM, 2022), o *LayerZero* oferece garantias condicionais, dependentes do modelo de verificação configurado pela própria aplicação, permitindo ao desenvolvedor escolher os verificadores (DVNs) e os custos associados (ZARICK et al., 2024; ZHENG; LEE; QIAN, 2023). Embora esta flexibilidade aumente a escalabilidade, ela transfere a "propriedade exclusiva"e parte da responsabilidade da segurança para o nível da aplicação, exigindo que o desenvolvedor avalie se o conjunto de verificadores escolhido é suficientemente resiliente à colusão.

3.4 Modelo de Confiança e Trade-offs

O modelo de confiança do LayerZero baseia-se na hipótese de não colusão entre *Oracle* e *Relayer* (ou DVNs) (DE; AGRAWAL; ABBADI, 2024; DENG et al., 2025). Esta suposição assenta na premissa de que é estatisticamente impossível validar uma prova de transação contra um cabeçalho de bloco sem o conhecimento prévio desse cabeçalho e vice-versa. Contudo, as fontes indicam que este mecanismo não equivale a um modelo puramente *trustless*, uma vez que a segurança do sistema depende da honestidade e fiabilidade dos validadores externos selecionados (ANIDO-RIFÓN, 2025; VONEITZEN et al., 2024).

Quando comparado a soluções como o *Wormhole*, que dependem de um conjunto fixo de guardiões (modelo PoA), o *LayerZero* reduz a concentração de poder em uma única entidade ao permitir que as aplicações (OApps) definam o seu próprio conjunto de verificadores. No entanto, ainda se mantém uma dependência explícita de componentes *off-chain*, o que introduz vulnerabilidades se os agentes de verificação forem comprometidos (ZARICK et al., 2024; VONEITZEN et al., 2024). Em relação ao *Stargate*, que operam como *bridges* baseadas em liquidez *omnichain*, o *LayerZero* posiciona-se em um nível mais fundamental, como a primitiva de comunicação de baixo nível (Layer-0), fornecendo a camada de mensageria sobre a qual esta ponte de liquidez é construída para permitir transferências de ativos nativos com finalidade instantânea (HUANG; YAN; TESSONE, 2024; YAN; HUANG; TESSONE, 2025; ZHANG et al., 2024).

Comparado as arquiteturas baseadas em *hubs*, como *Polkadot* e *Cosmos*, o *Layer-*

Zero evita a necessidade de uma cadeia central ou de um consenso intermediário, reduzindo o acoplamento estrutural e eliminando pontos únicos de falha ou gargalos de desempenho típicos de modelos hub-and-spoke (LAWRENCE, 2025a; LAWRENCE, 2025b). No entanto, essa escolha implica abrir mão de garantias globais de segurança fornecidas por um conjunto comum de validadores, como a segurança partilhada da *Relay Chain* do Polkadot (VONEITZEN et al., 2024).

Enquanto em soluções como *CCIP* e *Axelar*, a segurança está fortemente ancorada em redes externas de validadores e oráculos (DONs) ou numa *middle chain* com validadores próprios, no LayerZero a aplicação detém a propriedade exclusiva das suas configurações de segurança, podendo escolher diferentes combinações de verificadores (ARULKUMARAN et al., 2024; DE; AGRAWAL; ABBADI, 2024). Este modelo oferece uma flexibilidade sem precedentes, permitindo equilibrar o custo e a confiança, mas exige que o desenvolvedor assuma a responsabilidade de garantir que os oráculos e relatores escolhidos são independentes e não coludem (ZARICK et al., 2024).

Em suma, o LayerZero pode ser interpretado como uma camada de comunicação configurável, que fornece mecanismos básicos de entrega de mensagens, mas delega garantias mais fortes — como confiabilidade, ordenação e tolerância a falhas bizantinas — às escolhas de projeto feitas no nível da aplicação. Essa característica representa simultaneamente uma oportunidade e um desafio, reforçando a necessidade de análises críticas e experimentações práticas, como as desenvolvidas nos capítulos seguintes.

4 Implementação Experimental com OFT

Este capítulo apresenta a implementação experimental realizada com o objetivo de observar, em um cenário prático, o funcionamento do protocolo LayerZero. Diferentemente dos capítulos anteriores, que se concentram na análise conceitual e arquitetural da interoperabilidade blockchain, esta etapa busca confrontar a teoria com uma aplicação real, evidenciando custos operacionais, complexidades de configuração e decisões de segurança impostas ao desenvolvedor.

A aplicação desenvolvida é uma implementação de um *Omnichain Fungible Token* (OFT) utilizando o protocolo *LayerZero V2* para comunicação *cross-chain*. O objetivo principal é permitir a transferência nativa de tokens entre diferentes blockchains EVM de forma segura e descentralizada.

A escolha pela implementação de um OFT fundamenta-se na intenção de analisar um dos casos de uso mais recorrentes da interoperabilidade blockchain: a transferência de ativos fungíveis entre diferentes redes. Tokens fungíveis representam uma classe amplamente utilizada em aplicações descentralizadas, especialmente em contextos de finanças descentralizadas (DeFi), o que torna esse experimento representativo do uso prático do protocolo.

O experimento teve como objetivos observar o fluxo real de uma transferência cross-chain mediada pelo LayerZero; analisar os custos de gas associados às etapas de envio, verificação e execução da mensagem; identificar a complexidade operacional envolvida na configuração do protocolo; e, por fim, avaliar as dependências externas introduzidas pelo modelo omnichain.

4.1 Tecnologias Utilizadas

As tecnologias empregadas neste trabalho foram escolhidas com o objetivo de viabilizar a implementação prática do experimento e, ao mesmo tempo, permitir uma análise técnica consistente do protocolo LayerZero em um ambiente real de desenvolvimento.

Os contratos inteligentes foram desenvolvidos em **Solidity (versão 0.8.22)**, linguagem padrão para aplicações no ecossistema de EVMs, escolhida pela maturidade da linguagem e pelos mecanismos de segurança incorporados nas versões mais recentes.

A comunicação cross-chain foi implementada por meio do **LayerZero V2**, protocolo de mensageria omnichain que constitui o objeto central deste estudo.

O ambiente de desenvolvimento e *deploy* foi estruturado com o **Hardhat**, utilizado para compilação, gerenciamento de redes de teste e execução de scripts de implantação. De forma complementar, o **Foundry** foi empregado em etapas de compilação e testes, oferecendo uma alternativa para validação do comportamento dos contratos.

Os scripts auxiliares de *deploy* e configuração foram escritos em **TypeScript**, facilitando a organização do código e a automação das etapas experimentais. Além disso, foram utilizadas bibliotecas da **OpenZeppelin**, especialmente implementações consolidadas de padrões como ERC20 e módulos de controle de acesso, com o objetivo de reduzir riscos associados a implementações manuais e concentrar a análise nos aspectos de interoperabilidade.

4.2 Arquitetura do Projeto

O contrato `MyOFT` estende o padrão OFT da LayerZero:

```
contract MyOFT is OFT {
    constructor(
        string memory _name,
        string memory _symbol,
        address _lzEndpoint,
        address _owner
    ) OFT(_name, _symbol, _lzEndpoint, _owner) Ownable(_owner) {
        _mint(_owner, 1000000000000000000000000); // 1 milhao de tokens
    }
}
```

O contrato implementa o padrão ERC-20 completo, integra com o *EndpointV2*

da *LayerZero* para mensageria *cross-chain*, além de permitir a queima (*burn*) na *chain* de origem e cunhagem (*mint*) na *chain* de destino automaticamente

O contrato foi implantado em duas redes EVM em ambiente de teste: **Base Sepolia** (endpoint ID 40245), utilizada como cadeia de origem, e **Avalanche Fuji** (endpoint ID: 40106), por sua vez, como cadeia de destino. A escolha por testnets se justifica pela possibilidade de experimentação sem custos financeiros reais, além de facilitar a inspeção detalhada das transações.

O funcionamento desse experimento pode ser melhor compreendido por meio de uma visão sequencial do processo de transferência cross-chain. A Figura 4.1 apresenta o fluxo completo de uma operação de envio utilizando o padrão Omnichain Fungible Token (OFT), desde a chamada da função `send()` na cadeia de origem até a execução do `lzReceive()` na cadeia de destino, evidenciando as etapas de queima (burn), verificação pelo protocolo LayerZero e cunhagem (mint) dos tokens no destino.

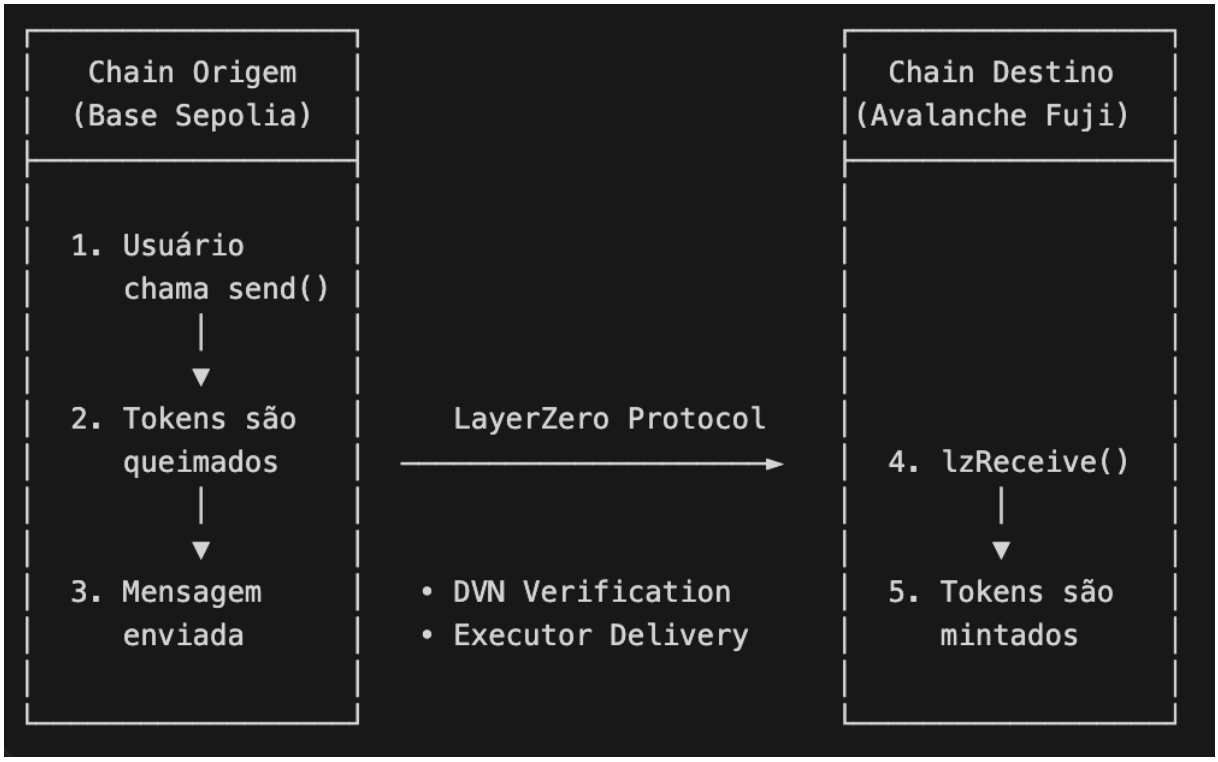

Figura 4.1: Fluxo de Transferência Cross-Chain.

Além disso, estrutura do projeto foi organizada de forma a separar claramente as responsabilidades entre contrato, configuração, automação e testes, facilitando tanto

a implementação do experimento quanto a análise do funcionamento do protocolo LayerZero.

O contrato `contracts/MyOFT.sol` concentra a lógica principal do token omnichain, sendo responsável pelas operações de queima, emissão e integração com o endpoint do LayerZero.

O processo de implantação foi automatizado por meio do script `deploy/MyOFT.ts`, que define as etapas de deploy do contrato nas redes de teste utilizadas.

As definições relacionadas à interoperabilidade, como configuração de pathways e escolha dos *Decentralized Verifier Networks* (DVNs), foram centralizadas no arquivo `layerzero.config.ts`, permitindo ajustar parâmetros de comunicação e segurança de forma explícita.

Para a execução das transferências cross-chain durante o experimento, foi utilizada a task `tasks/sendOFT.ts`, integrada ao Hardhat, responsável por acionar o fluxo de envio do token entre redes distintas.

Por fim, o diretório `test/` reúne os testes unitários e de integração, empregados para validar o comportamento esperado dos contratos e auxiliar na identificação de erros durante o desenvolvimento e a execução do experimento.

4.3 Configuração de Segurança e Funcionalidades

O projeto utiliza o DVN (*Decentralized Verifier Network*) da LayerZero Labs para verificação das mensagens cross-chain:

```
const pathways: TwoWayConfig[] = [
    [
        BaseSepoliaContract,
        AvalancheContract,
        [['LayerZero Labs'], []], // DVN obrigatorio
        [1, 1], // Confirmacoes necessarias
        [EVM_ENFORCED_OPTIONS, EVM_ENFORCED_OPTIONS],
    ],
```

]

Além da configuração de segurança, o projeto implementa um deploy automatizado multi-chain; configuração bidirecional de pathways (wiring); transferência de tokens cross-chain via CLI; suporte a Simple Workers para testnets sem DVNs nativos; testes unitários com mocks do EndpointV2; e,rastreamento de transações via LayerZero Scan.

4.4 Principais Comandos

Para a condução do experimento foram utilizados comandos específicos do toolchain do LayerZero, responsáveis pela criação do projeto, implantação dos contratos e execução de transferências cross-chain. A seguir são apresentados os principais comandos empregados, com o objetivo de documentar o fluxo operacional adotado.

```
# Criacao de um projeto com base em OFT
npx create-lz-oapp@latest --example oft

# Deploy do contrato
pnpm hardhat lz:deploy
    --tags MyOFT

# Configurar conexoes entre chains
pnpm hardhat lz:oapp:wire
    --oapp-config layerzero.config.ts

# Transferir tokens cross-chain
pnpm hardhat lz:oft:send
    --src-eid 40245
    --dst-eid 40106
    --amount <QUANTIDADE>
    --to <ENDERECO>
```

4.5 Resultados

4.5.1 Detalhamento do Contrato MyOFT

A Tabela 4.1 resume as principais especificações técnicas do contrato MyOFT utilizado no experimento, incluindo versão do compilador, padrão de token adotado, política de decimais e configuração de otimização. Esses parâmetros definem o escopo da implementação e garantem a reprodutibilidade dos resultados apresentados nas seções seguintes.

Parâmetro	**Valor**
Versão Solidity	0.8.22
Padrão de Token	ERC-20 + OFT (LayerZero)
Supply Inicial	1.000.000 tokens (10^{24} wei)
Decimais	18 (local)/ 6 (shared decimals para cross-chain)
Otimizador	Enabled, 200 runs

Tabela 4.1: Especificações Técnicas do Contrato

4.5.2 Hierarquia de Herança do Contrato

A hierarquia de herança do contrato MyOFT foi definida de forma a isolar responsabilidades e reutilizar componentes consolidados, tanto do protocolo *LayerZero* quanto da biblioteca *OpenZeppelin*. O esquema apresentado a seguir ilustra a composição do contrato a partir do padrão OFT, destacando a integração entre módulos de mensageria (*OAppSender* e *OAppReceiver*), o núcleo de comunicação (*OFTCore*) e as abstrações padrão de token e controle de acesso.

```
MyOFT
+-- OFT (LayerZero)
    +-- OFTCore
    |   +-- OAppSender
    |   +-- OAppReceiver
    |   +-- OAppCore
    +-- ERC20 (OpenZeppelin)
```

```
+-- Ownable (OpenZeppelin)
```

Observa-se que a lógica específica da aplicação permanece concentrada no contrato MyOFT, enquanto aspectos de comunicação cross-chain e controle de acesso são herdados de módulos especializados, reforçando a separação de responsabilidades adotada no projeto.

A fim de explicitar como o modelo omnichain do LayerZero se materializa na interface do contrato, a Tabela 4.2 apresenta as principais funções herdadas e utilizadas pelo MyOFT. Essas funções refletem diretamente o fluxo de comunicação cross-chain descrito no Capítulo 3, abrangendo envio, verificação e recebimento de mensagens entre cadeias.

Função	**Tipo**	**Descrição**
`send()`	payable	Envia tokens cross-chain
`quoteSend()`	view	Estima taxa de envio
`lzReceive()`	payable	Recebe tokens de outra chain
`setPeer()`	onlyOwner	Configura contratos pareados
`setEnforcedOptions()`	onlyOwner	Define opções de gas obrigatórias

Tabela 4.2: Funções Principais Herdadas

4.5.3 Deploy do Contrato

Para fins de validação experimental e rastreabilidade das operações realizadas, a Tabela 4.3 apresenta os endereços dos contratos MyOFT implantados nas redes de teste Base Sepolia e Avalanche Fuji, utilizados como cadeias de origem e destino, respectivamente.

Rede	**Endereço**
Base Sepolia	0xd235b72BC4f2b89507C4649f2835c3698e321372[1]
Avalanche Fuji	0x97248f2bD1b0315AE72Da4b12215f2b108CbdaFf[2]

Tabela 4.3: Endereços dos Contratos Implantados

[1]Disponível em: https://sepolia.basescan.org/address/0xd235b72BC4f2b89507C4649f2835c3698e321372
[2]Disponível em: https://testnet.snowtrace.io/token/0x97248f2bD1b0315AE72Da4b12215f2b108CbdaFf

4.5.4 Custo de Gas

Gas é uma medida de trabalho computacional na EVM (Ethereum Virtual Machine), calculado pela razão entre a taxa da transação e o preço do gas.

O custo computacional associado à implantação do contrato é um fator relevante na avaliação prática de soluções cross-chain. A Tabela 4.4 apresenta os valores de gas consumidos durante o processo de deploy do contrato MyOFT nas redes de teste consideradas, permitindo analisar diferenças de custo e implicações econômicas entre os ambientes avaliados.

Rede	**Gas Usado**	**Gas Price**	**Transaction Fee**	**Transaction Hash**
Base Sepolia	2.933.454	0.00144 Gwei	0.00000422 ETH	0x60c7426a...75c2255
Avalanche Fuji	2.933.454	0.082	0.0116 AVAX	0x672bf4c8...60efe07

Tabela 4.4: Custo de Gas para Deploy do Contrato

Para complementar os valores apresentados na Tabela 4.4 e garantir a rastreabilidade dos resultados experimentais, a Figura 4.2 apresenta os detalhes da transação de deploy do contrato MyOFT na rede Base Sepolia, conforme registrados no explorador de blocos. Essa visualização permite verificar informações como consumo de gas, taxa aplicada e hash da transação, reforçando a validade empírica dos dados coletados.

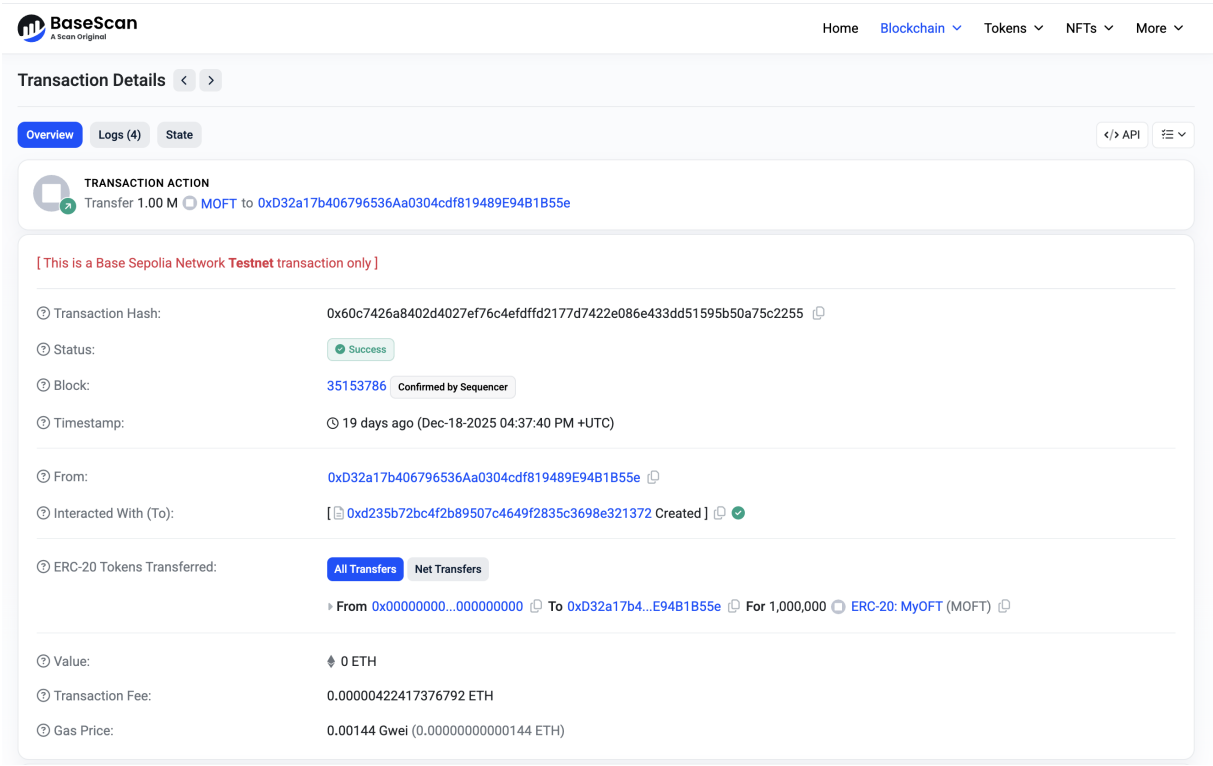

Figura 4.2: Detalhes da Transação de Deploy na rede Base Sepolia.[3].

De forma análoga, a Figura 4.3 apresenta os detalhes da transação de deploy do contrato MyOFT na rede Avalanche Fuji, obtidos a partir do respectivo explorador de blocos. A comparação entre as duas figuras evidencia diferenças nos custos finais de transação, associadas às políticas de precificação de gas e às características específicas de cada rede de teste.

[3]Disponível em:
https://sepolia.basescan.org/tx/0x60c7426a8402d4027ef76c4efdffd2177d7422e086e433dd51595b50a75c2255

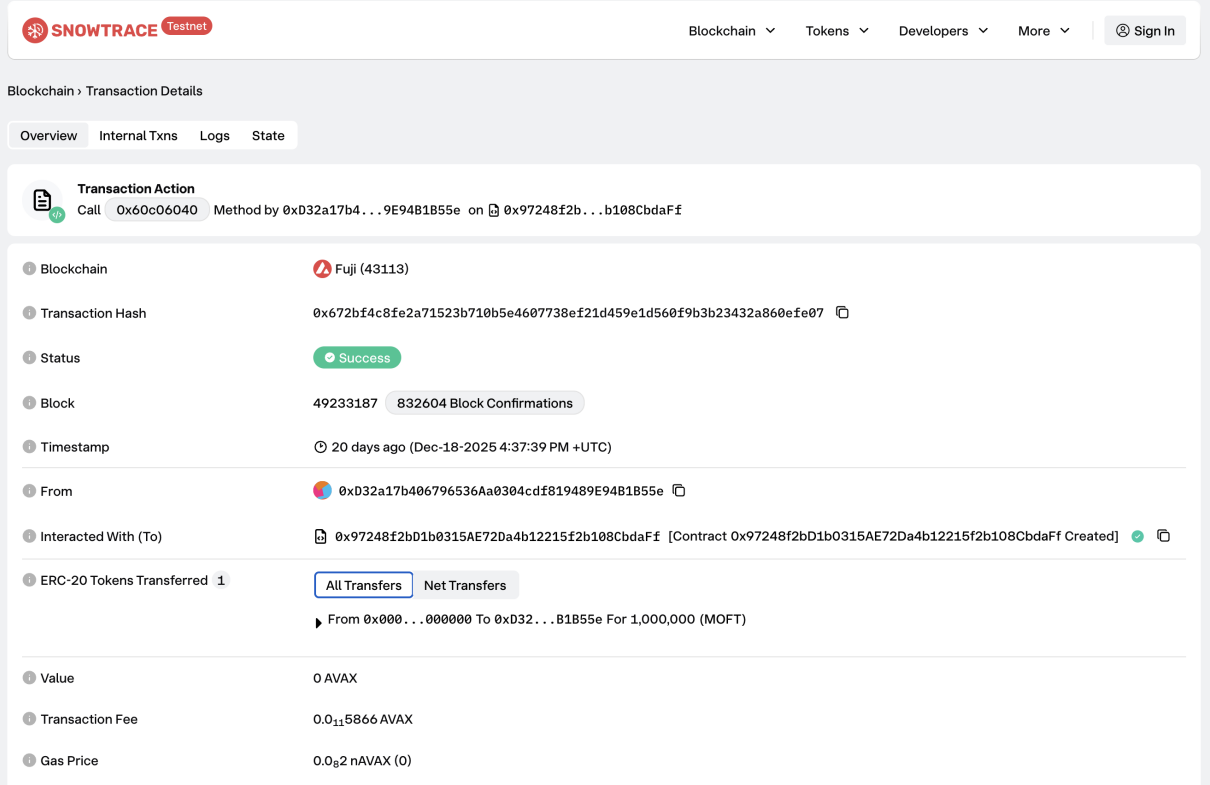

Figura 4.3: Detalhes da Transação de Deploy na rede Avalanche Fuji.[4].

A operação experimental de transferência cross-chain utilizando o padrão OFT permitiu observar, de forma concreta, o fluxo completo de mensageria provido pelo protocolo LayerZero, bem como os custos e responsabilidades associados a cada etapa.

A transferência de **1 MOFT** da rede Base Sepolia para a Avalanche Fuji foi concluída com sucesso em aproximadamente **um minuto**, envolvendo uma transação de origem responsável pela queima do token e envio da mensagem, seguida da execução da função `lzReceive()` na rede de destino para a cunhagem do ativo. Os resultados evidenciam que o usuário arca exclusivamente com os custos na chain de origem, compostos pelo gas da transação `send()` e pela taxa LayerZero, que engloba a atuação do DVN e do executor remoto.

No experimento, o custo total pago na origem foi de aproximadamente **0.000102 ETH**, enquanto o consumo de gas na rede de destino foi integralmente coberto pelo executor, sem impacto direto ao usuário final. Essa separação explícita entre envio, verificação e

[4]Disponível em:
https://testnet.snowtrace.io/tx/0x672bf4c8fe2a71523b710b5e4607738ef21d459e1d560f9b3b23432a860efe07

execução confirma, na prática, o modelo arquitetural proposto pelo LayerZero, ao mesmo tempo em que evidencia a complexidade operacional e a importância da correta parametrização de gas e de componentes de segurança por parte do desenvolvedor em aplicações cross-chain.

A Figura 4.4 apresenta os detalhes completos da transação cross-chain observados no explorador do LayerZero, incluindo os identificadores da transação de origem e de destino, o estado de entrega da mensagem, o tempo total de processamento e a atuação dos componentes de verificação e execução. Esses dados confirmam que a mensagem foi entregue com sucesso (Delivered), validada por uma Rede de Verificadores Descentralizada (DVN) e executada pelo executor configurado.

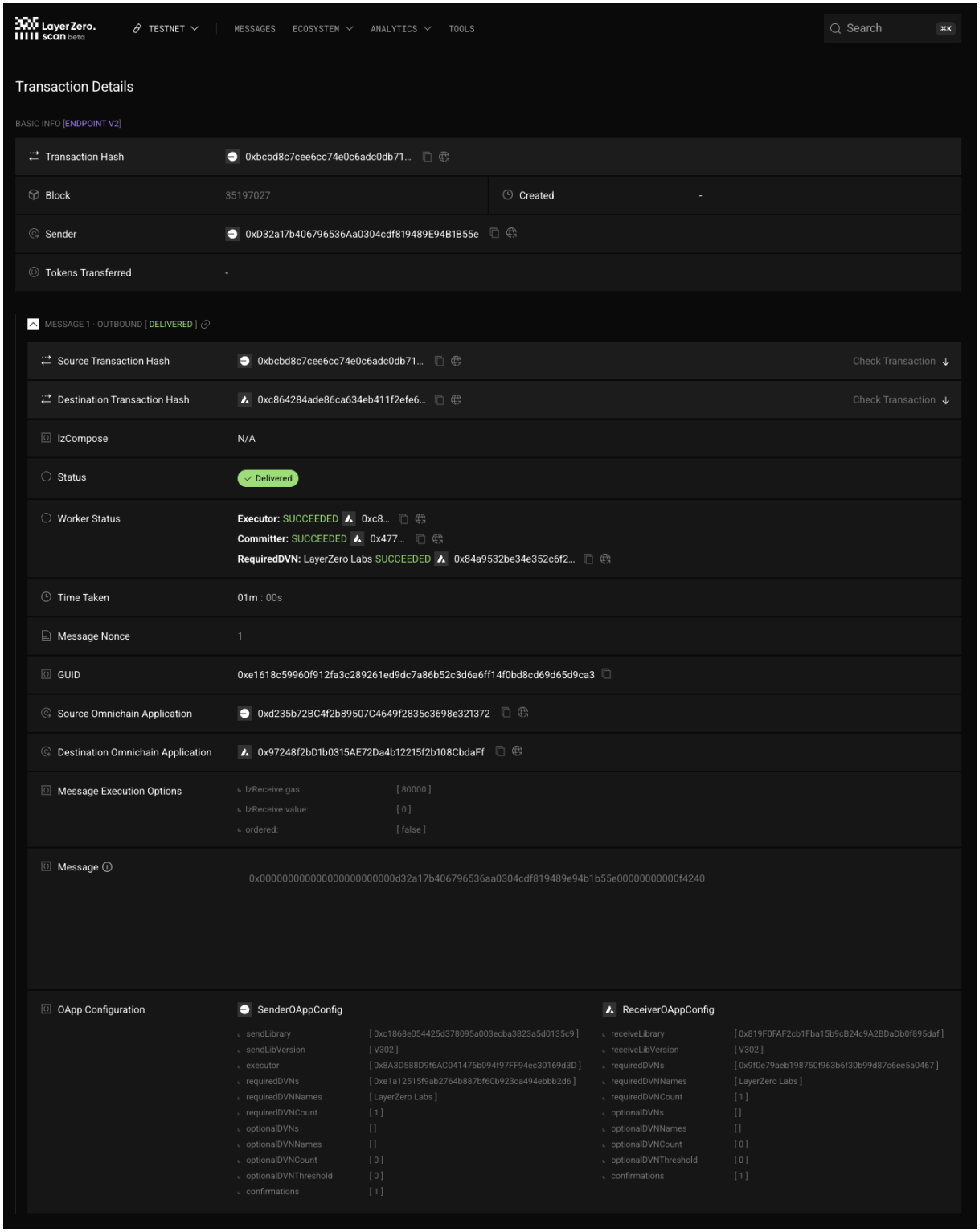

Figura 4.4: Detalhes da Transação Cross-Chain.[5]

A Figura 4.5 detalha a transação correspondente à chamada da função `send()`

[5]Disponível em:
https://testnet.layerzeroscan.com/tx/0xbcbd8c7cee6cc74e0c6adc0db713b490453d1ddef0985efd8598cfda6be1609f

na cadeia de origem (Base Sepolia), evidenciando o consumo de gas associado à queima do token e ao envio da mensagem cross-chain. Observa-se que essa transação concentra os custos diretos arcados pelo usuário, incluindo o gas da execução local e a taxa do protocolo LayerZero.

De forma complementar, a Figura 4.6 apresenta a execução da função `lzReceive()` na cadeia de destino (Avalanche Fuji), responsável pela cunhagem do token. Como ilustrado, o consumo de gas nessa etapa é integralmente coberto pelo executor, não gerando custo adicional ao usuário final, o que reforça a separação entre envio e execução proposta pelo modelo omnichain do LayerZero.

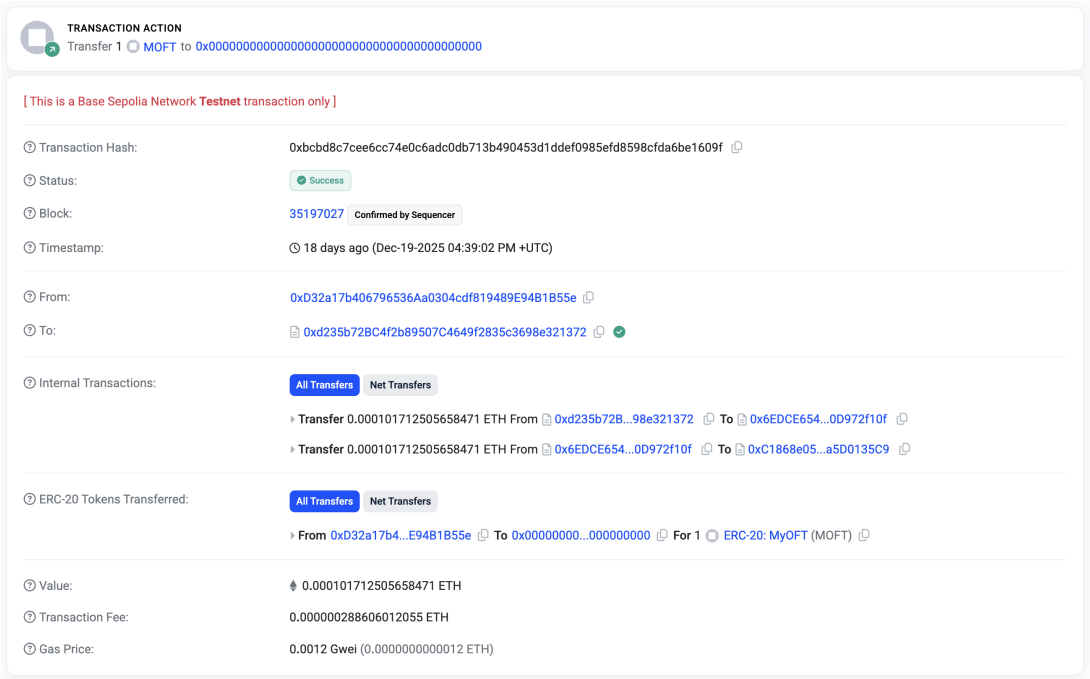

Figura 4.5: Detalhes da Operação send()[6]. - Gas usado: 240.505.

[6]Disponível em: https://sepolia.basescan.org/tx/0xbcbd8c7cee6cc74e0c6adc0db713b490453d1ddef0985efd8598cfda6be1609f

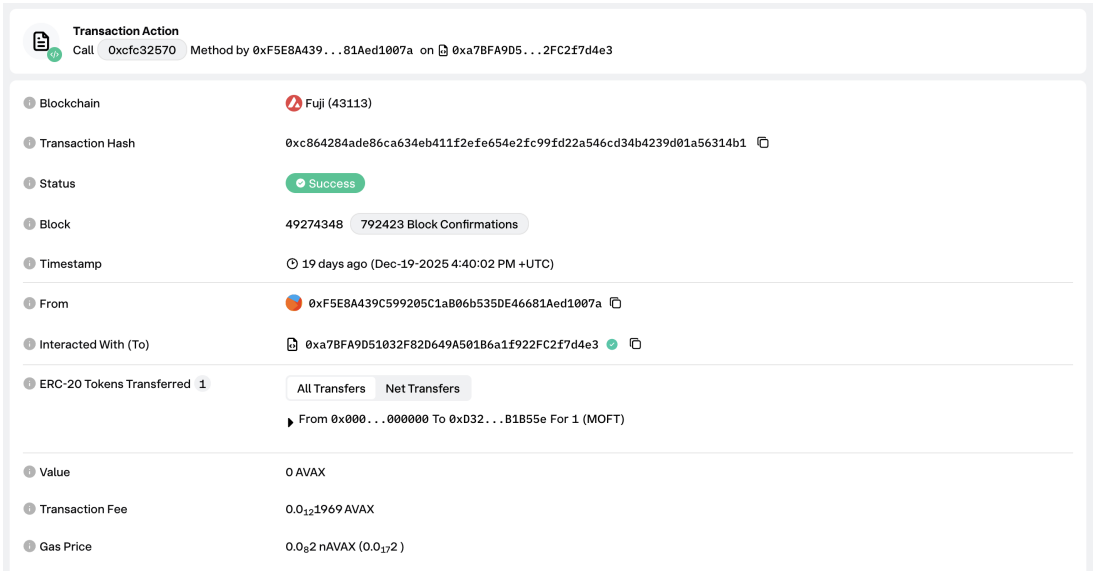

Figura 4.6: Detalhes da Operação lzReceive()[7]. - Gas usado: ∼80.000[8].

A Tabela 4.5 sintetiza os custos observados ao longo da operação, discriminando o gas consumido na chamada `send()`, a taxa LayerZero associada à verificação e entrega da mensagem e o custo de execução da função `lzReceive()` na cadeia de destino. Essa consolidação evidencia, de forma quantitativa, a assimetria de custos entre origem e destino e reforça o papel central da configuração do security stack na previsibilidade econômica de aplicações cross-chain.

Componente	**Valor**
Gas do `send()`	0.000000289 ETH
Taxa LayerZero (DVN + Executor)	0.000101713 ETH
Total pago na origem	∼0.000102 ETH
Gas do `lzReceive()`	∼0.00002 AVAX *(pago pelo Executor)*

Tabela 4.5: Custo Total da Operação Cross-Chain.

4.5.5 Desempenho

O desempenho da comunicação cross-chain foi avaliado a partir da decomposição do fluxo em suas principais fases operacionais, considerando tempos de confirmação, ve-

[7]Disponível em:
https://testnet.snowtrace.io/tx/0xc864284ade86ca634eb411f2efe654e2fc99fd22a546cd34b4239d01a56314b1
[8]Gas alocado conforme configurado no `layerzero.config.ts`; o gas real consumido pode ser menor.

rificação e execução observados durante o experimento. A Tabela 4.6 apresenta uma estimativa dos tempos típicos associados a cada etapa, desde a confirmação da transação na cadeia de origem até a execução da lógica de destino, permitindo analisar o impacto de componentes externos, como a Rede de Verificadores Descentralizada (DVN) e o executor, no tempo total de entrega da mensagem.

Fase	**Tempo Típico**	**Descrição**
Transação Origem	∼2-5s	Confirmação na chain de origem
Confirmações DVN	1 bloco	Configurado em `layerzero.config.ts`
Verificação DVN	∼15-60s	LayerZero Labs DVN verifica a mensagem
Execução Destino	∼2-5s	Executor entrega e executa lzReceive
Total End-to-End	∼30s - 2min	Testnets (pode variar)

Tabela 4.6: Tempo de Execução Cross-Chain

Os tempos observados estão diretamente relacionados às configurações de execução definidas pela aplicação, em especial à alocação de gas para a etapa de execução no destino. No LayerZero, essas configurações são explicitadas no arquivo `layerzero.config.ts`, onde o desenvolvedor define parâmetros obrigatórios para o executor, como o limite de gas reservado para a função `lzReceive()`. O trecho de código a seguir ilustra a configuração utilizada no experimento, na qual foi alocado um limite de `80,000` unidades de gas para a execução da lógica de cunhagem no destino.

```
// layerzero.config.ts - Gas configurado
const EVM_ENFORCED_OPTIONS: OAppEnforcedOption[] = [
    {
        msgType: 1, // SEND
        optionType: ExecutorOptionType.LZ_RECEIVE,
        gas: 80000, // Gas alocado para lzReceive
        value: 0,   // Native value (se necessario)
    },
]
```

Essa configuração evidencia que o desempenho end-to-end em aplicações omnichain depende não apenas das características das redes envolvidas, mas também das decisões explícitas de configuração adotadas pelo desenvolvedor, reforçando o caráter configurável e os trade-offs do modelo proposto pelo LayerZero.

4.5.6 Estrutura do Payload OFT

A compreensão da estrutura interna das mensagens cross-chain é essencial para analisar tanto o custo quanto a flexibilidade do padrão Omnichain Fungible Token (OFT). Em particular, o formato do payload determina quais informações são transmitidas entre as cadeias e como essas informações são interpretadas na execução da lógica de destino. A **??** apresenta a estrutura mínima do payload de uma mensagem OFT, destacando a organização dos campos e seus respectivos tamanhos em bytes.

Bytes 0-31	Recipient Address (bytes32)
Bytes 32-39	Amount (uint64 - shared decimals)
Bytes 40+	Compose Message (opcional)

Tabela 4.7: OFT Message Payload

No OFT, a comunicação cross-chain é realizada por meio de um payload binário estruturado, que encapsula as informações necessárias para a transferência e eventual composição de mensagens adicionais. O payload mínimo de uma mensagem OFT possui 40 bytes, sendo composto por três partes principais: o endereço do destinatário, representado em 32 bytes (bytes32), o valor transferido, codificado como um inteiro de 64 bits (uint64) em unidades de shared decimals, e, opcionalmente, um campo adicional destinado a mensagens compostas (compose message).

Embora o payload OFT em si seja relativamente compacto, o custo total da mensagem cross-chain é significativamente impactado pelo overhead introduzido pelo protocolo LayerZero. Esse overhead, que inclui cabeçalhos, informações de roteamento e dados de verificação, varia tipicamente entre 200 e 500 bytes por mensagem, influenciando diretamente o consumo de gas e o valor da taxa cobrada na chain de origem. Essa característica evidencia que, em aplicações omnichain, o custo não está apenas associado à lógica do

contrato, mas também à infraestrutura de mensageria subjacente.

4.5.7 Cálculo de Taxa Cross-Chain

A taxa paga para a execução de uma operação cross-chain via LayerZero é composta por múltiplos elementos, refletindo a separação arquitetural entre envio, verificação e execução. De forma geral, o custo total inclui: (i) a **taxa do DVN** (Decentralized Verifier Network), responsável pela verificação da mensagem e do estado da chain de origem (∼0.00005-0.0001 ETH em testnets); (ii) a **taxa do executor**, que cobre o gas necessário para a execução da função de recebimento na chain de destino; e (iii) a **taxa do protocolo**, definida no projeto e configurável conforme o modelo adotado.

4.5.8 Tamanho do Bytecode e Conformidade com a EVM

A implementação do contrato `MyOFT` resultou em um bytecode de inicialização com aproximadamente 14,5 KB, enquanto o bytecode efetivamente implantado na rede apresentou cerca de 12,5 KB. Esses valores permanecem confortavelmente abaixo do limite imposto pela EIP-170, que estabelece um tamanho máximo de 24 KB para contratos implantados na Ethereum Virtual Machine. Esse resultado indica que a adoção do padrão OFT, mesmo incorporando dependências do LayerZero e da OpenZeppelin, não impõe restrições práticas relacionadas ao tamanho do contrato, preservando margem para extensões futuras.

4.5.9 Comparativo com Bridges Tradicionais

Quando comparado a bridges tradicionais baseadas nos modelos lock–mint ou lock–release, o padrão OFT apresenta diferenças estruturais relevantes. Em termos de tempo de finalização, as transferências omnichain observadas no experimento foram concluídas em intervalos da ordem de 30 segundos a 2 minutos, enquanto bridges convencionais frequentemente demandam períodos significativamente maiores, variando de alguns minutos a dezenas de minutos.

Outro aspecto central é o modelo de oferta do token. No OFT, a lógica de burn-and-mint preserva uma oferta global única, evitando a fragmentação do supply en-

tre múltiplas redes. Em bridges tradicionais, a emissão de tokens wrapped em chains de destino frequentemente resulta em múltiplas representações do mesmo ativo, com implicações diretas para fungibilidade e risco sistêmico. Além disso, o modelo de segurança do OFT se apoia em redes de verificação descentralizadas, enquanto muitas bridges ainda dependem de multisig ou entidades fortemente centralizadas.

As diferenças observadas entre o padrão Omnichain Fungible Token (OFT) e bridges tradicionais podem ser sintetizadas a partir das métricas analisadas ao longo deste capítulo. A Tabela 4.8 apresenta um comparativo técnico entre a abordagem omnichain baseada no LayerZero e bridges tradicionais do tipo lock-mint, considerando aspectos como mecanismo de transferência, tempo de finalização, modelo de oferta, pressupostos de segurança, custos operacionais e implicações para fungibilidade e liquidez.

Métrica	**OFT (LayerZero)**	**Bridge Lock-Mint Tradicional**
Mecanismo	Burn & Mint nativo	Lock na origem, mint wrapped
Tempo	30s - 2min	5-30min
Supply	Unificado (soma = constante)	Fragmentado por chain
Segurança	DVN descentralizado	Multisig centralizada
Custos	Taxa DVN + Executor	Fee da bridge + gas
Fungibilidade	Token nativo	Token wrapped
Risco de liquidez	Zero	Pool de liquidez necessária

Tabela 4.8: Diferencial Técnico vs Bridges Tradicionais

Observa-se, portanto, que o modelo omnichain elimina a fragmentação de oferta e reduz dependências de liquidez, ao custo de uma maior responsabilidade de configuração no nível da aplicação, refletindo os trade-offs discutidos ao longo da fundamentação teórica.

4.6 Considerações Finais

O projeto demonstra com sucesso a implementação de interoperabilidade blockchain usando o protocolo LayerZero. O token MOFT pode ser transferido entre Base e

Avalanche em aproximadamente 1 minuto, com custo inferior a US$ 0,01 em testnets, mantendo supply unificado e sem fragmentação de liquidez.

A arquitetura é escalável para adicionar novas chains (Arbitrum, Optimism, Polygon, etc.) apenas modificando o arquivo de configuração e executando novo deploy + wiring. O trabalho valida a viabilidade técnica de tokens omnichain como alternativa superior às bridges tradicionais para transferência de ativos entre blockchains.

O código-fonte completo da implementação experimental está disponível em: `https://github.com/ceciliaromao/layer-zero`

5 Análise e Discussão

Este capítulo apresenta uma análise crítica dos resultados obtidos a partir da implementação experimental do padrão Omnichain Fungible Token (OFT) com o protocolo LayerZero, confrontando-os com os pressupostos, promessas e limitações discutidos na literatura sobre interoperabilidade blockchain. O objetivo não é validar ou refutar isoladamente o protocolo, mas compreender de que forma suas escolhas arquiteturais se manifestam na prática e quais implicações emergem para o desenvolvimento de aplicações cross-chain.

5.1 Teoria versus prática na interoperabilidade omnichain

A literatura recente descreve o LayerZero como uma abordagem omnichain que busca reduzir acoplamento entre redes e oferecer uma camada genérica de mensageria cross-chain, separando explicitamente envio, verificação e execução de mensagens (ZARICK; PELLEGRINO; BANISTER, 2021; GAUTHIER et al., 2023).

Na prática, o experimento confirma que essa separação existe do ponto de vista arquitetural, mas evidencia que ela não elimina a complexidade do sistema. Ao contrário, desloca parte dessa complexidade para a configuração e para as decisões tomadas pelo desenvolvedor. A teoria sugere maior flexibilidade e modularidade; a prática mostra que essa flexibilidade exige conhecimento profundo do protocolo e atenção constante às hipóteses de segurança adotadas.

5.2 Benefícios observados do LayerZero

Entre os benefícios efetivamente observados durante a implementação, destaca-se a redução do acoplamento direto entre contratos nas diferentes cadeias. O *LayerZero* atua como uma primitiva de comunicação de baixo nível, utilizando o *EndpointV2* como

uma interface estável e imutável que abstrai detalhes de comunicação e complexidades das redes subjacentes que, em abordagens tradicionais baseadas em bridges, costumam exigir lógica específica para cada par de redes (ZARICK et al., 2024). Esta arquitetura permite que as aplicações (OApps) comuniquem de forma agnóstica à rede, facilitando a integração de novas *blockchains* sem exigir alterações estruturais significativas na lógica da aplicação (ANIDO-RIFÓN, 2025; VONEITZEN et al., 2024).

Outro ponto positivo é a rastreabilidade e transparência operacional. Ao atribuir um Identificador Único Global (GUID) a cada pacote, o protocolo permite que utilizadores e desenvolvedores monitorizem o fluxo completo e o estado da mensagem em tempo real através da ferramenta *LayerZero Scan*. Este nível de visibilidade contrasta com as *bridges* opacas analisadas na literatura, fornecendo um registo imutável e verificável em cadeia que reforça a responsabilidade e a segurança do ecossistema (ANIDO-RIFÓN, 2025).

Além disso, a implementação do OFT confirma que o modelo *omnichain* permite a transferência de ativos sem a necessidade de manter grandes quantidades de liquidez bloqueada em contratos intermediários. Ao contrário das redes de liquidez tradicionais, que sofrem de fragmentação de capital e exigem depósitos colaterais em cada par de redes, o modelo do LayerZero (especialmente via OFT) promove a eficiência de capital ao evitar a dependência de *pools* de terceiros ou tokens intermediários para facilitar as trocas. Esse aspecto dialoga diretamente com críticas recorrentes sobre a colateralização excessiva e o risco de custódia centralizada em infraestruturas de interoperabilidade (SEVIM, 2022; HUANG; YAN; TESSONE, 2024).

5.3 Limitações e tensões identificadas

Apesar dos benefícios, o experimento evidencia limitações importantes. A principal delas refere-se à dependência de componentes externos, cuja correta operação é assumida como hipótese básica de segurança. Embora a literatura discuta o modelo de não colusão como um compromisso razoável (ANIDO-RIFÓN, 2025; VONEITZEN et al., 2024), a prática revela que a segurança de todo o sistema depende estritamente da fiabilidade e integridade destes validadores externos e a escolha inadequada desses componentes pode comprometer todo o sistema. Caso estas entidades sejam controladas pelo mesmo

agente ou ajam de forma maliciosa, podem comprometer a comunicação entre redes ou validar transações não autorizadas. Além disso, o fato de o *LayerZero* deter o controle sobre os contratos de interoperabilidade e a gestão centralizada dos endpoints introduz preocupações de privacidade e pontos únicos de falha.

Outra limitação observada diz respeito aos custos de gas. Embora, em literatura, os *Ultra Light Nodes* sejam destacados pela sua eficiência ao transmitirem cabeçalhos sob demanda, evitando a sincronização total de blocos, eles não eliminam o peso financeiro das operações *cross-chain* (ANIDO-RIFÓN, 2025; ZHENG; LEE; QIAN, 2023). em comparação com abordagens que exigem verificação completa de cabeçalhos (LU; JAJOO; NAMJOSHI, 2024a). O experimento demonstra que os custos agregados (pagamento a múltiplos verificadores e executores) permanecem significativamente superiores aos de uma transação local, especialmente quando são exigidas múltiplas confirmações para mitigar riscos de finalidade. Provas mais complexas, como as baseadas em conhecimento zero ou caminhos com múltiplos saltos (*multi-hop*), aumentam o tamanho dos pacotes e, consequentemente, o consumo de recursos na rede de destino

Adicionalmente, erros recorrentes durante a implementação — como falhas na estimativa de custos (estimateFees) ou na configuração de caminhos — reforçam a perceção de que o LayerZero não elimina a complexidade, mas redistribui-a ao longo da pilha de desenvolvimento. O desenvolvedor deixa de se preocupar com a infraestrutura da rede, mas assume a propriedade exclusiva da segurança e do custo do protocolo, tendo de gerir manualmente o seu "*Security Stack*"e coordenar componentes off-chain independentes. Esta flexibilidade exige uma compreensão profunda das nuances técnicas de cada rede conectada, sob pena de configurações inadequadas resultarem em falhas de execução ou vulnerabilidades sistémicas.

5.4 Implicações para desenvolvedores de aplicações *cross-chain*

Do ponto de vista do desenvolvedor, o *LayerZero* impõe uma mudança conceitual relevante: a segurança deixa de ser uma propriedade implícita do protocolo e passa a

ser uma escolha explícita de configuração. Ao contrário dos modelos de segurança monolíticos, as fontes indicam que o LayerZero utiliza um modelo modular onde as aplicações omnichain (OApps) detêm a propriedade exclusiva da sua "Security Stack"(Pilha de Segurança), permitindo-lhes selecionar quais as Redes de Verificadores Descentralizadas (DVNs) e bibliotecas de verificação pretendem utilizar. Este nível de controle exige que o desenvolvedor assuma a responsabilidade pela integridade da aplicação, uma vez que a segurança do sistema passa a depender da fiabilidade dos validadores externos escolhidos e da garantia de não-colusão entre os agentes de verificação (ZARICK et al., 2024; ZHANG et al., 2024; VONEITZEN et al., 2024; KATE et al., 2025).

Esta arquitetura aproxima o desenvolvimento de aplicações omnichain de um processo de engenharia de sistemas distribuídos, no qual decisões de arquitetura têm impacto direto sobre segurança, custo e desempenho da aplicação, exigindo-se, assim, uma compreensão profunda de conceitos como:

- Tolerância a Falhas e Liveness: O desenvolvedor deve planear a recuperação da "liveness"do canal através da reconfiguração da Security Stack caso os verificadores falhem ou fiquem offline (ZARICK et al., 2024).

- Modelos de Confiança Parcial: É necessário avaliar o risco de dependência de terceiros, compreendendo que a abstração do protocolo não elimina riscos, mas desloca-os para a infraestrutura off-chain (VONEITZEN et al., 2024; KATE et al., 2025).

- Gestão de Assincronia: O desenvolvimento exige lidar com a natureza não-atómica das transações cross-chain, onde a consistência do estado deve ser gerida através de mecanismos como o lzCompose para garantir a execução lógica em múltiplas redes (ÖZ et al., 2025).

As decisões de arquitetura tomadas pelo desenvolvedor têm um impacto direto no desempenho e no custo da aplicação. Por exemplo, a escolha de métodos de prova mais robustos (como ZKPs) ou o aumento do número de verificadores para mitigar riscos de segurança pode introduzir latência adicional e custos de gas significativamente superiores às transações locais (GAUTHIER et al., 2023; ZARICK; PELLEGRINO; BANISTER, 2023b; LU; JAJOO; NAMJOSHI, 2024b). Assim, o desenvolvedor deixa de ser apenas

um programador de contratos inteligentes para se tornar um gestor de infraestrutura, encarregado de equilibrar os trade-offs do Trilema da Interoperabilidade: confiança, extensibilidade e generalização (BELCHIOR et al., 2024; ZHAO et al., 2023).

A literatura reforça que esta tendência é inevitável em ambientes multi-chain, onde a interoperabilidade é vista como um pré-requisito essencial para a escalabilidade dos serviços (DENG et al., 2025; ZHENG; LEE; QIAN, 2023), e o experimento reforça essa conclusão. O LayerZero, ao atuar como uma primitiva de comunicação de baixo nível, fornece as ferramentas para esta integração, mas transfere a complexidade da coordenação e da segurança para a camada aplicacional (ZARICK et al., 2024; ANIDO-RIFÓN, 2025; ARULKUMARAN et al., 2024).

5.5 Impacto arquitetural em aplicações cross-chain

A análise conjunta da teoria e da prática sugere que o *LayerZero* representa uma evolução arquitetural relevante em relação às *bridges* tradicionais, mas não uma solução definitiva para o problema da interoperabilidade. O seu principal mérito está em atuar como uma primitiva de comunicação de baixo nível que separa rigorosamente a camada de verificação da camada de execução, permitindo que as aplicações (OApps) detenham a propriedade exclusiva das suas configurações de segurança e custo, explicitando *trade-offs* que, em outras abordagens, permanecem implícitos ou ocultos. Ao contrário dos modelos monolíticos, o *LayerZero* isola aspetos intrínsecos (como a resistência à censura e a proteção contra replay) em Endpoints imutáveis, enquanto delega aspetos extrínsecos (como algoritmos de assinatura e redes de verificadores) a módulos configuráveis (ZARICK et al., 2024).

Do ponto de vista arquitetural, as aplicações que adotam o *LayerZero* passam a incorporar a interoperabilidade como parte central do seu desenho, através de semânticas de rede universais, e não como um componente acessório. A introdução de primitivas como o lzCompose permite uma composição semântica uniforme entre blockchains heterogéneas, isolando falhas de contratos compostos e garantindo que a interoperabilidade não seja apenas um acessório, mas uma funcionalidade integrada na lógica de execução. Isto implica maior responsabilidade para o desenvolvedor, que deve agora gerir o seu próprio

Security Stack, mas também confere maior controle sobre o comportamento do sistema (ZARICK et al., 2024).

O modelo de confiança do *LayerZero*, baseado na independência e não-colusão entre Oráculos (ou DVNs) e Relatores, redefine a segurança como um espetro configurável (ZARICK; PELLEGRINO; BANISTER, 2023b). No entanto, esta flexibilidade não elimina os riscos; a segurança da OApp permanece dependente da honestidade dos validadores externos selecionados. Se as entidades configuradas coludirem, a integridade da comunicação é comprometida, podendo resultar em transações fraudulentas ou ataques de MEV cross-chain (VONEITZEN et al., 2024; KATE et al., 2025).

Em síntese, o *LayerZero* não elimina riscos nem garante interoperabilidade automática. Ele oferece um conjunto de mecanismos que, quando bem compreendidos e configurados, podem resultar em aplicações cross-chain mais flexíveis e transparentes. Eles operam numa fronteira de Pareto entre custo e confiança, exigindo que as decisões técnicas sejam tomadas de forma informada e consciente (ZARICK et al., 2024). Como demonstrado em aplicações como o *Stargate*, o sucesso desta arquitetura depende do equilíbrio entre a eficiência de capital e a resiliência dos componentes *off-chain* escolhidos (HUANG; YAN; TESSONE, 2024; ZHENG; LEE; QIAN, 2023).

6 Conclusões

Este trabalho investigou o problema da interoperabilidade entre blockchains a partir de uma perspectiva arquitetural e analítica, com foco no protocolo omnichain LayerZero. Partindo da constatação de que o ecossistema blockchain evoluiu para um cenário altamente fragmentado, no qual múltiplas redes operam de forma isolada, buscou-se compreender como soluções recentes de interoperabilidade lidam com desafios clássicos de sistemas distribuídos, como confiança, segurança e comunicação entre domínios heterogêneos.

O objetivo geral do trabalho foi analisar o modelo omnichain proposto pelo LayerZero, confrontando suas premissas teóricas com uma implementação prática baseada no padrão Omnichain Fungible Token (OFT). Para isso, foram definidos como objetivos específicos: (i) revisar criticamente a literatura sobre interoperabilidade blockchain e comunicação cross-chain, (ii) descrever e analisar a arquitetura e o modelo de comunicação do LayerZero, e (iii) observar, por meio de um experimento prático, como essas escolhas arquiteturais se manifestam no desenvolvimento de aplicações reais.

Como principal contribuição, o trabalho oferece uma análise integrada entre teoria e prática, algo ainda pouco explorado na literatura sobre interoperabilidade blockchain. A partir da revisão bibliográfica, foi possível identificar padrões recorrentes de soluções e vulnerabilidades em bridges tradicionais, bem como limitações inerentes a modelos que concentram confiança ou liquidez. Em contraste, o estudo do LayerZero evidenciou uma proposta arquitetural que busca reduzir acoplamento entre redes e tornar explícitos os trade-offs de segurança, deslocando parte das decisões para o desenvolvedor da aplicação.

A implementação experimental com OFT permitiu observar que, embora o LayerZero ofereça maior flexibilidade e modularidade em comparação a abordagens tradicionais, essa flexibilidade vem acompanhada de um aumento na complexidade operacional e na responsabilidade do desenvolvedor. Aspectos como a configuração de componentes externos, o gerenciamento de custos de gas e a compreensão das hipóteses de não colusão mostraram-se centrais para o funcionamento correto e seguro da aplicação. Dessa forma,

os resultados indicam que o protocolo não elimina riscos, mas os redistribui ao longo da arquitetura do sistema.

Entre as limitações deste trabalho, destaca-se o escopo predominantemente qualitativo da análise. Embora a experimentação prática tenha fornecido evidências relevantes sobre o funcionamento do LayerZero, não foram conduzidas avaliações quantitativas sistemáticas, como medições comparativas de latência, custo ou taxa de falhas em relação a outros protocolos de interoperabilidade. Além disso, a análise de segurança foi baseada principalmente em pressupostos arquiteturais e na literatura existente, não incluindo auditorias formais ou testes adversariais.

Como trabalhos futuros, destacam-se diversas possibilidades de aprofundamento. Avaliações quantitativas comparativas entre o LayerZero e outros protocolos, como IBC, Axelar, Wormhole ou soluções baseadas em CCIP, poderiam fornecer métricas mais objetivas sobre desempenho e custo. Outra linha promissora envolve a análise de segurança em maior profundidade, incluindo simulações de falhas, cenários de colusão e avaliação de impacto econômico. Por fim, estudos focados na experiência do desenvolvedor e na usabilidade de soluções omnichain podem contribuir para compreender os desafios práticos de adoção dessas tecnologias em larga escala.

Em síntese, este trabalho contribui para o entendimento crítico da interoperabilidade blockchain ao analisar o LayerZero não apenas como uma proposta teórica, mas como um sistema em operação, inserido em um contexto real de desenvolvimento. Ao articular conceitos de redes de computadores, sistemas distribuídos e contratos inteligentes, espera-se que os resultados apresentados sirvam como base para futuras pesquisas e para decisões mais informadas na construção de aplicações cross-chain.

Bibliografia

ANIDO-RIFÓN, L. Proposal of standardization of a blockchain-based interoperability platform for academic credentials. *blockstand*, 2025.

ARULKUMARAN, R. et al. Cross-chain nft marketplaces with layer zero and chain link. *Modern Dynamics: Mathematical Progressions*, 2024. Disponível em: ⟨https://api.semanticscholar.org/CorpusID:272780941⟩.

Banaeian Far, S.; Hosseini Bamakan, S. M. Third layer blockchains are being rapidly developed: Addressing state-of-the-art paradigms and future horizons. *Journal of Network and Computer Applications*, v. 233, p. 104044, 2025.

BELCHIOR, R. et al. A brief history of blockchain interoperability. *Commun. ACM*, Association for Computing Machinery, v. 67, n. 10, p. 62–69, 2024.

BELCHIOR, R. et al. A survey on blockchain interoperability: Past, present, and future trends. *ACM Comput. Surv.*, Association for Computing Machinery, v. 54, n. 8, 2021.

CHEN, B. et al. *A Comprehensive Survey of Blockchain Scalability: Shaping Inner-Chain and Inter-Chain Perspectives*. 2024. Disponível em: ⟨https://arxiv.org/abs/2409.02968⟩.

DAVIDSON, S. *Secure and Efficient Message Routing and Delivery Across Blockchain Networks*. [S.l.]: University of Waterloo, 2024.

DE, A.; AGRAWAL, D.; ABBADI, A. E. Crocrpc: Cross-chain remote procedure calls framework for dapps. *Proceedings of the VLDB Endowment. ISSN*, v. 2150, 2024.

DENG, Z. et al. *Enhancing Blockchain Cross Chain Interoperability: A Comprehensive Survey*. 2025. Disponível em: ⟨https://arxiv.org/abs/2505.04934⟩.

DOTY, M. et al. Golden gate. *GGX Interchain Infrastructure Protocol*, 2023.

EPURE, L. I. Baron chain project - the quantum-safe ai-driven blockchain. *Authorea Preprints*, Wiley, 2024.

GAJERA, H.; REDDY, A.; REDDY, B. *Omnichain Web: The Universal Framework for Streamlined Chain Abstraction and Cross-Layer Interaction*. 2025. Disponível em: ⟨https://arxiv.org/abs/2411.10132⟩.

GAUTHIER, T. et al. *Topos: A Secure, Trustless, and Decentralized Interoperability Protocol*. 2023. Disponível em: ⟨https://arxiv.org/abs/2206.03481⟩.

HAN, Y. et al. A study of blockchain-based liquidity cross-chain model. *Plos one*, Public Library of Science San Francisco, CA USA, v. 19, n. 6, p. e0302145, 2024.

HUANG, C.; YAN, T.; TESSONE, C. J. Seamlessly transferring assets through layer-0 bridges: An empirical analysis of stargate bridge's architecture and dynamics. In: *Companion Proceedings of the ACM Web Conference 2024*. New York, NY, USA: Association for Computing Machinery, 2024. (WWW '24), p. 1776–1784. ISBN 9798400701726. Disponível em: ⟨https://doi.org/10.1145/3589335.3651964⟩.

ILISEI, D. *Analyzing the Role of Bridges in Cross-Chain MEV Extraction*. Tese (Doutorado) — Master's Thesis. Technical University of Munich, 2024.

KATE, A. et al. *HyperLoop: Rationally secure efficient cross-chain bridge*. 2025. Cryptology ePrint Archive, Paper 2025/176. Disponível em: ⟨https://eprint.iacr.org/2025/176⟩.

KHAN, K. M. et al. Empirical investigation on blockchain interoperability. *VFAST Transactions on Software Engineering*, v. 11, n. 1, p. 25–36, 2023.

LAWRENCE, A. *Bridging Blockchains: The Rise of Cross-Chain Protocols and Interoperable Ecosystems*. [S.l.], 2025.

LAWRENCE, A. Unlocking seamless blockchain communication: A deep dive into interoperability solutions. *OSF Preprints*, OSF, 2025.

LESAVRE, L.; VARIN, P.; YAGA, D. *Blockchain networks: Token design and management overview*. [S.l.], 2020.

LU, H.; JAJOO, A.; NAMJOSHI, K. S. *Atomicity and Abstraction for Cross-Blockchain Interactions*. 2024. Disponível em: ⟨https://arxiv.org/abs/2403.07248⟩.

LU, H.; JAJOO, A.; NAMJOSHI, K. S. A two-phase protocol for atomic multi-chain transactions. In: *Proceedings of the ACM Conext-2024 Workshop on the Decentralization of the Internet*. [S.l.]: Association for Computing Machinery, 2024. p. 21–27.

SCHULTE, S. et al. Towards blockchain interoperability. In: *International Conference on Business Process Management*. [s.n.], 2019. Disponível em: ⟨https://api.semanticscholar.org/CorpusID:197630889⟩.

SEVIM, H. O. A survey on trustless cross-chain interoperability solutions in on-chain finance. *Distributed Ledger Technologies Workshop*, 2022.

VONEITZEN, C. D. et al. Bridging the gap: Achieving seamless interoperability between ethereum-based blockchains using inter-blockchain communication protocols. *Authorea Preprints*, Authorea, 2024.

YAN, T.; HUANG, C.; TESSONE, C. J. *Tracing Cross-chain Transactions between EVM-based Blockchains: An Analysis of Ethereum-Polygon Bridges*. 2025. Disponível em: ⟨https://arxiv.org/abs/2504.15449⟩.

YEDDOU, A. F.; BELOUCHRANI, A.; MOKRANE, A. Enhancing peer-to-peer energy trading in smart grids through blockchain interoperability using layer zero. In: *2024 6th International Conference on Blockchain Computing and Applications (BCCA)*. [S.l.: s.n.], 2024. p. 624–629.

YOUNG, V. *Methods and systems for implementing an omni-chain interoperability protocol in an omni-chain network*. [S.l.]: Google Patents, 2024. US Patent 12,131,320.

ZARICK, R.; PELLEGRINO, B.; BANISTER, C. *LayerZero: Trustless Omnichain Interoperability Protocol*. 2021. Disponível em: ⟨https://arxiv.org/abs/2110.13871⟩.

ZARICK, R.; PELLEGRINO, B.; BANISTER, C. *Trustless omnichain communication protocol platforms*. [S.l.]: Google Patents, 2023. US Patent App. 18/120,485.

ZARICK, R.; PELLEGRINO, B.; BANISTER, C. *Trustless omnichain communication protocol platforms implementing resource balancing*. [S.l.]: Google Patents, 2023. US Patent App. 18/120,475.

ZARICK, R. et al. *LayerZero*. 2024. Disponível em: ⟨https://arxiv.org/abs/2312.09118⟩.

ZARICK, R.; ZHANG, I. *Using self-regulating functions to implement blockchain-based token attribution with reduced computational complexity*. [S.l.]: Google Patents, 2025. US Patent App. 18/483,568.

ZHANG, M. et al. Cross-chain bridges: Attack taxonomy, defenses, and open problems. In: *Proceedings of the 27th International Symposium on Research in Attacks, Intrusions and Defenses*. New York, NY, USA: Association for Computing Machinery, 2024. (RAID '24), p. 298–316. ISBN 9798400709593. Disponível em: ⟨https://doi.org/10.1145/3678890.3678894⟩.

ZHAO, Q. et al. A comprehensive overview of security vulnerability penetration methods in blockchain cross-chain bridges. *Authorea Preprints*, Authorea, 2023.

ZHENG, J.; LEE, D. K. C.; QIAN, D. An in-depth guide to cross-chain protocols under a multi-chain world. *World Scientific Annual Review of Fintech*, v. 01, p. 2350003, 2023.

ÖZ, B. et al. *Pandora's Box: Cross-Chain Arbitrages in the Realm of Blockchain Interoperability*. 2025. Disponível em: ⟨https://arxiv.org/abs/2501.17335⟩.