Pagamentos Automatizados Programáveis em Carteiras Auto-Custodiadas: Uma Exploração Técnica

Ana Julia Bittencourt Fogaça[1], **Saulo Popov Zambiasi**[2]

[1]Universidade do Sul de Santa Catarina (UNISUL)
Tubarão - SC - Brasil

[2]Universidade do Sul de Santa Catarina (UNISUL)
Florianópolis - SC - Brasil

anajuliabit@gmail.com, saulopz@gmail.com

Abstract.

1. Introdução

A tecnologia blockchain, primeiramente introduzida por Satoshi Nakamoto em 2008, é identificada como uma megatendência computacional capaz de revolucionar múltiplos setores industriais[World Economic Forum 2015]. As características distintas de segurança, transparência e rastreabilidade inerentes à blockchain têm incentivado uma ampla gama de setores a explorar seu uso na reestruturação de suas operações fundamentais. A aplicabilidade dessa tecnologia ultrapassa o domínio das criptomoedas, abarcando setores como pagamentos, gerenciamento de identidade, saúde, eleições governamentais e outros[Pu and Lam 2021].

A publicação do whitepaper do Ethereum em 2014 simbolizou um avanço considerável na evolução da tecnologia blockchain[Buterin 2014]. Diferentemente do Bitcoin, concebido originalmente como uma moeda digital, o Ethereum inaugurou uma funcionalidade disruptiva no campo da tecnologia blockchain: os contratos inteligentes. A inovação trazida pelo Ethereum reside na incorporação de uma máquina virtual capaz de processar códigos em linguagens de programação *Turing complete* na blockchain, habilitando assim a construção de aplicativos descentralizados. Estes aplicativos propõem a substituição dos sistemas de back-end por contratos inteligentes que operam em uma blockchain[Dannen 2017]. Entretanto, apesar do seu imenso potencial, a complexidade intrínseca à aplicação prática dessa tecnologia representa um dos entraves para sua adoção em grande escala[Sadhya and Sadhya 2018].

O Ethereum, ao contrário do Bitcoin que emprega um esquema de UTXO [Zahnentferner 2018], adotou um sistema de contas, cujos detalhes serão explorados na seção subsequente. As *Externally Owned Accounts* (EOAs), o tipo de conta amplamente utilizado pelas carteiras auto-custodiadas, como a MetaMask[21 2023], apresenta limitações significativas que tornam aplicações como pagamentos automáticos impraticáveis na blockchain. Além disso, a experiência oferecida ao usuário final fica aquém quando comparada à interatividade e usabilidade de aplicativos da internet atuais. A situação atual é análoga à necessidade de inserir sua senha (chave privada) para cada ação que não seja puramente consumir dados, ou, em uma linguagem mais técnica, para cada operação de leitura.

Em face ao desafio emergente, são regularmente introduzidas propostas inovadoras com a finalidade de aprimorar a tecnologia blockchain e, por conseguinte, fomentar a adoção de DApps. Uma dessas proposições recentes é a *Account Abstraction* (AA)[Buterin et al. 2021]. A AA representa uma inovação notável, conferindo maior flexibilidade ao funcionamento das contas no Ethereum e viabilizando novos casos de uso. Pagamentos automáticos são um exemplo e foram recentemente objeto de investigação da Visa, uma empresa pioneira em soluções de pagamento. A Visa propôs a possibilidade de efetuar pagamentos automáticos na blockchain sem a necessidade de terceirizar a custódia, explorando as vantagens proporcionadas pela AA[Beams et al. 2023]. Entretanto, os detalhes técnicos ou o código-fonte para tal implementação não foram publicizados.

Esta pesquisa se dispõe a explorar em profundidade essa funcionalidade e propor uma implementação prática que viabilize pagamentos automáticos em carteiras autocustodiadas. O objetivo é prover uma perspectiva elucidativa e diretrizes pragmáticas que possam atuar como ponto de referência para futuras aplicações de pagamentos automatizados em blockchains que funcionam no esquema *account-based* [Zahnentferner 2018], como o Ethereum. A estrutura deste artigo se organiza da seguinte maneira: após esta introdução, prosseguimos com uma revisão bibliográfica abrangente, iniciando com uma explanação detalhada do funcionamento do Ethereum, delineando as responsabilidades da Ethereum Virtual Machine (EVM), esclarecendo os diferentes tipos de contas e elucidando as propostas da AA.Na seção de desenvolvimento, realizamos uma análise do estudo conduzido pela Visa e sugerimos uma implementação em Solidity com o objetivo de habilitar pagamentos automáticos programáveis em carteiras auto-custodiadas. Nossa discussão culmina com uma reflexão sobre as implicações e possíveis aplicações desta nova funcionalidade nas blockchains.

2. Revisão Bibliográfica

Como destacado anteriormente, a peculiaridade do Ethereum reside em sua capacidade de processar códigos, denominados de contratos inteligentes, em uma máquina virtual conhecida como *Ethereum Virtual Machine* (EVM) [Hildenbrandt et al. 2018]. A EVM, uma máquina virtual global que opera em um formato de instância única, é executada repetidamente em uma multiplicidade de computadores ao redor do mundo. Cada nó no Ethereum mantém uma cópia local da EVM, responsável por validar e executar contratos inteligentes. Qualquer alteração de estado resultante da execução desses contratos inteligentes é registrada no Ethereum[Antonopoulos and Wood 2018]. O Ethereum visa possibilitar a execução de aplicações e scripts arbitrários que operam em transações, utilizando uma blockchain para sincronizar o estado global de maneira totalmente verificável por qualquer participante do sistema[Hildenbrandt et al. 2018].

O estado, que é único e compartilhado entre todos os nós, confere uma característica essencial aos contratos inteligentes: a necessidade de serem determinísticos. A linguagem mais utilizada para a criação de contratos inteligentes é o Solidity[Solidity contributors]. O Solidity é uma linguagem de programação de alto nível e necessita ser compilada para o código EVM - bytecode que a EVM pode executar nativamente[Wood et al. 2014]. Devido a essas características, o Ethereum é frequentemente caracterizado como um 'computador mundial de propósito geral'[Antonopoulos and Wood 2018].

Como uma plataforma de código aberto, o Ethereum adota um procedimento conhecido como *Ethereum Improvement Proposal* (EIP) [Becze et al. 2015] para introduzir melhorias. Qualquer membro da comunidade pode propor um EIP, que subsequentemente será avaliado e discutido pelos demais membros. Quando um EIP recebe a sanção da governança do Ethereum, tem potencial para se transformar em um *Ethereum Request for Comments* (ERC), uma categoria de EIP que estabelece padrões no nível dos aplicativos e é incorporado nos contratos inteligentes, ao invés de ser implementado no protocolo Ethereum em si. Nos parágrafos subsequentes, abordaremos a AA, para entender como ela viabiliza o cenário de uso que estamos investigando: pagamentos automáticos.

Para uma compreensão completa da AA, é crucial entender as diferenças entre os dois tipos de contas no Ethereum. No Ethereum, existem duas categorias distintas de contas: as contas de propriedade externa (*externally owned accounts* - EOAs), que são controladas por chaves privadas, e as contas de contrato (*contract accounts* - CAs), que são controladas pela lógica incorporada no código dos contratos inteligentes a elas associados[Buterin 2014].

As EOAs contêm os seguintes campos: *nonce*, que representa o número de transações enviadas pela conta, e *balance*, que indica o saldo da conta em Wei[Antonopoulos and Wood 2018]. Por outro lado, as CAs possuem dois campos adicionais: *storageRoot*, que contém o hash da raiz de uma estrutura de dados em árvore de Merkle[Becker 2008], usada para armazenar dados associados à conta, e *codeHash*, que armazena o hash do contrato inteligente compilado no código EVM, que é ativado sempre que uma transação é enviada para a CA ou quando a CA é acionada por outro contrato inteligente[Wood et al. 2014].

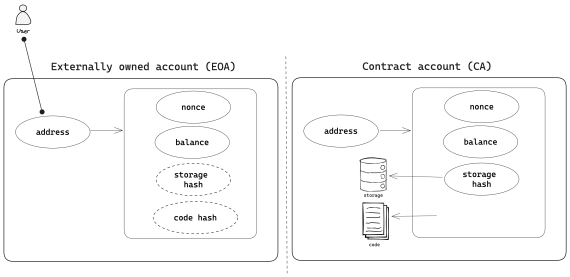

Figura 1. Tipos de conta no Ethereum

Apenas as EOAs têm a capacidade de iniciar transações. Uma transação contém os seguintes campos: *nonce*, que também denota a quantidade de transações enviadas por essa conta, *from* - a EOA que assina a transação, *recipient* - o endereço da conta de destino (se for uma EOA, transferirá ETH, se for uma CA, acionará o contrato inteligente da CA), *value* - a quantidade de ETH a ser transferida para o destinatário em WEI, *signature* - a assinatura gerada com a chave privada da EOA que está transmitindo a transação[Wood et al. 2014]. Existem ainda campos adicionais relacionados ao gás[Antonopoulos and Wood 2018], que não serão discutidos neste artigo.

As EOAs são controladas exclusivamente por chaves privadas que utilizam o esquema de assinatura ECDSA[National Institute of Standards and Technology 2013]. Essa restrição limita sua utilidade, pois para usá-las de forma segura, os usuários precisam de conhecimento técnico para armazenar e utilizar chaves privadas. Além disso,

a usabilidade é restrita pela necessidade de assinar cada transação em tempo real, o que torna impraticável a realização de pagamentos automáticos pré-autorizados. Outra consideração importante é que o algoritmo ECDSA pode se tornar obsoleto com o avanço da computação quântica[Oder et al. 2014].

A AA é uma inovação proposta para o Ethereum com a finalidade de consolidar os dois tipos de contas existentes: External Owned Accounts (EOAs) e Contract Accounts (CAs). O intuito primordial da AA é oferecer aos usuários a alternativa de utilizar CAs, ao invés de EOAs, como suas contas primárias, desvinculando, desta maneira, a ligação intrínseca das contas auto-custodiadas com o esquema de assinatura ECDSA.

Essa transformação acarreta um aumento significativo na flexibilidade, possibilitando a criação de contas com autenticação multi-assinaturas, autenticação de dois fatores, fixação de limites para saques e habilitação de pagamentos automáticos[Ethereum 2023a, Beams et al. 2023, Coyle 2023]. O impacto da inovação fornecida pela AA é análogo à revolução que os Cartões Virtuais Revolut[Revolut 2023] geraram no âmbito dos cartões de crédito.

Foram propostas diversas abordagens para a implementação da AA no Ethereum[Ethereum 2023b, Ethereum 2023a]. A proposta EIP-4337, no entanto, se destacou por prescindir de qualquer modificação na camada de protocolo do Ethereum, razão pela qual nos referimos a ela como ERC daqui em diante. A ERC-4337 introduz uma estrutura que descreve uma operação de usuário, designada como *UserOperation*[Buterin et al. 2021]. É importante salientar que, para evitar confusões terminológicas, este objeto não recebe a denominação de transação, embora contenha, assim como uma transação, os campos previamente mencionados, com algumas variações e acréscimos. Detalharemos essas questões na seção de implementação técnica.

Os usuários submetem mensagens *off-chain* que contêm o objeto *UserOperation* para um *mempool* exclusivamente reservado para mensagens que incluem objetos *UserOperation*. Participantes conhecidos como *bundlers* são incumbidos de ler as mensagens recebidas pelo *mempool*, coletá-las e agrupá-las em uma única transação. Subsequentemente, realizam uma chamada para um método chamado *handleOps* em um contrato especial denominado *EntryPoint*. Este contrato é responsável por executar essas transações agrupadas e, consequentemente, inseri-las em um bloco na rede[Buterin et al. 2021].

A chamada para o método *handleOps* executará chamadas para os contratos inteligentes associados às contas relacionadas aos objetos *UserOperation*. O contrato inteligente de cada conta é responsável por implementar a lógica de execução da transação e validar sua legitimidade. Devido a EVM ser *Turing-complete*, é possível criar lógicas de execução de transações arbitrárias. Na próxima seção, proporemos uma implementação de contrato inteligente que pode ser adotada por contas compatíveis com AA para habilitar pagamentos automatizados.

Referências

(2023). Metamask.

Antonopoulos, A. and Wood, G. (2018). *Mastering Ethereum: Building Smart Contracts and DApps*. O'Reilly Media.

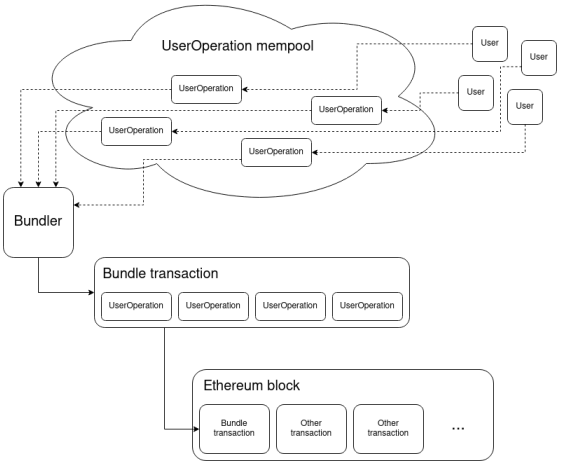

Figura 2. UserOperation meempool

Beams, A., Gu, C., Raghuraman, S., Minaei, M., and Kumaresan, R. (2023). Auto payments for self-custodial wallets.

Becker, G. (2008). Merkle signature schemes, merkle trees and their cryptanalysis. *Ruhr-University Bochum, Tech. Rep*, 12:19.

Becze, M., Jameson, H., et al. (2015). EIP-1: EIP Purpose and Guidelines. Ethereum Improvement Proposals. no. 1, October.

Buterin, V. (2014). Ethereum white paper.

Buterin, V., Weiss, Y., Gazso, K., Patel, N., Tirosh, D., Nacson, S., and Hess, T. (2021). Erc-4337: Account abstraction using alt mempool. Ethereum Improvement Proposals.

Coyle, K. (2023). Account abstraction: Use cases, technical overview, and security considerations.

Dannen, C. (2017). *The EVM*, pages 47–67. Apress, Berkeley, CA.

Ethereum (2023a). Eip 2938: Account abstraction.

Ethereum (2023b). Eip 86: Abstraction of transaction origin and signature.

Hildenbrandt, E., Saxena, M., Rodrigues, N., Zhu, X., Daian, P., Guth, D., Moore, B., Park, D., Zhang, Y., Stefanescu, A., and Rosu, G. (2018). *KEVM: A Complete Formal Semantics of the Ethereum Virtual Machine*, pages 204–217. Institute of Electrical and Electronics Engineers.

National Institute of Standards and Technology (2013). Digital signature standard. Technical Report NIST.FIPS.186-4, National Institute of Standards and Technology.

Oder, T., Pöppelmann, T., and Güneysu, T. (2014). Beyond ecdsa and rsa: Lattice-based digital signatures on constrained devices. In *Proceedings of the 51st Annual Design Automation Conference*, DAC '14, page 1–6, New York, NY, USA. Association for Computing Machinery.

Pu, S. and Lam, J. S. L. (2021). Blockchain adoptions in the maritime industry: a conceptual framework. *Maritime Policy & Management*, 48(6):777–794.

Revolut (2023). A virtual card.

Sadhya, V. and Sadhya, H. (2018). Barriers to adoption of blockchain technology.

Solidity contributors. Solidity documentation. https://docs.soliditylang.org/en/v0.8.20/.

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

World Economic Forum (2015). Deep shift - technology tipping points and societal impact. http://www3.weforum.org/docs/WEF_GAC15_Technological_Tipping_Points_report_2015.pdf.

Zahnentferner, J. (2018). Chimeric ledgers: Translating and unifying utxo-based and account-based cryptocurrencies. Cryptology ePrint Archive, Paper 2018/262. https://eprint.iacr.org/2018/262.