Migração de Arquitetura de Identidade Autossoberana: Transição de Hyperledger Indy para Hyperledger Besu na Plataforma ChainID

Osvaldo Filho[1], **Silvio Queiroz**[1], **Leobino Sampaio**[1]

[1]Instituto de Computação – Universidade Federal da Bahia (UFBA)
Salvador – BA – Brasil

{osvaldo.lima,silvio.queiroz,leobino}@ufba.br

Abstract. *Self-Sovereign Identity (SSI) enables users to maintain full control over their digital data without centralized intermediaries. However, pioneering frameworks like Hyperledger Indy face increasing challenges regarding scalability and compatibility with modern distributed ecosystems. This paper presents the migration of the ChainID platform from the Hyperledger Indy blockchain to Hyperledger Besu. The transition aims to overcome technical limitations, such as the latency of the RBFT consensus and the technical debt of the legacy AnonCreds protocol, by adopting W3C standards and Ethereum Virtual Machine (EVM) compatibility. The results demonstrate that the new architecture, based on Veramo framework, enhances interoperability and facilitates integration with broader initiatives such as the CarbonID project.*

Resumo. *A Identidade Digital Descentralizada (IDD) permite que usuários controlem seus dados sem intermediários. Contudo, frameworks pioneiros como o Hyperledger Indy enfrentam desafios de escalabilidade e integração com ecossistemas modernos. Este artigo apresenta o processo de migração da plataforma ChainID da blockchain Hyperledger Indy para o Hyperledger Besu. A transição visa superar limitações técnicas, como a latência do consenso RBFT e a dívida técnica do protocolo AnonCreds, adotando padrões W3C e a compatibilidade com a Ethereum Virtual Machine (EVM). Os resultados demonstram que a nova arquitetura, baseada no framework Veramo, amplia a interoperabilidade e facilita a integração com projetos como o CarbonID.*

1. Introdução

A gestão de identidades digitais tem passado por uma mudança de paradigma, transitando de modelos centralizados e federados para a Identidade Autossoberana (SSI). Nesse modelo, o uso de Identificadores Descentralizados (DIDs) e Credenciais Verificáveis (VCs) permite que os indivíduos controlem seus próprios dados sem a dependência de autoridades centrais [Preukschat and Reed 2021]. Frameworks pioneiros, como o Hyperledger Indy, foram fundamentais para estabelecer as bases da SSI. Contudo, a evolução do ecossistema revelou limitações críticas no Indy que comprometem sua viabilidade a longo prazo. Além das latências inerentes ao protocolo de consenso RBFT (*Redundant Byzantine Fault Tolerance*), que dificultam a escalabilidade, a plataforma enfrenta um processo de obsolescência técnica acentuado pela redução do suporte da comunidade desenvolvedora e pelo isolamento em relação às redes modernas compatíveis com a Ethereum Virtual Machine (EVM) [Pflanzner et al. 2022, Hyperledger Committee 2024].

A principal contribuição deste artigo é o desenvolvimento do Veramo Agent ChainID, um agente de borda modular e extensível em TypeScript que substitui a solução legada baseada em ACA-Py, aderindo estritamente aos padrões W3C para DIDs e VCs por meio do método *did:ethr*. A arquitetura proposta introduz um mecanismo de revogação *on-chain* fundamentado no padrão ERC-5539 sobre a blockchain Hyperledger Besu, estabelecendo uma fonte de verdade única, auditável e imutável que otimiza a verificação externa de credenciais. Adicionalmente, o trabalho descreve uma estratégia de *multitenancy* segura com isolamento físico de bancos de dados PostgreSQL e gestão segregada de chaves criptográficas, o que viabiliza a oferta de SSI como serviço (SaaS) com alta confiabilidade. Por fim, demonstra-se por meio de análise quantitativa que a nova estrutura supera a pilha tecnológica original em métricas críticas, alcançando um ganho de desempenho de 9,5x na ativação de organizações e reduzindo significativamente a latência no ciclo de vida de credenciais verificáveis.

2. Referencial Teórico

Nesta seção, apresentam-se os conceitos fundamentais que sustentam a transição da plataforma ChainID [Queiroz et al. 2021], abordando os pilares da Identidade Autossoberana e as características das redes Hyperledger Indy e Besu.

2.1. Identidade Autossoberana e Padrões W3C

A Identidade Autossoberana baseia-se na premissa de que o indivíduo deve possuir e controlar sua identidade digital de forma independente de qualquer autoridade central. Essa arquitetura fundamenta-se em três pilares definidos pela W3C: os Identificadores Descentralizados, as Credenciais Verificáveis e os Registros de Dados Verificáveis (VDR), como as blockchains [Preukschat and Reed 2021]. Enquanto o DID provê um identificador persistente e globalmente único que não requer um registro central, as VCs permitem a troca de informações atestadas por emissores de forma segura e criptograficamente verificável [W3C 2019a, W3C 2019b].

2.2. Hyperledger Indy

O Hyperledger Indy foi projetado especificamente para SSI, introduzindo inovações como o protocolo de prova de conhecimento zero (ZKP) via *AnonCreds*. Entretanto, sua arquitetura baseia-se no consenso RBFT, que exige que todas as transações de escrita passem por um nó mestre, gerando gargalos de latência em redes com alto volume de operações ou grande número de validadores [Vukolić 2017].

Além das limitações de performance, o ecossistema Indy/Aries enfrenta um isolamento técnico. Por não ser compatível com a EVM, a integração de identidades Indy com smart contracts modernos exige pontes (*bridges*) complexas, aumentando a superfície de ataque e a dívida técnica. Conforme relatórios recentes da Linux Foundation (2024), observa-se uma migração global de grandes projetos para redes baseadas em EVM devido à maior maturidade das ferramentas de desenvolvimento e ao suporte da comunidade [Hyperledger Committee 2024].

2.3. Hyperledger Besu e o Método *did:ethr*

O Hyperledger Besu surge como uma alternativa robusta, sendo um cliente Ethereum open-source que suporta redes permissionadas. Ao contrário do Indy, o Besu implementa

Tolerância a Falhas Bizantinas de Quorum (QBFT), que oferece finalidade imediata e maior throughput para redes corporativas [Hyperledger Besu Documentation 2025].

A migração para o Besu permite a adoção do método *did:ethr*, que ancora os DIDs diretamente na blockchain Ethereum ou em redes compatíveis [Looker et al. 2018]. Essa abordagem simplifica a gestão de chaves e a revogação de credenciais, que pode ser realizada por meio de contratos inteligentes na própria rede, garantindo interoperabilidade nativa com o ecossistema DeFi e projetos de rastreabilidade como o CarbonID.

2.4. O Framework Veramo

Para viabilizar essa transição sem comprometer a flexibilidade do ChainID, utiliza-se o framework Veramo[1]. Diferente do Aries Cloud Agent Python (ACA-Py), que possui um acoplamento forte com o ecossistema Indy, o Veramo é uma biblioteca modular em TypeScript que permite a criação de agentes de identidade agnósticos à rede. Através de plugins, o Veramo facilita a gestão de múltiplos métodos de DID e formatos de credenciais como JWT e JSON-LD.

3. Arquitetura e Implementação da Solução

A solução desenvolvida, denominada Veramo Agent ChainID, substitui o agente legado (ACA-Py) por uma estrutura modular em TypeScript/Node.js. Este agente implementa nativamente os padrões W3C para DIDs e VCs utilizando o método *did:ethr* e garantindo compatibilidade com a EVM através de contratos inteligentes.

3.1. Seleção de Framework e Validação Tecnológica

A escolha do novo framework de agente foi precedida por uma pesquisa exploratória e experimental, visando equilibrar a compatibilidade com a arquitetura original e os novos requisitos de integração.

A escolha do novo framework baseou-se em uma matriz de decisão ponderada (Tabela 1), avaliando critérios como suporte a padrões W3C, maturidade da comunidade e compatibilidade nativa com a EVM. O critério de maior peso foi a 'Compatibilidade nativa com EVM/Besu', fator considerado eliminatório para viabilizar a criação de DIDs diretamente na rede. Embora o ACA-Py ofereça excelente suporte ao protocolo Aries, sua falta de compatibilidade nativa com ecossistemas EVM o penalizou severamente. Consequentemente, o framework Veramo despontou como a escolha vencedora. Sua arquitetura modular e o suporte nativo ao ecossistema Ethereum (via plugin *did-provider-ethr*) permitiram o cumprimento dos requisitos da rede Hyperledger Besu.

3.2. Arquitetura em Camadas

O sistema foi estruturado seguindo o padrão de arquitetura em camadas para garantir a separação de responsabilidades e facilitar a manutenção. Conforme ilustrado na Figura 1, o sistema organiza-se em quatro camadas principais:

1. Camada de Interface (API): Expõe endpoints RESTful para as aplicações clientes, validando tokens e abstraindo a complexidade das operações de SSI.

[1] `https://veramo.io`

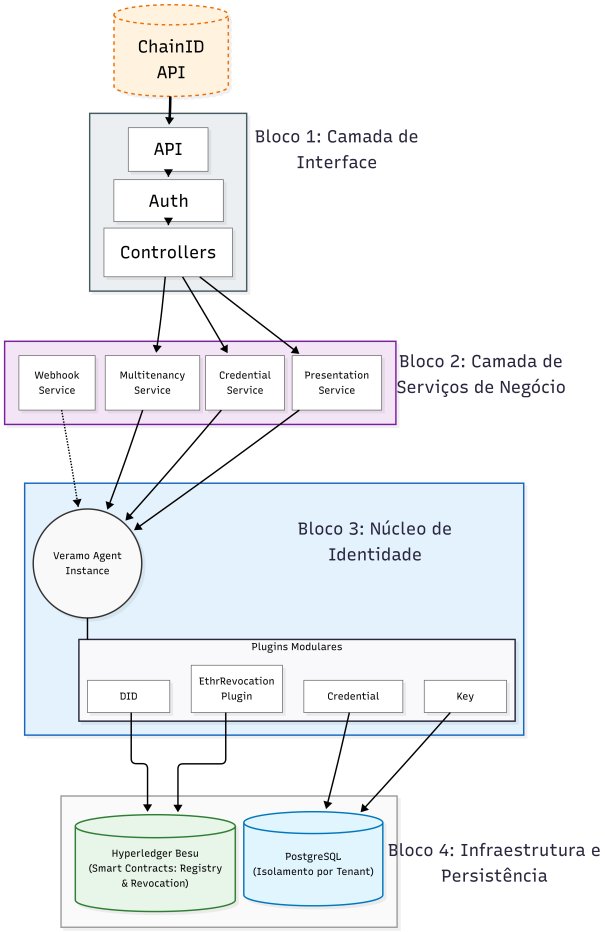

Figura 1. Arquitetura em Camadas do Veramo Agent ChainID

2. Camada de Serviços e Negócio: Onde reside a lógica de orquestração, incluindo os serviços de multitenancy e o ciclo de vida de credenciais.
3. Núcleo de Identidade: Sustentada pelo framework Veramo, gerencia as primitivas criptográficas, criação de DIDs e assinaturas de VCs seguindo padrões W3C.
4. Camada de Infraestrutura e Persistência: Composta pelo banco de dados PostgreSQL (*off-chain*) para metadados e chaves criptografadas, e pela rede Hyperledger Besu (*on-chain*), que atua como o VDR.

3.3. Estratégia de *Multitenancy* e Isolamento

Para preservar a capacidade multi-inquilino da plataforma original, foi implementada uma estratégia de isolamento total no nível do banco de dados. Cada organização (*tenant*) possui um banco de dados PostgreSQL dedicado e um cofre de chaves segregado. O *Multitenancy Service* gerencia o provisionamento dinâmico dessas instâncias, garantindo que os DIDs e as chaves privadas de um inquilino sejam fisicamente inacessíveis por outros.

3.4. Integração com Hyperledger Besu e Padrões W3C

A migração para o Besu permitiu a adoção do método *did:ethr*, baseado no padrão ERC-1056 [Looker et al. 2018]. Diferente do modelo anterior, os identificadores são mapeáveis

Tabela 1. Matriz de Decisão de Frameworks de Identidade

Critérios	**Peso**	**Veramo**	**ACA-Py + Universal Resolver**	**DIDKit**
Compatibilidade nativa com EVM/Besu	60	60	0	60
Suporte ao protocolo Aries	40	40	40	0
Fornece API RESTful	25	0	25	0
DIDs e VCs W3C	10	10	10	10
Validação de Prova	5	5	5	5
Webhook	10	0	10	0
Suporte a multi-tenant	5	0	5	0
Suporte a wallet	5	5	5	5
Licença open-source	5	5	5	5
Persistência	5	5	5	5
Comunidade*	10	10	10	5
Suporte	10	0	0	0
Fórum	10	10	10	0
Total	**200**	**150**	**130**	**95**

*Critério Comunidade: Ativa (10), Moderada (5), Existente (1), Inexistente (0).
Fonte: Elaborada pelo autor (2025).

para endereços Ethereum, permitindo que os atores assinem transações na rede utilizando a mesma infraestrutura de identidade.Para a gestão de revogação, a arquitetura utiliza o padrão ERC-5539 (*Revocation List Registry*) [Bolte et al. 2022]. Ao contrário do modelo de tails files do Indy, o status da credencial é alterado via transação *on-chain* em um contrato inteligente, fornecendo uma fonte de verdade única, imutável e facilmente auditável por verificadores externos. As credenciais são emitidas no formato W3C JWT, facilitando a interoperabilidade com o ecossistema Web3 e aplicações tradicionais.

4. Resultados e Análise Comparativa

Para validar a eficácia da nova arquitetura, a solução foi submetida a uma avaliação em duas frentes: uma análise qualitativa, comprovando a interoperabilidade com sistemas legados (Moodle) e aplicações Web3 (CarbonID), e uma análise quantitativa de desempenho.

4.1. Metodologia e Ambiente de Testes

Os testes consistiram na execução de 10 iterações para cada operação crítica, visando obter médias estatisticamente representativas. O ambiente experimental foi configurado da seguinte forma:

- Agente: Estação de trabalho com processador Intel Core i7-13700KF, 32 GB de RAM e sistema operacional Windows 11. No que tange aos ambientes de *runtime*, utilizou-se o Node.js v21.7.1 e o Java OpenJDK v18.0.2. Especificamente para a arquitetura legada, o agente ACA-Py (v0.12.1) foi instanciado via Docker (v24.0.6).
- *Mobile Wallet*: Dispositivo Samsung Galaxy A55 com Android 16. No cenário da arquitetura legada, utilizou-se a *BC Wallet* como aplicação cliente, em virtude

de inconsistências de comunicação observadas entre a versão original da *ChainID Wallet* e a infraestrutura da *Indicio Testnet*. Para a nova arquitetura, utilizou-se a versão mais recente da *ChainID Wallet*, desenvolvida a partir da customização do *framework* de código aberto da Sphereon.

- Infraestrutura Blockchain: Para a arquitetura legada (Indy), utilizou-se a rede *Indicio Testnet*. Para a nova arquitetura, utilizou-se o Testbed da RNP rodando Hyperledger Besu com consenso QBFT e tempo de bloco de 4 segundos.

Ressalta-se uma limitação metodológica inerente a este cenário: a diferença de infraestrutura (Testnet pública vs. Testbed controlado) introduz um viés de latência de rede. Optou-se por manter o ACA-Py na Indicio Testnet para refletir com fidelidade o cenário de uso real e o gargalo histórico enfrentado pela plataforma legada, embora isso infle parcialmente o ganho de desempenho medido frente ao ambiente local do Besu.

4.2. Análise Qualitativa

A validade funcional da arquitetura foi demonstrada em dois cenários que atestam a versatilidade do Veramo Agent ChainID em ecossistemas de confiança distintos.

- Interoperabilidade Federada (Moodle): Comprovou-se a capacidade de atuar como ponte (*gateway*) entre identidades federadas tradicionais (*Web 2.0*) e SSI. Utilizando o IDP Blockchain (Apereo CAS) como tradutor de protocolos, o agente validou as credenciais dos usuários na rede Besu e converteu os atributos em asserções SAML 2.0 para acesso ao Moodle. O teste validou a eficácia dos módulos de verificação e das notificações assíncronas via webhooks.
- Integração com Ecossistema Web3 (CarbonID): Validou-se a interoperabilidade nativa com redes baseadas na EVM. Por meio do método *did:ethr*, os identificadores gerados tornaram-se mapeáveis para endereços Ethereum, permitindo que os atores utilizem a mesma infraestrutura de chaves para sua identidade soberana e para assinar transações na blockchain Besu.

4.3. Análise de Desempenho

Os resultados demonstram um ganho de desempenho expressivo na maioria das operações com a adoção da nova arquitetura baseada em Besu e Veramo, conforme detalhado na Tabela 2.

Tabela 2. Comparativo de Tempo de Resposta (Média e Desvio Padrão em milissegundos)

Operação	ACA-Py (Indy)	DP	Veramo (Besu)	DP	Variação
Ativar organização	10.081	± 504	1.054	± 52	∼9,5×
Deploy de esquema	6.612	± 330	1.519	± 76	∼4,3×
Criar credencial	76	± 4	29	± 2	∼2,6×
Revogar credencial	5.514	± 275	1.638	± 82	∼3,3×
Criar solicitação de prova	200	± 10	784	± 39	0,25×
Validar prova de credencial	800	± 40	250	± 12	∼3,2×

Fonte: Elaborada pelo autor (2025).

4.4. Análise Quantitativa e Discussão Técnica

Os dados coletados (Tabela 2) revelam que a migração para uma arquitetura baseada em EVM e padrões W3C superou a performance da pilha tecnológica original na maioria dos indicadores críticos.

Abaixo, discutem-se os fatores técnicos que impulsionaram esses resultados:

- Otimização de Provisionamento (9,5× de Speedup): A ativação de novas organizações apresentou o ganho mais expressivo. No ACA-Py, o suporte a múltiplos inquilinos exige a criação de carteiras lógicas criptografadas dentro do banco de dados do agente, um processo de alto custo computacional. Na nova arquitetura, o isolamento é realizado via provisionamento direto de bancos de dados PostgreSQL dedicados, permitindo uma inicialização significativamente mais ágil.
- Ciclo de Vida e Custo Criptográfico: A emissão de credenciais foi 2,6× mais rápida. Isso ocorre porque as *AnonCreds* exigem a construção de ZKPs já no ato da emissão, enquanto o formato JWT adotado no Veramo utiliza assinaturas digitais padrão (EdDSA/ECDSA), que possuem menor custo computacional.
- Eficiência na Revogação (3,3×): A interação direta com o contrato inteligente ERC-5539 na rede Besu demonstrou latência inferior à operação de atualização de acumuladores e tails files necessária no Hyperledger Indy.
- Verificação de Provas (3,2×): Validar uma assinatura JWT é uma operação atômica e leve comparada à verificação de apresentações *AnonCreds*, que envolve validação matemática complexa de predicados ZKP e consultas a estados de revogação não acumulados.

A única métrica que apresentou retrocesso foi a criação de solicitações de prova ($0,25\times$). Isso ocorreu por uma decisão de design na nova arquitetura: para garantir a resiliência em fluxos assíncronos, cada solicitação agora gera registros persistentes no banco de dados do inquilino antes de retornar ao cliente, enquanto o modelo anterior priorizava operações em memória.

5. Conclusão

A evolução dos ecossistemas de Identidade Autossoberana (SSI) tem exigido a transição de soluções isoladas para arquiteturas interoperáveis e integradas à infraestrutura da Web3. Este trabalho apresentou a migração da plataforma ChainID para uma infraestrutura fundamentada em Hyperledger Besu e no framework Veramo. O objetivo de superar a obsolescência tecnológica e as barreiras de integração impostas pelo modelo anterior foi plenamente alcançado. A nova arquitetura demonstrou ser capaz de gerenciar identidades descentralizadas e credenciais verificáveis diretamente sobre uma rede compatível com a EVM.

A migração envolveu compromissos arquiteturais. A substituição do formato *AnonCreds* pelo JWT resultou na perda da capacidade nativa de divulgação seletiva e provas de conhecimento zero (ZKP). Esse trade-off é perfeitamente aceitável em cenários de transparência pública corporativa, como a rastreabilidade ambiental do CarbonID. Contudo, torna-se crítico em casos de uso de dados pessoais sensíveis, onde a minimização da exposição é fundamental.

Para trabalhos futuros, planeja-se a reintrodução da divulgação seletiva. Além de investigar assinaturas BBS+, será priorizada a adoção do padrão SD-JWT, alinhando a plataforma às recentes diretrizes da identidade digital europeia (EUDI Wallet). Planeja-se também a integração do suporte ao padrão DIDComm para ampliar a interoperabilidade.

Em suma, a migração documentada reposiciona a ChainID como uma plataforma moderna e performática, preparada para a próxima geração de aplicações descentralizadas.

Agradecimentos

Este trabalho foi viabilizado pelo apoio da Rede Nacional de Ensino e Pesquisa (RNP) por meio do projeto GT-CarbonID.

Referências

Bolte, P., Leifermann, L., and von der Bey, D. (2022). Eip-5539: Revocation list registry. Ethereum Improvement Proposal. Accessed: 2025-11-23.

Hyperledger Besu Documentation (2025). Proof of authority consensus in hyperledger besu. Online Documentation. Accessed: 2025-11-23.

Hyperledger Committee (2024). 2024 annual review: Hyperledger indy. Project report. Accessed: 2025-12-19.

Looker, T., Sabadello, M., et al. (2018). Eip-1056: Ethereum lightweight identity (did registry). Ethereum Improvement Proposal. Accessed: 2025-11-23.

Pflanzner, T., Baniata, H., and Kertesz, A. (2022). Latency analysis of blockchain-based ssi applications. *Future Internet*, 14(10).

Preukschat, A. and Reed, D. (2021). *Self-Sovereign Identity: Decentralized Digital Identity and Verifiable Credentials*. Manning, Shelter Island, NY.

Queiroz, S., Greve, F., Sampaio, L. N., and Marques, E. (2021). Plataforma para gestão de identidades descentralizadas baseada em blockchain. In *Anais do XXI Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais (SBSEG 2021)*, pages 29–42, Belém, PA, Brasil. Sociedade Brasileira de Computação (SBC).

Vukolić, M. (2017). Rethinking permissioned blockchains. In *Proceedings of the ACM Workshop on Blockchain, Cryptocurrencies and Contracts*, BCC '17, page 3–7, New York, NY, USA. Association for Computing Machinery.

W3C (2019a). Decentralized identifiers (dids) v1.0: Core architecture, data model, and representations. Specification, World Wide Web Consortium (W3C). Acessado em: 2025-11-23.

W3C (2019b). Verifiable credentials data model 1.0: Expressing verifiable information on the web. Recommendation, World Wide Web Consortium (W3C). Acessado em: 2025-11-23.