Personal Blockchain: Uma Estratégia de Gerenciamento de Consentimento de Dados Pessoais Dirigida ao Usuário

1st Mateus Bruno Araújo
Programa de Pós-Graduação em Ciência da Computação
Instituto Federal do Ceará
Fortaleza, Brasil
mateus.bruno.araujo01@aluno.ifce.edu.br

2nd Cidcley Teixeira de Souza
Programa de Pós-Graduação em Ciência da Computação
Instituto Federal do Ceará
Fortaleza, Brasil
cidcley@ifce.edu.br

***Resumo*—Os marcos legais de proteção de dados têm como objetivo dar poder aos usuários de decidir e de consentir sobre seus dados e a quem esses são remetidos. O principal marco desse tema é a GDPR (*General Data Protection Regulation*) na Europa. O fornecimento de estratégias para o suporte à implementação das questões relacionadas às novas legislações tem se tornado cada vez mais evidente. Contudo, o que se tem observado é que os sistemas atuais basicamente estão registrando o consentimento, ainda continuam armazenando e gerenciando as informações dos usuários de forma unilateral. É proposto nesse trabalho a Personal Blockchain, uma carteira de dados baseada em blockchain para armazenamento de dados pessoais. O intuito da Personal Blockchain é empoderar os usuários com o maior controle possível sobre os seus dados. Para viabilizar a Personal Blockchain, também, foi desenvolvido um novo modelo arquitetural, a blockchain matricial com âncora. Esse modelo trará aumento na densidade da blockchain tradicional e proporá uma estrutura de auditoria por meio de uma blockchain tradicional pública.**

***Index Terms*—gerenciamento de consentimento, blockchain, LGPD, GDPR**

I. Introdução

Os marcos legais sobre proteção de dados têm como objetivo dar poder aos usuários de decidir e de consentir sobre seus dados e a quem esses são remetidos. Dois dos principais marcos regulatórios desse tema são a GDPR (*General Data Protection Regulation*) na Europa [18], e a Lei Geral de Proteção de Dados (LGPD) no Brasil [5].

Contudo, o que se tem observado é que os sistemas atuais basicamente estão registrando o consentimento do usuário e continuam armazenando e manipulando as informações. O usuário sempre depende de terceiros que detém a guarda dos dados, ou seja, têm pouco, ou quase nenhum, poder direto sobre seus dados.

Este trabalho focará no direito de consentimento; seus corolários, os direitos de retificação e de esquecimento (artigo 5º e 17º da GDPR); e na minificação do processamento dos dados [14]. Sempre, dando ênfase ao empoderamento do usuário frente ao modelo tradicional de controle dos dados pelos requisitantes (empresas ou instituições que de forma autorizada tem a posse dos dados).

É importante salientar que nos últimos anos tem se observado um crescimento notório de aplicações baseadas em *blockchain*. Essas aplicações são utilizadas para dar suporte às mais variadas formas de aplicações [9]. Porém, *blockchain*, por conceito, é uma tecnologia que traz a imutabilidade como um de seus diferenciais, dessa forma, causando um claro conflito com o espírito legal das leis de proteção de dados [6].

Nesse cenário, essa pesquisa apresenta um novo mecanismo de gerenciamento de consentimento, no qual, tanto o armazenamento dos dados, quanto às ações realizadas sobre eles são feitas pelo próprio usuário. Para tanto, é proposto o conceito da *Personal Blockchains*, uma carteira de dados pessoais com uma estrutura de dados baseada em *blockchain* para armazenar dados pessoais e os eventos do ciclo de vida dos dados armazenados. Para viabilizar esse conceito, foi criada uma arquitetura de *blockchains* distribuídas de cadeias curtas, que são ancoradas criptograficamente em uma *blockchain* pública de cadeia longa, esse modelo arquitetural é a *blockchain* matricial com âncora.

II. Solução Proposta

Com o intuito de possibilitar uma melhor gestão dos dados por parte do dono dos dados, dando maior ingerência sobre o ciclo de vida desses dados, é proposto nesse trabalho a *personal blockchain*, uma carteira de dados pessoais baseada em *blockchain*. Como forma de viabilizar tecnicamente a *personal blockchain*, o trabalho também propõe o modelo arquitetural da *blockchain* matricial com âncora. Esse modelo arquitetural tem como razão a capacidade de possibilitar o uso de *blockchains* pequenas e pessoais (com apenas um único ator na rede).

Uma carteira de dados, aqui, é considerada como um meio de armazenamento e compartilhamento de dados. De forma análoga pode ser colocado as carteiras digitais de cartões que são utilizadas para pagamentos etc. No caso, a *personal blockchain* é uma solução que viabiliza em seu modelo o controle e compartilhamento de dados.

Nessa sessão, será apresentada a motivação técnica para a criação do modelo proposto, além de apresentar como o

modelo mitiga ou soluciona os problemas de uma carteira de dados pessoais. Por fim, serão apresentados os conceitos formais da solução *personal blockchain*.

A. Motivação e Soluções

Com o aumento nas promulgações de leis de proteção de dados, cada vez mais, faz-se necessário que sistemas tenham que aderir aos seus princípios legais. Com o intuito de possibilitar a criação de uma solução de compartilhamento de dados sensíveis do usuário, o trabalho traz a *personal blockchain*, uma carteira de dados pessoais baseada em *blockchain*. A estrutura de dados *blockchain* foi utilizada com base principal de dados, pois suas características inatas, como a auditabilidade e a irretratabilidade, são requisitos importantes para a solução.

No desenvolvimento da solução, foram identificadas duas grandes dificuldades para possibilitar uma rede *blockchain* individual aderente à GDPR e a LGPD, que são a possibilidade de um ataque de consenso e a facilidade de recriação dos blocos, caso um bloco interno seja adulterado.

Um ataque de consenso acontece quando um ator ou um grupo de atores se utilizam da maioria dos nós em uma regra de consenso para criar ou modificar dados de forma maliciosa [2]. Já a solução proposta tem como premissa a existência apenas de um ator na rede, ou seja, esse ator sempre detém a maioria na regra de consenso. Caso queira, ele pode incluir ou retirar dados sem a necessidade de passar por crivo de outros. Porém, essa possibilidade abre uma vulnerabilidade: o usuário pode adulterar os dados e imputar ao cliente dos dados (a quem foi compartilhado os dados) o fato de que ele está utilizando informações inverídicas ou adulteradas. Tal ocorrência pode causar danos ao negócio que se utilizou da solução.

Para solucionar o problema do ataque de consenso, foi utilizada uma estratégia de âncora externa [19]. Com a âncora as informações inseridas na carteira de dados são lastreadas em uma *blockchain* pública tradicional com vários outros participantes. Caso seja necessário, podem ser auditados todos os dados da carteira.

A segunda dificuldade encontrada foi a possibilidade de recriação da *blockchain* com utilização de colisões de hash. Teoricamente, as funções de criação de hash são injetoras, ou seja, a cada semente passada para a função é gerado um hash distinto. Entretanto, existem casos em que a função de hash gera uma colisão, isto é, gera o mesmo hash para entradas diferentes. O risco aumenta levando-se em consideração que todos os dados inseridos estão no domínio de apenas um ator. Esse ator tendo poder computacional suficiente, é possível adulterar a estrutura da *blockchain* mesmo com a âncora.

Para mitigar o problema de uma recriação total ou parcial da cadeia, foi desenvolvido um modelo arquitetural bidimensional para a *blockchain*, denominada de *blockchain* matricial. Como pode ser visto na Figura 1, a *blockchain* matricial é estruturada acoplando blocos adicionais à estrutura padrão da *blockchain* tradicional. Todos os blocos são interdependentes, utilizam-se do conteúdo do bloco anterior para gerar seu hash, mitigando assim a possibilidade de recriação da cadeia, pois seria necessário colidir 2N hashes pelo menos, sendo N a ordem de entrada das informações na cadeia (ex: o quinto bloco de informação tem ordem 5).

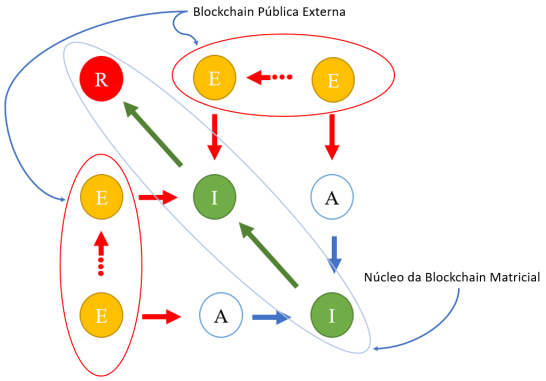

Figura 1. Representação de uma *blockchain* matricial

B. Personal Blockchain

A *personal blockchain* é uma carteira de dados com aderência aos princípios de consentimento, de esquecimento, de retificação e de minificação do processamento de dados. Ela é uma implementação do conceito de *blockchain* matricial com âncora. Seu principal objetivo é empoderar os seus usuários com a capacidade de gestão direta sobre seus dados. O conceito apresentado traz uma forma de possibilitar o gerenciamento dos dados, incluindo seu compartilhamento e revogação de acesso. Também, traz a capacidade de registrar todos os eventos do ciclo de vida dos dados, dotando a arquitetura de características como a irretratabilidade do evento, confidencialidade e integridade dos dados. Assim, dependendo da implementação da *personal blockchain* e do ambiente de implantação, pode-se agregar outras características de segurança.

A *personal blockchain* é uma carteira pessoal de dados, dentro dela, conterão todos os dados que o usuário quiser gerenciar e os eventos que ocorrerem com esses dados. Eles estarão registrados em uma *blockchain* de rede individual e ancorados em uma *blockchain* externa. Internamente, a carteira de dados realizará uma deleção lógica, quando necessário. Mesmo sendo uma deleção lógica, ninguém, além do usuário, terá a possibilidade de resgatar os dados excluídos. Uma exclusão análoga é utilizada para as versões anteriores de dados retificados. Dessa forma, a solução adere aos princípios de consentimento, esquecimento e retificação dos dados.

Todo evento ocorrido no ciclo de vida do dado é registrado na *blockchain* pessoal e ancorado na *blockchain* externa, assim, torna possível a entidade que recebeu os dados compartilhados e o dono dos dados se defender contra acusações de adulteração de dados. A estrutura prevê um processamento restrito das informações, apenas o usuário da *personal blockchain* e o solicitante processam os dados, minificando a quantidade de processamento dos dados.

A seguir será apresentado em mais detalhes o modelo físico arquitetural da solução e também será especificada a estrutura

de comunicação do sistema distribuído que contém a *personal blockchain* e seus serviços e métodos.

1) Modelo Físico: O modelo físico arquitetural da *personal blockchain* pode ser dividido em duas partes: o modelo estrutural interno e a arquitetura física de comunicação.

O modelo estrutural interno define como é montada a *blockchain* matricial com âncora. Como pode ser visto na Figura 1, existem 4 tipos de blocos. Os blocos identificados com a letra "I" guardam as informações inseridas na carteira digital e o ciclo de vida dessas informações. Pode ser visto o ciclo de vida do dado na Figura 2. Também, é possível evidenciar o algoritmo interno de criação dessa estrutura no Algoritmo 1.

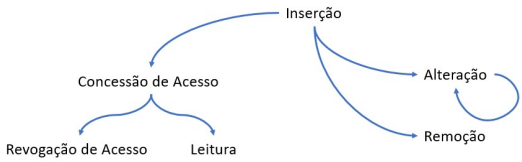

Figura 2. Ciclo de vida do dado.

Algoritmo 1: Algoritmo de criação de blocos da *blockchain* matricial com âncora.

Input: informacao

$Inicialização$:
$blocoInformação \leftarrow gerarBlocoI(informação)$
$últimoDireita \leftarrow gerarBlocoA(blocoInformação)$
$últimoEsquerda \leftarrow gerarBlocoA(blocoInformação)$
$anchorGenerate(últimoDireita)$
$anchorGenerate(últimoEsquerda)$
return

Além dos blocos de informações, existem os blocos auxiliares identificados com a letra "A", são responsáveis por dificultar a recriação maliciosa da cadeia; os blocos de eixo, identificados pela letra "E", são blocos que apoiam os blocos auxiliares para dificultar a recriação da cadeia, mas tem como função principal ancorar a *blockchain* matricial à *blockchain* pública. Os blocos de eixo não fazem parte fisicamente da *blockchain* matricial, a ligação é apenas lógica, servindo para possibilitar a auditoria nos blocos internos. O quarto tipo de bloco é o raiz, identificado pela letra "R", e tem o mesmo objetivo que em *blockchains* tradicionais. É importante perceber que a estrutura da *blockchain* tradicional ainda está presente no núcleo da *blockchain* matricial.

A arquitetura física de comunicação detalha a interação entre as diferentes aplicações que compõem o sistema distribuído. Existem três aplicações envolvidas na solução da *personal blockchain*: a própria *personal blockchain*, a carteira de dados; o sistema do requerente das informações, o cliente; e a *blockchain* pública que lastreia as informações inseridas na *personal blockchain*.

Na Figura 3, pode ser visto o modelo geral de funcionamento e, também, todas as fachadas de serviços baseados nos contratos.

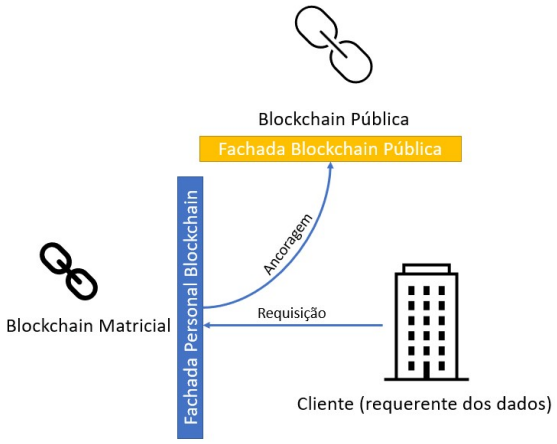

Figura 3. Arquitetura física de comunicação.

2) Contratos de Serviços: Toda comunicação entre atores do sistema é realizada por meio de implementações de seus contratos. A seguir serão apresentados os contratos que devem ser seguidos para a comunicação ser realizada. Ressalta-se que o modelo apresentado levou em conta a utilização do cenário de comunicação via internet, ou seja, respeitando a pilha TCP/IP, utilizando HTTPs e comunicação via RESTful. Também, será indicado os códigos de status de respostas HTTP que serão retornados seguindo a RFC 2616 e RFC 7231 [13].

Os métodos presentes no contrato da interface de programação da aplicação (API) da *personal blockchain* são:

- *insert*: método utilizado pelo dono da *personal blockchain* para inserir novas informações. Ao realizar a inserção, o dado é registrado na *blockchain* matricial com âncora e será retornado HTTP 201. Caso o usuário tente inserir dois dados com o mesmo identificador, não será armazenado o segundo e o método retornará erro HTTP 400.
- *update*: método utilizado pelo dono da *personal blockchain* para atualizar uma informação já inserida anteriormente. A informação é registrada como um novo bloco de informação na *blockchain* matricial com âncora. Será retornado HTTP 200. Se não houver registro com identificador informado, será retornado HTTP 400.
- *remove*: método utilizado pelo dono da *personal blockchain* para remover uma informação já inserida anteriormente. A informação de remoção é registrada como uma nova informação na *blockchain* matricial com âncora. Quando a operação for realizada com êxito retornará HTTP 200, caso não, retornará HTTP 400.
- *read*: método utilizado pelo cliente para ler os dados que tem acesso na *personal blockchain*. Os dados são lidos um de cada vez e para cada requisição é verificado se o requisitante tem permissão para acessar o dado solicitado. Caso não tenha permissão, será retornado HTTP 403.

Caso não exista o identificador informado, será retornado HTTP 400. Caso o identificador seja localizado e o cliente tenha permissão, será retornado HTTP 200 junto com a informação no corpo da resposta.

- *consent*: método utilizado pelo cliente para solicitar consentimento de utilização de um dado. Após realizado essa requisição, o dono da carteira poderá aceitar ou não compartilhar o dado. Caso a resposta seja negativa ou o dado não exista na carteira, o retorno será HTTP 400. Caso seja positiva, o retorno será HTTP 200 e no corpo do pacote terá um identificador único do cliente. O identificador deverá ser utilizado para se identificar quando for necessário utilizar o método *read*.
- *revoke*: método utilizado pelo dono da *personal blockchain* para revogar, parcialmente ou totalmente, o acesso a dados compartilhados, retorna HTTP 200 caso haja o registro do cliente e HTTP 400 caso não.
- *getConsents* e *getHistory*: são métodos utilizados pelo dono da carteira de dados para identificar as permissões atualmente concedidas e o histórico de algum dado em específico. O retorno padrão é HTTP 200. Caso o identificador do dado não seja encontrado, retorna HTTP 400.

Todos os métodos apresentados, quando acionados, registram seu acionamento e modificações realizadas na *blockchain* matricial com âncora.

A fachada da *blockchain* pública externa indicada na Figura 3 só apresenta um único método o *anchorGenerate*. Esse método tem a função de criar um bloco na *blockchain* pública que servirá de âncora para a *personal blockchain*.

O cliente não necessitará de uma fachada por padrão, apenas deve se atentar que os dados obtidos não devem ser armazenados em seus domínios, se não forem estritamente necessários. Caso os dados sejam armazenados, fica a cargo do cliente consultar os dados na API da *personal blockchain* periodicamente para apagar ou modificar os dados, quando necessário.

III. Trabalhos Relacionados

A crescente procura pelo tema *blockchain* com direitos de proteção de dados é notório como pode ser visto em Haque [7] e em Huang [9]. Dentre os principais temas encontrados nas revisões bibliográficas, é possível encontrar: deleção, modificação de dados e gerenciamento de consentimento, porém todos os artigos encontrados nas revisões focam no lado do cliente das informações, não no lado do dono dos dados. As soluções discutidas a seguir também focam no requerente dos dados.

Os termos deleção e modificação de dados são, nada mais, que as expressões técnicas do direito de esquecimento e de retificação, dois dos princípios das leis de proteção de dados pelo mundo [11]. Sobre esses princípios, Stan e Miclea [17] ressaltam a incompatibilidade entre eles e sistemas baseados em *blockchain*. Dentro do trabalho, é concluído que a única maneira do direito de deleção ou retificação ser possível com estruturas baseadas em *blockchain* é considerar que a exclusão de dados não significa necessariamente exclusão física, ou seja, a exclusão lógica ser considerada aderente aos princípios citados.

Nos casos de exclusão e modificação dos dados em um sistema baseado em *blockchain*, é possível considerar que a edição, nada mais é, que a substituição de um dado registro por outro [15]. Em Kadena [11], é sugerido um algoritmo de consenso em uma *blockchain* que se deseja excluir dados. Há uma votação e a maioria aprovando, os dados podem ser removidos. Nesse caso, pode ocorrer outro problema: remover alguns blocos antigos e interromper a cadeia. Em contraste ao trabalho aqui proposto, toda a gestão dos dados está em uma única parte do processo, mesmo que seja utilizada uma estratégia distribuída, a solução de Kadena ainda apresentará problemas com o princípio da minificação do processamento de dados.

Uma outra solução para exclusão e retificação é o apresentado em Al-Zaben [1]. No trabalho, é proposto uma arquitetura *blockchain* com uma estrutura de dados *off-chain*, que usa um banco de dados local para preservar e armazenar os dados. Ao armazenar os dados fora da cadeia, o sistema estará em conformidade com o direito de esquecimento e retificação, uma vez que, os dados fora da cadeia podem ser excluídos ou alterados a qualquer tempo. Bayle também utilizou processo semelhante [3]. Comparativamente à *personal blockchain*, é possível identificar que todo o processo é executado por outrem, sendo que o dono dos dados apenas dá a instrução, o gerenciamento de fato dos dados é realizado pelo cliente dos dados, que nesse caso se comporta quase como um dono, visto que tem a guarda dos dados.

Em Lee [12], é proposta uma *blockchain* customizável, que os usuários têm o direito de retirar o consentimento, assim sendo, solicitar a remoção dos dados. Nessa estrutura, os proprietários dos dados escolhem o nível de dificuldade de modificação de sua transação antes de enviá-la. Importante salientar que esse método utiliza uma estrutura *multichain*: uma única cadeia principal para a aprovação da transação e várias cadeias laterais para modificação da transação; as modificações de transação são realizadas nas *sidechains*. Apresentam as mesmas diferenças ao trabalho proposto que Bayle e Al-Zaben.

Não menos importante, a literatura também aborda muito sobre as questões do gerenciamento de consentimento e revogação de acesso aos dados, como pode ser visto em Al-Zaben [1], Belen Saglam [4], Herian [8], Huang [10] e Sim [16]. As abordagens trazem pequenos sistemas e utilizam *smart contracts* para viabilizar o gerenciamento de consentimento e revogação de acesso. Quando retirado o consentimento, e consequente exclusão dos dados, é utilizada estratégias similares as já comentadas. Entretanto, a utilização de contratos inteligentes, quando usados em sistemas distribuídos, não é aderente ao princípio da minificação dos processadores e dos consumidores. Quando utilizado em um sistema interno, com apenas uma parte sendo detentora do sistema, este sistema estará sob o controle do cliente dos dados, o dono dados apenas exercerá um poder parcial. O gerenciamento do ciclo de vida dos dados será do cliente dos dados.

Em toda a bibliografia apresentada, não foi encontrado

nenhuma arquitetura com intuito similar de possibilitar uma blockchain de rede individual, ou seja, com apenas um ator na rede. Também, não foi encontrado um arquitetura de registro de dados focado no gerador dos dados no contexto das leis de proteção de dados. Sendo assim não foi possível realizar comparações diretas entre soluções já existente e a proposta.

IV. Cenário de uso

Para mostrar a viabilidade da solução proposta, são apresentados cenários de uso que abrangem as principais situações encontradas no uso da *personal blockchain* para armazenamento de informações do usuário e para o gerenciamento do consentimento do usuário, incluindo o direito de esquecimento e o direito de retificação.

De forma geral, os cenários iniciam a interação com o cliente (o requisitante) solicitando o endereço da API da *blockchain* pessoal. O usuário da *personal blockchain* fornecerá o endereço. Em posse do endereço, o cliente poderá iniciar a o processo de solicitação e obtenção dos dados. A Figura 4 demonstra o processo de obtenção de consentimento.

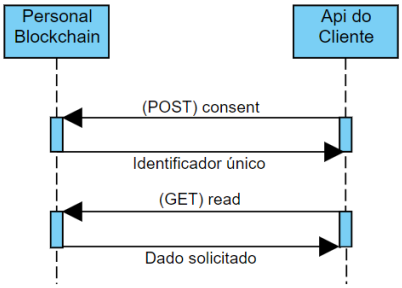

Figura 4. Diagrama de sequência de concessão do consentimento e atualização de dados.

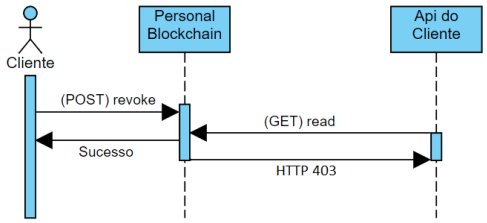

Figura 5. Diagrama de sequência de concessão da remoção e revogação.

Após o consentimento ter sido dado, o cliente poderá solicitar o dado em si. Ao solicitar o dado, a *personal blockchain* informa o último valor daquele dado para o cliente. Com a posse da informação, o cliente poderá armazena-la em sua base de dados ou não, porém a fonte primária dos dados é a *personal blockchain* e todas as alterações de dados e de consentimento estarão registradas na *personal blockchain* e ancoradas na *blockchain* pública. Caso o cliente decida armazenar os dados em seus domínios, ele precisará periodicamente consultar a API da *personal blockchain* para verificar se houve alguma modificação no dado ou na concessão do acesso.

O usuário, a qualquer tempo, pode alterar ou inserir novos dados na carteira. Também poderá remover dados ou revogar acessos de clientes, como mostrado na Figura 5. Nesse ponto, acontece a inversão da lógica de responsabilidades dos sistemas tradicionais, aqui, o requerente dos dados é quem tem que verificar se ainda tem as concessões de utilização dos dados, não o dono dos dados que tem a obrigação de informar aos clientes.

1) Caso exemplo: Concessão de dados: O usuário deseja fazer uma compra na empresa XPTO S.A., mas não tem cadastro nela.

O usuário informa o endereço da sua *personal blockchain*. Com posse do endereço, a empresa solicita via API o acesso aos dados, o usuário concede o acesso aos dados. Ao receber resposta positiva, a empresa solicita os dados à carteira de dados pessoais. Após realizada a leitura, o usuário consegue realizar suas compras com o cadastro já realizado, sem necessitar preencher um cadastro manualmente.

2) Caso exemplo: Retificação de dados: Meses depois da primeira compra, o usuário decide novamente realizar uma compra na empresa XPTO S.A. Porém, durante os meses anteriores, ele havia se mudado e atualizado o endereço na *personal blockchain*.

Ao realizar a compra, o sistema da XPTO S.A. traz a informação do endereço atualizada, pois já havia atualizado os dados. Com os dados corretos, a compra é finalizada.

3) Caso Exemplo: Solicitação de Esquecimento: Pouco tempo após a segunda compra, o usuário considera que não quer mais ter vínculo com a empresa XPTO S.A.

Por meio da aplicação da *personal blockchain*, ele revoga as permissões. Periodicamente, a empresa XPTO S.A. verifica se houve alguma modificação nos dados ou nos acessos dos dados presentes na *personal blockchain*, quando identifica que o acesso foi revogado, a empresa faz a exclusão dos dados necessários.

4) Caso Exemplo: Auditoria: Alguns meses depois, órgãos de controle decidem fiscalizar a empresa XPTO S.A. sobre o cumprimento das leis de proteção de dados vigentes. Durante o processo, identifica que existem dados do usuário registrados na base de dados da empresa. Para se certificar que os dados realmente tinham sido autorizados, o órgão solicita a *personal blockchain* do usuário. Os dados da carteira são validados com suas últimas ancoras geradas. Após a validação, é identificado que o usuário realmente havia concedido os dados, porém já havia retirado a concessão.

V. Conclusão

O trabalho traz um novo modelo arquitetural de sistema distribuído para propiciar o desenvolvimento de uma carteira digital de dados pessoais, a *Personal Blockchain*. Foram realizadas algumas análises de viabilidade da arquitetura, as quais indicaram a possibilidade do uso em um ambiente produtivo.

A abordagem proposta possibilita que uma *blockchain* curta tenha uma robustez maior do que uma *blockchain* tradicional

de tamanho similar. O auxílio de uma estrutura de âncora apoia a resiliência da *blockchain* matricial, tornando-a mais confiável e auditável, duas das principais características das *blockchains* tradicionais.

Os próximos passos do desenvolvimento é a inclusão das estruturas de segurança para obrigar um maior controle de acesso aos dados. Também, será realizado testes com medição de quantidade de dados gerados em casos reais e com quantidade de dados massivos, estressando o sistema ao máximo, comparando o tempo de processamento em relação as *blockchain* tradicionais. Outro ponto que deve ser abordado em projetos futuros, é o desenvolvimento de benchmarks de carteiras de dados que não utilizem *blockchain*.

Por fim, será desenvolvido um aplicativo mobile para possibilitar uma interface amigável para solução. A partir desse aplicativo, poderá ser iniciado testes de aceitação da solução, provendo uma possibilidade para uso de forma real no dia a dia.

REFERÊNCIAS

[1] Al-Zaben, N., Hassan Onik, M. M., Yang, J., Lee, N.-Y., and Kim, C.-S. (2018). General Data Protection Regulation Complied Blockchain Architecture for Personally Identifiable Information Management. In 2018 International Conference on Computing, Electronics Communications Engineering (iCCECE), pages 77–82, Southend, United Kingdom. IEEE.

[2] Antonopoulos, A. M. (2017). Consensus attacks. In McGovern, T., editor, Mastering Bitcoin: Programming the Open Blockchain, 1005 Gravenstein Highway North, Sebastopol, CA 95472. O'Reilly Media, Inc.

[3] Bayle, A., Koscina, M., Manset, D., and Perez-Kempner, O. (2018). When Blockchain Meets the Right to Be Forgotten: Technology versus Law in the Healthcare Industry. In 2018 IEEE/WIC/ACM International Conference on Web Intelligence (WI), pages 788–792, Santiago. IEEE.

[4] Belen Saglam, R., Aslan, C. B., Li, S., Dickson, L., and Pogrebna, G. (2020). A Data-Driven Analysis of Blockchain Systems' Public Online Communications on GDPR. In 2020 IEEE International Conference on Decentralized Applications and Infrastructures (DAPPS), pages 22–31, Oxford, United Kingdom. IEEE.

[5] Brasil (2018). Lei geral de protec ao de dados pessoais (LGPD).

[6] Casino, F., Politou, E., Alepis, E., and Patsakis, C. (2020). Immutability and Decentralized Storage: An Analysis of Emerging Threats. IEEE Access, 8:4737–4744.

[7] Haque, A. B., Najmul Islam, A., Hyrynsalmi, S., Naqvi, B., and Smolander, K. (2021). GDPR Compliant Blockchains – A Systematic Literature Review. IEEE Access, pages 1–1.

[8] Herian, R. (2020). Blockchain, GDPR, and fantasies of data sovereignty. Law, Innovation and Technology, 12(1):156–174.

[9] Huang, H., Kong, W., Zhou, S., Zheng, Z., and Guo, S. (2021). A Survey of State-of-the-Art on Blockchains: Theories, Modelings, and Tools. ACM Computing Surveys, 54(2):1–42. Number: 2.

[10] Huang, W.-C., Yeh, L.-Y., and Huang, J.-L. (2019). A Monitorable Peer-to-Peer File Sharing Mechanism. In 2019 20th Asia-Pacific Network Operations and Management Symposium (APNOMS), pages 1–4, Matsue, Japan. IEEE.

[11] Kadena, E. and Holicza, P. (2018). Security Issues in the Blockchain(ed) World. In 2018 IEEE 18th International Symposium on Computational Intelligence and Informatics (CINTI), pages 000211–000216, Budapest, Hungary. IEEE.

[12] Lee, N.-Y., Yang, J., Onik, M. M. H., and Kim, C.-S. (2019). Modifiable Public Blockchains Using Truncated Hashing and Sidechains. IEEE Access, 7:173571–173582.

[13] MDN WEB DOCS COMMUNITY. Códigos de status de respostas HTTP - HTTP — MDN. Disponível em: ¡https://developer.mozilla.org/pt-BR/docs/Web/HTTP/Status¿. Acesso em: 5 ago. 2022.

[14] PARLAMENTO EUROPEU. (UE) 2016/679 (Regulamento Geral sobre a Proteção de Dados), 27 abr. 2016.

[15] Politou, E., Casino, F., Alepis, E., and Patsakis, C. (2020). Blockchain Mutability: Challenges and Proposed Solutions. IEEE Transactions on Emerging Topics in Computing.

[16] Sim, W. L., Chua, H. N., and Tahir, M. (2019). Blockchain for Identity Management: The Implications to Personal Data Protection. In 2019 IEEE Conference on Application, Information and Network Security (AINS), pages 30–35, Pulau Pinang, Malaysia. IEEE.

[17] Stan, O. P. and Miclea, L. (2019). New Era for Technology in Healthcare Powered by GDPR and Blockchain. In Vlad, S. and Roman, N. M., editors, 6th International Conference on Advancements of Medicine and Health Care through Technology; 17–20 October 2018, Cluj-Napoca, Romania, volume 71, pages 311–317. Springer Singapore, Singapore. Series Title: IFMBE Proceedings.

[18] Voigt, P. and von dem Bussche, A. (2017). The EU General Data Protection Regulation (GDPR): A Practical Guide. Cham, 1st ed. 2017 edition.

[19] Weber, I., Lu, Q., Tran, A. B., Deshmukh, A., Gorski, M., and Strazds, M. (2019). A Platform Architecture for Multi-Tenant Blockchain-Based Systems. In 2019 IEEE International Conference on Software Architecture (ICSA), pages 101–110, Hamburg, Germany. IEEE.