EDUARDO ALEXANDRE FRANCISCON

MODELO DOSSIÊ ESTENDIDO PARA MULTIPASTA COM BLOCKCHAIN PARA GERENCIAMENTO DE DADOS IMUTÁVEIS

Dissertação apresentada ao Programa de Pós-Graduação em Informática da Pontifícia Universidade Católica do Paraná como requisito parcial para obtenção do título de Mestre em Informática.

Área de Concentração: *Ciência da Computação*

Orientador: Prof. Dr. Edson Emílio Scalabrin

CURITIBA

Outubro/2020

AGRADECIMENTOS

Aqui me desprendo das formalidades da dissertação, para que possa demonstrar de forma sincera meus agradecimentos a cada nome aqui citado, também não serei breve, afim de ser justo com todos que me ajudaram. Antes de tudo, sou grato a Deus, por ter me dado energia em momentos que pensei que esgotaria. Esther, minha filha, você ainda não entende isso, mas o papai só começou esse mestrado, porque pensava em ti, você nenê, foi a primeira e principal inspiração do pai ter lutado por esse título, e por isso sempre terei essa lembrança de carinho por você minha filhota. Também sou muito grato a minha família, por ter me dado apoio nos momentos de dificuldade, que não foram poucos, pois a rotina de viagens semanais, desemprego e apertos financeiros foram grandes obstáculos a serem vencidos. Por isso, meu pai Nilton, e minha mãe Realci, serei eternamente grato e ciente do esforço da parte de vocês, a meus irmãos, Gabriel e Sarah, eu agradeço pelo apoio em toda a trajetória. Agradeço a Hevelin, minha ex-esposa, que apesar de não estarmos mais juntos hoje, foi uma figura importante, que aguentou firme os momentos de dificuldade financeira e trabalhou duro para mantermos a casa enquanto eu estudava. Sou muito grato ao professor Jones Granatyr, que me aconselhou e me iniciou no mestrado, e ainda foi companheiro de artigos publicados mais tarde. Obrigado professor Roberto Cesar da Silva Padilha por ter aceito ser meu orientador do TCC num momento em que não pude continuar com meu outro orientador e foi parceiro até o final, já sabendo que eu entraria para o mestrado no ano seguinte. Professor Gilson Reis, sou grato pelo apoio e a carta de recomendação que o senhor enviou num pedido que lhe fiz, guardo considerações. Edson Doebber, lhe sou muito grato pelo emprego concedido no momento em que eu estava desempregado, mesmo sendo um curto período de tempo, garantiu minha vida financeira para os estudos. Aos Motoristas Valdecir Luiz Togni e João da Silva, Claudia e envolvidos da casa de apoio em Curitiba, assim como as meninas do setor da saúde da prefeitura de Francisco Beltrão, agradeço pelas caronas até Curitiba e as voltas até Francisco Beltrão, me pouparam bastante dinheiro, o que tornou capaz a continuação de meus estudos. Em especial seu João, agradeço e guardo calorosa consideração pelos serviços de frete que o senhor me chamou e me ajudou a passar por outra fase muito difícil de aperto financeiro. Agradeço também ao Menin, que me chamou para trabalhar nessa fase complicada. Odair Andrade, lhe sou grato pelos serviços em obras nos momentos em que foi difícil juntar dinheiro, obrigado também Renata Bruna Szepanhuk Grohe, pela vaga no Formosa Pub na fase final de meu mestrado. Por último, mas não menos importante, sou muito grato a minha querida amiga Geovani Muller, pela oportunidade de trabalho, breve, mas importante em um momento difícil. Enquanto estudei, fiz grandes amigos dentro da PUC-PR, e por essas amizades, sou muito feliz em telas e grato pelos momentos de estudos e ajudas, mas em especial gostaria de agradecer ao Marcos Nascimento Pereira, por ser meu companheiro e grande amigo em todos os momentos desse mestrado, saiba que fez uma grande diferença em minha carreira com nossas ideias trocadas. Ewerton Schaedler, pela grande ajuda no desenvolvimento prático desse projeto, sem você isso teria sido muito mais difícil. Otto R. Lessing, outro querido

amigo, me ajudou e foi parceiro de produção científica, tendo comigo dois artigos. Professor Roberto Hirochi Herai, grande figura que foi amigo e conselheiro em grande período do meu mestrado, tenho muita consideração por sua pessoa. Quero dizer que no período de aulas, o time de professores PPGIa foi ótimo e guardo boas recordações de todas as aulas, aprendi e evolui muito nesse período, mas, em especial eu deixo registrado meu carinho pelas professoras Sheila Reinehr e Andreia Malucelli, que em suas aulas tiveram paciência com todas as minhas dúvidas e foram grandes orientadoras na minha produção científica, a qual gerou frutos e publiquei bons artigos. Ao professor Edson Emílio Scalabrin eu abro espaço especial para um agradecimento que faço com toda sinceridade possível, foi e ainda é mais que professor para mim, ao mesmo tempo que me orientou na universidade, foi um grande amigo, abriu as portas de sua casa no momento que precisei de pouso em Curitiba, confiou ao ponto de apresentar sua família, a qual também sou grato pela calorosa recepção todas as vezes que estive em sua casa. Professor Scalabrin, seu exemplo como profissional e amigo é louvável, saiba que os exemplos deixados pelo senhor a mim, serão repassados no futuro, se um dia eu tiver a oportunidade de orientar novos estudantes. Minha mais sincera gratidão por isso tudo professor. Agradeço a Andreia dos Santos Goffi, que foi minha professora de inglês e me ajudou a passar no meu exame de suficiência na língua. Também agradeço a Juliane Mayara Casarim Machado pelo companheirismo em me ajudar a encontrar uma universidade para fazer o exame de suficiência. Por fim, agradeço a Monaliza Fortuna, querida companheira que me ajudou e me deu energia na reta final desse trabalho, quando eu já estava cansado. Monaliza, você apareceu a pouco na minha vida, mas em um momento importante, saiba que guardo carinho especial por ti a respeito disso tudo. Se esqueci de alguém, minhas sinceras desculpas, mas saiba que mesmo assim sou verdadeiramente grato.

Sumário

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIAÇÕES

- MAS (Multi-Agent Systems)
- DF (Directory Facilitator)
- GAD (Grafo Acíclico Direcionado)
- WTA (Winner-Take-All)
- AHP (Analytic Hierarchy Process)

Título: Modelo *Dossiê* estendido para *multipasta* com *Blockchain* para gerenciamento de dados imutáveis

Resumo: *Este projeto visa a concepção e avaliação de um modelo computacional que permite desenvolver um sistema multiagente, cujos registros das transações e feedbacks sejam locais. As hipóteses que norteiam a proposta fundamentam-se: (h1) no Dossiê—que é uma estrutura de dados particular e de controle—que permite manter registros locais imutáveis; e (h2) na rede Blockchain descentralizada que fornece infraestrutura para implementar mecanismos de controle de dados logicamente e fisicamente distribuídos; (h3) na estruturação multipasta do Dossiê que facilita a organização e comunicação da informação entre um agente requerente A e um agente requerido B. A garantia de segurança e integridade de cada transação é feita por meio de smart contract. O resultado obtido é um ambiente descentralizado, onde cada agente A pode usar a informação de cada pasta do Dossiê de um agente B com a convicção de que a informação não foi alterada desde a sua geração. Para avaliação do projeto, foi desenvolvido um protótipo, em que cada entidade distribuída é dotada de uma instância do gerenciador local de cadeias de blocos e dispõe de uma rede P2P para o manter uma estrutura pública, no formado de livro-razão—ou Ledger, com o resumo de cada Dossiê de cada agente. Deve-se salientar que um dado agente C, que solicita o Dossiê D de outro agente P, tem a prerrogativa de consultar ou não o resumo de D registrado no livro-razão público para verificar a integridade de D. Essa possibilidade de fazer consulta por amostragem é importante como comportamento social.* O resultado do projeto é um sistema baseado em agentes, onde cada agente implementa uma estrutura que permite manter de forma decentralizada o histórico dos seus feedbacks recebidos; uma particularidade é que cada agente pode gerir um Dossiê com multidimensionalidade e garantir que as informações trocas são confiáveis.

Palavras-chaves: *Blockchain*, *Ledger*, Agente de Software, *Dossiê*.

Title: Extended *Dossier* model for multi-folder with blockchain for unmodified data management

Abstract: This project aims to provide the conception and evaluation of a computational model, that allows developing a multiagent system, which records transactions and feedbacks in a local place. The hypotheses that guide the proposal are based on: (h1) in the Dossier - which is a particular data and control structure - that allows unchanging local records to be maintained; and (h2) in the decentralized Blockchain network that provides infrastructure to implement logically and physically distributed data control mechanisms; (h3) in the multi-folder structure of the Dossier that facilitates the organization and communication of information between a requesting agent A and a required agent B. The guarantee of security and integrity of each transaction is made through a smart contract. The result obtained is a decentralized environment, where each agent A can use the information from each folder of the agent B Dossier with the conviction that the information has not been changed since its generation. To evaluate the project, a prototype was developed, in which each distributed entity is provided with an instance of the local block chain manager and has a P2P network to maintain a public structure, in the form of ledger — or Ledger, with the summary of each Dossier for each agent. It should be noted that a given agent C, who requests Dossier D from another agent P, has the prerogative to consult or not the summary of D recorded in the public ledger to verify the integrity of D. This possibility of consulting by sampling is important as social behavior. The result of the project is a system based on agents, where each agent implements a structure that allows maintaining the history of feedbacks received in a decentralized way; a particularity is that each agent can manage a Dossier with multidimensionality and ensure that the information exchanged is reliable.

Keywords*: Blockchain, Ledger, Software Agent, Dossier.*

1. INTRODUÇÃO

A confiança é um tema tratado em várias partes da comunidade humana, em especial em comunicações que agregam algum tipo de valor, tanto no longo prazo quanto no curto prazo. Estas comunicações acontecem de forma muito natural, quando se dirige ao mercado e adquire um produto físico ou quando se adquire algo pela Internet, se deseja algo que assegure a efetividade da transação relativa à compra.

Porém, estabelecer confiança para realizar uma transação é uma tarefa cada vez mais desafiadora, pois, hoje uma parte significativa das transações são realizadas por sistemas eletrônicos, onde as pessoas não têm acesso facilitado para saber se estão sendo enganadas ou não. A quebra da confiança que pode resultar em roubo e esse último pode se apresentar de várias formas, e.g., juros abusivos, aumento indevido do preço de um produto. É necessário estabelecer uma forma em que a transação ocorra sem a quebra de confiança. Este é um desafio que a ciência da computação enfrenta e tenta resolver por meio de modelos de confiança implementados nos sistemas [SILVA, 2017].

Com a expansão das redes de Internet e a facilidade de acesso a ela, as pessoas estão cada vez mais dispostas a disponibilizar produtos e serviços por meio da Internet. Portais que expõem estoques gigantes em uma rede distribuída já não são mais novidades há algum tempo [SUBMARINO, 2019]. Portais como *Mercado Livre* fornecem facilidades para que as pessoas possam comprar e vender produtos a partir de um *smartphone* [MERCADO LIVRE, 2019].

Por trás de uma plataforma web de compra e venda, há um sistema complexo que tenta garantir que toda transação feita seja segura e confiável. Porém, ainda há casos de pessoas que compram ou vendem seus produtos e nunca recebem sua parte da transação. Falsidade ideológica que é a cumplicidade entre indivíduos para prejudicar outro, são alguns dos problemas que a informática ainda não tratou por inteiro. E essa necessidade de unir a segurança de sistema com a engenharia social é

tratada por meio de um modelo de confiança, e.g., o modelo Dossiê mantenha segura e inviolada em um ambiente aberto de agentes de software [SILVA, 2017].

Em termos técnicos, um dos esforços de trabalho concerne o uso da tecnologia *Blockchain* para implementar o modelo de confiança *Dossiê* [SILVA, 2017]. Tal modelo permite que cada agente de um dado sistema armazene localmente seus *feedbacks* e os disponibilize aos outros agentes sobre demanda. A abordagem *Dossiê* é importante na medida que ela permite selecionar bons parceiros para realizar as transações de forma eficiente e segura. O modelo proposto baseia-se no paradigma numérico, utiliza fontes de informação direta e indireta e também evita que um dado agente omita informações sobre *feedbacks* recebidos. O nosso desafio foi dar ao modelo *Dossiê* uma arquitetura e uma implementação robusta para fomentar aplicações baseadas em agentes e na tecnologia *Blockchain*.

Ao unir o modelo Dossiê e a tecnologia *Blockchain*, assume-se que todos os integrantes da rede de agentes desejam prover algum tipo de serviço e hospedar localmente o seu *Dossiê*. Logo, o *Dossiê* replicado para todos os agentes envolvidos torna-se uma aplicação *Multidossiê*. Em outras palavras, a existência de vários agentes fornecedores de serviços dá origem a distintos *dossiês* compondo uma arquitetura descentralizada de informação.

1.1 Motivação e Descrição do Problema

Problemas relacionados a confiança em transações é objeto de pesquisa importante dentro do campo da computação. Soluções com sistemas centralizados ou distribuídos já foram aplicadas. Os modelos de confiança surgiram para resolver problemas que podem ocorrer na comunicação dos agentes envolvidos em uma transação [ARTZ e GIL, 2007].

O uso de sistema centralizado com a função de mediador tem-se mostrado ineficiente. Quando se observa escândalos de corrupção, ou até mesmo a simples falta de compromisso de uma das partes em algum esquema de troca de favores ou valores já desqualifica o uso dessa abordagem [DIALLO, 2018]. A descentralização de sistemas vem sendo estudada e traz uma nova abordagem no quesito confiança para agentes

de *software*. Tecnologias existentes são capazes de encontrar padrões ou reconhecer possíveis falhas ou corrupção por meio de desempenho técnico, pois uma vez que vários indivíduos tem acesso ampliado dos dados existentes na rede, a detecção de irregularidades se torna mais evidente, sem necessidade de abordagens complexas para detecção de anomalias. O uso de um *hash* ou de uma cadeia de blocos descentralizada prova isso [NOERLINA, 2018]. A tecnologia *Blockchain* tem na sua gênese uma forma de trabalhar a transparência e a imutabilidade da informação. Ela traz novas perspectivas de sistemas para implementação de transações a nível comercial ou público [DIALLO, 2018].

A eliminação de um agente centralizador interfere drasticamente na questão da confiança em relação a um agente que deseja negociar algo em uma rede de provedores de serviços, e precisa de garantias que o resultado de tal negociação acontece corretamente. Uma abordagem com essas características traz um novo modelo de se estabelecer a confiança. Um exemplo disso é o Mercado Livre, que conforme as pessoas compram produtos de um vendedor, as mesmas avaliam o vendedor e estabelecem no tempo um conceito relacionado a esse vendedor, podendo vir na forma simbólica ou numérica. Quando outras pessoas desejam negociar com tal vendedor, é possível verificar a confiança a ele atribuída no decorrer de suas vendas, trazendo assim mais tranquilidade para a pessoa que deseja comprar. Por outro lado, é preciso pensar que essa abordagem tem pouco a contribuir com vendedores que iniciaram suas vendas a pouco tempo.

Mesmo assim, é notório o aumento de indivíduos que fazem negócios com outras pessoas sem se importar com existência ou não de um mecanismo que garanta segurança em uma negociação. O Uber, aluguel de quartos para viajantes ou compras entre dois agentes comuns são exemplos cada vez mais presentes no cotidiano. Essas transações são facilitadas por modelos computacionais de fácil acesso às pessoas. Por exemplo, aplicativos de *smartphones* tem o poder de estabelecer uma comunicação de qualidade, o que facilita muitos negócios entre pessoas de forma descentralizada.

Um sistema multiagente tem fragilidade relacionado a confiança entre seus integrantes. Porém, é possível projetar um ambiente em que todos os traços das transações sejam registrados de forma imutável, permitindo que interações entre os

agentes sejam visíveis por todos, forçando-os a manter a sua integridade. Baseado na tecnologia *Blockchain* e no modelo de confiança *Dossiê*, a proposta encerra um ambiente descentralizado *Multidossiê*, onde cada agente pode interagir com outro, sem a necessidade de agente centralizador que garanta a confiança e que ainda seja transparente, imutável e capaz de estabelecer um perfil de cada agente para verificar a integridade de cada agente.

1.2 Hipóteses

As hipóteses que norteiam essa trabalho fundamentam-se: (h1) no Dossiê—que é uma estrutura de dados particular—que permite manter registros locais imutáveis, assim testemunhos mantidos pelos agentes avaliados podem ser considerados legítimos à medida que as informações trafegadas são verificadas quanto a integridade e autenticidade por meio de algoritmos de criptografia assimétrica; (h2) na rede *Blockchain* descentralizada que fornece infraestrutura para implementar controle e dados logicamente e fisicamente distribuídos; e *(h3) na estruturação multipasta do Dossiê que facilita a organização e comunicação da informação entre um agente requerente e um agente requerido.*

1.3 Objetivos

Grande parte dos modelos de confiança que fazem parte de uma rede distribuída ou descentralizada não usam um ambiente *Blockchain* para garantir a imutabilidade dos registros das transações. Desta forma, este trabalho visa implementar o modelo de *Dossiê* em um ambiente descentralizado se utilizando da tecnologia *Blockchain*. A integração deste modelo de confiança com a tecnologia *Blockchain* visa definir um cenário onde se possa verificar a integridade dos dados e dos agentes envolvidos por meio da proposta de um multipasta, ou seja, onde cada agente envolvido no ambiente Blockchain possui seu *Dossiê*, que contém um sistema de multipasta, para cada serviço prestado, assim também, como garantir transações seguras por meio de *smart contracts*, imutabilidade e transparência.

O ambiente final deve prover um ambiente descentralizado, onde o agente *A* possa usar as informações do *Dossiê* do agente *B* com a convicção de que elas não foram alteradas desde a sua geração. O ambiente gerencia cadeias de blocos descentralizadas com transações individuais. As transações são imutáveis e os registros de dados são garantidos por *smart contracts* em cada Dossiê local; local deve-se ler o espaço lógico e/ou físico de cada agente.

1) Caracterizar o estado da arte em relação as arquiteturas propostas de construção de sistemas baseados em registros de dados imutáveis.
2) Por meio de um ambiente totalmente descentralizado por Blockchain Ethereum, avaliar a simplicidade das cadeias de blocos imutáveis como estrutura básica de implementação de aplicação com baixo esforço.
3) Desenvolver um protótipo de sistema multiagente, onde cada entidade distribuída é dotada de uma instância do gerenciador de cadeias de blocos local.
4) Por meio de transações no ambiente Ethereum, avaliar o esforço de implementação do protocolo de atualização de cadeia de blocos para a abordagem Dossiê *multipasta*.

2. TECNOLOGIA BLOCKCHAIN

Este Capítulo apresenta a tecnologia *Blockchain* juntamente com os fundamentos teóricos necessários para o entendimento deste trabalho, bem como o estado da arte da cadeia de blocos e suas respectivas ferramentas. As tecnologias que envolvem cadeias de blocos são compostas de vários pontos importantes, os quais são necessários para o entendimento em si e apresentados nas próximas seções

2.1 Blockchain

Mantendo a distribuição e a independência de cada agente, a viabilização do armazenamento compartilhado e imutável dos traços das transações entre os participantes, por exemplo de uma transação, passa pela adoção de uma arquitetura e de um suporte tecnológico apropriado. A operacionalização mais simples seria a sincronização dos dados. Nesta linha, a sincronização total ou parcial dos traços das transações é o primeiro passo, como visto na Figura 1, onde mostra-se diversos agentes fazendo uso dos mesmos dados compartilhados. O próximo passo é garantir a disponibilidade, o compartilhamento e a imutabilidade de tais traços. Como já dito, a proposta é usar as abordagens: *Blockchain* como estrutura básica para a indexação dos dados e *Ledger (livro-razão)* como estrutura intermediária para o registro do traço de cada transação de forma imutável.

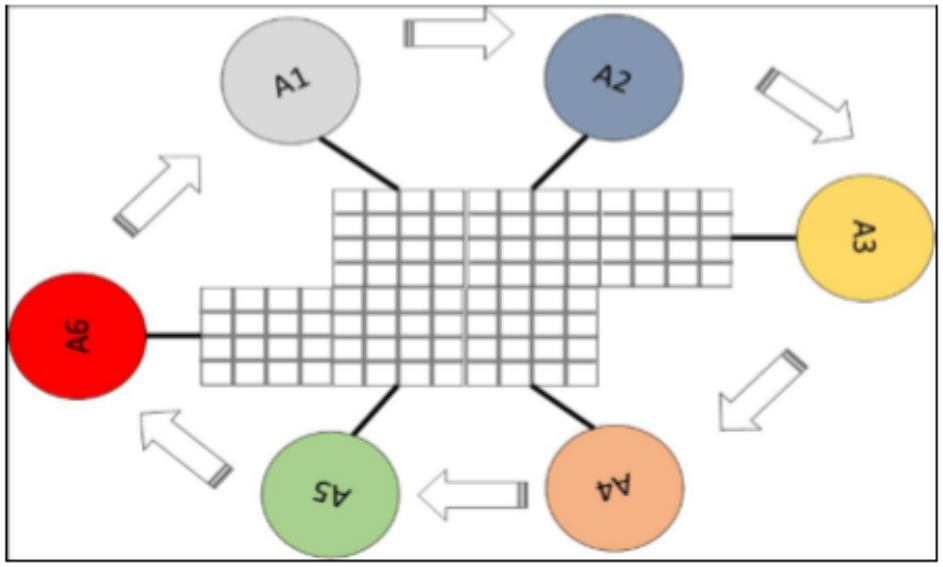

Figura 1. Interação entre aplicativos com a sincronização do registro dos traços de transações.

A Figura 1 ilustra que cada nó—agente aplicação—compartilha o mesmo espaço de dados e o fluxo ilustra que cada transação feita por um nó torna-se conhecida dos demais. O espaço compartilha deve ser tratado como uma estrutura imutável, similar a um livro razão.

Assim, a estrutura *Blockchain* viabiliza as transações *on-line* seguras [(NAKAMOTO, 2008), (SWAN, 2015)]. Uma *Blockchain* operacionaliza um *Ledger* digital descentralizado que registra as transações em muitos computadores de tal forma que as transações registradas não possam ser alteradas retroativamente (ANTONOPOULOS, 2014). Isso permite aos participantes verificar e auditar as transações de maneira barata. Tais transações são autenticadas pela colaboração em massa, impulsionada por interesses coletivos próprios. O resultado é um fluxo de trabalho robusto onde a incerteza dos participantes em relação à segurança dos dados é marginal. O uso da *Blockchain* remove a característica de **reprodutibilidade infinita de um ativo digital**. Ele assegura que cada unidade de valor é transferida apenas uma vez, resolvendo o problema de dupla despesa (CHAUM, 1985). As *Blockchains* foram descritas como um protocolo de troca de valores. Uma Blockchain pode atribuir o direito a um título porque fornece um registro que obriga a oferta e a aceitação. Do ponto de vista técnico, uma *blockchain* é uma *hashchain* dentro de outra *hashchain*.

Uma base de dados estruturada na forma de uma *blockchain* consiste em dois tipos de registros: transações e blocos. Cada bloco mantem um lote de transações válidas que são *hash* codificados em uma árvore *Merkle* [(MERKLE, 1987), (ANTONOPOULOS, 2014)]. Cada bloco inclui o *hash* do bloco anterior na cadeia de blocos, ligando os dois. Os blocos ligados formam uma cadeia. Este processo iterativo assegura a integridade do bloco anterior e, de todo o caminho de volta para o bloco original de gênese.

Toda transação deve ser validada. O processo segue o seguinte fluxo: as transações são propagadas na rede por difusão, cada nó ao receber uma transação deve enviá-la para outros nós vizinhos até que toda a rede seja atingida. Entretanto, antes do envio, o nó verifica se a transação atende um conjunto de critérios. Desta forma apenas transações válidas são propagadas na rede. As transações inválidas são descartadas na primeira validação feita pelo nó mais próximo. Os critérios são definidos conforme as necessidades de cada sistema. Após validação bem-sucedida, a transação é enviada ao pool de transações. Cada nó tem o seu pool, em outras palavras, trata-se de uma área local onde as transações permanecem até serem incluídas em um novo bloco. Enquanto isso não acontece, as transações permanecem associadas a um bloco candidato. O nó Operacional—ou minerador—tem as funções de escutar, validar e propagar transações, além de tentar construir um novo bloco, escutar novos blocos descobertos e manter uma cópia local da *Blockchain*, a lista de todos os blocos criados. Quando um novo bloco é descoberto significa o fim da construção do bloco e o início da competição para o próximo bloco. Durante as tentativas de criação do bloco, o nó Operacional coleta e armazena centenas ou milhares de transações em seu pool. No momento em que o bloco é recebido, o nó Operacional remove as transações, confirmadas nesse bloco, do pool de transações. As demais transações não confirmadas pelo bloco N se mantêm no pool para serem incluídas no bloco que pode deixar de ser um bloco candidato, se o nó Operacional encontrar uma solução para o algoritmo da prova de trabalho (DWORK e NAOR, 1992).

A disponibilidade do sistema seria assegurada por meio de um mecanismo de replicação do *Ledger* em vários nós Operacionais em uma rede *Peer-to-Peer* (NAKAMOTO, 2008). A replicação também é importante para evitar fraudes nos traços

registrados, a Figura 2 mostra que cada agente A pode manter uma versão dos dados descentralizados atualizados em sua posse. Para cada ativo registrado no *Ledger* pode-se incluir: direitos de propriedade, acesso remoto e negociação, bem como requisitos legais. Deve-se frisar aqui que o foco ilustrado neste figura é que cada nó—ou aplicativo—tem uma cópia atualizada do conjunto de transações realizadas em cada nó da rede.

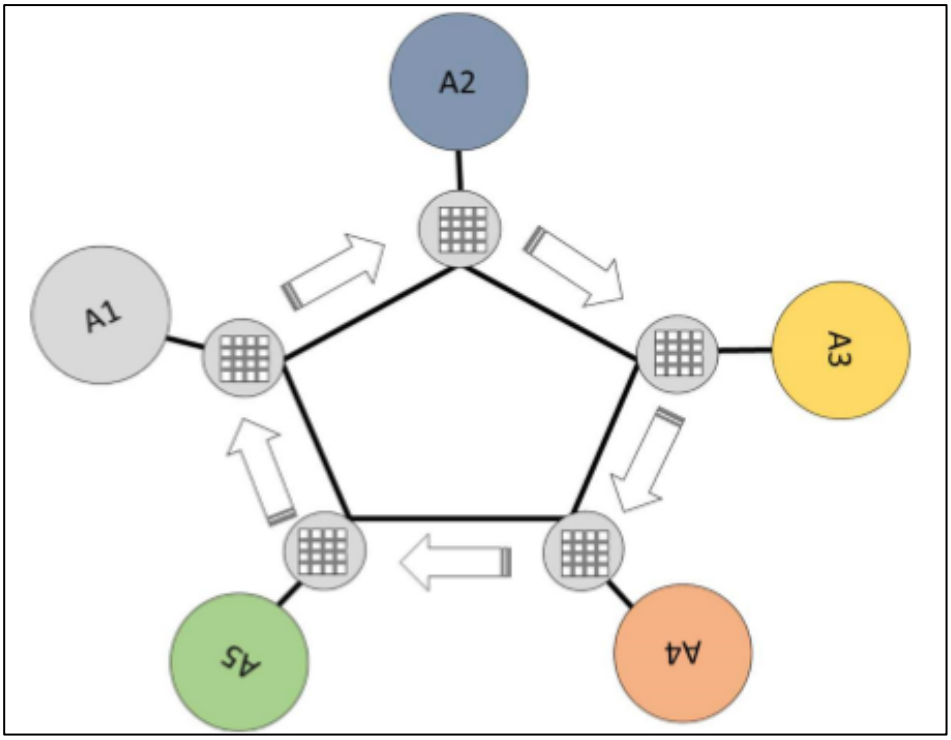

Figura 2. Interação entre aplicativos com a sincronização e a replicação do registro dos traços das transações em uma rede Peer-to-Peer.

Ao armazenar os dados e o controle na rede, a abordagem *Blockchain* elimina os riscos ligados aos dados, quando os mesmos são mantidos de forma centralizada. As cadeias descentralizadas de blocos podem usar a troca de mensagens *ad-hoc* e distribuída em rede. Sua rede não possui pontos centralizados de vulnerabilidade que *hackers* de computadores podem explorar ou qualquer ponto central de falha. Os métodos de segurança usados incluem o uso da criptografia de chave pública. Uma chave pública é uma sequência longa de caracteres e de números aleatórios; ela é um endereço na *Blockchain*. Uma chave privada é uma senha que dá ao seu proprietário acesso aos seus ativos digitais. Os dados armazenados no *Ledger* são geralmente considerados incorruptíveis.

Cada nó Operacional em um sistema descentralizado tem uma cópia da cadeia de blocos. A qualidade dos dados é mantida por meio de uma maciça replicação de base de dados e confiança computacional. Nenhuma cópia oficial centralizada existe e nenhum usuário é confiável mais do que qualquer outro. As transações são transmitidas na rede usando *software*. Cada mensagem é veicula/entregue numa base de melhor esforço. Cada nó Operacional valida as transações, adicionam-nas ao bloco que está criando e depois transmite o bloco concluído a outros nós. Cada Blockchain usa um esquema de *time-stamping* como prova de trabalho. O crescimento de uma cadeia descentralizada de blocos está sujeito ao risco de centralização do nó, porque os recursos computacionais necessários para operar com grandes volumes de dados se tornam caros.

De forma resumida, o fluxo de trabalho seguro com uma *Blockchain* inclui: (a) a criação de transação assinada digitalmente ts; (b) o envio da transação ts ao nó Operacional que a valida tsv; (c) a difusão da transação tsv para todos os nós conectados em rede; (d) a aceitação da transação tsvn na rede—se os dados forem válidos; e (e) a receptação da transação tsv pelo destinatário.

2.1.1 Smart Contracts

O *smart contract* é um contrato digital auto executável. Serve para formalizar um acordo entre as partes envolvidas utilizando tecnologia *Blockchain*.

Em [SZABO, 1996], utilizou uma cadeia de blocos em conjunto com a tecnologia de chaves privadas e públicas para ser usado em *smart contract*, também chamados de contratos de auto execução, contratos de *Blockchain* ou contratos digitais. Mostrando assim a usabilidade de um *smart contract* dentro de uma rede *Blockchain.* Funciona de tal forma que os contratos podem ser convertidos em código de computador, armazenados e replicados no sistema e supervisionados pela *Blockchain*. Isso também resultaria em *feedback* da razão da transação, como transferência de dinheiro e recebimento do produto ou serviço.

Todo *smart contract* é executado de forma automática quando duas pessoas realizam uma transação, não é necessário que ambas as partes realizem algum

processo para validá-lo. O sistema funciona com base na premissa *If-then*, a Figura 3 mostra as principais funções de um *smart contract*. Cada contrato encerra um *token* contendo um evento que transfere um montante de um emissor para um destinatário. A implementação deste procedimento é dada pela função *sendCoin* da porção de código da Figura 3.

```
contract token {
    mapping (address => uint) public coinBalanceOf;
    event CoinTransfer(address sender, address receiver, uint amount);

  /* Initializes contract with initial supply tokens to the creator of the contract */
  function token(uint supply) {
        if (supply == 0) supply = 10000;
        coinBalanceOf[msg.sender] = supply;
    }

  /* Very simple trade function */
    function sendCoin(address receiver, uint amount) returns(bool sufficient) {
        if (coinBalanceOf[msg.sender] < amount) return false;
        coinBalanceOf[msg.sender] -= amount;
        coinBalanceOf[receiver] += amount;
        CoinTransfer(msg.sender, receiver, amount);
        return true;
    }
}
```

Figura 3. Modelo de *Smart Contract*.

O resultado da transação é testemunhado por centenas de pessoas, garantindo o sucesso da negociação. Os passos para a concretização de uma transação com a utilização de um *smart contract* é ilustrado na Figura 4, em que dois indivíduos realizam uma negociação. O contrato é executado, validado e permanece disponível para todos os indivíduos da rede visualizar.

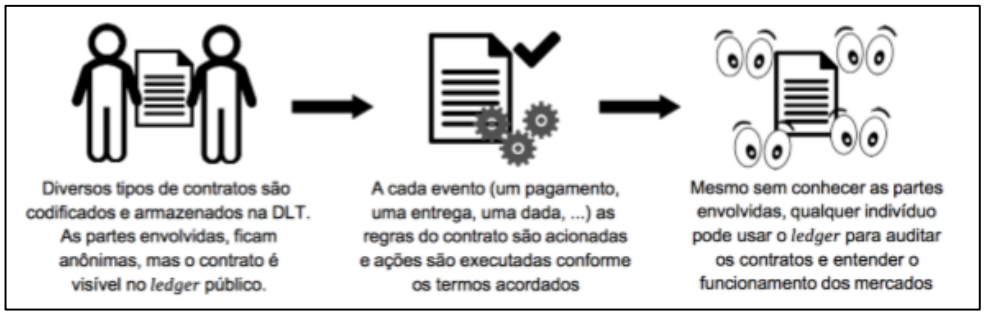

Figura 4. Funcionamento dos *smart contracts*. Na primeira figura a transação realizada pelos agentes, na segunda figura o registro e propagação do *smart contract*, e na última figura o *smart contract* já propagado no *Ledger* [Silva, 2017].

Exemplo: Locação de Imóvel

Na locação de um imóvel, o pagamento do aluguel é realizado com moeda virtual. O locatário tem seu recibo do pagamento gerado e registrado por meio de um *smart contract*. O locador envia uma chave virtual para um período de tempo limitado. Se a chave enviada pelo locador não chegar a tempo para o locatário, o mesmo será reembolsado. Se o locatário devolver a chave em data posterior ao acordado ele pagará multa. As regras são registradas e executadas automaticamente na forma de um *smart contract* com a premissa *if-else*, e ingressado na cadeia de blocos, onde são visíveis por milhares de pessoas por meio da replicação da *blockchain*, garantindo para ambos os lados confiança da negociação.

2.2 Arquiteturas de *Blockchain*

O interesse pela tecnologia *Blockchain* tem inspirado diversos modelos [Franciscon et al, 2019]. Novas arquiteturas para consenso de rede, tráfego de dados ou até mesmo gerenciamento de *Ledger* estão em foco e crescente com o passar do tempo. Um bom exemplo é o *smart contract.* Ele é um contrato digital capaz de garantir o acordo executado entre as partes. Outro exemplo é o *Hyperledger,* que é uma extensão da ideia/modelo onde se administra e armazena mais dados dentro da cadeia de blocos. As novas plataformas de desenvolvimento privado ou público também são fortes impulsionadoras da inovação da tecnologia *Blockchain.*

A *Blockchain,* desde seu início, tem apresentado evoluções constantes. O surgimento da rede *Blockchain Ethereum* é uma evolução na forma de apresentar as cadeias de blocos de modo privado ou híbrido. Isto é importante para permitir a eliminação das provas de trabalho na tarefa de criar consenso de rede. Assim, foi possível a construção de soluções em outros formatos para resolver problemas que envolvem segurança, praticidade e integridade de dados e de sistemas. Na Figura 5 pode-se observar as características da *Blockchain* identificadas até então.

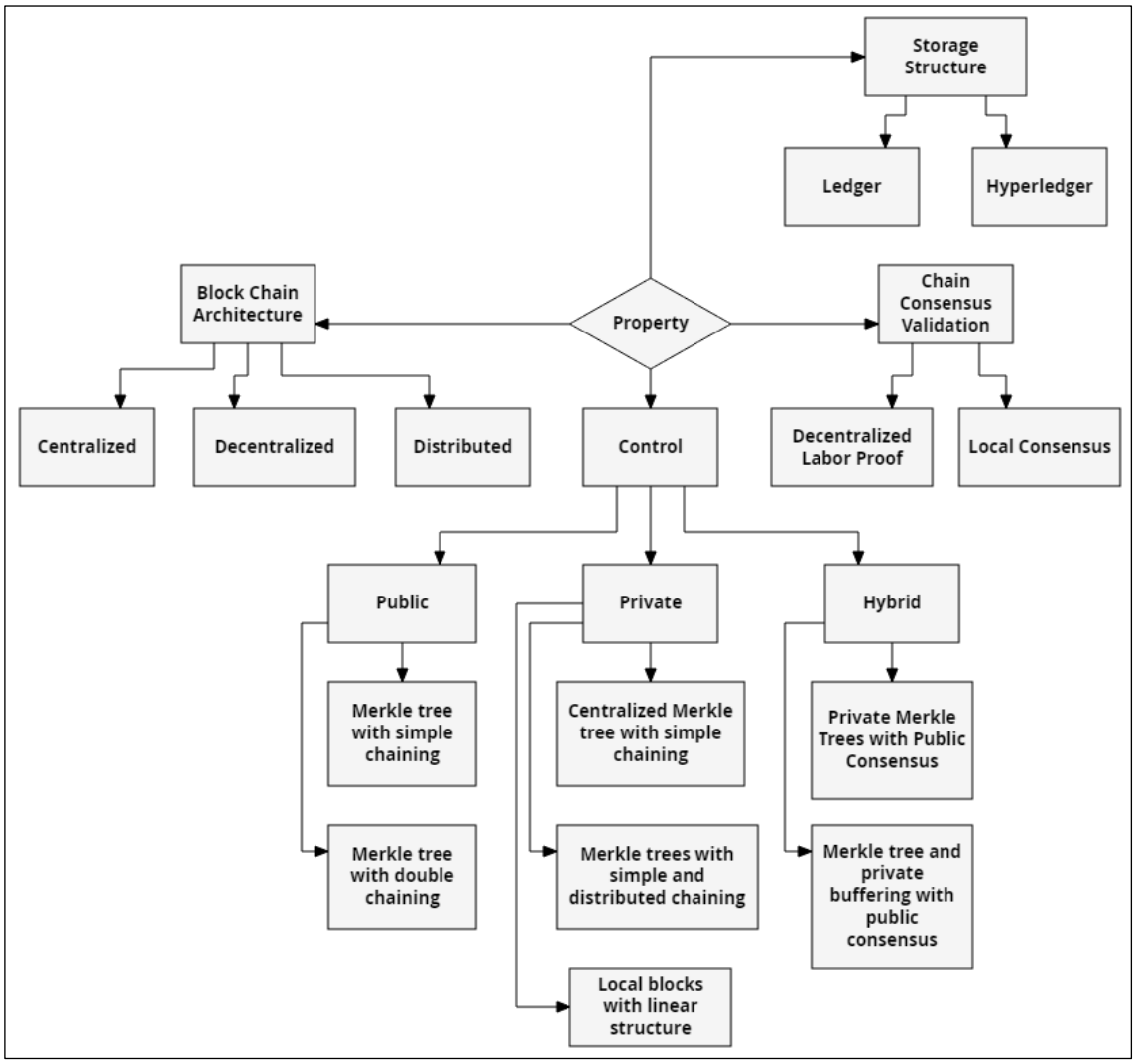

Figura 5.Propriedades do *Blockchain*.

Na sequência, cada uma destas características é apresentada em mais detalhes.

2.2.1 Cadeia de blocos

As propriedades da *Blockchain,* cf. Figura 5, permite-nos perceber que a cadeia de blocos pode ser estruturada de três maneiras. ***De forma centralizada***, quando a cadeia de blocos permanece armazenada em modo privado, em um conjunto limitado de servidores; aqui os dados são armazenados sem o consenso público de uma rede e se há um *Ledger* envolvido, ele também é centralizado, i.e., não compartilhado com nenhum outro agente. ***De maneira distribuída***, quando os dados são armazenados em vários servidores, mas, há um consenso entre tais servidores. Os *Ledgers* podem ser compartilhados entre si de acordo com o desejo dos agentes, a utilização de um *smart contract*, seguido do registro na *Blockchain* privada que fornece a perspectiva de

escolha do agente de compartilhamento ou não, i.e., os *Ledgers* podem ser disponibilizados livremente ou seletivamente. O consenso pode ser trabalhado de diversas maneiras, a exemplo do uso de *smart contracts* ou prova de trabalho. Este tipo de rede é utilizado em estruturas privadas ou híbridas de *Blockchain*. Por fim, tem-se a cadeia ***descentralizada*** de blocos—i.e., uma *Blockchain* do tipo pública—, quando todos os nós que participam estão interconectados entre si e o consenso é estabelecido de forma pública. O *Ledger* é único e replicado na rede (e.g., Bitcoin). Todos as partes envolvidas são visíveis por todos ou outros agentes com consenso público.

A estrutura de *Blockchain* híbrida, como apresentada por Sharma e Park (2017), mostra duas cadeias de blocos, uma cadeia de blocos centralizada, ou seja, que é independente, porém pode se comunicar com outras redes, é denominada rede doméstica e outra cadeia chamada de rede pública. Este conceito é apresentado, teoricamente, como se fosse interligar *Smart Cities* por meio de cadeias domésticas e *Ledgers* centralizados, juntamente com outra cadeia pública descentralizada com *Ledger* global alcançando o consenso de forma pública baseada na prova de trabalho com toda a rede.

Em Roehrs (2017) é exibido um modelo de *Blockchain* privado aplicado em sistemas de prontuários médicos, por meio de cadeias de blocos centralizadas, que comunicam com vários outros cadeias centralizadas e formam uma estrutura de cadeia de blocos privada e distribuída. Ainda sobre prontuários médicos, Guo (2018) mostra um esquema que encerra armazenamentos centralizados, dentro uma rede privada para a verificação da autenticidade das informações por meio de consenso local. Dagher et al (2018) e Xia et al (2017) criaram cadeias de blocos centralizadas para prontuários médicos com o uso de *smart contracts* para validar transações e obter consenso da rede privada de forma local. Dagher utilizou a plataforma Ethereum para seus experimentos e Xia montou uma arquitetura de *Blockchain* em nuvem.

Yue et al (2016) utilizaram a arquitetura em nuvem para trabalhar com prontuários médicos imutáveis. E um ambiente distribuído com um software contendo o histórico de cada paciente cujo controle ocorre de forma centralizada. Por sua vez, quando são realizadas as transações, elas permanecem armazenadas na nuvem em um

ambiente descentralizado. Em [Hussein et al, 2018] é mostrado uma cadeia de blocos privada com sistema de *hash* MD5, utilizando *Ledger* privado e consenso descentralizado para armazenar dados, com um mecanismo de verificação local. A cadeia de blocos funciona sobre um sistema de prontuários médicos, onde os pacientes podem compartilhar ou não seus dados.

Castaldo e Cinque (2018) mostraram um ambiente descentralizado para prontuários médicos, onde se armazena os logs dos prontuários dentro do *Blockchain* para garantir a integridade das informações. Shae e Tsai (2017), da mesma forma, criaram um *Blockchain* em nuvem para reunir e garantir a integridade de todas as pesquisas de saúde clínica; com a utilização de *smart contracts* para formalizar a transação, garantir a integridade e estabelecer o consenso local.

Jaffe (2017), por meio de uma *Blockchain* privada e distribuída, mostrou um modelo onde as pessoas utilizam bicicletas equipadas com um pequeno computador acoplado, e podem registrar a quilometragem feita dentro da cadeia de blocos privada e assim ser recompensadas por isso com moedas digitais por meio de *wallets* cadastradas na plataforma pagante. O computador acoplado a bicicleta estabelece consenso com a rede por meio de *smart contracts*. Em Engelenburg (2017), mostrou-se por meio de uma aplicação distribuída com *Blockchain* híbrida, uma maneira de coletar dados de empresas de segurança a partir de seus livros-razão—ou *ledgers*—para análise e fortalecimento de segurança pública em uma *Blockchain* pública.

Bistarelli (2017) apresentou uma arquitetura descentralizada com cadeias de blocos para sistema eleitoral. Os votos são registrados dentro da cadeia de blocos da *Bitcoin*. Conforme a votação ocorre, os votos são registrados e os blocos também são formados, formando assim, uma arquitetura descentralizada com *ledger* global para registro seguro de votos de um sistema eleitoral.

A diferença entre as três estruturas de cadeias de blocos apresentadas está na centralização da cadeia, na medida que a rede é construída sem um consenso dos participantes e para assegurar a integridade das informações, usa-se mecanismos de consenso local. Na cadeia de blocos distribuída, as partes devem ter a relaxação de restrições da rede para participar do consenso dentro da própria cadeia de blocos

privada e distribuída, os *ledgers* são trocados entre si com o aceite das partes responsáveis. E na *Blockchain* descentralizada, que é o formato público de uma *Blockchain*, se obtém o consenso por meio de prova de trabalho. O *Ledger* é global e visível para todos os participantes da rede.

2.2.2 Armazenamento

O armazenamento de dados pode ocorrer por meio da estrutura de *Ledger* ou de *Hyperledger*. Os dados registrados dentro da *Blockchain* permanecem armazenados no *Ledger*, que pode ser público ou privado. Cada *Ledger* pode ser responsável pelo armazenamento de um bloco ou de toda a cadeia de blocos. Diferentemente disto, a IBM lançou em 2016, uma plataforma denominada *Hyperledger*, trabalhando uma forma particular/própria para realizar os registros dos dados, no caso em questão, de forma distribuída. Assim, ao mesmo tempo em que a IBM cria uma plataforma de desenvolvimento *Blockchain*, ela traz também um novo modelo de *Ledger*, o *Hyperledger*.

Gao et al (2018) desenvolveram uma estrutura *Blockchain* para cadeia de suprimentos utilizando *Hyperledger* para fazer o registro dos dados e a comunicação com todas as entidades envolvidas na cadeia de blocos híbrida. Cada entidade possui o seu *Ledger* que é compartilhado para dar suporte a integridade do histórico dos suprimentos que circulam na cadeia de blocos distribuída. Ahram et al (2017) desenvolveram uma *Blockchain* para prontuários médicos que armazena os registros dos pacientes e as partes envolvidas por meio de *ledgers* distribuídos na plataforma *Hyperledger*. O consenso da rede privada é alcançado por *smart contracts*.

No registro de dados de *ledgers* públicos, Goldwasser e Park (2017) mostraram que uma estrutura *Blockchain* pode ser útil para trabalhar com o sistema de leis de um país, por meio de uma estrutura de blocos e consenso público. Bore et al (2017) e Bdiwi et al (2017), criaram uma *Blockchain* privada em nuvem com *IoT* distribuídos para o setor de educação. Criou-se um ambiente seguro para trocar conteúdos de estudos ou currículos de forma descentralizada com *smart contracts* e *ledgers* distribuídos para formalizar a transação e garantir o consenso da rede.

Entende-se, dessa forma, que quando os *ledgers* estão em uma rede pública descentralizada, o mesmo é global e é responsável por toda a rede. Quando é privado, o mesmo pode participar de toda a cadeia privada, bem como de diferentes nós da rede, de forma distribuída; podendo ser compartilhada as informações dos agentes envolvidos, ou não, Franciscon et al (2019).

2.2.3 Controle

O controle da cadeia de blocos é de três formas: privado, público e híbrido. Dentro destas formas de controle, criaram-se novas arquiteturas para trabalhar com a cadeia de blocos. Na *Blockchain* pública descentralizada, Nakamoto (2008) mostrou uma estrutura de árvore de Merkle com um *Ledger* único público interligando blocos e mineradores para estabelecer prova de trabalho, alcançar consenso e manter a rede confiável. Wang et al (2018) e Shaheen et al (2017), utilizaram tal arquitetura para validar um sistema de votação com consenso por prova de trabalho.

Kaijun et al (2018), apresentam uma arquitetura de *Blockchain* para o setor de agricultura. Aqui, um modelo distribuído de agendamento e gerenciamento de recursos comerciais agrícolas públicos foi proposto. Com uma cadeia de blocos pública de recursos de negócios, onde os dados registrados são separados em duas cadeias de blocos, uma para guardar as informações dos usuários e outra para registrar as transações do mesmo, usando uma árvore de *Merkle*. Introduziram uma plataforma de serviços para a implementação comercial e de suporte técnico. Com isto, pode-se garantir credibilidade, confiança e eficiência no sistema público quanto a administração do mercado agrícola. Por meio de uma árvore de *Merkle* pública, a estrutura de dados é dividida em cadeias de blocos públicas de recursos de negócios agrícolas baseadas nas "cadeias de informações do usuário" e na "cadeia de transações".

Blockchains privadas encerram novas formas de trabalhar com os dados. Elas podem ser da mesma forma que o *Bitcoin*, mas possuem limitações por serem privadas. Pode-se citar como exemplo a aplicação de *Blockchain* na estrutura de redes de energia inteligentes [Magnani et al (2018)]. A cadeia de blocos é estruturada em

uma árvore de *Merkle* e o consenso é por prova de trabalho, com um *Ledger* global a todos os participantes da rede privada.

Analogamente, Malomo et al (2018), apresentaram uma arquitetura de *Blockchain* do tipo federação, que consiste em várias cadeias de blocos privadas. Elas comunicam entre si e formam uma rede distribuída e interligada de cadeias de blocos. Estas cadeias obedecem a uma hierarquia para se interligarem. Seus *ledgers* são públicos nas cadeias de blocos e em toda a estrutura *Blockchain*. O consenso é realizado por meio de prova de trabalho das partes responsáveis por manter a integridade da rede. Todas as cadeias de blocos utilizam árvore de Merkle e *smart contract* para colocar em prática transações entre suas cadeias de blocos. Margheri et al (2017), criaram uma *Blockchain* privada em nuvem utilizando a plataforma *Ethereum*, em ambiente distribuído, utilizando *smart contracts* para formalizar suas transações e obter consenso da rede.

Wang et al (2017), mostraram uma nova forma de *Blockchain*. Eles apresentaram um novo modelo de compartilhamento de rede e algoritmos de consenso. Eles criam uma arquitetura descentralizada. O objetivo da proposta foi criar um ambiente funcional de compartilhamento de dados com eficiência, integridade e segurança. Em vez de usar uma árvore *Merkle* para mapear várias transações, permitiu-se que apenas um bloco montasse uma transação e ligasse na anterior de forma linear. A comunicação entre os blocos é ilimitada, permitindo que cada bloco encontrasse qualquer bloco que desejasse, em que o esforço para tal limita-se apenas em seguir os links da rede de blocos de forma linear até onde necessitar.

A arquitetura híbrida de *Blockchain* é composta por duas cadeias de blocos, operando de maneira paralela. Uma cadeia de bloco é privada e ela armazena os dados particulares do agente. A outra cadeia de blocos é pública e ela tem a função de realizar o consenso das redes privadas. A cadeia pública garante que as transações realizadas nas cadeias de blocos privados sejam verídicas. Porém, não mostra o conteúdo das transações. Estes conteúdos estão nos agentes e são compartilhados se desejado pelos agentes.

Esta estrutura de *Blockchain* híbrida é apresentada por Sharma e Park (2018). Eles mostram duas cadeias de blocos, uma privada e outra pública para fazer ligação de várias entidades que fazem parte de *Smart Cities*. A coleta de dados é feita por meio de *IoT*, armazenados em seus *ledgers* privados e não são disponibilizados na rede pública. Conforme ocorrem as transações das redes privadas, as mesmas são processadas na *Blockchain* pública para garantir o consenso e registrar no *Ledger* público.

Uddin et al (2018) e Theodouli et al (2018) se utilizaram da arquitetura híbrida em nuvem para criar um ambiente de prontuários médicos. Uddin utilizou *IoT* para capturar os dados, salvar na rede privada e gerar consenso na rede pública. Theodouli utilizou *smart contracts* na cadeia de blocos privada para registrar as transações e obter consenso na rede pública. Assim como Zhang et al (2018), que se utilizaram do consenso na rede de prontuários médicos com *smart contracts*; a implementação foi feita na plataforma *Ethereum*.

Gao et al (2018) com a solução *Hyperledger* criaram uma *Blockchain* híbrida, em que é possível fazer um encadeamento duplo na cadeia de blocos. A estrutura permite que sejam alocados blocos públicos de uma cadeia para que se consiga armazenar seus dados e mais tarde fazer consenso do bloco na cadeia pública. Xu et al (2017) utilizaram da mesma arquitetura para um ambiente de cadeia de suprimentos com cadeia de blocos privada para, no momento do consenso público, disponibilizar os dados dos conteúdos que circulam na cadeia.

2.2.4 Consenso

O consenso de uma estrutura *Blockchain* depende se a cadeia de blocos é pública ou privada. Em uma *Blockchain* pública, o consenso pode ser alcançado por meio de mineradores que realizam provas de trabalho e consolidam o novo bloco dentro da cadeia de blocos. Em caso de mineração, ela pode gerar recompensas. A prova de trabalho não é necessariamente recompensada, em uma *Blockchain* privada. Pode haver prova de trabalho apenas para consolidar o consenso da rede, buscando manter a integridade da cadeia de blocos. O consenso privado também pode acontecer por meio de mecanismos locais, e.g., *smart contract*.

Sommer et al (2018) criaram uma estrutura de *Blockchain* pública para armazenar dados curriculares de alunos por meio de *smart contracts* para facilitar o intercâmbio de estudantes em diferentes universidades. O consenso da rede foi feito usando prova de trabalho de mineradores. Diallo et al (2018) também utilizaram consenso público com mineradores que executam protocolos especializados para alcançar o consenso da rede. No entanto, se utilizaram de *smart contracts* para o ingresso de dados na cadeia de blocos, no contexto de administração de um sistema governamental eletrônico.

Bartolucci et al (2018) criaram uma *Blockchain* pública para armazenar os dados de um sistema de votação eletrônica. Em seu *Ledger* público, o consenso também é realizado por um protocolo. Niya et al (2018) utilizaram a plataforma *Ethereum* para uma *Blockchain* cuja função era coletar dados de sensores sobre poluição. As transações eram realizadas com *smart contracts* e o consenso firmado por meio do protocolo de prova de trabalho chamado *Ethereum Light Client*.

Arquiteturas privadas de *Blockchain* podem realizar o processo de consenso da mesma forma que as públicas. Assim como Magnani et al (2018) fizeram para alcançar o consenso de cadeia de blocos privada, o objetivo era gerenciar uma rede de energia inteligente. O consenso da rede foi por prova de trabalho, similarmente ao que acontece na *Blockchain* pública.

Outra forma de alcançar o consenso da cadeia de blocos é por meio de transações que ofereçam segurança na autenticidade dos dados. Para alcançar isso, utiliza-se os *smart contracts*. Alansari et al (2017) utilizaram os *smart contracts* para fazer transações em um sistema federado em nuvem. Com base na plataforma *Ethereum*, cada transação é firmada por um *smart contract,* formalizando no mesmo momento o consenso da cadeia de blocos.

2.2.5 Observações dos artigos

Foi possível observar que há uma dinâmica particular nas aplicações *Blockchain*, com seus ambientes públicos, privados ou híbridos, implementando diversos modelos de sistemas, tendo como base as aplicações de *smart contracts*, consensos alternativos e

centralização ou não de *ledgers*. Na tabela 1, pode-se observar o que cada autor usou em seus trabalhos, com por exemplo, abordagem privada (PV), híbrida (HB) ou pública (PU). A arquitetura da cadeia de blocos (ACB) adotada na maioria das vezes é a distribuída (DTD), principalmente nas abordagens privada ou híbrida. Porém, na cadeia de blocos pública a arquitetura adotada foi sempre a descentralizada (DCD), devido a sua natureza de dados explícitos.

As estruturas de armazenamento (E.A), com exceção de Gao et al (2018) e Ahram et al (2017) que usaram *Hyperledger* (HPL), todas as demais utilizaram *Ledger* (LDG). O consenso (CSS) pode ocorrer de duas formas, por meio de prova de trabalho descentralizada (PTD), ou consenso local (CL). Dependendo da estrutura em que a *Blockchain* está inserida e o que está sendo registrado em seus *ledgers*.

As plataformas utilizadas, em sua maioria, derivam da solução *Hyperledger* (HPL) ou da *Ethereum* (ETH), exceto, Bistarelli et al (2017) que utilizou a plataforma *Bitcoin* (BTC). Alguns autores não citaram as plataformas utilizadas para o desenvolvimento de seus trabalhos. Outros trabalhos utilizam a estrutura do *Blockchain* dentro de *clouds* (CLD). Isto foi observado nos casos das aplicações em saúde ou nas aplicações do tipo federação. Com *IoT* foram encontrados exemplos em vários setores. Em relação ao uso de *smart contracts* (S.C), percebeu-se que o uso desta ferramenta foi muito presente nos diversos trabalhos examinados.

Os setores (STR) governamentais que se enquadraram neste artigo, Franciscon et al (2019), foram: Sistema Governamental (SG), Logística (LG), Segurança (SR), Federações (FD), Educação (ED), Energia (EG), Urbanismo (UB), Sistema Eleitoral (SE), Poluição (PL), Agricultura (AG) e Saúde (SD).

Tabela 1. Resumo dos artigos examinados.

Autor	CTL	CBA	E.A	CSS	PTF	CLD	*IoT*	S.C	STR
Sharma P. K. and Park J. H	HB	DTD	LDG	PTD	ETH	X	X		SG
Xu L. et al	HB	DTD	LDG	PTD		X			LG
Uddin. MD A. et al	HB	DTD	LDG			X	X		SD
Theodouli A. et al	HB	DTD	LDG	CL		X		X	SD
Engelenburg S. V. et al	HB	DTD	LDG	CL					SR
Zhang P. et al	HB		LDG	CL	ETH			X	SD
Gao Z. et al	HB	DTD	HPL	PTD	HPL				LG
Hussein A. F. et al	PV	DCD	LDG	CL					SD
Roehrs A. et al	PV	DTD	LDG	CL					SD
Dagher G. G. et al	PV	DTD	LDG	CL	ETH			X	SD
Margheri et al	PV	DTD	LDG	CL	ETH	X		X	FD
Alansari S. A. et al	PV	DTD	LDG	CL	ETH	X		X	FD
Ahram T. et al	PV	DTD	HPL	PTD	HPL	X		X	SD
Wang L. et al	PV	DCD	LDG	PTD					SG
Bore N. et al	PV	DTD	LDG	CL		X	X	X	ED
Magnani A. et al	PV	DTD	LDG	PTD					EG
Xia Q. et al	PV	DCD	LDG	CL		X		X	SD
Shae Z and Tsai J. J. P.	PV	DTD	LDG	CL		X		X	SD
Jaffe C. et al	PV	DTD	LDG	CL	ETH		X	X	UB
Guo R. et al	PV	DTD	LDG	CL	ETH				SD
Malomo O. O. et al	PV	DTD	LDG	PTD		X		X	FD
Castaldo L. and Cinque V.	PV	DCD	LDG	CL					SD
Bdiwi R. et al	PV	DTD	LDG			X	X	X	SD
Yue X. et al	PV	DTD	LDG	CL		X			SD
Wang B. et al	PU	DCD	LDG	CL	ETH			X	SE
Niya S. R. et al	PU	DCD	LDG	PTD	ETH		X	X	PL
Diallo N. et al	PU	DCD	LDG	PTD				X	SG
Bartolucci S. et al	PU	DCD	LDG	PTD					SE
Bistarelli S. et al	PU	DCD	LDG	PTD	BTC				SE
Kaijun L. et al	PU	DCD	LDG	PTD					AG
Goldwasser S. and Park S.	PU	DCD	LDG	PTD		X			SG
Sommer T. et al	PU	DCD	LDG	PTD	ETH			X	ED
Shaheen S. H. et al	PU	DCD	LDG	PTD			X		SE

2.2.6 Considerações sobre *Blockchain*

Observa-se que as publicações aconteceram no período de 2016 a 2018, nota-se que o assunto enquanto área de pesquisa científica é recente. E sendo emergente, e com sua característica disruptiva, é natural que ainda haja poucos pesquisadores estudando o assunto.

A partir da leitura dos artigos selecionados, percebeu-se que há diversas formas de aplicar a tecnologia *Blockchain*. Cada arquitetura implementada foi pensada com base no tipo de aplicação alvo, dando origem a inúmeras arquiteturas criadas ou reformuladas. Com isto, se percebeu que nem todos os trabalhos nomearam suas arquiteturas propostas, mostrando apenas como uma solução baseada em *Blockchain*. Ou seja, uma arquitetura *Blockchain* pode variar para cada tipo de aplicação, tendo características especificas que se encaixam nas necessidades do projeto desenvolvido, não tendo obrigatoriamente um modelo definitivo.

Para Batubara et al (2018) o uso prático da *Blockchain* em sistemas com grandes responsabilidades, como os sistemas governamentais é bastante crítico. Com isso, sugere-se cuidados na implementação de sistemas orientados a *Blockchain* para área pública, buscando nesse caso responder se realmente é útil e eficiente. Afinal, segundo Angraal et al (2017), a quantidade de dados e informações sigilosas é muita.

Xie et al (2018) mostraram que a eficiência, desempenho e segurança das transações distribuídas são os grandes desafios a serem vencidos. Zhao et al (2018) apresentaram o uso de estrutura de *smart contracts,* desempenhando um importante papel para resolver o problema de replicação e de compartilhamento de dados, como também com custo reduzido, abrindo espaço para uma computação baseada em serviços. Assim como Feng et al (2018), que apresentaram uma proposta de aplicação de *Blockchain* para armazenar dados provenientes de sensores e de equipamentos dentro de uma rede de nuvens do tipo WSN, para suportar serviços de *IoT*. Funciona com uma arquitetura de armazenamento distribuída para a plataforma de processo de segurança de dados da WSN. Preocupando-se sempre com a funcionalidade adequada e mostrando a eficiência que *ledgers* distribuídos podem aportar.

A tecnologia *Blockchain* tem a proposta de um novo modelo de trabalhar com os dados, Hou (2017). Esta solução faz com que os sistemas operem de modo mais eficiente eliminando burocracias, pois a facilidade na realização das transações efetuadas de forma segura dispensa o acompanhamento e a interferência de agentes externos com o trabalho de garantir a realização de uma operação com segurança. Por estes motivos, Heintze e Bretschneider (2000) defendem o uso de tecnologias nas organizações para melhorar o desempenho e a efetividade. Como dito por Gilad et al (2017), o uso de *Blockchain*, com novas técnicas de consenso, reduzem significativamente o tempo de realização de cada transação e requisitos de energia do computador.

As soluções apresentadas nos artigos mostram os principais benefícios do uso da *Blockchain*. Porém, não é apresentada uma reflexão das possíveis consequências que podem ocorrer. Sendo assim, é necessário ter em mente que o ambiente de *Blockchain* vem acompanhado de desafios que precisam ser observados, estudados e planejados antes da sua implantação.

2.3 Modelo de Confiança

Lopes (2006) aponta a confiança e a reputação como os dois principais conceitos para o funcionamento de um sistema de reputação. A confiança parte do ponto de vista do indivíduo, o quanto ele confia em outros indivíduos. E para que seja confiável o indivíduo, é necessário que o mesmo preste um bom serviço, com integridade e honestidade para com os outros indivíduos que negociam ou comunicam com ele. Desta forma, se gera a reputação como consequência da soma das boas transações.

Granatyr et al (2015) diz que um modelo de confiança é uma combinação de dimensões e princípios de confiança. A discussão dessas dimensões, levando em consideração alguns tipos de interação encontrados em sistemas multiagentes, como coalizão, argumentação, negociação e recomendação, são essenciais para a formulação de um modelo. Com isso, afirma-se que a reputação é definida como uma coleção de opiniões sobre um agente, dadas por outros agentes, ou a expectativa de comportamento deste agente baseada nas suas interações anteriores.

Para construir sistemas se utilizando destes conceitos, com o passar dos anos, pesquisadores desenvolveram arquiteturas específicas, levando em conta os recursos necessários para sua aplicação, os quais são definidos como um modelo de confiança e reputação. Em Granatyr et al (2015) , tem-se que o modelo de reputação é uma arquitetura desenvolvida principalmente para três propósitos: (1) extrair dados do ambiente ou de outros usuários; (2) usar esses dados para calcular a confiança; e (3) com base nos valores calculados auxiliar nos processos de tomada de decisão.

Nesta seção, é apresentada de forma resumida as dimensões dos modelos de confiança e reputação mais significativos, publicados em Granatyr et al (2015).

2.3.1 Dimensões dos Modelos de Confiança

1. **Tipo de Paradigma:** Essa dimensão está relacionada ao método usado para construir o modelo, e pode ser classificado como cognitivo, numérico ou híbrido. O paradigma cognitivo está relacionado às crenças e estados mentais que os agentes possuem. A Confiança é medida pelo grau de tais crenças e as consequências da decisão não confiar em outro agente. Este paradigma está ligado a arquiteturas cognitivas, Sabater e Sierra (2005). O paradigma numérico não usa representações cognitivas. É baseado na agregação numérica de interações passadas e apresenta um conjunto de probabilidades subjetivas de que os agentes executarão corretamente uma determinada tarefa, Sabater e Sierra (2005). Quase todas as abordagens numéricas usam métodos estatísticos para calcular a confiança. As abordagens híbridas fazem uso de ambos os paradigmas, cognitivo e numérico, por exemplo, um sistema de argumentação, que é usado para raciocinar sobre as crenças dos agentes por meio de uma equação que calcula o nível de crença em um argumento específico. Se um agente fornecer informações nas quais não confia muito, o agente não infere nenhuma conclusão a partir dessas informações, Granatyr et al (2015).
2. **Fontes de Informação:** Para calcular os valores de confiança e reputação, os agentes precisam extrair dados de outros agentes ou do ambiente onde interagem. Existem várias maneiras de fazer isso.

a) Interação direta: Ocorre quando o agente A precisa interagir diretamente com o agente B para poder avaliar a transação. Um exemplo é a compra de um produto, onde o agente precisa adquirir o produto e a transação é posterior a avaliação.
b) Observação direta: A observação direta é baseada na observação de outras interações para analisar o comportamento do agente envolvido; aqui não é necessário que haja interação direta entre o agente avaliado e avaliador.
c) Informações de testemunhas: São usadas quando os agentes não têm informações diretas e precisam perguntar a outros agentes sobre a confiança do alvo. Quando o agente A precisa interagir com o agente B e não há interação direta entre eles, o agente A pode consultar um agente C para obter a sua opinião sobre o agente B.
d) Reputação certificada: Ocorre quando o agente avaliado possui uma lista de outros agentes que podem testemunhar sobre ele. Quando o agente A avalia o agente B, B armazena localmente uma referência de A como testemunha. Se outro agente precisar interagir com as testemunhas de B, deve realizar uma consulta direta a B.
e) Informações sociológicas: É usada quando os relacionamentos entre os agentes e seus papéis na comunidade tem importância, na medida que em influenciam os relacionamentos com outros. Semelhante à interação direta, aqui deve haver muitas interações entre agentes, pois requer-se análise de rede social.
f) Preconceito: O preconceito consiste em sinais que identificam o indivíduo como membro de um grupo, como cor da pele, religião, classe social, qualificações educacionais, entre outros, Sabater e Sierra (2005). Uma vez que o ambiente tem diferentes variáveis, os recursos de avaliação podem alterar para adaptação do ambiente.
g) Regras: São normas sociais predefinidas dentro de um modelo e são semelhantes ao conceito de reputação do sistema proposto em Regret, Sabater e Sierra (2002). Eles são usados para padronizar o

comportamento dos agentes, evitando comportamentos que vão contra as regras.

3. **Suposições de Trapaça:** Essa dimensão está relacionada à habilidade dos agentes em identificar fraudes na comunicação. Sabater e Sierra (2005) definiram três níveis: 0, 1 e 2. O nível 0 está relacionado aos modelos que não consideram informações fraudulentas e não apresentam mecanismos para filtrar agentes maliciosos. O nível 1 é aplicado a modelos que consideram informações tendenciosas, como agentes que podem esconder informações, mas não mentem. E o nível 2 está relacionado a modelos que empregam mecanismos para detectar trapaceiros.
4. **Semântica de Confiança:** A semântica multidimensional trabalha de forma que seleciona um valor que mostra se o agente é ou não confiável, e transforma este valor único em uma média de vários aspectos, convertendo-o em uma medida composta, Sabater e Sierra (2002).
5. **Preferências de Confiança:** Este modelo semântico permite a representação de confiança por meio de alguns fatores relacionados a uma aplicação específica. Porém, as preferências são relacionadas à possibilidade que os agentesdevem definir pesos para cada atributo de acordo com suas necessidades, Aljazzaf et al (2010).
6. **Delegação:** Os conceitos de confiança e delegação foram introduzidos por Castelfranchi e Falcone (1998), que argumentam que a confiança é necessária na delegação de tarefas. Desta forma, cada agente tem um conjunto de recursos visíveis para os outros e um conjunto de tarefas que devem ser realizadas em cooperação. O objetivo é determinar o melhor parceiro para delegar uma tarefa com base nas preferências e capacidades dos agentes.
7. **Medida de Risco:** Lu et al. (2009), diz que é necessário usar a gestão de risco em modelos de confiança e reputação para monitorar mudanças ambientais. Também é importante calcular as medidas de risco antes da interação com outro agente, especialmente em situações onde a delegação de tarefas é necessária. Nessa dimensão, a confiança só é exigida em situações de risco quando os agentes têm algo a perder quando a confiança é violada.

8. ***Feedback* de Incentivo:** Nessa dimensão os agentes são incentivados a dar *feedback* pelo serviço prestado, o problema nesses modelos é que os agentes envolvidos não tem interesse em fornecer o *feedback* por questões de interesse, pois, quando os agentes falam a verdade, eles podem aumentar a competitividade de outros. Por outro lado, se os agentes relatarem informações falsas, estão diminuindo a competitividade e aumentando sua própria reputação, Jurca e Faltings (2003).
9. **Confiança Inicial:** Granatyr et al (2015) utiliza dois citações para explicar essa dimensão. De acordo com duPreez [2009], a confiança é composta por três fases distintas, a partir do estabelecimento inicial da confiança, sua construção, e terminando com a dissolução da confiança ao longo do tempo. Aljazzaf et al (2010) afirmam que na maioria dos estudos, é assumido que a confiança já existe. Porém, é necessário inicializar o valor dos agentes que acabaram de entrar no sistema, basicamente por dois motivos. A primeira razão é que novos agentes não podem ser prejudicados por falta de avaliações, enquanto que o segundo motivo é, os agentes que já pertencem ao meio ambiente não podem ser enganados pelos novos.
10. **Ambiente Aberto:** Ambientes abertos são sistemas onde os agentes pertencentes a grupos diferentes e podem entrar e sair do sistema a qualquer momento. Os agentes são não determinados e não se tem controle sobre o ambiente, sendo impossível controlá-los usando uma autoridade central, Huynh et al (2004). Por este motivo, confiança nesse tipo de ambiente é visto como um problema crítico.
11. **Segurança Rígida:** Segurança rígida consiste no padrão de técnicas amplamente utilizadas em sistemas de *software*, como identidade, integridade, privacidade e autenticidade. Junto com segurança rígida é possível usar segurança suave, que utilizam técnicas de confiança e reputação relacionadas à lógica do sistema, com base nos comportamentos dos agentes. Desta forma, a mesclagem das duas técnicas é possível, o que torna o ambiente de segurança rígida mais robusto, Granatyr et al (2015).

2.4 Sistemas Descentralizados

Um ambiente clássico *Blockchain* permite descentralizar com transparência nas transações e imutabilidade dos dados armazenados. Esse ambiente oferece uma nova forma de implementar agentes de *software.* As características importantes vão na direção de permitir implementar diversos serviços relacionados à usuários em uma rede e requer-se fontes seguras de dados.

A reputação é elemento chave na construção de agentes de *software*. Reputação pode ser definida como a coleta de opiniões recebidas de outros agentes [GRANATYR, 2015]. Ela é usada para construir uma visualização ou perspectiva de confiança de algum agente visando uma transação. A coleção de opiniões é um indicador excelente para a construção da confiança em uma rede de agentes. A reputação é importante para a construção de um ambiente com interações entre pessoas ou agentes confiáveis.

A autenticidade e integridade dessas informações, bem como a transparência e correção do cálculo da reputação, são requisitos cruciais para a construção de um mecanismo confiável de reputação. As abordagens que gerenciam reputação podem ser baseadas em abordagens institucionais, sociais e de segurança.

As abordagens institucionais assumem existência de um agente central, em que o mesmo é responsável por controlar, observar e administrar as ações e perfil dos agentes na rede. Porém, são vulneráveis, na medida que eles podem sofrer ataques e ter os dados adulterados, assim como colocar em risco todos os agentes que ele administra. Abordagens sociais partem do princípio que os próprios participantes são capazes de modelar e verificar a integridade de outro agente envolvido na rede, porém também está sujeito a falhas, pois conluios podem acontecer e criar-se uma falsa integridade. As abordagens baseadas em segurança se concentram apenas em garantir a integridade e a autenticidade dos dados por meio de criptografia, como por exemplo, a rede do *Bitcoin* [CALVARESI, 2018].

2.4.1 Modelo de Agente & Blockchain

Calvaresi et al (2018) utilizam a abordagem multiagente na composição de um ambiente descentralizado com a *Blockchain*. Em seu trabalho, ele implementou e testou três contribuições:

1) Um estudo sobre como e por que integrar *Blockchain* na arquitetura de sistema multiagente.
2) Uma implementação de reconciliação que integra um sistema multiagente e *Hyperledger Fabric* v1.0, usando *Jade.*
3) Uma implementação para o cálculo da reputação de um agente operando por meio de *smart contract*.

Jade é um *framework* baseado na orientação a objetos para o desenvolvimento de agentes de software, com integração à banco de dados. O *Hyperledger Fabric* é um *framework* que implementa uma estrutura de *Blockchain*. Ele facilita o desenvolvimento de aplicações modulares, disponibilizando estruturas funcionais para trabalhar com *smart contracts*, consenso e serviços de associação, *plug-and-play*.

Calvaresi et al (2018) tratam a concepção e a implementação do sistema apresentado, estabelecendo os seguintes objetivos:

1) Identificar quais funcionalidades de um sistema multiagente precisam ser substituídas, melhoradas ou estendidas;
2) Estabelecer onde, na estrutura de agente existente, o *Blockchain* pode ser integrado;
3) Testar e avaliar os efeitos de tais intervenções, discutindo as vantagens e desvantagens.

Desta forma, o trabalho foi examinado olhando partes especificas de seu desenvolvimento, iniciando pelo *design* e depois pela implementação.

2.4.2 Design

Partindo da hipótese, que onde há autenticidade de dados e que um agente precisa gerenciar alguma informação confidencial, no modelo do sistema, cada agente pode ser associado conceitualmente a um *Blockchain* privado. Calvaresi et al (2018) projetou

essa solução, usando *Jade* como plataforma agente. Além disso, foi usado o *Hyperledger Fabric* como um componente *Blockchain*.

Para garantir confiança e autenticidade em sistema multiagente, a infraestrutura de chave pública (PKI) é empregada. Nesse cenário, todo agente gera chaves privadas e públicas, e há também uma autoridade de certificação para verificar a identidade de uma entidade e criar um certificado da chave pública da entidade. Usando o par de chaves e o certificado, o agente pode ser autenticado e todas as ações do agente podem ser rastreadas. Um componente importante do *Blockchain* autorizado por meio do uso das chaves, é um serviço de associação. O serviço de associação é uma entidade que hospeda uma autoridade de certificação e gerencia as identidades de rede de todos os pares, mantém um controle baseado em uma lista de controle de acesso sobre a atividade de rede e garante que todas as transações sejam rastreáveis para um usuário registrado.

Usa-se uma única autoridade certificadora. Uma autoridade certificadora é um agente que engloba a funcionalidade de um sistema multiagente padrão, que permite que agentes comuniquem entre si e efetuem transações, as quais precisam ser oficiais, fidedignas e visíveis perante a comunidade, e conecta uma *Blockchain* e sistema multiagente, fornecendo uma interface para que os outros agentes interajam com o agente de autoridade certificadora para o gerenciamento de registro e identidade.

O facilitador adotado implementa o conceito de diretório, seguindo o padrão FIPA3 para gerenciamento de agentes. O facilitador geralmente é representado por um agente e cada comunidade tem o seu, mas também pode ser representado por mais de um agente. Ele é responsável por acompanhar qual agente oferece qual serviço e fornecer tal informação a quem solicitá-la. No projeto proposto, o agente facilitador é substituído pelo *Blockchain*, mas o conceito foi mantido e aprimorado. Então, o facilitador assume a forma de um livro razão distribuído—ou *Service Ledger*, pois é permitido que qualquer agente comunique com qualquer outro agente que disponibilize o serviço desejado. Esta escolha de projeto permite:

- Remover a possibilidade de um único ponto de falha;
- Fornecer uma solução para a harmonização do estado atual do facilitador;

- Garantir a imutabilidade e a rastreabilidade das informações. Portanto, recursos como o rastreamento da evolução dos serviços oferecidos por determinado agente ao longo do tempo pode ser facilmente introduzido.

O conceito de reputação pode estar associado a cada agente. Além disso, a solução, pode assumir:

- Um valor global—classificando a reputação geral (média) de um agente,
- Um valor específico—associado a determinado serviço, tanto como provedor quanto como cliente.

A reputação é computada fazendo uso do histórico imutável de todos os outros valores de avaliação fornecidos anteriormente. Dentro de uma interação ambos os agentes cliente e provedor são capazes de avaliar o resultado da interação. Então, com base no valor da avaliação fornecida pelo cliente e provedor para interação corrente e seu valor de reputação atual, o contrato inteligente, que está incorpora o mecanismo para calcular a reputação, é executado por todos os nós da *Blockchain* para atualizar o valor de reputação.

Na Figura 6 mostra seus principais componentes do sistema. Pode-se distinguir dois tipos de agentes caracterizados pelos componentes sistema multiagente e *Blockchain*. Um agente padrão é caracterizado pela infográfica de uma mente, executando um mecanismo de raciocínio clássico de agente. Esse último pode ser acoplado a um nó de uma rede *Hyperledger Fabric, <Blockchain, Agente>, ou simplesmente <BC,Ai>*. Cada par (*BC-Ai*) mantém dois *ledgers* independentes:

1) Um *Service Ledger* (SL) que armazena as informações sobre o serviço que um agente provedor fornece na forma de uma *tupla*: {*Agente*; Serviços; informações adicionais}
2) Um *Ledger Transacional* (TL) que armazena as informações sobre as interações que ocorreram na comunidade e a avaliação relacionada tanto ao provedor de serviços quanto ao agente cliente.

O agente de autoridade certificadora (CA-A1) é caracterizado por uma *mente*, executando também um mecanismo de raciocínio clássico de agente; e uma *interface* para interagir com o componente *Blockchain*.

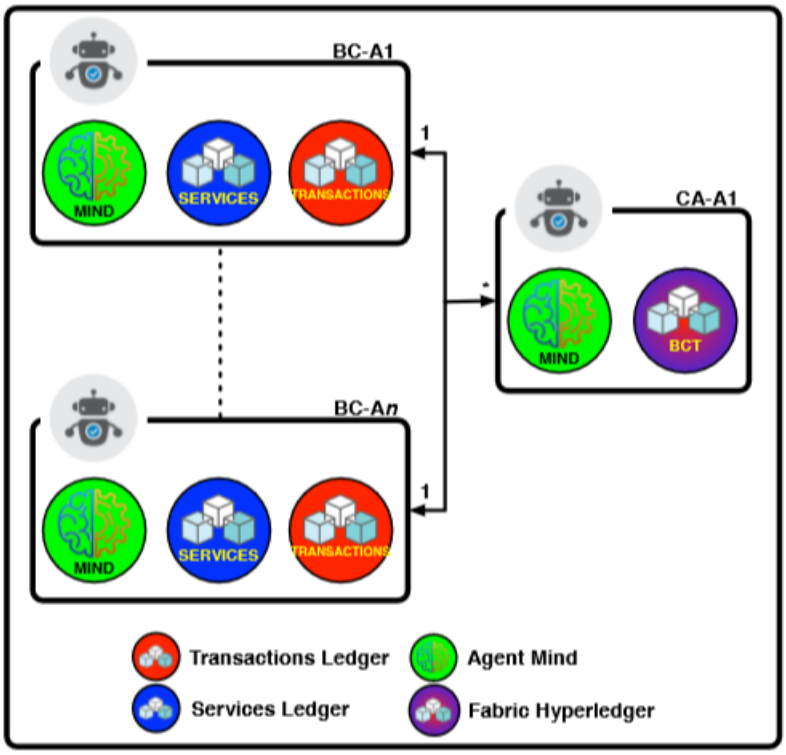

Figura 6. Design dos componentes do sistema [Calvaresi, 2018].

2.4.3 Implementação

O agente da autoridade certificadora (CA-A) (cf. Figura 7) é responsável por interagir com o servidor *Fabric Hyperledger*. Seus comportamentos realizam as seguintes tarefas:

- Encaminhar mensagens dos agentes para o *Fabric Hyperledger* e distribuir os certificados para interagir com a BC (Blockchain) por meio de uma solicitação válida. (letra “a” na Figura 7);
- Conectar-se à rede. (letra “b” na Figura 7);
- Desencadear a revogação de um certificado (por exemplo, se um agente provedor exibir o comportamento malicioso). (letra “c” na Figura 7)

O *CA-Agent* é composto de uma mente que manipula sua dinâmica de agente e um conector baseado em *node-js* (nó *java scritp*) para interagir com a rede *blockchain*.

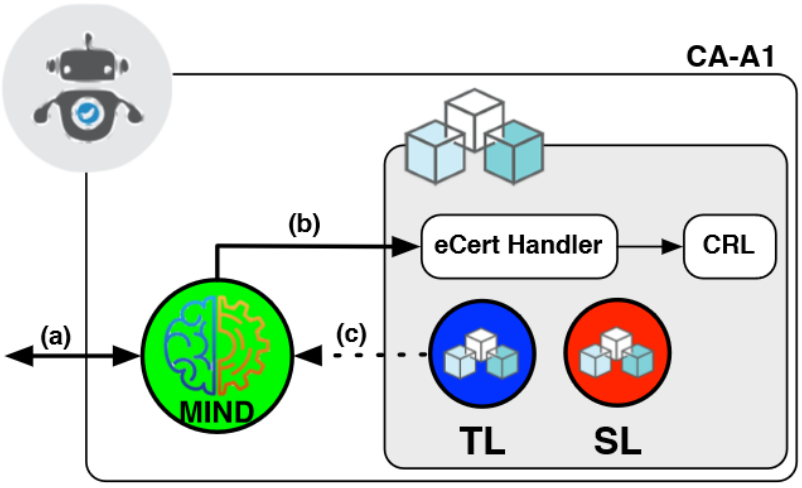

Figura 7. Registro de agente e solicitação de certificado [Calvaresi, 2018].

Os agentes BC (BC-A) (cf. Figura 6), são agentes de domínio que compõe uma comunidade de agentes e interagem entre si. Seus comportamentos realizam as seguintes tarefas:

- requisitar um certificado para o agente da autoridade certificadora (*CA-A*);
- requisitar a execução de um determinado serviço;
- ler e escrever nos *ledgers*: SL e TL;
- carregar cenário (a partir de um arquivo de configuração XML4);
- enviar/receber mensagens (cf. Figura 6).

O *Ledger* de Serviço (SL), Figura 7, contém uma lista de *tuplas*, onde cada elemento refere-se a um agente e o serviço fornecido. Todo serviço oferecido pelo agente tem um parâmetro genérico, que, dependendo do cenário, pode representar um custo ou um tempo necessário para a execução do serviço. As operações permitidas no *Ledger* de serviço (SL) são: adicionar um novo serviço, editar um serviço, remover um serviço e procurar um serviço.

Para acompanhar as transações ocorridas ao longo do tempo e permitir o cálculo da reputação de cada agente, o *Ledger* de transações possui o seguinte formato (cf. , Figura 8):

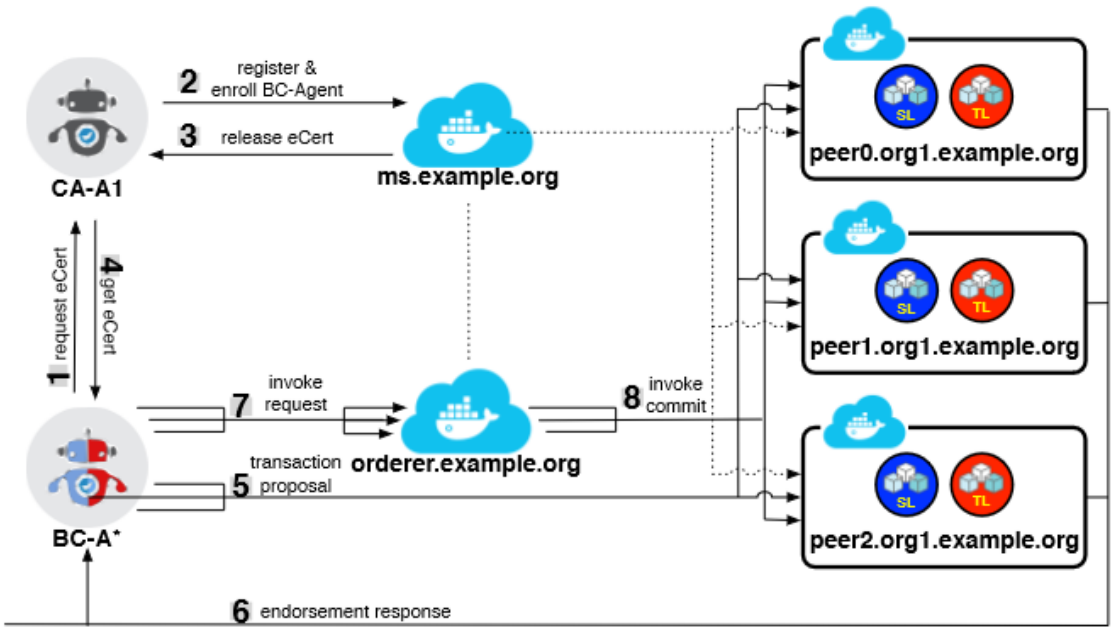

Figura 8. Arquitetura do sistema implementado [Calvaresi, 2018]

De acordo com o projeto proposto, tanto o provedor quanto o cliente precisam escrever no *Ledger* de transações (TL) na conclusão de uma transação. Isso é necessário para identificar um possível desalinhamento entre as avaliações e a percepção dos resultados. As operações permitidas no *Ledger* de transações (TL), Figura 7, são: adicionar uma nova transação, procurar uma transação (usando parâmetros diferentes como filtros), procurar um agente (usando parâmetros diferentes como filtros), e computar reputação.

A arquitetura implementada por Calvaresi *et al (2018)*, respeita o projeto apresentado na Figura 8. Os componentes do sistema são os seguintes: *n* agentes de domínio (BC-A); um único agente de autoridade certificadora (CA-A1), cuja funcionalidade pode ser distribuída entre vários agentes para eliminar o risco de um único ponto de falha; serviço de inclusão e atualização de *Ledger* na *Blockchain (ms.example.org)*; três pares de <*Blockchain, agente*> *(peeri.org1.example.com)*; e serviço de encomenda (orderer.example.com).

Para fazer parte desse sistema descrito na Figura 8, um agente de domínio (BC-A) deve estar inscrito nele. Assim, envia uma mensagem ao agente de autoridade certificadora (CA-A1), para requerer o registro na comunidade para obter credenciais e os certificados para operar no *Ledger* de serviços (SL) e *Ledger* de transações (TL) (Figura 8 : 1). O agente de autoridade certificadora (CA-A1), controla as credenciais do

requerente e, se estiver satisfeito, requer um certificado para o BCT-CA (Figura 8 : 2). Uma vez que o certificado é liberado (Figura 8 : 3), o agente de autoridade certificadora (CA-A1), envia uma mensagem contendo o eCert relacionado ao BC-A (Figura 8 : 4). Neste ponto, em relação à *Ledger* de serviços (LS), o BC-A é capaz de: publicar um serviço que está disposto a oferecer, e procurar um serviço que esteja disposto a solicitar.

Para publicar um serviço, o BC-A realiza uma transação de atualização do *Ledger* de serviços. Para procurar os serviços disponíveis, o agente realiza uma transação de consulta. Para executar uma transação, primeiro, a proposta de transação é transmitida para os pares (Figura 8 : 5). Os nós verificam se a proposta de transação está bem formada, se ela já não foi submetida no passado (proteção contra *replay-attack*), se a assinatura é válida (usando mineração), e se o BC-A está devidamente autorizado (certificado do BC-A é válido e a política de indexação de blocos foi atendida).

Os agentes respondentes as propostas usam as entradas das propostas como argumentos para a invocação de *smart contract*. O *smart contract* então é executado sobre uma base de dados para produzir o resultado da transação enviada na forma de uma resposta de endosso (Figura 8: 6). Em seguida, o BC-A confirma as assinaturas de endosso e compara as respostas da proposta para determinar, se as respostas de endosso são as mesmas enviadas anteriormente com a transação (Figura 8 : 7).

Quando o agente solicitante recebe as transações, ele as ordena cronologicamente e cria blocos de transações, que são entregues a todos os pares no canal, por meio da mineração, (cf. Figura 8 : 8), para atualizar o *Ledger*. Como assume-se ter dois *Ledgers*, as transações emitidas para cada uma delas são enviadas por meio de canais separados.

2.5 Dossiê

O sistema do *Dossiê* por Silva (2017) parte do princípio que todas as transações realizadas entre os participantes são assinadas e que cada agente tem um certificado

digital para identificação, utilizando reputação certificada por meio de fontes de informação.

Silva (2017) ilustra seu modelo com a seguinte descrição:

> *"um agente provedor p, fornece um serviço a um agente consumidor c. O serviço pode ser entendido como qualquer ação destinada a satisfazer as necessidades do solicitante, seja uma transação comercial, uma assistência médica ou uma mera pergunta a ser respondida. Na sequência, o agente c avalia o serviço e envia um feedback f para o agente p. O agente p armazena f localmente. O conjunto de feedbacks recebidos e armazenados por p, é chamado de Dossiê e denotado por $D(p)$. Esse último é utilizado como testemunho a respeito de p e pode ser consultado por outro agente que desejar aferir a confiança de p. Assim, para uma determinada interação i, um agente c avalia um agente p por meio da atribuição de um valor v— que expressa um grau de confiança—para um termo t. O termo é similar ao conceito da dimensão contexto, i.e., o termo pode ser qualquer característica a ser avaliada como, por exemplo, em transações comerciais: preço, prazo, qualidade, atendimento, dentre qualquer outro contexto necessário para o agente. Finalmente, um feedback é representado pela quíntupla $f = (c,p,i,v,t)$."*

O modelo Dossiê resolve dois problemas que são comuns em modelos de confiança aplicados em redes online abertas. São eles: a falta de interesse dos agentes em compartilhar suas experiências e, na medida em que a comunidade cresce; e o aumento do número de mensagens necessárias para localizar boas testemunhas.

No modelo Dossiê, o agente avaliado guarda localmente os *feedbacks* recebidos, eliminando assim a necessidade de um outro agente testemunhar sobre o *feedback*. A vantagem dessa abordagem é que o agente consumidor não precisa usar abordagens sofisticadas para alcançar confiança a respeito da avaliação do agente provedor. Pois, os *feedbacks* já se encontram disponíveis localmente em cada agente provedor, e protegidos de adulterações, pois o Ledger garante a dignidade de todos os feedbacks armazenados, por meio da replicação e da transparência de alguma adulteração.

2.5.1 Cálculo de Confiança

Um agente pode receber um valor *v* de um provedor *p* que retrata a opinião geral sobre o agente avaliado, e também o cliente *c* pode atribuir qualificações de diferentes pesos para cada termo diferente.

Silva (2017) explica o cálculo de confiança do Dossiê por meio do seguinte exemplo:

> *"Na avaliação de c, por exemplo, o preço e a qualidade podem ser mais relevantes que o prazo de entrega. Nesse caso c atribui um peso w para o termo ti do feedback f, denotado por $w(ti)$. Do mesmo modo, c pode diferenciar feedbacks por meio de suas relações sociais sp, por exemplo, feedbacks de agentes mais próximos como amigos, familiares ou colegas de trabalho, podem ter peso social $w(sp)$ superior aos demais. O preconceito sobre p: $w(pp)$, também pode ser necessário, de modo hipotético, avaliações de agentes estrangeiros podem ser menos relevantes que avaliações nacionais. Os pesos $w(ti)$, $w(sp)$ e $w(pp)$ devem assumir valores no intervalo [0, 1]. Caso seja desconsiderado algum destes parâmetros o valor padrão é 1. A confiança do agente c em relação ao agente p é denotada por $T(c,p)$. Ela é calculada pelo próprio agente consumidor c a partir dos feedbacks contidos no dossiê D(p) do agente provedor p. A Equação 1 apresenta a cálculo de confiança de c sobre p."*

$$T(c,p) = \frac{\sum_{f_i \epsilon D(p)} \alpha(f_i).W(t_i).W(S_p).W(p_p)}{\sum_{f_i \epsilon D(p)} \alpha(f_i)} \quad (1)$$

Aqui, o cálculo da confiança envolve o fator tempo para levar o decaimento da relevância dos *feedbacks* com o passar do tempo. Essa redução é dada pelo coeficiente α. Os *feedbacks* mais recentes são mais relevantes que os mais antigos. O coeficiente α é representado pela seguinte função exponencial (Eq. 2):

$$\alpha(f_i) = e^{-\frac{\Delta_t.(v_i)}{y}} \quad (2)$$

Onde:

- Δt representa o tempo decorrido entre o momento da criação do *feedback* e o momento em que o cálculo de confiança é feito; e
- γ é o fator que determina a velocidade do decremento da função exponencial.

2.5.2 Criptografia no Dossiê

O modelo Dossiê garante, por sua natureza, que os dados são legítimos para resolver os problemas ligados aos agentes maliciosos participantes da comunidade que tentam obter vantagens indevidas. Para evitar tais problemas, usou-se a técnica de criptografia assimétrica [Merkle, 1987] que acompanham os *feedbacks* utilizados nas transações da comunidade de agentes.

Na comunidade de agentes, cada agente possui um certificado digital que o identifica perante outros agentes, garantindo a integridade das mensagens trocadas e opcionalmente a confidencialidade. Nesta comunidade virtual, o certificado digital garante a autenticidade das mensagens trocadas. A assinatura digital pode ser implementada usando criptografia para atestar a autoria de um documento eletrônico. A Figura 9 mostra o envio e o recebimento de mensagens assinadas.

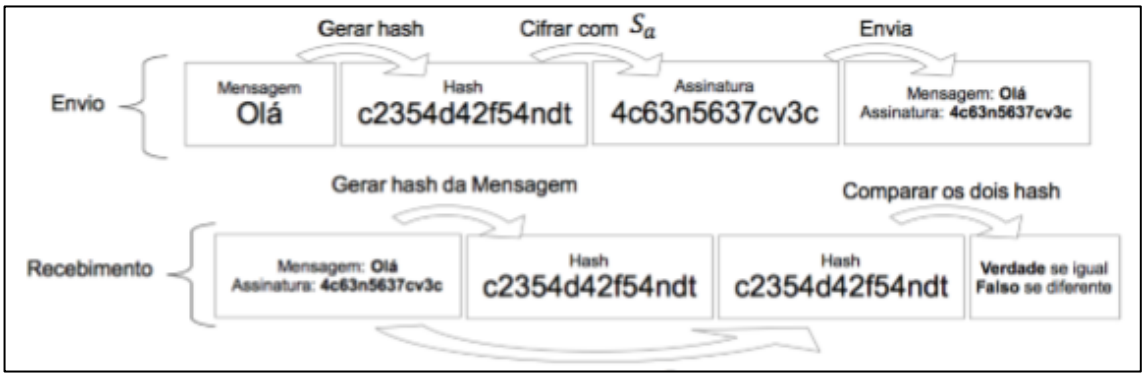

Figura 9. Verificação de mensagem por assinatura digital. FONTE [Silva, 2017]

Vanderson B. Silva (2017), descreve o seguinte cenário para explicar o funcionamento básico da sua abordagem:

> *"dado um agente a e seu par de chave: secreta Sa e pública Pa. Quando a envia um dado Da para um destinatário, a calcula a função* hash *de Da que resulta em hDa = hash(Da) e criptografa hDa com a sua chave secreta ADa = encrypt(hDa, Sa), onde ADa é chamada de assinatura digital do dado Da. Assim, a ao enviar Da, precisa enviar seu certificado digital que possui a chave pública Sa. O destinatário da mensagem pode verificar se Da foi criado por a calculando novamente o hash da mensagem recebida hDa e decifrando ADa a partir da chave pública Pa, logo se: decipher(ADa, Pa) = hDa assume-se que Da foi criado pelo agente a."*

2.5.3 Cadeia de blocos no Dossiê

O modelo Dossiê trabalha para que cada indivíduo tenha os *feedbacks* recebidos sob seu controle. O *Dossiê* é necessário para manter a integridade dos dados de cada agente, com o objetivo de proteger-se de agentes maliciosos que podem modificar os *feedbacks* recebidos afim de beneficiar-se. A assinatura digital garante a confiança e imutabilidade dos *feedbacks*.

O modelo do *Dossiê* foi implementado usando uma estrutura particular ilustrada na Figura 10. O resumo do *Dossiê* é representado pela raiz da árvore de *Merkle*. O *feedback* de origem armazena o *hash* do *feedback* incluído no Dossiê. O campo de transação anterior é responsável por interligar todas as transações que o agente já recebeu. Dessa forma, é possível percorrer a *árvore* no sentido inverso das mudanças do Dossiê e obter todos os *feedbacks* de origem. O campo identificador do agente avaliado possui o *hash* do certificado digital referente ao agente avaliado. Com este campo é possível verificar as transações de qualquer indivíduo. Se um agente resolve eliminar um *feedback* ou iniciar uma nova estrutura de Dossiê, a fraude é descoberta, pois os *feedbacks* recebidos anteriormente estão armazenados e são imutáveis dentro da *Ledger*.

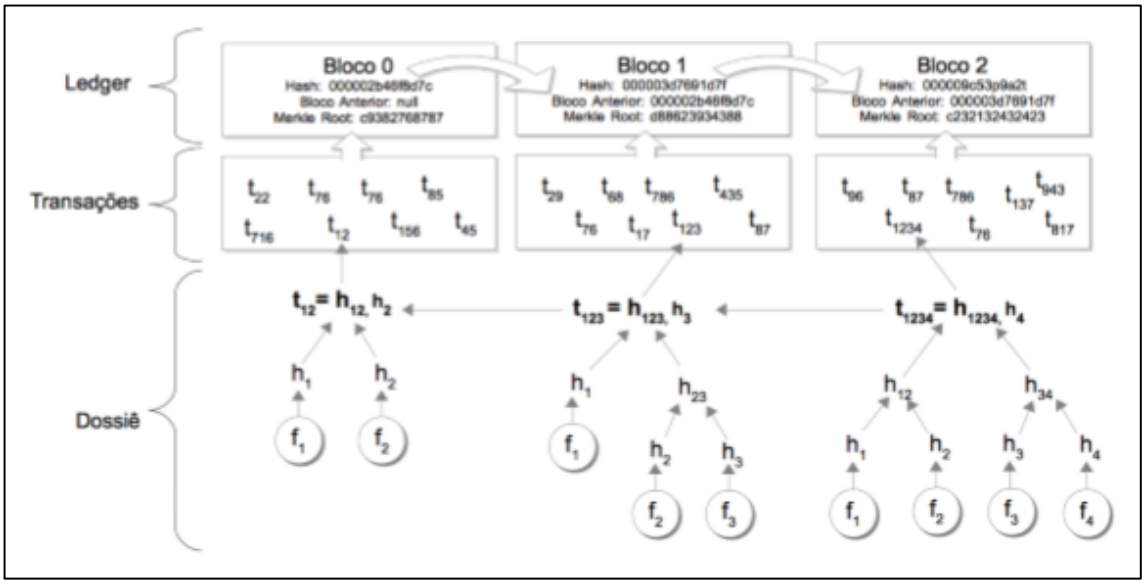

Figura 10. Dossiê integrado em uma estrutura de ledgers. FONTE [Silva, 2017]

Na Figura 11 é possível visualizar todo o processo entre um agente consumidor quando solicita um serviço para o agente provedor e o envio do *feedback* do agente

consumidor ao agente provedor, que pode abrir uma transação de denúncia se necessário, o processo é encerado quando armazenado no *Ledger*.

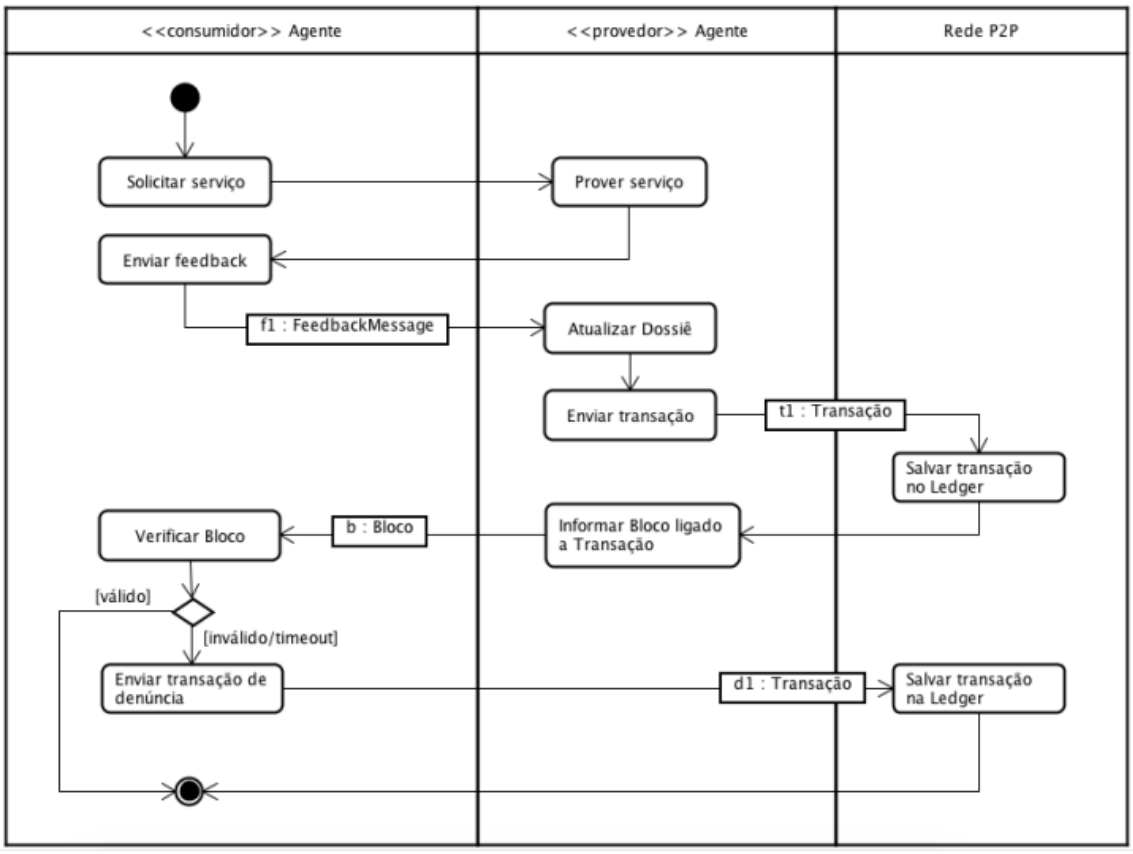

Figura 11. Envio do feedback e atualização do Dossiê no *Ledger*. **FONTE [Silva, 2017]**

Quando o *feedback* é enviado ele faz o seguinte caminho. O emissor do *feedback* é um agente consumidor *c* e o receptor é um agente provedor *p*. O agente provedor *p* repasse o certificado de registro local de *feedback* no *Ledger* global; lembrando que esse último guarda apenas o último *hash h* de registro de transação de Dossiê de cada agente. O agente consumidor *c* aguarda retorno do agente provedor *p* com a identificação do bloco que contém o *hash h* da sua transação. Dessa forma, o agente consumidor *c* tem a confirmação de que seu *feedback* se encontra no Dossiê do agente provedor *p*. Se o agente consumidor *c* não receber a confirmação, ele insere no *Ledger* uma transação de denúncia. Uma denúncia é um registro no *Ledger* que contém as identificações dos agentes provedor *p* e consumidor *c*, juntamente com o *feedback* não registrado no *Dossiê* do agente provedor *p*. Com esses dados, qualquer indivíduo da rede pode verificar se a denúncia é verdadeira, pois, o *feedback* é assinado pelos

agentes participantes da transação e sua inclusão é verificada ao percorrer a cadeia de blocos.

2.6 Modelo *TrustChain*

Em [Otte, 2017] é apresentada uma nova arquitetura de *Blockchain*, denominada *TrustChain*, capaz de criar transações confiáveis entre estranhos sem controle central. A abordagem oferece escalabilidade, abertura e resistência à ataques *Sybil* ao substituir a prova de trabalho por um mecanismo que estabelece a validade e a integridade das transações. *TrustChain* é uma estrutura de dados construída para resistir adulteração e destinada ao armazenamento de registros de transações de agentes.

TrustChain inclui um novo algoritmo resistente a *Sybil* chamado *NetFlow* para determinar a confiabilidade de cada agente em uma comunidade *online*. O *NetFlow* garante que os agentes que recebem e devolvem recursos na comunidade são legítimos. A proposta mostra que mesmo sem consenso global, os registros históricos das transações oferecem segurança e escalabilidade contínua. A experimentação mostrou que a taxa de transferência de transações do *TrustChain* supera a taxa das arquiteturas tradicionais de *Blockchain*, por exemplo *Bitcoin* [Otte, 2017].

2.6.1 Arquitetura do modelo *TrustChain*

O *TrustChain* é construído em torno da ideia de agentes que comunicam uns com os outros. As transações incluem a troca de arquivos, compra ou venda de mercadorias, transferências de dinheiro, etc. Cada transação é criptograficamente assinada por ambas as partes usando um mecanismo seguro de assinatura. Isso significa que a participação do usuário envolvido na transação é irrefutável.

Cada agente acompanha as transações em que ele está envolvido. Uma maneira de organizar esse acompanhamento é encadear as transações de maneira inviolável, ou seja, em uma cadeia de blocos. As transações são armazenadas usando uma estrutura de dados padrão *Blockchain*, em que cada bloco contém uma transação com as assinaturas de ambos agentes. Uma função de *hash* segura garante as assinaturas.

A abordagem difere das arquiteturas tradicionais de *Blockchain* no sentido de que cada participante mantém sua própria cadeia de transações. Arquiteturas como *Bitcoin* ou *Ethereum* mantêm uma cadeia única e global contendo um rastreamento de todas as transações realizadas pelos usuários. A consistência da cadeia é garantida por um sistema de consenso como prova de trabalho. Uma diferença adicional é que, em *Blockchains* tradicionais, muitas vezes as transações múltiplas são excluídas em um bloqueio para aumentar a taxa de transferência da transação, enquanto que em *TrustChain*, cada bloco descreve somente uma transação. Após a conclusão de uma transação entre dois agentes, ambas as partes assinam a transação e a insere um novo bloco na cadeia de blocos.

Embora a *Blockchain* seja uma boa estrutura para interações históricas, há uma vulnerabilidade nessa abordagem: as cadeias—mantidas por agentes—são nulas de qualquer controle, uma vez que cada cadeia local é mantida apenas por uma entidade. Os agentes que realizam transações podem decidir não anexar uma transação à sua cadeia local. A lógica por trás desse comportamento é que uma transação pode ser desfavorável para um dos participantes. Além disso, um agente pode reescrever sua cadeia local reordenando transações e recalculando ponteiros anteriores sem muito esforço computacional, corrompendo assim o histórico de suas transações, tornando o mesmo favorável a si, mesmo sendo falso.

2.7 Dossiê x *TrustChain*

Tanto o modelo Dossiê quanto o modelo *TrustChain* assume que cada agente está envolvido em uma comunidade. E cada agente provedor tem seu *Ledger*, contendo no caso do modelo Dossiê, os *feedbacks* recebidos dos outros agentes que negociaram com ele, e no *TrustChain* as transações que o mesmo realizou. A diferença técnica em ambos os modelos está na arquitetura da cadeia de blocos, i.e., na forma que se oficializa a transação, na forma que se executa o consenso da cadeia de blocos, na segurança dos dados na rede.

2.7.1 Arquitetura da cadeia de blocos

Dossiê: O modelo Dossiê funciona de forma que o *Ledger* é registrado em uma cadeia de blocos descentralizada usando uma árvore de Merkle, i.e., em uma estrutura *Blockchain* tradicional. Os agentes participantes realizam suas transações e consequentemente recebem *feedbacks*. Estes são registrados no histórico de cada agente, tornando-se da perspectiva de controle, um histórico local.

TrustChain: O modelo TrustChain parte do princípio que cada bloco armazena apenas uma transação, sendo essa registrada diretamente na cadeia de blocos. Por sua vez, a cadeia de blocos é isenta de qualquer controle vindo por parte de algum mecanismo de consenso, pois é mantida por apenas uma entidade, no caso o agente. Porém, o modelo *TrustChain* interliga cada bloco a duas conexões *hash*, i.e., a rede expande não somente em uma direção, do nó pai para o nó filho. A cadeia de blocos se constrói de forma que na primeira ligação *hash* liga-se linearmente todas as transações realizadas pelo agente detentor dessas transações, no caso o agente *A*, a segunda ligação *hash* liga-se com as transações realizadas pelo agente *B*, que da mesma forma que o agente *A*, possui sua própria cadeia de transações. Nesse sentindo, por meio da ligação com o agente *B*, o agente *A* irá herdar uma série de outras transações de outros agentes que o mesmo não tem contato, até que se realize uma transação entre esses. É desta forma que a cadeia se entrelaça e forma uma rede disforme, com padrões pré-definidos como a árvore de *Merkle*.

Isto permite que a própria rede alcance um consenso formado por apenas uma fração de toda a cadeia de blocos. Um determinado número de agentes consegue estabelecer o consenso da rede sem que haja a necessidade de envolver toda a cadeia de blocos, pois uma vez que a cadeia é modificada por algum agente malicioso, outro agente que faz parte da cadeia deste mesmo percebe a não-regularidade e pode denunciar.

2.7.2 Oficialização da transação

Dossiê: O modelo Dossiê funciona de forma que toda transação é acompanhada de um *feedback,* o qual é oficializado por meio de um *smart contract*, que é enviado para o agente provedor no momento seguinte da prestação de um serviço para um agente

consumidor. O *feedback* atualiza o *Dossiê* do agente provedor e é enviado para o *pool* de transações, onde é minerado e depois salvo na *Blockchain* pública.

TrustChain: A *TrustChain* tem como abordagem a transação única por bloco. O processo de inclusão de transação se dá quando um agente faz um negócio com outro agente e ele completa a transação. Essa última é assinada por ambas as partes usando um mecanismo seguro. Assim, uma vez efetuada a transação, um novo bloco contendo-a é gerado e indexado na cadeia de blocos.

2.7.3 Consenso da cadeia de blocos e segurança dos dados

Dossiê: As transações são efetivadas por *smart contract* na cadeia de blocos, iniciando pela validação das transações armazenadas no *pool* de transações, seguindo para o agrupamento das mesmas em um bloco, o qual é validado por meio de um cálculo matemático e terminando o ciclo com a seleção da cadeia de blocos mais longa.

A segurança das transações é garantida por meio do consenso global do *Ledger*, onde o mesmo é transparente para visualização de qualquer indivíduo. No que diz respeito a alteração de blocos na *Blokchain*, ela tem garantia autenticidade dos dados por meio da replicação da cadeia principal de blocos, que é protegida. Logo, se um agente malicioso alterar algum bloco, o mesmo terá que replicá-la para toda a *cadeia de blocos* novamente, levando em consideração que ainda estão sendo criados novos blocos, o que torna inviável um ataque à *Blockchain*.

TrustChain: No modelo de cadeia de blocos com duplas ligações de *hash* entre os blocos, o consenso se estabelece na forma de um mecanismo de “fofoca”, onde é possível dizer que, em um cenário que engloba toda a cadeia de blocos da *TrustChain*, há diversos consensos que dizem respeito às ramificações das transações dos agentes. Essa ramificação é formada com ligações duplas entre os blocos; uma das ligações é responsável por alinhar as transações de um agente *A* em particular e a outra por montar a cadeia de transações ligando-se com os outros agentes com que o agente *A* negociou, herdando assim uma fração do *Ledger* dos *n* outros agentes.

O mecanismo de “fofoca” entra em operação quando um agente percebe na sua rede um bloco alterado ou indexado indevidamente. A percepção se dá pela

quebra da igualdade da sua estrutura com a estrutura dos outros agentes da ramificação da rede; denuncia-se a quebra de estrutura aos demais agentes que participam da ramificação. É importante perceber que a *TrustChain* assegura o conteúdo de cada transação por meio de mecanismo de criptografia. Porém, um novo bloco falso, por exemplo, não se tem mecanismo para combater tal presença. A abordagem assume que cada agente da rede cuida e denuncie irregularidade na rede, para que seja corrigida.

2.8 Considerações do Capítulo

Inicialmente, nesse Capítulo foi apresentado a tecnologia *Blockchain* e suas características, bem como o funcionamento de sua arquitetura por meio de uma cadeia de blocos baseada na árvore de *Merkle* e a comunicação criptografada entre as partes por meio de *hash.* O funcionamento da agregação de valor da cadeia usando o processo de mineração com consentimento global para formação de consenso dos participantes, e finalmente, a atualização de toda a cadeia para todos os integrantes dela.

O trabalho de Calvaresi et al, (2018) permite pensar na junção de agentes de *software* executando em um ambiente descentralizado com a tecnologia *Blockchain*. O modelo de confiança *Dossiê*, criado por Silva (2017), também mostra uma cadeia de blocos local capaz de acomodar um *Ledger* local, pois, com um sistema sofisticado de avaliações, o mesmo permite assegurar a integridade do *Dossiê* de cada agente. Finalmente, o modelo *TrustChain* [Otte, 2017] mostra uma arquitetura de que se utiliza de um mecanismo de “fofoca” para substituir indiretamente o consenso por prova de trabalho.

3. ARQUITETURA

Tendo em mente um ambiente onde indivíduos podem comunicar entre si e fazer transações, a segurança nas transações e a imutabilidade de dados é altamente relevante, e ela precisa ser transparente entre participantes. A imutabilidade é relevante, não apenas por garantir o correto registro de uma transação, no caso dos agentes, é também relevante que cada *feedback* seja imutável e armazenado localmente; a partir dos *feedbacks*, métricas de confiança podem ser geradas sobre uma base histórica. Nestes termos, o modelo Dossiê fornece uma estrutura bastante apropriada para o nosso objetivo, armazenamento de dados e *feedbacks* localmente sobre o controle de cada agente avaliado.

A confiança dentro de um mecanismo transacional em rede requer um ambiente que permita garantir tecnicamente o registro persistente e imutável de forma facilitada e segura. Então, no decorrer deste Capítulo será apresentado um ambiente descentralizado usando *Blockchain,* cuja ideia é integrá-lo ao modelo de confiança Dossiê (SILVA, 2017). Deve-se destacar que o modelo Dossiê foi criado no laboratório de Agentes de *Software* do PPGIa/PUCPR.

Nossa premissa é que os mecanismos estudados nos permitem propor um ambiente computacional completo, onde os indivíduos de uma sociedade de agentes podem fazer negociações confiáveis entre si, sem a necessidade de um “cartório” que autentique todo o processo. Desta forma, serão apresentados a seguir os detalhes técnicos de como funciona uma transação fidedigna sem o auxílio de “cartório” para oficialização/registro de processo/contrato e sua execução. É importante perceber que neste modelo de negócio a reputação de um indivíduo é importante, pois, ela é negociada diretamente com o mesmo, sem uma entidade centralizadora para garantir o negócio realizado entre ambas as partes.

3.1 Mecanismo da Sociedade

É entendido que o Dossiê é um *Ledger* local de um indivíduo/agente que deseja fazer transações e proteger sua reputação. Da mesma forma, quando solicitar um serviço de outro agente, ele possa obter informações confiáveis para calcular níveis apropriados de confiança *vis-à-vis* a transação alvo.

Quando um agente *c* deseja negociar com outro agente *p*, o agente *c* usa o Dossiê do agente *p* como fonte segura de informação/histórica de transações, e com essa fonte deve ser capaz de gerar uma percepção se o agente *p* confiável ou não. O Dossiê, teoricamente, é uma estrutura confiável, segura e eficiente para ser usada como sistema de registro local de dados imutável descentralizado. A nossa proposta é transformar o Dossiê em *framework* que permita facilmente especializá-lo/estendê-lo para projetar diferentes aplicação em diversos domínios, mantendo suas características originais. A primeira investida consiste em dotar o modelo *Dossiê* de um conjunto de facilidades programadas sobre uma plataforma agente e *Blockchain,* que represente o estado da arte em tecnologia. O objetivo não é gerar um produto final/comercial, mas gerar um conjunto de experimentos e avaliações em termos de codificação descentralizada, segura, imutável permitindo ilustrar diferentes cenários de aplicações.

Nesta mesma linha, a operacionalização de cada transação deve ser feita usando *smart contract*. As facilidades aqui devem acontecer em termos de *templates* que podem ser especializadas para diferentes tipos de transações/objetivos. Assim, o conceito de *smart contract* deve permear a realização de toda transação, em particular, para que a mesma seja realizada sem que necessite de um terceiro agente garantidor de transação. Para que toda transação seja validada para toda a comunidade, a mesma deverá ser anexada a uma *Blockchain* pública, onde o consenso é global e seguro por sua natureza descentralizada.

Deve-se garantir que o *Dossiê* de cada agente provedor *p* seja sempre atualizado, caso um *feedback* seja retornado pelo agente consumidor *c* que o contratou. A atualização de um *Dossiê* resulta na anexação da nova transação no histórico do agente *p*; tal atualização poderá gerar novos valores à sua reputação. Caso

um agente consumidor *c* não receba o serviço contratado, ele pode enviar uma transação de denúncia para a rede *Blockchain* pública. Desta forma, se o agente provedor *p* agir de má fé, então será dada a oportunidade aos demais agentes de conhecer o fato. Tal denúncia é registrada na cadeia de blocos e poderá ser exibida para todos os agentes que desejarem avaliar a informação.

Assume-se então que a *Blockchain* por meio de cada *Dossiê local*, terá a função de garantir fidelidade da informação que trafega na rede. O Dossiê faz parte da rede e ele é uma fonte de informação segura que cada agente pode usar para inferir a sua reputação e permitir que a sua reputação seja também inferida.

3.2 Transações para Ambiente Descentralizado

A tecnologia *Blockchain,* dada a sua natureza, possibilita que a cadeia de blocos seja replicada integralmente para todos os agentes que utilizam a *Blockchain*, e garante a transparência necessária para a visualização de quaisquer transações realizadas dentro do domínio da rede. A imutabilidade de dados é assegurada por meio do uso de *hash* criptográfico, impedindo que os dados sejam reformulados e transmitidos para toda a rede. Tal imutabilidade é fundamental em nossa proposta, pois no ambiente projetado somente agentes—assumindo os seguintes papéis: cliente/solicitante ou provedor—poderão operar na sociedade/comunidade. Na Figura 12 é detalhada a forma como o *feedback* é registrado na *Blockchain* e após isso replicado para toda a cadeia de usuários. Tornando então seguro e eficiente o uso da abordagem.

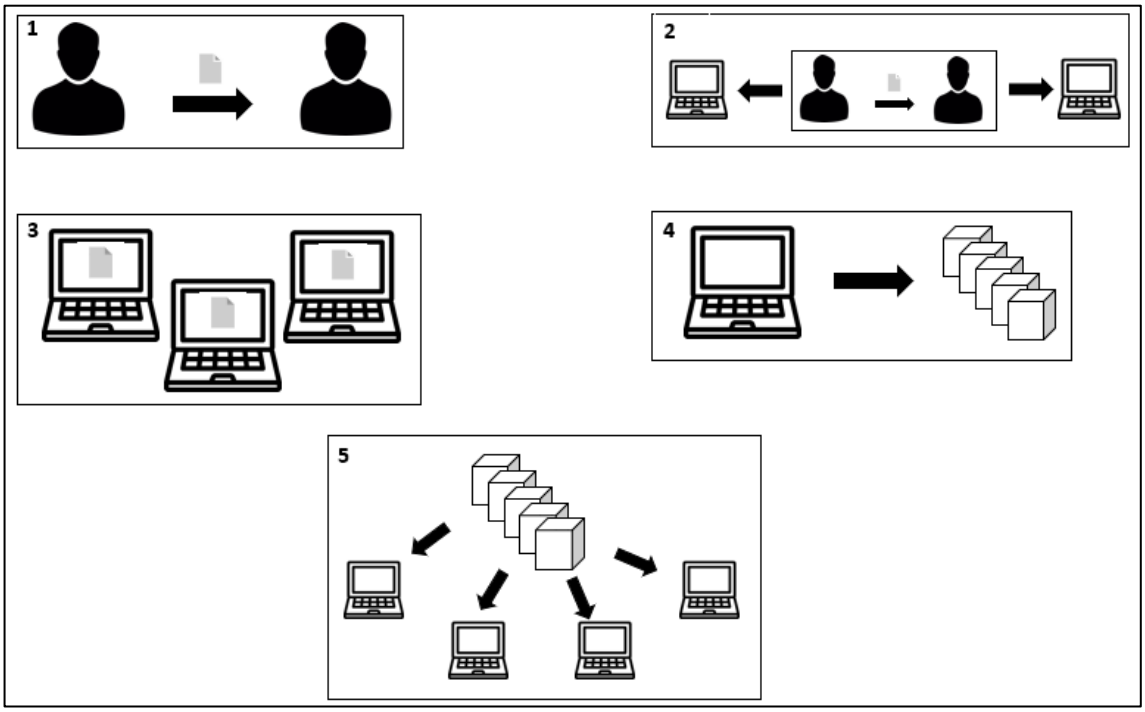

Figura 12. Replicação do Dossiê por meio da *Blockchain*. Na situação 1, o envio do *feedback* à um agente. Na situação 2, a ilustração dos agentes com seus *feedbacks* localmente. Na situação 3. a administração local de cada Dossiê. Na situação 4, o Dossiê fazendo parte da cadeia de blocos. Na situação 5, o Dossiê visível para cada integrante da *Blockchain*.

Assume-se que todo agente é capaz de negociar qualquer valor com qualquer outro indivíduo/agente. Isto deverá ocorrer sem a interferência de um outro agente garantidor de sucesso da transação, i.e., a confiança entre as partes deve emergir do ambiente de negociação ou de funcionamento da comunidade de agentes. Para isso, cada agente tem o seu *Dossiê*, que é atualizado por meio de transações de *Dossiê*. Este último é a fonte principal de dados para o cálculo de reputação. Para toda interação realizada, entre um agente provedor e um agente consumidor, um *feedback* é gerado e inserido como imutável no *Dossiê* do provedor/executor de um serviço; o incremento no histórico de agente provedor provoca uma revisão na sua reputação. Desta forma, quanto melhor o índice de reputação do agente provedor *p*, maior é a reputação de *p vis-à-vis* os demais agentes. A Figura 13 ilustra o mecanismo que apresenta o *Dossiê* implementado em uma rede *Blockchain,* onde todos os agentes se comunicam diretamente entre eles sem a necessidade de um intermediador.

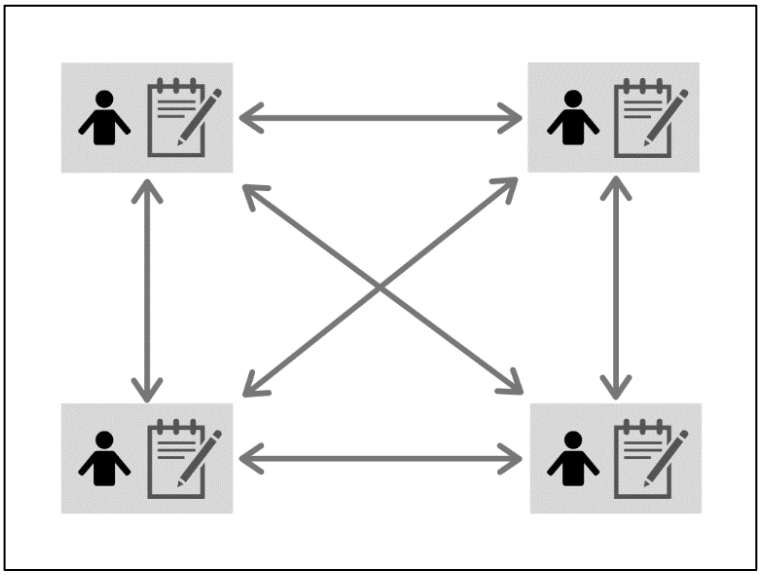

Figura 13. Cada agente tem um Dossiê em um ambiente descentralizado**.**

A rede descentralizada, própria de um ambiente *Blockchain,* coloca o *Dossiê* no centro da proposta. Cada *Dossiê* de cada agente provedor é uma fonte de dados. O *Dossiê* permite que o serviço de reputação defina individualmente a confiança de cada agente participante da rede. Pois, cada indivíduo participante da rede pode tanto solicitar quanto disponibilizar serviço.

Toda vez que um indivíduo pretende solicitar um serviço, ele seleciona outro indivíduo que mostra por meio de sua reputação as credencias para realizar uma transação. A reputação se constrói favoravelmente com o bom serviço realizado do agente. Caso o agente realize serviços ruins, sua reputação é avaliada negativamente pela sociedade.

Deve-se salientar que o *Dossiê* original não dispõe de uma estrutura para representar *feedbacks* por tipo de serviço. Nesta proposta, o Dossiê de cada agente provedor implementa uma estrutura com múltiplas pastas. Cada pasta acomoda *feedbacks* de um único tipo de serviço, i.e., cada tipo de serviço pode representar uma especialidade ou uma competência de um provedor. A justificativa pode ser dada pelo seguinte cenário: um dado agente provedor *p* pode ser muito especializado no serviço S1 e consequentemente irá obter excelentes *feedbacks* na prestação de S1. O mesmo agente provedor *p* pode também fornecer um serviço que ele é menos especializado S2 e consequentemente irá obter *feedbacks* não tão bons do que quando ele executa S1. De forma mais concreta, pode-se ilustrar esse cenário assumindo que uma dada

pessoa (*p*) é um desenvolvedor *back-end* Java com certificação e dez anos de experiência (S1). A mesma pessoa (p) fornece também serviço de culinária francesa (S2), mas ela ainda é amadora, logo a sua avaliação não deverá ser tão boa quando aquela recebida como desenvolvedor (S1). Em outras palavras, para permitir um agente provedor ser multisserviço sem penalizá-lo naquilo que ele é melhor, a abordagem *Dossiê multipasta* nos parece relevante como estrutura para compor distintas fontes de dados para gerar valores reputação de um agente provedor, onde a reputação será por serviço/competência e não agente provedor.

A concepção e desenvolvimento de uma estrutura *Dossiê multipasta* com seus protocolos de atualização e garantia da imutabilidade de *feedbacks* é o desafio do projeto.

3.3 Mecanismo de Transação

A rede deve funcionar de forma que toda transação seja realizada por meio de *smart contract* e implementada usando as facilidades do *framework* Truffle [Truffle, 2019]. A transação em si é o *feedback* ou os processos de solicitação e aceite de visualização do Dossiê. Ela faz uso de moeda digital como mecanismo de funcionamento da rede.

Essa moeda digital é gerada pela própria aplicação, i.e., não pertence a nenhum *Blockchain* especifico, como *Bitcoin* ou *Ether*. A moeda é gerada por meio do *Ganache* [Ganache, 2019], ferramenta disponibilizada pelo *Truffle*. A administração das moedas é feita pelo *Metamask* [Metamask, 2019], um *framework*, responsável em ligar o *Blockchain Ethereum* ao navegador, i.e., o *Metamask* é uma extensão que permite usar uma carteira de moedas sem baixar localmente a cadeia de blocos *Ethereum*. O Metamask é um mecanismo que permite gerir carteiras para administrar as moedas que são gastas ou adquiridas nas transações. O funcionamento básico é o seguinte: o Metamask comunica com o *Ganache*, assim, toda vez que uma transação é feita, uma atualização nas carteiras das partes envolvidas na transação é feita.

A quantidade de transações por bloco pode ser redefinida de acordo com o fluxo de transações no *Blockchain*. Os passos de todo o processo são:

1. um agente cliente *c* envia o seu *feedback f* para outro agente provedor *p*, a operacionalização do registro de *f* no Dossiê de *p* é feito por um *smart contract*;
2. um *smart contract* é uma transação remunerada. A remuneração é implementada pelo componente *Truffle*;
3. o Ganache insere a única transação em um bloco e atualiza a cadeia;
4. as carteiras de ambos os agentes, administradas pelo Metamask—recebem a atualização do saldo em moedas; e
5. o *feedback* recebido pelo agente cliente *c* também atualiza o Dossiê do agente provedor *p*, que na atual proposta deverá ser exibida em página *web* de consulta. O método de cálculo/atualização da reputação usa um mecanismo de decaimento do seu valor em função do tempo.

O desenvolvimento do *software* de experimentação/validação da proposta é em ambiente de programação *web*, usando o Truffle e o Metamask. Todavia, a grande parte dos mecanismos descritos até aqui faz parte de desenvolvimento *back-end*. Porém, para a administração dos agentes foi desenvolvida uma interface simples na forma de uma página *web*, onde é possível visualizar o *Dossiê* de cada agente. Deve-se notar que a abordagem inclui uma versão particular do Dossiê original [Silva, 2017], permitindo gerir múltiplas pastas; uma para cada tipo de serviço ofertado um agente provedor *p*.

A reputação de cada agente provedor é verificada no momento em que o Dossiê é recebido. O Dossiê recebido passa por uma avaliação inicial por preconceito (regras predefinidas) e recebe um peso dado por um conjunto de julgamentos definidos previamente. A operacionalização dos julgamentos pode ser dada pela abordagem AHP ([LESSING et al., 2019], [SAATY, 1980], [SAATY, 1996]).

4. DOSSIÊ MULTIPASTA

O modelo proposto, denominado de Dossiê *Multipasta*, derivado da proposta de [Silva, 2017] é um modelo para a estrutura da informação para o cálculo de confiança para sistema multiagente, com uma abordagem descentralizada, onde dados e controle são fisicamente e logicamente distribuídos. Esta abordagem permite que qualquer indivíduo de uma rede de agentes virtuais aberta, possa obter informações sobre outro indivíduo por meio de testemunhos armazenados localmente—espaço logico e/ou físico do próprio agente—, uma vez que tida a permissão do mesmo para o compartilhamento do testemunho, de modo seguro, sem que incidam modificações ou omissões de informações. A informação a ser compartilhada por cada agente e entre os agentes, é o Dossiê em questão, que essencialmente contém o histórico dos serviços prestados de cada indivíduo/agente.

Deve-se notar que cada agente pode prestar mais de um serviço, e que a qualidade de serviço prestado não é igual obrigatoriamente, necessitando assim de um mecanismo que armazene e atualize de forma segura e confiável o histórico de *feedbacks* de cada tipo de serviço. A estrutura para essa classificação de feedback por tipo de serviço assume o formato de um *Multipasta*, agregada ao modelo Dossiê para a organização por meio de pastas locais rotuladas com as descrições dos serviços de cada agente. A dificuldade aqui é garantir a atualização dos *feedbacks* de cada serviço de forma individualizada por agente e por tipo de serviço no espaço de cada agente.

4.1 O agente no Dossiê Multipasta

Em nosso contexto, todo agente é um indivíduo que presta ou solicita um serviço em uma comunidade de agentes, em que se estabelece como fonte padrão de informações a estrutura Dossiê. Em outras palavras, um agente pode prover e solicitar um serviço de qualquer outro agente que faça parte da comunidade. Desta forma, o agente dispõe de funcionalidades que podem ser utilizadas em determinadas situação

quando se faz necessário a avaliação do Dossiê—ou parte deste uma pasta da estrutura Multipasta—de outro agente da comunidade. A estrutura multipasta também pode ser vista como multidimensionalidade.

Um agente pode:

- **Solicitar o histórico das transações de outro agente:** quando um agente necessita saber a reputação de outro agente para estabelecer uma relação confiança no negócio que pretende fazer.
- **Liberar a visualização do Dossiê para um agente:** quando o mesmo recebe um pedido de visualização do seu Dossiê privado e o mesmo aceita liberar a visualização.
- **Negar a visualização do Dossiê para um agente:** quando o mesmo recebe um pedido de visualização do seu Dossiê privado e o mesmo se nega liberar a visualização.
- **Enviar um *feedback* para um agente:** quando o mesmo deseja emitir avaliação sobre o serviço prestado pelo agente provedor, encaminhando uma nota e uma observação do serviço prestado para o agente executor do serviço.
- **Receber um *feedback*:** obrigatoriamente, quando um agente recebe uma nota pelo seu serviço, a mesma é anexada com as demais notas em seu Dossiê—de acordo com a estrutura Multipasta—, no determinado serviço e propagado na rede *Blockchain*.

4.2 Multipasta

A estruturação da Multipasta dentro do modelo Dossiê é ilustrada pela Figura 14, que mostra a comunicação entre dois agentes com seus respectivos Dossiês. Deve-se observar a separação dos serviços prestados por cada indivíduo. Onde se observa três tipos de serviços com notas diferentes para cada um.

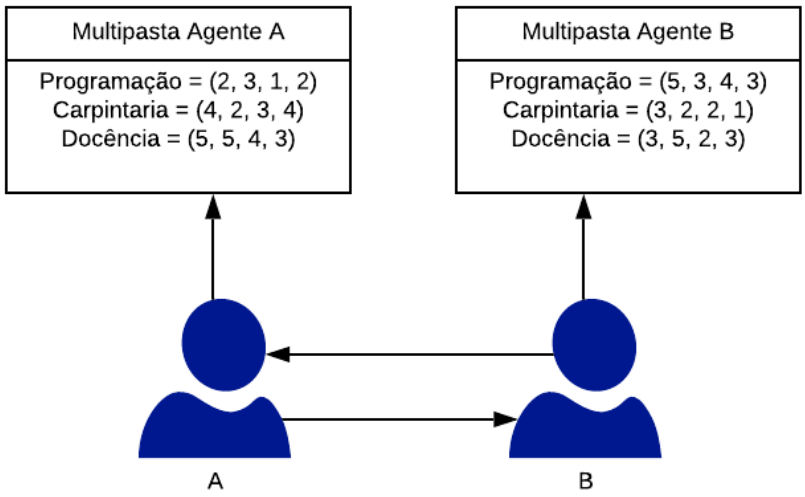

Figura 14. Dossiê Multipasta entre agentes.

Suponha-se que, o agente A solicite ao agente B o seu Dossiê com interesse no serviço de carpintaria. O agente B recebe a solicitação de agente A e libera a visualização do Dossiê para agente A, o qual tem um conjunto de notas para cada tipo de serviço que presta, onde as notas vão de 0 a 5. Os demais históricos dos serviços Dossiê do agente B também ficam disponíveis para visualização.

4.3 O fluxo

O fluxo de atualização de Dossiê, tomando com ponto de partida um serviço prestado, é mostrado na Figura 15, em que o processo de uma atualização se inicia com a própria solicitação e termina com a confirmação da operação de atualização, ou seja, com a inclusão da nova nota do serviço prestado na devida pasta do Dossiê do agente prestador.

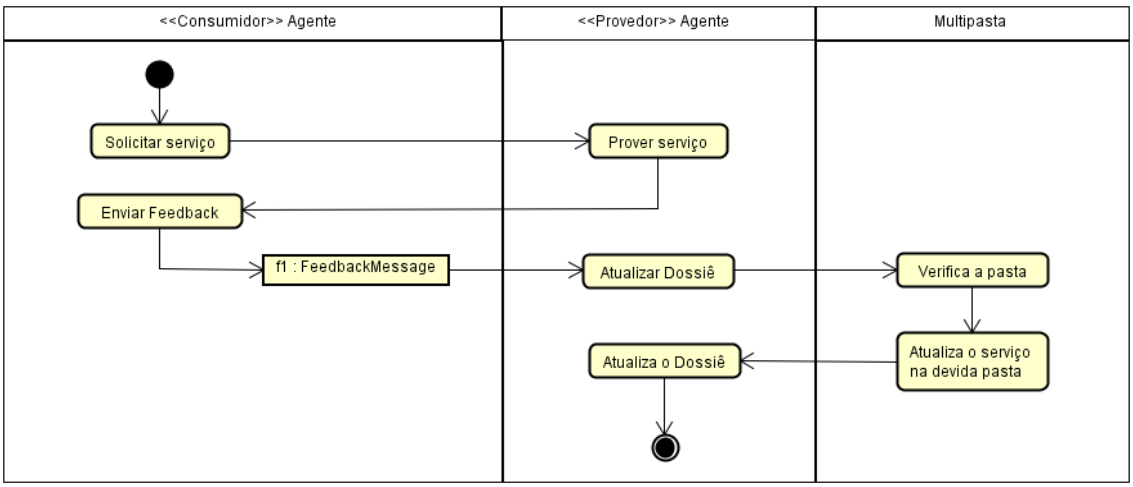

Figura 15. Atualização da multipasta do Dossiê.

Deve-se notar que o fluxo de atualização de Dossiê, ocorre apenas quando o serviço já foi prestado/realizado e o agente cliente deseja avaliar o seu agente provedor (Silva, 2017), desta forma ele envia, fazendo uso de um protocolo especializado, uma nota—que pode ser acompanhada por um comentário—, a qual é indexada a estrutura Multipasta do Dossiê do devido agente avaliado. Assim, o Dossiê do agente provedor é atualizado e tal atualização repercute no estrutura de resumo de *Hash* na Blockchain de forma automática, eliminando o risco do agente avaliado manipular ao seu favor o *feedback;* a atualização de *Hash* na estrutura Blockchain global é o mecanismo de proteção da autenticidade da informação para o cálculo de confiança e reputação baseado no Dossiê.

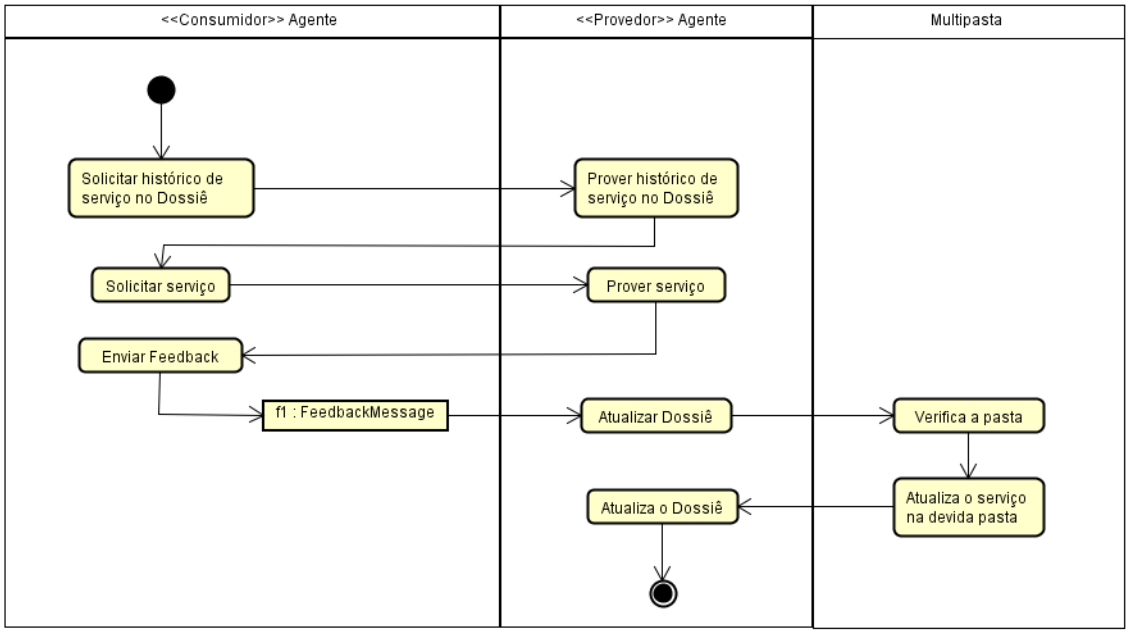

Figura 16. Solicitação de histórico e prestação de serviço com atualização da multipasta no Dossiê.

Na solicitação de visualização do Dossiê de um agente, a aceitação ou rejeição do compartilhamento acontece antes de se enviar o *feedback* ao agente provedor. Porém, a verificação da reputação apoia-se nas propriedades de segurança da informação herdada da Blockchain. Desta forma, antes do envio do *feedback*, de um dado serviço, três passos são executados:

- A solicitação da visualização do Dossiê do agente provedor pelo agente consumidor;
- A aceitação de visualização pelo agente provedor de seu Dossiê a favor do agente cliente, em o Dossiê fica visível para o agente solicitante/cliente; ou

- A negação de visualização pelo agente provedor de seu Dossiê contrariando a solicitação do agente cliente, em que a solicitação fica registrada como um evento dentro da cadeia de blocos como uma negação de informação.

Depois, a transação em si segue um processo que envolve apenas o agente cliente e o agente provedor. O processo completo foi mostrado na Figura 16.

De forma resumida, o processo completo de uma transação entre dois agentes consiste em passos que se iniciam com o agente cliente, que solicita um histórico de serviço do Dossiê de um agente Provedor. O agente provedor fornece o histórico solicitado para agente cliente, que decide solicitar um serviço, o agente provedor efetua o serviço, e quando pronto, o agente cliente faz sua avaliação e envia seu *feedback* para o agente provedor. O agente provedor então atualiza seu Dossiê na estrutura multipasta correspondente ao tipo de serviço prestado, e encerra a transação.

4.4 Componentes de Desenvolvimento

Para a operacionalização da lógica de fornecimento/consumo de serviços entre os agentes, é necessário que ocorram transações entre os mesmos. Essas transações são oficializadas com micro pagamentos, e estes micro pagamentos são administrados por meio de uma carteira digital individual para cada agente. A carteira individual de cada dispõe de um determinado valor de criptomoeda, que pertence a cada indivíduo, e este valor é relativo à cada agente, pois a quantidade de criptomoeda que cada agente possui, é determinada diretamente pela quantidade de micro pagamentos e recebimentos que o mesmo participa por meio das transações efetivamente realizadas.

A criptomoeda é gerada pela plataforma Truffle [Truffle, 2019], se utilizando de ambientes pré-definidos. Esses ambientes são três: (1) o *Ganache* que possibilita a criação de uma cadeia de blocos, levando em consideração que se pode fazer alterações na mesma e trabalhar na arquitetura desejada; (2) o *T Ruffle* que provê a criação de *smart contracts*, vinculando os mesmos na cadeia de blocos; e (3) o *react.js* [React, 2020], que auxilia no desenvolvimento de *front-end* do Projeto.

4.5 Fluxo técnico

A arquitetura Blockchain que envolve os agentes participantes da rede operacionalizam suas de transações por meio de comunicações diretas de solicitação de Dossiê e envio de *feedbacks*; tais transações são pagas por meio de micro pagamentos se utilizando de *smart contract*.

Basicamente, o processo como um todo consiste em três partes fundamentais para o funcionamento completo. Primeiramente, a solicitação para verificar o histórico de serviço, onde o mesmo pode ser aceito ou negado pelo agente provedor. O segundo passo é a contratação do serviço e a efetuação do mesmo, onde o mecanismo de decisão da contratação não faz parte da proposta aqui apresentada, pois ela consiste em criar um modelo de confiança entre os agentes participantes da Blockchain. O terceiro passo é o *feedback* retornado pelo agente consumidor ao agente provedor, que tem como pós-condição o seu correto armazenamento do Dossiê local do agente provedor.

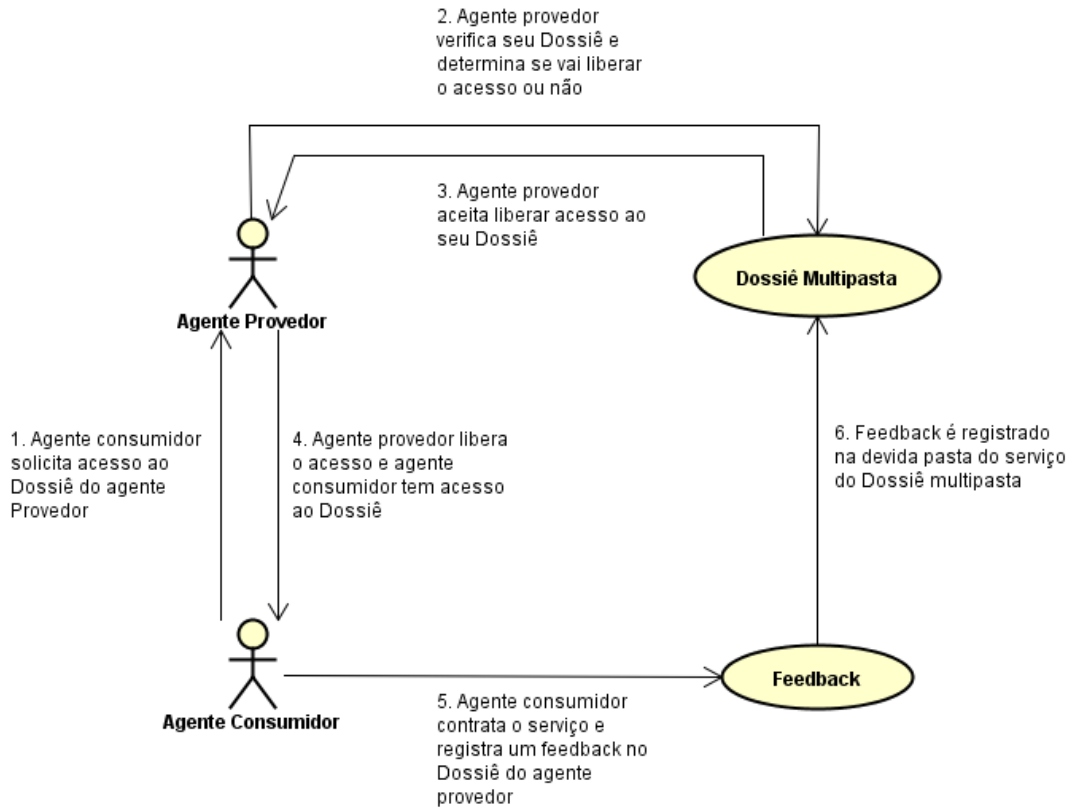

Figura 17. Funcionalidade Dossiê Multipasta.

Dentro da proposta apresentada, é importante explicitar o que são e como funcionam cada um dos componentes definidos dentro da arquitetura como um todo. Sabe-se que a arquitetura funciona de forma descentralizada por meio de uma arquitetura *P2P*. E que cada agente tem seu histórico de serviços, estruturado no seu Dossiê, que por sua vez é subdivido em diferentes pastas, uma para cada tipo de serviço, formando, como dito anteriormente, um modelo de Dossiê Multipasta. Cada agente também possui uma carteira com criptomoedas, usada com a finalidade de efetuar micro pagamentos relativos as transações dentro da comunidade de agentes.

O fluxo de operação com a estrutura Dossiê Multipasta é ilustrado na Figura 17. Os passos a seguir mostram onde cada componente técnico da Projeto está inserido dentro do contexto Blockchain.

1. O agente cliente solicita para o agente provedor a visualização do seu Dossiê.
    a. A solicitação fica de forma pendente na conta do agente provedor.
    b. A solicitação é paga, de forma que se desconta da carteira de criptomoeda do agente cliente.
2. O agente provedor tem uma solicitação de visualização em sua conta. O mesmo pode escolher em aceitar ou rejeitar tal solicitação.
    a. Se aceitar, o agente cliente terá acesso ao seu Dossiê. Como consequência, a aprovação da visualização é paga pelo agente provedor, para liberar acesso ao seu Dossiê, de forma que desconta da carteira de criptomoeda do agente provedor.
    b. Se negar, o registro de solicitação de *feedback* fica armazenado na cadeia de blocos como negado.
3. O agente cliente teve acesso ao Dossiê do agente provedor. Deve-se salientar que o agente cliente pode solicitar Dossiê com todos os agentes provedores de serviço que deseja.
4. O agente cliente fecha negócio com o agente provedor e registra um *feedback* que é inserido no Dossiê do agente provedor.
    a. O *feedback* é inserido de forma automática no Dossiê do agente provedor, não a necessidade de aceite prévio.

b. O *feedback* é pago, descontando-se um determinado valor da carteira de criptomoeda do agente consumidor.

5. O *feedback* é inserido na subpasta do determinado serviço prestado do Dossiê do agente provedor.

As transações realizadas são utilizadas para movimentar os processos do sistema em si. De forma que cada solicitação de visualização de Dossiê de um agente é uma transação, e é paga. E também o envio de *feedback* para o agente provedor também gera uma transação, e é paga. Em resumo, cada vez que se movimenta uma informação, gera-se uma transação com custo, e é paga por um micro pagamento.

O Dossiê de cada agente está associado a um *smart contract*, que está persistido e protegido em um *Ledger* ligado a cadeia Blockchain. Cada vez que o agente recebe uma solicitação de visualização do seu Dossiê ou *feedback*, o *smart contract* é atualizado de forma que ele irá conter todas as solicitações de visualização, com suas aprovações ou rejeições, e também todos os *feedbacks* já recebidos. E então, inserido na cadeia de blocos novamente, ficando atualizado o Ledger para todos os nós, tendo cada nó de forma local, uma cópia da Blockchain.

4.6 Ledger e Dossiê

A atualização do *Ledger* só ocorre quando uma transação é efetuada, e esta transação é sempre operacionalizada por um *smart contract*, por restrição de arquitetura do modelo Dossiê, i.e., todas as transações são registradas por *smart contract* e inseridas em uma estrutura *Ledger*.

O *smart contract* responsável pelos registros de agentes tem uma única finalidade, registrar de forma sistemática e segura cada indivíduo que venha a fazer parte da comunidade ou sistema. Sua estrutura é composta por alguns atributos para operacionalizar as transações e registros.

Sendo que: o *id* é o identificador do agente cadastrado. Name, lastname, email e cpf são informados pelo próprio indivíduo no momento em que se realiza seu cadastramento. A atributo createdAtDate armazena a data de inscrição do agente

dentro do Sistema e updatedAtDate salva as alterações que esse agente realiza dentro do seu Dossiê.

```
struct Agent {
    address id;
    bytes32 name;
    bytes32 lastName;
    bytes32 email;
    bytes32 cpf;
    bytes32 createdAtDate;
    bytes32 updatedAtDate;
  }
```

Quando é criado um usuário dentro do Sistema, este usuário é indexado a uma carteira de criptomoeda, pois é por meio dessa carteira que se administrará os recursos usados nos micros pagamentos das transações entre agentes clientes e provedores. Quem faz o gerenciamento de cada carteira é o Metamask [Metamask, 2019], (ver detalhes de funcionamento no Capítulo 3.3), o qual opera na forma de um mecanismo de transação.

O outro *smart contract* que também faz parte do *Ledger* é o próprio *smart contract* do Dossiê, que contém atributos próprios para dar suporte a operacionalização de todas as transações realizadas com micros pagamentos. Sua estrutura se dá da seguinte forma:

```
struct Feedback {
    address apraiser;
    address ratedAgent;
    string description;
    string serviceType;
    uint feedback;
    string createdAtDate;
 }
```

O atributo “apraiser” armazena o identificador do agente avaliador. O atributo “ratedAgent” armazena o identificador do agente avaliado, “description” armazena a descrição que um agente cliente pode fornecer ao agente provedor do serviço. O atributo “serviceType” é o tipo de serviço disposto em uma subpasta da multipasta do Dossiê do agente provedor. O atributo “feedback” denota a nota atribuída ao agente provedor. Por fim, o atributo “createdAtDate” registra a data de ocorrência de *feedback*.

Na medida em que cada *feedback* ocorre, se atualiza o Dossiê do agente provedor no *smart contract* responsável pelo Dossiê, e também se atualiza

automaticamente o *Ledger, que com os resumos dos últimos Hashs de cada Dossiê do Sistema*. Então, deve-se observar que o *Smart Contract* Dossiê tem na sua tutela a responsabilidade em manter o registro e a integridade de todos os *feedbacks* gerados pelos agentes, inserindo-os na rede de dossiês na medida que cada novo *feedback ocorre*, escrevendo também no *Ledger* de resumos de *Hashs*—conforme já explicado anteriormente.

4.7 Anatomia do Dossiê

Tendo em mente que o Dossiê é atualizado a cada pedido de visualização de Dossiê, sendo ele aceito ou não, e a cada *feedback* registrado sobre um agente provedor, todo o processo de atualização do *Smart Contract* Dossiê é subdivido em duas fases. Na primeira, a requisição do Dossiê de um agente cliente para um provedor. E depois o envio do *feedback* do agente cliente ao agente provedor em questão.

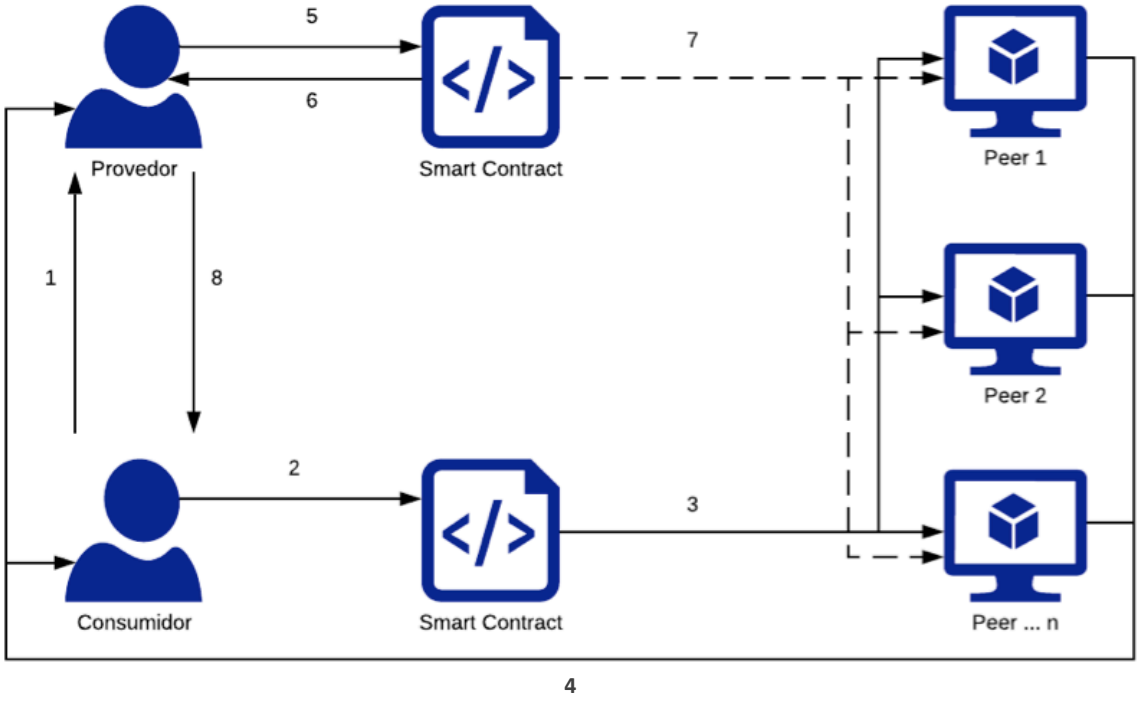

Figura 18. Estrutura de solicitação de Dossiê.

Desta forma, na Figura 18, por meio de oito passos, delineia-se o fluxo percorrido por um agente que solicita de outro agente o seu Dossiê para visualização.

1. Agente cliente pede permissão para visualização do Dossiê do agente provedor. O agente cliente pode fazer a solicitação da visualização do serviço para

quantos agentes desejar, por *default*, o mesmo não tem acesso livre aos dossiês solicitados.

2. A solicitação de visualização do Dossiê fica armazenada no *Smart Contract* Dossiê, que faz parte do *Ledger* público, ou seja, que atualiza a cadeia de blocos e propaga na rede a informação relativa à solicitação. A solicitação de visualização do Dossiê encerra um micro pagamento, gerenciado pela carteira do Metamask, que opera com a plataforma Ganache para propagar a atualização *Ledger* público.
3. O *Smart Contract* Dossiê, responsável pelo armazenamento de toda solicitação e *feedback*, se utilizando de transação remunerada por micro pagamento de um agente, atualiza e propaga em toda a cadeia de blocos a solicitação de visualização de Dossiê.
4. Todos os agentes da Blockchain, tem em seus nós da cadeia de blocos a solicitação de Dossiê do agente cliente ao agente provedor, depois que a transação foi feita com o micro pagamento mediada por um *Smart Contract,* o Ganache propagada a atualização na cadeia de blocos.
5. O agente provedor verifica o pedido de visualização do seu Dossiê feito por um agente cliente, que o realizou por meio de um micro pagamento administrado na forma de uma carteira do Metamask, e propagado em toda a rede pelo Ganache. O agente provedor insere esse pedido em seu Dossiê de forma pendente, e para liberar, precisa apenas aceitar o pedido de visualização do seu Dossiê. Se o agente provedor não liberar o acesso, permanece apenas o pedido de forma pendente em seu Dossiê, em modo de espera.
6. O agente provedor libera o acesso do Dossiê para o agente cliente. Tal liberação de visualização é realizada por meio de um micro pagamento por parte do agente provedor, que desconta em sua carteira administrada pelo Metamask.
7. A liberação da visualização do Dossiê atualiza o *Smart Contract* Dossiê e propaga em toda a cadeia de blocos se utilizando do serviço do Ganache.
8. O agente cliente tem acesso ao Dossiê do agente provedor, e todos os nós da Blockchain tem, em seu nó, o histórico desde a solicitação até o aceite do pedido pelo agente cliente.

A Figura 19 exibe o processo pós serviço prestado entre os agentes clientes e provedores. Onde o agente cliente envia um *feedback* para o agente provedor; tal feedback repercute na cadeia de blocos por meio da atualização de Hash na estrutura *Ledger* pública e o feedback em si fica registrado de fato no Dossiê provedor.

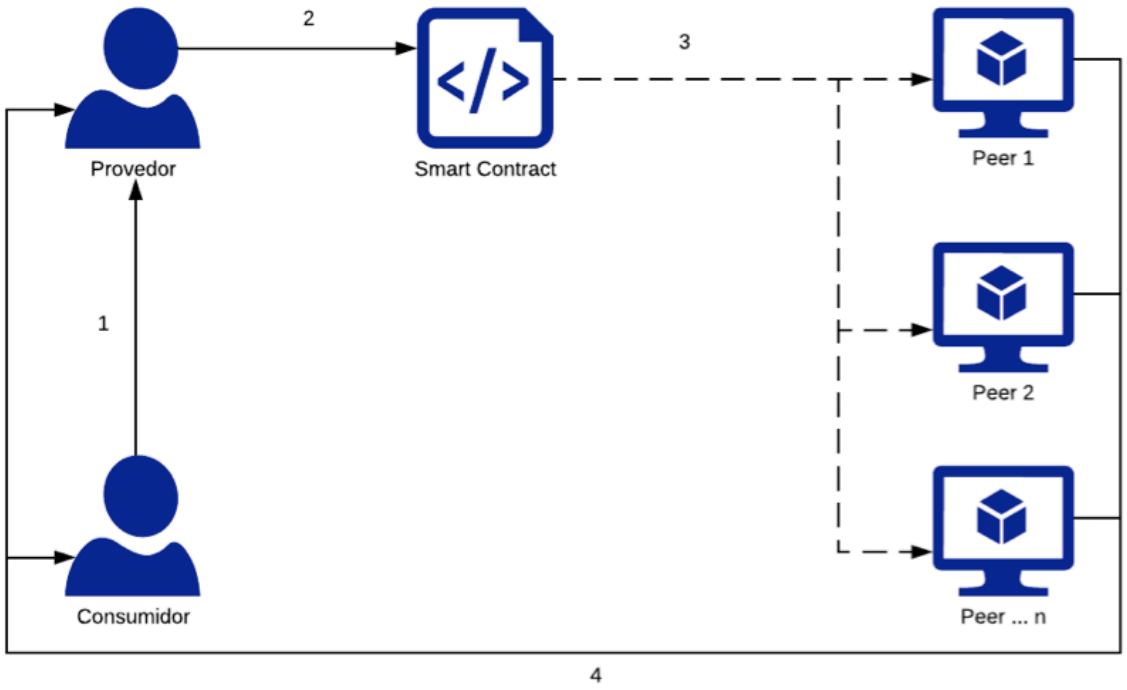

Figura 19. Estrutura de envio de feedback para agente Provedor.

1. Agente cliente envia um *feedback* ao agente provedor do serviço prestado, o *feedback* gera um micro pagamento que desconta da carteira do agente cliente, gerenciada pela Metamask.
2. O Dossiê do agente provedor é atualizado na *Smart Contract*, não havendo necessidade de o mesmo aceitar o recebimento do *feedback*, tal processo é automático para evitar a seleção, por exemplo, dos bons *feedbacks*.
3. *Smart Contract* propaga a transação de *feedback* no *Ledger* público se utilizando do serviço do Ganache, todos os nós da cadeia de blocos são atualizados com a transação de *feedback* entre o agente cliente e o provedor.
4. A cadeia de blocos é atualizada se utilizando do serviço do Ganache, e o resultado do *smart contract* é atualizado para todos os nós da cadeia de blocos através do Ledger, replicado por meio de mineração.

4.8 Atualização do *Smart Contract* Dossiê

O *Smart Contract* Dossiê é o intermediador entre toda comunicação para a realização de uma transação entre agentes, e é a partir dele que se atualiza a cadeia de blocos em cada nó da rede P2P. O *Smart Contract* é necessário para que se fixe de forma imutável e fidedigna todas as transações que ocorram na linha do tempo do sistema Dossiê. Na Figura 20 ilustra o processo de atualização da cadeia de blocos por parte do serviço que o *Smart Contract* realiza. Os passos descritos são repetidos de forma completa, ou até certo ponto, todas as vezes que os agentes decidem solicitar ou compartilhar *feedbacks* entre si.

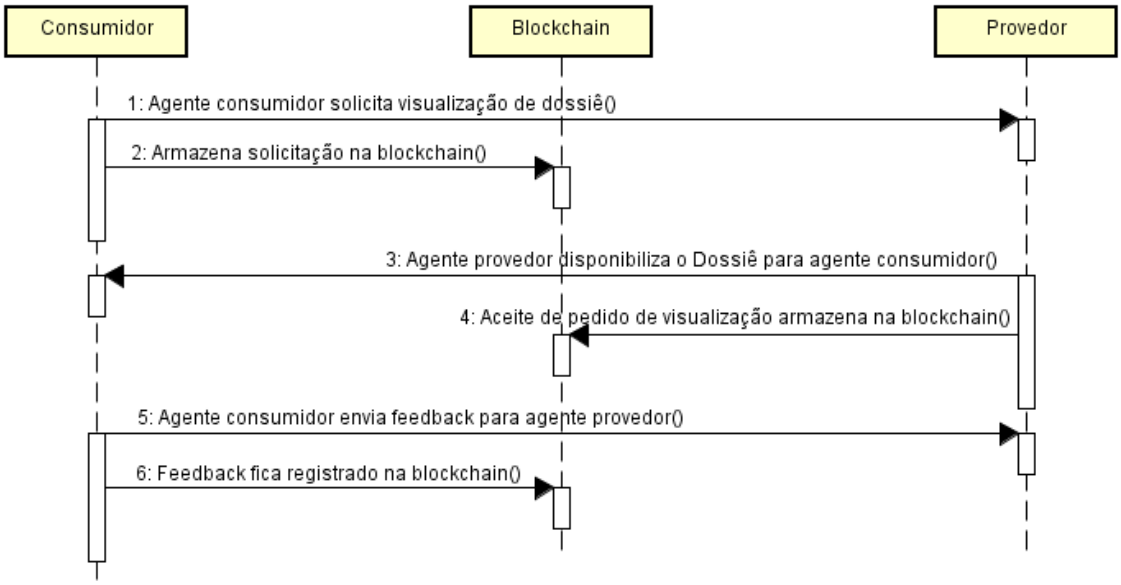

Figura 20. Intermediação do Smart Contract nas transações.

No passo 1, o agente cliente solicita a visualização do Dossiê do agente provedor, e nesse momento um micro pagamento já é registrado no *Smart Contract* a solicitação. Tal micro pagamento é dispersado na Blockchain na interação 2.

Na interação 3, o agente provedor aceita disponibilizar seu Dossiê para o agente cliente, e realizando um micro pagamento por parte do provedor, o agente cliente tem acesso ao Dossiê, o *Smart Contract* é novamente atualizado e propaga-se de forma automática na cadeia de blocos a operação na interação 4.

Após um serviço prestado, o agente cliente envia ao agente provedor um *feedback*, interação 5, que também é descontado de sua carteira resultado de um

micro pagamento. Tal transação atualiza o *Smart Contract* e propaga-a em toda a cadeia de blocos, na interação 6, que de um ponto de vista estrutural, é a última atualização do processo entre agentes que se envolveram na prestação e consumo de um serviço.

5. APLICAÇÃO

5.1 Introdução

O resultado desse estudo é mostrado por meio de uma aplicação Web. Não há limites para participantes na aplicação. A ilustração é feita por meio de um ambiente simples, se utilizando de uma *localhost*, em que a descentralização encerra uma comunidade de quatro agentes participantes, todos operando com suporte dos mecanismos do Ganache, como explicado anteriormente, e fazendo suas movimentações por meio de transações geridas monetariamente pela Metamask.

5.2 Cenário: Exibindo Dinâmica de troca de informação

O cenário ilustrativo com quatro agentes faz interações entre si dentro da estrutura de cadeia de blocos do Dossiê, sendo eles: Paulo Garcia, Roberto Pereira, Cesar Almeida e Eduardo Franciscon. Deve-se observar a atuação do mecanismo de solicitação de Dossiê e envio de *feedback* entre eles. A aplicação parte do cadastro dos agentes envolvidos no Sistema. A Figura 21 ilustra a interface do cadastramento de cada agente. A interface de cadastro é o ponto de entrada de cada agente na Blockchain Dossiê.

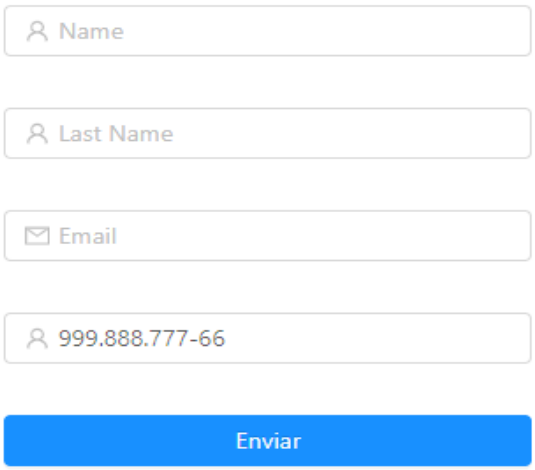

Figura 21. Cadastro de usuários/agentes na rede Dossiê.

O cadastro de um usuário já movimenta uma transação na Metamask, i.e., já se inicia uma carteira digital de criptomoeda nesse momento, e a primeira transação é realizada no cadastro do usuário dentro da rede Dossiê. A Figura 22 permite verificar o modelo da carteira de administrada por uma extensão do Metamask.

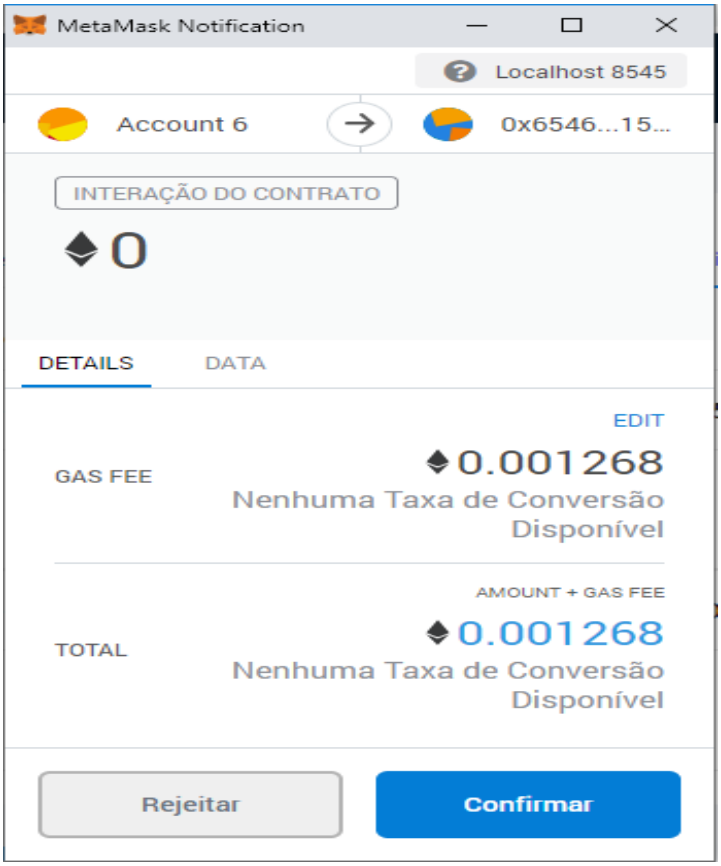

Figura 22. Carteira digital MetaMask para o gerenciamento de criptomoedas.

Após o cadastro, cada agente tem seu perfil dentro da comunidade Dossiê—criado sem nenhum valor inicial—, e se utilizando desse perfil, o mesmo gerencia seus pedidos de visualização do seu Dossiê e observa seu histórico de *feedbacks*, recebidos ao longo do tempo por agentes clientes de seus serviços. A partir de uma interface simples, onde se tem os dados cadastrais e um menu, em que se visualiza os *feedbacks* recebidos, os pedidos de visualização para aceite ou não, e encaminhamento de *feedbacks* para outro agente prestador de serviço. A Figura 23 mostra a interface do agente *Cesar Almeida* cadastrado na rede Dossiê.

Assim que criado o perfil do agente, sua reputação se inicia em um estado neutro, sem nenhum *feedback* recebido. Com o passar do tempo, o agente realiza

transações e *feedbacks* são recebidos por seus serviços, sua reputação começa a se formar.

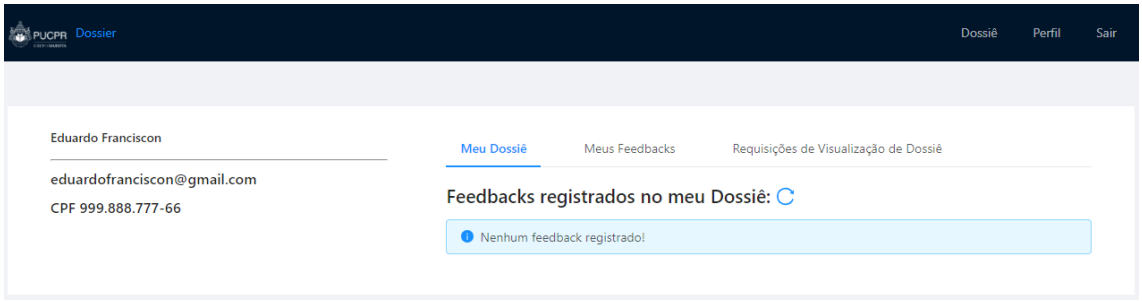

Figura 23. Perfil do usuário na rede Dossiê.

É importante salientar que os serviços prestados pelos agentes são definidos pela aplicação, de forma que um agente não pode cadastrar um novo tipo de serviço para ser prestado, ou seja, nessa aplicação, os serviços prestados são: consultas médicas, pizzas, informática, mecânica, orientação escolar e jardinagem. Na aba "meu Dossiê" é possível visualizar os *feedbacks* recebidos de outros agentes. A Figura 24 mostra a estrutura multipasta dentro do Dossiê, com os *feedbacks* recebidos e organizados pelos serviços prestados.

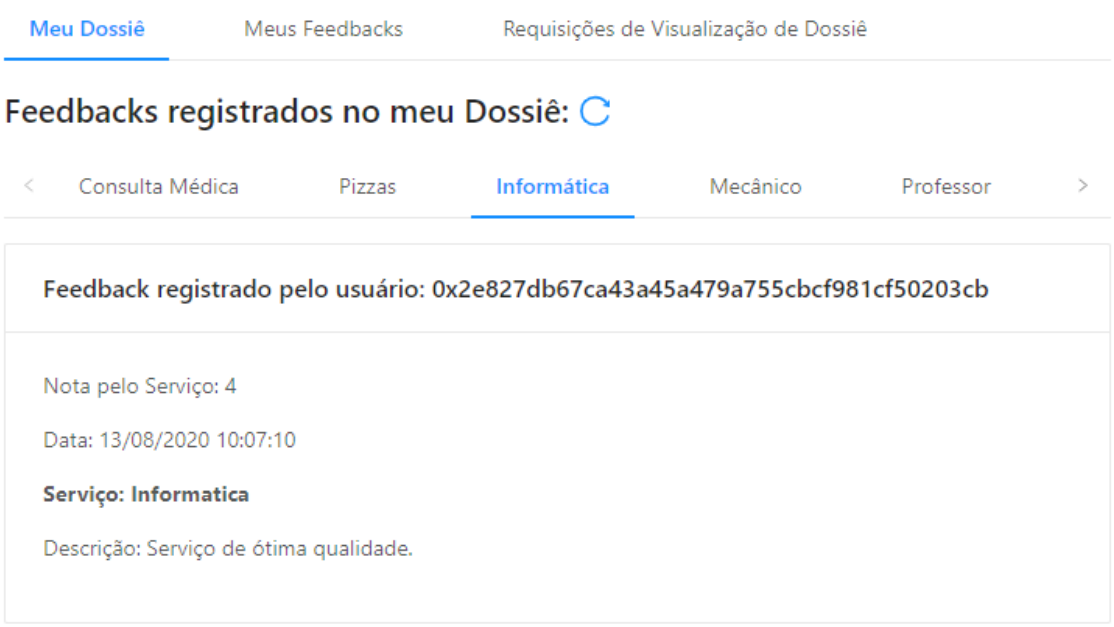

Figura 24. Registro dos feedbacks recebidos no Dossiê.

No *feedback* recebido e registrado pelo agente identificado no *hash* "0x2e827db67CA43A45A479a755CbcF981CF50203CB", tem-se a nota, data e hora do envio, o serviço que foi prestado pelo agente provedor e por fim uma descrição, que o agente cliente pode fazer a respeito do trabalho prestado.

O agente cliente pode encaminhar um *feedback* de forma rápida, se utilizando de uma interface de registro, cf. Figura 25, uma transação armazena na cadeia de blocos o registro do *feedback* de forma imutável. As mesmas variáveis visíveis no recebimento do *feedback* (cf. Figura 24) são verificadas na interface de registro na Figura 25. Em "Address" informa-se o *hash* do usuário que irá receber o *feedback*, por exemplo "0x3eDD9b5961ee3FaDb48fbC74B368aa356906eccf". Ao comandar a ação "Registrar" o Metamask de forma automática registra o *feedback* com uma transação.

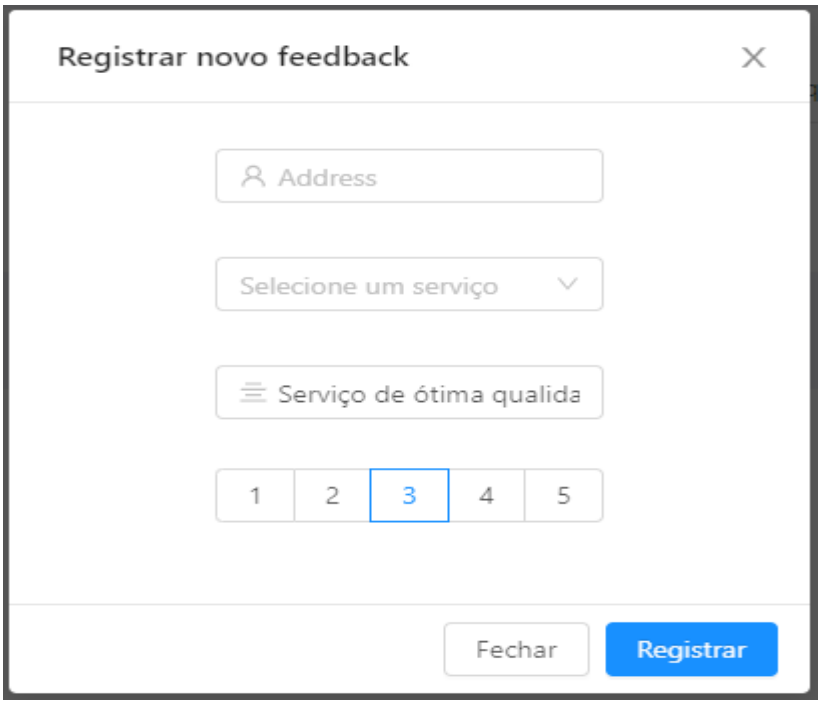

Figura 25. Registro de feedback.

Para solicitar acesso ao Dossiê de um agente provedor, num primeiro momento o agente cliente solicita o acesso por meio da ação "Requisitar visualização", que está disponível na aba "Dossiê", cf. Figura 23. Após a solicitação do Dossiê do agente provedor, há duas situações podem ocorrer relativo a tal solicitação, ela pode ser aceita selecionando-se a opção "Dossiê", caso contrário, permanece-se como pendente o pedido de visualização selecionando-se a opção "Pendente". A Figura 26 mostra que todas as situações em que um pedido de visualização de Dossiê pode figurar-se.

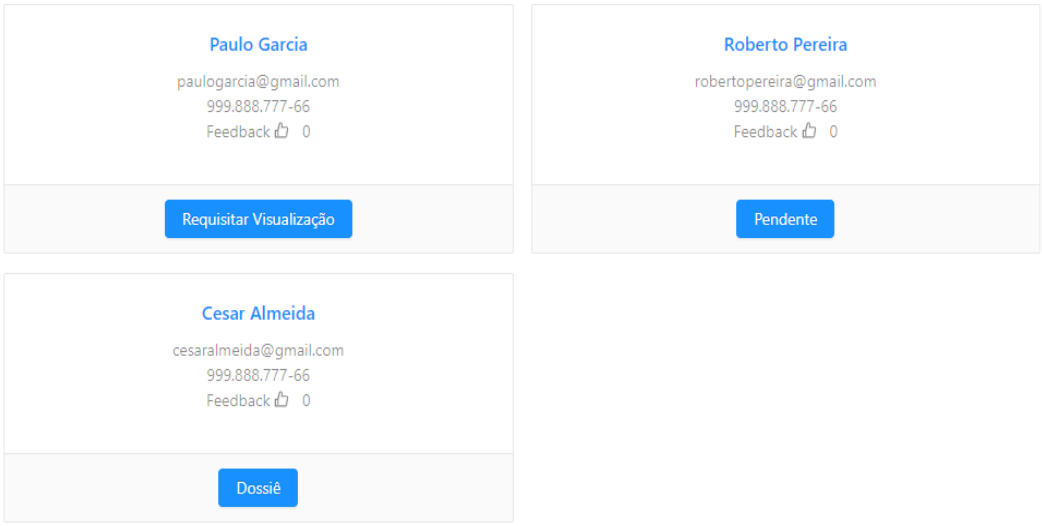

Figura 26. Fases de um pedido de visualização de Dossiê.

O agente provedor que recebe a solicitação para visualização de seu Dossiê, dispõe de um mecanismo simples para aceitar ou rejeitar a visualização. No exemplo em questão, interagindo com um ator ser humano via interface Web, a ação de deslizar um recurso gráfico (e.g., botão), para o lado do aceite, disponibiliza seu Dossiê para o agente solicitante, ou seja, deslizando o recurso gráfico na forma positiva, uma micro transação é realizada, envolvendo aqui o gerenciado de carteira Metamask e o Dossiê do agente requerido fica disponível para o agente solicitante.

A Figura 27 mostra duas solicitações de Dossiê de dois agentes diferentes, na primeira solicitação o agente disponibilizou seu Dossiê para visualização, na segunda opção ele optou por manter sob pendência—ou seja, não aceitou compartilhar.

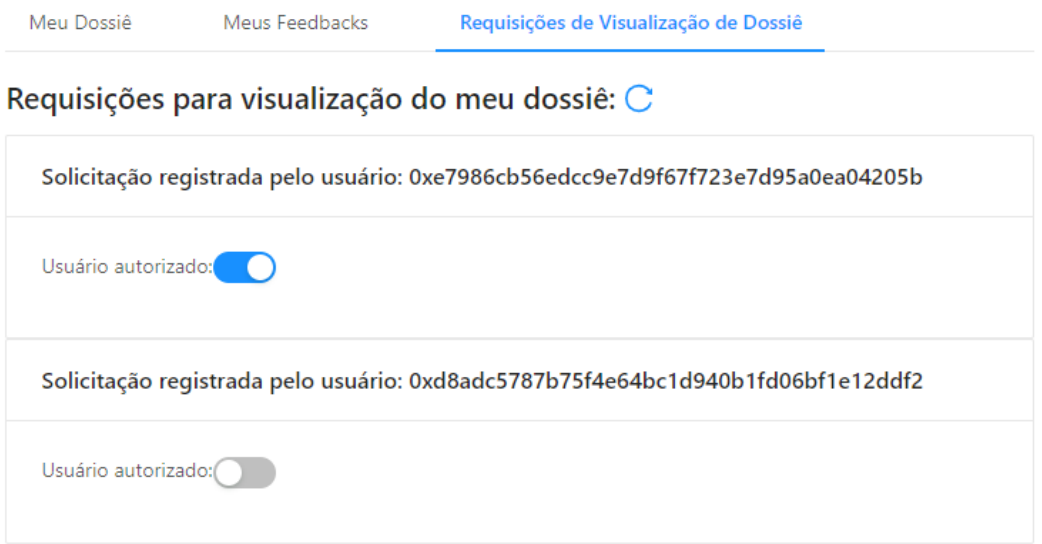

Figura 27. Solicitações de visualização de Dossiê.

Quando um agente cliente recebe o aceite de visualização de Dossiê do agente provedor, pode-se verificar na aba Dossiê (cf. Figura 23), a mesma aba que se faz a solicitação, que o botão de ação mudou, (cf. Figura 26), e a indicação de ação que antes era de “Requisitar Visualização” torna-se “Dossiê”, a ação de selecionar (por exemplo, por meio de um click do mouse), abre-se o Dossiê Multipasta do agente provedor, cf. Figura 28, em que é possível verificar o histórico de todos os serviços prestados pelo agente em questão.

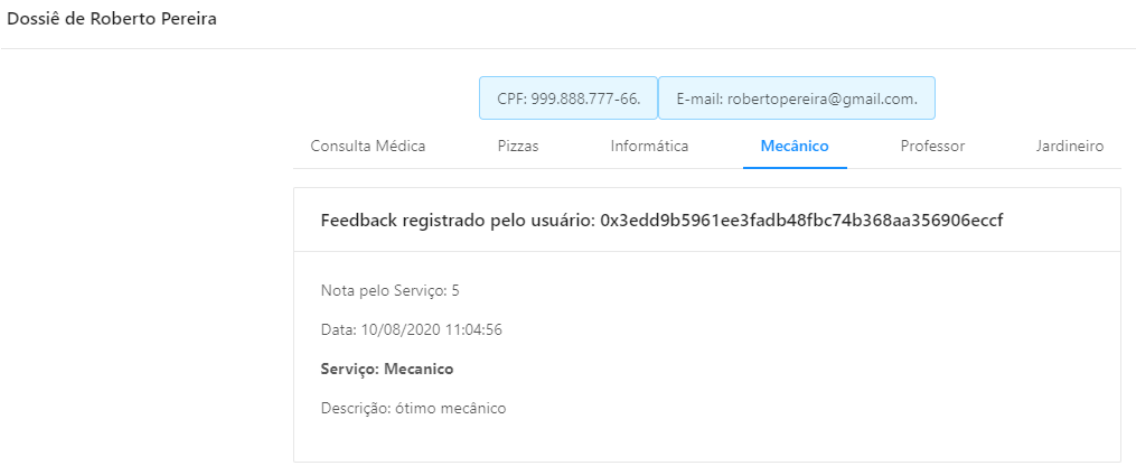

Figura 28. Cada agente tem um Dossiê em um ambiente descentralizado.

Cada agente pode solicitar o Dossiê de todos os agentes prestadores que desejar, da mesma forma, ele pode prestar todos os serviços dispostos na plataforma. Os mecanismos de solicitação e prestação de serviços funcionam sistematicamente seguindo os passos da arquitetura proposta neste trabalho, atualizando a Blockchain e propagando a atualização da rede. Desta forma, em um ambiente prático, esses passos ilustrados, podem ser repetidos entre diversos agentes enquanto for necessário.

5.3 Considerações do Capítulo

Este Capítulo mostrou, tomando como ponto de partida um cenário bem simples, a dinâmica da troca de informações na forma de leitura de estruturas privadas. Cada leitura de uma informação privada segue um protocolo rigoroso transacional e remunerado. Tal mecanismo mantem a rastreabilidade, em particular, das operações relativas as requisições de informações e as avaliações de serviços na forma de feedbacks. Essa dinâmica foi implementada se utilizando de uma plataforma real Blockchain, a Ethereum, na qual foram programados fluxos de atividades para mostrar

todo o processo de solicitação de Dossiê e envio de *feedbacks*. Pode-se avaliar a dinâmica do Sistema navegando sobre um conjunto de “janelas” de uma aplicação Web. E assim, corroborando com o nosso objetivo geral, que era mostrar que é possível integrar o modelo Dossiê Multipasta dentro de uma plataforma Blockchain real, no caso a Ethereum, e fazê-lo funcionar com seus mecanismos e estruturas propostos: estrutura de dados (Dossiê individual de cada agente) e mecanismos de controle descentralizados (transações de Dossiê).

6. CONCLUSÃO

Nesse trabalho foi apresentado o modelo Dossiê Multipasta, um modelo de suporte ao confiança em sistema multiagente, baseando-se na tecnologia *Blockchain*, em particular, na plataforma de desenvolvimento e execução *Ethereum*. O esforço empreendido permitiu corroborar com o nosso objetivo geral, que era mostrar de forma prática o uso da tecnologia Blockchain integrada a um ambiente de suporte ao desenvolvimento de agentes de software. O principal ganho desta integração, considerando a natureza de um sistema multiagente, concerne a viabilização do armazenamento de dados sensíveis de forma decentralizada ou local em cada agente, e.g., feedbacks relativos a serviços prestados, mantendo as propriedades de imutabilidade, rastreabilidade e segurança; em resumo, o ganho consiste em permitir interações seguras entre agentes sem o auxílio de uma entidade centralizadora para garantir a fidedignidade das transações e seus traços.

A arquitetura do sistema desenvolvido, vai além do modelo de cliente/servidor utilizado por sistemas baseados apenas na interação agente cliente e provedor de um serviço. Com um paradigma descentralizado, o modelo Dossiê Multipasta faz o uso da própria estrutura para distribuir as informações do sistema e descentralizar os dados, pois o próprio mecanismo de descentralização, através da mineração, faz com que todos os nós participantes da Blockchain tenham de forma local uma cópia do Ledger, reduzindo problemas de ataques que podem beneficiar um ou mais agentes não honestos ou hostis; tal garantia apresenta um interesse real da utilização da tecnologia Blockchain em ambientes com muitos agentes distribuídos.

O ambiente, quando administrado pelo modelo Dossiê Multipasta, quebra o paradigma de disposição de histórico e registro de avaliações, encontrados em plataformas *online* populares. Pois, em um primeiro momento, talvez não seja interessante para o indivíduo que disponibiliza um serviço, mostrar seu histórico para qualquer outro indivíduo que deseje, e desta forma, fica a seu critério autorizar ou não

a visualização deste. Em um segundo momento, o registro de uma avaliação/*feedback* sobre um agente que prestou um serviço, é registrado por meio de uma transação monetizada, com um micro pagamento em moeda digital, realçando o interesse e conclusão apenas de negócios bem-estruturados.

Por fim, o modelo Dossiê Multipasta, utiliza um modelo de registro e solicitações de informações com micro pagamentos, com um valor monetário pouco significante na perspectiva de custo/benefício. Pois, esse ambiente construído tem a finalidade apenas de manter a integridade dos históricos dos agentes participantes da rede. Assumiu-se que os indivíduos que participam dessa comunidade, interessam-se em prestar serviços e manter íntegra sua competência face as avaliações dos serviços prestados para uma comunidade. Assim, os micro pagamentos entre os indivíduos que interagem a comunidade, dão a opção tanto de fornecer algo, quanto de ser avaliado, registrando a ação em um Ledger descentralizado, visível para toda a comunidade que se utiliza do sistema, pois cada agente participante da Blockchain também é um nó que minera a cadeia de blocos e atualiza o Ledger.

6.1 Contribuições

A principal contribuição desse trabalho foi um ambiente prático, totalmente descentralizado, se utilizando de uma estrutura Multipasta, em que agentes podem interagir entre si, de forma a manter seus históricos individualizados e segmentados, e enviar avaliações para outros agentes. Todas as interações entre agentes são realizadas por meio de transações acompanhadas de micro pagamentos, então, outra contribuição importante consiste na estratégica de como uma transação digital movimenta um valor monetário, e como uma carteira de moeda digital funciona, com mecanismo de *hash* para garantir a integridade e segurança da rede antes de distribuir a transação entre os seus nós. Deve-se ressaltar a carência de esforços colocando em sinergia a arquitetura Blockchain e agente de *software*, em particular, na comunidade científica, neste sentido esse trabalho é também uma contribuição.

A revisão sistemática que foi empreendida no início desta pesquisa aplicada, relacionada as arquiteturas Blockchain é também uma contribuição importante, pois

ela permitiu mostrar com mais clareza as aplicações e mecanismos que uma Blockchain pode assumir para atender a diversos cenários de interação entre indivíduos.

6.2 Trabalhos futuros

A pesquisa feita até o momento deixa em aberto várias oportunidades de trabalhos futuros, listadas como segue:

1. avaliar o uso da rede de Dossiê Multipasta para um grande sistema, simular a uma rede cartórios, em que cada cartório local é gerido por um agente com o seu Dossiê Multipasta. A distribuição do controle e dos dados seria assegurada pela abordagem agente e a imutabilidade da informação da rede seria assegurada por um *Ledger* público leve, armazenando apenas o último *hash* de atualização de cada Dossiê de cada agente.

2. avaliar estratégias que implementem novas funcionalidades para integrar de maneira mais objetiva um *e-commerce*, em que problemas de engenharia social, como conluio e preconceito entre indivíduos sejam tratados de forma adequada.

3. avaliar estratégias que permitem tratar de forma eficiente situações que um agente recebe milhares de solicitações de visualização de Dossiê em um pequeno espaço de tempo.

4. avaliar estratégias que permitem verificar e apresentar eficientemente *feedbacks* e o resultado do cálculo de confiança de um agente provedor, por exemplo, colocando em destaque pontos negativos e positivos do avaliado.

5. avaliar estratégias que permitem priorizar tomadas de decisões a partir de métodos, por exemplo, usando a abordagem AHP já desenvolvida no laboratório de Agentes de Software [Lessing et al., 2019].

6. testar ambiente, envolvendo uma grande quantidade de agentes, colocando em perspectiva as alternativas de consenso e prova de trabalho para armazenamento no *Ledger* de grandes quantidades de dados, já que nesse momento, cria-se um novo bloco na cadeia de blocos para cada transação realizada.

7. avaliar modelo para extrair a reputação a partir de texto [Granatyr 2017]

6.3 Publicações

Parte do trabalho apresentado, deu origem, em particular, uma publicação científica (a primeira da lista) em conjunto com outros colegas (ou ex-colegas) grupo/laboratório de Agentes de Software do PPGIa da PUCPR. Segue abaixo a lista de publicações:

1. EA Franciscon, MP Nascimento, J. Granatyr, MR Weffort, OR Lessing e EE Scalabrin, "A Systematic Literature Review of Blockchain Architectures Applied to Public Services", *2019 IEEE 23ª Conferência Internacional sobre Trabalho Cooperativo em Design Apoiado por Computador (CSCWD)* , Porto , Portugal, 2019, pp. 33-38, doi: 10.1109 / CSCWD.2019.8791888.

2. O. R. Lessing, M. Weffort, J. Granatyr, E. Franciscon and E. E. Scalabrin, "Decision-Making: From Pure Uncertainty to Measurable Risk," *2019 IEEE 23rd International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, Porto, Portugal, 2019, pp. 170-175, doi: 10.1109/CSCWD.2019.8791890.

3. C. F. Scatambulo Costa and E. Alexandre Franciscon, "Data Mining applied to the navigation task in autonomous robots," *2020 IEEE Symposium on Computers and Communications (ISCC)*, Rennes, France, 2020, pp. 1-6, doi: 10.1109/ISCC50000.2020.9219731.

7. Referências

Ahram T.; Sargolzaei A.; Sargolzaei S.; Daniels J.; Amaba B. Blockchain technology innovations. IEEE Technology & Engineering Management Conference (TEMSCON). p. 137–141. 2017

Alansari S. A.; Paci F.; Sassone V. Distributed Access Control System for Cloud Federations. IEEE 37th International Conference on Distributed Computing Systems. p. 1231-1236. 2017

Aljazzaf Z. M., Perry M. and Capretz M. A. M. 2010. Online trust: Definition and principles. Proceedings of the Computing in the Global Information Technology (ICCGI). 20–25 Sept. Valencia, Spain: IEEE, 163–168.

Angraal, S.; Krumhols H. M.; Schuls W. L. Blockchain Technology: Applications in Health Care. Circulation. Cardiovascular quality and outcomes. 2017

ANTONOPOULOS, Andreas M. Mastering Bitcoin: unlocking digital cryptocurrencies." O'Reilly Media, Inc.", 2014.

Artz D.; Gil Y. A survey of trust in computer science and the Semantic Web. Web Semantics: Science, Services and Agents on the World Wide Web 5 (2007) 58–71

Bartolucci S.; Bernat P.; Joseph D. SHARVOT: secret SHARe-based VOTing on the blockchain. ACM/IEEE 1st International Workshop on Emerging Trends in Software Engineering for Blockchain. p. 30-34. 2018

Batubara, F. R.; Ubacht J.; Janssen M. Challenges of Blockchain Technology Adoption for eGovernment: A Systematic Literature Review. 19th Annual International Conference on Digital Government Research. Article No. 76. 2018

Bdiwi, R.; Runz C.; Faiz S.; Cherif A. A. Towards a new Ubiquitous Learning Environment Based on Blockchain Technology. IEEE 17th International Conference on Advanced Learning Technologies. p. 101-102. 2017

Bistarelli S.; Manitacci M.; Santancini P.; Santini F. An End-to-end Voting-system Based on Bitcoin. SAC '17 Proceedings of the Symposium on Applied Computing. p. 1836-1841. 2017

Bore, N.; Karumba S.; Mutahi J.; Darnell S. S.; Wayua C.; Weldemariam K. Towards Blockchain-enabled School Information Hub. Association for Computing Machinery. p. 16-19. 2017

Calvaresi D.; Dubovitskaya A.; Retaggi D.; Dragoni A. F.; Schumacher M. Trusted Registration, Negotiation, and Service Evaluation in Multi-Agent Systems throughout the Blockchain Technology. Publication at: https://www.researchgate.net/publication/327382202

Castaldo L. and Cinque V. Blockchain-Based Logging for the Cross-Border Exchange of eHealth Data in Europe. Security in Computer and Information Sciences. p. 46–56. 2018.

Castelfranchi C. and Falcone R. 1998. Principles of trust for MAS: Cognitive anatomy, social importance, and quantification. In Proceedings of 3rd International Conference on Multi Agent Systems. 72–79.

CHAUM, David. Security without identification: Transaction systems to make big brother obsolete. Communications of the ACM, v. 28, n. 10, p. 1030-1044, 1985.

Dagher G. G.; Mohler J.; Milojkovic M.; Marella P. B. Ancile: Privacy-preserving frameworkforaccess control and interoperability of electronic health records using blockchain technology. Sustenable Cities and Society 39. p. 283-297. 2018

Diallo, N.; Shi W.; Xu L.; Gao Z.; Chen L.; Lu Y.; Shah N.; Carranco L.; Le T.; Surez A. B.; Turner G. eGov-DAO: a Better Government using Blockchain based Decentralized Autonomous Organization. International Conference on eDemocracy & eGovernment (ICEDEG) p. 166-171 .2018

DIFFIE, Whitfield; HELLMAN, Martin. New directions in cryptography. IEEE transactions on Information Theory, v. 22, n. 6, p. 644-654, 1976.

duPreez M. 2009. Trust and new technologies: Marketing and management on the Internet and Mobile media. Online Information Review. Vol. 33 Iss: 6, 1208–1209. Emerald Group Publishing Limited. eBay. 2015. http://www.ebay.com.

DWORK, Cynthia; NAOR, Moni. Pricing via processing or combatting junk mail. In: Annual International Cryptology Conference. Springer Berlin Heidelberg, 1992. p. 139-147.

Engelenburg S. V.; Janssen M.; Klieving B. Design of a software architecture supporting business-to-government information sharing to improve public safety and security. Journal of Intelligent Information Systems. p. 1–24. 2017

Feng L.; Zhang H.; Lou L.; Chen Y. A Blockchain-Based Collocation Storage Architecture for Data Security Process Platform of WSN. Proceedings of the 2018 IEEE 22nd International Conference on Computer Supported Cooperative Work in Design. p. 75-80. 2018

Franciscon E. A. 2019. A Systematic Literature Review of Blockchain Architectures Applied to the Government. CSCWD 2019: International Conference on Computer Supported Cooperative Work in Design.

FIPA, A. C. L. Fipa acl message structure specification. Foundation for Intelligent Physical Agents, http://www. fipa. org/specs/fipa00061/SC00061G. html (30.6. 2004), 2002.Huynh T.; N. Jennings; N. Shadbolt. An integrated trust and reputation model for open multi-agent systems. Springer Science, LLC 2006.

Ganache. One Click Blockchain. Homepage: https://www.trufflesuite.com/ganache

Gao Z.; Xu L.; Chen L.; Zhao L.; Lu Y. CoC: A Unified Distributed Ledger Based Supply Chain Management System. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY. p. 237–248. 2018

Gilad, Y.; Hemo R.; Micali S.; Vlachos G.; Zeldovich N. Algorand: Scaling Byzantine Agreements for Cryptocurrencies. SOSP '17 Proceedings of the 26th Symposium on Operating Systems Principles. p. 51-68. 2017

Goldwasser, S and Park, S. Public Accountability vs. Secret Laws: Can They Coexist?. WPES '17 Proceedings of the 2017 on Workshop on Privacy in the Electronic Society. p. 99-110. 2017

Granatyr J., V. Botelho, O. R. Lessing, E. E. Scalabrin, J.-P. Barth`es, and F. Enembreck, “Trustand reputation models for multiagent systems,” ACM Computing Surveys (CSUR), vol. 48, no. 2, p. 27, 2015.

Granatyr, Jones. MODELO AFETIVO DE CONFIANÇA E REPUTAÇÃO UTILIZANDO PERSONALIDADE E EMOÇÃO. Tese de doutorado. Pontifícia Universidade Católica do Paraná. Programa de Pós-Graduação em Informática. 2017. 150f.

Guo, R.; Shi H.; Zhao Q.; Zheng D. Secure Attribute-Based Signature Scheme With Multiple Authorities for Blockchain in Electronic Health Records Systems. IEEE Access. p. 11676- 11686. 2018

Heintze, T. and Bretschneider, S. Information Technology and Restructuring in Public Organizations: Does Adoption of Information Technology Affect Organizational Structures, Communications, and Decision Making? Journal of Public Administration Research and Theory. p. 801–830. 2000

Hou, H. The application of blockchain technology in E-government in China. 26th International Conference on Computer Communications and Networks (ICCCN). p. 1–4. 2017

Hussein A. F.; Kumar N. A.; Gonzalez G. R.; Tavares J. M. R. S.; Abdulhay E.; Albuquerque V. H. C. A medical records managing and securing blockchain based system by a Genetic Algorithm and Discrete Wavelet Transform. Cognitive Systems Research 52. p. 1–11. 2018

Huynh T. D.; Jennings N. R.; and Shadbolt N. R. 2004. FIRE: An integrated trust and reputation model for open multi-agent systems. In Proceedings of the 16th European Conference on Artificial Intelligence (ECAI). 18–22.

Jaffe, C.; Mata C.; Kamvar S. Motivating Urban Cycling Through a Blockchain-Based Financial Incentives System. Program in Media Arts and Sciences (Massachusetts Institute of Technology. p. 81-84. 2017

Jones Granatyr, Vanderson Botelho, Otto Robert Lessing, Edson Emílio Scalabrin, Jean-Paul Barthès, and Fabrício Enembreck. 2015. Trust and Reputation Models for Multiagent Systems. ACM Comput. Surv. 48, 2, Article 27 (November 2015), 42 pages. DOI:https://doi.org/10.1145/2816826

Jurca R. and Faltings B. 2003. An incentive compatible reputation mechanism. In Proceedings of the 6th International Workshop on Deception Fraud and Trust in Agent Societies (at AAMAS'03), ACM.

Kaijun, L.; Bi Y.; Jing L.; Fu H. Research on agricultural supply shain system with double chain architecture based on blockchain technology. Future Generation Computer Systems Volume 86. p. 641-649. 2018

KALISKI, Burt. PKCS# 7: Cryptographic message syntax version 1.5. 1998.

LESSING, O. R. ; WEFFORT, M. R. ; GRANATYR, J. ; FRANCISCON, E. A. ; SCALABRIN, E. E. . Decision-Making: From Pure Uncertainty to Measurable Risk. In: 2019 IEEE 23rd International Conference on Computer Supported Cooperative Work in Design, 2019, Porto - Portugal. Proceedings of the 2019 IEEE 23rd International Conference on Computer Supported Cooperative Work in Design. Danvers: IEEE, 2019. v. 1. p. 170-175.

Lopes, A. C. F. (2006) Um método para a geração de estimativas de reputação mais precisas perante a oscilação de comportamento das entidades avaliadas. Dissertação de Mestrado (Programa de Pós-Graduação em Computação). Universidade Federal Fluminense, Niterói-RJ.

Lu G.; Lu J.; Yao S. and Yip J. 2009. A review on computational trust models for multi-agent systems. In International Conference on Internet Computing, pp. 325–331.

Magnani, A.; Calderoni L.; Palmieri P. Feather forking as a positive force: incentivising green energy production in a blockchain-based smart grid. CryBlock'18. Proceedings of the 1st Workshop on Cryptocurrencies and Blockchains for Distributed Systems. p. 99-104. 2018

Malomo O. O.; Rawat D. B.; Garuba M. Next-generation cybersecurity through a blockchain-enabled federated cloud framework. The Journal of Supercomputing. p. 5099–5126. 2018

Margheri A.; Ferdous Md. S.; Yang M.; Sassone V. A Distributed Infrastructure for Democratic Cloud Federations. IEEE 10th International Conference on Cloud Computing. p. 688-691. 2017

Marsh S. Formalising trust as a computation concept – University of Stirling, 1994.

Mercado Livre: Loja mercado livre. Homepage: https://www.mercadolivre.com.br/

MERKLE, Ralph C. A digital signature based on a conventional encryption function. In: Conference on the Theory and Application of Cryptographic Techniques. Springer Berlin Heidelberg, 1987. p. 369-378.

MetaMask. Brings Ethereum to your browser. Homepage: https://www.trufflesuite.com/

NAKAMOTO, Satoshi. Bitcoin: A peer-to-peer electronic cash system. 2008. OMOHUNDRO, Steve. Cryptocurrencies, smart contracts, and artificial intelligence. AI Matters 1, 2 (December 2014), 19-21. DOI=http://dx.doi.org/10.1145/2685328.2685334

Niya, S. R.; Jha S.; Bocek T.; Stiller B. Design and Implementation of an Automated and Decentralized Pollution Monitoring System with Blockchains, Smart Contracts, and LoRaWAN. IEEE/IFIP Network Operations and Management Symposium, At Taipei-Taiwan. 2018

Noerlina *et al*., "Development of a Web Based Corruption Case Mapping Using Machine Learning with Artificial Neural Network," *2018 International Conference on Information Management and Technology (ICIMTech)*, Jakarta, 2018, pp. 400-405.

Otte P., et al., TrustChain: A Sybil-resistant scalable blockchain, Future Generation Computer Systems (2017), http://dx.doi.org/10.1016/j.future.2017.08.048.

Parino, F., Beiró, M.G. & Gauvin, L. EPJ Data Sci. (2018) 7: 38. https://doi.org/10.1140/epjds/s13688-018-0170-8

React. Uma biblioteca JavaScript para criar interfaces de usuário. Homepage: https://pt-br.reactjs.org/

Roehrs A.; Costa C. A.; Righi R. R.; OmniPHR: A distributed architecture model to integrate personal health records. Journal of Biomedical Informatics 71. p.70-81. 2017

SAATY, T. L. “The Analytic Hierarchy Process.” McGraw-Hill, New York. 1980.

SAATY, T. L. Decision Making with Dependence and Feedback: the Analytic Network Process, RWS Publications, Pittsburgh (USA). 1996.

Sabater J. and Sierra C. 2005. Review on computational trust and reputation models. Artificial Intelligence Review 24, 1, 33–60.

Sabater J. and Sierra C. 2002. Reputation and social network analysis in multi-agent systems. In Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS). 475–482, Bologna, Italy.

Satoshi Nakamoto. Bitcoin: A Peer-to-Peer Electronic Cash System. 2008. Link: https://bitcoin.org/bitcoin.pdf

Seuken S., D.C. Parkes, Sybil-proof accounting mechanisms with transitive trust, in:Proceedings of the 2014 International Conference on Autonomous Agents and Multi-Agent Systems, International Foundation for Autonomous Agents and Multiagent Systems, 2014, pp. 205–212.

Sharma P. K. and Park J. H. Blockchain based hybrid network architecture for the smart city. Future Generation Computer Systems 86. p. 650-655. 2018

Shae, Z and Tsai, J. J. P. On the Design of a Blockchain Platform for Clinical Trial and Precision Medicine. IEEE 37th International Conference on Distributed Computing Systems. p. 1972-1980. 2017

Shaheen, S. H.; Yousaf M.; Jalil M. Temper Proof Data Distribution for Universal Verifiability and Accuracy in Electoral Process Using Blockchain. 13th International Conference on Emerging Technologies (ICET). p. 1-6. 2017

Sharma P. K. and Park J. H. Blockchain based hybrid network architecture for the smart city. Future Generation Computer Systems 86. p. 650-655. 2018

Silva B. V. Um modelo de confiança certificado baseado em assinatura digital aplicado a sistemas multiagentes Dissertação (mestrado) – Pontifícia Universidade Católica do Paraná, Curitiba, 2009.

Silva B. V. DOSSIÊ: MODELO DE CONFIANÇA PARA SISTEMAS MULTIAGENTES. Tese de Doutorado apresentada ao Programa de Pós-Graduação em Informática da Pontifícia Universidade Católica do Paraná. 2017.

Sommer T.; Deppe G.; Stehling V.; Haberstroh M.; Hees F. Request for Comments: Proposal of a Blockchain for the Automatic Management and Acceptance of Student Achievements. E-Prüfungs-Symposium 13. und 14. RWTH Aachen. 2018

Submarino: Loja submarino. Homepage: https://www.submarino.com.br/

SWAN, Melanie. Blockchain: Blueprint for a new economy. O'Reilly Media, Inc., 2015.

Szabo. N. (1996). Smart Contracts: Building Blocks for Digital Markets. Available: http://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/smart_contracts_2.html

Theodouli A. Arakliotis S.; Moschou K.; Votis K.; Tzovaras D. On the design of a Blockchain-based system to facilitate Healthcare Data Sharing. 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/ 12th IEEE International Conference On Big Data Science And Engineering. p. 1374-1379. 2018

Truffle Suite. SWEET TOOLS FOR SMART CONTRACTS. Homepage: https://www.trufflesuite.com/

Uddin. MD A.; Stranieri A.; Gondal I.; Balasubramanian V. Continuous Patient Monitoring With a Patient Centric Agent: A Block Architecture. IEEE Access. p. 32700-32726. 2018

Wang B.; Sun J.; He Y.; Pang H.; Lu N. Large-scale Election Based On Blockchain. Procedia Computer Science 129. p. 234–237. 2018

Wang, L.; Liu W.; Han X. Blockchain-Based Government Information Resource Sharing. IEEE 23rd International Conference on Parallel and Distributed Systems. p. 804-809. 2017

Xia, Q.; Sifah E. B.; Asamoah K. O.; Gao J.; Du X.; Guizani M. MeDShare: Trust-Less Medical Data Sharing Among Cloud Service Providers via Blockchain. IEEE Access. p. 14757-14767. 2017

Xie W.; Zhou W.; Kong L.; Zhang X.; Min X.; Xiao Z.; Li Q. ETTF: A Trusted Trading Framework Using Blockchain in E-commerce. Proceedings of the 2018 IEEE 22nd International Conference on Computer Supported Cooperative Work in Design. p. 612-617. 2018

Xu L.; Chen L.; Gao Z.; Lu. Y.; Shi W. CoC: Secure Supply Chain Management System based on Public Ledger. 26ª Conferência Internacional sobre Comunicação e Redes de Computadores (ICCCN). 2017

Yue, X. Jin D.; Wang H.; Li M. Healthcare Data Gateways: Found Healthcare Intelligence on Blockchain with Novel Privacy Risk Control. Springer Science Business Media New York. 2016

Zhao Y.; Tan W.; Zhao L. Blockchain-Based UDDI Data Replication and Sharing. Proceedings of the 2018 IEEE 22nd International Conference on Computer Supported Cooperative Work in Design. p. 384-389. 2018

Zhang P.; White J.; Schmidt D. C.; Lenz G.; Rosenbloon S. T. FHIRChain: Applying Blockchain to Securely and Scalably Share Clinical Data. Computational and Structural Biotechnology Journal 16. p. 267–278. 2018.