PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO GRANDE DO SUL

ESCOLA POLITÉCNICA

ENGENHARIA DE COMPUTAÇÃO

Lucas Cadore Nardi

APLICAÇÃO DE UM BLOCKCHAIN PRIVADO E CONTRATOS INTELIGENTES EM UM SOFTWARE DISTRIBUÍDO DE CONTROLE OPERACIONAL INDUSTRIAL

Porto Alegre

2022

LUCAS CADORE NARDI

APLICAÇÃO DE UM BLOCKCHAIN PRIVADO E CONTRATOS INTELIGENTES EM UM SOFTWARE DISTRIBUÍDO DE CONTROLE OPERACIONAL INDUSTRIAL

Trabalho de conclusão de curso de graduação apresentado na Escola Politécnica da Pontifícia Universidade Católica do Rio Grande do Sul, como requisito parcial para obtenção do grau de Engenheiro de Computação.

Orientador: Anderson Royes Terroso

Porto Alegre

2022

Dedico este trabalho aos meus pais, pois sem seu apoio e dedicação constante, eu nunca teria chegado até este momento.

AGRADECIMENTOS

Agradeço ao meu orientador, professor Anderson Royes Terroso, por sua disponibilidade e ajuda durante o desenvolvimento deste trabalho. As empresas, que forneceram informações suficientes para possibilitar a execução deste trabalho. Aos meus pais, por terem me proporcionado ensino, carinho e suporte durante todo o curso e o desenvolvimento deste trabalho. A minha namorada, por todo apoio e motivação durante toda a graduação.

RESUMO

O avanço da Indústria 4.0 têm trazido inúmeros benefícios aos mais diversos processos industriais. Em contrapartida, a inserção de tecnologias de natureza estrangeira ao contexto industrial, introduziram novos problemas em relação à segurança e privacidade de sistemas industriais. Dentre as principais propostas à solução destes problemas, a Tecnologia de Registros Distribuídos (DLT) é a mais proeminente. Portanto, este trabalho tem como objetivo propor, com base em referências bibliográficas e no modelo arquitetural da Web 3.0 de Kasireddy (2021a), um modelo arquitetural para a implementação de DLTs no contexto da Indústria 4.0, sendo a DLT escolhida o *blockchain* do *Ethereum*. Estudou-se a viabilidade de implementação deste modelo através da implementação de melhorias funcionais e da análise de dados quantitativos da atual rede pública do *Ethereum*. A partir desta análise, constatou-se melhorias no contexto de segurança, privacidade e funcionalidade, porém existindo a necessidade de maior maturação e estudo às tecnologias utilizadas.

Palavras-chave: Indústria 4.0; Tecnologia de Registros Distribuídos; Web 3.0; *Blockchain*.

ABSTRACT

The advancement of Industry 4.0 has brought numerous benefits to the most diverse industrial processes. On the other hand, the insertion of technologies of a foreign nature to the industrial context, introduced new problems in relation to the security and privacy of industrial systems. Among the main proposals to solve these problems, the Distributed Ledger Technology (DLT) is the most prominent. Therefore, this monography aims to propose, based on bibliographic references and the architectural model of Web 3.0 created by Kasireddy (2021a), an architectural model for the implementation of DLTs in the context of Industry 4.0, with the chosen DLT being the Ethereum blockchain. The feasibility of implementing this model was studied through the implementation of functional improvements and the analysis of quantitative data from the current public Ethereum network. From this analysis, improvements were found in the context of security, privacy and functionality, but there is still a need for greater maturation and study of the technologies used.

***Keywords**: Industry 4.0; Distributed Ledger Technology; Web 3.0; Blockchain.*

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE SIGLAS

ABI – *Application Binary Interface* ou Interface Binária de Aplicação

API – *Application Programming Interface* ou Interface de Programação de Aplicações

B2B – *Business-to-Business* ou interação empresa-empresa

BMBF – *Bundesministerium für Bildung und Forshung* ou Ministério Federal da Educação e Pesquisa da Alemanha

CPS – *Cyber-Physical Systems* ou Sistemas cyber-físicos

DApps – *Descentralized Applications* ou Aplicações Descentralizadas

DApps – *Descentralized Applications* ou aplicações descentralizadas

DLT – *Distributed Ledger Technology* ou Tecnologia de Registros Distribuídos

EVM – *Ethereum Virtual Machine* ou Máquina Virtual do *Ethereum*

GAF – Grafos Acíclicos Dirigidos

HTTP – *Hypertext Transfer Protocol*

IA – Inteligência Artificial

IOT – *Internet of Things* ou Internet das Coisas

IPFS – *InterPlanetary File System* ou Sistema de Arquivos Interplanetário

JSON – *Javascript Object Notation* ou Objetos de Notação *JavaScript*

JSON RPC – Procedimento remoto encapsulado em JSON

JWT – JSON Web *Token*

MFA – *Multifactor Authentication* ou Autenticação Multifator

NSF – *National Science Foundation* ou Fundação Nacional da Ciência

OTP – *One-time Password* ou senha de uso único

P2P – *Peer-to-peer* ou Ponto-a-Ponto

PoA – *Proof-of-authority* ou prova de autoridade

PoS – *Proof-of-stake* ou prova de *stake*

PoW – *Proof-of-work* ou prova de trabalho

REST – *Representational State Transfer* ou Transferência Representacional de Estado

RFID – Identificadores por radiofrequência

ROM – *Read Only Memory* ou memória restrita a leitura

RSA – Rivest-Shamir-Adelman

SPA – *Single Page Application* ou aplicação de página única

SSO – *Single Sign On* ou autenticação única

WEF – *World Economic Forum* ou Fórum Econômico Mundial

SUMÁRIO

1 INTRODUÇÃO

Os exponenciais avanços tecnológicos das últimas duas décadas têm transformado os mais diversos setores da economia mundial, sendo um dos seus reflexos a quarta revolução industrial, conhecida também como Indústria 4.0 (SCHWAB, 2015). Esta, é caracterizada pela maturação de sistemas cyber-físicos (CPS – *Cyber-physical Sistems*) e sua inserção no meio industrial, o qual tem o potencial de desconstruir e otimizar os mais diversos processos de produção, gestão e governança (KAGERMANN; WAHLSTER; HELBIG, 2015).

O termo "sistemas cyber-físicos" surgiu nos Estados Unidos da América em 2006, sendo cunhado por Helen Gill na Fundação Nacional da Ciência (NSF – *National Science Foundation*) (LEE; SESHIA, 2015, p. 5). Entretanto, sua definição na comunidade científica tem variado durante os anos perante as diferentes perspectivas sobre a tecnologia (GUNES et al., 2014). Sendo a definição de Lee e Seshia (2015, p.1) a mais aceita, a qual dita que:

> Um sistema cyber-físico é uma integração da computação com processos físicos cujo comportamento é definido pelas partes cibernéticas e físicas do sistema. Computadores e redes embarcados monitoram e controlam os processos físicos, geralmente com *loops* de *feedback* onde os processos físicos afetam os cálculos e vice-versa.

Porém, é válido ressaltar que a descrição do termo não define uma tecnologia, mas sim um princípio funcional. Deste modo, diferentes conceitos e tecnologias, que compartilham semelhanças em sua descrição, acabam sendo considerados partes essenciais de um CPS, alguns destes sendo o *Big Data*, Internet das Coisas (IOT – *Internet of Things*), Computação em Nuvem e a Inteligência Artificial (IA) (GUNES et al., 2014).

A existência de um ecossistema industrial com grande acessibilidade a sensores, sistemas de coleta de dados, redes de computadores e máquinas com interface de rede, possibilitou uma rápida adoção aos CPSs. O seu principal objetivo é a criação de sistemas inteligentes, resilientes, eficientes e auto adaptáveis (LEE; BAGHERI; KAO, 2014). Esta rápida adoção possibilitou a maturação de diversas tecnologias essenciais aos CPSs, como também gerou um grande interesse na adesão à Indústria 4.0. No entanto, a característica em maturação destes, trouxe à tona a falta de consenso nas definições arquiteturais, de design e modelagem, como também os desafios que surgem à adição de tecnologias estrangeiras ao contexto industrial (JAMAI; AZZOUZ; SAIDANE, 2020).

Dentre os diversos novos desafios aparentes a Indústria 4.0, a segurança e privacidade dividem o centro das atenções (LEE; SESHIA, 2015, p. 460). Estes sendo externalizados pela

abrangência de superfícies vulneráveis a ataques, advindos da crescente interconectividade dos sistemas industriais entre si ou a internet (JAMAI; AZZOUZ; SAIDANE, 2020). Surgem assim, inúmeras propostas para a sua solução ou atenuação, os quais variam desde melhorias arquiteturais a agregações tecnológicas. Entretanto, a proposta mais proeminente na atualidade é a aplicação de Tecnologias de Registros Distribuídos (DLT – *Distributed Ledger Technology*) aos novos sistemas industriais (JAVAID et al., 2021).

A tecnologia de registros distribuídos já tem forte impacto nos setores financeiros e logísticos, porém, no contexto da Indústria 4.0 o processo ainda é jovem (ALLADI et al., 2019). Segundo Alladi *et al*. (2019) a aplicação de DLTs tende a continuar expandindo durante os próximos anos, motivado principalmente pelas potenciais soluções para os problemas de segurança e privacidade. Entretanto, este traz consigo seus próprios desafios e necessita, para atingir seu potencial, de mais pesquisas e tempo para evolução.

Sendo assim, este trabalho pretende avaliar de forma prática a atual viabilidade da implementação de DLTs no contexto da Indústria 4.0, como também, quais desafios esta tecnologia consegue solucionar de maneira eficiente.

2 FUNDAMENTAÇÃO TEÓRICA

Nesta seção serão abordadas explicações teóricas sobre os diversos temas presentes neste trabalho.

2.1 INDÚSTRIA 4.0

A rápida expansão tecnológica das últimas duas décadas causou um grande impacto nos paradigmas sociais e econômicos, sendo um de seus reflexos a quarta revolução industrial. Antes mesmo de se definir a quarta revolução industrial, deve-se compreender os motivos que a distinguem de uma extensão a atual presente revolução industrial (SCHWAB, 2015).

A terceira revolução industrial teve seu início na década de 50 e tem como expectativa atingir seu fim entre 2030 e 2040, sendo popularmente conhecida como a "Revolução Digital". As duas principais características desta revolução são o grande crescimento da eficiência energética e a transição de eletrônicos analógicos para digitais. Diferentemente das duas revoluções anteriores, esta teve um alcance muito mais descentralizado, tendo impactos em nível mundial rapidamente. Sendo algumas de suas inovações os transistores, computadores e a internet (MOHAJAN, 2021).

Os processos de revolução industrial não são homogêneos, ou seja, enquanto partes do mundo já estão na terceira revolução industrial outras ainda passam pela segunda. De mesmo modo, toda e qualquer revolução industrial não é caracterizada somente pela introdução de novas tecnologias e pelos ganhos operacionais no processo produtivo, mas também, pelas grandes mudanças estruturais na sociedade e economia (POPKOVA; RAGULINA; BOGOVIZ, 2018).

Sendo assim, as primeiras menções a uma possível ocorrência de uma nova revolução industrial foram introduzidas, de maneira estruturada, em 2011 através da pesquisa "Recomendações para a implementação da iniciativa estratégica Indústria 4.0", patrocinada pelo Ministério Federal da Educação e Pesquisa da Alemanha (BMBF – *Bundesministerium für Bildung und Forshung*). Esta pesquisa buscava introduzir recomendações para o setor industrial alemão se adaptar a um possível emergente modelo de industrialização, de modo a manter seu papel de liderança industrial em relação aos países concorrentes. Esta pesquisa propunha que o novo modelo de industrialização seria caracterizado pela implementação de redes globais de sistemas, instalações e máquinas, através da utilização de CPSs (KAGERMANN; WAHLSTER; HELBIG, 2013).

Entretanto, somente quatro anos depois, durante o Fórum Econômico Mundial (WEF – *World Economic Forum*) de 2015, Klaus Schwab popularizou e distinguiu estas recomendações a Indústria 4.0 como uma nova revolução industrial em pleno movimento (POPKOVA; RAGULINA; BOGOVIZ, 2018).

Independentemente do crescimento exponencial da aplicação e estudo a Indústria 4.0 nos últimos anos, ainda se faz presente a falta de consenso em relação a sua arquitetura e definição. Mesmo assim, todas definições convergem no embasamento tecnológico, o qual dita que o coração da Indústria 4.0 é a integração e implementação de sistemas *cyber*-físicos e suas tecnologias agregadas ao meio industrial (JAMAI; AZZOUZ; SAIDANE, 2020).

2.2 SISTEMAS CYBER-FÍSICOS

A definição apresentada na introdução deste trabalho é válida para este termo, porém deve-se ressaltar a definição de CPSs feita por Gunes *et al.* (2014), a qual pretende compilar as diferentes definições de CPSs ao longo do tempo:

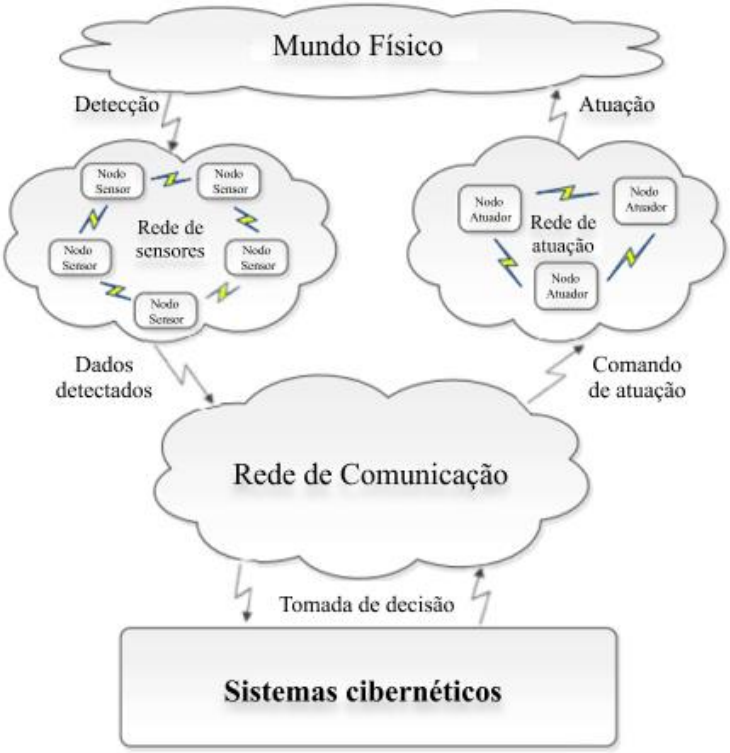
> Os sistemas cyber-físicos são sistemas de engenharia de próxima geração complexos, multidisciplinares e fisicamente conscientes que integram a tecnologia de computação embarcada (parte cibernética) aos fenômenos físicos usando abordagens de pesquisa transformadoras. Essa integração inclui principalmente aspectos de observação, comunicação e controle dos sistemas físicos a partir de uma perspectiva multidisciplinar.

É possível notar que em ambas as definições não são expostos delimitadores tecnológicos ou funcionais, possibilitando assim, correlações entre o termo e tecnologias altamente especificadas. Dentre estas correlações, algumas são consideradas como essenciais ou básicas a aplicação de CPS, sendo algumas destas, a Indústria 4.0, *Big Data* e a Internet das Coisas (LEE; SESHIA, 2015, p. 5).

A grande gama de tecnologias que podem compor CPSs, dificultam a criação de modelos arquiteturais e funcionais, os quais sejam genéricos. Entretanto, a introdução de certas abstrações possibilitou Gunes *et al.* (2014) propor um modelo genérico de CPS que pode ser visualizado na Figura 1. Os sistemas físicos e cibernéticos operam entre si através de redes de sensores, que coletam e interpretam os dados e os distribuem através de uma rede de comunicação. A rede de comunicação, deve ter capacidade de enviar os dados para os sistemas cibernéticos e comunicar para a rede de atuadores, a ação que estes devem tomar. Os atuadores

presentes dentro da rede de atuadores, interpretam a ação e afetam o meio físico através de comandos e modelo de escolha.

Figura 1 – Diagrama de alto nível arquitetural do sistema descentralizado



Fonte: Gunes *et al.*, 2014.

2.2.1 Desafios e limitadores de CPS

A natureza de atuação de CPSs, os quais inevitavelmente controlam ou obtêm dados do meio físico, traz consigo inúmeros desafios de implementação e execução. Alguns destes podendo ser a necessidade de garantias temporais, implementação de algoritmos de tratamento de falhas e até mesmo a representação de maneira uniforme dos dados obtidos (GUNES *et al.*, 2014). Sobreposto a estes desafios de cunho operacional, o presente processo de maturação expõe também desafios na adaptabilidade, unificação, dependabilidade e consistência de CPS (HU; VASQUEZ; PATTERSON, 2014).

Somente o tempo, através do contínuo estudo e aplicação de CPS, possibilitará medir e encontrar soluções para estes desafios expostos. Entretanto, alguns desafios são privados do tempo necessário para esta maturação e necessitam resoluções o quanto antes possível, sendo estes, os desafios de segurança e privacidade (LEE; SESHIA, 2015, p. 460). Sendo assim, a resolução destes desafios é inerente ao sucesso da implementação de CPSs à indústria e ao avanço da Indústria 4.0 como um todo (KAGERMANN; WAHLSTER; HELBIG, 2013).

O desafio de segurança dita a dificuldade de se criar um estado sistêmico seguro a qualquer tipo de dano externo, enquanto o desafio de privacidade dita a dificuldade de manter um estado sistêmico afastado de observações externas (LEE; SESHIA, 2015, p. 460). Ambos são fortemente correlacionados à introdução de uma maior interconexão entre os meios físicos e cibernéticos através da Internet, tecnologias agregadas, como o IOT e a Computação em Nuvem (*Cloud Computing*), e por novos meios de comunicação. Deste modo, cria-se uma maior superfície industrial propícia a ataques externos (JAMAI; AZZOUZ; SAIDANE, 2020). Estes ataques podem variar fortemente em sua complexidade e objetivo, alguns recentes exemplos são o ataque à *Colonial Pipeline Company* em maio de 2021 e o *malware* Triton descoberto em 2017.

O ataque à empresa *Colonial Pipeline Company*, a qual é responsável pelo oleoduto americano que transporta gasolina e combustíveis de avião entre Houston, Texas, e o sudeste americano, visou tomar controle dos sistemas de controle operacional do oleoduto e obter dados confidenciais do sistema (GONZALEZ; LEFEBVRE; GELLER, 2021). O grupo de *hackers* conhecido como *DarkSide* foi responsável pelo ataque e teve êxito, obtendo cerca de 100 *gigabytes* de dados confidenciais além de bloquear o funcionamento do oleoduto por seis dias, o que acarretou picos de preços e faltas de combustíveis em todo o território americano (THORBECKE, 2021) (ROBERTSON; TURTON, 2021).

O *malware* Triton foi encontrado em uma indústria petroquímica na Arábia Saudita em junho de 2017, e teve como objetivo inibir o funcionamento de controladores de segurança. Através deste *malware*, os controladores puderam ter seus valores adulterados e até mesmo foram desligados por completo, sem notificação ao sistema industrial. Este *malware* pode causar danos catastróficos ao patrimônio industrial e as vidas dos colaboradores, visto que falhas no sistema de controle de segurança em indústrias podem causar por exemplo a vasão de gases tóxicos, como também explosões (GILES, 2019).

Ambos os ataques representam dois exemplos em uma imensidão de ataques que acontecem quase que diariamente, porém em diferentes níveis de complexidade e perigo. Reforçando a necessidade de teste, implementação e validação de diferentes métodos para a sua solução, sendo um dos mais proeminentes a inserção da Tecnologia de Registros Distribuídos ao contexto da Indústria 4.0 (ALLADI *et al.*, 2019).

2.3 TECNOLOGIA DE REGISTROS DISTRIBUÍDOS

As Tecnologias de Registros Distribuídos visam definir protocolos e princípios funcionais que possibilitem a criação de sistemas distribuídos totalmente descentralizados, através de algoritmos criptográficos de consenso distribuído, que garantam a imutabilidade, transparência e confiabilidade dos seus dados. Esta é considerada recentemente, impulsionada pela popularização de *blockchains* e criptomoedas, como uma das mais importantes inovações do século 21 (CHOWDURY *et al.*, 2019).

O conceito de descentralização não é algo novo, porém o seu funcionamento depende, na maioria das vezes, de algum órgão central. As DLTs conseguem criar sistemas realmente descentralizados a partir da eliminação *a priori* da Falha Bizantina, a qual, de maneira simplificada, tem origem no conceito de que sistemas computacionais não são capazes de lidar de maneira efetiva com informações conflitantes ou ambientes adversos (LAMPORT *et al.*, 1982).

Independentemente da popularidade do uso recíproco entre DLTs e *blockchains* na literatura, estes não são iguais. *Blockchains* são uma das possíveis tecnologias presentes dentro das diferentes DLTs. Esta confusão é causada pois, a primeira aplicação prática de uma DLT a se tornar popular foi um *blockchain*, este surgindo em 2009 e sendo chamado de Bitcoin (CHOWDURY *et al.*, 2019).

Deste modo, deve-se salientar a definição formal realizada por Rauchs *et al.* (2018), a qual dita que:

> DLTs são sistemas de registros eletrônicos que permitem que uma rede de participantes estabeleça um consenso em torno do ordenamento autoritário de transações criptograficamente validadas (ou “assinadas”). Esses registros são persistentes, pois replicam os dados em vários nós da rede, e invioláveis, através da vinculação por *hashes* criptográficos entre todo e qualquer registro. O resultado compartilhado do processo de reconciliação/consenso serve como a versão oficial para esses registros.

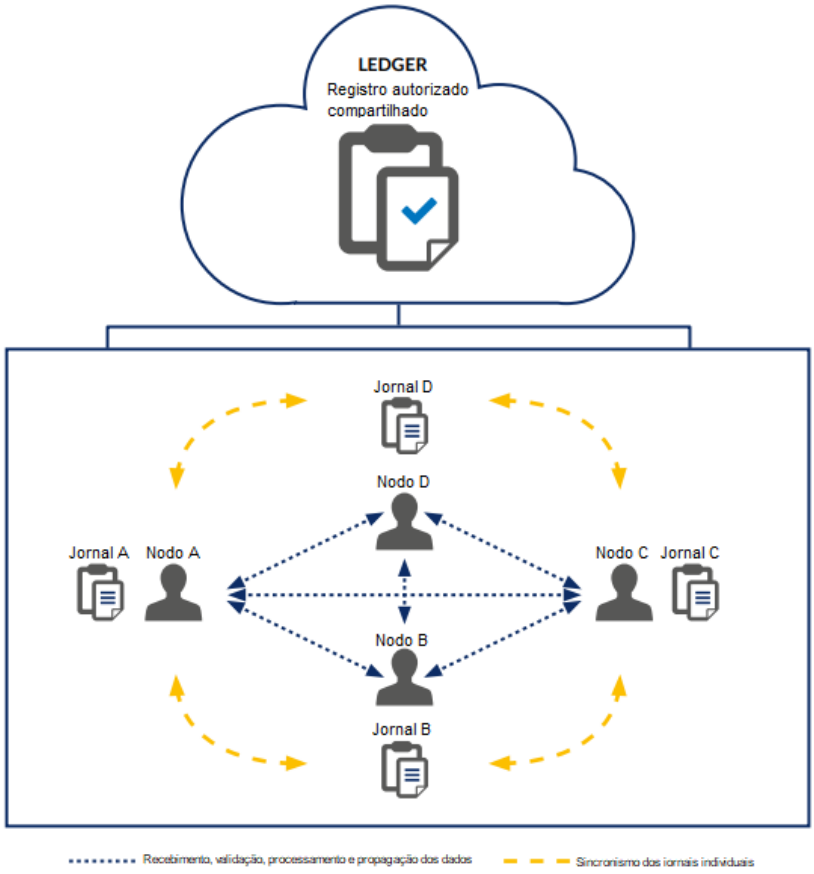
2.3.1 Terminologia comum

A terminologia utilizada para definir diversas etapas e processos de DLTs, por muitas vezes, é ambígua. Fazendo-se necessário a definição de alguns dos principais conceitos e termos. Sendo assim, Rauchs *et al.* (2018) define que:

a) *Ledger* (ou livro-razão) é o conjunto de registros autorizados mantidos pela maioria dos nodos (ou participantes) da rede, em qualquer momento qual que os registros sejam improvavelmente deletados ou alterados.
b) Transação é qualquer proposta de modificação ou adição ao *ledger*. Por mais que a palavra denote um sentido de transação financeira, este não necessita envolver nenhum tipo de custo ou moeda.
c) *Logs* (ou histórico) é um conjunto desordenado de transações válidas dentro de um nodo da rede, as quais, ainda não foram validadas pelo consenso distribuído.
d) Registro é todo dado de transações que já foram validadas e distribuídas por toda a rede.
e) Jornal é o conjunto de registros mantidos dentro de um nodo da rede, que pode ou não ser consistente com o *ledger*. Comumente jornais são parciais, provisionais e heterogêneos.

Estas definições teóricas podem parecer de certo modo abstratas, mas serão muito úteis quando for definido tecnicamente o funcionamento de DLTs, como o Bitcoin. Sendo assim, é possível representar estes conceitos de modo visual e conciso, através da Figura 2.

Figura 2 – Diagrama de representação dos conceitos de DLTs



Fonte: Traduzido de Rauchs *et al.*, 2018.

2.3.2 Anatomia sistêmica

O processo contínuo de evolução das DLTs faz necessária a implementação de um padrão genérico de elementos suficientes para estas, passando pela delimitação de camadas funcionais, elementos operacionais e métodos de comunicação. Deste modo, Rauchs *et al.* (2018) definiu que DLTs podem ser divididas em três camadas funcionais, sendo estas a camada de protocolo, rede e dados.

A camada de protocolo é a fundação de todo DLT, sendo responsável pelo conjunto de regras definidas por software que determinam a operação do sistema. O protocolo aplicado é constituído de dois componentes, o componente de gênese e o componente de alterações. O componente de gênese, como o seu próprio nome define, é instanciado em tempo de criação da DLT e define a base de código e uma arquitetura funcional inicial. Este componente define consigo a primeira transação da rede, comumente chamada de transação de gênese. O componente de alterações define os aspectos de governança como algumas definições prévias de evolução do sistema (RAUCHS *et al.*, 2018).

A camada de rede é responsável pela implementação prática dos protocolos, definindo as operações de validação, autenticação, processamento e compartilhamento de dados entre todos os participantes da rede. Esta é composta por três componentes, sendo estes o componente de comunicação, processamento de transação e de validação. O componente de comunicação específica modelos de acesso, compartilhamento de dados e autorização. O componente de processamento de transações define as permissões de transação como todos os participantes desta rede entram em consenso. O componente de validação valida as transações da rede em sua conformidade com o protocolo implementado, pois a transação não deve conter dados falsos, inválidos ou absentes (RAUCHS *et al.*, 2018).

A camada de dados é responsável pelo processamento das informações e pelo tratamento dos dados gerados no sistema, e que formam o ledger. Esta consiste em dois componentes, sendo estes, o componente de operações e o jornal. O componente de operações é composto pelos processos que criam e modificam registros. Enquanto o jornal é composto pelo conteúdo armazenado em cada registro (RAUCHS et al., 2018). A Figura 3 representa as três camadas e seus componentes, de maneira simplificada.

Figura 3 – Diagrama de representação simplificada da camada de dados de DLTs

Fonte: Compilação realizada pelo autor, obtida de RAUCHS et al., 2018.

2.3.3 Blockchains

Atualmente, os principais *blockchains* são o *Bitcoin* e o *Ethereum*, porém inúmeros outros *blockchains* existem de maneira paralela, tentando solucionar diferentes problemas. É importante salientar que *blockchains* compartilham conceitos tecnológicos entre si, porém seu comportamento, protocolos e uso são extremamente variados, dificultando assim uma definição aprofundada que seja geral (REIFF, 2022). Isso ficará mais claro, quando forem descritos a fundo o *Bitcoin* e o *Ethereum*.

Independentemente, é possível generalizar *blockchains* a partir da sua semelhança funcional e da etimologia do termo. O nome *blockchain* surge do seu princípio básico fundamental, onde os dados deste sistema são registrados em blocos (*block*) de maneira consecutiva, o qual, cada novo bloco possui uma referência criptografada ao bloco anterior, formando assim uma corrente (*chain*) de dados. Este registro de dados é distribuído e armazenado pelos nodos de uma rede ponto-a-ponto (P2P – *peer-to-peer*), onde a cada novo bloco criado, este é validado através de um algoritmo de consenso distribuído e transmitido a todos os participantes desta rede (CHOWDURY et al., 2019).

2.3.4 Outras tecnologias de DLTs

No contexto deste trabalho, não serão abordados outros tipos de DLTs, porém é importante salientar, mesmo que de maneira sucinta, a presença destes. Além de *blockchains*, os outros DLTs mais proeminentes são os Grafos Acíclicos Dirigidos (GAD), *Hashgraphs*, *Holochain* e o Tempo (*Radix*). Cada uma destas DLTs opera de maneira diferente, mas seu propósito é o mesmo, a criação de um sistema descentralizado utilizando algoritmos de consenso distribuído (GERONI, 2020).

2.4 BITCOIN

O *Bitcoin* surge em 2009, fortemente impulsionado pelos efeitos da crise financeira de 2008, propondo uma solução para o problema do gasto duplo em transações financeiras digitais, sem a necessidade de um órgão central regulador. Esta solução foi fortemente influenciada pelo artigo publicado em 2002, por Adam Back, nomeado de "*Hashcash – A Denial of Service Counter-Measure*" e pela proposta publicada em 1998, por Wei Dai, conhecida como "*B-money*" (NAKAMOTO, 2009). O criador, ou criadores, do Bitcoin adotou o pseudônimo de Satoshi Nakamoto e continua até os dias de hoje sendo um mistério (HAAR, 2022).

A solução para o problema do gasto duplo, proposta por Nakamoto (2009), se fundamenta em um sistema de pagamentos eletrônicos utilizando provas criptográficas ao contrário da confiança mútua, eliminando assim, a necessidade de um intermediador para as transações. O sistema utilizaria um servidor de registros de data e hora (*timestamps)* distribuído e P2P, o qual gera provas computacionais em ordem cronológica e possibilita transações através do uso de uma moeda digital, também chamada de *Bitcoin* (ALBUQUERQUE; CALLADO, 2015)

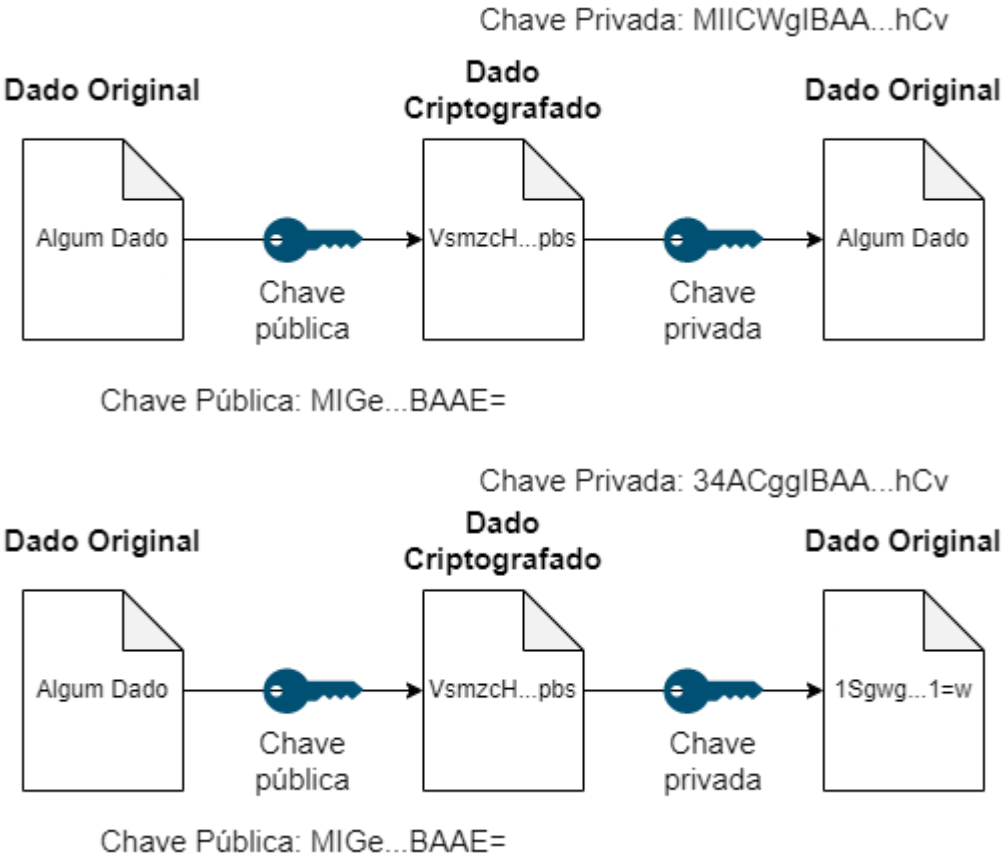
2.4.1 Criptografia de chave pública

Criptografia de chave pública é um tipo de criptografia comumente baseado em algoritmos de Rivest-Shamir-Adelman (RSA), o qual apresenta um par de chaves que possibilitam a autenticação ou assinatura de dados. Este par é constituído de uma chave pública (*public key*) e uma chave privada (*private key*) (IBM, 2021).

Este tipo de criptografia funciona a partir da criptografia dos dados, conhecido também como um *hash*, através da chave pública. Este *hash* só poderá ser descriptografado utilizando a

chave privada correspondente a chave pública utilizada para a criptografia. Caso a chave privada não corresponda com a chave pública, não há bloqueio de descriptografia, porém o dado não será correspondido ao original e será duplamente criptografado. Se em um futuro este dado duplamente, ou enésimas vezes criptografado, for pareado com a correta chave privada o dado original será encontrado (IBM, 2021). Este comportamento é visível na Figura 4.

Figura 4 – Diagrama de funcionamento da criptografia RSA



Fonte: Diagrama compilado pelo próprio autor baseado nos dados presentes em IBM, 2021.

A criptografia de ordem reversa também é possível, mesmo que não seja indicada, sendo possível criptografar os dados baseados na chave privada e descriptografá-los com a chave pública. Além dos benefícios de segurança de dados, este traz a possibilidade de autenticação de envio, pois é possível garantir a identidade do usuário que enviou ou criptografou os dados (IBM, 2021).

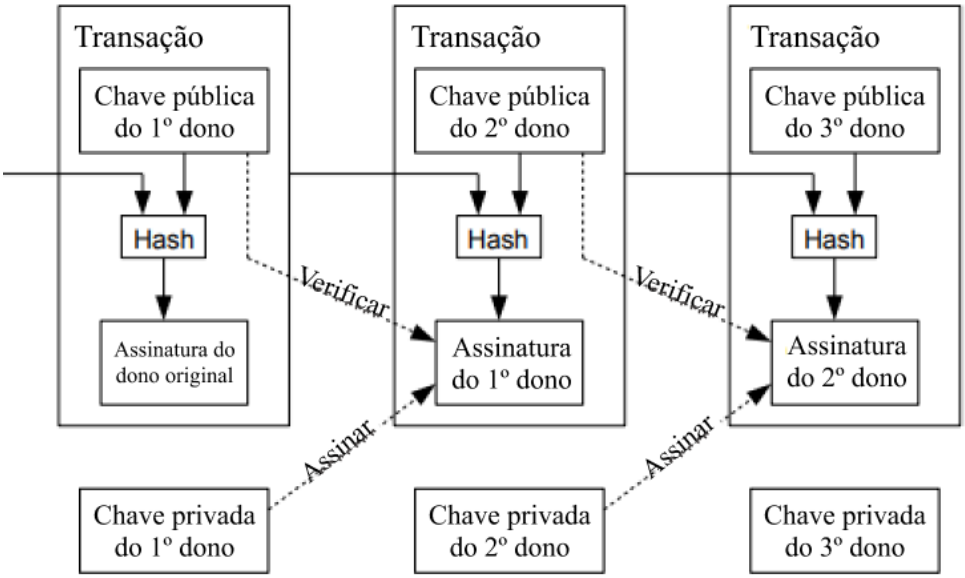
2.4.2 Transações

Nakamoto (2009) define uma moeda digital como uma cadeia de assinaturas digitais, onde o proprietário de cada moeda pode transferir suas moedas para um terceiro, através da assinatura digital de um *hash* da transação anterior com sua chave privada, a qual será verificada

com a chave pública do mesmo e por fim ambas as informações serão adicionadas ao final da moeda.

Este formato funcional pode ser visualizado na Figura 5 e possibilita a verificação das assinaturas em cada moeda, garantindo a veracidade de posse desta. Além disso, o sistema necessita garantir que não existam transações duplicadas, sendo a mais recente a que importa. Para essa garantia ser atingida, todas transações devem ser públicas perante o sistema e todos os participantes do sistema devem concordar em um único histórico de transações, onde a maioria concorda em qual foi a transação mais recente recebida (NAKAMOTO, 2009).

Figura 5 – Diagrama transação de posse de Bitcoins

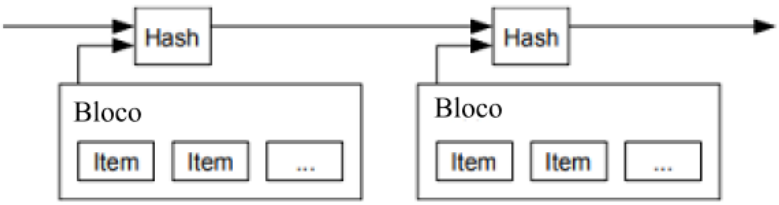


Fonte: Traduzido pelo próprio autor, de Nakamoto, 2009.

2.4.3 Servidor de timestamps

A solução proposta por Nakamoto (2009) se fundamenta fortemente em um servidor de *timestamps*. Este funciona pela inserção contínua de *hashes*, os quais contém o seu *timestamp* de criação, o *timestamp* do *hash* anterior e um bloco de itens. Esta inserção contínua forma a corrente de dados, reforçando todos os *hashes* gerados anteriormente ao mais atual. A Figura 6 exemplifica este funcionamento.

Figura 6 – Diagrama funcional de um servidor de *timestamps*



Fonte: Traduzido pelo próprio autor, de Nakamoto, 2009.

2.4.4 Blocos

Os blocos são os responsáveis por permanentemente armazenar os dados de transação da rede. A sua composição estrutural é feita de cinco conjuntos de dados principais, representados na Tabela 1 e na Tabela 2, sendo alguns destes o tempo atual, uma referência ao bloco imediatamente anterior ao atual e uma lista de transações (BITCOIN WIKI, 2021).

Para um novo bloco ser inserido no *blockchain* este deve ser primeiro "encontrado", este processo sendo conhecido como o processo de mineração. Deve-se salientar que a cadeia de blocos pode possuir ramos, estes ramos ocorrendo na ocasião de dois mineradores encontrarem soluções distintas para o mesmo bloco. Nestas ocasiões a rede é capaz de escolher uma das distintas soluções como a melhor escolha e deletar o ramo contrário. Esta escolha se baseia no ramo com a mais longa cadeia de blocos, onde a maior distância não é calculada pelo maior número de blocos, mas ao maior somatório de dificuldade (BITCOIN WIKI, 2021).

Tabela 1 – Estrutura de um bloco no Bitcoin

Dado	Descrição	Tamanho
Número Mágico	Valor sempre igual a 0xD9B4BEF9	4 bytes
Tamanho do Bloco	Números de bytes que compõem o bloco	4 bytes
Cabeçalho do Bloco	Consiste em 6 dados específicos	80 bytes
Contador de transações	Inteiro positivo	1-9 bytes
Transações	A lista (não vazia) de transações	1-n bytes

Fonte: Bitcoin Wiki, 2021.

Tabela 2 – Estrutura do cabeçalho de um bloco no Bitcoin

Nome do Dado	Propósito	Atualizado quando...	Tamanho (Bytes)
Version	Número de versão do bloco	O programa é atualizado em uma nova versão	4
hashPrevBlock	*Hash* de 256 bits do cabeçalho do bloco anterior	Um novo bloco é inserido	32
hashMerkleRoot	*Hash* de 256 bits baseado em todas as transações do bloco	Uma transação é aceita	32
Time	O *timestamp* do bloco atual em segundos desde 1970-01-01T00:00 UTC	Um pequeno tempo de segundos passa	4

Bits	O valor da dificuldade retornada pela rede em forma compacta	A dificuldade da rede é atualizada	4
Nonce	Um número de 32 bits que inicializa em 0	Um *hash* é tentado	4

Fonte: Bitcoin Wiki, 2021.

2.4.5 Mineração

O processo de mineração é o processo responsável pela adição de novos blocos válidos ao *blockchain*. Este processo é propositalmente difícil e custoso computacionalmente, de modo em que seja possível manter o nível de blocos introduzidos ao *blockchain* estável, os quais para serem inseridos necessitam passar por uma validação de prova de trabalho. Outro papel da mineração é a garantia da dificuldade computacional de alteração do histórico de transações adicionados ao *blockchain*.

Para a mineração funcionar, se faz necessária a presença de mineradores, ou seja, participantes da rede que forneçam poder computacional para executar as tarefas deste processo. Se faz assim presente incentivos à presença de mineradores através de recompensas em Bitcoins, possibilitando a disseminação de moedas de forma descentralizada.

As recompensas recebidas pelos mineradores, em especial a recompensa relativa à descoberta de novos blocos, é reduzida pela metade a cada 210.000 novos blocos inseridos. No atual momento, a criação de novos blocos é o que garante a maior recompensa, cerca de 6,25 *Bitcoins* por bloco (BITCOIN WIKI., 2021).

2.4.6 Prova de Trabalho

O conceito de prova de trabalho, ou *proof-of-work* (PoW), surge a partir do artigo publicado por Adam Back em agosto de 2002, chamado de *"Hashcash – A Denial of Service Counter-Measure"*, o qual tem como propósito propor um mecanismo para intervenção ao abuso sistêmico de recursos da internet. Tal mecanismo é composto por um pequeno pedaço de dados difícil de produzir computacionalmente, porém facilmente verificado. Este pedaço de dados é gerado a partir de um processo randômico com baixa probabilidade de sucesso, no qual a velocidade de solução é influenciada somente pelo poder computacional de geração de novos possíveis dados válidos (dados por segundo) (BACK, 2002). Entretanto, comumente se faz presente na literatura a concepção do acréscimo probabilístico da solução a partir de soluções

falsas encontradas, tal conceito é inválido e é facilmente explicado a partir da falácia de Monte Carlo.

O PoW no contexto do Bitcoin deve produzir, através do algoritmo criptográfico SHA-256C, um hash de 32 bytes dos dados do bloco anterior somados a um valor inteiro, conhecido como nonce, tal que este hash contenha uma quantia pré-fixada de zeros em seu início. A quantia de zeros é definida a partir de um valor, conhecido como dificuldade, o qual é baseado na quantidade de novos blocos criados na rede a cada dez minutos, tal qual que quanto maior a dificuldade, maior a quantidade de zeros em seu início. A não existência de somente um hash possivelmente válido, faz com que mineradores disputem entre si o hash válido mais rapidamente encontrado (BITCOIN WIKI., 2021).

Como os blocos são encadeados e possuem o hash do bloco anterior, todo o trabalho computacional realizado para a criação dos blocos anteriores está contido no bloco mais recente. Caso algum dado de um bloco predecessor tenha seus dados alterados, todos os hashes dos blocos seguintes serão alterados, obrigando o ator malicioso a descobrir enésimos blocos consecutivos de modo que a rede alterada seja mais longa que a correta.

É possível realizar um exemplo simples do seu funcionamento a partir do texto "*Hello, world!*". Hipotetize que a dificuldade da rede esteja configurada a quatro zeros, então deve-se encontrar um *hash* do texto somado a um valor inteiro positivo, *nonce*, que possua seus primeiros quatro bytes iguais a zero. Iniciando o *nonce* em zero e incrementando este até encontrar o *hash* válido levaria 4251 tentativas, ou incrementos ao *nonce*. Este exemplo é demonstrado na Figura 7 (BITCOIN WIKI., 2021).

Figura 7 – Exemplo de *hashes* do texto "*Hello, world!*" incrementado ao *nonce*

```
"Hello, world!0" => 1312af178c253f84028d480a6adc1e25e81caa44c749ec81976192e2ec934c64
"Hello, world!1" => e9afc424b79e4f6ab42d99c81156d3a17228d6e1eef4139be78e948a9332a7d8
"Hello, world!2" => ae37343a357a8297591625e7134cbea22f5928be8ca2a32aa475cf05fd4266b7
...
"Hello, world!4248" => 6e110d98b388e77e9c6f042ac6b497cec46660deef75a55ebc7cfdf65cc0b965
"Hello, world!4249" => c004190b822f1669cac8dc37e761cb73652e7832fb814565702245cf26ebb9e6
"Hello, world!4250" => 0000c3af42fc31103f1fdc0151fa747ff87349a4714df7cc52ea464e12dcd4e9
```

Fonte: Bitcoin Wiki, 2021.

2.4.7 Rede

Nakamoto (2009) define que as regras de funcionamento da rede do *Bitcoin* são:

a) Novas transações são transmitidas a todos nodos da rede;
b) Cada nodo coleta novas transações e as inserem em blocos;
c) Cada nodo trabalha de modo a encontrar um *proof-of-work* para seu bloco;

d) Quando um nodo encontra um *proof-of-work*, transmite a todos os nodos este bloco;
e) Os nodos aceitam o bloco recebido somente se todas as transações dentro deste forem válidas;
f) Os nodos expressam seu aceite ao bloco através do trabalho a criação de um novo bloco seguinte a este na cadeia de blocos, usando o *hash* do bloco aceito como o *hash* anterior.

A cadeia de blocos válida sempre será a mais longa e caso dois ramos surjam de maneira concorrente e válida, estes existirão concorrentemente até que um destes gere o próximo bloco válido em sua cadeia ramificada. Isso expõe inexistente obrigação de compartilhamento a todos os nodos da rede, pois desde que a informação chegue a uma quantia suficiente para validação do bloco já é suficiente para o funcionamento correto da rede. Caso um nodo perca uma atualização da cadeia de blocos, no próximo bloco válido recebido da cadeia este requisitará os blocos perdidos a rede, adicionando-os a sua cópia da cadeia (NAKAMOTO, 2009).

2.5 ETHEREUM

O *Ethereum* foi idealizado em 2014, por Vitalik Buterin, propondo melhorias aos conceitos de scripts, moedas alternativas e protocolos em *blockchains*, como também a criação de uma camada fundamental que possibilitaria a desenvolvedores a construção de aplicações descentralizadas utilizando os paradigmas de *blockchains*. Esta camada fundamental seria abstrata suficientemente para possibilitar a criação de pedaços de código computacionalmente universais, consciente de valores, consciente de *blockchains* e finitos em estado, os quais executariam sobre um *blockchain*, este sendo o Ethereum (BUTERIN, 2014).

De forma simplificada, o *Ethereum* é um *blockchain* que se baseia em uma máquina de estados compartilhada entre todos os participantes da rede e validada através de mineração utilizando PoW. Quando novas transações ocorrem na rede, um novo estado é proposto e caso seja válido, é escolhido pela máquina de estados como o próximo estado válido. O *Ethereum* é funcionalmente similar ao *Bitcoin*, porém os seus blocos contêm uma cópia da lista de transações como também o mais recente estado da rede (BUTERIN, 2014).

Os pedaços de código computacionalmente universais são conhecidos como *Smart Contracts*, ou Contratos Inteligentes. Estes são pedaços de software, que operam somente sobre *blockchains*, e permitem a criação de aplicações descentralizadas, aplicando então os conceitos funcionais de DLTs em programas altamente escaláveis. Com a evolução do *Ethereum* e de

seus contratos inteligentes, criou-se então o conceito de aplicações totalmente descentralizadas rodando sobre *blockchains*, estas são conhecidas como DApps (*Descentralized Applications*) (CHOWDURY et al., 2019).

2.5.1 Transações, blocos e Ether

A máquina de estados do *Ethereum* é composta de inúmeros objetos conhecidos como contas, as quais são divididas em duas categorias, as contas externas e as contas de contratos. As contas externas são controladas por chaves privadas e não possuem nenhum tipo de código embutido, estas podem enviar mensagens a outras contas através de uma transação assinada. As contas de contrato possuem código embutido, o qual possui contexto de execução limitado à chegada de transações de contas externas, podendo armazenar dados, enviar transações e até mesmo criar outras contas de contratos. A transição de estados da máquina de estados do *Ethereum* nada mais é do que a transação de valores ou informações entre estas contas, sendo estes valores representados pelo *Ether*, a criptomoeda do *Ethereum* (BUTERIN, 2014).

Uma conta, independentemente de sua categoria, possui quatro campos:

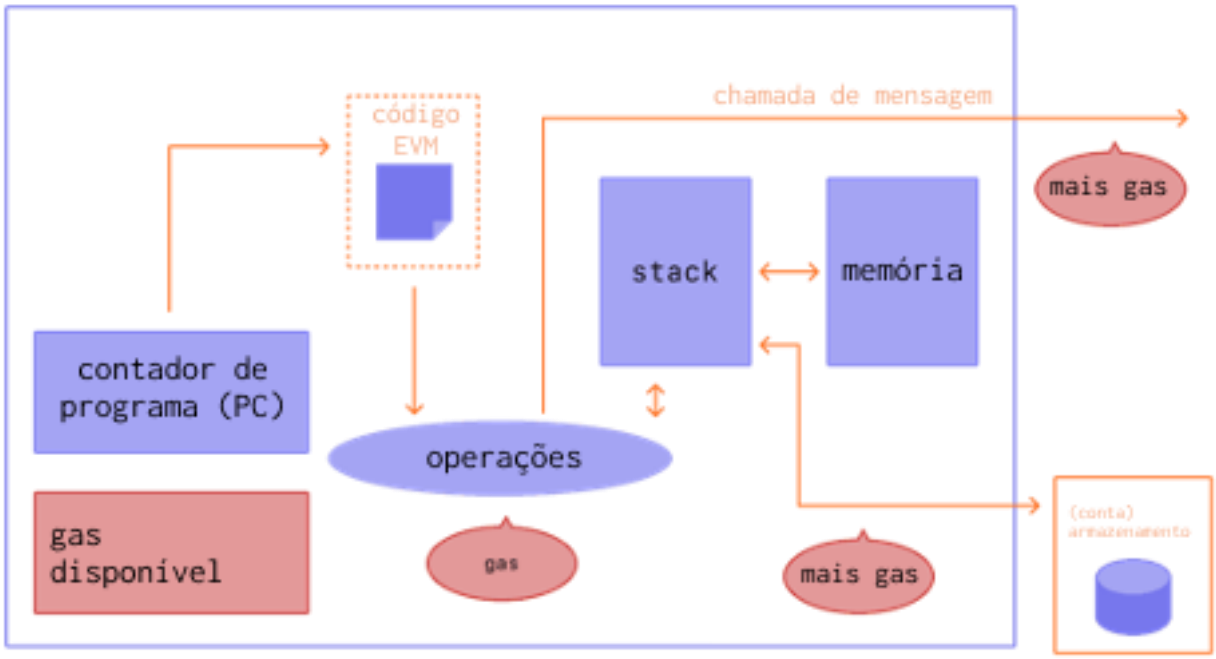
a) Um *nonce* responsável por garantir a unicidade de suas transações;
b) Um balanço da conta em montante de Ethers;
c) O código do contrato, se for uma conta de contratos;
d) E uma seção de armazenamento, iniciado como vazio.

As transações no *Ethereum*, são a assinatura dos valores ou informações a serem transmitidas na rede por contas externas ou como respostas de invocações de contratos inteligentes, estas são compostas de (ETHEREUM ORGANIZATION, 2022):

a) A conta do receptor da transação;
b) A assinatura que identifica o emissor da transação;
c) A quantidade de Ether a ser transferida do emissor para o receptor;
d) Um campo opcional de dados;
e) Um valor chamado de *gasLimit* que representa a quantidade máxima de operações computacionais que uma transação pode realizar;
f) Um valor chamado de *maxPriorityFeePerGas* que representa a quantidade máxima de taxas que serão disponibilizadas para o minerador;
g) Um valor chamado de *maxFeePerGas* que corresponde ao total máximo de taxas a ser pagas pela transação, incluindo os valores de *gasLimit* e *maxPriorityFeePerGas*.

O campo opcional de dados não é comum em outros *blockchains*, já que é utilizado como meio de passagem de parâmetros para contratos inteligentes que operam no *Ethereum*. Enquanto os valores relativos a taxas, nomeado de *gas*, são utilizados para coibir o bloqueio da rede por atores malicioso como também limitar a execução de contratos inteligentes. As taxas, ou *gas*, presentes nas transações da rede são pagas em frações de *Ether* conhecidas como *gwei*. O *gwei* é uma das unidades do *Ether*, a qual equivale a 10^{-9} *Ether*, existindo inúmeras outras unidades como o *wei* e o *pwei*. A Figura 8 demonstra todas as operações computacionais do *Ethereum* que necessitam de taxas para serem executadas (ETHEREUM ORGANIZATION, 2022).

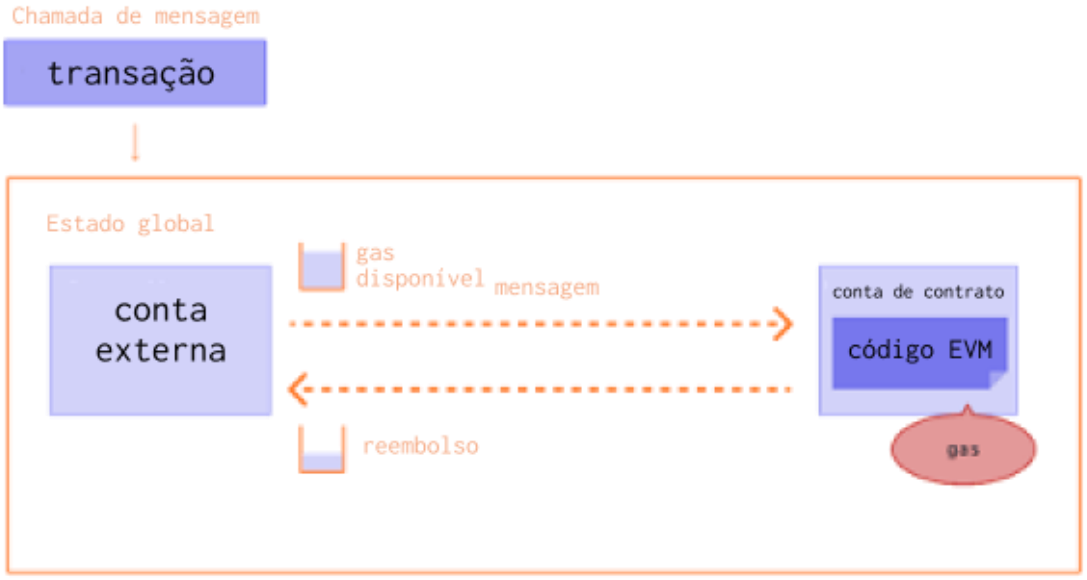
Figura 8 – Diagrama de operações de transação e consumo de *gas* do *Ethereum*



Fonte: Traduzido pelo próprio autor, de Ethereum Organization, 2022.

Neste trabalho não será abordado como o *Ethereum* calcula as taxas para cada operação, porém é importante ressaltar que o parâmetro *maxFeePerGas* garante que exista um limite definido pelo emissor da transação em relação as taxas que este está disposto a pagar. Este valor máximo é inicialmente cobrado do emissor, consumido durante a execução da transação e em caso de sobra de *gas*, esta sobra é reembolsada ao emissor. Entretanto, em caso de falta de *gas* para a transação ser totalmente executada, esta é cancelada e o *gas* não é reembolsado. O primeiro caso pode ser observado na Figura 9, o qual um contrato foi executado e não necessitou de todo o *gas* configurado através do *maxFeePerGas*, o qual reembolsou a sobra para o emissor da transação (ETHEREUM ORGANIZATION, 2022).

Figura 9 – Exemplo de processo de transação com sobra de *gas* no *Ethereum*



Fonte: Traduzido pelo próprio autor, de Ethereum Organization, 2022.

Os blocos do *Ethereum* armazenam transações ocorridas na rede e referenciam o bloco anterior através de um *hash* do bloco anterior. Estes blocos compartilham muitas semelhanças aos blocos do *Bitcoin*, variando em nomenclatura de parâmetros e dados armazenados, é possível notar esta pequena diferença comparando as Tabelas 1 e 2 a Tabela 3 (ETHEREUM ORGANIZATION, 2022).

Tabela 3 – Estrutura de um bloco no Ethereum

Dado	Descrição
timestamp	O registro no tempo ao qual o bloco foi minerado
blockNumber	O comprimento do *blockchain* em blocos
baseFeePerGas	O mínimo valor do *gas* para a transação ser inserida no bloco
difficulty	O valor da dificuldade para a mineração do bloco
mixHash	Um *hash* de identificação do bloco
parentHash	O *hash* de identificação do bloco anterior
transactions	As transações contidas no bloco
stateRoot	Todo o estado do sistema
nonce	Um número de 32 bits que inicializa em 0

Fonte: Ethereum Organization, 2022.

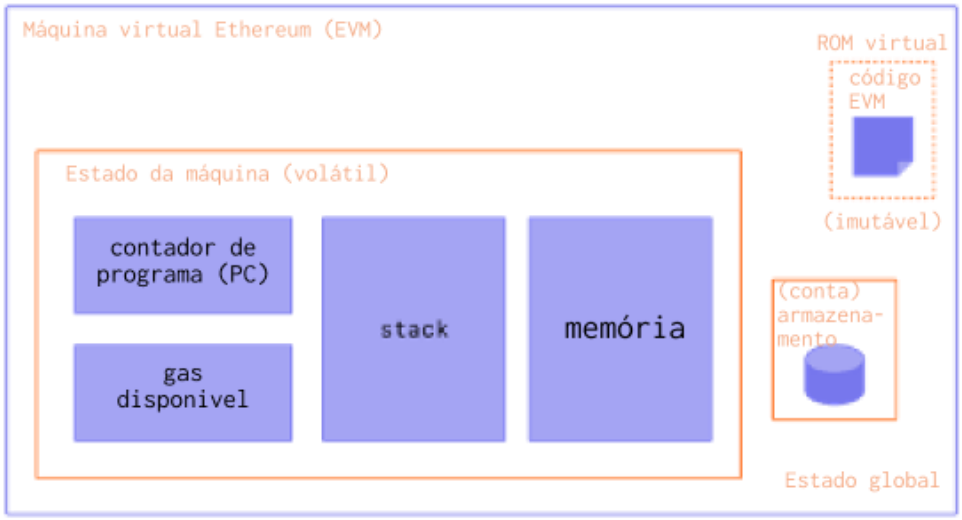
2.5.2 Máquina Virtual do Ethereum

O *Ethereum*, diferentemente do *Bitcoin* e de outros *blockchains*, opera sobre uma única máquina de estados contínua, imutável e interrompível, a qual é fisicamente instanciada através de todos os participantes da rede. Esta máquina de estados, conhecida como a Máquina Virtual do *Ethereum* (EVM), funciona como uma extensão de uma DLT, possibilitando a agregação de operações computacionais as características funcionais de um *blockchain*.

O estado da EVM nada mais é do que uma grande árvore de dados que mantém todas as contas existente conectadas entre si através de *hashes*, as quais todas podem ser reduzidas em uma única raiz desta árvore. A EVM altera entre estados a partir de um método determinístico que pode ser simplificado na equação matemática $f(S,T) = S'$, tal qual que, dado um anterior estado válido S e um novo grupo de transações válidas T, um novo estado válido S' será gerado. A execução desta função de mudança de estados, como também de transações da rede ocorre em um contexto volátil, o qual a EVM instancia um espaço de memória não persistente a cada execução. A Figura 10 demonstra a arquitetura, de maneira simplificada, da EVM como os seus espações de memória e quais dados são persistentes ou voláteis (ETHEREUM ORGANIZATION, 2022).

A EVM não realiza computações de maneira tradicional pois é considerada uma máquina de pilha (*stack*), ou seja, todas as computações são executadas dentro de uma pilha limitada a 1024 elementos contendo dados de até 256 bits (SOLIDITY, 2022).

Figura 10 – Estrutura computacional da EVM



Fonte: Traduzido pelo próprio autor, de Ethereum Organization, 2022.

2.5.3 Anatomia de Contratos Inteligentes

Como dito anteriormente, contratos inteligentes são programas computacionais universais que executam sobre um *blockchain* e são vinculados a uma conta da rede. A programação destes contratos pode ser comparada com outras linguagens como o *C++*, *JavaScript* ou *Python*, porém existem diversas limitações intrínsecas impostas pelo *blockchain*. Algumas destas limitações são a existência de um limite máximo de 24 *kilobytes* em seu tamanho, a falta de acesso nativo às informações estrangeiras ao *blockchain* e a natureza *a priori* de código aberto. Entretanto, existem técnicas para contornar algumas destas limitações, como a técnica dos diamantes para possibilitar contratos maiores que 24 *kilobytes* e os *Oracles* para obtenção de dados externos ao *blockchain*. Atualmente a linguagem de programação de contratos inteligentes mais proeminente é o *Solidity* (ETHEREUM ORGANIZATION, 2022).

Uma das principais diferenças de contratos para *softwares* é o seu modelo de execução, o qual deriva do comportamento dos contratos como contas na rede, sendo executado por toda a rede e não por uma máquina específica. Isto é possível pois as contas armazenam o código do contrato em formato de *bytecode*, gerado pela linguagem de programação utilizada, e quando uma transação é realizada para a conta que possui o contrato, o código é invocado e executado utilizando o *bytecode* (ETHEREUM ORGANIZATION, 2022).

Outra diferença importante é a necessidade de declaração de contexto de armazenamento para as variáveis do contrato. Existem dois tipos de armazenamento, o *storage* e o *memory*, o primeiro é um espaço de memória fixo, persistente após execução e operacionalmente custoso, enquanto o segundo é um espaço de memória variável, que existe somente em tempo de operação e é barato operacionalmente. Como a execução destes contratos é vinculada a quantidade de *gas* disponibilizada, existe a necessidade de um grande foco na otimização do uso da memória como também na limitação de uso de funções de loop (SOLIDITY, 2022).

Uma outra consideração é a existência de uma operação chamada de *selfdestruct* que remove os dados e o código do contrato, enviando a quantia de *Ether* disponível na conta para um terceiro escolhido. Esta operação é altamente perigosa, pois após a operação ser executada todo e qualquer *Ether* enviado para o contrato é perdido para sempre. Mesmo que o contrato e seus dados sejam removidos, todo o histórico do contrato ainda estará armazenado na rede. Contratos inteligentes também podem gerar novos contratos através da instanciação de construtores, estes quando invocados realizam a transação de criação de um novo contrato baseando-se nos parâmetros passados pelo contrato criador ao construtor (SOLIDITY, 2022).

Por fim, contratos inteligentes utilizam um modelo estandardizado de comunicação conhecido como Interface Binária de Aplicação (ABI) que possibilita a comunicação entre contratos e programas estrangeiros ao *blockchain*. O ABI é encapsulado em um objeto de notação *JavaScript* (JSON), servindo como um protocolo de chamada de procedimento remoto encapsulado em um JSON (JSON RPC), que possui um vetor das funções, eventos e descrições de erros presentes no contrato (SOLIDITY, 2022).

2.5.4 Redes Privadas

O *Ethereum*, *Bitcoin* e a grande maioria dos *blockchains* existentes no mercado são categorizados como *blockchains* públicos, ou seja, qualquer indivíduo pode participar, contribuir e operar na rede. Entretanto, existem casos em que se é benéfico a utilização de uma estrutura mais privada, limitando acessos a dados, operações e funcionalidades, deste modo surge o conceito de *blockchains* privados (ETHEREUM ORGANIZATION, 2022).

Blockchains privados possibilitam a flexibilização da configuração da rede por parte dos fornecedores e desenvolvedores, em contrapartida, os usuários perdem os benefícios da descentralização. Redes privadas possuem, nativamente, todas as funcionalidades do *blockchain* utilizado para instanciá-la, entretanto os dados presentes nesta rede não são válidos na rede pública paralela (ETHEREUM ORGANIZATION, 2022).

Rodar uma rede privada do *Ethereum* é bem simples, se instancia o bloco de gênese da rede contendo as configurações de protocolo da rede e se cria o estado inicial da EVM desta rede. O processo para criação de nodos e mineradores pode seguir o mesmo processo da rede pública, como também é possível configurar partes destes. O *Ethereum* possui uma configuração que possibilita a utilização de prova de autoridade, ou *proof-of-authority* (PoA) para redes privadas, a qual possibilita a eliminação de taxas em transações e execuções de contratos. O PoA é um algoritmo de consenso distribuído centralizado que utiliza nodos específicos para a validação e criação de blocos na rede, não tendo nenhuma grande desvantagem em relação ao PoW (ETHEREUM ORGANIZATION, 2022).

2.5.5 Proof-Of-Stake

O *Ethereum* tem a expectativa de alterar o seu mecanismo de consenso distribuído até o final do ano de 2022, substituindo o atual *proof-of-work* pelo *proof-of-stake*, ou prova de *stake* (PoS). Esta mudança é fortemente influenciada pela necessidade de maior escalonamento da

rede do *Ethereum* e o alto consumo energético necessário para a execução do PoW. Do mesmo modo que estas necessidades são importantes e requerem certa urgência, o expressivo desafio técnico de se alterar o consenso distribuído de um *blockchain* requer tempo para testes em ambientes adequados e programações em caso de erros (ETHEREUM ORGANIZATION, 2022).

O PoS não necessita de grandes poderes computacionais para a chegada de um consenso, pois ao contrário do PoW, os mineradores não competem entre si visando a resolução de um *hash* e estes são substituídos por validadores. Validadores são participantes da rede que utilizaram 32 *Ethers* para obterem este papel, o valor em *Ether* é bloqueado de ser utilizado pelo validador desde que este deseje manter o papel. Validadores são escolhidos de maneira aleatória para a criação e validação de novos blocos, caso um novo bloco válido seja criado ou validado, uma recompensa é paga ao validador, entretanto, se este tentar criar blocos inválidos ou falhar nas validações, partes ou o todo de seu *Ether* podem ser retiradas do seu montante bloqueado (ETHEREUM ORGANIZATION, 2022).

Esta atualização de algoritmo de consenso distribuído possibilitará uma participação mais descentralizada na validação e criação de novos blocos, como também, trará o conceito de *shards* à rede do *Ethereum*. *Shards* são *blockchains* separados que operarão sobre um *blockchain* central chamado de *beacon chain* possibilitando o escalonamento exponencial das transações por segundo da rede (ETHEREUM ORGANIZATION, 2022).

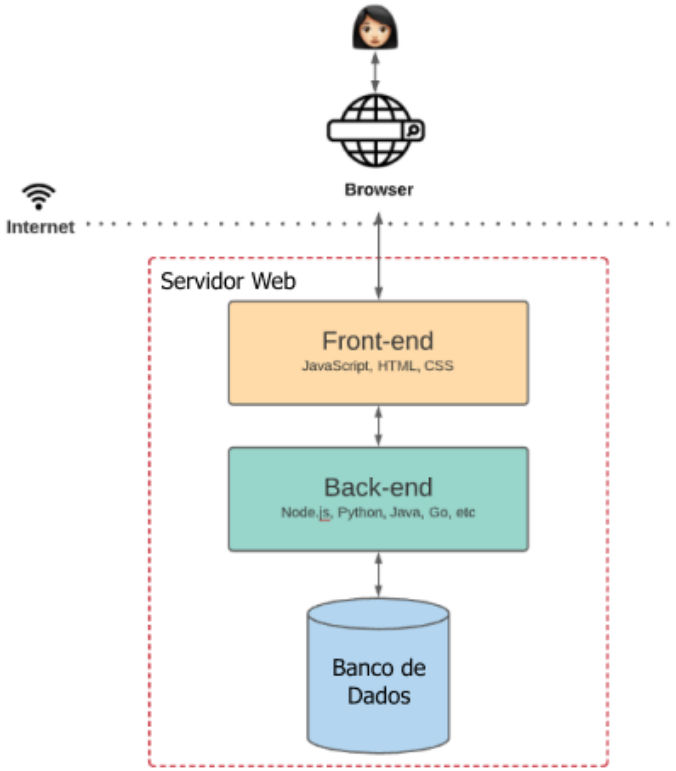
2.6 WEB 3.0

O conceito de Web 3.0 surge a partir da consolidação de tecnologias como DLTs e contratos inteligentes, as quais impulsionam as DApps. A Web 3.0 segue os princípios dos *blockchains*, ou seja, é decentralizada, de igual acesso, com pagamentos nativos e opera utilizando algoritmos de consenso distribuído. Este conceito ainda é extremamente imaturo e evolui ao longo que as tecnologias que o compõem evoluem, necessitando inúmeras definições em relação a arquitetura e escopo de alcance (KASIREDDY, 2021a).

A estrutura atual da web, conhecida como Web 2.0, já é extremamente complexa de um ponto de vista operacional, este sendo altamente impulsionada pela introdução de sistemas de computação em nuvem e a necessidade de estruturas mais resilientes a falhas e a invasões. Mesmo assim, a criação de serviços na Web 2.0 é atualmente considerada como trivial, pois diversas soluções surgiram possibilitando a generalização da implementação através de modelos arquiteturais estruturados e iterações de melhorias funcionais. Uma arquitetura

simplificada da Web 2.0 é demonstrada na Figura 11, sendo composta pela aplicação de *front-end, back-end* e algum banco de dados para armazenar os dados da aplicação.

Figura 11 – Modelo arquitetural simplificado da Web 2.0



Fonte: Traduzido pelo próprio autor, de Kasireddy, 2021a.

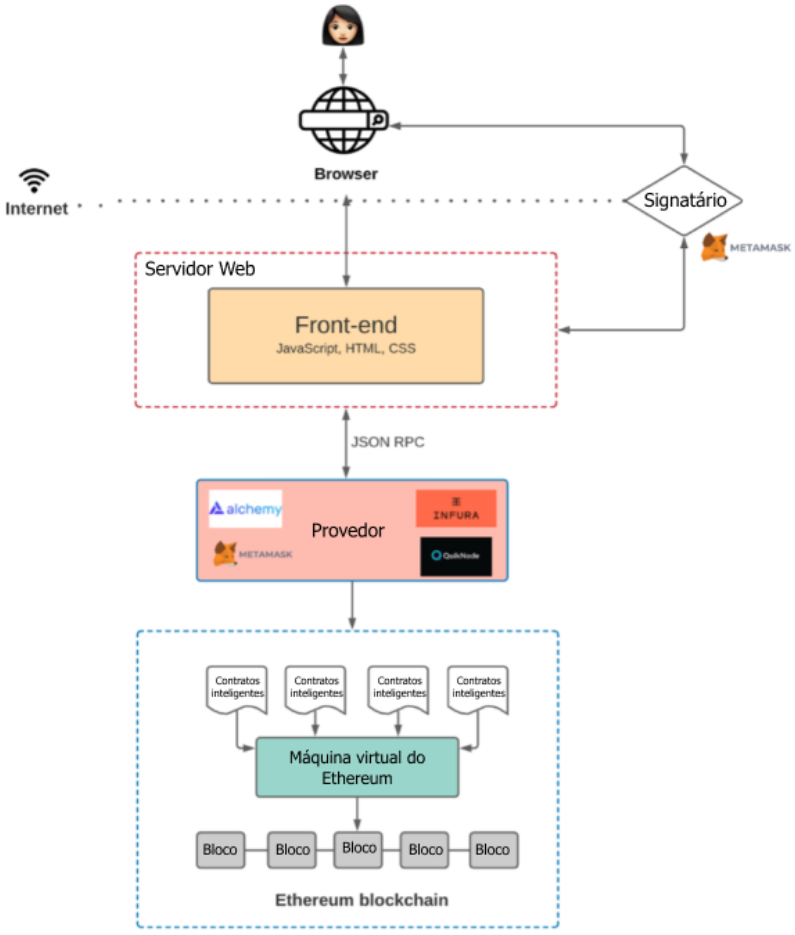
2.6.1 Arquitetura da Web 3.0

A complexidade arquitetural da Web 3.0 cresce exponencialmente à introdução de tecnologias como *blockchains* e contratos inteligentes ao contexto de execução. Os pontos de acesso do usuário como a camada visual das aplicações da web permanecerão minimamente alteradas, sendo as maiores modificações focadas no *backend*, banco de dados e o modelo de interação entre *front-end* e *back-end* (KASIREDDY, 2021a).

Nesta nova estrutura, se faz necessária a criação de um método para o *front-end* se comunicar com os contratos inteligentes e os dados do *blockchain*, podendo ocorrer através de serviços terceiros ou de um nodo configurado pelos próprios desenvolvedores. Os serviços terceiros, como o *Metamask* ou *Infura*, instanciam nodos e possibilitam a conexão através da internet a estes nodos, enquanto a instanciação própria depende da configuração e manutenção ativa de nodos pelo time de desenvolvimento. Este mecanismo de comunicação entre *front-end* e o *blockchain* é conhecido como provedores (KASIREDDY, 2021a).

Os provedores utilizam o protocolo JSON RPC para delimitar as operações acessíveis pelas aplicações de *front-end*, podendo ser aplicado em diversos protocolos de comunicação como o *Hypertext Transfer Protocol* (HTTP) ou *sockets*. Como todas as transações do *blockchain* devem ser assinadas por uma conta, necessita-se então de um serviço que dê acesso aos usuários as suas contas do *blockchain*, estes serviços são conhecidos como as carteiras digitais (KASIREDDY, 2021a). A comunicação entre o *front-end* e os contratos inteligentes disponíveis no *blockchain* possibilitam que todo o *back-end* de nosso sistema seja instanciado na própria rede. A Figura 12 demonstra a agregação destes serviços no modelo arquitetural simplificado da Web 2.0.

Figura 12 – Modelo arquitetural parcial da Web 3.0



Fonte: Traduzido pelo próprio autor, de Kasireddy, 2021a.

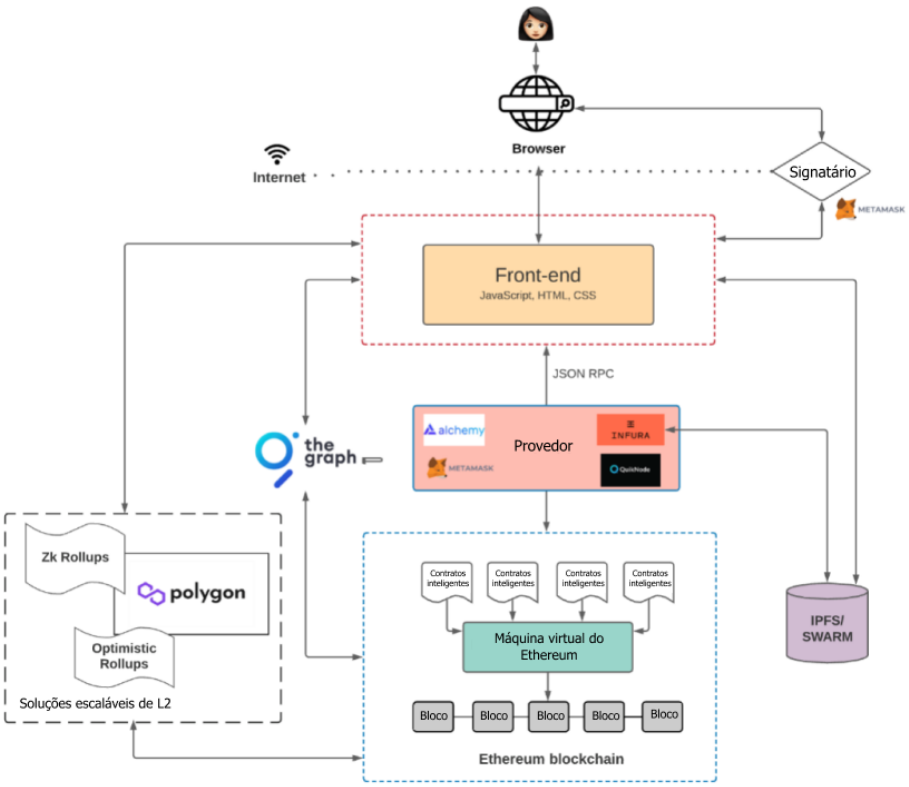
Até aqui foi definido como o código do *front-end* se comunicará e realizará operações em seu novo *back-end*, o *blockchain*, porém armazenamento e consulta de dados ainda não foram endereçados. Para isto, surgem serviços de armazenamento P2P que se comunicam com *blockchains*, como o Sistema de Arquivos Interplanetário (IPFS) e o *Swarm*, eliminando o alto custo de armazenamento de dados diretamente na rede. Outra solução possível, mas centralizada, é a reutilização da prática comum em Web 2.0 de bancos de dados distribuídos. O

IPFS e o *Swarm*, também possibilitam a hospedagem de códigos de *front-end*, criando mais um desligamento a serviços centralizados como a *Amazon Web Services* ou a *Azure* (KASIREDDY, 2021a).

Nativamente contratos inteligentes possuem um sistema de eventos, que possibilita que aplicações, através de provedores, escutem a respostas de execução de funções e a dados de contratos. Esta ferramenta nativa já é suficiente para a maioria das atuais aplicações, porém, ao longo que novas aplicações mais complexas surgem este modelo se prova ineficiente. Com isto, se inclui uma ferramenta chamada de *The Graph*, entre o *front-end* e o *blockchain*, a qual possibilita a consulta e retorno de dados relativos a contratos de maneira eficiente, modelada e segura (KASIREDDY, 2021a).

Por fim, um dos principais atuais problemas do *Ethereum* é a falta de escalabilidade, isto impacta fortemente o desempenho e a viabilidade de aplicações descentralizadas. Para solucionar estes problemas de escalabilidade, é proposta a integração com soluções de camada dois. Estas soluções de camada dois possibilitam a redução de taxas e do tempo de execução, através da execução de diversas operações em *blockchains* paralelos ao *Ethereum*. A Figura 13 compila todas as soluções comentadas em um único modelo simplificado arquitetural da Web 3.0 (KASIREDDY, 2021a).

Figura 13 – Modelo arquitetural da Web 3.0



Fonte: Traduzido pelo próprio autor, de Kasireddy, 2021a.

3 METODOLOGIA

O presente trabalho visou estender o modelo arquitetural da Web 3.0, apresentado por Kasireddy (2021a), para o contexto da Indústria 4.0, através da implementação de aprimoramentos funcionais. Cabe então à metodologia deste trabalho enumerar todas as etapas necessárias para a implementação e estruturação destas melhorias e do modelo arquitetural estendido proposto.

Primeiramente, a partir do embasamento teórico, foi possível determinar quais as principais áreas dentro do contexto da Indústria 4.0 que DLTs, *blockchains* e contratos inteligentes poderiam ser eficientemente utilizados. Os conteúdos expostos por Gunes *et al.* (2014), Lee e Seshia (2015) e Jamai, Azzouz e Saidane (2020) explicitaram a necessidade de um ecossistema industrial focado na segurança e privacidade, dois pilares da DLT. Criou-se assim, a primeira conexão direta à aplicação destas tecnologias ao contexto industrial, sendo escolhido o *blockchain* do *Ethereum* como a fundação tecnológico para esta aplicação.

A partir da fundação tecnológica escolhida, foi possível então, aplicar conceitos presentes nos trabalhos de Kasireddy (2021a), Asif *et al.* (2022), Amrutiya *et al.* (2019) e Alladi *et al.* (2019) para alastrar as conexões com a Indústria 4.0 além das encontradas incialmente. Doravante à definição destes conceitos, foram definidos aprimoramentos funcionais com foco no *supply-chain* e controle de qualidade industrial, paralelamente às melhorias inicialmente propostas de segurança e privacidade.

O desenvolvimento de todas as melhorias propostas ocorreu através da criação de pedaços de código, modelos arquiteturais e estruturas funcionais configuráveis de redes de *blockchain.* Todo o desenvolvimento foi realizado utilizando o maior nível de generalização possível, visando criar uma base estrutural e tecnológica que seja capaz de evoluir ao longo que novas alterações e soluções surjam. Esta necessidade adveio da atual realidade em relação às tecnologias e ferramentas utilizadas neste trabalho, as quais estão em processo de maturação e evoluem em grande velocidade.

Ao início do desenvolvimento ressurgiu a necessidade de novos embasamentos teóricos, fomentados pelo desconhecimento em relação aos atuais processos e sistemas que englobam as melhorias propostas. Deste modo, buscou-se contato com duas empresas do ramo industrial de *hardware* do Rio Grande do Sul, que desejaram manter-se anônimas, para o fornecimento de conhecimentos suficientes sobre seus sistemas de controle de qualidade, manufatura, fatura, segurança e privacidade. A partir destes conhecimentos, foi possível enquadrar as

responsabilidades de cada uma das tecnologias utilizada de maneira definitiva, esclarecendo sua relação operacional aos atuais sistemas presentes.

Ao longo do desenvolvimento foram registradas peculiaridades e possíveis débitos técnicos presentes nas tecnologias utilizadas, como também foi procurado compilar a maior quantidade de estudos, matérias e artigos que analisem quantitativamente estas tecnologias. Com estes dados foi possível realizar uma posterior análise de viabilidade de implementação destas melhorias ao contexto industrial, levando em consideração pontos como escalabilidade, segurança tecnológica, necessidade de retrabalho e custos operacionais.

Ao final do desenvolvimento e de sua análise de viabilidade, propôs-se um modelo arquitetural genérico da Web 3.0, que possibilite a aplicação incremental das tecnologias da Web 3.0 aos atuais sistemas e arquiteturas presentes na Indústria 4.0. Este modelo, serve como fundamentação teórica geral para aplicações futuras das tecnologias abordadas e expõe pontos operacionais da indústria que podem interagir de maneira proveitosa e eficiente, com o ambiente da Web 3.0.

Por fim, analisou-se qualitativamente os principais pontos em que, nos dias de hoje, se faz válida a integração, validação e implementação das tecnologias abordadas e quais seus complicadores à adoção. Esta análise levou em consideração o estado da arte das tecnologias aplicadas, tecnologias concorrentes presentes no mercado e as possíveis alterações futuras do ecossistema da Web 3.0 como um todo.

4 DESENVOLVIMENTO

Os dados apresentados nesta seção pretenderam explicitar o processo de desenvolvimento, o modelo de melhoria proposto e a análise dos resultados obtidos. Para a sua melhor compreensão, os dados presentes foram comprimidos em subseções relativas a cada uma das melhorias propostas.

4.1 AUTENTICAÇÃO, AUTORIZAÇÃO E PRIVACIDADE

Um dos principais vetores para ataques em sistemas computacionais industriais, são os seus sistemas de autenticação e autorização de usuários. O recente crescimento no montante de ataques aos sistemas industriais, têm tornado cada vez mais comum a terceirização destes sistemas de autenticação e autorização industrial. Estes terceiros, geralmente, são grandes empresas de cyber-segurança que fornecem estruturas tecnológicas, serviços de *Single Sign On* (SSO) industrial e autenticações multifator (MFA). Entretanto, o acesso a estes serviços ainda é restrito a indústrias de grande porte, dado o seu alto custo de implementação, forçando a opção de soluções tradicionais, menos seguras e por muitas vezes desenvolvidas sem a estrutura necessária, pelas indústrias de pequeno e médio porte.

Propõe-se então um modelo de implementação de sistemas de autenticação e autorização de usuário, através da utilização de conceitos presentes em *blockchains* e a instanciação do código através de contratos inteligentes, que seja capaz de reduzir os vetores de ataque. É importante ressaltar que para o contexto deste trabalho, serão abstraídas certas partes da implementação, como a integração software e hardware, pois o foco é criar uma base para construções futuras e não um sistema completo.

Para o desenvolvimento deste modelo, primeiramente foi necessária a categorização deste sistema em relação à localidade de autorização e autenticação, sendo categorizado em *in loco* e remota. Para validar ambos os modelos, foi desenvolvida uma aplicação que possui a seguinte estrutura: uma **Single Page Application** (SPA) implementada utilizando **JavaScript** e o framework **React** que possibilita a ação de autenticação por parte do usuário, uma API desenvolvida em **Node.js** para a intermediação da comunicação e contratos inteligentes implementados em **Solidity** para a execução da ação de autenticação.

O modelo remoto teve forte embasamento nos artigos de Kasireddy (2021a), Amrutiya *et al.* (2019) e Asif *et al.* (2022) e foca na autorização e autenticação via internet de DApps que tenham acessos a dados e serviços industriais. O modelo é separado em quatro partes funcionais,

o cliente, a SPA, o serviço de autenticação e os contratos inteligentes, sendo suas atividades utilizadas para a descrição do funcionamento do modelo como um todo.

O serviço de autenticação inicia pela sinalização da ação de autenticação pelo cliente (usuário) através da SPA, invocando funções e chamadas de interface de programação de aplicações (API) que iniciam o processo junto aos contratos inteligentes. A autenticação segue um padrão simplificado do protocolo *OAuth* 2.0 e utiliza tokens de acesso JWT de 60 minutos para a autorização persistente. Sendo assim, caso um token esteja presente no cliente, este será enviado juntamente com os dados do usuário para validação por um contrato inteligente. Inicialmente o token e os dados disponibilizados são validados durante o processo de autenticação, o qual em caso de sucesso, indica que o cliente foi autenticado com sucesso dentro dos últimos 60 minutos, não necessitando de novos requerimentos de autenticação ou um novo token. Entretanto, qualquer requisição por parte do cliente que envolva manipulação de dados ou interação com o *blockchain*, será necessária uma autenticação de MFA adicional.

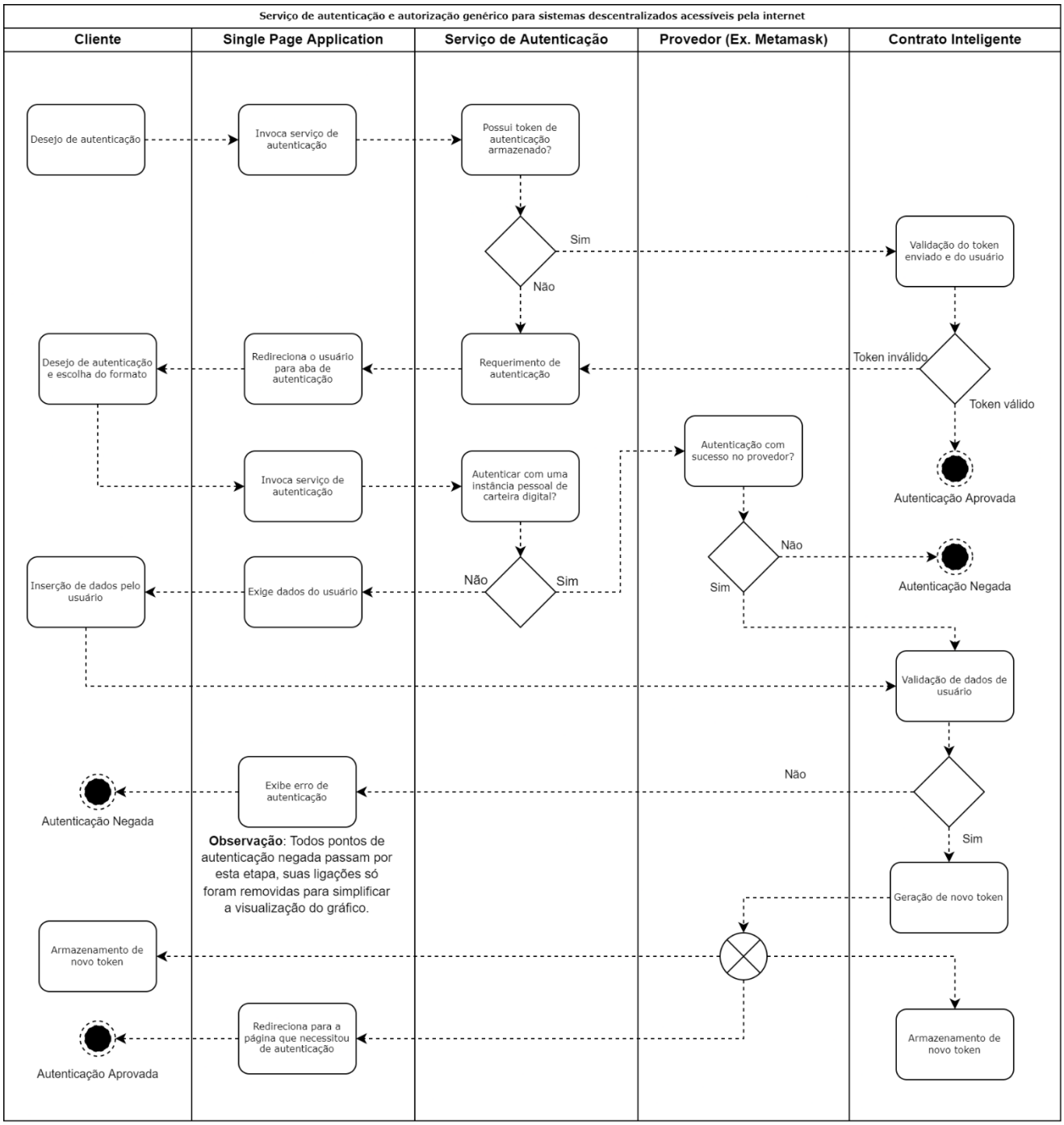
Caso nenhum token esteja presente no cliente, o cliente poderá escolher se autenticar através do seu provedor ou através de dados pessoais como uma chave pública, senha e nome de usuário. Ambos os processos terão seus dados validados através de um contrato inteligente, que em caso de sucesso retorna um novo token JWT com duração máxima de 60 minutos. Este JWT é armazenado no banco de dados descentralizado possibilitando a consulta futura para a sua validação. Todo o funcionamento descrito até aqui é descrito visualmente pela Figura 14.

Como salientado anteriormente, este token JWT não é suficiente para a execução de operações no sistema, possibilitando somente a leitura de dados pelo usuário. Para a autorização de execução de operações no sistema foi utilizado um protocolo MFA adicional de senhas de uso único (OTP). Este protocolo foi instanciado em um serviço secundário vinculado a um identificador de *hardware* único, que é responsável pela geração da senha e a comunicação desta senha ao serviço de autenticação. Quando o cliente deseja realizar uma operação que necessite de uma OTP, a aplicação gera uma requisição para a geração de uma nova senha, gerando então um *hash* baseado na junção da chave privada do usuário, identificador único de *hardware* e um valor pseudorrandômico, utilizado como a OTP. Esta senha tem validade máxima de 60 segundos e logo que é gerada é transmitida ao contrato de autenticação e ao serviço de OTP, possibilitando a visualização ao cliente e a validação da autenticação por parte do contrato. Este comportamento é descrito pela Figura 15.

Um ponto extremamente importante que ainda não foi abordado até o momento, é o processo de inserção de novos usuários e suas permissões em relação aos diferentes sistemas e dados. Este processo foi proposto através da criação de uma rede privada minimalista do

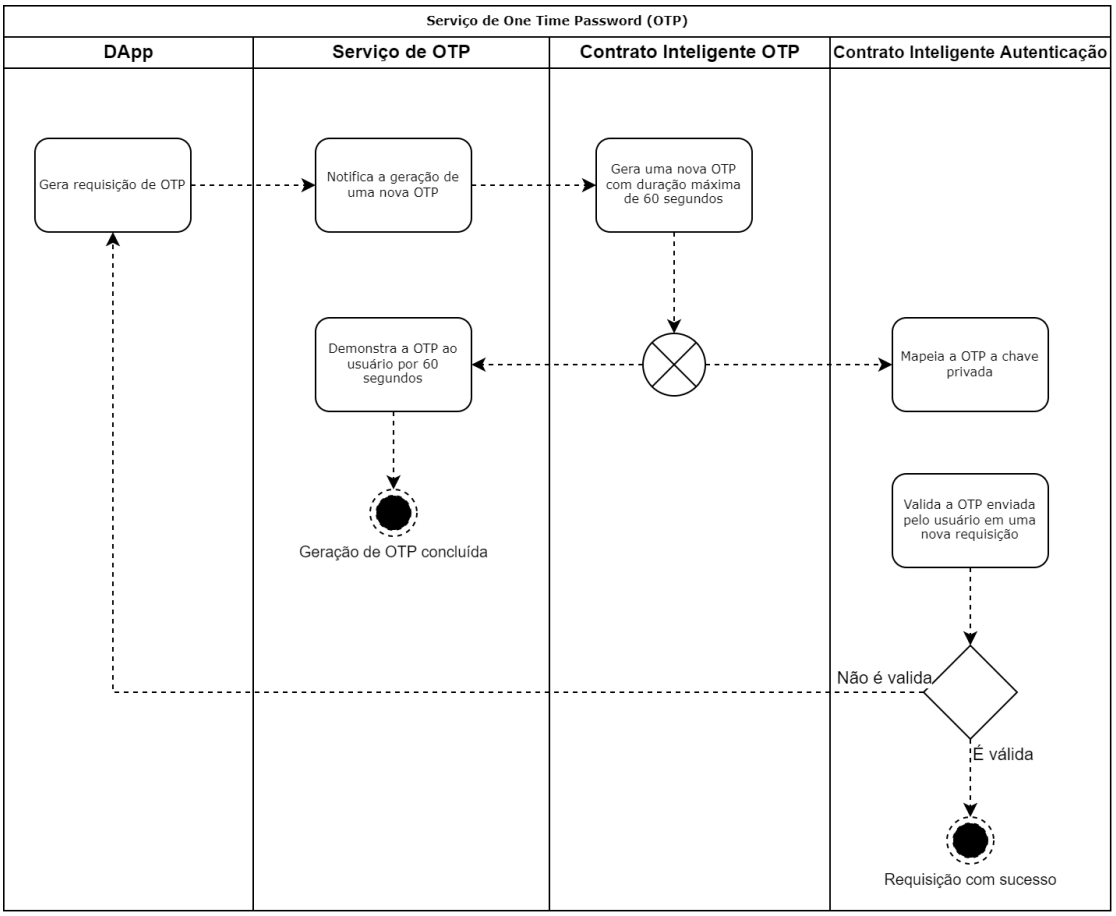
Ethereum, a qual tem contexto exclusivo de operação a este processo. Os nodos desta rede são usuários administradores, em forma de *hardware* instanciado *in loco*, de uma empresa participante da rede. Cada nodo valida automaticamente operações de modificação ou adição advindas de outras empresas, porém, para operações advindas do contexto de sua própria empresa, requerem uma adicional validação manual e distribuída, realizada *in loco* através de administradores humanos credenciados.

Figura 14 – Modelo de autenticação e autorização remoto



Fonte: Imagem compilada pelo próprio autor.

Figura 15 – Modelo do serviço de OTP



Fonte: Imagem compilada pelo próprio autor.

O modelo de autenticação e autorização *in loco* é muito parecido funcionalmente com o modelo remoto, entretanto sua complexidade estrutural é consideravelmente maior. Sendo esta realidade proporcionada principalmente pela necessidade de integração de dados presentes em pedaços de *hardware*, como identificadores por radiofrequência (RFID) ou escaneadores biométricos, ao sistema. Diferentemente do modelo remoto, toda a estrutura deste sistema pode ser instanciada sem conexões às redes públicas, possibilitando a negação dos vetores de ataque advindos de malfeitores externos.

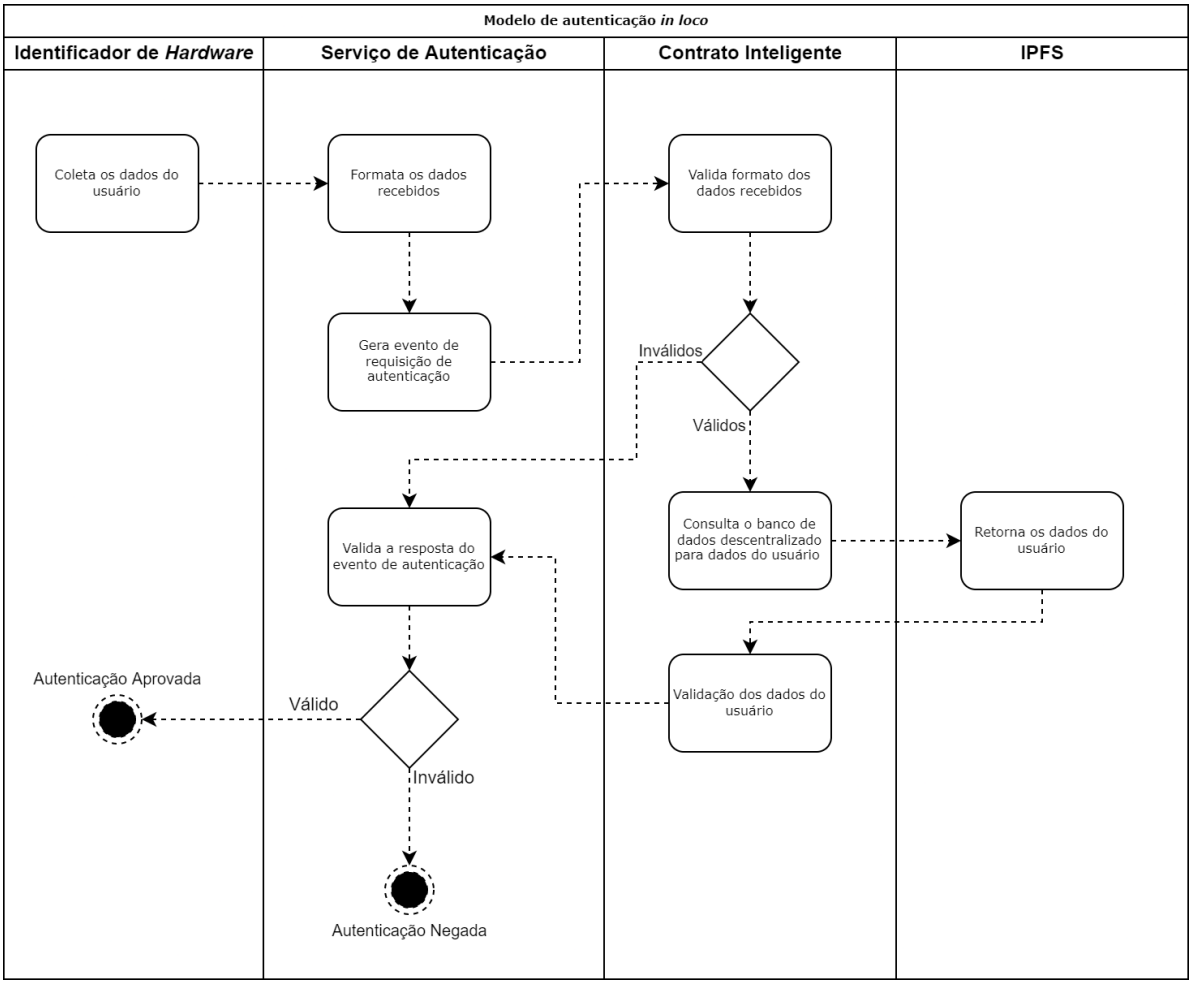
Para a descrição do funcionamento deste modelo foi substituído o cliente pelo identificador de *hardware* e o provedor como também o SPA puderam ser removidos, visto que não há interface visual e as chaves privadas e públicas serão armazenadas pelo próprio identificador de *hardware*, através de uma memória restrita a leitura (ROM). Neste modelo o serviço se torna responsável pela formatação dos dados obtidos pelo identificador de *hardware* em momento de requisição, e a subsequente geração de eventos únicos de requisição de autenticação em forma de fila. O contrato inteligente recebe o evento da fila, valida a presença de todos os dados necessários e a sua formatação, em sequência, obtém dados do usuário a

partir do banco de dados descentralizado e por fim valida a requisição. Em qualquer caso, o evento é retornado ao serviço, que valida os dados do resultado do evento, autenticando ou não o usuário. Este comportamento do modelo é descrito visualmente na Figura 16.

É importante que o credenciamento de novos usuários, como a remoção de permissões ou usuários, não dependa de uma rede paralela neste modelo em específico, para o melhor aproveitamento da estrutura da rede interna. Sendo assim, a própria rede utilizada para manter o serviço e seus contratos inteligentes pode ser utilizada como meio de instanciação do serviço responsável por prover alterações aos dados de usuário. Além disto, para gerar uma ainda maior eficiência computacional, os *hardwares* que instanciam os nodos da rede podem servir também como intermediadores para a comunicação de dados industriais internos e como mantenedores da estrutura do banco de dados descentralizado interno de maneira distribuída.

É indispensável ao modelo em questão utilizar algum algoritmo de consenso distribuído, entretanto em relação ao modelo remoto, este consenso é limitado aos nodos presentes na rede local e não necessariamente utiliza do mesmo algoritmo para o consenso distribuído.

Figura 16 – Modelo de autenticação e autorização *in loco*



Fonte: Imagem compilada pelo próprio autor.

A autenticação e autorização de usuários é essencial à operação segura dos processos industriais, porém, caso não exista a privacidade dos dados compartilhados, nenhum sistema de autenticação e autorização será totalmente seguro. A natureza destes dados no contexto da indústria é altamente variada como também a sua importância ao processo industrial, porém, estes geralmente originam de sensores, controladores e máquinas industriais.

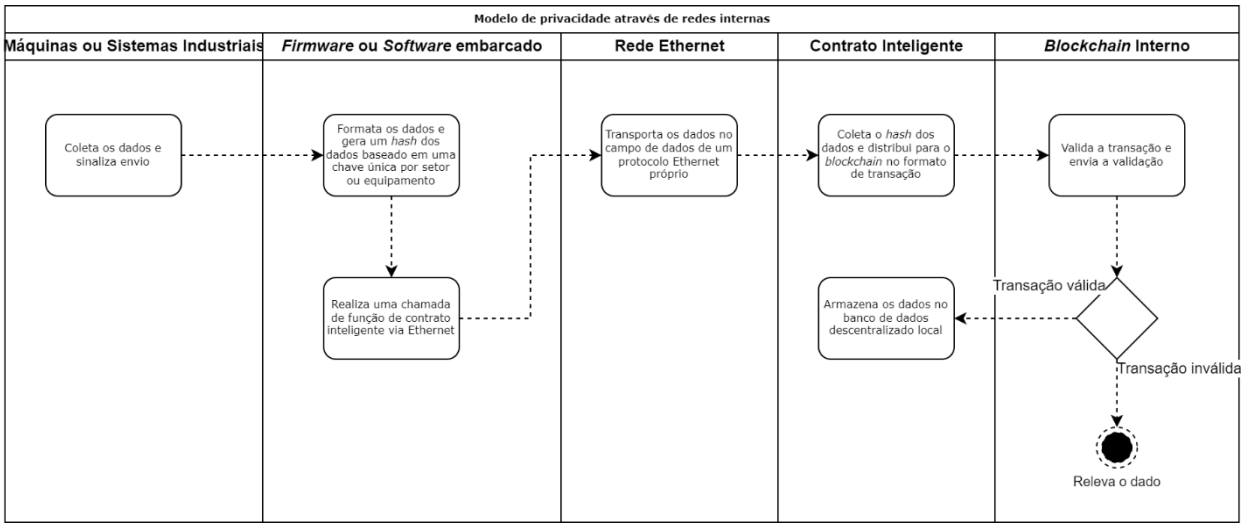
Na grande maioria dos casos, a instanciação de uma estrutura de rede interna, sem acessos às redes externas, já é suficiente para a geração de um razoável nível de privacidade, visto que os dados são transportados em um contexto extremamente reduzido. Não obstante a isto, para malfeitores coletarem estes dados, seria necessária a invasão da rede local, a qual obrigatoriamente só pode ser acessada dentro dos perímetros empresariais. Mesmo que este modelo seja suficiente para muitas situações, dada a existência de uma rede privada do *Ethereum* previamente configurada, foi possível aproveitá-la para a melhoria da privacidade.

Para o aprimoramento da privacidade, inicialmente foi criado um protocolo de encapsulamento de rede, exclusivo para uso interno, com codificação hexadecimal igual a *0x99AA*. Esse código identifica ao receptor como os dados transportados devem ser processados, o qual no contexto desta melhoria, seria o redirecionamento do pacto à um contrato inteligente acessível na rede interna. Como comumente estes dados são transportados direto entre máquinas com funcionalidades de IOT aos seus receptores, foi proposta a adição de uma etapa de criptografia dos dados baseado em um *hash* do identificador único da máquina e um par de chaves privada-pública configurada exclusivamente para cada máquina.

Ao receber os dados, o contrato inteligente valida o formato da requisição como também se o *hash* advém de uma máquina cadastrada, e em caso de sucesso, o dado é transmitido como uma transação para a rede. Após a realização da transação, o contrato salva o *hash* dos dados no banco de dados descentralizado disponível à rede, sendo este comportamento descrito na Figura 17. Este banco de dados pode ser consultado em tempo real, por usuários ou aplicações credenciadas disponíveis na rede interna, através de chamadas de API ou de interfaces de aplicações descentralizadas.

Este modelo não descreveu formas de consulta através de redes externas, pois tem como princípio a remoção máxima das funcionalidades às redes externas. Entretanto caso seja de interesse o acesso a alguns dos dados industriais de maneira remota, é possível replicar partes do banco de dados descentralizado de modo que contratos inteligentes funcionem como APIs para o retorno destes *hashes* e sua descriptografia para usuários credenciados. Esta abordagem não foi tomada no contexto deste trabalho, mas é válido ressaltá-la, visto o crescimento da distribuição do controle industrial.

Figura 17 – Modelo de privacidade



Fonte: Imagem compilada pelo próprio autor.

4.2 FATURA E CONTROLE DE QUALIDADE

Com o crescimento do acesso à internacionalização dos serviços em indústrias de pequeno e médio porte, a utilização de moedas fiduciárias estrangeiras, como o dólar e o euro, se torna cada vez mais comum em negociações entre empresas. Esta internacionalização é benéfica por diversos motivos, mas principalmente em relação à conexão de empresas a insumos anteriormente inacessíveis e à possibilidade de internacionalização do preço dos produtos produzidos. Entretanto, a atual estrutura de pagamentos internacionais, em geral, é altamente custosa, lenta e burocrática. Juntamente a isto, a necessidade de adequação de faturas a leis locais e exteriores complica o acesso a este tipo de mercado. Surge assim, uma grande possibilidade de implementação de sistemas P2P automatizados para pagamentos utilizando moedas fiduciárias digitais com foco na transação *Business-to-Business* (B2B) internacional.

Para melhor entender como os atuais sistemas de fatura operam dentro da indústria, foi realizada uma pesquisa com duas empresas, as quais desejaram permanecer-se anônimas, do ramo industrial de *hardware* do Rio Grande do Sul que possuam familiaridade com comércio internacional. A realidade coletada sobre esses sistemas, expôs a grande necessidade de trabalho manual próprio ou terceirizado em relação a verificação de faturas e a presença de uma considerável taxação e demora nas transações internacionais.

Para a solução destes problemas expostos, foi proposta a implementação de uma plataforma virtual P2P que abstrai legislações e taxas através de contratos inteligentes configurados na rede pública do *Ethereum*. Esta plataforma opera tanto como possível

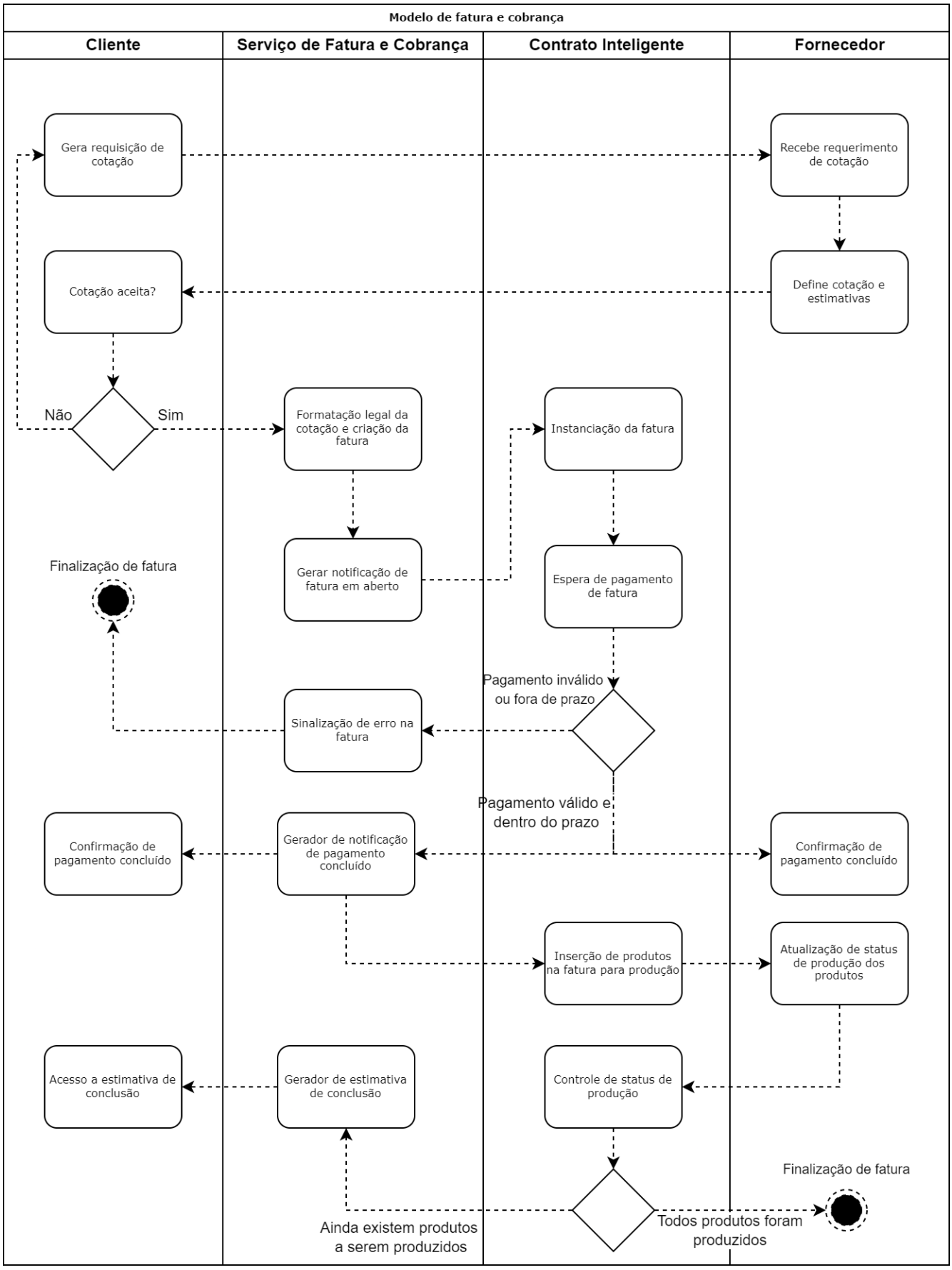
intermediador de transação, através da abstração de carteiras e contas, ou como conector entre contas e carteiras já existentes na rede. As transações são realizadas através de contratos inteligentes, utilizando soluções de camada dois para a aceleração das transações e redução de *gas* por operação, e utiliza valores fiduciários como câmbio para as faturas. A utilização de contratos inteligentes possibilitou uma ampla abordagem modular em relação às diferentes taxas e legislações presentes em faturas. Essa modularidade foi possível de ser atingida pois contratos inteligentes possuem nativamente herança e polimorfismo de *software*.

Não obstante ao alto nível de modularidade atingido, alguns dos processos manuais não puderam ser removidos, dada a natureza de transações industriais, a qual necessita de diversas cotações sequenciais até o fechamento de um pedido. A plataforma se torna um mediador de cotações até o momento em que ambas as partes aceitem uma cotação final, a partir daí, a plataforma formaliza a cotação em uma fatura contendo todas as obrigações financeiras legais. Esta fatura tem tempo de duração configurável por parte do fornecedor e é instanciada em um contrato inteligente que opera como o *gateway* de pagamento da fatura, retendo valores de obrigações financeiras legais na transação para pagamentos futuros.

Caso a fatura tenha sido paga dentro do prazo e de maneira válida, uma ordem de manufatura é gerada e inserida nos sistemas de produção do fornecedor, sendo a mesma comunicada também ao cliente como uma confirmação de início de produção. Ao longo que a produção dos produtos desta fatura ocorre, um contrato inteligente controla, através de uma máquina de estados finita, os estados de cada um destes. Paralelamente, o cliente pode observar estimativas de conclusão de produção de sua fatura, baseadas nos estados atuais dos produtos. Este sistema finaliza sua execução no momento que não existam produtos dentro da fatura a serem produzidos. Este comportamento pode ser visualizado na Figura 18.

O mesmo processo de internacionalização responsável pelas modificações aos modelos de negócios empresariais, contribui também a uma abordagem mais distribuída e complexa ao *supply-chain* industrial. Uma pesquisa realizada pela organização *Peerless Research Group* (2016) demonstrou que 84% de empresas terceirizam ao menos uma de suas etapas de produção e que esta prática, anteriormente exclusiva às empresas de grande porte, espalhou-se entre todos os tamanhos empresariais. A pesquisa ainda expôs que esta prática possibilitou a redução dos custos de manufatura e o acréscimo de qualidade do produto, dado o acesso às empresas com mais experiência e conhecimento em processos especializados. Fora isto, 33% das empresas pretendiam expandir a terceirização e 46% pretendiam mantê-la nos próximos anos, indicando um processo constante de expansão desta prática.

Figura 18 – Modelo de cobrança e fatura



Fonte: Imagem compilada pelo próprio autor.

Em contrapartida, a pesquisa explicitou a existência de pontos de falha que afetam diretamente a eficiência e efetividade desta prática, dentro destas falhas as principais são a falta de visibilidade, responsabilidade de estoque e transparência entre as empresas. Ademais, as pesquisas realizadas por Ford (2015) e por Wildan e Pribadi (2020) documentaram a necessidade do esforço ao combate da soberania do controle de qualidade nos processos de *supply-chain* mais distribuídos.

A partir destas informações, propôs-se a expansão da plataforma utilizada para o modelo de fatura, de modo que ela contenha estados e etapas adicionais que possibilitem o controle de qualidade de maneira descentralizada entre as empresas colaboradoras. Cada nova etapa de controle de qualidade realizada para um produto, é armazenada em formato de transações no *blockchain*, possibilitando a criação de uma cadeia de estados do produto acessível e atualizável por todas as partes.

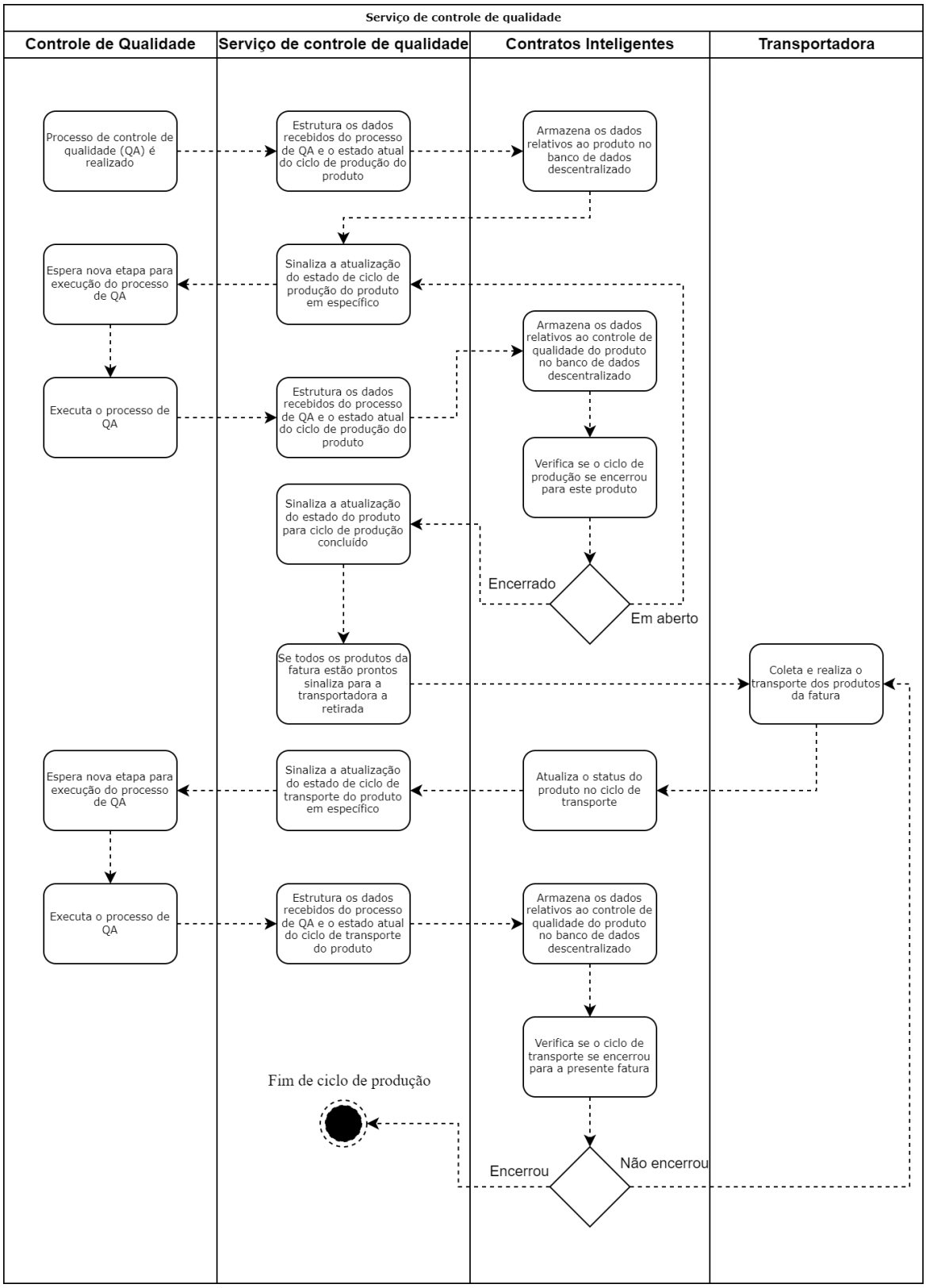
Este registro encadeado possibilita a verificação e controle do estado de qualidade do produto a cada nova etapa de produção, visualizando gargalos operacionais e os principais processos responsáveis por defeitos. Além disto, este sistema gera maior grau de transparência entre as diferentes empresas que compõem o *supply-chain* e facilita a redução da soberania sobre o controle de qualidade, visto que existe a necessidade de cooperação mútua no controle de qualidade do produto.

Os atuais processos de controle de qualidade automatizados possibilitam a implementação deste sistema de forma reativa, excluindo a necessidade de adição de novas etapas ao processo de produção para o seu uso. Entretanto, processos de controle de qualidade manuais ou parcialmente automatizados necessitarão de uma etapa adicional de sinalização, sendo responsável pela coleta do resultado do processo de controle de qualidade e a identificação do produto.

Esta expansão sistêmica inicia sua execução a partir do recebimento de uma sinalização por parte do processo de controle de qualidade, contendo o identificador do produto e dados sobre a qualidade do produto. Estes dados são validados e em caso de sucesso são armazenados no banco de dados descentralizado, sinalizando que a atual etapa de controle de qualidade ocorreu com sucesso. A partir disso, o sistema espera subsequentes novas chamadas do processo de controle de qualidade para o produto até o seu encerramento de produção. Ao fim de sua produção e dos produtos complementares da fatura, o sistema sinaliza o início do processo de transporte destes. O processo de transporte também faz parte do controle de qualidade, visto a possibilidade de extravios, danificações ou até mesmo perda de produtos durante o seu transporte. Ao fim do transporte, os produtos não necessariamente passaram por

todo processo de produção, no caso de necessidade de novos processos de produção, os estados são reiniciados, mas agora baseado em um novo processo de produção. Este processo pode ser visualmente representado pela Figura 19.

Figura 19 – Modelo de controle de qualidade



Fonte: Imagem compilada pelo próprio autor.

4.3 MODELO ARQUITETURAL ESTENDIDO

O modelo arquitetural da Web 3.0 proposto por Kasireddy (2021a) é altamente suficiente para aplicações com contexto operacional limitado à internet, entretanto, sistemas que operam sobre dados do meio físico exigem componentes estruturais que cubram as necessidades destas operações. Outro ponto importante é que essas aplicações possuem um alto nível de segurança, dada a realidade do *blockchain*, porém possuem um baixo nível de confidencialidade, visto a publicidade e acessibilidade dos dados aos usuários da rede.

O meio industrial é altamente competitivo por natureza, o qual expõe a necessidade da confidencialidade de dados, entretanto os recentes avanços da Indústria 4.0 e suas tecnologias, trouxeram novos pontos de falha à segurança e privacidade destes dados. Paralelamente, o avanço das DLTs e do poder computacional acessível ao mercado, possibilitou o surgimento de redes seguras por natureza, altamente configuráveis e com alto nível de privacidade.

Entretanto, a dificuldade de compreensão tecnológica de DTLs e a necessidade de mão de obra extremamente capacitada para sua implementação, são os principais responsáveis pela lenta adoção do mercado. Sendo assim, a partir das melhorias desenvolvidas anteriormente, foi sugerido um modelo arquitetural simplificado que seja capaz de explicitar os contextos de aplicação das tecnologias que compreendem a DLT escolhida neste trabalho servindo como fundamentação para futuras implementações.

O modelo arquitetural em questão utilizou diferentes instâncias do *blockchain* do *Ethereum* para a criação de contratos inteligentes e armazenamento de dados. A rede principal do *Ethereum* teve papel secundário, diferentemente do proposto por Kasireddy (2021a), visto que este modelo tentou isolar ao máximo os dados e operações industriais do meio público. Deste modo, foi necessária a implementação e configuração de uma instância privada do *Ethereum* que servirá como a rede principal para o modelo.

Para a criação desta rede privada utilizou-se a ferramenta de licença gratuita *Hyperledger Besu*, a qual é uma implementação do protocolo do *Ethereum* utilizando a linguagem de programação *Java*, porém a mesma rede pode ser criada com outras implementações como o *Go Eth*. A primeira etapa necessária, além da instalação da ferramenta, foi a criação de uma conta utilizada para a assinatura do bloco de gênese da rede, realizada através de linha de comando. Com os dados da conta armazenados de maneira segura em um objeto físico, foi criado então um arquivo JSON nomeado de *genesis.json* que conterá os dados que definem o comportamento da rede e que gerará o bloco de gênese da rede. Os dados inicialmente configurados no bloco de gênese são demonstrados na Tabela 4.

Tabela 4 – Estrutura do arquivo *genesis.json*

Dado	Descrição	Valor
config	Objeto de configuração básica da rede	Dados representados na Tabela 5
nonce	Utilizado na computação de novos blocos	0x0
coinbase	Endereço de pagamento de recompensas de mineração	0x0000000000000000000000000000000000000000
timestamp	Data e tempo de criação do bloco	0x0
difficulty	Identificador da dificuldade de mineração da rede	0x1
mixHash	Identificador combinado ao *nonce* que identifica o consenso	0x63746963616c2062797a616e74696e65206661756c7420746f6c6572616e6365
gasLimit	Limite de *gas* utilizado para as transações presentes em um bloco	0x29b92700
number	Número de transações no bloco	0x0
alloc	Definição de alocação inicial de *Ethers*	Vetor de validadores iniciais da rede e seus respectivos valores
gasUsed	Quantidade de *gas* utilizado para a transação	0x0
parentHash	O *hash* do bloco anterior	0x0000000000000000000000000000000000000000000000000000000000000000

Fonte: *Hyperledger Besu*, 2022.

Tabela 5 – Estrutura do objeto *config* do arquivo *genesis.json*

Dado	Descrição	Valor
chainid	Identificador numérico da rede	10.000
homesteadBlock	Indica qual versão do *Ethereum* utilizar	0
eip150Block	Configuração de utilização de melhoria EIP150	0
eip155Block	Configuração de utilização de melhoria EIP155	0
eip158Block	Configuração de utilização de melhoria EIP158	0
byzantiumBlock	Configuração de utilização de melhoria Bizantina	0
qbft	Objeto que identifica o protocolo de consenso e sua configuração	Dados representados na Tabela 6
transitions	Objeto que identifica blocos futuros de configuração da rede	Vetor de blocos futuros e período máximo de computação de bloco

Fonte: *Hyperledger Besu*, 2022.

Tabela 6 – Estrutura do objeto *qbft* do arquivo *genesis.json*

Dado	Descrição	Valor
epochlength	Quantidade de blocos para a reinicialização dos votos	50.000
blockperiodseconds	O tempo mínimo em segundos para criação de blocos	2
requesttimeoutseconds	O tempo de *timeout* em segundos para casos de falhas	4

Fonte: *Hyperledger Besu*, 2022.

O protocolo de consenso distribuído escolhido para a instanciação da rede foi o *proof-of-authority* QBFT, o qual define a necessidade de aceite de 66% dos validadores da rede para a inserção de um novo bloco. Neste protocolo, a adição ou remoção de validadores ocorre a partir da proposição e votação pelos validadores presentes, sendo altamente eficiente em um contexto sem acesso a redes públicas. Este protocolo possibilita também que caso metade dos validadores da rede percam conexão ou sejam comprometidos, nenhum novo bloco seja criado na rede.

Foram declarados quatro validadores iniciais, presentes no objeto de configuração *alloc*, os quais servirão para a adição de eventuais validadores à rede e para a validação de transações e blocos. Juntamente, foram instanciados os contratos inteligentes responsáveis pela autenticação e autorização de usuários, impossibilitando desde a gênese o acesso indevido a dados ou funcionalidades.

Para o controle das transações foi definido um valor ao *gasLimit*, o qual leva em consideração o poder computacional disponível para a manutenção de um ritmo constante de transações na rede. É comum pensar que redes privadas não devem ser restritas ao componente de *gas*, entretanto, caso um valor razoável não seja configurado, ataques ou instabilidades sistêmicas podem ser geradas a partir de execuções em *loop* infinito de contratos inteligentes, além de que, o ritmo de transações se torna imprevisível.

Antes da transação de gênese ser executa, é necessária a configuração da rede industrial de modo em que seja possível a descoberta dos validadores e a execução de transações para a rede privada. Algumas configurações necessárias são a instanciação de *proxies* reversos, direcionamento de portas e a abertura de portas P2P para o protocolo UDP. Após estas configurações, a transação de gênese pode ser executada, gerando o bloco de gênese da rede.

Paralelamente à configuração da rede, foi necessária a configuração de um *cluster* IPFS que opere dentro da rede interna de maneira distribuída entre todos os validadores da rede. Este *cluster* é responsável pela formatação do banco de dados descentralizado utilizado pela rede e pelo armazenamento das aplicações descentralizadas que operarão sobre a rede privada. Além disto, foram configurados *proxies* e aberturas de portas necessárias para a o funcionamento do *cluster*, como também configurações de acesso e formatação de dados.

Com todas estas configurações realizadas, a primeira transação da rede pode ser transmitida, validada pelos validadores e o primeiro bloco inserido no *blockchain*. A partir deste momento, todos os contratos restantes podem ser instanciados na rede e serviços de autenticação e configuração de novos usuários podem ser iniciados, juntamente com suas aplicações descentralizadas.

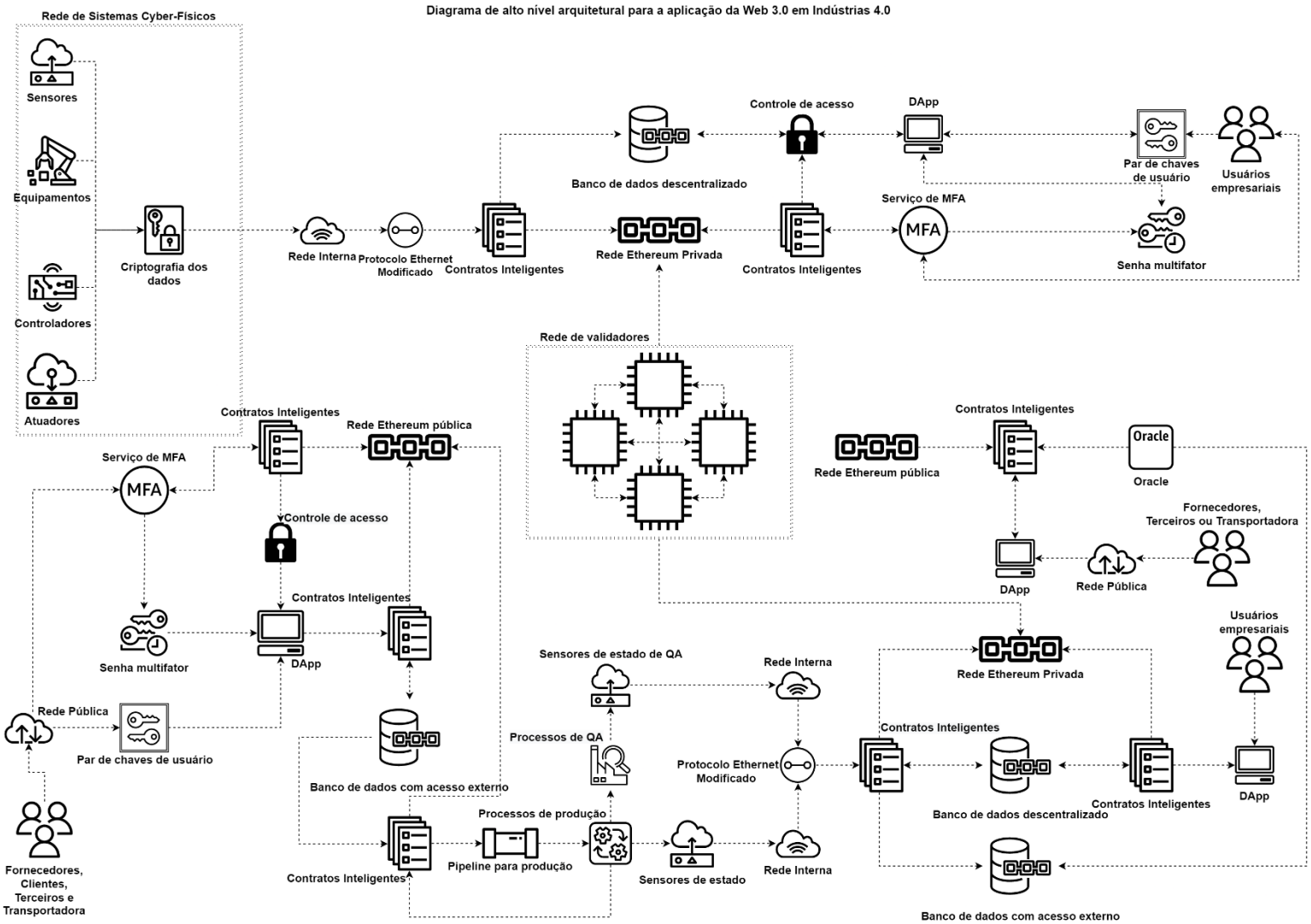
Para a integração de sistemas presentes na rede pública do *Ethereum*, como o sistema de fatura e cobrança, foi criado um banco de dados IPFS, o qual não faz parte do *cluster* interno, que serve como intermediador de dados entre a rede pública e a rede interna do sistema. Este banco de dados, se faz presente na rede interna, porém possui conexões às redes externas, e seus dados são exclusivos ao armazenamento de ordens de fatura e o seu status. A partir destes dados, o sistema de manufatura coleta as ordens de manufatura e insere no *pipeline* de produção.

O sistema de manufatura juntamente ao controle de qualidade, operam dentro da rede interna e se comunicam com a rede pública do *Ethereum*, isto é possível através da adição de um *Oracle* de *input* periódico aos contratos inteligentes da rede pública, que realiza chamadas ao banco de dados descentralizado acessível à rede externa. A partir desta funcionalidade, a cooperação no *supply-chain* e o acesso as estimativas de produção ao consumidor podem ser geradas e demonstradas visualmente em uma DApp.

Os validadores da rede não serão utilizados para a conexão às redes externas, impossibilitando ataques via internet, entretanto, como comentado anteriormente, alguns serviços deverão ser responsáveis pela comunicação com os contratos inteligentes e as DApps disponíveis na rede pública. Estes serviços podem ser intermediadores instanciados na própria estrutura de rede empresarial com conexão às redes externas, ou podem ser serviços instanciados na rede pública do *Ethereum* que se comunicam através de *Oracles* com a rede empresarial.

Toda esta estrutura comentada até aqui pode ser visualizada na Figura 20, de maneira abstraída e simplificada, para o melhor entendimento da distribuição tecnológica e estruturação de comunicações.

Figura 20 – Modelo arquitetural da Web 3.0 expandido para a Indústria 4.0



Fonte: Imagem compilada pelo próprio autor

4.4 ANÁLISE DO DESENVOLVIMENTO E COMPORTAMENTO SISTÊMICO

Através de dados qualitativos coletados durante o desenvolvimento das melhorias propostas e da consulta bibliográfica a análises quantitativas ao desempenho da rede do *Ethereum*, possibilitou-se a análise sistêmica *a priori* a sua aplicação ao contexto industrial.

O processo de desenvolvimento de contratos inteligentes não pode ser comparado aos processos de desenvolvimento de *back-end* comumente praticados. Mesmo que contratos inteligentes operem funcionalmente como o *back-end* das DApps, necessitou-se de uma readequação e aprendizado constante a um novo modelo procedural. Como exposto por Kasireddy (2021b), a imutabilidade dos dados presentes no *blockchain* impossibilitam a correção de *bugs* em contratos inteligentes em tempo de execução, facilitando a exposição a malfeitores e podendo gerar falhas catastróficas aos sistemas que os utilizam. Ademais, ao longo que novas funcionalidades, operações e aprimoramentos são adicionados à rede e a linguagem *Solidity*, somente em sequentes instanciações de novos contratos inteligentes. Sendo assim, o conhecimento das limitações operacionais e do comportamento sistêmico do *blockchain* do *Ethereum* e da linguagem *Solidity*, são essenciais previamente ao processo de desenvolvimento.

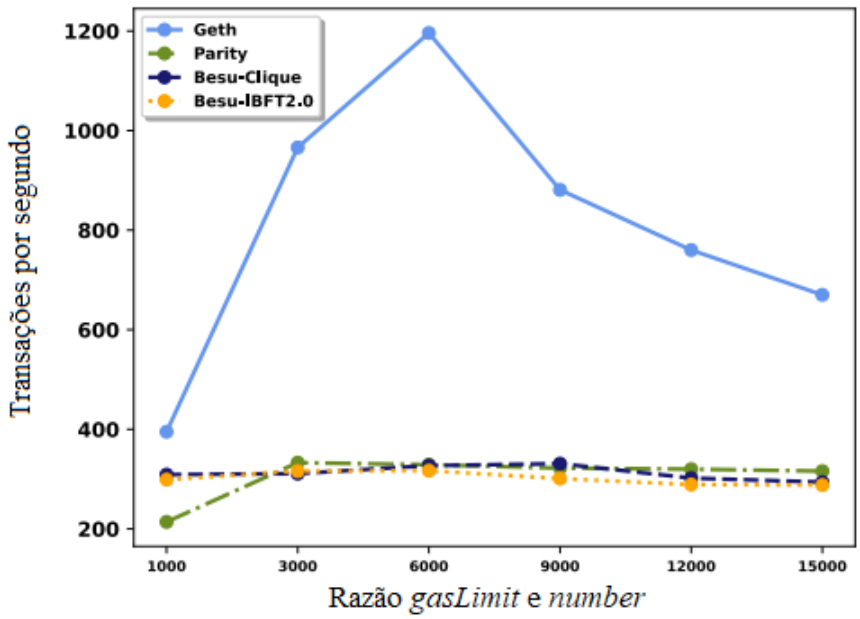
A grande maioria das melhorias propostas foi instanciada em uma instância privada do *blockchain* do *Ethereum*, de modo que foi possível relevar a necessidade de eficiência de *gas* presente no desenvolvimento de contratos. Contratos inteligentes presentes na rede pública necessitam da maior eficiência operacional em relação ao uso de *gas* viável, pois correm o risco de passarem dos limites da rede de uso de *gas* e falharem em sua execução. No caso da rede privada utilizada, a limitação da eficiência foi o poder computacional disponível, que deve manter uma taxa de execução constante e não limitante. Sendo assim, foi evidenciado a obrigatoriedade de um conjunto de testes exclusivo à performance contratual.

O processo de desenvolvimento de DApps pode ser comparado aos processos de desenvolvimento de código de *front-end* comumente praticados. Isto foi possível dada a abstração presente em atuais ferramentas que facilitam a conexão à rede do *Ethereum* e seus contratos inteligentes e a proximidade dos serviços de computação em nuvem, como a *AWS*, ao serviço IPFS. Entretanto, dificuldades foram recorrentemente encontradas na formatação das requisições de transação, escuta de eventos e na formatação de dados. A limitação proposital de tipos de dados, presente na linguagem de programação *Solidity*, forçou o desenvolvimento de funções intermediárias que traduzam o tipo de dado presente, neste caso, na linguagem

JavaScript. Este processo, não é algo novo, porém a pouca documentação presente para a linguagem *Solidty* dificultou consideravelmente o desenvolvimento.

A pesquisa realizada por Samuel *et al* (2021), evidenciou que a escolha do cliente que implementa a rede privada do *Ethereum* é de extrema importância à eficiência e desempenho da rede. Além do cliente, a razão entre a configuração de limite de *gas* e do número de transações por bloco, impacta diretamente a quantidade de transações por segundo que a rede é capaz de realizar. Samuel *et al* (2021), demonstrou diferenças marginais entre os clientes *Parity, Besu Clique* e *Besu* IBFT 2.0 na quantidade de transações por segundo, independente da variação da razão presente na rede. Entretanto, o cliente *Geth* apresentou melhorias significativas em uma razão com valor de 6.000 e durante toda a variação desta razão possui valores significativamente superiores, podendo serem visualizados na Figura 21.

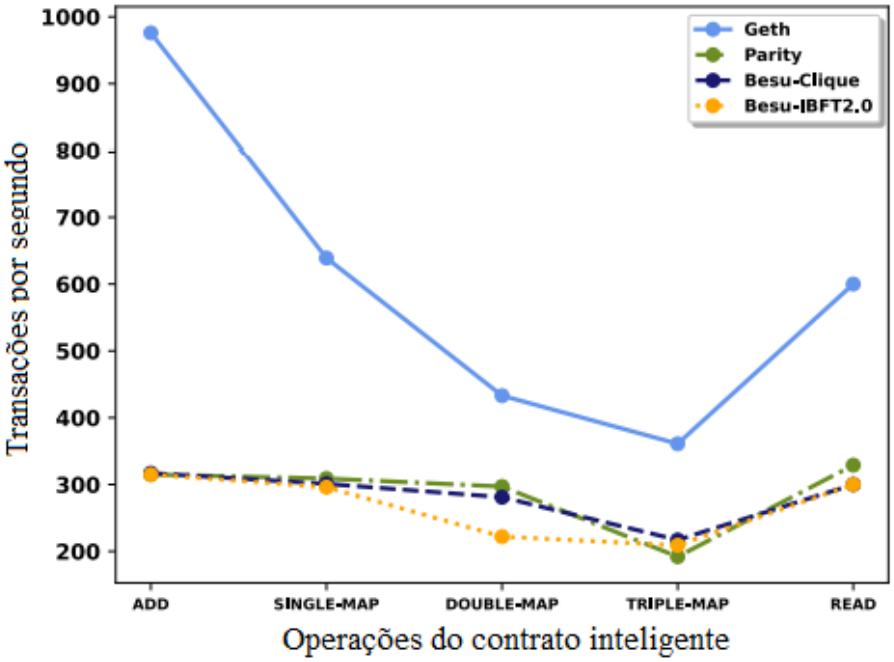
Figura 21 – Teste de variação da razão entre *gasLimit* e *number*



Fonte: Traduzido pelo próprio autor de Samuel *et al*, 2021.

Samuel *et al* (2021) validou adicionalmente o impacto de diferentes transações realizadas por contratos inteligentes, baseando-se na complexidade operacional e custo de *gas*, e a variação da performance da rede relativa à adição de novos nodos (teste de escalabilidade). Em relação as transações, foi delimitada uma razão de *gasLimit* e *number* de 6.000 mantendo o melhor caso operacional, o qual possibilitou validar que a performance segue uma correlação inversa à complexidade das operações em todos os clientes. Independentemente da operação realizada, o cliente mais eficiente se manteu o *Geth*. Este comportamento pode ser visualizado na Figura 22.

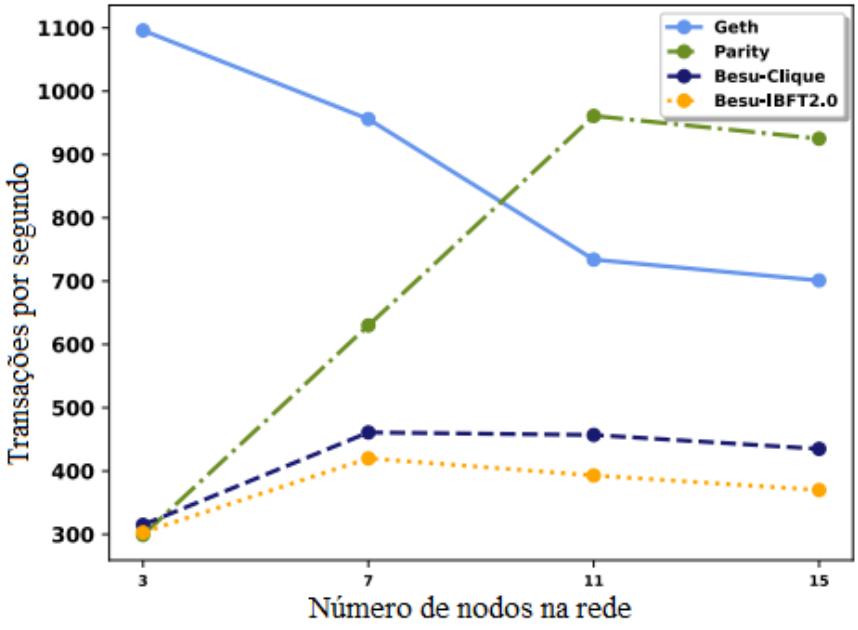
Figura 22 – Teste de variação de complexidade de operações



Fonte: Traduzido pelo próprio autor de Samuel *et al*, 2021.

Para o teste de escalabilidade, foi mantida a razão de 6.000 e a operação escolhida para os contratos foi o *ADD*. Neste caso, nota-se uma considerável queda de eficiência ao acréscimo de novos nodos no cliente *Geth*, entretanto este manteve-se como o mais eficiente entre todos os clientes até a adição de 11 nodos na rede, sendo superado pelo *Parity* a partir daí. Os clientes *Besu* tiveram leves acréscimos no número de transações por segundo até 7 nodos, adicionais nodos já geraram queda na quantidade de transações.

Figura 23 – Teste de escalabilidade



Fonte: Traduzido pelo próprio autor de Samuel *et al*, 2021.

O protocolo de consenso distribuído escolhido para a implementação da rede privada foi o QBFT implementado no cliente *Besu*, os quais não foram utilizados nas validações realizadas por Samuel *et al* (2021). Entretanto, pôde-se extrapolar os dados obtidos para o modelo IBFT 2.0 implementado no cliente *Besu*, visto que o protocolo QBFT é um aprimoramento do IBFT 2.0 com funcionamento sistêmico praticamente idêntica.

A atual quantidade de transações por segundo do cliente utilizado foi ineficiente para a aplicação de todas as melhorias propostas, e mesmo que fosse utilizado o cliente *Geth*, em um contexto industrial complexo, seria muito improvável que todas as melhorias operassem eficientemente em conjunto. Contudo, empresas de pequeno e médio porte possuem uma maior facilidade para implementação, visto a menor quantidade de sensores, controladores e equipamentos industriais.

As melhorias propostas de autenticação e autorização *in loco*, controle de qualidade e de manufatura puderam ser implementadas com uma alta taxa de eficiência visto a baixa quantidade de transações por segundo que estas realizam. Não obstante, a melhoria de privacidade acaba sendo ineficiente *a priori*, visto a necessidade de baixa latência e múltiplas atualizações de sensores concorrentes por segundo, ultrapassando os limites operacionais da rede privada.

As melhorias instanciadas na rede pública do *Ethereum* não possuem muitos dificultadores, porém a privacidade de dados não pôde ser coberta. No modelo de autenticação e autorização, isto pouco importa visto que o acesso é restrito somente a operações de leitura, entretanto nos demais modelos, dados transacionais e de segredo industrial puderam ser impropriamente visualizados. Existem soluções presentes no mercado, como o *Tessera*, que poderiam possibilitar a totalidade de privacidade aos dados, entretanto, dada a limitação temporal do trabalho, estas não conseguiram ser abordadas.

É válido ressaltar que o *Ethereum* possui a atualização chamada de *Ethereum* 2.0 que pretende resolver a baixa quantia de transações por segundo que a rede suporta atualmente e pretende substituir o atual protocolo PoW pelo PoS. Ao que compete as modificações propostas por este trabalho, a atualização chamada de *Shard Chains*, a qual introduz a operação de *sharding*, possibilitará a quebra da rede em redes secundárias, aumentando exponencialmente a quantidade de transações por segundo que a rede principal consegue realizar. Esta modificação também possibilitará a instanciação de contratos inteligentes em diferentes níveis de rede, aumentando não somente a eficiência da rede, mas da execução dos códigos de contratos.

5 CONCLUSÃO

O presente trabalho visou propor a integração de DLTs ao contexto da Indústria 4.0 através de melhorias funcionais incrementais e um modelo arquitetural de alto nível. A DLT escolhida foi o *blockchain* do *Ethereum*, levando em consideração as funcionalidades presentes na tecnologia, como contratos inteligentes, e a facilidade de configuração. A partir desta escolha, as melhorias funcionais basearam-se no modelo arquitetural e nas tecnologias expostas por Kasireddy (2021a).

A partir da implementação, teste e validação destas melhorias, foi possível verificar benefícios em privacidade, segurança e funcionalidade pela integração de DLTs, contratos inteligentes e uma abordagem mais descentralizada aos processos industriais. Estes benefícios são introduzidos de maneira segura, com baixo custo de implementação e manutenção e são facilmente escaláveis.

Foi possível visualizar também, que DLTs se posicionam como tecnologias indispensáveis ao futuro da Indústria 4.0 e ao mercado como um todo. Deste modo, modelos arquiteturais e definições funcionais são extremamente importantes neste momento de validação tecnológica tanto da Indústria 4.0 como das DLTs. O modelo desenvolvido em questão, mesmo que abstraído e de alto nível, serve já como uma fundamentação inicial de posicionamento tecnológico dentro do contexto industrial e deve ser iterado inúmeras vezes, até a maturação da tecnologia.

Entretanto a complexidade de implementação, o processo presente de maturação das tecnologias, o baixo número de transações por segundo e a falta de definições gerais arquiteturais dificultam a disseminação de sua aplicação. Esta realidade expõe a necessidade de maiores estudos e aprimoramentos funcionais em todas as tecnologias que compõem a Web 3.0. Aprimoramentos no desempenho e eficiência da rede e na abrangência funcional da linguagem *Solidity*, são pilares para a maturação e solidificação da tecnologia como uma opção viável a longo prazo. Ademais, a falta de flexibilidade para a resolução de *bugs* em contratos inteligentes, a baixa presença de documentação tecnológica e a falta de mão de obra qualificada, explicitam a necessidade de mais tempo, estudos e aplicações práticas às tecnologias.

A partir dos dados obtidos, é possível definir que a implementação de tecnologias presentes no novo modelo da internet chamado de Web 3.0 ao contexto industrial, é majoritariamente benéfico e agrega valor de maneira significativa aos mais diversos setores industriais. Entretanto, sem novas pesquisas, implementações e desenvolvedores qualificados, estas tecnologias podem tornar sistemas industriais altamente ineficientes e inviáveis. Para

evitar esta ocorrência, se faz necessário a criação de modelos arquiteturais e funcionais que sejam capazes de disseminar o conhecimento e minimizar pontos de falha de implementação.

Juntamente a isto, a implementação modular de melhorias possibilita uma aproximação desacelerada aos conceitos presentes na Web 3.0 e nas DLTs, podendo facilitar as suas implementações.

É proposto, em um próximo momento, realizar implementações mais complexas e *in loco* das melhorias modeladas, de modo em que possa se verificar de maneira assertiva os limitadores funcionais, principalmente em relação ao tempo de resposta e a viabilidade de uso dada a quantia de transações por segundo atual. De mesmo modo, propõe-se que sejam abordadas novas tecnologias para a implementação deste modelo, em especial o *Tangle*, da instituição IOTA, e o *Tessera*. Deste modo, será possível comparar quantitativamente não somente as diferentes abordagens do *Ethereum*, mas também tecnologias presentes no mercado. Além disto, se propõe a revisitação da implementação destas melhorias quando a rede do *Ethereum* seja atualizada para a chamada *Ethereum* 2.0.

REFERÊNCIAS

ALBUQUERQUE, Bruno Saboia de; CALLADO, Marcelo de Castro. **Understanding Bitcoins: Facts and Questions**. Revista Brasileira de Economia, Rio de Janeiro, v. 69, n. 1, jan./mar. 2015. DOI https://doi.org/10.5935/0034-7140.20150001. Disponível em: http://old.scielo.br/scielo.php?script=sci_arttext&pid=S0034-71402015000100003. Acesso em: 21 abr. 2022.

AL-JAROODI, Jameela *et al*. **Industrial Applications of Blockchain**. IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC), [*s. l.*], p. 550-555, 9 jan. 2019. DOI doi: 10.1109/CCWC.2019.8666530. Disponível em: https://ieeexplore.ieee.org/abstract/document/8666530. Acesso em: 29 mar. 2022.

ALLADI, Tejasvi *et al*. **Blockchain Applications for Industry 4.0 and Industrial IoT: A Review**. IEEE Access, [*s. l.*], v. 7, p. 176935-176951, 29 nov. 2019. DOI 10.1109/ACCESS.2019.2956748. Disponível em: https://ieeexplore.ieee.org/document/8917991. Acesso em: 29 mar. 2022.

AMRUTIYA, Varun *et al*. **Trustless Two-Factor Authentication using Smart Contracts in Blockchains**. 2019 International Conference on Information Networking (ICOIN), [*S. l.*], p. 66-71, 11 jan. 2019. DOI 10.1109/ICOIN.2019.8718198. Disponível em: https://ieeexplore.ieee.org/document/8718198. Acesso em: 31 maio 2022.

ASIF, Muhammad *et al*. Blockchain-Based Authentication and Trust Management Mechanism for Smart Cities. **Sensors**, [*s. l.*], 29 mar. 2022. DOI https://doi.org/10.3390/s22072604. Disponível em: https://www.mdpi.com/1424-8220/22/7/2604. Acesso em: 31 maio 2022.

BACK, Adam. **Hashcash - A Denial of Service Counter-Measure**. [S. l.], 1 ago. 2002. Disponível em: http://www.hashcash.org/papers/hashcash.pdf. Acesso em: 14 abr. 2022.

BAHETI, Radhakisan; GILL, Hellen. The impact of control technology. **Cyber-physical Systems**, [*s. l.*], v. 12, ed. 1, p. 161-166, 2011. Disponível em: https://www.researchgate.net/profile/Mohamed-Mourad-Lafifi/post/What_is_the_difference_between_Cyber_Physical_Systems_and_Networked_Control_Systems/attachment/59d6407379197b807799caa6/AS%3A431158354812928%401479807570298/download/IoCT-Part3-02CyberphysicalSystems.pdf. Acesso em: 23 mar. 2022.

BITCOIN WIKI. **Block**. [S. l.], 13 mai. 2021. Disponível em: https://en.bitcoin.it/wiki/. Acesso em: 20 abr. 2022.

BUTERIN, Vitalik. **Ethereum: A Next-Generation Smart Contract and Decentralized Application Platform**. [*S. l.*], 2014. Disponível em: https://ethereum.org/669c9e2e2027310b6b3cdce6e1c52962/Ethereum_Whitepaper_-_Buterin_2014.pdf. Acesso em: 13 abr. 2022.

CHOWDHURY, Mohammad Jabed Morshed *et al*. A Comparative Analysis of Distributed Ledger Technology Platforms. **IEEE Access**, [*s. l.*], 3 dez. 2019. DOI

10.1109/ACCESS.2019.2953729. Disponível em: https://ieeexplore.ieee.org/document/8902067. Acesso em: 31 mar. 2022.

EDITORS OF ENCYCLOPAEDIA BRITANNICA. **Industrial Revolution**. [*S. l.*], 22 mar. 2022. Disponível em: https://www.britannica.com/event/Industrial-Revolution. Acesso em: 30 mar. 2022.

ETHEREUM ORGANIZATION. **Ethereum Development Documentation**. [S. l.], 14 abr. 2022. Disponível em: https://ethereum.org/en/developers/docs/. Acesso em: 14 abr. 2022.

FORD, Matthew W. **Supply Chain Quality Management and Environmental Uncertainty: A Contingency Perspective**. [*S. l.*], set. 2015. Disponível em: https://www.researchgate.net/publication/282327156_Supply_Chain_Quality_Management_and_Environmental_Uncertainty_A_Contingency_Perspective/citations. Acesso em: 20 abr. 2022.

GERONI, Diego. **Distributed Ledger Technology: Simply Explained**. [S. l.], 2 dez. 2020. Disponível em: https://101blockchains.com/distributed-ledger-technology/. Acesso em: 12 abr. 2022.

GILES, Martin. **Triton is the world's most murderous malware, and it's spreading**. [*S. l.*], 5 mar. 2019. Disponível em: https://www.technologyreview.com/2019/03/05/103328/cybersecurity-critical-infrastructure-triton-malware/. Acesso em: 5 mar. 2022.

GONZALEZ, Gloria; LEFEBVRE, Ben; GELLER, Eric. **Cyber-Physical Systems'Jugular' of the U.S. fuel pipeline system shuts down after cyberattack**. [*S. l.*], 8 maio 2021. Disponível em: https://www.politico.com/news/2021/05/08/colonial-pipeline-cyber-attack-485984. Acesso em: 13 abr. 2022.

GUNES, Volkan *et al*. A Survey on Concepts, Applications, and Challenges in Cyber-Physical Systems. **KSII TRANSACTIONS ON INTERNET AND INFORMATION SYSTEMS**, [*S. l.*], v. 8, n. 12, p. 4242-4268, 31 dez. 2014.

HAAR, Ryan. **What is Bitcoin?** [S. l.], 19 abr. 2022. Disponível em: https://time.com/nextadvisor/investing/cryptocurrency/what-is-bitcoin/. Acesso em: 27 abr. 2022.

HU, Fei; VASQUEZ, Roger; PATTERSON, Cameron. Integrated Computing and Engineering Design. In: HU, Fei. **Cyber-Physical Systems: Integrated Computing and Engineering Design**. [S. l.: s. n.], 2014. cap. 2, p. 16-32.

HYPERLEDGER BESU. **Https://besu.hyperledger.org/en/stable/**. [*S. l.*], 2022. Disponível em: https://besu.hyperledger.org/en/stable/. Acesso em: 12 abr. 2022.

IBM. **Public key cryptography**. [S. l.], 3 maio 2021. Disponível em: https://www.ibm.com/docs/en/ztpf/1.1.0.14?topic=concepts-public-key-cryptography. Acesso em: 20 abr. 2022.

JAMAI, Intissar; AZZOUZ, Lamia Ben; SAIDANE, Leıla Azouz. **Security issues in Industry 4.0**. 2020 International Wireless Communications and Mobile Computing (IWCMC), [*s. l.*], v. 3, p. 481-488, 27 jun. 2020. DOI 10.1109/IWCMC48107.2020.9148447. Disponível em: https://www.sciencedirect.com/science/article/abs/pii/S221384631400025X?msclkid=ac217b2db21711ec886deb950cbd4738. Acesso em: 23 mar. 2022.

JAVAID, Mohd *et al*. **Blockchain technology applications for Industry 4.0: A literature-based review**. Blockchain: Research and Applications, [*s. l.*], v. 2, ed. 4, p. 481-488, 11 ago. 2021. DOI https://doi.org/10.1016/j.bcra.2021.100027. Disponível em: https://www.sciencedirect.com/science/article/pii/S2096720921000221. Acesso em: 29 mar. 2022.

KAGERMANN, Henning; WAHLSTER, Wolfgang; HELBIG, Johannes. **Recommendations for implementing the strategic initiative INDUSTRIE 4.0**. [*S. l.*], Acatech, abril 2013. Disponível em: https://www.din.de/blob/76902/e8cac883f42bf28536e7e8165993f1fd/recommendations-for-implementing-industry-4-0-data.pdf. Acesso em: 23 mar. 2022.

KASIREDDY, Preethi. **The Architecture of a Web 3.0 application**. [*S. l.*], 22 set. 2021. Disponível em: https://www.preethikasireddy.com/post/the-architecture-of-a-web-3-0-application. Acesso em: 13 abr. 2022.

KASIREDDY, Preethi. **The hardest concept for developers to grasp about Web 3.0**. [*S. l.*], 5 out. 2021. Disponível em: https://www.preethikasireddy.com/post/the-hardest-concept-for-developers-to-grasp-about-web-3-0. Acesso em: 13 abr. 2022.

LEE, Edward Ashford; SESHIA, Sanjit Arunkumar. About the Term "Cyber-Physical Systems". *In*: LEE, Edward Ashford; SESHIA, Sanjit Arunkumar. **Introduction to Embedded Systems**: A Cyber-Physical Systems Approach. Segunda. ed. [*S. l.*: *s. n.*], 2015. cap. 1, p. 5-5.

LEE, Jay; BAGHERI, Behrad; KAO, Hung-An. A Cyber-Physical Systems architecture for Industry 4.0-based manufacturing systems. **Manufacturing Letters 3**, [*s. l.*], v. 3, p. 18-23, 10 dez. 2014. DOI https://doi.org/10.1016/j.mfglet.2014.12.001. Disponível em: https://www.sciencedirect.com/science/article/abs/pii/S221384631400025X?msclkid=ac217b2db21711ec886deb950cbd4738. Acesso em: 1 abr. 2022.

MOHAJAN, Haradhan. Third Industrial Revolution Brings Global Development. **Munich Personal RePEc Archive**, [*s. l.*], 10 set. 2021. Disponível em: https://mpra.ub.uni-muenchen.de/110972/1/MPRA_paper_110972.pdf. Acesso em: 23 mar. 2022.

MOKYR, Joel. **The Second Industrial Revolution**, 1870-1914. [*s. p.*], [*s. l.*], ago. 1998. Disponível em: https://faculty.wcas.northwestern.edu/jmokyr/castronovo.pdf. Acesso em: 23 mar. 2022.

MULLET, Valentin *et al*. **A Review of Cybersecurity Guidelines for Manufacturing Factories in Industry 4.0**. IEEE Access, [*S. l.*], v. 9, p. 23235-23263, 27 jan. 2021. Disponível em: https://ieeexplore.ieee.org/abstract/document/9345803. Acesso em: 23 mar. 2022.

NAKAMOTO, Satoshi. **Bitcoin: A Peer-to-Peer Electronic Cash System**. [*S. l.*], 2008. Disponível em: https://bitcoin.org/bitcoin.pdf. Acesso em: 31 mar. 2022.

PEERLESS RESEARCH GROUP. **Outsourcing Manufacturing: A 20/20 View**. [*S. l.*], 9 jan. 2016. Disponível em: https://www.scmr.com/wp_content/e2open_wp_outsourcing_mfg_011316.pdf. Acesso em: 20 abr. 2022.

POPKOVA, Elena; RAGULINA, Yulia; BOGOVIZ, Aleksei. **Industry 4.0: Industrial Revolution of the 21st Century: Fundamental Differences of Transition to Industry 4.0 from Previous Industrial Revolutions**. Studies in Systems, Decision and Control, [*s. l.*], v. 169, abr. 2018. DOI https://doi.org/10.1007/978-3-319-94310-7. Disponível em: https://link.springer.com/book/10.1007/978-3-319-94310-7. Acesso em: 31 mar. 2022.

RAUCHS, Michel et al. **Distributed Ledger Technology Systems: A Conceptual Framework**. The Cambridge Centre for Alternative Finance, [s. l.], 13 ago. 2018. DOI http://dx.doi.org/10.2139/ssrn.3230013. Disponível em: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3230013. Acesso em: 12 abr. 2022.

REIFF, Nathan. **Bitcoin vs. Ethereum: What's the Difference?** [S. l.], 21 fev. 2022. Disponível em: https://www.investopedia.com/articles/investing/031416/bitcoin-vs-ethereum-driven-different-purposes.asp. Acesso em: 20 abr. 2022.

ROBERTSON, Jordan; TURTON, William. **Colonial Hackers Stole Data Thursday Ahead of Shutdown**. [*S. l.*], 8 maio 2021. Disponível em: https://www.bloomberg.com/news/articles/2021-05-09/colonial-hackers-stole-data-thursday-ahead-of-pipeline-shutdown. Acesso em: 13 abr. 2022.

RÜSSMANN, Michael *et al*. **Industry 4.0: The Future of Productivity and Growth in Manufacturing Industries**. [*S. l.*]: Boston Consulting Group, 9 abr. 2015. Disponível em: https://www.bcg.com/pt-br/publications/2015/engineered_products_project_business_industry_4_future_productivity_growth_manufacturing_industries. Acesso em: 23 mar. 2022.

SAMUEL, Cyril Naves *et al*. Choice of Ethereum Clients for Private Blockchain: Assessment from Proof of Authority Perspective. **2021 IEEE International Conference on Blockchain and Cryptocurrency**, [*s. l.*], 24 jun. 2021. DOI 10.1109/ICBC51069.2021.9461085. Disponível em: https://ieeexplore.ieee.org/document/9461085. Acesso em: 18 maio 2022.

SCHWAB, Klaus. **The Fourth Industrial Revolution: What It Means and How to Respond**. [*S. l.*], Foreign Affairs, 12 dez. 2015. Disponível em: https://www.foreignaffairs.com/articles/2015-12-12/fourth-industrial-revolution. Acesso em: 23 mar. 2022.

SOLIDITY. **Solidity Documentation**. [S. l.], 15 fev. 2022. Disponível em: https://docs.soliditylang.org/en/v0.8.13/index.html. Acesso em: 14 abr. 2022.

SUKHODOLOV, Yakov A. The Notion, Essence, and Peculiarities of Industry 4.0 as a Sphere ofIndustry. **Studies in Systems, Decision and Control**, [*s. l.*], v. 169, abr. 2018. DOI

https://doi.org/10.1007/978-3-319-94310-7. Disponível em: https://link.springer.com/book/10.1007/978-3-319-94310-7. Acesso em: 31 mar. 2022.

THORBECKE, Catherine. **Gas hits highest price in 6 years, fuel outages persist despite Colonial Pipeline restart**: Fallout from the Colonial Pipeline cyberattack persists Monday. [*S. l.*], 17 maio 2021. Disponível em: https://abcnews.go.com/US/gas-hits-highest-price-years-fuel-outages-persist/story?id=77735010. Acesso em: 13 abr. 2022.

WILDAN, M; PRIBADI, Firman. **Supply Chain Quality Management (SCQM) Practice and its Impact on Company Operational Performance Achievement**. [*S. l.*], mai. 2020. Disponível em: https://www.researchgate.net/publication/351935006_Supply_Chain_Quality_Management_SCQM_Practice_and_its_Impact_on_Company_Operational_Performance_Achievement. Acesso em: 20 abr. 2022.