**Capítulo 16**

**Blockchain Empresarial**

Neste capítulo, investigaremos blockchains empresariais. Qual é a arquitetura padrão dos blockchains empresariais? Por que eles são necessários? Responderemos a essas perguntas e também tentaremos responder à grande questão de por que blockchains públicos atuais não são necessariamente uma escolha adequada para casos de uso empresariais.

Também apresentaremos várias plataformas de blockchain empresarial, incluindo o Quorum. Ao longo do caminho, abordaremos os seguintes tópicos:

* Soluções empresariais e blockchain
* Fatores limitantes
* Requisitos
* Blockchain empresarial versus blockchain público
* Casos de uso de blockchains empresariais
* Arquitetura de blockchain empresarial
* Projetando soluções de blockchain empresarial
* Blockchain na nuvem
* Blockchains empresariais atualmente disponíveis
* Desafios do blockchain empresarial
* Quorum
* VMware Blockchain
* Configurando o Quorum com IBFT
* Outros projetos Quorum

Discutimos vários tipos diferentes de blockchain no Capítulo 1, Blockchain 101, incluindo blockchains permissionados, públicos, privados e de consórcio. Blockchains empresariais são um tipo de cadeia de consórcio permissionada que atende principalmente aos requisitos empresariais.

Permissionado não significa privado. Blockchains permissionados também podem ser públicos e permitir acesso apenas a participantes conhecidos. Blockchains empresariais geralmente são privados e permissionados e são executados entre membros do consórcio.

**Blockchain Empresarial**

Embora blockchains públicos forneçam garantias de integridade, consistência, imutabilidade e segurança, eles carecem de certos recursos, o que os torna menos adequados para uso empresarial. Discutiremos essas limitações em detalhes em breve.

Primeiramente, veremos o que são soluções empresariais e como o blockchain pode se encaixar em uma empresa. Em segundo lugar, veremos quais perguntas devem ser respondidas antes de introduzir uma solução de blockchain empresarial em um negócio.

**Soluções empresariais e blockchain**

Soluções empresariais integram diferentes fragmentos de um negócio e permitem que ele atinja seus objetivos, fornecendo informações críticas para os stakeholders. Aqui, consideraremos a questão: o blockchain se encaixa nessa definição e ajuda a alcançar objetivos empresariais?

Para descobrir se um blockchain é adequado para uma empresa ou não, podemos fazer algumas perguntas:

* Como os processos empresariais podem ser melhorados usando a tecnologia blockchain?
* Eu realmente preciso de um blockchain empresarial? Para isso, podemos fazer algumas perguntas para racionalizar se precisamos ou não de uma solução de blockchain empresarial. Meu caso de uso:
  + Precisa de dados compartilhados entre os participantes?
  + Tem participantes de outras organizações que não são necessariamente confiáveis e têm interesses conflitantes?
  + Precisa de auditoria rigorosa? Um blockchain pode fornecer isso, pois é uma cadeia de registros imutável e à prova de adulterações que pode fornecer uma trilha de auditoria definitiva das atividades realizadas na cadeia (sistema empresarial).
  + Precisa garantir a confidencialidade das transações?
  + Precisa garantir o anonimato dos participantes?
  + Precisa de atualizações controladas, mas transparentes, no livro razão?
  + Não precisa de uma autoridade confiável única? Em vez disso, as decisões na rede (atualizações no livro razão) devem ser conduzidas por membros do consórcio e acordadas entre os membros?

Se a resposta a qualquer uma das perguntas anteriores for sim, então usar uma solução de blockchain empresarial pode ser uma boa opção. Caso contrário, um banco de dados tradicional pode oferecer uma alternativa melhor. Além das perguntas mencionadas aqui, ao propor uma solução de blockchain empresarial, também precisamos responder a outras perguntas do ponto de vista empresarial que ajudam a estabelecer a visão geral e a estratégia da implementação da solução blockchain:

* Qual é o objetivo geral da solução blockchain proposta? Está alinhado com os objetivos de negócios da empresa? É um projeto de Prova de Conceito (PoC), destinado apenas a demonstrar uma ideia, ou um projeto de produção com entregas reais de negócios?
* Onde será implantado? Nuvem ou hospedagem local? Quem gerenciará a solução blockchain uma vez implantada?
* **Considerações de gerenciamento de risco** — a solução segue alguma diretriz estabelecida para gerenciamento de risco? Por exemplo, o NIST 800-37 (<https://csrc.nist.gov/publications/detail/sp/800-37/rev-2/final>) é uma estrutura de gerenciamento de risco que fornece um processo usado para gerenciar riscos de segurança e privacidade para sistemas e organizações de TI.
* Existem outros projetos que essa organização já pode ter implementado usando blockchain empresarial? Podemos aprender com experiências anteriores e aproveitar alguns dos recursos e práticas recomendadas que podem ter sido desenvolvidos anteriormente?

Essas perguntas devem ser — e geralmente são — feitas ao desenvolver qualquer outra solução empresarial, mas em relação ao blockchain, essas questões se tornam ainda mais críticas devido à natureza nascente e imatura da tecnologia blockchain empresarial. Uma definição clara dos objetivos, juntamente com um alinhamento claro com os requisitos do negócio, resultará em uma implementação que atenda às metas empresariais.

A seguir, imaginaremos alguns fatores de sucesso que podem ajudar os blockchains empresariais a se tornarem bem-sucedidos e adotados nas empresas.

**Fatores de sucesso**

Existem alguns fatores orientados ao negócio que devem ser abordados para uma solução de blockchain empresarial bem-sucedida.

De suma importância é o requisito de que a solução blockchain deve trazer algum valor econômico e ajudar a alcançar metas reais de negócio. Além disso, as soluções empresariais devem estar alinhadas com os objetivos comerciais.

O principal valor dos blockchains empresariais está na propriedade de ser um livro razão compartilhável, replicável e permissionado entre organizações, o que resulta imediatamente em redução de custos ao eliminar a necessidade de troca de dados. Ao fazer isso, também eliminamos a necessidade de infraestrutura e ferramentas para suportar tais trocas. Além disso, devido à segurança, imutabilidade e auditoria fornecidas como recursos inerentes de um blockchain, não há necessidade de investir separadamente nesses requisitos.

Soluções blockchain que se integram fácil/nativamente com sistemas existentes oferecem melhor valor porque sistemas legados já maduros (pelo menos nesta fase) fornecem um mecanismo para se conectar ao blockchain, ler seus dados e salvá-los em um formato conhecido com o qual a empresa já esteja familiarizada. Além disso, apenas uma rede blockchain por si só não é inteiramente útil nas empresas se não for integrada com sistemas administrativos existentes, como sistemas de Planejamento de Recursos Empresariais (ERP), bancos de dados de backend, sistemas de reconciliação de registros ou outras ferramentas de relatórios organizacionais.

Uma solução de blockchain empresarial deve ser vista como uma solução empresarial completa de ponta a ponta como parte de uma arquitetura empresarial maior, em vez de apenas uma rede blockchain isolada. Exploraremos esse tópico mais adiante neste capítulo, na seção “Projetando soluções de blockchain empresarial”.

Governança, controle e segurança de nível empresarial também são recursos desejáveis do ponto de vista do negócio, pois permitem que as partes interessadas apliquem regras e políticas organizacionais já estabelecidas ao blockchain. Isso também pode ajudar a alcançar requisitos regulatórios e de conformidade.

**Fatores limitantes**

Discutimos vários benefícios da tecnologia blockchain em geral no Capítulo 1, Blockchain 101. Embora todos esses benefícios sejam alcançáveis usando blockchains públicos, diversos recursos estão ausentes nesses blockchains, o que os torna inadequados para casos de uso empresariais. O interesse em blockchains empresariais surge dessas limitações dos blockchains públicos, juntamente com requisitos específicos de qualquer negócio.

Descreveremos algumas das preocupações mais comuns a seguir:

* **Desempenho lento**: Blockchains públicos são lentos e podem processar apenas algumas transações por segundo. O Bitcoin processa de 3 a 4 transações por segundo, enquanto o Ethereum processa cerca de 14. Essa baixa taxa de transações não é adequada para empresas que geralmente exigem alta velocidade de transação. Por exemplo, empresas de pagamentos com cartão normalmente precisam processar milhares de transações por segundo.
* **Falta de governança de acesso**: Blockchains públicos estão disponíveis para qualquer um participar, o que facilita para investidores e entusiastas de criptomoedas aderirem e ajuda com o efeito de rede. No entanto, em uma empresa, todos os participantes devem ser conhecidos para que todos saibam com quem estão lidando. Essa falta de um mecanismo de controle de acesso e identificação torna os blockchains públicos bastante inadequados para os negócios.
* **Falta de privacidade**: Blockchains públicos são inerentemente transparentes, e tudo no livro razão é visível para todos. Isso significa que qualquer um pode facilmente visualizar detalhes de transações e os participantes envolvidos em uma transação. A falta de privacidade resulta em um problema de vazamento de estratégia, que descreveremos a seguir.
* **Problema de vazamento de estratégia**: Como blockchains públicos são abertos a todos, é possível que, usando análise de tráfego, alguém descubra qual parte está se comunicando com qual outra parte e, assim, possa inferir quem está fazendo mais negócios. Além disso, mesmo uma simples contagem de transações de um endereço pode ser usada para inferir a escala do negócio e, portanto, a receita que uma entidade pode estar gerando.
* **Consenso probabilístico**: Blockchains públicos tradicionais geralmente usam um mecanismo de consenso do tipo Prova de Trabalho (PoW), que é inerentemente um protocolo probabilístico que fornece finalização probabilística. Embora o mecanismo de confirmações, como discutido no Capítulo 6, Arquitetura do Bitcoin, forneça um certo nível de confiança de que uma transação é irrevogável, ainda é possível que a cadeia bifurque e as transações sejam perdidas. Essa questão é particularmente preocupante para empresas onde, uma vez que uma transação é confirmada, ela é considerada final. Imagine receber um documento de propriedade de imóvel em uma cadeia de alguém apenas para descobrir depois que o blockchain bifurcou e que você não é mais o proprietário da propriedade.
  + Para lidar com essa limitação, blockchains empresariais usam algoritmos de consenso determinísticos, que fornecem finalização imediata.
* Observe que alguns blockchains públicos mais novos, como Avalanche e Cosmos, têm finalização determinística. No entanto, tradicionalmente, cadeias públicas têm sido — e algumas ainda são — baseadas em PoW, por exemplo, o Bitcoin.
* **Taxas de transação**: No Ethereum ou outros blockchains semelhantes, uma taxa de transação é cobrada em criptomoeda nativa para cada execução de transação. Embora esse mecanismo forneça incentivos para os mineradores e proteção contra spam, não há necessidade de tal arranjo em blockchains empresariais. Se uma organização fosse usar blockchains públicos para suas transações comerciais, então ela precisaria manter criptomoeda em reserva para pagar pelas operações no blockchain. Esse custo extra pode ser indesejável para algumas empresas. Além disso, quando a rede está congestionada, as taxas de gás aumentam significativamente, o que resulta em ainda mais custos.
* **Congestionamento da rede**: É possível, em redes blockchain públicas, que devido a dApps movimentadas, meu próprio dApp seja impactado porque a rede está ocupada. Se toda a rede estiver congestionada devido a muitas transações, então todos os dApps funcionarão lentamente, e este é um risco que pode se mostrar prejudicial para um negócio. Embora as preocupações mencionadas aqui sejam consideradas limitações nos blockchains públicos, com base nessas preocupações e limitações, podemos derivar e definir vários requisitos, ou características, de um blockchain que o tornarão adequado para uso empresarial. Em outras palavras, as limitações nos blockchains públicos podem ser vistas como requisitos dos blockchains empresariais.

A próxima seção trata de vários recursos que um blockchain deve possuir para se tornar adequado às empresas.

**Requisitos**

Além de integridade e consistência, que também são fornecidas por blockchains públicos, há vários outros requisitos especificamente para blockchains empresariais que os tornam adequados para casos de uso empresariais. Em alguns casos, certos requisitos se tornam ainda mais rigorosos em blockchains empresariais em comparação com blockchains públicos. Por exemplo, em blockchains públicos, consistência eventual é aceitável. No entanto, em blockchains empresariais, no momento em que uma transação é confirmada, ela deve ser imediatamente finalizada e se tornar irrevogavelmente parte do registro global (estado). Assim, começaremos definindo brevemente integridade e consistência antes de introduzir requisitos específicos para blockchains empresariais:

* **Integridade**: Este atributo de um blockchain é fornecido pelo uso de funções hash e assinaturas digitais e desempenha um papel vital na segurança geral do blockchain. Funções hash permitem verificar quaisquer modificações nos dados, enquanto assinaturas digitais garantem que as mensagens que se originaram de um remetente não tenham sido alteradas.
* **Consistência**: Este atributo de um blockchain assegura que todos os nós honestos concordem sobre a mesma sequência de blocos. Para alcançar isso, são usados vários mecanismos, como protocolos de consenso PoW ou de Tolerância a Falhas Bizantinas (BFT).

Você pode consultar o Capítulo 1, *Blockchain 101*, para relembrar esses conceitos, pois introduzimos essas propriedades lá em mais detalhes.

Agora, apresentaremos três requisitos fundamentais que devem ser atendidos para que um blockchain se torne adequado para empresas. Esses requisitos são: **privacidade**, **desempenho** e **governança de acesso**.

**Privacidade**

A privacidade é de extrema importância em blockchains empresariais. A privacidade tem duas facetas: primeiro, confidencialidade, e segundo, anonimato.

A confidencialidade é um requisito fundamental em uma empresa. Espera-se que, em blockchains empresariais, todas as transações ocultem seus conteúdos, de modo que o valor das transações não seja revelado a ninguém que não tenha conhecimento da transação.

Transações privadas podem ser definidas como transações que atendem aos requisitos de privacidade de um caso de uso empresarial. Existem dois tipos de transações privadas, conforme definido pela Enterprise Ethereum Alliance (EEA):

* **Transações privadas restritas**: Esse tipo de transação privada é transmitido apenas às partes na rede blockchain que têm conhecimento da transação.
* **Transações privadas irrestritas**: Sob o paradigma de transação privada irrestrita, transações privadas são transmitidas a todos os participantes da rede, independentemente de terem ou não conhecimento da transação. O conteúdo ainda é criptografado e confidencial, mas a própria transação é transmitida para toda a rede.

Existem muitas abordagens para alcançar a privacidade em blockchains empresariais e em blockchain em geral. Esses métodos variam desde mecanismos fora da cadeia (off-chain), como gerenciadores de privacidade (usados no Quorum e em algumas outras cadeias empresariais), até o uso de provas de conhecimento zero (ZKPs), hardware confiável e computação multipartidária segura (SMPC). Abordaremos algumas dessas técnicas no próximo capítulo, Capítulo 17, *Escalabilidade*, mas neste capítulo nos concentraremos principalmente na privacidade baseada em gerenciadores de privacidade, onde um componente fora da cadeia é usado para fornecer serviços de privacidade. Discutiremos a abordagem baseada em gerenciadores de privacidade na seção sobre Quorum mais adiante neste capítulo.

O anonimato em blockchains empresariais pode não parecer um requisito rigoroso à primeira vista, pois todos os participantes são conhecidos e identificados. No entanto, ele é necessário para cenários com alguns participantes concorrentes realizando negócios na cadeia. Pode ser essencial, por exemplo, que em um cenário onde as partes X e Y estão transacionando entre si, a parte Z não descubra quais partes estão transacionando. Mesmo que os valores das transações não estejam visíveis, eles ainda podem revelar detalhes sobre o negócio que as duas partes estão fazendo. Se essa informação estiver disponível publicamente, então outras partes na cadeia do consórcio ganharão inteligência de mercado e poderão tentar influenciar esse processo por meio de marketing ou outros métodos.

**Desempenho**

Devido às exigências de alta velocidade nos negócios, blockchains empresariais devem ser capazes de processar transações a uma taxa elevada.

O desempenho tem duas facetas: **escalabilidade** e **velocidade**. Velocidade lida com quantas transações podem ser processadas em um determinado intervalo de tempo. Diz respeito à capacidade de um sistema lidar com um grande volume de transações a uma velocidade aceitável.

Por outro lado, temos o número de participantes em um sistema. Blockchains públicos podem suportar um grande número de usuários. Isso é especialmente verdadeiro em blockchains de criptomoedas, como Bitcoin e Ethereum.

Isso é possível devido aos algoritmos de consenso PoW ou Prova de Participação (PoS) usados nesses blockchains públicos. No entanto, em blockchains empresariais, uma classe diferente de algoritmos de consenso (mais comumente, BFT) é usada, os quais não funcionam bem com um grande número de nós. Isso resulta em limitação do número de usuários em uma cadeia de consórcio.

Aqui, devemos considerar uma troca. Idealmente, um blockchain empresarial deveria ter bom desempenho com um grande número de usuários; no entanto, dada essa troca, blockchains empresariais devem ser capazes de processar transações a uma alta taxa. É nisso que a maioria dos blockchains empresariais se concentra, já que, em muitos casos de uso, o número de nós não é tão alto, e a velocidade pode ser priorizada em detrimento da escalabilidade.

Em blockchains públicos, às vezes, a congestão da rede causada por altos volumes de tráfego pode gerar problemas de desempenho e aumentar os tempos de processamento de transações. Isso é prejudicial para dApps empresariais, que precisam de processamento de transações e tempos de resposta mais rápidos. A congestão da rede também resulta em aumento nos preços do gás, o que também pode ser uma preocupação para empresas do ponto de vista de custos.

**Governança de acesso**

Por outro ângulo, sendo um blockchain permissionado, o controle de acesso de nível empresarial (na forma de um novo mecanismo na cadeia ou controle conduzido por um SSO empresarial já existente) é um requisito fundamental nos blockchains empresariais. Como todos os participantes devem ser identificáveis em uma cadeia de consórcio, é essencial construir um mecanismo de controle de acesso que facilite esse processo. Esse recurso pode ser alcançado usando mecanismos de controle de acesso de nível empresarial, como o Controle de Acesso Baseado em Funções (RBAC).

O mecanismo de controle de acesso também pode abordar requisitos de Conheça Seu Cliente (KYC).

Além de privacidade, desempenho e governança de acesso — que geralmente são identificados como três requisitos fundamentais dos blockchains empresariais —, também há alguns outros requisitos que são altamente desejáveis, mas podem ser considerados um tanto opcionais, pois são mais dependentes do caso de uso. Descreveremos esses requisitos a seguir.

**Requisitos adicionais**

Nesta seção, apresentaremos alguns requisitos adicionais que são muito úteis e podem aumentar a adequação/eficácia das soluções de blockchain empresarial.

**Conformidade**

Uma preocupação comum é a conformidade. Blockchains públicos não são adequados para casos de uso empresariais devido a exigências legais e regulatórias rígidas em quase todos os setores, como financeiro, de saúde e governamental. Os desafios de conformidade incluem principalmente conformidade regulatória, conformidade com padrões, soberania de dados e preocupações com responsabilidade.

Vamos definir esses pontos brevemente a seguir:

* **Conformidade com padrões e regulamentações**: Frequentemente, em casos de uso empresariais, é exigida a conformidade com um padrão técnico ou leis. Por exemplo, a conformidade com o GDPR é obrigatória na União Europeia e no Espaço Econômico Europeu. Outro exemplo é a conformidade com as regulamentações da Autoridade de Conduta Financeira (FCA) no Reino Unido. Além disso, pode ser necessário, em certos casos de uso, cumprir padrões técnicos como os padrões de criptografia publicados pelo NIST. Um exemplo é o uso de curvas aprovadas pelo NIST, como descrito aqui:  
  <https://csrc.nist.gov/Projects/elliptic-curve-cryptography>
* **Soberania de dados**: A soberania de dados é um tópico amplo que, em essência, sujeita os dados às leis do país em que eles estão localizados. Por exemplo, sob o GDPR, a transferência de dados pessoais para fora da UE está sujeita a decisões de adequação (Artigo 45) e salvaguardas apropriadas (Artigo 46).

Como blockchains públicos são sistemas sem fronteiras e geograficamente distribuídos, o cumprimento de tais regulamentos pode ser desafiador.

* **Responsabilidade**: Em sistemas empresariais e de TI tradicionais, a responsabilidade legal muitas vezes recai sobre uma parte que presta um serviço específico; por exemplo, um provedor de serviços em nuvem é responsável por lidar com dados de acordo com as leis e regulamentações locais. Em um blockchain público descentralizado, os dados estão em um blockchain público e, portanto, torna-se desafiador manter uma única parte responsável pela gestão dos dados ou pela prestação dos serviços. Em caso de incidentes maliciosos, novamente, não é possível responsabilizar uma única parte.

Essa limitação representa um desafio significativo em ambientes empresariais onde, normalmente, uma parte responsável está no controle da prestação de serviços e pode ser responsabilizada. Por outro lado, sabemos que, em sistemas tradicionais, em caso de qualquer problema, o sistema jurídico pode ajudar. No entanto, em um blockchain descentralizado, pode se tornar bastante difícil culpar uma única parte por suas ações. Portanto, em blockchains empresariais, a capacidade de cumprir regulamentos e leis se torna um atributo muito desejado e pode ser alcançada por meio de identificação apropriada, KYC e mecanismos de controle de acesso. Em uma cadeia empresarial, os participantes são conhecidos e podem ser responsabilizados por suas ações.

**Interoperabilidade**

À medida que o ecossistema de blockchain empresarial evolui, também surge a necessidade de poder trocar dados entre blockchains empresariais e públicos distintos. A falta de padronização também agrava esse problema; no entanto, esforços de padronização estão em andamento, como a EEA (discutida na seção sobre desafios do blockchain empresarial). Também há soluções de interoperabilidade sendo desenvolvidas e disponíveis para blockchains que permitem interoperabilidade entre cadeias, como ION, protocolo IBC, Polkadot e Interledger.

**Integração**

Nenhum blockchain empresarial é uma solução isolada de ponta a ponta. Ele precisa se integrar com sistemas empresariais existentes ou outros sistemas fora da cadeia que fazem parte da solução empresarial completa para atingir os objetivos de negócios. Portanto, blockchains empresariais devem fornecer interfaces para integração. Isso pode ser tão simples quanto fornecer endpoints RPC ou tão complexo quanto construir conectores e plugins específicos para blockchain que se integrem a barramentos de serviço empresariais ou a outros sistemas legados — mais sobre isso na seção “Arquitetura de blockchain empresarial”.

A integração com dispositivos de segurança, como Módulos de Segurança de Hardware (HSMs), também é bastante desejável para muitos casos de uso empresariais em que segurança rigorosa é exigida, ou por exigências regulatórias ou de conformidade.

**Facilidade de uso**

Normalmente, sistemas empresariais são fáceis de implantar e usar. A implantação em empresas é fácil e rápida, e muitas vezes depende de estruturas e ferramentas maduras, como Ansible e outras ferramentas proprietárias, mas esse não é o caso com blockchain.

A implantação de sistemas empresariais é uma área bem estudada, compreendida e madura. Com ferramentas de orquestração empresarial e técnicas estabelecidas ao longo dos anos, a implantação empresarial tornou-se fácil. No entanto, blockchains não são tão fáceis de implantar quanto outros sistemas empresariais. Com a disponibilidade de Blockchain como Serviço (BaaS) e ferramentas de automação de implantação, isso está mudando. No entanto, ainda há trabalho a ser feito.

**Monitoramento**

O monitoramento com ferramentas de visualização desempenha um papel vital em qualquer solução empresarial. Sem a capacidade de monitorar e visualizar um sistema, é quase impossível garantir a integridade do sistema empresarial. Também é um recurso desejável em soluções de blockchain empresarial poder visualizar a rede blockchain. Monitorar um blockchain permite que um administrador acompanhe a saúde da rede e suas operações. Permite a um administrador monitorar e responder a eventos de interesse, como um nó caindo, lentidão no link de comunicação, um nó sendo incapaz de sincronizar com o blockchain e muitos outros cenários.

**Cálculo seguro fora da cadeia**

Em alguns cenários, é desejável poder descarregar alguns cálculos intensivos para sistemas fora da cadeia; por exemplo, se houver um requisito para fazer algum cálculo que exija recursos de Computação de Alto Desempenho (HPC).

Isso se sobrepõe um pouco com a integração, mas é mencionado aqui separadamente devido a requisitos específicos de segurança que os cálculos fora da cadeia devem ter — eles precisam ser comprovadamente corretos, com garantias de integridade e autenticidade.

**Melhores ferramentas**

Normalmente, em sistemas empresariais, há muitas ferramentas e utilitários de suporte agrupados com o produto principal para operar o software. Por exemplo, incluem-se ferramentas de administração, ferramentas de implantação, utilitários para desenvolvedores, ferramentas de visualização, ferramentas de gerenciamento e ferramentas para usuários finais. Plataformas de blockchain com melhor suporte de ferramentas são muito mais desejáveis devido ao melhor suporte ao usuário. Ferramentas como exploradores de blocos, módulos de administração de usuários e dApps para gerenciar contratos inteligentes são bastante úteis. Elas estão se tornando gradualmente mais maduras à medida que todo o ecossistema de blockchain cresce.

Agora que compreendemos os recursos dos blockchains empresariais, apresentaremos uma comparação entre blockchains públicos e empresariais para ajudar a entender as principais diferenças.

**Blockchain empresarial versus blockchain público**

Nesta seção, forneceremos uma comparação entre blockchains públicos e empresariais. Considere a tabela mostrada aqui, que avalia alguns pontos de comparação entre os dois tipos de blockchain. Esta não é uma lista exaustiva; no entanto, abordaremos os pontos mais importantes:

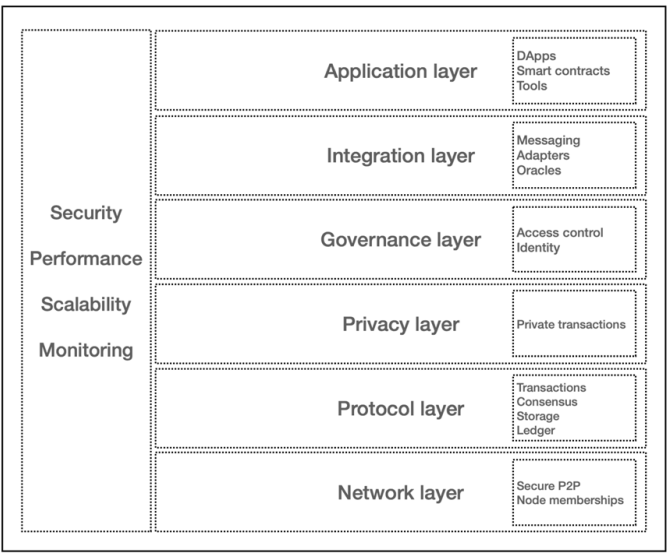
| **Aspecto** | **Cadeias públicas** | **Cadeias empresariais** |
| --- | --- | --- |
| Confidencialidade | Não. | Sim. |
| Anonimato | Não. | Sim. |
| Associação | Sem permissão. | Permissionada via votação, KYC, geralmente sob blockchain empresarial. |
| Identidade | Usuários anônimos. | Usuários conhecidos. |
| Consenso | PoW/PoS. | BFT. |
| Finalidade | Principalmente probabilística. | Requer finalização imediata/instantânea. |
| Velocidade de transação | Mais lenta. | Mais rápida (geralmente, deve ser). |
| Escalabilidade | Melhor. | Normalmente, não muito escalável, devido à escolha do consenso. |
| Conformidade regulatória | Normalmente não exigida - difícil. | Frequentemente exigida - relativamente mais fácil de atingir. |
| Confiança | Totalmente descentralizada. | Semi-centralizada, gerenciada via consórcio e mecanismos de votação. |
| Contratos inteligentes | Não são estritamente necessários (ex: Bitcoin). | Estritamente exigidos para dar suporte a funções empresariais. |

A tabela anterior compara vários aspectos importantes das plataformas públicas e empresariais. A seguir, consideraremos brevemente alguns dos casos de uso dos blockchains empresariais.

Na próxima seção, descreveremos a **arquitetura de blockchain empresarial**, que pode nos ajudar a nos comunicar com stakeholders, promover decisões iniciais de design, servir como um modelo reutilizável para desenvolvimento, construir entendimento compartilhado e compreender como o sistema é estruturado.

**Arquitetura de blockchain empresarial**

Uma arquitetura típica de blockchain empresarial contém vários elementos. Vimos uma arquitetura genérica de blockchain no Capítulo 1, *Blockchain 101*, e podemos expandi-la e modificá-la um pouco para transformá-la em uma arquitetura de blockchain empresarial que destaque os requisitos centrais de um blockchain empresarial. Esses requisitos são principalmente conduzidos pelas necessidades empresariais e pelos casos de uso:

*Figura 16.1: Arquitetura em camadas de blockchain empresarial*

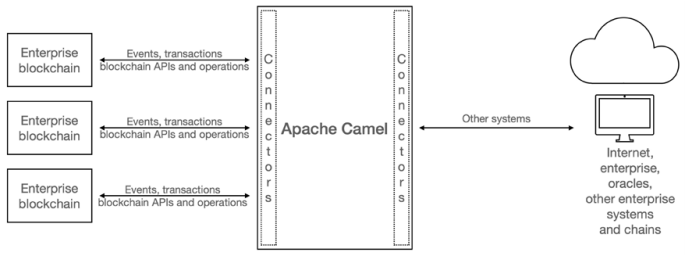
Discutiremos cada uma dessas camadas da seguinte forma:

* **Camada de rede**: A camada de rede é responsável pela implementação dos protocolos de rede, como os protocolos peer-to-peer (P2P) usados para disseminação de informações.
* **Camada de protocolo**: Esta é a camada do livro-razão propriamente dita, ou camada de blockchain, onde os elementos centrais de consenso, gerenciamento de transações e armazenamento são implementados.
* **Camada de privacidade**: Esta camada é responsável por fornecer um dos principais recursos do blockchain empresarial: a privacidade. Existem várias formas de alcançar privacidade, incluindo gerenciadores de privacidade off-chain, conhecimento zero (zero knowledge) e privacidade assistida por hardware. A privacidade baseada em hardware geralmente é suportada usando ambientes de execução confiáveis (TEEs). Por outro lado, privacidade baseada em software ou algoritmos, como as ZKPs, também são bastante comuns em blockchains empresariais.
* **Camada de governança**: Esta camada é responsável por fornecer o mecanismo de controle de acesso de nível empresarial que controla a associação à rede de consórcio. Isso pode ser controlado por um sistema de permissões on-chain implementado em contratos inteligentes, como parte do cliente de software, ou pode ser integrado a sistemas de permissão empresariais off-chain existentes, como Single Sign-On (SSO) ou Active Directory (AD).
* **Camada de integração**: Esta camada fornece APIs e um mecanismo utilizado para integrar com sistemas legados ou sistemas administrativos existentes. Pode ser algo simples como uma camada RPC que fornece APIs sobre RPC ou constituir conectores e plugins integrados para integração com o barramento de serviço empresarial.
* A camada de integração é responsável por garantir a integração com sistemas administrativos, legados e outros sistemas off-chain existentes. Ela não faz parte do protocolo central, mas, como parte da visão holística da solução blockchain de ponta a ponta, essa camada é vital para a entrega de resultados comerciais. Embora existam muitas técnicas disponíveis, uma estrutura comum de integração usada em ambientes empresariais é o Apache Camel. Isso também pode ser usado em soluções blockchain, pois vem com o componente de biblioteca Web3J para Ethereum.

Apache Camel é uma estrutura de integração empresarial de código aberto. Ele permite integração fácil entre vários sistemas usando padrões de integração empresarial. Possui centenas de componentes para diferentes sistemas, como bancos de dados, APIs e MQs para integração fácil entre sistemas.

Por exemplo, o conector Web3J (conector Ethereum do Apache Camel) é um conector rico em recursos disponível no Apache Camel que permite integração entre cadeias Ethereum e outros sistemas. Mais detalhes sobre o componente Web3J estão disponíveis aqui:  
<https://camel.apache.org/components/latest/web3j-component.html>

O conector Ethereum do Apache Camel funciona com nós Ethereum Geth, Quorum, Parity e Ganache. Ele oferece suporte à API JSON RPC via HTTP e IPC, com implementações para diferentes operações blockchain como net, eth, shh e assim por diante. Ele oferece suporte a filtros Ethereum, ao Ethereum Name Service (ENS) e à API JSON RPC, e é também uma solução totalmente testada (testes unitários e de integração).

*Figura 16.2: Design de alto nível do Apache Camel, que permite integração com blockchain*

* **Camada de aplicação**: A camada de aplicação, como o nome sugere, consiste em dApps, contratos inteligentes, ferramentas e outros softwares relevantes para dar suporte a casos de uso empresariais.
* **Segurança, desempenho, escalabilidade, monitoramento**: No lado esquerdo do diagrama de arquitetura (Figura 16.1), segurança, escalabilidade, desempenho e monitoramento são mostrados. Como cada camada no blockchain empresarial se beneficia (e, na verdade, exige) segurança, escalabilidade e desempenho, esta camada é mostrada como englobando todas as outras. O monitoramento também desempenha um papel vital em qualquer solução empresarial. Sem visualizações eficazes em redes tão complexas, é quase impossível acompanhar tudo. Portanto, essa camada é mostrada como relevante para todas as camadas de blockchain empresarial.

**Projetando soluções de blockchain empresarial**

Um blockchain isolado em uma empresa não é suficiente para resolver problemas de negócios. Além de escolher uma plataforma de blockchain, há outros fatores a considerar ao introduzir um blockchain em uma empresa. No topo da lista desses fatores está a **integração com os sistemas administrativos e legados existentes**.

No diagrama anterior, podemos ver vários blockchains empresariais conectando-se ao Apache Camel usando conectores Ethereum do Apache Camel. Além disso, outros sistemas se conectam ao Apache Camel por meio de seus respectivos conectores. O Apache Camel, via conectores, é responsável por integrar todos esses sistemas. Por exemplo, é totalmente possível usar o Apache Camel para extrair dados de blockchains empresariais e armazená-los em bancos de dados SQL tradicionais da empresa por meio de conectores apropriados.

Sem qualquer estrutura de arquitetura, torna-se bastante desafiador ter uma visão holística de uma empresa e ver como todos os processos de negócios se encaixam. Assim, surge a pergunta sobre se podemos aproveitar estruturas existentes para atender às necessidades arquitetônicas do blockchain empresarial. A resposta é sim. Já existem estruturas estabelecidas e maduras para facilitar o desenvolvimento de arquitetura empresarial.

O objetivo das **estruturas de arquitetura empresarial** é permitir que uma organização execute sua estratégia de negócios de forma eficaz. Ela permite que a organização se enxergue sob diferentes perspectivas, incluindo negócio, informação, processo e tecnologia, e tome decisões comerciais eficazes para alcançar os objetivos.

Agora, vamos explorar as estruturas de arquitetura empresarial mais populares, o **The Open Group Architecture Framework (TOGAF)** e o **Zachman Framework**.

**TOGAF**

TOGAF significa The Open Group Architecture Framework. Ele é desenvolvido pelo Open Group.

É uma estrutura que permite às organizações projetar, planejar e implementar soluções empresariais sistematicamente. Ela possui quatro domínios arquitetônicos:

* **Domínio de arquitetura de negócios**: Este domínio define a estratégia de negócios, estrutura organizacional, processos de negócios e governança da organização.
* **Domínio de arquitetura de dados**: Este domínio descreve as estruturas dos ativos de dados da organização e os recursos relevantes de gerenciamento de dados.
* **Domínio de arquitetura de aplicações**: Este domínio descreve os aplicativos empresariais, os planos de implantação, os relacionamentos e interações entre os aplicativos, juntamente com os relacionamentos que esses aplicativos têm com os processos de negócios dentro da empresa.
* **Domínio de arquitetura tecnológica**: Este domínio define os requisitos da arquitetura técnica para implementação dos aplicativos empresariais. Isso inclui a descrição do hardware, software, middleware e infraestrutura de rede necessários para a implementação das aplicações empresariais.

Agora, reexaminando cada domínio com o blockchain em mente, podemos incluir blockchain em cada camada e tomar decisões como: requisitos de negócios, arquitetura de aplicações, ativos lógicos e físicos de dados e, finalmente, a infraestrutura necessária para implementar a solução de blockchain empresarial.

Após essa introdução básica ao TOGAF, vamos nos aprofundar um pouco para entender o método que o TOGAF usa para desenvolver uma arquitetura de TI.

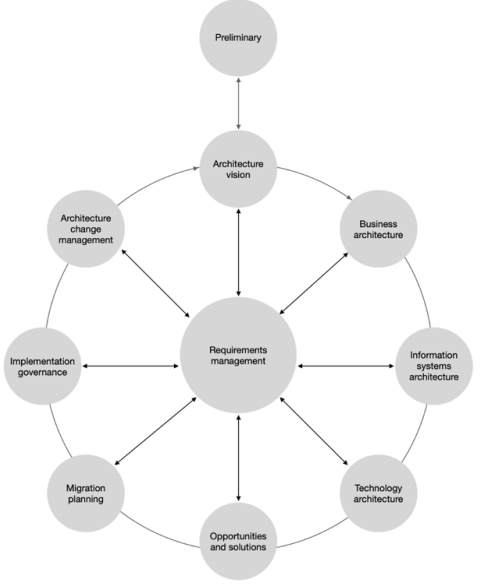
O site oficial do TOGAF pode ser encontrado aqui: <https://www.opengroup.org/togaf>

**Método de desenvolvimento da arquitetura (ADM)**

O TOGAF ADM é um método para desenvolver uma arquitetura de TI que atenda às necessidades comerciais organizacionais. Ele descreve o processo de transição da arquitetura fundacional do TOGAF para uma arquitetura específica da organização. Esse é um processo iterativo com gerenciamento contínuo de requisitos, que resulta no desenvolvimento de uma arquitetura específica para uma organização e que atenda às suas necessidades. Uma vez concluído o desenvolvimento da arquitetura, ela pode ser publicada em toda a organização para criar um entendimento comum.

O ADM é um processo testado e repetível para o desenvolvimento de arquiteturas empresariais. O processo segue as seguintes etapas: estabelecer uma estrutura de arquitetura, desenvolver o conteúdo da arquitetura, governança da arquitetura e implementação das arquiteturas.

O ADM possui nove fases, e cada fase pode ser dividida em múltiplas etapas. O modelo ADM é mostrado aqui:



\*Figura 16.3: Modelo ADM\*

Agora, descreveremos cada fase em mais detalhes:

* **Fase preliminar**: Esta fase define as bases da arquitetura. Define metodologias, princípios de arquitetura, o escopo da arquitetura e suposições sobre a arquitetura. Também define os responsáveis pela entrega da arquitetura. Do ponto de vista do blockchain, podemos definir a estratégia geral de blockchain nesta fase.
* **Visão da arquitetura**: Esta fase valida os princípios de negócio, metas e estratégia de negócios. Define os principais requisitos de negócios, bem como uma descrição de alto nível do valor esperado do trabalho arquitetônico. Esta fase também analisa o impacto da nova arquitetura em outros processos.
* **Arquitetura de negócios**: Esta fase propõe a arquitetura de negócios atual (baseline) e desenvolve uma arquitetura de negócios alvo, juntamente com uma análise de lacunas. Em um cenário de blockchain, podemos definir a arquitetura do estado atual e a arquitetura de estado alvo neste estágio.
* **Arquitetura dos sistemas de informação**: Esta fase define a arquitetura de dados e a arquitetura de aplicações. A arquitetura de dados inclui: construção da arquitetura de dados atual, descrição da arquitetura de dados, construção de modelos de arquitetura de dados, análise de impacto, revisão de modelos de referência e pontos de vista. A arquitetura de aplicações define: a arquitetura de aplicações atual, construção de modelos da arquitetura de aplicações e proposição de aplicações. Ambas as atividades também realizam uma análise de lacunas para validar a arquitetura em desenvolvimento, bem como encontrar quaisquer deficiências entre a arquitetura atual e a arquitetura alvo. Semelhante à arquitetura de negócios, para uma solução de blockchain empresarial, podemos definir a arquitetura atual e a arquitetura alvo.
* **Arquitetura tecnológica**: Esta fase está principalmente preocupada com a revisão das arquiteturas atuais de negócios, dados e aplicações, e a construção de uma descrição da arquitetura tecnológica atual da empresa. Também propõe a arquitetura tecnológica alvo. Nesta fase, podemos propor uma plataforma de blockchain empresarial alvo e a solução alvo.
* **Oportunidades e soluções**: Esta fase realiza avaliações e seleciona várias arquiteturas alvo propostas. Além disso, uma estratégia e plano de implementação são propostos neste estágio. Podemos aplicar isso ao blockchain de forma semelhante, avaliando ou selecionando a arquitetura alvo que abrange soluções de blockchain empresarial. Ao avaliar soluções de blockchain, vários recursos apresentados anteriormente na comparação entre blockchain empresarial e público também podem ser utilizados; por exemplo, confidencialidade, escalabilidade e finalização.
* **Planejamento de migração**: Esta fase cria e finaliza o plano abrangente de implementação para a migração da arquitetura atual para a arquitetura alvo.

O diagrama original pode ser encontrado em: <https://pubs.opengroup.org/architecture/togaf9-doc/arch/chap05.html>

* **Governança da implementação**: Esta fase trata da implementação da arquitetura alvo. Uma estratégia para governar a implantação e migração geral é desenvolvida aqui. Também podemos realizar testes da solução blockchain e elaborar uma estratégia de implantação durante esta fase.
* **Gerenciamento de mudanças da arquitetura**: Esta fase é responsável por criar diretrizes e procedimentos de gerenciamento de mudanças para a arquitetura alvo recém-implementada. Do ponto de vista do blockchain, esta fase pode fornecer procedimentos de gestão de mudanças para a solução blockchain empresarial recém-implementada.

Com isso, concluímos nossa introdução ao TOGAF e exploramos a ideia de que soluções de blockchain empresarial devem ser vistas sob a ótica da arquitetura empresarial.

A ideia fundamental a ser compreendida aqui é que soluções de blockchain empresarial **não** são meramente uma questão de iniciar rapidamente uma rede, criar alguns contratos inteligentes, criar uma interface web e esperar que isso resolva os problemas de negócio. Essa configuração pode ser útil como uma **prova de conceito (PoC)** ou para um caso de uso extremamente simples, mas certamente **não** é uma solução empresarial.

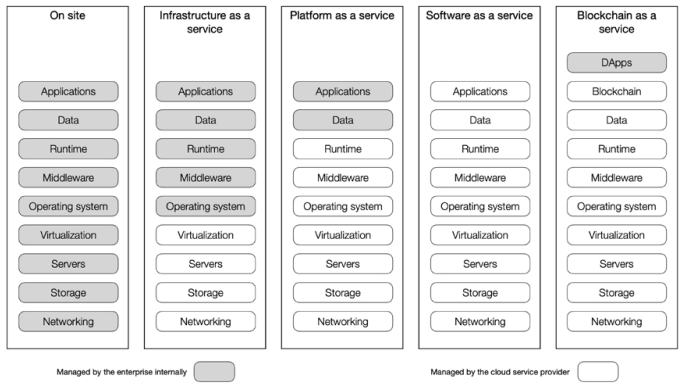
Sugerimos que uma solução de blockchain empresarial deve ser analisada pela perspectiva da **arquitetura empresarial** e considerada como uma **solução empresarial completa**, de nível profissional. Isso para que possamos alcançar efetivamente os objetivos de negócio que se pretende resolver com soluções de blockchain empresarial.

**Blockchain na nuvem**

A computação em nuvem oferece excelentes benefícios às empresas, incluindo eficiência, redução de custos, escalabilidade, alta disponibilidade e segurança. A computação em nuvem fornece serviços de computação como infraestrutura, servidores, bancos de dados e software pela internet. Existem diferentes tipos de serviços em nuvem disponíveis; uma comparação padrão é feita entre **Infraestrutura como Serviço (IaaS)**, **Plataforma como Serviço (PaaS)** e **Software como Serviço (SaaS)**. Surge então a pergunta: **onde o blockchain se encaixa?**

**Blockchain como Serviço**, ou **BaaS**, é uma extensão do SaaS, na qual uma plataforma blockchain é implementada na nuvem para uma organização. A organização gerencia suas aplicações no blockchain, e o restante do gerenciamento de software, gerenciamento da infraestrutura e outros aspectos, como segurança e sistemas operacionais, são gerenciados pelo provedor da nuvem. Isso significa que o software e a infraestrutura do blockchain são fornecidos e mantidos pelo provedor da nuvem. O cliente ou empresa pode se concentrar em suas aplicações de negócio sem se preocupar com outros aspectos da infraestrutura.

A seguir está uma comparação entre diferentes abordagens:



\*Figura 16.04: Soluções em nuvem\*

O BaaS pode ser considerado um tipo de SaaS, onde o software é um blockchain. Nesse caso, assim como no SaaS, todos os serviços são gerenciados externamente. Em outras palavras, os clientes recebem uma rede blockchain totalmente gerenciada sobre a qual podem construir e gerenciar seus próprios dApps. Observe que, no diagrama anterior, sob a coluna "Blockchain como Serviço", **Aplicações** foram substituídas por **Blockchain**, como um diferencial em relação a outros serviços de nuvem. Aqui, o blockchain é o software (aplicação) fornecido e gerenciado pelo provedor de serviços de nuvem. Além disso, observe que os dApps foram adicionados no topo, os quais são gerenciados internamente pela empresa.

Há muitos provedores de BaaS. Alguns deles são listados a seguir:

* **AWS**: <https://aws.amazon.com/blockchain/>
* **Azure**: <https://azure.microsoft.com/en-gb/solutions/blockchain/>
* **Oracle**: <https://www.oracle.com/uk/application-development/cloud-services/blockchain-platform/>
* **IBM**: <https://www.ibm.com/uk-en/cloud/blockchain-platform>

**Blockchains empresariais atualmente disponíveis**

Como esta é uma área muito promissora para pesquisa — e, de fato, o mercado está muito ativo nesse espaço — diversas soluções de blockchain empresarial foram desenvolvidas e disponibilizadas nos últimos anos. Notavelmente, o ano de **2019** foi chamado de “o ano do blockchain empresarial”, à medida que muitas startups e empresas com foco nessa área surgiram.

Nesta seção, não cobriremos todas essas plataformas em detalhes, mas forneceremos uma breve introdução e links para mais detalhes sobre essas cadeias:

* **Quorum** é uma plataforma de blockchain empresarial de código aberto. Ela visa dar suporte às necessidades empresariais e permite que uma empresa atinja seus objetivos de negócio (e desbloqueie valor econômico) aproveitando a tecnologia blockchain. Ela aborda requisitos empresariais cruciais, incluindo **privacidade**, **desempenho** e **permissão empresarial**. Exploraremos o Quorum com mais profundidade posteriormente neste capítulo.
* **Fabric** é um projeto da Hyperledger. Trata-se de um livro-razão distribuído de nível empresarial que permite o desenvolvimento de soluções blockchain com uma arquitetura modular. Possui uma arquitetura permissionada, oferece suporte à modularização e consenso plugável, e permite contratos inteligentes. Discutimos o Hyperledger e outros projetos relacionados, como o Sawtooth, em detalhes no Capítulo 14, *Hyperledger*.

Vamos considerar uma comparação entre as principais plataformas de blockchain empresarial. Cobriremos a maioria dos recursos desejáveis de blockchains empresariais. Isso pode ser usado como referência para uma comparação rápida entre essas plataformas. Esta comparação é baseada nas implementações atuais disponíveis dessas plataformas no momento da redação. No entanto, como esta é uma área que muda rapidamente, alguns recursos podem mudar com o tempo, ou novos recursos podem ser adicionados ou aprimorados:

| **Recurso** | **Quorum** | **Fabric** | **Corda** |
| --- | --- | --- | --- |
| Indústria-alvo | Intersetorial | Intersetorial | Intersetorial |
| Desempenho (TPS aproximado) | 700 (\*) | 560 (∞) | 600 (@) |
| Mecanismo de consenso | Múltiplos plugáveis: Raft, IBFT, PoA | Plugável: Raft | Plugável, baseado em notários |
| Ferramentas | Ferramentas empresariais ricas | SDKs | Ferramentas empresariais ricas |
| Linguagem de contrato inteligente | Solidity | Golang | Kotlin/Java |
| Finalidade | Imediata | Imediata | Imediata |
| Privacidade | Sim (transações privadas restritas) | Sim (transações privadas restritas) | Sim (transações privadas restritas) |
| Controle de acesso | Mecanismo de permissão empresarial | Provedores de serviço de associação/certificado | Serviço de aprovação (Doorman)/KYC |
| Linguagem de implementação | Golang, Java | Golang | Kotlin |
| Associação de nós | Contrato inteligente e software do nó | Via provedor de serviços de associação | Gerenciado por arquivos de configuração |
| Identificação de membros | Chaves públicas/endereços | Baseado em PKI via provedor de associação | Baseado em PKI, com suporte à identidade organizacional |
| Criptografia usada | SECP256K1, AES, Keccak256, Curve25519, etc. | SECP256R1, RSA – PKCS1 | Curve25519 + XSALSA20 + POLY13050, ED25519 |
| Runtime de contrato inteligente | EVM | Containers Docker isolados | JVM determinística |
| Contrato inteligente atualizável | Possível com alguns padrões, não nativo | Permitido via transações de atualização | Permitido com privilégios administrativos |
| Suporte à tokenização | Flexível — herdado dos padrões do Ethereum | Programável | SDK de token Corda |

* “Resultados de TPS para Quorum baseados em: <https://arxiv.org/pdf/1809.03421.pdf>  
  ∞ Resultados de TPS para Hyperledger Fabric baseados em: <https://hyperledger.github.io/caliper-benchmarks/fabric/performance/2.0.0/nodeContract/nodeSDK/submit/empty-contract/>  
  @ Resultados de TPS para Corda baseados em: <https://www.r3.com/corda-enterprise/>”

Essa comparação também pode ser usada para avaliação rápida dessas plataformas e sua adequação para um caso de uso empresarial.

**Desafios do blockchain empresarial**

Embora os blockchains empresariais tenham abordado os principais requisitos empresariais (privacidade, desempenho e governança) até certo ponto, ainda existem mais desafios que precisam ser enfrentados. Há progresso significativo sendo feito no sentido de resolver essas questões. No entanto, ainda há muito trabalho necessário para lidar adequadamente com essas limitações. Alguns desses desafios são listados a seguir:

**Interoperabilidade**

Soluções de blockchain são construídas por diferentes equipes de desenvolvimento com diferentes requisitos-alvo. Como resultado, existem agora muitos tipos diferentes de blockchains, variando desde blockchains públicos de criptomoeda até blockchains específicos de aplicação, desenvolvidos para um único aplicativo de negócio. A troca de dados entre essas cadeias é uma preocupação crucial. Enquanto as redes blockchain continuam a crescer de forma independente, a integração e interoperabilidade entre essas cadeias permanecem uma grande preocupação.

Esse problema também decorre da falta de padronização. Se uma especificação padrão estivesse disponível, todas as cadeias que seguissem esse padrão se tornariam compatíveis automaticamente. Mas e quanto às cadeias que já estão implantadas em produção, incluindo cadeias públicas? Como alcançamos interoperabilidade entre elas? Muitos casos de uso de negócios possuem o requisito de obter dados de uma organização para outra ou de uma rede blockchain para outra. Isso também é verdade no caso de requisitos de troca de dados entre uma blockchain pública e uma rede de consórcio.

**Falta de padronização**

A falta de padronização é uma preocupação comumente destacada nos blockchains empresariais e, em geral, em todo o ecossistema blockchain. Sistemas tradicionais são desenvolvidos, em sua maioria, alinhados com padrões definidos por órgãos de padronização, como os padrões NIST FISMA para segurança da informação.

O Federal Information Security Management Act (FISMA) de 2002 é uma lei federal dos Estados Unidos que exige que todas as agências federais desenvolvam, documentem e implementem um programa de segurança da informação em toda a agência. O FISMA foi alterado em 2014 como a Federal Information Security Modernization Act. Mais informações sobre o FISMA aqui: <https://www.congress.gov/bill/113th-congress/senate-bill/2521>

Padrões também são essenciais para alcançar interoperabilidade. Diversas iniciativas foram tomadas para abordar os desafios mencionados aqui e também para padronizar plataformas de blockchain empresarial. Uma organização líder neste campo é a **EEA**, uma organização de padronização gerida por seus membros que visa desenvolver especificações de blockchain empresarial.

A EEA publica regularmente especificações técnicas que podem ser baixadas no link fornecido aqui:

<https://entethalliance.org/technical-specifications/>  
Site oficial da EEA: <https://entethalliance.org>

**Conformidade**

A conformidade continua sendo uma preocupação significativa nos blockchains empresariais. Os blockchains públicos, por exemplo, são, por definição, sem fronteiras e descentralizados. Os dados armazenados em blockchains públicos não podem ser alterados ou excluídos. Além disso, uma vez que os dados são propagados e armazenados em uma rede de nós global, a soberania dos dados é desafiada. Isso levanta várias preocupações, como:

* Como garantir que os dados não sejam armazenados em nós em países que não sejam compatíveis com certas regulamentações?
* Como aplicar a exclusão de dados sob o GDPR?
* Como manter a conformidade com padrões de segurança como FISMA, NIST, ISO/IEC 27001 e ISO/IEC 27002?

Essas são apenas algumas das muitas preocupações de conformidade que surgem com blockchains empresariais. O desenvolvimento de políticas, procedimentos e mecanismos técnicos adequados para permitir conformidade em soluções blockchain é essencial. Isso inclui suporte a criptografia padrão, anonimato, privacidade, gerenciamento de identidade e integração com provedores de identidade externos e confiáveis.

**Desafios de negócios**

Junto com desafios técnicos, existem diversos desafios de negócios enfrentados por soluções de blockchain empresarial. Eles incluem:

* **Custo**: Custos associados à adoção e integração de soluções blockchain, incluindo custos de desenvolvimento, treinamento, licenciamento e migração.
* **Resistência organizacional à mudança**: Equipes técnicas ou administrativas existentes podem resistir a adotar novas tecnologias que mudam significativamente processos estabelecidos.
* **Falta de compreensão de requisitos**: As organizações muitas vezes não têm clareza suficiente sobre o que esperam alcançar com blockchain, o que resulta em adoção sem justificativa de negócio clara.
* **Adoção limitada do ecossistema**: Um consórcio blockchain requer colaboração entre múltiplas organizações. Se partes importantes não participarem, o valor da rede pode ser comprometido.

**VMware Blockchain**

VMBC é uma plataforma blockchain de nível empresarial que fornece recursos adequados para casos de uso empresariais. Ela introduziu muitas inovações que resultam em melhor desempenho, flexibilidade, segurança e escalabilidade de aplicações descentralizadas (dApps) em blockchain.

<https://entethalliance.org/technical-specifications/>.

O site oficial da EEA é <https://entethalliance.org>.

<https://entethalliance.org/technical-specifications/>  
<https://entethalliance.org>

**Componentes**  
Os componentes centrais do VMware Blockchain são:

* **Framework modular** – um framework modular tal que vários motores de execução de contratos inteligentes (como DAMLe ou EVM) podem ser usados. DAMLe, ou o motor de execução DAM, basicamente converte ações da camada de aplicação em eventos na camada de razão distribuída.
* **Rede de réplicas** – um conjunto de nós de réplica que apresenta uma única e consistente máquina de estados aos nós cliente.
* **Nós de réplica:**
  + Os nós de réplica usam um mecanismo de consenso para implementar o protocolo de replicação de máquina de estados SBFT, que forma uma rede de réplicas composta por nós de réplica.
  + Cada nó de réplica possui um identificador único, armazena uma máquina de estados replicada e utiliza um banco de dados local de chave-valor (RocksDB). Uma réplica é selecionada como líder durante um ciclo de consenso. Além disso, alguns nós de réplica possuem papéis de coletor. Existem dois tipos de coletores: C-coletor e E-coletor. O C-coletor coleta mensagens de confirmação e uma assinatura combinada é enviada de volta às réplicas para confirmar o commit. O E-coletor coleta todas as mensagens de execução, e uma assinatura combinada é enviada de volta às réplicas e aos clientes como um certificado de prova da execução de suas solicitações.
* **Grupo de nós cliente:**
  + **Nós cliente regulares** – fornecem uma interface entre aplicações e a rede de réplicas. Os nós cliente incorporam o cliente BFT e interfaces de aplicação como o servidor de API do livro-razão DAML.
  + **Nó cliente de cópia completa** – recebe todas as atualizações, em contraste com o nó cliente regular, que recebe uma visão filtrada dos dados que tem permissão para acessar.
  + **Nó cliente duplicado** – usado para fornecer alta disponibilidade e balanceamento de carga.
* **VMWare Blockchain Orchestrator** – um appliance virtual autônomo que implanta nós do VMware Blockchain on-premises.
* Também há ferramentas disponíveis para implantação de nós, monitoramento e gerenciamento de frota.
* Um armazenamento de objetos é usado para arquivamento dos dados.

Assim como outras blockchains, a VMware depende de um protocolo de consenso para garantir o acordo entre seus participantes, o qual discutiremos a seguir.

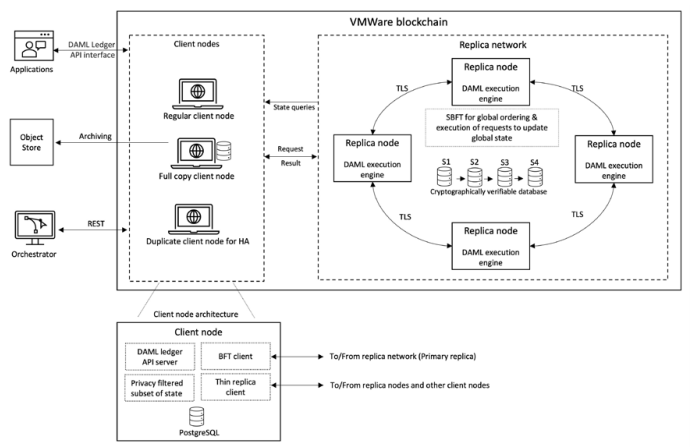
**Protocolo de consenso**  
O consenso no VMware Blockchain é uma variante do PBFT, chamada SBFT. Ele é mais escalável e descentralizado em comparação com variantes anteriores do PBFT. Garante segurança, vivacidade e linearidade. Linearidade, em termos simples, significa mensagens de tamanho constante, o que resulta em um custo constante de operações na cadeia. O mecanismo de eleição de líder no SBFT garante que um grupo diferente de coletores seja escolhido para cada bloco, o que distribui a carga da liderança primária e dos coletores entre todas as réplicas. O SBFT relatou cerca de 70 transações por segundo em uma rede geograficamente dispersa e com alta carga.

**Enterprise Blockchain**  
Há três melhorias principais, incluindo:

* Comunicação reduzida devido ao uso de coletores, o que permite comunicação linear e evita comunicação quadrática todos-para-todos. Além disso, uma única confirmação é enviada aos clientes.
* Implementação de assinaturas BLS mais rápidas e de tamanho menor (33 bytes), o que reduz o comprimento da prova. Assinaturas limiares também reduzem a comunicação do cliente de linear para constante.
* Caminho otimista de consenso em uma rodada, o qual é importante para reduzir a latência e resulta em um acordo mais rápido. Também há um caminho otimista mais rápido para execuções em casos normais.

Com todos os componentes introduzidos até agora, podemos agora descrever a arquitetura do VMware Blockchain.

**Arquitetura**  
Podemos visualizar a arquitetura de alto nível do VMware Blockchain no diagrama mostrado abaixo:

O diagrama mostra uma rede de réplicas composta por quatro nós e um banco de dados criptograficamente verificável. O estado final é alcançado após a execução do algoritmo de consenso SBFT. Os nós cliente se comunicam com a rede de réplicas por meio de canais de comunicação seguros via TLS. Os nós cliente (mostrados no lado esquerdo do diagrama) podem fazer consultas de estado e solicitar informações do nó de réplica primário na rede de réplicas.

**Um nó cliente consiste** em um servidor de API do livro-razão DAML e um subconjunto filtrado por privacidade do estado, já que um nó só vê o que está autorizado a ver. O nó também consiste em um cliente BFT que se comunica com a réplica primária na rede de réplicas, e um cliente de réplica leve, que executa o protocolo de réplica leve (*thin replica protocol*).

O nó cliente aceita solicitações vindas das aplicações e as envia para as réplicas, recebe resultados da rede de réplicas e os envia de volta para as aplicações. O *dApp* (aplicação descentralizada) envia comandos para o componente de servidor de API do livro-razão DAML do nó cliente. Os componentes da API DAML encaminham o comando para um conjunto de clientes BFT, que envia as solicitações para a rede de réplicas para execução e aguarda a coleta dos resultados dos nós de réplica. O cliente BFT coleta e assegura a validade da resposta (resultados) que recebe da rede de réplicas e encaminha os resultados de volta à API do livro-razão DAML. A API do livro-razão DAML recebe os resultados da execução usando a notificação do PostgreSQL. Usando o protocolo de réplica leve, um nó de réplica envia os resultados para outros nós cliente que estão inscritos para receber essas atualizações. O nó cliente, usando o protocolo de réplica leve, recebe a atualização e a escreve em seu banco de dados PostgreSQL local. Uma notificação indicando a atualização no banco de dados local é enviada para a API do livro-razão DAML. O banco de dados PostgreSQL armazena uma *view* materializada da rede de réplicas, que são os dados que o nó cliente específico está autorizado a visualizar em um formato processado. Cada nó de réplica contém uma máquina de estados replicada (*replicated state machine*, RSM), um armazenamento local de chave-valor autenticado (RocksDB), e o DAMLe – o mecanismo de execução DAML. O banco de dados é criptograficamente verificável. Ele suporta pontos de verificação para arquivamento. A ordenação das requisições é obtida via consenso.

Outros componentes do blockchain incluem aplicações que se comunicam com os nós cliente via interface da API do livro-razão DAML. Um armazenamento de objetos também existe, que é usado para arquivar dados via um nó cliente de cópia completa. Finalmente, um appliance orquestrador é usado para criar, implantar e gerenciar o VMware Blockchain via uma interface REST.

O VMBC oferece suporte a DAML e EVM. Execução paralela é suportada pelo mecanismo DAML.

DAML é uma linguagem funcional de contratos inteligentes com tipagem estática – uma DSL influenciada por Haskell. É uma linguagem concisa, semelhante ao inglês, fácil de entender e escrever, orientada a negócios. Ela foi projetada para construir uma aplicação composta multiusuário. Também é chamada de “Haskell Empresarial”. É basicamente Haskell, com primitivas adicionais para contratos inteligentes, regras de autorização e privacidade. DAML visa resolver problemas de portabilidade, interoperabilidade e privacidade no blockchain. Ela se concentra na privacidade dos dados e na autorização das aplicações distribuídas. Nesta linguagem, as partes são primitivas de primeira classe. É construída com privacidade em mente e permite rastreamento e autorização em cada etapa do fluxo de trabalho – ou seja, cada contrato inteligente tem sua própria privacidade definida. Ao abstrair a privacidade dos dados e a autorização, DAML permite que os desenvolvedores se concentrem na lógica do fluxo de trabalho, em vez de se preocupar com primitivas criptográficas concretas. Também é independente da camada de armazenamento, portátil e interoperável, o que significa que o código DAML escrito para uma plataforma funcionará em outras plataformas também, sem qualquer alteração. Por exemplo, o mesmo código DAML funcionará tanto para Hyperledger quanto para Ethereum, e até mesmo para um banco de dados SQL, sem nenhuma modificação.

Mais informações sobre este blockchain estão disponíveis aqui:  
<https://www.vmware.com/uk/products/blockchain.html>

<https://www.vmware.com/uk/products/blockchain.html>  
<https://www.vmware.com/uk/products/blockchain.html>

**Enterprise Blockchain**  
**VMware Blockchain para Ethereum**  
O Ethereum público é inadequado para casos de uso empresariais devido à escalabilidade, governança estrita, suporte operacional de nível empresarial e exigências de privacidade. No entanto, é a plataforma mais usada para desenvolvimento de contratos inteligentes devido ao seu ecossistema de desenvolvimento maduro. Ele pode se tornar adequado para casos de uso empresariais se conseguirmos lidar com as limitações da cadeia pública do Ethereum. O VMware Blockchain para Ethereum resolve essas questões e fornece uma plataforma que permite desenvolvimento fácil, operações eficientes e redes blockchain Ethereum escaláveis. O suporte ao Ethereum pode ser fornecido além do suporte ao DAML já disponível no VMware Blockchain. O VMware Blockchain para Ethereum é uma plataforma de alto desempenho e alta vazão com capacidades de privacidade programável baseadas em ZKP, operações de nível empresarial e recursos de governança. Os desenvolvedores podem usar ferramentas familiares como MetaMask, Remix, Hardhat e Truffle para desenvolver nesta plataforma. Além disso, ela oferece suporte a APIs padrão do Ethereum, proporcionando uma experiência de desenvolvedor fluida.

Em resumo, podemos dizer que, com todas as inovações, melhorias e escolhas inteligentes de recursos amigáveis à empresa, o VMware Blockchain é uma escolha valiosa para casos de uso empresariais.

**Quorum**  
Quorum é uma plataforma de blockchain empresarial de código aberto. Trata-se de um fork leve do Ethereum. O Quorum não apenas se beneficia da inovação e pesquisa sendo realizadas no projeto Ethereum público (Geth), como também introduz muitos recursos excelentes voltados para empresas. Esses recursos empresariais focam principalmente em fornecer privacidade, desempenho e controle de acesso (permissionamento) de nível empresarial.

Primeiro, vejamos a arquitetura do Quorum.

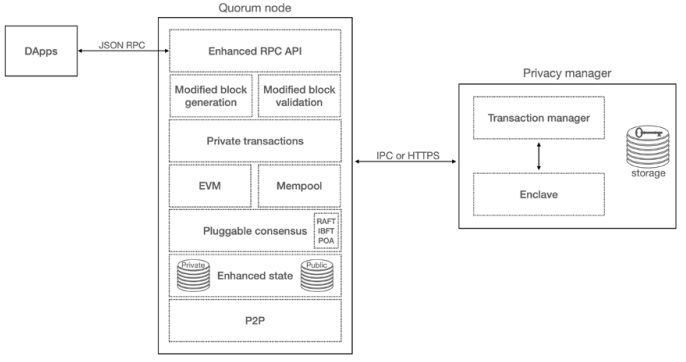
**Arquitetura**  
O Quorum aborda três questões fundamentais das blockchains públicas, o que o torna uma excelente escolha para casos de uso empresariais:

* Privacidade
* Desempenho
* Governança empresarial

Mais informações sobre DAML estão disponíveis aqui:  
<https://www.digitalasset.com/developers>.  
Observe que DAML é desenvolvido pela Digital Asset Holdings.

<https://www.digitalasset.com/developers>

Em alto nível, a arquitetura do Quorum consiste em nós e seus respectivos gerenciadores de privacidade. A arquitetura do Quorum pode ser visualizada usando o seguinte diagrama:



**Nós**  
O nó Quorum é uma versão modificada e aprimorada do Geth público que oferece suporte a transações privadas. Os nós do Quorum se comunicam via HTTPS com gerenciadores de privacidade, os quais são responsáveis por fornecer privacidade ao armazenar o conteúdo (payload) das transações em formato criptografado em seu armazenamento local. Esses nós do Quorum mantêm os estados público e privado separadamente.

Várias alterações foram feitas no cliente Quorum para torná-lo adequado a casos de uso empresariais. A seguir, apresentamos uma visão geral de cada uma dessas alterações:

* **Camada P2P aprimorada:** Essa camada foi modificada para permitir conexões apenas entre nós autorizados.
* **Estado aprimorado (privado e público):** Além do trie de estado público, há um *Merkle Patricia trie* adicional para o estado privado. No Ethereum, um *Merkle Patricia trie* modificado (MPT) é uma estrutura de dados usada para fornecer um armazenamento de chave-valor autenticado criptograficamente. Mais detalhes técnicos sobre isso estão disponíveis aqui:  
  <https://eth.wiki/en/fundamentals/patricia-tree>.
* **Consenso plugável:** O Quorum oferece suporte a vários algoritmos de consenso, como IBFT, Proof of Authority (PoA) e Raft. Sendo uma blockchain empresarial, o Quorum não precisa de um consenso baseado em Prova de Trabalho (PoW).
* **Sem taxas de transação:** O custo do *gas* foi definido como zero, pois não há necessidade de taxas de transação em redes de consórcio.

<https://eth.wiki/en/fundamentals/patricia-tree>  
<https://eth.wiki/en/fundamentals/patricia-tree>

**Enterprise Blockchain**

* **Transações privadas:** Transações privadas são identificadas usando 37 ou 38 como valor de V na transação. O mecanismo de criação de transações foi modificado para permitir a substituição dos dados da transação (*input*) pelo hash do conteúdo criptografado.
* Lembre-se de que, no Ethereum padrão, como aprendemos no Capítulo 9, *Arquitetura do Ethereum*, o valor V de uma transação pode ser 27 ou 28, mas no Quorum, ele foi alterado para 37 ou 38 para indicar transações privadas. Transações públicas ainda são identificáveis por um valor V igual a 27 ou 28, enquanto transações privadas são identificáveis com um valor V de 37 ou 38.
* **Mecanismo de geração de blocos modificado:** A lógica para gerar blocos foi modificada com uma nova verificação para a raiz do estado público global, em vez da raiz do estado global.
* **Mecanismo de validação de blocos modificado:** A lógica de validação de blocos foi modificada para substituir a verificação da raiz do estado global pela verificação da raiz do estado público global — também conhecida como *global public state root*.
* **API RPC aprimorada:** O Quorum oferece APIs RPC adicionais que nos ajudam a interagir com os recursos empresariais aprimorados do Quorum, como o sistema de permissionamento e os mecanismos de consenso.

Agora, discutiremos o **gerenciador de privacidade**, que consiste em dois componentes: um **gerenciador de transações** e um **enclave**.

**Gerenciador de privacidade**  
O gerenciador de privacidade é um mecanismo fora da cadeia (*off-chain*) que fornece confidencialidade às transações. Ele está emparelhado em uma base de um-para-um com um nó Quorum. Permite que os nós Quorum compartilhem o conteúdo da transação de forma segura entre participantes autorizados.

O gerenciador de transações é responsável por:

* Armazenar conteúdos criptografados das transações
* Gerenciar o acesso aos conteúdos criptografados
* Propagar conteúdos criptografados para outros gerenciadores de transações na rede
* Descobrir outros gerenciadores de transações na rede

O Quorum desenvolveu dois gerenciadores de transações. Inicialmente, foi disponibilizada uma implementação em Haskell chamada **Constellation**. No entanto, ela não é mais desenvolvida ativamente, sendo substituída por um gerenciador de privacidade mais completo, baseado em Java, chamado **Tessera**. Tessera é desenvolvido em Java. É usado para armazenamento, criptografia, descriptografia e propagação de dados de transações privadas.

Um **enclave** é um elemento isolado e independente que fornece serviços criptográficos para criptografar e descriptografar o conteúdo das transações. Ele só pode estar associado a um único gerenciador de transações.

Para alcançar todos os recursos de privacidade desejados, vários protocolos e primitivas criptográficas foram usados na plataforma Quorum. Forneceremos a seguir uma visão geral rápida sobre eles.

**Criptografia**  
O Quorum herda sua pilha criptográfica do Ethereum público. Ela inclui assinaturas padrão ECDSA, criptografia AES em modo CTR para carteiras e DEVP2P, e funções de hash Keccak256. O gerenciador de privacidade utiliza Curve25519, Elliptic-curve Diffie-Hellman (ECDH), poly1305 e operações criptográficas Xsalsa20 para fornecer garantias de confidencialidade exigidas por transações privadas no Quorum.

Agora, exploraremos como cada uma das principais áreas — privacidade (confidencialidade), controle de membros de nível empresarial (controle de acesso e mecanismo de permissionamento), e desempenho — são implementadas no Quorum.

**Privacidade**  
O Quorum oferece suporte a transações públicas e privadas. Para transações privadas, ele usa um mecanismo chamado **gerenciador de transações privadas**, que é um componente fora da cadeia para facilitar a confidencialidade das transações. Descreveremos agora como funcionam as transações privadas no Quorum.

Suponha que existam três partes: A, B e C. As partes A e B têm acesso a uma transação, enquanto a parte C não. Agora, exploraremos como a transação privada é gerada, como flui entre as partes e é propagada na rede enquanto mantém a confidencialidade. Chamaremos essa transação de transação AB:

1. Para iniciar a transação privada AB, a parte A cria uma transação e a assina antes de enviá-la ao seu nó Quorum, o nó A. A transação é composta por um conteúdo de transação e a chave pública do destinatário pretendido. Essa lista de chaves públicas dos destinatários é mantida na lista **PrivateFor**. Ela pode conter uma única chave pública ou múltiplas, dependendo do caso.
2. Há dois mecanismos de assinatura no Quorum. Para transações públicas, é usado um mecanismo baseado no EIP55, e para transações privadas, um mecanismo de assinatura baseado no Ethereum Homestead é utilizado. Transações no Quorum também podem ser assinadas de forma independente, sem usar o mecanismo de assinatura do próprio Quorum.
3. O nó Quorum A envia a transação para o gerenciador de transações A para processamento.
4. O gerenciador de transações A faz uma solicitação de criptografia ao seu enclave, para criptografar o conteúdo da transação.
5. O enclave da parte A criptografa o conteúdo da transação e envia para o gerenciador de transações A.
6. O gerenciador de transações A armazena o conteúdo da transação e o envia para outros gerenciadores de transações, neste caso, o da parte B.
7. Um hash da transação é retornado ao nó Quorum A pelo gerenciador de transações A, que substitui o conteúdo original da transação por esse hash e altera o valor V da transação para 37 ou 38, para indicar que se trata de uma transação privada.
8. A transação é propagada para outros nós por meio do protocolo P2P normal do Ethereum.
9. O bloco que contém a transação AB é finalizado e propagado pela rede entre todos os nós (A, B e C).
10. Quando o bloco contendo a transação é recebido pelos nós do Quorum, eles reconhecem a transação como privada porque o valor V da transação é 37 ou 38.

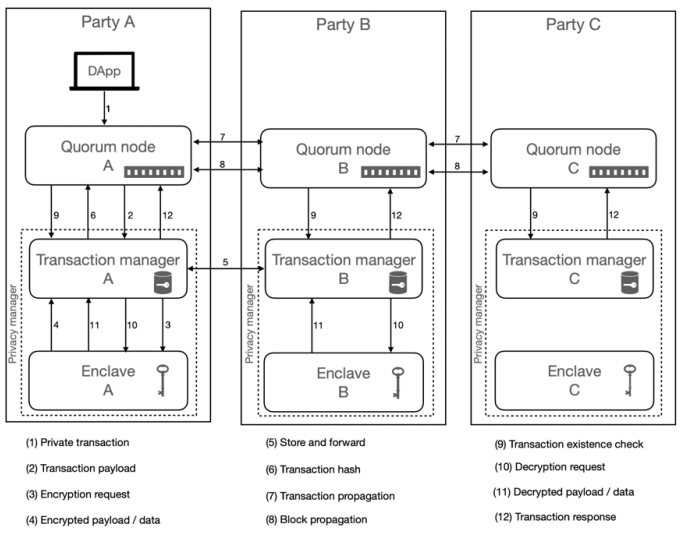
**11.** Se a transação for identificada como privada, os nós do Quorum consultarão seus respectivos gerenciadores de transações para determinar se participam ou não da transação. Isso é feito verificando se há uma entrada disponível no banco de dados com o hash da transação. Neste caso, as partes A e B terão o hash da transação armazenado em seus respectivos gerenciadores, enquanto a parte C não terá.

**12.** Os gerenciadores de transações das partes A e B fazem uma solicitação de descriptografia do conteúdo da transação aos seus enclaves correspondentes.

**13.** Os enclaves das partes A e B descriptografam as transações privadas.

**14.** Quando os gerenciadores de transações recebem de volta o conteúdo descriptografado de seus enclaves, eles enviam esses dados aos seus respectivos nós do Quorum. Em nosso exemplo, os gerenciadores de transações das partes A e B enviarão o conteúdo da transação para seus respectivos nós Quorum, A e B. Os nós do Quorum executarão a transação (contrato) via EVM e atualizarão seus bancos de dados de estado privado de acordo. O gerenciador de transações da parte C retornará uma mensagem ao seu nó Quorum indicando que não participa (ou não é destinatário) da transação. Isso significa que o nó Quorum da parte C simplesmente ignorará a transação.

Esse processo pode ser visualizado com o seguinte diagrama:



**Vamos agora expandir o passo 4 do processo de obtenção de privacidade, a criptografia no enclave**, pois ela consiste em várias subetapas realizadas dentro do enclave.

**Criptografia no enclave**

Quando um enclave recebe uma transação, ele executa várias etapas para criptografar a transação. Todas essas etapas são mostradas a seguir e são realizadas pelo enclave como parte do processo de criptografia do conteúdo da transação:

1. Gera uma chave simétrica para a transação.
2. Gera dois nonces aleatórios.
3. Criptografa o conteúdo da transação e um dos nonces com a chave simétrica gerada no primeiro passo.
4. Criptografa a chave da transação para cada destinatário. Para isso, são realizados os seguintes passos:
   * Gera uma chave simétrica compartilhada utilizando ECDH. O ECDH usará a chave privada do remetente e a chave pública do receptor.
   * Criptografa a chave simétrica da transação para cada receptor separadamente, usando a chave simétrica compartilhada recém-gerada e o segundo nonce.
   * Esse passo será repetido para cada destinatário.
5. Finalmente, o gerenciador de transações recebe o conteúdo criptografado da transação, todas as chaves simétricas criptografadas para as transações, ambos os nonces e as chaves públicas dos remetentes e receptores.

**Agora, vejamos como o gerenciador de transações armazena e propaga transações para outro gerenciador de transações. Expandiremos agora o passo 5 do processo de obtenção de privacidade mencionado anteriormente.**

**Propagação da transação aos gerenciadores de transações**

Um gerenciador de transações executa os seguintes passos após receber uma resposta criptografada do enclave:

1. Gera o hash SHA3-512 do conteúdo criptografado recebido do enclave.
2. Armazena o conteúdo criptografado, o hash e a chave simétrica criptografada da transação, ambos os nonces e as chaves públicas do remetente e do receptor.
3. Envia, usando HTTPS, o conteúdo criptografado, o hash e a chave simétrica criptografada da transação usando a chave pública do gerenciador de transações B, para o gerenciador B.
4. O gerenciador de transações A aguarda uma mensagem de confirmação do gerenciador de transações B. Se a confirmação não for recebida, a transação não será propagada para a rede.

**Vamos agora descobrir como funciona o processo de descriptografia nos enclaves. Isso corresponde ao passo 11 do processo de obtenção de privacidade.**

**Descriptografia no enclave**

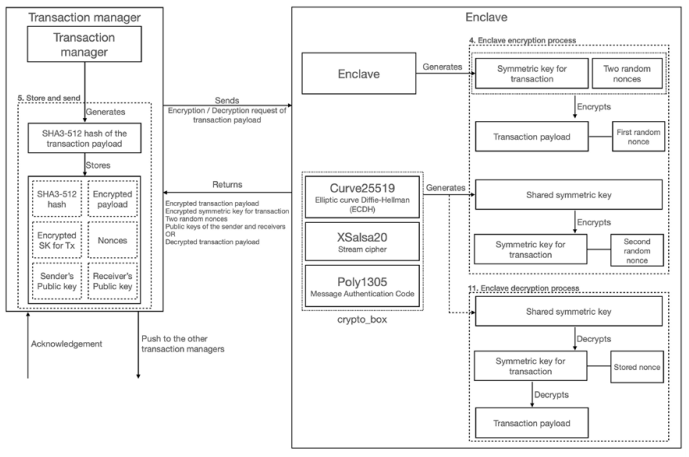
O processo de descriptografia do conteúdo da transação começa quando uma solicitação de descriptografia é feita a um enclave. O enclave executa os seguintes passos para descriptografar um conteúdo:

1. Deriva a chave simétrica compartilhada. O processo de derivação funciona da seguinte forma:
   * A parte A, sendo o remetente da transação, deriva a chave simétrica compartilhada usando sua chave privada e a chave pública do receptor.
   * A parte B, sendo o receptor da transação, deriva a chave simétrica compartilhada usando sua chave privada e a chave pública do remetente.
2. Descriptografa a chave simétrica da transação com a chave simétrica compartilhada, junto ao conteúdo criptografado e o nonce, buscados no banco de dados.
3. Descriptografa os dados da transação com a chave simétrica da transação e os dados criptografados e nonce buscados no banco de dados.
4. Finalmente, os dados descriptografados da transação privada são enviados ao gerenciador de transações.

O enclave, por ser separado do gerenciador de transações, resolve a separação de responsabilidades e permite paralelização para melhorar o desempenho. Separação de responsabilidades é uma forma de abstração que permite segregar diferentes componentes do software em unidades separadas e distintas, de modo que cada unidade trate de uma preocupação específica. Isso simplifica o design e ajuda a modularizar o software.

**O processo geral**

(criptografia no enclave, armazenamento e propagação da transação para outros nós pelo gerenciador de transações, e descriptografia no enclave), correspondente aos Passos 4, 5 e 11 na Figura 16.7, pode ser visualizado no seguinte diagrama:

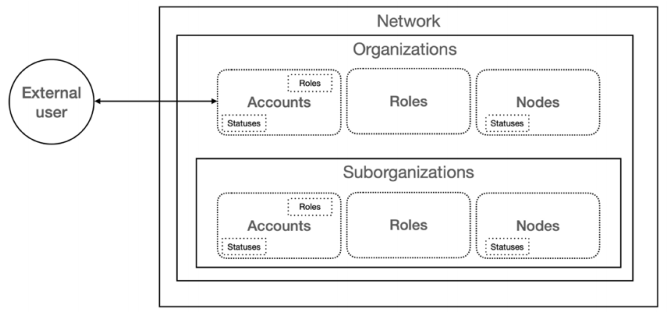
Cobrimos bastante teoria aqui, mas com isso, concluímos agora nossa introdução às transações privadas no Quorum. Em seguida, exploraremos como o controle de acesso funciona no Quorum.

**Controle de acesso com permissionamento**

Como uma plataforma de blockchain empresarial, o Quorum inclui um mecanismo de controle de acesso de nível empresarial que gerencia a participação na rede por membros do consórcio. Ele é implementado em contratos inteligentes escritos na linguagem Solidity. Suporta muitos recursos esperados de um mecanismo de permissionamento de nível empresarial. Inclui funcionalidades para suporte à gestão de nós, permissionamento ao nível de contas e suporte para decisões por votação em ações de permissionamento.

**A arquitetura geral do mecanismo de permissionamento do Quorum**

pode ser visualizada no seguinte diagrama:



*Figure 16.9: Quorum permissioning mechanism*

Esse mecanismo de permissionamento é inspirado por um padrão amplamente aceito na indústria chamado **Controle de Acesso Baseado em Papéis** (RBAC). Ele é padronizado pelo **American National Standards Institute (ANSI)** e é utilizado na maioria dos softwares de nível empresarial, como sistemas operacionais e sistemas em nuvem.

O padrão RBAC está disponível aqui:  
<https://standards.incits.org/apps/group_public/project/details.php?project_id=1658>.

**Os recursos que o mecanismo de permissionamento do Quorum oferece incluem:**

* Gerenciamento de papéis
* Atribuição de papéis às contas (sujeitos)
* Gerenciamento de permissões
* Gerenciamento de nós

Um **papel** pode ser definido como uma função de trabalho nomeada dentro de uma organização.

Em um nível fundamental, o modelo de gerenciamento de permissões oferece controle granular sobre:

* Transferência de fundos
* Criação de contratos inteligentes
* Execução de contratos inteligentes

Ele controla quais contas podem transferir fundos, criar contratos inteligentes ou executar contratos inteligentes. Essas permissões são usadas para criar papéis, que então são atribuídos a contas para fazer cumprir as permissões.

<https://standards.incits.org/apps/group_public/project/details.php?project_id=1658>

**As permissões são divididas em duas categorias amplas:**

1. **Permissões de conta** – controlam quais funções uma conta pode executar
2. **Permissões de nó** – controlam a participação dos nós na rede

As contas podem ser atribuídas com diferentes papéis e status, enquanto os nós podem ter diferentes status atribuídos.  
Os **status de contas** incluem:

* Ativo
* Inativo
* Suspenso
* Em lista negra (*blacklisted*)

Os **nós** podem estar nos estados:

* Aprovado
* Desativado
* Aguardando aprovação
* Em lista negra

Os tipos de acesso possíveis incluem:

* Somente leitura
* Transação
* Implantação de contrato
* Acesso total

**Desempenho**

Desempenho é um assunto amplo e pode significar coisas diferentes em diferentes cenários. Na maioria das vezes, é simplesmente uma medida de quantas transações um sistema pode executar em um dado intervalo de tempo, geralmente medido em transações por segundo (TPS). Também pode significar a vazão total do sistema, a qualidade do serviço e a capacidade de escalar. Do ponto de vista empresarial, pode significar desempenho do negócio. No entanto, aqui lidaremos apenas com o cenário de TPS e apresentaremos alguns estudos realizados para avaliar o desempenho do blockchain Quorum.

A ideia central por trás da melhoria de desempenho no blockchain Quorum é o uso de algoritmos de consenso determinísticos e rápidos. O Quorum oferece suporte a vários mecanismos de consenso adequados para redes de consórcio. Comparados a outros mecanismos de consenso de blockchains públicas, os algoritmos usados no Quorum oferecem melhor desempenho.

Várias avaliações foram feitas e publicadas em relação ao desempenho do Quorum. Esses estudos reportaram que o desempenho do Quorum pode chegar até 2.500 TPS.

**Agora, apresentaremos os mecanismos de consenso suportados pelo Quorum.**

**Consenso plugável**

O Quorum oferece suporte a vários algoritmos de consenso que podem ser utilizados conforme os requisitos do caso de uso.  
Por exemplo, se a necessidade for apenas tolerância a falhas por falha de processo (*crash-tolerance*), os usuários podem escolher o **Raft**. Se for necessário mais segurança e tolerância a falhas bizantinas, então os usuários podem escolher o **IBFT**.  
Listamos abaixo os mecanismos de consenso disponíveis no Quorum:

* **Raft:** Um algoritmo de consenso tolerante a falhas por falha de processo (crash fault-tolerant)
* **IBFT:** Um algoritmo BFT inspirado no PBFT
* **Clique:** Prova de Autoridade (PoA), herdado do Ethereum público

Esses estudos estão disponíveis aqui:

* <https://scandiweb.com/blog/jpmorgan-s-quorum-blockchain-performance-testing>
* <https://arxiv.org/pdf/1809.03421.pdf>

**Enterprise Blockchain**

Discutimos todos esses algoritmos em detalhes no **Capítulo 5 – Algoritmos de Consenso**. Você pode revisar esses tópicos em detalhes lá.

Agora que passamos por diversos recursos e construímos um entendimento teórico sobre as diversas funcionalidades do blockchain empresarial Quorum, vejamos como configurar uma rede Quorum.

**Configurando uma rede Quorum com IBFT**

Nesta seção, configuraremos uma rede Quorum com quatro nós, demonstrando como uma rede IBFT pode ser montada e como realizar transações privadas no Quorum. Esses passos podem ser executados manualmente, ou podemos usar uma ferramenta chamada **Quorum Wizard** para criar rapidamente uma rede Quorum.

Para detalhes sobre a configuração manual, consulte as instruções detalhadas no site oficial do Quorum aqui:  
<https://docs.goquorum.consensys.net/tutorials/quorum-dev-quickstart>.  
Usaremos o Quorum Wizard para configurar nossa rede.

**Instalando e executando o Quorum Wizard**

**Quorum Wizard** é uma ferramenta de linha de comando escrita em JavaScript que permite aos usuários criar rapidamente uma rede local de nós Quorum. Ela é executada como um módulo npm e, portanto, requer Node.js e npm para funcionar.

Para verificar a versão atual instalada, execute o seguinte comando:

$ npm -v

Esse comando produzirá a seguinte saída, indicando a versão instalada do npm (Node Package Manager):

6.14.4

Este comando mostrará a versão atual do Node instalada:

$ node –v

Este comando exibirá a versão do Node instalada:

v12.18.0

Se o número da versão esperada for exibido após a execução desses dois comandos, então já podemos instalar o Quorum Wizard.

O Node.js pode ser instalado a partir deste endereço:  
<https://nodejs.org/en/>

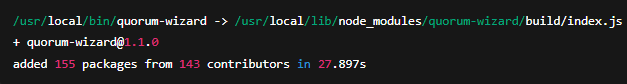
<https://docs.goquorum.consensys.net/tutorials/quorum-dev-quickstart>  
<https://nodejs.org/en/>

**Instalando o Quorum Wizard**

O **Quorum Wizard** pode ser instalado usando o comando npm install. É uma excelente ferramenta que reduz o tempo de instalação de horas para minutos:

$ npm install -g quorum-wizard

Esse comando produzirá uma saída semelhante à mostrada aqui:



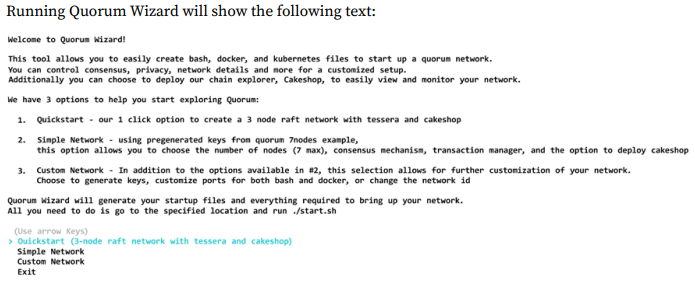
**Executando o Quorum Wizard**

Instalado o Quorum Wizard, podemos executá-lo com o seguinte procedimento.

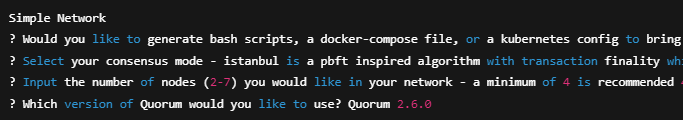
Vamos criar uma rede IBFT com 4 nós neste exemplo. Basta executar o Quorum Wizard e seguir os passos conforme solicitado:

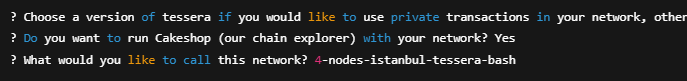
$ quorum-wizard

Ao executar o Quorum Wizard, será exibido o seguinte texto:

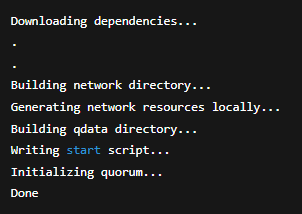


Selecione **Simple Network** e selecione todas as opções usando as teclas de seta para cima e para baixo. Pressione Enter após cada escolha, o que nos levará para a próxima pergunta do assistente (Wizard):

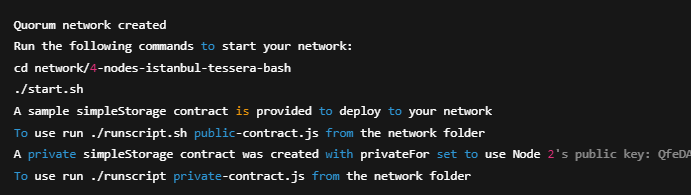




Uma vez que todas as opções tenham sido selecionadas, conforme mostrado no trecho acima, a ferramenta fará o download e a instalação das dependências. Esse processo produzirá uma saída semelhante à mostrada abaixo. Por brevidade, a saída completa não está incluída:



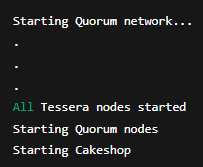
Finalmente, será exibida a seguinte saída, indicando o sucesso da operação:



Agora, vá para o diretório 4-nodes-istanbul-tessera-bash, como mostrado abaixo, e inicie a rede:



Isso produzirá uma saída semelhante à mostrada aqui. Parte da saída foi truncada por brevidade:



E é isso! Agora temos uma rede Quorum com quatro nós executando o protocolo IBFT.

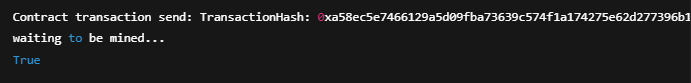
**Executando uma transação privada**

Nesta seção, exploraremos como transações privadas podem ser criadas e executadas em uma rede Quorum:

1. De dentro do diretório 4-nodes-istanbul-tessera-bash, execute o seguinte comando:



1. Uma vez que o comando acima for executado, ele produzirá uma saída semelhante à mostrada abaixo, indicando que a transação foi enviada e que um hash de transação foi gerado:



Aqui, basicamente executamos uma transação privada e implantamos um contrato inteligente privado que é visível apenas para o Nó 1 e o Nó 2. Os Nós 3 e 4 não têm acesso a essa transação e não devem ser capazes de visualizar o conteúdo do código nem os valores (estado) do contrato. Demonstramos neste exemplo que isso de fato ocorre.

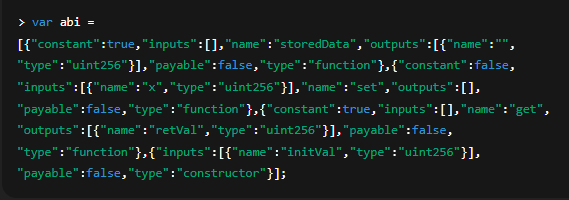
**Anexando o Geth aos nós**

Em seguida, vamos anexar o Geth a cada nó, conforme mostrado nos comandos nas próximas seções, e executar os comandos no console conforme os exemplos para cada nó.

**Nó 1:**  
Para interagir com o Geth via IPC, insira o seguinte comando no terminal do sistema operacional. Isso abrirá o console JavaScript do Geth, onde os usuários podem interagir com o blockchain usando vários métodos expostos pelo Geth:



Com o console do Geth aberto, insira o seguinte comando, que declara uma variável chamada abi:



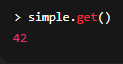
Agora, crie uma nova instância de contrato usando o seguinte comando. Podemos usar esse objeto para acessar todos os métodos e eventos do contrato:



Em seguida, usamos o objeto de instância criado anteriormente para obter uma abstração completa de nosso contrato com o seguinte comando:



Agora podemos usar o objeto simple para acessar todos os métodos em nosso contrato inteligente. No exemplo a seguir, usamos o método get() para retornar o valor armazenado em nosso contrato:



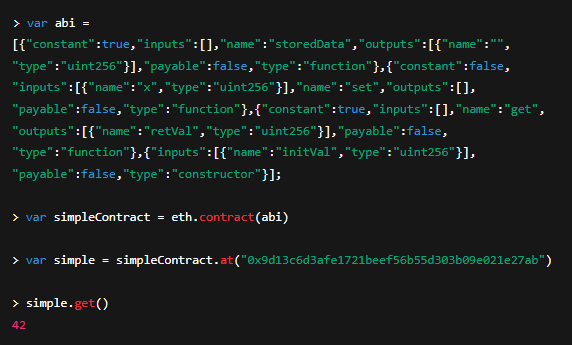
Esse comando retorna **42**, que é o valor armazenado no contrato.

Obs.: Após cada instrução inserida no console do Geth, exceto o último comando simple.get(), uma mensagem undefined também será exibida. Basta ignorar isso; é apenas a forma padrão do JavaScript indicar variáveis não inicializadas, propriedades inexistentes de objetos ou situações semelhantes.

**Nó 2:**  
De forma semelhante ao Nó 1, abra o console do Geth para o Nó 2 com o seguinte comando:



No console do Geth, insira os seguintes comandos:



Assim como no Nó 1, o comando final retorna **42**, que é o valor armazenado.

**Nó 3:**  
Novamente, abra o console do Geth para o Nó 3 com:



No console, insira o seguinte:

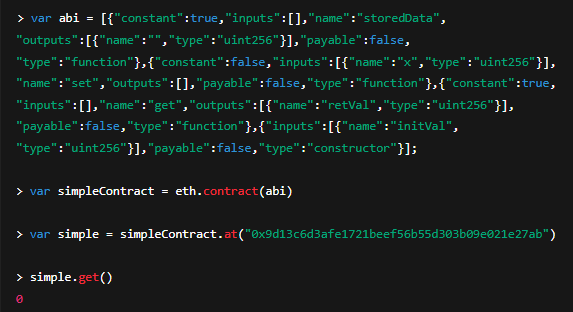


Observe que o valor retornado é **0**, o que indica que o Nó 3 não pode acessar o valor 42, como esperado, já que ele não participa da transação.

**Nó 4:**  
Finalmente, abra o console do Geth para o Nó 4 com:



No console, insira os mesmos comandos que para os nós anteriores:



Como esperado, os Nós 3 e 4, que não participam da transação, verão o valor **0** como resultado.  
Por outro lado, os Nós 1 e 2 conseguem ver o valor real da transação, que é **42**.

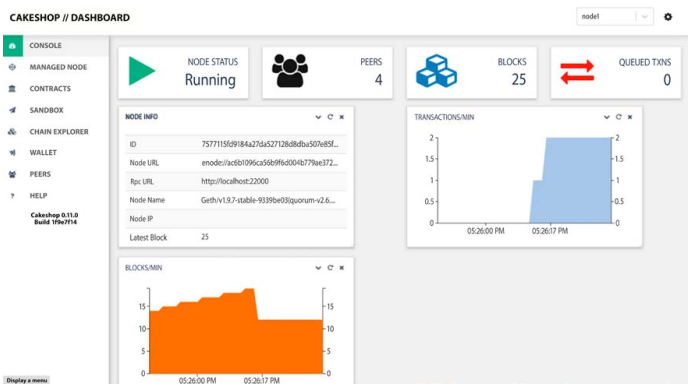
**Visualizando a transação no Cakeshop**

Lembre-se de que, como discutido anteriormente neste capítulo, precisamos monitorar e visualizar redes blockchain empresariais. O **Cakeshop** atende a essa necessidade.

**Cakeshop** é uma poderosa ferramenta de visualização e administração com diferentes funcionalidades, como gerenciamento de nós, explorador de blocos (*block explorer*) e gerenciamento de contratos. O Cakeshop é instalado como parte do processo de criação da rede pelo Quorum Wizard.  
Uma vez que a rede está em execução com sucesso, podemos navegar até:

<http://localhost:8999>

onde o Cakeshop estará sendo executado.

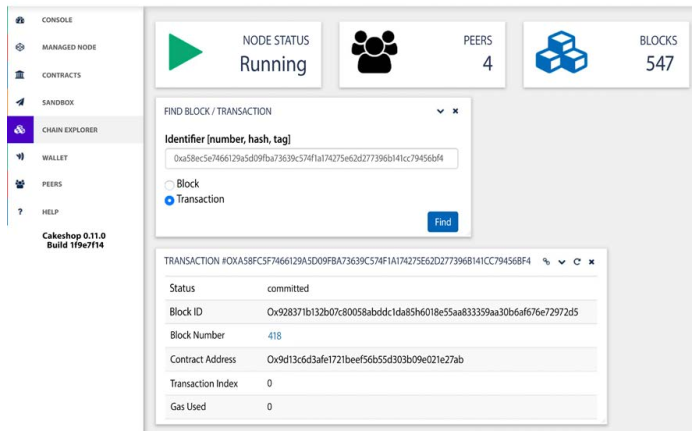


No Cakeshop, podemos ver a transação e seus atributos relevantes. Basta navegar até o **explorador da cadeia** (*chain explorer*) e inserir o **hash da transação**:

0xa58ec5e7466129a5d09fba73639c574f1a174275e62d277396b141cc79456bf4

(retirado da seção *Executando uma transação privada*).

O Cakeshop exibirá o status da transação, o ID do bloco, número do bloco, endereço do contrato e vários outros atributos, o que facilita a exploração detalhada do processo da transação.



**Investigação adicional com Geth**

Também podemos visualizar os recibos de transação e o código do contrato usando o console do Geth.

Para isso, conecte-se ao Geth usando o seguinte comando:

$ geth attach

Faremos isso em cada nó, conforme segue.

**Nó 1:**  
Primeiramente, abrimos o console do Geth usando

$ geth attach qdata/dd1/geth.ipc

No console do Geth, insira o seguinte comando, que usa o hash da transação da etapa de implantação do contrato:



A saída será semelhante a:



Observe que o valor v é 0x26 em hexadecimal, que corresponde a **38** em decimal. Esse valor indica que a transação é **privada**.

Além disso, repare no campo input, que representa o conteúdo criptografado da transação.

Também podemos obter o **recibo da transação** com o comando:



A saída será semelhante a: 

Observe que o campo status tem o valor "0x1", o que indica **sucesso** da transação.  
Para obter o código desse contrato inteligente, usamos o endereço do campo contractAddress:



A saída será semelhante a:

"0x60606040526000357c01000000000000000000000000000000000000000000000000000000900463ffffffff1680632a1afcd914605157806360fe47b11460775780636d4ce63c146097575b600080fd5b3415605b57600080fd5b606160bd565b6040518082815260200191505060405180910390f35b..."

Esse código representa o **bytecode hexadecimal** do contrato inteligente.

**Nó 2 – que participa da transação:**

Assim como fizemos no Nó 1, abra o console do Geth no Nó 2 com o seguinte comando:

$ geth attach qdata/dd2/geth.ipc

No console, execute o seguinte comando para obter o código do contrato inteligente (usando o mesmo endereço do contrato):



A saída será semelhante à seguinte (truncada para brevidade):

"0x60606040526000357c01000000000000000000000000000000000000000000000000000000900463ffffffff1680632a1afcd914605157806360fe47b11460775780636d4ce63c146097575b600080fd5b..."

Como esperado, o **código do contrato é visível no Nó 2**, pois ele **é participante** da transação.

**Nó 3 – que não participa da transação:**

Abra o console do Geth para o Nó 3:

$ geth attach qdata/dd3/geth.ipc

No console, execute o seguinte comando:



A saída será:

"0x"

Esse resultado indica que o **código do contrato não está disponível** neste nó, pois o **Nó 3 não participa** da transação privada.

**Nó 4 – também não participa da transação:**

Abra o console para o Nó 4:

$ geth attach qdata/dd4/geth.ipc

E execute:



Saída:

"0x"

Mais uma vez, o resultado indica que o **contrato não está acessível neste nó**, como esperado.

**Resumo da demonstração**

Este experimento demonstra que o contrato inteligente está **disponível apenas nos nós que participam da transação**.  
Primeiro, implantamos nosso contrato privado a partir do **Nó 2**, incluindo o **Nó 1** como participante.  
Usamos o console do Geth para interagir com o contrato e verificamos que o valor 42 só está acessível nos Nós 1 e 2.  
Além disso, verificamos que **não apenas o valor**, mas **também o código do contrato** só está disponível nos nós participantes da transação privada.

Essa rede agora está disponível para experimentos adicionais.  
Você pode criar seus próprios contratos privados com diferentes combinações de nós como participantes, e também explorar como outros métodos — como debug\_traceTransaction — se comportam ao interagir com contratos privados.

**Outros projetos do Quorum**

Como mencionado anteriormente, o **Quorum** é uma plataforma rica em recursos e está em constante evolução e desenvolvimento. Como resultado, novos recursos e projetos são introduzidos regularmente. Alguns dos outros projetos sob o guarda-chuva do Quorum são listados aqui, juntamente com breves descrições:

**Plugin para o Remix**

Este é um **plugin para o popular IDE Remix**, que oferece suporte a contratos inteligentes privados na blockchain Quorum.  
Mais informações sobre esse plugin estão disponíveis em:

* <https://docs.goquorum.consensys.net/tutorials/quorum-dev-quickstart/remix#docusaurus_skipToContent_fallback>

**Arquitetura plugável**

O Quorum oferece uma **arquitetura plugável** que permite adicionar novos recursos ao cliente Quorum como plugins.  
Essa abordagem permite manter os serviços principais do Quorum **separados dos novos recursos**, o que proporciona **maior modularidade e extensibilidade** sem modificar o cliente central.

Mais informações sobre essa funcionalidade estão disponíveis aqui:

* <https://docs.goquorum.consensys.net/concepts/plugins>

**Encerramento do capítulo**

O **Quorum** é um projeto grande com muitos recursos empresariais excelentes.  
Não é possível cobrir todos eles neste capítulo.  
No entanto, o material fornecido deve ser suficiente para que você comece a trabalhar com o **setup da blockchain Quorum** e entenda o conceito de **transações privadas**.

Há uma grande quantidade de informações disponíveis na **documentação oficial do Quorum**, em:

* <https://docs.goquorum.consensys.net/en/latest/>

Recomenda-se fortemente que você explore essa documentação para obter uma compreensão mais aprofundada do Quorum.

**Casos de uso e disponibilidade**

O Quorum tem sido utilizado em **diversas indústrias**, como:

* Finanças
* Cadeia de suprimentos
* Saúde
* Mídia
* Setor público/governo

Alguns exemplos de uso incluem:

* **Rastreamento de produtos de luxo:**  
  <https://www.coindesk.com/louis-vuitton-owner-lvmh-is-launching-a-blockchain-to-track-luxury-goods>
* **Plataforma de pós-negociação de petróleo:**  
  <https://cointelegraph.com/news/oil-trading-blockchain-platform-vakt-launches-with-shell-bp-as-first-users>

Além disso, o **Quorum** está disponível em várias **plataformas em nuvem**, incluindo:

* [Azure](https://azure.microsoft.com/en-us/solutions/web3/#overview)
* [Kaleido](https://www.kaleido.io)

**Resumo**

Neste capítulo, começamos com uma introdução a soluções empresariais e blockchain.  
Vimos alguns dos fatores limitantes das blockchains públicas que as tornam inadequadas para casos de uso empresariais.  
Também examinamos alguns requisitos que, quando atendidos, tornam uma blockchain adequada para uso corporativo.

Além disso, abordamos como **projetar soluções de blockchain empresarial** e argumentamos que essas soluções devem ser vistas no contexto da **arquitetura corporativa**.  
Exploramos **computação em nuvem** e a definição de **Blockchain como Serviço (BaaS)**.  
Também examinamos brevemente os esforços em curso para **padronizar as especificações do Ethereum empresarial**.

Nas seções finais, abordamos plataformas de blockchain empresarial, incluindo o **Quorum**, e demonstramos como configurar uma rede **IBFT** e implementar transações privadas no Quorum.  
Também apresentamos blockchains empresariais mais recentes, como o **VMware Blockchain**.