@s Livres

Abandonando as plataformas deles,

construindo os nossos protocolos!

por Lienko Labs

2025

Apresentação

Da proposta, premissas e do que se deseja com tudo isso.

A ideia

@s livres quer dar um passo na direção de uma internet social e desintermediada, uma internet liberta das plataformas que fazem o "meio de campo" entre usuários e conteúdos digitais.

Adotando protocolos para interação online, usuários podem trocar conteúdo diretamente, sem intermediação.

É partindo dessa ideia que nasce o projeto @s livres: um conjunto de protocolos que, juntos, criam uma rede descentralizada de comunicação digital. Por meio dessa rede uma pessoa pode publicar online seus conteúdos e acessar conteúdos publicados por outras pessoas, sem ficar presa a uma plataforma ou formato.

Num mundo em que as interações sociais digitais estão quase completamente intermediadas pelas plataformas (Facebook, Instagram, TikTok, etc), @s livres convida toda e qualquer pessoa a buscar essa libertação, se tornando parte da rede e participando da construção coletiva de protocolos sociais diversos e dos próprios protocolos que colocam de pé essa estrutura descentralizada.

Não é preciso ter conhecimento prévio ou ser de uma área tecnológica, basta querer participar. Uma internet verdadeiramente livre só pode ser construída coletivamente!

O projeto

Na ciência da computação, protocolo é o nome que se usa para um conjunto de regras que permite e gerencia a troca de informações entre dispositivos eletrônicos (computadores, celulares, etc).

Então um protocolo é basicamente um programa de computador que define regras que as máquinas (dispositivos) seguem para trocar informações digitais entre si. Não parece, mas isso é bem poderoso.

Protocolos já regem a internet. Toda vez que se acessa um site pelo navegador (seja ele o Facebook ou o site de uma loja de sapatos), protocolos são usados para se comunicar com o computador na outra ponta da web onde a informação está guardada e é processada, e que

responde de volta pra essa máquina através dos mesmos protocolos.

É possível então construir um combo de protocolos que possibilite a interação de pessoas pela internet sem que uma dada plataforma seja dona dessas interações? **Sim, é possível.**

@s livres é uma coleção de protocolos que se propõe exatamente a essa finalidade!

Só isso? Sim, só isso. E isso é bem poderoso.

Contexto

Para explicar a função de cada protocolo proposto na desintermediação das interações, é preciso primeiro entender o papel que esses intermediários (plataformas) desempenham.

Início da intermediação

Antes das plataformas poucas pessoas se davam ao trabalho de construir um site para ter uma presença digital, pra publicar conteúdo próprio na internet. Foi com a chegada delas que o protagonismo do usuário como criador do conteúdo se acelerou.

Elas facilitaram muito o processo de publicar online. Criando interfaces "intuitivas" (e viciantes), e embalando em um único sistema assinatura, controle de acesso, armazenamento, ferramentas de publicação e compartilhamento, regras de formato e curadoria do conteúdo, as plataformas trouxeram pra internet milhões de pessoas que, de outra forma, provavelmente não estariam online.

A inclusão veio com o preço do aprisionamento total desse usuário, que tem sua @ (ou seja, sua assinatura digital), seu conteúdo e todas as suas conexões permanentemente atreladas e submetidas à plataforma, que passa a ser "dona" da persona digital e do conteúdo de cada um de seus usuários, ditando não apenas o regime de visibilidade das publicações, mas também (e talvez principalmente) seus formatos.

Assim as interações sociais online passaram a ser quase que completamente intermediadas, e o sonho da internet como um espaço livre e democrático se tornou mais distante, mesmo com a massiva presença digital das pessoas.

Apropriação

Os conteúdos publicados pelos usuários são armazenados e distribuídos "gratuitamente" pelas plataformas.

- Da perspectiva do **armazenamento**

  As informações (sejam vídeos, textos, reações, etc) criadas pelos usuários são salvas pela plataforma usada na publicação, e ficam a ela submetidas. Não há soberania sobre os conteúdos. A plataforma se torna dona da informação no sentido físico, já que a informação fica presa nos servidores deles (sem garantias de que o usuário possa recuperar ou apagar o conteúdo caso deseje). Também se torna dona no sentido jurídico, pois passa a

poder usar o conteúdo para diversos fins (treinamento de IA, publicidade direcionada ao usuário, ações de marketing da própria empresa — até sem conhecimento dos autores).

- Da perspectiva da **distribuição**

  As publicações tem um regime de visibilidade opaco, e não é possível saber quando (e se!) uma conexão é notificada sobre uma dada publicação. As plataformas distribuem os conteúdos usando um sistema de curadoria que atende seus próprios interesses e agendas, interesses esses que não coincidem com os do usuário.

É um contra-senso que para publicar um mesmo conteúdo através de várias plataformas diferentes seja preciso criar cópias da mesma postagem em cada uma delas (mesmo quando não é necessário um ajuste de formato!).

Uma solução livre precisa garantir que o armazenamento possa acontecer de forma independente, e que quaisquer que sejam as publicações que referenciem um mesmo conteúdo, ele possa ser acessado de forma direta.

Uma solução livre também precisa garantir que a distribuição das publicações possa ser transparente e democrática, de forma que as aplicação e interfaces disponíveis privilegiem os interesses do usuário.

Desacoplamento, por favor!

Promovido como uma facilidade, o acoplamento de múltiplas funcionalidades em uma única aplicação é na verdade o mecanismo que mantém o usuário aprisionado.

Não há motivo tecnológico para o acoplamento, a razão é o modelo de negócio. Algumas das principais funções propositalmente acopladas pelas plataformas:

- **Armazenamento da mídia**
- **Identidades** (as @s)
- **Interações** (postar, curtir, seguir, comentar, compartilhar, doar, pagar, contratar...)

  Num mundo desacoplado, as interações seriam o que o protocolo social proveria
- **Curadoria**

  Hoje realizada de forma opaca pelos algoritmos de recomendação das Big Techs, pelo regime de visibilidade de postagens
- **Interface**

  A interface gráfica, onde o usuário consegue interagir com a aplicação, que hoje é totalmente engessada

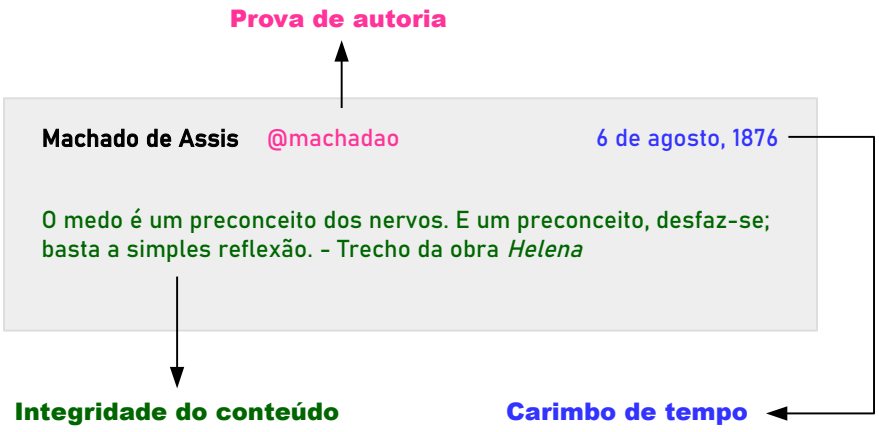

Figura 1: Garantias necessárias para interações digitais, hoje providas pelo intermediário.

Essas funcionalidades são distintas, independentes entre si, e não deveriam ser oferecidas pelas mesmas empresas/produtos/aplicações.

Ao migrar para tecnologias de desintermediação, ou seja, **adotando protocolos para as interações digitais, todas essas funcionalidades são desacopladas como consequência.**

Serviços prestados

As atribuições ligadas à intermediação das interações sociais digitais que de fato precisam ser atendidas, e que são realizadas pelas plataformas, são:

- **Prova de autoria**

  Quando um conteúdo é publicado, a plataforma garante a autoria daquele conteúdo. Ela garante a autenticidade da @ que assinou cada postagem.

- **Integridade do conteúdo**

  A plataforma garante que um conteúdo já publicado não possa ser adulterado ou corrompido. Ela assegura que um conteúdo assinado e publicado permanece íntegro ao ser acessado pelos demais usuários da rede.

- **Carimbo de tempo e assincronia**

  Toda publicação tem um carimbo de tempo. Ao disponibilizar um conteúdo publicado, a plataforma garante a data e horário associados ao momento da publicação. Garante também que esse conteúdo possa ser acessado de forma assíncrona: não é preciso estar online no momento da publicação para ver o conteúdo, ele pode ser visto em um momento posterior.

A figura 1 representa essas atribuições em um exemplo de publicação.

Tirando o intermediário

A criptografia já tem capacidade de libertar o usuário de dentro das plataformas.

Existem hoje tecnologias bem estabelecidas com capacidade de realizar as principais funções das plataformas sem a necessidade de uma autoridade centralizada, sem um intermediário promovendo essas garantias.

- **Prova de autoria → Assinatura criptográfica de chave pública**

  A autoria do conteúdo pode ser garantida sem intermediação por meio da assinatura criptográfica de chave pública. Basta o usuário associar sua @ (apelido) à chave pública de um sistema de criptografia assimétrico (soa complicado, mas não é! Assinaturas assim já são usadas rotineiramente por milhões de pessoas em aplicações como as de banco).

  Ao assinar o conteúdo publicado online com a chave privada, qualquer pessoa com acesso à chave pública associada à @ pode verificar a autoria, num esquema em que nenhuma autoridade central é necessária para garantir a validade da assinatura.[1]

- **Integridade do conteúdo → Hash**

  A integridade de um conteúdo pode ser garantida criando o hash do conteúdo no momento da publicação e disponibilizando essse hash junto ao conteúdo. Como o hash é único para cada conteúdo, uma espécie de CPF da informação, ao disponibilizar o hash uma verificação posterior pode ser feita diretamente, garantindo a integridade sem necessidade de uma autoridade central.[2]

- **Carimbo de tempo e assincronia → Blockchain**

  A tecnologia de blockchain resolve a questão do carimbo de tempo e do acesso assíncrono ao conteúdo publicado. Se cada conteúdo assinado e associado ao seu respectivo hash for publicado via blockchain, haverá um carimbo de tempo atrelado a essa publicação, e o bloco da blockchain em que a publicação for processada e suas informações temporais podem ser verificados sem necessidade de intermediário. Como a blockchain fica disponível publicamente, acessar de forma assíncrona as informações dessa publicação também é resolvido pela tecnologia.[3]

A construção dos protocolos que compõe @s livres parte da premissa de que o conjunto dessas tecnologias já pode libertar os usuários!

No mundo desacoplado

O armazenamento de mídia pode sempre ser feito de forma totalmente independente, e com várias possibilidades de provedores (inclusive o auto-armazenamento).

[1] Um pouco mais sobre assinaturas criptográfricas de chave pública no anexo A.
[2] Mais sobre hashs no anexo B.
[3] Mais sobre a tecnologia de blockchain no anexo C.

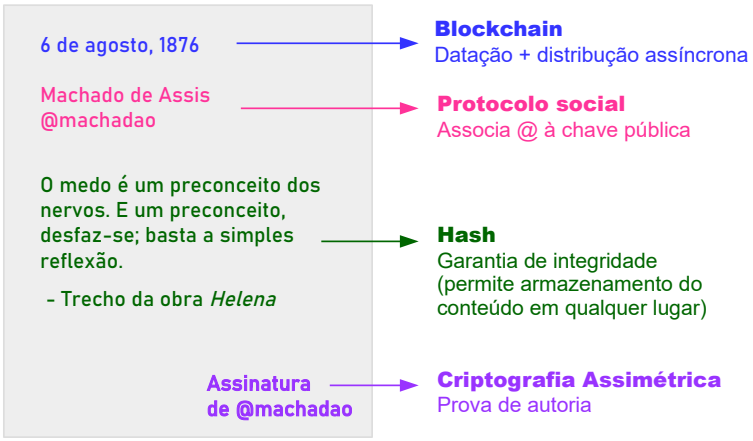

Figura 2: Garantias necessárias para interações digitais atendidas de forma desintermediada.

O protocolo que gerencia as identidades pode incluir uma referência para o local desse armazenamento (referência que pode ser atualizada caso o provedor mude). O hash da mídia funciona como chave entre a publicação na blockchain e arquivo no local do armazenamento.

No caso da interface e da distribuição isso é ainda mais intuitivo, pois com todas as interações gerenciadas por protocolos que alimentam a blockchain (pública), qualquer interface pode implementar o envio de novas instruções e fazer a leitura de instruções anteriores que já estejam na blockchain.

Com todas as conexões e complexidades do social contempladas via protocolo, nada se perde entre aplicações. A escolha de interface adotada por qualquer participante da rede passa a ser indiferente. Qualquer um pode construir uma interface e garantir seu regime de distribuição, e nenhuma interface pode aprisionar o usuário e suas interações lá dentro.

A figura 2 representa um exemplo de instrução para publicação digital num mundo desacoplado, com as atribuições atendidas de forma desintermediada.

A internet, livre.

O que estamos esperando?

Para além da viabilidade das tecnologias existentes, @s livres parte de três premissas.

- **Libertar o usuário deve ser um trabalho coletivo.**

  É preciso envolver muito mais do que um grupo pequeno de tecnólogos e/ou capitalistas para construir uma solução livre que seja melhor e mais atraente que as opções atualmente oferecidas pelas grandes plataformas. Pra isso é preciso envolver no processo de criação de alternativas pessoas diversas em todos os sentidos, que possam trazer aprendizados dos mais variados campos do conhecimento humano.

- **A construção coletiva nunca vai acontecer a partir das ferramentas que já temos.**

O desenho das aplicações digitais disponíveis hoje segue uma filosofia que é, por definição, de individualismo. Tudo é pensado para indivíduos, perfis. Dificilmente migrar a discussão e organização de alternativas do Twitter para o Mastodon, ou do Whatsapp para o Signal, resolverá o problema estrutural desses desenhos, que causam mais ruído do que convergência. Ainda que seja importante fortalecer e divulgar soluções livres, o design dessas aplicações não foi concebido para a construção coletiva de algo novo.

- **Não estão sendo exploradas todas as possibilidades do digital.**

  Muito pelo contrário, estamos empacados com aplicações criadas para nos viciar, que aprisionam formatos, pensamentos e relações. Existe um universo de possibilidades inexploradas para interações digitais, formatos não experimentados e que promoveriam espaços digitais muito diferentes dos que temos hoje. Para explorar toda essa riqueza criativa é necessário um ambiente propício para essas experimentações.

A inauguração de um espaço digital pensado para fomentar experimentação coletiva e colaborativa é suficiente para testar as três premissas!

Construindo de dentro

Para testar a viabilidade dessa visão e dar um passo incial na jornada de libertar o usuário, @s livres parte de uma suite (conjunto) de protocolos.

- **Passo 1: Brisa**

  Historicamente, a construção começa com um protocolo de consenso pensado para interações digitais, que fica responsável por colocar de pé uma rede cripto descentralizada que irá processar essas trocas de informação. Ao abrir mão de um pouco de segurança, é possível conseguir maior capacidade de processamento em uma rede cripto. Assim, se propõe a rede *Brisa*.

- **Passo 2: Apelidos**

  Com uma versão inicial para o funcionamento da rede descentralizada em pé, segue a questão do gerenciamento das identidades. É preciso propor uma solução para associar as @s (ou apelidos) dos usuários às suas respectivas chaves públicas, ainda garantindo a soberania dos usuários sobre suas @s. Assim, um segundo protocolo responsável pelo gerenciamento desses identificadores é construído para rodar em cima da rede cripto. É o protocolo *Apelidos*.

- **Passo 3: Motiró**

  Para concluir, é testado o conceito de experimentação digital, com a proposta de um protocolo social para construção coletiva: *Motiró*. Ele roda sobre a rede descentralizada *Brisa* numa camada acima da camada do protocolo *Apelidos*, e funciona como um espaço digital

para construções coletivas, onde publicações e curadorias podem ser assinadas e gerenciadas por coletivos, e não apenas indivíduos.

Motiró é o carro-chefe da proposta inicial de desintermediação, testando a possibilidade de construção de alternativas de dentro da nova tecnologia proposta, numa espécie de bootstraping, onde ***Motiró* existe como o local da construção de si mesmo e da infraestrutura que o sustenta**.

Além disso, propor coletivamente e construir protocolos sociais como o *Motiró* se torna bem mais simples partindo da estrutura criada. É mais fácil experimentar quando não é preciso construir toda a infra do zero para testar uma ideia.

Motiró inaugura então um espaço para construir coletivamente essas alternativas!

Conceito de protocolo social

Protocolos sociais, para @s livres, são os conjuntos de regras que ditam o modo de uma dada interação digital. As regras são primeiro imaginadas, definidas, e então são enforçadas pela implementação (pela programação, literalmente) do protocolo.

Alguns exemplos de formatos de interações digitais bem conhecidos e que poderiam representar protocolos sociais:

- **Email:** formato de interação onde há um endereço e uma caixa de entrada aberta (não é preciso pedir permissão para enviar um email para um endereço). Tem funcionalidade de encaminhamento livre sem verificação de autoria. Conteúdo trocado é livre e não moderado.
- **Fórum de internet:** ambiente digital com ou sem moderação para publicação de membros. O ambiente pode ou não ser público. A única funcionalidade adicional é responder a uma publicação.
- **ICQ:** formato de interação que permite troca de mensagens de forma síncrona entre usuários compartilhando um mesmo canal.
- **Twitter:** ambiente digital aberto que permite a publicação de um texto curto assinado. Posteriormente incluiu a funcionalidade de 'retweet', curtir e responder.

O protocolo social proposto por @s livres é *Motiró*: um ambiente digital onde é possível performar ações em nome de um coletivo. Coletivos são formados por grupos dinâmicos de usuários. Qualquer ação realizada em nome do coletivo gera uma verificação automática de maioria. Coletivos podem assinar publicações, promover e endossar conteúdos, e agendar e gerenciar eventos.

Motiró é uma proposta inicial, um primeiro passo para facilitar a experimentação coletiva, longe de considerar ser o único, ou de estar fechado em si. @s livres apresenta a infra para que **qualquer pessoa ou comunidade possa explorar, propor e experimentar protocolos sociais alternativos.**

Convite

A proposta de @s livres é, acima de qualquer coisa, um convite à participação de todas as pessoas que desejam colaborar com a construção coletiva de alternativas livres para interação digital. Todo mundo é bem-vindo a se juntar ao projeto, testar a tecnologia, propor o que ela deve se tornar.

Em especial, é um convite a grupos de quaisquer origens e áreas de atuação a propor o tipo de formato de interação digital — o protocolo social! — que gostariam de ver funcionando. @s livres traz uma infraestrutura pensada para testar possibilidades, novos desenhos e formatos.

Não há uma resposta pronta de como várias dessas dinâmicas devem se mover, mas a partir de um espaço coletivo onde construir, é possível desenhar as soluções que funcionam para cada comunidade, para cada propósito.

A ideia geral da proposta está ilustrada em `https://freehandle.org/` (contando inclusive com uma história em quadrinhos!). O site também se liga a uma interface funcional de Motiró.

Protocolos

Um pouco sobre o funcionamento dos três principais protocolos que compõe @s livres: *Brisa*, *Apelidos* e *Motiró*. E sobre como eles se conectam.

Brisa: a rede cripto

A primeira camada da estrutura é o protocolo *Brisa*, que estabelece o funcionamento da rede cripto e as regras de consenso Proof-of-Stake[4]. A partir dessa camada, outras blockchains e outros protocolos podem ser implementados.

O importante é que *Brisa* coloca de pé a funcionalidade da publicação aberta, não censurada, descentralizada e com prova de marcação temporal.

Processar as interações em uma rede descentralizada garante a liberdade dos usuários, pois ninguém se torna "dono" da rede, ela existe como nós independentes que processam um mesmo protocolo e concordam com as mesmas saídas desse processamento.

O desafio para o caso das interações sociais é que a rede cripto proposta precisa conseguir processar uma grande quantidade de instruções. Não é necessário o mesmo nível de segurança, por exemplo, de uma rede cripto pensada para processar transações financeiras. O importante é que a principal função nesse caso — a garantia de marcação temporal das instruções referentes aos protocolos sociais — esteja garantida.[5]

Para desacoplar as funções da rede cripto daquelas dos protocolos sociais, uma solução inédita foi desenvolvida: a instrução **void**.

Instrução Void, a novidade

A palavra void, que pode ser traduzida como "vazio", faz referência na programação a funções que não retornam nenhum valor pra quem as chama.

Brisa prescreve então a existência de uma instrução do tipo **void**, que é transparente para o protocolo. Essa instrução é aceita e processada, mas não tem efeito sobre a rede para além do gerenciamento de tokens de processamento.

A instrução **void** é a grande inovação proposta por *Brisa* em relação a outras rede cripto. *Brisa* a propõe como forma de embarcar protocolos subjacentes sem aumentar a complexidade

[4]Mais sobre o mecanismo de consenso Proof-of-Stake no anexo D.

[5]Mais sobre o funcionamento de *Brisa* e de seu protocolo de consenso *Maré* no anexo E.

de suas próprias funções, sendo a instrução void o ponto de partida para a experimentação de protocolos sociais em cima da rede cripto.

É como se void funcionasse como uma carta, enquanto a rede regida pelo protocolo *Brisa* funcionasse como o correio. O correio cobra pelo selo e encaminha a carta. O conteúdo da carta, no entanto, é indiferente do ponto de vista do funcionamento do correio. O correio não precisa abrir nem verificar o que está escrito para realizar sua função.

Dentro da instrução void é possível incluir instruções direcionadas a um protocolo específico subjacente (um protocolo social, por exemplo!), de maneira que isso fique transparente para o protocolo que processa void.

Esse formato pode ser aplicado em cascata, num esquema em que cada protocolo prescreve sua própria instrução void para permitir que outro protocolo subjacente seja processado e, com isso, sejam incluídas novas funcionalidades.

Voids em cascata

Se cada protocolo subjacente incluir por si uma instrução do tipo void, esse mecanismo permite o processamento de protocolos em sequência, onde cada protocolo só se preocupa em validar o que a ele corresponde, passando para frente o que será processado pelo protocolo seguinte.

Da mesma forma, o protocolo subjacente não precisa se preocupar com a funcionalidade do protocolo anterior, e pode ser construído pensando apenas nas suas próprias especificidades.

É nesse formato que @s livres é construído, com o protocolo *Brisa* rodando a rede cripto (que resolve o carimbo de tempo), a instrução void de *Brisa* usada para rodar o protocolo subjacente *Apelidos* (que resolve as assinaturas e identidades), a instrução void de *Apelidos* usada para rodar o protocolo subjacente *Motiró* (que inclui o hash do conteúdo e resolve ação coletiva).[6]

A tarefa fundamental de *Brisa*, pensando em desintermediação, é publicar a instrução com um carimbo de tempo. Então é ideal que o protocolo que coloca a rede *Brisa* de pé não precise se preocupar com mais nada.

O mesmo vale para os demais protocolos. *Brisa* não valida nada que esteja dentro do seu void. Da mesma forma, nenhum protocolo nunca valida nada que esteja dentro de seu próprio void. Cada protocolo consome e processa apenas o conteúdo a ele pertinente, que chega através da instrução void do protocolo anterior. O que não lhe pertence é passado adiante em nova instrução void.

Assim, as atribuições de todas as etapas da infraestrutura ficam desacopladas, com cada protocolo especializado cuidando apenas de sua própria função. É um formato que segue a filosofia UNIX: faça apenas uma coisa, bem feita, e passe para a frente!

É possível comparar esse desenho a uma cadeia de distribuição. O pacote (instrução void de *Brisa*) chega ao centro de distribuição regional (protocolo *Brisa*). *Brisa* verifica as informações da etiqueta que a ele se referem e encaminha o pacote — sem checar seu conteúdo — ao centro

[6] Os protocolos *Apelidos* e *Motiró* são explicados melhor nas duas próximas seções.

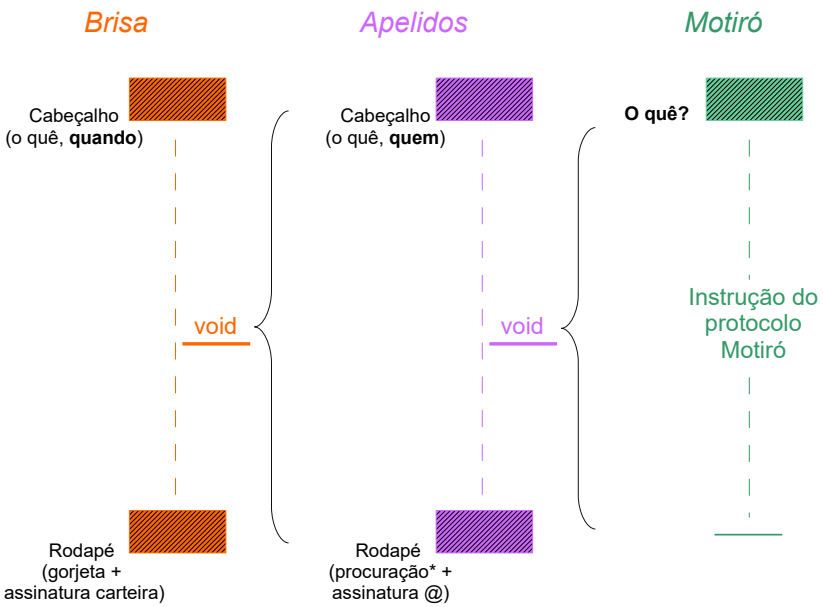

Figura 3: Formato em cascata dos principais protocolos que compõe @s livres, implementado através das instruções **void**.

de distribuição local (protocolo *Apelidos*), que verifica as informações da etiqueta que a ele se referem e encaminha o pacote (instrução void de *Apelidos*) — sem checar o conteúdo — para o destinatário final (protocolo *Motiró*). Ao receber o pacote, *Motiró* abre e verifica se o conteúdo está de acordo com as normas esperadas. Caso esteja, ele aceita o conteúdo e o diponibiliza para uso/consulta. Caso não, o conteúdo é descartado.

Um esquema do formato em cascata dos principais protocolos que compõe @s livres pode ser visto na figura 3, e sua composição na figura 4.

Não há até o momento (e até onde foi possível verificar) informação de nenhum uso parecido para protocolos de consenso e blockchains. A instrução **void** é a grande novidade na proposta de *Brisa*. A primeira rede cripto a propor esse tipo de instrução.

Inclusive, parece uma boa ideia que todo protocolo social proposto considere incluir uma instrução void, garantindo assim que seja sempre possível expandir sua funcionalidade.

Apelidos: o gerenciador de identidades e procurações

A publicação com carimbo de tempo é resolvida pela rede cripto estabelecida por *Brisa*. Assim, a segunda camada da estrutura proposta por @s livres é o protocolo *Apelidos*, que vai lidar com o gerenciamento de identidades.

Para resolver a questão das assinaturas na desintermediação das interações sociais, é preciso associar uma chave criptográfica pública[7] a cada nome de usuário — que no caso seria a @ — que participe da rede.

[7]Mais sobre assinaturas criptográficas de chave pública no anexo A.

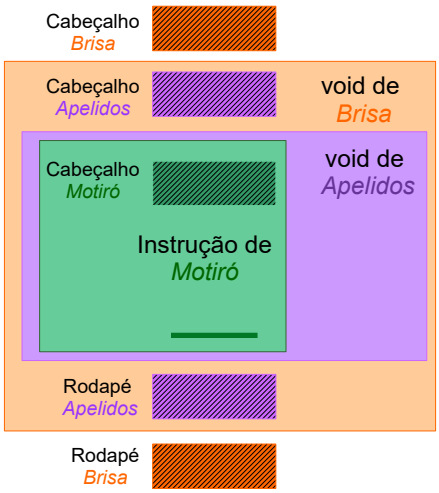

Figura 4: Composição em cascata das instruções void dos principais protocolos de @s livres.

Além de associar as @s às chaves públicas, *Apelidos* propõe uma segunda funcionalidade, a concessão ou revogação de procurações.

Procurações

Para **garantir a liberdade do usuário** mantendo a possibilidade de que terceiros possam fornecer aplicações para interagir via os protocolos, foi pensado o esquema da procuração. A ideia é que usuários possam conceder procurações para que aplicações assinem conteúdo em seu nome, caso assim desejem.

Essa funcionalidade remove o ruído da interface do usuário. Para adotar o uso de uma interface/aplicação provida por terceiros para interagir via um dado protocolo, basta conceder uma procuração à chave pública associada a esse provedor.

No momento em que o usuário decidir trocar de aplicação, por qualquer motivo, ele pode simplesmente revogar a procuração concedida àquela aplicação, sem perda alguma. Ao revogar uma procuração o usuário mantém sua @, suas conexões, seu conteúdo, suas interações.

Tudo está desacoplado e funciona de forma independente, rodando em protocolos, então qualquer que seja a interface ou a aplicação escolhida para enviar as instruções e ler instruções já processadas, ela é indiferente do ponto de vista dos usuários da rede.

Isso porque qualquer que seja a aplicação escolhida por cada participante do protocolo, o formato e o tipo das instruções trocadas será mantido (enforçado pelo próprio protocolo). Ao enforçar que toda a complexidade social seja contemplada pelo protocolo, a interface se torna indiferente.

A possibilidade de concessão e revogação de procurações, livre e sem perdas para o usuário, é a garantia de sua liberdade das aplicações ou plataformas.

Usuários diferentes podem usar interfaces providas por atores completamente diferentes e independentes, e sua interação acontecerá da mesma forma. Qualquer usuário pode alterar, a

qualquer momento, o provedor de sua interface.[8]

É indiferente para qualquer participante da rede qual a aplicação usada para interagir via protocolos!

Motiró: o protocolo social para construção coletiva

Com toda a estrutura de gerenciamento de identidades, assinaturas e procurações provida por *Apelidos* rodando em cima da rede *Brisa*, está de pé a base para experimentar com protocolos sociais!

Para testar a proposta, para servir como local de embarque de novos usuários, e para funcionar como um espaço digital de construção coletiva, foi construído o protocolo social *Motiró*.

A ideia é prover uma funcionalidade que na prática não existe na internet: a ação digital coletiva, permitindo publicações em co-autoria, ou em nome de grupos de pessoas, de acordo com regras pré-acordadas para aceite.

Entre as principais abstrações trazidas por *Motiró* estão:

- **Coletivos:** grupos dinâmicos de participantes gerenciados por uma regra de maioria

  Coletivos agem como uma entidade única dentro do protocolo, podem assinar publicações, endossar conteúdos, criar espaços de curadoria e marcar eventos.

- **Consenso:** votações como consequência, não como funcionalidade

  No sentido de que quaisquer ações tomadas em nome do coletivo, por qualquer um de seus integrantes, geram automaticamente uma verificação de maioria entre os demais integrantes do coletivo, de acordo com a regra de consenso pré-acordada.

- **Carimbo:** como forma de coletivos endossarem conteúdos

  Coletivos podem carimbar um conteúdo que julgam que mereça seu endosso, como uma espécie de "selo de qualidade" de publicações quaisquer dentro do protocolo.

- **Murais:** um espaço de curadoria "lenta", onde conteúdos podem ser destacados num regime de visibilidade de maior permanência

  Coletivos podem gerenciar murais, onde conteúdos pode ser afixados e desafixados, criando um espaço digital de maior retenção para conteúdos considerados pertinentes, ou que devem permanecer com visibilidade por um período de tempo maior. A ideia é criar um espaço digital inspirado nos murais analógicos, de faculdades e escolas.

Motiró foi pensado, principalmente, para ser o lugar da construção coletiva de si mesmo, e da infraestrutura que o coloca de pé (*Brisa*, *Maré*, *Apelidos*). Ao ingressar no coletivo **Motiró**, construído dentro do protocolo *Motiró*, usuários passam a fazer parte das votações que podem alterar o protocolo, e propor mudanças, endossos, novos protocolos.

[8] Mais detalhes sobre a estrutura de *Apelidos* e as ações por ele processadas se encontram no anexo H.

Em *Motiró* há um coletivo para cada protocolo de @s livres e há, também, o coletivo @s livres.

Quanto mais pessoas participarem das escolhas e do que deve ser a construção dessa tecnologia, melhor esses protocolos refletirão o que as pessoas querem ver funcionando.

Motiró está no ar, e pode ser acessado por uma interface beta em https://freehandle.org/synergy/. [9]

Licenças

Os protocolos que compõe @s livres estão em **código aberto e livre**.

São publicados sob a licença permissiva Apache 2.0 e sua implementação está disponível em https://github.com/freehandle.

[9] Mais detalhes sobre o protocolo *Motiró* no anexo I.

Anexos

A. Como funciona assinatura criptográfica de chave pública?

Criptografia é o estudo e uso de técnicas para estabelecer comunicação segura na presença de possíveis adversários.

Adversários são atores que desejam de alguma maneira se intrometer nessa comunicação (tentando interceptar ou alterar o conteúdo da mensagem, enviar mensagens falsas, etc.)

As técnicas de criptografia atuais se fazem valer do fato de que a segurança da informação deve ser garantida apenas pela força das chaves trocadas, ou seja, os algoritmos usados para criptografar e para descriptografar a informação em qualquer esquema criptográfico podem, e devem, ser públicos.

Há dois principais grupos de técnicas de criptografia:

- **criptografia de chave privada:**

  As duas partes trocam um segredo (chave) previamente e utilizam esse segredo para criptografar e descriptografar mensagens trocadas posteriormente. Também é chamada de **criptografia simétrica**, já que ambas as partes utilizam uma mesma chave, e essa mesma chave tanto criptografa quanto descriptografa as informações trocadas.

- **criptografia de chave pública:**

  O destinatário da mensagem gera um par de chaves. Esse par é composto por uma chave pública e uma chave privada. A chave pública do destinatário pode ser disponibilizada por um canal inseguro. Essa chave pública é então usada pelo remetente da mensagem para criptografar a informação, que só pode ser descriptografada utilizando a chave privada do destinatário.

  A chave privada do par gerado só está em posse do destinatário (ninguém mais possui acesso a essa chave), por isso é garantido que apenas ele seja capaz de descriptografar a mensagem. A mensagem criptografada pode então ser trocada por um canal inseguro.

  Também é chamada de **criptografia assimétrica**, pois cada parte possui seu próprio par de chaves para estabelecer a troca de mensagens, e a chave usada para criptografar a informação é diferente da chave usada para descriptografar.

Para assinar uma mensagem, um esquema de assinatura de chave pública pode ser feito da seguinte maneira:

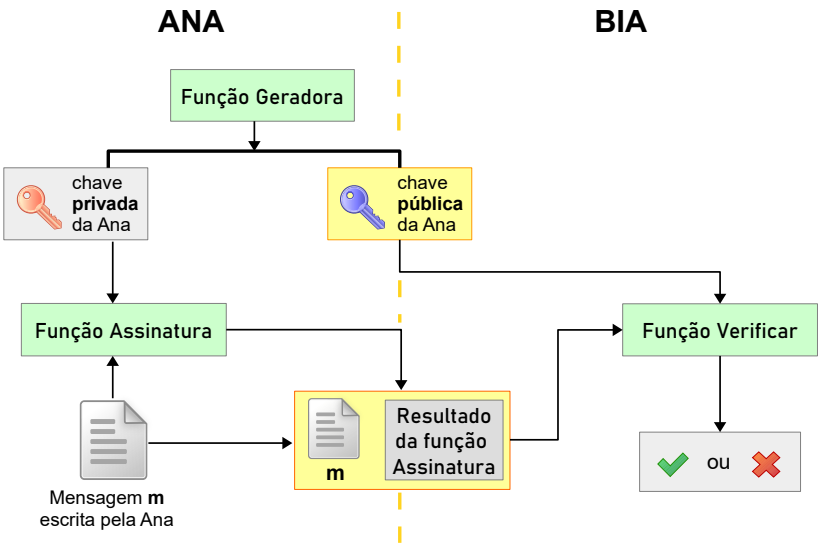

Figura 5: Esquema de assinatura criptográfica de chave pública. Ana gera o par de chaves (pública e privada), escreve a mensagem **m** e computa a função assinatura publicando **m** e o resultado. Bia, a partir da publicação e da chave pública de Ana computa a função verificar, cujo resultado valida (ou não) a assinatura.

- O remetente da mensagem gera o par de chaves (chave pública, chave privada) a partir da função geradora
- A chave pública do par é disponibilizada publicamente, de forma que qualquer um possa acessá-la
- Quando o remetente deseja enviar uma mensagem m, ele computa a função assinatura de m, utilizando para isso sua chave privada do par gerado
- O remetente publica a mensagem e o resultado da função assinatura
- Qualquer pessoa que queira verificar a veracidade da assinatura computa o resultado da função verificar da mensagem m e da assinatura, utlizando para isso a chave pública do remetente
- A veracidade da assinatura é garantida se o resultado da função verificar for válido

Esse esquema está ilustrado na figura 5.

A partir do momento da publicação da assinatura, que garante a autenticidade do remetente como dono daquela chave privada, todas as mensagens posteriores usando a chave tem sua autoria garantida.

Qualquer um, a qualquer momento, pode verificar a assinatura da mensagem (propriedade de transferabilidade) através da chave pública. É impossível negar uma assinatura, no sentido de que uma vez publicada e assinada uma mensagem, há garantia de que o autor é de fato o

remetente (não-repudiabilidade).[10]

Assim, ao publicar e assinar uma mensagem associando uma arroba a uma chave pública, a autoria das publicações pode ser garantida posteriormente sem necessidade de intermediação.

O esquema de criptografia de chave pública adotado por @s livres é $ed25579$[11].

B. Como funciona um hash?

Funções hash são funções matemáticas que mapeiam uma informação de tamanho arbitrário para uma informação de tamanho fixo (como exemplifica a figura 6).

Essas funções são irreversíveis, no sentido de que não é possível a partir do resultado da função (informação de tamanho fixo) identificar qual foi a entrada usada (informação original, de tamanho arbitrário).

No entanto, sempre é possível garantir a igualdade entre duas informações a partir do resultado do hash, pois duas entradas diferentes nunca geram resultados de hash iguais (para todos os sentidos práticos).[12]

Por funcionarem como uma espécie de código de identificação para uma informação, o resultado de uma função hash pode ser usado para verificar a integridade de um conteúdo, já que é sempre possível computar o hash de uma informação e compará-lo ao hash recebido.

No caso de @s livres, a premissa é que uma instrução de publicação pode sempre conter o hash do conteúdo publicado, ainda que o conteúdo em si esteja armazenado em outra localidade referenciada (fora da blockchain, por exemplo). Esse hash é a garantia de que a integridade do conteúdo publicado sempre pode ser posteriormente verificada.

@s livres usa a função SHA-256 para gerar os hashs. [13]

C. Como funciona uma blockhain?

Blockchain é uma tecnologia que permite que computadores conectados em uma rede possam construir um registro histórico distribuído e público de um conjunto de variáveis.

As variáveis podem conter informações das mais diversas, desde cadastros de apelidos e publicações (como é o caso de @s livres), até dados de transações financeiras (como é o caso de Bitcoin).

Blockchains funcionam gravando novos registros de forma cronológica, formando uma longa corrente de blocos que não pode ser posteriormente alterada. Essa imutabilidade vem

[10] Este anexo foi baseado no conteúdo de Introduction to Modern Cryptography, por Jonathan Katz e Yehuda Lindell, 2007, CRC PRESS.

[11] Josefsson, S.; Liusvaara, I. (Janeiro de 2017). Edwards-Curve Digital Signature Algorithm (EdDSA). IRTF. doi:10.17487/RFC8032

[12] https://mathworld.wolfram.com/HashFunction.html

[13] Uma visualização das etapas de cálculo do hash SHA-256 para uma entrada de texto pode ser vista em https://sha256algorithm.com/

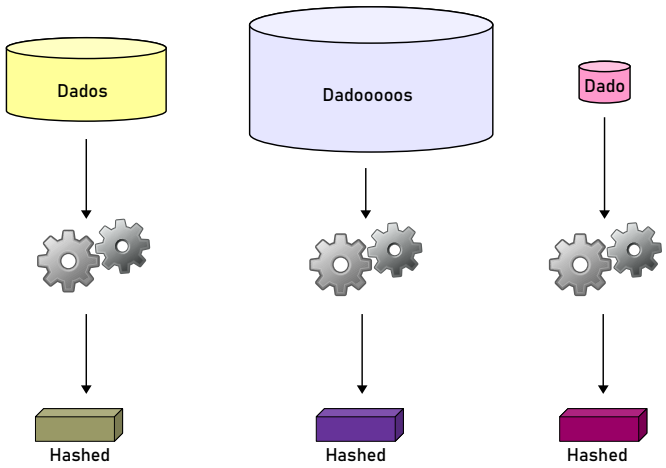

Figura 6: Representação da função hash aplicada a conjuntos de dados diferentes. Não importa o tamanho da informação de entrada, o hash resultante tem tamanho fixo. Também funciona como código de identificação pois, para todos os sentidos práticos, o hash resultante é único para cada conjunto de dados.

do fato de que cada novo bloco incluído na blockchain referencia o bloco anterior. Sendo assim, para alterar um bloco no passado, é preciso alterar também todos os blocos seguintes. Um exemplo desse mecanismo encontra-se na figura 7.

A construção distribuída se dá por meio de um mecanismo de consenso que permite que participantes da rede concordem a respeito das novas instruções que serão registradas na blockchain, ou seja, dos novos blocos a serem incluídos.

A garantia de integridade do registro através desse mecanismo de consenso não exige uma autoridade central para se manter funcionando, ou seja, o consenso é atingido de forma distribuída a partir da formação de um sub-grupo dos participantes, e está garantida mesmo que uma parcela desses participantes não seja confiável.

Uma vez que chega-se a um consenso sobre um bloco, ele é incluído na blockchain e divulgado para os demais participantes da rede (computadores conectados à rede mas que não participaram da formação do consenso).

Essas cópias vão sendo atualizadas por cada participante da rede cripto (os chamados nós). Cada nó pode verificar por si o estado do registro das variáveis.

A tecnologia blockchain foi primeiramente proposta por Satoshi Nakamoto[14] em 2008. Desde sua primeira aplicação (Bitcoin) centenas de blockchains alternativas foram testadas e propostas, em sua maioria tendo como foco o processamento de transações financeiras.

Da perspectiva de @s livres, a tecnologia de blockchain tem como funcionalidade principal um carimbo de tempo garantido e descentralizado nas publicações, e o protocolo de consenso

[14]Bitcoin: A Peer-to-Peer Electronic Cash System, Satoshi Nakamoto. Disponível em português em https://bitcoin.org/files/bitcoin-paper/bitcoin_pt.pdf

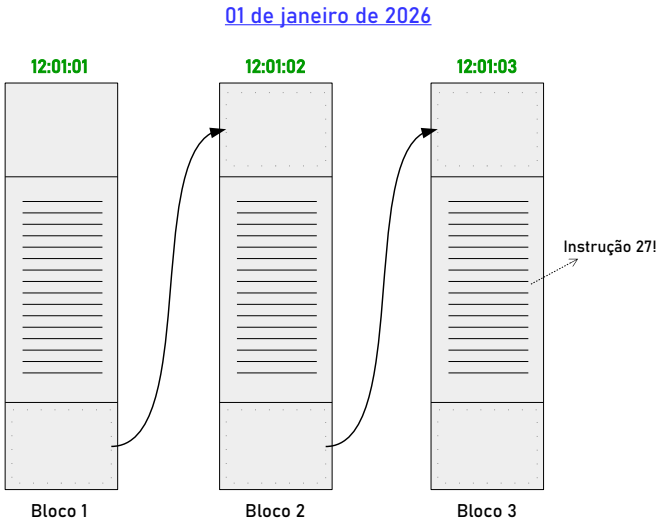

Figura 7: Representação do funcionamento da blockchain. Qualquer um pode conferir uma instrução e o horário em que foi processada.

utilizado para a construção da blockchain *Brisa* é baseado no protocolo Tendermint.[15]

D. O que é Proof-of-Stake?

A formação descentralizada da blockchain pela rede cripto depende do protocolo de consenso, que funciona a partir da seleção dinâmica de um sub-grupo de participantes da rede (nós) que formam um comitê para votar a inclusão de cada novo bloco.

A formação desse consenso precisa acontecer mesmo quando parte dos nós participantes não é confiável.

Na primeira proposta de blockchain (o Bitcoin) o protocolo de consenso procurava garantir que "uma CPU, um voto", e para isso foi proposto um mecanismo chamado **Proof-of-Work** (PoW). Essa técnica prescreve que cada nó performe um processamento matemático complexo (e portanto que demanda bastante CPU) na etapa de proposta do novo bloco. A maioria é representada pela blockchain mais longa que tem, portanto, o maior custo de processamento Proof-of-Work nela investido.

Para alterar um bloco passado, nesse formato, seria preciso refazer todo o trabalho de processamento Proof-of-Work daquele bloco e de todos os seguintes para conseguir passar o tamanho da blockchain já consensuada. Assim, enquanto a maioria do poder computacional estiver na mão de nós honestos, não é possível corromper a blockchain.[16]

Com o sucesso da tecnologia, o crescimento de participantes da rede e de transações financeiras, tornou-se cada vez mais demandante computacionalmente manter a blockchain com um

[15] *The latest gossip on BFT consensus*. Ethan Buchman, Jae Kwon e Zarko Milosevic. Disponível para consulta em https://arxiv.org/abs/1807.04938

[16] Bitcoin: A Peer-to-Peer Electronic Cash System, Satoshi Nakamoto - 4. Proof-of-Work.

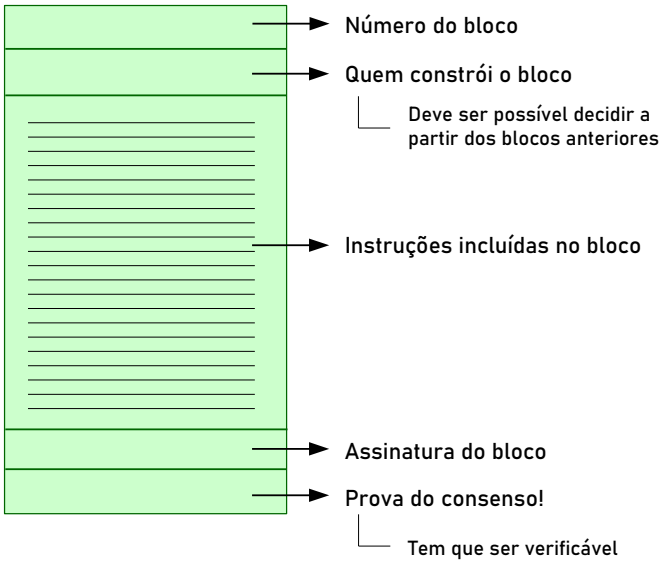

Figura 8: Representação de um bloco formado por mecanismo de consenso Proof of Stake (PoS).

mecanismo de consenso Proof-of-Work. A ineficiência computacional (e portanto energética) para esse processamento gera a demanda por um mecanismo de consenso alternativo.

Assim nasce a proposta de **Proof-of-Stake** (PoS). Nesse mecanismo, cada nó participante do consenso coloca como garantia de seu voto uma quantidade de tokens (que seria uma espécie de moeda ou ficha subjacente da blockchain). De todos os nós que colocam um valor de garantia para o seu voto, é formado um sub-grupo de onde virá o consenso.

O processo de escolha do comitê pode ser parcialmente aleatório, no sentido de que nós que colocam garantias maiores tem chances maiores de serem sorteados. Há protocolos em que existe um limite para a quantidade de tokens que pode ser colocada como garantia.

Cada blockchain possui suas próprias regras para definir os valores de garantia aceitos, as recompensas para os participantes do consenso e penalidades de mau comportamento. No formato Proof-of-Stake os nós que votaram com a maioria são recompensados, enquanto há penalidades para nós que não se comportam de forma honesta.

É possível garantir e penalizar como cada nó votou pois os votos são assinados.

O que realmente importa no mecanismo PoS usado é que cada um decide por si quem formou o bloco. Toda informação está no bloco, fica registrado quem foi responsável pela formação de cada bloco (conforme esquema na figura 8).

Quem verifica o bloco proposto assinou, pela regra dada pelo consenso é possível saber quem deve propor, quem deve validar e quantas validações são necessárias. Todo o resto é acidental.[17]

[17] O protocolo de consenso *Maré* proposto por @s livres está descrito em maior detalhe no anexo E, como item da descrição da rede *Brisa*.

E. Como funciona *Brisa*?

Uma breve esquematização do funcionamento do protocolo *Brisa*, que gerencia a rede cripto descentralizada, e seu respectivo procolo de consenso *Maré*.

Instruções processadas

A blockchain proposta processa 4 tipos de ações:

- depósito de tokens de garantia
- resgate de tokens garantia
- transferência de tokens
- void

As instruções referentes às movimentações de tokens são necessárias para o funcionamento do mecanismo de consenso Proof-of-Stake da rede cripto. No entanto, não foi programada escassez artificial de tokens e, ainda que eles gerenciem o espaço de publicação e o custo de infraestrutura física de processamento, a blockchain proposta não foi desenhada para armazenamento de ativos financeiros.

Para maximizar a capacidade de processamento da rede, o protocolo de consenso abre mão de um pouco de segurança. No caso de ativos financeiros isso seria imprudente, porém para interações sociais os riscos envolvidos são muito menores. Caso uma instrução social não seja processada, não há grandes perdas no reenvio, e em processá-la num bloco posterior.

Embarcando novos protocolos

Para que cada protocolo adjacente funcione em cima da rede cripto *Brisa*, é preciso que respeite uma estrutura simples incluindo:

- prescrição dos tipos de ações possíveis
- variáveis de estado
- alterações das variáveis de estado

Estado é como uma 'foto' dos valores guardados pelas variáveis em um dado momento.

Qualquer protocolo que possa ser definido em torno dessa estrutura pode ser processado a partir da instrução void de *Brisa*.

Maré: o protocolo de consenso

Maré é o protocolo de consenso da rede *Brisa*. É baseado em Tendermint[18], com modificações para acelerar a formação dos blocos.

[18] *The latest gossip on BFT consensus*. Ethan Buchman, Jae Kwon e Zarko Milosevic. https://arxiv.org/abs/1807.04938

De maneira bem simplificada, o protocolo prevê:

- 10 formadores de consenso por bloco, escolhidos entre 100 responsáveis pela formação de blocos, durante um período de 15 minutos
- a possibilidade de criação de blocos nulos, que é permitida e incentivada quando o consenso não é atingido rapidamente
- que o voto de cada participante possa ser apenas no hash recebido ou no bloco nulo, dessa forma não é tentada a verificação de outros hashs
- terminação forçada, desistindo de formar o consenso e seguindo em frente, caso não seja possível atingir maioria

Em todos os outros aspectos, o protocolo *Maré* segue a especificação do Tendermint.

A blockchain

É uma rede de percolação[19], com a seguinte dinâmica:

- O líder formador do bloco transmite o bloco proposto para os 10 formadores de consenso, que o transmitem então para os vizinhos, que o transmitem para seus vizinhos, e por assim vai até atingir os 100 participantes

  Isso implica que não há uma carga grande ao líder de precisar compartilhar o bloco com todos os 100 participantes da formação.

- Caso não seja possível atingir consenso, aquele bloco é perdido

  Nesse sentido está a flexibilidade de segurança da blockchain proposta. Para o processamento de transações financeiras, é bastante problemática a perda de um bloco. Para interações sociais, isso só quer dizer que algumas publicações de conteúdo podem não ter sido processadas e precisarão ser repetidas.

- Cada nó validador revalida o bloco consensuado, envia (faz um 'commit') por sua própria iniciativa e transmite esse envio a quem de fora do comitê estiver ouvindo (além dos 100 que já fazem parte da piscina de validadores)

  O nó transmite para quem estiver abaixo dele, que tipicamente serão os protocolos sociais e atores que estiverem oferecendo serviço de gateway (roteamento de instruções).

- Mau comportamento no protocolo *Maré* é punido

  Fazer essa verificação é simples, pois todo validador assina seu bloco.

- Os blocos são criados a partir de um checkpoint no passado, ou seja, são criados sem necessariamente conhecer o bloco anterior

[19] Um pouco sobre redes de percolação no anexo F.

- Uma validação de checkpoint acontece a cada 15 minutos

  Nessa validação, todos os nós participantes e candidatos calculam o checksum do estado naquele bloco.

- O checksum calculado é publicado junto dos blocos posteriores e, ao final, se há consenso de $50\% + 1$ dos validadores em relação ao checksum, um novo comitê é formado escolhendo aleatoriamente entre os nós que ofereceram o checksum correto

  O próprio valor do checksum é usado como semente de um hash para essa aleatoriedade na escolha do comitê.

- Se não for atingido o consenso de $50\% + 1$, os nós retornam ao último checksum consensuado e toda a janela é desprezada

  São 10 nós validando 1 bloco por segundo. São grandes as chances de haver blocos descartados por não atingir consenso, mas a capacidade de processamento é maximizada.[20]

F. O que é rede de percolação?

Há um campo de estudo na física estatística e na matemática chamado de teoria da percolação, que busca modelar o comportamento de uma rede quando novos nós ou conexões são adicionados.

No caso dos protocolos de consenso, a rede de percolação é a rede de nós validadores cujo comportamento é descrito segundo esses modelos probabilísticos.

Uma das características desses modelos é o chamado threshold (ou limite) da percolação, que descreve a probabilidade a partir da qual toda a rede descrita pelo modelo passa a estar infinitamente conectada.[21]

G. Benchmark de *Brisa*

Alguns dos resultados de teste de processamento do protocolo *Brisa*, apenas gerando a blockchain, ou seja, desconsiderando o custo de processamento do consenso.

- Processamento de cerca de $5k$ instruções de tamanho aproximado de $1Kb$, por segundo

  Uma estimativa de 400+MM instruções por dia.

- Para blocos de aproximadamente $10MB$ de tamanho, com custo de networking de consenso, é viável o processamento em infraestrutura média cerca de 10 nós/consenso (publicando formação do bloco numa rede de 100MB/s)

- 1 Bloco/Segundo e 10 nós/consenso: não é seguro. É uma otimização de performance em troca de certa segurança

[20] Informações sobre o benchmark da blockchain de *Brisa* no anexo G.

[21] https://mathworld.wolfram.com/PercolationThreshold.html.

- O consenso de Checkpoint envolve muito mais nós (100): a rede é muito mais segura nas janelas de consenso do que nos blocos do consenso

  Blocos não consensuados vão ser substituídos por blocos nulos. Mas checkpoints não consensuados vão ser retroagidos para o último checkpoint.

A conclusão é de que possível colocar de pé uma rede Proof-of-Stake "frágil" processando 400MM de instruções pequenas por dia numa conectividade de 100MB/s (R$ 1K/nó/mes) + custos de infraestrutura para distribuir os blocos para aplicações.

Não é a quantidade de interações de uma aplicação como o Instagram, mas é possível atingir a quantidade do Twitter (500MM).

Uma pergunta relevante é o que acontece se algo der errado dentro do processamento de um **void**. Basicamente a instrução da rede *Brisa* ainda é processada, com troca de tokens pelo processamento, porém o protocolo subjacente não vai aceitar.

Nesse caso tokens vão ser trocados a nível do *Brisa*, mas o protocolo social que recebe a informação do **void** não vai considerar a instrução.

Nenhum protocolo valida o que está dentro do seu void, isso sempre fica à cargo do protocolo subjacente, e qualquer erro no protocolo seguinte é transparente ao processamento da instrução do protocolo anterior.

H. Como funciona *Apelidos*?

Breve detalhamento do protocolo *Apelidos*.

Estrutura

O protocolo *Apelidos* vai funcionar a partir do void processado pela blockchain *Brisa*. Portanto, sua estrutura prevê: ações possíveis, variáveis de estado e alterações das variáveis de estado.

As **ações possíveis** aceitas por *Apelidos* são:

- Associar um apelido (a @ do usuário) a uma chave pública
- Conceder uma procuração, associando a chave pública da @ a uma chave pública de procurador
- Revogar uma procuração, removendo uma associação existente
- Void, com prova de autoria ou procuração válida

As **variáveis de estado** do protocolo:

- @s registradas e suas respectivas chaves associadas
- procurações ativas

As **mudanças de estado** do protocolo:

- novas concessões de procuração
- novas revogações de procuração
- novas @s associadas a chaves públicas

I. Como funciona *Motiró*?

O protocolo *Motiró* parte do void do protocolo *Apelidos*, que por sua vez tem sua instrução processada partir do void do protocolo da rede *Brisa*.

Assim como é o caso de *Apelidos*, a estrutura de *Motiró* prevê: ações possíveis, variáveis de estado e alterações das variáveis de estado.

As **ações possíveis** aceitas por *Motiró* são:

- Relativas a **coletivo**
  - Criar
  - Atualizar (exige consenso para ser processada)
  - Pedir para ingressar em um coletivo (exige consenso para aceite)
  - Remover um integrante de um coletivo (exige consenso para ser processada)
- Relativas a **esboço**
  - Publicar (pode ser realizada em nome de um indivíduo, grupo de indivíduos ou coletivo. Exige consenso para ser processada)
  - Lançar, ou seja, publicar que o esboço está em versão final (só pode ser realizada por um dos autores. Exige consenso para ser processada)
  - Carimbar, ou seja, endossar uma publicação que esteja em versão final (só pode ser realizada em nome de um coletivo. Exige consenso para ser processada)
  - Afixar/Desafixar em um mural (só pode ser realizada por um editor do mural. Exige consenso da edição do mural para ser processada)
  - Propor edição (deve fazer referência a um esboço publicado, e pode ser realizada em nome de um indivíduo, grupo de indivíduos ou coletivo. Exige consenso para ser processada)
- Relativas a **mural**
  - Criar (só pode ser realizada em nome de um coletivo. Exige consenso para ser processada)
  - Atualizar (só pode ser realizada por um editor de mural. Exige consenso da edição do mural para ser processada)

  - Incluir editor de um mural (só pode ser realizada por um editor do mural. Exige consenso da edição do mural para ser processada)
  - Remover editor de um mural (só pode ser realizada por um editor do mural. Exige consenso da edição do mural para ser processada)

- Relativas a **evento**
  - Criar (só pode ser realizada em nome de um coletivo. Exige consenso para ser processada)
  - Atualizar (só pode ser realizada por um gestor do evento. Exige consenso da gestão do evento para ser processada)
  - Cancelar (só pode ser realizada por um gestor do evento. Exige consenso da gestão do evento para ser processada)
  - Pedir para participar (aceite aceite de ao menos um gestor do evento)
  - Dar boas-vindas (só pode ser realizada por um gestor do evento, em resposta a um pedido de participação)

- **Gerais**
  - Subscrever-se ao protocolo
  - Reagir
  - Votar (em resposta a toda vez que um mecanismo de consenso de que o usuário fizer parte for acionado)

As **variáveis de estado** do protocolo:

- @s inscritas no protocolo e suas respectivas chaves públicas
- Esboços
- Edições
- Lançamentos
- Eventos
- Coletivos
- Murais
- Propostas

  Instruções pendentes que aguardam consenso para serem processadas (aceitas ou negadas em nome dos participantes do consenso)

- Index

  Mapa de usuários e suas respectivas conexões com os itens do protocolo

As **mudanças de estado** do protocolo incluem:

- itens (esboços, edições, eventos, etc) criados
- itens atualizados
- novas @s inscritas no protocolo
- novas conexões incluídas no index

A especificação do protocolo pode ser acessada em `https://freehandle.org/synergy/board/Protocolos+sociais`.