GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

UFRJ

UnB

UFBA

Correntes de Blocos: Algoritmos de Consenso e Implementação na Plataforma Hyperledger Fabric

Gabriel Antonio F. Rebello, Gustavo F. Camilo, Leonardo G. C. Silva, Lucas Airam C. de Souza, Lucas C. B. Guimarães, Eduardo A. P. Alchieri, Fabíola Greve e Otto Carlos M. B. Duarte

XXXIX Congresso da Sociedade Brasileira de Computação
Jornada de Atualização em Informática (JAI)

Agradecimentos a FAPERJ, CAPES, CNPq e Fapesp

Quais assuntos tratados neste JAI?

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- **Parte I - aborda e procura responder algumas perguntas conceituais tais como:**
  - O que é corrente de blocos?
  - Qual a dificuldade de se transferir moedas de um lado para o outro?
  - Qual foi a proposta inovadora e genial de Satoshi Nakamoto ao propor o Bitcoin?
  - Em que consiste o consenso por prova de trabalho
  - Por que o consenso de prova de trabalho é probabilístico?
  - Qual foi a inovação trazida pelo Ethereum em relação ao Bitcoin?
  - Por que corrente de blocos é considerada uma tecnologia disruptiva?

Quais assuntos tratados neste JAI?

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- **Parte II - aborda e procura responder algumas perguntas sobre CONSENSO tais como:**
  - O que é consenso? Consenso é importante em blockchain?
  - O que é protocolo tolerante a falha de parada e bizantina?
  - O que são os algoritmos de prova de trabalho (PoW), prova de posse (PoS), prova de posse delegada (DPoS),tolerante a falha bizantina (BFT), etc.?
  - O que é Ripple, Stellar, Bitshare, EOSIO, etc?
  - Qual é o melhor protocolo de consenso?
  - Existem mais de 1.000 criptomoedas. Quais as diferenças e como compará-las?

Qual a dificuldade de fazer uma aplicação distribuída usando uma das plataformas que existe?

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- **Parte III - apresenta um estudo de caso de sistema de segurança para virtualização de função de rede e uma proposta de solução**
- **Parte IV – apresenta a implementação da proposta de solução usando a plataforma Hyperledger Fabric**

Conclusões e perspectivas futuras

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- **Parte V – aborda as principais conclusões e as perspectivas futuras**
  - Os bancos vão deixar de existir em alguns anos?
  - Uber, Airb&b e toda a economia compartilhada vão desaparecer?
  - O que é a Internet de Valores?
  - O que seria a camada de confiança na nova Internet do Futuro?
  - Qual o futuro? Quais os desafios?

Autores e Apresentadores

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- **Gabriel Antonio Fontes Rebello**
  - Graduação - UFRJ
  - Engenheiro de Computação e Informação
  - Mestrado em andamento – COPPE/UFRJ
  - Atua em
    correntes de blocos (blockchain),
    segurança em redes de computadores,
    virtualização e inteligência computacional
  - **2019** Melhor artigo do II Workshop em Blockchain: Teoria, Tecnologias e Aplicações (WBlockchain - SBRC'2018
  - **2018** Melhor artigo do I Workshop em Blockchain: Teoria, Tecnologias e Aplicações (WBlockchain - SBRC'2018)

Autores e Apresentadores

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- **Prof. Eduardo Alchieri**
  - Departamento de Ciência da Computação
  - Universidade de Brasília (UnB)
  - Bacharel, mestrado e doutorado - UFSC
  - Atua principalmente tolerância a falhas e intrusões em sistemas distribuídos e de segurança computacional

Um especialista em consenso

Autores e Apresentadores

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- **Profa. Fabiola Greve**
  - GAUDI/DCC
  - Universidade Federal da Bahia
  - Graduação UFBA
  - Mestrado – Unicamp
  - Doutorado Université Rennes I e Laboratórios IRISA-INRIA
  - Pós-doutorado no LIP6, Université Pierre et Marie Curie (Paris-Sorbonnes Universités)
  - Atua principalmente nas áreas de algoritmos e sistemas distribuídos, tolerância a falhas e desenvolvimento de sistemas confiáveis.

Uma especialista em consenso

Autores e Apresentadores

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- **Gustavo F. Camilo**
- **Leonardo G. C. Silva**
- **Lucas Airam C. de Souza**
- **Lucas C. B. Guimarães**
  - Alunos de Iniciação Científica da UFRJ
  - Principal interesse
    - se formarem o mais rápido que puderem




Autores e Apresentadores

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- **Otto Carlos M. B. Duarte**
  - Graduação em Engenharia – UFRJ
  - Mestrado - COPPE/UFRJ
  - Doutorado - Ecole Nationale Supérieure des Télécommunications
  - Pós-doutorados
    - LIP6/UPMC, ICSI, University of California at Berkeley
  - Atua em tópicos da moda que permitam recrutar alunos muito bons e interessados que sejam capazes de submeter e ter artigos aceitos nas principais conferências realizadas nas melhores cidades do planeta

A Internet não transfere coisas físicas

Um arquivo digital pode ser copiado e enviada uma cópia do arquivo.
Dinheiro digital eu posso fazer cópia **mas isto é um problema** e deve ser evitado, pois qualquer pessoa poderia repetir (fabricar) moeda digital e usá-la diversas vezes.

Transferência de Ativos

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- Ativo: um recurso **único** que possui um **valor** e pertence a um **dono**

146.164.69.202

- Transferência física de ativos: enviar ou receber o ativo
  - Desvantagem → alcance ou tempo de transferência

O Gasto Duplo

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

Quando se transfere dinheiro de Aline para

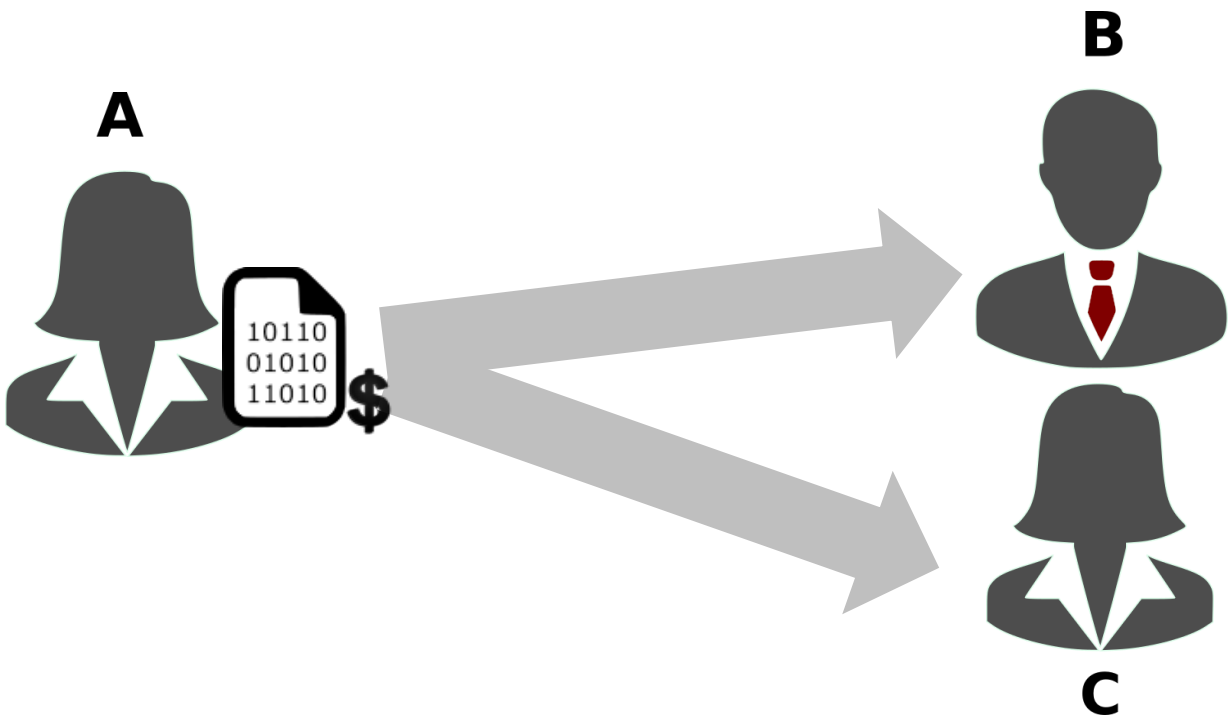
Transferência de moeda física de uma pessoa para outra é trivial, pois o dinheiro físico é único.

Transferência eletrônica de moeda digital é um problema complexo, pois a cópia é viável

O Problema do Gasto Duplo

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

Entidades pares não possuem **confiança mútua**



Solução atual - INTERMEDIAÇÃO

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- INTERMEDIAÇÃO

Um intermediário provê a CONFIANÇA ENTRE AS PARTES

NO ENTANTO: um intermediário provoca
Custo: taxas
Atrasos: longo tempo p/ executar a transferência

Solução Tradicional (Intermediação)

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- Transferência de ativos online → **intermediários**

Como transferir ativos online **sem um intermediário**?

  - Ponto único de falha

Resolvendo o Problema do Gasto Duplo

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- Para resolver o problema do gasto duplo de maneira **descentralizada**, é necessário garantir que:

A corrente de blocos **reúne** conceitos **simples** para resolver o problema do gasto duplo

Transferência de Moedas

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- Transferência de moedas (ativos) digitais é um problema estudado há décadas
- Novembro de 2018, em uma lista de e-mails de criptografia, Satoshi Nakamoto envia um email com a solução do problema do gasto duplo usando técnicas criptográficas simples e conhecidas

Bitcoin: A Peer-to-Peer Electronic Cash System

.

Bitcoin: A Peer-to-Peer Electr

TA / UFRJ
O DE TELEINFORMÁTICA E AUTOMAÇÃO

A

a

A

A t

em

inte

sso

A pro

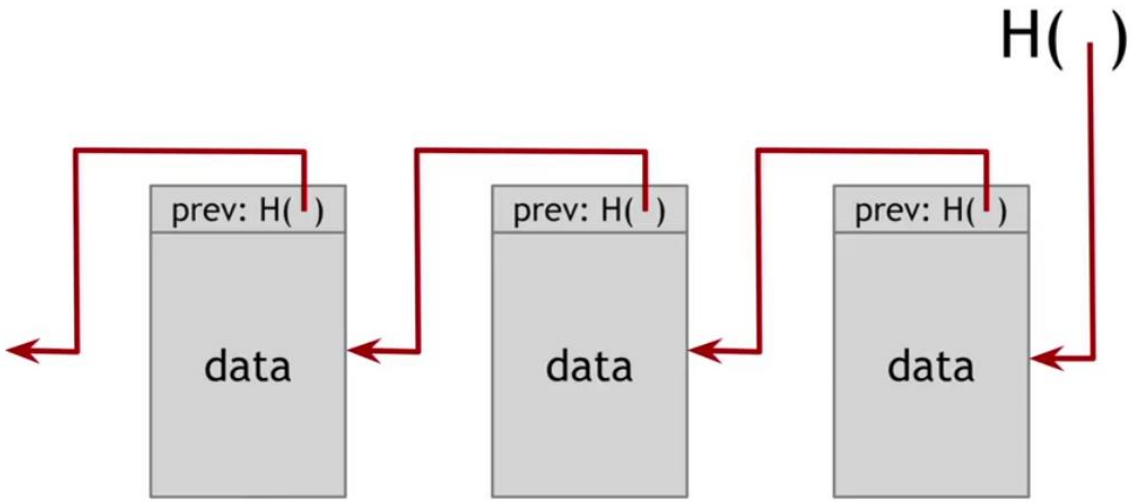
iários

tecnologias simples já conhecidas

A ideia mais básica da corrente de blocos

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- Estrutura de dados de um livro de registros (*ledger*) usando funções resumo (*hash*) criptográficas que a torna imutável



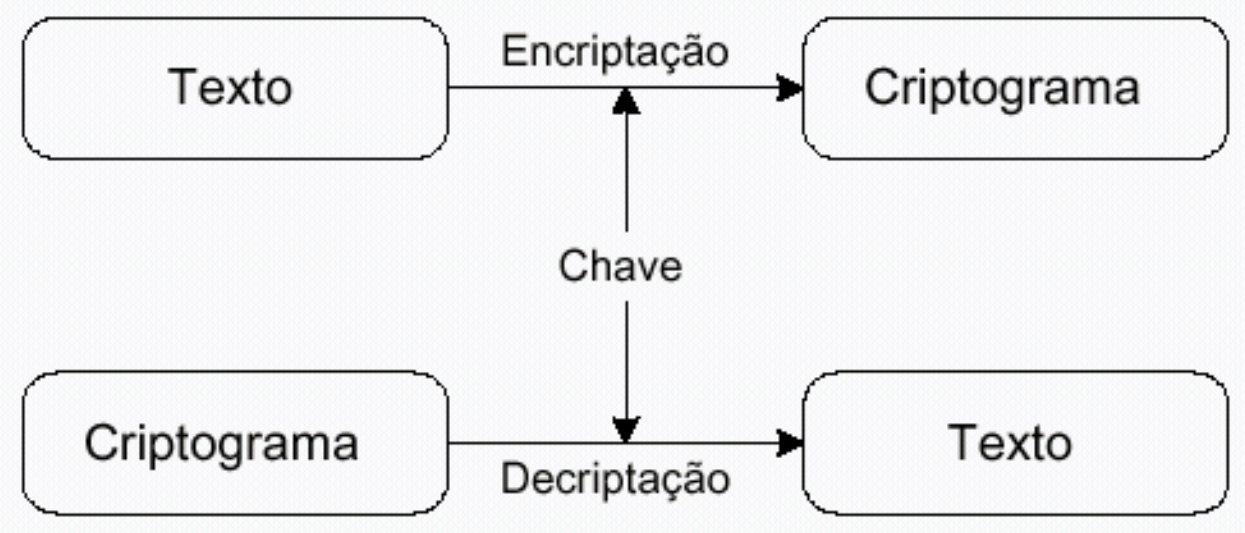
GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

Revisão de Fundamentos de Criptografia

Tipos de Criptografia

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- Simétrica ou de chave secreta



PAR de chaves

Uma chave pública e outra chave privada

Pr **A)**

Procedimento para assinar uma mensagem e verificar uma assinatura

Ana assina uma mensagem com a sua chave privada. A autenticação ocorre porque apenas Ana conhece a chave privada dela.

--

Qualquer pessoa que possua a chave pública da Ana pode verificar a autenticidade de uma mensagem ao decriptar a mensagem com a chave pública da Ana

pública de Ana (A) e a transmite para Ana (A)

Segurança criptográfica

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- Não existe algoritmo criptográfico incondicionalmente seguro
- Busca-se algoritmos computacionalmente seguros
  - Custo em "quebrar" deve ser superior ao valor da informação
  - Tempo necessário para "quebrar" deve superar a "vida útil da informação"

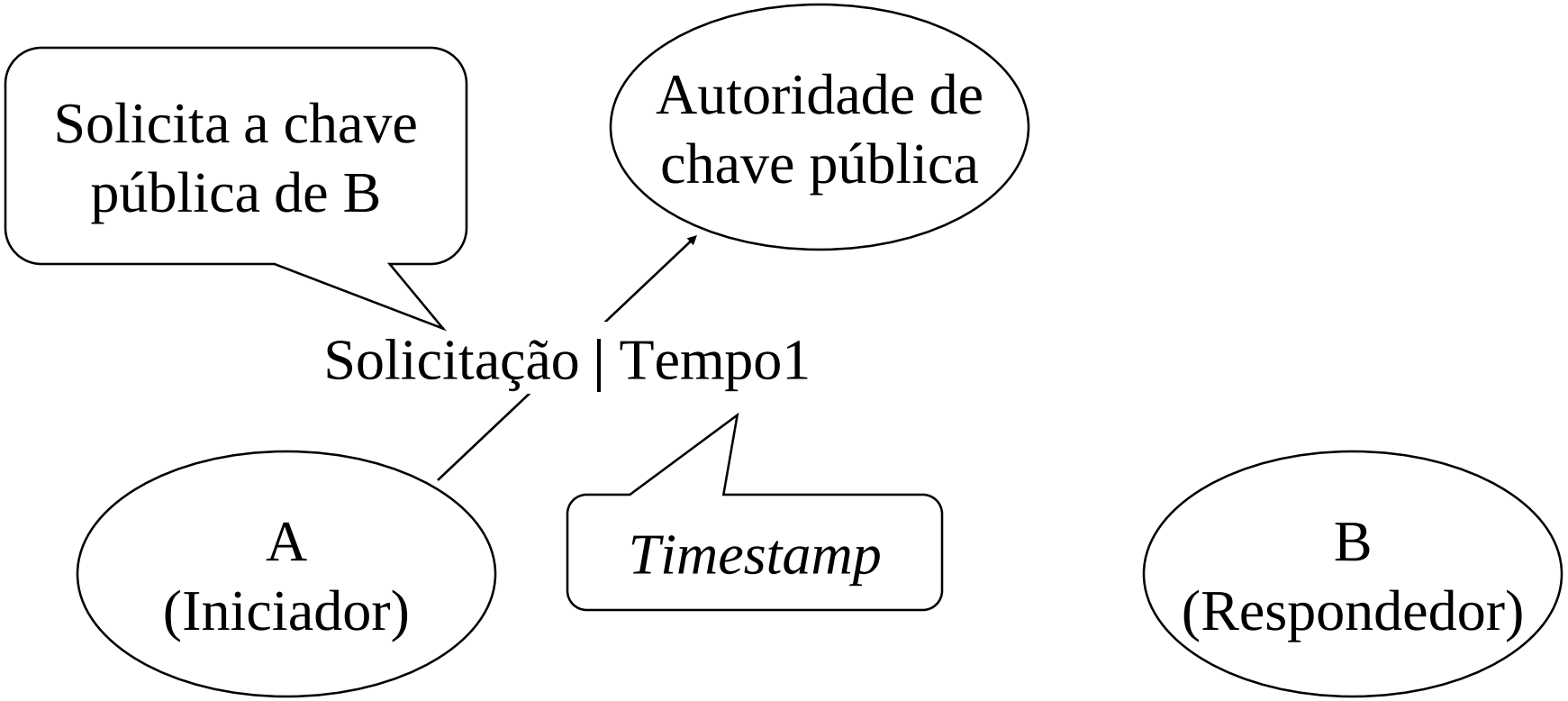
Como obter a chave pública?

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

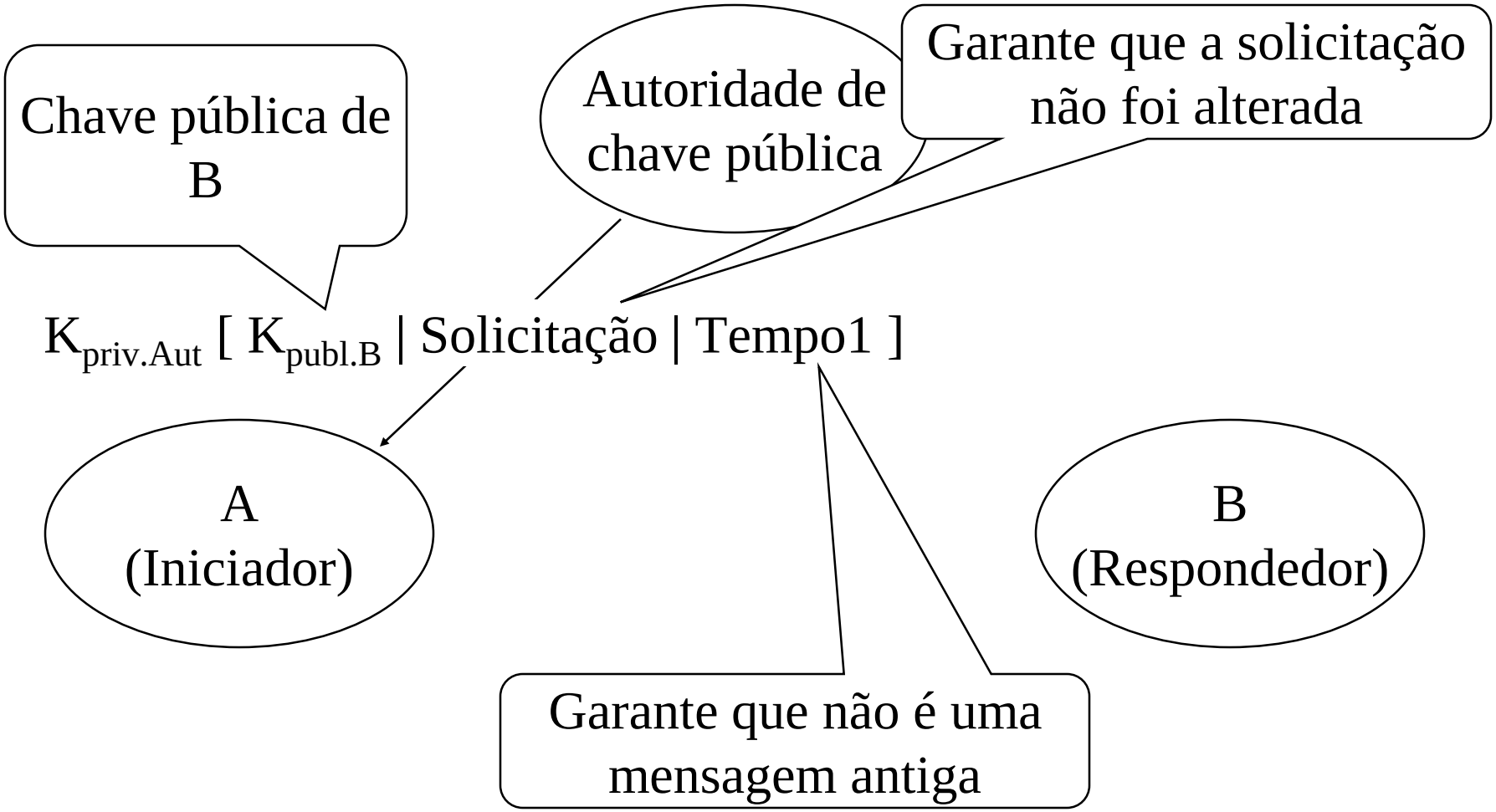
- Envio por email? Inseguro
- Colocar em uma página web? Inseguro
- Uso de autoridades certificadoras

Autoridade de chave pública

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

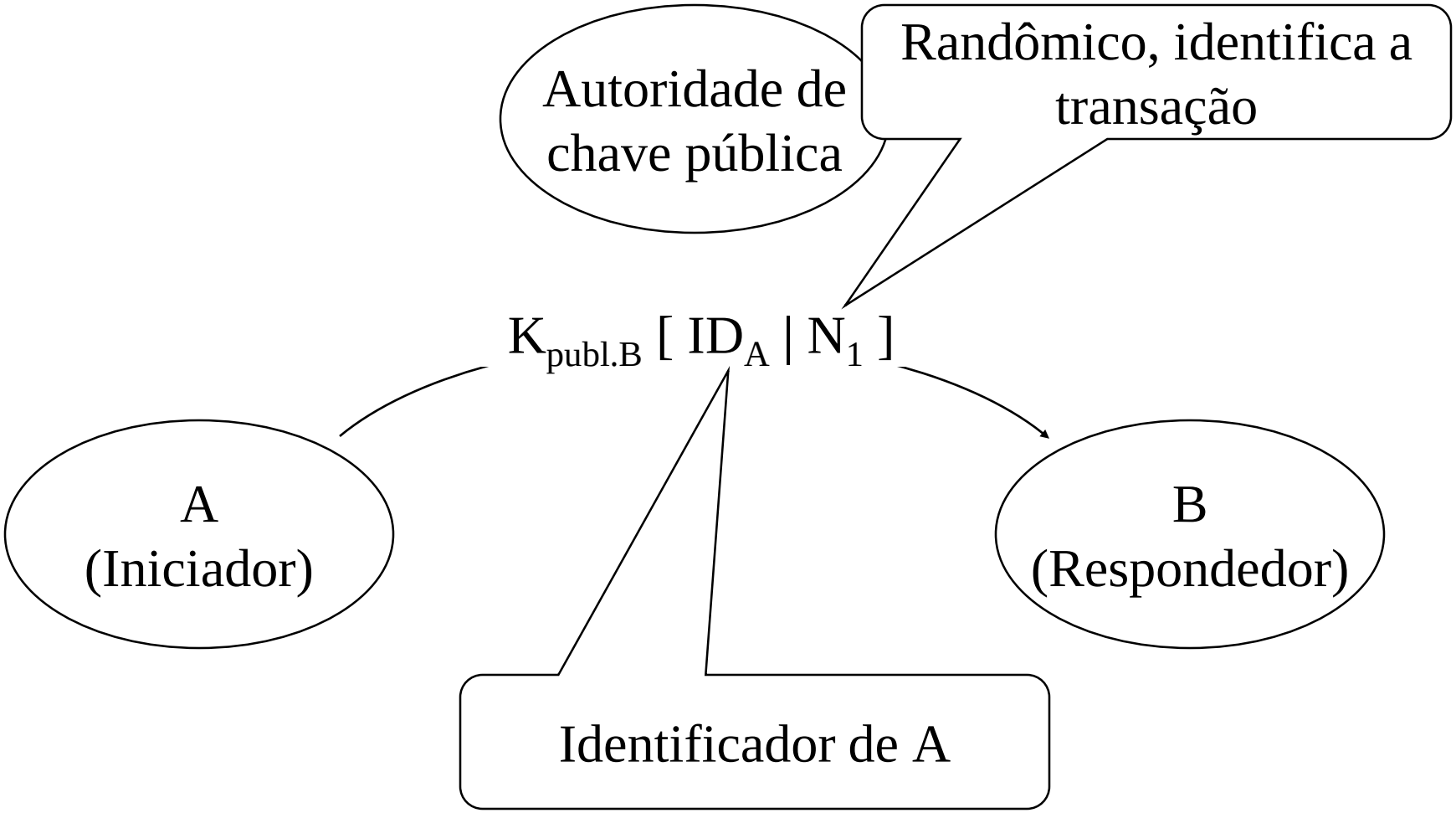


Autoridade de chave pública

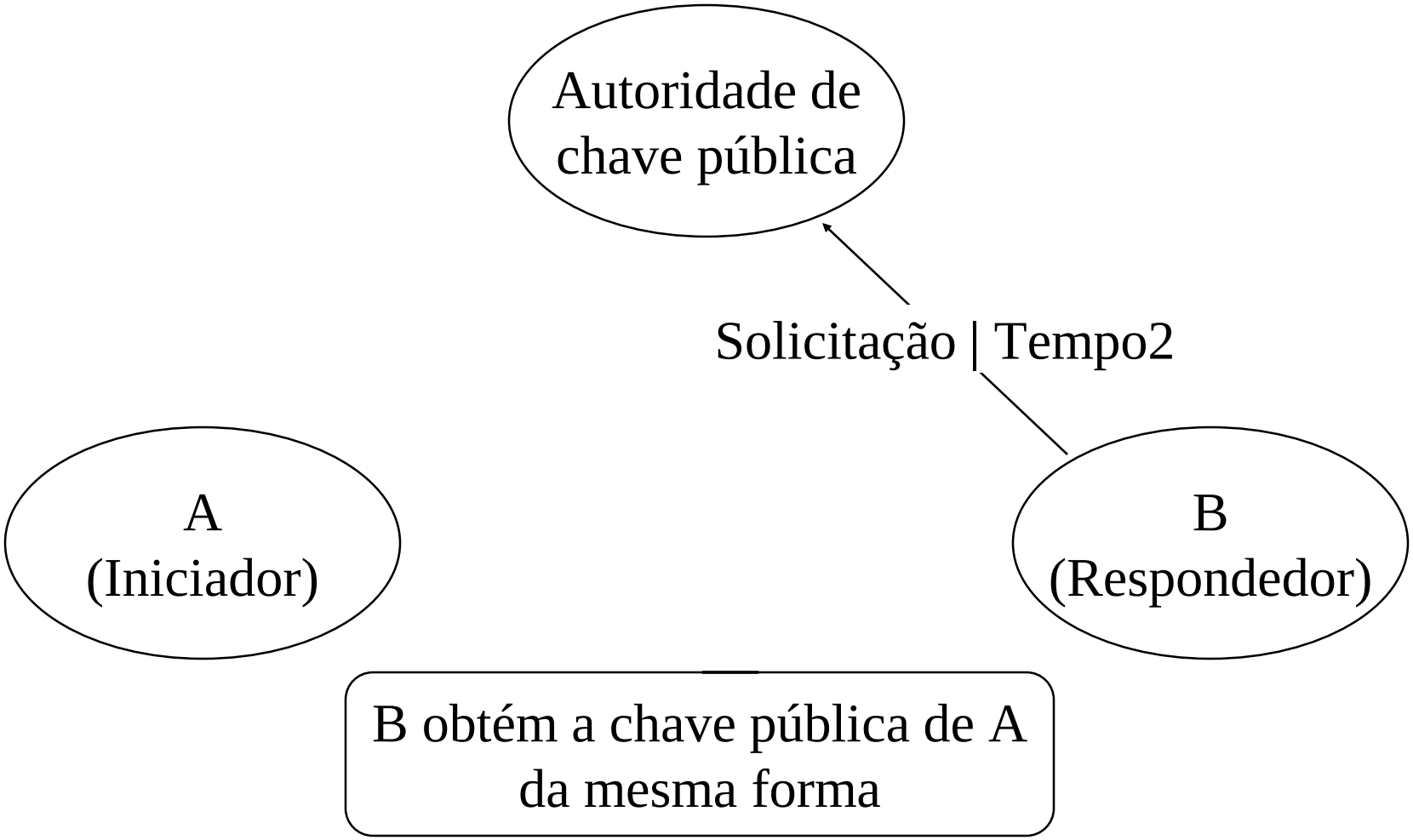


Autoridade de chave pública

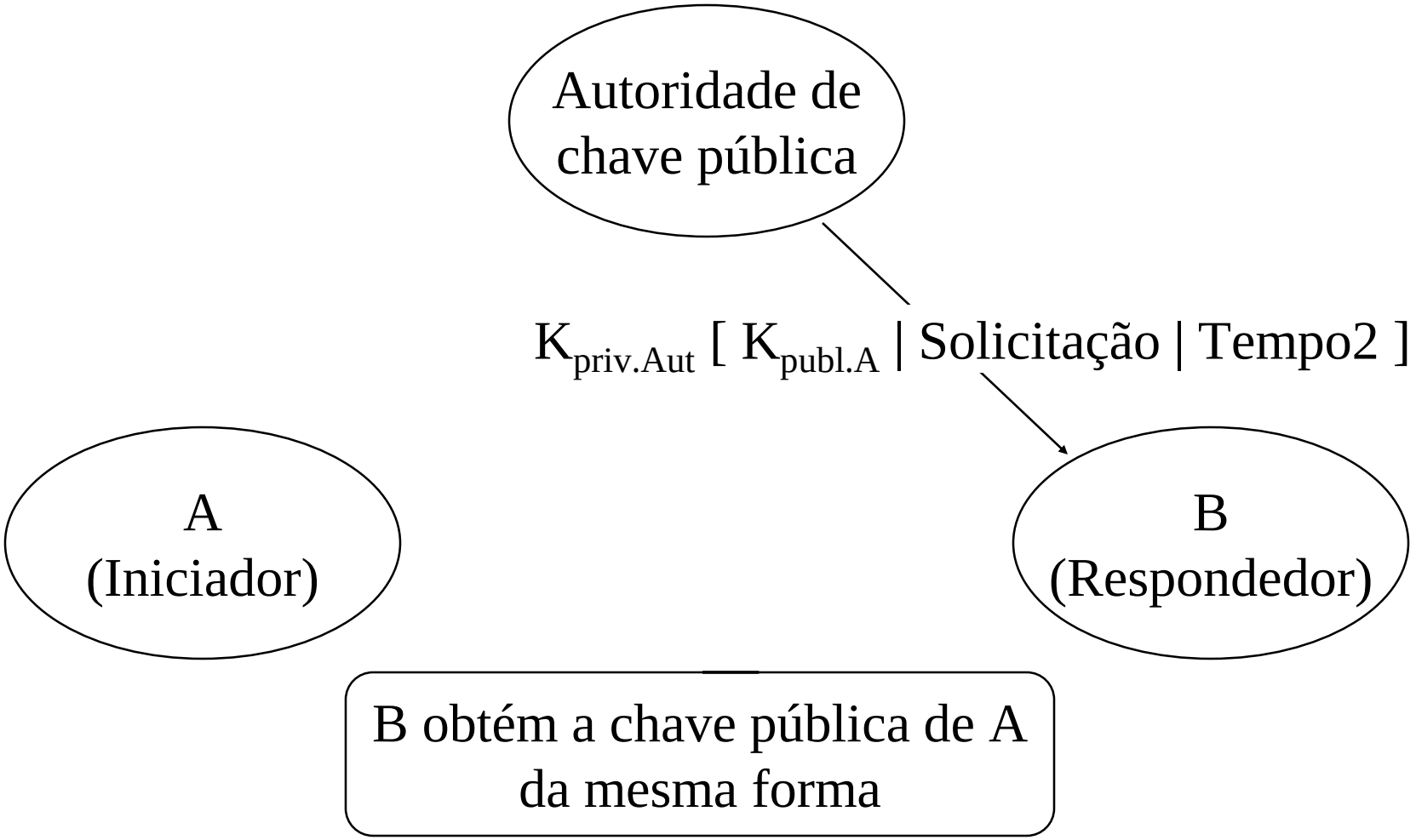
GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO



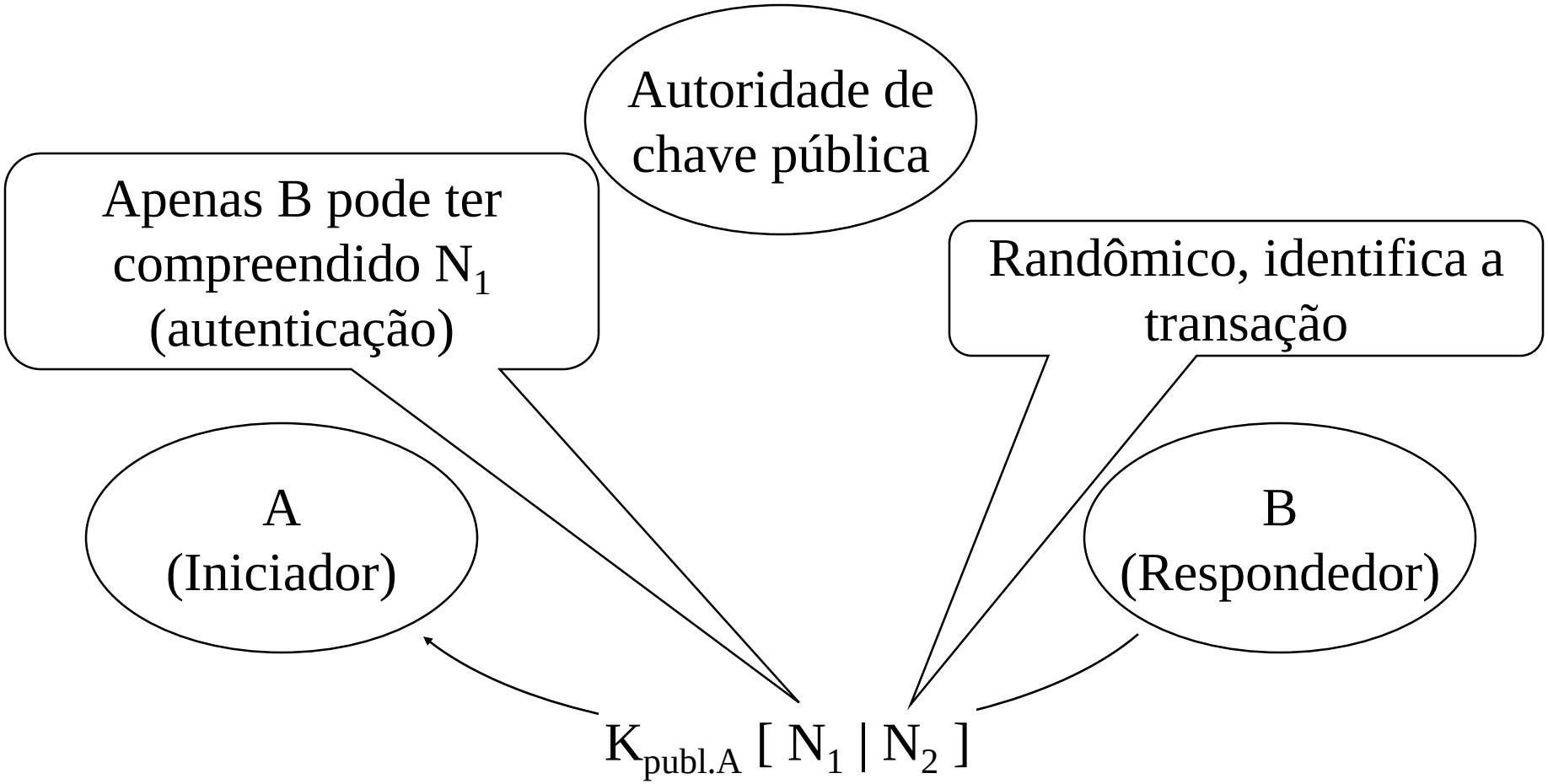
Autoridade de chave pública



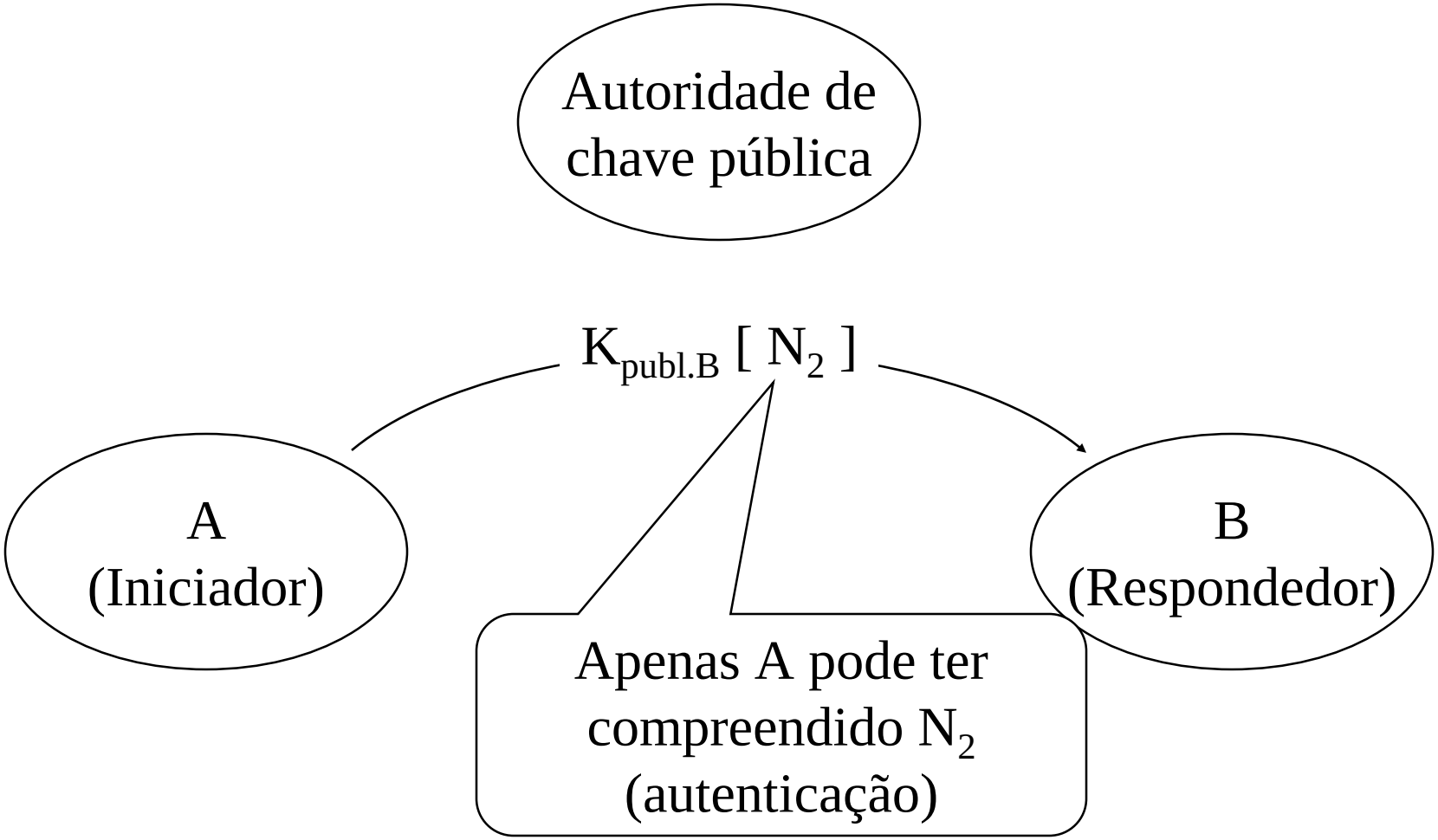
Autoridade de chave pública



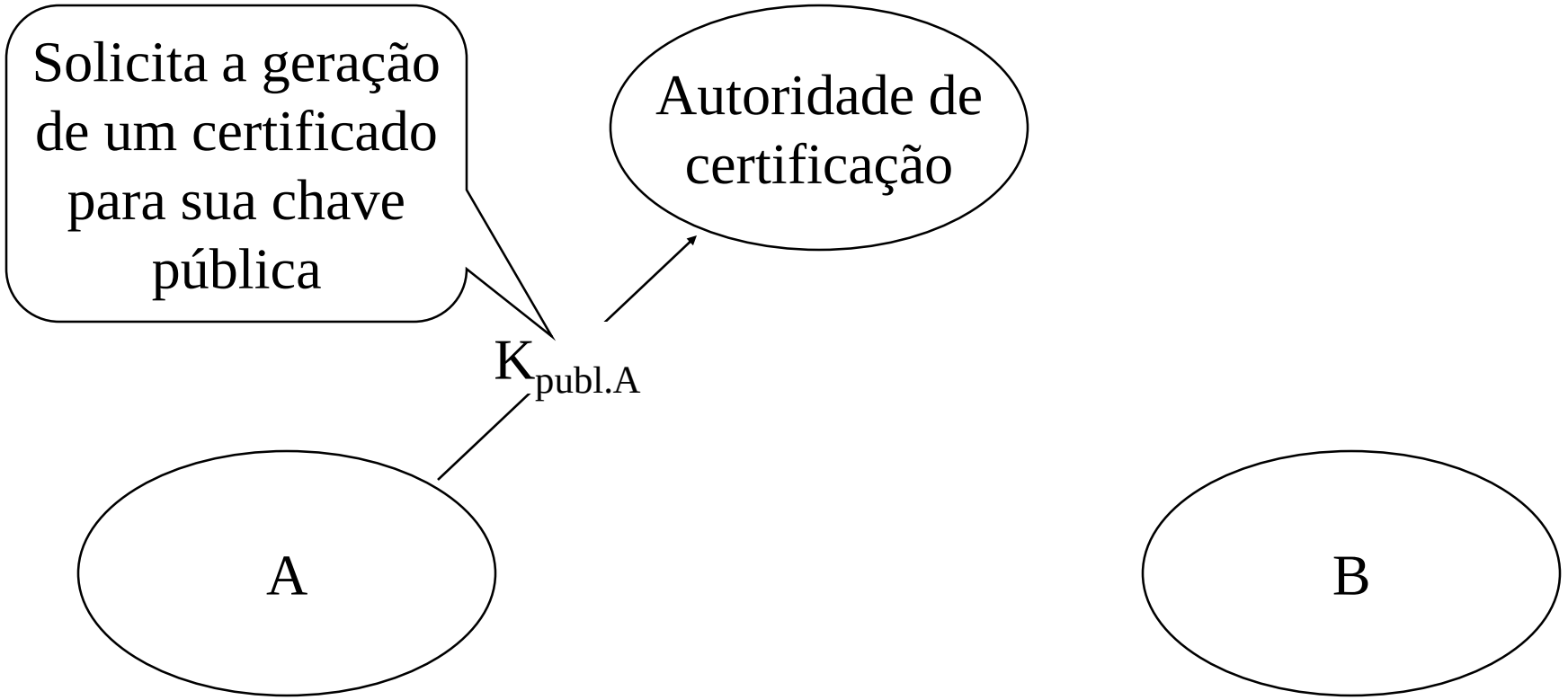
Autoridade de chave pública



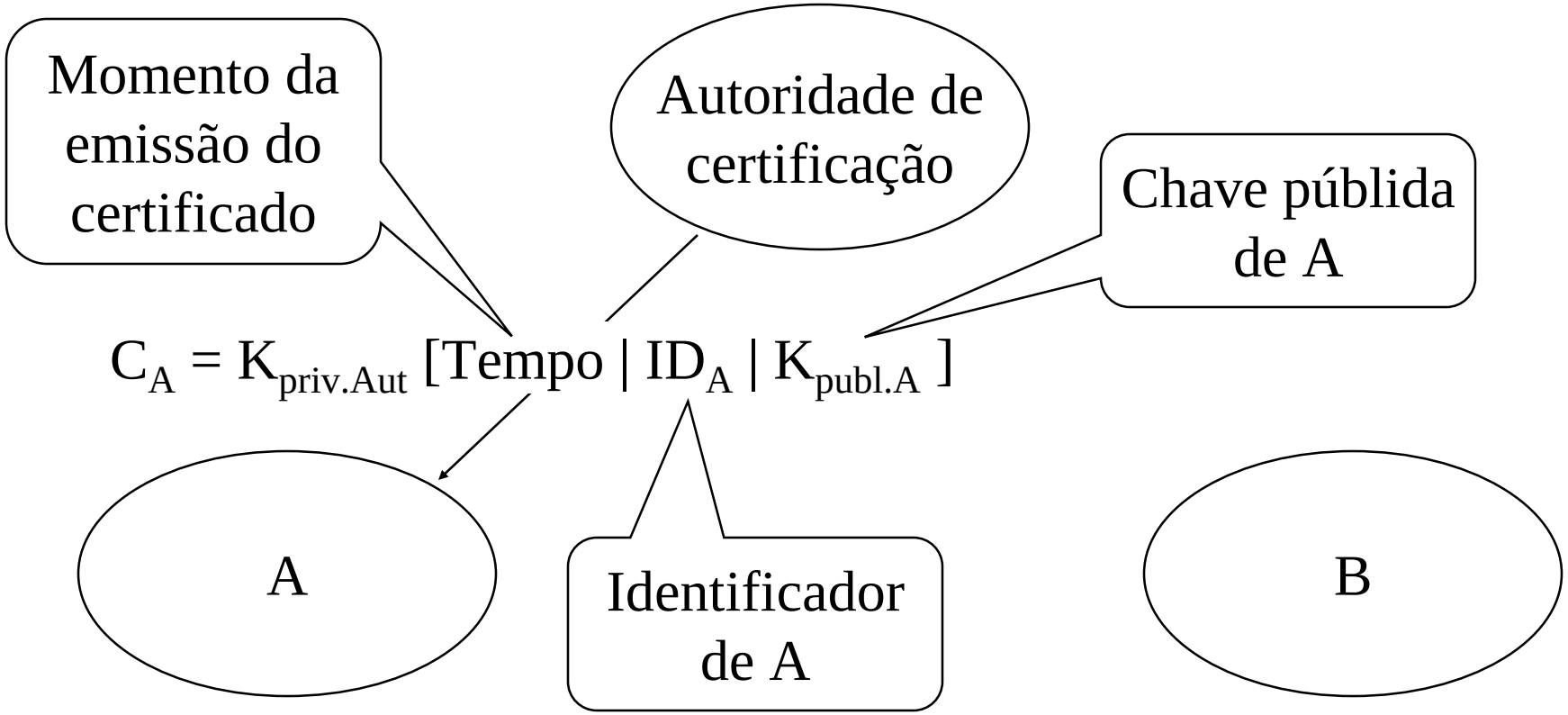
Autoridade de chave pública



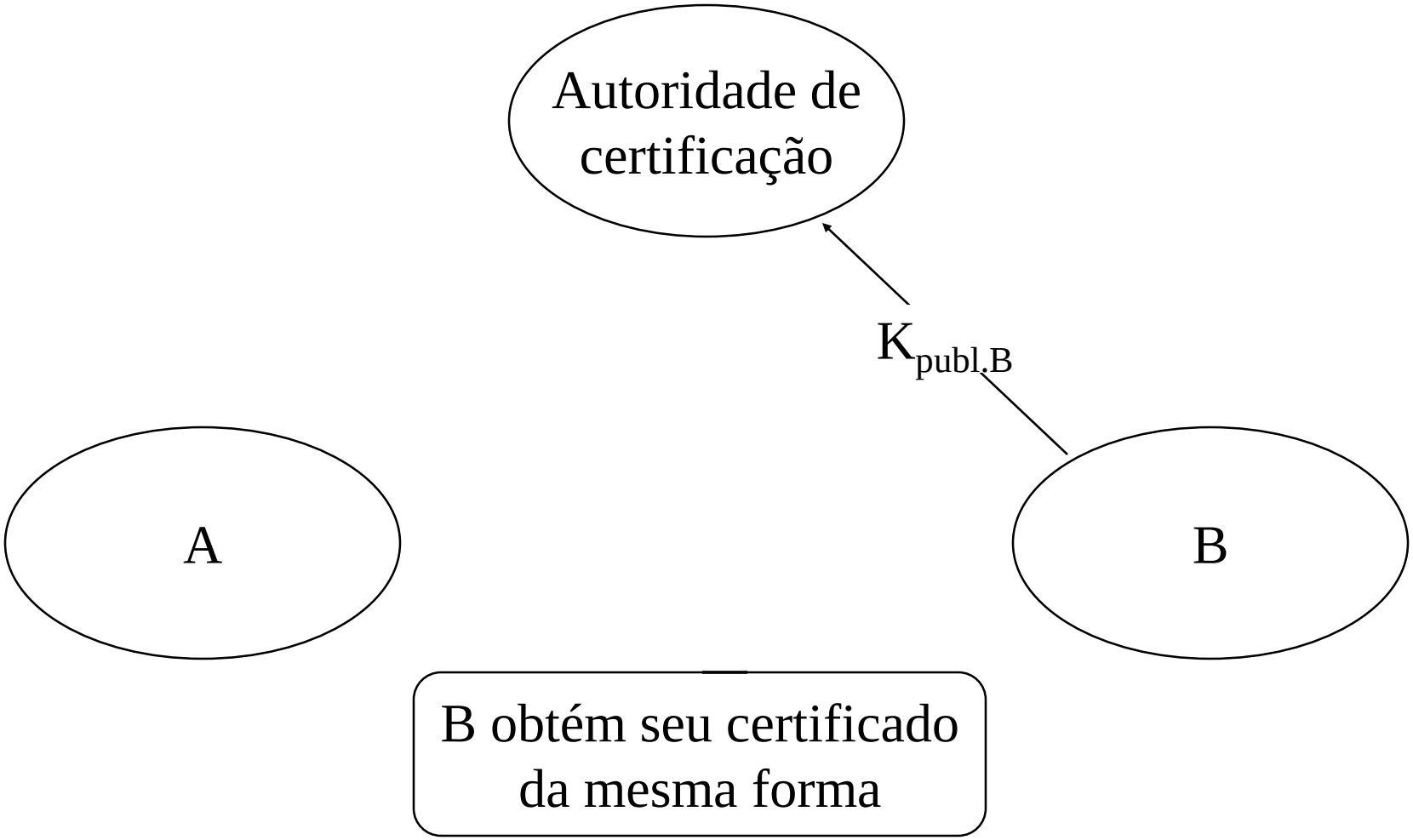
Certificados de chave pública



Certificados de chave pública

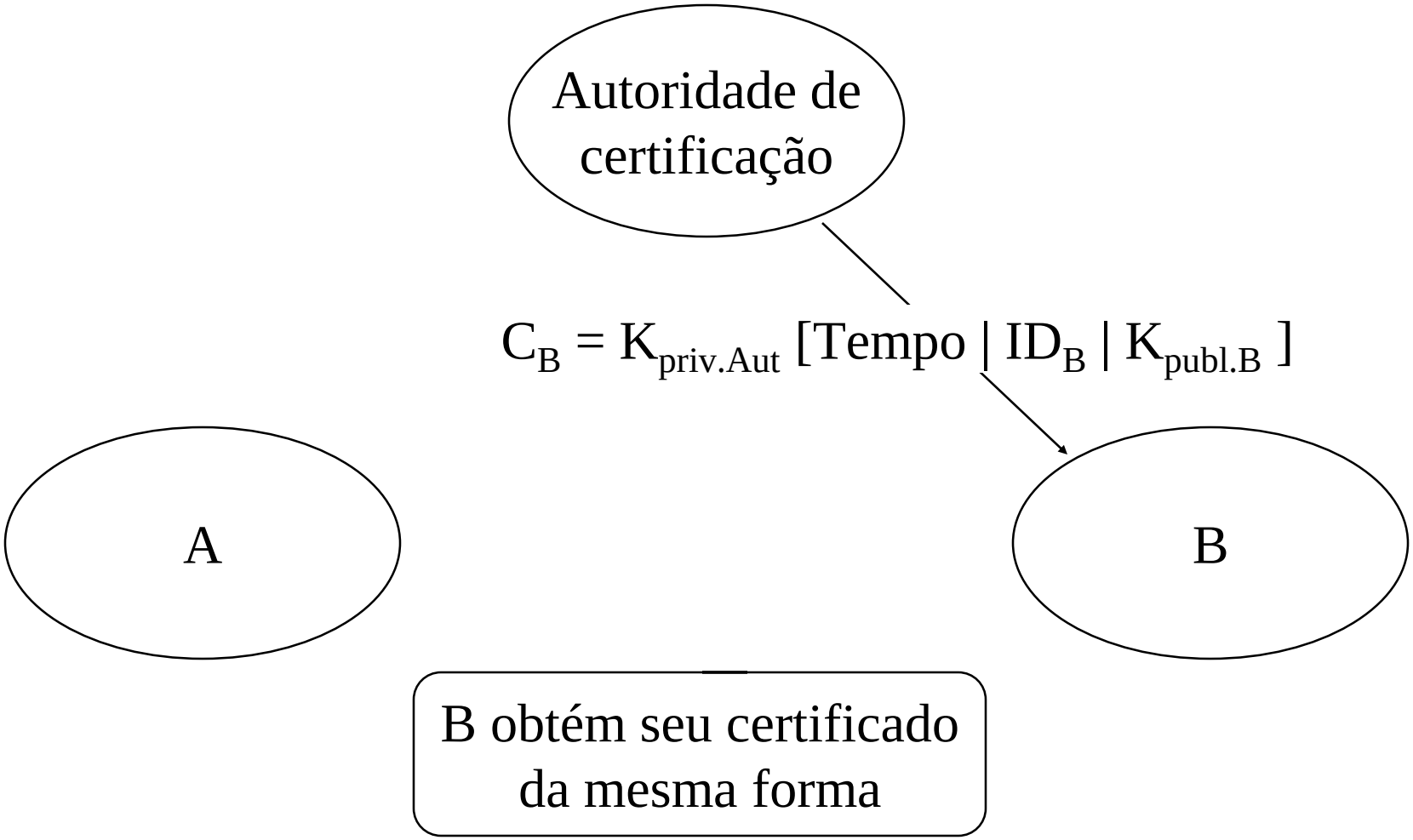


Certificados de chave pública



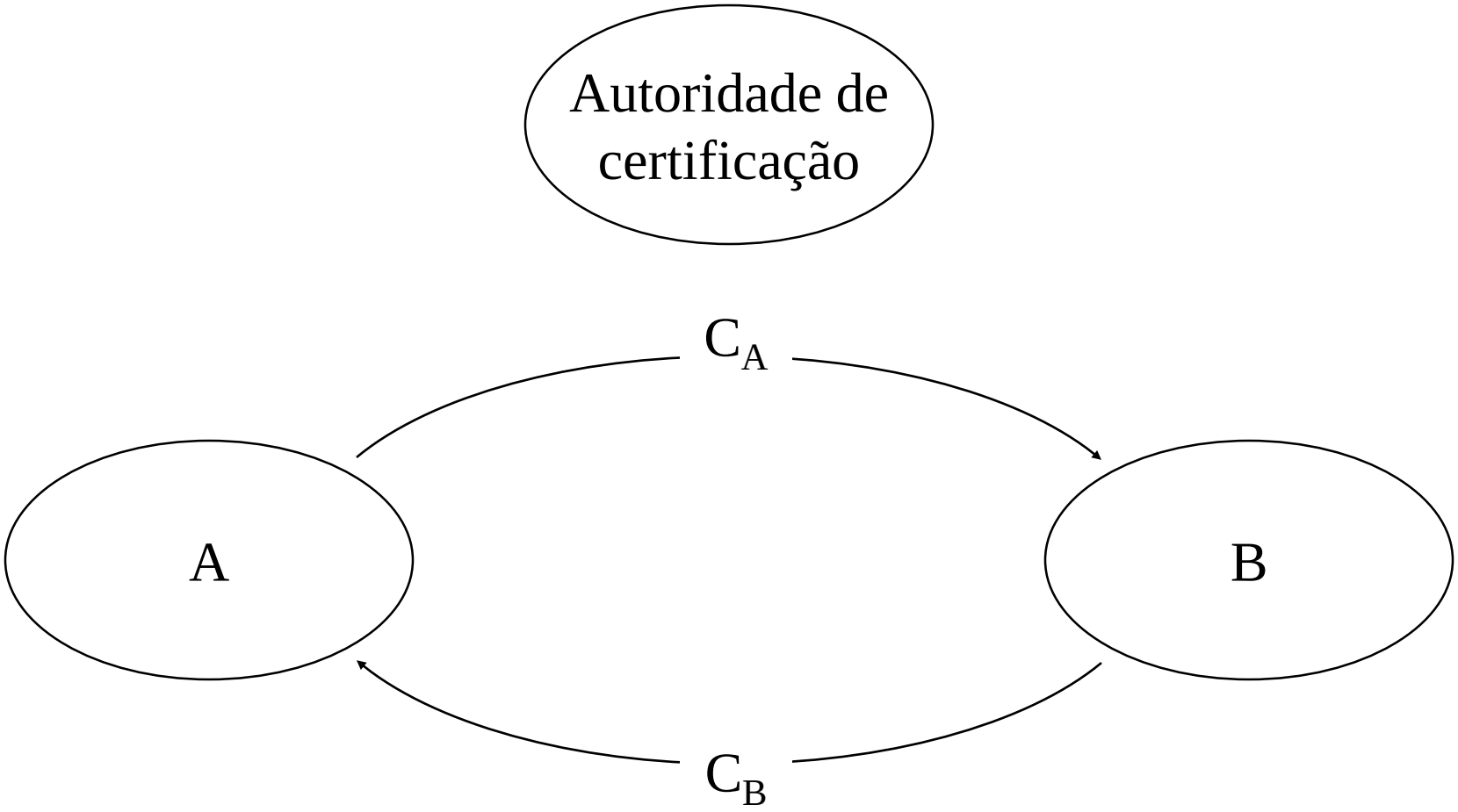
Certificados de chave pública

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO



Certificados de chave pública

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO



Certificados de chave pública

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- Verificação:
  - $D_{Kpubl.Aut}$ (C_A) =
  - $D_{Kpubl.Aut}$ ($K_{priv.Aut}$ [Tempo | ID_A | $K_{publ.A}$]) =
  - [Tempo | ID_A | $K_{publ.A}$]

Caso o receptor tenha conhecimento de um certificado mais recente, este certificado é considerado inválido

RSA

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- *Rivest-Shamir-Adleman* (1977)
- Chaves de tamanho variável
- Dado com tamanho menor que o da chave
- Criptograma com o tamanho da chave

RSA

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- escolher p e q números primos grandes
  - então $n=pq$ e $\varphi(n) = (p-1)q-1)$
- escolher d (decripção) e e (encriptação) tal que
  - $de = 1 \bmod \varphi(n)$
  - $d \rightarrow$ inverso multiplicativo de '$e \bmod \varphi(n)$'
- Assim, para qq x
  - $x^{de} = x \bmod n$
- $<d,n>$ é a chave privada
- $<e,n>$ é a chave pública

RSA - Criptografando

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- dada a mensagem m onde $m<n$
- encriptação
  - $c = m^e \bmod n$

RSA - Decriptografando

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- dada a mensagem m onde $m<n$
- encriptação
  - $c = m^e \bmod n$
- *decriptação*
  - $c^d = (m^e \bmod n)^d = m^{ed} \bmod n =$
    $= m^{(1 \bmod \varphi(n))} \bmod n = m$

RSA - Assinando

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- dada a mensagem m onde $m<n$
- assinatura
  - $s = m^d \bmod n$

RSA – Verificando Assinatura

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- dada a mensagem m onde $m<n$
- assinatura
  - $s = m^d \bmod n$
- *verificação da assinatura*
  - $s^e = (m^d \bmod n)^e = m^{ed} \bmod n =$
  $= m^{(1 \bmod \varphi(n))} \bmod n = m$

Curvas Elípticas

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- Segurança equivalente ao RSA com chaves bem menores
- Segurança provida por uma chave RSA de 1024 bits pode ser obtida com uma chave de curva elíptica de 160 bits, uma de 2048 bits RCA equivale a uma chave de curva elíptica de 244
- 1024 ⇨ 160
- 2048 ⇨ 244

Função Resumo (*Hash Function*)

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- Função unívoca
  - Uma entrada gera uma saída
  - Praticamente impossível de descobrir a entrada a partir da saída
- Entrada de tamanho arbitrário gera uma saída de tamanho fixo
- Saída de tamanho fixo e com número de 1's mais ou menos igual ao de 0's
- Saídas completamente descorrelacionadas
  - Duas entradas que diferem de 1 bit geram saídas completamente diferentes

Como pode isso?

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

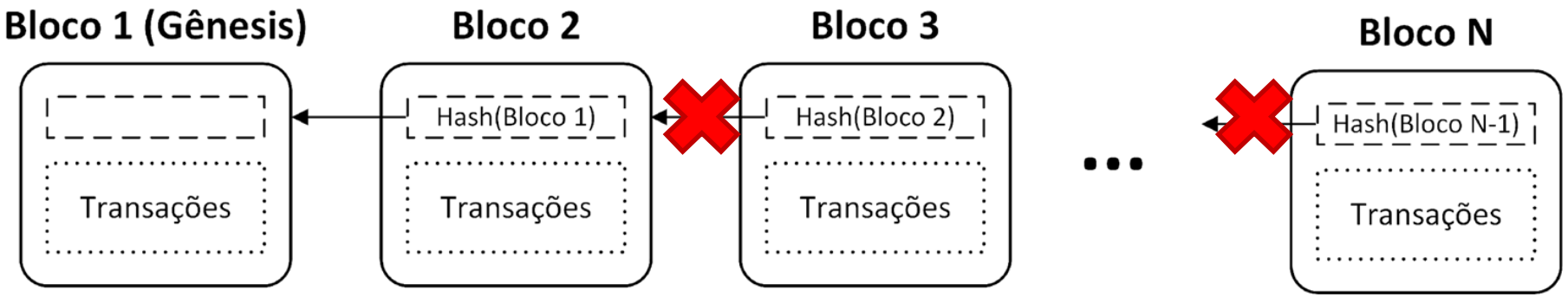
- Mensagens de 1000 bits gerando resumos de 128 bits
- Assim, há, em média, 2^{872} mensagens que são mapeadas em cada resumo de 128 bits
- É possível achar duas (ou mais) entradas que são mapeadas no mesmo resumo
- Mas, baseado na aleatoriedade da função resumo, é necessário testar um número muito grande de entradas para achar um mesmo resumo
  - Praticamente impossível

Estrutura de Dados de uma Corrente de Blocos

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- Livro-razão **distribuído** e **público** organizado em **blocos**
  - Cada bloco contém transações e um *hash* do bloco anterior
  - Cada participante possui uma réplica da corrente de blocos

Alterações em transações passadas quebram a sequência de funções resumo



Propriedades das Correntes de Blocos

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- imutável
- inviolável
- disponível
- traçável
- autenticável
- anônima
- confidencial

Revisão

O que é corrente de blocos

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

Corrente de blocos é uma tecnologia disruptiva baseada em **criptografia, comunicação fim-a-fim, replicação de máquina de estados, consenso e teoria dos jogos** que permite prover uma **camada de confiança** e, com isto, permitir a **transferência segura de ativos**

A corrente de blocos em si é uma estrutura de dados encadeada e **imutável** pelo uso de funções *hash* criptográficas. A corrente de blocos só cresce.

Para que um bloco seja acrescido na corrente de blocos tem existir **consenso** entre os participantes

A corrente de blocos é **disponível** em todos os nós participantes.

Qual a dificuldade de se transferir moedas de um lado para o outro?

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

O maior problema é transferir ativos (moedas, valores, etc.) entre pessoas que **não possuem confiança mútua**

A autenticação criptográfica garante o conhecimento da origem e do destino, mas não garante a validade da operação

Por que eu não posso transferir o que eu não tenho? Que tal verificar através de prova o que a origem possui? Que tal esperar que o destino comprove que recebeu?

Esperar a realização da transferência não garante a validade pois, neste meio tempo, a origem pode gastar o mesmo recurso várias vezes

Qual foi a proposta inovadora genial de Satoshi Nakamoto ao propor o Bitcoin?

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

Resolveu o problema do gasto duplo sem intermediários com tecnologias simples já existentes

O consenso é estudado há mais de 30 anos e sempre foi um enorme desafio. Com o Bitcoin, Satoshi Nakamoto resolveu o problema do consenso de forma probabilística propondo o consenso de Prova de Trabalho (Proof of Work – PoW)

Em que consiste o consenso por prova de trabalho

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

Satoshi Nakamoto propõe o consenso probabilístico através da prova de trabalho que consiste em **resolver um desafio** para que um novo bloco seja acrescido na corrente de blocos

A solução do desafio requer o uso de muito processamento e com isto um gasto energético muito grande

Para compensar o custo do gasto energético os ganhadores dos desafios são **incentivados** com uma **remuneração**

Por que o consenso PoW é probabilístico?

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

Por que pode acontecer bifurcações (*fork*)

Qual foi a inovação trazida pelo Ethereum em relação ao Bitcoin?

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

O Ethereum é uma evolução do Bitcoin que provê uma máquina de Turing, permitindo executar os contratos inteligentes

O contrato inteligente é um programa autoexecutável (sem a interferência humana) que envolve a transferência de moeda virtual entre duas ou mais pessoas. Os contratos são programas executados em computadores que definem regras e consequências estritas, declarando as obrigações, os benefícios e as penalidades das partes. Diferentemente de um contrato tradicional, um contrato inteligente é capaz de obter informações, processá-las e tomar as devidas ações previstas de acordo com as regras do contrato.

Por que corrente de blocos é considerada uma tecnologia disruptiva?

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

Por prover confiança entre entidades que não confiam entre si sem requerer intermediários

Tem aplicações em quase todas as áreas economia, saúde, transporte etc.

Vai permitir a Internet de Valor

Vai revolucionar o nosso modo de viver

Corrente de blocos

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

Transferência de valores sem taxas

Transferência imediata de valores

Eliminação de intermediários

FIM dos bancos e instituições financeiras

Corrente de Blocos e a Internet

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- Blockchain proverá a camada de confiança distribuída
- A Internet do Futuro será a blockchain
- A Internet atual de transferência de mensagens e arquivo será substituída pela Internet do Futuro com *blockchain* que será a **Internet de Valores**

Corrente de blocos e a economia compartilhada

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- Eliminação de intermediários

Corrente de blocos e a economia compartilhada

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- Eliminação de intermediários

tinder

Soluções de corrente de blocos

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

Moeda digital

- comércio eletrônico
- pagamentos globais
- transferências monetárias
- Microfinanças

Valores mobiliários

- **títulos, participações**
- **financiamento coletivo**
- **Dívidas**
- **mercado privado**

Contrato inteligente

- direitos autorais
- jogos e apostas
- depósito, caução

Arquivamento de registros

- propriedade intelectual
- propriedade, posse
- títulos, escrituras
- assistência médica
- votação
- controle de proveniência

A tecnologia mais significante depois da Internet
Provê a camada de CONFIANÇA da nova Internet

Blockchain: uma tecnologia disruptiva

O nascer de uma nova era

- Blockchain é o futuro
- Blockchain proverá confiança distribuída
- Blockchain eliminará intermediários
- Blockchain acabará com os Bancos
- Blockchain eliminará os governos
  - Blockchain acabará com a fome
  - Blockchain acabará com a miséria
  - Blockchain acabará com as doenças
  - Blockchain acabará com a dor de dente
  - Blockchain acabará com a TPM

Moedas Tradicionais

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- Dinheiro Físico: moeda e papel-moeda

Moedas Digitais

  - Não-rastreamento
  - Não-répúdio
- Desvantagens
  - Alcance e desempenho limitados
  - Não podem ser usadas diretamente em sistemas eletrônicos em linha (*online*)

Transferências Financeiras Convencionais

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- Ana, no Brasil, para transferir um valor para Beto, na

Transações digitais envolvem **confiança** em uma instituição INTERMEDIÁRIA

Moedas Eletrônicas

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- Motivação
  - Desconfiança nas instituições financeiras
  - Custos de transações (e.g. DOC, TED)
- Principais desafios
  - Segurança da transação
    - Legitimidade do emissor
    - Legitimidade das moedas
  - Problema do **gasto duplo**

Bitcoin (BTC)

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- Satochi Nakamoto propôs em uma lista de e-mails de criptografia em 2008, implemetada em janeiro de 2009
  - Satoshi Nakamoto é um pseudônimo
- É a mais antiga e a mais valiosa criptomoeda
  - 1 BTC = R$40978,85 (15/07/2019 – varia em média 10% ao dia)
  - Já valeu R$64.000,00
- Segura - Nunca sofreu um ataque
- Hash de 256 bits – SHA256
- Criptografia de curva elíptica – SECP 256 k1

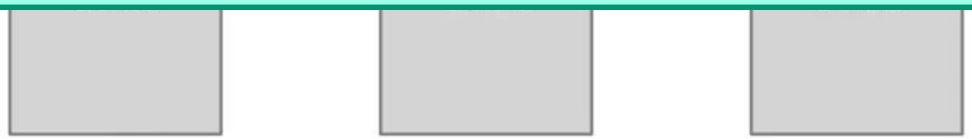
Corrente de Blocos - *Blockchain*

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- registro ordenado contendo todas as transações da rede

no

Problema do gasto duplo é resolvido se a corrente é uma estrutura única (sem bifurcações), imutável e se os registros são validados e colocados de forma ordenada



Inserção de novos blocos

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- Como a corrente cresce?
- Como colocar novos blocos com novas transações
- Qualquer nó pode propor novos blocos? Qual nó pode propor novos blocos?

SOLUÇÃO – COMPETIÇÃO

Quem vencer primeiro um desafio computacional

Quem achar primeiro um hash com X zeros no início

Desafio no Bitcoin (BTC)

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- Encontrar um hash com X zeros no início
- Solução por força bruta – testando nonce de 32 bits
- Quanto maior o número de zeros mais tempo e processamento são necessários
- Ajusta-se a dificuldade do desafio para que em média se leve 10 minutos para que alguém vença um desafio.
- Dificuldade ajustada a cada 2016 blocos
  - Bitcoin começou com 14 zeros e hoje está em zeros
- Estimou-se no projeto que a informação de descoberta do desafio (confirmação de um novo bloco) levaria 1 minuto para chegar em todos os participantes
  - Os mineradores gastaria apenas 10% do tempo minerando blocos já descobertos

Recompensa no Bitcoin

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

Problemas

A recompensa pode deixar de ser atrativa?

-

As taxas cobradas por transação aumentariam, para compensar a baixa remuneração?

-

Transações com valores de taxas baixos são mineradas em blocos?

Vazão da Bitcoin

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

Desafio importante da corrente de blocos

-

Como aumentar a vazão em transações por segundo?

-

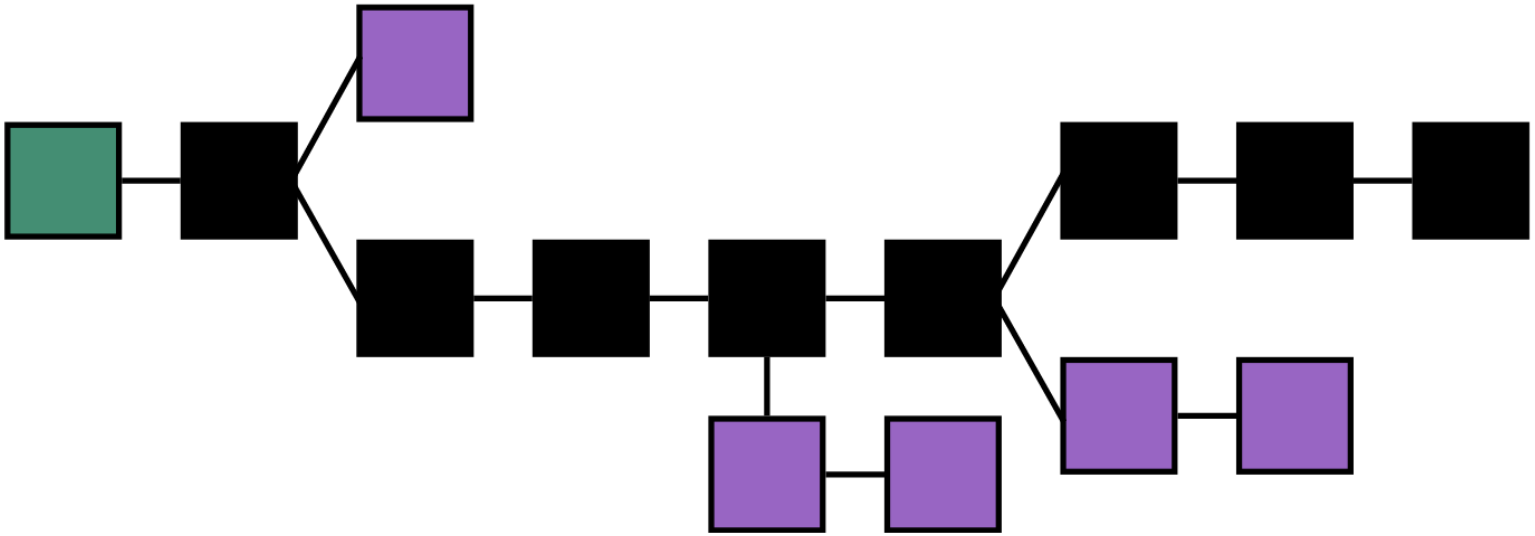
O mecanismo de prova de trabalho consegue uma vazão alta?

50.000 por segundo em pico

Bifurcações na Corrente

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

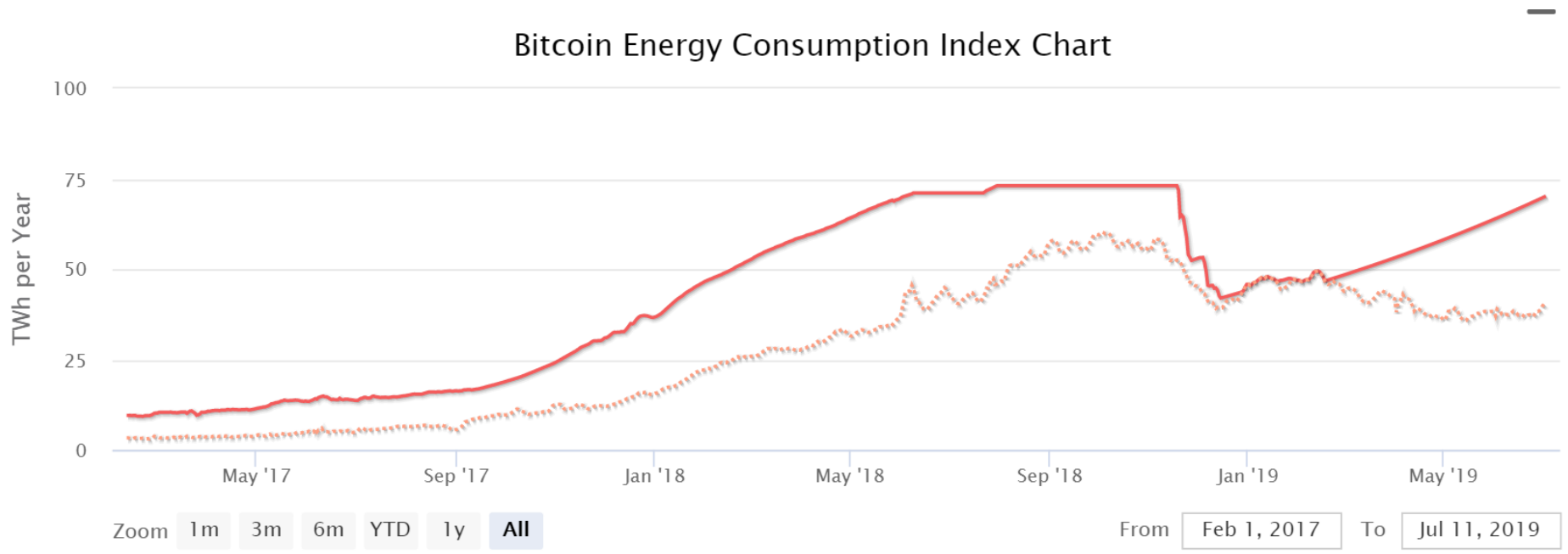
- Quando mineradores resolvem o desafio ao mesmo tempo
  - ocorre uma bifurcação na corrente de blocos
- Regra - vence o ramo com a corrente mais longa
  - maior gasto computacional
- Um bloco só é confirmado após 6 rodadas (estimativa)
  - Blocos podem ser **abandonados**



Prova de Trabalho – Consumo Energético Bitcoin

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- Consumo energético = **70 TWh** (2019)
  - Angra 1 + Angra 2 = 15,6 TWh (2014)
  - Itaipu: 96,5 TWh (2018)
  - Mais energia do que **160 países**!



Fonte: https://digiconomist.net/bitcoin-energy-consumption (Acessado em 4/7/2019)

Prova de Trabalho – Centralização

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

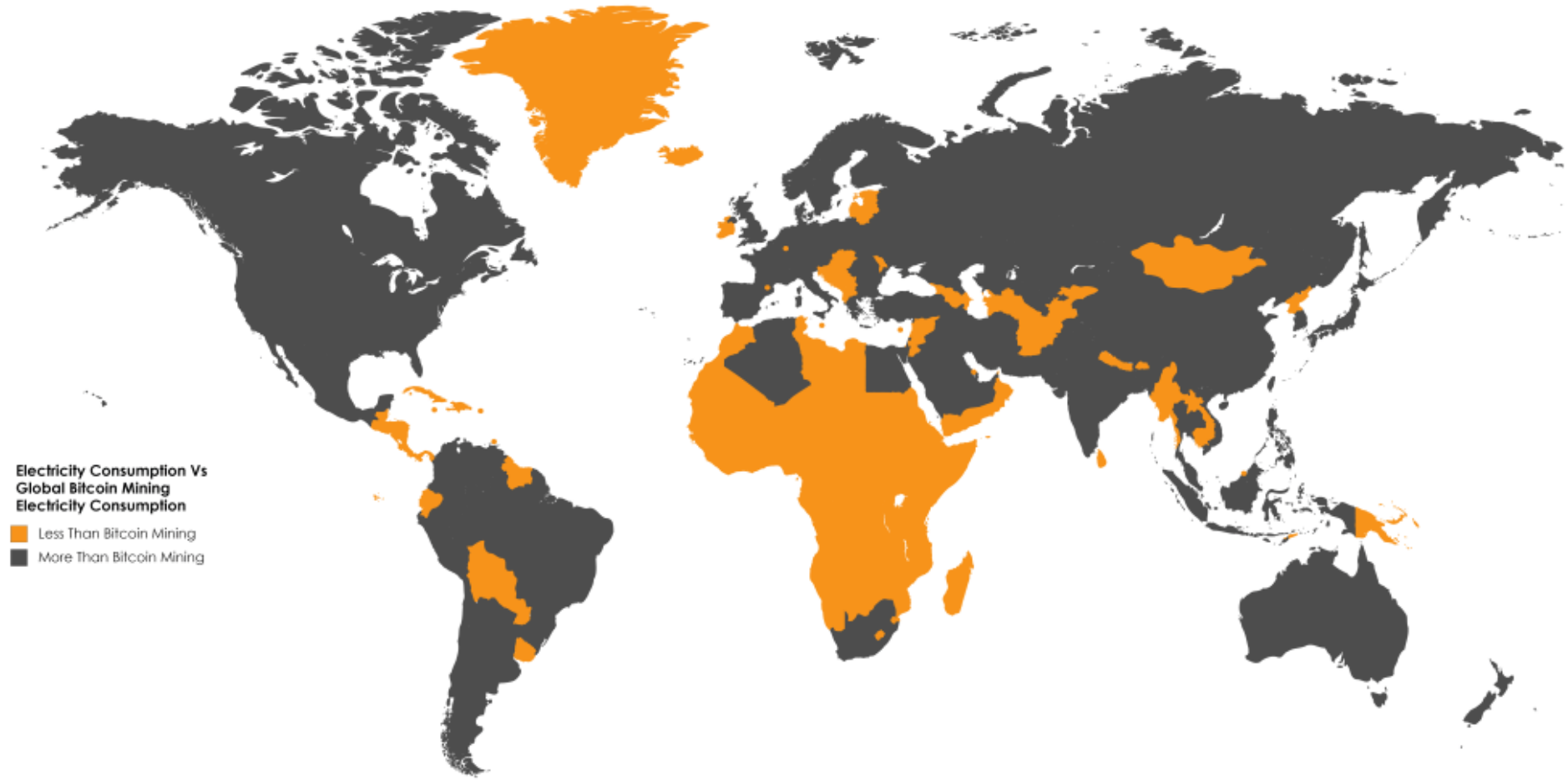
- Centralização em ASICs com alto poder computacional

Países onde os equipamentos são mais baratos são beneficiados (e.g. China)

Fonte: https://www.blockchain.com/pt/pools (Acessado em 4/7/2019)

Prova de Trabalho – Consumo Energético Bitcoin

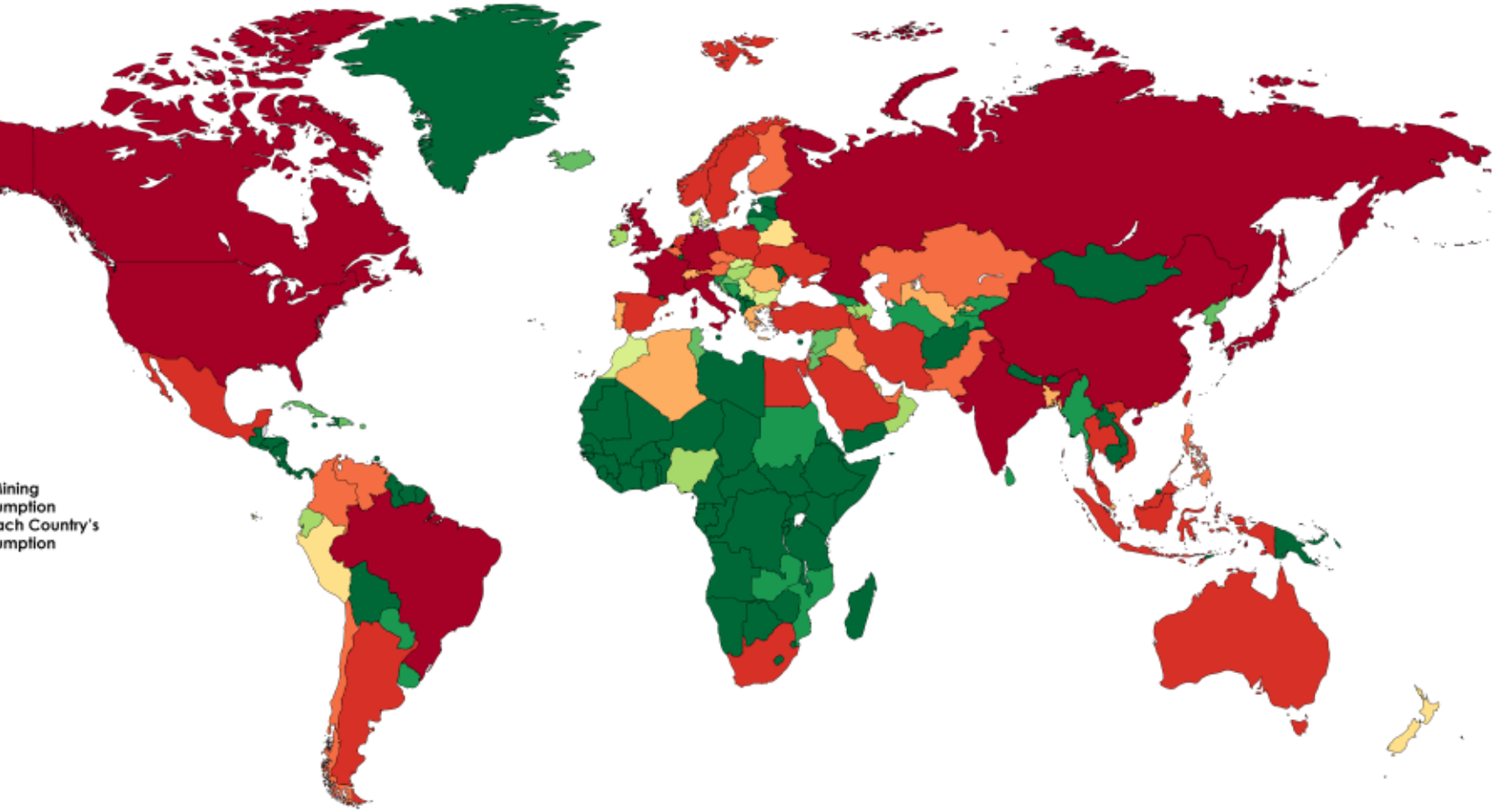
GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO



Source: https://powercompare.co.uk/bitcoin/

Prova de Trabalho – Consumo Energético

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO



Source: https://powercompare.co.uk/bitcoin

Prova de Trabalho – Consumo Energético

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- Custo médio por transação (Bitcoin): 472 kWh
  - Consumo residencial médio no Brasil: 159,8 kWh/mês (2017)

Tabela 3.53 Consumo médio residencial por região e UF (kWh/mês)

Average residential consumption by region and state (kWh/month)

	2012	2013	2014	2015	2016	Δ% (2016/2015)
Brasil	**158,9**	**163,0**	**167,2**	**161,5**	**159,8**	**-1,0**

A energia gasta em **uma transação** de Bitcoin sustentaria uma residência brasileira por **3 meses**!

Prova de Trabalho – Consumo Energético

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

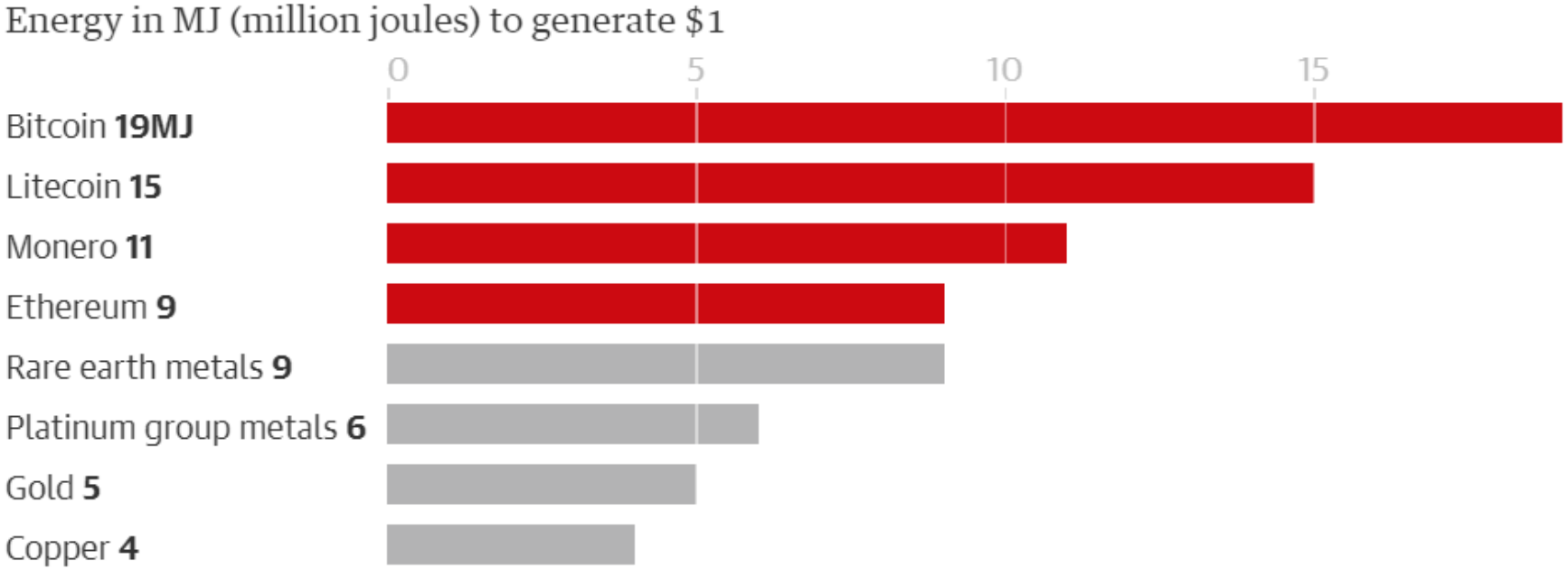
- Custo de minerar 1 Bitcoin por país (em USD)

Fonte: https://www.elitefixtures.com/blog/post/2683/bitcoin-mining-costs-by-country/ (Acessado em 4/7/2019)

Prova de Trabalho – Centralização

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- Minerar Bitcoin é energeticamente **menos eficiente** do que minerar **ouro**



Fonte: https://www.theguardian.com/technology/2018/nov/05/energy-cost-of-mining-bitcoin-more-than-twice-that-of-copper-or-gold (Acessado em 4/7/2019)

Ethereum

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- Proposto por Vitalik Buterin em 2014 (20 anos)
- Baseado em prova de trabalho, como o Bitcoin
- Grande inovação: contratos inteligentes
  - Implementa uma máquina de Turing Completa
    - Ethereum Virtual Machine

Ethereum

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- Tempo de adicionar um bloco: 15 segundos (vs 10min do Bitcoin)
  - Tempo de confirmação de um bloco é menor que 1min
- Vazão média: 15 transações por segundo (vs 7 tx/s do Bitcoin)
- Algoritmo ethash
  - Gargalo é o acesso à memória
  - Menos suscetível a ASICs

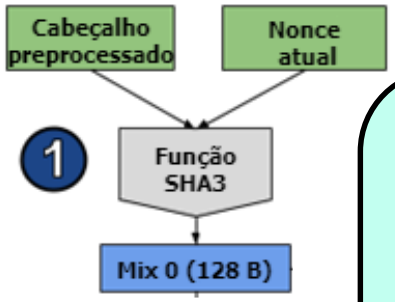
Ethereum - Funcionamento do ethash

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- Depende de um dataset pseudorrandômico
  - Dataset chamado de **DAG** - *Directed Acyclic Graph*
    - Matriz de tipo inteiro sem sinal de dimensão N x 16
  - Regenerado a cada 30.000 blocos → Em torno de 5 dias
  - Crescimento linear → Atualmente 3.19 GB *
    - Acompanha o crescimento da corrente de blocos

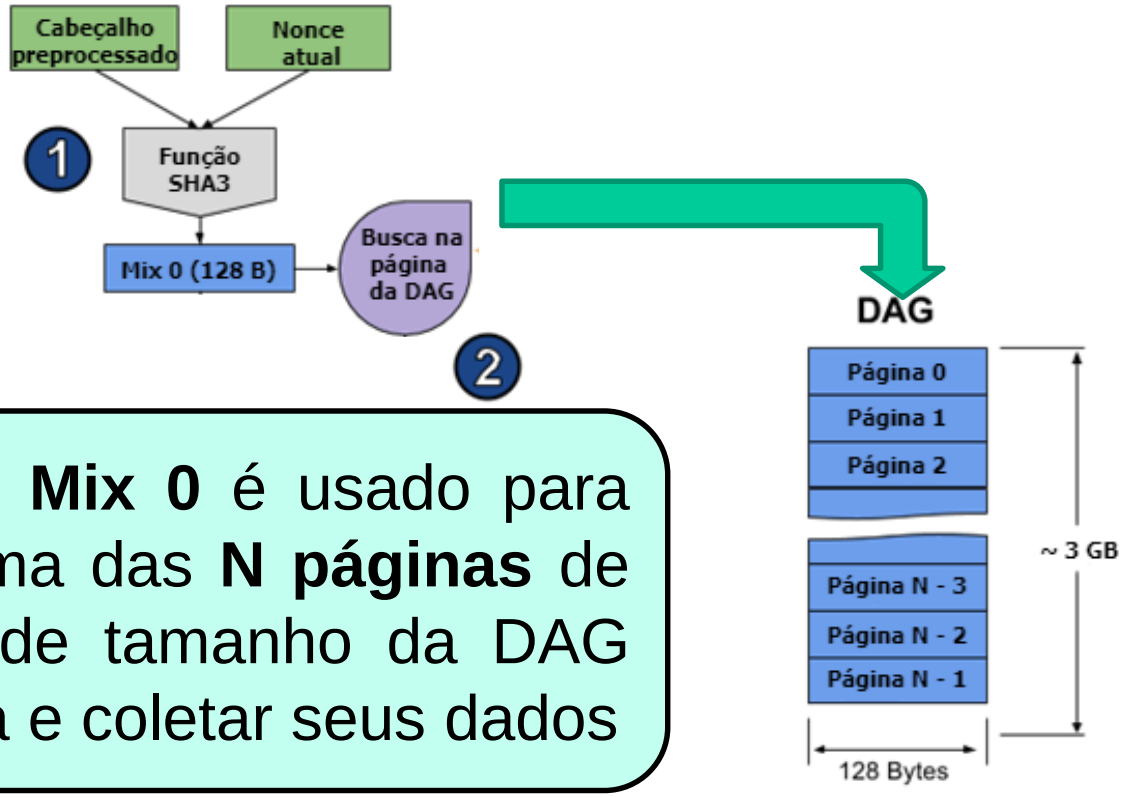
* Fonte: https://investoon.com/tools/dag_size (Acessado em 1/7/2019)

Ethereum – Funcionamento do ethash



Etapa 1: o cabeçalho preprocessado do último bloco e o nonce atual são combinados com uma função hash **SHA3** gerando uma saída de 128 bytes chamada de **Mix 0**.

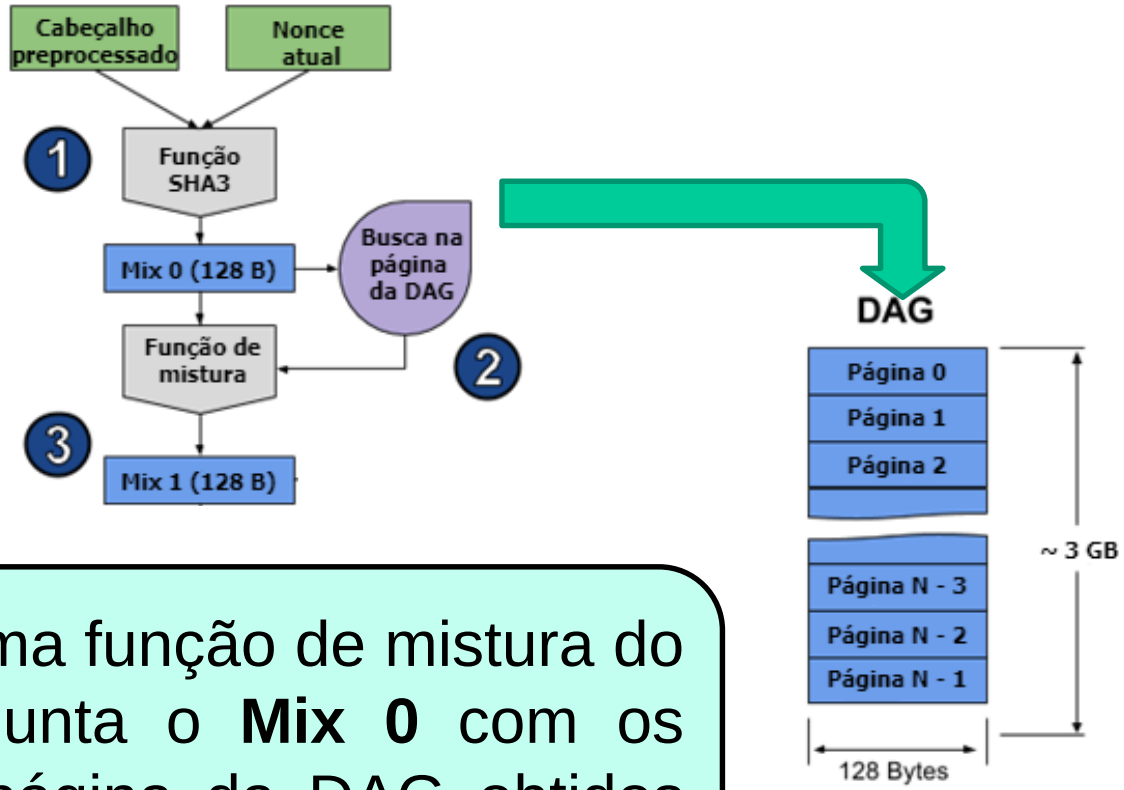
Ethereum – Funcionamento do ethash



Etapa 2: o **Mix 0** é usado para escolher uma das **N páginas** de 128 bytes de tamanho da DAG na memória e coletar seus dados

Ethereum – Funcionamento do ethash

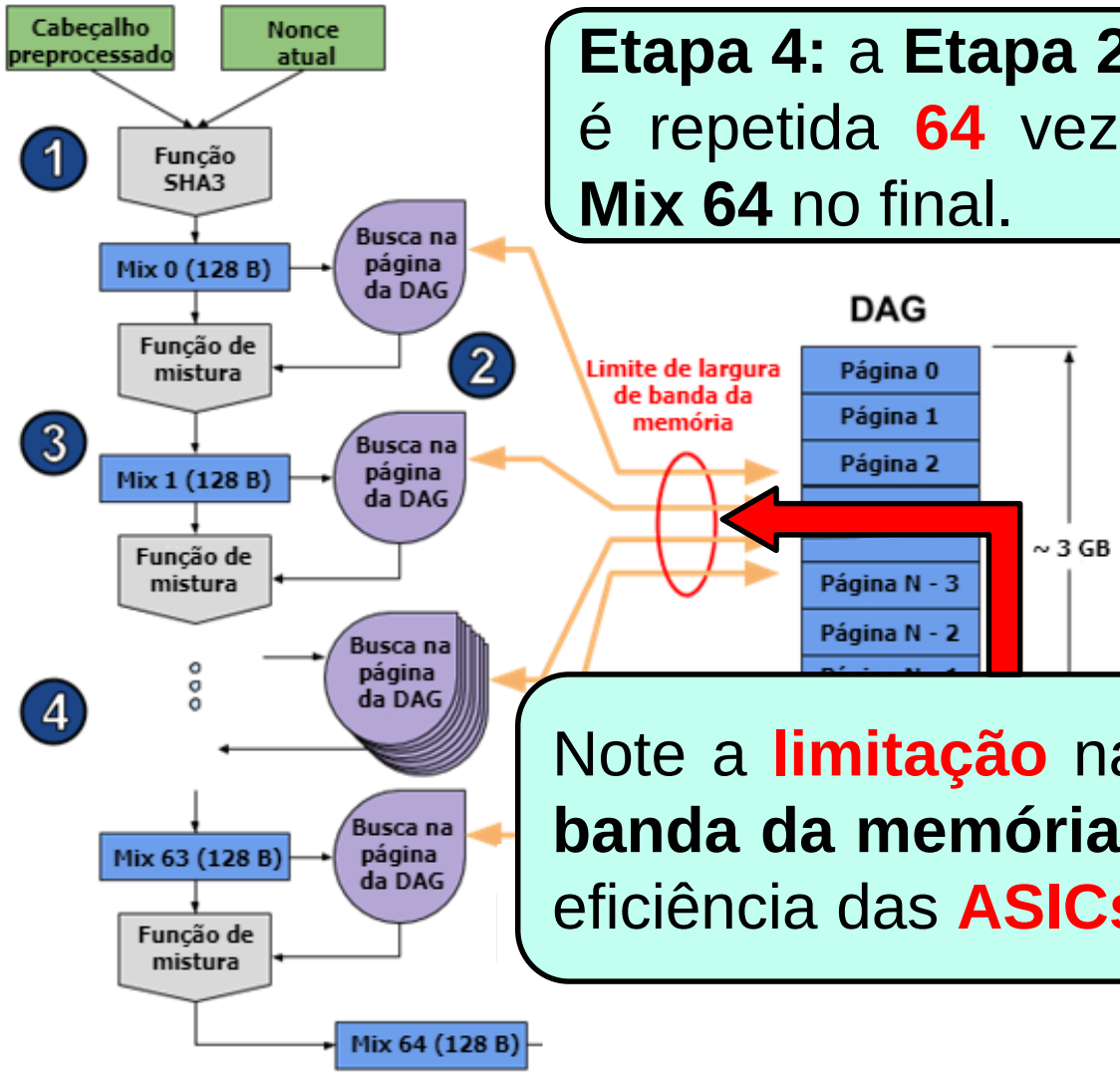
GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO



Etapa 3: uma função de mistura do Ethereum junta o **Mix 0** com os dados da página da DAG obtidos na **Etapa 2**, assim gerando o **Mix 1**

Ethereum – Funcionamento do ethash

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO



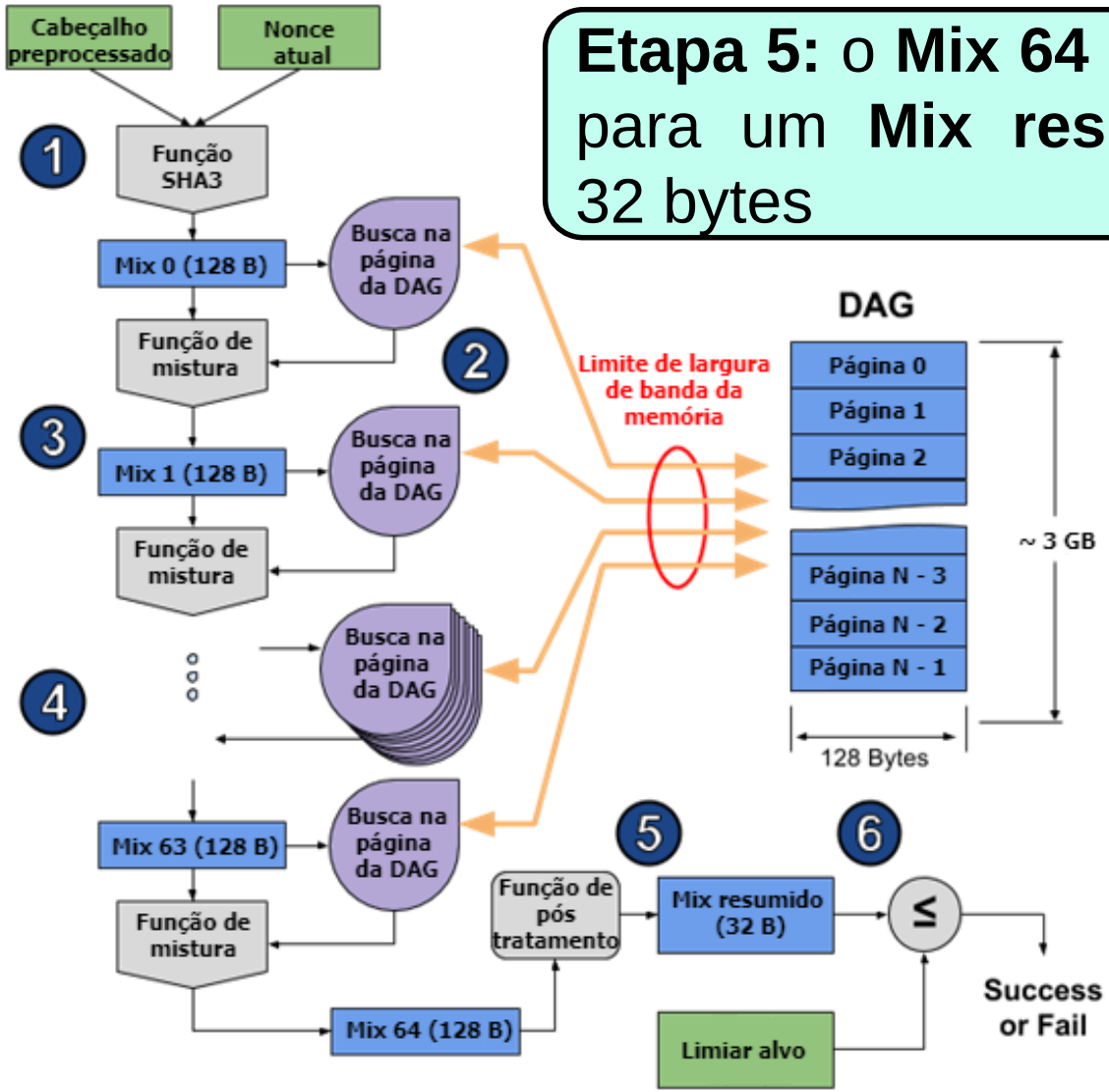
Etapa 4: a **Etapa 2** e a **Etapa 3** é repetida **64** vezes e gera o **Mix 64** no final.

Note a **limitação** na **largura de banda da memória** que reduz a eficiência das **ASICs**

Ethereum – Funcionamento do ethash

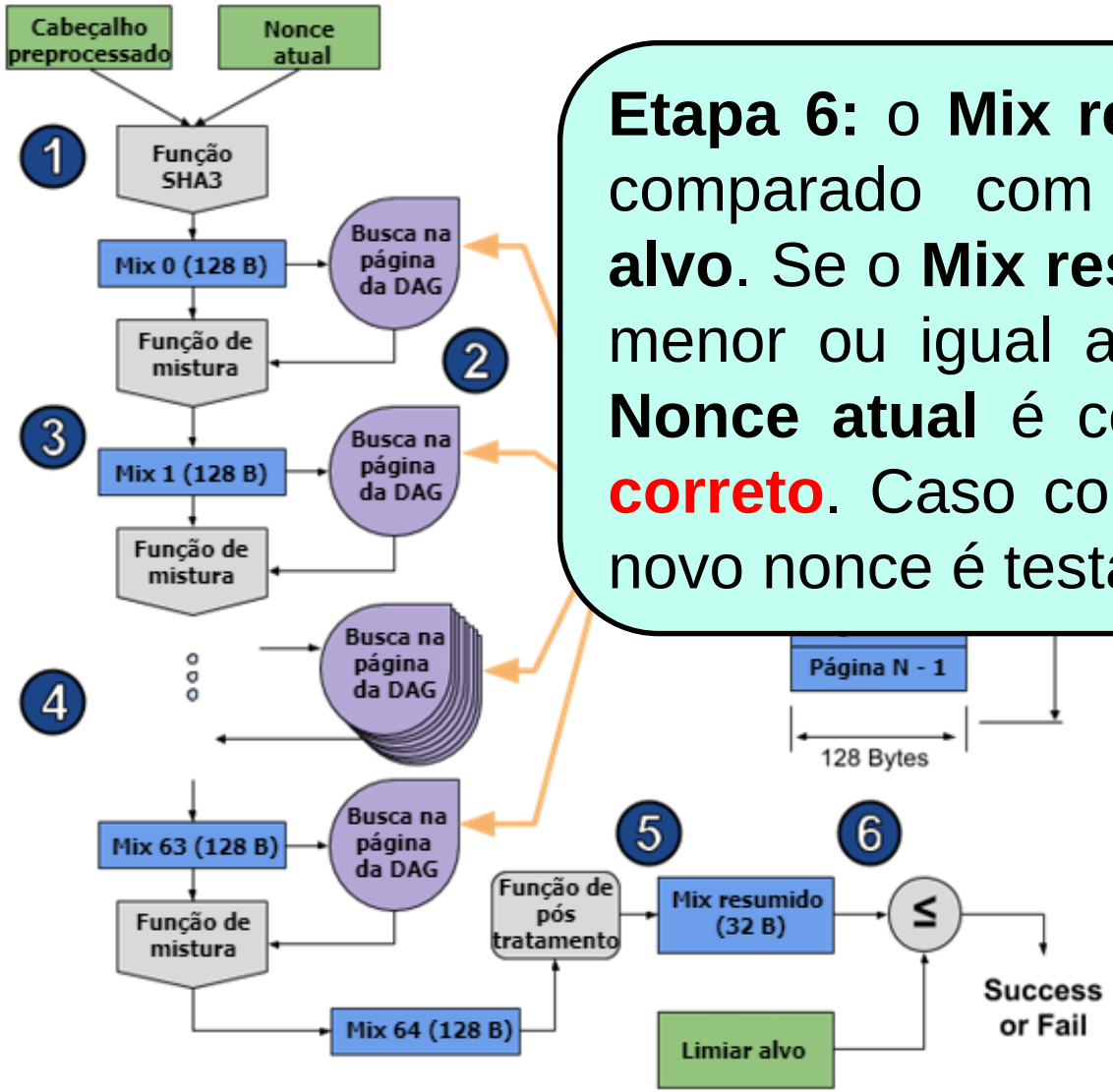
GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

Etapa 5: o **Mix 64** é reduzido para um **Mix resumido** de 32 bytes



Ethereum – Funcionamento do ethash

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO



Etapa 6: o **Mix resumido** é comparado com o **Limiar alvo**. Se o **Mix resumido** for menor ou igual ao limiar, o **Nonce atual** é considerado **correto**. Caso contrário, um novo nonce é testado

Desafios de protocos de consenso

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

Consumo energético inviabiliza a Prova de Trabalho?

Como conseguir aumentar a vazão em transações por segundo?

Existem propostas sem riscos de centralização?

Existe a possibilidade de uma corrente de blocos escalável que atenda a Internet? IoT? Bilhões de dispositivos?

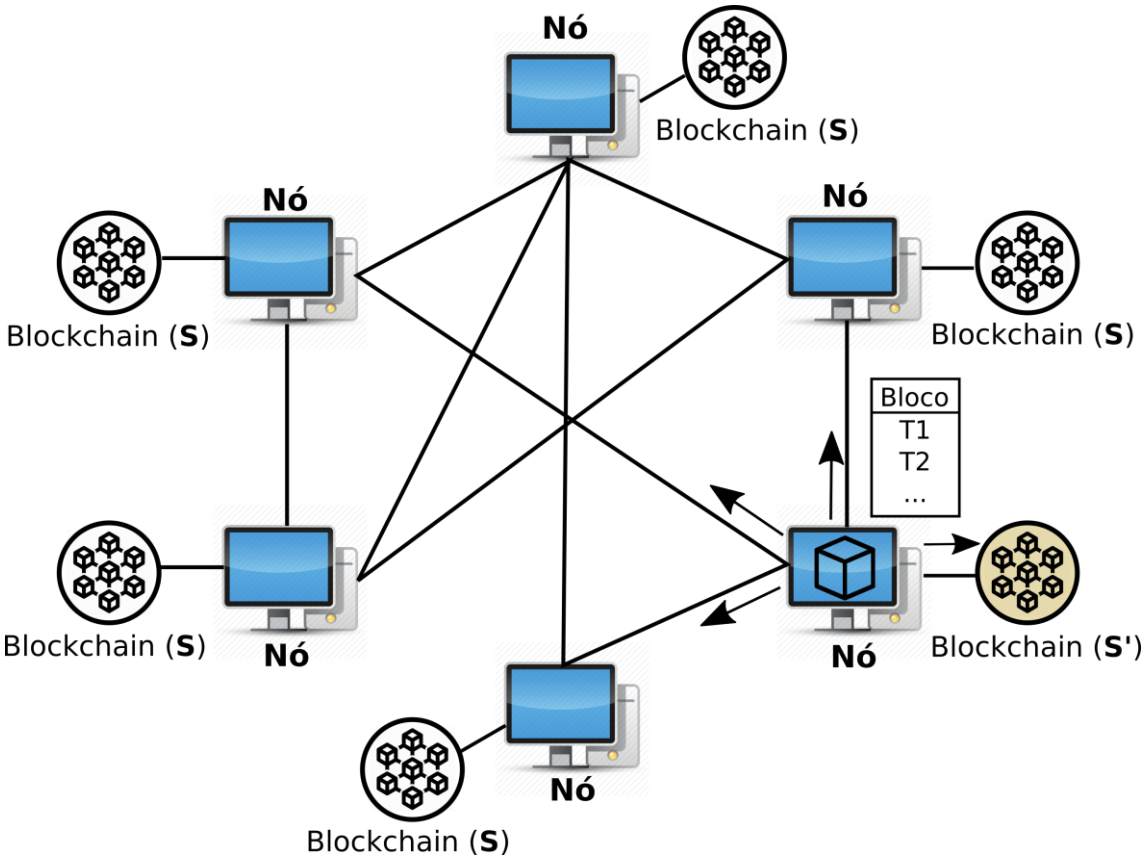
GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

Parte II
Protocolos de Consenso em Corrente de Blocos

Consenso em Correntes de Blocos

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

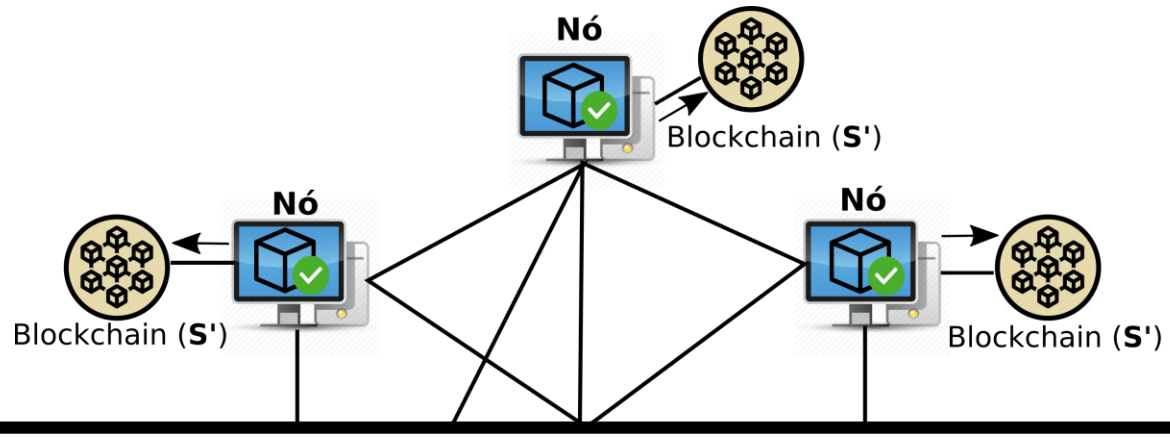
- A corrente de blocos é um **sistema distribuído**
  - Os nós da rede devem concordar sobre o estado global



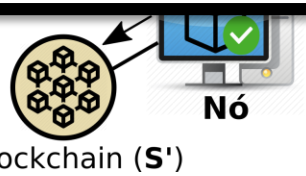
Consenso em Correntes de Blocos

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- A corrente de blocos é um **sistema distribuído**
  - Os nós da rede devem concordar sobre o estado global



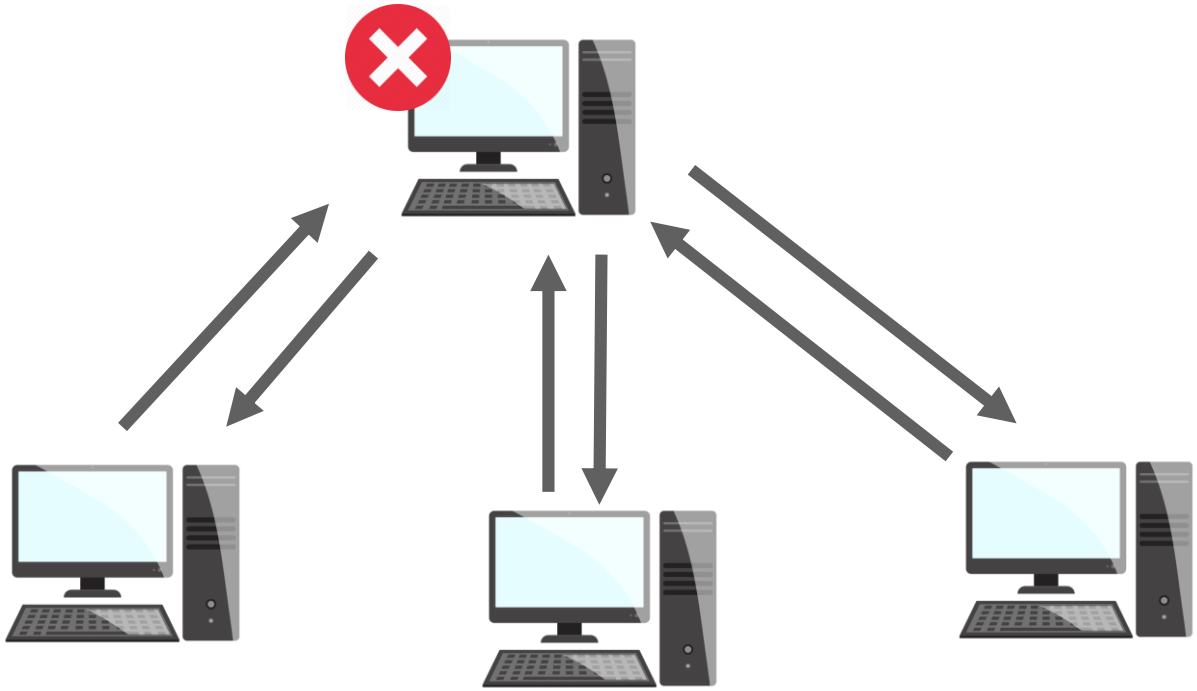
Desafio: **obter consenso** em correntes de blocos



Tipos de Falha – Falhas de Parada (Desastrosas)

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- Falhas de parada ou desastrosas (*crash faults*)
  - O participante para de responder às mensagens da rede



Problema dos Generais Bizantinos

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- Um grupo de **n** generais está acampado com seus batalhões nos arredores de uma cidade inimiga

Problema: Como garantir que as múltiplas entidades cheguem em um acordo sobre o plano de combate?

- Os generais desconfiam se os outros são honestos ou traidores

O ataque só é bem-sucedido se a maioria dos generais atacar

Problema dos Generais Bizantinos

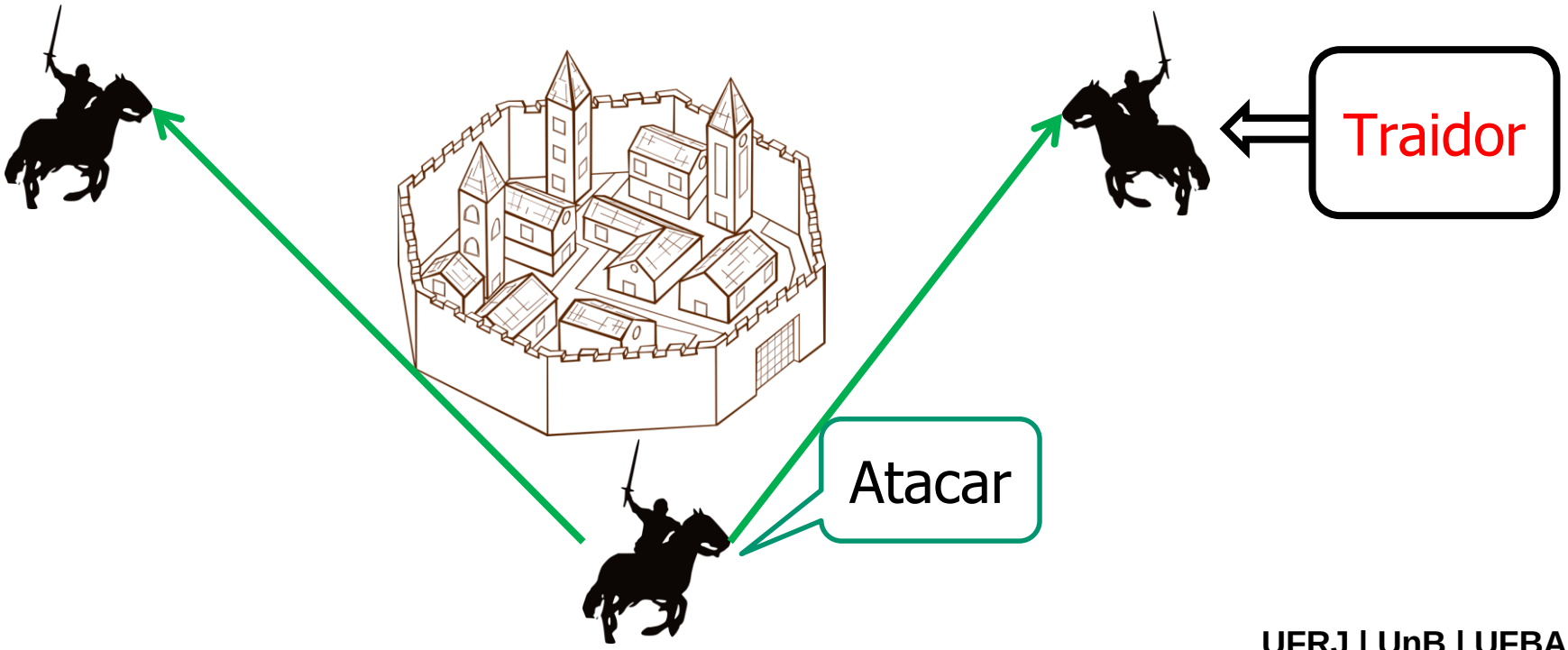
GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- Condições:
  - Todos os generais honestos decidem no mesmo plano de ação
  - Um pequeno número de traidores não pode deixar os generais honestos chegarem na decisão errada

Problema dos Generais Bizantinos

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

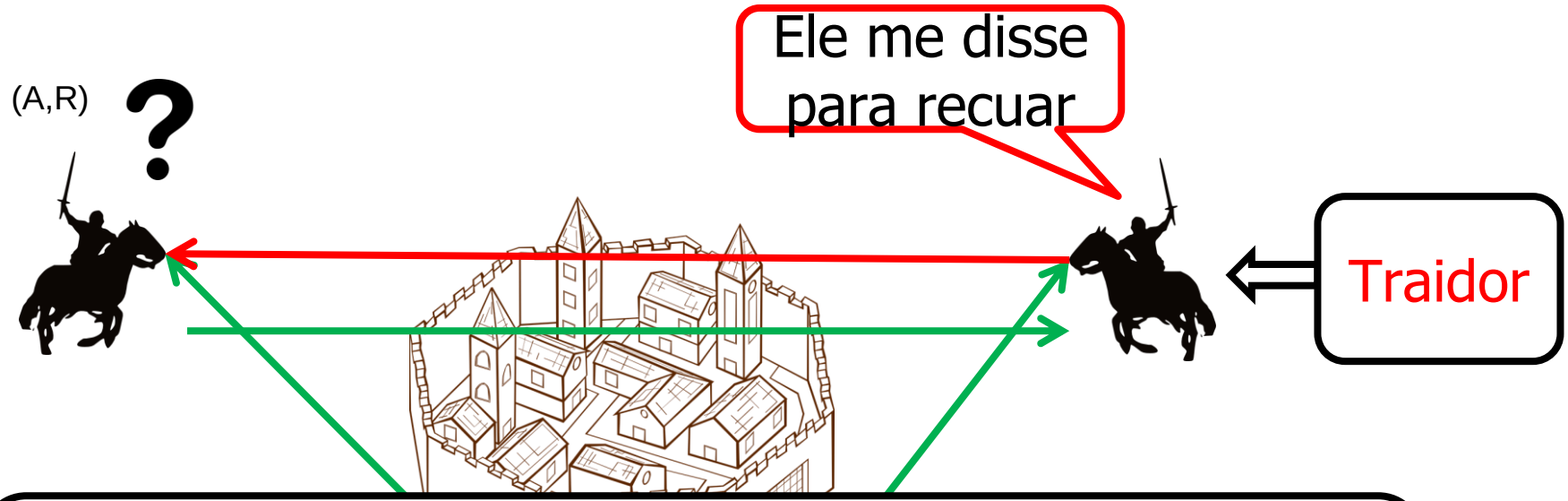
- Considerando 3 generais e 1 traidor



Problema dos Generais Bizantinos

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- Considerando 3 generais e 1 traidor

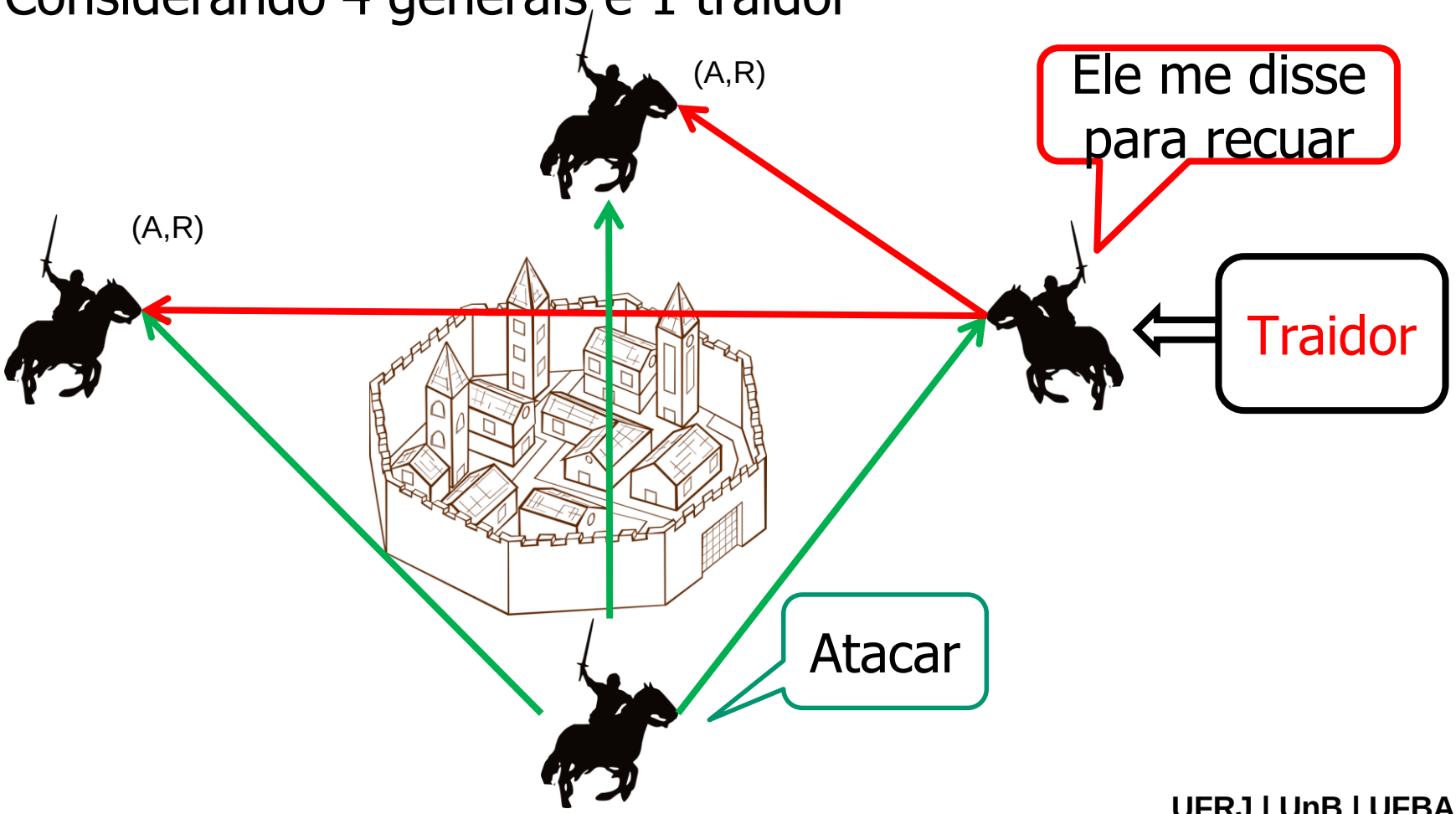


Não existe solução para 3 generais!

Problema dos Generais Bizantinos

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

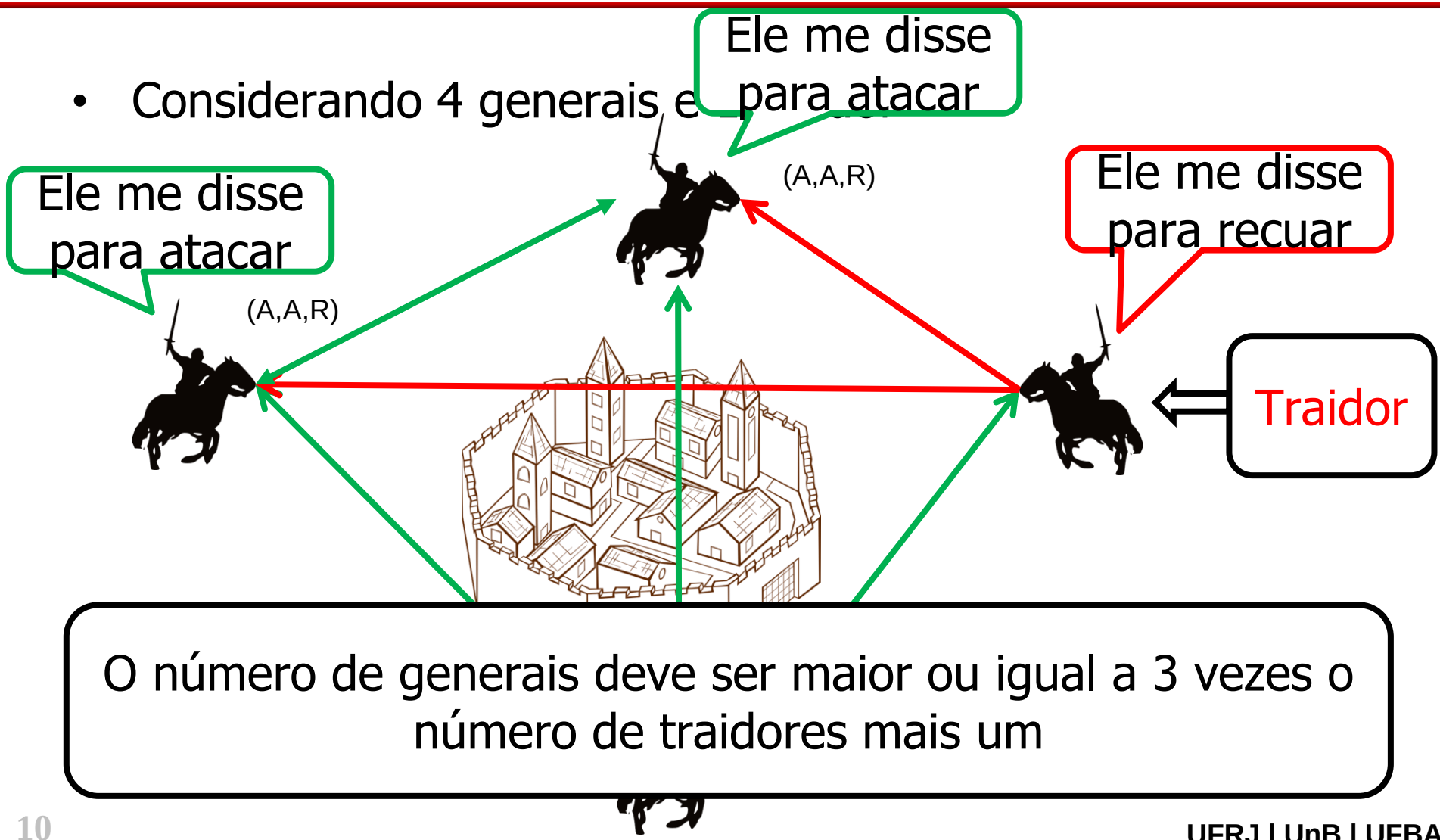
- Considerando 4 generais e 1 traidor



Problema dos Generais Bizantinos

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- Considerando 4 generais e [illegible]



O número de generais deve ser maior ou igual a 3 vezes o número de traidores mais um

Consenso em Correntes de Blocos

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- Participantes do consenso podem **falhar**
  - Falhas desastrosas → panes ou perda de conectividade
  - Falhas bizantinas → comportamento malicioso

- Idealmente, o consenso deve garantir três propriedades
  - **Terminação (*liveness*)**: todo nó correto eventualmente decide por um bloco B_i
  - **Acordo (*safety*)**: O bloco B_i de todo nó correto é idêntico.
  - **Validade (validity)**: o bloco B_i é o bloco proposto por

Resultado FLP: é **impossível garantir** consenso em sistemas distribuídos assíncronos (Fischer, Lynch, Patterson - FLP, 1985)

Consenso Probabilístico – PoW, PoS, etc.

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- Assume-se **consenso eventual** entre os participantes
  - **Premissa**: Em um tempo finito, todos os participantes aceitarão o novo bloco
    - Participantes adicionam blocos de forma independente e os difundem na rede
      - Não há troca de mensagens
    - **Acordo** não é garantido (bifurcações)
    - É necessário **incentivar** participantes a aceitarem o bloco e **prevenir *spam*** de blocos
- Alta **escalabilidade** e em detrimento do **desempenho**
  - Ideal para redes **públicas**

Consenso Determinístico – CFT e BFT

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- Assume-se **eventual sincronia** da rede
  - **Premissa:** mensagens chegam ao destino em tempo finito
    - Participantes podem trocar mensagens para concordar sobre um novo bloco
    - É necessário **conectividade** e **identificação** de todos os participantes
    - **Terminação** não é garantida (perda de mensagens)
- Alto **desempenho** e em detrimento da **escalabilidade**
  - Ideal para **redes federadas** ou **empresariais**
- Protocolos tolerantes a falhas de paradas ou bizantinas (CFT - BFT)

Protocolos de Consenso Tolerantes à Falhas

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

Protocolos de consenso determinístico tolerantes a falhas

Garante que a consistência do sistema o tempo todo

—

Não existem bifurcações na corrente de blocos.

à

)

Falhas de parada e bizantina

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- da
- as

(Bizantine Fault Tolerant - BFT)

Protocolos tolerantes a falhas bizantinas são mais confiáveis, pois são resistentes a comportamento maliciosos e até em conluio

Para tolerar a mesma quantidade de falhas, os protocolos que toleram falhas bizantinas requerem mais réplicas que os protocolos que toleram falhas por parada
CFT 2f+1
BFT 3f+1

Protocolos de Consenso Tolerantes a Falhas de Paradas

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- Protocolos da família PAXOS
  - Protocolos baseados em troca de mensagens
  - Todo nó correto pode enviar mensagens para todos os nós participantes corretos com garantia de entrega ordenada
    - Difusão atômica
  - Procedimento dividido em "épocas" com um líder responsável por cada época

Difusão Atômica

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- Assegura que cada nó correto entrega a mesma sequência de mensagens nos eventos de entrga
- Assegura as seguintes propriedades
  - **validade**: se um no correto *p* difunde a mensagem *m*, então *p* eventualmente entrega *m*
  - **acordo**: se uma mensagem *m* é entregue a algum nó correto, então *m* é eventiualmente entregue por todos os nós corretos
  - **integridade**: nenhum nó correto entrega a mesma mensagem mais de uma vez, além disso, se um nó correto entrega a mensagem *m* e o originador *p* de *m* é um nó correto, então *m* foi previamente difundido por *p*
  - **ordem total**: para as mensagens m_1 e m_2 suponha p e q dois nós corretos que entregam m_1 e m_2. Então *p* entrega m_1 antes de m_2 se e somente se *q* entrega m_1 antes de m_2

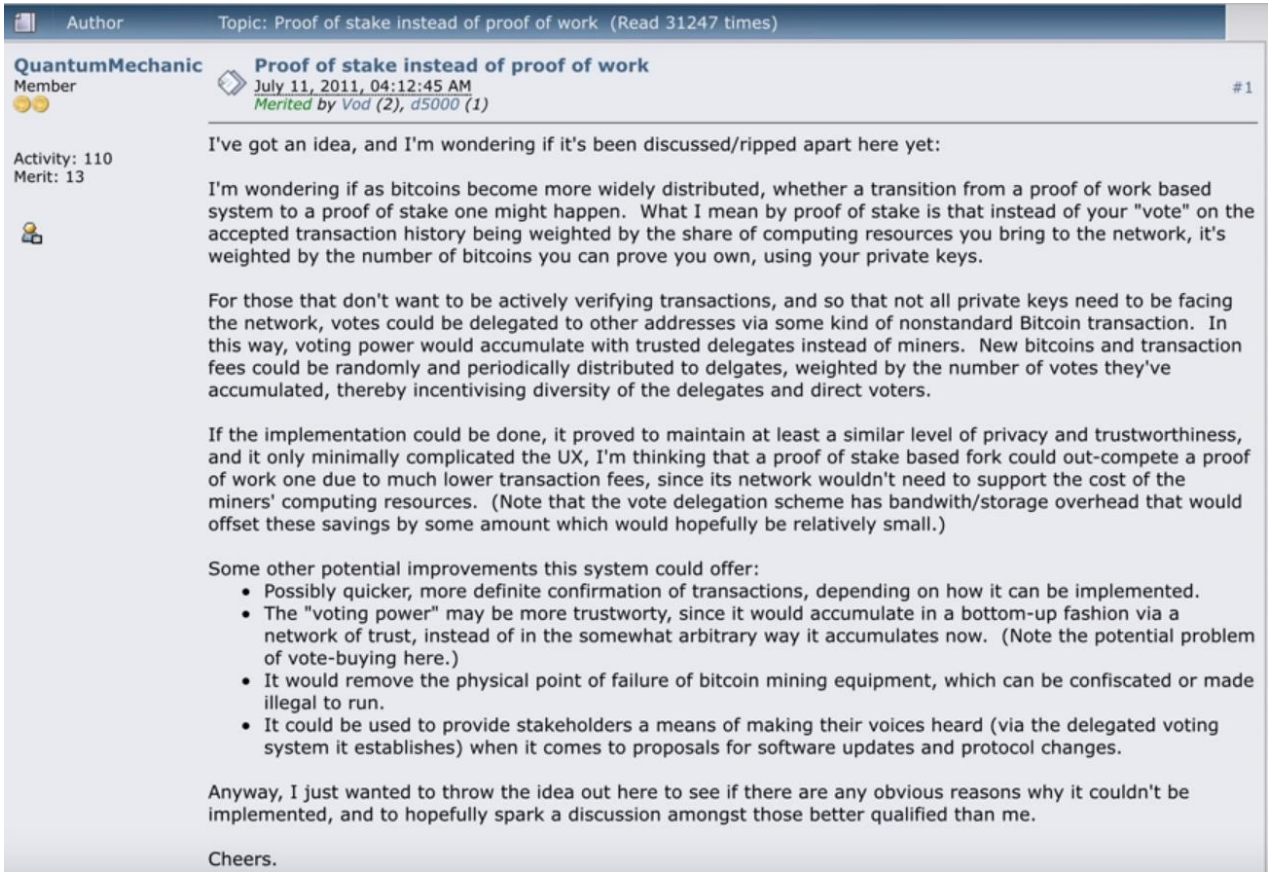
GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

Protocolos Baseados em Consenso Probabilístico – Prova de Posse (PoS)

Prova de Posse (*Proof of Stake* – PoS)

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- Proposto por um usuário no fórum do Bitcoin em 2011



Author | Topic: Proof of stake instead of proof of work (Read 31247 times)

QuantumMechanic
Member

Activity: 110
Merit: 13

Proof of stake instead of proof of work
July 11, 2011, 04:12:45 AM
Merited by Vod (2), d5000 (1)

#1

I've got an idea, and I'm wondering if it's been discussed/ripped apart here yet:

I'm wondering if as bitcoins become more widely distributed, whether a transition from a proof of work based system to a proof of stake one might happen. What I mean by proof of stake is that instead of your "vote" on the accepted transaction history being weighted by the share of computing resources you bring to the network, it's weighted by the number of bitcoins you can prove you own, using your private keys.

For those that don't want to be actively verifying transactions, and so that not all private keys need to be facing the network, votes could be delegated to other addresses via some kind of nonstandard Bitcoin transaction. In this way, voting power would accumulate with trusted delegates instead of miners. New bitcoins and transaction fees could be randomly and periodically distributed to delgates, weighted by the number of votes they've accumulated, thereby incentivising diversity of the delegates and direct voters.

If the implementation could be done, it proved to maintain at least a similar level of privacy and trustworthiness, and it only minimally complicated the UX, I'm thinking that a proof of stake based fork could out-compete a proof of work one due to much lower transaction fees, since its network wouldn't need to support the cost of the miners' computing resources. (Note that the vote delegation scheme has bandwith/storage overhead that would offset these savings by some amount which would hopefully be relatively small.)

Some other potential improvements this system could offer:

- Possibly quicker, more definite confirmation of transactions, depending on how it can be implemented.
- The "voting power" may be more trustworty, since it would accumulate in a bottom-up fashion via a network of trust, instead of in the somewhat arbitrary way it accumulates now. (Note the potential problem of vote-buying here.)
- It would remove the physical point of failure of bitcoin mining equipment, which can be confiscated or made illegal to run.
- It could be used to provide stakeholders a means of making their voices heard (via the delegated voting system it establishes) when it comes to proposals for software updates and protocol changes.

Anyway, I just wanted to throw the idea out here to see if there are any obvious reasons why it couldn't be implemented, and to hopefully spark a discussion amongst those better qualified than me.

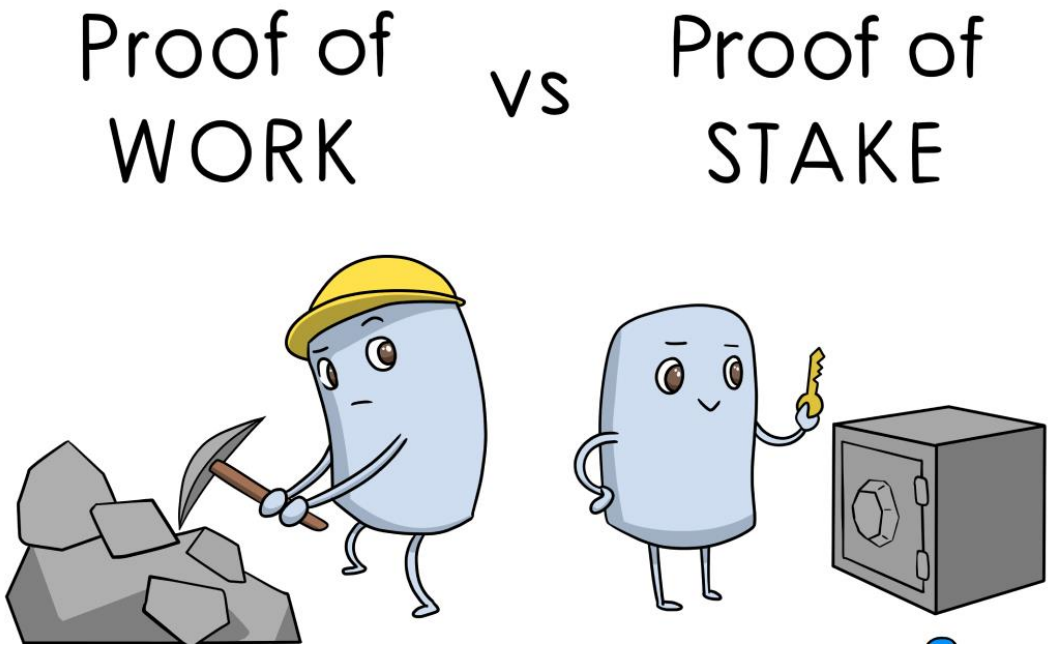
Cheers.

Fonte: https://bitcointalk.org/index.php?topic=27787.0 (Acessado em 4/7/2019)

Prova de Posse (Proof of Stake – PoS)

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- Ideia: probabilidade de "minerar" um bloco é **proporcional ao valor** que o participante **investe**
  - Exige pouco ou nenhum poder computacional
  - "Mineração virtual"

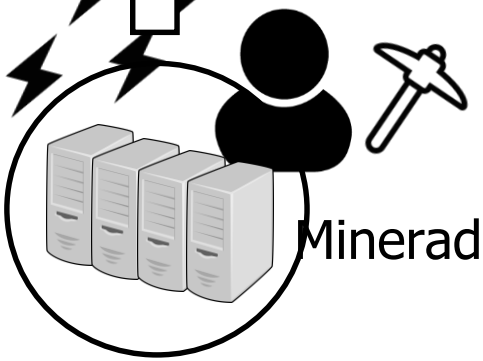


Prova de Participação (*Proof of Stake* – PoS)

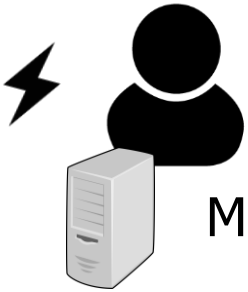
GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

Maior poder computacional → propõe e de ser e → Maior probabilidade de resolver o desafio

montante/participação que possui na rede



Minerador



Minerador

Parte interessada (*stakeholder*)

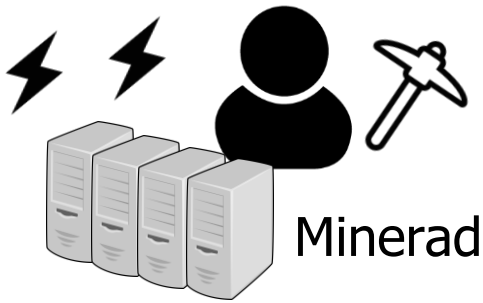
Parte interessada (*stakeholder*)

Prova de Participação (*Proof of Stake* – PoS)

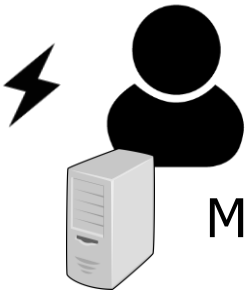
GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

Maior probabilidade de propor o bloco

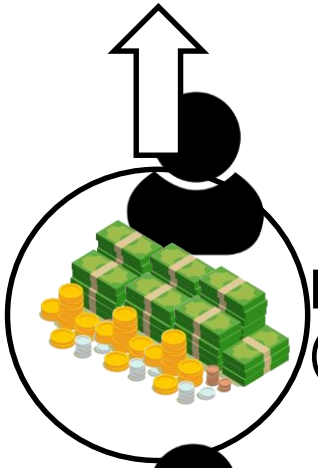
Maior participação



Minerador



Minerador



Parte interessada (*stakeholder*)

Parte interessada (*stakeholder*)

Prova de Posse (*Proof of Stake* – PoS)

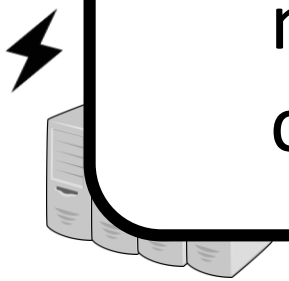
GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

Maior probabilidade de pro

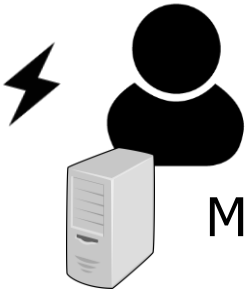
ente

Maior

Conclusão: A prova de posse não demanda alto poder computacional!



ssada
(*stakeholder*)



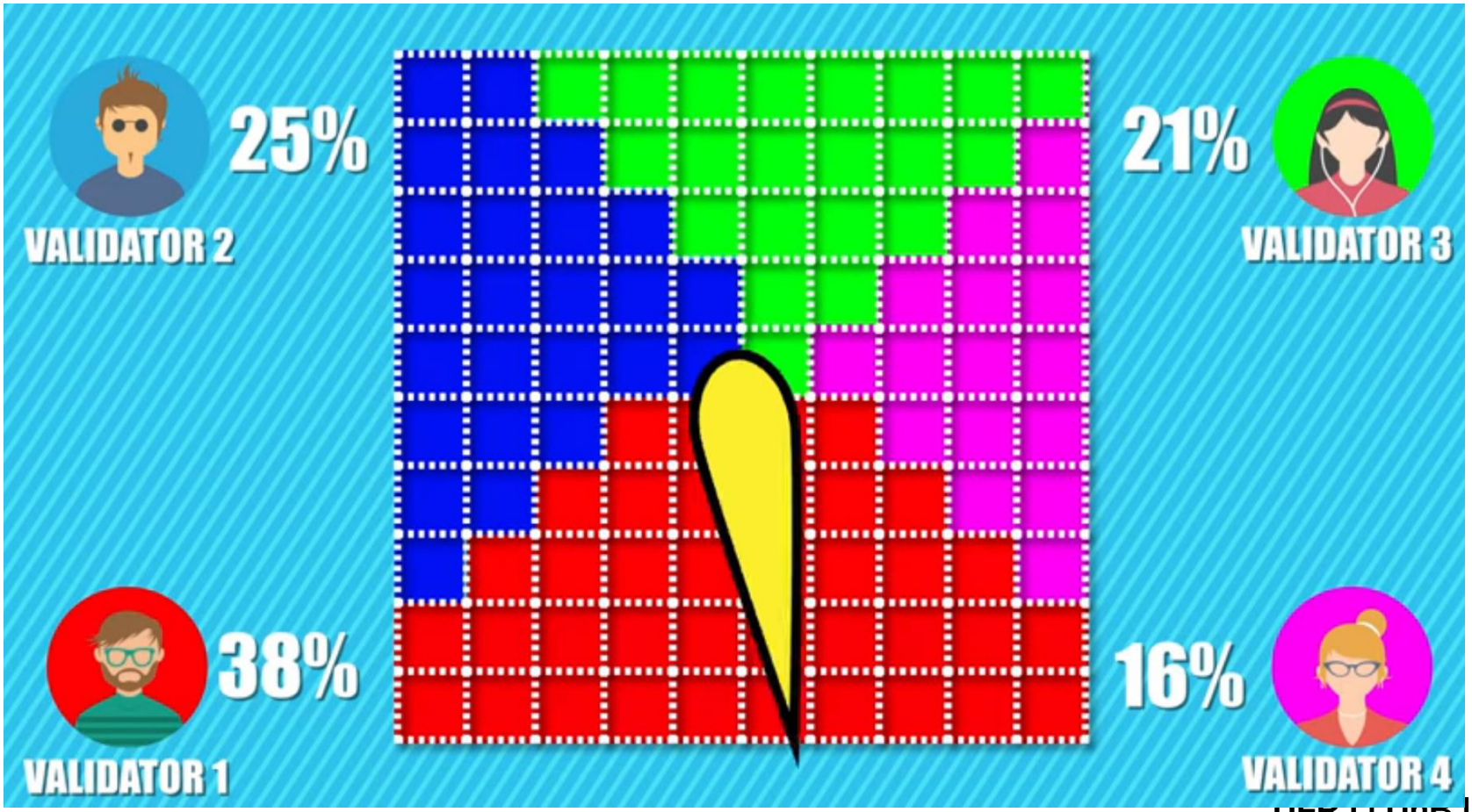
Minerador

Parte interessada
(*stakeholder*)

Prova de Posse – Conceitos

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- Como um sorteio com pesos proporcionais à posse...



Prova de Posse – Conceitos

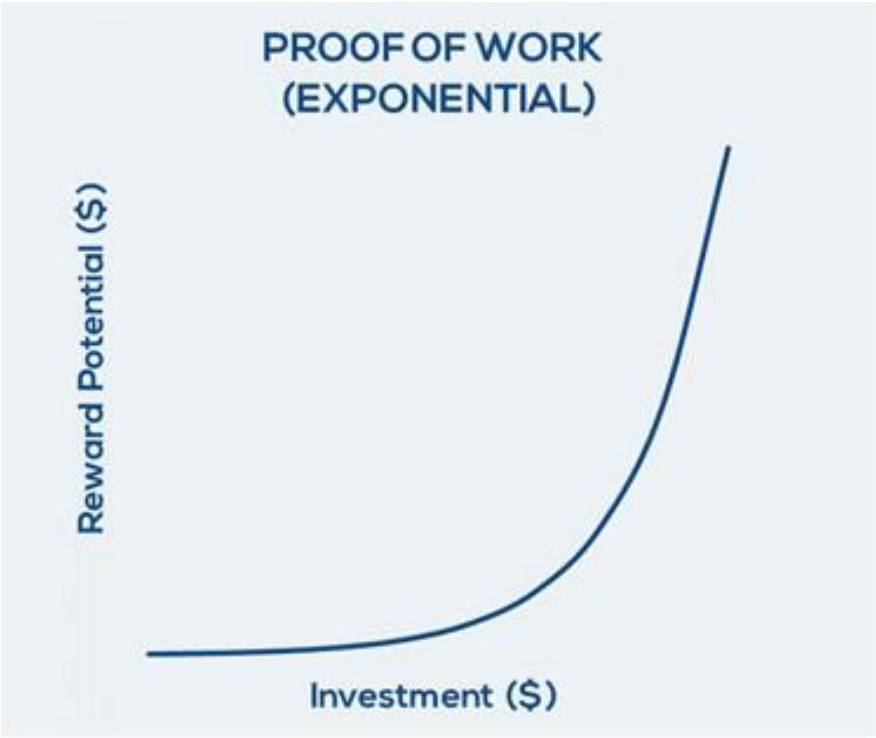
GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

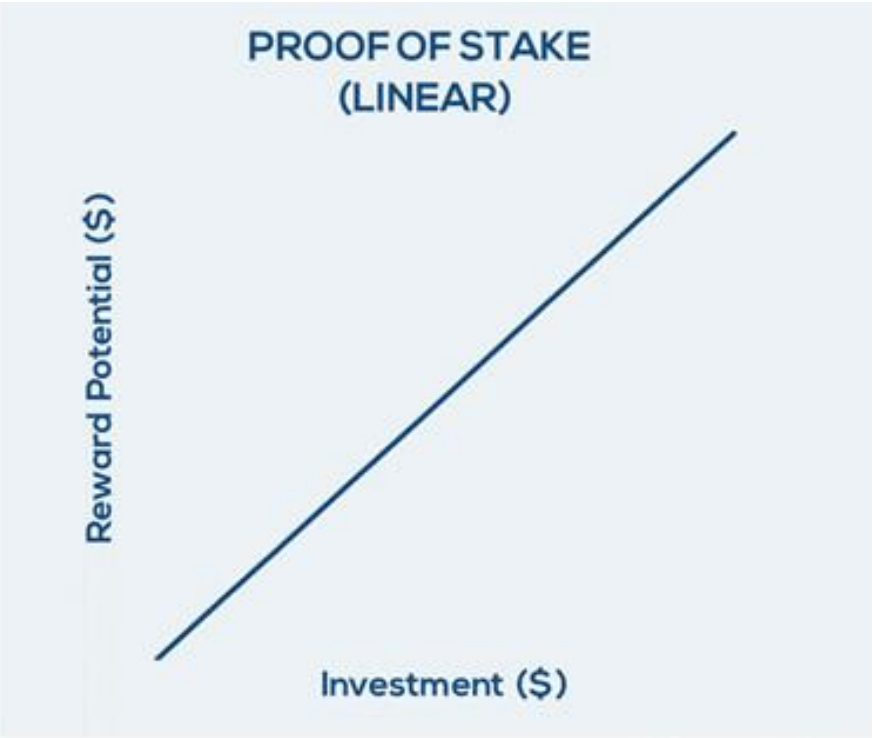
- Mineradores → validadores/*minters*|*stakeholders*
- Os validadores possuem interesse na manutenção do sistema
- Elimina as recompensas por gasto computacional
  - Porém mantém a recompensa das taxas de transações

Prova de Posse (PoS) vs Prova de Trabalho (PoW)

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- Potencial de recompensa
  - PoW: vulnerável a **economia de escala**
  - PoS: "um dólar é um dólar"





Fonte: https://www.bitdegree.org/tutorials/proof-of-work-vs-proof-of-stake/ (Acessado em 4/7/2019)

Prova de Posse – Desafios

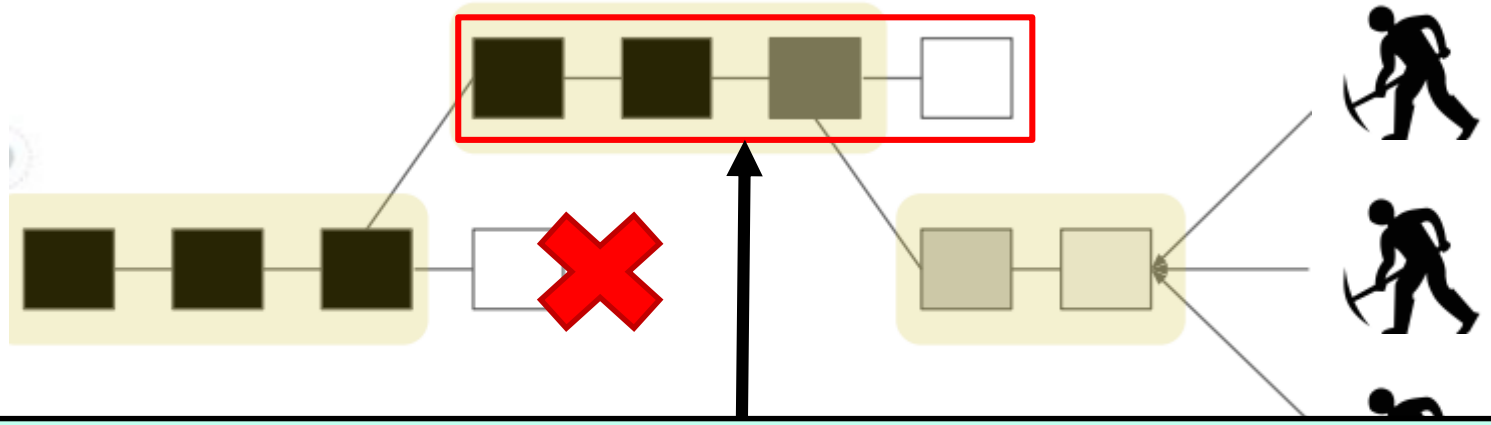
GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- Porém, a ausência de custo computacional para propor um bloco traz novos desafios...
  - Problema do "nada a perder" (*nothing at stake*)
  - Problema do ataque de longo alcance (*long range attack*)

Prova de Posse – Ataques de Longo Alcance

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- Com a finalidade probabilística, podem acontecer bifurcações...

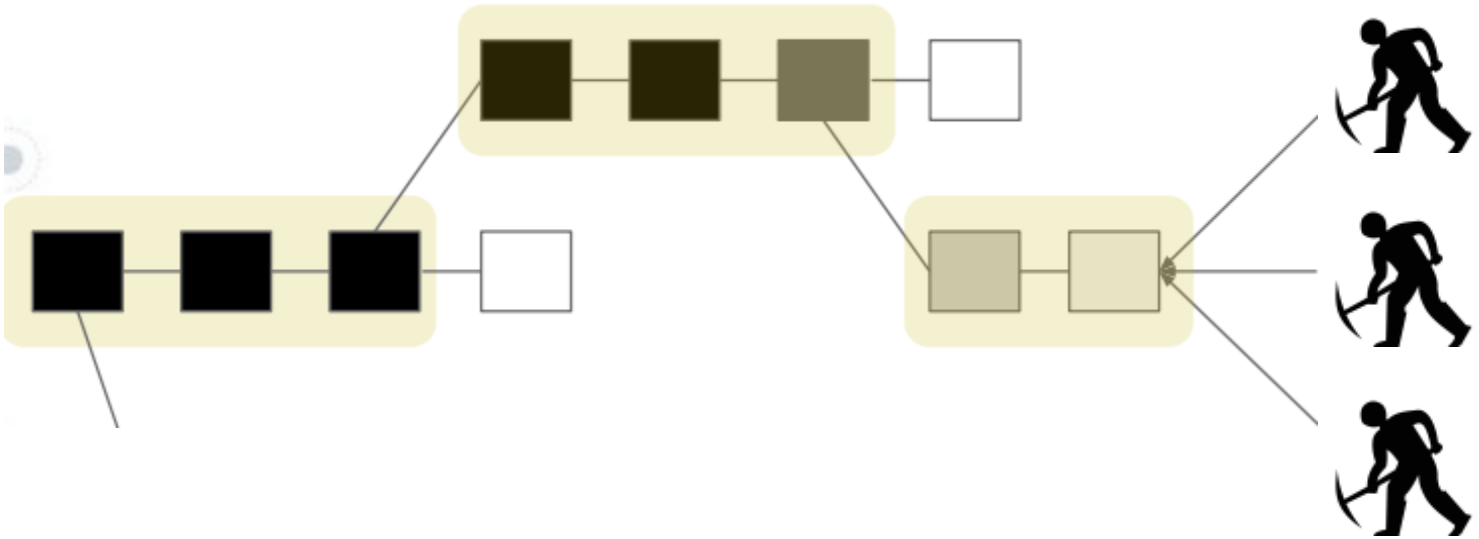


O caminho legítimo é definido pela regra da maior cadeia e os outros caminhos são abandonados

Prova de Posse – Ataques de Longo Alcance

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

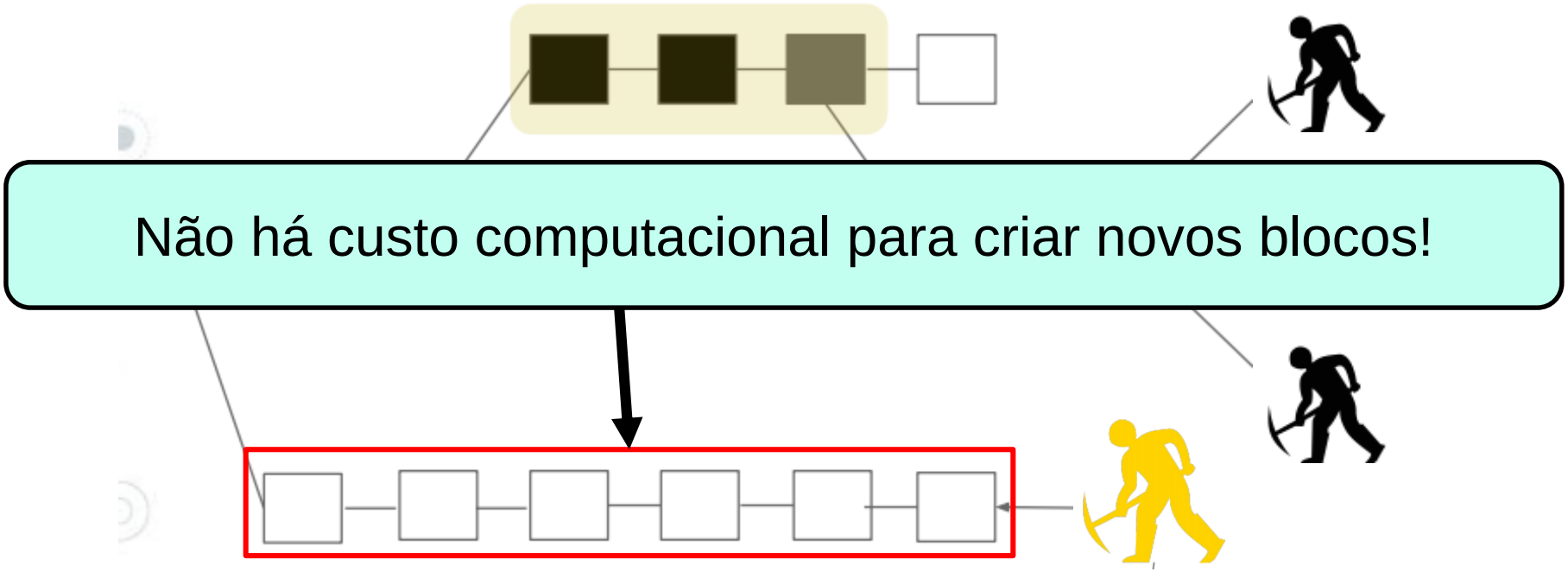
- No entanto, um atacante pode criar um bloco que aponta para outro muito no passado...



Prova de Posse – Ataques de Longo Alcance

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

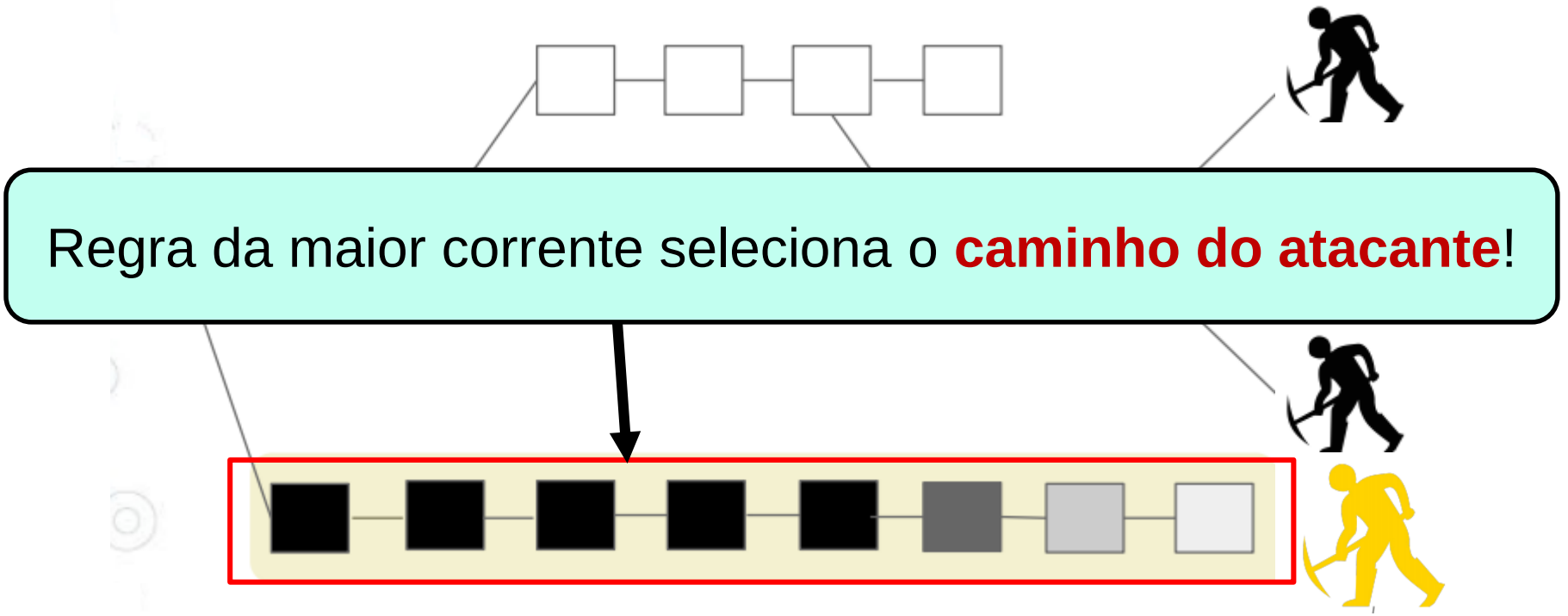
- ... e criar uma bifurcação intencional de vários blocos.



Prova de Posse – Ataques de Longo Alcance

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- Caso vença uma rodada de consenso, o atacante pode apontar para o caminho com todos os blocos que criou!
  - É vantajoso mesmo que possua apenas 1% de chance!

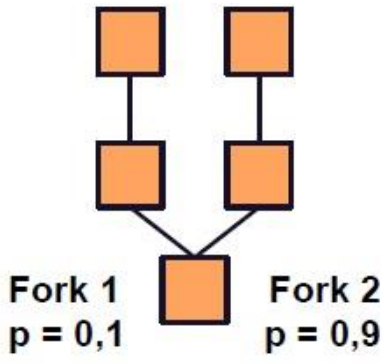


Prova de Posse – Nada a Perder

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- Ocorre quando existem dois caminhos conflitantes
  - Um validador deve escolher entre os caminhos

No vote :
EV = 0

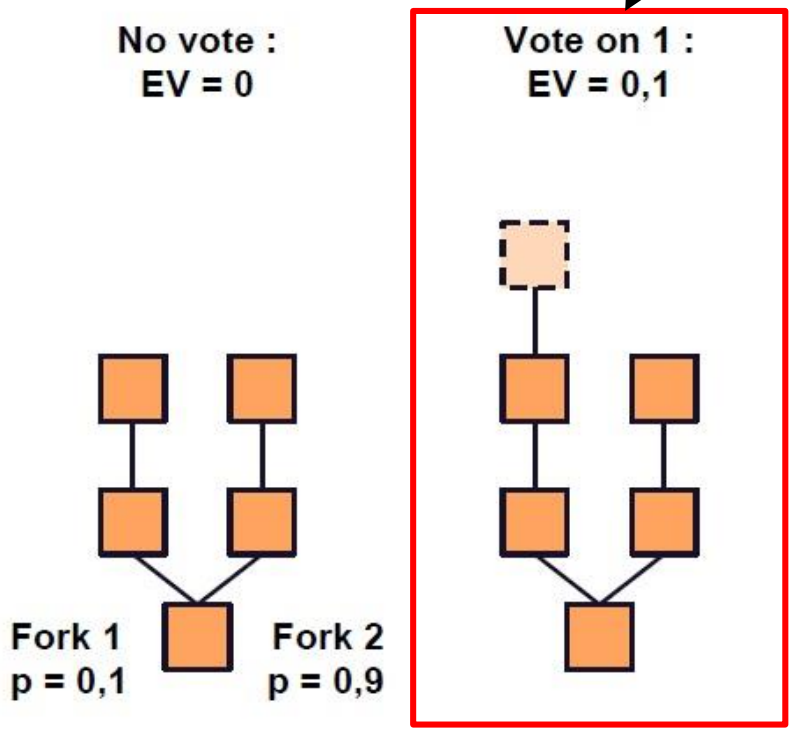


Prova de Posse – Nada a Perder

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

Voto no caminho 1 → probabilidade de ganhar recompensa: 0,1*stake (**p**)

- Um validador **p** deve escolher entre os caminhos

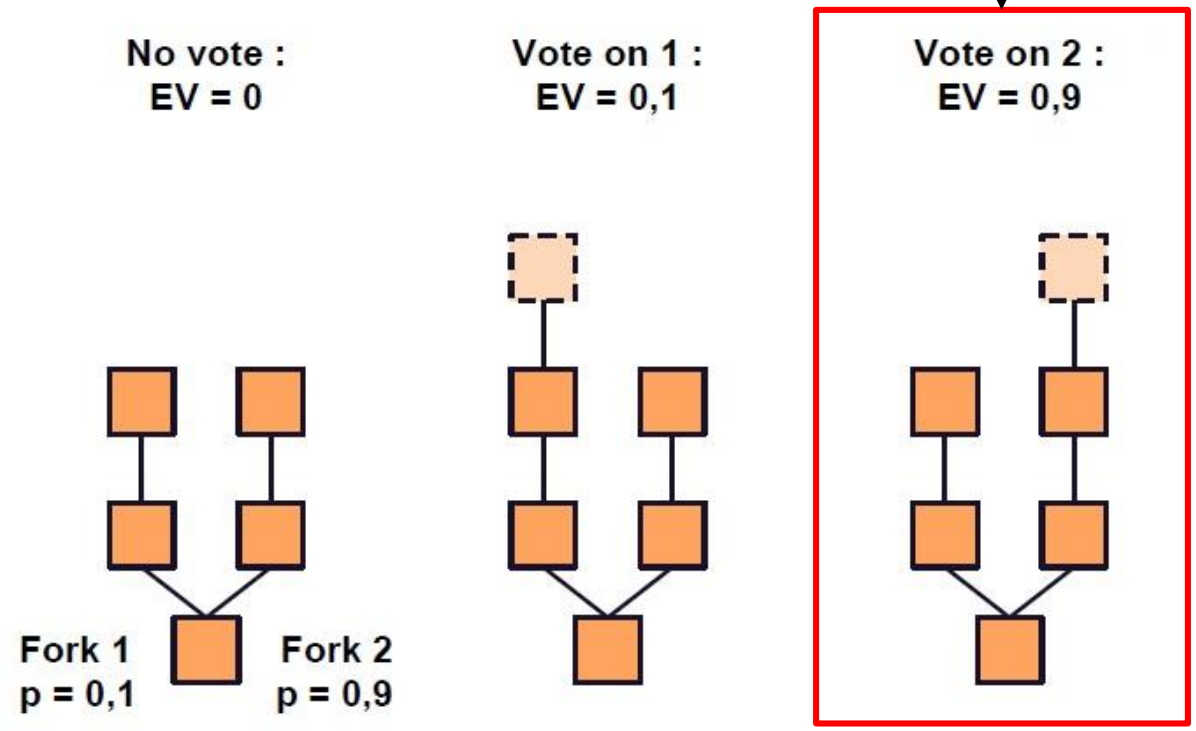


Prova de Posse – Nada a Perder

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

Voto no caminho 2 → probabilidade de ganhar recompensa: 0,9*stake (**p**)

- Um validador **p** deve escolher entre os caminhos

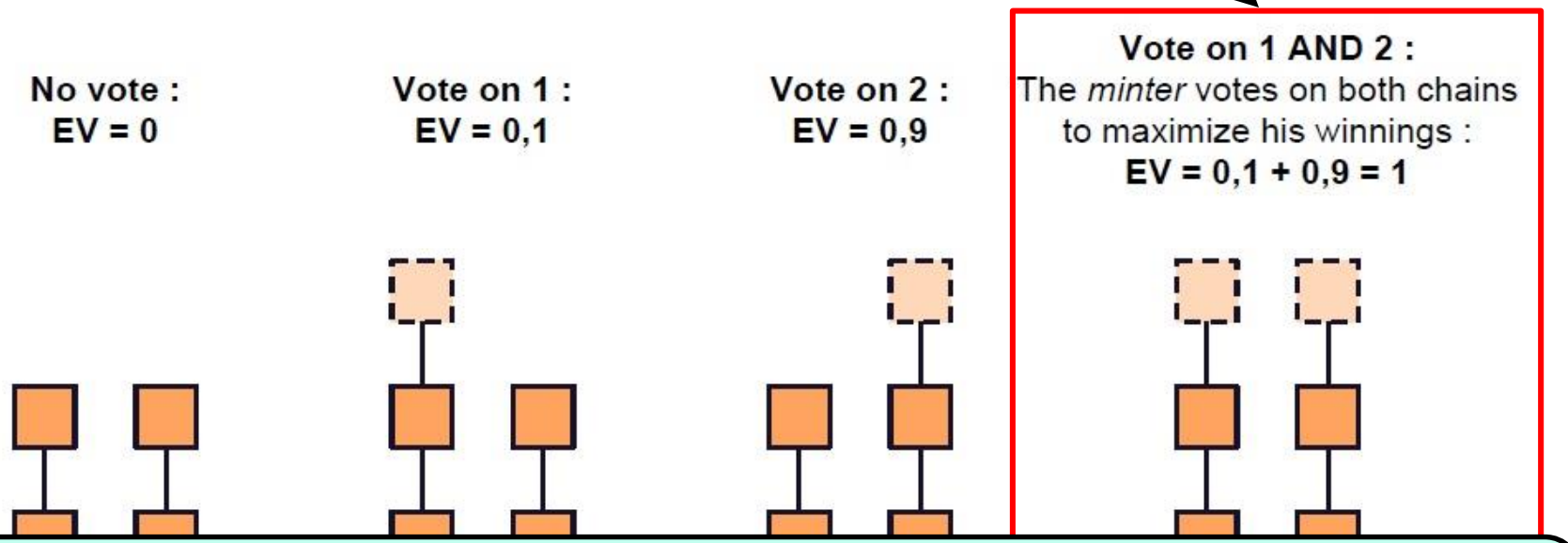


Prova de Posse – Nada a Perder

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

Voto nos dois caminhos → probabilidade de ganhar recompensa: 0,1*stake(**p**) + 0,9*stake (**p**) = stake(**p**)

– Um validador **p** deve escolher entre os caminhos



Decisão que maximiza a **probabilidade de ganhar** é sempre votar em **todos os caminhos** possíveis!

Prova de Posse – Nada a Perder

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- Problema de ”nada a perder”: Participantes podem validar caminhos conflitantes **sem gastar recursos**
  - Na prova de trabalho, os recursos seriam divididos
- Se todos os participantes agirem racionalmente, pode **não haver consenso** mesmo sem a presença de atacantes
- Basta que o atacante possua mais recursos que os **participantes altruístas**
  - “Ataque do 1%” (*1% attack*)
  - “Ataque da propina” (*Bribering attack*)

Prova de Posse – Chain-based PoS

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- Primeiras implementações de prova de posse
  - Chain-based PoS
    - Algoritmo similar à prova de trabalho
    - Dificuldade do participante diminui de acordo com sua posse
  - Peercoin (2012), NXT (2013) e Blackcoin (2013)

peercoin

NXT GENERATION OF CRYPTOCURRENCY

BLACKCOIN

Prova de Posse – Chain-based PoS

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- Algoritmo de geração de blocos (Peercoin e NXT)

```
/* PoS-based block generation                    */
Function BlockGen():
    Pack up transactions and prepare a block header
    Set up a clock (whose tick interval is a constant) and
    check for the following condition per clock tick:
        Hash(𝒞|clock_time) < target × stake_value
    wherein more preceding zero bits in target indicates
    a higher mining difficulty per unit of stake value;
    return new block;
end
```

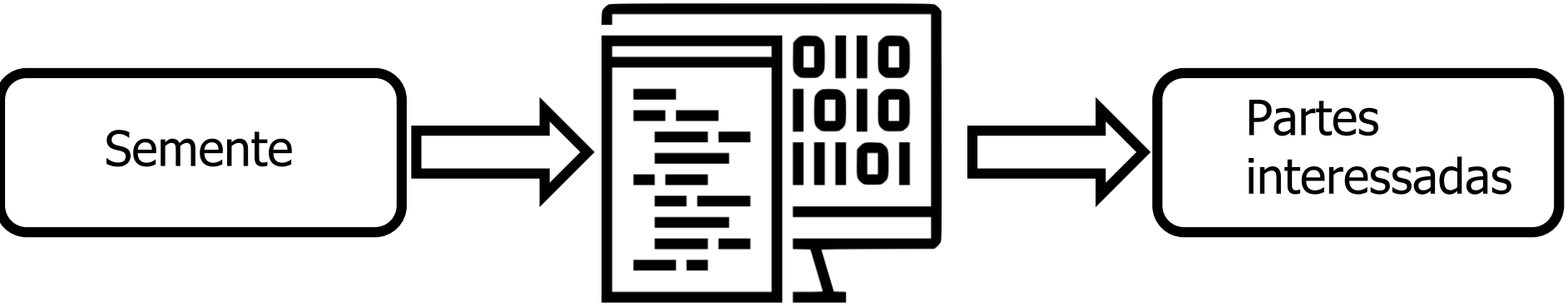
Dificuldade diminui proporcionalmente à posse do participante

Fonte: Xiao, Yang, et al. "A Survey of Distributed Consensus Protocols for Blockchain Networks." arXiv preprint arXiv:1904.04098 (2019).

Prova de Posse – Chain-based PoS

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- Prova de Atividade
  - Alternativa proposta por Bentov (2014)
  - Algoritmo Siga-o-Satoshi (*Follow-the-Satoshi* – FTS)
    - Sorteia uma moeda a partir de uma semente aleatória
    - Seleciona o dono da moeda como proponente do bloco
  - Participantes devem sempre gerar sementes



Siga-o-Satoshi (*Follow-the-Satoshi)*

Prova de Posse – Chain-based PoS

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- Vantagens
  - Menos custos energéticos que a prova de trabalho

- Desvantagens
  - Ainda gasta recursos computacionais
  - Vulnerável a economia de escala
  - Não provê **finalidade**
  - ... as mesmas desvantagens do PoW!

- Soluciona o nada a perder e o ataque de longa distância com **os mesmos mecanismos** do PoW

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

Protocolos Baseados em Consenso Determinístico – Tolerância a Falhas Desastrosas

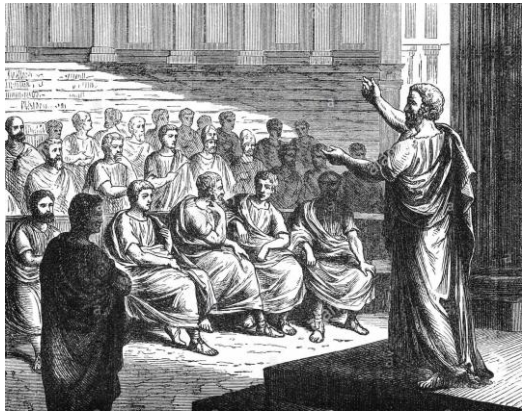
Paxos: The Part-Time Parliament

Leslie Lamport

Digital Equipment Corporation

O Protocolo Paxos

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO



- Baseado no parlamento da ilha de Paxos, Grécia
  - Cada legislador possui um livro registo onde:
    - Registra os decretos em uma sequência numerada
    - Registros escritos com tinta permanente, impossível modificar registros antigos
- Proposto por Leslie Lamport
- Uma solução para a replicação de máquina de estados

O Protocolo Paxos

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- Protocolo de consenso tolerante a falhas de parada
  - *Crash Fault Tolerant* (CFT)
    - No parlamento, o legislador não estar presente na seção é equivalente à parada de uma máquina
- Usado por:
  - Google: **Chubby**, serviço de bloqueio distribuido baseado em paxos
  - Yahoo: **Zookeper**, serviço de bloqueio distribuido baseado em paxos
  - Microsoft: tolerância a falhas de aglomerado (*Windows Server Failover Clustering*)
- Paxos é considerado um protocolo de difícil compreensão
  - Lamport publica o artigo "Paxos Made Simple"

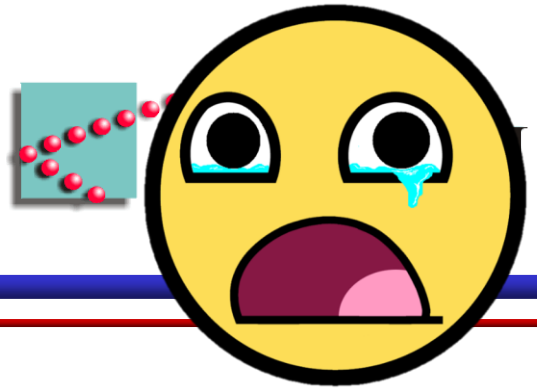
Considerações do Protocolo Paxos

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- Rede **parcialmente síncrona**
- Não existem falhas bizantinas
- Comunicações dos participantes através de mensagens
  - Entrega de mensagens pode demorar arbritariamente
  - Mensagens podem ser duplicadas
  - Mensagens podem ser perdidas, mas nunca corrompidas

- Participantes podem falhar por parada (*crash*)
  - Pode ou não reiniciar e voltar a participar da rede

Fases do Protocolo Paxos

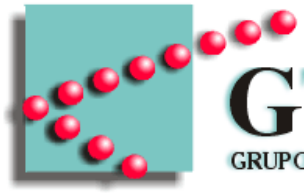


- O Paxos divide o consenso em duas fases

Continua **difícil de entender**... Poxa Lamport!

Solução: Protocolo de consenso **Raft**

- se 2a: **aceitação** (*accept*)
  - Quando o proponente recebe promessas do valor N da maioria da rede, envia pedidos de aceitação
  - 2b: **aceitado** (*accepted*)
  - Participantes aceitam a proposta e notificam a rede



GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

Raft: In search of an understandable consensus algorithm.

Diego Ongaro, John Ousterhout

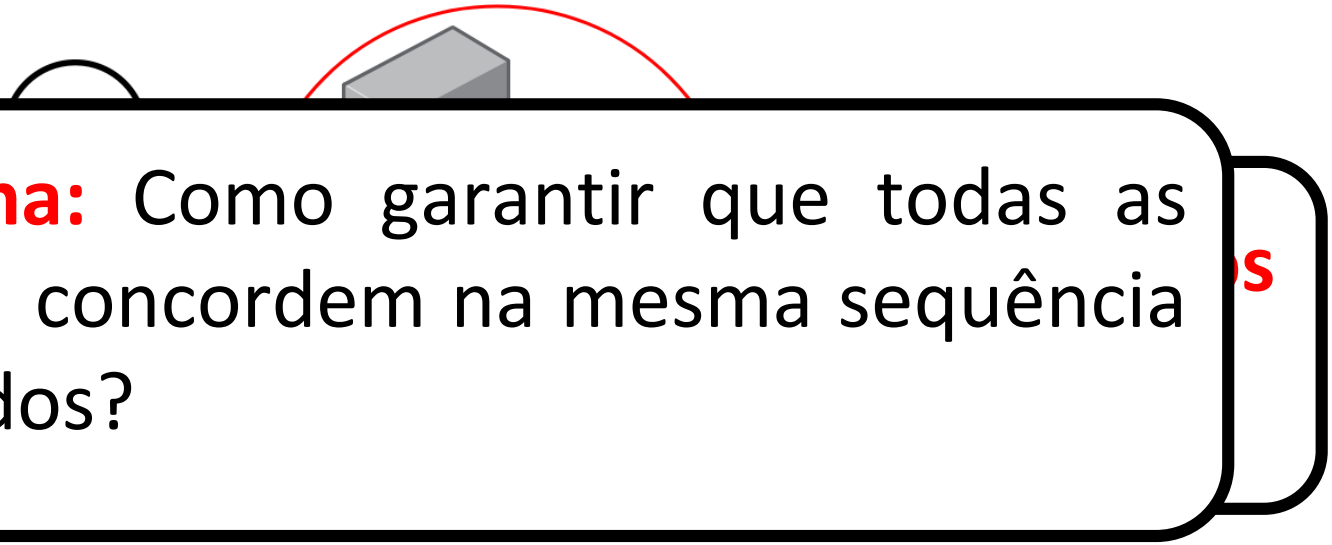
Stanford University, California

Raft: Motivação

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- Como garantir uma disponibilidade de 99,999% com constantes falhas?

Raft: Replicação de Máquina de Estados



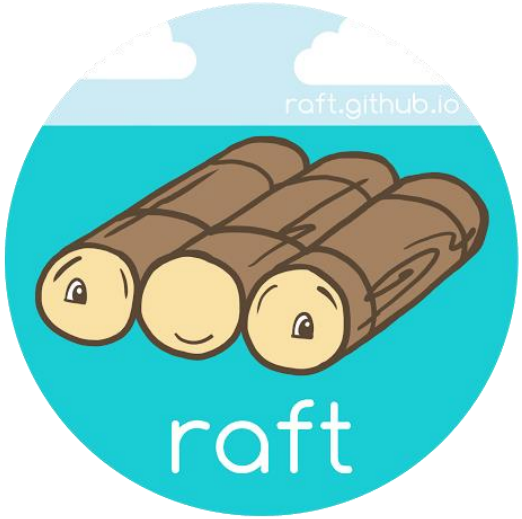
Problema: Como garantir que todas as réplicas concordem na mesma sequência de estados?

- Tornar o servidor **determinístico**
  - A mesm… qualquer servidor
- Replicar o s…
- Garantir rép…
  - **Mesma sequência** de estados em todas as réplicas

Solução: Protocolo de consenso **Paxos**

Protocolo Raft

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO



- Baseado no protocolo Paxos
  - Raft é mais **compreensível** que o Paxos
  - Raft é tão eficiente quanto o Paxos
- Proposto por Diego Ongaro e John Ousterhout
- Consenso tolerante a falhas desastrosas (*crash*)
- Utiliza a Replicação de Máquina de Estados (RME)
- Divide o protocolo em 3 fases --> Facilita compreensão
- Mais de **100** implementações diferentes

Considerações do Protocolo Raft

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- Mesmas considerações do Paxos
  - Rede **parcialmente síncrona** com relógios defeituosos
  - Não existem falhas bizantinas
  - Comunicações dos participantes através de mensagens
    - Entrega de mensagens pode demorar arbritariamente
    - Mensagens podem ser duplicadas
    - Mensagens podem ser perdidas, mas nunca corrompidas
  - Participantes podem falhar por parada (*crash*)
    - Pode ou não reiniciar e voltar a participar da rede

Raft: Estados dos Participantes

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- **Líder**
  - Responde às entradas do cliente
  - Replica às entradas nos seguidores
- **Seguidor**
  - Responde a pedidos do líder e do candidato
- **Candidato**
  - Seguidor inicia processo de eleição e torna-se um candidato
  - Possível novo líder caso receba a maioria dos votos

Principais Características do Protocolo Raft

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- **Líder forte**
  - realiza maior parte do trabalho
    - Emite todas as atualizações do registro
- **Eleição do líder**
  - Iniciada pelo candidato
  - Uso de temporizadores de duração **aleatória** para indicar falha do líder atual e iniciar uma nova eleição
    - Limite aconselhado: 150 ~ 300 ms
- **Troca de participantes**
  - Proposta de um consenso com diferentes cenários
    - Maioria dos votos dos dois cenários (atual e após troca) é necessário para obtenção de consenso

Raft: Divisão do Tempo

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- Divisão do tempo em **mandatos**
  - Tempo de duração do mandato é indefinido
    - Termina apenas quando detectado uma falha do líder atual
  - Numerado com inteiros consecutivos
  - Início de eleição gera um novo mandato

- Temporizadores de duração aleatória
  - Aleatoriedade da duração garante uma baixa probabilidade da existência múltiplos candidatos
  - Cada participante possui o seu próprio temporizador
  - Utilizado para detectar falhas de:
    - Parada do líder
    - Envio de mensagens

Raft: Divisão do Consenso em 3 Fases

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

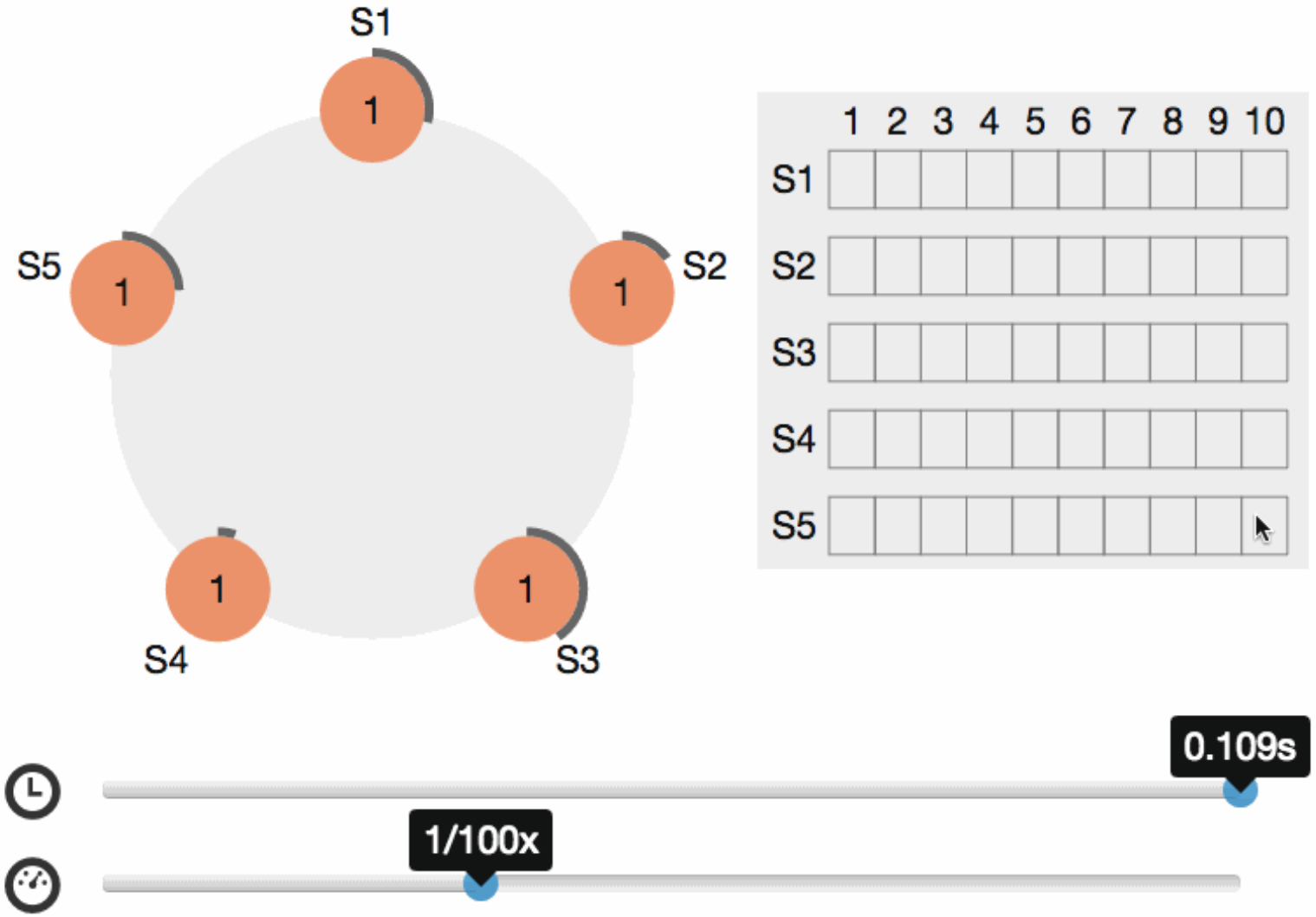
- **Eleição do líder**
  - Primeiro seguidor que zerar seu temporizador sem resposta do líder iniciará a nova eleição e se tornará o candidato
  - Candidato pede votos aos seguidores por RPC
    - Eleição falha ao detectar seguidor com número de mandato superior ao próprio
  - Eleição termina ao receber votos de metade dos seguidores
- **Replicação do registro**
  - Replicação comandada pelo líder
  - Líder decide quando armazenar o comando do cliente no livro de registro
  - Líder decide quando aplicar o comando na máquina de estados

Raft: Divisão do Consenso em 3 Fases

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

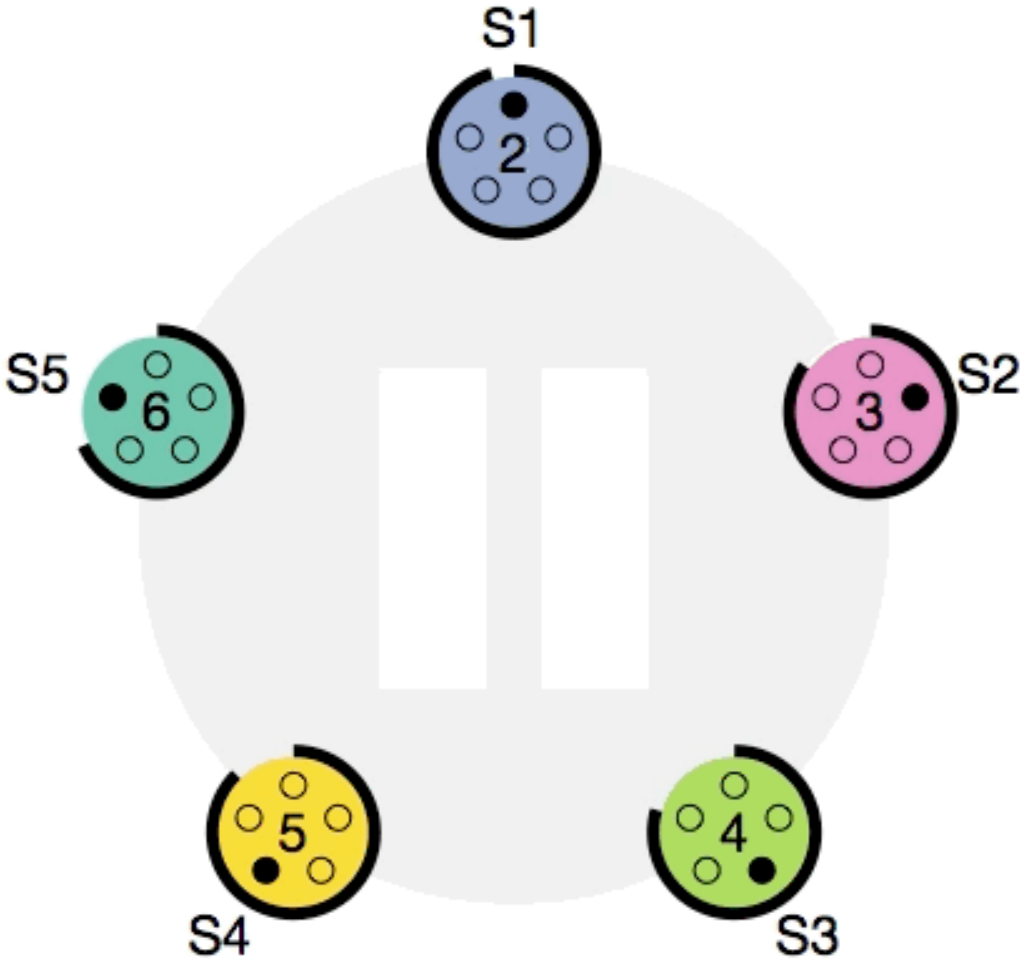
- **Segurança do registro (*safety*)**
  - Livro de registro do líder é **absoluto**
  - Divergência de registros
    - Registro pertencente ao maior mandato permanece
  - Parada de seguidores e candidatos
    - Solucionado enviando o pedido de voto ou ordem por RPC repetidamente

Funcionamento do Protocolo Raft



Funcionamento do Protocolo Raft

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO



GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

Protocolos Híbridos
-
Prova de Posse e Tolerante a Falhas Bizantinas (PoS-BFT)

Casper the Friendly Finality Gadget

Vitalik Buterin and Virgil Griffith

Ethereum Foundation

Casper the Friendly Finality Gadget (Casper FFG)

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- Proposto por Vitalik Buterin em 2017
- Primeiro passo da transição do Ethereum para o PoS
- Híbrido PoW/PoS-BFT
  - PoW para propor blocos
  - PoS-BFT para prover **finalidade**
    - Escolher o ramo correto da bifurcação

ethereum

ETHEREUM
casper

Casper the Friendly Finality Gadget (Casper FFG)

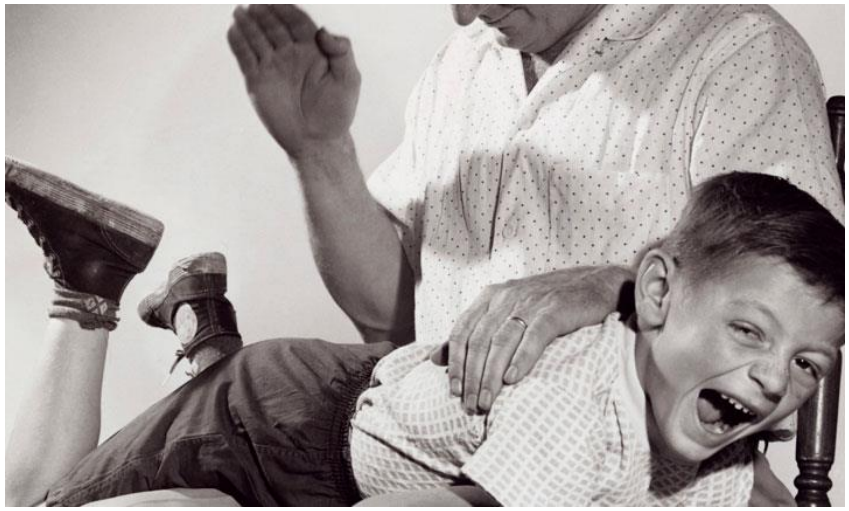
GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- Para tornar-se um validador, basta **identificar-se** e **depositar recursos** no contrato inteligente do Casper FFG

Casper the Friendly Finality Gadget (Casper FFG)

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- Por que **identificar** validadores e exigir um **depósito**?
  - Para **punir malfeitores**, evitando o problema nada a perder



- Um malfeitor tem seu depósito retirado através de mecanismos de punição
  - E quem denunciar ganha uma parcela!

Casper FFG – Caso Padrão

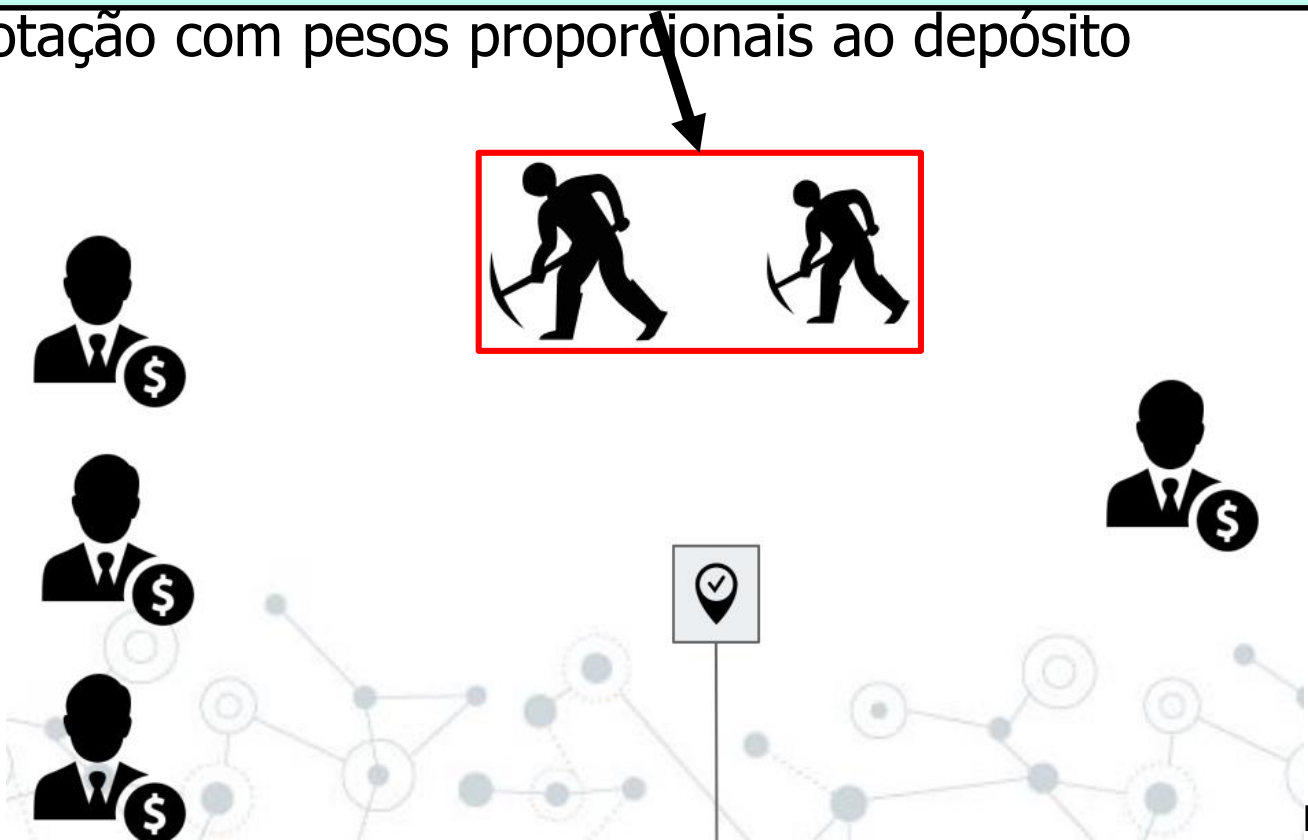
GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- Finalidade baseada em *checkpoints*
  - Cada bloco múltiplo de 50 é um *checkpoint*
  - *Checkpoints* são validados através de PoS
  - Votação com pesos proporcionais ao depósito

Casper FFG – Caso Padrão

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

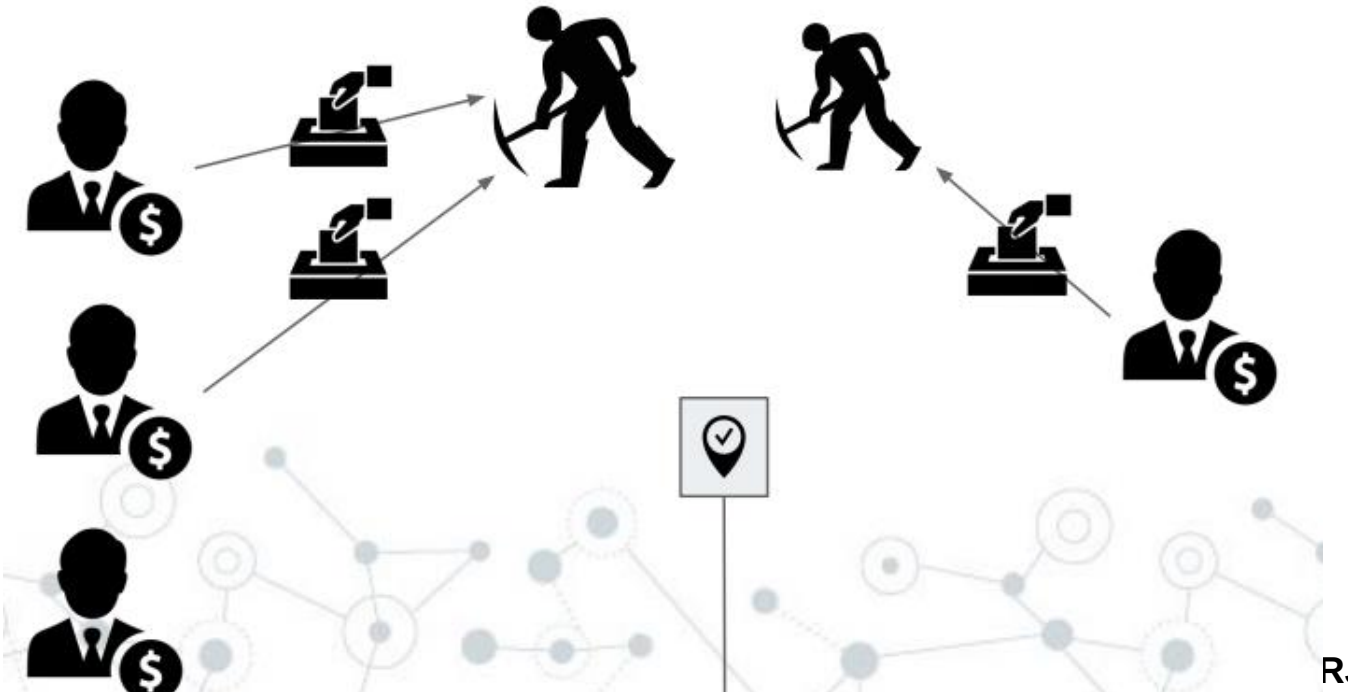
- Cada bloco múltiplo de 50 é um *checkpoint*
  - Mineradores participam do protocolo propondo blocos
  - Votação com pesos proporcionais ao depósito



Casper FFG – Caso Padrão

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

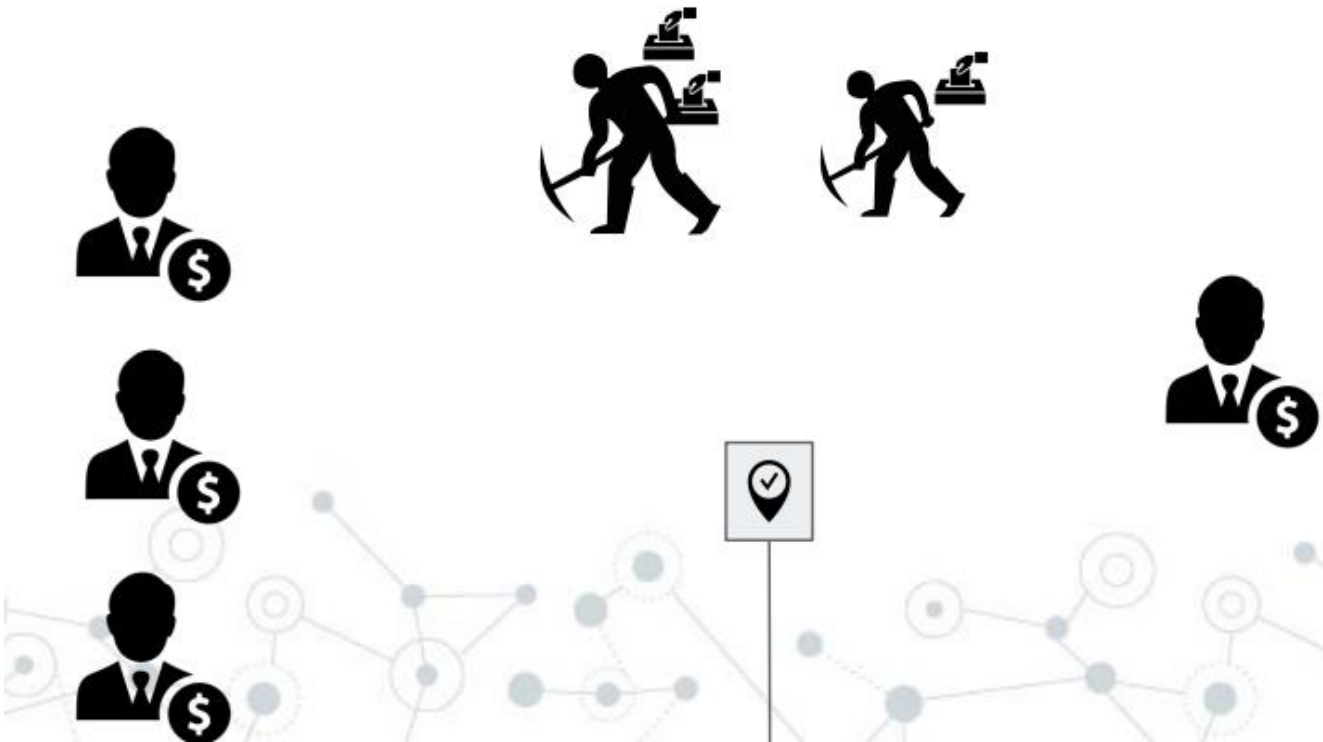
- Validadores enviam seus votos aos mineradores
  - Votos são **assinados** e possuem **pesos associados**
  - Votos comprovam que o validador **reconhece o checkpoint**



Casper FFG – Caso Padrão

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

– Mineradores recebem os votos e os inserem no bloco minerado

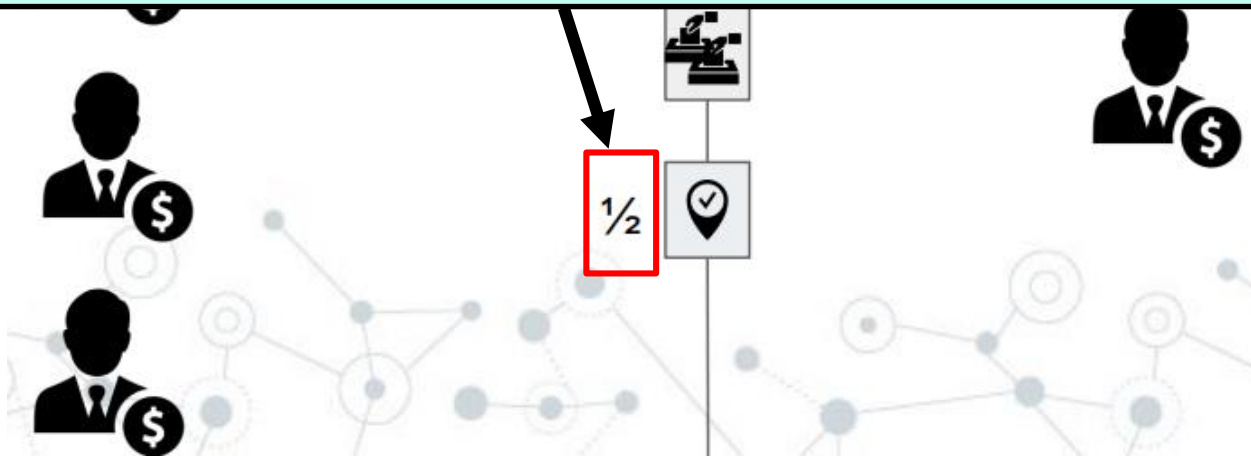


Casper FFG – Caso Padrão

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

– Mineradores vencedores propõem um bloco contendo todos os votos recebidos

Parcela de validadores que reconhecem o checkpoint aumenta

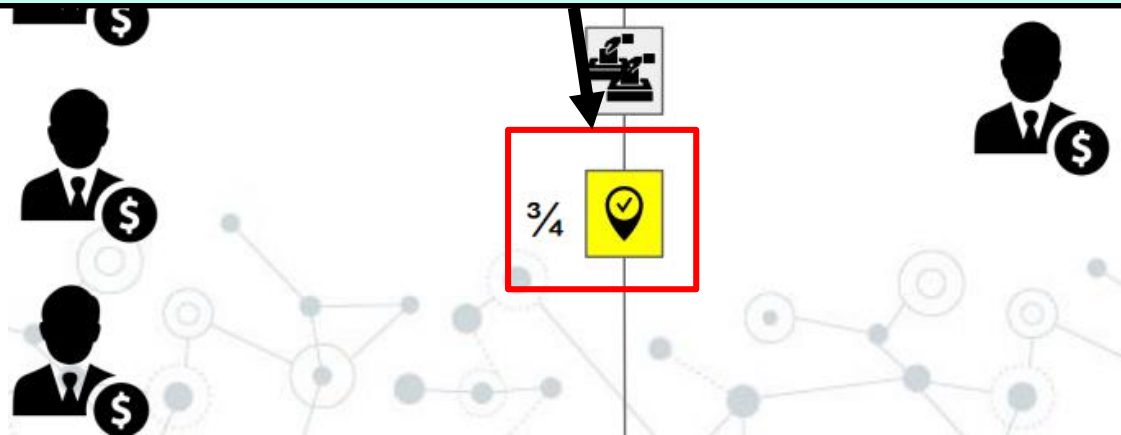


Casper FFG – Caso Padrão

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

– Mineradores vencedores propõem um bloco contendo todos os votos recebidos

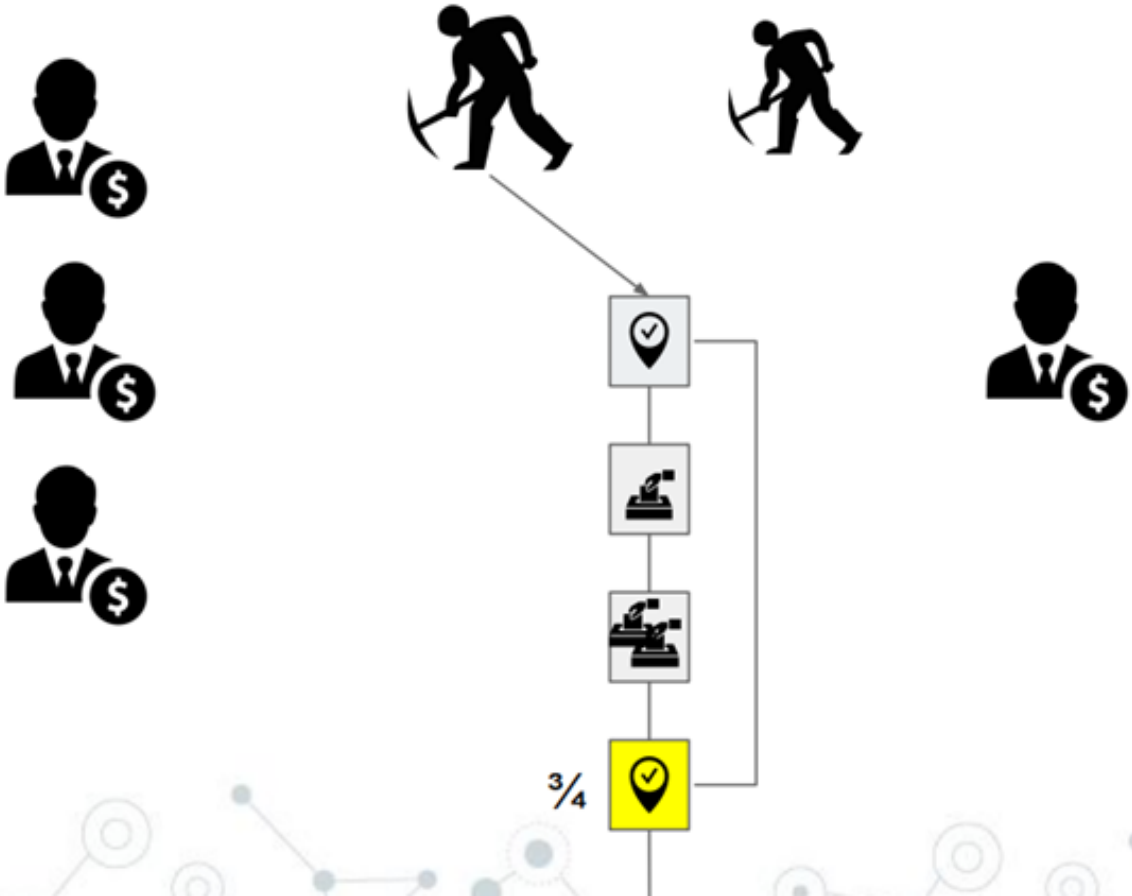
Quando possuir mais de 2/3 dos votos registrados, o checkpoint é **reconhecido** pelo sistema (porém ainda **não confirmado**)



Casper FFG – Caso Padrão

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

– Após 50 blocos, surge um novo checkpoint e uma nova votação



Casper FFG – Caso Padrão

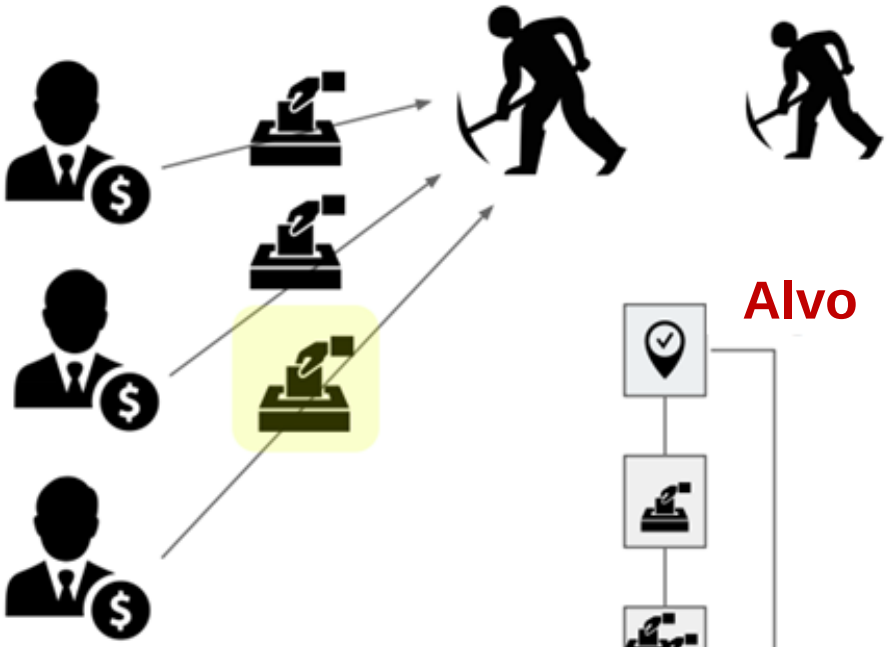
GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

– O período entre dois checkpoints é uma **época**

Casper FFG – Caso Padrão

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

– Os novos votos **reconhecem** o novo checkpoint (alvo) e **confirmam** o checkpoint antigo (fonte)



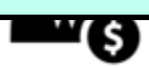
Voto:
[id_validador,
hash_alvo,
época_alvo,
época_fonte]

Casper FFG – Caso Padrão

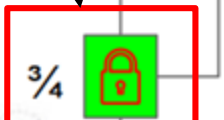
GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

– Quando o checkpoint alvo é reconhecido, cria-se um **enlace de supermaioria** entre os checkpoints

Quando possuir mais de 2/3 dos votos como **fonte**, o checkpoint é **confirmado** pelo sistema e sua finalidade é garantida

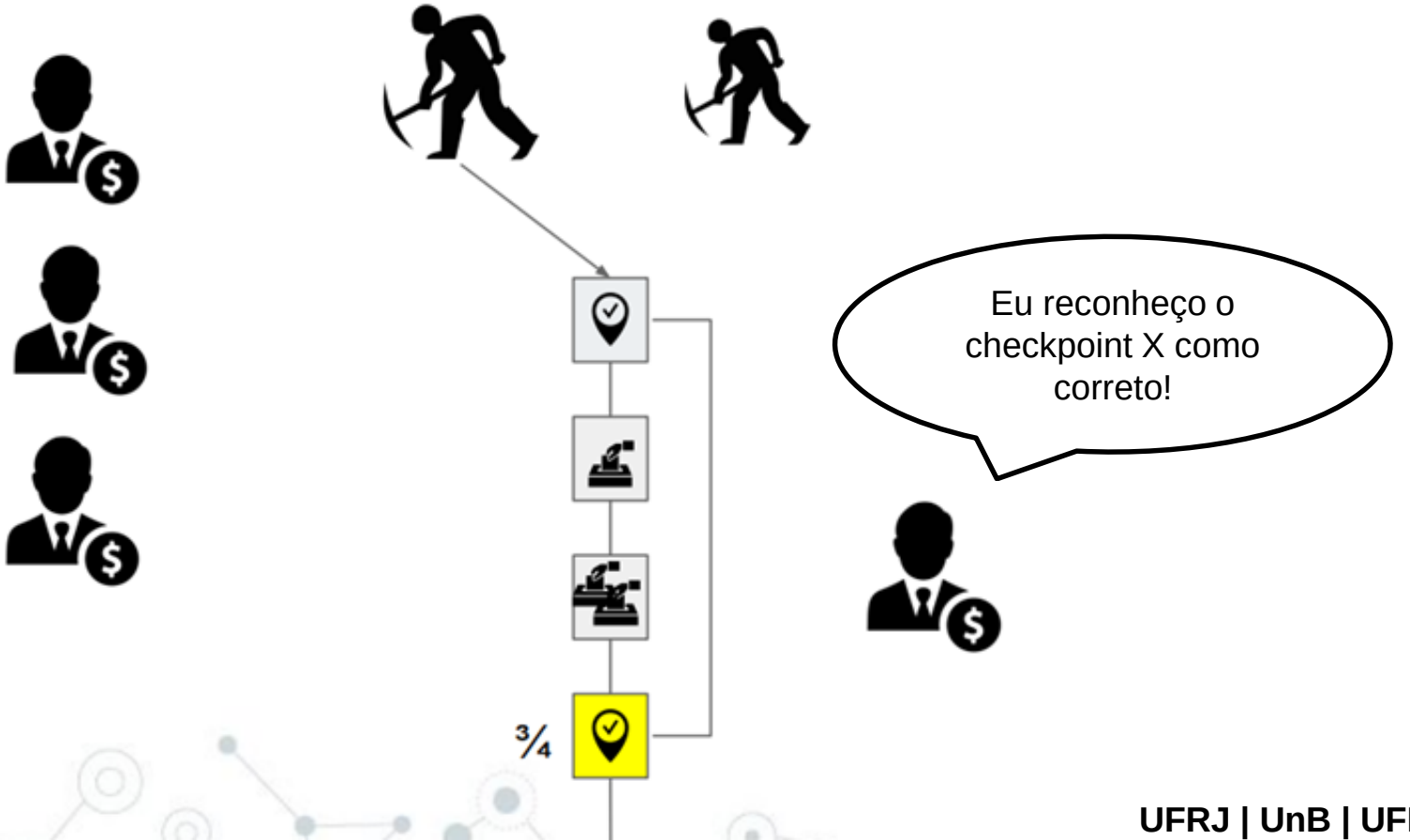


(> 2/3 de votos p/ ambos os checkpoints)



Casper FFG – Caso Padrão

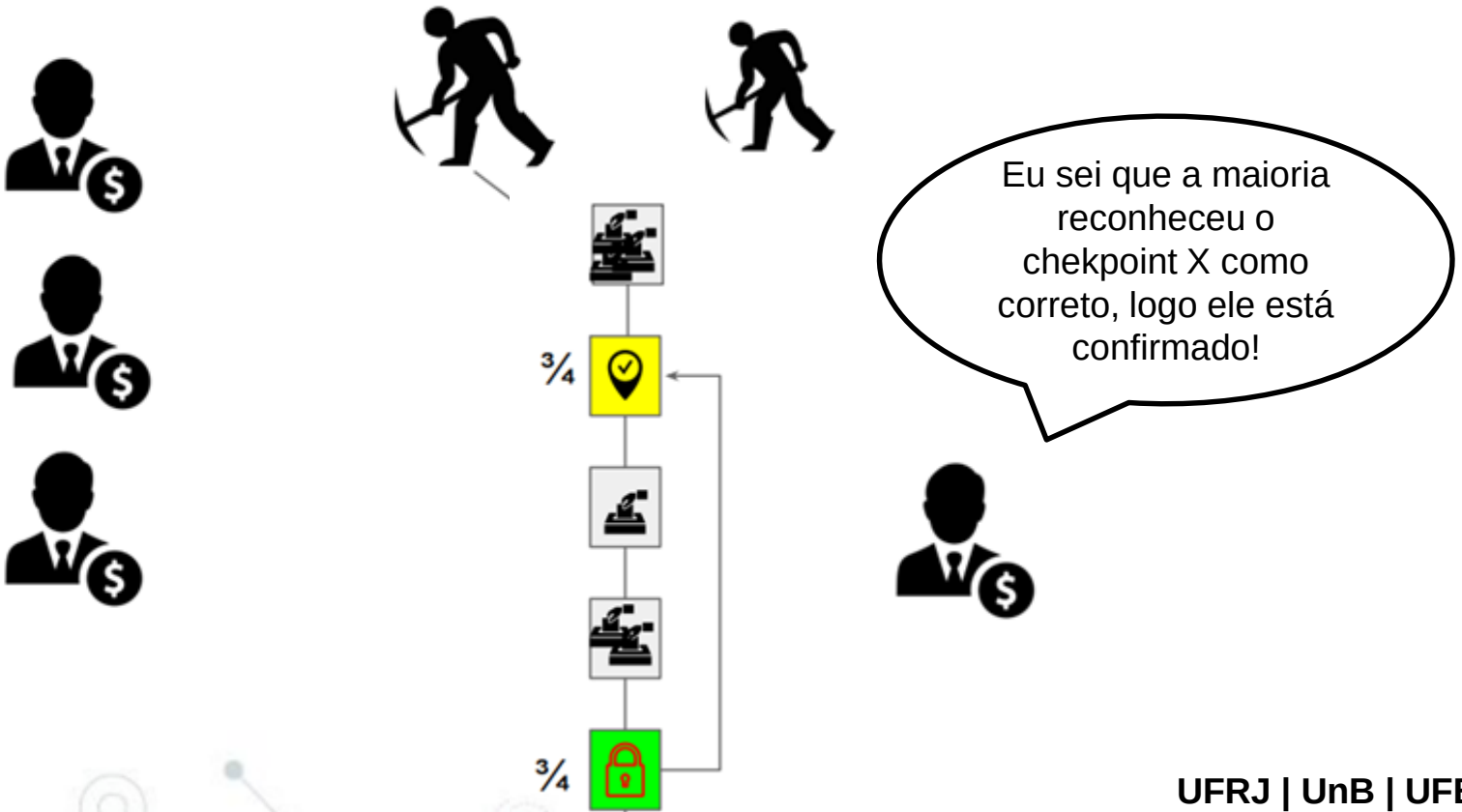
- Mas por que dois passos?
  - Passo 1: declarar que o checkpoint é correto



Casper FFG – Caso Padrão

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- Mas por que dois passos?
  - Passo 2: confirmar que os outros validadores reconheceram o checkpoint



Casper FFG – Mecanismo de Punição

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- Condições de punição (*Slashing Conditions*)
  - 1. Votar em dois caminhos distintos
  - 2. Votar em um caminho contido em outro

- Participantes podem **denunciar malfeitores** e receber parte de seus depósitos como **recompensa**

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

Ouroboros: A Provably Secure Proof-of-Stake Blockchain Protocol

Aggelos Kiayias[1], Alexander Russell[2], Bernardo David[3], Roman Oliynykov[4]

[1] Univeristy of Edinburgh
[2] University of Connecticut
[3] Aarhus University
[4] IOHK

O Protocolo Ouroboros

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- Baseado na prova de participação
- Desenvolvido por Aggelos Kiayias

O Protocolo Ouroboros

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- Baseado na prova de participação
- Desenvolvido por Aggelos Kiayias
- **Origem do nome**
  - Rodada atual depende da rodada anterior

O Protocolo Ouroboros

- Baseado na prova de participação
- Desenvolvido por Aggelos Kiayias
- Origem do nome
  - Rodada atual depende da rodada anterior
- **Utilizado pela plataforma Cardano**
  - Nome em homenagem ao matemático Girolamo Cardano
  - Executa a criptomoeda ADA
    - Nome em homenagem à matemática Ada Lovelace
    - 10ª maior criptomoeda em valor de mercado

CARDANO

Considerações do Protocolo Ouroboros

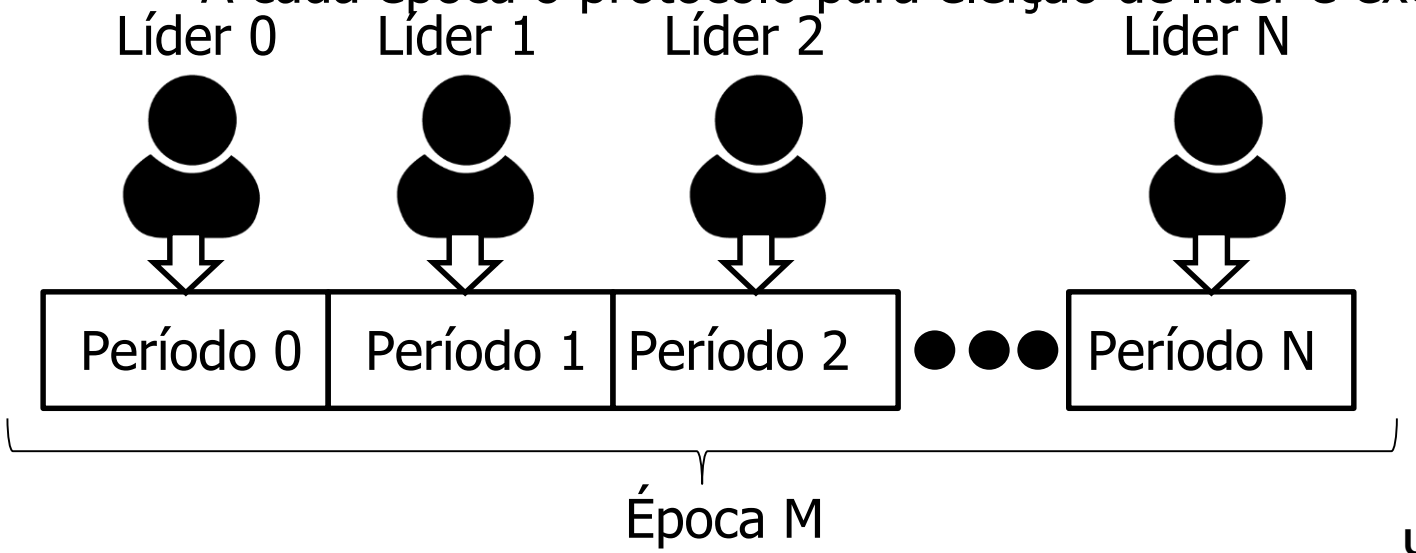
GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- Tempo é dividido em unidades discretas chamadas períodos
  - A cada período um líder eleito propõe o bloco
  - Cada bloco pode ser associado a um período
  - Um conjunto de períodos é chamado de época
    - A cada época o protocolo para eleição de líder é executado

Considerações do Protocolo Ouroboros

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- Tempo é dividido em unidades discretas chamadas períodos
  - A cada período um líder eleito propõe o bloco
  - Cada bloco pode ser associado a um período
  - Um conjunto de períodos é chamado de época
    - A cada época o protocolo para eleição de líder é executado

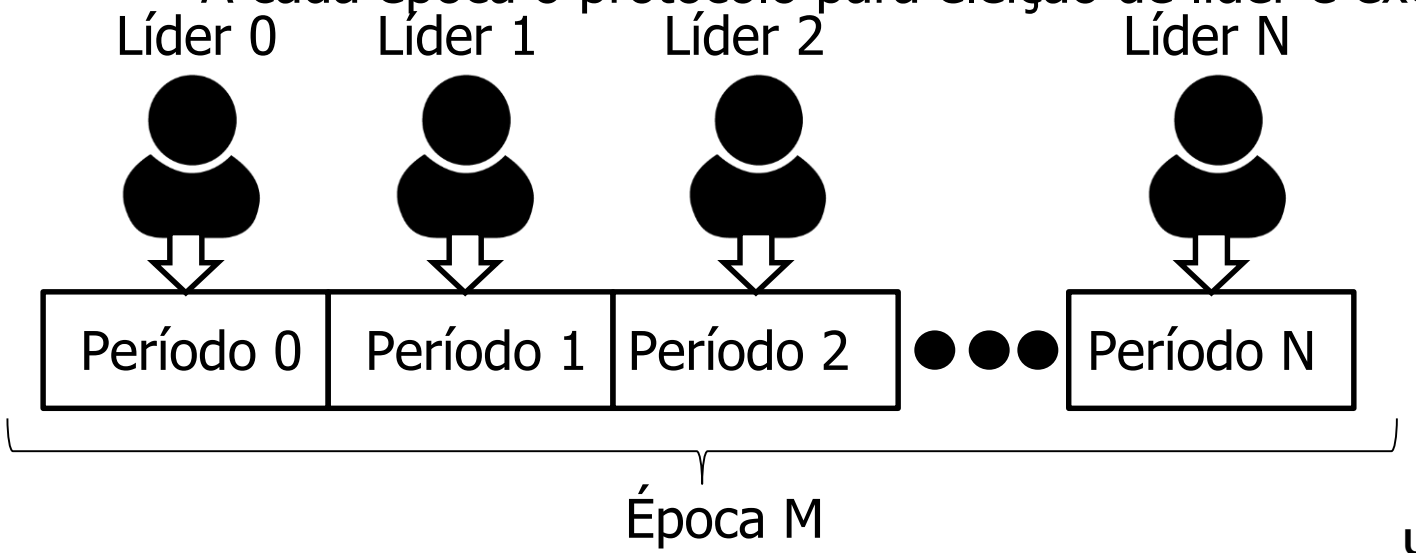


Considerações do Protocolo Ouroboros

- Tempo é dividido em unidades discretas chamadas períodos

Como inserir a aleatoriedade na escolha do líder?

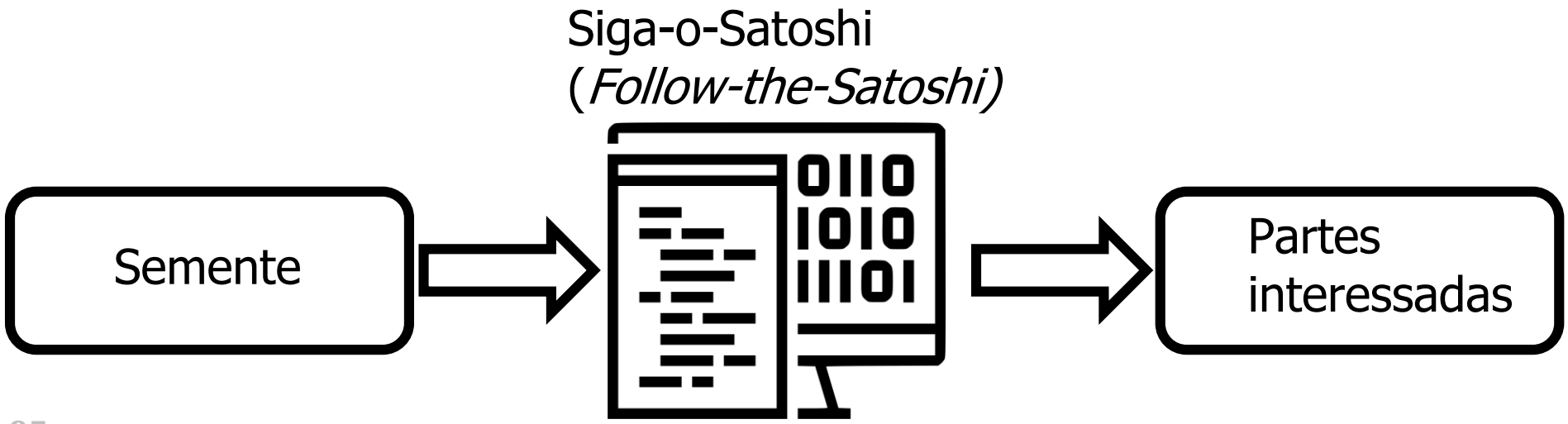
- A cada época o protocolo para eleição de líder é executado



Escolha do Líder no Ouroboros

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- Formação de uma semente (*seed*) aleatória
  - Gerada pelos participantes do consenso (eleitores)
  - A semente é usada pelo algoritmo Siga-o-Satoshi (*Follow-the-Satoshi* – FTS)
    - Algoritmo usa a semente para selecionar uma moeda. A saída do algoritmo é a parte interessada dona da moeda.



Escolha do Líder no Ouroboros

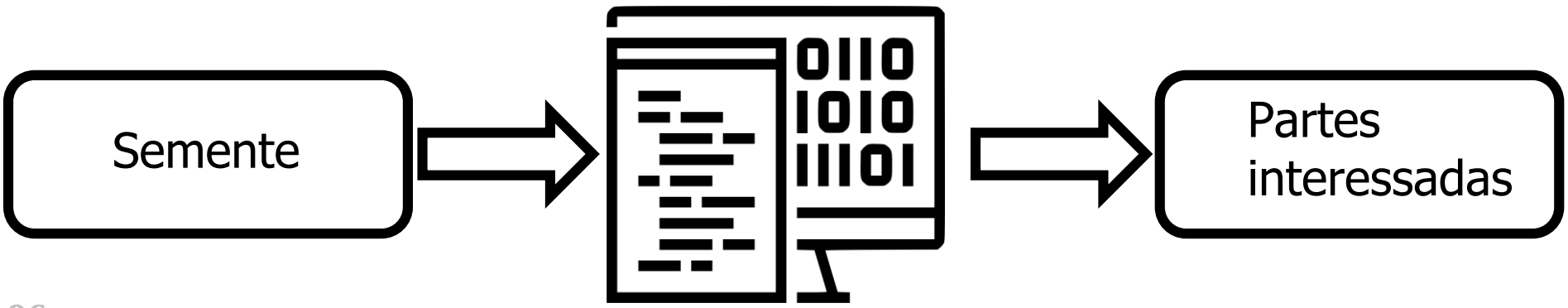
GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- Formação de uma semente (*seed*) aleatória

O algoritmo seleciona os líderes de períodos da próxima época e os eleitores da próxima época

do algoritmo é a parte interessada dona da moeda.

Siga-o-Satoshi
(*Follow-the-Satoshi)*



Geração da Semente (*seed*)

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- Cada eleitor gera uma string aleatória
  - As strings geradas são compartilhadas entre os eleitores
  - As strings geradas são combinadas para formar a semente

Como impedir que um participante influencie a geração da semente para se beneficiar?

Geração da Semente (*seed*)

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- **Solução:** Separar a geração da semente em duas fases

Como impedir que um participante influencie a geração da semente para se beneficiar?

Fases do Protocolo Ouroboros

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- Fase de efetivação (*Commitment*)

E se um participante abandonar o protocolo e não revelar a string gerada?

os

  - Cada eleitor envia parcelas da string de maneira secreta aos outros eleitores

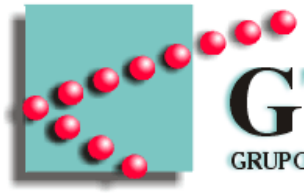
- Fase de revelação (*Reveal*)
  - Os eleitores revelam os segredos da fase de efetivação

Fases do Protocolo Ouroboros

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- O Ouroboros utiliza o esquema de compartilhamento de segredo verificável (*Verifiable Secret Sharing* – VSS)
  - Segredos não revelados podem ser recuperados se os participantes honestos compartilharem as parcelas recebidas durante a fase de efetivação

- Fase de recuperação
  - Recuperação dos segredos não-revelados
  - Geração da semente



GTA / UFRJ

GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

The Ripple Protocol Consensus Algorithm

David Schwartz, Noah Youngs e Arthur Britto

Casos de Uso de *Blockchain* e suas Semelhanças

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- Financiamento do comércio
  - Transações domésticas e internacionais → Envolve pagamentos!
- Pagamentos ruins **dificultam** todos esses casos de uso
- Empréstimo
  - Co
- Cont
  - C s!

Necessidade de um meio **eficaz** de realizar **pagamentos**

Cenário Atual dos Pagamentos Internacionais

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- Redes de pagamentos **desconectas**
  - Bitcoin e Ethereum não se comunicam com a Visa

Internet de Valor: mover ativos/dinheiro tão simples quanto transferir arquivos na internet

- **Altas taxas**

Ripple: Um Protocolo para Movimentar Dinheiro

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- Utilizado na criptomoeda **XRP**
  - **3ª** criptomoeda com maior valor de mercado*
  - Taxa de **1500** transações por segundo
  - Câmbio decentralizado
    - Dólar, Bitcoin, Ethereum...
- Consenso bizantino federado
  - Principais participantes são Bancos e instituições financeiras

Itaú

Santander

AMERICAN EXPRESS

*Fonte: https://coinmarketcap.com (Acessado em 4/7/2019)

Ripple: Um Protocolo para Movimentar Dinheiro

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- Ripple Labs propõe Ripple em 2012
  - Publicado como artigo em 2014
    - "The Ripple Protocol Consensus Algorithm."
      - Autores **David Schwartz**, Noah Youngs e Arthur Britto

Autores do *White Paper*: David Schwartz

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- Atual diretor técnico e criptógrafo chefe da Ripple
- Diretor técnico da WebMaster Incorporated (2001-2011)
- Engenharia elétrica na Universidade de Houston (1990)

Autores do *White Paper*: Noah Youngs

- Atual Cientista chefe de dados da Vitol Group
- Ph.D. em ciência da computação na Universidade de Nova Iorque
- Mestrado em matemática computacional na Universidade Stanford
- B.A. em economia, matemática e física na Universidade Columbia

Autores do *White Paper*: Arthur Britto

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

"Ele é excelente, uma das pessoas mais inteligentes que já conheci ... Ele é um oráculo."

Alan Safahi, ex-conselheiro da Ripple

Rastros de Arthur Britto

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

Agreement

With Arthur Britto

September 17, 2012

Chris Larsen, Jed McCaleb and Arthur Britto (the "Founders"), whom developed a distributed open source software platform for making and receiving payments and virtual currency ("Ripple") hereby agree, as of the date first written above (the "Effective Date"),as follows:

1. The Founders agree that 80% of all Ripple Credits shall be allocated to the Company, as determined by the percentage share of all existing Credits set forth in the ledger created, approved and adopted by the majority of Founders as the Official Ledger.

2. The Founders further agree that Arthur Britto shall receive 2% of all the Ripple Credits of the Official Ledger. The Founders acknowledge that these Credits have no value as of the Effective Date and that any compensation for work performed by Arthur Britto is provided in a separate consulting agreement with OpenCoin Inc. It is anticipated that a total of 100 billion credits shall be recorded on the Official Ledger. If the Official Ledger is revised, or any other ledger is created within 36 months of the date of this Agreement that sets forth a lesser percentage of Credits for Britto than the number set forth in the Official Ledger, Britto shall have the right to acquire additional credits at no cost to him, sufficient to bring his Credit Grant to 2% of the total number of credits.

3. The Founders further agree that the Ripple platform will be made available for distribution and licensed under a permissive Open Source license as soon as operationally optimal. It is agreed that Britto shall consent to "open source" his contribution to the Ripple platform at the same time that all other Ripple Founders do the same. In exchange for assigning to the Company his IP rights in Ripple, Britto shall have a lifetime, fully paid up license to develop apps or new functionalities on the Ripple platform.

Agreed:

Jed McCaleb

Chris Larsen

Arthur Britto

From: Arthur Britto <ahbritto@gmail.com>
Date: Tue, 19 Nov 2013 12:01:34 -0800
Message-ID: <CAFjXj6NUK7z-YpPF2ky5NARZg+RtMxWRcaDdV47OFy3HUMNk4w@mail.gmail.com>
To: Manu Sporny <msporny@digitalbazaar.com>
Cc: public-webpayments@w3.org

On Tue, Nov 19, 2013 at 11:48 AM, Manu Sporny <msporny@digitalbazaar.com>wrote:

> On 11/18/2013 04:08 PM, Evan Schwartz wrote:
> > The purpose of having XRP in the network as an asset is twofold:
> > first it helps bridge between different currencies and second it's
> > used for very small transaction fees that help prevent people from
> > spamming the network.
>
> Right, so let's split the requirement of XRP into two parts.
>
> 1) As a bridge currency
> 2) As a network attack prevention mechanism
>
> This email concerns the second item, which is as a network attack
> prevention mechanism.
>
> The most concerning downside about XRP that I can see is that there are
> a finite number of them (100B XRP, IIRC?). The probability that Ripple
> will last more than 30 years as a financial network is slim (not having
> done the math, but I'd imagine that ~3B transactions per year is being
> conservative wrt. the transaction volume that will happen over the
> network if it becomes popular). I do realize that many of these
> transactions may be rolled into one to

Funcionamento do Protocolo de Consenso Ripple

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- Protocolo dividido em **2 fases**

Consenso dura cerca de **4 segundos**

- Fase de **validação**
  - Incluir e efetivar o conjunto de transações no livro-razão

ram o

Rápida confirmação das transações!

Funcionamento do Protocolo de Consenso Ripple

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

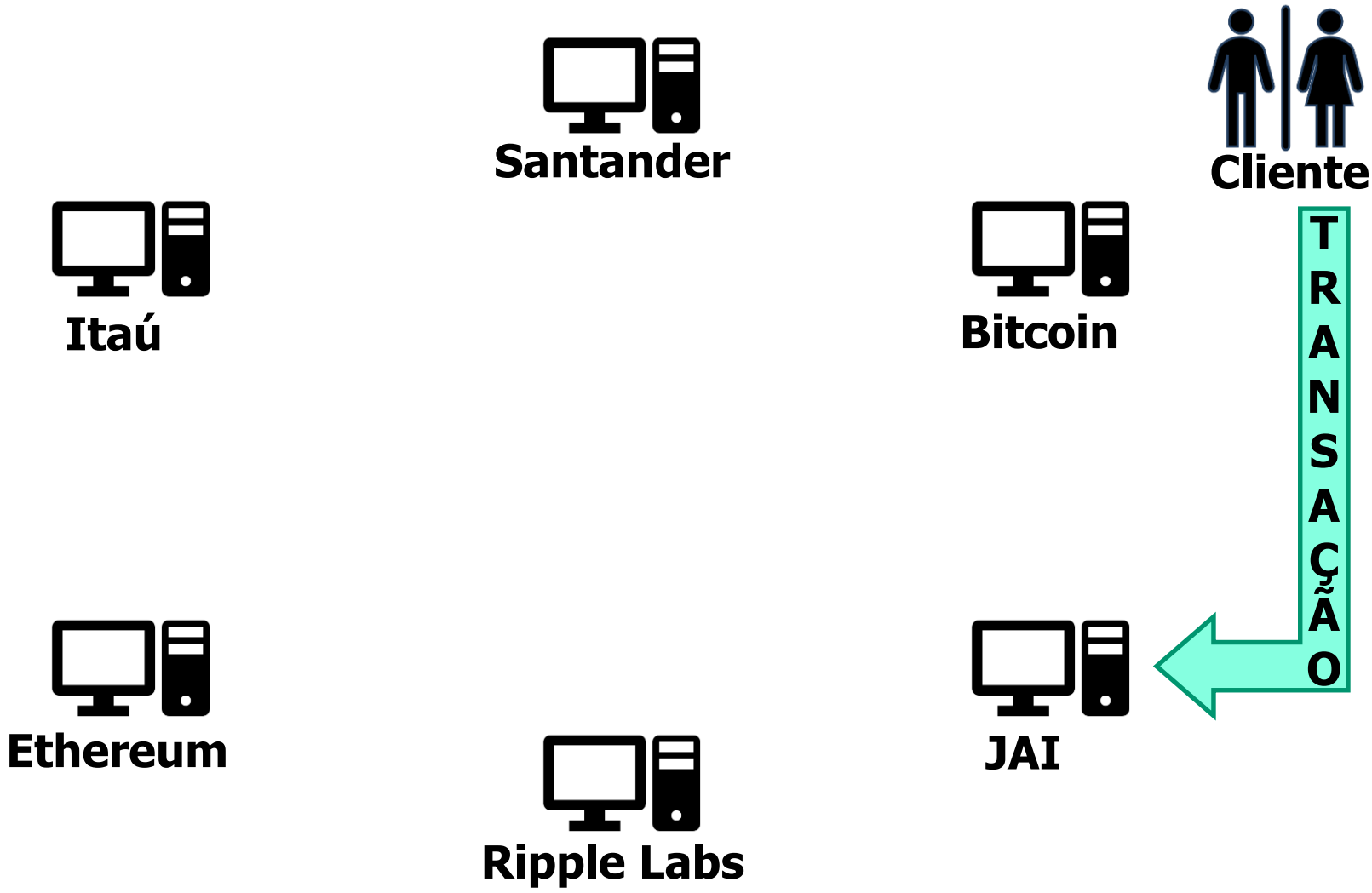
- Participantes do consenso são geralmente bancos, instituições financeiras e alguns nós do Ripple Labs
  - Participantes do consenso → **validadores**

- Validadores escolhem em quem confiam
  - Liberdade para compor a lista de validadores confiáveis
    - *Unique Node List* – **UNL**
      - Cada validador tem a sua lista
      - Estar na lista **não** significa que o validador é **honesto**!

Funcionamento da Fase de Deliberação

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

Funcionamento da Fase de Deliberação



Funcionamento da Fase de Deliberação

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

Itaú

Santander

Cliente

Bitcoin

UNL

Itaú
Santander
Ripple

Ethereum

JAI

Ripple Labs

Funcionamento da Fase de Deliberação

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

Santander

Cliente

Itaú

Bitcoin

UNL

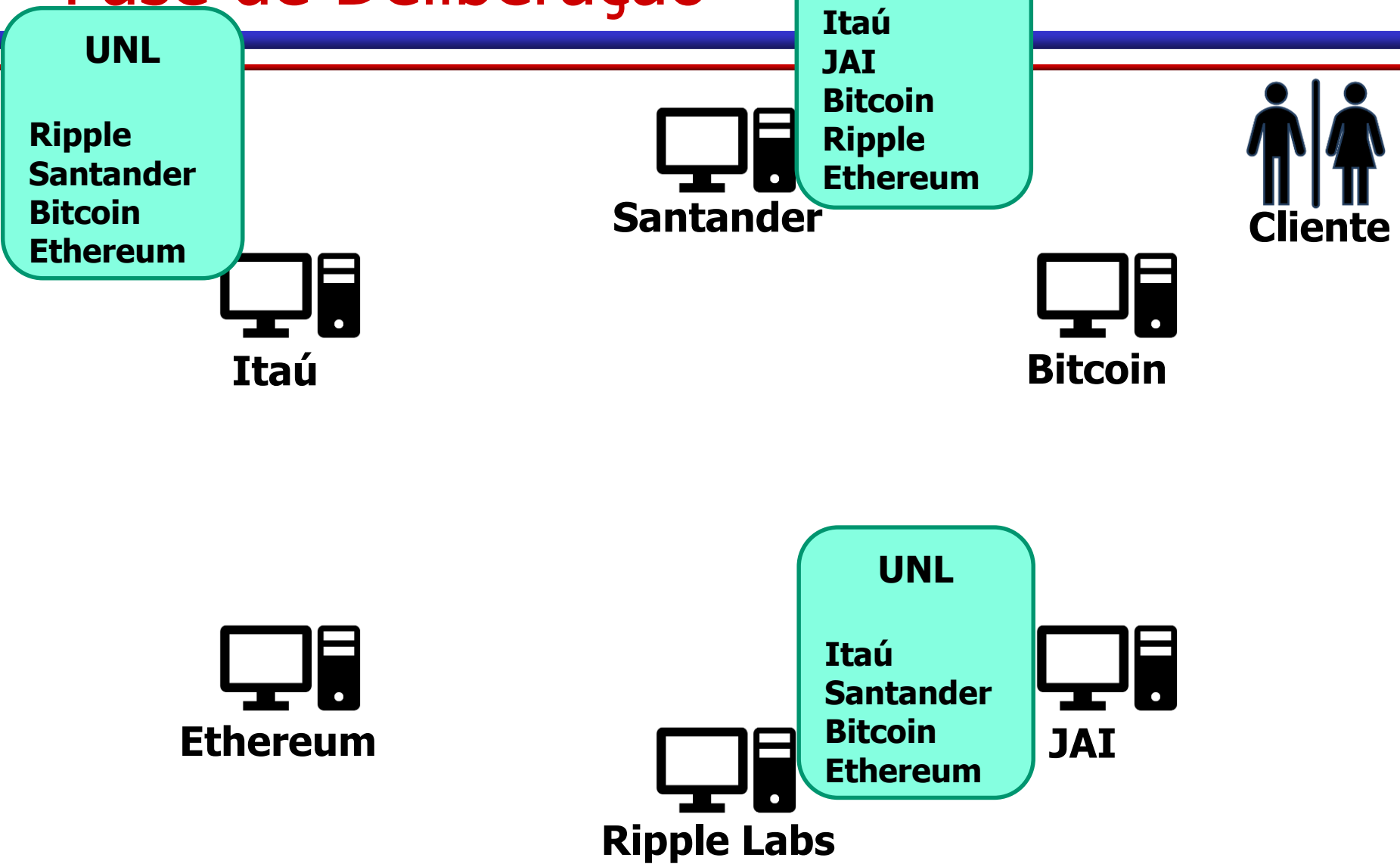
Itaú
Santander
Ripple

Ethereum

JAI

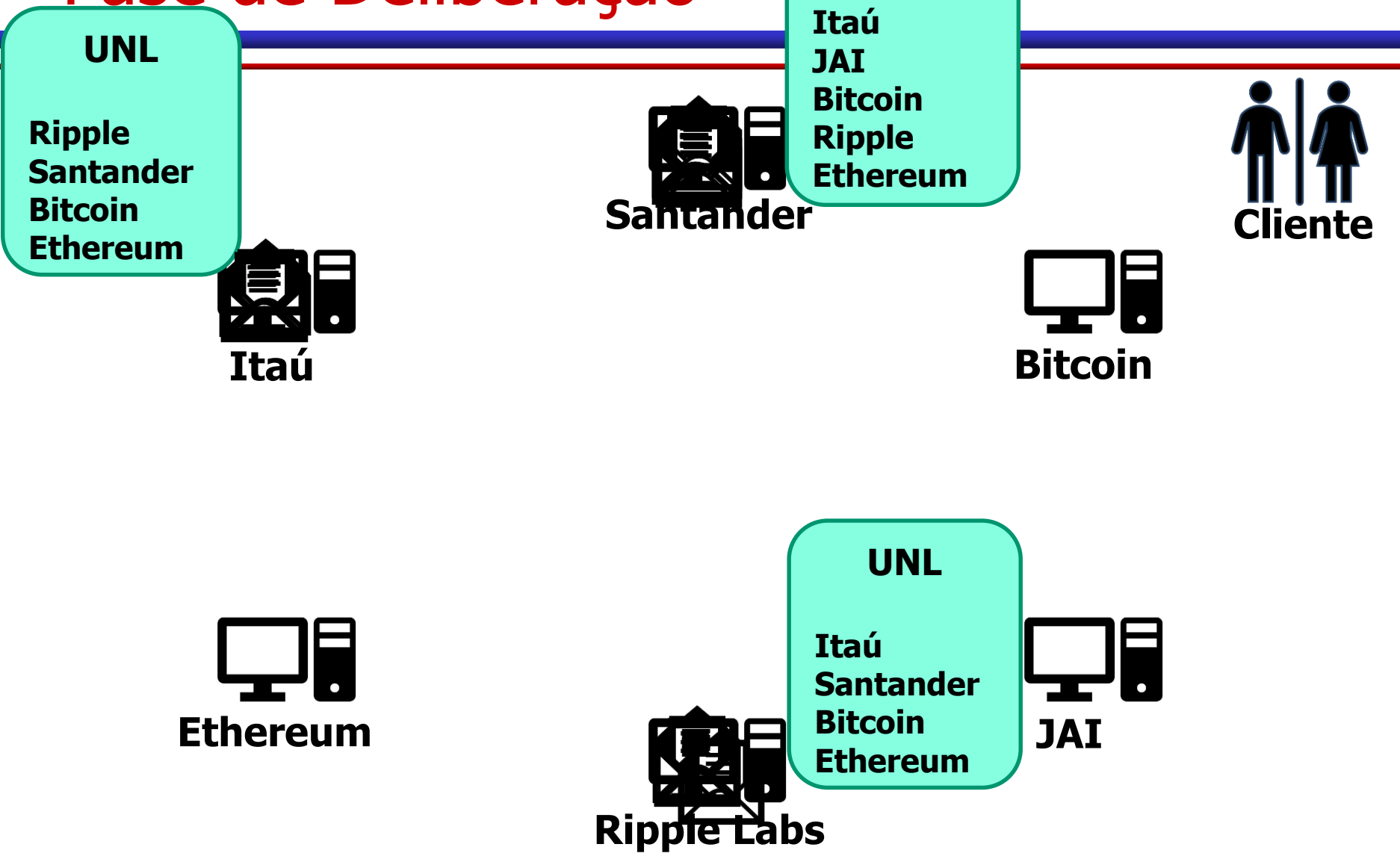
Ripple Labs

Funcionamento da Fase de Deliberação



Funcionamento da Fase de Deliberação

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO



Funcionamento da Fase de Deliberação

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO


Santander


Itaú


Bitcoin


Ethereum


Ripple Labs


JAI

Fase de Deliberação: Validador Itaú

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

UNL do Itaú

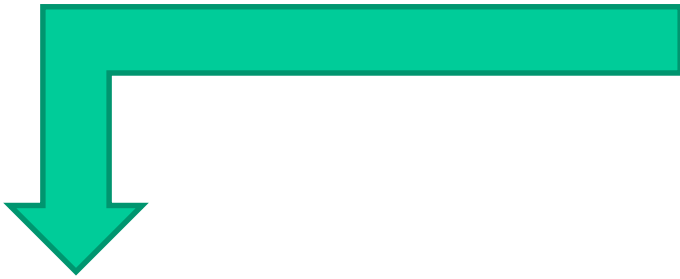
Ripple
Santander
Bitcoin
Ethereum

Ripple			
Santander			
Bitcoin			
Ethereum			
Limiar de aprovação: **50%**			

Fase de Deliberação: Validador Itaú

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

Transações dos clientes



UNL do Itaú

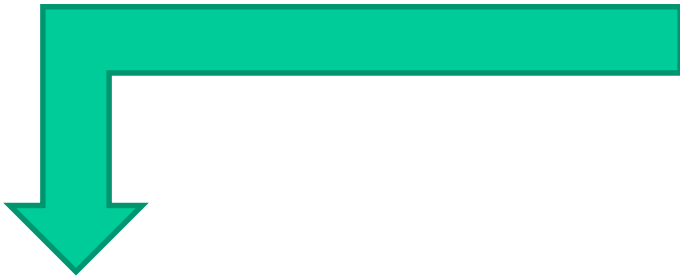
Ripple
Santander
Bitcoin
Ethereum

Ripple			
Santander			
Bitcoin			
Ethereum			
Limiar de aprovação: **50%**			

Fase de Deliberação: Validador Itaú

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

Transações dos clientes



UNL do Itaú

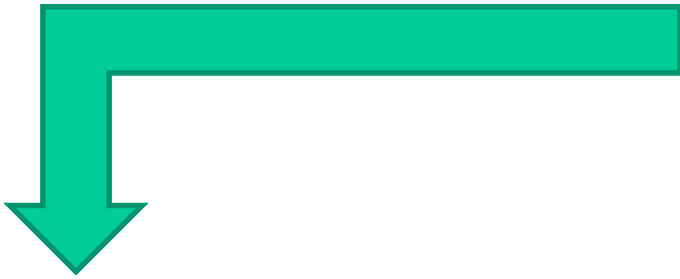
Ripple
Santander
Bitcoin
Ethereum

	T1		
Ripple			
Santander			
Bitcoin			
Ethereum			
Limiar de aprovação: **50%**			

Fase de Deliberação: Validador Itaú

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

Transações dos clientes



UNL do Itaú

Ripple
Santander
Bitcoin
Ethereum

	T1		
Ripple			
Santander			
Bitcoin			
Ethereum			
Limiar de aprovação: **50%**			

Fase de Deliberação: Validador Itaú

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

Proposta de conjunto de transações de outros validadores

Transações dos clientes

UNL do Itaú

Ripple
Santander
Bitcoin
Ethereum

	T1	T2	
Ripple			
Santander			
Bitcoin			
Ethereum			
Limiar de aprovação: **50%**			

Fase de Deliberação: Validador Itaú

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

Proposta de conjunto de transações de outros validadores

Transações dos clientes

Proposta do Ripple

UNL do Itaú

Ripple
Santander
Bitcoin
Ethereum

	T1	T2	T3
Ripple			
Santander			
Bitcoin			
Ethereum			
Limiar de aprovação: **50%**			

Fase de Deliberação: Validador Itaú

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

Proposta de conjunto de transações de outros validadores

Transações dos clientes

Proposta do Ripple

UNL do Itaú

Ripple
Santander
Bitcoin
Ethereum

	T1	T2	T3
Ripple	✓	✓	✓
Santander			
Bitcoin			
Ethereum			
Limiar de aprovação: **50%**			

Fase de Deliberação: Validador Itaú

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

Proposta de conjunto de transações de outros validadores

Transações dos clientes

Proposta do Bitcoin

UNL do Itaú

Ripple
Santander
Bitcoin
Ethereum

	T1	T2	T3
Ripple	✓	✓	✓
Santander			
Bitcoin			
Ethereum			
Limiar de aprovação: **50%**			

Fase de Deliberação: Validador Itaú

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

Proposta de conjunto de transações de outros validadores

Transações dos clientes

Proposta do Bitcoin

UNL do Itaú

Ripple
Santander
Bitcoin
Ethereum

	T1	T2	T3
Ripple	✓	✓	✓
Santander			
Bitcoin	✓	✗	✗
Ethereum			
Limiar de aprovação: **50%**			

Fase de Deliberação: Validador Itaú

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

Proposta de conjunto de transações de outros validadores

Transações dos clientes

Proposta do Ethereum

UNL do Itaú

Ripple
Santander
Bitcoin
Ethereum

	T1	T2	T3
Ripple	✓	✓	✓
Santander			
Bitcoin	✓	✗	✗
Ethereum			
Limiar de aprovação: **50%**			

Fase de Deliberação: Validador Itaú

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

Proposta de conjunto de transações de outros validadores

Transações dos clientes

Proposta do Ethereum

UNL do Itaú

Ripple
Santander
Bitcoin
Ethereum

	T1	T2	T3
Ripple	✓	✓	✓
Santander			
Bitcoin	✓	✗	✗
Ethereum	✓	✓	✗
Limiar de aprovação: **50%**			

Fase de Deliberação: Validador Itaú

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

Proposta de conjunto de transações de outros validadores

UNL do Itaú

Ripple
Santander
Bitcoin
Ethereum

Tempo da rodada **terminou**!

	T1	T2	T3
Ripple	✓	✓	✓
Santander			
Bitcoin	✓	✗	✗
Ethereum	✓	✓	✗
Limiar de aprovação: **50%**			

Fase de Deliberação: Validador Itaú

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

Proposta de conjunto de transações de outros validadores

UNL do Itaú

Ripple
Santander
Bitcoin
Ethereum

Tempo da rodada **terminou**!

	T1	T2	T3
	75%	**50%**	**25%**
Ripple	✓	✓	✓
Santander	✗	✗	✗
Bitcoin	✓	✗	✗
Ethereum	✓	✓	✗
Limiar de aprovação: **50%**			

Fase de Deliberação: Validador Itaú

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

Proposta de conjunto de transações dos outros validadores

Transações dos clientes

T1 e **T2** viram a proposta de conjunto do Itaú e passa para próxima rodada

UNL do Itaú

Ripple
Santander
Bitcoin
Ethereum

	T1	T2	T3
	75%	50%	25%
Ripple	✓	✓	✓
Santander	✗	✗	✗
Bitcoin	✓	✗	✗
Ethereum	✓	✓	✗
Limiar de aprovação: 50%			

Fase de Deliberação: Validador Itaú

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

Proposta de conj
de transaçõe
outros valida

Transações dos clientes

Na segunda rodada, tempo é reiniciado e o limiar é aumentado para **60%**

UNL do Itaú

Ripple
Santander
Bitcoin
Ethereum

	T1	T2	T3
	75%	**50%**	**25%**
Ripple	✓	✓	✓
Santander	✗	✗	✗
Bitcoin	✓	✗	✗
Ethereum	✓	✓	✗
Limiar de aprovação: **60%**			

Fase de Deliberação: Validador Itaú

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

Proposta de conj
de transaçõe
outros valida

Transações dos clientes

Na segunda rodada, temporizador é reiniciado e o limiar é aumentado para **60%**

	T1	T2	T3
Ethereum	✓	✓	✗
Limiar de aprovação: **60%**			

Limiar de aprovação das rodadas:
50%→60%→70%→80%...

U
R
S
Bitcoin
Ethereum

Fase de Deliberação: Validador Itaú

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

Proposta de co

de t

out

Transações dos clientes

A fase de deliberação termina quando há acordo (aceitar ou rejeitar) em **80%** das transações.

U

R

S

Bitcoin

Ethereum

Ethereum

Limiar de aprovação: **60%**

Fase de Deliberação: Validador Itaú

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

Proposta de co

de t

out

Transações dos clientes

No final, o conjunto das transações que obtiveram uma incidência superior ou igual ao limiar de aprovação será inserido no livro-razão, assim chegando a um **consenso**

U

R

S

Bitcoin

Ethereum

Ethereum	✓	✓	✗

Limiar de aprovação: **60%**

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

The Stellar Consensus Protocol: A Federated Model for Internet-level Consensus

David Mazières

Introdução

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

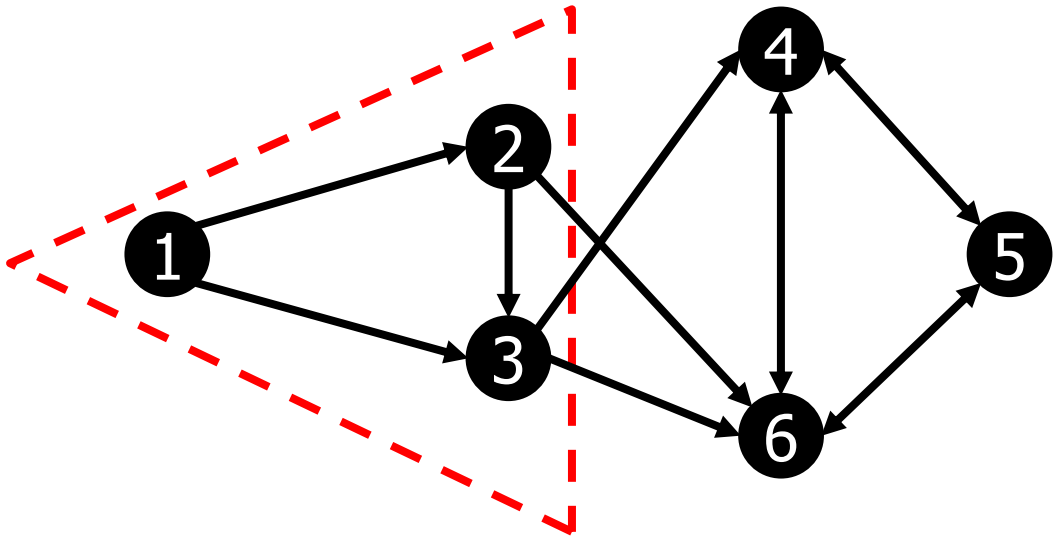
- Problemas em mecanismos de consenso convencionais
  - Prova de posse
    - Problema do "nada a perder"
  - Prova de trabalho
    - Alto consumo energético
  - Acordo bizantino
    - Os interessados devem concordar em uma lista de participantes do consenso
    - Ataques Sybil

É necessário um novo protocolo de consenso

Acordo Bizantino Federado (*Federated Byzantine Agreement* – FBA)

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

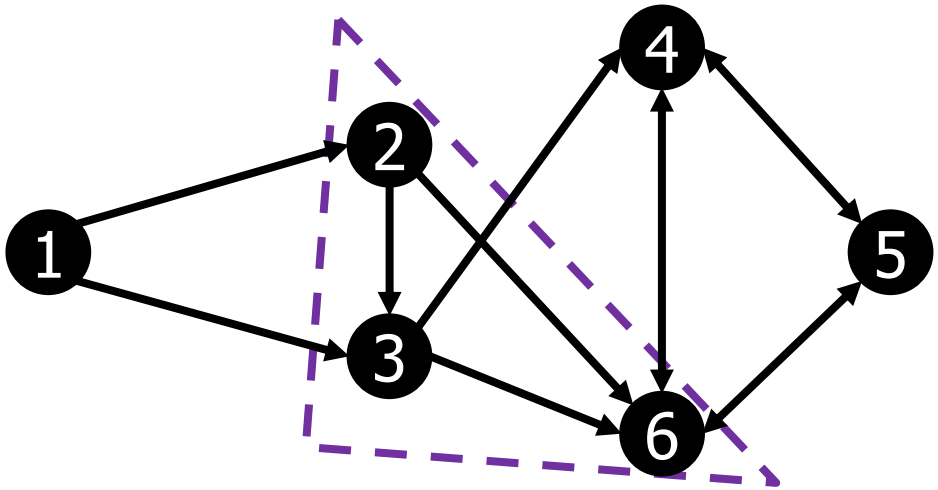
- Maior liberdade aos participantes
  - Cada participante define em quem confia
    - O participante e a lista em quem confia forma uma fatia de quórum (*quorum slice*)



Acordo Bizantino Federado (*Federated Byzantine Agreement* – FBA)

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

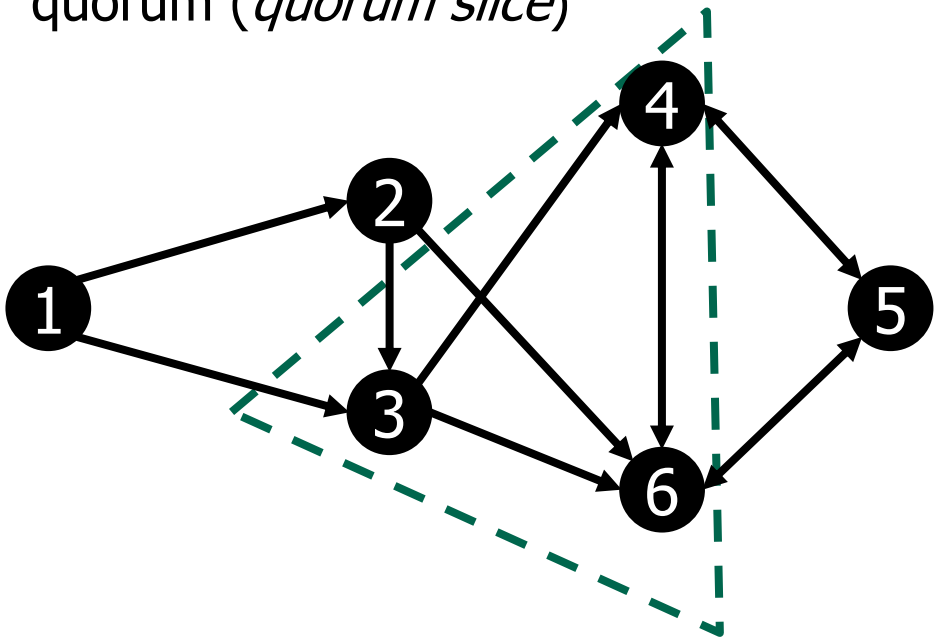
- Maior liberdade aos participantes
  - Cada participante define em quem confia
    - O participante e a lista em quem confia forma uma fatia de quórum (*quorum slice*)



Acordo Bizantino Federado (*Federated Byzantine Agreement* – FBA)

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

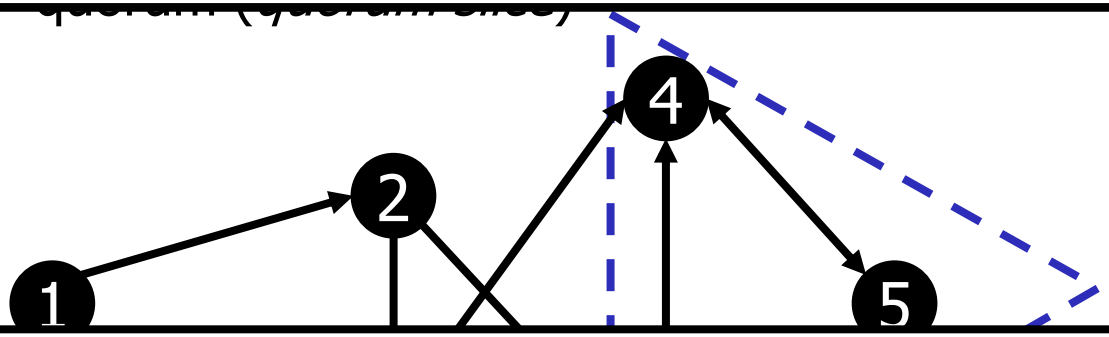
- Maior liberdade aos participantes
  - Cada participante define em quem confia
    - O participante e a lista em quem confia forma uma fatia de quórum (*quorum slice*)



Acordo Bizantino Federado
(*Federated Byzantine Agreement* – FBA)

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- Como encontrar um quórum a partir de uma fatia de quórum?

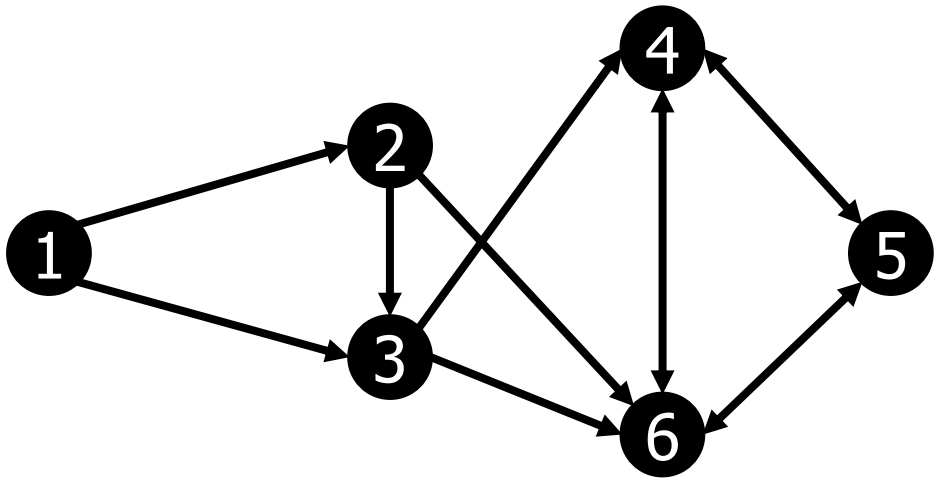


As interseções nas fatias de quórum garantem o consenso global

Acordo Bizantino Federado (*Federated Byzantine Agreement* – FBA)

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

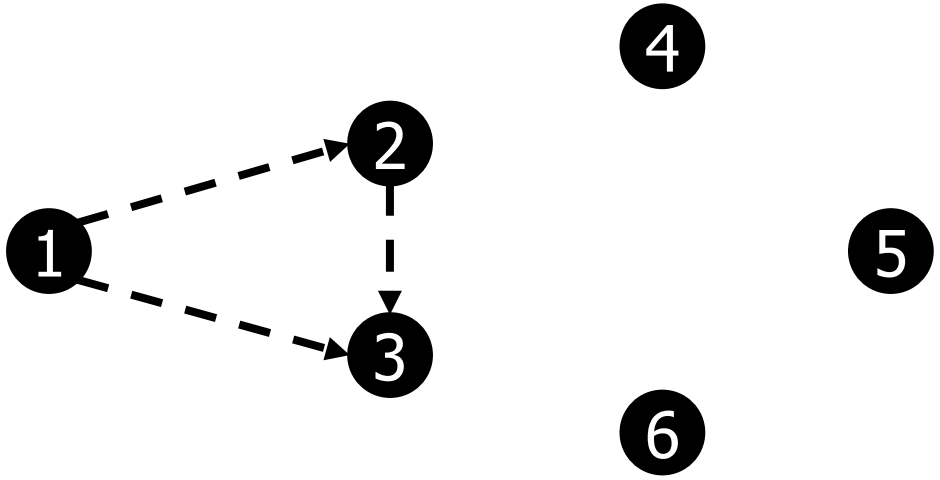
- 1) Iniciar com uma fatia de quórum
- 2) Adicionar as fatias de todos os membros
- 3) Repetir o passo 2 até chegar em um nó que já foi adicionado



Acordo Bizantino Federado (*Federated Byzantine Agreement* – FBA)

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

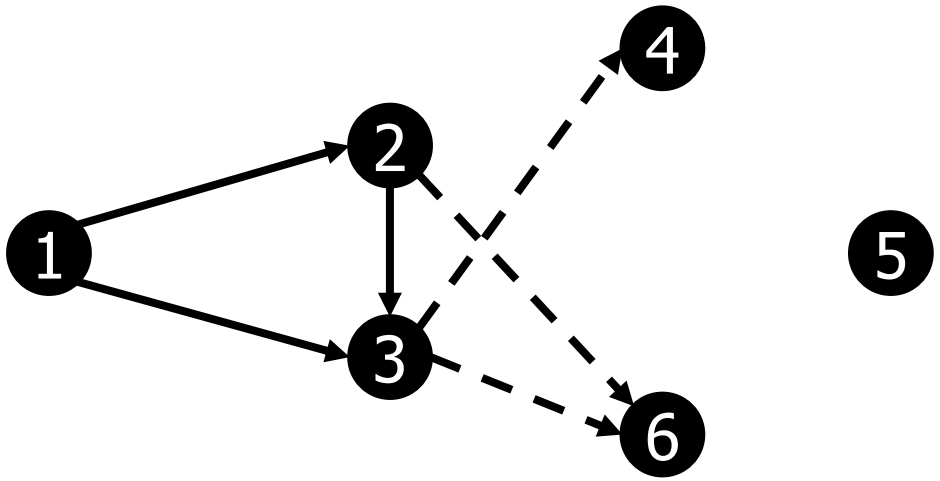
- **1) Iniciar com uma fatia de quórum**
- 2) Adicionar as fatias de todos os membros
- 3) Repetir o passo 2 até chegar em um nó que já foi adicionado



Acordo Bizantino Federado (*Federated Byzantine Agreement* – FBA)

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

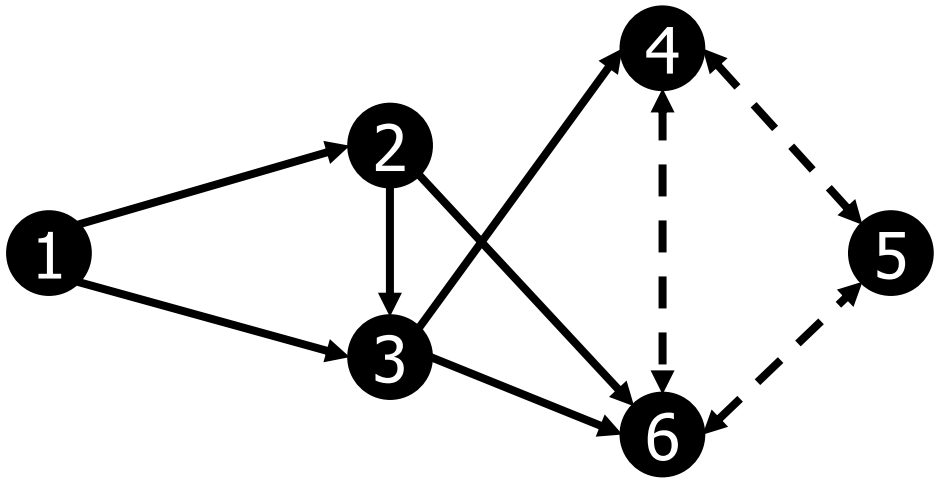
- 1) Iniciar com uma fatia de quórum
- **2) Adicionar as fatias de todos os membros**
- 3) Repetir o passo 2 até chegar em um nó que já foi adicionado



Acordo Bizantino Federado (*Federated Byzantine Agreement* – FBA)

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

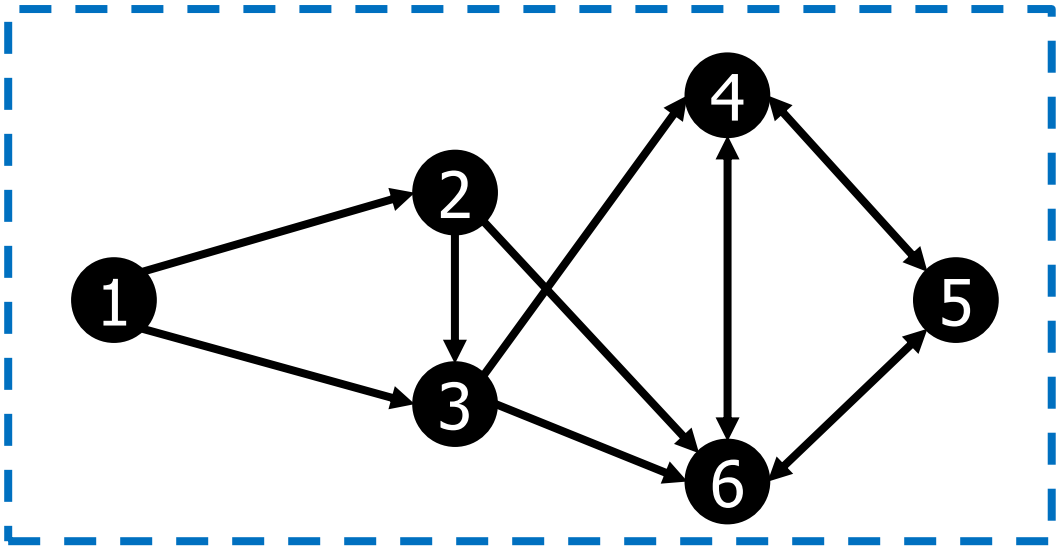
- 1) Iniciar com uma fatia de quórum
- 2) Adicionar as fatias de todos os membros
- **3) Repetir o passo 2 até chegar em um nó que já foi adicionado**



Acordo Bizantino Federado (*Federated Byzantine Agreement* – FBA)

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- 1) Iniciar com uma fatia de quórum
- 2) Adicionar as fatias de todos os membros
- **3) Repetir o passo 2 até chegar em um nó que já foi adicionado**



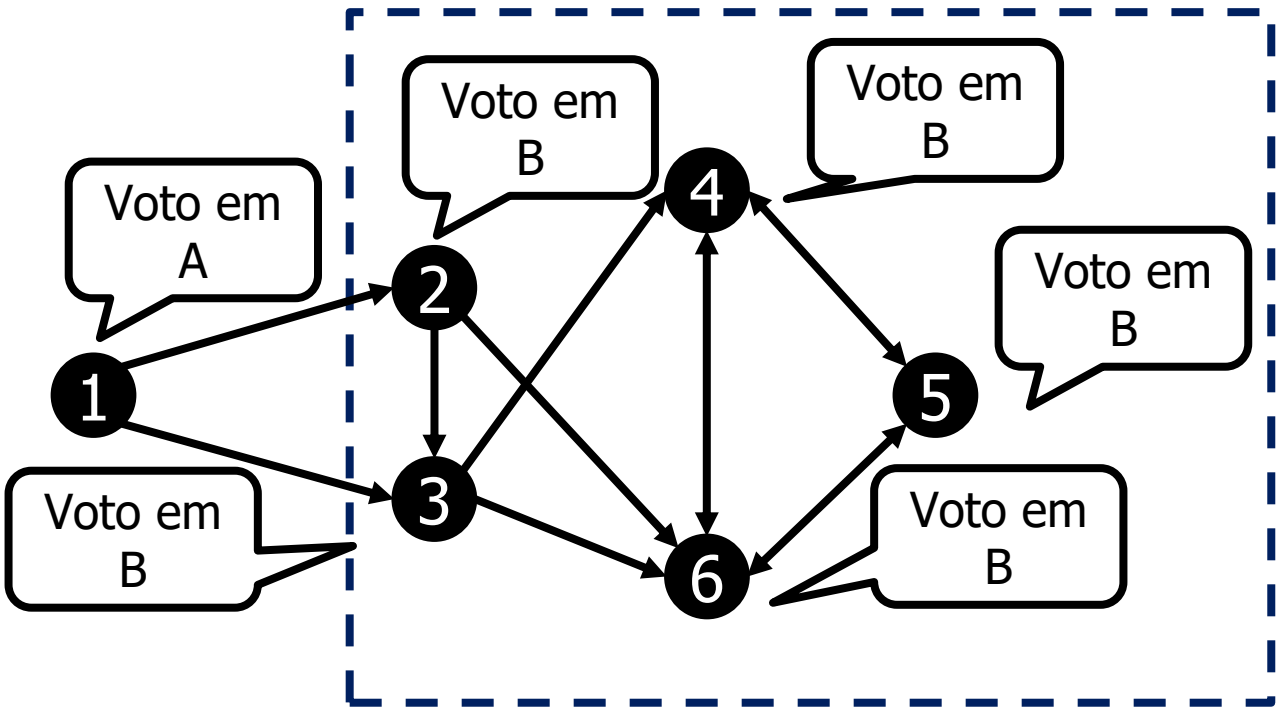
Menor quórum que o nó 1 pode encontrar

Votação Federada

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- Procedimento em que nós descobrem se um quórum concorda com uma proposta
  - Os nós devem escolher um entre diversos valores como saída de uma rodada
    - Cada nó vota em um valor. Se achar um quórum que vota no mesmo valor, ele passa a aceitar o valor votado.
    - Um nó pode votar em um valor e aceitar um valor diferente, se o valor diferente for aceito na rede.
  - Acordo é atingido com troca de mensagens

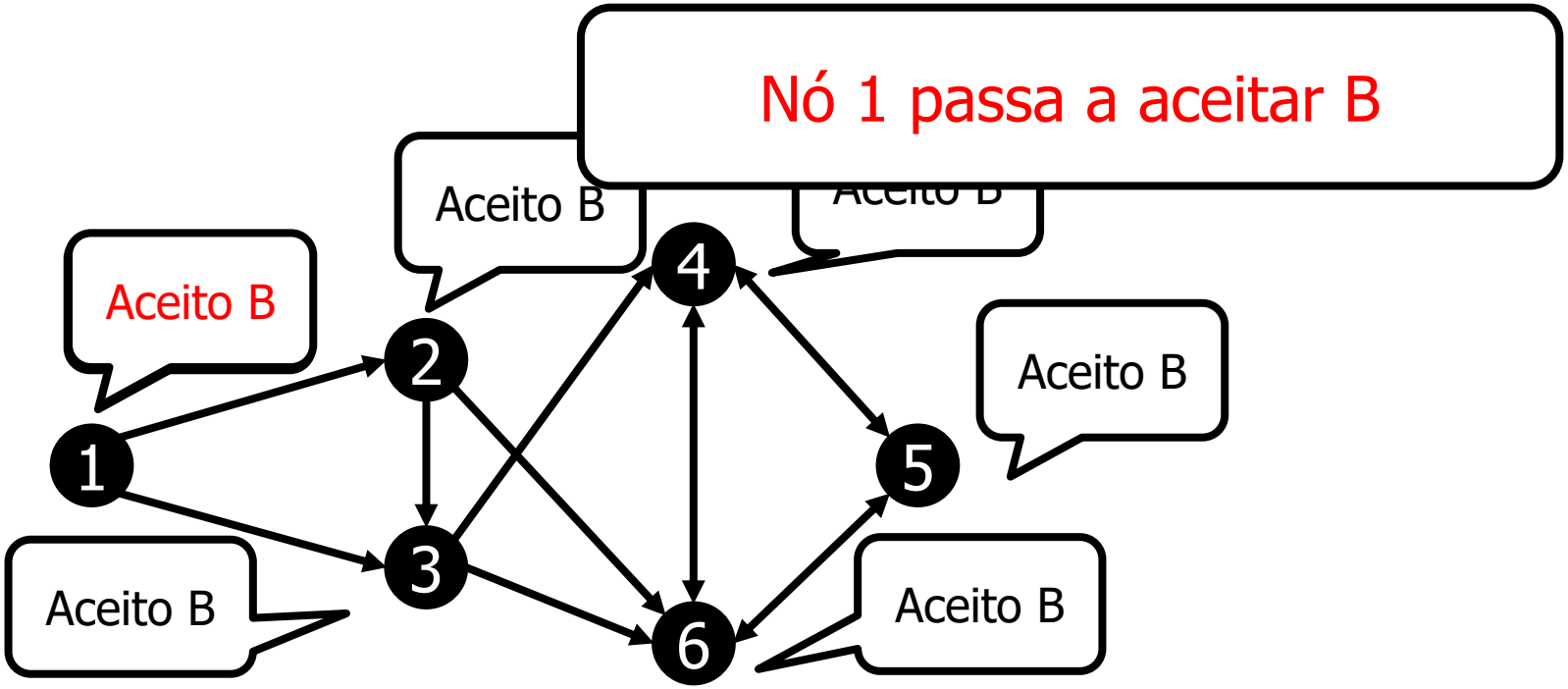
Votação Federada



Quórum votou em B

Quórum passa a aceitar B

Votação Federada



Quórum passa a aceitar B

O Protocolo de Consenso Stellar (*Stellar Consensus Protocol* – SCP)

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- Stellar Consensus Protocol (SCP) é um protocolo descentralizado para enviar e receber dinheiro em qualquer par de moedas de maneira rápida, segura a um baixíssimo custo
- Diferenciais
  - Qualquer duas moedas
  - Qualquer parte do mundo
  - Transação realizada em poucos segundos
  - Tarifas muito baixas, quase zero
  - Software aberto, gerenciamento sem fins lucrativos
  - Criador David Mazières, professor de Stanford líder do grupo sistemas seguros de computadores

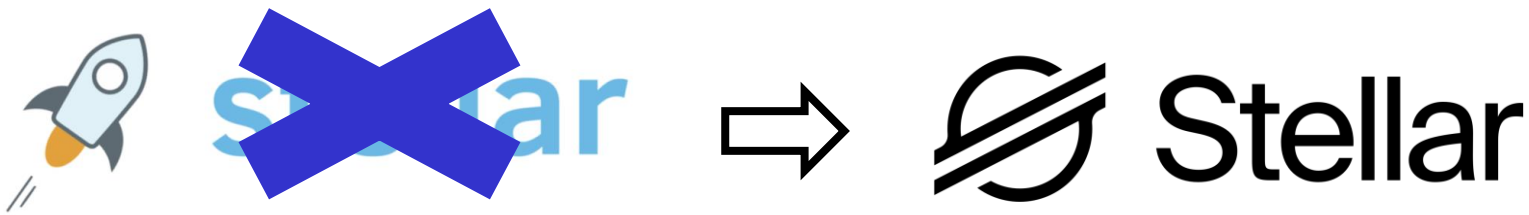
Página pessoal

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

O Protocolo de Consenso Stellar (*Stellar Consensus Protocol* – SCP)

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- Protocolo baseado no acordo bizantino federado
  - Primeiro protocolo de acordo bizantino a prover liberdade de confiança ao participante
  - Utilizado pela criptomoeda da plataforma Stellar
    - 12ª criptomoeda em valor de mercado



O Protocolo de Consenso Stellar (*Stellar Consensus Protocol* – SCP)

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- Os nós devem trocar mensagem para chegar a um acordo sobre um valor
  - 1º: Os nós devem escolher um valor
  - 2º: Os nós devem votar se aceitam o valor proposto

- O Protocolo de Consenso Stellar é separado em duas fases
  - Fase 1: Nomeação (*Nomination)*
  - Fase 2: Votação (*Balloting)*

O Protocolo de Consenso Stellar (*Stellar Consensus Protocol* – SCP)

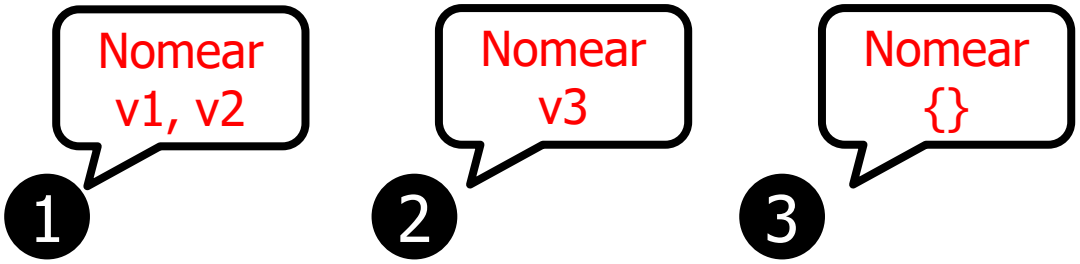
GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- Fase de nomeação
  - Nós nomeiam um valor
  - Nós propagam os valores nomeados
    - O conjunto de valores nomeados deve convergir
  - Nós devem combinar os valores nomeados em um valor composto

O Protocolo de Consenso Stellar (*Stellar Consensus Protocol* – SCP)

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

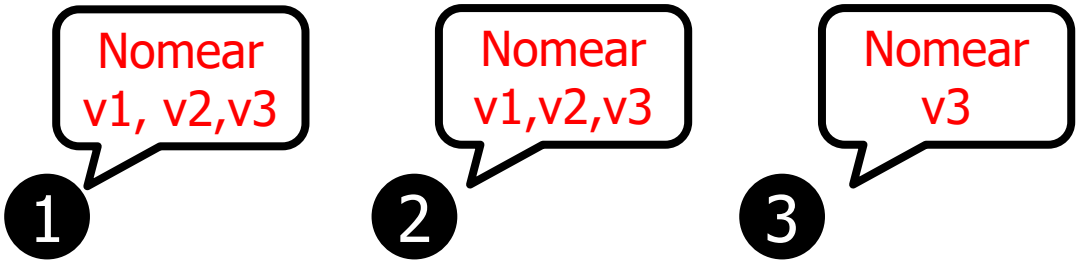
- Fase de nomeação
  - Nós nomeiam um valor
  - Nós propagam os valores nomeados
    - O conjunto de valores nomeados deve convergir
  - Nós devem combinar os valores nomeados em um valor composto



O Protocolo de Consenso Stellar (*Stellar Consensus Protocol* – SCP)

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

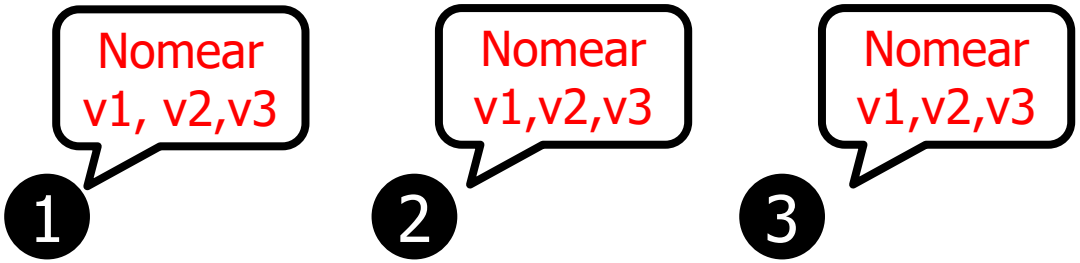
- Fase de nomeação
  - Nós nomeiam um valor
  - Nós propagam os valores nomeados
    - O conjunto de valores nomeados deve convergir
  - Nós devem combinar os valores nomeados em um valor composto



O Protocolo de Consenso Stellar (*Stellar Consensus Protocol* – SCP)

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- Fase de nomeação
  - Nós nomeiam um valor
  - Nós propagam os valores nomeados
    - O conjunto de valores nomeados deve convergir
  - Nós devem combinar os valores nomeados em um valor composto



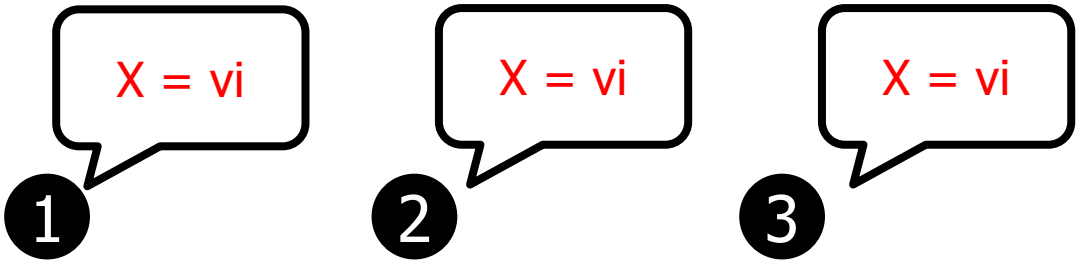
O Protocolo de Consenso Stellar (*Stellar Consensus Protocol* – SCP)

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

Problema: Como o protocolo é assíncrono, os nós não sabem quando a nomeação acaba

- O conjunto de valores nomeados deve convergir

– Nós devem combinar os valores nomeados em um valor composto



O Protocolo de Consenso Stellar (*Stellar Consensus Protocol* – SCP)

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

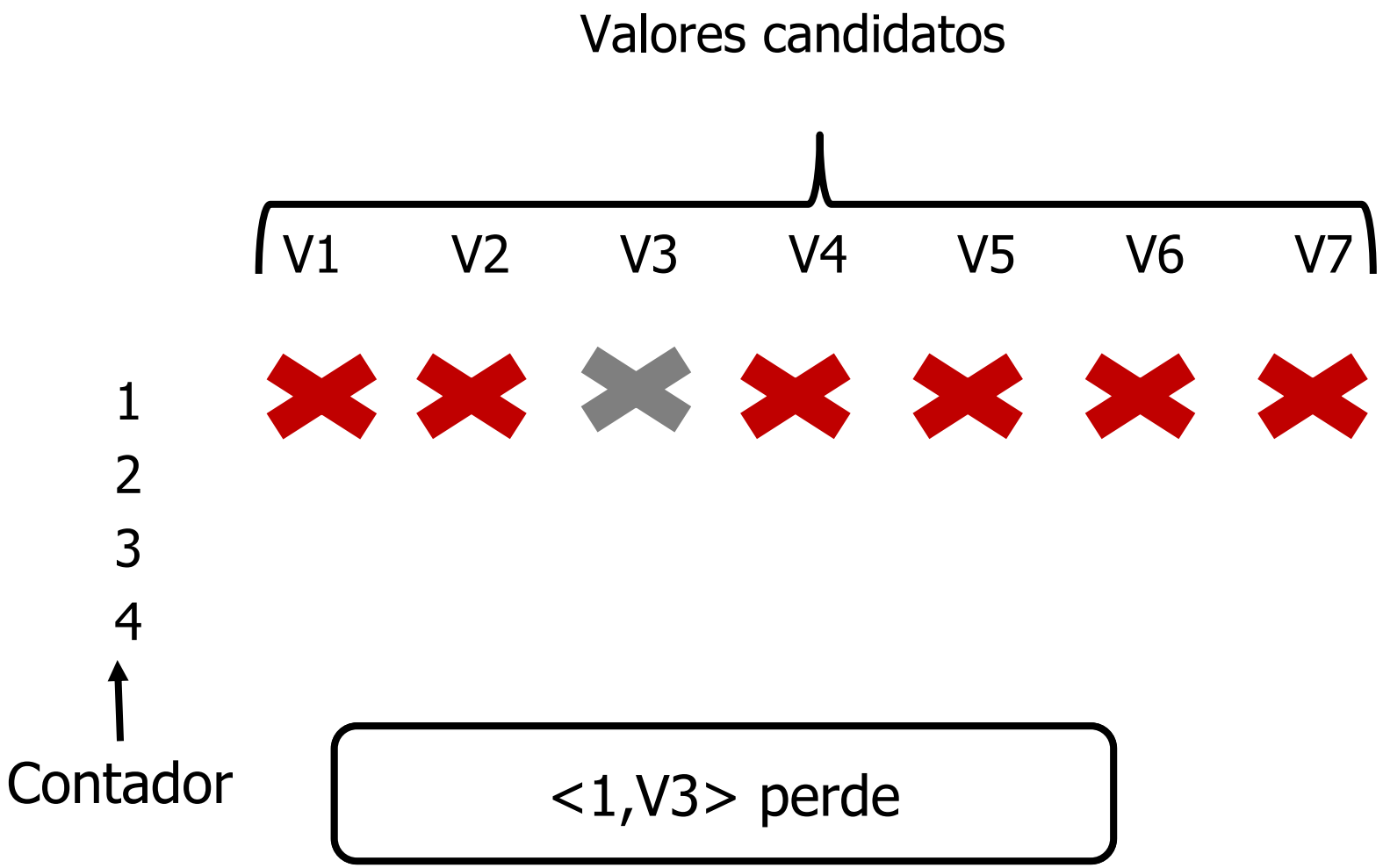
- Fase de votação
  - Fase acontece mesmo que a etapa de nomeação ainda não tenha convergido
  - Nós executam uma votação federada e votam em cédulas
    - Uma cédula é formada por um contador e um valor a ser votado
    - Nós votam para efetivar ou eliminar um valor
    - As cédulas devem estar **preparadas** para serem votadas

O Protocolo de Consenso Stellar (*Stellar Consensus Protocol* – SCP)

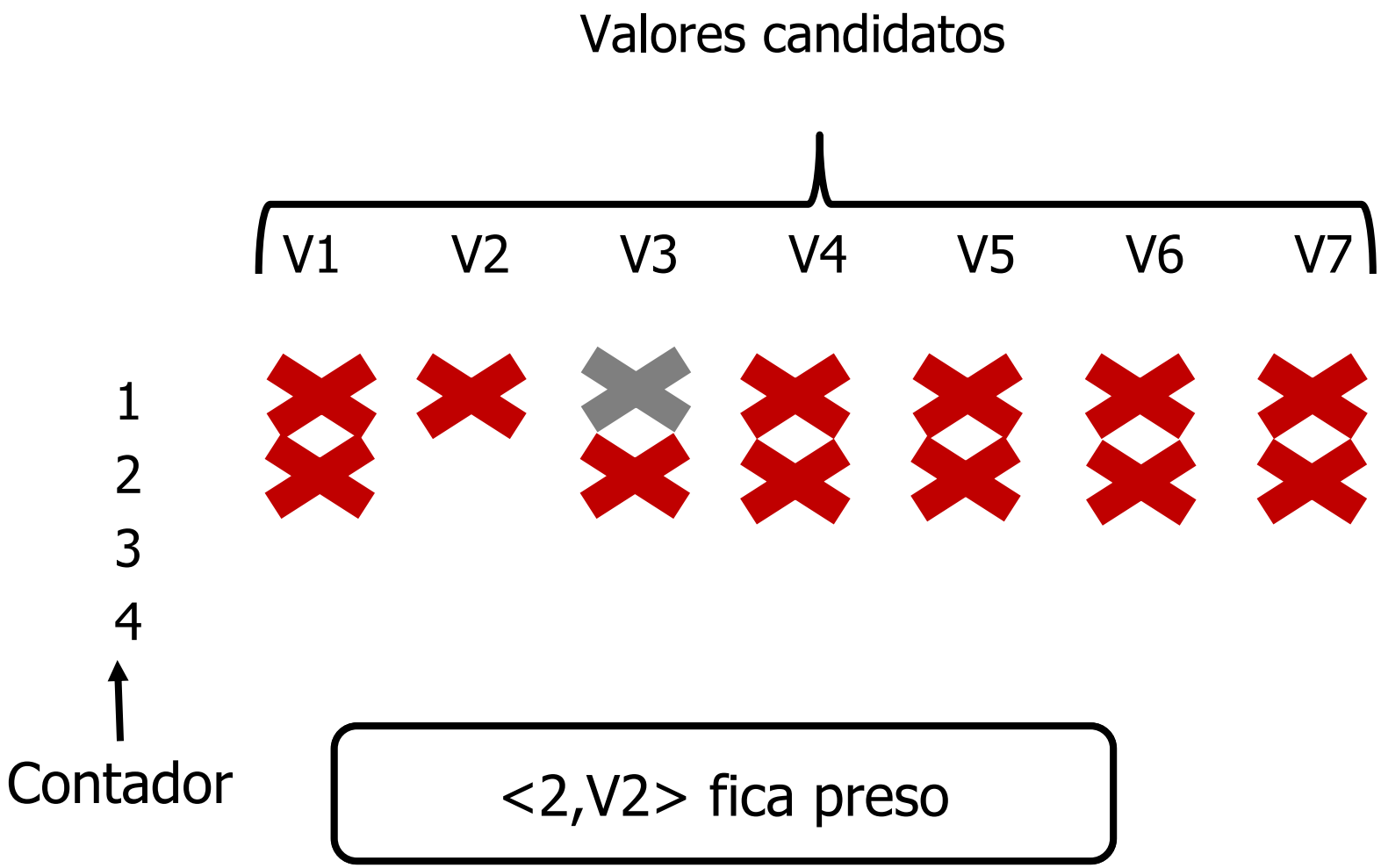
GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- Fase de votação
  - Fase acontece mesmo que a etapa de nomeação ainda não tenha convergido
  - Nós executam uma votação federada e votam em cédulas
    - Uma cédula é formada por um contador e um valor a ser votado
    - Nós votam para efetivar ou eliminar um valor
    - As cédulas devem estar preparadas para serem votadas

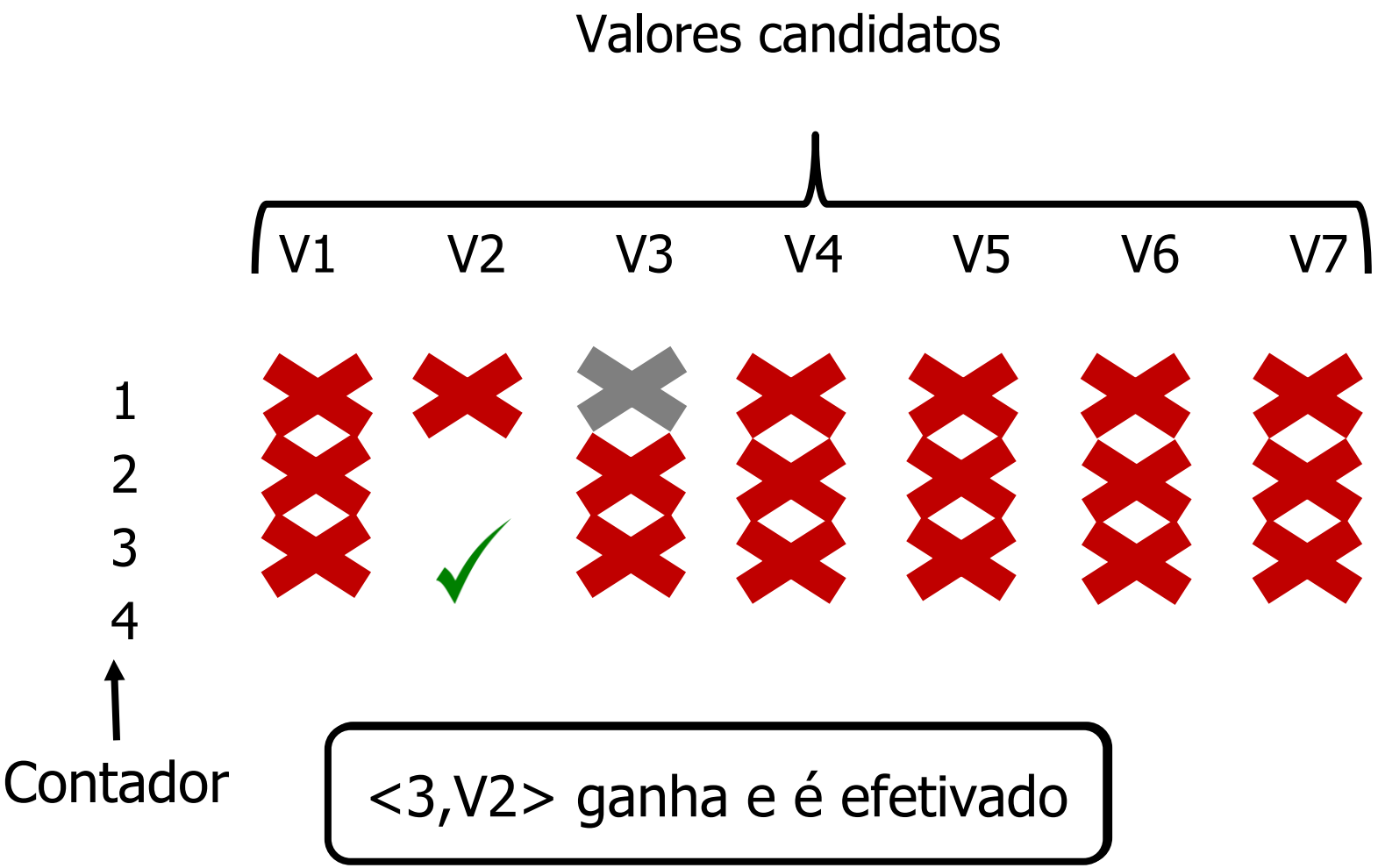
O Protocolo de Consenso Stellar (*Stellar Consensus Protocol* – SCP)



O Protocolo de Consenso Stellar (*Stellar Consensus Protocol* – SCP)



O Protocolo de Consenso Stellar (*Stellar Consensus Protocol* – SCP)



EOS.IO

Daniel Larimer

Dan Larimer

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- Dan Larimer
- Criou a plataforma BitShares em outubro de 2015
- Co-fundou a corrente de blocos social Steem
- Atualmente preside a Block.one, que desenvolve a EOS

Bitshare

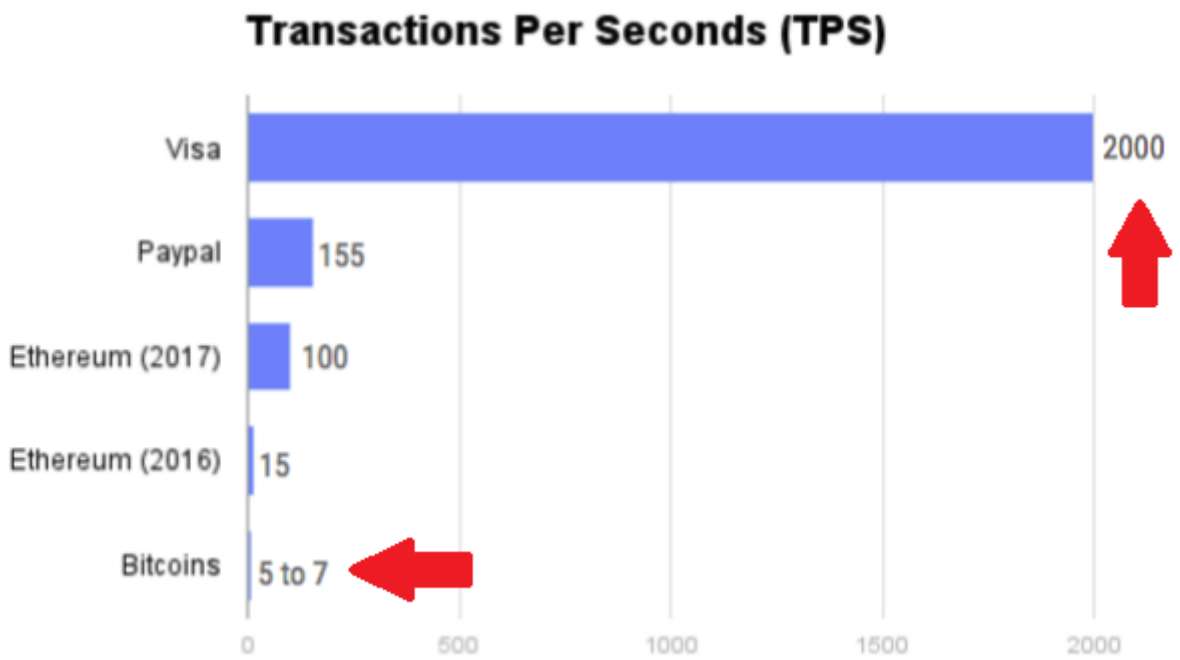
GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- Criada por Dan Larimer em outubro de 2015

Taxa de Transações por Segundo

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- Baixa taxa de transações por segundo (tps) de criptomoedas tradicionais



EOS

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- Proposta: novo protocolo de consenso
  - Rápido
  - Seguro

EOS

- Prova de Posse Delegada (*Delegated Proof of Stake* - DPoS)
  - Não requer mineração
  - Validação depende somente dos delegados
    - Mais rápido que PoW e PoS

- Tolerância a Falhas Bizantinas (*Byzantine Fault Tolerance* - BFT)

Características do EOS

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- **Flexível**
  - Pode ser usado para correntes de blocos públicas ou privadas

- **Amigável- Usuários e Desenvolvedores**
  - Design voltado para facilitar o desenvolvimento e a execução de aplicações distribuídas

- **Escalável**
  - Atualmente suporta até 4000 transações por segundo
  - Teoricamente suporta milhões de transações por segundo

- 6a maior moeda em valor de mercado
  - Mais de US$5,000,000,000

Prova de Posse Delegada

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

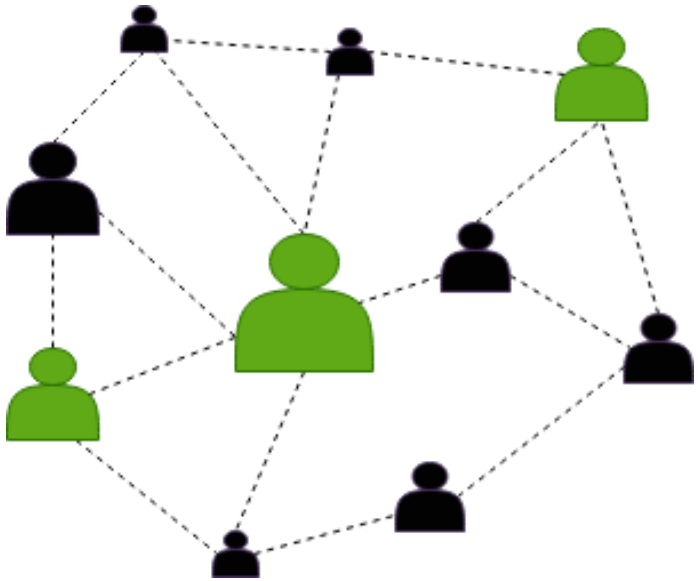
- Participantes da rede elegem delegados
  - Votos feitos utilizando ativos do participante
  - Quantidade limitada de delegados
    - EOS: 21
    - Lisk: 101
    - Ark: 51

- Delegados responsáveis por criar e validar blocos
  - Não requer mineração → Maior velocidade e menor consumo energético

Eleição de Delegados

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

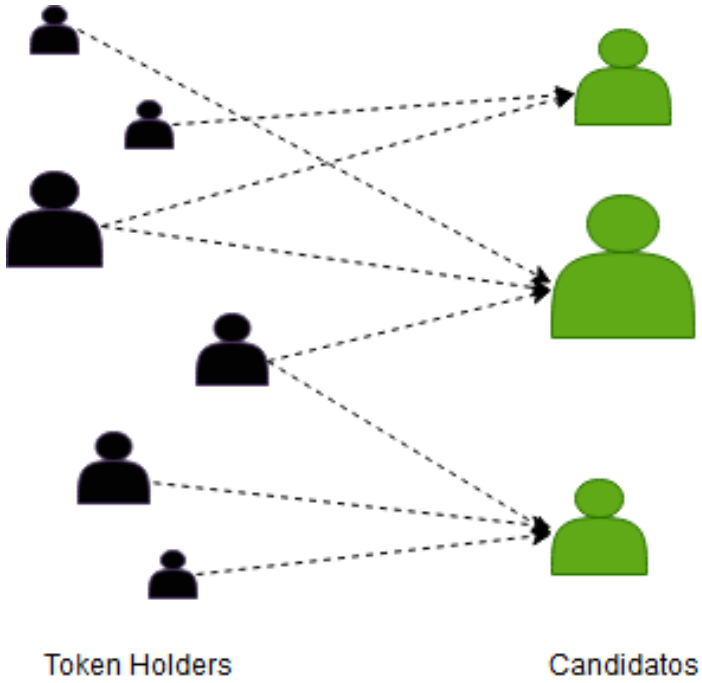
- Nós indicam interesse em se tornarem delegados
  - Buscam votos interagindo com a comunidade e provando serem confiáveis



Eleição de Delegados

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- Participantes da rede votam em nós utilizando tokens
  - Podem votar em múltiplos candidatos
  - Votos podem ser realocados a qualquer momento



Eleição de Delegados

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- Selecionam-se os N candidatos com maior quantidade em votos para se tornarem delegados
  - N depende do protocolo

Protocolo BFT-DPoS

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- Utiliza conceitos da Prova de Posse Delegada (DPoS) e Tolerância a Falhas Bizantinas (BFT)
  - DPoS possibilita alta taxa de transações
  - BFT usado na finalização de blocos
    - Garante irreversibilidade do bloco adicionado dentro de 1 segundo

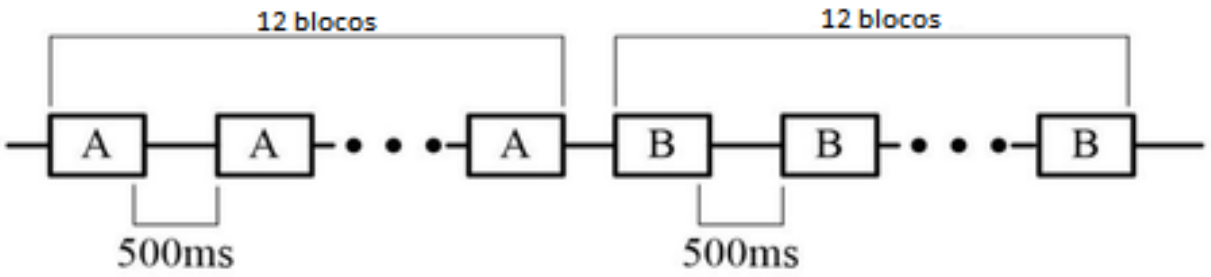
- Delegados são denominados produtores de blocos
  - 21 produtores eleitos a cada rodada

Protocolo BFT-DPoS

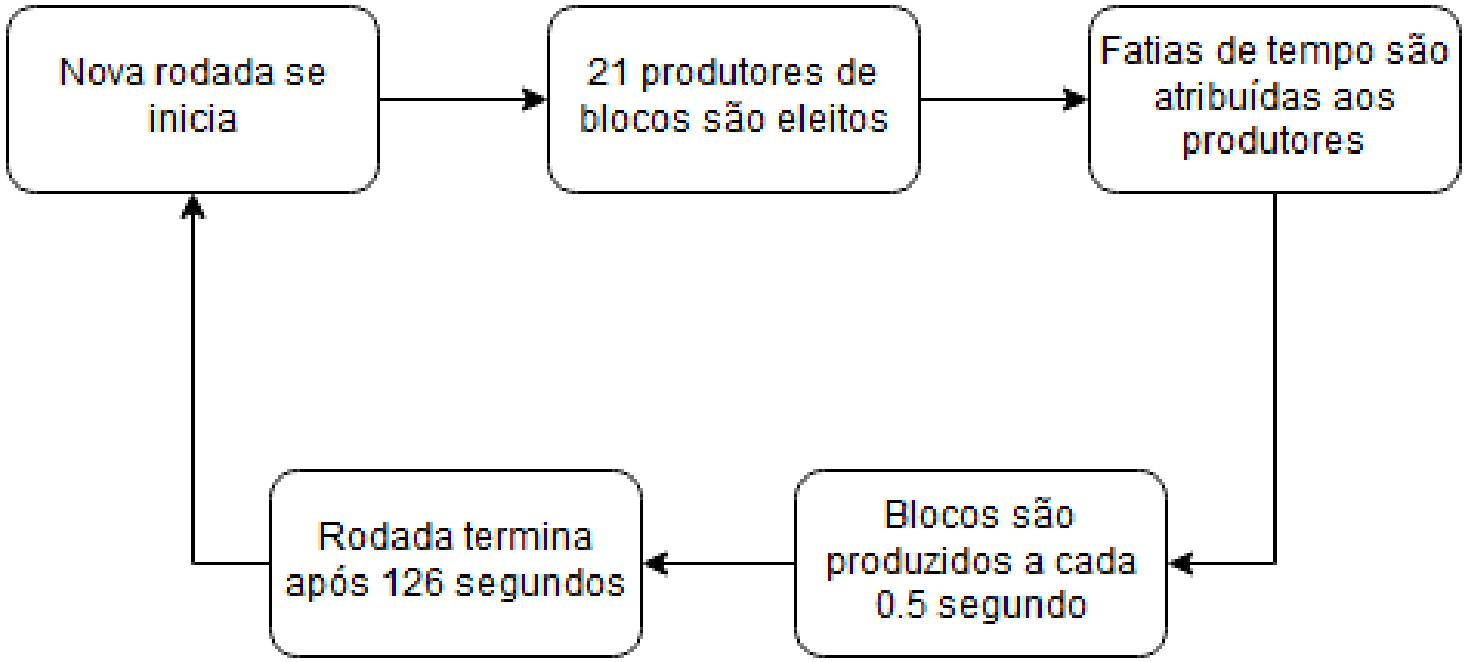
GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- Produtores concordam em uma ordem de produção após término da eleição
  - Ordem deve ser concordada por pelo menos 15 produtores

- Blocos produzidos em rodadas
  - Cada produtor é responsável por produzir 12 blocos
  - Novo bloco produzido a cada 0.5 segundo
    - Pulado caso não seja produzido a tempo



Rodada no Protocolo BFT-DPoS



Exemplo Protocolo BFT-DPoS

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

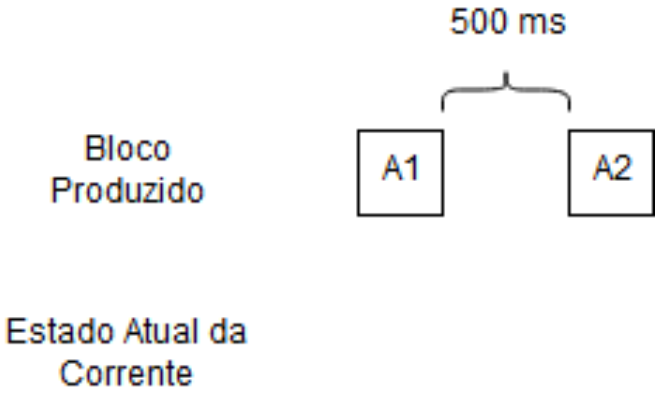
Bloco Produzido: A1

Estado Atual da Corrente

Confirmações Recebidas (+⅔ para ser adicionado à corrente)					
A1	A2	A3	A4	A5	A6
1/21					

Exemplo Protocolo BFT-DPoS

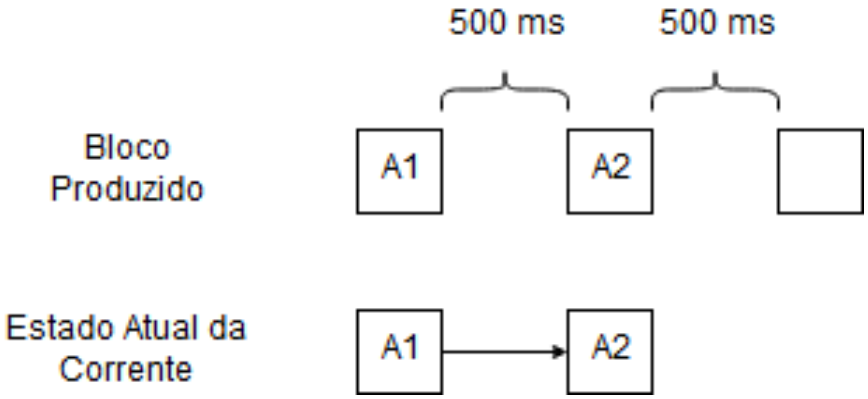
GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO



Confirmações Recebidas (+⅔ para ser adicionado à corrente)					
A1	A2	A3	A4	A5	A6
14/21	1/21				

Exemplo Protocolo BFT-DPoS

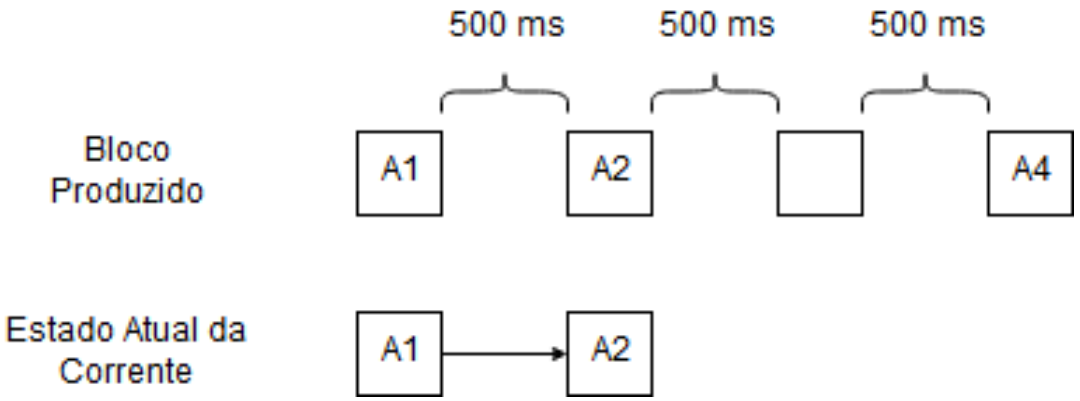
GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO



Confirmações Recebidas (+⅔ para ser adicionado à corrente)					
A1	A2	A3	A4	A5	A6
21/21	17/21				

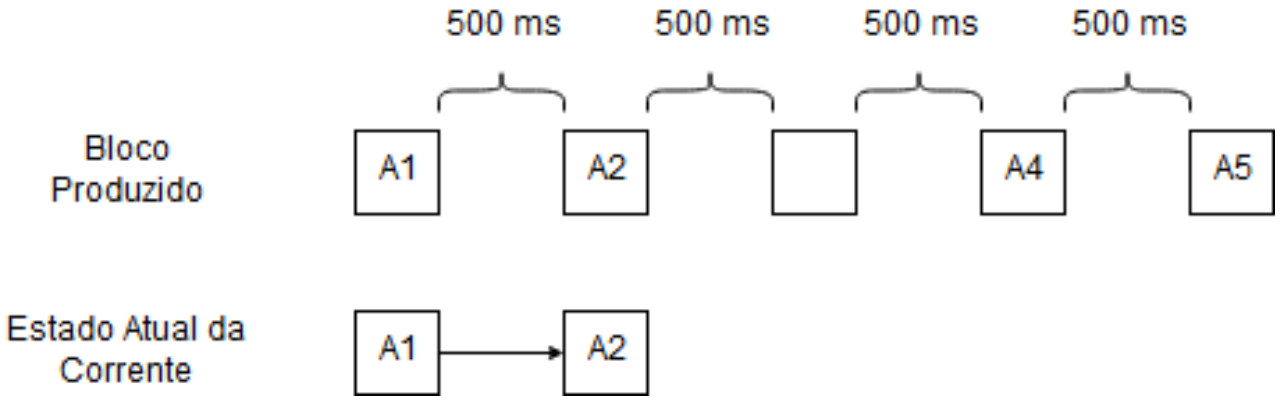
Exemplo Protocolo BFT-DPoS

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO



Confirmações Recebidas (+⅔ para ser adicionado à corrente)					
A1	A2	A3	A4	A5	A6
21/21	21/21		1/21		

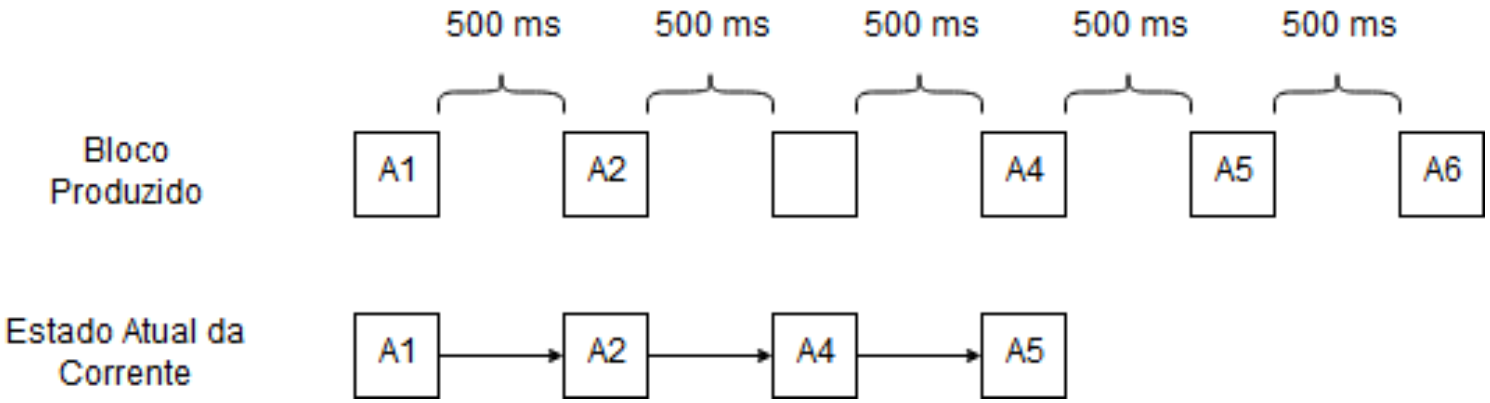
Exemplo Protocolo BFT-DPoS



Confirmações Recebidas (+⅔ para ser adicionado à corrente)					
A1	A2	A3	A4	A5	A6
21/21	21/21		12/21	1/21	

Exemplo Protocolo BFT-DPoS

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO



Confirmações Recebidas (+⅔ para ser adicionado à corrente)					
A1	A2	A3	A4	A5	A6
21/21	21/21		21/21	16/21	1/21

Transação como Prova de Posse (TaPoS)

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- Cada transação inclui parte do hash do cabeçalho de um bloco recente
  - Evita que uma transação seja repetida em forks que não incluam o bloco referenciado
  - Indica a rede que determinado usuário e seu montante pertencem a determinada fork

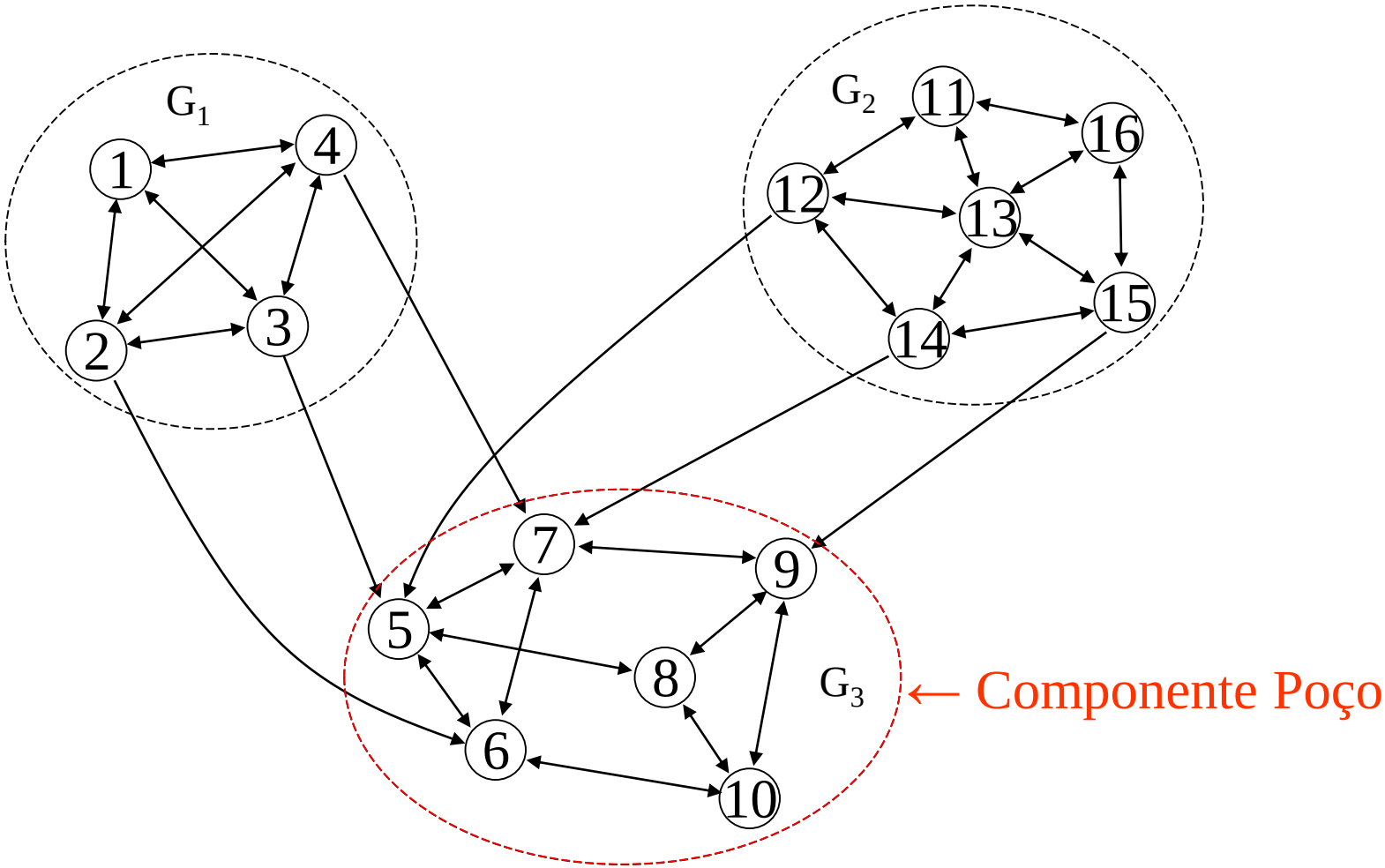
- Dificulta a migração de transações entre forks

BFT-CUP

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- Determinação do Poço
- Visa estabelecer quem são os processos membros da componente poço
- Na fase anterior, membros da componente poço obtêm uma visão parcial do sistema menor do que a visão obtida pelos demais processos
- Cada participante conhece pelo menos os membros da componente a que pertencem e os membros do poço.

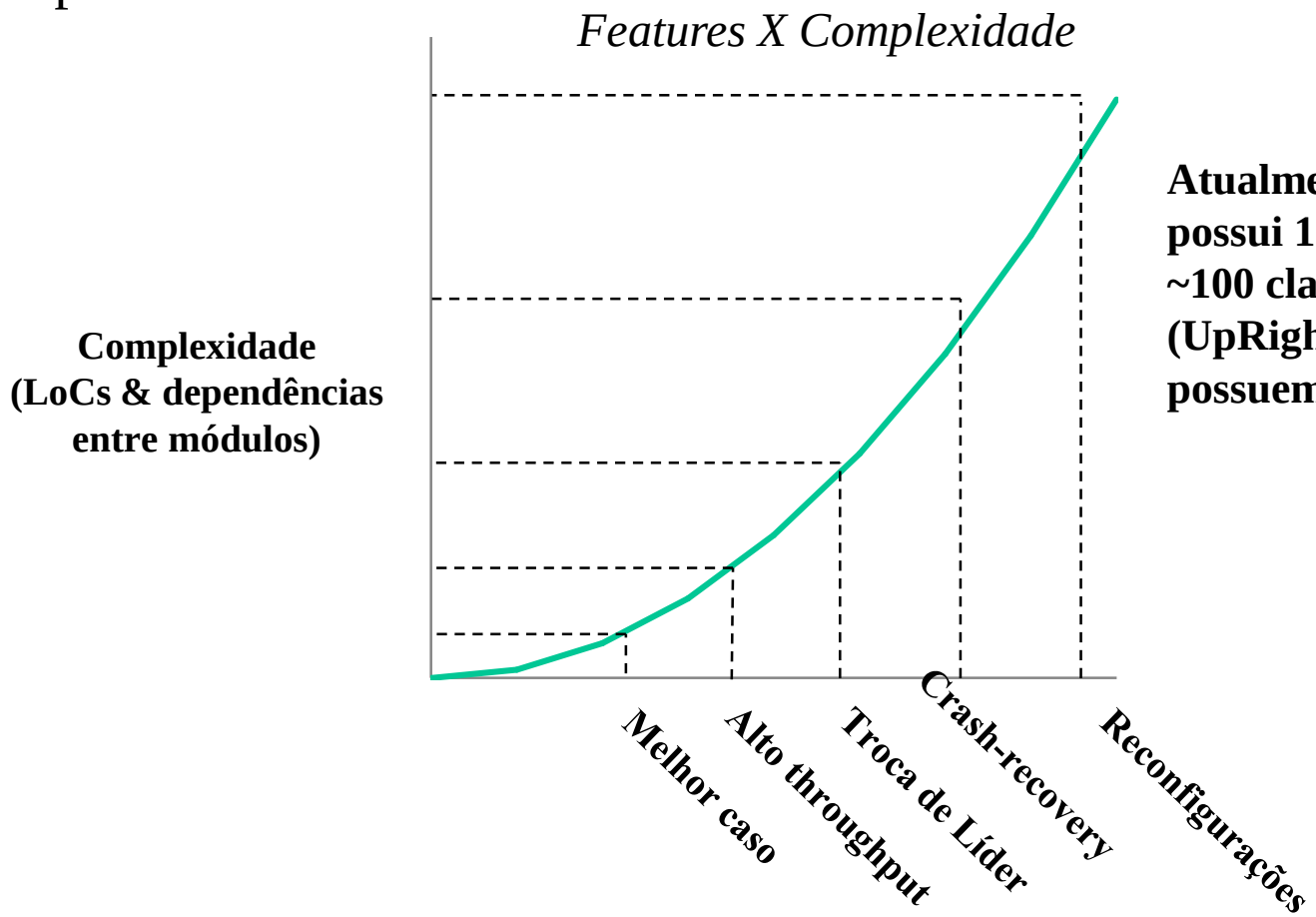
Determinação do Poço



BFT-SMaRt – funcionalidades e complexidade

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- Implementar RME não é trivial



Atualmente, o BFT-SMaRt possui 13Klocs e ~100 classes (UpRight and PBFT possuem ~20Klocs)

Avaliação de Desempenho

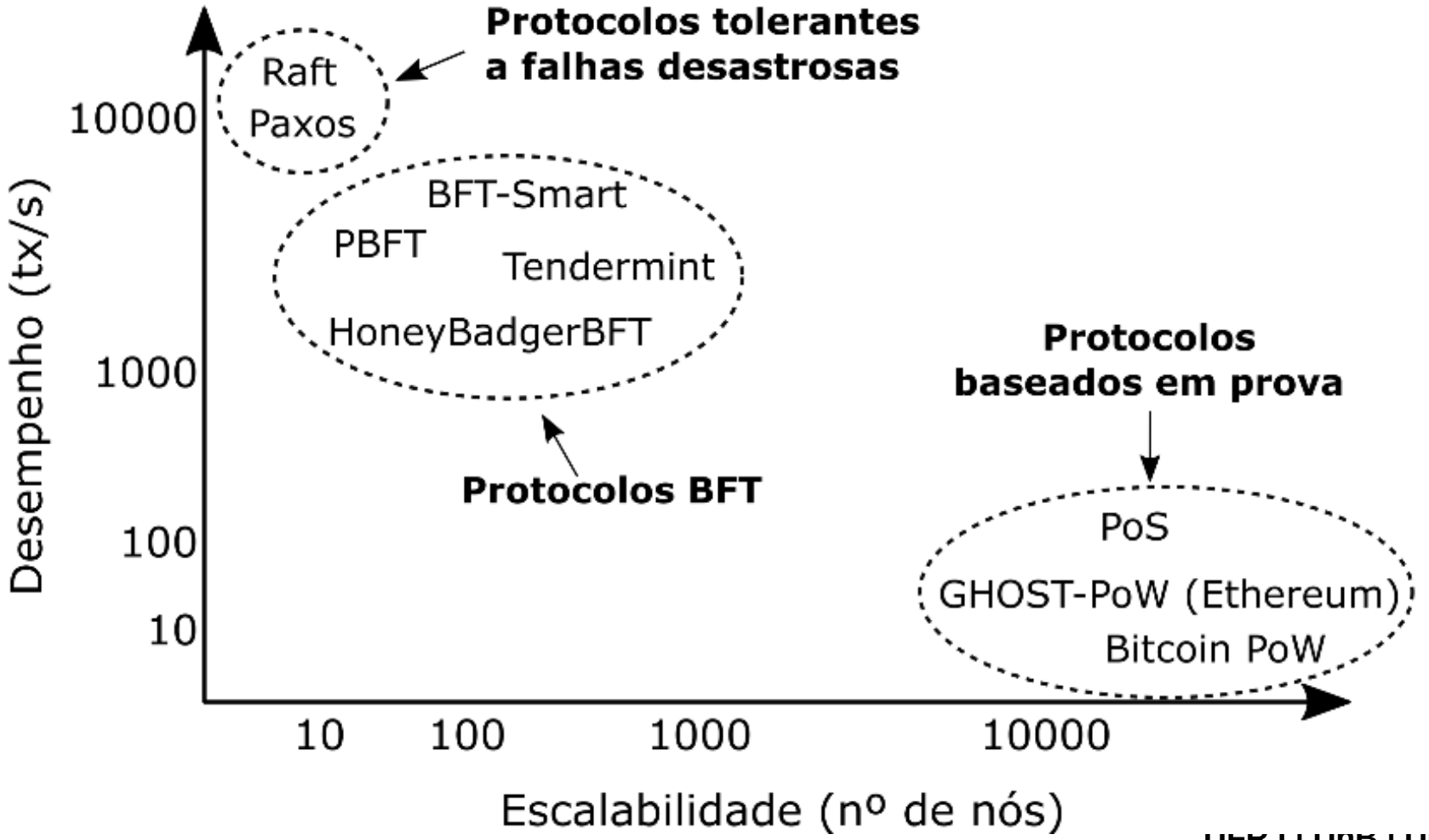
GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

Comparação com outros (*f* = *1*, requisição/resposta = 0/0)

Sistema	*Throughput*	*Clientes*	*Throughput 200*
BFT-SMaRt (BFT)	83801	1000	66665
PBFT	78765	100	65603
UpRight	5160	600	3355
BFT-SMaRt (CFT)	90909	600	83834
JPaxos	62847	800	45407

Comparação entre Protocolos de Consenso

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

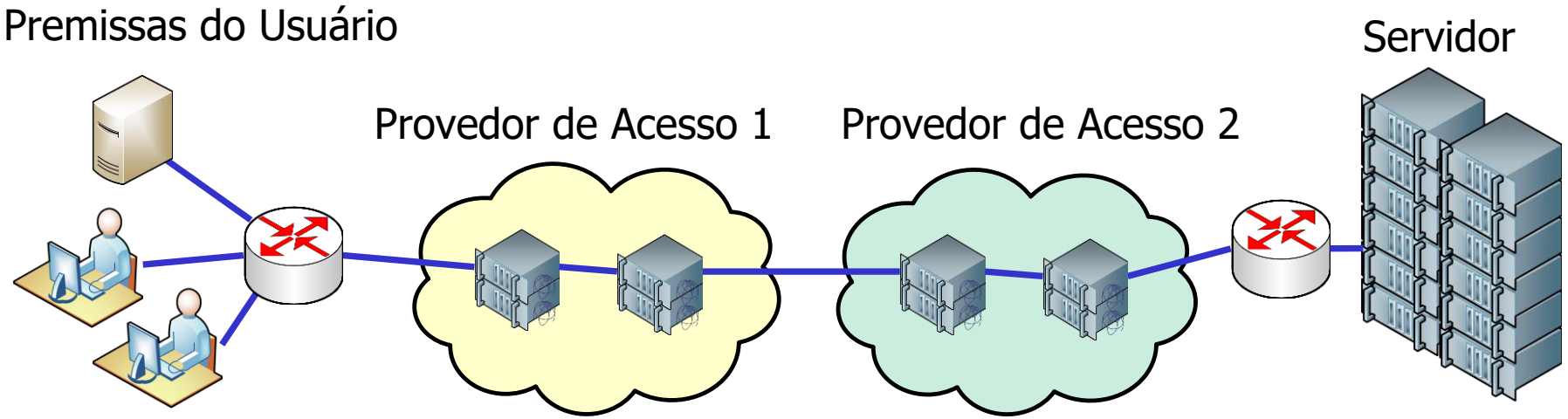


Parte III
-
Estudo de Caso
Uso de Corrente de Blocos em Telecomunicações

Redes de Computadores Tradicionais

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- Oferta de serviços fim-a-fim
- Serviços providos por equipamentos intermediários (*middleware* - *hardware*)
  - Firewall, balanceador de carga, balanceador de carga



Desafio das Redes de Computadores Tradicionais

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- Dificuldade de instalar novos serviços de rede
  - Altos custos com espaço físico e instalação
  - Longo tempo até chegar ao mercado → **ordem de anos**

Software como uma solução

  - Capacitação de recursos humanos

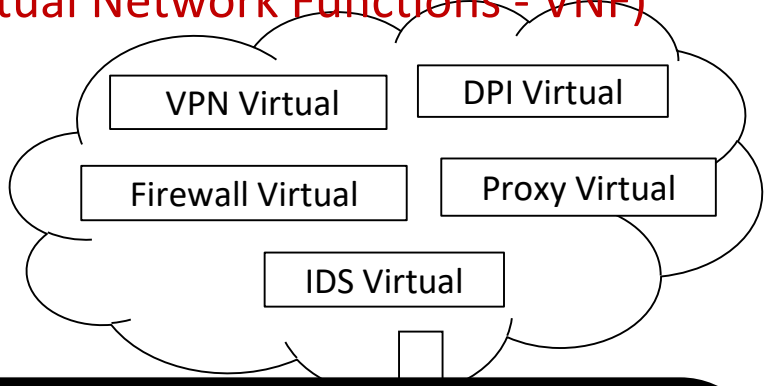
Alto custo de operação (OpEx)

Virtualização de Funções de Rede (Network Function Virtualization - NFV)

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

Funções Dedicadas de Rede

Funções Virtuais de Rede (Virtual Network Functions - VNF)



Objetivo: Reimplementar em ***software*** os serviços baseados em ***hardware***

Hardware de Uso Geral

Encadeamento de Funções de Serviço (SFC)

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- Encadeamento de funções de *hardware*

Agilidade, Flexibilidade e Baixo Custo

permite **alocar** ou **migrar** funções de rede de acordo com os

requisitos do serviço

Escalabilidade, agilidade e flexibilidade sem precedentes

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

Virtualização de Funções de Rede (NFV) visa o núcleo da rede

- oferecer programabilidade do núcleo da rede

→ **riscos de segurança**

- Cadeias de Serviço de Rede (SFC) comprometidas
- afetam todo o tráfego encaminhado pelo encadeamento
  - alto número de vítimas
  - dificuldade de identificar e de rastrear os ataques

- Mais fácil de implantar e de gerenciar
- **Mais difícil de garantir segurança e de auditar**

Problema de segurança

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

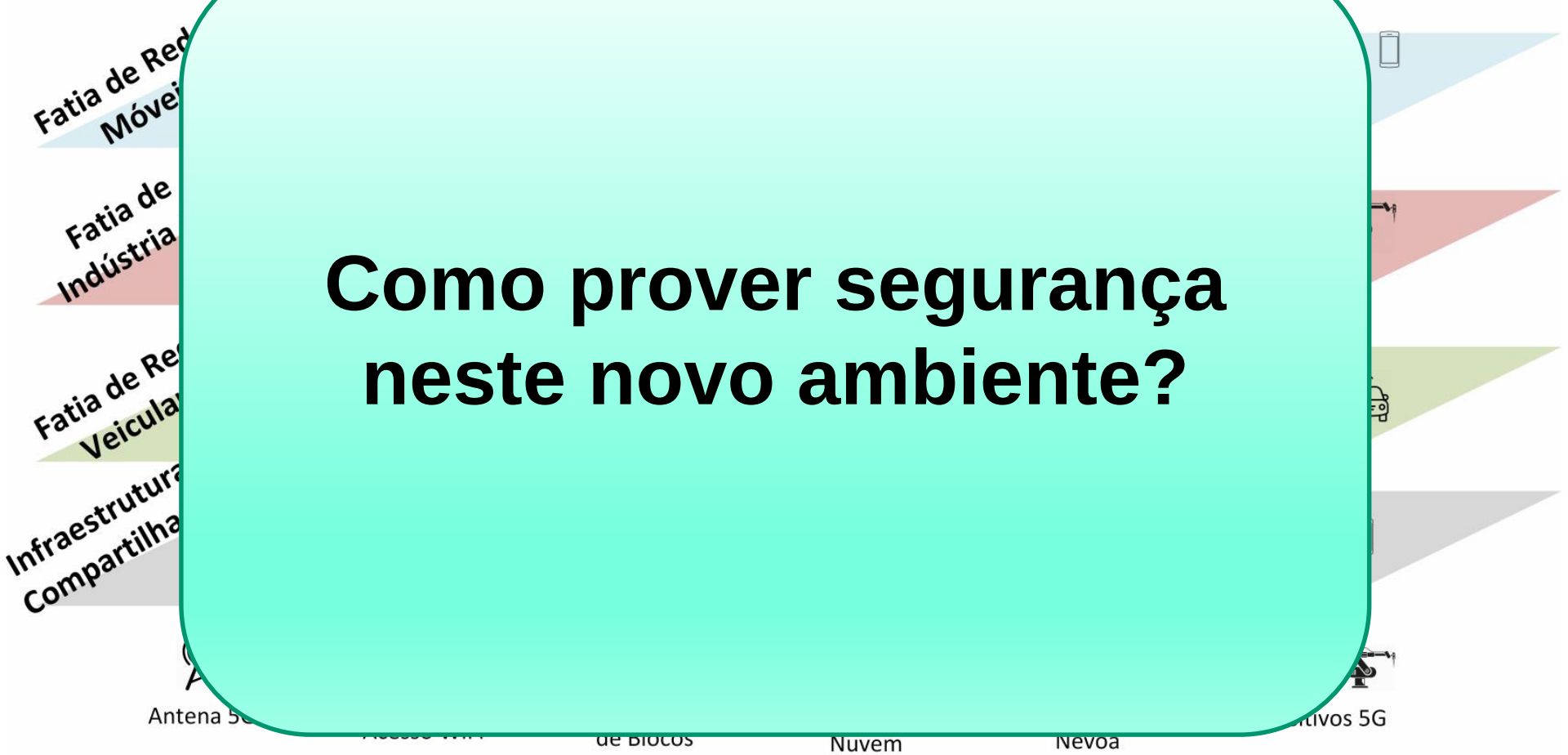
- A Internet do Futuro usará as tecnologias de virtualização de funções de rede (*Network Function Virtua* NFV) e fatiamento de rede (*Network Slic*
- Um enc *Cha* as a
- As f dife Situa uma prejudicar a outra

Como prover segurança neste novo ambiente?

Fatiamento Seguro de Rede

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- Cada fatia é protegida por uma corrente de blocos **adaptada**

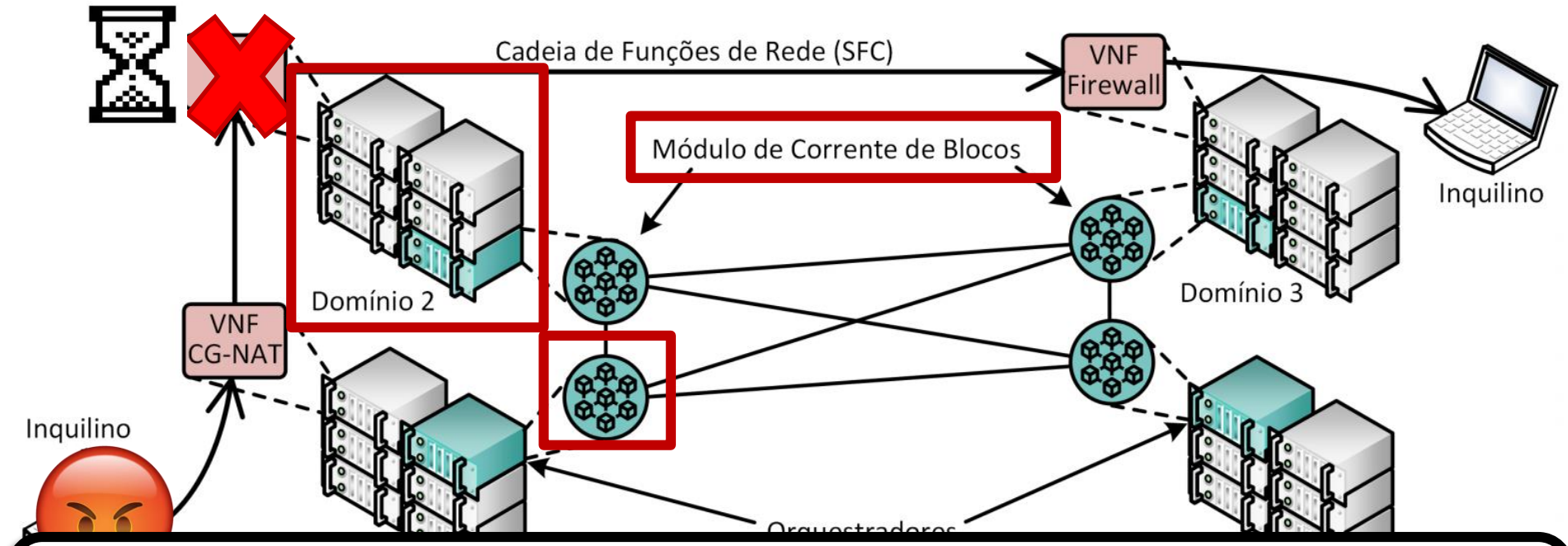


Como prover segurança neste novo ambiente?

Cenário de Utilização
não há confiança mútua

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- **Comunicação fim-a-fim entre inquilinos**



De quem é a culpa em caso de falha?

Problema de responsabilização

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- Como identificar a fatia que falhou?
- Qual função virtual de rede que falhou?
- Como responsabilizar a empresa que cometeu e é responsável pela falha?

Requisitos Arquiteturais

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- Dificultar ao máximo a descoberta
  - Localização, identidade e proprietário
  - Identidade e localização do inqulilino (*tenant*)
- Pedidos de configurações devem ser
  - confidenciais → uso de criptografia de chave pública
  - autenticados → uso assinatura criptográfica
  - anônimos → clientes identificados pela chave pública
  - imutáveis → garantido p/ propriedade da corrente de blocos
  - permanentes → garantido pela replicação da corrente de blocos
  - traçáveis → resolvido pelo esquema de transações proposto

Proposta de Solução do Problema

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

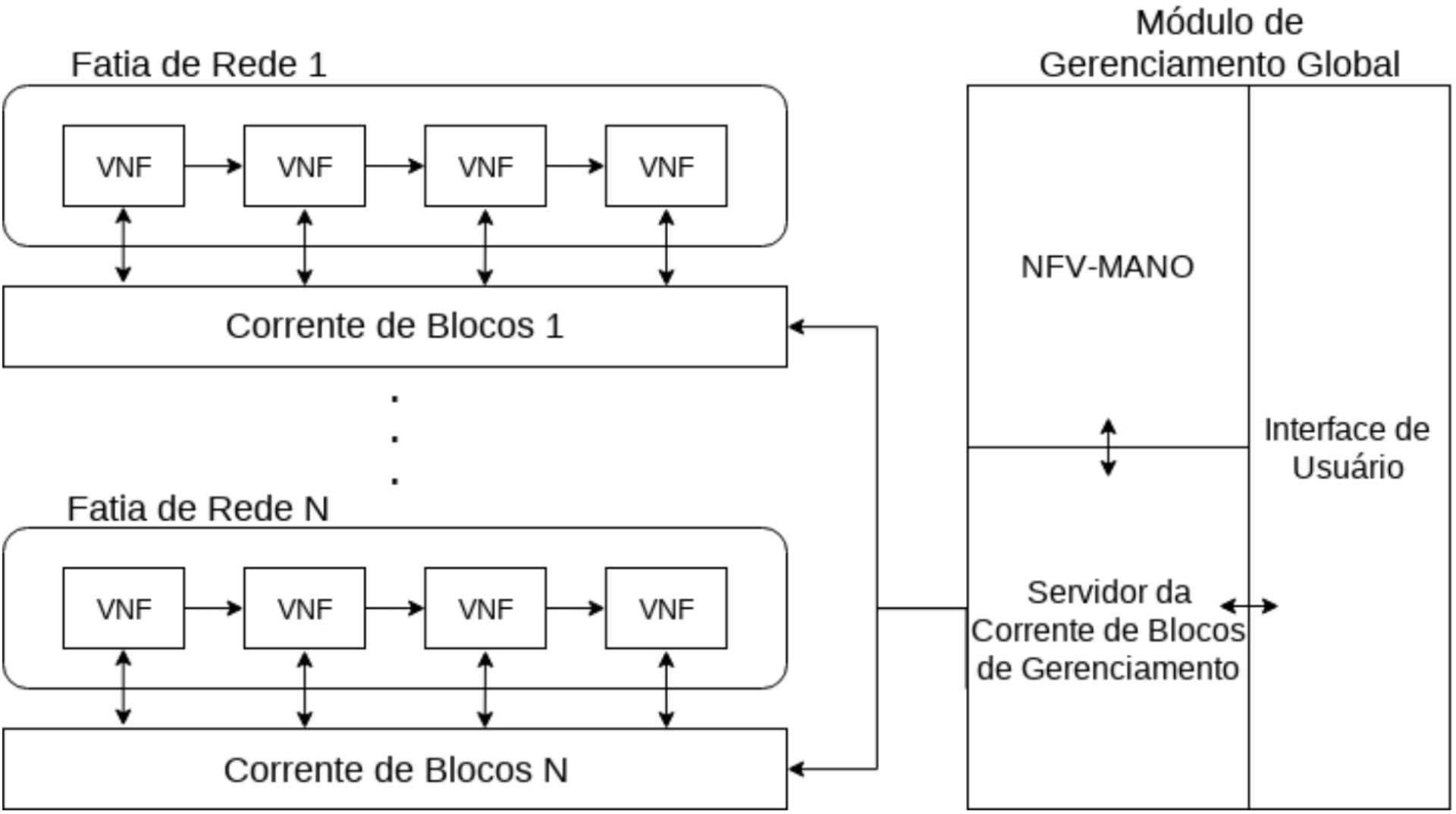
- Descobrir a empresa responsável p
  funcionamento
  - Registrar de forma imutável as ope
    rede
  - Garantir a análise forense de todos comandos enviados para as funções de rede virtualizadas
- Descobrir no encadeamento de funções qual(is) função(ões)
  - Registrar de forma imutável as o
    virtualizadas de rede
  - Garantir a análise forense das fu

Prover uma corrente de blocos para o sistema de segurança inteiro

Prover uma corrente de blocos por fatia de rede

A Arquitetura Proposta

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO



A Arquitetura Proposta – dois tipos de corrente de blocos

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- Uma corrente de blocos de gerenciamento de fatias
  - Registra os comandos de criação e modificação de fatias
- Uma corrente de blocos adaptada para cada fatia
  - Registra as informações relevantes à fatia

Dois artigos

Rebello, G. A. F., Camilo, G. F., Silva, L. G. C., Guimarães, L. C. B., Souza, L. A. C., Alvarenga, I. D. and Duarte, O. C. M. B. - "Provendo uma Infraestrutura de Software Fatiada, Isolada e Segura de Funções Virtuais através da Tecnologia de Corrente de Blocos", in II Workshop em Blockchain: Teoria, Tecnologias e Aplicações (WBlockchain SBRC 2019), Gramado, Brazil, May 2019.

Rebello, G. A. F., Camilo, G. F., Silva, L. G. C., Guimarães, L. C. B., Souza, L. A. C., Alvarenga, I. D. and Duarte, O. C. M. B. - "Providing a Sliced, Secure, and Isolated Software Infrastructure of Virtual Functions Through Blockchain Technology", in IEEE International Conference on High Performance Switching and Routing (HPSR 2019), Xi'An, China, May 2019.

Tipos de proteção por corrente de blocos

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

I) fatias de um único domínio e administrador único

- controle de permissão e consenso **centralizados**
  - E.g. corrente de blocos registra de forma imutável operações nas fatias de rede para análise forense posterior

II) fatias multi-domínios com provedores "confiáveis"

- tolerantes a falhas desastrosas
  - correntes de blocos com administração **permissionada** baseada em consórcio
  - participantes podem ter falhas de parada (*crash*)
  - tipo de consenso: protocolos tolerantes a falhas desastrosas (*Crash Fault Tolerant* - CFT)

Tipos de proteção por corrente de blocos

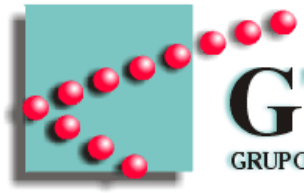
GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

III) fatias multi-domínio - provedores sem confiança mútua

- tolerantes a falhas bizantinas
- participantes podem falhar **de forma maliciosa**
  - Não responder, mentir, deletar mensagens etc.
- administração **permissionada** baseada em consórcio
- tipo de Consenso: protocolos tolerantes a falhas bizantinas (BFT)

IV) Fatias públicas escaláveis

- administração **não-permissionada** e distribuída
- participantes podem falhar **de forma maliciosa**
- tipo de consenso: protocolos baseados em prova (PoX)
  - E.g. fatias de rede de Internet das Coisas



GTA / UFRJ

GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

Desenvolvimento e implementação da solução usando a Plataforma Hyperledger Fabric

Hyperledger Fabric

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- Plataforma da IBM para desenvolver correntes de blocos permissionadas
  - Apropriada para ambientes empresariais
  - Tipicamente constituídos por até dezenas de nós
- Arquitetura modular que permite o uso de diferentes
  - Protocolos de consenso
  - Linguagens de programação para criar contratos inteligentes
  - Bancos de dados para armazenar a corrente de blocos e seu estado global

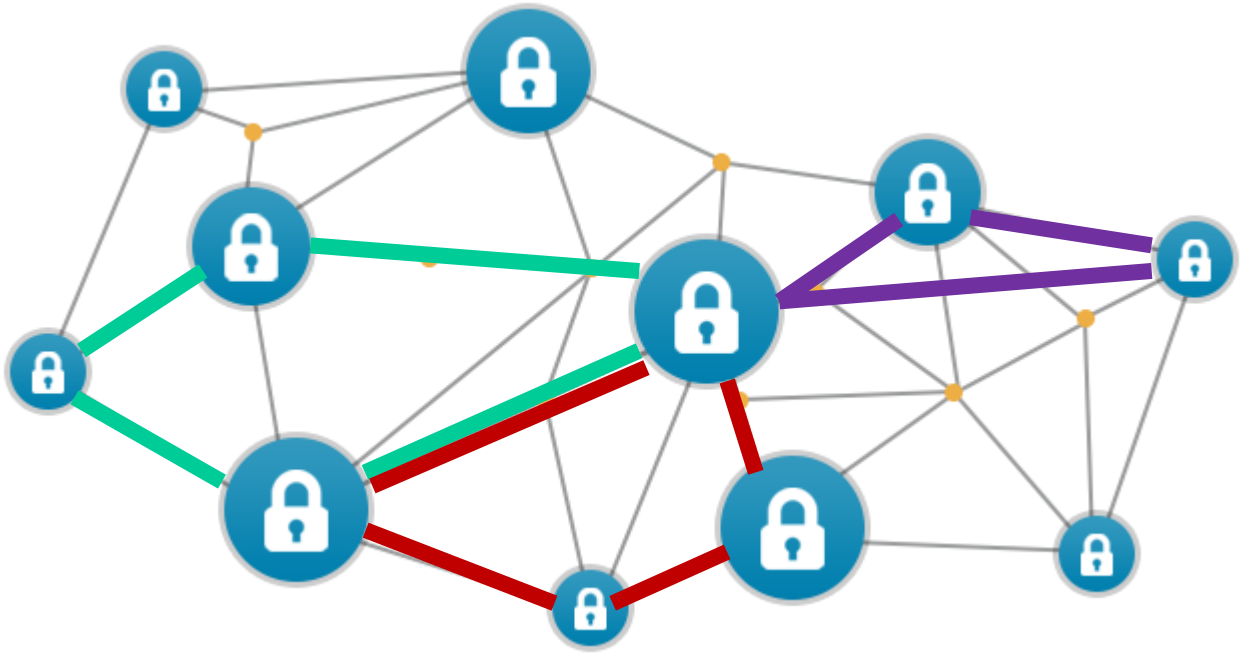
HYPERLEDGER

HYPERLEDGER
FABRIC

A Arquitetura do Fabric

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

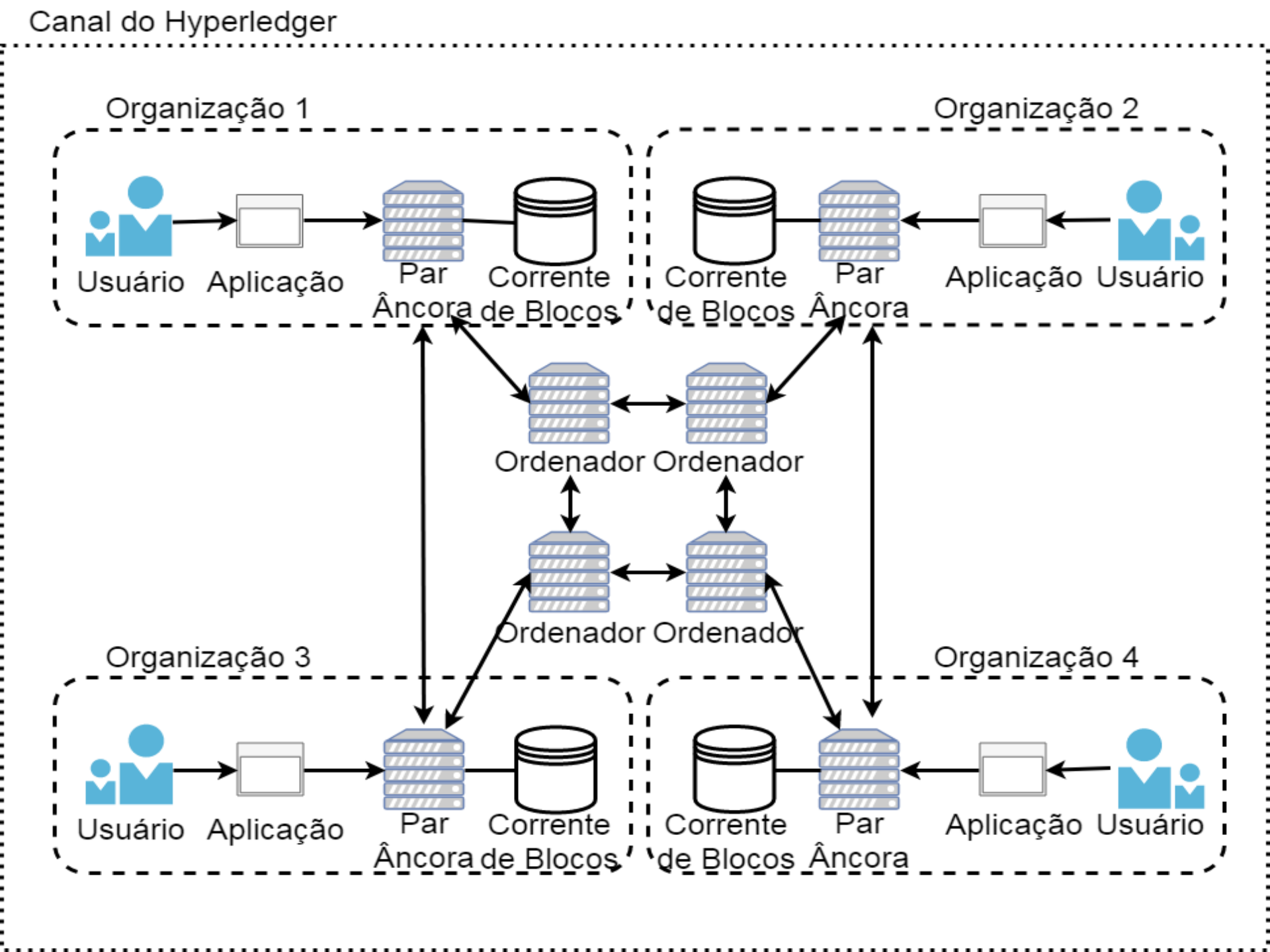
- Consórcio de organizações
  - Cada consórcio possui uma blockchain e um canal de comunicação isolados



A Arquitetura do Fabric

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- **Organização:** entidade interessada na troca de recursos
  - Cada organização possui
    - **Pares:** máquinas que armazenam a corrente de blocos
    - **Pares-âncora:** pares que servem de *gateway* para comunicação com outras organizações
- **Canal:** meio de comunicação entre organizações
  - Cada canal é isolado através de criptografia
- **Ordenador**: entidade responsável por ordenar as transações através de consenso
  - Cada organização associa-se a um ordenador


Canal do Hyperledger
Organização 1
Usuário
Aplicação
Par
Âncora
Corrente
de Blocos
Organização 2
Corrente
de Blocos
Par
Âncora
Aplicação
Usuário
Ordenador
Ordenador
Ordenador
Ordenador
Organização 3
Usuário
Aplicação
Par
Âncora
Corrente
de Blocos
Organização 4
Corrente
de Blocos
Par
Âncora
Aplicação
Usuário

Estudo de Caso: Criação de uma Corrente de Blocos Permissionada

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- Implementação de uma corrente de blocos na plataforma Hyperledger Fabric
  - Cada par é um contêiner em um único computador
  - Implementa-se um contrato inteligente
    - Executado em todos os nós da rede
    - Visível a todos os nós da rede

Participantes da Rede

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- O Hyperledger Fabric permite que participantes possuam diferentes permissões de acesso à corrente
  - Administrador → Transações de configuração
    - Modifica permissões de leitura e escrita dos outros participantes
  - Usuário comum → Transações comuns
    - Transferência de ativos entre duas entidades

Participantes da Rede

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- Entidades que compõem a corrente de blocos do Fabric; podem ser:
  - Clientes → Usuários que enviam transações aos pares para validação e assinatura
  - Pares (*peers*) → Validam e executam as transações; também mantém os registros da corrente de blocos
  - Ordenadores → Estabelecem a ordem e o empacotamento de todas as transações em um bloco
    - Feito através de um protocolo de consenso

Canais

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- Sub-rede de comunicação entre um conjunto de nós
  - Fornece privacidade e confidencialidade às transações
    - Dados transmitidos são sigilosos e inacessíveis a qualquer entidade externa ao canal
    - Mensagens trocadas são criptografadas

- Canais podem possuir diferentes configurações, incluindo:
  - Formato dos blocos
  - Formato das transações
    - Definido através de contratos inteligentes
  - Protocolo de consenso

Contratos Inteligentes (*smart contracts*)

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- Protocolo executado por todos os nós da rede
- O contrato implementado realiza a orquestração de VNFs através de dois tipos de transação
  - Transação de instrução
    - Enviada quando o cliente solicita uma fatia
  - Transação de resposta
    - Enviada após a instrução recebida ser executada

Pseudocódigo para Transações de Instrução

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

```
struct instructionTransaction
{
    command                 string
    transactionType         string
    transactionName         string
    issuer                  string
}
initialize queue
initInstruction (instruction <command,name,issuer>)
{
    if instruction is not unique or instruction is not well-formatted:
        return error
    putState (instruction.name, instruction)
    put (transactionID, queue)
    notify orchestrator
    return success
}
```

Parte IV: Atividade Prática

Material da Aula Prática

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- Recursos
  - Notebook com >4GB de RAM
  - Virtualbox 6.0+
  - Máquina virtual com Debian 9.0+
- Dontpad: www.dntpad.com/jai-2019
- Uma VM com todos os pacotes instalados está disponível em pen-drives

Considerações sobre a Atividade Prática

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- Desenvolvida utilizando a versão 1.4.1 do Hyperledger Fabric
  - Comandos executados podem ser incompatíveis com outras versões
- A imagem de máquina virtual disponibilizada contém os pré-requisitos da atividade instalados; estes incluem:
  - Docker e Docker Compose, utilizados para a criar e executar múltiplos contêineres
  - A linguagem de programação Go, utilizada pela plataforma Hyperledger Fabric

Pré-preparo
Instalando o VirtualBox

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- Acessar o

  site: https://www.virtualbox.org/wiki/Downloads
- Selecionar a opção “Windows hosts”
- Baixar e execute o arquivo .exe
- Após executar o arquivo:
  - Selecionar “Next” para avançar pelo processo de setup
  - Selecionar “Yes” quando aparecer o aviso sobre interfaces de rede
  - Selecionar “Install” para iniciar a instalação
  - Selecionar "Finish" para concluir a instalação

Pré-preparo
Importando a Imagem

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- Baixar a imagem utilizada pela atividade prática: https://www.gta.ufrj.br/JAI2019.ova
- Iniciar o VirtualBox
- Selecionar a opção “Importar”
- Selecionar a imagem baixada anteriormente
- Clicar em “Next”, e então em “Import” para importar a imagem

Principais etapas da aula prática

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

1. Gerar os certificados das chaves
2. Configurar a corrente de blocos e criar o bloco gênesis
3. Definir as configurações do canal
4. Definir os pares-âncoras das organizações
5. Instanciar contêineres
6. Criar e incluir os pares no canal
7. Instalar e instanciar o contrato inteligente
8. Definir o caminho dos certificados
9. Gerar uma transação
10. Consultar a corrente de blocos

Instalação do Hyperledger Fabric

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- Todos os comandos a seguir devem ser executados como super usuário. O ambiente da máquina virtual disponibilizada já possui o Hyperledger Fabric instalado. A senha da máquina virtual disponibilizada é "JAI2019".

```
su
cd ~/fabric-samples/first-network
```

Geração dos Certificados

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- O arquivo crypto-config.yaml contém informações dos participantes iniciais da rede, como:
  - Endereço de cada nó
  - Número de ordenadores
  - Organizações participantes
  - Quantidade de participantes por organização

- Informações utilizadas pela ferramenta cryptogen
  - Gera pares de chaves assimétricas e certificados para os nós

Geração dos Certificados

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- O comando abaixo gera as chaves e os certificados que serão usados na rede, através do arquivo crypto-config.yaml. Ele deve ser executado de dentro do diretório `first-network`

```
../bin/cryptogen generate --config=./crypto-config.yaml
```

Configurações da Corrente de Blocos

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- O arquivo configtx.yaml também contém informações sensíveis a rede, como:
  - Tipo de consenso utilizado
  - Tempo máximo para criação de um novo bloco
  - Tamanho máximo das transações de um bloco
  - Tamanho máximo do bloco
  - Políticas para validação de transações

- Informações utilizadas pela ferramenta configtxgen
  - Cria o bloco gênesis que contém as informações de configuração da rede

Criação do Bloco Gênesis

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- Os comandos abaixo geram o bloco gênesis, contendo as configurações da corrente de blocos criada

```
export FABRIC_CFG_PATH=$PWD
../bin/configtxgen -profile SampleDevModeKafka -
channelID byfn-sys-channel –outputBlock
./channel-artifacts/genesis.block
```

Definição das Configurações do Canal

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- O comando abaixo cria um arquivo channel.tx contendo as configurações do canal

```
export CHANNEL_NAME=jaichannel &&
../bin/configtxgen -profile TwoOrgsChannel  -
outputCreateChannelTx ./channel-
artifacts/channel.tx -channelID $CHANNEL_NAME
```

Pares-Âncora (*anchor peers*)

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- Responsáveis por realizar a conexão entre organizações
  - Cada organização possui ao menos um par-âncora
  - Pares de uma organização conectam-se a pares-âncora para descobrir pares de outra organização

Definição dos Pares-Âncora

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- Os comandos abaixo definem os pares-âncora das duas organizações criadas nesta atividade

```
../bin/configtxgen -profile TwoOrgsChannel -
outputAnchorPeersUpdate ./channel-
artifacts/Org1MSPanchors.tx -channelID
$CHANNEL_NAME -asOrg Org1MSP
../bin/configtxgen -profile TwoOrgsChannel -
outputAnchorPeersUpdate ./channel-
artifacts/Org2MSPanchors.tx -channelID
$CHANNEL_NAME -asOrg Org2MSP
```

Instanciação dos Contêineres

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- O comando abaixo utiliza a ferramenta Docker Compose para criar os participantes da rede
  - Cada participante é implementado como um contêiner
  - Utiliza-se o arquivo de configuração docker-compose-kafka.yaml para facilitar a configuração e instanciação dos contêineres

```
docker-compose -f docker-compose-cli.yaml -f
docker-compose-kafka.yaml up -d
```

Criação do Canal

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- Os comandos abaixo acessam o contêiner cliente e criam o canal com as configurações definidas anteriormente

```
docker exec -it cli bash
export CHANNEL_NAME=jaichannel
peer channel create -o orderer.example.com:7050 -
c $CHANNEL_NAME -f ./channel-artifacts/channel.tx
--tls --cafile
/opt/gopath/src/github.com/hyperledger/fabric/pee
r/crypto/ordererOrganizations/example.com/orderer
s/orderer.example.com/msp/tlscacerts/tlsca.exampl
e.com-cert.pem
```

Inserção dos 4 Pares no Canal

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- Os comandos abaixo adicionam o peer0 da organização org1 no canal jaichannel

```
source scripts/peer0_org1.sh
peer channel join -b jaichannel.block
```

- Para adicionar o peer1 da organização org1 no canal, execute estes comandos

```
source scripts/peer1_org1.sh
peer channel join -b jaichannel.block
```

Inserção dos 4 Pares no Canal

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- Os comandos abaixo adicionam o peer0 da organização org2 no canal jaichannel

```
source scripts/peer0_org2.sh
peer channel join -b jaichannel.block
```

- Para adicionar o peer1 da organização org2 no canal, execute estes comandos

```
source scripts/peer1_org2.sh
peer channel join -b jaichannel.block
```

Configurando os Pares-Âncora das Organizações

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- Anteriormente, definimos o par peer0 da organização org1 como um par-âncora
- Os comandos abaixo atualizam o canal jaichannel com as informações do par-âncora da organização org1

```
source scripts/peer0_org1.sh
peer channel update -o orderer.example.com:7050 -
c $CHANNEL_NAME -f ./channel-
artifacts/Org1MSPanchors.tx --tls --cafile
/opt/gopath/src/github.com/hyperledger/fabric/pee
r/crypto/ordererOrganizations/example.com/orderer
s/orderer.example.com/msp/tlscacerts/tlsca.exampl
e.com-cert.pem
```

Configurando os Pares-Âncora das Organizações

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- Para a organização org2, definimos o par peer0 como um par-âncora
- Os comandos abaixo atualizam o canal jaichannel com as informações do par-âncora da organização org2

```
source scripts/peer0_org2.sh
peer channel update -o orderer.example.com:7050 -c $CHANNEL_NAME -f ./channel-artifacts/Org2MSPanchors.tx --tls --cafile /opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.com-cert.pem
```

Instalação do Contrato Inteligente

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- O contrato deve ser instalado em **todos** os pares que executam e validam transações
- Os comandos abaixo instalam o contrato inteligente no par peer0 da organização org1

```
source scripts/peer0_org1.sh
```

```
go get -u github.com/golang-collections/go-datastructures/queue && peer chaincode install -n mycc -v 1.0 -p github.com/chaincode/contract
```

Instalação do Contrato Inteligente

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- Os comandos abaixo instalam o contrato inteligente no par peer1 da organização org1

```
source scripts/peer1_org1.sh

go get -u github.com/golang-collections/go-datastructures/queue && peer chaincode install -n mycc -v 1.0 -p github.com/chaincode/contract
```

Instalação do Contrato Inteligente

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- Os comandos abaixo instalam o contrato inteligente no par peer0 da organização org2

```
source scripts/peer0_org2.sh
```

```
go get -u github.com/golang-collections/go-datastructures/queue && peer chaincode install -n mycc -v 1.0 -p github.com/chaincode/contract
```

Instalação do Contrato Inteligente

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- Os comandos abaixo instalam o contrato inteligente no par peer1 da organização org2

```
source scripts/peer1_org2.sh

go get -u github.com/golang-collections/go-datastructures/queue && peer chaincode install -n mycc -v 1.0 -p github.com/chaincode/contract
```

Instanciação do Contrato Inteligente

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- O comando abaixa instancia o contrato na rede

```
peer chaincode instantiate -o
orderer.example.com:7050 --tls --cafile
/opt/gopath/src/github.com/hyperledger/fabric/pe
er/crypto/ordererOrganizations/example.com/order
ers/orderer.example.com/msp/tlscacerts/tlsca.exa
mple.com-cert.pem -C $CHANNEL_NAME -n mycc -v
1.0 -c '{"Args":["init"]}' -P
"AND('Org1MSP.peer','Org2MSP.peer')"
```

Definindo os Caminhos dos Certificados

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

```
CERTIFICADO_ORDENADOR=/opt/gopath/src/github.com
/hyperledger/fabric/peer/crypto/ordererOrganizat
ions/example.com/orderers/orderer.example.com/ms
p/tlscacerts/tlsca.example.com-cert.pem
CERTIFICADO_PEER_ORG1=/opt/gopath/src/github.com
/hyperledger/fabric/peer/crypto/peerOrganization
s/org1.example.com/peers/peer0.org1.example.com/
tls/ca.crt
CERTIFICADO_PEER_ORG2=/opt/gopath/src/github.com
/hyperledger/fabric/peer/crypto/peerOrganization
s/org2.example.com/peers/peer0.org2.example.com/
tls/ca.crt
```

Geração de uma Transação

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- Comando `invoke`
  - Chama uma função definida no contrato inteligente
  - Emite uma transação

Geração de uma Transação

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

```
peer chaincode invoke -o
orderer.example.com:7050 --tls --
cafile $CERTIFICADO_ORDENADOR -C
$CHANNEL_NAME -n mycc --
peerAddresses peer0.org1.example.com:7051 --
tlsRootCertFiles $CERTIFICADO_PEER_ORG1 --
peerAddresses peer0.org2.example.com:9051 --
tlsRootCertFiles $CERTIFICADO_PEER_ORG2 -c
'{"Args":["initInstructionTransaction",
"transaction", "issuer", "instruction"]}'
```

Consultando a Corrente de Blocos

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- O Hyperledger Fabric permite serviços de busca e pesquisa do histórico de transações
  - Recebe como argumentos o nome do canal, o nome do contrato e a mensagem para o contrato em formato JSON
- O comando abaixo consulta o histórico da corrente criada no decorrer da atividade

```
peer chaincode query -C $CHANNEL_NAME -n mycc -c
'{"Args":["getHistoryForTransaction",
"transaction"]}'
```

Parte V
Conclusões e perspectivas

Desafios do prova de posse

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- Similar ao consenso de prova de trabalho mas sem o consumo energéticos
- Implementação complexa
- Pode apresentar vazões altas

Desafios do BFT

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- Anonimato
- Evitar a centralização de mineradores
- Escalabilidade
- Compromisso vazão vs escalabilidade
- Apropriado para pequenas redes e redes empresariais

Protocolos Híbridos

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- As propostas híbridas procuram as vantagens de cada protocolo mas, INFELIZMENTE, também herdam as DESVANTAGENS deles
- Diferentes propostas com ajustes para o que se quer privilegiar
- Compromisso desempenho vs centralização
- Outras alternativas inovadoras??????????????

Desafios de Vazão e Escalabilidade

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- É possível se alcançar a vazão de transações por segundo média de 2.000 e pico de 56.000 requerida pela VISA?
- É possível atender à Internet das Coisas, com bilhões de dispositivos realizando transações simultaneamente?

Economia compartilhada

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- Empresas que se servem de economia compartilhada deixarão de existir?
- AirB&B, Uber, eBAY, MercadoLivre, etc.

Normatização e legalização

GTA / UFRJ
GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

- International Organization for Standartization - ISO)
  - comitê técnico ISO/TC 307 2016
    - formalizar os riscos de segurança, ameaças e vulnerabilidades da tecnologia, além de normalizar a arquitetura de referência, taxonomia, contratos inteligentes e a proteção de privacidade e de informações pessoais.
- Internet Research Task Force (IRTF)
  - criou o grupo de pesquisa da infraestrutura descentralizada da Internet (De-centralized Internet Infrastructure Research Group - DINRG)
    - serviços de infraestrutura beneficiados pela descentralização
- International Telecommunications Union - ITU)
  - Focus Group on Applications of Distributed Ledger Technology - FG DLT)
- Europe Blockchain Working Group
  - Association Trade Communication - ISITC
    - discute a adoção da tecnologia de livro-razão