



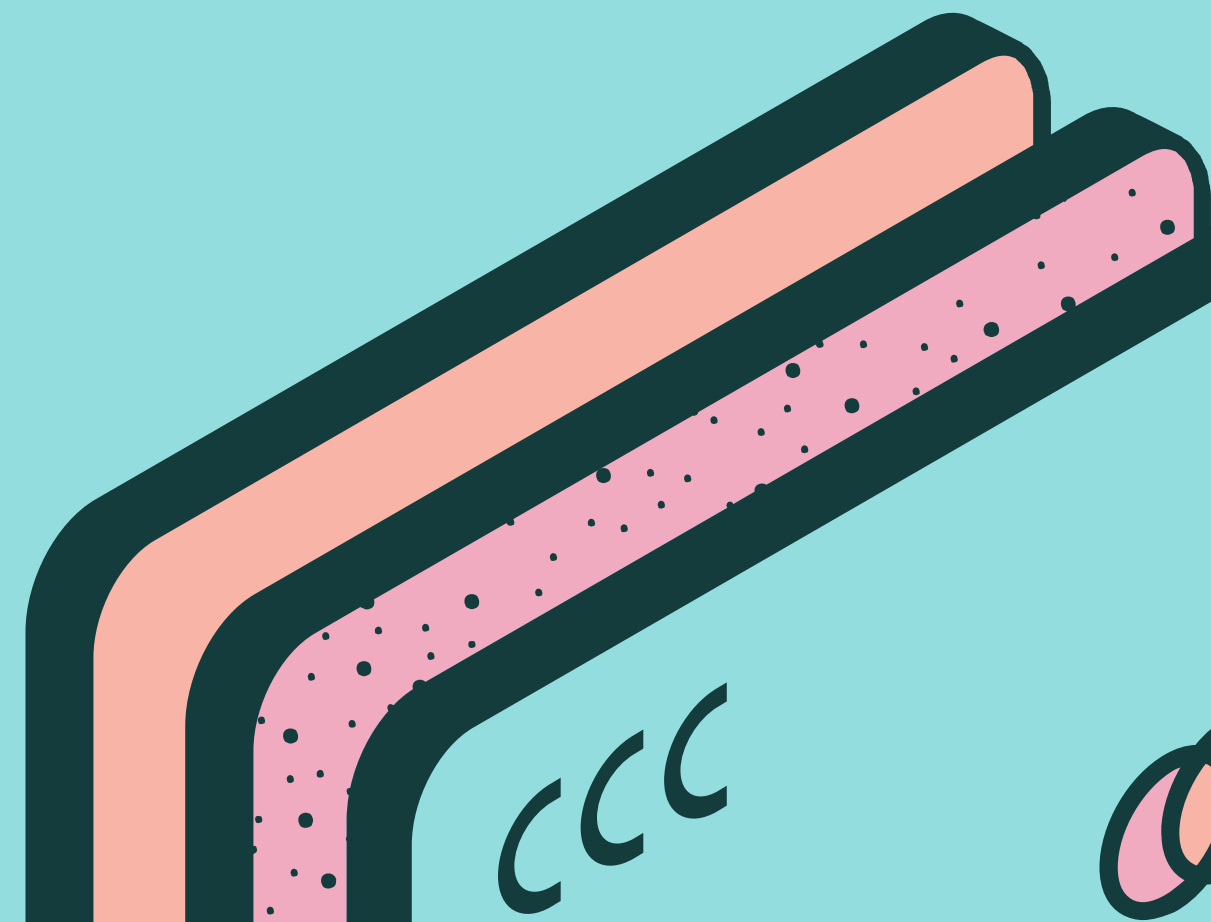
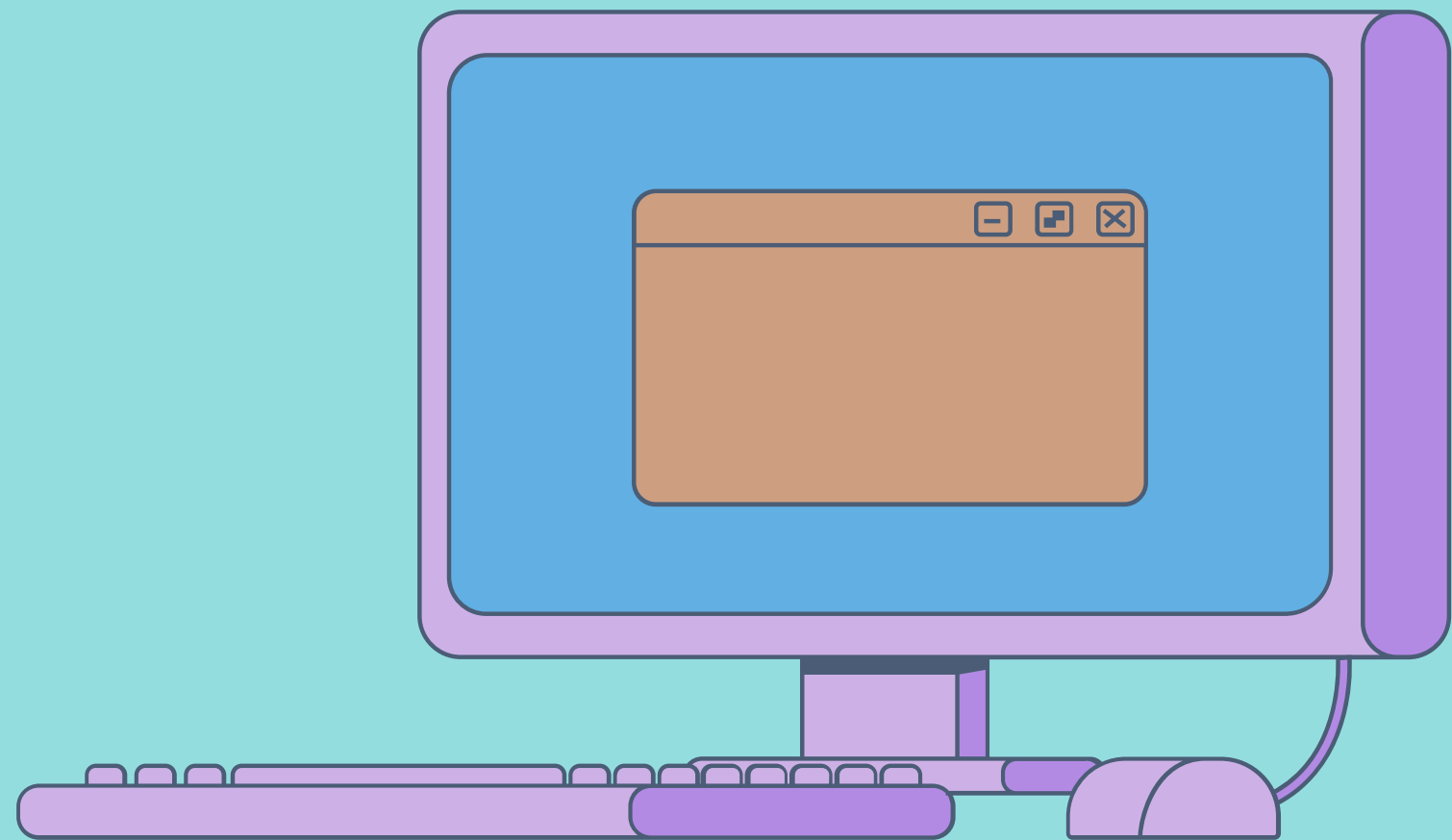
Integração de Sistemas Operacionais e Blockchain

O Desafio da Conciliação Bancária

Introdução

A integração de sistemas operacionais é vital para instituições financeiras, mas a conciliação entre eles enfrenta desafios.

Problemas como erros e atrasos nas transações são frequentes, devido à falta de comunicação entre os sistemas.



Problema



Dificuldades na conciliação são evidentes em transações financeiras com múltiplos sistemas, resultando em divergências nos registros.

Falta de uniformidade nos dados, protocolos e processos contribui para discrepâncias e atrasos na conciliação.

Solução - Blockchain



A implementação da tecnologia blockchain oferece uma solução promissora.

O blockchain proporciona consenso transparente e imutável, eliminando erros e atrasos na conciliação.

Por que Blockchain?



Consenso Distribuído

Consenso distribuído é essencial para garantir confiabilidade nas transações.

Mecanismos como Proof of Work (PoW) ou Proof of Stake (PoS) eliminam a necessidade de uma autoridade central.

Contratos Inteligentes

Contratos inteligentes são programas auto executáveis que automatizam processos de conciliação.

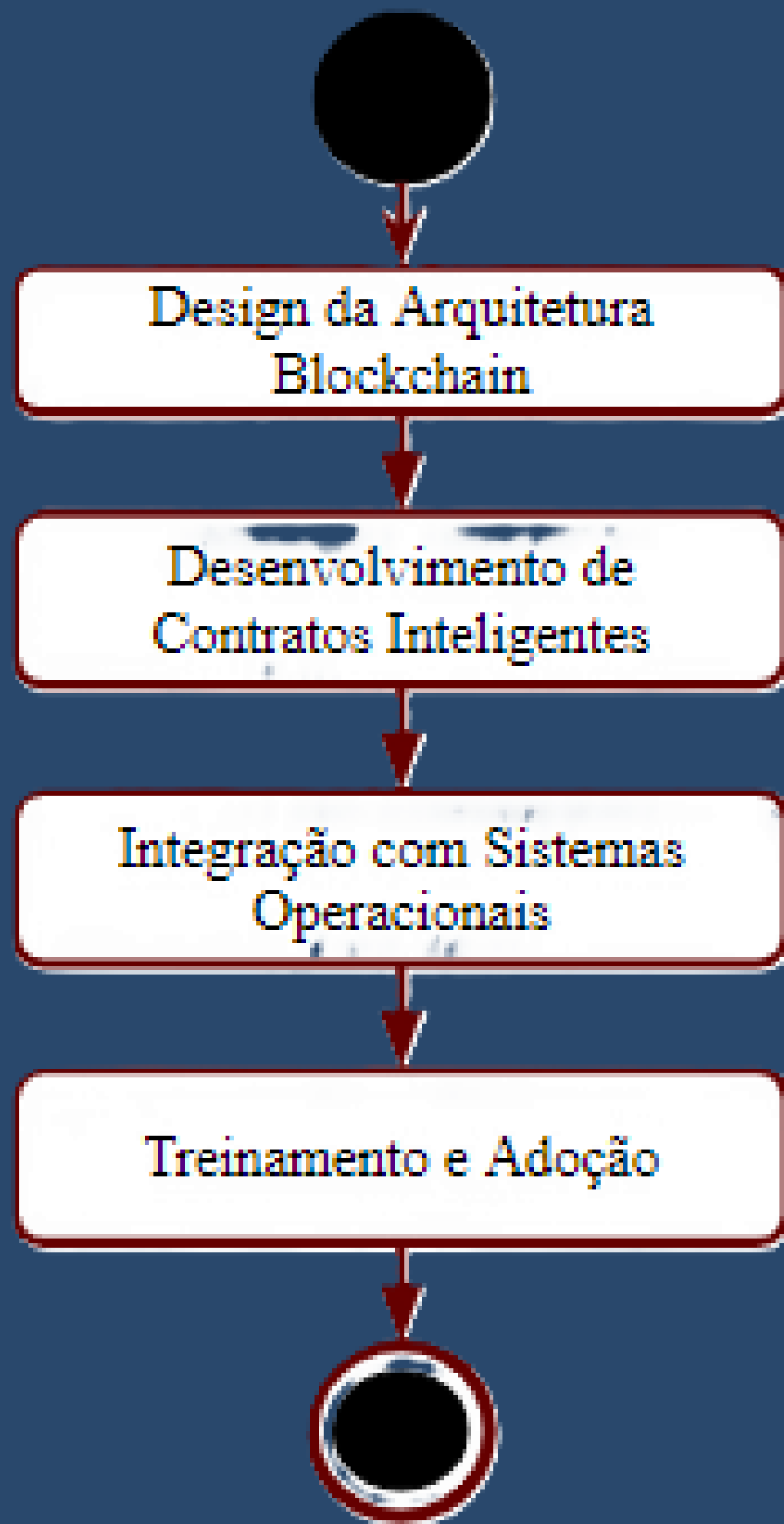
Eliminam ambiguidades e erros humanos, oferecendo uma abordagem confiável.

Criptografia e Segurança

Criptografia, com chaves públicas e privadas, garante a confidencialidade e integridade dos dados.

Blockchain utiliza esses mecanismos para assegurar transações seguras e imutáveis.

Fundamentos Teóricos



Design da arquitetura blockchain: Identificação de participantes, escolha do tipo de blockchain, definição de regras de consenso.

Desenvolvimento de contratos inteligentes: Especificação das condições, lógica de execução automática.

Desenvolvimento de interfaces e padrões de comunicação compatíveis com a rede blockchain.

Treinamento e adoção gradual da nova infraestrutura.

Mapeamento de um diagrama de fluxo

Metodologia

Conciliação eficiente

A implementação do middleware permite uma conciliação mais eficiente entre transações, reduzindo significativamente os erros e atrasos.

Registro Imutável

A utilização de contratos inteligentes na blockchain garante um registro imutável e transparente de todas as transações, aumentando a confiança e a segurança

Resultados e Conclusões

Redução de Custos e Melhoria na Eficiência

A integração eficaz entre sistemas operacionais resulta em uma redução de custos operacionais, ao mesmo tempo em que melhora a eficiência nas operações financeiras.

Benefícios no segmento financeiro

- Aumentar a transparência
- Fortalecer a confiança nas transações
- Servir de modelo para outras instituições com desafios similares

Resultados e Conclusões

Código exemplificando a execução de um contrato inteligente utilizando a linguagem Solidity.

```
// Versão da linguagem utilizada
pragma solidity ^0.8.0;

// Contrato para a transferência
automática entre contas a partir de uma
condição determinada

contract ContratoTransferencia {
    address public proprietario;
    //endereço da conta do proprietário
    address public destinatario;
    //endereço da conta do destinatário
    uint public montante;
    bool public condicaoAtendida;

    // Evento para notificar a execução
da transferência
    event TransferenciaRealizada(address
remetente, address destinatario, uint
montante);

    // Modificador que verifica se a
condição foi atendida
    modifier condicaoFoiAtendida() {
        require(condicaoAtendida, "A
condição não foi atendida.");
        _;
    }

    // Construtor do contrato
    constructor(address _destinatario,
uint _montante) {
        proprietario = msg.sender;
        destinatario = _destinatario;
        montante = _montante;
        condicaoAtendida = false;
    }

    // Função para atender à condição e
realizar a transferência
    function atenderCondicao() public {
```

```
        require(msg.sender == proprietario,
"Somente o proprietário pode atender à
condição.");
        condicaoAtendida = true;

        realizarTransferencia(proprietario,
destinatario, montante);

        emit TransferenciaRealizada(proprietario,
destinatario, montante);
    }

    // Função para realizar a
transferência quando a condição foi
atendida
    function realizarTransferencia()
public condicaoFoiAtendida {
        require(address(this).balance >=
montante, "Saldo insuficiente no
contrato.");

        payable(destinatario).transfer(montante);
        condicaoAtendida = false; //
Reinicia a condição para futuras
transferências
    }
}
```

- As funções podem e são executadas em diferentes nós da rede blockchain.
- Cada transação/função é executada no nó que a inicia.
- As verificações de condições e estado do contrato é compartilhado entre todos os nós.
- Condição não atendida leva a reversão da transação.

Resultados e Conclusões

Perguntas e Discussão

Integração de Sistemas
Operacionais e Blockchain
O Desafio da Conciliação



Obrigado pela
atenção!

