

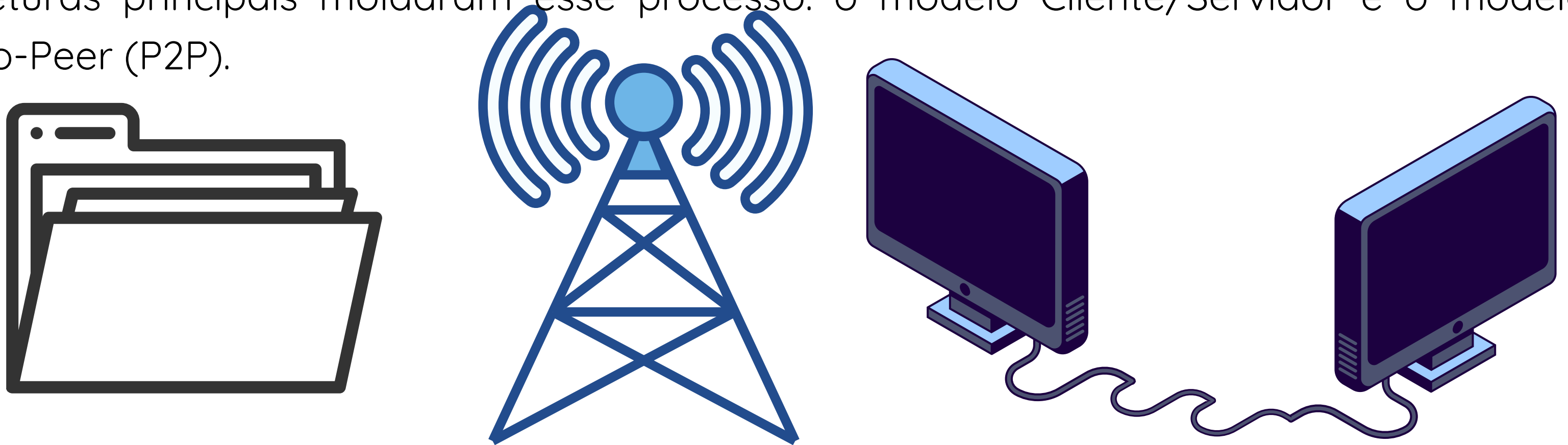
COMPARTILHAMENTO DE ARQUIVOS: P2P VS. CLIENTE/SERVIDOR



Guilherme Epifânio, Maria Eduarda, Victor Gabriel

A ESSÊNCIA DO COMPARTILHAMENTO DIGITAL

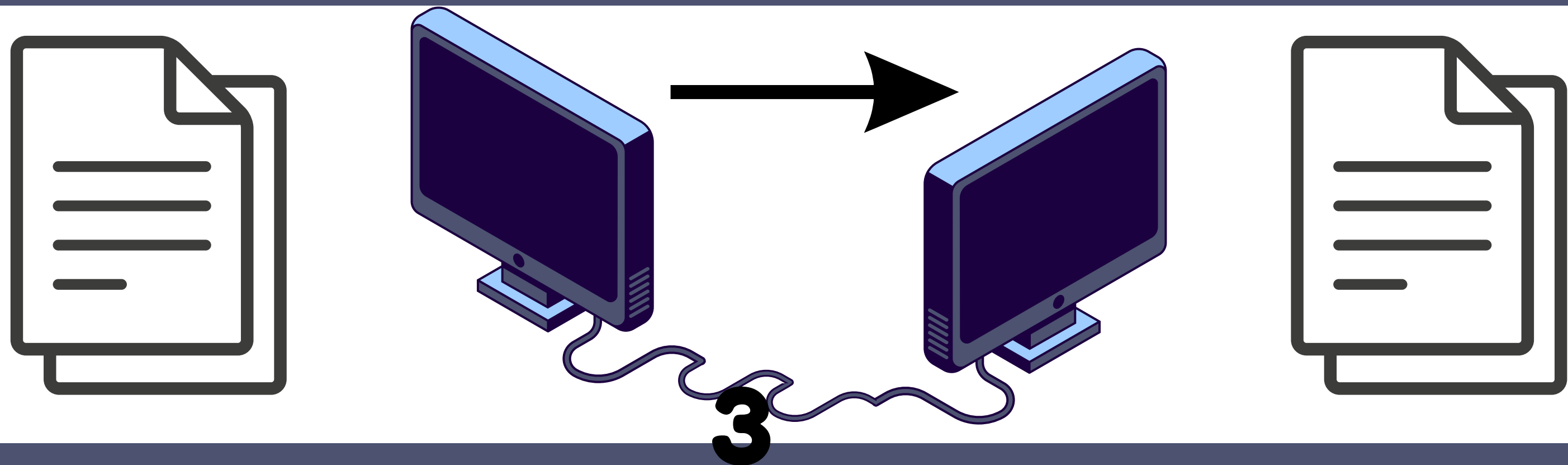
O compartilhamento de arquivos é a espinha dorsal da internet moderna, fundamental para a distribuição de músicas, vídeos, documentos e softwares. Desde os anos 90, duas arquiteturas principais moldaram esse processo: o modelo Cliente/Servidor e o modelo Peer-to-Peer (P2P).



O modelo Cliente/Servidor, com sua centralização, dominou a era inicial da web em serviços como FTP, HTTP, Google Drive e Dropbox, oferecendo controle administrativo e políticas de segurança unificadas.

A ESSÊNCIA DO COMPARTILHAMENTO DIGITAL

Já o P2P, popularizado por sistemas como Napster e BitTorrent no final dos anos 90, descentralizou o poder, transformando usuários em provedores e consumidores simultaneamente, o que favoreceu a escalabilidade e a tolerância a falhas. Essa distinção tem implicações diretas em desempenho, segurança, escalabilidade e confiabilidade.



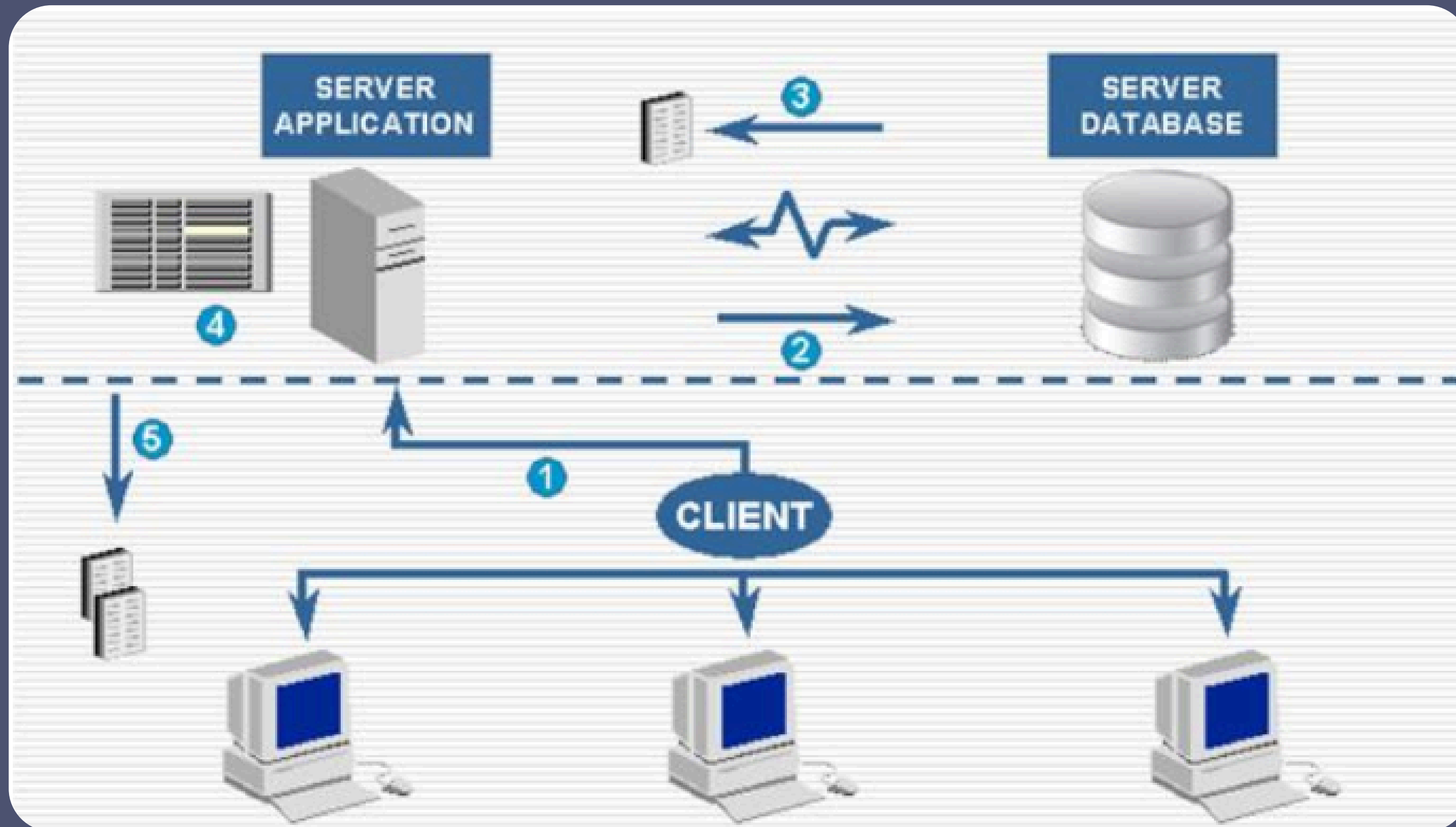
ARQUITETURA CLIENTE/SERVIDOR: CENTRALIZAÇÃO E CONTROLE

A arquitetura Cliente/Servidor é um modelo de computação distribuída onde as responsabilidades são claramente divididas. O **servidor** atua como provedor de serviços, centralizando o armazenamento e o controle absoluto dos dados, enquanto os **clientes** solicitam esses recursos.

- **Funcionamento:** Clientes estabelecem conexão, enviam requisições (download, upload), o servidor processa e retorna a resposta.
- **Exemplos Históricos e Atuais:** HTTP, FTP, Google Drive, Dropbox.
- **Vantagens:** Controle administrativo, segurança centralizada, facilidade de backup e manutenção, consistência de dados.
- **Limitações:** Ponto único de falha, custos de infraestrutura elevados para alta demanda, gargalos de desempenho em picos, escalabilidade limitada pelo hardware.



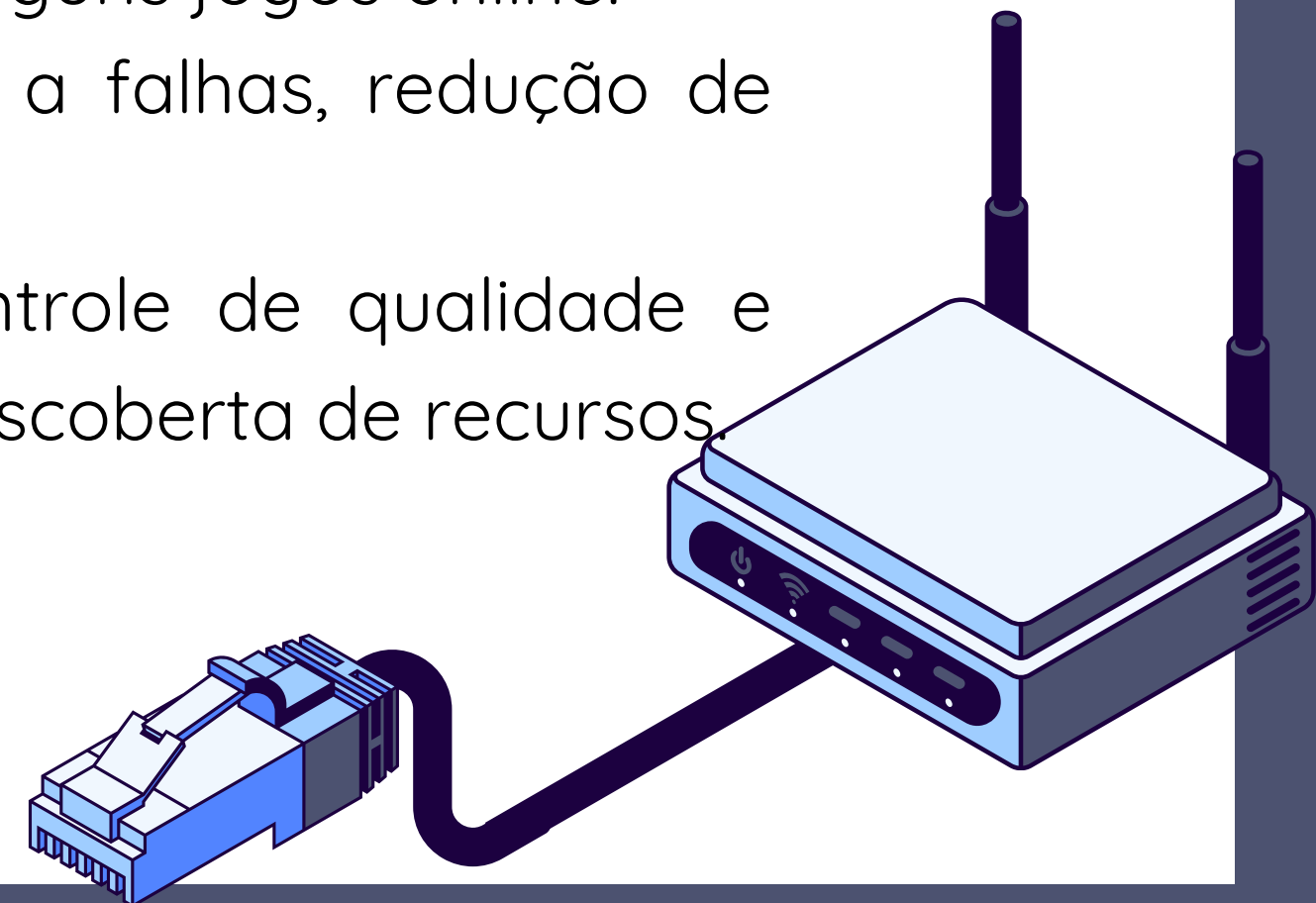
ARQUITETURA CLIENTE/SERVIDOR: CENTRALIZAÇÃO E CONTROLE



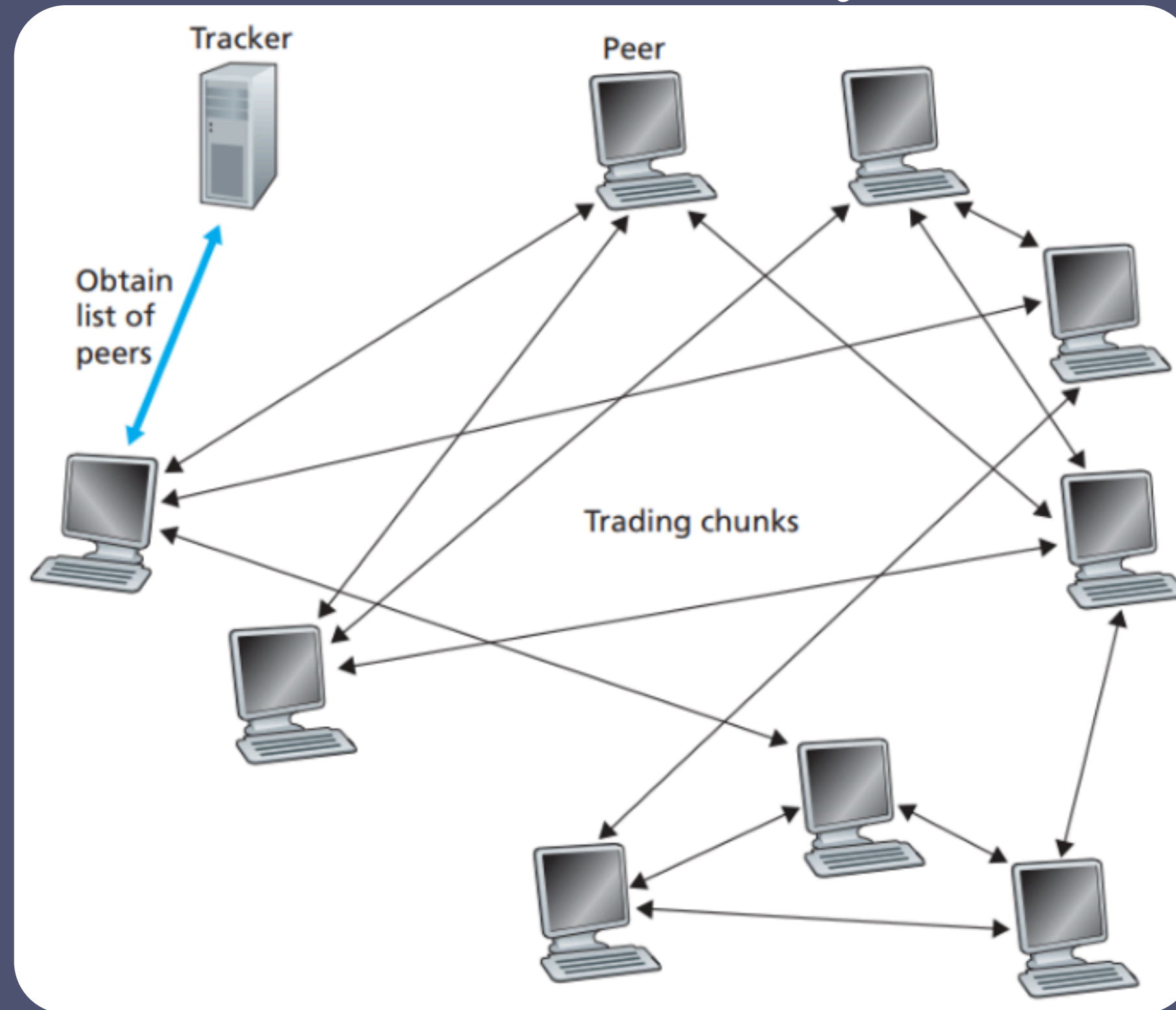
ARQUITETURA P2P: DESCENTRALIZAÇÃO O E COLABORAÇÃO

O modelo Peer-to-Peer (P2P) é uma abordagem descentralizada onde cada participante (**peer** ou **nó**) atua simultaneamente como cliente e servidor, eliminando a necessidade de uma autoridade central.

- **Funcionamento:** Peers colaboram diretamente para compartilhar, distribuir e manter o conteúdo, segmentando arquivos para download simultâneo de múltiplas fontes.
- **Exemplos Históricos e Atuais:** BitTorrent, Gnutella, Blockchain, alguns jogos online.
- **Vantagens:** Escalabilidade ilimitada, redundância e tolerância a falhas, redução de custos de infraestrutura, resistência à censura.
- **Limitações:** Maior complexidade técnica, dificuldades no controle de qualidade e segurança do conteúdo, inconsistência de dados, desafios na descoberta de recursos.



ARQUITETURA P2P: DESCENTRALIZAÇÃO E COLABORAÇÃO



CLIENTE/SERVIDOR VS. P2P: UMA ANÁLISE DETALHADA

A escolha da arquitetura ideal depende das prioridades do projeto. Abaixo, uma comparação sistemática das características fundamentais.

Característica	Cliente/Servidor	P2P
Centralização	Servidor central controla todos os recursos.	Rede descentralizada sem ponto central.
Escalabilidade	Limitada pela capacidade do servidor.	Alta - cresce com o número de peers.
Tolerância a falhas	Baixa - falha do servidor afeta todo sistema.	Alta - múltiplos pontos de redundância.
Custo de infraestrutura	Alto - servidores potentes necessários.	Baixo - usa recursos dos próprios usuários.
Controle de qualidade	Total controle sobre conteúdo e acesso.	Limitado - depende da colaboração.
Segurança	Centralizada e mais controlável.	Complexa - múltiplos pontos de entrada.
Desempenho	Consistente, mas pode ter gargalos.	Variável - depende dos peers ativos.

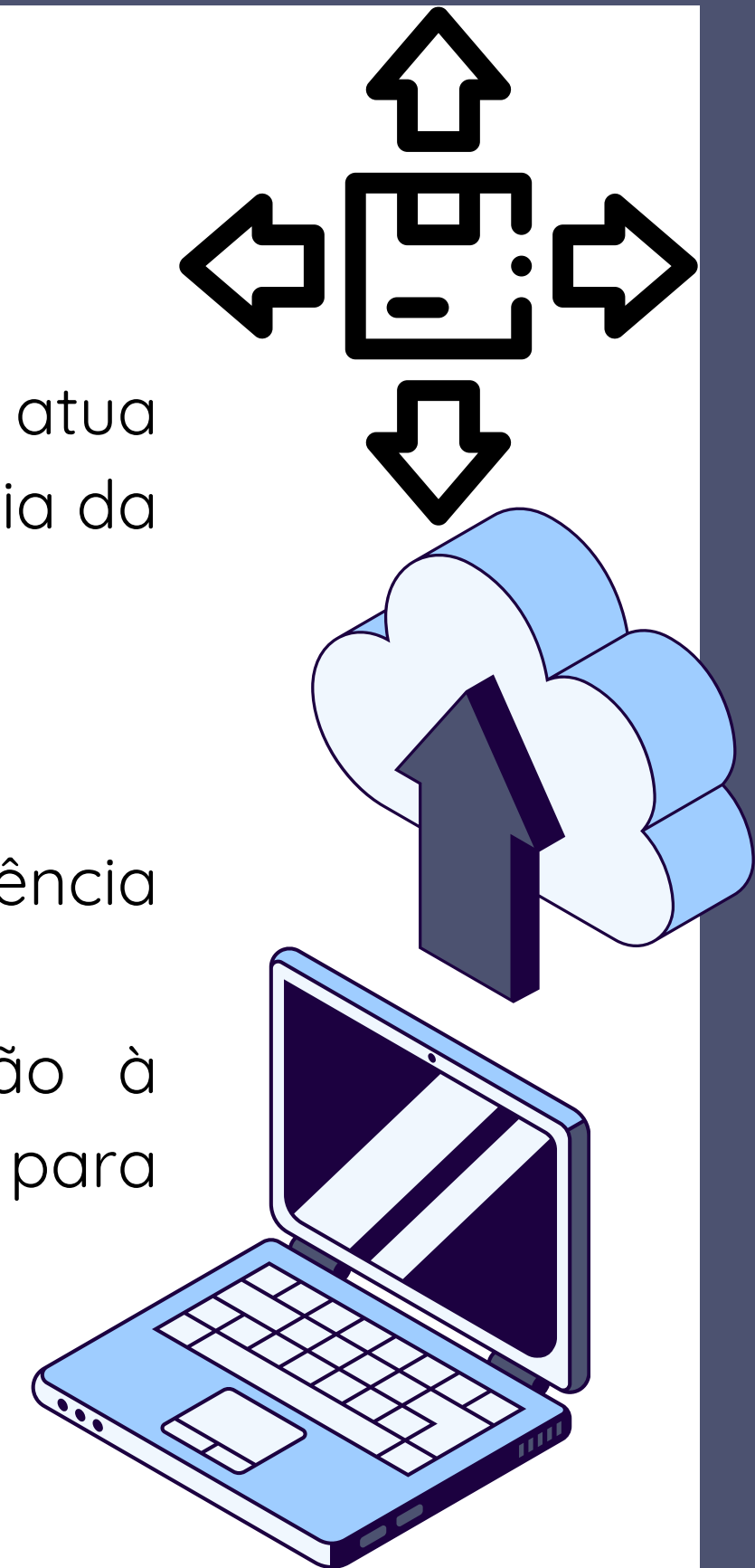
IMPLEMENTAÇÕES REAIS: BITTORRENT

BitTorrent - P2P Puro

Protocolo P2P fundamental para distribuição de arquivos grandes. Cada peer atua como cliente e servidor, distribuindo a carga de trabalho e otimizando a eficiência da transferência.

Vantagens e Desvantagens

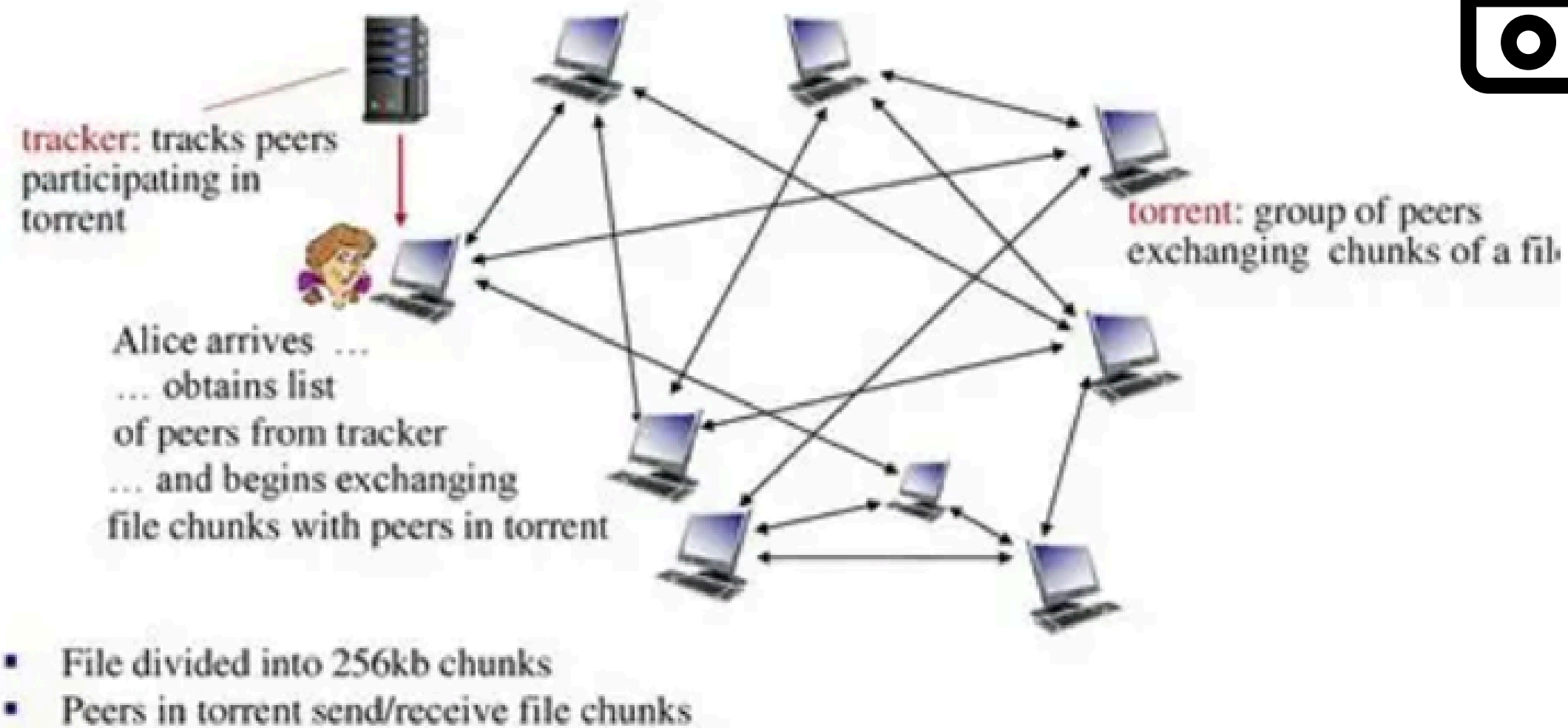
- **Vantagens:** Escalabilidade superior (mais usuários = mais capacidade), eficiência de banda, alta tolerância a falhas.
- **Desvantagens:** Dependência da proporção seeders/leechers, associação à violação de direitos autorais, complexidade de configuração de rede para usuários iniciantes.



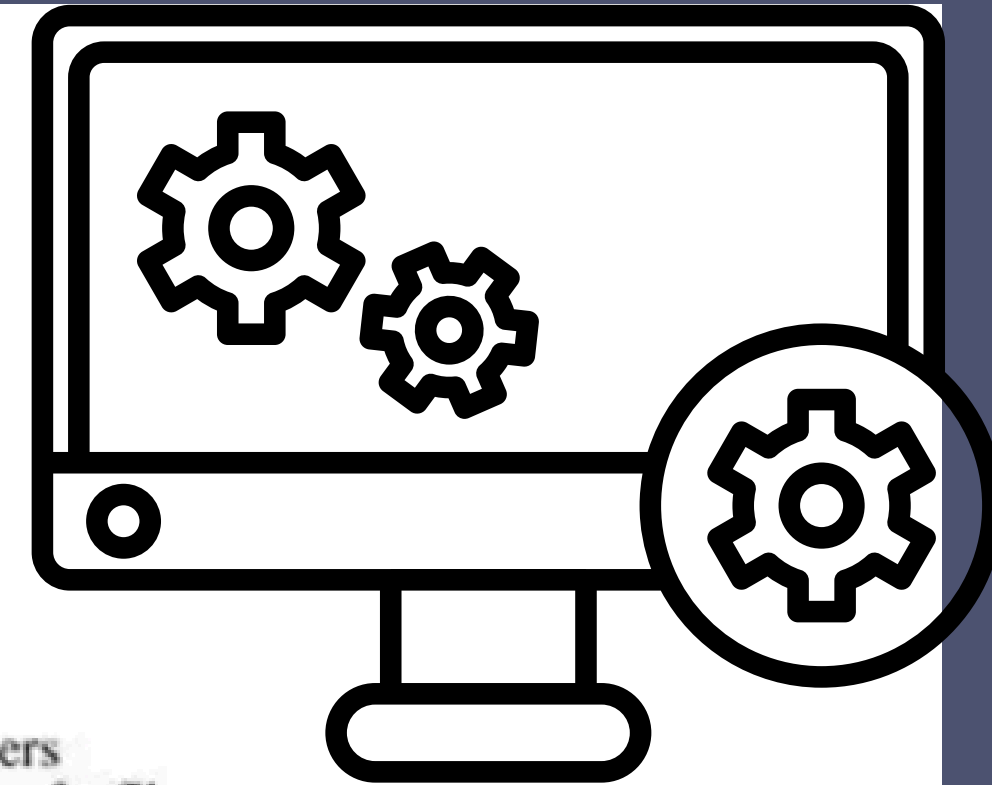
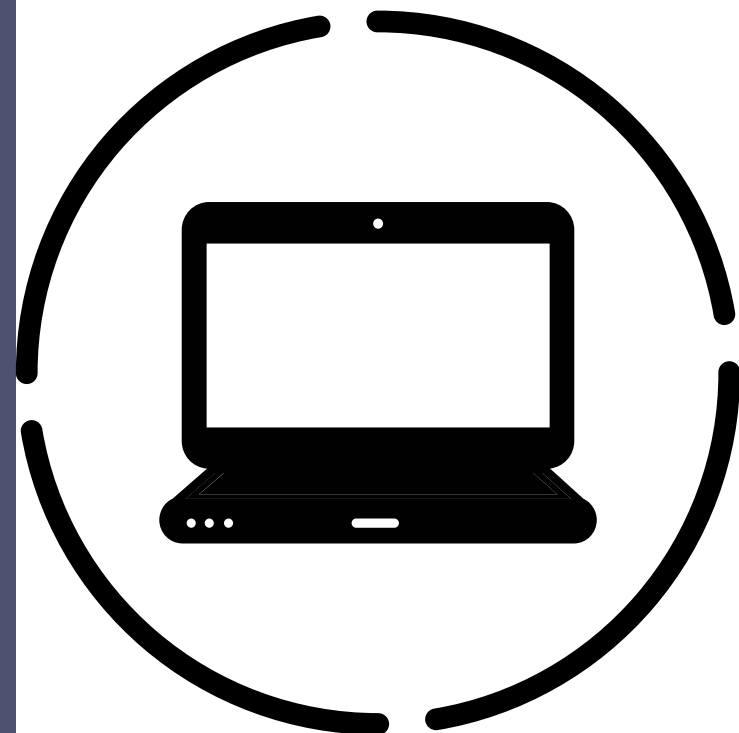
IMPLEMENTAÇÕES REAIS: BITTORRENT

BitTorrent File Distribution

P2P File Distribution: BitTorrent



11



IMPLEMENTAÇÕES REAIS: GOOGLE DRIVE

Google Drive - Cliente/Servidor Moderno

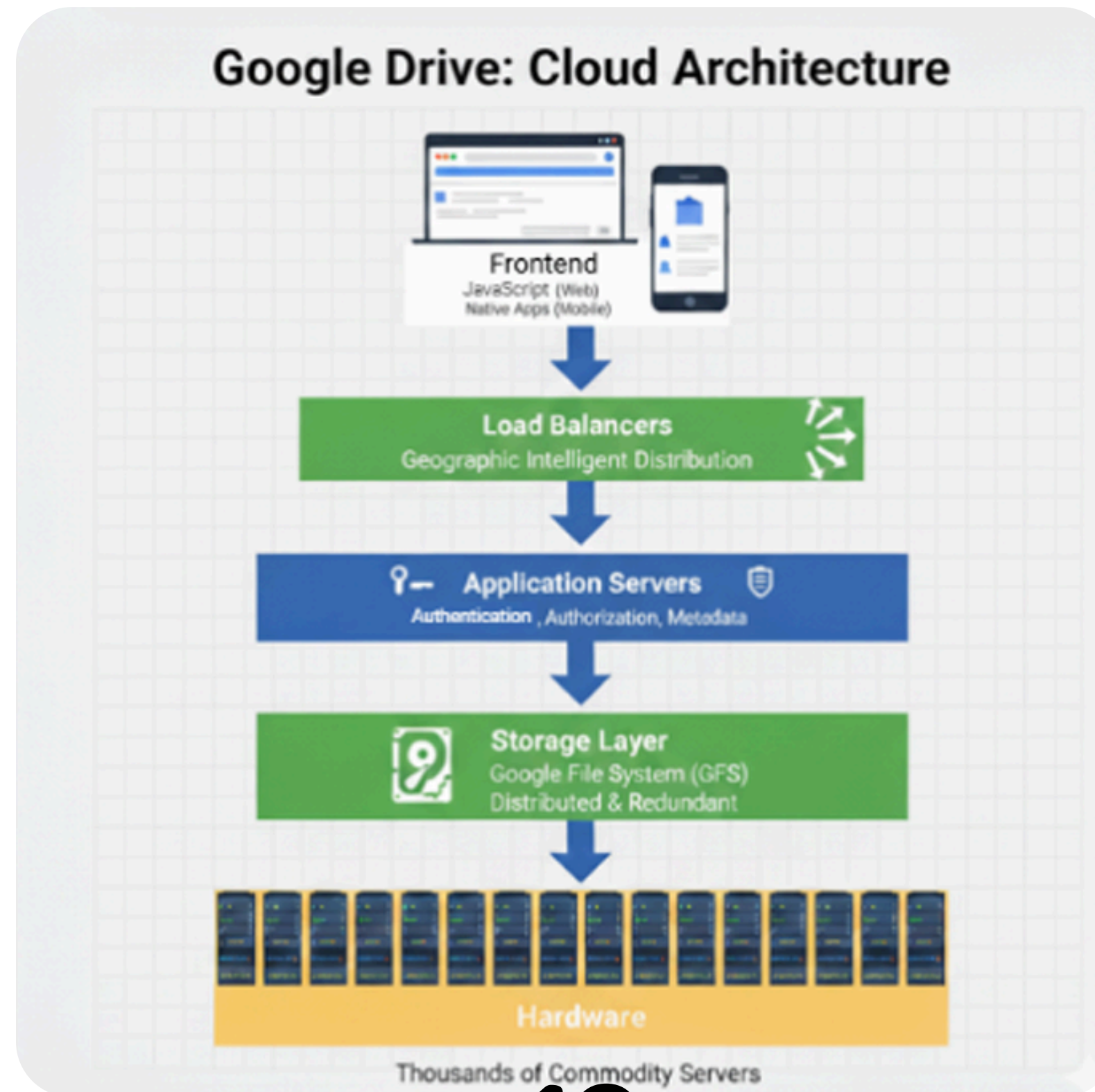
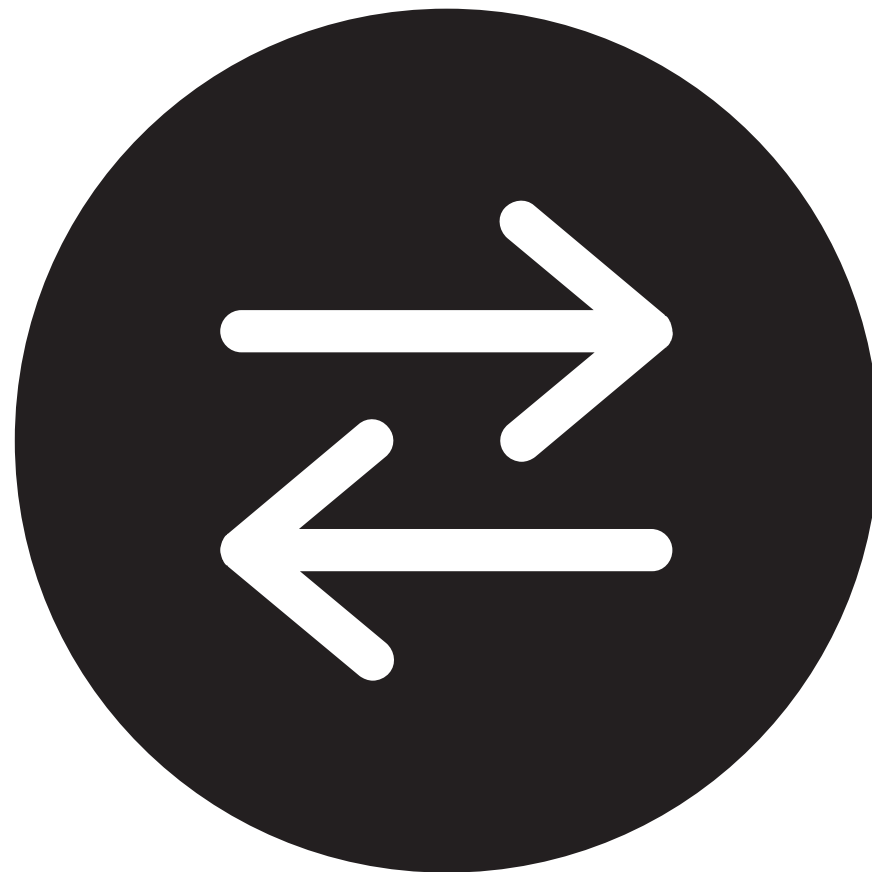
Serviço de armazenamento em nuvem baseado em uma arquitetura Cliente/Servidor moderna, com data centers globalmente distribuídos e o Google File System (GFS).

Vantagens e Desvantagens

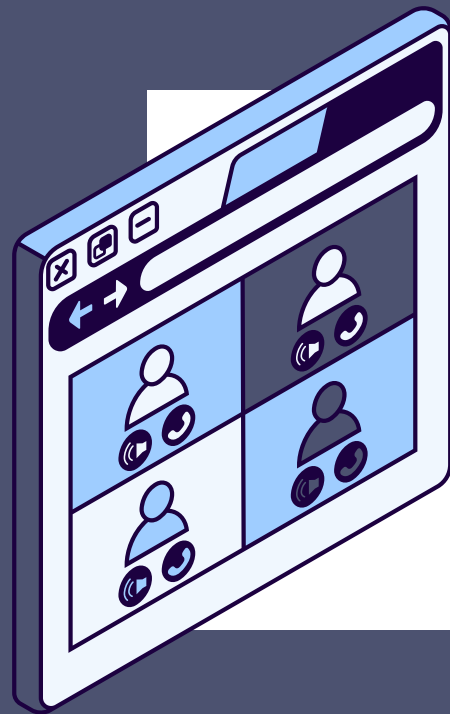
- **Vantagens:** Disponibilidade global, alta confiabilidade com replicação de dados, colaboração em tempo real, escalabilidade transparente.
- **Desvantagens:** Dependência de conectividade com a internet, questões de privacidade e soberania dos dados, limites de armazenamento na versão gratuita.



IMPLEMENTAÇÕES REAIS: GOOGLE DRIVE



ARQUITETURAS HÍBRIDAS EM JOGOS ONLINE



Cliente/Servidor Autoritativo (CS2)

Servidor único mantém o estado do jogo, prevenindo trapaçças. Clientes são "terminais burros", garantindo precisão e controle. Técnicas como previsão do cliente e compensação de lag mitigam latência.



P2P Determinístico Lockstep (AoE II)

Cada cliente calcula o estado do jogo, exigindo determinismo absoluto. Baixo uso de banda, mas latência de entrada é proporcional ao ping mais lento. Vulnerável a dessincronização.



Arquitetura Híbrida (Minecraft)

Roda sempre em modo cliente/servidor (mesmo single-player), com cliente e servidor lógicos. Oferece flexibilidade e escalabilidade, usando código único para todas as versões (single-player, LAN, online).

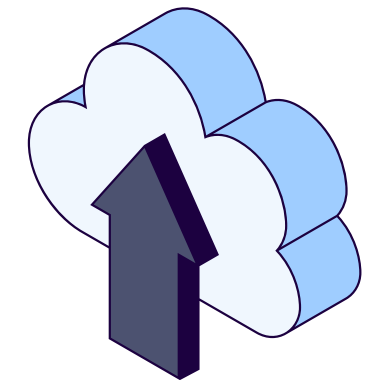
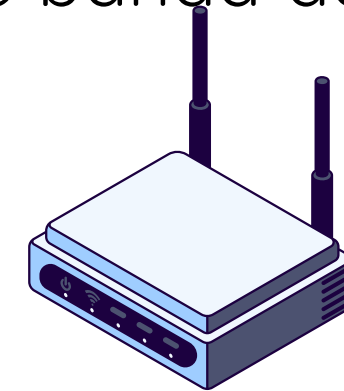


DESENVOLVIMENTO DE UMA REDE P2P SIMPLIFICADA

Nosso projeto consiste em uma implementação de rede P2P para compartilhamento de arquivos, inspirada no BitTorrent. Esta aplicação permite que múltiplos nós atuem como clientes e servidores, otimizando a distribuição de dados e a largura de banda da rede através de uma arquitetura híbrida e algoritmos inteligentes.

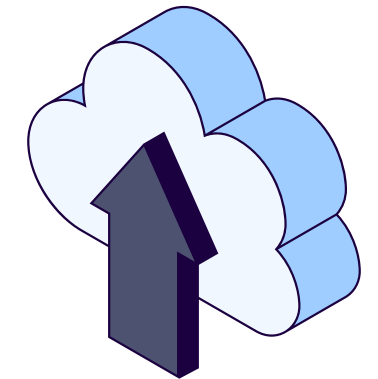
Componentes Chave

- Arquivo .iso: "Mapa" do conteúdo a ser baixado, com dados que simulam uma imagem do Ubuntu.
- Tracker (Servidor Rastreador): Componente centralizado para coordenação, registra pares ativos e pedaços de arquivo, gerencia uma estrutura de dados compartilhada de forma segura.
- Seeder: Nó que possui 100% do arquivo e apenas faz upload.
- Leecher: Nó que está baixando, mas também compartilha os blocos já recebidos.

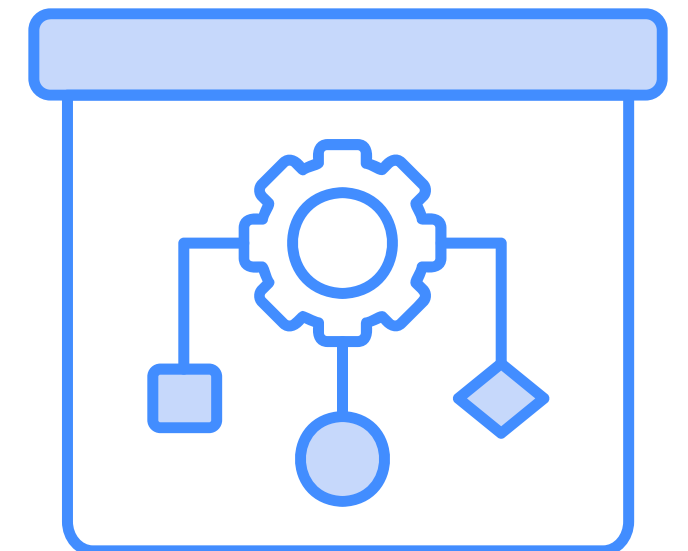
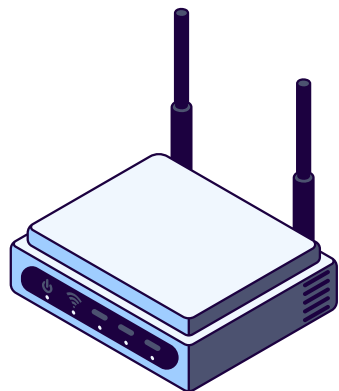


DESENVOLVIMENTO DE UMA REDE P2P SIMPLIFICADA

Algoritmos Inteligentes



- Piece Selection ("Rarest First"): Prioriza o download dos pedaços menos disponíveis na rede, aumentando a redundância e resiliência.
- Peer Selection ("Altruísta"): Atende a todos os pares igualmente, enviando pedaços solicitados sem exigir colaboração em troca. A cooperação é incondicional.
- Cálculo de Downloads Ótimos: Adapta o número de downloads simultâneos com base na "saúde da rede" (número de seeds e peers disponíveis).




CONCLUSÃO


A escolha entre P2P e Cliente/Servidor é contextual. Para aplicações corporativas, o modelo Cliente/Servidor oferece controle rigoroso, segurança e consistência de dados. Já o P2P brilha na distribuição de conteúdo em larga escala, com alta escalabilidade e redução de custos.


- **Trade-offs de Segurança:** Cliente/Servidor centraliza vulnerabilidades; P2P distribui riscos, mas com desafios na uniformidade dos controles.
- **Desempenho:** Cliente/Servidor é previsível, mas pode ter gargalos; P2P varia, mas melhora com mais participantes.
- **Tendências Futuras:** Convergência para arquiteturas híbridas que combinam o melhor de ambos os mundos. Tecnologias como edge computing, blockchain e Web3 continuarão a expandir as possibilidades.


A evolução futura é clara: soluções que aproveitem a força de cada arquitetura, adaptando-se dinamicamente às necessidades específicas das aplicações, serão a chave para sistemas de compartilhamento de arquivos mais eficientes, resilientes e escaláveis.

REFERÊNCIAS

 KUROSE, James F.; ROSS, Keith W. Redes de Computadores e a Internet: Uma Abordagem Top-Down. 6ª ed. São Paulo: Pearson Education do Brasil, 2013.

 COHEN, Bram. Incentives build robustness in BitTorrent. In: Workshop on Economics of Peer-to-Peer systems, Berkeley, 2003.

 TANENBAUM, A. S.; WETHERALL, D. J. Redes de Computadores. 6ª ed. São Paulo: Pearson, 2021.

 ARMBRUST, Michael et al. A view of cloud computing. Communications of the ACM, v. 53, n. 4, p. 50-58, 2010.

!OBRIGADO!



Guilherme Epifânio, Maria Eduarda, Victor Gabriel