

30-769

# Sistemas Distribuídos

MSc. Fernando Schubert



# Conteúdo da Disciplina

## **Introdução**

- Definição de um sistema distribuído
- Tipos de sistemas distribuídos
- Estilos arquitetônicos
- Arquiteturas versus middleware

## **Comunicação**

- Fundamentos
- Comunicação orientada a mensagem
- Comunicação orientada a fluxo
- Modelo Cliente-Servidor

## **Nomeação**

## **Sincronização**

- Sincronização de relógios
- Algoritmos de eleição

## **Consistência e replicação**

- Modelos de consistência centrados em dados e no cliente
- Protocolos de consistência
- Aspectos de tolerância a falha em SDs
- Comunicação confiável

## **Sistemas distribuídos baseados em objetos, na Web e em coordenação**

- Arquitetura
- Processos
- Comunicação
- Sincronização

# Introdução aos Sistemas Distribuídos - Conceito

- O que é um sistema distribuído?
  - É uma coleção de computadores independentes que aparenta ser um único computador aos seus usuários.
- Um **sistema distribuído** é aquele no qual os componentes localizados em computadores interligados em **rede** se comunicam e coordenam suas ações apenas passando mensagens. Essa definição leva às seguintes características especialmente importantes dos sistemas distribuídos: **concorrência de componentes**, **falta de um relógio global** e **falhas de componentes independentes**. (Colourius)

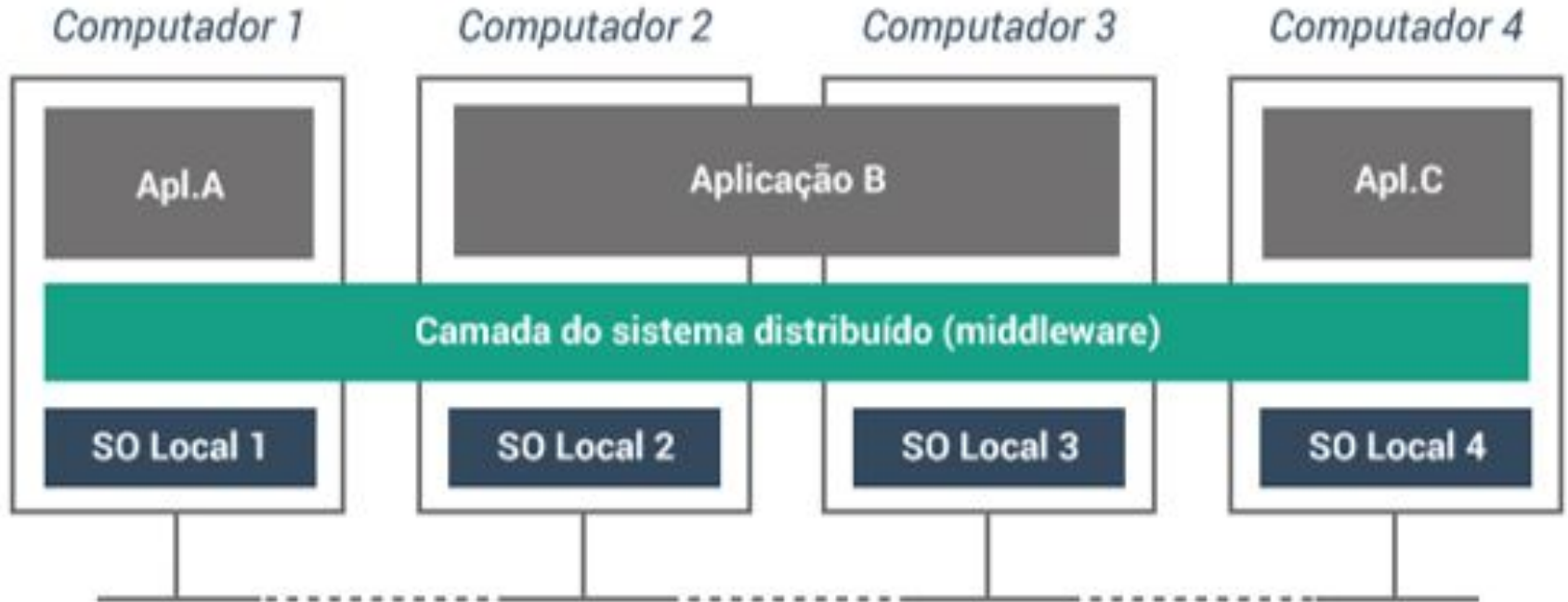
# Introdução aos Sistemas Distribuídos - Conceito

- **Concorrência:** em uma rede de computadores, a execução concorrente de programas é a norma. A capacidade do sistema de manipular recursos compartilhados pode ser ampliada pela adição de mais recursos (por exemplo, computadores) na rede.
- **Inexistência de relógio global:** quando os programas precisam cooperar, eles coordenam suas ações trocando mensagens. A coordenação frequentemente depende de uma noção compartilhada do tempo. Não existe uma noção global única do tempo correto. Essa é uma consequência direta do fato de que a única comunicação se dá por meio do envio de mensagens em uma rede

# Introdução aos Sistemas Distribuídos - Conceito

- **Falhas independentes:** todos os sistemas de computador podem falhar. Nos sistemas distribuídos, as falhas são diferentes. Falhas na rede resultam no isolamento dos computadores que estão conectados a ela, mas isso não significa que eles param de funcionar. Na verdade, os programas neles existentes talvez não consigam detectar se a rede falhou ou se ficou demasiadamente lenta. Cada componente do sistema pode falhar independentemente, deixando os outros ainda em funcionamento.

# Introdução aos Sistemas Distribuídos - Conceito



# Introdução aos Sistemas Distribuídos - Evolução

- **Origens dos Sistemas Distribuídos:**
  - Surgiram da necessidade de compartilhar recursos computacionais e aumentar a confiabilidade e escalabilidade dos sistemas.
  - As primeiras ideias surgiram na década de 1960, com sistemas como o ARPANET e o Projeto MAC.
- **Desenvolvimento Inicial:**
  - Na década de 1970, a pesquisa em sistemas distribuídos começou a se concentrar na comunicação entre computadores e na compartilhamento de recursos.
  - Surgiram tecnologias como o Unix e o TCP/IP, que possibilitaram a comunicação entre sistemas distribuídos.

# Introdução aos Sistemas Distribuídos - Evolução

- **Expansão nos Anos 80 e 90:**
  - Nos anos 80, houve um aumento significativo na pesquisa e no desenvolvimento de sistemas distribuídos.
  - Surgiram arquiteturas como a cliente-servidor e sistemas de banco de dados distribuídos.
  - As redes locais (LANs) e as redes de área ampla (WANs) se tornaram mais comuns, impulsionando ainda mais o desenvolvimento de sistemas distribuídos.
- **Internet e World Wide Web:**
  - O surgimento da Internet na década de 1990 trouxe uma nova dimensão aos sistemas distribuídos.
  - A World Wide Web (WWW) permitiu o acesso e compartilhamento de informações em uma escala global, introduzindo conceitos como servidores web e navegadores.

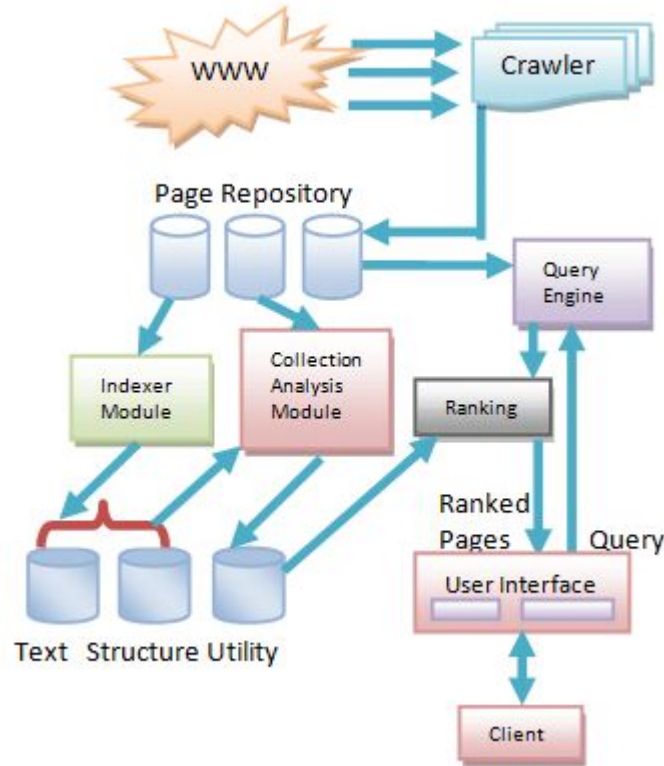


# Introdução aos Sistemas Distribuídos - Evolução

- **Computação em Nuvem:**
  - Nas últimas décadas, a computação em nuvem emergiu como uma evolução dos sistemas distribuídos.
  - Plataformas como Amazon Web Services (AWS), Microsoft Azure e Google Cloud Platform oferecem recursos computacionais distribuídos como serviços.
- **Tecnologias e Protocolos:**
  - Diversos protocolos e tecnologias foram desenvolvidos para facilitar a comunicação e coordenação em sistemas distribuídos, como RPC (Remote Procedure Call), SOAP (Simple Object Access Protocol) e REST (Representational State Transfer).
- **Desafios e Avanços Futuros:**
  - A escalabilidade, a consistência e a tolerância a falhas continuam sendo desafios fundamentais em sistemas distribuídos.
  - Tecnologias emergentes, como blockchain e computação em névoa (fog computing), estão moldando o futuro dos sistemas distribuídos.

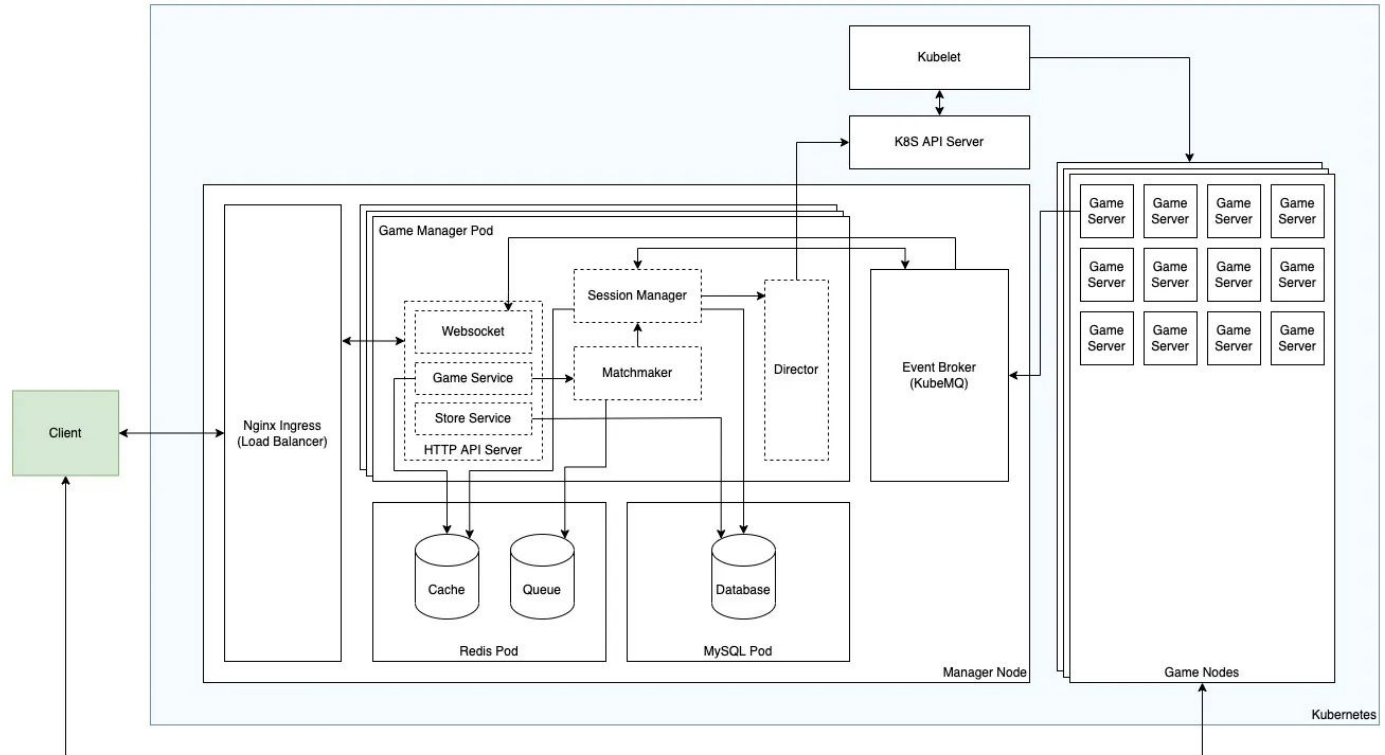
# Introdução aos Sistemas Distribuídos - Exemplos de SDs

- Pesquisa na Web



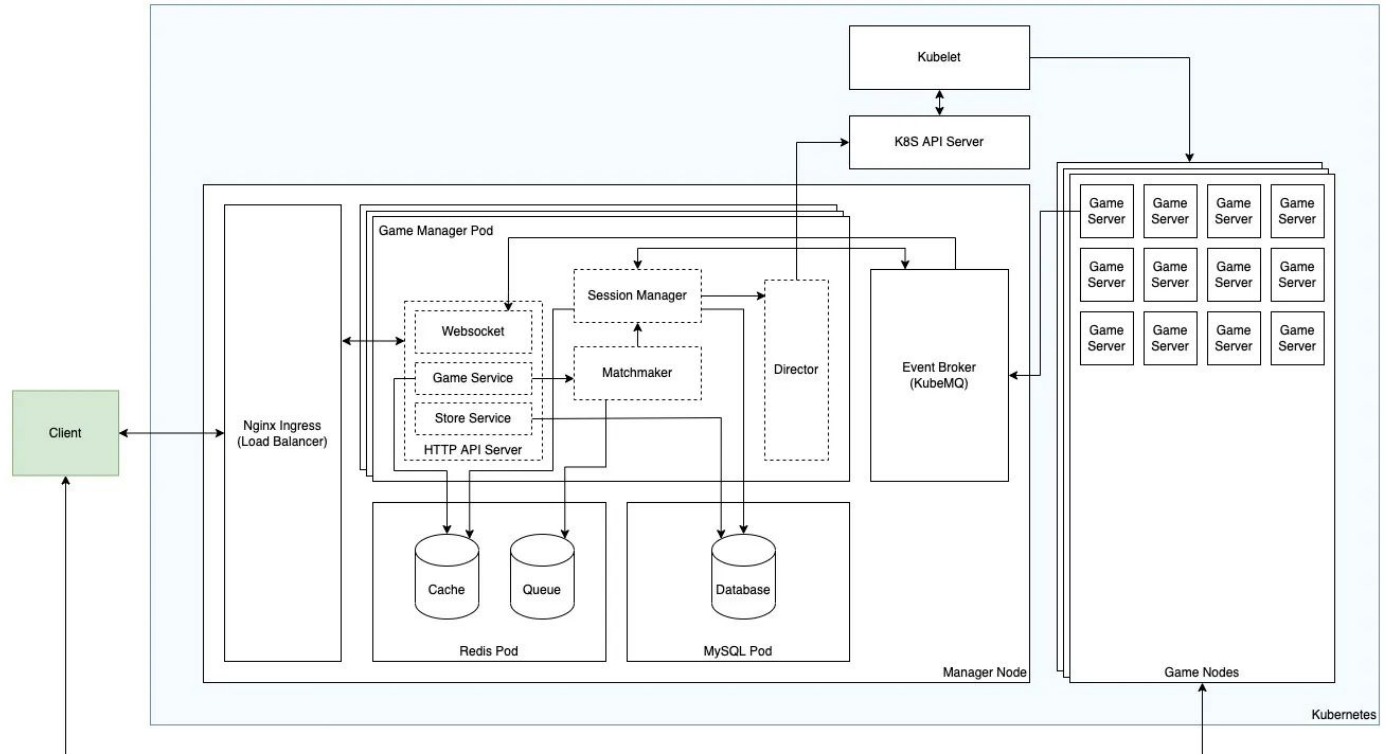
# Introdução aos Sistemas Distribuídos - Exemplos de SDs

- Gaming



# Introdução aos Sistemas Distribuídos - Exemplos de SDs

- Gaming



# Introdução aos Sistemas Distribuídos - Desafios

## Heterogeneidade

A heterogeneidade (isto é, variedade e diferença) se aplica aos seguintes aspectos:

- **redes;**
- **hardware de computador;**
- **sistemas operacionais;**
- **linguagens de programação;**
- **implementações de diferentes desenvolvedores.**

# Introdução aos Sistemas Distribuídos - Desafios

## Heterogeneidade

Soluções:

- Uso de protocolos padronizados
- Criação de uma camada de software que fornece uma abstração de programação e mascaramento da heterogeneidade das redes, do hardware, dos sistemas operacionais e das linguagens de programação. Essa camada é comumente chamada de **middleware**

# Introdução aos Sistemas Distribuídos - Desafios

## Sistemas Abertos

- Diz-se que um sistema computacional é aberto quando ele pode ser estendido e reimplementado de várias maneiras.
- Os sistemas abertos são caracterizados pelo fato de suas principais interfaces serem publicadas.
- Os sistemas distribuídos abertos são baseados na estipulação de um mecanismo de comunicação uniforme e em interfaces publicadas para acesso aos recursos compartilhados.
- Os sistemas distribuídos abertos podem ser construídos a partir de hardware e software heterogêneo, possivelmente de diferentes fornecedores. Para que um sistema funcione corretamente, a compatibilidade de cada componente com o padrão publicado deve ser cuidadosamente testada e verificada.

# Introdução aos Sistemas Distribuídos - Desafios

## Segurança

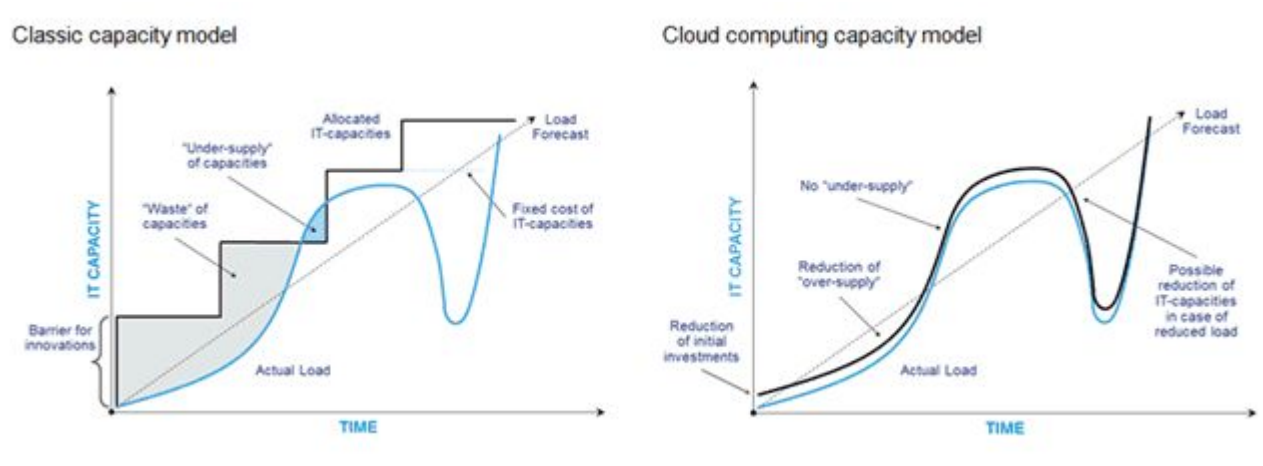
- A segurança de recursos de informação tem três componentes: confidencialidade (proteção contra exposição para pessoas não autorizadas), integridade (proteção contra alteração ou dano) e disponibilidade (proteção contra interferência com os meios de acesso aos recursos).



# Introdução aos Sistemas Distribuídos - Desafios

## Escalabilidade

- Um sistema é descrito como escalável se permanece eficiente quando há um aumento significativo no número de recursos e no número de usuários.



# Introdução aos Sistemas Distribuídos - Desafios

## Tratamento de Falhas

- As falhas em um sistema distribuído são parciais – isto é, alguns componentes falham, enquanto outros continuam funcionando. Portanto, o tratamento de falhas é particularmente difícil.

# Introdução aos Sistemas Distribuídos - Desafios

## Tratamento de Falhas

- As falhas em um sistema distribuído são parciais – isto é, alguns componentes falham, enquanto outros continuam funcionando. Portanto, o tratamento de falhas é particularmente difícil.
- **Tolerância a falhas:** a maioria dos serviços na Internet apresenta falhas – não seria prático para eles tentar detectar e mascarar tudo que possa ocorrer em uma rede grande assim, com tantos componentes. Seus clientes podem ser projetados de forma a tolerar falhas, o que geralmente envolve a tolerância também por parte dos usuários. Por exemplo, quando um navegador não consegue contatar um servidor Web, ele não faz o usuário esperar indefinidamente, enquanto continua tentando – ele informa o usuário sobre o problema, deixando-o livre para tentar novamente.
- **Recuperação de falhas:** a recuperação envolve projetar software de modo que o estado dos dados permanentes possa ser recuperado ou “retrocedido” após a falha de um servidor. Em geral, as computações realizadas por alguns programas ficarão incompletas quando ocorrer uma falha, e os dados permanentes que eles atualizam (arquivos em disco) podem não estar em um estado consistente.
- **Redundância:** os serviços podem se tornar tolerantes a falhas com o uso de componentes redundantes.

# Introdução aos Sistemas Distribuídos - Desafios

## Concorrência

- Tanto os serviços como os aplicativos fornecem recursos que podem ser compartilhados pelos clientes em um sistema distribuído. Portanto, existe a possibilidade de que vários clientes tentem acessar um recurso compartilhado ao mesmo tempo. Por exemplo, uma estrutura de dados que registra lances de um leilão pode ser acessada com muita frequência, quando o prazo final se aproximar.

# Introdução aos Sistemas Distribuídos - Desafios

## Transparência

- A transparência é definida como a ocultação, para um usuário final ou para um programador de aplicativos, da separação dos componentes em um sistema distribuído, de modo que o sistema seja percebido como um todo, em vez de como uma coleção de componentes independentes. As implicações da transparência têm grande influência sobre o projeto do software de sistema.

# Introdução aos Sistemas Distribuídos - Desafios

<b>Transparência</b>	<b>Descrição</b>
Acesso	Esconde diferenças na representação de dados e como um recurso é acessado
Localização	Esconde onde um recurso está localizado
Migração	Esconde que um recurso pode mover-se para outra localização
Relocação	Esconde que um recurso pode ser movido para outra localização enquanto esta sendo usado
Replicação	Esconde que um recurso pode ser compartilhado por vários usuários concorrentes
Concorrência	Esconde que um recurso pode ser compartilhado por vários usuários concorrentes
Falha	Esconde a falha e recuperação de um recurso
Persistência	Esconde quando um recurso (software) está em memória ou em disco

# Introdução aos Sistemas Distribuídos - Desafios

## Qualidade de Serviço (QoS)

- Uma vez fornecida a funcionalidade exigida de um serviço (como o serviço de arquivo em um sistema distribuído), podemos passar a questionar a qualidade do serviço fornecido. As principais propriedades não funcionais dos sistemas que afetam a qualidade do serviço experimentada pelos clientes e usuários são a confiabilidade, a segurança e o desempenho. A adaptabilidade para satisfazer as configurações de sistema variáveis e a disponibilidade de recursos tem sido reconhecida como um aspecto muito importante da qualidade do serviço.

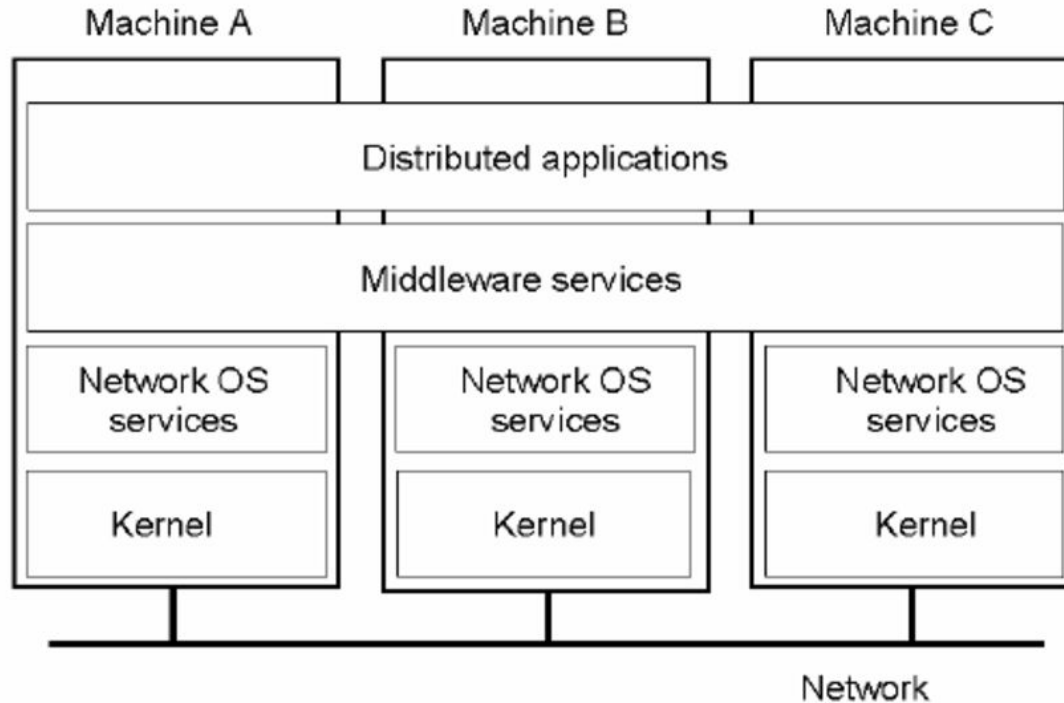
# Comunicação em Sistemas Distribuídos

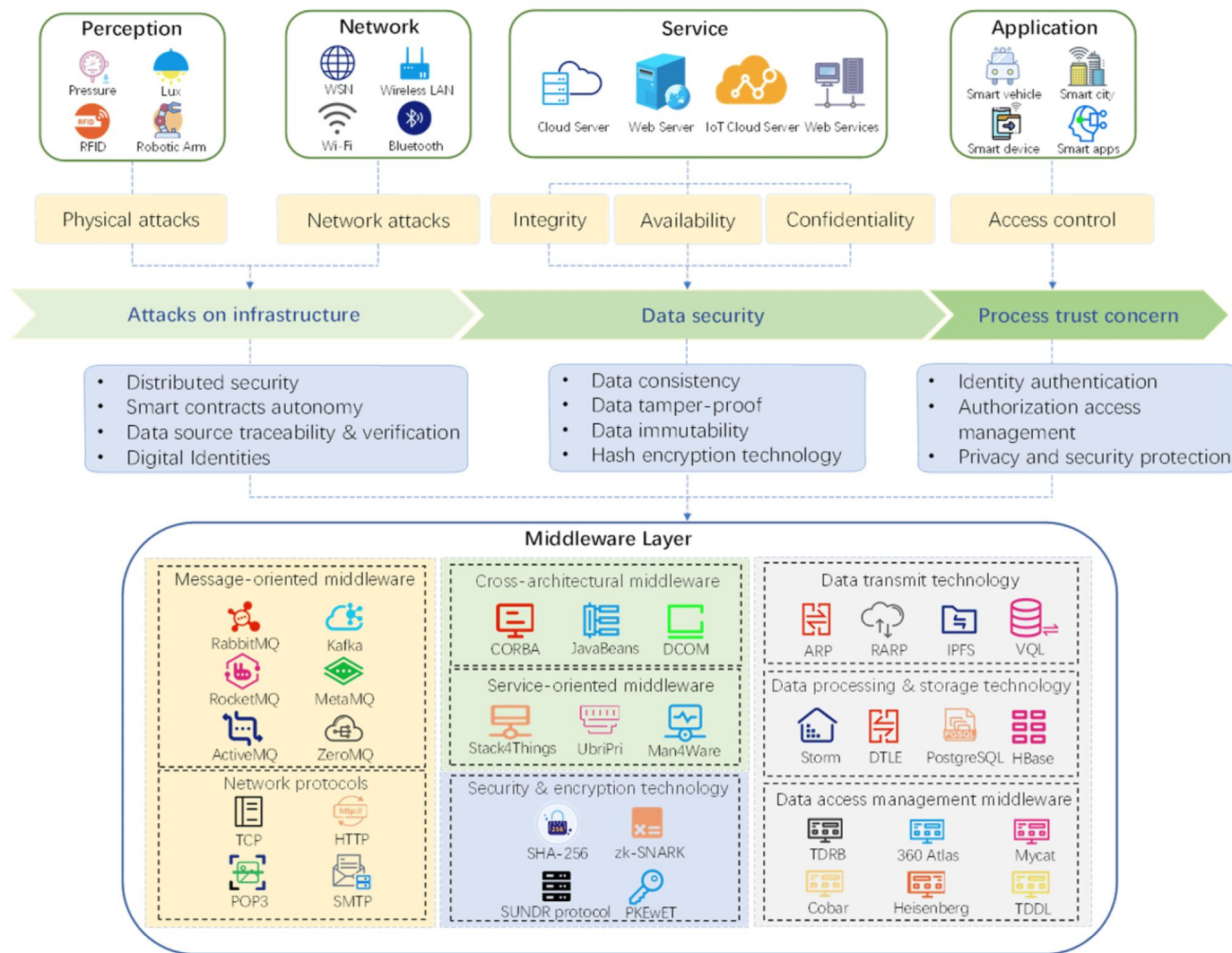
## Middleware

- Softwares conhecidos como middleware ajudam a fornecer a portabilidade, transparência e interoperabilidade em sistemas distribuídos.
- Portabilidade habilita a mudança de um sistema ou componente de um ambiente para outro sem alterar o sistema ou componente.



# Comunicação em Sistemas Distribuídos





# Comunicação em Sistemas Distribuídos

- **RPC (Remote Procedure Call)**
- **RMI (Invocação a método remoto)**
- **CORBA (Common object request broker architecture)**
- **DCOM (Distributed Component Object Model)**
- **Migração de processos em SD**
- **Middleware**
- **REST**
- **SOAP**
- **WebServices**
- **Microservices**
- **...**

# Atividade

Um survey utilizando o template da **SBC** para artigos científicos sobre as diferentes tecnologias de comunicação interprocessos e interhosts em sistemas distribuídos.

<https://www.sbc.org.br/documentos-da-sbc/summary/169-templates-para-artigos-e-capitulos-de-livros/878-modelosparapublicaodeartigos>

- Contendo visão geral de cada método de comunicação
- Comparativo com prós e contras e características principais ( por exemplo tabela )
- Finalizando com tendências e novas tecnologias.