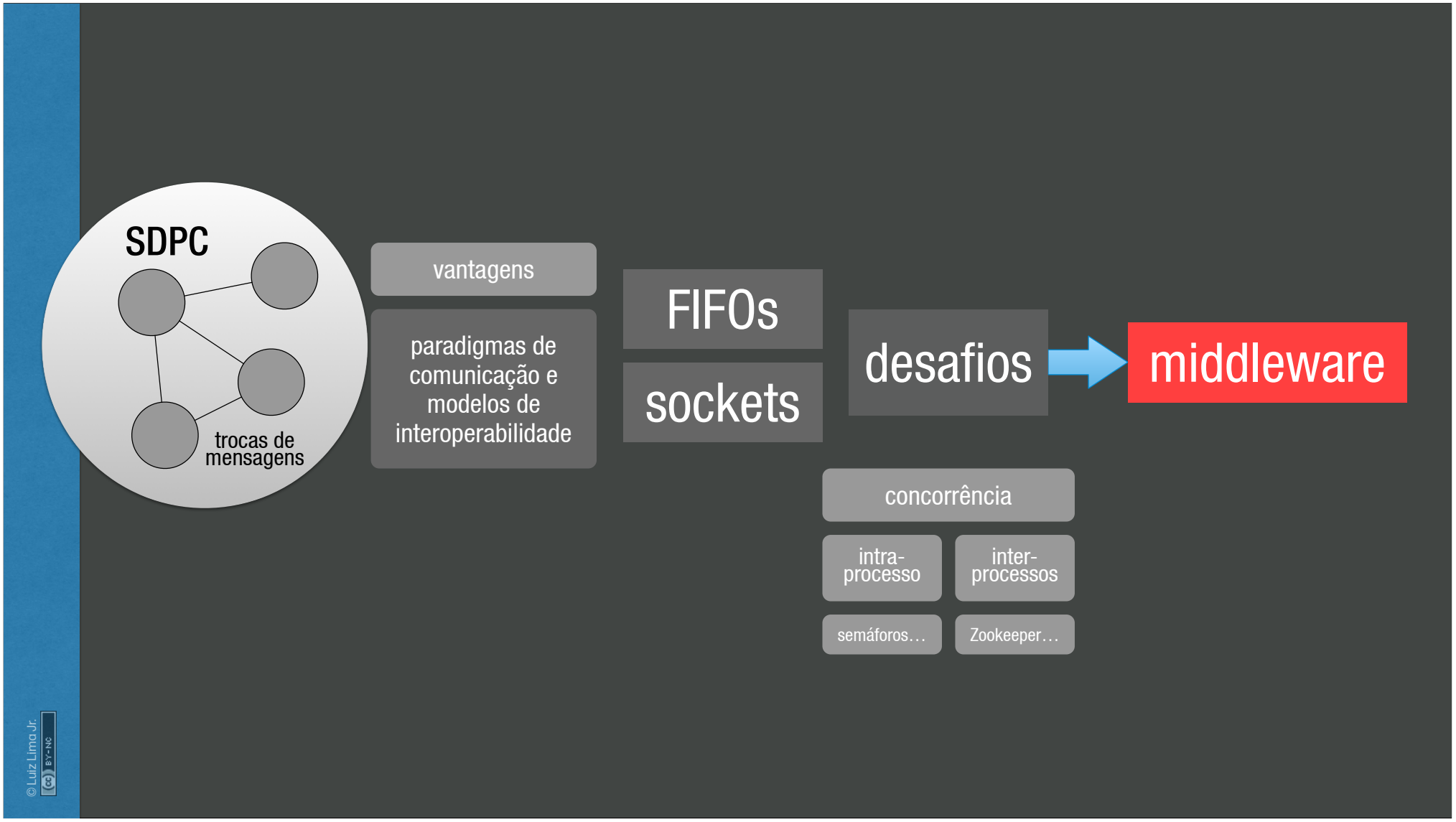




Sistemas Distribuídos e Concorrência
Escola Politécnica – PUCPR
Luiz A. de P. Lima Jr. • luiz.lima@pucpr.br

# Introdução a Middleware


# Middleware

- Problemas com a construção de sistemas distribuídos complexos usando diretamente *sockets*:
  - Necessidade de definição de protocolos
    - ***Tedioso e suscetível a erros…***
  - Serialização de dados
    - ***Sequência de bytes***
    - ***TCP: "stream" = necessidade de delimitação***
  - Conversões de formatos
    - ***Little-endian - big-endian***
    - ***Entre linguagens de alto nível. Exemplo:***
      - **char: C ⟹ byte: Java**

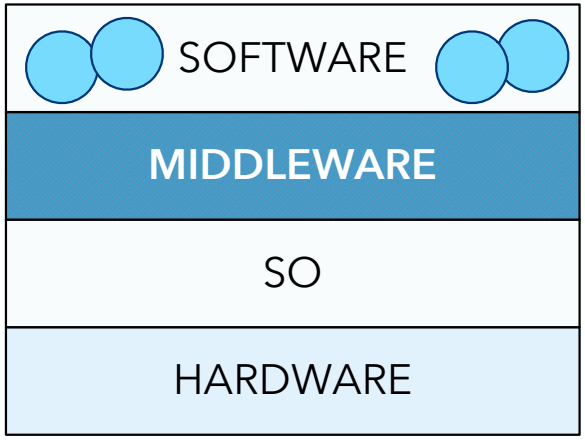

# Middleware

- Solução:
  - Usar uma "**PLATAFORMA**" para facilitar o desenvolvimento de aplicações distribuídas

MIDDLEWARE

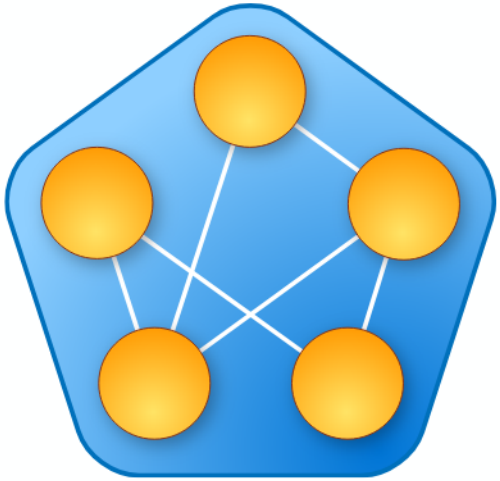

# Middleware

- Transparência de **localização**
- Transparência de **representação de dados**
- **Operações remotas** (ao invés de *bytes*)
- **Migração** de objetos
- Outros **serviços**…



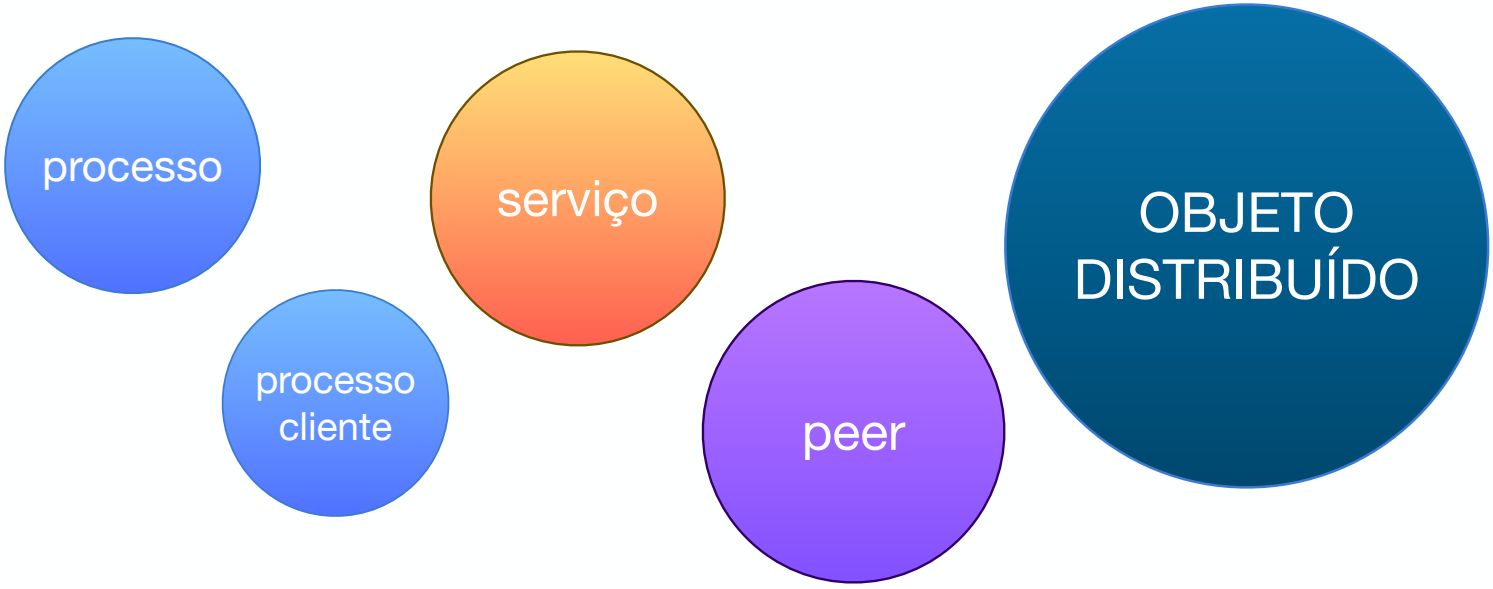

# Middleware

Distribuição
Atualização
Manutenção
Re-uso
Flexibilidade
Disponibilidade
Tolerância a faltas
Etc.




# Middleware

- Quem são as entidades computacionais comunicantes?

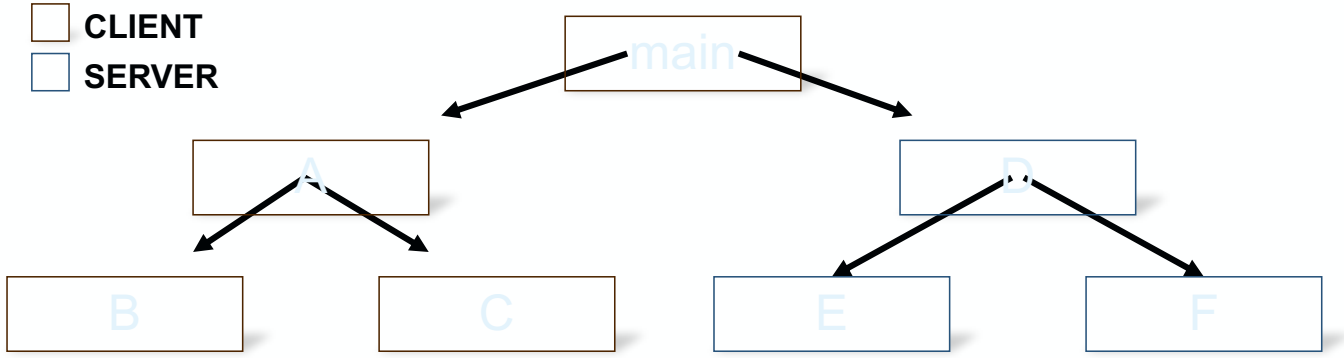



# Remote Procedure Call (RPC)

Middleware

# *Remote Procedure Calls* (RPC)

- Usar o paradigma de **chamada de procedimento** para construir software cliente/servidor
  - (programação “orientada a procedimentos”)




# *Remote Procedure Calls* (RPC)

- **Dificuldades** trazidas ao SO pelas RPCs:
  - **Localização** e **invocação** de serviços (ambiente *runtime*)
  - **Falhas** (*timeout*, *retrial*)
  - **Segurança** (autenticação, etc.)
  - **Ligação** cliente/servidor (diretórios)
  - **Representação de dados** da rede (XDR)
  - Passagem de **parâmetros**