Requisitos de Middleware

Ao ler esta lista de requisitos de middleware é importante considerar dois pontos. Primeiro, a lista não é exaustiva. Logo, dependendo do domínio de aplicação considerado pode haver diversos outros requisitos de middleware não presentes nesta lista. Segundo, ela está em permanente atualização e estruturação.

1. Tipo de comunicação
   * síncrona, assíncrona
2. Ambiente de execução:
   * desktop, celular, sensor, palmtop, qualquer combinação entre estes ambientes.
3. Heterogeneidade suportada
   * sistema operacional, linguagem de programação, rede e hardware.
4. Interface de programação disponibilizada para a construção da aplicação
   * API, biblioteca, framework.
5. Infra-estrutura de rede disponível para comunicação:
   * cabeada, sem-fio (Bluetooth, zigbee, 802.11, outras), heterogênea.
6. Comunicação entre processos ou modos de envio de mensagens
   * Nível de serviço esperado, e.g., guaranteed, once-only, transactional, priority, non-transactional
   * Operações de envoi/recebimento
   * Modo de envoi: assíncrono, síncrono, multicast, broadcast, canal seguro, canal confiável
   * send and forget, store and forward, guaranteed delivery, once-only delivery, are all these variant types of communication services required?
   * What are the requirements on message versioning and the assembly of messages?
   * Does the system need to store message type definition and or metadata in a common repository? If so, is there any constraint on which repository?
   * For the receive or subscribe operations:
   * Is there a need to register or subscribe to a group or type of messages; is waiting synchronously a desired feature? What about timeouts or wait times? Do they need to be specified?

* How critical is the feature of dynamically registering an interest in a topic or message type?
* Suporte a QoS
* Receptor programável para (des)registrar interesse em
* uma mensagem/grupo (síncrono ou assíncrono)
* Tempo de espera programável
* 1-to-many mensagens
* Endereços de mensagens lógicos
* Múltiplas mensagens para um único destino
* Envio e espera de mensagem
* Tratamento de tipos de dados diferentes
* Envio de arquivos como mensagem
* Suporte de mensagens de até 64Mbytes

1. O middlewa proposto irá precisar interoperar com outros sistemas de middleware
   * RMI, CORBA
   * JBOSS
   * JMS-compliant
2. Marshalling
   * Mensagens de tamanho fixo
   * Regras de mapeamento (e.g., CDR)
   * Suporte a XML
3. Operação, gerenciamento, requisitos de implantação
   * Necessidade de monitorar mensagens trocadas, reenvio de mensagens
   * Necessidade de reconfiguração dinâmica do middleware,
   * Gerenciamento via SNMP.
4. Serviço de transação
   * Mensagens send/receive com context transactional
   * Contexto transactional como uma transação de banco de dados,
   * Recuperação de falha, transações de longa duração
5. Eventos
   * Tratamento de eventos
6. Serviço de nomes
7. Requisitos de desempenho
   * Tempo de resposta aceitável para a aplicação, vazão, tamanho e duração da transação
   * Número de publishers/ subscribers
   * Balanceamento de carga
   * Suporte multithread
8. Linguagem adotada para construir a aplicação: Java, IDL + Java, IDL + C++, nesC, C, etc
9. Escalabilidade
10. Tolerância a falhas
11. Segurança
    * autenticação, autorização, criptografia, assinatura digital, gerenciamento e distribuição de chaves
12. Política de Invocação
    * *Thread* por mensagem, pool de *threads*, *thread* por cliente, único *thread*
13. Estratégia de dispatching
    * Localizar e ativar o objeto remoto,
    * Pre-ativar o objeto remoto
14. Camada de Transporte
    * Confiável/não confiável, combinação
    * TCP, UDP, outros
15. Protocolo de Interoperabilidade (formato da mensagem, sequencia de mensagens)
16. Outros

# Referências

1. Liu, A. & Liu, A. (2001), Gathering Middleware Requirements, *in* 'Proc. 15th International Conference on Information Networking', pp. 81--86.
2. Singhai, A.; Singhai, A.; Sane, A. & Campbell, R. (1998), Quarterware for middleware, *in* A. Sane, ed.,'Proc. 18th International Conference on Distributed Computing Systems', pp. 192--201.