

Integração de Sistemas

02 – Níveis de Integração de Serviços

Catarina Oliveira
1º Semestre
2020/2021



DEPARTAMENTO DE CIÊNCIA
E TECNOLOGIA



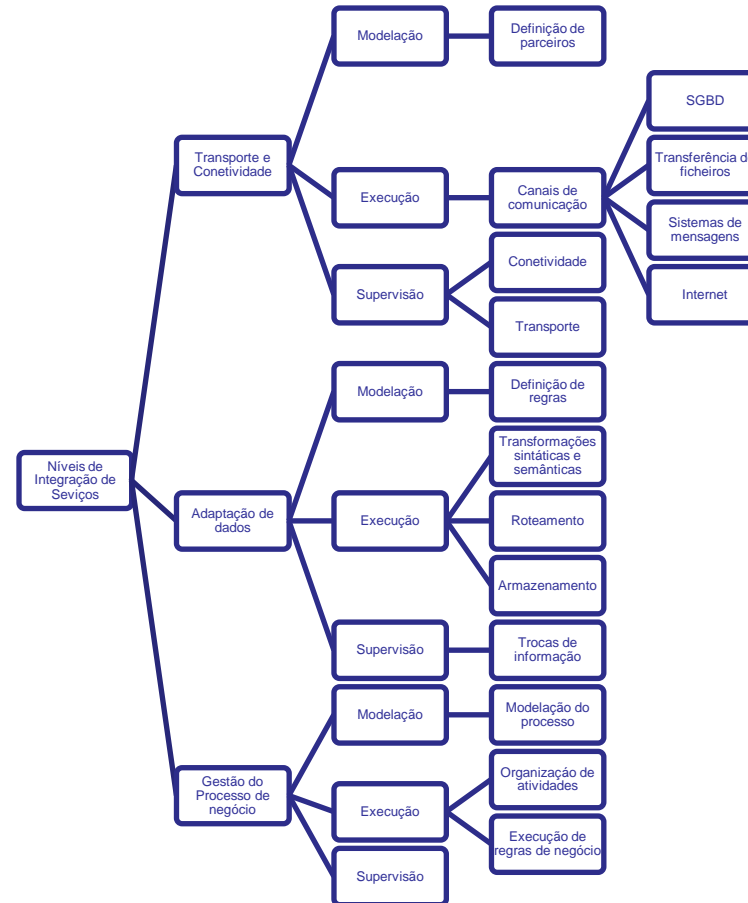
UNIVERSIDADE PORTUGALENSE

CONTEÚDO

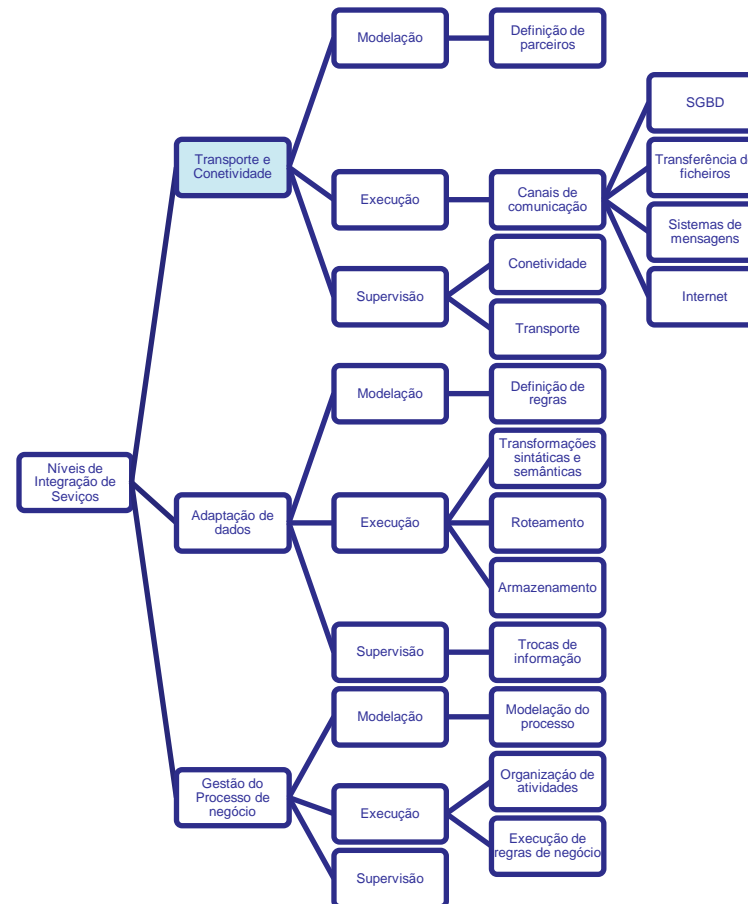
1. Transporte e conectividade
 1. Modelação: definição de parceiros
 2. Execução: canais de comunicação
 3. Supervisão
2. Adaptação de dados
 1. Modelação: definição de regras
 2. Execução
 1. Transformações
 2. Roteamento
 3. Armazenamento
 3. Supervisão de trocas de informação
3. Gestão do Processo de negócio
 1. Modelação do processo
 2. Execução
 3. Supervisão
4. Processo de negócio e integração: mediação e trocas
 1. Nível de processo de negócio e nível de integração
 2. Subnível de mediação de processo
 3. Subnível do processo de troca
 4. Interação entre subníveis
 5. Interação entre processo de integração e processo de negócio
 6. Relações e interações entre níveis e subníveis
 7. Como escolher a arquitetura de trocas
 1. Comunicação síncrona/assíncrona
 2. Arquitetura
 1. Centralizada
 2. Distribuída
 3. *Snowflake*

Níveis de integração de serviços

- Diz respeito aos níveis de serviço em que se dá a integração de sistemas



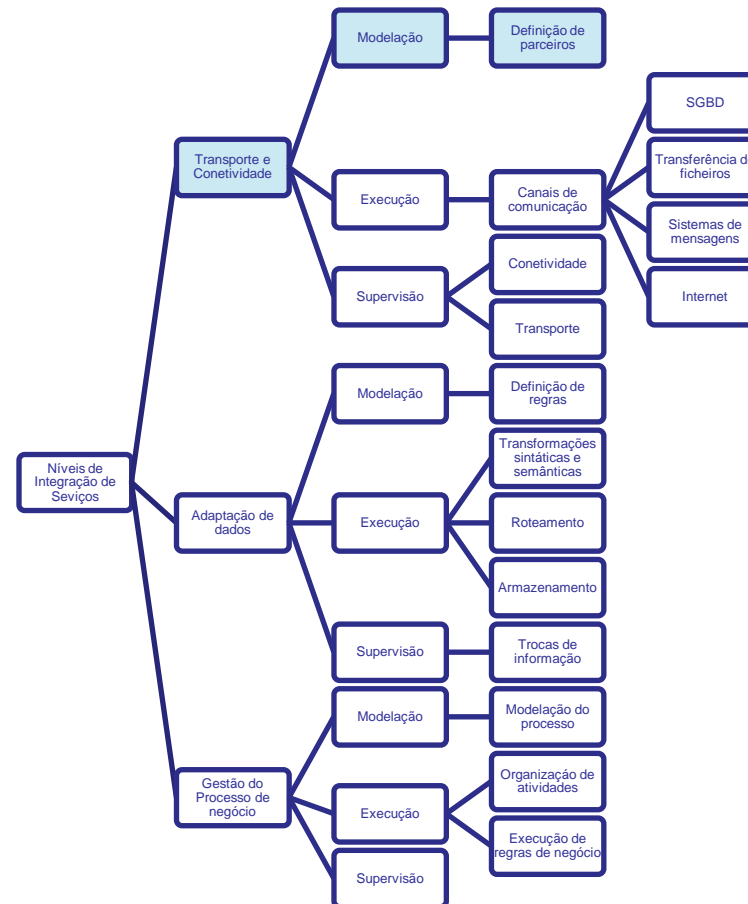
Níveis de integração de serviços



Transporte e conectividade

1. Captura informação ou eventos gerados pelas aplicações “fonte”
2. Transporta-os para as aplicações “recetoras”
3. Entrega as informações /dados que as aplicações “recetoras” esperam

Níveis de integração de serviços

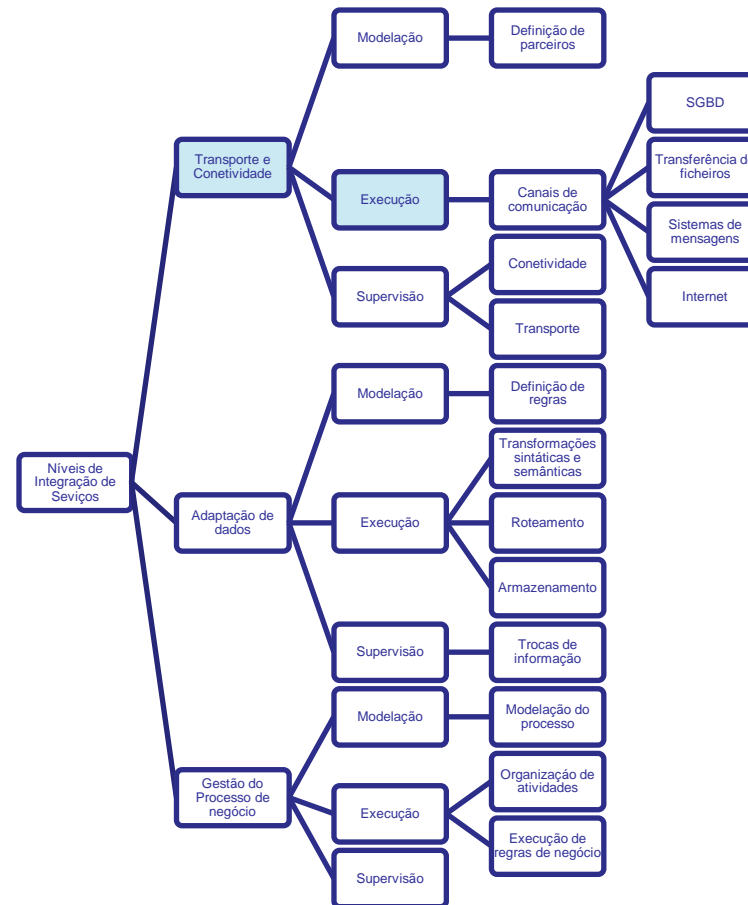


Modelação: definição de parceiros

- Para separar a aplicação dos parâmetros técnicos:
 - Parâmetros devem ser extraídos e centralizados num repositório central
 - Repositório:
 - Descreve o conjunto completo de elementos
 - Para garantir que os dados ou eventos sejam transferidos entre as aplicações.
 - O conteúdo do repositório pode ser distribuído às componentes de execução.
 - O repositório deve conter a seguinte informação:

Topografia das aplicações a serem integradas (informação para aceder às aplicações)	Possíveis condições para capturar e entregar dados ou eventos	Informação de segurança
Exemplos: <ul style="list-style-type: none"> • Localização (dentro da empresa ou, em caso de B2B, no sistema do parceiro) • Plataforma para a qual efetuar a ligação • Canal de comunicação a utilizar (inter-application messaging system, transferência de ficheiros, DBMS, HTTP, SMTP, ...) • Protocolos a utilizar • Endereço de rede • Identificador do gestor de BD • Identificador de tabelas no SGBD • Identificador para ficheiros • Nome das filas • Endereço do sistema de mensagens 	Exemplos: <ul style="list-style-type: none"> • <i>Timeslot</i> para captura ou entrega • Aplicações a despoletar na receção 	Exemplos: <ul style="list-style-type: none"> • Autenticação • Encriptação • <i>Hashing</i> • Não-repúdio

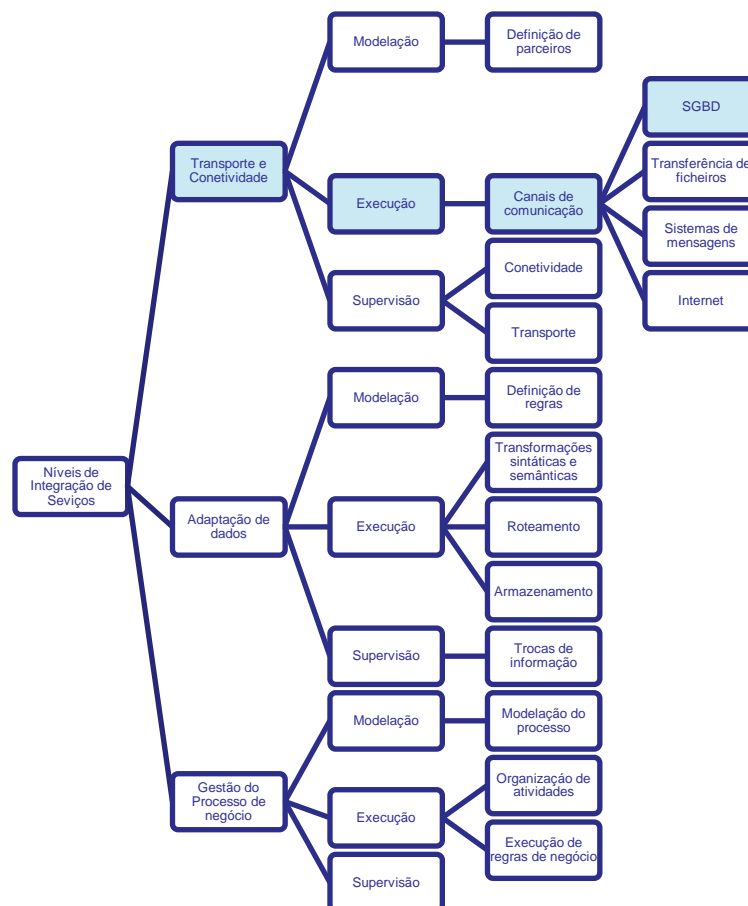
Níveis de integração de serviços



Execução: transporte de dados

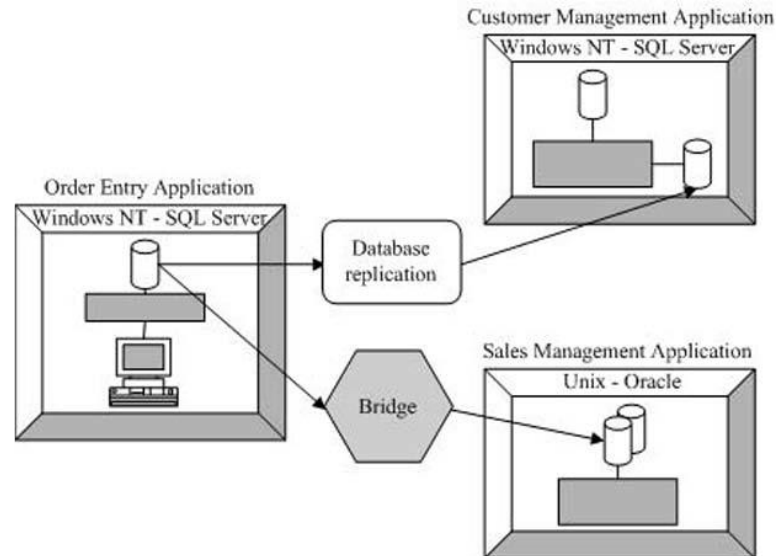
- Requer *bus* de comunicação multicanal.
- Deve adaptar-se aos diferentes suportes usados
 - Sem impor o uso privilegiado de nenhum em particular
 - (caso contrário iria requerer custos de conversão)

Níveis de integração de serviços

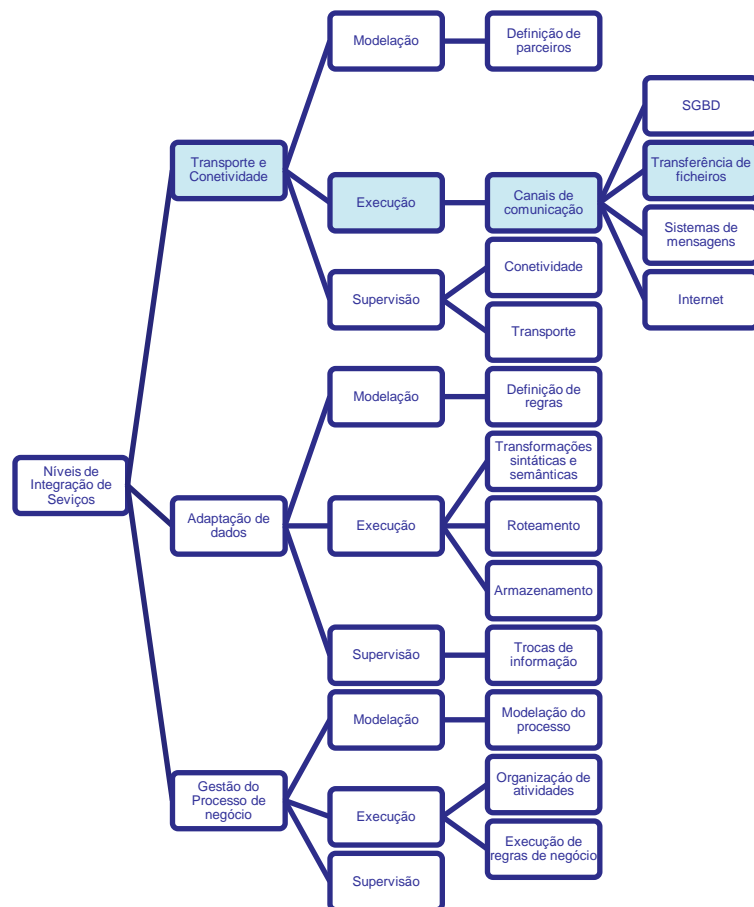


Sistema de gestão de bases de dados (SGBD)

- Oferece mecanismos de replicação básicos, ou inter-DBMS access bridges (gateways)
- Utilizados maioritariamente em projetos que requeiram integração de dados
- Essencialmente projetos de integração através de propagação de dados ou aplicações *composite*
- Na maioria dos casos, a semântica e modelos de dados usados nas aplicações integradas são compatíveis.



Níveis de integração de serviços



Transferência de ficheiros

- Forma de transporte de dados mais antigo em tecnologia de redes.
- Extensão: FTP (File Transfer Protocol)
- Vantagem: simples de implementar (standard, básico para a maior parte dos SO)
- Desvantagens:
 - pouca segurança,
 - não apto para ficheiros volumosos,
 - necessita de ação manual do utilizador
- Surgiu nos anos 1980 gestores de transferência de ficheiros mais elaborados
 - Oferecendo transferências de ficheiros mais seguras e automatizadas
 - (Managed File Transfer – MFT), compostos por:
 - serviços de rede
 - serviços de protocolo de transferência de ficheiros
 - serviços de gestão.

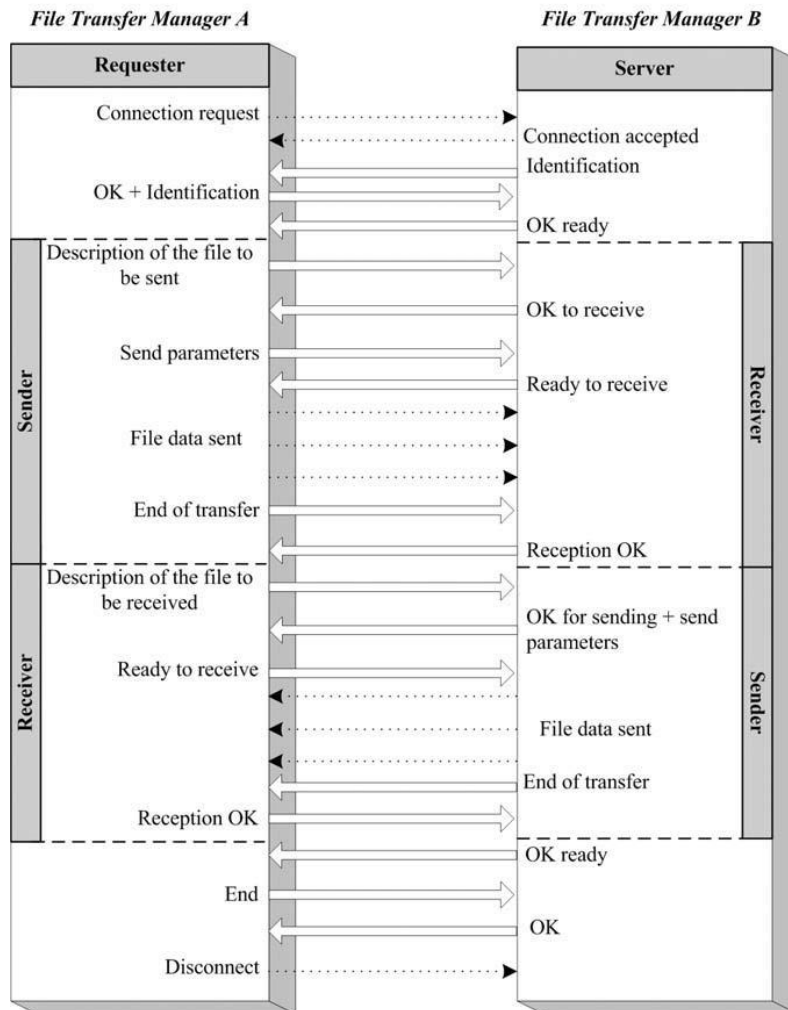
Transferência de ficheiros: MFT

- serviços de rede autorizam:
 - suporte para diferentes tipos de redes (TCP/IP, X.25, SNA, DSA, DNA, etc.)
 - comutação entre redes (ex: TCP/IP para X.25)
 - capacidade de iniciar uma ligação (papel: *requester*), ou esperar que a ligação de outro gestor (papel: *required*)
- serviços de protocolo de transferência de ficheiros permitem:
 - suporte para diferentes protocolos de troca de ficheiros (PeSIT, ETEBAC, OFTP, FTAM, etc.)
 - comutação entre protocolos de transferência (ex: ETEBAC para PeSIT)
 - compressão de dados
 - designação de pontos de sincronização para evitar ter de recomeçar uma transferência interrompida desde o princípio
 - garantia de entrega
- serviços de gestão providenciam:
 - capacidade de roteamento de ficheiros para outros destinos (mecanismo *store-and-forward*)
 - tratamento automático do reinício em caso de interrupção da transferência
 - ter em conta prioridades de transferência
 - agendamento de transferências
 - *logging* das operações

Transferência de ficheiros: API

- São, também, disponibilizadas APIs
 - Para que as aplicações possam aceder ao registo e, por exemplo:
 - Fazer pedidos de transferência
 - Consultar o estado de uma transferência
 - Incorporam mecanismos de segurança para garantir:
 - Autenticação
 - Encriptação
 - *Hashing*
 - Não-repúdio.

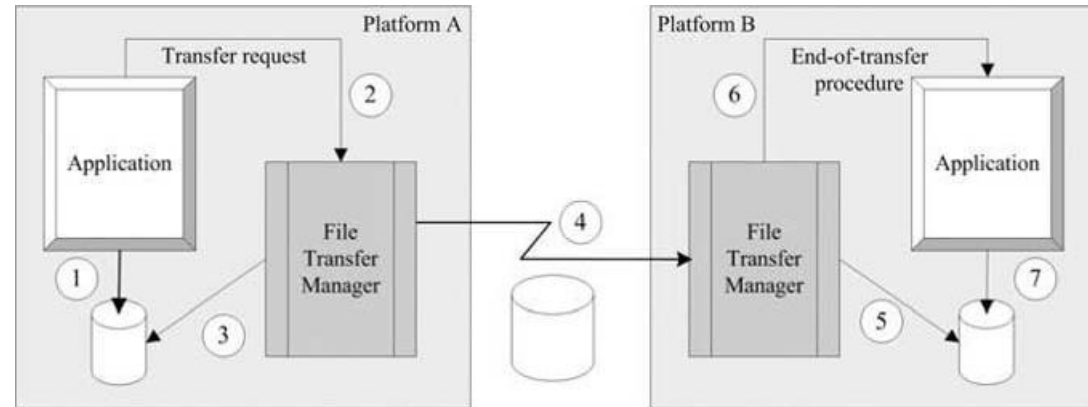
Transferência de ficheiros: Protocolo tradicional



Limitações:

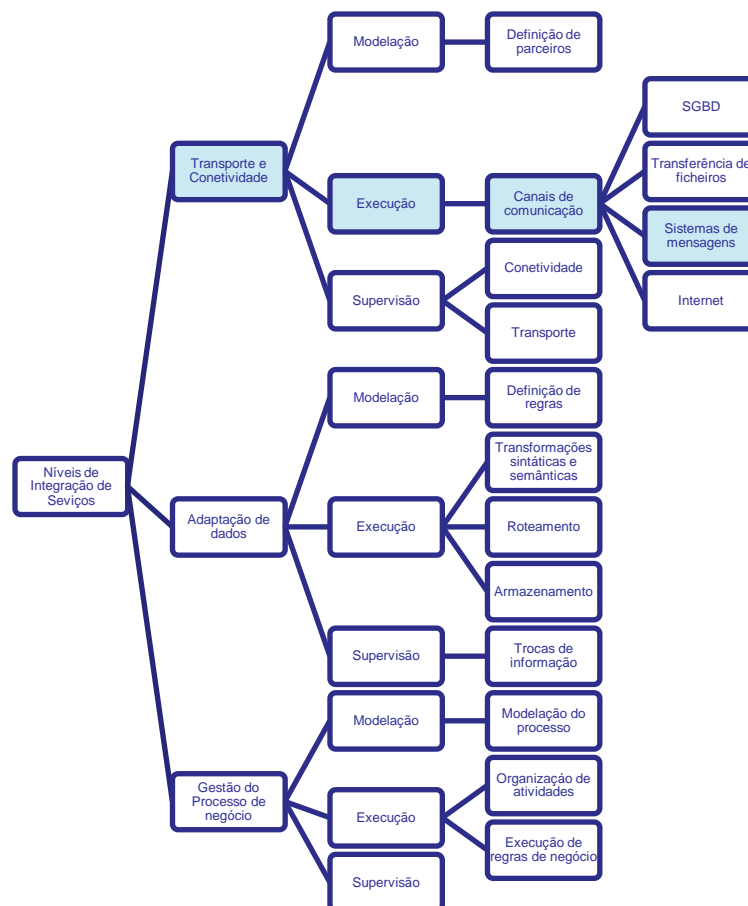
- Requisito de que os intervenientes suportem o mesmo tipo de rede e protocolo
- Trata relação manager-to-manager, mas não A2A
- Orientado a transferências *batch*
- Inércia da implementação e operação em estruturas pequenas

Gestor de transferências de ficheiros que ultrapassa limitações do protocolo tradicional



1. Uma aplicação na plataforma A é executada e cria um ficheiro
2. A aplicação envia o pedido de transferência ao gestor de transferência, especificando o nome do ficheiro a ser transferido
3. O gestor verifica a existência do ficheiro a transferir e as suas permissões e determina o gestor que o irá receber
4. Para iniciar a transferência, o gestor de envio estabelece uma sessão com o gestor recetor e implementa o protocolo de transferência de ficheiros mais adequado
5. O gestor recetor cria o "envelope" para o ficheiro ser recebido e escreve lá os dados recebidos
6. Na conclusão da transferência, utilizando um procedimento *end-of-transfer*, o gestor recetor despoleta a aplicação que irá consumir o ficheiro recebido
7. A aplicação na plataforma B consome o ficheiro recebido

Níveis de integração de serviços



Sistemas de mensagens inter-aplicações

- *Inter-application messaging systems*
- *Message-oriented Middleware (MOM)*
- Diferente de gestores de transferência de ficheiros
 - Permitem trocas A2A de eventos unitários na forma de mensagens
 - Adaptando-se ao ritmo de operação das aplicações
 - Através do uso de um sistema de filas (*queues*)
- Permite comunicação entre aplicações:
 - Na mesma máquina ou
 - Em máquinas diferentes
 - Exige que MOM esteja instalado em cada máquina)
- A comunicação é feita por troca de mensagens entre as MOMs instaladas nas diversas máquinas.
- Integra: serviços de transporte, serviços internos, serviços de gestão

Sistemas de mensagens inter-aplicações: serviços integrados

- Serviços de transporte autorizam:
 - Suporte a diferentes tipos de rede (TCP/IP, SNA, etc.)
 - Comutação de redes (ex: TCP/IP para X.25)
 - Capacidade de gerir ligações bidirecionais com outros MOM
 - Compressão de mensagens
 - Agrupamento de mensagens
 - Integridade de dados
 - Garantia de entrega (uma e apenas uma vez) das mensagens
- Serviços internos incluem:
 - Persistência de mensagens
 - Múltiplos modos de acesso a filas (FIFO, acesso direto, etc.)
 - Envio de mensagens em unidades transacionais apropriadas para as aplicações
 - Reinício em caso de falha
 - Comutação de mensagens
 - Notificação de receção
- Serviços de segurança providenciam
 - Lançamento automático de uma aplicação ao receber uma mensagem
 - APIs para a aplicação
 - *Logging* e supervisão das operações

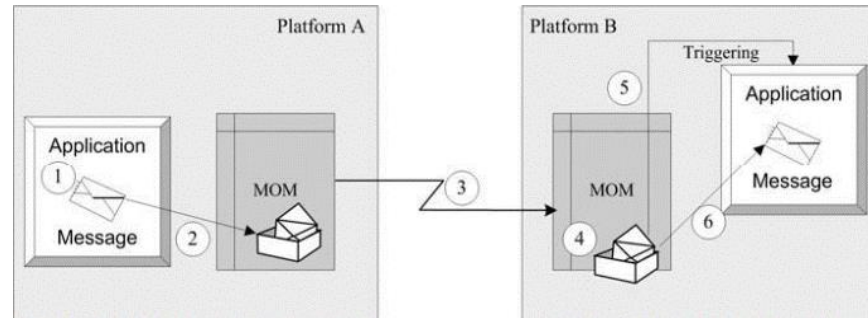
MOM

- MOM assenta num registo que descreve:
 - Características de outros MOM com quem comunica
 - Filas que gere e as suas características
- E contém mecanismos de segurança para assegurar:
 - Autenticação de parceiros
 - Encriptação de dados
- Oferece estes níveis de serviço na maioria das plataformas utilizadas em grandes SI
- Utilizado maioritariamente em projetos de integração de sistemas com múltiplos passos e por vezes em B2C, em certos projetos de integração de aplicações *composite*

MOM: Desvantagens

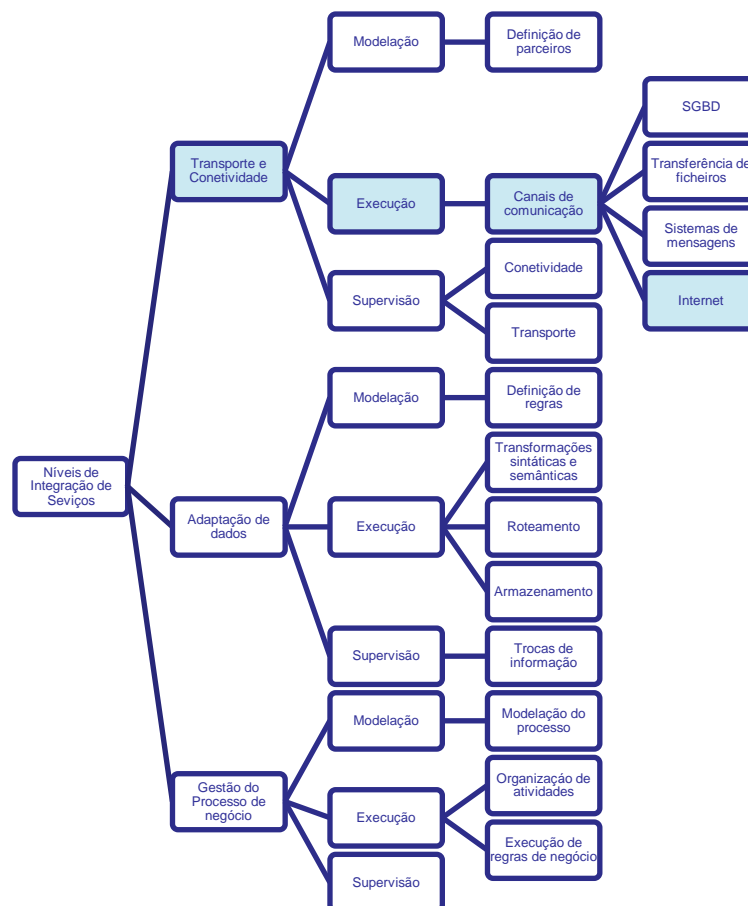
- Aspeto intrusivo: aplicações têm de ser alteradas para usar filas MOM
- Uma aplicação tem de saber com que aplicações comunica
 - Uma vez que é necessário dar o nome da fila recetora para chamar o serviço MOM.
 - Ultrapassado pela implementação de mecanismos de publicação/subscrição
 - Permitem publicar uma mensagem sem saber as aplicações que a vão utilizar
 - As aplicações interessadas neste tipo de mensagens é que subscrevem a fila para as receber
- Em modo persistente (segurança para cada mensagem) a performance é menos relevante do que a de transferência de ficheiros

MOM: Exemplo



1. A aplicação na plataforma A é executada, criando uma mensagem
2. A aplicação deposita a mensagem no MOM, especificando o nome da fila recetora na plataforma B
3. O MOM da plataforma A determina a localização da fila indicada. Como essa fila não está na plataforma A, envia-a para a plataforma indicada (neste caso a B)
4. Uma vez recebida a mensagem, o MOM da plataforma B coloca-a na fila recetora
5. Despoleta a aplicação que irá consumir a mensagem
6. A aplicação na plataforma B consome a mensagem

Níveis de integração de serviços



Internet: protocolo de comunicação HTTP

- **HTTP** (*Hypertext Transfer Protocol*)
 - Protocolo de comunicação cliente/servidor desenvolvido pelo W3C (*World Wide Web Consortium*)
 - Utilizado para trocas de todo o tipo de dados entre o cliente HTTP e o servidor HTTP
 - A principal utilização baseia-se na implementação de aplicações Web que gerem trocas com browsers através de páginas HTML (*Hypertext Markup Language* – também um standard W3C)
 - HTTPS é uma versão de HTTP cuja segurança assenta em protocolos:
 - SSL (*Secure Sockets Layer*), ou
 - TLS (*Transport Layer Security*)

Internet: protocolo de comunicação SMTP

- **SMTP** (*Simple Mail Transfer Protocol*)
 - Protocolo para troca de mensagens entre servidores através da internet.
 - Protocolo utilizado para envio de mensagens.
 - O recetor usa um cliente de mensagens e, para ir buscar as mensagens ao servidor, protocolo:
 - POP3 (*Post Office Protocol 3*), ou
 - IMAP (*Internet Message Access Protocol*)

Internet: protocolo de comunicação SOAP

- **SOAP** (*Simple Object Access Protocol*)
 - Protocolo para troca de dados em ambientes descentralizados e distribuídos.
 - Usa XML (*eXtensible Markup Language*) e define 3 componentes:
 - Envelope: *framework* que descreve o conteúdo da mensagem e como a processar
 - Conjunto de regras de codificação para os tipos de dados
 - Convenção para representar chamadas a procedimentos remotos e respostas associadas
 - Pode ser utilizado com vários outros standards
 - A especificação apenas descreve como implementar em comunicações HTTP.
 - É um dos standards em que assenta a definição de *Web Services*

Internet: protocolo de comunicação ebMS

- **ebMS** (*ebXML Messaging Service*)
 - Especificação ebXML (*Electronic Business using eXtensible Markup Language*)
 - Define um standard de transporte de mensagens seguro e confiável.
 - Características técnicas do protocolo:
 - Usa sintaxe SOAP para anexos
 - Não imposto nenhum protocolo de comunicação
 - Mais utilizados: HTTP e SMTP
 - Mensagens transportadas não são necessariamente expressas em XML
 - Aceita formatos tradicionais de EDI
 - Aceita mensagens baseadas em XML.
 - Usa técnicas de segurança existentes, como:
 - Assinaturas digitais XML(XMLDSIG)
 - Encriptação de chave pública

Internet: protocolo de comunicação RNIF

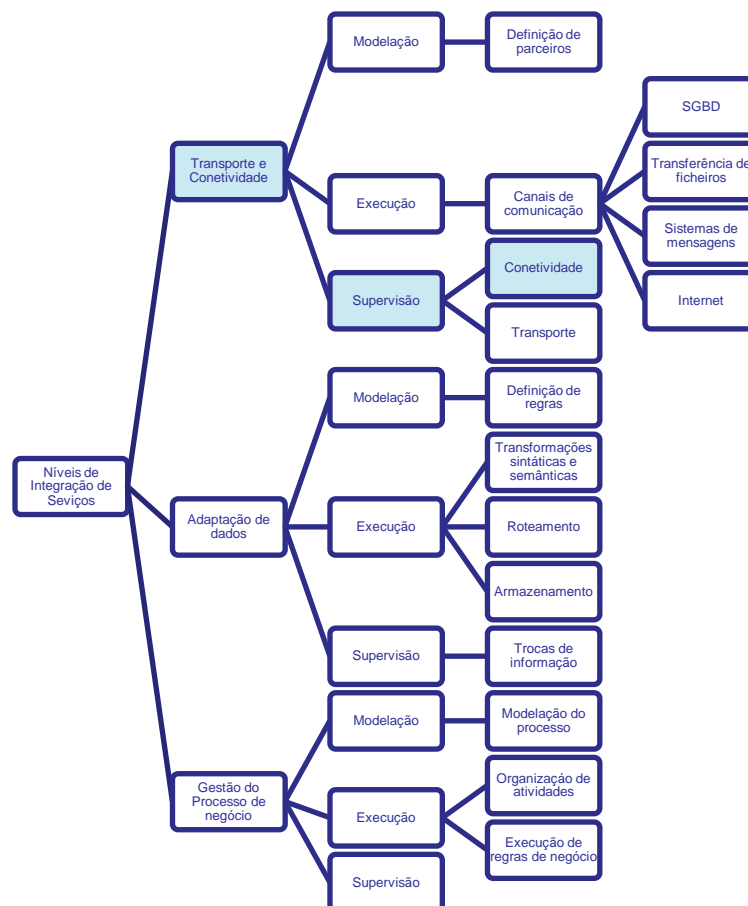
- **RNIF** (*RosettaNet Implementation Framework*):
 - Providencia a camada para
 - Empacotamento
 - Roteamento
 - Transporte
 - ... de mensagens para a execução de RosettaNet PIPs (Partner Interface Processes).
 - Usa os 2 standards MIME e S/MIME
 - Não impõe nenhum protocolo de comunicação
 - Favorece HTTP e SMTP

Internet: protocolo de comunicação EDIINT

- **EDIINT**

- Protocolos desenvolvidos por utilizadores de EDI para transportar dados através da internet.
- Usa os standards MIME e S/MIME (para encriptação e assinatura)
- Integra funções indispensáveis à troca de documentos:
 - Notificação de receção/não receção
 - Recibo de não-repúdio
- O remetente recebe uma notificação de que a mensagem foi corretamente descriptada
 - (MDN: Message Disposition Notification)
 - garante que processo de transação se deu corretamente do princípio ao fim
 - e que está assinado e encriptado.

Níveis de integração de serviços



Supervisão: Conetividade (*Connectivity*)

- O *bus* de comunicação multicanal assegura o transporte de dados/eventos:
 - Adaptadores: Os dados/eventos produzidos por uma aplicação (*upstream*) devem ser capturados e transmitidos para a aplicação (*downstream*) que os espera

Supervisão: Abordagem geral

- A primeira preocupação ao integrar uma aplicação consiste na não-intrusão no que diz respeito:
 - À aplicação
 - Às outras aplicações do SI
- Um dos maiores problemas nas aplicações existentes é o seu conhecimento:
 - das aplicações recetoras
 - das necessidades das aplicações recetoras
 - das regras de negócio das aplicações recetoras

} frequentemente *hardcoded*
- Necessidade de não-intrusão pode ser reduzida tentando permanecer não intrusivo antes das aplicações existentes terem de começar a respeitar as regras da “urbanização”.

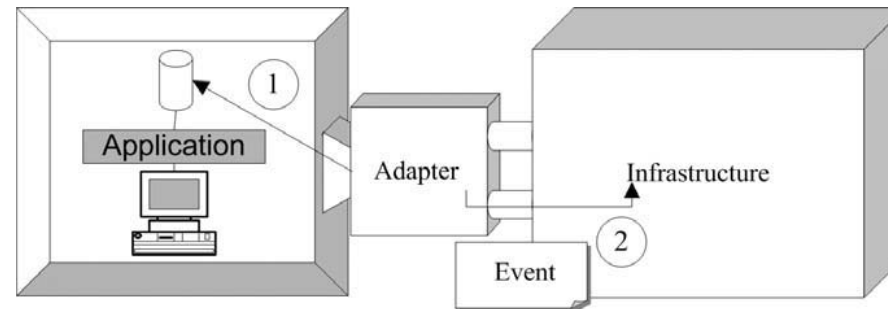
Supervisão: Abordagem geral – não intrusão

- Diferentes aspetos:
 - A informação deve ser obtida da aplicação
 - No formato em que foi produzida (ex: ficheiro, mensagem numa *queue*, elemento de uma tabela da BD, etc)
 - Respeitando a estrutura e os formatos dos dados
 - Na plataforma de hardware em que foi criada
 - A informação deve ser disponibilizada às aplicações
 - No formato esperado (ex: ficheiro, mensagem numa *queue*, elemento de uma tabela da BD, etc)
 - Respeitando a estrutura e formato esperados
 - Na plataforma de hardware esperada
 - Não deve ser necessário modificar as aplicações existentes no SI para comunicarem com novas aplicações
- Técnica para respeitar o princípio de não-intrusão: implementação de adaptadores (*adapters*)

Adaptadores

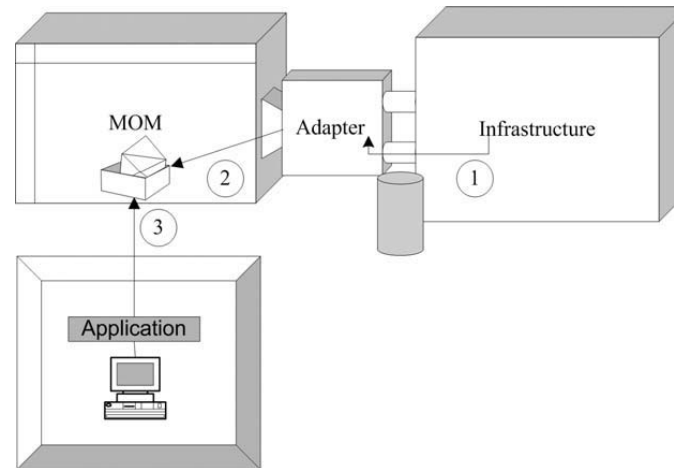
- Por vezes chamados de conetores (*connectors*)
- Objetivo: ligar fisicamente a aplicação à infraestrutura de comunicação
- Deve implementar 2 elementos
 - Ligação à aplicação
 - Ligação à infraestrutura
- A natureza do adaptador
 - Diretamente ligada com a capacidade da aplicação de comunicar com o mundo exterior
 - Mais especificamente, dos meios que o adaptador providencia para comunicar:
 - APIs (*Application Programming Interface*)
 - Interfaces com pacotes de *middleware* de comunicação (MOM, gestores de transferência de ficheiros, etc.)

Adaptadores: Exemplo de mecanismo *file adapter*



- Aplicação que produz ou lê ficheiros não oferece um meio específico de comunicação. O adaptador deve
 1. Detetar quando a aplicação cria um ficheiro (mecanismo de monitorização de catálogos ou pastas), e depois
 2. Gerar um evento que indica à infraestrutura que o ficheiro foi criado.
- Inversamente, o adaptador deve ser capaz de:
 1. Gerar um ficheiro para a aplicação
 2. Assegurar-se de que a aplicação o consegue processar, quer diretamente (*triggering*) ou pela publicação de um evento num *scheduler* que irá lidar com o lançamento da aplicação.
- Uma aplicação transacional ou cliente/servidor que use um SGBD, provavelmente não irá oferecer outro meio de comunicação. O adaptador deve:
 1. Detetar os eventos ocorridos na BD (triggers da BD, ou monitorizando alterações à BD)
 2. Gerar um evento indicando à infraestrutura que ocorreu uma alteração
- Inversamente, deve:
 1. Ser capaz de atualizar a BD diretamente

Adaptadores: Exemplo de mecanismo *file/message adapter*



- Uma aplicação *near-real-time* apenas sabe comunicar através do MOM ao qual está ligado.
- Para que seja capaz de, por exemplo, receber eventos de uma aplicação *batch* que os gera num ficheiro, sem ter de modificar a aplicação *batch* precisa deste adaptador
- O adaptador:
 1. Lê os eventos no ficheiro produzido pela aplicação *batch*
 2. Gera mensagens ou grupos de mensagens
 3. Coloca mensagens nas filas para que sejam consumidas pela aplicação *near-real-time*.

Adaptadores: *file/message adapter*

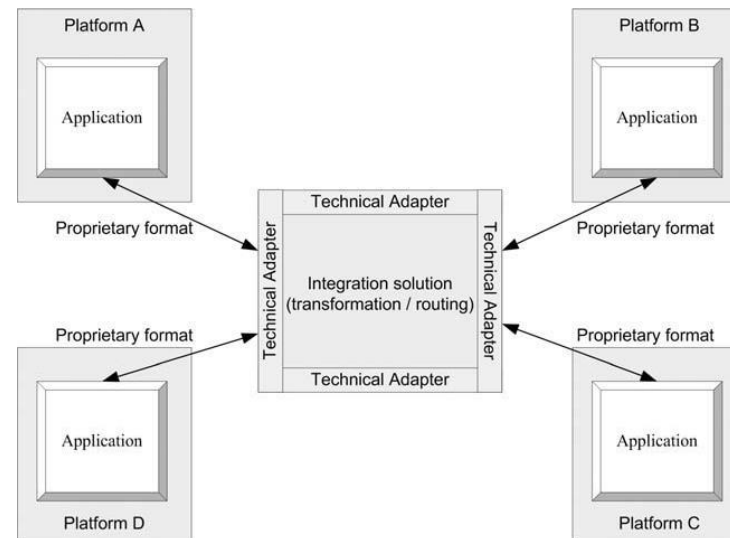
- Uma aplicação que ofereça múltiplos pontos de comunicação irá facilitar o desenvolvimento deste adaptador.
- É o caso de certos ERP (ex: R3 da SAP), que oferecem diversos métodos para trocas de informação com os módulos que os compõem:
 - API (BAPIs)
 - Comunicação em modo batch (BTCL)
 - Comunicação em modo de conversação (ALE)
 - etc.

Adaptadores: *file/message adapter*

- No entanto, estes modos de comunicação não apresentam a mesma cobertura ou capacidades técnicas
 - em termos de volume a processar
- Na maioria dos casos, com estes tipos de pacote de software
 - São necessários vários adaptadores
 - Cada um utilizando um método de comunicação
 - Em certos casos, um único adaptador que permita todos os métodos.
- Para além disso, um adaptador deve, sempre que possível, permitir:
 - Que a aplicação obtenha a descrição (metadados/metadata) dos dados que manipula,
 - para que seja possível popular o dicionário
 - Em que são definidas as regras de transformação das estruturas dos eventos/dados.

Technical (“thin”) adapters

- Não requerem instalação de nenhum componente na plataforma hospedeira da aplicação
 - Porque usam meios de comunicação já disponíveis nessa plataforma.
- Comunicação entre a aplicação a integrar e o *core* da solução de integração
 - Para assegurar o transporte da informação no formato proprietário de cada aplicação
- Processos de transformação e roteamento executados na plataforma hospedeira do *core* da solução de integração

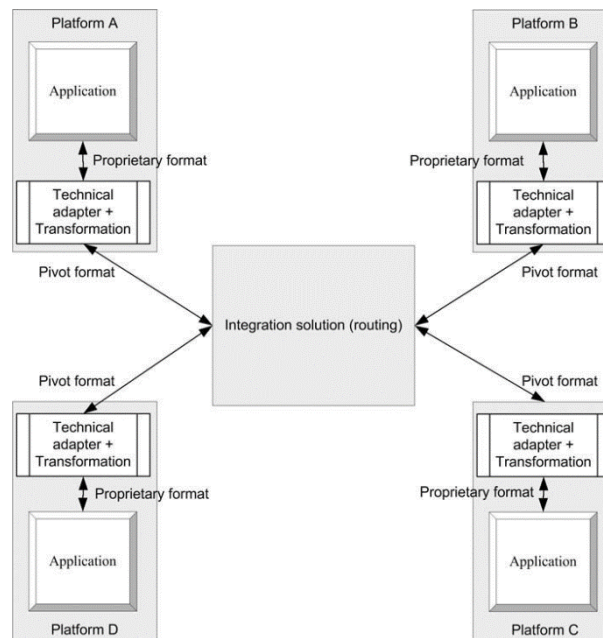


Technical (“thin”) adapters

Vantagens	Desvantagens
<ul style="list-style-type: none"> • Transformação apenas na plataforma hospedeira do core da solução de integração • <i>Deployment</i> facilitado por não ser necessária instalação de componentes na plataforma hospedeira da aplicação a ser integrada • Muito pouco sensível ao desenvolvimento de pacotes de software, por ser independente do formato de dados e ligado apenas ao tipo de comunicação utilizado pelos pacotes a integrar • Capacidade de gerir regras de transformação complexas, por estarem centralizadas no core da solução de integração • Preço acessível 	<ul style="list-style-type: none"> • Não carrega direta e automaticamente as descrições das estruturas já definidas no ERP, possibilitando dificuldades na entrega de: <ul style="list-style-type: none"> • parâmetros pré-definidos para uso standard em pacotes de software • regras de transformação que podem ser definidas entre formatos proprietários • Não processa automaticamente os estados para ter em conta <i>dataflows</i> • Define ligações <i>point-to-point</i> entre as aplicações por meio de regras de transformação (recria spaghetti <i>architecture</i>) [pode ser ultrapassado usando um formato pivot na sequência de transformações] • Complexidade de certas regras de transformação • Potencial bottleneck, porque todas as transformações são executadas no core da solução [possibilidade de escalabilidade pode ajudar a mitigar]

Business (“thick”) adapters

- Podem ser usados para assegurar funções de transformação de informação descentralizadas.
- Tarefa:
 1. Transformar os dados do formato proprietário da aplicação remetente para um formato pivot ou canónico
 2. Transformar o formato canónico no formato da aplicação recetora
- Requerem instalação de componentes na plataforma hospedeira da aplicação a integrar e na que hospeda o core da solução de integração.



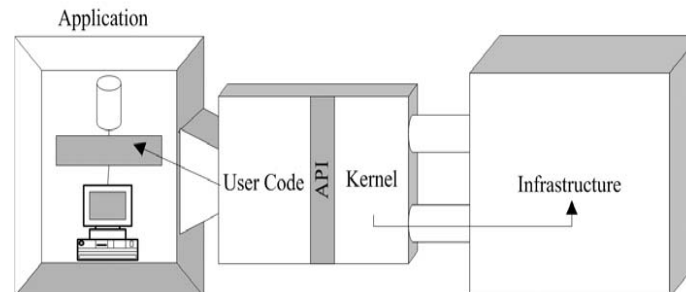
Business (“thick”) adapters

Vantagens	Desvantagens
<ul style="list-style-type: none"> • Uso sistemático de um formato pivot ou canônico para constituir um repositório de eventos de negócio standardizados • Possibilidade de envio de um conjunto de parâmetros juntamente com a solução de integração (<i>plug-and-play</i>) para o uso standard de certos pacotes de software, devido ao uso do formato canônico • Alocação da carga de trabalho das transformações entre as plataformas que suportam os adaptadores 	<ul style="list-style-type: none"> • Necessidade de transformações duplas, aumentando a carga global de trabalho, com potencial impacto no desempenho da solução • Parâmetros predefinidos para ligar pacotes de software não são utilizáveis se a empresa não usar formatos standard nesses pacotes • Aumento da sensibilidade ao desenvolvimento de pacotes de software, dado que é muitas vezes diretamente ligado à versão do pacote de software utilizado • Necessidade de publicar componentes na plataforma da aplicação com atualizações frequentes das definições principais de parâmetros • Não têm em conta aplicações <i>legacy</i> • Preço mais alto, por incorporarem funções de transformação avançadas

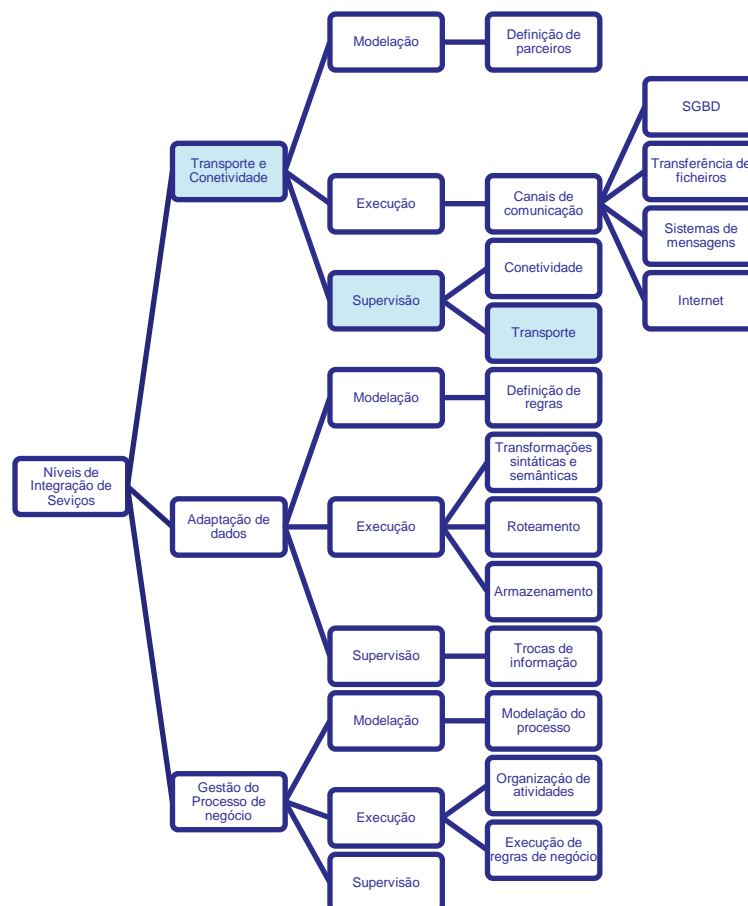
Por vezes é difícil escolher entre *thick* e *thin adapters*, sendo preferível utilizar ambos ao mesmo tempo

Ferramentas para o desenvolvimento de adaptadores

- É improvável que as soluções standard existentes no mercado consigam providenciar o conjunto necessário de adaptadores para integrar todos os tipos de aplicação existentes em diferentes SI
- É recomendado utilizar uma ferramenta para desenvolver o conjunto de adaptadores em falta
- No mínimo, esta ferramenta deve ter:
 - Um core, para assegurar a comunicação com a infraestrutura
 - Uma API, que possa ser usada para chamar as diferentes funções do core
- Depois pode desenvolver-se o código necessário para a integração.
 - Este código irá implementar o algoritmo necessário para encadear os acessos à aplicação e as chamadas aos serviços core de forma a gerar o conjunto de eventos correspondente na infraestrutura.
- código + API + core = adaptador



Níveis de integração de serviços



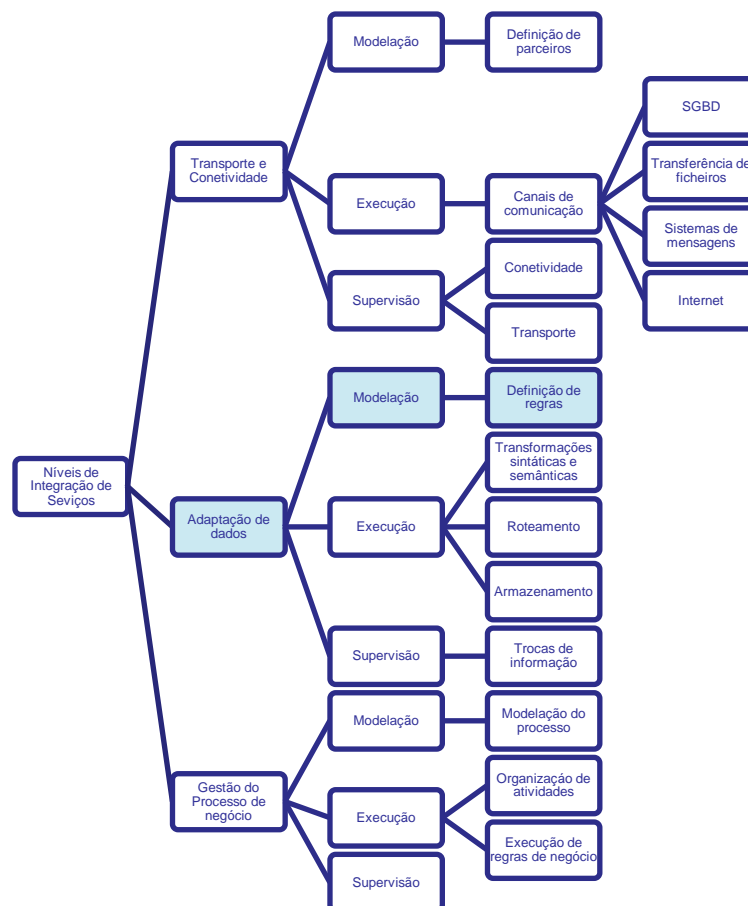
Supervisão de transporte

- Os dados/eventos transmitidos entre aplicações podem conter informação crítica para a empresa
- Deve ser possível determinar, em qualquer altura, onde podem ser encontrados
- Isto é ainda mais importante porque os mecanismos de troca são frequentemente assíncronos
- Quando duas aplicações trocam informações de modo **síncrono**, estão ligadas uma à outra.
 - Se uma delas fica indisponível, a outra imediatamente deteta o facto e executa as instruções planeadas para esse caso (ex: publicação de erro ou mensagem de alerta)
- Em caso de comunicação **assíncrona**, as aplicações não são informadas de problemas que ocorram durante a troca
 - Objetivo a ser atingido durante a implementação da infraestrutura de integração:
 - Aplicações devem ser independentes das condições técnicas da troca.
 - Os incidentes devem ser detetados pelas ferramentas

Supervisão de transporte

- As ferramentas de transporte e os adaptadores utilizados devem possibilitar envio de informação sobre o progresso das operações de roteamento para um sistema de supervisão.
- Devem indicar assim que ocorra um incidente:
 - Problemas de ligação entre gestores de transferência de ficheiros, entre MOMs ou com a aplicação
 - Reinícios que possam ser executados para transmitir a informação
 - Problemas do ambiente (ex: falta de espaço em disco para criar um ficheiro, inexistência ou saturação de uma fila, etc.)
- Em caso de incidentes graves, estas ferramentas devem publicar alertas que permitam aos operadores intervir de forma a corrigir o problema e voltar a lançar as operações interrompidas
 - É necessário um sistema de supervisão global que:
 - Centralize a totalidade da informação e eventos de supervisão
 - Ofereça uma interface em que os operadores possam verificar a informação e tratar das intervenções.
 - Interface poderá apresentar *dashboards* que permitam a visualização de;
 - Infraestrutura de transporte
 - *Dataflows*
 - Indicadores que permitam avaliação global do processo.

Níveis de integração de serviços



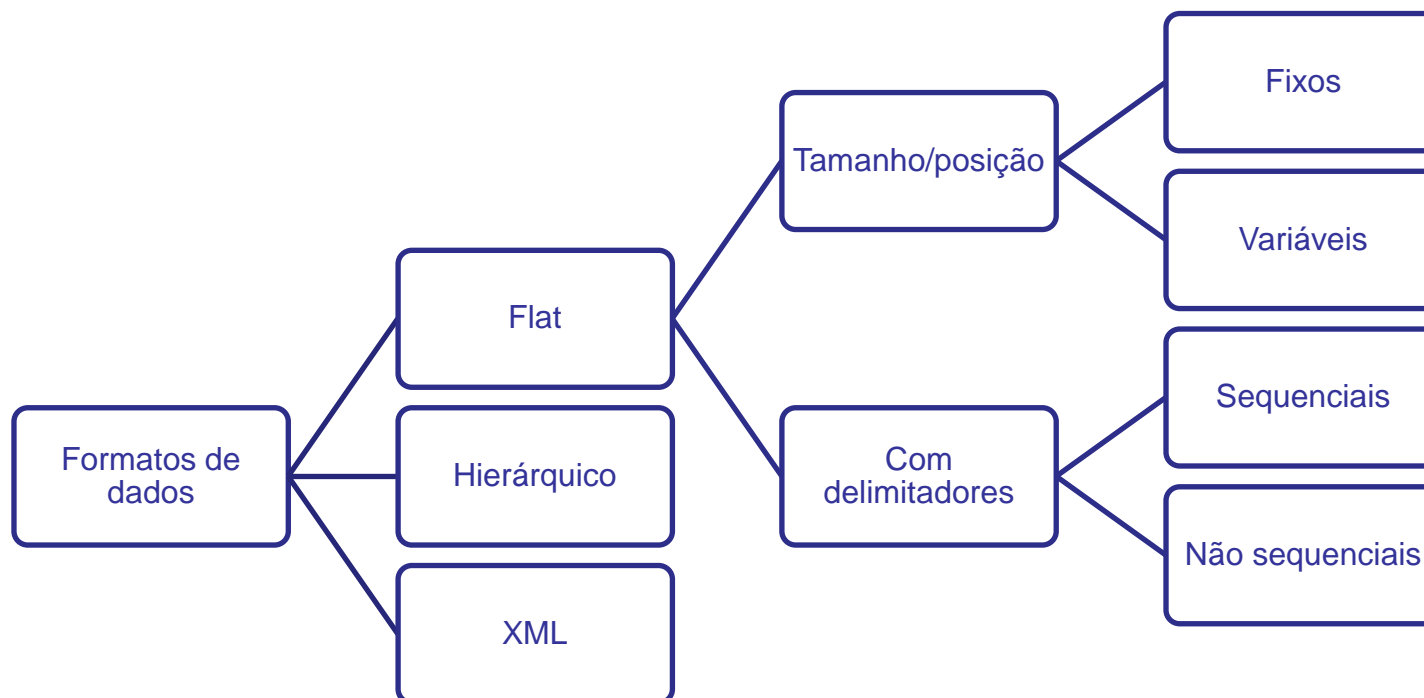
Adaptação de dados/informação

- A captura, transporte e entrega de informação aos recetores não são os únicos problemas da integração.
- Deve também ser possível adaptar os eventos produzidos pelas aplicações originais ao esperado pelas aplicações recetoras
- Deve ser possível determinar as aplicações recetoras e entregar o que esperam no momento certo.
- É este o objetivo do nível de adaptação de informação

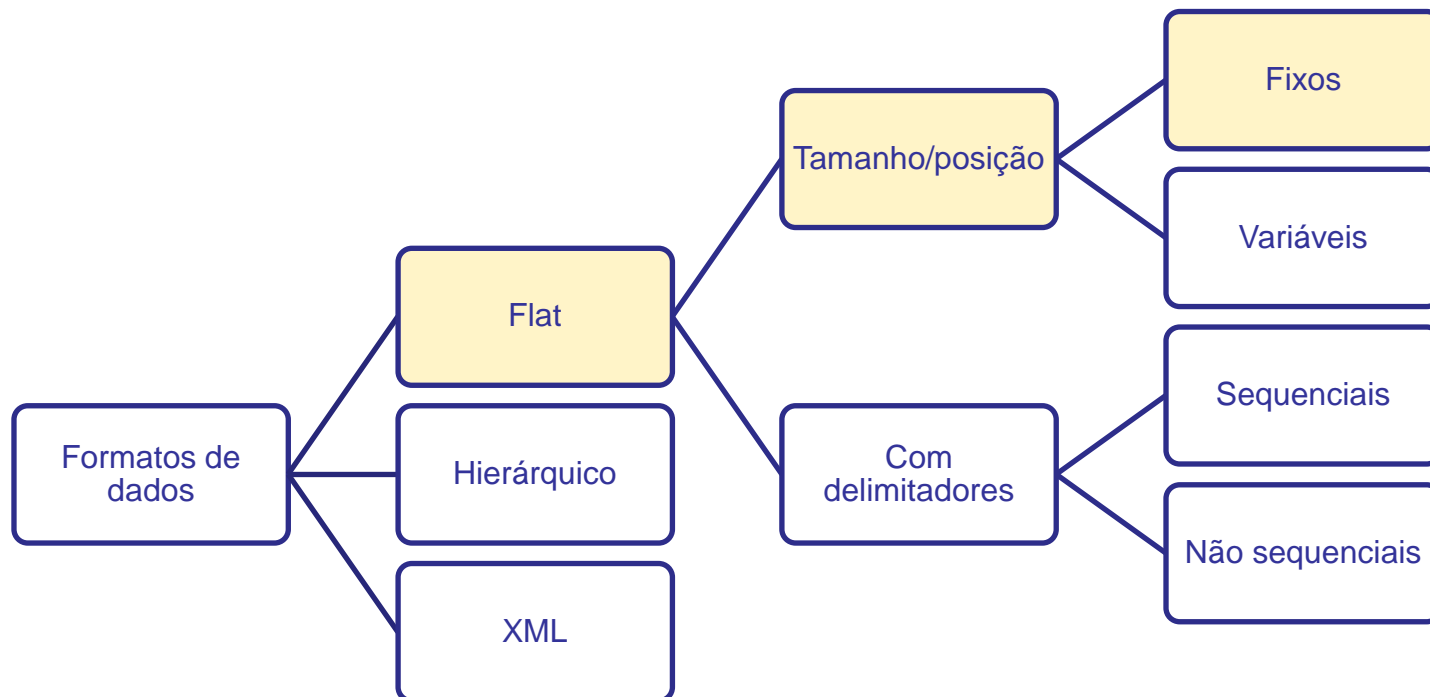
Transformação

- Um evento deve ser transformado quando é expresso num formato ou utilizando sintaxe que não é diretamente compreensível pelas aplicações recetoras.
- No entanto, a necessidade de transformar eventos pode também ter origem no facto de conter informação utilizada para gerar outros eventos de natureza diferente (ex: ordem de compra gerar movimento de stock)
- A infraestrutura de integração de aplicações deve oferecer as ferramentas necessárias para descrever as regras de transformação para os eventos e executá-las, independentemente da natureza da transformação a ser executada.
- O aparecimento de novos formatos ligados à internet reforçou esta necessidade.

Formatos de dados



Formatos de dados

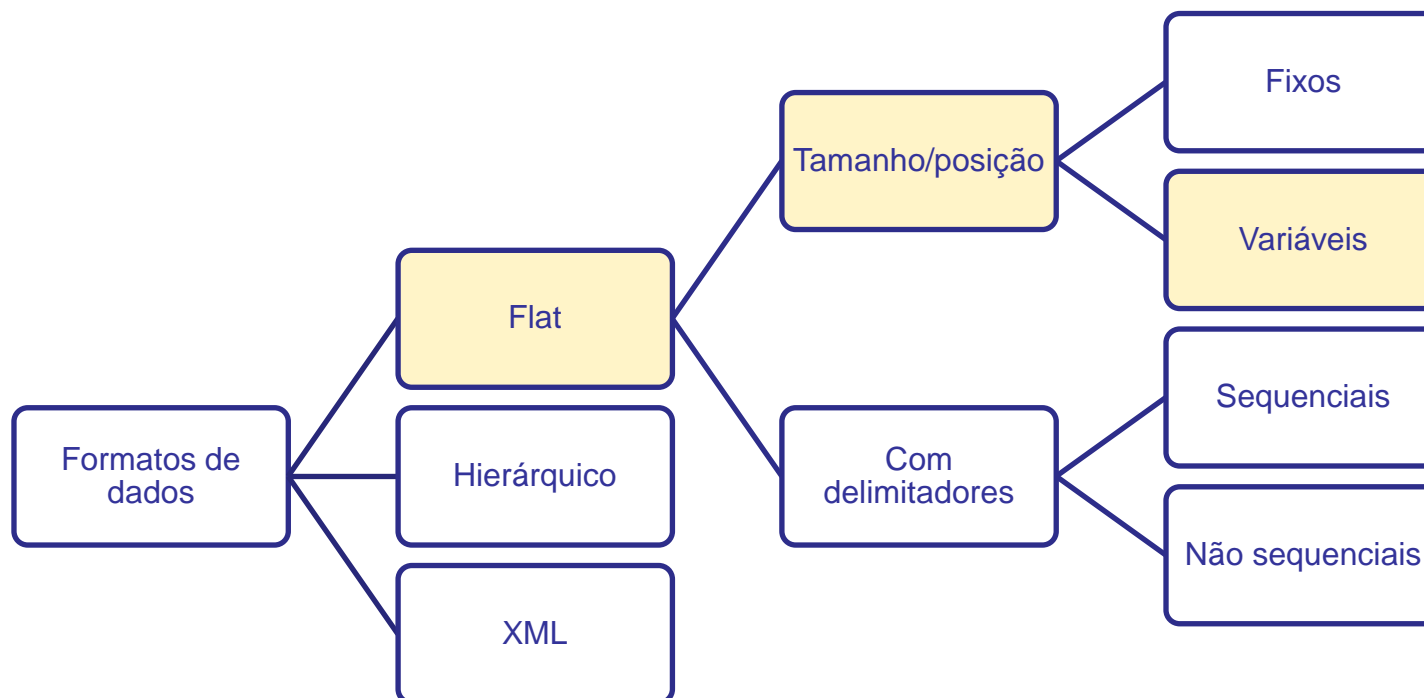


Flat format com tamanho/posição fixos

[dados 1] [dados 2] [dados 3] ... [dados n]

- Formato mais antigo e com melhor performance no domínio de gestão da computação
- Cada item dos dados está numa posição pré-determinada e tem sempre o mesmo tamanho.
- Quando o tamanho é menor ou quando há elementos em falta, preenche-se com valor pré-definido (ex: 0).
- Possibilita a localização imediata de um elemento específico, não tendo impacto na performance aquando da manipulação dos dados.

Formatos de dados

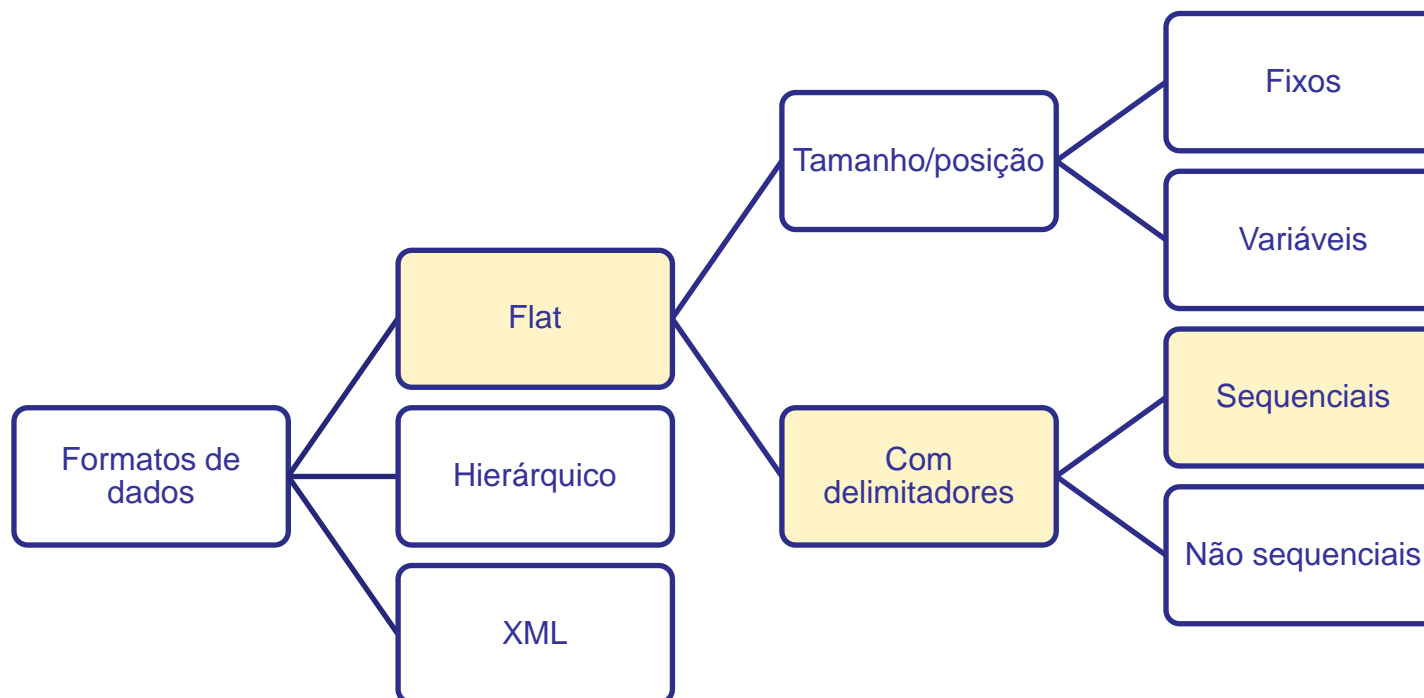


Flat format com tamanho/posição variáveis

[c 1][dados 1] [c 2][dados 2] [c 3][dados 3] ... [c n][dados n]

- Variante do formato anterior.
- Posição depende do tamanho.
- Cada elemento é precedido por um contador que indica o seu tamanho.
- Elementos em falta têm contador = 0.
- Para localizar um determinado elemento é necessário analisar o tamanho representado no contador que o precede.
 - Implica *parsing*, o que pode ter impacto na performance aquando da manipulação de dados.

Formatos de dados

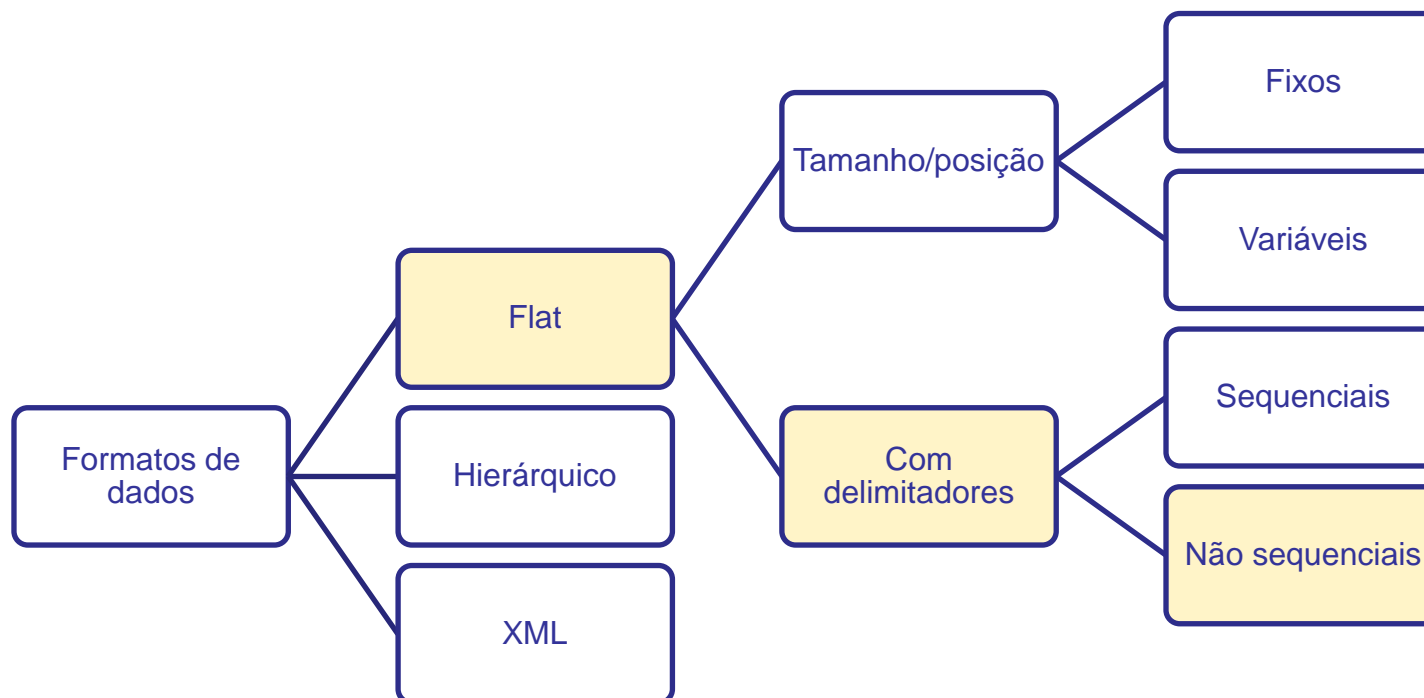


Flat format sequencial com delimitadores

[dados 1],[dados 2],[dados 3], ... ,[dados n]

- Dados sempre com a mesma ordem e separados por um carater (neste caso a vírgula).
- É necessário analisar a mensagem para determinar o separador
 - Localização de um elemento depende da sua posição relativa
 - Impacto na performance aquando da manipulação dos dados

Formatos de dados

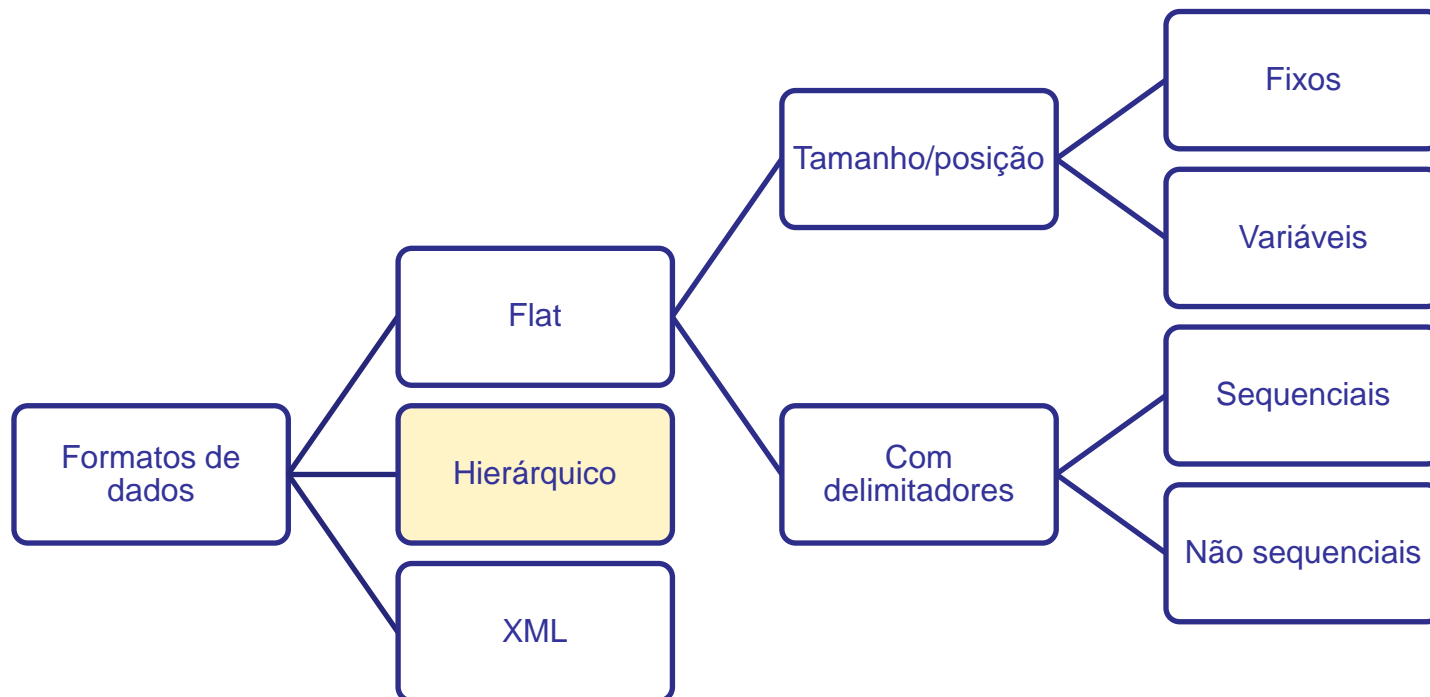


Flat format não sequencial com delimitadores

kw 2=dados 2, kw n=dados n, ..., kw 1 = dados 1

- Dados podem chegar em qualquer ordem.
- Para além do delimitador, são necessárias *keywords* para os identificar.
- Análise necessária para localizar os dados
 - Impacto na performance aquando da manipulação de dados.

Formatos de dados

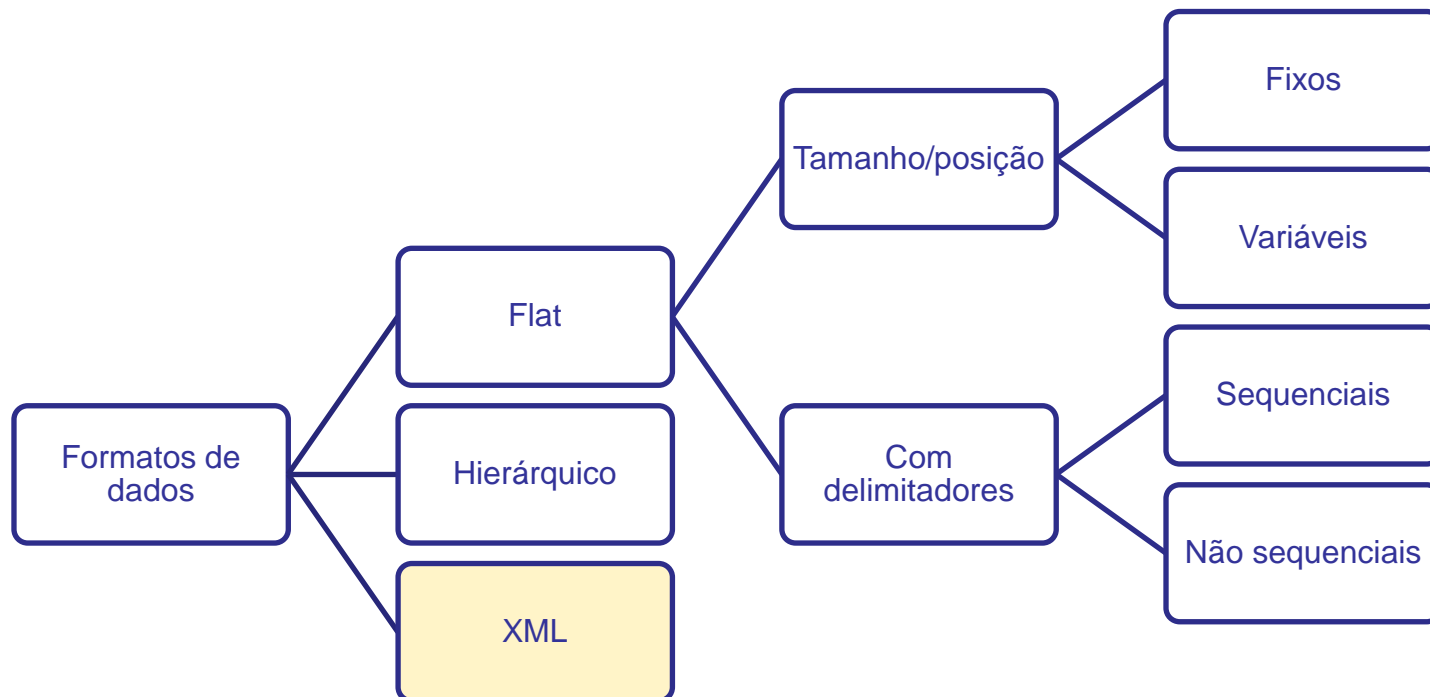


Formato hierárquico (em árvore)

kw 1=(dados 1, dados 2,kw 11=(dados 11a, dados11b),kw 5=dados 5),kw 7=dados 7,

- Pode ser combinado com os formatos referidos anteriormente.
- Certos dados (compostos) são constituídos por outros dados (elementares) ou outros conjuntos de dados compostos.
- A localização dos dados complica a análise
 - Impacto na performance

Formatos de dados



Formato XML (eXtensible Markup Language)

<https://www.w3.org/standards/xml/core>

- Linguagem que permite descrever a estrutura lógica de documentos
 - baseando-se num sistema de tags que marcam:
 - elementos que compõe a estrutura
 - relações entre esses elementos
- objetivos:
 - separar a estrutura do conteúdo
 - formalizar a estrutura
 - standardizar as tags (sintaxe)
 - restringir os documentos usando modelos semânticos
 - facilitar o parsing

XML: tipos de documentos

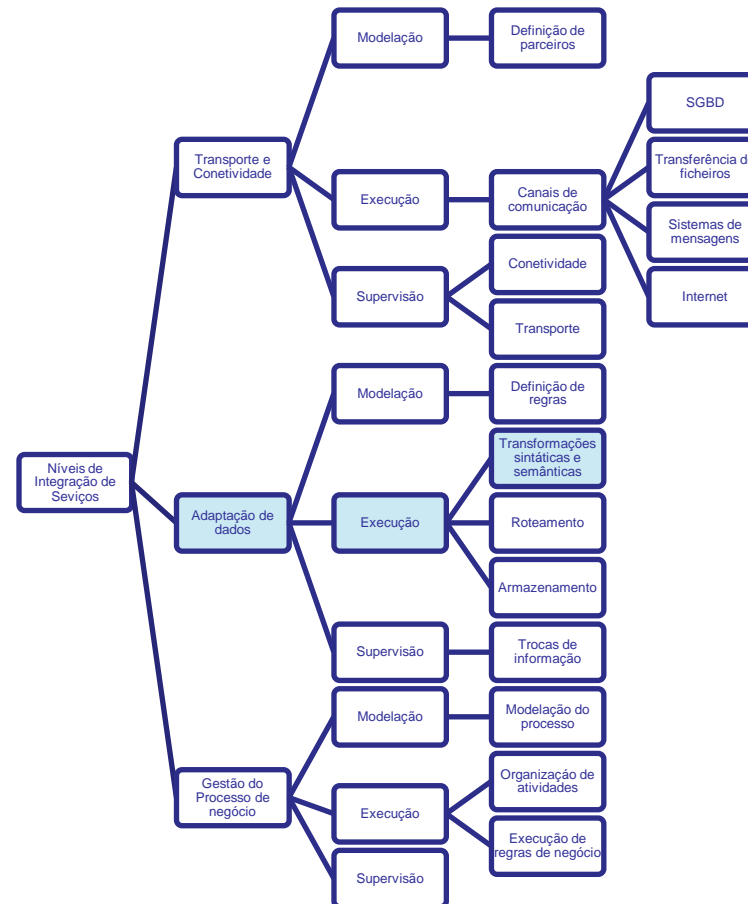
Documentos XML	DTDs (<i>Document Type Definition</i>)	XML Schemas
	modelos usados para descrever os elementos e valores default de atributos	modelos usados para definir as propriedades dos elementos, as relações entre si, os tipos de dados e hierarquia de classes. Usam linguagem XML
<pre> <Order> <CurrencyCode>FRF</CurrencyCode> <PaymentMode>transfer</PaymentMode> <OrderDate>20000310</OrderDate> <CustomerReference> <Name>Mr. BEAN</Name> <CustomerCode>007</CustomerCode> <Company>Editions HERMES</Company> <Address> <Address1>Editions HERMES</Address1> <Address2>8, qual du Marché-Neuf</Address2> <PostalCode>75004</PostalCode> <City>Paris</City> <Country>FRANCE</Country> </Address> </CustomerReference> <OrderLine> <Quantity>25</Quantity> <ProductDescription>PC, Pentium III 500 MHz, 8 processors, RAM 256 Mb</ProductDescription> </OrderLine> </Order> </pre>	<pre> <!ELEMENT doc (A, B, C?, (D E)+, F*) > <!ATTLIST doc Date CDATA "20000114" Color (Red Black) #IMPLIED > </pre> <ul style="list-style-type: none"> • A, B, C and the group D E are ordered elements, • C is optional, F has cardinality 0,N, • D E has cardinality 1,N, • Conditioned presence: D or E, • Date is a constant "20000114", • Color is an optional list of values 	<pre> XML Schema <element type="Book" > <element type="#title" occurs="OPTIONAL" /> <element type="#author" occurs="ONEORMORE" /> </element type> XML Document generated <Book> <author>Henry Ford</author> <author>Samuel Crowther</author> <title>My Life and Work</title> </Book> </pre>

XSL (*eXtensible Stylesheet Language*): linguagem standard W3C usada para construir stylesheets que são aplicadas aos elementos de um documento XML

Domínios de utilização de XML

- VML (Vector Markup Language): standard para representar vetores proposta pela Microsoft para o domínio das aplicações gráficas
- PGML (Precision Graphics Markup Language): standard para representar vetores proposta pela Adobe para o domínio das aplicações gráficas
- SMIL (Synchronized Multimedia Integration Language): standard para representar elementos multimédia numa página Web, usada no campo da multimédia
- MathML (Mathematical formula Markup Language), usada no domínio das ciências
- CML (Chemical Markup Language), usada no domínio das ciências
- AML (Astronomical Markup Language), usada no domínio das ciências

Níveis de integração de serviços



Transformação sintática

- Objetivo: modificar a representação de um evento para que seja utilizável pela aplicação responsável pelo seu processamento
- Pode dizer respeito a:
 - Formato global do evento
 - Representação de cada elemento de informação que compõe o evento
- Pode gerar modificações
 - na ordem em que a informação está representada dentro do evento
 - no modo de representação de um elemento de informação (alfabética, alfanumérico, binário...)

Transformação sintática: exemplo

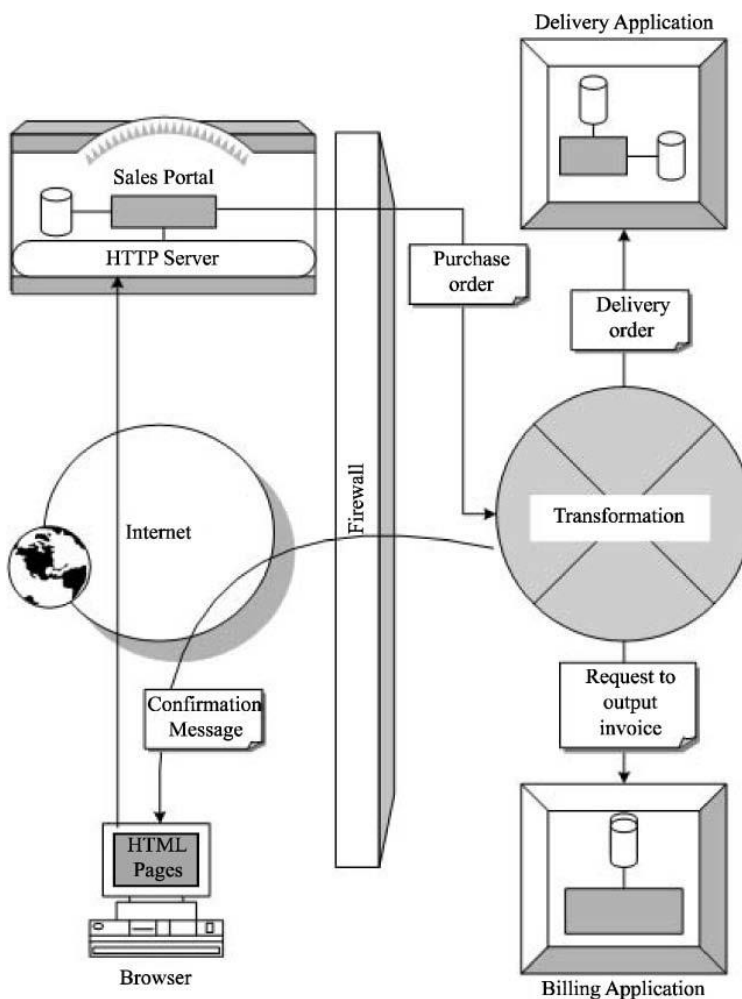
- Encomenda feita por um cliente numa loja online.
- A encomenda envolve informação expressa em XML.
- Antes de a ordem poder ser processada por uma aplicação antiga escrita em COBOL, deve ser transformada para que seja utilizável pela aplicação.
- A transformação não irá modificar o significado do evento, mas simplesmente formulá-la com uma sintaxe aceitável pela aplicação que o processa

XML Order	
<pre> <Order> <CurrencyCode>EUR</CurrencyCode> <PaymentMode>transfer</PaymentMode> <OrderDate>20000310</OrderDate> <CustomerReference> <Name>Mr BEAN</Name> <CustomerCode>007</CustomerCode> <Company>Editions HERMES</Company> <Address> <Address1>Editions HERMES</Address1> <Address2>8, quai du Marché-Neuf</Address2> <PostalCode>75004</PostalCode> <City>Paris</City> <Country>FRANCE</Country> </Address> </CustomerReference> <OrderLine> <Quantity>25</Quantity> <ProductDescription>PC, Pentium III 500 MHz, 8 processors, RAM 256 Mb</ProductDescription> </OrderLine> </Order> </pre>	
Order for application written in COBOL	
<pre>10.....20.....30.....40.....50.....60 EURtransfer 20000310Mr BEAN 007HERMES70.....80.....90.....100.....110.....120 Editions HERMES 8, quai du Marché-Neuf 75004Paris130.....140.....160.....170.....180.....190 France 25PC, Pentium III 500 MHz,8 processors,200 RAM 256 Mb </pre>	

Transformação semântica

- Objetivo: modificar o significado da totalidade ou de parte da informação para/de um evento original, para deduzir outros eventos que irão ser introduzidos no SI.
- Com o exemplo anterior, a informação da encomenda pode ser usada para gerar:
 - uma mensagem para o cliente a confirmar a encomenda
 - uma ordem de entrega para a aplicação de gestão de entregas
 - um pedido para criar uma fatura para a aplicação de faturação
- Para executar este tipo de transformações criação de eventos, são necessárias transformações mais complexas, por exemplo:
 - obter os endereços do cliente para enviar a mensagem de confirmação e calcular o preço total da compra
 - fornecer as referências dos produtos encomendados ao sistema de entregas
 - modificar o número do cliente para corresponder ao do sistema de faturação
 - fornecer a informação de faturação para o processamento da fatura
- Estas manipulações requerem
 - acesso a repositórios externos para procurar a informação necessária
 - que sejam efetuados alguns cálculos

Transformação semântica: exemplo

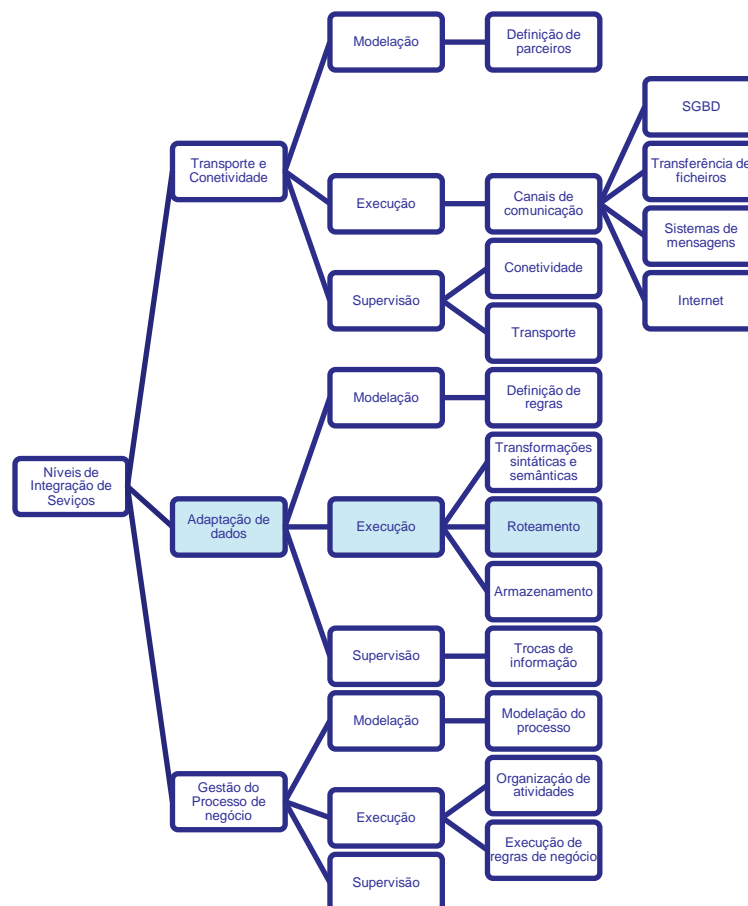


XML Order	
<pre> <Order> <CurrencyCode>EUR</CurrencyCode> <PaymentMode>transfer</PaymentMode> <OrderDate>20000310</OrderDate> <CustomerReference> <Name>Mr BEAN</Name> <CustomerCode>007</CustomerCode> <Company>Editions HERMES</Company> <Address> <Address1>Editions HERMES</Address1> <Address2>8, quai du Marché-Neuf</Address2> <PostalCode>75004</PostalCode><City>Paris</City> <Country>FRANCE</Country></Address> </CustomerReference><OrderLine><Quantity>25</Quantity> <ProductDescription>PC, Pentium III 500 MHz, 8 processors, RAM 256 Mb</ProductDescription> </OrderLine> </Order> </pre>	
Delivery request for COBOL application	
.....10.....20.....30.....40.....50.....60	
25 REF025643220000310 CLI007Mr BEAN Editions	
.....70.....80.....90.....100.....110.....120	
HERMES 8, quai du Marché-Neuf	
.....130.....140.....160.....170.....180.....190	
75004Paris France	
Billing request for COBOL application	
CLI00720000310Mr BEAN 007HERMES	
.....70.....80.....90.....100.....110.....120	
Editions HERMES 8, quai du Marché-Neuf	
.....130.....140.....160.....170.....180.....190	
75004Paris France 25REF0256432	
.....200.....210.....220.....230.....240.....250	
PC, Pentium III 500 MHz,8 processors, RAM 256 Mb	
.....260.....270.....280.....290.....300.....310	
Text format message	
<pre> beans@editions-hermes.com;;Confirmation of order;Sir ,Crwe have the pleasure of confirming your order dated CR10/03/ 2000CRforCR25 x PC, Pentium III 500 MHz,8 processors, RAM 256 MbCR for a pre-tax amount of CR EUR 200.000CRSincerely,CRJ. DupondCRCustomer Service </pre>	

Transformação semântica: exemplo - problemas

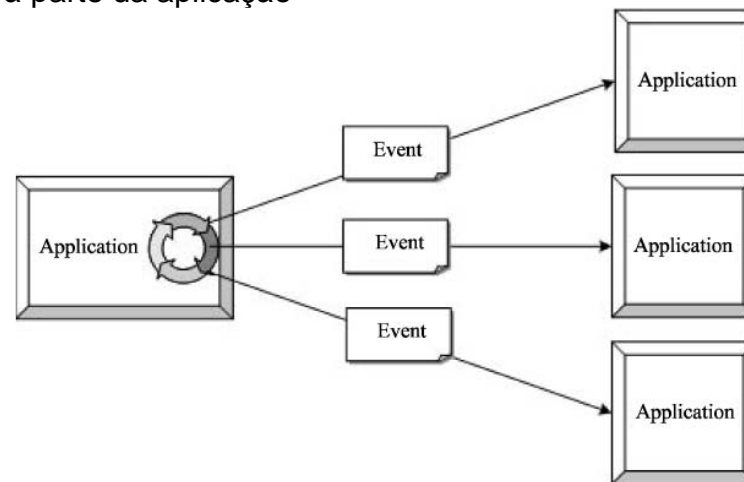
- Gerar os eventos necessários para completar o processo de negócio que trata a encomenda.
 - Coberto pelas transformações sintáticas e semânticas
 - que geram os eventos esperados quando aplicadas ao evento/input
- Determinar as aplicações recetoras destes eventos.
 - Problema de routing

Níveis de integração de serviços



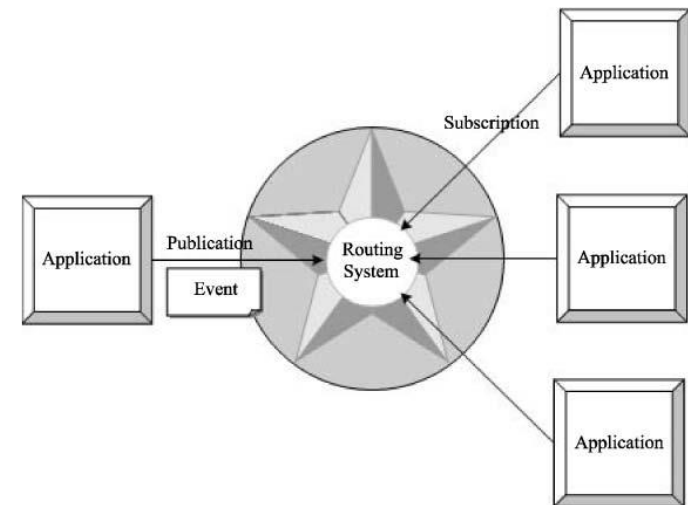
Roteamento

- Determinação do recetor dos eventos/dados gerados por uma aplicação.
 - Pode ser feito de diversas formas.
- Caso spaghetti system: a aplicação endereça diretamente os recetores da informação que produz.
 - É a aplicação que os conhece e por isso que trata do roteamento.
- Dado que um dos objetivos da solução de integração é assegurar a independência das aplicações entre si, o roteamento deve ser tratado por um sistema à parte da aplicação



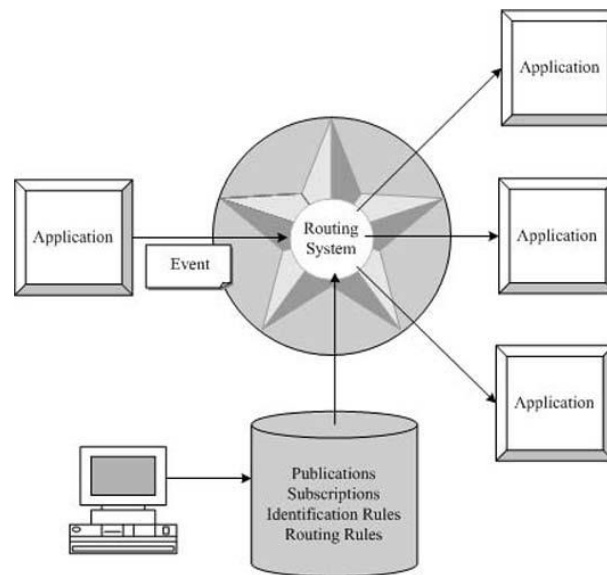
Mecanismo de publicação/subscrição

- A aplicação que cria o evento “publica-o” para que esteja disponível para as aplicações que dele necessitem.
 - Estas “subscrivem” a receção deste evento
- Aplicações emissoras indicam ao sistema de roteamento a natureza dos eventos (Ex: tema, evento, cliente, etc.)
 - Aplicações interessadas subscrivem os eventos do sistema de roteamento através da mesma informação
- Sistema de roteamento será capaz de determinar as aplicações recetoras de cada evento produzido.
 - Se uma nova aplicação pretender receber o evento, apenas tem de o subscriver, não tendo impactos no sistema existente
- Desvantagens deste sistema
 - Aplicação deve ser desenhada de base para usar o mecanismo de publicação/subscrição
 - Necessidade de partilha entre as aplicações de uma lista com a definição das denominações das publicações



Externalização das regras de roteamento

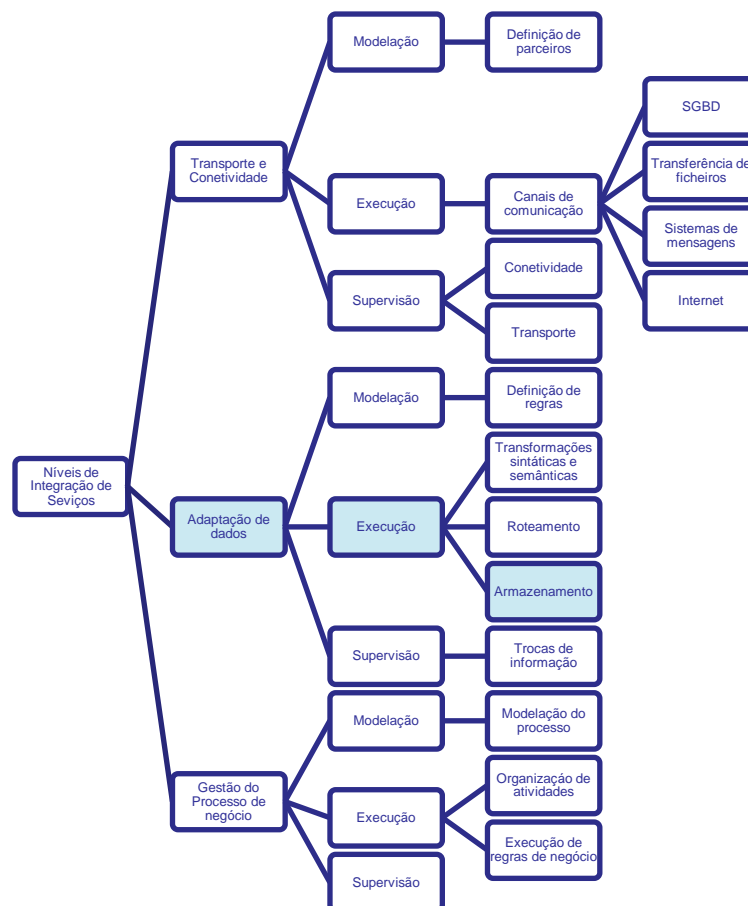
- Para contornar as desvantagens do sistema anterior
- Sistema deve permitir a implementação de operações de publicação/subscrição externas à aplicação
- Pode ser feito por funcionários responsáveis por definir as denominações e implementar as subscrições das aplicações.
 - Esta informação ficaria guardada numa BD que o sistema de roteamento iria usar.



Externalização das regras de roteamento

- O problema da publicação ainda não está resolvido.
- Processamento simples em aplicações que produzam um único tipo de evento: na BD do sistema de roteamento indicar que os eventos vindos desta aplicação devem ser publicados com as características predefinidas existentes na BD
- Processamento mais complexo em aplicações que produzam eventos de múltiplas naturezas: necessidade de definir regras de identificação que irão permitir ao sistema de roteamento usar:
 - Informação declarativa vinda da aplicação que publica o evento
 - Análise do conteúdo do evento para determinar a sua natureza
 - Pode ser necessário analisar o conteúdo para determinar se um evento tem de ser subscrito por mais do que uma aplicação
- A implementação de mecanismos de publicação/subscrição necessita de respeitar a sequência de operações:
 - Identificação do evento/input
 - Transformação
 - Identificação dos eventos gerados
 - Aplicação de subscrições e regras de roteamento baseadas no conteúdo
- Noções de publicação/subscrição, regras de identificação e regras de roteamento contribuem para a implementação de SOA

Níveis de integração de serviços



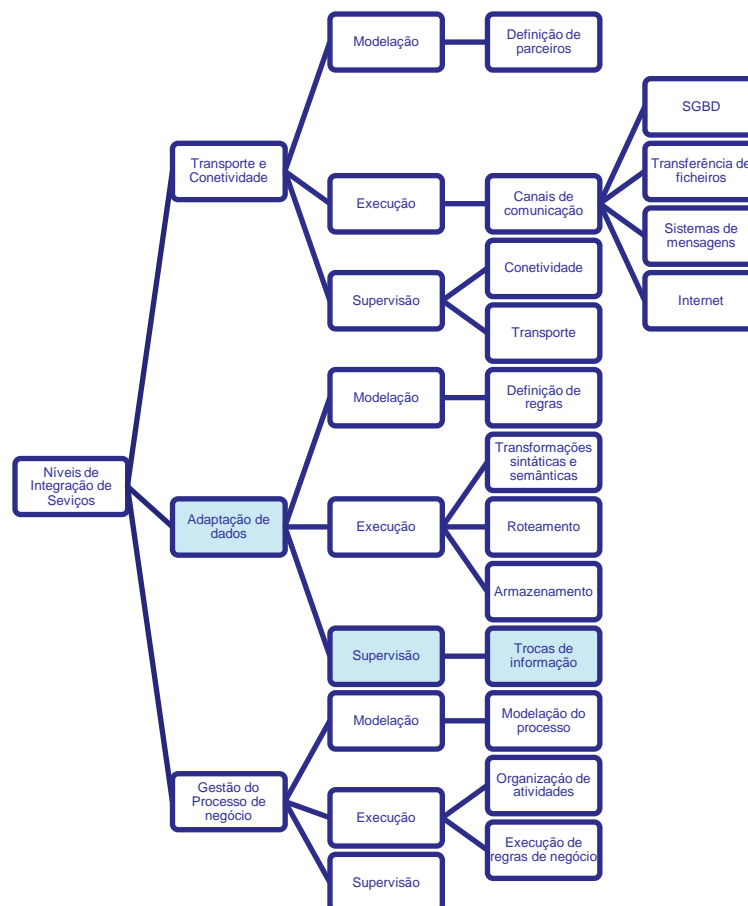
Armazenamento

- Diferentes tipos de aplicações têm diferentes ritmos de operação
- Quando os eventos transitam entre aplicações, deve ser possível adaptar a troca ao ritmo de operação de cada aplicação
- A infraestrutura de integração deve oferecer mecanismos para armazenamento temporário dos eventos para os disponibilizar às aplicações recetoras no momento certo
- Dependendo do evento, este armazenamento pode ser feito em:
 - Sistemas de gestão de ficheiros
 - BDs
 - Filas

Definição de regras

- A adaptação dos eventos à forma de operação das aplicações necessita da definição de regras para transformação, roteamento e armazenamento.
- As regras devem ser centralizadas num “dicionário” global e distribuído pela infraestrutura de integração
- O dicionário deverá ser acessível e utilizável tanto por profissionais de TI como por especialistas do negócio
- Deverá incluir:
 - Definição dos eventos que circulam no sistema
 - Definição da estrutura dos eventos
 - Identificação de regras para os eventos
 - Regras de transformação a aplicar
 - Informação sobre publicação e subscrição
 - Itinerários dos eventos antes da entrega
 - Regras de roteamento
 - Regras para armazenamento temporário

Níveis de integração de serviços



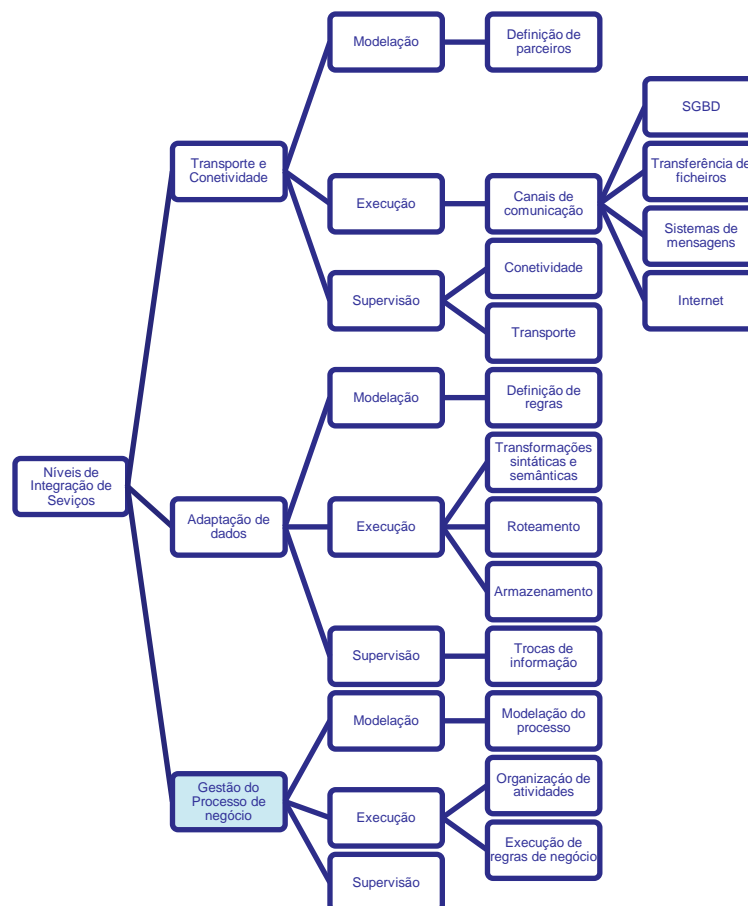
Supervisão das trocas de informação

- A integração entre aplicações implementa um conjunto de funções que em certos casos podem ser bastante complexas:
 - Adaptadores para capturar e restaurar informação
 - Regras de transporte e outras regras associadas
 - Transformação
 - Roteamento
 - Armazenamento
- Deve ser possível fazer *tracking* do processo de integração (completo) e mostrar o estado de um evento a qualquer momento
- Ferramentas de transporte, adaptadores, componentes de transformação e componentes de roteamento devem publicar informação sobre o progresso das operações de roteamento.

Supervisão das trocas de informação

- Supervisão providencia uma *overview* das trocas e dos seus estados, com a possibilidade de detalhar passos específicos.
- Sistema global de supervisão deve mostrar os indicadores para avaliar a atividade global e informação e alertas dos processos de transformação e de roteamento.
- Alertas:
 - Detecção de erros em diferentes passos do processamento
 - Deduzidos de regras de verificação
- Se um evento é rejeitado pelas funções de transformação ou roteamento, deve haver uma ferramenta que permita corrigir manualmente os eventos errados e voltar a inseri-los no circuito sem ter de reexecutar a aplicação original

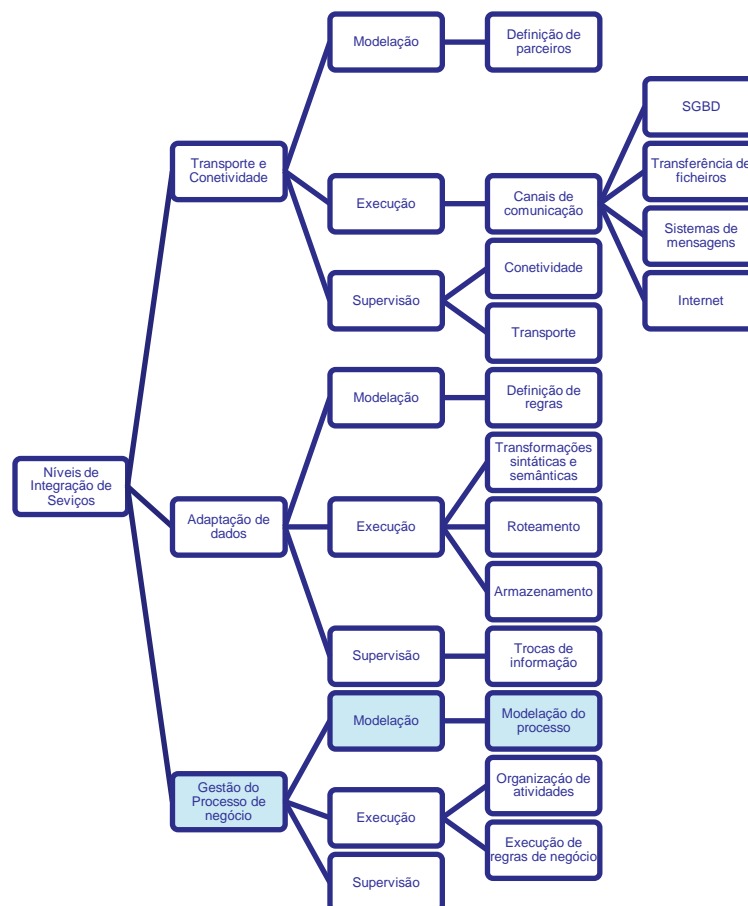
Níveis de integração de serviços



Automatização do processo de negócio

- Funções de transporte, conectividade e adaptação possibilitam o tratamento de problemas relacionados com a transmissão de dados na integração de aplicações
- Processamento de casos correspondentes a integração de processos *multi-step* requer funções complementares de automatização do processo de negócio
- Este é o domínio do BPM (*Business process management*)

Níveis de integração de serviços



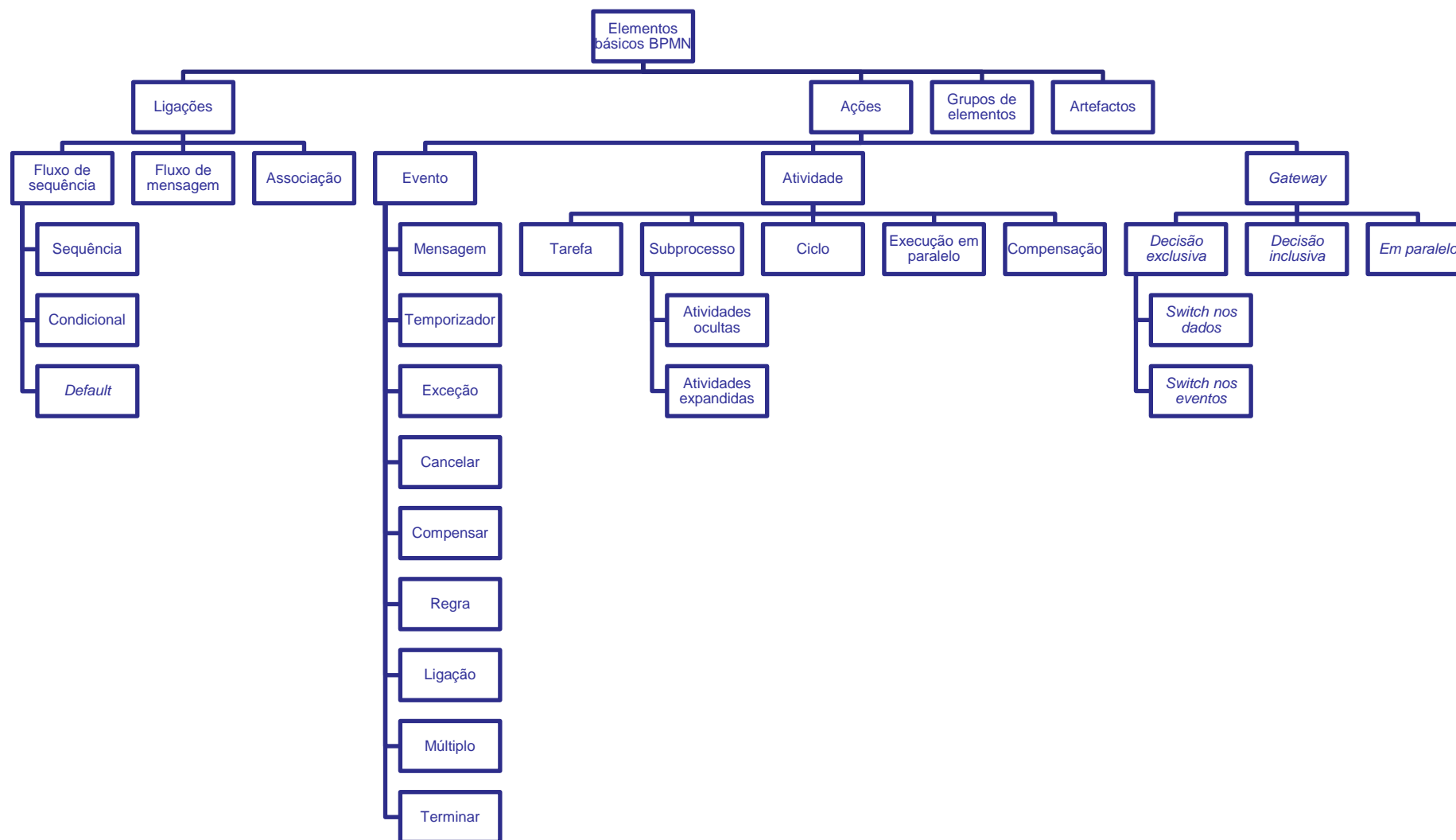
Modelação do processo de negócio

- Objetivos: documentar, clarificar, melhorar o processo de negócio
- Ferramentas: exemplo: bpmn.io <https://demo.bpmn.io/>
- A modelação do processo de negócio deve representar o processo para que possa ser implementado automaticamente.
- Define:
 - conjunto de atividades
 - relações entre elas
 - critérios que definem o início e fim de um processo
 - toda a informação relativa a cada atividade (manual ou automática):
 - participantes (pessoas)
 - aplicações
 - dados
 - etc

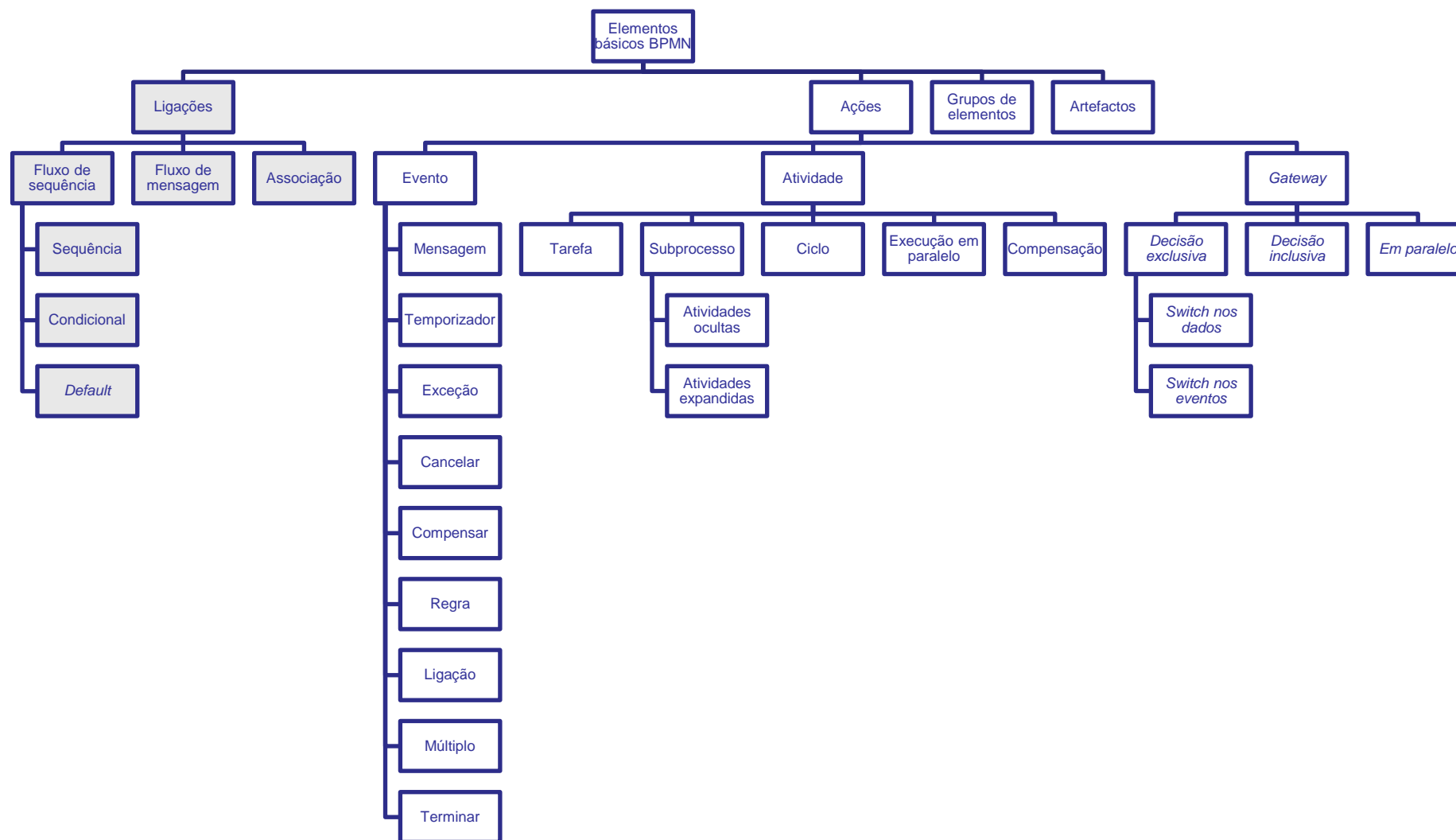
BPMN

- BPMN (*Business process management notation*): notação gráfica standard desenhada para descrever processos de negócio sob a forma de diagramas.
- Composto por 4 categorias de elementos básicos:
 - ligações
 - ações
 - grupos de elementos
 - artefactos






Categorias de elementos BPMN



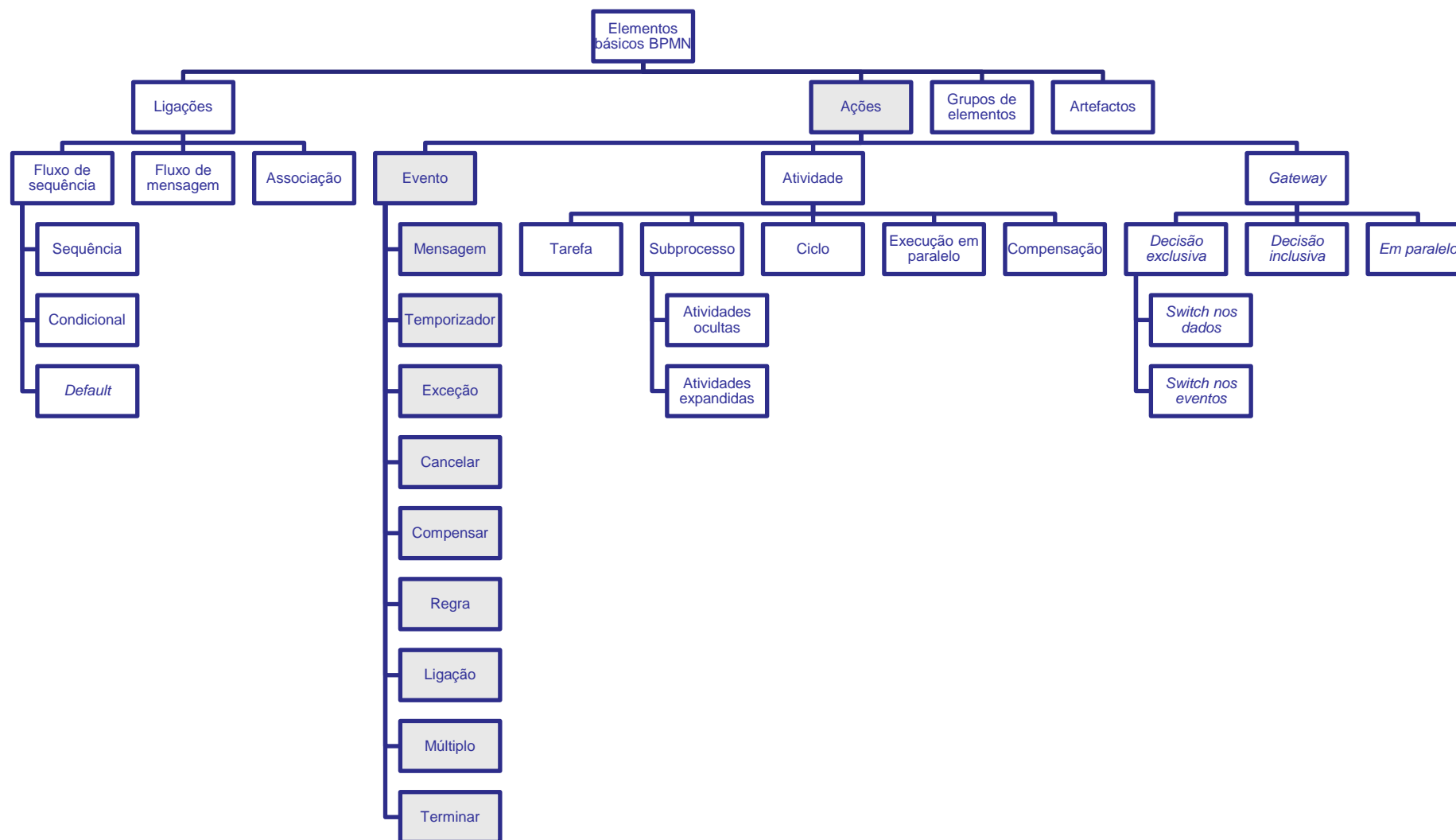
Categorias de elementos BPMN












Ligações

Tipo de ligação	Subtipo/Descrição	Representação
Fluxo de sequência	Execução de sequência: especificar por que ordem as atividades são executadas	
	Execução condicional: uma expressão condicional vai ser avaliada durante a execução para determinar se a sequência é ou não usada	
	Execução default: sequência que irá ser executada caso todas as outras condições não se verifiquem	
Fluxo de mensagem	Representa o fluxo de mensagens entre duas entidades em grupos de elementos diferentes	
Associação	Ligar um elemento de informação (texto/objetos gráficos, não representa ações) com ações	

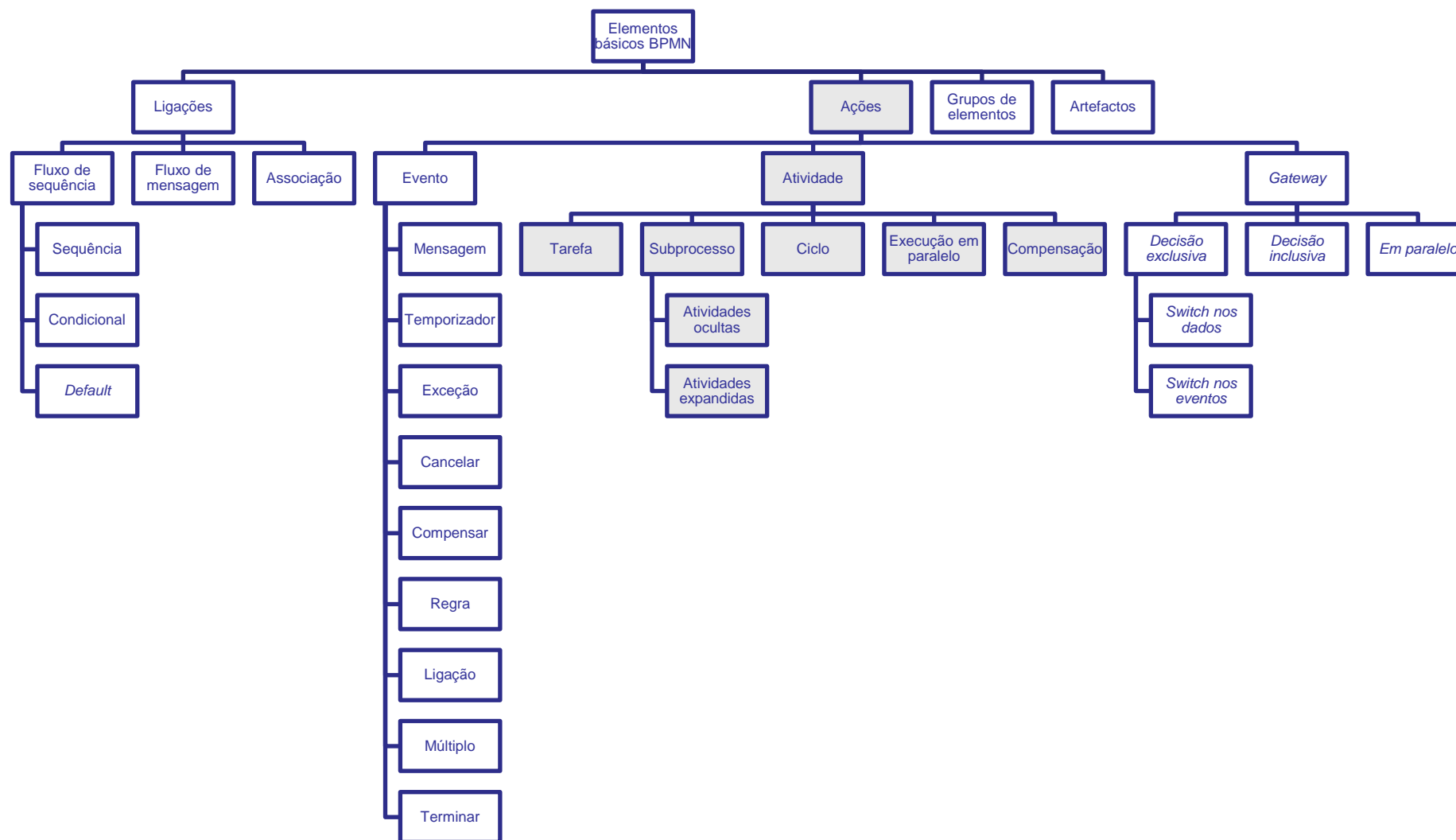
Categorias de elementos BPMN



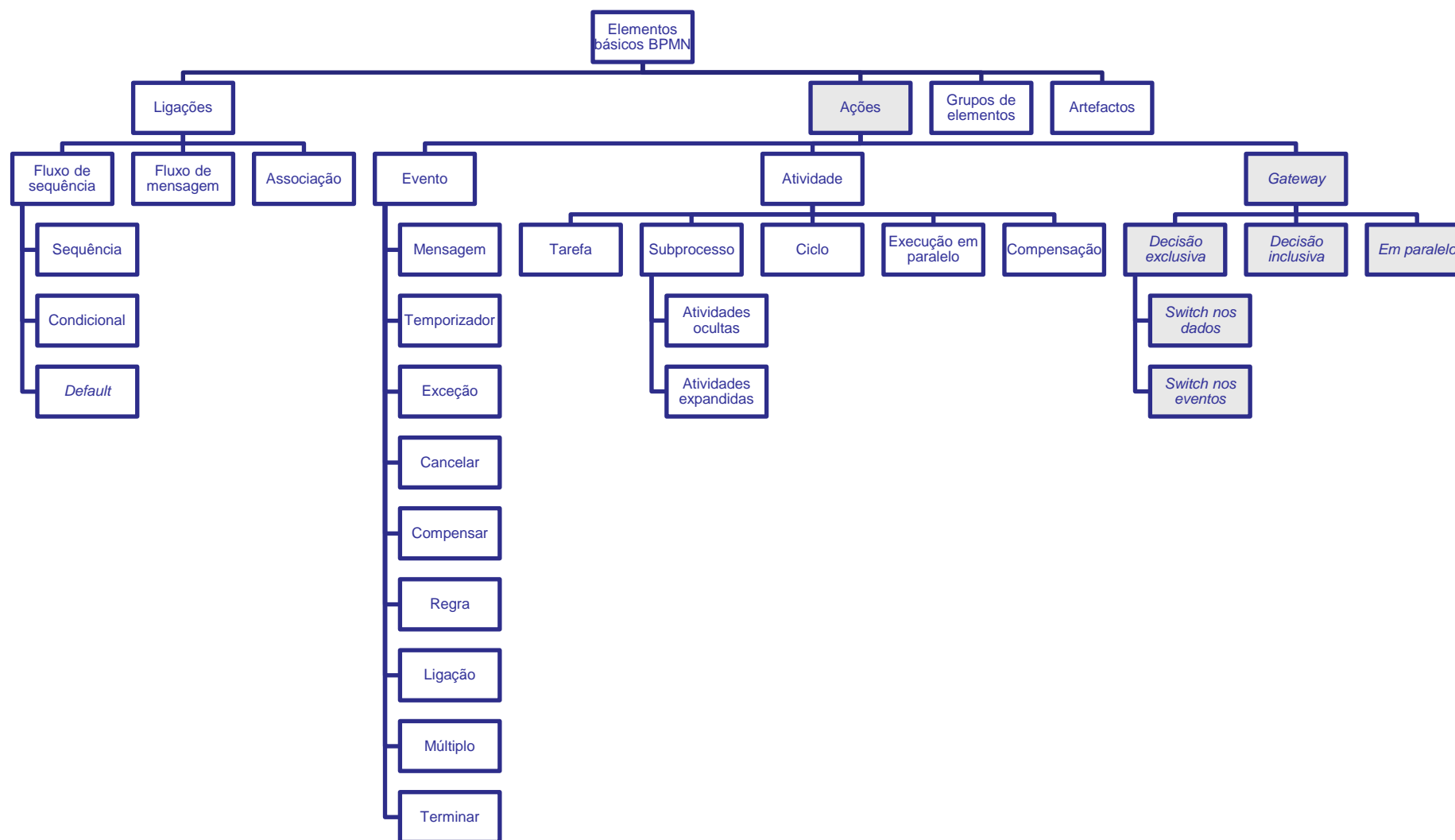
Ações: Eventos

Tipo de ação	Subtipo/Descrição	Representação
<p>Evento: algo que ocorre durante a execução de um processo de negócio. Tem impacto na progressão da execução e pode surgir ao início do processo (linha simples) e desencadeá-lo, durante o processo (linha dupla) para desencadear uma atividade, ou ao fim (linha escura) para o terminar.</p> <div style="display: flex; align-items: center; gap: 10px;"> <div style="border: 1px solid black; width: 20px; height: 20px; border-radius: 50%;"></div> <div style="border: 2px solid black; width: 20px; height: 20px; border-radius: 50%;"></div> <div style="border: 3px solid black; width: 20px; height: 20px; border-radius: 50%;"></div> </div>	Mensagem: uma mensagem chega de um participante	
	Temporizador: expirado	
	Exceção: ocorreu um erro durante a execução de uma ação ou processo	
	Cancelar: cancela a transação em progresso	
	Compensação: voltar à situação original em caso de cancelamento	
	Regra: despoleta um evento se as condições da regra forem verificadas	
	Ligação: conexão entre subprocessos	
	Múltiplo: vários tipos de eventos podem ocorrer	
	Terminar: termina imediatamente todas as atividades do processo	





Categorias de elementos BPMN



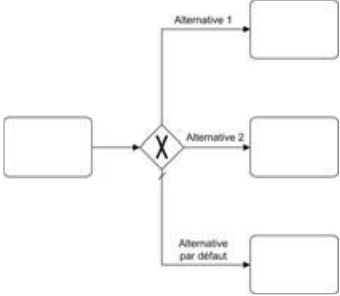
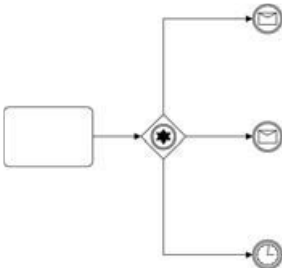
Categorias de elementos BPMN



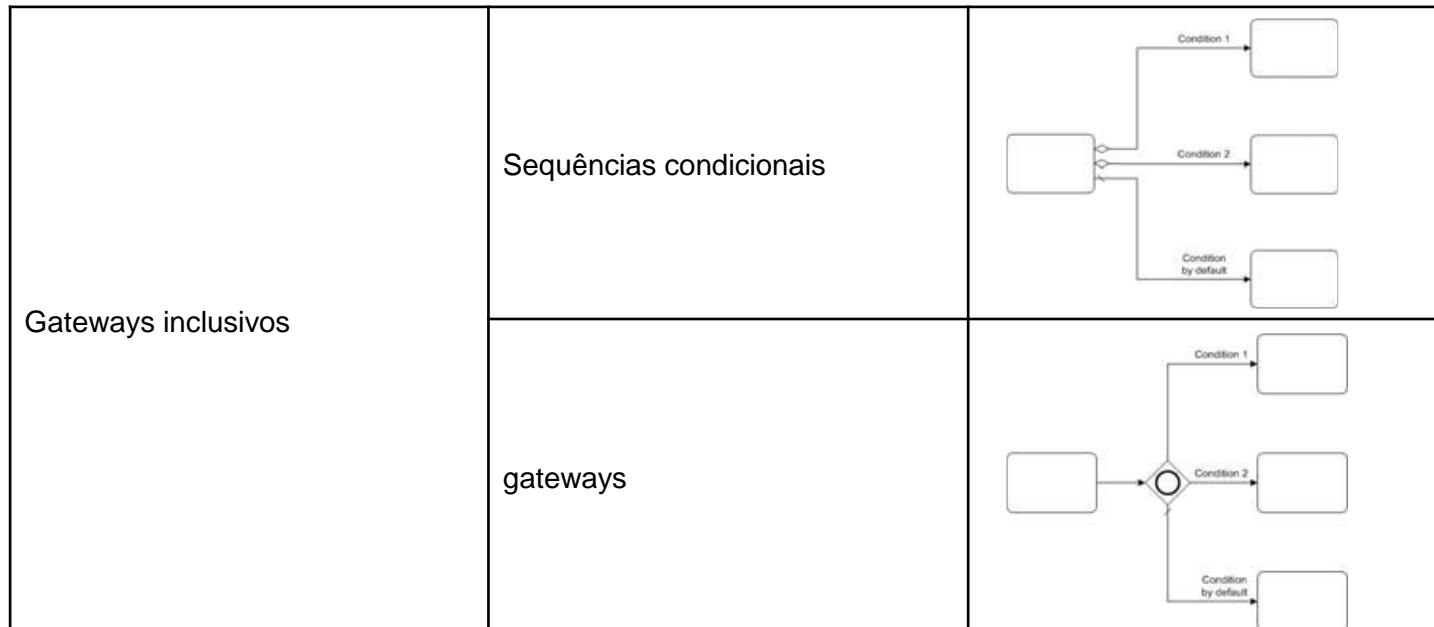
Ações: gateways

Tipo de ação	Subtipo/Descrição	Representação
Gateways: definem o modo como a execução de um processo pode convergir, divergir ou executar em paralelo	Decisão exclusiva (XOR): apenas uma sequência pode ser executada (switch nos dados)	
	Decisão exclusiva (XOR): apenas uma sequência pode ser executada (switch nos eventos)	
	Decisão inclusiva (OR): múltiplas sequências podem ser executadas	
	Em paralelo (fork/join): diversas sequências são executadas em paralelo, ou indicação de um join entre múltiplas sequências	

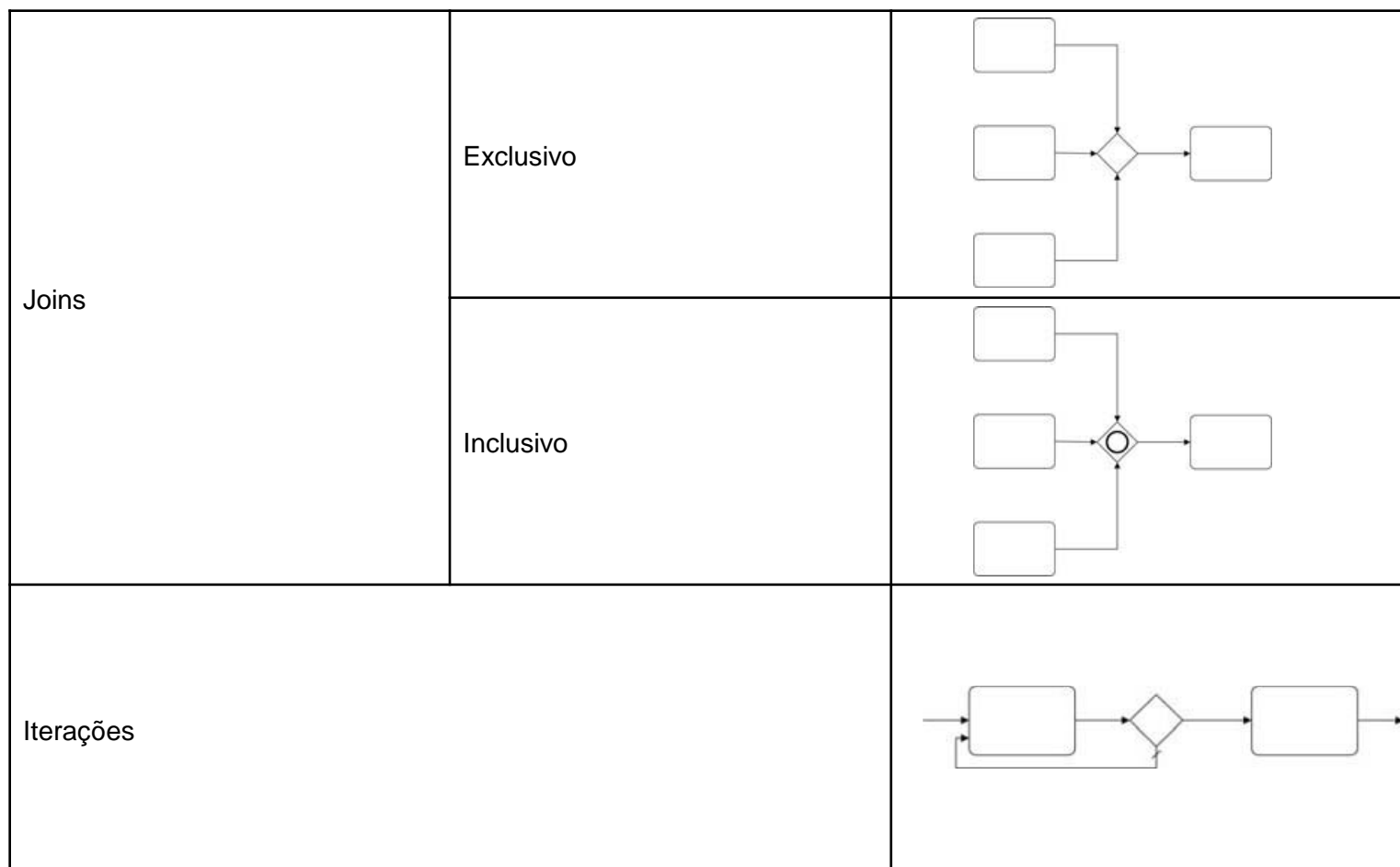
Exemplos de gateways: gateways exclusivos

Gateways exclusivos em dados	Sem marcador	
	Com marcador	
Gateways exclusivos em eventos		

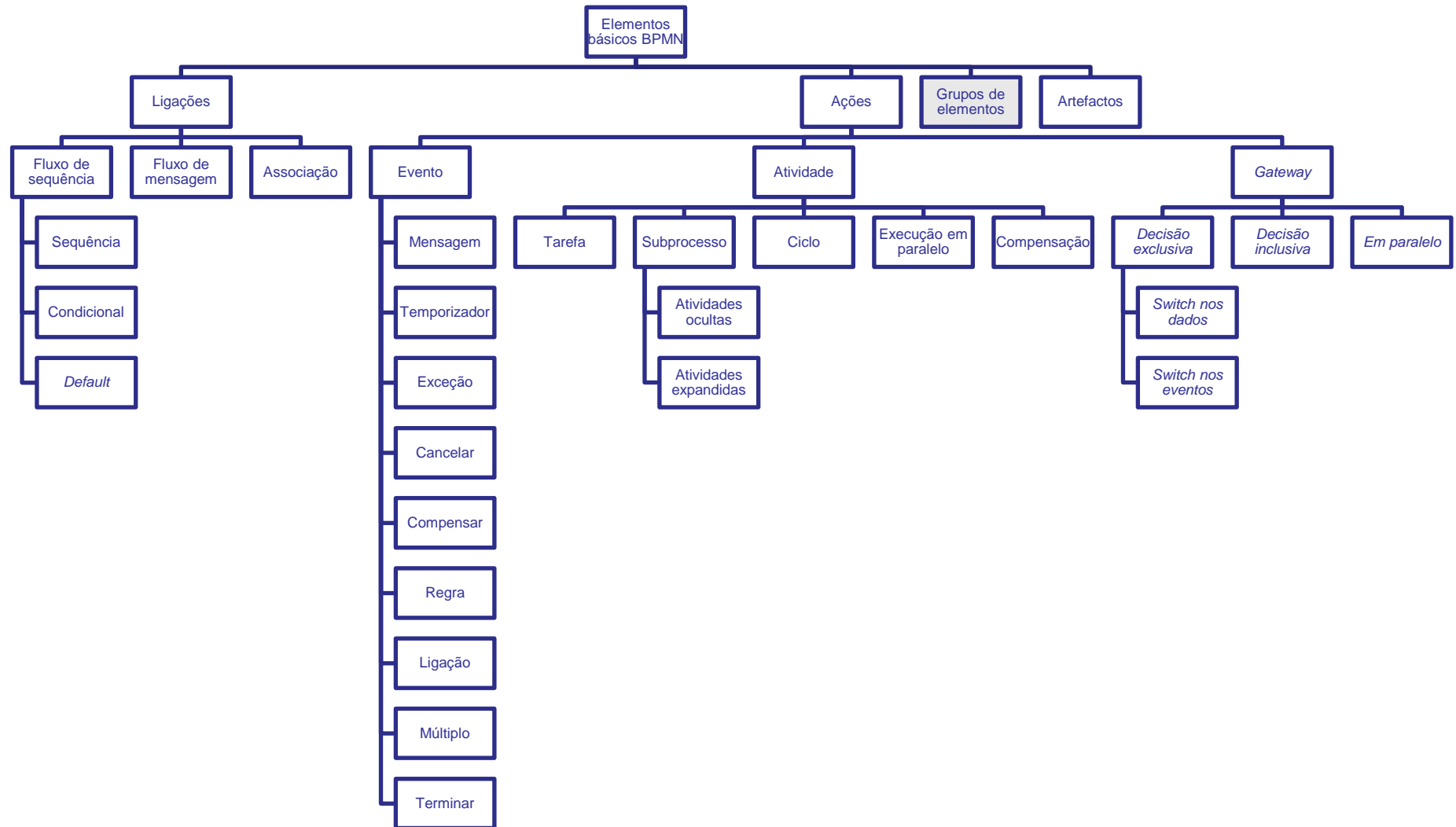
Exemplos de gateways: gateways inclusivos



Exemplos de gateways: joins e iterações



Categorias de elementos BPMN



Grupos de elementos

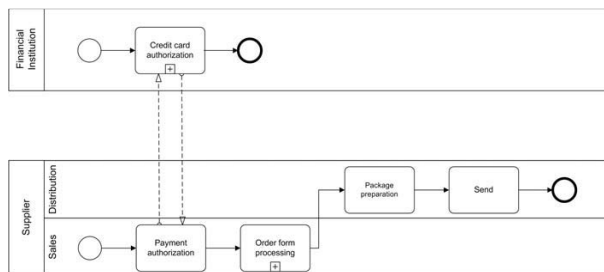
- Usados para agregar ações levadas a cabo por cada participante no processo
- Representado por Pool



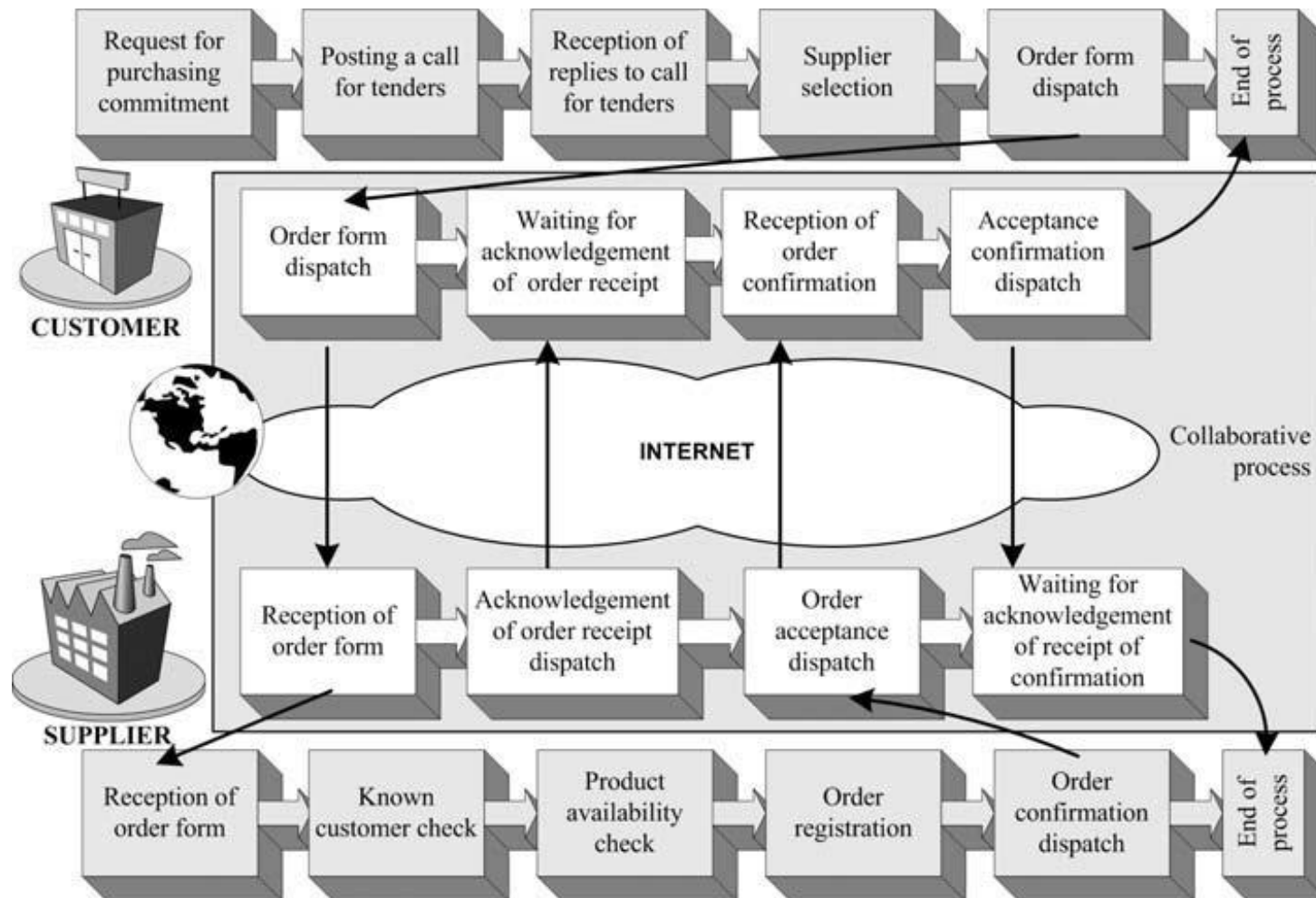
- Com swimlanes
 - subconjunto de uma Pool, que permite organizar e categorizar as atividades, representando papéis internos, sistemas, ou entidades organizacionais



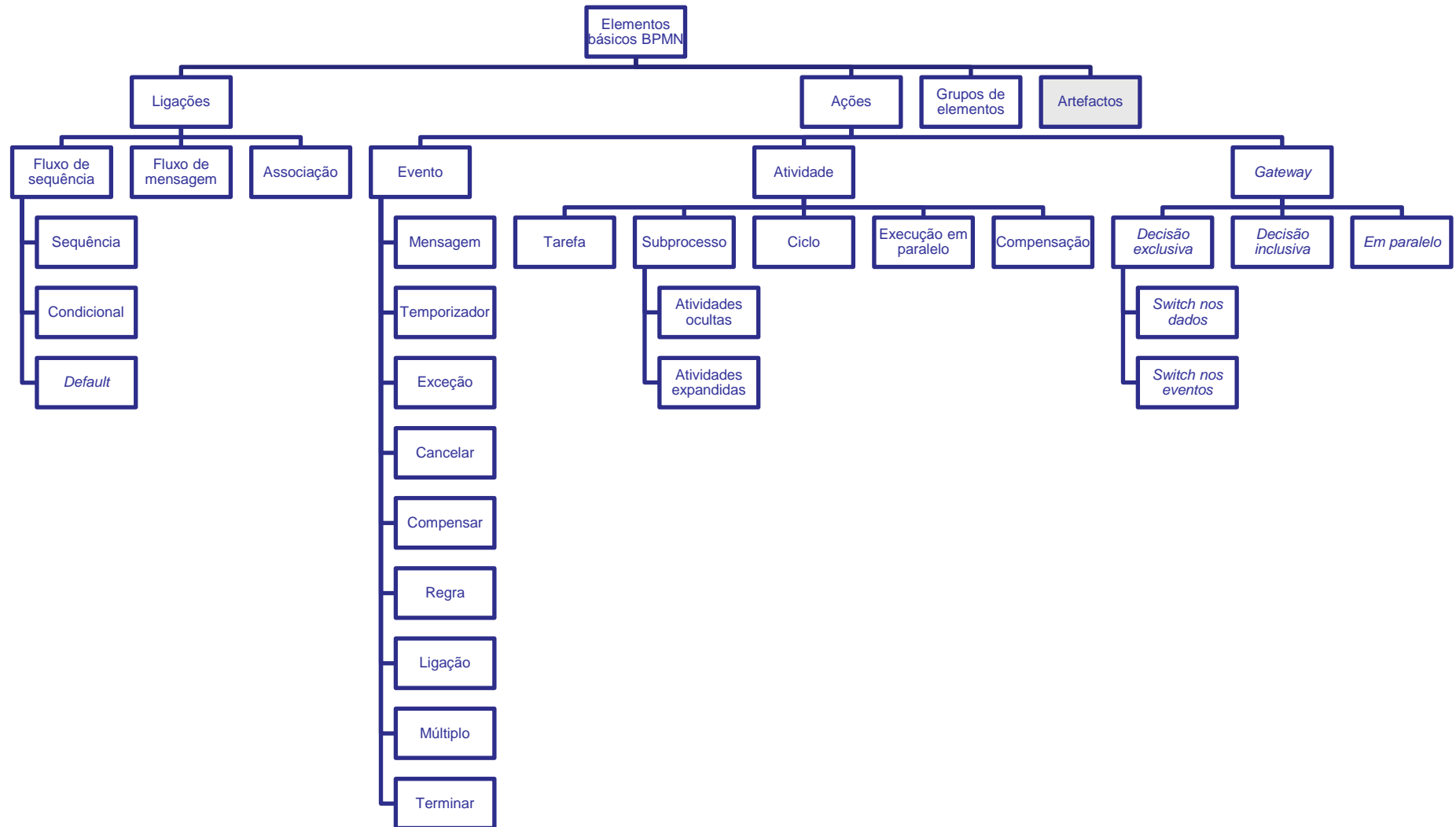
- Processo privado e colaborativo:



Exemplo: processo de tratamento de uma encomenda cliente/fornecedor

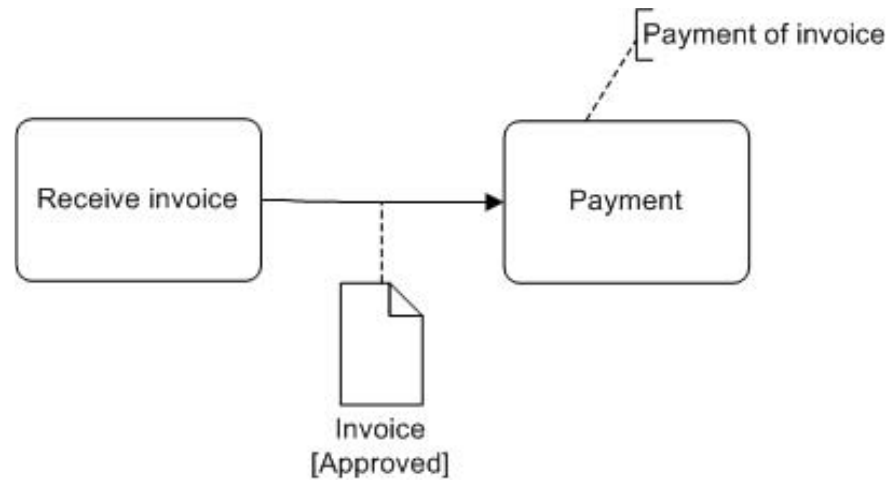


Categorias de elementos BPMN

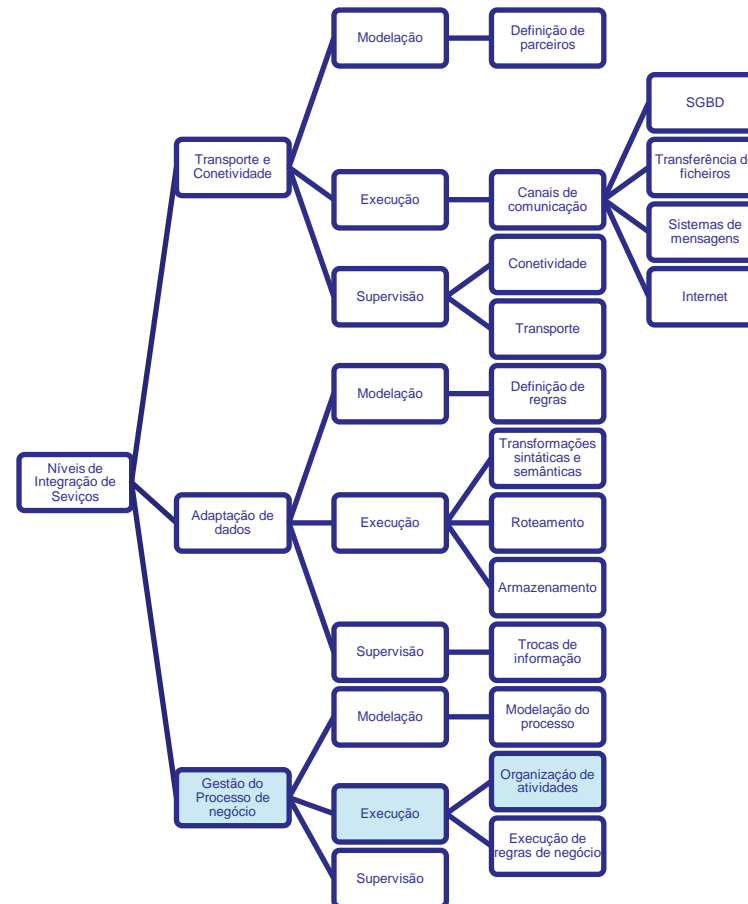


Artefactos

- Representação de informação complementar, não diretamente ligada à ação ou cadeia de mensagens.
- 2 tipos:
 - Dados
 - Anotações



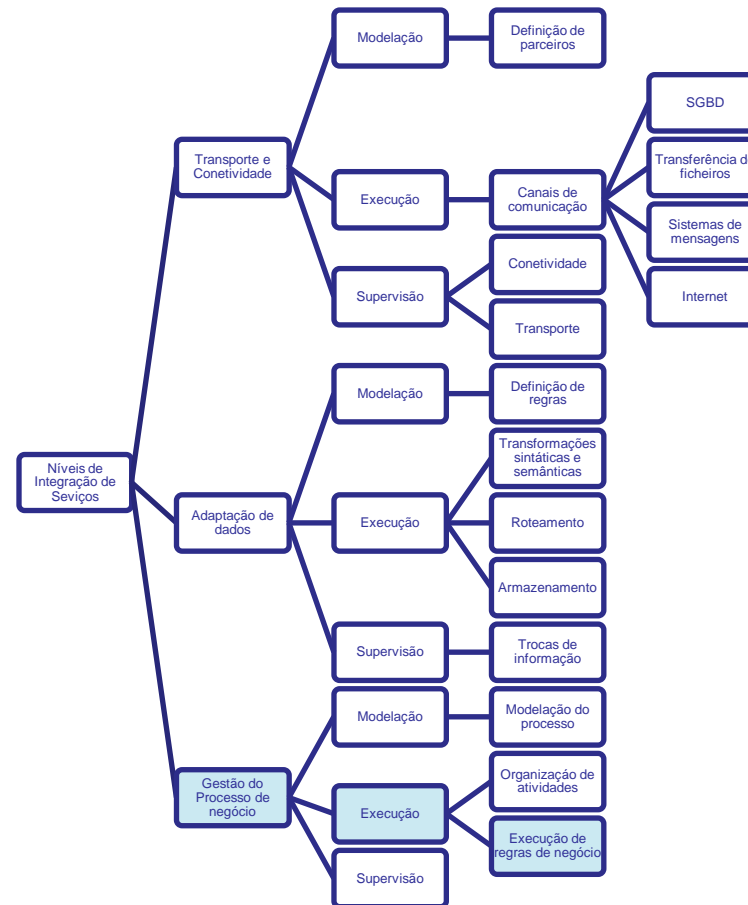
Níveis de integração de serviços



Organização de atividades

- Depois de complete a fase de modelação, é necessário:
 - lançar cada atividade na ordem definida
 - recuperar os resultados de cada fase
 - aplicar regras de encadeamento conforme os resultados observados
 - assegurar que as sucessivas atividades são alimentadas com os dados ou eventos que requerem
 - publicar informação para o sistema de supervisão
 - gerir cada passo do processo, sem omissões
- Como parte de um projeto de integração, todas estas ações devem ser levadas a cabo em plataformas distribuídas e heterogêneas, criando um problema que na maior parte das vezes é difícil de resolver
- Estas funções são executadas por um motor de execução de processo, que automaticamente executa o processo de negócio definido durante a modelação. Para isso, implementa os objetos do modelo por:
 - Instâncias de processo, cada uma compreendendo uma ocorrência do processo de negócio definido
 - Instâncias de atividades, para cada ocorrência da atividade no processo, uma das seguintes:
 - Unidade de trabalho, para cada ocorrência de atividades manuais
 - Instância de aplicação, para cada ocorrência de atividade automática

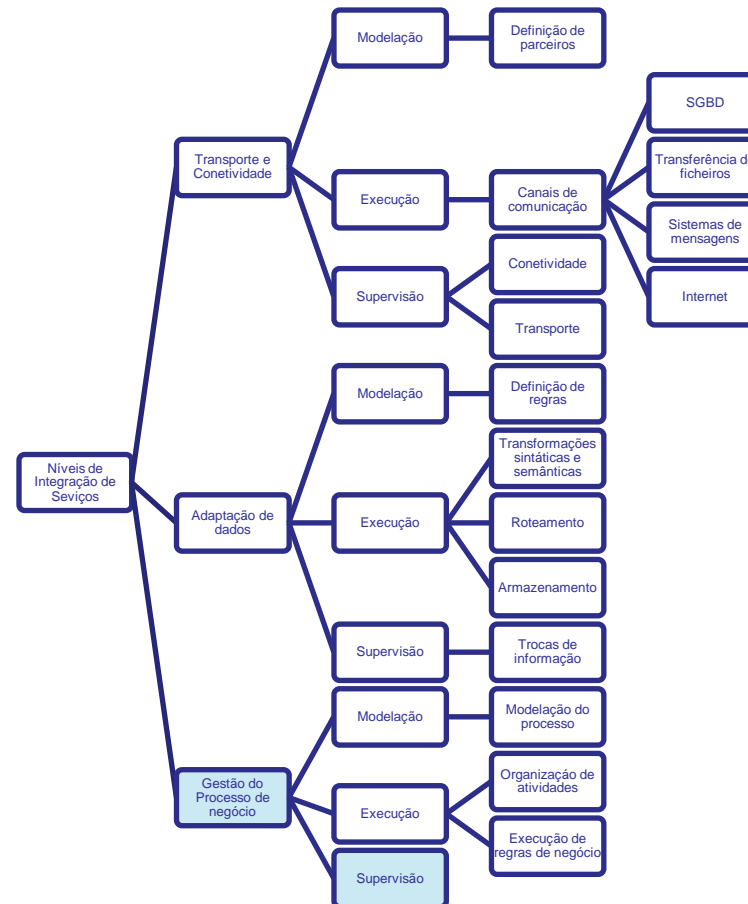
Níveis de integração de serviços



Execução de regras de negócio

- Para além do motor de processos, é necessário um motor para as regras de negócio:
 - (BRE, *Business Rules Engine*) para descrever e executar as regras que irão guiar o processo através dos vários caminhos possíveis no modelo.
- Este é um elemento fundamental na automatização de processos
 - Permite que as regras de programação das aplicações estejam guardadas externamente
 - Evita a modificação da aplicação em caso de alteração das regras de negócio
 - Confere à gestão de negócio a responsabilidade de desenvolver e refinar regras
 - Aliviando a carga de trabalho das equipas de desenvolvimento.
- A definição destas regras pode ser feita diretamente durante a modelação, usando por exemplo Java na descrição do processo, ou um motor externo.
- Evita modificação da modelação do processo no caso de se ter de mudar as regras.
 - Só é preciso modificar e republicar a regra alterada, sem impacto na lógica do processo.
- A utilização destes motores possibilita a gestão de regras elaboradas

Níveis de integração de serviços



Supervisão do processo de negócio

- Assegurar a sequência correta das várias aplicações/tarefas que compreendem o processo.
- Como a execução do processo anda entre os níveis de adaptação e de transporte, deve providenciar uma *overview* destas operações
- BPM associado a BAM (*Business Activity Monitoring*)
- BAM dá aos utilizadores indicadores em tempo real para avaliar a eficiência dos processos, e tem em conta o contexto para alertar os utilizadores em caso de aticidade anormal.
- O sistema BAM tem inputs constantes, em particular pelo motor de execução do processo, e oferece aos utilizadores informação sobre o que está a acontecer globalmente

Processo de negócio e integração: mediação e trocas

- Os papéis para os diferentes níveis de serviço e as interações para cada serviço devem ser definidos
- São frequentemente confundidos
- É importante clarificação

Nível de processo de negócio e nível de integração

Nível de integração

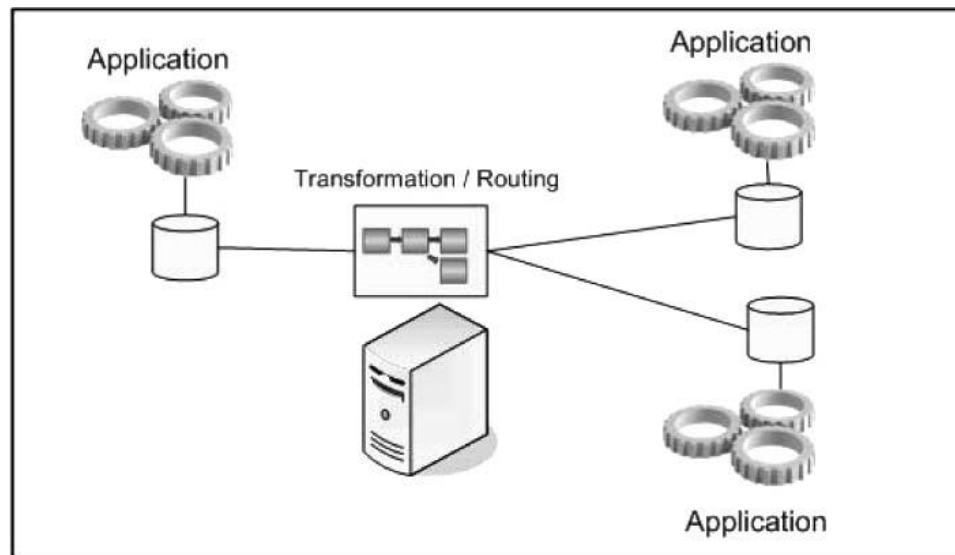
- Tarefas de propagar de dados e assegurar a sua consistência
- Requerem funções para transporte e adaptação de dados (transformação/roteamento)

Nível de processo de negócio:

- Gestão de processos *multi-step* que necessitam de coordenação de atividades de negócio com o nível de serviço oferecido pelo BPM.
- Divide-se em 2 subníveis:
 - Processo de negócio privado
 - Processo colaborativo
- Várias relações e interações existem entre estes subníveis.

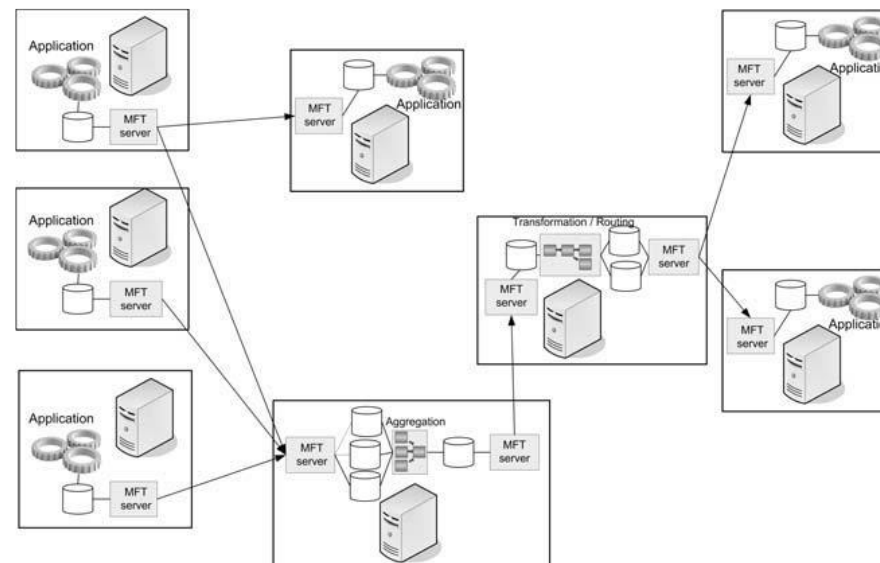
Subnível de mediação de processo

- O nível de integração compreende diferentes serviços de infraestrutura que fazem parte da camada de transporte e conectividade (como gestão de transferência de ficheiros e gestão de filas), bem como serviços na camada de adaptação
- Cada grupo de serviços é executado por um processo de mediação no nível de integração

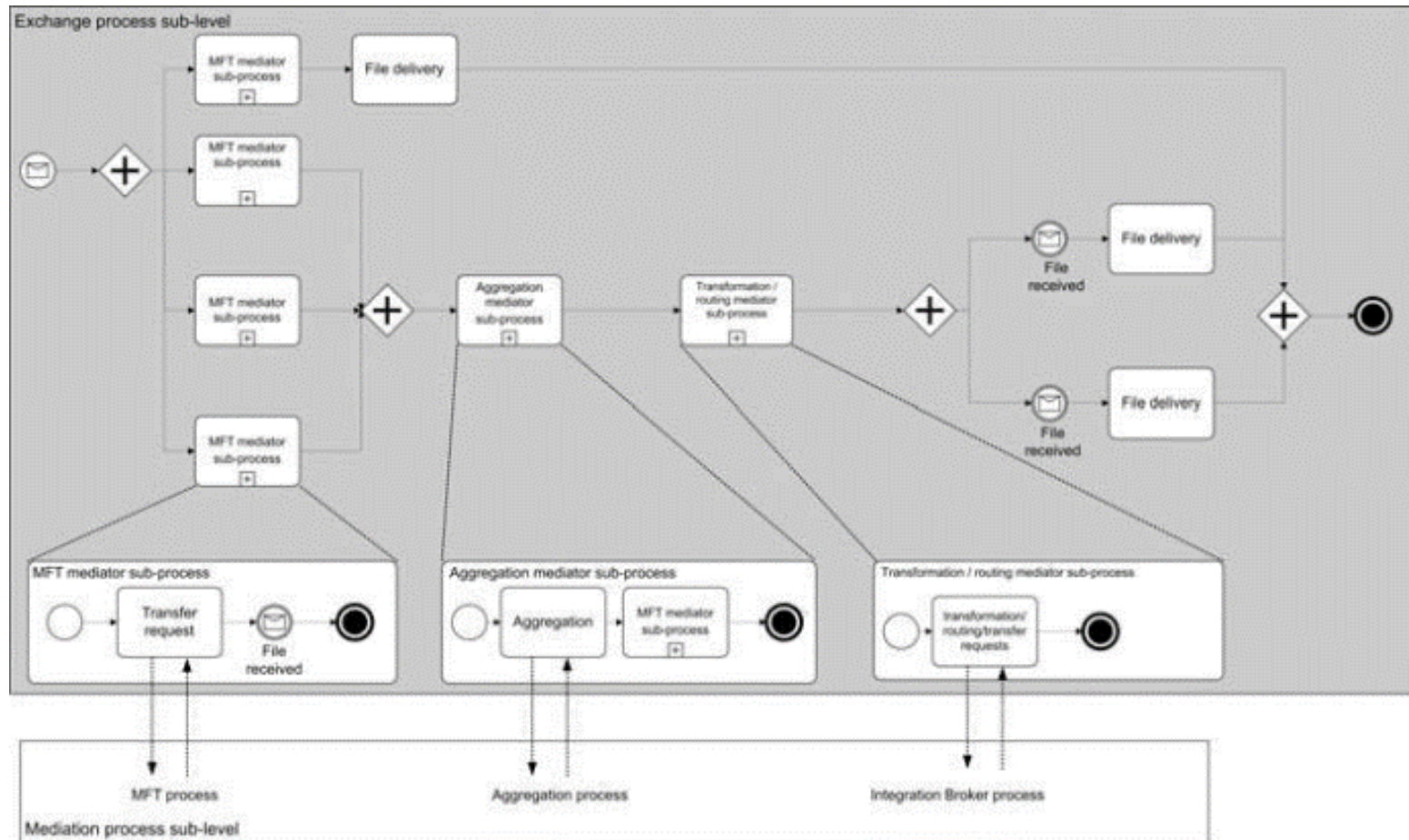


Subnível do processo de troca

- Cada grupo de serviços pode ser usado separadamente
- A maior parte das vezes num ambiente distribuído, múltiplos serviços distribuídos em diferentes plataformas são encadeados em sequência.
- O subnível do processo de troca controla a execução destes serviços

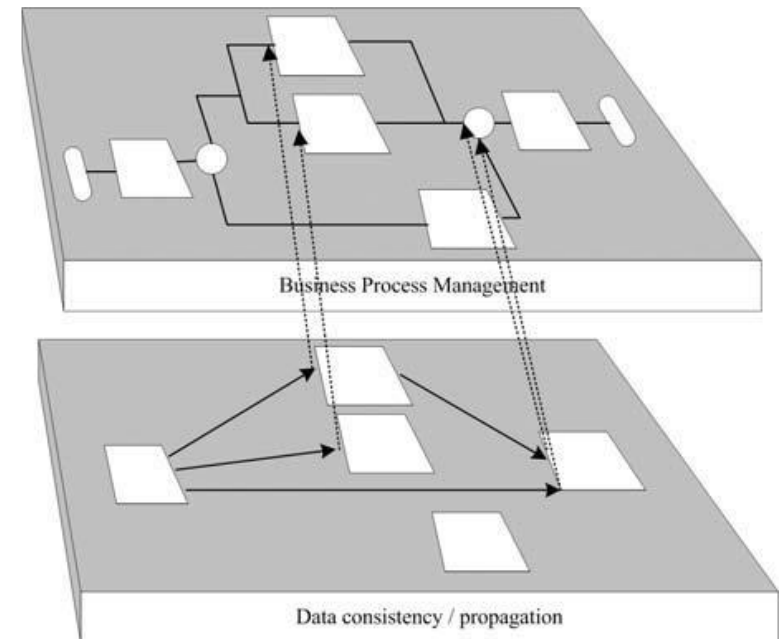


Interação entre subníveis

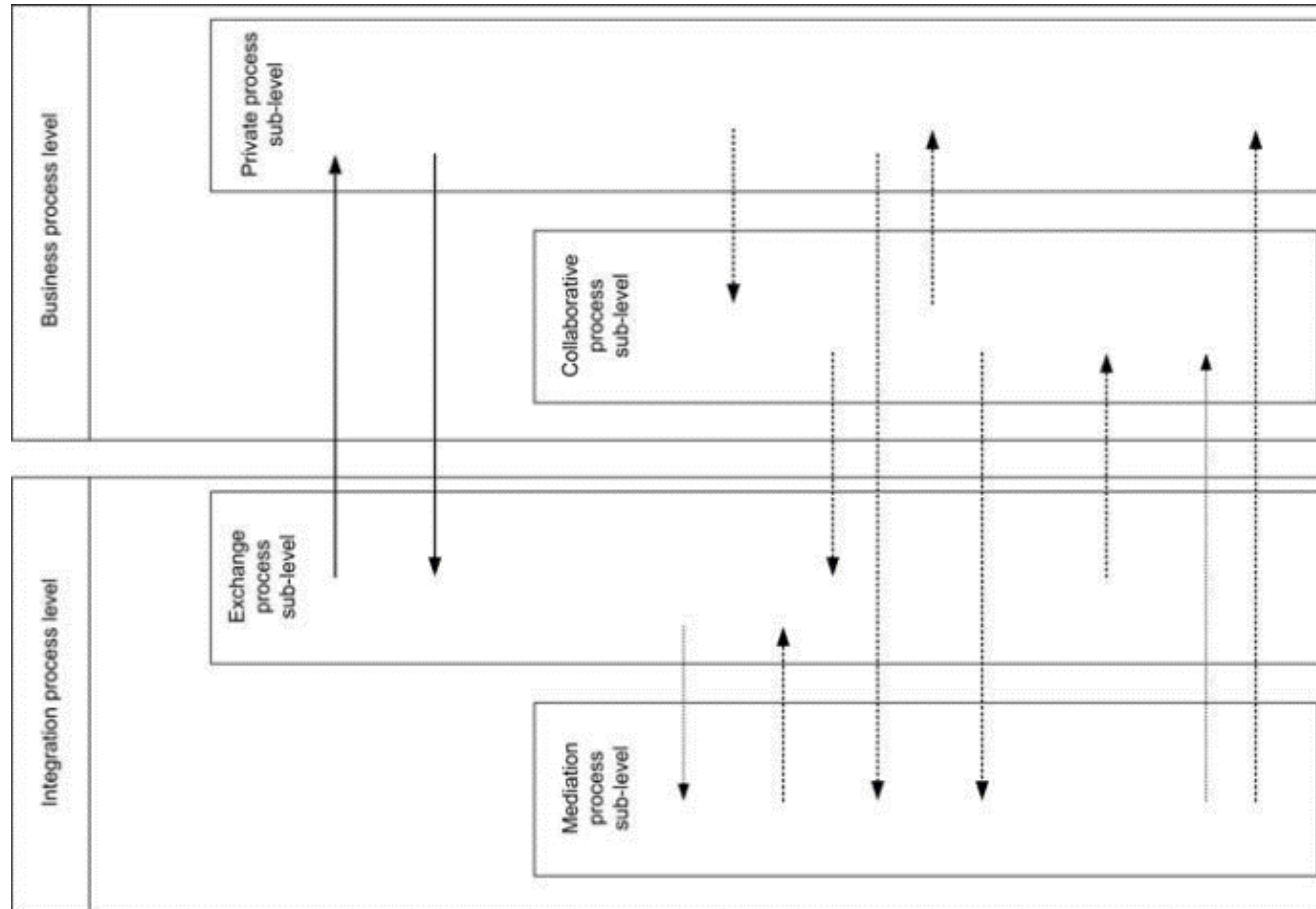


Interação entre processo de integração e processo de negócio

- Por si só, os serviços BPM não cobrem a propagação de dados
 - o que pode ser necessário para tornar os dados disponíveis para uma atividade no processo de negócio
- Haverá um certo número de relações e interações entre o nível de processo e o nível de integração:
 - A mesma propagação de dados pode introduzir dados que pertencem a diferentes processos de negócio
 - Um processo de negócio pode necessitar de vários processos de propagação para input das diferentes atividades que o compõem.
 - O processo de negócio deve então coordenar as diferentes atividades necessárias para levar a cabo essas propagações
 - Os níveis de processo e de integração não andam necessariamente ao mesmo ritmo



Relações e interações entre níveis e subníveis



Como escolher a arquitetura de trocas

- Após determinadas as fronteiras do projeto:
 - Domínios alvo de integração
 - Tipo de aplicações a integrar
 - Cartografia das trocas
- Podemos olhar para novas questões:
 - **Que tipo de comunicação usar entre as aplicações a integrar?**
 - **Qual a arquitetura de comunicação a implementar?**

Comunicação síncrona/assíncrona

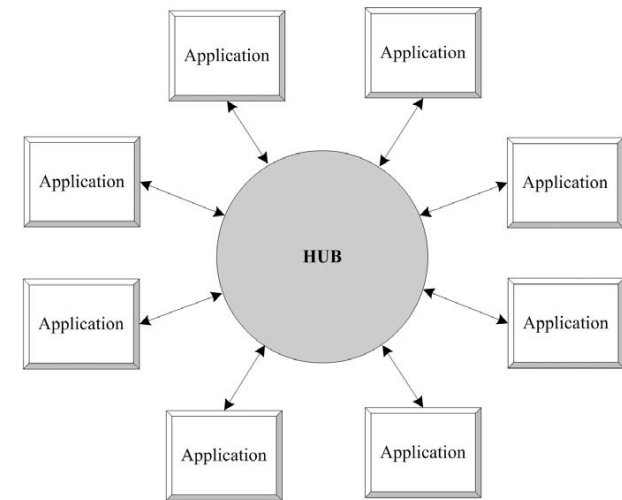
- O resultado da integração não deve recriar um *spaghetti system* dentro da solução de integração
- Para manter o SI flexível e reativo, deve favorecer-se a existência de ligações “relaxadas”
- Integração por propagação de dados e integração de processos *multi-step* apenas implementam interações assíncronas e unidirecionais
- Neste caso, dá-se preferência na utilização de meios assíncronos de comunicação, tais como MOMs ou gestores de transferência de ficheiros
- Apenas a integração de aplicações *composite* necessita de interações síncronas e bidirecionais para as quais se deve usar tecnologias tais como:
 - SGBD
 - ORBs (*Object Request Brokers*)
 - chamadas RMI (*Remote Method Invocation*)
 - chamadas RPC (*Remote Procedure Call*)
 - chamadas a *Web Services* como parte de uma abordagem SOA.

Arquitetura: centralizada ou distribuída?

- Depois de determinar o tipo de comunicação, é necessário definir a arquitetura
- A escolha da arquitetura pode influenciar as ferramentas para a sua implementação

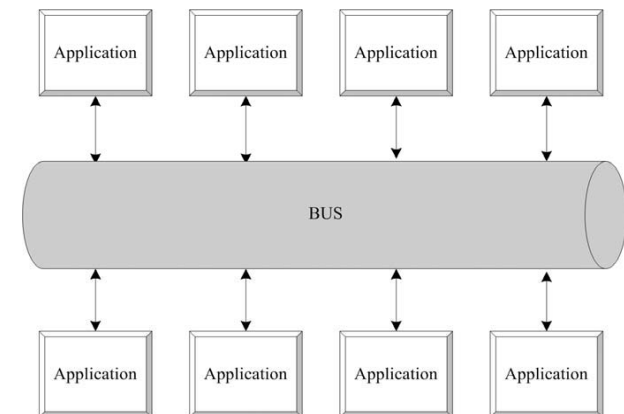
Arquitetura centralizada

- Concentra-se num determinado ponto para completar um conjunto de serviços assegurados pela infraestrutura
- As aplicações são integradas para se ligar a este ponto usando os adaptadores apropriados
- **Vantagem:** Numa arquitetura com múltiplas máquinas, esta arquitetura (denominada “*hub*” e “*spoke*” na terminologia de ferramentas de EAI) possibilita não ser preciso fazer *deploy* de todos os componentes da infraestrutura no conjunto completo de plataformas
- **Desvantagem:** se a plataforma que suporta o *hub* ficar indisponível ou *overloaded*, pode potencialmente gerar um ponto de falha (SPOF: *single point of failure*) ou de contenção no SI



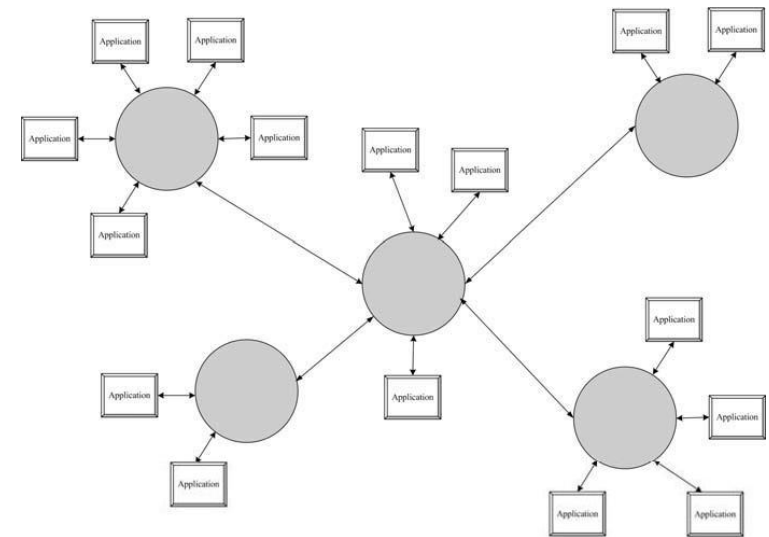
Arquitetura distribuída

- Partilha a infraestrutura pelas diversas máquinas que suportam as aplicações a ser integradas, com cada aplicação a ligar-se localmente à infraestrutura através de adaptadores
- **Vantagem:** Não introduz SPOF ou de contenção no sistema
- **Desvantagem:** requer ferramentas adaptadas à heterogeneidade das plataformas em que os seus componentes devem ser publicados, bem como ferramentas para supervisionar e manter esses componentes nas diversas plataformas



Arquitetura snowflake

- Uma vez que a distribuição pode funcionar ao nível da integração de serviços, vemo-nos na presença de uma arquitetura tipo bus.
- Os serviços de integração são distribuídos de forma a assegurar uma “rede” na infraestrutura de comunicação, tornando a solução mais robusta (sem SPOF)
- Isto também pode ser alcançado pela publicação de múltiplas plataformas de integração no sistema, criando uma arquitetura floco-de-neve.
 - Este tipo de arquitetura é importante no caso de projetos em larga escala em grandes empresas.
 - Cada domínio de responsabilidade está sobre controlo das suas comunicações e comunica de uma forma standard com outras grandes entidades da empresa.
 - É uma ilustração operacional de como lidar com problemas em A2B (Application to Business), ou BC (Business Collaboration)





UNIVERSIDADE
PORTUCALENSE

Do conhecimento à prática.