Análise Arquitetural

Fabio Vinicius Binder

fabio.binder@pucpr.br

PUCPR

Agenda

- Revisão de Conceitos
- Método ATAM
- Exemplo de aplicação: Adventure Builder
  - Cenários de atributos de qualidade
  - Arquitetura
  - Análise arquitetural
- Abordagens arquiteturais SOA
- Questões de projeto SOA

PUCPR

ATAM

- Architecture Tradeoff Analysis Method
- Proposto pelo Software Engineering Institute (SEI) da Carnegie Mellon University
- Ajuda a escolher a arquitetura adequada para um sistema de software
- O método ATAM analisa cenários de requisitos não funcionais (atributos de qualidade) e permite entender
  - o quanto uma arquitetura particular satisfaz os objetivos desses cenários
  - e como os atributos de qualidade interagem

PUCPR

ATAM

- Os participantes do sistema determinam a importância dos atributos de qualidade e a adequação de uma arquitetura a esses atributos
- O método foi criado para descobrir os riscos e tradeoffs refletidos em decisões arquiteturais
- Atributos de qualidade (requisitos não funcionais)
  - Desempenho
  - Disponibilidade
  - Segurança
  - Testabilidade
  - Interoperabilidade
  - Confiabilidade
  - Extensibilidade

PUCPR

ATAM

- Cenários declaram precisamente requisitos de uso, performance e crescimento, vários tipos de falhas e várias ameças e modificações potenciais
- Uma vez identificados os atributos de qualidade, as decisões arquiteturais relevantes para cada cenário prioritário podem ser compreendidas e analisadas
- A análise revela
  - Riscos: decisões arquiteturais que podem criar problemas futuros para alguns atributos de qualidade
    - Exemplo: A versão atual do SGBD não é mais suportada pelo vendedor; portanto, não serão mais criados pacotes de correção de vulnerabilidade de segurança

PUCPR

ATAM

- A análise revela (continuação)
  - Não riscos: decisões arquiteturais que são apropriadas no contexto dos atributos de qualidade que elas afetam
    - Exemplo: a decisão de introduzir concorrência melhora a latência; o pior tempo de execução para todas as threads é menor que 50% do seu limite
  - Tradeoffs: decisões arquiteturais que têm um efeito em mais de um atributo de qualidade
    - Exemplo: introduzir concorrência melhora latência, mas aumenta o custo de manutenção dos módulos afetados
  - Pontos sensíveis: uma propriedade de um ou mais componentes, e/ou relacionamentos entre componentes, crítica para a satisfação de um requisito específico
    - Exemplo: decisão de usar REST em vez de SOAP para comunicação entre usuários e provedores de serviço

PUCPR

ATAM

- Etapas
  - Apresentação do método: a **equipe de avaliação** apresenta os passos do ATAM, as técnicas usadas e as saídas do processo
  - Apresentação das diretrizes do negócio: o **gerente do sistema** apresenta brevemente as diretrizes do negócio e o contexto da arquitetura
  - Apresentação da arquitetura: o **arquiteto** apresenta uma visão geral da arquitetura
  - Identificação das abordagens arquiteturais: **a equipe de avaliação** e o **arquiteto** listam as abordagens arquiteturais mostradas no passo anterior
  - Geração da árvore de atributos de qualidade: um pequeno **grupo de participantes técnicos** identifica, prioriza e refina os objetivos mais importantes dos atributos de qualidade em forma de árvore

PUCPR

ATAM

- Etapas (continuação)
  - Análise das abordagens arquiteturais: a **equipe de avaliação** examina as abordagens arquiteturais com base nos atributos de qualidade para identificar riscos, não riscos e tradeoffs (utiliza-se um questionário de apoio)
  - Brainstorm e priorização de cenários: um **grande e mais diversificado grupo de participantes** cria cenários que representam seus interesses; então, o grupo vota para priorizar os cenários com base em sua importância relativa
  - Análise das abordagens arquiteturais: a **equipe de avaliação** continua a identificar riscos, não riscos e tradeoffs enquanto anota o impacto de cada cenário nas abordagens arquiteturais
  - Apresentação dos resultados: a **equipe de avaliação** recapitula os passos ATAM, saídas e recomendações

PUCPR

ATAM

- Participantes (stakeholders)
  - Produtores de sistemas: arquitetos de software, desenvolvedores, controladores de uso dos serviços, testadores, integradores, desenvolvedores de manutenção, diretores-executivos de informação (CIOs)
  - Consumidores de sistemas: diretores-executivos de segurança (CSOs), gerentes de negócio, analistas de negócio/clientes, usuários finais, desenvolvedores de usuários de serviços, desenvolvedores de manutenção
  - Provedores de infra-estrutura: administradores de sistemas, administradores de redes, administradores de bancos de dados, desenvolvedores externos de provedores de serviços

PUCPR

Atributos de Qualidade

- Exemplos de Cenários
  - Desempenho:
    - Uma requisição esporádica para o serviço X é recebida pelo servidor durante sua operação normal. O sistema processa a requisição em menos de Y segundos.
    - O provedor de serviços pode processar até X requisições simultâneas durante a operação normal, mantendo o tempo de resposta menor que Y segundos.
    - O tempo de ida e volta de uma requisição de um serviço de usuário da rede local para o serviço X leva menos que Y segundos.

PUCPR

Atributos de Qualidade

- Exemplos de Cenários
  - Disponibilidade:
    - Uma mensagem em formato incorreto é recebida pelo sistema. É gravada e o sistema continua a operar sem *downtime*
    - Um alto número de requisições suspeitas é recebido (ataque DOS) e o sistema é sobrecarregado. O sistema grava as requisições, avisa o administrador e continua normalmente
    - Manutenção não programada é necessária no servidor X. O sistema continua operando em modo *degraded* durante a manutenção
    - Uma requisição é processada de acordo com sua especificação em pelo menos 99,99% dos casos
    - Um novo serviço é implementado sem impactar nas operações
    - Um serviço externo pára de responder. O sistema continua a operar sem falhas

PUCPR

Atributos de Qualidade

- Exemplos de Cenários
  - Segurança:
    - Um serviço de terceiros com código mal intencionado é usado pelo sistema. O sistema avisa o administrador e não permite que o serviço acesse dados ou interfira com as operações.
    - Um ataque é executado para tentar acessar dados confidenciais. O atacante não é capaz de quebrar a criptografia. O sistema grava o ataque e avisa o administrador.
    - Uma requisição precisa ser enviada para um serviço de terceiros mas a identidade do provedor não pode ser validada. O sistema não faz a requisição e grava informações relevantes. O provedor e o administrador são avisados.

PUCPR

Atributos de Qualidade

- Exemplos de Cenários
  - Segurança:
    - Um serviço não autorizado tenta invocar um serviço protegido mas tem o acesso negado.
    - Um atacante está modificando requisições que chegam com o objetivo de efetuar um ataque na infraestrutura. O sistema identifica e descarta as mensagens alteradas, grava informações e avisa o administrador.
    - Um atacante tenta alterar o registro de serviço para redirecionar requisições. O sistema nega acesso, grava informações e avisa o administrador.

PUCPR

Atributos de Qualidade

- Exemplos de Cenários
  - Testabilidade:
    - Um testador de integração executa testes em uma nova versão de serviço que oferece uma interface para observar a saída de dados. 90% de cobertura é alcançada com uma pessoa/semana.

PUCPR

Atributos de Qualidade

- Exemplos de Cenários
  - Interoperabilidade:
    - Um novo parceiro de negócios que usa a plataforma X é capaz de implementar um módulo de serviços que trabalha com nosso serviços na plataforma Y em 2 dias
    - Uma transação de um sistema legada que executa em uma plataforma X é disponibilizada como um Web Service para uma aplicação coorporativa que está sendo desenvolvida na plataforma Y. O processo de transformar a operação legada em um serviço, incluindo segurança, gerenciamento de transações e tratamento de erros é feito em 10 dias

PUCPR

Atributos de Qualidade

- Exemplos de Cenários
  - Extensibilidade:
    - Um provedor muda a implementação de um serviço mas a sua interface não muda. A alteração não afeta os usuários.
    - Um provedor muda a interface de um serviço público. A versão antiga continua funcionando por 12 meses e nenhum usuário é afetado no período.

PUCPR

Atributos de Qualidade

- Exemplos de Cenários
  - Confiabilidade:
    - Uma falha súbita ocorre em um provedor. Após a recuperação, todas as transações são completadas ou desfeitas de maneira apropriada. Os dados persistidos do sistema mantém-se confiáveis.
    - Um serviço fica indisponível. O sistema detecta o problema e restaura o serviço em menos de 2 minutos.

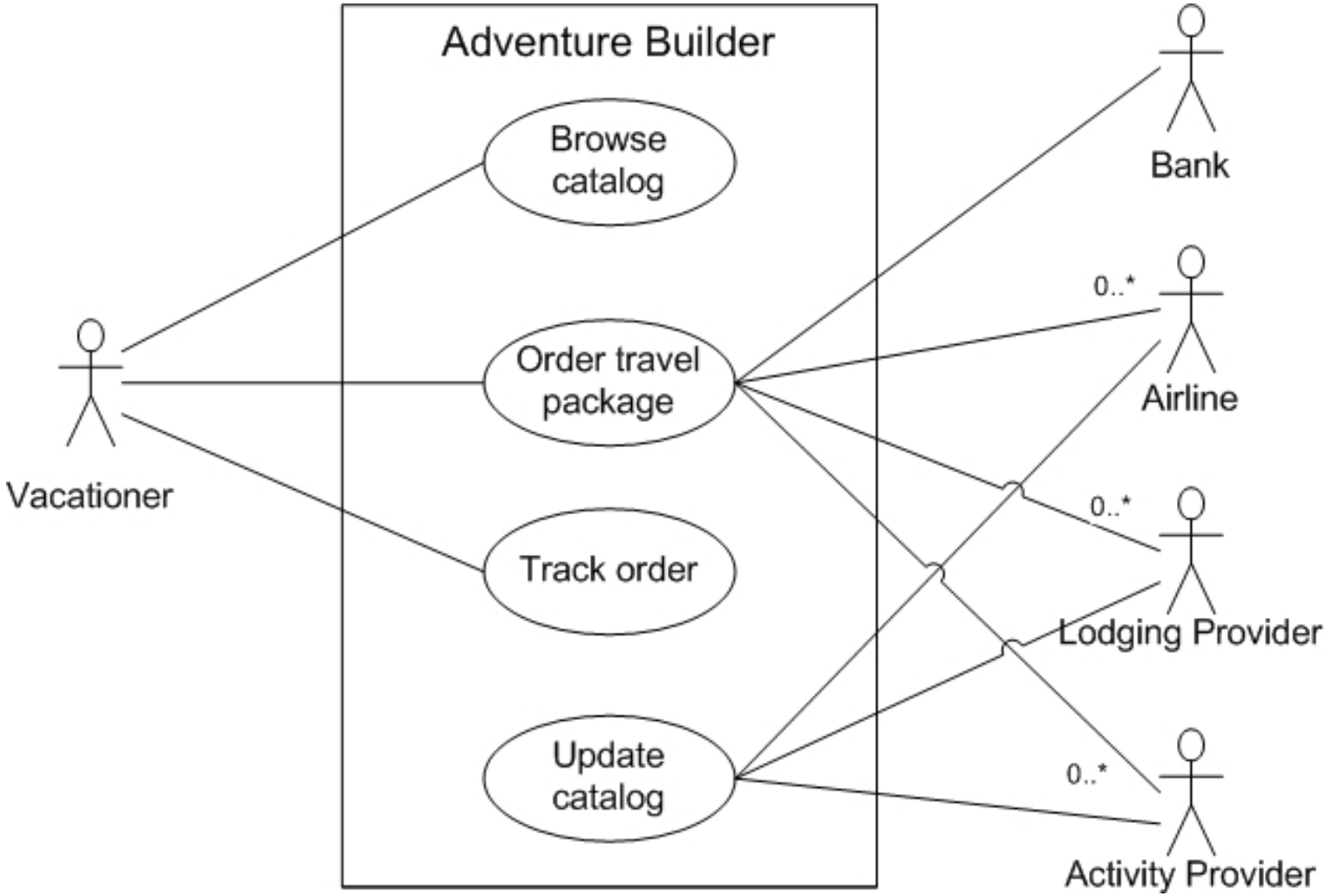
PUCPR

Descrição dos Cenários

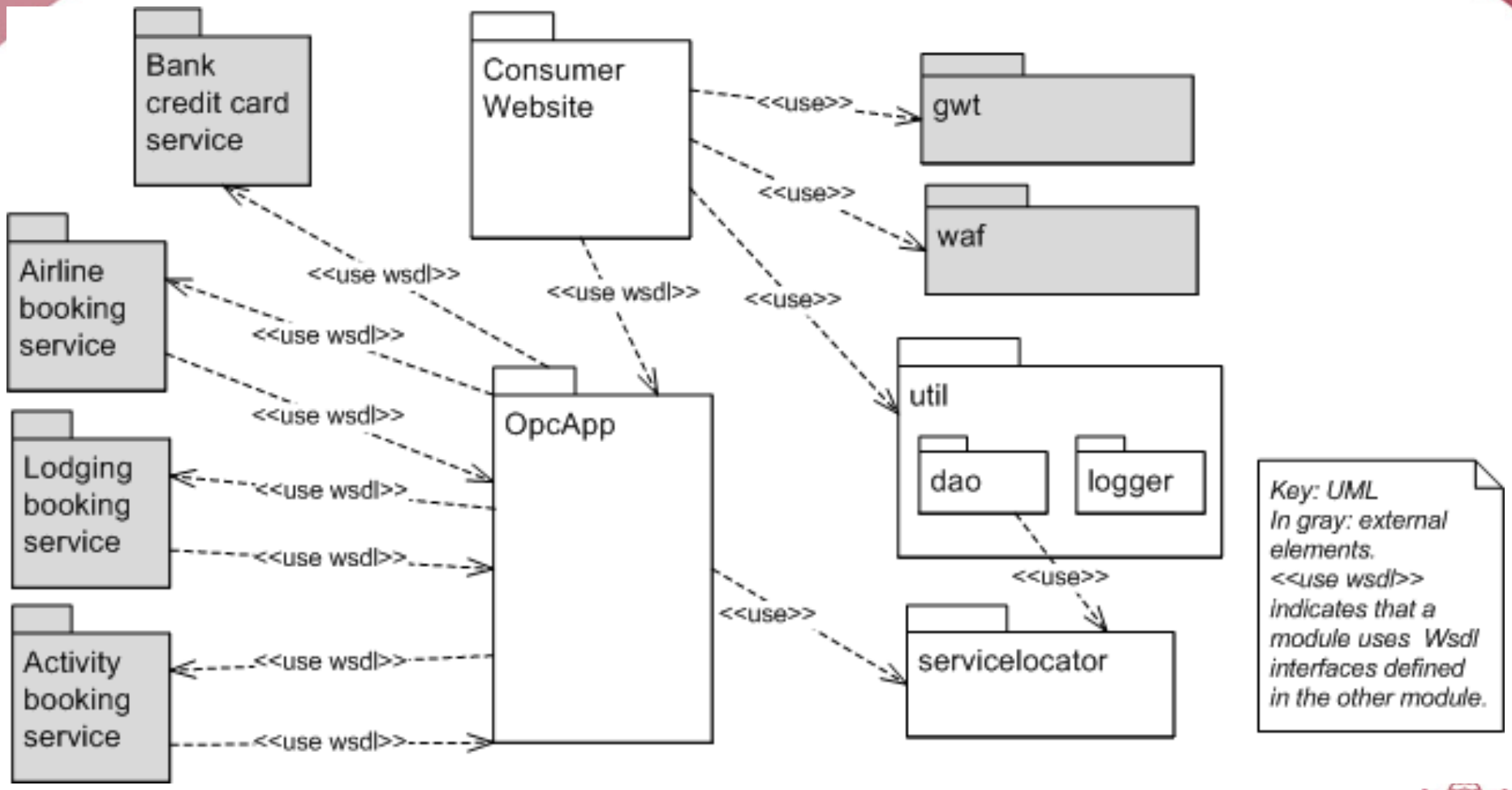
- Source (Fonte)
- Estimulus (Estímulo)
- Artifact (Artefato)
- Environment (Ambiente)
- Response (Resposta)
- Response Measure (Medida máxima de resposta)

PUCPR

Adventure Builder

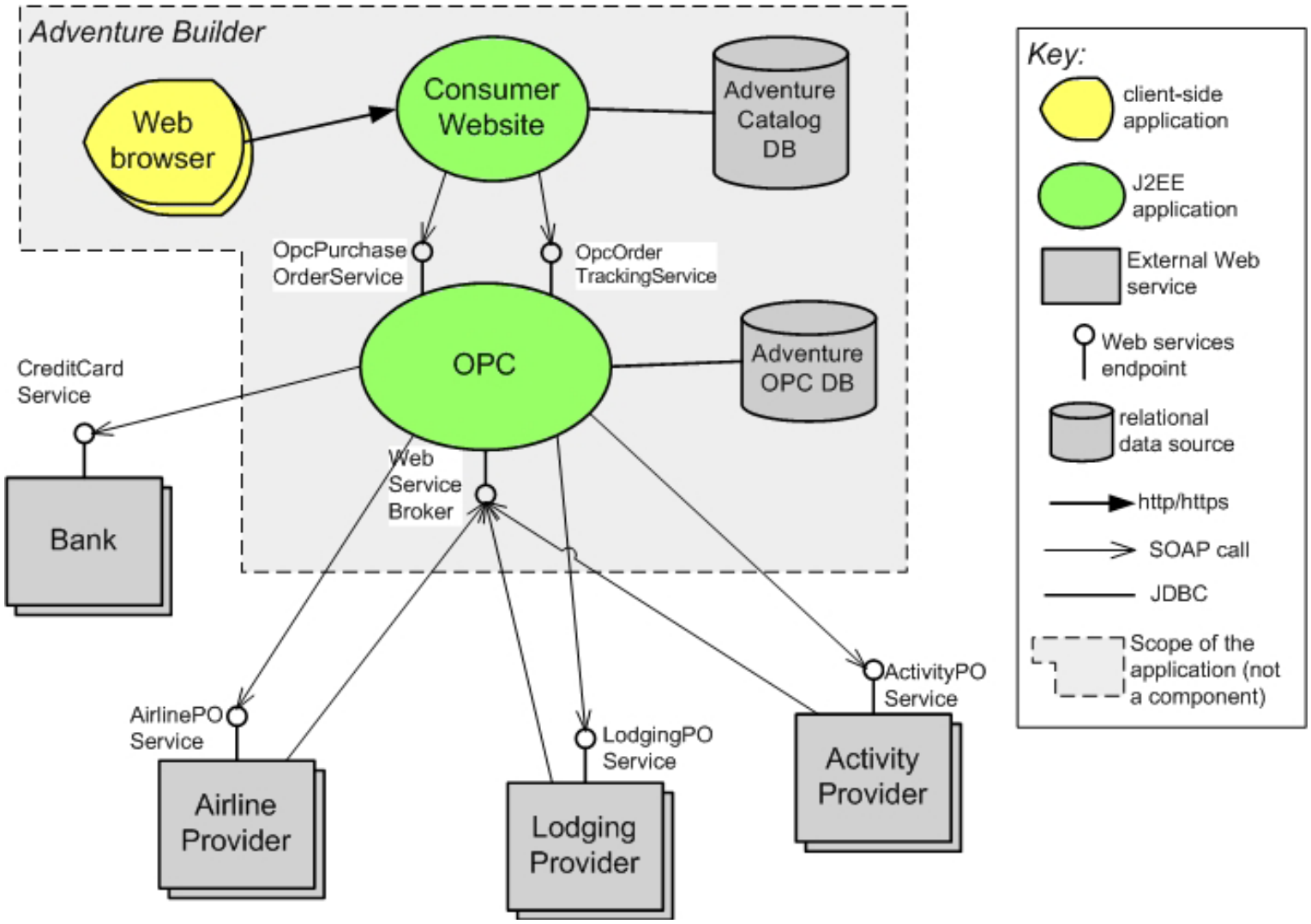


PUCPR



PUCPR

Arquitetura do AB



PUCPR

Cenários para o AB

Quality Attribute	Scenario
Scenario 1. Modifiability	• **(Source)** Business Analyst/Customer • **(Stimulus)** Add a new business partner (transportation, lodging, or activity provider) to use Adventure Builder's predefined Web services. • **(Artifact)** OPC • **(Environment)** Business partner familiar with the OPC interface and Web services technology • **(Response)** New business partner is added using Adventure Builder's Web services • **(Response Measure)** No more than one person-day of Adventure Builder team effort is required for the implementation (legal and financial agreements are not included).
Scenario 2. Modifiability	• **(Source)** Business Analyst/Customer • **(Stimulus)** Add a new airline provider that uses its own Web services interface. • **(Artifact)** OPC • **(Environment)** Developers have already studied the airline provider interface definition. • **(Response)** New airline provider is added that uses its own Web services. • **(Response Measure)** No more than 10 person-days of effort are required for the implementation (legal and financial agreements are not included).

PUCPR

Cenários para o AB

Quality Attribute	Scenario
Scenario 3. Modifiability	• **(Source)** Business Analyst/Customer • **(Stimulus)** Add weather information for selected destinations that includes average daily temperature and average monthly precipitation. • **(Artifact)** Consumer Web site • **(Environment)** Developers familiar with the interface definition of the weather service • **(Response)** The external service that provides weather information is integrated with the system, and the new feature is available to Adventure Builder users. • **(Response Measure)** No more than two person-months of effort are required for the implementation.
Scenario 4. Performance	• **(Source)** User • **(Stimulus)** User submits an order for a package to the Consumer Web site. • **(Artifact)** Adventure Builder system and the Bank • **(Environment)** Normal operation • **(Response)** The Consumer Web site notifies the user that the order has been successfully submitted and is being processed by the OPC. • **(Response Measure)** The system responds to the user in less than five seconds.

PUCPR

Cenários para o AB

Quality Attribute	Scenario
Scenario 5. Reliability	• **(Source)** External to system • **(Stimulus)** The Consumer Web site sent a purchase order request to the OPC. The OPC processed that request but didn't reply to Consumer Website within five seconds, so the Consumer Web site resends the request to the OPC. • **(Artifact)** Adventure Builder system • **(Environment)** Normal operation • **(Response)** The OPC receives the duplicate request, but the consumer is not double-charged, data remains in a consistent state, and the Consumer Web site is notified that the original request was successful. • **(Response Measure)** In 100% of the transactions

PUCPR

Cenários para o AB

Quality Attribute		Scenario
Scenario 6.	Reliability	• **(Source)** System failure in the OPC • **(Stimulus)** The OPC sends a request to the bank to charge a credit card for a purchased travel package; before receiving the reply from the bank, the OPC crashes. • **(Artifact)** OPC and bank service • **(Environment)** Failure mode • **(Response)** The system recovers in a correct and consistent way, and the credit card is charged only once. • **(Response Measure)** In 100% of the cases
Scenario 7.	Availability	• **(Source)** Internal to the system • **(Stimulus)** Fault occurs in the OPC • **(Artifact)** OPC • **(Environment)** Under normal operation • **(Response)** The system administrator is notified of the fault; the system continues taking order requests; another OPC instance is created; and data remains in consistent state. • **(Response Measure)** The fault is detected, and failover action is taken within 30 seconds.

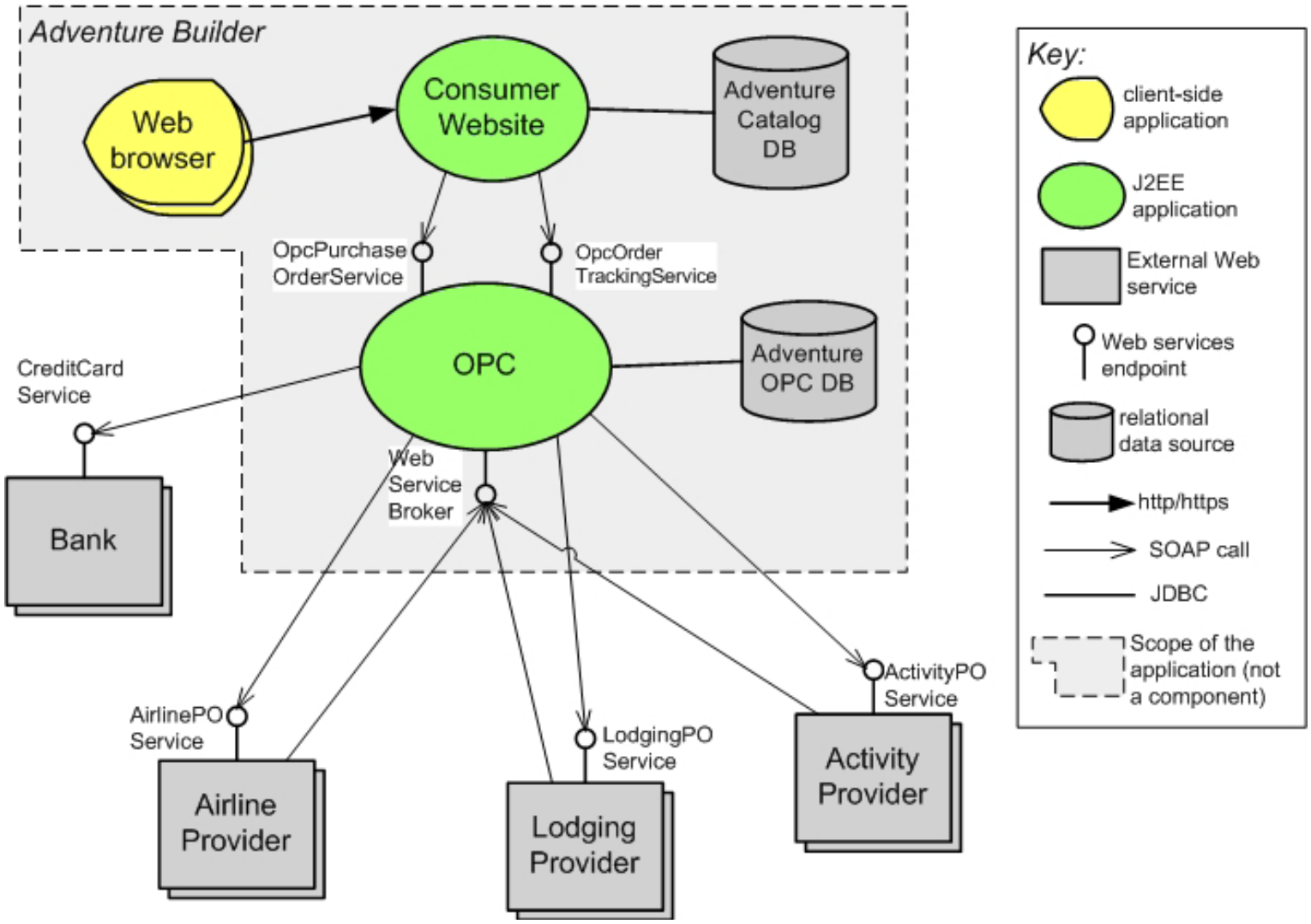
PUCPR

Cenários para o AB

Quality Attribute		Scenario
Scenario 8.	Security/ Availability	• **(Source)** External to system • **(Stimulus)** The OPC experiences a flood of calls through the Web Service Broker endpoint that do not correspond to any current orders. • **(Artifact)** OPC • **(Environment)** Normal operation • **(Response)** The system detects the abnormal level of activity and notifies system administrators. • **(Response Measure)** The system continues to service requests in degraded mode.
Scenario 9.	Security	• **(Source)** User • **(Stimulus)** Credit approval and payment processing functions are requested for a pending order. • **(Artifact)** OPC and the Bank's service • **(Environment)** Normal operation • **(Response)** The credit approval is completed securely and cannot be refuted by either party. • **(Response Measure)** In 100% of the transactions

PUCPR

Arquitetura do AB



PUCPR

Arquitetura do AB

- Embora um ESB (Enterprise Service Bus) não seja usado, o OPC (Order Processing Center) tem um gerenciador centralizado de workflow que contém todas as regras de negócio e a lógica de coordenação do processamento dos pedidos do cliente
- O uso do OPC como mediador reduz as dependências entre os provedores de serviço externos, promovendo modificabilidade
- O OPC se comunica com o Consumer Web Site usando SOAP (Web Services), proporcionando interoperabilidade com uma ampla variedade de tecnologias
- As interfaces Web Services (descritas em WSDL) propiciam baixo acoplamento entre o OPC e os softwares dos parceiros

PUCPR

Arquitetura do AB

- Web Services entre o OPC e o Consumer Web Site permite que o web site seja hospedado fora do firewall em uma zona desmilitarizada enquanto o OPC permanece dentro do firewall
- A comunicação é realizada através de uma porta http disponível no firewall
- Web Services permitem que o Consumer Web Site e o OPC sejam instalados em plataformas de hardware e software diferentes
- Uma avaliação focada em integração de serviços não cobre todas as decisões arquiteturais importantes
  - O padrão arquitetural usado no Consumer Web Site é o MVC (Model-View-Controller) com o objetivo de promover modificabilidade

PUCPR

Análise Arquitetural

- A análise proposta pelo método ATAM não tem a intenção de ser precisa e detalhada
- Não fornece valores numéricos
- Os objetivos são
  - Obter informação arquitetural suficiente para identificar riscos, que resultam da correlação entre decisões arquiteturais e seus efeitos nos atributos de qualidade
  - Avaliar se os requisitos dos atributos de qualidade podem ser satisfeitos
  - Capturar as abordagens arquiteturais e identificar seus riscos, não riscos, pontos sensíveis e tradeoffs

PUCPR

Descrição da Análise dos Cenários

- Sumário
- Objetivos de Negócio
- Atributo de Qualidade
- Abordagem Arquitetural
- Riscos
- Trade-offs

PUCPR

Análise Arquitetural

Analysis for Scenario 2	
Scenario Summary	A new airline provider that uses its own Web services interface is added to the system in no more than 10 person-days of effort for the implementation.
Business Goal(s)	Permit easy integration with new business partners.
Quality Attribute	Modifiability, interoperability
Architectural Approaches and Reasoning	• Asynchronous SOAP-based Web services (see Sections 4.1 and 5.2) • Interoperability is improved by the use of document-literal SOAP messages (see Section 4.1) for the communication between OPC and external services. • Adventure Builder runs on Sun Java System Application Server Platform Edition V8.1. This platform implements the WS-I Basic Profile V1.1, so interoperability issues across platforms are less likely to happen (see Section 5.1).
Risks	• The design does not meet the requirement in this scenario, because it assumes that all external transportation providers implement the same Web services interface called AirlinePOService (as shown in Figure 10 and Figure 11). The design does not support transportation providers that offer their own service interface.[7]
Tradeoffs	• The homogenous treatment of all transportation providers in OPC increases modifiability. However, intermediaries are needed to interact with external providers that offer heterogeneous service interfaces, as in this scenario. These intermediaries represent a performance overhead, because they may require routing messages and extensive XML processing.

PUCPR

Análise Arquitetural

Analysis for Scenario 4	
Scenario Summary	User submits an order for a package to the Consumer Web site. The system responds to the user in less than five seconds.
Business Goal(s)	Provide satisfactory user experience.
Quality Attribute	Performance
Architectural Approaches and Reasoning	• The use of document-literal SOAP results in better performance, because there is no overhead associated with encoding (see Section 4.1). • Static Web service (see Section 4.4) prevents the overhead of looking up a registry. • The Web services were designed around the documents handled, such as purchase orders and invoices. The OPC Purchase Order Service interface (see Figure 10) is coarse grained in the interest of improving system performance (see Section 5.3). This interface reduces the overhead of calling a fine-grained service for each step of a business process. • The OPC interacts with the Bank in a synchronous fashion (see Section 5.2). The charge is authorized quickly so that processing of the order may proceed. Then the OPC sends requests to transportation, lodging, and activity providers, which will later respond through the Web Service Broker callback endpoint. These requests are sent asynchronously to improve scalability and throughput and also because of the nature of the legacy systems supporting this interface.

PUCPR

Análise Arquitetural

Risks	• The Adventure Builder architects have no control over the latency of external service providers.[8]
Tradeoffs	• Using Web services for communication with the Consumer Web site and external entities promotes loose coupling and interoperability. However, the overall latency of requests increases because of the overhead required for translating among WSDL, Java, and the other XML processing involved (see Section 5.8).

PUCPR

Análise Arquitetural

Analysis for Scenario 5	
Scenario Summary	The Consumer Web site sent a purchase order request to the OPC. The OPC processed the request but didn't reply to the Consumer Web site within five seconds. So the Consumer Web site resends the request to the OPC. The OPC receives the duplicate request, but the consumer is not double-charged; data remains in a consistent state; and the Consumer Web site is notified that the original request was successful.
Business Goal(s)	Provide satisfactory user experience by preventing overcharges and double booking.
Quality Attribute	Reliability
Architectural Approaches and Reasoning	• No transactions are distributed across multiple databases. Each piece of an order is a separate transaction. The centralized workflow manager in the OPC contains the state of each order during processing, and the database supports atomic transactions. • A correlation identifier to match requests and asynchronous responses allows idempotent endpoints for service providers that update or change state (see Section 5.4). The use of idempotent endpoints promotes reliability.
Risks	None
Tradeoffs	• A single database promotes reliability and reduces complexity at the expense of availability by introducing a single point of failure in the OPC. • The use of idempotent endpoints promotes reliability but imposes performance overhead and adds complexity to implementation.

PUCPR

Análise Arquitetural

Analysis for Scenario 9	
Scenario Summary	Credit approval and payment processing functions must be secure and provide for non-repudiation.
Business Goal(s)	• Provide customers, Adventure Builder's business, and business partners with confidence in security. • Meet contractual, legal, and regulatory obligations for secure credit transactions.
Quality Attribute	Security
Architectural Approaches and Reasoning	• Adventure Builder uses SSL mutual authentication (see Section 5.6). OPC and the Bank exchange digital certificates through an SSL handshake. Communication is encrypted using https. • Declarative authorization is used to set authorization rights (see Section 5.7).
Risks	• If certificate management is not done carefully, modifiability and interoperability will be negatively impacted. • The Adventure Builder system has only contractual (not technical) enforcement of information security management stored in partner systems.
Tradeoffs	• Implementing SSL mutual authentication adds complexity, hence increasing the time and costs of development and maintenance. • Encryption of messages exchanged with external partners adds some performance overhead.

PUCPR

Abordagens SOA

- A avaliação de um sistema SOA foca a integração dos serviços e os padrões de comunicação, não a arquitetura das aplicações subjacentes
- Abordagens arquiteturais SOA
  - Abordagens de comunicação
  - Abordagem de integração
  - Linguagem de execução de processos de negócio
  - Serviços estáticos ou dinâmicos

PUCPR

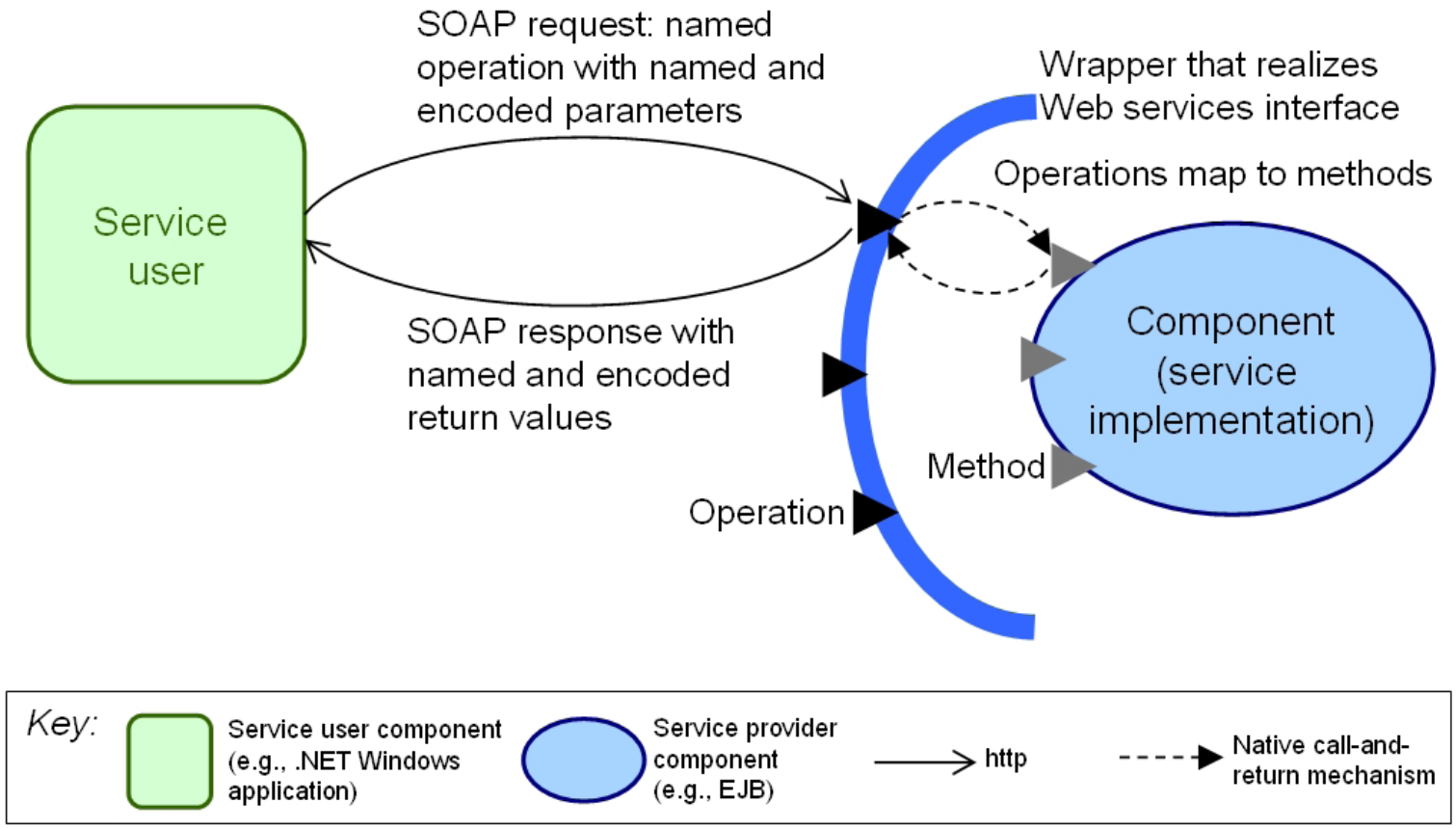
Comunicação

- SOAP-Based Web Services
  - RPC-Encoded SOAP
  - Document-Literal SOAP
- REST
- Messaging Solutions

PUCPR

Comunicação

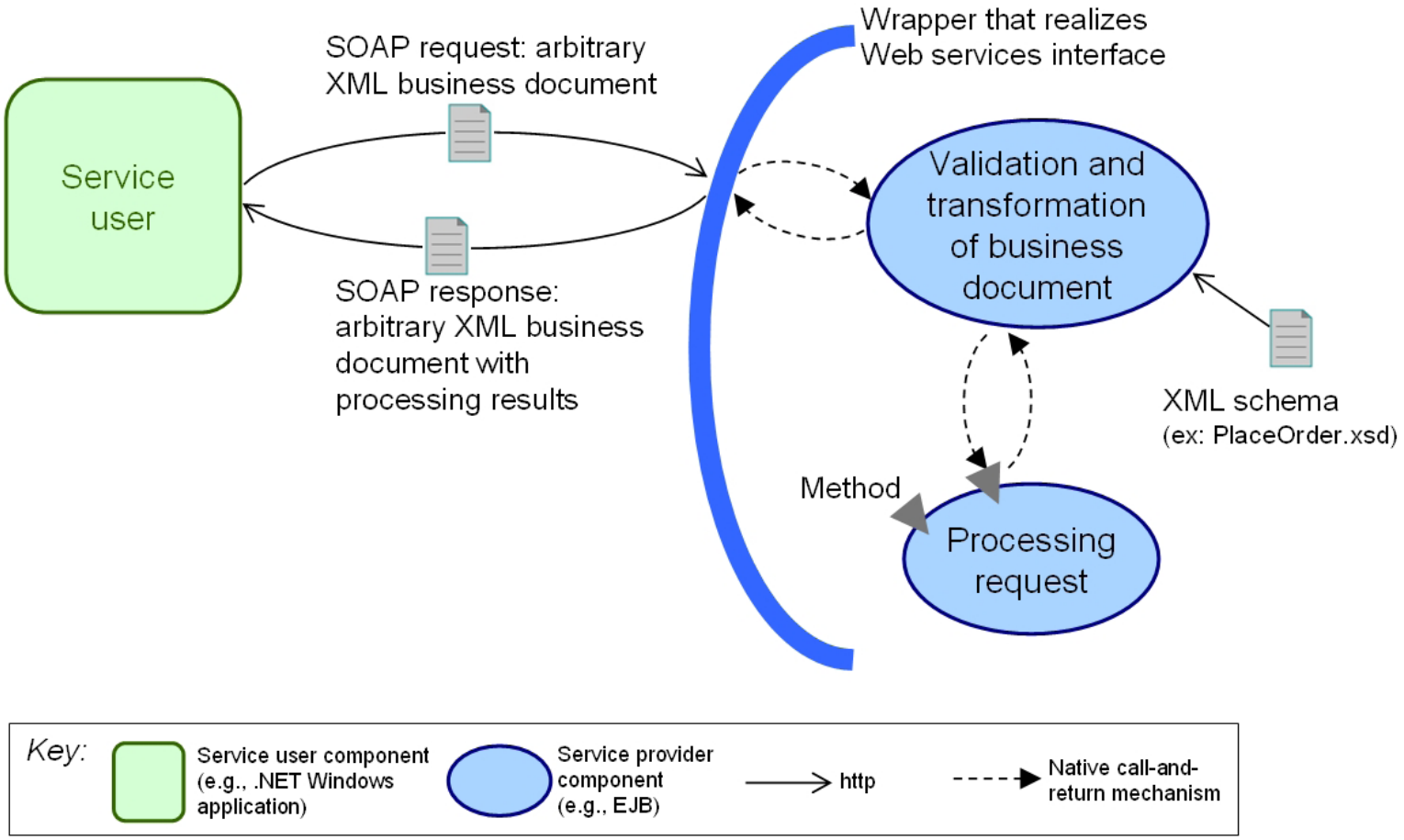
- RPC-Encoded SOAP



PUCPR

Comunicação

- Document-Literal SOAP



PUCPR

Comunicação

Quality Attribute	RPC-Encoded	Document-Literal
Interoperability	☹ Is less interoperable due to incompatibility in SOAP encoding across platforms	☺ Is more interoperable and recommended by WS-I
Performance	☹ May yield worse performance due to processing overhead required to encode payloads	☺ Requires no encoding overhead
	☹ Requires DOM parsing	☺ Allows other parsing technologies (e.g., SAX)
Modifiability	☺ In theory yields better modifiability because service interfaces are closer to programming language interfaces with operations and parameters. This similarity also enables the use of automatic object-to-WSDL translation.	☹ Is usually harder to implement because the XML schema definitions and the code to process and transform the XML documents are usually not created automatically
	☹ In practice, any change to the syntax of an operation requires changes in the service users, resulting in increased coupling.	☺ Yields less coupling. There is more flexibility to change the business document without affecting all service users.

PUCPR

Comunicação

- REST
  - Representational State Transfer
  - Interações entre usuários e provedores de serviços dependem do protocolo HTTP
  - O protocolo HTTP tem 4 operações básicas:
    POST, GET, PUT e DELETE
  - A execução dessas operações sobre recursos URI (Uniform Resource Identifier) corresponde a:
    CREATE, RETRIEVE, UPDATE e DELETE (CRUD)
  - Interface uniforme: em vez de usar métodos e parâmetros, o serviço é publicado como recursos de informação sobre os quais um conjunto fixo de operações pode ser aplicado
  - Para cada recurso deve ser definida uma representação (normalmente em formato XML)
  - Stateless: não guarda estado entre múltiplas requisições

PUCPR

Comunicação

- Exemplo REST
  - Recursos:
    - Tempo Atual para o zip code 15213
    - Previsão para amanhã em Detroit
    - Previsão para 10 dias em 15213
    - Temperaturas médias em outubro em Detroit
  - URIs:
    - http://www.weather.com/current/zip/15213
    - http://www.weather.com/forecast/tomorrow/city/Detroit
    - http://www.weather.com/forecast/tenday/zip/15213
    - http://www.weather.com/avg/city/Detroit?month=10

Comunicação

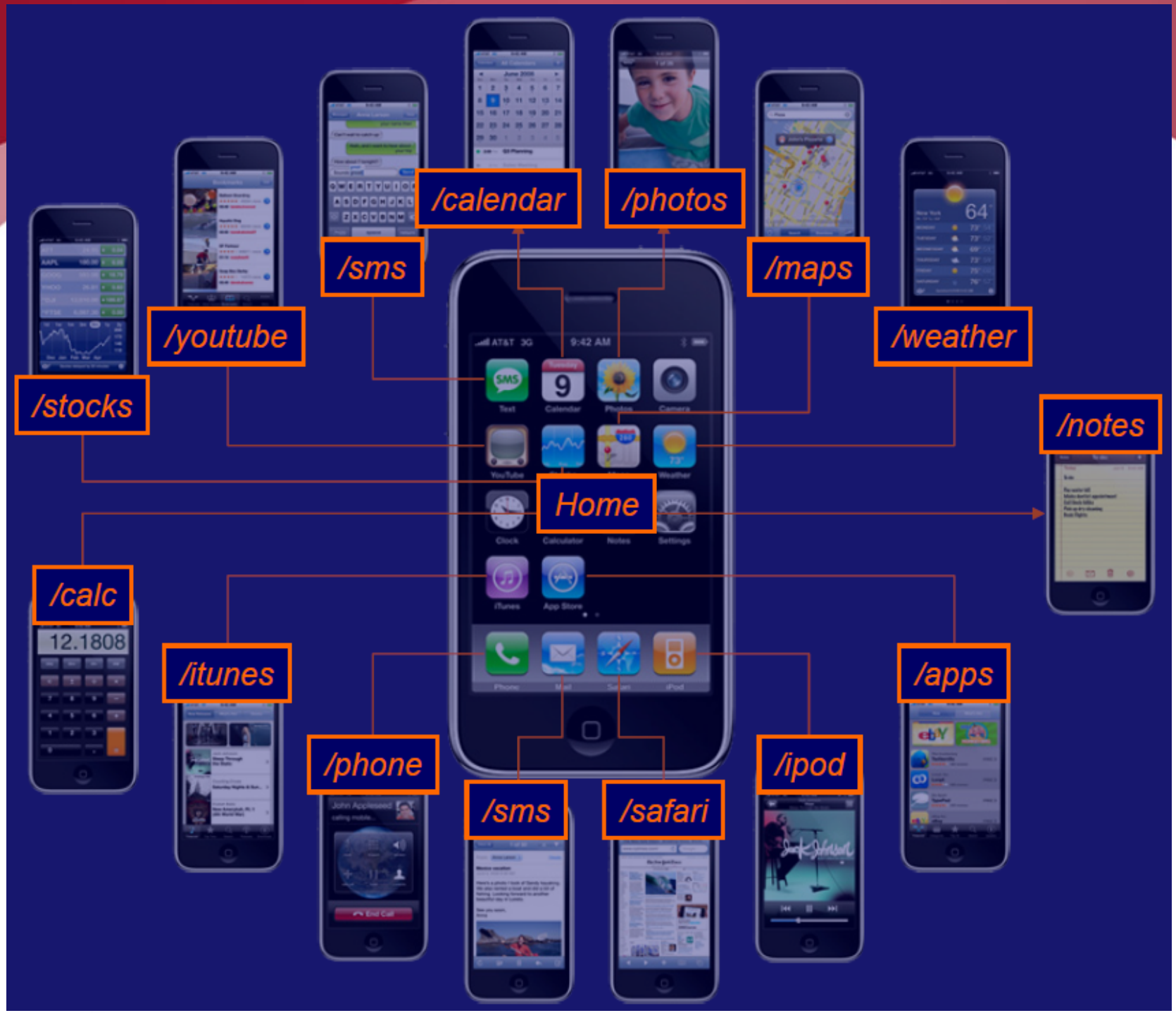
- Diferença de paradigma: SOAP é centrado em operações, REST em acesso a recursos
- Vantagens do REST sobre o SOAP
  - Maior modificabilidade: com SOAP é necessário conhecer a estrutura dos dados e as operações que podem ser executadas; com REST basta conhecer a estrutura dos dados
  - Mais fácil de implementar
  - Maior interoperabilidade: requer apenas http padrão
  - Melhor performance: caching de respostas, ausência de intermediários, message wrapping e serialização

PUCPR

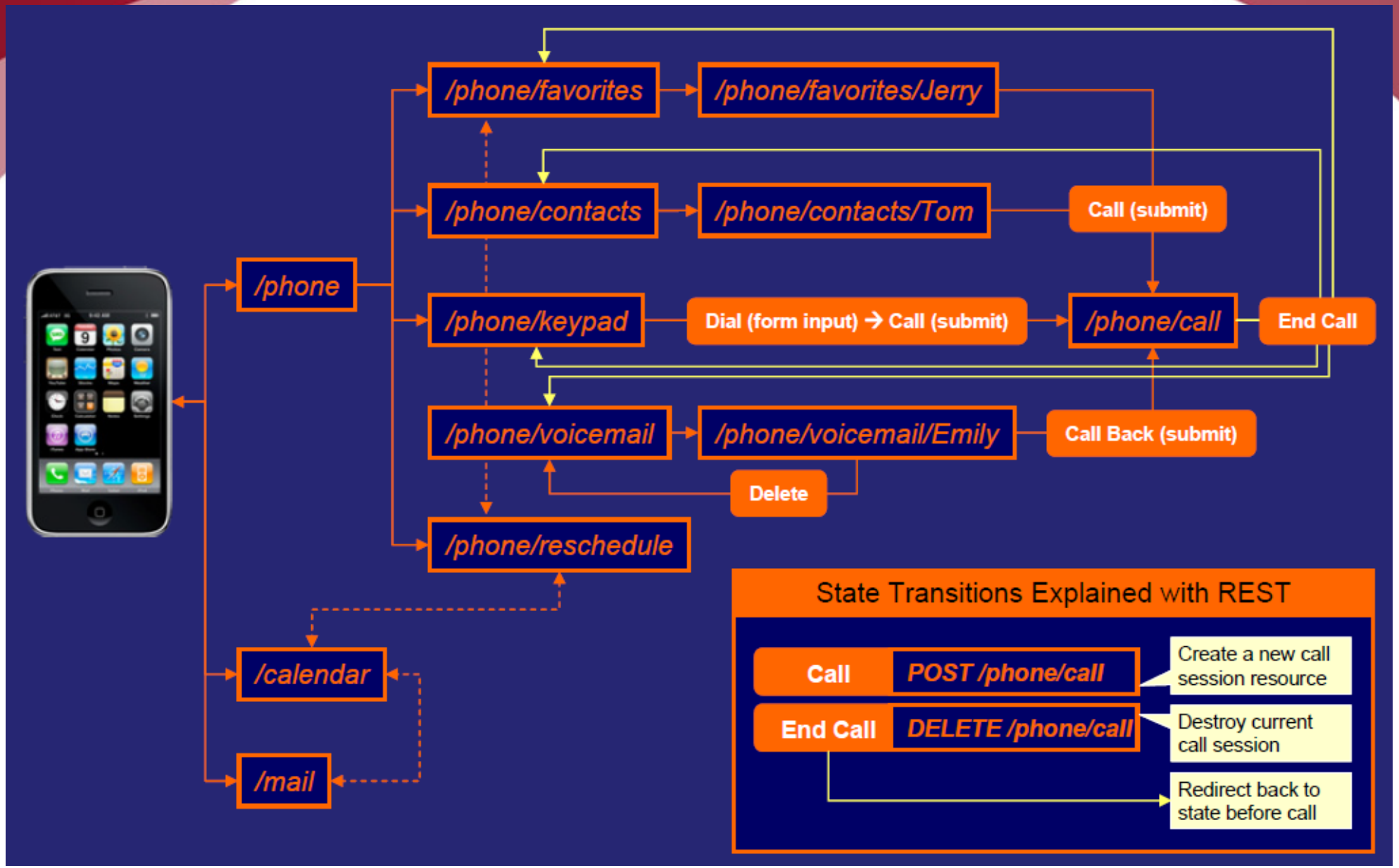
Comunicação

- Quando usar
  - REST
    - Acesso a recursos estáticos ou quase estáticos
    - Dispositivos portáteis com banda limitada (msgs simples)
  - Tecnologia Web Services
    - Melhor suporte para segurança, reliable messaging e gerenciamento de transação
    - Grande quantidade de conhecimento sobre Web Services está disponível na web e em comunidades profissionais
    - Melhor suporte de ferramentas para desenvolvimento
  - REST e Web Services (Amazon e eBay)
    - Para atender múltiplos usuários e parceiros de negócio

PUCPR



PUCPR



Quem usa REST?

- Facebook
- Netflix
- Amazon
- Atom Pub

PUCPR

Comunicação

- Messaging Solutions
  - Sistemas de mensagens: IBM WebSphere MQ, Microsoft MSMQ, Oracle AQ, Sonic MQ
  - Características predominantes desses produtos
    - Troca assíncrona de mensagens entre aplicações distribuídas de modo ponto-a-ponto (sender-receiver) ou publish-subscribe (EDA – event-driven architecture; mensagens são eventos e filas são canais)
    - Um administrador configura as filas de mensagens
    - As aplicações podem conectar com as filas e enviar ou receber mensagens

PUCPR

Comunicação

- Messaging Solutions
  - Principais benefícios
    - Maior confiabilidade com entrega garantida de mensagens
    - Acoplamento fraco entre produtores e consumidores de mensagens
    - Especialmente útil para conectar sistemas muito diferentes e aplicações legadas
    - Alta escalabilidade para suportar um número crescente de clientes (basta adicionar mais instâncias de consumidores de mensagens)
    - Podem ser programados para trabalhar offline: as mensagens são armazenadas no sender e, quando houver conectividade, são enviadas ao receiver

PUCPR

Comunicação

- Messaging Solutions
  - Desafios
    - O modelo de programação assíncrona é mais complexo
    - Custo de perfomance para empacotar as mensagens e armazená-las no computador do sender e/ou do receiver
    - Interoperabilidade: sistemas proprietários normalmente são baseados em protocolos proprietários e não estão disponíveis em todas as plataformas; bridges de terceiros costumam ser necessárias
  - Há soluções SOAP sobre sistemas de mensagens
  - Padrões em elaboração
    - WS-Reliability e WS-ReliableMessaging
    - Definem mensagens SOAP confiáveis via aknowledgments

PUCPR

Abordagens SOA

- A avaliação de um sistema SOA foca a integração dos serviços e os padrões de comunicação, não a arquitetura das aplicações subjacentes
- Abordagens arquiteturais SOA
  - Abordagens de comunicação
  - Abordagem de integração
  - Linguagem de execução de processos de negócio
  - Serviços estáticos ou dinâmicos

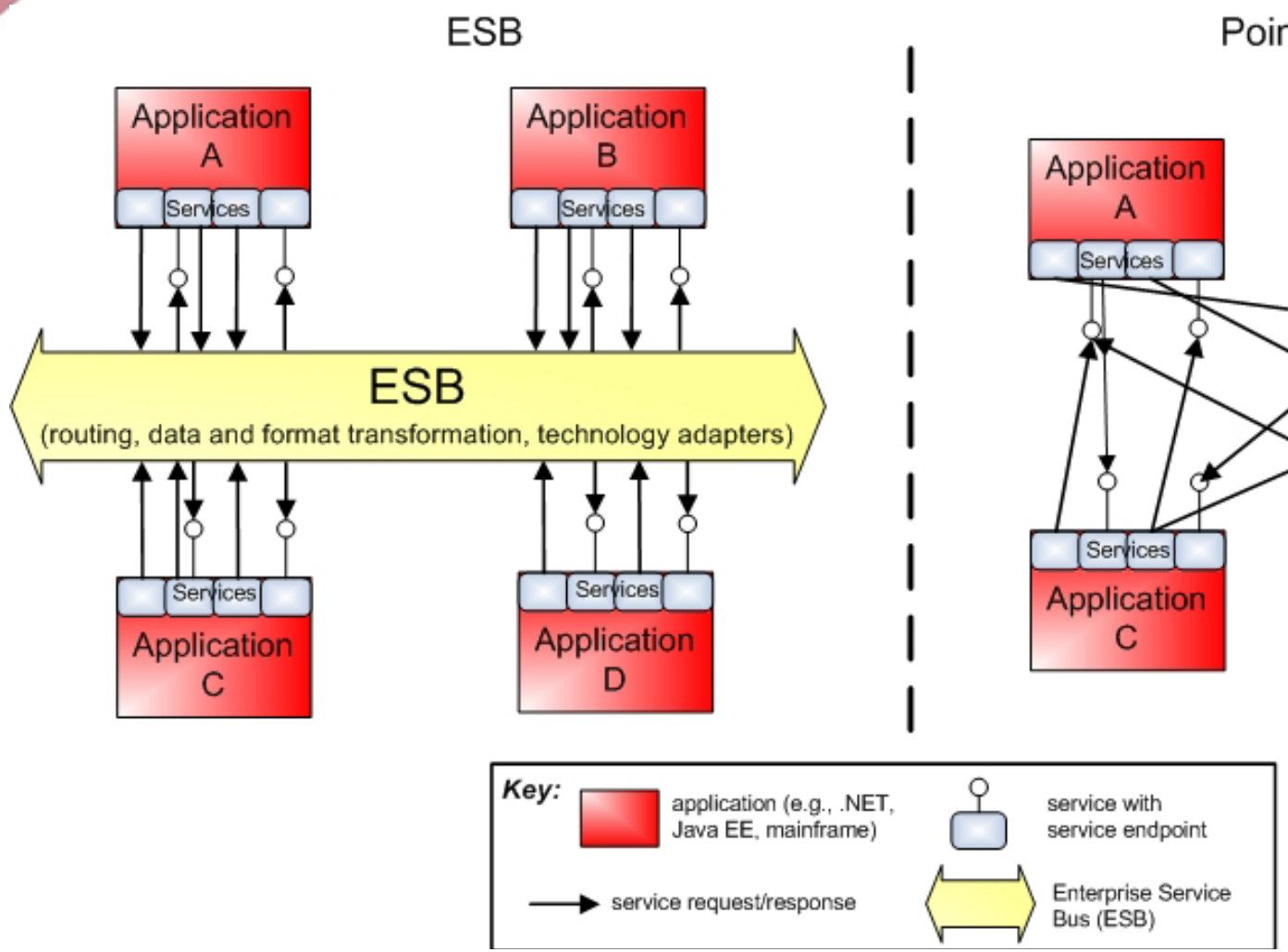
PUCPR

Integração

- Principais opções de padrão de integração
  - Direta ponto-a-ponto
    - A responsabilidade pelas interações entre usuários e provedores de serviços está distribuida
  - Hub-and-spoke
    - As interações entre usuários e provedores de serviço é mediada por software de brokering (ESB – Enterprise Service Bus; termo clássico EAI – Enterprise Application Integration)

PUCPR

Integração



PUCPR

Integração

- ESB refere-se a um padrão arquitetural e a produto
- ESB fornece suporte para Web Services
- ESB pode rotear mensagens para uma ou mais aplicações de modo fixo ou dinâmico, usando vários critérios de roteamento
- ESB pode transformar dados
- A funcionalidade do ESB pode ser distribuída por múltiplos servidores
- ESB suporta adaptadores para conectar com aplicações legadas e de prateleira (COTS – commercial off-the-shelf)
- ESB pode suportar autenticação, autorização e criptografia utilizando múltiplos padrões de segurança, como WS-Security, Kerberos e SSL (secure socket layer)
- Atributos de qualidade endereçados: interoperabilidade, modificabilidade e extensibilidade (adicionar serviços)

PUCPR

Integração

- Conseqüências do uso de ESB
  - Performance pode piorar: transformações de mensagens e message hops adicionais
  - Complexidade do sistema e custo de implementação inicial aumentam
  - Uma organização que adota ESB necessita
    - Definir uma estratégia de longo prazo compreendendo políticas e padrões para uso de ESB
    - Estabelecer processos que assegurem o uso do ESB
    - Avaliar a infra-estrutura do ESB e a plataforma de suporte

PUCPR

Integração

- Fatores para a escolha da estratégia de integração (ponto-a-ponto, hub-and-spoke, híbrida)
  - Número atual e planejado de aplicações e tecnologias integradas
  - Requisitos de vazão e tempo de resposta das aplicações integradas atuais e futuras
  - Padrões de comunicação (p.ex., síncrono, message queueing, publish-subscribe) e número crescente de serviços integrados pelas aplicações atuais e futuras
  - Requisitos de suporte a novas aplicações, transações de negócio e requisitos de dados
  - Taxa de adoção e maturidade de novas tecnologias e padrões
  - Dinânica de negócios, organizacional e regulatória (p.ex., velocidade de integração de companhias adquiridas)

PUCPR

Abordagens SOA

- A avaliação de um sistema SOA foca a integração dos serviços e os padrões de comunicação, não a arquitetura das aplicações subjacentes
- Abordagens arquiteturais SOA
  - Abordagens de comunicação
  - Abordagem de integração
  - Linguagem de execução de processos de negócio
  - Serviços estáticos ou dinâmicos

PUCPR

BPEL

- Business Process Execution Language
- Padrão usado para descrever processos de negócio orientados a workflow
- Um fluxo coordenado por BPEL define um processo de negócio através de
  - Regras para coordenar o fluxo de dados
  - Interfaces para serviços que o processo provê e usa
  - Meios para tratar condições de exceção
- Existem ferramentas BPEL que permitem o desenvolvimento visual de workflows por programadores de negócio não técnicos

PUCPR

BPEL

- A linguagem BPEL é baseada em XML e possui primitivas como: receive, reply, throw e wait
- O código BPEL é hospedado por um BPEL engine (também chamado BPEL server) e roda no servidor de aplicação
- Quando um evento dispara um workflow, o BPEL engine coordena a invocação de serviços usando o código BPEL como um script

PUCPR

BPEL

- Tipos de processamento possíveis
  - Fluxos seqüenciais de invocação de serviços
  - Invocações paralelas de serviços
  - Correlação request-reply
  - Espera cronometrada
  - Tratamento de erros no processo de negócio
    - Expiração do tempo para entrega de mensagens
    - Synchronous retry ou abort após falha
    - Asynchronous retry compensating transaction após falha
    - Notificação e resolução heurística após falha

PUCPR

Abordagens SOA

- A avaliação de um sistema SOA foca a integração dos serviços e os padrões de comunicação, não a arquitetura das aplicações subjacentes
- Abordagens arquiteturais SOA
  - Abordagens de comunicação
  - Abordagem de integração
  - Linguagem de execução de processos de negócio
  - Serviços estáticos ou dinâmicos

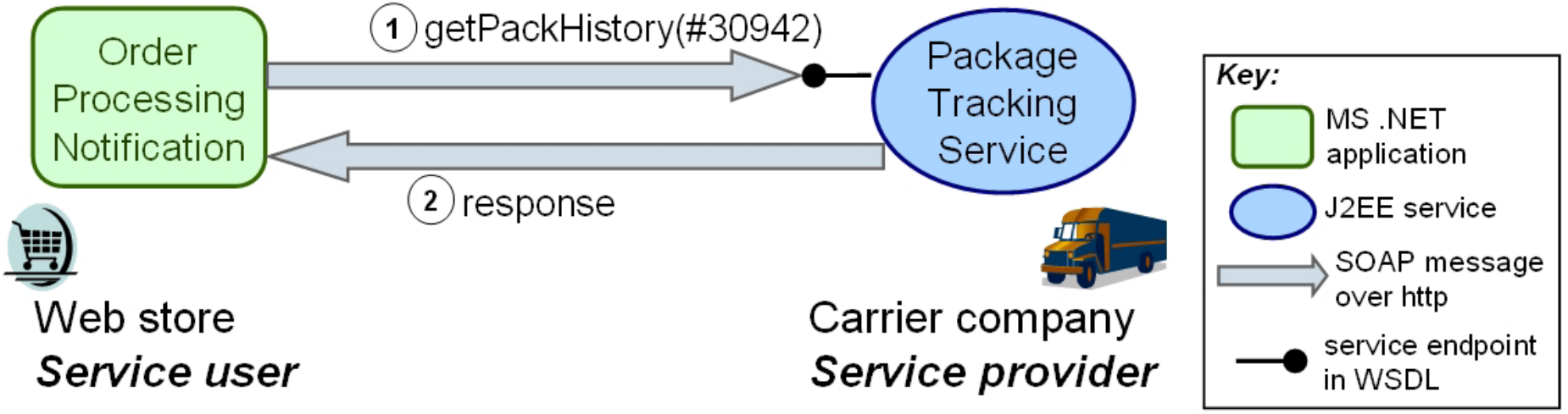
PUCPR

Tipos de Binding

- Para invocar um provedor de serviços é necessário
  - Determinar a interface do serviço: operações disponíveis, entradas e saídas esperadas
  - Localizar o serviço na rede
- Binding estático
  - A interface do serviço e sua localização devem ser conhecidos quando o usuário do serviço é implementado ou instalado
  - Normalmente, o usuário do serviço tem um stub da interface e recupera a localização de um arquivo de configuração local
  - O usuário do serviço pode invocar o provedor diretamente, não envolvendo um diretório público ou privado

PUCPR

Binding Estático



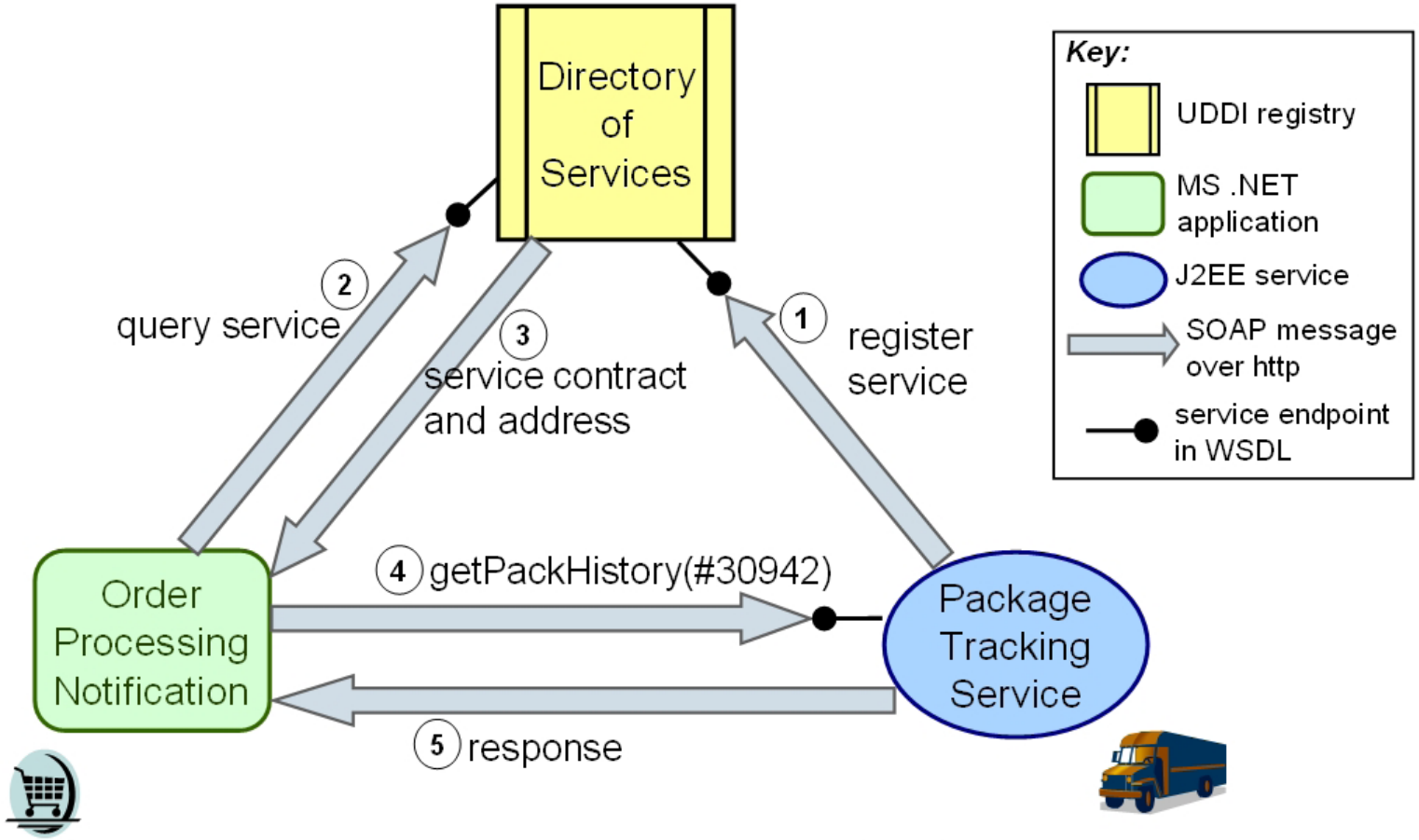
PUCPR

Binding Dinâmico

- O provedor do serviço deve registrá-lo em um diretório de serviços
- O diretório é pesquisado pelo usuário do serviço em tempo de execução a fim de obter o endereço do provedor e o contrato do serviço (interface)
- Depois de obter essas informações, o usuário do serviço pode invocar operações do provedor

PUCPR

Binding Dinâmico



PUCPR

Comparando Bindings

- Binding Estático
  - Acoplamento mais forte entre provedores e usuários
  - Mudança de localização do provedor ou de contrato pode causar parada do usuário quando o serviço é invocado em tempo de execução ou durante o marshalling e unmarshalling dos objetos
  - Melhor performance (evita comunicação com o registry / serviço de diretório)
  - Usado com freqüência porque é suficiente para a maioria dos cenários de negócio

PUCPR

Comparando Bindings

- Binding Dinâmico
  - Acoplamento reduzido entre provedores e usuários
  - A localização do provedor do serviço pode mudar sem afetar os usuários
  - Múltiplas versões das interfaces podem ser gerenciadas pelo serviço de registro (diretório) e coexistir em produção
  - A performance pode ser afetada negativamente pela interação com o serviço de registro (diretório)
  - O overhead pode ser compensado pelo uso do registro para load balancing

PUCPR

Questões SOA

- A arquitetura deve ser analisada para revelar pontos fortes e pontos fracos e descobrir riscos
- Preocupações mais importantes no contexto SOA
  - Plataforma destino
  - Serviços síncronos versus assíncronos
  - Granularidade dos serviços
  - Tratamento de exceção e recuperação após falha
  - Segurança
  - Otimização XML
  - Uso do serviço de registros
  - Integração com sistemas legados
  - BPEL e coordenação de serviços
  - Controle de versões de serviços

PUCPR