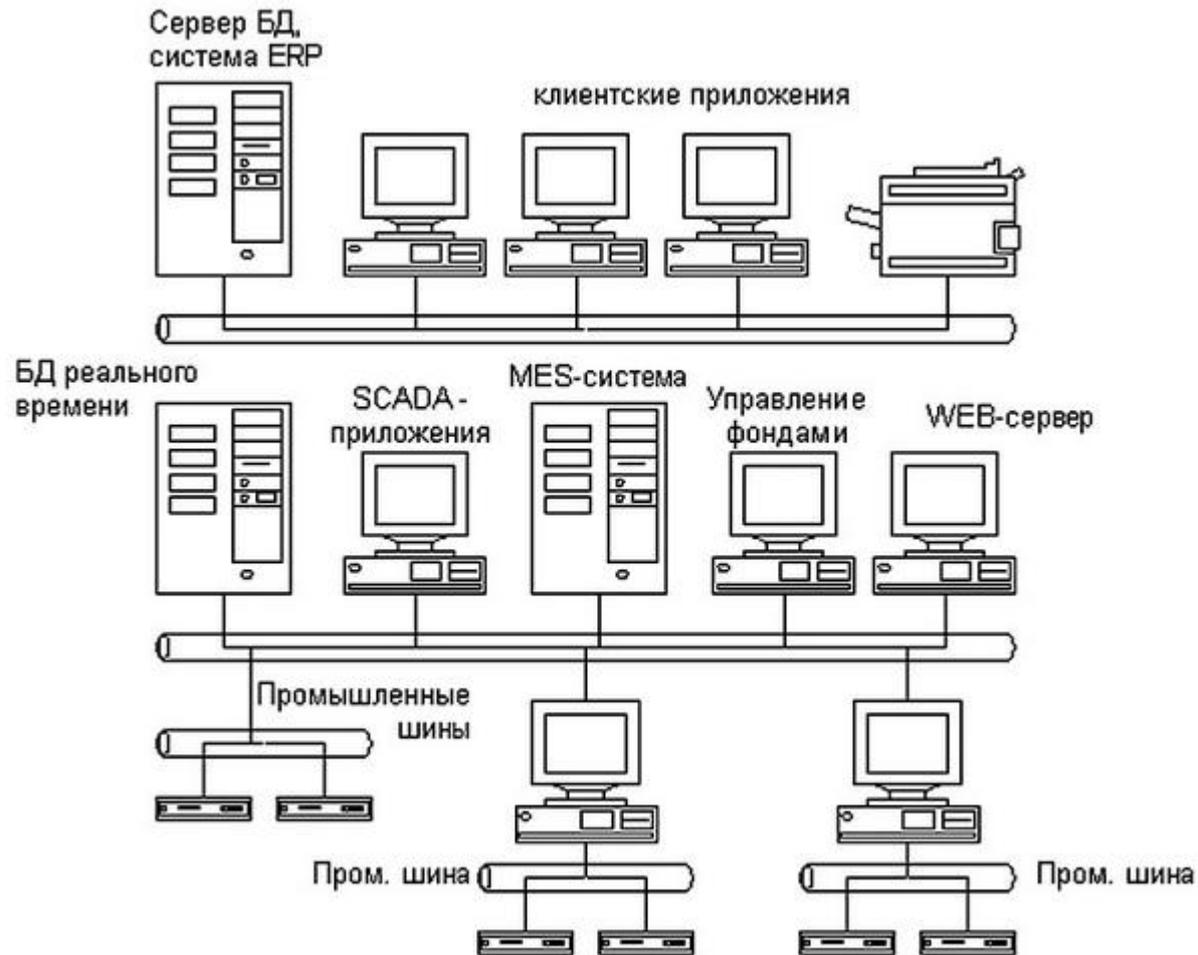


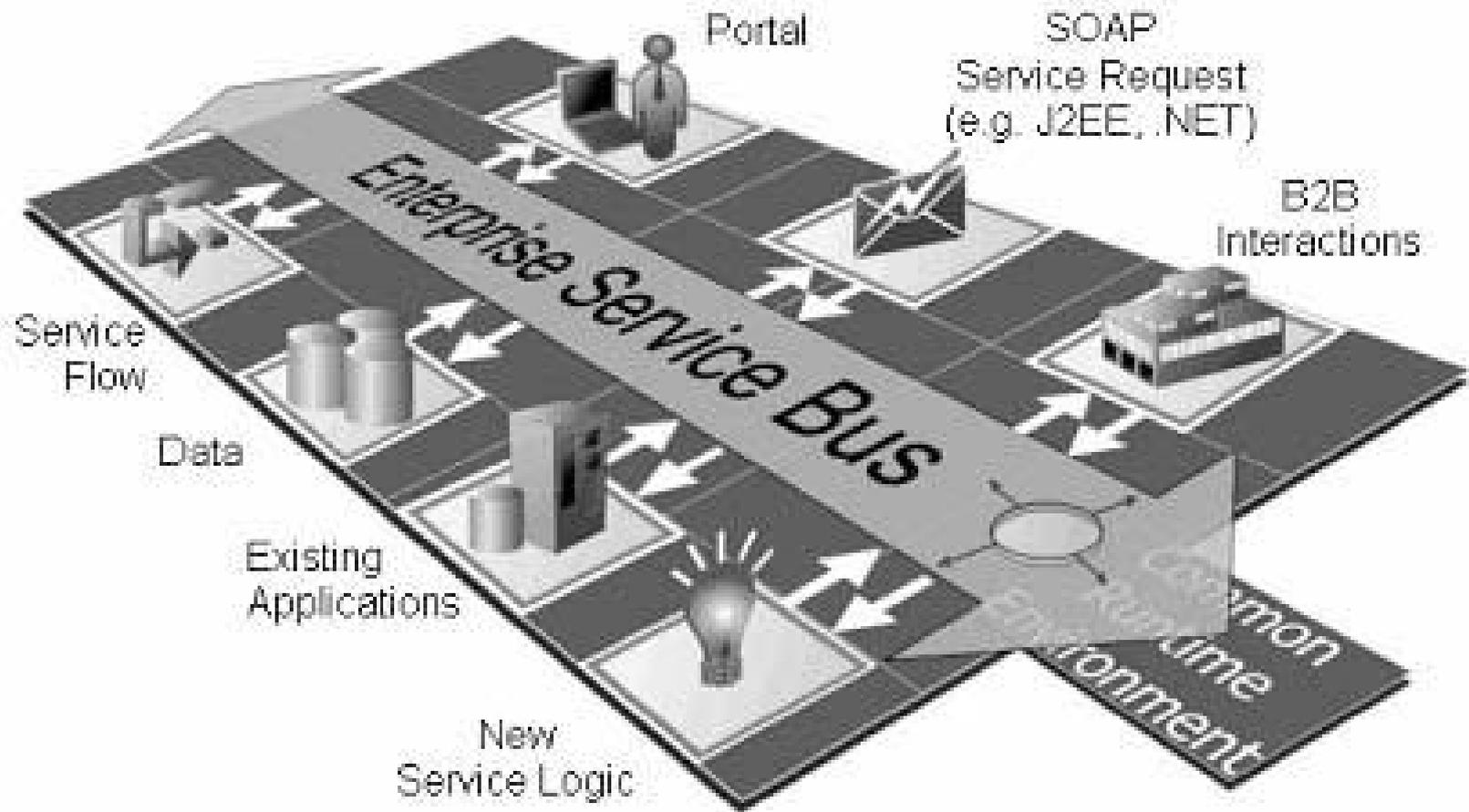
Лекция №3

SOA

Пример



SOA / ESB



Виды интеграции

- 1) Интеграция на уровне данных
- 2) Интеграция на функционально-прикладном и организационном уровнях
- 3) Интеграция на уровне корпоративных программных приложений
- 4) Интеграция при помощи Web-сервисов

SOA

Сéрвис-ориентíрованная архитекту́ра (**service-oriented architecture**) — модульный подход к разработке программного обеспечения, основанный на использовании распределённых, слабо связанных (англ. *low coupling*) заменяемых компонентов, оснащённых стандартизированными интерфейсами для взаимодействия по стандартизированным протоколам.

Архитектура не привязана к какой-то определённой технологии. Она может быть реализована с использованием широкого спектра технологий, включая такие технологии как REST, RPC, DCOM, CORBA или веб-сервисы. SOA может быть реализована, используя один из этих протоколов и, например, может использовать дополнительно механизм файловой системы для обмена данными.

Сервис / Service

Сервис – это видимый ресурс, выполняющий повторяющуюся задачу и описанный внешней инструкцией

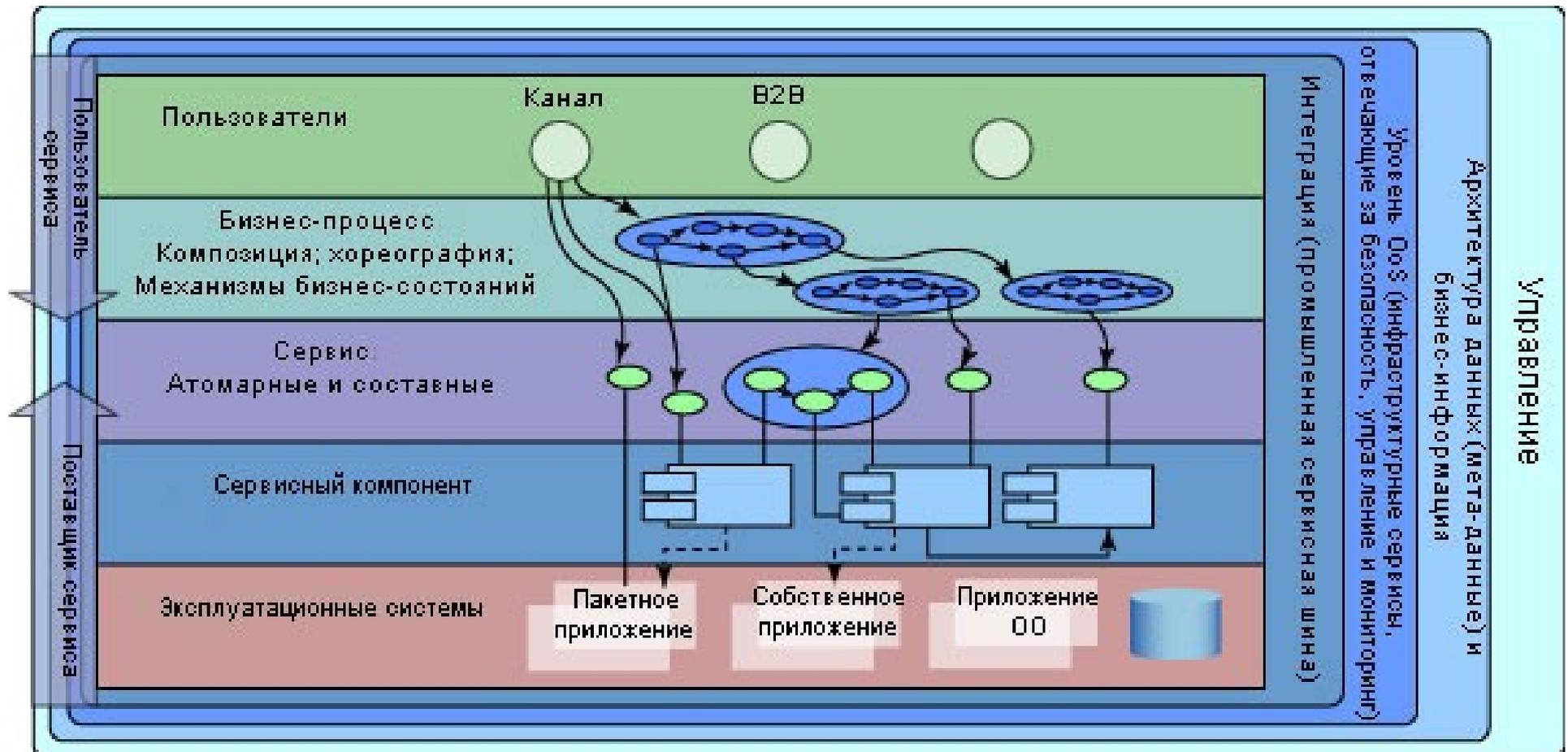
- Выполняет определенную бизнес-логику;
- Самодостаточен;
- Черный ящик для других служб;
- Может использовать или содержать другие службы.

Архитектура

"Сервис-ориентированная архитектура (SOA) – архитектурный стиль для создания ИТ-архитектуры предприятия, использующий принципы ориентации на сервисы для достижения тесной связи между бизнесом и поддерживающими его информационными системами."



Структура решений SOA



ЖЦ SOA



Правила проектирования

- 1) Нужды бизнеса важнее технологических решений
- 2) Стратегические цели важнее преимуществе проекта
- 3) Внутренняя совместимость важнее индивидуальной интеграции/настройки
- 4) Совместные ресурсы важнее узкоспециализированных реализаций
- 5) Гибкость важнее оптимизации
- 6) Эволюционные (инкрементные) изменения важнее первоначального совершенстве реализации

Принципы

- Standardized service contract
- Service reference autonomy (an aspect of loose coupling)
- Service location transparency (an aspect of loose coupling)
- Service longevity
- Service abstraction
- Service autonomy
- Service statelessness
- Service granularity
- Service normalization
- Service composability
- Service discovery
- Service reusability
- Service encapsulation

Программирование

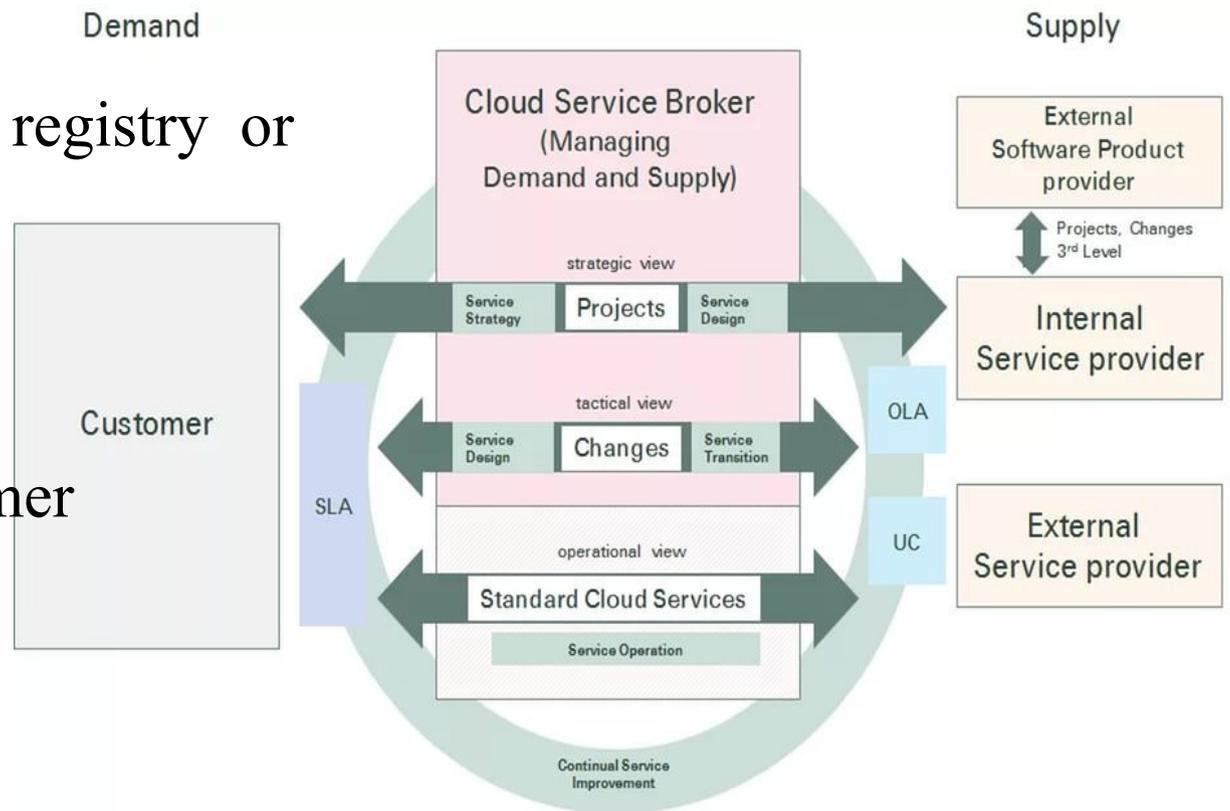


Шаблоны

Service provider

Service broker, service registry or service repository

Service requester/consumer



Реализации

- CORBA
- Web services based on WSDL and SOAP
- Messaging, e.g., with ActiveMQ, JMS, RabbitMQ
- RESTful HTTP, with Representational state transfer (REST) constituting its own constraints-based architectural style
- OPC-UA
- WCF (Microsoft's implementation of Web services, forming a part of WCF)
- Apache Thrift
- SORCER