

## XML и SOA

- XML является фундаментом практически всех стандартов web-сервисов, в том числе XML Schema, SOAP, WSDL (Web Services Description Language) и UDDI (Universal Description, Discovery, and Integration). Эти стандарты опираются на основополагающую концепцию основанных на XML представлений - поддерживаемый во всем мире формат обмена информацией между провайдерами сервисов и инициаторами запросов в SOA.
- Использование XML решает проблему работы с различными форматами данных в различных приложениях, работающих на разных платформах.
- Преимущество XML заключается в простоте представления, являющегося по своей природе текстовым, гибким и расширяемым.

### Примеры стандартов, основанных на XML и используемых в SOA:

- **SOAP.** Этот простой основанный на XML протокол позволяет приложениям обмениваться информацией по транспортным протоколам, таким как HTTP. Благодаря использованию XML протокол SOAP является:
  - Платформенно-независимым.
  - Пригодным для использования в Интернете.
  - Читательным, структурированным и текстовым
- **WSDL.** Это документ, написанный на XML и описывающий web-сервис. Он определяет месторасположение сервиса и отображаемые им операции (или методы), позволяющие обращаться к этому сервису. WSDL-файл описывает четыре главные вещи:
  - Сервисы, доступные через интерфейс web-сервиса, такие как список имен методов и сообщений-атрибутов.
  - Тип данных сообщений.
  - Адрес сервиса, используемый для его вызова

## Реестры сервисов

- Реестр сервисов реализует SOA слабое связывание. Храня месторасположения конечных точек сервисов, он устраняет тесное связывание, приводящее к жесткой привязке потребителя к провайдеру. Он также облегчает потенциальные сложности замены одной реализации сервиса другой при необходимости.
- Реестр сервисов позволяет системным аналитикам исследовать корпоративный портфель бизнес-сервисов. Исходя из этого, они

могут определить, какие сервисы доступны для автоматизации процессов с целью удовлетворения актуальных бизнес-потребностей, а какие нет. Это в свою очередь позволяет узнать, что нужно реализовать и добавить в портфель, формируя каталог доступных сервисов.

- Реестр сервисов может выполнять функцию управления сервисами, обязывая подписывающиеся сервисы быть согласованными. Это помогает гарантировать целостность руководства (governance) сервисами и стратегий. Дополнительная информация о руководстве и важности SOA приводится далее в данном руководстве.

## Бизнес-процесс

**Бизнес-процесс** — набор взаимосвязанных задач, относящихся к деятельности, имеющей функциональные границы. Бизнес-процессы имеют начальные и конечные точки и являются повторяемыми.

### Элементы бизнес-процесса:

- Входные данные (input) — информация необходимая для получения результат;
- Выходные данные (output) — все данные полученные в качестве результата;
- События (events) — уведомления чего-то важного;
- Подпроцесс (subprocess) — мелкий процесс или этап в рамках процесса;
- Действие — элемент работы в процессе;
- Показатели производительности (performance metrics) — атрибуты эффективности процесса.

## Управление транзакциями

Для решения проблемы транзакций был разработан ряд спецификаций web-сервисов. К ним относятся:

- **WS-Coordination.** Позволяет зарегистрированным процессам принимать участие в создании общего контекста, ответственного за хранение текущих данных и распространяемой между ними информации.
- **WS-AtomicTransaction.** Используется в краткосрочных распределенных действиях.
- **WS-BusinessActivity.** Этот протокол используется с долго работающими транзакциями.

## Составляющие базовой архитектуры SOA



- **Провайдер сервиса.** Предоставляет сервисы, контракт по активизации которых и месторасположение опубликованы.
- **Потребитель сервиса.** Потребляет сервисы, соответствующие его бизнеспотребностям и обнаруженные в каталоге сервисов.
- **Каталог сервисов.** Служит для публикации и ведения списка сервисов, доступных для потребителей.

## Роль ESB в архитектуре SOA

**Enterprise Service Bus (сервисная шина предприятия)** — подход к построению распределённых корпоративных информационных систем.

**Роли ESB в информационной системе:**

1. Предоставляет интеграционную инфраструктуру, соответствующую принципам SOA:
  - a. Устанавливает явные независимые от реализации интерфейсы для организации слабого связывания.
  - b. Использует коммуникационные протоколы, независимые от расположения взаимодействующих сторон.
  - c. Способствует определению сервисов, инкапсулирующих повторно используемые бизнес-функции.
2. Предоставляет средства для управления инфраструктурой сервисов.
3. Функционирует в распределенной гетерогенной среде через поддержку синхронных и асинхронных взаимодействий, а также использование стандартных интерфейсов.
4. Централизует управление и распределяет обработку.
5. Реализует защиту и обеспечение качества сервиса в проектах SOA.

**Недостатки ESB:**

- Требуется достаточно больших трудозатрат и специфических знаний для реализации, при этом сама по себе (без дальнейшей реализации SOA) практически не приносит ощутимой пользы для бизнеса;

- По сравнению с простейшей (точка-точка) интеграцией между системами, вносит задержки, связанные с преобразованием XML-сообщений.
- Требуется тщательного продумывания и контроля над версионностью сообщений, в противном случае может увеличить связность систем друг с другом (при недостаточной унификации сообщений).

## Оркестровка и хореография

**Оркестровка в бизнес-процессах** — это серия действий в управляемом потоке работ, обычно имеющем одну линию выполнения.

**Хореография** — видимый обмен сообщениями, правила взаимодействия и соглашения между двумя и более сервисами.

**Ключевые элементы проектирования:**

- Для **оркестровки**: участник и его роль, переменные и свойства, определяющие взаимодействие участников, обработчики ошибок, события.
- Для **хореографии**: структура сообщений, асинхронная и синхронная коммуникация сервисов, служебные сообщения



Оркестровка отличается от хореографии тем, что она описывает процесс, протекающий между сервисами, контролируемый основным участником. В хореографии нет участника, ведущего обмен сообщениями.

**Технические требования для оркестровки и хореографии:**

1. Гибкость;
2. Простые и структурированные действия;
3. Рекурсивная композиция.

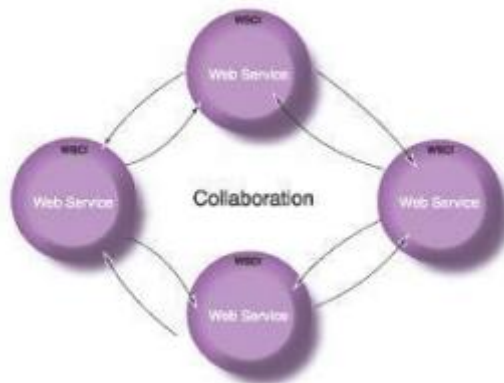
**Дополнительно, оркестровка и хореография предъявляют требования к целостности и стабильности взаимодействий. Они включают:**

1. Хранение состояний и корреляция запросов. Способность хранить состояние между запросами веб-сервисов особенно важно, когда

работа ведется с асинхронными сервисами. Язык и инфраструктура должны обеспечивать хранение данных и корреляцию запросов для построения диалогов более высокого уровня.

2. Обработка исключений и транзакции. Долго выполняемые сервисы должны обеспечивать транзакционную целостность и управление исключениями.

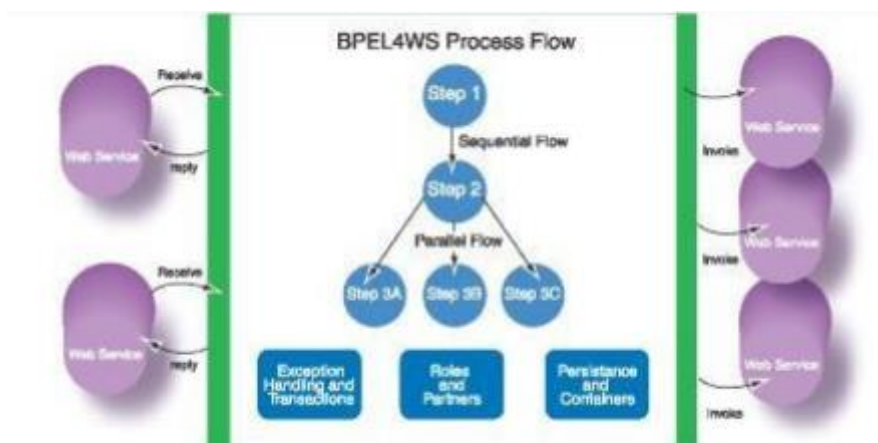
## WSCI



## BPEL4WS

**Бизнес-протокол** поддерживает публичный обмен сообщениями между участниками обмена. Его нельзя выполнить и он не определяет внутреннее выполнение процесса.

**Выполняемый процесс** моделирует выполнение действий. Он обеспечивает оркестровку, в то время как бизнес-протоколы сфокусированы на хореографии.



**Процесс, определенный в BPEL4WS, состоит из:**

- Действий (activities);
- Ссылок на партнеров;

- Переменных;
- Корреляционных наборов (correlation sets);
- Обработчиков неисправностей (fault handlers);
- Обработчиков событий (event handlers);
- Корректирующих обработчиков (compensation handlers).