# Глава 1. Основные понятия

**Windows Communication Foundation (WCF)** – это унифицированная модель программирования распределенных приложений на платформе Microsoft.

Технология Windows Communication Foundation (WCF) – это службы.

В основе своей **служба** – это множество оконечных точек (endpoints), которые предоставляет клиентам некие полезные возможности.

**Оконечная точка** – это просто сетевой ресурс, которому можно посылать сообщения.

Чтобы воспользоваться предоставляемыми возможностями, клиент посылает сообщения оконечым точкам в формате, который описывается контрактом между клиентом и службой. Службы ожидают поступления сообщений на адрес оконечной точки, предполагая, что сообщения будут записаны в оговоренном формате.

**REST**

Для приложенийпотребителей с 2008 года превалирует интерфейс к Web службам на основе стандарта REST (Representational Entity State Transfer), который сочетает протокол HTTP и четко определенную схему составления URI для адресации XML данных. REST предполагает, что манипуляции с данными описываются паттерном CRUD (Create Read Update Delete – «создать читать обновить удалить»); основной отличительной особенностью REST является простота.

**SOAP**

Для бизнесприложений основным интерфейсом к Web-службам в 2008 году стал протокол SOAP (Simple Object Access Protocol). Он обеспечивает наиболее надежную модель обмена сложными данными. Сообщения SOAP состоят из конверта и тела, поэтому их можно зашифровать и безопасно передать по сети Интернет. Если сообщение является частью логического сеанса или транзакции, то эта семантика прописывается в конверте и распространяется вместе с сообщением. Если информацию необходимо защитить, то тело сообщения шифруется, а информация о способе защиты помещается в конверт. Сообщения SOAP строго типизированы, что дает дополнительное удобство для разработчика. Как и в случае REST, сообщения SOAP обычно передаются по протоколу HTTP в текстовом формате. WCF безразлична к протоколу передачи и формату сообщения.

**АПК: адрес, привязка и контракт**

* **«А» обозначает адрес**, то есть «куда». Адрес определяет, куда следует отправлять сообщения, чтобы оконечная точка их получила
* **«П» обозначает привязку**, то есть «как». Привязка определяет канал для коммуникаций с оконечной точкой. По каналам передаются все сообщения, циркулирующие в приложении WCF. Канал состоит из нескольких элементов привязки (binding element). На самом нижнем уровне элемент привязки – это транспортный механизм, обеспечивающий доставку сообщений по сети. Элементы привязки, расположенные выше, описывают требования к безопасности и транзакционной целостности.
* **«К» обозначает контракт**, то есть «что». Контракт определяет набор функций, предоставляемых оконечной точкой, то есть операции, которые она может выполнять, и форматы сообщений для этих операций. Описанные в контракте операции отображаются на методы класса, реализующего оконечную точку, и включают в частности типы параметров, передаваемых каждому методу и получаемых от него.

Чтобы службы можно было найти, предусмотрена специальная инфраструктурная оконечная точка, называемая **Metadata Exchange (MEX)** (обмен метаданными). Клиенты могут обращаться к ней, если хотят получить описания АПК службы на языке Web Service Description Language (WSDL).

**System.ServiceModel** - пространство имен устанавливается вместе с .NET 3.0 и содержит большую часть кода WCF.

**Хостинг служб:**

* IIS
* Self-hosting
  + App
  + Windows service

# Глава 2. Контракты

**Контракт** – это соглашение между двумя или более сторонами о формате сообщений, которыми они обмениваются. Контракт представляет собой описание сообщений, передаваемых оконечным точкам службы и возвращаемых ей. Каждая оконечная точка определяется своими АПК.

В WCF есть контракты трех видов:

* **Контракт о службе** описывает функциональные операции, реализуемые службой. Он отображает методы класса .NET на описания служб, типов портов и операций на языке WSDL. Внутри контракта о службе имеются контракты об операциях, которые описывают отдельные операции службы, то есть методы, реализующие ее функции.
* **Контракт о данных** описывает структуры данных, используемые службой для взаимодействия с клиентами. Контракт о данных отображает типы CLR на определения на языке XML Schema Definitions (XSD) и определяет, как их следует сериализовывать и десериализовывать. Он описывает все данные, получаемые и отправляемые операциями службы.
* **Контракт о сообщениях** отображает типы CLR на сообщения протокола SOAP и описывает формат последних, что находит отражение в определениях сообщений на языках WSDL и XSD. Контракт о сообщениях позволяет точно контролировать состав заголовков и тел SOAP-сообщений.

Чтобы контракты были интероперабельны с максимально широким диапазоном систем, они выражаются на языке WSDL (Web Service Description Language – язык описания Web-служб).

Консорциум W3C – организация, устанавливающая стандарты для веб (в нее входят в частности компании Microsoft, IBM и другие.

Типы операций:

* **Синхронные операции запрос-ответ** - клиент посылает запрос на сервер и дождаеться пока сервер обратотает запрос и вернет результат.
* **Асинхронные операции запрос-ответ** - реализуется на стороне клиента посредством использования "Синхронных операций" и много поточности.
* **Односторонние операции** -клиент посылает запрос на сервер, дожидается отчета о доставке запроса и не ожидает результатов работы сервера. Реализуется на стороне сервера.
* **Дуплексные операции** - между клиентом и службой создается двунаправленный канал свзяки (в стучае HTTP два канала в обе стороны) по которому клиент може отсылать сообщения службе и в ту же очередь служба может отсылать сообщения клиенут.

**Изменения контрактов о данных**

* Не нарушают совместимость с существующими клиентами:
  + Добавление новых необязательных членов;
  + Удаление старых необязательных членов.
* Нарушают совместимость с существующими клиентами:
  + Изменение имени или пространства имен контракта о данных;
  + Переименование существующего обязательного члена данных;
  + Добавление нового члена данных с именем, которое раньше уже было задействовано;
  + Изменение типа существующего члена данных;
  + Добавление новых членов с признаком IsRequired=true в атрибуте DataMemberAttribute;
  + Удаление существующих членов с признаком IsRequired=true в атрибуте DataMemberAttribute.

Контракты о сообщениях описывают структуру SOAP-сообщений, получаемых и отправляемых службой, и позволяет просматривать и управлять большинством деталей заголовка и тела SOAP. Если контракты о данных обеспечивают возможность интероперабельности по стандарту XML Schema Definition (XSD), то контракты о сообщениях позволяют общаться с любой системой, которая поддерживает спецификацию SOAP. Нельзя смешивать программирование на основе сообщений и на основе параметров, поэтому невозможно передать операции в качестве входного аргумента DataContract и ожидать, что она вернет MessageContract, или наоборот. Можно смешивать типизированные сообщения с не типизированными, но не контракты о данных с контрактами о сообщениях. Если вы попытаетесь это сделать, то получите исключение при генерации WSDL-документа службой.

# Глава 3. Каналы

**Стек каналов** – это многоуровневый коммуникационный стек, составленный из одного или нескольких каналов, обрабатывающих сообщения. Каналы бывают транспортными и протокольными. Транспортные каналы располагаются внизу стека и отвечают за передачу сообщений с помощью некоторого транспортного механизма (например, HTTP, TCP, MSMQ). Протокольные каналы реализуют различные протоколы (безопасности, надежной доставки, транзакционности и т.д.) посредством трансформации и модификации сообщений.

Фабрики и прослушиватели каналов лежат в основе приема и передачи сообщений. Они отвечают за создание стека каналов и предоставление его приложениям.

WCF умело скрывает детали модели каналов от разработчиков. Большинству разработчиков достаточно класса, производного от ClientBase<>, для отправки сообщений и класса ServiceHost для размещения служб. Эти классы надстроены над архитектурой каналов.

Архитектура каналов лежит в основе всех коммуникаций в WCF. Разобравшись с ее составными частями – стеками каналов, каналами, фабриками и прослушивателями каналов, – разработчик может расширить модель коммуникаций или подстроить ее под свои нужды.

# Глава 4. Привязки

Привязкой называется заранее сконфигурированный стек каналов. Привязки описывают соглашения между клиентом и сервером о порядке передачи данных по сети. В привязке задается способ транспортировки, кодирование и протоколы, участвующие в коммуникации. WCF с помощью привязок инкапсулирует конфигурацию в различных сценариях коммуникации. Для наиболее распространенных сценариев – Web-служб, служб REST/POX и приложений на основе очередей – имеются уже готовые привязки. Например, привязка basicHttpBinding предназначена для работы с Web-службами, созданными в ASP.NET или совместимыми со спецификацией WS-I Basic Profile 1.1. Привязки ws2007HttpBinding и wsHttpBinding похожи на basicHttpBinding, но поддерживают больше возможностей, в частности надежную доставку и транзакции, а также основаны на более современных стандартах, таких, как WS-Addressing

**Привязки, начинающиеся с префикса net, предназначены для коммуникации между .NET-приложениями.**

**Привязки, начинающиеся с префикса «ws», следует использовать, когда необходимо реализовать интероперабельность с помощью Web-служб.**

Для создания заказной привязки достаточно указать транспортный протокол и кодировщик. Кодировщик можно опускать, если для транспортного протокола имеется кодировщик по умолчанию.

# Глава 5. Поведения

Поведения – это классы, которые влияют на работу WCF на этапе выполнения. Они вызываются, когда инициализируется исполняющая среда WCF на стороне клиента или сервера, а также в процессе передачи сообщений между ними. Поскольку поведения работают в критически важные моменты, их можно использовать для реализации многих встроенных в WCF функций. Кроме того, они представляют собой важный механизм расширения WCF.

Типы поведений (**Behavior**)

* **Поведения службы** работают на уровне службы и имеют доступ ко всем оконечным точкам. Они управляют такими вещами, как создание экземпляров и транзакции. Поведения службы могут также использоваться для авторизации и аудита.
* **Поведений оконечной точки** ограничена одной оконечной точкой службы. Они хорошо приспособлены для инспектирования входящих и исходящих сообщений и выполнения над ними тех или иных действий.
* **Поведения операции** функционируют на уровне одной операции и подходят для сериализации, управления потоком транзакций и обработки параметров операции.
* **Поведения обратного вызова**, которые работают аналогично поведениям службы, но контролируют оконечные точки, созданные на стороне клиента в процессе дуплексной коммуникации.

Поведение InstanceContextMode применяется для управления созданием экземпляров службы:

* **Single**. Один экземпляр класса службы обрабатывает все входящие запросы. Тем самым реализуется паттерн Singleton (синглет, одиночка);
* **PerCall**. На каждый входящий запрос создается отдельный экземпляр класса службы.
* **PerSession**. Один экземпляр класса службы создается на каждый сеанс связи с клиентом. При использовании безсеансовых каналов, все вызовы службы ведут себя как в случае PerCall, даже если для поведения InstanceContextMode установлено значение PerSession.

Поведение службы ConcurrencyMode применяется для управления параллелизмом внутри одного экземпляра службы. Принимаемое по умолчанию значение ConcurrencyMode.Single.

* **Single**. В каждый момент времени к классу службы может обращаться только один поток. Это самый безопасный режим, поскольку операции службы могут не заботиться о безопасности относительно потоков.
* **Reentrant**. В каждый момент времени к классу службы может обращаться только один поток, но этот поток может покидать класс, а затем снова возвращаться в него.
* **Multiple**. К классу службы одновременно могут обращаться несколько потоков. В этом случае класс должен быть написан с учетом безопасности относительно потоков.

**Транза́кция** (англ. transaction, от лат. transactio — соглашение, договор) — минимальная логически осмысленная операция, которая имеет смысл и может быть совершена только полностью.

ACID-транзакции обладают следующими свойствами:

* **Атомарность (Atomic).** Либо все обновления, включенные в транзакцию, выполнены успешно, либо вообще не выполнены. Частичные обновления не допускаются. Например, если при переводе денег с одного банковского счета на другой дебетование выполнено успешно, а кредитование завершилось с ошибкой, то операция дебетования откатывается, чтобы деньги не появлялись из ничего и не уходили в никуда.
* **Непротиворечивость (Consistent).** После завершения операции все данные оказываются корректны с точки зрения бизнесправил. Например, в случае перевода денег исходный и конечный счета должны быть допустимы, иначе транзакция отменяется.
* **Изолированность (Isolated).** Во время выполнения операции промежуточные результаты обновления не видны вне самой транзакции. Например, при переводе денег никакие другие клиенты не должны видеть частично обновленных балансов.
* **Долговечность (Durable).** После того как транзакция зафиксирована, данные должны быть сохранены так, чтобы не терялись при последующих сбоях в системе.

# Глава 7. Размещение

**Владельцем** службы называется процесс операционной системы, отвечающий за время ее жизни и необходимый контекст. Он запускает и останавливает службу, а также предоставляет базовые функции управления. Помимо этого, владелец почти ничего не знает о WCF-службе, которая работает в его адресном пространстве.

**Способы размещения служб:**

* Internet Information Services (IIS)
* Windows Process Activation Services (WAS)
* Service Control Manager (SCM)
* Внутри оконного или консольного приложения

# Глава 8. Безопасность

Основные составляющие безопасности:

* Аутентификация
* Авторизация
* Конфиденциальность
* Целостность

**Аутентифика́ция** (англ. authentication от греческого : αὐθεντικός authentikos, "реальный, подлинный, " от αὐθέντης authentes, «автор») — процедура проверки подлинности, например:

* проверка подлинности пользователя путём сравнения введённого им пароля с паролем, сохранённым в базе данных пользователей;
* подтверждение подлинности электронного письма путём проверки цифровой подписи письма по открытому ключу отправителя;
* проверка контрольной суммы файла на соответствие сумме, заявленной автором этого файла.

**Авториза́ция** (англ. authorization — разрешение, уполномочивание) — предоставление определённому лицу или группе лиц прав на выполнение определённых действий; а также процесс проверки (подтверждения) данных прав при попытке выполнения этих действий.

Под **конфиденциальностью** понимается механизм, предотвращающий чтение информации, которой обмениваются клиент и служба, посторонними лицами. Обычно это достигается за счет шифрования, и WCF пре доставляет для этого разнообразные средства.

**Целостность** – механизм который гарантируеть, что сообщение не было изменено в процессе передачи от клиента службе и наоборот. Обычно, чтобы добиться этого, отправитель включает в сообщение цифровую подпись или подписанную свертку сообщения (дайджест), а получатель проверяет подпись, сверяя ее с содержимым полученного сообщения. Если результат вычисления подписи на стороне получателя не совпадает со значением, которое пришло в составе сообщения, сообщение надлежит отбросить. Отметим, что целостность можно обеспечить даже тогда, когда конфиденциальность не нужна. Иногда допустимо посылать информацию в открытом виде при условии, что получатель способен проверить ее неизменность с помощью цифровой подписи

Категории безопасности:

* **Безопасность на уровне транспорта.** Защита данных при передаче по сети, обеспечивает защиту отправляемых данных безотносительно к их содержимому. Обычно для этой цели применяют протокол Secure Sockets Layer (SSL), с помощью которого шифруются и подписываются пакеты, передаваемые по протоколу HTTPS. Если в сети имеется дополнительные узлы то нельзя быть уверенным в том что соединение будет безопасно на всех участках сети. При использовании сертификатов необходимо их регистрировать в системе.
* **Безопасность на уровне сообщения.** Способ защиты самого сообщения вне зависимости от механизма его транспортировки.

**Сертификаты** и представляемые с их помощью утверждения – это безопасный универсальный метод доказательства подлинности

Законспектировать моменты шифрования на транспортном уровне и на уровне сообщений. Так же выделить разницу, приэмущества и недостатки.

Дать кракткость по в сетях Интранет и Интернет

Написать примера аудита – и вообще понять зачем оно ножуно. В конуце главы это говнецо