

ГЛОССАРИЙ WEB-СЛУЖБ

Составлен рабочей группой W3C 11 февраля 2004 г.

* * *

Данная версия:

<http://www.w3.org/TR/2004/NOTE-ws-gloss-20040211/>

Последняя версия:

<http://www.w3.org/TR/ws-gloss/>

Предыдущая версия:

<http://www.w3.org/TR/2003/WD-ws-gloss-20030808/>

Аннотация

Настоящий документ представляет собой глоссарий терминов, касающихся Web-служб, которые определяются и используются в Архитектуре Web-служб (Web Services Architecture) [WS Arch]. Документ предназначен для использования единообразных, согласованных терминов в спецификациях Web-служб

Статус этого документа

В данном разделе описывается статус этого документа на момент его публикации. Документ может быть заменен другими документами. Список текущих публикаций W3C и самую последнюю версию этого технического отчета можно найти в [W3C technical reports index](http://www.w3.org/TR/) в <http://www.w3.org/TR/>.

Представляемый глоссарий является официальной Запиской Рабочей Группы ([Working Group Note](#)). Он был создан Рабочей группой W3C по Архитектуре Web-служб ([W3C Web Services Architecture Working Group](#)), которая является частью Активности W3C по Web-службам ([W3C Web Services Activity](#)). Данная публикация Записки совпадает по времени с окончанием периода действия устава Рабочей Группы.

Для обсуждения этого документа можно обращаться по адресу: www-ws-arch@w3.org ([public archives](#)).

Публикация документа в качестве Записки Рабочей Группы не предполагает его подтверждения членами W3C. Публикуемый документ является проектом глоссария и в любой момент может быть обновлен или заменен другими документами.

Содержание

Аннотация	2
Статус этого документа	2
Содержание	3
1 Введение	4
2 Определения	4
3 Ссылки	21

1 Введение

Документ содержит список терминов, относящихся к Web-службам и принятых в Архитектуре Web-служб [\[WS Arch\]](#). В терминах используются заглавные буквы, когда это имеет значение, в остальных случаях употребляются строчные буквы.

Некоторые определения в этом документе взяты дословно из внешних документов. В таких случаях указывается ссылка на источник. Ссылки перечислены в разделе 3 - [Ссылки](#)

2 Определения

access – доступ

Возможность взаимодействия с системной сущностью (system entity) с целью ее обработки, использования, извлечения сведений и/или получения представления о некоторых или всех ресурсах системной сущности. [\[RFC 2828\]](#)

access control - управление доступом

Защита ресурсов от неавторизованного доступа; процесс, посредством которого использование ресурсов регулируется в соответствии с политикой безопасности ([security policy](#)) и разрешается только авторизованным системным сущностям ([system entity](#)), [\[RFC 2828\]](#).

access control information - информация для управления доступом

1. Любая информация, используемая в целях управления доступом ([access control](#)), включая контекстную информацию. [\[X.812\]](#)
2. Контекстная информация может содержать исходные IP-адреса, шифры, тип запрашиваемой операции, время суток и т.д. Часть информации для управления доступом ([access control](#)) может относиться к самому запросу, часть - к связи ([connection](#)), посредством которой запрос передается, часть к "среде" (например, время суток) [\[RFC 2829\]](#).

access rights - права доступа

Описание типов авторизованных взаимодействий, которые субъект может иметь с ресурсом. Например, это чтение, запись, исполнение, добавление, модификация и удаление. [\[WSIA Glossary\]](#)

actor - актер

1. Человек или организация ([person or organization](#)) - которые могут быть владельцами агентов ([agents](#)), запрашивающих или поставляющих Web-службы ([Web services](#)).
2. Физический или концептуальный объект, который может выполнять какие-то действия, например, люди, компании, машины, работающее программное обеспечение. Актер может выполнять одну или несколько ролей. Актер на одном

уровне абстракции может рассматриваться как роль на более низком уровне абстракции.

agent - агент

Программа, действующая от имени человека или организации ([person or organization](#)) (Это определение является конкретизацией определения в [\[Web Arch\]](#) и соответствует понятию агента программного обеспечения в [\[Web Arch\]](#)).

anonymity - анонимность

Свойство или состояние ([state](#)), при котором имя или идентичность неизвестны или скрыты. [\[RFC 2828\]](#)

architecture - архитектура

1. Архитектура системы программного обеспечения или вычислительной системы - это структура или структуры этой системы. Такая структура включает программные компоненты, визуализированные свойства этих компонент, отношения между компонентами и ограничения на их использования (это определение основано на определении архитектуры, данном в [\[Soft Arch Pract\]](#)).

2. Архитектура программного обеспечения - это абстракция элементов системы программного обеспечения в течение некоторой фазы ее работы. Система может состоять из многих уровней абстракции и многих фаз работы, и каждый уровень и каждая фаза могут иметь свою архитектуру. [\[Fielding\]](#)

artifact - артефакт

Порция цифровой информации. Артефакт может иметь любой объем и может состоять из других артефактов. Примеры артефактов: сообщение, URI (Uniform Resource Identifier), XML-документ, PNG-изображение, поток битов.

asynchronous - асинхронный

О взаимодействии говорится как об асинхронном, когда ассоциируемые сообщения хронологически и процедурно развязаны. Например, во взаимодействии "запрос - ответ" агент клиента может обрабатывать ответ в некоторый неопределенный момент в будущем, когда обнаружится, что этот ответ существует. К механизмам реализации такого взаимодействия относятся опрос, уведомление о получении другого сообщения и т.д..

attribute - атрибут

Отличительная характеристика объекта. Говорится, что атрибуты объекта описывают объект. Атрибуты реальных объектов часто специфицируются в терминах физических характеристик объектов, таких как размер, форма, вес, цвет и т.д. Атрибуты объектов киберпространства могут описывать их размер, способ кодирования, сетевой адрес и т.д. [\[WSIA Glossary\]](#)

audit guard - защита аудита

Используемый от имени владельца механизм, который позволяет контролировать действия и агентов ([agents](#)) для подтверждения выполнения обязательств ([obligations](#)).

authentication - аутентификация

Процесс проверки и подтверждения, что потенциальный партнер по общению действительно представляет человека или организацию ([person or organization](#))

authorization – авторизация

Процесс определения по соответствующей информации для управления доступом ([access control information](#)), разрешены ли для субъекта указанные виды доступа к *конкретному* ресурсу. Обычно авторизация выполняется в контексте аутентификации. Если субъект аутентифицирован, он может быть авторизован для выполнения различных видов доступа. [[STG](#)]

binding - связывание

1. Связь между интерфейсом ([interface](#)), конкретным протоколом ([protocol](#)) и форматом данных. Связывание определяет, какой протокол и какой формат данных нужно использовать для передачи сообщений, определенных связанным интерфейсом [[WSD Reqs](#)].
2. Установление соответствия между интерфейсом ([interface](#)) и связанными с ним операциями ([operations](#)), с одной стороны, и форматом конкретного сообщения и протоколом его передачи ([protocol](#)), с другой стороны.
3. См. также [SOAP binding](#).

capability - возможность, способность

Поименованная часть функциональности (или свойство), которая объявлена как поддерживаемая или запрашиваемая агентом ([agent](#)).

choreography – хореография

1. Хореография определяет последовательность и условия обмена сообщениями между многочисленными взаимодействующими независимыми агентами ([agent](#)). При помощи этого обмена выполняется задача достижения целевого состояния.
2. Хореография Web-служб касается взаимодействия служб с их пользователями. Любой пользователь Web-службы, автоматический или нет, является клиентом этой службы. В качестве пользователей могут выступать другие Web-службы, приложения или люди. Транзакции между Web-службами и их клиентами должны быть четко определены во время их выполнения и могут состоять из множества отдельных взаимодействий, чья композиция составляет полную транзакцию. Эта

композиция, ее протоколы передачи сообщений, интерфейсы, установление последовательности и ассоциированная логика и составляет то, что называется хореографией. [\[WSC Reqs\]](#)

component - компонента

1. Объект программного обеспечения, предназначенный для взаимодействия с другими компонентами, инкапсулирующий некоторую функциональность или набор функциональностей. Компонента имеет четко определенный интерфейс и ведет себя заранее определенным образом, общим для всех компонент в рамках архитектуры [\[CCA T&D\]](#).
2. Абстрактная единица программного обеспечения и внутреннего состояния, которая обеспечивает преобразования данных через свой интерфейс [\[Fielding\]](#).
3. Единица архитектуры с определенными границами.

confidentiality - конфиденциальность

Гарантия обеспечения секретности информации с разрешением доступа ([access](#)) только соответствующим лицам [\[NSA Glossary\]](#).

configuration - конфигурация

Совокупность свойств, которые могут меняться. Свойство может влиять на поведение сущности.

connection - соединение

Виртуальный канал *связи* транспортного уровня, установленный между двумя программами для их коммуникации [\[RFC 2616\]](#).

control - управление

Цель управления - вызвать желаемое изменение состояния. Системы управления могут контролировать жизненный цикл управляемых Web-служб ([manageable Web services](#)) или поток информации, например, сообщений.

conversation - диалог

Диалог Web-службы касается поддержания некоторого состояния во время взаимодействия, затрагивающего многочисленные сообщения ([message](#)) и/или участников.

credentials - удостоверения

Данные, которые передаются для установления требуемой идентичности принципала ([principal](#)) [\[X.800\]](#).

delivery policy - политика доставки

Политика ([policy](#)), которая определяет ограничения для методов доставки сообщений ([message](#)).

digital signature - цифровая подпись

Значение, которое вычисляется по криптографическому алгоритму и присоединяется к объекту данных таким образом, что любой получатель этих данных может использовать эту подпись для проверки источника и целостности данных. (См data origin authentication service, data integrity service, digitized signature, electronic signature, signer в [\[RFC 2828\]](#)).

discovery - обнаружение

Определение местонахождения машинно-обрабатываемого описания ресурса, связанного с Web-службой ([Web service](#)) Ресурс должен отвечать некоторым функциональным критериям и может быть заранее неизвестен. Поиск описания выполняется путем сопоставления набора функциональных и других критериев с набором описаний ресурсов. Цель поиска - найти соответствующий ресурс, связанный с Web-службой.

discovery service - служба обнаружения

Служба ([service](#)), позволяющая агентам находить описания ресурсов, связанных с Web-службами ([Web service](#)).

document - документ

Любые данные, которые могут быть представлены в цифровой форме [\[UeB Glossary\]](#).

domain - домен

Идентифицированное множество агентов ([agent](#)) и/или ресурсов, которое является субъектом ограничений одной или более политик ([policy](#)).

Electronic Data Interchange (EDI) - обмен электронными данными

Автоматизированный обмен любыми заранее определенными и структурированными данными в сфере бизнеса между информационными системами двух или более организаций [\[ISO/IEC 14662\]](#).

encryption - шифрование

Криптографическое преобразование данных из открытого текста ("plaintext") в зашифрованную форму ("ciphertext"), что позволяет скрыть исходное содержание текста и, тем самым, предохранить его от прочтения или использования. Если это преобразование обратимо, соответствующий обратный процесс называется

дешифровкой. Дешифровка возвращает зашифрованный текст в его исходное состояние. [\[RFC 2828\]](#)

end point - конечная точка

Ассоциация между связыванием ([binding](#)) и сетевым адресом, специфицированным через URI (Uniform Resource Identifier), которая может быть использована для связи с экземпляром службы ([service](#)). Конечная точка указывает конкретное место для доступа к службе с использованием конкретного протокола ([protocol](#)) и формата данных [\[WSD Reqs\]](#).

gateway - шлюз

Агент, который завершает передачу сообщения ([message](#)) на входном интерфейсе ([interface](#)), чтобы представить его через выходной интерфейс как новое сообщение. В отличие от прокси ([proxy](#)) шлюз принимает сообщения так, как если бы он был конечным получателем сообщения. Из-за возможного рассогласования входного и выходного интерфейсов сообщение в процессе преобразования может измениться и даже утратить часть своего значения. Например, операция HTTP PUT не имеет эквивалента в SMTP.

Замечание: шлюз может быть, а может и не быть SOAP-узлом (SOAP node), однако, шлюз никогда не бывает SOAP-посредником (SOAP intermediary), т.к. шлюз завершает передачу сообщения, а SOAP-посредники пересылают их дальше. Шлюз выполняет свою роль перманентно, в то время как SOAP-посредник включается только при передаче сообщений.

idempotent - идемпотентный

Идемпотентным считается взаимодействие, чьи результаты и побочные эффекты не зависят от того, было ли это взаимодействие однократным или выполнялось много раз [\[RFC 2616\]](#). Безопасные взаимодействия (Safe) по своей сути являются идемпотентными.

identifier - идентификатор

Имя ресурса, однозначно идентифицирующее ресурс.

initial SOAP sender - исходный SOAP-отправитель

SOAP-отправитель (SOAP sender), который порождает SOAP-сообщение (SOAP message) в начальной точке пути SOAP-сообщения (SOAP message path).

integrity - целостность

Гарантия того, что информация не будет случайно или умышленно изменена или уничтожена [\[NSA Glossary\]](#).

loose coupling - слабое связывание

Связывание - это зависимость между взаимодействующими системами. Эта зависимость может быть разделена на реальную и искусственную.

1. Реальная зависимость - это совокупность свойств или служб, которые одна система заимствует у других систем. Реальная зависимость всегда существует и не может быть уменьшена.

2. Искусственная зависимость - это совокупность факторов, которым система должна подчиняться, чтобы иметь возможность пользоваться свойствами или службами, предоставляемыми другой системой. Типичными факторами искусственной зависимости являются языки программирования, платформы, API и т.д. Искусственная зависимость всегда существует, но она или ее стоимость могут быть уменьшены.

Слабое связывание описывает конфигурацию, при которой искусственная зависимость сведена к минимуму.

manageable service - управляемая служба

Web-служба ([Web service](#)) становится управляемой службой с дополнительной семантикой, четко сформулированной политикой и возможностями мониторинга и управления (выраженными через интерфейс управления ([management interface](#))).

management - управление

Использование возможностей управления в системе управления, с тем чтобы система могла выполнять мониторинг стоимости, отслеживание состояний и управление сущностями с целью создания и поддержания устойчивой операционной среды.

management capability - возможности управления

Возможности, которыми располагает Web-служба ([Web service](#)) для управления или мониторинга, и которые могут быть предоставлены системе управления для управления службой.

management interface - интерфейс управления

Интерфейс ([Interface](#)), через который представляются возможности управления службы ([management capabilities](#)).

management policy - политика управления

Политика ([Policy](#)), связанная с Web-службой с целью описания обязательств ([obligations](#)) управления и полномочий ([permissions](#)) для службы.

management semantics - семантика управления

Семантика управления службы является расширением семантики службы ([semantics of a service](#)) за счет добавления специфической семантики управления. Семантика управления определяет соглашение между поставщиком ([provider entity](#)) и потребителем ([requester entity](#)), касающееся требований к управлению и политике управления службой.

message - сообщение

1. Основная единица данных, пересылаемая от одного агента ([agent](#)) Web-служб ([Web services](#)) другому в контексте Web-служб.
2. Основная единица связи между Web-службой ([Web service](#)) и потребителем ([requester](#)): данные, передаваемые или получаемые Web-службой как одна логическая пересылка [[WSD Reqs](#)].
3. См. также [SOAP message](#).

message correlation - корреляция сообщения

Связь сообщения ([message](#)) с контекстом. Корреляция сообщения гарантирует, что агент потребителя ([requester agent](#)) может установить соответствие между ответом и запросом, особенно когда возможно получение нескольких ответов.

message exchange pattern (MEP) - шаблон для обмена сообщениями

1. Шаблон, свободный от семантики приложения, представляющий образец для обмена сообщениями ([messages](#)) между агентами ([agents](#)). Шаблон описывает отношения (например, временные, причинные, последовательные и т.д.) между многими сообщениями, обмен которыми происходит в соответствии с шаблоном, а также нормальное и аварийное завершение любого обмена сообщениями, соответствующего шаблону.
2. См. [SOAP message exchange pattern \(MEP\)](#).

message receiver - получатель сообщения

Агент ([agent](#)), который получает сообщение ([message](#)).

message reliability - надежность сообщения

Степень уверенности в том, что сообщение ([message](#)) будет доставлено и что и отправитель ([sender](#)) и получатель ([receiver](#)) будут иметь одинаковое представление о статусе пересылки.

message sender - отправитель сообщения

Агент ([agent](#)), который посылает сообщение ([message](#)).

message transport - транспортировка сообщения

Механизм, который могут использовать агенты ([agents](#)) для доставки сообщений ([messages](#)).

non-repudiation - подтверждение

Метод, с помощью которого отправитель данных получает доказательство их доставки, а получателю предоставляется возможность идентифицировать отправителя, так что ни один из них не сможет позже отрицать, что данные были обработаны. [[INFOSEC Glossary](#)].

obligation - обязательство

Вид политики ([policy](#)), которая определяет действия и/или состояния агента и/или ресурса.

operation - операция

Совокупность сообщений ([messages](#)), относящихся к действию одной Web-службы ([Web service](#)). [[WSD Reqs](#)].

orchestration - оркестровка

Оркестровка определяет последовательность и условия, при которых одна Web-служба ([Web service](#)) вызывает другие Web-службы, чтобы выполнить некоторую полезную функцию. Т.е. оркестровка - это образец взаимодействия, которому должен следовать агент Web-службы для достижения своей цели.

permission - разрешение

Вид политики ([policy](#)), которая определяет допустимые действия и состояния агента и/или ресурса.

permission guard - защита разрешения

Механизм, запускаемый от имени владельца, чтобы привести в действие политику разрешения ([permission](#)).

person or organization - человек или организация

Владелец агентов ([agents](#)), которые поставляют или запрашивают Web-службы ([Web services](#)).

policy - политика

Ограничение на поведение агентов ([agents](#)) или субъектов и организаций ([person or organization](#)).

policy guard - защита политики

Механизм, который приводит в действие одну или несколько политик ([policies](#)). Запускается от имени владельца.

principal - принципал

Элемент системы ([system entity](#)), чья идентификация может быть аутентифицирована [[X.811](#)].

privacy policy - политика секретности

Набор правил и действий, которые определяют или регламентируют, как человек или организация ([person or organization](#)) собирают, обрабатывают (используют) и раскрывают персональные данные другой стороны в результате взаимодействия. ные данные другой стороны в результате взаимодействия.

provider agent - агент поставщика

Агент ([agent](#)), который способен и уполномочен выполнять действия, связанные со службой ([service](#)), от имени своего владельца - поставщика ([provider entity](#)).

provider entity - поставщик

Человек или организация ([person or organization](#)), предоставляющие Web-службу ([Web service](#))

protocol - протокол

Набор формальных правил, описывающих, как пересылаются данные, особенно по сети. Протоколы низкого уровня определяют электрические и физические стандарты, которые должны соблюдаться, последовательность битов и байтов, передачу данных и обнаружение ошибок, коррекцию потока битов. Протоколы высокого уровня регулируют форматирование данных, включая синтаксис сообщений, терминал для диалога с компьютером, наборы символов, последовательность сообщений и т.д. [[FOLDOC](#)].

проху - прокси

Агент ([agent](#)), который ретранслирует сообщение агента потребителя ([requester agent](#)) агенту поставщика ([provider agent](#)), выступая перед Web-службой ([Web service](#)) в качестве потребителя.

quality of service - качество обслуживания

Обязательство ([obligation](#)) для потребителей услуг, принятое и объявленное поставщиком ([provider entity](#)).

reference architecture - эталонная архитектура

Обобщенная архитектура ([architecture](#)) нескольких конечных систем, относящихся к одной или нескольким общим областям деятельности. Эталонная архитектура определяет инфраструктуру, общую для конечных систем, и интерфейсы для компонент, которые будут включены в эти системы. Используя эталонную архитектуру, можно создавать архитектуру программного обеспечения для конкретной системы. Определение эталонной архитектуры облегчает построение и расширение новых архитектур программного обеспечения для классов систем. Таким образом, эталонная архитектура играет двойную роль по отношению к конкретным целевым архитектурам программного обеспечения. Во-первых, она обобщает и выделяет общие функции и конфигурации. Во-вторых, обеспечивает основу для создания целевых систем, которые используют эту общую основу более надежно и более выгодно [[Ref Arch](#)].

registry - системный реестр

Надежное, находящееся под центральным управлением хранилище информации.

requester agent - агент потребителя

Агент ([agent](#)), который хочет взаимодействовать с агентом поставщика ([provider agent](#)), чтобы запросить выполнение задачи от имени своего владельца - потребителя ([requester entity](#)).

requester entity - потребитель

Человек или организация ([person or organization](#)), которые хотят использовать Web-службу ([Web service](#)) поставщика ([provider entity](#)).

safe - безопасное взаимодействие

Взаимодействие, которое состоит только в поиске информации [[RFC 2616](#)].

security administration - администрирование безопасности

Конфигурирование, обеспечение безопасности и/или развертывание систем или приложений, что обеспечивает существование областей безопасности ([security domain](#)).

security architecture - архитектура безопасности

Совокупность правил для административной области и ее областей безопасности ([security domain](#)), которые описывают службы безопасности ([security services](#)), необходимые системе для удовлетворения потребностей ее пользователей,

элементы системы, необходимые для реализации этих служб, и уровни выполнения, необходимые этим элементам для общения со средой, представляющей опасность. Полная архитектура безопасности для системы касается административной безопасности, безопасности связи, безопасности компьютеров, безопасности излучений, безопасности персонала, физической безопасности и предписывает политику безопасности каждому из этих видов безопасности. Полная архитектура безопасности должна учитывать как преднамеренные, задуманные опасности, так и опасные случайности. Архитектура безопасности должна развиваться вместе с развитием административной области. [\[RFC 2828\]](#)

security auditing - аудит безопасности

Служба ([service](#)), которая надежно и безопасно фиксирует события, связанные с безопасностью, оставляя информацию, по которой можно реконструировать и изучать последовательность событий. К событиям безопасности можно отнести события аутентификации, принудительные решения политики и другие. Итоговая информация, полученная в результате аудита, может быть использована для обнаружения атак, подтверждения согласованности с политикой, предотвращения злоупотреблений и для других целей.

security domain - область безопасности

Среда или контекст, определенные моделями безопасности ([security models](#)) и архитектурой безопасности ([security architecture](#)), и включающие набор ресурсов и набор системных элементов, которые авторизованы ([authorized](#)) для доступа ([access](#)) к ресурсам. Одна или несколько областей безопасности могут находиться в одной административной области. Характерные черты конкретной области безопасности со временем могут изменяться. [\[RFC 2828\]](#)

security mechanism - механизм безопасности

Процесс (или устройство, запускающее такой процесс), используемый системой для запуска службы безопасности ([security service](#)), которая поставляется системой или существует в ней самой.

security model - модель безопасности

Семантическое описание множества объектов и отношений, определяющее набор служб безопасности ([security services](#)), предоставляемых системой или существующих в системе. [\[RFC 2828\]](#)

security policy - политика безопасности

Набор правил и действий, которые определяют, как система или организация поставляют службы безопасности ([security services](#)) для защиты ресурсов. Политики безопасности являются компонентами архитектуры безопасности ([security architectures](#)). Значительная часть политик безопасности реализуется через службы безопасности, при этом используются выражения политики безопасности ([security policy expressions](#)). [\[RFC 2828\]](#)

security policy expression - выражение политики безопасности

Соответствие между идентификацией принципала ([principal](#)) и/или его атрибутами и допустимыми действиями. Выражения политики безопасности часто являются по существу списками управления доступом. [\[STG\]](#)

security service - служба безопасности

Служба ([service](#)) обработки данных или связи, которая предоставляется системой для выполнения определенного вида защиты ресурсов, причем эти ресурсы могут принадлежать как данной системе, так и другим системам, например, служба аутентификации или служба атрибуции и аутентификации документов, базирующихся на PKI (Public Key Infrastructure). Служба безопасности является надмножеством служб AAA (authentication, authorization, accounting). Службы безопасности обычно реализуют политики безопасности ([security policies](#)) и сами реализованы через механизмы безопасности ([security mechanisms](#)) [\[RFC 2828\]](#).

service - служба

1. Абстрактный ресурс, представляющий возможность выполнения задач, которые имеют четкие функции с точки зрения поставщиков ([providers entities](#)) и потребителей ([requesters entities](#)). Чтобы службой можно было воспользоваться, она должна быть реализована конкретным агентом поставщика ([provider agent](#)).
2. Служба WSDL: совокупность конечных точек ([end points](#)) [\[WSD Reqs\]](#).
3. См. Web-служба ([Web service](#)).

service description - описание службы

Набор документов, описывающих интерфейс ([interface](#)) и семантику ([semantics](#)) службы ([service](#)).

service interface - интерфейс службы

1. Абстрактная граница, которую проводит службу ([service](#)). Она определяет типы сообщений ([messages](#)) и шаблоны для обмена сообщениями ([message exchange patterns](#)), которые участвуют во взаимодействии со службой, а также условия, подразумеваемые этими сообщениями.
2. Логическая группировка операций ([operations](#)). Интерфейс представляет собой абстрактный тип службы, не зависящий от протокола передачи ([protocol](#)) и формата данных [\[WSD Reqs\]](#).

service intermediary - служба-посредник

1. Web-служба ([Web service](#)), главное назначение которой - трансформировать сообщения ([messages](#)) путем добавления некоторых данных. (Если говорить о

передаче сообщений, посредник обрабатывает сообщения на пути от одного агента к другому). Более точно, мы говорим, что посредник - это служба, чьи исходящие сообщения эквивалентны его входящим сообщениям в некотором смысле, определенном приложением.

2. См. [SOAP intermediary](#).

service provider - поставщик службы

См. [provider agent](#) и [provider entity](#). См. также discussion about service provider в [\[WS Arch\]](#).

service requester - потребитель службы

См. [requester agent](#) и [requester entity](#). См. также discussion about service requester в [\[WS Arch\]](#).

service role - роль службы

Абстрактный набор задач, который субъект или организация ([person or organization](#)), предлагающие службу ([service](#)), идентифицируют как релевантный. Роли службы связываются также с отдельными аспектами сообщений ([messages](#)), предназначенных для обмена со службой.

service semantics - семантика службы

Семантика службы ([service](#)) - это поведение службы, ожидаемое при взаимодействии с ней. Семантика служит выражением контракта (не обязательно узаконенного) между поставщиком ([provider entity](#)) и потребителем ([requester entity](#)). Она описывает также эффект от вызова службы. Семантика службы может быть формально описана в машинно-читаемой форме, может быть идентифицирована, но не определена формально или определена неформально посредством отдельного соглашения между поставщиком и потребителем.

service-oriented architecture - служебно-ориентированная архитектура

Набор компонент ([components](#)), которые можно вызывать, и чьи описания интерфейса ([interface](#)) могут быть опубликованы и обнаружены ([discovered](#)).

session - сессия

Продолжительное взаимодействие объектов системы ([system entities](#)), часто с участием пользователя, в течение которого сохраняется некоторое состояние ([state](#)) этого взаимодействия [\[WSIA Glossary\]](#). Такое взаимодействие не обязательно ограничивается одним соединением ([connection](#)) между элементами системы.

SOAP - *Simple Object Access Protocol*

Формальный набор соглашений, определяющих формат и правила обработки SOAP-сообщений. ([SOAP message](#)) Эти соглашения касаются взаимодействия между SOAP-узлами ([SOAP nodes](#)), генерирующими и принимающими SOAP-сообщения с целью обмена информацией на пути передачи SOAP-сообщения ([SOAP message path](#)).

SOAP application - SOAP-приложение

Сущность программного обеспечения, которая создает, обрабатывает или выполняет другие действия над SOAP-сообщениями ([SOAP messages](#)), согласованные с SOAP-моделью обработки сообщений.

SOAP binding - SOAP-связывание

Формальный набор правил для передачи SOAP-сообщения ([SOAP message](#)) внутри или над другим протоколом (нижележащий протокол) с целью обмена. Примерами SOAP-связываний могут служить переносы SOAP-сообщения внутри тела сущности HTTP или над потоком TCP.

SOAP body - SOAP-тело

Совокупность блоков элементов информации (может быть, пустая), направляемая конечному SOAP-получателю ([ultimate SOAP receiver](#)) на пути SOAP-сообщения ([SOAP message path](#)).

SOAP envelope - SOAP-конверт

Наружный блок элементов информации SOAP-сообщения ([SOAP message](#)).

SOAP fault - SOAP-отказ

Блок элементов информации SOAP, содержащий информацию об отказе, сгенерированную SOAP-узлом ([SOAP node](#)).

SOAP feature - SOAP-свойство

Расширение рамок работы с SOAP-сообщениями, обычно касающееся обмена сообщениями между связанными SOAP-узлами ([SOAP nodes](#)). Примерами свойств являются "надежность", "безопасность", "корреляция", "маршрутизация" и концепция шаблонов обмена сообщениями.

SOAP header - SOAP-заголовок

Совокупность блоков заголовка ([SOAP header blocks](#)) (может быть, пустая), каждый из которых может иметь целью любого SOAP-получателя на пути SOAP-сообщения ([SOAP message path](#)).

SOAP header block - блок SOAP-заголовка

Элемент информации, используемый для разграничения данных, которые логически составляют одну вычислительную единицу внутри SOAP-заголовка ([SOAP header](#)). Тип блока заголовка идентифицируется полным именем элемента информации блока.

SOAP intermediary - SOAP-посредник

Посредником может быть как SOAP-получатель ([SOAP receiver](#)), так и SOAP-отправитель ([SOAP sender](#)), и обращение к нему происходит из SOAP-сообщения ([SOAP message](#)). Посредник обрабатывает направленные ему блоки SOAP-заголовка ([SOAP header blocks](#)) и передает сообщение дальше - конечному SOAP-получателю ([ultimate SOAP receiver](#)).

SOAP message - SOAP-сообщение

Основная единица связи между SOAP-узлами ([SOAP nodes](#)).

SOAP message exchange pattern (MEP) - SOAP-шаблон для обмена сообщениями

Шаблон для обмена SOAP-сообщениями ([SOAP messages](#)) между SOAP-узлами ([SOAP nodes](#)), используемый одним или более связываниями SOAP-протокола ([SOAP protocol bindings](#)). SOAP MEP является примером SOAP-свойства ([SOAP feature](#)).

SOAP message path - путь SOAP-сообщения

Совокупность SOAP-узлов ([SOAP nodes](#)), через которые проходит отдельное SOAP-сообщение. В эту совокупность входит начальный узел-отправитель ([SOAP sender](#)), несколько узлов-посредников ([SOAP intermediaries](#)) (может быть, ни одного) и конечный узел-получатель ([ultimate SOAP receiver](#)).

SOAP node - SOAP-узел

Овеществление логики обработки, необходимой для передачи, получения, обработки и/или ретрансляции SOAP-сообщения ([SOAP message](#)) в соответствии с правилами, определенными этой рекомендацией. SOAP-узел отвечает за выполнение правил, управляющих обменом SOAP-сообщениями. Он осуществляет доступ к службам, поддерживаемым базовыми протоколами, через одно или несколько SOAP-связываний ([SOAP bindings](#)).

SOAP receiver - SOAP-получатель

SOAP-узел ([SOAP node](#)), который принимает SOAP-сообщение ([SOAP message](#)).

SOAP role - SOAP-роль

Ожидаемое функционирование SOAP-узла ([SOAP node](#)) при обработке сообщения.. SOAP-узел может выполнять много ролей.

SOAP sender - SOAP-отправитель

SOAP-узел ([SOAP node](#)), который отправляет SOAP-сообщение ([SOAP message](#)).

state - состояние

Набор атрибутов ([attributes](#)), представляющих свойства компоненты ([component](#)) в некоторый момент времени.

synchronous - синхронный

О взаимодействии говорится как о синхронном, когда участвующие в нем агенты должны быть доступны для получения и обработки сообщений с момента начала взаимодействия и до тех пор, пока не будут получены все сообщения или пока не возникнет какая-то аварийная ситуация. Точное значение определения "должны быть доступны для получения сообщений" зависит от характеристик участвующих во взаимодействии агентов (включая протокол передачи, который они используют); оно может (не обязательно) подразумевать сжатое время синхронизации, блокирование треда и т.д.

system entity - системная сущность

Активный элемент компьютерной или сетевой системы. Например, автоматизированный процесс или ряд процессов, подсистема, человек или группа людей, с которыми связан определенный набор функциональностей [\[RFC 2828\]](#).

transaction - транзакция

Свойство архитектуры, обеспечивающее координацию результатов или операций во включенном состоянии при многошаговом взаимодействии. Основное свойство транзакции - способность объединять множество действий в одну единицу работы таким образом, что все эти действия либо успешно завершаются, либо заканчиваются аварийно как одно целое.

ultimate SOAP receiver - конечный SOAP- получатель

SOAP-получатель ([SOAP receiver](#)), который является конечным адресатом для SOAP-сообщения ([SOAP message](#)). Он отвечает за обработку содержимого направленных ему SOAP-тела ([SOAP body](#)) и блоков SOAP-заголовков ([SOAP header blocks](#)). При некоторых обстоятельствах SOAP-сообщение может не дойти до конечного получателя, например, из-за проблем с SOAP-посредником ([SOAP intermediary](#)). Конечный SOAP-получатель не может быть SOAP-посредником для того же SOAP-сообщения.

usage auditing - аудит использования

Служба ([Service](#)), которая надежно и безопасно фиксирует события, связанные с использованием системы, формируя записи, которые позволяют реконструировать

и изучать последовательность событий. Примером таких событий может быть выделение и освобождение ресурсов.

Web service - Web-служба

Много есть на свете такого, что можно было бы назвать "Web-службами". Однако, исходя из целей данной Рабочей Группы и из данной архитектуры, и без предубеждения по отношению к другим определениям мы будем использовать следующее определение: Web-служба - это система программной поддержки взаимодействия "машина - машина" через сеть. Web-служба имеет интерфейс, описанный в формате, доступном для машинной обработки (WSDL). Другие системы взаимодействуют с Web-службой так, как это определено ее описанием, с использованием SOAP-сообщений, передаваемых обычно по протоколу HTTP с XML-сериализацией в сочетании с другими Web-стандартами.

3 Ссылки

CCA T&D

[CCA Terms and Definitions, CCA Forum, Kate Keahey](http://www.cca-forum.org/glossary.shtml) (See <http://www.cca-forum.org/glossary.shtml>)

Fielding

[Architectural Styles and the Design of Network-based Software Architectures, PhD dissertation, R. Fielding, 2000](http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm) (See <http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm>)

FOLDOC

[The Free On-line Dictionary of Computing, D. Howe](http://www.foldoc.org/) (See <http://www.foldoc.org/>)

INFOSEC Glossary

National Information Systems Security (INFOSEC) Glossary, National Security Telecommunications and Information Systems Security Instruction (NSTISSI) No. 4009, 5 June 1992

ISO/IEC 14662

[Information technology -- Open-edi reference model, International Standard, ISO/IEC 14662:1997](http://www.iso.org/iso/en/CatalogueDetailPage.CatalogueDetail?CSNUMBER=25154) (See <http://www.iso.org/iso/en/CatalogueDetailPage.CatalogueDetail?CSNUMBER=25154>)

NSA Glossary

NSA Glossary of Terms Used in Security and Intrusion Detection, NSA, April 1998

Soft Arch Pract

Software Architecture in Practice, ISBN 0201199300, L. Bass, P. Clements, R. Kazman, 1997

Ref Arch

[Using the Architecture Tradeoff Analysis Method\(SM\) to Evaluate a Reference Architecture: A Case Study, B. Gallagher, June 2000](http://www.sei.cmu.edu/publications/documents/00.reports/00tn007/00tn007.html) (See <http://www.sei.cmu.edu/publications/documents/00.reports/00tn007/00tn007.html>)

RFC 2616

[Hypertext Transfer Protocol -- HTTP/1.1, IETF RFC 2616, R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, T. Berners-Lee June 1999](http://www.ietf.org/rfc/rfc2616.txt) (See <http://www.ietf.org/rfc/rfc2616.txt>)

RFC 2828

[Internet Security Glossary, IETF RFC 2828, R. Shirey, May 2000](http://www.ietf.org/rfc/rfc2828.txt) (See <http://www.ietf.org/rfc/rfc2828.txt>)

RFC 2829

[Authentication Methods for LDAP, IETF RFC 2829, M. Wahl, H. Alvestrand, J. Hodges, R. Morgan, May 2000](#) (See <http://www.ietf.org/rfc/rfc2829.txt>)

STG

[Security Taxonomy and Glossary, L. Wheeler](#) (See <http://www.garlic.com/~lynn/secure.htm>)

SOAP12 Part1

[SOAP Version 1.2 Part 1: Messaging Framework, W3C Recommendation, M. Gudgin, M. Hadley, N. Mendelsohn, J-J. Moreau, H. Nielsen, 24 June 2003](#) (See <http://www.w3.org/TR/2003/REC-soap12-part1-20030624/>)

UeB Glossary

[UN/CEFACT eBusiness Glossary, UN/CEFACT Working Draft Revision 0.53, 13 December 2002](#)

Web Arch

[Architecture of the World Wide Web, First Edition, W3C Working Draft, I. Jacobs, 9 December 2003](#) (See <http://www.w3.org/TR/2003/WD-webarch-20031209/>)

WS Arch

[Web Services Architecture, W3C Working Group Note, D. Booth, H. Haas, F. McCabe, E. Newcomer, M. Champion, C. Ferris, D. Orchard, 11 February 2004](#) (See <http://www.w3.org/TR/2004/NOTE-ws-arch-20040211/>)

WSIA Glossary

[Glossary for the OASIS WebService Interactive Applications \(WSIA/WSRP\), OASIS draft, 3 May 2002](#) (See <http://www.oasis-open.org/committees/wsia/glossary/wsia-draft-glossary-03.htm>)

WSC Reqs

[Web Services Choreography Requirements 1.0, W3C Working Draft, D. Austin, A. Barbir, E. Peters, S. Ross-Talbot, 12 August 2003](#) (See <http://www.w3.org/TR/2003/WD-ws-chor-reqs-20030812/>)

WSD Reqs

[Web Service Description Requirements, W3C Working Draft, J. Schlimmer, 28 October 2002](#) (See <http://www.w3.org/TR/2002/WD-ws-desc-reqs-20021028/>)

X.800

[Information processing systems Open Systems Interconnection Basic Reference Model Part 2: Security Architecture, ISO 7498-2:1989, ITU-T Recommendation X.800 \(1991\)](#) (See <http://www.itu.int/itudoc/itu-t/rec/x/x500up/x800.html>)

X.811

[Security Frameworks for Open Systems: Authentication Framework, ITU-T Recommendation X.811 \(1995 E\), ISO/IEC 10181-2:1996\(E\)](#) (See <http://www.itu.int/itudoc/itu-t/rec/x/x500up/x811.html>)

X.812

[Security frameworks for open systems: Access control framework, ITU-T Recommendation X.812 \(1995 E\), ISO/IEC 10181-3:1996\(E\)](#) (See <http://www.itu.int/itudoc/itu-t/rec/x/x500up/x812.html>)