RooX UIDM Integration Guide

Оглавление

Выбор интерфейса интеграции

Обзор предоставляемых способов

Перед интеграцией веб-приложения или сервиса с сервером RooX UIDM необходимо выбрать предпочтительный вариант интеграции.

Выбор зависит от множества факторов, таких как:

- Технология, на которой написано приложение
- Версия и имя программы веб-сервера/сервера приложений, обслуживаюшего приложение
- Насколько требования по интеграции отличаются от типовых (возможно ли требования к доступу к Web-ресурсам описать простыми URL-правилами) и требуется ли специфическая логика, не реализованная в стандартных средствах
- Тип приложения: Web-сайт, API
- Уровень доверия к приложению, в частности находится ли приложение в общем домене второго уровня с сервером RooX UIDM
- Необходимость авторизации приложения от лица пользователя
- Возможность доработки приложения
- Требования к производительности решения
- Сроки и стоимость внедрения

Прежде всего перечислим возможные способы интеграции с технической точки зрения.

Отдельное Web-приложение, выступающее в роли прокси перед защищаемым веб-приложением

В данном варианте перед защищаемым приложением устаналивается Policy Gateway - компонент, работающий в режиме прокси-сервера. Policy Gateway реализован как отдельное веб-приложение на Java, которое перехватывает запросы к целевому веб-приложению, проверяет аутентифирован ли пользователь, вычисляет политики доступа к запрошенному URL.

Если условия не удовлетворены, производится редирект браузера пользователя на сервис аутентификации в RooX UIDM. После успешной аутентификации браузер пользователя направляется обратно на целевой адрес.

Авторизованные запросы Policy Gateway пропускает к веб-приложению, передавая HTTP-заголовки содержащие идентификатор аутентифицированного пользователя и опционально атрибуты сессии и/или профиля.

Вариант предпочтителен для приложений

- Состоящих из полностью отделенных по URL-маскам открытой и закрытой зон
- Модификация которых невозможна или не желательна
- Недоверенных приложений
- Требующих быстрого внедрения интеграции с RooX UIDM

Вариант не подходит для приложений

- Предоставляющих API
- Так называемых открытых порталов, где нет четко выраженных зон, а контент динамически изменяется в зависимости от условий параметров авторизационной сессии
- С высокой нагрузкой (более тысячи просмотров защищенных ресурсов в секунду)
- Не находящихся в одном домене второго уровня с сервером RooX UIDM
- Требующих разрешения доступа к своим данным от лица пользователя

Policy Agent - фильтр, встроенный в защищаемое веб-приложение

В данном случае фильтр как програмный модуль встраивается в защищаемое веб-приложение или в веб-сервер. Реализация зависит в первую очередь от технологии, на которой написано веб-приложение. Поставляются фильтры для веб-серверов Apache HTTP Server 2.2, 2.4, Microsoft IIS 6, 7, 8, Java Servlet >= 2.5 Compatible Server.

Вариант интеграции похож на вариант с отдельным Policy Gateway, отличие только в способе развертывания.

Policy Agent перехватывает запросы к целевому веб-приложению, проверяет аутентифирован ли пользователь, вычисляет политики доступа к запрошенному URL.

Если условия не удовлетворены, производится редирект браузера пользователя на сервис аутентификации в RooX UIDM. После успешной аутентификации браузер пользователя направляется обратно на целевой адрес.

Авторизованные запросы Policy Agent пропускает к веб-приложению, передавая виртуальные HTTP-заголовки содержащие идентификатор аутентифицированного пользователя и опционально атрибуты сессии и/или профиля.

Вариант предпочтителен для приложений

- Состоящих из полностью отделенных по URL-маскам открытой и закрытой зон

Вариант не подходит для приложений

- Предоставляющих API
- Так называемых открытых порталов, где нет четко выраженных зон, а контент динамически изменяется в зависимости от условий параметров авторизационной сессии
- Веб-приложений, модификация которых невозможна или не желательна
- Недоверенных приложений
- Не находящихся в одном домене второго уровня с сервером RooX UIDM
- Требующих разрешения доступа к своим данным от лица пользователя

Использование SDK для авторизации запросов

Функции авторизации запросов ложатся на защищаемое веб-приложение. Поставляется библиотека интеграции (доступны реализации на Java и C), которая содержит методы проверки валидности сессии, получения атрибутов сессии и профиля пользователя, вычисления политик.

Как правило, веб-приложение содержит фильтр, который при поступлении запроса производит вызов SDK, передавая в фукнцию HTTP-заголовки запроса. В ответе от SDK содержится булево значение валидности сессии, время жизни сессии, принципал и опционально ассоциативный массив с атрибутами сессии и профиля.

Далее фильтр устаналивает контекст безопасности и передает управление бизнес-логике веб-приложения, которая сверяется с контекстом безопасности перед выполнением защищенных сценариев. За счет возможности использования кастомного кода обработки контекста безопасности обеспечивается максимальная гибкость интеграции.

НАПОМИНАЕМ

Для приложений, использующих Spring Framework доступен модуль интеграции с Spring Security. SDK интегрируется с Spring Security, выполняя функции авторизации запросов, редиректа на сервис аутентификации и полной поддержки SecurityContext API и SpEL-выражений для @PreAuthorize

SDK предоставляет встроенную поддержку кеширования сессий и атрибутов пользователей.

Вариант предпочтителен для приложений

- Так называемых открытых порталов, где нет четко выраженных зон, а контент динамически изменяется в зависимости от условий параметров авторизационной сессии
- Со сложной логикой авторизации запросов
- Предоставляющих API

Вариант не подходит для приложений

- Веб-приложений, модификация которых невозможна или не желательна
- Реализованных на языках, отличных от Java и C и не имеющих средств вызова нативного кода
- Недоверенных приложений
- Не находящихся в одном домене второго уровня с сервером RooX UIDM
- Требующих разрешения доступа к своим данным от лица пользователя

Использование REST API для авторизации запросов

Функции авторизации запросов ложатся на защищаемое веб-приложение. Приложение выполняет запросы к RooX UIDM REST API для проверки валидности сессии, получения атрибутов сессии и профиля пользователя, вычисления политик. Приложение должно самостоятельно поддержать логику кеширования сессий и средства синхронизации конфигурации с центральным сервисов конфигурации.

ЗАМЕТКА

Важным отличием данного способа является независимость от используемых языков и технологий, однако требует большей квалификации от компании-разработчика, и как правило требует больше времени для разработки и внедрения.

Вариант предпочтителен для приложений

- Так называемых открытых порталов, где нет четко выраженных зон, а контент динамически изменяется в зависимости от условий параметров авторизационной сессии
- Со сложной логикой авторизации запросов
- Предоставляющих API

Вариант не подходит для приложений

- Веб-приложений, модификация которых невозможна или не желательна

- Требующих максимально быстрого внедрения
- Недоверенных приложений
- Не находящихся в одном домене второго уровня с сервером RooX UIDM
- Требующих разрешения доступа к своим данным от лица пользователя

Использование OAuth2.0/OpenID Connect API для авторизации запросов

Функции авторизации запросов ложатся на защищаемое веб-приложение. Приложение выполняет запросы к RooX UIDM OAuth2.0/OpenID Connect API для проверки валидности токена, получения атрибутов сессии и профиля пользователя, вычисления политик. Приложение должно самостоятельно поддержать логику кеширования сессий и средства синхронизации конфигурации с центральным сервисов конфигурации.

При использовании OAuth2.0 подключаемому сервису доступна не SSO-сессия пользователя, а токен доступа (access token), выданный конкретно этому сервису. Перед выдачей токена у пользователя спрашивается разрешение на предоставление доступа этому приложению к своим данным, при этом область выдаваемых разрешений может делиться на т.н. скоупы. Например, пользователь может выдать приложению "E-Shop" разрешение на выполнение регулярных платежей, но не выдавать разрешения на просмотр баланса счета и выполнение покупок.

Вариант предпочтителен для приложений

- Требующих разрешения доступа к своим данным от лица пользователя
- Не находящихся в одном домене второго уровня с сервером RooX UIDM
- Так называемых открытых порталов, где нет четко выраженных зон, а контент динамически изменяется в зависимости от условий параметров авторизационной сессии
- Со сложной логикой авторизации запросов
- Предоставляющих API
- Недоверенных приложений

Вариант не подходит для приложений

- Веб-приложений, модификация которых невозможна или не желательна
- Требующих максимально быстрого внедрения

Как выбрать подходящий способ?

Прежде всего необходимо отсечь принципиально неподходящие способы. Если останется более одного способа, необходимо выбрать решение, удовлетворяющее требованиям по срокам и полной стоимости внедрения.

Если требуется большая гибкость интеграции (сейчас или в ближайшей перспективе) рассмотреть варианты интеграции с использованием SDK или REST API, даже если эти варианты потребуют больших сроков или стоимости внедрения.

Для подключения недоверенных сервисов, находящихся вне зоны домена второго уровня компании или требующих разрешения от лица пользователя, необходимо использовать исключительно интеграцию по OAuth2.0/OpenID Connect.

Для быстрого прототипирования используйте Policy Gateway.

Детальное описание интерфейсов интеграции

@TODO

Policy Gateway

@TODO

REST API

@TODO

OAuth2.0/OpenID Connect API

@TODO

Policy SDK

@TODO

Java SDK

@TODO

C SDK

@TODO

Policy Agent

@TODO

Java Policy Agent

@TODO

IIS Policy Agent

@TODO

Apache Policy Agent

@TODO