

Ак Барс
Банк



AK BARS
DIGITAL

Аутентификация OAuth / OpenID Connect в Enterprise и среде Open API

Александр Семёнов, главный архитектор
Ак Барс Цифровые Технологии

В чем ценность хранения учетной записи пользователя у вас в базе?

Results		Messages						
	Id	UserName	PasswordHash	AccessFailedCount	Firstname	Patronymic	Surname	
11	5a51fe69-e35a-43bc-9acf-93c9643e8f1e	asafkond@gmail.com	AGAAAAEAC:GAAAAECg5h3LEDZw8l3g-wgNpRwvPFIJGwW5wTugHw5Z...	0	Kond	NULL	Su	
12	5dc2c78f-3295-4200-baf3-14bda94d1270	111	AGAAAAEAC:GAAAAE7wqkE2496w4EJFysuDegeA7Twew-JUPhAPWwCBT...	0	1	NULL	1	
13	63a1130d-4b70-4a40-8c67-6609407e4cef	etp@tut.by	AGAAAAEAC:GAAAAE7w-168DQZq-y6TjE3wW3HFBwHqLQw545w533wE...	4	Виткоп	NULL	Гаврилович	
14	7419ea37-574a-4857-b8c6-8046926be485	egafidov@mail.ru	AGAAAAEAC:GAAAAELVwep7w-wi3wep8G11ZBw4wLAg7WDFRwA-wJZF7...	0	Рустем	NULL	Габеевич	
15	958b5770-9d57-4f45-8f56-f3868764dd8	erfu@tut.by	AGAAAAEAC:GAAAAELg7ZF29wCC2XWVNGwKw9wPw7wvGAwNwLwWwFY...	0	Олег	NULL	Сосновский	
16	acbec3aa-8830-4a26-87be-b78dcbd4ffd4	ntrofandoff.serya@yandex.ru	AGAAAAEAC:GAAAAEE3wHFPg893D7wvyaD5-w8pHwLwAwAD4pHwHwFw...	0	Александр	NULL	Морозов	
17	bc4a7799-4aae-47aa-a657-ca881c42b9a9	1	AGAAAAEAC:GAAAAEEY0wZwvH87wGHEwF4Z3UPwv5PwvHwHwHw5w...	0	1	NULL	1	
18	c2f21504-fea6-4959-8dcd-ffb19792ccda	skv@tut.by	AGAAAAEAC:GAAAAELw8pALwH5w1-wEY8pHwLwLwHwvHwEY0wepW5...	0	Констант	NULL	Ситен	
19	c339ec63-4b65-4c92-bbc6-01216a6c284b	efmow@me.com	AGAAAAEAC:GAAAAEZ1V35wR5EwPOTWFIwHwHwRQwvKESwHwLwLw...	0	Александр	NULL	Евдоким	
20	cb17d56d-aa03-4bca-980b-5e302966d8c6	Flux	AGAAAAEAC:GAAAAE1YD7wQ53wHw-CHWBM35ZwHwL7wQ3wLw-gwv9w...	0	NULL	NULL	NULL	
21	e0d644d1-00da-47e9-8453-1a6c11cd73f8	test@y.com	AGAAAAEAC:GAAAAEG3L8DwHwHwQ3wHwHwPLOwL7HwLwvHwHwLwP2H...	0	Test	NULL	x	
22	e7277e95-8b21-4993-8320-dc5a488556af	Penk.kan@bk.ru	AGAAAAEAC:GAAAAEJwvExep8wLwLwHw5D7wLwPwLwLwLwLwLwLwLw...	0	Мари	NULL	Гаврилович	
23	fb392616-44fa-48a6-9e3f-5ce636989de4	serofuyp@gmail.com	AGAAAAEAC:GAAAAEG3KFP3wvHwHwHwHwHwHwHwHwHwHwHwHwHwHw...	0	Сергей	NULL	Савин	

- Пользователи лгали заполняя информацию о себе
- При входе возможно определить только тот факт, что это все тот же человек (или кот) который зарегистрировал учетную запись
- Но это не точно, потому что вам нужно позаботиться о безопасности учетных записей и препятствовать их взлому

Реализовать правильно систему аутентификации сложно

Если вы не эксперт по безопасности, и не причисляете себя к списку крутых компаний, способных реализовать устойчивую к взлому аутентификацию, лучше доверьте это профессионалам.

Многие слышали метод аутентификации **“Войти через ...”**

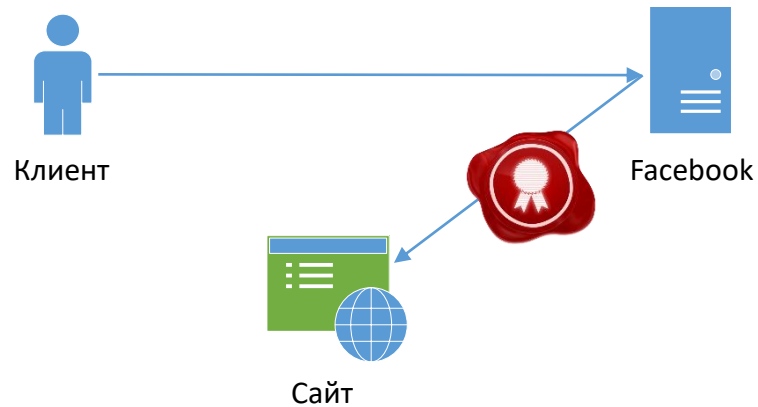
- Google,
- Facebook,
- VK,
- etc.

Использование 3rd party аутентификации

“Войти через ...”

Предположим, мы решили довериться Facebook:

- Клиент входит на сайт Facebook, идентифицируя себя
- Facebook пересылает вместе с клиентом подписанную информацию о клиенте



Использование 3rd party аутентификации

“Войти через ...”

Что мы получаем:

- Определяем тот факт, что это все тот же человек (или кот) который заходил к нам до этого
- Какую-то информацию о клиенте, которую он ввел при регистрации на Facebook
- Решение всех проблем с безопасностью учетных записей за нас

Использование 3rd party аутентификации

“Войти через ...”

Точно информация верная?

- Facebook пересылает перечень утверждений (claims) о пользователе
- Эта информация подписана ключом Facebook, что делает невозможным фальсификацию этой информации: изменение, добавление или удаление утверждений

кот Мурзик,
четыре лапы,
5 лет



Security tokens

Стандарты токенов доступа использующихся для передачи данных

JSON Web Token (JWT)

- JSON формат
- Симметричные и асимметричные подписи (HMACSHA256-384, ECDSA, RSA)
- Симметричное и асимметричное шифрование (RSA, AES/CGM)
- <http://self-issued.info/docs/draft-ietf-oauth-json-web-token.html>

Simple Web Token (SWT)

- Form/URL формат
- Симметричная подпись

SAML 1.1/2.0

- Базируется на XML
- Много вариантов шифрования и подписи
- Достаточно сложный для работы

Использование 3rd party аутентификации

“Войти через ...”

Информация верная точно!

- Провайдерами аутентификации могут выступать организации, проверившие пользователя!
- Конечно, информация может быть передана только с согласия самого клиента

Для этого как раз существует экран согласия, который вы иногда видите, используя “**вход через ...**” в какой-либо сервис

 Войти через Сбербанк Онлайн 

Это безопасно и удобно. Ваши данные
защищены надёжным шифрованием.



Использование 3rd party аутентификации



Вход через АК БАРС


Воспользуйтесь логином и паролем АК БАРС Online для

Войти

Аутентификация Ак Барс Онлайн 2.0 - Microsoft Edge

https://auth.akbars.ru/...

Ak Bars
Bank



Аутентификация Ак Барс Онлайн

Логин

Пароль

Войти

OAuth 1.0, 2.0

- Создание протокола началось в 2007 году несколькими крупными вендорами, и его целью было решение задач авторизации
- 2008 – к работе подключилась IETF
- 2010 – RFC 5849 определил OAuth 1.0, который получился очень тяжелым
- 2010 – WRAP (Web Resource Authorization Profile) был предложен Microsoft, Yahoo, Google как расширение, так же различные другие компании начали независимо расширять стандарт и реализовывали различные варианты протокола (из 30+ черновиков протокола каждый реализовывал свою версию)
- 2012 – Ведущий автор и редактор стандарта ушли из проекта и попросили удалить свои имена из всех документов
- 2012 – RFC 6749, RFC 6750 определил OAuth 2.0

Идея протокола OAuth 2.0 в разделении приложения и владельца ресурса, и возможности выдавать владельцем приложению временный ограниченный доступ к ресурсу.

Связь OAuth 1.0, 2.0 / OpenID Connect 1.0

- OAuth 1.0 и OAuth 2.0 не связаны никак, кроме общего имени (различные концепции),
- OpenID Connect (OIDC) является развитием OAuth 2.0, добавляя аутентификацию.

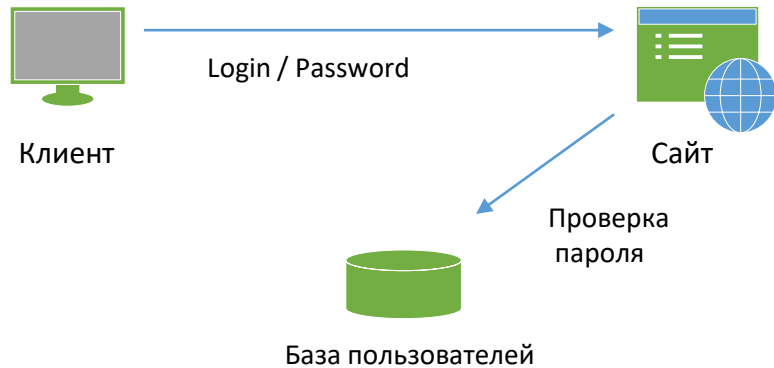
OpenID Connect появился в 2014 и является современным стандартом, объединяющим в себе:

- OpenID Attribute Exchange 1.0,
- OpenID 2.0,
- OAuth 2.0

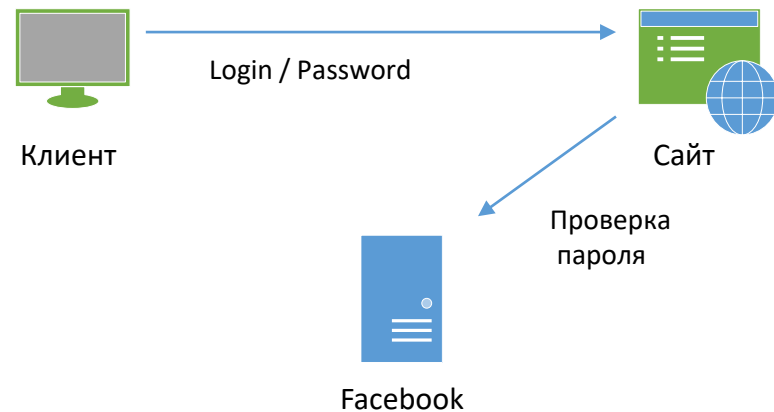
Добавляет протоколы discovery & dynamic registration, сессии, концепции:

- Информация о пользователе,
- ID Token

Работа протокола

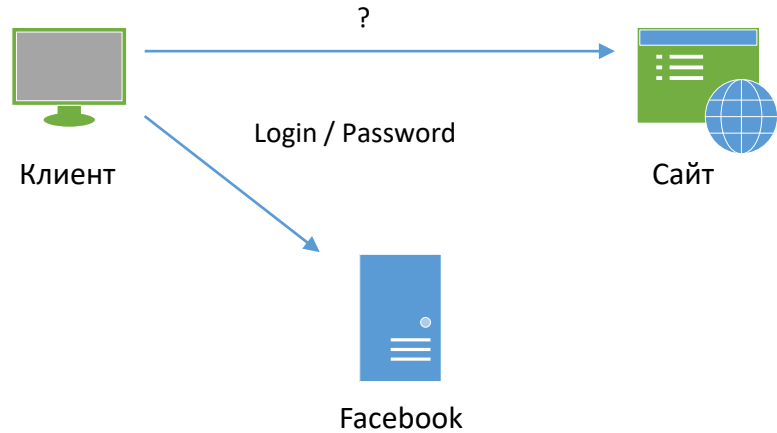


- Передача в каждом запросе логина и пароля (при использовании HTTPS это не так уж и плохо) (если клиент и сервер наш)



- Если пользователь не наш (Facebook)?
Пользователь передает свои логин и пароль от Facebook на сайт, а тот проверяет их на сервере Facebook.

Работа протокола

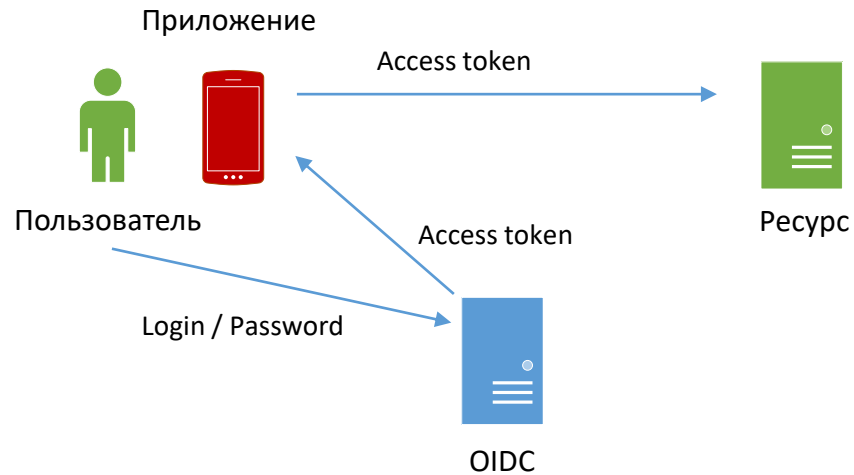


- Пользователь входит на сайт Facebook
- Пересылает удостоверяющую эту информацию на сайт

- Информация подписана ключом Facebook
- Содержит некоторые утверждения:
id = 0x69332
...

Access token

Работа протокола



- Пользователь использует стороннее (потенциально враждебное) приложение для доступа к своим данным
- Стороннее приложение используя браузер открывает страницу OIDC, где пользователь вводит свой логин и пароль
- OIDC передает access token в приложение

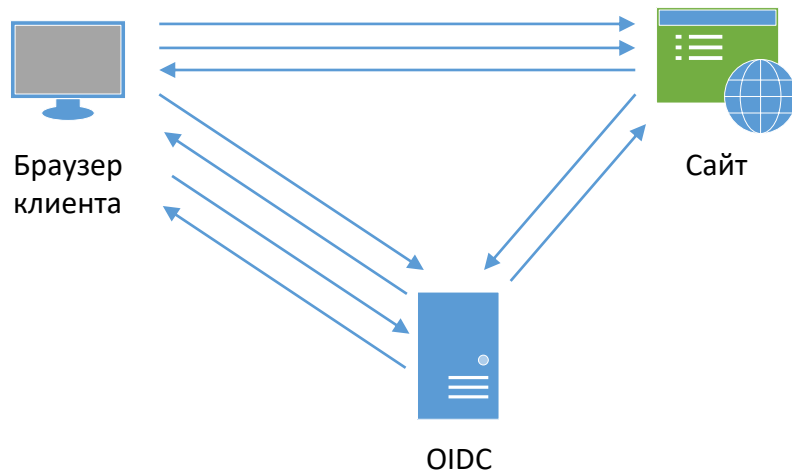
Access token:

- Имеет ограниченное время действия
- Может иметь ограничение на доступ



Работа протокола OpenID Connect

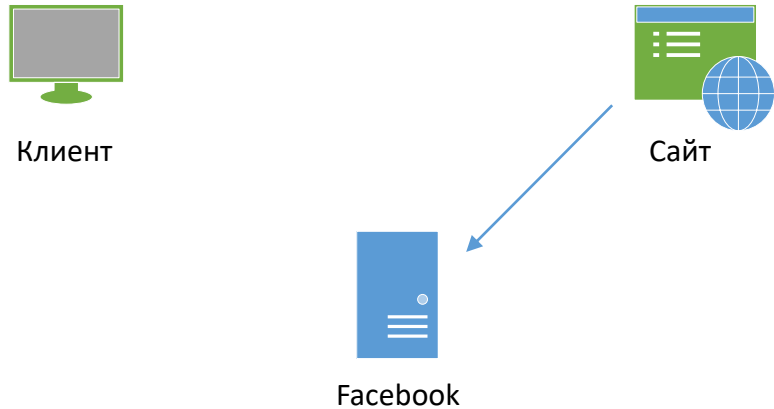
- Authorization Code
- Resource Owner Password Credentials
- Client Credentials Grant
- Implicit Grant



1. Браузер открывает страницу сайта
2. Сайт перенаправляет клиента на сервер OIDC
3. OIDC отображает страницу аутентификации
4. Клиент вводит логин и пароль
5. OIDC перенаправляет браузер обратно на сайт, передавая код авторизации
6. Сайт обменивает код авторизации на токен доступа

Сайту по сути не нужен access token, если он не обращается к внешним сервисам от лица клиента, но он нужен для получения дополнительной информации от OIDC о пользователе

Получение информации о пользователе



- Сайт, получив токен доступа для пользователя, может выполнить с ним запрос к OIDC для получения информации о пользователе.
(доступную, разрешенную информацию)
- Доступ к информации управляется через scopes
- Scopes вшиты в токен доступа, и выдаются OIDC

Scope – некое мнемоническое имя, имеющее значение в определенных контекстах

Примеры:

- profile,
- email,
- <https://auth.akbars.ru/scopes/gopoints>

Адреса и явки Open ID Connect

Точка получения мета-информации:

(<https://auth.akbars.ru/.well-known/openid-configuration>)

```
{
  "issuer": "https://auth.akbars.ru/",
  "authorization_endpoint": "https://auth.akbars.ru/connect/authorize",
  "token_endpoint": "https://auth.akbars.ru/connect/token",
  "introspection_endpoint": "https://auth.akbars.ru/connect/introspect",
  "end_session_endpoint": "https://auth.akbars.ru/connect/logout",
  "userinfo_endpoint": "https://auth.akbars.ru/connect/userinfo",
  "jwks_uri": "https://auth.akbars.ru/.well-known/jwks",
  "token_endpoint_auth_methods_supported": [ "client_secret_basic", "client_secret_post" ],
  "introspection_endpoint_auth_methods_supported": [ "client_secret_basic", "client_secret_post" ],
  "grant_types_supported": [ "implicit", "authorization_code", "refresh_token", "client_credentials", "password" ],
  "response_types_supported": [ "token", "code", "code token", "id_token", "id_token token", "code id_token", "code id_token token" ],
  "response_modes_supported": [ "form_post", "fragment", "query" ],
  "scopes_supported": [ "openid" ],
  "id_token_signing_alg_values_supported": [ "RS256" ],
  "code_challenge_methods_supported": [ "plain", "S256" ],
  "subject_types_supported": [ "public" ]
}
```

Использование OpenID Connect

В приложении (сайте):

- При необходимости аутентификации формировать redirect на сайт OIDC, предоставив все требуемые данные в строке запроса согласно стандарту
- Принимать возврат клиента с получением и обработкой данных от OIDC
- Устанавливать факт аутентификации клиента

Обычно реализуется подключением библиотеки OIDC, в том числе готовых под конкретных провайдеров (Google, Facebook и т.д.) для минимальной конфигурации

На сервере OIDC должно быть зарегистрировано ваше приложение

Реализация OpenID Connect

Сервер?

- Если вы планируете давать сторонним приложениям возможность входа вашим пользователям
- Централизовать для приложений логику аутентификации ваших пользователей
- Обеспечить единый вход в ваши приложения вашим пользователям

Реализовать правильно систему аутентификации сложно.

Надежную, устойчивую к взлому, полностью соответствующую всем стандартам

Стандарт формулирует поведение недостаточно жестко, указывая для важных проверок “Should”, “Shall”, “May” (изначально планировался как протокол, но затем статус изменили на фреймворк)

OAuth 2.0 Threat Model and Security Considerations

<https://tools.ietf.org/html/rfc6819>

Реализация OpenID Connect

- Можно взять один из готовых серверов OIDC, и используя их механизмы интеграции подключить к своей базе пользователей
- Если мы говорим о реализации поддержки стандарта, то мы можем реализовать необходимые API и логику непосредственно в нашем приложении (backend)
- Можем построить независимый сервис, и пользоваться им в нашем приложении, а так же использовать его в других наших системах
- Пользователи смогут пользоваться сервисами поддерживающими наш OIDC зарегистрировавшись у нас!

Реализация OpenID Connect

Помимо реализации OIDC для аутентификации пользователей, так же необходим IDM (Identity Management):

- Управление пользователями (создание, блокировка и т.д.)
- Управление правами пользователей
- Восстановление пароля
- и т.д.

Готовый OIDC сервер

- Это самостоятельный сервер, которым можно управлять через конфигурацию, и указать ему базу пользователей, с которой он будет работать
- Поведение сервера обычно можно дорабатывать подключая дополнительные свои, или разработанные кем-то то еще модули
- Гибкость таких решений крайне низкая, подойдет для сценариев предоставления OIDC для промышленных решений или продуктов во внутренней инфраструктуре требующих OIDC
- Можно настроить решение под определенную специфику, но для этого обычно нужно приложить много усилий

Пример: **Keycloak** (<https://www.keycloak.org>)

Turn key OIDC библиотека / фреймворк

- Идеален, если вы хотите построить OIDC вокруг своей единственной базы пользователей
- Есть ограниченный набор интерфейсов для типичных расширений, предусмотренных разработчиками
- Аутентификация заработает как только реализуете минимальные интерфейсы
- Если потребуется гибкость сверх предоставленной разработчики – возможность реализации под вопросом

Примеры:

Identity Server (<https://identityserver.io>)

OpenIddict (<https://github.com/openiddict>)

Низкоуровневые фреймворки OIDC

- Это тонкая прослойка между протоколом OpenID Connect и вашим кодом
- Аутентификация не заработает пока вы не реализуете код необходимых проверок во фреймворке
- У вас будет полный контроль и свобода во всех аспектах построения поведения аутентификации
- Рекомендуются для построения аутентификации Enterprise уровня, но требует серьезной работы

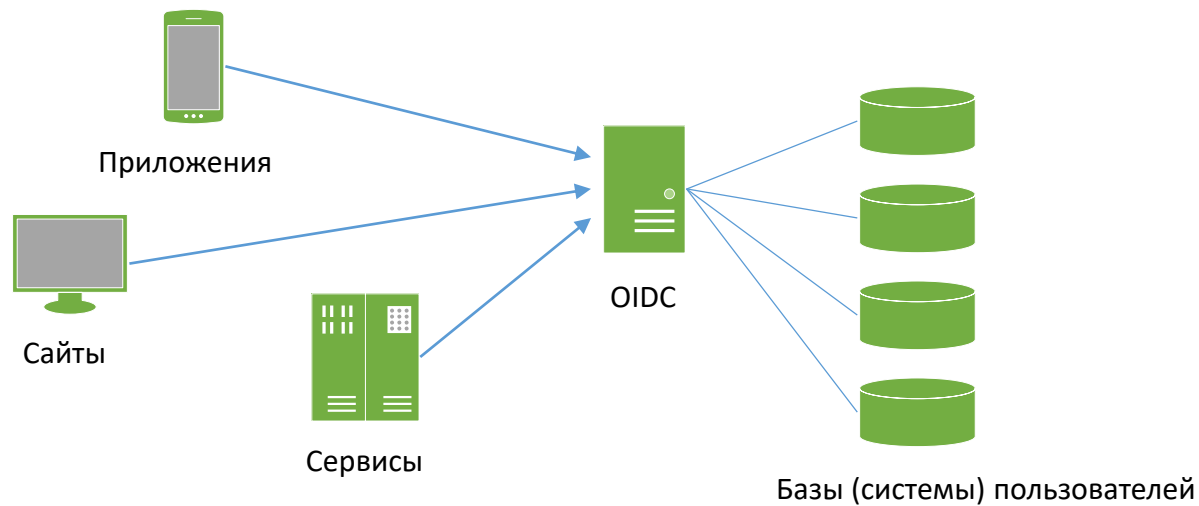
Пример: **AspNet Security OpenIdConnect Server**

(<https://github.com/aspnet-contrib/AspNet.Security.OpenIdConnect.Server>)

В чем ценность создания еще одного провайдера аутентификации OIDC?

OpenID Connect в Enterprise

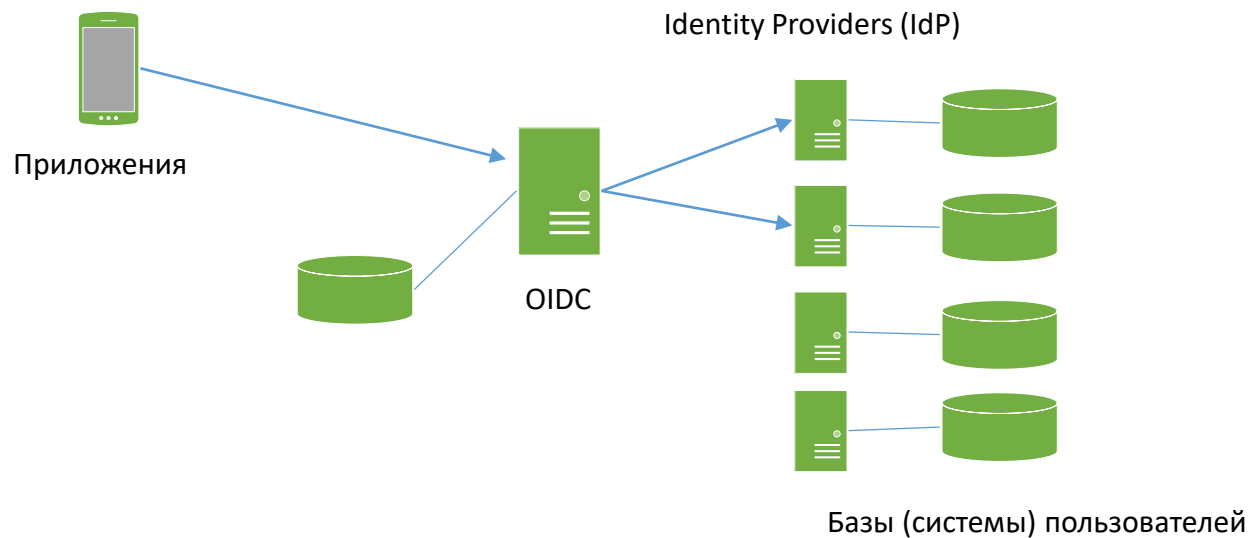
- В Enterprise гораздо больше одной базы пользователей
- Гораздо больше одного приложения, с различными настройками, использующие разные, пересекающиеся базы пользователей



- Появляется много нестандартных сценариев входа

OpenID Connect в Enterprise

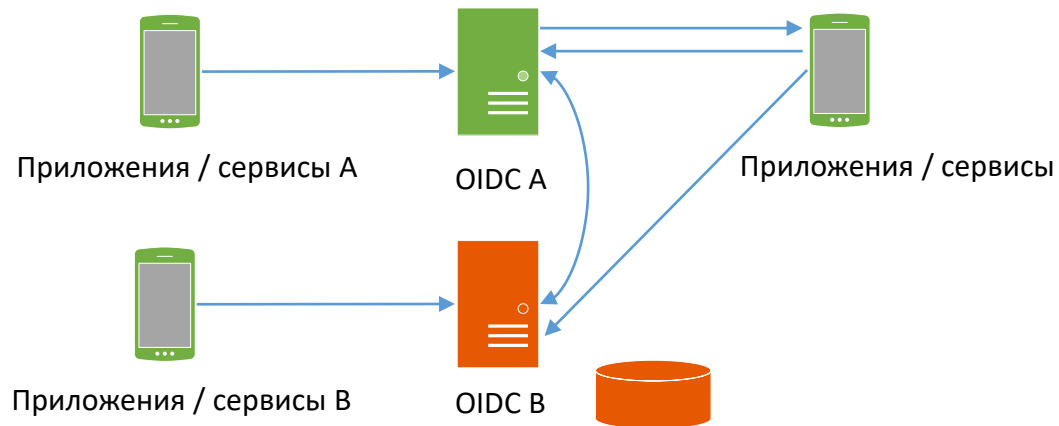
- Из OIDC выносятся специфика работы с конкретными хранилищами пользователей
- Приложение конфигурируется на OIDC для использования определенных IdP
- Возможность объединять Identities в Principal на OIDC



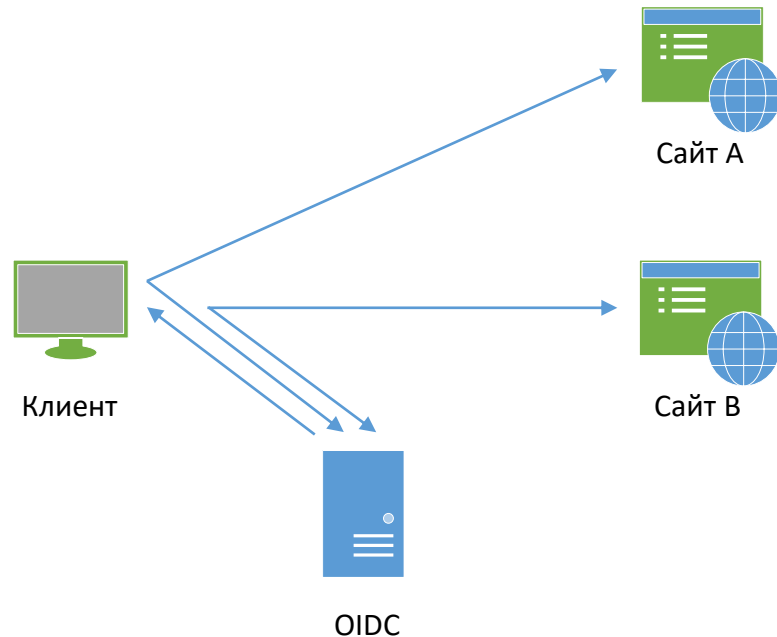
- Выбирается очень гибкий фреймворк для построения OIDC

Интеграция OIDC

- Если в периметре есть несколько OIDC, как их интегрировать?
- Разные приложения доверяют разным OIDC
- В случае использования аутентификации через браузер возможен federation, иначе будет компрометация данных аутентификации на промежуточном OIDC
- Интерфейс OIDC усложняет интеграцию, нужно поддерживать несколько OIDC
- Нужна ли интеграция?

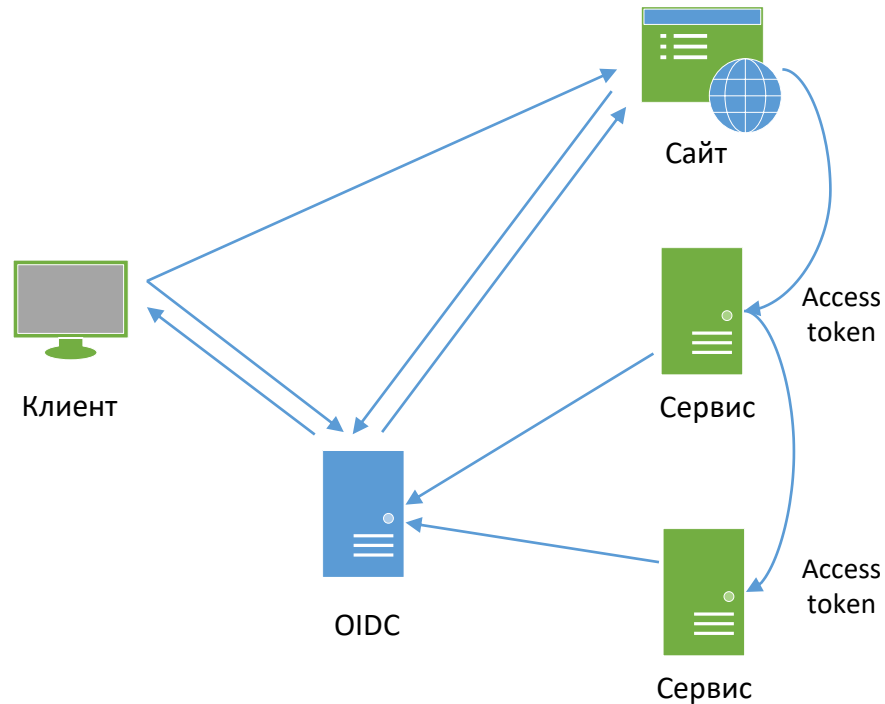


Single Sign On в OIDC



1. Клиент заходит на сайт A, который перенаправляет его на сервер OIDC
2. Пользователь идентифицирует себя используя логин и пароль
3. Сервер OIDC выполняет аутентификацию пользователя, и в случае успеха выдает Authentication Cookie зашивая в нее нужную ему техническую информацию, выполняя его вход на сайт OIDC
4. Клиент перенаправляется на сайт A
5. Клиент заходит на сайт B, который перенаправляет его на сервер OIDC
6. Пользователь уже имеет Authentication Cookie для сайта OIDC, из которой извлекается информация, и если она соответствует требованиям сайта B, то OIDC выдает доступ клиенту, либо запрашивает дополнительную аутентификацию

Делегирование авторизации в OIDC



1. Клиент заходит на сайт, выполнив аутентификацию на OIDC с кодом авторизации
2. Сайт обменивает код авторизации на токен доступа
3. Для выполнения задачи сайту необходимо обратиться к сервису
4. Он может использовать токен клиента (токен может быть получен для ресурса (audience))
5. Если в рамках выполнения задачи сервису нужно обратиться к дополнительному сервису, он может получить для него токен самостоятельно, но будет потеряна информация об исходном клиенте
6. Есть сценарий для сохранения информации об исходном клиенте “On-Behalf-Of”:
<https://docs.microsoft.com/en-us/azure/active-directory/develop/active-directory-v2-protocols-oauth-on-behalf-of>

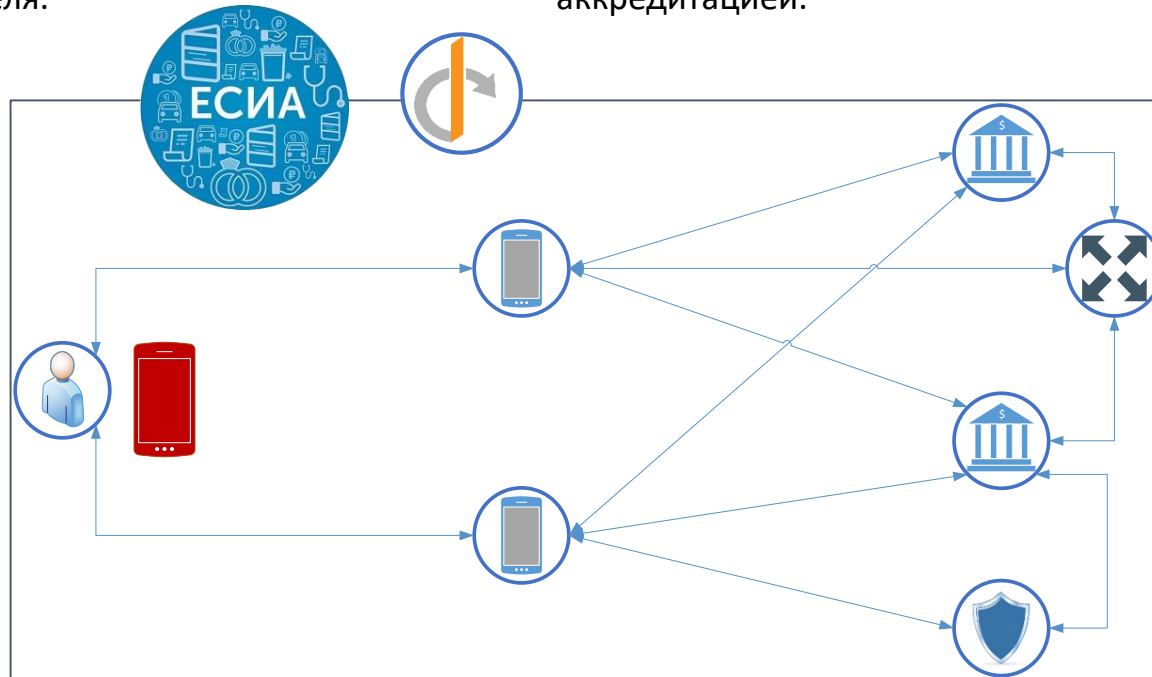
Open API



Пользователь финансового приложения, выступающий от своего имени, как ФЛ, или от имени ЮЛ, в качестве его представителя.

Субъект экономической деятельности, который создал финансовое приложение. Обладает соответствующей аккредитацией.

Субъект экономической деятельности, который имеет соответствующую лицензию на работу в сфере финансовых услуг, и владеет необходимой инфраструктурой для работы с финансовыми средствами клиентов.

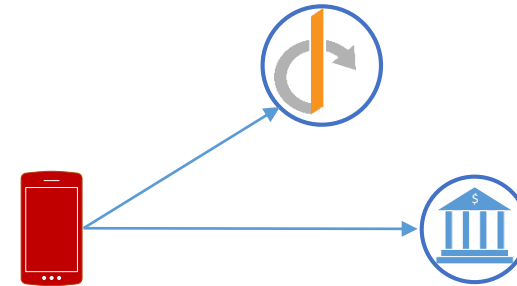


- Банки
- Страховые компании
- СБП
- и т.д.

Ак Барс Open API Хакатон

Планируем Open API Хакатон в поисках инновационных идей использования этой технологии!
В качестве материала мы планируем предоставить все! сущности имеющиеся в банке:
(в песочнице)

- Клиенты (юридические, физические лица)
- Персональная информация
- Расходы
- Документы
- Договора
- Депозиты, кредиты, кредитная история
- Сделки, потенциальные сделки
- Финансовые показатели
- Продукты
- Объекты недвижимости
- Претензии
- ...



Вы сами придумаете что можно создать имея доступ к этой информации!



Спасибо за внимание
