Synerdocs

электронный обмен документами

www.synerdocs.ru

Synerdocs
электронный обмен документами

Построение централизованной системы идентификации и аутентификации на базе **IdentityServer**

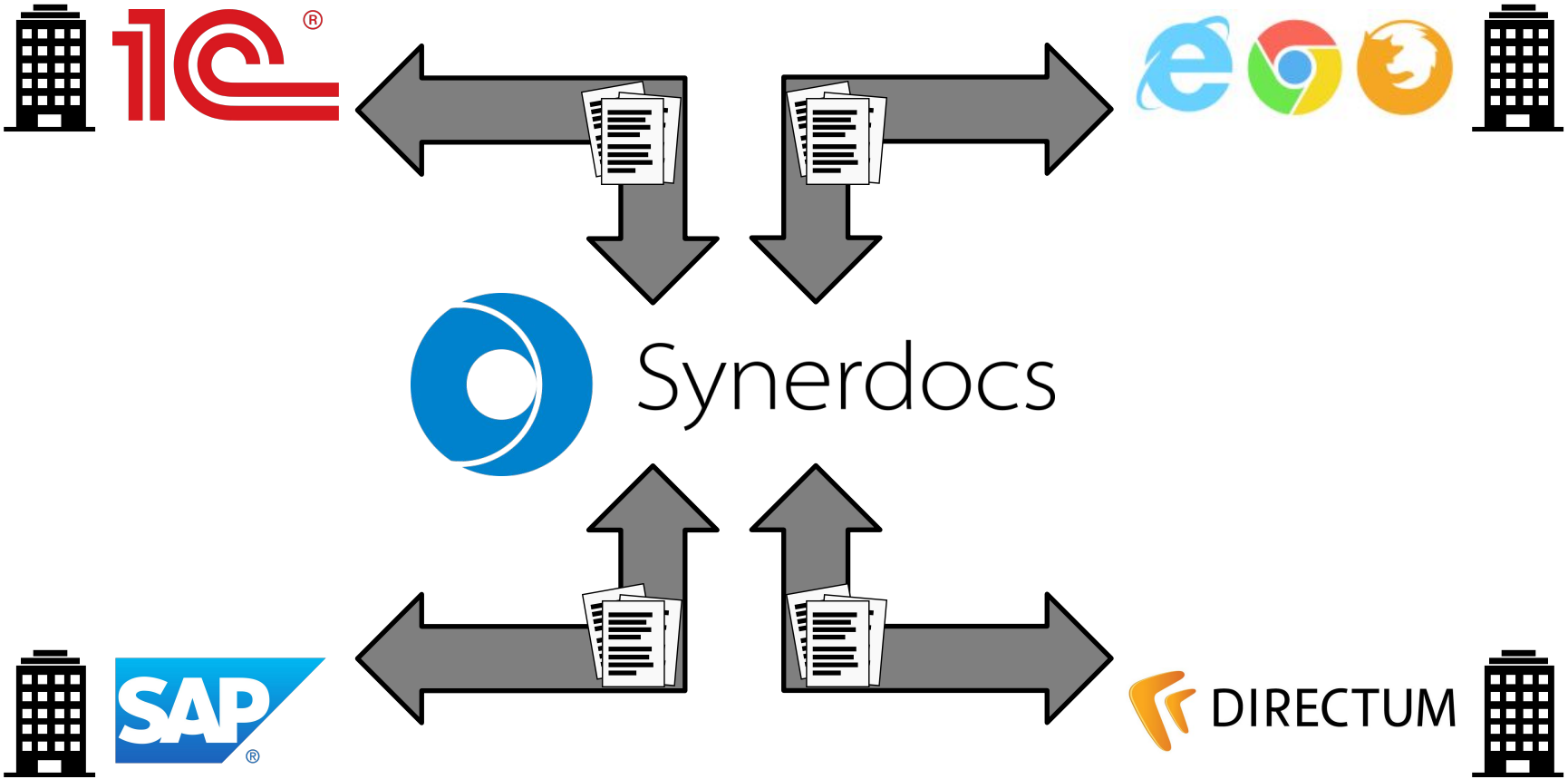
www.synerdocs.ru

Synerdocs
электронный обмен документами

О продукте

www.synerdocs.ru

Synerdocs
электронный обмен документами



www.synerdocs.ru

Synerdocs

Поиск документов

Иван Иванович Иванов
Тестовая организация №1, Головное подразделение

Отправить документ

- Входящие +749
- Исходящие
- Внутренние
- К отправке 145
- События +1656
- Контрагенты
- Настройки
- Справка
- Связаться с нами

Инструкция по созданию ТрН грузоотправителем

Уважаемые абоненты. Инструкция по созданию ТрН доступна для скачивания: Инструкция по созданию ТрН грузоотправителем

Порядок заполнения полей, описан в веб-справке сервиса.

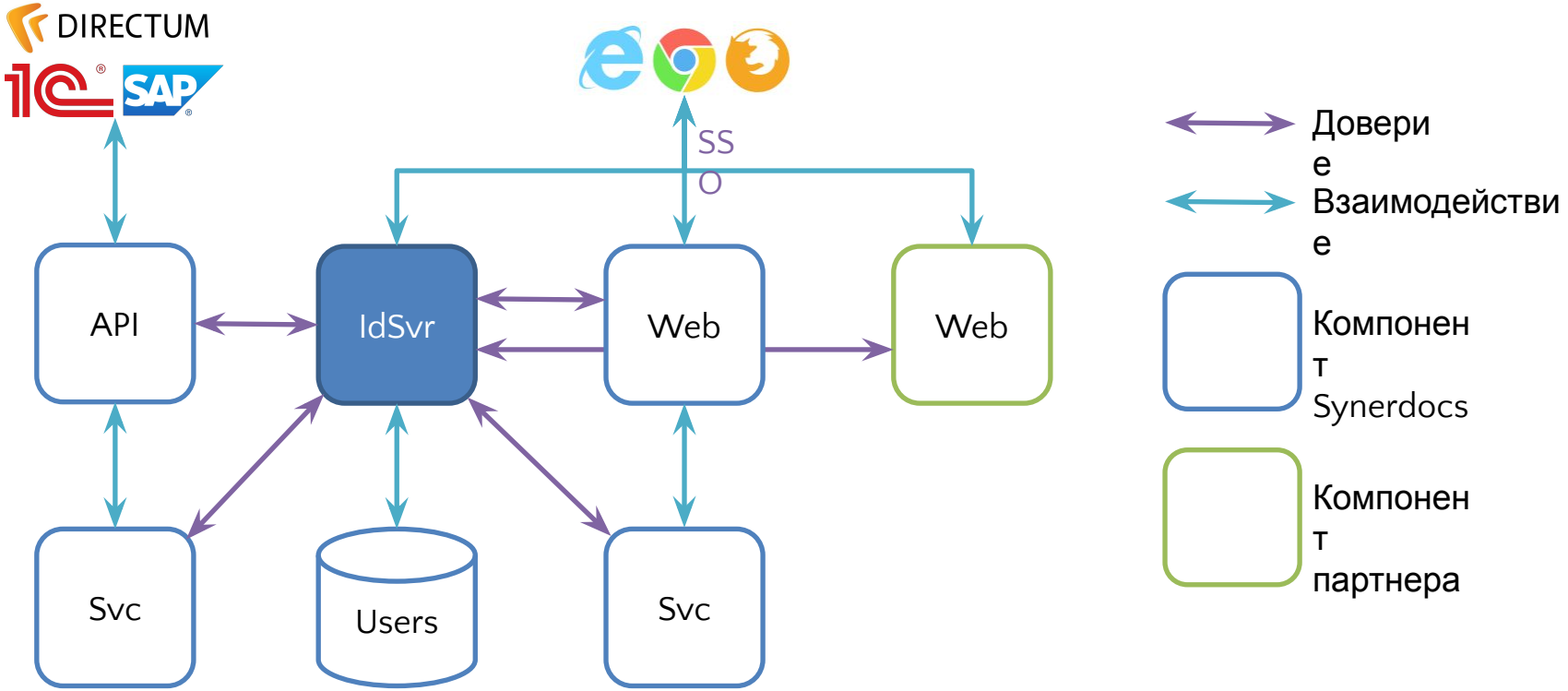
Подписать | Отказать | Аннулировать | Выгрузить | Переслать | Переместить | Согласовать | Еще | Фильтр

1-25 из 1209

Отправитель	Наименование документа	Сумма	Статус	Дата
Тестовая организация №2 , Головн... ↳ Головное подразделение	Счет-фактура и документ об отгрузке товаров (...	11 800,00р. НДС: 1 800,00р.	Выставлен	22.03.2019 17:25
	Счет-фактура и документ об отгрузке товаров (...	11 800,00р. НДС: 1 800,00р.	Выставлен	22.03.2019 17:25
	Счет-фактура и документ об отгрузке товаров (...	11 800,00р. НДС: 1 800,00р.	Выставлен	22.03.2019 17:25
ИП Павлов Игорь Петрович , Осно... ↳ Головное подразделение	Акт о выполнении работ (оказании услуг) № 15...	11 800,00р. НДС: 1 800,00р.	Требуется УОП	22.03.2019 17:24
Тестовая организация №2 , Головн... ↳ Головное подразделение	TestPlanTemplate_RUP.doc к документу Счет-фак...		Требуется подпись	22.03.2019 08:45
Тестовая организация №1 , Головно... ↳ Головное подразделение	Транспортная накладная № 201903140237150 от 14.... Груз сдан		Подписан	18.03.2019 08:44
Тестовая организация №1 , Головно... ↳ Головное подразделение	Транспортная накладная № 201903140237060 от 14.... Груз сдан		Подписан	18.03.2019 08:44
Тестовая организация №1 , Головно... ↳ Головное подразделение	Транспортная накладная № 201903140236570 от 14.... Груз сдан		Подписан	18.03.2019 08:44
Тестовая организация №1 , Головн... ↳ Головное подразделение	Транспортная накладная № 201903140236480 от... Груз сдан		Подписан	18.03.2019 08:44
Тестовая организация №1 , Головн... ↳ Головное подразделение	Транспортная накладная № 201903140236360 от... Груз сдан		Подписан	18.03.2019 08:43
Тестовая организация №1 , Головно... ↳ Головное подразделение	Транспортная накладная № 201903140236280 от 14.... Груз сдан		Подписан	18.03.2019 08:43
Тестовая организация №1 , Головн... ↳ Головное подразделение	Транспортная накладная № 201903140236140 от... Груз сдан		Подписан	18.03.2019 08:43
Тестовая организация №1 , Головн... ↳ Головное подразделение	Транспортная накладная № 201903140235580 от... Груз сдан		Подписан	18.03.2019 08:43
Тестовая организация №1 , Головн... ↳ Головное подразделение	Транспортная накладная № 201903140235330 от... Груз сдан		Подписан	18.03.2019 08:42

Synerdocs
электронный обмен документами

Архитектура



Synerdocs
электронный обмен документами

Терминология

www.synerdocs.ru

(I) Идентификация

– *Кто ты?*

(A) Аутентификация

– *Действительно ли ты являешься тем, за кого себя выдаешь?*

(A) Авторизация

– *Имеешь ли ты право выполнять данное действие?*

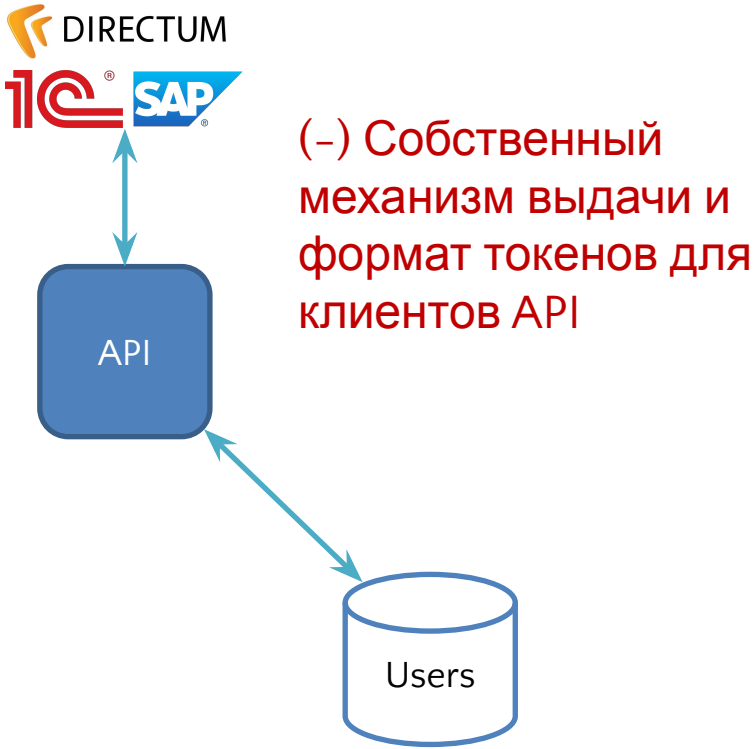
(D) Делегирование доступа

– *Предоставляешь ли ты этому субъекту свои права доступа?*

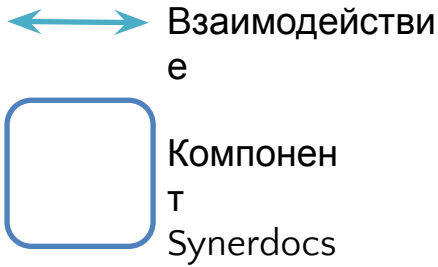
Synerdocs
электронный обмен документами

Предпосылки внедрения

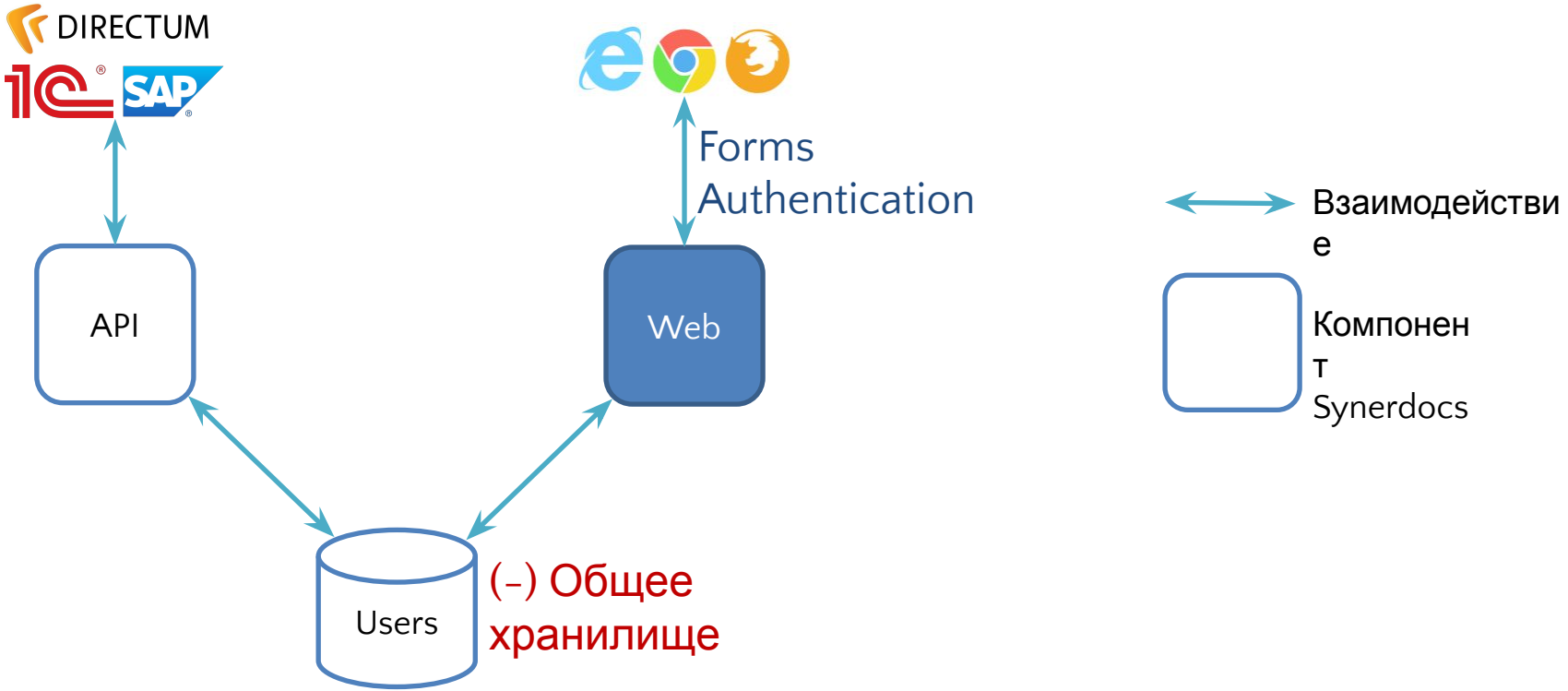
Единственное приложение API



(-) Собственный механизм выдачи и формат токенов для клиентов API

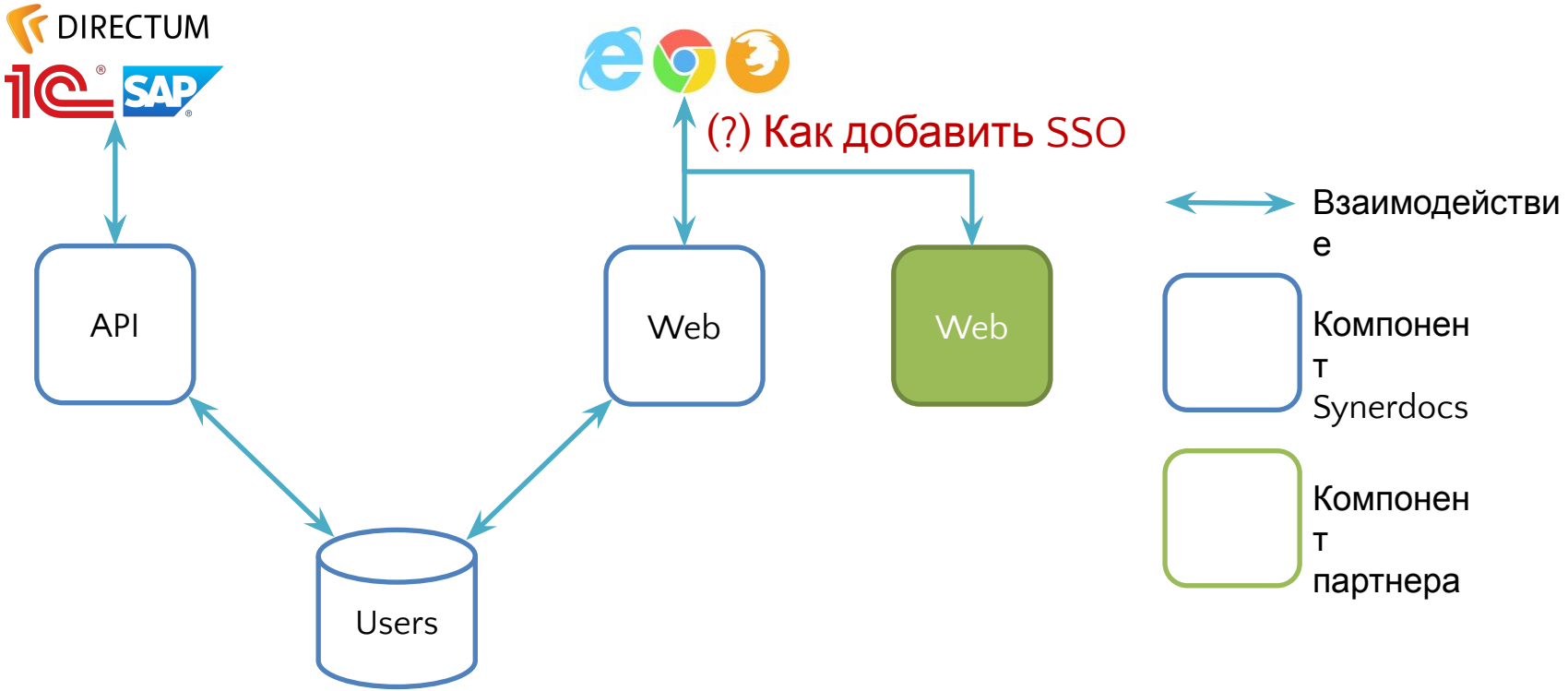


Добавление собственного веб клиента



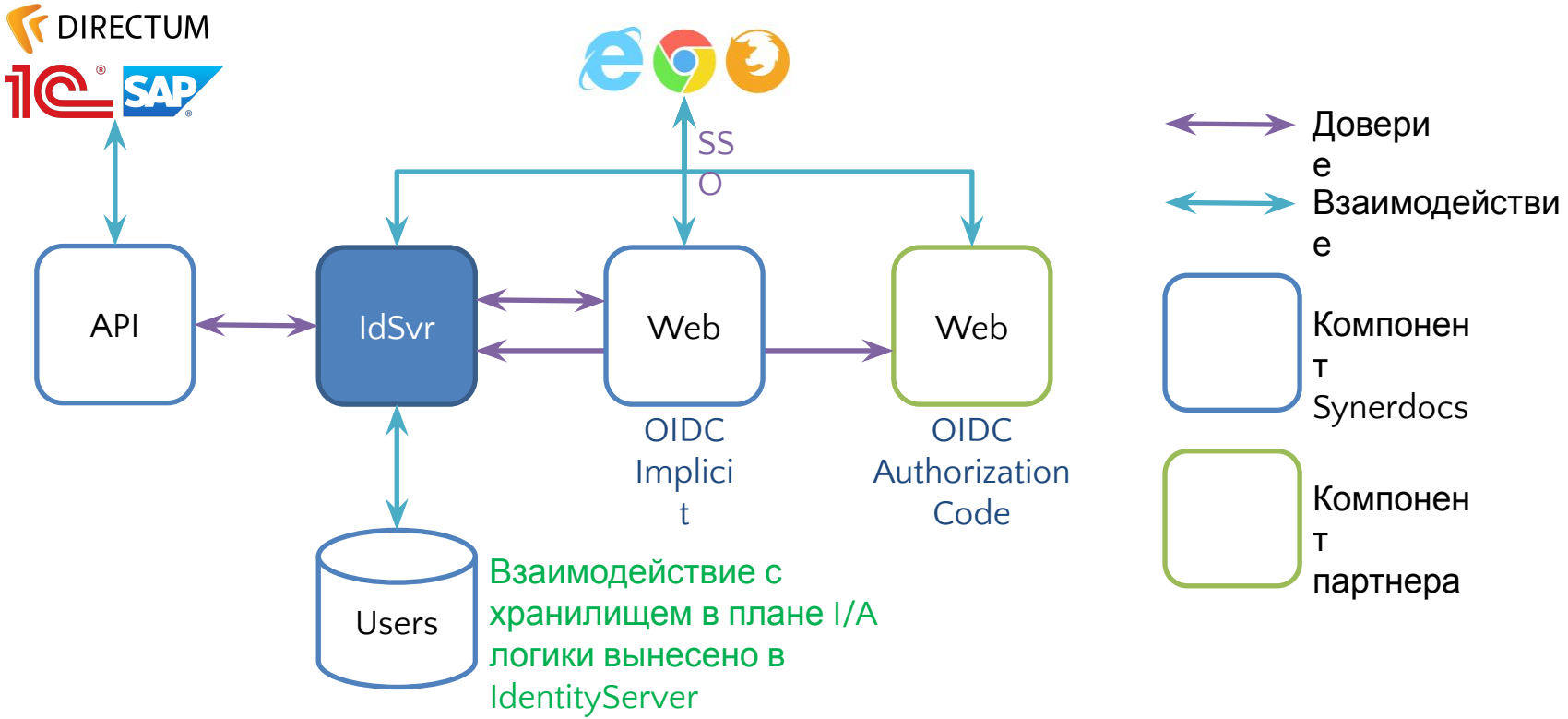
Synerdocs
электронный обмен документами

Добавление веб клиента партнера



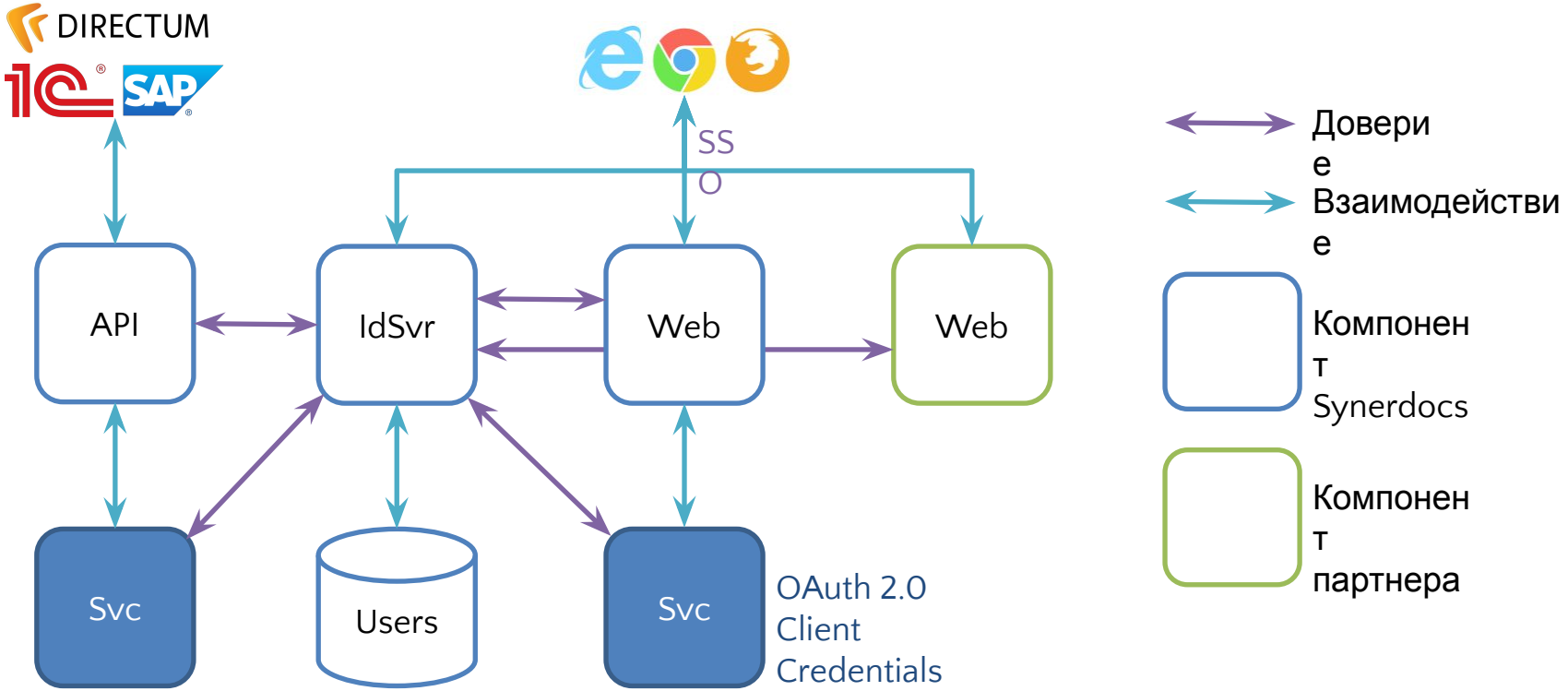
Synerdocs
электронный обмен документами

Добавление IdentityServer



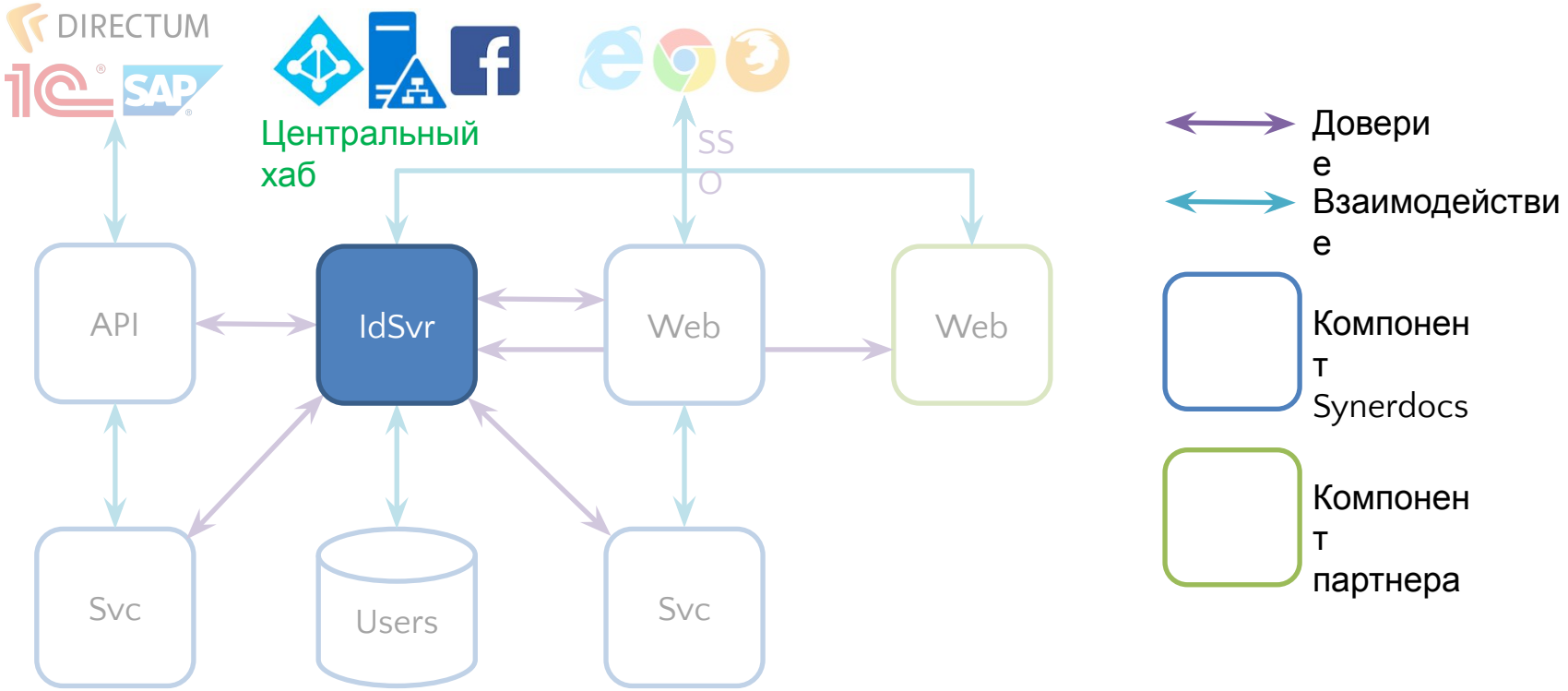
Synerdocs
электронный обмен документами

Добавление внутренних служб



Synerdocs
электронный обмен документами

Добавление внешних поставщиков



Synerdocs
электронный обмен документами

IdentityServer

www.synerdocs.ru

Общая информация

IdentityServer – это .NET Standard библиотека с открытым исходным кодом и развертываемый компонент (ASP.NET Core Middleware), реализующая следующие возможности:

Общая информация

IdentityServer – это .NET Standard библиотека с открытым исходным кодом и развертываемый компонент (ASP.NET Core Middleware), реализующая следующие возможности:

- Аутентификация на основе токенов

Общая информация

IdentityServer – это .NET Standard библиотека с открытым исходным кодом и развертываемый компонент (ASP.NET Core Middleware), реализующая следующие возможности:

- Аутентификация на основе токенов
- Технология единого входа (Single Sign-On - SSO)

Общая информация

IdentityServer – это .NET Standard библиотека с открытым исходным кодом и развертываемый компонент (ASP.NET Core Middleware), реализующая следующие возможности:

- Аутентификация на основе токенов
- Технология единого входа (Single Sign-On - SSO)
- Контроль доступа к API

Общая информация

IdentityServer – это .NET Standard библиотека с открытым исходным кодом и развертываемый компонент (ASP.NET Core Middleware), реализующая следующие возможности:

- Аутентификация на основе токенов
- Технология единого входа (Single Sign-On - SSO)
- Контроль доступа к API
- Стандарты OAuth 2.0 и OpenID Connect 1.0

История проекта

2012: IdentityServer1

2013: IdentityServer2

2015: IdentityServer3

2016: IdentityServer4

- OAuth 2.0 / OpenID Connect 1.0
- .NET Core 2.0 / ASP.NET Core 2.0

Как получить

Исходный код

GitHub / IdentityServer

NuGet пакет

NuGet / IdentityServer

Лицензия

Apache 2.0

Клиентские библиотеки

IdentityServer реализует только серверную часть стандартов. Для взаимодействия со стороны клиента нужно воспользоваться следующими библиотеками:

Классические ASP.NET / MVC / Web API приложения:

- NuGet / OWIN / OAuth 2.0
- NuGet / OWIN / OpenID Connect 1.0

ASP.NET Core приложения:

- NuGet / ASP.NET Core / OAuth 2.0
- NuGet / ASP.NET Core / OpenID Connect 1.0

Synerdocs
электронный обмен документами

Реализованные спецификации

www.synerdocs.ru

OpenID Connect 1.0

- OpenID Connect Core 1.0 (spec)
- OpenID Connect Discovery 1.0 (spec)
- OpenID Connect Session Management 1.0 - draft 28 (spec)
- OpenID Connect Front-Channel Logout 1.0 - draft 02 (spec)
- OpenID Connect Back-Channel Logout 1.0 - draft 04 (spec)

OAuth 2.0

- OAuth 2.0 (RFC 6749)
- OAuth 2.0 Bearer Token Usage (RFC 6750)
- OAuth 2.0 Multiple Response Types (spec)
- OAuth 2.0 Form Post Response Mode (spec)
- OAuth 2.0 Token Revocation (RFC 7009)
- OAuth 2.0 Token Introspection (RFC 7662)
- Proof Key for Code Exchange (RFC 7636)
- JSON Web Tokens for Client Authentication (RFC 7523)
- OAuth 2.0 Device Flow for Browserless and Input Constrained Devices (draft)

Synerdocs
электронный обмен документами

Суммарно это около **300** страниц текста

Можно ли быстро и корректно реализовать их самостоятельно?

www.synerdocs.ru

Synerdocs
электронный обмен документами

Токены

www.synerdocs.ru

Access Token, Identity Token, Reference Token, Bearer Token, JWT ...

Как разработчику, начинающему работать с этими стандартами, разобраться со всем этим многообразием?

Попробуем систематизировать токены, выполнив их классификацию.

Synerdocs
электронный обмен документами

Классификация по назначению

Токен удостоверения (Identity Token)

Id_token

Хранит учетные данные пользователя

Токен доступа (Access Token)

access_token

Предоставляет делегированный доступ

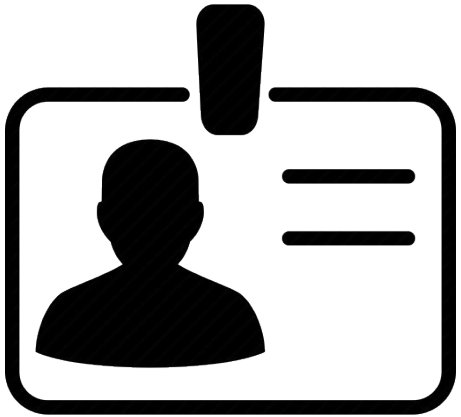
Токен обновления (Refresh Token)

refresh_token

Служит для получения нового токена доступа

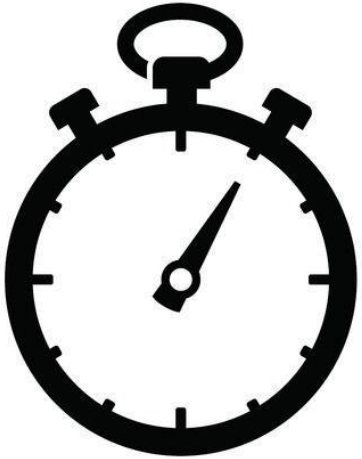
Классификация по назначению

Токен удостоверения (Identity Token)



Токен доступа (Access Token)

Токен обновления (Refresh Token)



Synerdocs
электронный обмен документами

Классификация по способу передачи

Передача по значению (By Value)

- Автономный или прозрачный токен (Self-Containing or Transparent Token)
- Содержит в себе все данные и подпись
- Для проверки и получения данных не требуется обращения к IdentityServer
- Трудно выполнить отзыв токена

Передача по ссылке (By Reference)

- Ссылочный или непрозрачный токен (Reference or Opaque Token)
- Содержит только ИД записи с данными
- Для проверки и получения данных требуется обращение к IdentityServer
- Просто выполнить отзыв токена

Классификация по способу передачи

Передача по значению (By Value)

Передача по ссылке (By Reference)

Классификация по способу использования

Токен на предъявителя (Bearer Token)

Для использования токена достаточно его предъявления, например в HTTP заголовке

Аналогии:
Наличные деньги
Ценные бумаги на предъявителя

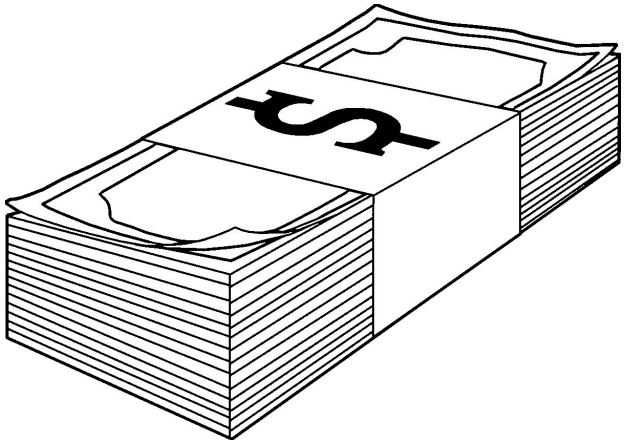
Токен владельца ключа (Holder of Key Token)

Для использования токена требуется дополнительная проверка его владельца

Аналогии:
Банковская карта с PIN-кодом
Именные ценные бумаги

Классификация по способу использования

Токен на предъявителя (Bearer Token)



Токен владельца ключа (Holder of Key Token)

Synerdocs
электронный обмен документами

Классификация по формату

JWT (JSON Web Token)

Заголовок (JSON)
Данные (JSON)
Подпись (Binary)

Есть возможность создать незащищенный токен (без подписи/шифрования или с ненадежной комбинацией алгоритмов)

Широко используется и поддерживается библиотеками

PASETO (Platform-Agnostic SEcurity TOkens)

Назначение (String)
Версия (String)
Заголовок (JSON)
Данные (JSON)
Подпись (Binary)
Подвал (JSON)

Практически невозможно создать незащищенный токен

Достаточно новый формат, который пока мало распространен

Synerdocs
электронный обмен документами

Классификация по формату

JWT (JSON Web Token)

HEADER
PAYLOAD
SIGNATURE

PASETO
(Platform-Agnostic
SEcurity TOkens)

LOCATION
VERSION
HEADER
PAYLOAD
SIGNATURE
FOOTER

Synerdocs
электронный обмен документами

Внедрение IdentityServer

www.synerdocs.ru

Synerdocs
электронный обмен документами

Серверное приложение

www.synerdocs.ru

Создание и настройка приложения с IdentityServer

```
public class Startup
{
    public void ConfigureServices(IServiceCollection services)
    {
        services.AddMvc();
        services.AddIdentityServer();
    }

    public void Configure(IApplicationBuilder app)
    {
        app.UseStaticFiles();
        app.UseIdentityServer();
        app.UseMvcWithDefaultRoute();
    }
}
```

Создание и подключение X509 сертификата

```
public class Startup
{
    public IHostingEnvironment Environment { get; }
    public void ConfigureServices(IServiceCollection services)
    {
        // ...
        var builder = services.AddIdentityServer();
        if (Environment.IsDevelopment())
            builder.AddDeveloperSigningCredential();
        else
            builder.AddSigningCredential(GetSigningCertificate());
    }

    private X509Certificate2 GetSigningCertificate()
    {
        // ...
    }
}
```

Synerdocs
электронный обмен документами

Объявление клиентских приложений

```
yield return new Client {
    ClientId = "web_client_id",
    AllowedGrantTypes = GrantTypes.Implicit,
    AllowedScopes = { "openid" },
    RedirectUris = { "https://web-client.synerdocs.ru/app/auth" },
    RequireConsent = false };
yield return new Client {
    ClientId = "api_client_id",
    AllowedGrantTypes = GrantTypes.Code,
    ClientSecrets = { new Secret("api_client_secret".Sha256()) },
    AllowedScopes = { "offline_access", "api_method" },
    RedirectUris = { "https://api-client.synerdocs.ru/app/auth" },
    AccessTokenType = AccessTokenType.Reference };
yield return new Client {
    ClientId = "svc_client_id",
    AllowedGrantTypes = GrantTypes.ClientCredentials,
    ClientSecrets = { new Secret("svc_client_secret".Sha256()) },
    AllowedScopes = { "svc_method" },
    AccessTokenType = AccessTokenType.Jwt };
```

Synerdocs
электронный обмен документами

UI

www.synerdocs.ru

Разработка всего UI полностью под ответственностью разработчика

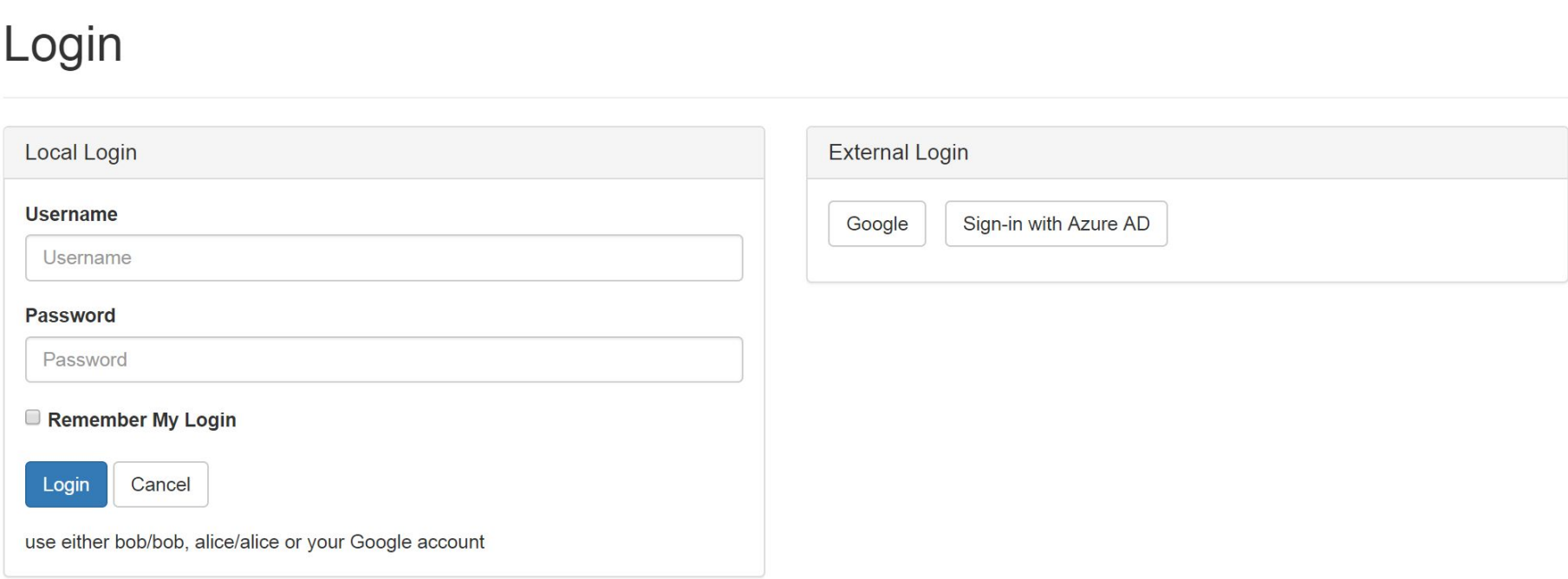
На GitHub доступен пример построения UI на базе

ASP.NET Core MVC, Bootstrap и jQuery

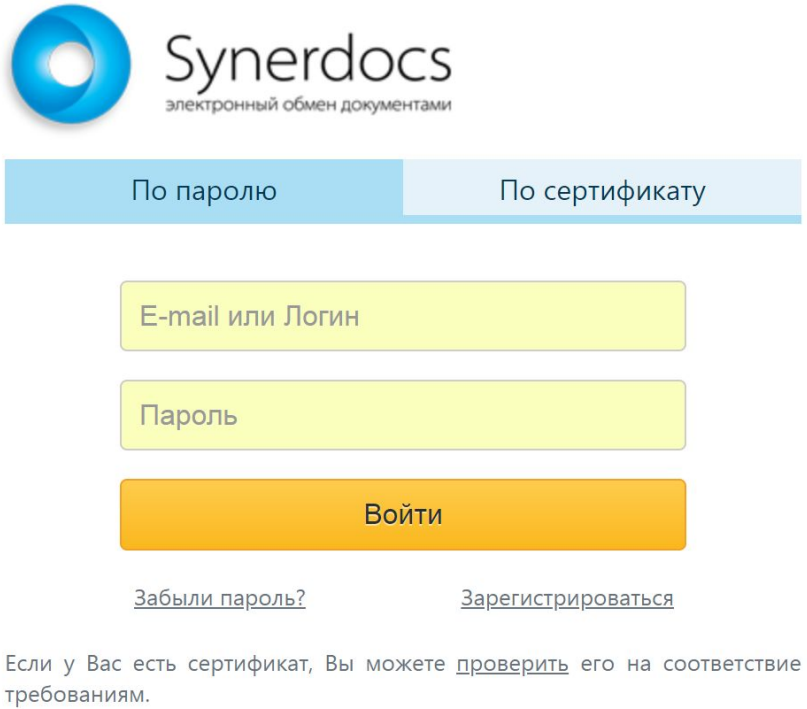
https://github.com/IdentityServer/IdentityServer4.Quickstart.UI

Synerdocs
электронный обмен документами

UI в примере



UI в Synerdocs



Synerdocs
электронный обмен документами

Хранилище пользователей

www.synerdocs.ru

Подключение хранилища пользователей

Для ASP.NET Membership / ASP.NET Identity готовые решения:

- https://www.nuget.org/packages/IdentityServer4.Contrib.Membership/
- https://github.com/IdentityServer/IdentityServer4.AspNetIdentity

В других случая потребуется самостоятельно реализовать две точки расширения:

- IResourceOwnerPasswordValidator
- IProfileService

IResourceOwnerPasswordValidator

```
public class ResourceOwnerPasswordValidator : IResourceOwnerPasswordValidator
{
    public IUserService UserService { get; set; } // DI.
    public Task ValidateAsync(ResourceOwnerPasswordValidationContext context)
    {
        var user = UserService.ValidateUser(context.UserName, context.Password);
        if (user == null)
            context.Result = new GrantValidationResult(
                TokenRequestErrors.InvalidGrant,
                "Incorrect username or password");
        else
            context.Result = new GrantValidationResult(user.UserId.ToString(), "custom",
                new[]
                {
                    new Claim(JwtClaimTypes.Name, user.Name),
                    new Claim(JwtClaimTypes.Email, user.Email),
                });
        return Task.CompletedTask;
    }
}
```

IProfileService

```
public class ProfileService : IProfileService
{
    public IUserService UserService { get; set; } // DI.
    public Task GetProfileDataAsync(ProfileDataRequestContext context)
    {
        var userId = context.Subject.Claims
            .FirstOrDefault(x => x.Type == JwtClaimTypes.Subject);
        if (string.IsNullOrEmpty(userId?.Value))
            return Task.CompletedTask;
        var user = UserService.GetUser(int.Parse(userId.Value));
        context.IssuedClaims = new[]
            {
                new Claim(JwtClaimTypes.FirstName, user.FirstName),
                new Claim(JwtClaimTypes.LastName, user.LastName),
                new Claim(JwtClaimTypes.MiddleName, user.MiddleName),
            }
            .Where(x => context.RequestedClaimTypes.Contains(x.Type))
            .ToList();
        return Task.CompletedTask;
    }
}
```

Synerdocs
электронный обмен документами

Клиентские приложения

www.synerdocs.ru

Synerdocs
электронный обмен документами

Web клиент Synerdocs
OIDC 1.0 – OWIN Security Middleware

www.synerdocs.ru

Подключение Cookie Authentication

```
public class Startup
{
    public void Configuration(IAppBuilder app)
    {
        // ...
        app.UseCookieAuthentication(new CookieAuthenticationOptions
        {
            AuthenticationType = "Cookies",
            ExpireTimeSpan = TimeSpan.FromHours(8),
            SlidingExpiration = false,
        });
        // ...
    }
}
```

Подключение OpenID Connect Authentication

```
public class Startup
{
    public void Configuration(IAppBuilder app)
    {
        // ...
        app.UseOpenIdConnectAuthentication(new
OpenIdConnectAuthenticationOptions
        {
            Authority = "https://identity.synerdocs.ru",
            RedirectUri = "https://web-client.synerdocs.ru/app/auth",
            ClientId = "web_client_id",
            ResponseType = "id_token",
            SignInAsAuthenticationType = "Cookies",
            Scope = "openid",
            UseTokenLifetime = false,
        });
        // ...
    }
}
```

Synerdocs
электронный обмен документами

Отключение аутентификации на основе форм

```
<authentication
    mode="Forms">
    <forms
        timeout="2880"
        loginUrl="~/Account/LogOn" />
</authentication>
```

Synerdocs
электронный обмен документами

Web клиент партнера
OAuth 2.0 – Authorization Code Grant

www.synerdocs.ru

Перенаправление и обратный вызов

```
private ActionResult RedirectToAuthority()
{
    return Redirect("https://identity.synerdocs.ru"
        + "/connect/authorize"
        + "?response_type=code"
        + "&client_id=api_client_id"
        + "&redirect_uri=https://api-client.synerdocs.ru/app/auth"
        + "&state=data"
        + "&scope=offline_access api_method");
}

public async Task<ActionResult> Auth(string code, string state, string error)
{
    await TakeAccessToken(code);
    return RedirectToAction("Index");
}
```

Synerdocs
электронный обмен документами

Получение токена доступа

```
private async Task TakeAccessToken(string authorizationCode)
{
    using (var httpClient = new HttpClient())
    {
        httpClient.BaseAddress = new Uri("https://identity.synerdocs.ru");
        var httpResponse = await httpClient.PostAsync("connect/token",
            new FormUrlEncodedContent(new[]
            {
                new KeyValuePair<string, string>("code", authorizationCode),
                new KeyValuePair<string, string>("grant_type", "authorization_code"),
                new KeyValuePair<string, string>("client_id", "api_client_id"),
                new KeyValuePair<string, string>("client_secret", "api_client_secret"),
                new KeyValuePair<string, string>("redirect_uri",
                    "https://api-client.synerdocs.ru/app/auth"),
            }));
        _tokenResponse = JsonConvert.DeserializeObject<TokenResponse>(
            await httpResponse.Content.ReadAsStringAsync());
    }
}
```

Вызов метода API

```
private async Task CallApiMethod()
{
    using (var httpClient = new HttpClient())
    {
        httpClient.BaseAddress = new Uri("https://api.synerdocs.ru");
        httpClient.DefaultRequestHeaders.Authorization
            = new AuthenticationHeaderValue("Bearer",
                _tokenResponse.AccessToken);
        var httpResponse = await httpClient.PostAsync("api/method",
            new FormUrlEncodedContent(new[]
            {
                new KeyValuePair<string, string>("param1", "val1"),
                new KeyValuePair<string, string>("param2", "val2"),
            }));
        _methodResult = await httpResponse.Content.ReadAsStringAsync();
    }
}
```

Synerdocs
электронный обмен документами

Обновление токена доступа

```
private async Task RefreshAccessToken()
{
    using (var httpClient = new HttpClient())
    {
        httpClient.BaseAddress = new Uri("https://identity.synerdocs.ru");
        var httpResponse = await httpClient.PostAsync("connect/token",
            new FormUrlEncodedContent(new[]
            {
                new KeyValuePair<string, string>("grant_type",
"refresh_token"),
                new KeyValuePair<string, string>("client_id",
"api_client_id"),
                new KeyValuePair<string, string>("client_secret",
                   "api_client_secret"),
                new KeyValuePair<string, string>("refresh_token",
                   _tokenResponse.RefreshToken),
            }));
        _tokenResponse = JsonConvert.DeserializeObject<TokenResponse>(
            await httpResponse.Content.ReadAsStringAsync());
    }
}
```

Synerdocs
электронный обмен документами

Внутренние службы
OAuth 2.0 – Client Credentials Grant

www.synerdocs.ru

Synerdocs
электронный обмен документами

Получение токена доступа

```
private async Task TakeAccessToken()
{
    using (var httpClient = new HttpClient())
    {
        httpClient.BaseAddress = new Uri("https://identity.synerdocs.ru");
        var httpResponse = await httpClient.PostAsync("connect/token",
            new FormUrlEncodedContent(new[]
            {
                new KeyValuePair<string, string>("grant_type",
                    "client_credentials"),
                new KeyValuePair<string, string>("client_id",
"svc_client_id"),
                new KeyValuePair<string, string>("client_secret",
                    "svc_client_secret"),
                new KeyValuePair<string, string>("scope", "svc_method"),
            }));
        _tokenResponse = JsonConvert.DeserializeObject<TokenResponse>(
                await httpResponse.Content.ReadAsStringAsync());
    }
}
```

Вызов метода службы

```
private async Task CallServiceMethod()
{
    using (var httpClient = new HttpClient())
    {
        httpClient.BaseAddress = new Uri("https://svc.synerdocs.ru");
        httpClient.DefaultRequestHeaders.Authorization
            = new AuthenticationHeaderValue("Bearer",
                _tokenResponse.AccessToken);
        var httpResponse = await httpClient.PostAsync("svc/method",
            new FormUrlEncodedContent(new[]
            {
                new KeyValuePair<string, string>("param1", "val1"),
                new KeyValuePair<string, string>("param2", "val2"),
            }));
        _methodResult = await httpResponse.Content.ReadAsStringAsync();
    }
}
```

Synerdocs
электронный обмен документами

Трудоемкость

Серверное приложение
1 человеко-неделя

Разработка UI
2 человеко-недели

Клиентские приложения
1 человеко-неделя

Подводные камни и особенности
2 человеко-недели

Synerdocs
электронный обмен документами

Подводные камни и особенности

www.synerdocs.ru

Трансформация claim'ов

```
JwtSecurityTokenHandler.InboundClaimTypeMap.Clear();
oidcOptions.Notifications = new OpenIdConnectAuthenticationNotifications
{
    SecurityTokenValidated = ctx =>
    {
        // Для корректной работы AuthorizationAttribute и ClaimsIdentity.Name.
        var oldIdentity = ctx.AuthenticationTicket.Identity;
        var newIdentity = new ClaimsIdentity(
            oldIdentity.FindAll(_ => NecessaryClaimNames.Contains(_.Type)),
            oldIdentity.AuthenticationType,
            nameType: "login",
            roleType: "role");
        ctx.AuthenticationTicket = new AuthenticationTicket(newIdentity,
            ctx.AuthenticationTicket.AuthenticationProperties);
        return Task.FromResult(0);
    },
};
```

Synerdocs
электронный обмен документами

Перенаправление после выхода

```
oidcOptions.Notifications = new OpenIdConnectAuthenticationNotifications
{
    SecurityTokenValidated = ctx =>
    {
        var oldIdentity = ctx.AuthenticationTicket.Identity;
        var newIdentity = new ClaimsIdentity(oldIdentity.Claims,
            oldIdentity.AuthenticationType);
        newIdentity.AddClaim(new Claim("id_token", ctx.ProtocolMessage.IdToken));
        ctx.AuthenticationTicket = new AuthenticationTicket(newIdentity,
            ctx.AuthenticationTicket.AuthenticationProperties);
        return Task.FromResult(0);
    },
    RedirectToIdentityProvider = ctx =>
    {
        // Для корректной работы перенаправления после выхода.
        ctx.ProtocolMessage.IdTokenHint
            = ctx.OwinContext.Authentication.User.FindFirst("id_token").Value;
        return Task.FromResult(0);
    },
};
```

Обработка nonce cookie

```
protected override void RememberNonce(OpenIdConnectMessage message, string nonce)
{
    var cookieOptions = new CookieOptions { HttpOnly = true, Secure = Request.IsSecure };
    if (!Options.NonceCookiePath.IsNullOrWhiteSpace())
        // Задаем отдельный путь для каждого сайта, если они развернуты на одном домене.
        cookieOptions.Path = Options.NonceCookiePath;
    if (Options.NonceThreshold > 0)
    {
        // Ограничиваем максимальное количество nonce cookie для одного клиента.
        var noncePrefix = GetNonceCookiePrefix();
        var oldNonces = Request.Cookies
            .Where(_ => _.Key.StartsWith(noncePrefix))
            .ToList();
        if (oldNonces.Count >= Options.NonceThreshold)
            for (var i = 0; i <= oldNonces.Count - Options.NonceThreshold; i++)
                Response.Cookies.Delete(oldNonces[i].Key, cookieOptions);
    }
    base.RememberNonce(message, nonce);
}
```

Другие ограничения

Отсутствие встроенной поддержки более старых протоколов WS-Trust, WS-Federation, SAML 1.1/2.0, OAuth 1.0, OpenID 1.0/2.0 и других

- Отдельный продукт IdentityServer4 SAML 2.0
- Отдельный продукт IdentityServer4 WS-Federation

Отсутствие встроенной панели управления

- Отдельный продукт AdminUI

Другие ограничения

Отсутствие поддержки IdP Initiated SSO в OpenID Connect 1.0

- Ведутся работы по стандартизации в черновике https://tools.ietf.org/html/draft-bradley-oauth-jwt-encoded-state-09#section-4.3

Katana Project на текущий момент почти не развивается и есть дефекты, а также проблемы интеграции OWIN и ASP.NET

- Переход на ASP.NET Core

Synerdocs
электронный обмен документами

Выводы

- ✔ Самостоятельно строить с нуля централизованную систему идентификации и аутентификации весьма сложно и рискованно

- ✔ Аналогично обстоит дело с самостоятельной реализацией таких стандартов, как OAuth 2.0 и OpenID Connect 1.0

- ✔ IdentityServer позволяет значительно упростить две перечисленные выше задачи, но понимание стандартов также может существенно помочь

Synerdocs
электронный обмен документами

Спасибо за внимание!
Ваши вопросы?

www.synerdocs.ru