Публичные контракты, как обеспечить их согласованность

Романюк Антон

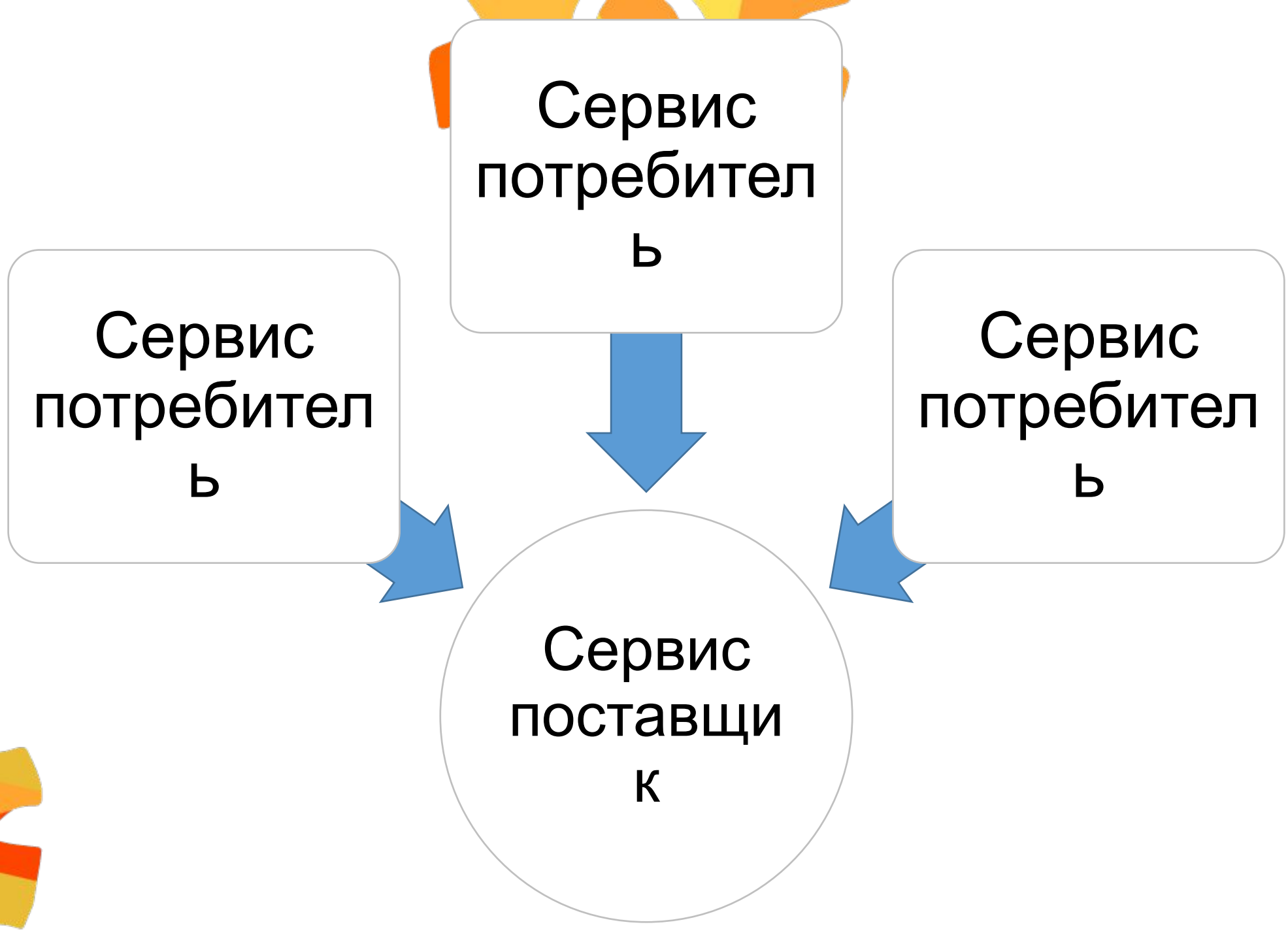
Apache Thrift™

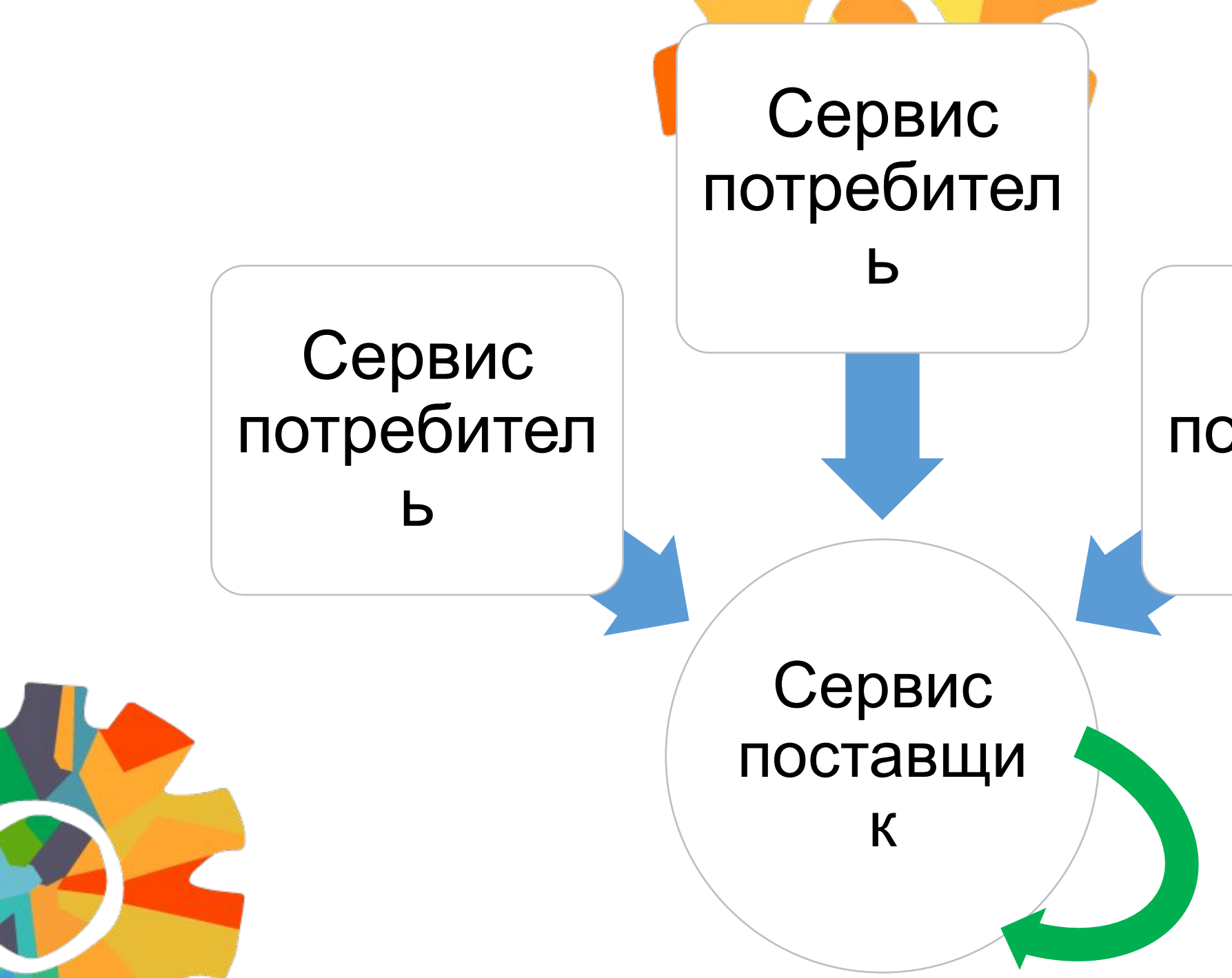
Swagger Supported by SMARTBEAR

gRPC

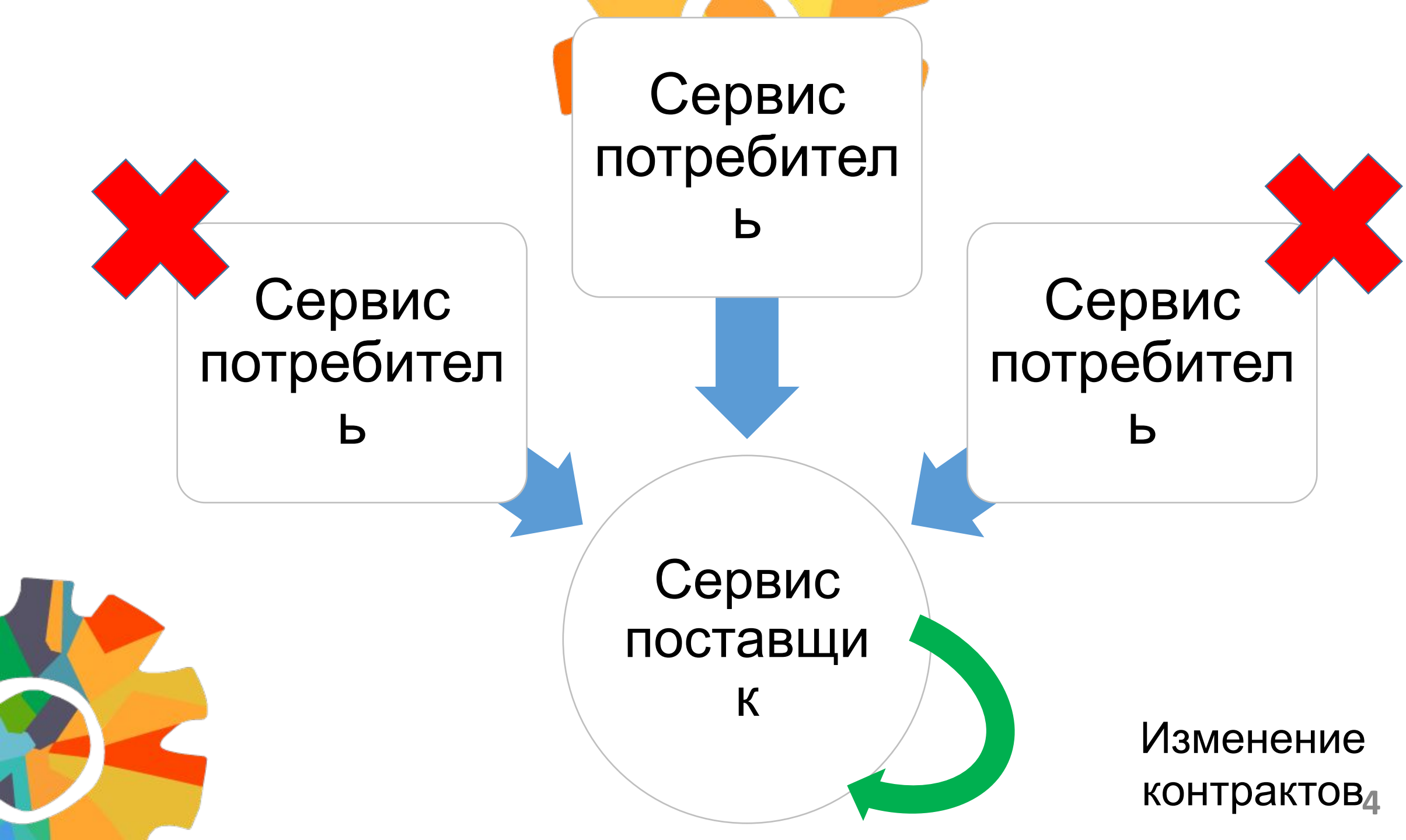
OPEN API INITIATIVE

PACTS


Сервис потребитель
Сервис потребитель
Сервис потребитель
Сервис поставщик

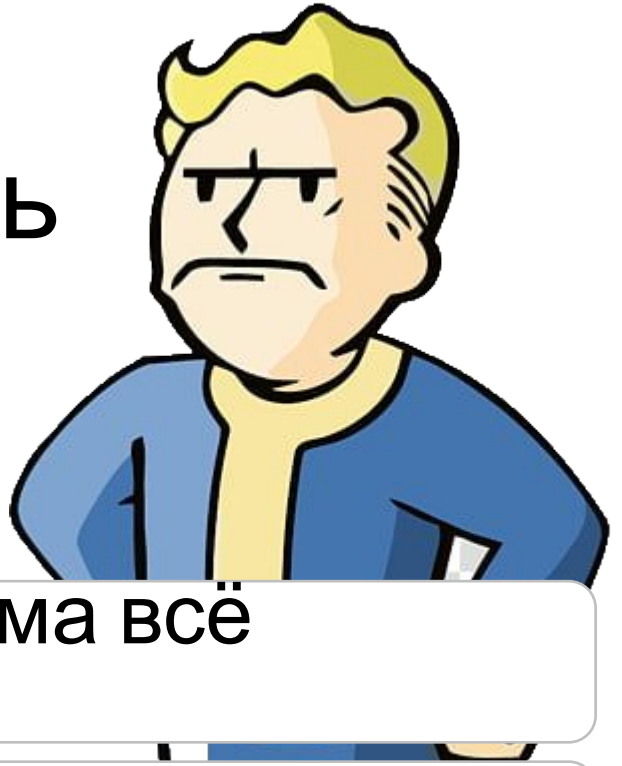


Изменение контрактов



Изменение контрактов

Нужно что-то предпринять



- [Красноречие] Команда сервиса поставщика сама всё поправит
- [Удача] Актуализировать код своего сервиса
- [Интеллект] Как сделать так, чтобы такое больше не повторялось?
- Сарказм

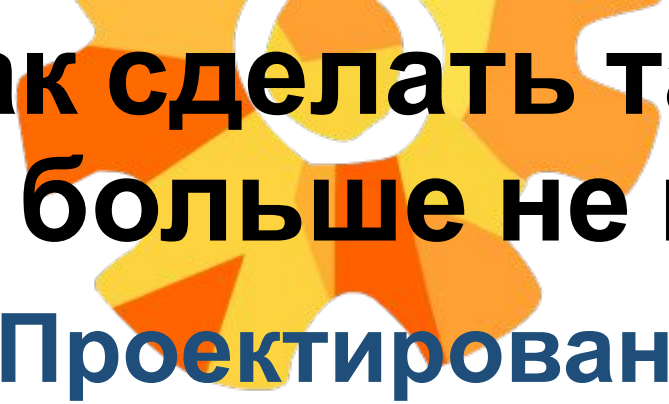
Нужно что-то предпринять

[Красноречие] Команда сервиса поставщика сама всё поправит

[Удача] Актуализировать код своего сервиса

[Интеллект] Как сделать так, чтобы такое больше не повторялось?

Сарказм



Как сделать так, чтобы такое больше не повторялось

Проектирование

- Унифицировать описание контрактов

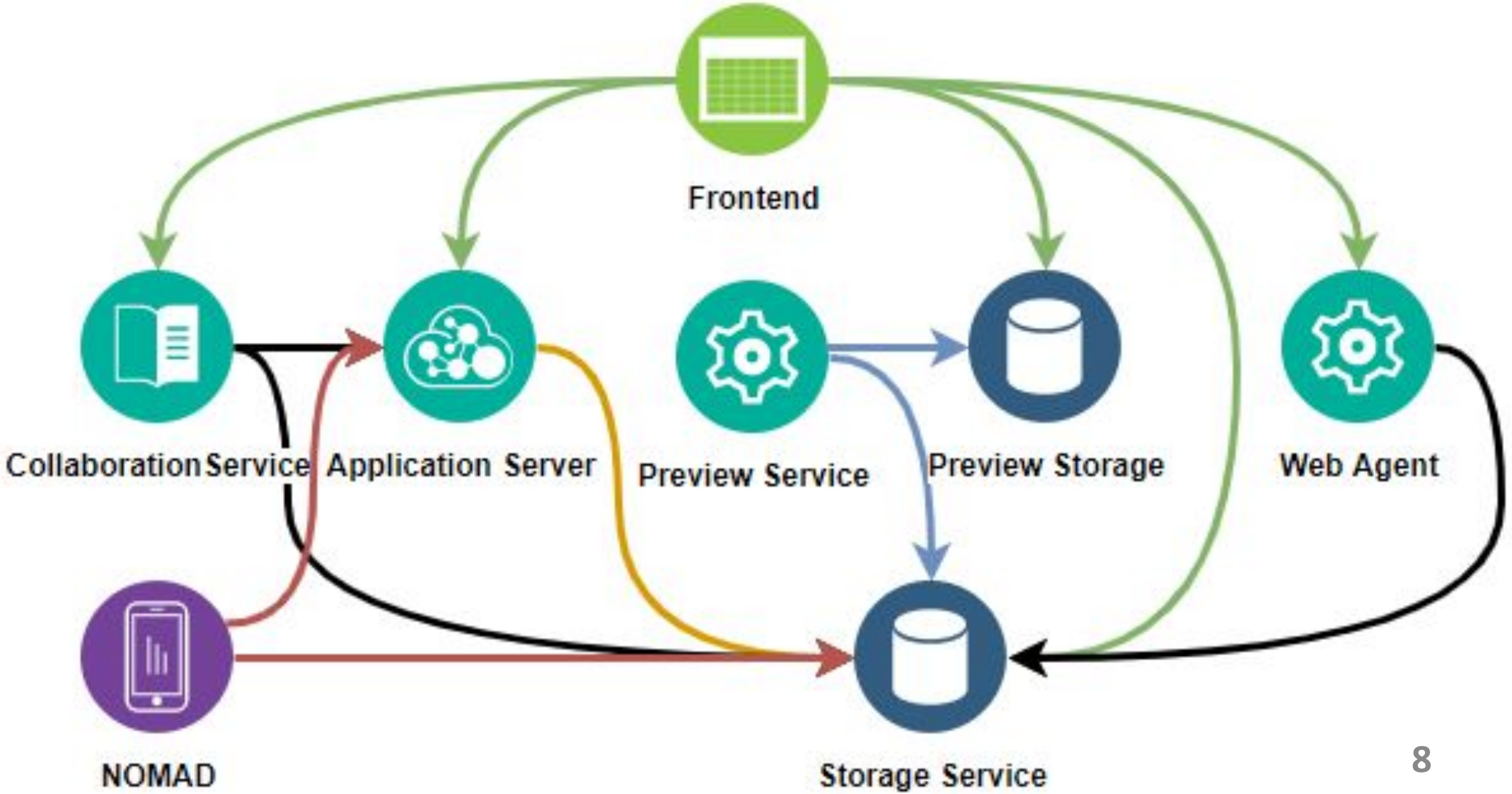
Реализация

- Автоматизировать генерацию кода сервера и клиента
- Версионировать API

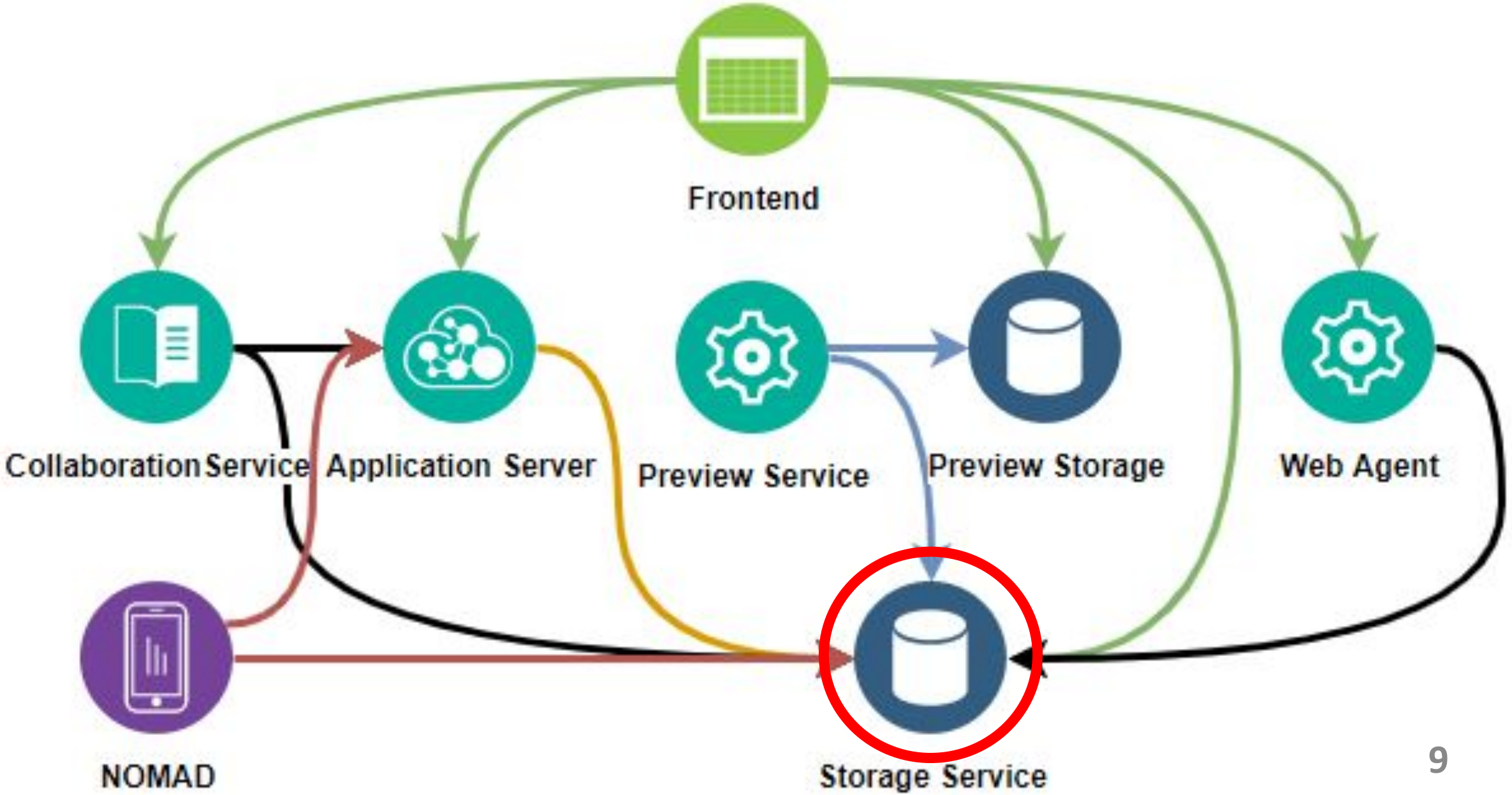
Тестирование

- Тестировать контракты
- Учитывать потребности клиентов

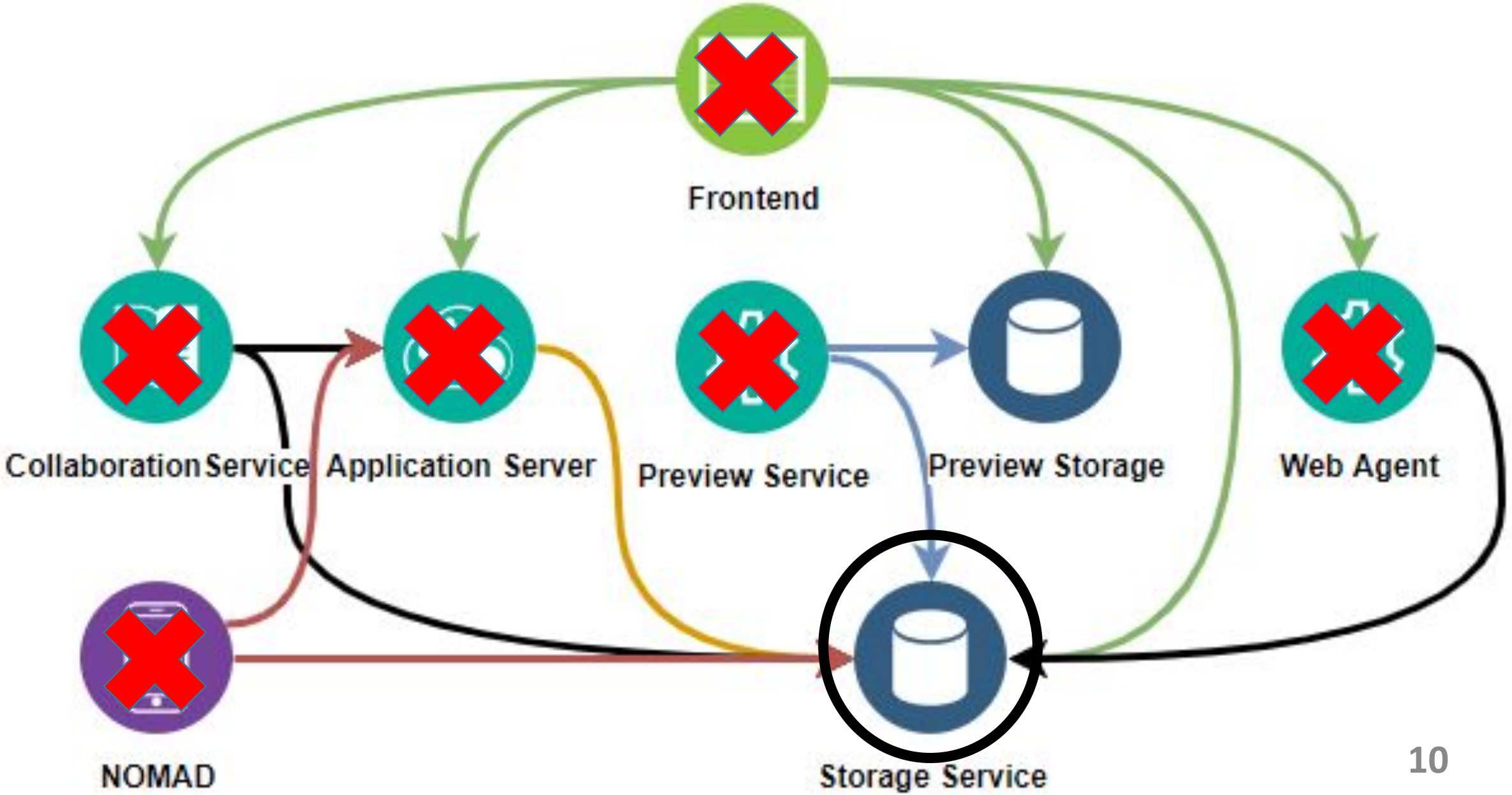
Экосистема Sungero



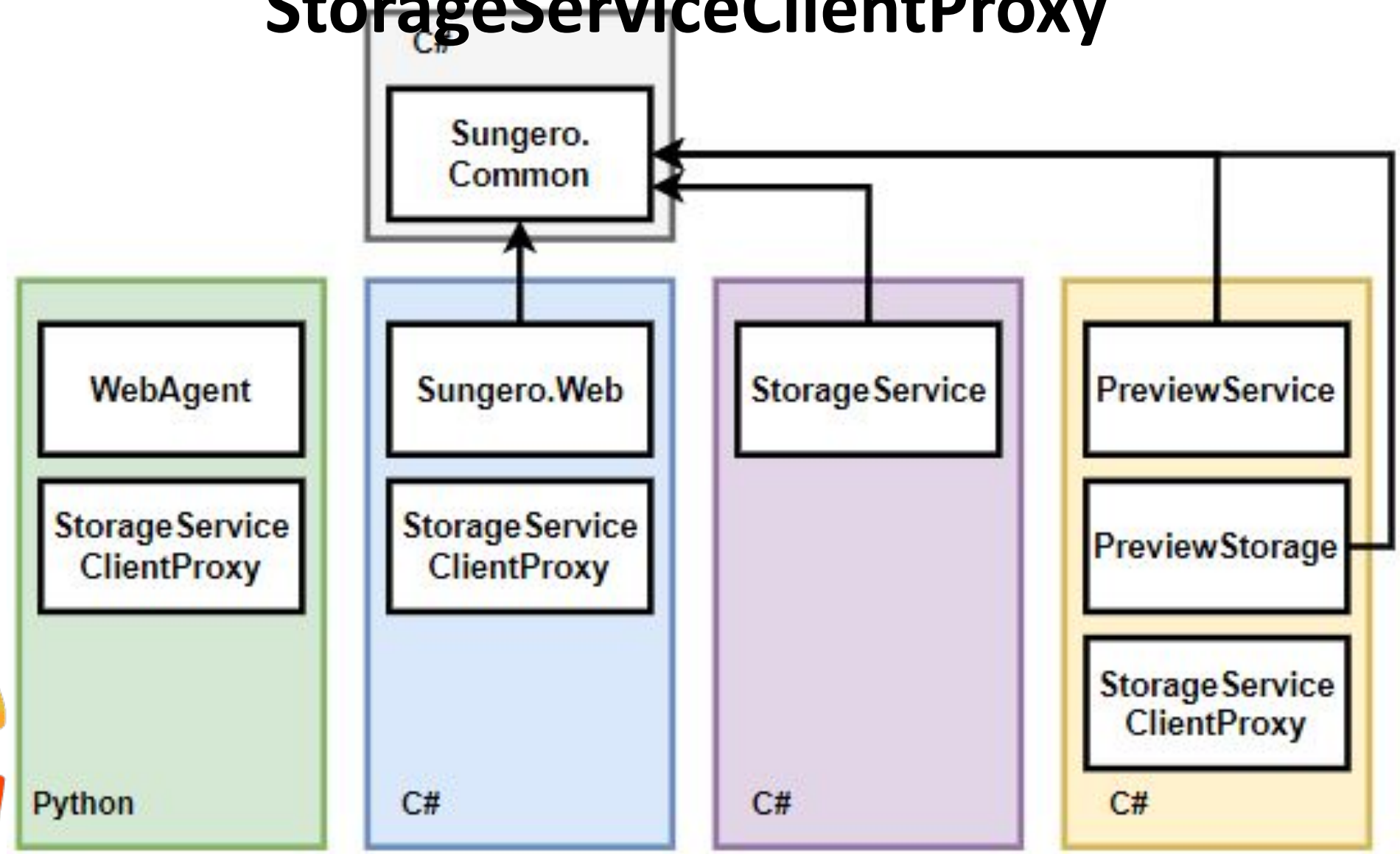
Экосистема Sungero



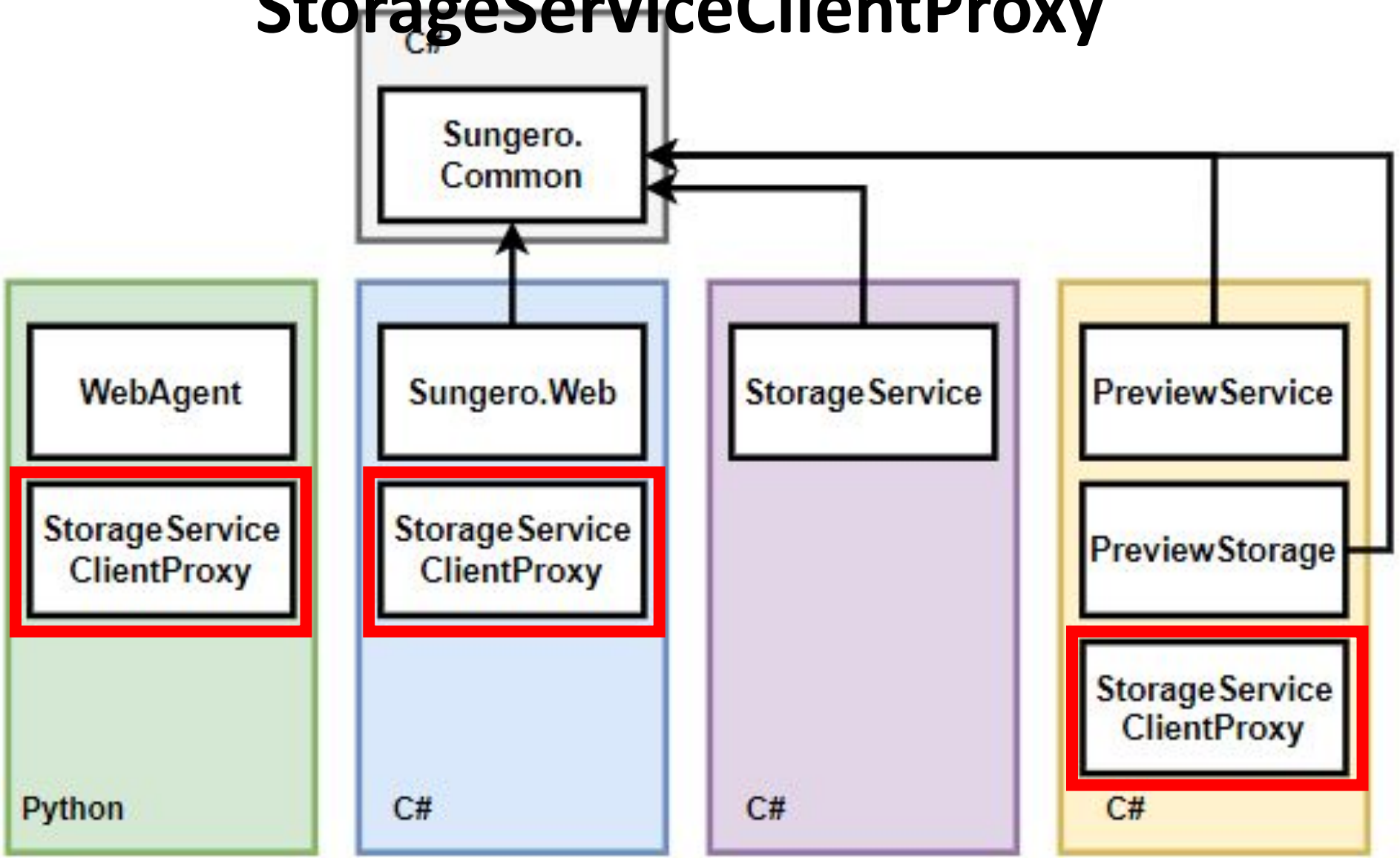
Экосистема Sungero



Проблема. В каждом сервисе своя реализация клиента StorageServiceClientProxy



Проблема. В каждом сервисе своя реализация клиента StorageServiceClientProxy



Постановка задачи

Нужно генерировать код клиента

Генерация должна настраиваться, например, для JWT

Сервисы взаимодействуют по http

Браузер клиент сервиса

Серверная часть – контроллеры уже написаны

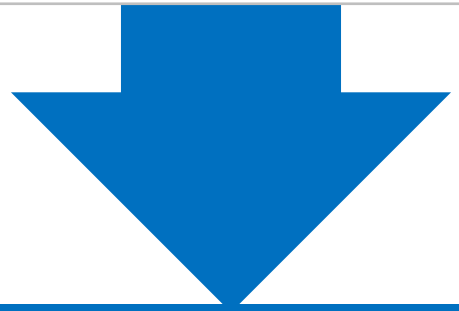
Постановка задачи

- Нужно генерировать код клиента
- Генерация должна настраиваться, например, для JWT
- Сервисы взаимодействуют по http
- Браузер клиент сервиса
- Серверная часть – контроллеры уже написаны

Вход: описание контрактов

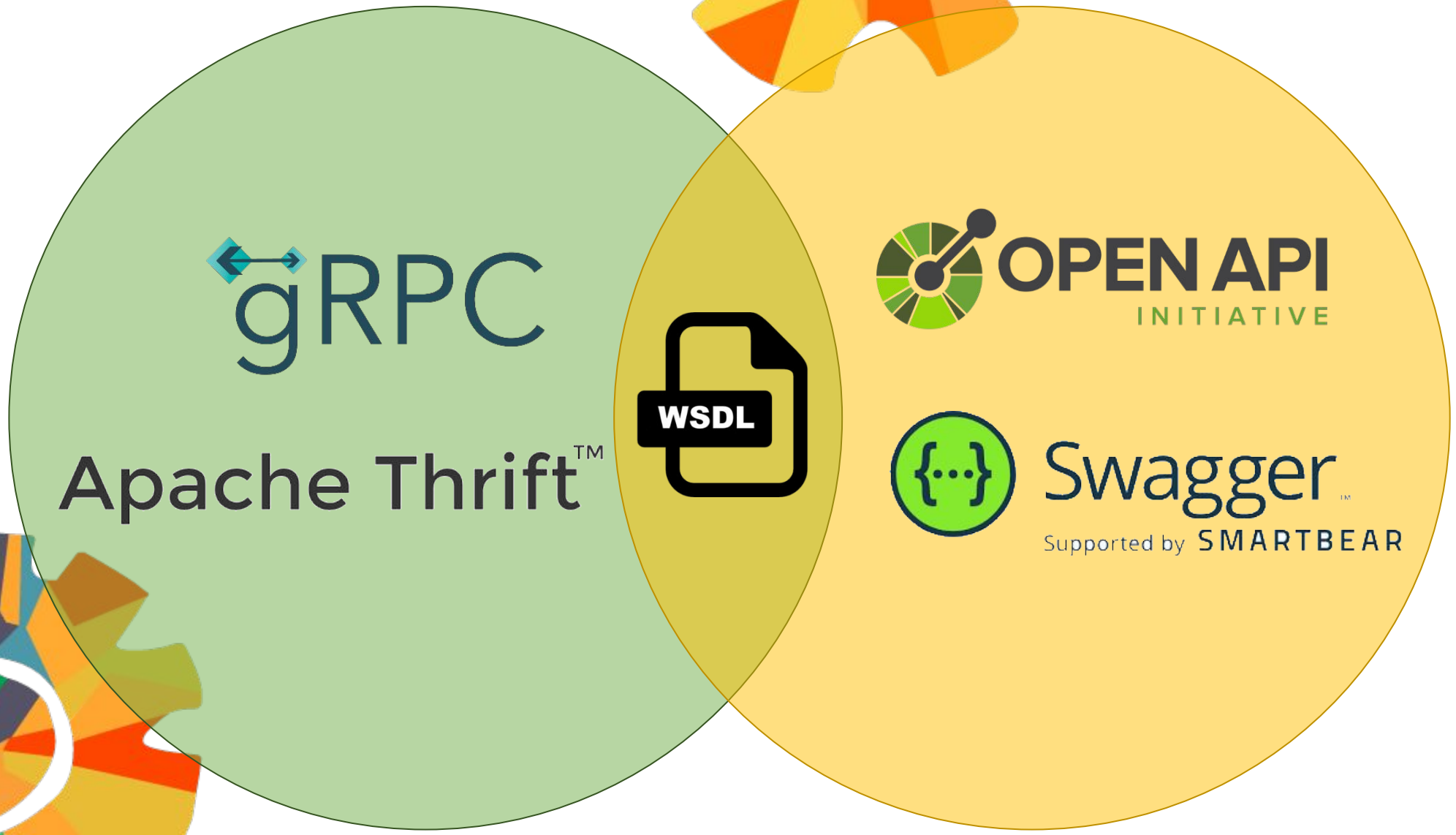
Вопросы:

Какой инструмент выбрать	Как получить контракты	Где расположить контракты	Где расположить код клиента	В какой момент выполнять генерацию



Выход: код для сервера и клиента

Инструменты



RPC

REST

REST vs RPC

RPC – Remote procedure call		REST Representational State Transfer	
Методы и процедуры		Ресурсы, HTTP глаголы и URL	
Создание	http://Restaurant:8080/Orders/PlaceOrder	POST	http://Restaurant:8080/Orders
Получен	http://Restaurant:8080/Orders/GetOrder?OrderNumber=1	GET	http://Restaurant:8080/Orders/1

Свойства	OpenAPI	WSDL	Thrift	gRPC
Тип	REST		RPC	
Платформа	Не зависит			
Язык	Не зависит			
Последовательность разработки*	code first spec first	code first spec first	spec first	spec first code first
Транспортный протокол	HTTP/1.1	любой (REST требует HTTP)	собственный	HTTP/2
Вид	спецификация		фреймворк	
Комментарий	Низкий порог вхождения, много документации	Избыточность XML, тянет за собой SOAP и т. п.	Высокий порог вхождения, мало документации	Средний порог вхождения, лучше документация

Свойства	OpenAPI	WSDL	Thrift	gRPC
Тип	REST		RPC	
Платформа	Не зависит			
Язык	Не зависит			
Последовательность разработки*	code first spec first	code first spec first	spec first	spec first code first
Транспортный протокол	HTTP/1.1	любой (REST требует HTTP)	собственный	HTTP/2
Вид	спецификация		фреймворк	
Комментарий	Низкий порог вхождения, много документации	Избыточность XML, тянет за собой SOAP и т. п.	Высокий порог вхождения, мало документации	Средний порог вхождения, лучше документация

Свойства	OpenAPI	WSDL	Thrift	gRPC
Тип	REST		RPC	
Платформа	Не зависит			
Язык	Не зависит			
Последовательность разработки*	code first spec first	code first spec first	spec first	spec first code first
Транспортный протокол	HTTP/1.1	любой (REST требует HTTP)	собственный	HTTP/2
Вид	спецификация		фреймворк	
Комментарий	Низкий порог вхождения, много документации	Избыточность XML, тянет за собой SOAP и т. п.	Высокий порог вхождения, мало документации	Средний порог вхождения, лучше документация

Свойства	OpenAPI	WSDL	Thrift	gRPC
Тип	REST		RPC	
Платформа	Не зависит			
Язык	Не зависит			
Последовательность разработки*	code first spec first	code first spec first	spec first	spec first code first
Транспортный протокол	HTTP/1.1	любой (REST требует HTTP)	собственный	HTTP/2
Вид	спецификация		фреймворк	
Комментарий	Низкий порог вхождения, много документации	Избыточность XML, тянет за собой SOAP и т.п.	Высокий порог вхождения, мало документации	Средний порог вхождения, лучше документация

Свойства	OpenAPI	WSDL	Thrift	gRPC
Тип	REST		RPC	
Платформа	Не зависит			
Язык	Не зависит			
Последовательность разработки*	code first spec first	code first spec first	spec first	spec first code first
Транспортный протокол	HTTP/1.1	любой (REST требует HTTP)	собственный	HTTP/2
Вид	спецификация		фреймворк	
Комментарий	Низкий порог вхождения, много документации	Избыточность XML, тянет за собой SOAP и т.п.	Высокий порог вхождения, мало документаци	Средний порог вхождения, лучше документация

Свойства	OpenAPI	WSDL	Thrift	gRPC
Тип	REST		RPC	
Платформа	Не зависит			
Язык	Не зависит			
Последовательность разработки*	code first spec first	code first spec first	spec first	spec first code first
Транспортный протокол	HTTP/1.1	любой (REST требует HTTP)	собственный	HTTP/2
Вид	спецификация		фреймворк	
Комментарий	Низкий порог вхождения, много документации	Избыточность XML, тянет за собой SOAP и т. п.	Высокий порог вхождения, мало документации	Средний порог вхождения, лучше документация

Свойства	OpenAPI	WSDL	Thrift	gRPC
Тип	REST		RPC	
Платформа	Не зависит			
Язык	Не зависит			
Последовательность разработки*	code first spec first	code first spec first	spec first	spec first code first
Транспортный протокол	HTTP/1.1	любой (REST требует HTTP)	собственный	HTTP/2
Вид	спецификация		фреймворк	
Комментарий	Низкий порог вхождения, много документации	Избыточность XML, тянет за собой SOAP и т. п.	Высокий порог вхождения, мало документации	Средний порог вхождения, лучше документация

gRPC

OPEN API INITIATIVE

Требует net Core 3.0 и net Standard 2.1

Требует HTTP/2

Нет поддержки в браузере из коробки

Не требует дополнительных обновлений

Обилие инструментов поддерживающих работу со спецификацией

gRPC

OPEN API INITIATIVE

Требует net Core 3.0 и net Standard 2.1

Требует HTTP/2

Нет поддержки в браузере из коробки

Не требует дополнительных обновлений

Обилие инструментов поддерживающих работу со спецификацией

	swashbuckle	NSwag	OpenAPITools
Поддерживаемые версии спецификации	Могут генерировать спецификацию в формате OpenApi v2, v3		
Поддержка code first	Есть	Есть	Нет
Поддерживаемые языки сервера	Нет	C#	Много
Поддерживаемые шаблоны клиентов	Нет	C#, TypeScript, AngularJS, Angular (v2+), window.fetch API	Много
Настройки генерации	Нет	Есть	Есть

	swashbuckle	NSwag	OpenAPITools
Поддерживаемые версии спецификации	Могут генерировать спецификацию в формате OpenApi v2, v3		
Поддержка code first	Есть	Есть	Нет
Поддерживаемые языки сервера	Нет	C#	Много
Поддерживаемые шаблоны клиентов	Нет	C#, TypeScript, AngularJS, Angular (v2+), window.fetch API	Много
Настройки генерации	Нет	Есть	Есть

	swashbuckle	NSwag	OpenAPITools
Поддерживаемые версии спецификации	Могут генерировать спецификацию в формате OpenApi v2, v3		
Поддержка code first	Есть	Есть	Нет
Поддерживаемые языки сервера	Нет	C#	Много
Поддерживаемые шаблоны клиентов	Нет	C#, TypeScript, AngularJS, Angular (v2+), window.fetch API	Много
Настройки генерации	Нет	Есть	Есть

	swashbuckle	NSwag	OpenAPITools
Поддерживаемые версии спецификации	Могут генерировать спецификацию в формате OpenApi v2, v3		
Поддержка code first	Есть	Есть	Нет
Поддерживаемые языки сервера	Нет	C#	Много
Поддерживаемые шаблоны клиентов	Нет	C#, TypeScript, AngularJS, Angular (v2+), window.fetch API	Много
Настройки генерации	Нет	Есть	Есть

	swashbuckle	NSwag	OpenAPITools
Поддерживаемые версии спецификации	Могут генерировать спецификацию в формате OpenApi v2, v3		
Поддержка code first	Есть	Есть	Нет
Поддерживаемые языки сервера	Нет	C#	Много
Поддерживаемые шаблоны клиентов	Нет	C#, TypeScript, AngularJS, Angular (v2+), window.fetch API	Много
Настройки генерации	Нет	Есть	Есть

	swashbuckle	NSwag	OpenAPITools
Поддерживаемые версии спецификации	Могут генерировать спецификацию в формате OpenApi v2, v3		
Поддержка code first	Есть	Есть	Нет
Поддерживаемые языки сервера	Нет	C#	Много
Поддерживаемые шаблоны клиентов	Нет	C#, TypeScript, AngularJS, Angular (v2+), window.fetch API	Много
Настройки генерации	Нет	Есть	Есть

Где расположить контракты? Как осуществить доступ сервисов к контрактам?

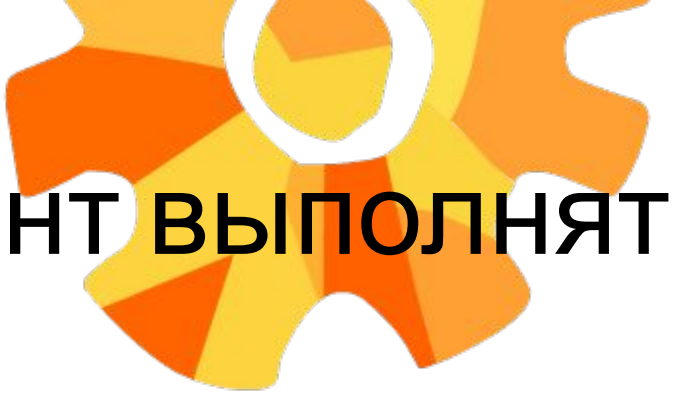
- Папка проекта
- Общая папка
- Через API сервиса (swagger.ui)
- Отдельный репозиторий для спецификаций
- Менеджер пакетов (swaggerhub)

Где расположить контракты? Как осуществить доступ сервисов к контрактам?

- **Папка проекта**
- Общая папка
- Через API сервиса (swagger.ui)
- Отдельный репозиторий для спецификаций
- Менеджер пакетов (swaggerhub)

В какой момент выполнять генерацию?

- Сервис потребитель генерирует сам по необходимости
- После сборки проекта web сервиса поставщика

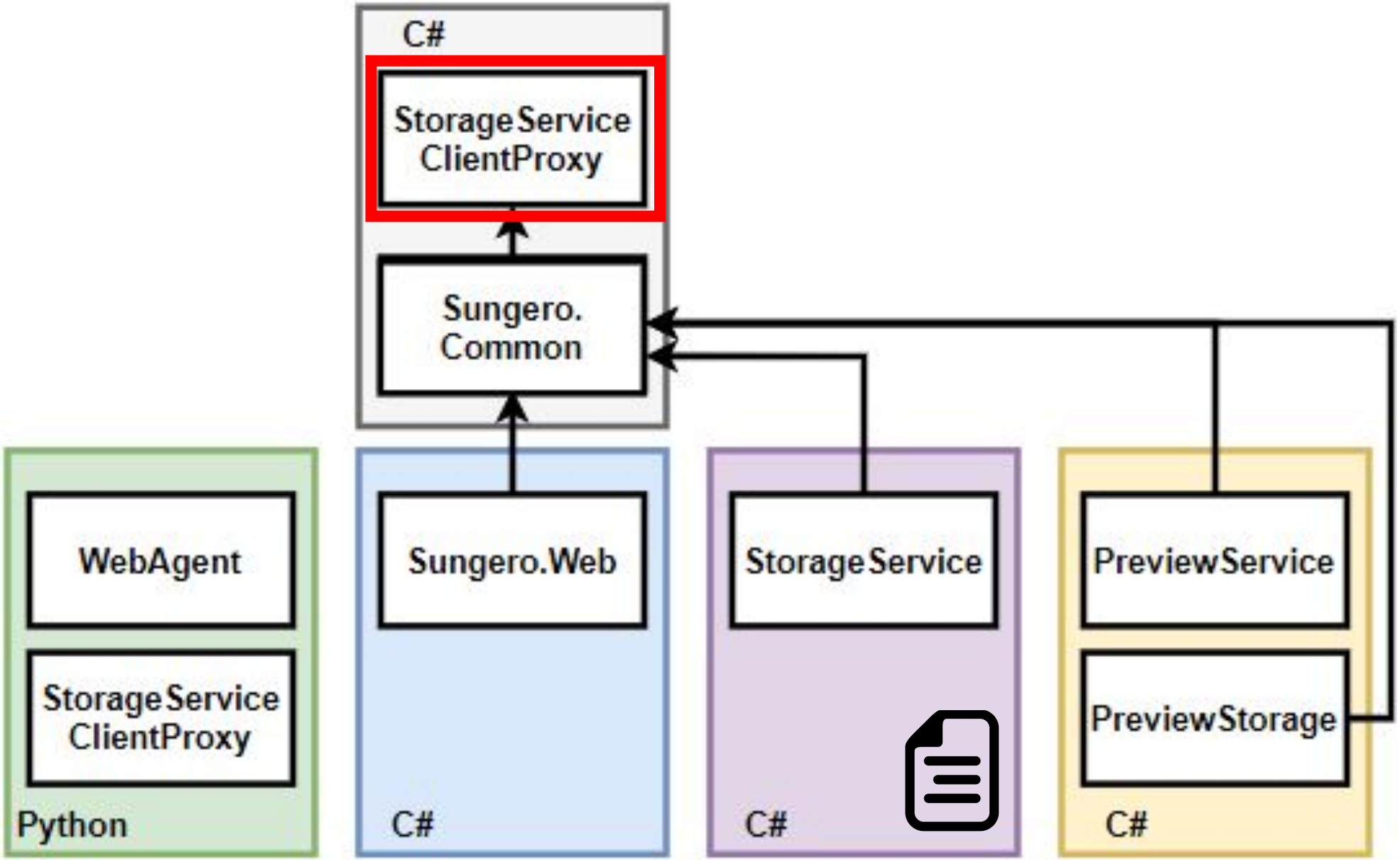


В какой момент выполнять генерацию?

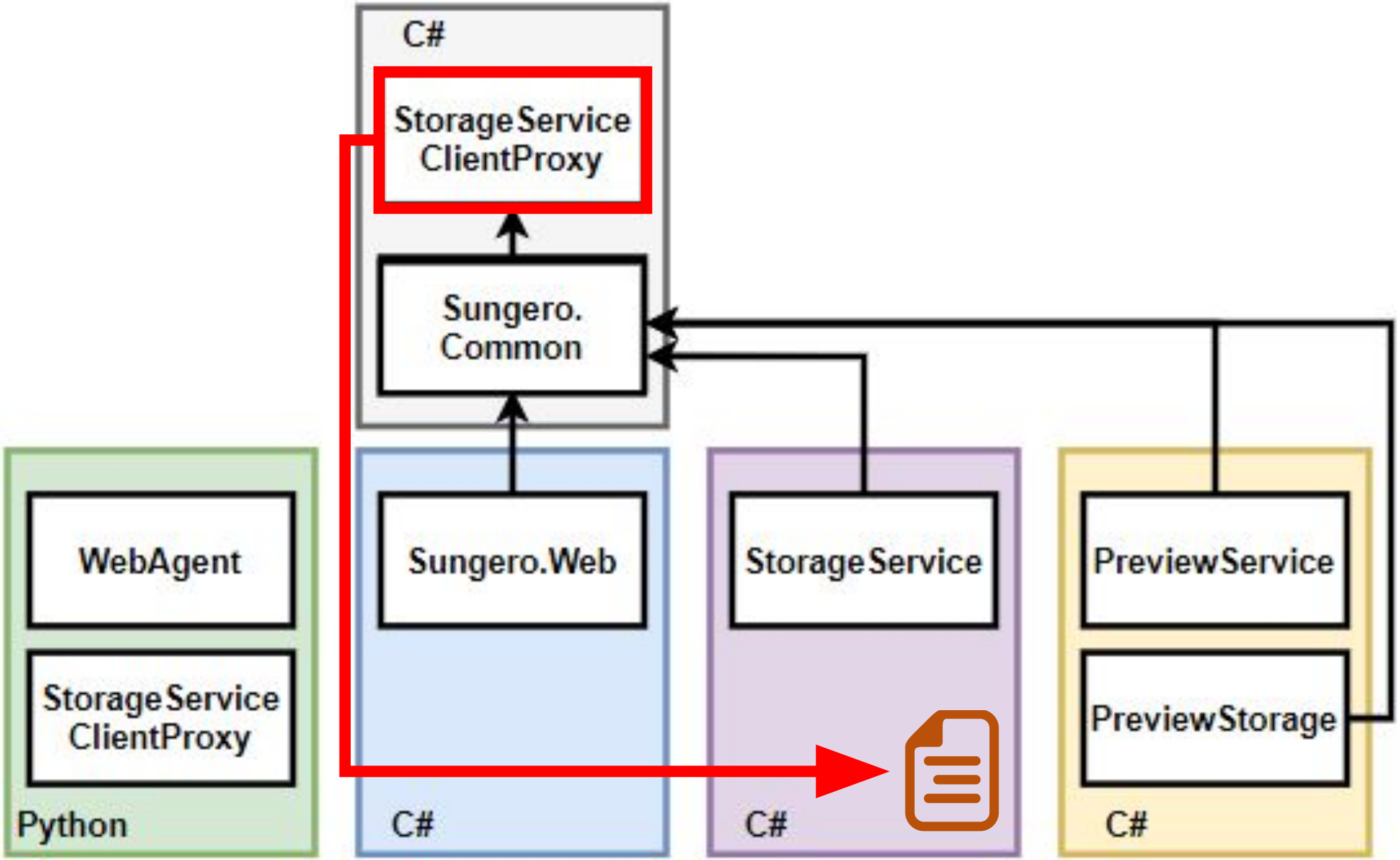
Сервис потребитель генерирует сам по необходимости

После сборки проекта web сервиса поставщика

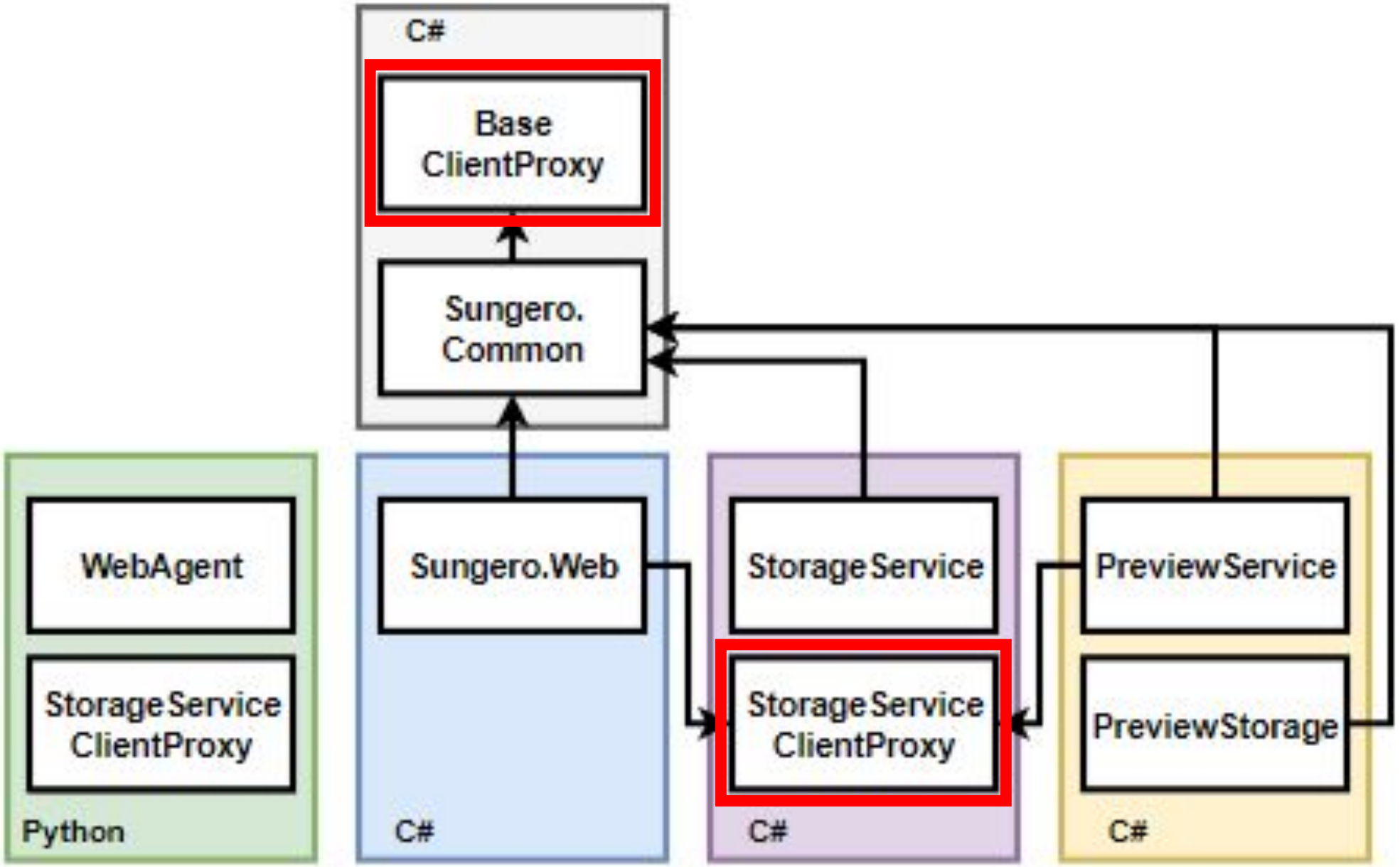
Где расположить код клиента?



Где расположить код клиента?



Где расположили код клиента



Детали реализации

Атрибуты контроллеров

Процессор для генерации спецификации для файловых операций

Конфигурация Nswag для генерации спецификации и кода

Как добавили поддержку для JWT

DescriptionController.cs

```
[Route("[controller]")]
[ApiController]
public class DescriptionController : ControllerBase {
    [OpenApiOperation("GetDescription")]
    [ProducesResponseType(typeof(ConversionDescription), 200)]
    [ProducesResponseType(401)]
    [ProducesResponseType(403)]
    [HttpGet("{pluginName}/{binaryDataId}")]
    public ActionResult<ConversionDescription> GetDescription(
        string pluginName, Guid binaryDataId) {  // код... }
```

FileController.cs

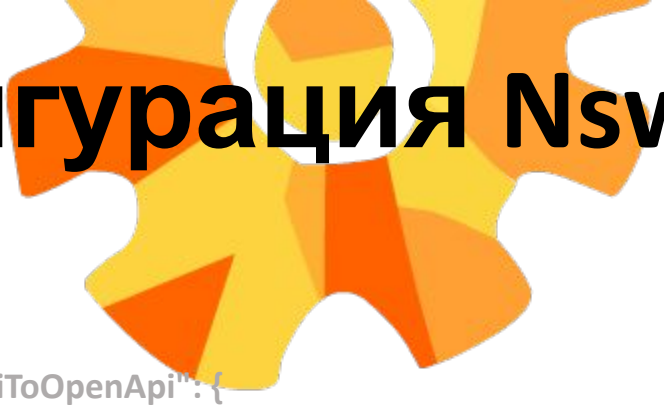
```
[Route("[controller]")]
[ApiController]
public class FileController : ControllerBase {
    [OpenApiOperation("SaveFile")]
    [ProducesResponseType(401)]
    [ProducesResponseType(403)]
    [HttpPost("{pluginName}/{binaryDataId}/{fileName}")]
    [FileUploadOperation]
    public async Task SaveFile() { // код... }
```

Процессор

FileUploadOperationAttribute : OpenApiOperationProcessorAttribute

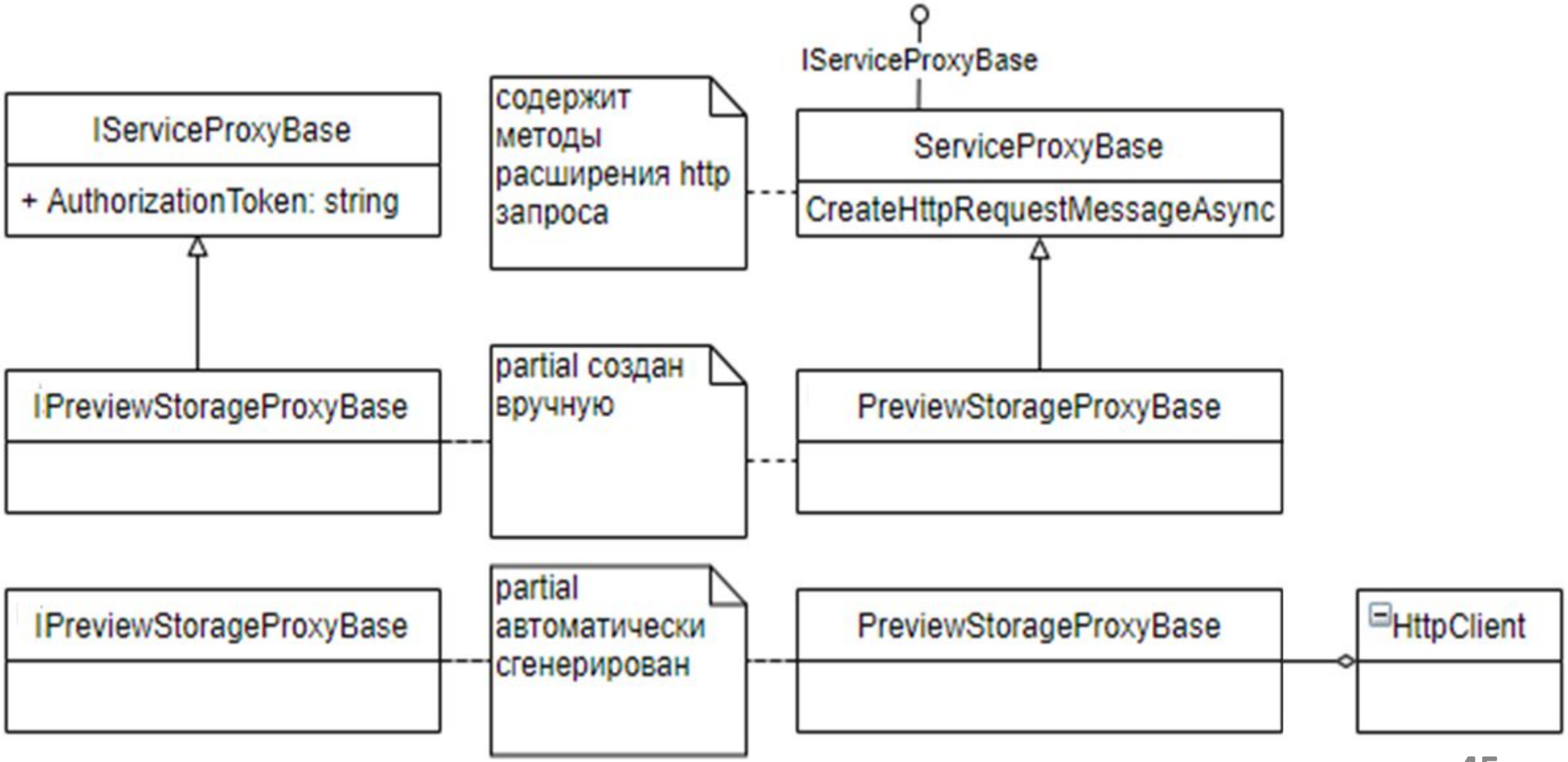
FileUploadOperationProcessor : IOperationProcessor

Конфигурация Nswag.json

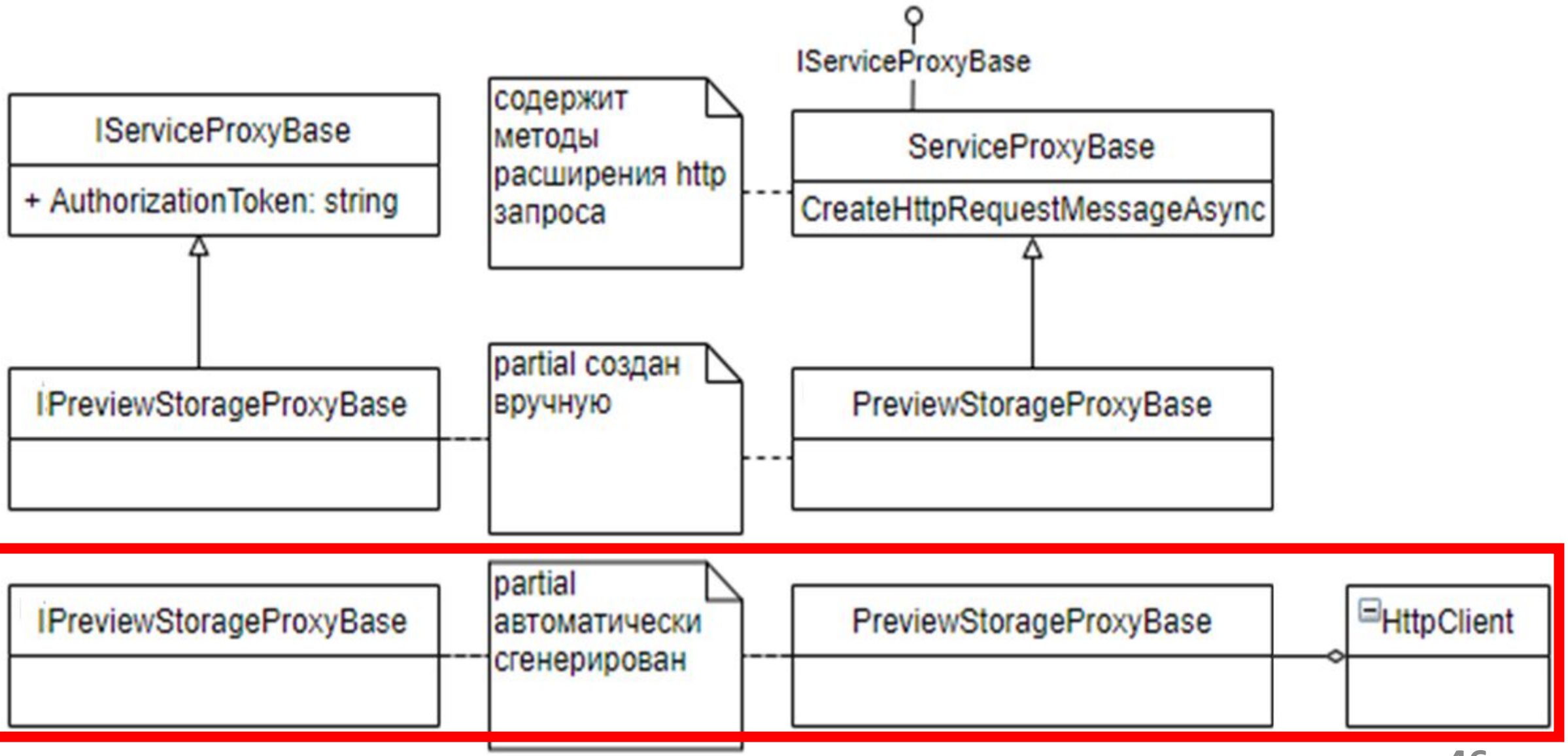


```
"runtime": "NetCore22",
"documentGenerator": {  "webApiToOpenApi": {
        "defaultUrlTemplate": "api/{controller}/{id?}",
        "infoTitle": "PreviewStorage", "infoVersion": "1.0.0",
        "documentName": "v1","allowNullableBodyParameters": true,
                "output": "../../api-docs/PreviewStorage_swagger.json",
                "outputType": "OpenApi3",
                "assemblyPaths": [
                "../../bin/$(Configuration)/PreviewStorage/netcoreapp2.2/PreviewStorage.dll“ ], }  },
"codeGenerators": { "openApiToCSharpClient": { "input": "../../api-docs/PreviewStorage_swagger.json",
        "namespace": "PreviewStorage.WebApiProxy",
        "generateClientInterfaces": true,
        "useHttpRequestMessageCreationMethod": true,
        “httpClientType": "System.Net.Http.HttpClient",
        "additionalNamespaceUsages": [ "PreviewService.Common.Model“ ],
        "output": "../PreviewStorage.WebApiProxy/PreviewStorageProxy.g.cs",
        "className": "{controller}PreviewStorageProxy",
        "operationGenerationMode": "SingleClientFromOperationId“ } } }
```

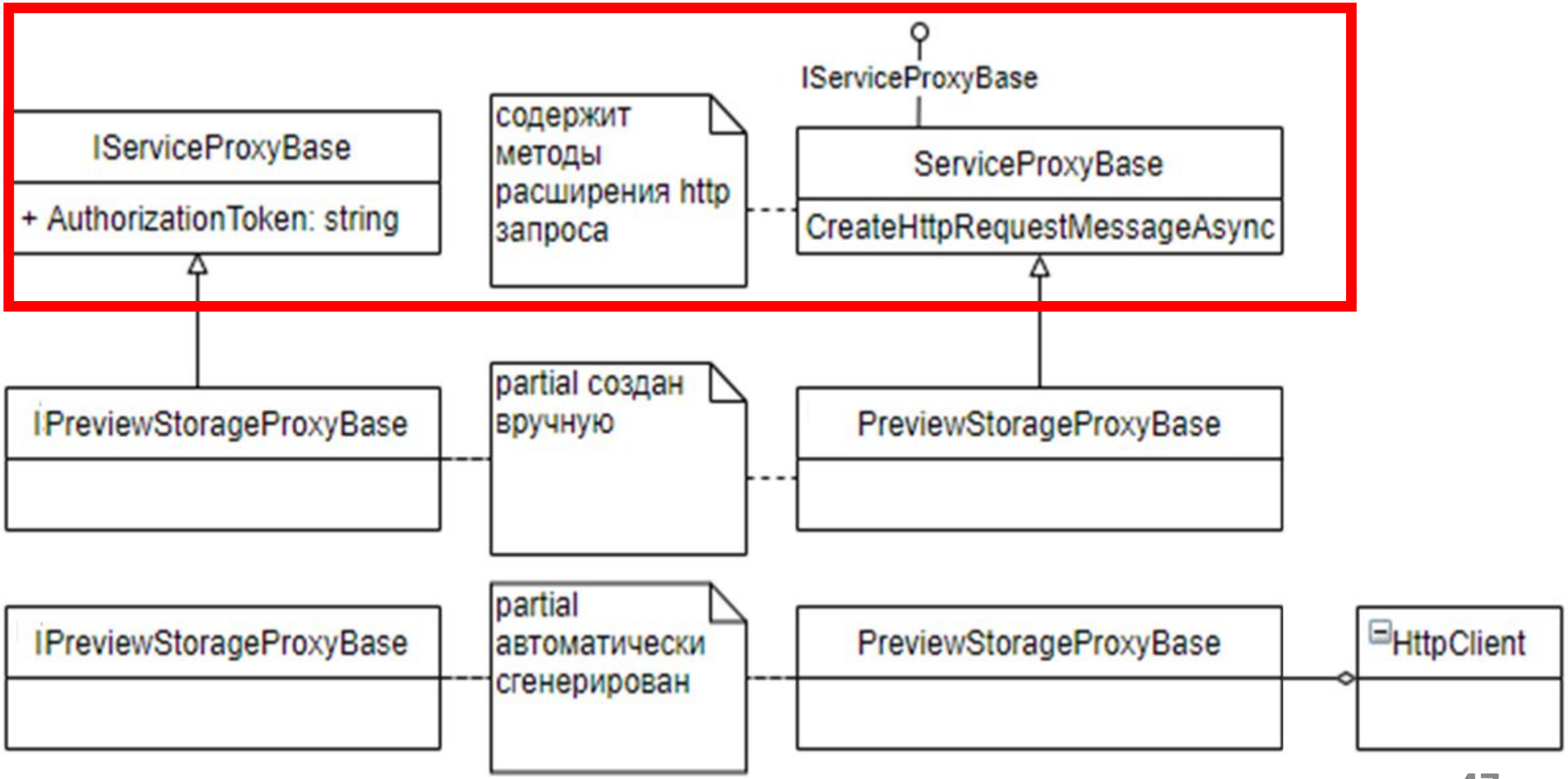
Как добавили поддержку JWT



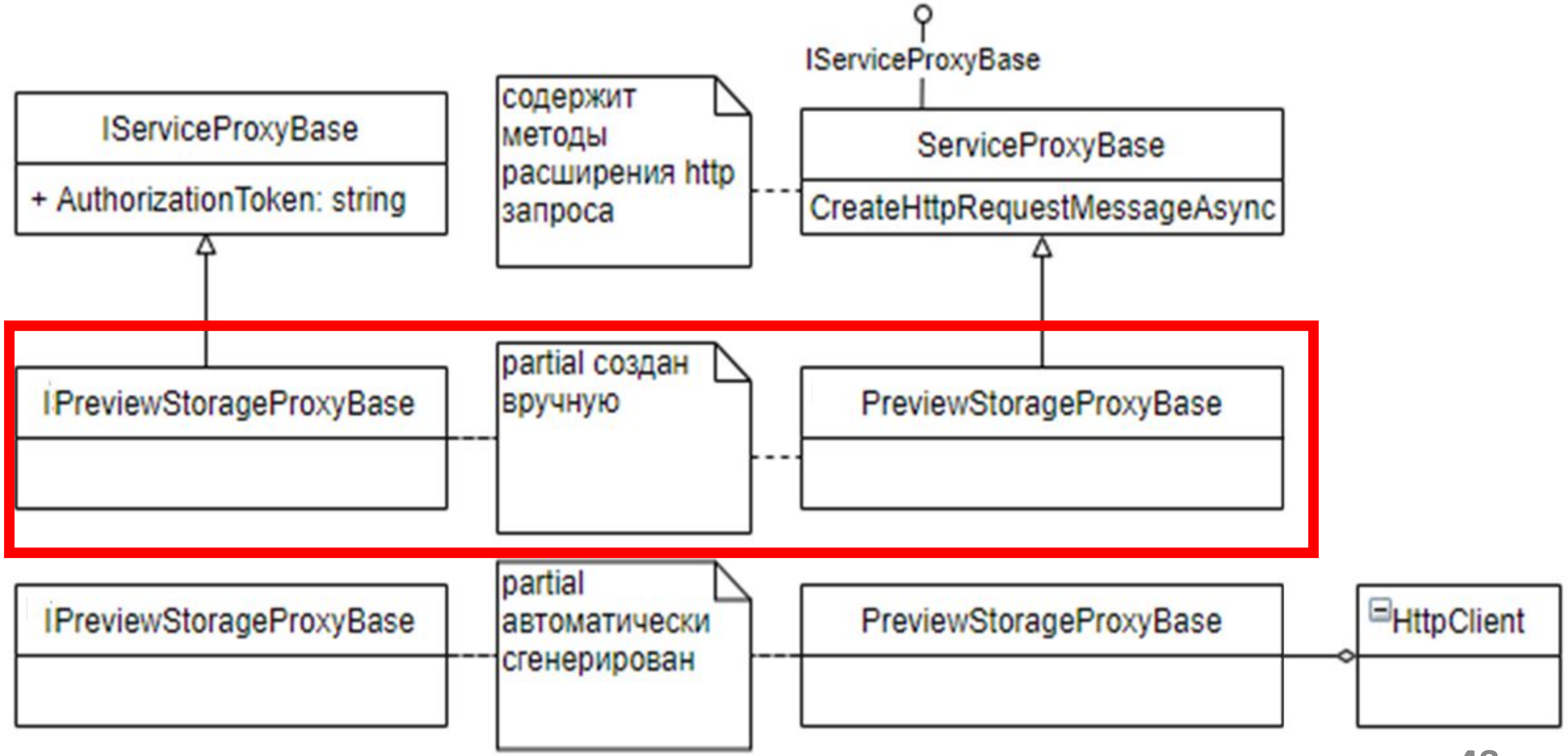
Как добавили поддержку JWT



Как добавили поддержку JWT



Как добавили поддержку JWT



Вывод

OPEN API INITIATIVE

Спецификация многословна

Генерация прикручивается быстро за счет хорошей документации к спецификации и инструментам её реализующим

Пришлось докрутить процессор для генерации спецификации

Много атрибутов на контроллерах – отвлекает

Вывод

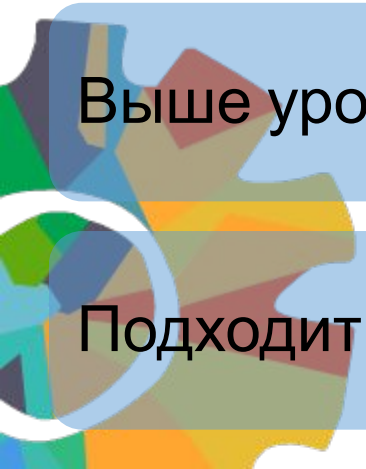
gRPC

Спецификация кажется более человекочитаемой по сравнению с OpenAPI

Использовать пока не имеет смысла без обновления фреймворка и всех зависимых проектов

Нет поддержки браузером из коробки

Выше уровень абстракции, не нужно явно работать с URL, HTTP и т.п.



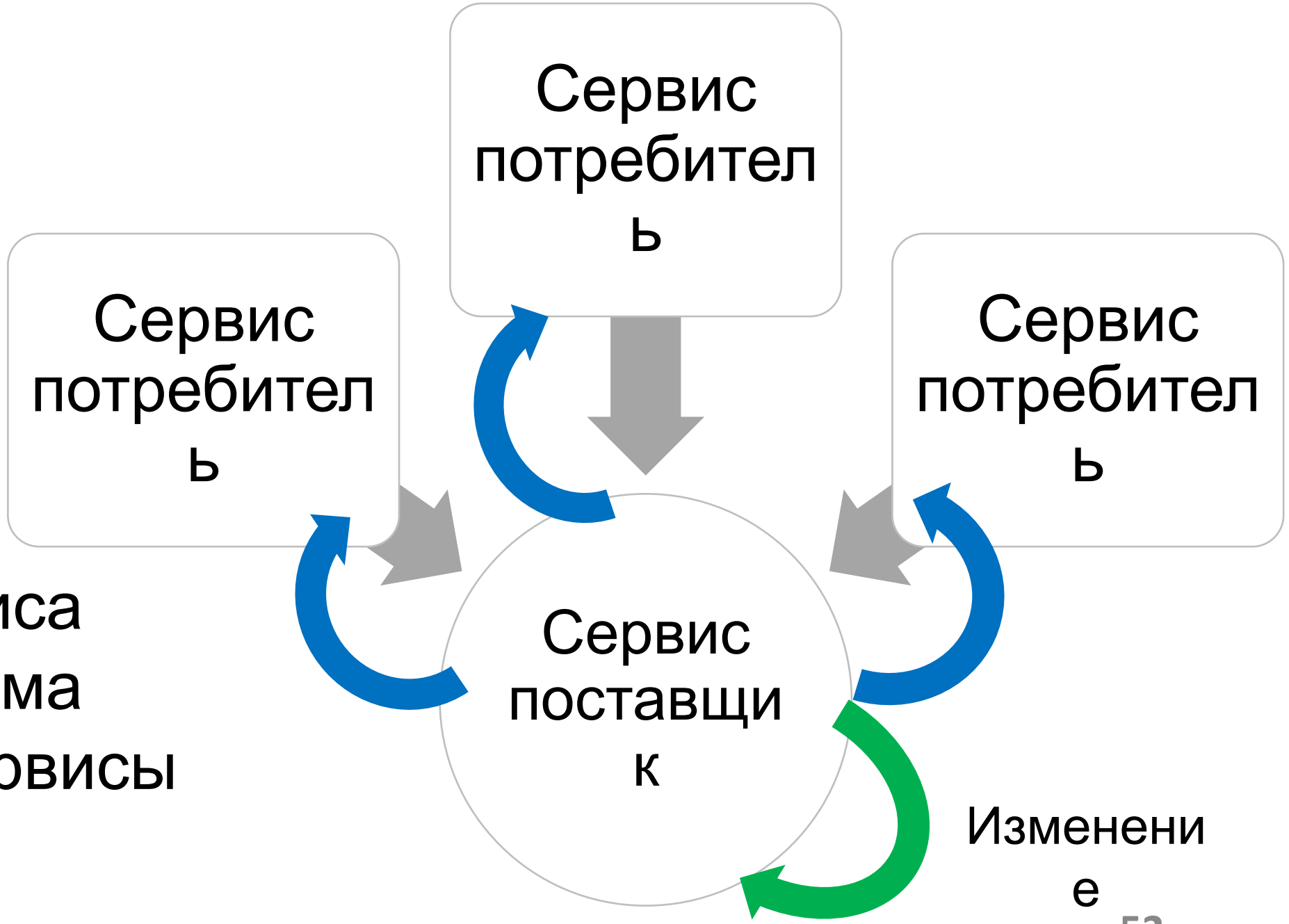
Подходит для общения микросервисов по HTTP/2

Версионирование

После изменений в сервисах поставщиках вся система должна оставаться в согласованном, рабочем состоянии

Нужно избежать breaking changes в API, чтобы не поломать клиентов

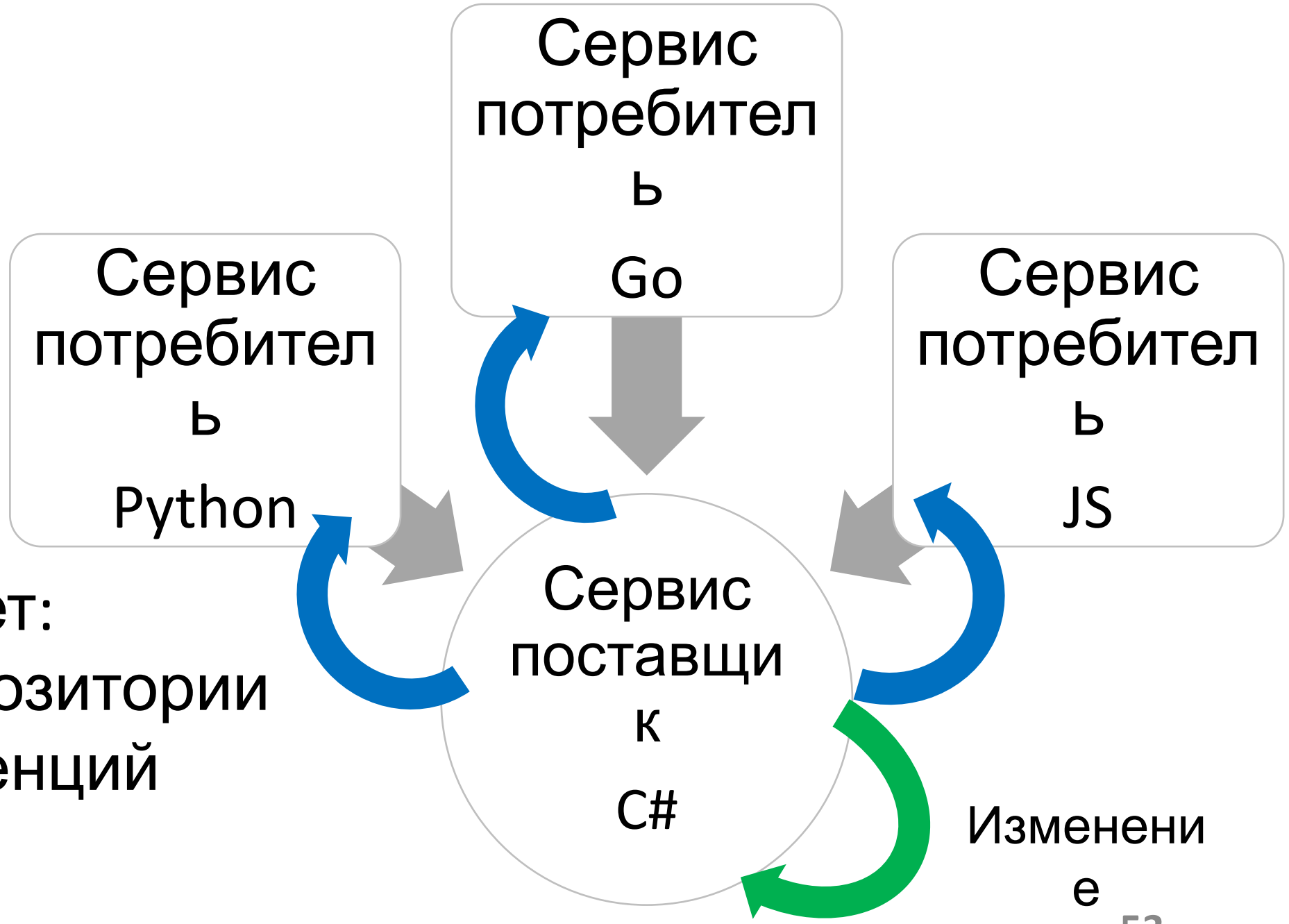
Варианты решения



Команда сервиса поставщика сама исправляет сервисы потребители.

Без

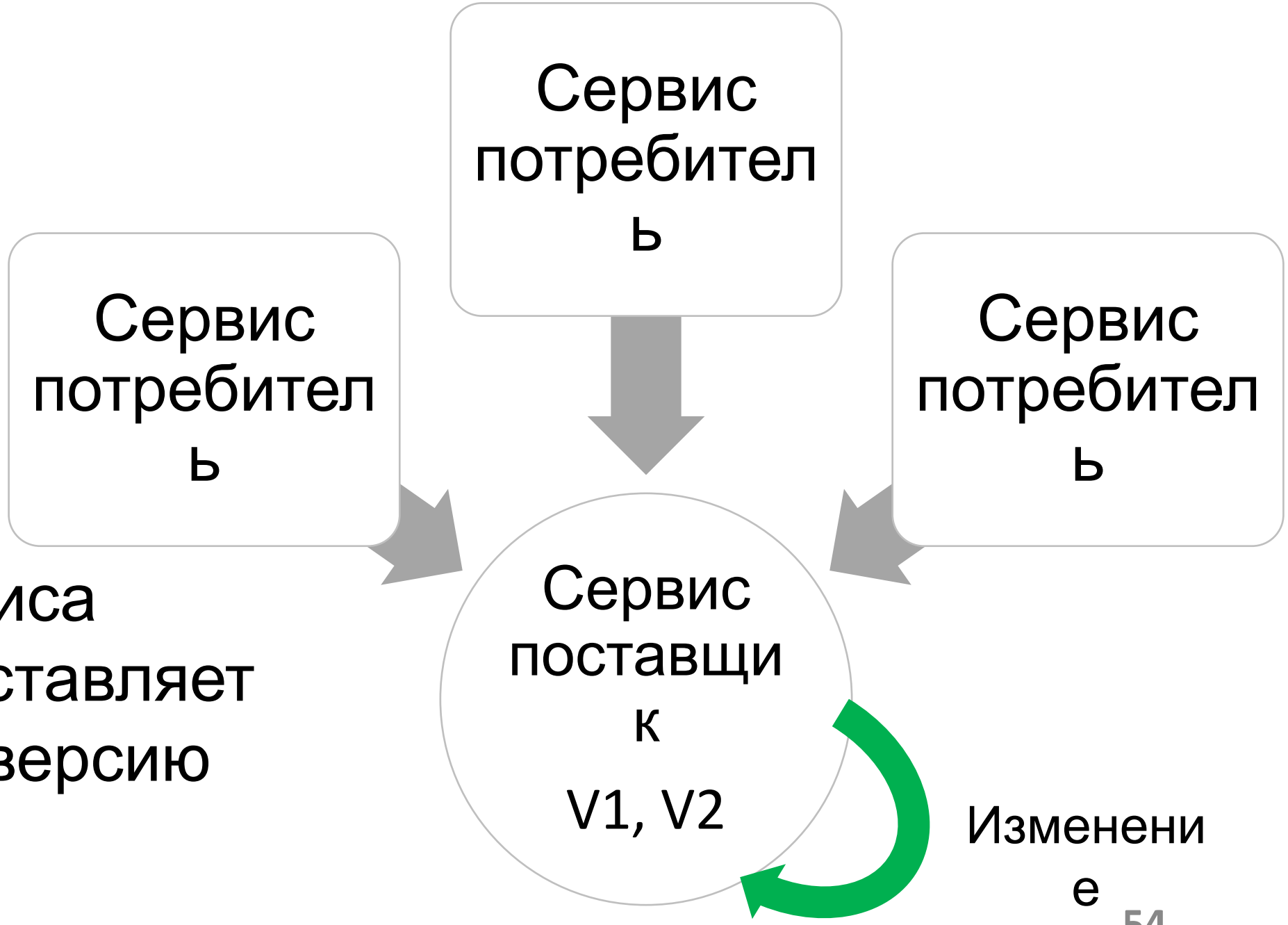
Варианты решения



Плохо работает:

- Разные репозитории
- Нет компетенций

Варианты решения



Команда сервиса поставщика оставляет предыдущую версию контрактов

Версионирование

	gRPC	OpenAPI
Атрибут версии	На уровне protobuf есть атрибут package [packageName].[Version]	На уровне спецификации есть атрибуты basePath (для URL) и Version
Атрибут Deprecated для методов	Есть, но не учитывается генератором кода под C#	Есть, помечается как Obsolete В Nswag не поддерживается при code first, нужно писать свой процессор
Атрибут Deprecated для параметров	Есть, помечается как Obsolete	Есть, помечается как Obsolete В Nswag не поддерживается при code first, нужно писать свой процессор

Версионирование

	gRPC	OpenAPI
Атрибут версии	На уровне protobuf есть атрибут package [packageName].[Version]	На уровне спецификации есть атрибуты basePath (для URL) и Version
Атрибут Deprecated для методов	Есть, но не учитывается генератором кода под C#	Есть, помечается как Obsolete В Nswag не поддерживается при code first, нужно писать свой процессор
Атрибут Deprecated для параметров	Есть, помечается как Obsolete	Есть, помечается как Obsolete В Nswag не поддерживается при code first, нужно писать свой процессор

Версионирование

	gRPC	OpenAPI
Атрибут версии	На уровне protobuf есть атрибут package [packageName].[Version]	На уровне спецификации есть атрибуты basePath (для URL) и Version
Атрибут Deprecated для методов	Есть, но не учитывается генератором кода под C#	Есть, помечается как Obsolete В Nswag не поддерживается при code first, нужно писать свой процессор
Атрибут Deprecated для параметро	Есть, помечается как Obsolete	Есть, помечается как Obsolete В Nswag не поддерживается при code first, нужно писать свой процессор

Версионирование

	gRPC	OpenAPI
Атрибут версии	На уровне protobuf есть атрибут package [packageName].[Version]	На уровне спецификации есть атрибуты basePath (для URL) и Version
Атрибут Deprecated для методов	Есть, но не учитывается генератором кода под C#	Есть, помечается как Obsolete В Nswag не поддерживается при code first, нужно писать свой процессор
Атрибут Deprecated для параметров	Есть, помечается как Obsolete	Есть, помечается как Obsolete В Nswag не поддерживается при code first, нужно писать свой процессор

Когда вводить новую версию

Для OpenAPI есть инструмент Azure opeanapi-diff проверки совместимости между 2 спецификациями

Для gRPC автоматического инструмента не обнаружил, есть только политики версионировния

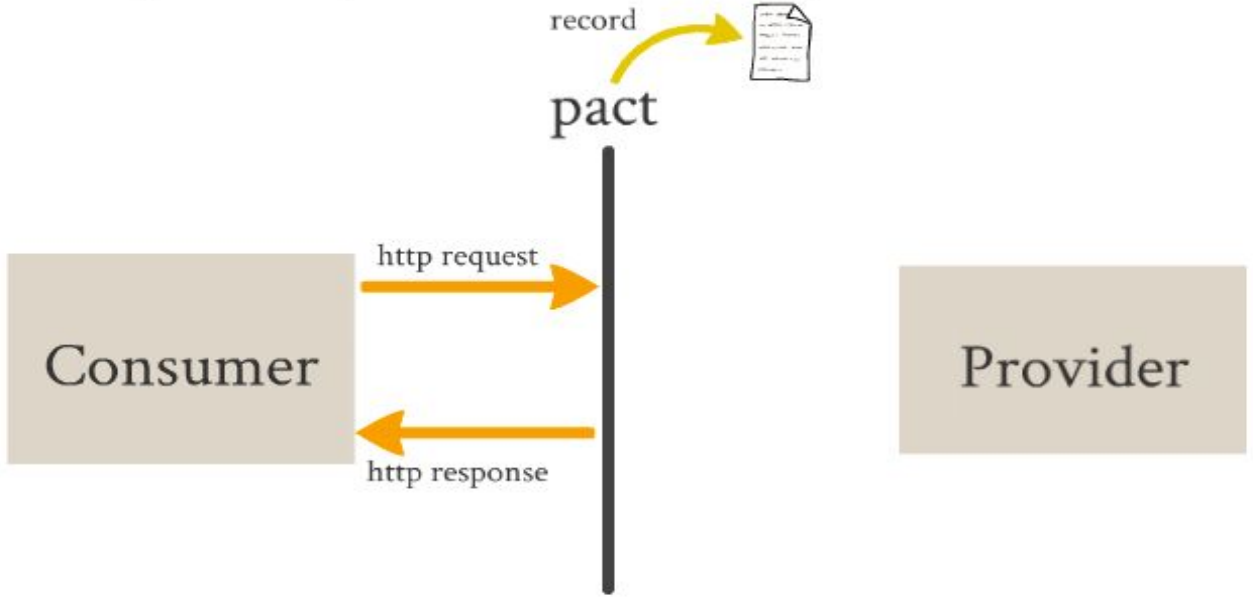
Тестирование контрактов

- Consumer driven contracts (CDC)
- Pact
- Как CDC можно встроить в CI

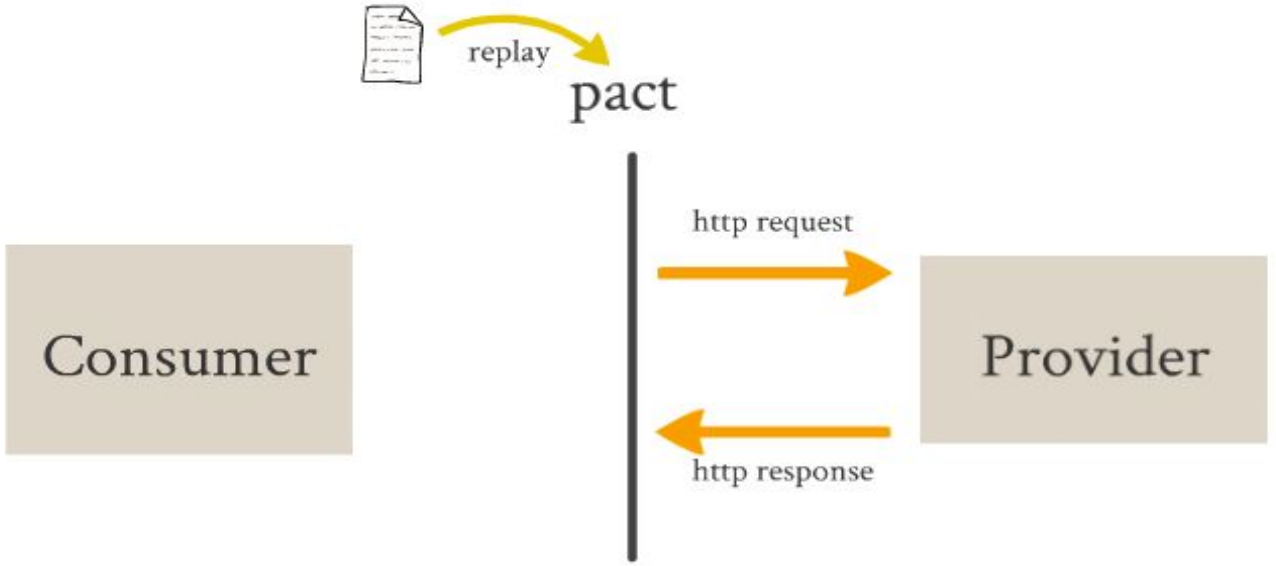
Consumer driven contracts

PACTS

Step 1 - Define Consumer expectations



Step 2 - Verify expectations on Provider



PACTS

Pact + Pact broker - самостоятельно

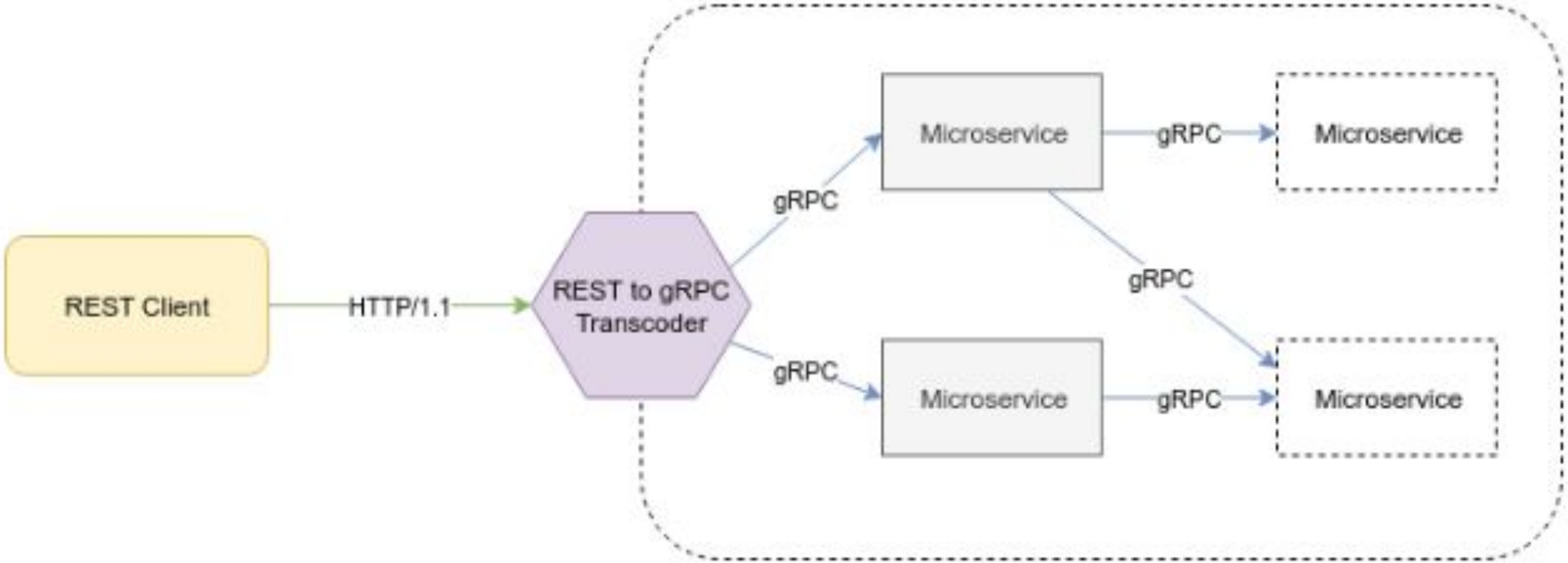
Pact Flow SaaS – приобрести

Спасибо за внимание!

Разработчик DIRECTUM
в г. Уфа **Романюк Антон**

romanyuk_aa@directum.ru

REST + gRPC

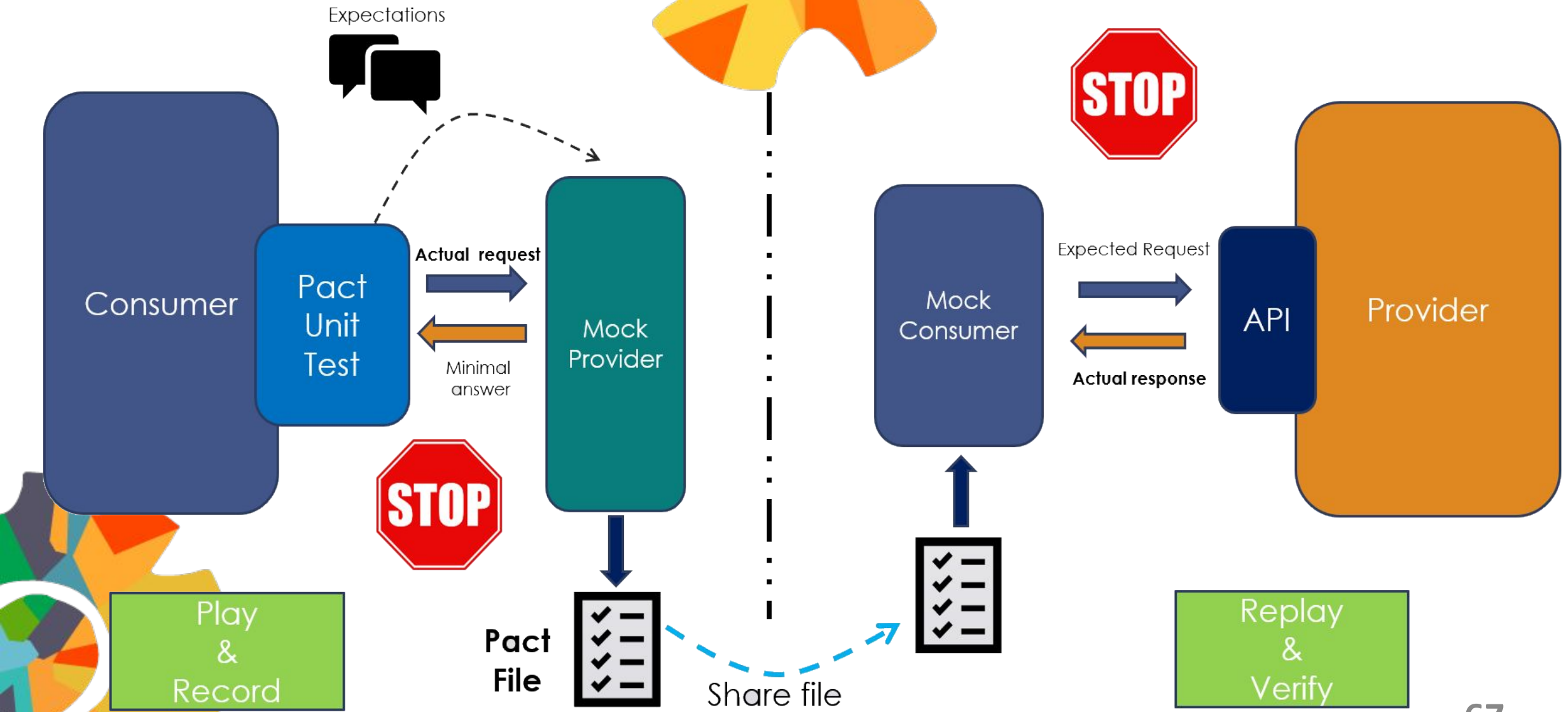

REST Client
HTTP/1.1
REST to gRPC Transcoder
gRPC
gRPC
Microservice
gRPC
Microservice
gRPC
Microservice
gRPC
Microservice

Настройка csproj

- Создали WebApi проект для клиента
- Написали конфиг для Nswag CLI – Nswag.json
- Написали PostBuild Target внутри csproj проекта сервиса поставщика

```
<Target Name="GenerateWebApiProxyClient“ AfterTargets="PostBuildEvent">
<Exec Command="$(NSwagExe_Core22) run nswag.json
/variables:Configuration=$(Configuration)" />
</Target>
```

Как CDC можно встроить в CI



Процессор

```
public class FileUploadOperationAttribute : OpenApiOperationProcessorAttribute {
public FileUploadOperationAttribute() : base(typeof(FileUploadOperationProcessor)) { } }

public class FileUploadOperationProcessor : IOperationProcessor {
public bool Process(OperationProcessorContext context) {
  var parameters = context.OperationDescription.Operation.Parameters;
  parameters.Add(new OpenApiParameter() {
      Name = "stream", Kind = OpenApiParameterKind.Body,
      Schema = new JsonSchema { Type = JsonObjectType.String, Format = "binary" },
      IsRequired = true, Description = "Файл.",
  });
  return true; } }
```

WSDL

- Language, platform, and transport independent (REST requires use of HTTP)
- Works well in distributed enterprise environments (REST assumes direct point-to-point communication)
- Standardized
- Provides significant pre-build extensibility in the form of the WS* standards
- Built-in error handling
- Automation when used with certain language products

Версионирование

Использовать номер версии в URL сервиса
http://service:8080/api/v1/action

- Как часто меняются контракты
- Как выглядит идентификатор версии в URL
- Как долго поддерживать прошлые версии

The following table offers a high-level comparison of features between gRPC and HTTP APIs with JSON.

Feature	gRPC	HTTP APIs with JSON
Contract	Required (*.proto*)	Optional (OpenAPI)
Protocol	HTTP/2	HTTP
Payload	Protobuf (small, binary)	JSON (large, human readable)
Prescriptiveness	Strict specification	Loose. Any HTTP is valid.
Streaming	Client, server, bi-directional	Client, server
Browser support	No (requires grpc-web)	Yes
Security	Transport (TLS)	Transport (TLS)
Client code-generation	Yes	OpenAPI + third-party tooling

- https://servicesblog.redhat.com/2019/01/31/comparing-openapi-with-grpc/
- https://docs.microsoft.com/en-us/aspnet/core/grpc/comparison?view=aspnetcore-3.1
- https://blog.maddevs.io/introduction-to-grpc-6de0d9c0fe61
- https://github.com/Azure/openapi-diff
- https://www.davidkaya.com/with-openapi-against-breaking-changes/
- https://github.com/grpc/grpc-web/issues/517 не подойдет для передачи больших файлов
- https://developers.google.com/protocol-buffers/docs/techniques?hl=en#large-data сообщения не больше 1 мб
- https://app.swaggerhub.com/help/integrations/api-auto-mocking

	swashbuckle	NSwag	OpenAPITools	Свой инструмент
Поддерживаемые версии спецификации	Могут генерировать спецификацию в формате OpenApi v2, v3			Ad hoc решение, зато умеет генерировать js для SignalR
Поддержка code first	Есть	Есть	Нет	Да\Нет
Поддерживаемые языки точек доступа	Нет	C#	Много	Нет
Поддерживаемые шаблоны клиентов	Нет	C#, TypeScript, AngularJS, Angular (v2+), window.fetch API	Много	TypeScript
Настройки	Нет	Есть	Есть	Нет

PACT

- PACT и OpenAPI (REST)
- PACT и gRPC
- PACT и Message Queues