



Версионирование и кодогенерация в **REST API**



Кирилл Лихтарович

Experienced developer

Telegram: @likhtarovich

Veeam в числах

Офисы R&D:



82%

Клиентов из Fortune 500



35+

Стран присутствия



400,000+

Клиентов в мире,



4400+

Сотрудников



В 3.5 раза

Выше удовлетворенность
клиентов, чем в среднем
по индустрии



\$1+ млрд.

Выручка в 2019

Продукты Сотрудники



REST API в Veeam



Veeam
Backup & Replication



Veeam Service
Provider Console



Veeam Availability
Orchestrator



Veeam Backup
Enterprise Manager



Veeam ONE



Veeam Backup
for Microsoft Office 365

Структура доклада

1. Версионирование
 1. Обратная совместимость
 2. Методы версионирования для REST API
 3. Два способа проектирования REST API
2. Кодогенерация
 1. Схема кодогенератора
 2. С чего начать собственную реализацию

Версия ПО

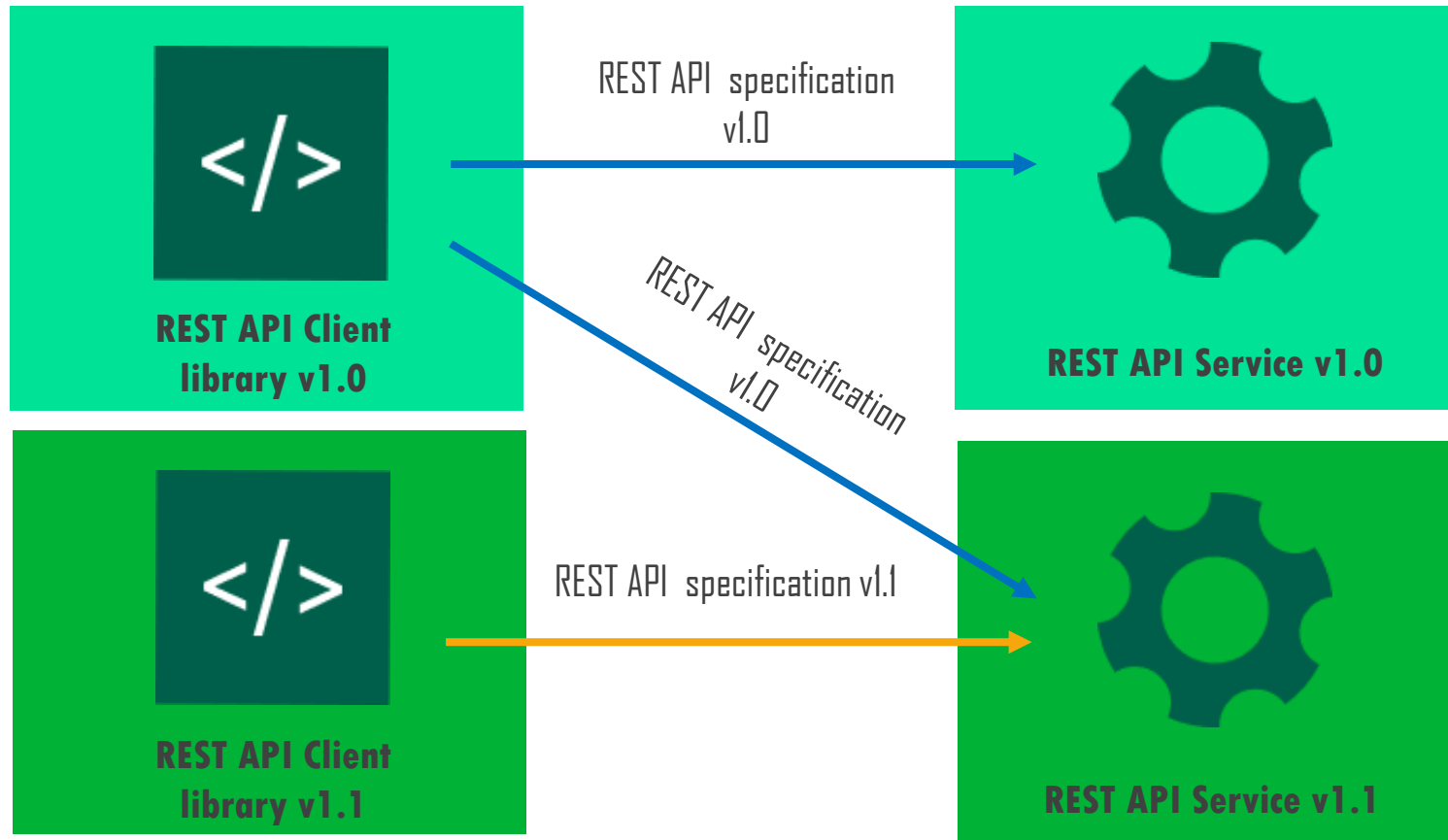
ProductVersion Major.Minor.Build

MyService 3.1.0-rev-8

Windows 10.0.18362.476

Обратная совместимость

6



Обратная совместимость

7



Виды изменений

Ломающие:

нарушающие предыдущий контракт

Неломающие:

дополняющие предыдущий контракт

V2 to v3 reference

This reference provides all resource differences between v2 and v3. For high-level differences between the two versions, see [Drive API v3 versus v2](#).

Method differences between v2 and v3

This table shows v2 methods that have changed in v3.

v2 Methods	v3 Methods
<code>apps.get</code>	n/a
<code>apps.list</code>	n/a
<code>children.delete</code>	<code>files.update</code> on the child with <code>?removeParents=parent_id</code>
<code>children.get</code>	<code>files.list</code> with <code>?q='parent_id'+in+parents</code>
<code>children.insert</code>	<code>files.update</code> on the child with <code>?addParents=parent_id</code>
<code>children.list</code>	<code>files.list</code> with <code>?q='parent_id'+in+parents</code>
<code>comments.insert</code>	<code>comments.create</code>
<code>comments.patch</code>	<code>comments.update</code>
<code>files.insert</code>	<code>files.create</code>
<code>files.patch</code>	<code>files.update</code>
<code>files.touch</code>	<code>files.update</code> with <code>{'modifiedTime':'timestamp'}</code>
<code>files.trash</code>	<code>files.update</code> with <code>{'trashed':true}</code>
<code>files.untrash</code>	<code>files.update</code> with <code>{'trashed':false}</code>
<code>parents.delete</code>	<code>files.update</code> with <code>?removeParents=parent_id</code>

Неломаящие изменения в REST

10

- добавление новых путей, ресурсов и действий над ними
- расширение существующих моделей
- добавление/изменение/удаление ссылок (если HATEOS)

Методы версионирования

URI

`https://api/v2/Tasks/{TaskId}`

Query Parameters

`https://api/Tasks/{TaskId}?v=2`

Custom Header

`HTTP x-api-version: 2`

Media Type

`application/App.v2.param.json`

Версия в URI



Google Cloud

HTTP POST

`https://storage.googleapis.com/storage/v1/b`



DELETE `/2015-03-31/functions/FunctionName?Qualifier=Qualifier` HTTP/1.1

Версия в Query Path



Azure DevOps

HTTP GET

https://status.dev.azure.com/_apis/status/health?services={services}&api-version=6.0-preview.1



HTTP GET

<https://autoscaling.amazonaws.com/?Action=SetInstanceHealth&InstanceId=i-12345678 &HealthStatus=Unhealthy&Version=2011-01-01>

Версия в Media Type



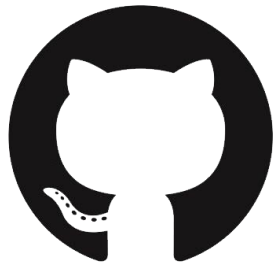
Amazon ECS

```
POST / HTTP/1.1
Host: ecs.us-east-1.amazonaws.com
Accept-Encoding: identity
Content-Length: 29
X-Amz-Target: AmazonEC2ContainerServiceV20141113.CreateCluster
X-Amz-Date: 20150429T163840Z
Content-Type: application/x-amz-json-1.1
Authorization: AUTHPARAMS
```

```
{
  "clusterName": "My-cluster"
}
```

Версия в X-Header

15



GitHub

```
curl https://api.github.com/users/technoweenie -I
HTTP/1.1 200 OK
X-GitHub-Media-Type: github.v3
curl https://api.github.com/users/technoweenie -I \
-H "Accept: application/vnd.github.full+json"
HTTP/1.1 200 OK
X-GitHub-Media-Type: github.v3; param=full; format=json
curl https://api.github.com/users/technoweenie -I \
-H "Accept: application/vnd.github.v3.full+json"
HTTP/1.1 200 OK
X-GitHub-Media-Type: github.v3; param=full; format=json
```

Комбинирование версий

16

Jobs In the current version, the REST API supports backup jobs with VMware VMs only.

GET `/api/v1/jobs` Get All Jobs

Gets an array of all jobs coordinated by the backup server.

Parameters

Name	Description
x-api-version * required string (header)	Version and revision of the client REST API. Must be in the following format: <code><version>-<revision></code> . <input type="text" value="1.0-rev1"/>

Реализации версионирования

Инкапсуляция предыдущей версии

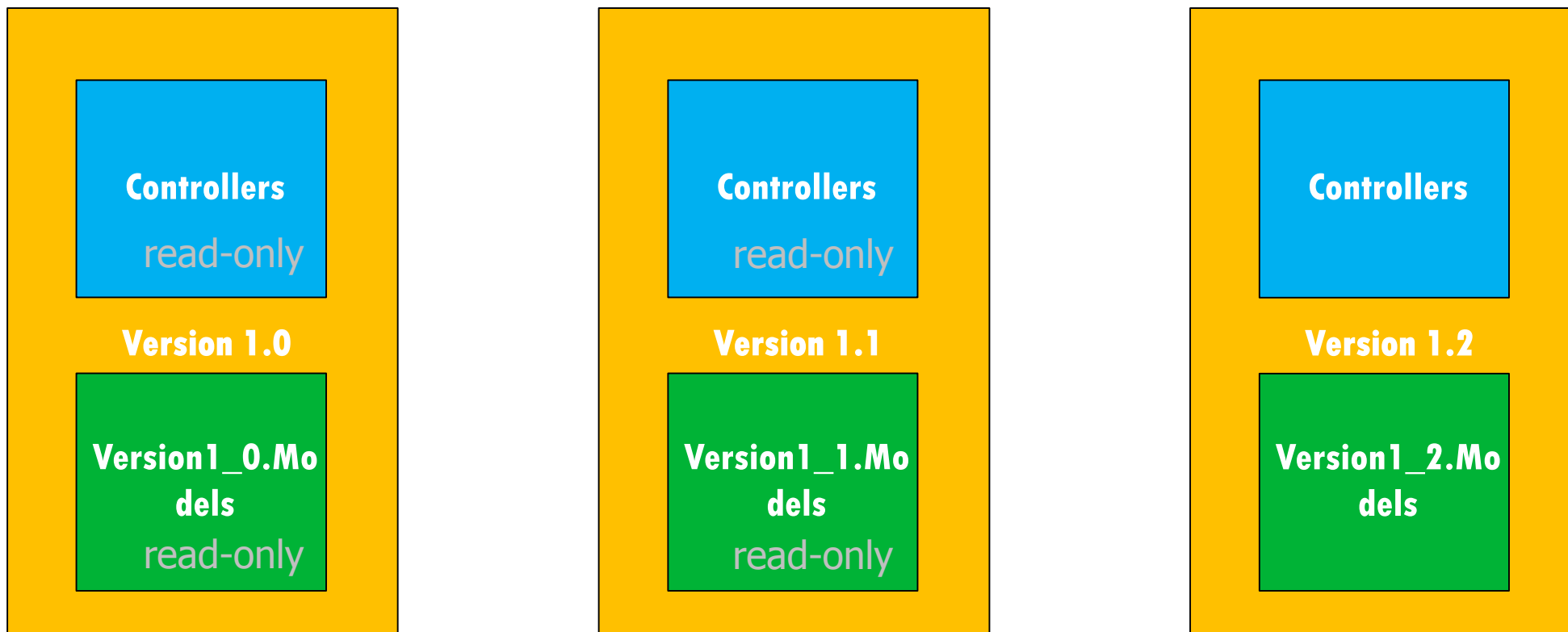
Отдельные модели и контроллеры
для нового контракта

Расширение предыдущей версии

Трансформация моделей между
версиями

Инкапсуляция предыдущей версии

18



Расширение предыдущей версии

```
public override async Task<ActionResult<UserModel>> CreateUser([Required]UserSpec userSpec)
{
    if (version == "v1.0")
    {
        UserModel userModel = CreateUserForVersion1_0(userSpec);
        return userModel;
    }

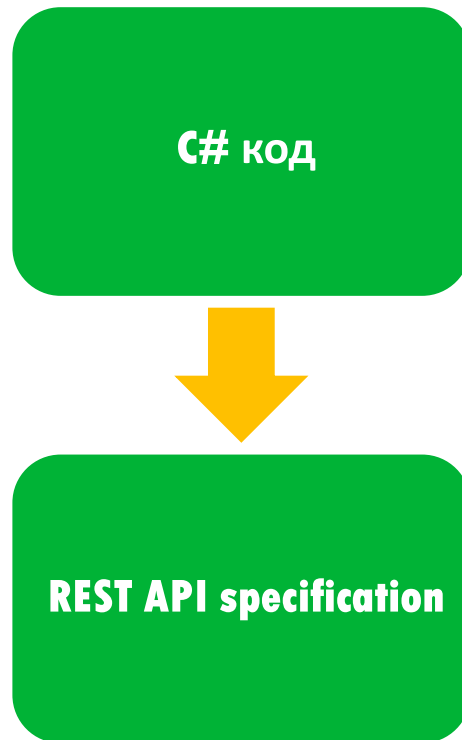
    if (version == "v1.1")
    {
        UserModel userModel = CreateUserForVersion1_1(userSpec);
        return userModel;
    }

    return BadRequest("Wrong version");
}
```

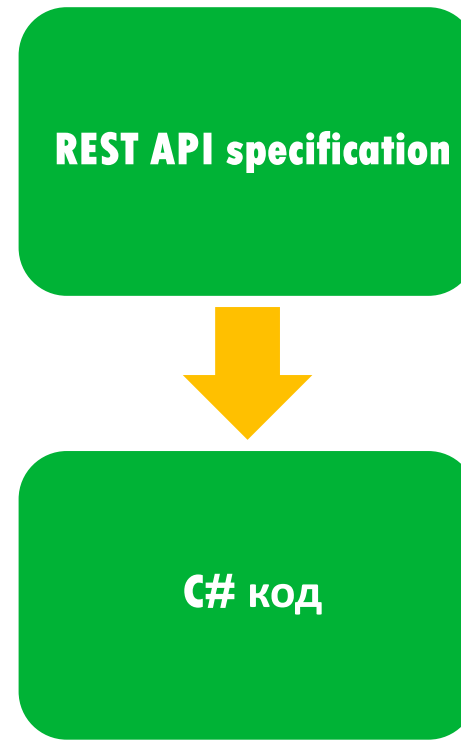
Проектирование REST API

20

Code-First

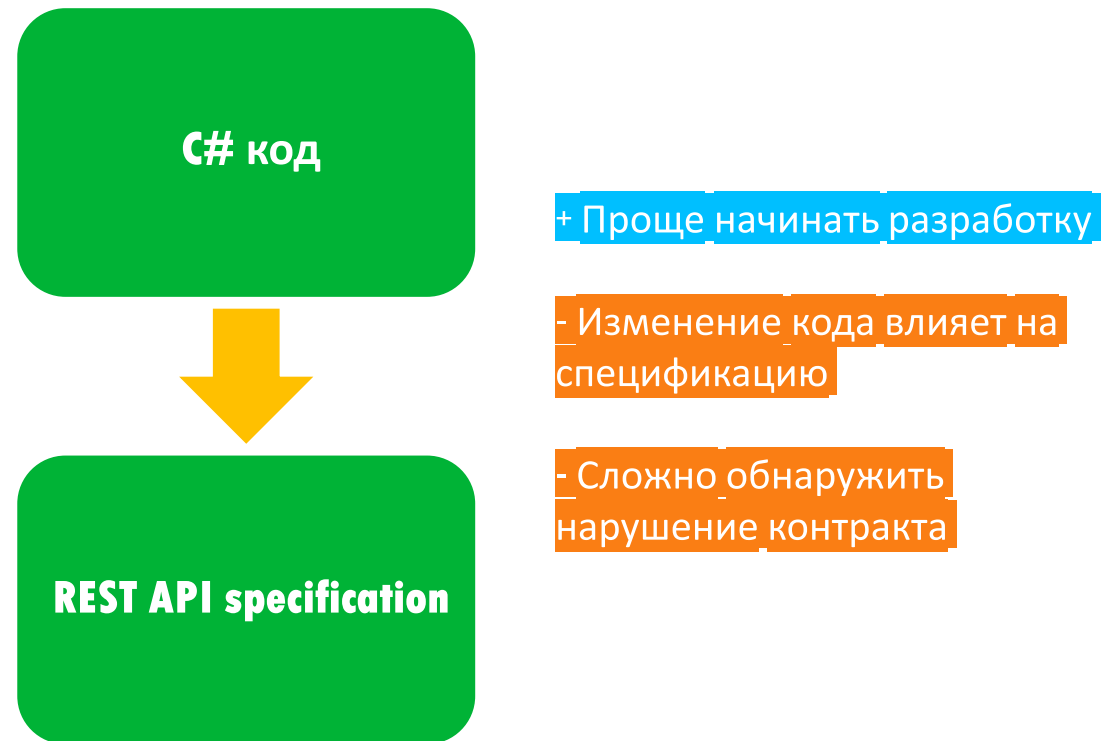


Contract-First



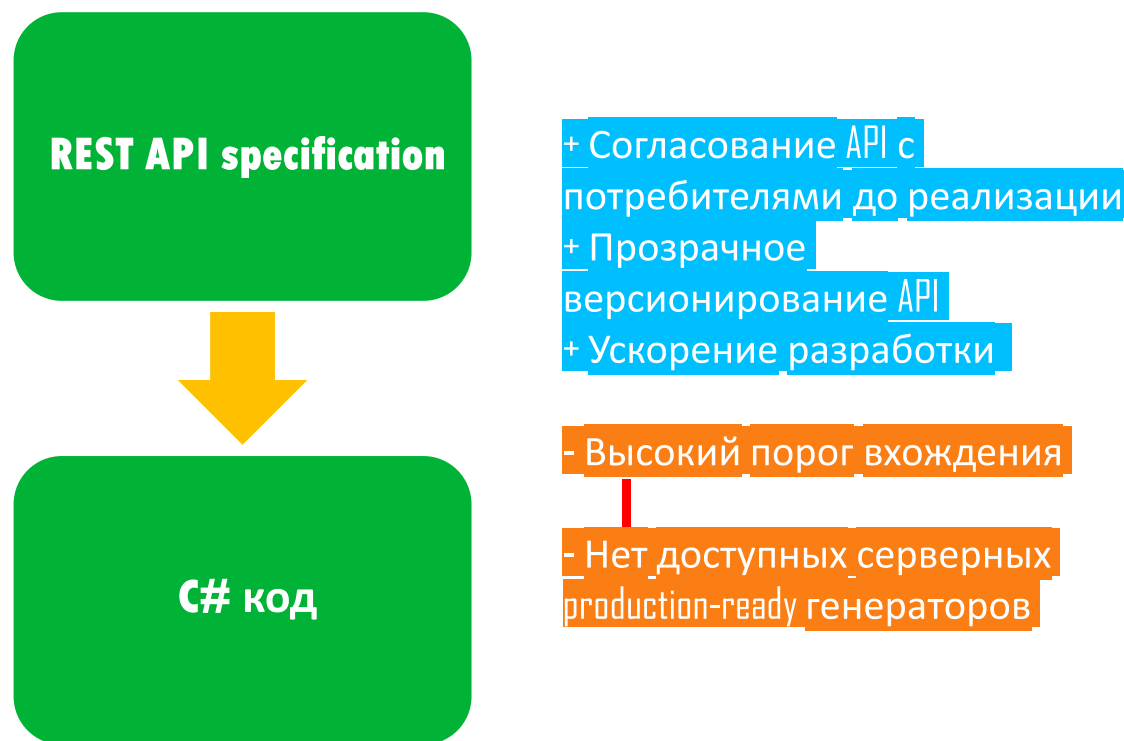
Code-First подход

Code-First



Contract-First подход

Contract-First



Почему Contract-first

- ✓ разработка клиента и сервера одновременно
- ✓ документация прямо в спецификации
- ✓ новые фичи – прямо в спецификации
- ✓ гарантированность контракта



Swagger Editor

editor.swagger.io

Incognito

Swagger Editor

File Edit Generate Server Generate Client

```
1 swagger: "2.0"
2 info:
3   description: "This is a sample server
4   Swagger at [http://swagger.io]
5   (http://swagger.io/irc/).
6   to test the authorization
7   version: "1.0.0"
8   title: "Swagger Petstore"
9   termsOfService: "http://swagger.
10  contact:
11   email: "apiteam@swagger.io"
12  license:
13   name: "Apache 2.0"
14   url: "http://www.apache.org/li
15  host: "petstore.swagger.io"
16  basePath: "/v2"
17  tags:
18   - name: "pet"
19     description: "Everything about yo
20   externalDocs:
21     description: "Find out more"
22     url: "http://swagger.io"
23   - name: "store"
24     description: "Access to Petstore
25   - name: "user"
26     description: "Operations about u
27   externalDocs:
28     description: "Find out more abo
29     url: "http://swagger.io"
30  schemes:
31   - "https"
32   - "http"
33  paths:
34   /pet:
35     post:
36       tags:
37       - "pet"
38       summary: "Add a new pet to the store"
39       description: ""
40       operationId: "addPet"
41       consumes:
42       - "application/json"
43       - "application/xml"
44       produces:
45       - "application/xml"
46       - "application/json"
47       parameters:
48       - in: "body"
```

ada-server	jaxrs-resteasy	restbed
aspnetcore	jaxrs-resteasy-eap	rust-server
erlang-server	jaxrs-spec	scala-lagom-server
finch	kotlin-server	scalatra
go-server	lumen	sinatra
haskell	msf4j	slim
inflector	nancyfx	spring
java-pkmst	nodejs-server	undertow
java-play-framework	php-silex	ze-ph
java-vertx	php-symfony	
jaxrs	pistache-server	
jaxrs-cxf	python-flask	
jaxrs-cxf-cdi	rails5	

store 1.0.0

2]

server. You can find out more about Swagger at <http://swagger.io> or on

sample, you can use the api key `special-key` to test the authorization filters.

Authorize

pet Everything about your Pets

Find out more

POST /pet Add a new pet to the store

PUT /pet Update an existing pet

GET /pet/findByStatus Finds Pets by status

GET /pet/findByTags Finds Pets by tags

GET /pet/{petId} Find pet by ID

Кодогенерация

```
30 paths:
31   /pet:
32     post:
33       tags:
34       - "pet"
35       summary: "Add a new pet to the store"
36       description: ""
37       operationId: "addPet"
38       consumes:
39       - "application/json"
40       - "application/xml"
41       produces:
42       - "application/xml"
43       - "application/json"
44       parameters:
45       - in: "body"
46         name: "body"
47         description: "Pet object that needs to be added to the store"
48         required: true
49         schema:
50           $ref: "#/definitions/Pet"
51       responses:
52       "405":
53         description: "Invalid input"
54       security:
55       - petstore_auth:
56         - "write:pets"
57         - "read:pets"
```

```

namespace CSharpClient.v3.Controllers
{
    [ApiController]
    public class PetApiController : ControllerBase
    {
        [HttpPost]
        [Route("/v2/pet")]
        [ValidateModelState]
        [SwaggerOperation("AddPet")]
        public virtual IActionResult AddPet([FromBody]Pet body)
        {
            throw new NotImplementedException();
        }

        [HttpDelete]
        [Route("/v2/pet/{petId}")]
        [ValidateModelState]
        [SwaggerOperation("DeletePet")]
        public virtual IActionResult DeletePet([FromRoute][Required]long? petId, [FromHeader]string apiKey)
        {
            throw new NotImplementedException();
        }

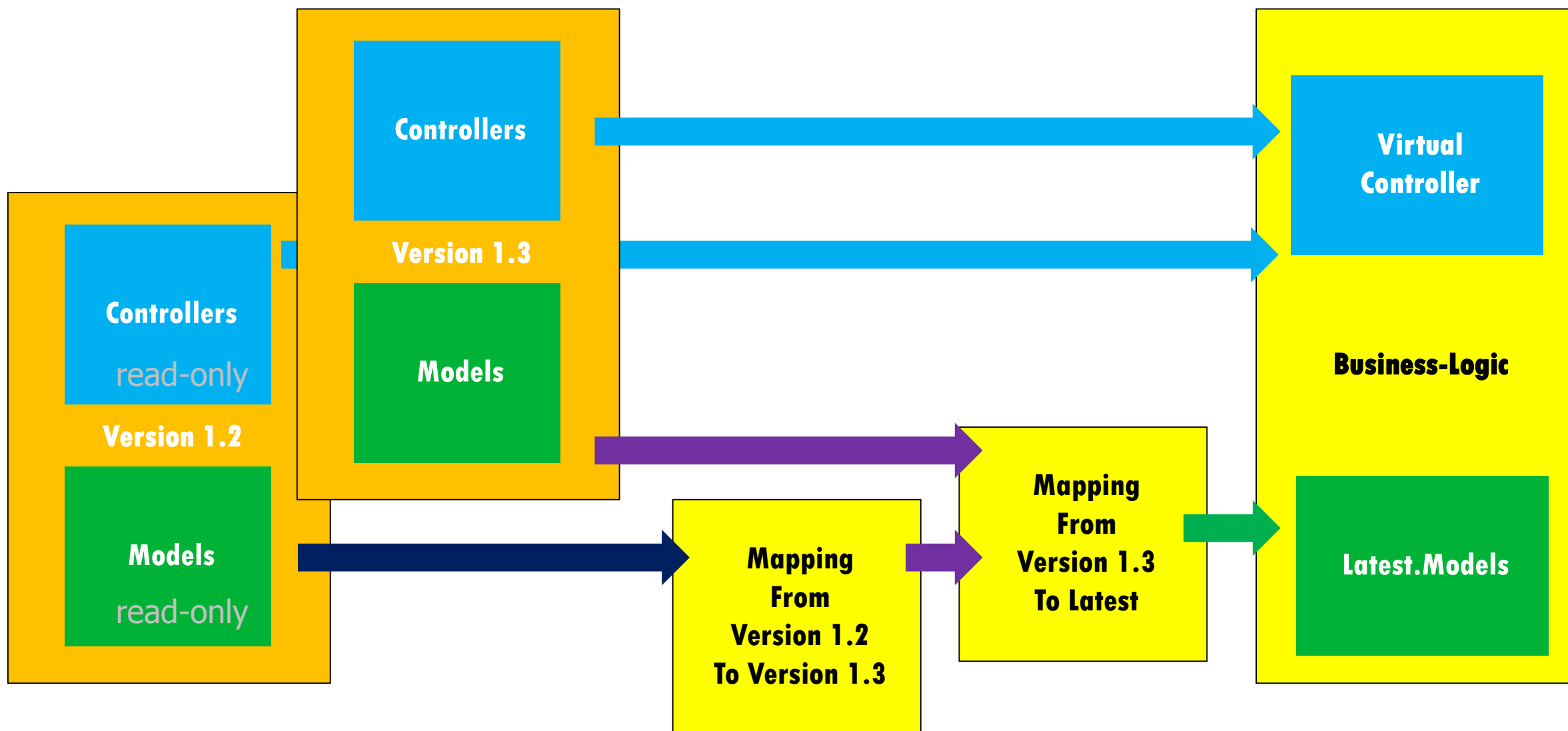
        [HttpGet]
        [Route("/v2/pet/findByStatus")]
        [ValidateModelState]
        [SwaggerOperation("FindPetsByStatus")]
        [SwaggerResponse(statusCode: 200, type: typeof(List<Pet>), description: "successful operation")]
        public virtual IActionResult FindPetsByStatus([FromQuery][Required()]List<string> status)
        {
            string exampleJson = null;
            var example = exampleJson != null
                ? JsonConvert.DeserializeObject<List<Pet>>(exampleJson)
                : default(List<Pet>); //TODO: Change the data returned
            return new ObjectResult(example);
        }
    }
}

```

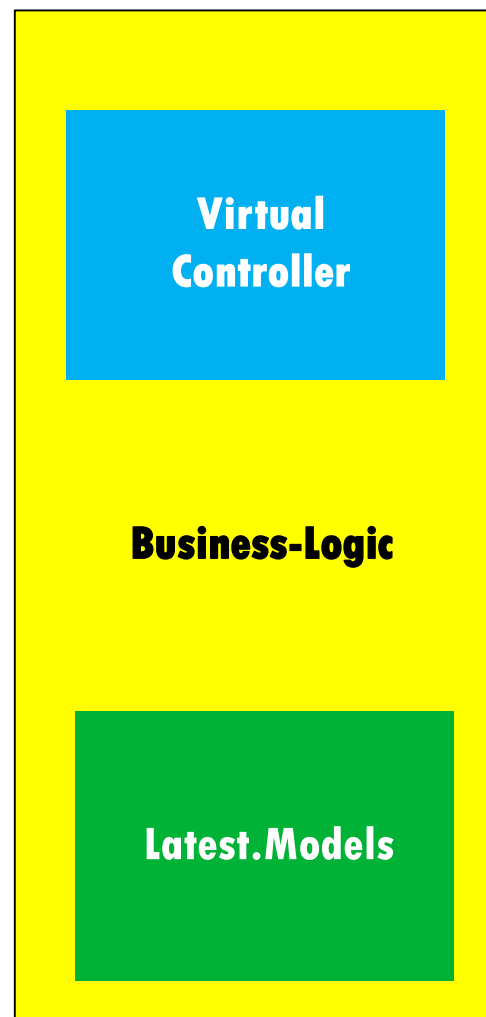
Основные требования

- ✓ Быстро добавлять новые ресурсы и пути
- ✓ Избежать дублирования бизнес-логики
- ✓ Минимизировать версионные изменения
- ✓ Гарантировать обратную совместимость

Принципиальная схема



Virtual controller



Virtual controller

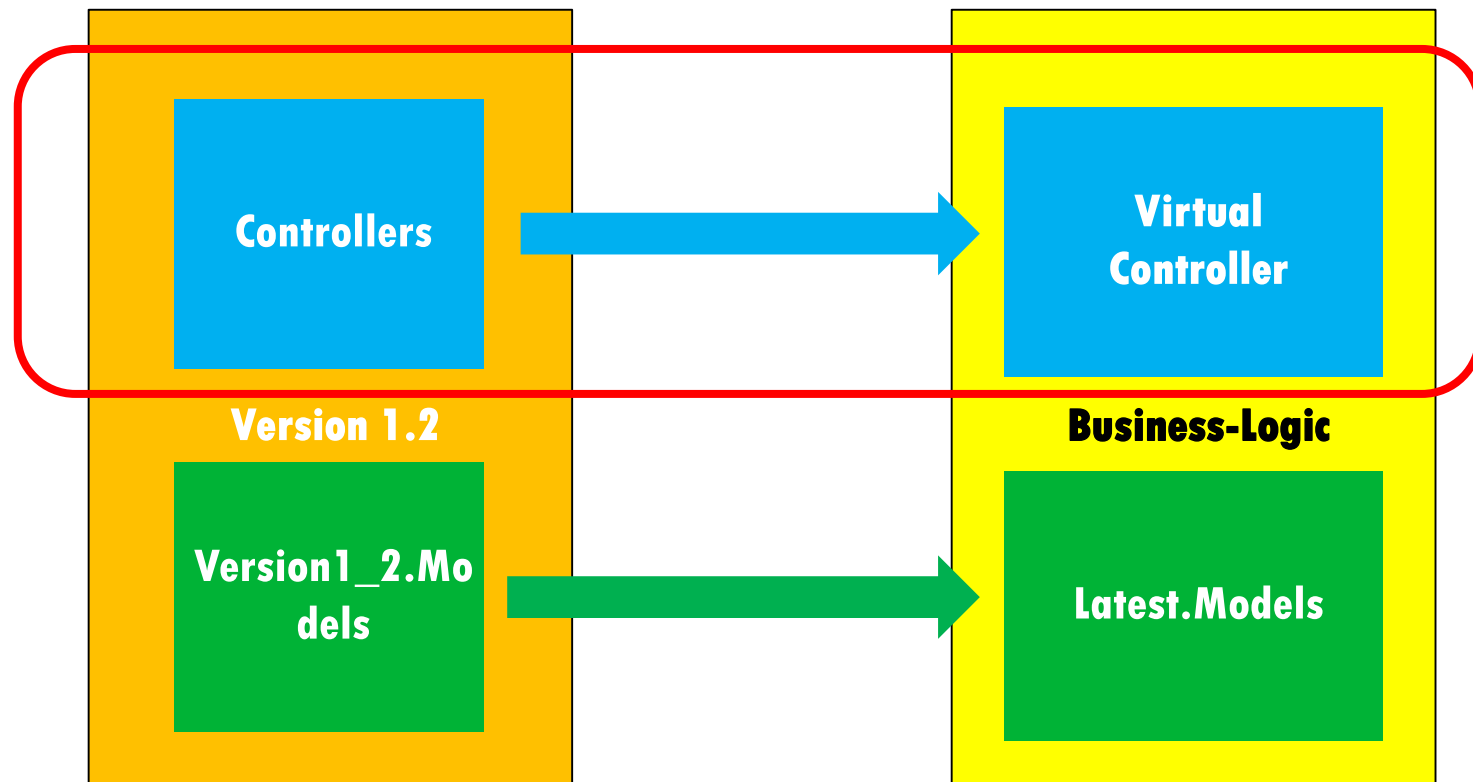
```
public sealed class UsersControllerImpl : VirtualControllerBase
{
    public async Task<VirtualActionResult<UserModel>> CreateUser( UserSpec spec )
    {
        if (!ValidateSpec(spec, out Exception exception))
            return BadRequest(exception.Message);

        return new UserModel("Ubuntu", "Example of creating user", false, EUsersType.DefaultAdmin);
    }

    private Boolean ValidateSpec( UserSpec spec, out Exception exception )
    {
        exception = null;
        //do smth
        return true;
    }
}
```

Абстракция бизнес-логики

32



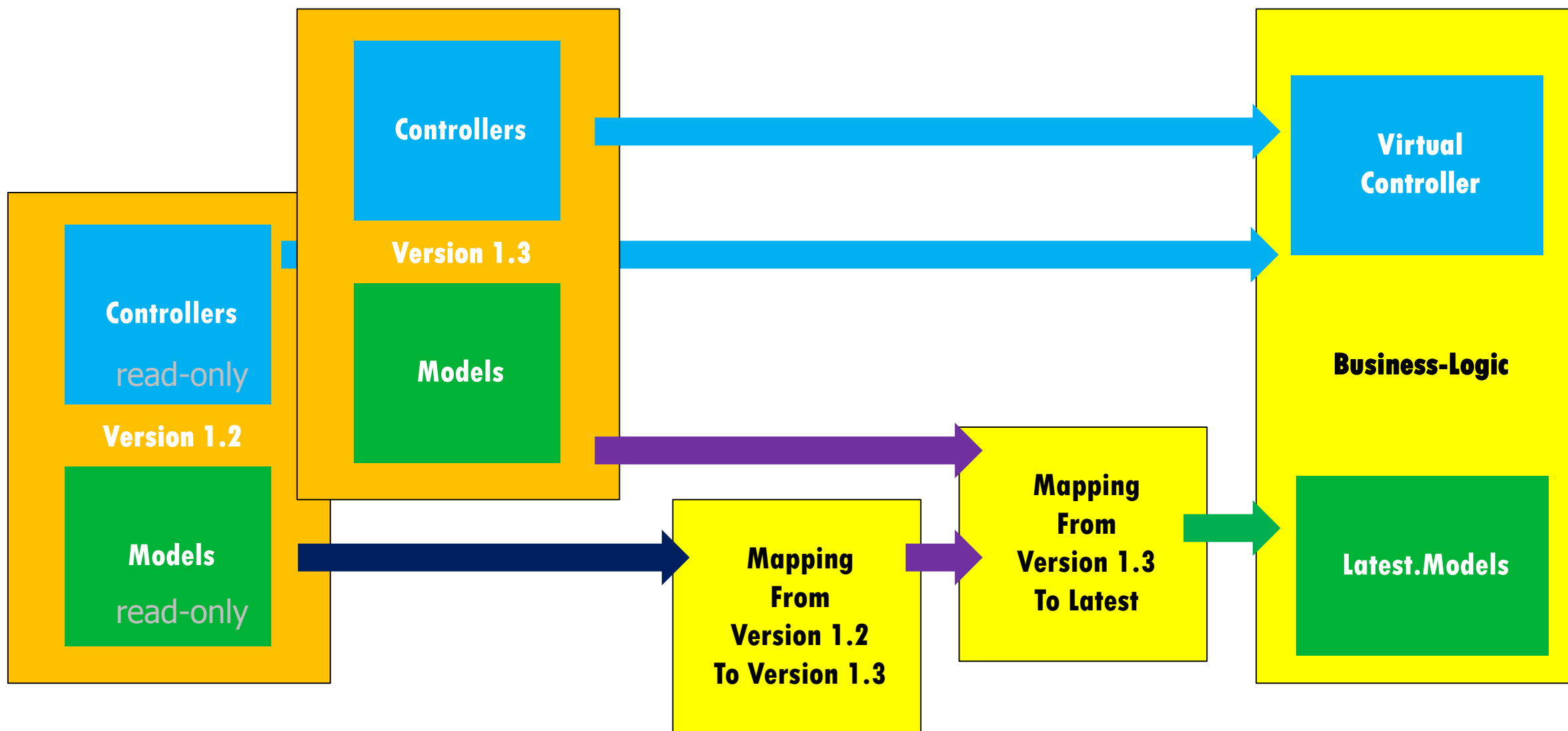
Абстракция бизнес-логики

```
public override async Task<ActionResult<UserModel>> CreateUser([Required]UserSpec userSpec)
{
    try
    {
        var result = await _impl.CreateUser(_mapper.Map<v_latest.Models.UserSpec>(userSpec));
        if (result.HasCustomResponse)
        {
            return ReturnResponse(result.CustomResponse);
        }
        return _mapper.Map<UserModel>(result.Value);
    }
    catch (Exception exception)
    {
        Log.Exception(exception);
        return ServerException(exception.Message);
    }
}
```

```
public async Task<VirtualActionResult<UserModel>> CreateUser( UserSpec spec )
{
    if (!ValidateSpec(spec, out Exception exception))
    {
        return BadRequest(exception.Message);
    }

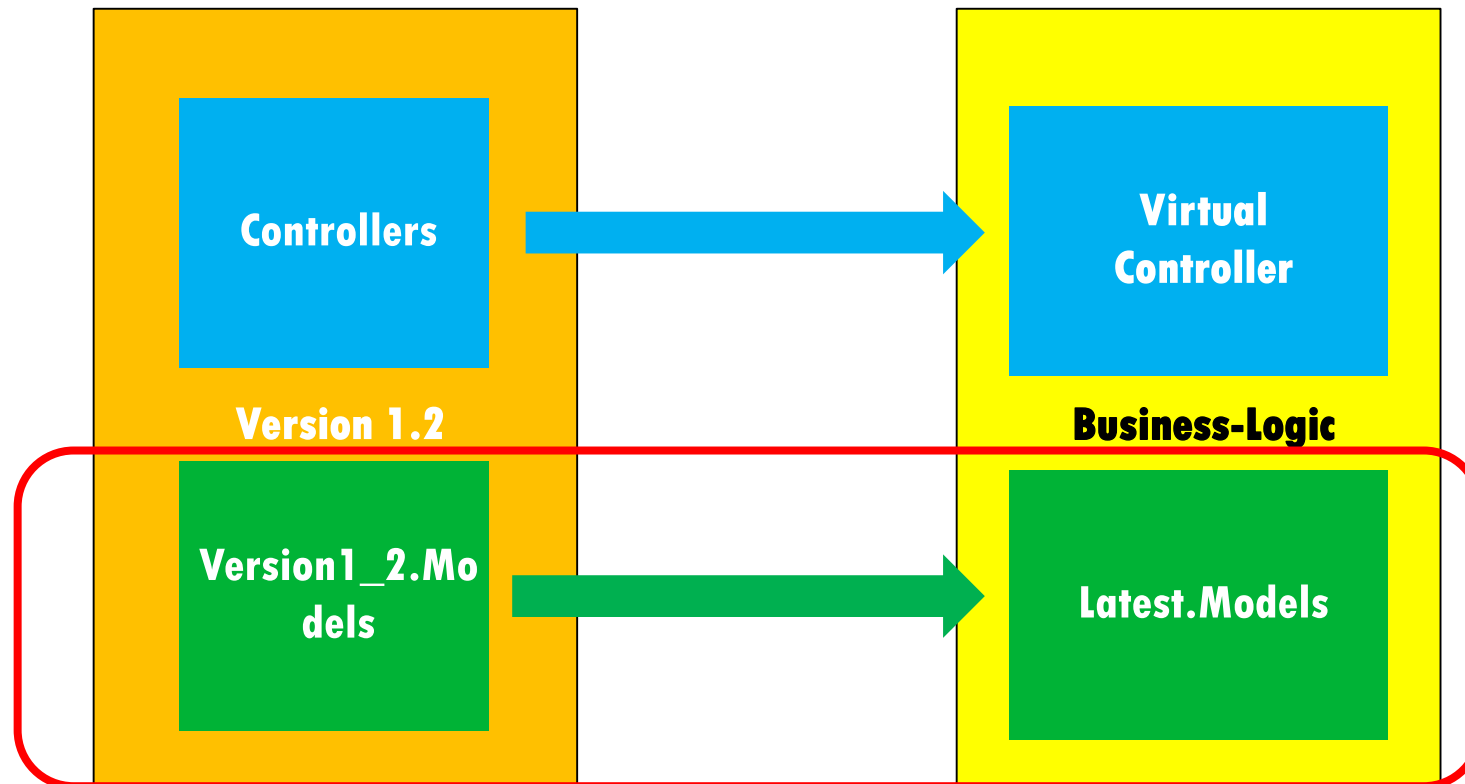
    return new UserModel("Ubuntu", "Example of creating user", false, EUsersType.DefaultAdmin);
}
```

Принципиальная схема



Абстракция моделей

35



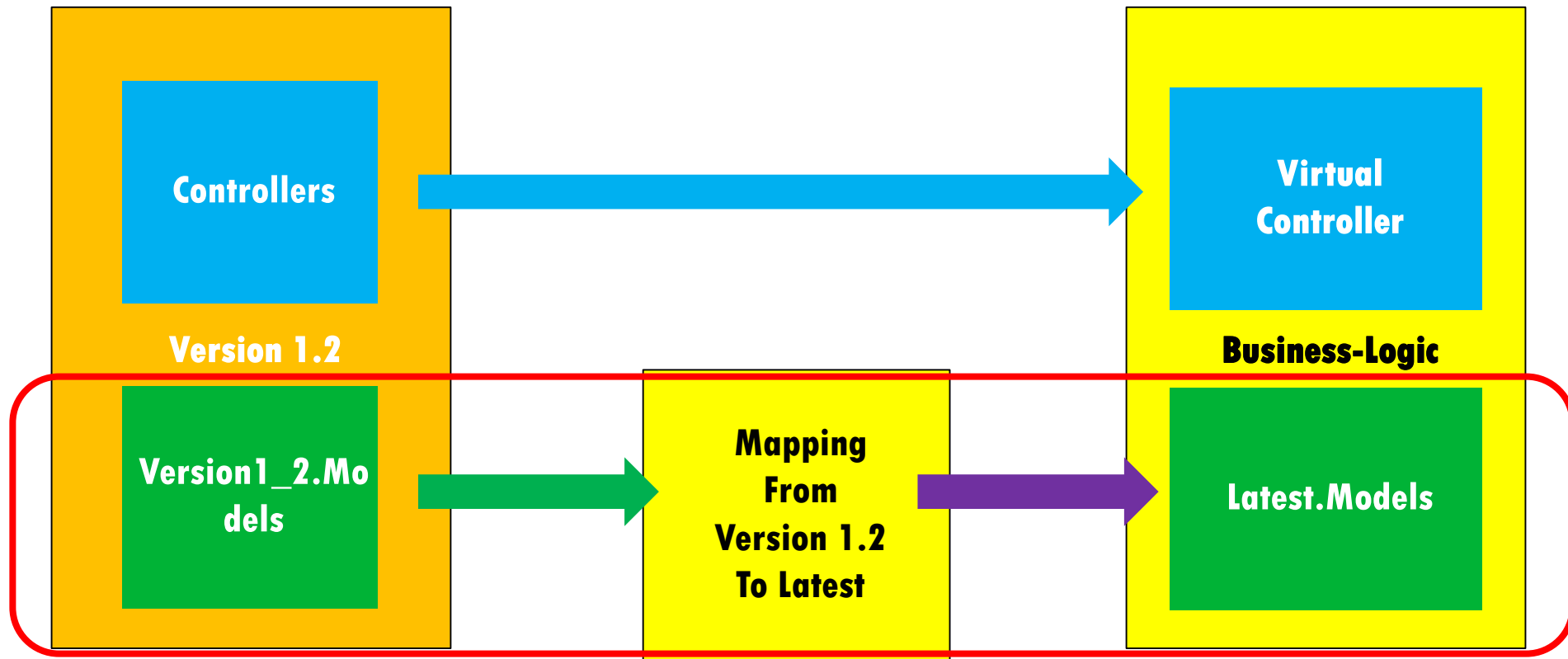
Абстракция моделей

```
namespace RestAPIService.Api.v_latest.Models
{
    public partial class CertificateModel
    { }
}
namespace RestAPIService.Api.v_1_2.Models
{
    public partial class CertificateModel
    { }
}
```

```
{
    /// <summary>
    /// CertificateModel
    /// </summary>
    [DataContract]
    public partial class CertificateModel
    {
        /// <summary>
        /// Initializes a new instance of the <see </summary>
        /// </summary>
        [JsonConstructorAttribute]
        public CertificateModel()...
```

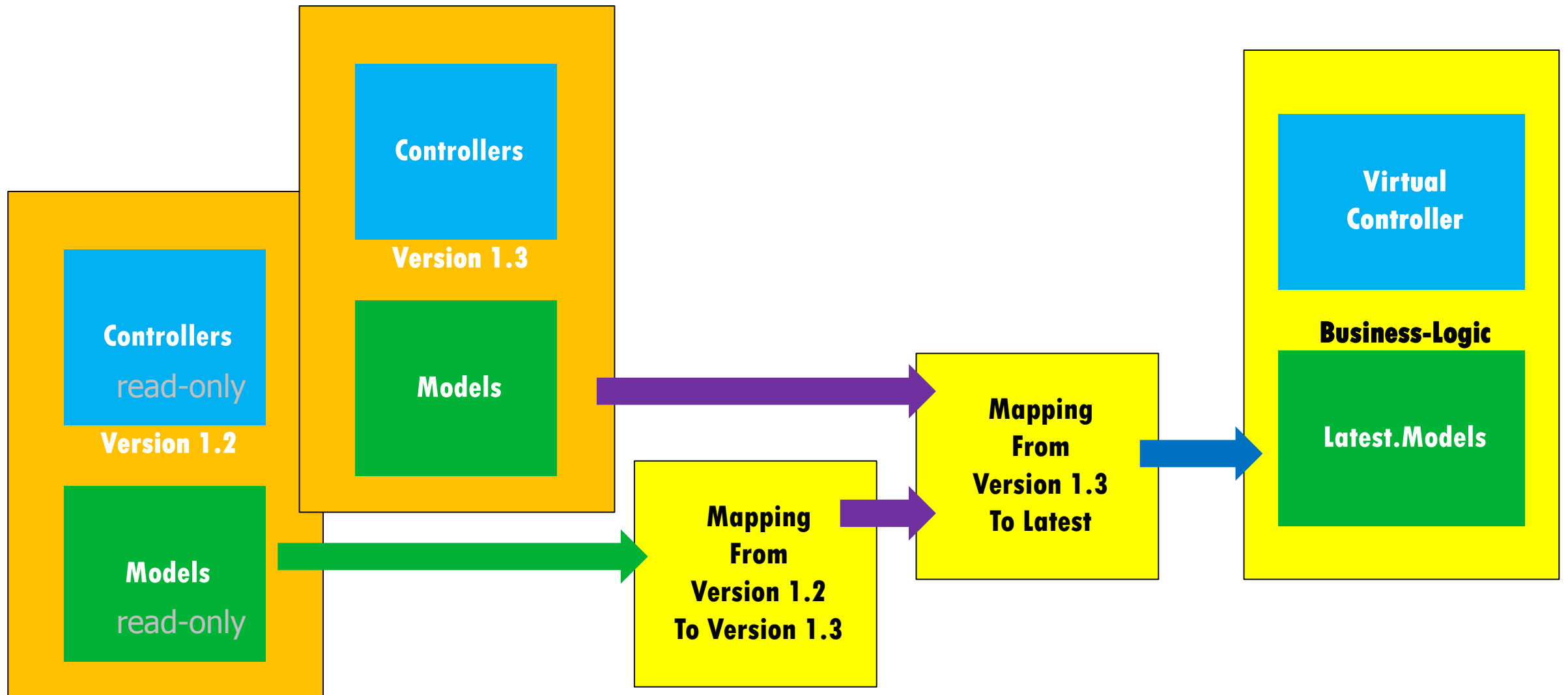
Трансформация моделей

37

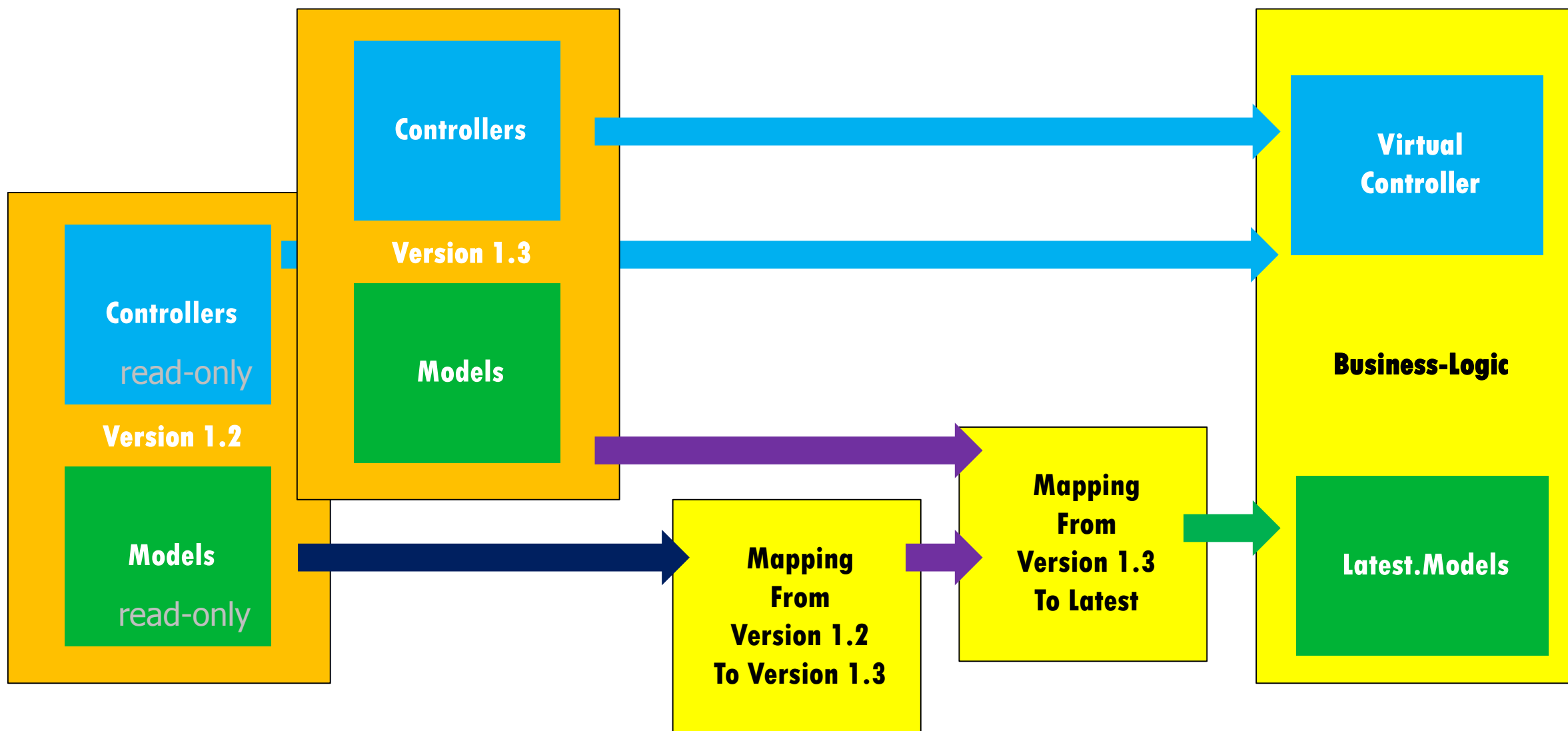


Трансформация моделей

38



Принципиальная схема



Полезные ресурсы для старта

КОДОГЕНЕРАТОРЫ:

<https://github.com/swagger-api/swagger-codegen>

<https://github.com/RicoSuter/NSwag>

ПОДДЕРЖКА ВЕРСИЙ:

<https://www.nuget.org/packages/Microsoft.AspNetCore.Mvc.Versioning>

ТРАНСФОРМАЦИЯ МОДЕЛЕЙ:

<https://www.nuget.org/packages/AutoMapper/>

РЕДАКТОР СПЕЦИФИКАЦИИ ДЛЯ **CONTRACT-FIRST** РАЗРАБОТКИ:

<https://editor.swagger.io/>

<https://swagger.io/blog/api-design/design-first-or-code-first-api-development/>

Мнение о кодогенераторе

41

«Он нам очень сильно сократил время разработки рест сервиса. Если бы делали с нуля, то думаю там несколько месяцев ушло бы, а так пара недель и уже можно было делать III и дёргать какие-то сервисные методы...»

Семен П., Разработчик

«Contract-first подход в разработке оказался очень удобным, просто описываешь спецификацию, а потом реализуешь бизнес-логику. Не надо вручную создавать контроллеры и модели»

Александр Ш., Разработчик

«Кодогенератор мы используем двух проектах. Он позволяет значительно ускорить работу и избавляет от необходимости писать рутинный код. Также автоматизированная генерация исключает возникновение ошибок в сгенерированном коде, особенно при рефакторинге – достаточно внести изменения в спецификацию и сгенерировать новые классы»

Андрей П., Разработчик



C# Developer



Full Stack C# Developer



C# API Developer



Technical Support Engineer



Спасибо!

veeam