СЛЁРМ Образование для инженеров и технических лидеров

Проектирование микросервисов на примере разделения монолита

Спикер: Щербаков Пётр
Solution Architect

План урока

01 Введение в нотацию C4

02 Выбор формата хранения контрактов

03 База знаний

Введение в нотацию C4

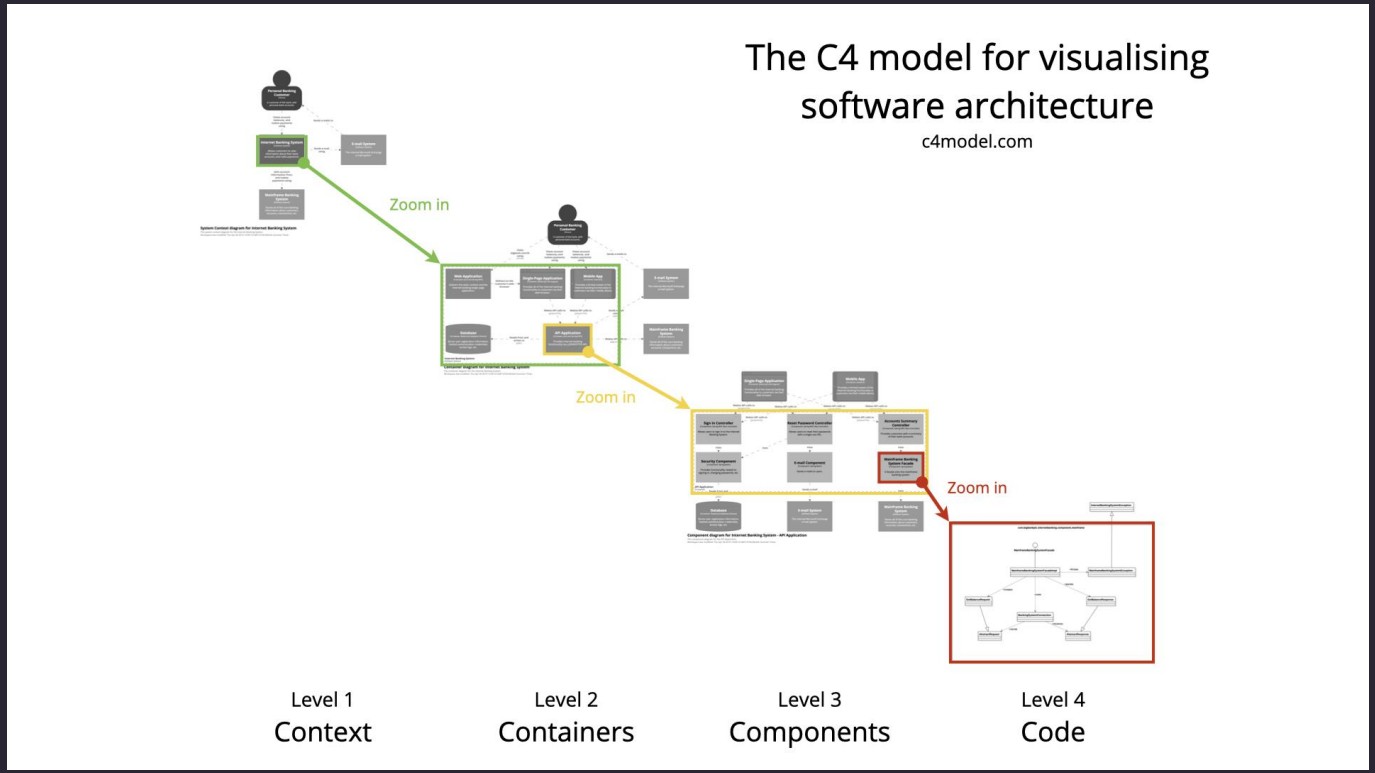
- Одна из самых простых нотаций
- Позволяет смотреть на архитектуру в режиме Zoom
- Подходит как для архитекторов так и разработчиков
- Позволяет смапить архитектуру на код

- https://c4model.com/

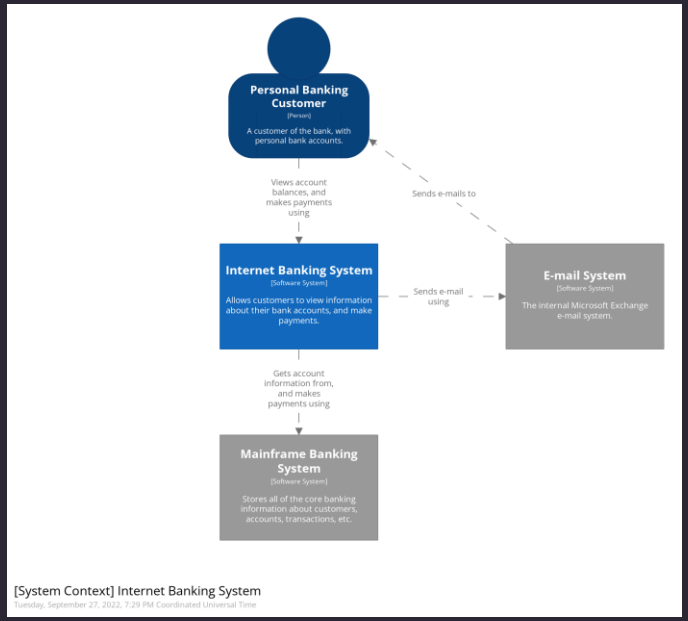
Введение в нотацию C4

- Server-side web application: Java EE (Apache Tomcat), ASP.NET MVC (Microsof IIS), Ruby on Rails (WEBrick), Node.js.
- Client-side web application: A JavaScript Angular, Backbone.JS, jQuery, etc.
- Client-side desktop application: A Windows desktop application written using WPF, an OS X desktop application written using Objective-C, a cross-platform desktop application written using JavaFX, etc.
- Mobile app: An Apple iOS app, an Android app
- Server-side console application: A standalone (e.g. "public static void main") application, a batch process, etc.
- Serverless function: A single serverless function (e.g. Amazon Lambda, Azure Function, etc).
- Database: A schema or database in a relational database management system, document store, graph database, etc such as MySQL, Microsoft SQL Server, Oracle Database, MongoDB, Riak, Cassandra, Neo4j, etc.
- Blob or content store: Amazon S3, Microsoft Azure Blob Storage, CDN
- File system: SAN, NAS
- Shell script: A single shell script written in Bash, etc.
- И многое другое

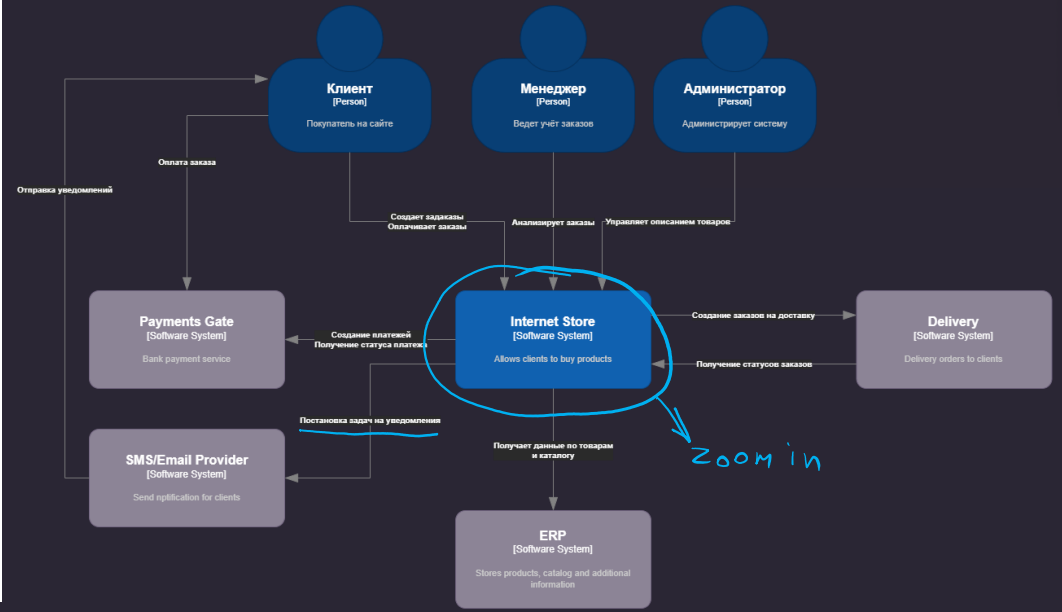
Введение в нотацию C4



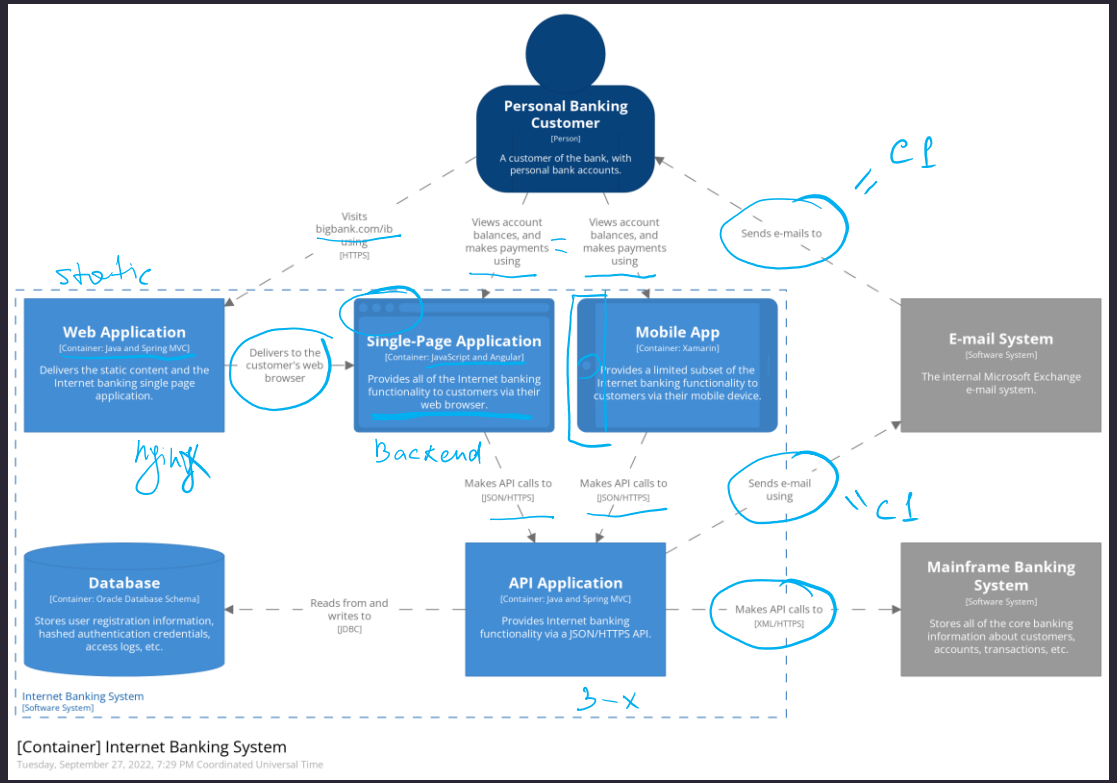
Введение в нотацию C4



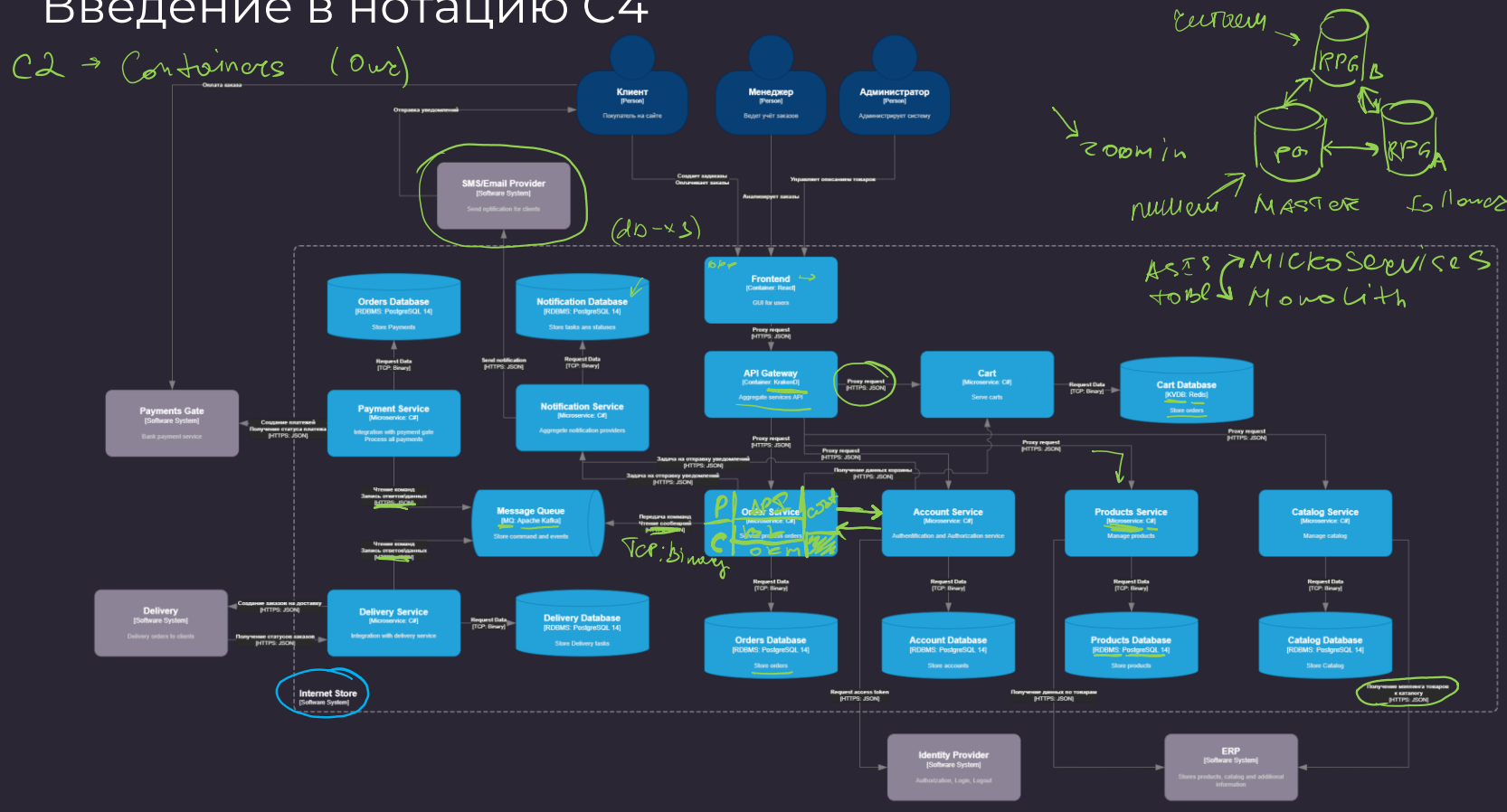
Context - C1



Введение в нотацию C4

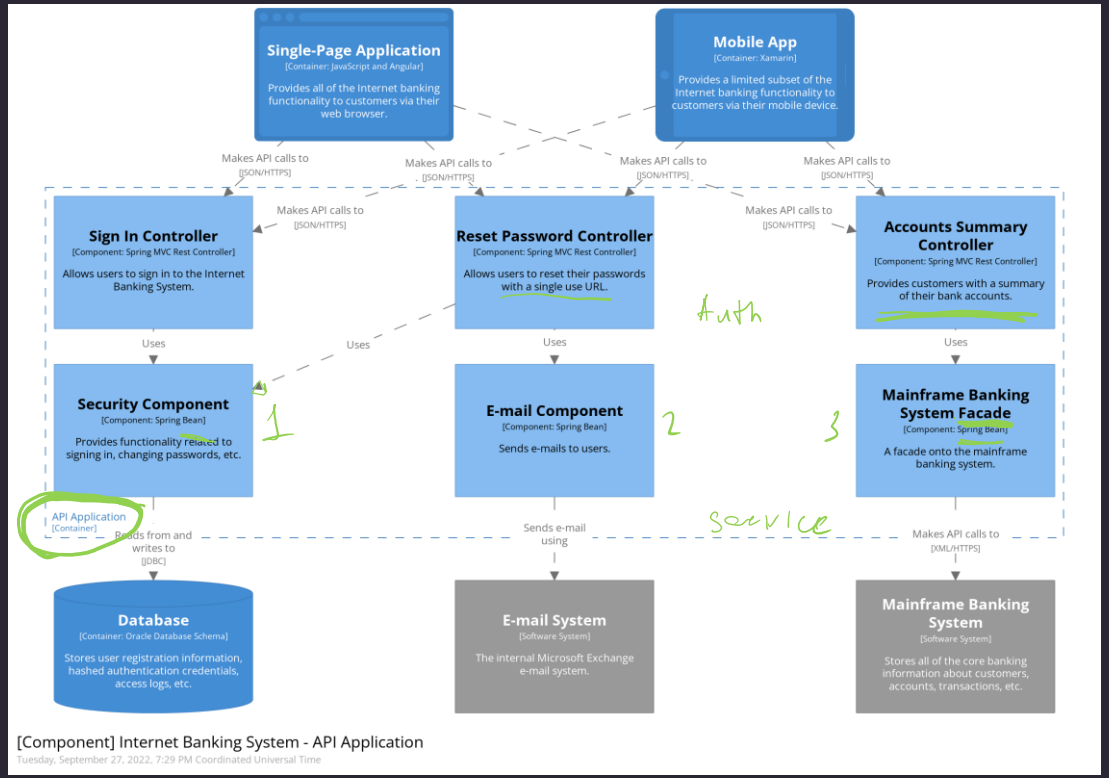


Введение в нотацию C4



Введение в нотацию C4

C 3 Component



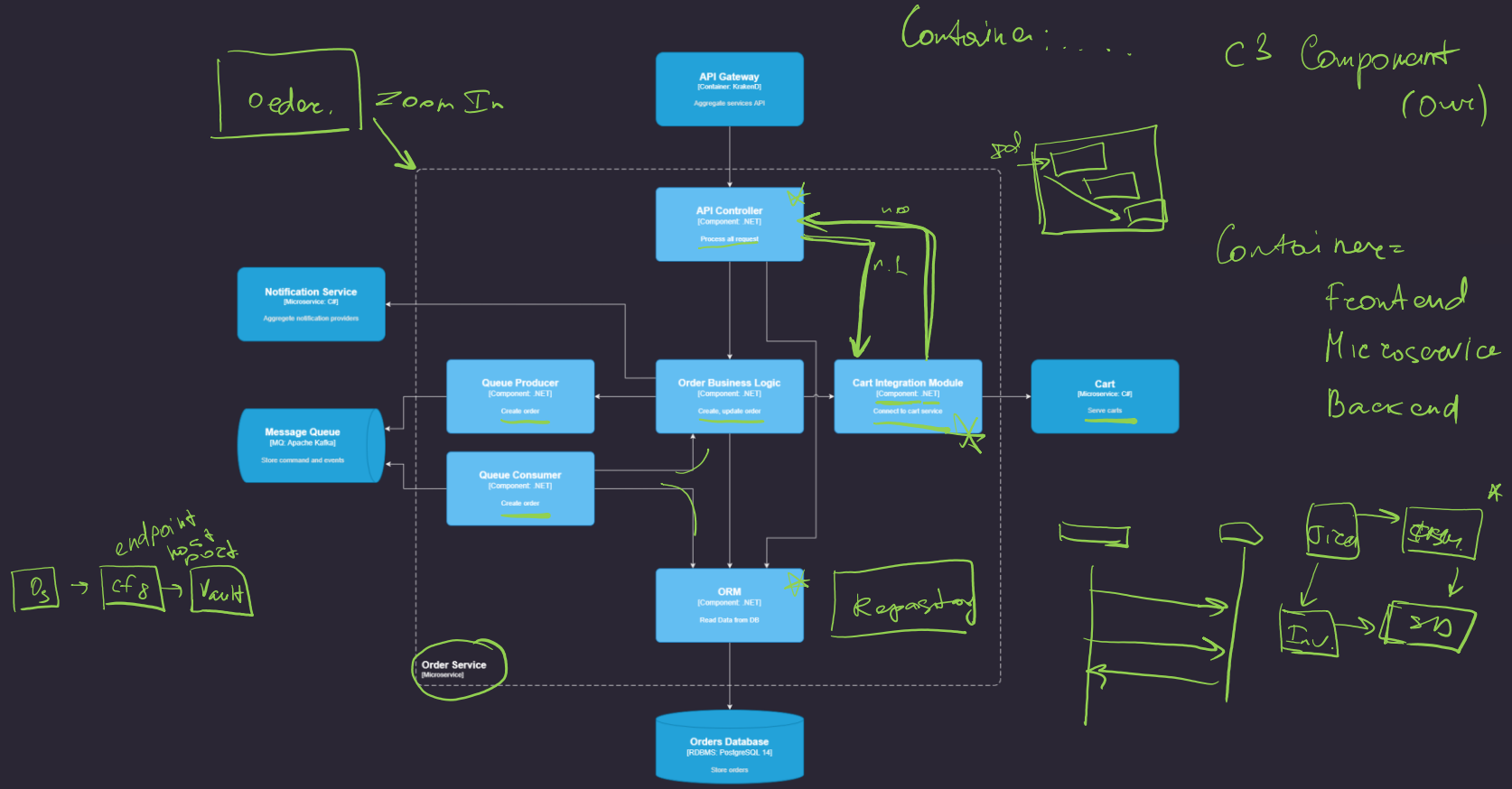
[Component] Internet Banking System - API Application

Tuesday, September 27, 2022, 7:29 PM Coordinated Universal Time

Handler

DAO

Введение в нотацию C4



Выбор формата хранения контрактов

Стандарты:
- Swagger
- AsyncApi

Где хранить?
- Git → JSON + APP
- Wiki
- Site (Swagger, AsyncApi) api.company.com/ru/sy

Выбор формата хранения контрактов

Swagger Docs

Сервис

```
{
  "title": "Sample Pet Store App",
  "description": "This is a sample server for a pet store.",
  "termsOfService": "http://example.com/terms/",
  "contact": {
    "name": "API Support",
    "url": "http://www.example.com/support",
    "email": "support@example.com"
  },
  "license": {
    "name": "Apache 2.0",
    "url": "https://www.apache.org/licenses/LICENSE-2.0.html"
  },
  "version": "1.0.1"
}
```

Среды

```
{
  "servers": [
    {
      "url": "https://development.gigantic-server.com/v2",
      "description": "Development server"
    },
    {
      "url": "https://staging.gigantic-server.com/v1",
      "description": "Staging server"
    },
    {
      "url": "https://api.gigantic-server.com/v1",
      "description": "Production server"
    }
  ]
}
```

Выбор формата хранения контрактов

Swagger

Динамическое описание сред

```
{
  "servers": [
    {
      "url": "https://{username}.gigantic-server.com:{port}/{basePath}",
      "description": "The production API server",
      "variables": {
        "username": {
          "default": "demo",
          "description": "this value is assigned by the service provider, in this example `gigantic-server.com`"
        },
        "port": {
          "enum": [
            "8443",
            "443"
          ],
          "default": "8443"
        },
        "basePath": {
          "default": "v2"
        }
      }
    }
  ]
}
```

Выбор формата хранения контрактов

Swagger

Объекты

```
"components": {
  "schemas": {
    "GeneralError": {
      "type": "object",
      "properties": {
        "code": {
          "type": "integer",
          "format": "int32"
        },
        "message": {
          "type": "string"
        }
      }
    },
```

Авторизация

```
  "securitySchemes": {
    "api_key": {
      "type": "apiKey",
      "name": "api_key",
      "in": "header"
    },
    "petstore_auth": {
      "type": "oauth2",
      "flows": {
        "implicit": {
          "authorizationUrl": "http://example.org/api/oauth/dialog",
          "scopes": {
            "write:pets": "modify pets in your account",
            "read:pets": "read your pets"
          }
        }
      }
```

Выбор формата хранения контрактов

Swagger

Параметры

```
40.   "parameters": {
41.     "skipParam": {
42.       "name": "skip",
43.       "in": "query",
44.       "description": "number of items to skip",
45.       "required": true,
46.       "schema": {
47.         "type": "integer",
48.         "format": "int32"
49.       }
50.     },
```

Ответы

```
62.   "responses": {
63.     "NotFound": {
64.       "description": "Entity not found."
65.     },
66.     "IllegalInput": {
67.       "description": "Illegal input for operation."
68.     },
69.     "GeneralError": {
70.       "description": "General Error",
71.       "content": {
72.         "application/json": {
73.           "schema": {
74.             "$ref": "#/components/schemas/GeneralError"
75.           }
```

Выбор формата хранения контрактов

AsyncAPI Docs

healthendpoint = method(s)

Сервис

Swagger.

```
{
  "title": "AsyncAPI Sample App",
  "description": "This is a sample server.",
  "termsOfService": "https://asyncapi.org/terms/",
  "contact": {
    "name": "API Support",
    "url": "https://www.asyncapi.org/support",
    "email": "support@asyncapi.org"
  },
  "license": {
    "name": "Apache 2.0",
    "url": "https://www.apache.org/licenses/LICENSE-2.0.html"
  },
  "version": "1.0.1"
}
```

Среды

```
{
  "production": {
    "url": "development.gigantic-server.com",
    "description": "Development server",
    "protocol": "kafka",
    "protocolVersion": "1.0.0"
  }
}
```

amqp.

amqp 0.9.1

Выбор формата хранения контрактов

AsuncApi

Динамическое описание сред

```
{
  "servers": {
    "production": {
      "url": "{username}.gigantic-server.com:{port}/{basePath}",
      "description": "The production API server",
      "protocol": "secure-mqtt",
      "variables": {
        "username": {
          "default": "demo",
          "description": "This value is assigned by the service provider, in this
        },
        "port": {
          "enum": [
            "8883",
            "8884"
          ],
          "default": "8883"
        },
        "basePath": {
          "default": "v2"
        }
      }
    }
  }
}
```

Выбор формата хранения контрактов

AsyncApi

Каналы

```
{
  "user/signedup": {
    "subscribe": {
      "message": {
        "$ref": "#/components/messages/userSignedUp"
      }
    }
  }
}
```

Каналы с параметрами

```
{
  "user/{userId}/signup": {
    "parameters": {
      "userId": {
        "description": "Id of the user.",
        "schema": {
          "type": "string"
        }
      }
    },
    "subscribe": {
      "message": {
        "$ref": "#/components/messages/userSignedUp"
      }
    }
  }
}
```

Выбор формата хранения контрактов

AsyncAPI

Объекты

```
{
  "type": "object",
  "required": [
    "name"
  ],
  "properties": {
    "name": {
      "type": "string"
    },
    "address": {
      "$ref": "#/components/schemas/Address"
    },
    "age": {
      "type": "integer",
      "format": "int32",
      "minimum": 0
    }
  }
}
```

Сообщения

```
{
  "messageId": "userSignup",
  "name": "UserSignup",
  "title": "User signup",
  "summary": "Action to sign a user up.",
  "description": "A longer description",
  "contentType": "application/json",
  "tags": [
    { "name": "user" },
    { "name": "signup" },
    { "name": "register" }
  ],
```

mimeType
enum

Выбор формата хранения контрактов

AsyncAPI

Payload сообщения

```
26    "payload": {
27      "type": "object",
28      "properties": {
29        "user": {
30          "$ref": "#/components/schemas/userCreate"
31        },
32        "signup": {
33          "$ref": "#/components/schemas/signup"
34        }
35      }
36    },
```

Сообщения

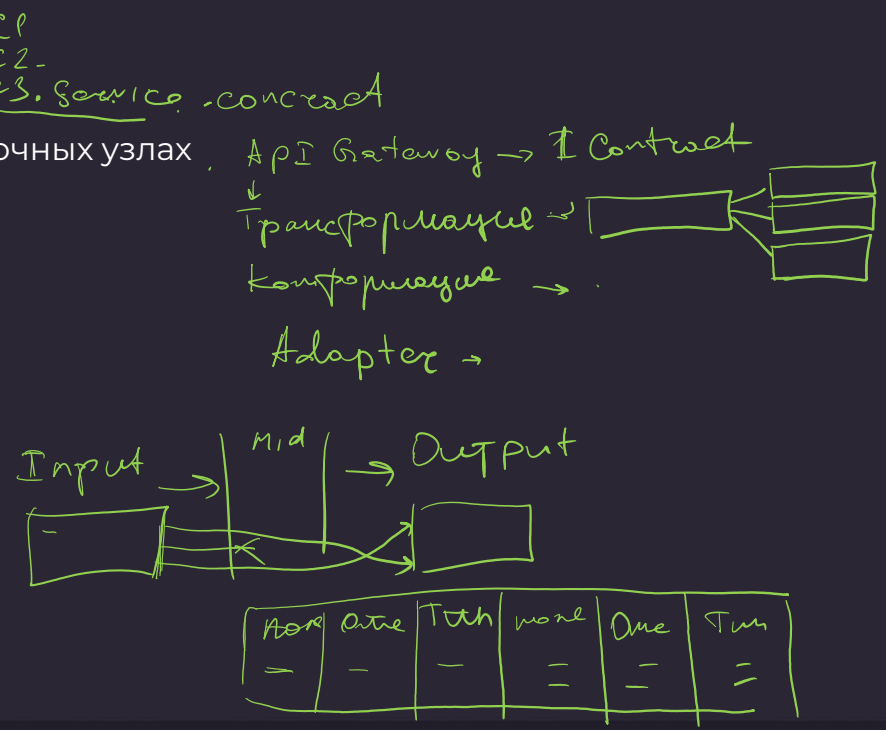
```
41    "traits": [
42      { "$ref": "#/components/messageTraits/commonHeaders" }
```

```
1  {
2    "schemaFormat": "application/vnd.apache.avro+json;version=1.9.0",
3    "contentType": "application/json"
4  }
```

База знаний

Наполнение базы знаний:

- Архитектура
- Описание сервисов
- Описание контрактов
- Описание контрактов на промежуточных узлах



База знаний

Наполнение базы знаний:

- Архитектура ✓
- Описание сервисов ✓
- Описание контрактов ✓

Требования к описанию знаний:

- Ссылки – ref.
- Теги – tag.
- Описание знаний → C4 – Link – Схема уровне / контекс / для сервис / система

Авторство

Дата.

Для рисования

А теперь перейдём в **real-time**