《Node.js / NestJS 在 Authing 架构中的演进和工程实践》

李洋洋

北京蒸汽记忆科技有限公司

自我介绍

Authing 全栈开发工程师

曾担任 ThoughWorks 高级咨询师、百度高级研发工程师

在大型企业服务架构、敏捷开发、DevOps、云原生等领域都具有较强的实践经验。

目录

无所不能的 JavaScript

Node.js 的应用场景

顺势而来的 TypeScript

TypeScript 带给我们的想象空间

NestJS 的到来

不止于 Web 框架

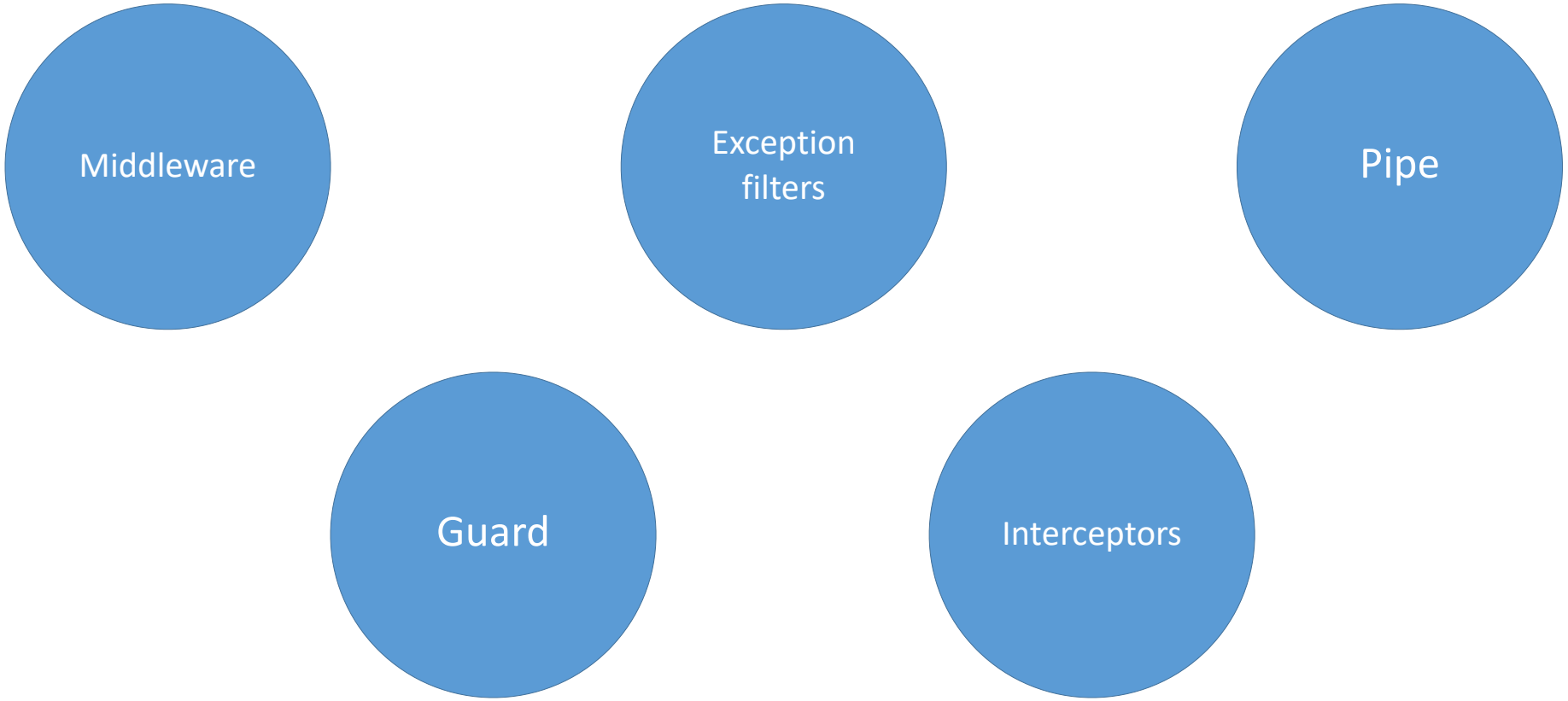
新思想

原生TS支持

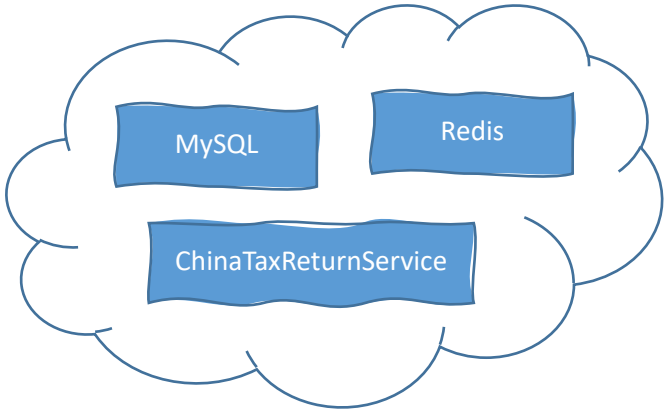
更多的探索

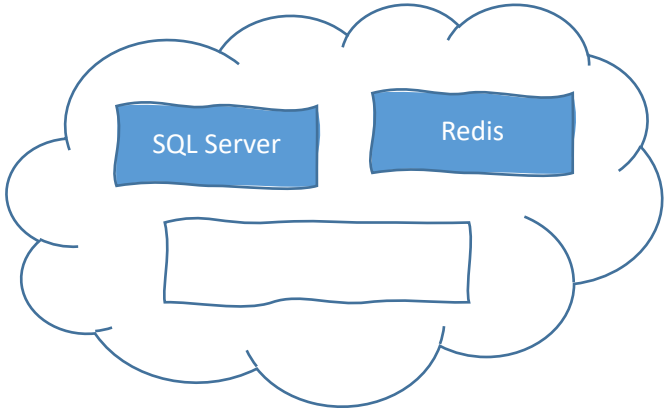
Nest 核心 – 模块系统

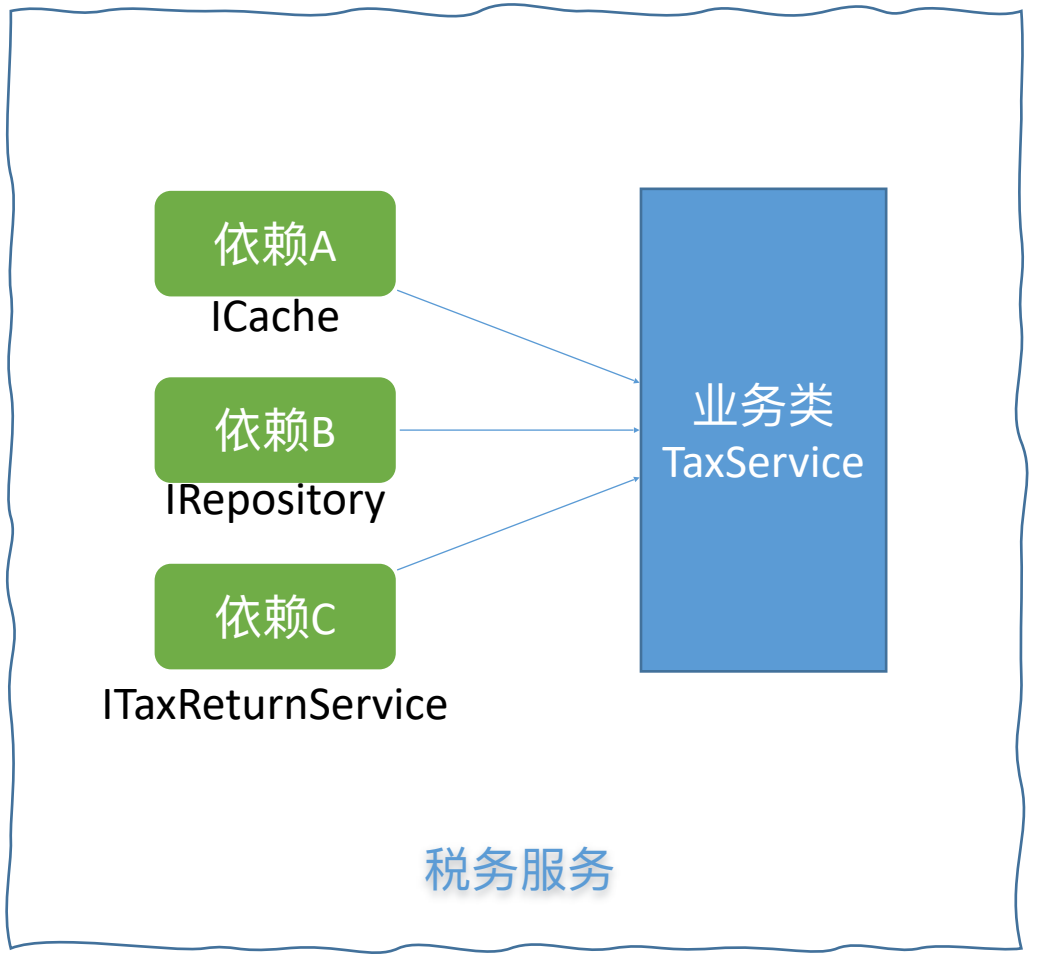
Nest 核心 – AOP



Nest 核心 – IoC 和 DI







框架之外

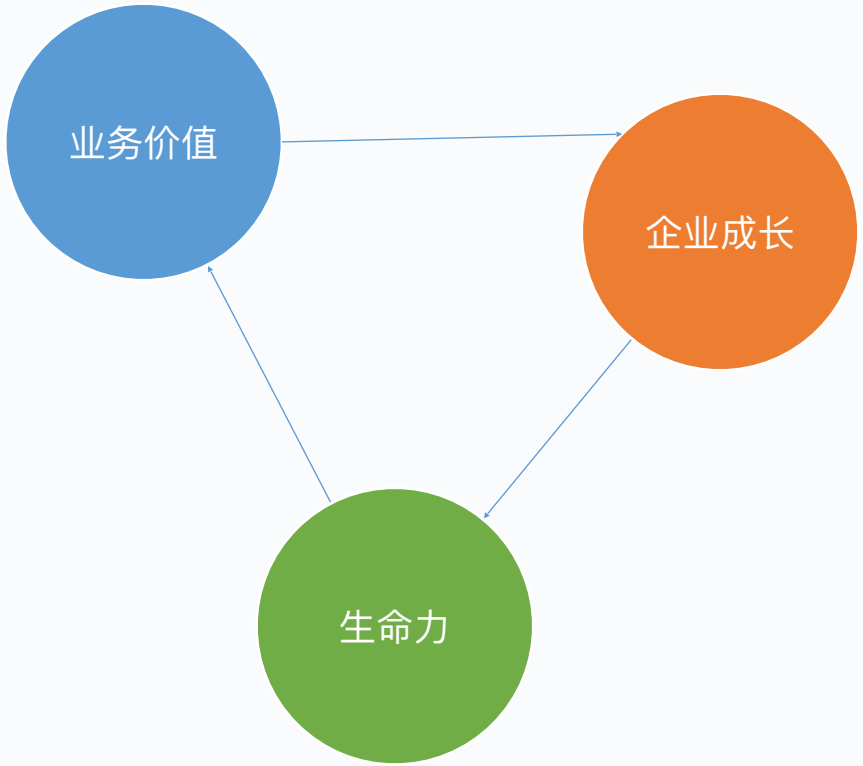
程序不止于框架

目录

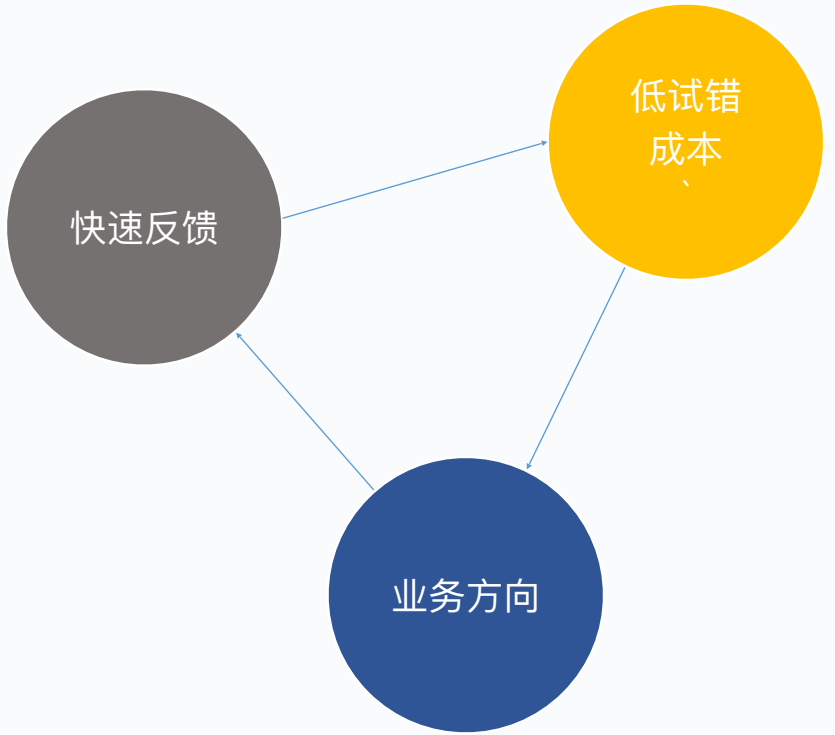
什么是敏捷开发

敏捷之道

业务价值



快速反馈



敏捷实践中的痛点

比较难解决的痛点

并行开发

DB 变更追踪

类型系统

产品质量保障

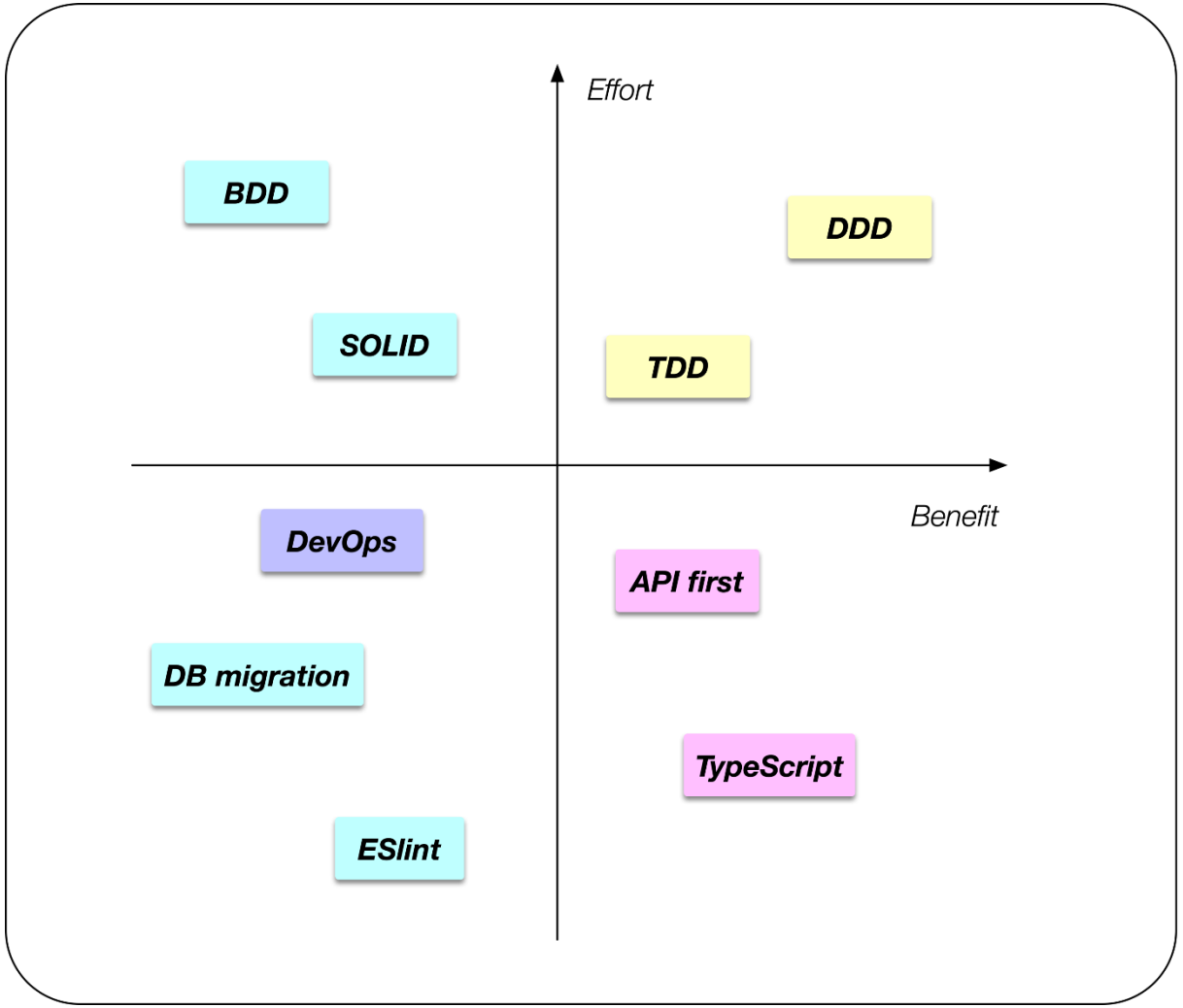
Validation

模块拆分

比较容易解决的痛点

编码风格

拼写检查



API First

前后端分离项目协作

前后端开发协作

团队较小，业务专注，沟通交流便捷，面对面随时沟通

前后端开发协作

团队人员变多，业务增多，一个人有时候会负责多个业务场景，实时沟通成本过高

前后端开发协作

Swagger

express-openapi-validator

Stoplight

基于 open API 3.0 的解决方案

前后端开发协作

```yaml
/user:
  post:
    summary: Create New User
    operationId: post-user
    responses:
      '200':
        description: User Created
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/User'
            examples:
              New User Bob Fellow:
                value:
                  id: 12
                  firstName: Bob
                  lastName: Fellow
                  email: bob.fellow@gmail.com
                  dateOfBirth: '1996-08-24'
                  emailVerified: false
                  createDate: '2020-11-18'
      '400':
        description: Missing Required Information
      '409':
        description: Email Already Taken
    requestBody:
      content:
        application/json:
          schema:
            type: object
            properties:
              firstName:
                type: string
                minLength: 3
                maxLength: 12
              lastName:
                type: string
                maxLength: 16
                minLength: 6
              email:
                type: string
              dateOfBirth:
                type: string
                format: date
            examples:
              Create User Bob Fellow:
                value:
                  firstName: Bob
                  lastName: Fellow
                  email: bob.fellow@gmail.com
                  dateOfBirth: '1996-08-24'
      description: Post the necessary fields for the API to create a new user.
    description: Create a new user.
```

Docs Try It

Create New User

Create a new user.

Request

POST /user

Body application/json

Post the necessary fields for the API to create a new user.

firstName string
>= 3 characters <= 12 characters

lastName string
>= 6 characters <= 16 characters

email string

dateOfBirth string<date>

Responses 200 400 409

User Created

Body application/json

id integer required
Unique identifier for the given user.

firstName string required

lastName string required

email string<email> required

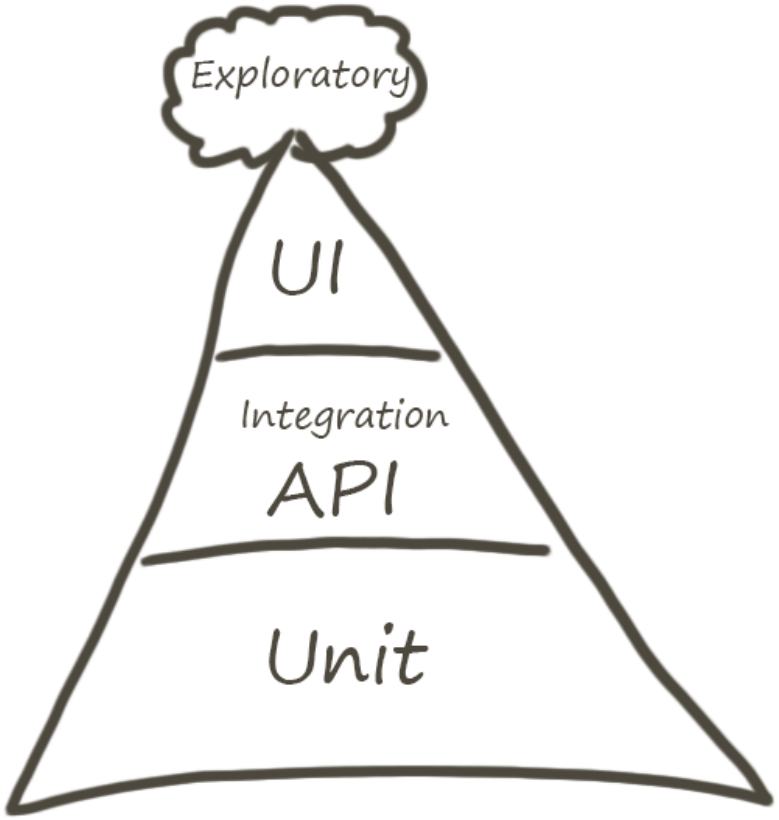
dateOfBirth string<date>

emailVerified boolean required
Set to true if the user's email has been verified.

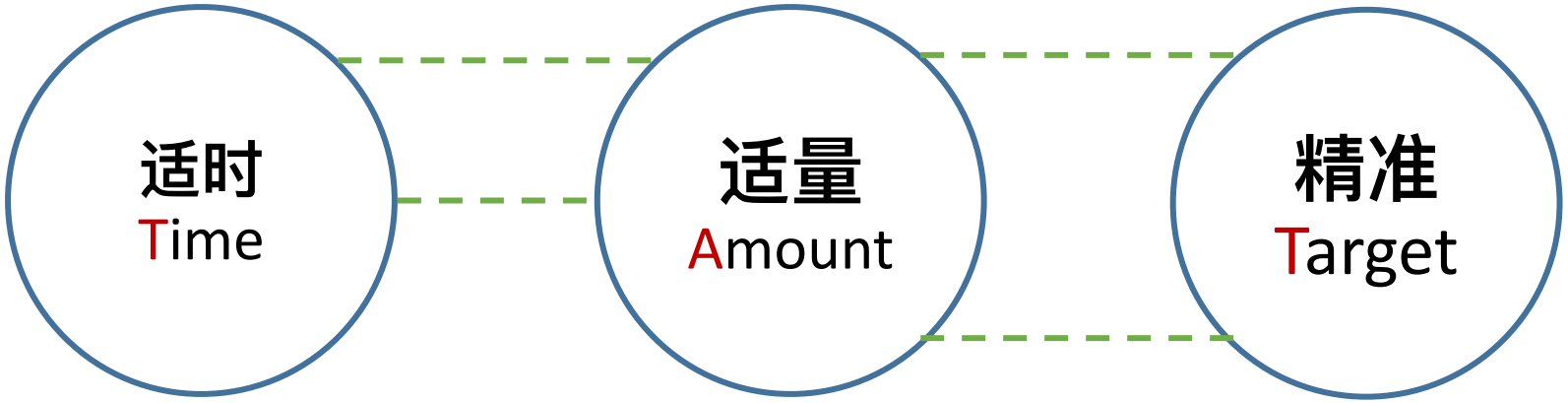
createDate string<date>
The date that the user was created.

你的测试覆盖率是 100% 么

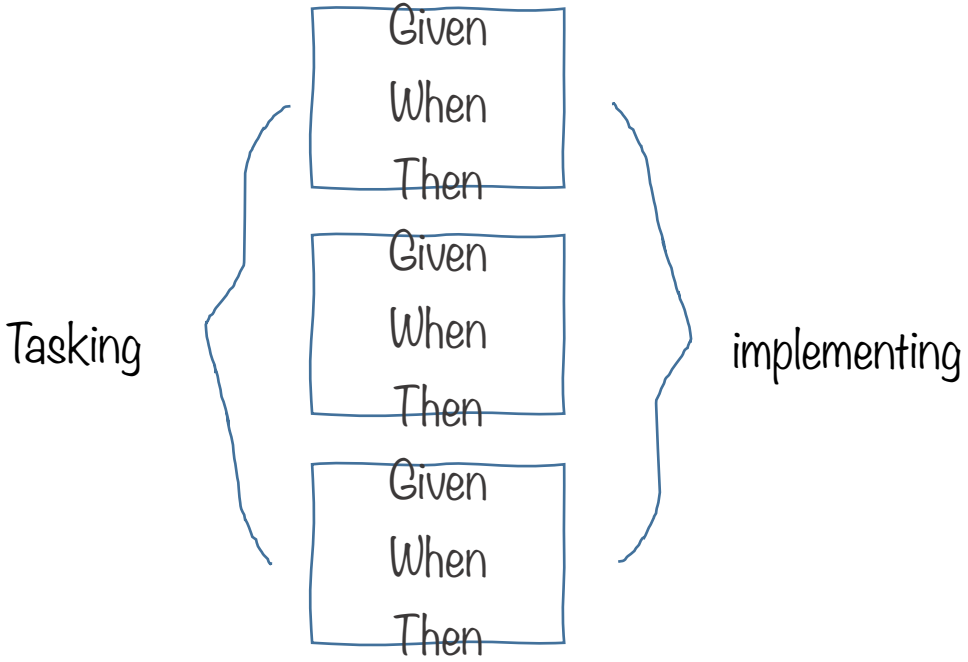
测试金字塔



精益测试



测试左移与测试驱动



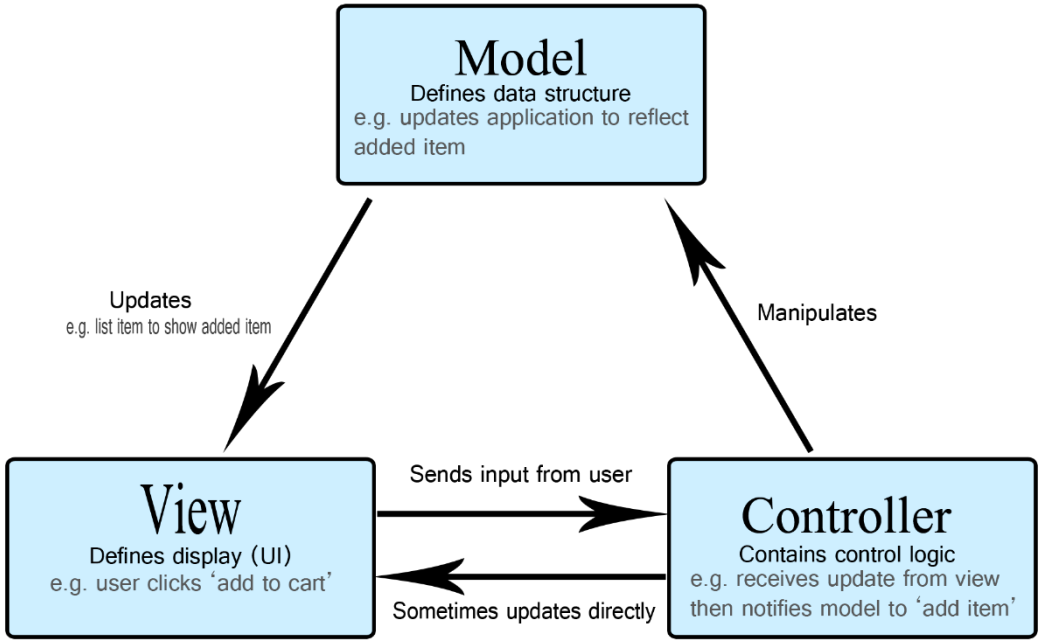
开发阶段

测试金字塔

目录

什么是好的架构

传统的 MVC 架构

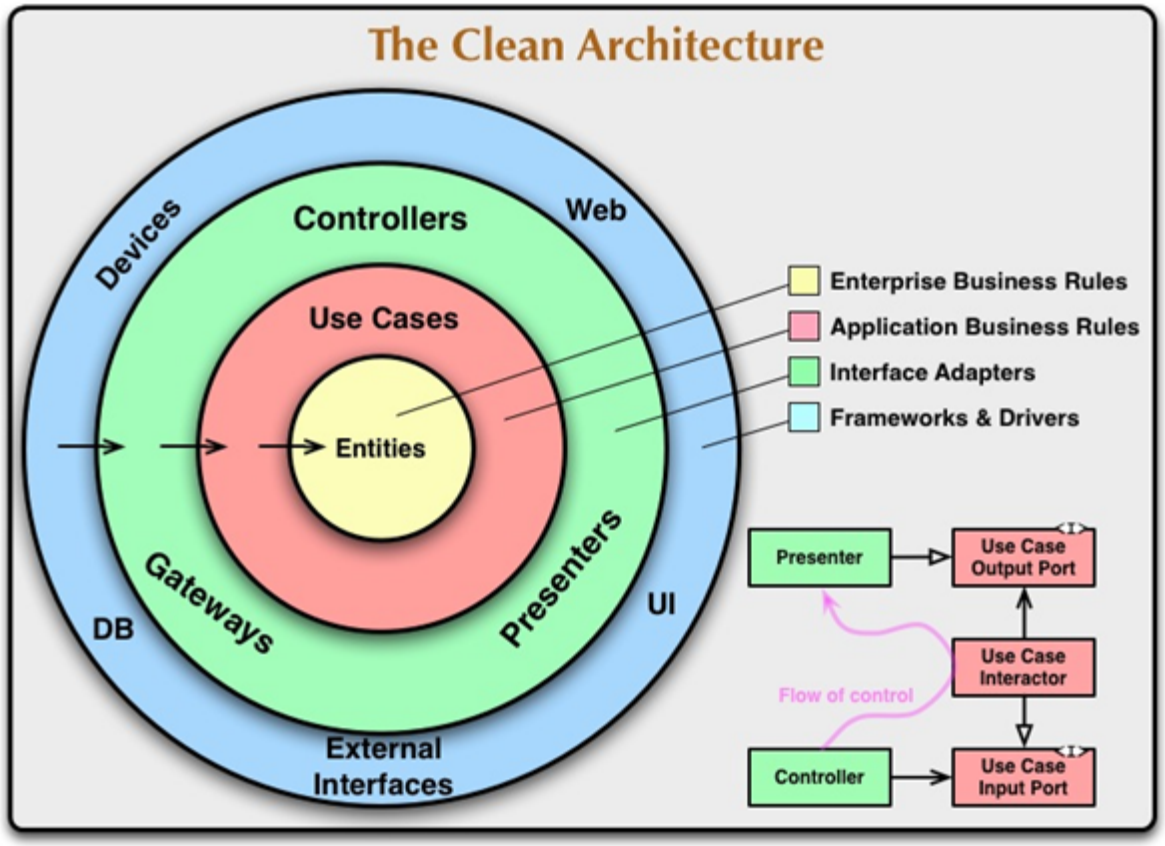


分层简单，复杂系统维护困难

以代码职责为划分依据，不能更好的表达业务逻辑

容易造成系统调用关系复杂

基于领域驱动设计 (DDD) 的架构实践

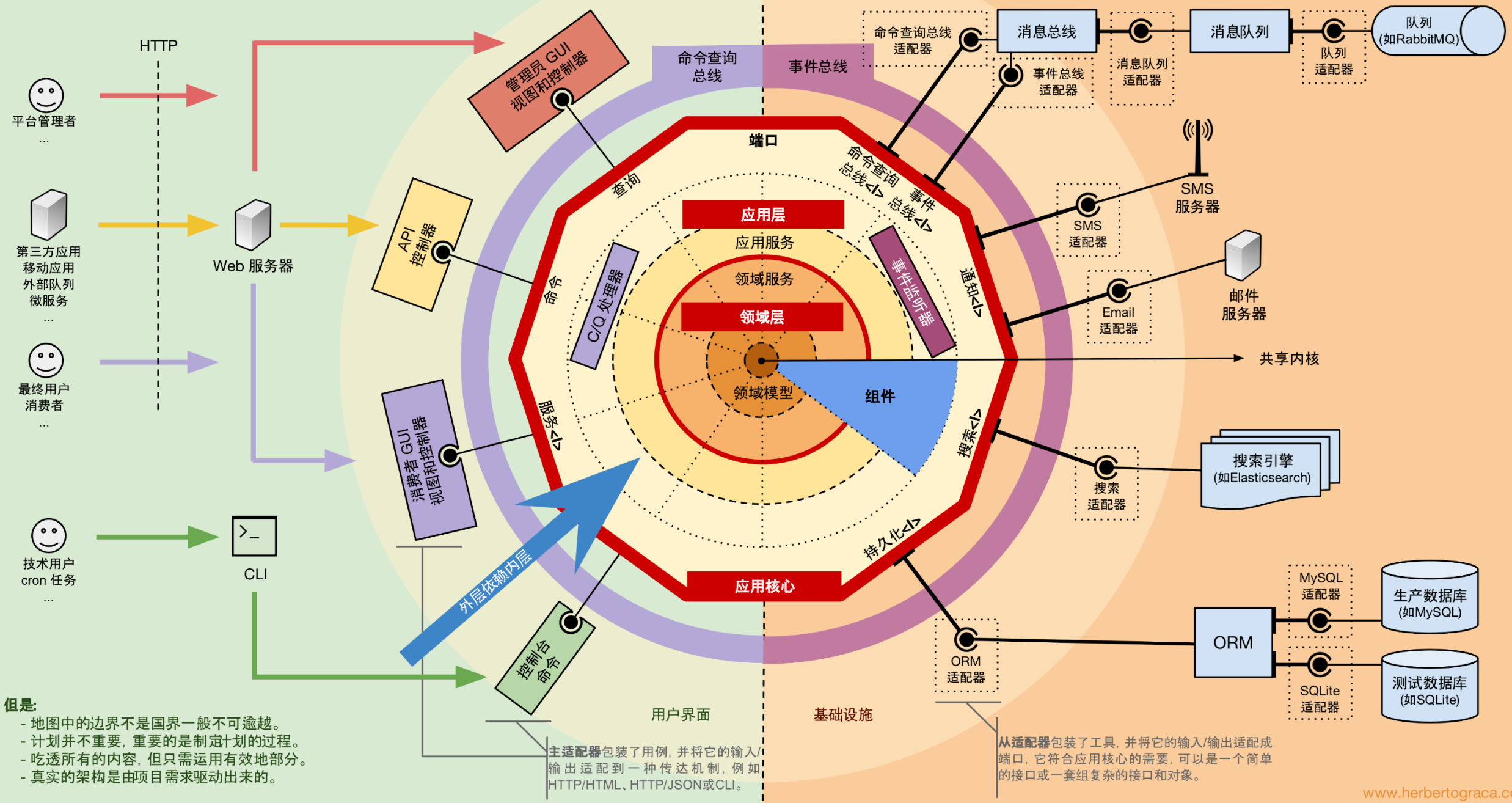


隔离变化，架构可以演进升级

更清晰的职责，更简单的依赖方向

基于业务进行上下文划分，更贴近业务方向

清晰架构



但是:

- 地图中的边界不是国界一般不可逾越。
- 计划并不重要，重要的是制定计划的过程。
- 吃透所有的内容，但只需运用有效地部分。
- 真实的架构是由项目需求驱动出来的。

主适配器包装了用例，并将它的输入/输出适配到一种传达机制，例如HTTP/HTML、HTTP/JSON或CLI。

从适配器包装了工具，并将它的输入/输出适配成端口，它符合应用核心的需要，可以是一个简单的接口或一套组复杂的接口和对象。

目录

总结

研发效能

质量保障

快速迭代

团队成长

谢谢

李洋洋

liyangyang@authing.cn

北京蒸汽记忆科技有限公司