

# Serverless & API

by @朴灵

# About me

- 阿里云 / Alinode team
- 《深入浅出 Node.js》作者

# Agenda

- Serverless demo
- Serverless 概念
  - API Gateway & Function Computing

**Hello world!**

# 函数即服务

```
hello.js
1 'use strict';
2
3 const Response = require('./helper').Response;
4
5 exports.index = function (event, context, callback) {
6   var res = new Response(callback);
7   res.send('hello Jackson Tian!\n');
8 };
9
```

# 部署函数

```
faas.yml
/
8 function-compute:
9   region: 'cn-shanghai'
10  services:
11    - name: 'fc'
12      description: 'fc test'
13      functions:
14        - name: 'helloworld'
15          description: 'hello!'
16          handler: hello.index
17          code:
18            - 'hello.js'
19            - 'helper.js'
```

# 配置路由

```
faas.yml
20
21 api-gateway:
22   endpoint: 'http://apigateway.cn-hangzhou.aliyuncs.com'
23   groups:
24     - name: 'apigw_fc'
25       description: 'API Gateway & Function Compute'
26       apis:
27         - name: 'helloworld'
28           method: 'GET'
29           path: '/helloworld'
30           function: 'cn-shanghai/fc/helloworld'
31
```

# 函数即服务

- 发布

```
$ ~/gist/lambda/bin/publish.js  
URL: GET http://a45b47945d0d4feab875bcbbe2031c0f-cn-  
hangzhou.alicloudapi.com/helloworld => cn-shanghai/fc/  
helloworld
```

- 服务 = 发布(函数, 配置文件)



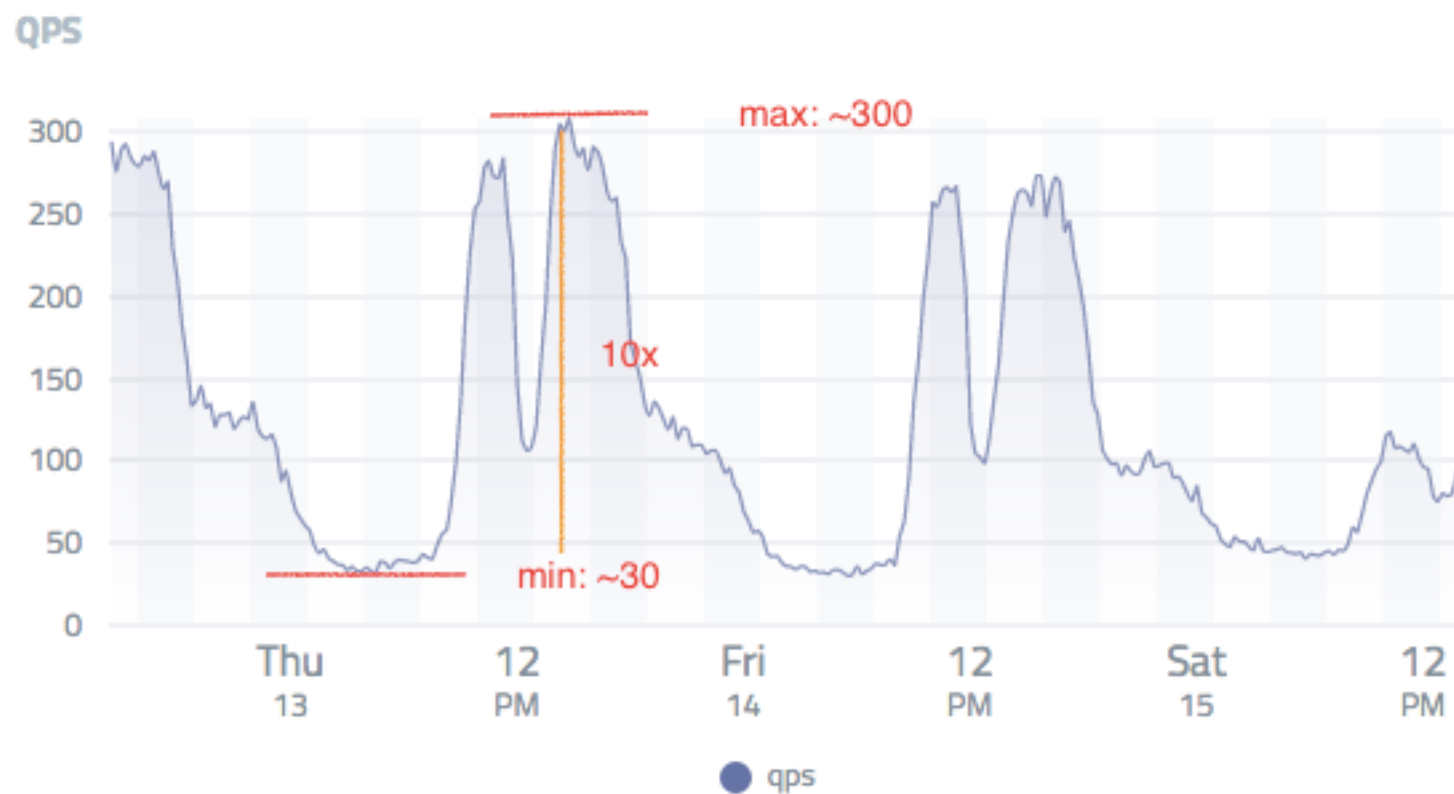
# 软件开发的复杂度

- 开发：IDE、编程语言、编译器、调试、...
- 测试：...
- 部署：依赖管理、自检、回滚
- 安全：鉴权、DDos、重放攻击、...
- 运维：系统环境、负载均衡、扩容、缩容、监控、报警

# 软件的考量问题

- 可运维性
- 可扩展性
- 安全性
- 稳定性
- 可靠性
- 高可用性

# 关于伸缩性问题



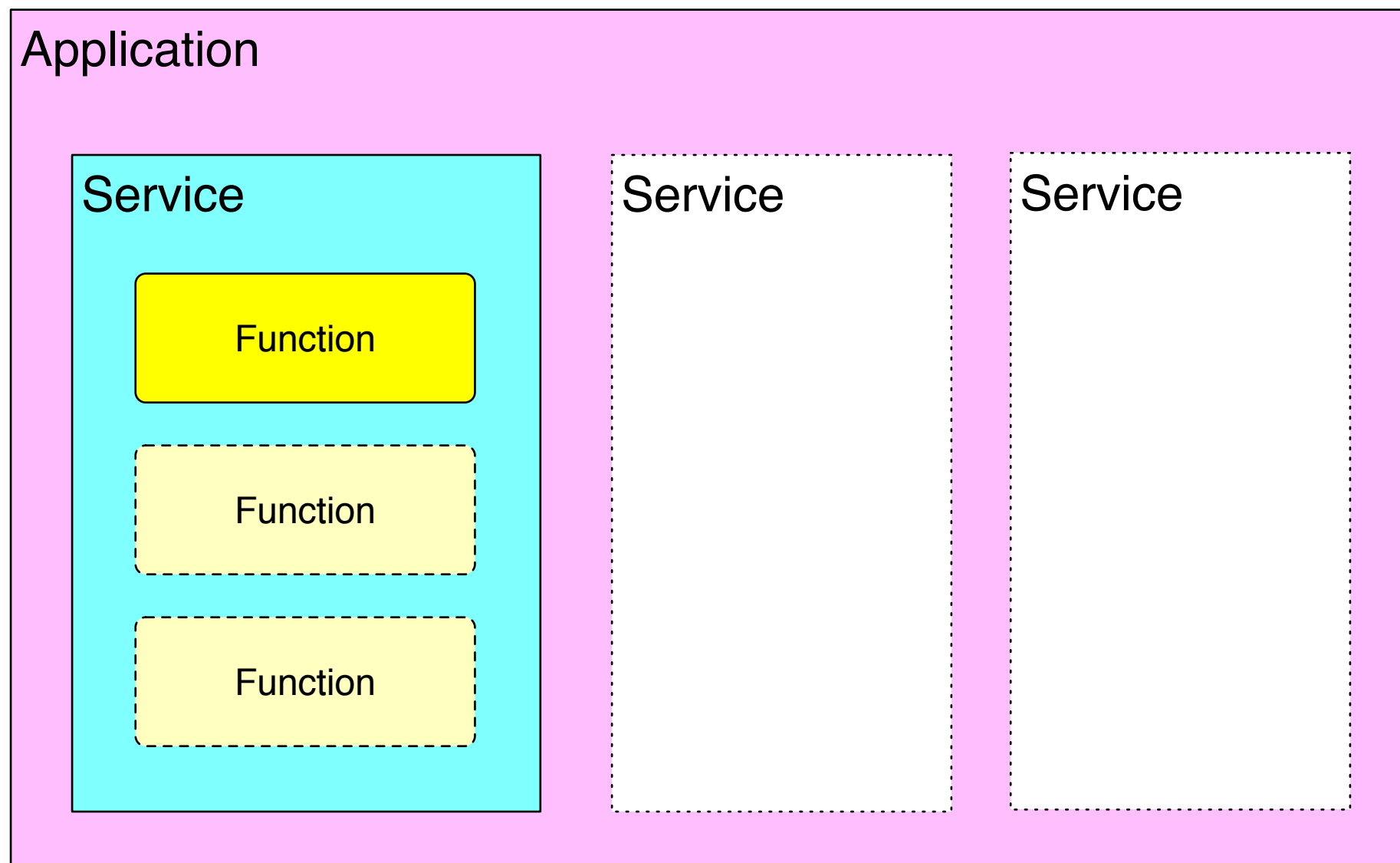
# XaaS 比较



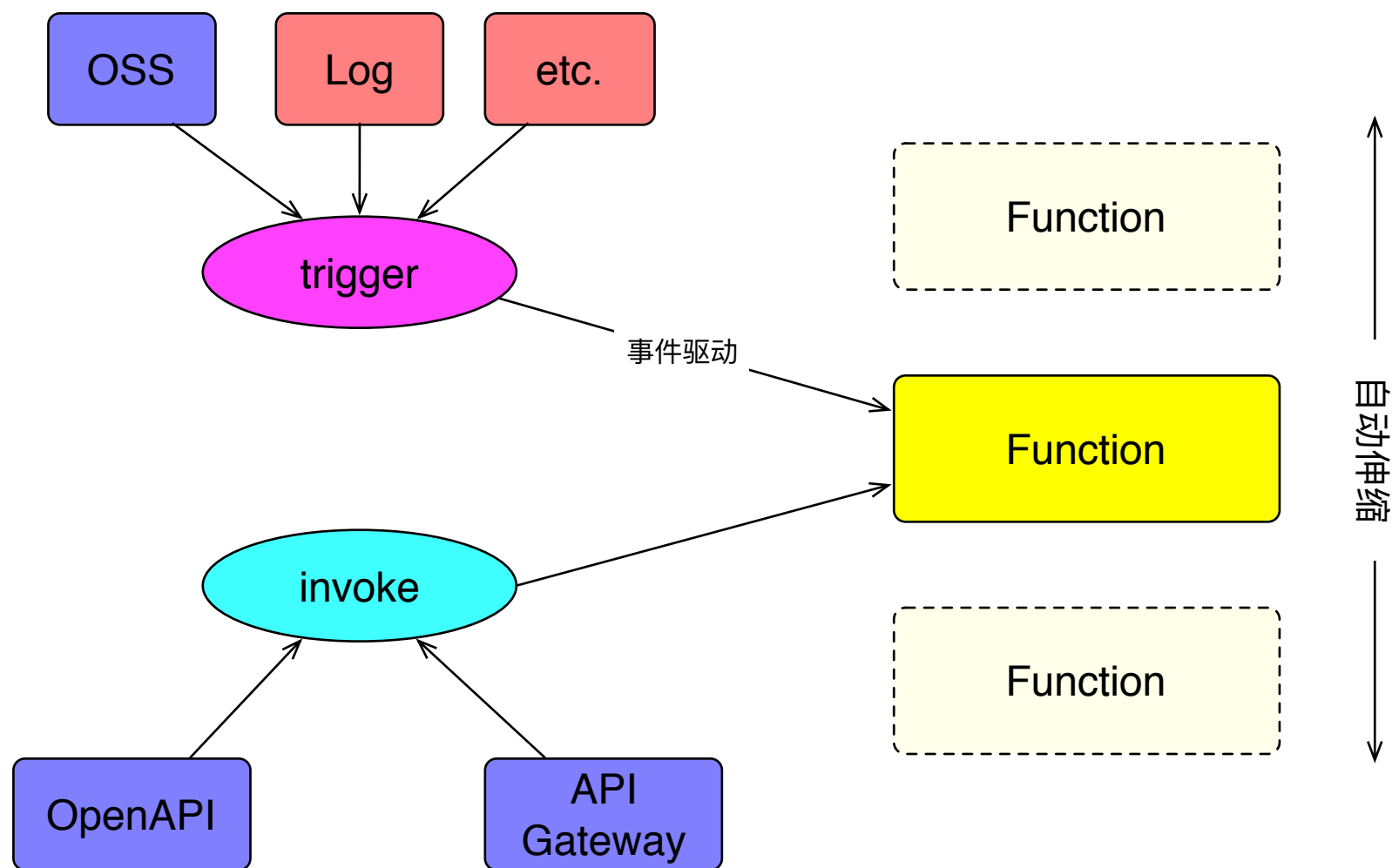
# XaaS 比较

维度/服务类型	IaaS	CaaS	PaaS	FaaS
开发效率	低	中	中	高
可伸缩性	低	中	中	高
可运维性	低	中	高	高
成本	高	高	高	低

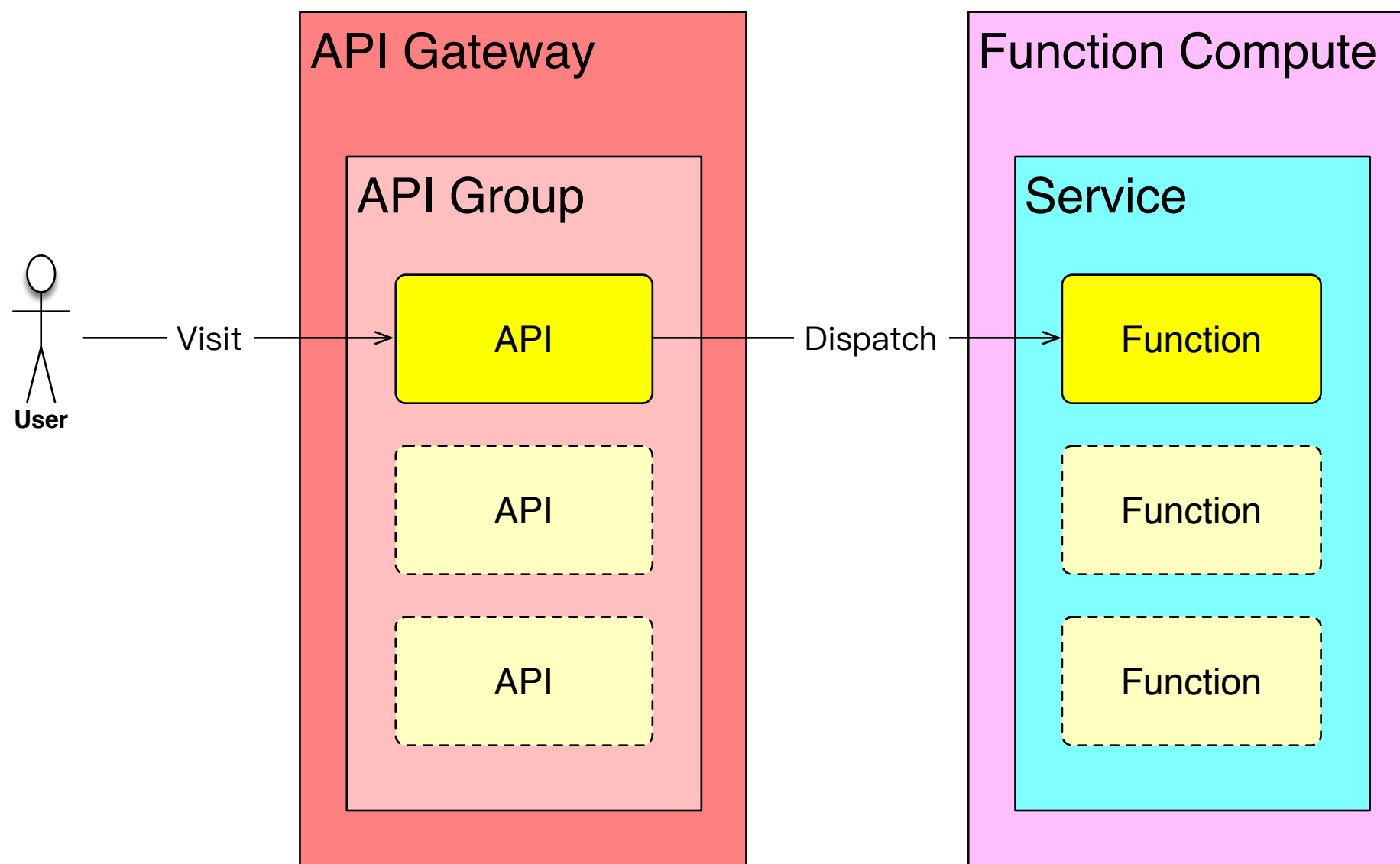
# 函数计算的应用架构



# 函数的执行方式



# 天作之合

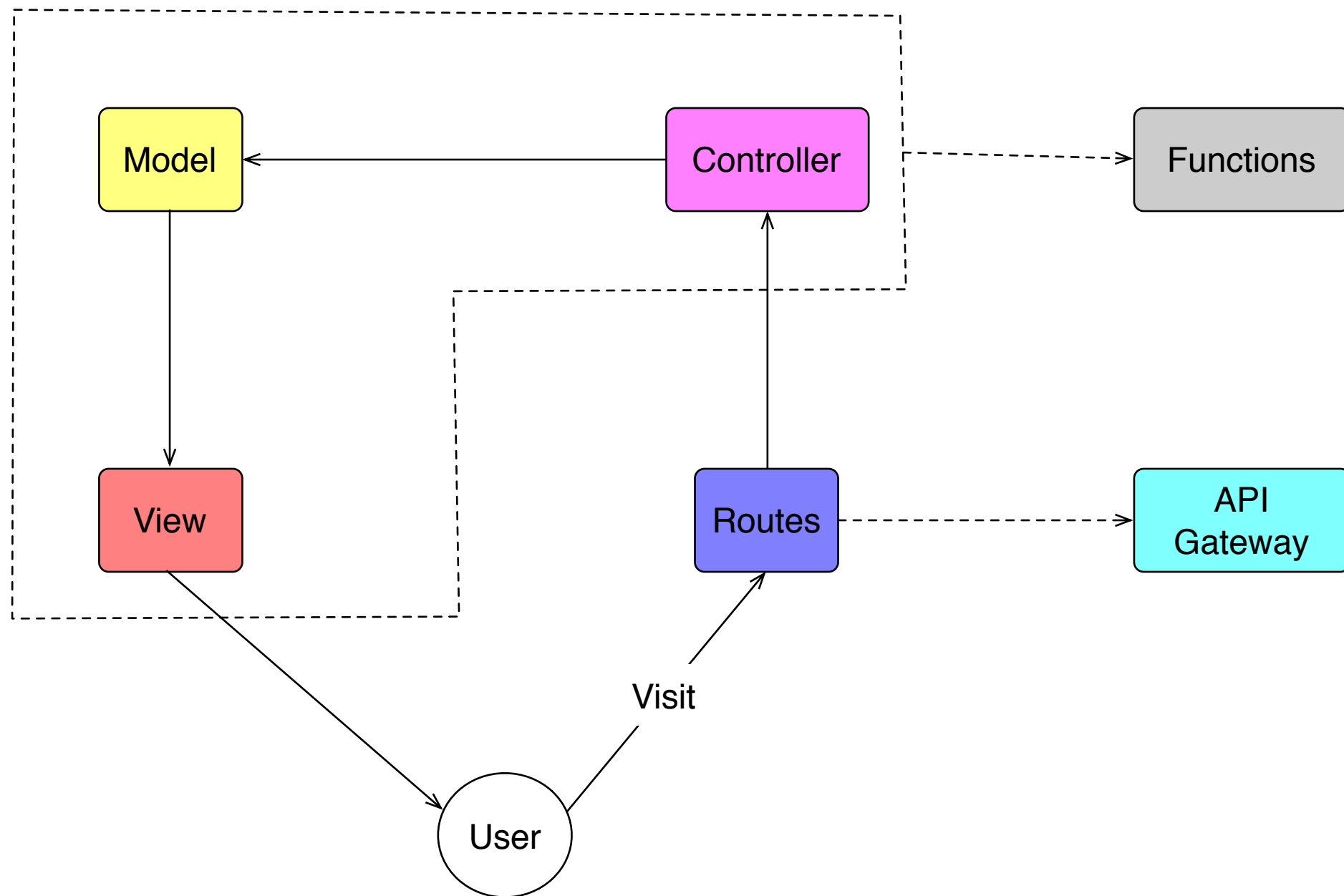




# API Gateway

- 防攻击、防重放、请求加密、身份认证、权限管理、流量控制
- API 定义、测试、发布、下线生命周期管理
- 监控、报警、分析、API市场

# 新的应用形态



# FaaS 的缺陷

- 运行环境的不确定性
  - IP变化
- 运行环境的数量，对依赖资源的压力
  - 比如：数据库的连接数限制

“谢谢。”

–*Jackson Tian*