

# Serverless 前端落地与实践

王俊杰

腾讯 Serverless 技术专家

# InfoQ官网 全新改版上线

促进软件开发领域知识与创新的传播



关注InfoQ网站  
第一时间浏览原创IT新闻资讯



免费下载迷你书  
阅读一线开发者的技术干货

# 王俊杰

Jenswang



腾讯 Serverless 前端技术专家。

负责腾讯云函数与大前端研发结合方案设计，负责 SCF 云函数编排、

Serverless 日志、监控、排障等相关 Topic。

同时担任腾讯云 Serverless 技术推动者，推动 Serverless 技术在行业大前端研发架构中的落地和实践。

曾担任百度搜索前端技术经理，负责百度搜索产品前端研发技术管理工作。



# 目录

前端与 Serverless 的不解之缘

Serverless 前端工程化的基本思路

Serverless Framework 原理与实现

Serverless Now

# 目录

## 前端与 Serverless 的不解之缘

Serverless 前端工程化的基本思路

Serverless Framework 原理与实现

Serverless Now

# What is Serverless ?

## Serverless = FaaS + BaaS ?

### What is FaaS ?

## Cloud Programming Simplified: A Berkeley View on Serverless Computing



*Eric Jonas  
Johann Schleier-Smith  
Vikram Sreekanti  
Chia-Che Tsai  
Anurag Khandelwal  
Qifan Pu  
Vaishaal Shankar  
Joao Menezes Carreira  
Karl Krauth  
Neeraja Yadwadkar  
Joseph Gonzalez  
Raluca Ada Popa  
Ion Stoica  
David A. Patterson*

Electrical Engineering and Computer Sciences  
University of California at Berkeley

Technical Report No. UCB/EECS-2019-3  
<http://www2.eecs.berkeley.edu/Pubs/TechRpts/2019/EECS-2019-3.html>

February 10, 2019

Cloud Programming Simplified: A Berkeley View on Serverless Computing

<http://www2.eecs.berkeley.edu/Pubs/TechRpts/2019/EECS-2019-3.pdf>

While cloud functions—packaged as FaaS (Function as a Service) offerings—represent the core of serverless computing, cloud platforms also provide specialized serverless frameworks that cater to specific application requirements as BaaS (Backend as a Service) offerings.

Put simply, serverless **computing** = FaaS + BaaS.

# Cloud Programming Simplified: A Berkeley View on Serverless Computing

Eric Jonas      Johann Schleier-Smith      Vikram Sreekanti      Chia-Che Tsai  
Anurag Khandelwal      Qifan Pu      Vaishaal Shankar      Joao Carreira  
Karl Krauth      Neeraja Yadwadkar      Joseph E. Gonzalez      Raluca Ada Popa  
Ion Stoica      David A. Patterson

UC Berkeley

serverlessview@berkeley.edu

### Abstract

Serverless cloud computing handles virtually all the system administration operations needed to make it easier for programmers to use the cloud. It provides an interface that greatly simplifies cloud programming, and represents an evolution that parallels the transition from assembly language to high-level programming languages. This paper gives a quick history of cloud computing, including an accounting of the predictions of the 2009 Berkeley View of Cloud Computing paper, explains the motivation for serverless computing, describes applications that stretch the current limits of serverless, and then lists obstacles and research opportunities required for serverless computing to fulfill its full potential. Just as the 2009 paper identified challenges for the cloud and predicted they would be addressed and that cloud use would accelerate, we predict these issues are solvable and that serverless computing will grow to dominate the future of cloud computing.



Serverless cloud **computing** handles virtually all the system administration operations needed to make it easier for programmers to use the cloud. It provides an interface that greatly simplifies cloud programming, and represents an evolution that parallels the transition from assembly language to high-level programming languages.

### Contents

- 1 Introduction to Serverless Computing 3
- 2 Emergence of Serverless Computing 5
  - 2.1 Contextualizing Serverless Computing . . . . . 6
  - 2.2 Attractiveness of Serverless Computing . . . . . 8
- 3 Limitations of Today’s Serverless Computing Platforms 9
  - 3.1 Inadequate storage for fine-grained operations . . . . . 12
  - 3.2 Lack of fine-grained coordination . . . . . 12
  - 3.3 Poor performance for standard communication patterns . . . . . 13
  - 3.4 Predictable Performance . . . . . 14



算力 算法 数据

**无服务器云计算（Serverless Computing）几乎封装了所有的底层资源管理和系统运维工作，使开发人员更容易使用云基础设施。**

**它提供了一个方式，极大地简化了基于云服务的编程，犹如汇编语言到高级编程语言般的转换。**



Web Developer



Front-end Developer

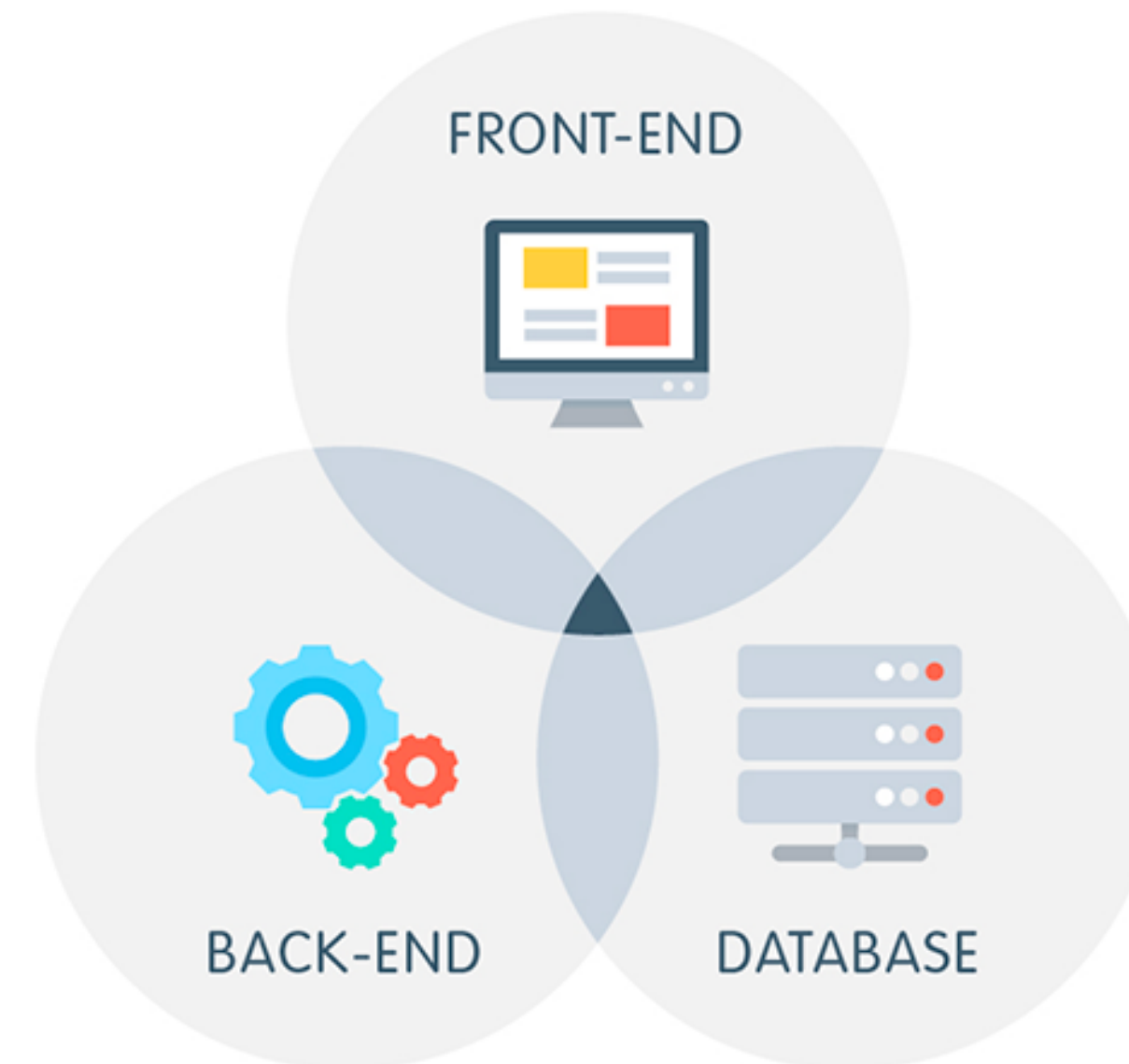


Full stack Developer



Interface discussion/integration commissioning/ API Dependence

## FULL-STACK DEVELOPMENT

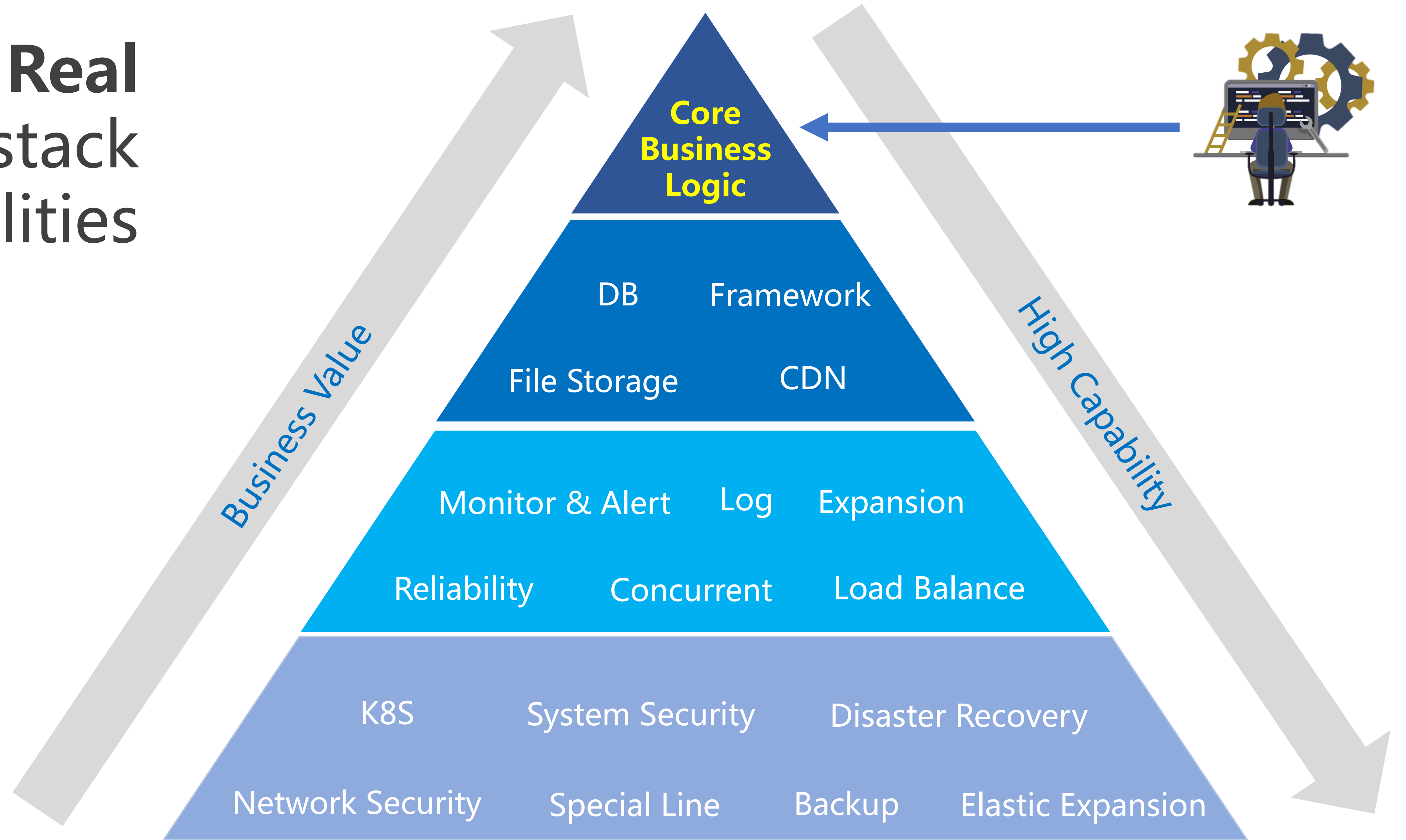


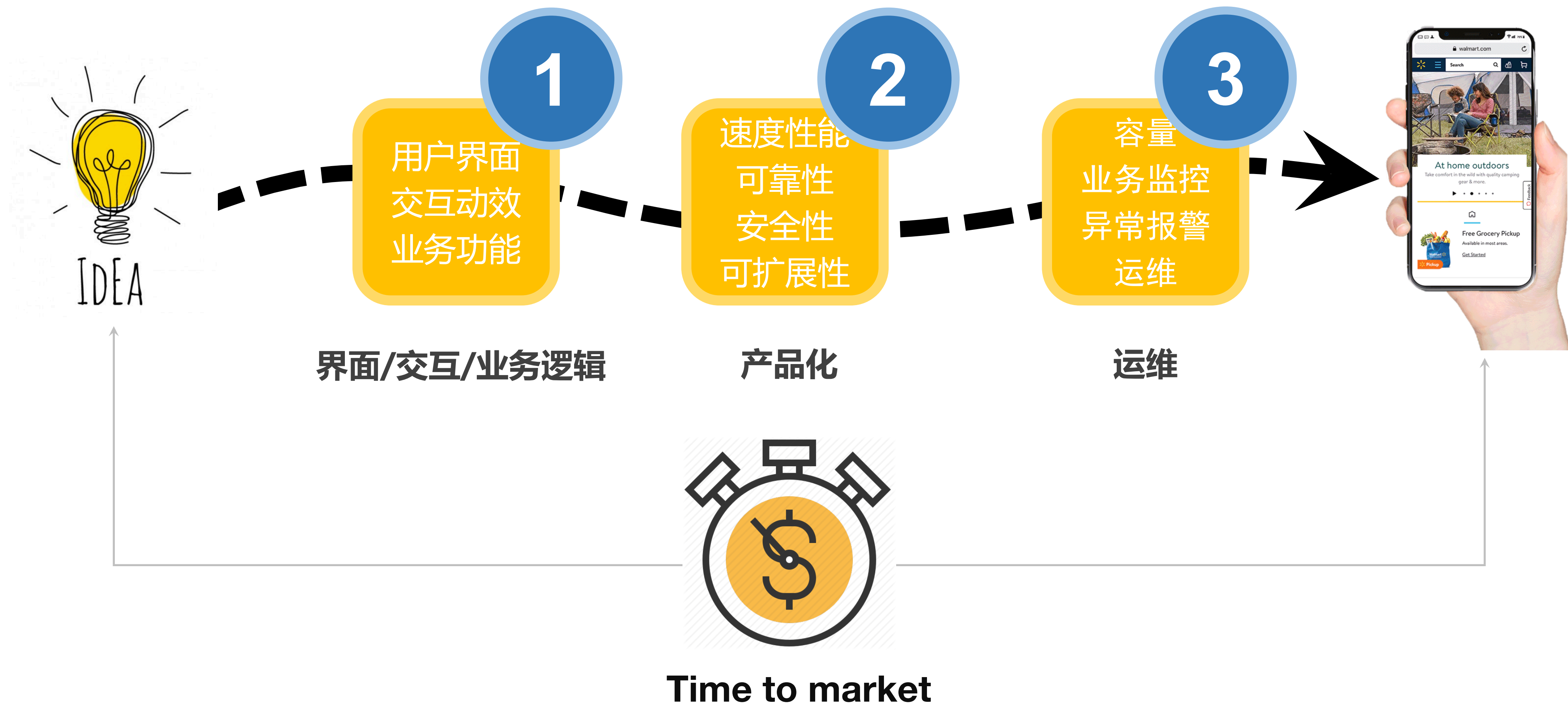
Enhancing the business efficiency.

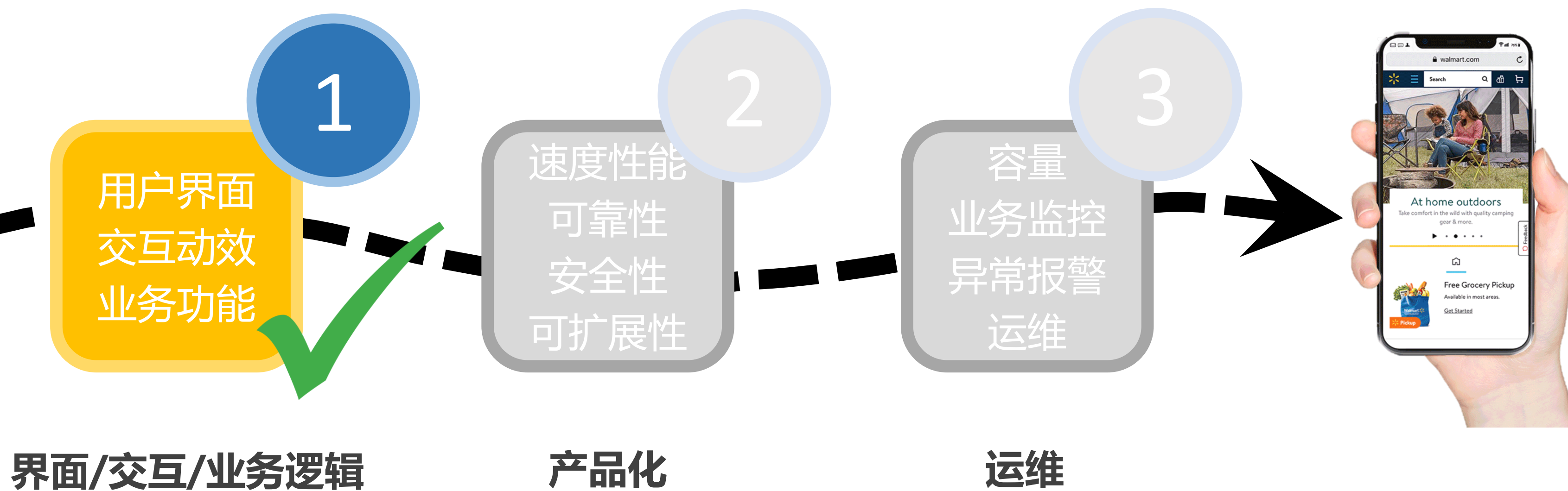


Full-stack, in eyes of front end developers

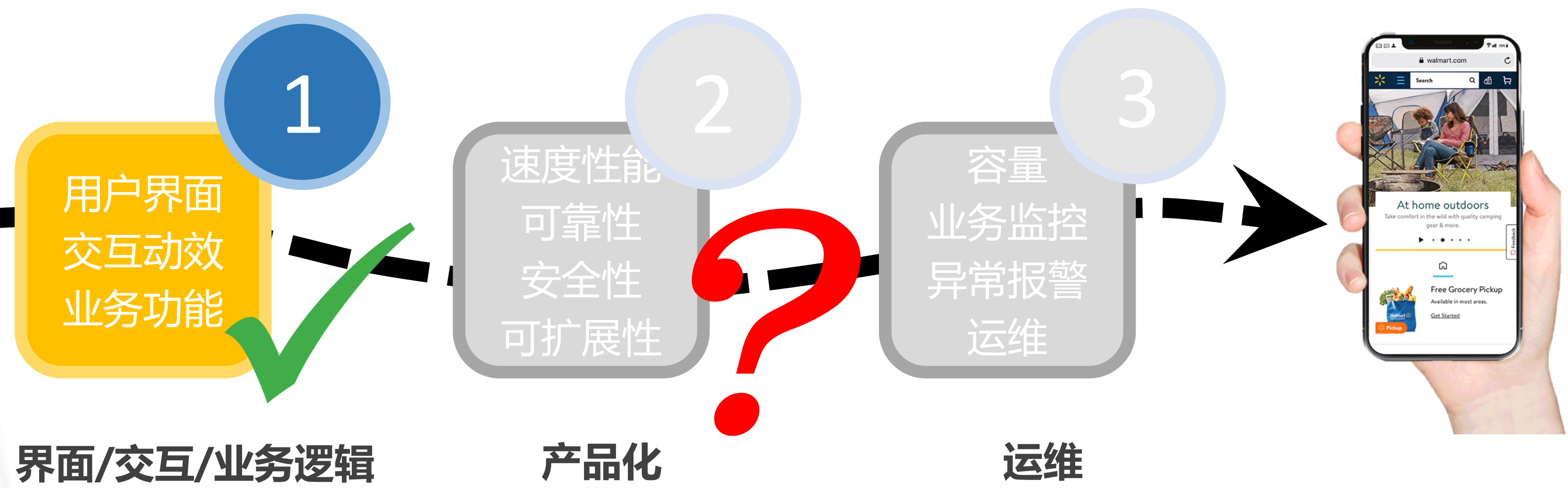
# Real Full-stack Capabilities











ALERTING MAINTENANCES SCALING MONITORING  
MONITORING ALERTING MAINTENANCES SCALING  
SCALING MONITORING ALERTING MAINTENANC  
E

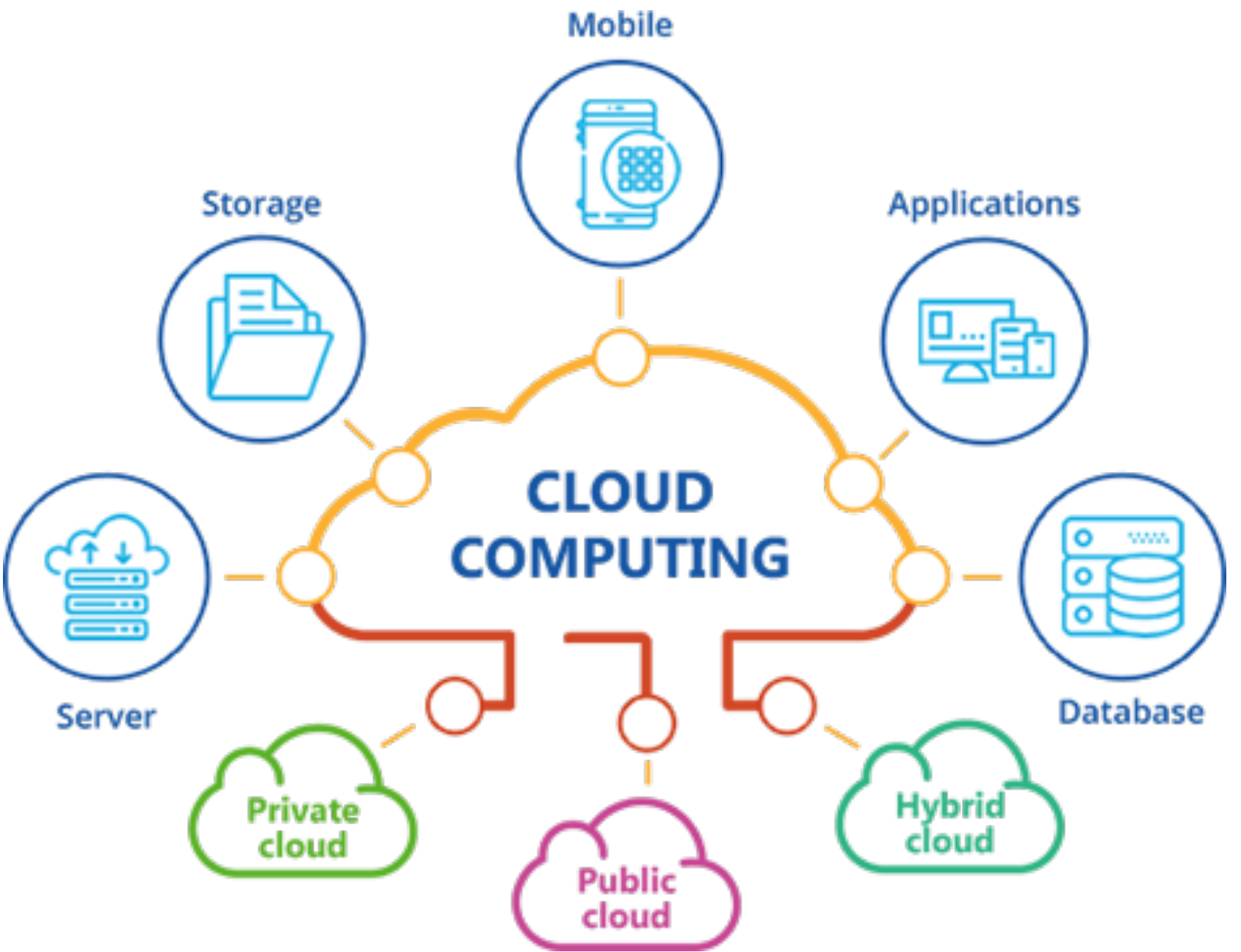
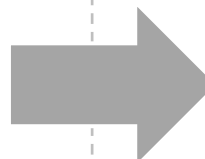
MAINTENANCES SCALING MONITORING ALERTING

```

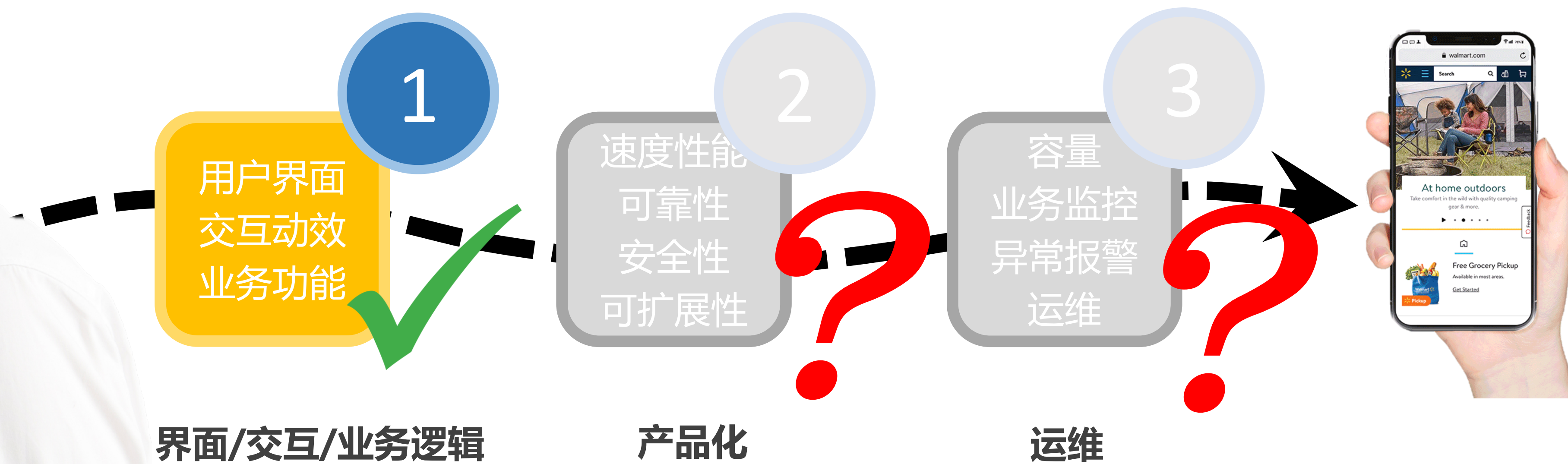
<div class="container">
  <div class="row">
    <div class="col-md-6 col-lg-8"> <!-- BEGIN NAVIGATION
    <nav id="nav" role="navigation">
      <ul>
        <li><a href="index.html">Home</a></li>
        <li><a href="home-events.html">Home Events</a></li>
        <li><a href="multiple-column-menu.html">Multiple Column Men
        <li><a href="tall-button-header.html">Tall But
        <li class="active"><a href="tall-logo.html">Ta
      </ul>
    </div>
    <div class="has-children"> <a href="#">Carousels</a>
  </div>
  <div class="variable-width-slider.html">Variat

```

Hundreds of lines of code



 Amazon.com: Managing Ku... amazon.com	 Amazon.com: Cloud Nativ... amazon.com	 The Kubernetes Book: Ama... amazon.co.uk	 Amazon.com: Kubernetes... amazon.com	 Programming Kubernetes: ... amazon.ca	 DevOps with Kubernetes - S... packtpub.com	 Amazon.com: Kubernetes o... amazon.com	 Kuberne... thenews
 cloud computing in 20... quora.com	 Cloud Native DevOps wit... oreilly.com	 Pro DevOps with Googl... amazon.com	 Dr. GP Pulipaka on Twit... twitter.com	 98 Best Cloud Computi... bookauthority.org	 Cloud Computing : Dan C ... blackwells.co.uk	 DevOps with Kubernetes packtpub.com	 Cloud Computing Mad... cb-india.com



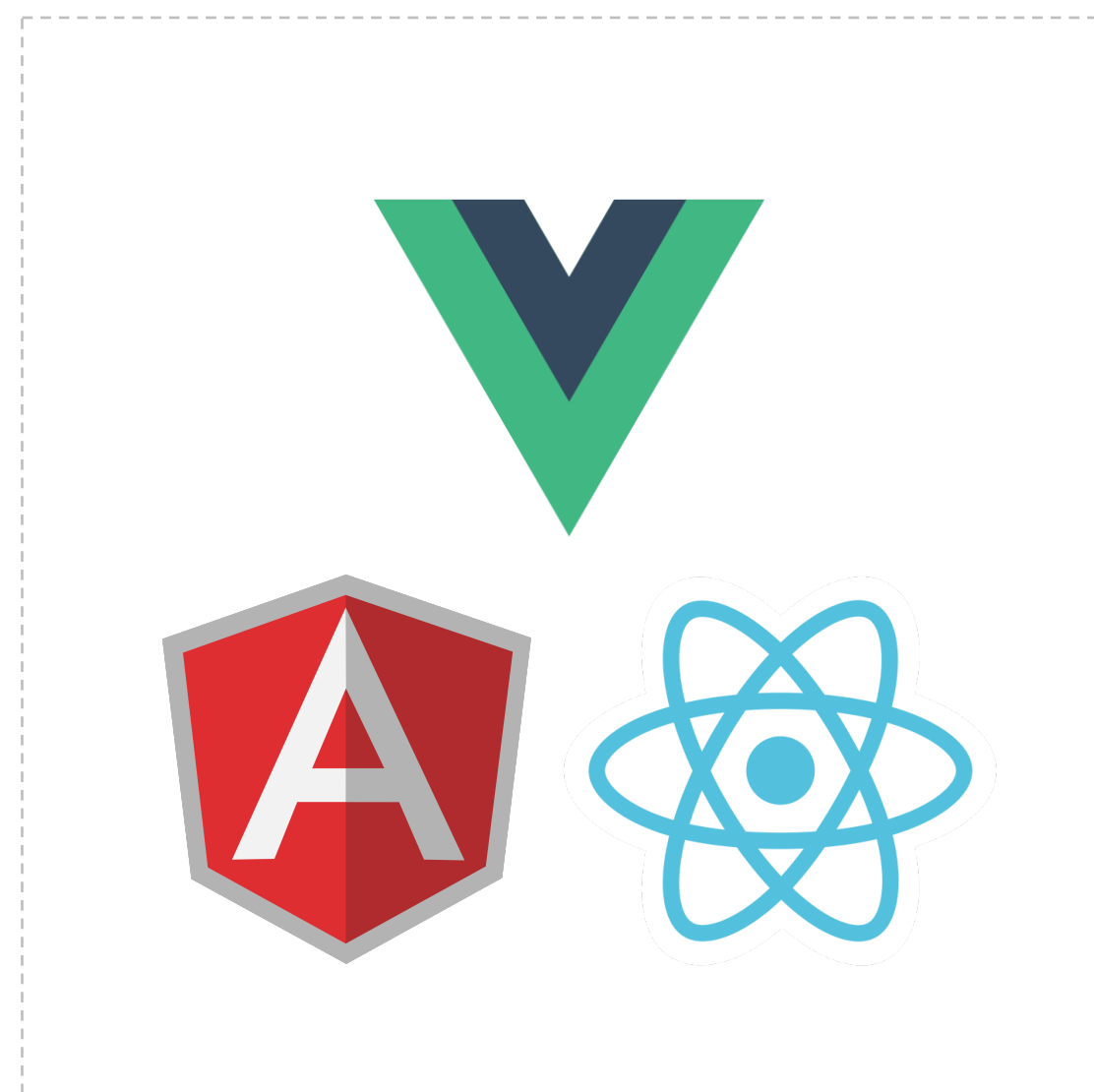
# 目录

前端与 Serverless 的不解之缘

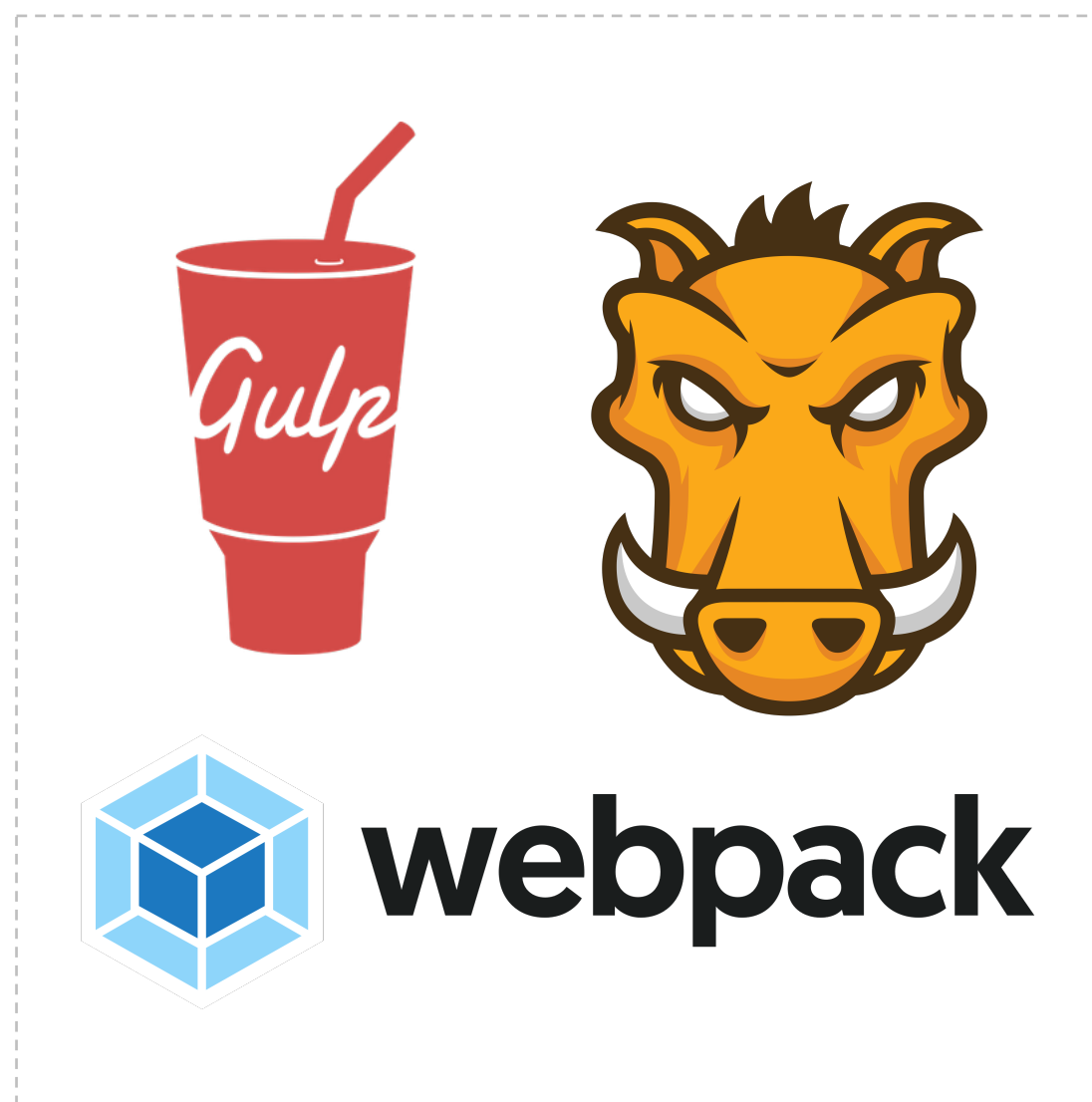
**Serverless 前端工程化的基本思路**

Serverless Framework 原理与实现

Serverless Now



组件化



工程化



Serverless

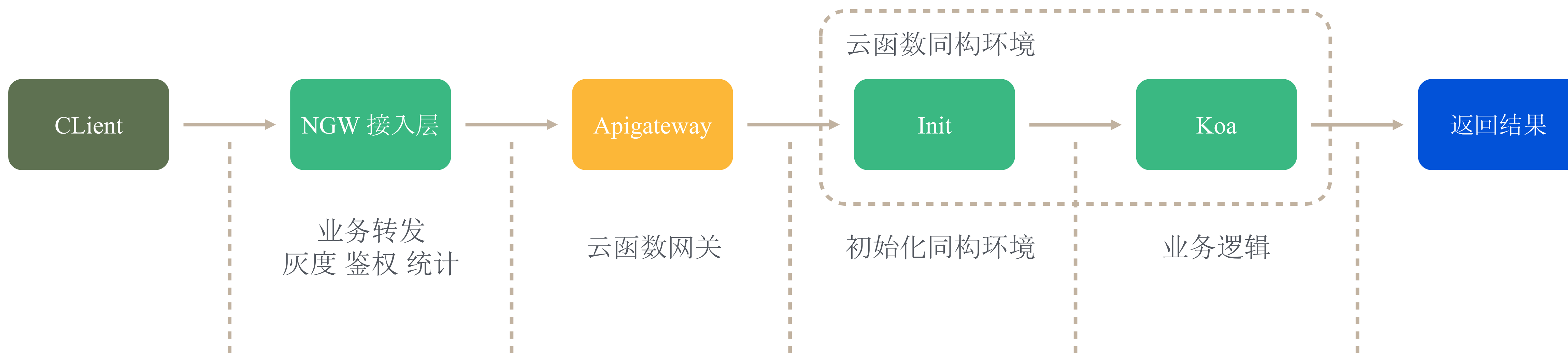
**Framework or not framework ?**

**That's a question !**

# IVWEB 团队在 Serverless 的业务落地



## Serverless 直出渲染 (Serverless Render)



接入产品: 手机QQ附近、NOW直播、花样直播、QQ群

腾讯云 SCF + NGW

免运维 全量日志 多维度数据统计 云 API 打通 workflow

业务上线 & 维护工作量 降低 80%



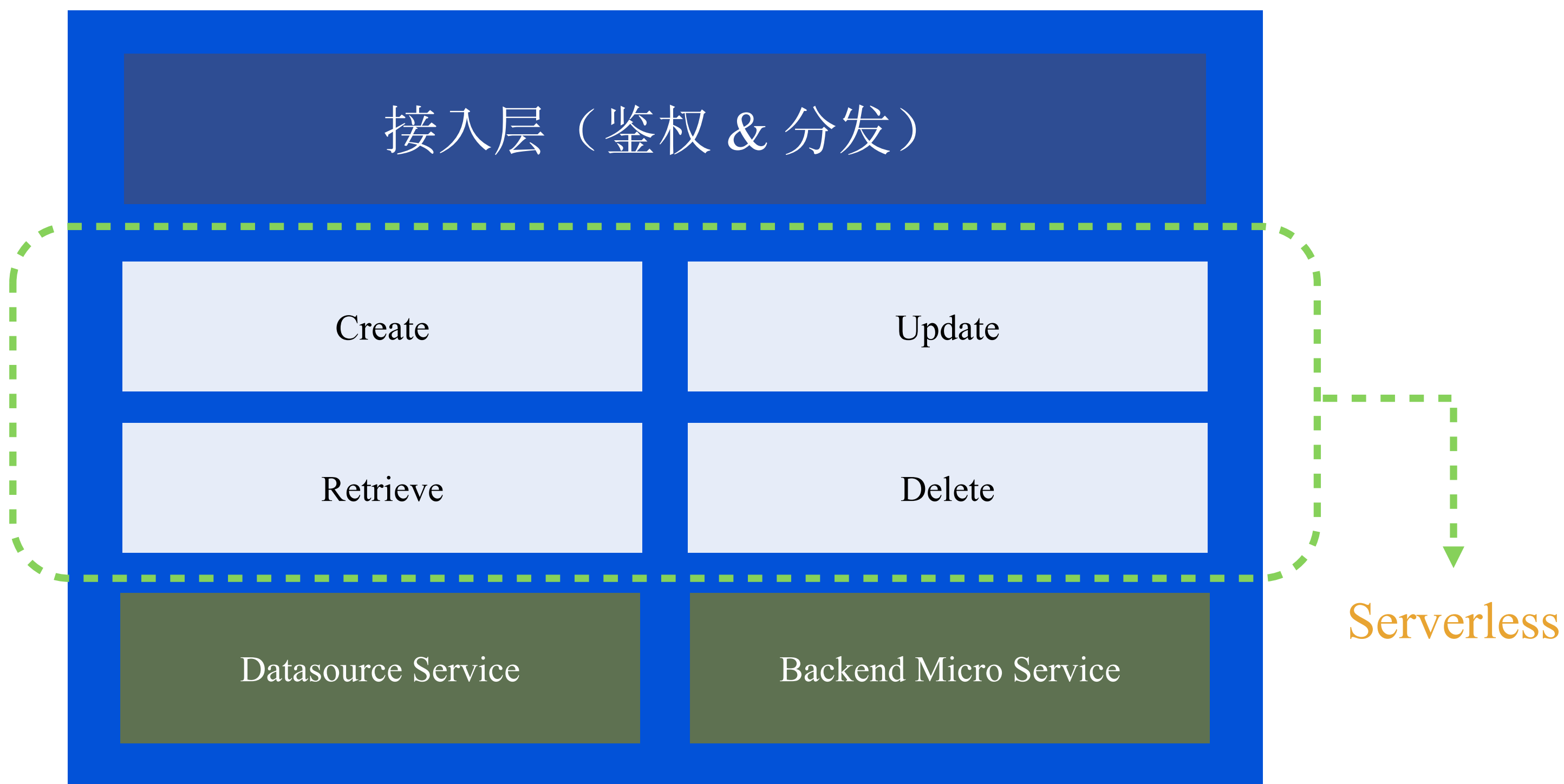


# IVWEB 团队在 Serverless 的业务落地



NOW 直播 B 侧运营平台

从 BFF 到 SFF

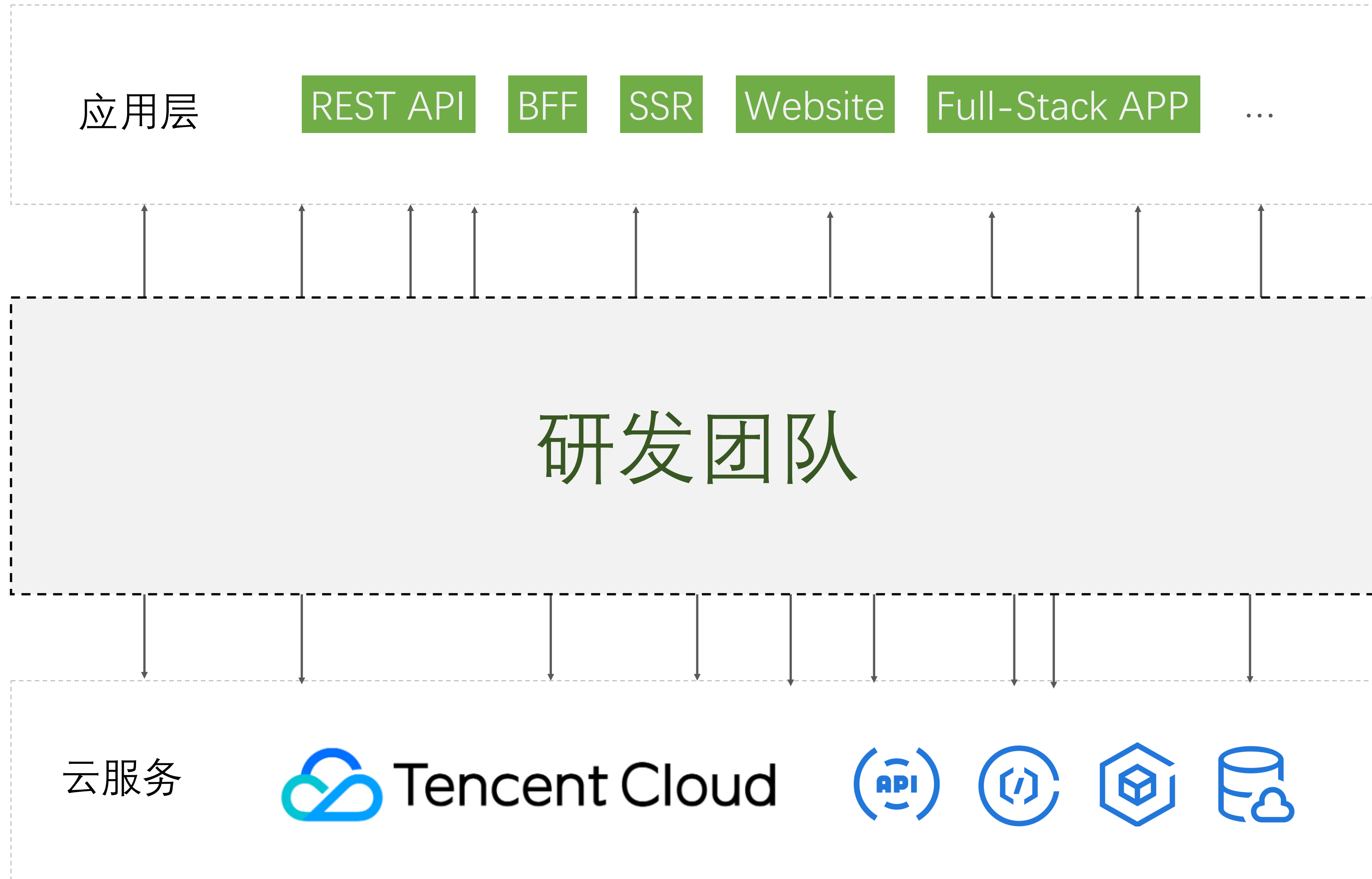


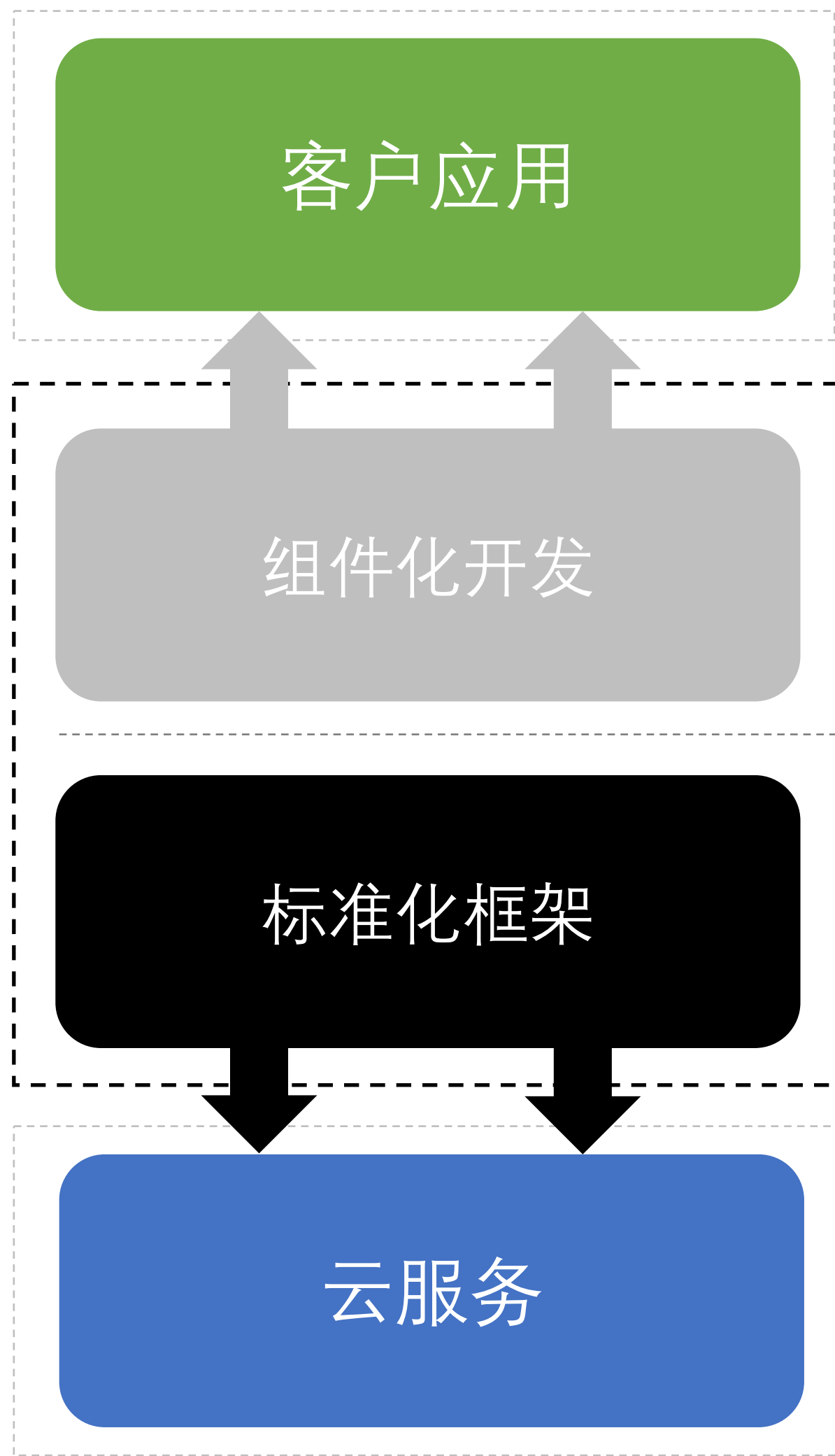
- 提高迭代效率
- 按需取用，用完即走
- 分离部署，安全可靠
- 细粒度的业务逻辑拆分

# Framework

代码重用 | 统一规范 | 降低门槛和难度 | 专注业务逻辑 | 社区优势 | 易于维护 | 提升效率

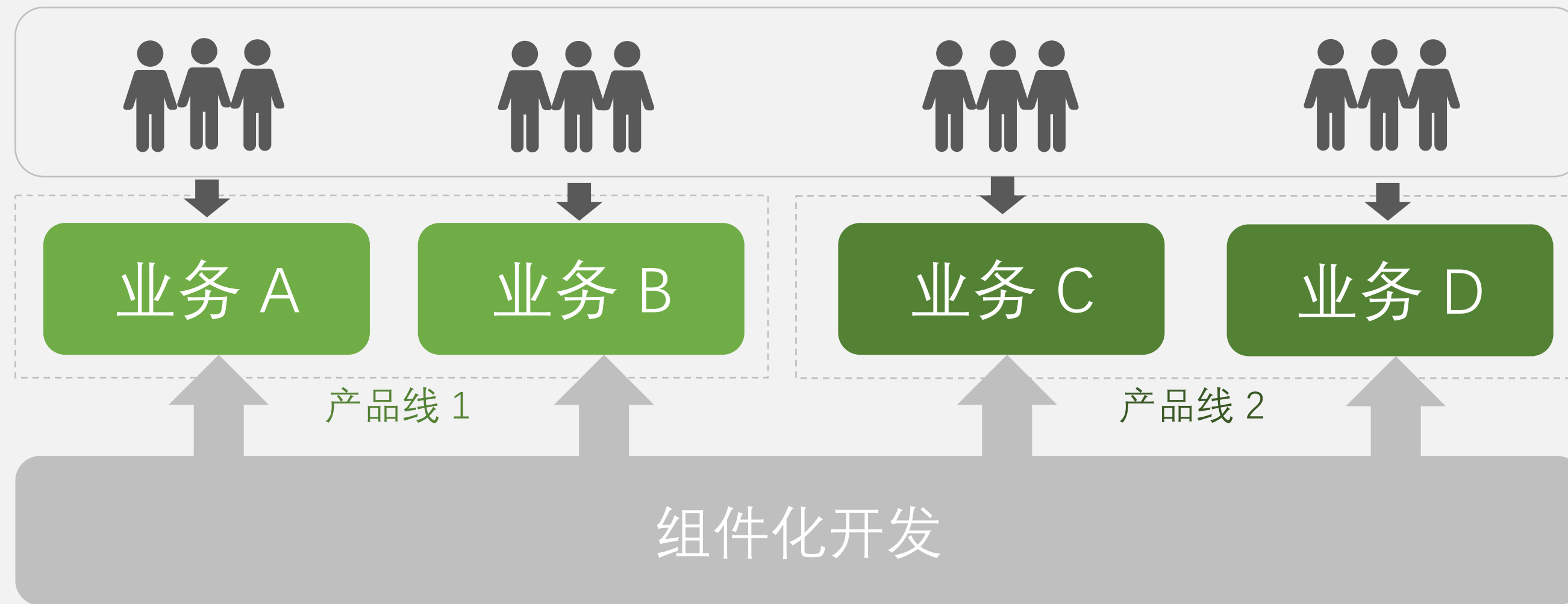
# Serverless的开发模式



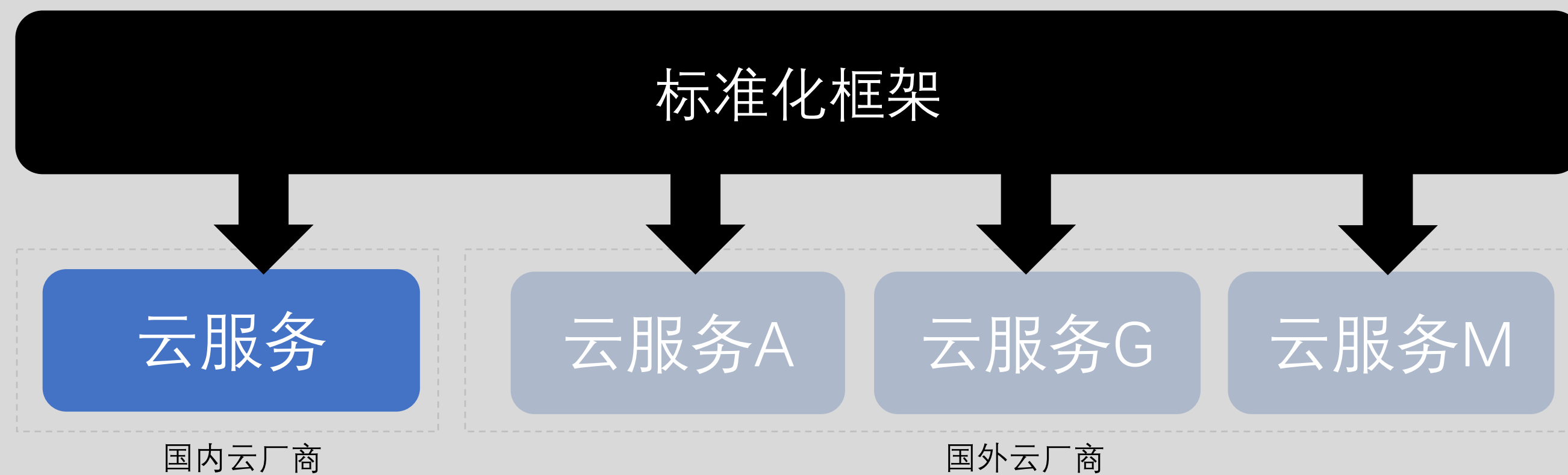


组件化开发

标准化框架



跨产品跨业务可组件复用，提升复用减少重复开发 / 人员技术栈要求相同，利于分工和招聘



框架完美支持异构的云厂商接口和底层，业务在多云间无缝平滑迁移部署

# 目录

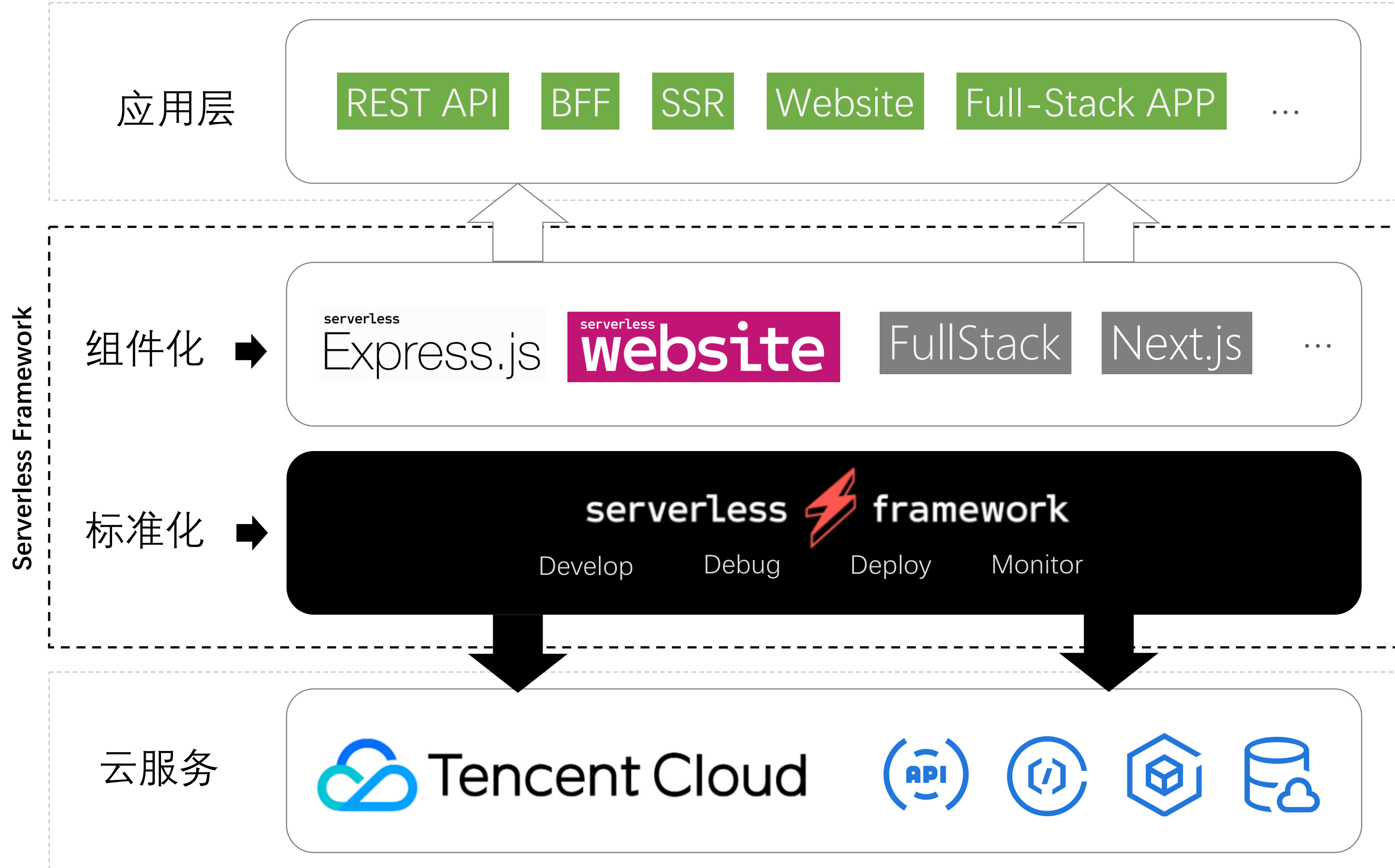
前端与 Serverless 的不解之缘

Serverless 前端工程化的基本思路

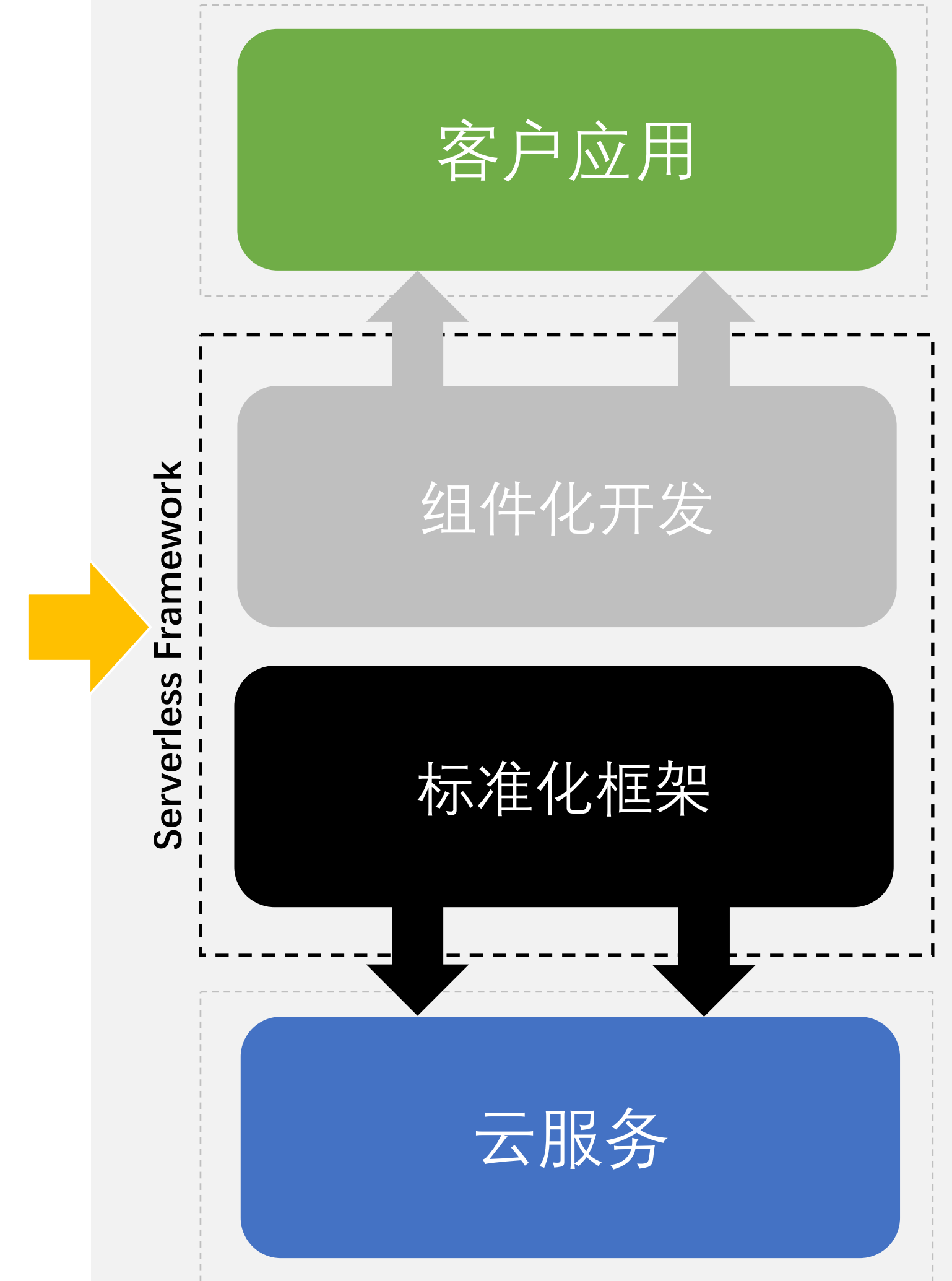
**Serverless Framework 原理与实现**

Serverless Now

Serverless Framework 是一个 **标准化、组件化** 的 Serverless 应用开发产品

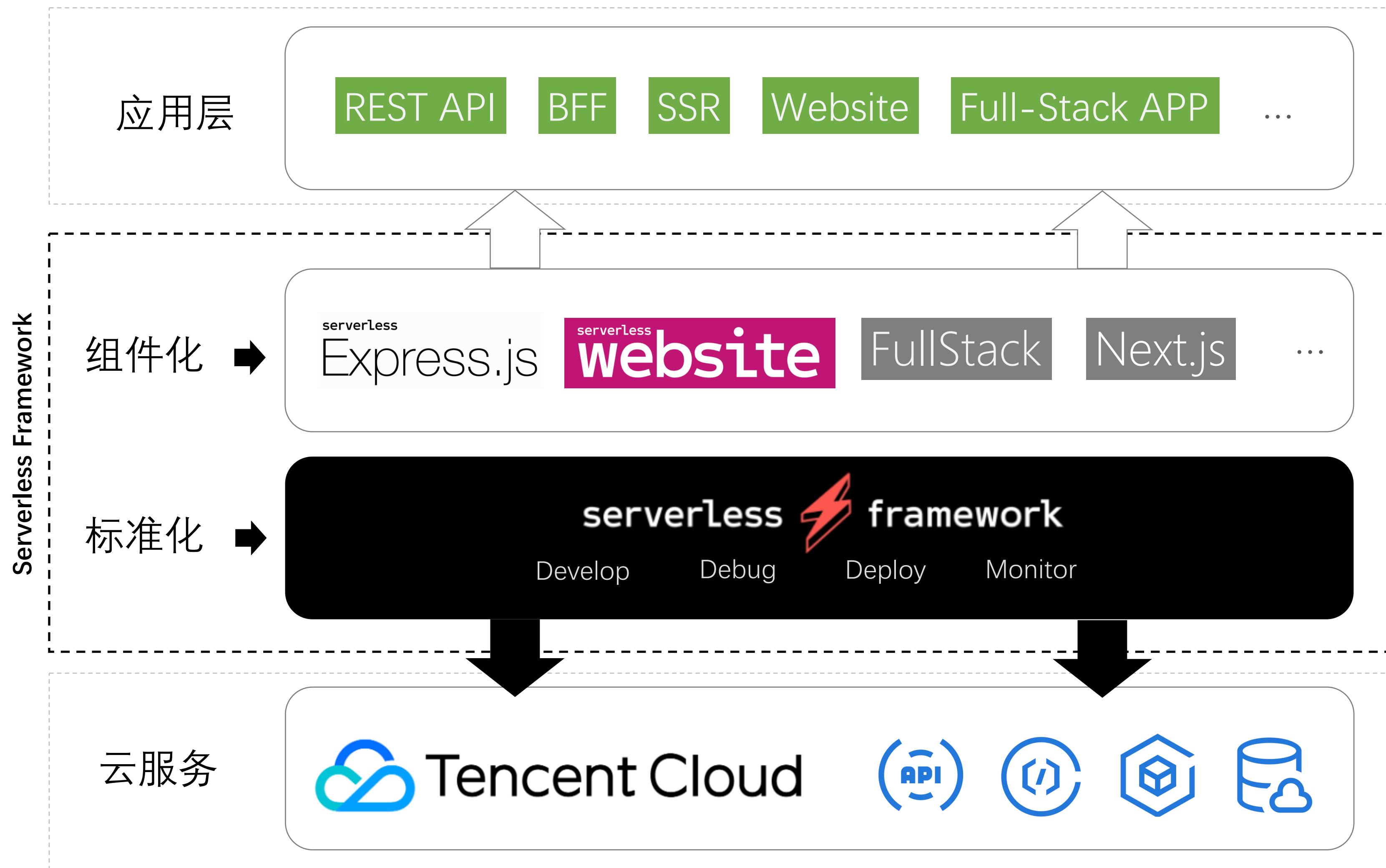


具象结构



抽象结构

Serverless Framework 是一个 标准化、组件化 的 Serverless 应用开发产品



## Serverless Framework

Serverless 应用生命周期管理

CLI & Dashboard

Serverless 应用

Serverless 资源

- ❑ 服务的部署/删除/查看/回滚
- ❑ 调试
- ❑ 查看日志
- ❑ 统计运行数据
- ❑ DevOps 支持

- ❑ 标准化框架
- ❑ 组件化开发

无缝对接腾讯云 Serverless 资源



```
JENSWANG-MB0:~ jenswang$ serverless -h
```

#### Commands

- \* You can run commands with "serverless" or the shortcut "sls"
- \* Pass "--verbose" to this command to get in-depth plugin info
- \* Pass "--no-color" to disable CLI colors
- \* Pass "--help" after any <command> for contextual help

#### Interactive Quickstart

- \* Run serverless (or shortcut sls) without any arguments to initialize an interactive setup of functionalities related to given service or current environment
- \* Pass "--help-interactive" for contextual help on interactive CLI options

#### Framework

- \* Documentation: <http://slss.io/docs>

#### Environment Variables

- \* Set SLS\_DEBUG=\* to see debugging logs
- \* Set SLS\_WARNING\_DISABLE=\* to hide warnings from the output

```
config ..... Configure Serverless
config credentials ..... Configures a new provider profile for the Serverless Framework
config tabcompletion install .. Install a <tab> completion for chosen shell
config tabcompletion uninstall Uninstall a <tab> completion for chosen shell
create ..... Create new Serverless service
install ..... Install a Serverless service from GitHub or a plugin from the Serverless registry
package ..... Packages a Serverless service
deploy ..... Deploy a Serverless service
deploy function ..... Deploy a single function from the service
deploy list ..... List deployed version of your Serverless Service
deploy list functions ..... List all the deployed functions and their versions
invoke ..... Invoke a deployed function
invoke local ..... Invoke function locally
info ..... Display information about the service
logs ..... Output the logs of a deployed function
metrics ..... Show metrics for a specific function
print ..... Print your compiled and resolved config file
remove ..... Remove Serverless service and all resources
rollback ..... Rollback the Serverless service to a specific deployment
rollback function ..... Rollback the function to the previous version
slstats ..... Enable or disable stats
plugin ..... Plugin management for Serverless
plugin install ..... Install and add a plugin to your service
plugin uninstall ..... Uninstall and remove a plugin from your service
plugin list ..... Lists all available plugins
plugin search ..... Search for plugins
login ..... Login or sign up for Serverless
logout ..... Logout from Serverless
generate-event ..... Generate event
test ..... Run HTTP tests
dashboard ..... Open the Serverless dashboard
```

# Serverless Framework CLI

# Serverless Components

```
# serverless.yml

myBucket:
  component: '@serverless/tencent-cos'
  inputs:
    bucket: my-bucket
```

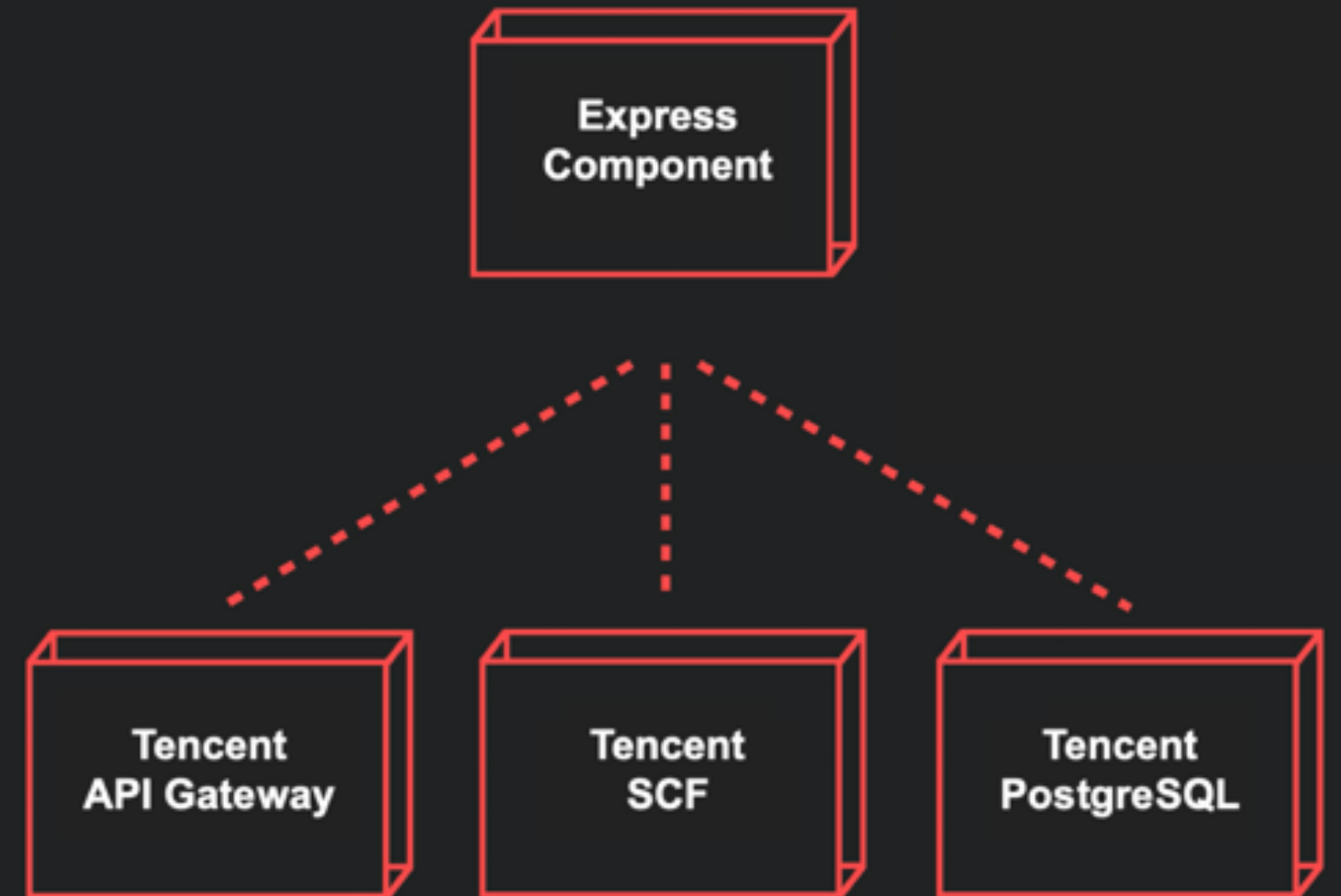


npm 模块，基于YAML配置文件，针对云服务的资源进行使用和配置

# Serverless Components

流式处理，借助云API结合

```
# serverless.yml
express:
  component: '@serverless/tencent-express'
  inputs:
    functionName: eslam-function
    serviceName: mytest
    code: ./code
    functionConf:
      timeout: 10
      memorySize: 128
      environment:
        variables:
          TEST: vale
    apigatewayConf:
      protocol: https
      environment: test
      usagePlan:
        usagePlanName: slscmp
        maxRequestNum: 1000
    postgresConf:
      region: ap-beijing
      dbName: postgres-ec289zvy
```



# Serverless Component 代码结构

```
// serverless.js
const { Component } = require('@serverless/core')
class MyComponent extends Component {
  /*
   * Default (必须)
   * - default 是用来执行、准备和更新你的组建的函数
   * - 执行命令 ` $ serverless ` 会运行此函数
   * - You can run this function by running the "$ serverless" command
   */
  async default(inputs = {}) {
    return {}
  }

  /*
   * Remove (可选)
   * - 如果你的组件需要删除基础设施, 推荐你添加他
   * - 执行命令 ` $ serverless remove ` 会运行此函数
   */
  async remove(inputs = {}) {
    return {}
  }

  /*
   * Anything (可选)
   * - 如果你想发布带有额外功能的组件, 你可以将逻辑写在一个函数里, 函数名可以自定义
   * - 执行命令 ` $ serverless anything ` 会运行此函数
   */
  async anything(inputs = {}) {
    return {}
  }
}
module.exports = MyComponent
```

# Serverless Full-stack Application

通过组装集成组件形成高阶组件

```
# serverless.yml
```

```
name: fullstack-application-postgres
```

```
frontend:
```

```
  component: '@serverless/tencent-website'
```

```
  inputs:
```

```
    protocol: https
```

```
    code:
```

```
      src: dist
```

```
      root: frontend
```

```
      hook: npm run build
```

```
  env:
```

```
    apiUrl: ${api.url}
```

```
api:
```

```
  component: '@serverless/tencent-express'
```

```
  inputs:
```

```
    region: ap-beijing
```

```
    code: ./api
```

```
    functionName: fullstack-api-pg
```

```
    role: execute-scf
```

```
  apigatewayConf:
```

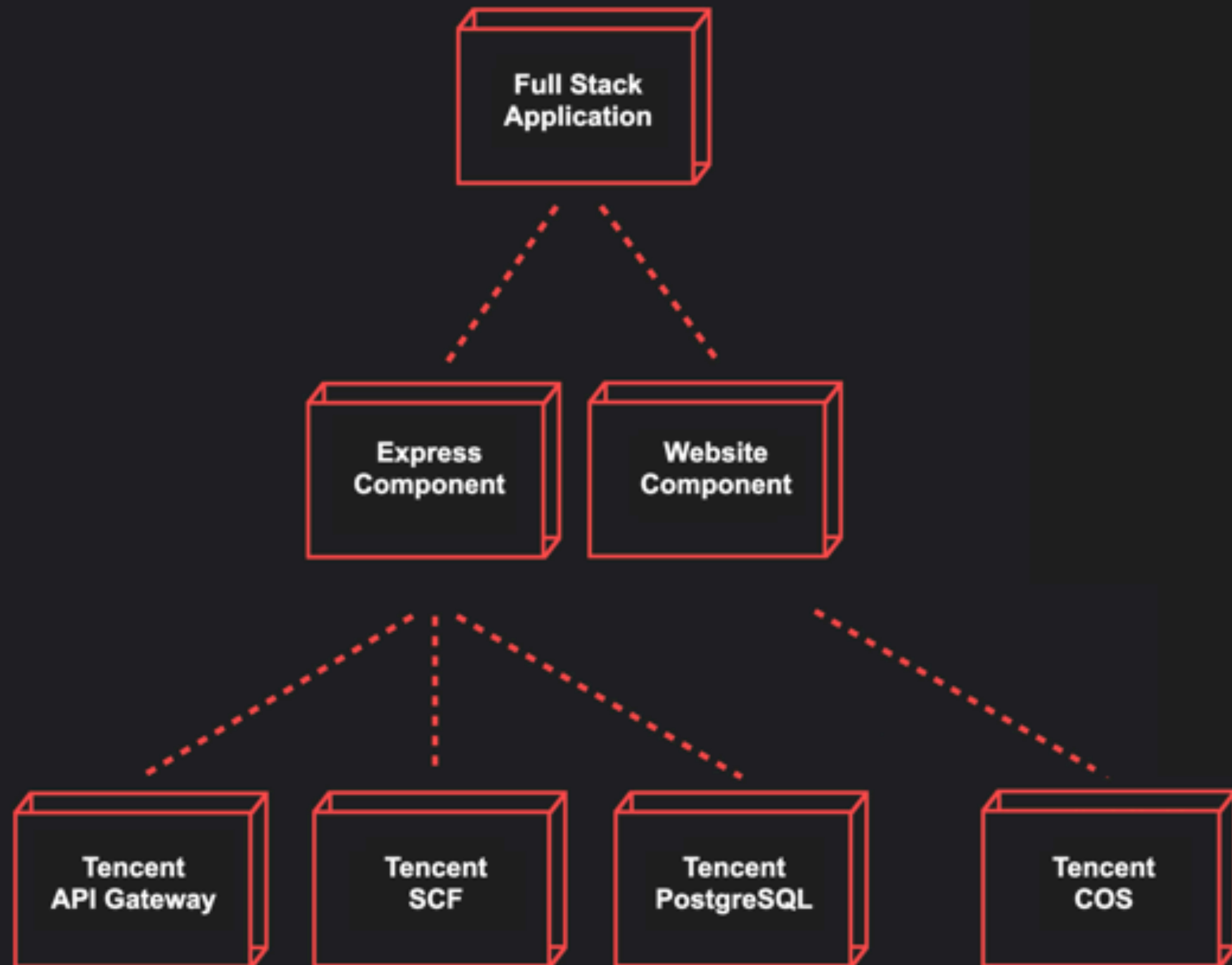
```
    protocol: https
```

```
postgres:
```

```
  component: '@serverless/tencent-postgres'
```

```
  inputs:
```

```
    region: ap-beijing
```



# 目录

前端与 Serverless 的不解之缘

Serverless 前端工程化的基本思路

Serverless Framework 原理与实现

**Serverless Now**

# Live Code



+

serverless



framework



D:\sf>

# a Website

- HTTP Server
- HTTPS SSL 证书
- 对象存储COS
- CDN
- Domain CNAME
- 进程监控/自动重启
- 防攻击/DDos
- 日志清理
- 数据分析/日志挖掘

Tencent Cloud  
**provider**  
serverless ⚡ framework

serverless  
**Express.js**  
serverless ⚡ framework + Tencent Cloud

serverless  
**website**  
serverless ⚡ framework + Tencent Cloud

serverless  
**Flask**  
serverless ⚡ framework + Tencent Cloud

serverless  
**React**  
serverless ⚡ framework + Tencent Cloud

serverless  
**Vue.js**  
serverless ⚡ framework + Tencent Cloud

serverless  
**{ REST }**  
serverless ⚡ framework + Tencent Cloud

serverless  
**Slim**  
serverless ⚡ framework + Tencent Cloud

serverless  
**fastify**  
serverless ⚡ framework + Tencent Cloud

serverless  
**PostgreSQL**  
serverless ⚡ framework + Tencent Cloud

serverless  
**django**  
serverless ⚡ framework + Tencent Cloud

serverless  
**Laravel**  
serverless ⚡ framework + Tencent Cloud

serverless

# Express.js

serverless  framework +  Tencent Cloud

<https://github.com/serverless-components/tencent-express>

serverless  
Express.js

## 1. 安装

通过 npm 安装 serverless

```
$ npm install -g serverless
```

## 2. 创建

本地创建 `serverless.yml` 文件:

```
$ touch serverless.yml
```

初始化一个新的 npm 包, 并安装 Express:

```
npm init          # 创建后持续回车  
npm i --save express # 安装express
```

创建一个 `app.js` 文件, 并在其中创建您的 Express App:

```
const express = require('express')  
const app = express()  
  
app.get('/', function(req, res) {  
  res.send('Hello Express')  
})  
  
// don't forget to export!  
module.exports = app
```

serverless

# Express.js

## 3. 配置

在 serverless.yml 中进行如下配置

```
# serverless.yml

express:
  component: '@serverless/tencent-express'
  inputs:
    region: ap-shanghai
```

```
# serverless.yml

express:
  region: ap-shanghai
  component: '@serverless/tencent-express'
  inputs:
    region: ap-shanghai
    functionName: eslam-function
    serviceName: mytest
    serviceId: service-np1uloxw
    code: ./code
  functionConf:
    timeout: 10
    memorySize: 128
    environment:
      variables:
        TEST: vale
    vpcConfig:
      subnetId: ''
      vpcId: ''
  apigatewayConf:
    protocols:
      - http
      - https
    environment: test
  usagePlan:
    usagePlanId: 1111
    usagePlanName: slscmp
    usagePlanDesc: sls create
    maxRequestNum: 1000
  auth:
    serviceTimeout: 15
    secretName: secret
    secretIds:
      - AKIDNSdvdFcJ8GJ9th6qeZH01l8r7dE6HHasuchJ
```

## 4. 部署

您可以通过 [微信](#) 扫描命令行中的二维码进行授权登陆和注册。

通过 `sls` 命令进行部署，并可以添加 `--debug` 参数查看部署过程中的信息

```
$ sls --debug

DEBUG - Resolving the template's static variables.
DEBUG - Collecting components from the template.
DEBUG - Downloading any NPM components found in the template.
DEBUG - Analyzing the template's components dependencies.
DEBUG - Creating the template's components graph.
DEBUG - Syncing template state.
DEBUG - Executing the template's components graph.
DEBUG - Compressing function ExpressComponent_7xRrrd file to /Users/dfounderliu/Desktop/temp/code/.serverless/ExpressComponent_7xRrrd.zip
DEBUG - Compressed function ExpressComponent_7xRrrd file successful
DEBUG - Uploading service package to cos[sls-cloudfunction-ap-shanghai-code]. sls-cloudfunction-default
DEBUG - Uploaded package successful /Users/dfounderliu/Desktop/temp/code/.serverless/ExpressComponent_7xRrrd.zip
DEBUG - Creating function ExpressComponent_7xRrrd
DEBUG - Created function ExpressComponent_7xRrrd successful
DEBUG - Starting API-Gateway deployment with name express.TencentApiGateway in the ap-shanghai region
DEBUG - Using last time deploy service id service-n0vs2ohb
DEBUG - Updating service with serviceId service-n0vs2ohb.
DEBUG - Endpoint ANY / already exists with id api-9z60urs4.
DEBUG - Updating api with api id api-9z60urs4.
DEBUG - Service with id api-9z60urs4 updated.
DEBUG - Deploying service with id service-n0vs2ohb.
DEBUG - Deployment successful for the api named express.TencentApiGateway in the ap-shanghai region.

express:
  region:          ap-shanghai
  functionName:    ExpressComponent_7xRrrd
  apiGatewayServiceId: service-n0vs2ohb
  url:             http://service-n0vs2ohb-1300415943.ap-shanghai.apigateway.myqcloud.com/release/

36s > express > done
```



## 5. 移除

通过以下命令移除部署的存储桶

```
$ sls remove --debug  
  
DEBUG – Flushing template state and removing all components.  
DEBUG – Removed function ExpressComponent_MHrAzr successful  
DEBUG – Removing any previously deployed API. api-kf2hxrhc  
DEBUG – Removing any previously deployed service. service-4ndfl6pz  
  
13s > express > done
```

# koa

next generation web framework for node.js



基于 Serverless 的 koa 和 egg 的 component , 已由社区开发者贡献

# What's next?

**将于近期发布...**

- **实时日志**
- **远程调试**
- **Serverless DB**
- **监控平台**

# 学习资源

## 中文技术社区

<https://serverlessCloud.cn>

## 官网

<https://serverless.com>

<https://serverless.com/cn> (即将上线)

## Github

<https://github.com/serverless>

<https://github.com/serverless-components>

## 腾讯云文档

<https://cloud.tencent.com/product/sf>

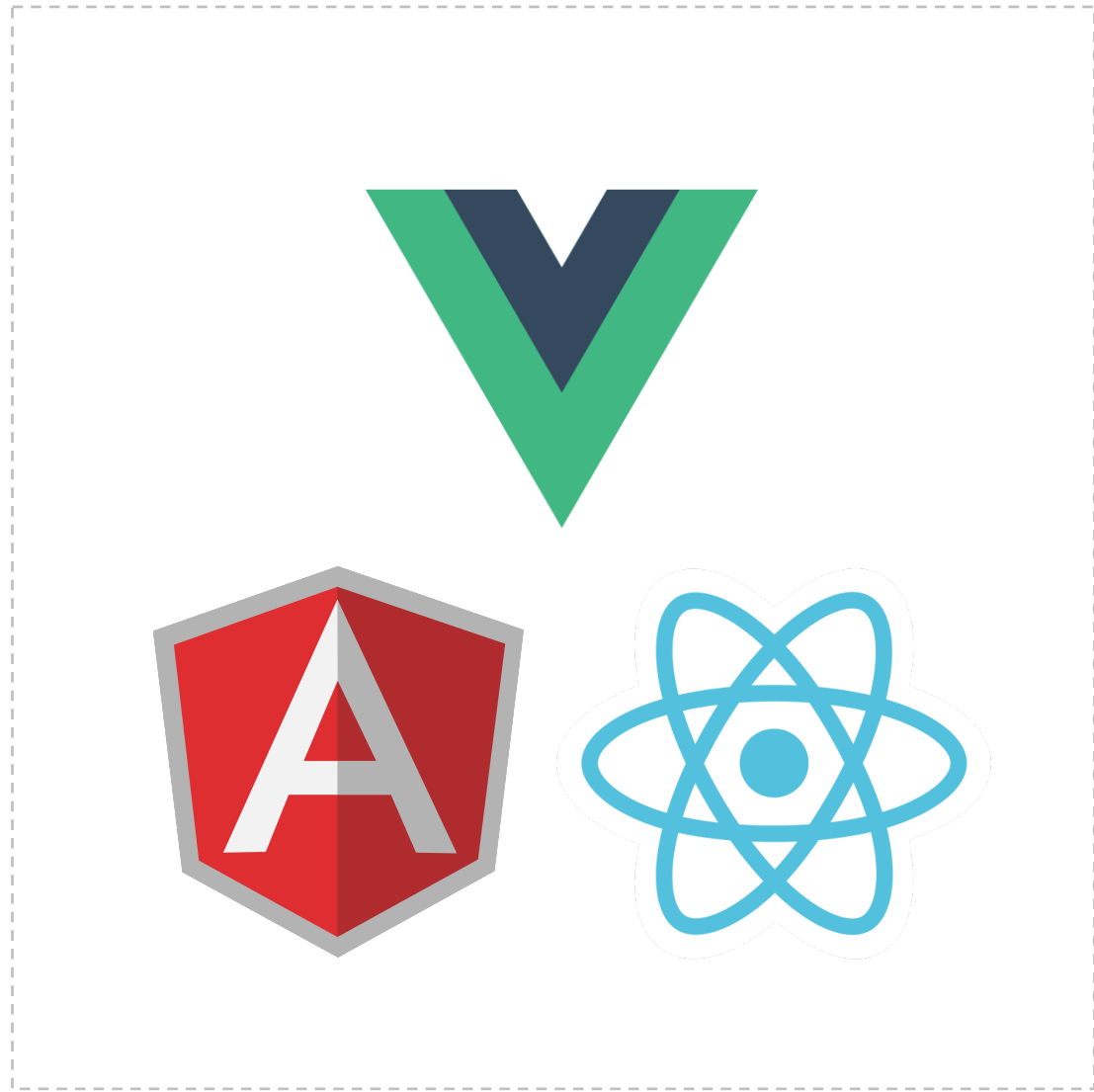
# 总结

前端与 Serverless 的不解之缘

Serverless 前端工程化的基本思路

Serverless Framework 原理

Serverless Now



组件化



工程化



Serverless



# Austen

**CEO and founder of serverless.com**

Serverless is the future of the cloud. Serverless is like superpowers for developers. We believe the future is all about developers and serverless especially empowers developers because it enables them to build more manage less.





# 收获国内外一线大厂实践 与技术大咖同行成长

- ✓ 演讲视频
- ✓ 干货整理
- ✓ 大咖采访
- ✓ 行业趋势





扫码查看

# 60 天学透算法与数据结构

600+ 名企内推通道向你打开!

✓ 大厂内推 ✓ 实战演练 ✓ 闭环体系 ✓ 社群连接

THANKS

GMTC  
全球大前端技术大会