



Cloud Foundry 体系结构

Jennifer Hickey



关于 Jennifer

- Cloud Foundry 工程师
- 专注于框架支持
- SpringSource 资深开发者
- 参与过众多 Spring 和 SpringSource 项目
- 热衷于在云中提高开发者的工作效率。

联系信息

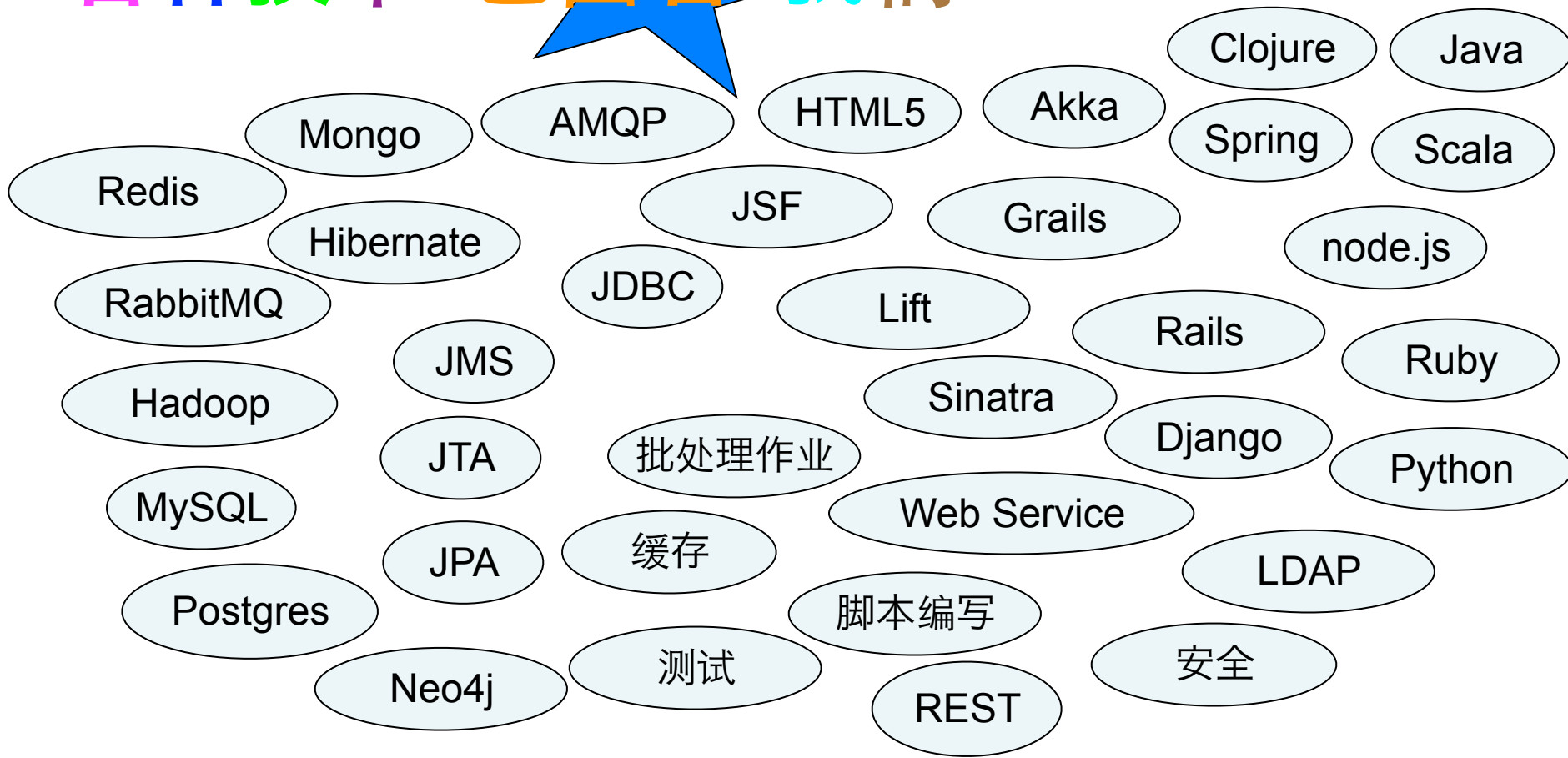
Twitter : @jencompgeek

电子邮件 : jhickey@vmware.com

各种技术“包围着”我们



各种技术“包围着”我们



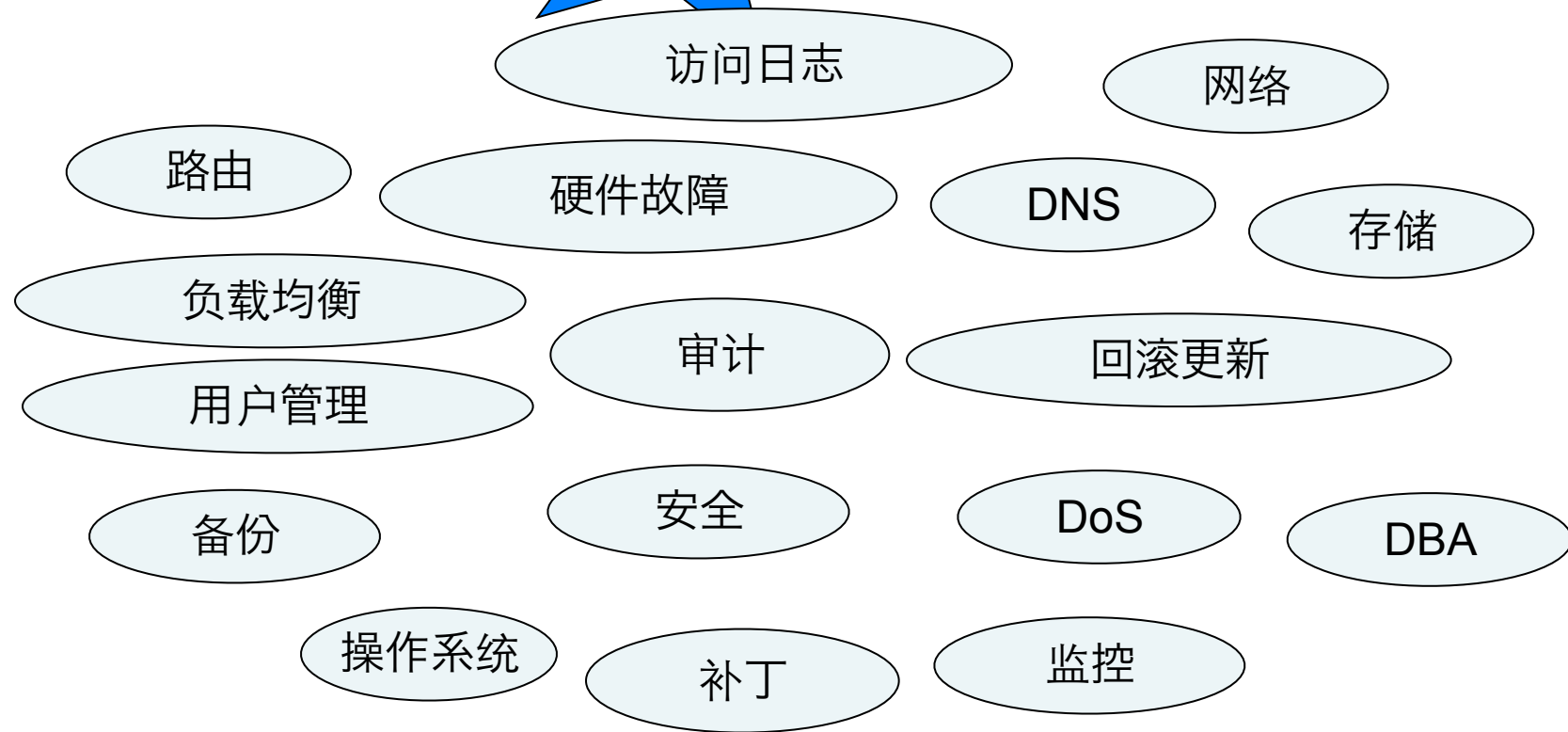
各种技术“包围着”我们



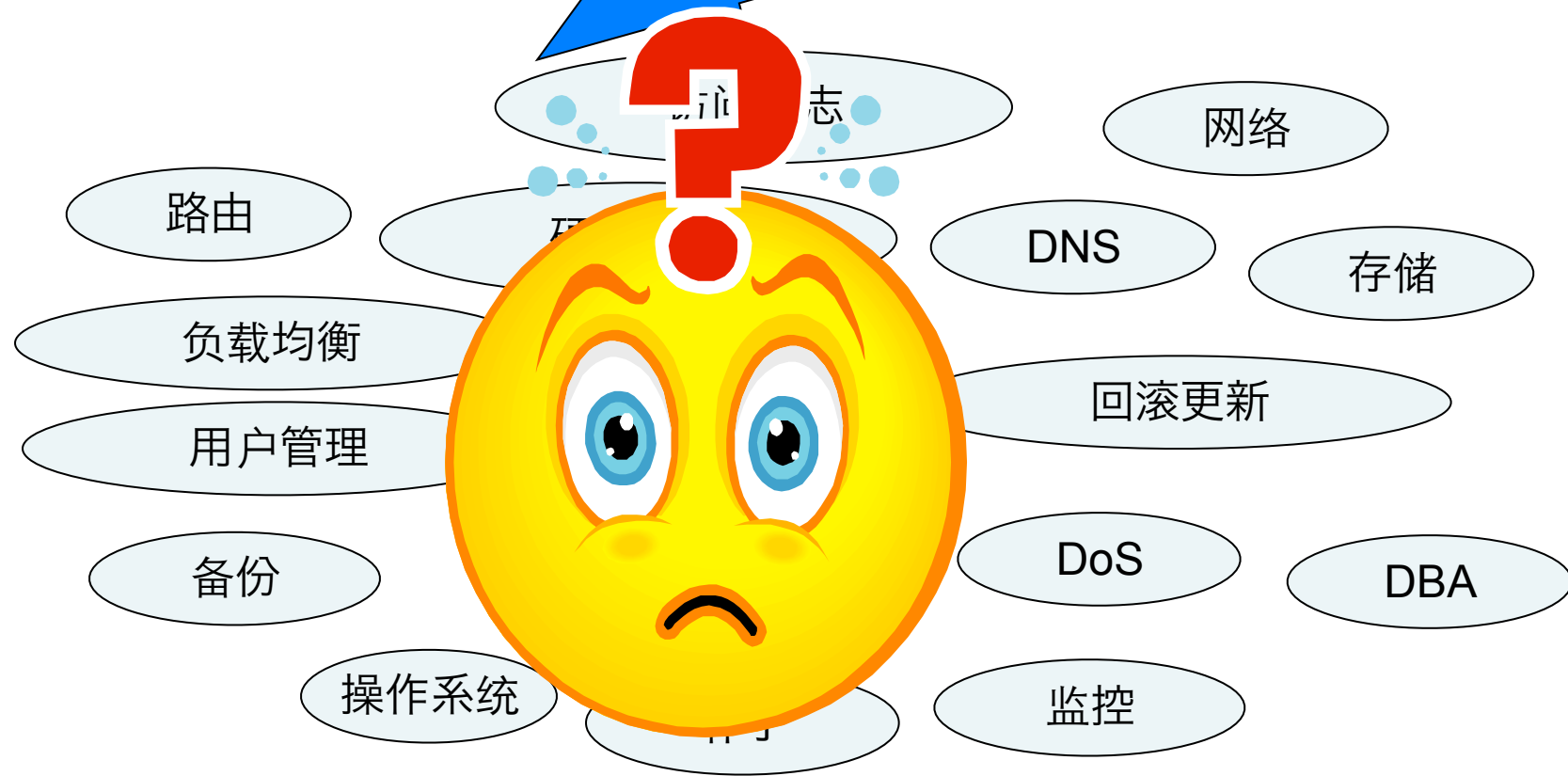
各种技术“包围着”我们



各种技术“包围着”我们



各种技术“包围着”我们



复杂性的各个方面：产品

- 更多功能
- 上市时间压力
- 复杂的集成
- 质量重要性提高



复杂性的各个方面：开发

- 稳健的体系结构：既面向未来又不激进
- 单元和集成测试
- 响应不断变化的业务需求
- 令人混乱



复杂性的各个方面：部署和操作

- 选择合适的硬件、操作系统和 Web 服务器
- 监控应用程序
- 响应扩展需求
- 处理硬件和系统级别的故障
- 不



DevOps Borat
@DEVOPS_BORAT



Follow

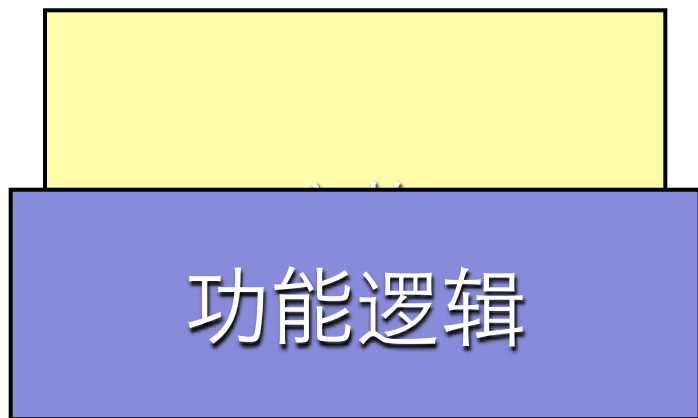
In cloud you need of launch 100 instance/day and terminate 99 instance/day just for stay in game.

内在与外在复杂性

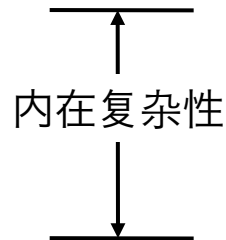
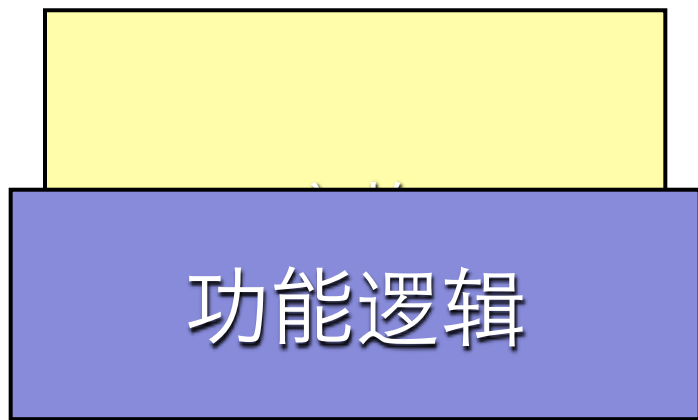


实施

内在与外在复杂性



内在与外在复杂性



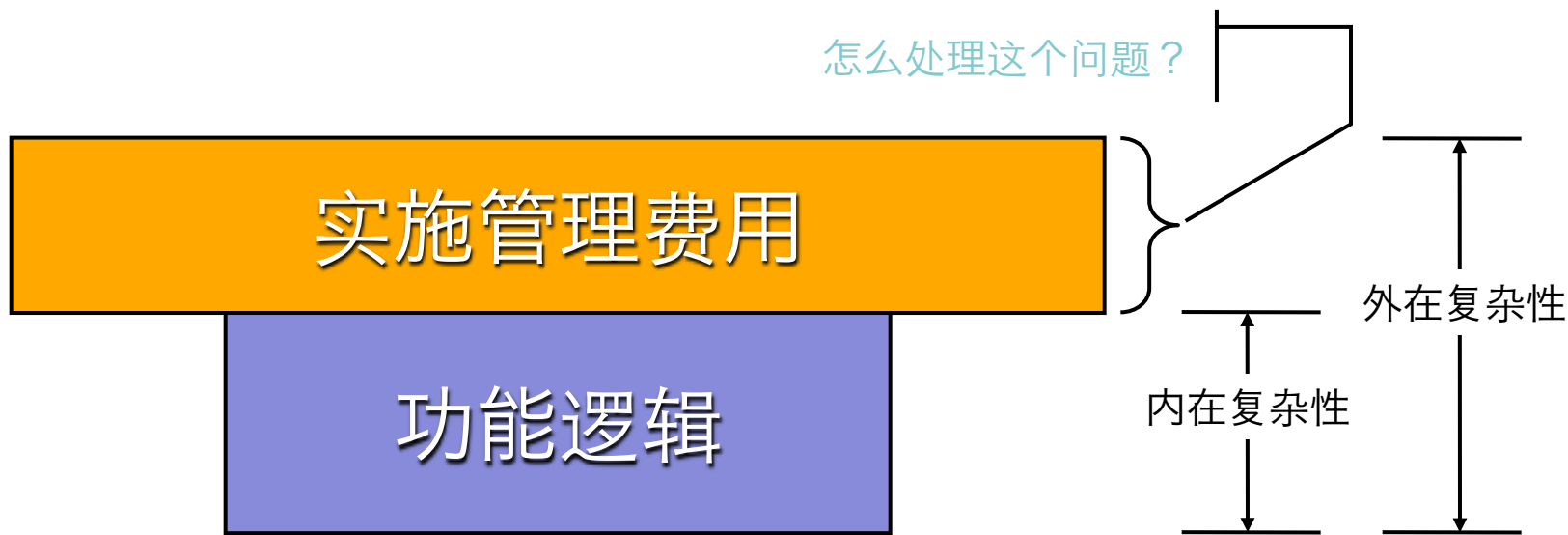
内在与外在复杂性



内在与外在复杂性



内在与外在复杂性



我们怎么办？



云计算三个层级

SaaS

软件即服务



PaaS

平台即服务



IaaS

基础架构即服务



Cloud Foundry 开源 PaaS – 云的选择



Cloud Foundry 开源 PaaS – 云的选择



Cloud Foundry 开源 PaaS – 云的选择



Apache2 许可证

Cloud Foundry 体系结构

架构目标

架构目标

- 无单点故障

架构目标

- 无单点故障
- 分布式状态

架构目标

- 无单点故障
- 分布式状态
- 自我修复

架构目标

- 无单点故障
- 分布式状态
- 自我修复
- 可水平扩展

体系结构原则

体系结构原则

- 松散耦合

体系结构原则

- 松散耦合
- 事件驱动

体系结构原则

- 松散耦合
- 事件驱动
- 异步

体系结构原则

- 松散耦合
- 事件驱动
- 异步
- 非阻塞式

体系结构原则

- 松散耦合
- 事件驱动
- 异步
- 非阻塞式
- 幂等 (Idempotent)

体系结构原则

- 松散耦合
- 事件驱动
- 异步
- 非阻塞式
- 幂等 (Idempotent)
- 最终一致性

体系结构原则

- 松散耦合
- 事件驱动
- 异步
- 非阻塞式
- 幂等 (Idempotent)
- 最终一致性
- 独立于语言的通信

Cloud Foundry 层次

Cloud Foundry 层次

客户端

Cloud Foundry 层次

客户端

内层

Cloud Foundry 层次

客户端

内层

外层

Cloud Foundry 层次



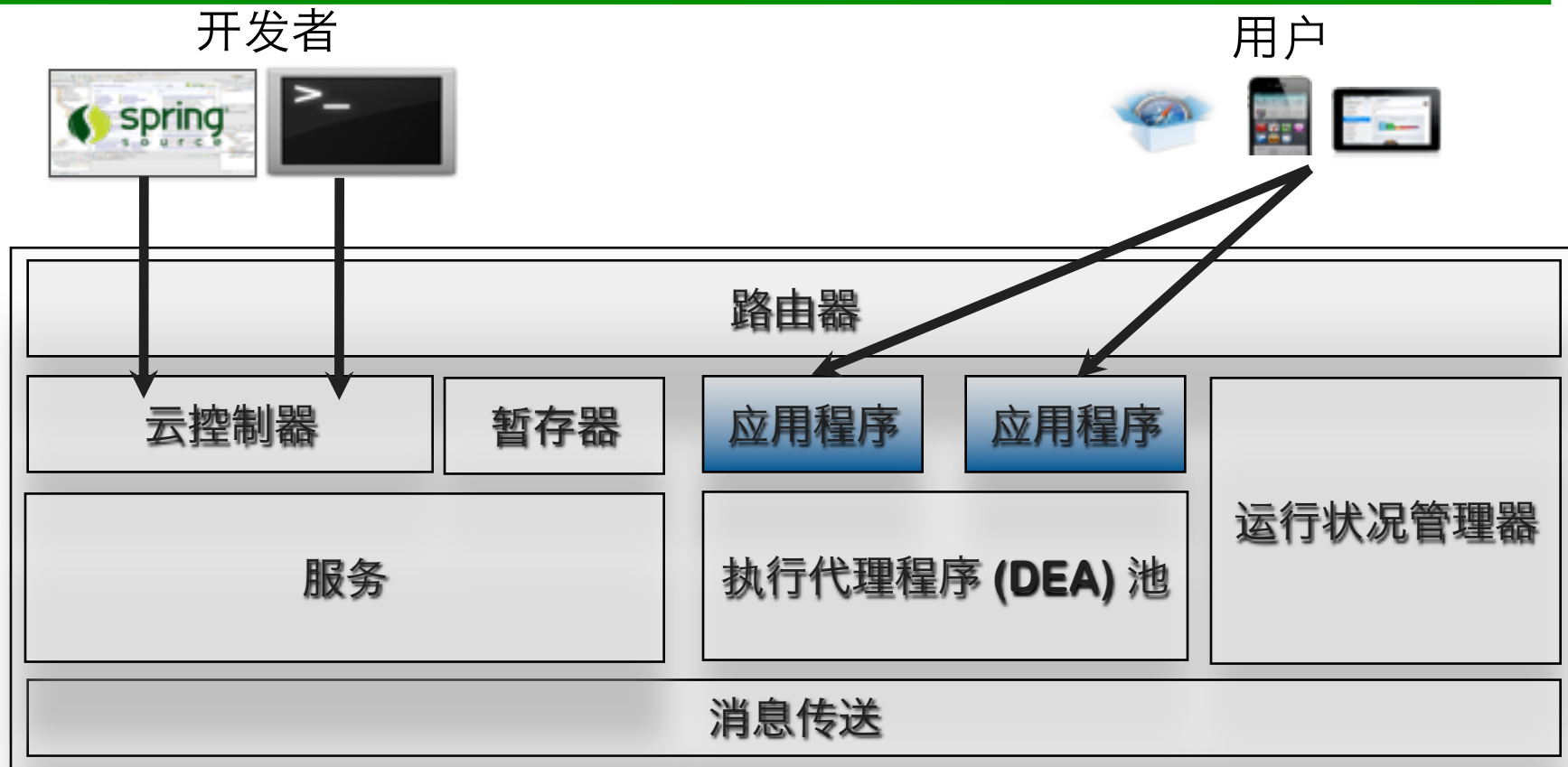
Cloud Foundry 层次



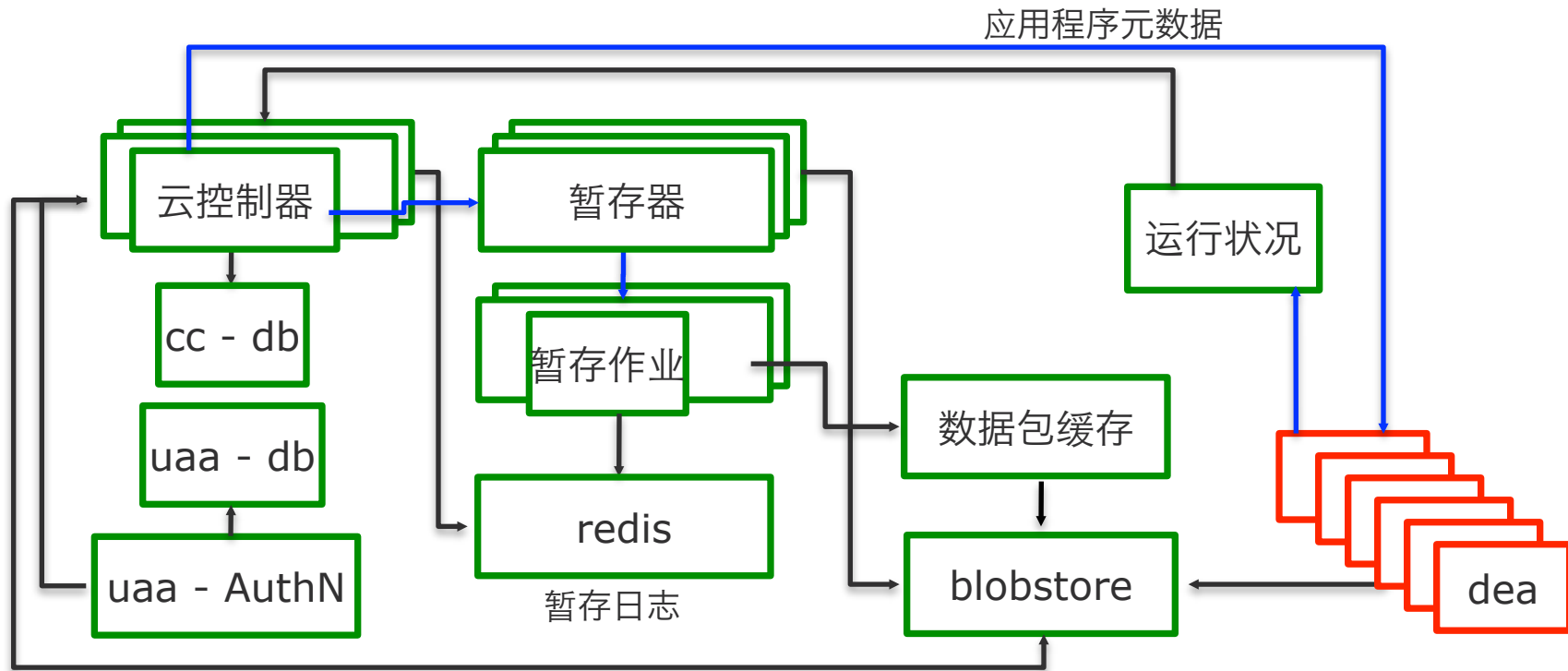
Cloud Foundry 核心组件

内层

Cloud Foundry 内层及运行



细节描绘...



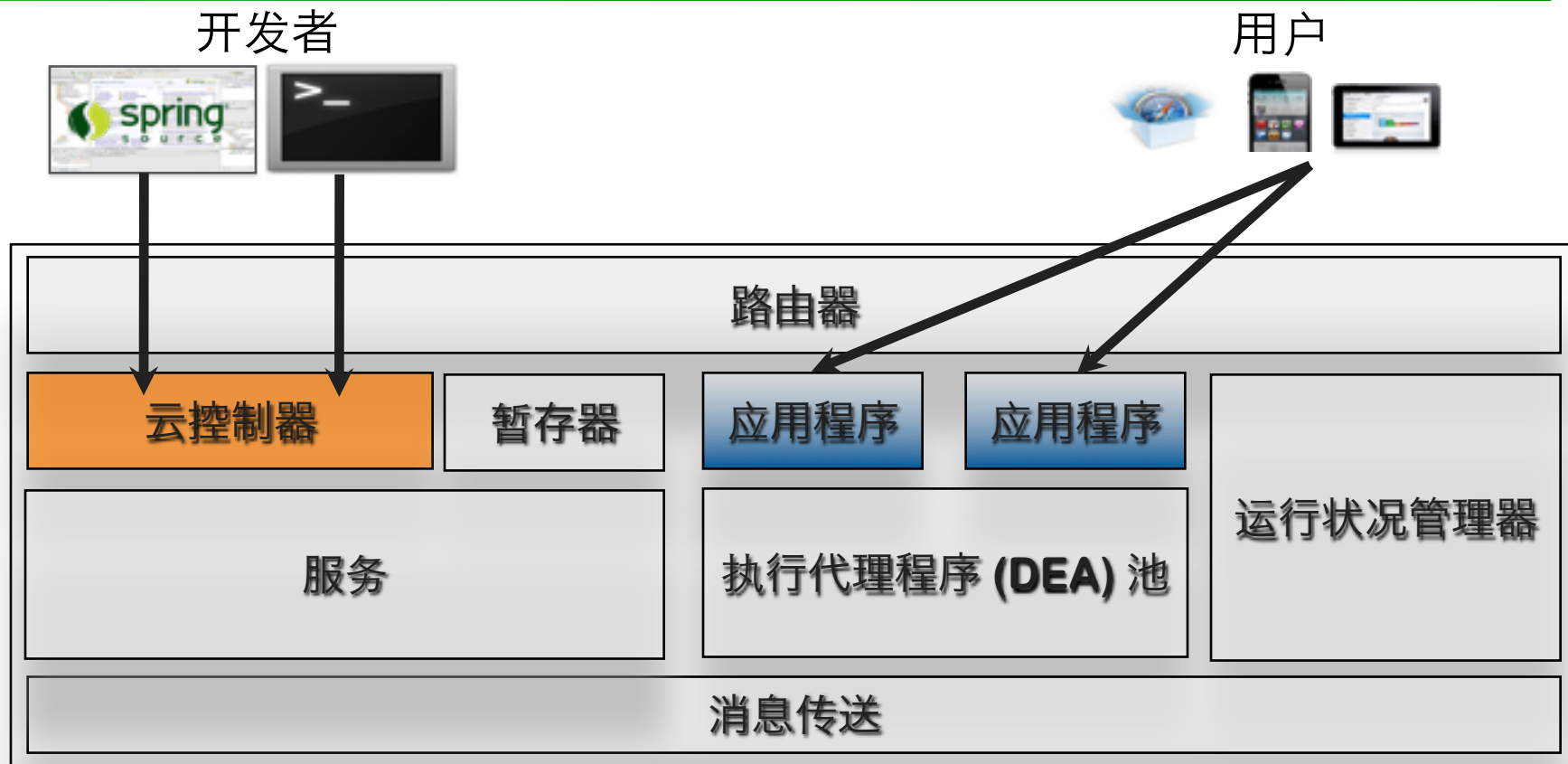
设计原则

- 可动态发现的组件
- 无组件间依赖性
 - 能以任何顺序启动
 - 可独立向上和向下扩展
- 可使用 HTTP 端进行监控

CLOUD CONTROLLER



Cloud Foundry 内层及运行



云控制器

云控制器

- 与客户端交互
 - VMC
 - STS
 - 门户

云控制器

- 与客户端交互
 - VMC
 - STS
 - 门户
- 向域对象提供 REST 接口
 - 应用程序
 - 服务
 - 组织
 - 空间

云控制器 REST API

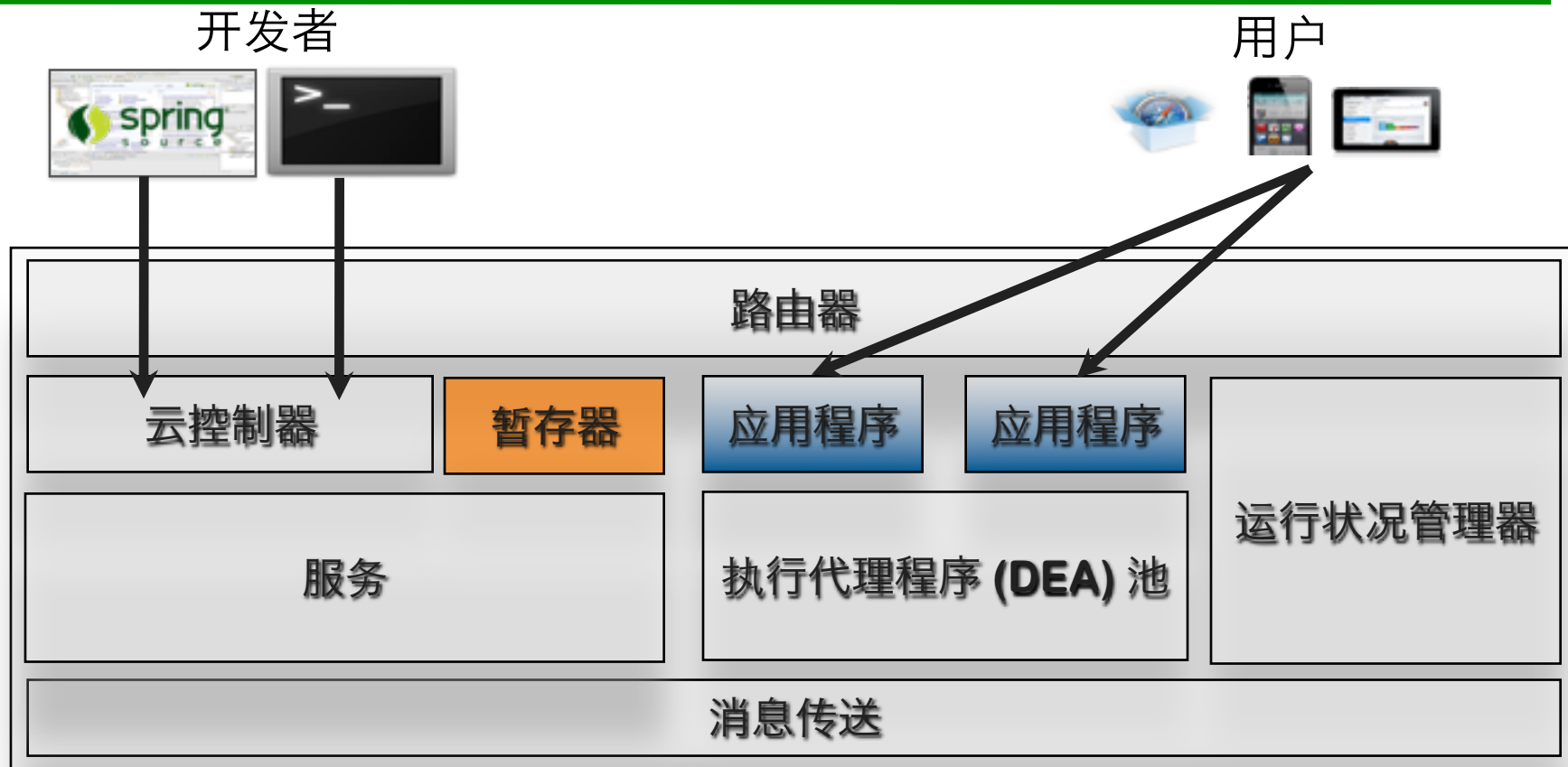
演示

STAGER



ALCANTARA

Cloud Foundry 内层及运行



暂存器

- 负责将用户应用程序改造成可执行程序
- 可插拔体系结构
 - 每个插件都能识别一个框架或运行时
- 让 DEA 能够统一地查看应用程序

暫存器的角色



从比特到可执行应用程序

从比特到可执行应用程序



app.war

从比特到可执行应用程序



从比特到可执行应用程序

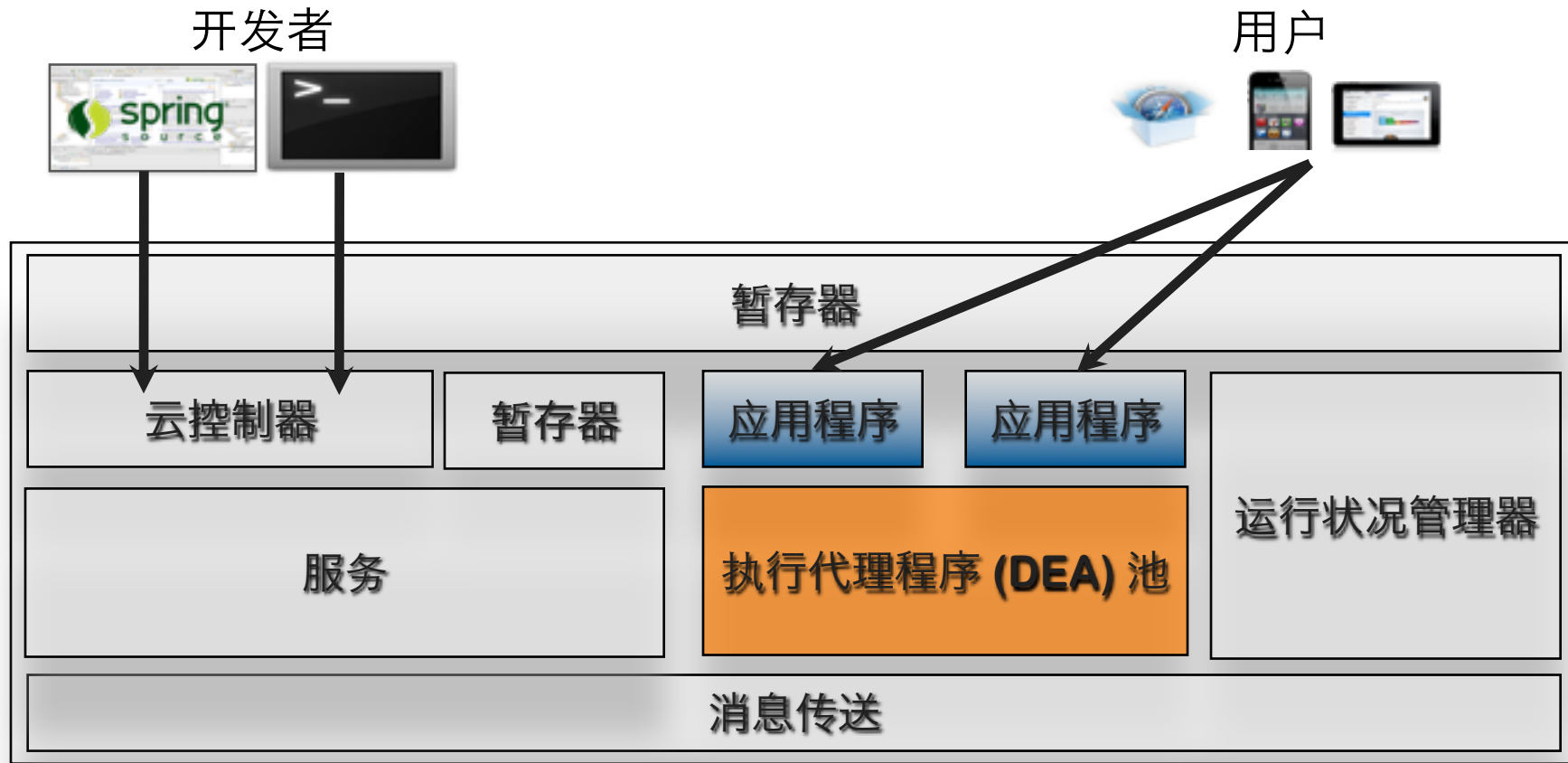


从比特到可执行应用程序



DEA

Cloud Foundry 内层及运行

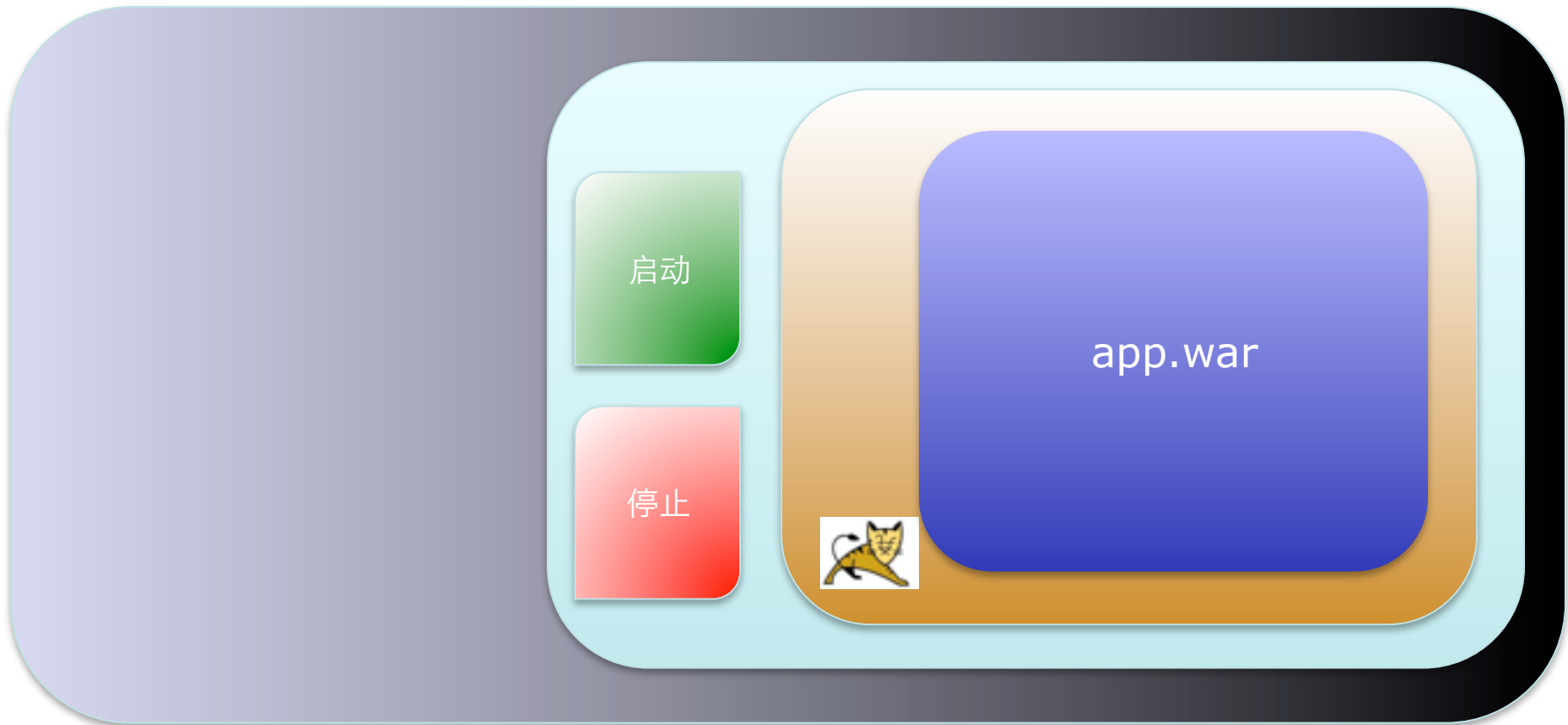


从比特到正在运行的应用程序

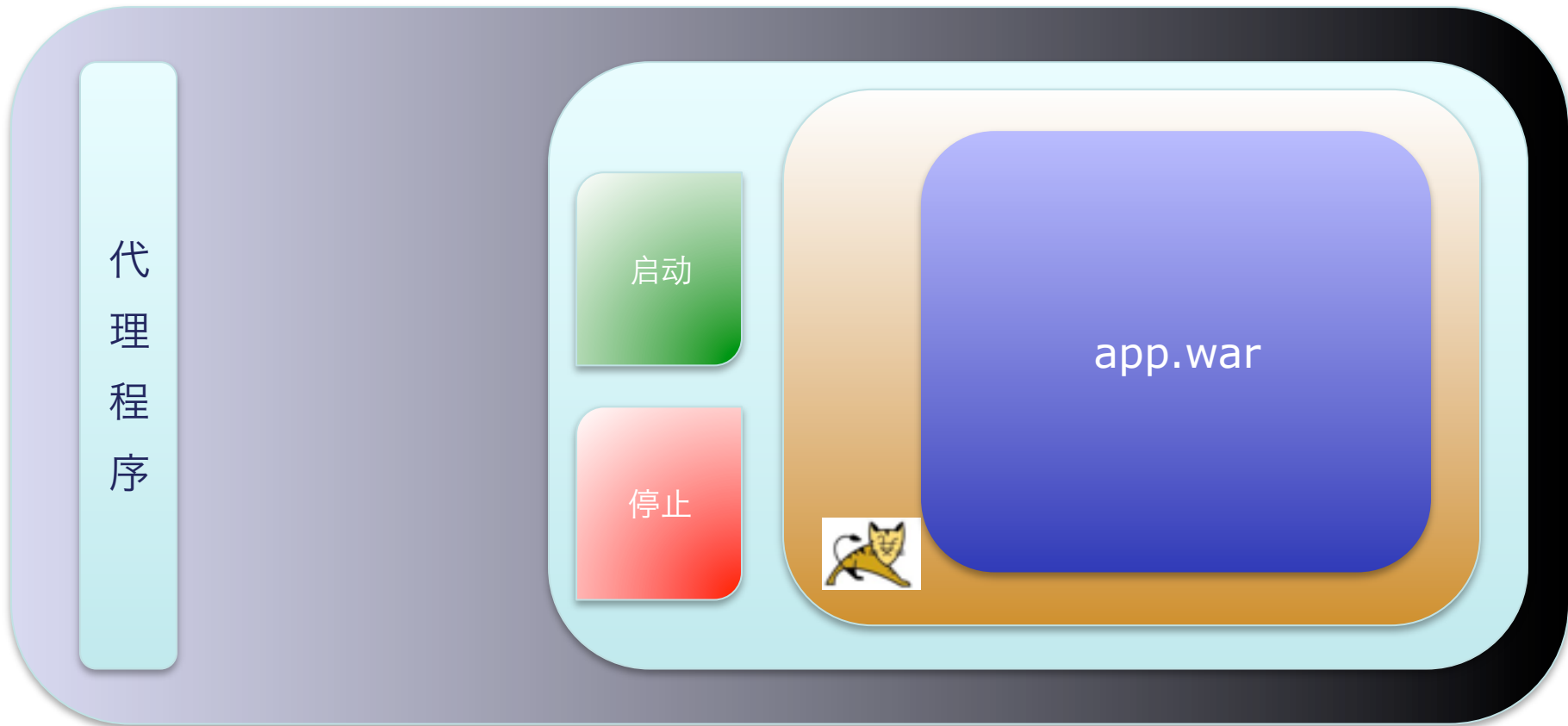
从比特到正在运行的应用程序



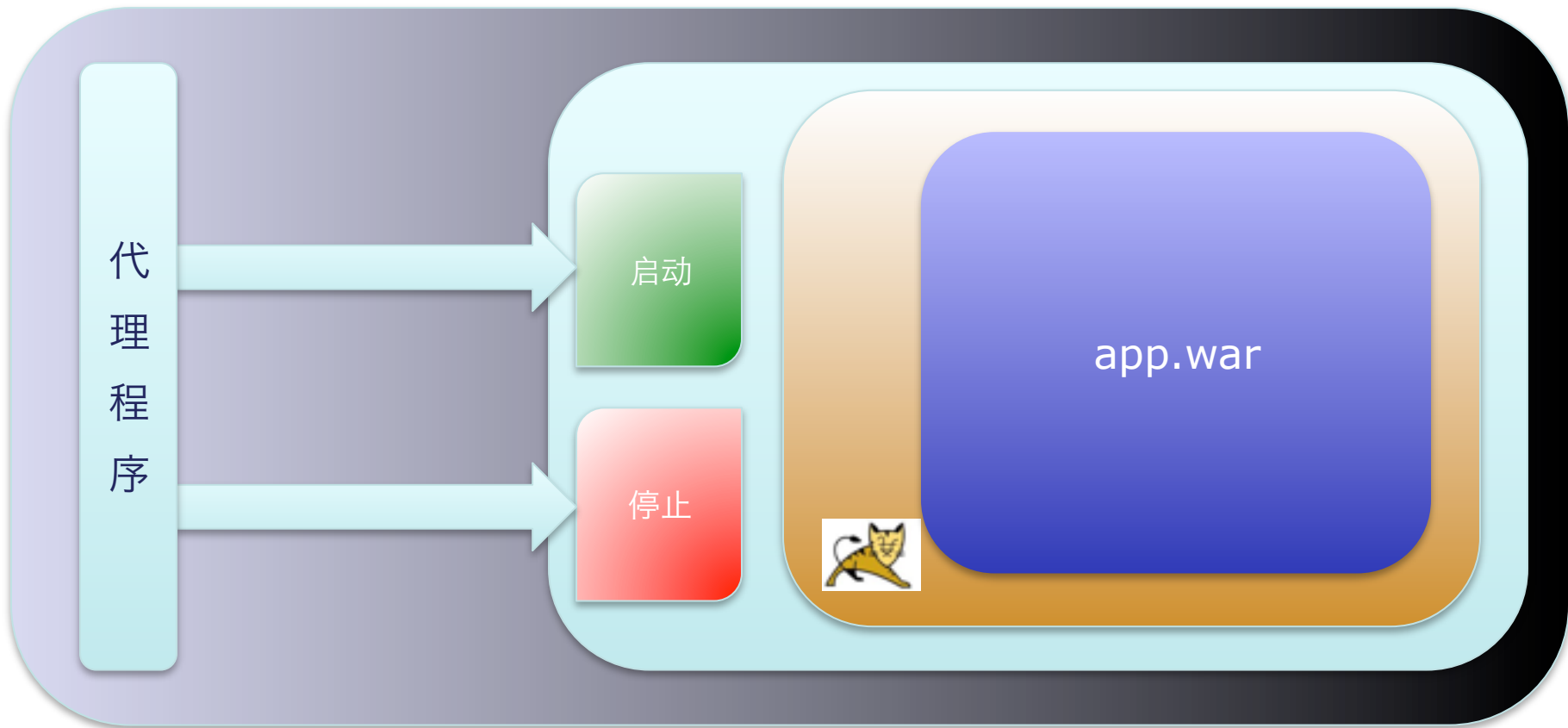
从比特到正在运行的应用程序



从比特到正在运行的应用程序



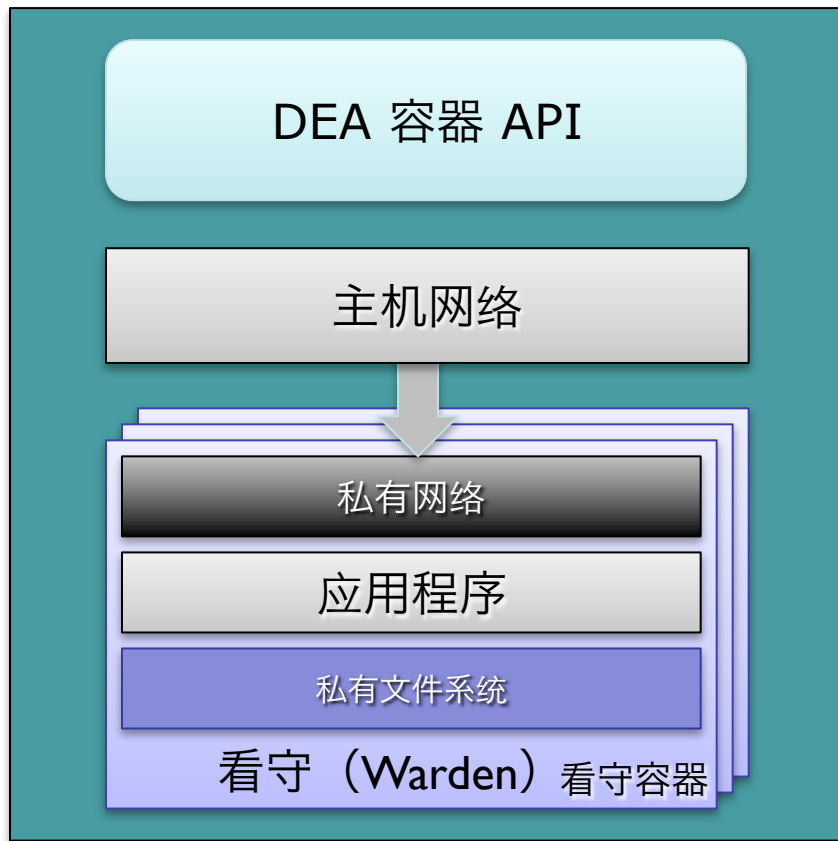
从比特到正在运行的应用程序



Droplet 执行代理程序 (DEA)

- 负责运行所有应用程序
- 监控应用程序
 - 内存和磁盘配额
 - 暂存更改
- 所有应用程序的统一视图
 - 暂存器排列的运行时/框架差异
- 确保应用程序的隔离

应用程序隔离

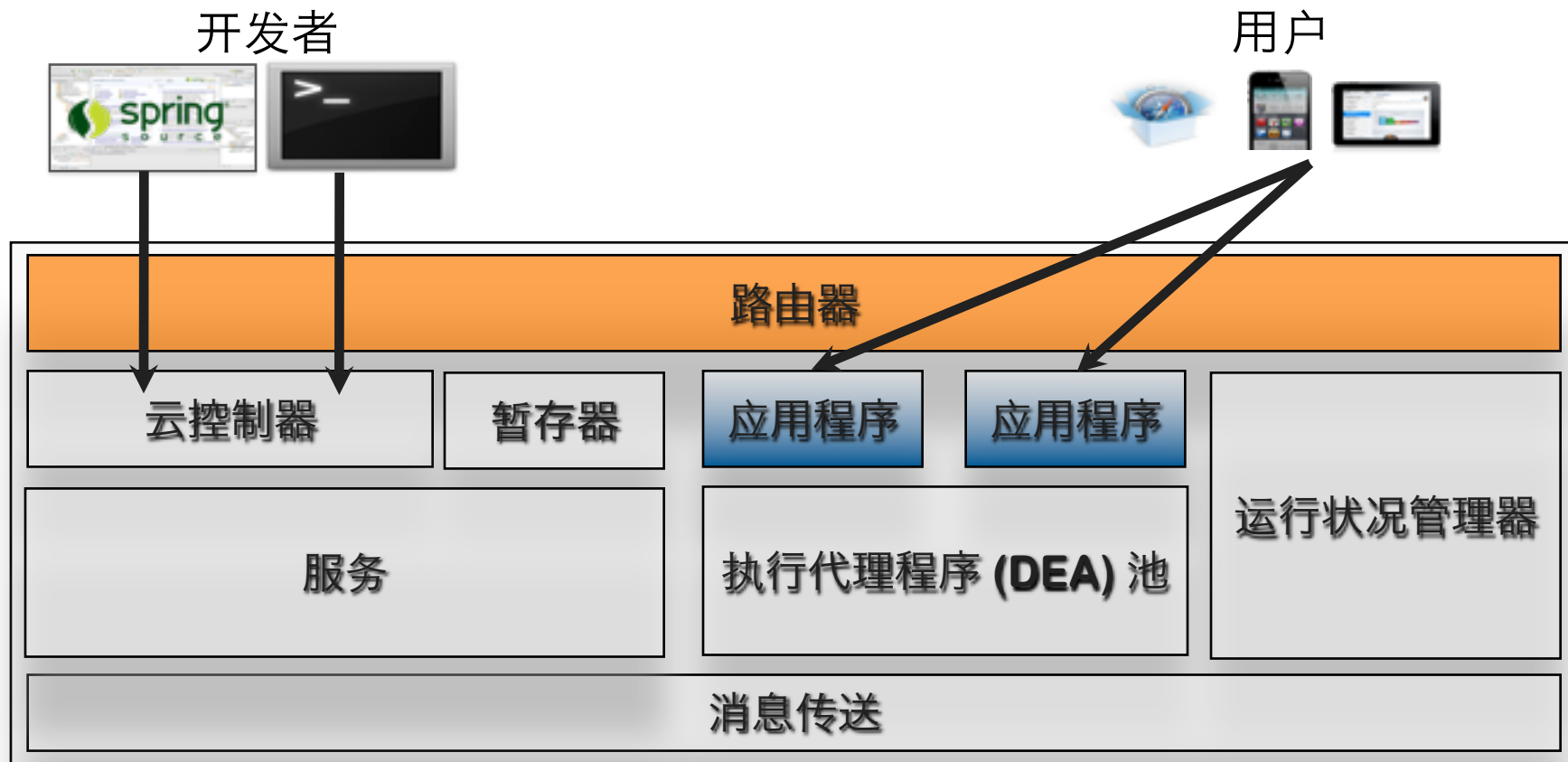


ROUTER



Sunday, December 9, 12

Cloud Foundry 内层及运行



路由器

- 负责将请求路由到
 - 用户应用程序
 - 面向外部的组件
 - 云控制器
 - UAA

应用程序路由

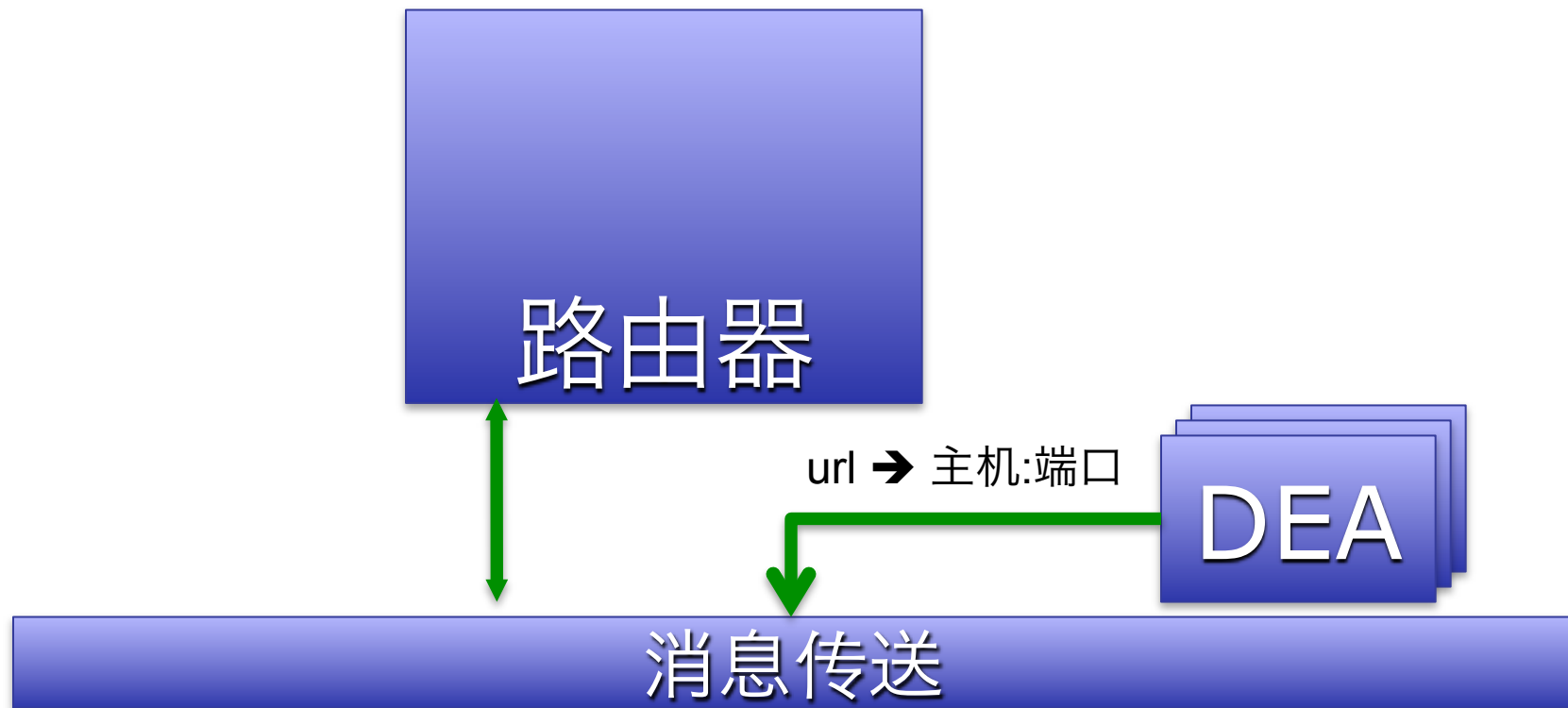
应用程序路由

消息传送

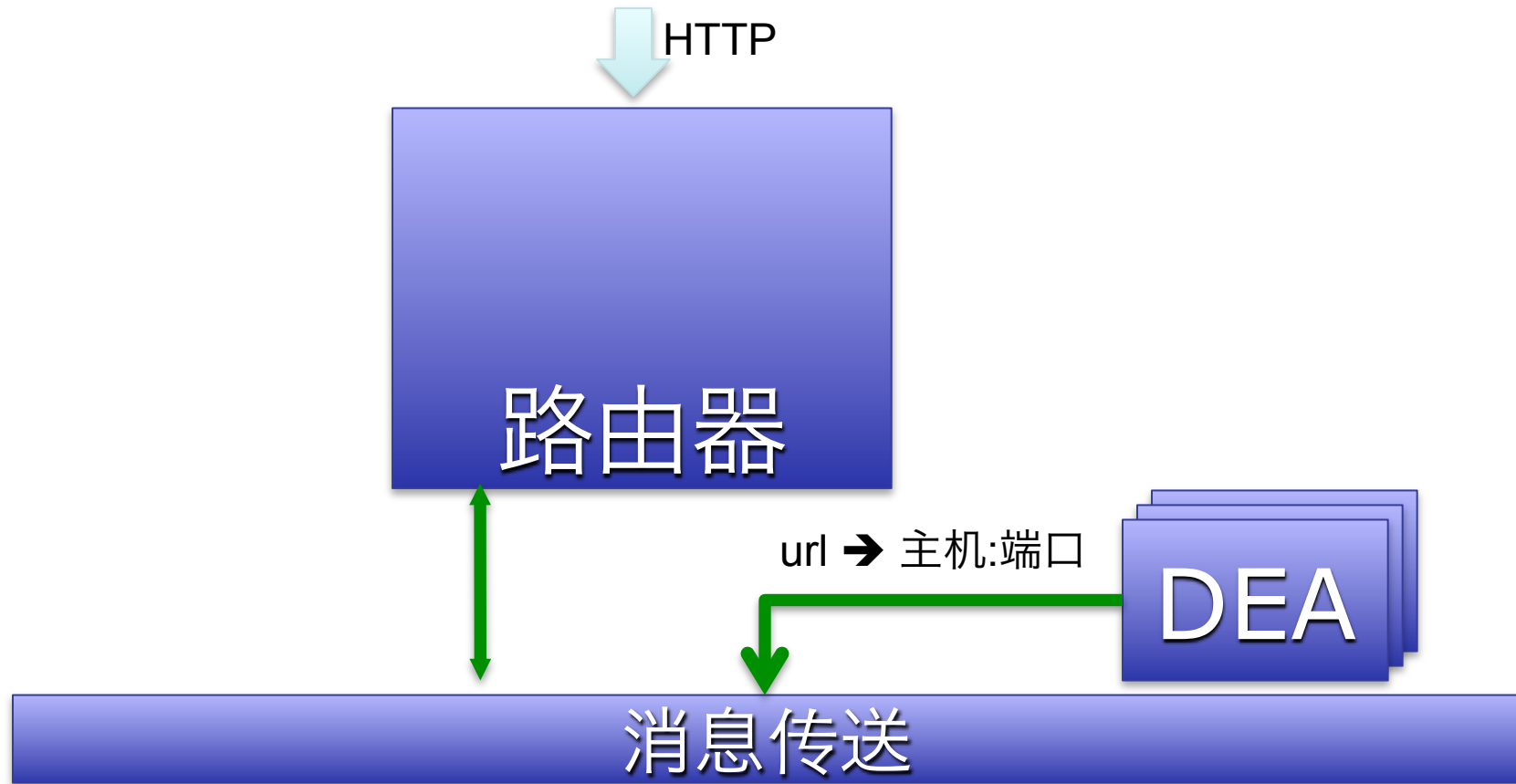
应用程序路由



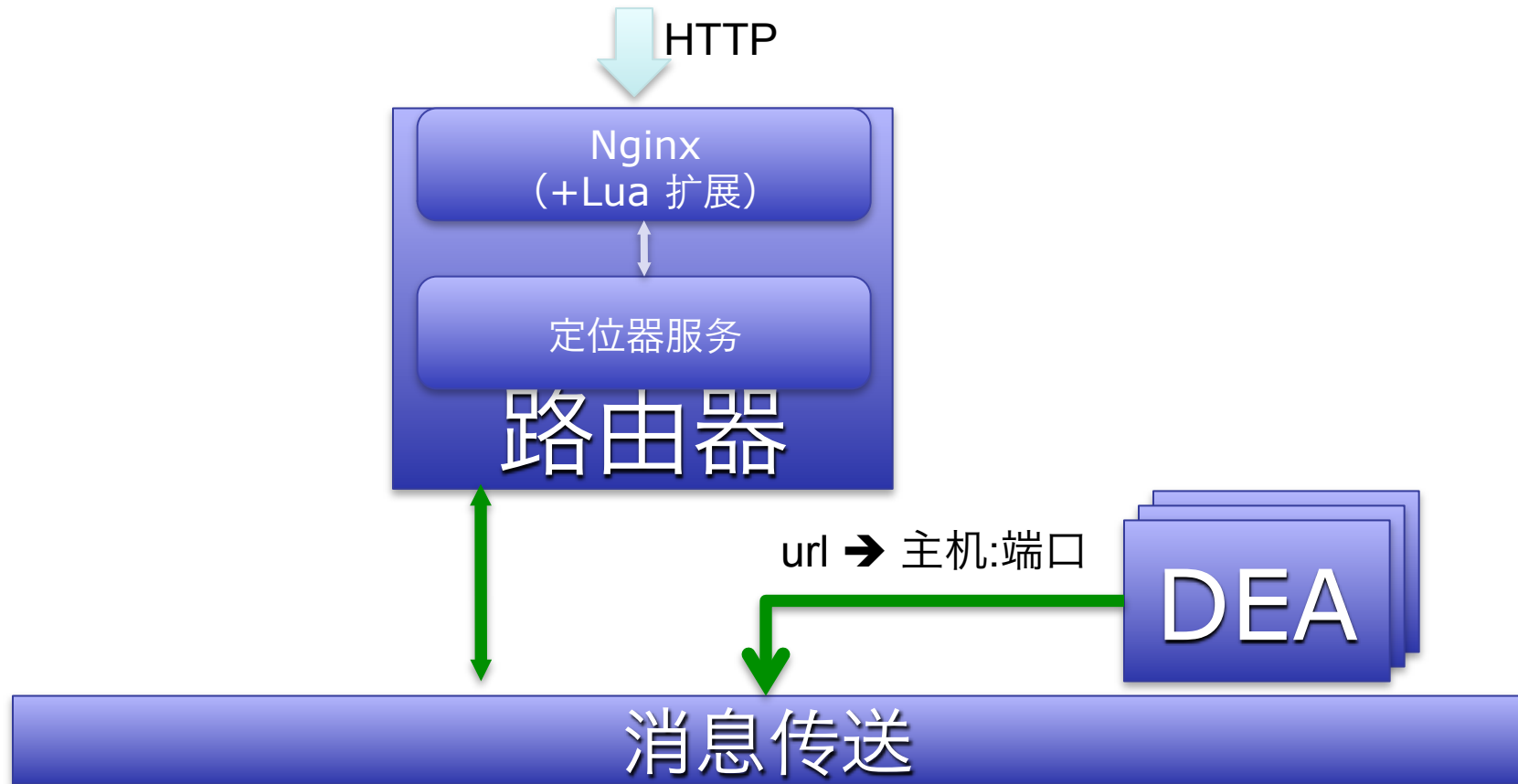
应用程序路由



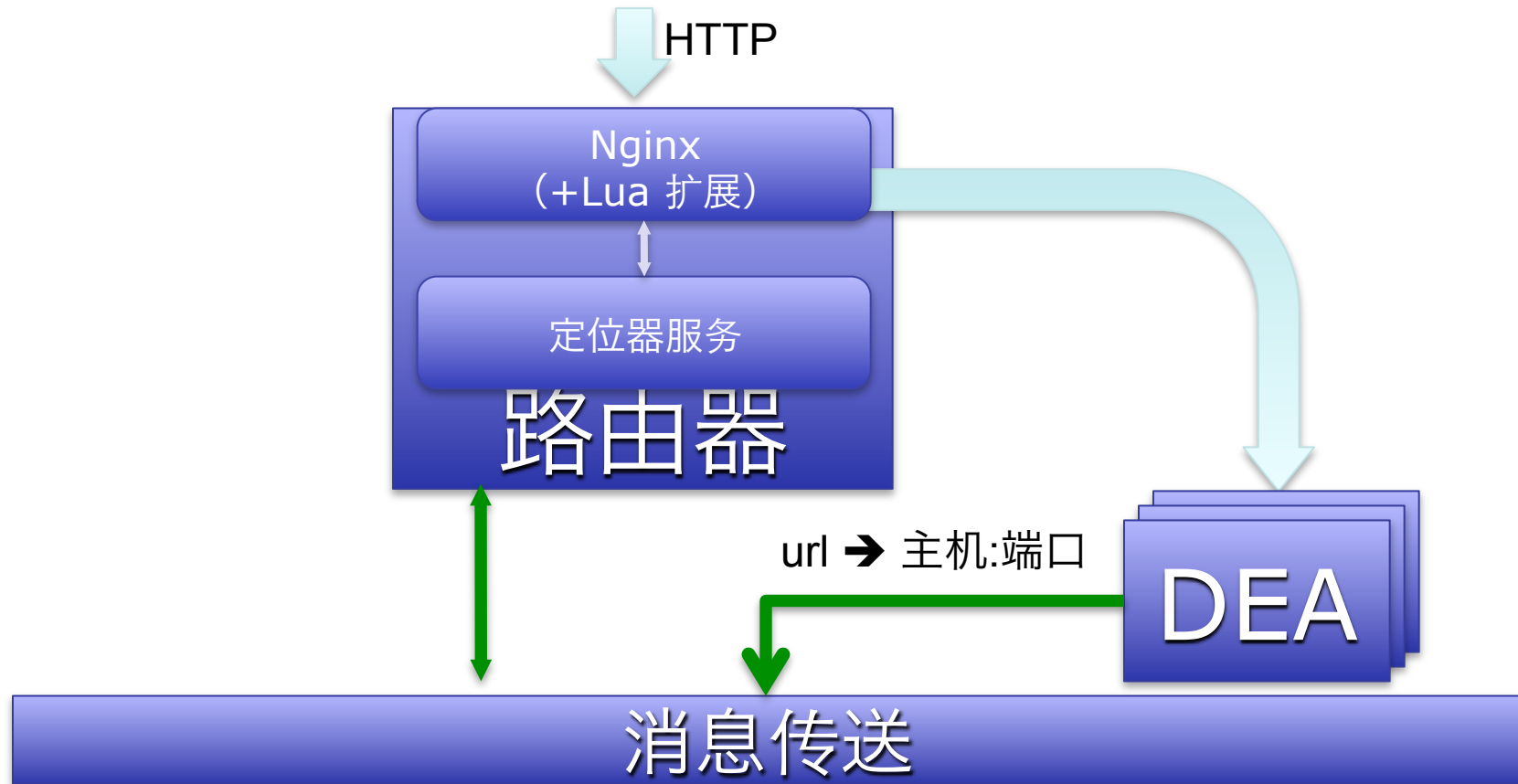
应用程序路由



应用程序路由



应用程序路由

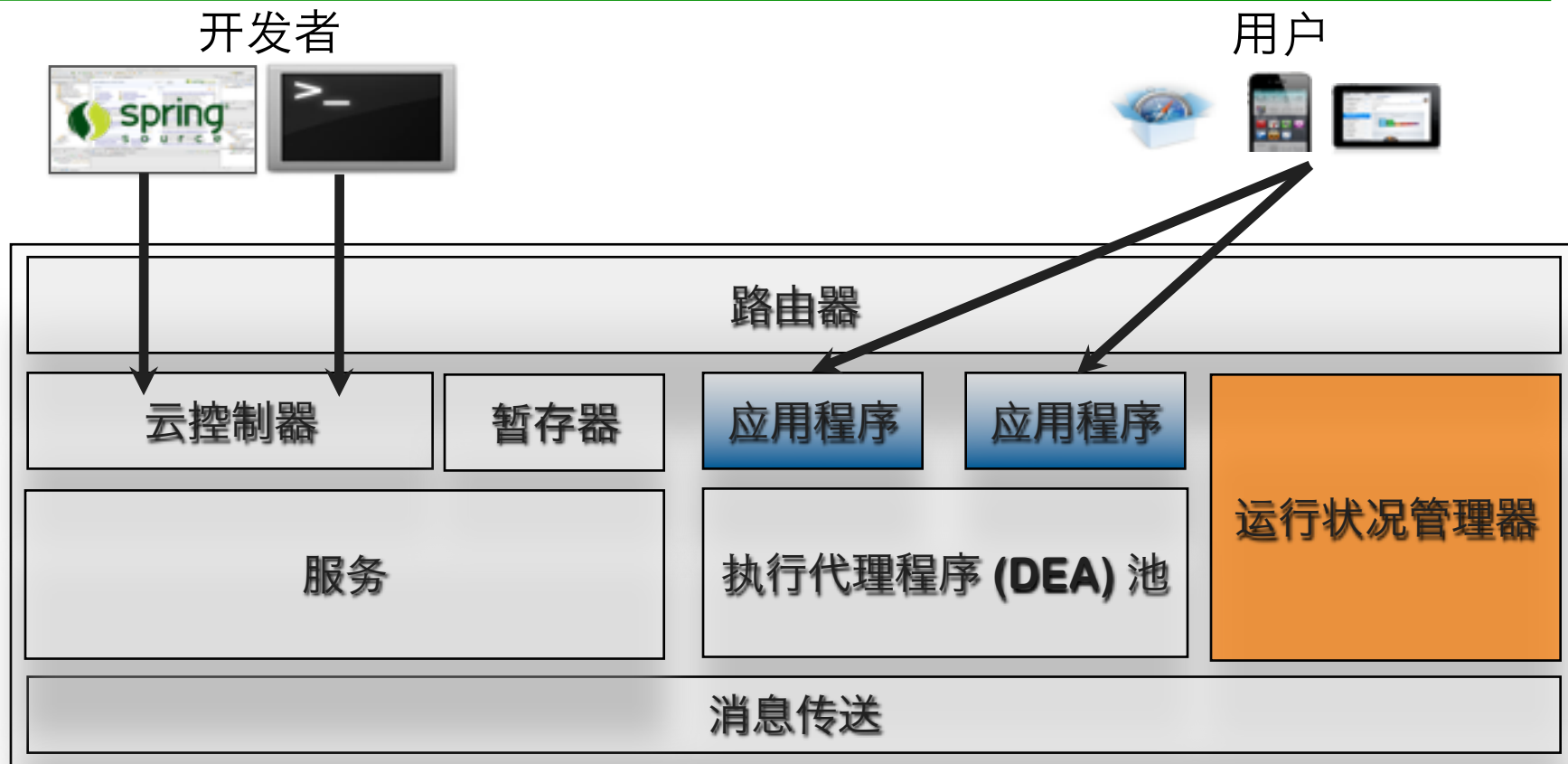




HEALTH

MANAGER

Cloud Foundry 内层及运行



运行状况管理器

- 预期状态
 - 云控制器
- 当前状态：
 - DEAs
- 当前状态 ← 预期状态

A hand is shown interacting with a multi-touch interface on a wall-mounted device. The device has a screen displaying 'FUJITSU' and '00071054', along with some Japanese text and icons. Below the screen is a numeric keypad and several function buttons. The hand is positioned over the screen, with fingers spread, suggesting a multi-touch gesture. The background is a plain wall.

USER

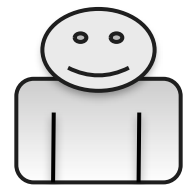
AUTHENTICATION AND AUTHORIZTION

UAA

- 集中化身份管理
 - 可验证来自多个来源的用户身份
 - 可向使用者提供单个标准协议
- 用户帐户管理
- 客户端应用程序注册
- OpenID Connect 和 OAuth2 – 委托的授权
 - 采用 Spring 安全性

使用Oauth2 的 Cloud Foundry

使用Oauth2 的 Cloud Foundry



用户

资源所有者



CF 门户

客户端



UAA

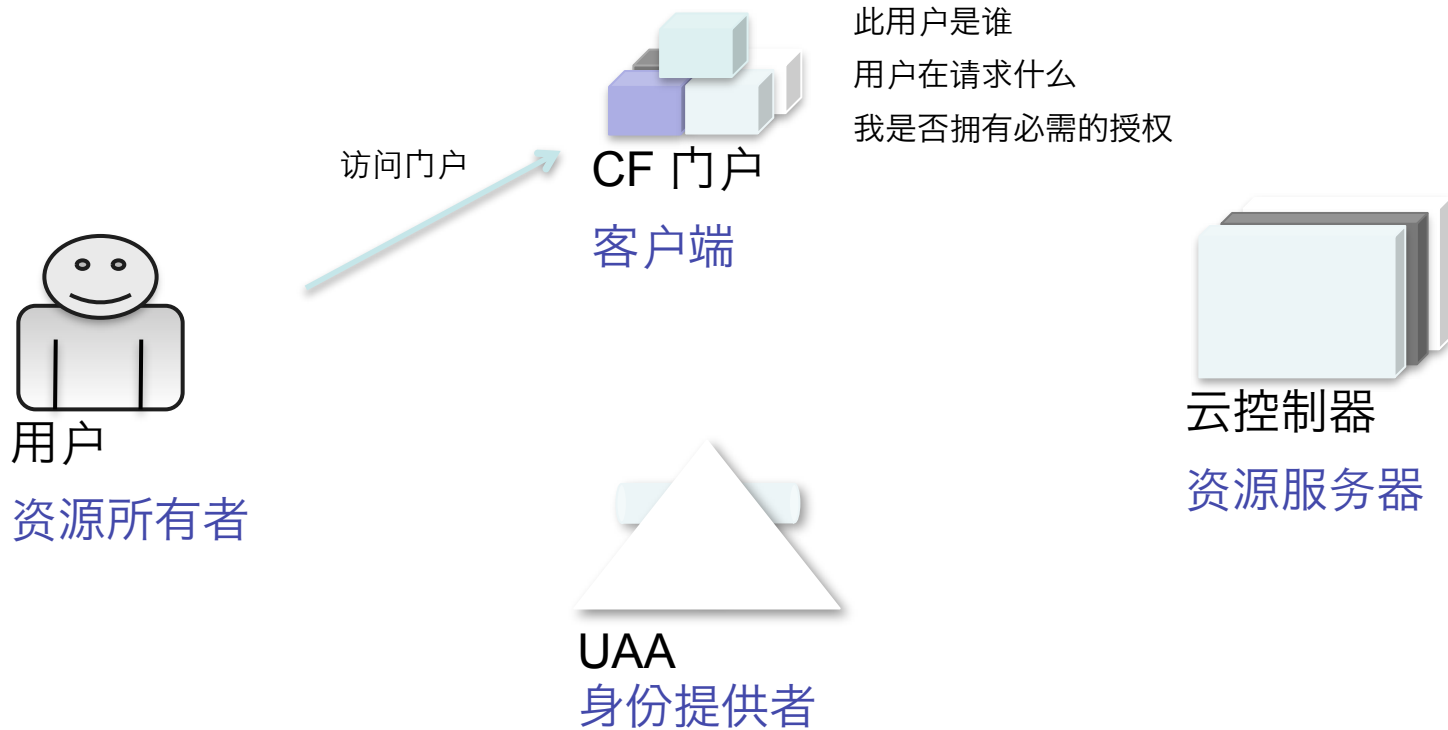
身份提供者



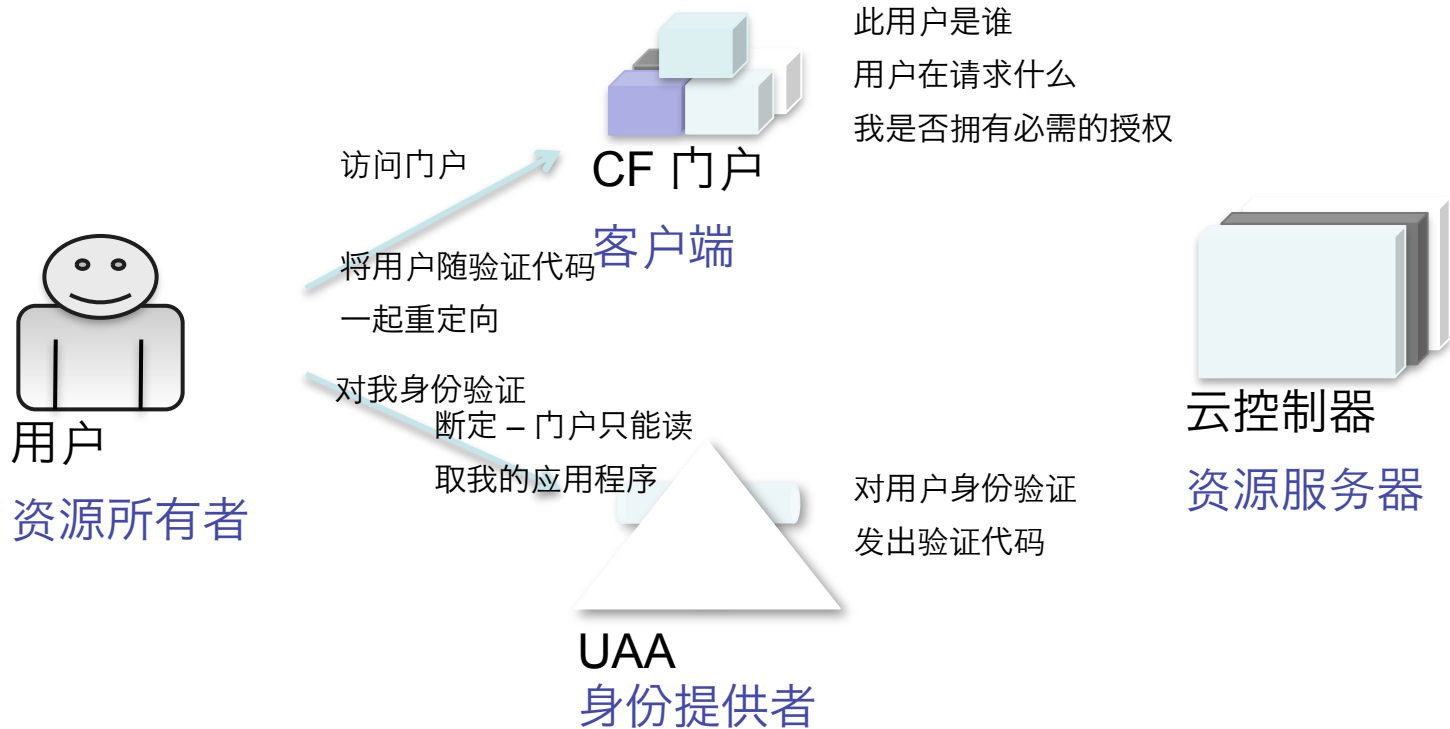
云控制器

资源服务器

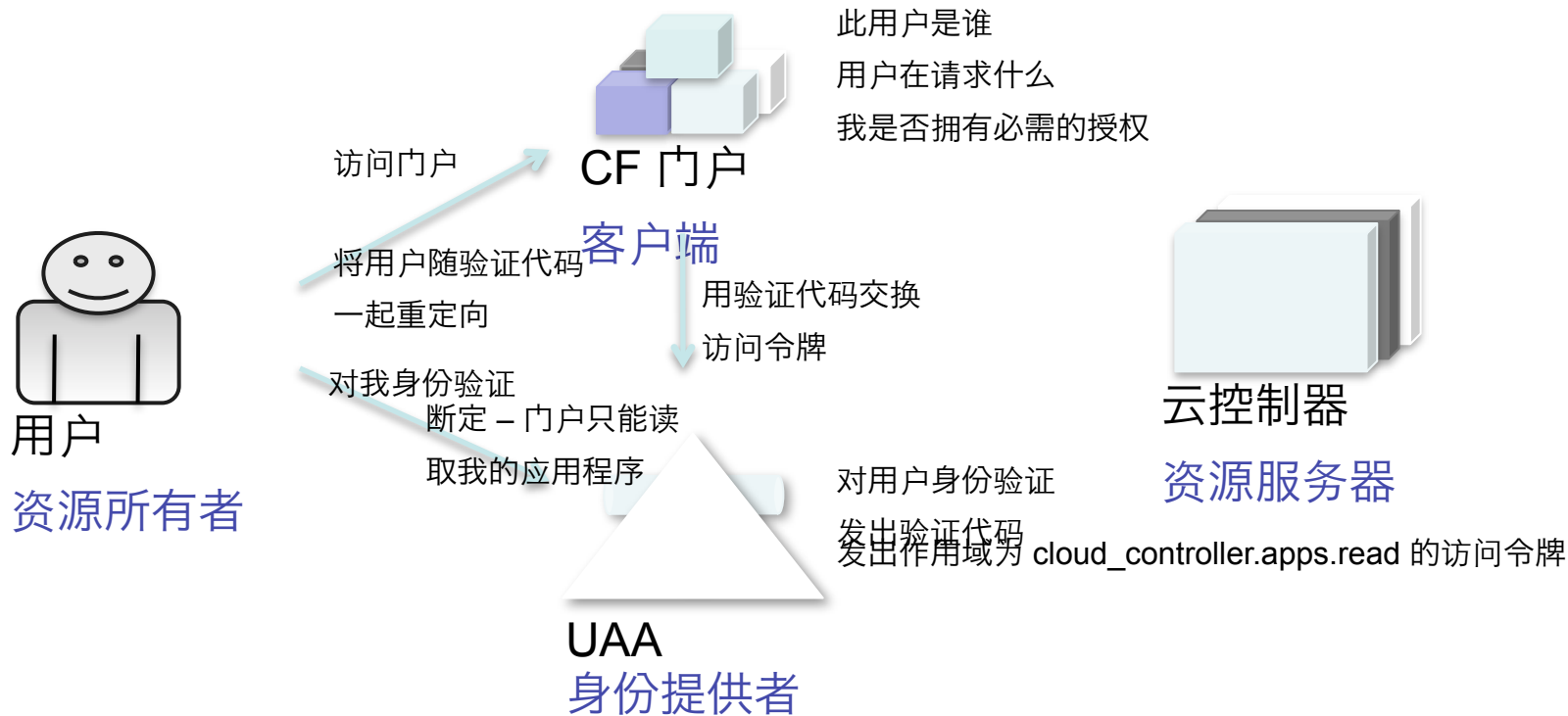
使用Oauth2 的 Cloud Foundry



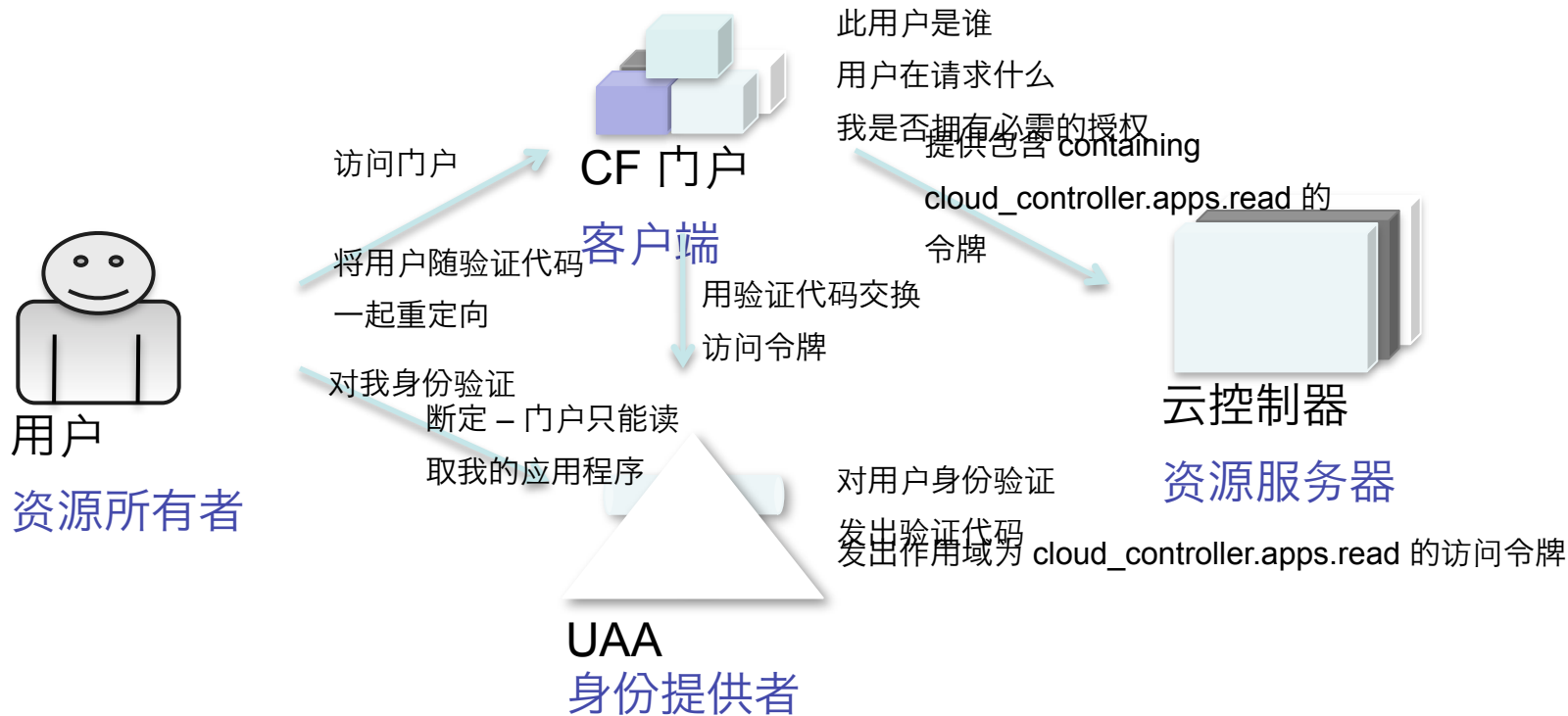
使用Oauth2 的 Cloud Foundry



使用Oauth2 的 Cloud Foundry



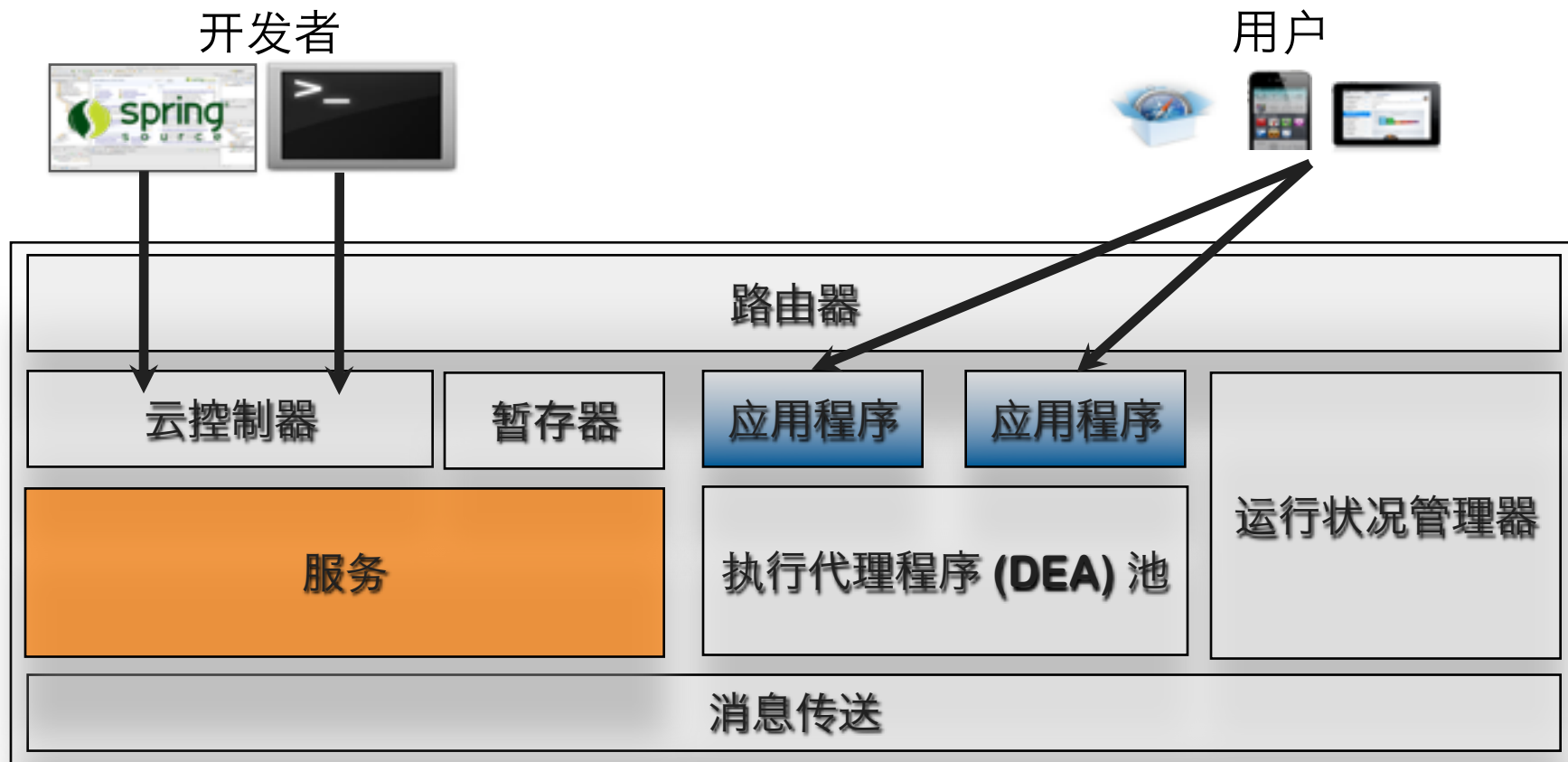
使用Oauth2 的 Cloud Foundry



SERVICE PROVIDER



Cloud Foundry 内存及运行



服务提供

- 服务网关
 - 服务的配置和取消
 - 向云控制器公布服务可用性
- 服务节点
 - 运行实际服务
 - 独立扩展

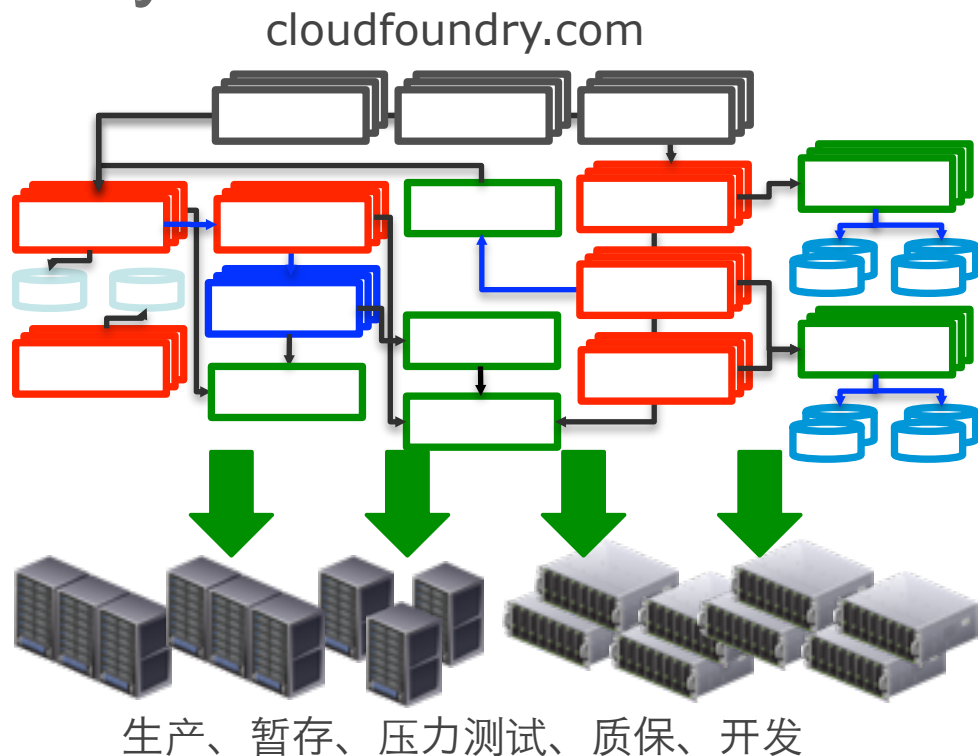
访问服务信息

```
VCAP_SERVICES: {  
  "postgresql-9.0": [{  
    "name": "env-postgresql",  
    "label": "postgresql-9.0",  
    "plan": "free",  
    "credentials": {  
      "name": "de24667f9344b4eeaad6b5a2326d52faa",  
      "host": "172.30.48.122",  
      "port": 5432,  
      "user": "u50ce600bba434bacbc99e034bb415644",  
      "password": "pf4dca5bd449d4732841f0c4ae3f299d0"  
    }  
  ]  
}
```

Cloud Foundry 的运行 外层

生产级别的 Cloud Foundry 群集

- 500 – 5,000 个虚拟机
- 超过 40 个独特节点类型
- 超过 75 个独特软件包
- 每星期 2 次 cf.com 更新
- 24x7x365 不间断运行
- 无中断时间的部署
- 可靠、稳健、可重复的部署、更新和功能调整



什么是 BOSH ?

github.com/cloudfoundry/bosh

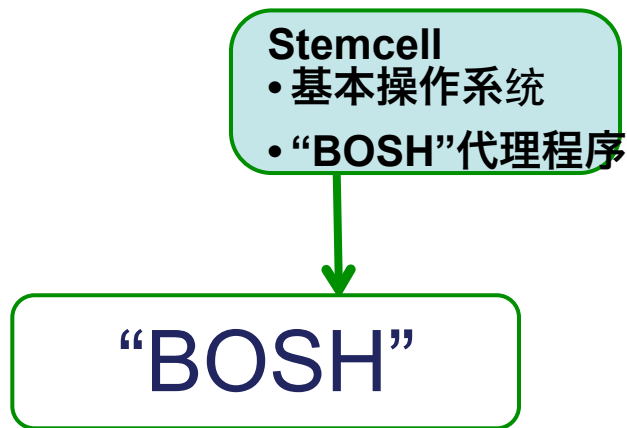
在生产环境中运行 Cloud Foundry 的服务演进技术

- 自动化虚拟基础结构
 - 云规模
 - 使用云提供商接口 (CPI) 进行抽象化
- 软件部署、配置和更新
 - 为最大限度减少中断时间而进行优化
 - 支持多个虚拟机角色
- 可重复的流程
 - 通过版本控制进行发行版管理
- 活动的监控和警报
- BOSH 自动发布以来就一直用于运行 CloudFoundry.com

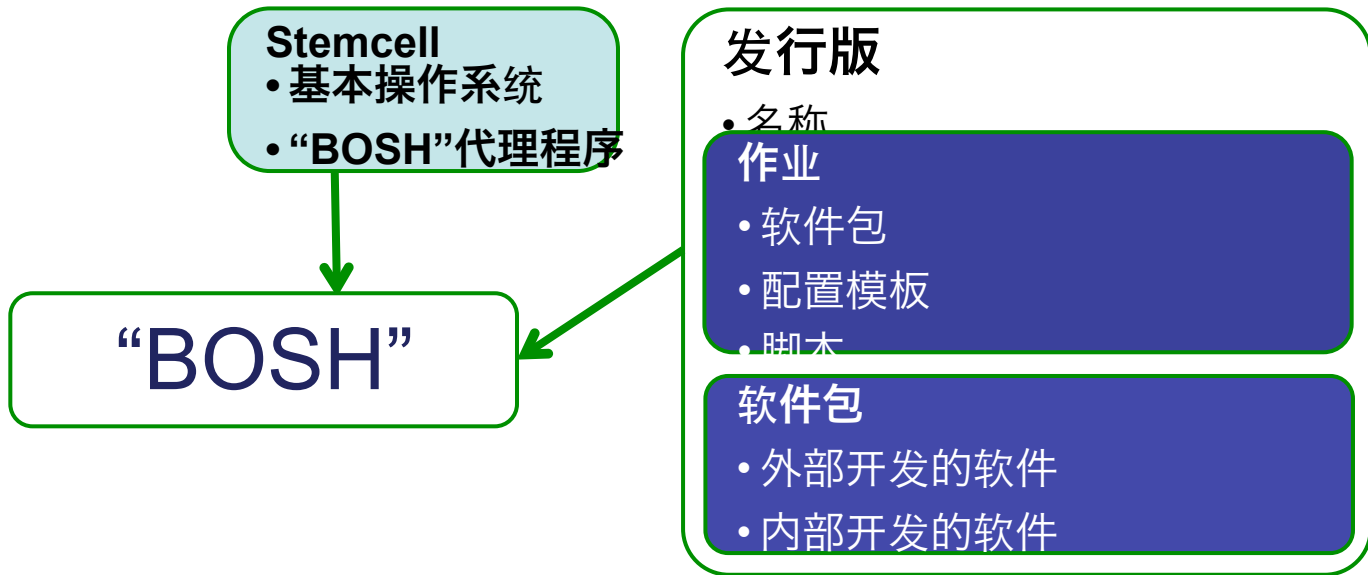
Cloud Foundry “BOSH” – 概念

“BOSH”

Cloud Foundry “BOSH” – 概念



Cloud Foundry “BOSH” – 概念



Cloud Foundry “BOSH” – 概念

部署清单

- 发行版名称和版本
- 每个作业的虚拟机数和参数
- 要使用的 Stemcell 数

Stemcell

- 基本操作系统
- “BOSH”代理程序

发行版

- 名称

作业

- 软件包
- 配置模板
- 脚本

软件包

- 外部开发的软件
- 内部开发的软件

“BOSH”

```
graph TD; A[部署清单] --> D[“BOSH”]; B[Stemcell] --> D; C[发行版] --> D;
```

Cloud Foundry “BOSH” – 概念

部署清单

- 发行版名称和版本
- 每个作业的虚拟机数和参数
- 要使用的 Stemcell 数

Stemcell

- 基本操作系统
- “BOSH”代理程序

发行版

- 名称

作业

- 软件包
- 配置模板
- 脚本

软件包

- 外部开发的软件
- 内部开发的软件

“BOSH”

```
graph TD; A[部署清单] --> D[“BOSH”]; B[Stemcell] --> D; C[发行版] --> D; D --> E[ ]; style E fill:none,stroke:none
```

Cloud Foundry “BOSH” – 概念

部署清单

- 发行版名称和版本
- 每个作业的虚拟机数和参数
- 要使用的 Stemcell 数

Stemcell

- 基本操作系统
- “BOSH”代理程序

发行版

- 名称

作业

- 软件包
- 配置模板
- 脚本

软件包

- 外部开发的软件
- 内部开发的软件

“BOSH”

环境

配置

软件包

Stemcell

回滚无状态组件的更新

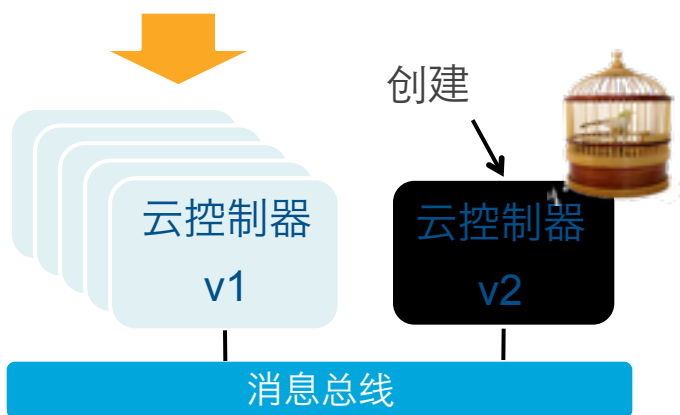
入站 HTTP 请求

入站 HTTP 请求

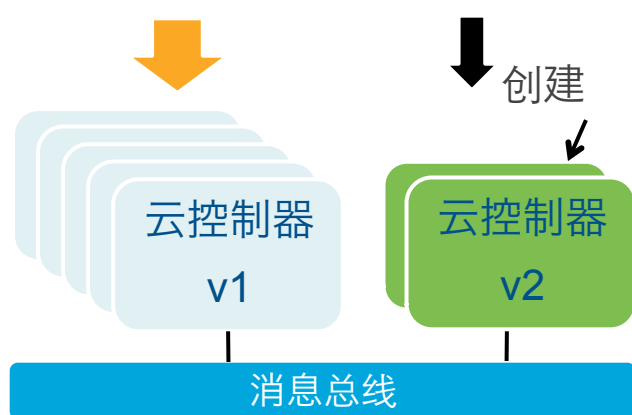
入站 HTTP 请求



用 v1 虚拟机开始



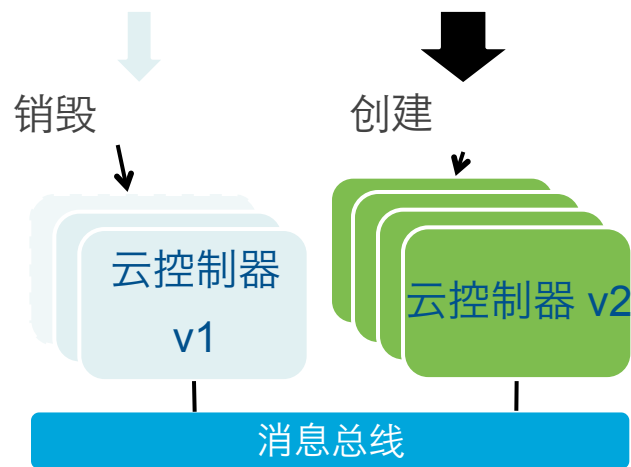
部署 v2 虚拟机 (canary)



如果顺行运行，则添加更多 v2 虚拟机...

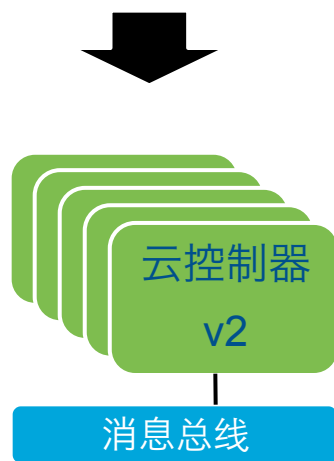
示例：回滚无状态组件的更新

入站 HTTP 请求



... 同时移除 v1 虚拟机...

入站 HTTP 请求



... 直到所有虚拟机都是
v2

最终结果：

我们通过构建新的虚拟机并销毁旧的虚拟机，从 v1 升级到 v2，且没有任何中断时间

部署

- 平行
- Canaries
- 最终一致性

```
1, zsh
~ > bosh deploy
Getting deployment properties from director...
Compiling deployment manifest...
Detecting changes in deployment...

Releases
cf-dev-oleg
  changed version:
    - 82.5-dev
    + 82.6-dev
Release version has changed: 82.5-dev -> 82.6-dev
Are you sure you want to deploy this version? (type 'yes' to continue): yes

Compilation
No changes

Update
No changes

Resource pools
No changes

Networks
No changes

Jobs
mysql_node
  changed instances:
    - 2
    + 4

Properties
No changes

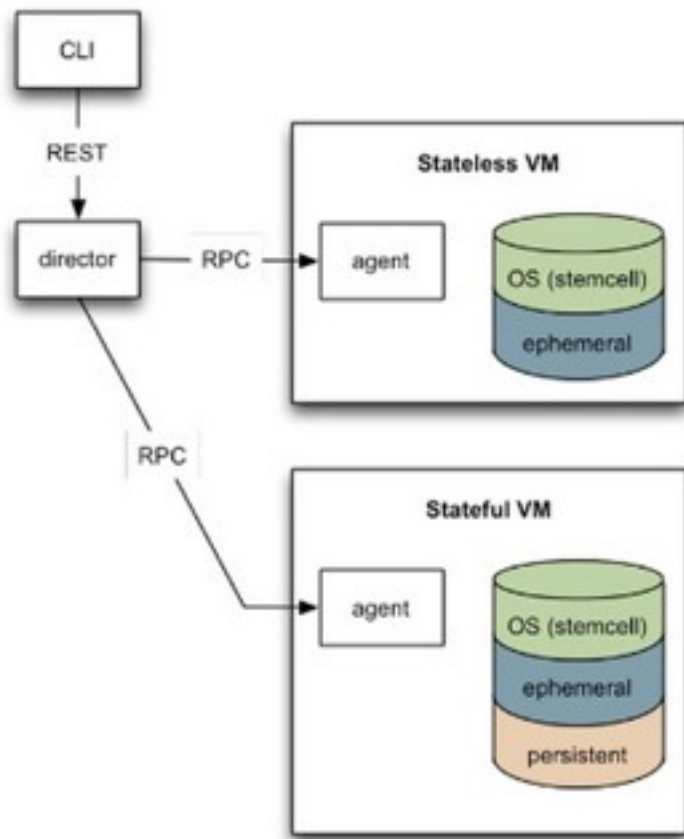
Please review all changes carefully
Deploying 'dev57.yml' to 'dev57' (type 'yes' to continue): yes
```

内部结构

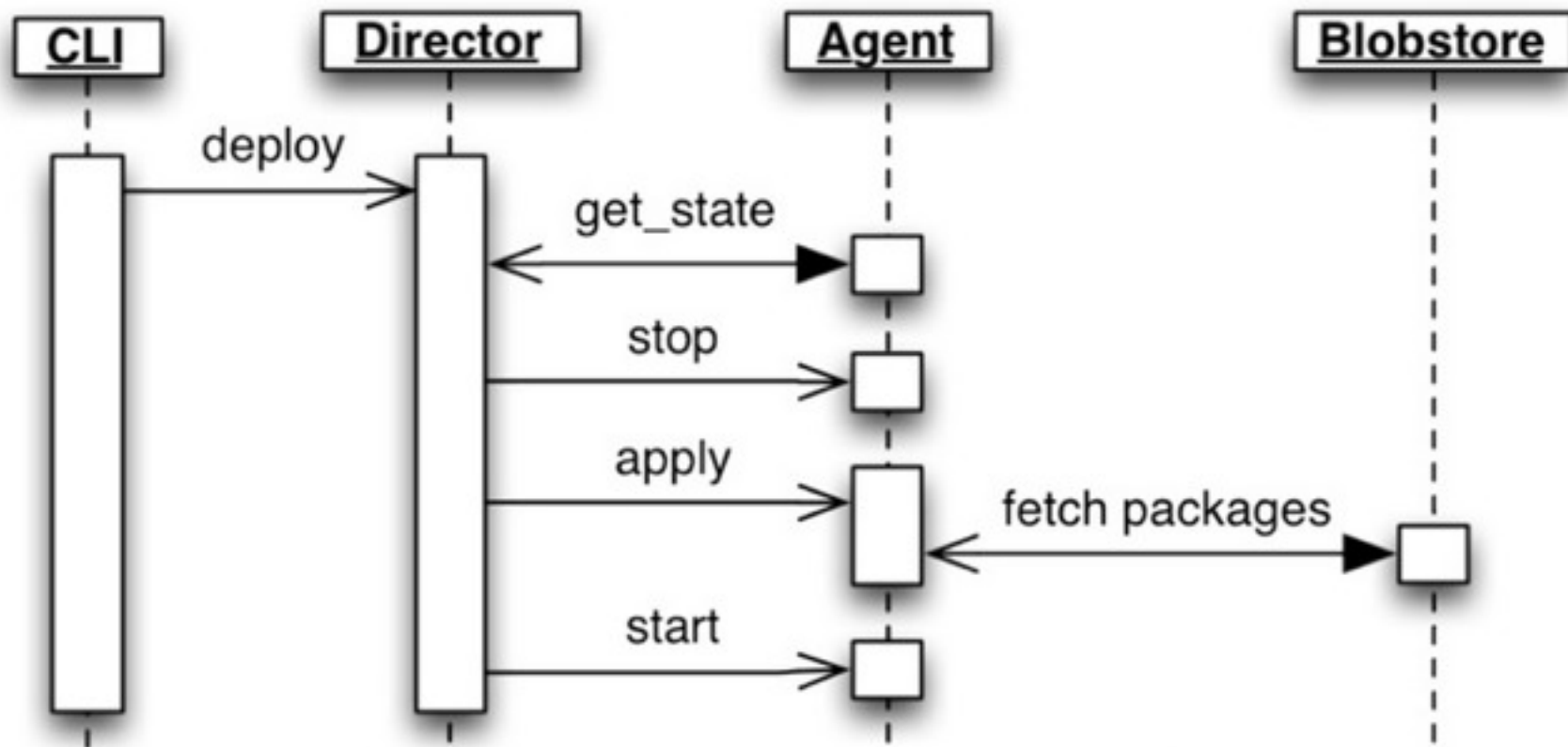
■ 组件

- 控制器
- 代理程序
- CLI

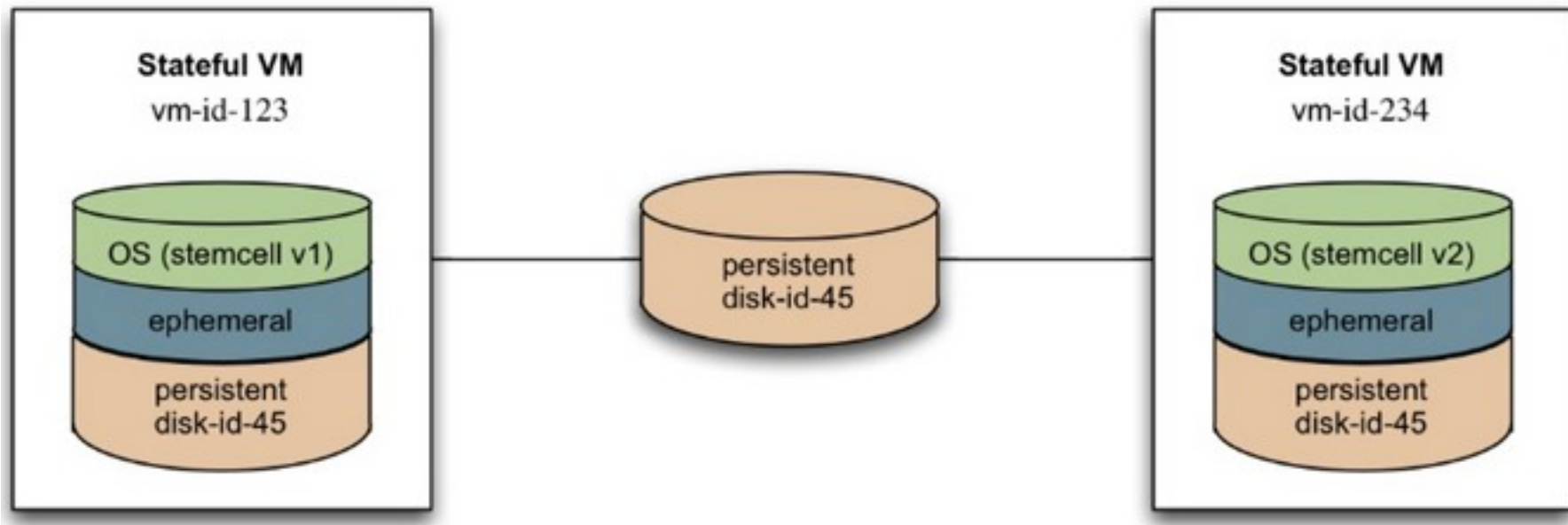
■ Stemcell



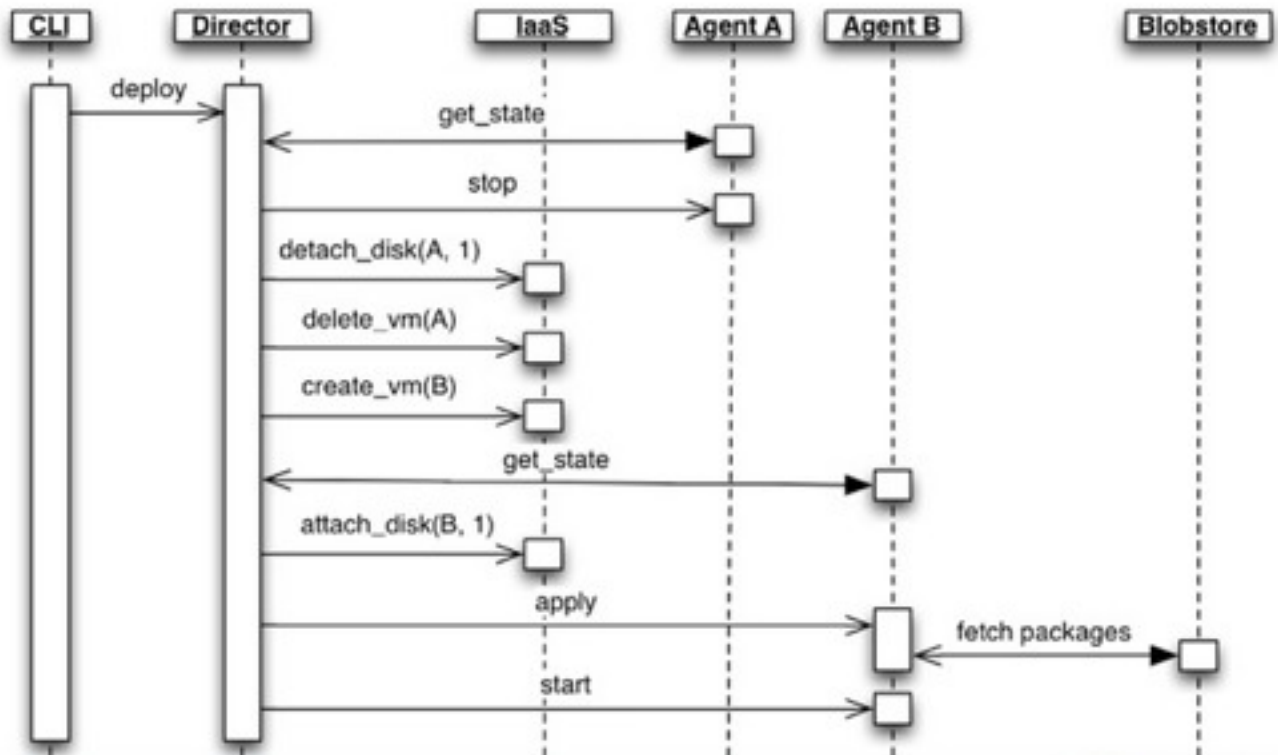
代码更改



虚拟机更新



虚拟机更新



云提供商接口

- 基础设施抽象
 - Stemcell
 - 虚拟机模板
 - 虚拟机
 - 磁盘
 - 网络

```
cpi.rb
1  class Cloud
2    def create_stemcell(image_path, cloud_properties)
3      end
4
5    def delete_stemcell(stemcell_id)
6      end
7
8    def create_vm(agent_id, stemcell_id, resource_pool,
9      end
10
11    def delete_vm(vm_id)
12      end
13
14    def reboot_vm(vm_id)
15      end
16
17    def configure_networks(vm_id, networks)
18      end
19
20    def create_disk(size, vm_locality = nil)
21      end
22
23    def delete_disk(disk_id)
24      end
25
26    def attach_disk(vm_id, disk_id)
27      end
28
29    def detach_disk(vm_id, disk_id)
30      end
31    end
32  end
```

源自设计的 IaaS 独立性

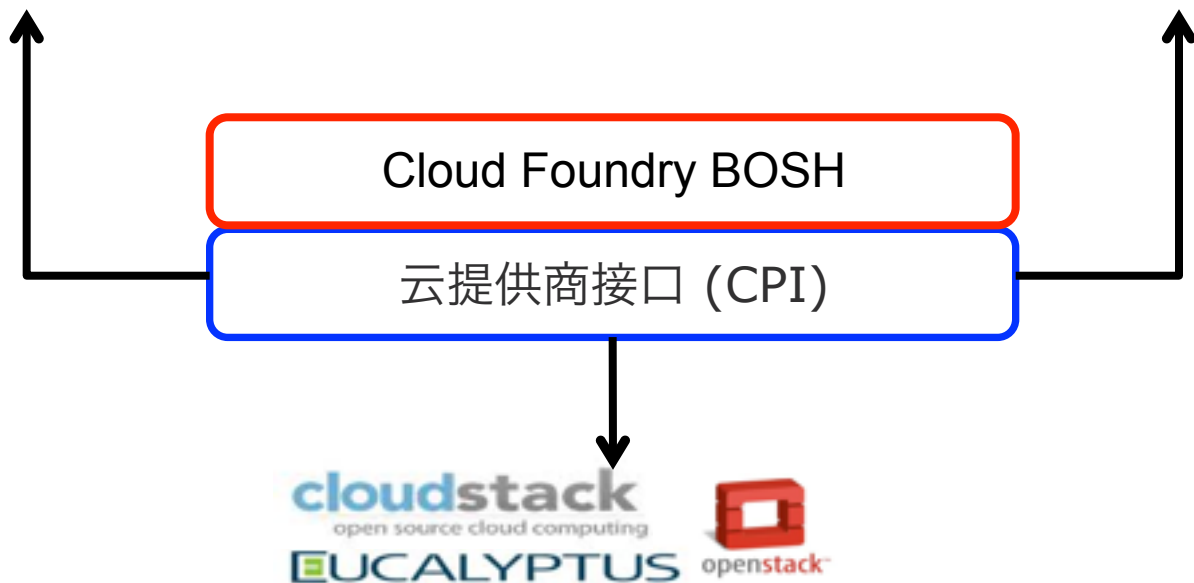


vSphere：历经实战考验，数千个部署实例

vCloud Director：“工作进展中”，2012 年下半年



CPI：代码完整功能状态“工作进展中”



contribute: github.com/cloudfoundry/bosh

CPI 提供商

vmware®

Piston
PISTON {CLOUD} COMPUTING

 **amazon**
webservices™


openstack™
CLOUD SOFTWARE

问题？

Cloud Foundry 启动营

在www.cloudfoundry.com注册账号并成功上传应用程序,
即可于12月8日中午后凭账号ID和应用URL到签到处换取Cloud Foundry主题卫衣一件。



iPhone5 等你拿

第二天大会结束前，请不要提前离开，将填写完整的意见反馈表投到签到处的抽奖箱内，即可参与“iPhone5”抽奖活动。



Birds of a Feather 专家面对面

所有讲师都会在课程结束后，到紫兰厅与来宾讨论课程上的问题

