

2015
安全应用超市@Docker



行盼宁XX



CONTENTS

目录

背景及意义

架构设计与实现

系统流程与展示

未来规划



CONTENTS

目录

背景及意义

架构设计与实现

系统流程与展示

未来规划

01

背景及意义



传统安全应用交付模式

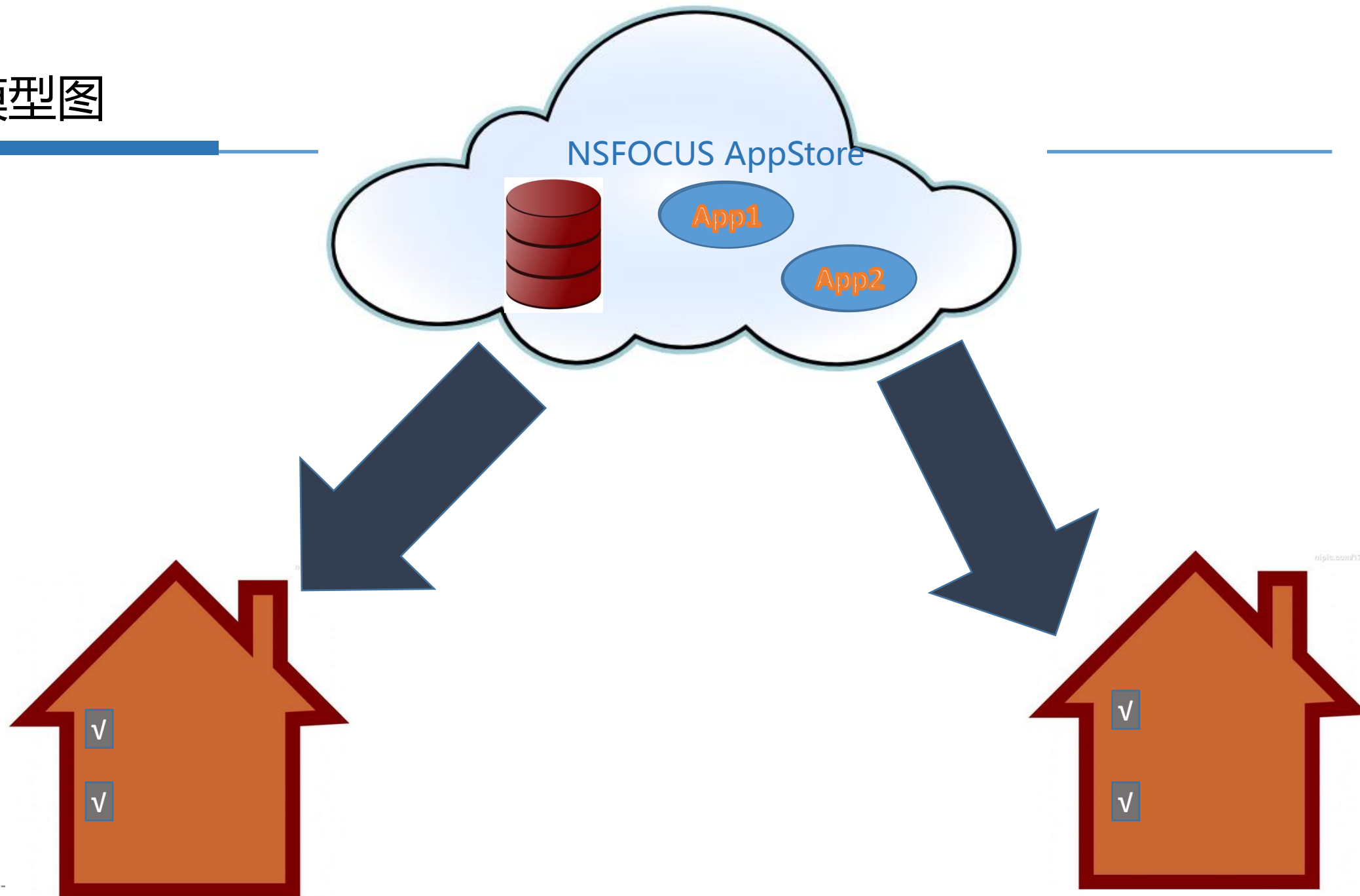
- 硬件设备模式
- 服务咨询模式



新型安全产品交付模式

- 安全产品SaaS化
- 类AppStore模式

模型图





CONTENTS

目录

背景及意义

架构设计与实现

系统流程与展示

未来规划

02

架构设计与实现

A

架构设计

- Security AppStore
- Client AppStore

B

Docker技术简介

- 简介
- 应用场景

C

架构实现

- Docker
- Orchestration
- UI

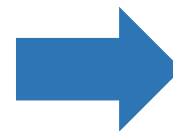
02

架构设计与实现



架构设计

- Security AppStore
- Client AppStore



Docker技术简介

- 简介
- 应用场景

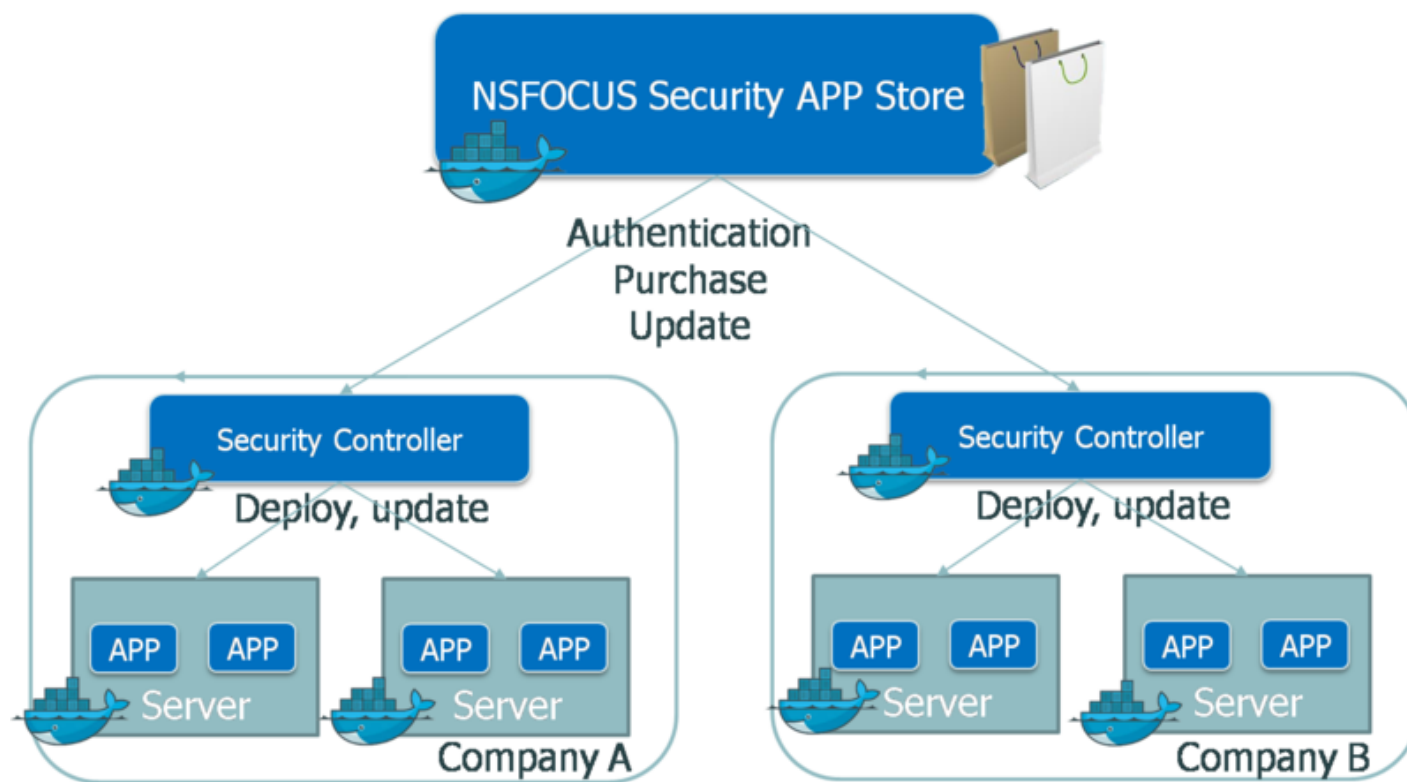


架构实现

- Docker
- Orchestration
- UI

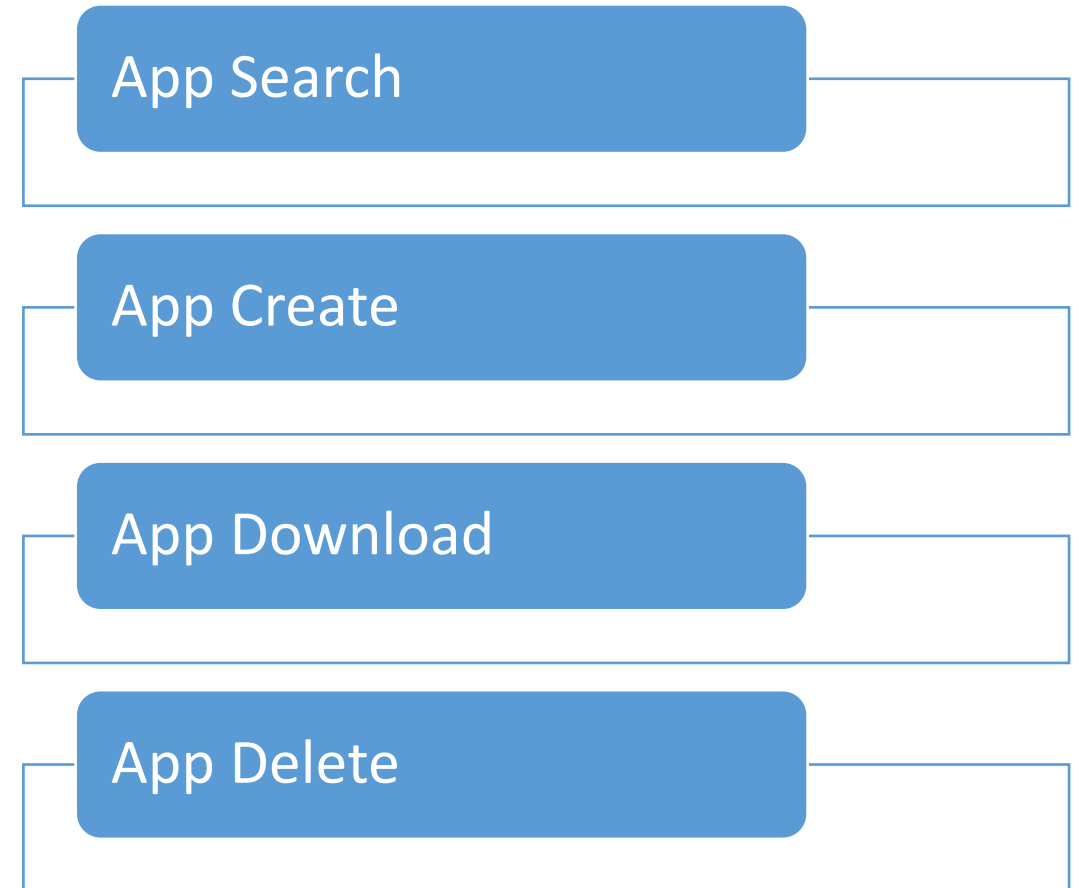
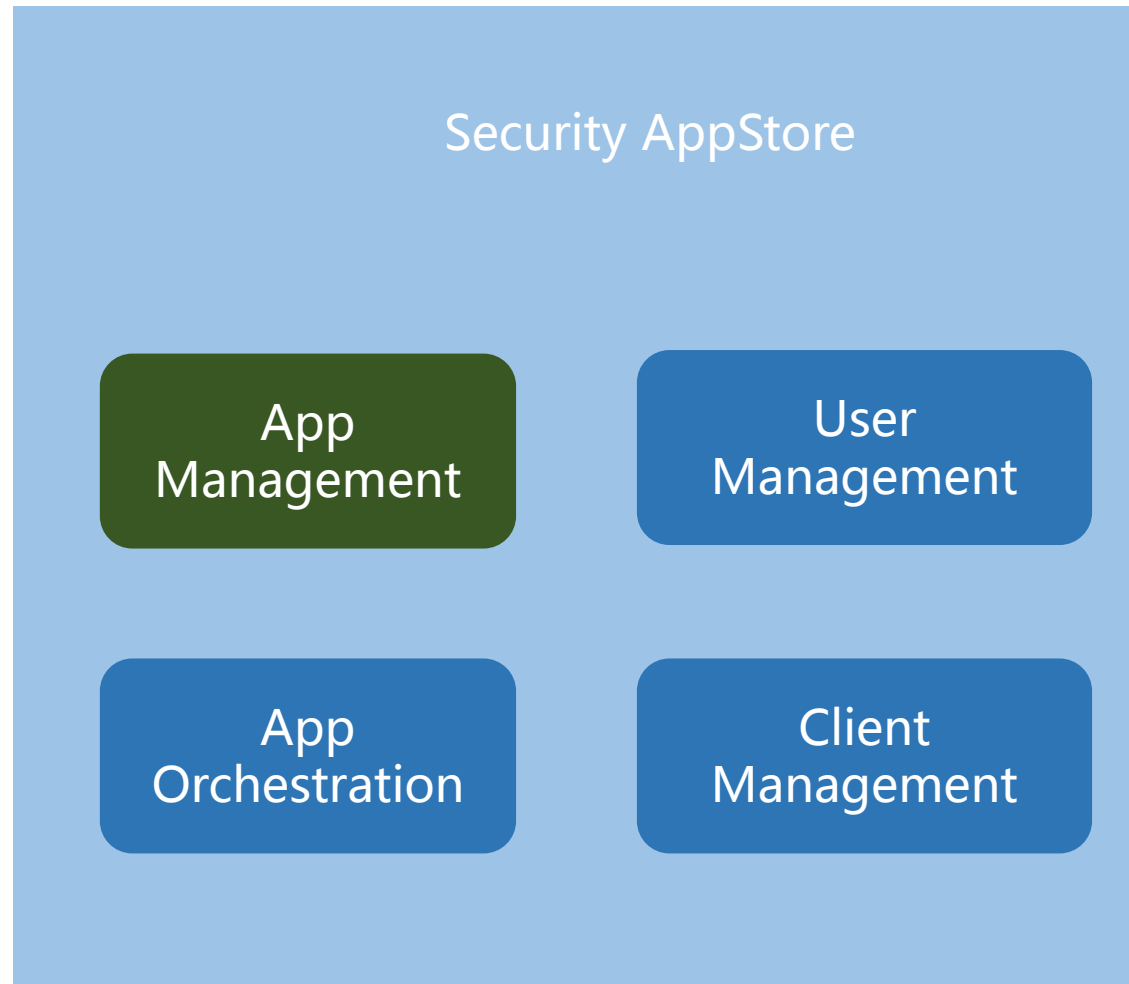


架构设计-架构图



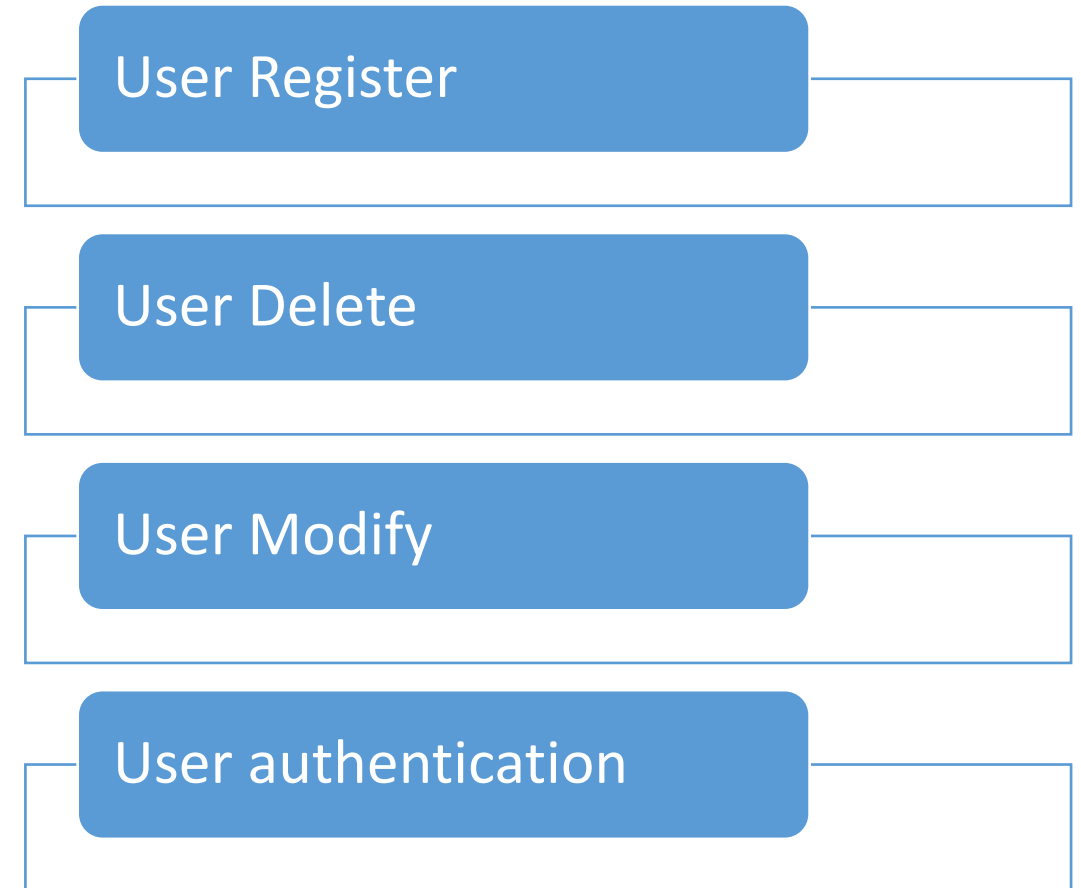
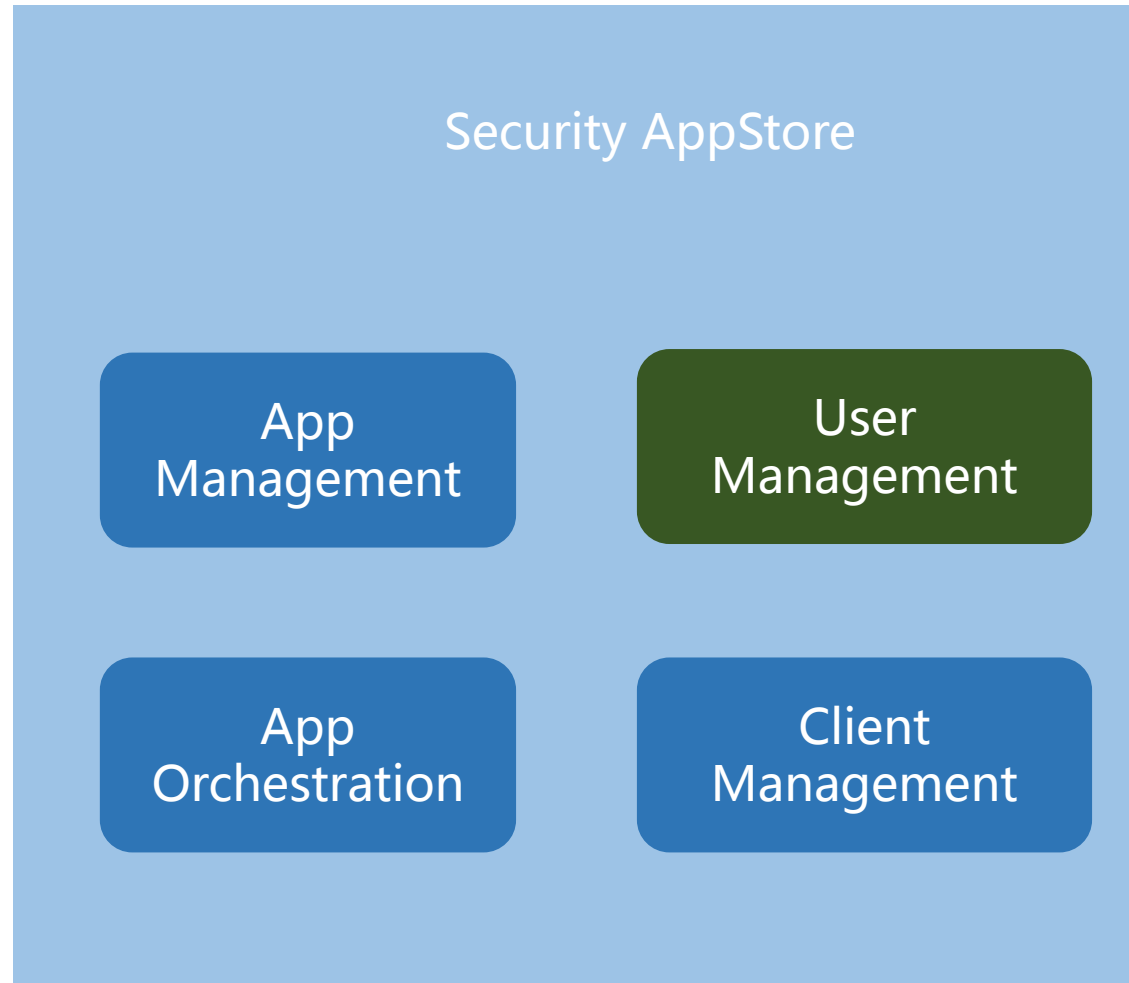


架构设计- Security AppStore



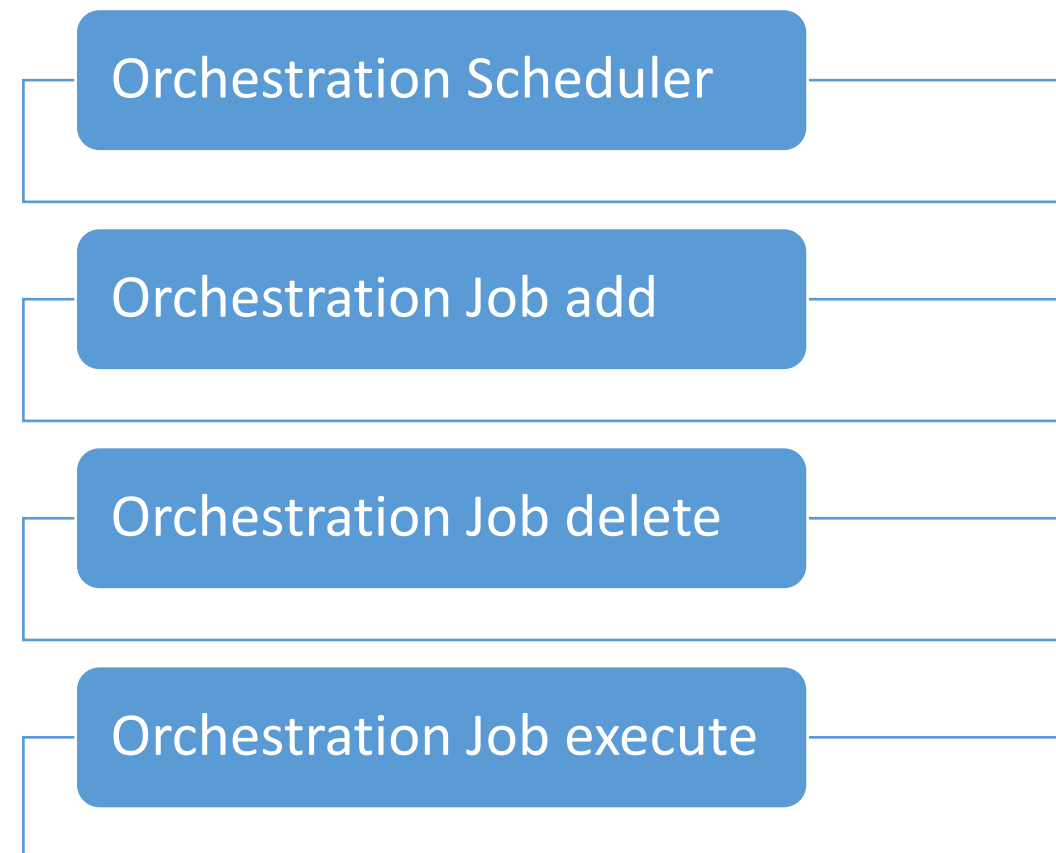
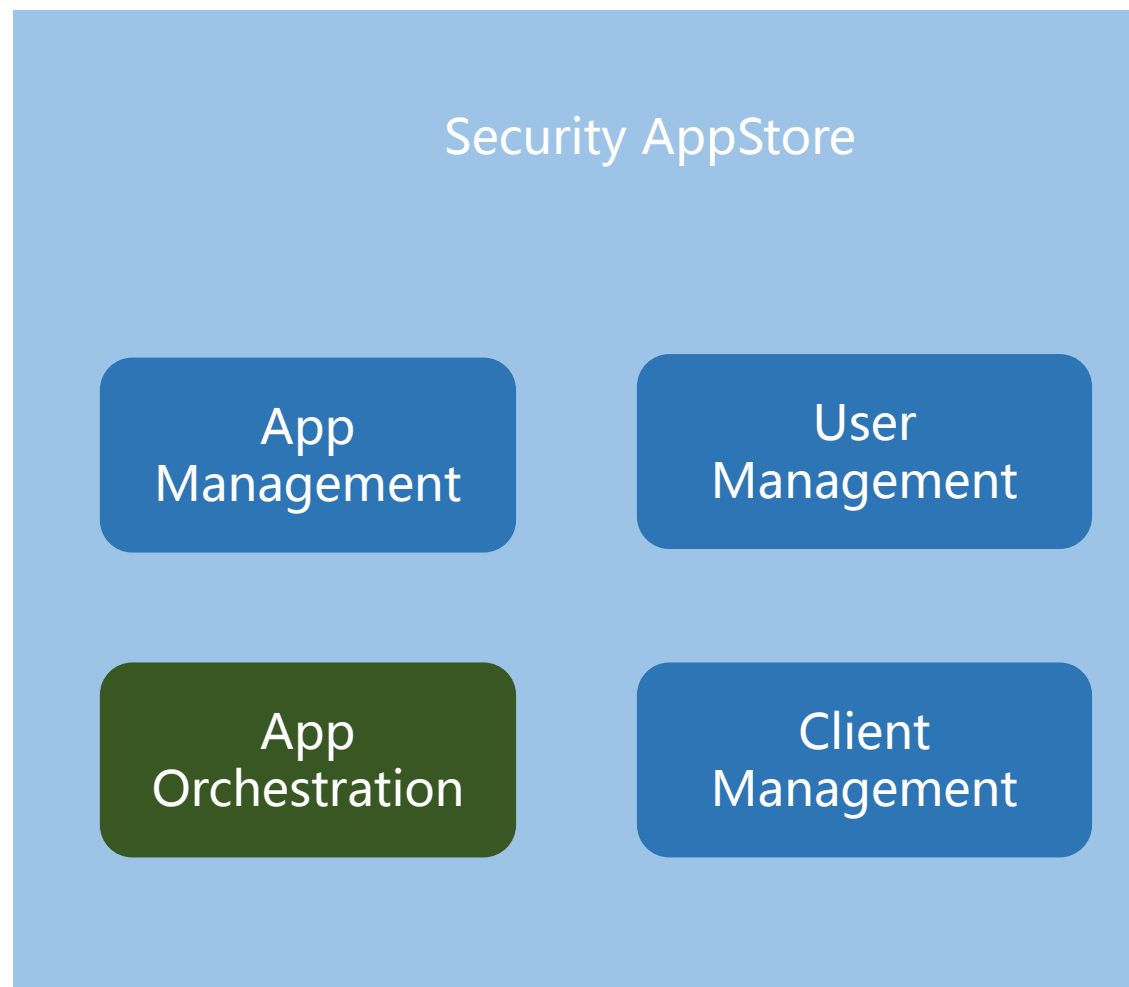


架构设计- Security AppStore



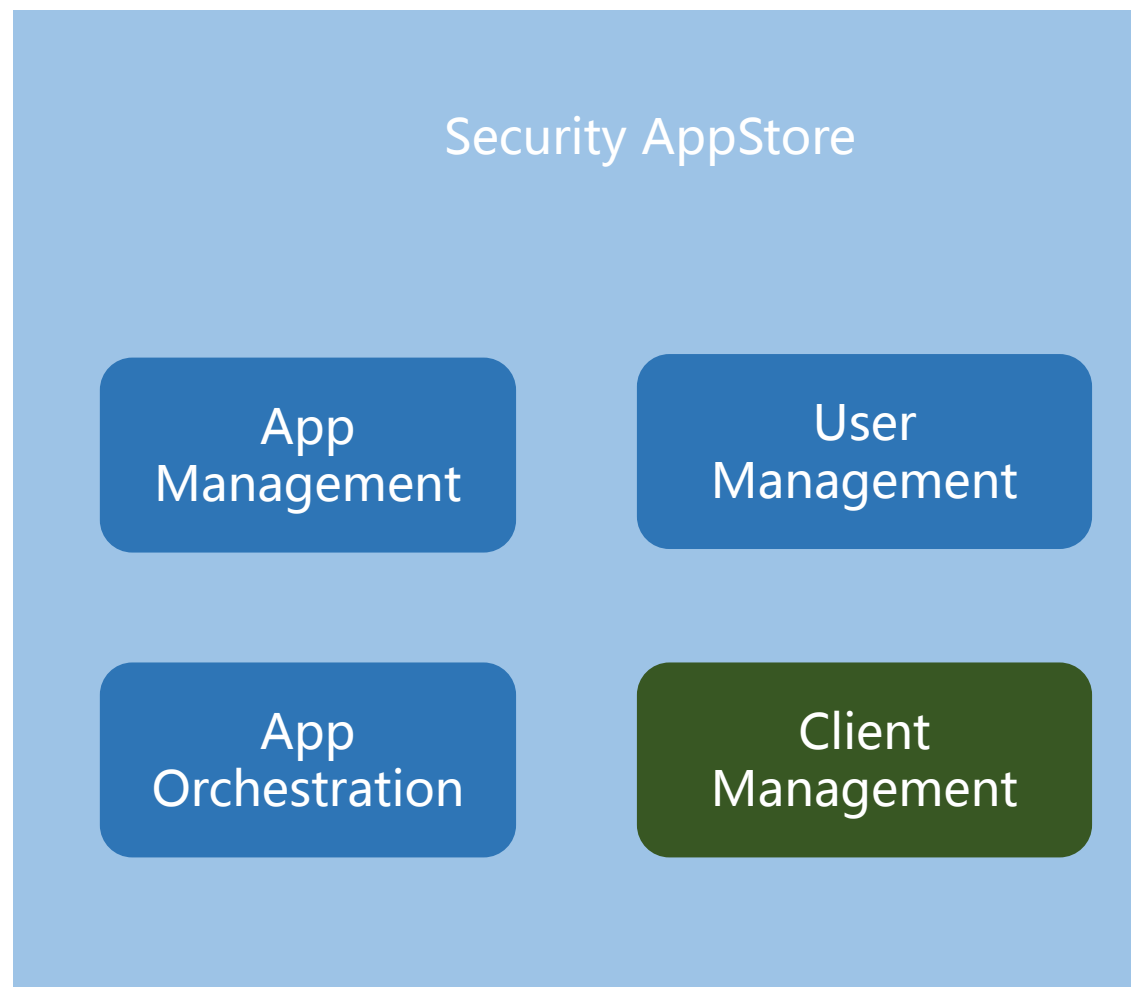


架构设计- Security AppStore



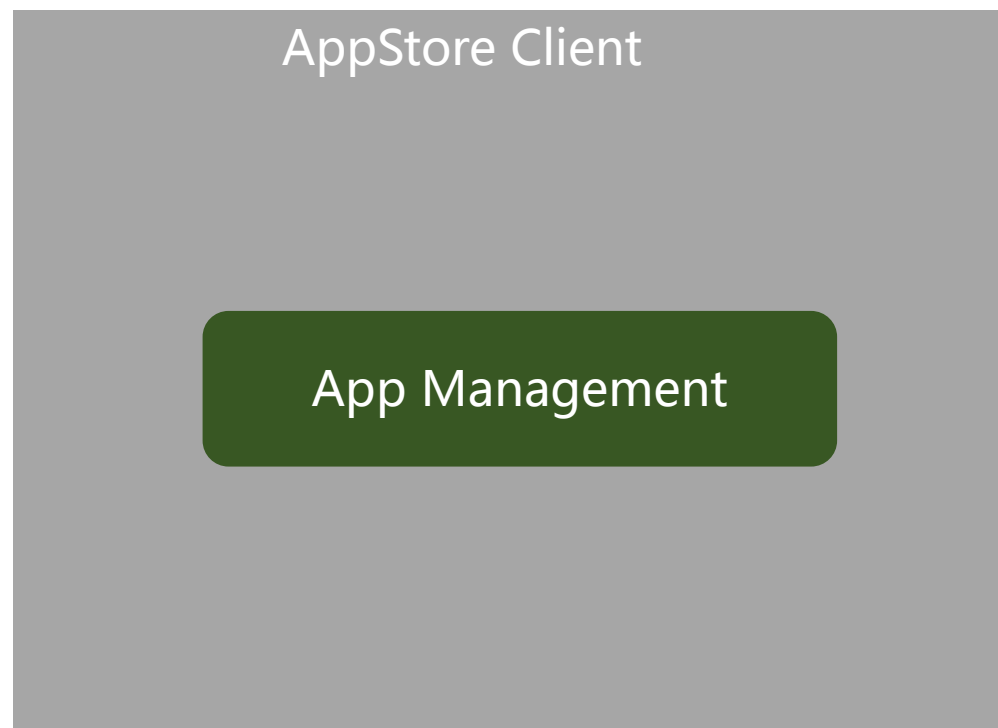


架构设计- Security AppStore





架构设计-AppStore Client (SC)



App Search (Local)

App Download(from remote)

App Update/Deploy

App Operation

02

架构设计与实现



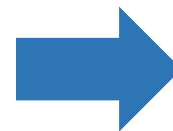
架构设计

- Security AppStore
- Client AppStore



Docker技术简介

- 简介
- 应用场景

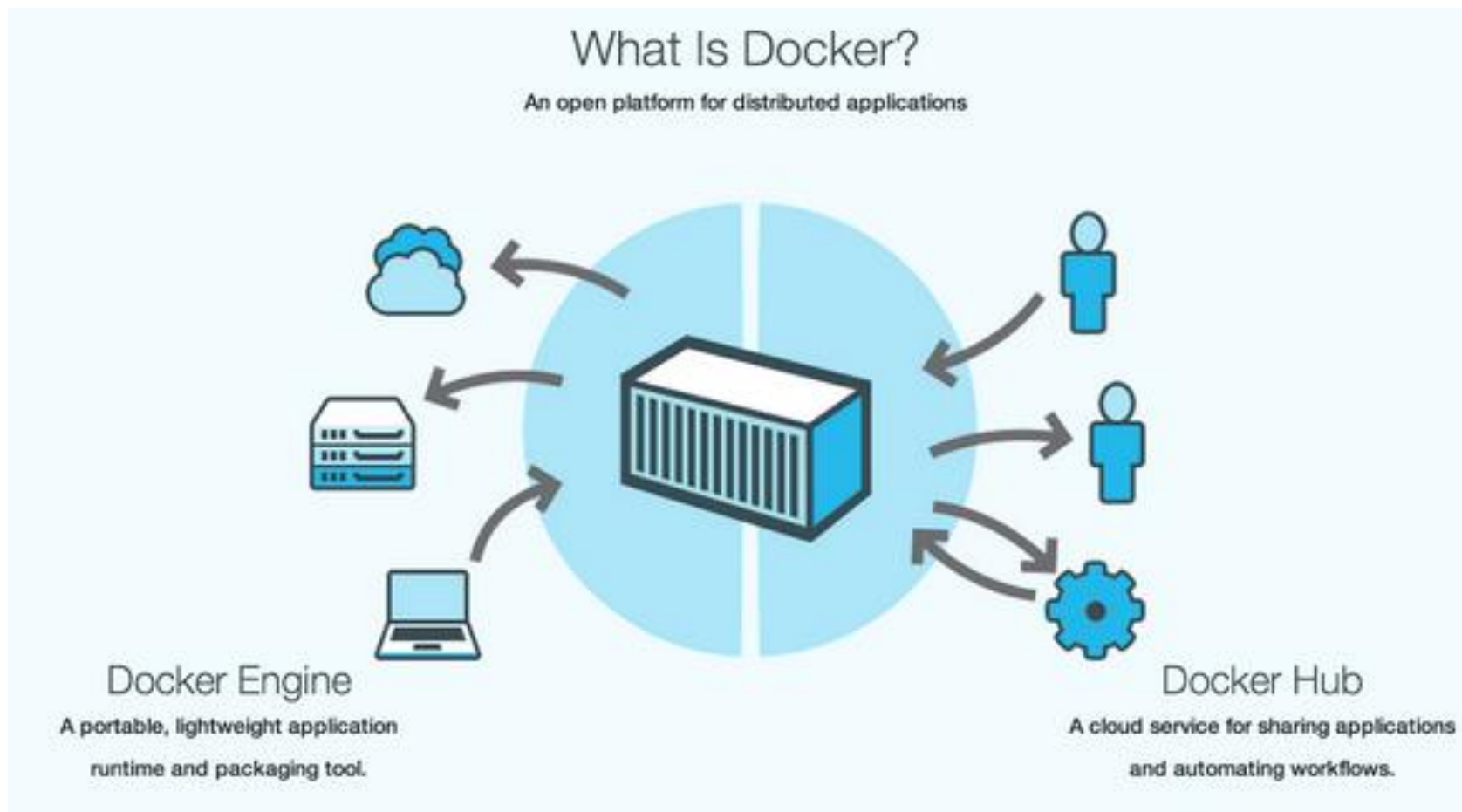


架构实现

- Docker
- Orchestration
- UI

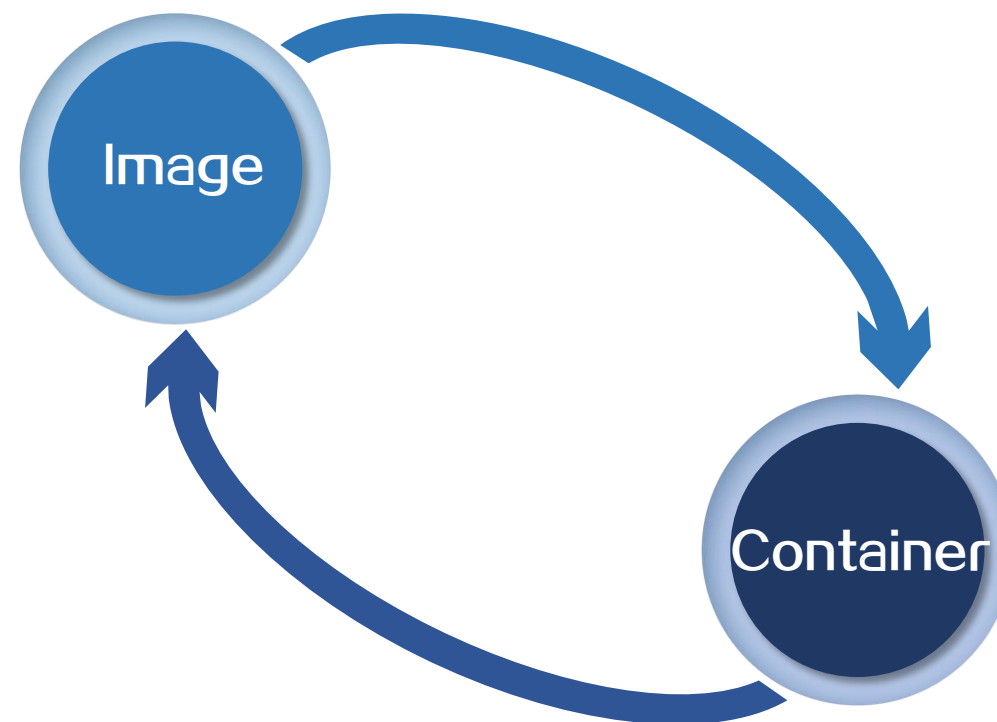
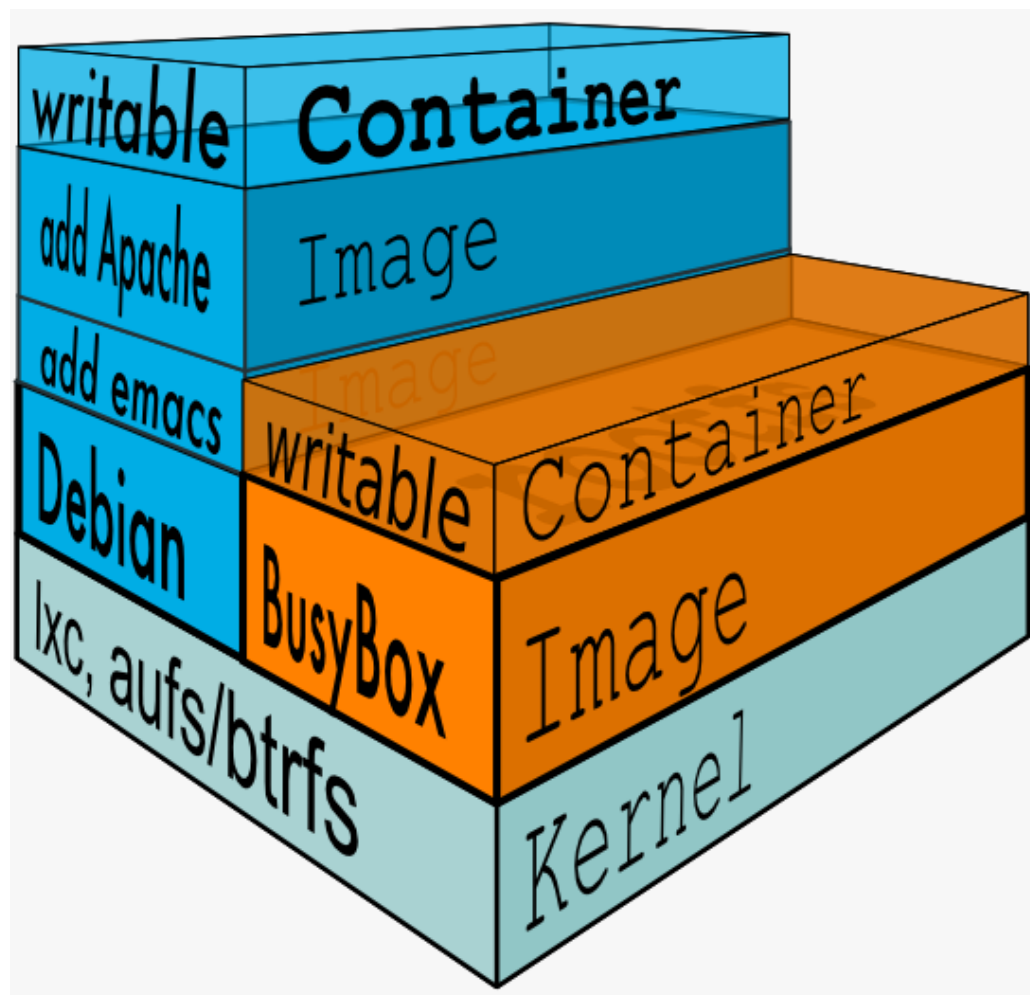


Docker简介-Docker是什么？

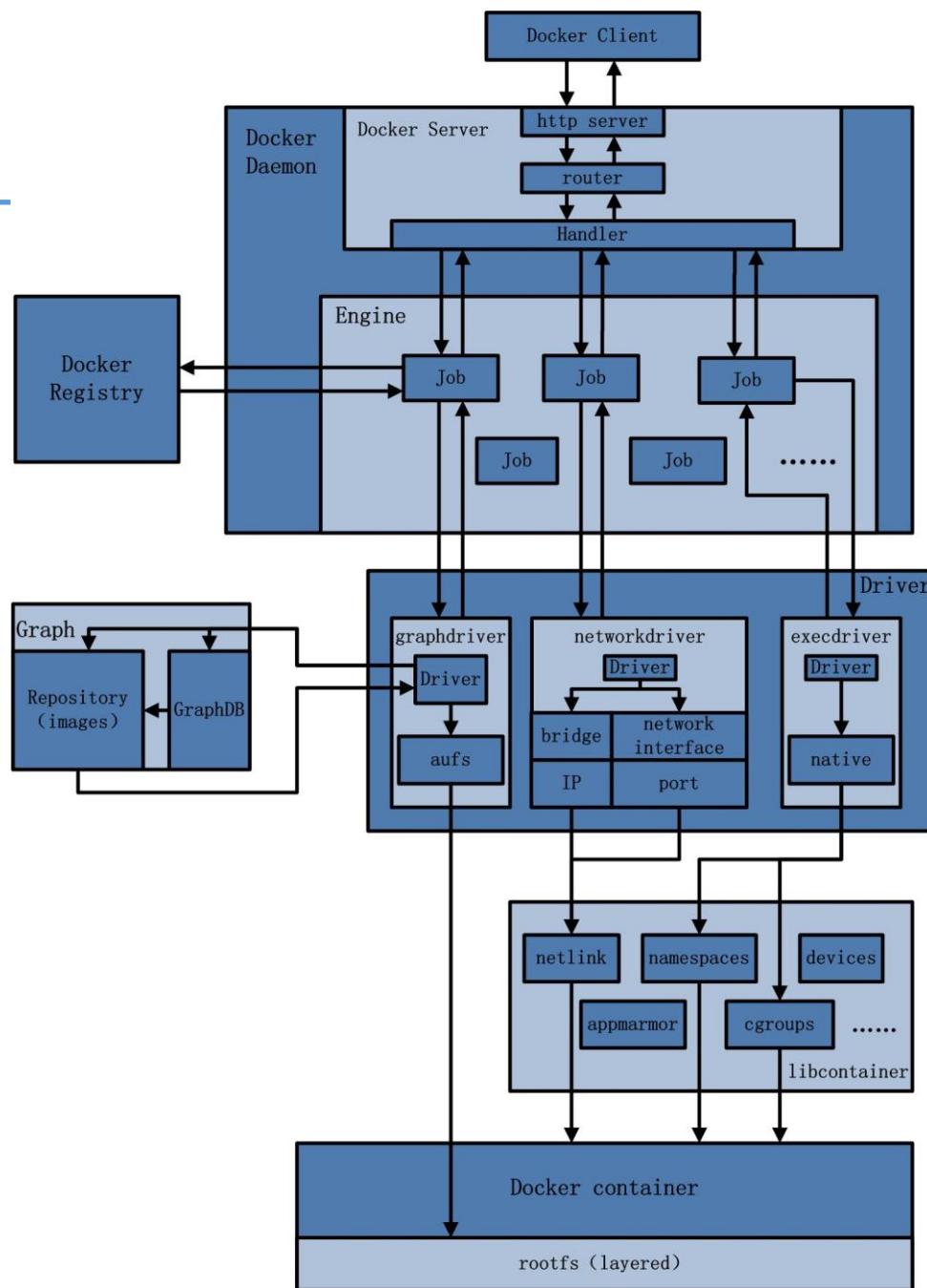




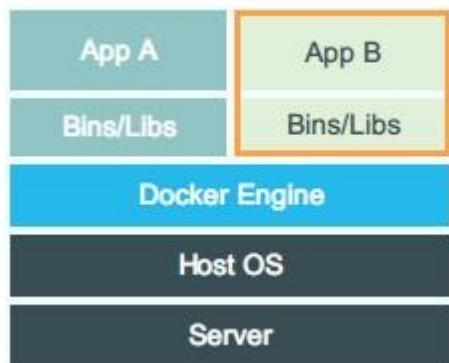
Docker简介-主要概念？



Docker简介-架构图



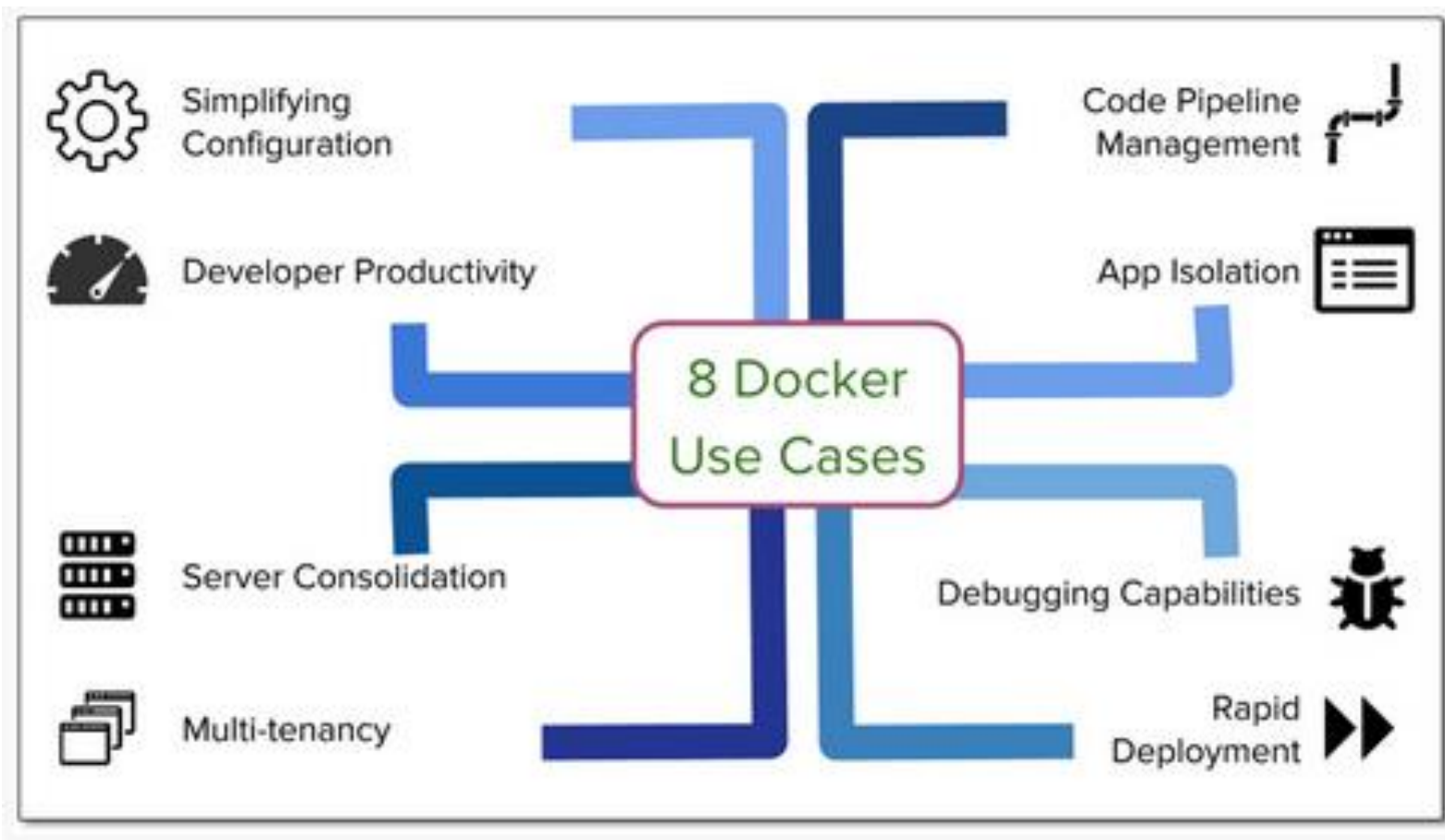
Docker vs VM



- ❑ VM的Hypervisor需要实现硬件的虚拟化，搭建自己的Guest操作系统，浪费了一定的资源。
- ❑ Docker基于操作系统之上通过docker engine实现轻量级容器，复用Host主机的操作系统。
- ❑ Docker engine实现docker容器的调度和隔离，降低了系统和资源的重量级，提高资源利用率，性能接近物理机。



Docker的应用场景



02

架构设计与实现

A

架构设计

- Security AppStore
- Client AppStore

B

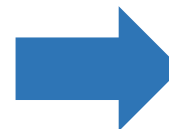
Docker技术简介

- 简介
- 应用场景

C

架构实现

- Docker
- Orchestration
- UI





架构实现

@docker

- Docker技术与AppStore映射关系。
- Docker Swam集群系统。
- Docker compose实现对容器的编排

@orchestration

- REST方式实现
- Java Quartz调度Job任务
- Job格式定义

@UI

- Python Django实现前端页面。
- 用户管理
- App信息保存等



架构实现@docker

@docker

- Docker技术与AppStore映射关系。
- Docker Swam集群系统。
- Docker compose实现对容器的编排



Docker仓库保存AppStore所需的App信息。



Docker Image对应AppStore中App , 可以多对一。



Docker Container对应运行的App实例。



Docker的pull操作 , 对应AppStore的App下载。

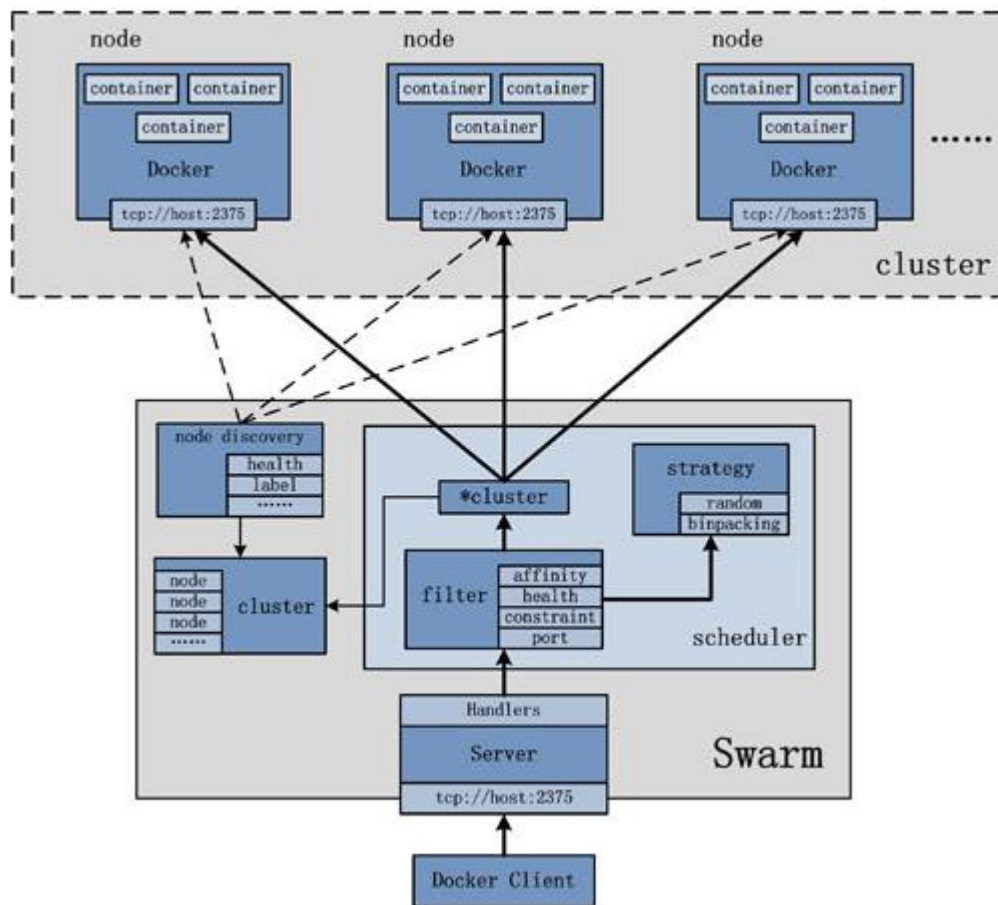


架构实现@docker



@docker

- Docker技术与AppStore映射关系。
- Docker Swarm集群系统。
- Docker compose实现对容器的编排

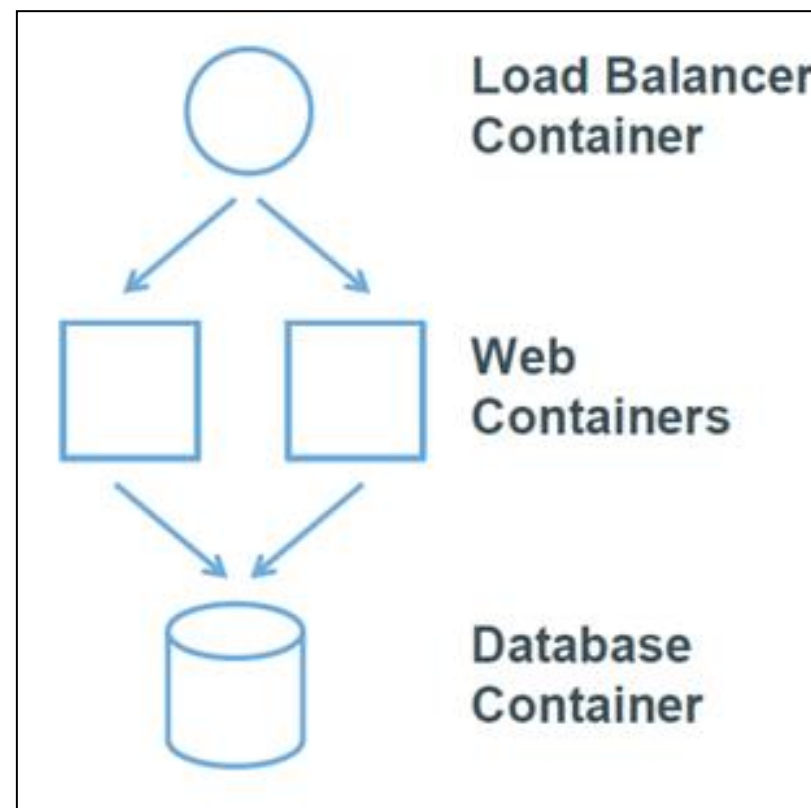




架构实现@docker

@docker

- Docker技术与AppStore映射关系。
- Docker Swam集群系统。
- Docker compose实现对容器的编排



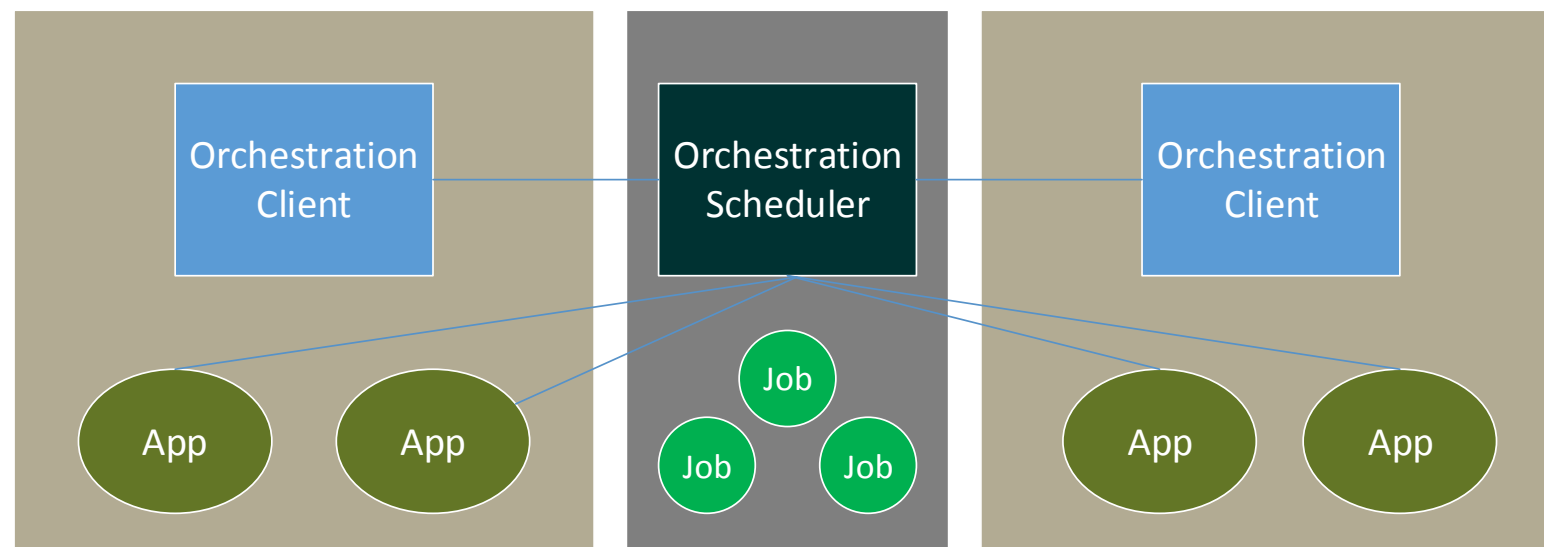


架构实现@orchestration



@orchestration

- REST方式实现
- Java Quartz调度Job任务
- Job格式定义





架构实现@orchestration



@orchestration

- REST方式实现
- Java Quartz调度Job任务
- Job格式定义

```
{
  "orchestration_name": "orchestration_name",
  "orchestration_nodes": [
    {
      "eventApp": {
        "app_event_url": "http://localhost/eventApp/event",
        "app_name": "event_app_name"
      },
      "triggerApp": {
        "app_name": "trigger_app_name",
        "app_trigger": "NOT_SAFE",
        "data_type": "STRING", //STRING, INT
        "compare_type": "EQ", //EQ, NEQ, GT, LT
        "app_trigger_url": "http://localhost/triggerApp/getTrigger"
      }
    }
  ]
}
```

架构实现@UI

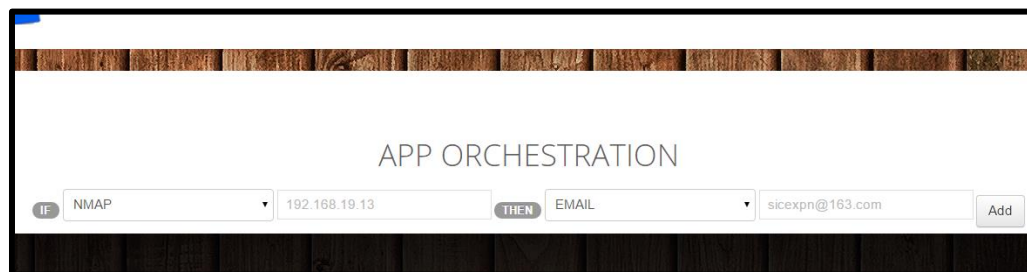
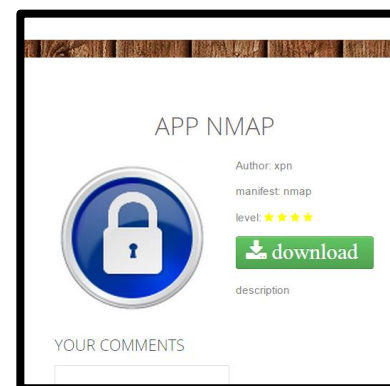
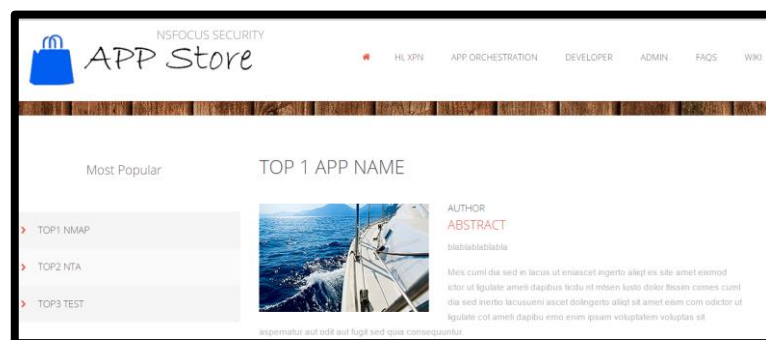


NSFOCUS SECURITY

APP Store

@UI

- Python Django实现前端页面。
- 用户管理
- App信息保存等





CONTENTS

目录

背景及意义

架构设计与实现

系统流程与展示

未来规划

03

系统流程与展示

A

AppStore

- App一键下载、部署
- App操作

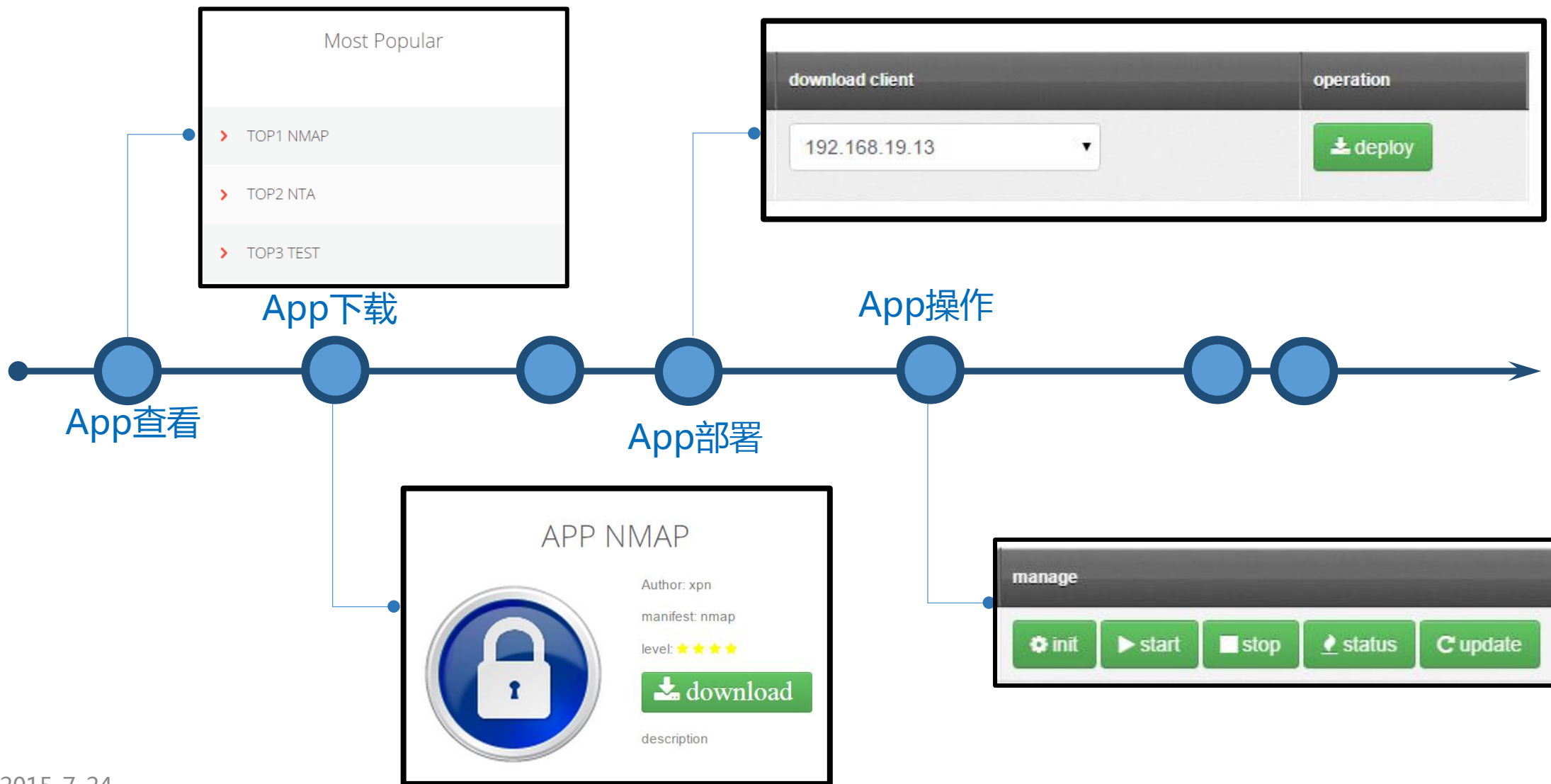
B

App Orchestration

- IFTTT

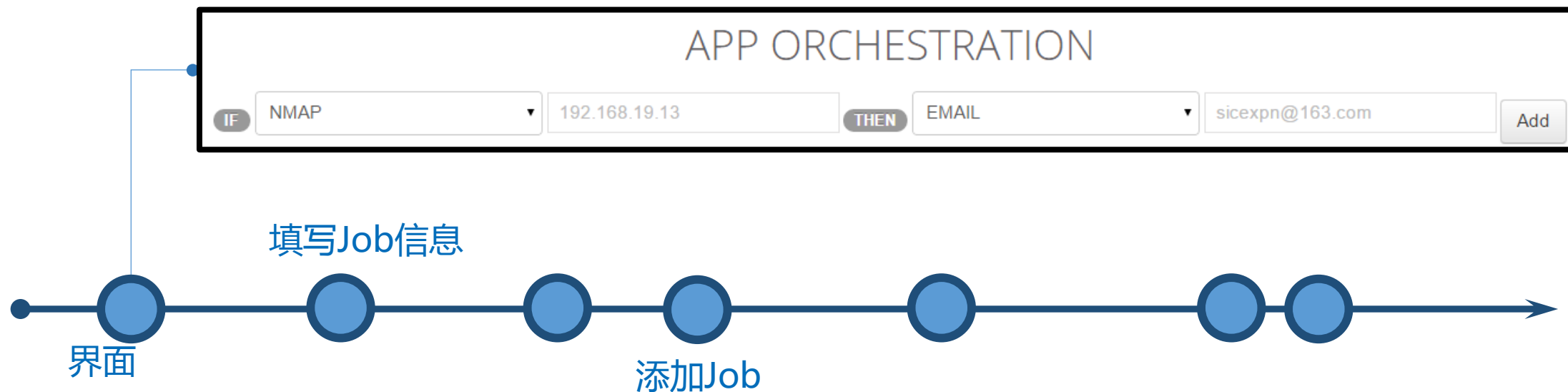


系统流程@AppStore





系统流程@Orchestration




系统展示



NSFOCUS SECURITY


APP Store

<http://appstore.research.intra.nsfocus.com:8000/>



NSFOCUS SECURITY

APP Store

 [HI, XPN](#) [APP ORCHESTRATION](#) [DEVELOPER](#) [ADMIN](#) [FAQS](#) [WIKI](#)


Most Popular

> TOP1 NMAP

> TOP2 NTA

> TOP3 TEST

TOP 1 APP NAME



AUTHOR

ABSTRACT

blablablabla

Mes cuml dia sed in lacus ut eniascet ingerto aliaqt es site amet eismod ictor ut ligulate ameti dapibus ticdu nt mtsen lusto dolor ltissim comes cuml dia sed inertio lacusueni ascet dolingerto aliaqt sit amet eism com odictor ut ligulate cot ameti dapibu emo enim ipsam voluptatem voluptas sit





CONTENTS

目录

背景及意义

架构设计与实现

系统流程与展示

未来规划

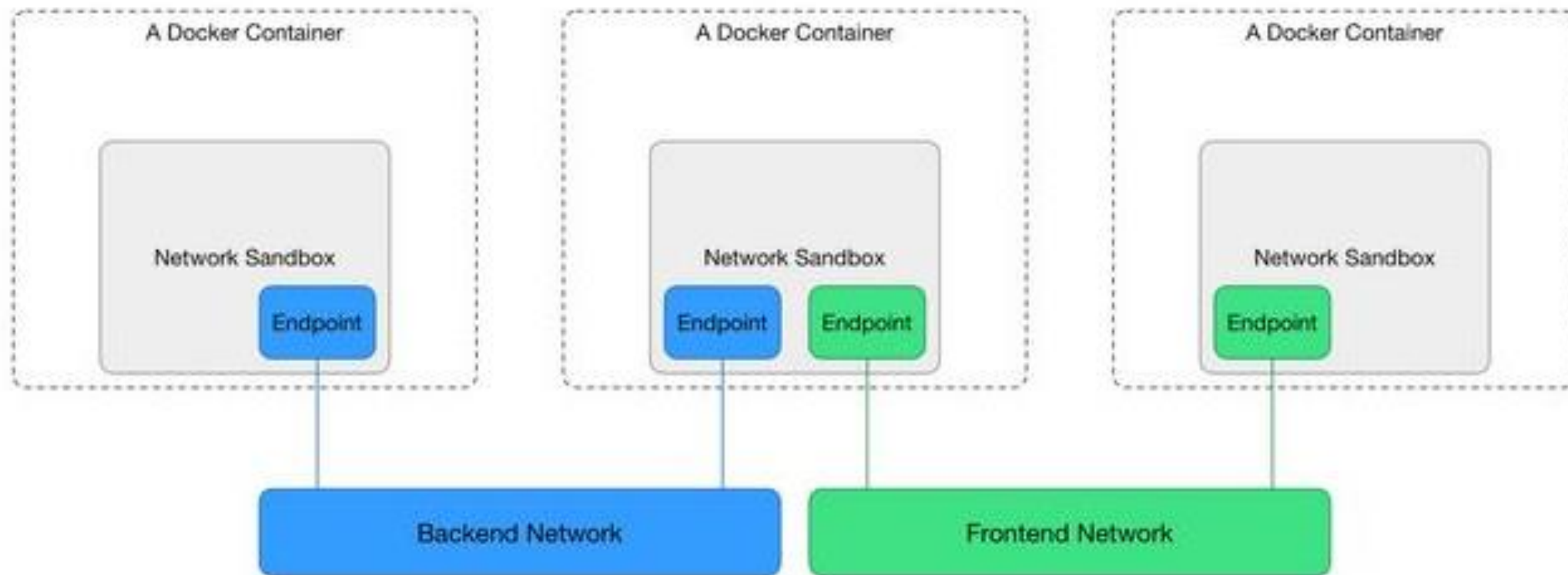
04

未来规划



- Docker代替现有的VM，实现安全设备的docker化。
- Docker 容器接入SDN网络，libnetwork

Libnetwork-容器网络模型



A hand is holding a white tablet. The left side of the image is covered by a semi-transparent blue overlay. The word 'THANKS' is written in large, bold, sans-serif capital letters across the center. The first part of the word, 'THAN', is white and positioned over the blue overlay, while the second part, 'KS', is blue and positioned over the white tablet.

THANKS

谢谢观赏