



安天网络安全有效价值研讨会

有效防护 价值输出

CipherGateway 产品安全实践与塔防模式探索

白小勇 北京炼石网络技术有限公司 CEO



炼石网络

冰峰峻立

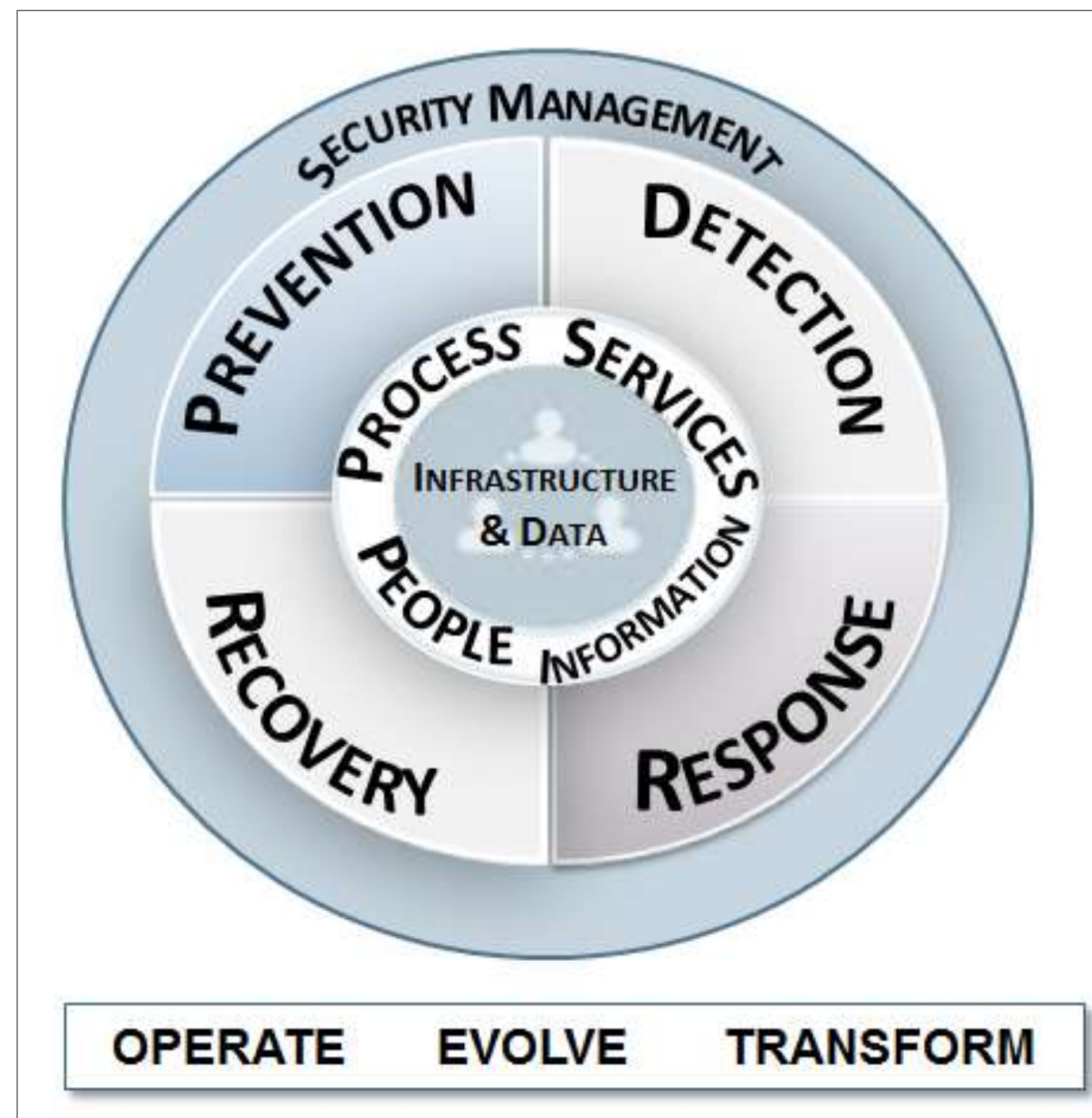
安天网络安全冬训营第四期

提纲

- 安全产品与产品安全
- 威胁场景与纵深应对措施
- 可复用模式输出

PART
01

安全产品与产品安全



安全产品与产品安全

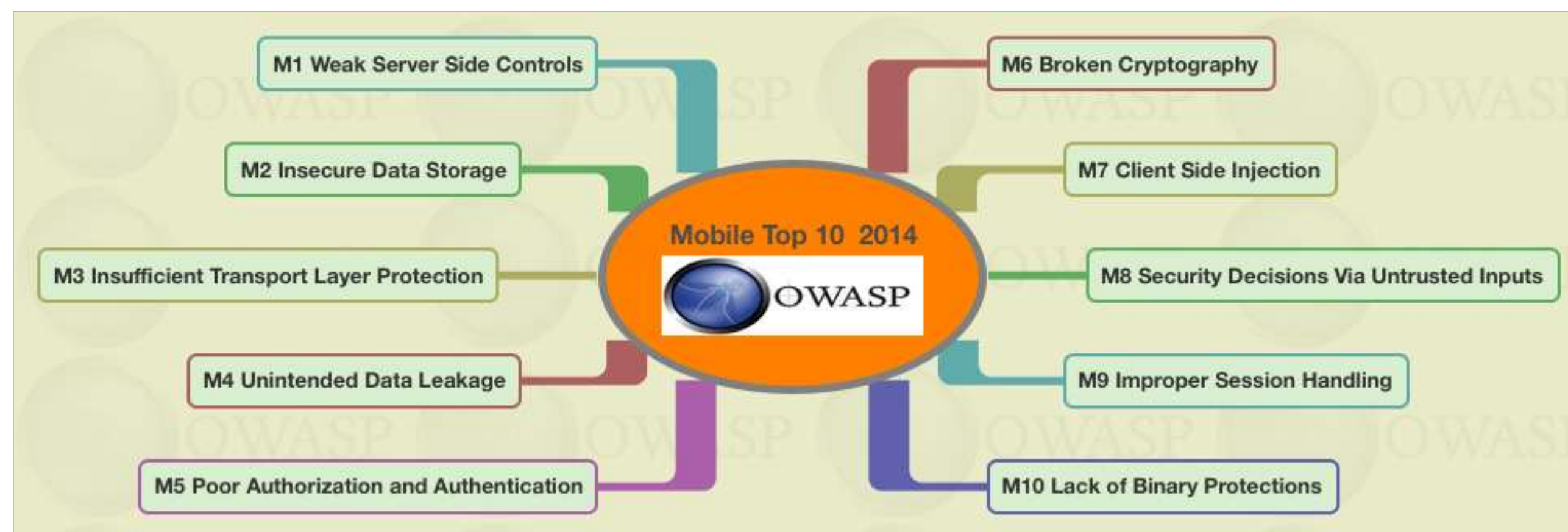
- **安全产品**

- 保卫网络、系统、数据等基础设施、资产和内容安全的硬件/软件产品

- **产品安全**

- 产品自身的安全

皮之不存毛将焉附？



2017年1月2日09:55

总结2016年十大安全产品漏洞，你们真的是搞安全的么？



信息安全软硬件产品用于保障信息系统的安全，但不要忘了安全产品本身也一样是代码写的，一样有漏洞，乌云漏洞库中可以搜索到众多安全产品的漏洞，甚至有不良厂家预留的后门，年关将至，各位好好检查一下你们购买的安全产品有没有问题吧！嘶吼一下这安全行业



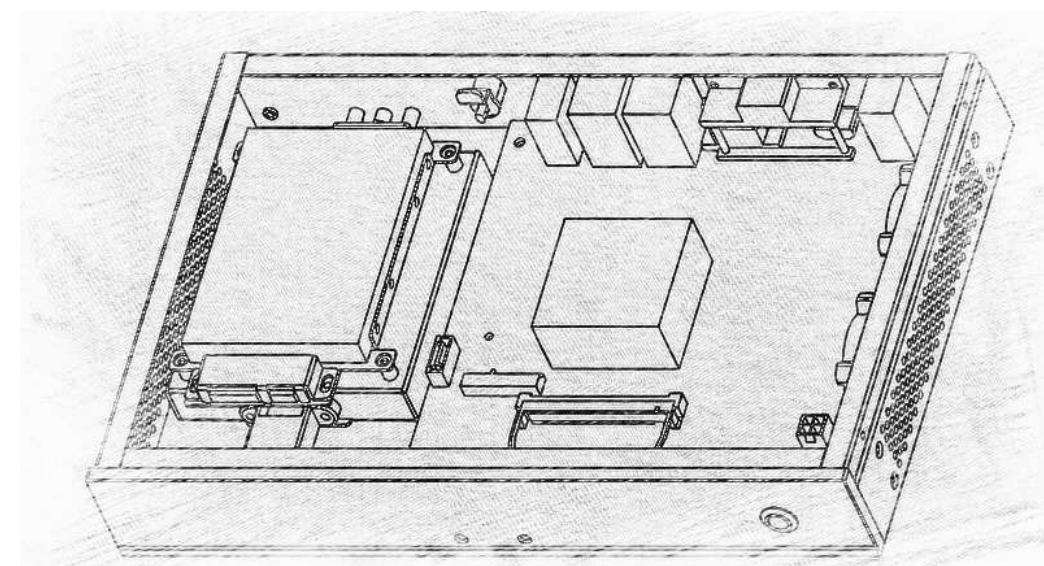
安全产品自身频出安全问题

- **趋势科技杀毒软件命令执行漏洞**
 - 允许黑客远程执行任意命令，并可以窃取用户使用其杀毒软件中内置的密码管理器所保存的密码。
- **思科防火墙设备漏洞**
 - 可以远程利用该漏洞来允许恶意软件绕过Firepower的检测机制；
 - 攻击者可以向系统发出伪造的SNMP包，执行任意代码，并且获得系统的完整控制权。
- **网神多款设备高危漏洞**
 - 多款网神设备皆存在高危漏洞，可以通过任意文件上传、SQL注入，甚至管理设备的用户名密码直接泄露漏洞，可以直接控制设备权限，可以在设备中留下各种后门。
- **方程式组织利用的多个漏洞**
 - 天融信、飞塔、Juniper

有效防护？

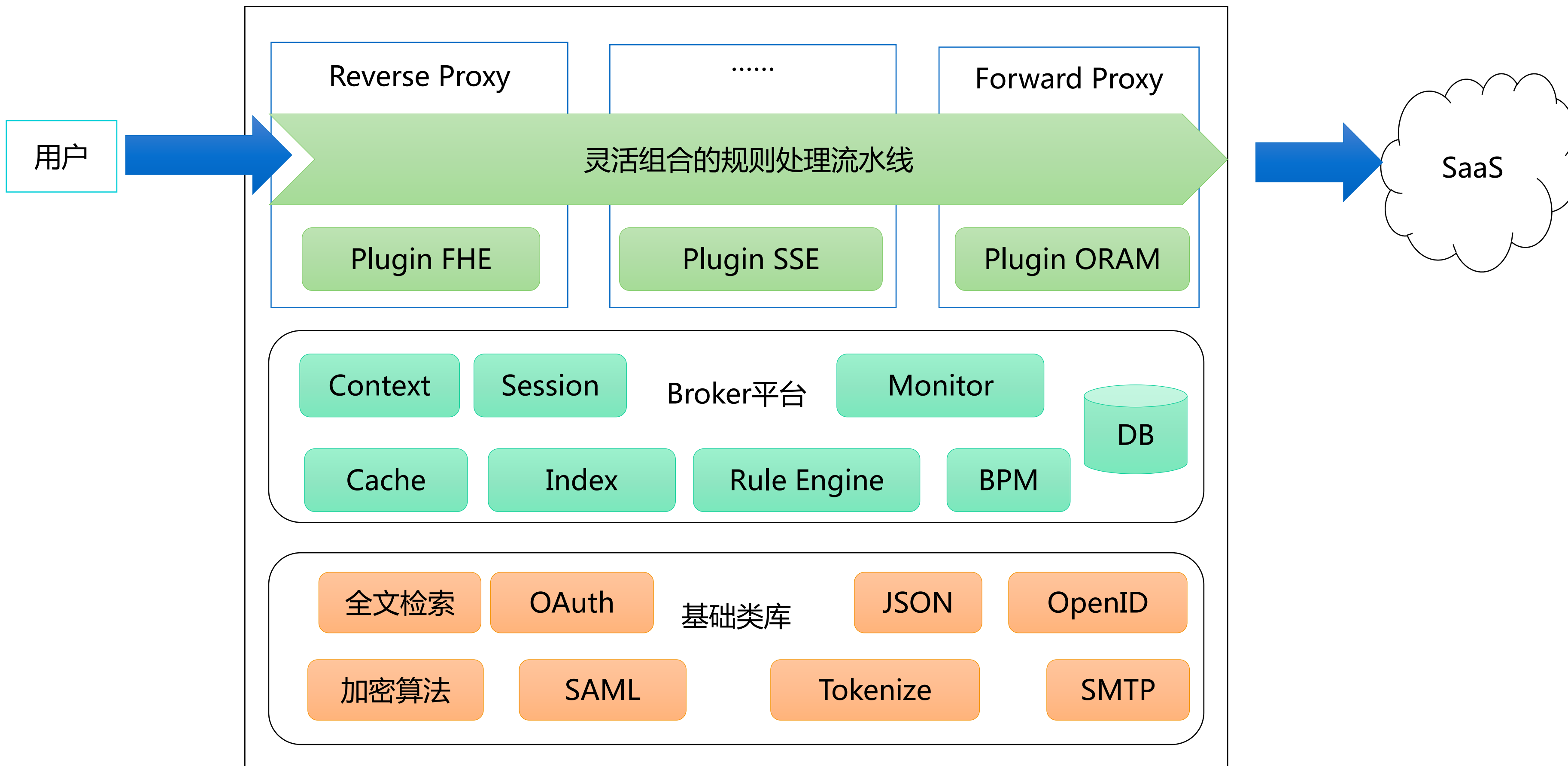
问题原因分析

- 导致安全产品自身出现问题的两大类原因：
 - 自身问题导致
 - 业务逻辑，堆栈、内存等资源控制，权限控制
 - 管理程序权限隔离不合理
 - 后端接受数据时校验不彻底
 - 测试场景覆盖不全导致在部分高压场景下内存溢出
 - 第三方问题导致
 - 使用了第三方开源组件，组件出现问题，导致产品被出问题，如OpenSSL、Docker
 - Nginx，权限提升漏洞CVE-2016-1247，空指针间接引用漏洞CVE-2016-4450
 - Redis，远程代码执行漏洞CVE-2016-8339，Lua字节码执行漏洞CVE-2015-4335
 - Pgsql，栈缓冲区溢出漏洞CVE-2015-5289，远程拒绝服务漏洞(CVE-2015-3165)
 - Java SE，远程安全漏洞CVE-2016-0687，远程拒绝服务漏洞CVE-2015-3165



CG更要注重产品安全

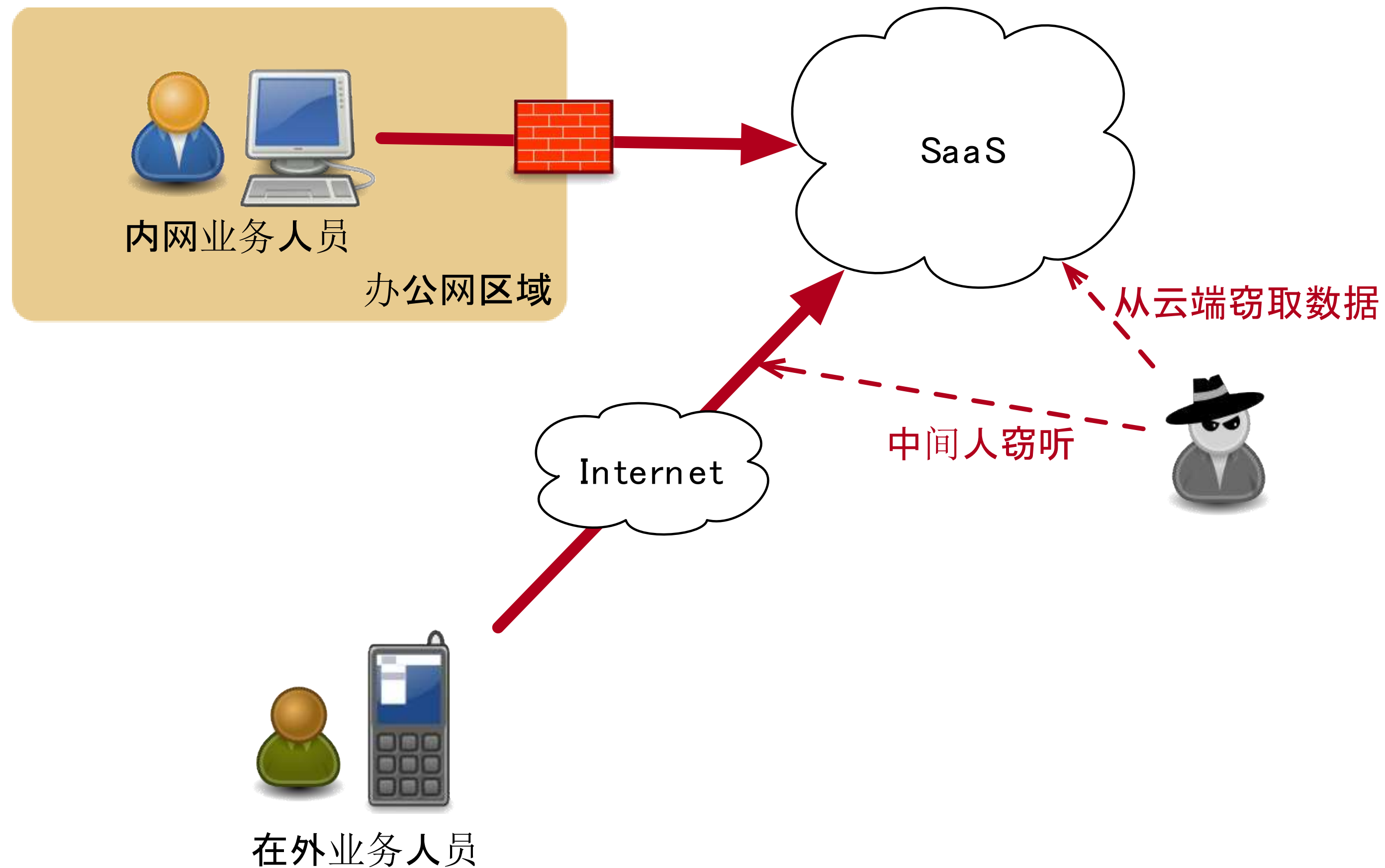
- 安全产品需要做好自身的安全防护，才能保障安全产品的有效性
- CG是一款更复杂的安全网关：
 - 懂业务应用上下文
 - 表面看个简单Box，实际是个完整的复杂应用系统
- 目标是成为安全的安全网关：
 - 产品架构设计遵循Design for Failure，贯彻纵深防御思想
 - 有效防护CG自身以保障CG有效性，进而对CG的防护对象提供有效防护



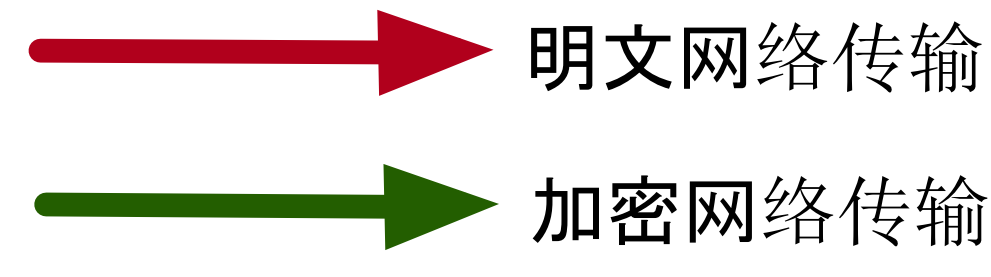
PART
02

威胁场景与纵深应对措施

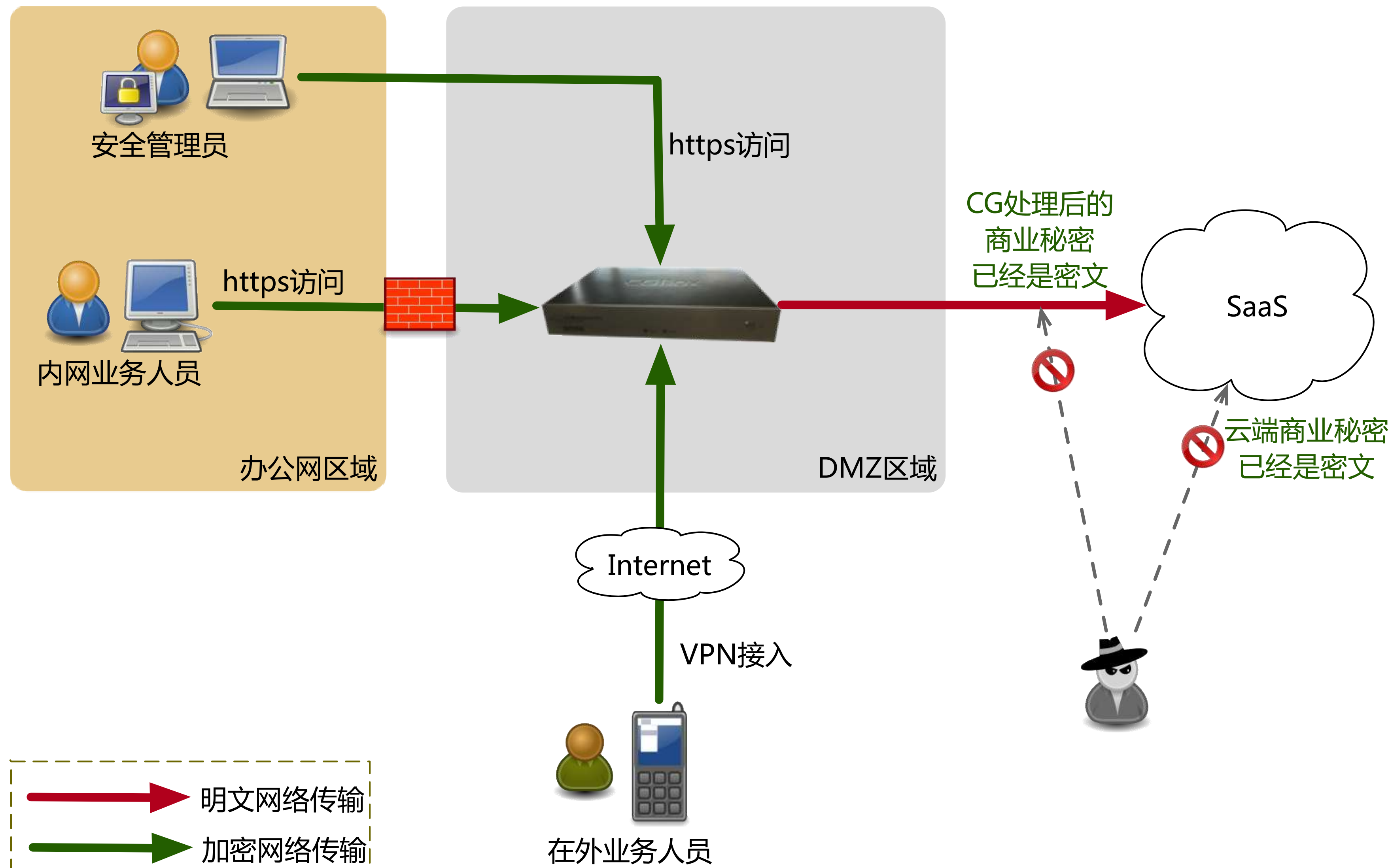
CG的使用场景（部署前）



- 企业级用户在使用各类云服务过程中，大量涉及到商业秘密的数据不得不离开企业，带来的数据安全风险直接关乎企业的商业利益。

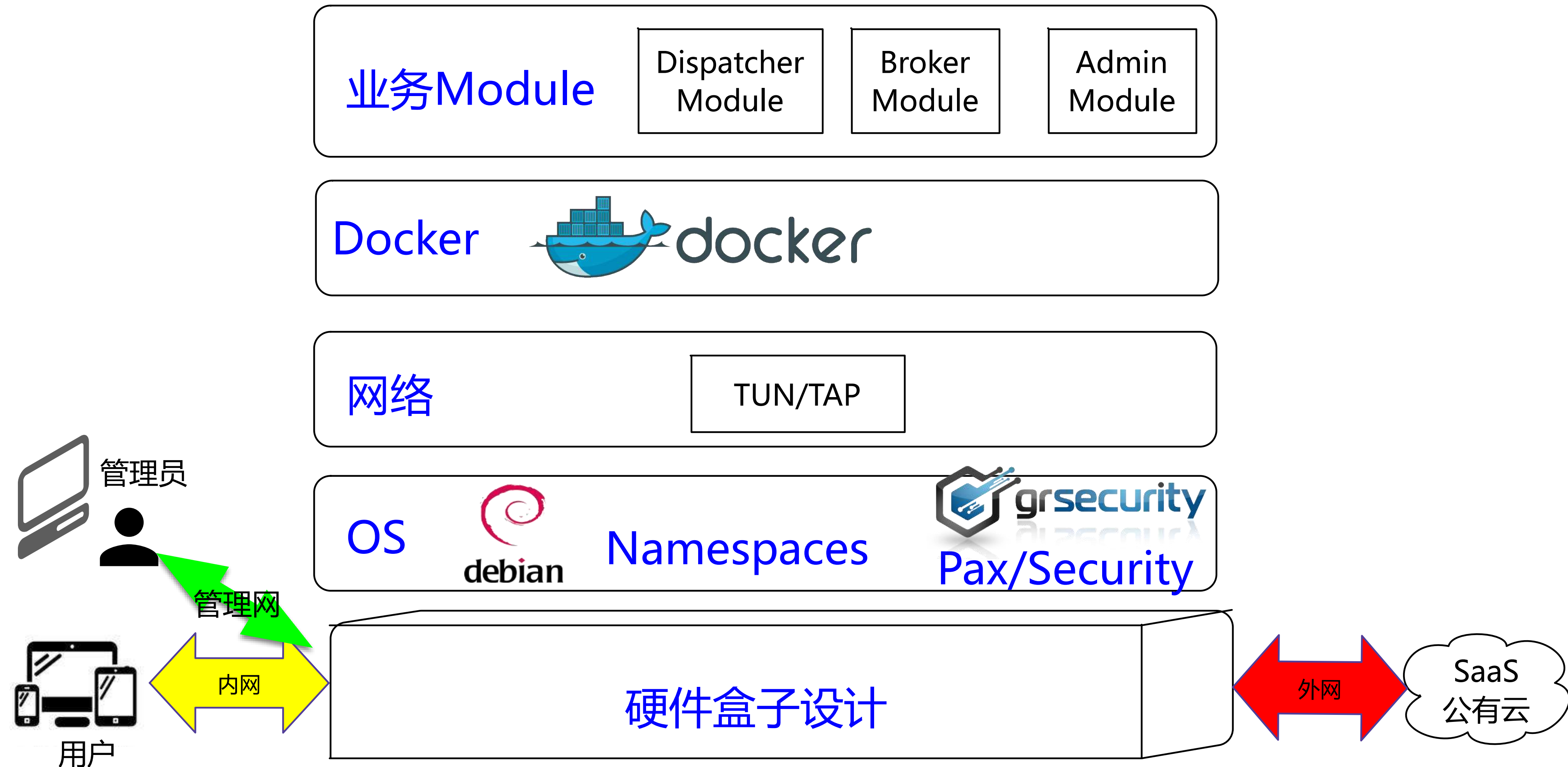


CG的使用场景（部署后）

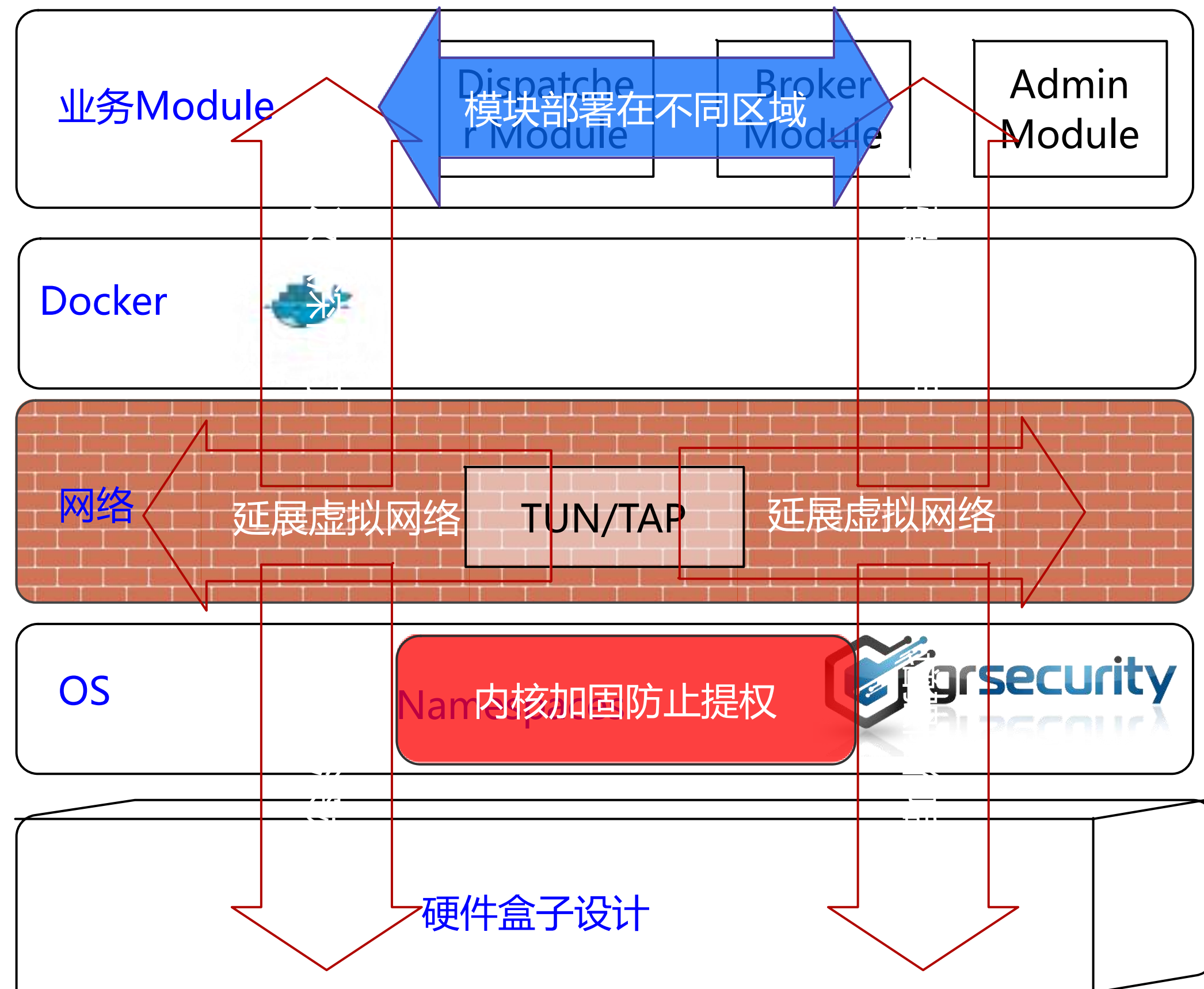


- 企业部署使用CG业务应用安全网关后，其商业秘密数据只要离开企业就会被加密处理，数据以密文形式存储在云端服务器。
- 密钥始终在企业侧、并由企业自己掌控，确保了企业的数据掌控权。
- 能够保证用户照常使用云应用的各项原生功能，比如搜索、统计等，在保障用户原有系统使用方式不改变的同时，也不需要云服务商改造系统。

【回顾去年冬训营】安全加固概要



重构设计要点



威胁场景列举：

- Webshell攻击，导致数据泄露
- 获取存储在密码卡中的密钥，致使加密毫无价值
- 将明文数据转发，使攻击者可以远程获取加密的信息
- 关掉敏感数据加密功能使其成为透明代理

安全防护手段列举：

- 每个加固点，不够有效，也难以输出价值
- 使用Pax防止内核提权，防护缓冲区溢出漏洞。
- 使用RBAC策略限制应用行为，避免应用出现漏洞后操作其他应用。
- 使用WAF防止web服务被注入webshell。
- 使用虚拟网络纵深来防护单个服务被入侵导致全面失控的问题。

利用先发优势，构建有利纵深

冰峰峻立

安天网络安全冬训营第四期



需要安全接入的用户



安全管理员



业务人员

设计概览



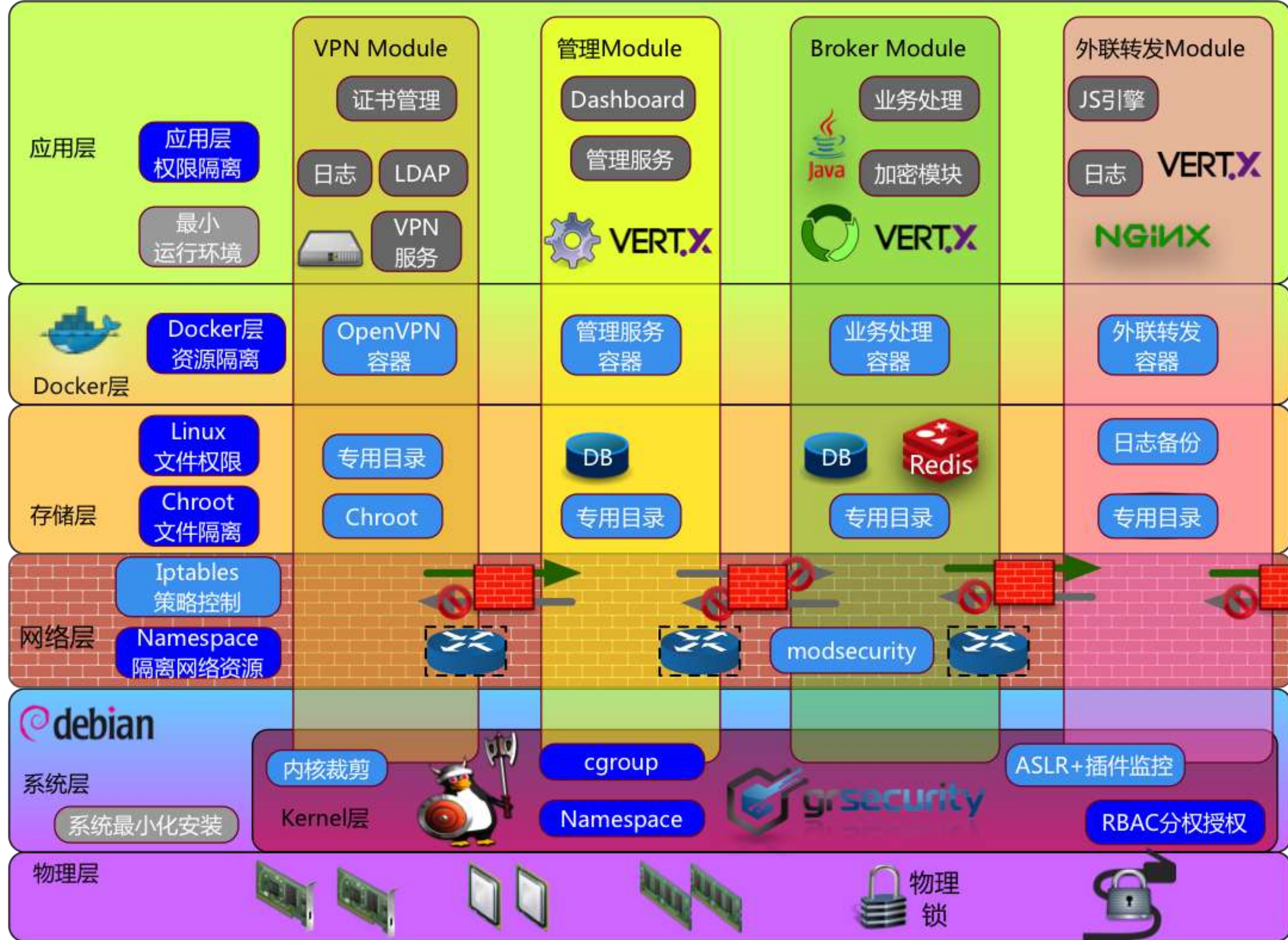
镜像管理



实例创建

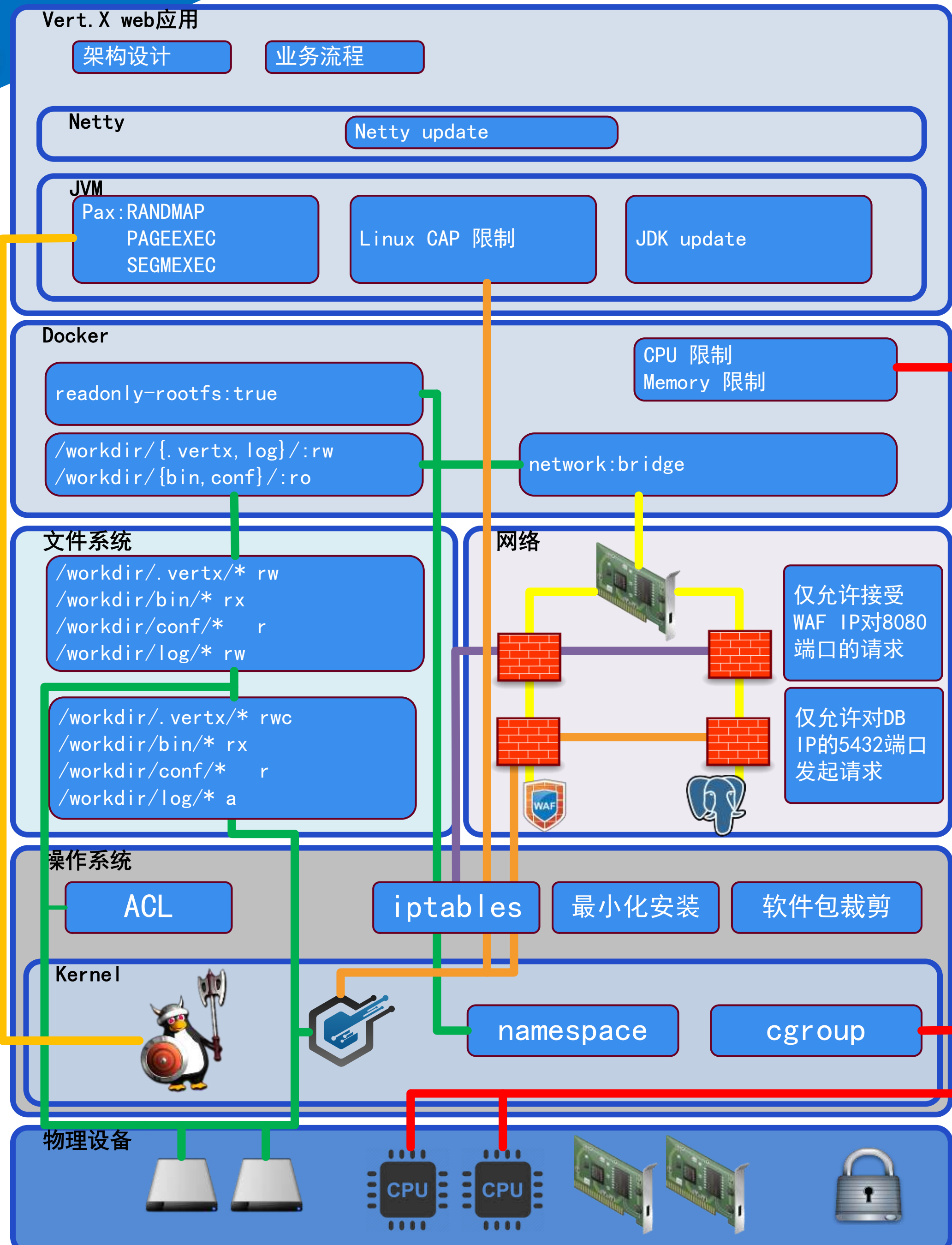


启停服务



- 基于面向失效的原则
- 构建出有效的纵深
- 纵深协同，而非纵深堆砌
- 考虑的方面
 - 从技术栈层次及内核
 - 从网络角度
 - 从多模块纵深角度
 - 从运维流程角度

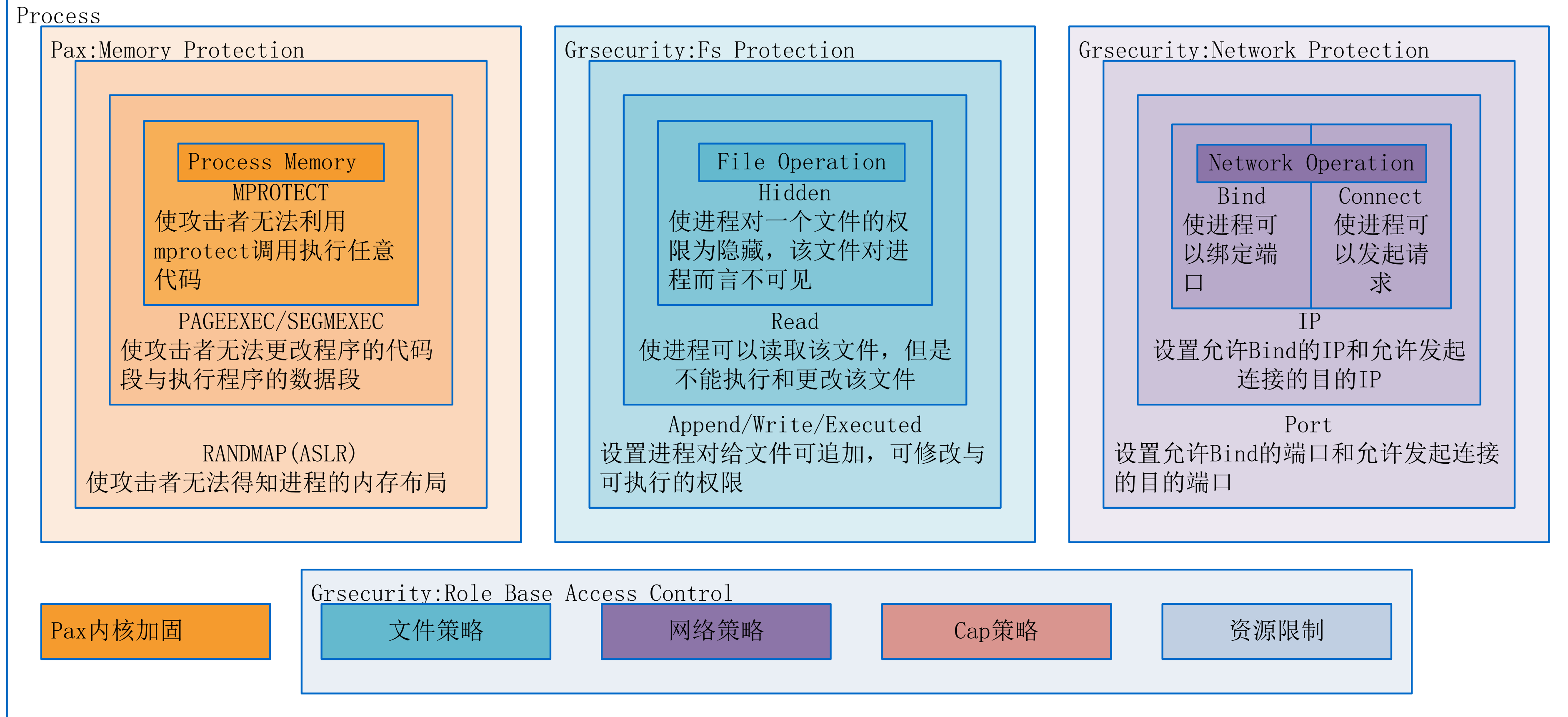
SaaS



技术栈层次的防护

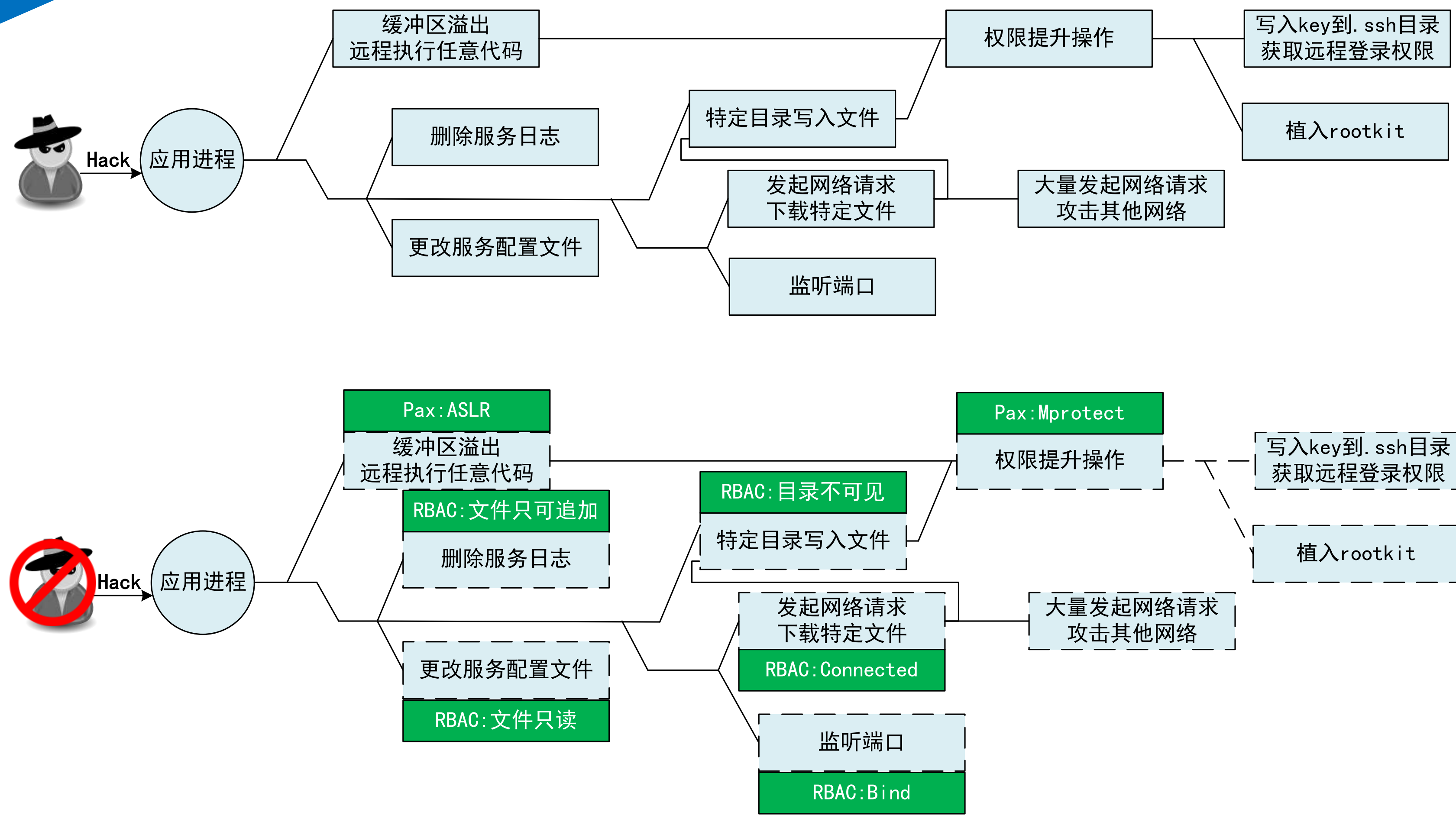
- 从业务设计上，以安全优先的原则合理划分成不同模块，并结合Linux系统的安全机制为每个模块规划了最小权限的资源分配
- 针对不同模块的需求，在技术栈的各层包含硬件层、系统层、网络层、存储层、Docker容器层、应用层，增强了多点纵深协同、面对失效有防护后手的组合安全策略
- 文件系统:
 - 体现在Docker的mount namespace能力
 - OS提供的ACL能力
 - grsecurity提供的RBAC能力中的文件部分
- 网络:
 - 体现在Docker的网络 namespace能力，操作系统提供的虚拟网络设备能力
 - Kernel提供的ipfilter能力
 - grsecurity提供的RBAC能力中的网络部分

PaX/grsecurity加固



- 为了应对系统内核Root提权导致的防护全盘失效，使用PaX/grsecurity结合各模块实际运行的不同需求从内核进行基于RBAC加固；
- PaX的能力：
 - 内存防护，能够抵御缓冲区溢出攻击，远程代码执行以及权限提升
- grsecurity的能力：
 - 强访问控制能力
 - RBAC能力，控制一个文件的读，写，执行，追加等权限。

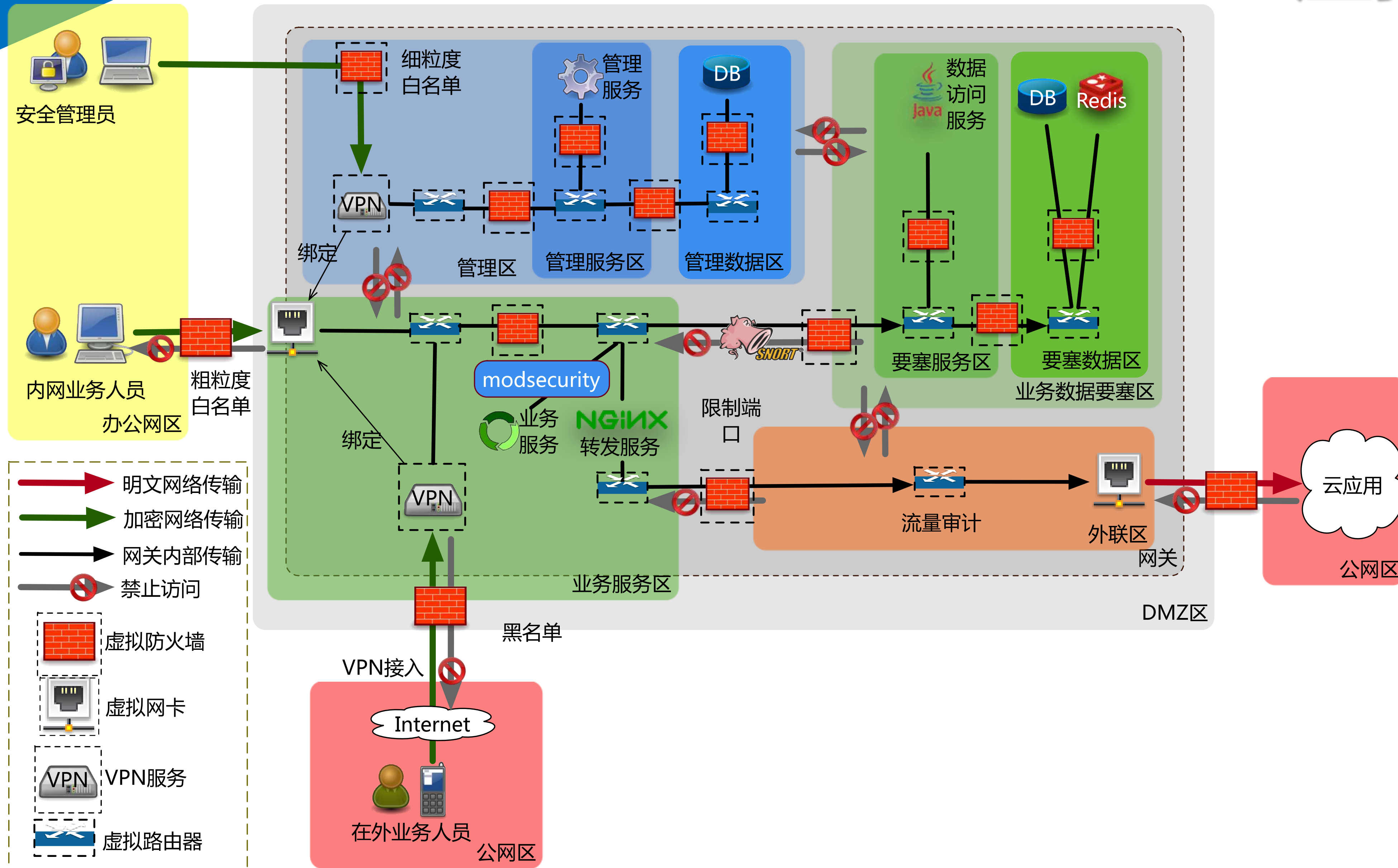
PaX/grsecurity的RBAC加固



```

role XXX u
subject /usr/bin/XXX
# 设置PAX的权限
+PAX_RANDMMAP
+PAX_MPROTECT
# 设置目录的权限
/XXX/log/* ca # 允许创建和追加
/XXX/conf/* r # 只允许读取
/lib/system/system/* h # 隐藏特定目录
/etc/rc.local h # 隐藏特定文件
#设置socket策略
# 只允许监听TCP 443端口
bind 0.0.0.0:443 stream tcp
# 只允许对数据库服务发起请求
Connect 172.17.0.1:5432 stream tcp
  
```

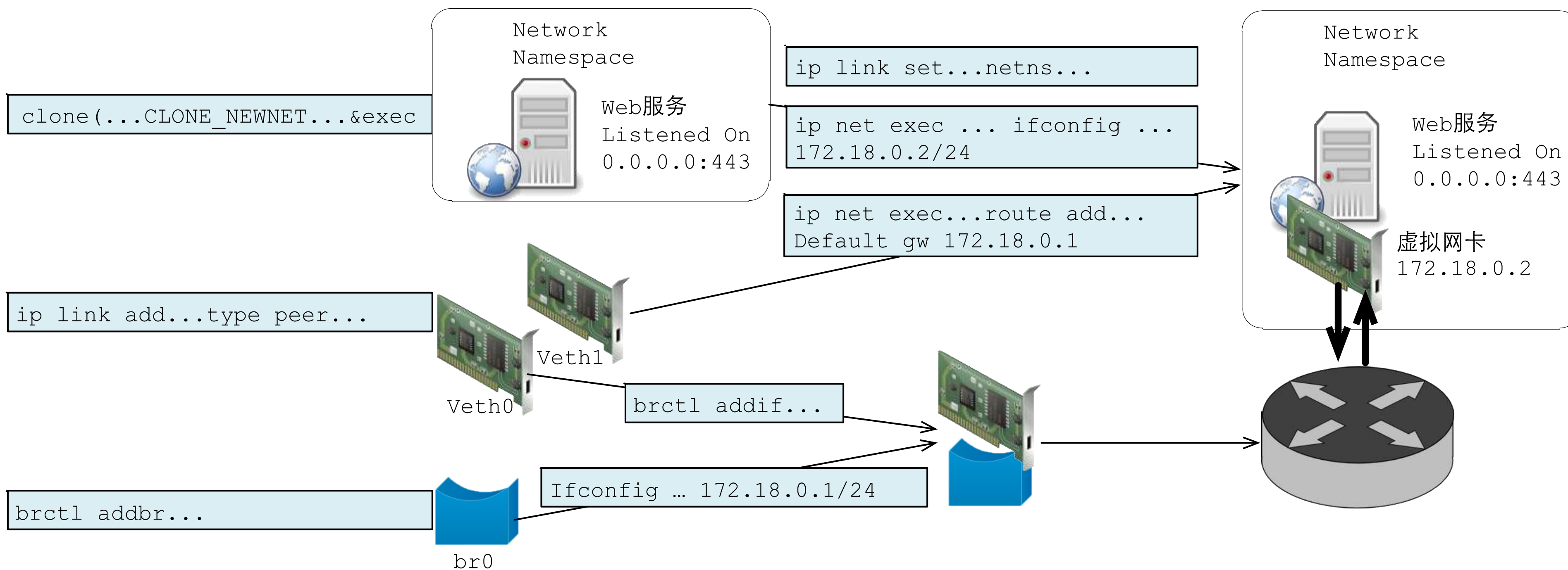
虚拟网络纵深



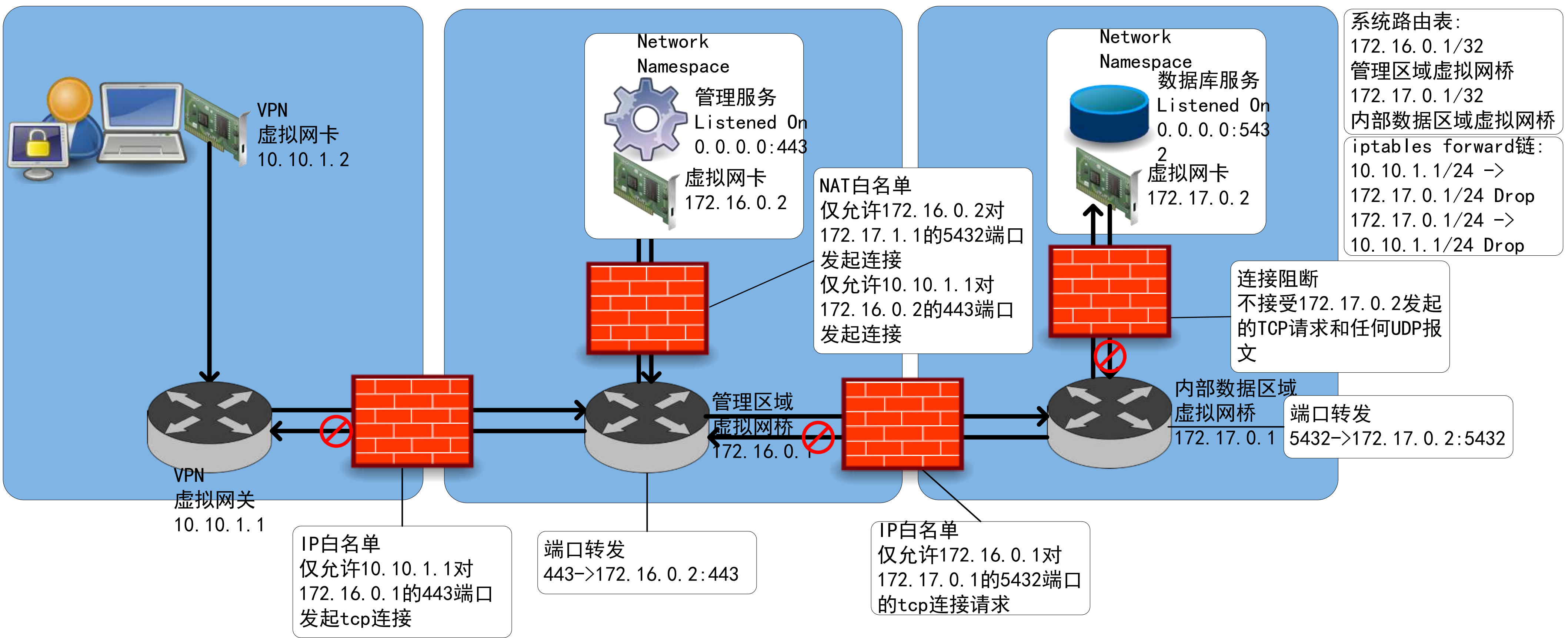
- 从网络层，根据对外服务的不同需求分隔出多个虚拟网络区域
- 通过服务发布和网络地址绑定，使网络连通
- 不同域之间加上访问控制安全策略，比如黑白名单访问
- 能够复用已有的iptables, snort, modsecurity等技术
- 每个区域内通过Namespace等技术实现资源隔离

构建流程:

- `clone&exec` 创建新的network namespace同时启动应用服务
- `ip link add` 创建虚拟网卡对
- `brctl addbr` 创建虚拟网桥
- `ip link set` 将虚拟网卡移动至服务所处的namespace中
- `brctl addif` 将另一张虚拟网卡添加到网桥上
- `ifconfig route`配置IP及路由

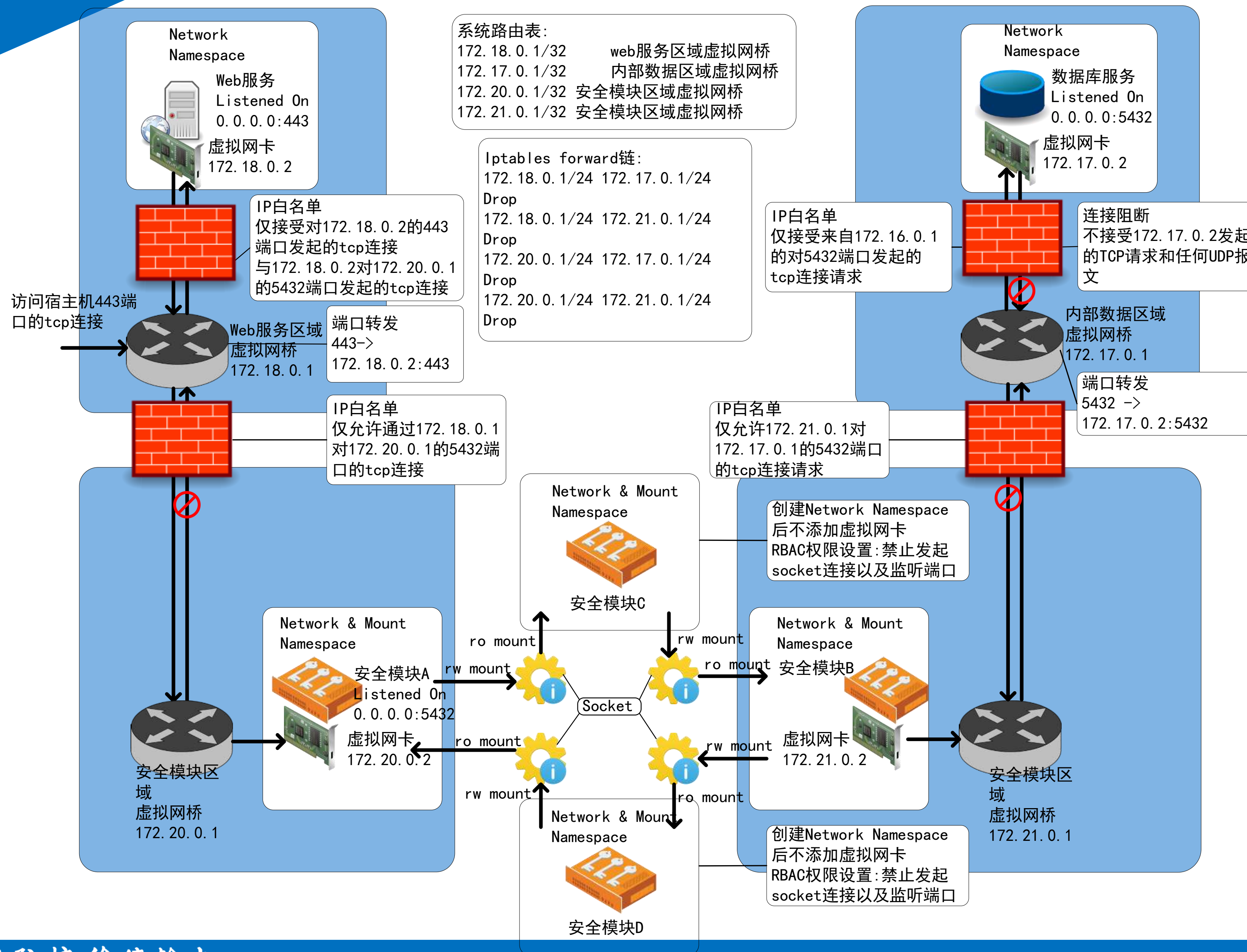


虚拟网络局部细节



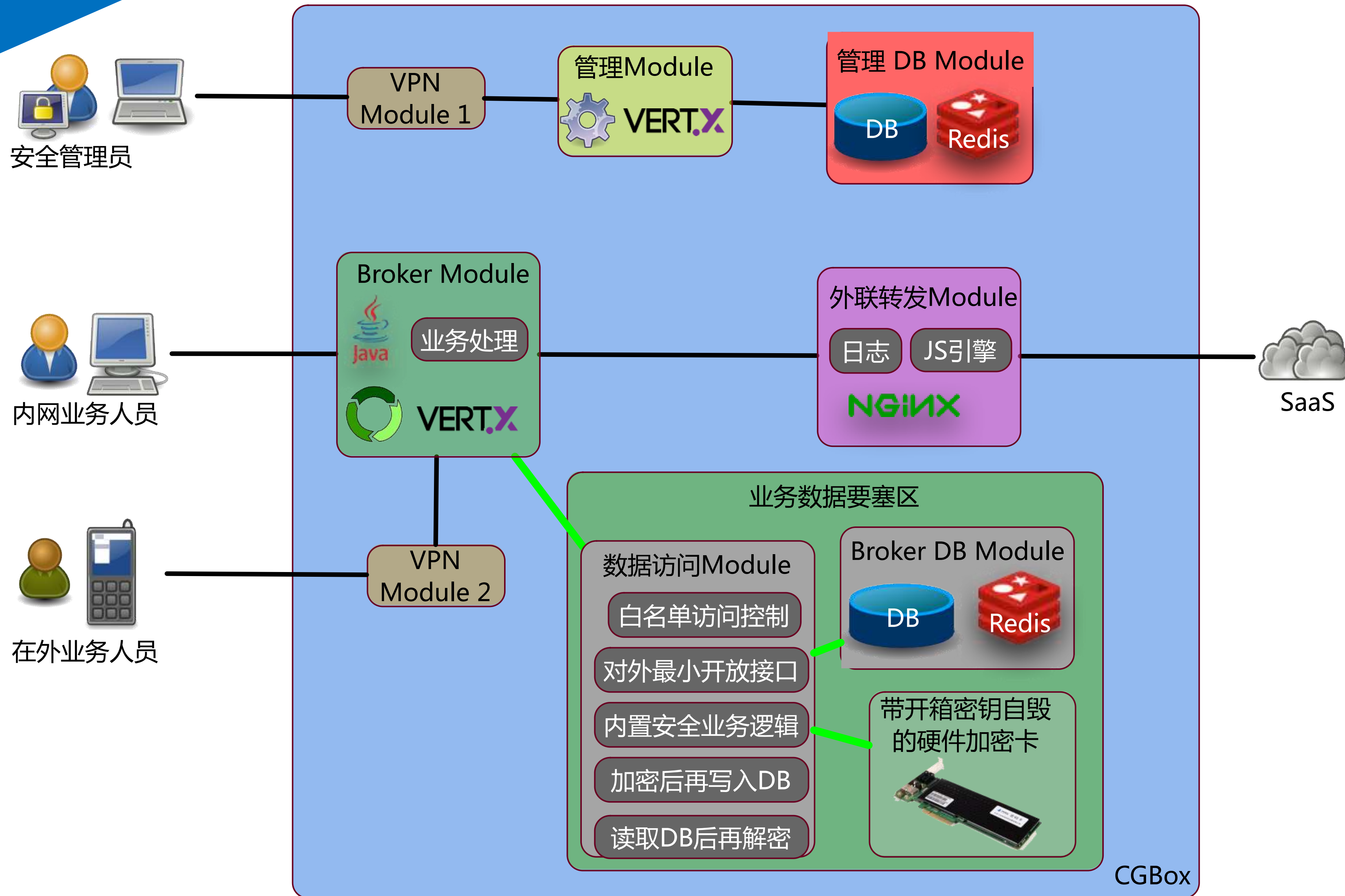
- 这个局部细节是安全管理员访问管理服务
- 管理员通过vpn接入管理区域，经过防火墙访问管理服务
- 管理服务模块和管理数据模块之间隔离
- 通过iptables实现白名单访问控制
- 通过NAT保护内部服务
- network namespace 网络隔离

模块间的虚拟网络单向通信



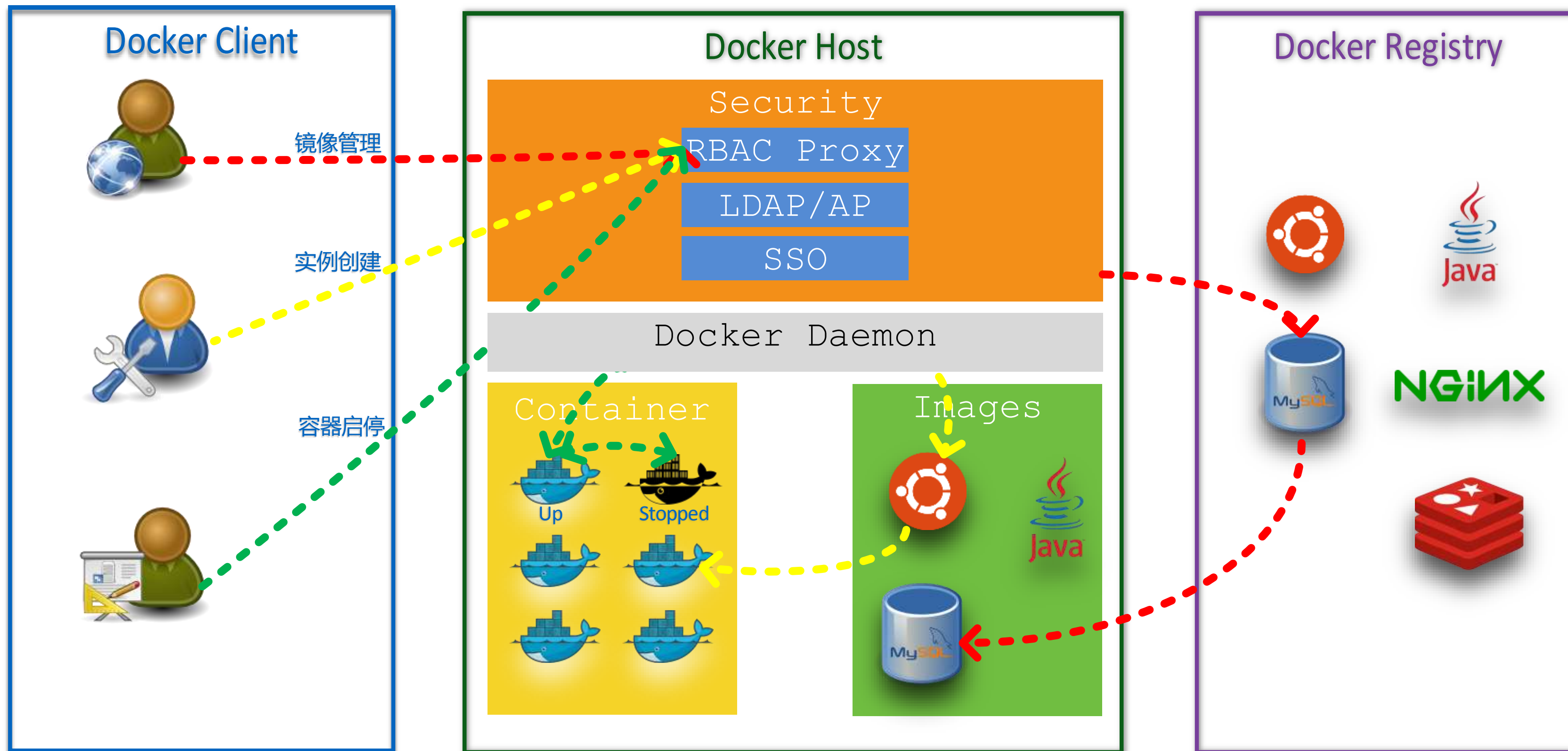
- 利用Mount Namespace的只读挂载，将虚拟网络通讯转换为共享socket文件后设计而成的一种模块间单向通信方式
- 由于网络通信被阻断，攻击者必须逐个攻陷安全模块才可以完整的应用控制权，然而在只攻陷单向通路时会造成整个通路的阻塞，从而阻断攻击者

模块纵深



- Broker DB Module是最为关键的数据
- 只能通过内置安全业务逻辑的数据访问Module读写DB
- 通过硬件加密卡处理后再把密文存入DB
- 如果未授权开箱被触发，带密钥自毁功能的硬件加密卡可以销毁密钥

为Docker增强RBAC机制

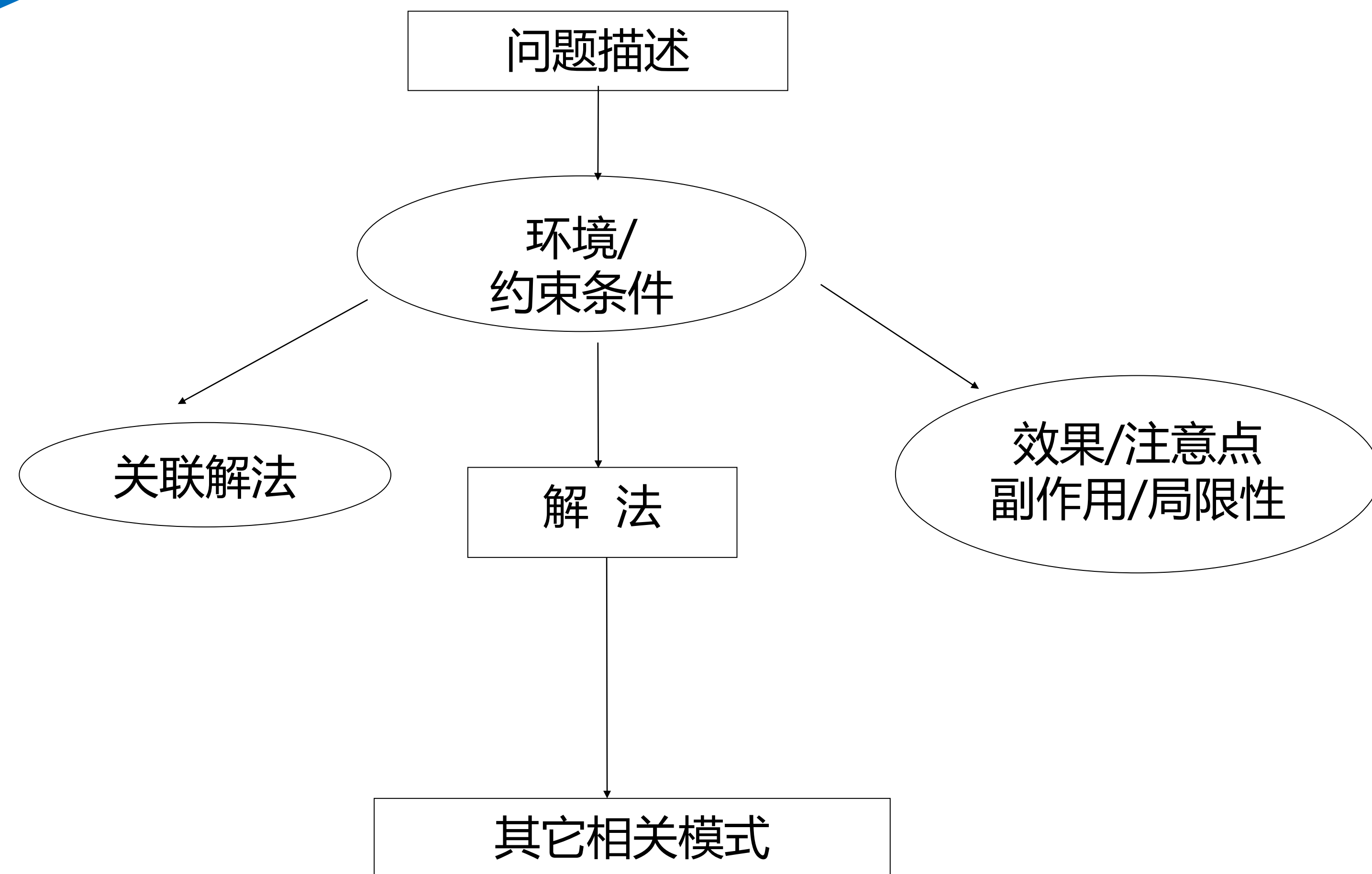


- 截至Docker最新版本1.12.5，对Docker管理缺乏分权授权机制
- 从部署和运维流程，为Docker增强分权授权策略，在不改动Docker的前提下，把RBAC机制加入到Docker的运维流程
- 在不改变Docker使用方式的情况下，通过使用RBAC proxy的方式为docker添加访问控制功能
- 角色权限划分：
 - 镜像管理
 - 实例创建
 - 容器启停

PART
03

可复用模式输出

关于设计模式发展

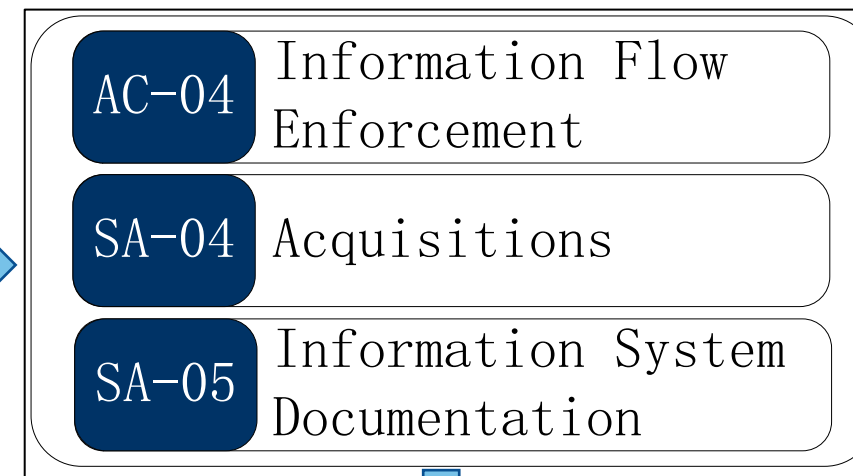
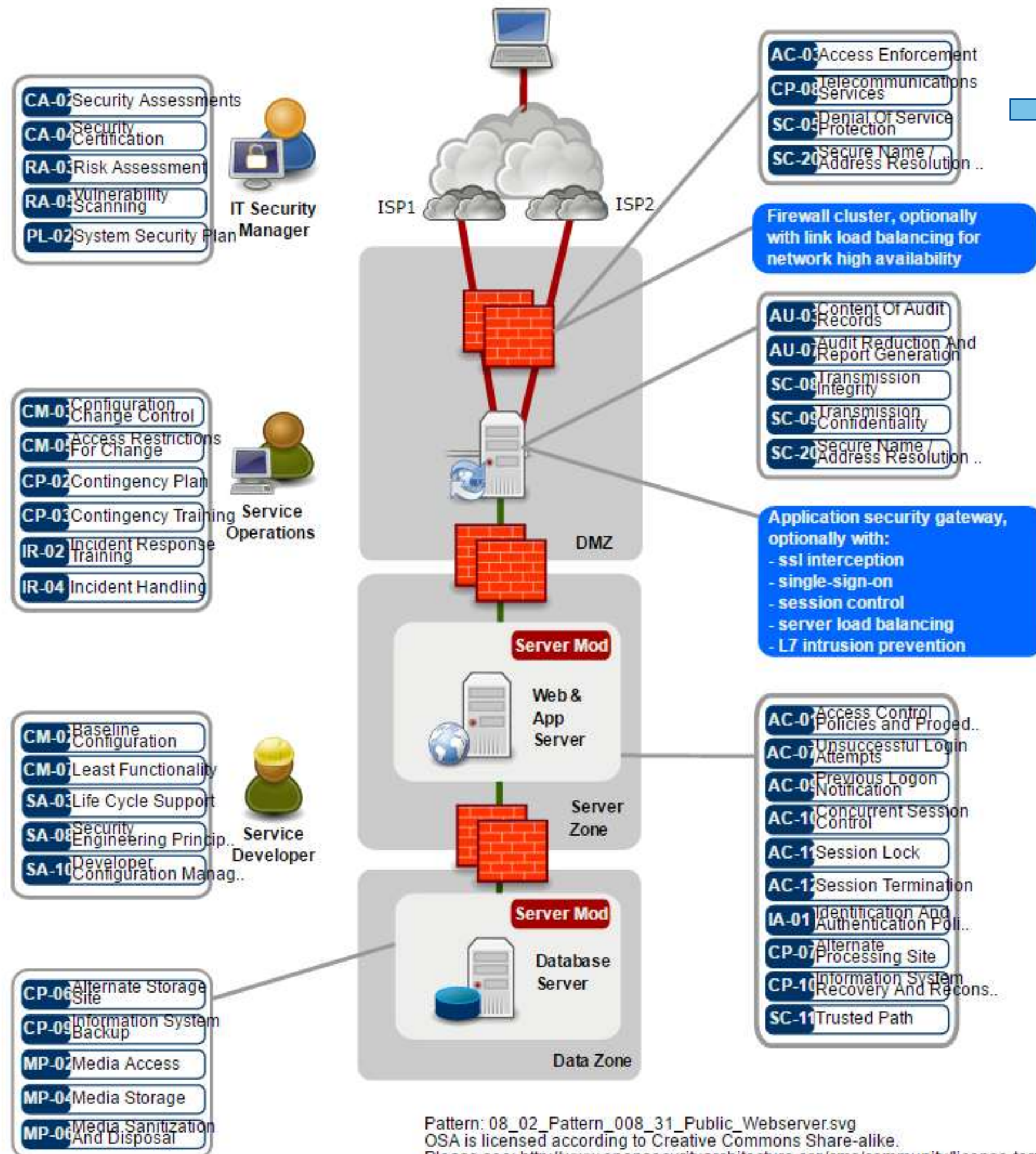


- 历史性著作《设计模式：可复用面向对象软件的基础》一书中描述了23种经典面向对象设计模式，创立了模式在软件设计中的地位。
- 模式经典定义：每个模式都描述了一个在我们的环境中不断出现的问题，然后描述了该问题的解决方案的核心，通过这种方式，我们可以无数次地使用那些已有的解决方案，无需再重复相同的工作。——*模式之父Alexander*
- 模式：A pattern is a solution to a problem in a context——**模式是在特定环境中解决问题的一种方案。**——*Martin Fowler*

Open Security Architecture介绍

SP-008: Public Web Server Pattern

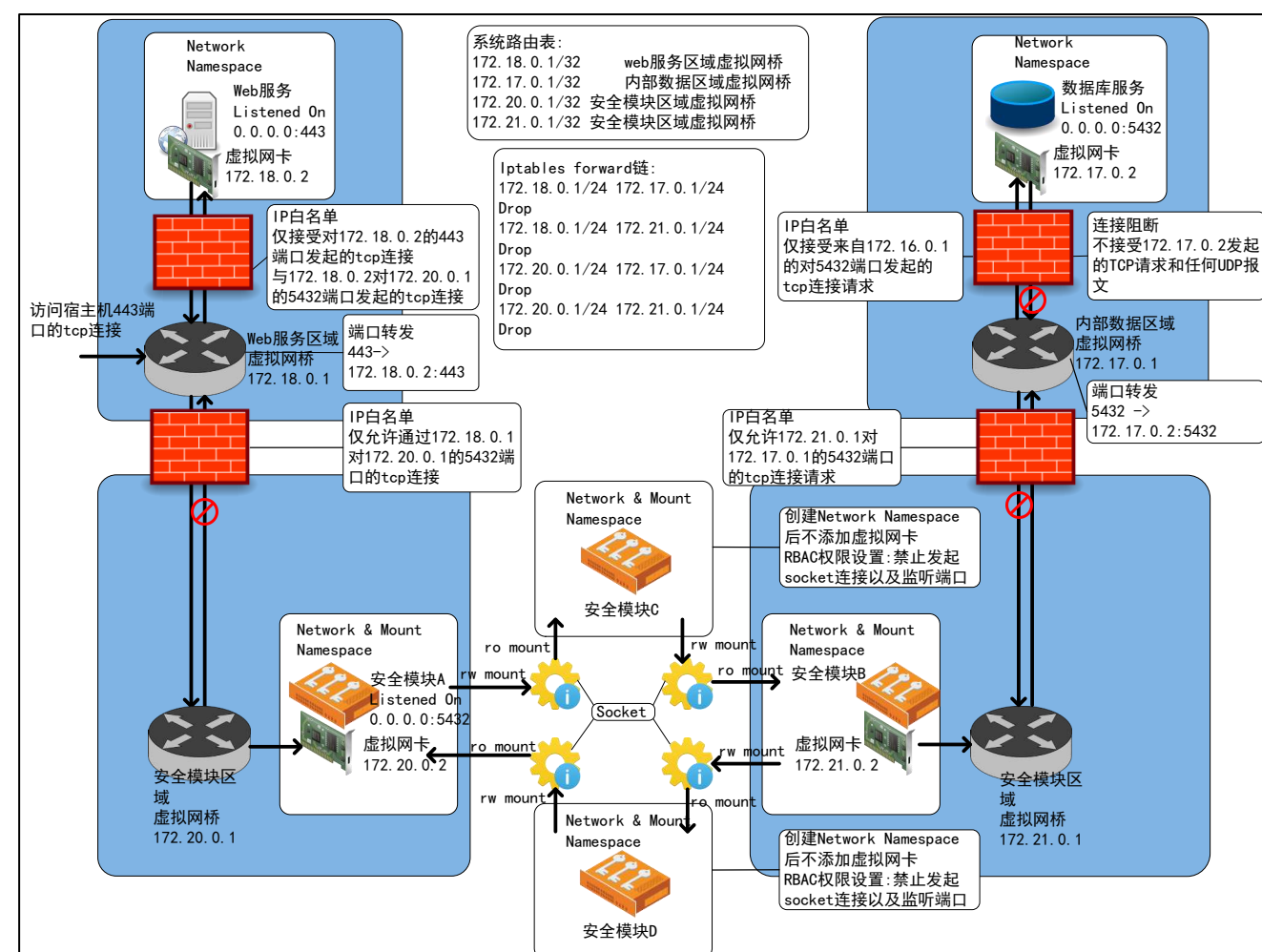
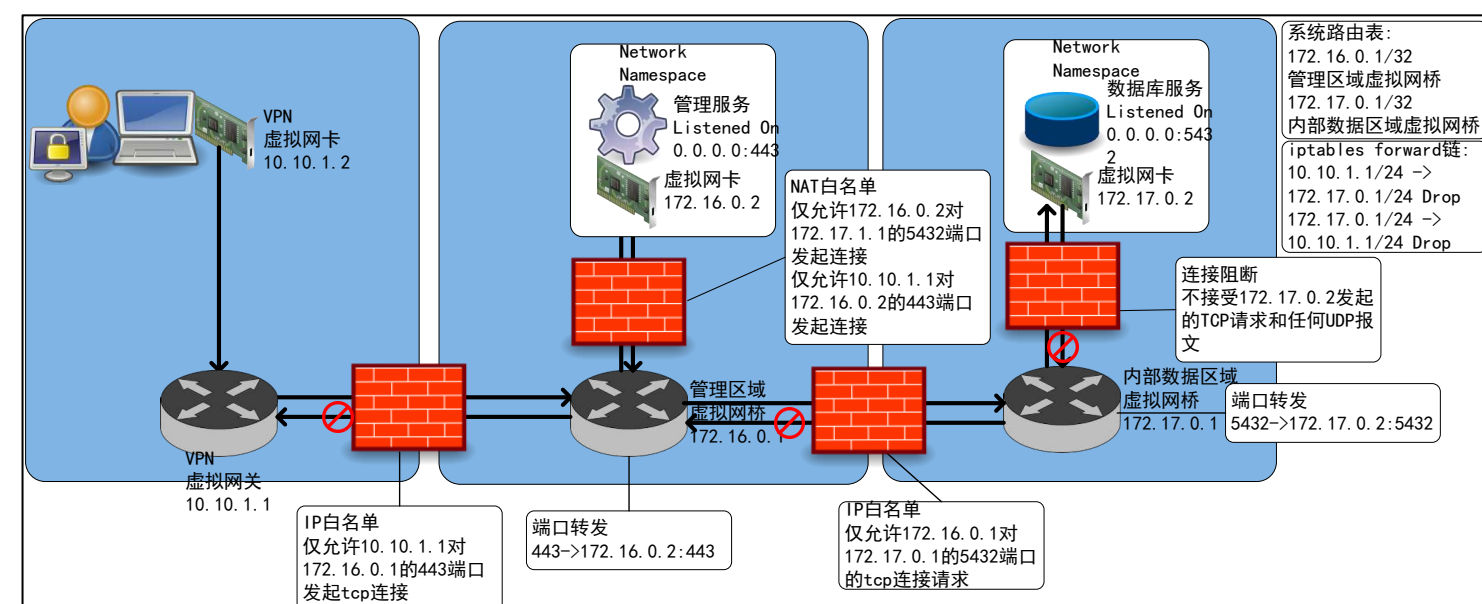
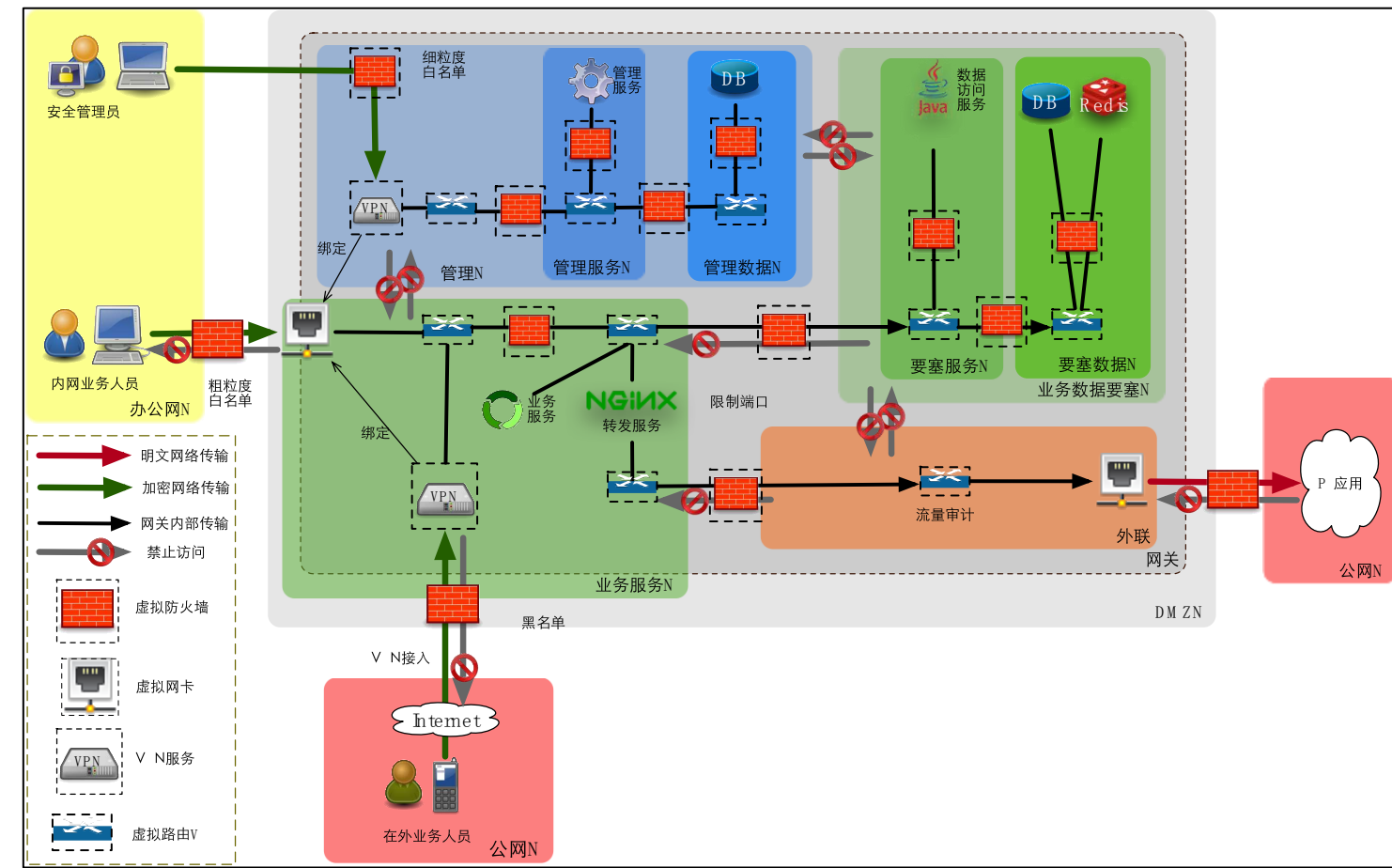
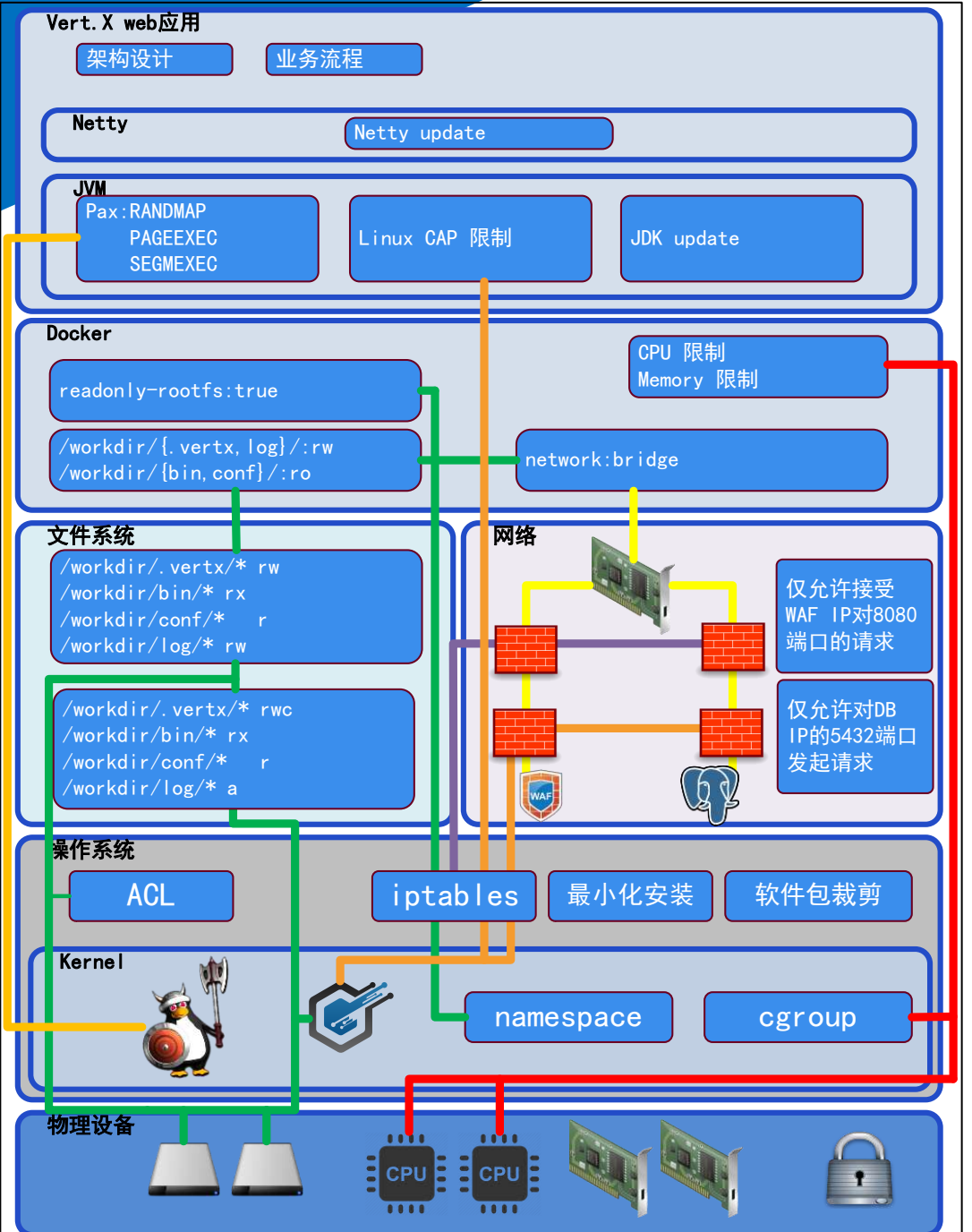
Diagram:



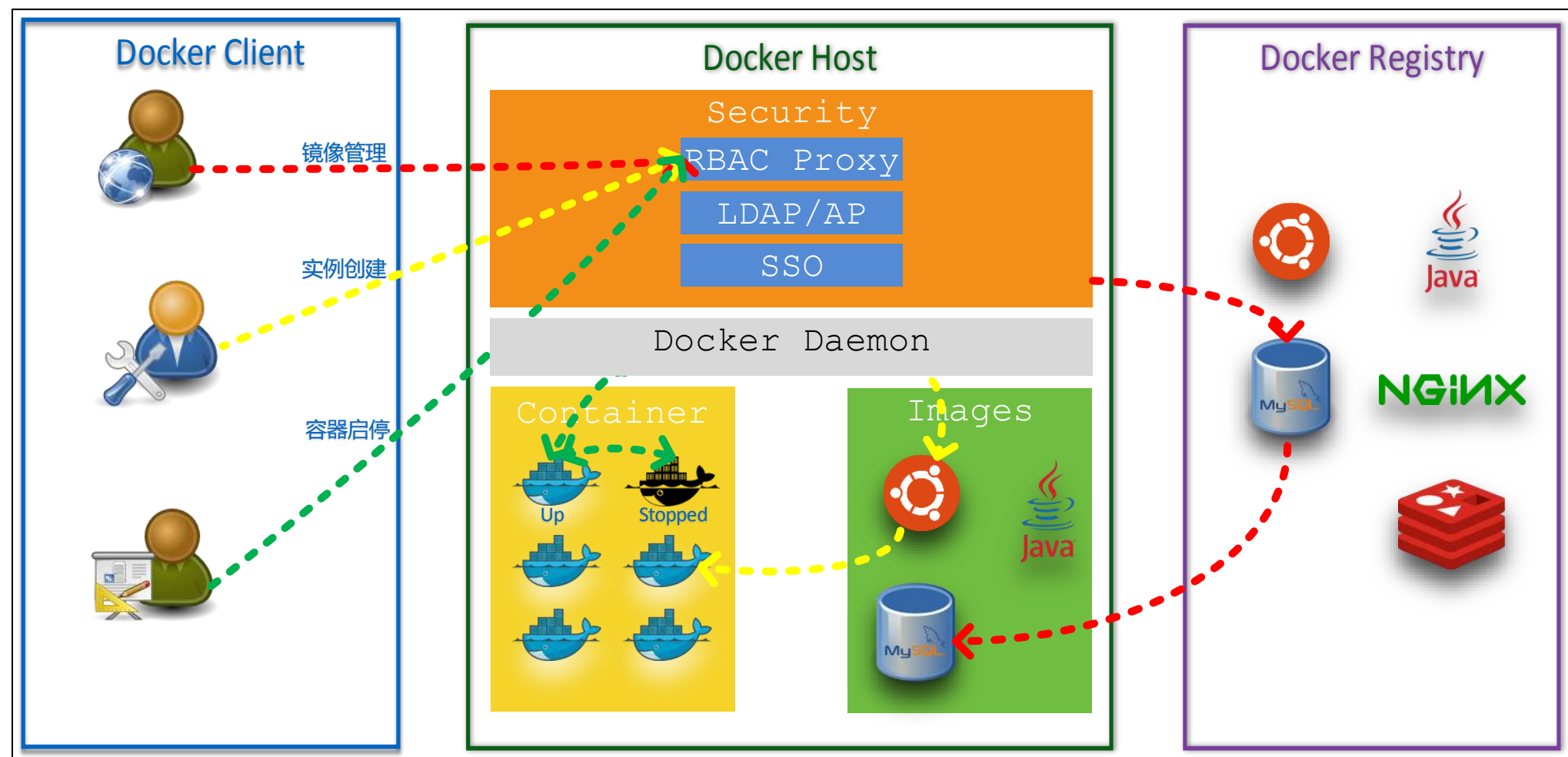
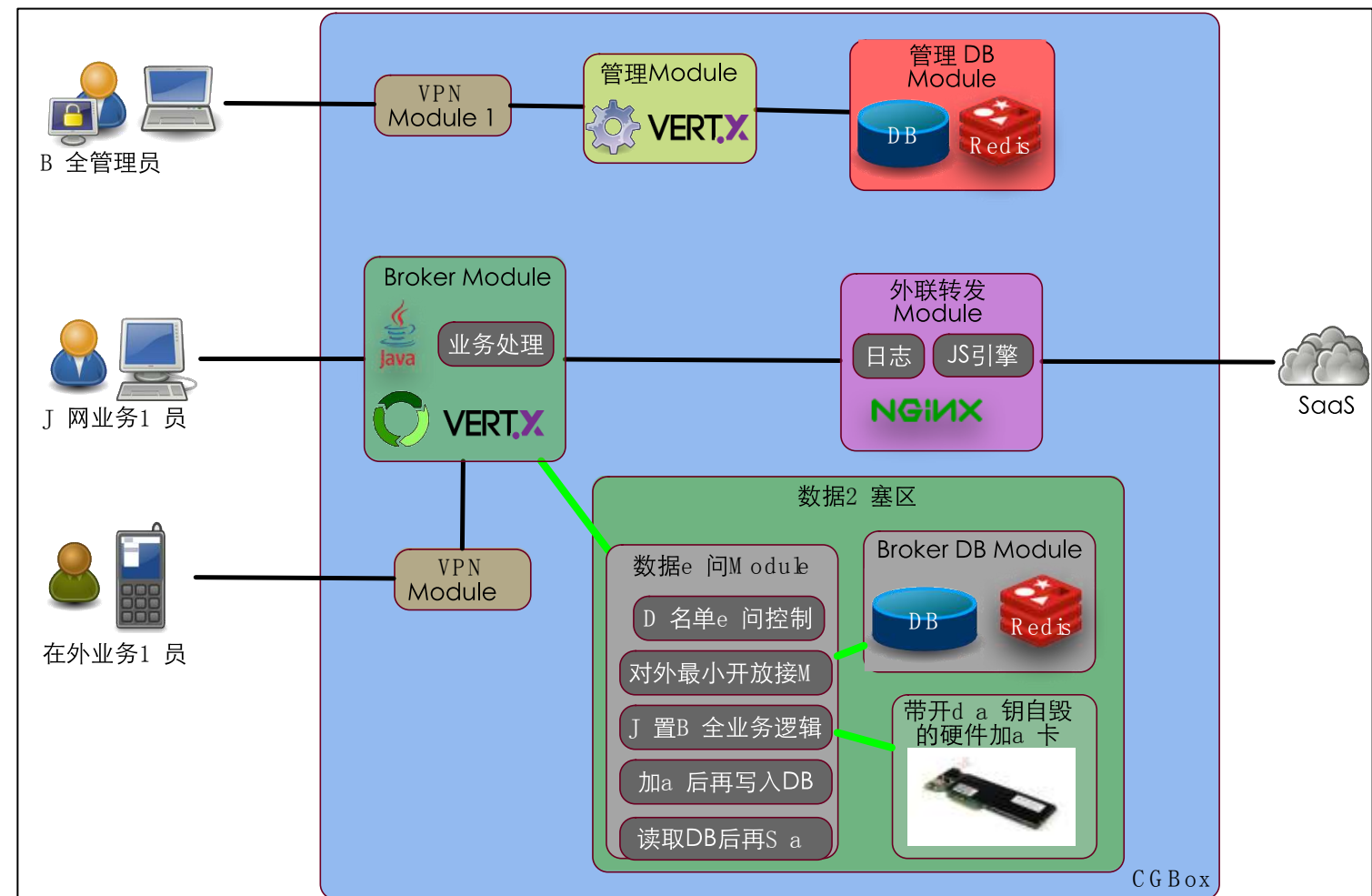
AU-01 Audit And Accountability Policy And Procedures
AU-02 Auditable Events
AU-03 Content Of Audit Records
AT-01 Security Awareness And Training Policy And Procedures
AT-02 Security Awareness
AT-03 Security Training
CP-01 Contingency Planning Policy And Procedures
CP-02 Contingency Plan
CP-03 Contingency Training
PL-01 Security Planning Policy And Procedures
PL-02 System Security Plan
PL-03 System Security Plan Update

- OSA provides a standard design template for common use cases that already show the control you need.
- OSA provides an industry standard architecture that supplies and consumers can both adopt.
- OSA provides a common control catalog mapped against relevant frameworks and standards reducing duplication and improving understanding.

探索可复用的塔防模式



- 我们相信贯彻“面向失效的设计”才能带来有效防护，进而带来安全的价值。
- 为了更好的输出这种价值给用户、安全产业同行、应用生态，我们尝试把这些安全设计提炼成“塔防模式”，让别人能在类似问题域场景下可以借鉴参考。
- 探索模式
 - 应用的技术栈层次纵深
 - 双口网关的网络分区纵深
 - 多模块间的虚拟网络纵深
 - 模块间虚拟网络单向通信
 - 针对数据要塞的模块纵深
 - Docker运维的RBAC防护
- 期待更多同行一起参与





谢谢聆听

THANK YOU FOR YOUR ATTENTION