从运维系统的开发谈安全架构设计

段继刚

<duanjigang1983@gmail.com>

SACC2011

想要说些什么？

- 系统架构师大会-应用系统安全专场
- **应用系统 架构设计 安全设计**
- 1)系统工具cmtk和运维平台opslob介绍
- 2)opslob设计之旅
- 3)opslob中的安全设计

前言

- 运维人员是一伙什么样的人？《运维人员之歌》

在那山的那边海的那边有一群人搞数据中心，

他们苦逼又艰辛，他们通宵接报警，

他们衣冠不整满眼通红奔去IDC，

但是回来他们还要做CASE STUDY Ou SA伤不起，

Ou OP伤不起，他们齐心协力累死累活，

上线了十万台机器，但是他们拿不到百万奖金

前言

- 运维人员的工作最大的特点：维护的机器数量多。。(一台随便折腾)
- 我的第一个运维工作：2008奥运会保障
- 运维工作通常的内容:
  - 批量执行命令或者脚本
  - 保持一批机器配置文件或者系统状态的一致
  - 大半夜，给一批机器同时去安装一个软件包，更新一个文件等等

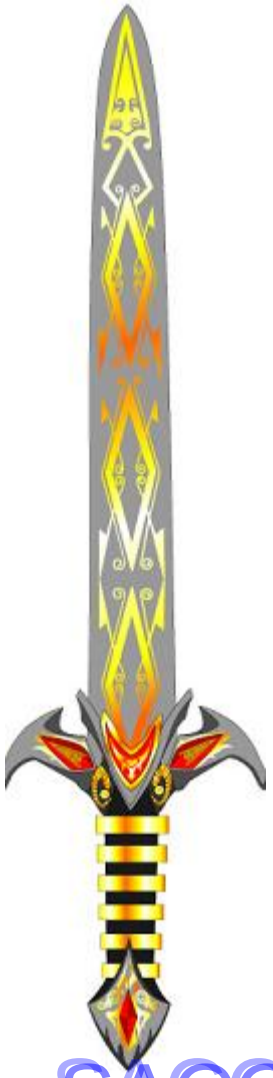
前言

- 运维人员的工作中不爽的地方:

- 1) 每个机器都得账号登录，累死了
- 2) 那么多机器要管却不能root访问
- 3) ssh功能很强大用起来总觉得不是那么爽
- 4) 今天执行脚本，明天发布，后天软件升级。。烦得要死

寻找一种方法或者一个利器来让人更轻松些:

- 第一版: cme_scanner到cmtk
- 第二版: cmtk到opslob



1/3 cmtk介绍

- cmtk:运维工具:
- 1):类似ssh的网络工具: 由cmtk命令行+cmtkd服务组成
- 2):无账户，基于源IP授权的访问控制策略
- 3):并发批量访问
- 4):日志审计

1/3 cmtk用法

- cmtk:功能:
- 1):执行命令
- **cmtk –p port –h host –c “command” –t timeout**
- **cmtk –p port –f hostlist.conf –c “command”**
- 2)传输小文件
- **cmtk –p port –h host –u src_file –d dst_file**
- **cmtk –p port –f hostlist.conf –u src_file –d dst_file**
- cmtk特点:
- 轻便，不依赖于外部组件
- 用法简单明了
- 输出结果多样(可读和程序分析格式)

1/3 cmtk演示

```
[root@localhost SPECS]# cmtk -h 192.168.1.101 -c "unset LANG;date"
(192.168.1.101):[192.168.1.101]
Sat Sep  3 07:26:01 CST 2011

[root@localhost SPECS]# cmtk -f dev.txt -c "unset LANG;date"
(127.0.0.1):[127.0.0.1]
Sat Sep  3 07:26:14 CST 2011

(192.168.1.101):[192.168.1.101]
Sat Sep  3 07:26:14 CST 2011

(localhost):[127.0.0.1]
Sat Sep  3 07:26:14 CST 2011
```

192.168.1.101:22

```
[root@localhost SPECS]# cmtk -h 192.168.1.101 -u 1.txt -d /tmp/2.txt
upload file [1.txt] to destination [/tmp/2.txt]:
(192.168.1.101):[192.168.1.101]
success

[root@localhost SPECS]# cmtk -f dev.txt -u 1.txt -d /tmp/2.txt
upload file [1.txt] to destination [/tmp/2.txt]:
(192.168.1.101):[192.168.1.101]
success

(localhost):[127.0.0.1]
success

[root@localhost SPECS]# cmtk -f dev.txt -c "cat /tmp/2.txt"
(192.168.1.101):[192.168.1.101]
this is 1.txt

(localhost):[127.0.0.1]
this is 1.txt
```

1/3 opslob介绍

- opslob概况:
- 1):结构：封装于cmtk之上的运维开发平台
- 2):概念：策略, 事务, 应用，主机和群组
- 3):特点：灵活，功能采用可配置的插件方式
- 4):人机界面：web和命令行两种用户界面

- opslob应用范畴:
- 1): 主机配置文件刷新
- 2): 定时软件发布
- 3): 系统配置状态维护
- 4): 日常系统管理
- 5): 应用系统二次开发

1/3 opslob用法

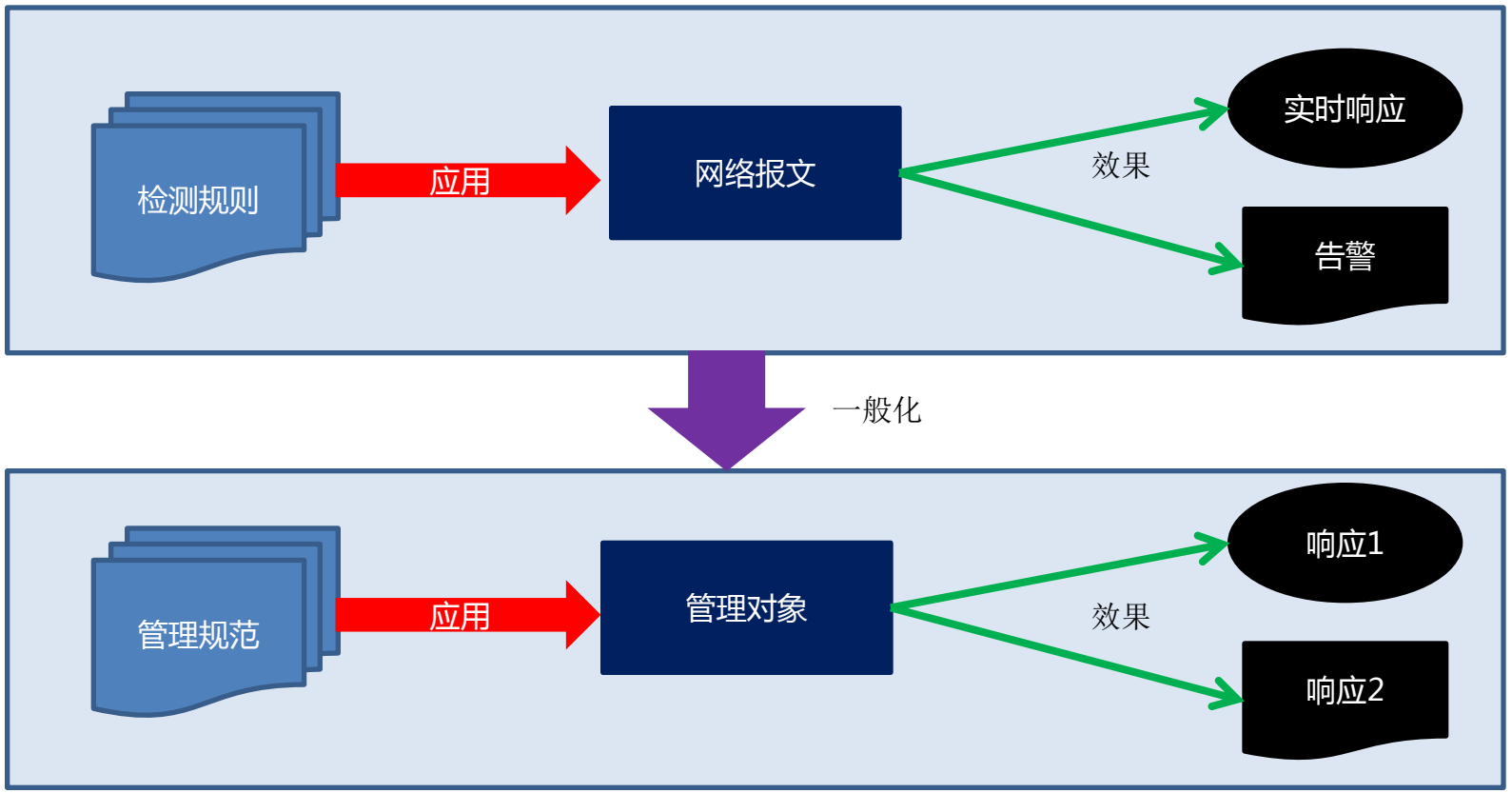
- 以安装mysql服务器为例:
- **1) 在控制中心 使用命令行polctl创建多个策略**
- A: mysql-rpm策略：负责上传mysql安装包
- B: install.sh负责安装mysql
- C: config.txt负责生成my.cnf文件
- D: Start.sh启动mysql服务
- **2) 使用transctl命令行创建事务install_mysql,其中包含策略A->B->C->D**
- **3) 使用devctl把要操作的主机添加到群组group1中**
- **4) 使用appclt命令创建application把事务install_mysql应用到群组group1上**

2/3 opslob设计之旅

- 结构设计可能并不好。。
- 讲述方式不一定清晰。。
- 你可能会听睡着，但是请不要离开。。
- 唯一目标:分享opslob设计时思维的过程

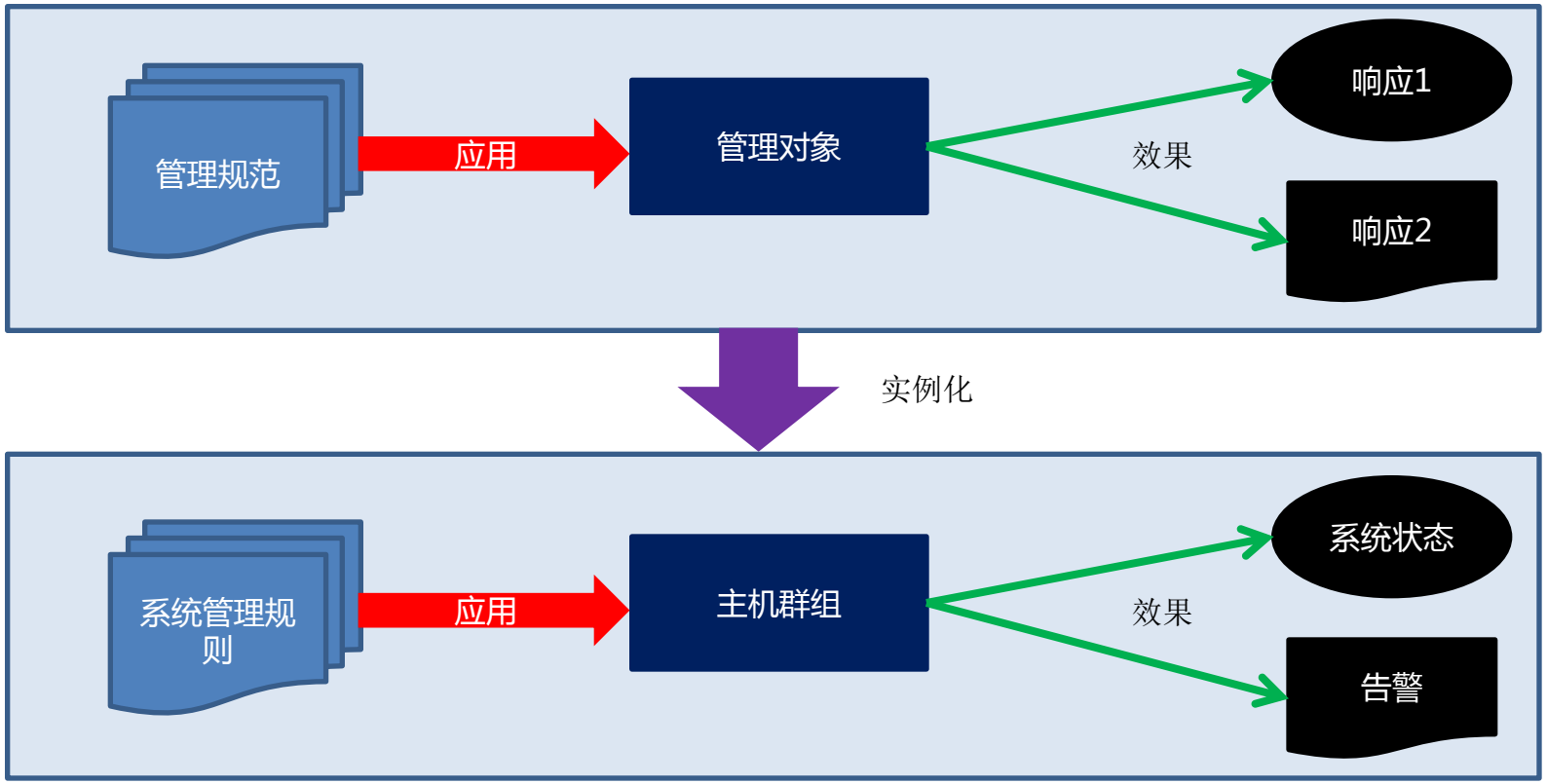
2/3 opslob设计-设计思路

- 安全产品(防火墙 或者IDS)的基本模型:



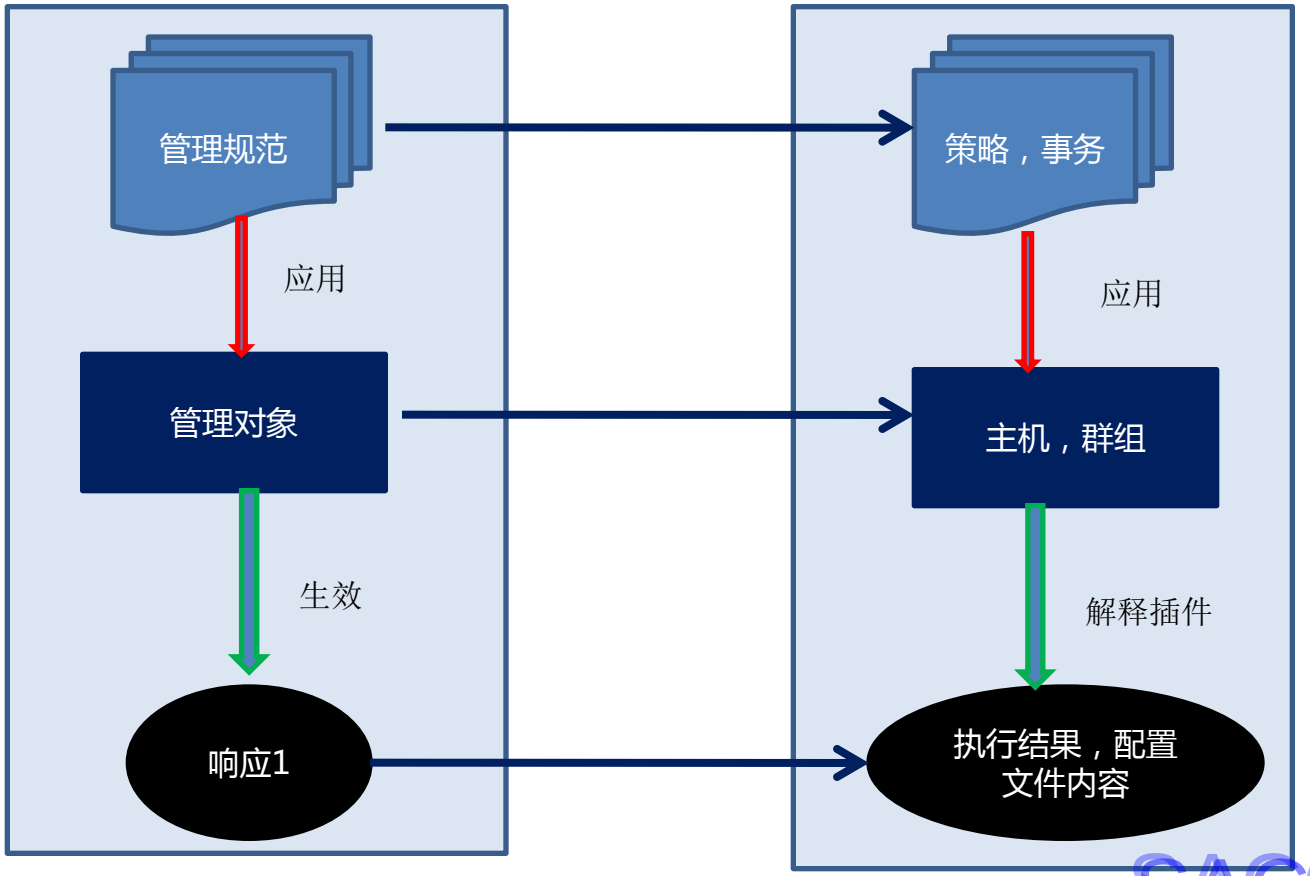
2/3 opslob设计-设计思路

- opslob模型的产生（特殊->一般->特殊）



2/3 opslob设计-设计理念

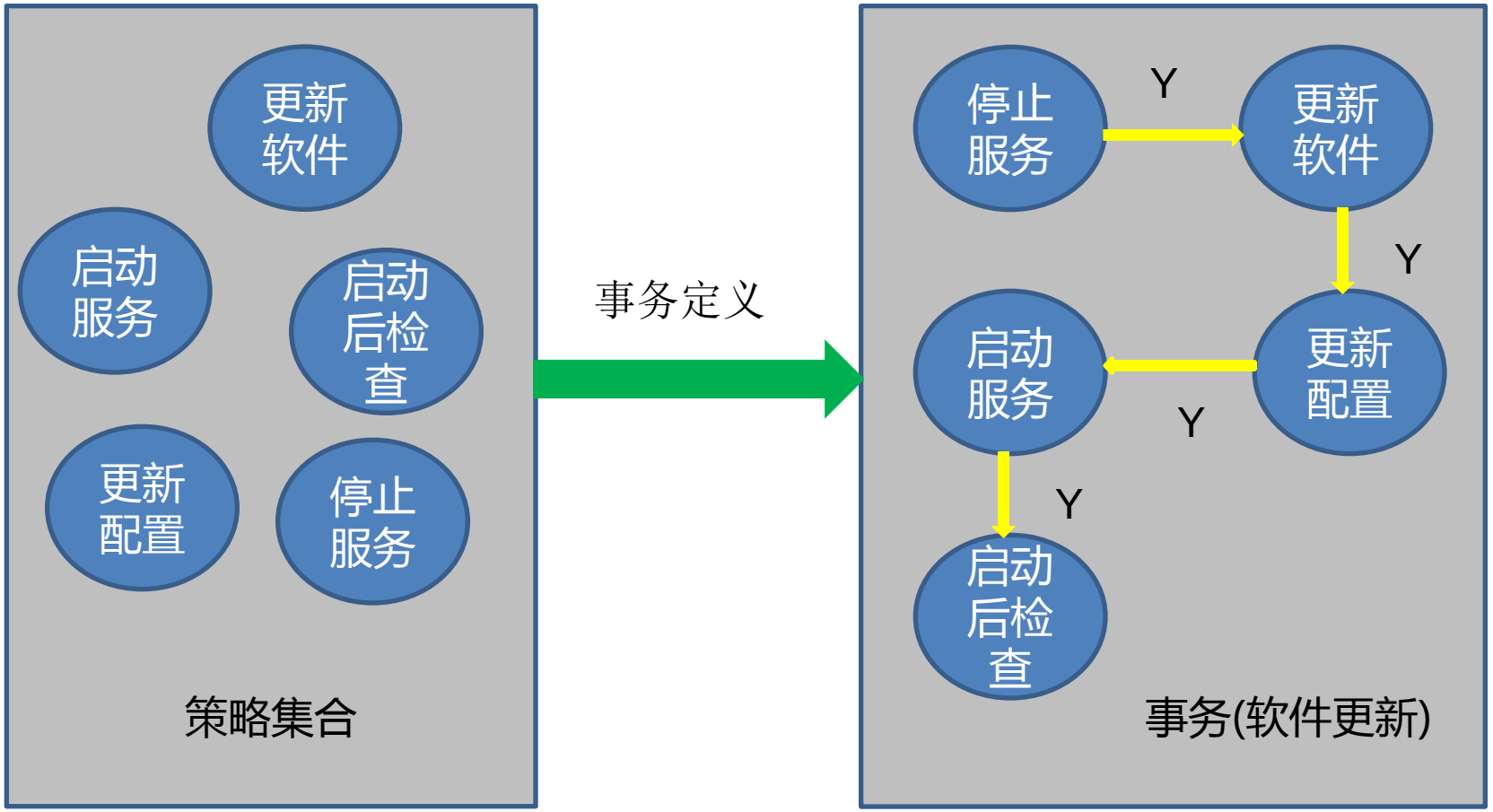
- opslob模型的进一步具体化



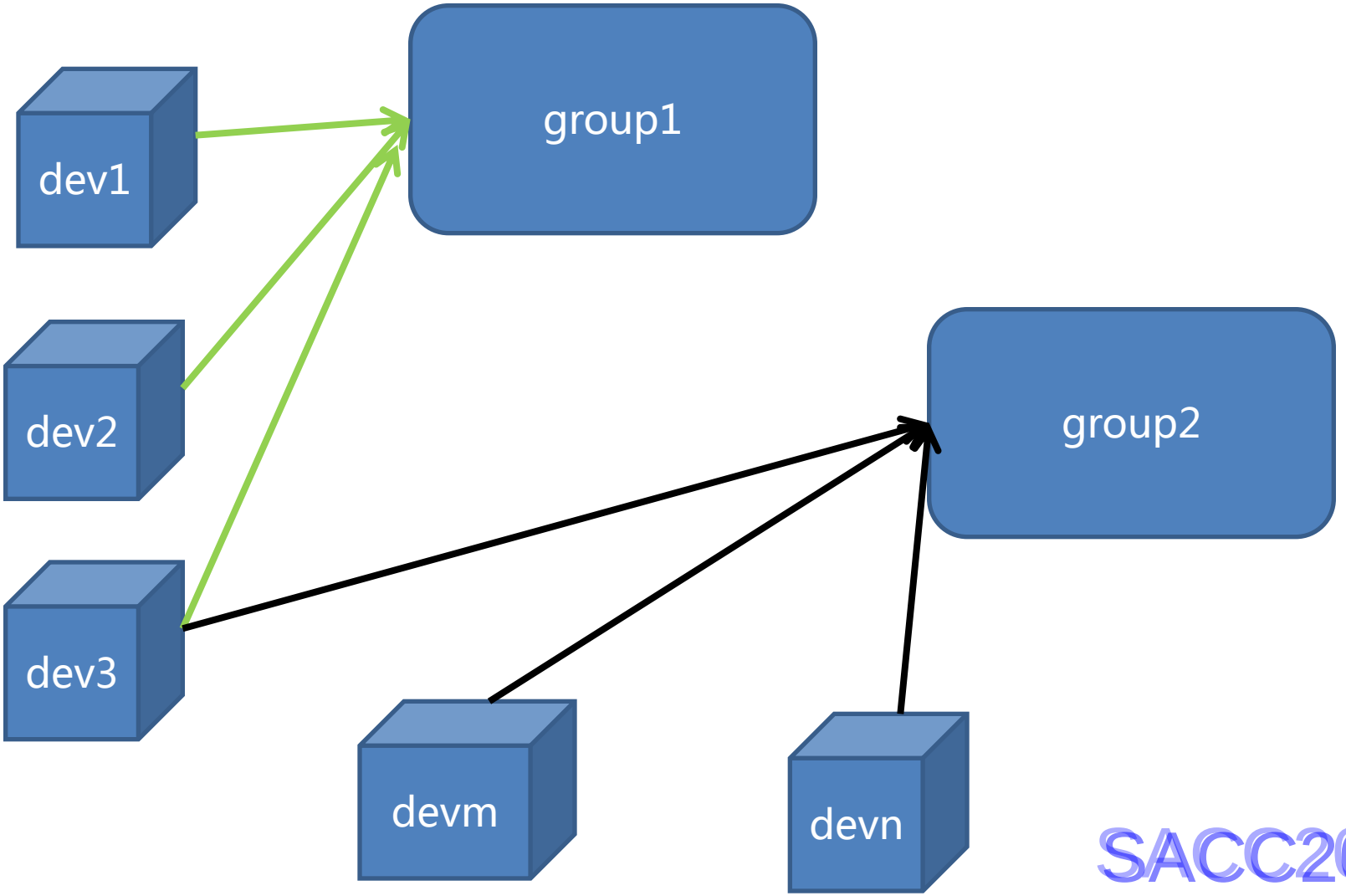
2/3 opslob设计-基本元素

- 策略(policy)-一个操作的最小定义原语
- 事务(transaction)-若干个策略组成的有序链表，各个节点之间存在依赖关系
- 应用(application)-将一个事务应用到若干个群组上去执行
- 设备(device)-设备，待管理的主机
- 群组(group)-待管理的主机组成的集合

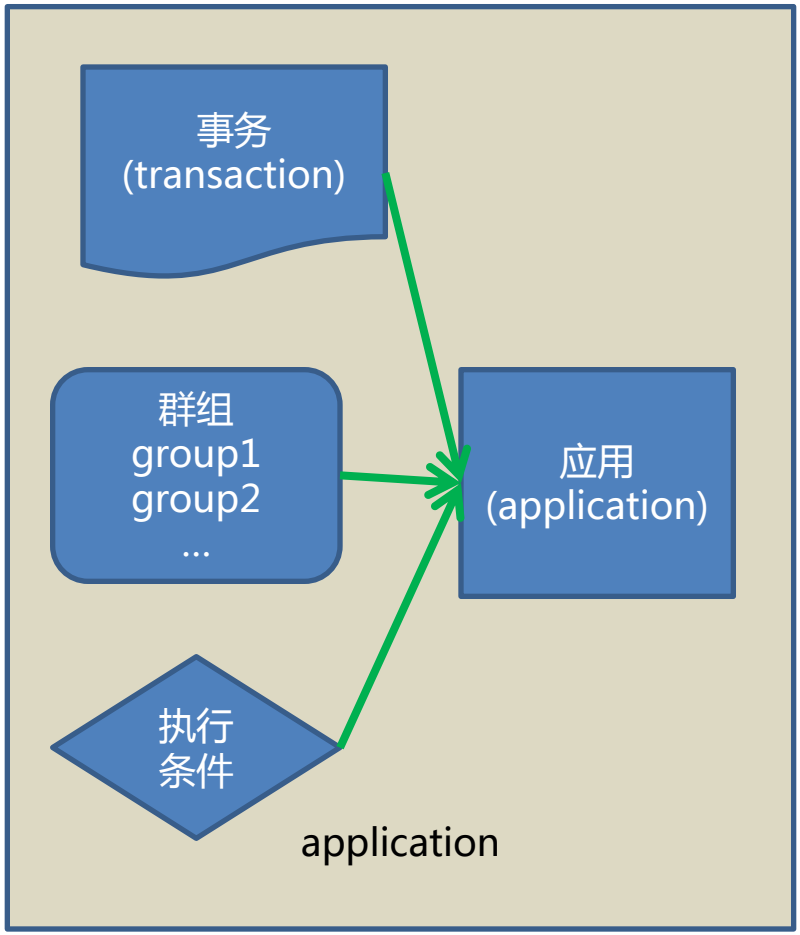
2/3 opslob设计-策略与事务关系

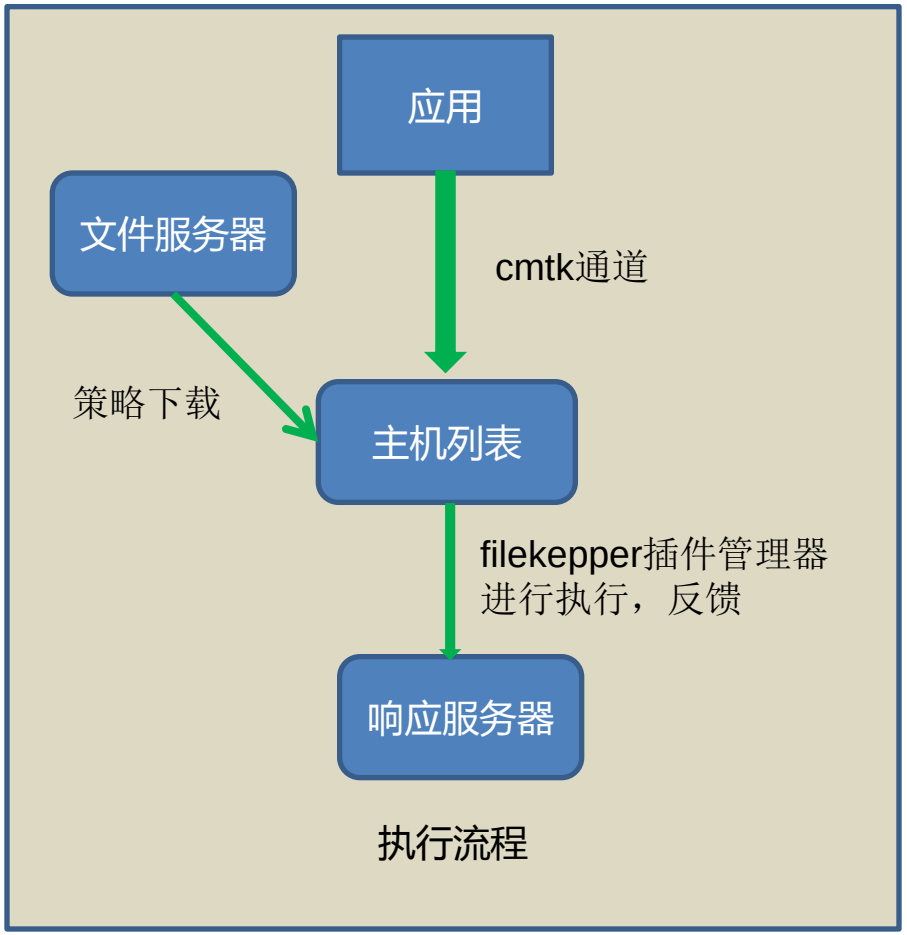


2/3 opslob设计-设备与群组关系



2/3 opslob设计-应用构建与执行

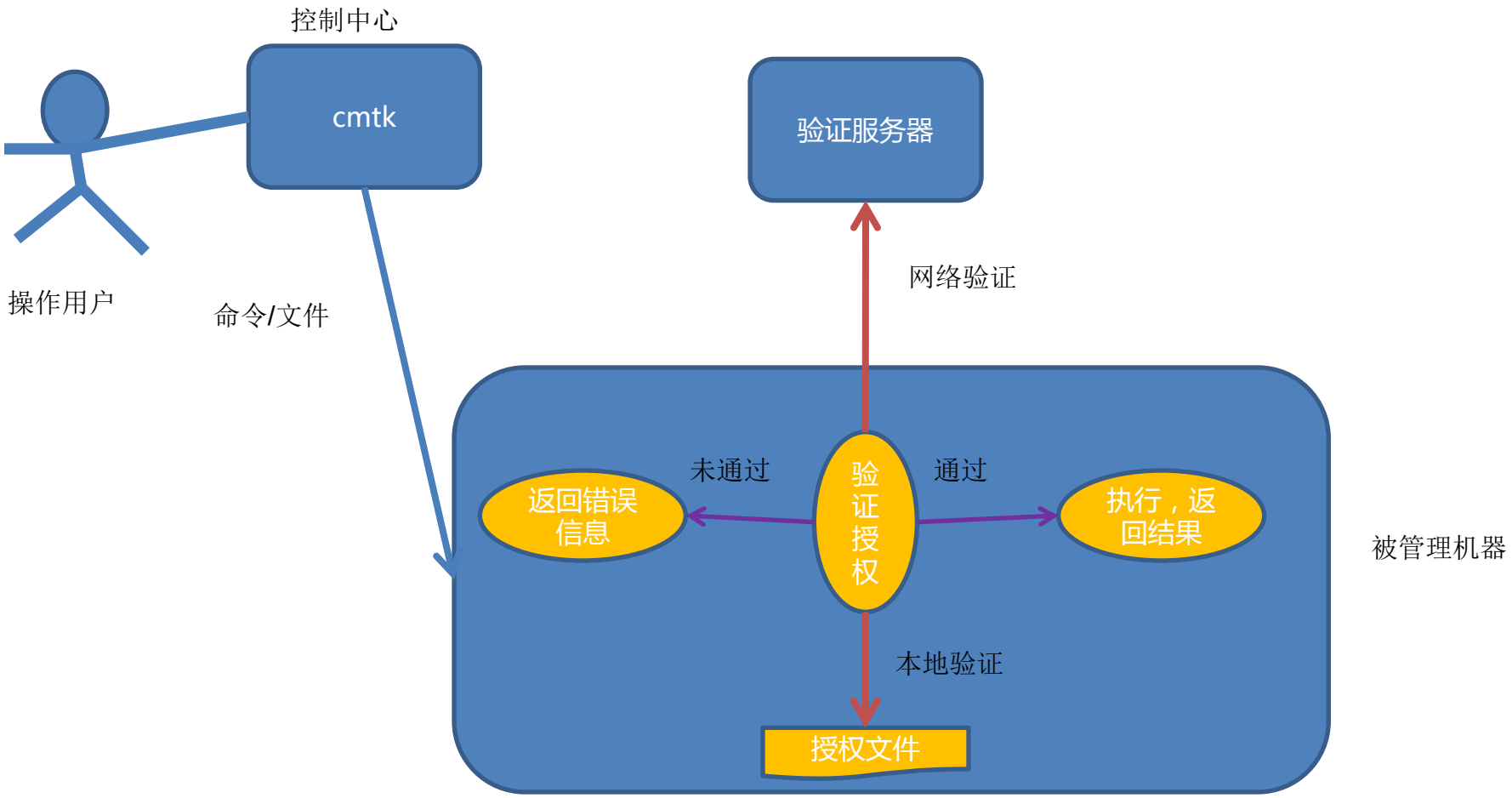




3/3 opslob设计中的安全

- cmtk中存在的安全隐患？
- opslob中的安全隐患？

3/3 安全的系统设计-cmtk安全解决



3/3安全的系统设计-cmtk安全解决

- cmtk已经是一个安全的工具了吗？还遗忘了哪些？

执行的命令?

传输的文件?

- 传输内容上并不能保证安全(从更高层面)
- 期待opslob能够解决此问题

3/3安全的系统设计-opslob

- opslob构成
- 1)控制中心命令行:
- polclt,transctl, devctl和appctl:
- 2)控制中心web界面:
- 提供和命令行相同的功能
- 3)cmdserver:
- 控制中心负责应用消息的分发
- 4)cmtkd:
- 控制中心和各个被管理机器上的通讯工具
- 5)filekeepd:
- 客户端插件管理程序，负责策略的解释和执行

3/3安全的系统设计-opslob

- opslob存在的安全隐患

1）策略的安全-恶意，不安全策略

2）事务的安全-恶意，不安全事务

3）应用的安全-错误，不合理应用

业务构建过程

4）传输过程的安全-报文劫持，数据修改

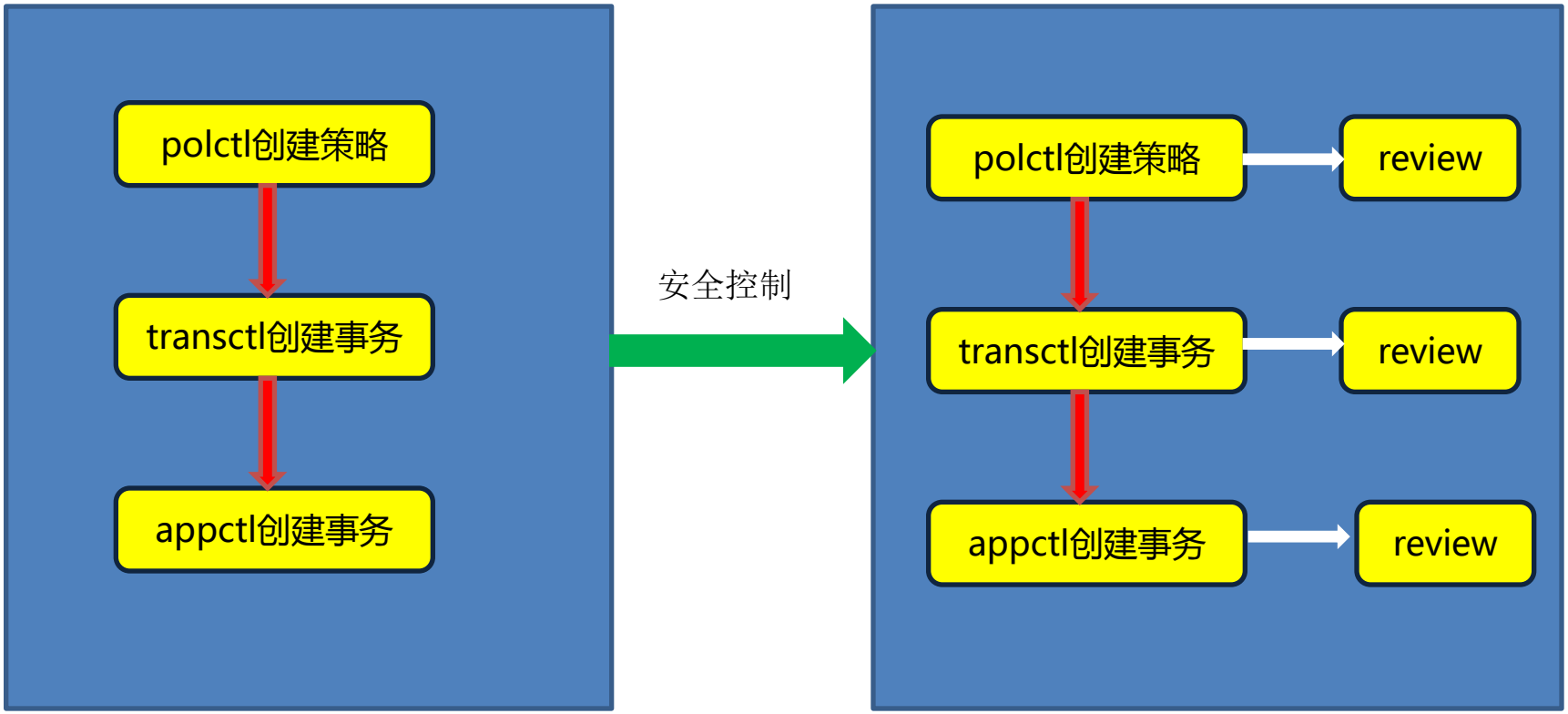
网络传输

5）执行过程的安全-策略内容校验

客户端执行

3/3 安全的系统设计-opslob

业务构建过程安全解决方案:通过流程控制来实现



3/3 安全的系统设计-opslob

网络传输过程安全解决方案:自定义协议和加密

1)采用ICE通讯中间件传输数据，自定义协议不易解析

2)ICE支持ssl加密传输，能达到更高级别的安全传输

3/3 安全的系统设计-opslob

客户端执行过程安全解决方案:签名与MD5校验

1)从任务列表中提取transaction和policy list信息

2)按照transaction中policy的顺序从文件服务器下载policy

3)每下载一个policy，核对该policy的MD5SUM和签名与任务列表中的是否一致。

4)下载完policy list并核对无误后，按照条件执行，并返回结果

3/3 安全的系统设计-opslob

- 总结
- 1):从 上层业务->网络传输->执行过程
- 不同层面保障安全
- 2):opslob对cmtk的调用只采用了文件传输功能
- 3):业务层安全解决方案保障了传输文件的安全
- 4):解决了自身的安全隐患，也解决了cmtk在使用时产生的隐患

结束: 请提问?

- email: duanjigang1983@gmail.com
- msn: duanjigang@hotmail.com
- 请关注 **chinaunix架构设计版**:
- http://bbs.chinaunix.net/forum-185-1.html
- cmtk测试版在架构设计板块已经提供下载

谢谢