World of Tech WOT

51CTO传媒

WOT 2015 互联网运维与开发者大会

■ 2015年04月10日-11日 ■ 北京珠三角JW万豪酒店

Zabbix运维自动化和基于监控服务调度

北北

Zabbix 自动化标准浅析

一 主机的命名规则

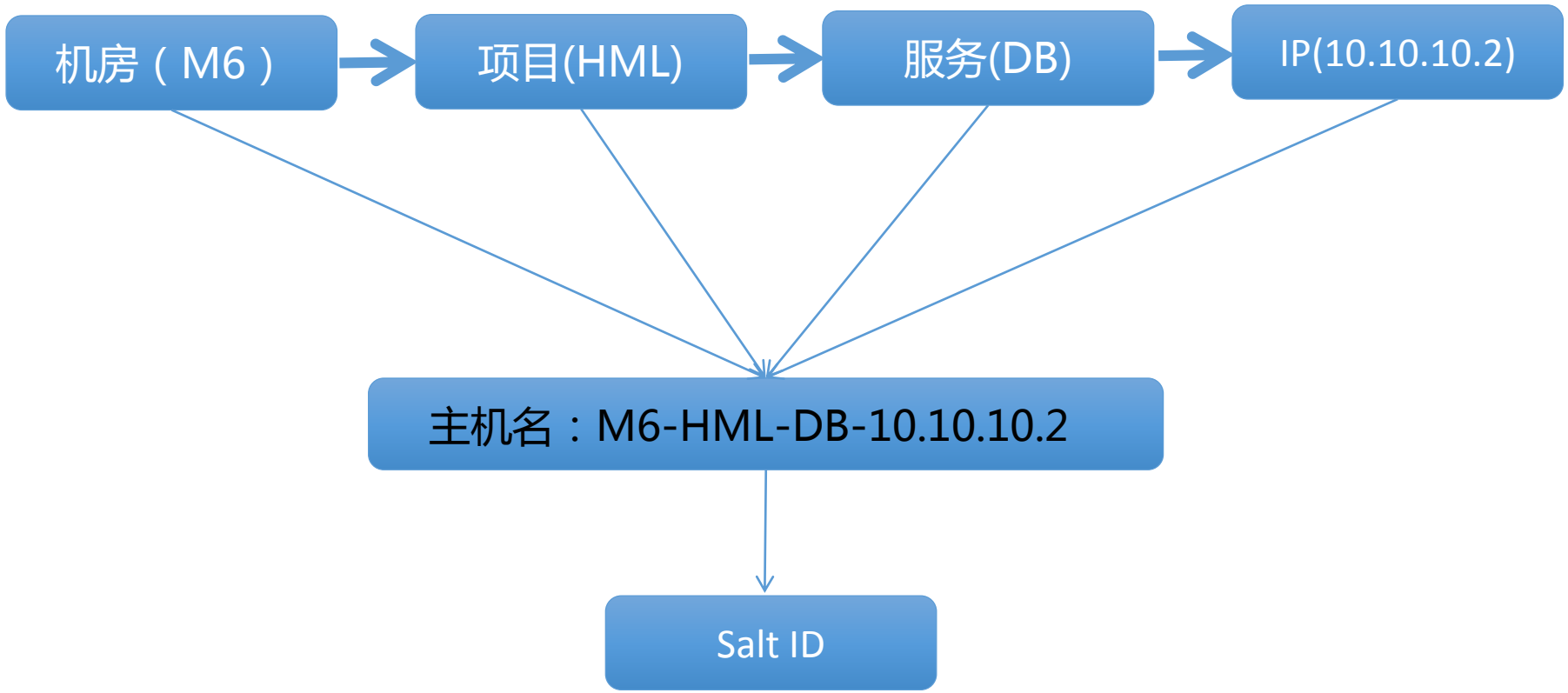
二 机器组和报警分组整合

三 LL(LOW LEVEL)规则

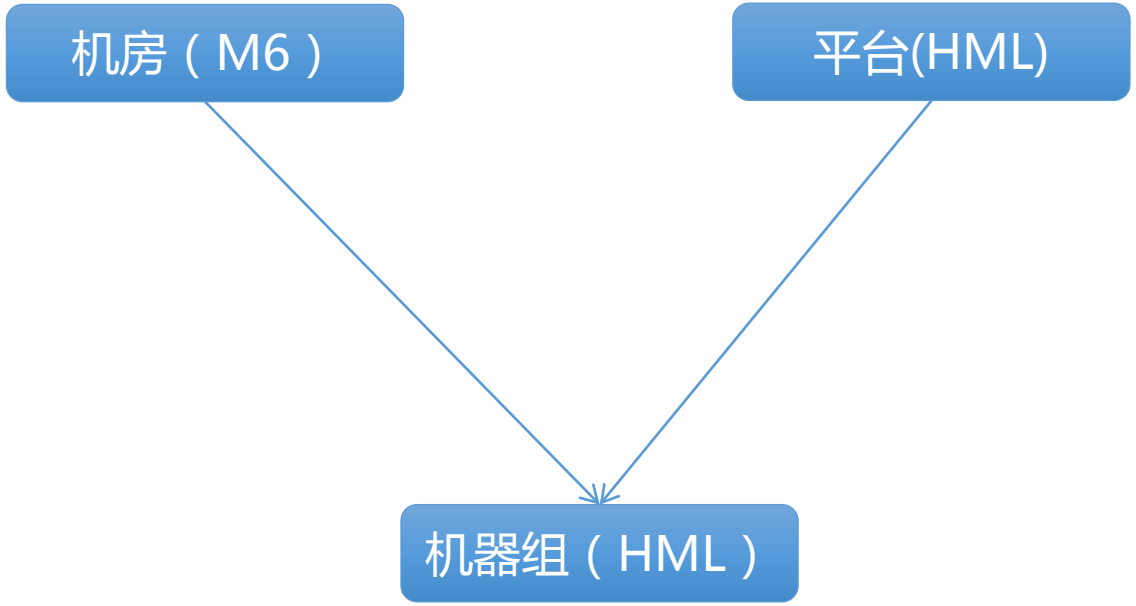
四 报警邮件和短信的分离

五 zabbix api简单利用

一 主机命名规则



二 主机分组



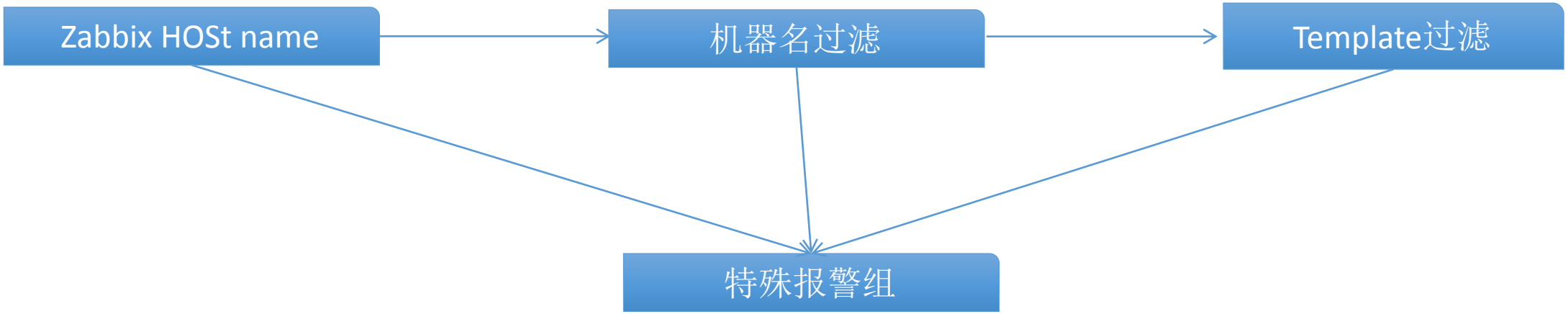
注：主机分组是用机房和平台组合自动生成

三 报警分组

1 根据机器组预先设定好基础报警组 如（HML）

2 根据模版预设定报警组

3 特殊报警组设定 如（DBA)



四 LL(low level discovery)

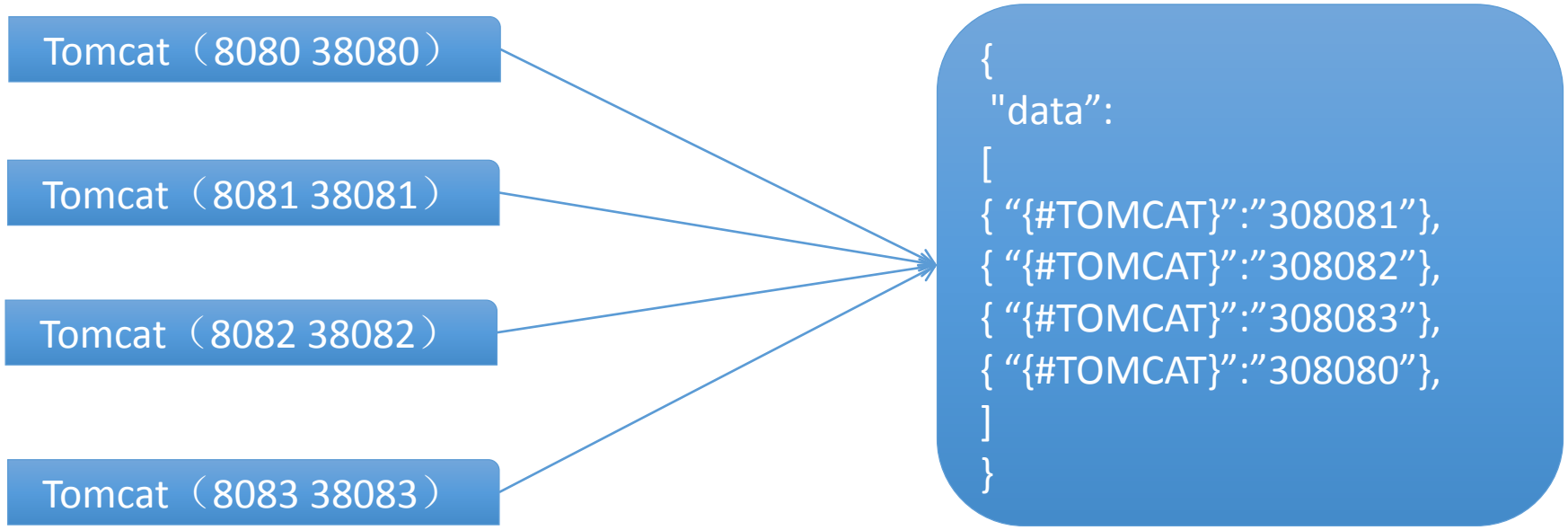
1 文件系统的自动发现.

2 网卡的自动发现.

3 SNMP自动发现.

4 多实例的自动发现.

五 多实例的自动发现一些应用（Tomcat）



注释：808x为tomcat启动端口 3808X是jmx端口

五 多实例的自动发现一些应用（Tomcat）

Class LoadedClassCount_{#TOMCAT}	Class[{#TOMCAT},LoadedClassCount]	3
Class TotalLoadedClassCounT_{#TOMCAT}	Class[{#TOMCAT},TotalLoadedClassCount]	3
Class UnloadedClassCount_{#TOMCAT}	Class[{#TOMCAT},UnloadedClassCount]	3
HeapMemoryUsage.commited_{#TOMCAT}	HeapMemoryUsage[{#TOMCAT},committed]	3
HeapMemoryUsage.max_{#TOMCAT}	HeapMemoryUsage[{#TOMCAT},max]	3
HeapMemoryUsage.used_{#TOMCAT}	HeapMemoryUsage[{#TOMCAT},used]	3
SessionsactiveSessions_{#TOMCAT}	Sessions[{#TOMCAT},activeSessions]	3
Sessions maxActiveSessions_{#TOMCAT}	Sessions[{#TOMCAT},maxActiveSessions]	3
Sessions maxActive_{#TOMCAT}	Sessions[{#TOMCAT},maxActive]	3
Sessions rejectedSessions_{#TOMCAT}	Sessions[{#TOMCAT},rejectedSessions]	3
Sessions sessionCounter_{#TOMCAT}	Sessions[{#TOMCAT},sessionCounter]	3
Threading PeakThreadCount_{#TOMCAT}	Threading[{#TOMCAT},PeakThreadCount]	3
Threading ThreadCount_{#TOMCAT}	Threading[{#TOMCAT},ThreadCount]	3
Threading TotalStartedThreadCount_{#TOMCAT}	Threading[{#TOMCAT},TotalStartedThreadCount]	3
tomcat.{#TOMCAT}	net.tcp.port[,{#TOMCAT}]	3

五 多实例的自动发现一些应用（标准化）

1 基于端口的服务需要有合理多实例规划

SERVICE	Port RANGER
TOMCAT	8080-80XX

2 基于服务名的自动发现

SERVICE	{X1,X2,X3,X4}

基于服务名的发现需要维护一个服务序列,用于自动发现脚本的过滤

五 多实例的自动发现一些应用（杂谈）

The filter can be used to only generate real items, triggers, and graphs for certain file systems. It expects POSIX Extended Regular Expression. For instance, if you are only interested in C:, D:, and E: file systems, you could put {#FSNAME} into “Macro” and "^C|^D|^E" regular expression into “Regexp” text fields. Filtering is also possible by file system types using {#FSTYPE} macro (e. g. "^ext|^reiserfs").
You can enter a regular expression or reference a global regular expression in “Regexp” field.
In order to test the regular expression you can use “grep -E”, for example:
for f in ext2 nfs reiserfs smbfs; do echo $f | grep -E '^ext|^reiserfs' || echo "SKIP: $f"; done

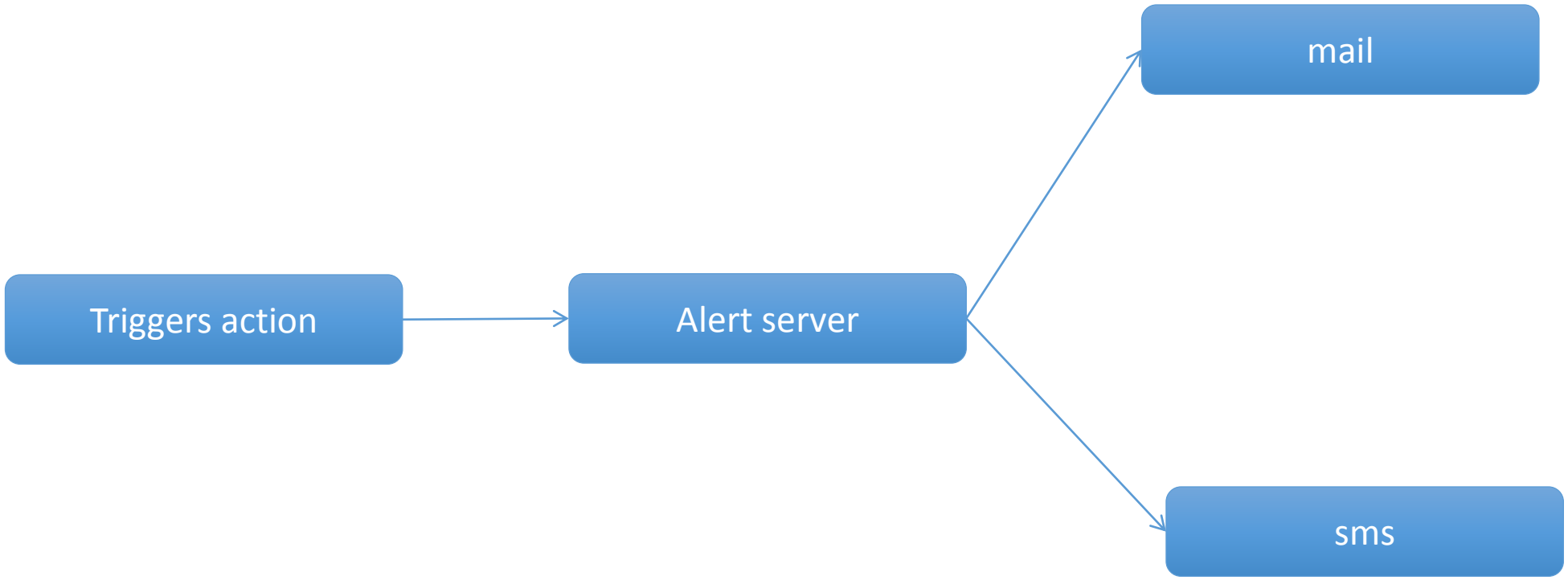
可以用宏的方式实现对服务自动发现的过滤（我没有尝试）

六 报警邮件和短信的分离（传统）

传统的报警模式一般采取*.sh *.py .pl 之类的脚本实现报警，当出现的大批量或者服务器网络延迟时候，会导致报警发送失败，或者zabbix会耗费大量的资源去尝试重新发送，会导致zabbix服务负载过高，会造成无状态假死。

传统报警会导致报警发送后无发记录,也没有办法控制短信和邮件的发送频率。邮件和短信发送要书写大量的action。

六 报警邮件和短信的分离



六 报警邮件和短信的分离(实现)

简单 alert server架构（django实现）

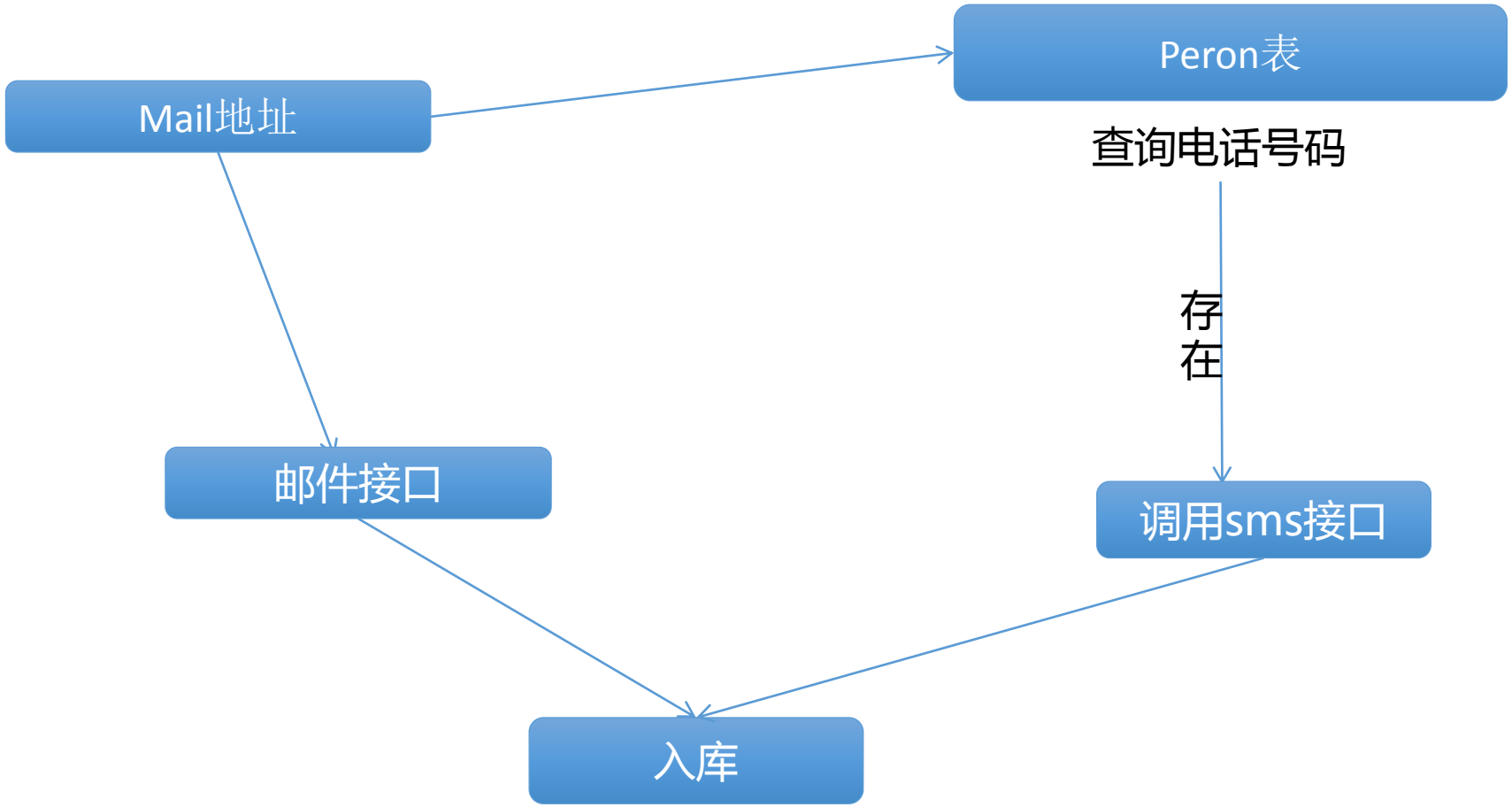
Person表

name	mail	sms
beibei	beibei@qq.com	138xxxxxx

Log表

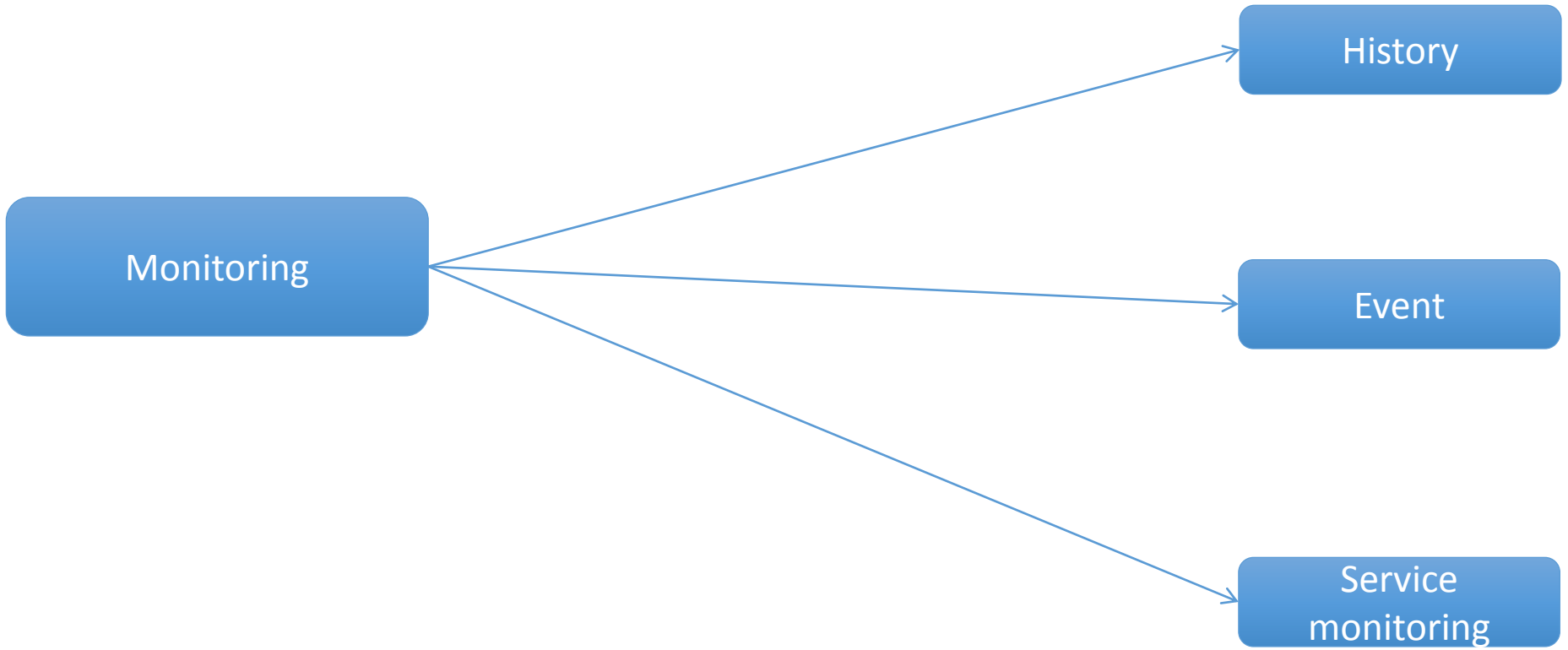
list	content	time
beibei#@qq.com	Xxx is down	2014-08-09 12:11:01

六 报警邮件和短信的分离

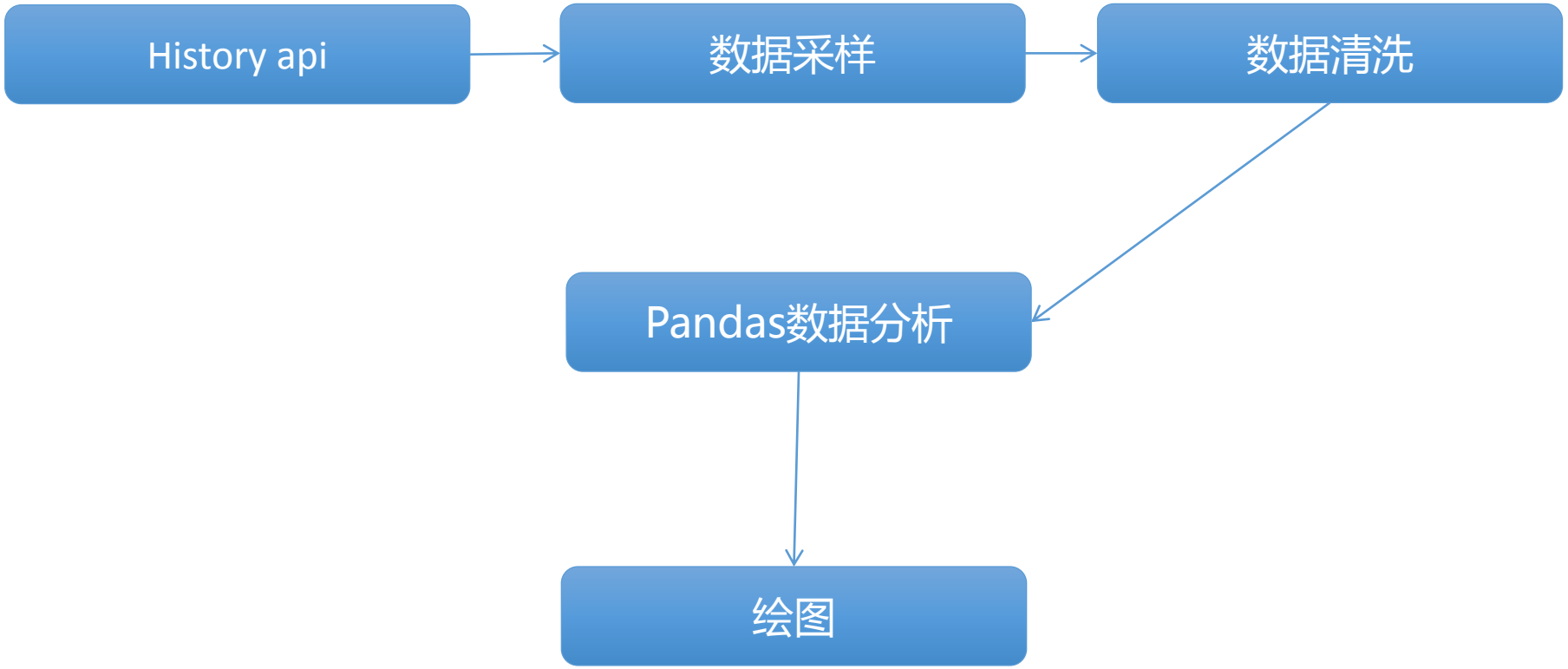


注释：其它解决方案 如openduty

七 zabbix api（概览）



七 zabbix api（采样分析）



七 zabbix api（历史趋势分析）

Zabbix 杂谈

谢谢大家