SACC 2016 第八届中国系统架构师大会
SYSTEM ARCHITECT CONFERENCE CHINA 2016
架构创新之路

光载无限监控体系的变革与演进

欧曜玮
ou.yaowei@wifire.net

在开始之前

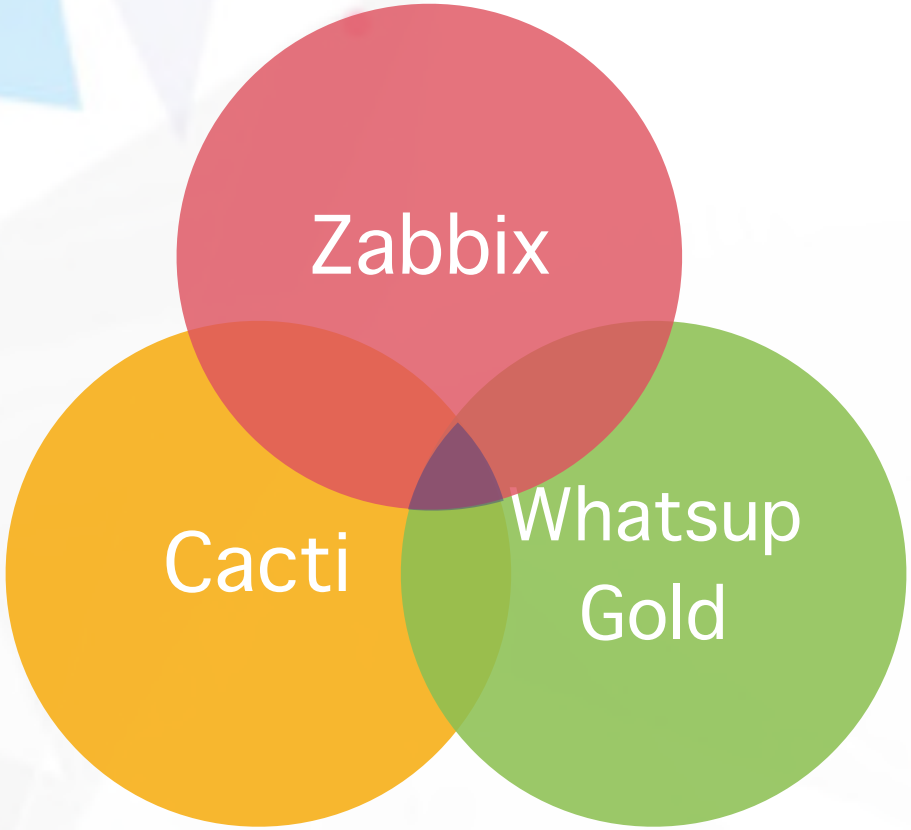
- 设备监控、资源监控、网络监控、服务监控、网站监控、业务监控、性能监控、应用性能管理（APM）、应用程序监控、安全监控

目录

- 过去的监控系统
- 我们为什么选择了 Open-Falcon
- 光载无限如何玩转 Open-Falcon
- 总结

过去的监控系统

过去的监控系统



- 难以扩展
- 二次开发难度高
- 多套系统，不同的操作方式
- 需要管理不同的配置，系统间同步
- 难以保证监控全复盖

需求总结

- 统一监控工具，降低维护以及排查的成本
  - 降低值班人员的学习成本与技术门槛，将解决问题的经验积累到系统上
- 提高监控系统的效能、可用性、用户体验
- 高可扩展性
  - 目标至少支持一万台服务器的规模
- 能满足内部的二次定制需求

寻找开源解决方案

有这么多的开源监控系统...

Zabbix, Nagios, Cacti, WhatsupGOLD, Sensu, Open-Falcon, Prometheus, Flapjack, Bonsun, OMD, Check_MK, graphite, collectd, Zenoss, monit, Munin, OpenNMS, Ganglia, Consul...

也有这么多的商业监控系统...

New Relic, Logical Monitor, DataDog, Splunk,
监控宝, Elastic-Marvel, Librato, Stackdriver...

我们如何选择开源项目？

- 现代化的设计与良好的架构
- 活跃的社区与大公司的支持
- 模块化、可扩展性高，易于二次开发
  - 可以很容易地利用既有的 Zabbix Script 或 Plugin
- 满足光载无限的业务需求
  - 网路质量、7x24 小时值班人员、等等

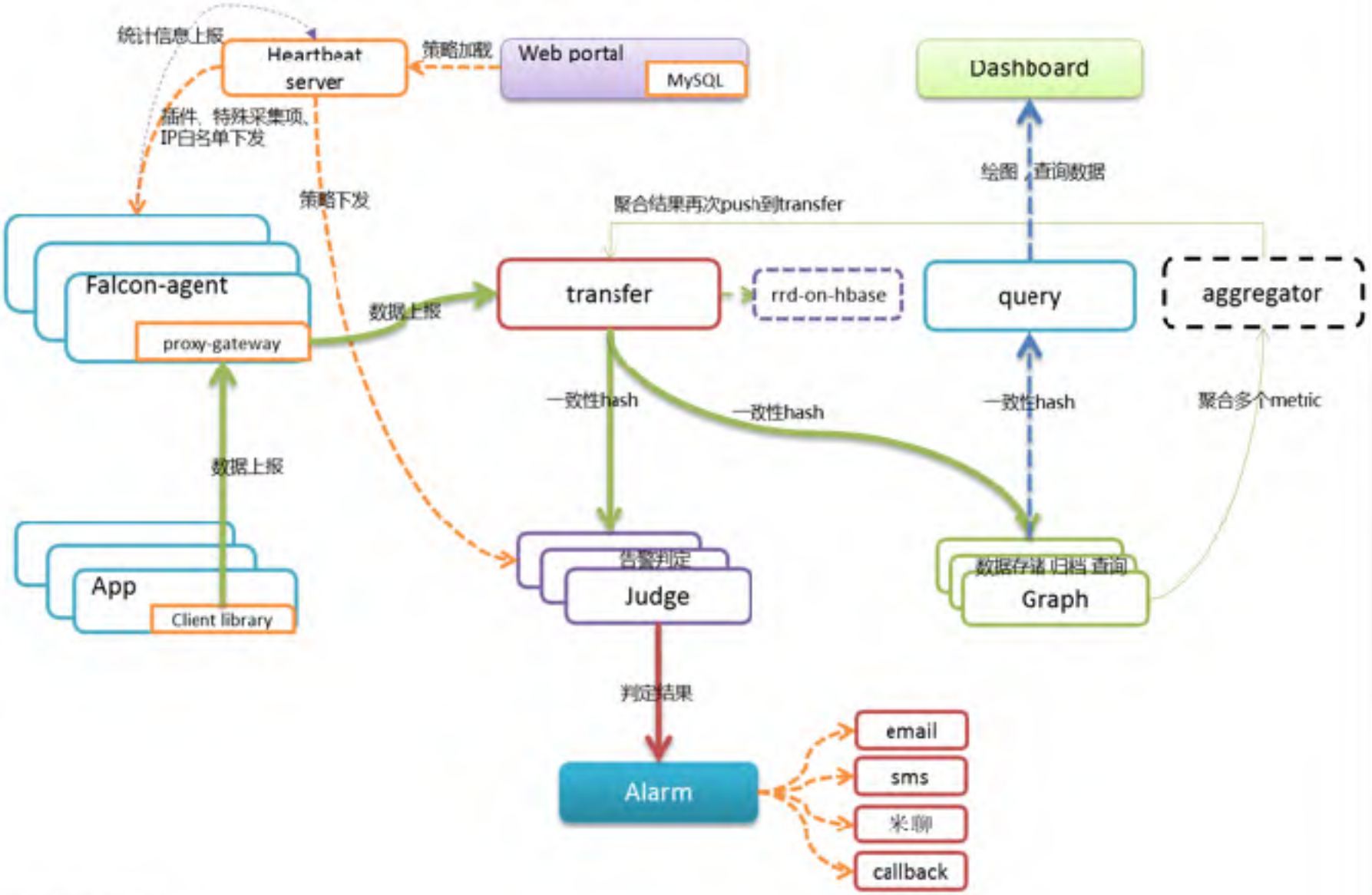
依照这标准，剩下...

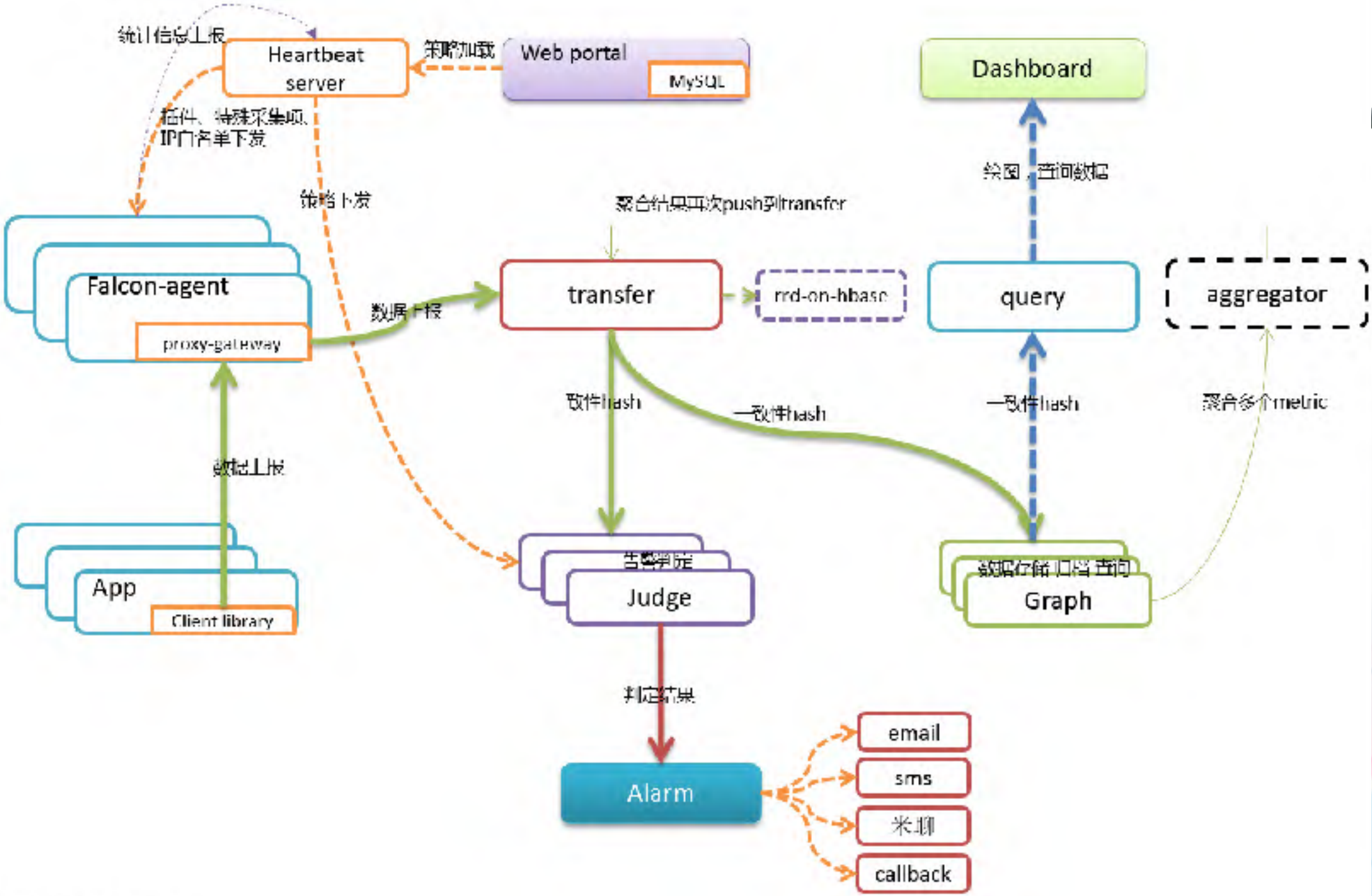
- Sensu
  - 2011
  - Heavy-Water
- Prometheus
  - 2012
  - SoundCloud
- Consul
  - 2013
  - HasiCorp
- Open-Falcon
  - 2015
  - Xiaomi

我们为什么选择 Open-Falcon?

- 系统设计简洁，易于二次开发
- 分布式组件，水平扩展容易，支持量级高
- 高效数据展示与查询
- Golang 二进制部署方便
- 完善的基础采集项
- 人性化的告警策略与模板设计

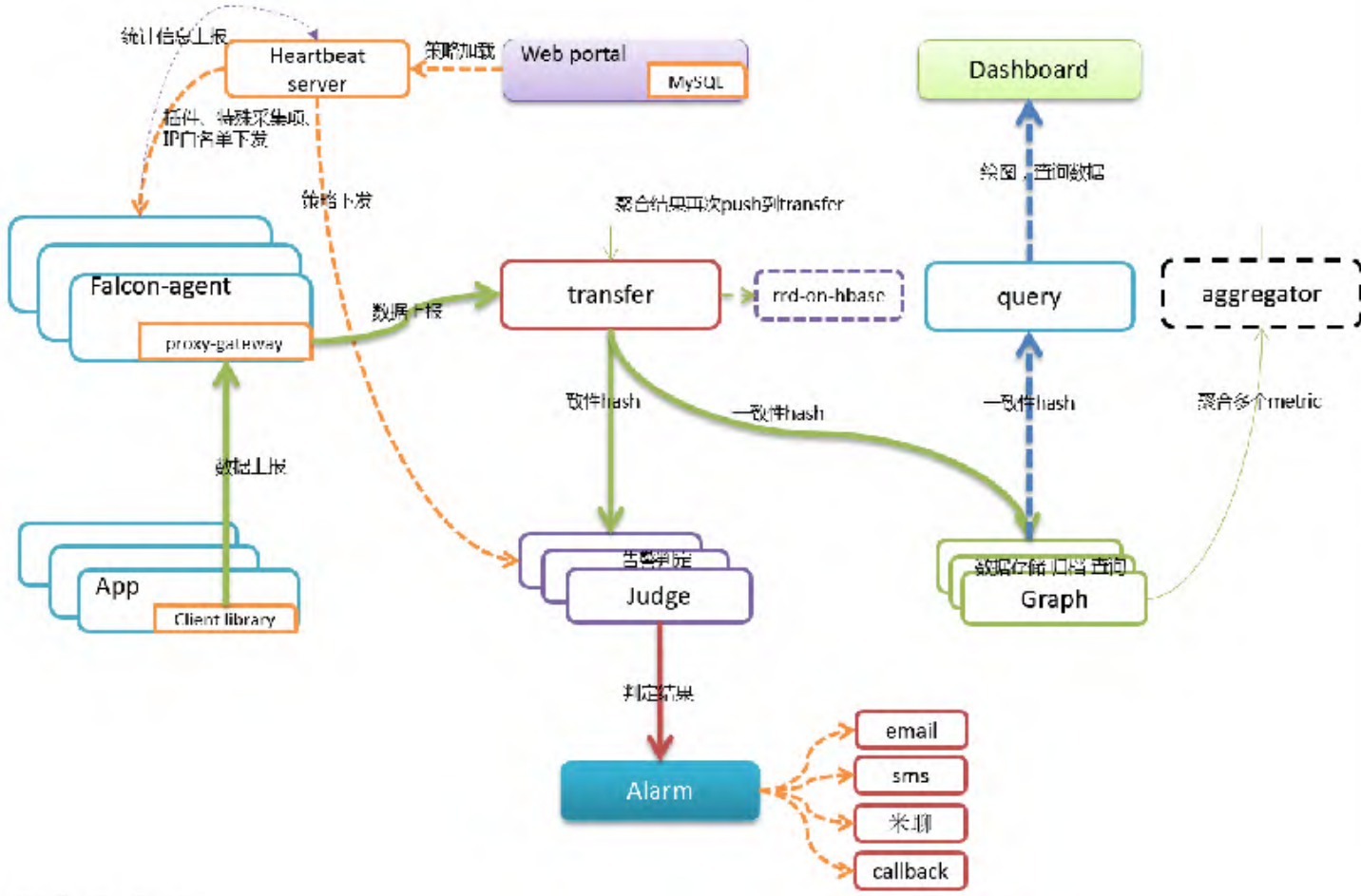
Open-Falcon 的问题




Heartbeat server
Web portal
MySQL
Dashboard
Falcon-agent
proxy-gateway
transfer
rrd-on-hbase
query
aggregator
App
Client library
Judge
Graph
Alarm
email
sms
callback

Open-Falcon 的问题

- 系统不易分发
- 安全性考量不到位
- 没有通盘考虑的权限设计
- 复杂度稍高
- 每个 Graph 实例均是单点
- Graph 扩容有损
- 上下游组件没有 naming
- 报警没有入库
- 报警现场没有保存

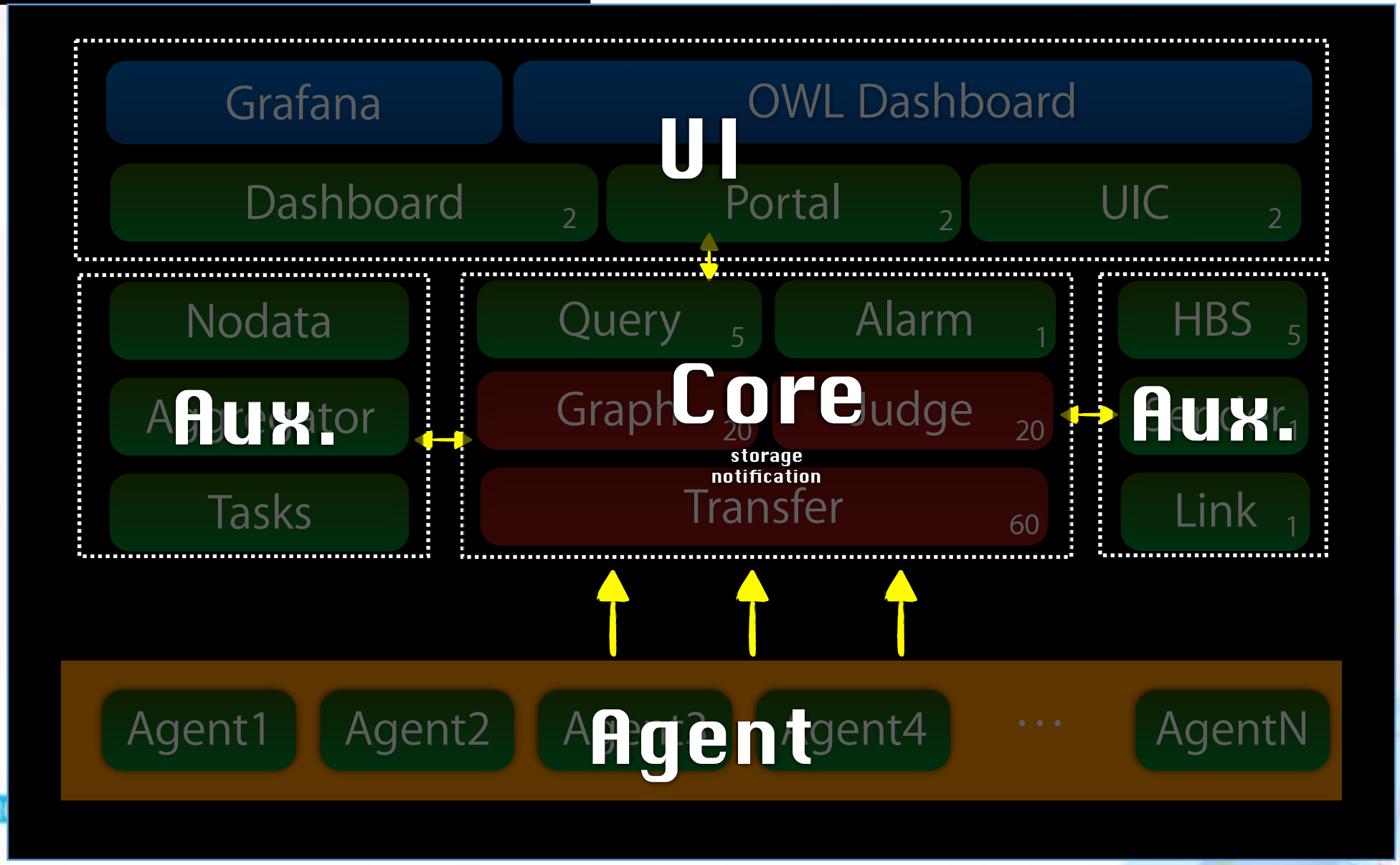


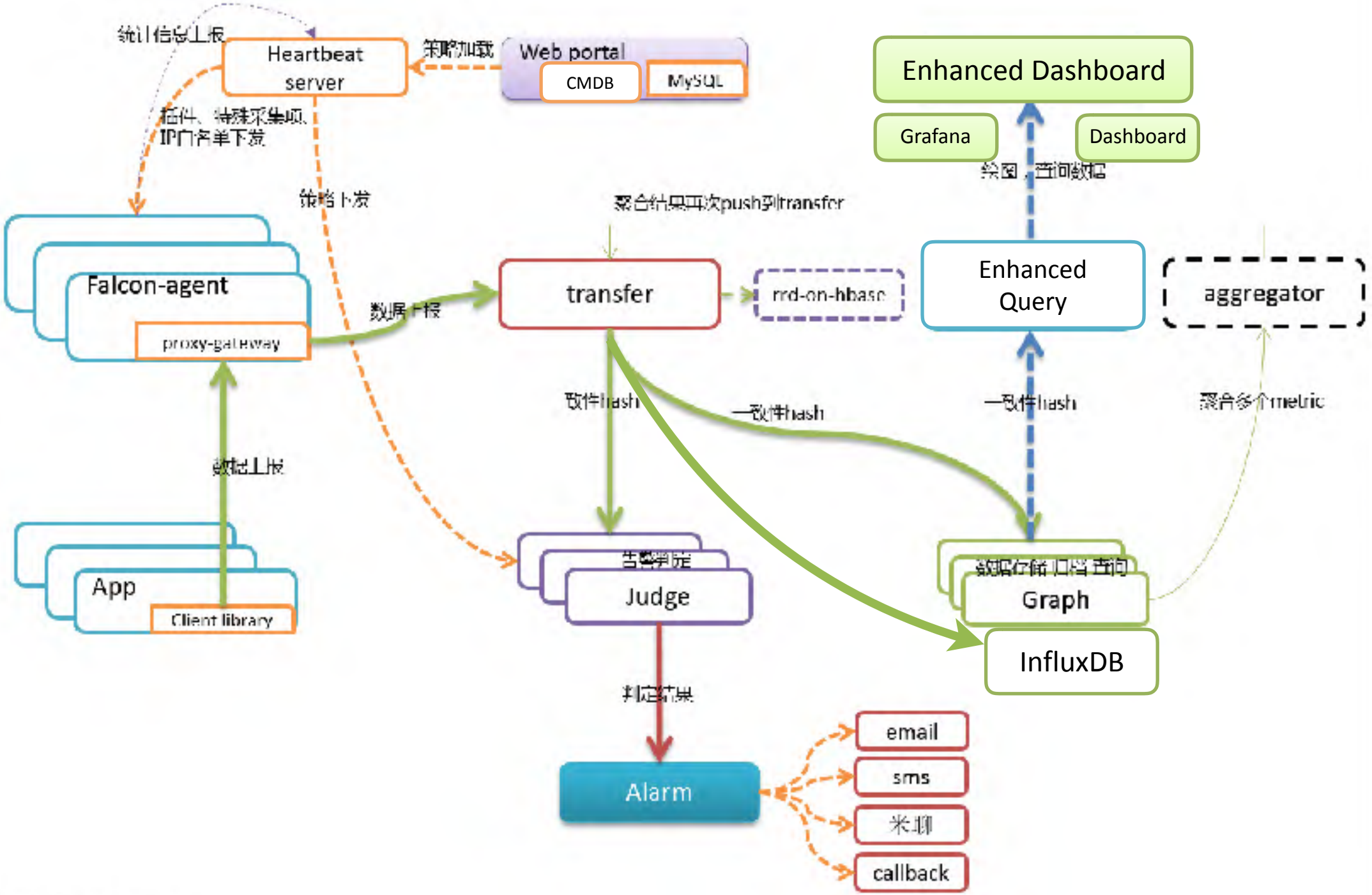
二次开发 Open-Falcon

玩转 Open-Falcon

- Container/Monitoring as a Service
- 安全性补强
- 用户界面改善与体验优化
- CMDB 与 Open-Falcon 结合
- 运维内部接口适配
- 后端模块统一与简化
- CDN 网络质量监测

我们是这么看待 Open-Falcon 的




Heartbeat server
Web portal
CMDB
MySQL
Enhanced Dashboard
Grafana
Dashboard
Falcon-agent
proxy-gateway
transfer
rrd-on-hbase
Enhanced Query
aggregator
App
Client library
Judge
Graph
InfluxDB
Alarm
email
sms
callback

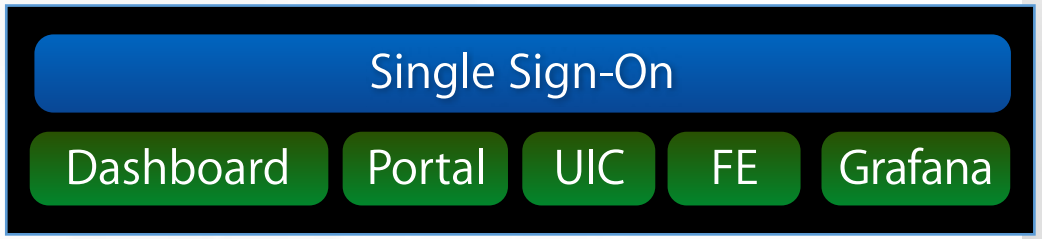
Container as a Service

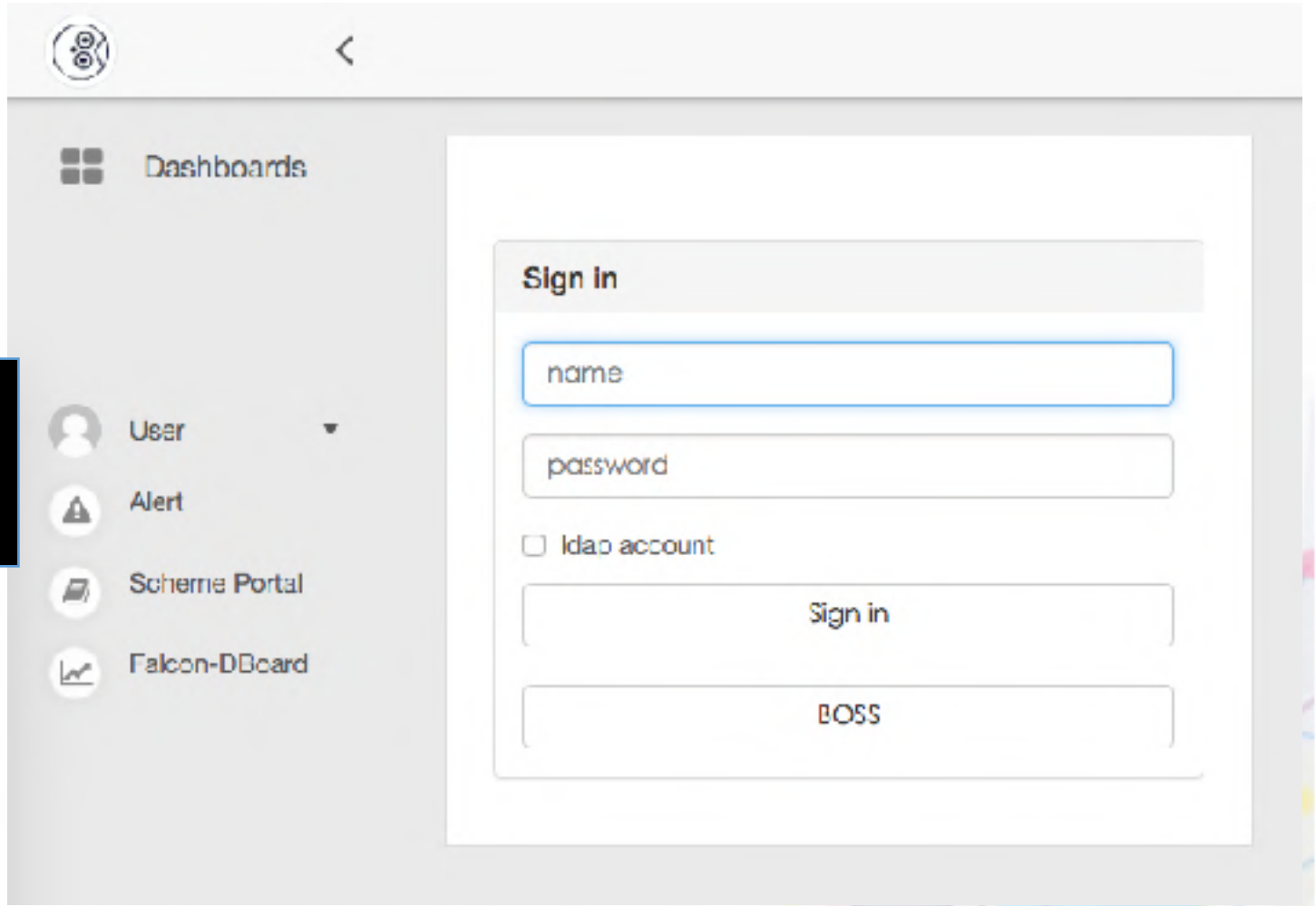
- Open-Falcon 所有组件容器化
- 搭配 Docker-Swarm 做资源调度

```
[root@        DockerCompose-Ope
  Name      Command     State
--------------------------------------------------------------
graph      ./run.sh    Up      0.0.0.0:6070->6070/tcp, 0.0.0.0:6071->6071/tcp
hbs        ./run.sh    Up      0.0.0.0:6030->6030/tcp, 0.0.0.0:6031->6031/tcp
judge      ./run.sh    Up      0.0.0.0:6080->6080/tcp, 0.0.0.0:6081->6081/tcp
query      ./run.sh    Up      0.0.0.0:9966->9966/tcp
transfer   ./run.sh    Up      0.0.0.0:6060->6060/tcp, 0.0.0.0:8433->8433/tcp
```

安全性补强

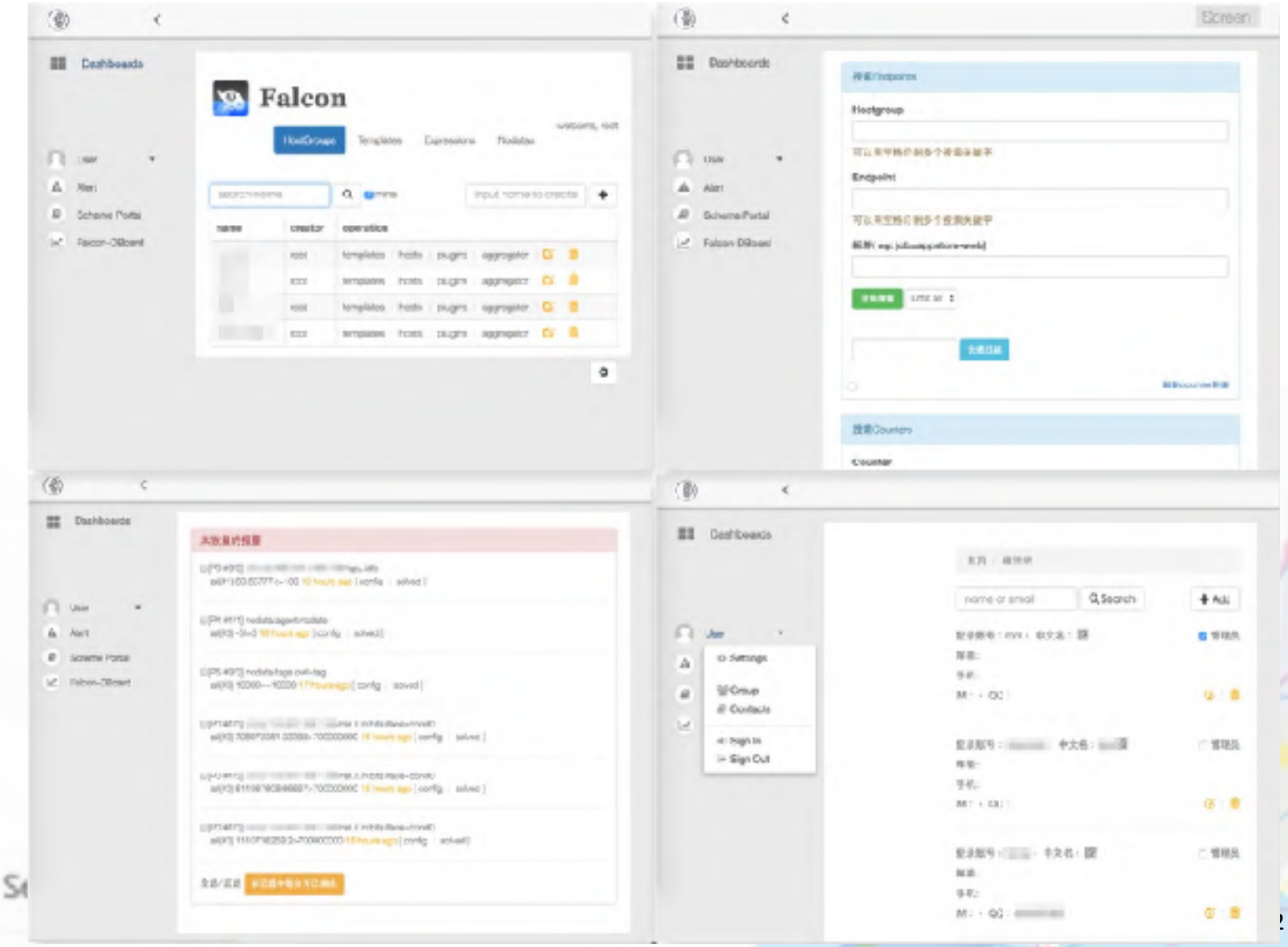
- SSO 单一登入
- Grafana 与 Open-Falcon 的账号整合





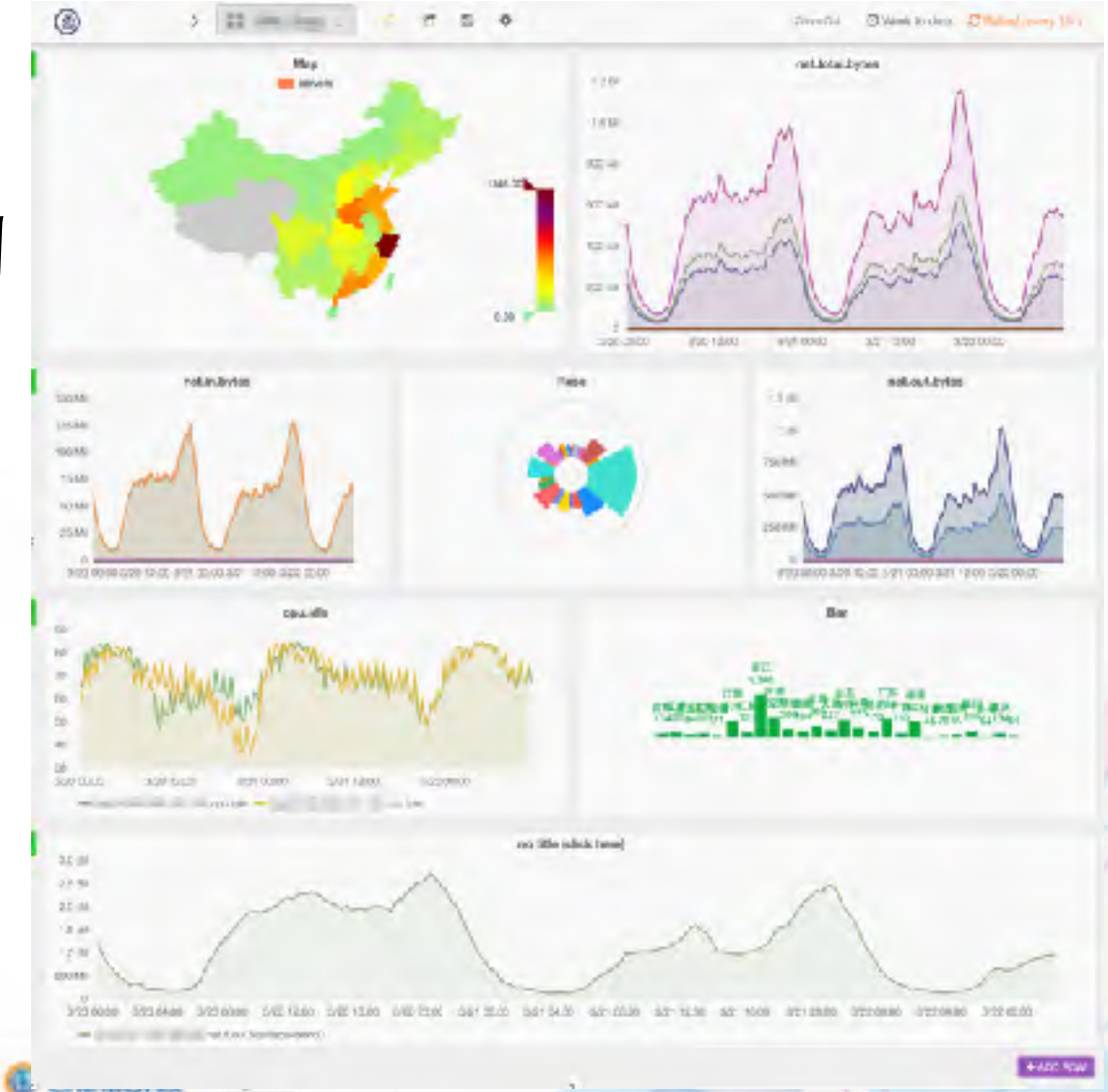
用户界面改善与体验优化（1/4）

- 统一入口不再迷失
- 加入侧边栏、导览列



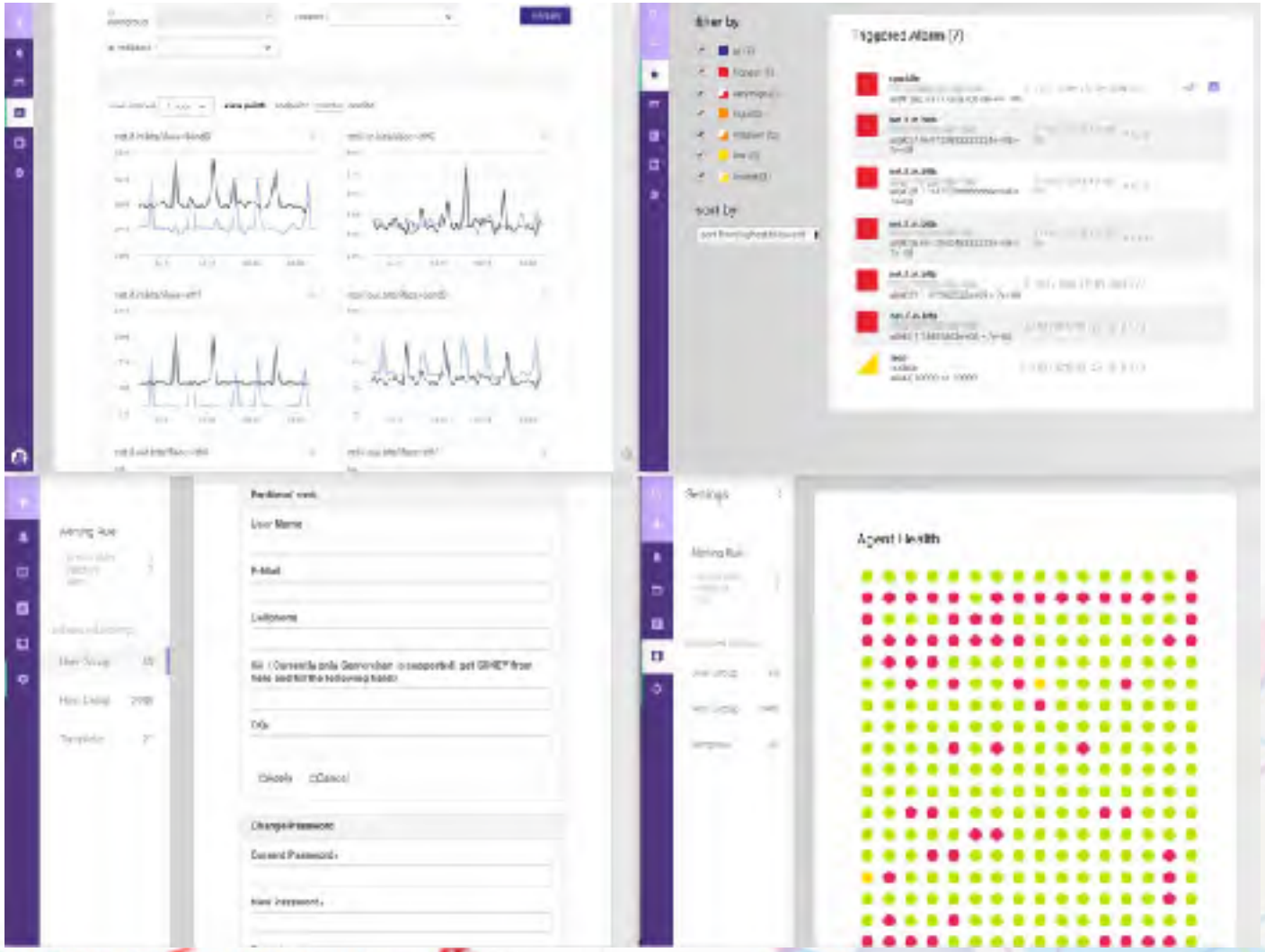
用户界面改善与体验优化（1/4）

- 探索趋势图以外的可能
- Grafana 与 Open-Falcon 的深度整合

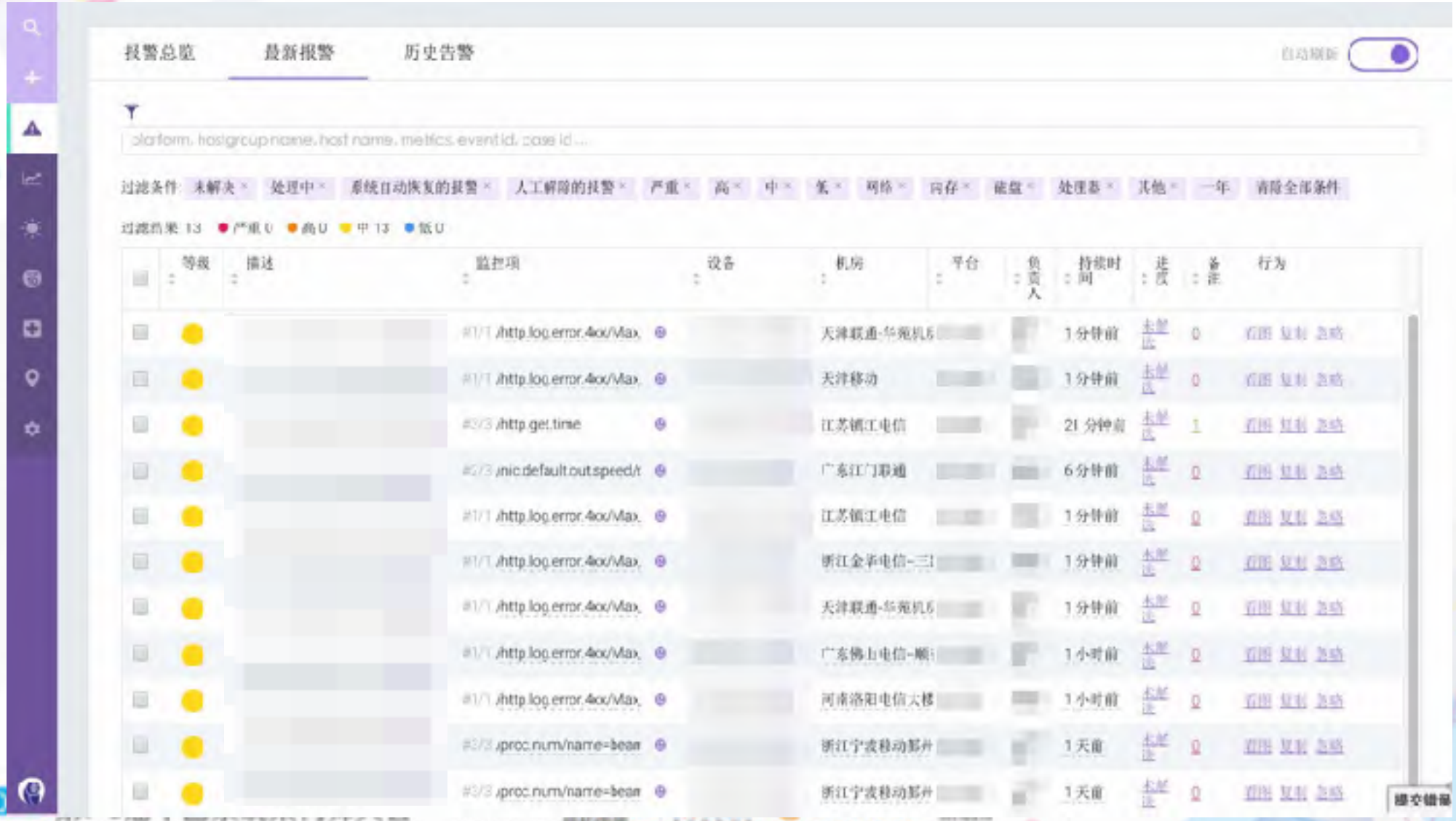


用户界面改善与体验优化（3/4）

- Isomorphic Web Design
- 单页式设计
- 服务端渲染
- 前端组件统一



用户界面改善与体验优化（4/4）



CMDB 整合（1/2）

- 支持以内部 BOSS 账号登入
- 设备列表跳转至概览页面
- 资产信息同步

CMDB 整合（2/2）

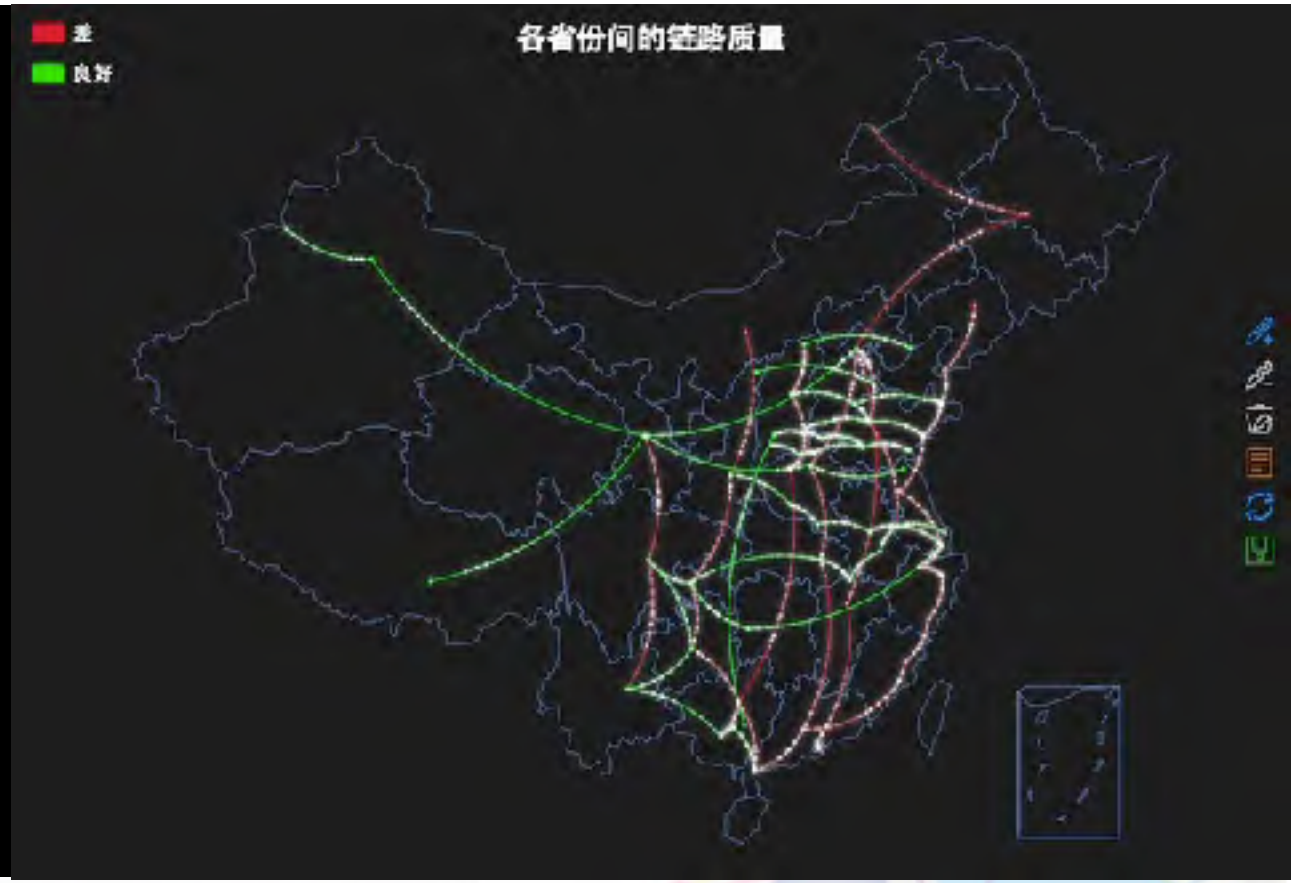
- CMDB 统一管理服务与角色，专用接口替换模板变量

运维内部接口适配

- 协助运维更快上手
  - Screen, Template, HostGroup 的初始范本
  - Zabbix 脚本移植、提供插件范本
- Agent 安装流程与配置简化
  - 如：RPM 包安装、/etc/init.d 启动脚本
- 支持 Zabbix API 接口
- 支持以 HostGroup 为单位查找，设备同步自 CMDB

网路质量监测

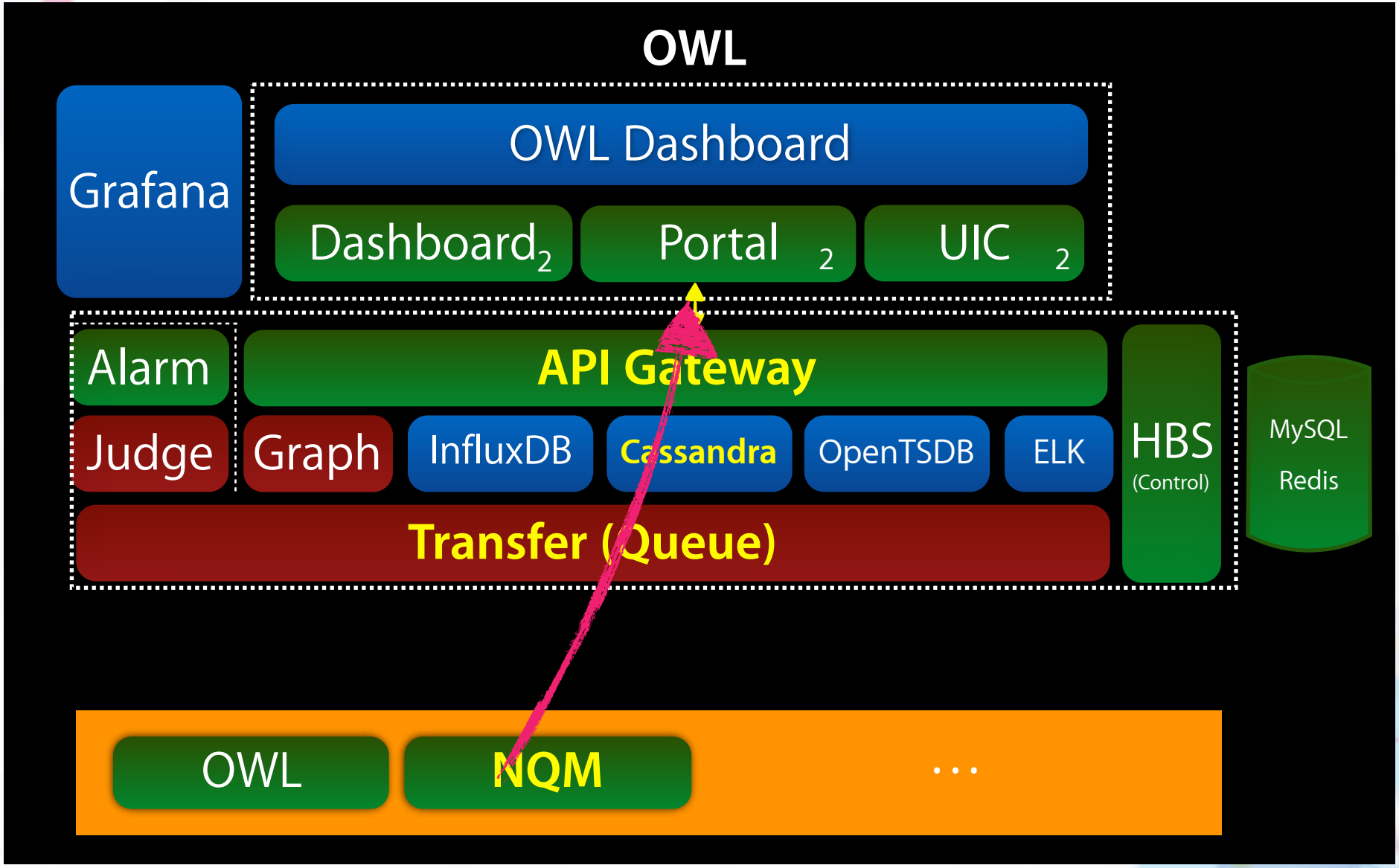
- 点对点的网络指标数据采集



网路质量监测的挑战

- 多维度
  - 机房、省份、城市、地区、运营商
- 批次计算
  - 平均值、最大/最小值、标准差
- 不同的分组：
  - 内网、外网、平台
- 向量
  - 方向与大小

架构上的调整


OWL
Grafana
OWL Dashboard
Dashboard2
Portal 2
UIC 2
Alarm
API Gateway
Judge
Graph
InfluxDB
Cassandra
OpenTSDB
ELK
HBS
(Control)
MySQL
Redis
Transfer (Queue)
OWL
NQM
...

网路质量监测（1/3）

网路质量监测（2/3）

网路品质监控　Nodes

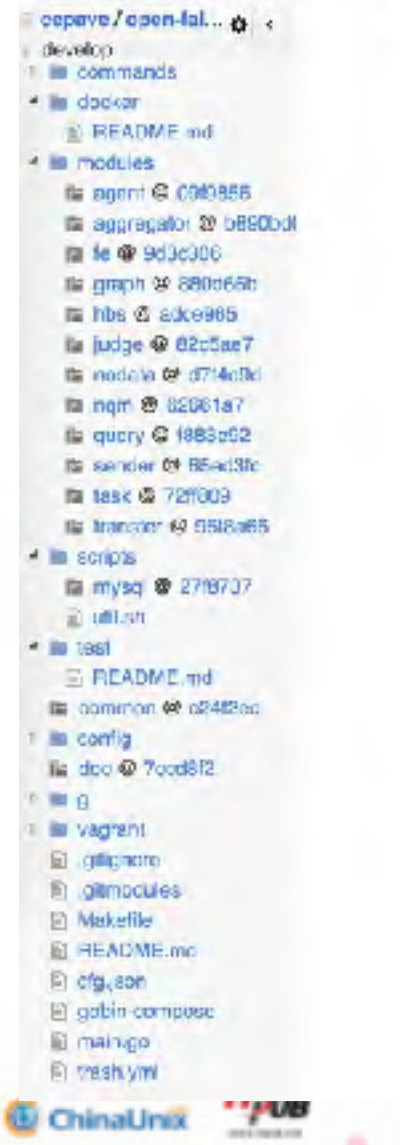
starttime=yyyy-mm-dd endtime=yyyy-mm-dd agent.province=省份1,省份2... agent.isp=isp1,isp2... target.isp=isp1,isp2...　搜寻

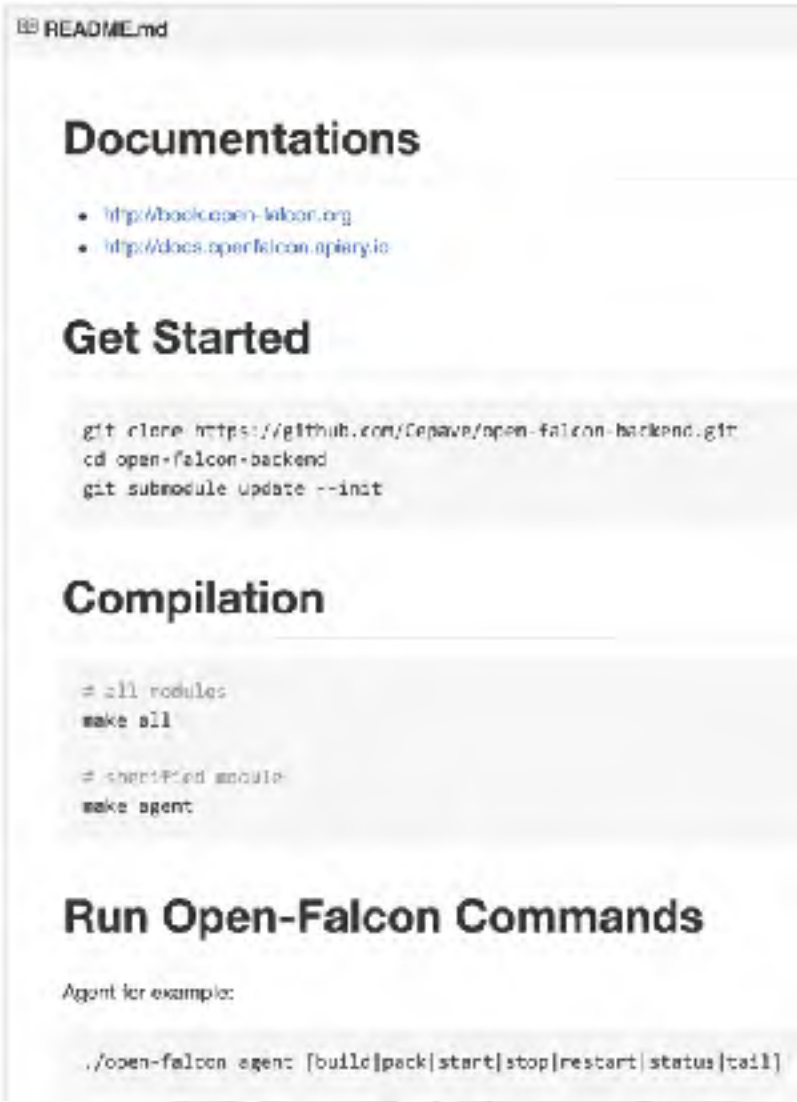
全中国

地区	被探测點	最大值	最小值	平均	丟包率	稳定度	中位數	标準差
河南	3	239	37	47.75	0.08%		48	0.44
辽宁	4	693	32	41.03	0.09%		36	0.84
广西	12	383	16	34.55	2.27%		27	2.25
四川	13	478	36	50.87	0.39%		48	2.58
江苏	33	577	5	44.2	5.43%		44	3.21

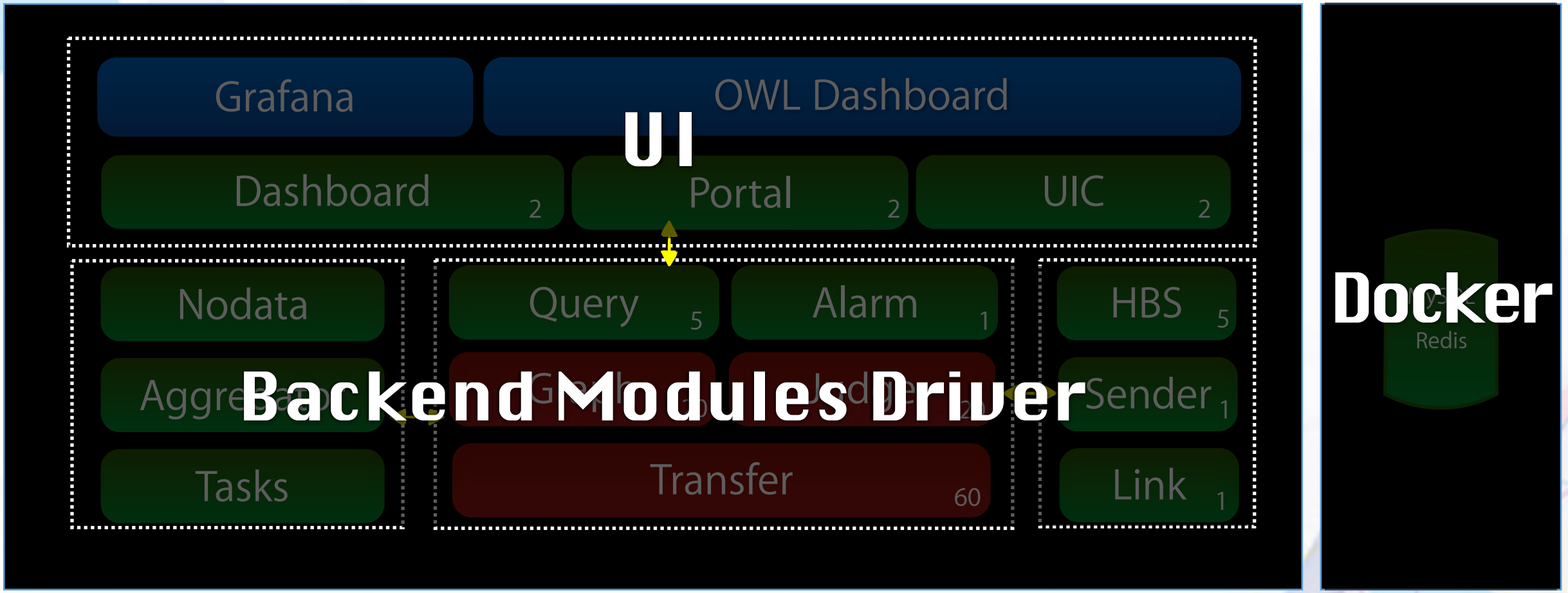
后端模块统一与简化（1/2）

- 借鉴 Consol 与 Docker-Compose
- 简化服务端维护与部署的负担
  - 配置档统一管理
  - 打造周边工具链
    - docker, vagrant, test, scripts, make
- 集中管理，聚人气
  - 代码、文档、问题、反馈、PR
    - ★399 agent (~2016/04/09)
    - ★262 of-release (~2016/04/09)





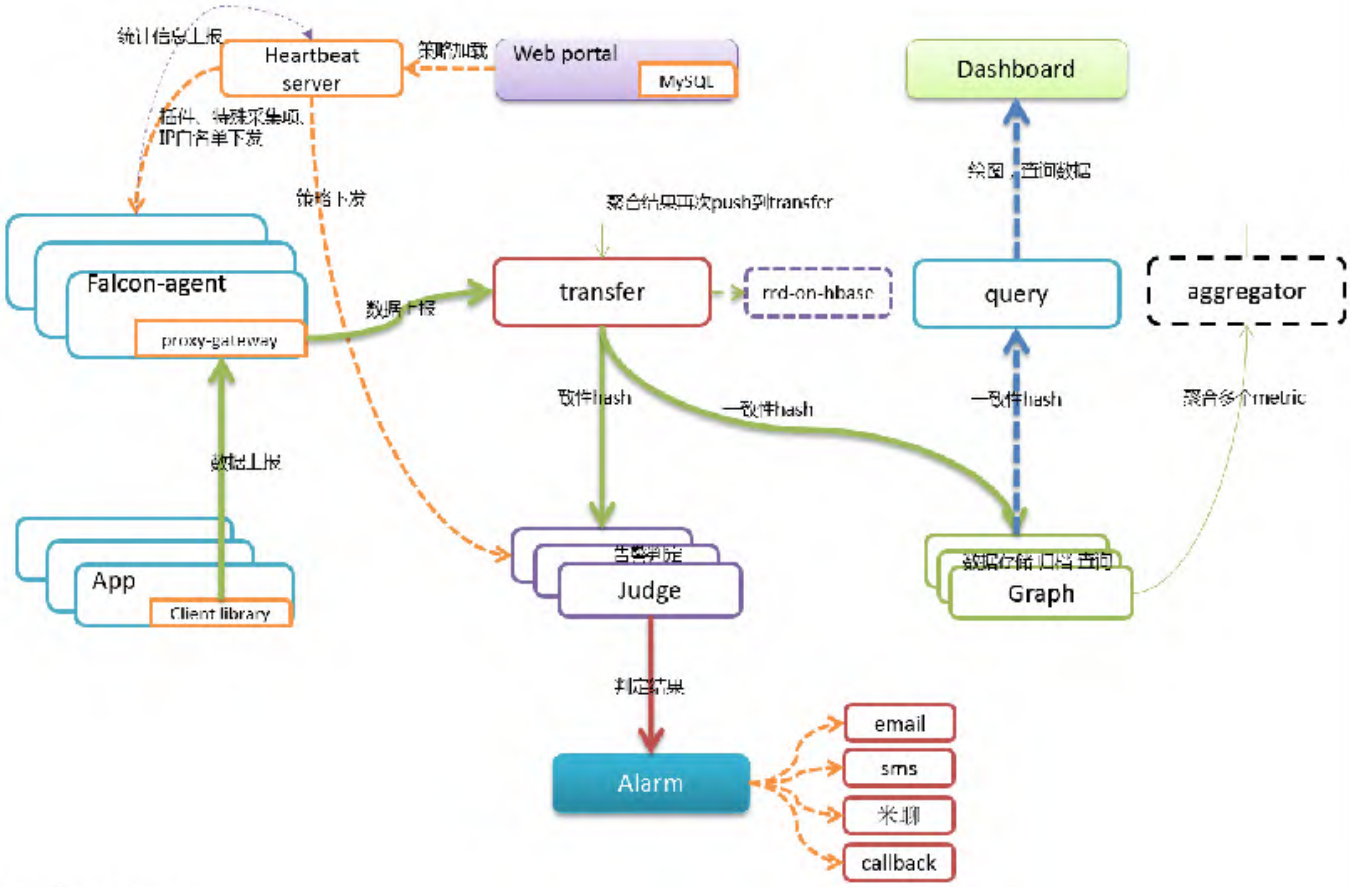
后端模块统一与简化（2/2）

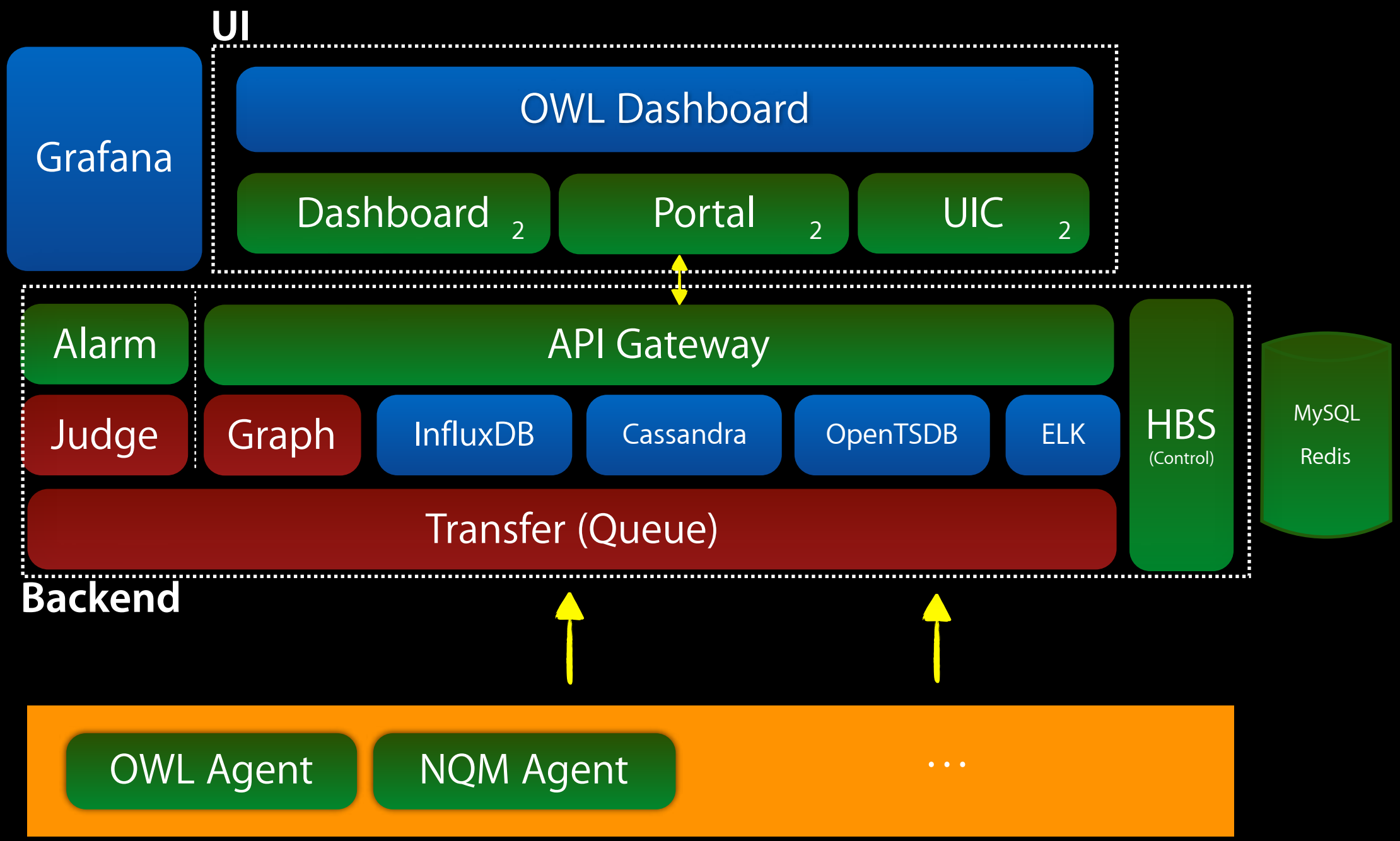


总结

回顾 Open-Falcon 的问题

- 系统不易分发
- 安全性考量不到位
- 没有通盘考虑的权限设计
- 复杂度稍高
- 每个 Graph 实例均是单点
- Graph 扩容有损
- 上下游组件没有 naming
- 报警没有入库
- 报警现场没有保存

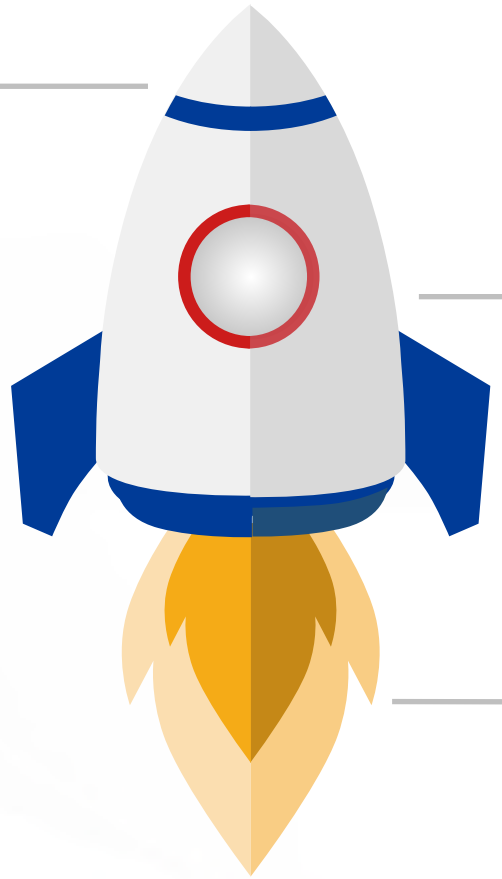



UI
Grafana
OWL Dashboard
Dashboard 2
Portal 2
UIC 2
Alarm
API Gateway
Judge
Graph
InfluxDB
Cassandra
OpenTSDB
ELK
HBS
(Control)
MySQL
Redis
Transfer (Queue)
Backend
OWL Agent
NQM Agent
. . .

社区发展

贡献文档

贡献代码



问题反馈

参与其中

吐槽也是一种参与

Q & A

THANK

SequeMedia 盛拓传媒 | IT168 | ChinaUnix | ITPUB