



GOPS 2019
Shanghai

GOPS

全球运维大会
2019
- AIOps 风向标



指导单位：



主办单位：



大会时间：2019年11月1日-2日

大会地点：上海光大会展中心国际大酒店



运营商网络运维工程师的转型之路





关于我



姓 名：李佐辉

毕业院校：浙江大学 信电系

工作单位：浙江移动 网络部

TITLE： SRE团队技术负责人 首席架构师
通信网运维专家 高级工程师



11年通信网络运维经验，深入理解通信网络运维的IT需求，浙江移动SRE团队发起人，通信工程师IT开发转型推动者。



目录



1

我们是网络运维工程师

2

我们遇到了点小麻烦

3

我们的SRE转型之路



维护的设备：厂家定制，专有软硬件的通信设备。



程控交换机



核心路由器



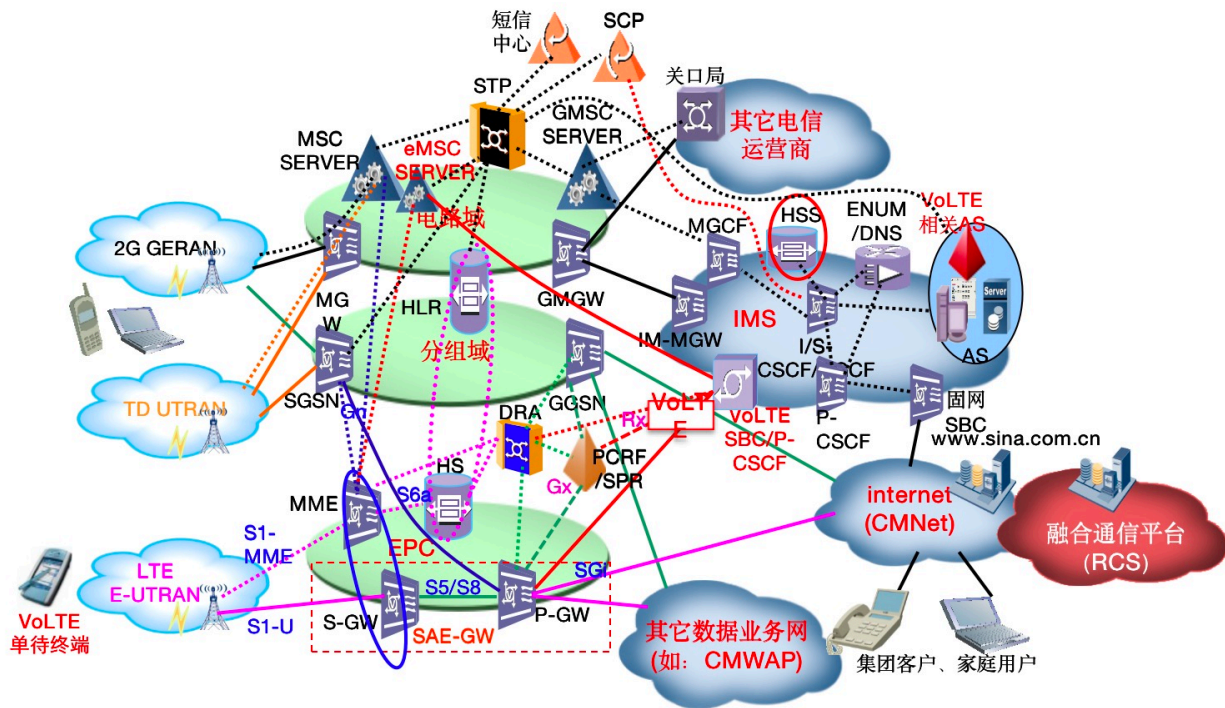
高端防火墙



3GPP标准核心网元



我们眼中的世界：路由、协议、网元、信令、局数据...





我们的工作内容



日常运维

- 保证网络稳定运行

网络优化

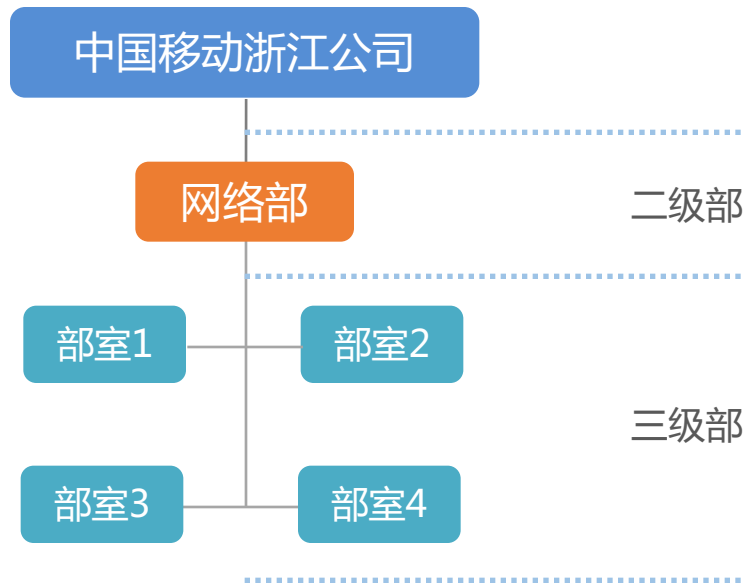
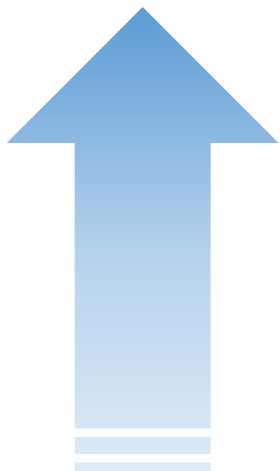
- 提升网络性能，提高用户满意度

应急响应

- 对故障和投诉的紧急处理



典型的垂直式行政组织结构





考核机制：KPI为导向的负向激励





人员技能：IT开发和设备维护技能弱，但是对IT开发转型充满热情。

总体技能调查

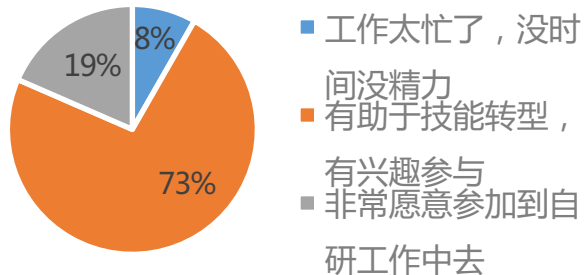
几乎所有人员都有编程经验，但是其他技能较弱



兴趣调查

对IT开发有较大热情占18%，有兴趣的占73%，共144人愿意参与运维开发工作。

运维开发转型兴趣调查



目录

1 我们是网络运维工程师

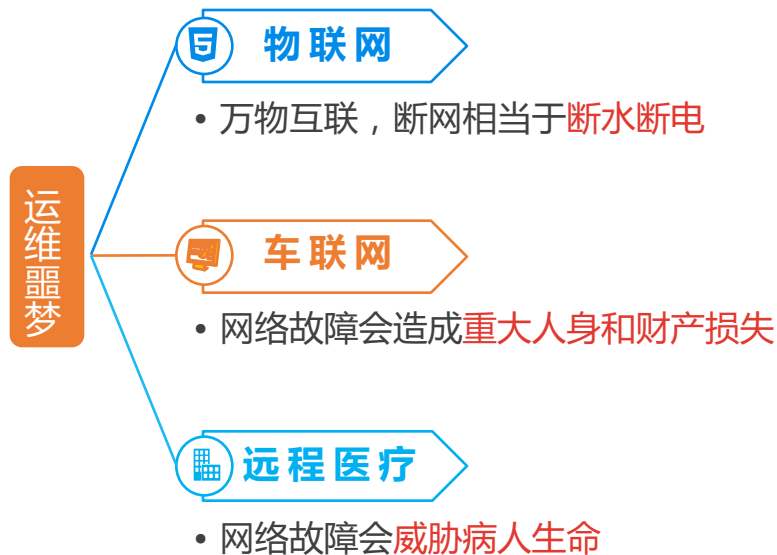


2 我们遇到了点小麻烦

3 我们的SRE转型之路



通信技术发展越来越快，网络运维的压力和责任越来越大





运维手段原始，大量工作处于人肉运维阶段；对标先进互联网公司，我们还有很长的路要走。

互联网公司

- 大量运维工作**以工具为主**，释放人力

标准化

各类数据、组件、流程等标准化，便于快速更新迭代。

自动化

使工具贴合运维提升生产效率，同时降低人工操作失误。

智能化

数据与AI技术的洞察能力

- DevOps**：研发运维模式 **SRE**：运维实践 **Cloud Native**：技术架构

我们

- 以人工为主，工具为辅。**
- 标准化、自动化、智能化**：正在不断建设和探索。
- 厂家定制开发**：烟囱式架构，开发由第三方厂家完成，开发响应慢
- 需求和开发跨部门协作**：存在部室“膜”



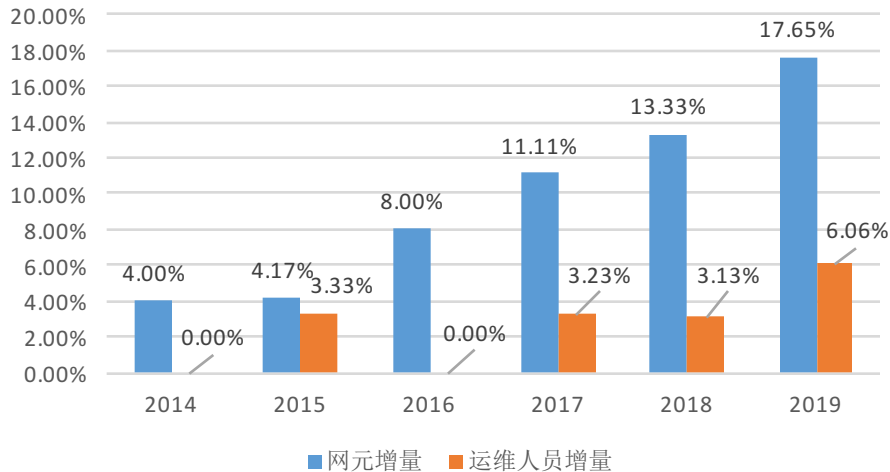


网络运维人的窘境

设备越来越多，但是人不见多。

2014到2019年，语音核心网元增加了80%，但是运维人员仅增加12%。

网元增量与运维人员增量对比图



事儿越来越多，头发越来越少



虽然支撑系统平台手段众多，但日常维护中仍存在大量的手工冗余操作

01

厂家平台针对新需求和自身缺陷的**开发周期、落地时间长**



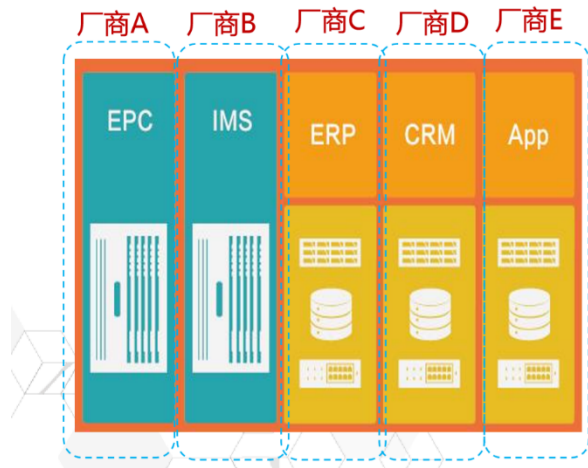
02

现有平台封闭，不具备开放性

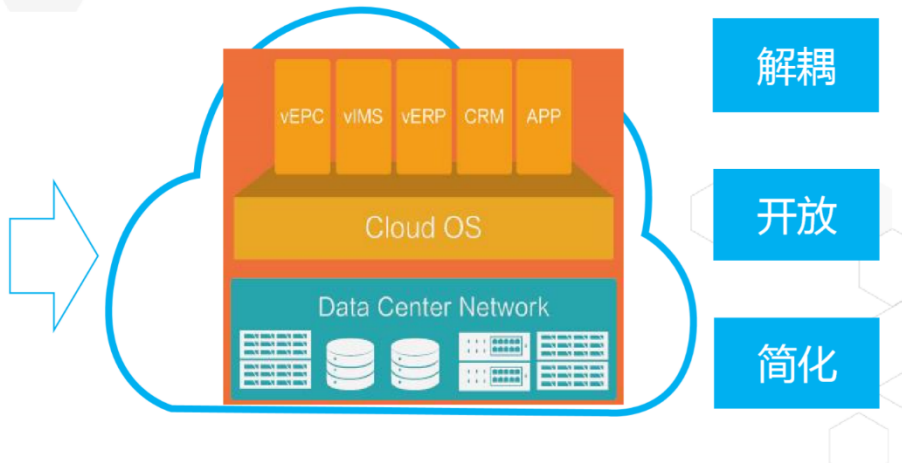


网络设备正在向IT云化演进，网络运维人亟需转变思维。

From: 传统烟囱式结构：软件和硬件合一



To: 软件和硬件解耦





复杂的云化架构有时发生故障根本无从查起，诡异的故障无法用科学解释。



无法用科学解释的时候，迷信就出现了

网络运维人压力山大

- 技术更新快
- 故障影响大
- 网络设备多
- 相对人力少
- 自动化水平低
- 云化结构复杂
- 故障现象诡异

目录

1 我们是网络运维工程师

2 我们遇到了点小麻烦

➔ 3 我们的SRE转型之路

① 转变思想

② 明确目标

③ 打造SRE生态圈



SRE是Google对DevOps的实践总结

- SRE的终极目标：系统稳定

- SRE团队的组成：

- 1) 50%~60%软件工程师，
- 2) 其他具备85%~99%软件技能，且具备一定程度其他IT技能的工程师。

- SRE模型的优势：

- 1) 运维人数相对少；
- 2) 开发团队和运维团队的冲突焦点消除；
- 3) SRE团队和研发团队之间的成员可以自由流动。

- SRE模型的问题：符合SRE的人才难招聘





SRE运维的特点

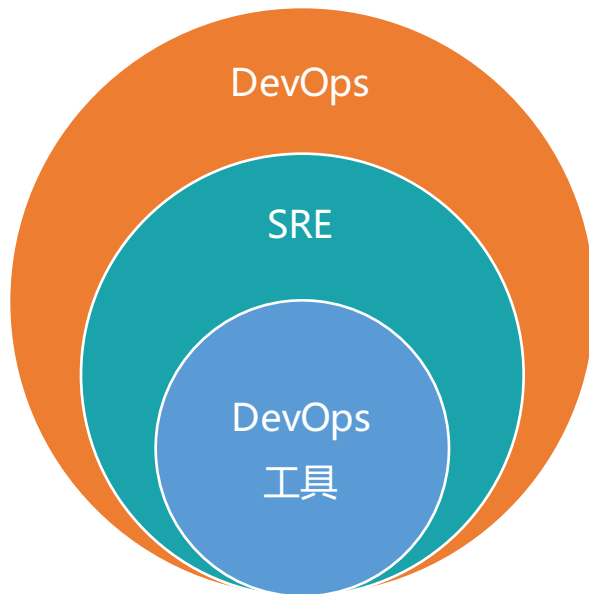
- 通过**主动运维**思维将运维经验、最佳实践、方式、流程、方法都固化成代码，**减少琐事**。
- 通过**自动化运维**来应对规模性扩张和复杂度上升的运维环境。
- 结合专业化能力，通过写工具，参与系统设计，参与可靠性的提高，来**打通与支撑团队的反馈回路**。

SRE人员的特点

- 从**内心鄙视**重复性、体力性的劳动
- 愿意做**主动性**的工作减少重复性劳动
- 有**开发能力**



SRE理念是DevOps的具体表现形式。DevOps和SRE是道，DevOps工具是术





确立目标

SRE团队的目标：

- 提倡主动运维思维，以运维开发为手段，依托SRE运维模式，提升运维人员ICT技能, 支撑各专业高效运维。

网络运维人也有自己的终极目标

钱多

- 建立合适的组织架构，打开晋升通道，减少行政束缚；
- 改善人员培养体系，提升个人竞争力；
- 改进激励机制，获得资源倾斜，打开收入上升通道。

活少

- 建立日常工作框架，支撑日常开发工作，将琐事自动化，减少工作量；
- 通过SRE的开发支撑体系将降低开发门槛，提升开发效率。





建立良好的生态圈

通过5种手段建立SRE工作机制，形成良好的SRE生态圈。

激励制度

- ✓ 建立OKR导向的正向激励绩效倾斜制度

人员培养

- ✓ 课程培训
- ✓ 技能评定
- ✓ 经验分享
- ✓ 业界大拿讲课

开发支撑环境

- ✓ 开发和部署环境支撑
- ✓ DevOps工具支撑
- ✓ 现有平台解耦
- ✓ 数据源开放

日常工作框架

- ✓ 建立开发机制
- ✓ 关注安全模式
- ✓ IT需求归口
- ✓ 建立技术规范

组织框架

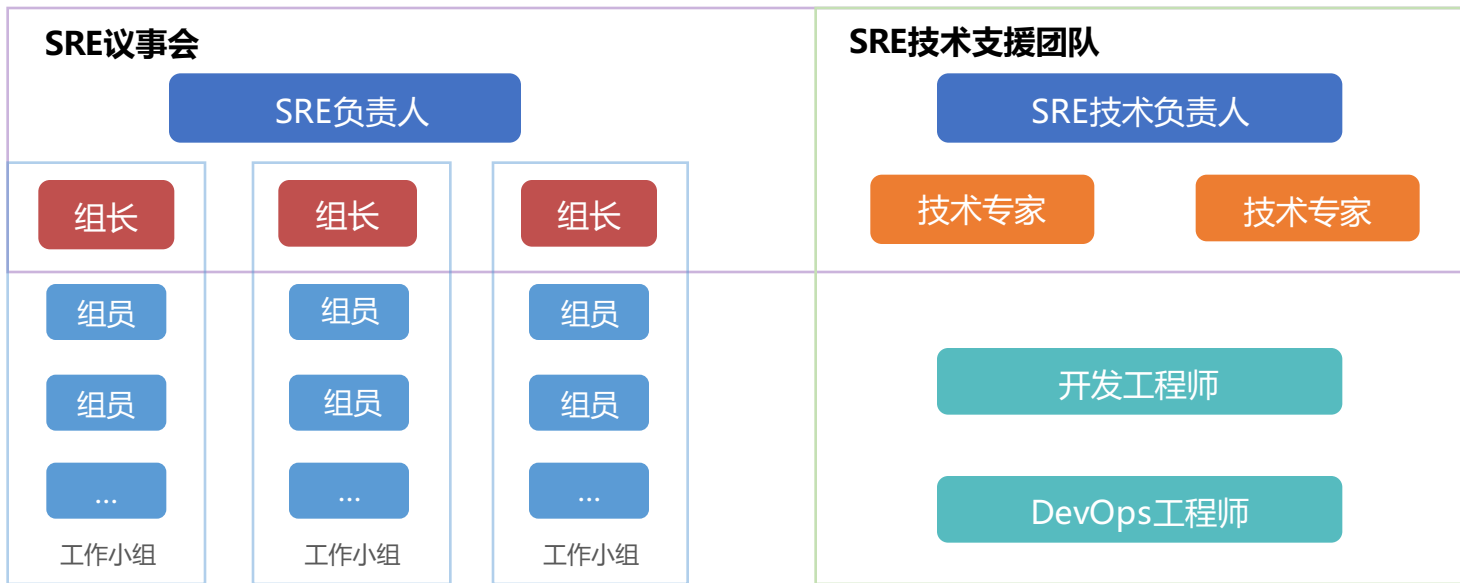
- ✓ 建立议事会负责制的半刚性组织架构制度





建立组织架构

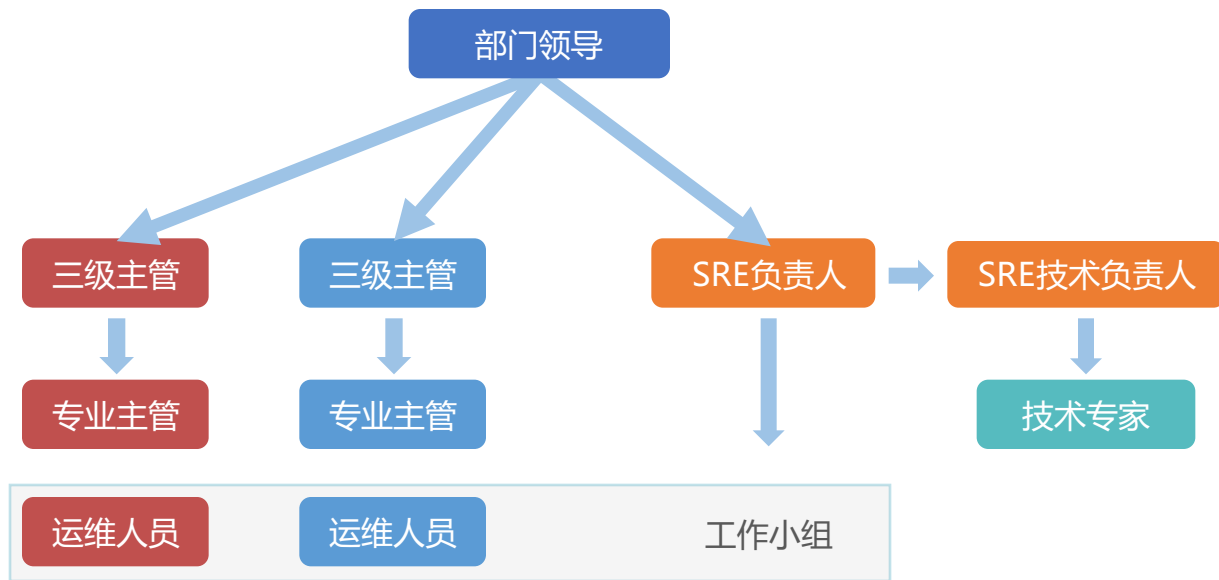
建立议事会负责制的半刚性组织架构制度：





改进激励制度

建立OKR导向的正向激励绩效倾斜制度



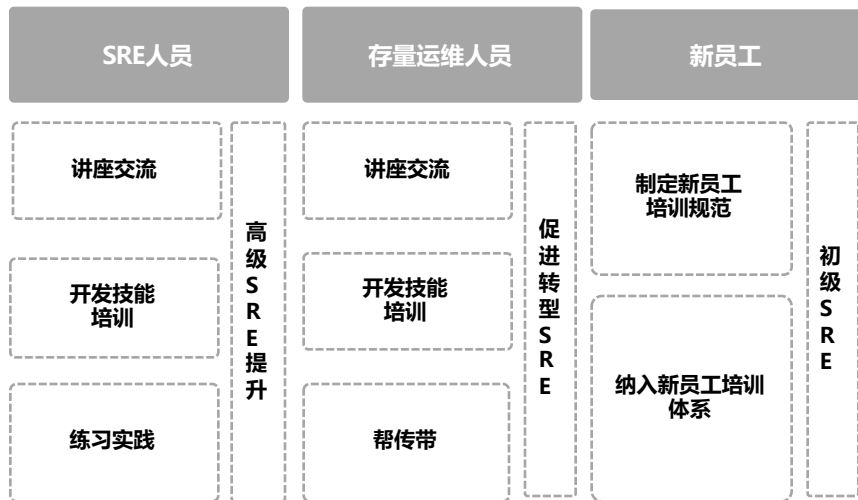


专注人员培养

SRE主要面向存量运维人员，推动IT化人员转型

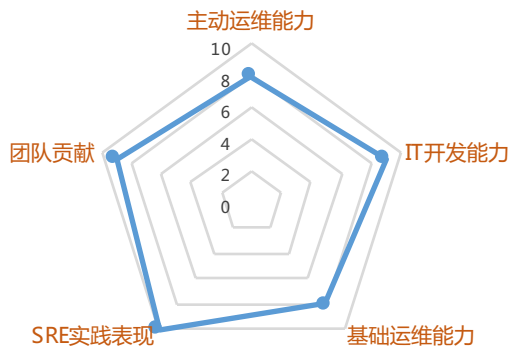
人员培养和创新

- **技能培养**：培养方向：依托IT和CT能力培养体系，**针对SRE人员、存量运维人员、新员工、协维人员**制定不同的人员培养体系



■ 人员成长：

- ◆ 从5个维度，21个评分项对SRE成员进行客观的综合评定，产生自画像。
- ◆ SRE成员输出个人OKR并纳入年度工作目标。





建立SRE日常工作框架，确定技术规范，避免技术债务。

完成技术选型

- 使用Python开发
- 规定Python框架和库的使用范围和版本

建立需求归口机制

- 参与协同或牵头应用开发需求
- 实现需求统一归口，避免重复开发，实现能力共享。

建立安全应对机制

- 建立开源软件安全评估准入机制
- 联合安全部探索开源安全防护新方法

建立开发规范

- 代码开发规范
- API调用规范
- 平台开放性规范
- 开发模式规范：小应用自开发、大应用联合开发

建立部署规范

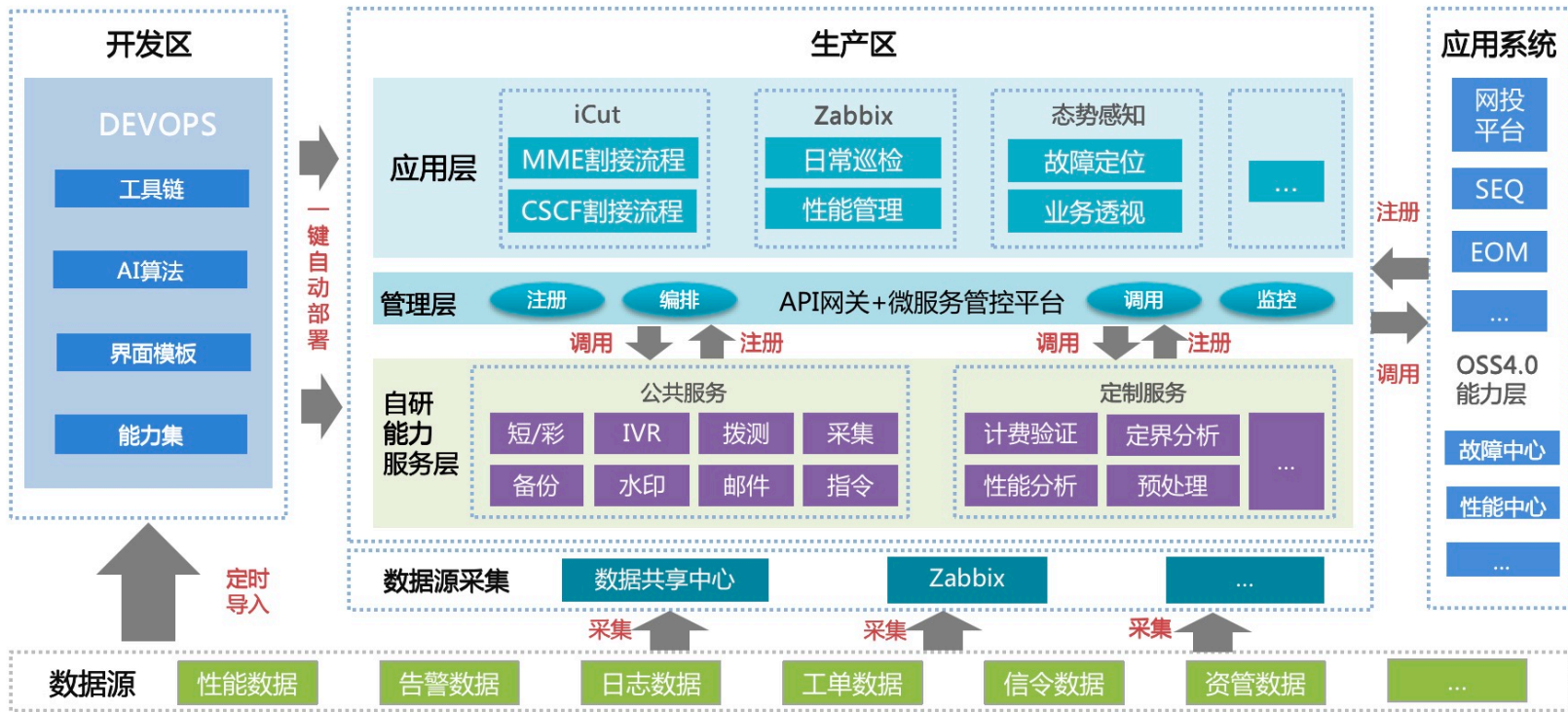
- 部署方式倾向于容器化和微服务化
- 通用公共组件抽象后微服务化部署



搭建开发支撑环境

通过建平台、解耦合两部分工作来搭建SRE的开发支撑环境。

1 建平台 SRE平台采用研发运维一体化模式，由两区三层组成，实现应用开发，同时输出服务能力。





搭建开发支撑环境

2

解耦合

推进原有支撑平台部分重构，使孤立封闭的、竖井式的平台解耦成开放、可扩展的运维开发支撑平台。



- ◆ 支撑平台分层解耦，各层间接口开放：通过开放接口，平台功能可供自研工具调用，提升开发效率。
- ◆ 共性的功能点进行独立，形成公共能力：同构调用公共能力降低开发难度。
- ◆ 支撑平台开放架构，可扩展插件：可以开发自研插件植入平台，扩展平台功能，降低投资成本。



SRE团队经过1年多的建设已出具成果，**共培养SRE团队成员30人**，完成**开发需求32个**。

01 需求实现快

基于开放解耦的平台进行搭积木式二次开发，单个功能点开发平均仅需**2个小时**，需求到上线时间**缩短551倍**。

02 痛点切入准

运维专家转型开发人员，网络痛点了然于心。
Cause劣化灵敏感知、节点负荷异常检测等应用能迅速感知异常，**早于用户投诉15分钟发现并完成故障处理，保证业务稳定。**



03 人力大解放

机器换人，完成32个运维应用开发，大量运维工作自动化，**值班人员减少55%**。

04 压力大释放

逐步接受与信任系统，**具体的运维工作逐步转交给自动化系统完成**，释放运维人员压力。
逐步过渡到系统干运维，运维人干系统。



Thanks

招聘广告

欢迎有主动运维思维的同学加入SRE团队

岗位要求：

- 1). 本科以上学历，2年以上python语言开发经验及团队开发经验
- 2). 责任心强，具有良好的沟通能力，逻辑思维能力强，工作积极主动；
- 3). 熟悉通信行业，有通信行业工作经验者优先。