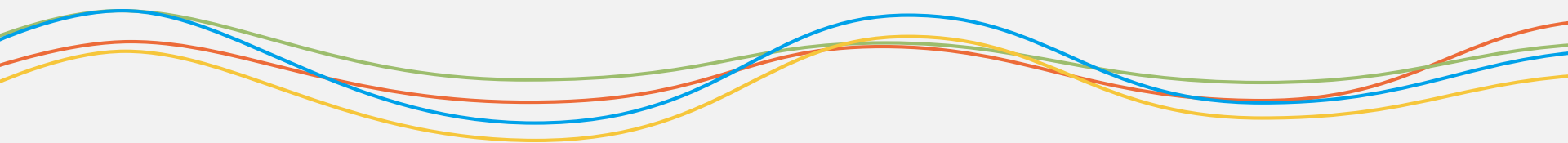


# 大宗能力交付从开始到未来

泰康保险集团运维自动化实践与前瞻



01

从开始到现在

02

从现在到未来

03

更远的未来

“真正的技术壁垒不是别人无法模仿的东西，而是别人压根就错误地低估甚至鄙视了的东西。”

微博 @白硕SH

20,000+  
设备/组件

400+  
人/内勤

30,000+  
日均保单数量

2.5亿  
客户总量

15%  
年增长率

500+  
套系统群



# 泰康保险集团

数据信息中心简介：致力于保险产品互联网化和互联网保险产品，以移动优先、算法驱动、云计算为主体建设思路。

泰康保险集团属于中小规模的IT组织

而自动化团队的定位是行业内的**追赶者**、**业务(第一驱动力)的服务者**

**进行业务互联网化转型**

**业务发展迅速**

**IT规模相近**

**大IT组织中想独立做运维自动化建设的分、子组织**

01

从开始到现在

02

从现在到未来

03

更远的未来

## 建设思路：

- 一个中心 —— 以监控为核心(以事件为监控的核心)；
- 两个维度 —— 时间维度(最大可能节约时间)、安全维度(最大可能确保安全)
- 从下至上 —— 从基础资源层到业务层；
- 从右至左 —— 运维视角出发进行自动化落地；

## 实际产出：

- 以资源交付为目标；
- 工具(脚本)为主要运维手段；
- 让IT资源的管理可控；

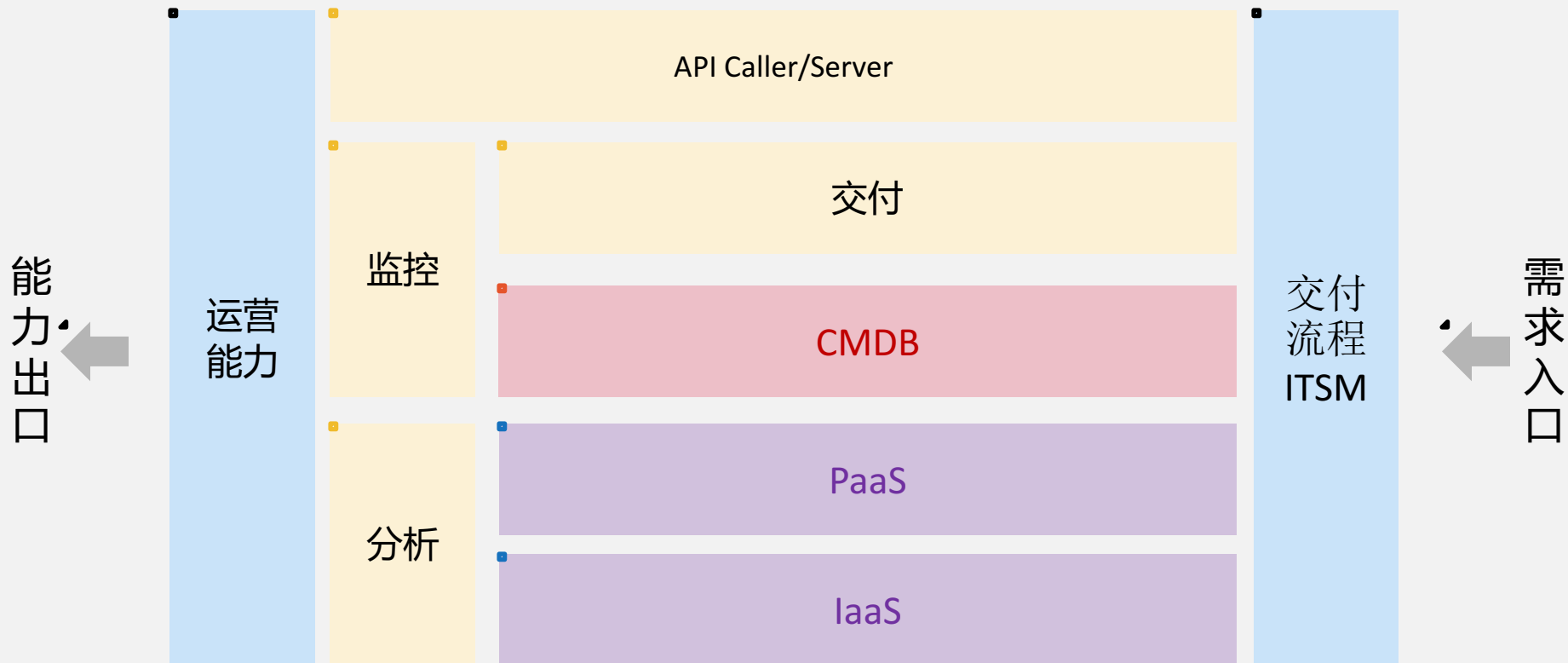
## 缺点：

- 离业务较远，无法快速响应业务要求；
- 效率低；
- IT问题往往不是运维侧的问题；

“当业务需求过来时，业务部门真正想要的是什么？”



“ IT组织中的运维部门 是 大宗能力的交付的部门 ”

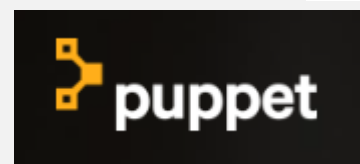


# 运维自动化平台结构模型 二级 现在 所有能力服务化



# 运维自动化平台 现在 产品对应

运营能力	自研	开源	
API	自研	开源	
监控	自研	开源	商用
分析	自研	开源	商用
交付	自研	开源	商用
CMDB	自研	商用	
PaaS	开源	商用	
IaaS	开源	商用	
ITSM	自研	商用	



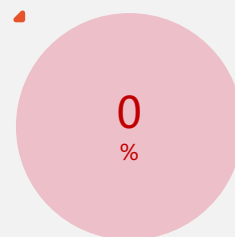
# 运维自动化平台 现在 一些指标



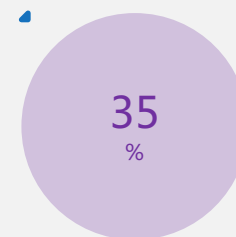
系统管理员团队  
人均管理设备数量



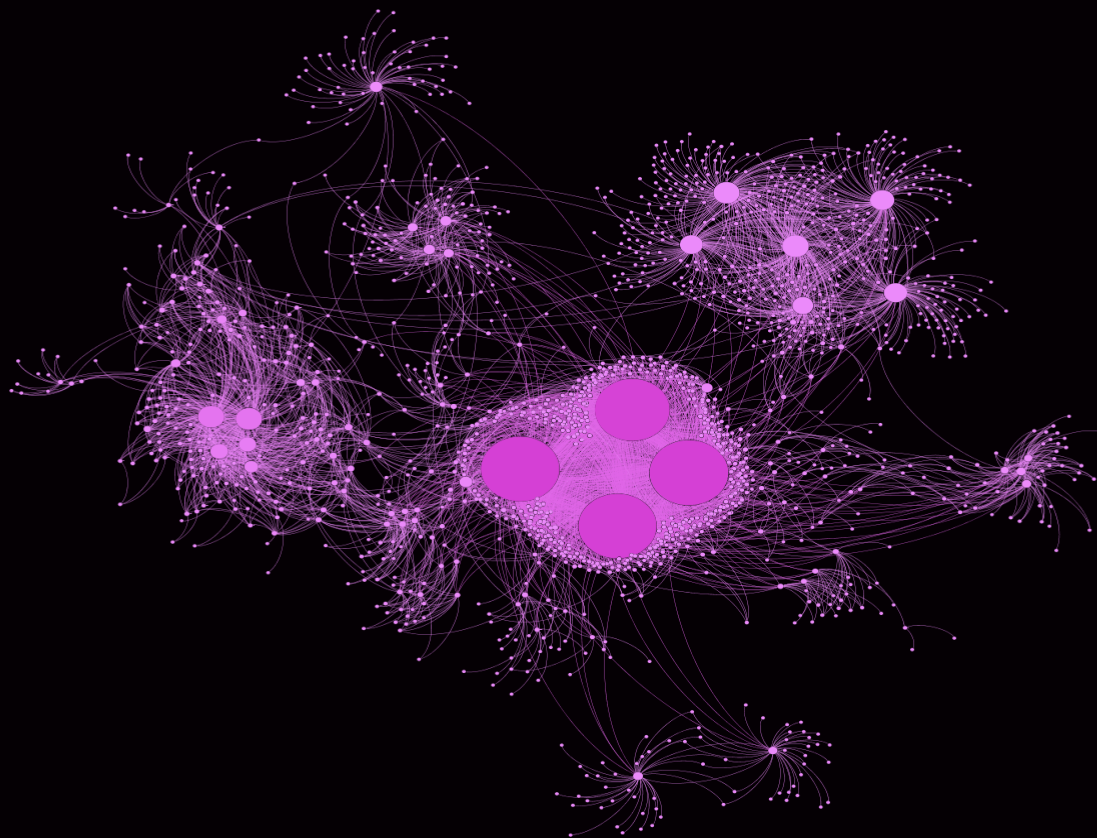
发布128个应用实例  
最快时间



运维人员  
离职率(近一年)



自助服务/工具  
占比



泰康内部IT某网段设备、组件关联关系自动发现  
智慧地图 ( Shell+Saltstack+ELK )

≡

系统	数量
Red Hat Enterprise Linux Server release 6.7 (Santiago)	535
Red Hat Enterprise Linux Server release 5.9 (Tikanga)	146
Red Hat Enterprise Linux Server release 6.4 (Santiago)	10
CentOS release 6.4 (Final)	4
CentOS Linux release 7.2.1511 (Core)	3
CentOS release 6.7 (Final)	2
CentOS release 6.6 (Final)	1

泰康内部IT某网段操作系统基线检查  
智慧地图 ( Shell+Saltstack+ELK )

# 运维自动化平台 现在 建设成果 应用性能分析案例



时间	Web应用过程	服务器响应时间(ms)
2017-04-27 09:17	Servlet/dispatcherServlet	3806
2017-04-27 09:18	Servlet/dispatcherServlet	3353
2017-04-27 09:18	Servlet/dispatcherServlet	3304

外部应用: bdcxzhfw-1 实例信息: t0-trolls-policy-server1:8080

Web应用过程: WebAction/SpringController/PolicyController/getPolicyInfo

URL: http://10.130.206.114:8080/api/v1/customers/legacyIds/policies

总耗时: 3772ms

应用层时间

最慢组件	调用次数	持续时间(s)	占比(%)
org.apache.http.impl.client.CloseableHttpClient.execute	1	3.788	99.527
com.netflix.zuul.http.ZuulServlet/service	1	0.011	0.289
org.springframework.cloud.netflix.zuul.web.ZuulController/handle	1	0.002	0.053
org.apache.tomcat.websocket.server.WsFilter/doFilter	1	0.001	0.026
javax.servlet.ServletRequestListener/requestInitialized	2	0.001	0.026
org.springframework.boot.web.filter.OrderedHiddenHttpMethodFilter/doFilter	1	0.001	0.026
org.springframework.boot.actuate.autoconfigure.MetricsFilter/doFilter	1	0.001	0.026
org.springframework.web.servlet.DispatcherServlet/service	1	0.001	0.026

org.apache.tomcat.util.net.SocketProcessorBase.run	(SocketProcessorBase.java:49)
java.util.concurrent.ThreadPoolExecutor.runWorker	(ThreadPoolExecutor.java:1142)
java.util.concurrent.ThreadPoolExecutor\$Worker.run	(ThreadPoolExecutor.java:617)
org.apache.tomcat.util.threads.TaskThread\$WrappingRunnable.run	(TaskThread.java:61)
java.lang.Thread.run	(Thread.java:745)
com.netflix.zuul.http.ZuulServlet/service	caused by com.netflix.client.ClientException: Number of retries on next server exceeded max 1 ret...
com.netflix.loadbalancer.LoadBalancerCommand\$4.call	(LoadBalancerCommand.java:345)
rx.internal.operators.OperatorOnErrorResumeNextViaFunction\$4.onError	(OperatorOnErrorResumeNextViaFunction.java:140)
rx.internal.operators.OperatorRetryWithPredicate\$SourceSubscriber\$1\$1.onError	(OperatorRetryWithPredicate.java:107)

泰康内部IT某应用的性能分析案例, 至代码级

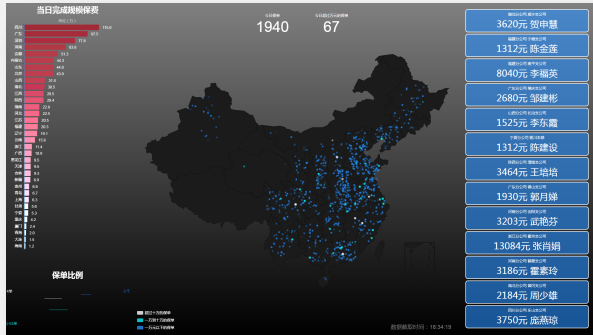
APM产品



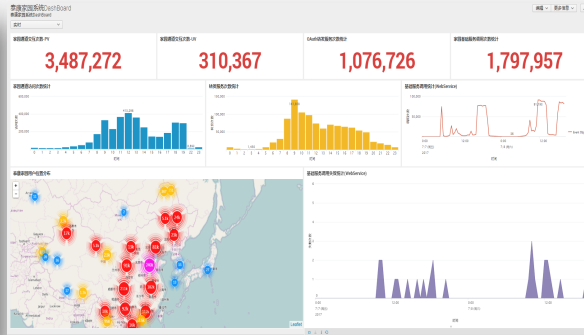
# 运维自动化平台 现在 建设成果



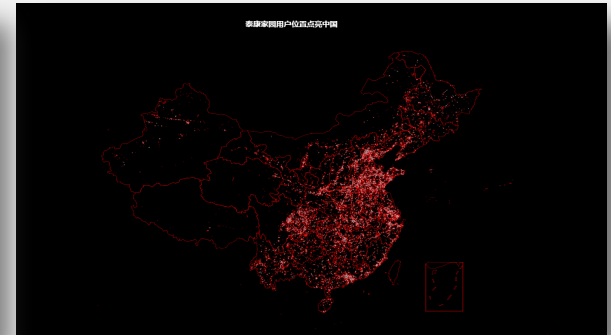
### 业务业绩



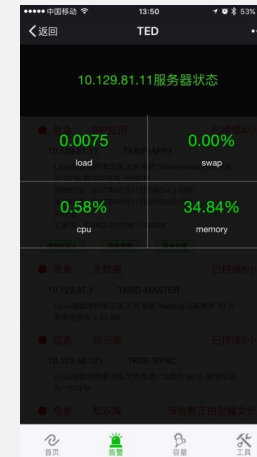
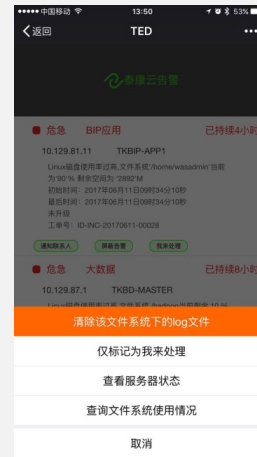
### 接口调用



### 用户分布



## 微信端运维工具



泰康内部用户、业务、业绩、接口、基础资源若干工具和可视化交付样例

D3 js + Echarts + JQuery + WeUI

# 运维自动化平台结构模型 现在 说说开源评估体系



用商业产品达到精通的程度，是快速构建服务能力的最快的方法；  
开源不代表自主可控，来自互联网公司的陷阱；  
对开源产品进行社区成熟度、产品成熟度、企业成熟度等多角度的基本面进行评估；  
别人用什么我就用什么？

评估项	评估内容	打分标准	评估结果
开发协作模式	开发协作模式: Pull Request, issues 托管平台, 如下: 开发协作: PR 在github.net上 (https://github.com/mongodb), issues 在jira.mongodb.org上	GitHub托管10分, launchpad/google code等, 其他0分	0
	社区用户对托管平台, 如下: 社区项目在Google Groups上 (http://groups.google.com/group/mongo-user) 社区项目托管在托管平台, 如下: 邮件列表在 Google Groups上 (http://groups.google.com/group/mongo-dev) 是否有FreeNode IRC 核心研发讨论社区, 如下: IRC 在FreeNode上 (http://webchat.freenode.net/?channels=mongodb)	自主社区 (web) 提供讨论和反馈, 托管在其他技术社区5分, 其他0分 有邮件列表10分, 其他0分 有IRC 10分, 其他0分	10 0 0
发布频率	代码发布频率, 来源代码托管对应的托管网站, 如GitHub (launchpad/google code等) 如每周有update, 每周有小版本release, 每月有补丁升级发布	代码以天更新10分, 以周更新5分, 其他0分	5
发布流程	是否有发布策略和流程公开网站, 如 有release策略公开网站, 分成(Active/Stable) (https://github.com/mongodb/mongo/branches#) 发布	提供发布流程和策略和明确流程的10分, 发布流程和策略公开, 无明确流程的5分, 其他0分	10
产品质量	Issue跟踪和阻止新版本发布完成时: Issue能否跟踪, 及时性, 功能/性能等全部提交问题 3.2版: 118 open, 338 closed(75.2%), Critical Bug 32个, Non-Critical Bug 465个 3.0版: 143 open, 599 closed(80%), Critical Bug 102个, Non-Critical Bug 65个	Issue数量在10个以内为满分, 如果问题在3.2%, 则满分 30 Critical/Bug/Non-Critical Bug 2比值为权重, 比值越高, 需要权重比值越低分	0
产品标准化	核心数据结构和通信架构或实现是否有明确行业标准化说明 如数据库行业标准的SQL和数据库的SQL 客户端服务协议: 2Wire+eth, 基于常规TCP/IP Socket, 请求, 应答风格	数据库符合行业标准10分, 部分符合5分, 完全私有0分 外部通信协议符合行业标准10分, 部分符合5分, 完全私有0分	10 10
产品模块化	如MongoDB服务器为mongodb, 和对应各种语言的Client, 有专用的配置管理工具	完全模块化10分, 部分模块化5分, 完整集成0分	10
开发人员数量	开发人员数量 有持续运行管理计划的, 根据排名打分, 前10名10分, 10-20名5分, 20名后0分, 无排名0分	开发人员超过100根根10分, 0-100之间按人数/10计算	0
社区活跃度	来源: 网上: github.com (Pull Request) 统计: 18 Open, 1094 Closed, 邮件列表: 200-10 post/day 社区活跃度: 来源: 项目官方网站: 网上活动在社区 (博客, 新闻, 项目) 社区项目 (Share, net, Twitter, Youtube) 线下主要活动(MUG, MongoDB World, 各地MongoDB线下开发者和用户会)	社区活跃度 10分, 10取整为满分, 开发社区每天超过10 new post 添加1分 社区官方主办的全球性会议和线上社区开发者和用户会议 10分 仅有社区活动 5分, 其他0分	0 10 0
社区影响力	如创始团队前身为10gen, 目前仍由MongoDB社区的核心开发组成	创始团队控制项目研发的10分, 参与社区研发的5分, 其他0分	5
商业市场影响力	如创始团队前身为10gen, 从2007年开始以开源方式开发MongoDB, 2009年开源后继续开发	项目创始团队直接创办的商业企业 10分, 参与创办的企业5分, 其他 0分	10
			总计 80/160 (50%)

评估项	评估内容	打分标准	评估结果
服务形式	加 MongoDB 官方服务仅提供社区支持, 包括: 官方GitHub(https://github.com/antirez/redis) PR提交 官方JIRA(https://jira.mongodb.org/issues)提交 官方邮件列表(http://groups.google.com/group/mongo-user) 问题交流 第三方商业社区托管 (Stackoverflow上 #Redis标签, ServerFault上有mongodb.org问题标签) MongoDB, Inc. 作为项目发起公司和商业支持公司, 提供企业版和商业支持服务	有JIRA/bugzilla等在跟踪问题和官方社区SLA控制 和其他社区支持通道 10分 仅有GitHub/issue 系统提供正式的Bug提交 5分 仅有官方社区论坛, 邮件列表, IRC渠道 3分 其他0分	3
服务内容	加 MongoDB官方服务仅提供社区支持, 社区SLA MongoDB, Inc. 作为项目发起公司和商业支持公司, 提供企业版和商业支持服务 (SLA形式, 沟通渠道, 邮件支持) SLA Professional Support - Blocker: 2小时, Critical 4小时, Major 8小时, Minor/Trivial 48小时 SLA Advanced Support - Blocker: 1小时, Critical 2小时, Major 4小时, Minor/Trivial 24小时	同时有官方社区支持SLA和商业支持SLA 10分 仅有官方社区支持SLA 5分 仅有商业支持SLA 3分 全部没有 0分	3
服务响应能力	加 MongoDB官方服务仅提供社区支持, 无SLA, 无实时响应时间 MongoDB, Inc. 作为项目发起公司和商业支持公司, 提供企业版和商业支持服务, 无中文本地化服务	同时有官方社区和商业支持响应时间 10分 仅有官方社区中文支持时间 5分 仅有商业支持响应时间 5分 全部没有 0分	3
服务本地化	加 MongoDB官方社区无中文本地化, 由一公司中国籍员工建立了MongoDB中文社区 MongoDB, Inc. 作为项目发起公司和商业支持公司, 提供企业版和商业支持服务, 无中文本地化服务	同时有官方社区和商业支持中文本地化 10分 仅有官方社区中文本地化 5分 仅有商业支持中文本地化 5分 全部没有 0分	10
服务影响力	加 MongoDB官方社区通过本表第一系列形式提供, 核心开发人员提供直接支持 MongoDB, Inc. 作为项目发起公司和商业支持公司, 提供企业版和商业支持服务	同时由官方社区核心团队提供社区支持和商业支持服务的 10分 仅有官方社区核心团队提供社区支持的 5分 仅有第三方商业机构提供技术支持的 5分 商业机构服务能力为满分10分, 1万到2万, 5千-1万 1分	10
服务的成本	加 MongoDB官方社区通过本表第一系列形式提供, 用户接受服务成本为零 MongoDB, Inc. 作为项目发起公司和商业支持公司, 提供企业版和商业支持服务 (前提是MongoDB企业版企业, 服务成本为按年计算的MongoDB企业产品订阅服务费)	客户接受服务成本受到地区和行业不同有较大差异, 此处评估为参考项。	0
			总计 24/50 (48%)

评估项	评估内容	打分标准	评估结果
项目年龄	1998 (来源: 项目主页, 项目官方github, 或Wikipedia)	10年内1年1分, 超过10年的10分	10
项目亮点	流行度/应用/排名/网站上的转载	有持续流行管理计划的, 根据排名打分, 前10名10分, 10-20名5分, 20名后0分, 无排名0分	10
许可证类型	短写项目主页官方许可证说明 GPLV2	Apache /BSD /MIT License 10分, LGPL GPLv3 5分, GPLv2 0分	0
发布模式	社区发布方式(来源: 官方社区主页)	直接由社区提供 10分, 其他0分	10
	商业发布和商业服务的正式对外发布时间(来源: 项目主页公司的官方发布)	有明确商业公司支持和稳定的商业发布计划的, 从建立和对外发布10年内1分, 超过10年的10分, 没有商业公司支持和稳定的商业发布计划的 0分	10
版本控制	是否有三副本结构 如版本号为: 主版本号(major), 次版本号(minor), 补丁级别(patchlevel) (x.x.x格式)	有开发发布标准三副本结构的10分, 其他0分	10
	是否有专门的ReleaseNotes, 发行记录页面	有稳定和稳定的明确版本的10分, 其他0分	0
	是否有专门的ReleaseNotes, 发行记录页面	稳定的版本号为在1.102范围内, 1个版本1分, 超过10个版本的, 10分	3
	是否有专门的ReleaseNotes, 发行记录页面	有明确的ReleaseNotes, 发行记录发布页面的10分, 其他0分	10
	是否有专门的ReleaseNotes, 发行记录页面	发行记录有明确的级别划分的10分, 其他0分	10
	是否有专门的ReleaseNotes, 发行记录页面	版本(次版本和补丁)升级速度在一个月内3分, 1-3个月之间的1分, 3个月以上的0分	10
支持平台	支持平台的Unix, *BSD, OS, Linux/Ubuntu, RHEL/CentOS, SuSE/SLES, Windows 10分, 有一个平台支持5分	支持平台的Unix, *BSD, OS, Linux/Ubuntu, RHEL/CentOS, SuSE/SLES, Windows 10分, 有一个平台支持5分	10
	是否有专门的ReleaseNotes, 发行记录页面	自身开发环境平台的10分, 有一个平台支持5分	10
	是否有专门的ReleaseNotes, 发行记录页面	建议全平台生产部署和运行的10分, 有一个平台支持5分	10
	是否有专门的ReleaseNotes, 发行记录页面	有图形界面安装的10分, 命令行安装5分, 有图形界面编译的5分	10
			总计 113/150(75%)

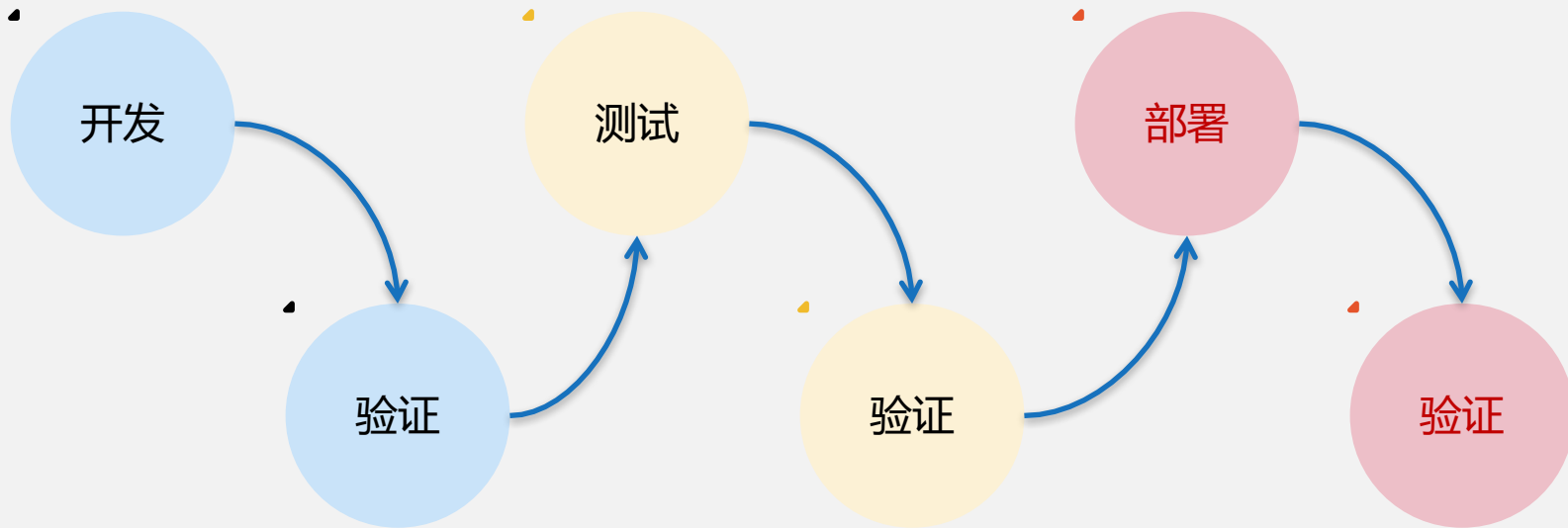
评估项	评估内容	打分标准	评估结果
可靠性	MongoDB官方提供集群方案, 实现: 高可用, 分区数据和最终一致性 集群由应用层节点(mongos)无状态和实现副本节点(replica sets)/mongo/节点	提供高可用, 分区数据和最终一致性 10分 仅提供高可用和分区数据最终保护能力 5分 提供弱一致性分区数据能力 1分 其他0分	10
可扩展性	MongoDB官方提供MongoDB_shardedCluster, 支持水平扩展, 自动分片, 自动实现平衡(内置balancer 模块, 自动实现进行平衡), 分片算法支持Hash和Range.	提供自动分区数据分片分布式计算, 并提供自动平衡 10分 提供自动分区数据分片分布式计算, 并提供手动手工数据平衡 5分 其他0分	5
性能	MongoDB官方提供mongo-perf基准测试工具, 支持工业标准VCSO基准测试	性能基准模块不同环境和测试用例结果各有不同, 此处为参考项	0
可用性	MongoDB集群有限制如下: 应用连接多"mongos"节点, mongos节点故障可以快速在本节点或其他节点重启。Config管理集群无限制, 从3节点开始, 可以自mongo/节点通过replica sets方式创建。后续每一个分片添加一个"mongos"节点, 多"mongos"节点为"replica sets"的Master节点, 支持高可用, 支持高可用自动failover, 支持分区管理, 支持高可用自动failover, 支持分区管理	支持多副本部署, 高可用自动failover, 支持分区管理和对等节点, 支持集群冗余设计 10分 支持多副本部署, 高可用自动failover, 支持分区管理和对等节点 5分 支持多副本部署, 高可用自动failover, 支持分区管理, 支持高可用自动failover, 支持分区管理 1分, 其他0分	10
易用性	MongoDB官方提供提供命令界面(CLI)和管理脚本, 可以通过管理脚本完成部署、配置、监控任务, 无需脚本。 MongoDB官方提供企业级运维工具Ops manager(实现部署、配置、监控、备份)	命令界面(CLI), 有图形管理界面(GUI), CLI/Gui/脚本完成大部分操作, 管理和运维部署任务 10分 命令界面(CLI), 有图形化监控和告警界面(GUI monitoring/alerting) 5分 命令界面(CLI), 能够通过脚本完成所有管理任务 3分, 其他0分	10
平台独立性	MongoDB使用C++和C语言开发, 对硬件和运行环境无依赖, 部署需要二进制安装, 无外部依赖。	完全无依赖/有少量依赖/网络/磁盘/非标准库 10分 无依赖/有少量依赖, 仅在标准库/网络/磁盘/非标准库 5分 有依赖/处理器/网络/依赖 0分	5
安全性	MongoDB官方支持ACLSHAM-SHA-1和OS, LDAP(Kerberos), 支持基于角色的控制RBAC(User Role, 特殊角色Collection Level), 角色继承, 支持加密通信(TLS/SSL证书, Identity验证, IP限制), 支持审计, 支持HTTPS/REST	有ACL, 有基于角色控制的RBAC, 支持用户权限访问控制, 支持客户端加密, 支持TLS/SSL证书, 支持基于角色的加密或支持 10分 有ACL, 支持基于角色的加密, 支持工业标准的加密算法保护 5分, 其他0分	5
可管理性	MongoDB官方社区没有提供监控/告警和备份外部部署工具。 MongoDB, Inc. 公司在企业版中提供(Ops manager)	有完整的控制, 报警备份外部部署工具 10分, 其他 0分	10
文档完备性	MongoDB官方社区提供了完备的开发者文档和集群部署和运维管理文档	有开发者和运维两方面完整文档的 10分, 有其5分, 其他 0分	5
故障恢复性	MongoDB有JIRA之类的故障库, 产品代码bug主要通过JIRA issue系统提交和处理	有故障库系统 如JIRA/Bugzacc之类的 10分, 仅有issue系统 5分, 其他 0分	0
培训	MongoDB, Inc. 公司提供培训内容和培训服务	有官方社区培训和培训的 10分, 有其5分, 其他0分	10
供应商独立性	MongoDB, Inc. 作为项目发起公司和商业支持公司, 提供企业版和商业支持服务	有多家第三方服务商且有本地服务的 10分, 无本地服务 5分, 其他0分	10
			总计 80/110(73%)

误操作是运维动作中，引发事故和故障较多、且毁灭性较大的一点。

在这一点上，我们认为 人应该面对程序，程序去面对环境。

涉及内容：

1. 工具封装(脚本/工具管理、脚本/工具规范等)；
2. 完备的开发、测试、生产环境的工具发布流程；



# 运维自动化平台结构模型 现在 说交付

基础资源的交付和应用的交付是IT组织需要打通的任督二脉。

最终都会落地在“运维管理平台”中，运管是IT组织中所有角色的门户。



## 标准化

- 基础，和秦始皇统一度量衡的意义是一样的；
- 同类组件的标准规范在不同的业务系列可以不同；
- 资源、日志、报文、流程等等；

## 自动化

- 规范和标准的技术校验和落地；
- 平台建设：监控平台、分析平台、自动化平台、作业调度、持续集成、持续交付；
- API化、场景化、多纬度视角；

## 数据化

- 用数据说话才有意义；
- 数据指标的可度量是优化依据；

说说CMDB：

- CMDB对象关系要简化(父子、依赖、连接)；
- CMDB管理大于技术，需要自动化但不能依赖自动化
- 如果不知道如何开始建设，就从消费CMDB的场景入手；

- 人最宝贵的是时间，节约时间是重要的事情；(提供了若干和IT其实无关的工具交付 活动报名、签到、统计、值班、报表呈批等)
- 安全和效率在很多场景下是相悖的；(适当的取舍，效率、服务、成本等)
- 业务和应用的治理，需要先从架构角度入手；
- 改变工作协作方式，其实可能是最立竿见影的提升效率的手段；(@凤凰项目 一起寻找瓶颈)
- 鼓励大众创新，运维自动化平台的平台化可以无缝的接入创新成果；
- 影响效率最大的是审批环节，当前ITIL流程的形成很多是亡羊补牢的结果，需要针对性的优化、简化；

01

从开始到现在

02

从现在到未来

03

更远的未来

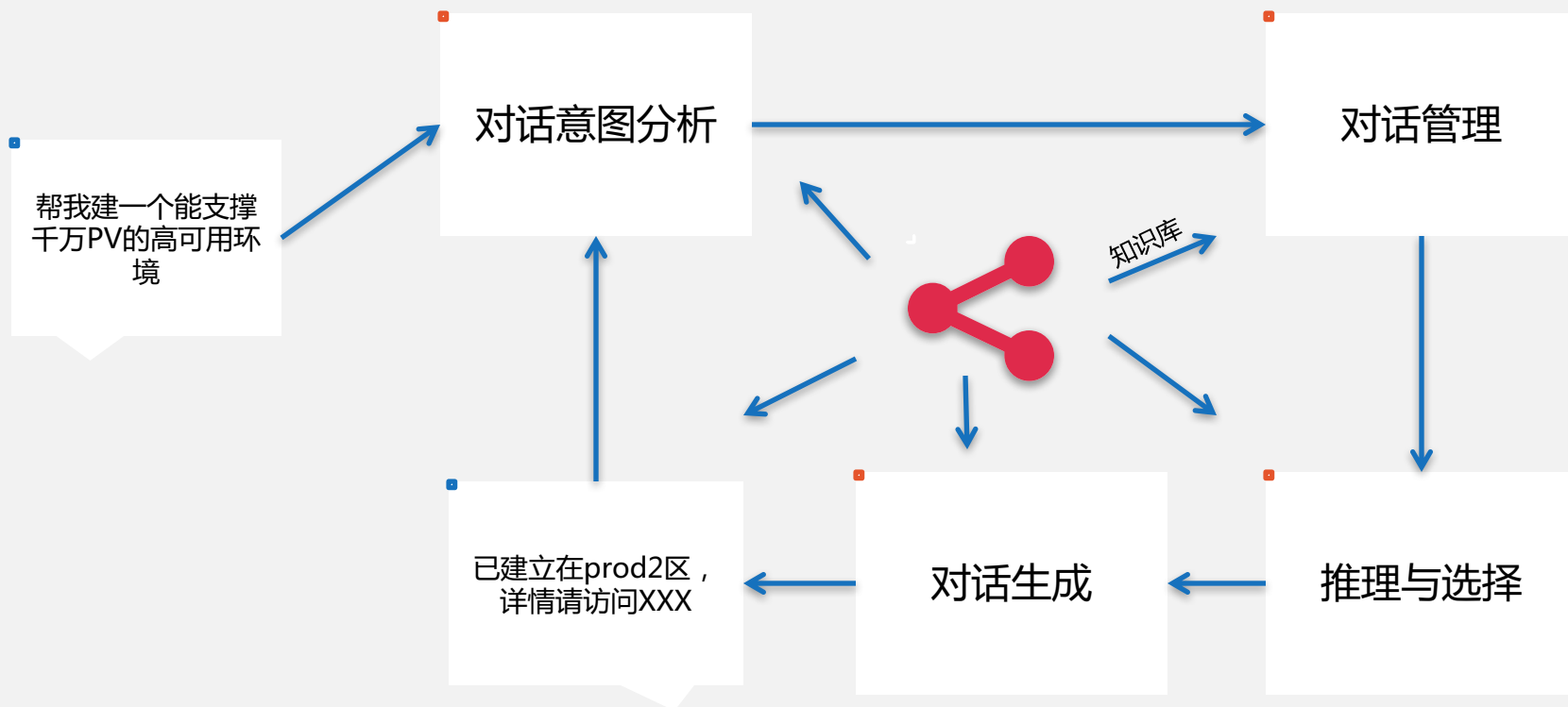


希望在泰康业务发展的同时能达到现在亚马逊、Google、BAT等技术驱动型公司在Devops上目前达到的成绩；

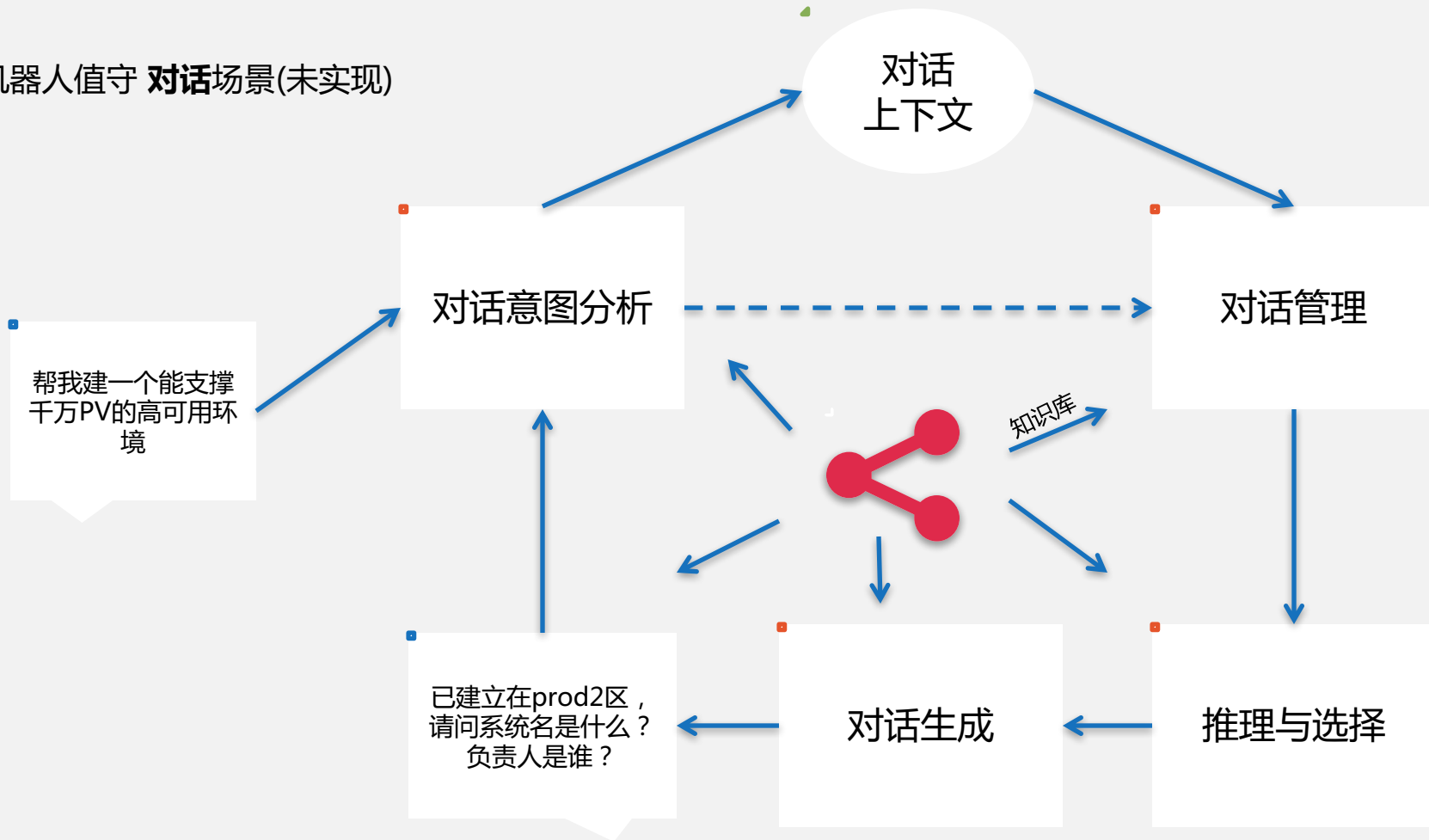
- Spring Cloud 微服务；
- 机器人运维；(巡检、知识库问答、服务台)测试阶段；



## 机器人值守 问答场景



机器人值守 **对话**场景(未实现)



01

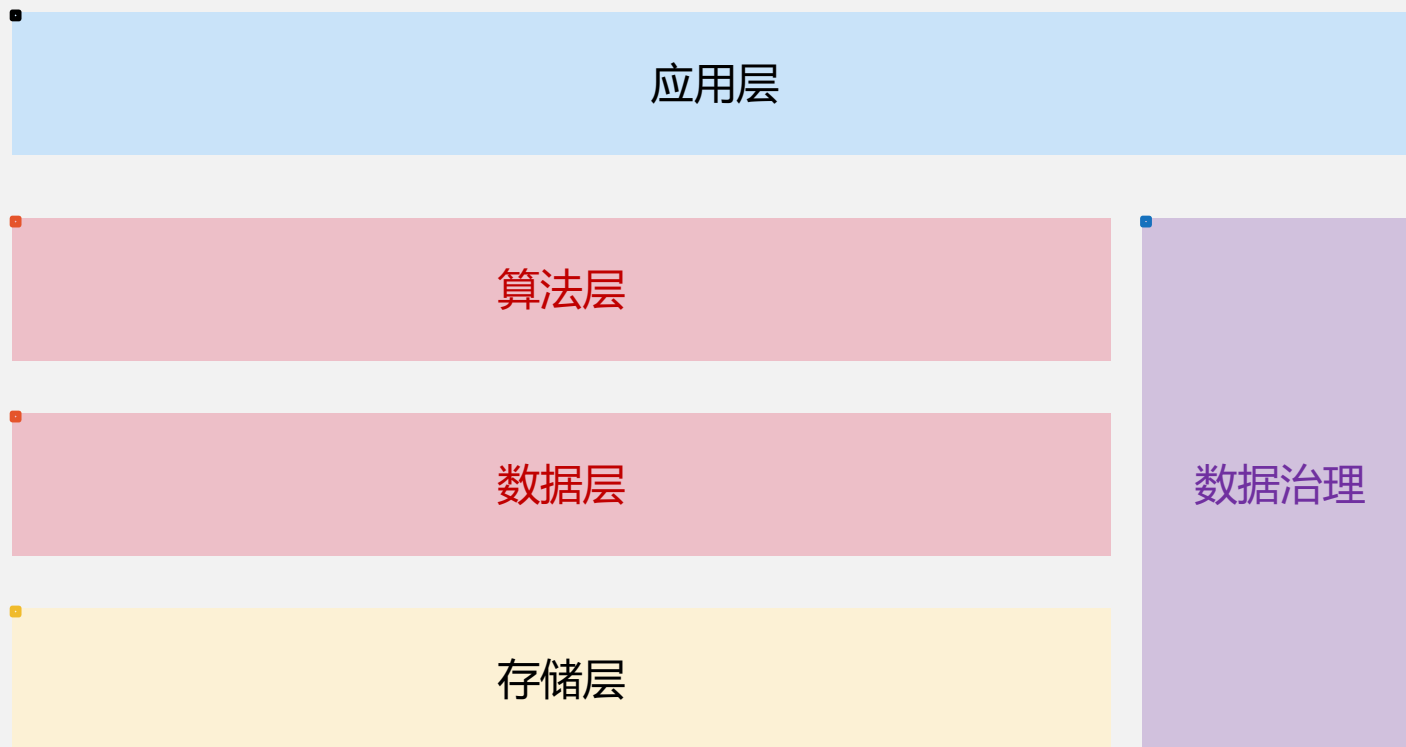
从开始到现在

02

从现在到未来

03

更远的未来



客观地讲，我们目前离互联网公司还有一定的差距，短时间内还是以一线互联网公司为追赶对象：

- 自动；
  - 自发；
  - 自愈；
  - 有用；
- 
- **智能运维是数据分析的结果，由AI 自动去落地到IT环境中；**

# DevOps

带我不死  
带运维不死

**THANK YOU**

