

互联网触碰传统产业背后的技术演进之路

丁丁

- 消费电商 vs 大宗产业电商
- 找钢网的技术演进过程
- 大运维体系如何驱动研发

## 消费电商



用户基数大



商品信息标准、价格稳定



交易习惯好、交易量巨大



运输流通透明

## 大宗产业电商



用户基数小



商品信息乱、价格波动频繁



交易环节复杂、单笔额度极高



运输成本高昂

## 消费电商



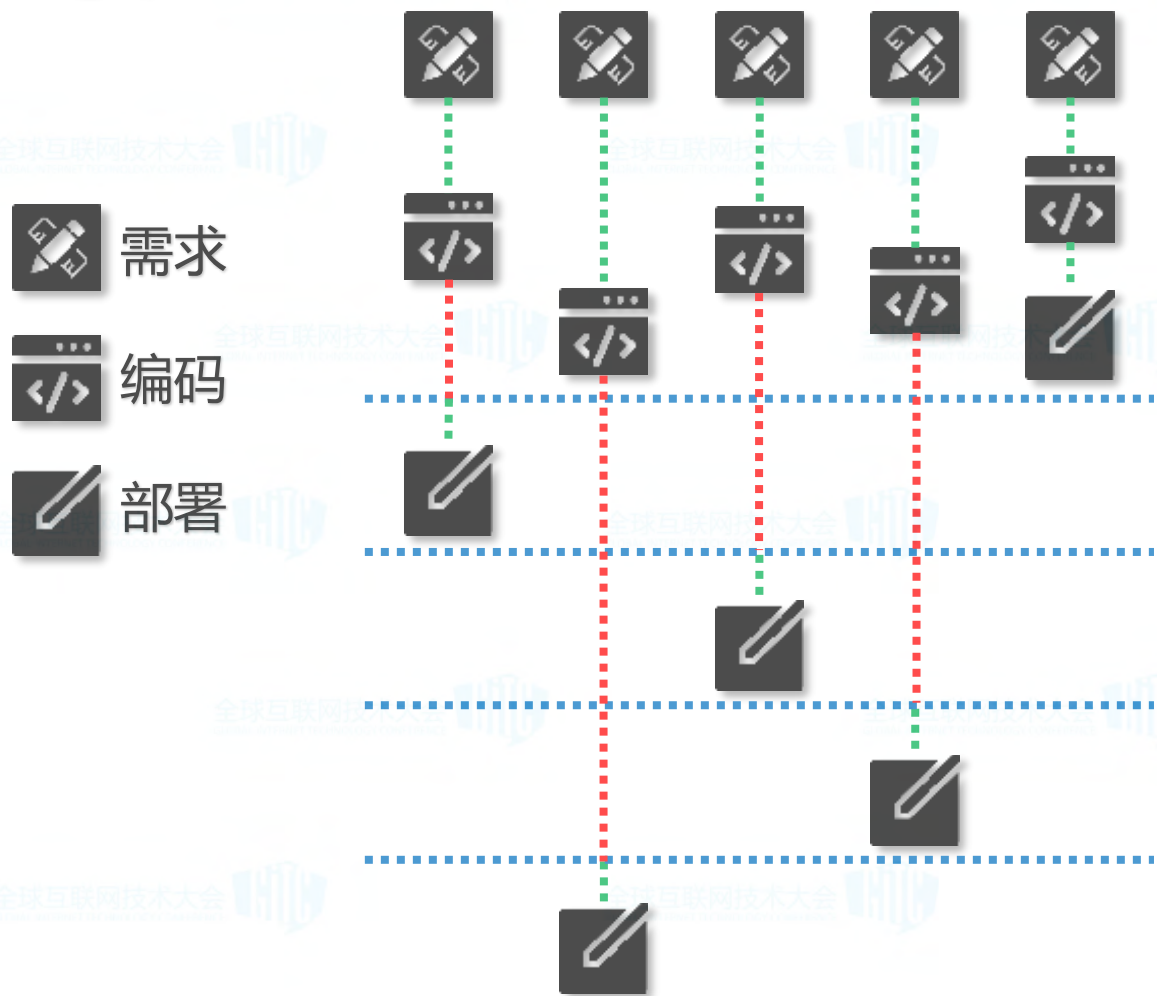
## 大宗产业电商



没有并发的电商系统应该怎么玩？



## 1. 没有并发，要不要分布式



### 常规的经验

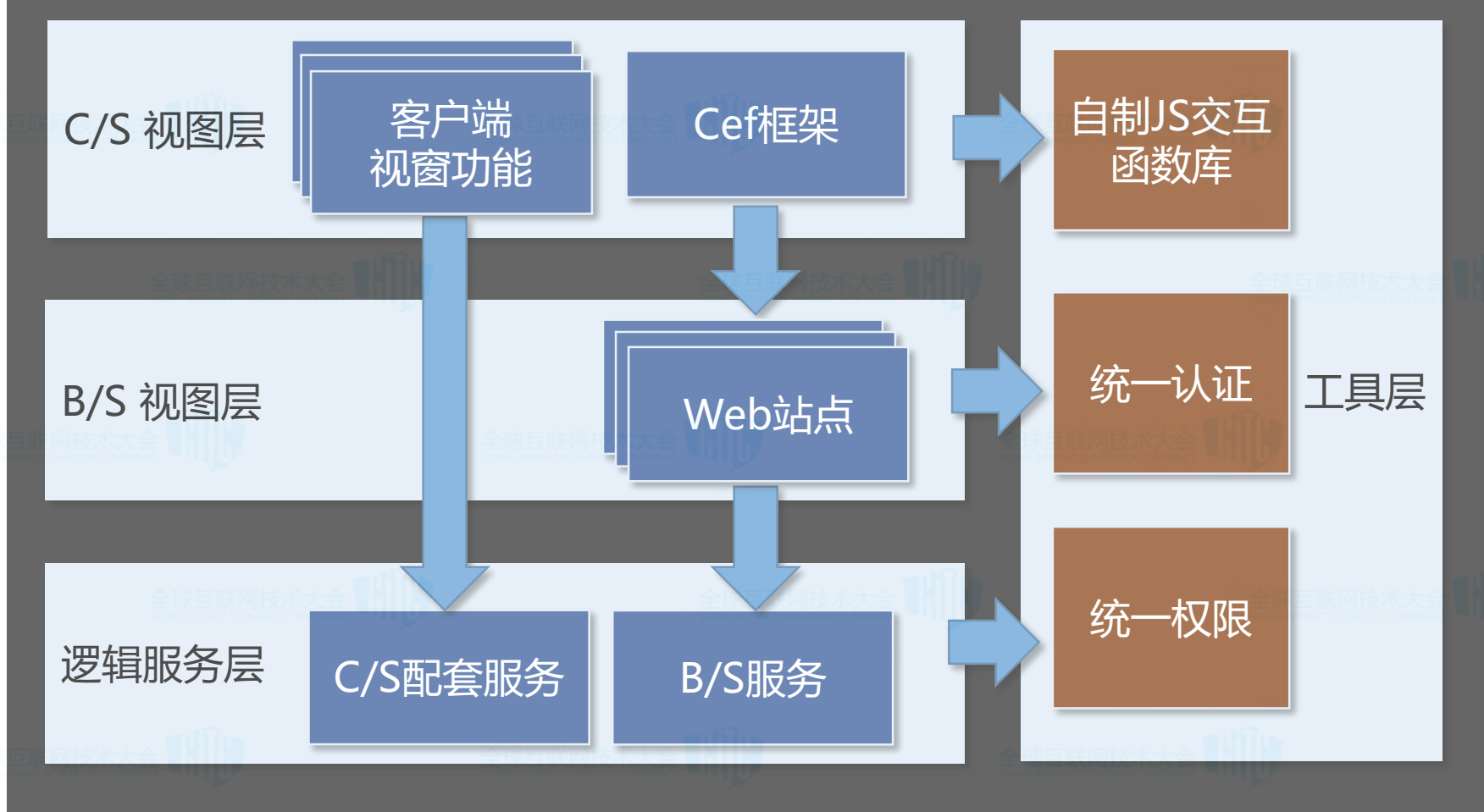
- 单机顶得住就没必要



### 实际的情况

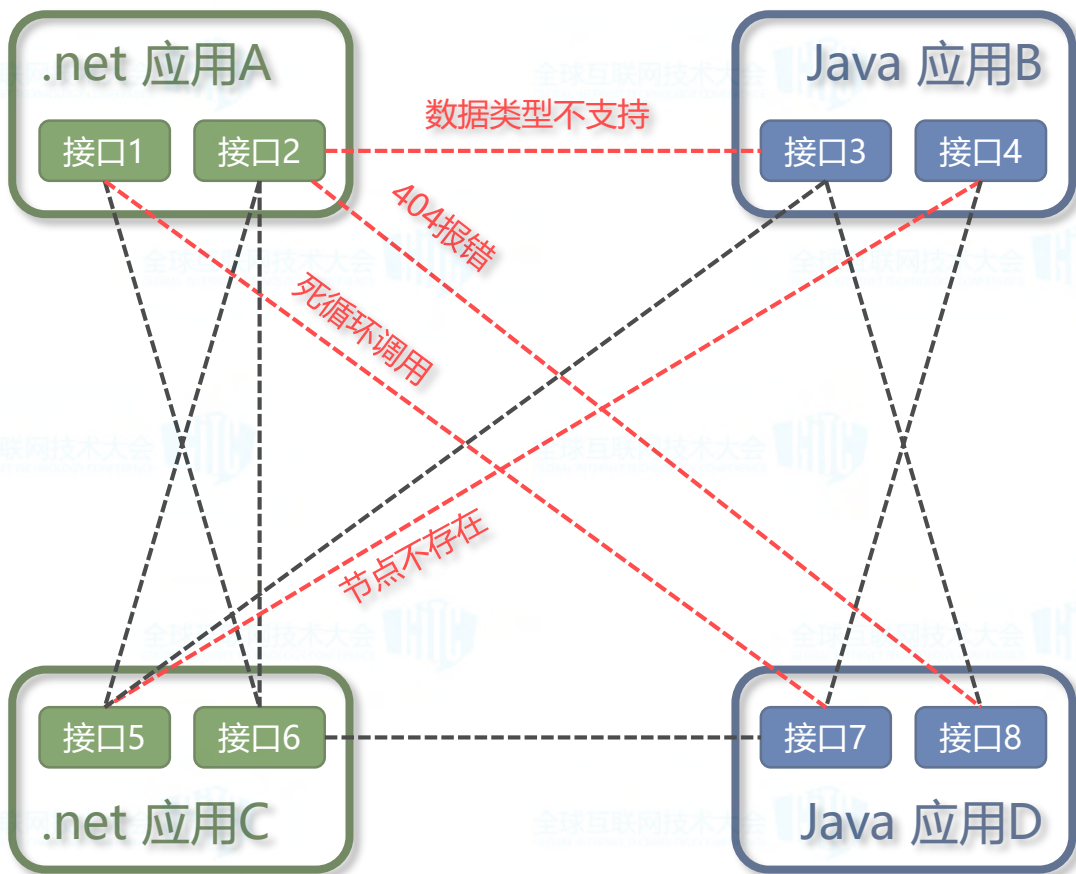
- 功能的多少与并发量并不具备捆绑性质的参考，单机系统功能多到一定程度后，代码的坏味道已经很浓郁，需求消化速度已经跟不上，牵一发动全身，而找钢的单机系统还是一个C/S架构的老古董.....

## 视图混合架构过渡方案





## 2. 没有并发，要不要服务治理



### 常规的经验

- 没并发就不需要弹性伸缩，没必要治理。

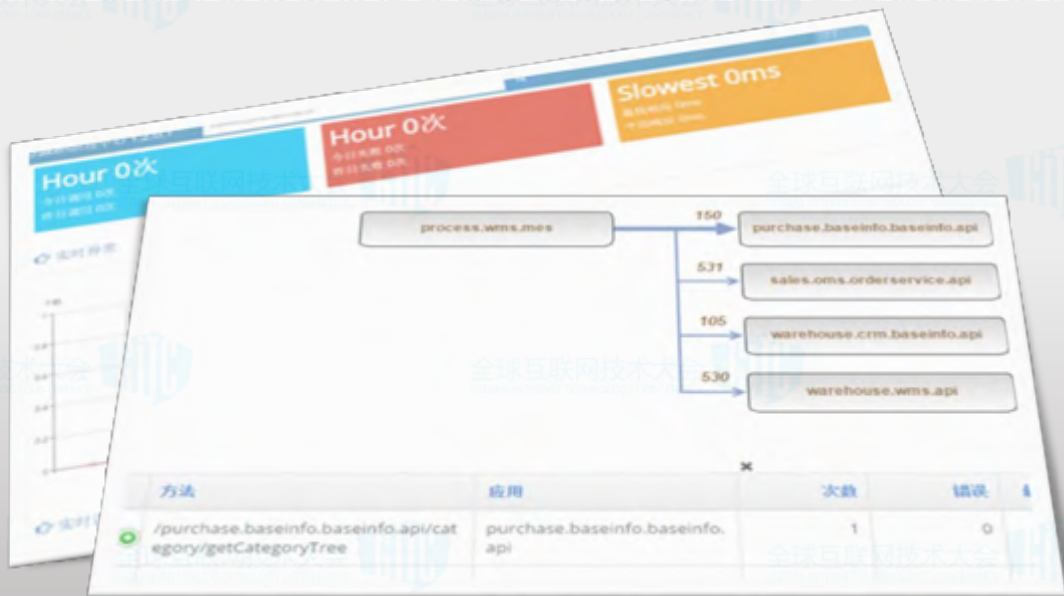


### 实际的情况

- 决定治理的因素不仅仅是并发量，技术栈的数量、系统规模都是重要的考量。如果没有治理，系统之间的关系就无法透明，不同技术栈的特殊数据类型无法识别，系统交互类似蜘蛛网，任何一个跨系统跨团队的项目都无法正确评估项目边界.....



### 轻便的分布式服务治理框架



## 跨语言支持

Kraken从一开始就为多语言支撑做好了准备，借鉴了Thrift Ice Hessian等RPC通讯框架，设立了轻量的中间数据类型，支持大多数语言的常用数据类型。

## 柔性落地

Kraken诞生之初就为企业级应用升级服务治理能力做好了准备，支持大多数主流的开发框架，以插件的形式附加在开发框架之上接管请求管道，无需大刀阔斧的改造系统。

## 丰富的IDE插件

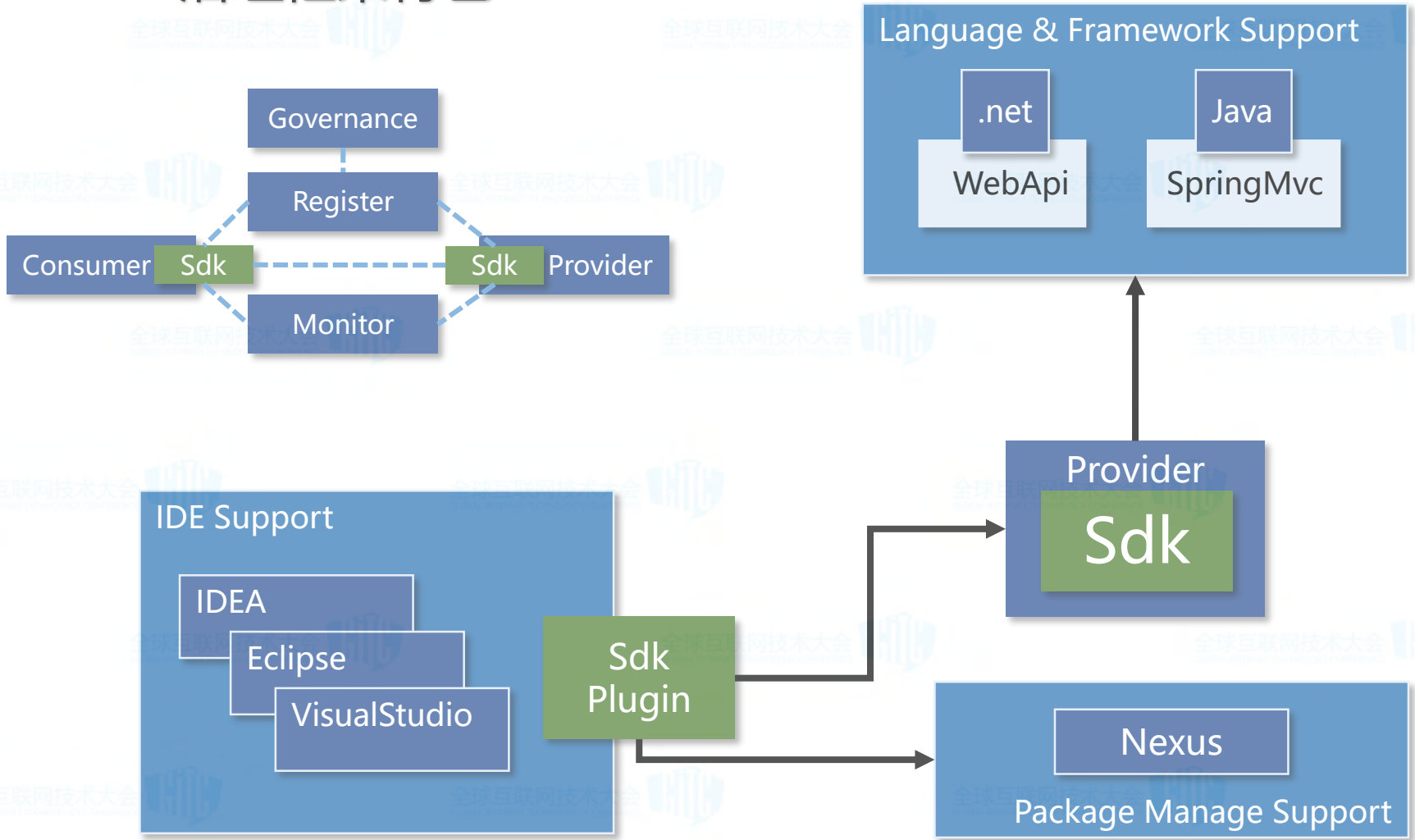
Kraken提供了很多主流IDE的治理插件来提升研发的效率，例如自动打包契约，自动生成接口描述，接口更行提示等。

## 全面的运行数据

每一个请求都会留下丰富的请求信息，用于各种维度的深度运行数据挖掘，例如链路健康检测，深度检测，接口实时拓扑等，让开发人员对自己的接口了如指掌。



## Kraken治理框架特色





### 3. 没有并发，需要管理软硬件资产关系吗



订单服务的  
IP是多少？



研发



运维



研发



运维



等我找找.....

—○	—○	—○	—○
—○	—○	—○	—○
—○	—○	—○	—○
—○	—○	—○	—○
—○	—○	—○	—○
—○	—○	—○	—○
—○	—○	—○	—○
—○	—○	—○	—○
—○	—○	—○	—○
—○	—○	—○	—○
—○	—○	—○	—○
—○	—○	—○	—○
—○	—○	—○	—○
—○	—○	—○	—○
—○	—○	—○	—○
—○	—○	—○	—○
—○	—○	—○	—○
—○	—○	—○	—○



### 常规的经验

- 没并发，集群节点数少，没必要急着管理软硬件资产关系。

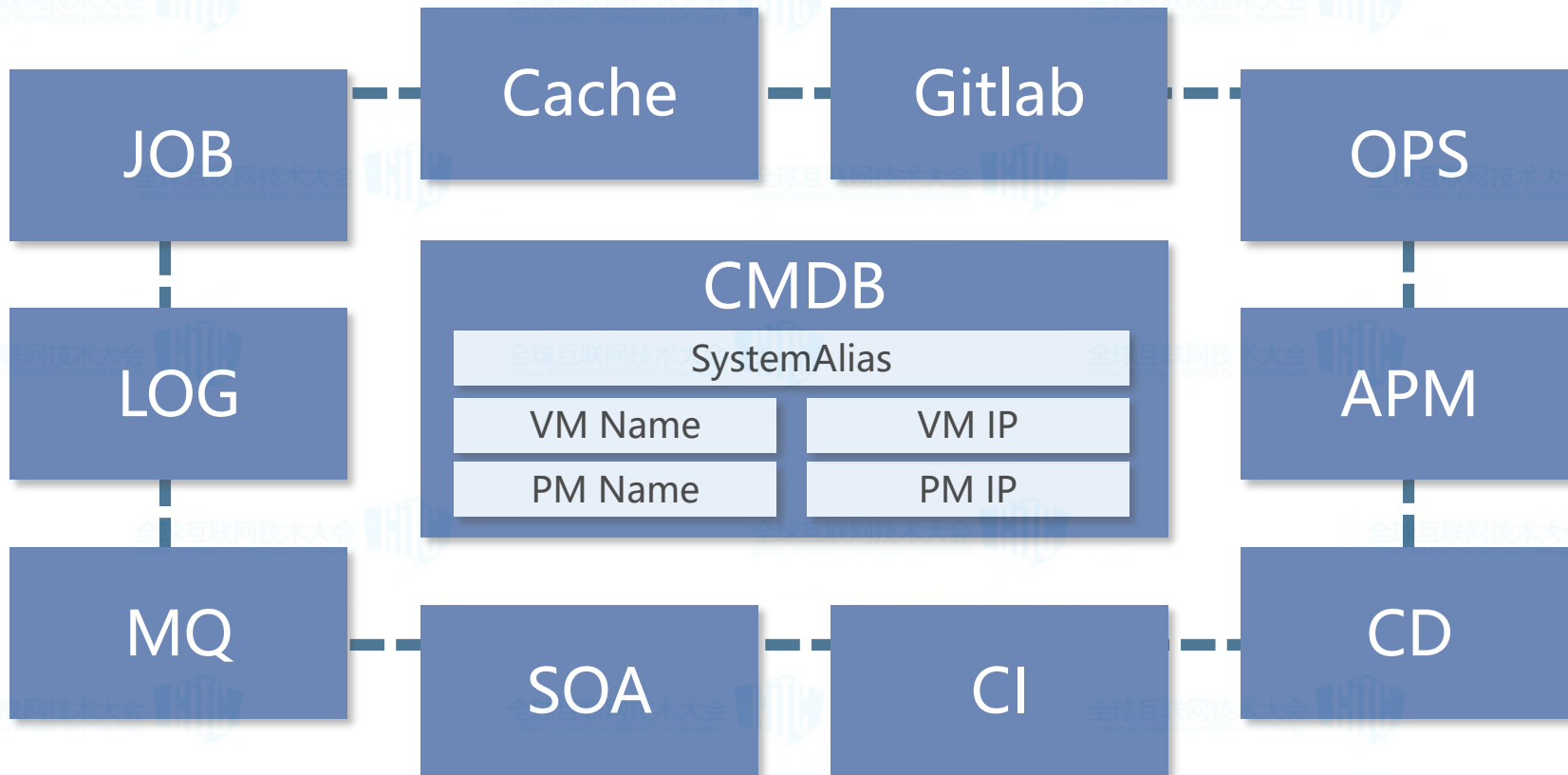


### 实际的情况

- 应用规模与没并发并不具备捆绑性质的参考，而是与实例数有捆绑参考。大宗产业电商的应用规模一点都不逊色于消费级电商，很少有人能对4个一组的IP有过目不忘的能力.....

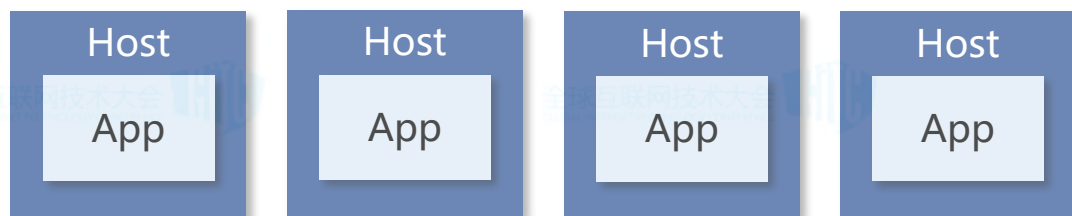
## SystemAlias规范 业务线.产品线.应用.应用类型

- arch.log.engine.service
- interest.ticket.manage.api





## 4. 没有并发，发布就交给手工行吗



排队的研发.....



### 常规的经验

- 没并发，交付时间对生产的影响不大，指个运维慢慢发就行。



### 实际的情况

- 参考应用规模的指标，即便没有并发，应用规模可一点都不小，如果没有自动部署的工作机制，一定会把负责部署的运维折磨到怀疑人生的.....



想发就发的持续集成交付平台

## 多环境支持

Zonic能够支持任意多的数据中心与集成环境，为不同规模的技术团队提供灵活便捷的多环境部署的能力。

## 丰富的代码质量控制能力

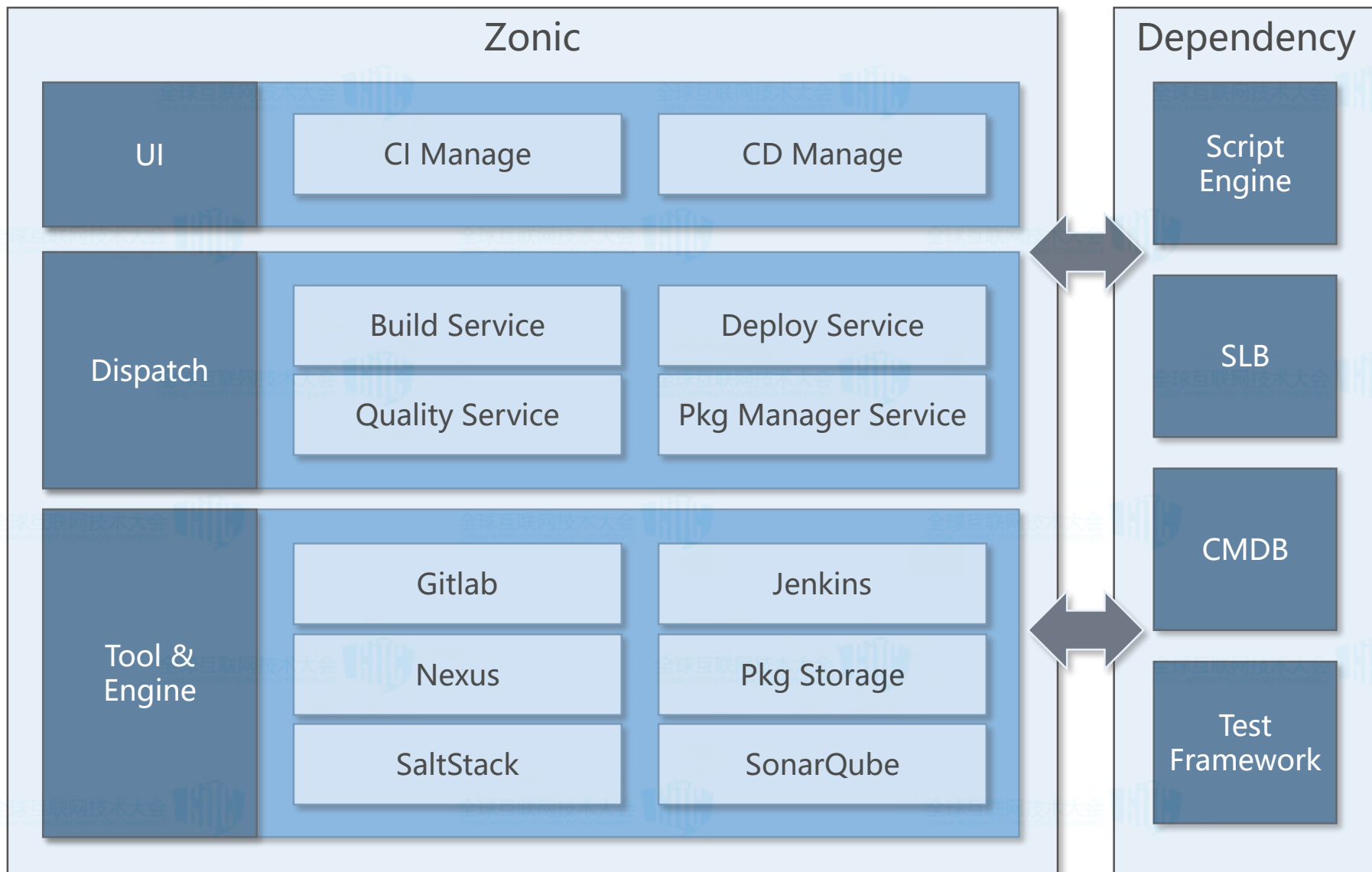
Zonic提供了单独的针对编译的支撑，将软件交付过程中代码质量的部分与部署进行了分离，确保任何系统的代码级问题，在编译阶段就能被发现。

## 丰富的交付策略

多样化的部署策略让不同规模的团队定制自己的部署策略，包含了灰度部署、回滚、点火等高级功能，让生产的部署平滑可靠。

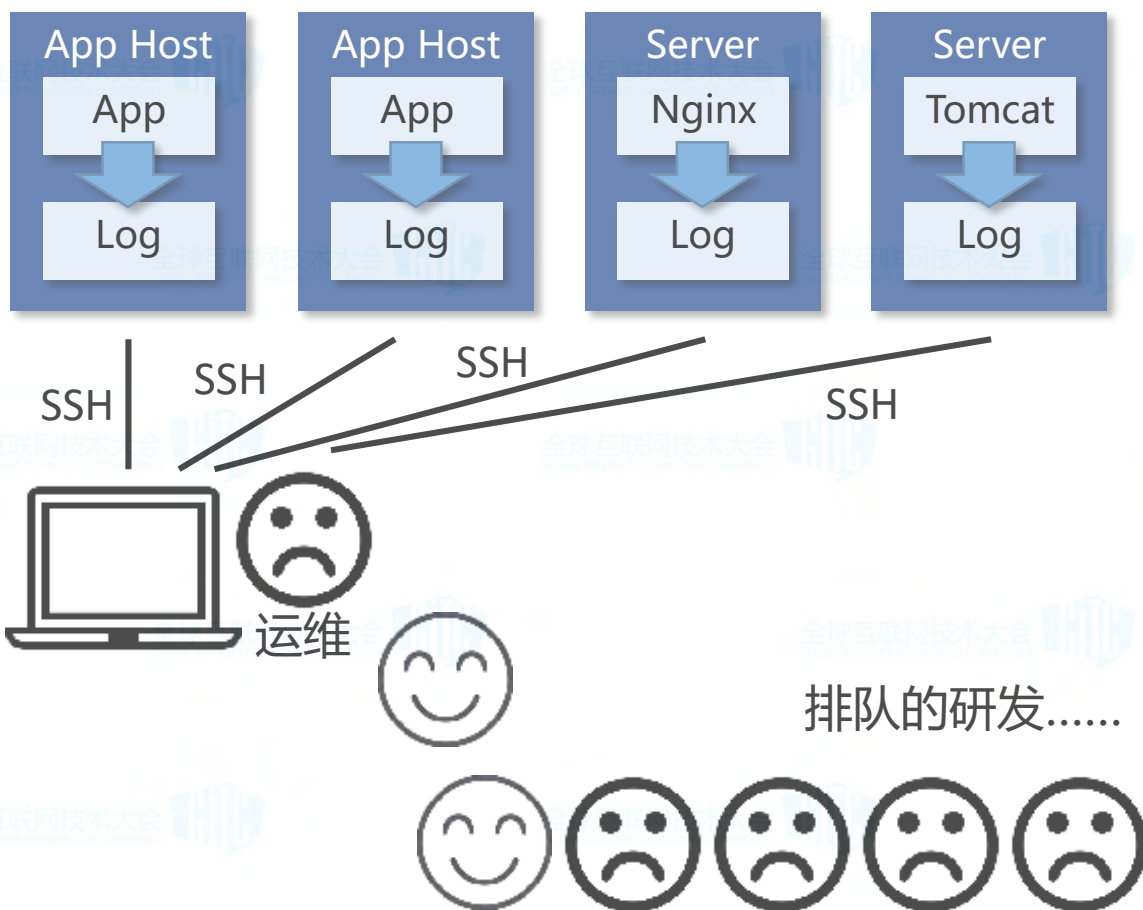
## 全面的交付分析

每一次部署都会留下详细的数据形成各种维度的报表，诸如编译成功率，部署成功率，利用这些数据可以全面的分析构建过程中的问题，提升下一次的效率。





## 5. 没有并发，日志就记在服务器上行吗



### 常规的经验

- 没并发，日志量就低，做好自动分割定期清理就行。



### 实际的情况

- 如果不整个专门用来存日志和查日志的玩意儿，烦死运维的绝对不是日志的存储和分割失败的问题，而是每天都要面对一群研发要求登服务器查日志，完全没法干正事儿.....



## 全设备全链路日志监控分析引擎

### 应用程序日志

森罗印象能接受任何语言的应用程序日志，提升应用程序运行时的调试效率和应用排障效率。

### 系统设备日志

森罗印象可以囊括交换机日志，操作系统日志，大大提升了运维和IT工程师对各类系统设备的维护和排查效率。

### 服务器日志

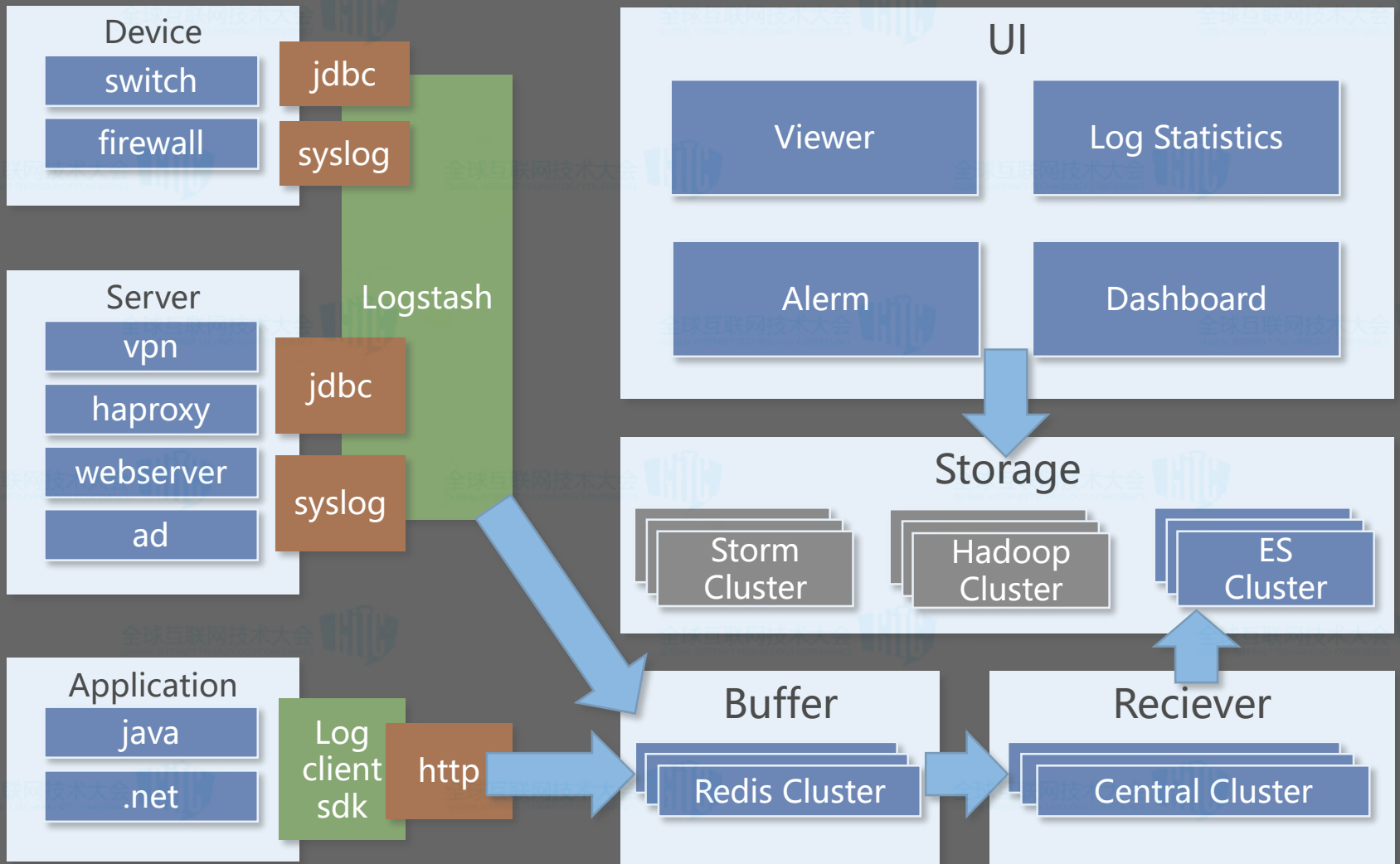
森罗印象能够记录各个环境的Web服务器日志，反向代理日志，Vpn日志，域日志，给予了全新的服务器日志查询方式，仅需打开浏览器输入条件即可得到期望的结果。

### 安全日志

森罗印象甚至可以容纳防火墙日志，通过实时分析每个请求，及时阻止任何心怀不轨的内外部网络访问，为系统的安全保驾护航。



# 森罗印象整体架构图







## DB驱动

- 曝光慢查询
- 消灭复杂存储过程
- 1库1应用

进程名	机器名	每次CPU耗时(S)	每次总耗时(S)#	执行总次数	SQL_ID	SQL文本
JDBC Thin Client	pbs_... ...lin_finance-basedata-platform-ui	4.036	32.972	1	3wck9gh8gnyz	select da unit_wel return_a
JDBC Thin Client	pbs_... ...lin_finance-basedata-platform-ui	1,686	27.768	1	6kmy628cb00fw	select da (p.added
JDBC Thin Client	pbs_... ...lin_finance-basedata-platform-ui	15.19	24,663	1	30j24mp7v9713	select t.p piece_we    last_r
JDBC Thin Client	pbs_... ...lin_finance-basedata-platform-ui	2.168	21,001	1	5jqbuf3v2126n	select PK LAST_MO
oracle@DB-ORACLE-ERP-01(j000),oracle@DB-ORACLE-ERP-01(j001)	DB-ORACLE-ERP-01	11.284	20,66	3	6kxp59fn8fc58	INSERT I SETTLEM ITEM_CC
JDBC Thin Client	pbs_... ...lin_finance-basedata-platform-ui	20.197	20.449	1	03vsp60f4tgj7	SELECT P DEPARTY ' '    PU

通过DB驱动，让研发在开发意识上进行提升，辅助应用架构师更好的对系统在数据层面进行高内聚低耦合的设计。

## 事故驱动

- 建立事故SLA
- 管理事故案例
- 学习事故经验
- 设立最佳实践

状态	业务线	描述	级别	发生时间	影响	操作
○	销售	10.14销售收款事故线上事故	A级	2015-10-14 10:09:00	1 day, 4:51:00	
○	财务	10/12 ERP用户收支款事故		2015-10-12 10:28:00	5:02:00	
○		11月13日消息系统故障		2015-11-13 15:40:00	1:05:00	
○		2/23ERP和CH系统对应备库的10.0.0.37出现数据延迟	C级	2016-02-23 00:00:00	13:00:00	
○	采购	2016/05/06 下午B1无法收款故障		2016-05-06 16:07:00	0:09:00	
○		3/21收款故障事故	A级	2016-03-21 09:00:00	2:30:00	
○	采购	4/14供应商基础刷数据事故		2016-04-15 10:00:00	3 days, 12:30:00	
○	销售	4/25 抢单2.0不可用故障	B级	2016-04-25 10:00:00	3:30:00	
○	采购	4/25联营可售库存开单报错故障	B级	2016-04-25 09:40:00	0:42:00	
○	财务	4/25财务无法收款事故		2016-04-25 13:50:00	3:40:00	
○	销售	4/6修改备注、价格服务不可用	C级	2016-04-06 13:03:00	0:32:00	
○	采购	4/6采购无法收款故障	C级	2016-04-06 09:30:00	1:30:00	
○	采购	5/16撮合解析上海冷链解析故障	C级	2016-05-16 09:22:00	0:44:00	
○	采购	5/20价格管理上下架故障	B级	2016-05-20 11:52:00	1:08:00	

通过事故驱动，让研发和运维，架构联系更紧密，从事故中学到代码之外的知识和技能，又通过代码之外的提升，重新审视如何去写代码。

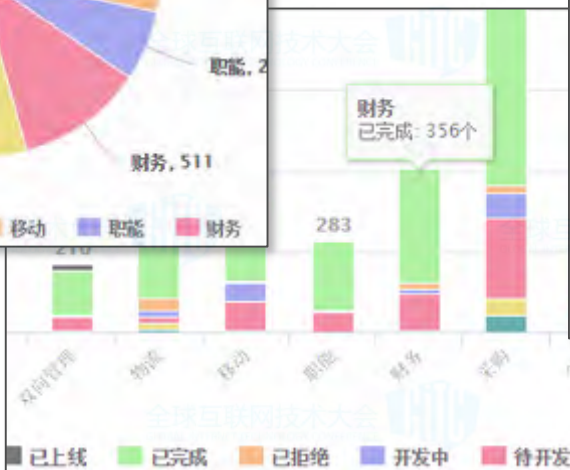
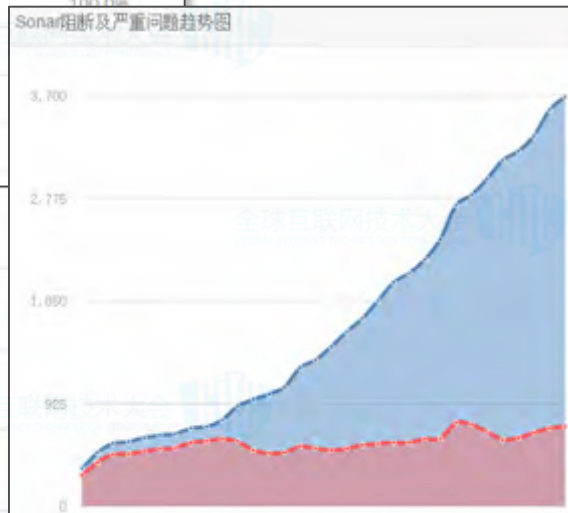
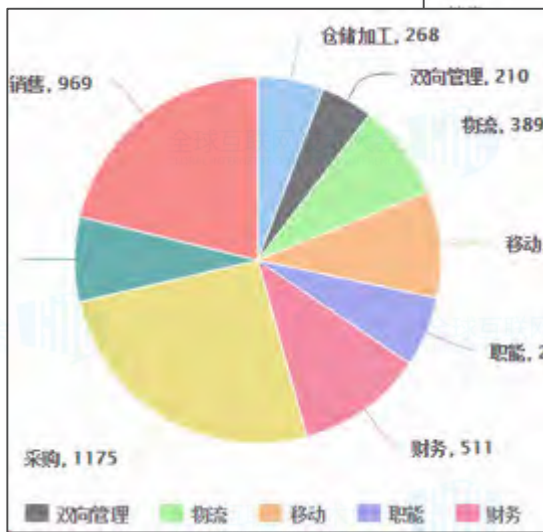


## 质量驱动

- 人效指标
- 代码指标
- 交付指标

### 应用发布成功率

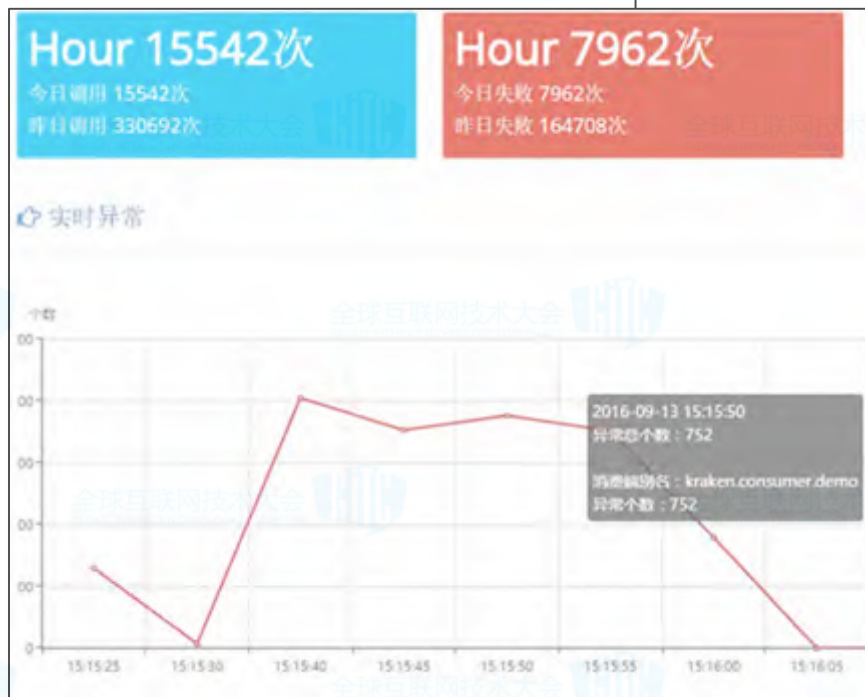
研发线	发布应用总数	发布应用失败数	发布成功率
财务	8	0	100.0%
物流	13	0	100.0%
仓储加工	13	0	100.0%
双向管理	6	0	100.0%
移动	6	0	100.0%
职能	5	0	100.0%
财务	17	0	100.0%
采购	6	0	100.0%



通过质量驱动，可以在软件生命周期的整个过程中提升软件各方面的质量，促进研发在敏捷方面的思考，同时让研发与运维共同走向Devops。

## 治理驱动

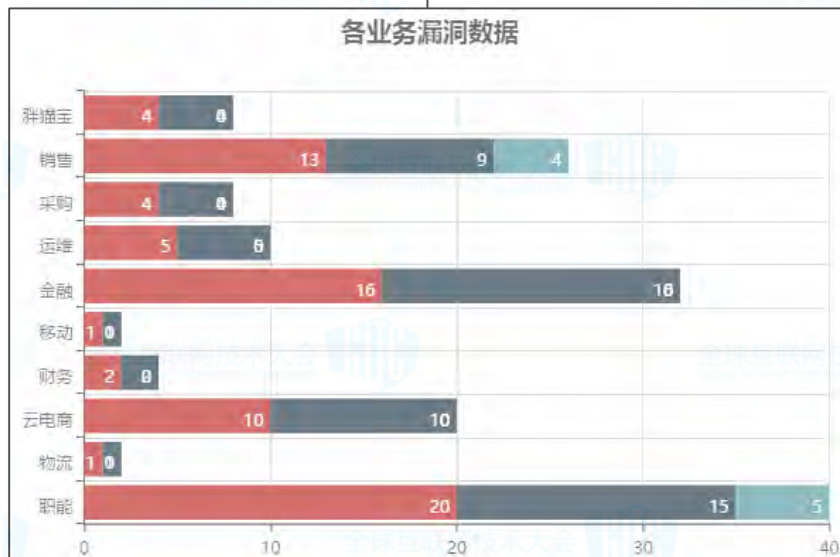
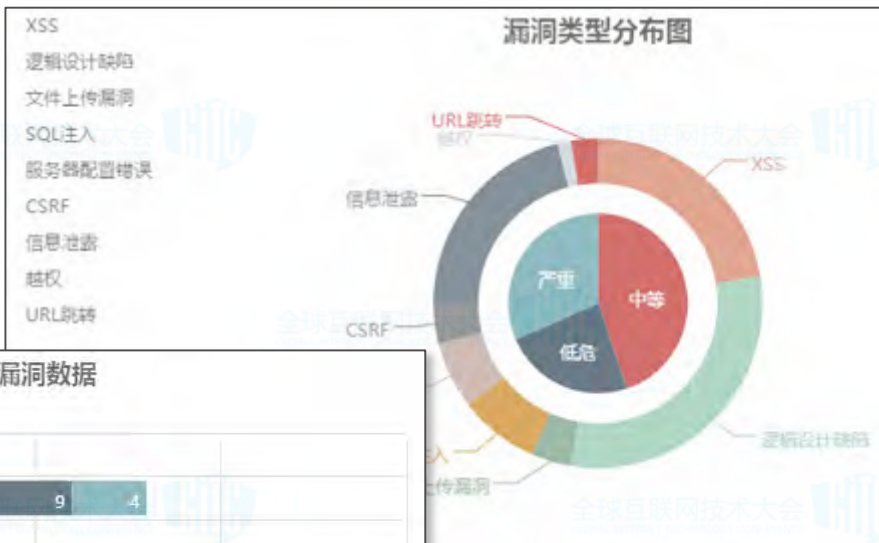
- 接口关系
- 接口深度
- 接口频次
- 接口健康



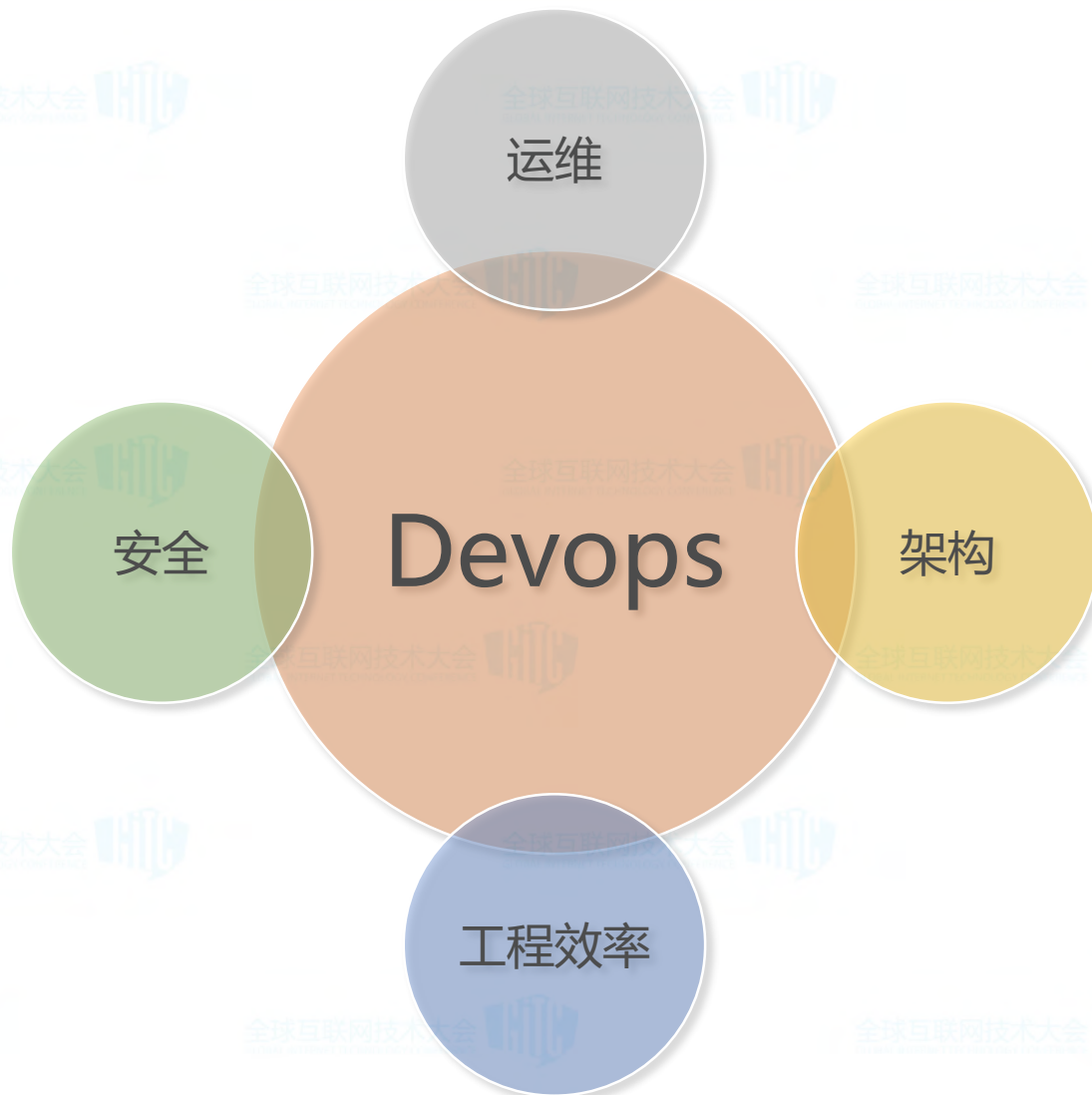
通过治理驱动，从服务治理中心的运行数据中得到各类分析统计结果，从系统耦合度，接口的粒度，调用合理性，等方面给予研发重要的参考依据，促进整个系统向着良性的方向演进。

## 安全驱动

- 动静态检测
- 设立漏洞SLA
- 安全宣讲
- 安全规范



通过安全驱动，从意识层面逐步提升研发对安全的认知。从执行层面尽早尽快的发现安全隐患，及时封堵安全漏洞。从协作方面，让安全工程师从研发的对立面转为积极面。







Thanks!

Q & A