

2018 | 中国·北京站
DevOps 落地，从这里开始

DevOps 国际峰会

暨 DevOps 金融峰会

指导单位： 云计算开源产业联盟
Open Source Cloud Alliance for Industry (OSCAI)

主办单位： DevOps时代

 高效运维社区
GreatOps Community

2018年6月29日-30日

地址：北京悠唐皇冠假日酒店

这次我们先不谈技术

金融IT系统的持续运维思考和实践

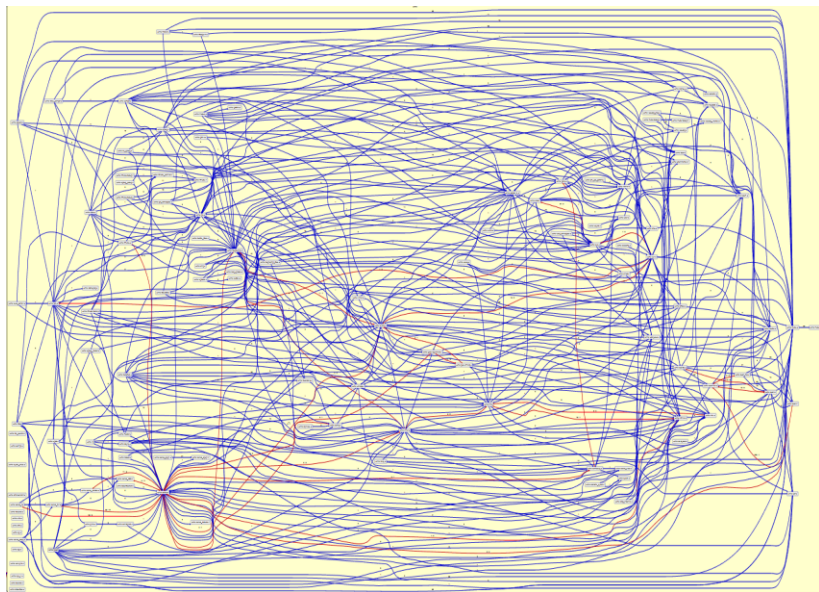
李卓 道富银行

目录

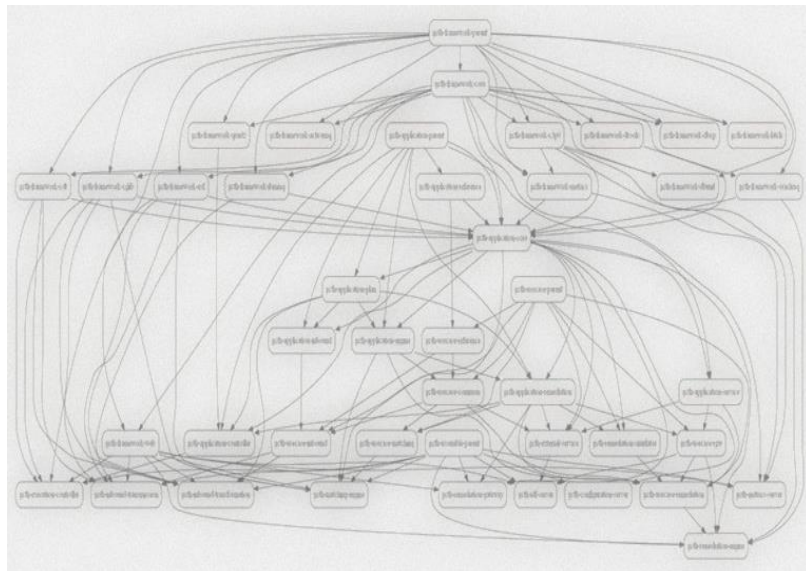
**1****背景****2****挑战****3****实践****4****总结**

论科技树该怎么爬

重构前



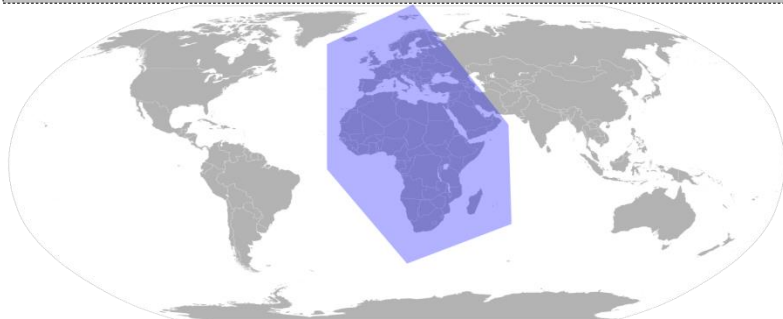
重构后



背景 – EMEA的业务系统运维

DOIS

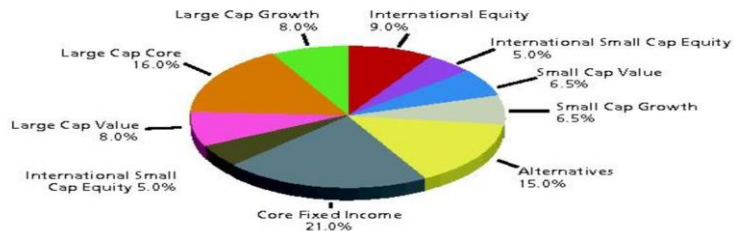
广泛的地理和时区分布



应用系统繁杂、人员不足



业务形态丰富、时间敏感度高



高安全和隐私保护要求、访问隔离



目录

1 背景

➔ 2 挑战

3 实践

4 总结

业务能力



Know What

Know Why

Know How

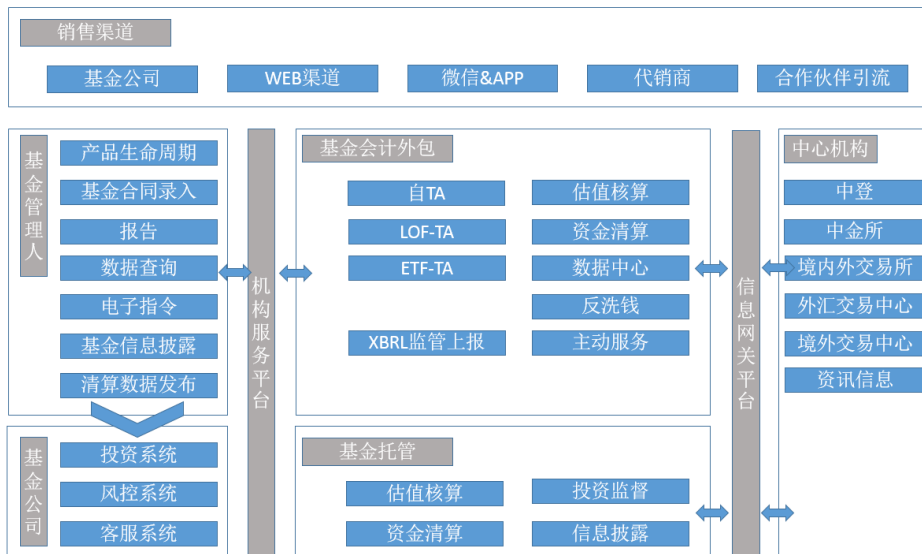
业务能力

- Know What

- 客户群

- 业务模型

- 用户特征



业务能力

- Know Why
 - 业务价值 (链)
 - 运维工作的价值



业务能力

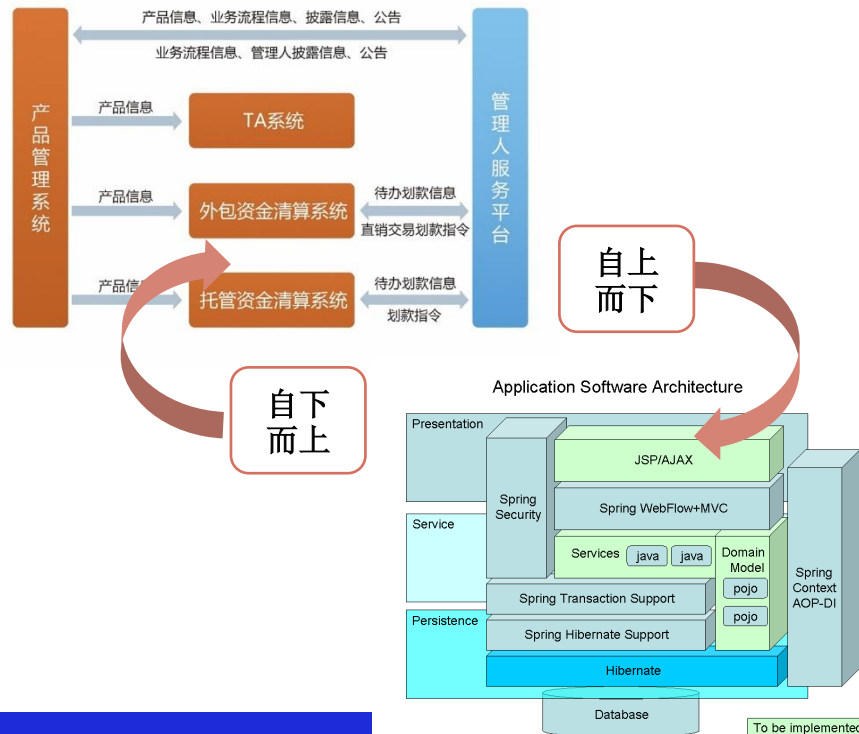
- Know How

- 系统分解

- 自上而下 (业务模型)
 - 自下而上 (系统模型)

- 目标定义

- 关键路径 (输入/处理/输出)
 - 关键时间窗口
 - 服务水平

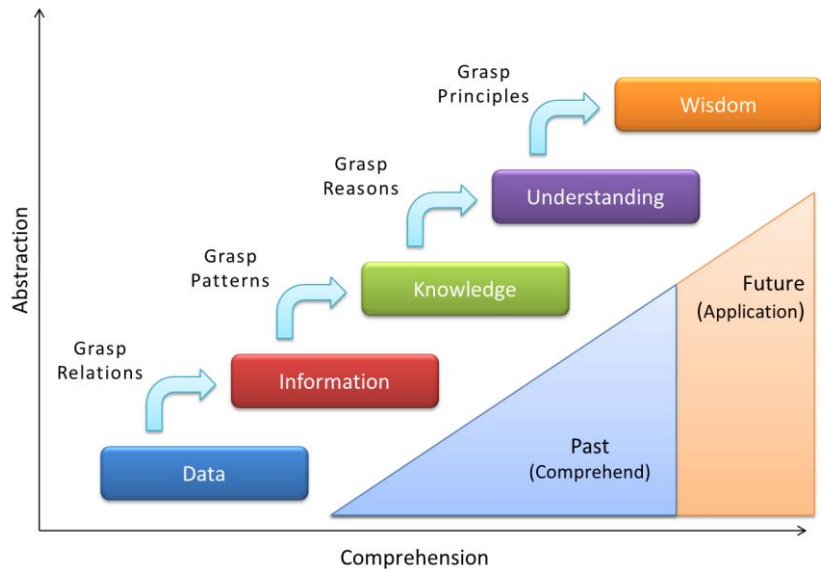


可持续



数据大爆炸

数据太多，信息太少



数据-信息-知识-理解-智慧

工具太多，合适就好



不要手里拿着锤子，就看什么都像钉子！

目录

1 背景

2 挑战

→ 3 实践

4 总结

理解问题

- What – 举个栗子

客户/用户

- 机构客户
- 内部运营用户

业务模型

- 基金托管
- 基金管理
与附加值
服务

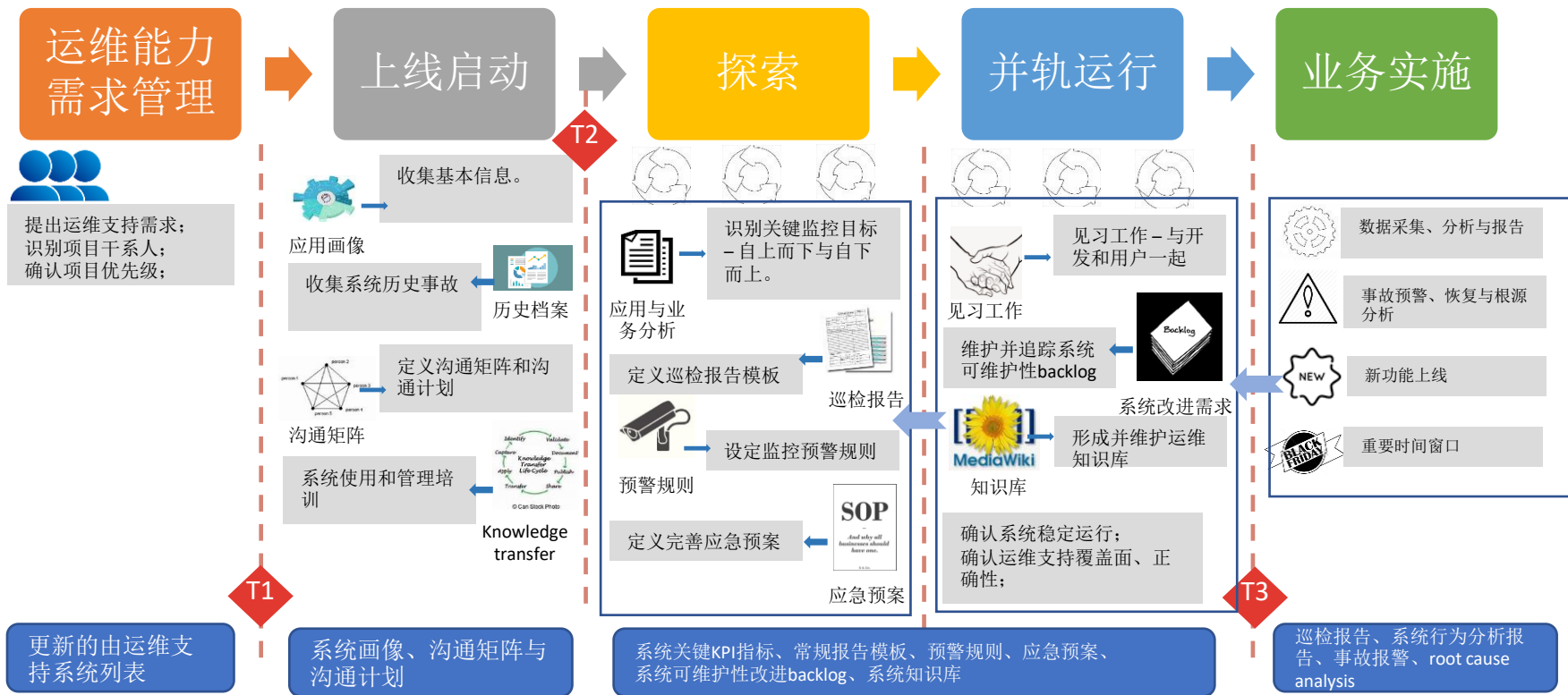
系统

- 数据核算
- 数据计算

理解问题

- Why – 举个栗子
 - 专业能力
 - 质量、效率
 - 规模经济

业务系统运维能力上线流程



可持续性

时间和空间流转

- 三地7*24时区和地理覆盖
 - 中国
 - 波兰
 - 印度
- 沟通矩阵
 - 责任人定位
 - 响应升级

多角色多团队流转

- 流程一致
 - 事故报警、处理、升级流程
 - 服务水平要求
- 工具集一致
 - 工作流工具
 - 监控工具
- 认知一致
 - 知识库
 - 应急预案

可持续性 - 系统开发生命周期



可持续性 – 业务运营全流程



| 核心 | 阶段 | 关注点 | 有效性评判 |
|---------------------|------|-------------|---|
| 保证 业务 延续 性 | 事前预防 | 常规巡检报告 | <ul style="list-style-type: none">• 系统可用性• 应急预案覆盖率• 巡检报告覆盖率• 事故排除有效性 |
| | | 趋势发现 | |
| | | 应急预案定义与演练 | |
| | 事中紧跟 | 及时预警 | <ul style="list-style-type: none">• 响应时间• 恢复时间• 用户满意度 |
| | | 保持接触 | |
| | | 响应与升级 | |
| | 事后回顾 | 业务保全与恢复行动落实 | <ul style="list-style-type: none">• 事故复发率• 系统可用性• 用户满意度 |
| | | 影响评估 | |
| | | 根源分析 | |
| | | 改进行动落实 | |

可持续性 – 业务运营全流程

- 要事第一 - hyper-care模式
 - 重要业务时间窗口
 - 重大功能、升级上线
- 方法
 - 主动自查
 - 同步支持
 - 支持热线



可持续性 – 人员流动



- Team WIKI

- 知识外化

- 流程

- 系统

- 业务

- 历史

- 知识共享

The screenshot shows the MediaWiki page titled "How to become a MediaWiki hacker". The page is in English and includes a sidebar with navigation links such as "Main page", "Get MediaWiki", "Get extensions", "Tech blog", "Contribute", "Support", "User help", "FAQ", "Technical manual", "Support desk", "Communication", "Development", "Bug tracker", "Code repository", "Code docs", "Statistics", "Wikimedia technology", "Wikimedia audiences", "MediaWiki.org", "Community portal", "Recent changes", "Translate content", "Random page", "Current issues", " Sandbox", "Print/export", "Create a book", "Download as PDF", "Printable version", and "Tools".

The main content area starts with a "Page" tab and a "Discussion" tab. Below the tabs is a search bar and a "Read" button. The title "How to become a MediaWiki hacker" is prominently displayed. Below the title is a "Translate this page" button and a list of "Other languages".

The text of the page includes:

This article is written to help developers learn the basic skills needed to contribute to **development of MediaWiki core and MediaWiki extensions**. Note that in most cases when working with MediaWiki, you do not want to hack MediaWiki core unless you really know what you're doing.

The main path to get started with Wikimedia development is to **contribute to Wikimedia projects that offer mentoring**. An alternative without mentoring is to fix an annoying little bug.

If you are an experienced developer who is familiar with using MediaWiki already, visit the **Developer hub** instead.

For other ways to get involved in the Wikimedia community, see **How to contribute**.

Overview

MediaWiki is the software that powers Wikipedia, its sister projects and thousands of wikis all over the world.

MediaWiki is written in the **PHP** programming language^[1]. It uses **jQuery** as the client JavaScript library.

MediaWiki is primarily written for the **LAMP platform**^[2] and runs on most operating systems. MediaWiki primarily uses the **MySQL** and **MariaDB** database servers.^[3]

Development happens in an **open source style**^[4], is largely coordinated online, and supported by the **Wikimedia Foundation**, though volunteer community developers play a huge part as well.

- **Development discussion** happens on various **mailing lists** and **IRC channels**. The main developer list is **wikitech-l**. The **#mediawiki** and **#wikimedia-dev**.
- **Source code** is managed using the **Git** revision control system.^[5]
- **Code review** is performed on **Gerrit**. Follow **this tutorial** to set up **Git** and **Gerrit** in order to submit patches.
- **Bug reports** and tasks are managed on **Phabricator**.

This page should help you become a contributor to MediaWiki. It is **not** a tutorial; it just points you to various places where necessary.

Set up your development environment

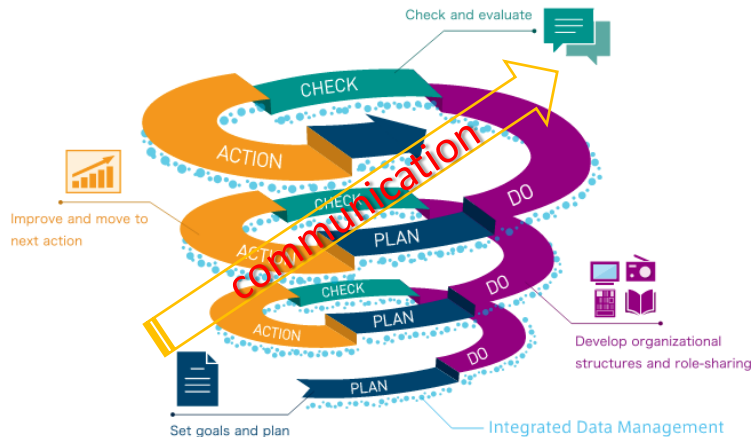
Most projects use **Git** and **Gerrit**. Follow the **Gerrit tutorial** to set up your developer account. Then you can move on to downloading our code, making changes, testing them, and submitting patches. There are two ways to set up your development environment: using a pre-configured virtual machine setup (**vagrant**), or a manual configuration approach.

The right sidebar contains a "Contents" section with a list of links: 1 Overview, 2 Set up your development environment, 2.1 Virtual Machine with Vagrant, 2.2 Manual Installation, 3 Suggested reading, 3.1 Feedback, questions and support, 3.2 Communicate that you work on a task, 4 Appendix, 4.1 PHP, 4.2 Database, 4.3 JavaScript and CSS, 4.4 MediaWiki, 4.5 MediaWiki extensions, 5 See also.

At the bottom right, there is a speedometer showing "60.2KB/s".

可持续性 – 形成持续改进反馈闭环

- 制定有效的沟通计划
 - stakeholder
 - 沟通渠道
 - 沟通载体
 - 沟通频率
- 保持稳定单纯的沟通窗口

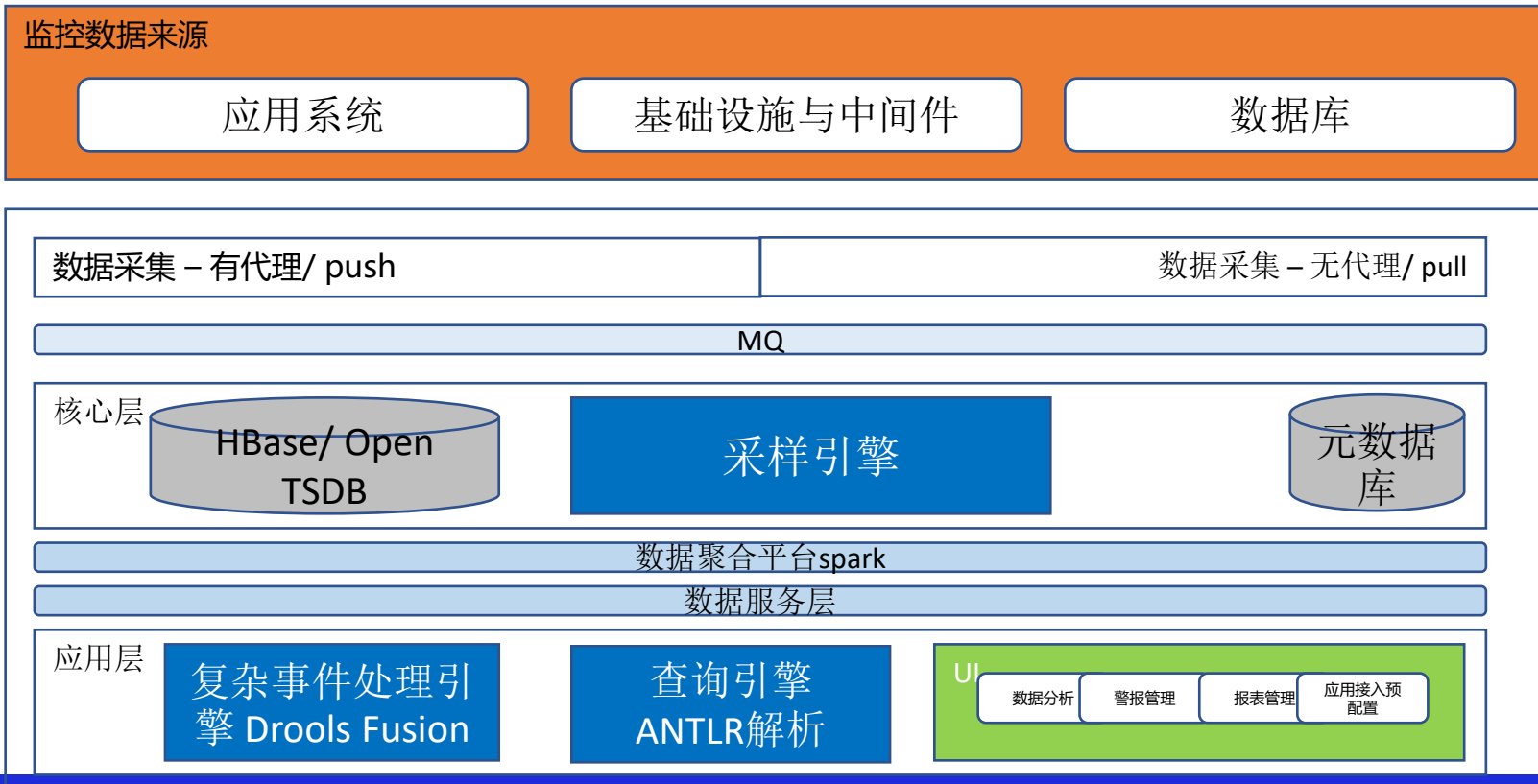


可持续性 – 有效沟通的原则

- Talk about facts, not opinion
- 追求双赢方案
- 沟通升级原则



还是要谈点技术



金融IT系统运维 - 监控系统设计



- 无侵入或少侵入
- 做自服务平台
- 数据到信息才是财富
- 隔离对产品环境的影响
- 不盲目追求最新技术

目录

1 背景

2 挑战

3 实践

➔ 4 总结

总结的总结



企业除了业务之外，没有别的问题



Thanks

DevOps 时代社区 荣誉出品

想第一时间看到高效运维社区的
最新动态吗?

