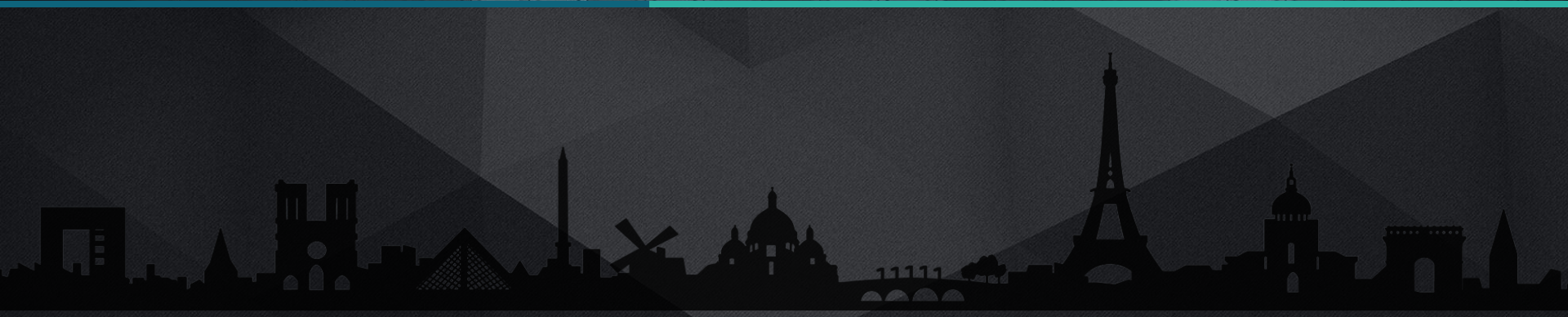
神州数码
Digital China

沪江技术学院

IT大咖说
知识分享平台

# 如何支撑微服务架构落地？



# 背景



与纯粹互联网公司不同的是，神州数码智慧城市集团每年需要开发、集成和交付很多的大型系统。如此，则更看重稳定的架构、知识的沉淀、交付的效率和快速适应变化的能力。这也是我们选择微服务架构的主要原因。

## 微服务架构

Microservice

微服务 ≈ 模块化开发 + 分布式计算

把因相同原因而变化的功能聚合到一起，而把因不同原因而变化的功能分离开，并利用轻量化机制（通常为HTTP API）实现通信。

## 微服务架构

Microservice

1，通过将系统拆小，减小系统的复杂度
2，每个模块只需要关心自身需要的特性，例如性能、稳定性

此处省略5万字

# 为什么不使用依赖库，而使用微服务架构？



- Bug修复后如何让库使用者升级新版本？
- 库使用者需要使用新功能，新旧版本兼容问题如何解决？
- 是否担心依赖库的bug导致业务系统不稳定？
- 依赖库与业务系统是共享数据库的，如何避免业务逻辑的耦合？
- 使用依赖库如何实现非功能需求？
- 如何以库的形式共享前端页面？
- 依赖库以Java开发，.Net / PHP团队如何复用？
- ……

# 微服务看起来很好，有没有给团队带来麻烦？

沪江技术学院 IT大咖说 知识分享平台

- 上手难度加大，怎么办？
- 需部署的包突然变多，如何运维？
- 业务模块如何拆分？
- 多个小服务如何组装成大系统？
- 微服务间的依赖关系错综复杂，如何管理？

开发觉得微服务是个好架构，可是运维不这么认为。。。

如果没有完善的支撑体系，就不要采用微服务

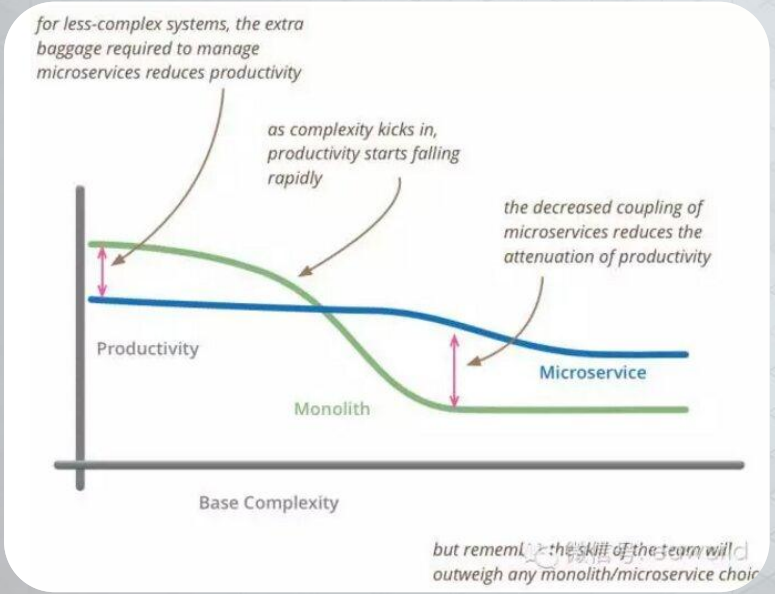
# 如何缓解微服务架构带来的弊端？



- 能力支撑
  - ✓ 提供通用的微服务组件（服务注册、服务发现、服务网关、用户、权限、配置、通知、积分、支付等）
  - ✓ 完善的运维平台（发布、升级、回滚、运行、日志）
  - ✓ 完善的监控系统（健康巡检、性能监控、业务指标监控）
  - ✓ 平台研发团队为业务研发团队提供完备的工具和技术支撑
- 自动化
  - ✓ 一键发布单个微服务（持续集成）
  - ✓ 一键发布整个系统（服务编排、数据库初始化、配置初始化）

# 什么样的系统需要采用微服务架构？

- **规模大**（人员多、代码行数多）
- **复杂度高**（业务复杂）
- **需要长期演进**（比如长期项目、软件产品）



# 采用微服务架构后，如何以正确的姿势做技术管

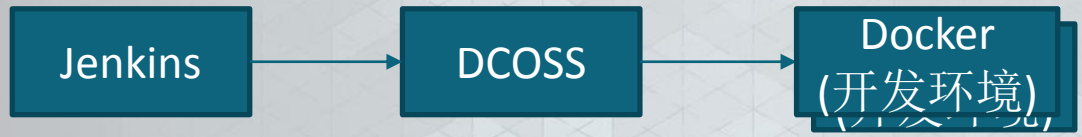
沪江技术学院 IT大咖说 知识分享平台

- 沟通方式（从数据共享到API调用）
- 职责划分

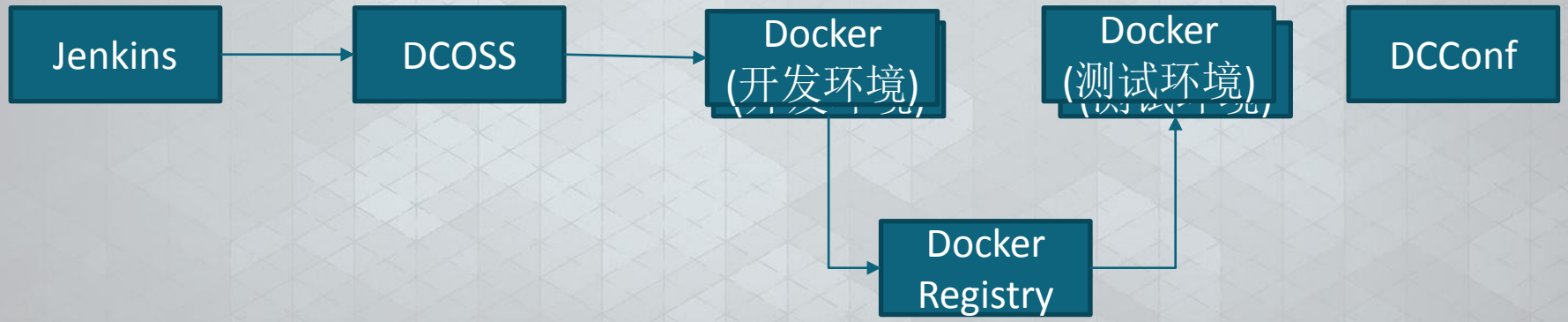
  You build it, you run it. --亚马逊CTO Werner Vogels
- 团队组织形式（自组织）
- 风险控制（避免全局性风险）
- 知识积累（以完整微服务的形式积累）

亚马逊创始人Jeff Bezos（2002年）：所有团队的模块都要以Service Interface的方式将数据和功能开放出来。不这样做的人会被炒鱿鱼。

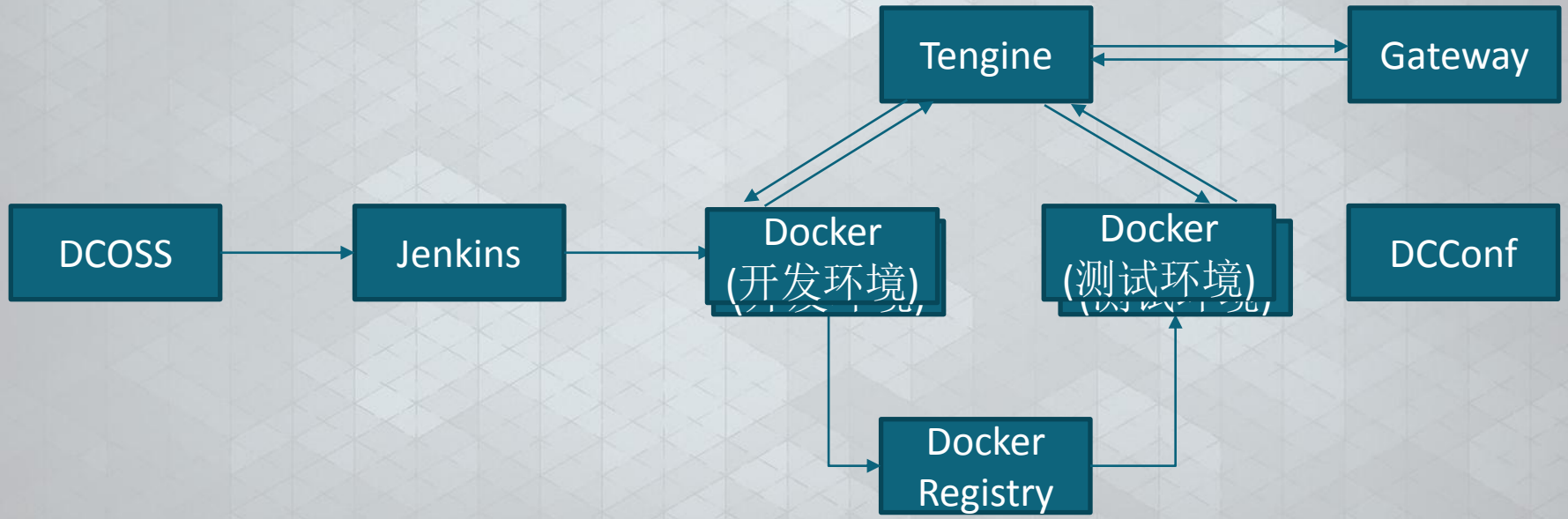
- 持续部署（Docker + Jenkins）



- 环境迁移（Docker Registry + DCConf）



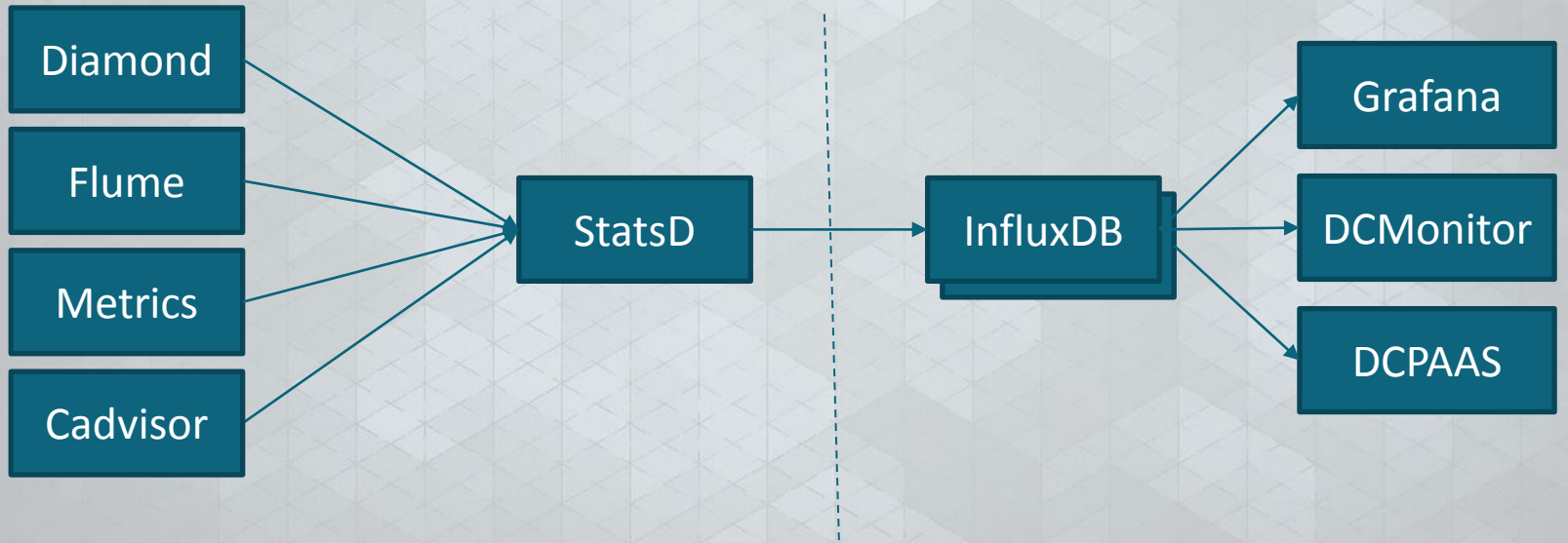
- API安全（公网）（Tengine + DCGateway）



# 我们采用的微服务架构技术



- 微服务监控（Grafana + InfluxDB + Diamond + Flume + Cadvisor + StatsD + DCMonitor）



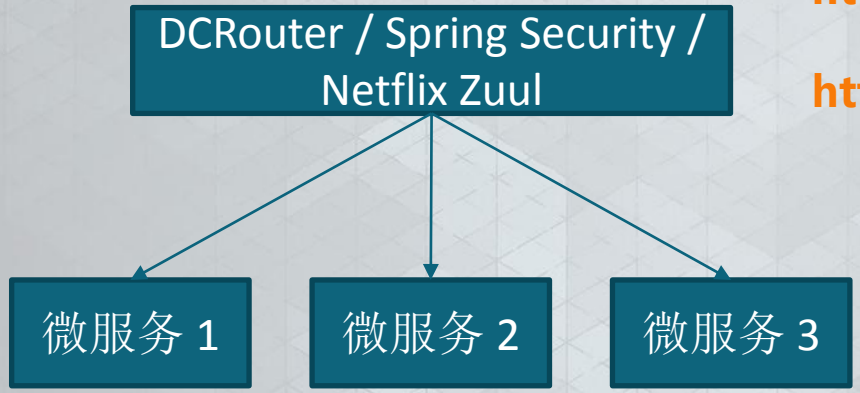
- API的管理及测试（Swagger，Rest Assured）


# 我们采用的微服务架构技术



- API调用（Eureka + Ribbon + RestTemplate + DCTrace）

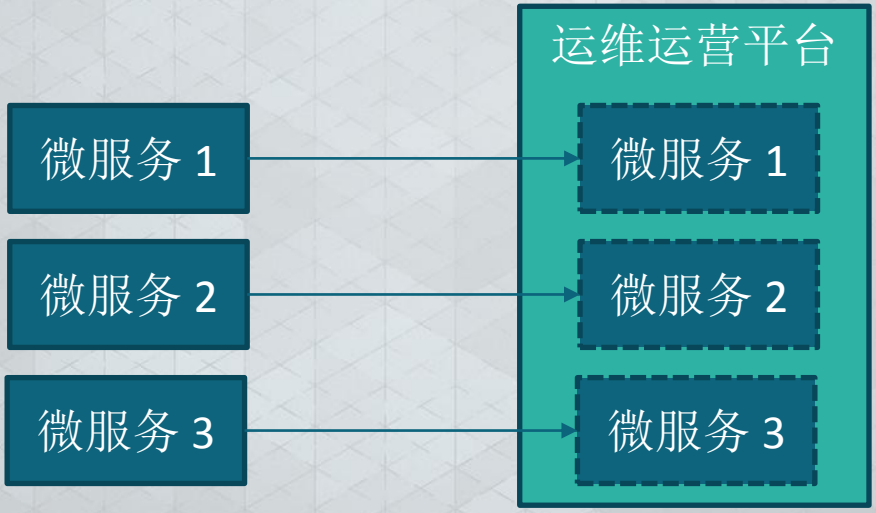

- API安全（内部穿透）（Spring Security + DCRouter）




http://www.xxx.com/ro/srv1/hello.do

http://srv1/hello.do

- 微服务整合（DCUPMS）



# 总结



微服务架构是技术升级，但更多的是管理模式的升级。

神州数码
Digital China

沪江技术学院

IT大咖说
知识分享平台

# 谢谢