

GhostCloud 精灵云科技

Build the world & Clone the world

微服务基础与案例分析

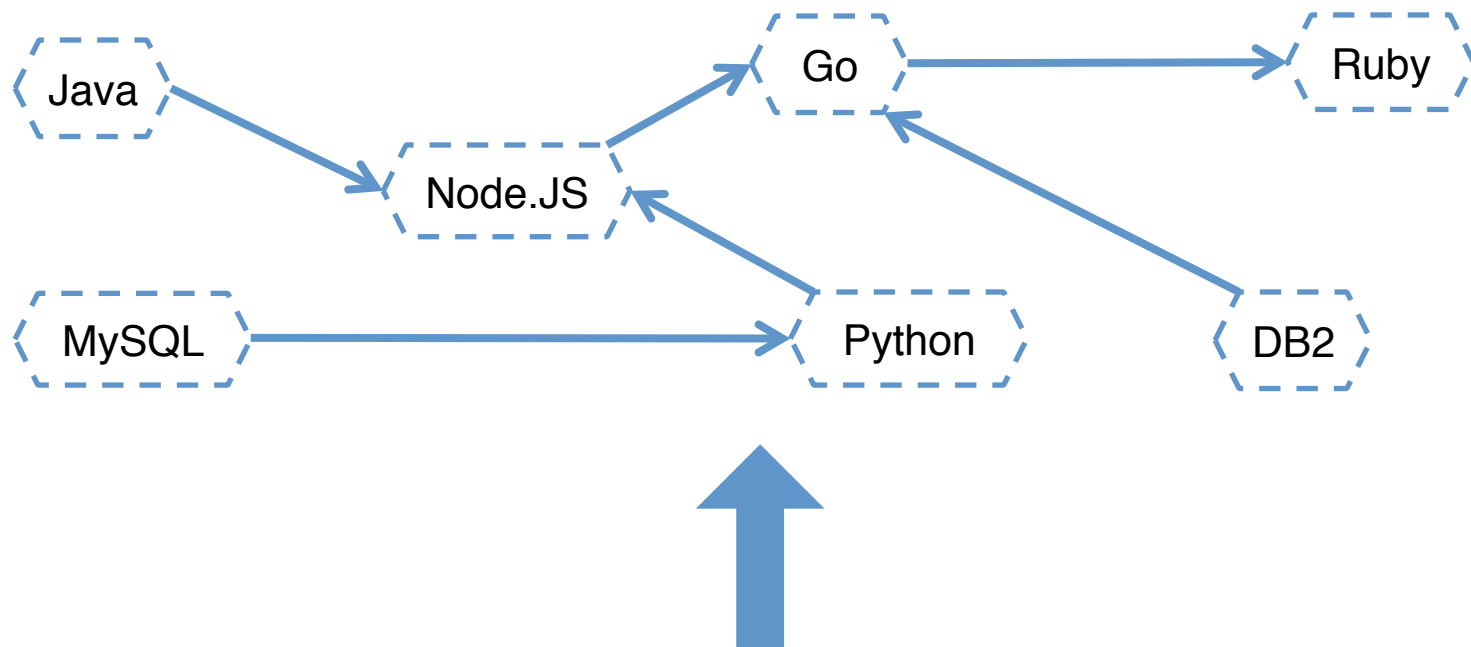
精灵云 马小平



团结 创新 实干 工匠精神

成都精灵云科技有限公司 (c) 版权所有

GhostCloud



微服务架构

移动互联网：构建灵活，扩展性，高可用性、容错

敏捷开发思想：CI/CD, DevOps, Full Stack

容器化技术：Docker, CoreOS

云计算技术：IaaS/PaaS/SaaS, 公有云, 混合云



优点:

- 易于开发、集中式管理
- 没有分布式调度和管理开销

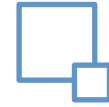
缺点:

- 开发效率低
- 测试成本高
- 维护成本高
- 交互周期长
- 扩展性不够
- 稳定性不高



The microservice architectural style is an approach to developing a single application as a suite of small services, each running in its own process and communicating with lightweight mechanisms, often an HTTP resource API. These services are built around business capabilities and independently deployable by fully automated deployment machinery. There is a bare minimum of centralized management of these services, which may be written in different programming languages and use different data storage technologies.

By James Lewis and Martin Fowler



- Componentization via Services
- Organized around Business Capabilities
- Products not Projects
- Smart endpoints and dumb pipes
- Infrastructure Automation
- Design for failure
- Evolutionary Design
- Decentralized



优点：

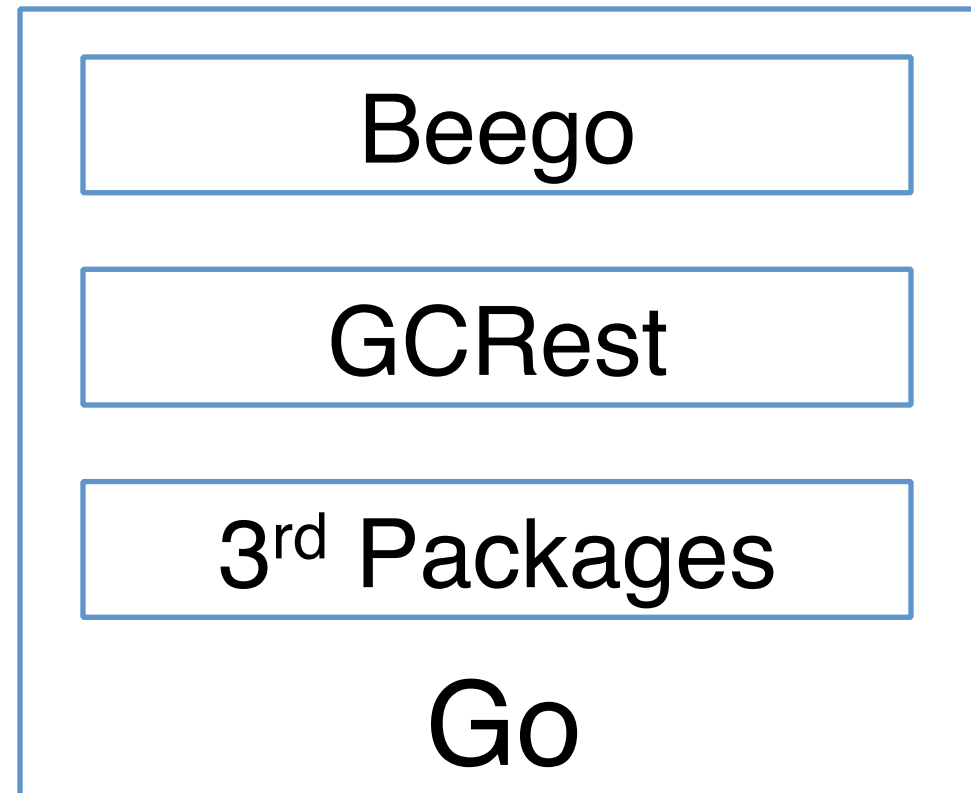
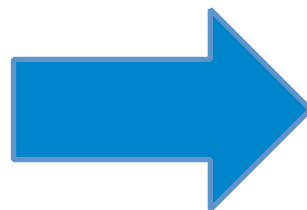
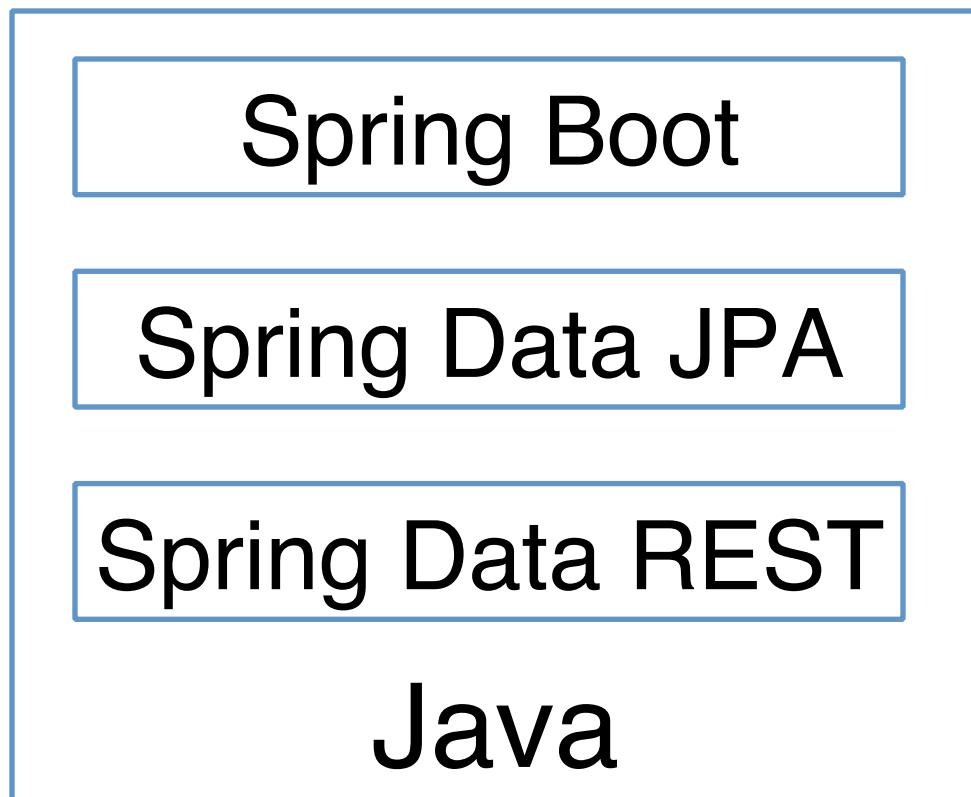
- 开发简单
- 模块独立，低耦合
- 迭代速度快
- 测试方便
- 技术选择多
- 高可用性
- 扩展性好

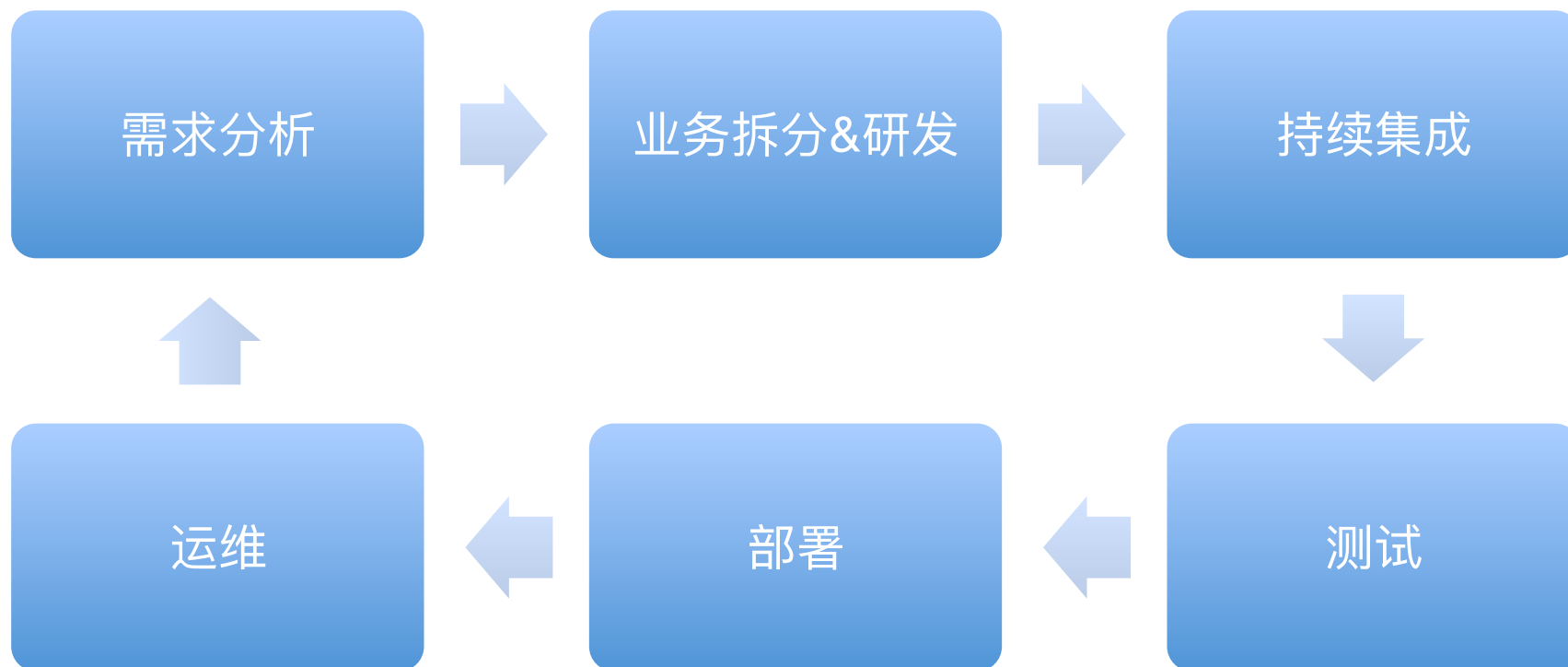
缺点：

- 设计要求高
- 系统复杂度增加
- 团队要求增加
- 运维成本增加
- 数据一致性
- 性能监控



- 敏捷思想
- IaaS/PaaS/CaaS
- 多持久化途径
- 分布式系统经验
- 多语言
- 虚拟化技术
- 沟通能力
- 终身学习
- 影响力







- 准备工作：
 - ms-tools工具包
 - 开发运行环境, builder镜像
 - 静态检查、测试、编译脚本
- 具体流程：
 - 生成builder镜像: 开发运行环境
 - Pull代码
 - 结合builder, 执行脚本: 执行静态检查、测试以及 编译生成bin文件
 - 环境配置: `docker-compose.yml`
 - 打包: `dockerfile`, `docker-compose`, 选择环境并运行
 - 后向测试: `docker exec container_id /test`
 - 自动化部署





- **基准代码**：一份基准代码，多份部署
- **依赖**：显示申明依赖关系
- **配置**：在环境中存储配置
- **后端服务**：把后端服务当成附加资源
- **构建、发布和运行**：严格分离构建和运行
- **进程**：以一个或多个无状态的进程运行应用
- **端口绑定**：通过端口绑定提供服务
- **并发**：通过进程模型进行扩展
- **易处理**：快速启动和优雅终止可以最大化健壮性
- **开发环境和线上环境等价**：尽可能保持开发，预发布，线上环境相同
- **日志**：把日志当着事件流
- **管理进程**：后台管理任务当作一次性进程运行



MODE	ACCESS	ENV
Bridge with HostIP and Port	Cluster Level	HostIP + HostPort
Bridge with Container IP	Host Level	ContainerIP
Bridge with Link	Host Level	Container Name
Host with Port	Cluster Level	HostIP + HostPort
Routed IP	Cluster Level	GlobalIP + GlobalPort
Overlay with link	Cluster Level	Container Name

