不连健康云原生微服务落地实践

邓志豪

禾连健康 CTO

目录



- 1 禾连简介&为什么要使用微服务
- 2 健康业务总体架构
- 3 容器化实践
- 4 Nacos/Dubbo实践
- 5 可用性保障

不连健康简介



成立于2014年,是一家从体检场景切入的健康管理服务公司。

使命:帮助老百姓做对体检,管好健康,少生病,不怕生病。

检前、检中、检后 一体化SaaS和服务

帮助医院实现 互联网健管中心 全国公立医院 体检和健康管理全案 家庭健康管理服务平台

医院

企业

用户

为什么要使用微服务

医院场景特点:高复杂度

低频/低流量 QPS<20

上班时段高可用 7:00-10:00-17:00-23:00

Bug容忍度低

医院内异构IT打通

医学领域复杂逻辑



Bug的延迟暴露 灰度发布失去了意义、压测?

非医院业务对医院业务的影响

Bug变医闹

我们出了问题还是医院IT出了问题

加项能有多复杂

亟需解决:技术风险、研发效率

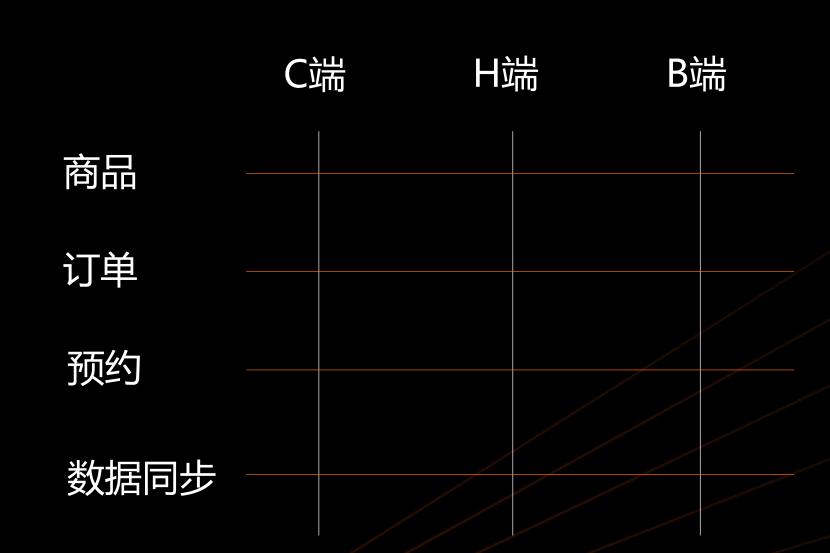
为什么要使用微服务

- 模块化 解决业务复杂度问题

- 模块 -> 服务
- 各端业务解耦,避免互相影响
- 独立的业务逻辑、数据状态自检
- 独立的急救工具
- 低流量下的成本考虑资源成本和维护成本

服务拆分:

业务拆分 vs 能力拆分



为什么要使用微服务

运维角色对决策的影响

运维开发人少全能

工具搭建、定制、维护

- DevSecOps/CICD
- 各类环境
- 监控报警

安全工作

成本优化

开发运维 日常运维落到开发身上

发布执行

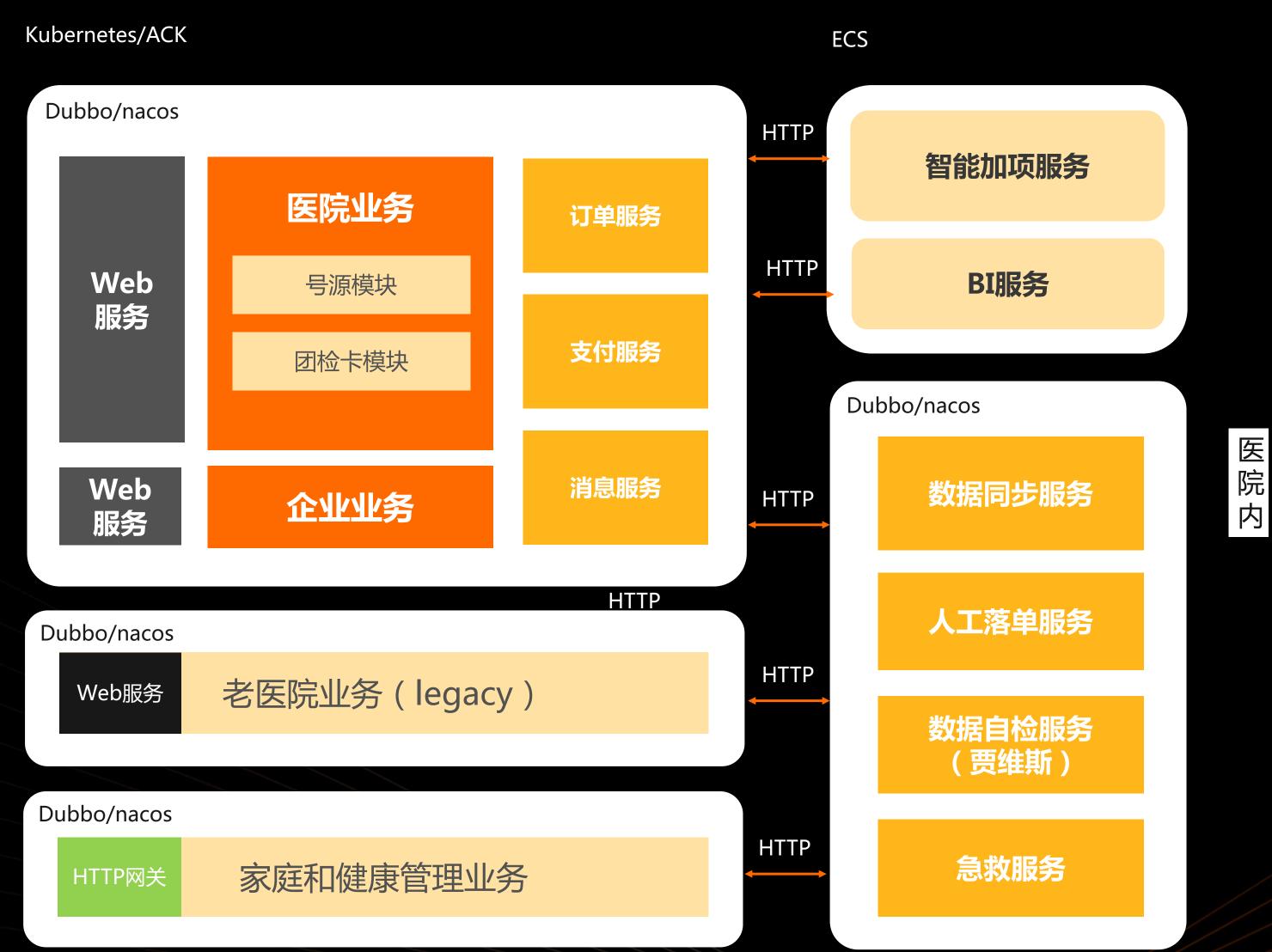
监控报警处理

部署架构调整



简单傻瓜的工具和云服务

健康业务总体架构



以能力为主的服务切分

容器化实践

Serverless vs Kubernetes

- Serverless成本问题 适合资源稳定的业务
 - 容器类云服务的定价
 - 对资源的弹性使用
- 稳定性:想保留一部分运维控制力 被搞过一次
- 开放性:云原生的各类方案

生产环境:ACK

测试环境:Rancher自搭建、Helm

开发环境:探索把本地服务挂上去的方案

Nacos/Dubbo实践

Nacos/Dubbo的特点

- 方便易用
- 自闭环,云原生时代前产品
- 活跃迭代
- 中文社区



碰到的坑

- Dubbo2服务模型带来的流量问题
- 间接导致nacos的运维问题



上云, 走MSE

Nacos/Dubbo实践

实践

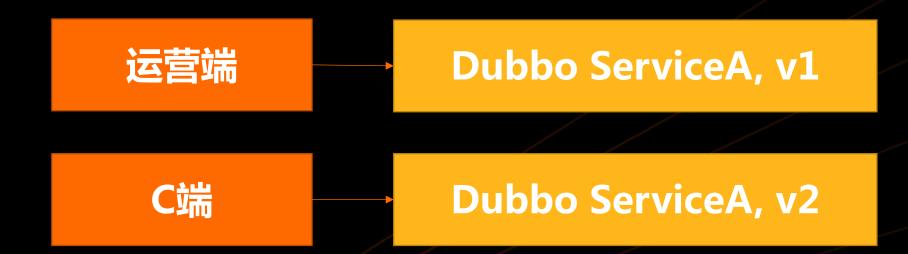
- 基于Nacos的特性开关 spring config集成
- HTTP协议转dubbo
- 魔改shenyu (神禹)

POST/GET -> 鉴权 -> Dubbo调用

• 基于Dubbo版本的服务实例隔离

问题

- 和K8S的service mesh格格不入
- 服务注册基于IP而不是servicename
- Nacos无法提供控制面
- Dubbo mesh?



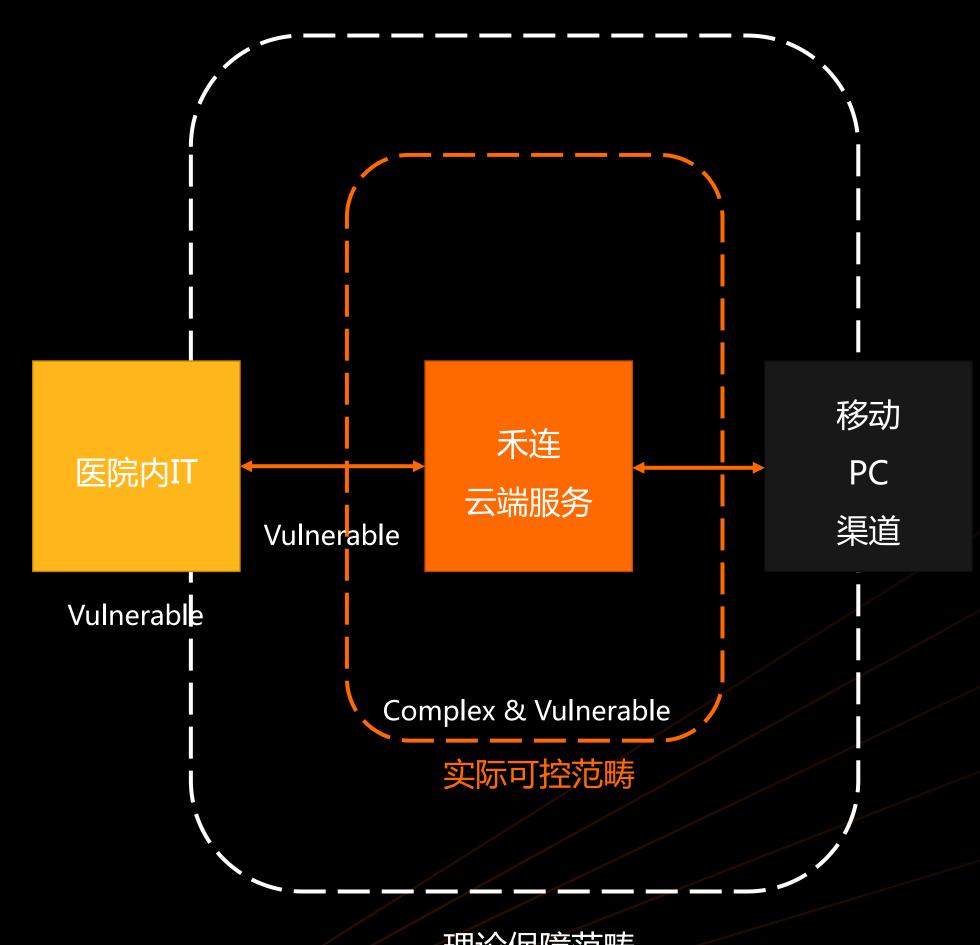
高可用保障一常规

运维手段

- 镜像快速发布/回滚 gitlab、云效/jenkins
- 监控告 Prometheus/ARMS
- 实例隔离
- 限流和熔断?
- 异地文备 成本考虑

开发手段

• 禁用存储过程和医院IT—刀两断



理论保障范畴

高可用保障一反脆弱

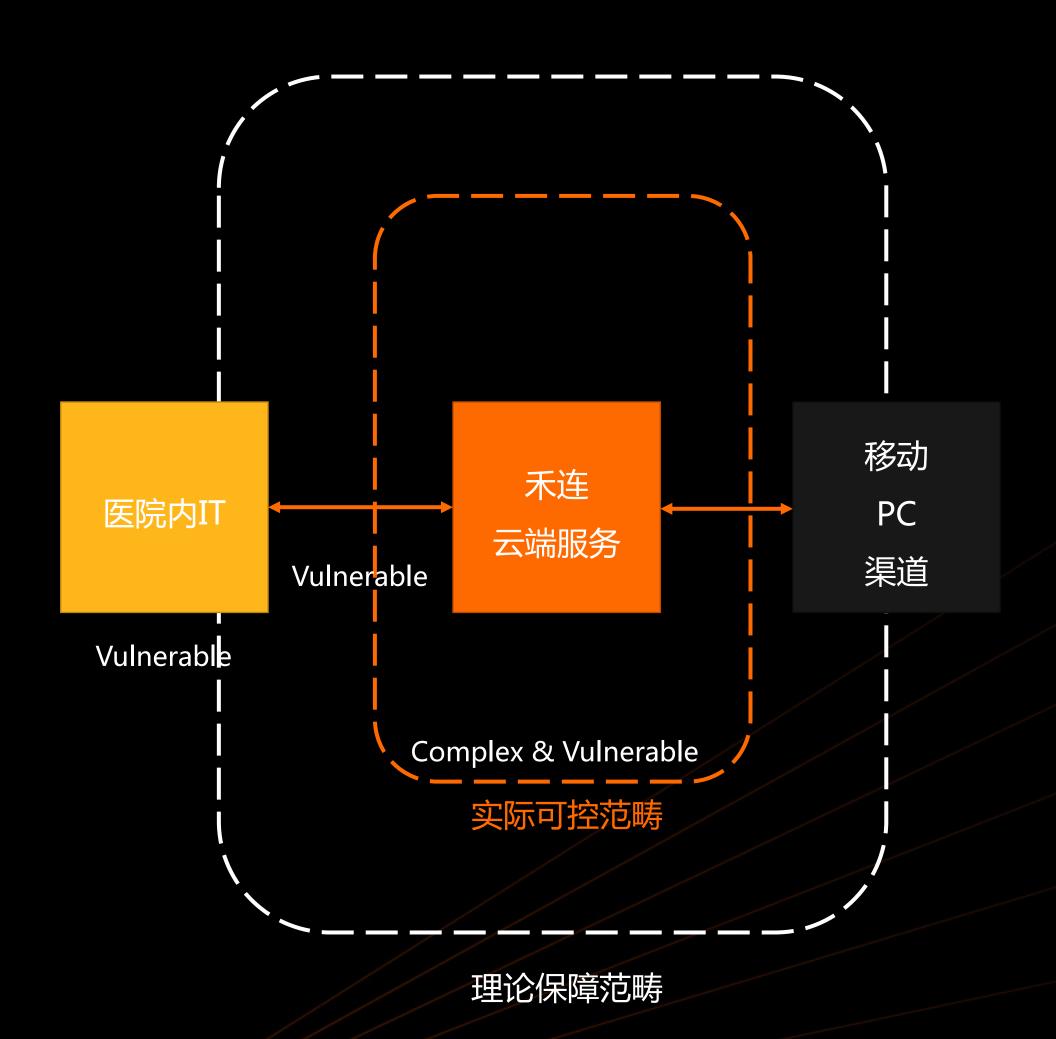
护士视角

• 系统出问题了,快解决,具体什么出了问题我不懂 甚至1/3是操作问题

一几乎没有边界的保障工作

反脆弱

• 如果系统一定会出问题,怎么保障服务履约?



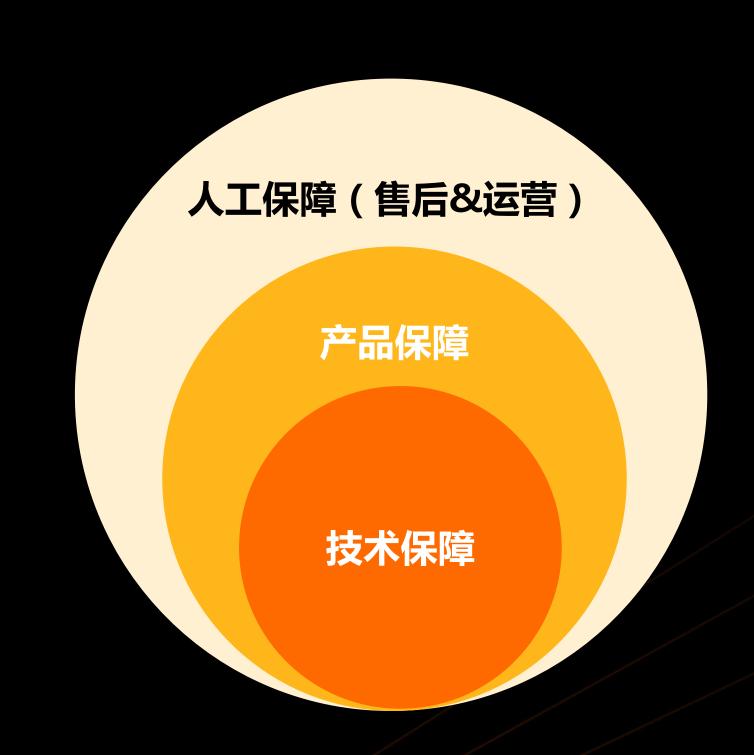
高可用保障一反脆弱

反脆弱

• 如果系统一定会出问题,怎么保障服务履约?

包心菜策略 突破技术职能边界

- 贾维斯系统:独立业务系统的数据合法性、一致性检查 云端、院内
 - 告警后的协同人工处理 预约类场景
- 急救箱系统:人工让用户能够正常体检 现场类场景



包心菜策略

Thanks 中间件开发者沙龙

禾连健康 邓志豪