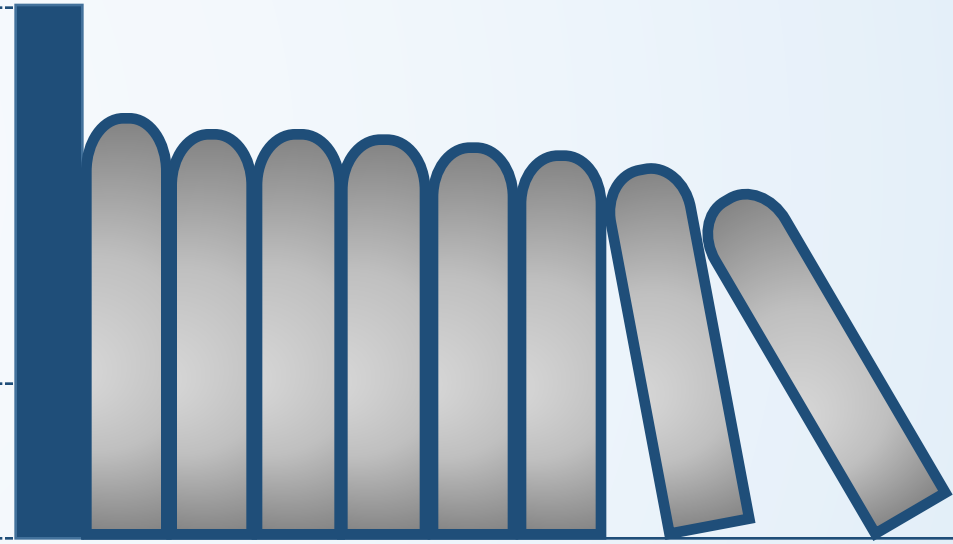
云原生前景

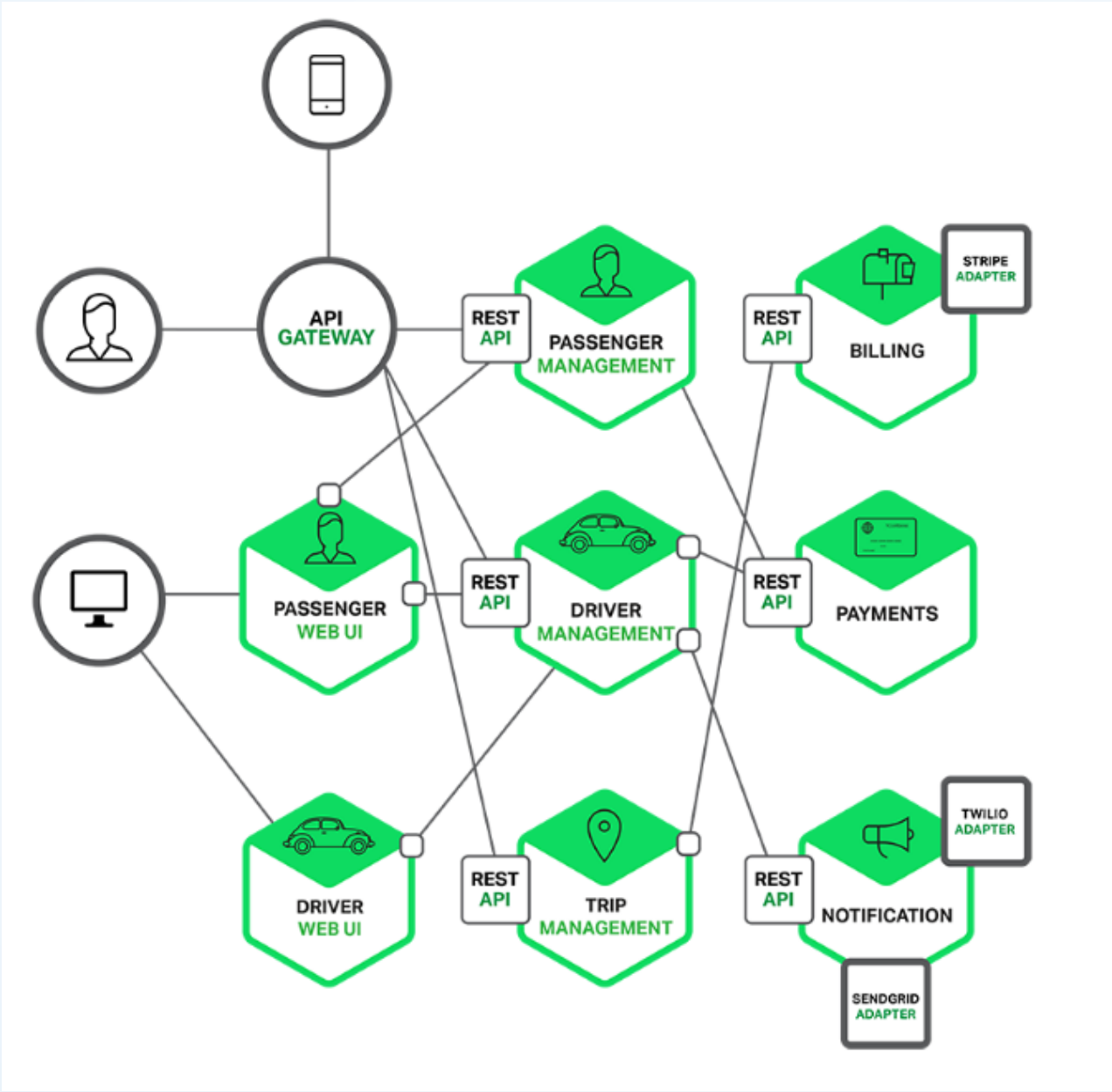
主讲人：宋小金



历史脉络

时代	时间	技术特点	典型问题
传统原始	2006年以前	物理机部署 单体应用，简单拆分 IOE生态	不隔离，资源利用率低，瀑布式开发，交付周期长
PC互联网	2006年 ~ 2010年	VMware/OpenStack SOA架构，MySQL，缓存 x86生态	高并发，资源池/环境不统一，互相依赖多，Dev/Ops没拉通
移动互联网	2011年 ~ 2017年	Docker/Kubernetes/Mesos 微服务架构，NoSQL 大数据，各种中间件 x86/GPU/ARM	迭代快，资源异构，混合部署，服务高可用，自动快速弹性
万物互联	2018年 ~	Docker/Kata/Kubernetes 去中心化，分布式架构 大数据/人工智能/边缘计算/区块链	资源差异，去中心，网络时效/带宽/拥塞

微服务架构


API
GATEWAY
REST
API
PASSENGER
MANAGEMENT
REST
API
BILLING
STRIPE
ADAPTER
PASSENGER
WEB UI
REST
API
DRIVER
MANAGEMENT
REST
API
PAYMENTS
DRIVER
WEB UI
REST
API
TRIP
MANAGEMENT
REST
API
NOTIFICATION
TWILIO
ADAPTER
SENDGRID
ADAPTER

微服务碰上容器

- 可以打包应用以及依赖包到一个可移植的容器中，部署到任何流行的Linux机器上，也可以实现资源及环境隔离。容器是完全使用沙箱机制，相互之间不会有任何接口。
- 可以把业务逻辑、数据库、储存、中间件等拆分成若干个容器，然后像搭积木一样组合起来，让彼此通信，从而形成微服务。
- 因此微服务很适合用容器封装，每个容器承载一个服务，每台服务器同时运行多个容器，统一编排调度，非常高效轻松地支撑业务的开发运维

总结：微服务生命周期与容器绑定，把对微服务的管理转变成对容器的管理

PS：Docker项目出现于2013年，Kubernetes出现于2014年

传统云 vs 容器云

类别	传统云	容器云
维度	资源维度	应用维度
服务	IAAS	CAAS
关注	物理资源（计算，网络，存储）的池化 实现资源与业务的解耦	底层环境（构架，运行时，中间件）服务对象化，实现业务与基础设施解耦
目标	提高资源利用率，降低硬件成本	支撑产品能够快速迭代 提升研发效率，保障高可用性 实现弹性伸缩应对业务爆发等
产品	OpenStack CloudStack VMware	Swarm Mesos Kubernetes

云原生

云原生的潜台词

- 起个新名词，提个新概念，利于区分宣传
- CNCF的成立， Cloud Native就是云原生
- 区分早期容器云概念，专指以Kubernetes为核心的容器云生态体系
- 原生本质即是开箱即用，无需太多额外定制适配开发

思考：CNCF为什么以Kubernetes为核心构建生态？

编排系统不止一个，理念先进也未必，毕竟Borg系统十几年前就有

云原生

问题：CNCF为什么以Kubernetes为核心构建生态？

- 为Docker为起点的容器技术和微服务架构的兴起，搅动了传统云市场，也让某些云厂商看到了弯道超车的机会，这个云厂商就是Google
- 目前全球云计算市场的格局是AWS领先，微软，阿里云，Google居后，其它云公司跟随，对于Google这家以技术见长的公司来说，现状是不可接受的
- 超车的方法就是主导容器生态的发展，屏蔽IaaS的差异，让容器生态朝自己有利的方向演进，有利于其它云厂商的客户迁移，提高市场份额
- 具体方法就是以Borg为蓝本推出Kubernetes，联合Redhat，CoreOS等公司成立CNCF基金会

背后实质即商业本质逻辑，要主导容器技术标准

前景 ---- all in kubernetes

现状：目前kubernetes+docker主要用于构建微服务管理平台及AI训练深度学习平台，ToB的容器平台，大厂，创业公司基本上都是PaaS平台及AI训练平台

前景：all in k8s，kubernetes会演变成一个资源调度引擎，统管资源池，快速资源交付，充分资源共享，提升资源利用率

时代背景：

AI， 5 G，物联网，边缘计算 -- 新一代基础设施，连接方式转换

消费互联网转向产业互联网，ToB产业兴起，技术进一步分工

IT架构从大中台转向工字型架构，即前端 / 数据层厚，中台薄，终端多样化，数据层在资源弹性，横向扩展能力的赋能下，承接更多中间件能力，这需要Kubernetes的加持