

创建可扩展，全球化和高可用的网络应用

姓名
职位
公司



日程



高可用性的设计
高扩展性的设计
高性能的设计

假设

你知道基础

Windows Azure Web/Worker 角色

SQL 数据库

Windows Azure 存储

异步编程

Windows Azure 分析

你部署过服务到 Windows Azure

所有的东西将要（迟早）会坏掉

Availability



为什么服务失败?

增加的工作量

失效

硬件
网络
平台服务
瞬态条件

人为

更新

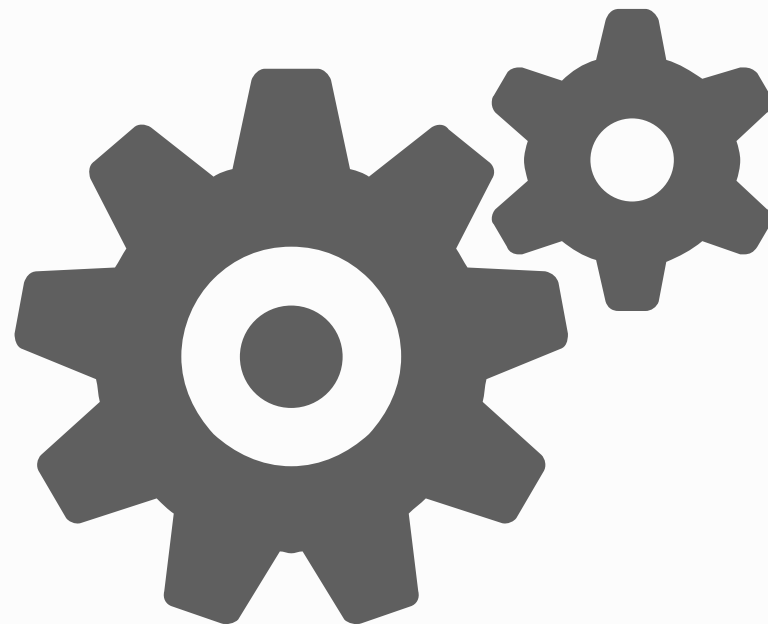


可用性的意思是什么？

相同功能

降级功能

故障安全



基础 – 你免费拿到的

弹性

简单部署计算资源并且向上或向下扩展

自动服务管理

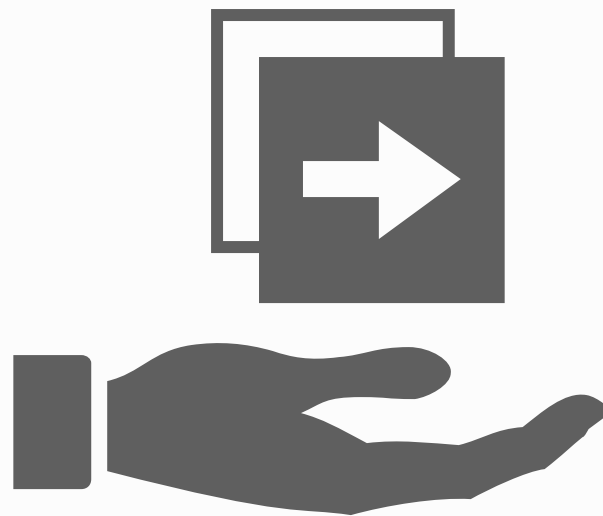
Windows Azure 会(自动) 恢复坏掉的节点

故障域

Windows Azure 跨故障边界部署服务

存储复原

维护三份存储备份



容错

当Windows Azure坏了，它自己修复

你可以吗？

法律化操作

升级域

在服务定义中配置

```
<ServiceDefinition name="RedDir" xmlns="http://schemas.microsoft.com/ServiceHosting/2008/10/ServiceDefinition" upgradeDomainCount="3">
```

数据中心瞬态条件

你有重试机制吗？

重试逻辑的意思是？

在数据中心/网络/服务中的瞬态条件

举例：

SQL Azure 错误 40501
服务当前很忙。10秒钟后请重试

临时失效处理框架

<http://windowsazurecat.com/2011/02/transient-fault-handling-framework/>

对任何事情重试可能是外部的且有瞬态条件*：

SQL 数据库
Windows Azure 存储
服务总线
第三方服务

重试

演示



服务的具体实现

没有那个平台服务你的服务失败吗？

你的服务可以从另外一个数据中心使用相同的平台服务吗？

你的服务可以临时不适用那个平台服务吗？

站点故障恢复

如果一个站点具体依赖的东西停止了，恢复到另外一个站点

简单: 使用流量管理器

困难: 写你自己的代码

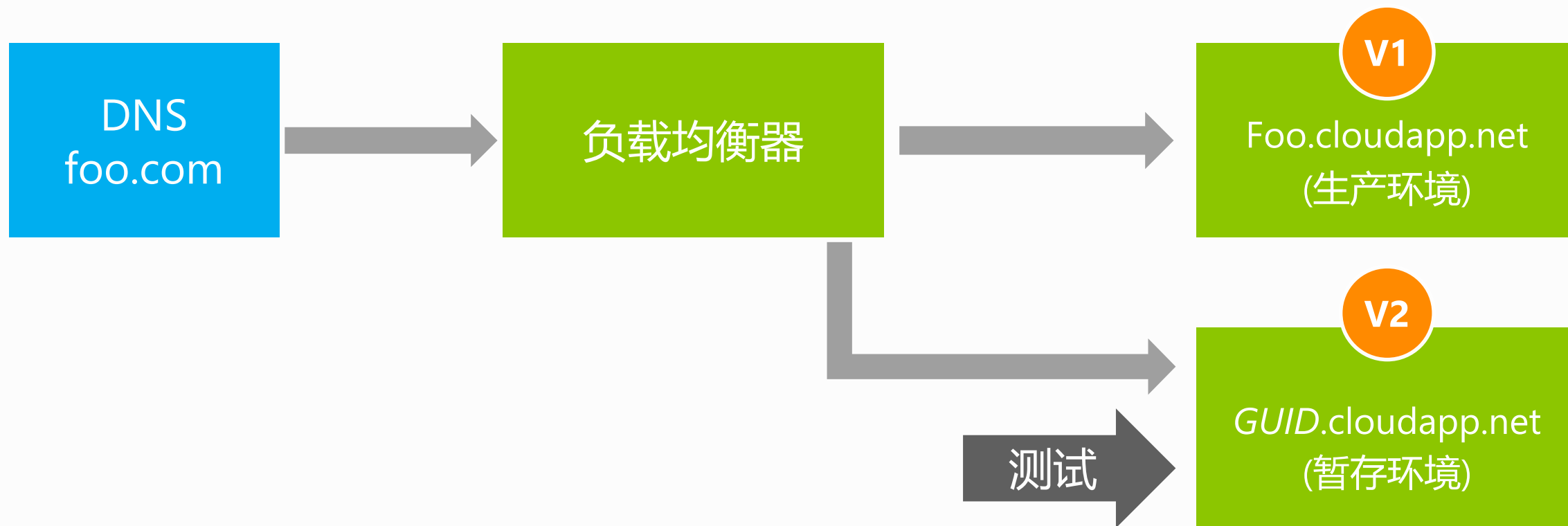


站点故障恢复

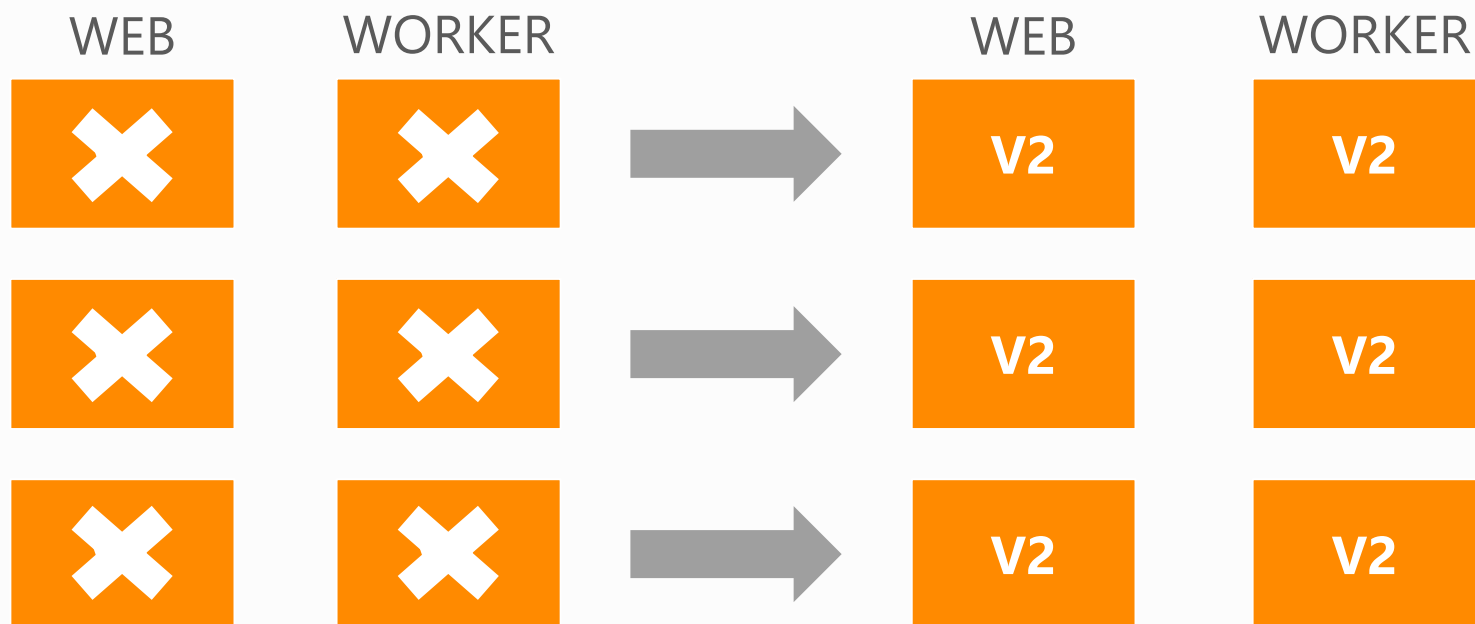
演示



更新策略: VIP 交换

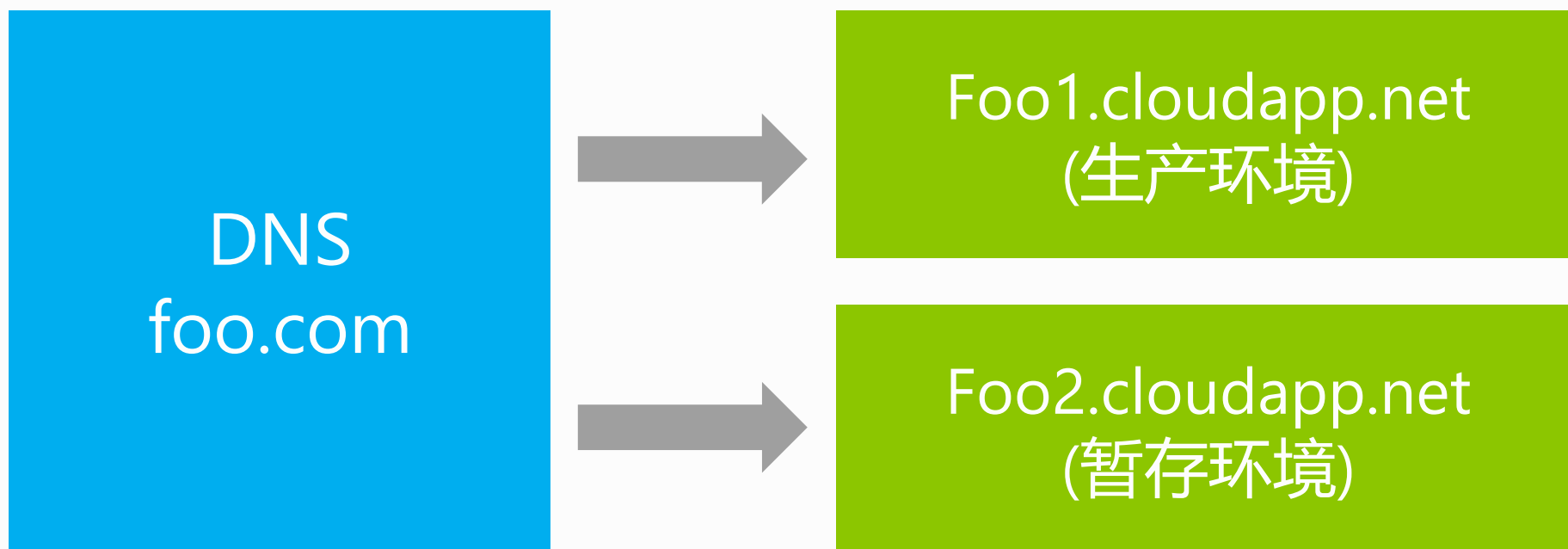


更新策略: 更新



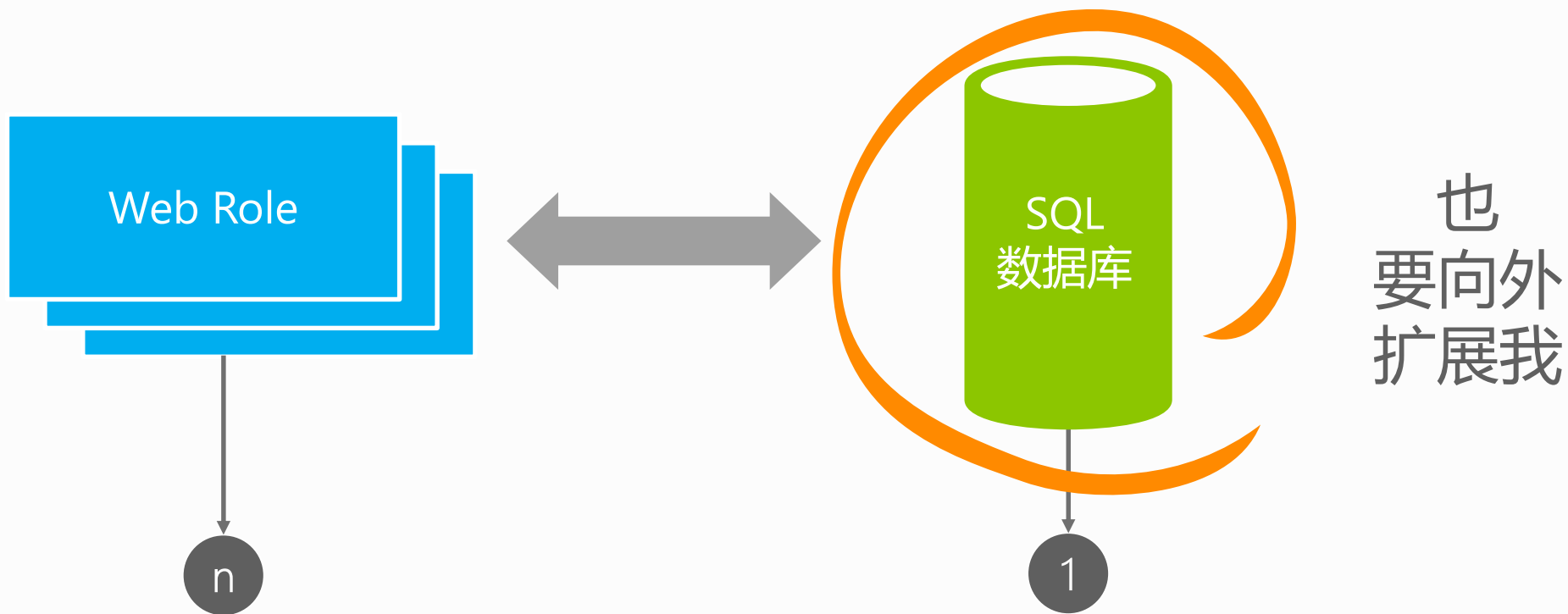
更新策略

新服务和交换DNS



可扩展性

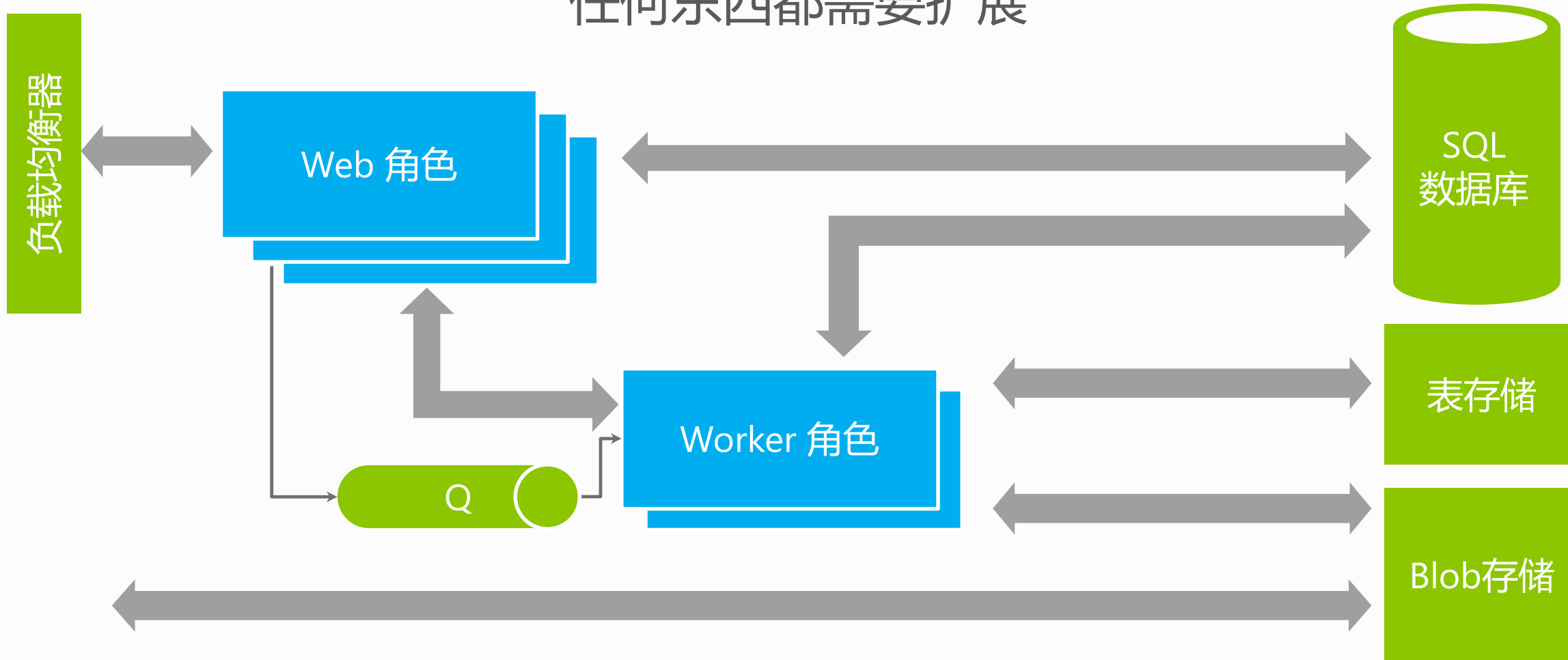
这有什么错？



50 x 1GB的数据库比1 x 50G的数据库要好

这个怎么样？

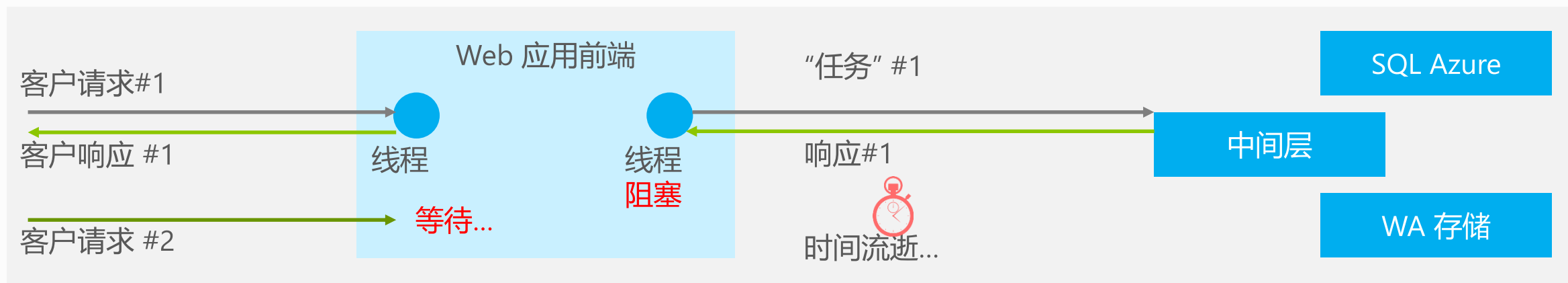
任何东西都需要扩展



同步设计模式

每个线程专用于一个未完成的请求

对每一个请求阻塞在每一步的“任务”完成，然后响应并重复



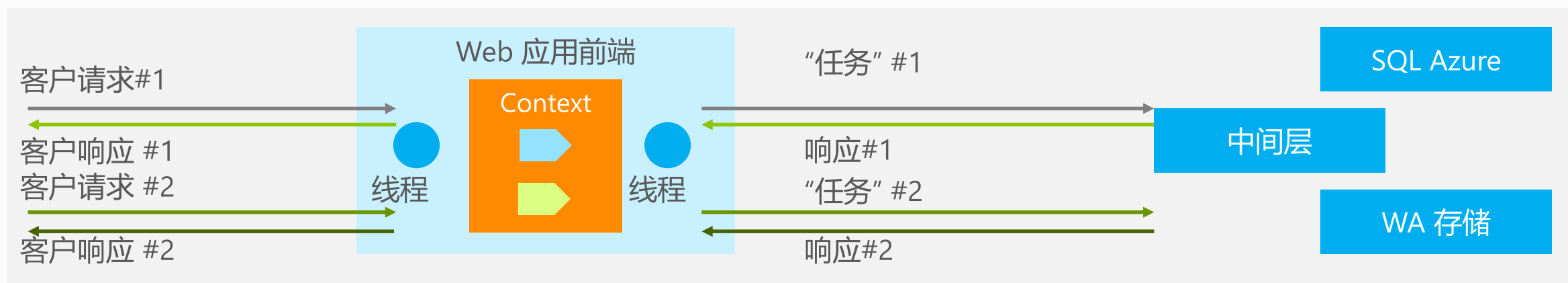
这种方法扩展性很差

每个未完成的请求存储在线程堆栈中。
在工作做完前线程阻塞
增加线程只能允许一个额外的并行请求

异步设计模式

每个线程在他们就绪时选取一个任务

一个线程处理一个请求，可以在第一个结束之前处理另外一个



这个方法扩展性很好

客户强求在应用的数据结构中显式跟踪。

当有任务要完成，线程从来不阻塞

每个线程可以处理许多并行请求

但是记账和同步会变得很难

性能



什么是Windows Azure 缓存?

- 使用虚拟机上空闲的内存作为高性能的缓存。
- 分布式的缓存集群同时存在已有的角色上，或者专用的角色上。
- 有高可用性和通知的命名的缓存
- 支持Memcached 协议

为什么使用Windows Azure缓存?

更快

没有外部的服务调用（额外的网络跃点）
在角色中同位

更便宜

没有外部的服务调用（额外的网络跃点）
使用你已经付钱的空闲内存

更可依赖

服务在运行 = 缓存可以用
在合租环境下没有限制

缓存

demo



为什么性能很重要

更响应的应用

更快的页载入时间

8 秒 vs. 3 秒?

更高交互性 – 新型应用

更好的用户体验 – 更多\$\$\$



全球考虑

网络延时

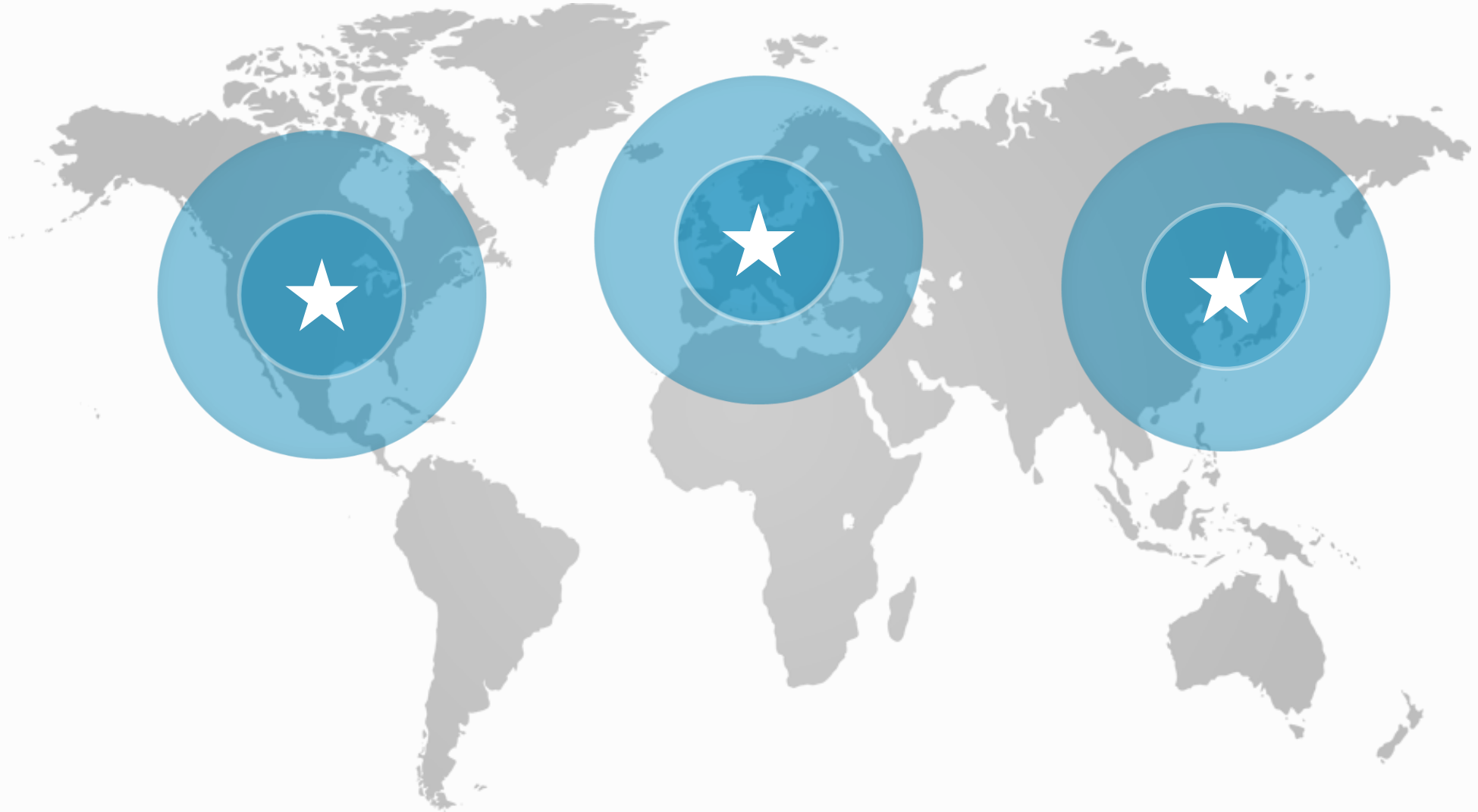
让计算更接近用户
让数据更接近用户

全球可用

数据中心停止
同步数据



网络延迟



内容分发网络(CDN)

高带宽的全球blob内容分发

24 全球位置 (美国, 欧洲, 亚洲, 澳大利亚和南美), 还在增加
无论用户在地球的什么地方, 无论存储账户在哪里, 体验完全相同

Blob 服务URL vs. CDN URL:

Windows Azure Blob URL: <http://images.blob.core.windows.net/>

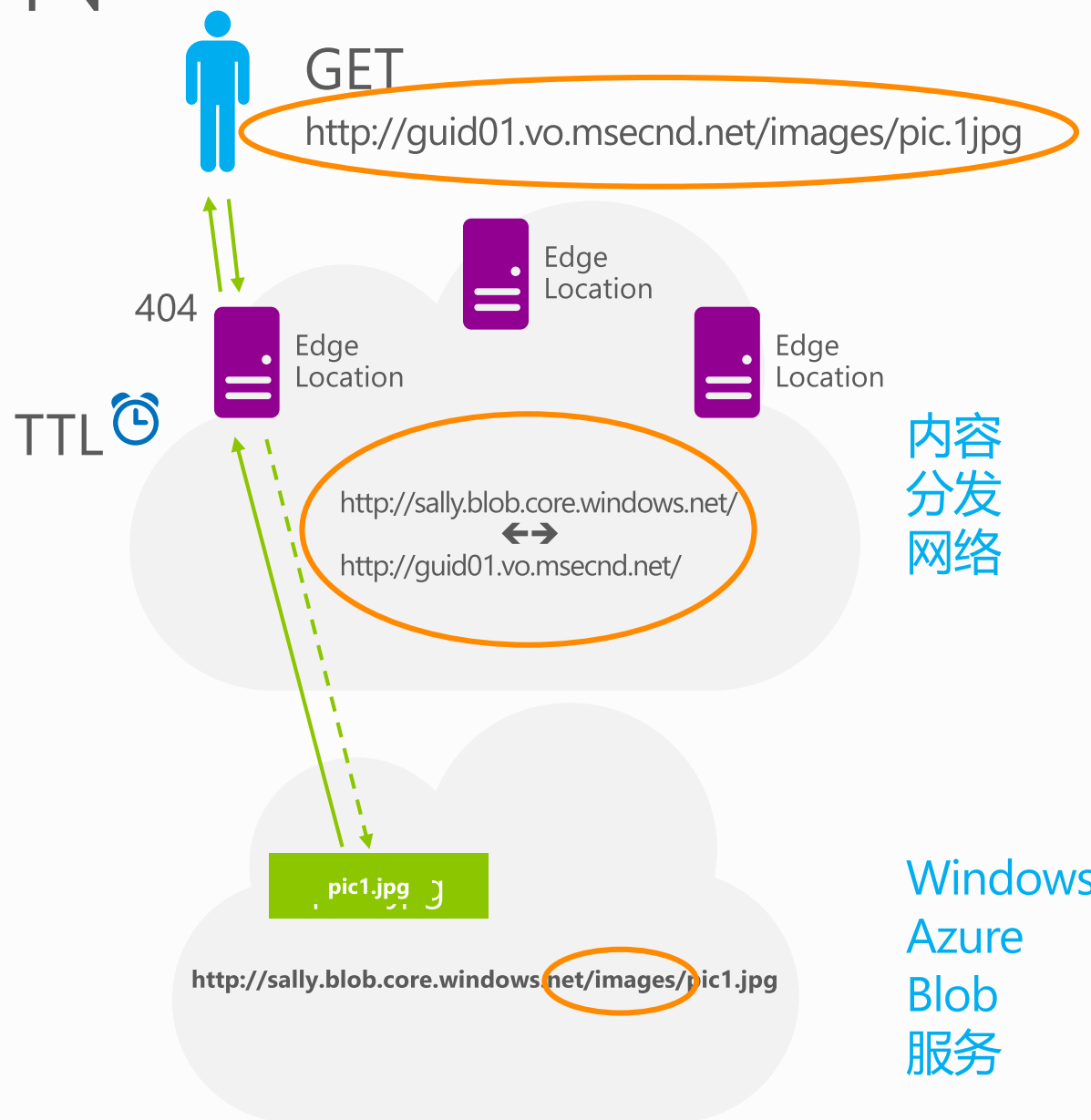
Windows Azure CDN URL: <http://<id>.vo.msecnd.net/>

CDN自定义域名: <http://cdn.contoso.com/>

Windows Azure CDN

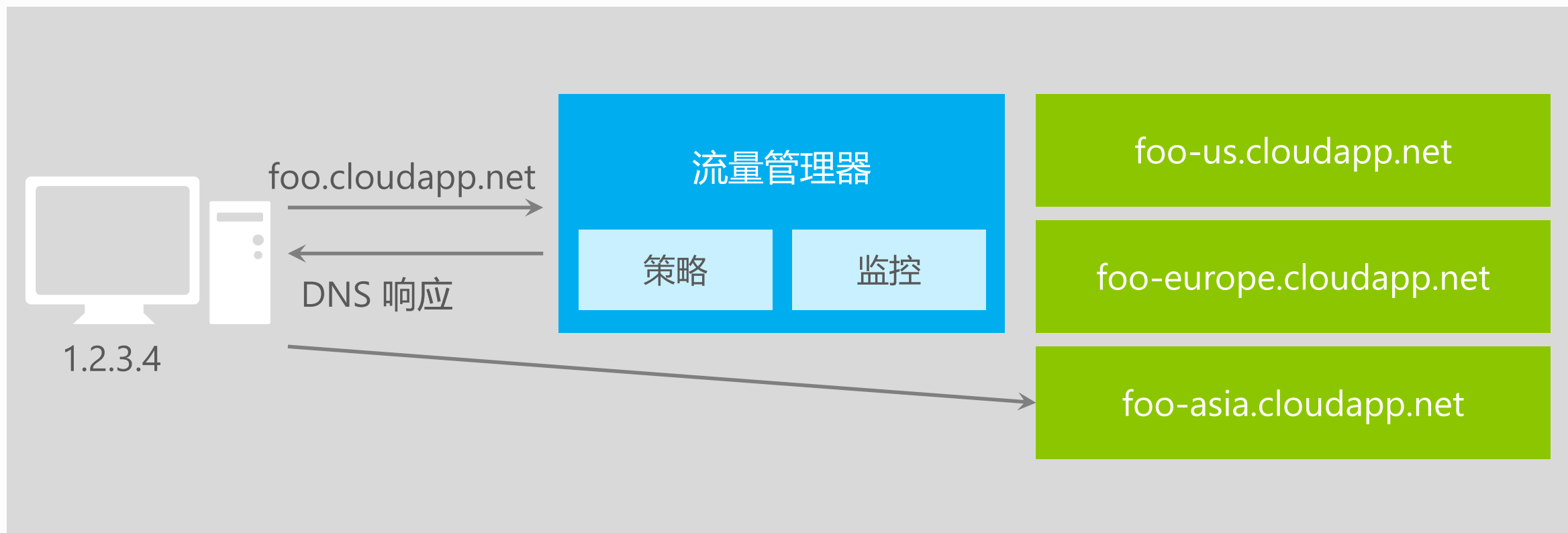
开始CDN:

通过Dev Portal注册CDN
将Container设置为公共



Windows Azure 流量管理器

通过Windows Azure流量管理器引导用户到最近区域的服务



流量管理器

demo



总结

Windows Azure 免费提供高可用
的能力

考虑向外扩展
处理瞬态条件

为可扩展性设计

异步模式
向外扩展

为最大性能和可达到性设计

缓存, CDN, 流量管理器, 等等.





© 2012 Microsoft Corporation. All rights reserved. Microsoft, Windows, Windows Vista and other product names are or may be registered trademarks and/or trademarks in the U.S. and/or other countries.

The information herein is for informational purposes only and represents the current view of Microsoft Corporation as of the date of this presentation. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information provided after the date of this presentation. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, AS TO THE INFORMATION IN THIS PRESENTATION.

Translated to Chinese Simplified Version by Shanghai Yungoal Info Tech Co., Ltd. [YunGoal](#)