

Windows Azure 服务总线

演讲者
职位
公司



日程



为什么用服务总线?

服务总线命名空间和访问控制

服务总线中继

服务总线消息传输

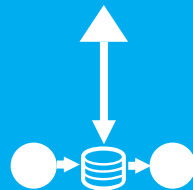
服务总线



连接性

服务中继
协议隧道事件

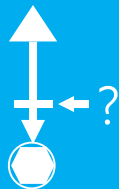
为连接应用程序跨越网络边界提供丰富的选项,



消息

队列的发布/订阅
可信的传输

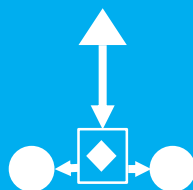
可靠的, 允许事务的商业应用云消息架构



服务管理

命名, 发现, 监控

持续的管理表面和服务观察能力



集成的路由

协调转换

基于内容的路由, 文档转换和流程协调

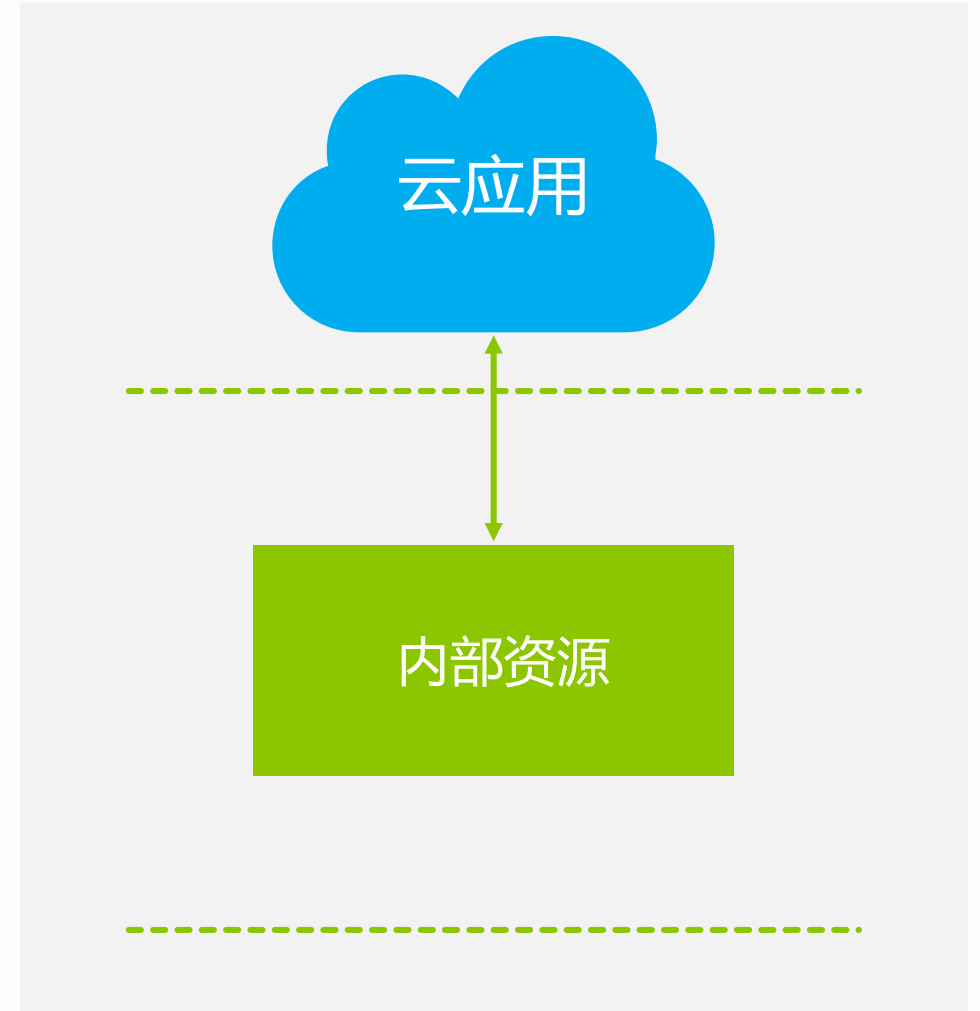
为什么用服务总线？



云/内部网络集成

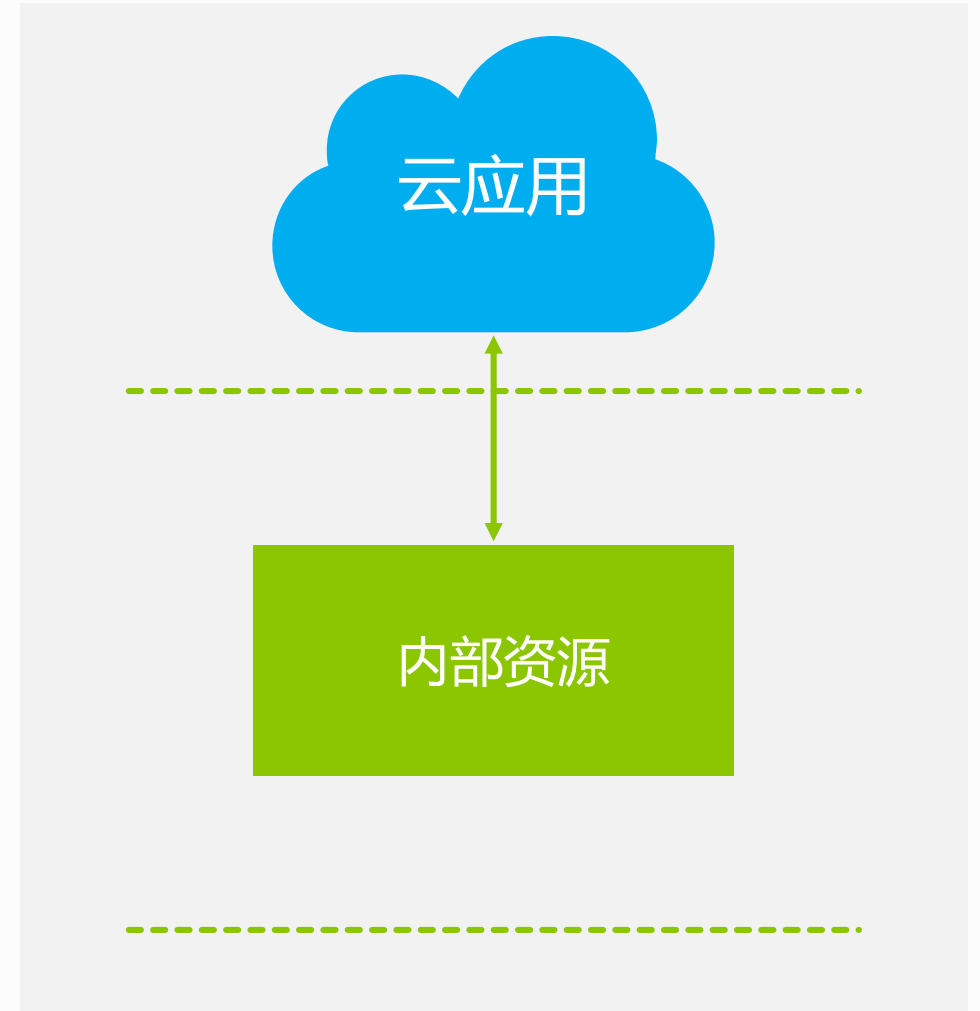
云托管，可依赖的异步消息架构，
支持发布/订阅

基于云的中继允许穿越NAT/防火
墙访问内部的资源



云/内部集成

服务注册允许把应用服务在不同网络环境中分布，把端点组织入一个通用的，可发现的网络表面。和访问控制集成，提供了联合身份认证的支持



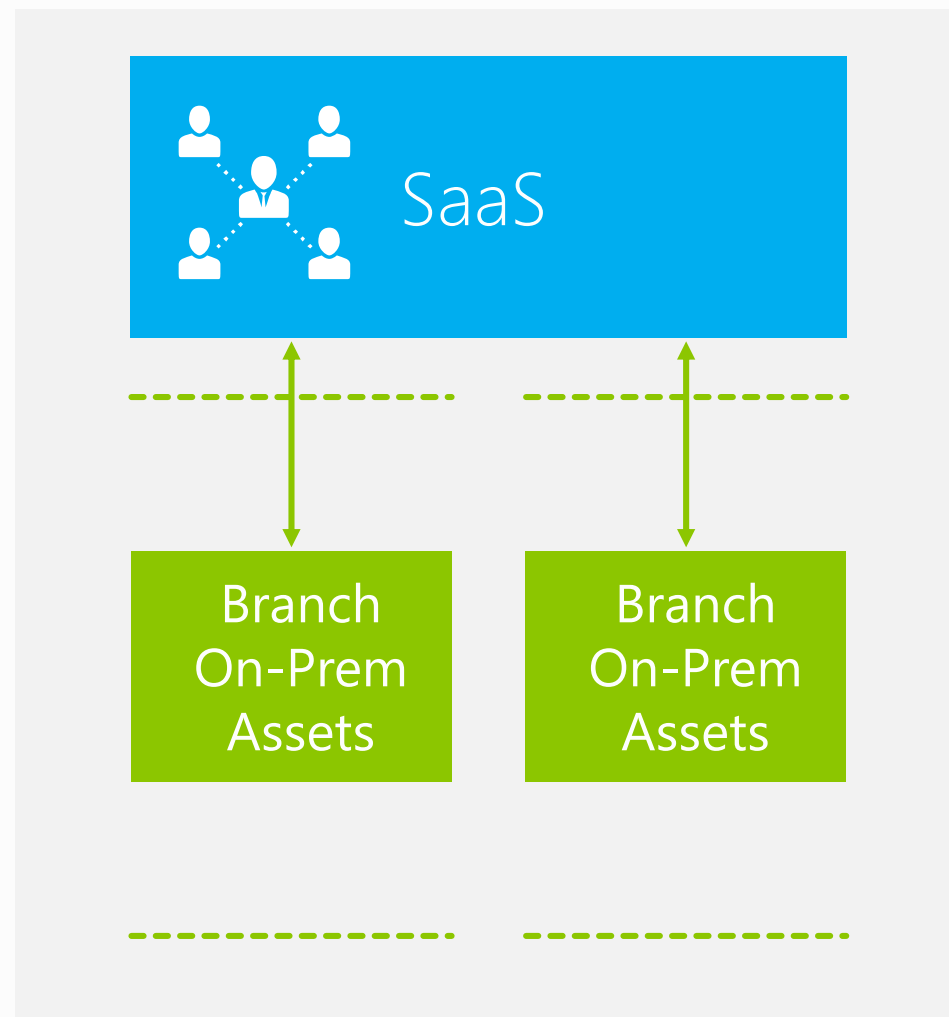
跨站点的联合 (SaaS)

端点联合而不是网络联合(VPN)
非侵入性的，不需要网络重新配置
允许集成的场景：

多租约

最小的项目信任

最小的或非控制内部网络环境



特许经营集成场景

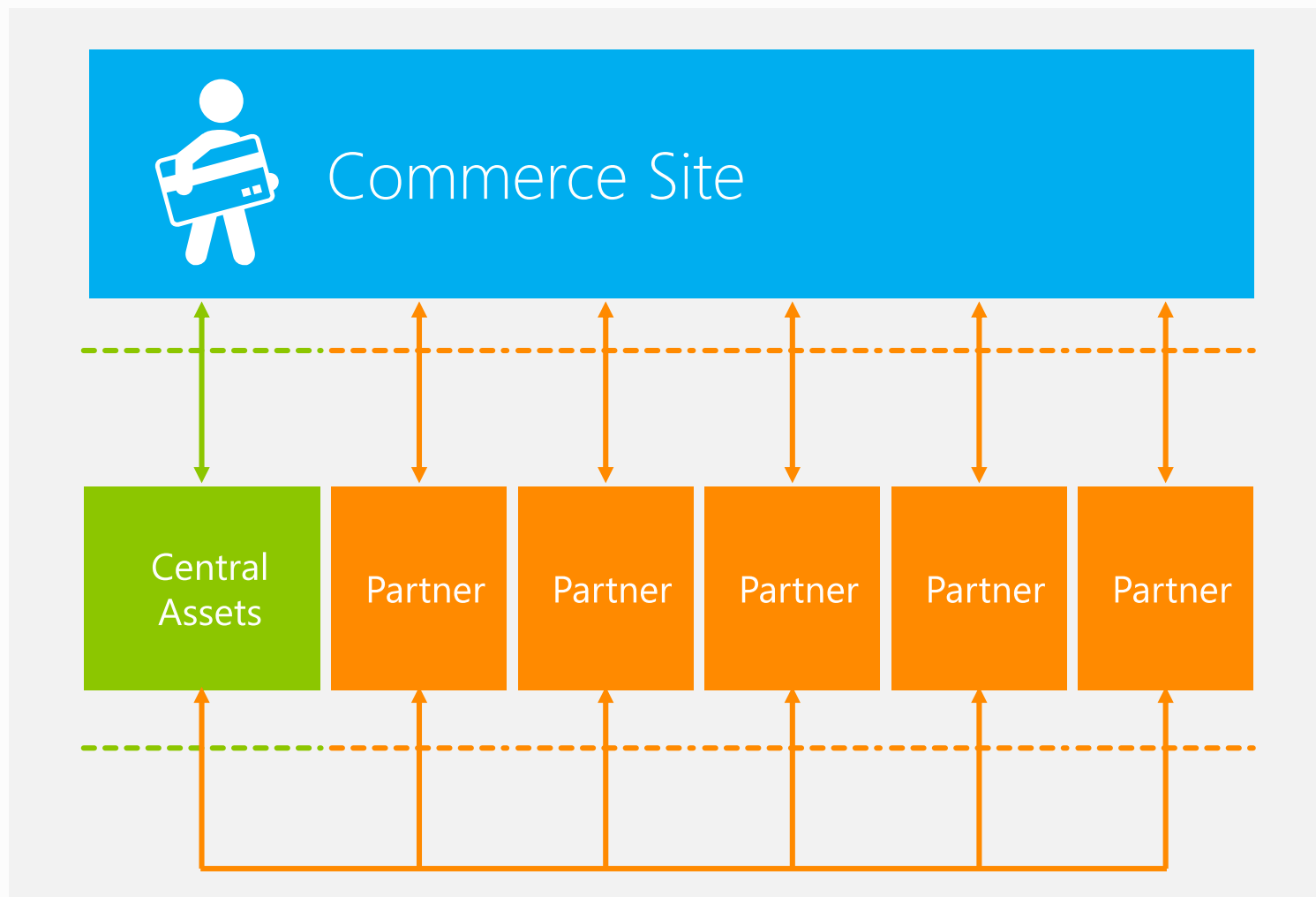
允许合伙人和特许经营环境的集成

低信任

限制的控制

不同的站点，不同的连接

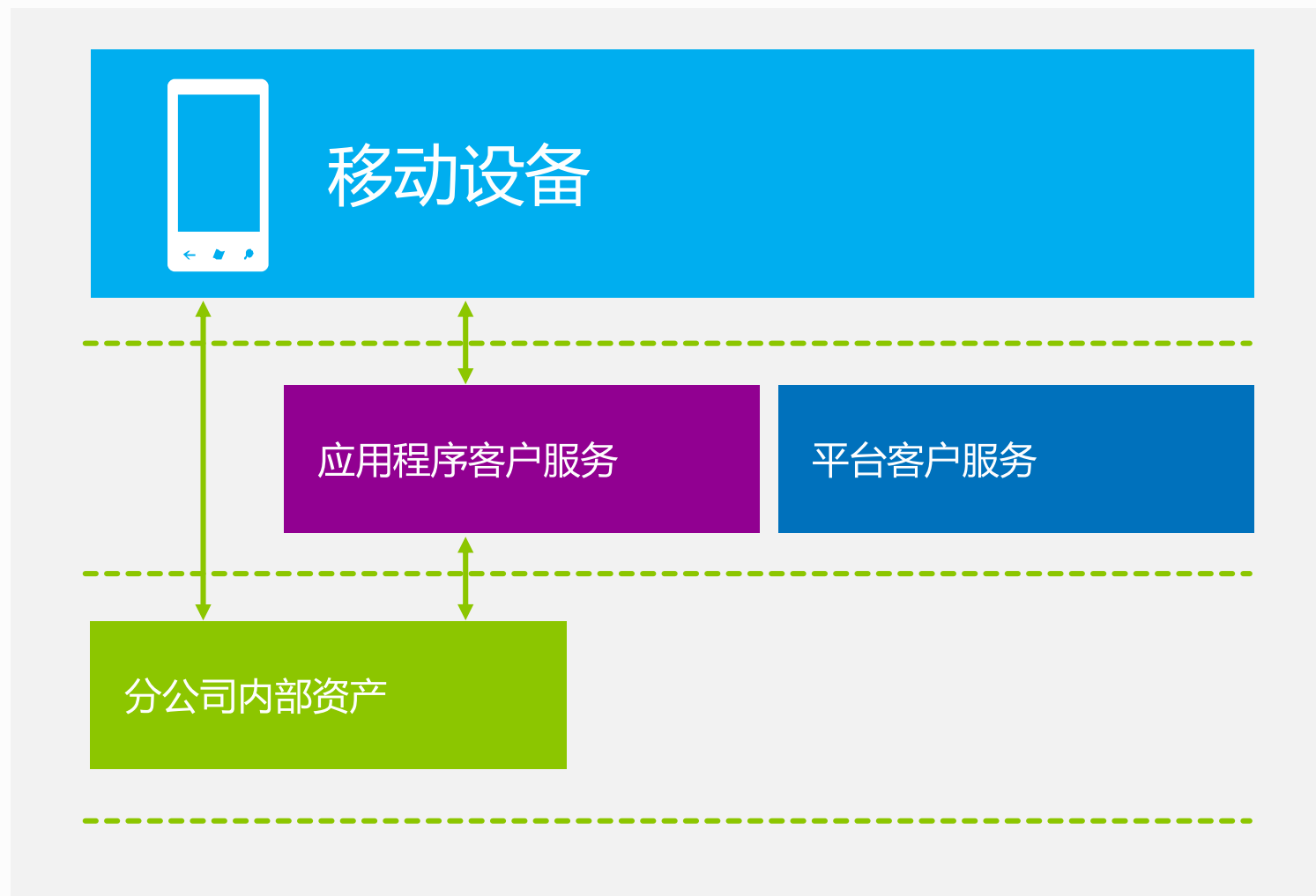
直接的点对点访问和云访问



移动的工作/客户集成场景

移动设备大量地不在防火墙内

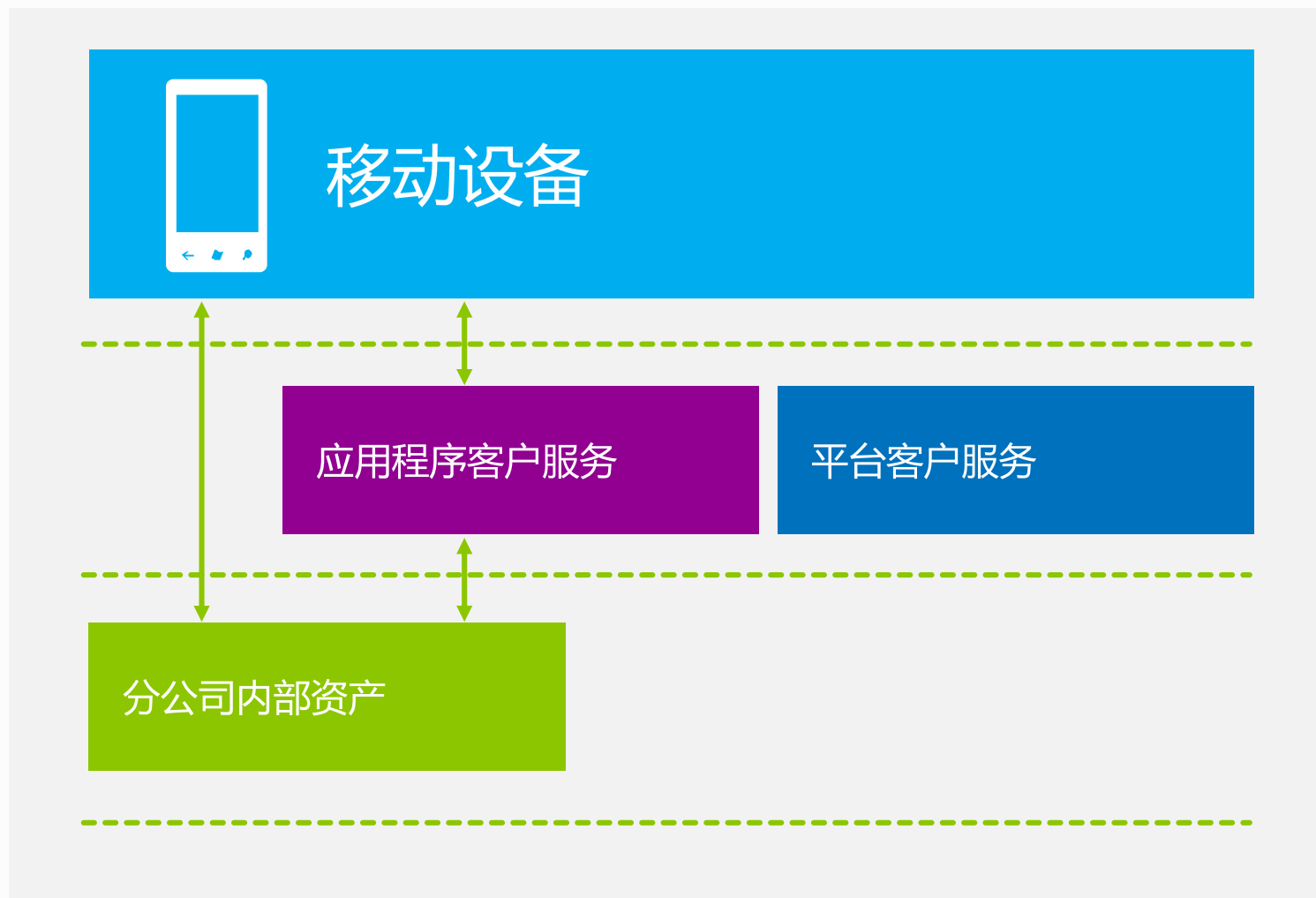
由于复杂的配置和管理，
VPN的解决方案非常不实际



移动的工作/客户集成场景

但是，移动设备需要能够访问内部资源

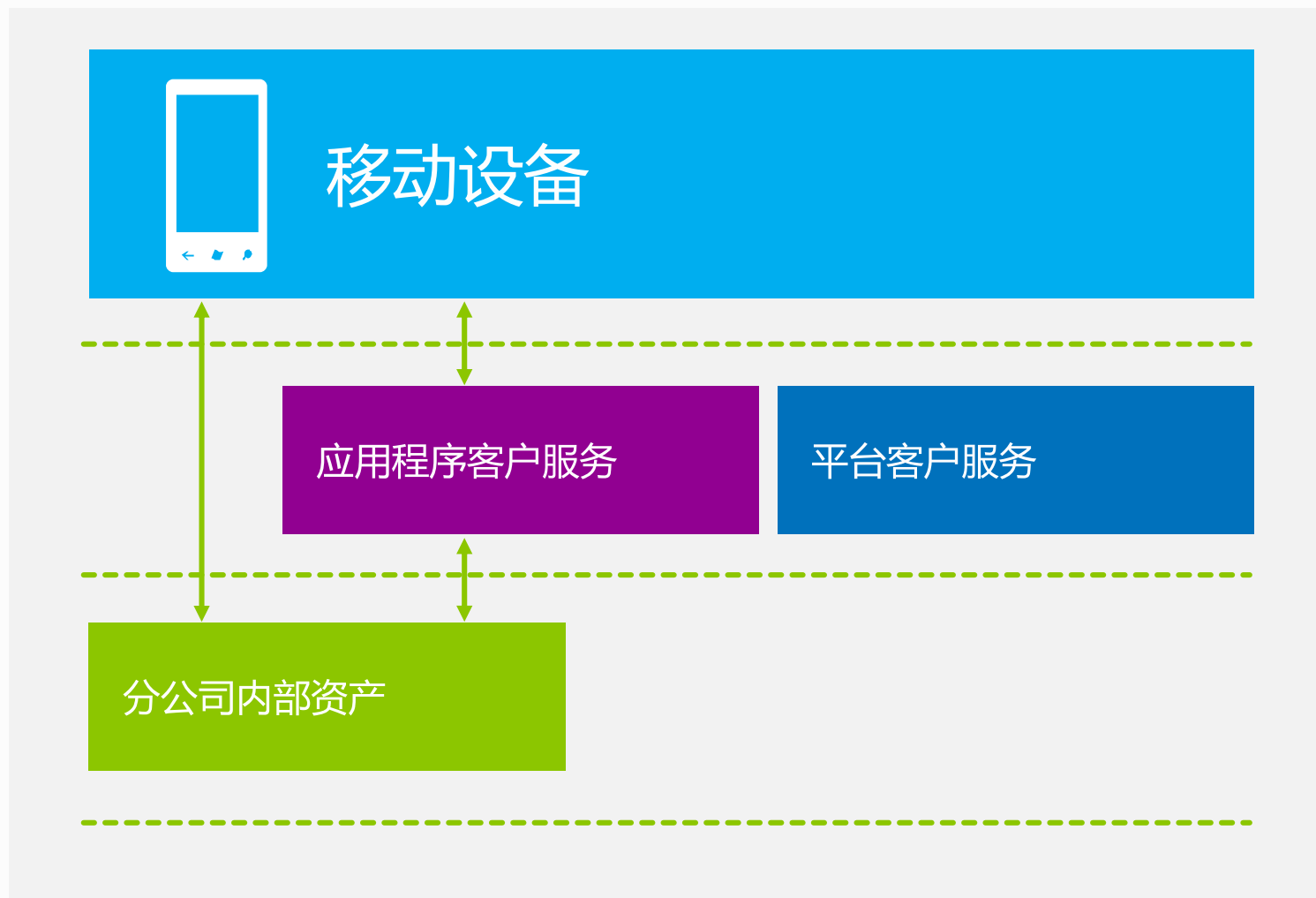
为了达到较大的企业，而不是为了那些没有静态IP或至少公共IP的小企业



移动的工作/客户集成场景

直接访问，通过使用ISV提供的服务的云

未来还会支持Azure固有的移动服务，例如服务总线移动设备推送支持

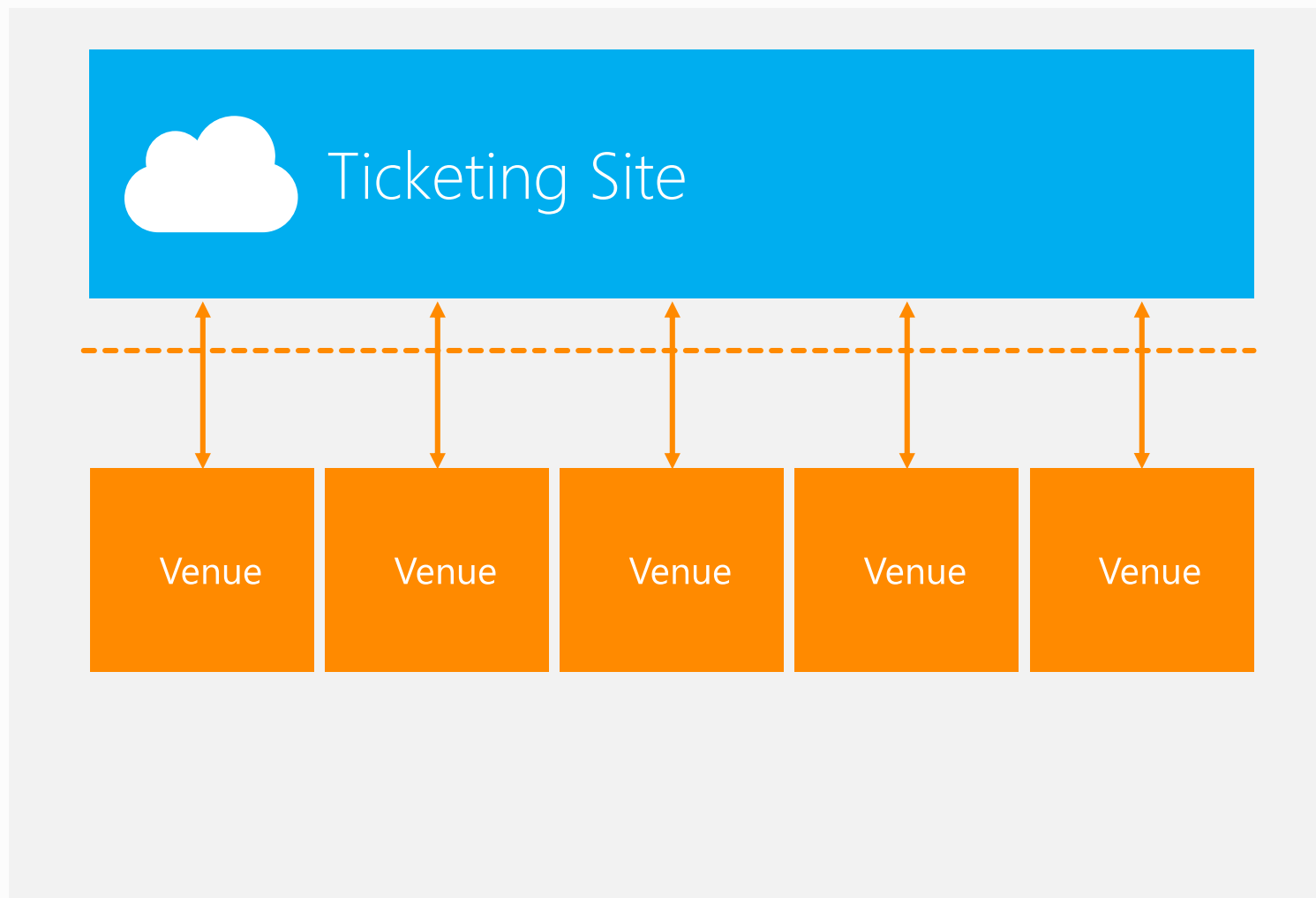


联合的云/内部解决方案

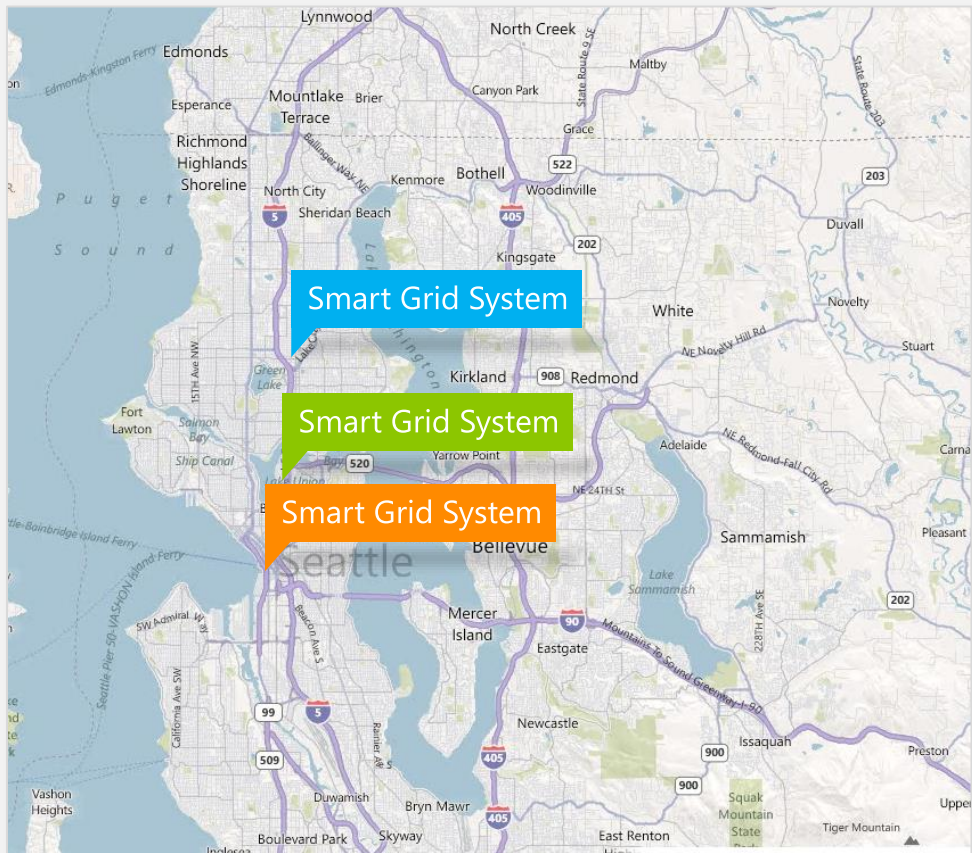
联合解决方案提供云里和内部相同的功能

云通过到达性和扩展性增强了内部的解决方案

内部方案提供了不妥协的可用性，即使网络断线



大规模的消息/命令控制



“最后一公里”的问题

规模上到达消费者或工业设备

在应用规模上广播事件数据

基于地理位置或者人口统计发送目标通知

大规模通知和广播将会在2012年成为服务总线的一部分

服务总线命名空间和 访问控制

服务总线命名空间

<https://yourapp.servicebus.windows.net/foo/bar/baz>

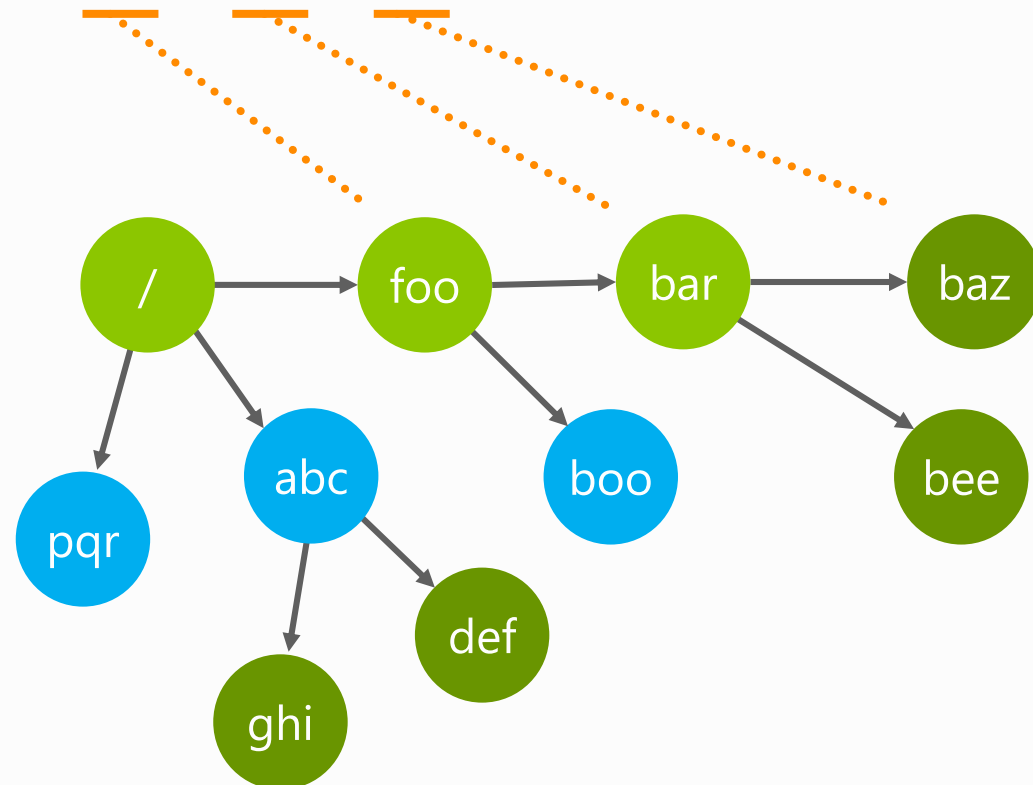
命名树

根上的ATOM Feed，用于发现
通过ATOM Feed层次使用REST管理
所有能够存在的名字都存在

“无限”深度

事实上: 32 段, 450 路径字符限制

实体拥有命名空间树的叶子结点
任何分支能够通过ACS管理安全性



服务总线和访问控制

服务总线和访问控制之间特殊的关系

每个服务总线命名空间在ACS中拥有'buddy'命名空间

'yourapp.servicebus.windows.net'

'yourapp-sb.accesscontrol.windows.net'

'-sb' 命名空间

依赖于服务总线根的预定义的命名空间

无法删除，系统管理的密钥，使用默认规则组

预定义的服务认证'所有者'

无法删除，通过默认规则组配置为超级用户

'对于所有者拥有监听，发送和管理的权限'



服务总线权限和声明

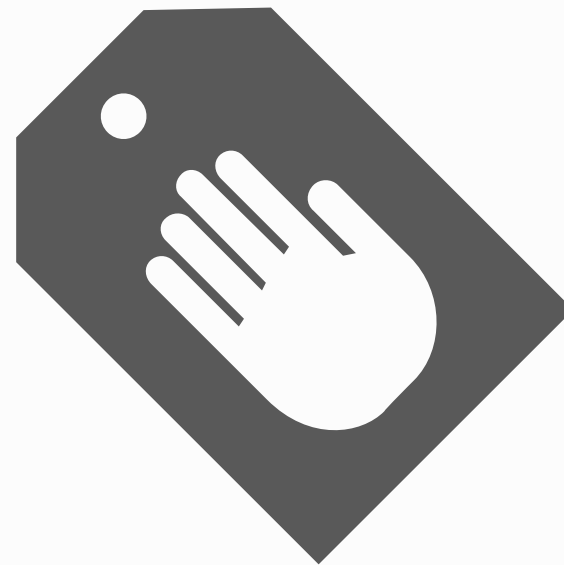
服务总线定义了一个授权声明类型，
包括3种可能的值描述不同的授权
操作

`'net.windows.servicebus.action'`

'Send' – 对于服务总线实体，允许发送操作

'Listen' – 对于服务总线实体，允许发送和接收操作

'Manage' – 允许例如创建，检查或者删除服务总线实体

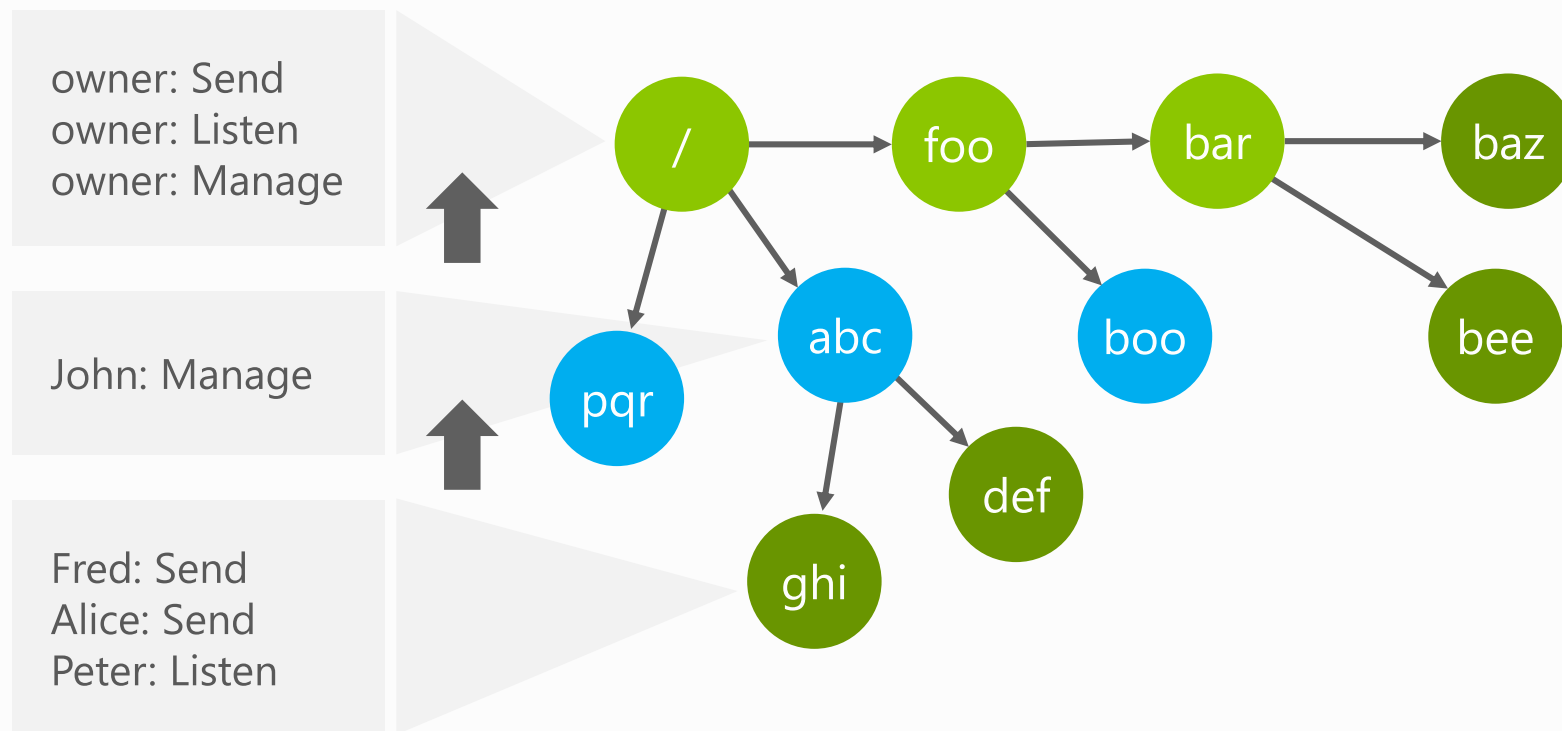


访问控制 – 概念模型

命名空间中每个名称/分支能够拥有一个相关的权限集合

通过ACS提供的联合认证提供者赋予权限

定义了服务总线实体的权限: 发送, 监听和管理



访问控制- 实现

<https://yourapp-sb.accesscontrol.windows.net>



命名空间和访问 控制

演示



服务总线中继

“把Web Service从任何地方传递到任何地方”



关键能力

单向中继，单播和多播
WCF NET.TCP直连中继
WCF HTTP支持REST和SOAP 1.1/1.2中继
使用访问控制保护端点



连接选项

外向TCP (端口9350-9353)
9350 单向非安全TCP (客户端)
9351 单向非安全TCP (所有的监听者，安全的客户端)
9352 安全 TCP 会合 (所有监听，除了单向)
9353 直连探测协议 (TCP 直连监听)
外向HTTP (端口80, 监听)
通过一对HTTP请求覆盖TLS/SSL形成TCP等价的隧道
备用的连接路径，如果外向TCP被禁用
外向HTTPS (端口443, 发送)

中继编程模型

完整的WCF编程模型

WCF 对称绑定功能

WebHttpRelayBinding (HTTP/REST)

BasicHttpRelayBinding (SOAP 1.1)

WS2007HttpRelayBinding (SOAP 1.2)

NetTcpRelayBinding (Binary transport)

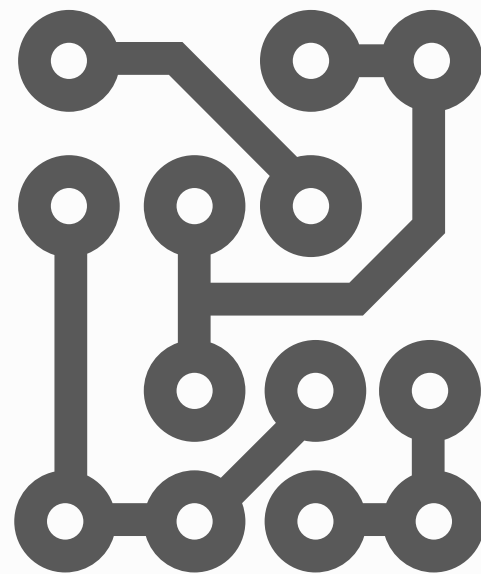
特殊的服务总线绑定

NetOnewayRelayBinding (多播单向)

NetEventRelayBinding (多播单向)

自定义的传输绑定

WebHttpRelayBinding 提供了完整的交互性，使用任何 HTTP/REST 客户端，BasicHttpRelayBinding 使用任何 SOAP 客户端



单向

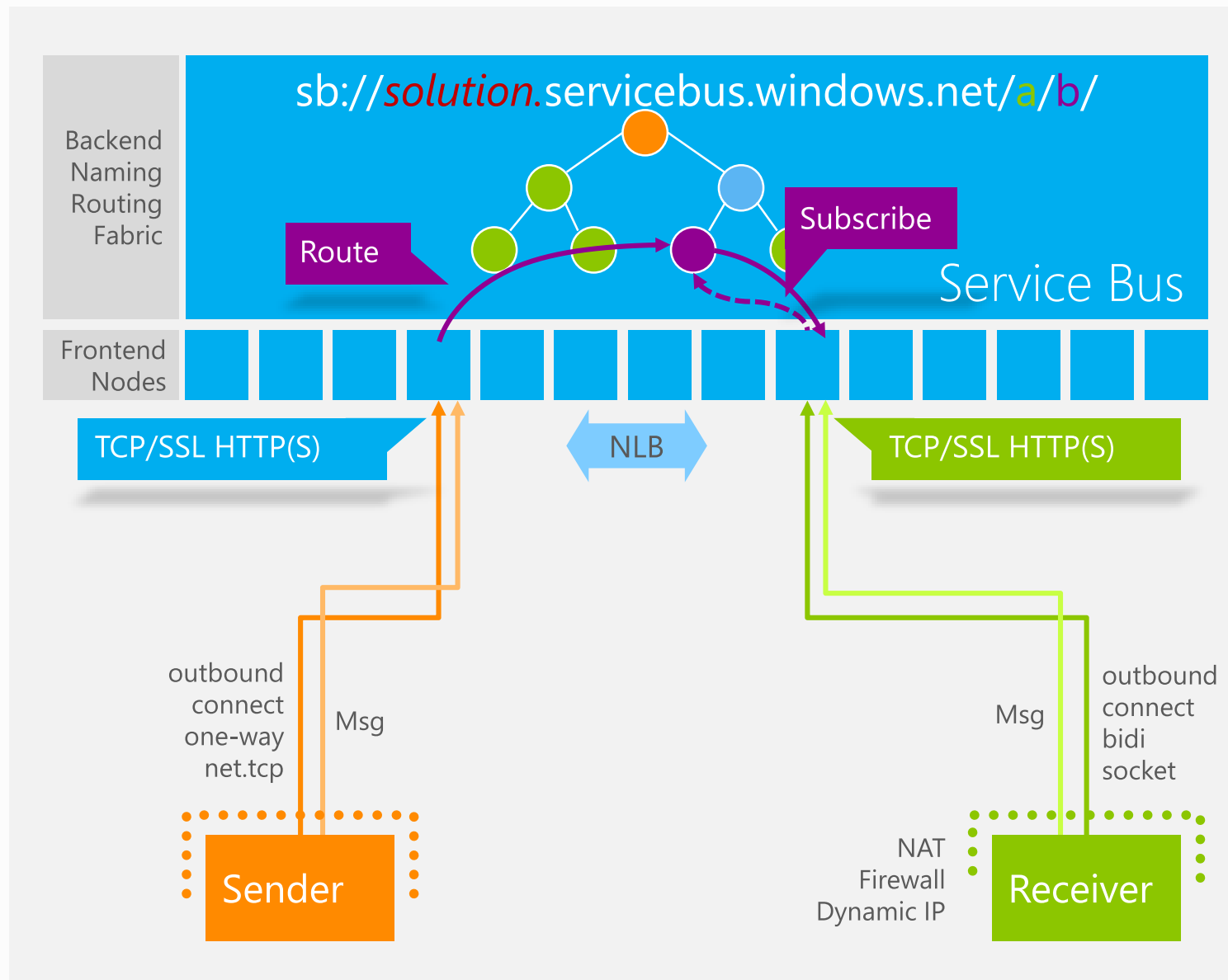
NetOnewayRelayBinding

所有TCP 和HTTP 监听使用
单向内部控制通道

60KB 消息大小

只能单向

没有会合的开销



消息

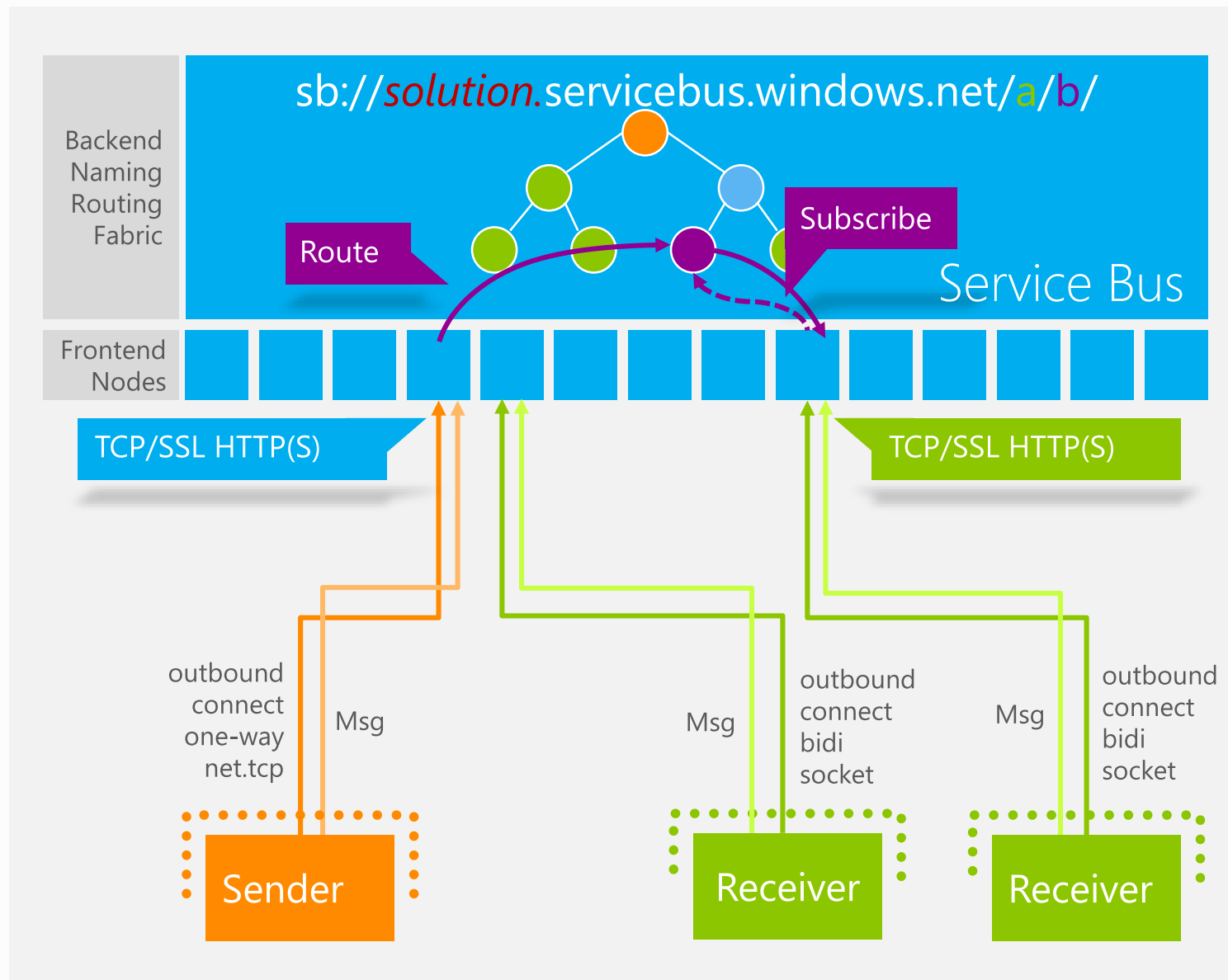
NetEventRelayBinding

小规模同步多播

60KB 消息大小

只能单向

没有会合的开销



Rendezvous 会合

(TCP & HTTP)

NetTcpRelayBinding

WebHttpRelayBinding

BasicHttpRelayBinding

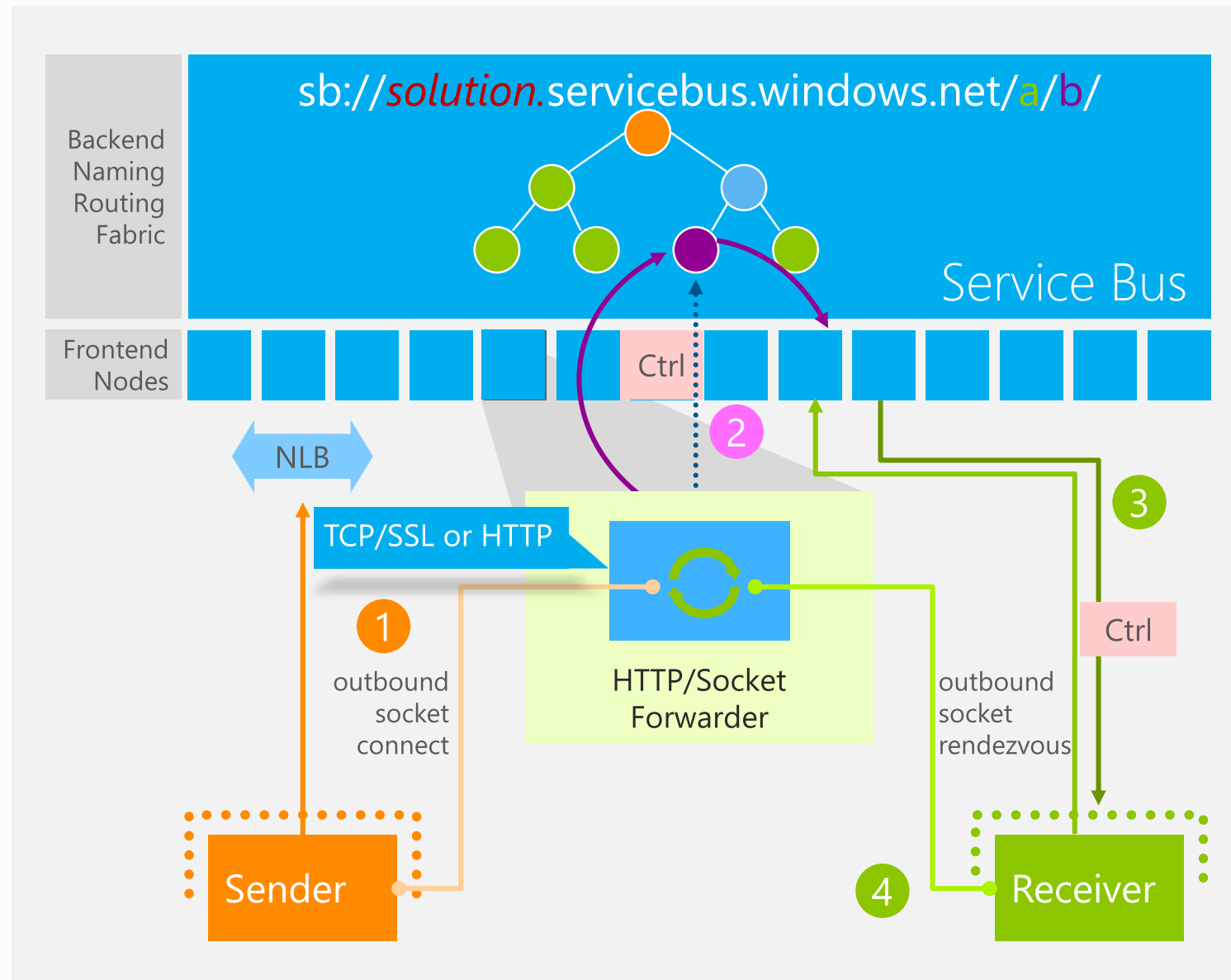
WS2007RelayBinding

Rendezvous 握手

双向

Net.Tcp 全双工

没有消息大小限制



混合连接

NetTcpRelayBinding的特殊模式

TcpRelayConnection-
Mode.Hybrid

作为中继连接开始

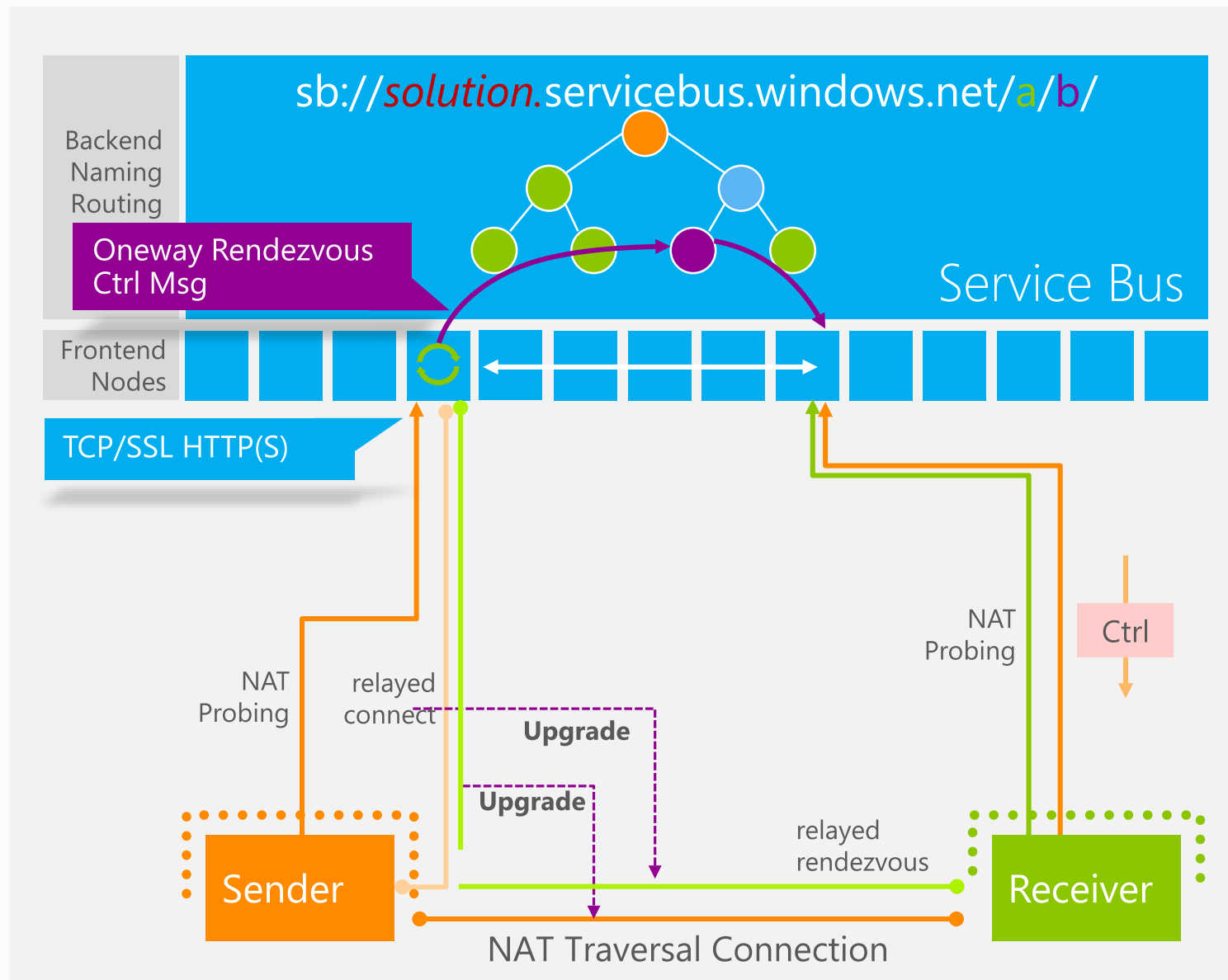
进行NAT探测和行为预测

创建直接连接和可能的升级

通过流量升级

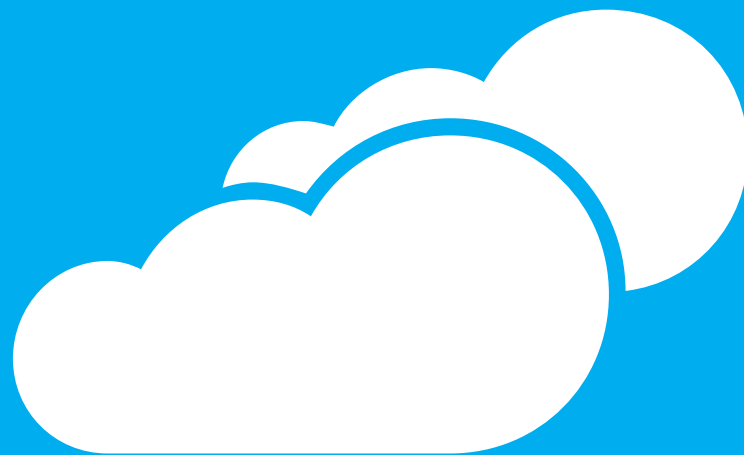
将中继上的传输降低

没有传输损失，低延迟



服务总线中继举 例

演示

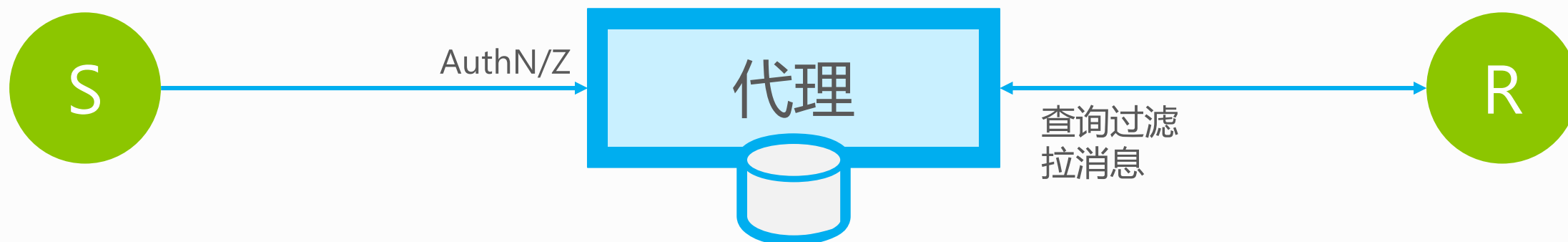


服务总线消息

中继 vs. 消息代理



中继直接将消息路由，并将反馈直接沿原路径返回到发送者

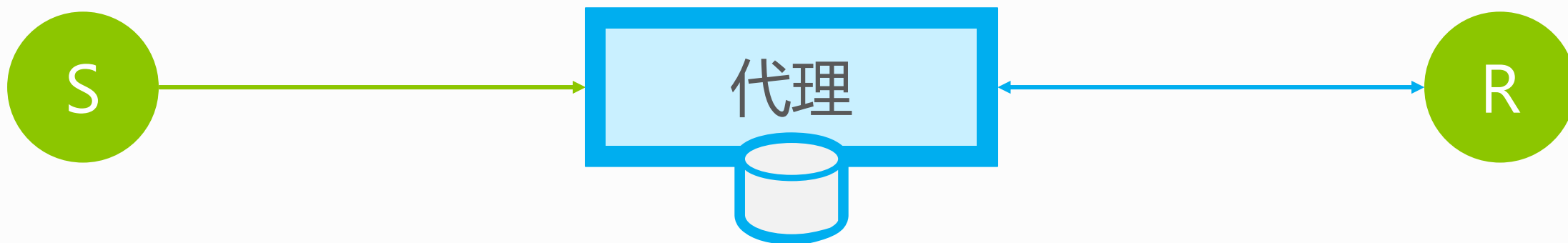


代理存储消息，可以重新获取或者队列化

推 VS. 拉



‘推’是发送者发起一个活动，导致接收者收到消息。接收者不会请求一条或者某条消息

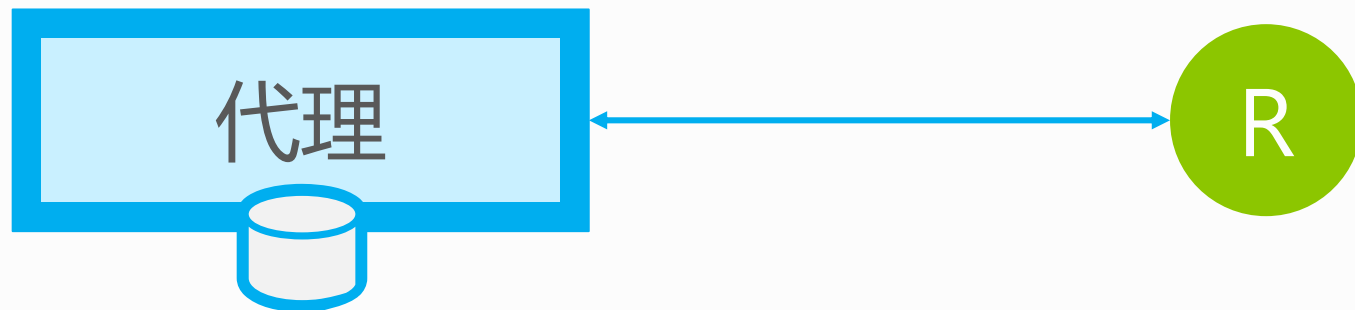


‘拉’是发送者发起一个活动，把消息存储在上下文中。这种上下文将消息发送的耦合操作解放出来。

拉的方法

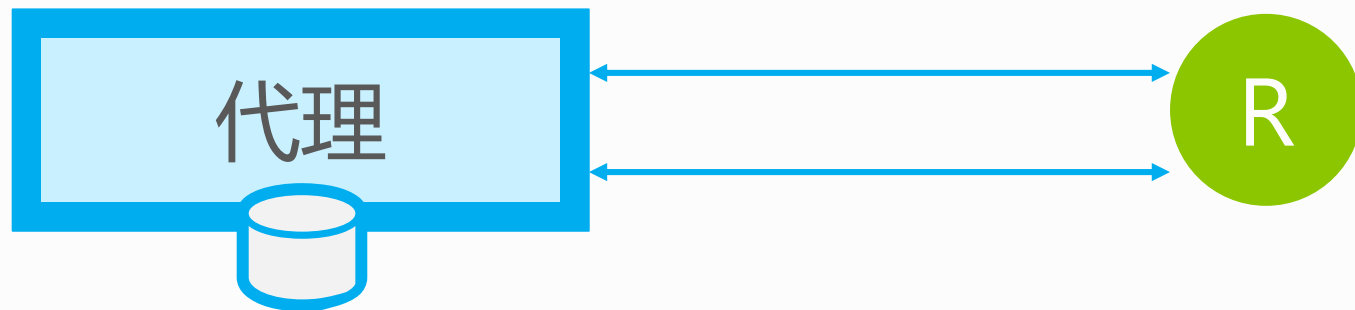
接收和删除

最快，如果接收者崩溃或者传输失败，那么消息丢失



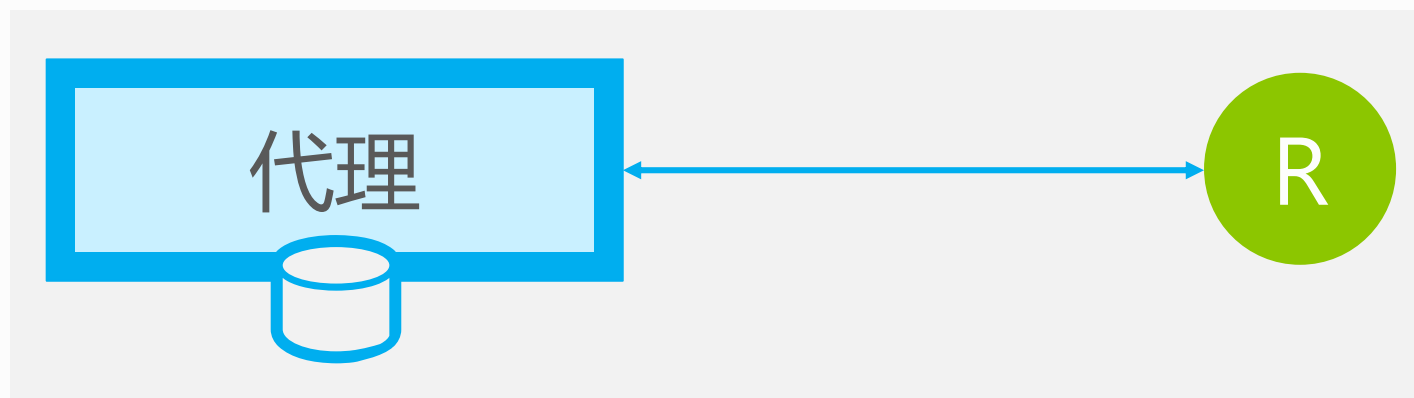
Peek 锁

消息在接收的时候锁定，在锁定超时内没有删除，则重新出现



事务的

本地模型



消息



代理消息属性不是SOAP的头

属性是key/value 对，能够非常好的进行有效载荷

消息有空的Body是很正常的

消息Body对于不透明的有效载荷是很有用的，对代理来说是未知的(内容加密)

队列



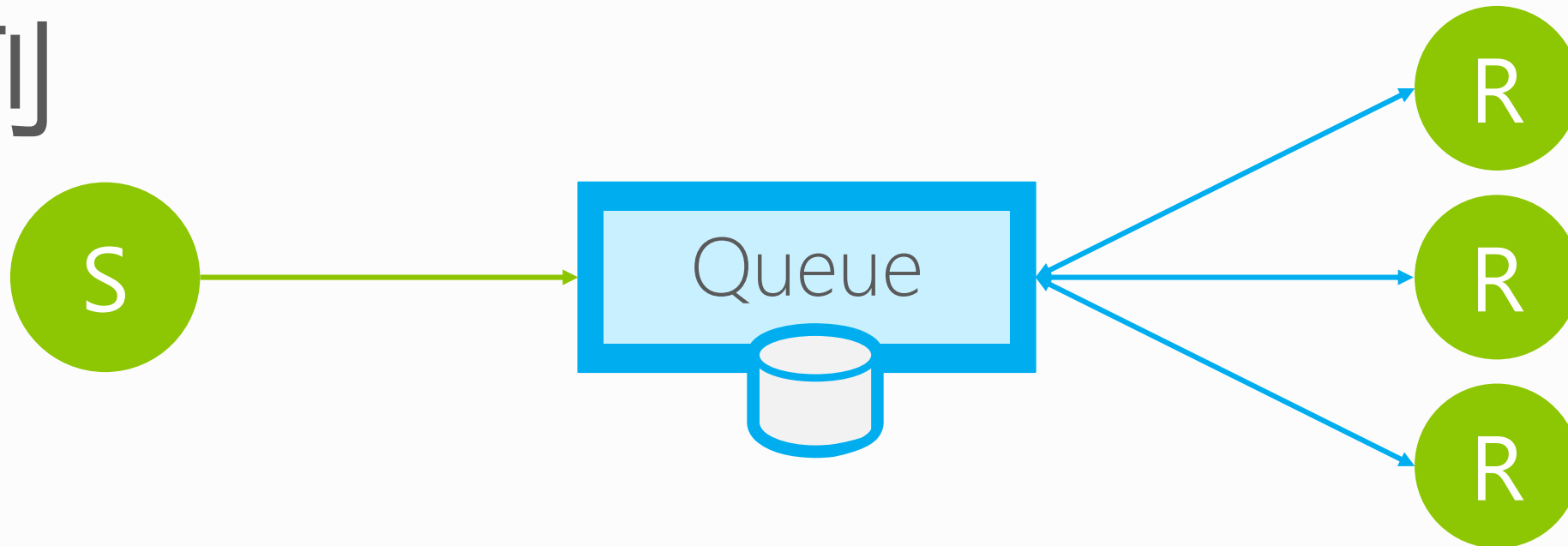
负载级别

接收者根据自己的节奏来接收消息，不会过载。当队列长度增加时，可以增加接收者。如果队列长度降低或者变成0可以减少接受者。优雅地处理流量的高峰，不会对后台造成压力。

断线/批处理

允许接收者因为某些原因下线，请求会被缓冲直到接受者重新可用。

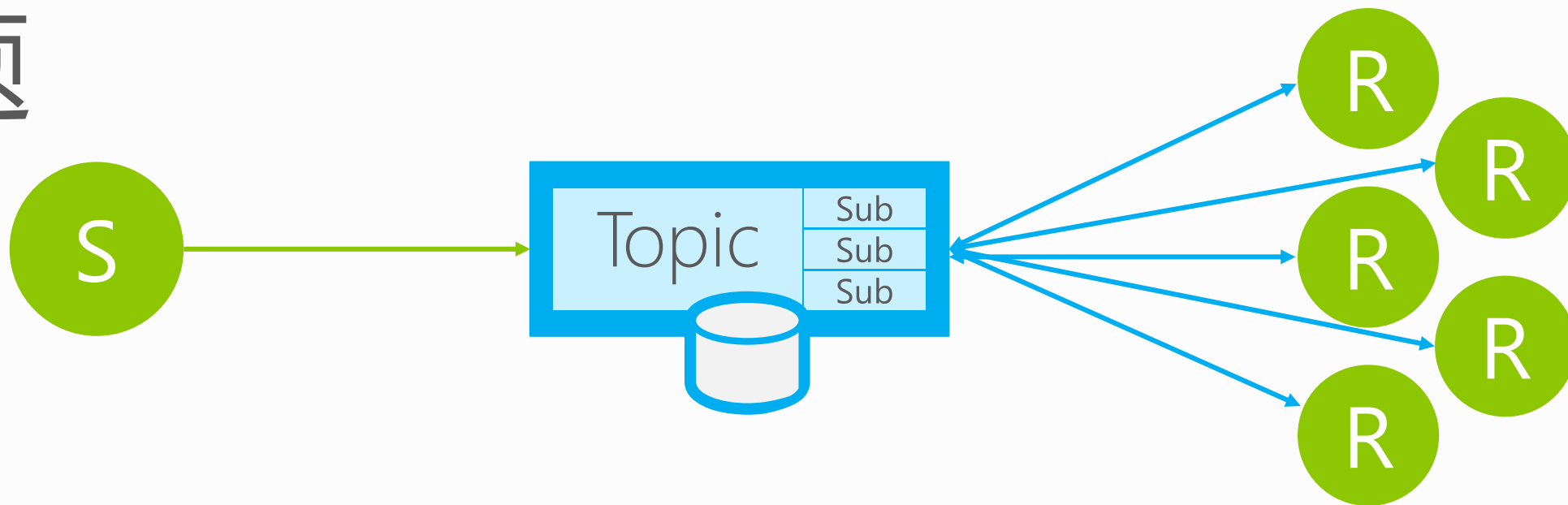
队列



负载均衡

多个接收者对同一个队列竞争获得消息（或者相同的订阅）。提供自动的负载均衡来志愿处理工作。通过对队列长度的观察来决定何时增加新的接收者

主题



消息分发

每个接收者获得自己的消息。订阅是独立的，允许多个独立的消息流，订阅者可以根据兴趣过滤。

有约束的消息分发

(Partitioning)

通过定义合适的过滤条件，接收者获得互斥的消息流的分片。

订阅过滤

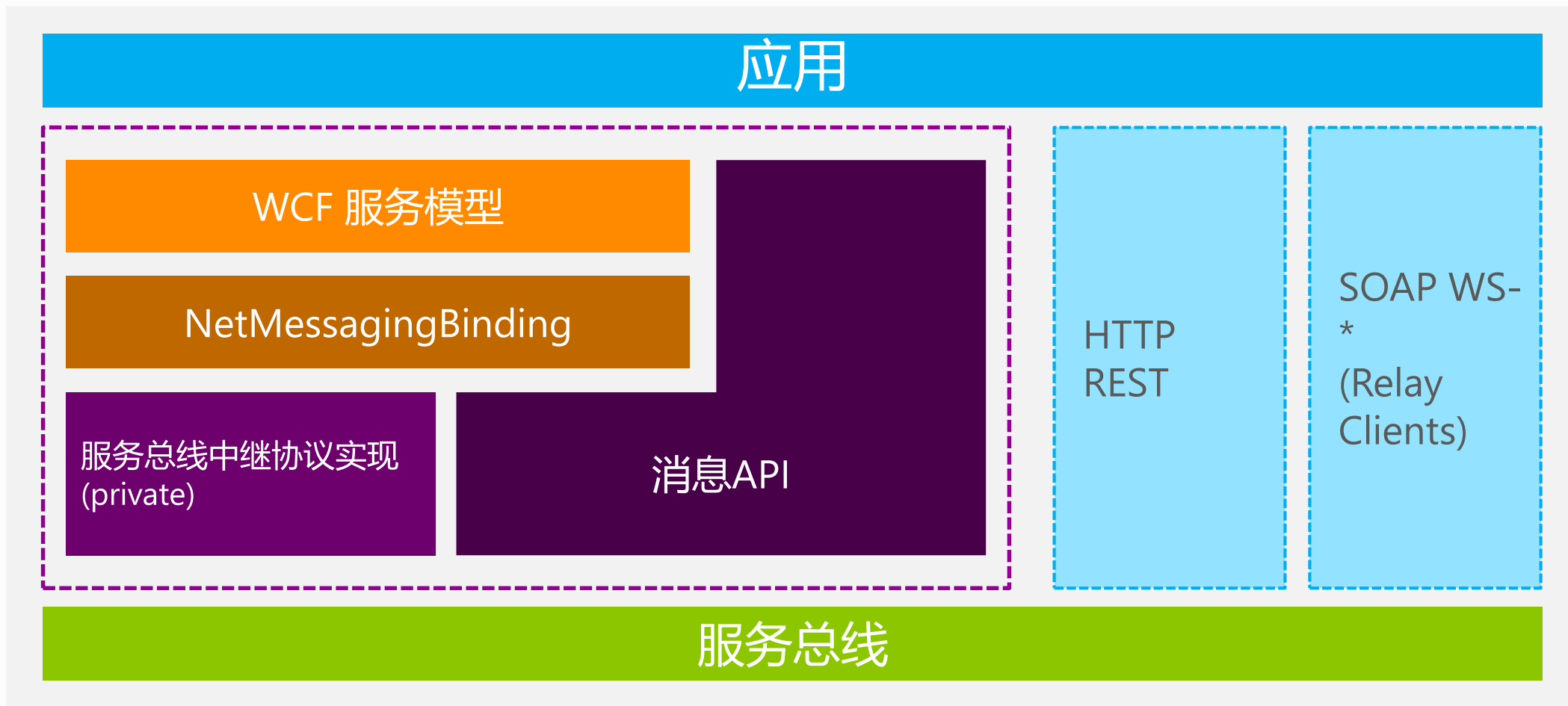
过滤条件在消息属性上进行操作，通过SQL'92语法进行表达

```
InvoiceTotal > 10000.00 OR ClientRating < 3  
ShipDestCtry = 'USA' AND ShipDestState='WA'  
LastName LIKE 'V%'
```

过滤行为可以修改/增加/删除属性

```
SET AuditRequired = 1
```

运行时API选择



Messaging API Hello World!

```
var nsm = NamespaceManager.Create();  
nsm.CreateQueue("newQueue");
```

1

```
var client = QueueClient.Create("newQueue");  
client.Send(new BrokeredMessage { Properties = {{ "Greeting", "Hello World!" }}});
```

2

```
var m = client.Receive();  
Console.WriteLine(m.Properties["Greeting"]);
```

3

```
<appSettings>  
  <add key="Microsoft.ServiceBus.ConnectionString"  
        value="Endpoint=sb://[your namespace].servicebus.windows.net;  
              SharedSecretIssuer=owner;SharedSecretValue=[your secret]" />  
</appSettings>
```

服务总线中继举例

- 
- 
- 

demo

工具的更新

服务器管理器

创建队列/主题

Examine queue/topic properties

Send test message

Receive message

角色模板

服务总线最佳实践

客户端对象生命周期管理

- 缓存QueueClient, SubscriptionClient, TopicClient
- 不需要以后关闭客户端. Close() 可能有异常, 使用try/catch.

处理传输错误

- 实现持续的重试机制
- 考虑传输失败管理框架

可依赖的消息处理 (Peeklock)

- 总是终结成功处理的消息, 调用Complete()
- 总是放弃没有处理过的消息, 调用Abandon()
- 保证消息在设计超时内处理

服务总线最佳实践

提高性能

- 重用client objects
- 选用Service Bus client protocol over HTTP
- 使用异步send/receive
- 在合适的时候使用
- 客户端批处理（异步方法）
- 批处理内部存储访问（EnableBatchedOperations = true）
- 预取(Prefetching)
- 使用多个队列



© 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](#)