**架构是业务数据与业务逻辑之间的折衷关系**

**Client/Server结构**

进程间通信IPC

**两层C/S**

前台客户端 + 后台数据库管理系统

**B/S（三层C/S）**

Browser - Server - Database

**四层C/S**

前台界面 - Web服务器 - 中间件/应用层 - 数据库

**MVC(多层C/S的一种标准化形式)**

视图层View <-> 控制器Controller <-> 模型Model

**SOA(Service-Oriented Architecture)**

面向服务的系统间服务通信机制，如：Web Service，两个门户网站之间相互开放一些内部“服务”。服务消费者Consumer、服务提供者Provider。

**ESB(Enterprise Service Bus)**

企业服务总线，常见于银行。基于中间件构建的连接器Connector，处理和分发data、request、response。

**EAI(Enterprise Application Integration)**

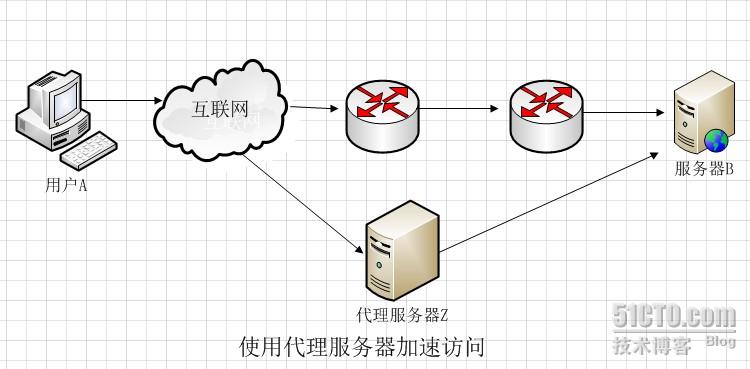
建立底层结构，贯穿所有系统，实现互连互通

**两地三中心**：总行IDC、同城灾备IDC、异地灾备IDC

**网站处理模式**:每进程/每请求，每线程/每请求，Nginx异步事件分发

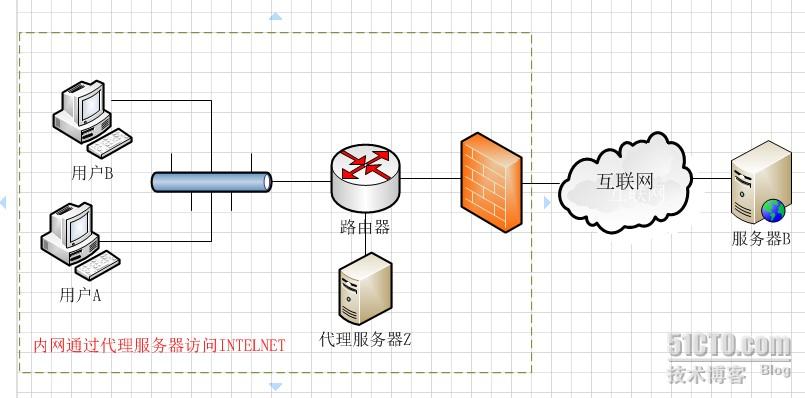
**代理Proxy：网络的中介节点**

1.翻墙或减少路由节点



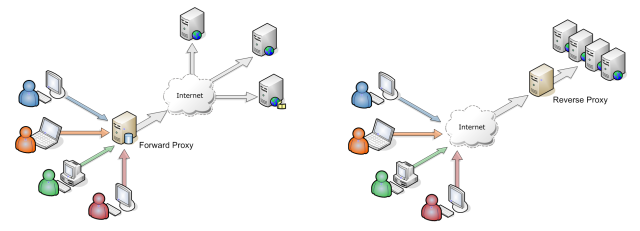
2.用于缓存加速访问

3.企业中，外网访问控制

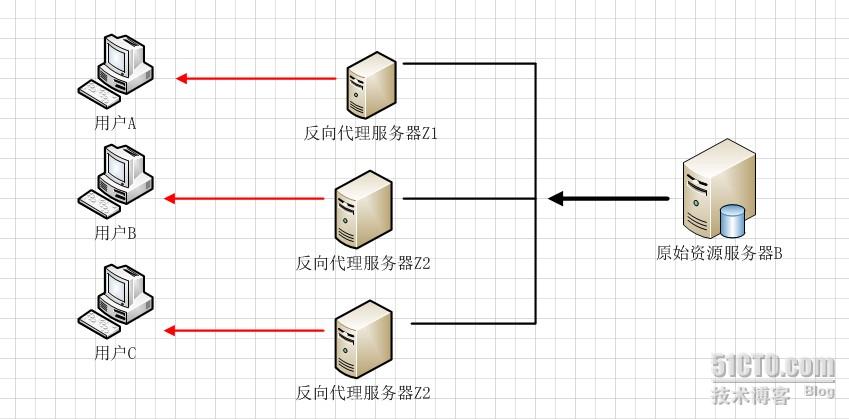


4.隐藏IP等痕迹

**反向代理Reverse Proxy**

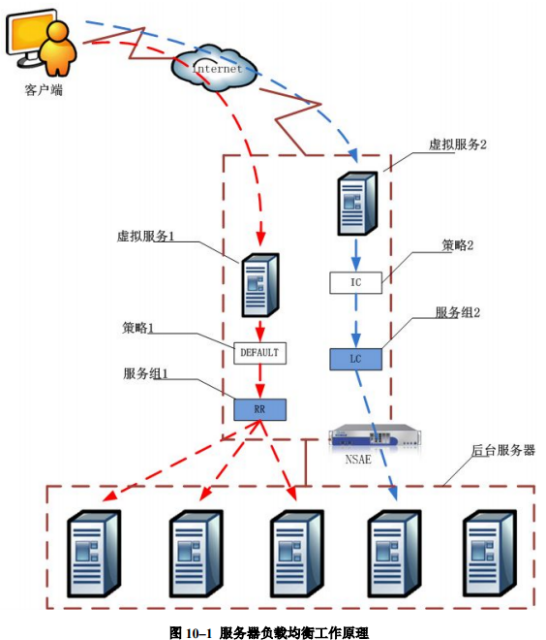
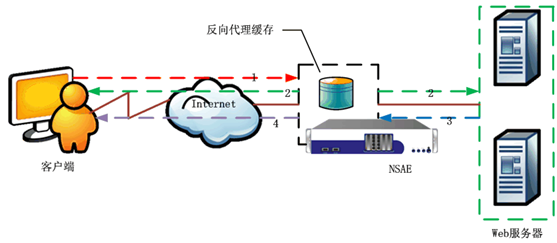


1.负载均衡



2.Content Delivery Network

**负载均衡Server Load Balancer**



服务器负载均衡旨在把网络流量和负载均衡分配到服务器组中的各个服务器上。

有缓存，返回结果。没缓存，分发请求。

健康检查：定期模拟请求

**单点登录Single Sign On(in Identity and Access Management)**

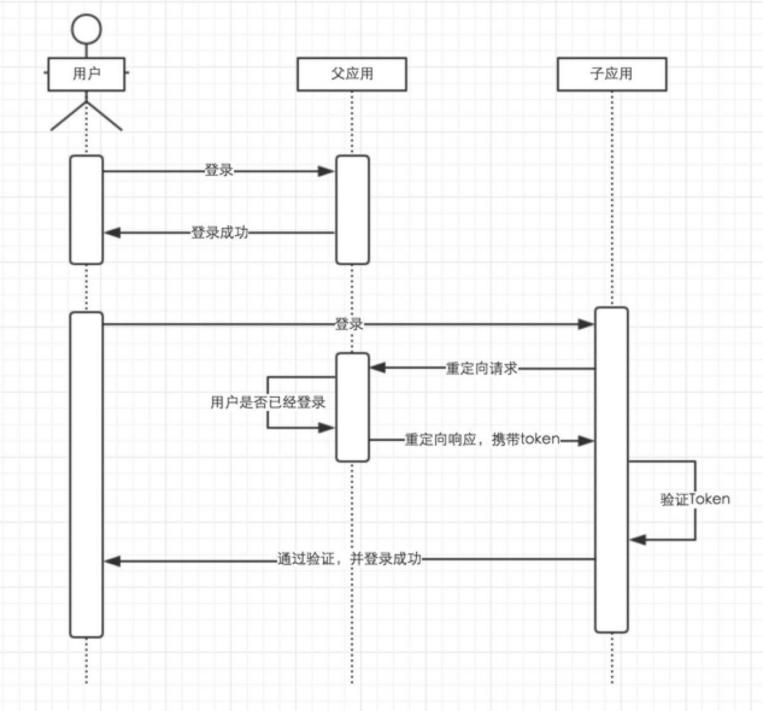
多系统应用群中登录一个系统，便可在其他所有系统中得到授权而无需再次登录。

HTTP: Cookie机制是在客户端保持状态 ，Session机制是在服务器端保持状态。

实现：存储信任、验证信任(安全与方便的矛盾)

1. 加密的Cookie,不能跨域名

2. 页面重定向(重定向到独立的登录系统?)



**串口通讯**

单工：数据单向传输，如：PC->打印机

半双工：同一根线，既可发送亦可接收，但不能同时，如：对讲机

双工：双向同时传输，如：电话