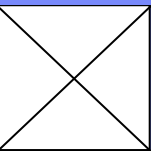


Advanced web technology

# Web高级开发与应用技术

web核心标准与架构

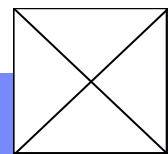


# 什么是Web?

**Web**是分布在全世界的基于**HTTP**通信协议的服务器(**Web服务器**)中所有互相连接的超文本集(**Web文档**)，它采用客户/服务器模式并使用超文本技术链接**Internet**上的信息和资源。服务器端存放用**HTML**编写的网页以及其他资源，客户机通过浏览器(**IE**， **Netscape etc.**) 可以访问全球范围内各个主机上的这些信息资源。

## ■基本特征

- 使用**HTML**技术来创建客户端界面
- 使用**HTTP**协议来传输内容
- 基于**B/S**模式
- URI**标识资源

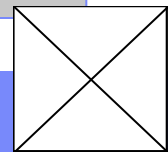


## Internet与Web的发展历程

- Internet起源于美国的ARPANet (1969年问世), 1984年ARPANET分解成两个网络。一个网络仍称为ARPANET, 是民用科研网。另一个网络是军用计算机网络MILNET。
- 汤姆.李提出了超文本(Hyper Text)的概念并创造出超文本标记语言(Hyper Text Markup Language, 简称HTML),同时开发出传输这种语言的协议HTTP
- 美国加州的大学生马克.安德森推出浏览器程序Mosaic
- World Wide Web(简称WWW, 或简称Web) 构成了世界上最大的电子信息仓库,有人称其为“万维网”

Web的主要目的是旨在成为一种共享的信息空间(a shared information space), 人们和机器都可以通过它来进行沟通。

-Berners-Lee



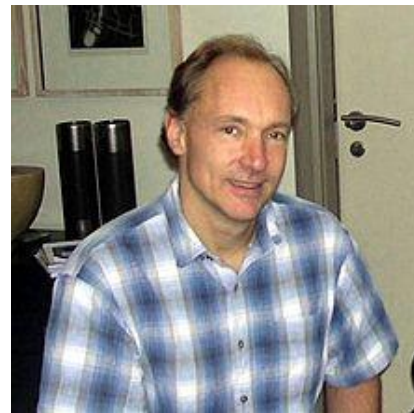
## Web 和 Internet的区别

- Internet是众多相互联结的计算机和其他设备的集合
- Web是建立在Internet上的文档、软件以及一些协议的集合
- 在一个抽象的意义上，Web是一个文档的巨大集合，这些文档之间通过链接相连，可以通过Web浏览器进行访问

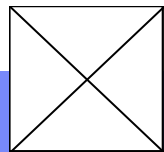
**Internet和Web是两个完全不同的概念。  
Internet是Web的基础平台，Web是  
Internet平台上的一种应用层服务**



**Vint Cerf 1973发明Internet**



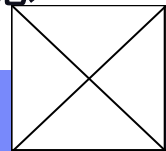
**Berners-Lee 1989发明web**



# Web 的特点

---

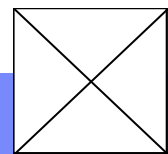
- **Web是一种分布式超媒体系统**
  - 超文本/超媒体/链接
- **Web是图形化的和易于导航的**
  - 只需从一个链接跳到另一个链接就可在各页各站点间进行浏览
- **Web与平台无关**
  - 从Windows、UNIX、Macintosh等都可以访问Web
- **Web是分布式的**
  - 物理上不一定在一个站点的信息在逻辑上一体化，从用户来看这些信息是一体的
- **Web 是动态的**
  - 信息的提供者可以经常的对站上的信息进行更新
- **Web是交互的**
  - 用户通过填写FORM可以向服务器提交请求，服务器可以根据用户的请求返回相应信息



# Web带来的革命

---

- 提供了全球的链接
- 为商业竞争创造了全新的竞争领域
- 用户可以掌握一切，扩大了客源。（客户自助服务）
- 提供了一种更新、更好的方式来完成已有的服务
- 基于Web的信息交换改善了商业伙伴之间的合作，优化物流、简化书面工作、减少供应链开销
- 使用内部互联网(Intranet)来共享信息，优化内部流程，在竞争中争取主动



下一个发展方向是什么？

**Service Web**

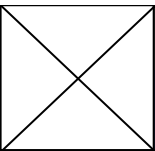
- ◆ **Web Services**
- ◆ **Generate XML**
- ◆ **SOAP, WSDL, UDDI**
- ◆ **Transactions initiated by program**

**Application Web**

- ◆ **Application Servers**
- ◆ **Business Logic**
- ◆ **Generate HTML**
- ◆ **Transactions**
- ◆ **Distributed process**
- ◆ **WAP, WML**

**Document Web**

- ◆ **Web Server**
- ◆ **HTTP协议**
- ◆ **HTML文档**



# 什么是Web-based Application?

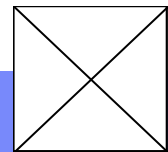
---

## ■ 基于Web的应用程序

- 以浏览器(比如IE)作为用户界面的窗口
- 以可交互的网页形式提供用户界面
- 网页由HTML语言写成
- 提供与普通windows application类似的功能
- 使用browser-server架构
- browser与server之间通讯基于http协议

谷歌CEO埃里克·施密特(Eric Schmidt):浏览器将成未来应用软件发展的基石

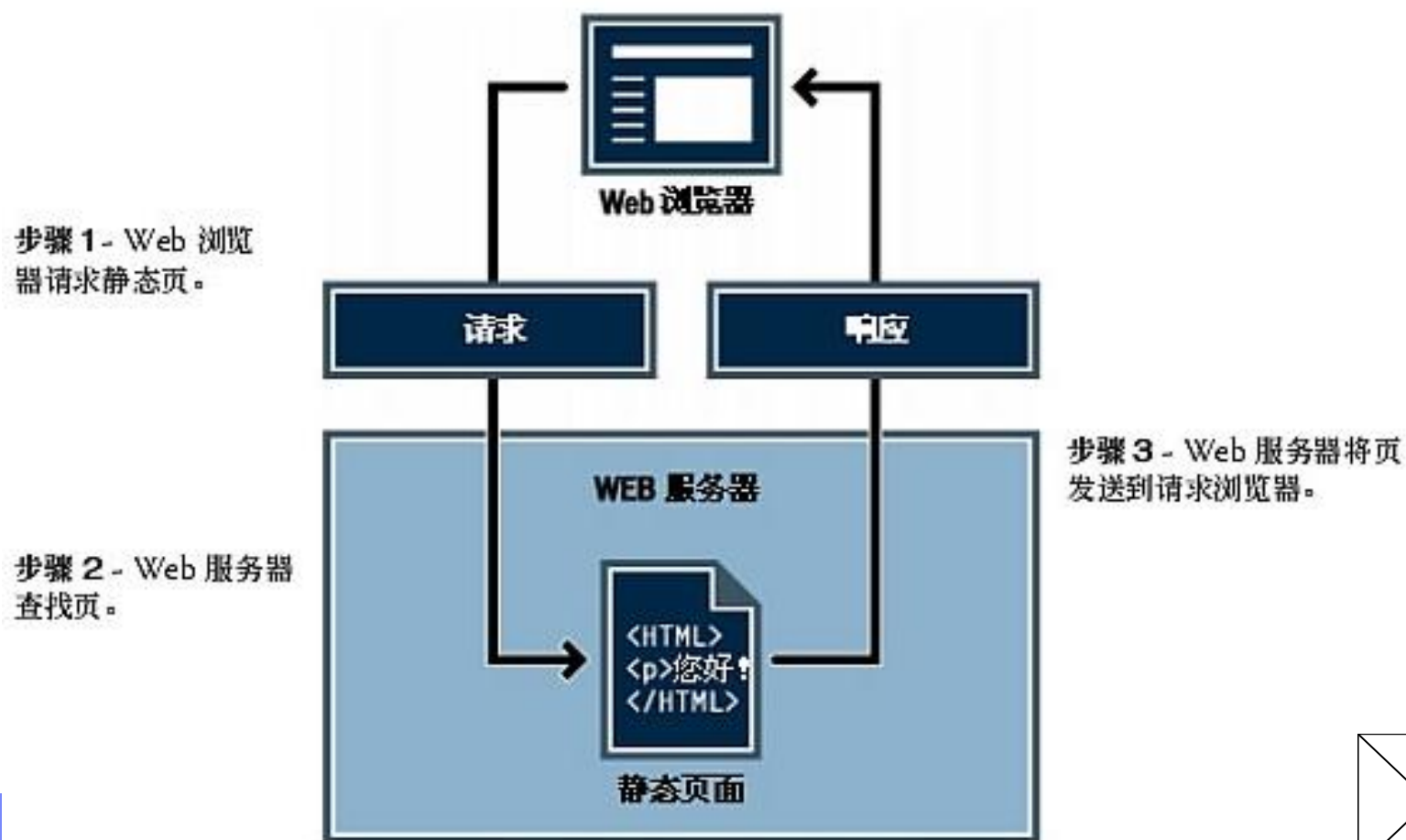
-----谷歌I/O 2009



# Web应用程序如何工作

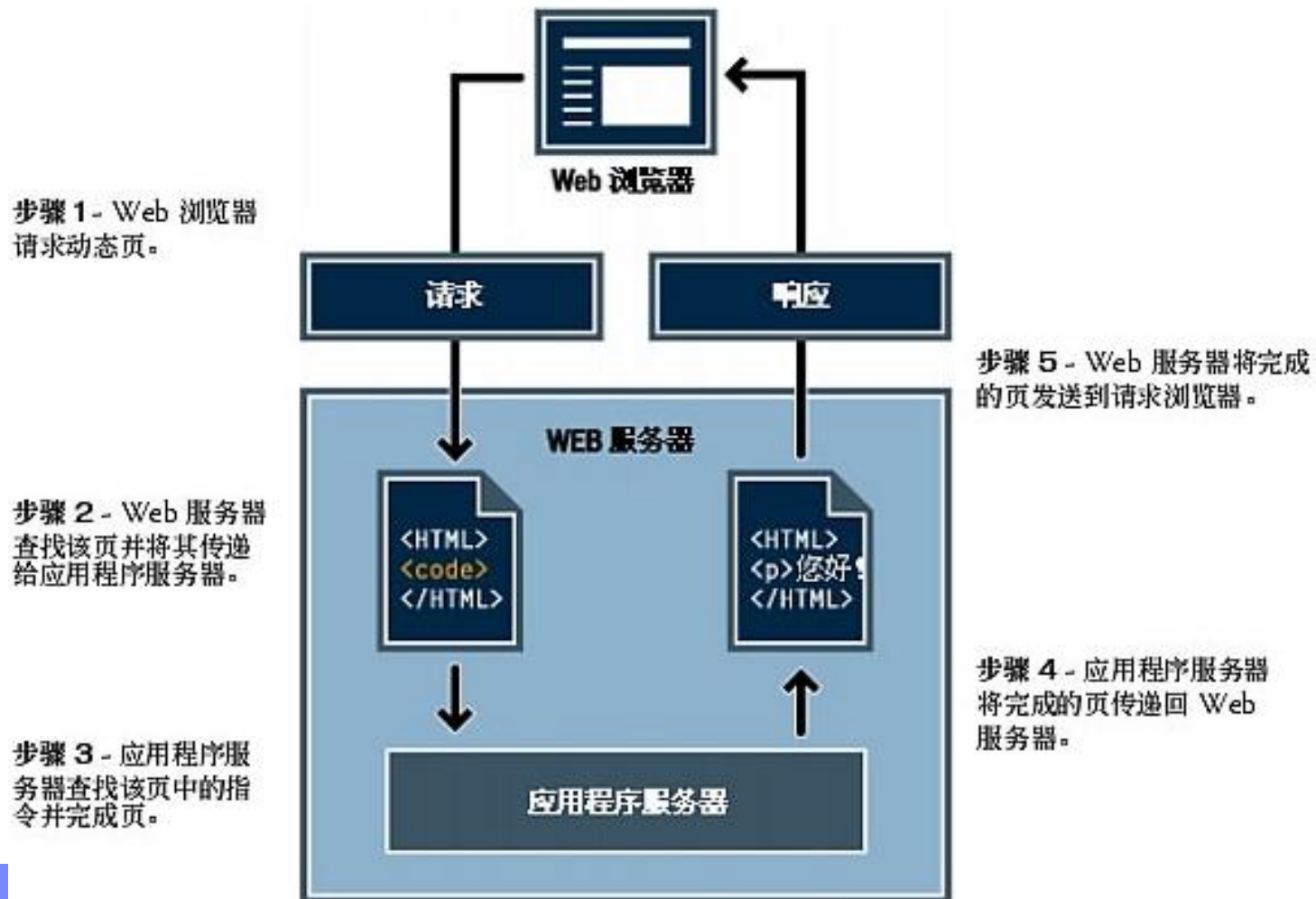
## ■ 处理静态 Web 页

- 静态页是在发送到浏览器时不进行修改的页



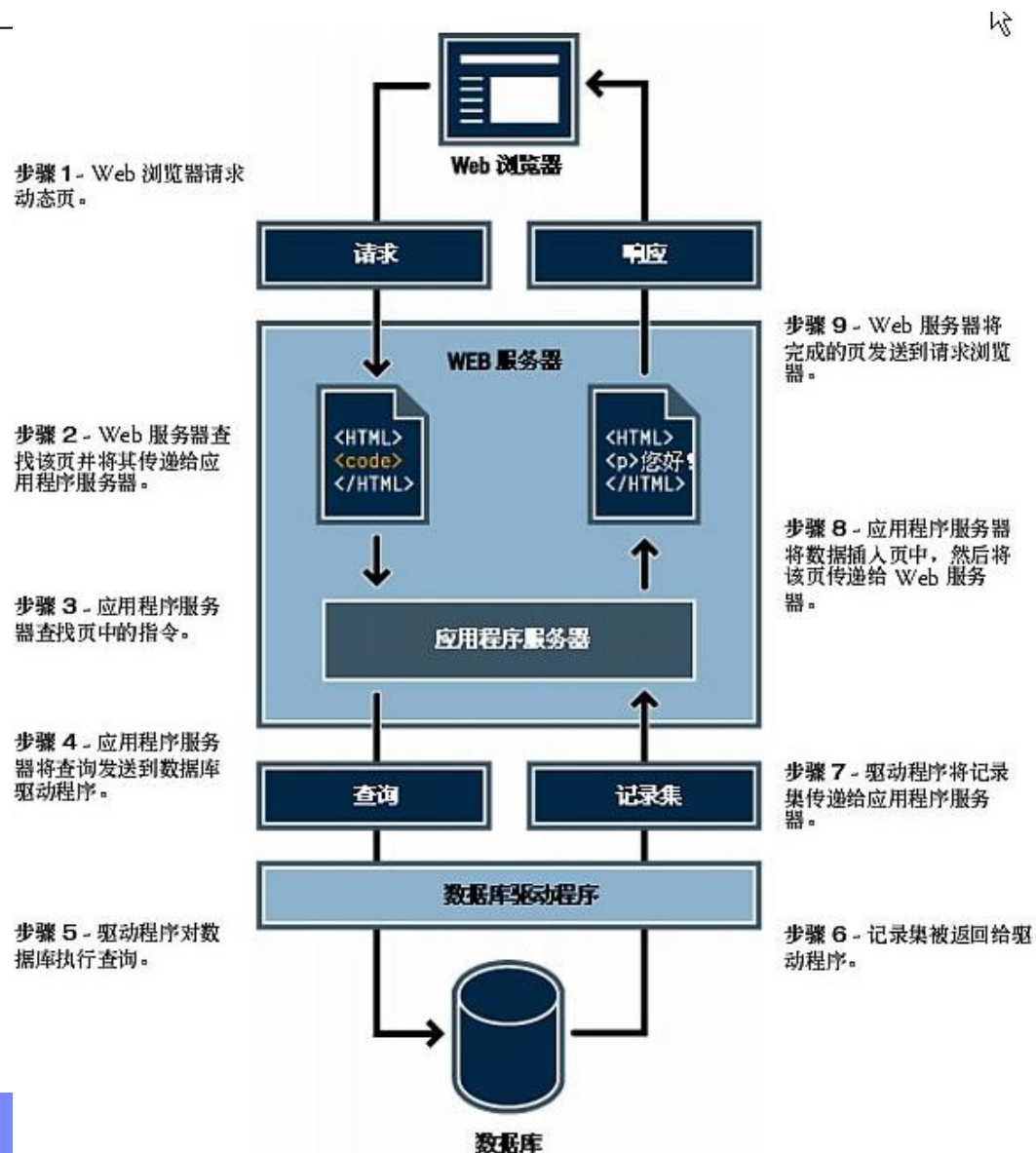
# Web应用程序如何工作

## ■ 处理动态 Web 页



# Web应用程序如何工作

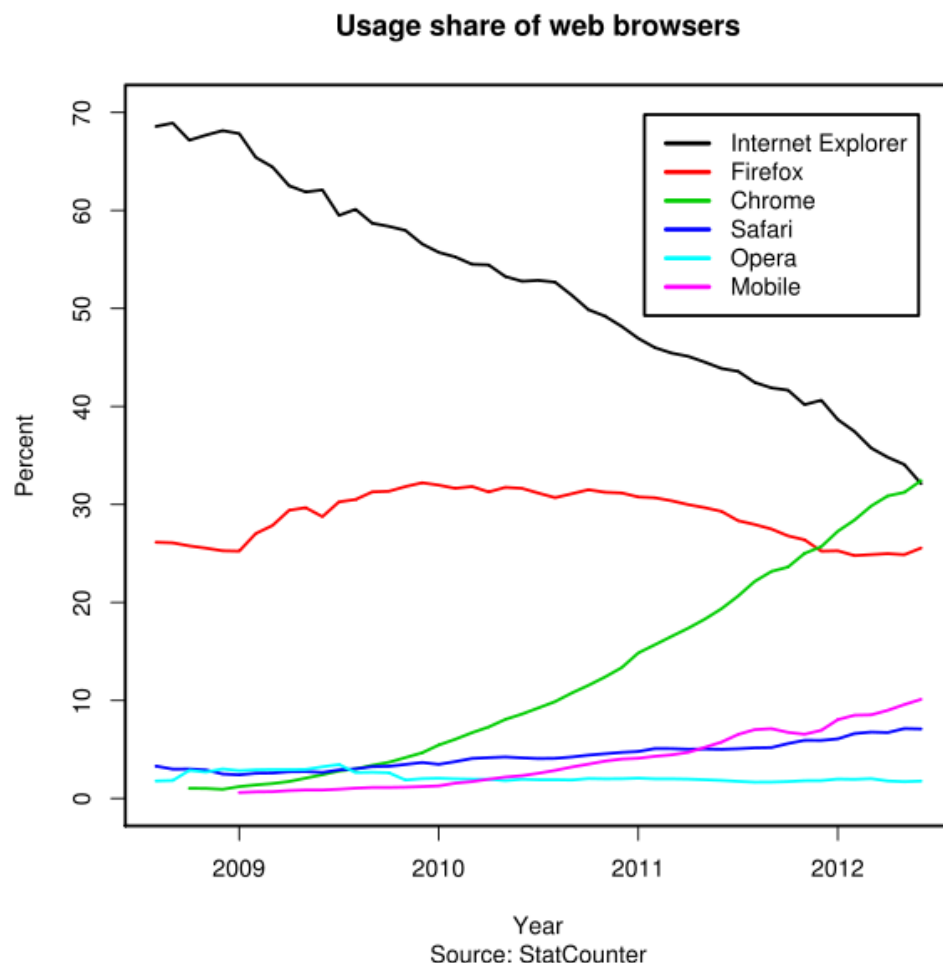
## ■ 访问数据库



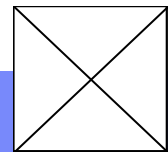
# 浏览器演变

- 1995 至 1998 年, IE vs. Netscape
- 2005 年到 2007 , IE vs. Firefox
- 2012, Chrome份额超过IE

Web即计算机，浏览器即桌面



On the Choice of Browser and Numerical Intelligence

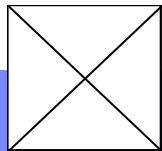


# HTML

---

## ■ HTML5

- 2008 年 W3C 制定出第一个 HTML5 草案, W3c Working Draft:  
<http://www.w3.org/TR/html5/>
- 广义上是包含了相应的javascript和CSS3的标准集
- 取代1999年所定订的HTML 4.01和XHTML 1.0 标准, 以期减少浏览器对于需要插件的丰富性网络应用服务的需求, 并且提供更多能有效增强网络应用的标准集。
- 强化了Web网页的表现性能, 提供了绘制、视频和音频工具,追加了本地数据库等Web应用的功能
- 促进了web上的和便携式设备的跨浏览器应用的开发。HTML5是驱动移动云计算服务方面的发展的技术之一
- 浏览器支持程度评测网站: <http://html5test.com/>



# HTML

## ■ HTML5新增标签：增强语义

**article**: 文章

**aside**: 内容旁边的侧边栏内容

**audio**: 音频

**canvas**: 2D 绘图

**command**: 命令按钮

**datalist**: 下拉选择框

**details**: 对象的细节

**dialog**: 对话框

**embed**: 外部插件或对象

**figure**: 一组媒体对象以及标签文字

**footer**: 页脚

**header**: 页首

**hgroup**: 文档某一部分的信息

**keygen**: 表单生成的 Key

**mark**: 标注的文字

**meter**: 预先定义的范围内的度量

**nav**: 导航条

**output**: 输出

**progress**: 进度条

**rp**: 标识 ruby 内容

**rt**: ruby 内容的解释

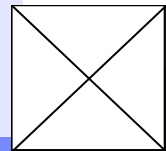
**ruby**: ruby 内容

**section**: 定义一个部分

**source**: 媒体的资源

**time**: 日期时间

**video**: 视频

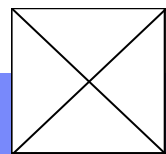


# HTML

---

## ■ HTML5特征

- 后台异步化、线程化操作的Web Worker, 使用postMessage和onmessage等方法,
- 在本地应用和服务端之间建立持续连接的WebSocket接口
- 对使用CSS3来管理GUI的支持, 这意味着HTML5可以是面向内容的
- 使用window.localStorage支持客户端的本地存储(大小限制5MB, HTTP Cookie 大小4KB)
- 使用 Web SQL Databases 支持SQL规范的本地数据库
- 画布Canvas(支持JS在之上绘图)和音视频支持 (audio\video), 可在无需安装第三方插件的情况下添加图形和视频

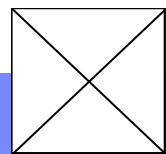


# HTML

---

## ■ HTML5特征

- Geolocation API规范，支持共享地理位置；通过使用智能手机定位功能来纳入移动云服务和应用
- 增强型的表单，其降低了下载JavaScript代码的这种必要性，允许在移动设备和云服务之间进行更多高效的通信。
- 作为HTML5中的 `<canvas>` 标签的一个特殊的上下文（experimental-webgl）实现在浏览器中的 WebGL
- 通过ApplicationCache接口使离线存储成为可能，离线存储使得你的web应用可以在用户离线的状况下进行访问
- 使用 **Notifications api 桌面提醒接口** 支持**Web通知**；提供了与本地文件交互的标准方法：File API规范；支持拖曳（Drag & Drop）



# HTML

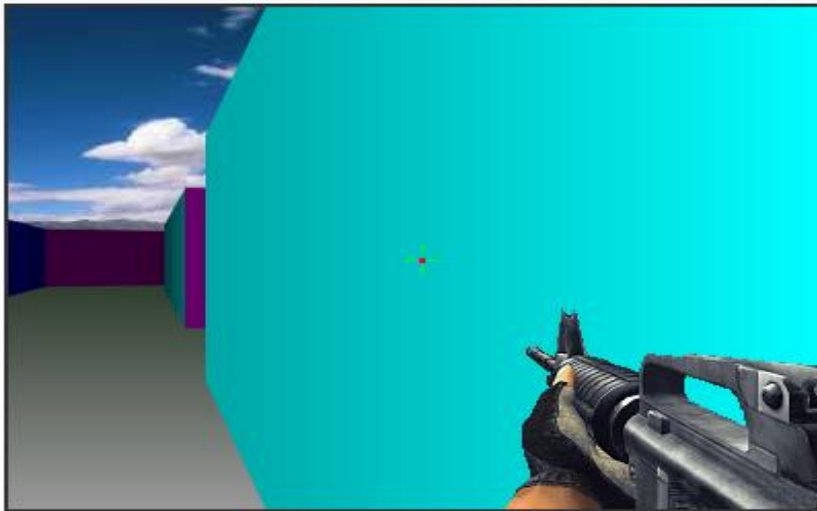
## ■ HTML5示例

- Canvas: <http://www.benjoffe.com/code/demos/canvascape/>
- Websocket: <http://demo.kaazing.com/livefeed/>

### Canvascape - "3D Walker"



[code.js](#)



### Real-time is instantly refreshing

Hello Palo Alto | Saturday September 08 | 5:10:25 PM | [Google Talk Account](#) |  Password | [About](#)

Currency Exchange

GBP / USD		17:10:18
SELL	1.62	38 <sup>5</sup>
BUY	1.62	43 <sup>5</sup>

EUR / USD		17:10:23
SELL	1.40	33 <sup>2</sup>
BUY	1.40	36 <sup>2</sup>

Grounded Since Victory, While Loser Has Soared  
Since Virginie Razzano upset Serena Williams in the French Open, the 29-year-old Frenchwoman has won just one tour-level match while Williams has won Wimbledon and two Olympic gold medals.  
By CHRISTOPHER CLAREY

Time-based Advertisement  
ARCHITECTURE. MOBILE. WEB APPS. COMMUNICATION. BROWSER SUPPORT. BUSINESS IMPACT.

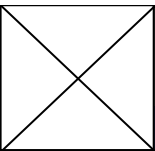
# HTML5 LIVE

Server Log

- Google from Mountain View, CA, United States is accessing Kaazing forums
- "VietNam Data Communication Company (VDC)" from Hanoi, Viet Nam is downloading Kaazing products
- "VietNam Data Communication Company (VDC)" from Hanoi, Viet Nam is accessing Kaazing forums
- "VietNam Data Communication Company (VDC)" from Hanoi, Viet Nam is visiting Kaazing.com
- "VietNam Data Communication Company (VDC)" from Hanoi, Viet Nam is reading Kaazing product info
- "Amazon.com" from Ashburn, VA, United States is visiting Kaazing.com
- "Amazon.com" from Ashburn, VA, United States is visiting Kaazing.com
- "VietNam Data Communication Company (VDC)" from Hanoi, Viet Nam is visiting Kaazing.com

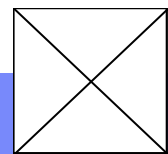
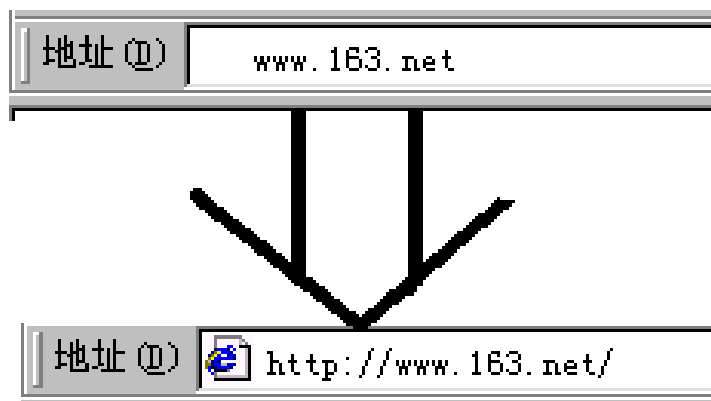
Keep the conversation going in real-time.

Try our full featured web-based chat client.  
Log on now »



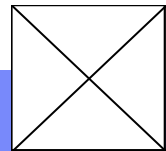
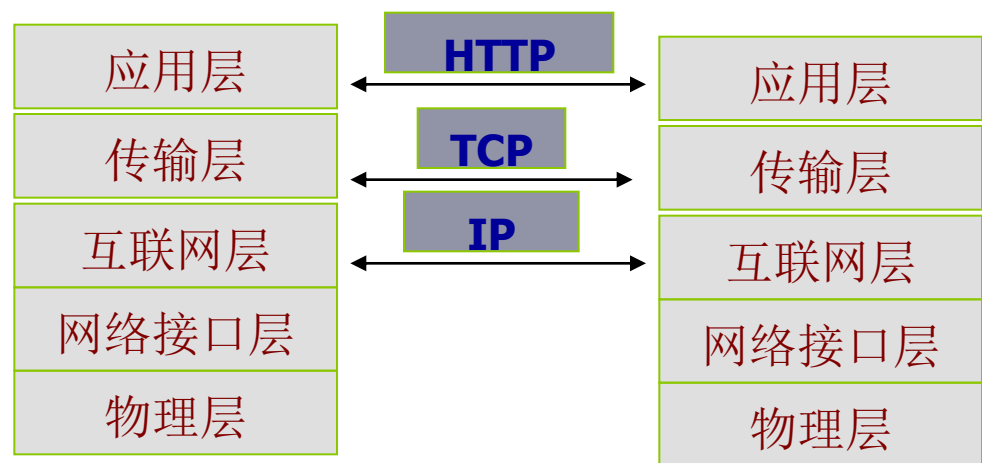
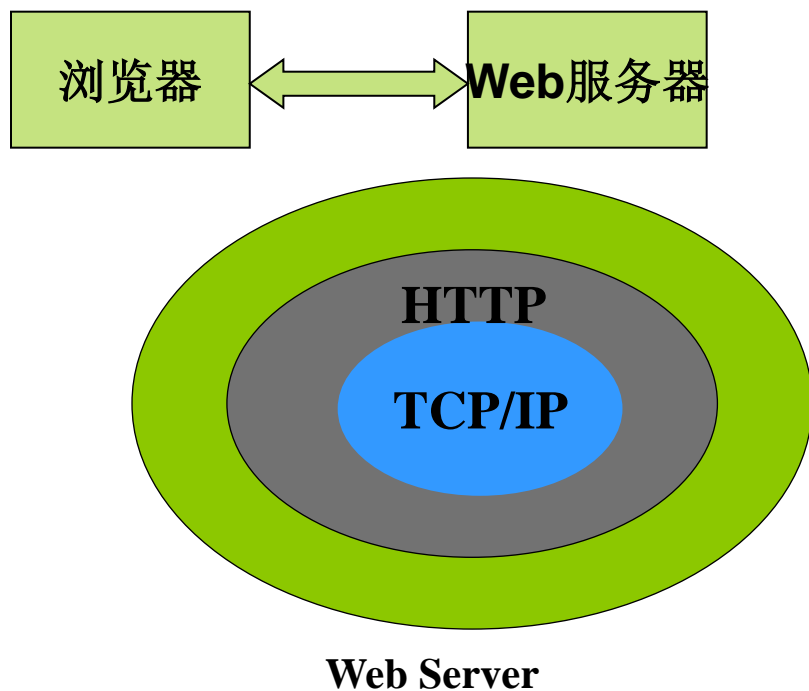
# HTTP协议

- Web服务器和客户端浏览器之间通过HTTP协议进行通信
- HTTP代表HyperText Transport Protocol（超文本传输协议）
- HTTP的两个重要功能
  - 传输文件
  - 实现动态交互应用



# HTTP协议

- HTTP是一种以TCP/IP通信协议为基础的应用协议
- HTTP协议采用“客户/服务器”（C/S）机制，也称之为“浏览器/服务器”（B/S）机制
- 无状态协议：与客户端之前请求的信息不做记忆



# HTTP请求

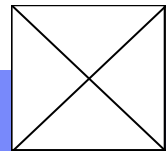
---

- HTTP请求由三个部分构成，分别是：

- 请求方法 URI 协议/版本
- 请求头 (Request Header)
- 请求正文

```
POST /sample.jsp HTTP/1.1
Accept: image/gif, image/jpeg, */*
Accept-Language: zh-cn
Connection: Keep-Alive
Host: localhost
User-Agent: Mozilla/4.0 (compatible; MSIE 5.01; Windows
NT 5.0)
Accept-Encoding: gzip, deflate

userName=kaiyu&password=1234
```



# HTTP响应

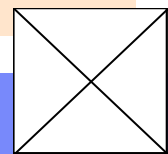
和HTTP请求相似，也由三个部分构成：

- 协议 状态代码 描述
- 响应头（Response Header）
- 响应正文

```
HTTP/1.1 200 OK
Server: ApacheTomcat/5.0.12
Date: Mon, 6 Oct 2003 13:13:33 GMT
Content-Type: text/html
Last-Modified: Mon, 6 Oct 2003
13:23:42 GMT
Content-Length: 112
```

```
<html>
<head>
<title>HTTP响应示例</title>
</head>
<body>
  Hello HTTP!
</body>
</html>
```

Demo of firebug,chrome



# Web应用演变

- 静态文本信息浏览和发布
- 静态多媒体信息浏览和发布
- 动态、交互式信息浏览和查
- 基于web技术的应用程序
  - 电子商务
  - 电子政务
  - 基于web技术的工作环境.....

21世纪要么电子商务, 要么无商可务  
——比尔·盖茨

**P2P**

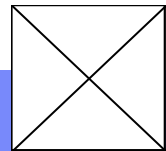
(Person  
To  
Person)



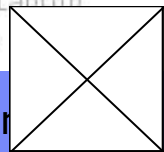
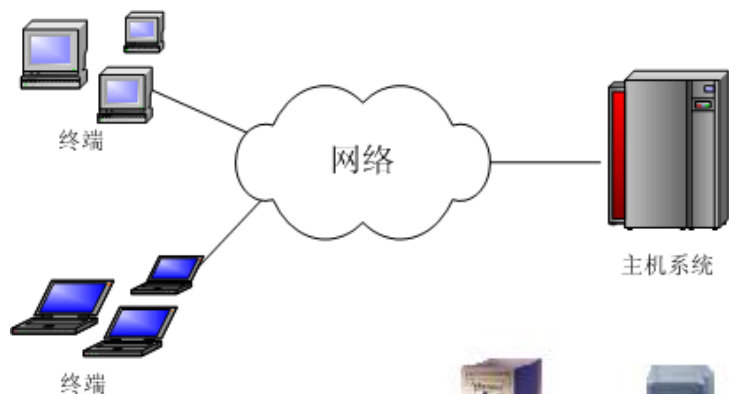
通过浏览器  
手工操作

**B2B**

(Business  
To  
Business)

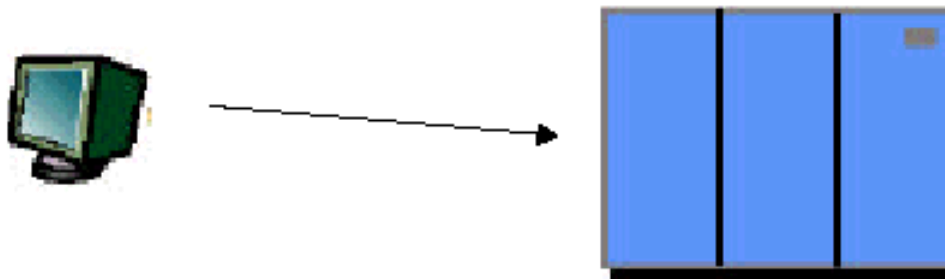


# 分布式计算的发展



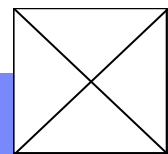
# 分布式计算的发展

---



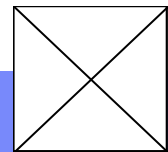
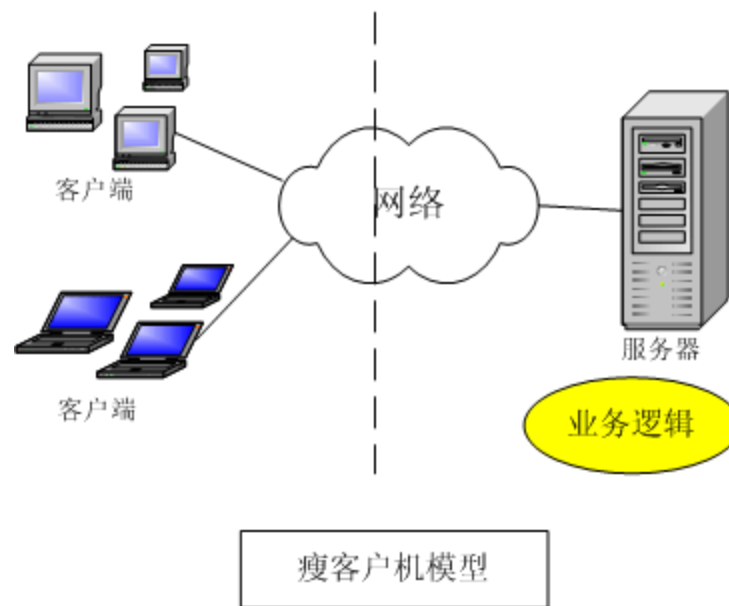
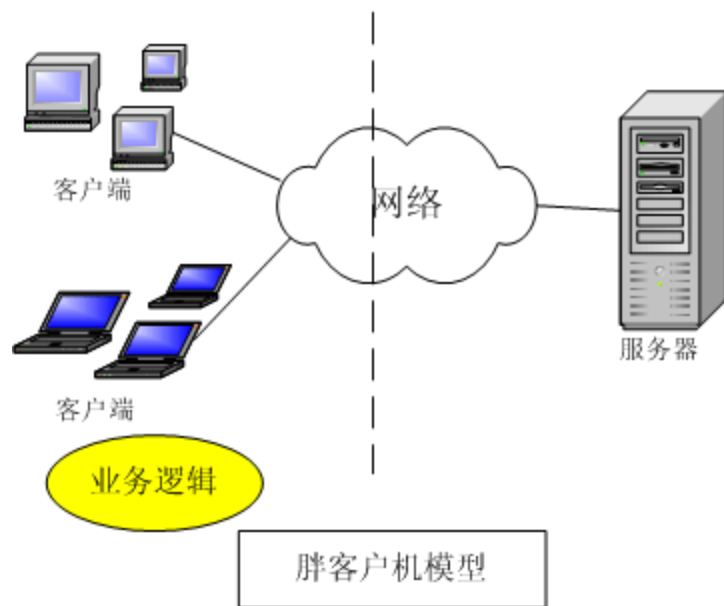
## ■ 主机系统或单层结构

- 数据访问、表示和业务逻辑在一个应用中
- 功能紧紧耦合在一起
- 代码复用、代码可维护性、和代码的修改十分困难。
- 不是分布式的，不具有可伸缩性。



# 分布式计算的发展

## 两层结构的C/S系统



# 瘦客户端

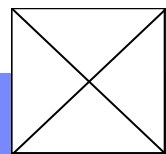
---

## ■ 基本含义

- 在客户/服务器的应用中，被设计得很小以至于大多数的数据操作均在服务器端进行的客户称为瘦客户

## ■ 优点

- 很容易部署
- 很容易使用
- 通过集中管理使系统管理更容易
  - 通过集中管理和监督可以很容易地发现问题
  - 在服务器端进行问题的解决
  - 新版本的软件只需安装在服务器上
- 因为复杂的处理在服务器端进行，所以瘦客户使用的客户端资源很少



# 胖客户端

---

## ■ 基本含义

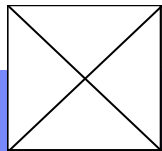
- 在本地执行大多数的数据处理，只有数据本身存储在服务器上的客户称为胖客户

## ■ 优点

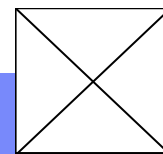
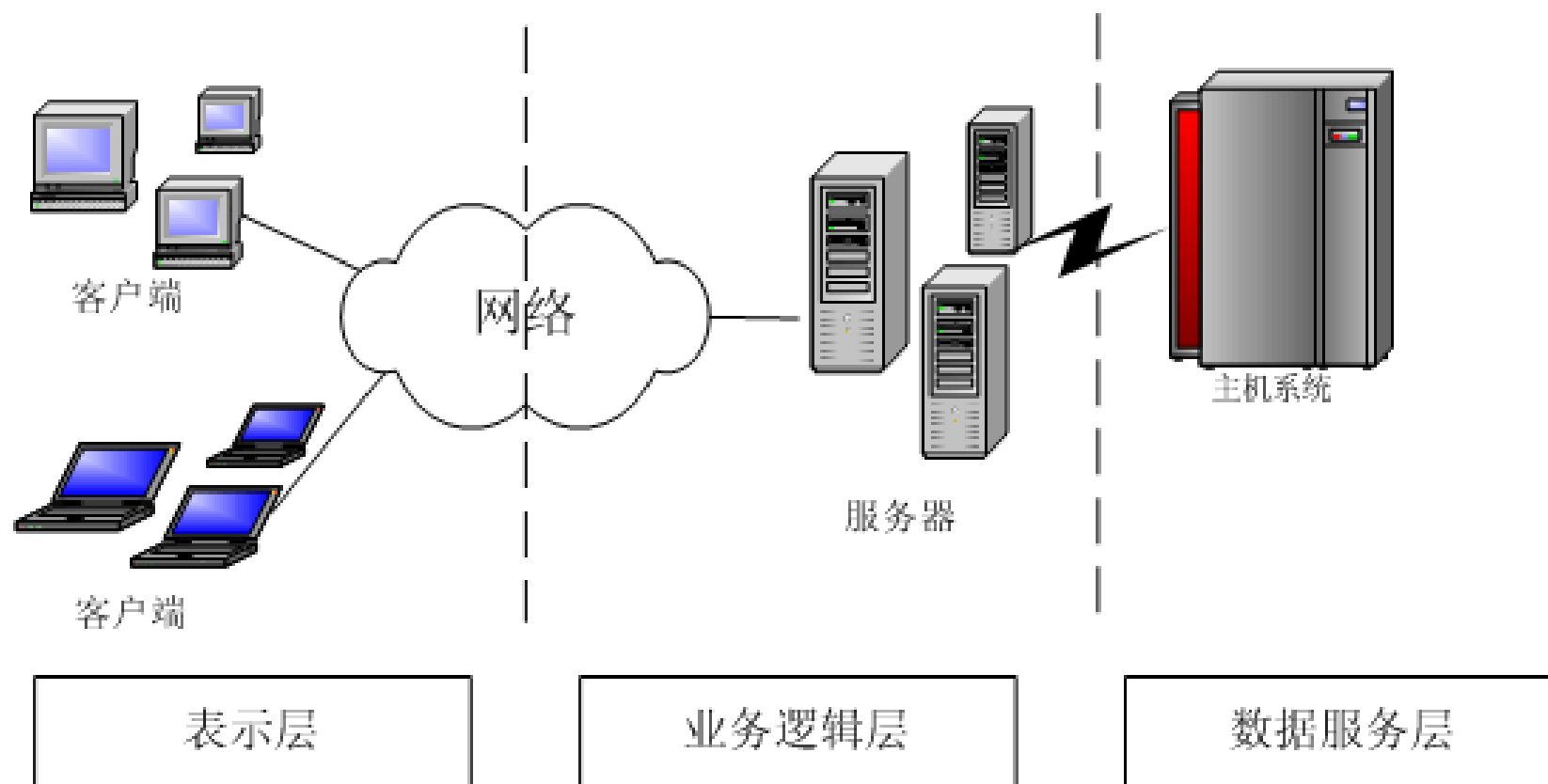
- 提供给终端用户更多的功能，使得终端用户根据自己的需要配置应用程序，因为胖客户机可以存储客户系统的大部分个人数据；
- 可以减少服务器的负担，因为复杂的计算操作是由客户端自己完成

## ■ 缺陷

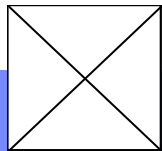
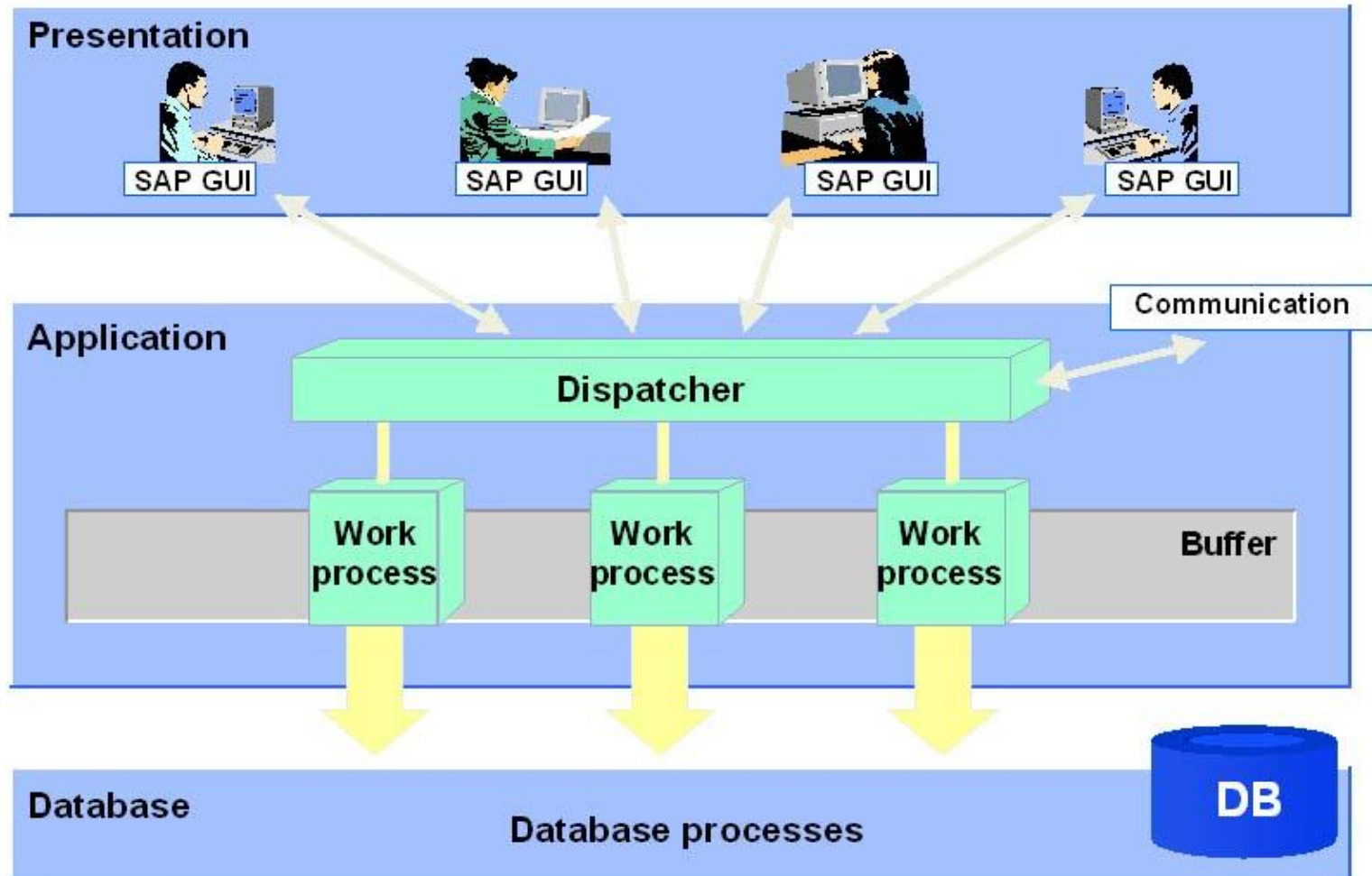
- 需要更多的可能导致错误的安装过程
- 使用起来需要教育培训
- 对新版本的软件需要重新安装
- 需要较多的客户端资源（如内存和CPU处理能力）



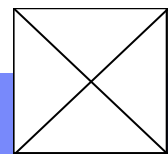
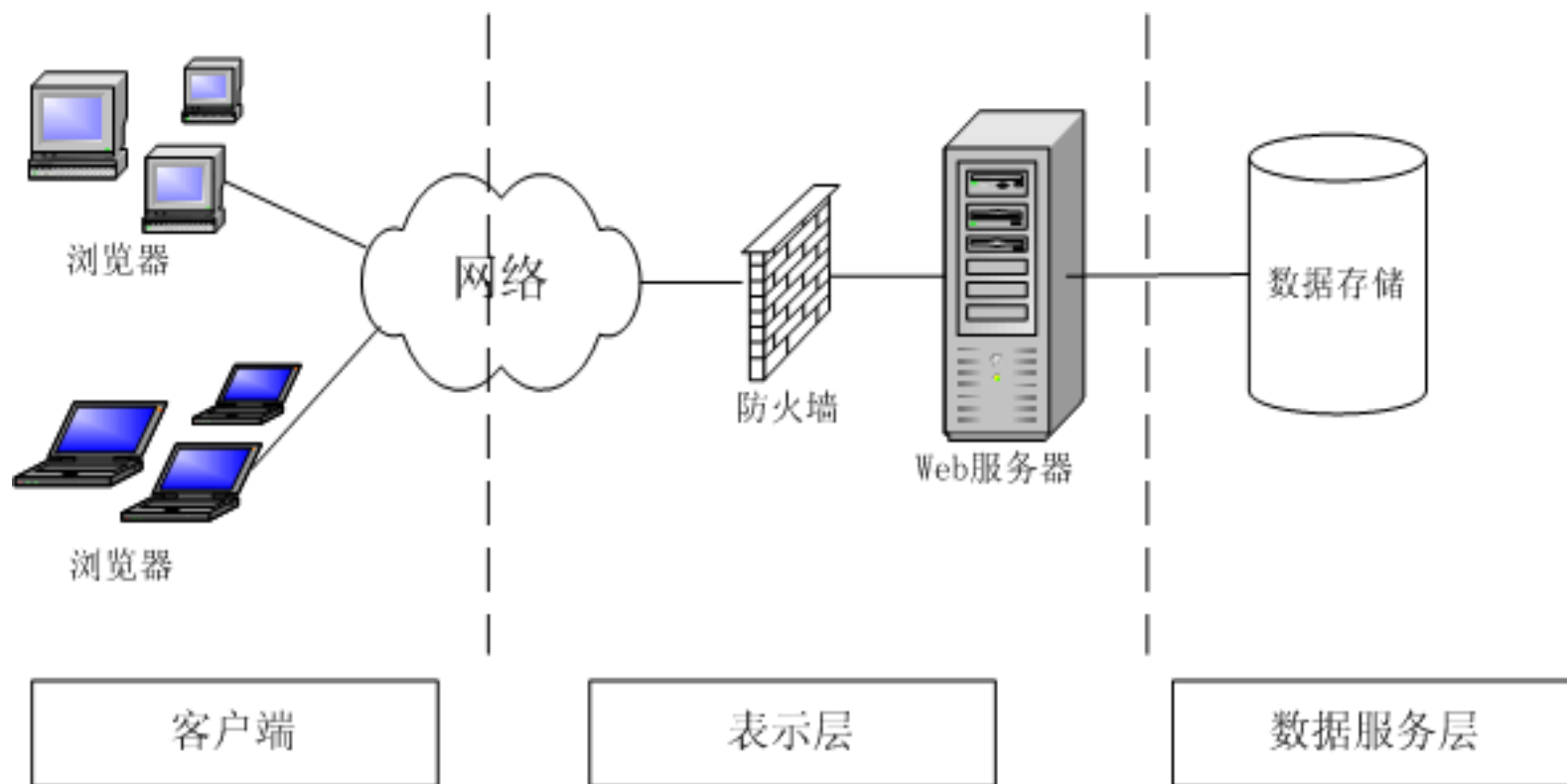
## 三层结构的C/S系统



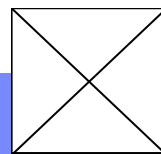
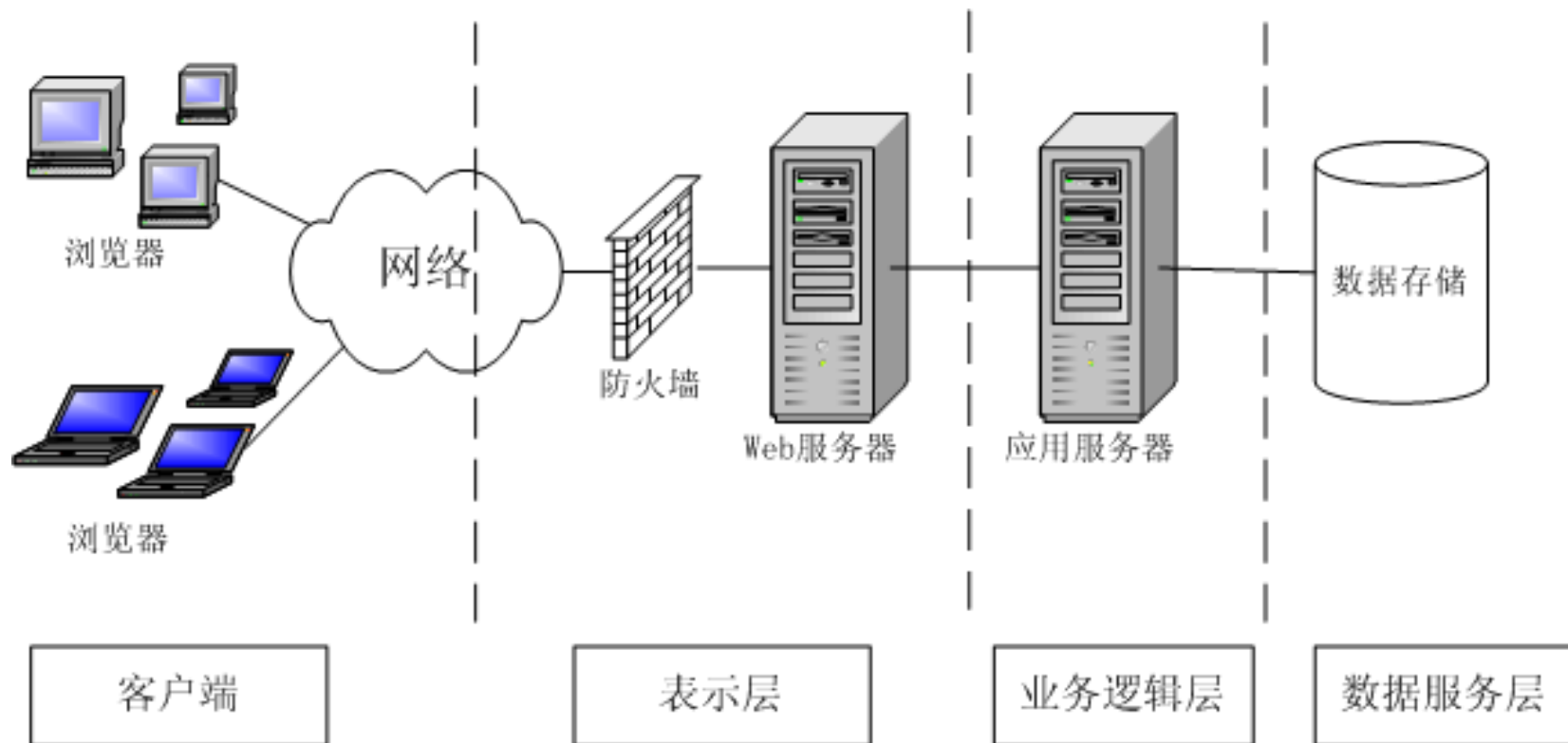
## 三层结构的C/S系统



## 两层结构的B/S系统



## 三层结构的B/S系统



# C/S vs. B/S

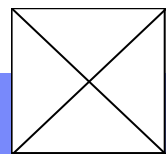
---

## ■ C/S的缺点

- 系统整合性差
- 配置和维护成本高
- 对客户机要求高
- 用户培训时间长
- 伸缩性差
- 软件复用性差

## ■ C/S的优点:

- 交互性强
- 性能
- 网络负载
- 安全
- 用户状态的维护



# C/S vs. B/S

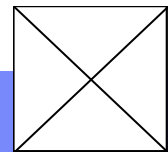
---

## ■ Web架构较C/S架构的优点

- 标准化
- 开发代价低
- 客户端“零花费”发布
- 升级容易
- 可以穿透防火墙
- 易于在异构平台上配置集成
- 降低客户培训费用
- .....

## ■ Web架构较C/S架构的缺点

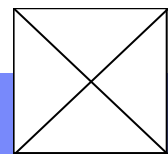
- 界面开发不如C/S方便
- 速度慢,难以满足实时系统要求



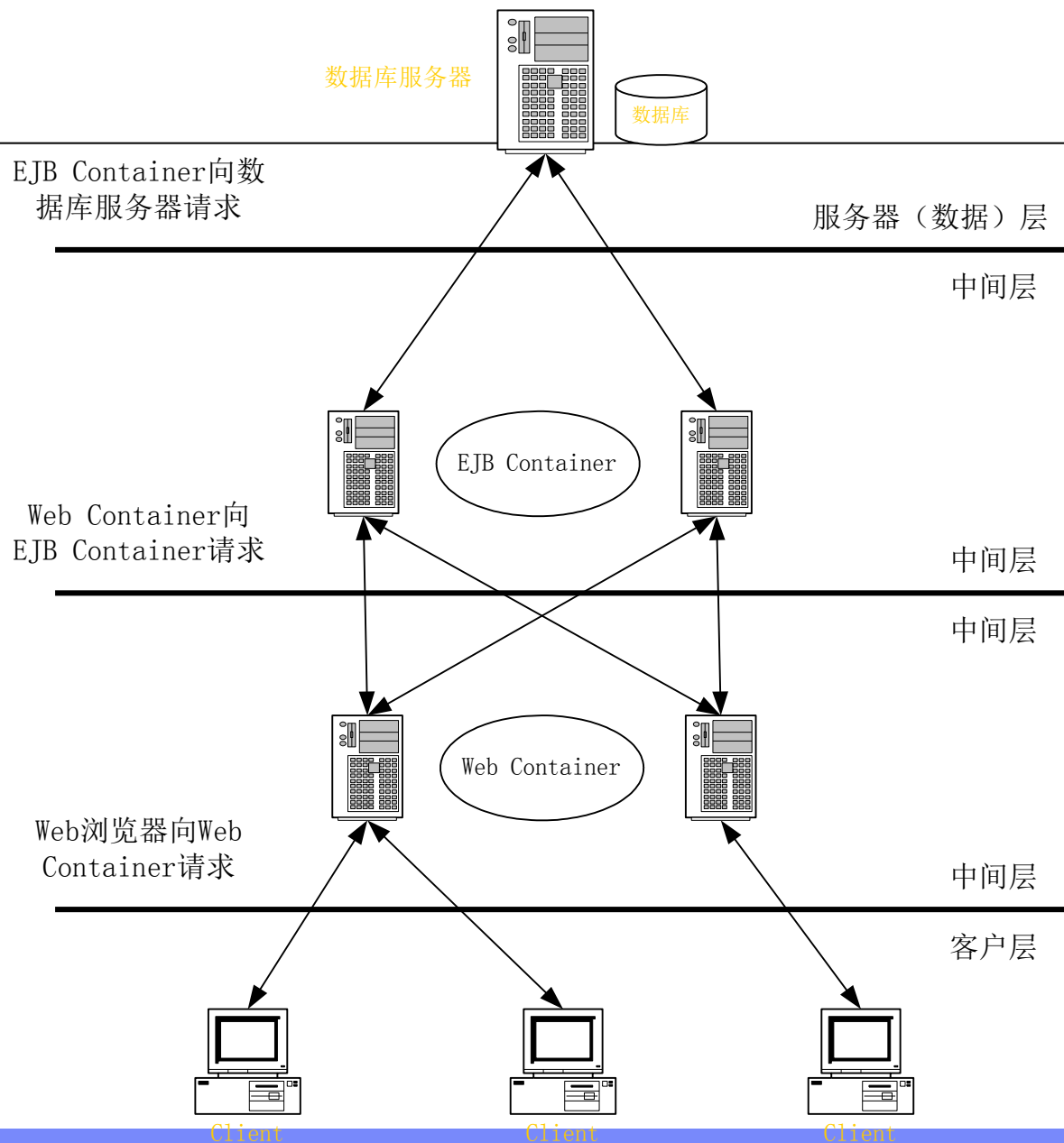
# N层架构

---

- 在3层结构中，客户层和数据层已被严格定义，但中间层并未明确定义。
- 中间层可以包括所有与应用程序的界面和持久数据存储无关的处理。
- 假定将中间层划分成许多服务程序是符合逻辑的，那么将每一主要服务都视为独立的层，则3层结构就成为了n层结构。
  - 如J2EE中间层可以分为
    - 实现任务分配机制和界面呈现的Web服务器层
    - 实现实际商业逻辑的EJB层。

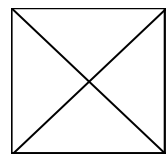
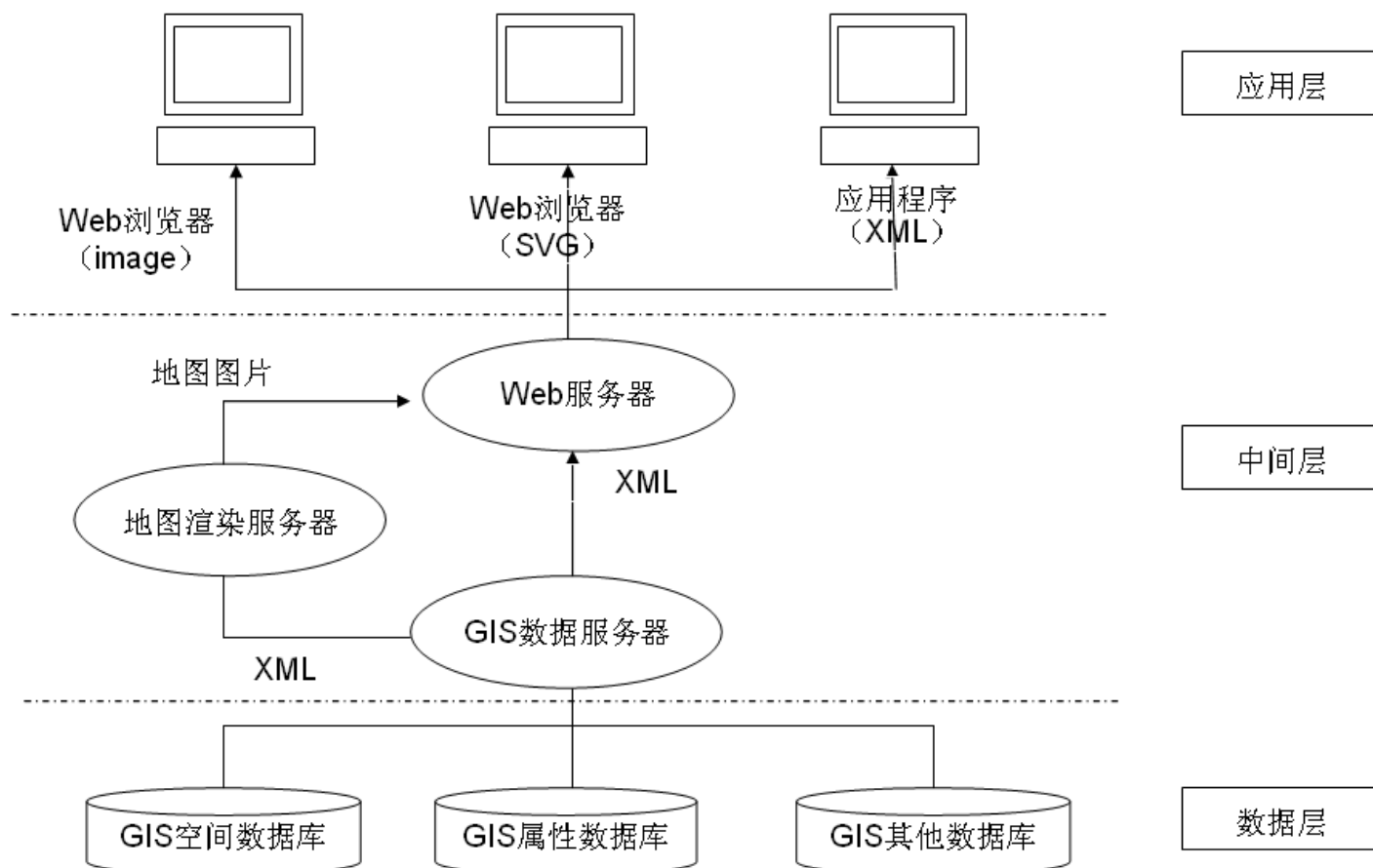


# N层架构



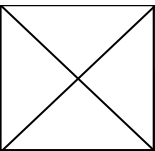
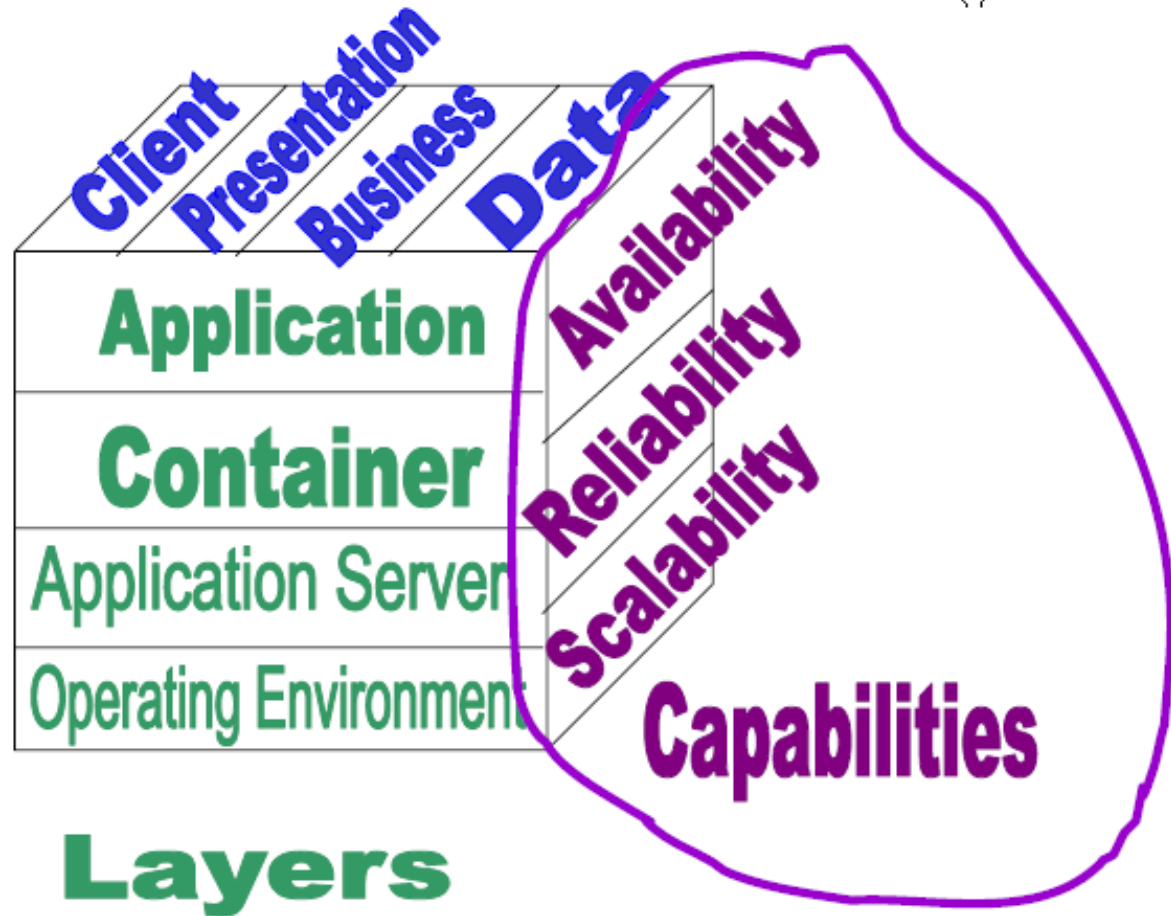
# N层架构

## ■ 分布式 WebGIS 模型

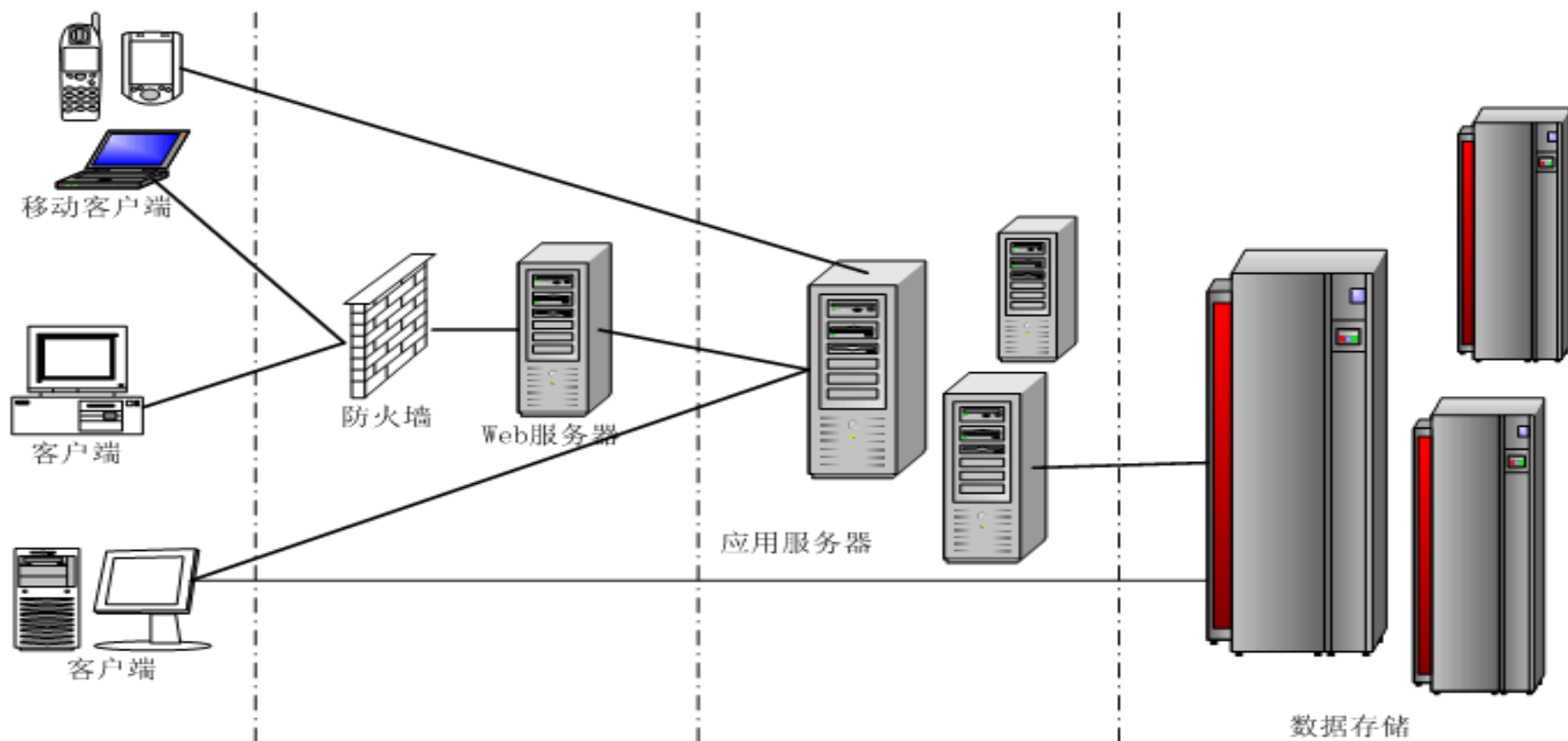


# N层架构

**Tiers**



# 目前主流的B/S系统架构情况



Security XML 底层网络硬件、协议支持

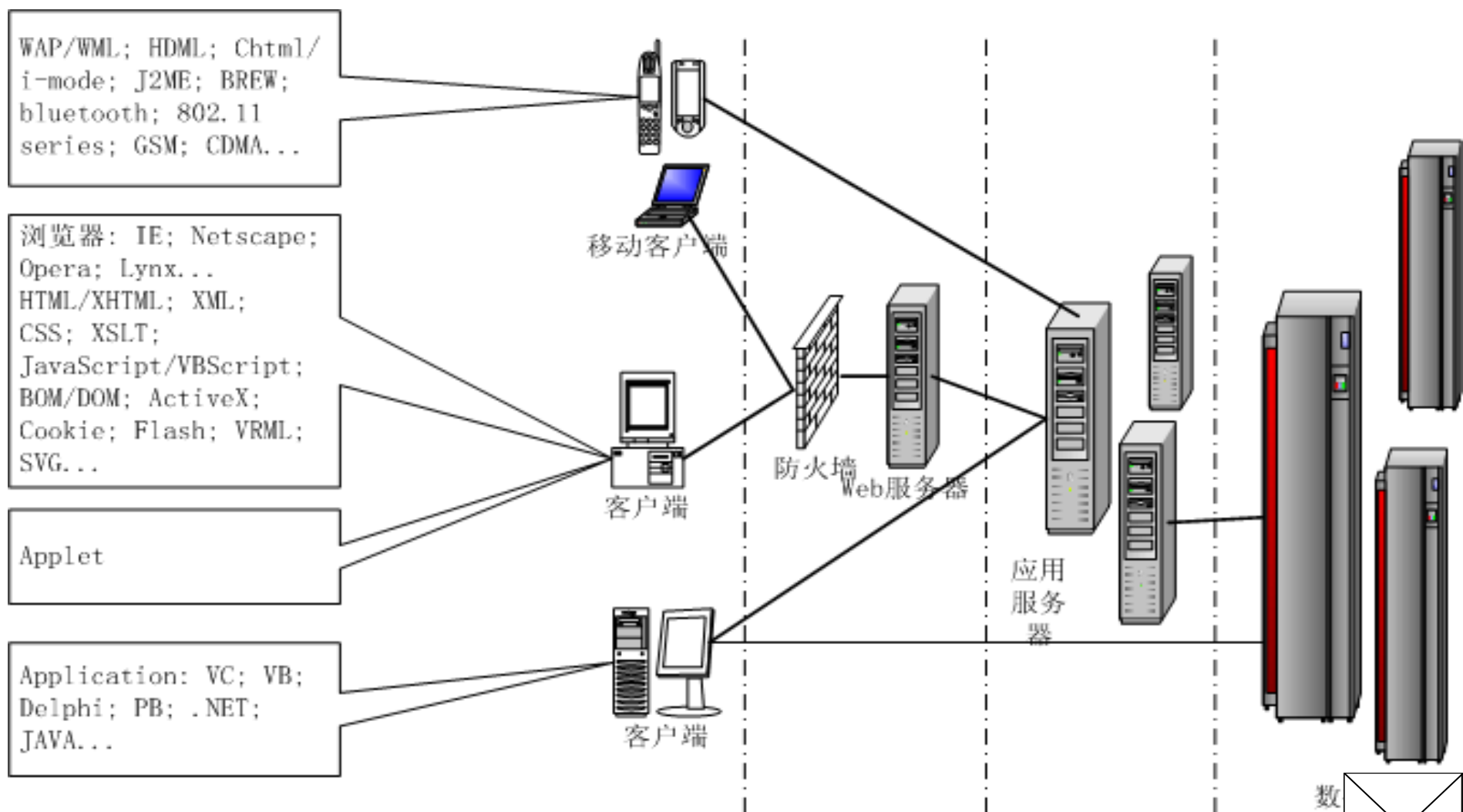
客户端

表示层

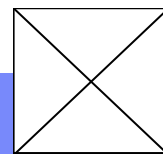
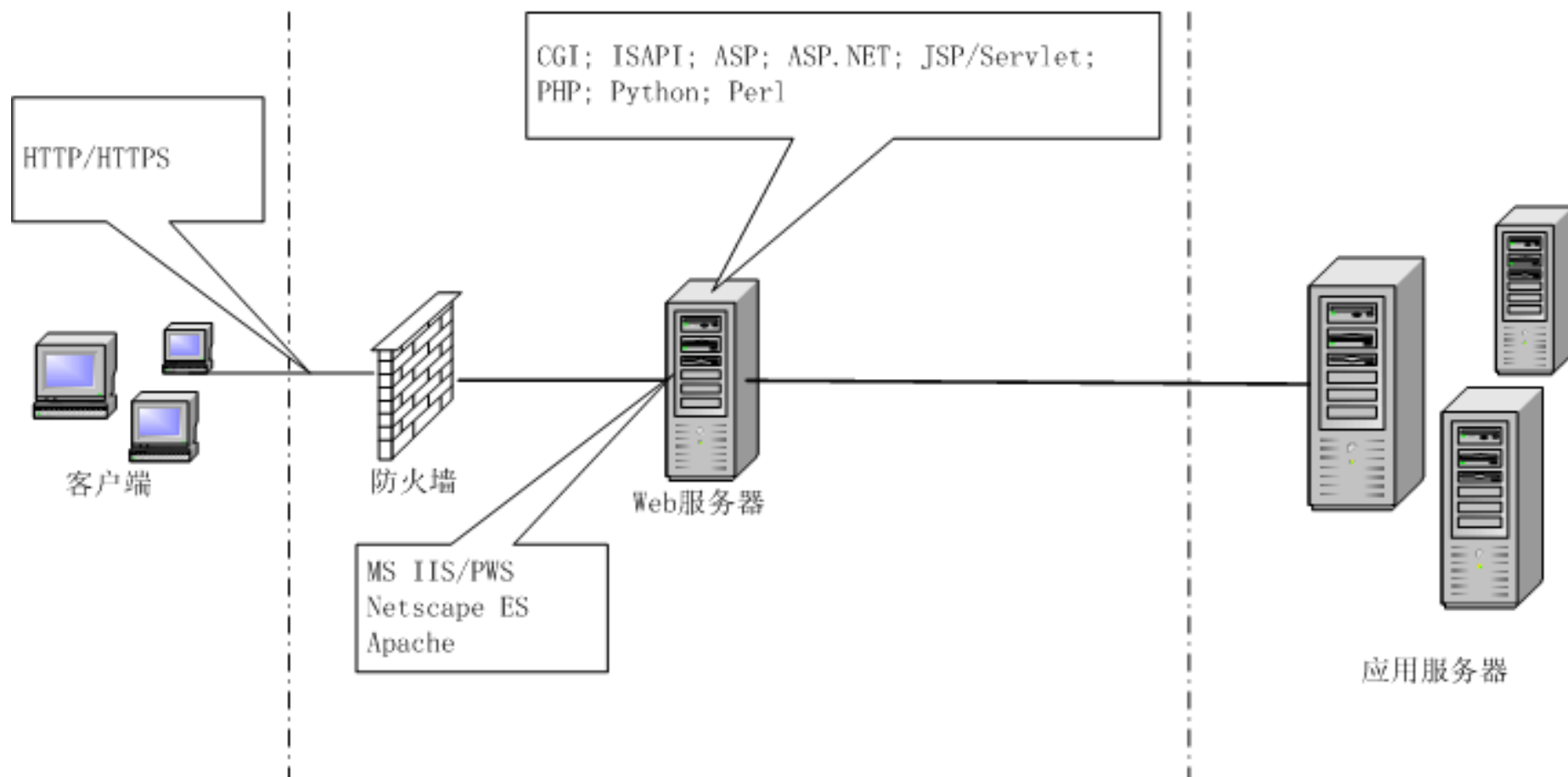
业务逻辑层

数据服务层

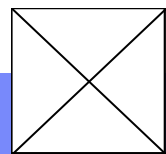
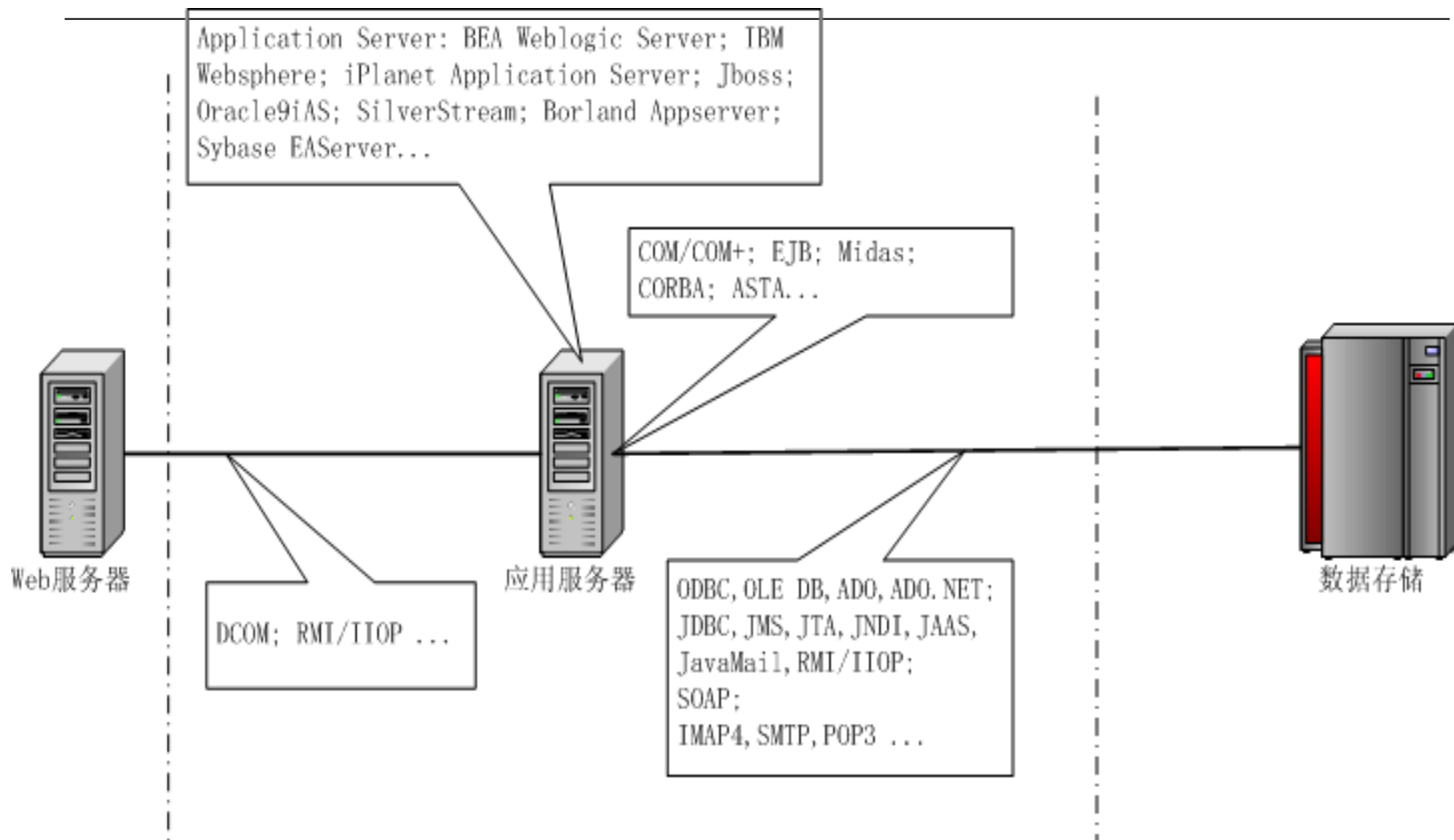
# 各部分主流技术—客户端



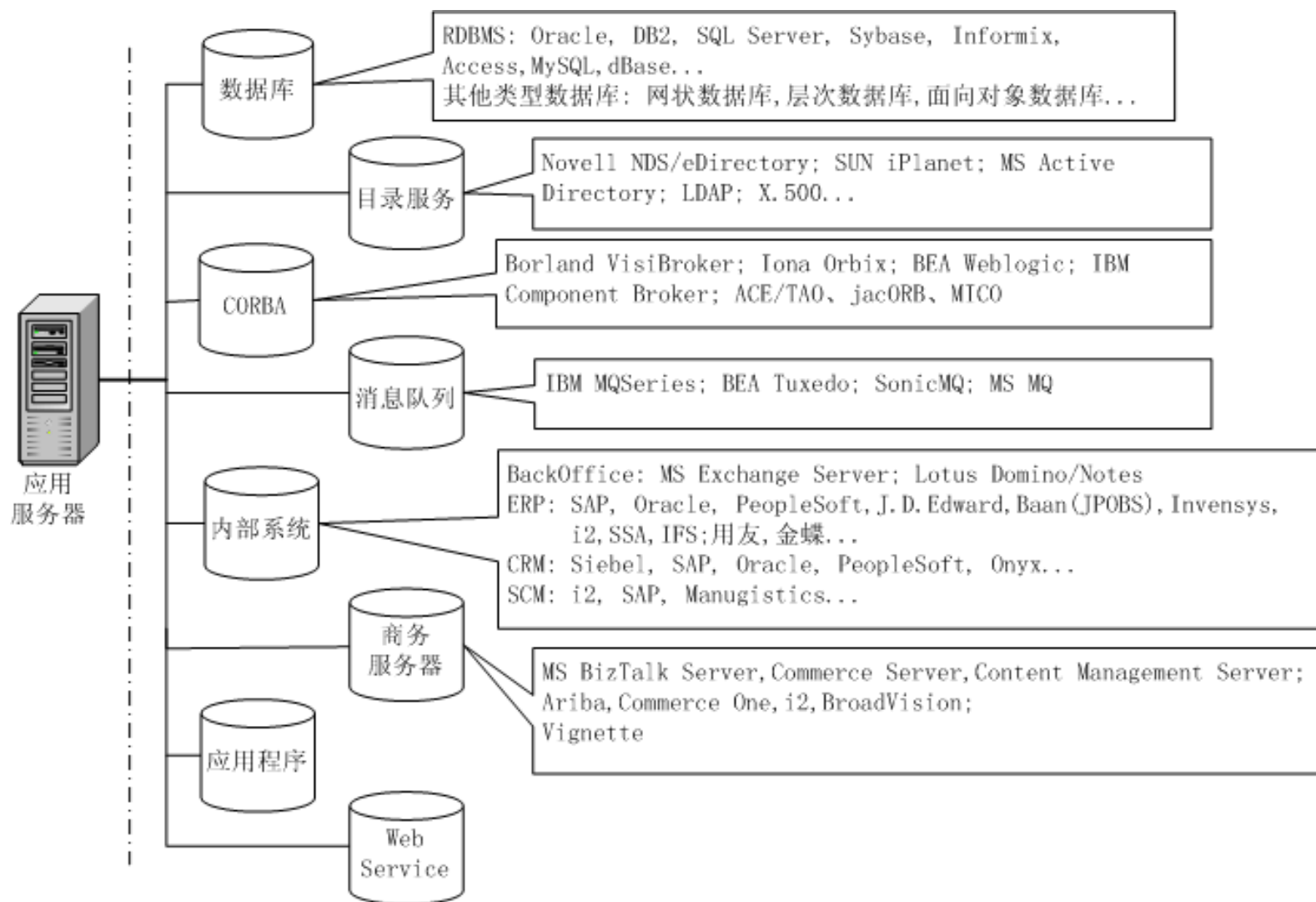
## 各部分主流技术—表示层



## 各部分主流技术—业务逻辑层

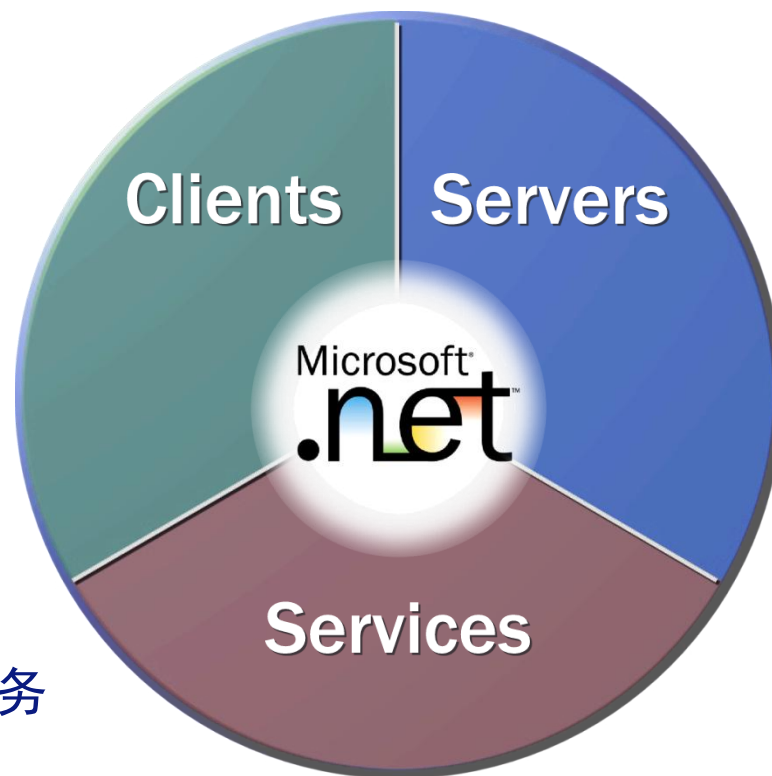


# 各部分主流技术—数据服务层

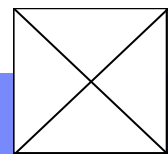


# Microsoft .net

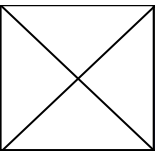
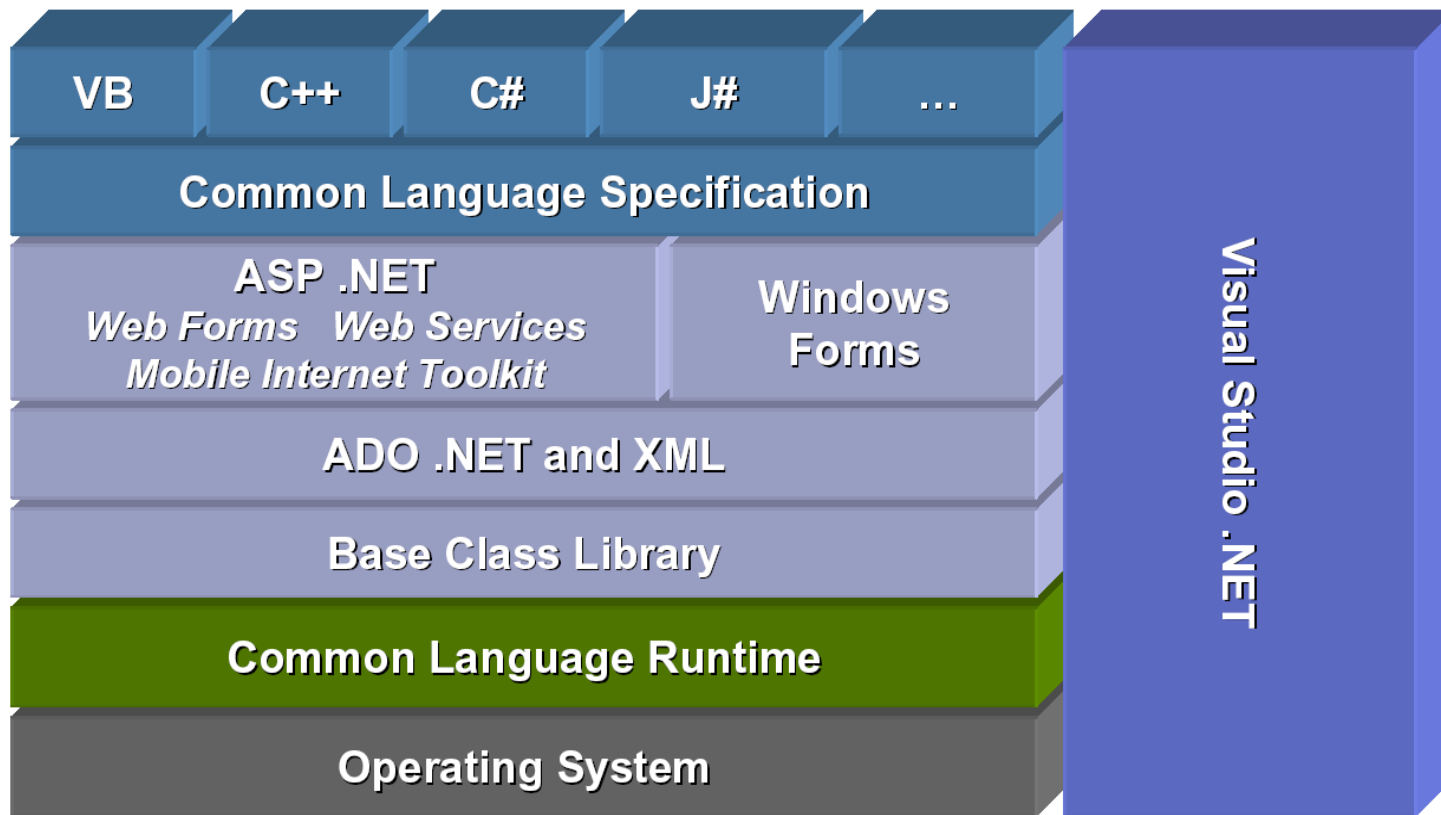
**Microsoft® .NET** 是一系列连接信息、使用者、应用系统和设备的软件技术



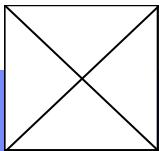
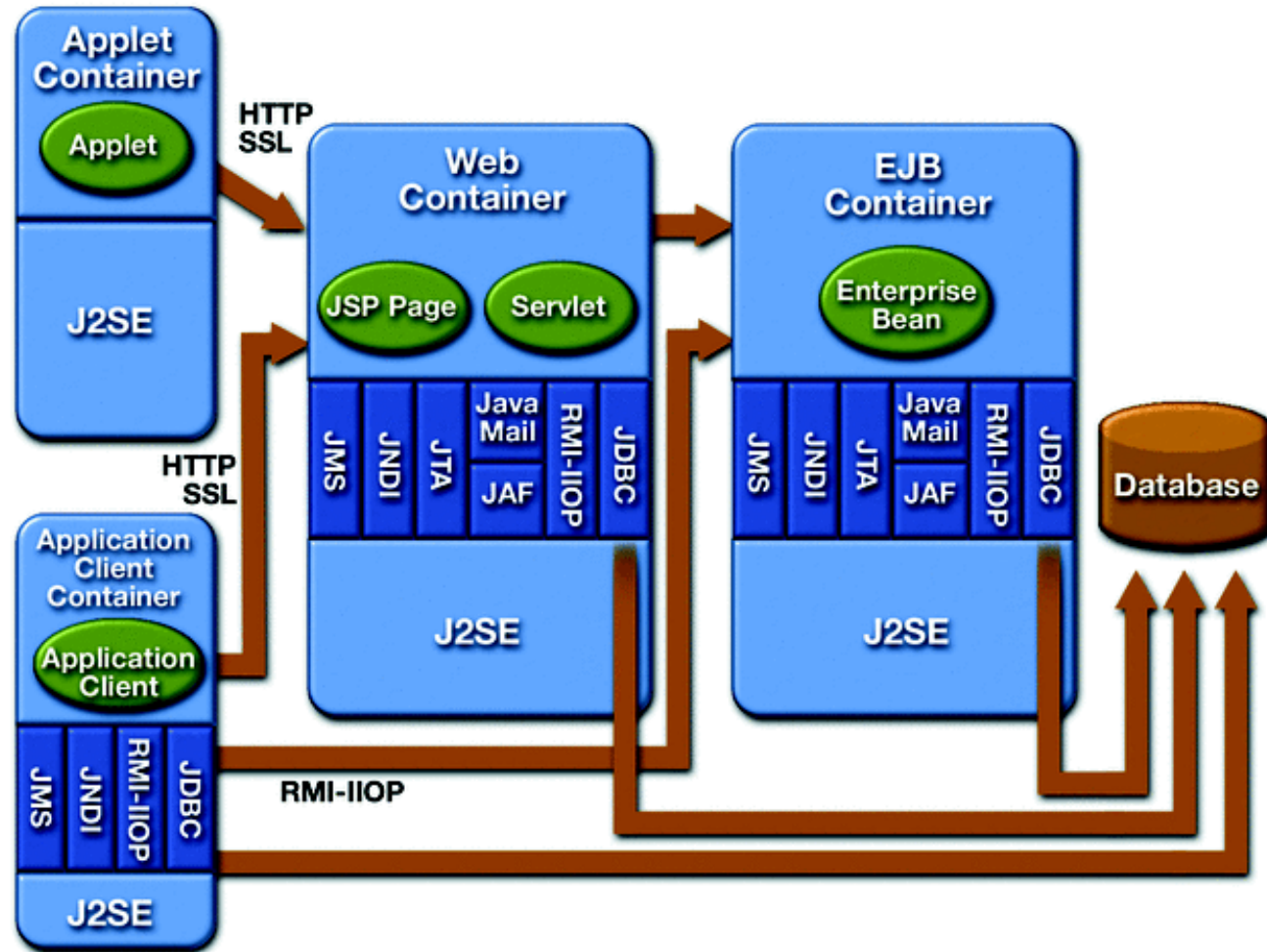
- 通用的编程模型
- 包括客户端、服务器和应用服务
- 开发工具



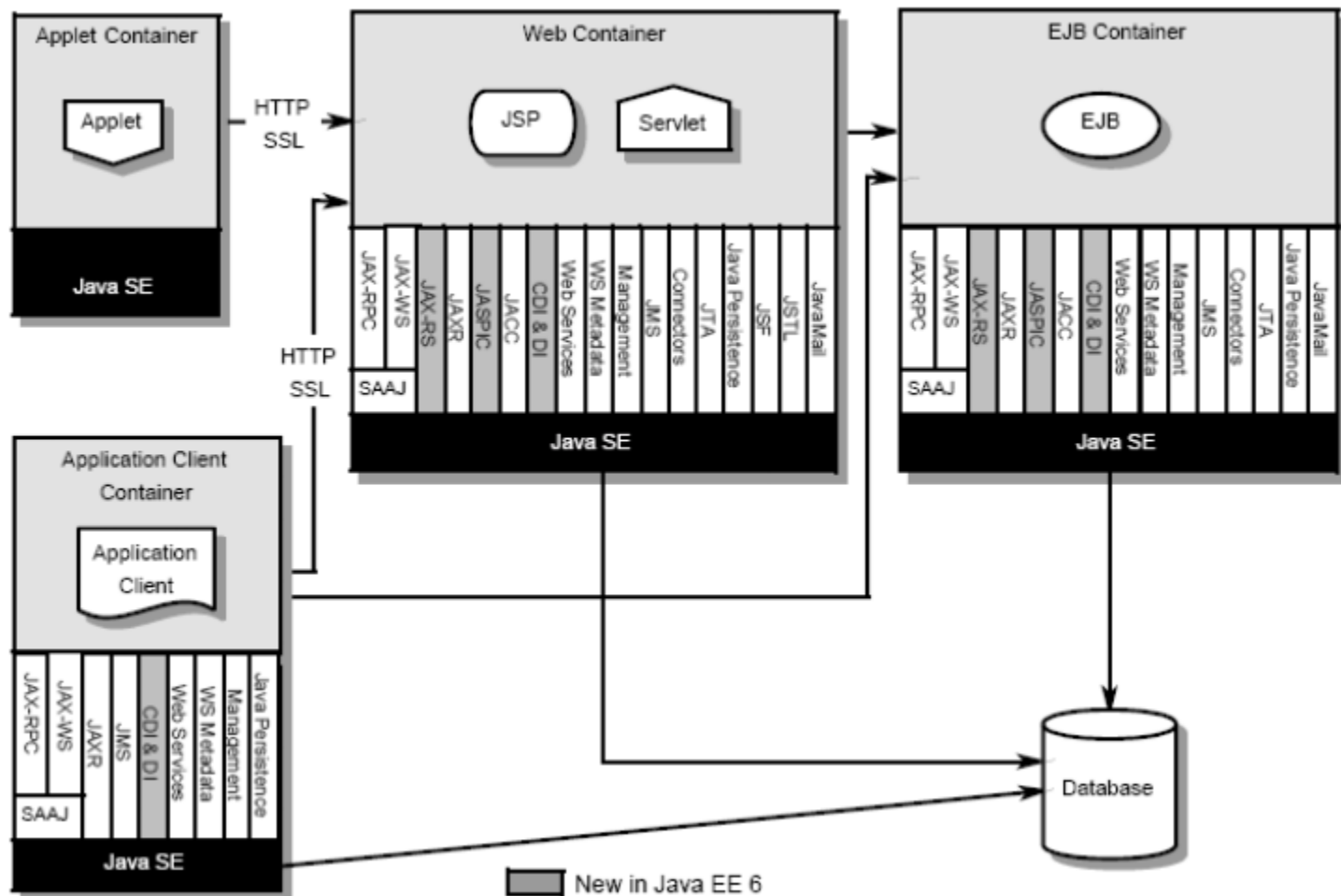
# Microsoft .net 架构、语言、工具



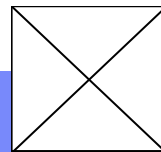
# J2EE Overview



# J2EE Overview



- ◆ RESTful Web Services Java API (JAX-RS)
- ◆ 面向Java EE平台的上下文和依赖性注入（CDI）
- ◆ Bean验证



# J2EE的主流框架技术和发展

---

- **SSH**

- **Spring**

- 反转控制(依赖注入)

- Spring AOP

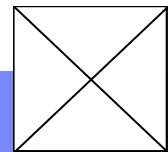
- **Struts2.0**

- **Hibernate**

- **EJB3.0**

- **JSF**

- ...



# LAMP

---

- **LAMP**

- **Linux**
- **Apache**
- **MySQL**
- **PHP、Perl或Python**

