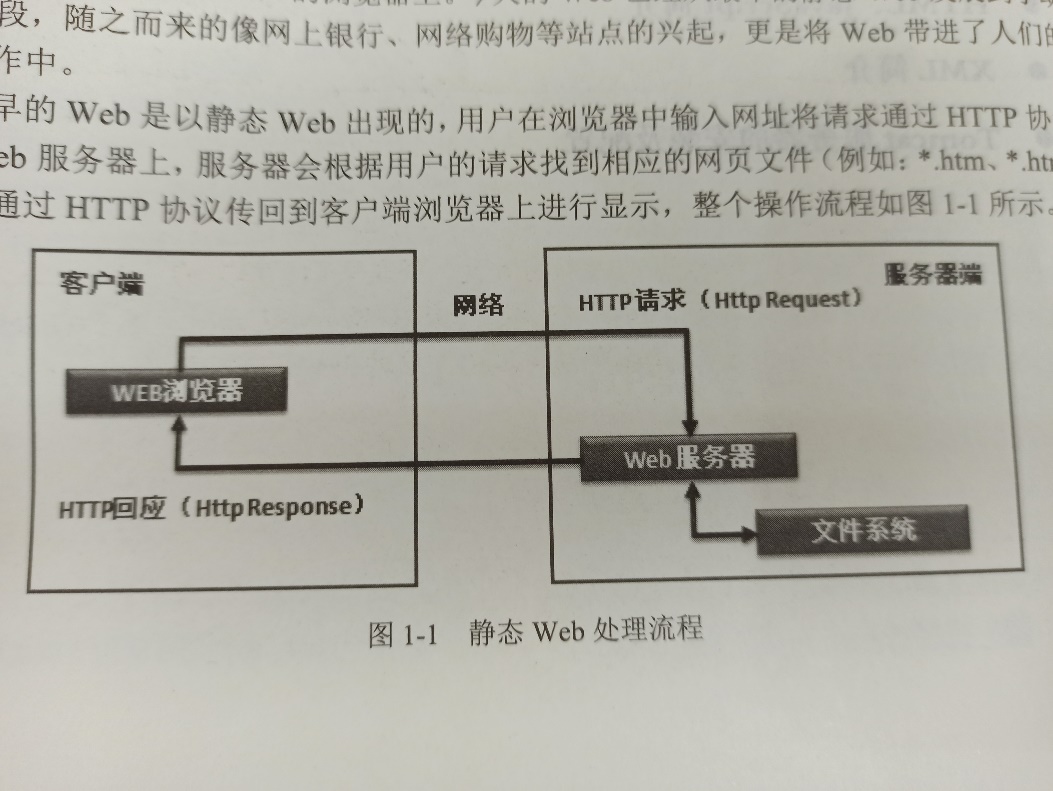
# Web发展历程

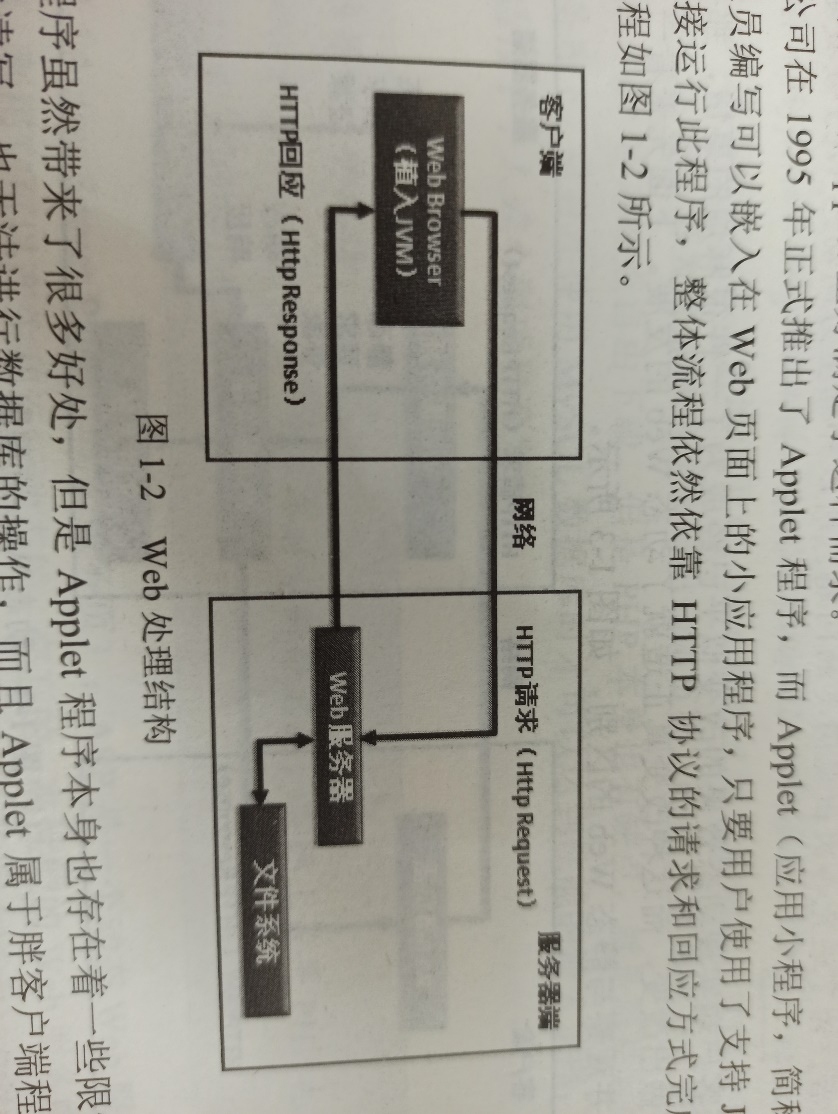
## 发展历程

1. 在早期，人们为了方便开展科学研究，设计出了Internet用于连接美国少数几个顶尖研究机构，之后随着进一步的发展，人们开始应用HTTP协议（Hypertext Transfer Protocol，超文本传输协议）进行超文本（hypertext）和超媒体（hypermedia）数据的传输，从而将一个个网页展示在每一个用户的浏览器上。
2. 最早的Web是以静态Web出现的，用户在浏览器中输入网址将请求通过HTTP协议传送到Web服务器上，服务器会根据用户的请求找到相应的网页文件（例如：\*.html、\*.htm），接着在通过HTTP请求协议传回到客户端浏览器上进行显示，整个流程如下：



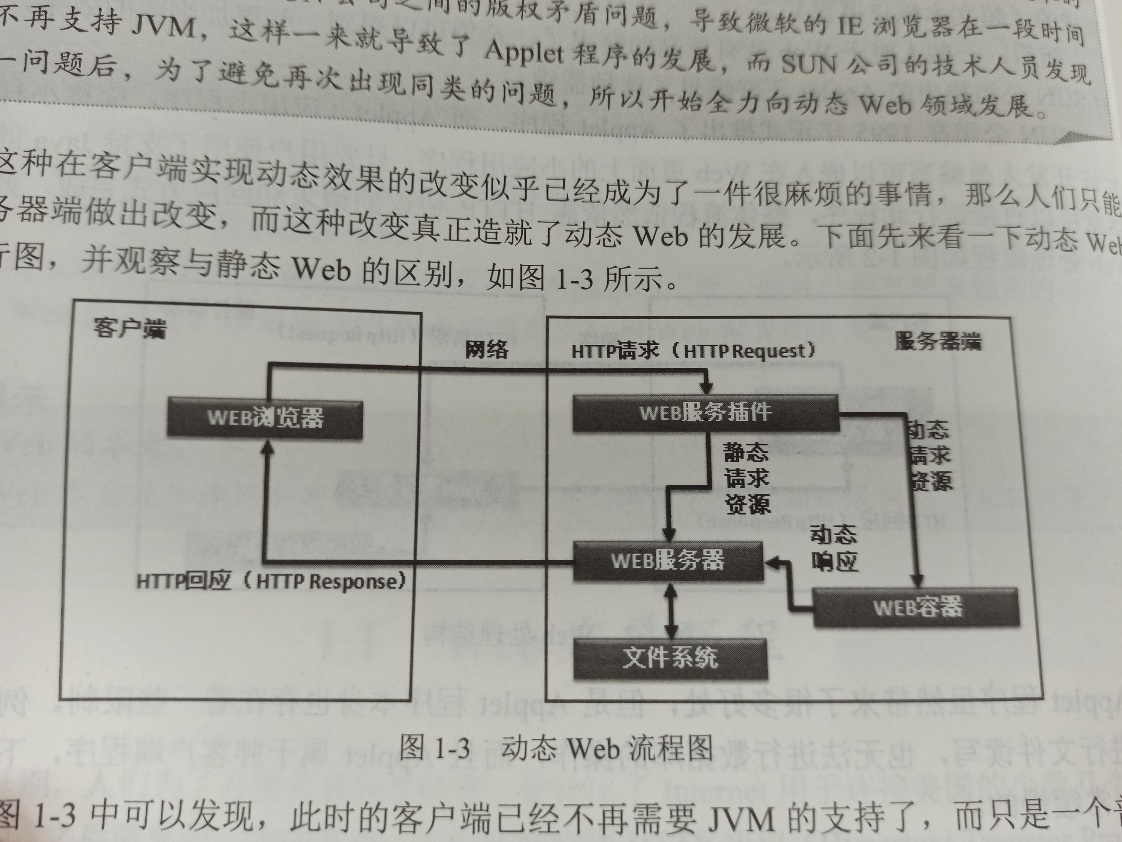
但是这种Web返回的只是电子文本的形式，在服务器生成之后，内容永远是固定的，实现静态Web的主要手段也就是利用HTML来完成的（超文本标记语言）。

1. 随后，人们对Web就有了更高的要求，希望可以得到一些更加绚丽的效果，而此时SUN公司推出了Applet程序（应用小程序，简称小程序），允许开发人员编写可以嵌入在Web页面上的小应用程序，只要支持了Java的浏览器就可以直接运行此程序，整体流程依然依靠HTTP协议的请求和回应方式完成，此时Web处理流程如下：



Applet程序虽然带来了很多好处，但是Applet程序本身也存在一些限制。例如，不能允许进行文件读写，也无法进行数据库的操作，而且下载速度也是非常缓慢的。并且Applet程序需要浏览器的支持，SUM和微软有矛盾，导致微软的浏览器在一段时间内不再支持JVM，这样一来就导致了Applet程序的发展。

1. 通过修改客户端在浏览器器中内嵌JVM运行Applet程序，这种在客户端实现动态效果的改变似乎已经成为了一件非常麻烦的是，于是人们只能在服务端做出改变，而正是这种改变造就了动态Web的发展。动态Web流程图如下：



此时的客户端不再需要JVM的支持了，只是一个普通的浏览器，但是服务端却发生了重大的改变。所有请求不再直接提交给Web服务器，而是通过Web服务器插件进行接收，此插件的目的是用于区分用户所发出的请求时动态请求还是静态请求。如果用户发出的是静态请求，则会将用户请求交给Web服务器，并通过文件系统将用户所需要的资源发回给客户端浏览器，这一点与最初的静态Web处理流程是一样的；但如果此时的请求时动态请求，则会将所有的请求交给Web容器进行处理，在Web容器中会采用拼凑代码的形式（主要是拼凑HTML）动态地生成数据并通过Web服务器发回给客户端浏览器。

1. 静态Web与动态Web的最本质的区别实际上只有一点，就是静态Web是无法进行数据库操作的，而动态Web可以进行数据库操作。
2. 动态Web最大的特点就是具备交互性，所谓交互性就是服务器端会自动根据用户请求的不同而显示不同的结果。

## 动态Web技术

要想实现动态Web，现在可以采用如下5种方式：

1. **CGI**

CGI（Common Gateway Interface 公共网关接口）：CGI是最早出现的实现动态Web的操作标准，可以采用任何语言实现（如C或VB），但是这种传统的CGI程序本身会采用多进程的机制进行处理，每当一个新用户连接到服务器上时，服务器都会为其分配一个新的进程，这种程序的执行效率是最低的。

1. **PHP**

PHP（Hypertext Preprocessor，超文本预处理），PHP是一种跨平台的服务器端嵌入式脚本语言。它大量地借用C、Java和Perl语言的语法，并结合自身特性，使Web开发者能够迅速地写出动态页面。而且PHP是完全免费的。但是PHP本身也有缺点，就是需要在运行Apache服务器下，只有在使用MySQL数据库时才可以达到性能的最大发挥，所以一般都只适合个人或小型项目开发。

1. **ASP**

ASP（Active Server Pages，动态服务页）是一个动态Web服务器端的开发环境，利用它可以产生和运行动态的、交互的、高性能的Web服务应用程序。ASP采用VBScript（类似于JavaScript）脚本语言作为自己的开发语言。由于ASP技术出现较早，所有一直到今天还在被陆续使用着，但ASP技术本身有一个最大的问题就是平台的支持，ASP只能运行在IIS（Internal Information Services，互联网信息服务）服务器上，且只能在SQL Server数据上才可以得到最大发挥。但是这套开发相对于Javja开发而言，性能是很差的，所以一般用于个人或中小型项目开发。

1. **ASP.NET**

ASP.NET是微软公司继ASP之后退出的新一代动态网站开发技术。ASP.NET基于.NET框架平台，用户可以选择.NET框架下自己喜欢的语言进行开发。ASP.NET技术是ASP技术的更新，也是微软公司目前主推的技术，但是由于微软的产品都会受到平台的限制，所以此技术往往用于中型项目的开发。

1. **JSP**

JSP（Java Server Page，Java服务页），使用Java完成动态Web开发，代码风格与ASP类似，都属于在HTML代码中嵌入Java代码以实现功能，由于Java语言的跨平台性，所以JSP不会受到操作系统或开发平台的限制，而且有多种服务器可以支持，如Tomcat、WebLogic，JBoss、Websphere等，所以经常在中大型项目开发中使用。JSP的前身是Servlet（服务器端小程序），但是由于Servlet开发过于复杂，所以SUN公司的开发人员根据ASP技术的特点，将Servlet程序重新包装，而形成的一门开发技术----JSP。

## 开发模式

在网络开发中有两种开发模式：C/S模式和B/S模式。

1. **C/S模式**

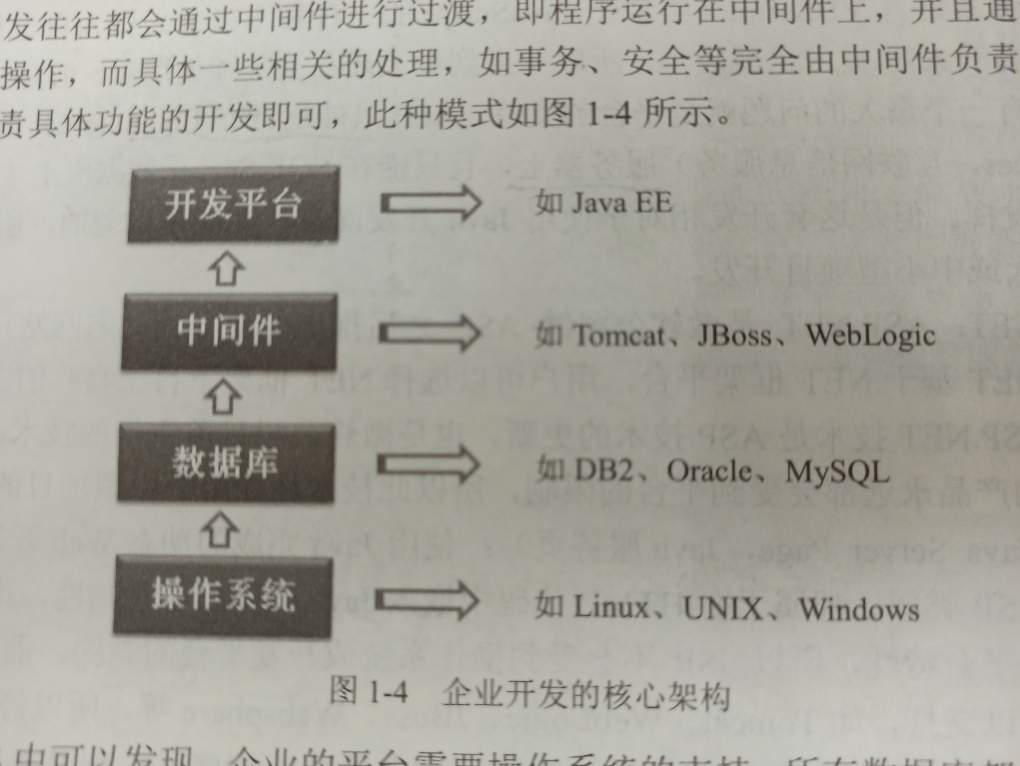
C/S模式（Client/Server模式），即客户端/服务器模式。在这种模式下，每个客户端都需要安装工具软件，管理和维护时客户端和服务器端都同时需要更改，对于开发而言比较麻烦。例如，日常生活中使用的QQ就属于C/S模式。

1. **B/S模式**

B/S（Brower/Server模式），即浏览器/服务器模式。相当于C/S模式中，以浏览器作为客户端的情况。在服务器器端安装软件，客户端通过浏览器访问服务器，从而实现信息、资源的交互和共享，只需要管理和维护服务器端即可。例如，网上购物或论坛网站等都属于B/S模式。

# 企业开发架构

在现代的企业平台开发中已经大量地使用了B/S开发模式，不管是使用何种动态Web实现手段，其操作形式都是一样的，核心操作的大部分都是围绕数据库进行的。但是如果直接使用编程语言进行数据的开发则程序员要处理诸如事务、安全等操作，所以现在的开发往往都会通过中间件进行过渡，即程序运行在中间件上，并且通过中间件进行数据库的操作，而具体的一些相关的处理，如事务、安全等由中间件负责，这样程序员只需要负责具体功能的开发即可。此种模式如下图所示：



可以发现，企业的平台需要操作系统的支持，所有的数据库都是建立在操作系统上的，之后开发平台（JavaEE就是一种开发平台）通过中间件进行数据库的操作。同时如果你是一个Java EE的开发人员，则必须会使用操作系统，而且对于数据库的基本操作和SQL语句也必须相当熟练。最重要的是，开发人员往往都必须会使用中间件，因为所有程序都要在中间件上部署和或运行。

# Java EE架构

## 认识JavaEE

Java EE（Java Enterprise Edition，也称J2EE）是在Java SE基础之上建立起来的一种标准开发架构，主要用于企业级应用程序的开发，以B/S作为主要开发模式。在Java EE中提供了多种组件及各种服务。

JavaEE主要由容器、组件和服务三大核心部分组成。

1. JavaEE容器：容器负责一种组件的运行。在Java中一共提供了4种容器，即Applet Container、Application Client Container、Web Container和EJB Container。每一个容 器负责运行自己的组件，处理各自的程序，且相互没有任何影响。例如运行Web 程序，则一定要有Web容器的支持。
2. JavaEE组件：每一种JavaEE容器实际上都表示着一种程序的开发。例如，Application 程序是使用主方法（main( )）运行的一种组件。每一个组件都由对应的容器负责运 行。例如Web容器中运行的是JSP和Servlet组件，Application Client Container容器

是负责运行main( )方法的容器。

1. JavaEE服务：JavaEE提供了各种服务，通过这些服务可以方便用户进行开发。例如， 进行数据库操作则使用JDBC服务。

## D:\Program Files\Tencent\QQ\1875889780\FileRecv\MobileFile\IMG_20200923_214716.jpg企业应用环境

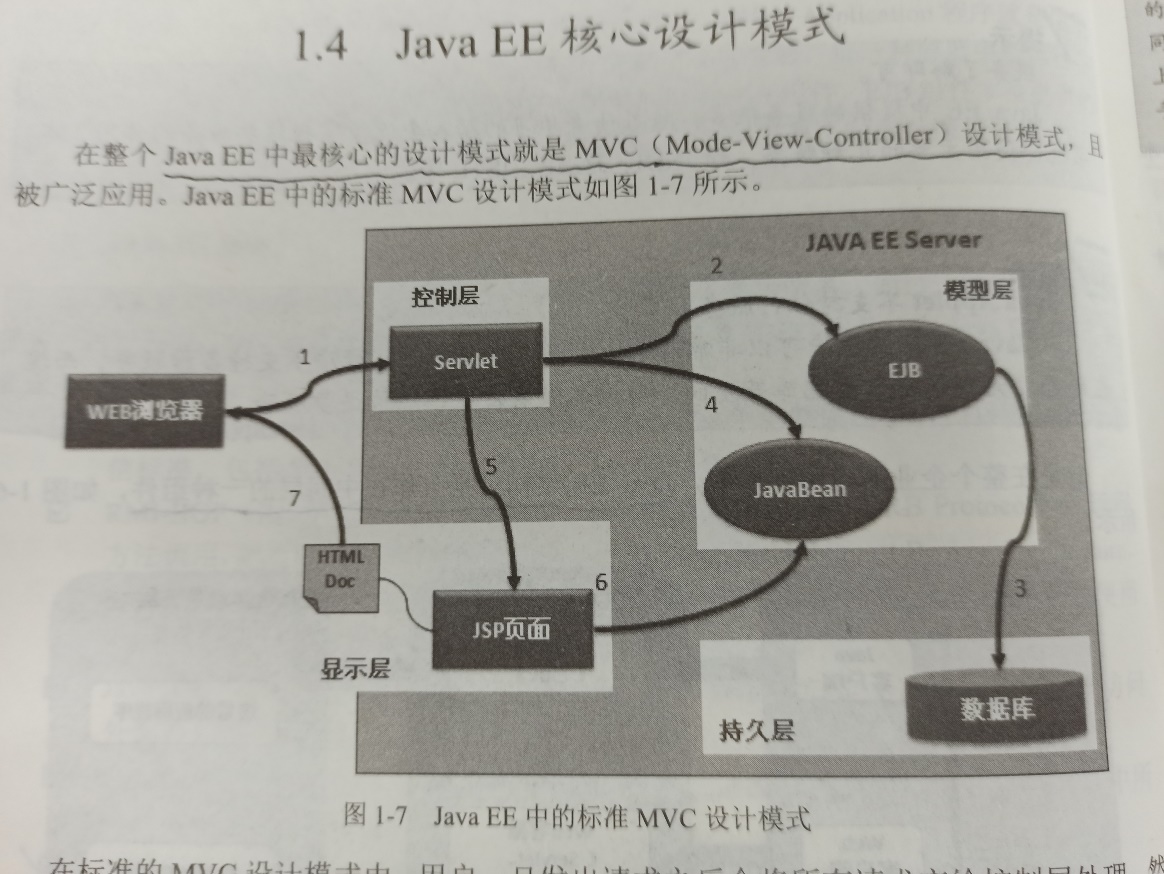
但是在整个企业的应用环境中，Java EE架构只是工作在中间层的一种组件，如下图：

在整个企业开发中主要分为如下3个层次：

1. 客户层：客户层分为内部用户和外部用户，客户端可以使用Web浏览器，也可以是Java编写的应用程序。
2. 中间层：为客户端访问提供服务，是使用Java EE中的各种组件技术进行搭建，且各个容器之间允许互相调用。
3. 企业信息系统层（Enterprise Information System，EIS）：例如，保存数据库的数据库就是工作在此层。

# Java EE核心设计模式

在整个JavaEE中最核心的设计模式就是MVC（Model-View-Controller）设计模式，且被广泛应用，是软件开发中的一大重要思想。Java EE中标准MVC设计模式如下图所示：



在标准的MVC设计模式中，用户通过浏览器发出请求后，请求交给控制层进行处理，然后由控制层调用模型层中的模型组件，并通过这些模型组件进行持久层的访问，再将所有结果都保存在JavaBean（Java类）中，最终由JSP和JavaBean一起完成页面的显示。但是此种情况在不同开发架构中也会存在一些区别，因为在开发中如果没有特殊的需要不一定会使用EJB技术。并且，需要特别注意：框架（Struct、Spring、Hibernate、SpringMVC、MyBatis等）只是一种很简单的应用，MVC是核心，是最重要的基础。

# Structs开发框架

使用标准的MVC设计模式进行开发，则肯定要进行过多的复杂设计，这对于一般项目而言是非常麻烦的，所以Apache专门提供了一套用于进行MVC开发的框架--------Structs。Structs是一个MVC框架，其内部实现原理依然是MVC设计模式。