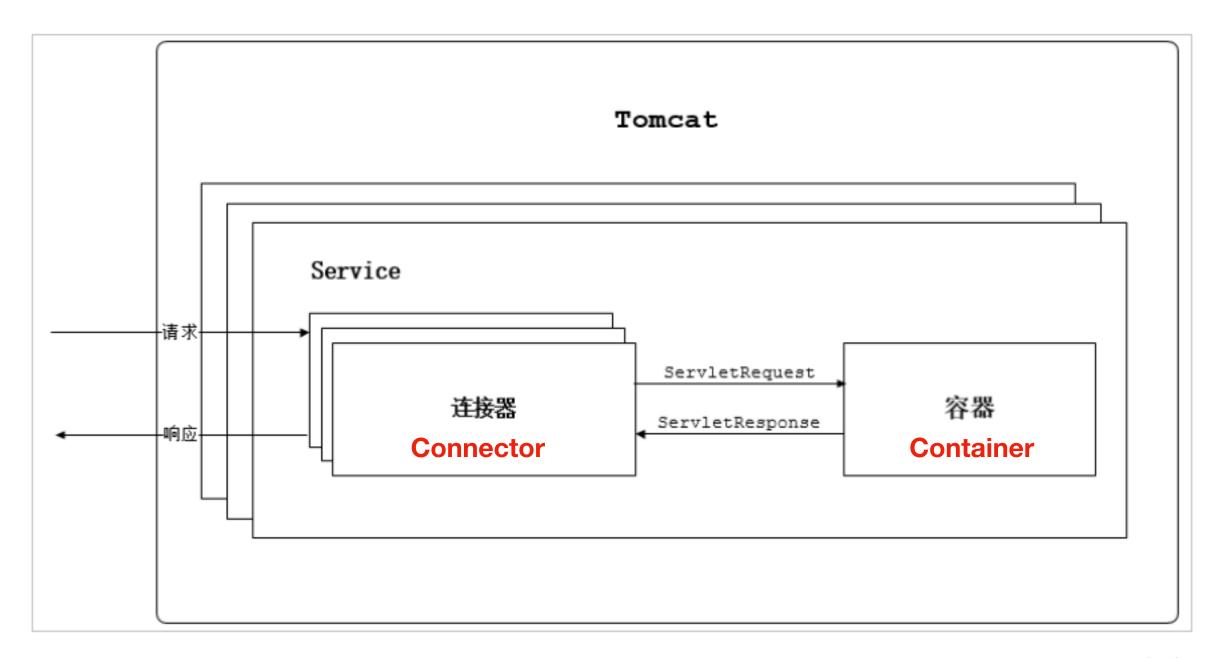
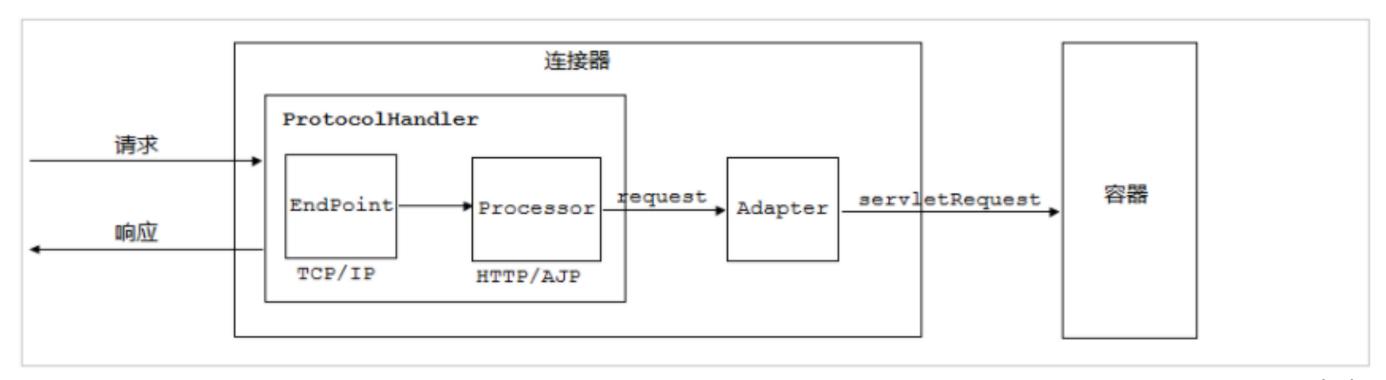


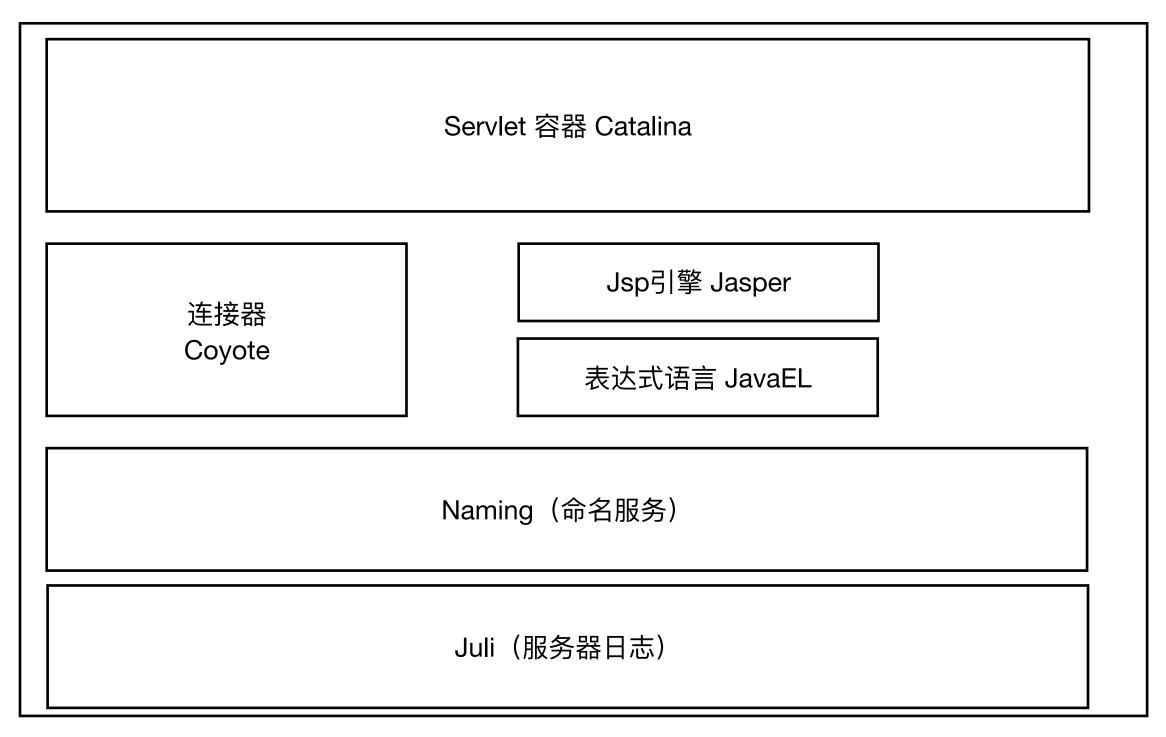
by 应癫



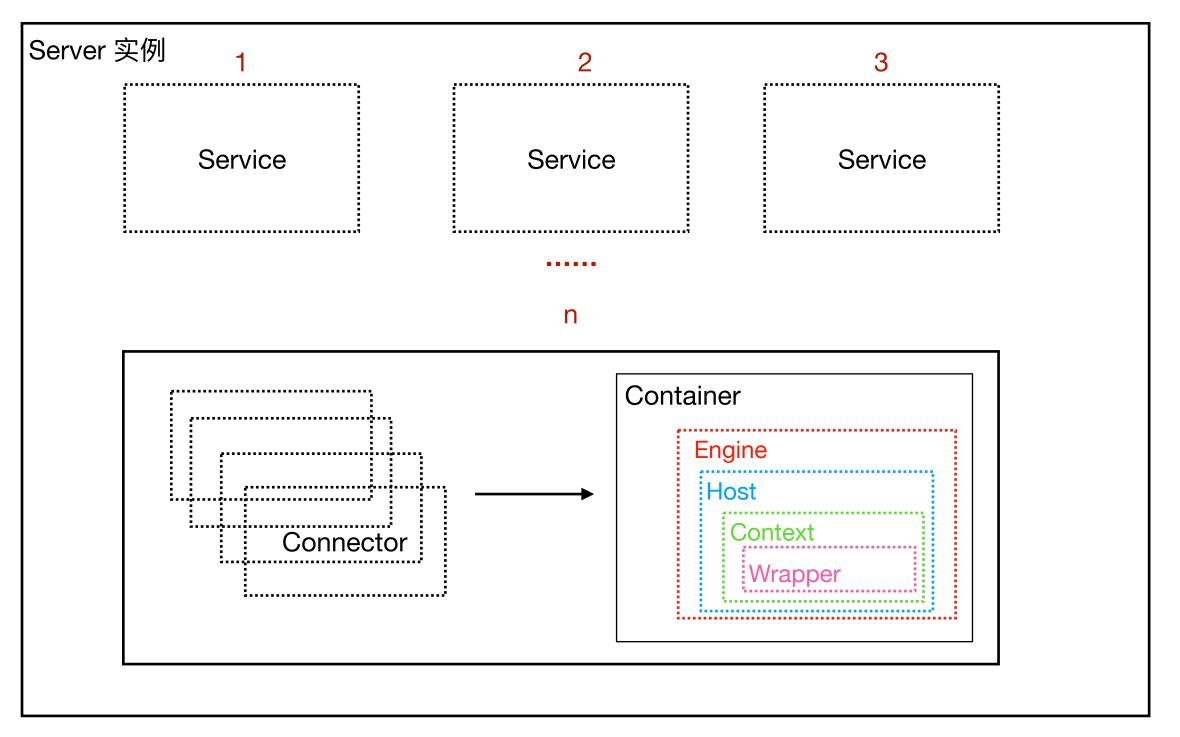
by 应癫



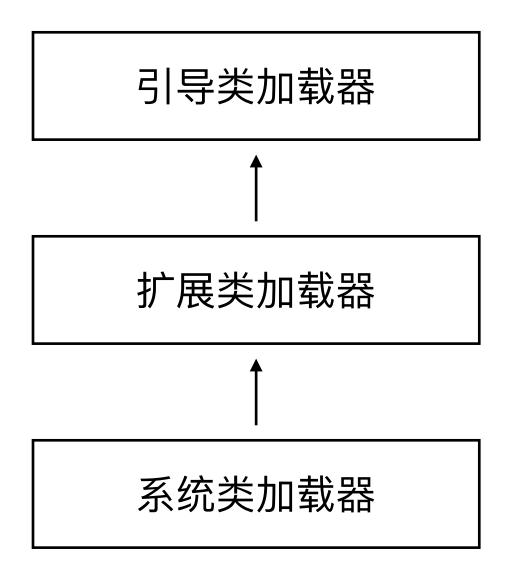
by 应癫



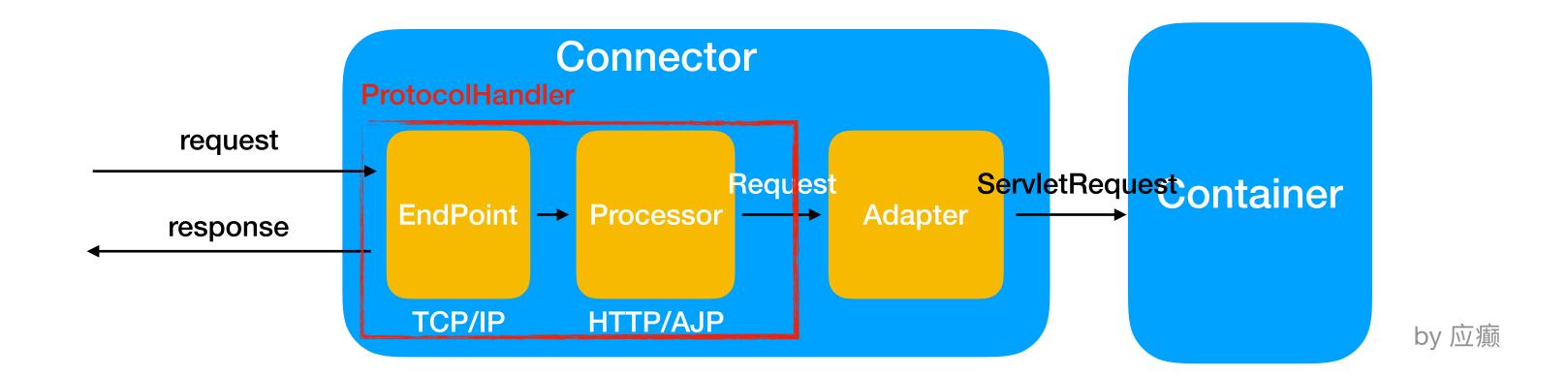
by 应癫

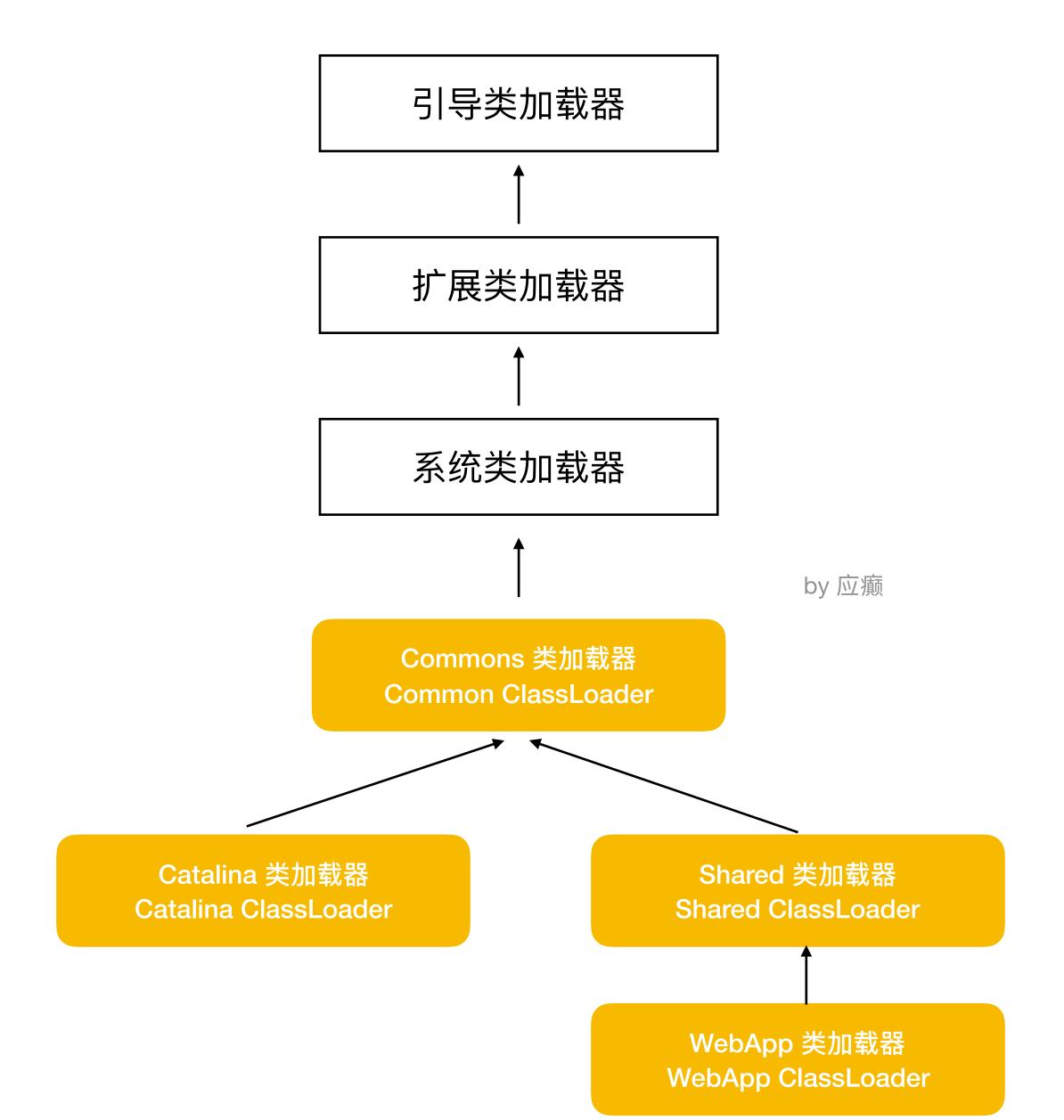


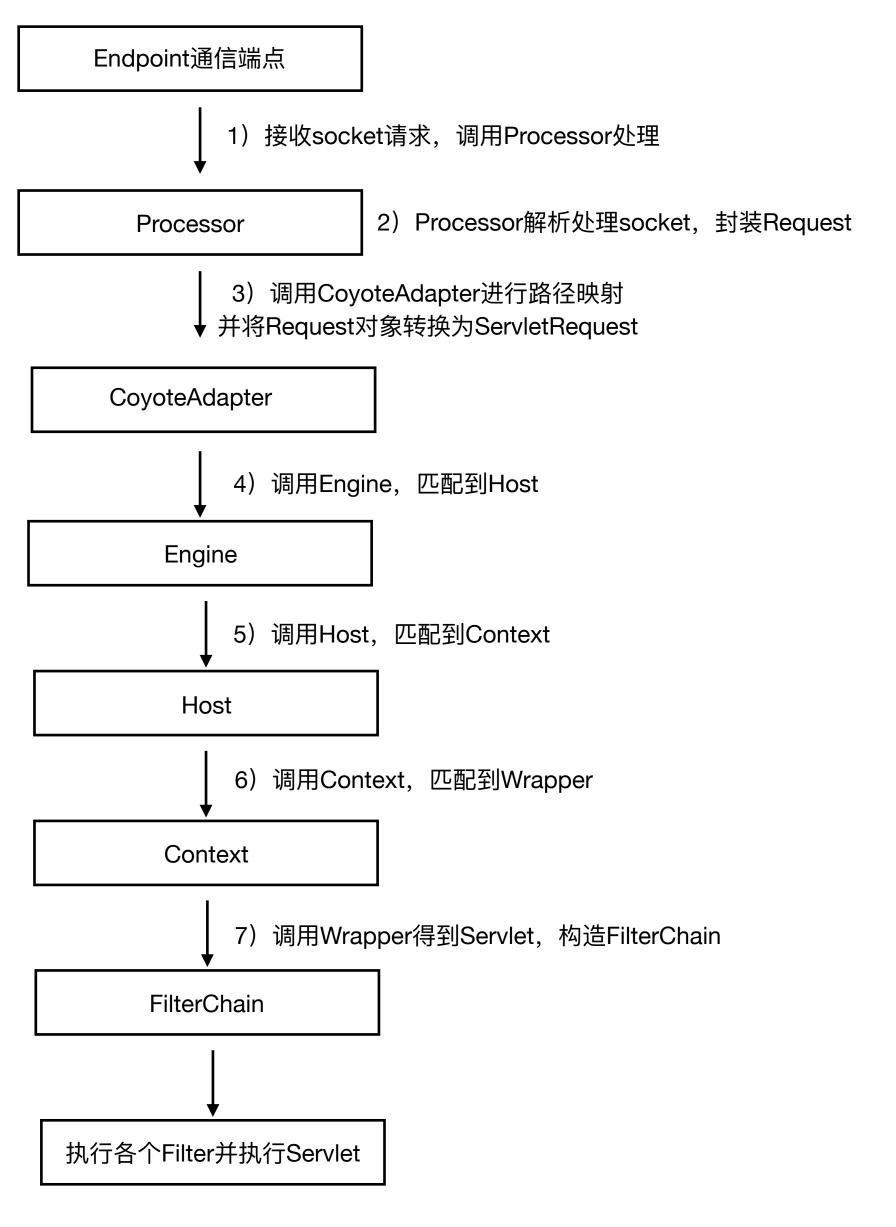
by 应癫

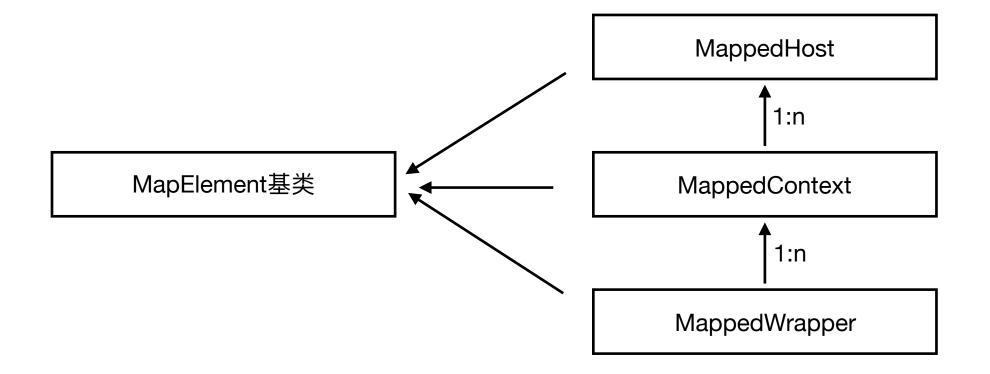


by 应癫









说明:Https在传输数据之前需要客户端与服务端之间进行一次握手,在握手过程中将确定双方加密传输数据的密码信息



- 3、浏览器获得网站证书之后要验证 证书合法性验证证书合法性(办法证书 的机构是否合法,证书中包含的 网站地址是否与正在访问的地址一致等)
- 4、如果证书受信息人,则浏览器 栏中会显示一个小锁头,否则 会给出证书不受信的提示
- 5、如果证书受信任,或者用户接受了不受信的证书,浏览器会生成一串随机数密码比如XXXXXYYYYYY,并用证书中提供的公钥加密

8、浏览器解密并计算握手信息的Hash, ◀

如果与服务器端发来的Hash一直,握手结束

1、浏览器将自己支持的一套加密规则发送给网站

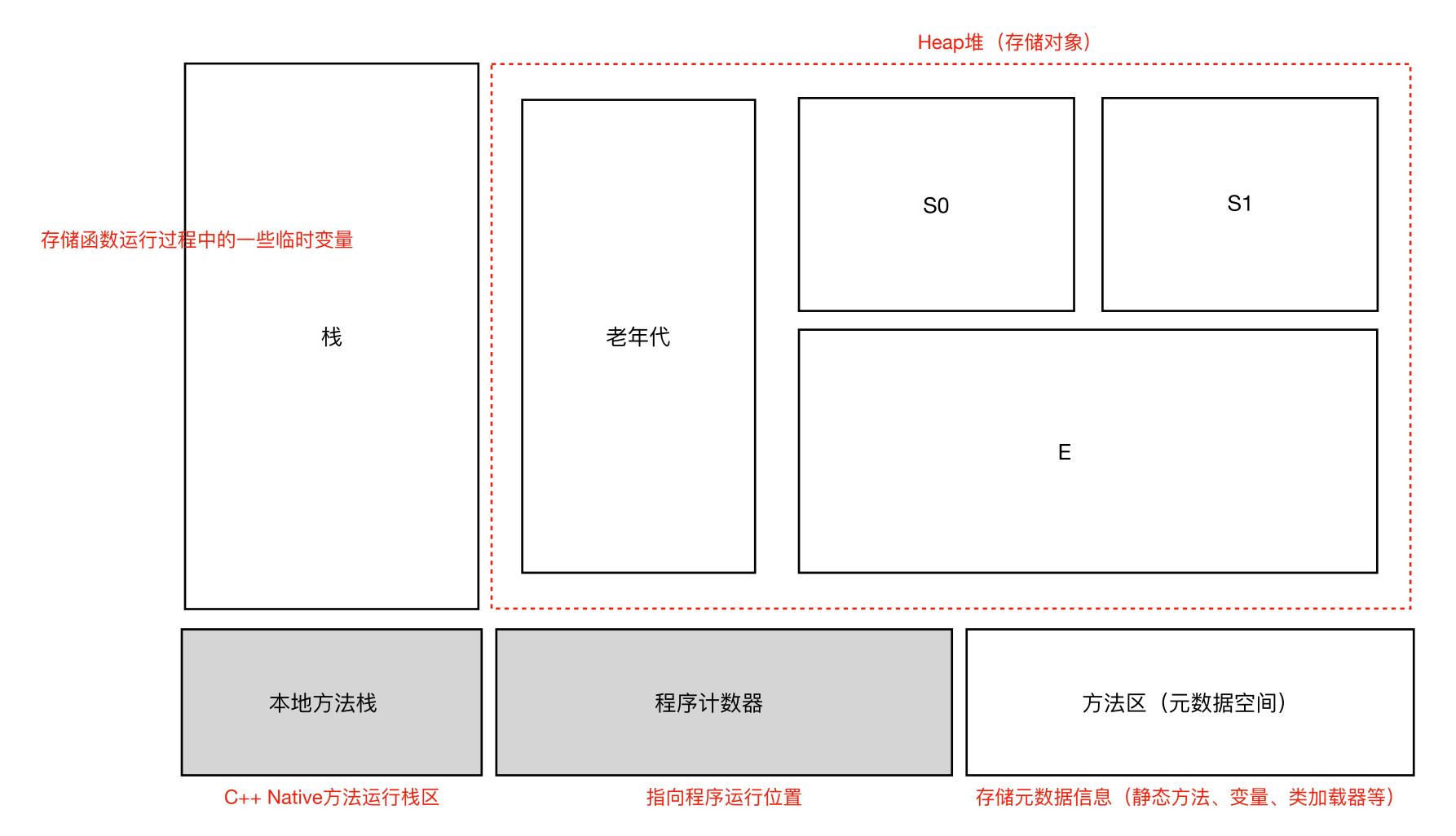
- 2、网站从中选择出一组加密算法,并将自己的身份信息 以证书的形式发回给浏览器。证书中包含了网站地址、加密公钥、 以及证书的颁发机构等信息
  - 6、使用公钥加密后的随机数密码加密握手消息,之后发给网站



7、网站接收浏览器发来的数据之后要做如下操作:

- a、使用私钥解密出密码
- b、使用密码解密握手信息
- c、使用密码再加密一段握手信息,发送给浏览器
- 9、之后所有通信数据由之前浏览器生成的随机密码并利用对称加密算法进行加密

# by 应癫 JVM 内存模型主体示意图



注意: 栈、本地方法栈、程序计数器是线程私有的; 堆区和方法区是全局共享的

### 反向代理需求一:

浏览器请求nginx(http://111.229.248.243:9003),nginx将请求转发给了目标服务器,我们看到的是目标服务器的响应页面,在整个过程中目标服务器相当于对客户端是不可见的,服务端向外暴露的就是Nginx的地址



服务端

#### 反向代理需求二:

在需求一的基础上,目标服务器有两个,分别是: 127.0.0.1:8080,127.0.0.1:8081 当访问http://111.229.248.243:9003/abc 的时候,实际访问目标服务器127.0.0.1:8080 当访问http://111.229.248.243:9003/def 的时候,实际访问目标服务器127.0.0.1:8081



服务端

### 反向代理需求一:

浏览器请求nginx(http://111.229.248.243:9003),nginx将请求转发给了目标服务器,我们看到的是目标服务器的响应页面,在整个过程中目标服务器相当于对客户端是不可见的,服务端向外暴露的就是Nginx的地址



服务端

#### 反向代理需求二:

在需求一的基础上,目标服务器有两个,分别是: 127.0.0.1:8080,127.0.0.1:8081 当访问http://111.229.248.243:9003/abc 的时候,实际访问目标服务器127.0.0.1:8080 当访问http://111.229.248.243:9003/def 的时候,实际访问目标服务器127.0.0.1:8081

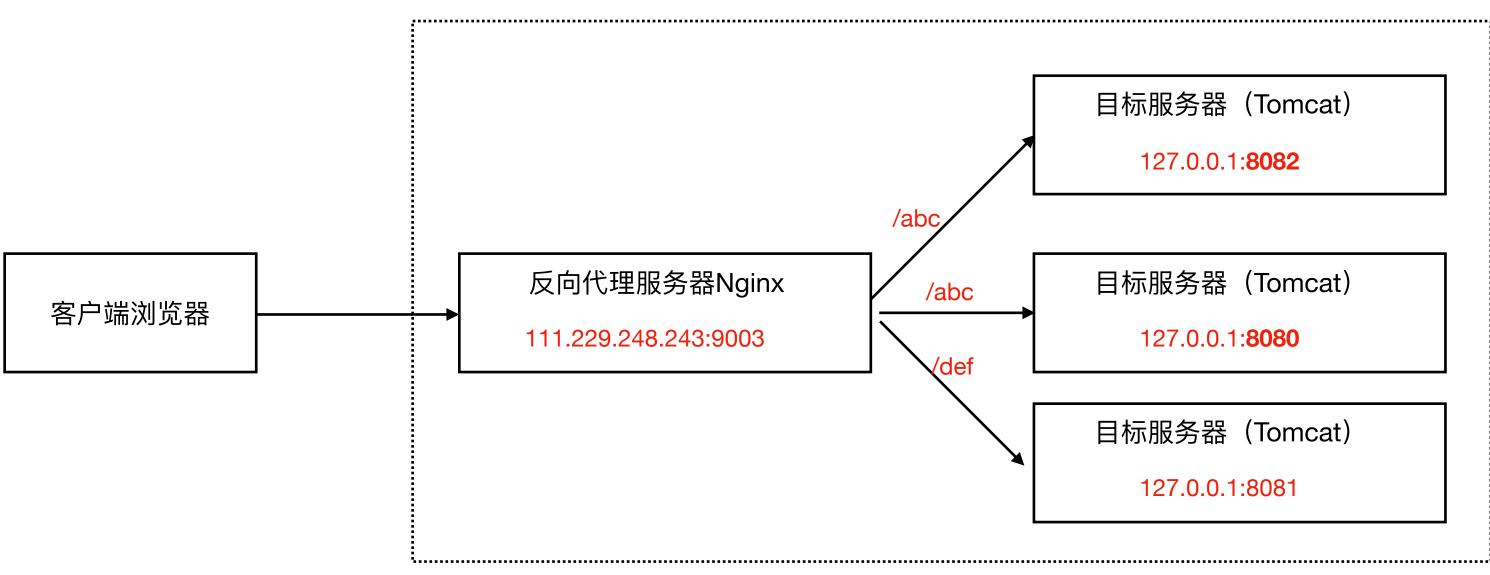


服务端

by 应癫

# 负载均衡需求:

新增一个tomcat监听8082端口,当访问 http://111.229.248.243:9003/abc,使用Nginx作为负载均衡器,将请求分配到 127.0.0.1:8080,127.0.0.1:8082

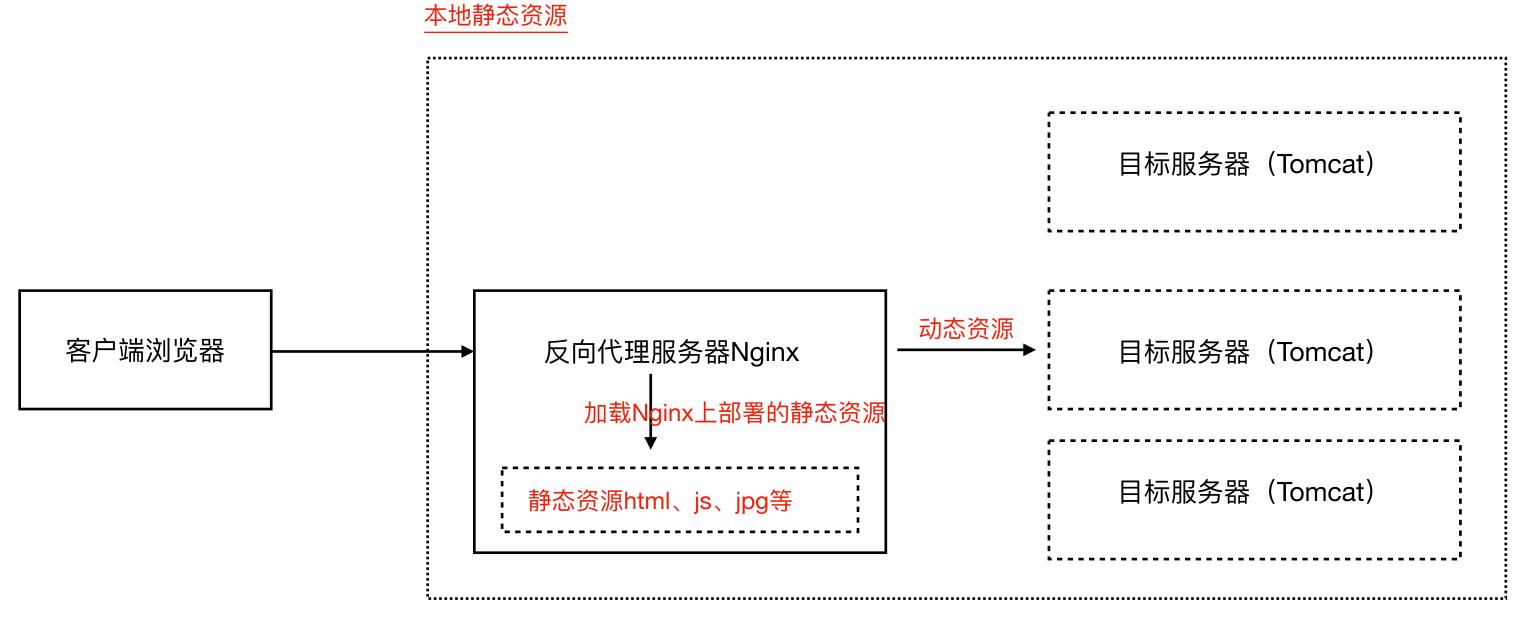


服务端

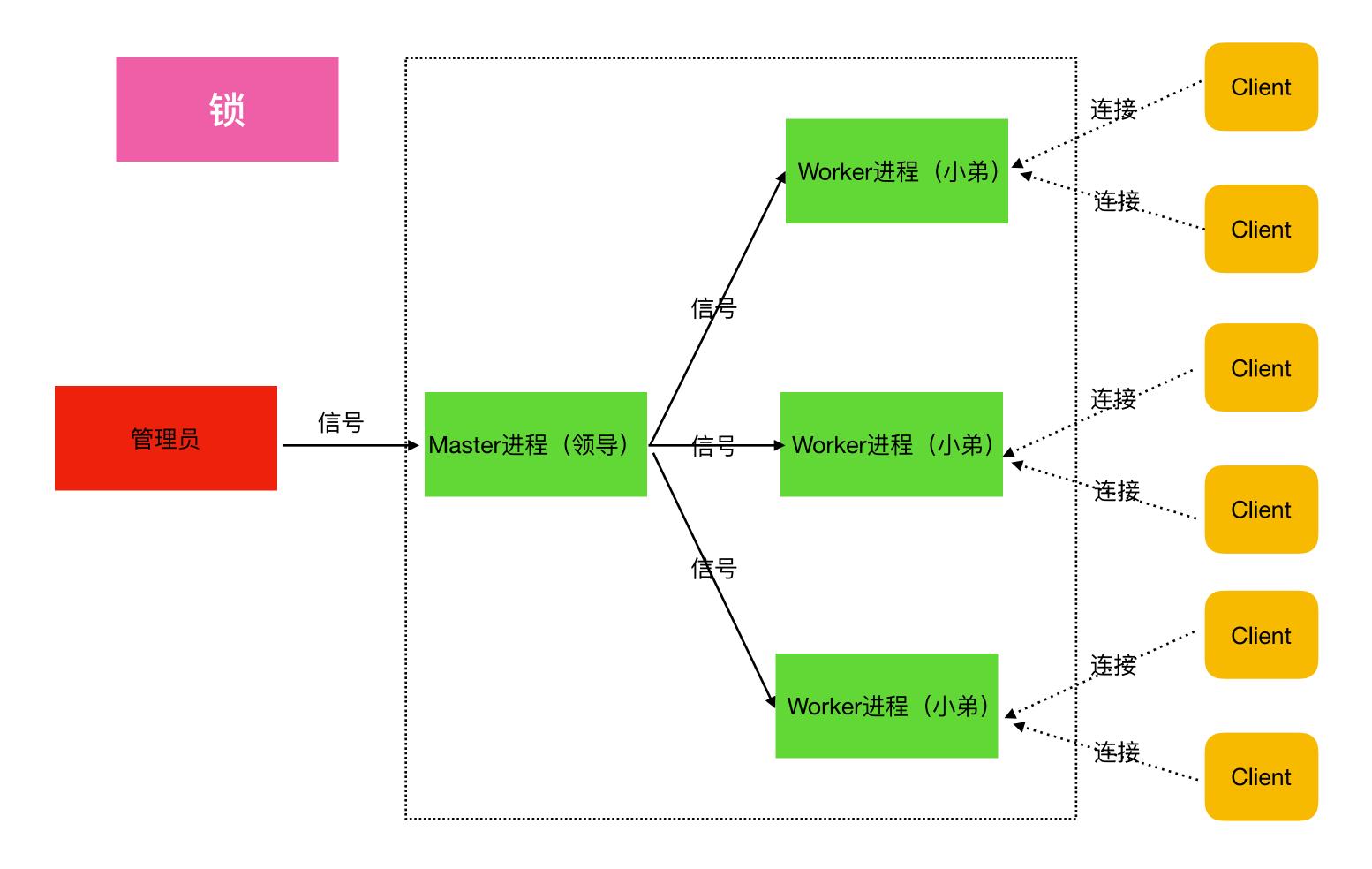
## 动静分离需求:

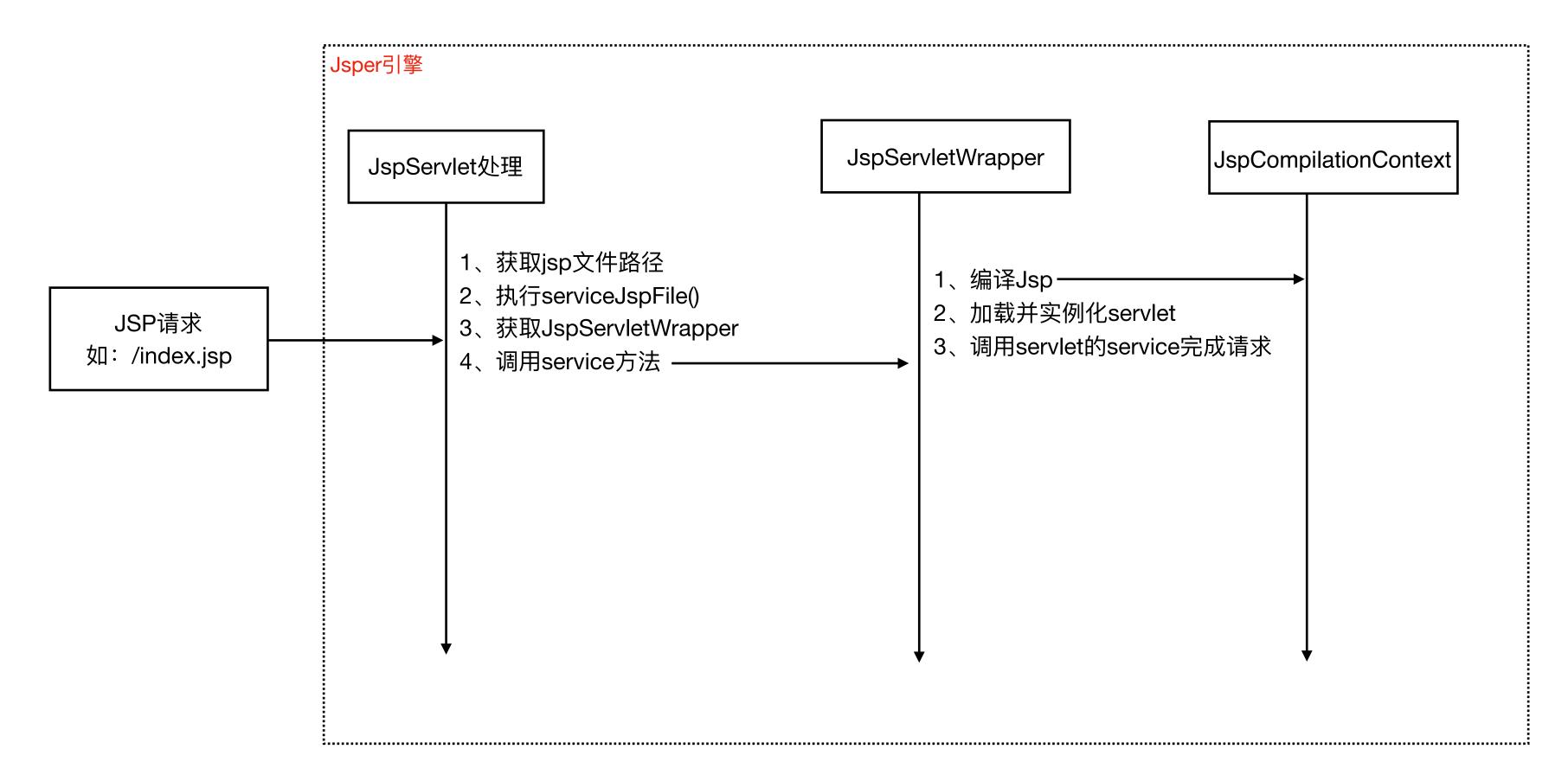
主要是静态资源的访问,因为我们之前Nginx反向代理到Tomcat能够看到Tomcat ROOT项目的index.jsp页面, 这本身就是一个动态资源的请求过程。

我们希望在Nginx服务器上部署静态资源,然后浏览器请求http://111.229.248.243:9003/static/abc.html,Nginx直接读取的本地静态资源



by 应癫





by 应癫

7应用层

这是面向用户的,最靠近用户,为了让用户和计算机交互, 应用层为用户提供了交互的接口,以此为用户提供交互服务。 那么在这一层最常见的协议有:HTTP,HTTPS,FTP,SMTP,POP3等

6表示层

该层提供数据格式编码以及加密功能, 确保请求段的数据能被响应端的应用层识别

5 会话层

请求发送到接受响应的这个过程之间存在会话,会话层就充当了这一过程的管理者, 从创建会话到维护会话最后销毁会话

4 传输层

该层建立端到端的连接,在传输层通信会涉及到端口号,本层常见的协议为TCP、UDP

3 网络层

确定网络通信的时候本机IP和对方的IP

2 数据链路层

为了确保请求正确,提供计算机MAC地址等通信信息

1 物理层

物理介质,比如网线、中继器等等设备