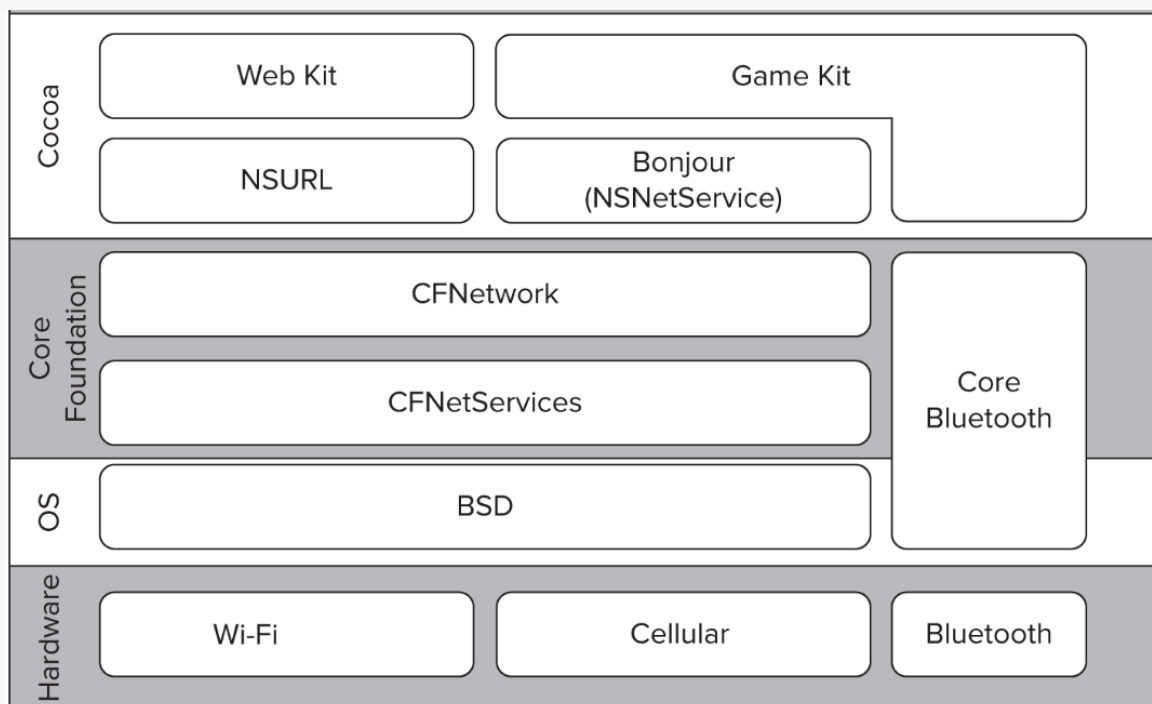


## iOS中的网络框架



- Cocoa层，包含了用于URL加载的OC API、Bonjour、Web Kit与Game Kit
- Core Foundation层，是一套C API，其中包含了CFNetwork，这是大多数应用级别的网络代码的基础。CFNetwork在CFStream与CFSocket之上提供了一个简单地网络接口，是针对BSD socket的轻量级封装
- OS层的BSD socket严格使用C来实现，提供了与远端设备与服务器通信的完全控制能力

从上层框架到下层移动，会获得更为严密的控制，但却也失去了上层提供的易用性与抽象能力。比如，BSD层的socket是无法访问系统范围的VPN，也无法激活Wi-Fi和蜂窝无线网络

## NSURLConnection

NSURLConnection是个Cocoa级别的API，构建在NSSream之上，设计时针对4个常见的URI模式进行了优化支持，文件、HTTP、HTTPS与FTP。NSURLConnection支持同步和异步两种请求

## Game Kit

Game Kit并不需要网络基础设施就能使用，能够创建自组（ad-hoc）的Bluetooth Personal Area Networks（PAN）来实现点对点的通信

---

## Bonjour

Bonjour是Apple对零配置（zeroconf）网络的实现，通过名字、服务类型与域这几个元组来引用服务

---

## NSStream

NSStream是一个Cocoa级别的API，构建在CFNetwork之上，作为NSURLConnection的基础，旨在完成一些底层的网络任务。NSStream可以支持使用telnet或SMTP等NSURLConnection不支持的协议进行通信。NSStream是异步的，通过NSStreamDelegate实现通信更新

---

## CFNetwork

CFNetwork API位于基础的BSD socket之上，与BSD socket之间的主要差别在于CFNetwork集成了运行循环。CFNetwork构建在Core Foundation层的CFSocket和CFStream API之上

---

## HTTP

---

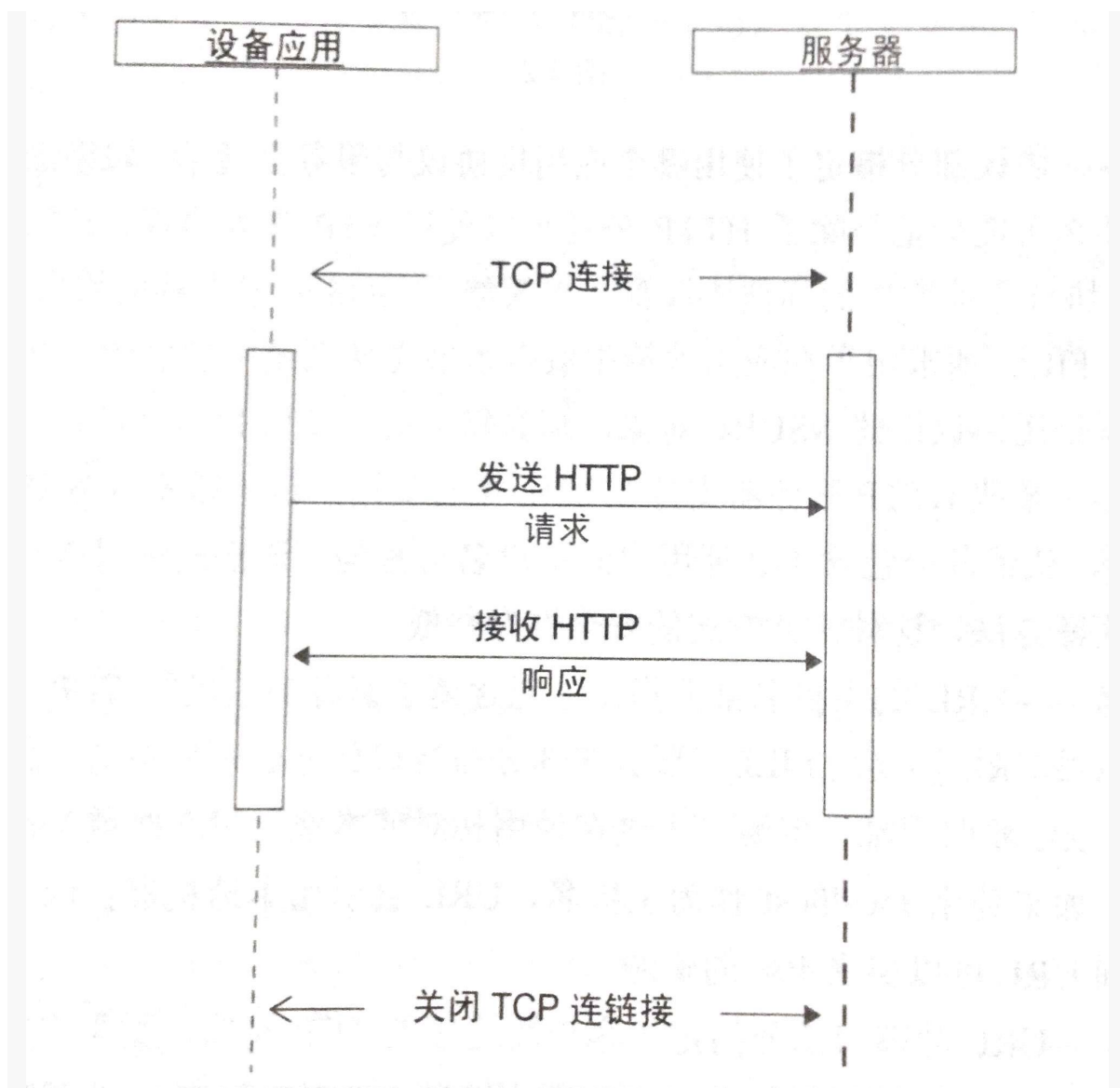
### HTTP介绍

Tim Berners-Lee于1990年创造了HTTP协议的首个版本，最初的的提案中有3个主要的创新：HTML、HTTP与URL。

- HTML定义了向文本添加样式的一种方式
- HTTP定义了在服务端与客户端之间传输数据的一种方式
- URL定义了在网络机器中定位唯一资源的一种方式

---

### HTTP请求与响应



HTTP的规范是IETF FRF 2616 <http://www.ietf.org/rfc/rfc2616.txt>

## URL结构

`http://user:password@hostname:port/absolute-path?query`

协议:[认证]主机名:端口/绝对路径/?查询字符串1&查询字符串2

绝对路径和查询字符串不可以包含空格、回车与换行符，因此，**URL**通常会使用百分号进行编码，**RFC 3986**(<http://tools.ietf.org/html/rfc3986>)规定了**URL**百分号编码的详细信息

## 请求内容

HTTP请求包含3部分：请求行、请求头与请求体

HTTP是无状态的协议，即HTTP服务器不会保留关于某个请求的任何信息以用在未来的请求中。**Cookie**机制提供了一种方式，可以将一些简单地状态信息存储在客户端，并在后续的请求中与服务器进行通信

请求体必须遵循客户端与服务器之间预先确定的数据编码，对于**Web**浏览器来说，这通常是表  
单编码数据，但对于移动应用来说，该编码通常是XML或JSON数据

## 响应内容

HTTP响应包含**3**部分：状态行、响应头与相应体

在iOS的URL加载系统中，**NSURLResponse**及其子类**NSHTTPURLResponse**封装了请求返回的数据

---

## Cocoa层 iOS HTTP API

在URL加载系统中，有**3**个主要的方式可以执行HTTP请求和接收响应

- 同步 （线程会被阻塞）
- 队列式异步 （创建一个队列并放入后台线程执行）
- 异步 （调用委托方法）

所有的URL加载请求方式都会用到这**4**类对象

NSURL、NSURLRequest、NSURLConnection与NSURLResponse对象