

我选择的软件是微信，作为一款多数人都在使用的即时通讯软件，微信的特点其实不太需要解释，我这里查询了有关微信的百科资料来进行分析。

微信（WeChat）是腾讯公司于 2011 年 1 月 21 日推出的一个为智能终端提供即时通讯服务的免费应用程序，由张小龙所带领的腾讯广州研发中心产品团队打造。微信支持跨通信运营商、跨操作系统平台通过网络快速发送免费（需消耗少量网络流量）语音短信、视频、图片和文字，同时，也可以使用通过共享流媒体内容的资料和基于位置的社交插件“摇一摇”、“朋友圈”、“公众平台”、“语音记事本”等服务插件。截止到 2016 年第二季度，微信已经覆盖中国 94% 以上的智能手机，月活跃用户达到 8.06 亿，用户覆盖 200 多个国家、超过 20 种语言。此外，各品牌的微信公众账号总数已经超过 800 万个，移动应用对接数量超过 85000 个，广告收入增至 36.79 亿人民币，微信支付用户则达到了 4 亿左右。微信提供公众平台、朋友圈、消息推送等功能，用户可以通过“摇一摇”、“搜索号码”、“附近的人”、扫二维码方式添加好友和关注公众平台，同时微信将内容分享给好友以及将用户看到的精彩内容分享到微信朋友圈。

由以上可知，微信的用户基数非常大，本身就集成了很多功能，并且允许第三方开发微信小程序来扩展服务，那么可想而知，它的架构一定是非常规范的模块化的，而且粒度很细。据微信开发团队的发布会发言内容，微信采取的是敏捷开发模

式，项目管理者对开发过程中的变更有极大的容忍度，相较于常规开发模式，这样这有利于创建一个充满活力的聊天软件，因为它具有非常高的自由度灵活度。但是微信同时具有很大的用户基数，这也意味着敏捷开发的试错成本会高很多，为此，微信的软件框架模块化分布，把大系统做小，采用扩展概念进行改进，较细的粒度划分也使敏捷开发更加如鱼得水。基于若干基础服务来实现新功能使开发流程具有基础的稳定度。由此可以看出软件架构的几个关键如下：

细粒度：当设计庞大系统的时候，应该尽量分割成更小的颗粒，使得项目之间的影响是最小的。

一切可扩展：在高稳定度、高性能的系统中间，为了稳定性能把它设计成不变化的系统，但为了支持敏捷需要让一切的东西都要变得可以扩展。

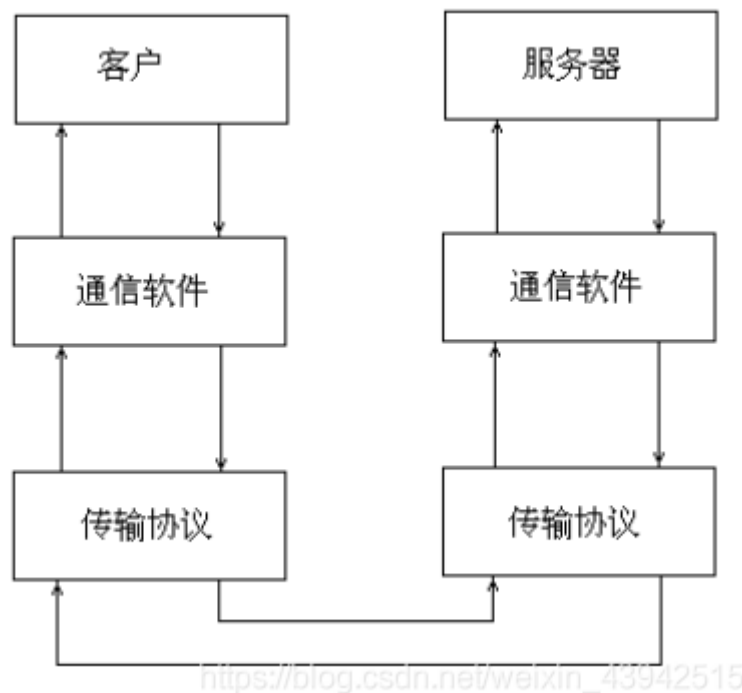
必须建立基础组件：要解决复杂问题的时候，需要将已有的经验固化下来，固化下来的东西会成为系统中的一部分。

分层开发：将不同的应用逻辑物理分割独立出来，用户注册登录、LBS 逻辑、摇一摇逻辑、漂流瓶逻辑、消息逻辑独立开来。把关键的逻辑混搭在一起，当所有的逻辑部署在同一个服务器上，确实也会带来很大敏捷上的好处，因为不需要额外的考虑部署和监控的问题。

当然，任何一个成熟的软件都不会从造轮子开始，在微信后台会有几种不同的基础组件。大致包括：Svrkit——

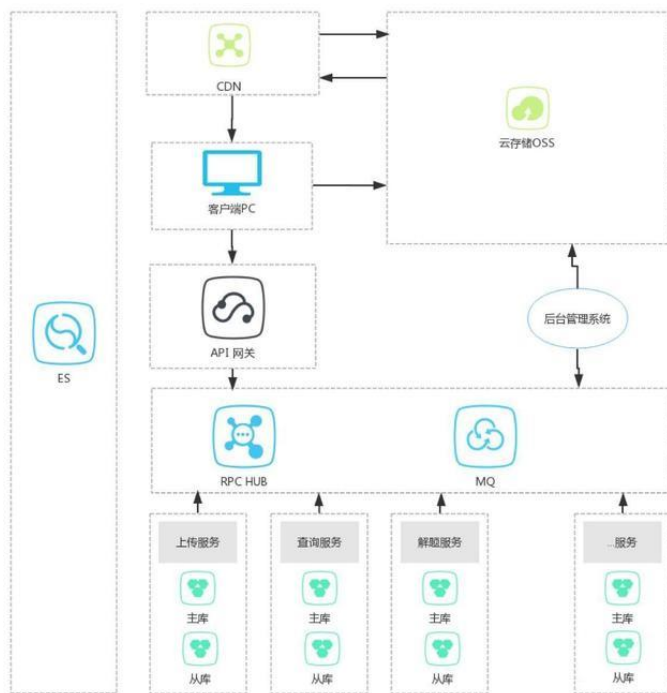
Client/Server 自动代码生成框架:10 分钟搭建内部服务器

LogicServer——逻辑容器：随时添加新逻辑 OssAgent——监控/统计框架：所见即所得的监控报表存储组件——屏蔽容灾/扩容等复杂问题。



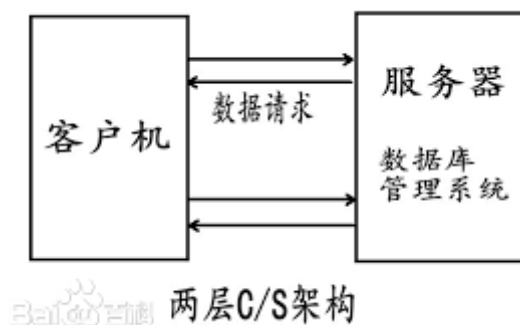
Client/Server 框架

微信对于功能划分采用了微服务模式，把业务分块，做了垂直切分，切成一个个独立的系统，每个系统各自衍化，有自己的库、缓存、ES 等辅助系统，系统之间的实时交互通过 RPC，异步交互通过 MQ，通过这种组合，共同完成整个系统功能。相对于一般数据模式，微服务模式具有高性能，可扩展性强，高可用的优点。



微服务模式

作为通讯软件，必然离不开客户机/服务器的概念，也就是所谓的 C/S 架构，它是软件系统体系结构，通过它可以充分利用两端硬件环境的优势，将任务合理分配到 Client 端和 Server 端来实现，降低了系统的通讯开销。



客户机通过局域网与服务器相连，接受用户的请求，并通过网络向服务器提出请求，对数据库进行操作。服务器接受客户机的请求，将数据提交给客户机，客户机将数据进行计算并将

结果呈现给用户。服务器还要提供完善安全保护及对数据完整性的处理等操作，并允许多个客户机同时访问服务器，这就对服务器的硬件处理数据能力提出了很高的要求。在 C/S 结构中，应用程序分为两部分：服务器部分和客户机部分。服务器部分是多个用户共享的信息与功能，执行后台服务，如控制共享数据库的操作等；客户机部分为用户所专有，负责执行前台功能，在出错提示、在线帮助等方面都有强大的功能，并且可以在子程序间自由切换。

C/S 结构在技术上已经很成熟，它的主要特点是交互性强、具有安全的存取模式、响应速度快、利于处理大量数据。但是 C/S 结构缺少通用性，系统维护、升级需要重新设计和开发，增加了维护和管理难度，进一步的数据拓展困难较多，所以 C/S 结构只限于小型的局域网。

优点

C/S 结构的优点是能充分发挥客户端 PC 的处理能力，很多工作可以在客户端处理后再提交给服务器。对应的优点就是客户端响应速度快。具体表现在以下两点：

(1) 应用服务器运行数据负荷较轻。最简单的 C/S 体系结构的数据库应用由两部分组成，即客户应用程序和数据库服务器程序。二者可分别称为前台程序与后台程序。运行数据库服务器程序的机器，也称为应用服务器。一旦服务器程序被启动，就随时等待响应客户程序发来的请求；客户应用程序运行

在用户自己的电脑上，对应于数据库服务器，可称为客户电脑，当需要对数据库中的数据进行任何操作时，客户程序就自动地寻找服务器程序，并向其发出请求，服务器程序根据预定的规则作出应答，送回结果，应用服务器运行数据负荷较轻。

(2) 数据的储存管理功能较为透明。在数据库应用中，数据的储存管理功能，是由服务器程序和客户应用程序分别独立进行的，并且通常把那些不同的（不管是已知还是未知的）前台应用所不能违反的规则，在服务器程序中集中实现，例如访问者的权限，编号可以重复、必须有客户才能建立订单这样的规则。所有这些，对于工作在前台程序上的最终用户，是“透明”的，他们无须过问背后的过程，就可以完成自己的一切工作。在客户服务器架构的应用中，前台程序不是非常“瘦小”，麻烦的事情都交给了服务器和网络。

缺点

随着互联网的飞速发展，移动办公和分布式办公越来越普及，这需要我们的系统具有扩展性。这种方式远程访问需要专门的技术，同时要对系统进行专门的设计来处理分布式的数据。

C/S 架构的劣势之一是高昂的维护成本且投资大。客户端需要安装专用的客户端软件。首先涉及到安装的工作量，其次任何一台电脑出问题，如病毒、硬件损坏，都需要进行安装或维护。特别是有很多分部或专卖店的情况，不是工作量的问题，

而是路程的问题。还有，系统软件升级时，每一台客户机需要重新安装，其维护和升级成本非常高。

传统的 C/S 体系结构虽然采用的是开放模式，但这只是系统开发一级的开放性，在特定的应用中无论是 Client 端还是 Server 端都还需要特定的软件支持，不够自由。