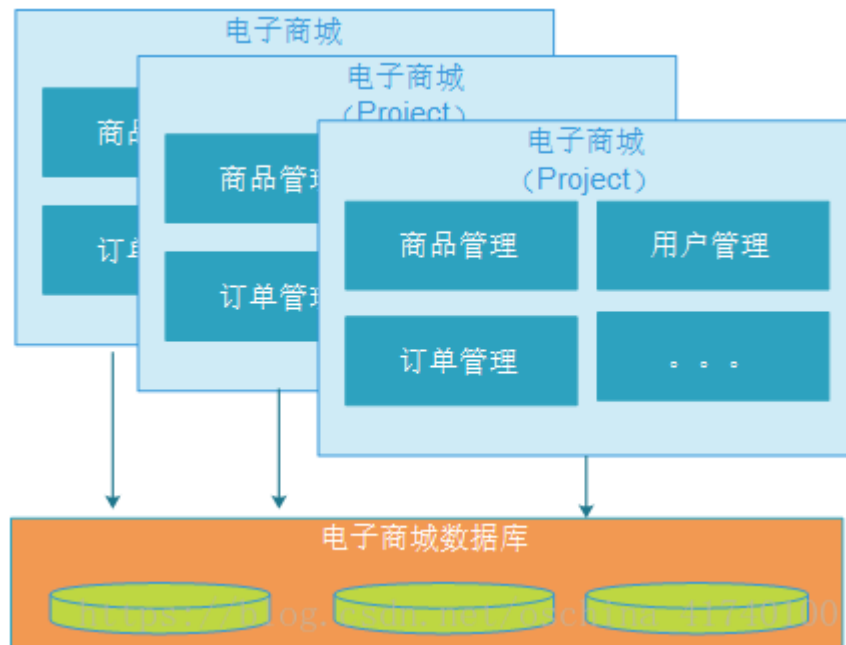


软件架构的演进

(由于内容主要借鉴于网络博文，如有雷同，不胜荣幸)

概括：软件架构的发展经历了单体结构、垂直结构、SOA架构到微服务架构的过程。

单体架构



特点：

- 1、所有的功能集成在一个项目工程中。
- 2、所有的功能打一个war包部署到服务器。
- 3、应用与数据库分开部署。
- 4、通过部署应用集群和数据库集群来提高系统的性能。

优点：

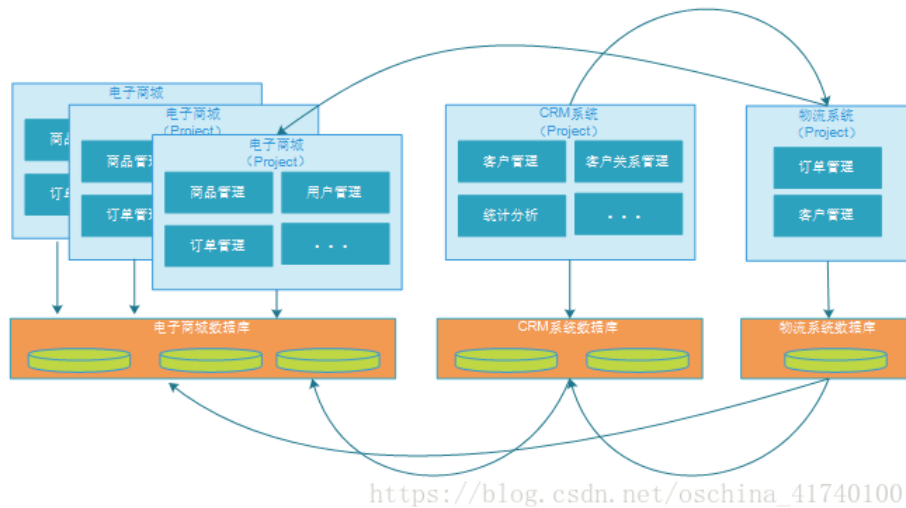
项目架构简单，前期开发成本低，周期短，小型项目的首选。

缺点：

- 1、全部功能集成在一个工程中，对于大型项目不易开发、扩展及维护。
- 2、系统性能扩展只能通过扩展集群结点，成本高、有瓶颈。

3、技术栈受限。

垂直架构



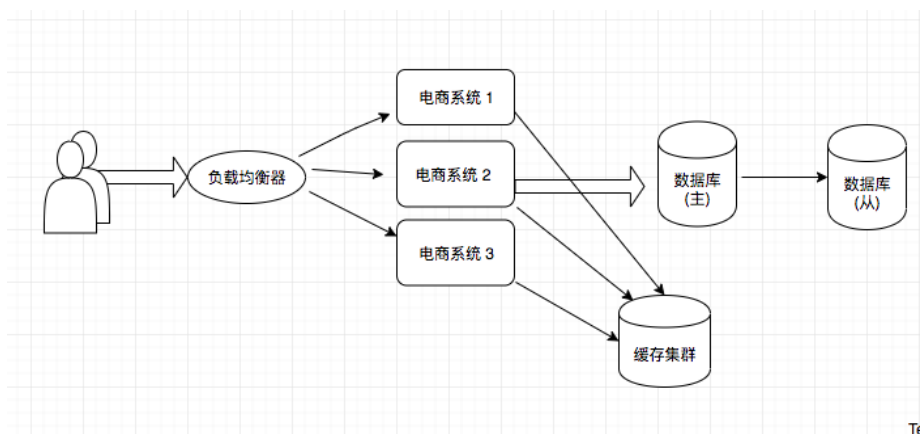
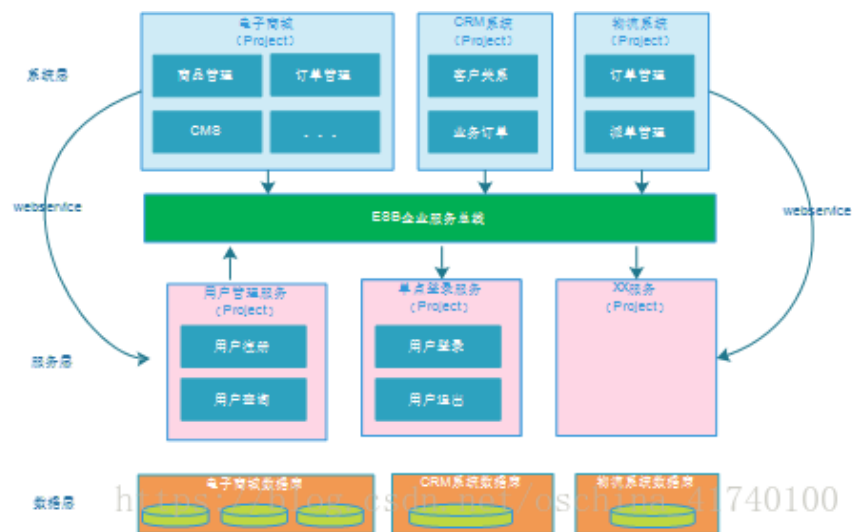
特点：

- 1、以单体结构规模的项目为单位进行垂直划分项目即将一个大项目拆分成一个单体结构项目。
- 2、项目与项目之间的存在数据冗余，耦合性较大，比如上图中三个项目都存在客户信息。
- 3、项目之间的接口多为数据同步功能，如：数据库之间的数据库，通过网络接口进行数据库同步。

缺点：

- 1、全部功能集成在一个工程中，对于大型项目不易开发、扩展及维护。
- 2、系统性能扩展只能通过扩展集群结点，成本高、有瓶颈。

SOA架构



特点:

- 1、基于SOA的架构思想将重复公用的功能抽取为组件，以服务的方式给各系统提供服务。
- 2、各各项目（系统）与服务之间采用webservice、rpc等方式进行通信。
- 3、ESB企业服务总线作为项目与服务之间通信的桥梁。

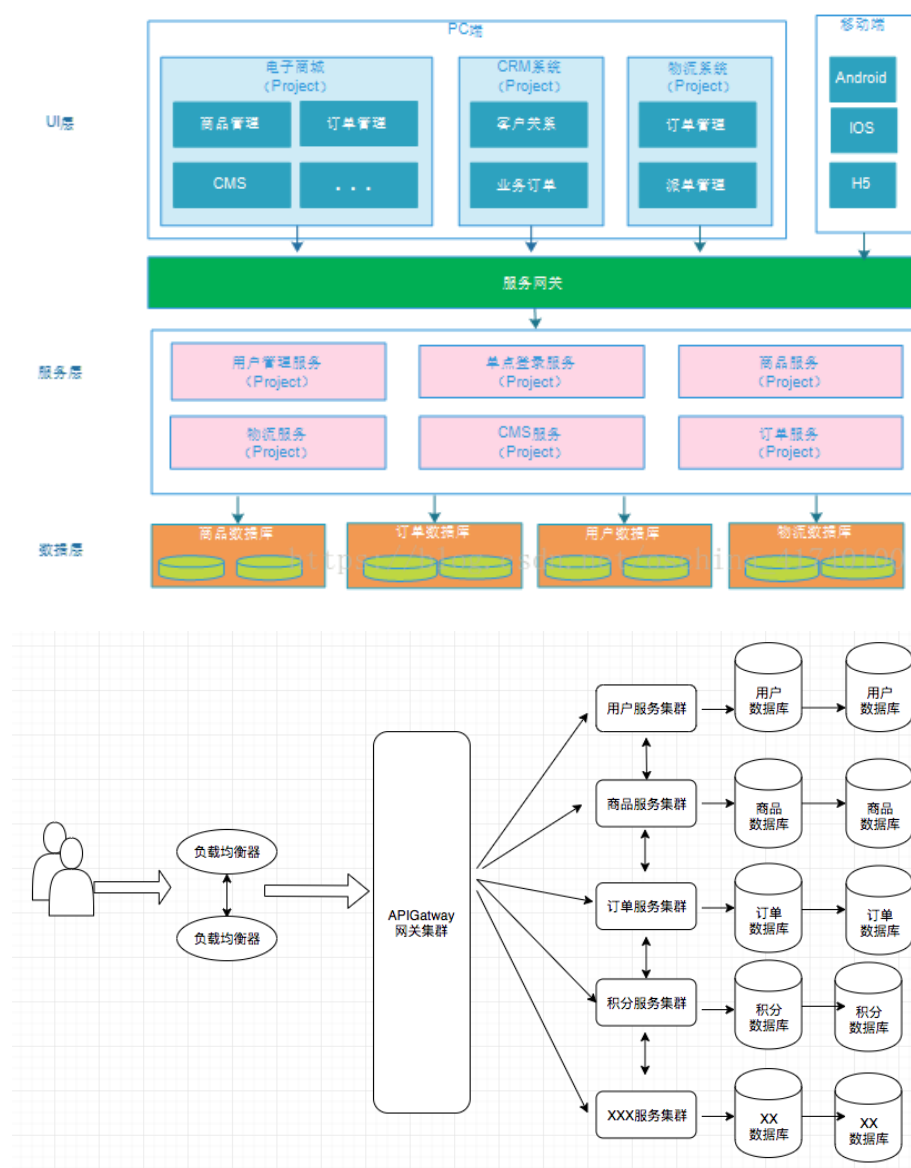
优点:

- 1、将重复的功能抽取为服务，提高开发效率，提高系统的可重用性、可维护性。
- 2、可以针对不同服务的特点制定集群及优化方案。
- 3、把模块拆分，使用接口通信，降低模块之间的耦合度。
- 4、可以灵活的进行分布式部署。
- 5、把项目拆分成若干个子项目，不同的团队负责不同的子项目
- 6、增加功能时只需要在增加一个子项目，调用其它系统的接口就可以

缺点：

- 1、系统与服务的界限模糊，不利于开发及维护。
- 2、虽然使用了ESB，但是服务的接口协议不固定，种类繁多，不利于系统维护。
- 3、抽取的服务的粒度过大，系统与服务之间耦合性高。

微服务架构



特点：

- 1、将系统服务层完全独立出来，并将服务层抽取为一个一个的微服务。
- 2、微服务遵循单一原则。
- 3、微服务之间采用RESTful等轻量协议传输。

优点：

- 1、服务拆分粒度更细，有利于资源重复利用，提高开发效率。
- 2、可以更加精准的制定每个服务的优化方案，提高系统可维护性。
- 3、微服务架构采用去中心化思想，服务之间采用RESTful等轻量协议通信，相比ESB更轻量。
- 4、适用于互联网时代，产品迭代周期更短。

缺点：

- 1、微服务过多，服务治理成本高，不利于系统维护。
- 2、分布式系统开发的技术成本高（容错、分布式事务等），对团队挑战大。