

软件即为服务

虽然Web网页也提供了因特网和组织界限访问应用的方式，但Web服务与Web网页并不一样。Web网页直接面向的是人，而Web服务的开发目标是访问者既可以是人也可以是自动化的应用程序。因此，分析一下“软件即为服务”的理念是非常有价值的，这个理念也是Web服务技术的基础。

“软件即为服务”这一理念非常新颖，它首先产生于应用服务提供商软件模型中，应用服务提供商是将软件、基础设施要素、业务以及专业的服务进行打包的公司，它们部署、维护并管理打包的应用程序，并通过提供集中管理方式，对跨网络的客户提供应用程序可用性、安全性。以订阅或租赁的方式在网络上提供应用软件，最终用户可以使用因特网或专线远程访问这些应用软件。本质上，ASP是一些公司将他们把对信息技术方面的部分甚至整个需求外包出去的一种方式。

ASP的基本出发点是向用户出租应用程序。整个应用程序的开发包括用户界面、 workflow、业务和数据组件，将这些组件捆绑在一起，并作为一个有效的解决方案提交给用户。ASP管理整个应用系统，用户除了生成一些表格，确定用户界面的外观等，基本无法定制应用软件。通过可供下载的报表，用户仅需要浏览以及进行人工订阅、交易，即可访问软件。对于企业来说，这意味着ASP在维护应用系统、相关的基础设施以及客户数据，并保障在任何需求的时候都可使用系统和数据。

ASP也可以采用另外一种方式提供软件模块，即用户可以根据自身的需求将软件模块下载到自己的站点。当软件无法以客户/服务器模式工作时，或者软件无法通过浏览器远程工作时，可采用这一方式。在一段时间后，可删除这些软件模块，也可以一直将这些软件模块保留在客户的计算机中，直到有新版本可以取代它们。也可能是合同到期，用户不能再保留这些软件。

虽然ASP模型首先引入了“软件即为服务”的理念，但是它也受到一些固有的束缚，如无法开发高度交互的应用程序；无法提供完全定制的应用；无法整合各种应用。这些紧耦合的方式将会导致一些必然的后果，如体系结构单一、比较脆弱、仅能针对特定客户、没有可复用的应用集成。

目前，“软件即为服务”这一理念正被进一步发展。新的体系架构支持松耦合方式的异步交互，这些异步交互基于XML标准，从而更容易访问因特网上的应用程序，这些应用程序之间也更容易通信。

Web服务规范进一步扩展了“软件即为服务”的理念，可将复杂的业务流程和事务也视为服务。

鉴于Web服务技术的一些优点，许多ASP修改了他们的技术基础架构和业务模型，变得更类似于Web服务提供者。Web服务给ASP提供了更加灵活的解决方案。业务和数据组件作为应用程序的核心，目前仍然驻留在ASP的机器上，但是可以通过Web服务接口可编程地访问。客户可以构建他们自身的特定的业务流程和用户接口，并且可以从网络上自由选择那些满足他们的业务需求的各类Web服务。

如果将Web服务与基于Web的应用程序进行比较，可以发现有四方面的显著差异：

1、对于请求或调用Web服务的应用程序而言，无论这种调用是否需要人的干预，请求或者调用的Web服务都可视为应用程序的资源。这意味着Web服务可以调用其他的Web服务，从而将复杂的事务中的一些处理交由其他Web服务实现。这提供了基于Web的应用目前无法达到的高度灵活性和适应性。

2、Web服务是模块化的、自感知和自描述的应用程序。Web服务知道它能完成什么功能，也知道何种输入会产生何种输出，并将其向潜在

3、相比于基于Web的应用程序，Web服务更容易被监控和管理。可在任何时候使用外部的应用管理和工作流系统来监控和管理Web服务的状态。尽管Web服务可能不在内部系统上运行，或者我们不熟悉编写Web服务所用的语言，但是本地应用程序依然可以使用这些Web服务。

4、可对Web服务进行评估和拍卖。假如几个Web服务完成同样的任务，Web服务可对所使用的服务进行招标。代理可基于Web服务的“竞价”属性进行选择。

本博客文章除特别声明，全部都是原创！

原创文章版权归过往记忆大数据（[过往记忆](#)）所有，未经许可不得转载。

本文链接: [【】](#)（）