# Summit架构分析

在2018年的6月，美国能源部在橡树岭国家实验室正式宣布了全新的超级计算机——Summit。

Summit系统前端使用C#开发，因此，只能在Windows环境下运行。界面相当友好。后端使用C/C++开发，由于历史遗留问题(毕竟90年代的系统)，几乎所有底层API都是使用C写的，后来C++出现后，只是在外围包裹了一层C++而已。Summit系统还有一个中间层，用来进行前/后端通讯。

Summit系统是典型的MVC结构系统，其中View层称为SummitFT，基于微软C# .NET技术；Control层分为2部分，一部分为Java开发的通信中间层，另一部分为C/C++编写的Summit主体部分；最后，Model层作为Summit业务数据抽象、存取层，基于ENTITY实现，支持主流的Oracle/SQL Server以及Sybase数据库。

SummitFT使用Infragistics的C#控件库作为基础，封装出了一套自己的控件。整个界面风格统一、控件布局合理，操作方便，对用户比较友好。在Control层，Summit作为典型的CS程序，客户端与服务端通讯采用了HTTP协议和WebService的方式。其中，SummitFT通过HTTP协议与通讯中件层通讯；通讯中间层采用WebService与etoolkit进程通讯，达到使用Summit后端服务的目的。这种设计的好处就是Control层不仅可以对接SummitFT，还提供了一套灵活的供其他客户端调用的方式，比如Summit就支持VBA、Java等其他语言的直接调用。由此可以看出，Summit系统在设计时已经考虑到了系统的开放性。通讯中间层采用Java语言编写，负责接收SummitFT的HTTP连接，并负责HTTP协议报文与SOAP报文之间的转换。Summit Business Control层即上文提到的etoolkit，etoolkit使用C/C++开发，实际上就是一个WebService Server，负责处理中间层的请求，并将结果封闭成SOAP报文，返回给通讯中间层。

Model层依赖Summit数据抽象ENTITY以及关系型数据库，目前支持Oracle, Sybase以及SQL Server。Model层进行Summit数据的序列化与反序列化。ENTITY即Summit系统的元数据，在Summit系统中，所有的数据（交易数据、静态数据、系统基础数据）都以ENTITY进行抽象。ENTITY不仅包含属性(Properties)，还会包含接口(Interface)和具体的方法(Method)。