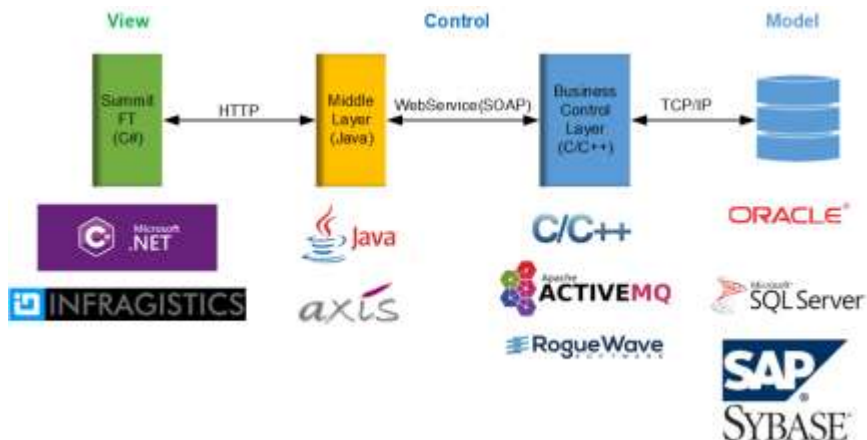


一、Summit 系统架构概览：



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。下面，我们从前端到后端，对 Summit 使用的主要技术进行介绍。

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。作为对比，Calypso 基于 Java 做的界面；Kondor 基于 C 做的界面，操作体验上来说，跟 SummitFT 是没法比。

我们再来说 Control 层，Summit 作为典型的 CS 程序，客户端与服务端通讯采用的不是 TCP/IP 直接通讯的方式，而是采用了 HTTP 协议和 Webservice 的方式。其中，SummitFT 通过 HTTP 协议与通讯中间层通讯；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，并将结果封装成 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)。因此，ENTITY 完全可以用现在的面向对象来理解。Summit 系统这一抽象体系始于 1990 年代，在那个面向对象尚未流行的年代，这种架构设计上的前瞻性令人佩服不已。

二、系统模块

前文提到, Summit 系统前后端通讯协议是 HTTP。首先, HTTP 协议是无状态的, 因此 Summit 系统需要提供一套类似于 Web Server 的会话服务。这个就是 Summit Naming Service 的作用。Summit Naming Service 提供服务的注册和查询功能, Summit 利用此服务来提供会话服务; 其次, Summit 后端功能都是用 C/C++ 开发的, 如何利用后端的 C/C++ 提供 HTTP 服务呢? Summit 采用的是 Web Service 的方式。Summit 后端提供了 etkservice, 这是一个 Web Service 服务端程序, 启动进程后, 会通过一个端口服务某个用户会话。Summit 会为每个客户端连接启动一个 etkservice 进程来提供服务, 利用 Middle Web Service 将 HTTP 请求内容转换成 SOAP(Web Service)请求并转发给相应的 etkservice 进程。etkservice 进程处理完成后, 结果通过 SOAP 报文能加给 Middle Web Service, 进而转换成 HTTP 报文返回给客户端。不仅如此, Middle Web Service 负责所有前、后端交互的 HTTP 报文 <-> SOAP 报文之间的转换工作。

Summit 系统不仅提供了用户操作的基石, 同时提供工作流和数据的生命周期管理的功能。这些功能是由 STP 服务来完成, STP 服务即事件处理服务。STP 服务会订阅系统事件, 每当 Sequencer 发布系统事件时, 订阅了相应事件的 STP 服务即会工作, 完成自动化的处理任务。常

见的 STP 服务如现金流产生服务、支付报文产生服务、额度计算服务、头寸计算服务、合规检查服务等。

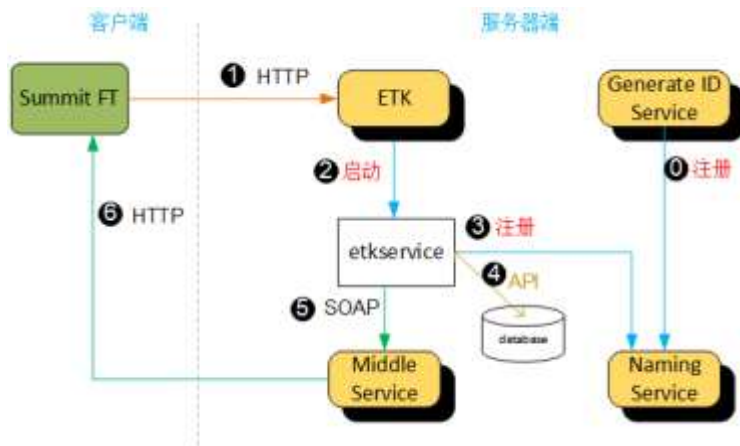
Summit 系统将 Generate ID 独立出来作为一个 Web Service 也是其架构向分布式演进的体现。Generate ID 服务用来生成全局唯一的 ID。Summit 会为其保存或管理的数据分配一个全局唯一的 ID，其内部包括事件生成、分发、存/取，都是以此 ID 作为标识，这些 ID 全部由 Generate ID 来生成。因此，Generate ID 会向 Naming Service 注册自己，并且所有 Generate ID 服务注册名是一样的，以此来保证只有一个实例提供服务。

MQ 作为消息中间件来处理 STP 订阅、BVS 消息发布和 Sequencer 的事件发布。当前，Summit 一般采用 Active MQ 实现。当 STP 服务启动时，会首先在 ActiveMQ 对应的队列订阅相应的事件；Sequencer 服务检索到事件发生时，会向 MQ 发布事件。MQ 将事件广播到对应队列，供相关的 STP 服务处理。BVS 服务启动后，会监听 BVS 请求队列，当 Summit FT 中，用户打开 Blotter view 时，会发送一个 BVS 请求。BVS 服务根据订阅数据筛选条件，将筛选的数据不断推向 BVS 响应队列。

最后，Summit 系统提供了一个用来管理所有的 STP 服务及 Sequencer 服务。STP 由于是监听器的角色，因此其是长时间运行的，这就需要提供一个管理工具，来监控各个 STP 服务的。Service Management Tool(SMT)就是这样一个管理工具。提供了基本的服务启动、停止、新建、删除的功能。

三、模块间的交互

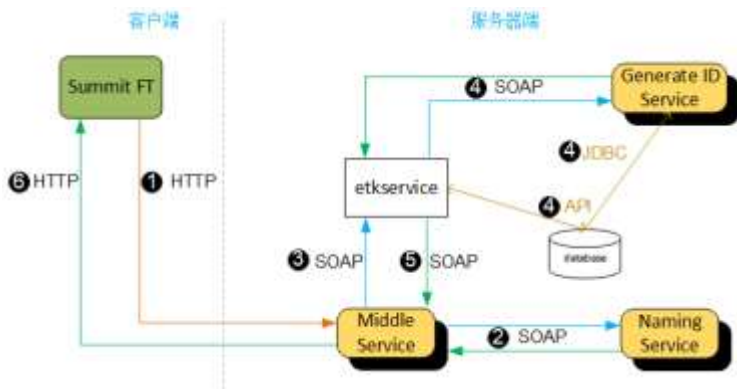
A: 登录过程



0. 系统启动时，Generate ID 需要启动，启动时向 Naming Service 注册自己
1. 用户登陆点击登陆后，Summit FT 向 ETK 服务发起 HTTP 请求
2. ETK 服务启动一个 etkservice 进程来服务此用户
3. etkservice 启动后，向 Naming Service 注册自己(Session ID <-> 端口号)
4. etkservice 读取数据库中的用户信息，验证密码及其他信息(比如是否过期、是否被锁定)
5. etkservice 验证完成后，将结果以 SOAP 报文的形式返回给 Middle Service
6. Middle Service 将 SOAP 报文转换成 HTTP 报文并返回给 Summit FT

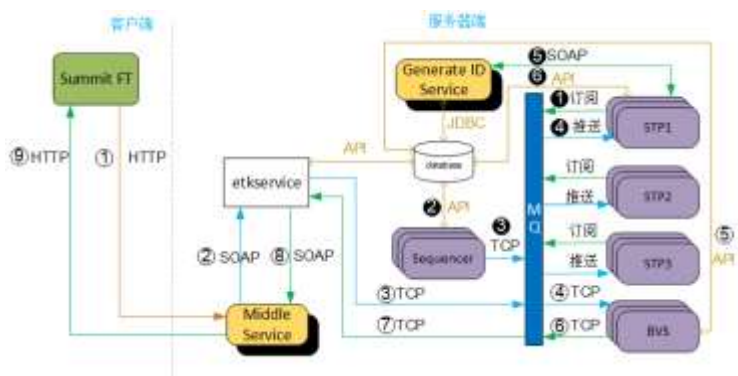
B. 操作过程

操作过程是指用户登陆成功后，在 Summit FT 客户端进行相关的操作，比如点击按钮、订阅实时消息操作。此时的交互方式与登陆有所差别



- 1、 Summit FT 发送 HTTP 请求至 Middle Service(带有 Session 标识)
- 2、 Middle service 向 Naming Service 发起 Web Service 查询此请求对应的 etkservice 端口 (使用 Session ID)，查询成功后，以 SOAP 报文形式返回
- 3、 Middle service 查询到 session 对应的 etkservice 后，向 etkservice 发起 Web Service 请求
- 4、 etkservice 响应请求，与数据库交互，如果需要生成新的 ID，则会向 Generate ID 发起 Web Service 请求生成 ID。Generate ID 生成 ID 后，以 SOAP 形式返回给 etkservice。
- 5、 etkservice 处理完成后，将结果以 SOAP 报文形式返回给 Middle Service
- 6、 Middle Service 将 SOAP 报文转换成 HTTP 报文并返回给 Summit FT

C、事件处理过程



Summit 系统的事件分成 2 个大类，第一类由 Sequencer 服务产生，产生后推向 MQ 队列并由 STP 服务来处理；第二类由用户产生自 Summit FT，比如打开了 Blotter View 来订阅实时数据，此类事件由 BVS 服务处理。因此，笔者将分成 2 个部分来说明 Summit 的事件处理过程。

注：为使交互图清晰，省略了前文提到的部分步骤及相关服务。