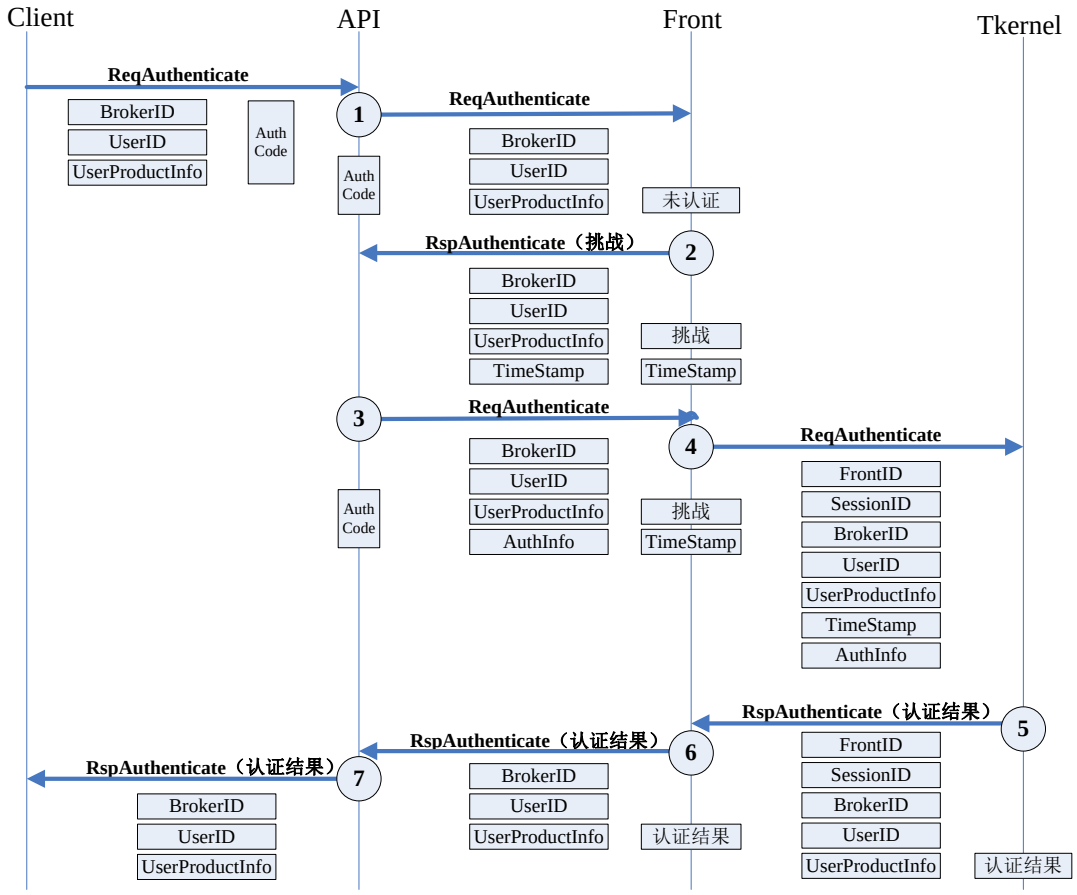
客户端认证

客户端认证的目的是为了保证投资者在交易过程中所使用的客户端是被该投资者所在经纪公司所认可的客户端产品。基本认证过程为：在客户端和综合交易平台（ASP）各自维护与经纪公司和客户端产品相关联的认证码，当客户端请求认证时由 ASP 生成随机字符串发送给客户端，客户端根据自身的认证码对该随机字符串进行处理后生成认证信息发送给 ASP，最终由 ASP 根据自身维护的认证码对随机字符串进行相同处理后的结果与客户端的认证信息进行比对，以判断客户端的合法性。其基本流程可由下图表示：



图表 1 客户端认证流程

步骤描述：

1、 API 在接收到请求客户端认证调用时，将客户端认证信息实体转发给前置，同时将客户端认证码（AuthCode）进行缓存。

2、 前置在接收到客户端认证请求时，需判定其与客户端的会话信息中的认证状态。若为未认证状态，则生成随机字符串作为时间戳填入客户端认证信息实体中反馈给客户端，且认证响应结果为挑战状态，同时更新会话信息中的认证状态为挑战并缓存时间戳；若为挑战状态，则为步骤 4；若为其他状态（认证结果），则为重复的认证请求，将前一次的认证结果返回即可。

3、 API 在接收到客户端认证响应报文时，若认证响应结果为挑战状态，则根据缓存的认证码对时间戳进行处理（进行 DES 加密后转换成字符串），将处理结果填入认证信息实体并清除时间戳后再次发起认证请求，同时将缓存的认证码清除；若为其他状态，则为步骤 7。

4、 前置再次接收到客户端认证请求时，其与客户端的会话信息中认证状态为挑战，则将会话信息中缓存的时间戳信息填入认证信息实体并添加会话引用后发送至交易核心（tkernel）。
5、 交易核心根据自身维护的认证码，采用与 API 相同的处理方式对认证信息实体中的时间戳信息进行处理后与认证信息实体中的认证信息进行比对，若保持一致则认证通过，反之则认证失败，并将认证结果保存在会话信息中。清除认证信息实体中的时间戳和认证信息后将认证结果返回。
6、 前置接收到认证响应报文是直接转发给 API，并将认证结果保存至会话信息中。
7、 API 接收到认证响应报文而且响应结果为认证结果时，将认证结果反馈给客户端程序。

附加说明：

1、 步骤 2 和步骤 4 根据前置与客户端的会话信息中的认证状态信息进行区分。
2、 步骤 3 和步骤 7 根据认证响应结果进行区分。
3、 客户端认证信息实体中的时间戳和认证信息字段对客户端程序保持透明。
4、 当客户端在同一会话中发起重复认证（即前置中已存在认证结果）时，前置直接将前一认证过程的认证结果返回给客户端，而不重复进行完整的认证过程。
5、 交易核心在认证过程中会将认证结果保存至会话信息中，因此需要对用户登录的处理流程进行适当的修改。即在登录前需要判断该客户端是否需要认证以及认证是否通过。
6、 作为过渡阶段，需要同时支持认证及非认证版本客户端的登录，因此交易系统需设置参数（IsAuthForce）指明是否强制认证。
7、 当认证码泄密并进行更新时，要求所有该类型客户端在同一时间点进行更新并不现实，因此允许每类客户端最多可同时存在两个认证码作为过渡阶段。即当认证码泄密时，将该认证码设置成为旧认证码，并重新设置新的认证码，此时两个认证码均可使用。当客户端几乎已全部更新时，则将旧认证码删除以完成认证码的更新过程。
8、 客户端认证请求中的 UserProductInfo 只填写产品信息，不包括版本号。
9、 认证码以加密方式存储在数据库中，交易核心在同步结束时进行解码获取认证码。Dbmt 和 drmt 模块通过调用存储过程获取加密后的认证码，也需进行解密后使用。