Q/ZXZQAPP-001-2023

中信证券股份有限公司企业标准

Q/ZXZQAPP-001-2023

中信证券移动金融客户端应用技术规范

Financial Mobile Application Technical Specification in CITICS

2023-08-25 发布　　2023-08-30 实施

中信证券股份有限公司　　发布

目　次

前　言

本标准依据 GB/T 1.1-2020《标准化工作导则 第一部分：标准化文件的结构和起草规则》给出的规则起草。

请注意本文件的某些内容可能涉及专利。本文件的发布机构不承担识别专利的责任。

本标准由中信证券股份有限公司提出。

本标准由中信证券股份有限公司归口。

本标准起草部门：中信证券股份有限公司财富委数字发展中心。

本标准主要起草人：杨利强、白玉、李鸿。

引　　言

随着移动互联网的蓬勃发展，用户对其的接受度和依赖度越来越高，金融行业自然而然为适应新的需求，将传统金融业务与移动互联网技术相结合。在当前的互联网金融浪潮中，移动金融客户端应用软件相关的前后台系统的建设面临着巨大的压力和前所未有的挑战。一方面需要满足监管部门的要求，确保基于移动互联网的业务开展合法合规，另一方面面临信息安全、隐私安全的挑战，以及海量用户对移动金融客户端软件快速迭代改善的诉求。其中涉及的各技术环节迫切需要规范管理。

为了迎接挑战和落实监管部门的要求，规范移动金融客户端应用软件所涉及的安全、灾备、服务稳定性、数据治理、版本开发、测试、发布等各技术环节的落地实施，进一步提升客户端软件在快速迭代中各版本的安全性和稳定性，切实保护客户和公司的权益。为此，中信证券股份有限公司特制定此标准。

移动金融客户端应用软件技术规范

1 范围

本标准规定了中信证券股份有限公司移动金融客户端应用软件的安全规范、灾备和服务稳定性规范、数据采集和治理规范、性能规范、缺陷处理规范，以及应用软件开发、测试、发布的规范。

本标准适用于面向客户的移动金融客户端应用软件，以及对应支持其功能的服务端程序。

2 规范性引用文件

下列文件对于本文件的应用是必不可少的。凡是注日期的引用文件，仅所注日期的版本适用于本文件。凡是不注日期的引用文件，其最新版本（包括所有的修改单）适用于本文件。

JR/T 0092-2019 移动金融客户端应用软件安全管理规范

JR/T 0175-2019 证券期货业软件测试规范

JR/T 0168-2020 云计算技术金融应用规范 容灾

JR/T 0171-2020 个人金融信息保护技术规范

JR/T 0191-2020 证券期货业软件测试指南 软件安全测试

JR/T 0192-2020 证券期货业移动互联网应用程序安全规范

JR/T 0197-2020 金融数据安全 数据安全分级指南

JR/T 0246-2022 面向老年人的证券期货业移动互联网应用程序设计规范

GB/T 37076-2018 信息安全技术 指纹识别系统技术要求

GB/T 37668-2019 信息技术 互联网内容无障碍可访问性技术要求与测试方法

GB/T 35273-2020 信息安全技术 个人信息安全规范

GB/T 38671-2020 信息安全技术 远程人脸识别系统技术要求

GB/T 39786-2021 信息安全技术 信息系统密码应用基本要求

3 术语和定义

3.1 敏感信息 sensitive information

一旦遭到泄露或修改，会对标识的信息主体造成影响的信息。

注:证券期货行业敏感信息包括客户姓名、客户详细地址、客户联系电话、客户证件号码、客户

开户行及账号、会员交易情况、会员持仓数量、会员可用资金等。

3.2 个人信息 customer information

移动客户端应用软件所处理的，与特定自然人、法人相关，能够单独或通过与其他信息结合识别该特定自然人的计算机数据。

注：计算机数据是与自然人身份属性、财产属性相关的一组数据。

3.3 越狱 jailbreak

解除移动平台操作系统的限制，获取操作系统最高权限的一种技术手段，已越狱的设备上可以安装未经许可的应用软件，其安全性没有保障。

3.4 恢复时间目标 recovery time objective (RTO)

灾难发生后，信息系统从停顿到必须恢复的时间要求。

3.5 恢复点目标 recovery point objective (RPO)

灾难发生后，数据必须恢复到的时间点要求。

3.6 缩略语

APP：客户端应用软件（Application software）

SDK：软件开发工具包（Software Development Kit）

UTF-8：针对 Unicode 的一种可变长度字符编码（8-bit Unicode Transformation Format）

Web API：网络应用程序可编程接口（Web Application Programming Interface）

4 安全要求

4.1 功能设计安全

4.1.1 接口安全

- 引入适当的身份验证机制，例如基于令牌的身份验证方式，确保只有经过授权的用户能够访问接口。
- 为不同的用户角色分配不同的访问权限，以防止未经授权的访问。
- 应使用 HTTPS 协议处理互联网接口请求，以确保数据在传输过程中的安全性和保密性。
- 防止 SQL 注入，对所有从客户端接收的输入进行验证和清洗，防止恶意数据或攻击。
- 在接口中实施合适的错误处理机制，提供安全的错误响应，以避免敏感信息泄露。
- 对于敏感操作，如涉及删除数据或者修改设置的接口调用，需要进行额外的身份验证和授权。
- 对接口进行定期的安全审计，以发现和修复潜在的漏洞。
- 记录接口的调用日志，并保留合理的期限，以便在出现问题时进行有效排查。

4.1.2 抗攻击能力

- 确保只有经过授权的用户才能访问敏感功能和数据，例如证券交易相关功能或查询客户资产情况等。
- 合理使用代码混淆和加密技术，防止黑客轻易地反编译和分析客户端 APP 代码。
- 对用户输入的数据进行充分的验证和过滤，防止恶意输入和操作，尤其是不符合人类行为特征的使用习惯。
- 避免在客户端应用中使用硬编码的敏感数据，如 API 密钥或密码等。
- 加密客户端本地存储的敏感数据，包括用户个人信息和隐私数据，以防止数据泄露。
- 记录客户端的安全事件和异常行为，以便快速响应和排查问题。
- 及时修复已发现的漏洞，支持强制升级，避免客户因信息滞后，暴露在攻击风险面前。
- 与业内专业的安全专家合作，进行安全评估和审查，确保客户端应用满足最佳的安全实践。

4.1.3 组件安全

- 在选择集成进客户端应用的组件时，需要选择来自受信任或者有声望的供应商，确保他们有良好的信誉和可靠的历史记录。
- 仔细查阅供应商提供的文档和示例，充分了解如何正确地集成和使用他们的组件或 SDK。
- 查看组件的安全漏洞历史，以评估供应商是否及时修复和披露漏洞。
- 确保定期更新和升级使用的第三方组件，以获取最新的功能和安全修复。
- 根据组件重要性和对客户、公司的影响，审查供应商提供的源代码，确保其中没有潜在的安全漏洞或恶意代码。

- 仔细查阅供应商的隐私政策，了解他们如何处理用户数据和隐私信息，保持第三方组件最小化权限申请，避免其有机会采集到非必要的隐私信息。
- 准备备选方案，以防止在遇到问题时，需要更换组件，降低潜在的风险。

4.2 运行环境安全

4.2.1 硬件环境检查

- 在客户端应用软件启动时，需要检测操作系统的运行环境，例如是否存在 ROOT 或越狱，是否为虚拟机或模拟器，是否存在动态劫持框架等。
- 软件启动时，还需要进行完整性校验。如果发现本地数据被篡改，则需要废弃该数据，并重新从服务端加载最新的有效安全数据。

4.2.2 环境隔离

- 在客户端应用开发过程中，需要对不同环境进行有效的隔离，包括但不限于开发环境、手动测试环境、自动化测试环境、灰度发布环境和生产环境等。要防止在不同环境中混用用户账户。
- 客户端应用涉及不同密钥和第三方组件的使用，需要保持生产环境和其他环境之间的差异性，避免密钥混用造成生产环境信息泄露。
- 避免使用数据批量导入导出功能，特别是生产环境数据，必须进行数据脱敏处理。

4.2.3 信息展示

- 客户端应用软件中，除必要的信息核对场景外，还需屏蔽敏感信息的关键字段。
- 用户使用软件时，如长时间未操作或离开，需要再次输入校验信息，以保护之前操作输入的数据信息。
- 在敏感信息核对场景下，需要进行操作者身份核实的步骤。
- 在涉及调用第三方服务平台的场景下，需要确保敏感信息不被第三方服务获取。

4.2.4 本地存储

- 客户端应用软件不得以任何形式存储用户的支付敏感信息和金融业务查询口令。
- 在满足法律、管理规定和监管部门要求的前提下，客户端应用软件应仅保存业务必需的个人金融信息，并限制数据存储量。
- 本地存储的客户数据需要验证其完整性，以避免被篡改。

4.3 生物识别安全

- 选择并使用经过安全审计和验证的生物识别库，支持在硬件级别支持的生物识别设备上开启生物识别相关功能，以提供更高的安全性。
- 在可行的情况下，进行本地生物识别，避免将生物特征数据发送到远程服务器。
- 结合生物识别技术与其他身份验证方式，如密码或 PIN 码，以提供更强的安全性。
- 要求客户端软件在指纹生物特征识别的错误拒绝率小于等于 3%的情况下，错误接受率应小于等于 0.002%。
- 要求客户端软件在人脸生物特征识别的错误拒绝率小于等于 5%的情况下，错误接受率应小于等于 0.001%。

4.4 身份认证安全

4.4.1 认证方式

- 客户端应用软件登录时，应根据使用场景采用适宜的验证要素，如口令、短信验证码、手势密码、指纹识别、面容识别等方式。
- 应确保采用的身份验证要素相互独立，即部分要素的损坏或泄露不应导致其他要素损坏或泄露，如用于登录验证的口令和用于交易的口令不能一致。
- 对于手势密码、短信验证码、生物特征信息作为验证要素或验证要素组合中的一种时，应满足如下要求：
    - 若采用手势密码作为验证要素，手势密码应至少设置连续不间断的 4 个点。
    - 若采用短信验证码作为验证要素，短信验证码应仅使用一次，仅限于规定时间内使用，短信验证码应具备长度和随机性的要求。
    - 若采用生物特征识别作为验证要素，应当符合国家、金融行业标准和相关信息安全管理要求，防止非法存储和复制。
- 采用图形验证码作为验证的辅助要素时，图形验证码应具有使用时间限制并仅能使用一次。图形验证码应由服务器生成，客户端源代码中不应包含图形验证码文本内容，图形验证码不得作为独立的身份验证要素。
- 在用户身份认证后，客户端应用软件进入移动终端系统后台时，如果超过设定时限后被唤醒切换到前台，交易相关场景应对用户身份重新认证。

4.4.2 失败处理

- 在身份认证失败时，避免提供过多的信息，以防止攻击者获取有关认证系统的详细信息，只显示有关错误类型的一般信息。
- 要保护账户隐私，不要明确指出是账号错误或密码错误，仅提示“认证失败”，有助于防止攻击者猜测有效的账号信息。
- 实施适度的锁定机制，以避免暴力破解。如果用户多次认证失败，暂时锁定账户，或需要进行额外的身份验证，如验证码等。

- 如果认证失败次数达到一定上限，提供安全的密码重置流程，确保用户可以通过安全通道重置密码。
- 如发现异常认证行为，如频繁失败的认证尝试，实施额外的安全措施，例如暂时封锁相关 IP 地址等。
- 记录认证失败事件，以便进行安全审计和监控。发现异常模式时，及时采取措施。

4.4.3 密码相关

- 客户端应用软件应与服务端配合，提供密码复杂度校验功能，以确保用户设置的密码达到一定的强度，避免采用简单密码或与客户个人信息相似度过高的密码。
- 应严格限制使用初始登录密码或初始交易密码，若设置初始密码，应强制用户在首次登录后修改初始密码。
- 在密码重置时，应使用短信验证码等方式对用户身份进行重新验证。
- 修改密码时，应对预防密码输入错误次数进行限制。
- 修改密码时，新密码不应与原密码相同。

4.5 通信安全

4.5.1 通讯传输

- 在客户端应用软件与服务器之间应该建立安全的信息传输通道，并且协议版本应该及时更新到安全稳定版本。
- 应该确保采用的安全协议不包含已知的公开漏洞。
- 在客户端中验证服务器的证书，以确保连接的目标服务器是可信的，并且防止中间人攻击。
- 在客户端和服务器之间传输敏感数据时，使用符合国家安全要求的密码算法和密码策略，从算法层面保障信息安全。
- 在通信中使用随机的令牌或一次性密码，以防止攻击者利用已捕获的数据进行重放攻击。

4.5.2 数据保密

- 敏感数据（如：登录口令、资金账户信息、持仓信息等）在通过公共网络传输时，应采用安全的通信协议和加密等措施确保其保密性，应使用国家密码主管部门认可的安全加密算法和密钥长度。
- 禁止在客户端应用软件本地缓存敏感数据；若因业务需求，同 Web 前端页面交互传输时，应该用采取白名单机制，仅支持可信赖域名下的网页获取解密后的数据，而敏感数据的获取依赖客户端内部的加密机制。

4.5.3 数据完整

- 使用数字签名来验证数据的完整性和身份认证。服务器生成数字签名，客户端使用服务器的公钥验证签名。
- 在发送数据前，对数据进行哈希处理，生成数据摘要。服务器在接收数据后，同样对数据进行哈希处理并比较数据摘要，以确保数据未被篡改。
- 在数据中加入可计算的数据校验码，以确保数据在传输过程中没有被修改。客户端和服务器可以分别计算验证。
- 确保密钥的安全管理和定期更新，以防止密钥泄漏和被滥用。

4.6 数据安全

4.6.1 安全输入

- 在交易相关场景下，启用自主开发的自定义软键盘进行输入保护。
- 自定义软键盘需要统一编码格式 UTF-8。
- 自定义软键盘不允许输入危险字符，如：<>"$+\`'。
- 用户输入密码等客户敏感信息时，通过不显示明文字符来保护用户隐私信息。
- 在涉及登录、交易、持仓查询等含有敏感信息的场景，默认使用不显示明文字符来保护用户隐私信息。
- 客户端接受的不可信数据源输入，必须根据使用场景进行有效性和可信性校验。
- 禁止直接使用不可信数据来拼接 SQL 语句。
- 禁止直接使用不可信数据来拼接 WebAPI 调用请求。
- 禁止直接使用不可信数据做本地序列化存储。
- 所有输入数据，在服务端需要进行二次校验。

4.6.2 防窃取

- 对存储在本地的敏感数据进行加密，确保即使数据被窃取，攻击者也无法轻易解密获取原始数据。
- 根据客户端软件运行的平台，使用安全的存储机制，如 Android 的 Keystore 或 iOS 的 Keychain，来安全地存储敏感数据，这些存储区域受到操作系统级别的保护。
- 永远不要在本地存储敏感数据的明文形式，如密码、API 密钥等。
- 尽可能不要在本地存储大量敏感数据，使用时，从服务器动态加载数据，减少本地存储的风险。
- 使用操作系统提供的权限机制，将本地数据的访问权限限制为应用自身，避免将数据对其他应用开放。
- 使用代码混淆技术，以及将敏感代码或数据混合在无关的代码中，增加反编译难度。
- 对于敏感信息，可以通过防止截屏的技术，防止攻击者通过屏幕截图获取数据。
- 定期清理本地缓存和数据，确保不需要的敏感数据不会在设备上长时间存在。

4.6.3 数据销毁

- 在不再需要的情况下，立即删除敏感数据，以减少数据存储的时间窗口。
- 使用合适的方法删除数据，确保数据不会被恢复。在移动设备上，可以使用专门的 API 来清除数据。
- 在删除文件或数据库记录时，确保使用适当的方法，彻底擦除数据，以免留下残留数据。
- 不同平台提供了数据销毁的指南和 API，需遵循响应的指南来执行数据销毁操作。
- 当客户端应用被卸载时，确保在清理应用数据时使用合适的方法，以防止敏感数据残留。
- 确保数据销毁过程符合适用的法规和隐私法律，特别是涉及用户隐私和敏感信息的方面。

4.7 风险提示

- 对运行时的网络环境进行安全检测，对可疑或非常用环境下对客户进行风险提示。
- 使用简单、清晰的语言，避免使用专业术语或复杂的语句，以确保用户能够准确理解风险。
- 在提示中突出强调关键的风险要点，以确保用户不会错过重要信息。
- 使用图标、图片、图表等视觉元素来强调和解释风险提示，提升用户的理解和记忆。
- 如果用户在风险提示后，决定不继续操作，要提供明确的退出选项，以免误操作。
- 定期更新风险提示，确保其与最新的金融市场情况和法规保持一致。
- 在风险提示中不仅要强调风险，还可以提供一些用户教育，帮助用户更好地管理风险。
- 确保风险提示符合适用的法规和金融行业的合规要求。

4.8 密钥管理

4.8.1 基本要求

- 在传输过程中，应采用密码算法来保护密钥（对）。
- 随机生成的密钥（对）应具有一定的随机性和不可预测性。
- 密钥（对）应加密存储，并确保密钥（对）存储位置和形式的安全。
- 不同移动金融客户端应使用不相关的密钥（对）。
- 不同用途的业务场景中，应使用不相关的密钥（对）。
- 同一移动金融客户端在不同的运行环境中，应使用不相关的密钥（对）。
- 应为密钥（对）设置有效期，并定期进行更换。

4.8.2 算法选择

- 对称密码算法应使用 SM4、AES 等对称算法。其中，AES 算法的密钥长度应至少为 256 位。优先建议使用 SM4 算法。
- 非对称密码算法应使用 SM2、RSA、ECC 等非对称算法。其中，RSA 算法的密钥长度应至少为 2048 位。优先建议使用 SM2 算法。

- 摘要算法应使用 SM3、SHA 等摘要算法。其中，SHA 应使用 SHA256 及以上。优先建议使用 SM3 算法。

5 隐私保护要求

5.1 系统权限

- 最小权限原则，仅请求移动端应用所需的最低权限，避免过度请求权限，以减少潜在的安全风险。
- 使用动态权限申请机制，仅在需要的时候向用户请求权限。在解释为何需要权限时，提供明确的解释。
- 在请求权限时，向用户提供清晰的解释，说明为何移动端应用需要该权限，以增加用户的信任。
- 适时撤销权限，在不再需要某个权限时，及时撤销该权限，以减少应用的攻击面。
- 对于敏感权限，如访问位置信息、相机、麦克风等，要确保用户了解为何应用需要这些权限。
- 在应用开发和更新过程中，定期审查和审核权限的使用，确保权限的合理性和安全性。
- 允许用户随时在应用设置中撤销某些权限，以增强用户对数据的控制。
- 确保权限请求和使用符合适用的法规和隐私法律要求。

5.2 采集相关

- 最小化收集，仅收集必要的信息，避免过度采集，以减少风险和提升用户信任。
- 在应用中明确披露将收集哪些信息以及其目的，让用户清楚知道数据采集行为的存在。
- 在采集敏感信息前，要求用户明确同意，并确保其了解移动端应用为何需要这些信息。
- 给用户提供选择，允许用户决定是否提供某些信息，尤其是敏感信息。
- 收集的数据应防止被第三方获取。
- 采集的数据需围绕业务指标开展，确保数据链路的完整性，可逐层追溯计算转化率。
- 具有相同含义的数据，要统一口径。
- 所有采集的数据点需具备基础数据和业务特征数据两部分，基础数据包括用户移动终端、渠道、个人属性等最基本的特征。
- 建立数据质量闭环管理机制，将组织、技术和流程三者进行有机结合，对数据进行全生命周期质控。
- 严格满足金融数据合规性以提高金融数据的可用性和正当性。

5.3 使用相关

- 在使用信息之前，明确告知用户将如何使用其信息，以增加透明度和信任。
- 仅在必要的情况下，使用用户的信息，避免滥用或不必要的信息共享。
- 在使用信息时，对数据进行匿名化或脱敏处理，以减少关联风险。
- 确保信息使用符合适用的法规和隐私法律要求，避免不合法的数据用途。

5.4 存储相关

- 使用加密存储敏感信息，确保数据在存储过程中得到适当的保护。
- 定期清理不再需要的信息，确保数据不会长时间存储在设备或服务器上。
- 对存储的数据实施严格的访问控制，确保只有授权人员可以访问。
- 确保存储过程符合适用的法规和行业的合规要求。
- 建立备份和恢复策略，以确保数据在意外情况下的安全性和可用性。

5.5 隐私政策

- 使用简单易懂的语言，避免使用复杂的法律术语，确保用户能够理解隐私政策的内容。
- 在隐私政策中详细列出将收集的信息类型、如何使用、存储、共享和保护这些信息。
- 说明收集信息的具体目的，让用户明确知道为何他们的信息会被收集。
- 说明用户可以如何访问、修改、删除他们的个人信息，以及如何撤回同意。
- 对于敏感信息，如金融账户或密码，详细说明安全措施和保护措施。
- 描述保留用户信息的时间期限，以及保留信息的合理理由。
- 定期更新隐私政策，并在政策变更时向用户发出通知。
- 将隐私政策链接置于应用内显眼位置，让用户容易找到并阅读。
- 在隐私政策中提供关于数据保护的简要教育，帮助用户更好地理解隐私问题。
- 在隐私政策中明确说明法律责任和免责条款。

6 软件管理要求

6.1 功能性

6.1.1 缺陷解决

分类定级

对客户上报、业务定期巡检、测试发现等不同环节提出的软件缺陷进行分级管理，不同级别的缺陷对应不同的修复和响应策略。缺陷问题分为高、较高、中、低四个优先级类别。

- 优先级高
    - 功能完全不可用或崩溃：缺陷导致关键功能完全不可用，应用崩溃或无法启动。
    - 敏感数据损失或泄露：缺陷导致用户的敏感数据遭受风险、泄露或损失。
    - 安全漏洞：缺陷导致应用受到严重的安全漏洞威胁，可能影响用户数据和隐私。
    - 严重的性能问题：缺陷导致应用性能问题明显，用户无法正常使用。
    - 核心功能异常：缺陷导致应用的核心功能无法正常工作，严重影响用户体验。
- 优先级较高

  - 功能部分不可用：缺陷导致某些功能在特定情况下不可用，但应用仍然可使用。
  - 非关键数据缺失：缺陷导致一些非关键数据缺失，但不涉及敏感信息。
  - 中等性能问题： 缺陷导致应用性能有问题，但不会严重妨碍用户使用。
  - 用户界面问题： 缺陷导致用户界面显示有问题，但不影响基本操作。
- 优先级中
  - 界面问题： 缺陷导致界面显示异常，但不会影响功能操作。
  - 少数用户受影响： 缺陷只影响少数用户，大多数用户不会受到影响。
  - 文案错误： 缺陷导致应用中的一些文案、标签错误，但不会影响功能操作。
  - 一般性问题： 缺陷不会对用户的正常使用造成显著影响，一般是小问题。
- 优先级低
  - 轻微的界面问题： 缺陷导致界面上的一些小问题，但不影响功能操作。
  - 图标、颜色问题： 缺陷影响图标、颜色等视觉细节，但不影响功能。
  - 用户体验细微问题： 缺陷导致用户体验上的细微问题，但不影响基本操作。
  - 其他小问题： 缺陷是一些细小问题，不会对用户产生明显影响。

修复标准

- 优先级高的缺陷：导致客户端应用软件核心功能无法正常工作的缺陷，需立即修复。修复时长应小于 4 小时，并做相应的复盘记录，定期回顾以降低未来发生的几率。
- 优先级较高的缺陷：对应用软件功能影响较大的缺陷，如结果错误、非核心功能失效、数据丢失等，需尽快修复。修复时长应小于 24 小时。
- 优先级中或低的缺陷：对应用软件功能影响一般，引起体验或需求理解歧义的缺陷，应按照优先级在版本发布前修复。

6.1.2 兼容性

- 参考中国大陆市场上月度占有率 TOP500 的 iOS、Android 系统机型，移动客户端应用软件需要通过兼容性测试标准，即在该机型上（搭配其默认或已更新到的主流操作系统版本均可），客户端的核心功能可以正常运行。
- 参考中国大陆市场上月度占有率 TOP20 的 iOS、Android 操作系统版本，移动客户端应用软件需要通过兼容性测试标准，即在该操作系统版本上，客户端的核心功能可以正常运行。信 E 投 iOS 客户端目前要求最低支持 iOS 9.0 版本，Android 客户端目前要求最低支持 Android 4.2 版本。
- 对即将发布的 iOS 和 Android 平台操作系统版本，需要提前根据官方文档和测试版系统制定适配计划，应在新版本操作系统正式推送时完成软件版本适配。
- 需要支持鸿蒙系统的更新，制定适配计划。
- 客户端应与其他券商移动应用在同一终端设备或系统中共存，不应设置任何阻碍共同运行的功能或检测逻辑。

6.1.3 信创实施

- 客户端软件的所有功能需要支持信创环境接入。

- 信创链路需保持独立，并且与当前行情等组件完全隔离。
- 信创环境的运维和维护标准与当前系统相同。

6.1.4 网络协议

- 客户端应用需要同时支持 IPv4 和 IPv6 网络。
- 在开发和测试阶段，需要对 IPv6 环境进行充分测试。
- 确保使用的网络库、框架或协议栈是支持 IPv6 的最新版本。
- 确保应用能够根据网络类型（IPv4 或 IPv6）选择合适的协议与地址。
- 确保应用使用的 TLS/SSL 库支持 IPv6，以保持安全连接在 IPv6 网络下正常工作。
- 确保集成的第三方组件也支持 IPv6。

6.1.5 会话管理

- 应采取会话保护措施，防止软件与后台服务器之间的会话被窃听、篡改、伪造、重放等攻击。
- 登录完成后的会话管理阶段的所有请求都需要对用户的合法身份进行鉴别，鉴别通过后才能进行操作。
- 应确保用户在执行注销/登出操作后，会话被安全终止。
- 会话结束后应立即清除敏感数据缓存，防止信息泄露。
- 应设计合理的账户登录超时控制策略，当用户闲置在线状态超过时限时，自动退出登录状态。
- 应限制会话并发连接数，避免恶意用户创建多个并发的会话来消耗系统资源，影响业务的可用性。

6.1.6 软件共存

- 客户端软件与常见的其他应用能够良好共存，避免因为应用冲突或公共系统资源争抢而影响用户体验。
- 确保在处理剪贴板数据时，保护用户敏感信息的安全性。
- 确保应用在用户界面和操作逻辑方面，符合所在平台推荐的用户操作指南或行为习惯。
- 确保与其他应用共存时，遵循平台规定的后台任务限制，避免过度使用系统资源。

6.1.7 适老相关

- 在客户端软件的需求设计和开发过程中，应满足 JR/T 0246-2022《面向老年人的证券期货业移动互联网应用程序设计规范》中的所有基本设计项目。在产品设计和业务场景设计方面，应满足老年人的正常使用需求。
- 在客户端软件的需求设计和开发过程中，尤其是用户体验和交互操作方面，应该满足 GB/T 37668-2019《信息技术互联网内容无障碍可访问性技术要求与测试方法》中与客户端应用相关的一级要求。

6.2 非功能性

6.2.1 运维管理

- 需要部署监控系统，实时监测应用的性能、稳定性和安全性。
- 设置报警机制，及时通知运维团队有关客户端应用的问题，以便快速响应和解决。
- 实施容灾方案，确保在服务器或网络故障时，能够快速切换到备份系统，保障业务连续性。
- 定期测试容灾系统，确保备份系统正常运行。
- 定期进行安全漏洞扫描和代码审查，修复可能的安全问题。
- 更新和维护应用所使用的安全库和框架，确保安全性。

6.2.2 稳定性

服务器关键指标

- 客户端依赖的服务器数据平台的容灾等级要达到 3 级。
- RTO 小于等于 5 分钟。
- RPO 小于等于 30 秒。
- 每个自然年非计划服务中断时间不超过 4 天，系统可用性至少达到 99%。

客户端指标

- 在开发、测试阶段完成对闪退、卡顿、内存泄漏的开发自查和测试。
- iOS 和 Android 端版本发布后的线上日均设备崩溃率不能超过万分之八。

6.2.3 性能标准

- 严格控制双平台客户端安装包的大小，每次发布前需要去除冗余资源文件和已废弃的代码文件等。
- 合理使用 CPU 和内存，优化耗电比率，保持应用软件和设备的整体运行流畅。
- 在移动网络稳定的情况下，客户端应用软件应选择最优最近的服务器发起网络请求。
- 控制移动网络流量的使用，不过度消耗客户的流量。
- 客户在使用交易或行情相关的功能时，每个操作的 UI 响应要连贯，保持给客户提供流畅的体验感。
- APP 后台服务器的接口并发响应容量需要大于等于 1000 QPS，且单次请求接口的访问时间（在客户自身网络正常的情况下）做到小于等于 1 秒。

6.2.4 版本更新

基本要求

- 客户端应用软件需要遵循规范的上线发布流程，在提交前进行最后的配置和签名校验。
- 软件应删除调试或测试中存留的敏感数据。
- 软件需要支持灰度更新的功能，以保障版本更新发布的安全和平稳过渡。
- 因监管、合规等原因引发的强制更新，必须提前以明显的方式通知客户。
- 在新版本灰度发布开始的一周内，需要及时使用线上环境进行回归测试，并关注各用户渠道反馈的问题，做到 24 小时内及时回复或解决，以降低未知风险对业务产生负面影响的程度。

配置审查

- 在 iOS 客户端提交应用商店之前，需要进行以下配置审查：
    - 确认使用生产环境配套的服务器地址和密钥。
    - 确认没有使用私有 API 或者苹果应用商店违禁的类和函数。
    - 确认未缺失需要支持的硬件架构。
    - 确认未支持非已知的 URL Schemes 功能跳转。
    - 确认使用适当的证书和签名。
    - 确认所有依赖的运行库编译选项正确。
    - 确认未申请非必需的权限。
- 在 Android 客户端提交各渠道应用商店之前，需要进行以下配置审查：
    - 核查 Manifest 文件设置，其中设置 allowBackup 为 false，设置 debuggable 为 false，不需要导出的组件设置 exported 为 false。
    - 确认未申请非必需的权限。
    - 确认 ProGuard 配置如预期。
    - 确认安装包经过加固。

日志审查

- 禁止在日志中输出敏感信息。
- iOS 客户端发布版本禁止使用 NSLog 或其他第三方日志库将应用日志输出到控制台。
- Android 客户端在发布版本中需要使用 ProGuard 来删除所有 Log 类方法的调用。

6.2.5 消息推送

- 非交易类功能应根据需求支持推送唤醒到达。
- 全量推送的设备到达率应不低于行业平均水平。
- 应支持指定客户的定向推送，可以按照自然人维度进行筛选。

- 推送消息内容必须合规。
- 运营活动推送消息在最终推送发布环节，必须经过人工确认才可以下发。

6.2.6 应急处理

- 客户端应用需要具备应急版本储备，以备特殊情况使用。这样，在特定情况下，用户可以快速回退到应急版本，而不影响主要功能的使用，尤其是交易相关功能。
- 在紧急情况下，可以采取临时修复措施，以减轻问题的影响。
- Web 相关功能页面需要支持快速升级更新，视紧急问题的严重程度和紧迫程度而定，并在此基础上决定升级时间。
- 客户端应用需要支持在不同站点之间切换，以便根据实时网络情况或站点负载情况进行调整。

7 创新要求

7.1 全息画像

- 确保在收集和处理用户数据时，严格遵循数据隐私法规和合规性要求，保护用户的隐私权益。
- 在用户授权的情况下，需使用用户的使用习惯数据等，为用户建立客观清晰的用户画像。
- 需充分利用机器学习、深度学习或数据分析等技术，为用户提供个性化的推荐信息或服务。
- 需基于用户行为的实时数据，调整界面显示、推荐内容等，以提供更好的用户体验。

7.2 智能投顾

- 确保智能投顾的建议和操作符合金融行业的法律法规和监管要求，包括投资建议的透明度和合规性。
- 基于用户的风险偏好、投资目标等信息，提供个性化的投资建议和资产配置。
- 充分利用大数据分析、机器学习、深度学习等技术，预测市场区域和资产表现，以支持智能投顾的决策。
- 支持用户根据投顾建议进行投资交易，确保交易执行的及时性和准确性。
- 确保用户的个人和投资数据受到充分的保护，防止数据泄露和滥用。

8 研发流程要求

8.1 需求设计

- 确保业务需求有清晰的理解，包括用户期望、功能要求、业务流程等。
- 了解目标用户的需求、偏好和习惯，以便设计出更贴合用户的功能。
- 对需求进行功能分析，确定软件需要实现的核心功能和附加功能。
- 根据用户需求，编写用户故事和用例，详细描述用户如何与软件交互，从而确保对功能的准确理解。
- 根据业务价值和用户需求，对功能进行优先级排序，确保在开发过程中能够先实现最重要的功能。
- 设计用户界面，确保界面清晰、易用，并符合用户体验设计的最佳实践。
- 确保需求设计符合金融行业的合规性和法规要求，包括数据隐私、安全性等方面的规定。

8.2 自主开发

8.2.1 开发规范

- 客户端应用软件的功能设计需要遵循安全、可靠、易用、可维护、可扩展、可测试的原则，需要有相关需求文档、功能设计文档的存档记录。
- 客户端应遵循合法、正当、必要的原则，不收集与所提供服务无关的个人金融信息，不收集敏感数据，并在遵循易用性的原则基础上，采用人工智能技术为客户提供合适的服务体验。
- 软件开发之前，需要做充足的预研和演示 Demo，对产品需求和安全性有深入的理解。
- 软件开发需要严格遵守项目管理流程，编码规范，功能有合理的优先级划分，有预计完成时间，可进行完整的测试。
- 接口或函数有严格的输入输出检查，并且各参数均需要配有清晰的注释描述，函数内复杂的业务逻辑也需要有足够的注释解释设计思路。
- 代码提交前，需要经过代码审查，负责审查的工程师需要做到可替代维护已审核的代码。
- 功能提测前，需要对编译器警告、静态分析结果进行自查，尤其对静态分析中暴露出的问题，不符合预期的要提起警惕，及时解决。
- 功能提测前，需要经过在至少一款主流机型和操作系统版本上的自测，复杂业务逻辑需要有配套的单元测试和场景测试用例支持。

8.2.2 开发工具

- 客户端应用软件开发中所用到的 IDE 开发环境（如：Xcode、Android Studio、Hbuilder、WebStorm、VisualStudio Code 等），均需来自官方可信赖的下载源，并对下载后的文件进行 MD5 校验。

- 避免使用非官方的编译器，或者其他未经认证的开发辅助工具、插件，若有特殊需求，需申请公司内部的安全审核团队复核确认其安全性，再引入项目组。

8.3 厂商开发

- 开发规范和开发工具要求与自主开发相同。
- 要求厂商工程师遵守保密协议，确保用户数据和知识产权的保密。
- 开发前，需要与厂商工程师充分沟通需求细节和难点，确保其遵循项目计划，按时交付任务，保证项目质量。
- 对厂商交付的代码进行充分的代码审查，按照公司自有的最佳实践逐条审核。
- 要求厂商工程师能够撰写清晰的文档，包括功能设计文档、接口文档和进度报告。

8.4 代码管理

- 所有客户端应用软件的源代码文件及相应的开发设计文档应及时加入到指定的源代码管理服务器中的指定源代码管理仓库中。
- 应用软件运行所必需的第三方软件、控件和其它支持库等文件也必须及时加入源代码管理服务器中。
- 源代码服务器对于共享的源码管理仓库的访问建立操作系统级的，基于身份和口令的访问授权。
- 在源码管理仓库中设置用户，并为不同用户分配不同的，适合工作的最小访问权限。
- 要求连接源码管理仓库时必须校验源码管理仓库中用户身份及其口令。在源码管理仓库中要求区别对待不同用户的可访问权、可创建权、可编辑权、可删除权、可销毁权。严格控制用户的读写权限，应以最低权限为原则分配权限；开发人员不再需要对相关信息系统源代码做更新时，须及时删除账号。
- 工作任务变化后要实时回收用户的相关权限，对源码管理仓库的管理要求建立专人管理制度专人专管。每个普通用户切实保证自己的用户身份和口令不泄露。
- 任何源代码文件包括设计文档等技术资料不得利用如微信、QQ 等涉外网络环境形式进行传输。

8.5 组件集成

- 应建立第三方组件 SDK 接入管理机制和工作流程，必要时应进行安全评估。
- 应向用户明确标识哪些服务由第三方提供。
- 应监督第三方组件 SDK 提供者加强个人信息安全管理，发现第三方产品或服务没有落实安全管理要求和责任的，应及时督促整改，必要时停止接入。

8.6 质控相关

- 功能测试和非功能测试的测试范围合理、覆盖全面，交易相关功能的覆盖率达到 100%。
- 应根据工作量预估，选择合理有效的测试策略。

- 应明确功能和非功能的测试范围、测试执行的准入/准出要求。
- 测试计划、范围、策略选择、方法选择、环境配置等均需留有记录存档。
- 明确记录每条测试用例的执行结果。
- 客户端软件发布前，相关的客户端和服务端均需要经过安全功能检查、代码安全测试、漏洞扫描、渗透测试，确定符合证券行业软件安全测试标准。
- 客户端应用软件发布前，需要对如下方面进行基本测试：身份认证、口令、访问权限、会话管理、通信、业务逻辑、输入数据、存储数据、提示信息、日志数据、算法、配置、拒绝服务、源代码检测、运行环境、安装卸载、已暴露的组件、权限、第三方库、安装包，所有结果应符合产品需求预期表现。

8.7 自动化测试

- 自动化测试应涵盖客户端软件的各个功能模块和业务场景，尤其是委托交易等关键核心功能。
- 确保自动化测试能够在不同设备、手机操作系统版本上运行，以验证软件在各种环境下可正常使用，功能表现符合产品需求预期。
- 应进行性能测试，验证客户端软件单位时间内对服务器请求压力，较上个版本的差异在服务端负载可接受范围内；验证 10 分钟高强度使用后，移动终端的温度表现保持在合理（Fair）状态以内。
- 应进行数据一致性测试，验证客户端软件发送数据与服务器接收数据是否保持一致。
- 应验证客户端软件在各种异常情况下的表现，如网络中断、服务器故障等，确定其符合产品需求预期。
- 应编写易于维护和扩展的自动化测试脚本，以便在软件更新和新功能添加时进行快速适应。
- 应设置自动化测试的定期执行计划，以及时捕获潜在的问题和缺陷。
- 应自动生成详细的测试报告，记录测试结果和问题，供开发团队修复和改进。

参考文献

[1] JR/T 0092-2019 《移动金融客户端应用软件安全管理规范》

[2] JR/T 0175-2019 《证券期货业软件测试规范》

[3] JR/T 0168-2020 《云计算技术金融应用规范 容灾》

[4] JR/T 0171-2020 《个人金融信息保护技术规范》

[5] JR/T 0191-2020 《证券期货业软件测试指南 软件安全测试》

[6] JR/T 0192-2020 《证券期货业移动互联网应用程序安全规范》

[7] JR/T 0197-2020 《金融数据安全 数据安全分级指南》

[8] JR/T 0246-2022 面向老年人的证券期货业移动互联网应用程序设计规范

[9] GB/T 37076-2018《信息安全技术 指纹识别系统技术要求》

[10] GB/T 37668-2019《信息技术 互联网内容无障碍可访问性技术要求与测试方法》

[11] GB/T 35273-2020《信息安全技术 个人信息安全规范》

[12] GB/T 38671-2020《信息安全技术 远程人脸识别系统技术要求》

[13] GB/T 39786-2021《信息安全技术 信息系统密码应用基本要求》