

T/ACCEM

团 体 标 准

T/ACCEM XXXX—2025

智能远程控连软件通用技术要求

General technical requirements for intelligent remote control and connection software

（征求意见稿）

2025 – XX – XX 发布

2025 – XX – XX 实施

中国商业企业管理协会 发 布

前 言

本文件按照GB/T 1.1—2020《标准化工作导则 第1部分：标准化文件的结构和起草规则》的规定起草。

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。

本文件由成都掠食鸟科技有限公司提出。

本文件由中国商业企业管理协会归口。

本文件起草单位：成都掠食鸟科技有限公司、XXX、XXX。

本文件主要起草人：XXX、XXX、XXX。

智能远程控连软件通用技术要求

1 范围

本文件规定智能远程控连软件在设计、开发、测试、验收及维护全生命周期的基本要求，总体架构、系统功能、软件运维与保障的要求。

本文件适用于智能远程控连软件从设计到维护的全生命周期管理活动。

2 规范性引用文件

下列文件中的内容通过文中的规范性引用而构成本文件必不可少的条款。其中，注日期的引用文件，仅该日期对应的版本适用于本文件；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。

GB/T 20273 信息安全技术 数据库管理系统安全技术要求

GB/T 25000.51 系统与软件工程 系统与软件质量要求和评价（SQuaRE） 第51部分：就绪可用软件产品（RUSP）的质量要求和测试细则

GB/T 30998 信息技术 软件安全保障规范

GB/T 37973-2019 信息安全技术 大数据安全管理指南

GB/T 39786-2021 信息安全技术 信息系统密码应用基本要求

YD/T 3956 电信网和互联网数据安全评估规范

3 术语和定义

下列术语和定义适用于本文件。

3.1

远程控连 remote control and connection software

通过网络通信技术，实现移动设备（如手机、平板）或其他控制终端对远程电脑设备进行连接并操控的过程，包括但不限于对电脑的开关机控制、桌面操作、文件管理等操作。

4 基本要求

4.1 易用性

4.1.1 界面设计

软件界面布局应简洁明了，各功能模块的入口清晰易找。操作按钮设计符合人体工程学和用户操作习惯，尺寸适中，便于点击。

4.1.2 操作流程

用户执行各项功能的操作步骤应尽量简化。对于初次使用的用户，软件应弹出新手引导界面，介绍关键功能的使用方法。

4.1.3 帮助信息

4.1.3.1 软件内配备完善的帮助文档和在线客服入口。帮助文档分类详细，涵盖软件所有功能的使用说明、常见问题解答等内容。

4.1.3.2 在线客服应能够及时响应用户咨询，提供有效的技术支持。

4.2 可靠性

4.2.1 稳定性

4.2.1.1 软件应具备高稳定性，在正常网络环境和常规电脑负载条件下，不出现死机、崩溃、闪退等异常情况。

4.2.1.2 在网络波动或电脑系统资源紧张（如内存占用较高）时，软件应能够自动调整运行策略，维持基本功能正常运行，不丢失已执行操作的数据。

4.2.2 容错性

4.2.2.1 当用户进行错误操作时，软件应给出明确的错误提示信息，并引导用户进行正确操作。

4.2.2.2 在数据传输过程中出现中断等异常情况，软件应具备自动重试机制，确保数据传输的完整性。

4.2.3 恢复性

若软件因意外情况（如系统故障、断电）导致运行中断，在恢复正常运行条件后，软件应能快速恢复到中断前的状态，继续未完成的操作或任务。

5 总体架构

5.1 架构设计原则

5.1.1 模块化原则

将软件系统划分为多个独立的功能模块，如远程控制模块、文件管理模块、安全模块等。模块之间通过清晰定义的接口进行通信和交互，降低模块间的耦合度，便于模块的独立开发、测试、维护和升级。

5.1.2 分层设计原则

采用分层架构，宜分为交互层、业务逻辑层和数据访问层。

——交互层负责与用户进行交互，展示软件界面和接收用户输入；

——业务逻辑层实现软件的核心业务功能，如远程控制指令的处理、文件传输逻辑等；

——数据访问层负责与数据存储进行交互，实现数据的读取、写入和管理。

5.1.3 灵活性原则

架构设计应能适应不同的运行环境和用户需求变化。

5.2 系统组成

5.2.1 客户端模块

5.2.1.1 安装在控制终端（如手机、平板、电脑）上，负责与用户进行交互，接收用户输入的远程控制指令，并将指令发送给服务端。同时，接收服务端返回的远程电脑桌面画面、文件数据、硬件信息等内容，并展示给用户。

5.2.1.2 客户端模块负责本地数据的缓存和管理，如用户登录信息、常用设备列表等。

5.2.2 服务端模块

5.2.2.1 部署在服务器上，作为客户端与远程电脑之间的桥梁。主要负责接收客户端发送的指令，对指令进行解析和验证，然后将指令转发给对应的远程电脑。

5.2.2.2 接收远程电脑返回的响应数据，如桌面画面数据、文件传输数据、硬件信息数据等，并将这些数据转发给客户端。

5.2.2.3 服务端模块还负责用户认证、设备管理、连接管理等核心业务逻辑的处理，确保系统的安全稳定运行。

5.2.3 通信模块

5.2.3.1 负责客户端与服务端之间、服务端与远程电脑之间的数据传输。

5.2.3.2 数据传输过程中，应对数据进行加密处理，防止数据被窃取和篡改。

5.2.3.3 通信模块应具备网络状态监测功能，能够实时监测网络连接状态，当网络出现异常时及时通知相关模块进行处理。

5.2.4 安全模块

负责保障软件的安全性，同时安全模块还具备安全漏洞检测和修复功能，定期对软件系统进行安全扫描，及时发现并修复潜在的安全漏洞。

- 用户认证时应提供多种安全的认证方式（如账号密码、指纹识别、面部识别等方式），对用户身份进行严格验证。
- 系统密码应符合 GB/T 39786—2021 第 5 章、第 10 章的要求，采用强化密码策略，应使用复杂的密码，并定期更改以增强安全性。
- 数据传输过程应支持数据加密技术，确保数据在传输和存储过程中不被窃取或篡改，数据安全应符合 GB/T 37973—2019 第 8 章的规定。
- 数据库应使用安全的加密算法和协议，数据库安全应符合 GB/T 20273 的规定。
- 访问控制时应根据用户角色和权限，对用户的操作进行授权管理，防止未授权访问和越权操作。

5.3 技术架构

5.3.1 架构类型

宜采用C/S（客户端/服务器）架构，客户端负责提供用户操作界面和本地数据处理，服务端负责集中处理业务逻辑和数据存储管理。

5.3.2 客户端

移动客户端采用原生开发技术。

- Android 客户端，宜使用 Java 或 Kotlin 语言进行开发，利用 Android 系统提供的开发框架和 API，实现与手机硬件的交互，以及良好的界面展示效果；
- iOS 客户端，宜使用 Swift 或 Objective-C 语言进行开发，应遵循苹果公司的人机交互设计规范，打造流畅、美观的用户界面。

5.3.3 桌面端

客户端根据不同操作系统特点选择合适的开发技术。

- Windows 系统上，宜使用 C++结合 MFC 或 Qt 框架进行开发，实现高效的桌面应用程序；
- MacOS 系统上，宜采用 Swift 或 Objective-C 语言结合 Cocoa 框架进行开发，确保与 MacOS 系统风格一致；
- Linux 系统上，宜使用 GTK 或 Qt 等跨平台框架进行开发，以适应不同 Linux 发行版的需求。

5.3.4 服务端

5.3.4.1 服务端应采用基于 Java Spring Boot 框架进行开发。

5.3.4.2 数据库宜选用 MySQL、MongoDB 等关系型或非关系型数据库。MySQL 适用于结构化数据存储和复杂查询场景，MongoDB 则在处理非结构化数据和高并发读写场景下具有优势。

5.3.5 通信技术

客户端与服务端之间的通信宜采用HTTP/HTTPS协议。在数据传输格式上，采用JSON格式。远程桌面控制等模块，宜采用WebSocket协议进行通信，实现全双工通信，降低通信延迟，提高实时交互性能。

6 系统功能

6.1 远程控制功能

6.1.1 远程开关机

6.1.1.1 响应时间

从客户端发送远程开机、关机或重启指令到服务端接收到指令并转发给远程电脑，再到远程电脑开始执行相应操作，整个过程的响应时间 ≤ 1 s。

6.1.1.2 兼容性

软件应支持市面上主流的电脑硬件品牌和型号，包括但不限于联想、戴尔、惠普、华为等品牌的台式机、笔记本电脑。同时，应兼容各种常见的电脑操作系统，如Windows系统、MacOS系统、以及Linux系统。在进行远程开关机操作时，不会因电脑硬件或操作系统的差异而出现兼容性问题。

6.1.2 远程桌面控制

6.1.2.1 建立远程连接所需时间 ≤ 3 s。

6.1.2.2 控制端看到被控端操作画面延迟 ≤ 7 ms。

6.2 文件管理功能

6.2.1.1 在网络带宽充足且网络稳定的情况下，文件传输速度应不低于 10 MB/s。

6.2.1.2 在网络条件较差时，软件应能自动调整传输策略，保证文件传输的稳定性，尽量避免传输中断。

6.2.1.3 在文件传输过程中，应确保数据的完整性和准确性。软件应具备传输错误检测和自动修复机制，当检测到传输错误时，能够及时重新传输错误部分的数据，确保文件传输的可靠性。

6.3 硬件信息查看功能

软件应能够远程获取并展示远程电脑硬件的详细信息，至少包括

- CPU 型号、核心数、主频；
- 内存容量、类型、频率；
- 硬盘型号、容量、接口类型；
- 显卡型号、显存容量；
- 主板型号、芯片组；网卡型号、MAC 地址等信息。

6.4 监控与传输功能

6.4.1 运行状态监控

6.4.1.1 软件应实时监控远程操作的运行状态，如远程开关机操作的执行进度、文件传输的进度。

6.4.1.2 远程桌面控制过程中的网络连接状态（连接正常、网络延迟高、连接中断等），也应实时展示给用户，方便用户及时了解操作进展和网络状况。

6.4.2 传输列表

6.4.2.1 应提供详细的文件传输列表，记录所有正在进行、已完成和已失败的文件传输任务。

6.4.2.2 传输列表应包括文件名称、文件大小、传输方向（上传或下载）、传输状态、开始时间、结束时间（若已完成）等信息。用户可在传输列表中对正在进行的传输任务进行暂停、继续、取消等操作，对已完成或已失败的任务进行查看详细信息或重新传输等操作。

7 软件测试与验收

7.1 测试准备

测试前应准备详细的测试计划，明确测试目标、测试范围、测试环境、测试资源（包括测试人员、测试设备、测试工具等）、测试进度安排等内容。同时，需准备好测试用例，测试用例应覆盖软件的所有功能点、性能指标和安全要求，具有可操作性和可衡量性。

7.2 测试实施

7.2.1 单元测试

对软件的各个功能模块进行单元测试，验证每个模块的功能是否符合设计要求，代码是否存在逻辑错误、语法错误等问题。单元测试由开发人员完成，采用白盒测试方法，结合代码审查进行。

7.2.2 集成测试

将各个已通过单元测试的模块进行集成，测试模块之间的接口是否匹配、数据传递是否正确、协同工作是否正常。集成测试采用灰盒测试方法，重点测试模块间的交互和协作情况，确保集成后的系统功能完整、稳定。

7.2.3 系统测试

在模拟真实使用环境的测试环境中，对软件进行全面的系统测试，包括功能测试、性能测试、安全测试等。

7.3 验收标准

应依据GB/T 25000.51-2016进行系统的验收。

8 软件运维与保障

8.1 运维计划

应依据GB/T 36626-2018第8章的要求，制定系统运维计划，内容包括但不限于备份恢复、故障处理、安全管理等方面的内容。

8.2 运维团队与职责

应建立专业的运维团队，团队成员包括运维工程师、技术支持人员等，负责软件的日常运行维护、系统监控、故障处理、版本升级等工作。

8.3 安全管理与漏洞修复

8.3.1 应定期进行漏洞扫描和修复。采取必要的安全措施，防止恶意攻击和数据泄露。

8.3.2 应依据 YD/T 3956 定期对数据安全进行评估。
