

TECHNICAL PROPOSAL

釘釘 → 微信 文件自動轉發系統

技術實現方案

受眾：IT 工程師 / 系統架構師

版本 V1.0

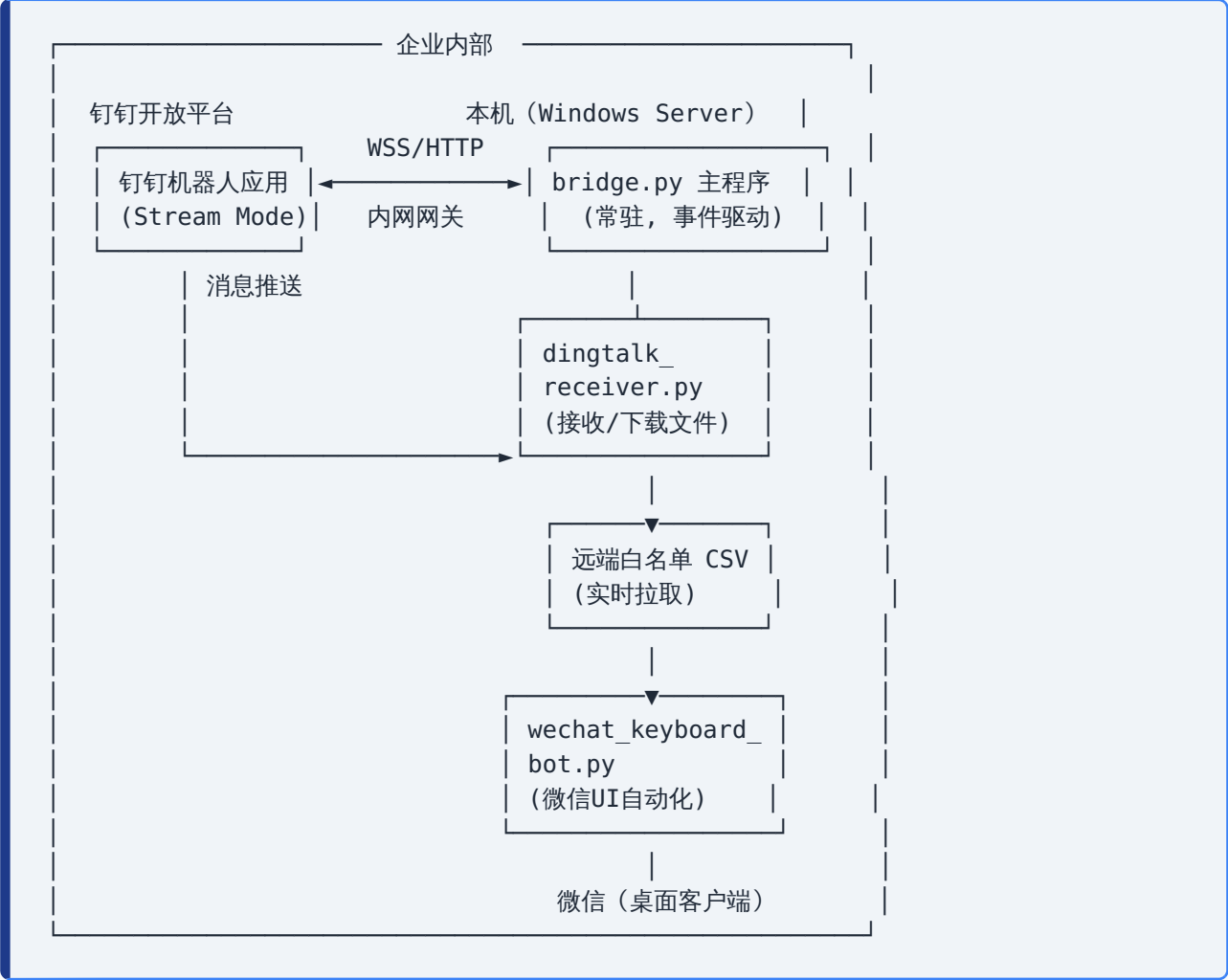
说明：本文档为 PPT 制作的前置脚本。每个 ## 章节对应一页幻灯片，含「页面标题 / 要点 / 讲稿备注」，可直接转为 PPT 大纲。

P0 封面

钉钉 → 微信 文件自动转发系统 · 技术实现方案

- 实时事件驱动 + 桌面 UI 自动化 + 白名单双层审核
 - 端到端：钉钉 Stream 长连接 → 本地落盘 → 微信窗口自动化发送 → 审计入库
-

P1 系统架构与组件



技术栈 : Python 3.11 / dingtalk-stream SDK 0.24.3 / uiautomation / pyautogui / pyperclip / pywin32 / requests / openpyxl

P2 端到端数据流（详细）

1. 接收：员工在企业钉钉给机器人发文件 → 钉钉 Stream 长连接（经内网网关 `dingplus.eavarytech.com`）把消息实时推给 `bridge.py`
 2. 解析：`dingtalk_receiver.py` 从消息提取 `senderNick`、`senderStaffId` (uuid)、`sessionWebhook`、`downloadCode`
 3. 下载：用 `AppKey/AppSecret` 换 `access_token` → 调钉钉 API 下载文件到本地 `downloads/`
 4. 审核：`resolve_target()` 实时拉取远端白名单 CSV，做姓名 + uuid 双匹配（见 P3）
 5. 命中 → 确定微信目标联系人；未命中 → 钉钉回复“不在白名单”，流程终止
 6. 重命名：文件改为 `<原stem>_<发送人>_<YYYYMMDDHHMM>.<ext>`，便于溯源
 7. 发送：独立线程 `WeChatKeyboardBot.send_file()` 在微信中搜索联系人并发送（见 P4）
 8. 审计：发送成功 → `append_record_to_excel()` 写入 `logs/bridge_records.xlsx`
 9. 反馈：全程用 `sessionWebhook` 回复钉钉状态（收到文件 → 安全扫描 → 发送中）
 10. 日志：所有步骤写入 `logs/bridge.log`
-

P3 双层审核机制（技术实现）

第一层 · 钉钉企业应用审核

- Stream Mode 连接需 `DINGTALK_APP_KEY / DINGTALK_APP_SECRET` 换取 WSS 凭据 (ticket)
- 走企业内部 API 网关 `DINGTALK_OPENAPI_ENDPOINT`，非企业内不可达
- 消息自带 `senderStaffId`（钉钉员工唯一身份），用于后续白名单核对

第二层 · 远端白名单（CSV）严格审核

`contact_map.py` 每次转发前实时拉取 `CONTACT_MAP_URL`（无本地缓存、无本地兜底）：

拉取远端 CSV（UTF-8 带 BOM，4 列：姓名 / 工号 / 微信备注名 / 钉钉uuid）

↓

姓名必须完全命中（先对昵称做括号剥离 `normalize_sender`）

↓

身份校验（二选一）：

- 行内填了 uuid → 要求钉钉 `senderStaffId` 与 uuid 完全一致
- 行内无 uuid → 退化为工号匹配

↓

命中 → 返回微信联系人；否则返回空 → 拒绝转发

失败兜底（安全优先）：- 白名单拉取失败（网络/HTTP 异常）→ 同样拒绝转发，并钉钉回复"白名单暂时无法获取，请稍后重试" - 支持 CSV / XLSX 自动识别（按 URL 扩展名与 Content-Type）

P4 微信自动化实现 (WeChatKeyboardBot)

采用 UIA (uiautomation) + pyautogui 桌面自动化：

1. 激活微信主窗口 (UIA 查找)
2. `Ctrl+F` 打开搜索 → 复制联系人姓名到剪贴板 → 粘贴 → 回车打开聊天窗口
3. 文件放入剪贴板 (Win32 `CF_HDROP`)
4. `Ctrl+V` 粘贴到输入框 → 回车发送
5. 附鼠标卡死检测/恢复机制，防止自动化中断

备用方案 3A：`explorer /select` 打开资源管理器 + 右键菜单复制文件（供主方案异常时切换）

P5 可靠性工程

机制	实现
开机自启	Startup 启动器 → start_bridge.bat
崩溃自愈	start_bridge.bat 内 :loop : bridge 退出后 5 秒自动重启
防重复实例	check_bridge.ps1 检测已有 bridge.py，避免重复监听
消息去重	_sent_messages 按 message_id 去重，同一条只处理一次
线程隔离	微信操作在独立线程，不阻塞钉钉事件循环
审计写入锁	Excel 写入加线程锁，多文件并发不冲突
微信拉起	启动时检测 Weixin.exe，未运行则自动拉起

P6 配置与部署

运行环境：Windows Server / Python 3.11 (venv)

`.env` 关键配置：

变量	说明
<code>DINGTALK_APP_KEY</code> / <code>DINGTALK_APP_SECRET</code>	钉钉企业内部应用凭证
<code>DINGTALK_OPENAPI_ENDPOINT</code>	内网 API 网关地址
<code>CONTACT_MAP_URL</code>	远端白名单 CSV 地址 (如 <code>https://ljrtec.cloud/contact_map.csv</code>)
<code>DOWNLOAD_DIR</code>	文件落盘目录
<code>WECHAT_TARGET</code>	固定联系人 (可选)
<code>LOG_LEVEL</code> / <code>UIA_TIMEOUT</code>	日志级别 / 自动化超时

启动：`start_bridge.bat` (自动拉起微信、防重复、后台运行 + 崩溃自愈)

依赖：`dingtalk-stream`、`uiautomation`、`pyautogui`、`pyperclip`、`pywin32`、`requests`、`openpyxl`、`python-dotenv`

P7 日志与审计

- 运行时日志 `logs/bridge.log` : 消息接收、白名单命中/拒绝、转发、错误全程留痕
- 转发审计表 `logs/bridge_records.xlsx` (7 列) :

发送时间	发送人姓名	uuid(工号)	发送至微信的人	文件大小(字节)	文件格式	文件保存位置
2026-08-14 10:42:10	熊熙辰	A2417346	熊熙辰	173355	pptx	...\downloads\...pptx

- 远端白名单变更即时生效 (每次转发实时拉取)

P8 测试验证矩阵

场景	结果
白名单命中（uuid 行）	转发成功 + 写 Excel
白名单命中（工号行）	转发成功
白名单外人员	拒绝 + 钉钉回复"不在白名单"
白名单拉取失败	拒绝（安全兜底）+ 回复"暂时无法获取"
带括号昵称（如 林韦义（Ace...））	normalize 剥离括号后命中
提示一致性	拒绝时不出现"发送中"误导提示
崩溃自愈 / 防重复 / 去重	均已实测

P9 已知边界与后续改进

- 微信 UI 自动化依赖窗口稳定，微信改版需同步适配；需要用户桌面会话运行（GUI 自动化，无法作为 Session0 系统服务）
 - 白名单为远端静态 CSV：集中维护、即时生效，但多端并发编辑需约定；后续可升级为在线审批式管理
 - 可扩展方向：图片/批量文件转发、发送失败自动重试队列、转发量报表/看板、失败告警通知
-

(技术层版完)