



# Browser Use Agent 在安保业务流程自 动化中的应用

Application of Browser User Agent in Security Process Automation

汇报人：李振邦

汇报时间：2025.12.26

# 目录

## CONTENTS



### 安保业务 系统

Security Management  
System

01



### 流程自动 化

Process Automation

02



### 浏览器自动 化智能体

Browser Use Agent

03



### 实际案例 演示

Case Demonstration


04

# 001 | 安保业务系统中缺少智能化的辅助工具



业务系统复杂度高、流程衔接自动化程度低、重度依赖人工经验


### 传统交互模式下安保业务应用系统的局限



操作步骤多、路径长  
交互繁琐




专业性强  
使用说明书800+页



各平台相互孤立  
需人工衔接

### 新型交互模式



自动执行任务  
业务系统流程自动化

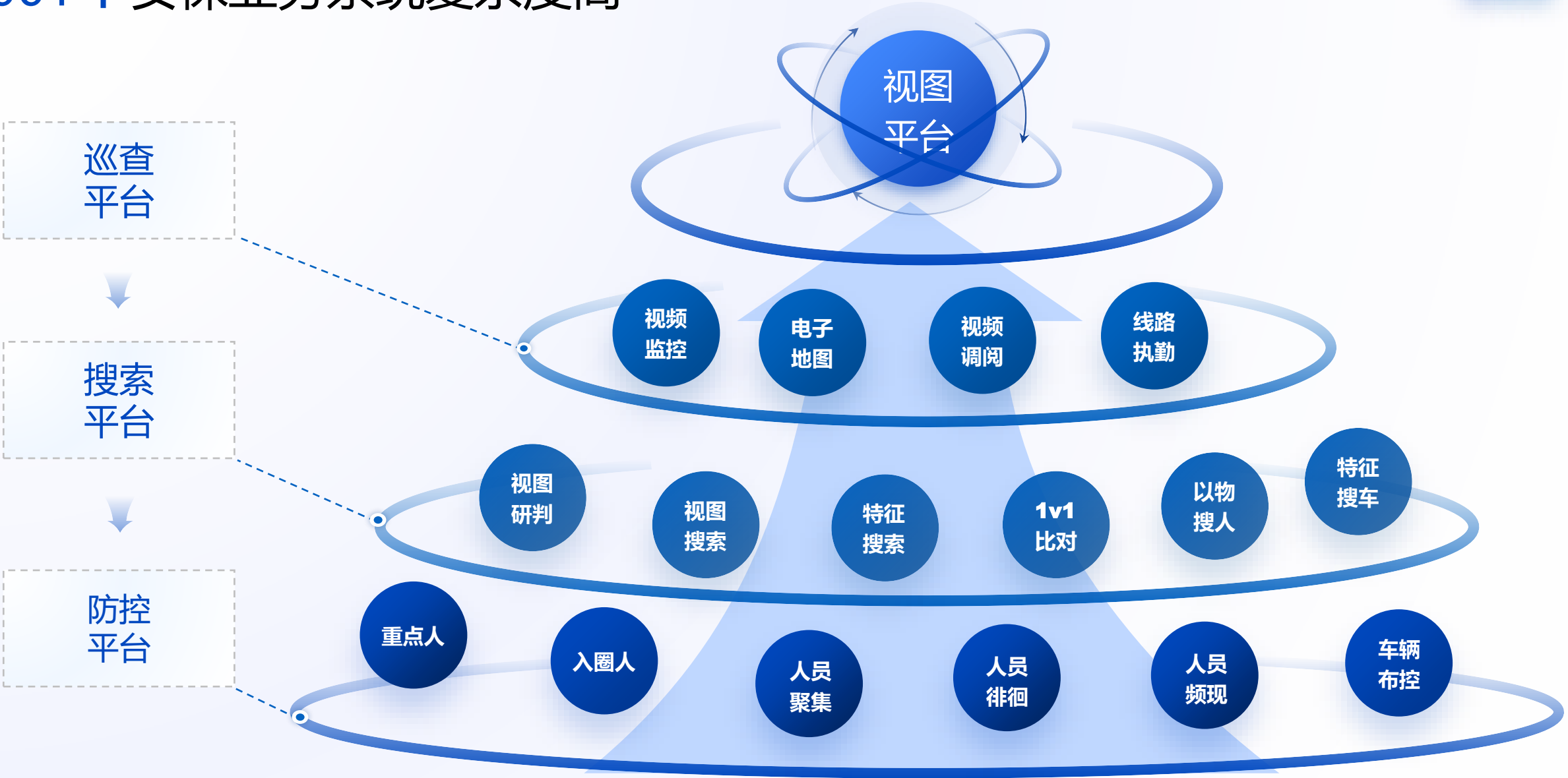


跨系统联动  
数据跨系统流转

海康视频图像综合应用平台产品v2.3使用说明书 (7).pdf

5.15 搜索.....	792
5.16 地图.....	793
5.17 人脸布控.....	795
5.18 车辆布控.....	796
5.19 消息.....	797
5.20 人员档案.....	798
5.21 车辆档案.....	799
5.22 个人中心.....	800

# 001 | 安保业务系统复杂度高



# 002 | 流程自动化软件



流程自动化的典型产品形态为 RPA (Robotic Process Automation)，通过模拟人类在计算机界面上的操作逻辑，在不改变企业现有 IT 架构的前提下，实现跨系统、跨平台的业务衔接，能够极大地提升数据流转的效率与精准度。



## 003 | 浏览器自动化



RPA软件的核心技术之一是浏览器自动化技术，用于模拟人类用户在Web环境下的交互行为，实现复杂业务流程的自动化处理。

### ○ 问题回顾01 | Review of annual issues one

#### 浏览器自动化技术的作用？

通过对DOM（文档对象模型）的解析，精准地识别网页上的输入框、按钮、下拉菜单等元素，执行点击、输入等动作，自动完成多系统之间的数据搬运和流程流转。

### ○ 问题回顾02 | Review of annual issues two

#### 与安保业务平台的关系？

目前绝大多数主流的安保业务平台（如视频监控系统 VMS、综合安防集成平台等）均为Web应用。面临挑战：系统多、接口闭塞、人工手动切换操作效率低。

# 003 | 浏览器自动化技术发展历史

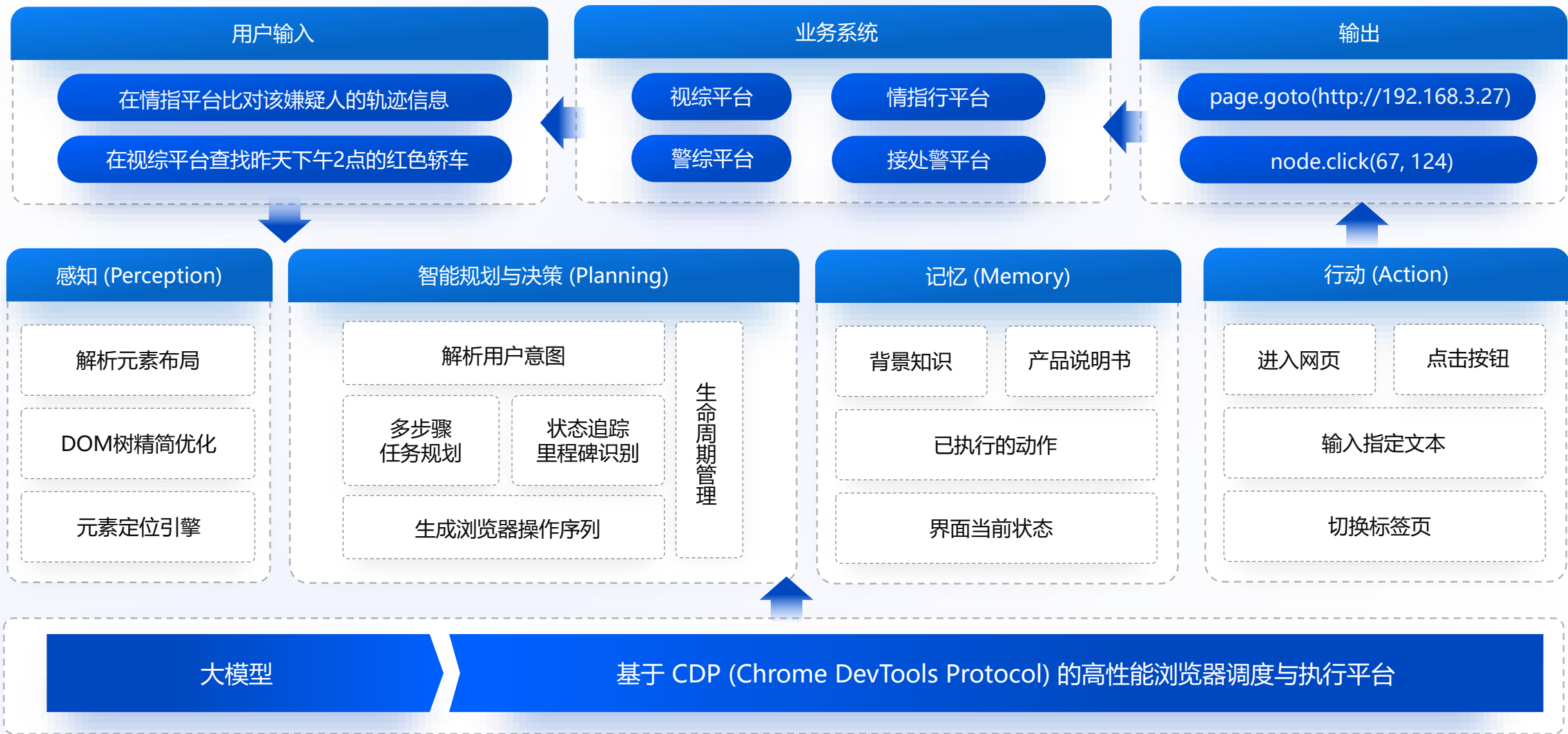


浏览器自动化技术经历了从初期依赖元素定位符的静态脚本编写，到中期基于底层通信协议的深度控制，再到当前利用大语言模型实现自然语言指令解析与动态环境自主适应的演进历程。





# 003 | Browser Use Agent: AI驱动的浏览器自动化框架







	GUI Agent	Browser Use Agent	Phone Use Agent	Computer Use Agent
运行平台	高层技术概念，按平台可分为后续三类具体Agent	浏览器环境 (Chrome, Edge 等 Web 平台)	移动端操作系统 (Android, iOS)	桌面端操作系统 (Windows, macOS, Linux)
概念介绍	泛指能够像人类一样通过视觉感知图形界面，并执行点击、输入等交互行为的 AI 代理。	专注于网页自动化，能够理解网页 DOM 结构，跨页面执行复杂的工作流。	模拟人类在手机屏幕上的操作，通过截图理解 UI 元素，实现 手机App 自动化任务。	具备全局操作系统权限，能跨窗口、跨软件进行通用操作，直接“接管”计算机鼠标键盘。
代表产品	UI-TARS AgentS3	Microsoft/playwright-mcp browser-use.com	OpenGLM 豆包手机	Claude Computer Use Microsoft UFO <sup>2</sup>

# 003 | 各 GUI Agent 可用性



Agent 名称	操作系统	任务类型	测试结果	耗时	使用模型	备注 / 表现
CUA	Windows 11	基于视觉 (OS)	成功	150s	glm-4.5v	表现稳定。
CUA	Ubuntu 24.04	基于视觉 (OS)	失败	250s	glm-4.5v	文本输入错误（武林林传），且错误误判为成功。
UI-TARS-desktop	Windows 11	基于视觉 (OS)	成功/失败	88s	doubao-1-5-thinking	结果1成功；结果2因未输入回车键且误判网络卡顿而失败。
UI-TARS-desktop	Windows 11	基于 DOM (Web)	成功	29s-48s	doubao-1-5-thinking	速度较快，适用于浏览器任务。
UI-TARS-desktop	Ubuntu 24.04	基于视觉 (OS)	失败	-	doubao-1-5-thinking	未执行有效操作却误判为成功。
UI-TARS-desktop	Ubuntu 24.04	基于 DOM (Web)	失败	-	doubao-1-5-thinking	未执行有效操作却误判为成功。
Microsoft UFO2	Windows 11	基于视觉 (OS)	成功	240s	qwen-vl-max	耗时较长，但任务完成成功。
browser-use	Ubuntu 24.04	基于 DOM (Web)	成功	109s	Qwen3-VL-235B	Ubuntu 平台上唯一推荐的 Web 任务 Agent。
Agent S3	Windows 11	基于视觉 (OS)	失败	9min+	gpt-5-2025-08-07	陷入死循环（反复聚焦地址栏），无法确认操作成功。

# 003 | 各 GUI Agent 可用性



基于视觉感知的 Computer Use Agent 尚处非成熟阶段，而基于DOM驱动的 Browser Use Agent 具有更好的执行成功率与响应速度优势，是目前解决基于Web的安保业务系统平台自动化难题现实最优技术路径。

测评  
结论

	Windows 11	Ubuntu 24.04
Web 任务	browser-use UI-TARS	browser-use
OS 任务	CUA UFO2	✗ 无可用推荐 (全失败)

Web 任务 仅浏览器/DOM 操作  
OS 任务 涉及桌面/全局视觉)

## 004 | 案例演示一：在模拟攒机网站自动配置电脑装机方案

The image displays a web browser window showing a PC configuration website (模拟攒机) and a terminal window running automation scripts.

**Website Interface (模拟攒机):**

- Header: 中关村在线 产品报价 网站导航 手机客户端 加入收藏 产品入库 商家库推广
- Navigation: 模拟攒机 攒机指南 我的方案 网友方案 大师方案 配置排行榜 网友首选配件 攒机问答 天梯图
- Current Page: 请选择配件 \*号为必选项
- Categories: CPU\*, 主板\*, 内存\*, 硬盘, 固态硬盘, 显卡, 机箱, 电源, 散热器, 显示器, 鼠标, 键盘, 更多
- Current Selection: CPU\* (Selected: Intel Core i5 13490F, Price: ¥1599)
- Other components: 主板\*, 内存\*, 硬盘, 固态硬盘, 显卡, 显示器, 机箱 (all marked as "请选择商品")

**Terminal Window (Terminal):**

```
(browser_use_py312) zhbli@zhbli-91C6A09QKX:~/p
projects/20251216GUIAgent/code/browser_use_exam
ple$ python ./my_browser_use.py
INFO [utils] Copied profile (Default) and
Local State to temp directory: /tmp/browser-us
e-user-data-dir- 2xn9rmu
INFO [service] Using anonymized telemetry,
see https://docs.browser-use.com/development/
telemetry.
INFO [Agent] Found URL in task: https:/
/zj.zol.com.cn/, adding as initial action...
INFO [Agent] Task:
https://zj.zol.com.cn/ 定制攒机方案

1. 配置 CPU 主板 内存 硬盘
当装机配置单中某个设备没有配置时，对应
条目会显示“请选择商品”，且该设备各型号产品对应的
“加入配置单”按钮为蓝色。
当装机配置单中某个设备成功配置时，对应
条目会显示具体型号产品，且被添加的产品对应的“
加入配置单”按钮为灰色。
当装机配置单中某个设备成功配置后，进行
下一个设备的配置，不能重复配置同一设备。切勿点
击已变成灰色的“加入配置单”按钮。

2. 上述四个硬件配置完成后，在 名称 文
本框 输入 我的装机配置单

3. 点击预览按钮。点击预览按钮后立即结
束所有操作。

INFO [Agent] Starting a browser-use agent
with version 0.11.2, with provider=openai and
model=qwen-vl-max
INFO [Agent] navigate: url: https://
zj.zol.com.cn/, new_tab: False
INFO [tools] Navigated to https://zj.zo
l.com.cn/
INFO [Agent]

INFO [Agent] Step 1:
```

**通过浏览器自动化实现自动配置电脑装机方案**



# 004 | 案例演示二：自动播放指定点位监控视频



```
zcc@kyl-1807-south-east1: ~/zhbli
文件(F) 编辑(E) 视图(V) 搜索(S) 终端(T) 标签(B) 帮助(H)

zcc@kyl-1807-south-east1: ~/zhbli

INFO [Agent] Step 3:
INFO [Agent] Eval: Waited for an additional 5 seconds for the device list to complete loading. Verdict: Success
INFO [Agent] Memory: We've been waiting for 10 seconds total (5+5) for the device list to load. The page still shows '加载设备中...' indicating it's still in progress. We need to continue waiting until the device list finishes loading and becomes interactive with a search box.
INFO [Agent] Next goal: Wait for an additional 5 seconds to allow more time for the device list to finish loading.
INFO [Agent] wait: seconds: 5
INFO [tools] waited for 5 seconds
INFO [Agent]

INFO [Agent] Step 4:
INFO [Agent] Eval: Waited for 15 seconds total (5+5+5) for the device list to load. The device list has now loaded successfully with multiple devices visible including our target device. Verdict: Success
INFO [Agent] Memory: We've successfully navigated to the URL and waited for the device list to load completely. The target device '独山大道-南阳市二十五小学-西报警柱-出入口-1' is now visible in the device list with index 1142. Next step is to click on this specific device entry before clicking the '播放视频' button.
INFO [Agent] Next goal: Click on the target device entry '独山大道-南阳市二十五小学-西报警柱-出入口-1' (index 1142) to select it.
INFO [Agent] click: index: 1142
INFO [tools] Clicked div "设备ID: 41130204001310020767 型号..."
INFO [Agent]

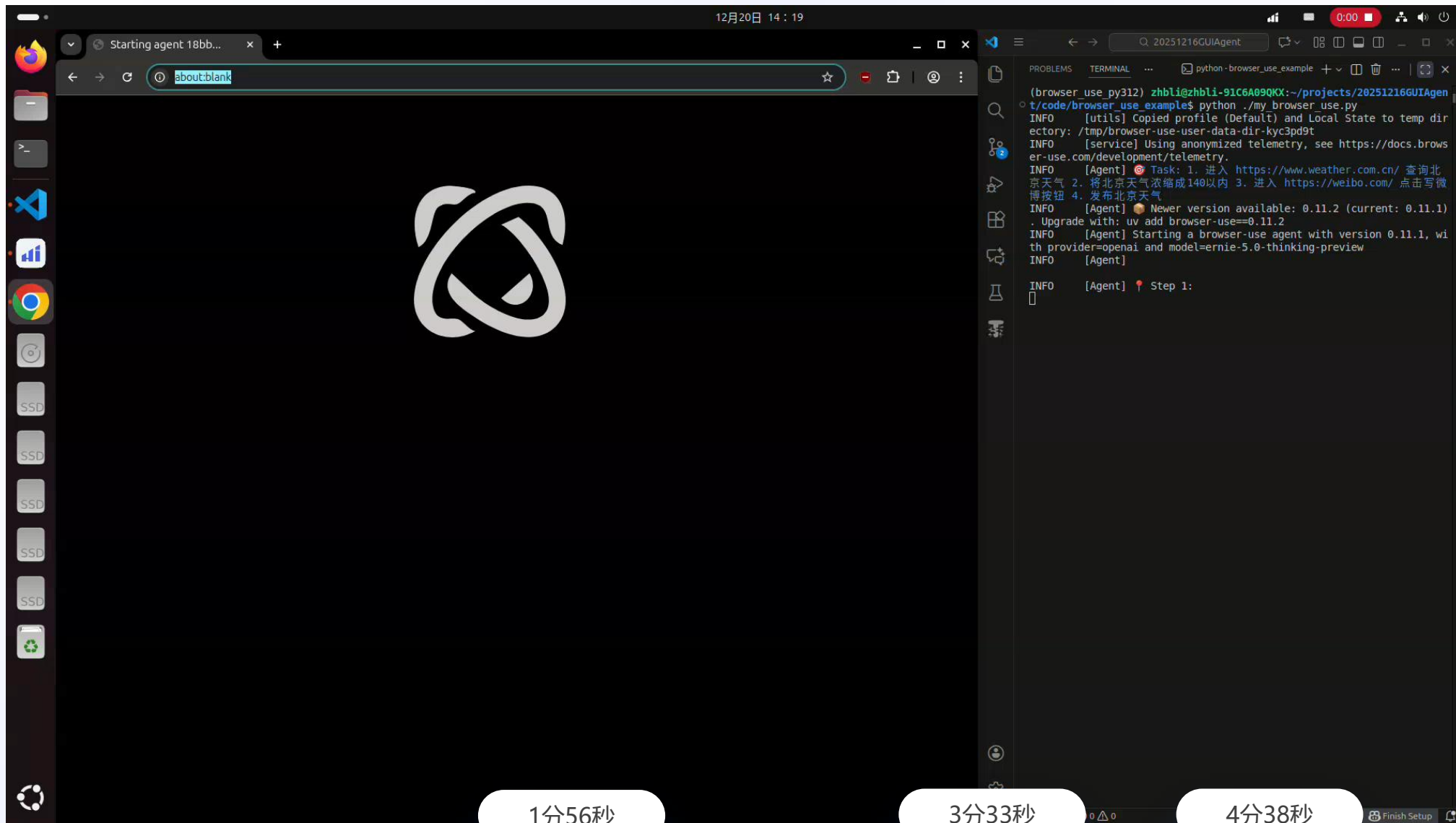
INFO [Agent] Step 5:
INFO [Agent] Eval: Successfully selected the target device entry and displayed its details in a popup window. Verdict: Success
INFO [Agent] Memory: Completed all previous steps: navigated to URL, waited for device list to load, found and clicked on target device entry. Now ready to click the '播放视频' button to complete the task.
INFO [Agent] Next goal: Click on the '播放视频' button (index 1883) to play the video for the selected device.
INFO [Agent] click: index: 1883
INFO [tools] Clicked button "播放视频"
INFO [Agent]

INFO [Agent] Step 6:
INFO [Agent] Eval: Successfully played the video for the target device as requested by user. Verdict: Success
INFO [Agent] Memory: Completed all 5 steps of user request: navigated to URL, waited for device list to load, searched for target device, selected it from results, and clicked play button. Video is now playing in a popup window.
INFO [Agent] Next goal: Call done action with success=true as all user requirements have been met.
INFO [Agent] done: text: The task has been successfully completed. The video for '独山大道-南阳市二十五小学-西报警柱-出入口-1' is now playing in a popup window., success: True, files_to_display: []
INFO [Agent]

Final Result:
The task has been successfully completed. The video for '独山大道-南阳市二十五小学-西报警柱-出入口-1' is now playing in a popup window.

INFO [Agent] Task completed successfully
INFO [BrowserSession] on BrowserStopEvent - Calling reset() (force=True, keep_alive=None)
INFO [BrowserSession] [SessionManager] Cleared all owned data (targets, sessions, mappings)
INFO [BrowserSession] Browser session reset complete
INFO [BrowserSession] Browser session reset complete
70.75240159034729
INFO [backoff] Backing off send_request(...) for 1.0s (requests.exceptions.ConnectionError: HTTPSConnectionPool(host='eu.i.posthog.com', port=443): Max retries exceeded with url: /batch/ (Caused by NewConnectionError("HTTPSConnection(host='eu.i.posthog.com', port=443): Failed to establish a new connection: [Errno 111] Connection refused")))
INFO [backoff] Backing off send_request(...) for 1.1s (requests.exceptions.ConnectionError: HTTPSConnectionPool(host='eu.i.posthog.com', port=443): Max retries exceeded with url: /batch/ (Caused by NewConnectionError("HTTPSConnection(host='eu.i.posthog.com', port=443): Failed to establish a new connection: [Errno 111] Connection refused")))
INFO [backoff] Backing off send_request(...) for 3.0s (requests.exceptions.ConnectionError: HTTPSConnectionPool(host='eu.i.posthog.com', port=443): Max retries exceeded with url: /batch/ (Caused by NewConnectionError("HTTPSConnection(host='eu.i.posthog.com', port=443): Failed to establish a new connection: [Errno 111] Connection refused")))
ERROR [backoff] Giving up send_request(...) after 4 tries (requests.exceptions.ConnectionError: HTTPSConnectionPool(host='eu.i.posthog.com', port=443): Max retries exceeded with url: /batch/ (Caused by NewConnectionError("HTTPSConnection(host='eu.i.posthog.com', port=443): Failed to establish a new connection: [Errno 111] Connection refused")))
(browser_use_py312) zcc@kyl-1807-south-east1: ~/zhbli
```

## 004 | 案例演示三：自动查询天气发送至微博（跨站点操作能力）



## 004 | 案例演示四：词条搜索能力



The screenshot displays a Linux desktop with a dark theme. On the left is a vertical dock containing icons for Firefox, a file manager, and several SSD storage icons. The main workspace is divided into two panels. The top panel is a code editor showing a Python file named `my_browser_use.py`. The code defines an asynchronous function `my_task()` that initializes a `ChatOpenAI` model, creates an `Agent` with a specific task and LLM, and then runs the agent. The task is to search for 'Wu Lin's biography' on Baidu. The bottom panel is a terminal window showing the command `python ./my_browser_use.py` being executed. The desktop background features a stylized illustration of a cat's face. In the bottom right corner, there is a home icon and the text '主目录'.

```
12月15日 16:26
0:00
my_browser_use.py U X
my_browser_use.py
24 async def my_task():
29
30     llm = ChatOpenAI(model='Qwen/Qwen3-VL-235B-A22B-Instruct', api_key=api_key, base
31
32     agent = Agent(
33         task="介绍百度百科中搜索'武林外传'词条",
34         llm=llm,
35         use_vision=True
36     )
37     start = time.time()
38     await agent.run()
39     end = time.time()
40     print(end-start)
41
42 # Just call it like any async function
(browser_use_py312) zhbli@zhbli-91C6A09QKX:~/projects/gui_agent_example/browser_use_example$ python ./my_
browser_use.py
```



## 004 | 案例演示五：后台部署能力



The screenshot shows a development environment with a terminal window and a browser window. The terminal window is titled "20251216GUIAgent" and shows a command prompt where the user has entered the command `python ./remote_deployment.py`. The browser window is titled "zhblli@zhblli-91C6A09QKX: ~/projects/20251216GUIAgent/code/browser\_use\_example" and displays the output of the command. The output shows the version of FFmpeg (6.1.1-3ubuntu5) and a list of enabled features.

```
(base) zhblli@zhblli-91C6A09QKX:~/projects/20251216GUIAgent/code/browser_use_example$ python ./remote_deployment.py
```

```
(base) zhblli@zhblli-91C6A09QKX:~/projects/20251216GUIAgent/code/browser_use_example$ ffmpeg -f x11grab -i :0 -f pulse -i default screen_record.mp4
ffmpeg version 6.1.1-3ubuntu5 Copyright (c) 2000-2023 the FFmpeg developers
  built with gcc 13 (Ubuntu 13.2.0-23ubuntu3)
  configuration: --prefix=/usr --extra-version=3ubuntu5 --toolchain=hardened --libdir=/usr/lib/x86_64-linux-gnu --incdir=/usr/include/x86_64-linux-gnu --arch=amd64 --enable-gpl --disable-stripping --disable-omx --enable-gnutls --enable-libaom --enable-libass --enable-libbs2b --enable-libcaca --enable-libcdio --enable-libcodec2 --enable-libdav1d --enable-libflite --enable-libfontconfig --enable-libfreetype --enable-libfribidi --enable-libglslang --enable-libgme --enable-libgsm --enable-libharfbuzz --enable-libmp3lame --enable-libmysofa --enable-libopenjpeg --enable-libopenmpt --enable-libopus --enable-librubberband --enable-libshine --enable-lsnpappy --enable-lssoxr --enable-lspeek --enable-libtheora --enable-libtwolame --enable-libvidstab --enable-libvorbis --enable-libvpx --enable-libwebp --enable-libx265 --enable-libxml2 --enable-libxvid --enable-libzimg --enable-openal --enable-opencore --enable-opengl --disable-sndio --enable-libvpl --disable-libmfx --enable-libdc1394 --enable-libdrm --enable-libiec61883 --enable-chromaprint --enable-frei0r --enable-ladspa --enable-libluray --enable-libjack --enable-libpulse --enable-librabbitmq --enable-librist --enable-librt --enable-libssh --enable-libsvtav1 --enable-libx264 --enable-libzmq --enable-libzvi --enable-lv2 --enable-sdl2 --enable-libplacebo --enable-librav1e --enable-pocketsphinx --enable-librsvg --enable-libjxl --enable-shared
  libavutil      58. 29.100 / 58. 29.100
  libavcodec     60. 31.102 / 60. 31.102
  libavformat    60. 16.100 / 60. 16.100
  libavdevice     60.  3.100 / 60.  3.100
  libavfilter     9. 12.100 /  9. 12.100
  libswscale     7.  5.100 /  7.  5.100
  libswresample  4. 12.100 /  4. 12.100
  libpostproc   57.  3.100 / 57.  3.100
```



打造以“智能体”为核心的安保业务流程自动化系统  
从“人工经验依赖”向“智能自主编排”的演进

