Stealth Architecture

The world's most advanced, undetectable, zero focus AI assistant for macOS and Windows.

This is the official technical whitepaper for **Ghost App by TsarChat.com**, the world's most advanced, powerful, and completely undetectable AI desktop assistant.

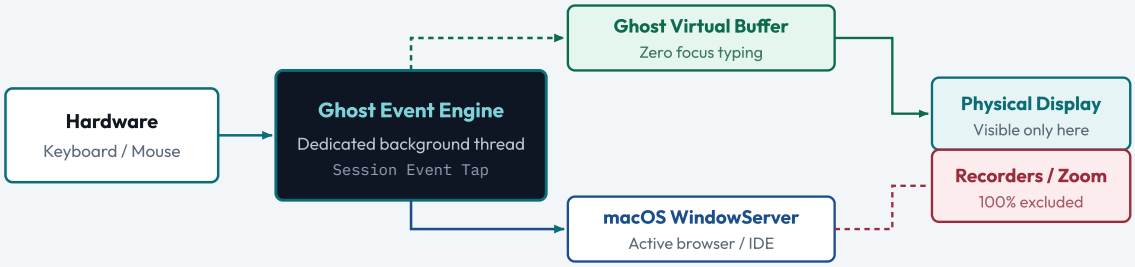
Engineered from the ground up for total privacy, zero focus operation, and absolute operational stealth, Ghost App combines proprietary event controls, display compositor exclusions, and deep operating system hardening to remain invisible to screen recorders, proctoring engines, software integrity tools, and accessibility scrapers.

EXECUTIVE SUMMARY

Traditional desktop applications leak window titles, trigger focus loss events, and expose shortcut patterns to browser event loggers. Ghost App operates in complete isolation beneath the application layer.

FIGURE 01 · OPERATIONAL PIPELINE

Hardware input is intercepted beneath WindowServer. Browser focus stays active. Screen recorders see nothing.



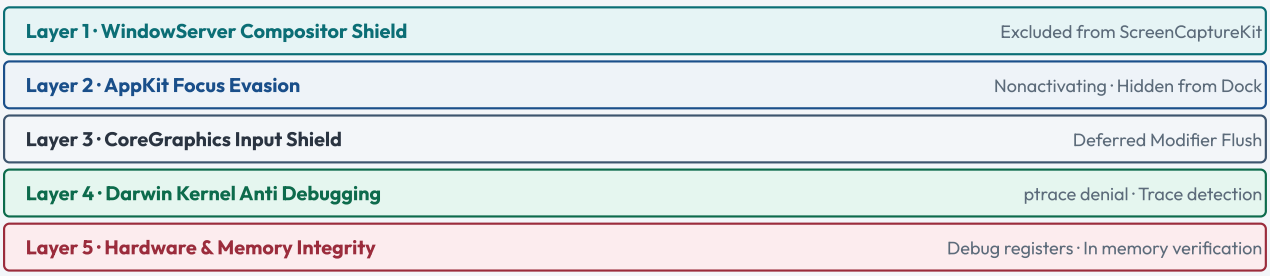
PROTECTION DIMENSION	GHOST APP BY TSARCHAT.COM
Screen Share & Recording	100% Invisible via OS Compositor
Browser Focus Loss (`onblur`)	Zero Focus Steal · Nonactivating
Hotkey Fingerprinting	Deferred Modifier Flush
Process Inspection	Dynamic Decoy Identity Rotation
Anti Debugging & Memory	Kernel Level ptrace Denial & Mach O Integrity
Input Capture Mode	Zero Focus Simulated Typing
Dock / App Switcher Presence	Invisible Agent Presence
Click Through Focus Mode	Full Passthrough Overlay
Secondary UI Capture Leaks	Protected Menus, Tooltips, and Popovers
Process & Accessibility Identity	System Style Camouflage
Context Capture	Silent OCR · No Flash · No Pasteboard
System Audio Capture	No Recording Indicator
Operator Recovery	Panic Hide and Live Privacy Reassertion

01 Low Level OS & Kernel Architecture

Ghost App operates at the interface of the macOS WindowServer, IOKit hardware layer, and the Darwin kernel, delivering a security posture no consumer AI desktop app can match.

FIGURE 02 · FIVE LAYER SHIELD

Active defense across compositor, focus, input, kernel, and hardware integrity.



A. WindowServer Compositor Invisible Rendering

When macOS captures the screen, WindowServer composites open windows into a unified frame. Ghost App directs the display server to render exclusively into the physical display buffer. To a capture stream, proctoring extension, or automated scraper, the Ghost App surface simply does not exist.

B. Nonactivating Focus Evasion

Monitoring tools track application focus loss through events such as `window.onblur`. Ghost App uses a nonactivating panel architecture. Clicking Ghost App or typing into its interface produces no application context switch. The browser or IDE underneath retains its active focus state at all times.

C. Kernel & Process Hardening

A multilayered defense engine neutralizes memory scanners, debuggers, and inspection tools:

Kernel Debugger Denial
Direct kernel directives prevent any process from attaching a debugger to Ghost App.

Process Trace Inspection
Actively checks kernel process structures for tracing flags and neutralizes unauthorized inspection.

Hardware Breakpoint Scanner
Inspects CPU debug registers at the Mach thread level to detect hardware breakpoints.

Mach O Symbol Integrity
Snapshots in memory symbol prologues to detect inline hooks or trampolines.

Dynamic Library Shield
Continuously audits mapped libraries and blocks unauthorized external injections.

Crash Dump Evasion
Intercepts fatal signals to prevent crash log generation. Zero forensic footprint.

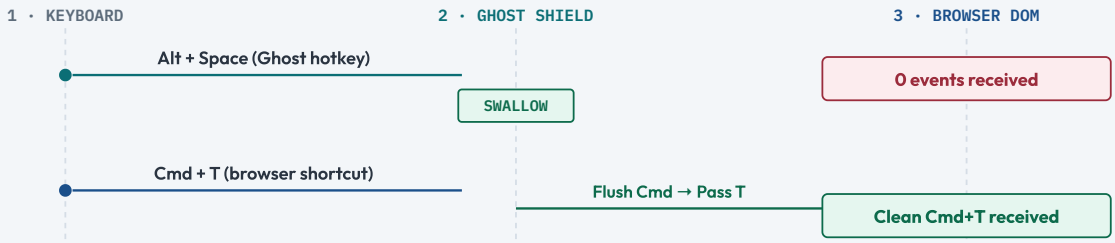
02 Next Generation Input & Stealth Features

A. Proprietary Browser Key Shield

Proctoring extensions listen for DOM `keydown` events. Ghost App's Browser Key Shield uses a Deferred Modifier Flush so Ghost chords never reach the browser, while normal shortcuts still work.

FIGURE 03 · HOTKEY VS BROWSER SHORTCUT

Modifiers are held in a pending queue. Ghost chords are swallowed. Legitimate shortcuts are flushed cleanly.



1. **Modifier Suppression**: Modifier keys are held in an isolated pending queue, not released to WindowServer.
2. **Hotkey Matching**: Ghost chords are swallowed entirely at the OS queue layer.
3. **Deferred Modifier Flush**: Other shortcuts, such as `Cmd+T`, are reinjected with a proprietary source tag and passed through cleanly.
4. **Synthetic Probe Resistance**: Hostile injected keystroke probes are distinguished from real hardware input so automated hotkey detectors cannot fingerprint the shield.

B. Zero Focus Simulated Typing

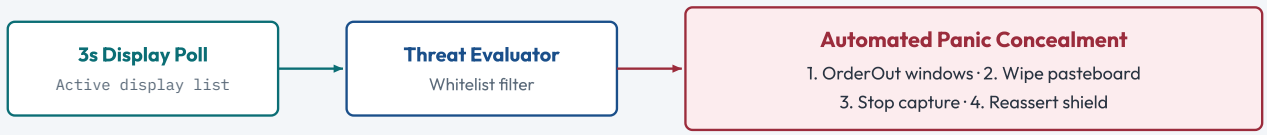
Traditional applications normally require window focus to accept typing. Ghost App intercepts keystrokes before WindowServer dispatches them to the frontmost browser. It routes characters into a simulated text buffer that supports insertion, deletion, caret navigation, and clipboard pasting with `Cmd+V`, while the underlying application remains active.

C. Active Countermeasures

- **Dynamic Decoy Identity**: An offscreen decoy window periodically rotates among generic utility titles. Enumeration tools see only innocuous noise.
- **Chain Head Hook Defense**: Ghost App periodically reasserts the front of the operating system event queue so external monitors cannot sit ahead of its shields.
- **Stream Detection & Panic Hide**: Unauthorized recording streams trigger automated Panic Hide. Ghost App removes its windows, clears the pasteboard, and stops capture in milliseconds.

FIGURE 04 · STREAM MONITORING & PANIC RESPONSE

Continuous display polling feeds a threat evaluator. Unauthorized streams trigger instant concealment.

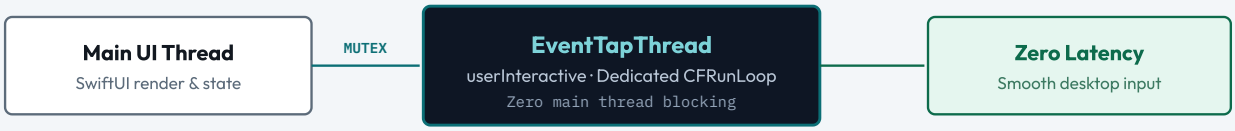


03 High Performance Threading Architecture

Stealth event controls never slow your computer. Ghost App runs its event engine on a dedicated background thread with high priority. The result is responsive input without desktop freezes.

FIGURE 05 · DECOUPLED EVENT TAP ENGINE

Input is serviced off the main run loop so SwiftUI rendering never blocks system events.



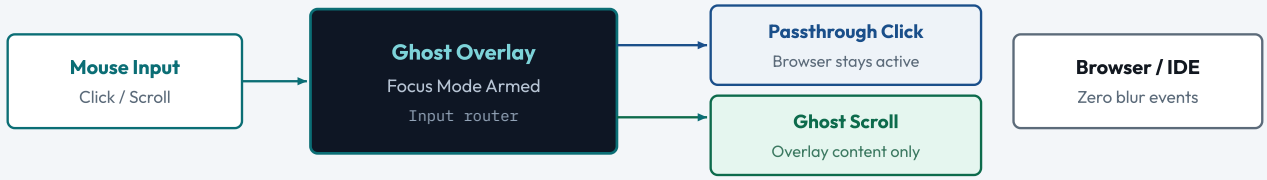
04 Invisible Presence & Click Through Focus Mode

Ghost App does not behave like an application. It behaves like a private display layer. Ordinary AI apps sit in the Dock, appear in Cmd+Tab, and steal mouse focus the moment you click them. Ghost App runs as a permanent invisible agent.

- **Zero Dock Footprint**: No Dock icon, no Cmd+Tab entry, and no Force Quit listing that looks like a consumer chat app.
- **Permanent Overlay Elevation**: Ghost App stays visually available above your work without ever becoming the active application.
- **Click Through Focus Mode**: Ghost App becomes a transparent input shield. Clicks pass directly to the browser or IDE underneath, while Ghost App remains visible only on your physical display.
- **Asymmetric Scroll Intelligence**: While Focus Mode is active, scrolling over Ghost App content scrolls Ghost App. Scrolling outside it controls the application beneath. There is no focus swap, no blur event, and no detectable window activation.

FIGURE 06 · CLICK THROUGH FOCUS MODE

Clicks pass through to the browser. Scroll over Ghost content stays with Ghost. The underlying app never loses focus.



05 Complete Capture Surface Hardening

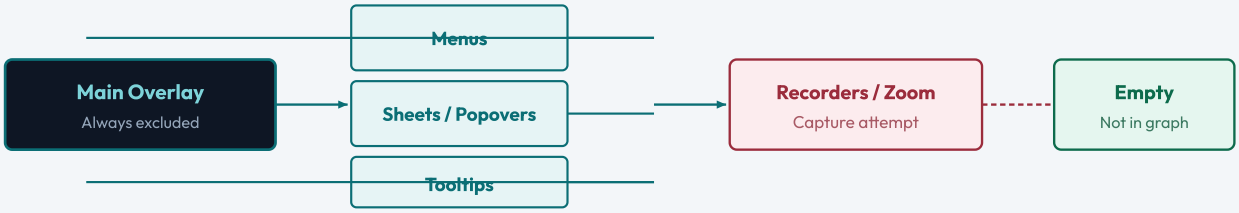
Invisibility is only as strong as the weakest popup. Ghost App does not protect only a single chat window and hope the rest of the interface stays quiet. Every secondary surface, including menus, sheets, popovers, and tooltips, inherits the same compositor exclusion policy as the main overlay.

- **Always On Capture Exclusion** protects the primary surface.
- **Automatic Inheritance**: Transient interface elements inherit protection, so a context menu cannot accidentally appear in Zoom, Meet, or a proctoring stream.
- **Timed Screen Share Mode**: When you intentionally want to demonstrate Ghost App, enable a controlled, temporary sharing window. When that window expires, invisibility reasserts itself automatically.
- **Self Capture Hygiene**: Even Ghost App's own screenshot tools exclude its private surfaces, so operational captures never reveal the overlay by accident.

For every recorder and scraper, Ghost App is not merely hidden. It is **not present in the capture graph**.

FIGURE 07 · CAPTURE SURFACE HARDENING

Main overlay, menus, sheets, and tooltips all inherit exclusion. Recorders see an empty capture graph.



06 Identity Cloaking & Launch Camouflage

Process lists and accessibility trees are attack surfaces. Ghost App refuses to look like an AI assistant.

- **Process Identity Camouflage**: Ghost App presents itself through innocuous system style identity patterns instead of an obvious consumer application name.
- **Polymorphic Launch Footprint**: Each launch uses a randomized, short lived execution path designed to minimize its static disk signature.
- **Accessibility Tree Camouflage**: Inspection trees do not advertise a chat assistant identity. Ghost App presents itself as ordinary system adjacent interface noise.
- **Dynamic Decoy Identity Rotation**: Window enumerators see rotating utility titles instead of a branded AI window.
- **Cursor & Chrome Discipline**: No flashy AI cursor or activation cues reveal the presence of an external overlay.

If an examiner lists processes, windows, or accessibility nodes, Ghost App is engineered to look like background system activity, not an interview assistant.

07 Silent Intelligence Capture

Capture context without drawing attention from operating system privacy indicators.

Silent OCR Snap

Pull onscreen question text into Ghost App memory with no screenshot flash, no pasteboard write, and no disruptive permission prompt during a session.

Stealth System Audio

Capture relevant system audio to understand spoken prompts without activating the macOS screen recording indicator that monitoring tools watch.

Content Blackout

Sensitive panels, including prompts and history, refuse to expose content while Click Through Focus Mode is active. This closes a common shoulder surfing and recording vulnerability.

Monitored Environment Design

Private context acquisition built for monitored environments, not a conventional screenshot feature with a marketing label.

08 Operational Command & Automatic Defense

Stealth is not a setting. It is an active defense system.

- **Instant Panic Concealment** (`Cmd+Shift+H`): One chord removes Ghost App from every display, stops input capture, optionally clears pasteboard residue, stops audio taps, and reasserts capture shields in milliseconds.
- **Configurable Panic Depth**: Operators choose the level of concealment, including clipboard clearing, audio shutdown, and complete application termination.
- **Live Recording and Mirror Detection**: Ghost App continuously evaluates display and session threat signals. Unauthorized mirroring or recording streams trigger automatic Panic Concealment without requiring human reaction time.
- **Stealth Health Check**: A live operational dashboard confirms the state of capture protection, focus evasion, input shields, and process hardening. A single **Reassert Privacy Mode** action restores every shield to its maximum posture.
- **Network Stealth**: Ghost App can route model requests through a SOCKS proxy that is compatible with Tor, concealing the direct client IP address. When the inference service provides a compatible onion endpoint, traffic can remain inside the Tor network from the desktop agent to the inference edge.

CAPABILITY SUMMARY

Ghost App by TsarChat.com is not a chat window with a simple hide from screenshot option. It is a complete invisible operating layer that combines compositor exclusion, zero focus input, click through presence, identity camouflage, silent context capture, and automatic panic defense in one coherent system that no consumer AI desktop app can match.