PrivacyDeck: A Tangible Interface for Low-Friction Desktop Privacy Management

Mustafa Durani
Ludwig Maximilian University of Munich
Management and Digital Technologies
Munich, Germany
m.durani@campus.lmu.de

Tristan Häuser
Ludwig Maximilian University of Munich
Media Informatics
Munich, Germany
tristan.haeuser@campus.lmu.de

Franziska Oberländer
Ludwig Maximilian University of Munich
Management and Digital Technologies
Munich, Germany
f.oberlaender@campus.lmu.de

Simon Rödig
Ludwig Maximilian University of Munich
Human-Computer Interaction
Munich, Germany
simon.roedig@campus.lmu.de

Figure 1: The PrivacyDeck tangible control panel placed on a desktop workspace, providing direct physical control over privacy and security functions. Video demonstration: https://www.youtube.com/watch?v=T16Eui8qbHk

Abstract

Protecting personal privacy on desktop computers remains cumbersome despite available system controls. Critical privacy features are often buried within complex menus, fragmented across applications, and lack immediate, interpretable feedback, leading users to neglect protective actions in favour of convenience.

We present PrivacyDeck, a tangible, USB-connected control panel that externalises high-frequency privacy actions into persistent physical controls. Users can lock the operating system, mute the microphone, and manage camera and connectivity settings with single gestures, while a modular personalised avatar provides real-time visual feedback on system exposure. The hardware integrates buttons, toggles, SPI screens, a slider, and an avatar with LEDs, communicating with an OS-level Python daemon and a graphical dashboard.

Initial deployment illustrates reliable operation on Linux with low-latency feedback, while exposing platform-specific limitations on Windows and macOS that underscore the need for a unified, open framework for cross-platform privacy and security functions. PrivacyDeck suggests that tangible interfaces can meaningfully reduce interaction friction in everyday privacy management, and that future devices would benefit from standardised system-level privacy APIs supporting proactive, situation-aware user behaviour.

CCS Concepts

• **Human-centered computing** → **Human computer interaction (HCI)**; *Interaction design*; • **Security and privacy** → **Usability in security and privacy**; *Human and societal aspects of security and privacy*.

Keywords

privacy management, tangible user interfaces, usable security, desktop privacy, physical computing, privacy awareness, situation awareness, ambient displays, mixed-initiative interfaces, explainable feedback, privacy fatigue, hardware privacy controls, avatar-based feedback

1 Introduction & Motivation

Protecting personal privacy on everyday computing devices remains unnecessarily difficult. Privacy failure is not primarily a knowledge problem, but a friction problem: although protective tools and settings are technically available, they are often buried within complex, nested operating system menus and fragmented across multiple applications. Users must navigate unintuitive interfaces, remember hidden shortcuts, and manage app-specific controls without a unified dashboard.

This complexity leads to the underutilisation of critical privacy features and the neglect of basic protective behaviours in favour of convenience, reducing users' sense of control [3]. The absence of centralised management and the lack of immediate, tangible feedback about privacy states, such as whether a camera or microphone is active, further amplify uncertainty and discomfort, especially when control appears to be taken away [2, 12].

As a result, current privacy management systems fail not because users are unaware of risks, but because interaction barriers, fragmented controls, and unclear feedback prevent timely, confident, and effective privacy action.

To address these challenges, the PrivacyDeck was developed: a compact, USB-connected physical control panel that moves high-frequency privacy actions out of fragmented software menus and into dedicated, always-accessible physical controls. By moving privacy management into a persistent tangible form, the device reduces the interaction overhead that prior work identifies as the primary barrier to consistent privacy behaviour [3, 12].

By placing controls directly on the desk within constant reach, the PrivacyDeck transforms passive security awareness into proactive behaviour. Instead of requiring users to navigate software menus under cognitive load, each protective action is reduced to a single physical gesture, building the kind of muscle memory that makes privacy protection sustainable across shared office environments, remote work settings, and public spaces.

2 Related Work

Researchers have pursued several approaches to closing the gap between privacy awareness and protective action, spanning software dashboards, tangible interaction frameworks, physical device prototypes, and icon design studies. Each addresses part of the problem; none fully resolves the combination of fragmented controls, absent feedback, and high interaction cost that defines everyday privacy management.

2.1 Software Dashboards

The most established approach is the software dashboard. Farke et al. [7] evaluated Google's My Activity, a transparency tool that lets users review and delete activity data collected across Google services. They found that while the tool surprised users with the volume of data collected, participants largely neither engaged with its control features nor changed their behaviour afterward. Thus, , transparency alone is insufficient to motivate protective action. Alashwali et al. [3] documented a parallel pattern in remote work: workers commonly experience privacy invasions during video calls, such as accidental camera or microphone exposure, yet only a minority take active countermeasures despite being aware of the risk. Shao et al. [16] formalised this disengagement as privacy fatigue, showing that repeated exposure to complex privacy decisions leads users to abandon protective behaviour entirely. Together, these findings establish that software based transparency and control tools tend to fall short when they demand sustained effort across fragmented interfaces. The PrivacyDeck responds to this by moving core privacy controls out of software menus and onto a single physical surface where each action requires one gesture.

2.2 Shared and Cognitively Demanding Environments

A second line of research examines why privacy breaks down in shared and cognitively demanding environments. Song et al. [18] studied account sharing in the workplace and found it to be widespread, with collaboration across shared accounts serving as a norm rather than an exception. While their work surfaces important challenges around access control and activity accountability, it does not address the resulting problem of ambiguous privacy state across sessions and users: when multiple people move through the same device or account, it becomes unclear what sensors are active, what data has been left behind, and what the current exposure level actually is. Endsley [6] provides a theoretical explanation for why this ambiguity is so consequential: under multitasking or cognitive load, situation awareness degrades in predictable and systematic ways, causing users to lose track of environmental states that require ongoing monitoring. Applied to shared workspaces, this means that even users who intend to manage their privacy actively are likely to miss critical state changes when attention is divided. The PrivacyDeck addresses this by externalising privacy state through persistent hardware indicators and an avatar that communicates risk at a glance, without requiring active monitoring or memory.

2.3 Tangible and Physical Interfaces

A third strand of work has proposed tangible and physical interfaces as alternatives to software controls. Ahmad et al. [2] studied how bystanders perceive IoT devices such as cameras and voice assistants and found that ambiguous device states, where "off" does not necessarily mean fully deactivated, create persistent uncertainty about whether recording is occurring. They introduced the concept of tangible privacy, recommending that devices convey sensor states through unambiguous physical indicators, hardware switches, and camera shutters, though their contribution remained at the level of design principles rather than a deployable system. The practical consequences of this ambiguity are measurable even outside IoT contexts. Portnoff et al. [15] found that only 45% of participants noticed an active webcam LED during ordinary computing tasks, a figure that fell to just 5% when attention was directed elsewhere. Even when physical feedback mechanisms exist, they can routinely fail under the realistic conditions of divided attention that characterise everyday desk work. Mehta et al. [12] operationalised these principles further by developing Privacy Care, an interaction framework for tangible privacy management in ubiquitous computing. The framework establishes Awareness and Control as core design goals and defines three interaction tenets, namely Direct, Ready to Hand, and Contextual, to guide the design of physical privacy tools. However, Privacy Care was validated through focus

groups with older adults rather than as an implemented device for everyday desktop use, which limits its direct applicability to the present context. Al Muhander et al. [13] came closest to implementation with PriviFy, a tangible interface for configuring IoT privacy preferences using knobs, buttons, lights, and notifications. Across iterative design sessions and a subsequent high-fidelity evaluation totalling forty participants, PriviFy achieved higher usability scores and lower perceived effort than conventional software controls, confirming that consistent, physically uniform controls support ease of use. However, as an arXiv preprint, these findings await formal peer review. PriviFy is also scoped to IoT device configuration and does not address desktop privacy functions such as camera and microphone control, clipboard management, or system locking. The PrivacyDeck extends this by implementing a fully functional physical panel for core desktop privacy actions, combining dedicated hardware switches with real-time visual feedback through LEDs, a live camera preview, and an audiometer for microphone input.

2.4 Visual Communication and Icon Design

Finally, researchers have investigated how visual communication affects the usability of privacy interfaces. Delgado Rodriguez et al. [5] elicited and clustered symbols and metaphors associated with privacy and security through brainstorming sessions and expert evaluation, finding that camera and microphone symbols were consistently grouped under privacy-related themes. While their sample was small and the study exploratory, the findings identify recognisable visual anchors for privacy interface design. Habib et al. [9] tested how effectively icons and link texts convey privacy choices and found that unfamiliar or ambiguous symbols without accompanying text frequently cause misinterpretation, establishing that icons alone are insufficient to communicate intent reliably. The PrivacyDeck incorporates both insights: each physical control is paired with a widely recognised icon *and* a printed text label, ensuring interpretability without prior familiarity with the device.

2.5 Summary

Across these four areas, a consistent gap emerges. Software dashboards provide transparency but not low effort action. Tangible privacy frameworks offer strong design principles but lack implementation for desktop environments. Physical prototypes demonstrate feasibility but remain scoped to IoT contexts. Icon research identifies effective symbols but cautions against deploying them without accompanying text. The PrivacyDeck integrates these contributions into a single device that consolidates fragmented controls, makes system state continuously visible through hardware feedback, and reduces each protective action to a physical gesture requiring no software navigation.

3 PrivacyDeck: Concept

PrivacyDeck is built around the core idea of tangible, low-friction privacy management through persistent physical controls and unambiguous ambient feedback. High-frequency privacy actions, such as locking the operating system, muting the microphone, blocking the camera, isolating the network, or clearing sensitive data, are mapped to dedicated physical buttons and toggles on the control panel. This externalisation eliminates the need to navigate software menus, enabling single-gesture activation that builds muscle memory for consistent use in any desktop context.

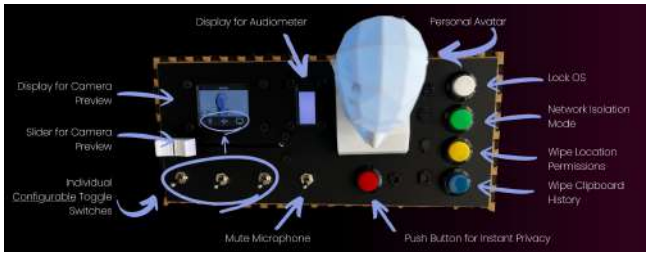


Figure 2: All components of PrivacyDeck and their respective functions.

The feedback mechanism is designed to be glanceable and non-intrusive, following calm technology principles [21]. The centerpiece is the modular, customisable avatar, which serves as both a personal companion and an ambient display using a NeoPixel LED ring and optical fibers. Key features include:

- **Camera status:** The avatar's eyes light up in red when the webcam is active.
- **Microphone status:** The mouth illuminates in red when the microphone is recording.
- **Contextual LED cues:** Remaining LEDs illuminate in context-specific colours synchronised with the corresponding privacy buttons to signal pending actions and prompt the user. For example:
  - Blue when clipboard history exceeds 5 entries (prompting clear)
  - Green when browser tracking or cookies surpass a threshold (prompting protection)
- **Optimal privacy state:** When all critical sensors are deactivated, protections are enabled, and data is cleared, the avatar and button LEDs remain unlit, providing a distraction-free confirmation of full protection.
- **Instant privacy button:** Pressing this triggers all necessary actions in sequence, with the avatar briefly flashing all LEDs red as visual confirmation before all lights extinguish upon completion.

This hardware-centric feedback is complemented by the software dashboard, which provides:

- A composite privacy score
- Real-time monitoring (camera preview, audio meter)
- Customisation options for the avatar's appearance
- An explainable Privacy Advisor offering contextual recommendations without interrupting workflow

By combining direct physical control with peripheral LED-based awareness, PrivacyDeck addresses privacy fatigue by making protective actions effortless and the system state intuitively perceptible at all times.

4 Implementation

As a foundation for the implementation, a true-to-scale Figma prototype enabled iterative testing of element sizes and positions as

shown in Figure 3. To evaluate ergonomics, proportions, and component placement, 3D paper prototypes were developed. In general, controls are organised across the deck into coherent spatial zones with consistent component types such as push buttons, toggles, and displays to support usability, findability, and reduced cognitive load in line with Muhander et al. [13].

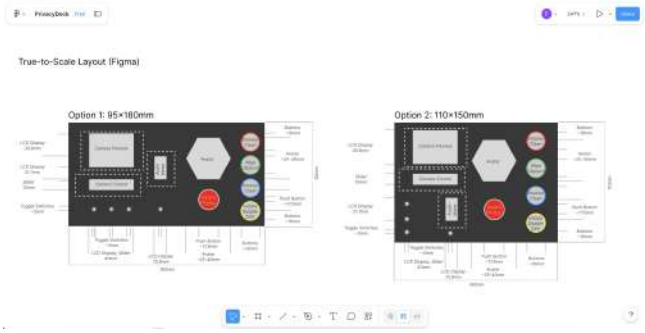


Figure 3: True-to-scale Figma screens for two potential layouts.

4.1 First Vertical Prototype

Before implementing the full system architecture, we developed an early vertical prototype to validate core end-to-end functionality. This prototype supported only two essential actions: locking the operating system and muting the microphone. The goal was not feature completeness, but architectural validation—testing the complete signal chain from physical input on the microcontroller, through serial communication, to OS-level execution and state feedback. By confirming reliable, low-latency system-level triggers early, we reduced integration risks before expanding the hardware and software stack.

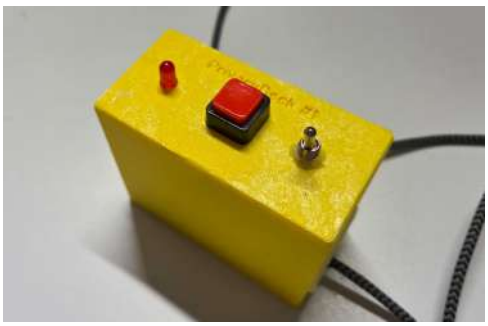


Figure 4: Early vertical prototype supporting OS lock and microphone mute functionality.

4.2 Hardware

4.2.1 Hardware Implementation. The hardware development process followed an iterative prototyping approach, beginning with the systematic testing of each individual component using dedicated test scripts. All electronic parts—including buttons, switches, LEDs, the microcontroller, and the NeoPixel elements were validated independently to ensure stable operation before integration. In parallel, all physical components were precisely measured to enable an accurate enclosure design.

The housing (Figure 5) was laser-cut from a parametric box template adapted from https://boxes.hackerspace-bamberg.de and refined over two iterations to match the PrivacyDeck dimensions. The top deck plate (Figure 6) was designed from scratch in Shapr3D (https://www.shapr3d.com/) and iteratively adjusted to ensure stable, ergonomic placement of all controls.

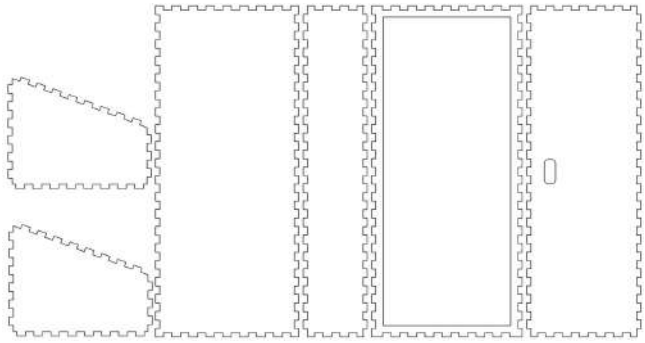


Figure 5: Laser-cut enclosure based on an adapted parametric box template.

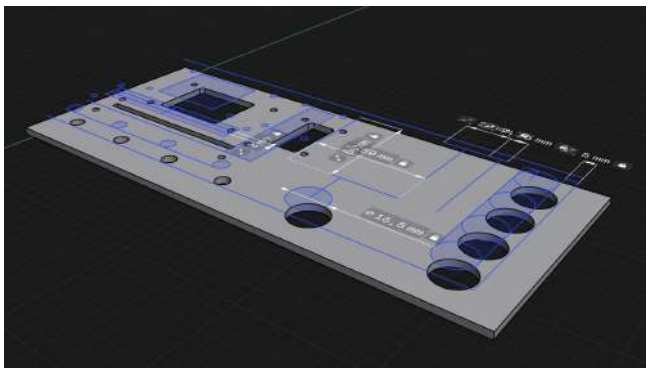


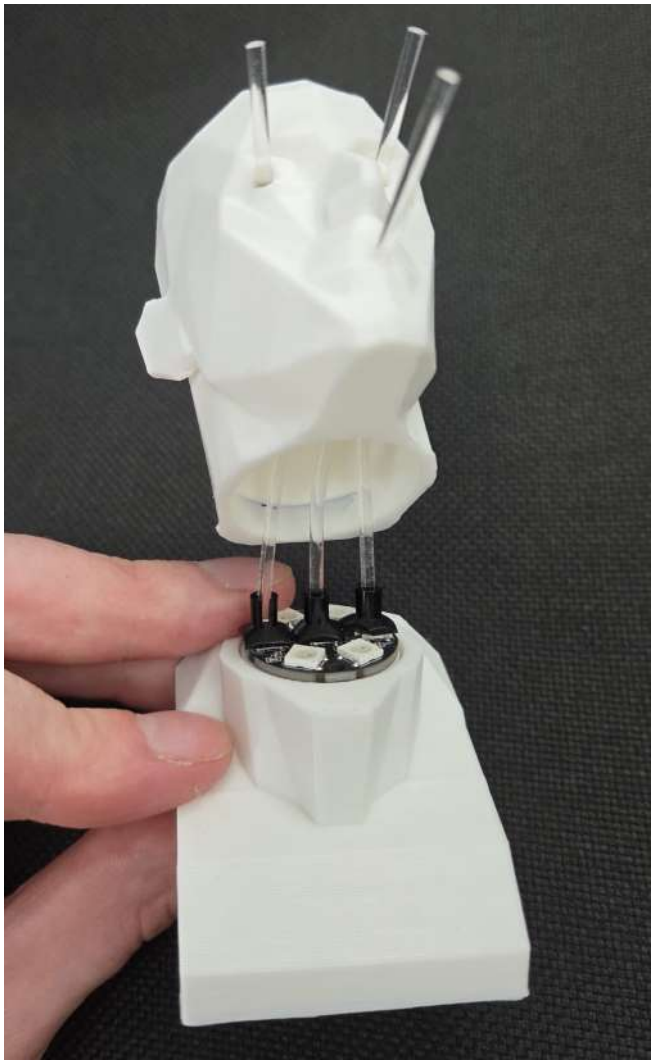
Figure 6: Custom-designed deck plate holding the individual hardware components.

After finalising the enclosure geometry, all components were installed and soldered, resulting in a compact and mechanically stable assembly.

4.2.2 Avatar Design and Fabrication. The avatar component represented the most technically demanding aspect of the hardware implementation. Rather than relying on an existing model, we designed a custom 3D model from scratch to meet both aesthetic and functional requirements. To enable internal hardware integration, the model was deliberately divided into three separate 3D-printable subcomponents: a base structure, a mid-section housing the electronics, and a detachable head element. This modular design allowed wiring and component placement prior to final assembly.

(a) The avatar divided into three 3D-printable components for modular assembly.

(b) Magnetic pogo-pin interface enabling snap-on avatar attachment.



(c) NeoPixel ring and optical fibre routing to the avatar's eyes and mouth.

Figure 7: Hardware construction and integration of the modular avatar component.

The avatar underwent three print iterations to optimise structural stability, internal spacing, and surface finish. Each avatar contains a NeoPixel ring with seven RGB LEDs. Three of these LEDs are optically routed to the avatar's eyes and mouth using optical fibres. The fibres channel light precisely from the LEDs to the eyes and mouth, in order to convey when the webcam is on (eyes) or the microphone is recording audio (mouth).

Integrating the optical fibres required substantial manual refinement. The fibres were fixed directly above selected LEDs and shaped using controlled heat application to achieve precise alignment with the eye and mouth openings. Covering each fibre with opaque tape would have reduced light scatter to other parts of the avatar.

4.2.3 Avatar Mounting and Electrical Connection. To allow modular attachment, the avatar connects to the PrivacyDeck using a magnetic pogo-pin interface. Two 3-pin pogo connectors were integrated: one part embedded in the deck and the corresponding counterpart installed inside the avatar base. Magnets integrated into the pogo-pin housings provide automatic alignment and mechanical stability when the avatar is placed onto the deck.

4.2.4 Avatar Identification Mechanism. To differentiate between two physical avatar variants, we implemented a hardware-based identification mechanism using distinct resistor configurations. Each avatar contains a predefined resistor network connected to an analog-to-digital converter (ADC) pin on the Raspberry Pi Pico microcontroller. Avatar 1 integrates two 20 kΩ resistors, while Avatar 2 uses one 20 kΩ and one 51 kΩ resistor. These differing resistor combinations result in distinct measurable voltage levels at the ADC input due to variations in current flow characteristics.

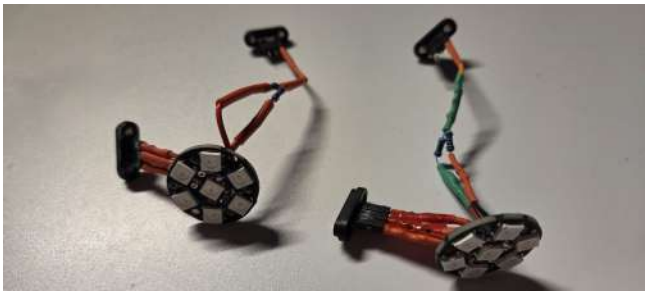


Figure 8: Internal hardware layout of both avatar variants, including resistor configurations.

By reading the voltage value, the microcontroller can reliably determine the connected avatar. This enables automatic avatar recognition without additional digital communication interfaces, reducing wiring complexity while ensuring reliable identification.

4.3 Software

The GitHub repository includes all software and hardware code, along with all assets: https://github.com/simonroedig/PrivacyDeck

Figure 9: Overview of the PrivacyDeck architecture.

4.3.1 OS Daemon. The operating system integration of the PrivacyDeck is handled by a lightweight Python-based daemon running on the host machine. Communication with the Raspberry Pi Pico microcontroller occurs via a serial connection over USB. The microcontroller transmits predefined command signals (e.g., `LOCK_SYSTEM`) which are parsed and executed by the daemon. This serial communication approach proved reliable and responsive for triggering discrete privacy-related actions such as locking the system, muting the microphone, or toggling connectivity settings.

While serial transmission was sufficient for control signals, bandwidth limitations appeared when integrating two SPI-connected displays for live webcam and microphone activity. To offload continuous data transfer, we implemented an alternative communication path using WebSockets over a local network connection, enabling stable real-time streaming for both webcam preview and audio visualisation in controlled network environments.

The daemon was tested across Windows, Linux, and partially macOS. Linux provided the most flexible integration, as privacy-relevant system functions can be executed directly via shell scripts with appropriate permissions; the final demonstration was therefore conducted on Linux. Windows presented the greatest challenges: several privacy functions, such as enabling airplane mode, are not accessible through standard Python libraries or publicly exposed system APIs, requiring graphical automation workarounds that simulate keyboard navigation through settings menus. This approach increases fragility due to UI dependency and makes reliable state introspection difficult, as the system can often only toggle a setting without robustly querying its current state.

These cross-platform inconsistencies highlight a broader infrastructural limitation: the absence of a unified, open, operating-system-level API for privacy and security controls. Developing such a standardised framework would significantly lower the barrier for future tangible privacy interfaces, enabling reliable, cross-platform access to system-level privacy functions without resorting to UI automation or platform-specific workarounds.

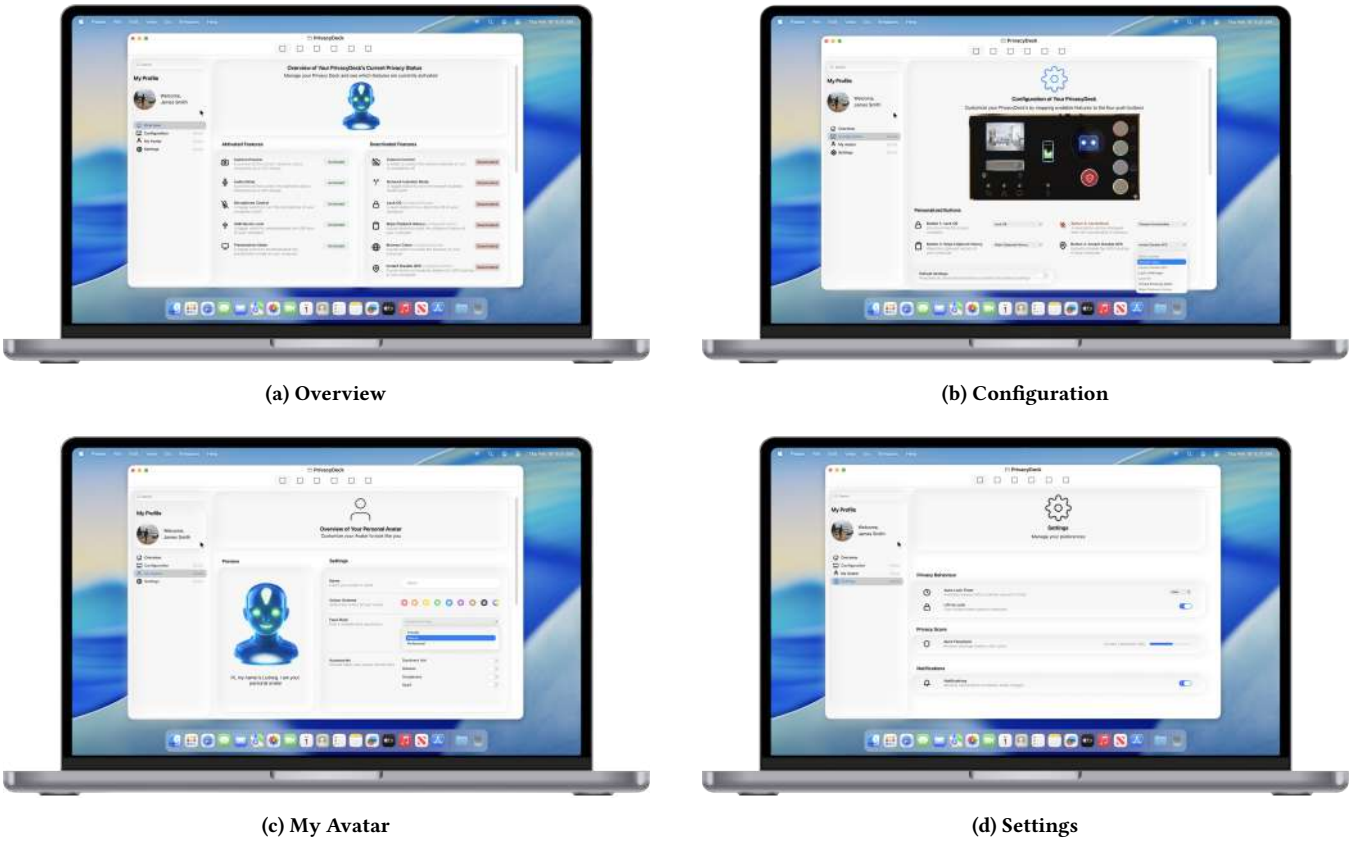
4.3.2 GUI Implementation. The companion desktop dashboard was developed as a standalone software to the physical deck. Its architecture anticipates a three-layer design: a firmware layer (currently the Raspberry Pi Pico communicating over serial USB, with a planned migration to an ESP32 forwarding events over TCP) that captures button events; the Python daemon that executes OS-level privacy controls and maintains the canonical feature state; and the React frontend that provides visualisation, interaction, and configuration. State is managed through shared React contexts (`PrivacyContext` and `AvatarContext`), providing a single source of truth; a dedicated WebSocket hook handles the connection lifecycle, keeping rendering concerns separated from daemon communication.

Initial design and iterations. As a first step, several Figma screens were developed and iteratively refined with the goal of maintaining simplicity and intuitiveness by following the macOS layout conventions. The process resulted in four final screens as the basis for the accompanying desktop dashboard as shown in Figure 10.

Privacy score and state feedback. As illustrated in Figure 11, privacy functions are organised into *exposure* controls (camera and microphone), *protection* controls (network isolation, GPS, USB lock, clipboard guard, and browser cleaning), and *monitoring* controls (audio meter and camera preview). A composite privacy score is computed in real time, weighted at 40 % for exposure safety and 60 % for protection coverage. This reflects a deliberate design heuristic: active sensors represent the most immediate vector for data leakage, whereas connectivity and data controls govern the potential blast radius of any failure. The weights are not empirically derived; they constitute an initial parameterisation open to future validation through user studies.

Rather than presenting the score as an opaque scalar, the interface exposes a live attribution panel that identifies which control was toggled and whether the transition improved or worsened privacy posture. This is motivated by Lim et al. [11], who showed that causal *why* and *why-not* explanations improve user understanding of context-aware systems compared with state-only feedback. An *Instant Privacy* button toggles every feature to its maximum protective state simultanously in a single gesture, embodying Shneiderman's principle of rapid operations on visible objects [17].

Privacy Advisor: context-aware adaptive guidance. The score and attribution panel tell users *where they stand*, but not *what to do next*. The Privacy Advisor, visible across all three panels of Figure 11, closes this awareness-to-action gap [1] by delivering context-sensitive, single-sentence recommendations in real time.



(a) Overview

(b) Configuration

(c) My Avatar

(d) Settings

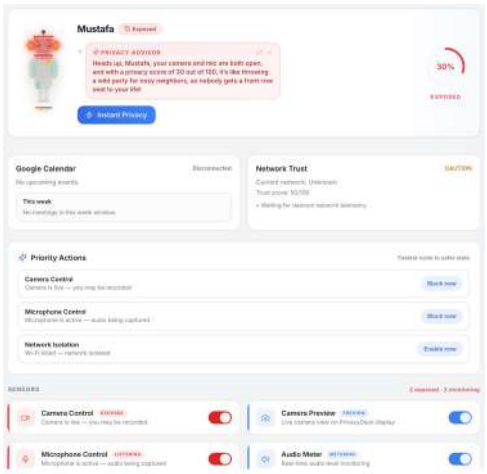
Figure 10: Final Figma screens for subpages.

Three HCI principles shape its design. First, following Horvitz's mixed-initiative framework [10], recommendations are presented as dismissable, refreshable suggestions rather than modal interruptions, preserving user control over attention allocation. Second, drawing on digital nudge theory [19, 20], the advisor surfaces the single most actionable risk at the moment a toggle changes, framing consequences in concrete terms rather than abstract threat levels. Third, to ensure glanceability under divided attention, the advisor surfaces exactly one recommendation per state change, colour-coded to match the current risk tier.

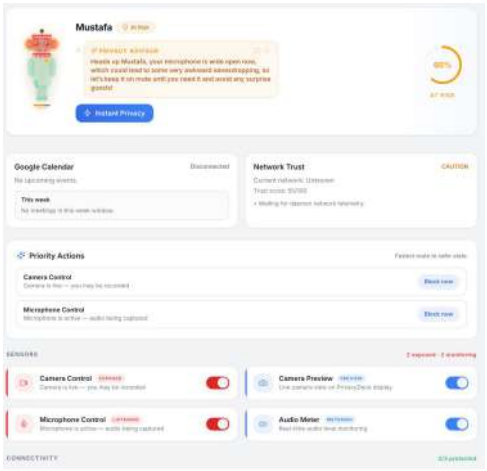
Critically, the advisor is not a thin wrapper around a language model. A four-stage local pipeline executes entirely in the browser before any external call is made. First, a behavioural pattern analysis flags *habitual risks*, features remaining in a risky state in at least 60 % of their last 20 recorded appearances, and computes a score trend via split-half comparison, following the just-in-time adaptive intervention principle that historical behaviour should inform the timing and content of guidance [14]. Second, a state-diffing function identifies the most recently changed feature within a three-second recency window and classifies the action as SECURED or EXPOSED. Third, a builder function assembles all upstream signals as current score, toggle classification, network trust status, habitual patterns, and score trend, into a single priority-ranked payload encoding domain-specific information triage. Only at the final stage is this compiled context forwarded to OpenAI's Chat Completions API[1] (model `gpt-4o-mini`, 75-token cap), where a constrained system prompt renders the pre-decided recommendation as a single personalised sentence. The language model thus functions as a *text rendering layer* for decisions already made upstream. The orchestrating hook adds debouncing (1 400 ms), state-key deduplication, race-condition guards, and graceful degradation to locally generated fallback messages when the API is unreachable.

Avatar customisation and peripheral feedback. Figure 12 shows the customisation panel through which users configure body shape, colour scheme, face style, and accessories. The configured avatar persists across all dashboard states; its eye and mouth illumination update continuously to reflect the current privacy score, keeping status visible in the periphery without demanding active inspection—consistent with Weiser and Brown's calm-technology vision [21]. By granting users authorship over the avatar's appearance, the system creates a personally resonant reference point for privacy posture. Birk et al. [4] demonstrated that avatar customisation strengthens identification and intrinsic motivation, directly motivating this design choice. Gamification signals such as vitality cues further reinforce protective behaviour, calibrated so that a fully-protected indication appears only when all controls are verifiably

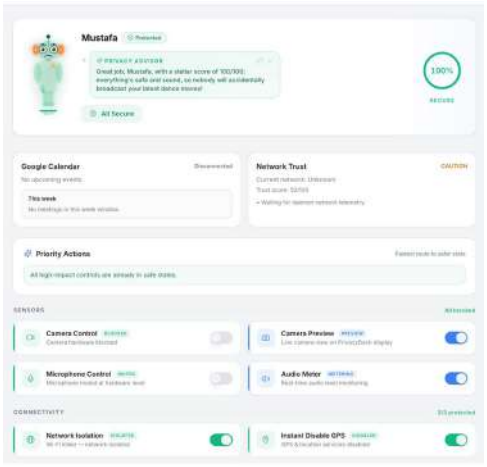
[1] https://platform.openai.com/docs/api-reference/chat



(a) camera and microphone active, score 30, Exposed.



(b) microphone active, camera blocked, score 60, At Risk.



(c) all sensors blocked and all protections active, score 100, Secure. Avatar expression, score-ring colour, and Privacy Advisor tone update in real time.

Figure 11: The PrivacyDeck dashboard across three privacy states.

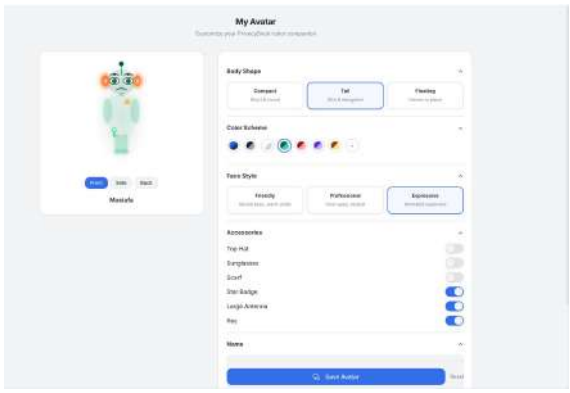


Figure 12: Avatar customisation panel of body shape, colour, face style, and accessories across all dashboard states.

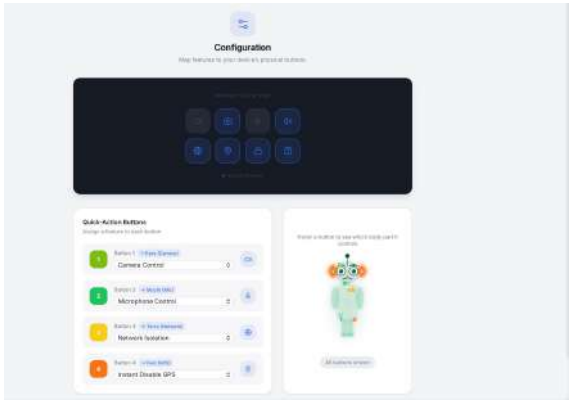


Figure 13: Configuration view for assigning privacy functions to physical buttons.

safe, consistent with Fogg's [8] principle that persuasive systems must not mislead users about consequences.

Button-mapping configuration. Figure 13 shows the view through which users assign privacy functions to each of the eight physical button slots. Hovering over a slot in the device preview highlights the corresponding avatar body part, reinforcing the physical-to-digital mapping before the user commits. All personalisation data (avatar appearance, button mappings) is stored locally, ensuring that no privacy-relevant configuration leaves the user's machine.

5 Discussion

The PrivacyDeck illustrates a tangible approach to addressing the core barriers identified in prior work: fragmented controls that induce privacy fatigue [16], absent feedback that degrades situation awareness under cognitive load [6], and ambiguous device states that erode user trust [2]. By mapping each protective action to a dedicated physical control, the system delivers immediate, low-latency responses that bypass the software navigation overhead Alashwali et al. [3] identified as a primary deterrent to consistent privacy behaviour. Simultaneously, the avatar and hardware indicators maintain continuous peripheral awareness, consistent with

Weiser and Brown's calm-technology vision [21], addressing the attentional limitations that Portnoff et al. [15] showed cause mo even existing webcam LEDs to go unnoticed in 95 % of cases when user attention is directed elsewhere. Our modular avatar extends prior tangible prototypes such as PriviFy [13] beyond IoT configuration into desktop privacy management, integrating real-time sensor state into a personalisable ambient display that leverages identification effects observed by Birk et al. [4].

Several limitations bound our current contribution. Cross- platform testing indicated reliable operation on Linux but exposed significant limitations on Windows, where the absence of publicly exposed privacy APIs necessitated brittle UI automation. This underscores a broader infrastructural gap: without a unified, open framework for privacy controls, tangible privacy interfaces remain constrained by platform-specific affordances. We propose a secure, standardized Open Privacy Control API to abstract OS-level privacy functions into a unified, cross-platform interface. This is challenging, because it needs consensus among proprietary OS vendors, but we believe defining such an open standard is a necessary prerequisite for the future development of scalable, trusted, and low-friction physical privacy tools. Furthermore the prototype has not been evaluated with end users; quantitative impacts on privacy fatigue and behavioural consistency remain open for controlled studies. The privacy score weighting (40 % exposure, 60 % protection) is a design heuristic, not an empirically calibrated metric. The Instant Privacy button currently prioritises speed over reversibility; future iterations should incorporate a brief undo window or confirmation gate to prevent disruptive accidental activation during live sessions such as video calls. Finally, the Privacy Advisor's reliance on an external language model introduces a latency and availability dependency; moreover, forwarding compiled recommendation context to an external provider represents a privacy tradeoff that future iterations should resolve through fully local generation.

6 Conclusion

PrivacyDeck provides a tangible solution that externalises high-frequency privacy actions into persistent physical controls with real-time, interpretable feedback. By reducing interaction friction and supporting situation awareness [6], the device aims to help users protect their desktop environments more consistently and confidently. This work underscores the value of physical computing for usable security and privacy, and advocates for an open, cross-platform API to support seamless integration of system-level privacy tools. Future iterations should incorporate formal user evaluations, expand hardware modularity, and explore adaptive feedback mechanisms that respond to individual risk profiles over time.

Acknowledgments

The authors used large language models (ChatGPT, Grok) as editorial tools during the revision process. All research design, system implementation, analysis, and argumentation are entirely the authors' own work.

References

[1] Alessandro Acquisti, Idris Adjerid, Rebecca Balebako, Laura Brandimarte, Lorrie Faith Cranor, Saranga Komanduri, Pedro Giovanni Leon, Norman Sadeh, Florian Schaub, Manya Sleeper, et al. 2017. Nudges for privacy and security: Understanding and assisting users' choices online. *ACM Computing Surveys (CSUR)* 50, 3 (2017), 1–41.

[2] Imtiaz Ahmad, Rosta Farzan, Apu Kapadia, and Adam J. Lee. 2020. Tangible Privacy: Towards User-Centric Sensor Designs for Bystander Privacy. *Proc. ACM Hum.-Comput. Interact.* 4, CSCW2, Article 116 (Oct. 2020), 28 pages. doi:10.1145/3415187

[3] Eman Alashwali, Joanne Peca, Mandy Lanyon, and Lorrie Faith Cranor. 2025. Work from home and privacy challenges: what do workers face and what are they doing about it? *Journal of Cybersecurity* 11, 1 (2025), tyaf010. doi:10.1093/cybsec/tyaf010

[4] Max V. Birk, Cheralyn Atkins, Jamie T. Bowey, and Regan L. Mandryk. 2016. Fostering Intrinsic Motivation through Avatar Identification in Digital Games. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '16)*. Association for Computing Machinery, 2982–2995. doi:10.1145/2858036.2858062

[5] Sarah Delgado Rodriguez, Anh Dao Phuong, Franziska Bumiller, Lukas Mecke, Felix Dietz, Florian Alt, and Mariam Hassib. 2023. Padlock, the Universal Security Symbol? - Exploring Symbols and Metaphors for Privacy and Security. In *Proceedings of the 22nd International Conference on Mobile and Ubiquitous Multimedia* (Vienna, Austria) *(MUM '23)*. Association for Computing Machinery, New York, NY, USA, 10–24. doi:10.1145/3626705.3627770

[6] Mica R Endsley. 1995. Toward a Theory of Situation Awareness in Dynamic Systems. *Human Factors: The Journal of the Human Factors and Ergonomics Society* 37, 1 (1995), 32–64.

[7] Florian M. Farke, David G. Balash, Maximilian Golla, Markus Dürmuth, and Adam J. Aviv. 2021. Are Privacy Dashboards Good for End Users? Evaluating User Perceptions and Reactions to Google's My Activity. In *30th USENIX Security Symposium (USENIX Security 21)*. USENIX Association, 483–500. https://www.usenix.org/conference/usenixsecurity21/presentation/farke

[8] Brian J. Fogg. 2003. *Persuasive Technology: Using Computers to Change What We Think and Do*. Morgan Kaufmann, San Francisco, CA.

[9] Hana Habib, Yixin Zou, Yaxing Yao, Alessandro Acquisti, Lorrie Cranor, Joel Reidenberg, Norman Sadeh, and Florian Schaub. 2021. Toggles, Dollar Signs, and Triangles: How to (In)Effectively Convey Privacy Choices with Icons and Link Texts. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (Yokohama, Japan) *(CHI '21)*. Association for Computing Machinery, New York, NY, USA, Article 63, 25 pages. doi:10.1145/3411764.3445387

[10] Eric Horvitz. 1999. Principles of mixed-initiative user interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Pittsburgh, Pennsylvania, USA) *(CHI '99)*. Association for Computing Machinery, New York, NY, USA, 159–166. doi:10.1145/302979.303030

[11] Brian Y. Lim, Anind K. Dey, and Daniel Avrahami. 2009. Why and Why Not Explanations Improve the Intelligibility of Context-Aware Intelligent Systems. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '09)*. Association for Computing Machinery, 2119–2128. doi:10.1145/1518701.1519023

[12] Vikram Mehta, Daniel Gooch, Arosha Bandara, Blaine Price, and Bashar Nuseibeh. 2021. Privacy Care: A Tangible Interaction Framework for Privacy Management. *ACM Trans. Internet Technol.* 21, 1, Article 25 (2021). doi:10.1145/3430506

[13] Bayan Al Muhander, Omer Rana, and Charith Perera. 2024. PriviFy: Designing Tangible Interfaces for Configuring IoT Privacy Preferences. arXiv:2406.05459 [cs.CR] https://arxiv.org/abs/2406.05459

[14] Inbal Nahum-Shani, Shawna N Smith, Bonnie J Spring, Linda M Collins, Katie Witkiewitz, Ambuj Tewari, and Susan A Murphy. 2016. Just-in-time adaptive interventions (JITAIs) in mobile health: key components and design principles for ongoing health behavior support. *Annals of behavioral medicine* (2016), 1–17.

[15] Rebecca S. Portnoff, Linda N. Lee, Serge Egelman, Pratyush Mishra, Derek Leung, and David Wagner. 2015. Somebody's Watching Me? Assessing the Effectiveness of Webcam Indicator Lights. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems* (Seoul, Republic of Korea) *(CHI '15)*. Association for Computing Machinery, New York, NY, USA, 1649–1658. doi:10.1145/2702123.2702164

[16] Han Shao, Xiang Li, and Guodi Wang. 2022. Are You Tired? I am: Trying to Understand Privacy Fatigue of Social Media Users. In *Extended Abstracts of the 2022 CHI Conference on Human Factors in Computing Systems* (New Orleans, LA, USA) *(CHI EA '22)*. Association for Computing Machinery, New York, NY, USA, Article 378, 7 pages. doi:10.1145/3491101.3519778

[17] Ben Shneiderman. 1983. Direct Manipulation: A Step Beyond Programming Languages. *Computer* 16, 8 (1983), 57–69. doi:10.1109/MC.1983.1654471

[18] Yunpeng Song, Cori Faklaris, Zhongmin Cai, Jason I. Hong, and Laura Dabbish. 2019. Normal and Easy: Account Sharing Practices in the Workplace. *Proc. ACM Hum.-Comput. Interact.* 3, CSCW, Article 83 (Nov. 2019), 25 pages. doi:10.1145/3359185

[19] Richard H. Thaler and Cass R. Sunstein. 2008. *Nudge: Improving Decisions About Health, Wealth, and Happiness*. Yale University Press, New Haven, CT.

[20] Markus Weinmann, Christoph Schneider, and Jan vom Brocke. 2016. Digital nudging. *Business & Information Systems Engineering* 58, 6 (2016), 433–436.

[21] Mark Weiser and John Seely Brown. 1996. Designing Calm Technology. *PowerGrid Journal* 1, 1 (1996), 75–85.

Character Count

This work contains 29393 characters (without spaces).