

Index

1.	From Distributed Networks to Cloud Centralization	8
2.	The Problems of Centralization and the P2P Alternative	24
3.	Case Studies	36
4.	Methodology	48
5.	Project	64
6.	Product Communication	120
7.	Conclusion	142

Introduction

In the last thirty years the digital world has drastically changed. The internet, at the start was an open and horizontal network where computers talk directly one to another, but after some time it started to transform into a system dominated by a few large platforms. The web started indeed as a decentralized frontier where data and infrastructure were distributed and not in the hands of few. However, when we moved into the "Cloud Era," this decentralized structure was replaced by the **Software as a Service (SaaS)** model. Basically we exchanged the independence of the early web for the extreme convenience of modern apps. This undoubtedly created digital tools that were easier to use, but created a big problem: all of our data is under the control of massive corporations.

We entered inside an "**Hub-and-Spoke**" architecture, caused by the centralization created by the presence of the "Cloud" itself, that makes this intermediary indispensable. This configuration creates inevitably a critical paradox: while service appears more reliable and strong, they are in reality extremely fragile due to the **Single Point of Failure** principle. (Wu, D. D., 2021) When a central provider experiences a technical or security compromise, the ripple effects are felt globally, affecting millions of users and businesses simultaneously. Furthermore, this model has fundamentally changed how the users and their digital assets interact with each other. In the current cloud-centric habitat, individuals don't "own" their data; they just have access to it, surrendering their **digital sovereignty** in exchange for the convenience of managed services.

The Cloud has both environmental and economic costs that are really significant, yet often obscured by the metaphor of convenience. The presence of these massive, energy-intensive data centers are required to sustain the whole cloud architecture generates huge ecological externalities, and this is just the beginning. Other than the ecological externalities another element to consider is economic dependence, in fact we rely on a few infrastructural gatekeepers that creates a "rent-seeking" economy that stifles local-first innovation and creates hidden costs to the final user. This thesis claims that the current state of digital centralization is not something always a technical necessity, but a design choice to prioritize intermediary control over individual agency. By identifying these systemic frictions, the research argues for a return to distributed architectures, at least for certain valuable use, exploring how **Peer-to-Peer (P2P)** technologies can be redesigned to restore the balance between high-fidelity usability and the non-negotiable right to data sovereignty.



Chapter 1

From Distributed Networks to Cloud Centralization

1.2 The Evolution of Distributed Architectures: A Historical Perspective

The history of Peer-to-Peer (P2P) computing is not a sudden technological outburst but a **gradual evolution of protocols and philosophies** that began in the late 20th century. To understand correctly how we arrived in the current state of decentralized communication, it is necessary to trace the milestones of the P2P technologies. (Felipe, 2018)

1.2.1 The Foundations: ARPANET and Usenet

ARPANET (Advanced Research Projects Agency Network) was the technical groundwork for P2P created in the late 1960s and 70s. It was used to make computers communicate between each other, and was the first stone to the concept of a networked society. In 1979, this led to the **creation of Usenet**, a decentralized discussion system. Usenet operated through “newsgroups,” where content and files were distributed across various servers without a single governing body. While it became a hub for early digital communities, its unmoderated nature and anonymity eventually created controversies regarding copyrighted material and illicit content, giving a hint of the legal and ethical tensions that would define the future of distributed networks and P2P technologies.

1.2.2 Napster and the Vulnerability of the Center

Napster made the P2P model something **mainstream**, it was launched in 1999 and offered a **seamless interface** to share MP3 files. Thanks to this Napster registered approximately 80 million users at its peak, creating a huge collective library of digital audio. Unfortunately, **Napster was a “hybrid” P2P system**: the file transfers itself were direct between users but to facilitate searches the network relied on centralized indexers. This architectural choice proved to be a pain point for the network itself. Because the indexers were hosted on central servers, the service could be, and eventually was, targeted by legal action from the RIAA. Following a landmark court battle over copyright infringement, Napster was forced to shut down in 2001, **proving that any centralized link in a distributed chain is a potential point of total failure**.

1.2.3 The Rise of Resilient Networks: Gnutella, Freenet, and BitTorrent

The shutdown of Napster made the people create some true decentralized platforms. Gnutella developers abandoned the central indexer model in favor of query flooding, a method where search requests are passed from node to node. This model inevitably sacrificed raw speed, but created an **environment immune to shutdowns**, as there was no more a central entity to target. Together with Gnutella, Freenet emerged with a

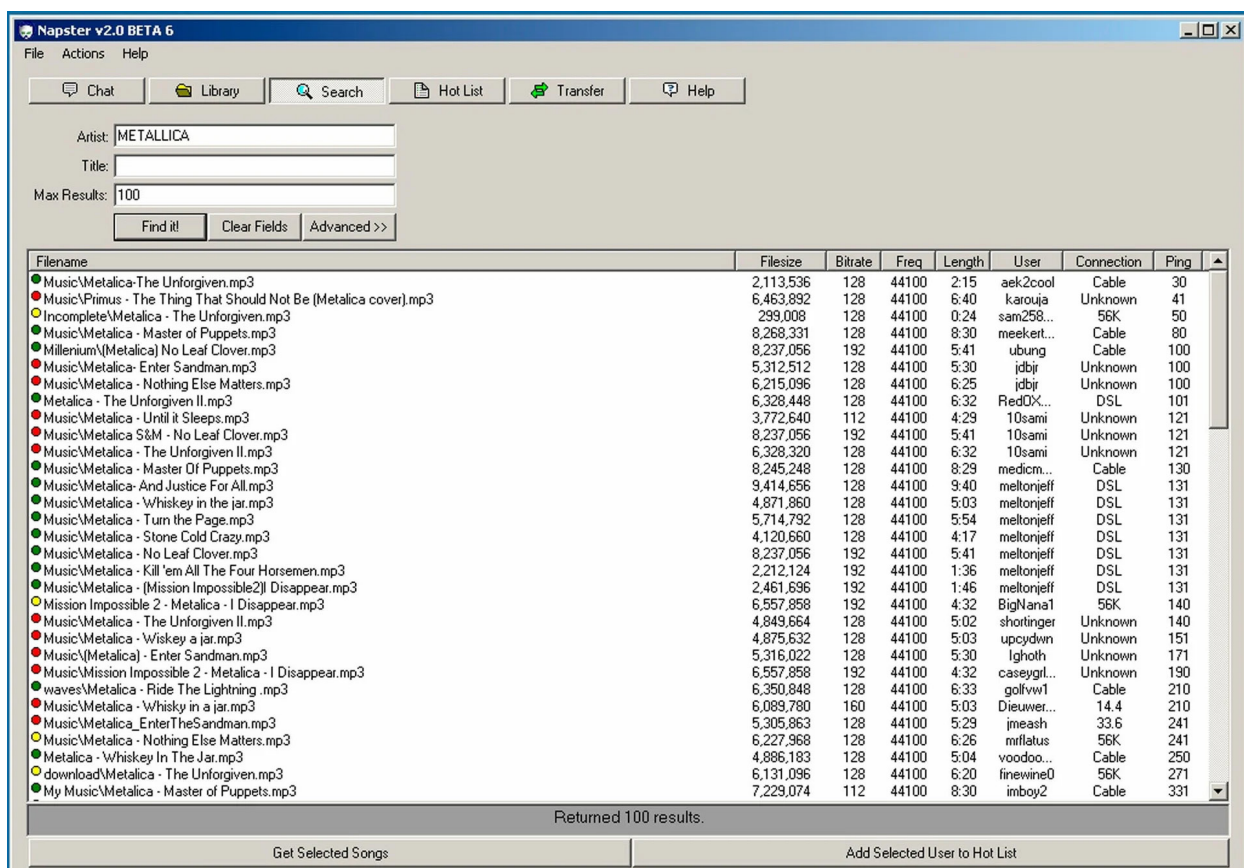
radical focus on anonymity and censorship resistance, utilizing encrypted data blocks stored across participants' hard drives that protected the users and the files.

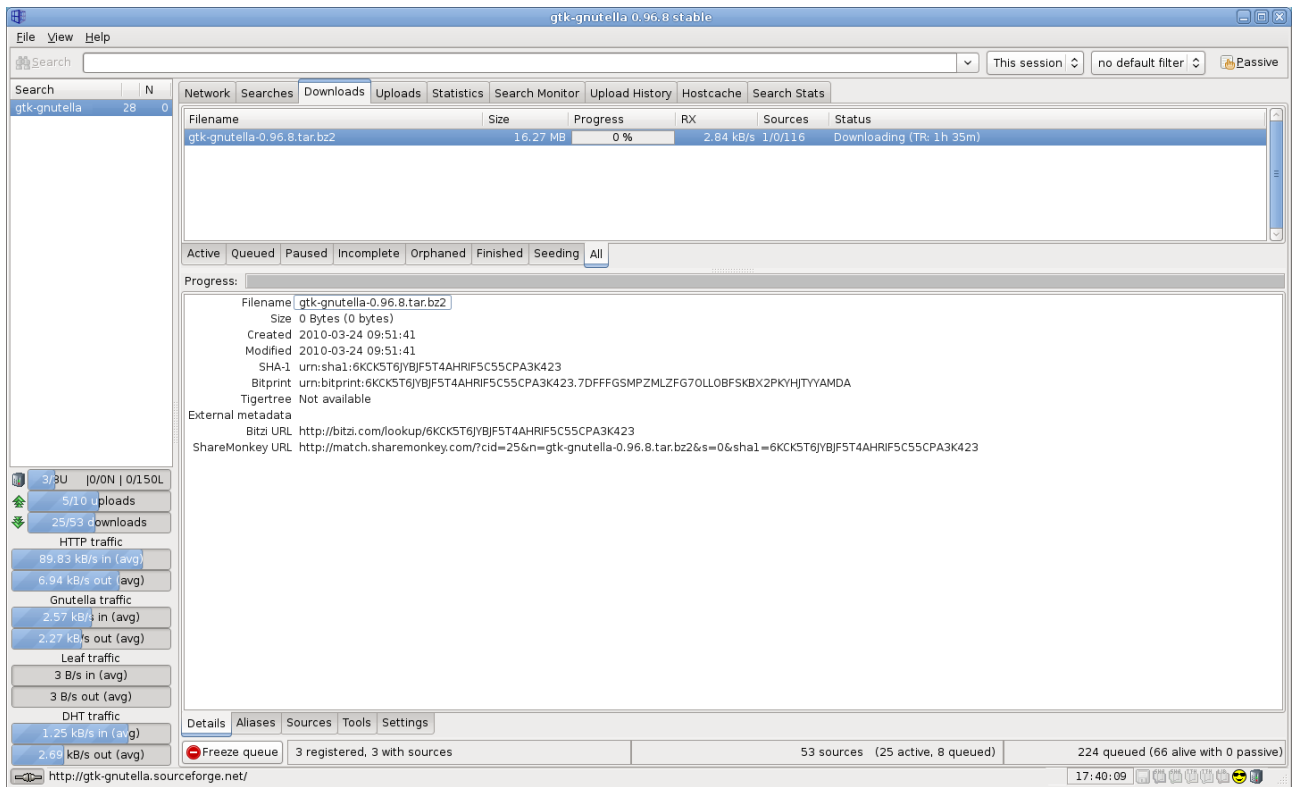
Bram Cohen created BitTorrent in the year 2001, initially BitTorrent versions still relied on central "trackers" to coordinate peers, but, after some time the protocol eventually migrated to a trackerless system through the implementation of a **Distributed Hash Table (DHT)**. The DHT allowed for peer discovery without any central intermediary, effectively creating a self-sustaining network for every shared file. (HenryR, 2008) The resilience of these systems was evidenced by the fact that, within months of Napster's closure, traffic on decentralized networks like Gnutella and BitTorrent **surpassed the highest activity levels** ever recorded by their centralized predecessor without having any shutdowns, as Napster suffered.

1.2.4 Optimizing Discovery and Communication: Kademia and Skype (2002–2003)

When P2P networks scaled to millions of users, the need for efficient node discovery became fundamental. Kademia protocol, in 2002 revolutionized this field. Kademia refined the DHT structure, providing a mathematically optimized way for nodes to locate information and each other without any central authority. It's main ability was to maintain **high performance** in massive, ever-changing networks made it the foundational protocol for most modern decentralized ecosystems.

▽ Napster UI v2.0 BETA 6





△ Gnutella UI
v0.96 Stable

▽ Bitcoin Miners



This technical maturity allowed P2P to go **beyond static file sharing**. In 2003, the launch of Skype demonstrated that distributed architectures could facilitate high-quality, real-time voice and video communication (VoIP). Skype used a decentralized architecture of “**supernodes**”, user devices that acted as local directories and, in this way Skype bypassed traditional telecommunication infrastructures. This was a crucial proof of concept: it showed that complex, low-latency services could be sustained entirely by the collective resources of a network’s participants.

1.2.5 Bitcoin and the Distributed Ledger (2009)

In 2009, the P2P landscape was further transformed by the introduction of **Bitcoin**. It wasn’t designed for file sharing, Bitcoin introduced the **Blockchain**: a distributed registry that serves a different systemic purpose than the DHT. Satoshi Nakamoto’s vision was to create an **immutable record of transactions** that could not be tampered with because it was stored and verified by every node in the network every 10 minutes. This shift proved that P2P architectures could manage not just data exchange, but “trust” and “value” without a central authority, establishing a new paradigm for digital financial sovereignty.

1.2.6 IPFS and the Birth of the Local-First Paradigm (2014)

The most recent significant shift occurred with the emergence of IPFS (InterPlanetary File System) in 2014. IPFS introduced a fundamental change in how data is addressed on the web, moving from “location-addressed” storage (IP addresses of central servers) to “content-addressed” storage (unique cryptographic hashes of the data itself). This transition is the cornerstone of the Local-First movement. (IPFS Docs., n.d.) It claims that **data should reside primarily on user devices**, ensuring that applications remain functional even without a constant internet connection. This paradigm shift directly informs the design philosophy of the Pear ecosystem, where data sovereignty is achieved by prioritizing local storage and direct peer synchronization, setting the stage for applications like PearPass.

1.3 The Paradox of Centralization: Why the Cloud Dominated the Digital Habitat

The dominance of Cloud-based architectures over decentralized alternatives is not a result of merely technical superiority, but rather the **mix of a complex ecosystem of decisions** from different fields, such as, economic incentives, corporate governance, and design strategies. While the previous section highlighted the resilience of P2P, this section analyzes the drivers that led to the current “Cloud-first” status quo.

1.3.1 The Hidden Supply Chain: From Data Collection to Brokerage Markets

The “Cloud-first” paradigm is sustained by an invisible but pervasive economic engine: the **data brokerage industry**. While users often perceive their relationship as a direct exchange with a service provider (e.g., using a “free” password manager or a social network), they are actually feeding a global supply chain where **personal information is the primary raw material**. In this ecosystem, data is not merely a byproduct of digital interaction; it is a commodity that is harvested, refined, and traded across a multi-billion dollar market.

The Role of Data Brokers: The Invisible Middlemen

Beyond the tech giants that collect data directly (First-Party Data), there exists a sprawling network of Data Brokers, entities like Acxiom, Experian, and Epsilon, whose main purpose is to **aggregate information from thousands of different and disparate sources**. (Singer, P., 2015) These brokers create a “Digital Twin” of the individual, combining browsing habits, financial records, location history, and even biometric indicators and puts all this information on sale. (University of Tennessee Office of Innovative Technologies, n.d.)

The systemic danger lies in the asymmetry of information: while the user sees a simple interface, the broker sees a predictive model. This allows for what Shoshana Zuboff defines as “Automated Architecture”, where data is used not just to monitor, but to nudge and manipulate user behavior at scale.

Real-World Scandals and the Failure of Trust

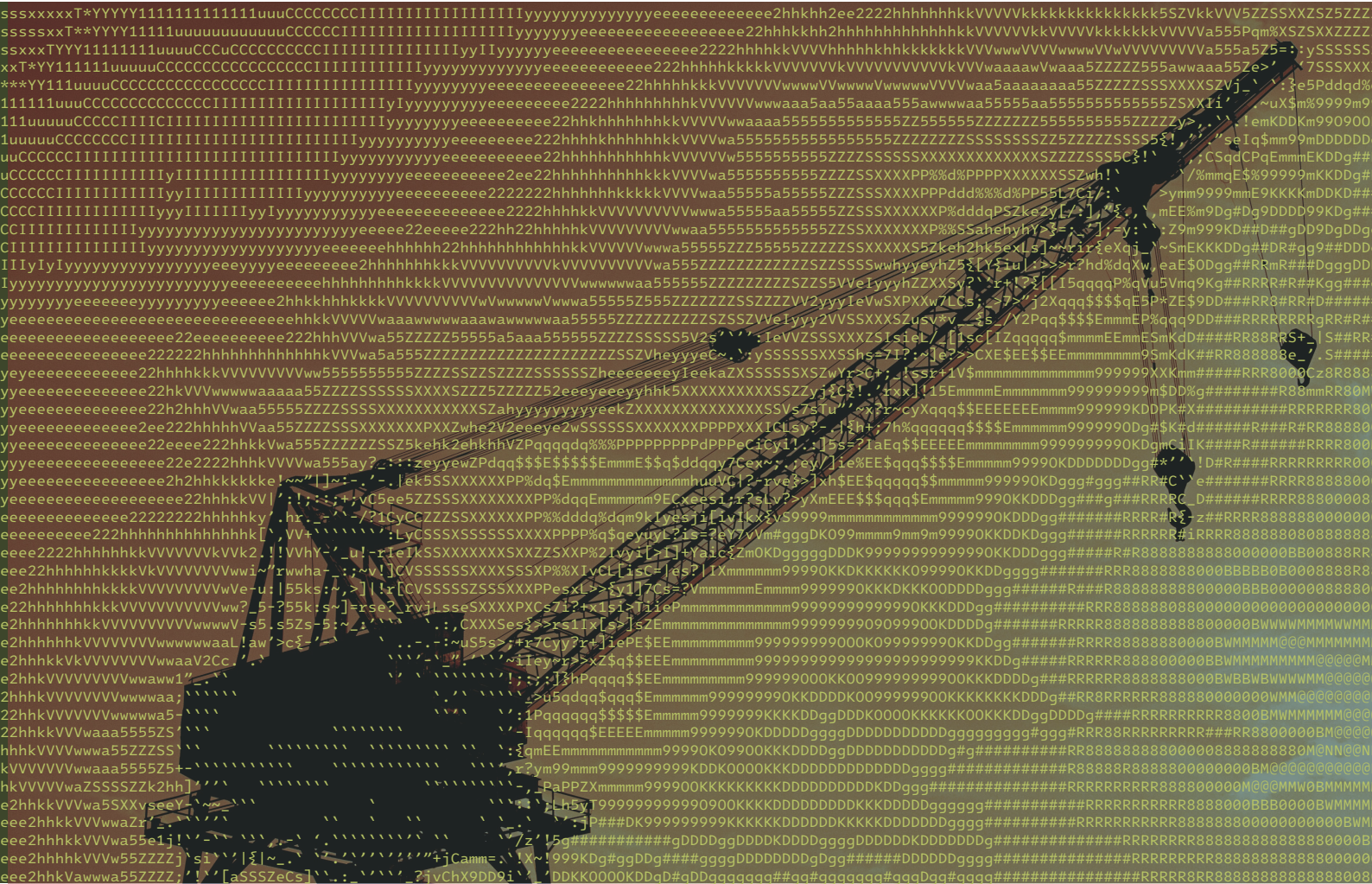
The history of centralized architectures is punctuated by scandals that reveal the true extent of this exploitation:

The first one is the Cambridge Analytica Scandal in 2008, probably this is the most known example of **data leakage** into the brokerage market. It demonstrated how big platforms such as Facebook could be used by third parties to obtain the digital shadows of millions of users without having their explicit consent, influencing through an absolutely terrible algorithmic micro-targeting.

Another particularly relevant case more related to this thesis is the Avast/Jumpshot one, where the antivirus company Avast was found out that was **selling the browsing history** of its users through one of his subsidiaries called Jumpshot. The users used a tool to get protection, while that same tool was selling and monetizing their most sensitive digital movements.

The last one is the Equifax Breach of 2017 where the data broker itself was robbed of sensitive data from over **147 million people**, this proved how even the company that should be the “custodian” of our digital identities are a single point of failure. (Framework Security., 2026)

The brokerage market represents a massive failure of the current digital metabolism. The reliance on central intermediaries has incentivized a race to the bottom in terms of privacy, where **the economic value of data outweighs the ethical necessity of security**. Transitioning to serverless P2P models effectively ‘de-commodifies’ user data: if the data is never centralized or stored, it cannot be harvested by brokers. This shift represents a fundamental redesign of the digital economy, **moving from a model of extraction to one of distributed resilience**.







Case Study: The Avast and Jumpshot Scandal and the Collapse of Digital Trust

The case of Avast and its subsidiary Jumpshot serves as a primary illustration of the “Security Paradox” within centralized digital habitats. For over a decade, Avast established itself as a global leader in cybersecurity, distributing antivirus software and **browser extensions marketed specifically to protect users from online surveillance** and invasive tracking mechanisms. However, while the front-end interface provided a reassuring aesthetic of protection, the back-end infrastructure was actively engineered for mass data extraction. Through Jumpshot, Avast systematically harvested the granular browsing histories, or clickstream data, of millions of active users. This operation captured highly sensitive digital footprints, such as precise search queries, interaction with professional networks, and visits to medical or deeply private domains. (Freedom of the Press Foundation., 2024)

The corporate justification was that all of these data were **anonymized**. Avast claimed that all data didn’t contain personally identifiable information before being monetized and sold to a vast network of corporate clients. However, regulatory investigations, made by the Federal Trade Commission, later revealed that these datasets utilized **unique device identifiers that allowed third-party buyers to easily cross-reference** and re-identify individual users. This practice exposed a profound information asymmetry: the user perceived a protective tool, while the infrastructure functioned as a surveillance mechanism.

The public exposure of these operations triggered an immediate and severe backlash from both the tech community and the general user base. Once investigative journalists exposed the scale of the **monetization scheme**, a massive user exodus occurred, driven by a collective sense of betrayal. Digital communities organized widespread boycotts on privacy forums, and major browser gatekeepers like Google, Mozilla, and Opera took the unprecedented step of removing Avast’s extensions from their platforms. This community response demonstrated that **when a security tool weaponizes user data**, the damage to the relationship between the user and the corporate is irreversible. (TechMonitor., 2020) For a project like PearPass, this scandal provides the ultimate ethical justification for a serverless architecture. By replacing “privacy by promise” with “privacy by architecture,” **P2P networks eliminate the central point of extraction**, ensuring that user trust is placed in cryptographic protocols rather than corporate integrity.

Through Jumpshot, Avast systematically harvested the granular browsing histories, or clickstream data, of millions of active users.

This operation captured highly sensitive digital footprints, such as precise search queries, interaction with professional networks, and visits to medical or deeply private domains.

1.3.2 The Designer as a Mediator of Domesticity and Dependency

Within the cloud-centric paradigm, the role of the designer has often been reduced to a **facilitator for the user**. By prioritizing “frictionless” interaction and cognitive ease, User Experience (UX) design has always obscured the physical and political reality of data infrastructures. This aesthetic of seamlessness functions as a “black box,” where the complexity of data extraction and centralized storage is hidden behind minimalist interfaces and reassuring micro-interactions.

Designers have historically favored the “illusion of simplicity,” encouraging users to delegate their digital sovereignty to central authorities in exchange for features such as automated password recovery or cross-device syncing. This approach has led to creating a passive user that doesn’t care too much about his **data ownership**. In this context, the designer is not merely an architect of interfaces but a strategic actor who, by simplifying the surface, has reinforced the systemic fragility of the core. The challenge for a P2P-based project like PearPass is to **reverse this trend**: moving from a design that pacifies the user to one that fosters awareness, transforming the “passive user” into an “active node” within a sovereign network.



1.3.3 The Subscription Economy and the Architecture of Vendor Lock-in

The structural shift toward cloud architecture was heavily accelerated by a fundamental mutation in the digital economic paradigm: the **transition from software ownership to the Software as a Service (SaaS) model**. In the early eras of computing, users purchased a lifetime license for a product, using the software and its relative data entirely on their local machines. While this model granted total control to the end-user, it limited the continuous revenue streams of technology corporations. By centralizing both application logic and data storage in the cloud, companies successfully introduced the **Subscription Economy**, transforming digital tools into a system of **permanent rent**. Central to the survival of this rent-seeking model is the sophisticated engineering of vendor lock-in, which operates as a structural barrier to digital mobility. Within a framework, vendor lock-in cannot be understood merely as a commercial inconvenience, but rather as an intentional design strategy that exploits “data gravity.” As a user accumulates credentials, files, and behavioral histories within a specific cloud repository, **the mass of this data increases**, making migration to alternative platforms logistically and technically prohibitive. (IONOS., 2023) Cloud providers deliberately design proprietary data schemas and closed environments that lack interoperability, meaning that exporting one’s digital assets often results in severe data fragmentation or total loss of metadata. The interface, therefore, makes the exporting of data complex: it facilitates the frictionless ingestion of user data while maximizing the material and computational friction of its extraction. This type of configuration effectively **paralyzes user autonomy**, ensuring that the cloud architecture maintains its monopoly not through superior utility, but through a system that makes you feel trapped.

1.4 Research Questions and Thesis Structure

Given the systemic vulnerabilities inherent in centralized cloud architectures and the contemporary erosion of digital sovereignty, this thesis investigates the operational space where distributed systems intersect with advanced user-centric design. The main ambition of this thesis is to challenge the historical idea that peer-to-peer (P2P) systems are inherently restricted to technical experts due to their architectural complexity and could not be easily used by a wider group of people. To guide this request, the research defines three fundamental Research Questions:

1. Is it possible to dismantle the convenience-sovereignty paradox, translating the complex, serverless architecture of a **P2P app into intuitive, high-fidelity user experiences?**
2. In what ways must traditional UI/UX paradigms change when transitioning from centralized, client-server models to local-first, peer-to-peer environments where concepts like data synchronization, digital identity, and recovery are entirely **managed by the user?**
3. How can the implementation of an application, specifically PearPass application, validate the efficacy of a **design-led product strategy** within decentralization-focused software engineering teams?

To respond to these questions, the thesis unfolds across six core chapters. **Chapter 1** as we read talks about the historical, economic, and industrial framework of decentralized systems and contemporary cloud monopolies. **Chapter 2** deepens the theoretical analysis of the cloud crisis, evaluating the concentration risks, the honeypot effect, and the socio-technical reasons where we find the need for a serverless P2P alternative. **Chapter 3** presents the field analysis and empirical state-of-the-art, documenting existing technical benchmarks, conducting a thorough competitor analysis, and detailing historical security vulnerabilities through localized case studies. **Chapter 4** establishes the research methodology using a Research through Design (RtD) framework. This chapter introduces a user form designed to map foundational needs and mental models, establishing the empirical data required to construct user personas. Subsequently, **Chapter 5** explores the application's lifecycle, moving from the release of version 1.0 to a critical review of informal web and social media commentary. After analyzing these organic community reactions, the chapter illustrates the iterative redesign into version 2.0. Finally, **Chapter 6** addresses the project's external ecosystem, detailing the architecture of the product website, synthetic media production, and digital marketing strategies.

[illegible]

Chapter 2

The Problems of Centralization and the P2P Alternative

In the contemporary digital landscape is present a profound dichotomy: while the user experience has reached unprecedented levels of fluidity and accessibility through centralized cloud architectures, this same convenience has come at the cost of resilience, data sovereignty, and environmental sustainability. This chapter interrogates the prevailing Cloud model of the modern web, identifying the most critical vulnerabilities inherent in centralized data custody.

The analysis begins by analyzing the socio-technical and ecological impact of the Cloud, arguing that the elimination of central servers is not merely a technical or privacy preference, but a necessity to reduce the carbon footprint and economic overhead of digital infrastructures. However, transitioning toward a serverless, Peer-to-Peer (P2P) ecosystem introduces a significant 'Usability Gap.' As cloud-based services have institutionalized specific interaction patterns, the shift to a decentralized model requires a radical reimagining of the User Experience.

By exploring the frictions between local-first autonomy and the 'always-on' expectations of modern users, this section maps the theoretical and practical framework for a new generation of sovereign applications. The objective is to demonstrate how design can bridge the divide between technical decentralization and high-fidelity usability, transforming the user from a passive consumer into an active, resilient node within a distributed network.



2.1 The Hegemony of the Cloud

The contemporary digital experience is almost entirely mediated by the Software as a Service (SaaS) model. In this paradigm, the user is no longer the owner of the tool or software used by the user, but a centralized utility accessed via subscription. From a systemic perspective, this creates an inherent asymmetry: the 'Cloud' functions as a digital landlord, where user data is stored in proprietary servers. This centralization of information not only erodes individual digital sovereignty but also creates massive 'honeypots' for cyber threats.

However, the vulnerability of the SaaS model extends beyond the application layer, reaching deep into the physical and logical infrastructure of the internet. In the last years 'concentration risk' has emerged: a big majority of global digital services rely on a strikingly small number of Hyperscalers, such as Amazon Web Services (AWS), Microsoft Azure, and Cloudflare. This hyper-centralization turns these providers into a systemic risk. As observed during major outages in recent years, such as the massive AWS US-EAST-1 disruptions in 2021 (ThousandEyes., 2021) and 2025, (TechUpkeep., 2025) or the global Cloudflare outages of 2025 (Inbound Fintech., 2025) and early 2026, (Cloudflare. 2026) a single configuration error in a lone data center can trigger failures worldwide. These incidents demonstrate that the 'Cloud' is not a resilient, ethereal entity, but a fragile chain of dependencies where the failure of one link can



paralyze millions of nodes simultaneously.

Beyond infrastructure fragility, the centralized SaaS model introduces a critical cybersecurity paradox known as the 'Honeypot Effect.' When aggregating the data of millions of individuals into single, centralized repositories, cloud providers create high-value targets for sophisticated cyber-attacks. While these corporations of course invest heavily in perimeter security, the systemic risk remains: a single breach or a privileged credential leak can compromise the integrity of an entire global user base. (Kaspersky., 2020)

Recent history has demonstrated that even the most robust security architectures are not immune to failure. From errors in configurations that lead into major breaches in cloud infrastructures to sophisticated supply-chain attacks like the SolarWinds incident, the 'fortress mentality' of centralized security has shown its limitations. In this context, the security of a service like Bitwarden or 1Password is fundamentally dependent on the provider's operational excellence, leaving the user with no agency in the event of a systemic compromise. The opposite, a P2P architecture shifts the security paradigm from 'perimeter defense' to 'distributed resilience.' By eliminating the central vault, the P2P model completely removes the honeypot, ensuring that even a successful attack on a single node does not threaten the stability or the privacy of the broader network.

Transitioning to a serverless, P2P architecture is therefore not just a matter of privacy, but a fundamental requirement for building a resilient digital habitat that can survive the inherent instability of centralized infrastructure.

Case Study: LastPass breach

The security incident that compromised LastPass between late 2022 and early 2023 serves as a definitive empirical critique of centralized credential management architectures, exposing the inherent vulnerabilities of the “cloud vault” model. When we deeply analyze the attack, the breach reveals that the primary weakness wasn’t in the cryptographic algorithms themselves, but rather in the structural concentration of risk. The attack unfolded through a sophisticated process of lateral movement, a technique where an intruder, after gaining an initial foothold in a low-security environment, systematically navigates through the network to reach high-value targets. This progression began far outside the corporate perimeter, specifically within the domestic sphere of a senior DevOps engineer. By exploiting a vulnerability in a third-party media software on a personal home computer, the attackers managed to install a keylogger: a malicious surveillance tool designed to record every keystroke made by the user. Technically, the keylogger allowed the attackers to capture the engineer’s master password at the very moment of input, effectively subverting the security of the interface before any encryption could take place.

Through this stolen credential, the attackers bypassed multi-factor authentication to gain access to the company’s cloud storage environment, specifically its Amazon S3 buckets, highly scalable, centralized “containers” used for storing vast amounts of data objects in the cloud. Accessing these servers allowed them to extract a massive amount of encrypted user vaults, creating a significant security paradox. Although the data was protected by a Zero-Knowledge architecture, a design principle where the service provider possesses no technical means to decrypt user information, the displacement of these vaults into the hands of unauthorized actors enabled offline brute-force attacks. In this scenario, attackers use automated scripts to systematically test millions of password combinations against the stolen data. Because the process happens “offline” on the attackers’ own hardware, it bypasses any server-side rate-limiting or monitoring systems, allowing the breach to continue indefinitely without detection. (LastPass., 2022)

Furthermore, the exposure of extensive unencrypted metadata, including vault URLs, email addresses, and IP addresses, revealed a secondary systemic vulnerability. This non-encrypted information effectively weaponized the users’ digital footprints, enabling highly targeted phishing campaigns and exposing behavioral patterns. Ultimately, the LastPass failure demonstrates that centralized security is only as resilient as its most privileged human link. Just this case provides a compelling justification for the transition toward peer-to-peer (P2P) architectures, where the elimination of a central authority and the distribution of data across sovereign, user-owned nodes transform security from a managed service into an emergent, resilient property of the network itself.

In this scenario, attackers use automated scripts to systematically test millions of password combinations against the stolen data.

This non-encrypted information effectively weaponized the users digital footprints, enabling highly targeted phishing campaigns and exposing behavioral patterns.

2.2 The Ecological and Economic Imperative: Distributed Sustainability

When we talk about the “Cloud” we always forget how the cloud itself acts, we project it with an image of ethereal weightlessness but the reality is that it is a heavy industrial reality with profound environmental impact. As a matter of fact, the centralized Software as a Service (SaaS) model is anchored in massive, energy-intensive data centers that function as the base of the internet we use every day. (Net Zero Insights., n.d.) These facilities not only require constant electricity for the computational work of millions of servers itself but also for industrial-scale cooling systems designed to dissipate the immense heat generated by concentrated hardware. From a systemic design perspective, this architecture creates a state, where the environmental cost of a service is inextricably linked to the maintenance of centralized, always-on infrastructures. By concentrating data and processing in a few geographical hotspots, the current model generates significant environmental externalities, creating a huge carbon footprint. Transitioning toward a serverless, Peer-to-Peer (P2P) architecture proposes a more harmonious and decentralized



structure based on the principle of resource efficiency. (MapMetrics., n.d.). A P2P network uses the “latent” computational capacity of end-user devices, smartphones, laptops, and tablets, that are already manufactured, powered, and integrated into the users’ daily lives. If the app distributes the data and computational power across these edge-nodes, the system doesn’t need anymore dedicated server farms. This redistribution can decentralize the carbon footprint of the application, utilizing only the existing energy flows of the user’s own hardware instead of requiring the continuous operation of redundant third-party infrastructure. In this sense, P2P design matches with the systemic goal of reducing embodied energy and maximizing the utility of existing hardware, proposing an ecosystem that scales organically without a linear increase in industrial energy demand.

Economically, the P2P paradigm fosters a shift from extractive models to sovereign ones. The traditional cloud-based ecosystem is built upon a “rent-seeking” infrastructure, but not only the users are victims of this system but even the developers are burdened by the “infrastructure tax” imposed by hyperscalers such as Amazon Web Services (AWS) or Microsoft Azure. This creates a precarious economic dependency: the survival and accessibility of digital services are contingent upon the financial stability and pricing policies of a few global gatekeepers. (1111 Systems., 2021) A serverless architecture disrupts this cycle by erasing the cost of server maintenance and renting. In a P2P system, instead of having an infrastructure that is a leased commodity we could have a shared, community-driven substrate. By removing the intermediary, the economic value is retained within the network of participants, ensuring that the service remains viable as long as its community of users exists. This transition democratizes the digital habitat, replacing centralized financial vulnerability with distributed economic resilience.

2.3 The Usability Gap: Designing for Sovereign Interaction and Transparency

The transition from centralized cloud services to Peer-to-Peer (P2P) architectures presents a profound challenge that is not only about technical feasibility but even into the realm of cognitive ergonomics and behavioral psychology. For over a decade, the “Cloud” has created a dominant design paradigm characterized by a “frictionless” user experience, where complexity is hidden behind a managed service. This model has cultivated specific mental models, the internal maps users hold of how a system works, where data is perceived as an omnipresent, indestructible entity managed by a benevolent third party. (NNGroup., 2024) For this reason, one big obstacle for P2P applications is the “Usability Gap”: the friction that starts when an individual is suddenly required to reclaim user agency, shifting from a passive consumer of a service to an active, responsible node within a network. In a systemic context, this requires a radical redesign of affordances to ensure that digital sovereignty does not translate into an overwhelming cognitive load.

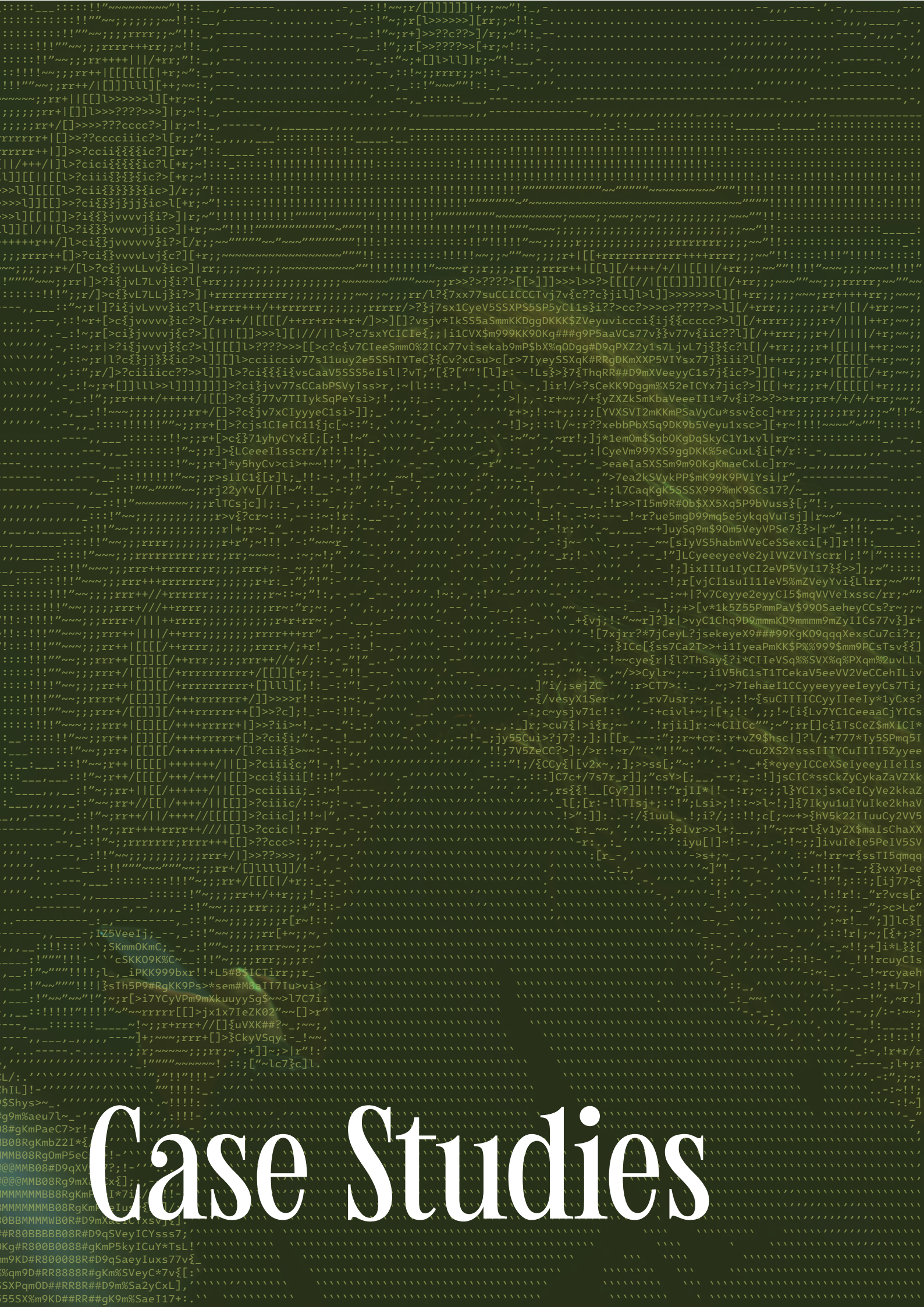
One of the most critical UX problems in serverless design is the management of identity and data recovery. In the centralized SaaS model, the “Forgot Password” function acts as a safety net provided by the service’s administrator. In a sovereign P2P system, without the presence of a central authority traditionally placed the entire burden of data custody on the user, where losing a master password meant the permanent loss of all information. However, contemporary studies in the P2P ecosystem bridged this gap by developing innovative, decentralized recovery protocols. By leveraging the synergy between secure distributed architectures and intuitive user flows, it is closer than ever the possibility to offer a “convenient” experience that rivals cloud services. These solutions, such as blind-peers recovery or local backups, allow users to regain access to their elements even in the event of total password loss. This evolution demonstrates that digital sovereignty does not preclude accessibility; instead, it challenges designers to build robust, decentralized systems that maintain autonomy while offering the psychological security typically associated with managed services.

Furthermore, unlike cloud-based applications that prioritize the illusion of instantaneous action, a systemic P2P interface must embrace Transparency of State. In a decentralized network, the “truth” of a shared resource is subject to network presence, meaning that updates take time to propagate across all peer nodes. Rather than employing specific patterns, which might mislead the user by pretending an action is complete before synchronization, the design must focus on true and reliable copies and UI. For example, when a user deletes a shared credential, the interface must explicitly communicate that the data may remain visible on the recipient’s device until a peer-to-peer connection is established.

This approach replaces the “black box” logic of the cloud with a series of Synchronous Feedback Loops. The goal is not to mimic the artificial speed of centralized servers, but to provide the user

with a clear understanding of the network's physical and logical constraints, presenting the reality as it is.

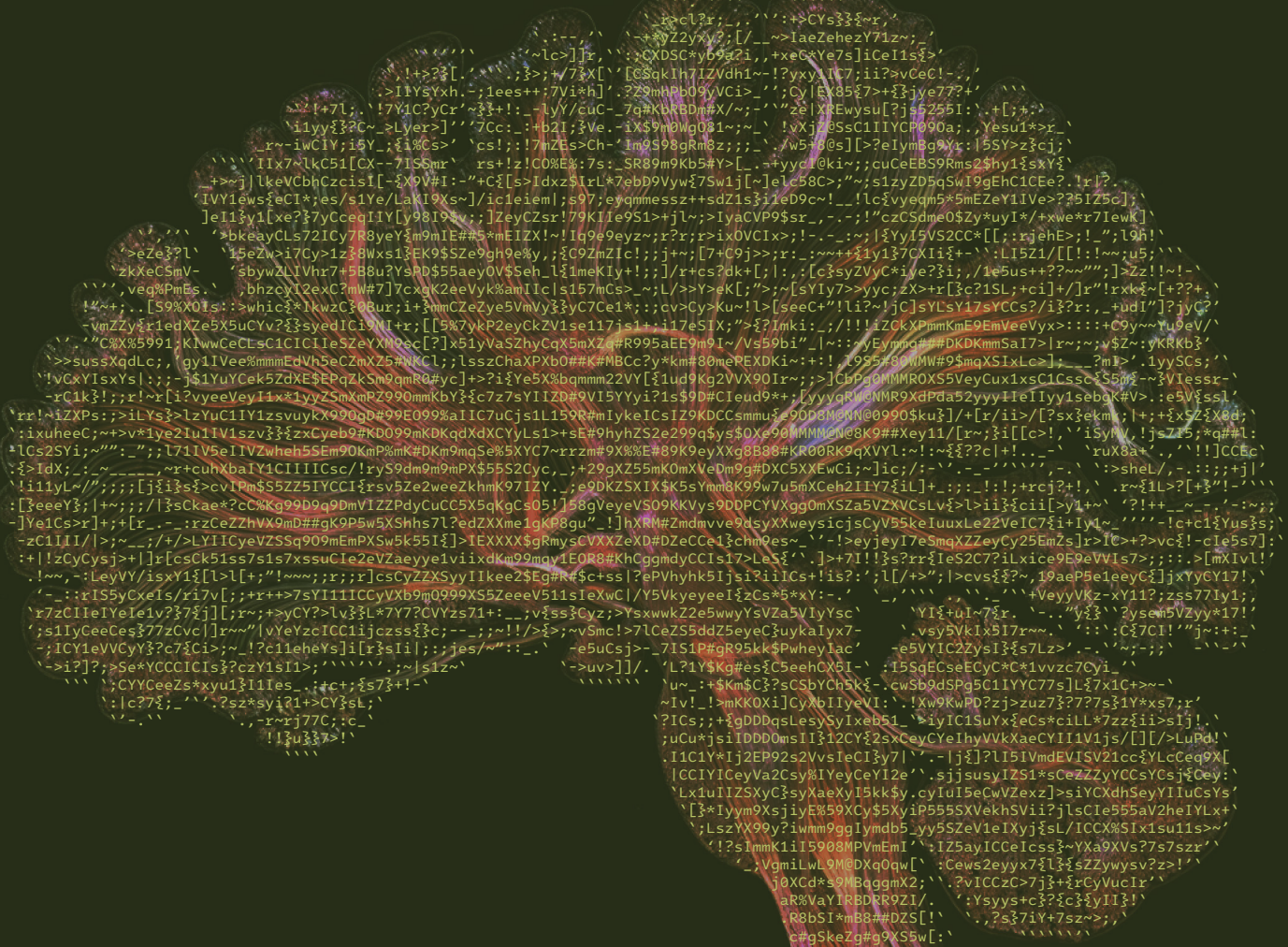
Chapter 3



Case Studies

3.1 Realigning Usability and Digital Sovereignty

The development of a serverless, peer-to-peer credential manager requires a comprehensive audit of existing state-of-the-art platforms. However, the design philosophy of PearPass does not advocate for a dogmatic or radical rejection of modern interface standards, which would risk alienation and cognitive rejection by the end-user. The objective is to design a digital habitat that feels familiar and intuitive, yet systematically recalibrates the contract of responsibility between the user and the infrastructure, granting individuals the transparent criteria needed to decide what is best for their own data environment. This to avoid a big cognitive load on the user, that in the case of an unfamiliar interface would need to understand not only the new P2P architecture but even a non-intuitive interface. Consequently, this chapter approaches the state of the art not to displace established interactive behaviors, but to identify foundational user expectations, such as seamless authentication, real-time data synchronization, and account recovery, and understand better how these flows can be reconfigured in a P2P way. By mapping the operational and design limits of market leaders, the audit establishes a conceptual baseline. It examines the exact points where interface design can maintain high levels of usability while replacing centralized data dependencies with local-first resilience. This macro-level diagnostic framework prepares the analytical ground for evaluating how current industry standards manage user trust, ultimately defining the structural criteria necessary to sustain a sovereign user experience.



3.2.1 Bitwarden: The Usability Traps of Hybrid Centralization

Infrastructural Topology and Trust Model

Bitwarden occupies a distinct position in the digital custody market by combining open-source transparency with a centralized cloud infrastructure. This dual identity creates a unique trust model: users are encouraged to trust the platform because its codebase is fully auditable, yet they must simultaneously submit to a centralized network metabolism. The application relies entirely on cloud routing to coordinate state changes between devices. From a systemic perspective, this configuration establishes a paternalistic relationship with the user. The data remains mathematically sovereign due to client-side encryption, but it is operationally dependent on a remote corporate server to achieve multi-device utility and vault synchronization.

Interface Layout and Information Architecture

The interface layout of Bitwarden translates this architectural tension into an experience of effortless but passive consumption. The information architecture organizes passwords, secure notes, and identity cards into clean visual containers, standardizing digital identity into discrete, searchable objects. This interface choice cultivates a rigid mental model where the user views security as something completely maintained by the application. The synchronization is hidden behind a minimalist layout, and this prevents the user from understanding the material location or movement of their credentials, sacrificing knowledge for immediate user convenience.

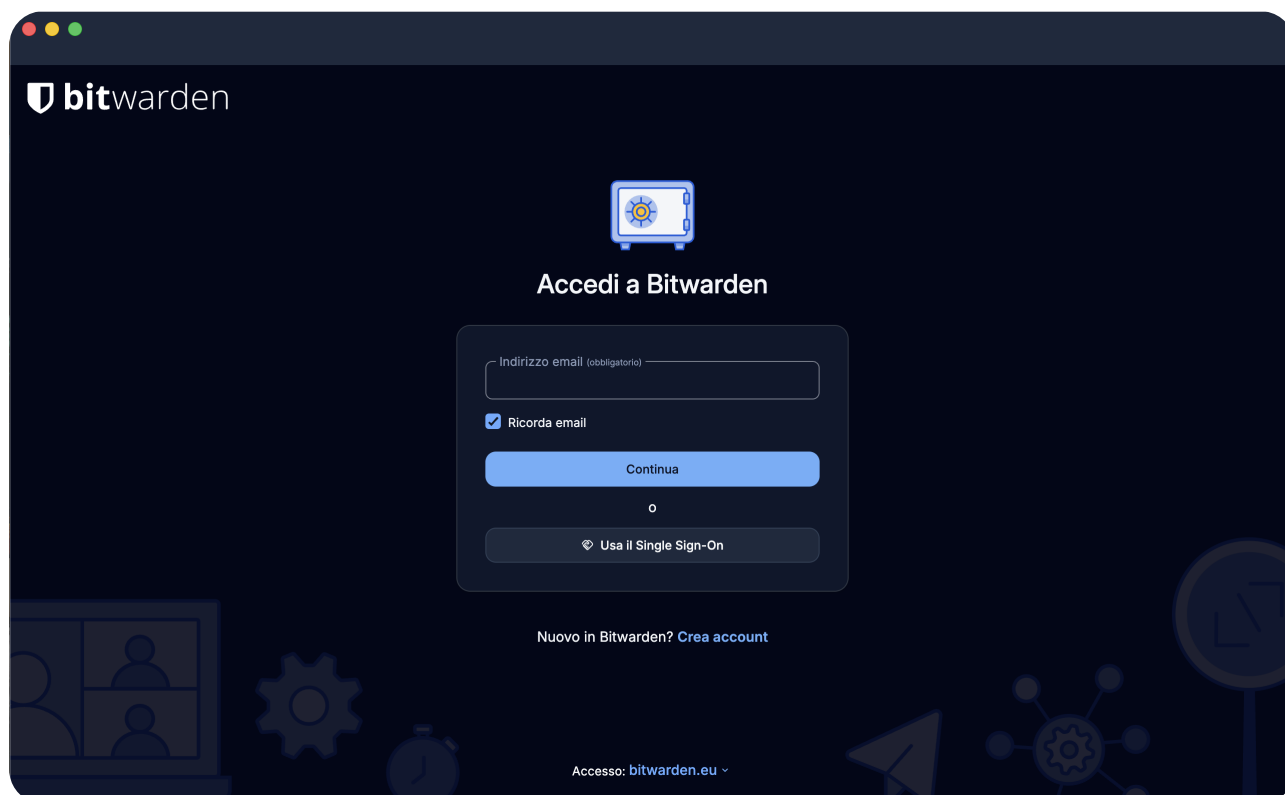
Systemic Friction and Interaction Breakdown

The primary systemic friction in Bitwarden's user experience emerges from the operational compromise between user convenience, cloud-first server partitioning, and Zero-Knowledge security boundaries. Because the platform adopts traditional SaaS onboarding patterns, the interface primes the user to expect the paternalistic and universal safety nets of the centralized web. However, this illusion of seamless ubiquity is structurally undermined by the platform's reliance on geographically siloed cloud data centers. Inside the picture we can see how the user needs to select manually the server region (e.g., separating accounts hosted on European servers from those on US nodes).

This server partition is set on the interface by a low-visibility dropdown menu located in the bottom part of the authentication interface. From a UX perspective this layout creates a severe pain point in the user. If an user unintentionally attempts to authenticate by selecting (which is precompiled) the incorrect server instance, the application delivers an uninformative rejection error, functioning as though the password is incorrect. This technical opacity shifts the burden of infrastructure localization entirely onto the individual, frequently triggering a state of acute user anxiety or panic regarding the permanent loss or deletion of their credentials. The breakdown demonstrates a fundamental tension within hybrid cloud designs: by concealing geopolitical and infrastructural partitioning behind an opaque UI

element, the system generates false mental models of universal access, leaving the user vulnerable to sudden operational lockouts.

▽ BitWarden Onboarding



3.2.2 1Password: The Premium Custody Standard and the Hybrid Paradox

Infrastructural Topology and Trust Model

1Password uses a framework that makes the company totally incapable of seeing the data saved by the users, it is done thanks to a data insulation through client-side encryption. This type of commitment is really noble by them, and the identity custodian retains absolute mathematical control over the encryption keys. Nevertheless, this kind of system contains in itself a profound compromise. The system tries to merge two fundamentally opposing paradigms: the centralized cloud distribution and the uncompromising local custody. Even if the user assets are encrypted they are still hosted inside some servers, this hybrid creates a system that is based on cloud computing without giving to the user the recovery convenience, creating a paradox.

Interface Layout and Information Architecture Analysis

The onboarding architecture of 1Password functions is of course the first step where the user gets in touch with the cryptographic complexity of the app. In the first image, the visual hierarchy reduces cognitive friction by focusing user attention on a minimalist password creation field. The interface purposely simulates the standard, reassuring aesthetics of a traditional commercial web login, yet it totally surprises the user expectation by embedding a critical disclosure card: “1Password cannot reset this password.” This copy reflects a conscious effort to prepare the user for his password ownership without alienating them through an overly dense technical interface.

This pedagogical scaffolding becomes explicit in the second picture, where the system talks clearly about the local key preservation. Because the infrastructure lacks a system where from the company side they can change passwords, the layout is forced to externalize the Secret Key through the generation of a static Emergency Kit PDF. The interface design frames this local archiving task as a premium security ritual, utilizing explicit, warning-hued typography to disclaim corporate liability. The user is instructed to store this artifact “somewhere safe,” a requirement that forces a digital cloud application to rely on the precariousness of local desktop file management or physical paper storage. This layout choice materializes the friction of the hybrid model, demonstrating how interface design is deployed to manage an inherent tension between remote hosting and local-first cryptographic custody.

Systemic Friction and Interaction Breakdown

This system is highly commendable within a commercial ecosystem that frequently commodifies user metrics. Without the ability from the corporate administrator to restore or override user credentials the app puts a big baseline for data integrity. The critics and the contradiction here is about its hybrid architecture. The platform is based on centralized cloud infrastructure to facilitate cross-device data distribution, a choice that makes the end-user to expect the effortless elasticity and safety nets of standard SaaS applications. Yet, to secure this cloud-hosted repository without central key management, the interface must introduce an offline, material fallback, the Emergency Kit PDF

documented in the second picture.

This configuration creates a severe operational misalignment. The user is caught in an architectural paradox: their encrypted identity and logins are contained inside a centralized remote silo, but the only way to access these files depends on his memory or on the analog desktop document. This friction demonstrates that forcing centralized cloud storage to coexist with localized cryptographic custody results in an unstable interaction model. It suggests that when a system demands absolute cryptographic responsibility from its users, maintaining data on remote corporate servers becomes an unnecessary systemic liability, exposing the conceptual limitations of hybrid cloud-zero-knowledge frameworks.

▽ 1Password Onboarding

The image displays three sequential screenshots of the 1Password mobile app's onboarding process. Each screenshot shows a status bar at the top with the time 9:41 and signal/battery icons. The navigation bar at the top of each screen includes a back arrow, the word 'Subscribe', and the 'Sign Up' title.

First Screenshot: The main heading is 'Set an account password you can easily remember'. Below it is an information box stating: '1Password cannot reset this password. If you forget it, you might lose access to your account.' There are two input fields: 'Password' and 'Confirm Password'. The 'Password' field has a hint 'Use 10 or more characters' and a character count '0/10'. A blue 'Create Account' button is at the bottom. A language selector at the very bottom shows 'English'.

Second Screenshot: The heading remains the same. The information box is still present. The 'Password' field now shows 10 dots and a green checkmark with the text 'Use 10 or more characters'. The character count is '19/10'. The 'Confirm Password' field also shows 10 dots. The blue 'Create Account' button is still visible. The language selector at the bottom shows 'English'.

Third Screenshot: The heading is 'Save your Emergency Kit'. It features an illustration of a document labeled 'Emergency Kit'. The text reads: 'If you get locked out of your account, you will need your Emergency Kit to sign in. Download your Emergency Kit PDF and keep it somewhere safe.' Below this is an information box: 'It contains your Secret Key, which we cannot access or recover for you if you lose it.' A blue 'Download PDF' button is at the bottom. The language selector at the very bottom shows 'English'.

3.2.3 KeePassXC: Isolated Local Vaults and Architectural Fragmentation

Infrastructural Topology and Trust Model

KeePass represents the historic platform of the fully offline data repository. This app is in total contrast with the cloud-dependent platform because it operates without the internet by default. The whole system relies only on local storage, keeping all the user credentials inside a single encrypted file stored inside the user's device. Of course, this local model is highly commendable because it acts with a strict standard of digital sovereignty. Using this local approach, KeePass completely deletes the external "honeypot", fully protecting its users from possible corporate data breaches. The trust architecture is entirely localized: security is not a managed third-party service, but a physical property of a file stored securely within the boundaries of the user's own hardware habitat.

Interface Layout and Information Architecture Analysis

The interface of KeePass is easily recognizable as an old interface, the layout is indeed characterized by high informational density and low visual abstraction. The information architecture relies on a rigid, two-panel tree hierarchy: the left sidebar lists multi-layered folder directories, while the right canvas exposes raw transactional tables of titles, usernames, and URLs. Unlike premium consumer applications, the layout offers minimal sensory scaffolding or automated visual feedback. The advanced settings of the app such as security parameters, key derivation options, and encryption algorithms are exposed directly to the user as advanced configuration inputs inside nested modular menus. While this transparency provides technical experts with absolute control, it imposes a severe cognitive load on standard users, creating a user experience that is a complex system administration task rather than a fluid daily tool.

Systemic Friction and Interaction Breakdown

When evaluated through a systemic lens, KeePass exposes a profound topological paradox that fractures its original sovereignty promises. The user's necessity for cross-device availability totally destroys the philosophy of the app, as a matter of fact the user needs to accept a compromise to gain cross-device function, move the encrypted file manually inside another device or upload that same file inside a cloud storage such as Dropbox, iCloud, or Google Drive.

This creates a nearly comic paradox, an application created and selected specifically to avoid centralized cloud structure needs to rely on a cloud to perform a basic operational sync. Of course the vault remains encrypted, but the transport and the availability of it are delegated to third-party gatekeepers.

Furthermore, saving multiple local database files creates lots of duplicates that creates data duplication or irreversible overwrites. KeePass's failure is therefore identified as an infrastructural limitation: by treating local storage as an absolute vacuum, it drives users to adopt unsafe centralized compromises, demonstrating the urgent design need for peer-to-peer network layers that natively execute secure, serverless synchronization without collapsing back into cloud enclosures.

Create a new KeePassXC database...

General Database Information

Please fill in the display name and an optional description for your new database:

Database Name: Passwords

Description:

Go Back

Continue

Cancel

△ Database (Vault) Creation

▽ Homepage

Demo Passwords* - KeePassXC

Database

Entries

Groups

Tools

View

Help

+

✖

🔑

🌐

🗂

⚙

Search (Ctrl+F)...

	📎	🕒	Title	Username	URL	Notes
🌐 Internet	📎	🕒	🍏 Apple	john.doe@icloud.com	https://www.icloud.com	Username is t
📁 Coding			📁 Dropbox	john.doe@example.com	http://www.dropbox.com	
🎮 Gaming			📺 Netflix	john.doe@example.com	https://www.netflix.com/	
🛒 Shopping			📁 Nextcloud	john.doe	https://apps.nextcloud.com/	
👤 Social			📁 Pocket	john.doe	http://getpocket.com/a/queue/	
💻 My Computer			📁 IFTTT	johndoe	https://ifttt.com	
🏠 Real world			📁 Google	johndoe@gmail.com	https://google.com	
🗑 Recycle Bin						

🍏 Passwords / Internet / Apple

550 709

🕒

✖

General

Advanced

Autotype

Username

john.doe@icloud.com

URL

https://www.icloud.com

Password

🔒 •••••

Expiration

Never

Tags

Notes

Username is the Apple ID

Searches and Tags

🔍 Clear Search

🔍 All Entries

🔍 Expired

🔍 Weak Passwords

3.3 Synthesis: Comparative Evaluation of Market Paradigms

Analytical and Interface-Compliance Focus

The comparative audit executed across market competitors reveals that the contemporary digital custody landscape is bounded by a fragmented trade-off between architectural security and user convenience. What this section wants to achieve is to synthesize the structural characteristics of the competitors of PearPass, underlying the operational archetypes. This evaluation matrix wants to isolate the core points of the interaction failure, making understand how this affects the user experience and behavioral workflows. This process isolates the core points of interaction failure, mapping how network topologies directly condition the user experience and behavioral workflows. To visually understand this socio-technical diagnostic, the evaluation matrix below maps the trade-offs of the audited platforms. By structuring the analysis from direct competitor observation to abstracted technical topologies, the table categorizes the contemporary market along critical axes: data storage infrastructure, identity verification, synchronization interfaces, and the primary human-computer frictions that emerge during operational crises.

Competitor	Architectural Archetype	Network Topology & Storage
Bitwarden	Cloud-Dependent SaaS	Centralized Hub-and-Spoke. Data aggregated in remote, multi-tenant server clusters.
1Password	Hybrid Zero-Knowledge	Centralized Hub-and-Spoke. Encrypted vaults stored remotely but unlocked via a local dual-parameter system.
KeePass	Isolated Local Vault	Fully Offline. Default zero-network footprint. Single database file stored strictly on local hardware.

3.4 Concluding Remarks: Bridging the Market Critique and Design Methodology

In conclusion, analyzing the current digital custody market highlights a fundamental problem: trying to combine centralized cloud distribution with local cryptographic custody creates confused user mental models and significant operational risks. The study of current market leaders shows that neither hybrid cloud systems nor completely offline vaults successfully balance user convenience with true data sovereignty. The infrastructure problems and interface failures identified in these platforms are not unchangeable technical limitations, but rather reveal a clear lack of user-centered design.

This synthesis gives the possibility to shift the focus of this thesis from the analysis of existing problems to practical design solutions. The limitations and user anxieties found in current tools provide the direct requirements for the PearPass application. To protect user data and give individuals true control without increasing cognitive load, the design language must actively guide the technology rather than just adapt to its constraints. Ultimately, these findings justify the transition from a theoretical review of market frictions to the Research through Design (RtD) framework explained in Chapter 4, where creating the app and iterating on that serves as the main tool for research and product strategy.

Identity & Recovery Model	Synchronization Interface	Primary UX Vulnerability / Friction
Zero-Knowledge framework anchored to traditional email login. Employs a fallback password hint via external SMTP protocols.	Automated background sync. Immediate interface validation masks actual network latency.	Infrastructural Dependency: Low-visibility regional server selectors trigger user anxiety and authentication failures if misconfigured.
Local Master Password coupled with a 128-bit Secret Key. Relies on an analog/ material fallback (Emergency Kit PDF).	Automated cloud-hosted replication across user nodes.	Architectural Incongruity: Dissonance between a frictionless cloud marketing promise and the high-stakes responsibility of manually preserving an offline PDF file.
Absolute self-sovereignty. Absolute accountability with no administrative or external override networks.	Natively non-existent. Synchronization requires manual file migration or third-party interventions.	Topological Paradox: Users are forced to reintroduce centralized commercial cloud storage (Dropbox, iCloud) to achieve sync, breaking privacy perimete

Chapter 4

Methodology

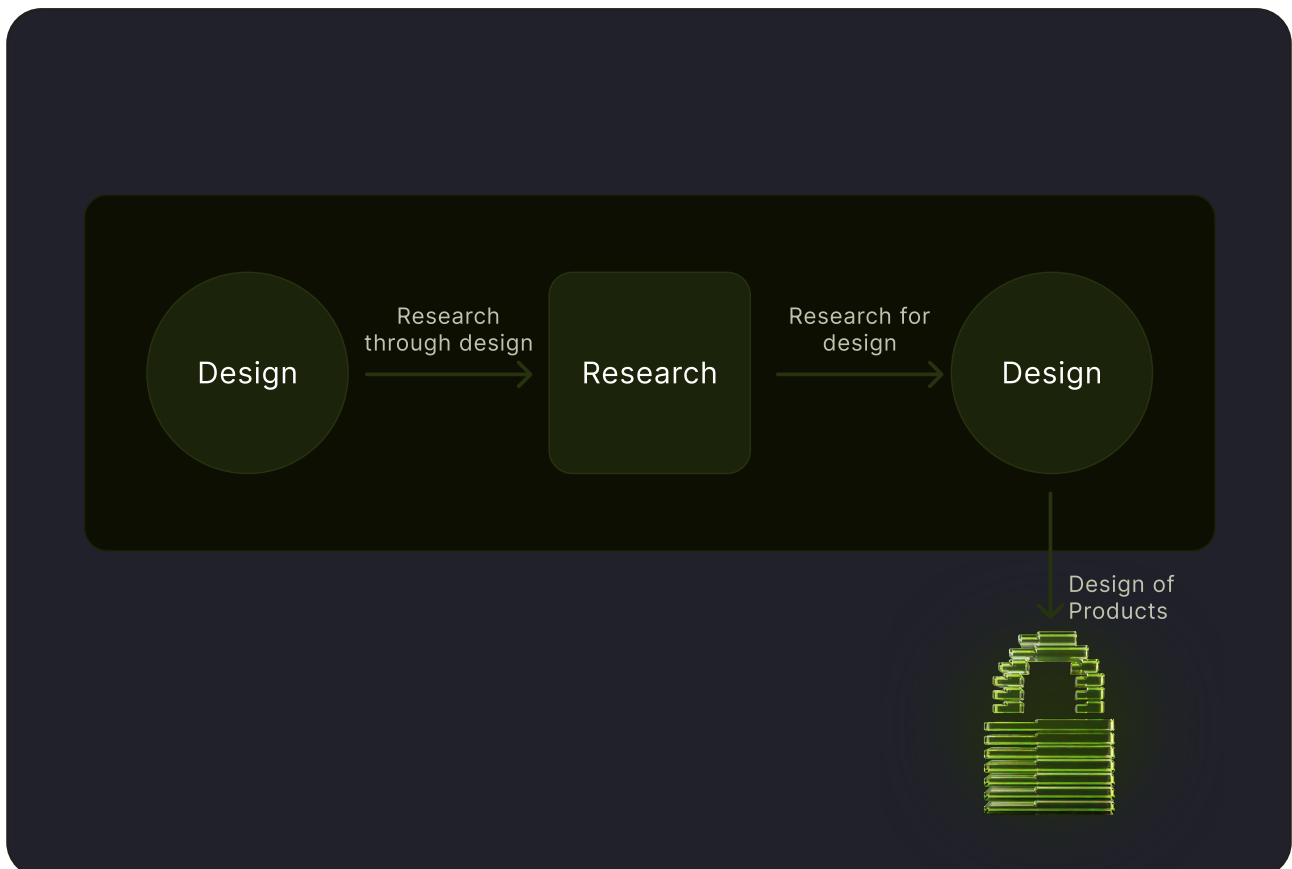
This chapter delineates the process undertaken to move from some theoretical principles of decentralized architectures into a functional and user-centric digital app. Moving from the critique of centralized infrastructures established in the previous chapter, we will now focus on the practical implementation of PearPass. The primary objective of this research is to investigate how good-crafted UX/UI design can be used to fulfill the gap between the rigorous requirements of peer-to-peer (P2P) engineering and the cognitive expectations of contemporary users. The investigative process is characterized by a non-linear, recursive approach that acknowledges the complexity of designing for a serverless ecosystem. To provide a comprehensive overview of the project's evolution, the chapter is structured into four key sections. First, it defines the Methodological Framework, introducing the concepts of Research through Design and Evolutionary Prototyping as the foundational pillars of the inquiry. Second, it identifies the User Archetypes and Proto-personas, mapping how different users have different needs or mental models. Third, it illustrates the Network Relationships and Information Flows, visualizing the direct, unmediated connections between devices and the movement of data within the mesh. Finally, it details the Iterative Prototyping phase, showcasing how technical constraints were transformed into transparent and intuitive design affordances. Through these different lenses, the chapter seeks to demonstrate that the design of PearPass is a strategic effort to foster digital sovereignty through resilient and informed interaction patterns.

4.1 Methodological Framework: Research through Design (RtD) and Design-Led Strategy

The development of PearPass adopts a Research through Design (RtD) methodology, where the creation of a digital artifact as the best tool for the designer itself. In this approach, the prototype is not merely a final output but an investigative medium, a tool that both developers and designers can use to explore and define the future of decentralized interactions. This methodology allows the designer to move beyond existing technical boundaries, using the project as a space for speculative and strategic reflection. A fundamental pillar of this process is the Design-Led Product Strategy, developed through continuous brainstorming sessions between the Designer and the Project Manager (PM). In these sessions, the vision, scope, and purpose of PearPass are established a priori, often anticipating the actual development of the technology needed itself. The designer, developing a high-fidelity prototype before technical implementation, can, inside with the product team, understand better the needs of the end-user, creating a precise functional roadmap for the engineering department. Thanks to this prototype, the team has a strategic blueprint that gives the possibility to optimize technical resources and avoids the development of superfluous and misaligned features. The design is indeed a proactive catalyst that challenges the underlying technology to create the best product. Consequently, the methodology is characterized by Visionary

Prefiguration and Recursive Refinement. The interface acts as a strategic bridge between high-level conceptual goals and technical implementation. The app evolves following these rules, while the technology provides the substrate, the design gives the direction. By documenting that the project vision is always preceding the technical

By documenting how the project vision consistently precedes the technological part, this research demonstrates that high-fidelity usability and digital sovereignty are not consequences of technology, but intentional outcomes of a design-led development process.



4.2 Primary research instruments: Quantitative Survey

To map the contemporary landscape of digital credential management, a dedicated primary research instrument was deployed in mid-2026. Gathering data within highly specialized privacy-centric domains and general consumer circles presented significant distribution challenges, resulting in a bounded sample size of 28 highly qualitative responses. Despite the compact volume of respondents, the dataset captured in the form provided a dense cross-section of highly polarized user profiles. This data density successfully illuminated the deep infrastructural anxieties and ecosystem dependencies necessary to extract and define our core protopersonas.

4.2.1 Structural Deconstruction of the Survey Instrument

The quantitative form was engineered not as a linear questionnaire, but as a multi-tiered diagnostic matrix. It searches elements across six core structural axes:

- **Socio-Demographic Positioning & Technical Literacy:** Inquiries regarding age group, main occupation, and self-rated tech-savviness established the foundational capacity of the human node. This layer maps how specialized engineering skills or standard consumer digital habits condition an individual's willingness to navigate non-traditional network topologies.
- **Ecosystem Path Dependency & Behavioral Topology:** By auditing current password-tracking methods (e.g., notebook, browser keychains, or dedicated SaaS tools), device usage distributions, and active account estimations, this dimension isolated the user's current structural locked-in state within centralized ecosystems.
- **Friction Vector Analysis & Switch Triggers:** Focusing on existing pain points (such as failing autofill frameworks or device synchronization latencies) alongside specific catalysts for switching applications, these questions identified the precise operational limits of hybrid-cloud models.
- **Cognitive Sovereignty & Risk Perception Dynamics:** This critical tier evaluated the psychological tension between managed cloud convenience and absolute data ownership. It explicitly measured the user's willingness to assume the total responsibility for local database backups in exchange for strict cryptographic privacy.
- **Atomic Component & Functional Prioritization:** A granular evaluation matrix prompted users to weight the necessity of secondary functional features (such as integrated 2FA authenticators, automated password generators, biometric gates, and passkey support), defining the baseline utility requirements for the interface.

- **Aesthetic & Operational Affordances:** The final exploratory tier analyzed user preferences regarding visual “feel” (minimalist vs. institutional vs. friendly) and the fundamental structural dichotomy between automation (paternalistic system action) and control (active user confirmation), providing a direct layout blueprint for the PearPass development cycle.

4.1.3 Empirical Findings and Data Synthesis

A quantitative and thematic analysis of the empirical data captured in the form reveals a profound systemic polarization among the 28 respondents, defining a trajectory for serverless application frameworks. In terms of technical baseline capability, the user sample is symmetrically divided between specialized practitioners and digital natives: 42.9% identify as “Expert” actors operating professionally within the tech/IT sector, while another 42.9% possess a “Medium” level of digital literacy, with the remaining 14.3% displaying high configuration literacy. This technical stratification directly conditions their current data-custody architecture. Exactly 35.7% of respondents delegate their credentials to automated, ecosystem-locked browser keychains, while the other 35.7% utilize dedicated, cloud-hosted SaaS password managers. The remaining 28.6% operate within fragmented, defensive manual fallbacks, relying on unsafe string variations (14.3%), unprotected text files (3.6%), or physical paper notebooks (7.1%).

Critically, the survey isolates a structural collapse of user trust regarding third-party managed server clusters. When evaluating their psychological security within centralized cloud infrastructures, the absolute majority of responses skewed toward moderate-to-low confidence levels, exposing a latent anxiety toward the centralized honeypot effect. This structural vulnerability is further illuminated by the primary triggers required to incentivize an application switch. For dedicated SaaS users, operational frustrations originate from micro-UI friction and macro-UX systemic failures, specifically failing autofill scripts on complex web domains (42.9%) and opaque cross-device sync latencies (28.6%).

Conversely, for non-users, the barrier to adoption is driven by structural distrust (fear of data harvesting or server leaks) paired with pricing inflation. The most definitive finding lies in the user's readiness to reclaim digital sovereignty: a striking 64.3% of respondents stated they are fully willing to assume absolute responsibility for managing their own local database backups and cryptographic security perimeters in exchange for the certainty that no corporate entity can access their digital twin. Furthermore, 64.3% confirmed that an explicit local-only declaration ("Your data never leaves this device") acts as a powerful confidence amplifier.

These metrics provide the definitive empirical requirements to dismantle the convenience-sovereignty paradox, justifying a serverless P2P mesh architecture where human agency is prioritized over paternalistic cloud automation.

4.3 Protopersonas Extraction

The transition from the quantitative diagnostic matrix detailed in section 4.2.1 to the qualitative synthesis of target users requires a conceptual shift from static data to needs. Users are not isolated operators but dynamic, human edge-nodes whose daily digital habits directly condition the tool choosed daily. By map-matching the empirical vectors of technical literacy, ecosystem path dependency, and risk perception captured in the form, this research crystallizes the raw survey metrics into three distinct protopersonas.

[illegible]

The Autonomous Expert

Vincenzo

29 years

Freelance DevOps Engineer

Berlin, Germany

Vincenzo represents the highly technical spectrum of data custody within contemporary networks. As a freelance DevOps engineer operating a small business alone in Milan, his daily workflow demands a complex, multi-device infrastructure where he constantly shifts between a heavy Linux desktop workstation, a corporate MacBook Pro, and his personal smartphone to monitor active deployments. This advanced operational layout shapes a rigorous mental model rooted in absolute digital sovereignty and verifiable cryptographic perimeters. Managing over 150 high-stakes infrastructure accounts and root SSH keys, he actively rejects standard first-party browser keychains and commercial password managers, which he diagnoses as single points of failure vulnerable to lateral movement and corporate data breaches. Instead, he enforces isolated client-side encryption, purposefully choosing high intentionality over immediate, frictionless convenience. His relationship with digital tools is strictly control-driven; he demands explicit confirmation loops and structural transparency of state at the interface level, ensuring that no third-party intermediary or opaque server architecture can compromise the mathematical finality of his local cryptographic keys or manipulate his sovereign data habitat.

Technology Level

Low

Medium

High

Expert

Vision of Cloud Services

Negative

Neutral

Positive

"Forcing an expert user to manually select between isolated, regional server silos just to authenticate is an architectural absurdity. The infrastructure should resolve network routing and server localization automatically behind the scenes, rather than offloading structural partitioning onto a low-visibility UI element."



The Ecosystem Fan

Matteo

24 years
Postgraduate Design Student
Turin, Italy

Matteo represents a significant portion of digital natives whose interaction with software is bounded by the illusion of frictionless simplicity. As a postgraduate university student in Turin specializing in sustainable tailoring and manual crafts, his primary focus is aesthetic execution and fluid user interaction. He relies on a MacBook Air and an iPhone to maintain his design portfolios and manage an emerging independent e-commerce portal. Exhibiting a medium level of technical literacy, Matteo views underlying network architectures as an invisible, magical “black box” managed by big corporates. Consequently, he delegates his entire digital identity, spanning 31 to 50 active online accounts, to automated, ecosystem-locked browser keychains for immediate cross-device utility. This complete cognitive offloading creates an extreme dependency on remote corporate servers. Because his mental model equates unmediated automation with security, he remains entirely oblivious to the material reality of data gravity, corporate data harvesting, and the strategic enforcement of proprietary vendor lock-in that paralyzes user autonomy once his credentials are pushed into closed cloud repositories.

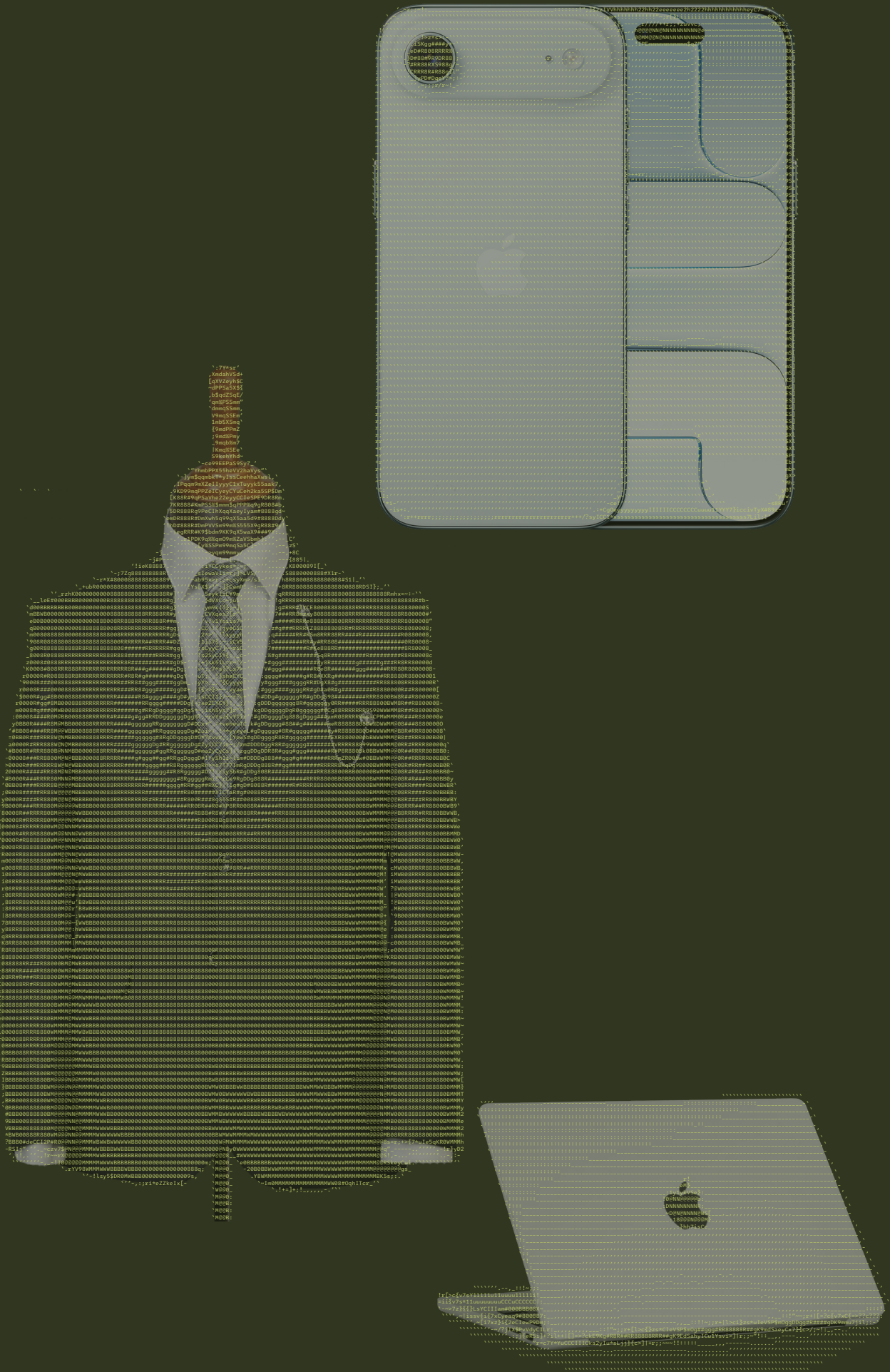
Technology Level

- Low
- Medium
- High
- Expert

Vision of Cloud Services

- Negative
- Neutral
- Positive

“I love the seamless automation of my browser keychain. It works so well between my Iphone and my Macbook. I would never change that”



The Data Paranoic

Elena

31 year
Digital Right Advocate
Rome, Italy

Elena embodies the acute psychological anxieties born from the contemporary crisis of digital centralization. Working in Rome as an independent digital rights advocate, she possesses a solid understanding of cognitive ergonomics but operates under a state of high anxiety regarding mass data extraction, predictive algorithmic profiling, and the multi-billion-dollar data brokerage market. Deeply affected by hyper-publicized infrastructural breaches and security paradoxes, such as the Avast/Jumpshot clickstream monetization scheme and the LastPass Amazon S3 bucket compromise, her trust in centralized SaaS models has completely collapsed. To preserve her digital twin and protect her non-negotiable right to digital sovereignty, she has executed a radical behavioral retreat from the web. She completely refuses to use any digital credential managers, opting instead for a highly fragile, defensive analog workaround by manually recording her entire cryptographic identity, banking access points, and professional databases within a physical paper notebook kept on her desk.

This manual ritual isolates her from cloud networks but forces a precarious dependency on a material object, leaving her highly vulnerable to geographic dislocation and operational paralysis.

Technology Level

- Low
- Medium
- High
- Expert

Vision of Cloud Services

- Negative
- Neutral
- Positive

"Every time a cloud provider claims to be an impenetrable fortress, they get hacked weeks later. I would honestly rather risk losing my physical paper notebook during a cross-country trip than surrender my sensitive personal data to another corporate cloud honeypot."

Chapter 5

Project

5.1 Introduction

This chapter marks the transition from methodological prefiguration to the practical execution and evolutionary design of the PearPass digital habitat. Chapter 5 explores how organizational team structures, community feedback loops, and user interfaces influenced the creation of the app. The narrative traces the application's lifecycle across key milestones to illustrate how complex decentralized protocols are progressively translated into intuitive human experiences. The chapter begins by examining the early dynamics of the development team, analyzing how the initial lack of a Project Manager influenced the software's usability. It then details the old interface, user flows, and interactive constraints of version 1.0. This baseline is subsequently contrasted with a qualitative study of informal web and social media commentary from platforms such as X (Twitter) and YouTube, mapping the problems highlighted by the community. Ultimately, these insights justify the final section, which documents the architecture of version 2.0 through its dedicated Design System, re-engineered sovereign workflows, and transparent interface components.

5.2 The Product Evolutionary Cycle: Team Dynamics and Professional Growth

PearPass during its development had diverse challenges, from a managerial field to the design one. As a first-time application designer, the development initial of version 1.0 was characterized by significant practical inexperience. At the start the team lacked a Project Manager (PM) that could act as a mediator between design and software engineering. Due to these reasons, the first iteration of the application was heavily driven by technical feasibility rather than user-centric principles.

However, the weeks following the initial release served as a critical turning point. During this period, the team collected organic feedback from the user community, at the same time there has been a significant evolution in my own design capabilities and technical understanding. To work on this new version, the team structure was expanded to include a PM and an additional designer. This structural shift allowed us to fundamentally change our development methodology. Consequently, the expanded team executed a major UX/UI redesign that re-engineered fundamental user flows, translating complex peer-to-peer protocols into intuitive and predictable digital affordances.

5.3 Technical and Interactive Baseline of Version 1.0

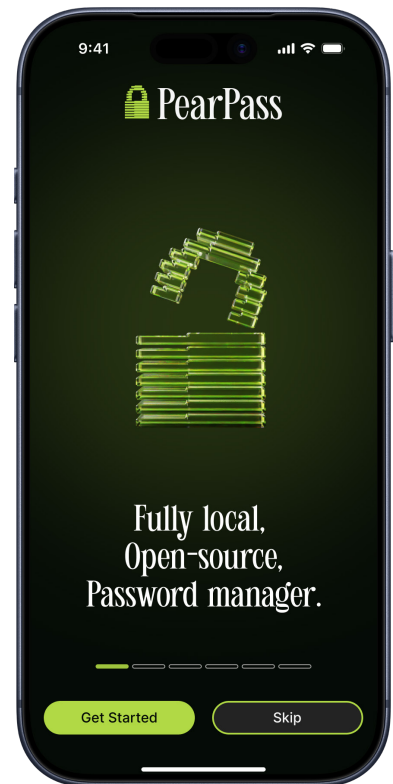
To fully understand the limitations of PearPass version 1.0, this section performs a detailed diagnostic audit of its primary interface pathways. By critiquing several core user flows, the analysis highlights the specific problems and design choices that the application's in the first release had.

Onboarding

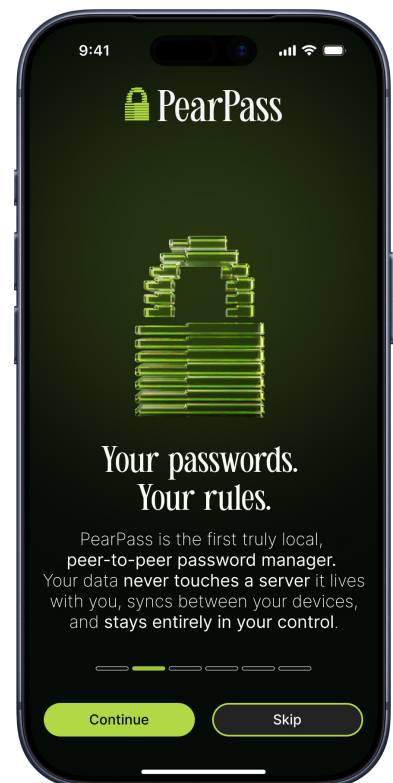
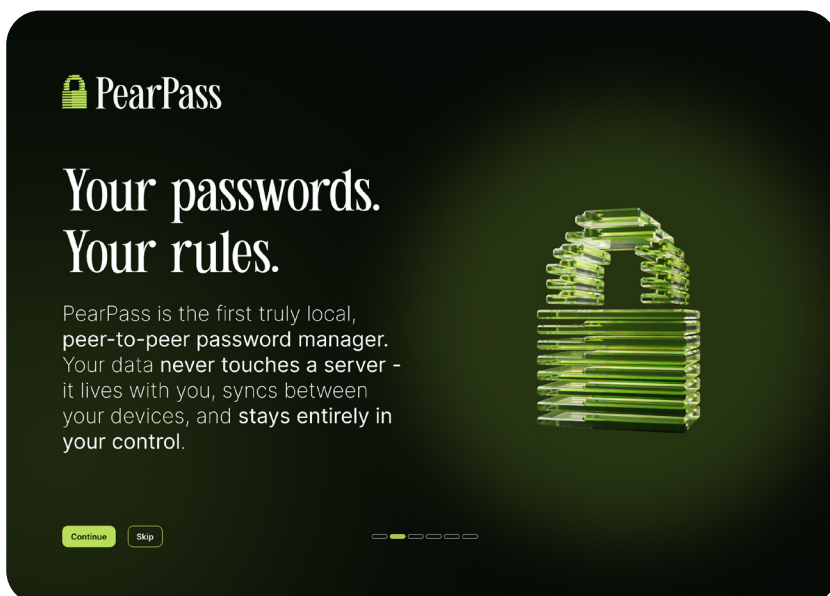
The initial onboarding and configuration flow of PearPass version 1.0 reveals several structural redundancies. As we can see inside the pictures, the interface introduces a long, six-step introduction carousel. The majority of the copywriting focuses on generic features common to standard password managers, such as organizing logins and credit cards. Giving all this space to standard features, the layout fails to highlight the unique peer-to-peer value proposition of the software, while misusing the user's cognitive attention before any security configuration begins. This linear friction continues into the infrastructure setup screens. After forcing the user to create a global Master Password, the interface presents a dedicated fork for vault configuration. Forcing the user to manually select "Create a new vault" during the first app launch creates unnecessary decision paralysis. Considering that a functional vault is mandatory for the application to operate, this step could easily occur by default behind the scenes. Finally, the "Create new Vault" screen introduces an optional secondary vault password right after the master credential creation. This choice creates a confusing security hierarchy, as the interface does not clarify the structural relationship or operational difference between the two password levels, leading to broken user mental models.

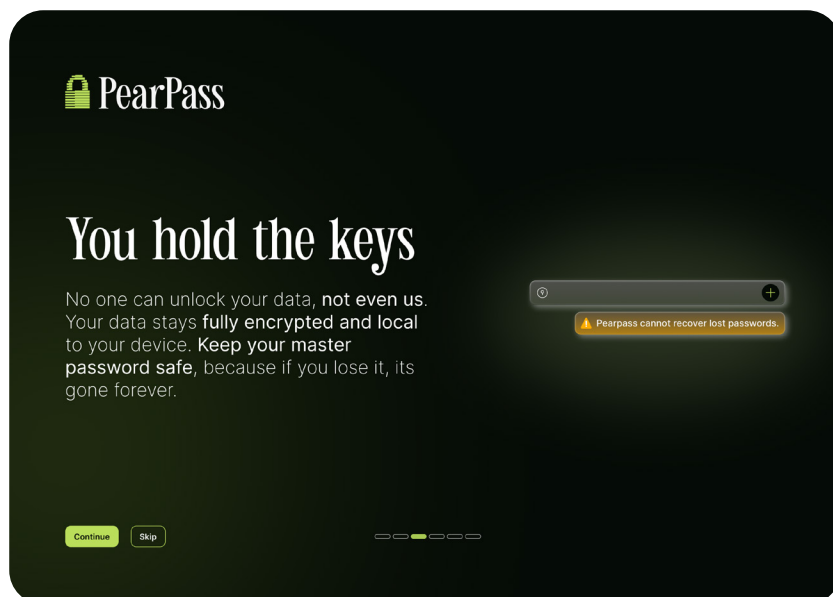


This hero screen establishes PearPass's local-first identity.

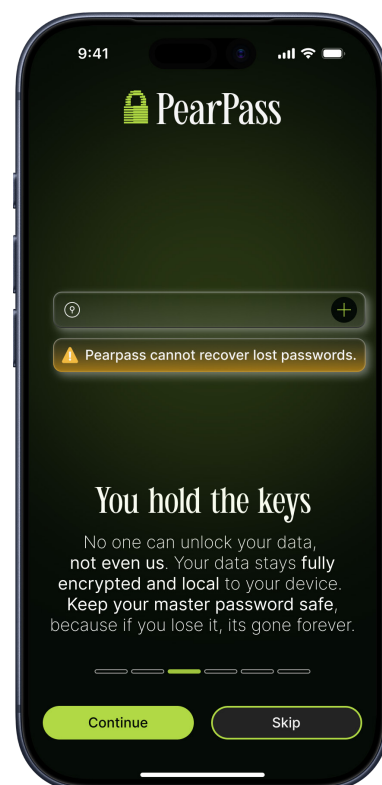


This view operationalizes the peer-to-peer paradigm, using straightforward micro-copy to inform the user that sensitive data is retained exclusively at the network edge without relying on centralized servers.

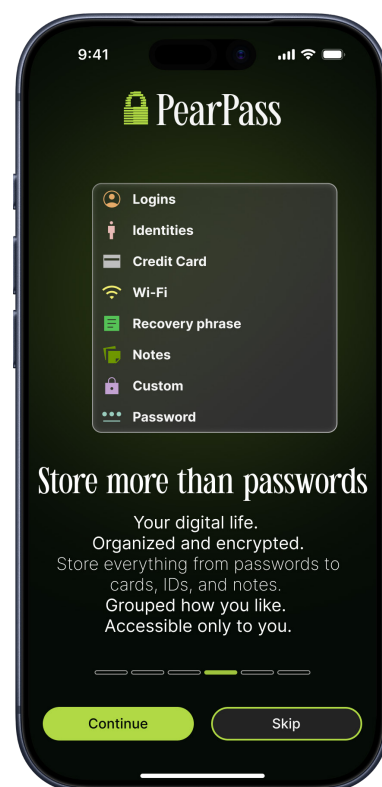
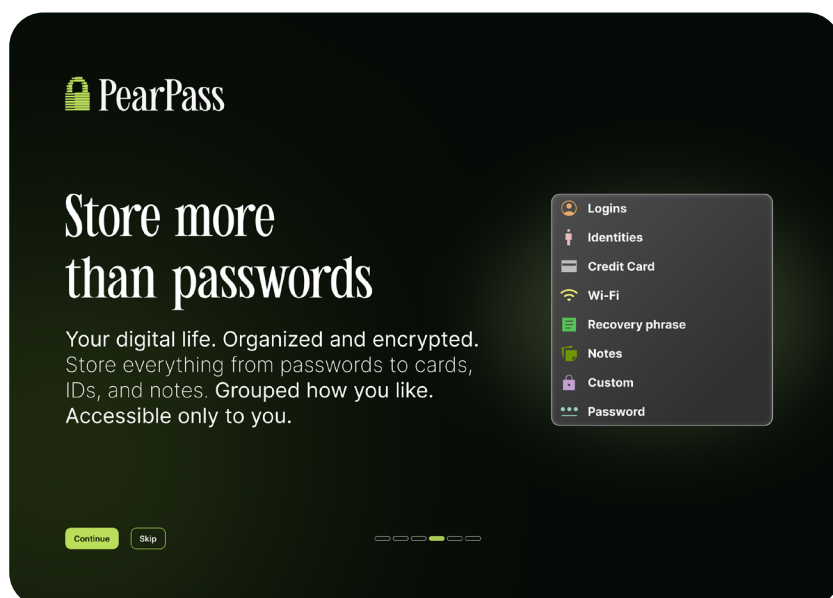




The user gets warned about the zero-knowledge architectures, with a warning animation that make the whole screen more dynamic and understandable.

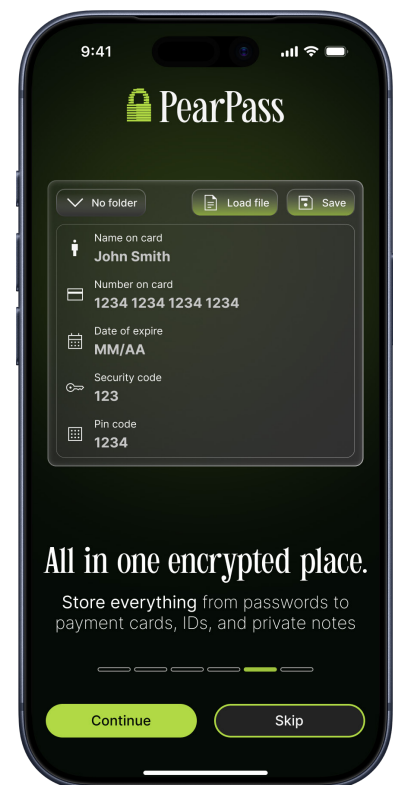


This screen shows the different type of entries you can store inside PearPass.

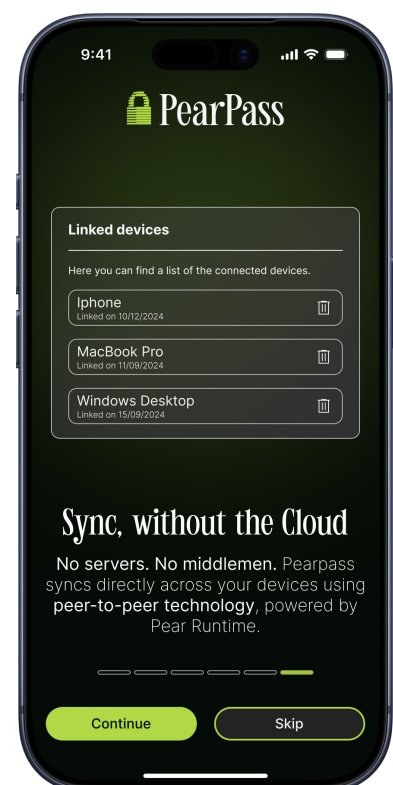
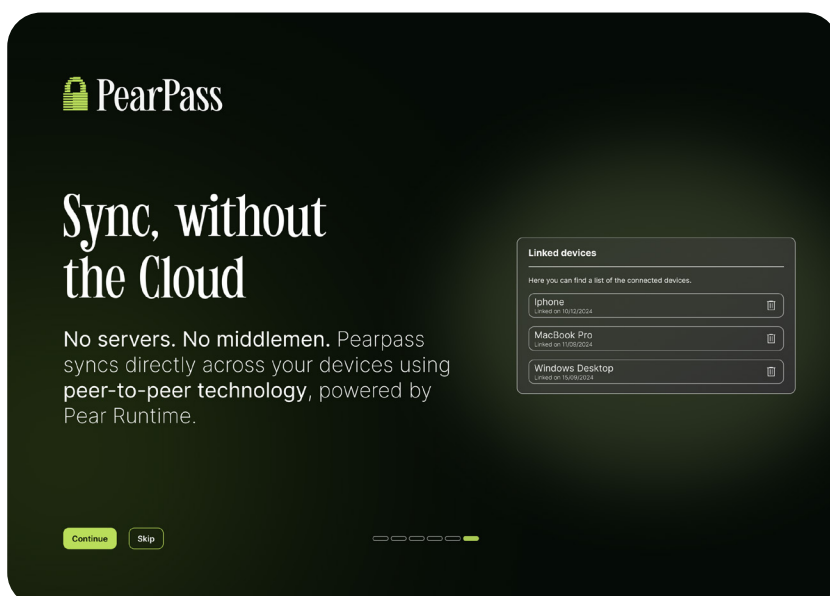


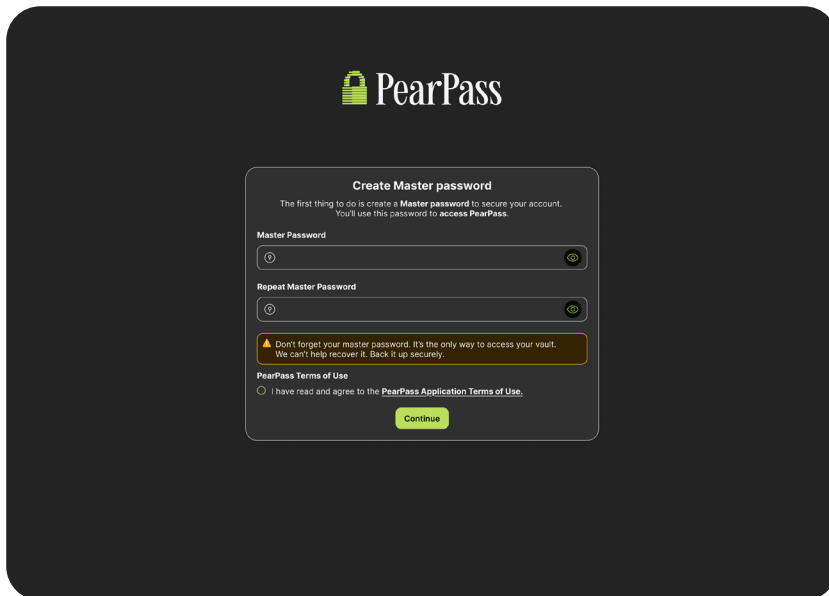


This step reinforce the type of items you can save inside the app and communicate the encrypted status of the Vault.

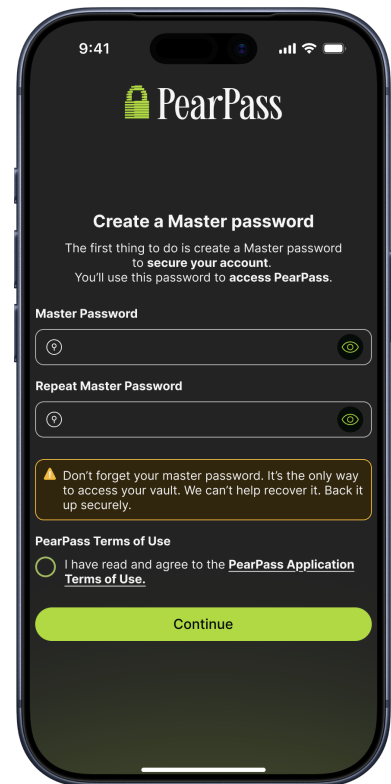


In this screen we explain how that the sync works totally offline relying on the Pear Runtime technology.

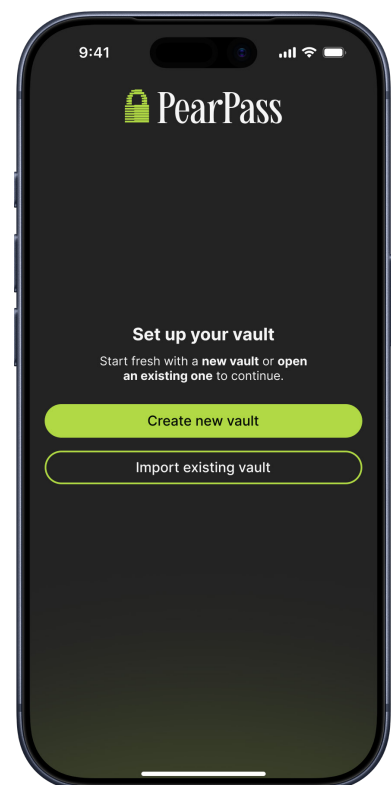
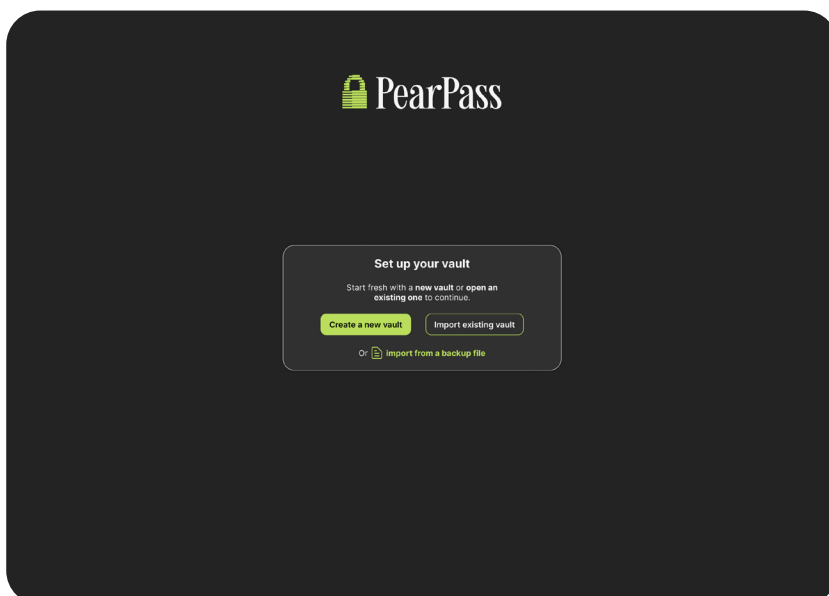


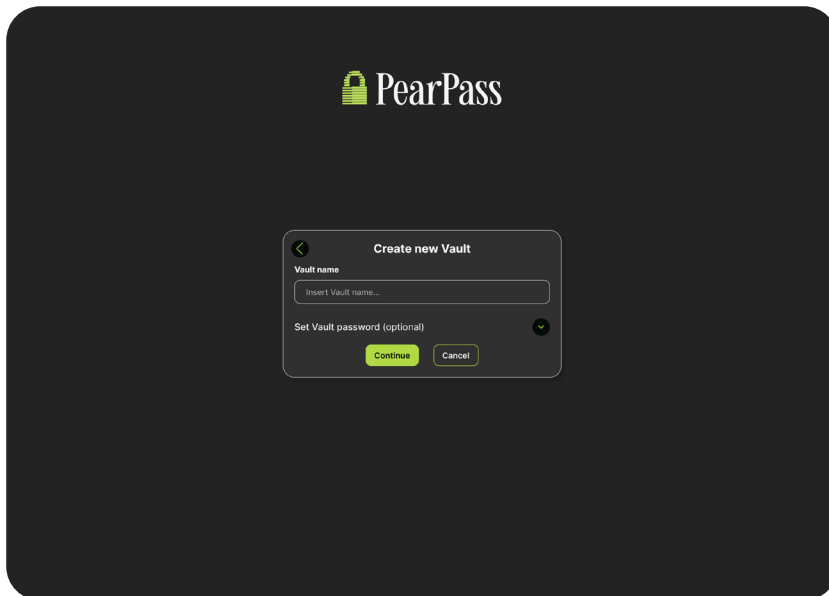


Here the user creates the Master Password and accepts the Terms of Use of the app. It gets warned again to store the Master Password properly.

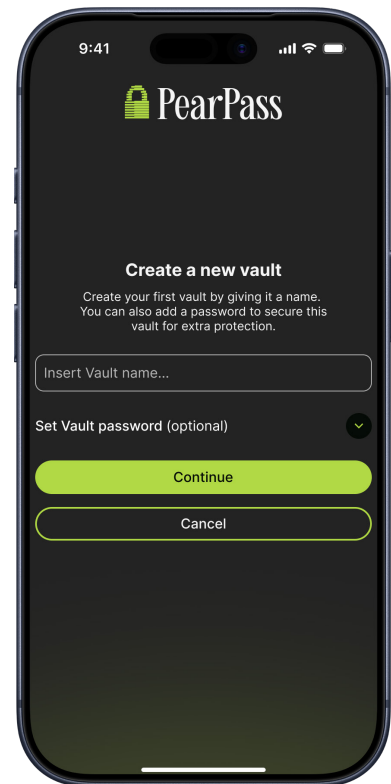


This screen gives the user the possibility to create a new vault, or importing one.

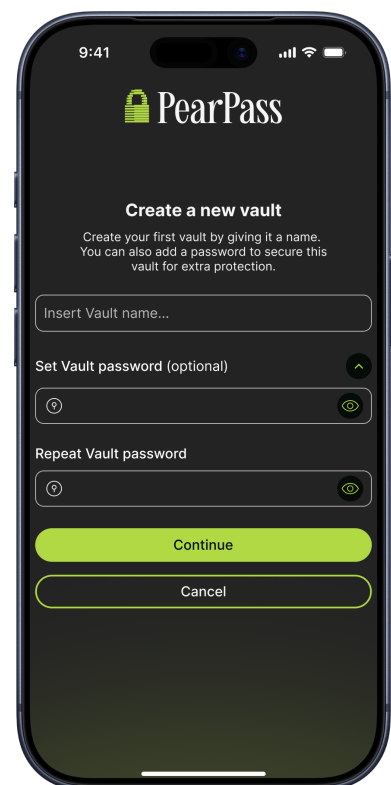
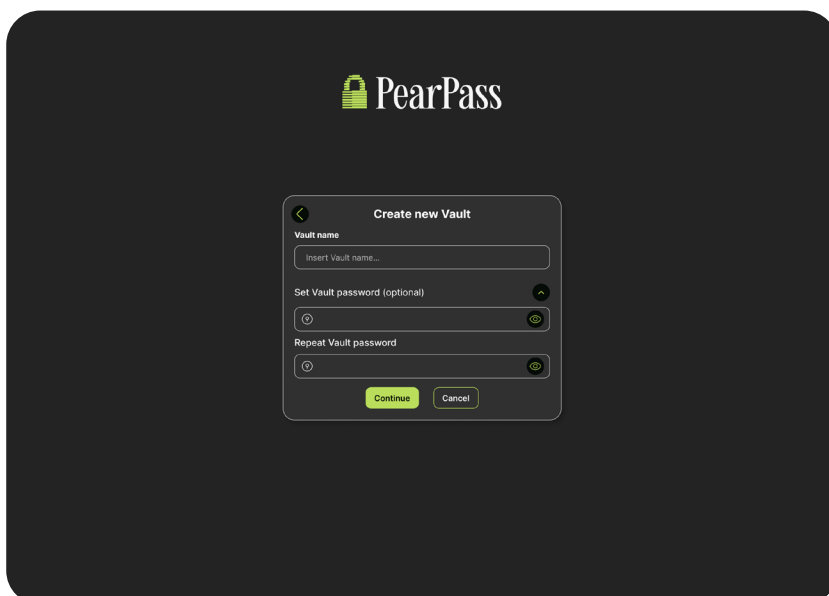




Here the user needs to name the vault he is creating.



The user can choose to create a specific password for the Vault only, in order to gain more security.



Homepage

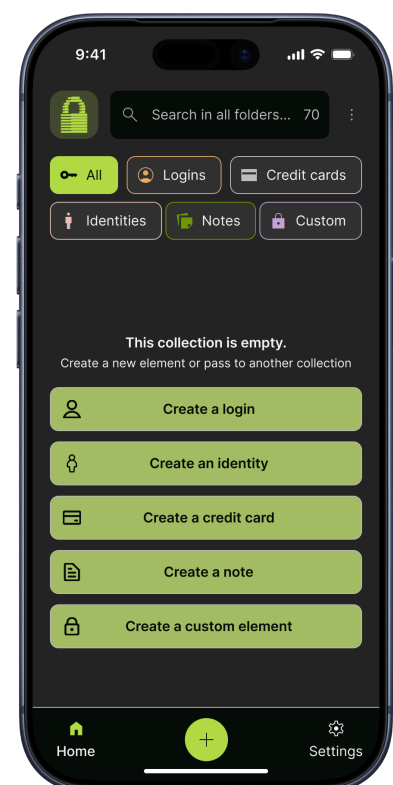
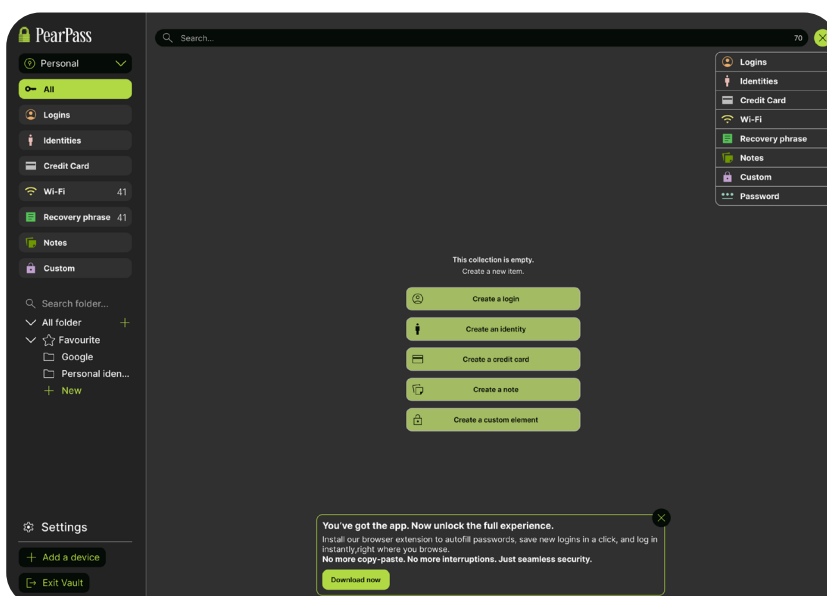
The main dashboard interface of PearPass version 1.0, illustrated in the picture, utilizes a traditional three-panel layout that introduces various layout inconsistencies and interactive breakdowns. First, the chosen color palette lacks a distinct professional identity, the use of gray gives to the whole app a feeling of a wireframe, as if there wasn't a proper study on the UI palette failing to communicate the premium security environment expected of a credential manager.

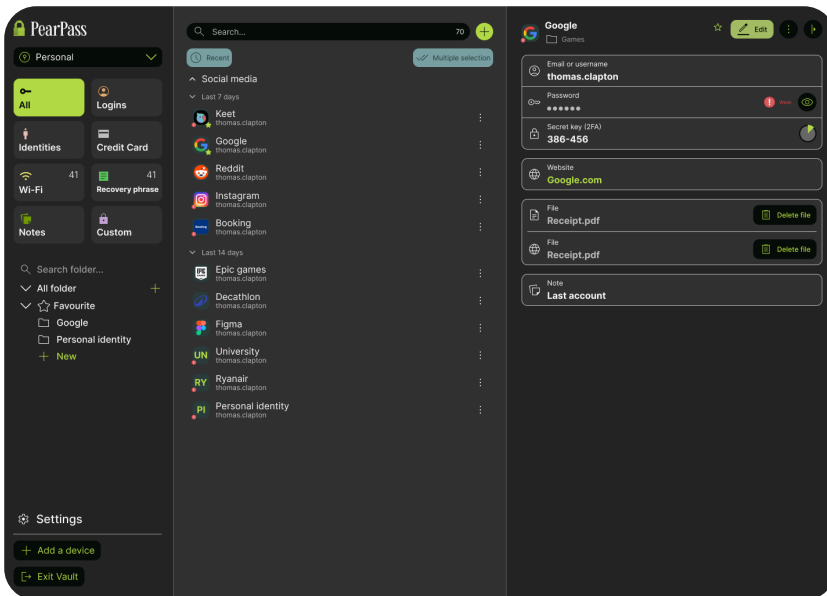
Typography in the whole app is poorly managed; the left sidebar as other elements inside the app displays an excessive use of bold font weights for category items, which flattens the visual hierarchy rather than helping the user navigate.

A significant interaction failure occurs within the item selection state. While the right panel displays the detailed information for the "Google" credential, the corresponding row in the central list provides no active visual feedback or selection styling, creating a mismatch inside the interface.

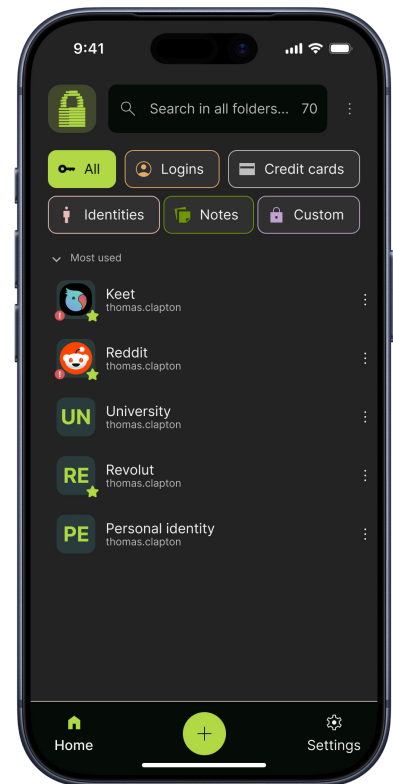
Finally, there are a lot of search bars in different positions, this doesn't help the user to focus on the action he is taking or to create a mental path when he wants to search for something inside the app.

Empty State of the app, the user is prompted to create a new item, there is even a message that suggest to install the browser extension to have a better overall experience.

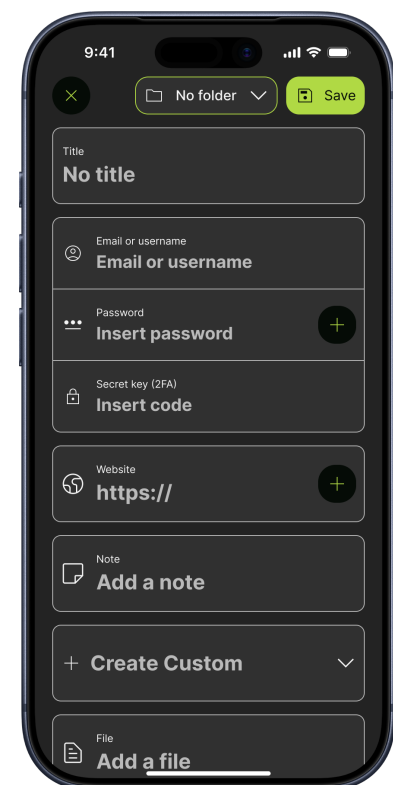
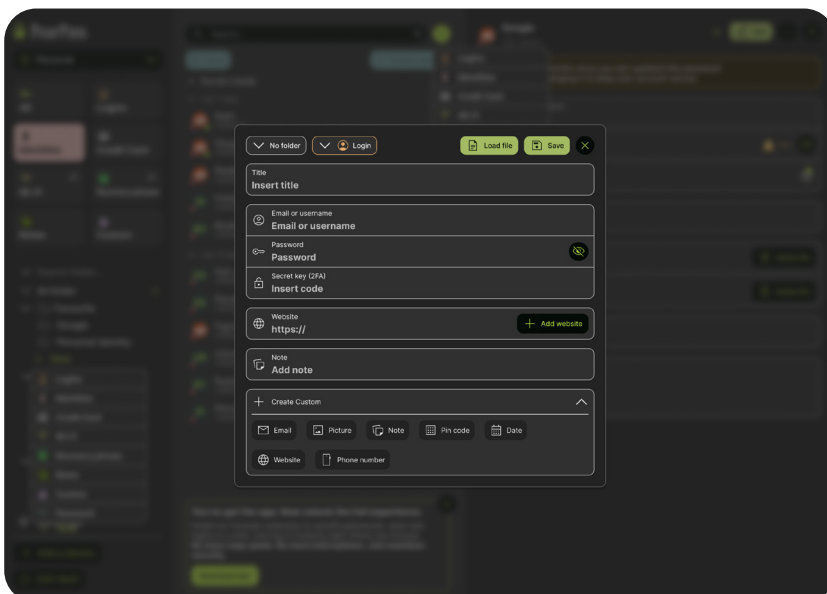




This is the interface with multiple items saved inside, with the list of items based on the creation date of the items itself.



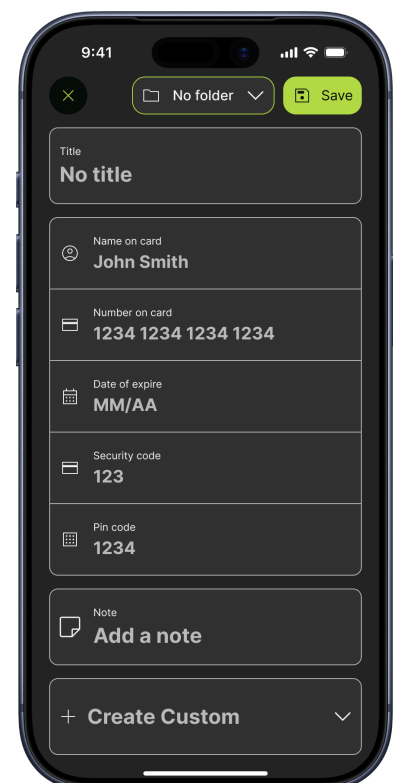
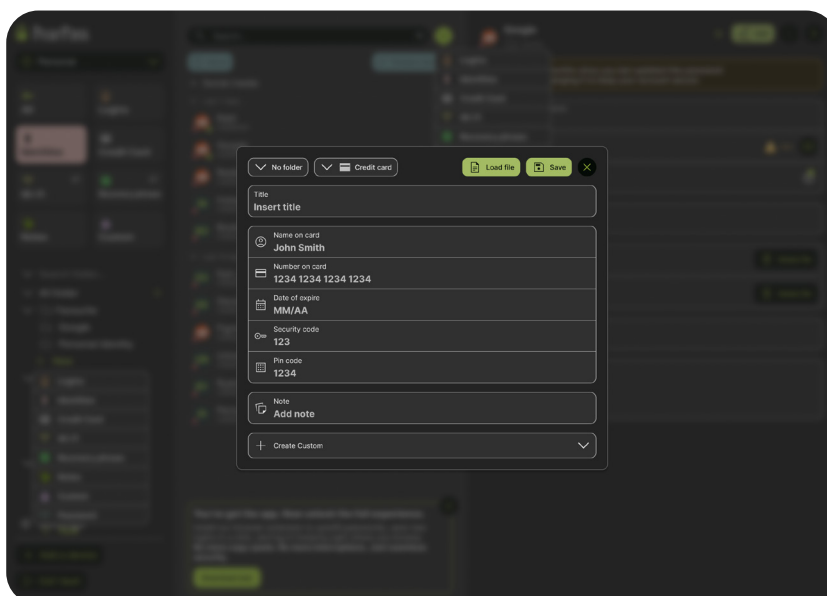
Creation Model for the Login item.



Creation Modal

Inside the interface for credential generation, we have several problems with visual hierarchy. The most critical one is the strict visual parity between the “Load file” and “Save” buttons in the upper-right corner of the modal. Despite their profound functional divergence, where one handles external file upload and the other commits the creation of a new credential, both actions are rendered using identical green-pill call to action. This lack of differentiation creates an immediate risk of accidental interaction, where a user might unintentionally misclick one of the two CTA. This lack of clear affordances is further worsened by typographic uniformity throughout the form inputs. The repetitive use of similar font weights and colors between field labels and placeholder text flattens the layout, forcing the user to expend extra cognitive effort to distinguish empty states from active input areas. By presenting these continuous micro-choices simultaneously, the interface induces visual clutter and decision fatigue, running counter to the principles of streamlined information architecture.

Creation Model of the Credit Card item.

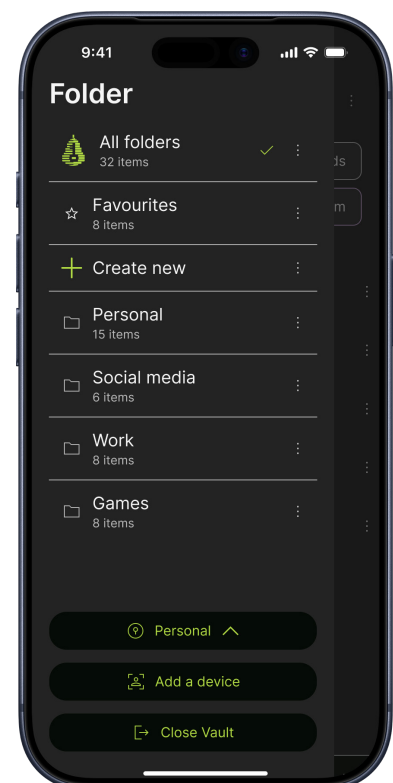
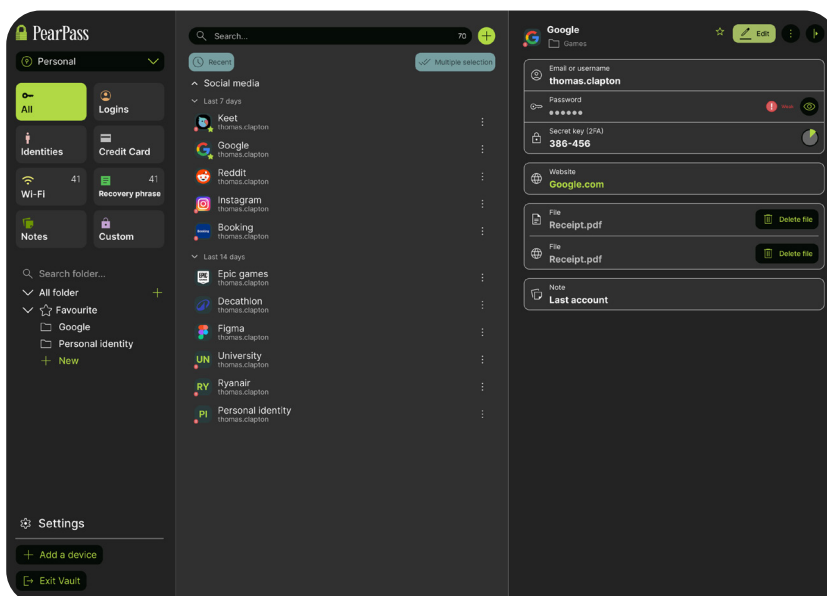


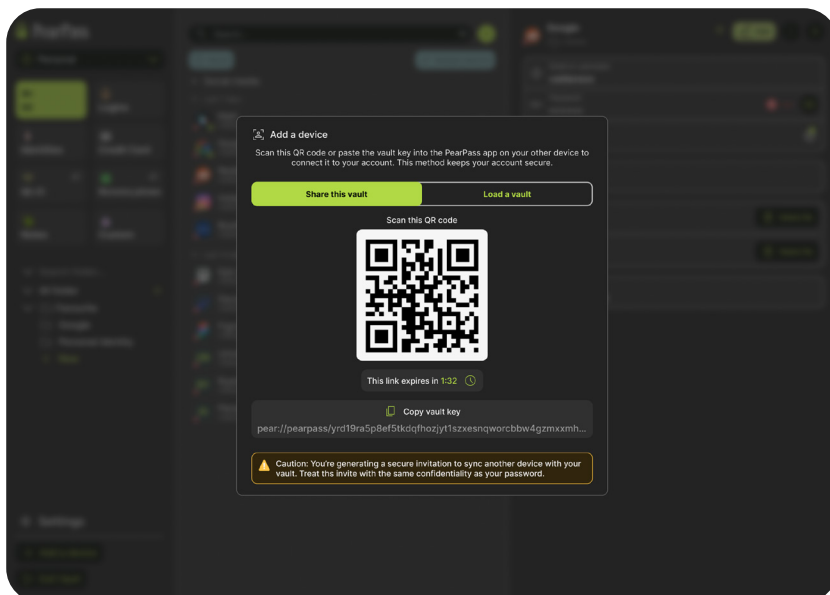
Cross Device Synchronization

The interface mechanism for linking multiple devices, introduces significant cognitive friction and architectural ambiguity. The process begins with the “Add a device” action located in the utility area of the primary navigation sidebar. Upon activation, the resulting modal attempts to house two completely opposite functional behaviors under a single layout. By utilizing a segmented toggle between “Share this vault” and “Load a vault,” the interface forces outbound data hosting and inbound data ingestion to coexist within the same visual container. This configuration causes directional confusion, as the overarching modal title remains “Add a device” even when the user switches to an inbound retrieval state.

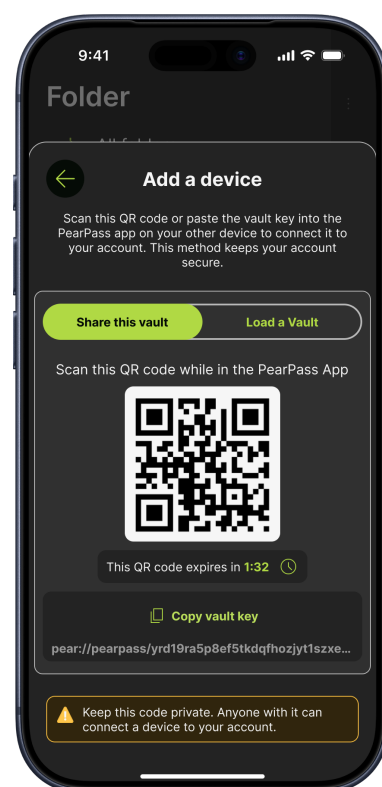
Furthermore, the presentation of information within the sharing panel is heavily overcrowded. The interface simultaneously displays a large QR code, a countdown timer, a raw cryptographic vault key string, and a high-visibility cautionary warning card. This layout density creates visual fatigue and elevates user anxiety. The lack of a clear layout distinction between exporting identity parameters and loading external vaults prevents the user from establishing an accurate mental model of data replication across devices.

This screen shows how to start the Synchronization flow, from the side panel in the desktop app, and from the sidebar (clicking on the lock in the top-left corner)

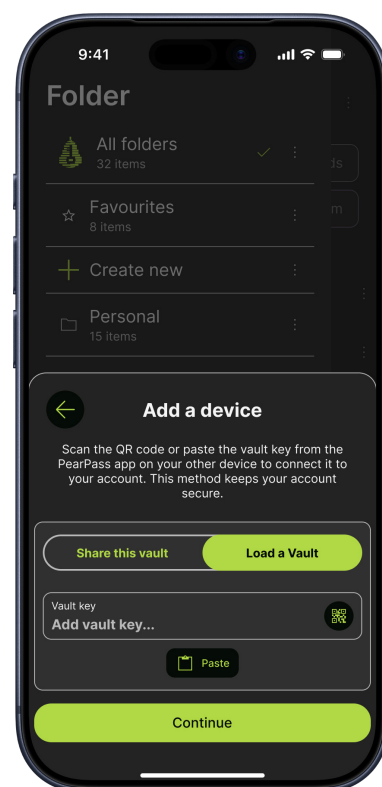
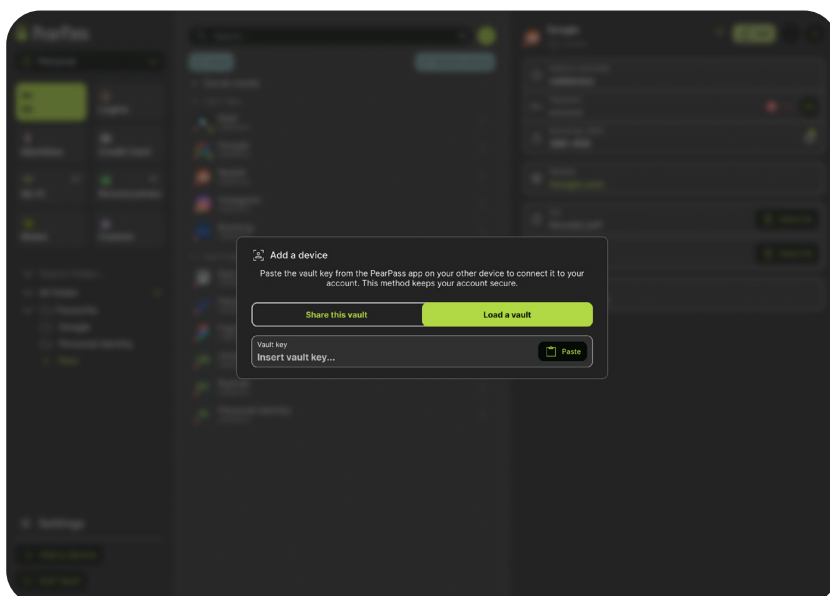




This is the view of the code or QR that gives you are possibility to sync the vault you are in with another device.



In this section instead you can attach the code (Desktop and Mobile) or scan the QR code with your camera (Mobile only)

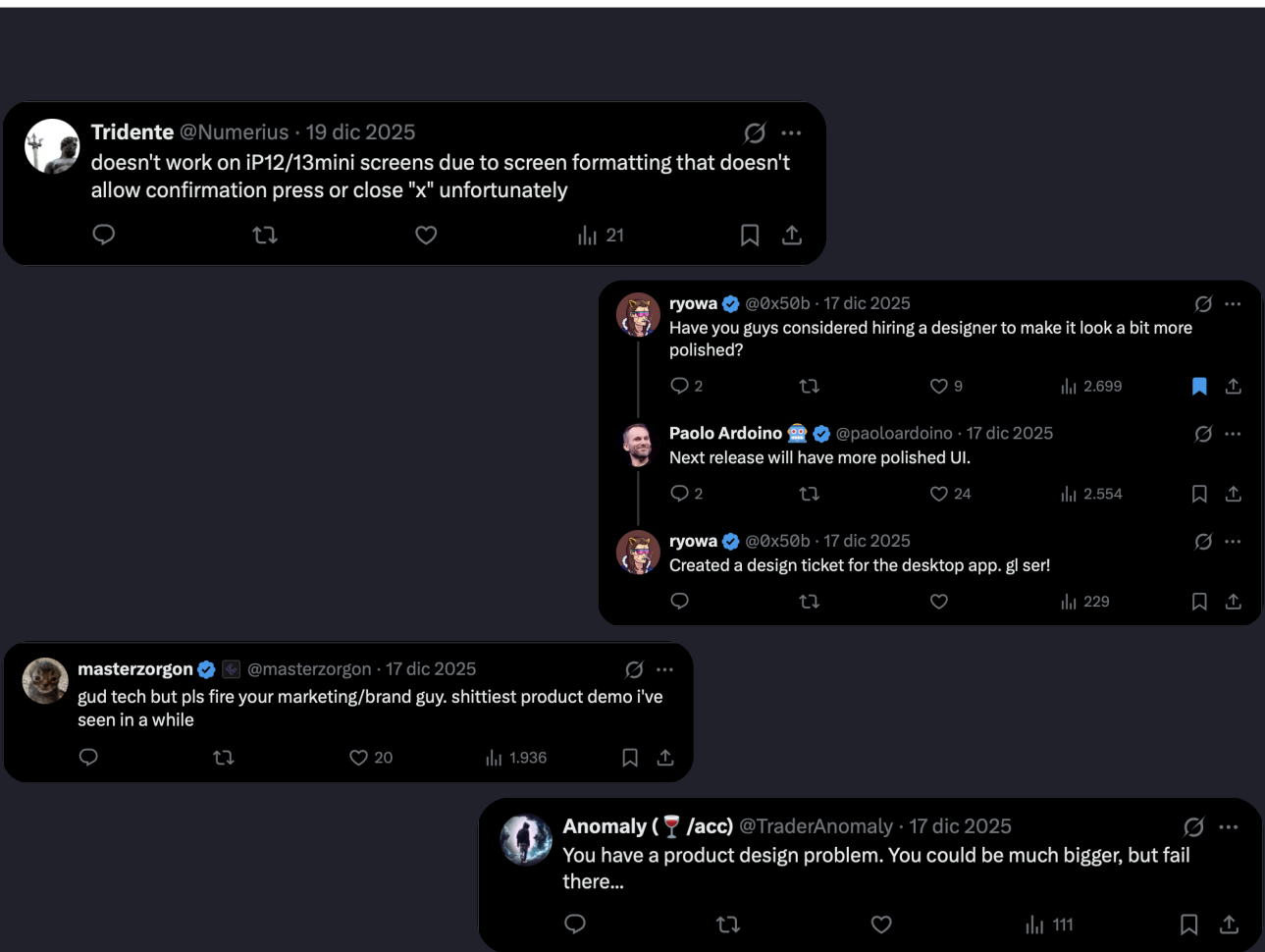


5.4 Organic Community Critique: Analyzing Informal Feedback

To transform unmediated public commentary into actionable product guidelines, this section evaluates the netnographic data harvested from user forums, Twitter/X, and YouTube following the initial deployment of PearPass version 1.0. This empirical corpus exposes a pronounced usability gap. While the foundational value proposition of a local-first, serverless credential manager receives strong conceptual validation, the concrete execution reveals critical friction points. These public data points are categorized into five core socio-technical axes.

1. Ergonomic Refinement and Viewport Responsiveness

A primary barrier to user onboarding stems from a raw, developer-led visual presentation. Early adopters explicitly identified a fundamental “product design problem,” noting that the interface layout lacked the professional authority necessary to compete with established security tools. Beyond aesthetic polish, version 1.0 suffered from rigid visual formatting across compact devices. On ultra-compact displays, such as the iPhone 12 and 13 Mini, layout scaling errors completely broke the information architecture by overlapping or hiding primary interactive elements, rendering confirmation controls and close buttons (“x”) physically inaccessible.



2. Decentralized Distribution and Infrastructure Paradoxes

A major architectural contradiction occurred regarding the application's distribution channels. Forcing a local-first software ecosystem to rely on centralized application markets alienated core privacy-centric demographics. Advanced Android practitioners operating "de-googled" devices flagged the requirement of Google Play Services as a severe vulnerability that compromised data sovereignty. Similarly, Windows users cited native installation requirements through the Microsoft Store as an institutional red flag. This deployment friction extended to open-source environments, where broken software packages, specifically the total failure of the AppImage format to launch on Fedora Linux distributions, completely restricted system access for technical actors.v



Nate Long *priv/acc* @literaryjoe · 18 dic 2025

I'm loving what I'm seeing. Please release a Windows desktop version that doesn't require installation via the Microsoft store. That's a major red flag.



202



Krilsano @krilsano · 21 dic 2025

Doesn't work on de-googled Android, it requires google play services 😞



106



Pears.com @Pears_p2p · 21 dic 2025

APK coming soon



96



Mircea Sava @mirceasava_ · 20 dic 2025

I don't trust any app that needs access to Google Play Store to operate! 🤢



34



3. Interoperability Gaps and Ecosystem Fragmentation

A significant usability barrier identified by the community involves the application's current lack of workflow integration and ecosystem interoperability. As documented in the comment attached, early adopters noted that while PearPass introduces a compelling architecture, it lacks the interoperability with certain platforms or functions. Specifically, the absence of the support for 2FA codes (TOTP) and the lack of the support of Firefox browser. By failing to support these essential touchpoints, the initial release forces users to split their authentication routines across multiple disjointed applications, directly undermining the convenience expected of modern credential managers and halting immediate adoption among technical practitioners.



@boija5746 1 mese fa (modificato)

Hardware key integration for securing the Pears Pass would be a game changer! And the linux app image is not working for me on fedora by any means. would be nice to fix

but greatwork!

Mostra meno



1



Rispondi



@nsarnav 1 mese fa

Keet is already very feature complete, PeerPass looks good but is missing some key features for me (TOTP and a Firefox extension). All in all, interesting and nice projects, something to keep an eye on, but still not quite there for me. Also, both apps on first start/install completely crash Internet (block connectivity) both on Fedora and Windows which is something I have experienced before when torrent apps use too many connections. This needs to be fixed.

Mostra meno



2



Rispondi

4. Cryptographic Custody and Ecosystem Maturity

The qualitative analysis reveals acute psychological anxieties regarding data permanence in an unmediated network. Users transitioning from cloud-centric paradigms felt unprotected by device-to-device replication alone. In lost-smartphone scenarios, the total absence of clear local recovery affordances generated acute panic, prompting demands for traditional cryptographic safety nets like deterministic recovery seeds or phrases. To establish a rigorous root of trust, technical users also requested the integration of external hardware security keys (e.g., YubiKeys) to anchor local vault decryption.



5. Ideological Validation and Security Communications

Despite the identified interaction bottlenecks, the qualitative data reveals a powerful market displacement effect that strongly validates the core serverless paradigm. Driven by recent pricing inflation and critical data vulnerabilities across legacy, cloud-hosted alternatives like 1Password and Bitwarden, users explicitly expressed an active readiness to transition toward decentralized alternatives. Community members robustly endorsed the architectural principle where user parameters “never touch a server,” treating localized encryption as the baseline for contemporary digital safety. However, this ideological alignment remains highly sensitive to visual and communicative trust markers. The deployment of generative AI assets within the framework’s promotional materials introduced an immediate perception of reduced engineering rigor, proving that user confidence in strict cryptographic protocols can be easily undermined by misaligned external brand communications.



@Racc8988 1 mese fa



The issue I have with Keet, is the use of AI in their advertisements and possible AI in other avenues which makes holepunch feel a little less secure

👍 5 🗨

Rispondi

5.4.1 Internal Context

Bridging the gap between user perception and systemic execution requires a critical examination of the internal factors that shaped the version 1.0 architecture. The following breakdowns articulate the technical debt, strategic prioritization, and workflow limitations that underlay the community's primary socio-technical critiques.

Cluster 1: Organizational Gaps and Design Implementation Failures

The design inconsistencies observed, specifically regarding typography, frame construction (e.g., auto-layout, constraint management), and visual hierarchy, originated from an organizational breakdown. The absence of a dedicated Product Manager (PM) and some difficulties from a design perspective left a critical void in mediation between design and development teams, preventing the clear communication of technical requirements and user-centric priorities. This structural deficiency resulted in an implementation where design intent was frequently compromised by unaligned development workflows.

Cluster 2: Evolutionary Distribution Strategy

The project's deployment strategy followed an iterative path from centralized gateways to sovereign accessibility. Initially, PearPass was deployed via official store channels to reach a broad audience; however, the release was accompanied by direct installers, ensuring users could avoid using centralized stores entirely. Critically, the application maintained an open-source architecture from the start, providing a transparent audit trail for all users.

Cluster 3: Prioritization and Technical Debt

Regarding ecosystem integration, development resources were strategically prioritized to support the most widely used browsing environments (Google Chrome) before expanding to further platforms. Features such as 2FA support were intentionally deferred to subsequent releases due to critical time constraints during the V1.0 development cycle, representing a deliberate management of technical debt.

Cluster 4: Cryptographic Custody and Blind Peer Recovery

User anxiety regarding data loss in lost-device scenarios remains a central R&D focus. The team is currently engineering a solution to enable secure account recovery utilizing "blind peer" technology, aimed at balancing cryptographic safety with user accessibility.

Cluster 5: Open Source as a Trust Anchor

The application's fully open-source nature serves as the definitive mechanism for verification. This transparency allows any user to inspect the codebase, mitigating marketing concerns and anchoring trust in verifiable software logic rather than the brand communications.

5.5 Design System

Typography and Palette

The evolutionary transition to PearPass version 2.0 marks a profound shift from a developer-centric interface to a systemic visual framework designed to cultivate user trust. Guided by an internal usability audit and public feedback loops, the design system is documented in structuralized color and typography. It treats them not as mere stylistic embellishments, but as relational vectors that convey safety, intent, and platform legitimacy. Chromatic frameworks are engineered to establish an atmospheric sense of strict data enclosure. By adopting a profound dark-mode matrix rooted in an absolute dark baseline (#080A05) and secondary container boundaries (#15180E). Within this localized space, visual navigation is driven by a vibrant, technical primary green token (#B0D944), used to highlight the most important action inside the interface. The containment system is further reinforced by clear outline thresholds (Border-Primary: #212814), while system exceptions are framed by highly responsive status signals (Surface-Error: #D13B3D). By organizing the visual field through these calibrated constraints, the application has a modern, safe dark aesthetic. Simultaneously, the structural information architecture fixes readability issues through an elegant typographic hierarchy. The implementation introduces a distinctive stylistic tension: title

▽ App Palette

Background

Used as a background for the whole page.

#080A05

Secondary

Used for highlighting different states of components, such as "hover" state.

#BAE061

Border-Primary

Used as a primary outlining color

#212814

Surface-Primary

Used as a secondary background for surfaces like cards, dialogs, items, and so on.

#15180E

Text-Primary

Used for showcasing the primary text.

#FFFFFF

Border-Secondary

Used as a secondary outlining color

#2C3618

Surface-Secondary

Used as a tertiary background for surfaces and components.

##FDCA40

Text-Secondary

Used for showcasing the secondary text.

#BDC3AC

Surface-Error

Used as a error surface color, for example, for icons

#D13B3D

Primary

Used for highlighting different aspects of our apps, such as text, CTA, and so on.

#B0D944

On-Primary

Used for mostly texts, when the "Text-Primary" can't be applied, when the contrast that the background creates isn't enough.

#08090C

Surface-Warning

Used as a warning surface color, for example, for icons

#D7D245

structures use an editorial serif typeface (Humble Nostalgia, 28px) to humanize first-run onboarding rituals and settings nodes, projecting stability and institutional rigor. This expressive layer is systematically paired with a robust sans-serif font family (Inter, 24px and 12px) for general body data and caption text. The crisp, clean scaling properties of the sans-serif font ensure that the interface in general remains clear and scannable even when squeezed into a floating, windowed layout workspace.

▽ App Typography

Display & Titles

First-run onboarding steps, Settings pages, Big empty states. This should feel special. Don't overuse it.

Font-family: Humble Nostalgia
Font-weight: Regular
Font-size: 28px
Line-Height: Auto
Letter-Spacing: 0%

The quick
brown fox...

Body Primary

General text style.

Font-family: Inter
Font-weight: Regular
Font-size: 14px
Line-Height: Auto
Letter-Spacing: 0%

The quick brown fox
jumps over the lazy
dog.

Body Primary Emphasized

Bolder general text style.

Font-family: Inter
Font-weight: Medium
Font-size: 14px
Line-Height: Auto
Letter-Spacing: 0%

The quick brown fox
jumps over the lazy
dog.

Caption & Supporting Texts

Used for smaller text as caption or supporting texts.

Font-family: Inter
Font-weight: Regular
Font-size: 12px
Line-Height: Auto
Letter-Spacing: 0%

The quick brown fox jumps
over the lazy dog.

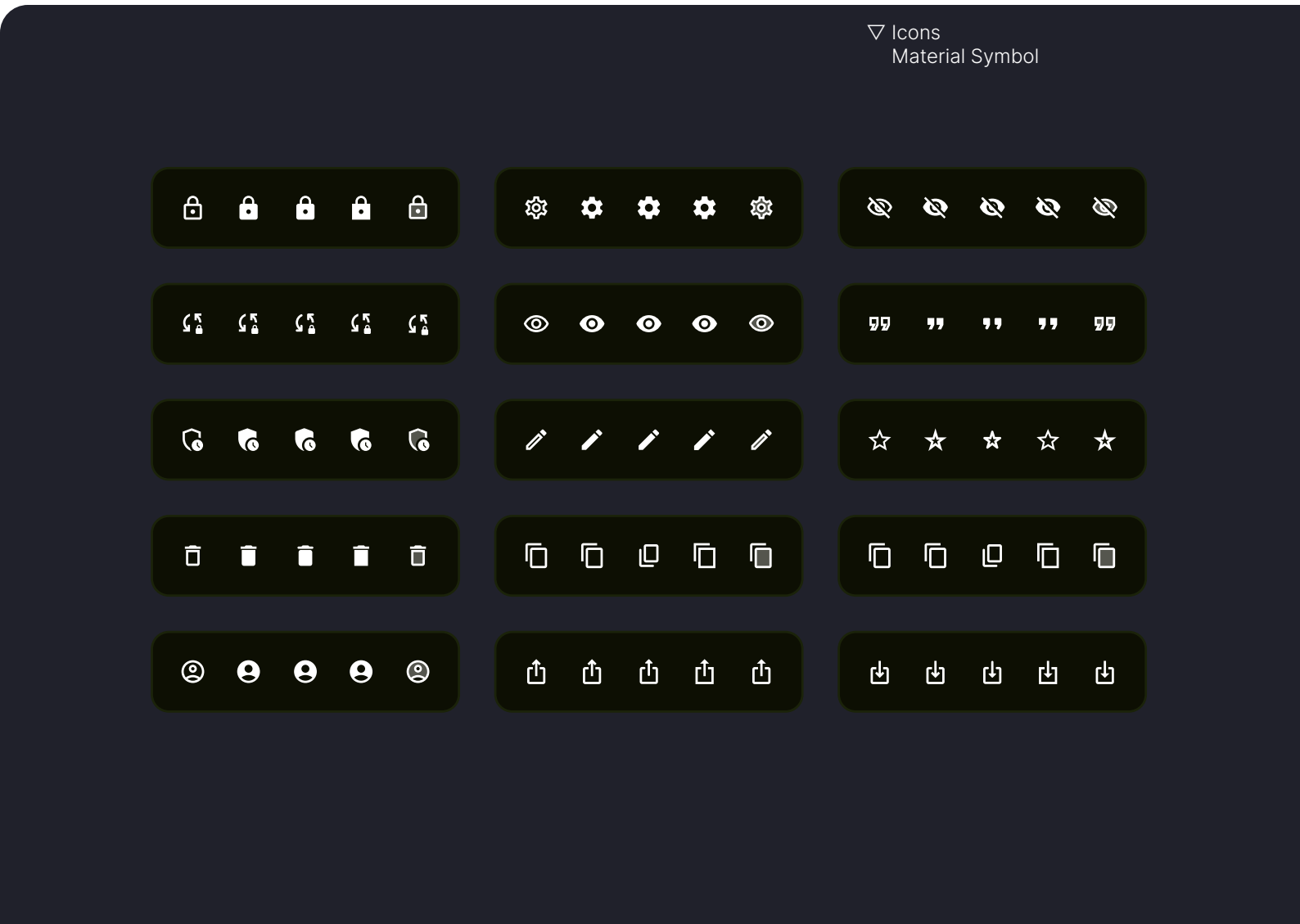
Iconography

Beyond chromatic and typographic foundations, the scalability of the PearPass design system relies heavily on a disciplined, atomic component architecture. To establish a coherent visual vocabulary across different platforms, the application integrates Google’s Material Symbols library as its core iconographic framework. Material Symbols provide a highly structured matrix of clean, geometric visual metaphors. By embedding this standardized library into the component library, the product strategy ensures that any future Pear apps set inherits a proven, recognizable visual ecosystem, optimizing development workflows without losing the eye on the design part.

Button and CTA

Inside of the Design System, we created a multiple Call-to-Action (CTA). To manage user intentionality across these operational boundaries, the design system defines four distinct functional button matrices: Primary, Secondary, Tertiary, and Destructive. While all four categories observe identical internal padding parameters and layout spacing constraints to maintain visual alignment, they use contrasting fill treatments and stroke definitions to reflect their functional weight within the application ecosystem.

- **The Primary Matrix**, leverages a high-contrast visual token, specifically, the vibrant technical green fill paired

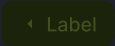
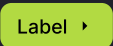
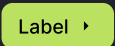
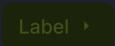


with dark typography, to command immediate attention. This configuration is reserved exclusively for high-priority actions that move the system forward, such as initiating vault creation routines or final database commitments.

- **The Secondary Matrix**, employs a subtle, outlined stroke composition enclosing a muted dark surface container, mapping out supportive operational pathways such as importing data.
- **The Tertiary Matrix**, utilizes a transparent, borderless architecture to handle low-emphasis utility adjustments or optional data parameters, keeping the interface clean and preventing choice overload.
- **The Destructive Matrix**, introduces a high-visibility red status warning fill. This visual indicator establishes a deliberate psychological friction point, clearly signaling actions that cause irreversible data deletions or irreversible changes.

Each macro-matrix is engineered to scale across precise dimensional states (Small and Medium) and adapts dynamically across four interactive feedback states: Passive, Hover, Pressed, and Disabled. By organizing these states alongside flexible layouts, including text-only, leading icon, trailing icon, and icon-only variants, the design system creates a predictable and highly responsive interface grid that minimizes human operational errors across all platform deployments.

▽ Primary CTA

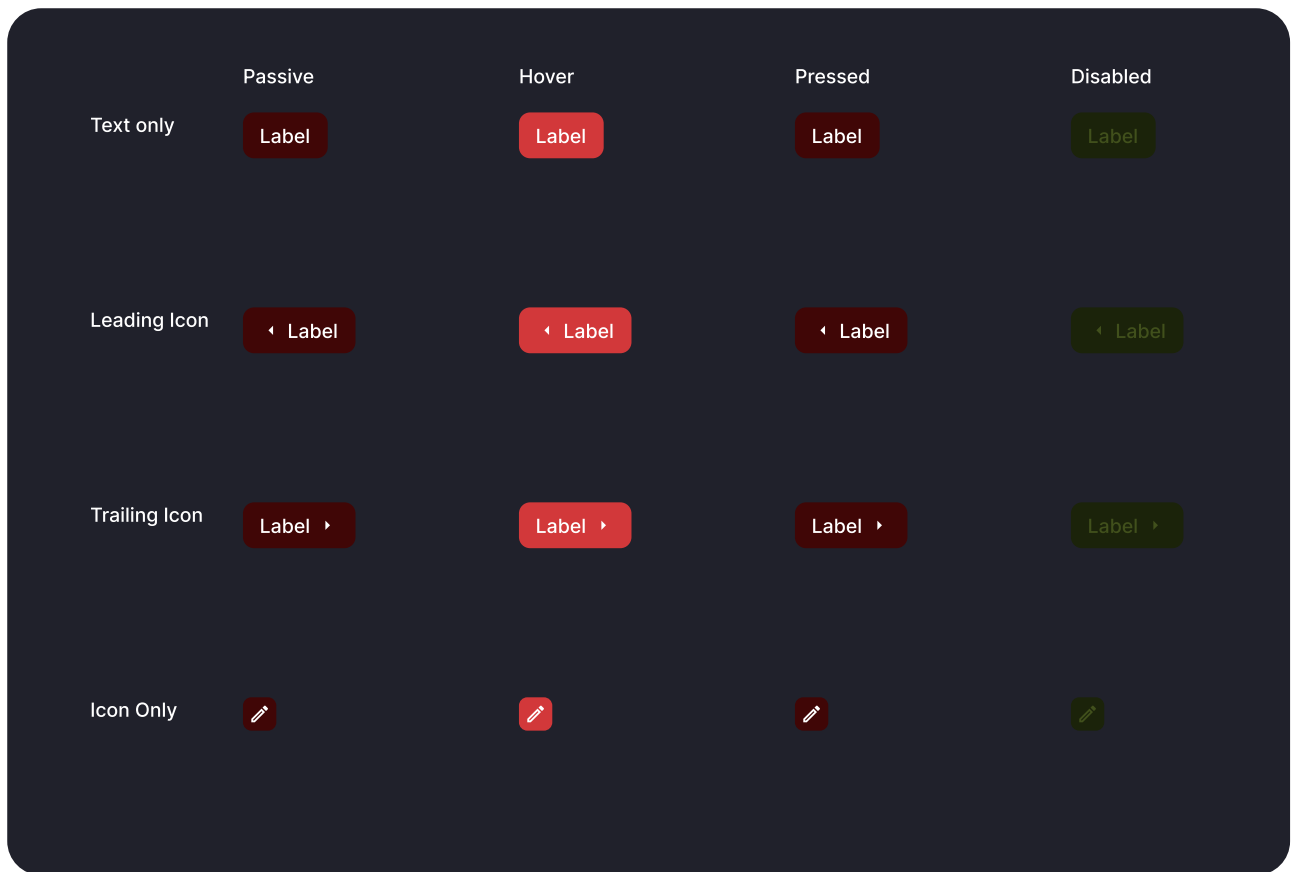
	Passive	Hover	Pressed	Disabled
Text only				
Leading Icon				
Trailing Icon				
Icon Only				

	Passive	Hover	Pressed	Disabled
Text only				
Leading Icon				
Trailing Icon				
Icon Only				

△ Secondary CTA

	Passive	Hover	Pressed	Disabled
Text only				
Leading Icon				
Trailing Icon				
Icon Only				

▽ Tertiary CTA



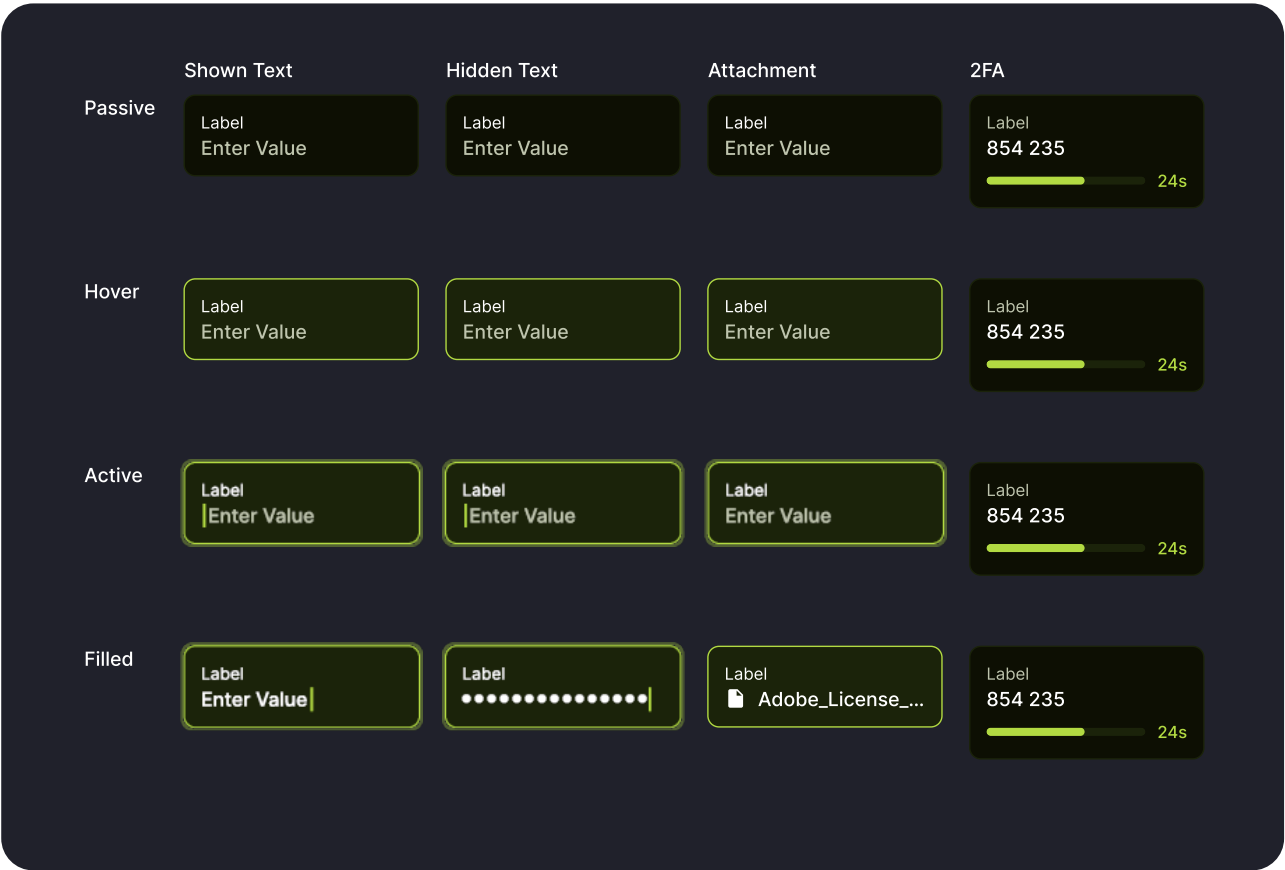
△ Destructive CTA

Input Field

The structural formulation of input field components within the PearPass design system acts as the definitive boundary for user data entry. Every input element adapts across six distinct interactive and operational states: Passive, Hover, Active, Filled, Rest, and Error.

This state architecture operates symmetrically across four fundamental content classifications, Shown Text, Hidden Text, Attachment, and 2FA, ensuring a predictable behavior and visually for every need and situation. When a field becomes Active, the layout introduces a high-visibility green token perimeter highlight and an active cursor, establishing a clear visual zone to input data. On the contrary, when the node encounters processing friction or empty requirements, the container transitions to an Error state, deploying a high-visibility red border paired with inline, contextual error text positioned directly beneath the field box. This immediate validation loop gives the user the possibility to understand clearly what the interface is trying to tell him.

▽ Input Field



	Shown Text	Hidden Text	Attachment	2FA
Rest	Label Enter Value	Label ●●●●●●●●●●	Label 📎 Adobe_License_I...	Label 854 235 24s
Error	Label Enter Value ⓘ Intended error message goes here	Label ●●●●●●●●●● ⓘ Intended error message goes here	Label 📎 Adobe_License_I... ⓘ Intended error message goes here	Label 854 235 24s ⓘ Intended error message goes here

△ Input Field

▽ Input Field with action

	Shown Text	Hidden Text	Attachment	2FA
Action	Label Enter V...   	Label Enter V...   	Label Enter V...   	Label 854 235 24s
Leading Icon	 Label Enter Value	 Label Enter Value	 Label Enter Value	Label 854 235 24s
Password Indicator	Label Enter...  Strong 	Label Enter...  Strong 	Label Enter...  Strong 	Label 854 235 24s

Dialogue

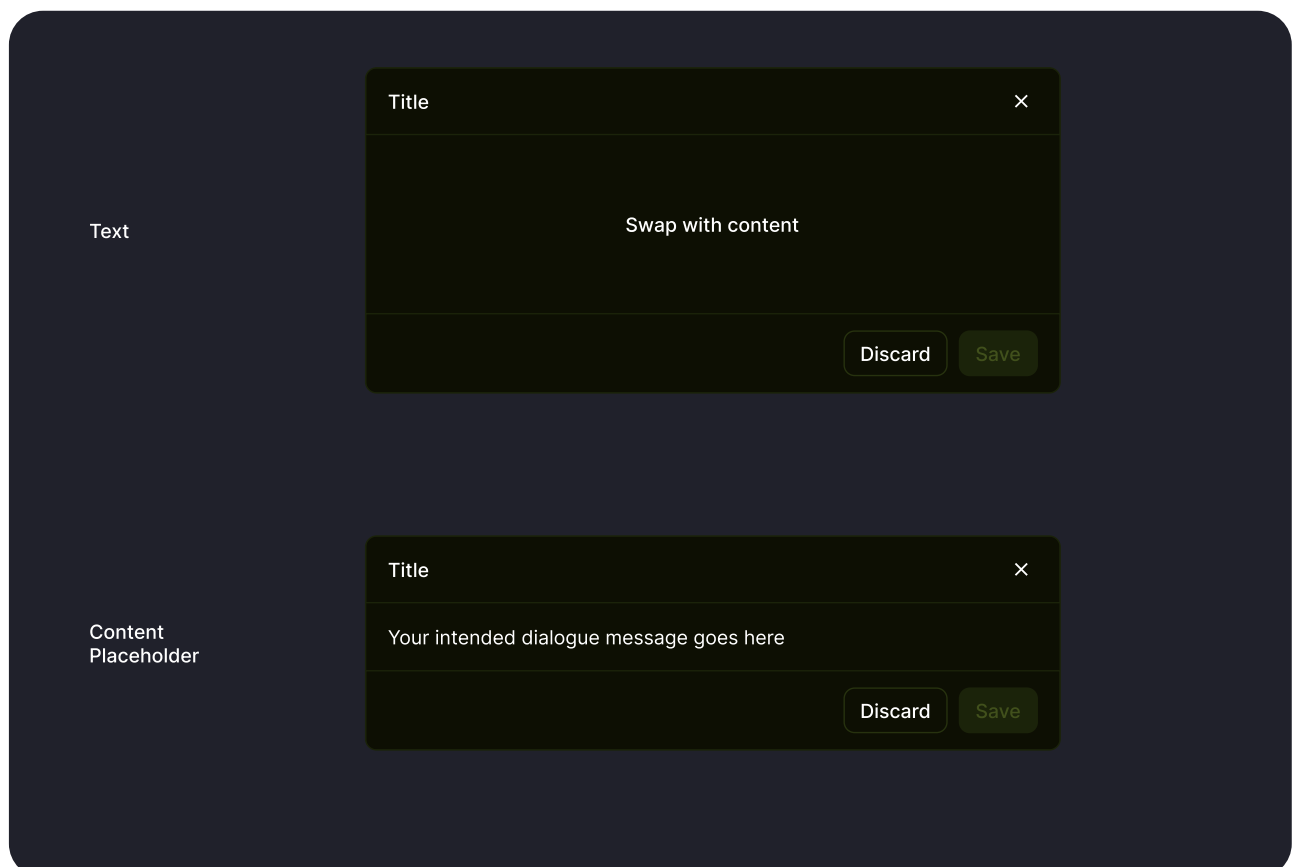
Another fundamental element Inside the Design System are the contextual dialogue overlays; they actively suspend the background workspace interaction loop to isolate user validation. These overlays are split into two core structural archetypes, Text-Driven Dialogues and Content Placeholders, which scale across responsive viewports to ensure strict cross-platform behavioral consistency.

The spatial topology of these components gracefully adapts to desktop and mobile form factors without fracturing the application's visual taxonomy:

- On Desktop Viewports, dialogues are rendered as elevated, centered canvas modals that focus user attention by dimming the surrounding system dashboard.
- On Mobile Interfaces, the components transform into bottom-anchored sheets or drawers, optimizing ergonomic thumb reachability while maintaining identical control hierarchies.

To manage developmental scaling and avoid component fragmentation, the Content Placeholder matrix utilizes a programmatic architecture. This architecture keeps the persistent outer shell of the dialog, including the layout title, close markers, and two bottom-anchored action CTA (Primary and Secondary), while leaving the central space entirely flexible.

▽ Dialogue modal



This slot-based approach allows product teams to swap complex layouts or contextual notifications freely into the component frame without detaching the core component instance. The resulting layout balances rigid visual boundaries with absolute functional versatility.

Dropdown

PearPass component library establishes a responsive dropdown selector. The system divides these overlay pickers into two functional tiers, Simple and Complex Dropdowns, engineered to adapt across desktop popover anchors and mobile bottom-sheet modules.

- The Simple Dropdown Architecture handles linear, scalar configuration parameters that work with simple actions, such as clipboard clearing intervals. On desktop viewports, it expands as a floating box aligned directly with the trigger text field. On mobile layouts, the interaction automatically transitions into an elevated bottom sheet titled "Clear Clipboard," using human thumb ergonomics to prevent data truncation on narrow screens.
- The Complex Dropdown Architecture functions as an advanced data-filtering component designed to handle intricate actions, such as choosing or switching specific vault account credentials. To mitigate visual fatigue and shorten data access pathways, this variation encapsulates a persistent search input module directly inside the header boundary of the expanded list. This allows for immediate local database search.

Ultimately, these components guarantee that whether the interface is handling simple, scalar configuration variables or rendering high-density database registries, the dropdown component maintains absolute structural unity. This approach ensures that data scannability, interactive boundaries, and structural clarity are perfectly preserved across all target form factors, anchoring the application's visual taxonomy regardless of the operational context.

Not expanded

1 Minute ▾

Expanded Dropdown

1 Minute ▾

1 Minute
3 Minutes
5 Minutes
10 Minutes
30 Minutes
1 Hour
3 Hours
Never

△ Simple Dropdown
Desktop

▽ Complex Dropdown
Desktop

Not expanded

 Microsoft 365
simon.j@gmail.com ▾

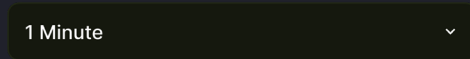
Expanded Dropdown

 Microsoft 365
simon.j@gmail.com ▾

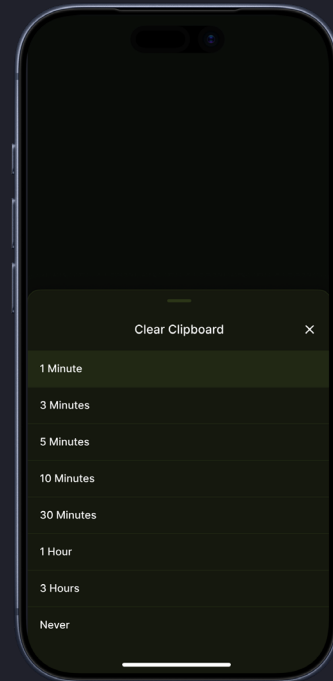
Search...

 Microsoft 365 simon.j@gmail.com	
 Slack acme.1994@gmail.com	
 GitHub acme.1994@gmail.com	
 Adobe simon.j@gmail.com	
 Amazon acme.co14@outlook.com	

Not expanded



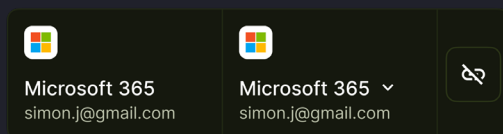
Expanded Dropdown



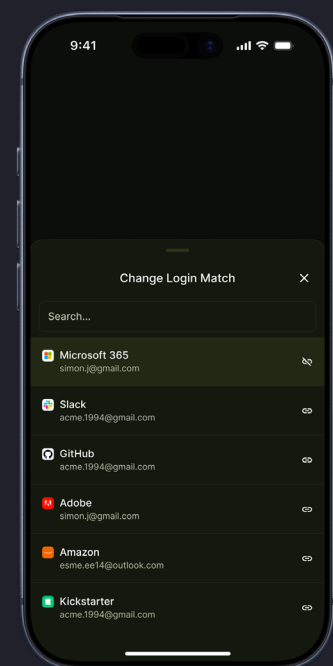
△ Simple Dropdown
Mobile

▽ Complex Dropdown
Mobile

Not expanded



Expanded Dropdown



Snackbar

To facilitate real-time operational information, the PearPass design system incorporates a lightweight, transient notification architecture via Snackbar components. These components act as micro-feedback loops that directly validate localized actions, such as duplicating sensitive credential strings or encryption keys to the device's native clipboard registry. The snackbar is designed as a high-contrast, pill-shaped overlay container designed to capture peripheral user attention during specific actions.

The snackbar presents two different configurations:

- **Leading Icon** one embeds a clear material symbol icon that immediately preceding the validation text, establishing a spatial semiotic anchor that accelerates readability and confirms the specific nature of the operation.
- **Without a Leading Icon** adopts a highly minimalist textual layout, prioritizing low distraction for repetitive everyday interaction sequences.

By utilizing a bright, high-contrast surface container against the application's signature dark-mode canvas matrix, the design system ensures instant glanceability.

▽ SnackBar

Leading Icon

No Leading Icon

 Copied to Clipboard

Copied to Clipboard

Context Menu

To regulate workflow navigation and enforce data categorization at the entry gate, the PearPass component directory implements a family of **Context Menus**. These popover overlays are split into specific behavioral modules, a specific one to Add Item and one more generic, which transform their physical layouts to preserve absolute interface consistency across desktop and mobile devices.

This structural adaptability is managed via distinct, platform-specific spatial arrangements:

- **On Desktop Interfaces**, the components manifest as floating, pixel-anchored popover blocks. When invoking the primary “+ Add Item” action, the menu expands downward to reveal a clean vertical grid where the user can find all the form of items thought for PearPass (Logins, Credit Cards, Identities, Notes, Recovery Phrases, Wi-Fi, Passwords, and Authenticator Codes). Instead, the generic filter icon deploys a localized sorting picker (Title A-Z, Last Used, Date Added)
- **On Mobile Environments**, to avoid layout fragmentation and maintain semantic alignment with dialogue sheets and dropdown modules, the menu morphs into an ergonomic **Bottom Sheet**. These dark, elevated drawers slide upward from the base of the viewport, stacking option rows neatly within natural thumb-reach parameters.

▽ Add Item Menu
Desktop

Not expanded

+ Add Item

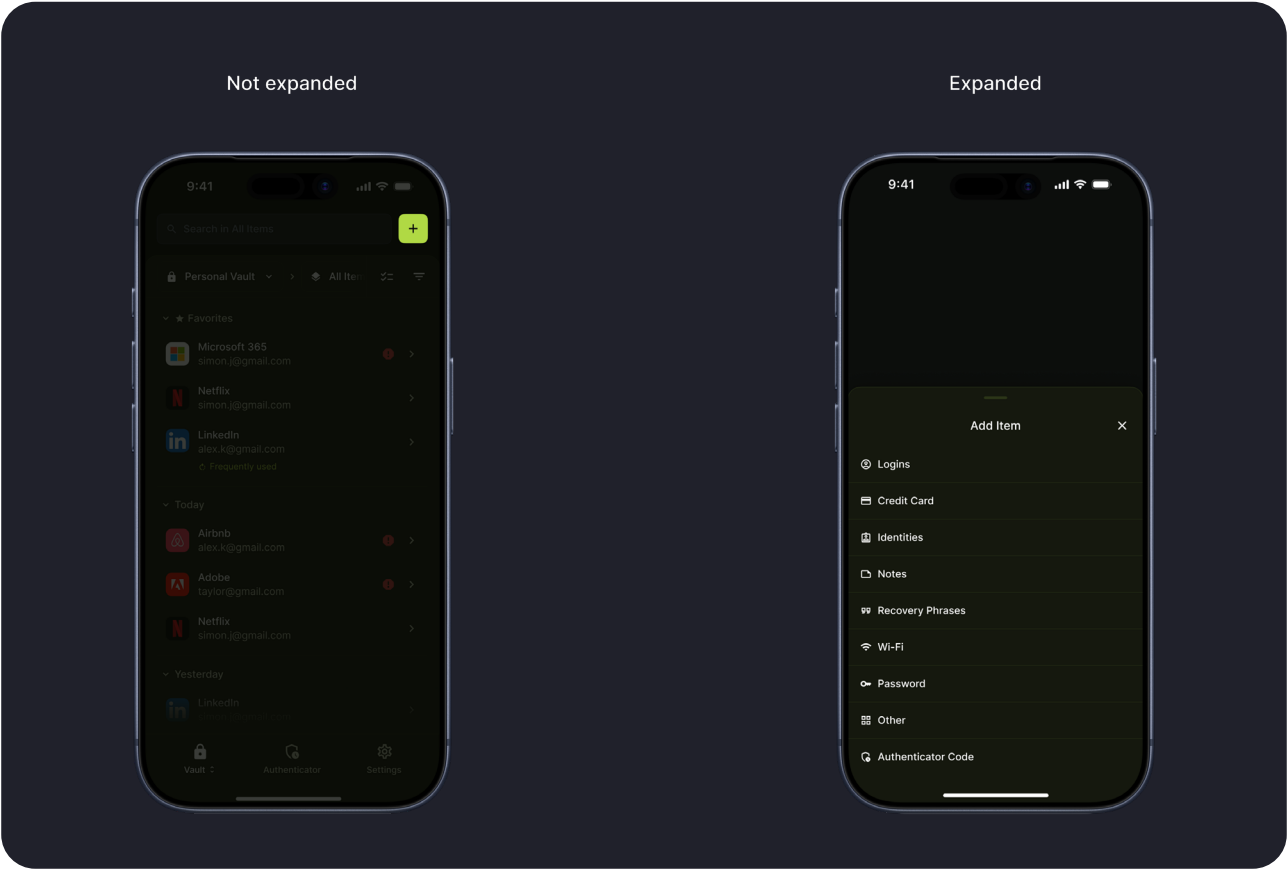
Expanded

+ Add Item

🔑 Logins
💳 Credit Card
👤 Identities
📄 Notes
🗑️ Recovery Phrases
📶 Wi-Fi
🔑 Password
🗄️ Other
🔑 Authenticator Code

By utilizing this unified approach to layout transformation, the design system mitigates cognitive friction and visual disorientation.

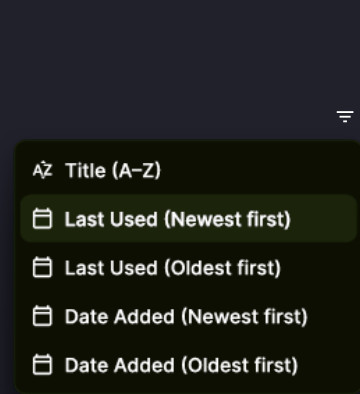
▽ Add Item Menu
Mobile



Not expanded



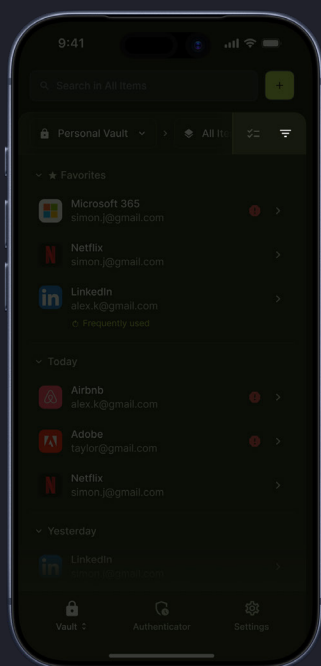
Expanded



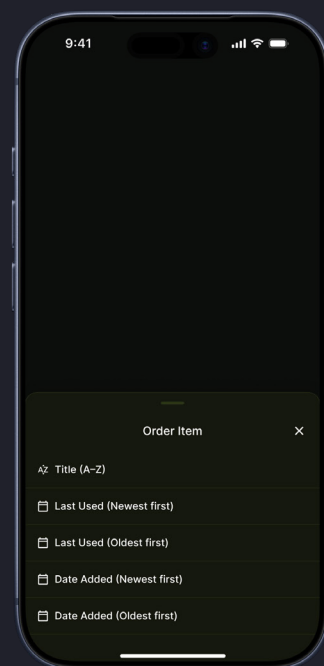
△ Filter Menu
Desktop

▽ Filter Menu
Mobile

Not expanded



Expanded



Alert Message

PearPass design system implements a responsive ecosystem of Alert Messages. These notifications are engineered across a rigorous dimensional architecture, having Big, Medium, and Small form factors, allowing them to scale dynamically depending on their spatial placement within the dashboard layout or credential detail panes.

This dimensional matrix balances semantic color coding with distinct functional layout adaptations to manage user awareness precisely:

- **The Big Variant** is designed for maximum exposure within prominent data canvases. It maps threats into two semantic categories: a dark, green-tinted Alert container and a high-contrast red Warning container, both displaying complete, un-truncated string metadata detailing local data leaks (e.g., "Compromised Password!").
- **The Medium Variant** optimizes layout density for windowed layouts or sidebar overlays. To prevent layout overflowing, it introduces defensive text truncation and introduces an integrated inline Text CTA component (e.g., "View Backup File"), creating a direct interactive pathway to resolve the system friction.
- **The Small Variant** scales down to ultra-compact workspace

▽ Alert Message
Big

Alert



Compromised Password!

This password has been detected in a data leak and should be changed immediately.

Warning



Compromised Password!

This password has been detected in a data leak and should be changed immediately.

modules, replacing the circular exclamation mark icon with a minimalist warning triangle glyph while embedding the same action triggers within a single compressed row layout.

By decoupling these components into strict dimensional variables, the design system provides development teams with an adaptable interface toolkit, ensuring that critical risk communication remains clear, responsive, and perfectly integrated into any platform viewport.

▽ Alert Message
Medium

Alert



Compromised Password!

This password has been detected in a data leak an...

Warning



Compromised Password!

This password has been detected in a data leak an...

Text CTA



Compromised Password!

This password has been detecte...

[View Backup File](#)

▽ Alert Message
Small

Alert

⚠ This password has been detected in a data leak and should be changed immediately.

Warning

⚠ This password has been detected in a data leak and should be changed immediately.

Text CTA

⚠ This password has been detected in a data leak and should be changed immediately. [View Backup File](#)

Breadcrumbs

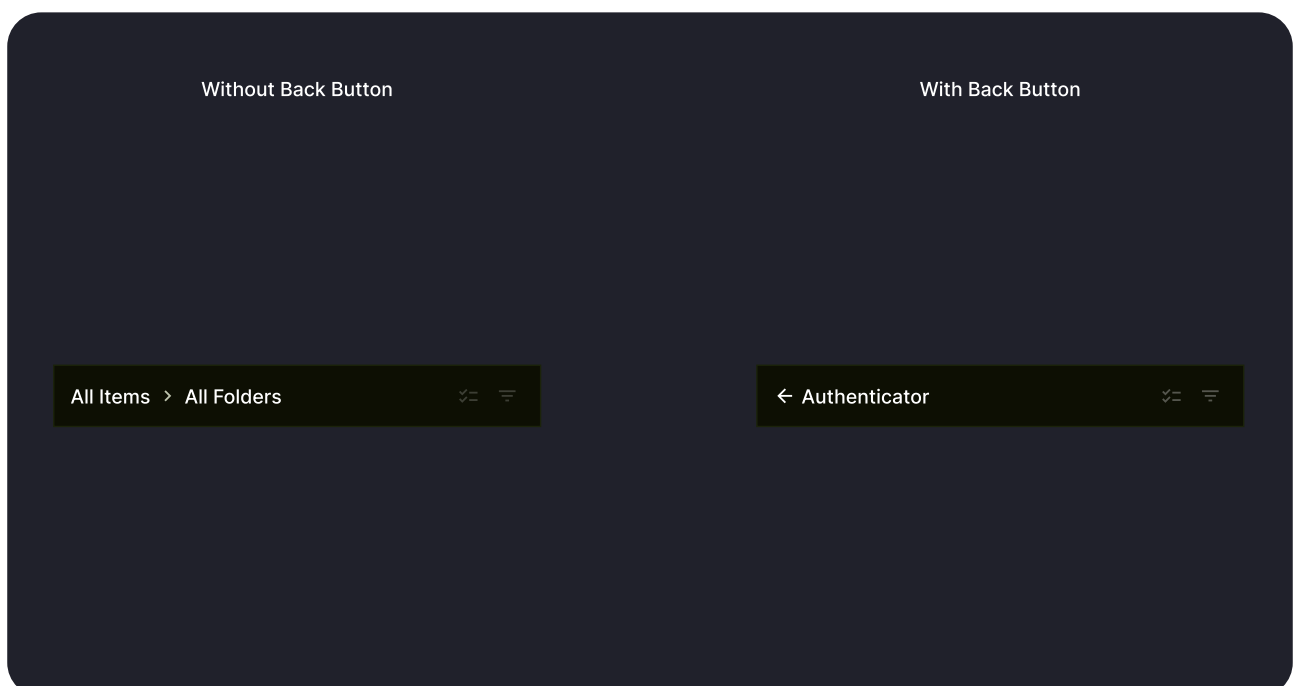
To preserve spatial orientation and ensure predictable navigation, the PearPass design system introduces an adaptive, dual-tier Breadcrumb infrastructure. Within a workspace, where data structures are partitioned into local vaults and folders directories, the user interface must act as a precise map of the current view-state. The application handles this requirement by deploying two asymmetric directional components: the Without Back Button linear trail and the With Back Button context-snapping header.

These two operational archetypes change based on the page the user is on:

- **The Linear Trail Layout (Without Back Button)** is utilized when navigating nested data hierarchies, displaying a clear progression from root categories to deep folders (e.g., All Items > All Folders). This layout visualizes relational database paths directly on the user interface, allowing him to understand the architectural location of their credentials.
- **The Stack-Based Header Layout (With Back Button)** is automatically invoked when entering isolated operational environments, such as the Authenticator module. Rather than displaying an extended linear tree, the interface focuses attention on the active section title and integrates a prominent left-facing chevron vector.

Both variations integrate secondary control tokens on the right axis, specifically, multi-select and sorting glyphs, unifying global directory layout rules. This structural encapsulation balances spatial flexibility with functional continuity, providing a reliable navigational anchor across all application workspaces.

▽ BreadCrumbs



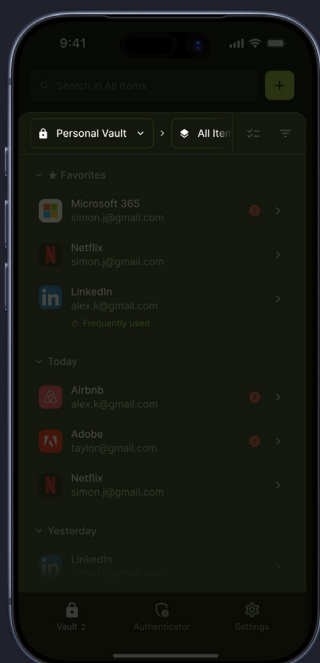
Horizontal Mobile Menu

To transition the comprehensive three-panel desktop dashboard into a highly restricted mobile viewport without fracturing the application's information architecture, the design system implements a fluid, space-optimized navigation paradigm. The mobile layout introduces a **horizontally scrollable segmented container** at the upper threshold of the main viewport (Homepage). This carousel aggregates top-level structural filters, specifically housing contextual triggers for active vault selection, item categories, and specific folder paths. By allowing users to swipe laterally through these control points, the interface maximizes interactive space while maintaining instant discoverability of data parameters.

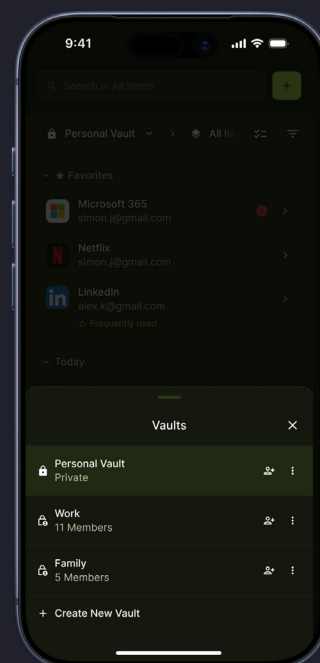
- **The Vault Selector:** Activating the primary vault token instantly deploys an elevated, bottom-anchored control sheet. This component explicitly details various Vaults of the user and gives the possibility to create a new one.
- **The Type and Folder Selector:** Tapping the subsequent interactive pills invokes dedicated, vertically stacked list drawers. These drawers handle granular database queries, such as partitioning entries into distinct identity types (Logins, Credit Cards, Identities) or custom personal directories, while keeping the background dashboard present but dimmed.

▽ Horizontal Bar Vault

Horizontal Bar



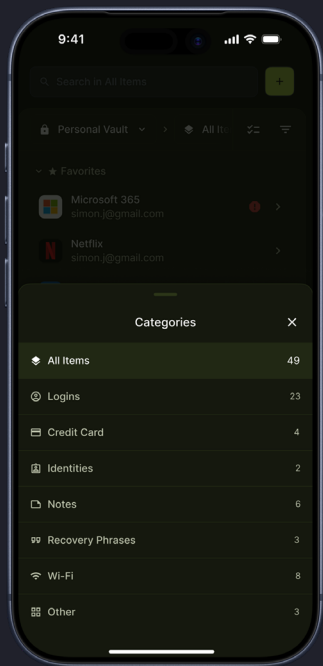
Vault Menu



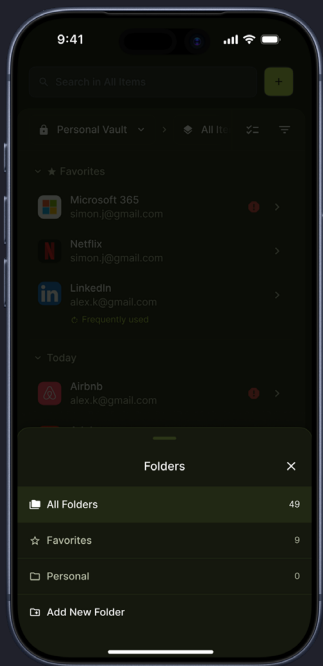
This layout choice guarantees that all structural configuration layers remain contained within a single, highly coherent mobile screen space, enforcing absolute clarity of state during local database filtration.

▽ Horizontal Bar
Categories-Folders

Categories Menu



Folders Menu



Motion Design

The structural deployment of functional micro-animations within PearPass Version 2.0 represents a deliberate shift toward emotional design paradigms, aimed at optimizing behavioral stickiness and long-term user retention. While our competitors prioritize clinical, static layouts that isolate emotional design, Version 2.0 treats aesthetic expression as a dynamic tool for digital attachment. By transforming subtle design details into interactive micro-rewards, the application adopts modern engagement methodologies popularized by industry benchmarks like Duolingo.

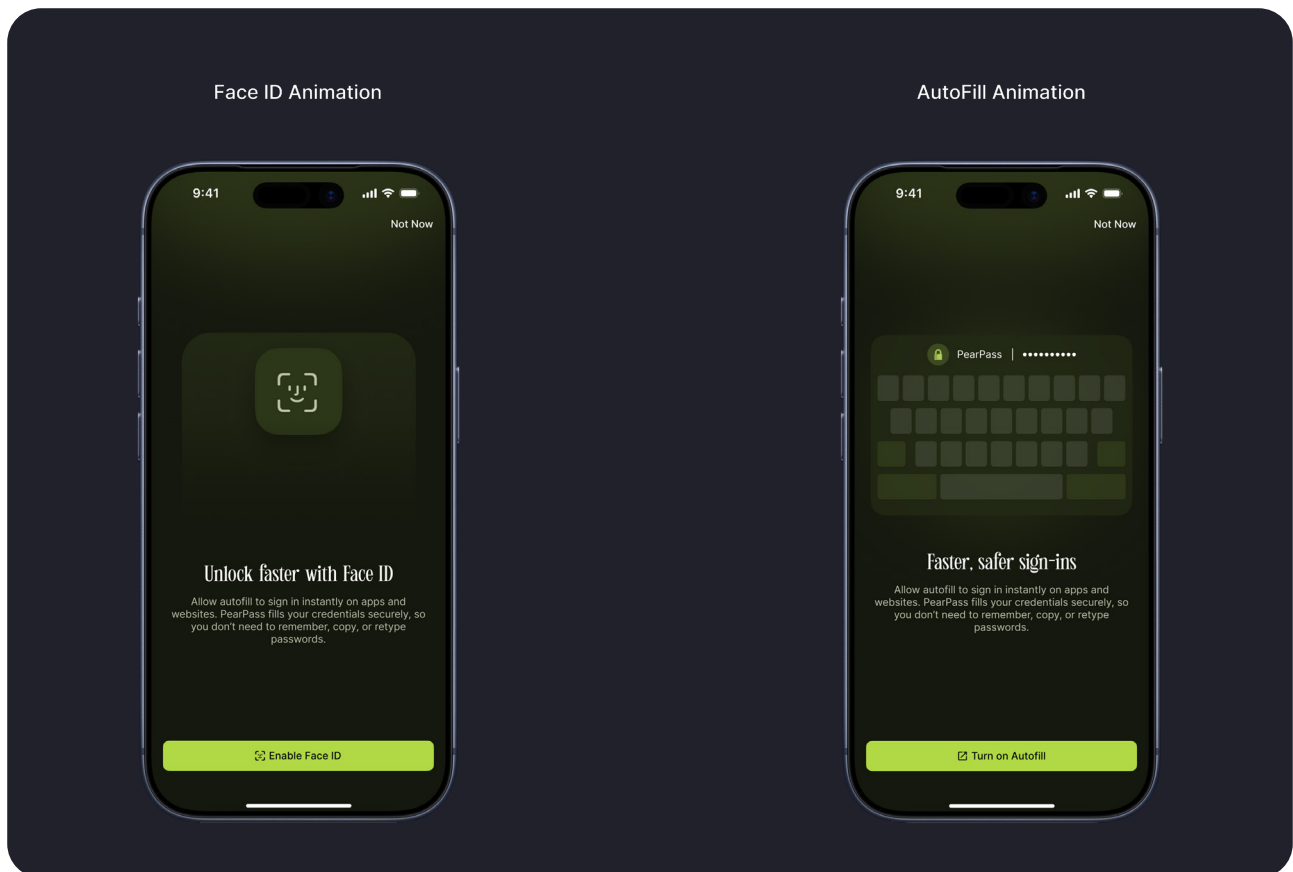
This use of the animations starts during the onboarding of the app, establishing a highly responsive visual environment across both desktop and mobile platforms. As documented in the onboarding sequences across, the interface replaces static technical copy with fluid, moving illustrations. Whether visualizing cloudless network synchronization on desktop environments or guiding mobile credentials through autofill, Face ID, and general biometric activation states, Rive's lightweight runtime engine handles these state changes dynamically based on direct user input. These micro-animations create a dynamic and playful interface, guiding cognitive focus and confirming system readiness through clear, organic transitions.

Beyond the introductory sequence, this engaging visual layer is embedded directly into everyday navigation components to foster ongoing user stickiness. As shown in the structured sidebar layout, the entire suite of category and utility icons incorporates animations. When a user hovers over or selects these navigation entries, the icons respond with subtle, eye-catching movements. Thanks to Rive we can make the interface feel more friendly and alive but keeping the whole app lightweight and highly responsive.

Conclusion

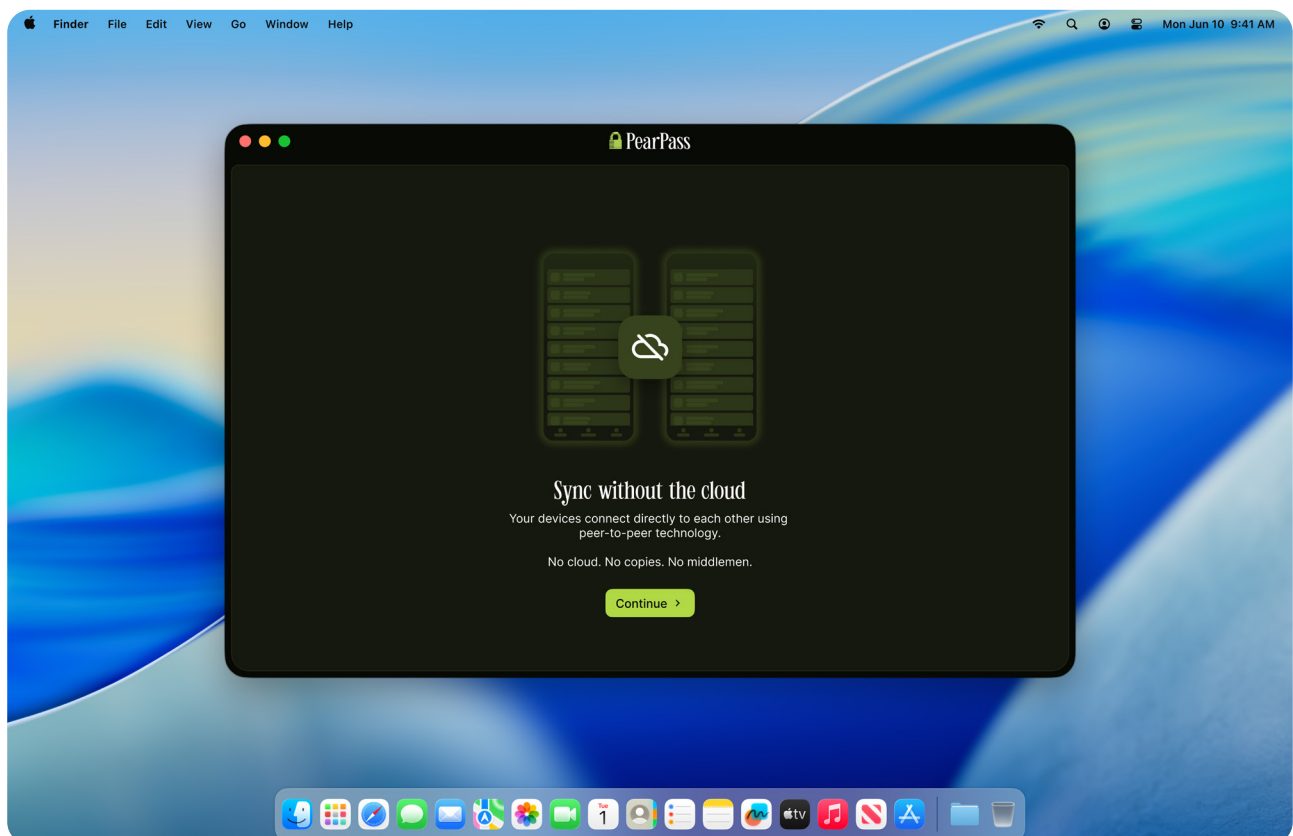
In summary, the comprehensive formulation of the PearPass Version 2.0 visual system, spanning from foundational chromatic constraints to responsive component architectures and animated micro-interactions, establishes a resilient UI framework. By deploying a highly disciplined dark-mode interface, universal iconographic standards via Material Symbols, and clear behavioral boundaries across call-to-action matrices, input fields, contextual dropdowns, and dialog slots, the front-end architecture eliminates the layout fragmentation and technical debt that was present in the earlier implementations.

Furthermore, the integration of real-time sensory confirmations through transient snackbars, adaptive mobile carousels, and responsive Rive animations creates a modern . It successfully replaces the paternalistic "black box" of legacy client-server models with absolute transparency of state. Ultimately, this unified visual habitat proves that high-fidelity usability and strict cryptographic data sovereignty are not mutually exclusive design choices. Instead, through a holistic, design-led product strategy, PearPass transforms the historically intimidating complexity of P2P network topologies into a deeply trustworthy, clean, and accessible domestic consumer experience.



△ Onboarding Animation
Mobile

▽ Onboarding Animation
Desktop



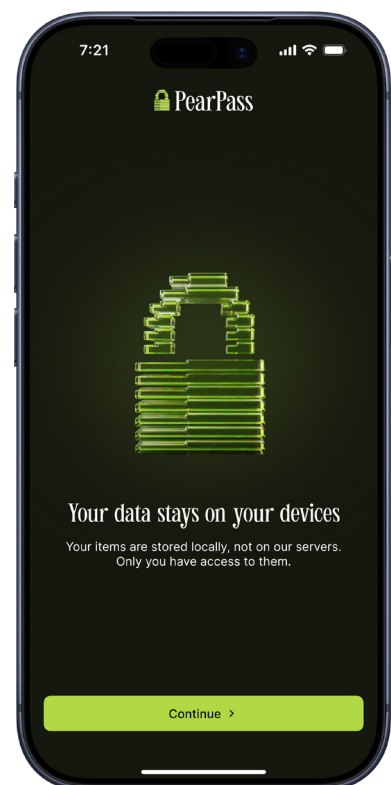
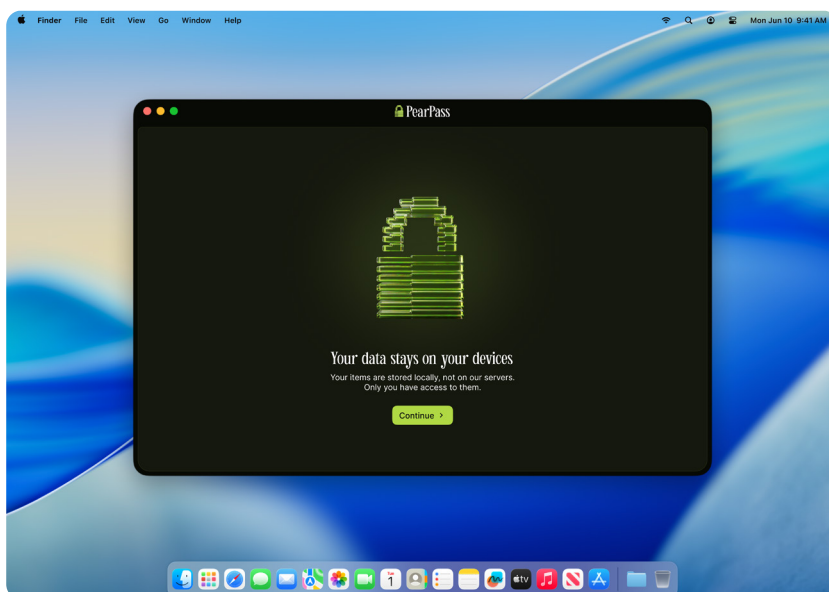
5.6 Restructuring Primary Interface Flows

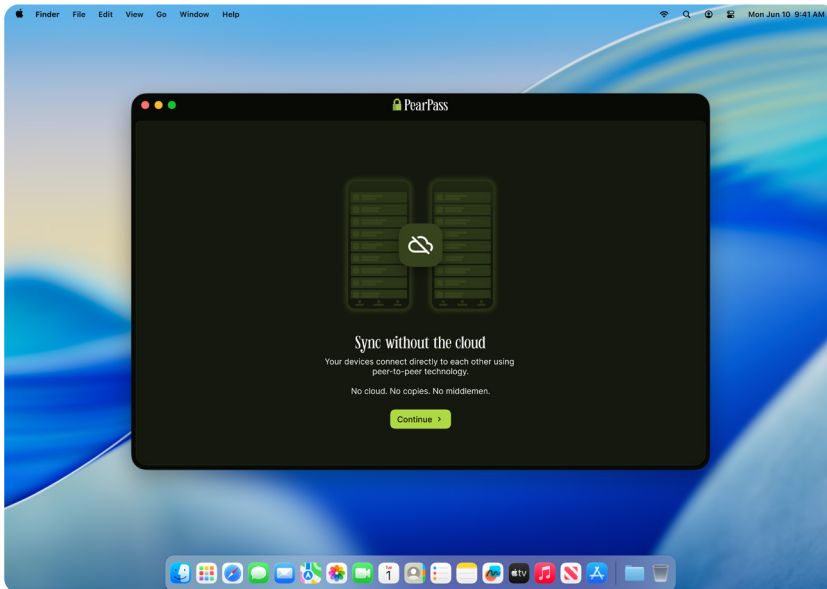
Onboarding

The redesign of the initialization sequence in PearPass version 2.0 addresses the systemic barriers of traditional serverless configurations by dramatically compressing the onboarding funnel. Upon application launch, the user is introduced to an interface that utilizes high-impact, minimalist value-proposition screens. This copy-centric strategy directly targets user mental models, explicitly differentiating the peer-to-peer paradigm from legacy SaaS architectures through declarative axioms (“No cloud. No copies. No middlemen”).

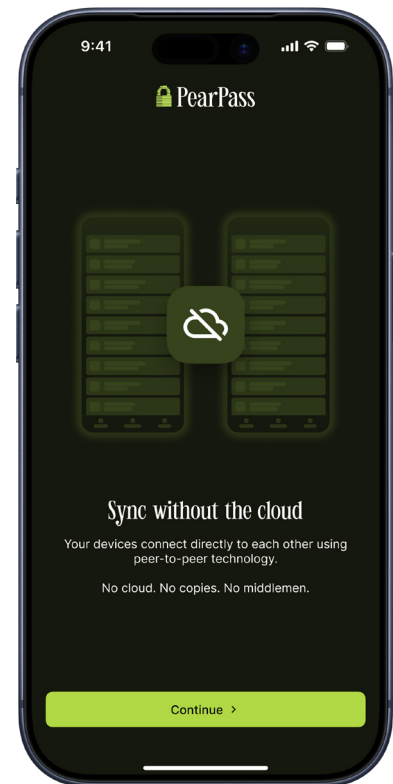
A major change occurs by completely bypassing any explicit, standalone vault-creation phase. Instead of treating the generation of local database containers as a separate interaction step, it completely avoids this tedious step by creating a default vault without the need of any action by the user. The interface monitors input complexity in real time, shifting security indicators from “Decent” to “Strong” while simultaneously rendering an amber high-visibility safety notice warning the user of the absolute, unrecoverable nature of local master password. Once confirmed, the system immediately presents the primary workspace workspace, deploying an instant contextual modal that suggests the user to integrate the app via browser extension. This layout optimization ensures that user friction is mitigated at the entry gate, accelerating the establishment of a secure local node.

First screen of the onboarding where the local structure of the app is presented.

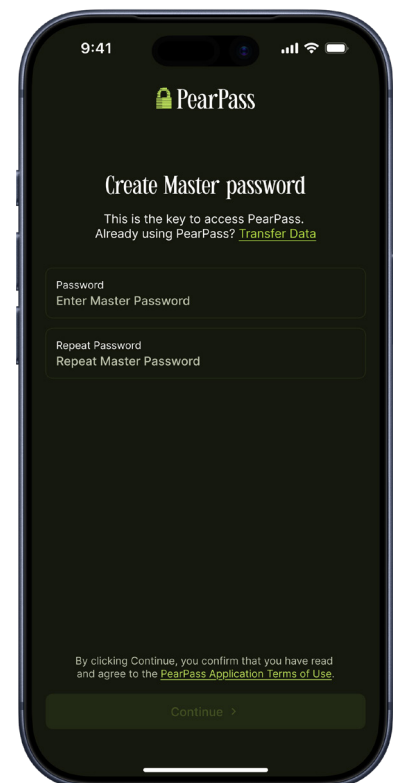
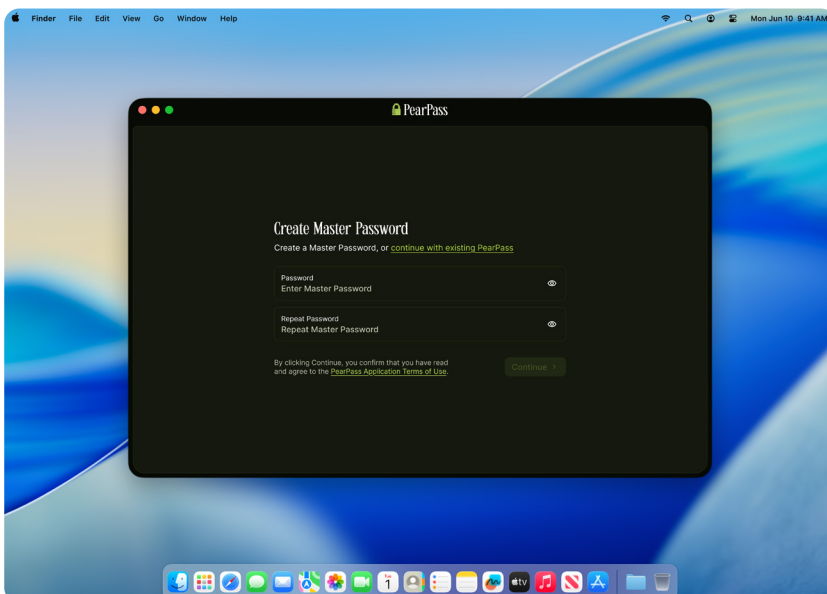


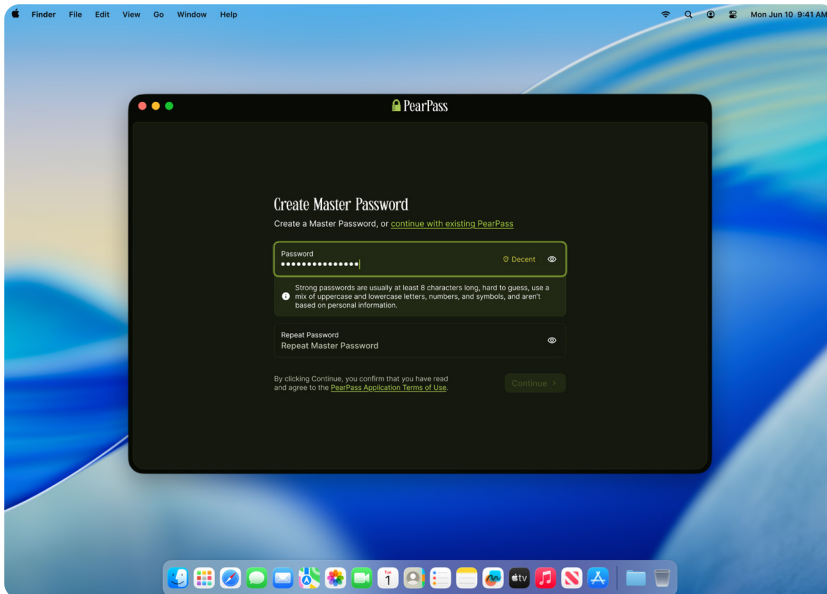


In the second screen we see the possibility to sync the data without the need of the cloud.

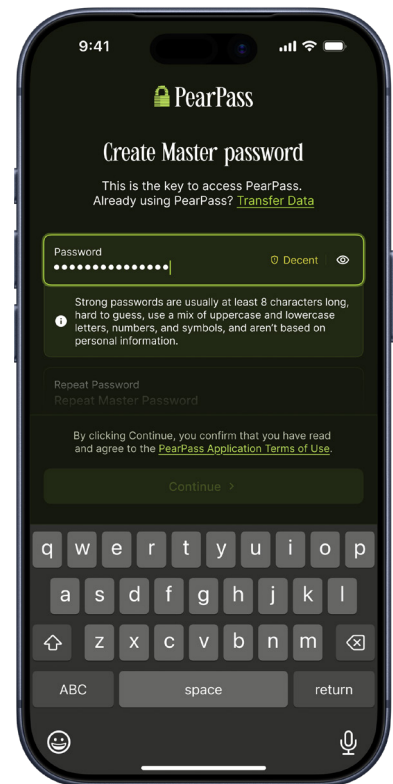


Screen to create the Master Password.

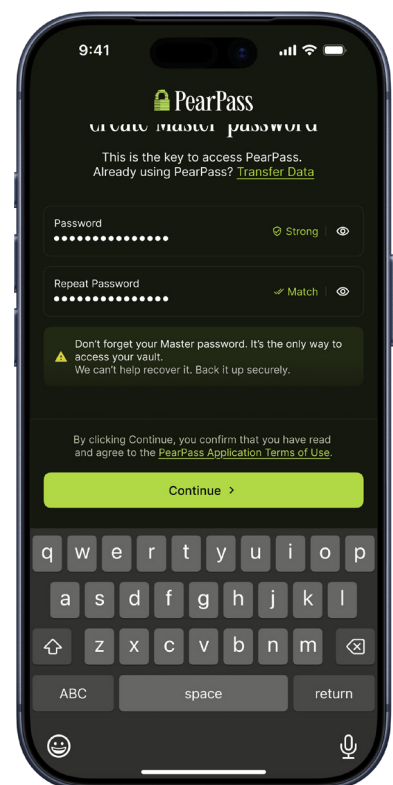
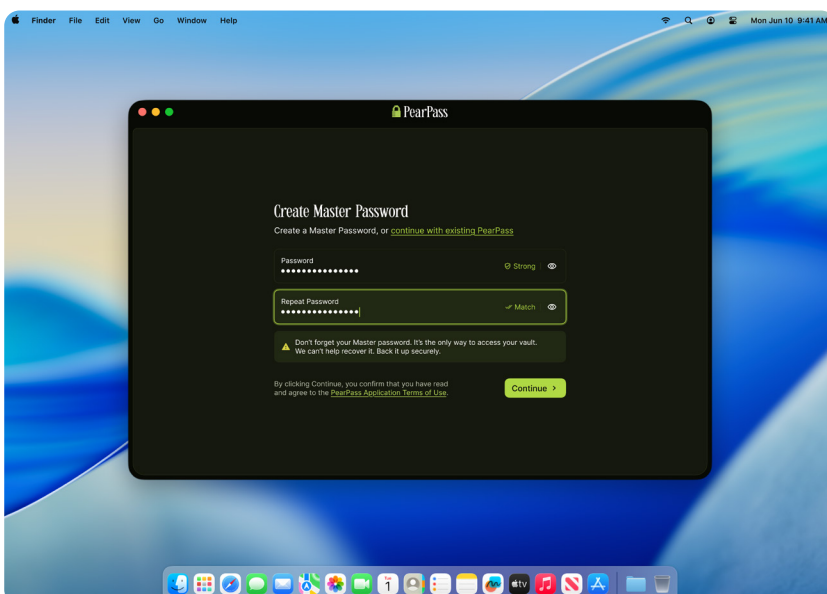


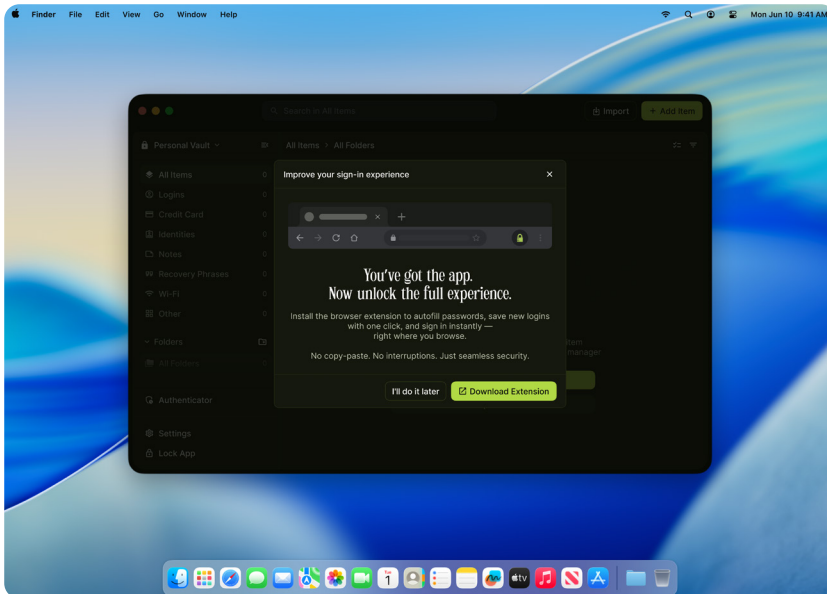


Message that suggest the user to create a strong password with specific elements inside.

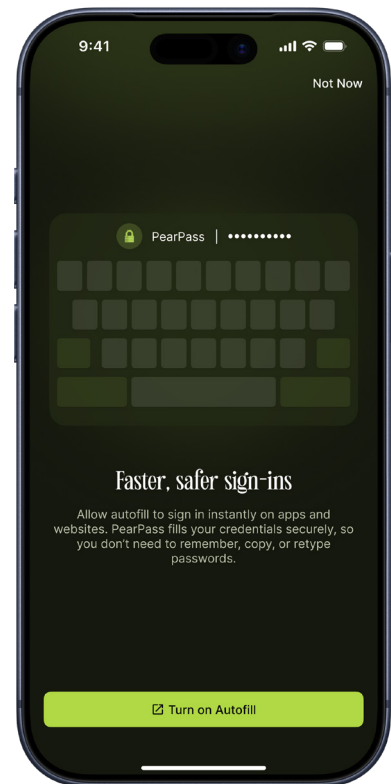


Screen where we remember the user that we can't help recover the master password.

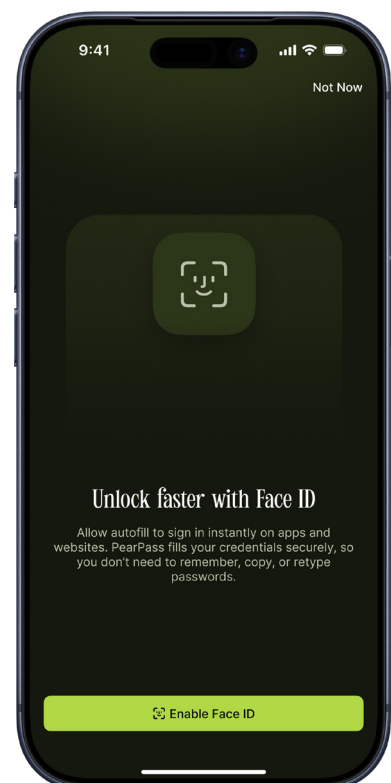




In these screen we suggest two different thing to complete the app experience, to install the browser extension on desktop and to turn on the AutoFill on mobile.



In the mobile app we suggest the user to enable Face ID or Fingerprint to have a faster login.



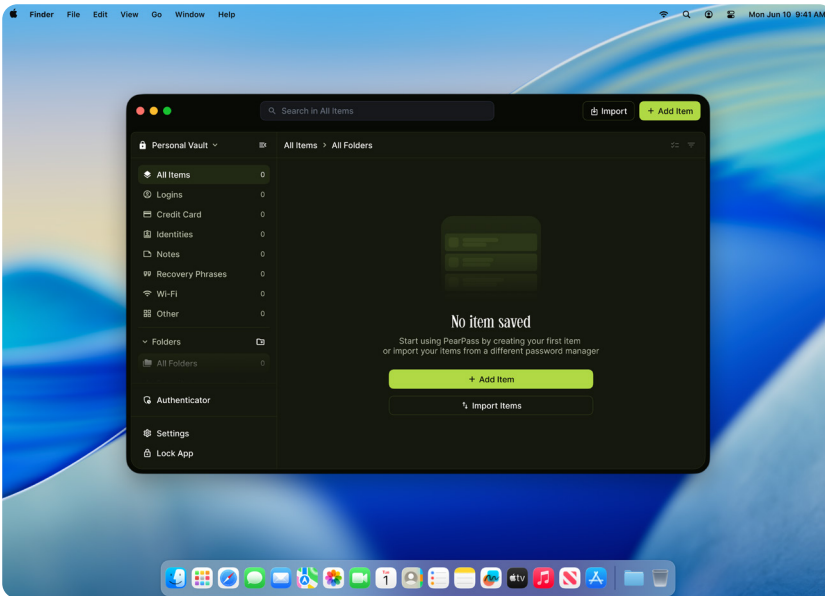
Homepage

To resolve the “wireframe aesthetic” and visual feedback breakdowns of the first iteration, the homepage of PearPass Version 2.0 undergoes a comprehensive visual and cognitive optimization. The restructured information architecture utilizes a disciplined dark-mode surface matrix that establishes an atmospheric sense of strict data enclosure.

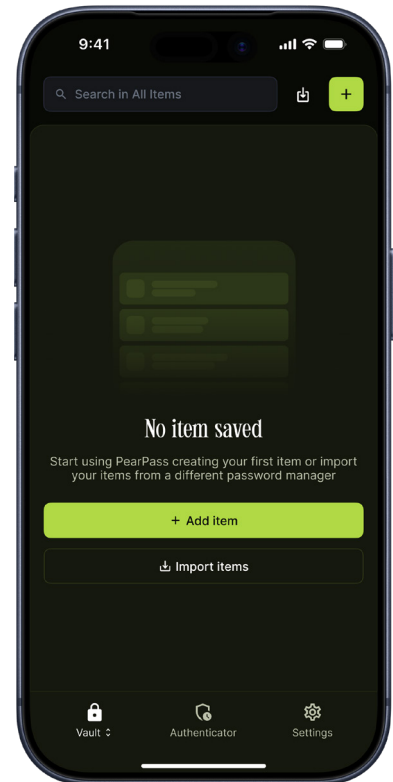
The left column organizes credentials into clear visual containers, utilizing distinct icons for diverse identity assets before transitioning into custom user folders. Directly beneath these components, the inclusion of the Authenticator section resolves a major interoperability gap identified by early adopters. This view presents a specialized interface where time-sensitive cryptographic tokens are aggregated into a single, secure environment. Rather than querying a central server for clock synchronization, the system relies on the local node’s temporal metrics, deploying immediate visual countdown loops (“Codes expiring in 24s”) to communicate the transient state of the tokens directly through the interface.

Cognitive friction is systematically mitigated in the upper control bar. Version 2.0 replaces the chaotic distribution of input fields from Version 1.0 with a single, highly discoverable global search bar. Flanking this global node are two distinct, asymmetric buttons: a secondary outlined “Import” button for integrating external local database formats, and a prominent, green-pill primary button (“+ Add Item”) designed to anchor creation workflows.

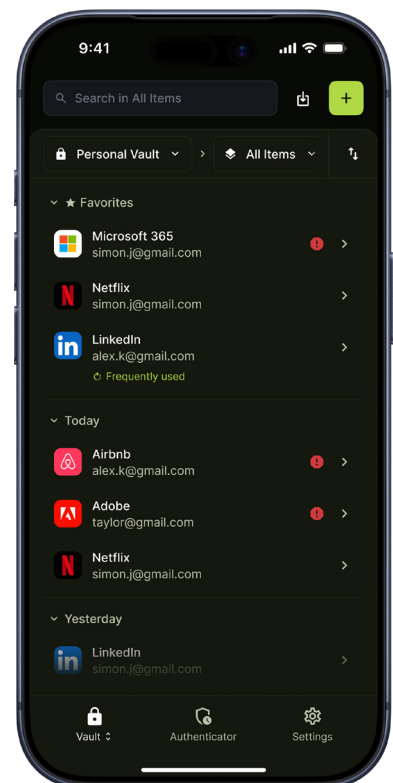
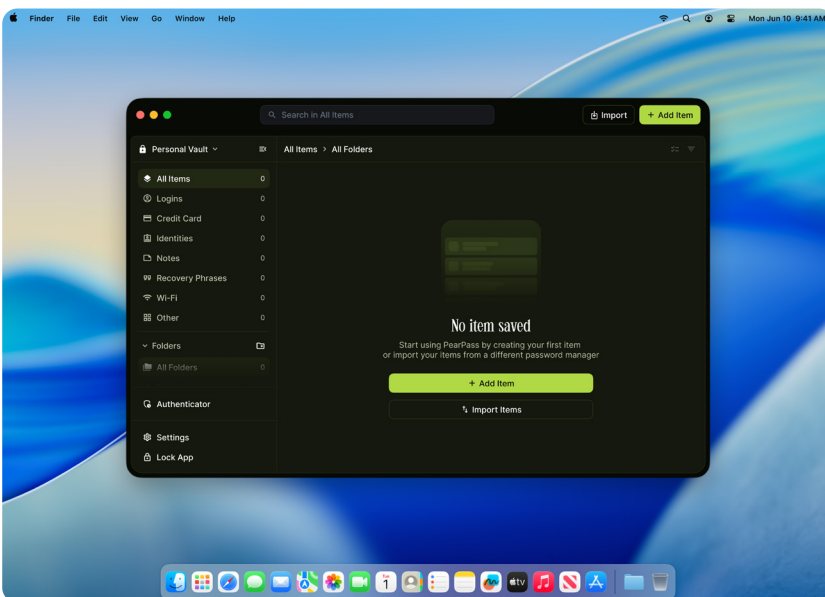
The central interaction zone incorporates structural scaffolding through a responsive breadcrumb component, paired with micro-interaction controls for multi-item manipulation and contextual sorting. True transparency of state is achieved by linking the central list directly to the right preview panel, an active row selection provides immediate, high-contrast feedback, projecting the item’s extended fields (such as passkeys and compromised password warnings) onto the right canvas while avoiding the visual clutter and decision fatigue that characterized the previous layout.



Empty State of the App.



Homepage of the App.

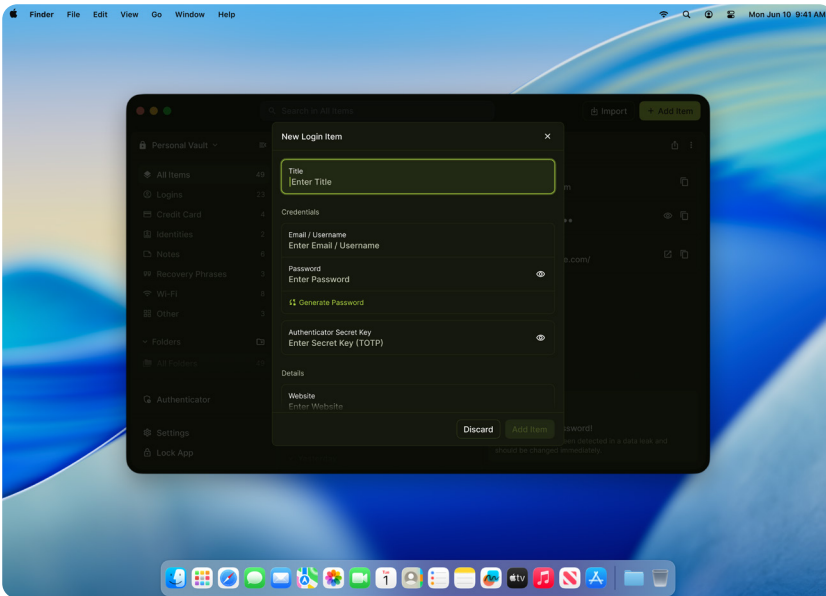


Creation Modal

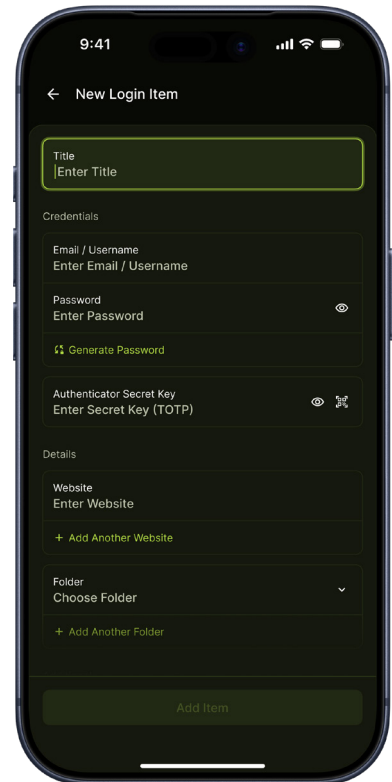
In PearPass Version 2.0, the restructuring of the login item generation form directly responds to the community's demand for ergonomic refinement and clean visual formatting. The layout transitions from a dense, flat input scheme to a layered visual hierarchy designed to reduce cognitive friction during data custody curation. Typographic uniformity has been replaced by a high-contrast typographic system that separates descriptive structural labels from interactive placeholders. Furthermore, as demonstrated in, active entry focus is managed via distinct high-visibility perimeter borders (e.g., the vibrant green highlight bounding the "Title" field), providing immediate sensory feedback and orienting the user's cognitive attention.

The modal interface gracefully handles secondary workflows without cluttering the initial viewport layout. As the user navigates downward through the interaction space, advanced attributes such as multiple URLs, custom folder routing, and operational comments are introduced through expandable, interactive triggers.

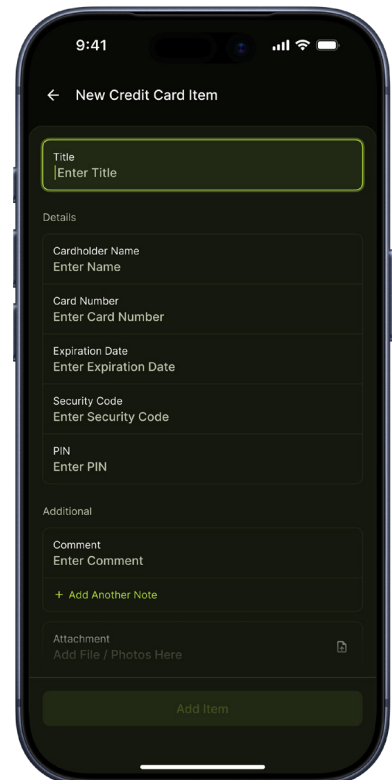
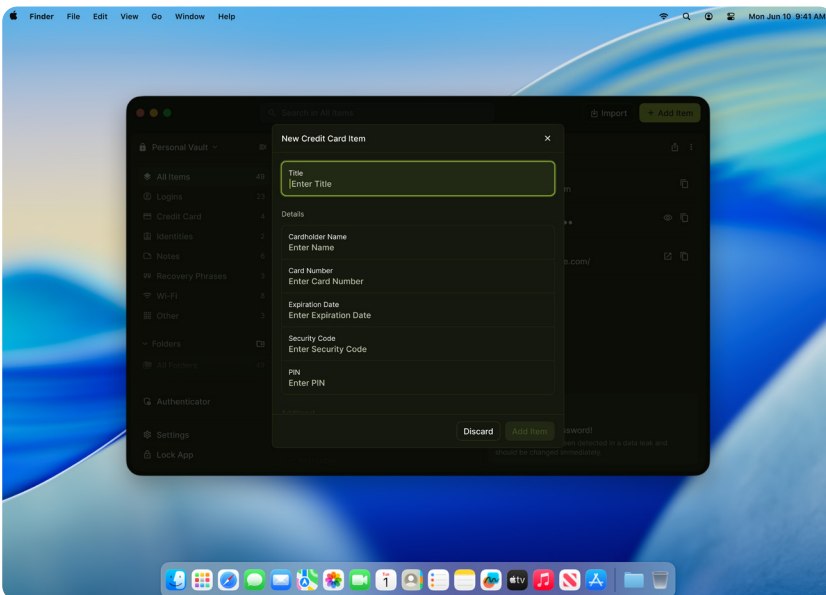
This hierarchical prioritization culminates in the lower quadrant of the canvas, where the "Attachment" and "Hidden Message" input modules are housed. By concealing these parameters behind a natural scrolling flow rather than enforcing absolute visibility, the application safeguards the user from choice overload and visual fatigue. The redesign successfully balances technical flexibility with modern UX design principles, transforming a complex local database configuration routine into a predictable, smooth, and highly polished human experience.



Login creation modal.



Credit card modal.

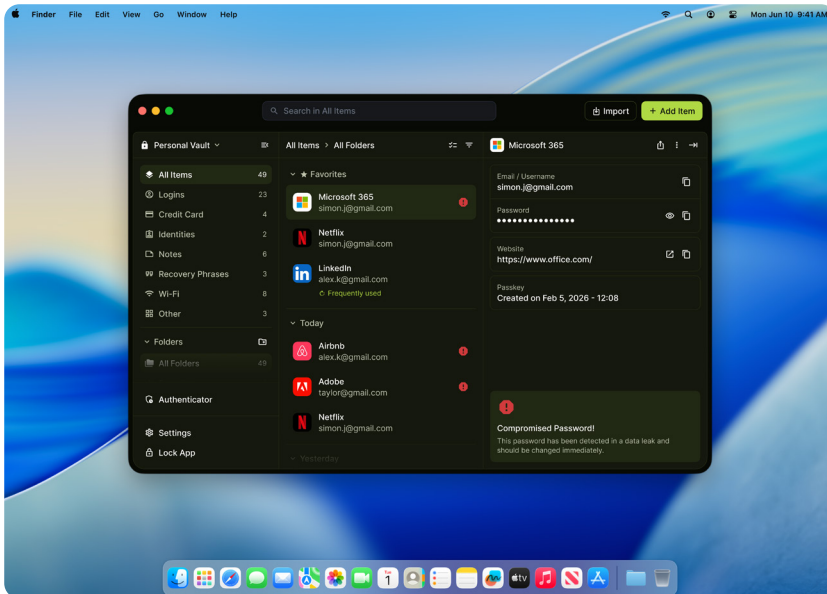


Cross Device Synchronization

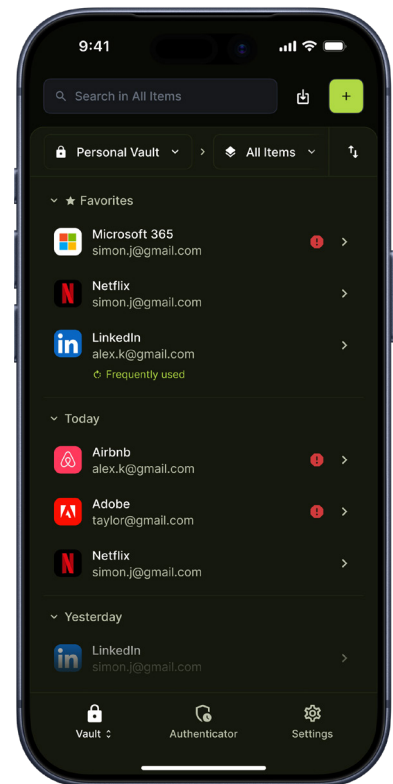
A core milestone in the architectural evolution of PearPass Version 2.0 is the complete branching of importing data and sharing vaults. In the initial v1.0 baseline, merging these actions in a singular, overloaded menu generated significant directional confusion, as users struggled to understand clearly whether they were importing a vault or sharing it. Version 2.0 resolves this tension by deploying two highly specialized pathways that separate the actions into two intentional operations. To facilitate intuitive vault orchestration, the interface introduces decentralized, omnipresent entry points that adapt to varied user mental models. Rather than confining architecture curation to deep application settings, users can immediately invoke vault lifecycle controls directly from the primary workspace view. By interacting with the top-left “Personal Vault” affordance, the interface expands into a dedicated structural left-sidebar panel, transforming a rigid navigation container into a dynamic node management space. This panel exposes the user’s local vaults alongside immediate, context-specific action triggers for sharing them.

The importing flow, triggered via the secondary “Import” action button on the top panel, is engineered as a highly optimized, three-click flow. Upon activation, the modal layout isolates focus onto a single parameter input: the network pointer token. To ensure strict transparency of state, the system does not execute an immediate, blind cryptographic merge. Instead, the interface presents a descriptive summary (“Vault Found: Work”) that allows the active node to preview individual credential records before committing them to the local physical database via the high-visibility “Save Shared Vault” control.

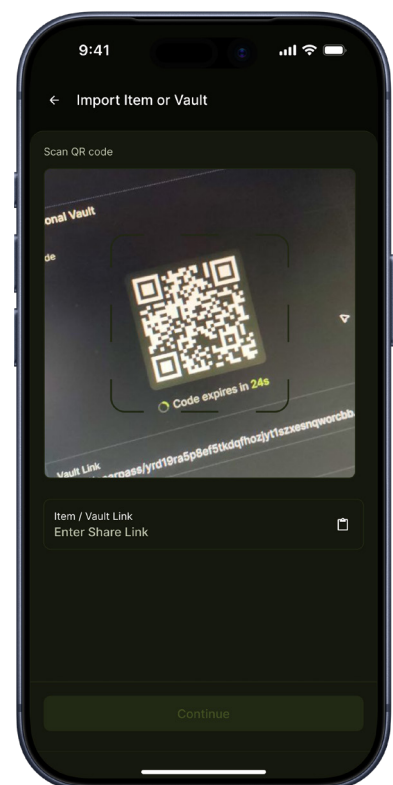
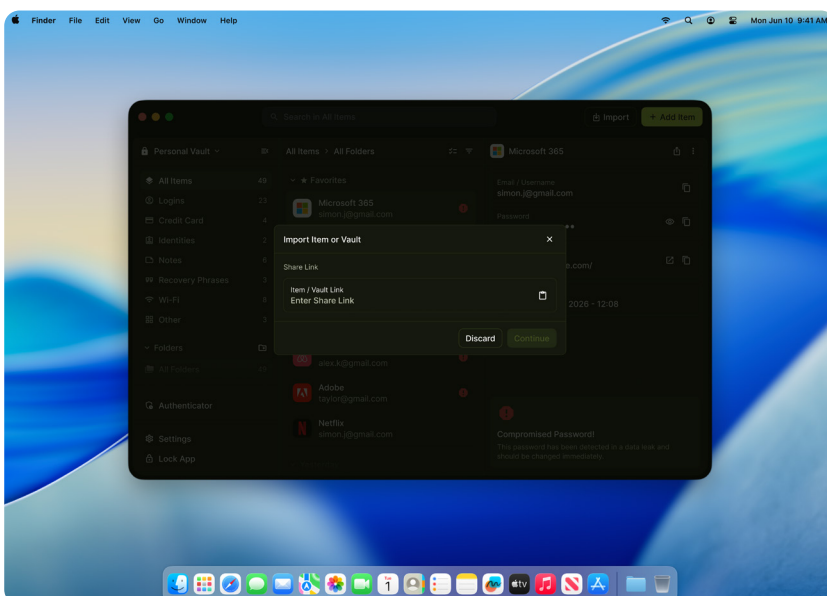
Symmetrically, the outbound broadcasting flow mirrors this linear precision while adapting to the strict temporal constraints of peer-to-peer security frameworks. Selecting the sharing icon next to a localized vault directory triggers the generation of a specialized cryptographic handshake container. This modal simultaneously displays a dynamic, high-density QR code paired with a custom code (`pear://pearpass/...`), which acts as a recognizable sharing code. The code to avoid exposure risk has a temporal decay indicator (“Code expires in 24s”). This visual feedback loop emphasizes the temporary window of the sharing action, supported by a cautionary message that tells the user that pairing grants total data access. By isolating these flows, the application successfully makes this action more understandable even by a basic user.



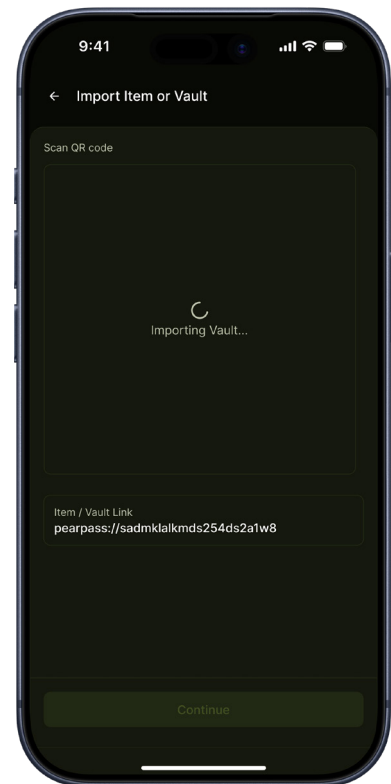
First step of the Importing Process, through the button in the top right angle.



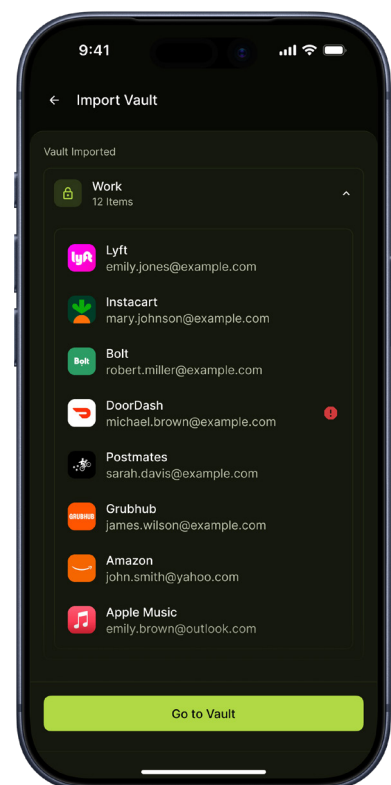
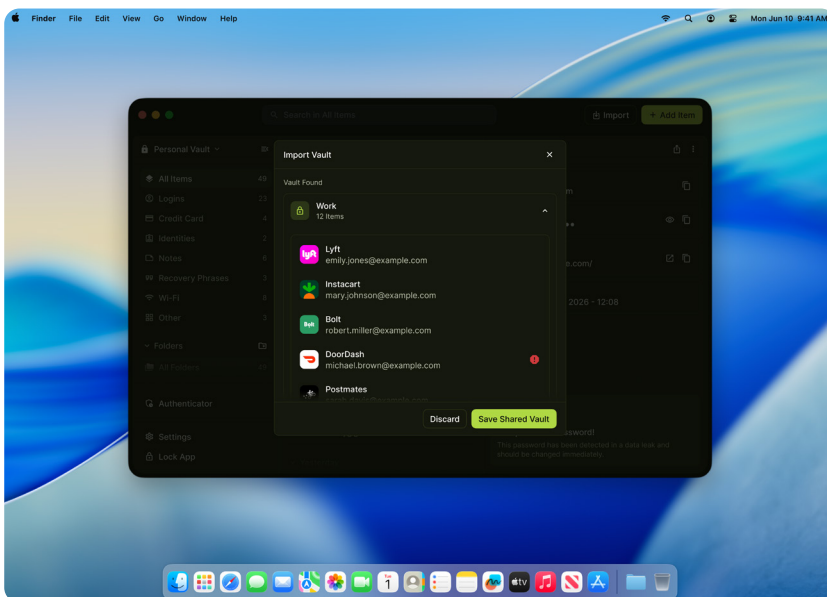
Second step where the user needs to paste or scan the QR code to start the importing process.

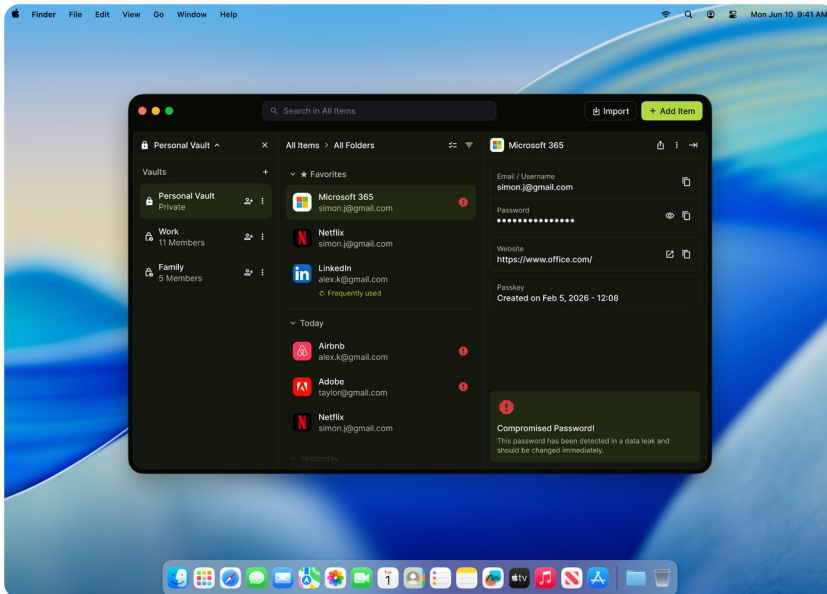


Loading screen of the importing process.

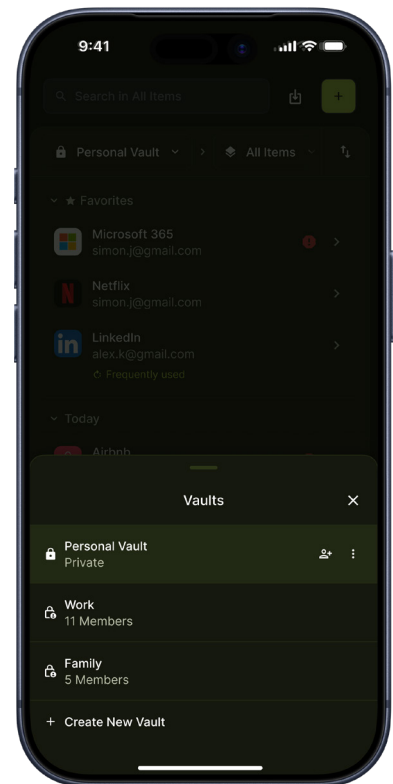


Here the user can check all the items that he will import in his device.

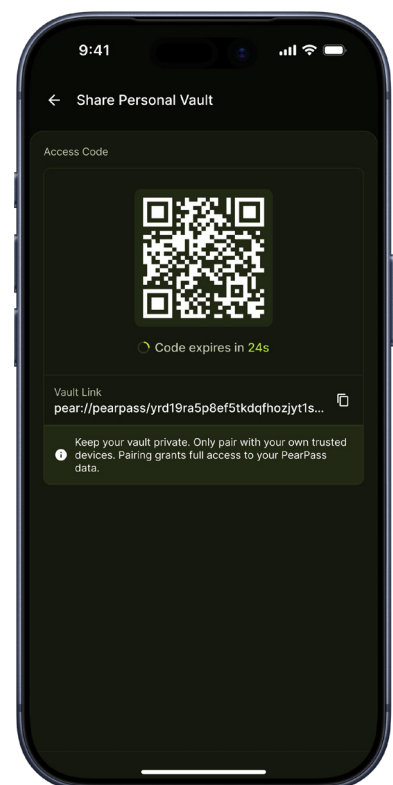
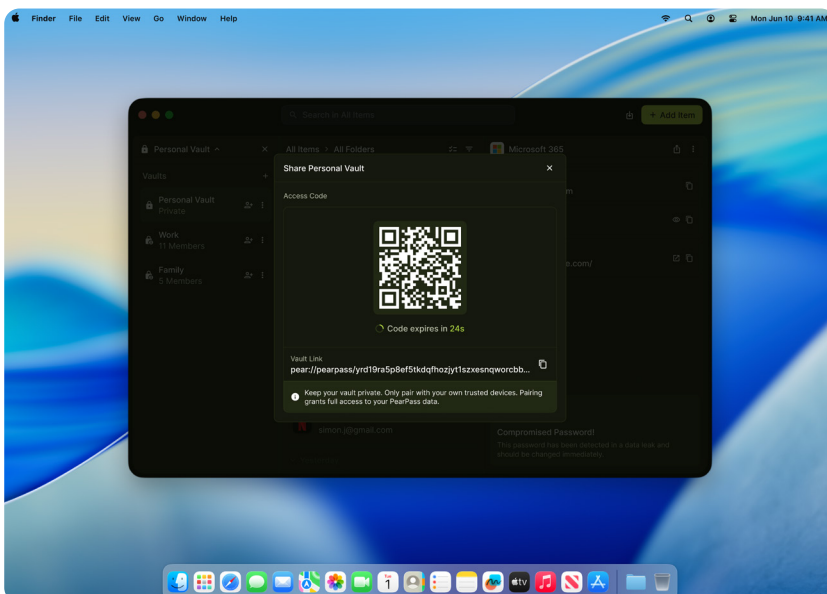




First step to start the sharing of the Vault, that starts clicking on the button “Personal Vault” (top left, desktop) or in the bottom menu of the mobile app.



Last step of the sharing process where the user can copy the code or scan through the mobile app.



Chapter 6

Product Communication

For a serverless peer-to-peer ecosystem to achieve mainstream domestic adoption, the interface design must be heavily supported by the marketing side. The dominance of the cloud-first paradigm has conditioned contemporary individuals to prioritize immediate convenience, frequently obscuring the systemic vulnerabilities inherent in centralized data custody. To address this behavioral bottleneck, the communication framework surrounding PearPass Version 2.0 acts as a pedagogical catalyst, strategically explaining the benefits of peer-to-peer (P2P) architectures. The corporate narrative is deliberately structured around critical risk vectors and cognitive offloading. Specifically, it highlights the severe dangers of centralized cloud storage, such as massive data breaches and honeypot vulnerabilities, while emphasizing the operational paralysis caused by server outages that routinely lock users out of their own digital identities. Concurrently, the communication addresses human behavioral flaws, such as the reuse of weak, predictable passwords tied to personal life events, by promoting the platform's core use: the ability to securely manage an entire cryptographic vault by memorizing a single, robust local Master Password. Through these multiple messages, the strategy simplifies serverless technology, presenting it not as a complex engineering feat, but as a necessary evolution for daily digital resilience.

To outline this design-led distribution strategy, Chapter 6 breaks down the external touchpoints that frame the application's launch. The analysis begins with the product website, it then reviews the visual strategy governing official application marketplace storefronts. The narrative concludes by analyzing the collaborative media assets produced by the marketing team, specifically evaluating promotional video documentation and integrated digital marketing collaterals, to illustrate how unified brand communications cultivate user confidence, helping individuals confidently step into a cloudless digital lifestyle.

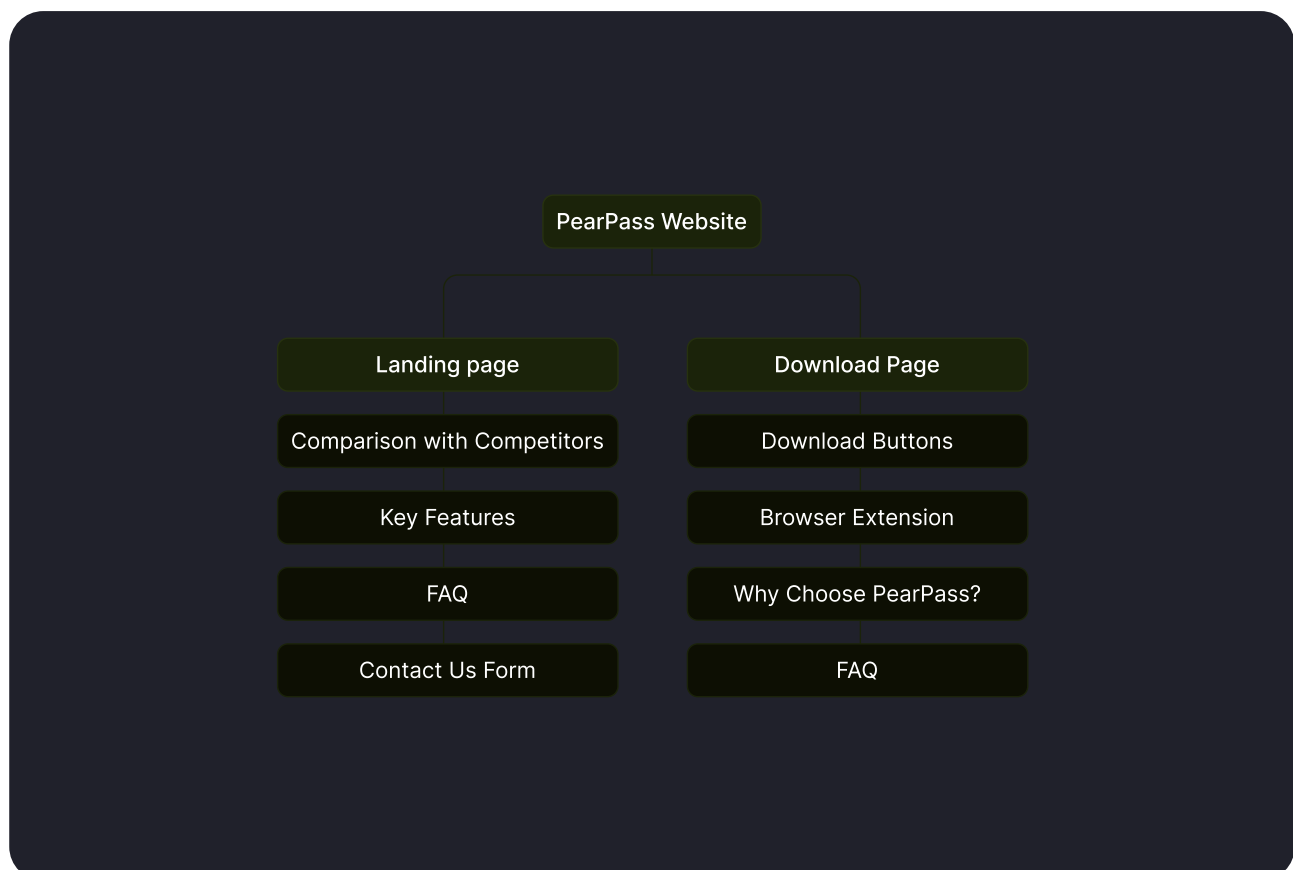
Table of content and Site Map

To ensure optimal information scent and minimize navigational friction, the structural architecture of the PearPass website is designed around a highly streamlined, flat sitemap. The website keeps the content in a few pages to avoid creating useless pages, as a matter of fact the site is organized into two primary pages supplemented by minor ones.

The core **Landing Page** acts as the foundational educational baseline. It integrates a direct feature matrix comparing PearPass against legacy, cloud-dependent competitors to expose market vulnerabilities. This primary view details the application's sovereign capabilities, specifically highlighting passkey support, cross-platform import tools, native peer-to-peer syncing, end-to-end encryption, and an independent security audit, alongside a comprehensive FAQ repository, an embedded contact form, and a direct community invitation to the community group inside Keet, a P2P private message app created always by Holepunch and Tether.

Instead, inside the **Download Page** the user can find all the installers needed to download the app on the various platforms supported. Other than that we present other less important features of the app such as browser extensions, the serverless nature of the app, customizable security timeouts, and the password generator tool. Landing and Download pages are the core ones, beside them we can find the contact page and the

▽ PearPass Website
Site Map



legal ones. Together with these the website redirects the user to the github page where the user could personally audit the code of the app and to the User Doc website, where the user can find a deeper explanation of the software together with some guides on how to properly use the app.

UI of the Website

To preserve the consistency between the platforms and products, the interface design of the public website natively inherits the precise visual taxonomy established for the core PearPass application. The web platform preserves the atmospheric dark-mode palette, typographic scale hierarchy, and iconographic matrices without alteration, modifying spatial attributes only to satisfy the responsive layout rules of web browsers.

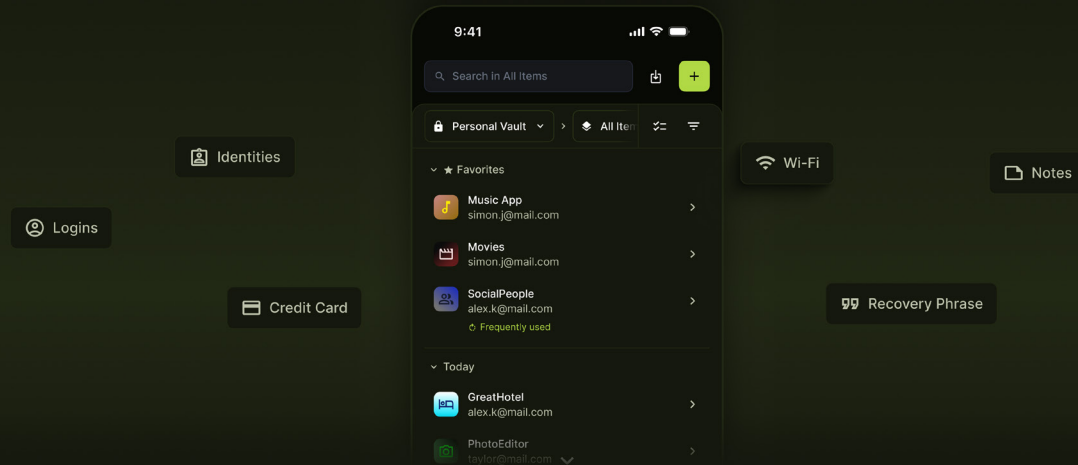
This choice gave the user the possibility to visually match the app and the communication behind it, giving him a hint about the interface he will find in the app itself.

- **The Dynamic Frame Enclosure:** To establish an immediate cognitive link with the software workspace, the landing page layout introduces a tailored animated transition immediately following the primary hero section. As the user navigates downward, an explicit motion routine smoothly encapsulates the scrollable web content within a persistent black frame container. This visual wrapper replicates the physical borders and aesthetic mass of the desktop and mobile application layouts.
- **Animations Aesthetics:** This structural metaphor is completed by incorporating subtle, purposeful interface animations throughout the web viewport. Rather than serving as static decorative graphics, these animated visuals make the website feel alive and highly responsive. This active visual layer bridges the gap between a standard informative webpage and a dynamic local software application, increasing user interest and establishing brand authority before data software conversion occurs.



Your Open-Source Password Manager

Secure P2P sync and sharing, no cloud involved.
Just your data, where it should be: **with you.**

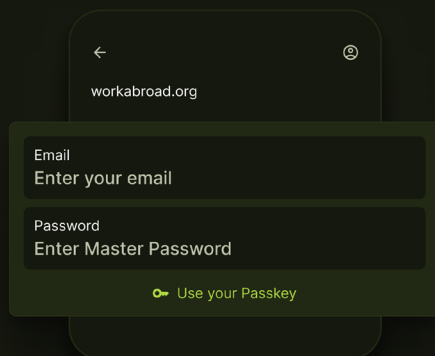
[Download](#)[Know More](#)

△ Website Landing Page

▽ Website Body Page

[Contact Us](#)[User Docs](#)[Github](#)[Download PearPass](#)

Key Features



Passkey

PearPass stores your unique digital keys locally on your hardware, ensuring your credentials stay off the cloud and under your total control.

[Learn more about Passkeys](#)

Import items

Moving to a local-first vault shouldn't be a chore. Securely migrate your credentials from legacy cloud managers directly into your encrypted PearPass vault in seconds.



6.2: App store and Play Store Visuals

When going through a digital marketplace, the primary point of contact between the user and the application consists of the visual assets uploaded to the app page. Consequently, these visuals act as the initial approach of the user to the app, and should create a conversion and paradigm validation, transforming abstract architectural properties into legible interaction paradigms. To maintain structural consistency across different hardware environments and operating systems, targeted mockups were designed for each distribution channel: the iOS App Store infrastructure utilizes high-fidelity iPhone frames, the Google Play Store environment leverages native Android device assets, and the Chrome Web Store utilizes contextual desktop browser interfaces. This strict platform-to-device alignment eliminates visual and cognitive dissonance, respecting the user's immediate operating system context while signaling seamless cross-platform continuity.

Mobile Platform

The narrative progression of the visual banners establishes a logical sequence of risk demystification and technical disclosure:

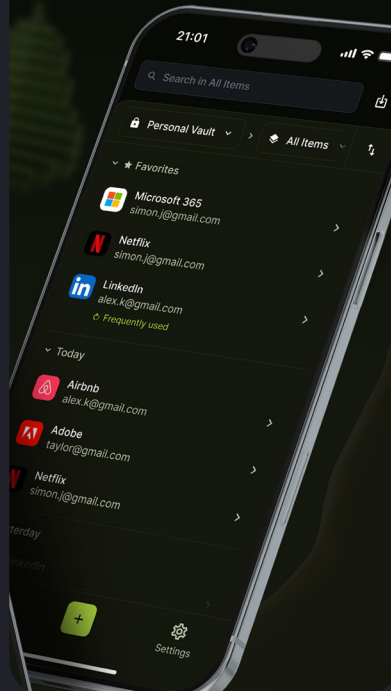
- **The Initial Trust Anchor** opens the carousel by presenting a minimalist corporate splash layout backed by a clear architectural axiom: "The fully local, open-source password manager. Your data stays on your devices." This declaration targets cloud anxieties directly, framing data isolation as a foundational product property.
- **The Local Storage and Encryption Quadrants** expand this premise by displaying actual high-contrast mobile user interface screens, such as the master password initialization layout, explicitly instructing the individual to "Keep your passwords off the cloud."
- **The Peer-to-Peer Synchronization Vector** addresses the core network architecture by showcasing an organic human interaction metaphor, a hand holding a device scanning an active QR token over a local desktop interface. This visual explains the unmediated node-to-node handshake sequence ("Sync your devices directly without cloud intermediaries"), reducing the abstract nature of the Pear network into a tangible physical gesture.
- **The Utility and Extension Node** completes the sequence by highlighting the local password generator matrix and extended asset vaults (Store More Than Passwords), proving that absolute cryptographic data sovereignty can be seamlessly delivered inside an elegant, lightweight, and high-fidelity mobile consumer product.

The fully local,
open-source
password manager



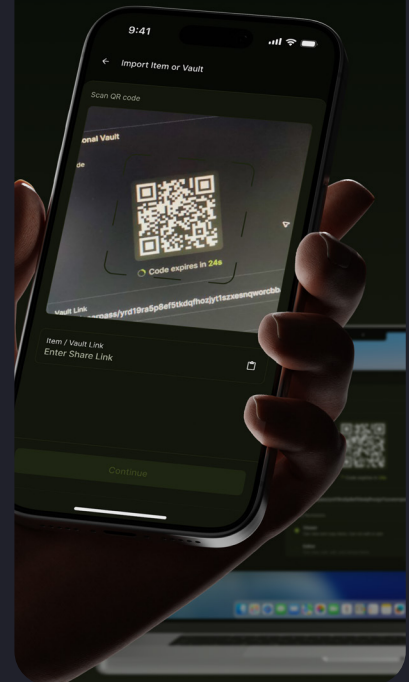
Local Storage

Keep your passwords off the cloud.



Peer-to-Peer Syncing

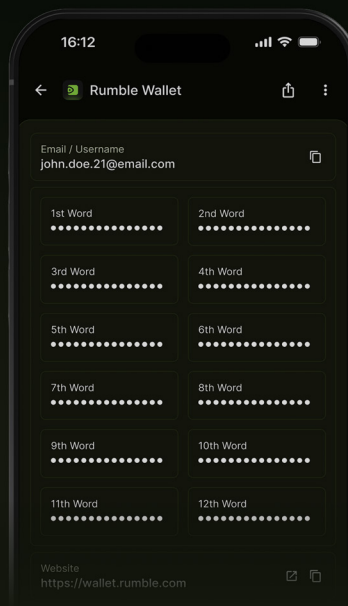
Sync your devices directly without
cloud intermediaries.



△▽ App Store Visual

Store More Than Passwords

Securely store logins, IDs, cards, and
notes—organized and encrypted.



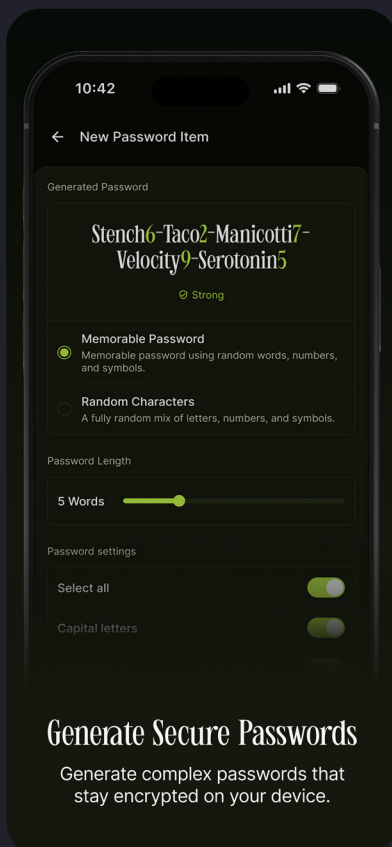
E2E Encryption

Only you can access your data.



Generate Secure Passwords

Generate complex passwords that
stay encrypted on your device.

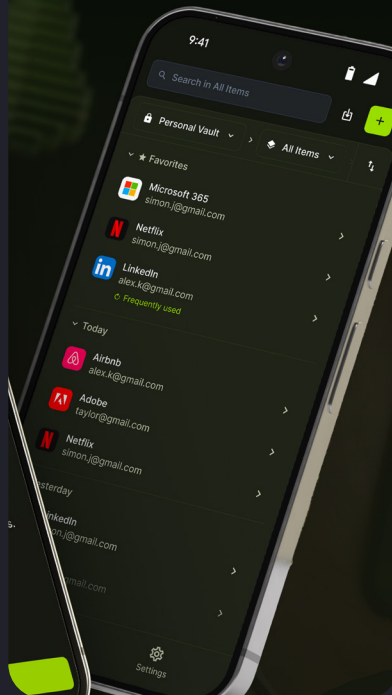


The fully local,
open-source
password manager



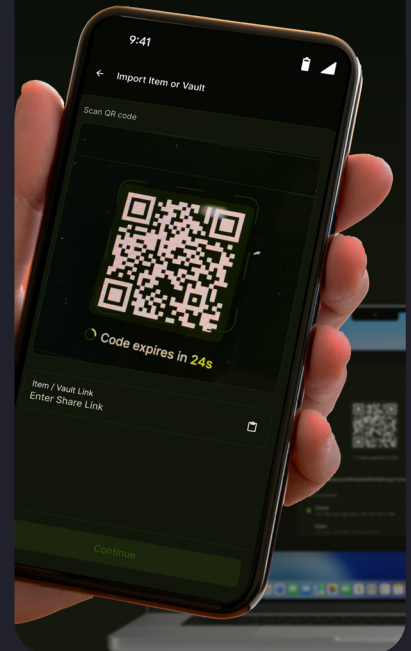
Local Storage

Keep your passwords off the cloud.



Peer-to-Peer Syncing

Sync your devices directly without
cloud intermediaries.



△▽Play Store Visual

9:41
← New Password Item

Generated Password

Stench6-Taco2-Manicotti7-
Velocity9-Serotonin5

Strong

Memorable Password

Memorable password using random words,
numbers, and symbols.

Random Characters

A fully random mix of letters, numbers, and symbols.

Password Length

5 Words

Password settings

Select all

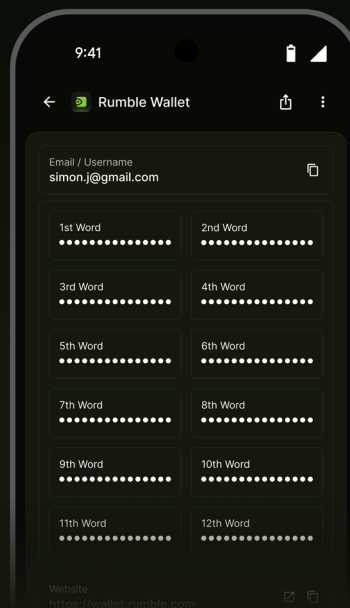
Capital letters

Generate Secure Passwords

Generate complex passwords that
stay encrypted on your device.

Store More Than Passwords

Securely store logins, IDs, cards, and
notes—organized and encrypted.



E2E Encryption

Only you can access your data.

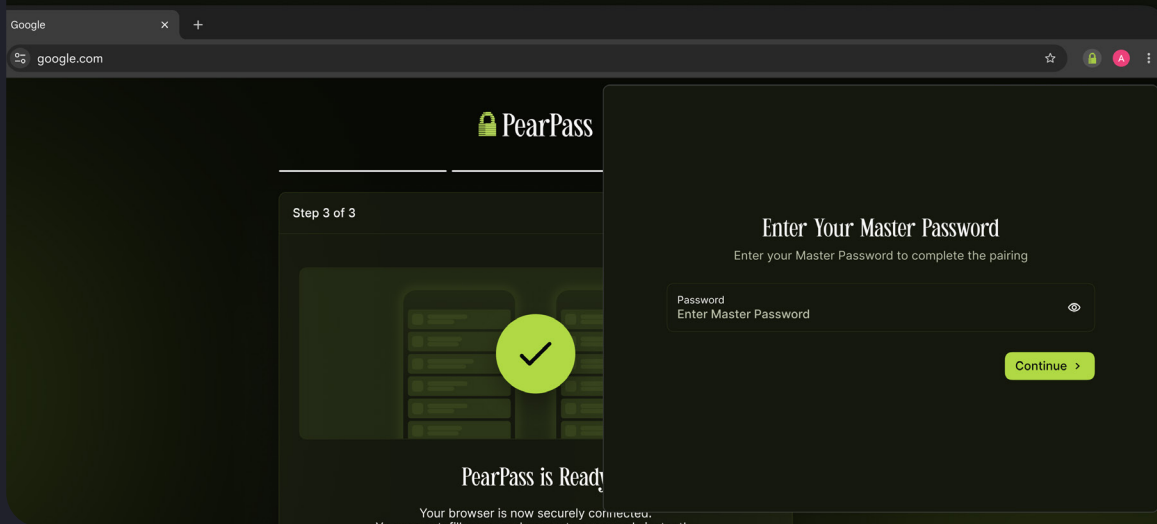


Browser Extension Platform

Within the Chrome Web Store, the visual assets map the extension of the local cryptographic vault into the user's primary desktop execution environment. The narrative sequence unfolds across four continuous storefront assets:

- **The Initial Trust Anchor and Runtime Initialization:** The carousel opens with a high-fidelity desktop browser layout asserting the core value proposition: "The fully local, open-source password manager." The UI highlights a contextual micro-interaction, the master password input overlay during a multi-step node pairing sequence ("Enter Your Master Password to complete the pairing").
- **The Local Storage and Vault Architecture:** The second banner shifts focus to data isolation with the directive, "Local Storage: Keep your passwords off the cloud." The accompanying interface mockup reveals the information architecture of the extension's popup view, displaying a structured layout of categorized credentials (e.g., Microsoft 365, Netflix, LinkedIn). By presenting a fully populated local database viewport dire
- **The Peer-to-Peer Synchronization Vector:** The third banner introduces the core network architecture, stating: "Peer-to-Peer Syncing: Sync your devices directly without cloud intermediaries." The visual utilizes a powerful spatial metaphor: a physical hand holding a mobile node to scan a localized cryptographic QR token displayed on a laptop screen. This cross-device composition models the unmediated node-to-node handshake sequence, translating abstract decentralized protocol interactions into an intuitive physical affordance that eliminates cognitive friction.
- **The Expanded Asset Node:** The final banner completes the sequence by establishing the product's functional versatility: "Store More Than Passwords: Securely store logins, IDs, cards, and notes, organized and encrypted." The UI showcases a the seed phrases form (A master key made of words to recover your crypto), proving that absolute data sovereignty does not limit utility.

The fully local, open-source password manager



△ ▽ Chrome Store Visual

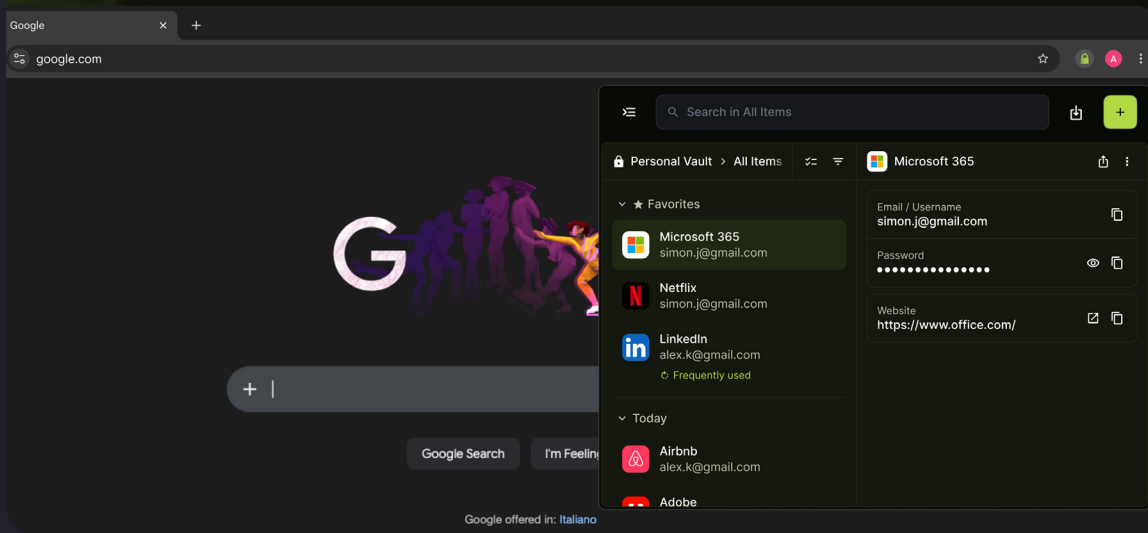
Peer-to-Peer Syncing

Sync your devices directly without cloud intermediaries.



Local Storage

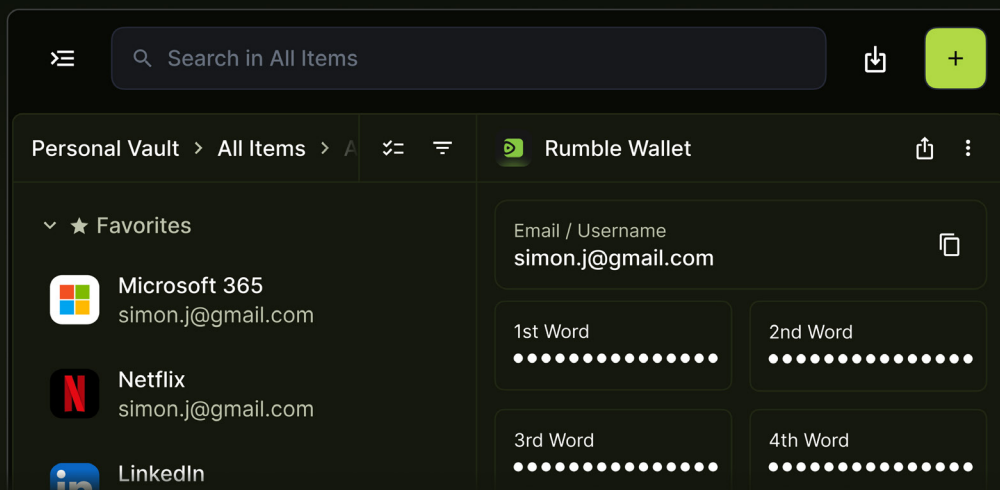
Keep your passwords off the cloud.



△ ▽ Chrome Store Visual

Store More Than Passwords

Securely store logins, IDs, cards, and notes—organized and encrypted.

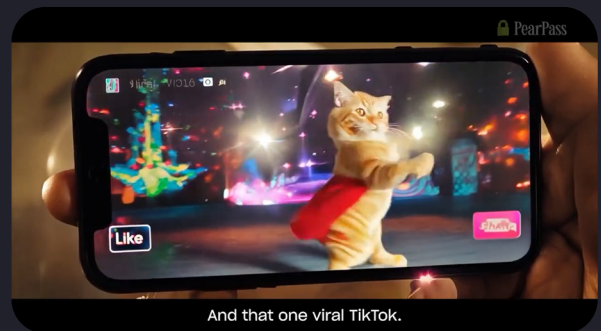
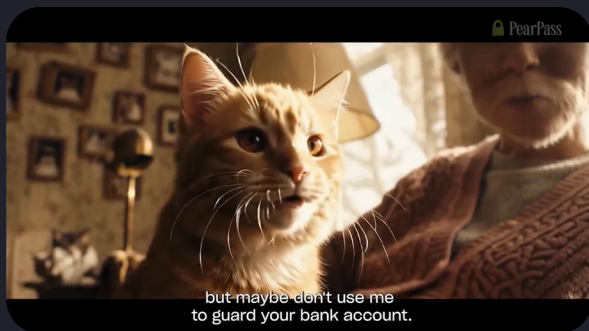
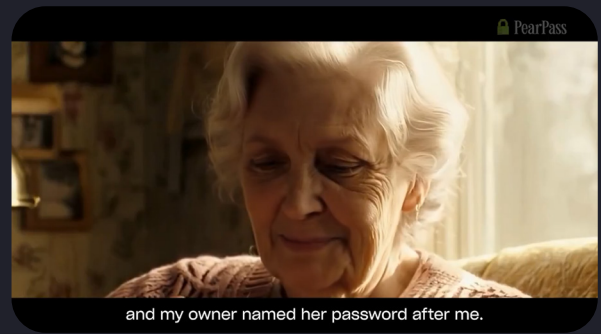


6.3 AI Video

Tether AI filmmaking team created various assets to promote the app on social media such as X and Instagram. Inside the communication strategy, these short-form video assets function as narrative toolkits that target two relational dimensions of contemporary digital vulnerability: human behavioral errors and centralized cloud infrastructure. Most of the videos use a satirical and ironic tone across digital channels like X (formerly Twitter) to expose the latent fragility of the cloud-first status quo, translating abstract network concepts into human scenarios.

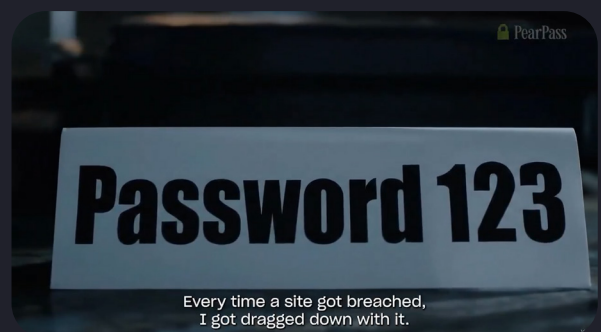
The campaign divides its pedagogical critique into a structured series of character-driven vignettes:

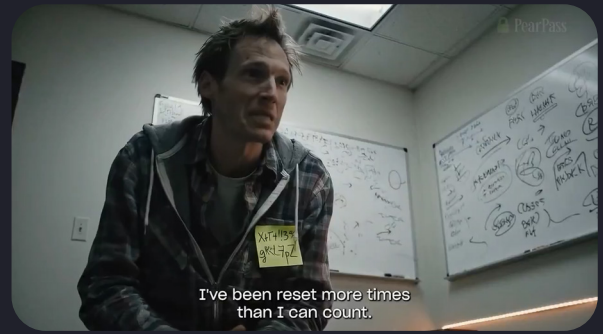
- **The Anthropomorphic Framing of Cognitive Flaws:** One short video addresses the severe psychological fatigue of bad user password hygiene. By characterising the common, insecure string “password123” as an overworked, exhausted man isolated in a dark room complaining about data breaches, the narrative transforms a mathematical security flaw into an empathetic behavioral warning. This approach is mirrored in another short video, which leverages a talking orange cat to critique the widespread reliance on simple, life-derived nomenclature (such as pet names used to secure high-stakes banking interfaces), highlighting how easily predictable data pools are harvested by malicious hackers.
- **The Memory Extraction Paradox:** Another video addresses the UX failures of legacy password enforcement. By interviewing a character wearing a randomized password tag that suffers because people continues to forgot them, constantly with the need to change the password, the campaign highlights the cognitive friction of modern security standards. PearPass is introduced as the ultimate cognitive offloading tool, demonstrating that users only need to retain one single, local Master Password while the client securely generates and encapsulates the rest.
- **The Critical Outage Matrix:** The final, high-urgency corporate narrative targets infrastructural path dependency. It features a professional locked out of essential workspaces due to cloud server downtime. This scenario visualizes the absolute operational paralysis induced by centralized gatekeepers. The subsequent introduction of PearPass highlights the resilience of serverless, local-first connectivity, proving that keeping cryptographic data localized on edge-nodes preserves absolute user autonomy and workflow continuity even when external networks fail.



△ Orange cat video

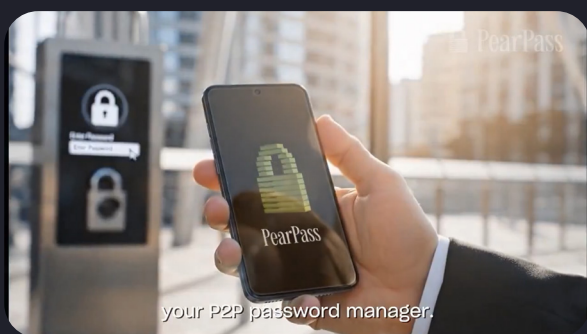
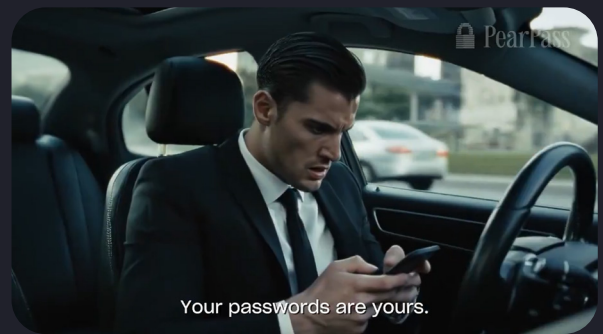
▽ Overused password video





△ Forgot complex password video

▽ User can't access data cause of the cloud video



6.4 Digital Marketing and Static Visuals

Other assets of PearPass's communication ecosystem are static digital marketing visuals that use visual dualism to challenge the server-dependent paradigm. As documented across the digital marketing campaign on X, the marketing team establishes a clear graphic group based on the use of colours and structural metaphors.

This informational strategy uses clear visual contrasts to dismantle the perceived safety of legacy database models:

- **The Architecture of Binary Opposition:** The layouts use a split-view format to draw sharp comparisons. The left axis uses a signature green tone to show localized device sovereignty ("Your device. Your security"), while the right axis uses an uninviting blue hue to expose centralized cloud models ("Their servers. Your risks"). By juxtaposing a self-contained device held in a hand against a phone leaking fluid data or a peaceful user against a hooded hacker in a cold server farm, the visuals transform abstract security differences into clear physical risks.
- **Infrastructural Failure Metaphors:** This critique of server weaknesses is intensified Whether rendering a dark storm cloud hovering over a digital circuit grid, a vulnerable server room , or an old computer melting into a puddle of leaked credentials , the imagery reinforces a core message: remote storage introduces dangerous single points of failure.
- **Cognitive Offloading and Usability:** Conversely, shifts the focus to human cognitive limitations. By showing a consumer overwhelmed by paper files next to a clean laptop layout, the graphic highlights the platform's core functional benefit: the ease of managing an entire database through one secure local Master Password.

Together, these static assets provide an accessible gateway that guides users away from cloud dependencies toward a secure, serverless local installation.

To summarize, the comprehensive configuration of PearPass's marketing assets, ranging from the public website to the visuals for the mobile stores mockups and social platform posts, establishes a unified marketing strategy that actively supports the spread of the app itself.



Your device.
Your security.

Cloud Password
Managers



Their servers.
Your risks.



Keep your
passwords
off the cloud.

DOWNLOAD FREE NOW



Pearpass remembers
your passwords
so you don't have to.

DOWNLOAD FREE NOW

Chapter 7

Conclusion

The research and development underlying this thesis demonstrate that the structural paradigm of credential management can be successfully transitioned from centralized cloud environments to serverless, peer-to-peer networks without sacrificing user experience. The primary ambition of this project was to dismantle the convenience-sovereignty paradox. By executing the PearPass digital habitat, the investigation proves that complex decentralized protocols can be translated into intuitive, high-fidelity interfaces accessible to ordinary individuals rather than remaining restricted to technical practitioners. This design-led optimization effectively de-commodifies personal information, removing the systemic vulnerabilities of central storage honeypots and restoring absolute data custody to the end-user.

Over the course of this extensive research and iterative design process, my professional methodology and technical capabilities as a designer have matured substantially. Transitioning from the developer-centric, constrained baseline of version 1.0 to the disciplined, responsive visual taxonomy of version 2.0 provided invaluable practical insights into evolutionary prototyping and team dynamics. Managing the integration of state transparency, sensory feedback loops, and atomic components forced a rigorous continuous refinement of my interaction design workflows.

Finally, sincere gratitude is extended to Tether for providing the strategic opportunity and resources necessary to develop this project. The collaboration allowed for a direct alignment between academic research and industrial software engineering frameworks. The prospective trajectory of this ongoing partnership offers a compelling space to further investigate and scale sovereign consumer technologies within the decentralized ecosystem.

Bibliography

- **Wu, D. D. (2021).**
Decentralized cloud computing. ISACA Journal.
<https://www.isaca.org/resources/isaca-journal/issues/2021/volume-5/decentralized-cloud-computing>
- **Paylo. (2023, November 19).**
The history of P2P networks and why they remain so important now.
<https://apaylo.com/the-history-of-p2p-networks-and-why-they-remain-so-important-now/>
- **Felipe. (2018, March 7).**
A brief history of P2P content distribution, in 10 major steps. Medium.
<https://medium.com/paratii/a-brief-history-of-p2p-content-distribution-in-10-major-steps-6d6733d25122>
- **HenryR (2008).**
Simple basic explanation of a distributed hash table (DHT). Stack Overflow.
<https://stackoverflow.com/questions/144360/simple-basic-explanation-of-a-distributed-hash-table-dht>
- **ITMedia Consulting & Centro di Ricerca ASK Università Bocconi. (n.d.).**
L'economia dei dati.
<https://www.itmedia-consulting.com/DOCUMENTI/economiadeidati.pdf>
- **University of Tennessee Office of Innovative Technologies. (n.d.).**
Digital privacy—What is a data broker?
<https://oit.utk.edu/security/learning-library/article-archive/what-is-a-data-broker/>
- **Singer, P. (2015, January 7).**
Acxiom Corp.: The “faceless organization that knows everything about you”. The Week.
<https://theweek.com/articles/474509/acxiom-corp-faceless-organization-that-knows-everything-about>
- **Framework Security. (2026, March 25).**
The Equifax hack: A cybersecurity catastrophe.
<https://frameworksecurity.com/post/the-equifax-hack-a-cybersecurity-catastrophe>
- **Freedom of the Press Foundation. (2024).**
Avast caught selling browsing data.
<https://freedom.press/digisec/blog/avast-caught-selling-browsing-data/>
- **TechMonitor. (2020, January 30).**
Avast CEO killing off Jumpshot data harvesting...
<https://www.techmonitor.ai/technology/data/avast-ceo-jumpshot>
- **IONOS. (2023, June 13).**
What is vendor lock-in?
<https://www.ionos.com/digitalguide/hosting/technical-matters/vendor-lock-in/>

- **JDSupra. (2024, April 21).**
Transitioning from traditional software licensing to SaaS.
<https://www.jdsupra.com/legalnews/transitioning-from-traditional-software-2800971/>
- **Kaspersky. (2020, June 25).**
Cos'è un honeypot? Gli honeypot: al servizio della sicurezza.
<https://www.kaspersky.it/resource-center/threats/what-is-a-honeypot>
- **Microsoft. (n.d.).**
Che cos'è la sicurezza dei dati nel cloud?
<https://www.microsoft.com/it-it/security/business/security-101/what-is-cloud-data-security>
- **ThousandEyes. (2021, December 7).**
AWS outage analysis: Dec. 7, 2021.
<https://www.thousandeyes.com/blog/aws-outage-analysis-dec-7-2021>
- **Inbound Fintech. (2025).**
Lessons from the Cloudflare outage 2025.
<https://www.inboundfintech.com/blog/lessons-from-the-cloudflare-outage-2025>
- **Cloudflare. (2026, February 20).**
Cloudflare outage February 20, 2026.
<https://blog.cloudflare.com/cloudflare-outage-february-20-2026/>
- **LastPass. (2022, June 2).**
Annuncio di un recente incidente di sicurezza.
<https://blog.lastpass.com/it/posts/notice-of-security-incident>
- **Net Zero Insights. (n.d.).**
Data centers environmental cost.
<https://netzeroinsights.com/resources/data-centers-environmental-cost/>
- **MapMetrics. (n.d.).**
Navigando verso un futuro più verde: riduzione CO₂.
<https://mapmetrics.org/it/blog/navigando-verso-un-futuro-piu-verde-riduzione-co2/>
- **1111 Systems. (2021).**
iland 5265 cloud economics whitepaper (Revision 1).
<https://1111systems.com/wp-content/uploads/2021/07/iland-5265-cloud-economics-whitepaper-r1.pdf>
- **NNGroup. (2024, January 25).**
Mental models and user experience design.
<https://www.nngroup.com/articles/mental-models/>