Grenze International Journal of Engineering and Technology, June Issue

A Biophilic Approach to Digital Fragmentation and Engagement: Mycelial Hub

Shweta Yenorkar[1], Mayank Shastri[2] and Nishchay Ninawe[3]
[1]Department of Electronics Engineering, Ramdeobaba University, Nagpur, India
Email: yenorkarsp@rknec.edu
[2-3]Department of Electronics and Computer Science Engineering, Ramdeobaba University, Nagpur, India
Email: 25shastrimn@rbunagpur.in, 25ninawens@rbunagpur.in

***Abstract*— Several productivity tools currently available often force users to choose between user privacy and convenience. Existing systems are typically difficult to use due to complex documentation and instructions. Herein we present Mycelial Hub, a work space that aims to improve user motivation while keeping user data secure. This system uses a biophilic design based growth system called "The Vigor Algorithm" designed to calculate user productivity as a score that can be visually displayed and Shannon entropy to track work that can differentiate between shallow activity and hard work. To examine how reliable this design is, we created two simulations to see if the system can differentiate between deep work and shallow activity. The initial results showed two scores ($\Delta G \approx 66.9$) for deep work and ($\Delta G \approx 30.4$) for shallow activity, clearly showcasing that the higher score is obtained by the user doing deep work. However, the current evaluations are limited to these simulations rather than actual user tested data. These findings do suggest that privacy friendly work spaces and convenience can indeed be integrated together without sacrificing one.**

***Index Terms*— Productivity, Local-First, Biophilic Design, Gamification, Cryptography, Distributed Systems.**

I. Introduction

It has now become common for digital tools for note taking and they are widely used in both academic and professional work. But even though it has become this popular, its implementation is quite lacking. In actual day to day usage, many basic features such as planning your day-to-day life, taking your lectures notes, tracking your habitual progress, or using calendars for scheduling are handled in separate applications even though these usually go together in practice. This makes it difficult to keep up with everything as there's no single workspace.

Productivity platforms available right now such as Notion/Obsidian tend to introduce features such as nested databases, custom workflows, plugin systems, and highly customizable layout. While impressive and useful, they tend to be extremely complicated and difficult to use and requires hours of documentation reading. Many interfaces are rather complicated with intricate features that require hours of learning and reading their documentations. Meanwhile productivity tools tend to heavily focus on sending more and more distracting notifications to its users.

Grenze ID: 01.GIJET.12.2.114_22

To address all of these issues the paper tries to give appropriate solutions using algorithms that are focused on integrating these features together while keeping it simple, engaging and intuitive to use. Rather than treating these features as separate components, we try to connect them that shows actual user behaviour.

II. Related Work and Theoretical Foundations

To locate Mycelial Hub in relation to the larger HCI landscape, we look at three other domains.

A. Behavioural Design

The developers used numerous elements of design in Mycelial Hub, for instance, self-motivation to represent focus, flow, and productivity. The system avoids the repeating triggers from the Fogg Behaviour Model [5] and constant notifications. Most applications use gamification which creates a transactional mindset and slightly counterproductive outcomes; however, Mycelial Hub avoids the Over justification Effect. Numerous nature inspired visuals [2] give the environment shape outlining areas where users can settle into a workspace for deep work [6].

B. Calm Technology

Furthermore, Weiser [1] claims that ambient features are used throughout the interface in Mycelial Hub because technology should communicate passively versus actively. The development team implemented a realistic amount of visual transition from “Passive Glow” to standard “Push” notifications. The different rates at which lights within the environment adjust create the appearance of glowing within the visual environment, particularly as it relates to deadlines. Keeping the user informed with this type of angle towards decreasing cognitive load will decrease the likelihood of distracting them from using typical alarms.

C. The CAP Theorem

In distributed systems, we typically have to make a choice a difficult one between consistency and offline availability which is quite annoying. In fact, the vast majority of cloud apps are server-first which means they fail when the internet is down. We took the opposite Instead, following the "Local-First" approach [3], we prioritize Availability and Partition Tolerance. Local device — we take this to mean that the primary database is the profile data (estimated stores) of a user. (1) The user remains the full owner of their work, even with network coverage or not.

III. System Architecture

Core Philosophy: Data Sovereignty We built Mycelial Hub on a simple rule: the user owns their data, not the server.

A. Local-First, Cloud-Optional Model

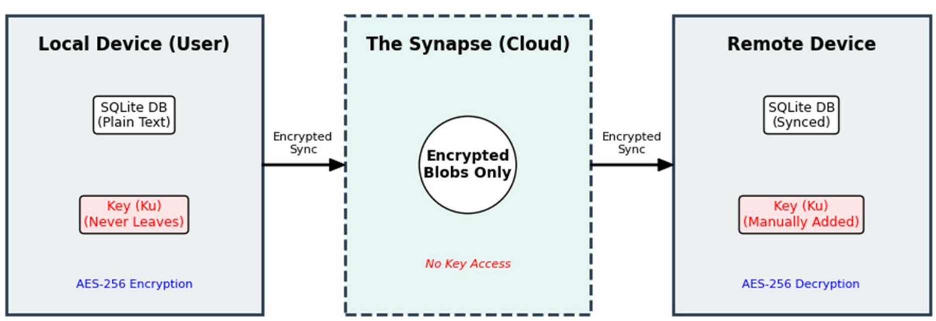

Figure 1. System Architecture. Data is stored locally on the user’s device and synced in encrypted form through the cloud, without exposing the user’s data or keys.

Our framework is designed in a way such that the data is primarily stored on the user's own computer instead of relying on a server. To make things clearer, let us assume that the information stored in the device corresponds to S_L (local state), while the one saved on the server can be referred to as S_R (remote state). In contrast to most applications using cloud storage, in our case, it is the local data that serves as the basis of operation, whereas the server acts only as a mediator. Before being uploaded to the server, the information is encrypted by means of the local key. Specifically, we implement a symmetric encryption algorithm (AES-256), whereby the same key is used both when the information is encrypted and decrypted. It should be noted that the key is generated on the user's device and never transferred to the server. Therefore, the data stored in the server remain inaccessible to it.

$$D_{cloud} = E(D_{local} \cdot K_u) \quad (1)$$

In this case, D_{local} stands for the original data held by the user on his device while K_u stands for the key created locally. $E(.)$ stands for the encryption process. D_{cloud} is the encrypted data sent to the server.

IV. DATA MODELLING AND STORAGE SCHEMA

Our data is stored using a local relational database that uses SQLite as its backbone. As a result, both complete history of accomplished tasks and correct data on sessions are ensured. Contrary to the complicated schemes used in cloud services, our scheme is relatively simple, containing only four main parts. In the first part, we keep track of user data, storing individual weights and utilizing them in order to measure the productivity of all involved entities. Moreover, it stores all tasks made by users, including their descriptions, deadlines, and accomplishment. Secondly, we also have sessions that include the duration of each and what kind of productivity the session was for. Additionally, we describe the development of our algorithm, including inactive sessions and changes in productivity.

V. CORE ALGORITHMS AND FEATURES

This section talks in detail about the mathematical and algorithmic foundations of the system.

A. The Visual Feedback Module ("The Mycelium")

Our main interface is the Mycelium, which glows brightly or glows down based on how much work the user gets done in a specified time (for example: a semester). Our decision to move away from classic loading bars comes from their inability to properly visualize work done in a creative manner to motivate someone. As such we view the concept of productivity as a network which is alive and driven by a metric we call as "Vigor".

The Vigor Function

To detect the "Digital Metabolism" we harvest a running record of Mt for the end user in each work session, and the system calculates Growth Delta (ΔG). The growth of the system is driven by a combination of task completion, time spent focusing, habit consistency, and note-taking activity. The system doesn't treat these metrics separately; it combines them to reflect overall user activity.

$$\Delta G = \alpha \ln(1 + T_c) + \beta \left(\frac{F_m}{60}\right) + \gamma H_s + \delta \left(\frac{N_{active}}{60}\right) \sigma \quad (2)$$

- $\alpha, \beta, \gamma, \delta$ (Weight Coefficients): They could be set by the users themselves according to their needs. Right now, it is done in order to change the algorithm according to users' preferences for the workflow. For instance, a user may have the condition like $\beta > \alpha$ meaning that the user wants to put more weight on "Deep Work" (Focus Time) then on "Task Volume."
- T_c (Task Completion): This is the number of tasks that have been completed during the present session. We do not allocate the same weight to all the tasks; rather, we apply the $\ln(1 + T_c)$ rule. This will ensure that the frequency of performing basic tasks does not carry more weight in the score, implying that performing multiple basic tasks will carry less weight than performing one big task.
- F_m (Focus Minutes): Sum of the duration of all Pomodoro's in minutes. Here $\frac{F_m}{60}$ is used so that the measurement units of this metric were converted to the units of hours to align it with other metrics in terms of scale.
- H_s (Habit Streak): It represents the consistency of the user over a specific period. H_s refers to the number that falls within the range of 0.0 to 1.0.
- N_{active} (Active Knowledge Generation): The time spent on note taking is taken using this variable. Like focus time, it is normalized to hours as by $\frac{N_{active}}{60}$ to fit the scales of the other variables.
- σ (Entropy Coefficient): This is a multiplier, which varies depending on how complex an activity is during a session.
- The Vigor formula comes with certain constraints on behavior, in order to balance other variables. The ($\alpha \ln(1 + T_c)$) log term exists "to prevent gamification fatigue." We purposely favored Focus Minutes and Active Knowledge Generation, so as to avoid the "productivity hack." Otherwise, it's very easy to do

nothing but churn out quick tasks to increase your Mycelium network. However, this weighting system does just the opposite - you end up spending less time on task completion because you're slowing down visual progress. This kind of bacterial growth mimics natural biological systems; initially rapid but already reaching steady state.

- In order to make sure that N_{active} accounts for actual work effort, we define it as the Entropy-Validated Duration:

$$N_{active} = \sum_{t=0}^{T}(\Delta t \cdot V(t)) \qquad (3)$$

Where Δt is the sampling interval (1 minute) and $V(t)$ is the validity function derived from Shannon Entropy (Fig. 2):

$$H(B) = -\sum_i P(x_i)\log_2 P(x_i) \qquad (4)$$

The validity check determines if the time block counts:

$$V(t) = \begin{cases} 1, & \text{if } H_{min} < H_{(B)} < H_{max} \\ 0, & \text{otherwise} \end{cases} \qquad (5)$$

- H_{min} Rejects low-entropy inputs (e.g., repetitive characters like "jjjjjj").
- H_{max} Rejects high-entropy noise (e.g., random character strings characteristic of bots).

If $V(t)$ = 0, the interval is discarded, ensuring that the Mycelium only grows from verifiable, high-quality academic work.

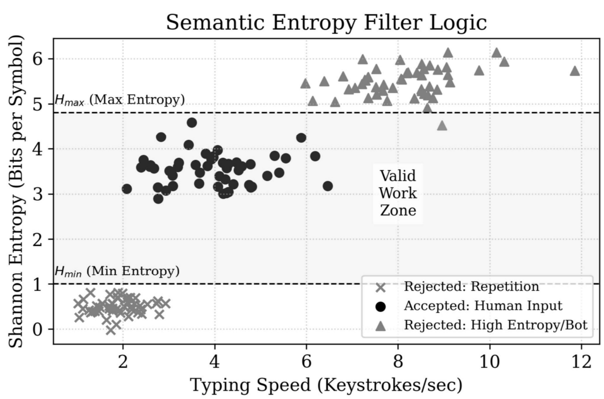

Figure 2: Semantic Entropy Filter: Values in the range (H_{min}<H<H_{max}) are considered for input purposes, all kinds of repetitions including those of "jjjjj" and alphabets in a random manner are ruled out.

The Decay Mechanic

In order to simulate the phenomenon of "wilting," the starting point would be the following formula, which describes exponential decay and depends on the current value:

$$\frac{dV}{dt} = -\lambda V \qquad (6)$$

Therefore, we obtain a common formula for the function: $V(t) = V_0 e^{-\lambda t}$. This is applicable to the continuous case; nevertheless, it is caused by the behavior of the users, who work in the form of sessions, but not in the form of a flow. Namely, it turns out that we need to make some changes in the formula for it to be applicable in our particular case. Decay will occur depending on the activity performed at specific time moments. Thus, at every interval, the value of Vdensity will firstly increase by ΔG, which is achieved at the previous session, and decay will occur according to Δt.

This gives the update rule:

$$V_{density}(t+1) = \left(V_{density}(t) + \Delta G\right)e^{-\lambda \Delta t} \qquad (7)$$

The Seasonality Mechanic

In other to avoid fatigue of infinite growth, Mycelium introduces a "Seasonality" cycle. At the end of a time period (for example, a semester), the network "blossoms" and renews itself, creating an image of the user's garden that stores to their personal archive. This would also enable players to see their progression in separate chapters, rather than a single grind that doesn't seem to end.

B. Differentiated Alert System

Consider a deadline that is just two days before. In a typical system, just the fact that a reminder was scheduled is enough to create an alert. In Mycelial Hub, when the value of T_{rem} is high and $I_{param} = I_{score}$, which results in a low Interruption Score (I_{score}) and only triggers Passive Glow. As the deadline approaches ($T_{rem} \rightarrow 0$),the denominator decreases exponentially, exponentially increasing I score. This activates the alert from passive to active only where cost of missing due-date is larger than cost of interruption in the current user focus.

Interruption Logic

For any given session, the I_{score} is calculated as:

$$I_{score} = \left(\frac{W_{imp}}{T_{rem}+\epsilon}\right) C_{ctx} \quad (8)$$

- W_{imp}: User-defined weight of importance (1-10).
- T_{rem}: Time remaining until the deadline (in hours).
- ϵ: A small constant to prevent division by zero.
- C_{ctx}: Context awareness factor (e.g., is the user currently in a Focus Session?).

$$f(x) = \begin{cases} \text{Passive Glow,} & \text{if } I_{score} < \theta_1 \\ \text{System Notification,} & \text{if } \theta_1 \leq I_{score} < \theta_2 \\ \text{Active Modal (Alarm),} & \text{if } I_{score} \leq \theta_2 \end{cases} \quad (9)$$

This ensure that users are only forcefully interrupted (Active Modal) for critical, imminent deadlines (θ_2) preserving their focus for deep work during all other times.

C. Unified Task Prioritization

All the three aspects mentioned above, that is, the user's behavior, his assigned job, and the project activities, will be combined in a "Next Action" view.

Priority Algorithm

We went a step further by coming up with a priority algorithm that takes into consideration the sense of urgency and the effort needed to complete the activity in question. This ensures that the tasks chosen can be accomplished within a single session and without overlooking any important task.

$$P_{task} = \frac{1}{(D_{due} - D_{now})} + \mu\left(E_{effort}\right) \quad (10)$$

In the above formula, μ is the coefficient that represents the influence of the E_{effort} parameter on a chosen task.

D. The Semantic Editor: Multimodal Input & Abstraction

To resolve the friction often associated with "geeky" local-first tools (like raw Markdown editors), Mycelial Hub implements a Hybrid Interaction Layer for its note-taking engine.

GUI-Driven LaTeX Abstraction

When it comes to local-first tools, the most glaring flaw is having to write raw code (e.g., LaTeX) in order to present math. This is a high bar to entry. Mycelial Hub does this by separating the Storage Format from the Input Method. On the front end of the app we have a "WYSIWYG" toolbar, but on the back end, we're still recording MathJax-able Markdown. This allows the user to click on algebraically complex pieces as for instance an integral or sum, and so forth without remembering all of the syntax.

Vector-Based Annotation Layer (Drawing)

Linear text is just simply too restrictive when you have to do some academic brainstorming. To solve this problem, we have developed a Spatial Annotation system that supports direct writing over/on top of notes. We built this system by stacking an SVG vector canvas (Layer 1) which lies on top of the normal text DOM (Layer 0). Those drawings are anchored to specific block IDs, so when you modify the text and the paragraph jumps, your sketches jump right along with it.

E. Context-Aware Calendar System

- Most apps treat 'Tasks' (to-do lists) and 'Events' (calendar) as two entirely different things. Instead we take a different approach by treating the two as the same object. In our system, an entry is flexible, a simple task in your backlog can remain a simple task until you pick it up and drag it on the calendar and the system gives it start and end times making it become a 'Focus Session', without any shift in its data structure.
- Regular calendars usually are not good at managing your academic semesters. We have built a native support for Recurring Academic Cycles (e.g. 'Every Monday at 10 AM for 14 weeks'). Unlike a common recurring event, our system creates a distinct session for each class. This way, you can tie some specific lecture notes or slides to that exact date and have your semester-long archive partially automated.

VI. Comparative Market Analysis

To prove that Mycelial Hub is needed, we performed a function comparison with the market leader in terms of Privacy, Integration and Motivation.

A. The "Trilemma" of Productivity Tools

For providing a context to the contributions made by Mycelial Hub, we examine architectural constraints of existing state-of-the-art technology. The forces of the market today compel users to decide on any two out of three properties they want, resulting in a "Productivity Trilemma":

- Cloud-Native (e.g., Notion): Prioritizes Integration and Motivation but compromises Privacy due to server-side data access.
- Local-First (e.g., Obsidian): Prioritizes Privacy and Integration but often lacks native Motivation (gamification) layers.
- Gamified Utilities (e.g., Forest): Prioritizes Motivation but lacks the Integration required for a full workspace.

Mycelial Hub does not seek to replace these tools but rather attempts to synthesize their distinct advantages into a single architecture. Table 1 illustrates this design space gap.

TABLE I. Comparative Analysis of Feature Sets Across Major Productivity Paradigms

Feature	Notion	Obsidian	Forest
Architecture	Cloud-Native	Local-First	Cloud-Native
Storage	Proprietary	Markdown	Proprietary
Motivation	Manual	Plugins	Gamified
Privacy	TLS	Local Only	TLS
Team & Sync	Yes	Complex	No

B. Analysis Matrix

Mycelial Hub is designed as an architectural proof-of-concept that aims to address all three criteria simultaneously, filling the gap between strictly local tools and cloud-native platforms.
This comparison highlights that while features exist in isolation, no current solution integrates with a Secure Workspace, confirming the unique value proposition of our research.

VII. Implementation Strategy

A. Technology Stack

The prototype is developed using Python for its robust libraries in data handling and GUI development.

- Backend Logic: Python 3.12 (utilizing sqlite3 for local storage and cryptography library for E2EE).
- Frontend Interface: Custom Tkinter/PyQt framework for rendering the procedural mycelial graphics.
- Procedural Generation:The networkx and matplotlib libraries are utilized to simulate the branching logic of the mycelial network.

VIII. Cryptographic Protocols and Security

The security architecture stresses a "No-Compromise" privacy model where the server serves only as a blind relay. We perform a Zero-Knowledge architecture: On account creation we derive the 256-bit Master Key (K_m) locally using PBKDF2. Prior to synchronisation, the local data are encrypted with AES-256-GCM ($C=\text{AES-GCM}(D,K_m,IV)$); only the cipher text is sent, guaranteeing that K_m never exits its user devices. For cooperation, we use Asymmetric Key Exchange (RSA-4096) to privately exchange session keys. Finally, to mitigate key loss without backdoors, we present a construction (k=2,n=3) where rebuilding the key requires both physical paper key and server-side share thereby avoiding server-only restoration.

IX. Theoretical Simulation And Scenario Analysis

Due to the lack of longitudinal user data, we assess (Eq. 2) and Alert Logic (Eq. 7) through a deterministic simulation.

We simulate asynchronous feedback from two contrasting user personas to illustrate how the system separates between "Healthy Productivity" and "Toxic Activity."

A. Scenario Setup

We define two hypothetical user behaviors over a 4-hour session (240 minutes):

- User A ("The Deep Worker"): Works for 3 hours on a single complex task. High Focus Time ($F_m = 180$), Low Task Count ($T_c = 1$), High Complexity ($\sigma = 1.25$).
- User B ("The Task Spammer"): Spends 3 hours clearing trivial emails. Low Focus Time ($F_m = 0$), High Task Count ($T_c = 20$), Low Complexity ($\sigma = 1.0$).

B. Vigor Growth Simulation

We apply the Vigor Function:

$$\Delta G = \alpha \ln(1 + T_c) + \beta \left(\frac{F_m}{60}\right) + \gamma H_s + \delta \left(\frac{N_{active}}{60}\right) \sigma$$

Applying the Vigor Function with constants $\alpha = 10$, $\beta = 20$, $\gamma = 5$ yields a significant disparity in growth scores:

- Calculating User A (Deep Worker): Task Term: $10 \ln(1 + 1) \approx 6.9$. Focus Term: $20\left(\frac{180}{60}\right) = 60$. Total Growth: $\Delta G_A \approx 66.9$ units.
- Calculating User B (Task Spammer): Task Term: $10 \ln(1 + 20) \approx 30.4$. Focus Term: $20\left(\frac{0}{60}\right) = 0$. Total Growth: $\Delta G_B \approx 30.4$ units.

This confirms that the logarithmic damping term ($\alpha \ln(1 + T_C)$) successfully prevents the system from rewarding shallow 'busy work' over deep work [4].

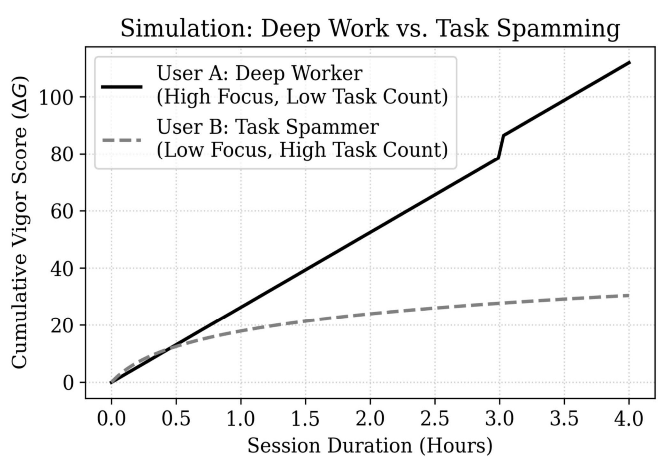

Figure 3: Simulation results comparing Deep Work and Task Spamming behavior over time.

C. Alert Threshold Simulation

We simulate the Interruption Score (I_{score}) for a deadline approaching in T_{rem} hours. Equation: $I_{score} = \frac{10}{T_{rem} + 0.1}$ (assuming $W_{imp} = 10, \epsilon = 0.1, C_{ctx} = 1$) (Fig. 4).

TABLE II. SIMULATION OF (I_{score}) VS. TIME REMAINING

Time Rem ($T_{\{rem\}}$)	I_{score}Calc	Alert Tier
48 Hours	10/48.1 ≈ 0.2	Passive
24 Hours	10/24.1 ≈ 0.4	Passive
4 Hours	10/4.1 ≈ 2.4	Notification
1 Hour	10/1.1 ≈ 9.0	ALARM
15 Mins (0.25h)	10/0.35 ≈ 28.5	CRITICAL

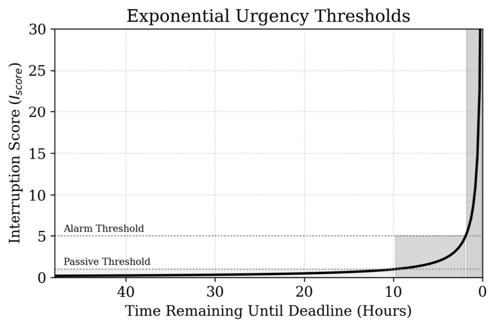

Figure 4. Simulation results: The score remains low until the final hour

X. CONCLUSION

This paper outlines the theoretical groundwork and blueprint for Mycelial Hub as a possible solution to combine user participation and digital privacy into one integrated system. The paper also highlights that a local-first architecture may offer a way more feasible resolution to support user control over their personal data while still having a simple and intuitive workspace. The comparative study in Section VI also displays that tools available often separate privacy, integration, and engagement rather integrate them into one singular system.

Since this is a design aiming towards being persuasive the paper also considers the ethical concerns of the visual feedback by keeping the focus on supporting motivation without relying too much on distracting notifications. Although at present the paper is limited to only simulations. Since the current evaluation is based on theoretical analysis and simulation, the results may not fully capture how users respond to the system over longer periods of use. As such the for future work the aim it to do longitudinal studies over users long term data. Additional work may also include refining the Entropy coefficient with machine learning capabilities to get even better results on classifying deep work while preserving privacy.

ACKNOWLEDGMENT

The authors wish to thank Ramdeobaba University for providing the necessary infrastructure to complete this work.

REFERENCES

[1] M. Weiser and J. S. Brown, "The Coming Age of Calm Technology," Xerox PARC, 1996.
[2] S. R. Kellert, "Biophilic Design: The Theory, Science and Practice of Bringing Buildings to Life," Wiley, 2008.
[3] M. Kleppmann, A. Wiggins, P. van Hardenberg, and M. Herlihy, "Local-First Software: You Own Your Data, in spite of the cloud," Onward! ’19, ACM, 2019.
[4] C. Newport, Deep Work: Rules for Focused Success in a Distracted World. Grand Central Publishing, 2016.
[5] B. J. Fogg, Persuasive Technology: Using Computers to Change What We Think and Do. Morgan Kaufmann, 2003.
[6] M. Csikszentmihalyi, Flow: The Psychology of Optimal Experience. New York: Harper & Row, 1990.
[7] J. Clear, Atomic Habits: An Easy & Proven Way to Build Good Habits & Break Bad Ones. Avery, 2018.
[8] N. Eyal, Hooked: How to Build Habit-Forming Products. Portfolio Penguin, 2014.
[9] M. Shapiro et al., "Conflict-free replicated data types," in Proc. 13th Int. Symp. Stabilization, Safety, and Security of Distributed Systems (SSS), 2011, pp. 386–400.
[10] C. L. North, "The user’s role in the privacy of personal data," Communications of the ACM, vol. 63, no. 10, pp. 29–31, 2020.