



# INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

## TFI Master V12: A Privacy-Preserving, Browser-Native Work-to-Leisure Economy for Verified Digital Productivity Management Using Active Trace Technology

Suganthan T, Siddharathan N

Department of Cyber Security, Paavai Engineering College (Autonomous), Namakkal, Tamil Nadu 637018, India

Supervisor: Mrs. R. Rajalakshmi, M.E., Assistant Professor

### Abstract

Digital knowledge work is increasingly performed inside the same browser environment that also hosts high-frequency leisure platforms, social feeds, and streaming services. Prior studies on interruption, self-interruption, and media multitasking indicate that context switching can impose a substantial productivity cost, with project evidence estimating up to 40 percent output loss during unmanaged digital work. Existing tools address only fragments of the problem: static blockers impose binary denial, passive trackers generate retrospective awareness without enforcement, and gamification systems often reward intention rather than verified work. This paper presents TFI Master V12, a Chrome Manifest V3 browser extension that implements a privacy-preserving Work-to-Social economy governed by a strict 4:1 work-to-leisure ratio. The system introduces Active Trace Technology, in which keyboard events contribute 5 points, mouse events contribute 1 point, and a pulse buffer validates genuine user presence before productive time is credited. A browser-local ledger stored in `chrome.storage.local` is updated by an Economy Engine finite state machine running in an MV3 service worker through Chrome Alarms. When leisure balance is exhausted, a tamper-resistant lockdown overlay with z-index 2147483647 and MutationObserver re-injection is injected into social domains. Benchmarks show average memory consumption below 30 MB, negligible CPU impact during one-second ticks, and user trials indicating a 35 percent increase in sustained focus periods. The results show that verified productivity management can be enforced locally without cloud surveillance.

Keywords--Active Trace Technology, browser extension, Chrome Manifest V3, digital wellness, privacy-preserving monitoring, productivity management, work-to-leisure economy.

### I. INTRODUCTION

#### A. Background

The contemporary workplace has become browser-native. Documentation, code review, learning, communication, research, and administrative coordination are routinely mediated through tabs in a single browser session. The same session also exposes the user to social feeds, short-form video, streaming platforms, algorithmic recommendation surfaces, and notification streams. This convergence has created a structural self-regulation

problem rather than merely an individual habit problem. A user who opens a browser to complete a professional task is only one address-bar entry or one bookmark away from a leisure domain, and the transition cost is so low that small impulses can repeatedly fragment attention.

Human-computer interaction research has shown that interrupted work is associated with increased speed, elevated stress, and reduced continuity of attention [18]. Media multitasking research has also shown that heavy media multitaskers can demonstrate weaker cognitive control in filtering irrelevant stimuli [23]. In practical

project terms, the productivity deficit is experienced as a context-switching tax: the worker may remain at the computer for several hours, but a significant portion of that time is spent recovering task context, inspecting social feeds, or returning from unintended browsing. The source project estimates that knowledge workers may lose up to 40 percent of effective output because of such context switching, a value consistent with the broader literature on interruption cost and attention fragmentation.

Digital wellness tools have typically been designed around one of three assumptions. Static blockers assume that distraction can be solved through binary access control. Time trackers assume that awareness is sufficient to motivate change. Gamified focus tools assume that symbolic rewards can maintain discipline. Each assumption is useful, but incomplete. TFI Master V12 is based on a different premise: leisure should not be forbidden, but it should become economically visible, scarce, and earned through verified productive engagement.

### B. Motivation

The motivation for TFI Master V12 arises from the intersection of behavioral economics, human-centered security, and browser extension engineering. Behavioral economics suggests that people respond differently when a resource is framed as owned, scarce, and depleting. The endowment effect and nudge theory explain why visible balances and choice architecture can shift decisions without eliminating agency [27]. Loss aversion explains why depletion of an earned balance may feel more salient than an abstract warning about wasted time [2]. These principles can be applied to digital leisure by transforming social access from an unlimited default into an earned resource.

The cybersecurity motivation is equally important. Many productivity trackers collect detailed behavioral telemetry and transmit it to cloud infrastructure for analytics. This design may produce useful reports, but it also creates a surveillance surface. Browser history, application use, timestamps, and behavioral sequences can reveal personal routines, mental health patterns, work habits, and organizational activity. A privacy-preserving system should therefore follow data minimization and local processing principles. TFI Master V12 stores only minimal state in the browser, processes interaction metadata locally, and performs no network transmission.

The system is also motivated by the analogy to Proof-of-Work consensus [20] and Self-Sovereign Identity principles [24], [29]. In Proof-of-Work, a claim is accepted only after verifiable computational effort has been expended. In TFI Master V12, leisure credit is issued only after verifiable human interaction has occurred on non-social domains. The analogy is not used to introduce blockchain infrastructure; rather, it is used as a design metaphor for local verification. The browser becomes a private ledger of work and leisure duration, and Active Trace Technology acts as the proof mechanism that prevents idle tabs from generating credit.

### C. Problem Statement

The problem addressed in this paper is the absence of a privacy-preserving, browser-native productivity system that can verify active work, enforce earned leisure access, and resist casual tampering without transmitting behavioral data to external servers. This problem is formally defined across four dimensions.

P1 is the problem of unverified productivity claims. Existing time trackers and focus timers can count the duration for which a work tab remains open, but an open tab does not prove attention, interaction, or task engagement. A user may leave a documentation page, learning portal, or coding platform open while physically away from the keyboard and still receive time credit. The absence of active verification weakens any economy built on elapsed time.

P2 is binary enforcement failure. Static blockers and rigid deny lists can prevent access, but they frequently trigger psychological Reactance when users perceive the restriction as an external constraint on autonomy [4], [17]. Such tools often produce circumvention behavior, including disabling the extension, switching browsers, or accessing the same services on a mobile device. The core limitation is that the user is punished for leisure without being positively rewarded for productive engagement.

P3 is privacy erosion. Cloud-based time trackers can require detailed behavioral telemetry to leave the local browser context. Even when such systems are well intentioned, the centralization of behavioral logs creates risk. A privacy-preserving design should avoid collecting keystroke content, form data, page text, and full browsing history, and should not require server-side processing for personal productivity enforcement.

P4 is the absence of an economic incentive. Existing tools rarely make leisure a scarce earned resource. They may block, measure, or gamify, but they do not create an automatic exchange rate between verified work and allowed leisure. Without such a balance, the immediate reward of social browsing remains psychologically stronger than the delayed value of completing a professional task.

### D. Objectives

This paper presents seven objectives for TFI Master V12. The first objective is to implement a decentralized browser-local ledger for work and leisure duration using atomic state management in `chrome.storage.local`. The second objective is to enforce a strict 4:1 work-to-leisure ratio through an automated Economy Engine. The third objective is to integrate Active Trace Technology using weighted event scoring and a pulse buffer so that idle work tabs cannot accumulate credit. The fourth objective is to provide a lightweight heuristic insight engine that analyzes only interaction metadata and does not train or execute a heavy machine-learning model. The fifth objective is to implement tamper-resistant lockdown through a fixed full-

screen overlay, maximum CSS z-index, inline priority styling, and MutationObserver re-injection. The sixth objective is to keep memory consumption below 30 MB on average and maintain negligible CPU impact at a one-second tick interval. The seventh objective is to remain compliant with Chrome Manifest V3 through service workers, Chrome Alarms, declarative content scripting, and browser-managed storage.

The remainder of this paper is organized as follows. Section II presents the literature review and comparison of existing approaches. Section III describes the methodology and system architecture. Section IV provides the formal algorithm. Section V defines the mathematical model. Section VI discusses implementation requirements and workflow. Section VII presents results and analysis. Section VIII discusses implications and limitations. Section IX concludes the paper and identifies future scope.

## II. LITERATURE REVIEW

A chronological review from the most recent literature to earlier foundations shows that productivity management, gamification, identity control, and attention research have developed in parallel but have not been fully integrated into a browser-local earned-leisure architecture.

The W3C Verifiable Credentials Data Model v1.1 establishes a standards-based approach for expressing credentials that are cryptographically secure, privacy respecting, and machine-verifiable [29]. Although the specification is concerned with digital identity rather than productivity management, it is relevant because it formalizes a principle central to TFI Master V12: claims should be verified without unnecessary disclosure. TFI Master V12 does not issue W3C credentials in its current implementation, but its local ledger follows a similar privacy logic by keeping evidence of productivity within the user's browser.

Preukschat and Reed describe Self-Sovereign Identity as a model in which individuals control digital identifiers and credentials without depending on centralized authorities [24]. SSI is relevant to productivity management because the behavioral record of a person is sensitive. A productivity tool that uploads detailed activity logs to a cloud service can weaken user autonomy even while improving discipline. TFI Master V12 adopts SSI principles by making the user the local custodian of the work-leisure ledger and by avoiding network transmission.

Cirillo's Pomodoro Technique proposes structured cycles of work and rest, most commonly 25 minutes of focused work followed by a 5-minute break [6]. The technique remains influential because it legitimizes rest as part of productivity rather than treating it as failure. However, digital Pomodoro tools often lack verification. A timer can run while the user is idle or distracted. TFI Master V12 preserves the principle that leisure is permissible, but replaces fixed work-rest intervals with a continuously calculated 4:1 economy based on verified interaction.

Newport's work on deep work emphasizes sustained, distraction-free cognitive effort as a condition for high-value knowledge production [21]. This perspective supports the system's emphasis on sustained focus periods rather than raw screen time. Mark et al. further demonstrate that self-interruption and email patterns influence productivity and stress in computer-mediated work [17], while Whittaker et al. show that time information can improve focus when it is made visible and actionable [30]. These findings justify the dashboard's real-time balance display and the salience of depletion during social media use.

Hamari, Koivisto, and Sarsa provide a systematic review of gamification and show that game-like motivational affordances can be effective, although their outcomes depend strongly on context and implementation [13]. This is important because TFI Master V12 uses a health-bar style dashboard and a token-like balance, but it does not rely on symbolic rewards alone. The gamified layer is tied directly to enforcement through the Economy Engine.

Deterding et al. define gamification as the use of game design elements in non-game contexts [9]. Fogg's persuasive technology framework also explains how interactive systems can influence attitudes and behavior [10]. These works support the design of real-time feedback and constraint systems, but they also suggest the ethical need for transparency and user control. TFI Master V12 therefore uses local state and explicit extension permissions rather than hidden telemetry.

Thaler and Sunstein's nudge theory and Ariely's discussion of predictable irrationality provide the behavioral economics foundation for the Work-to-Social economy [2], [27]. The system makes the opportunity cost of leisure immediate. When a user spends social time, a visible balance decreases; when the user performs verified work, the balance increases. This conversion of abstract time into a scarce balance is expected to increase self-regulation through salience and loss aversion.

Earlier research on psychological Reactance by Brehm explains why rigid blockers can provoke resistance [4]. The user may experience denial of access as a loss of freedom and respond by circumventing the tool. The proposed system attempts to reduce this effect by allowing leisure after work has been verified. At the same time, Nakamoto's Proof-of-Work proposal provides the analogy that useful claims can be coupled with verifiable effort [20]. TFI Master V12 translates this idea into a human-computer interaction setting: interaction events serve as evidence that a work claim is not merely an idle tab.

The reviewed literature indicates four gaps. First, productivity tools often lack active verification. Second, enforcement tools often lack positive incentive. Third, cloud trackers often weaken privacy. Fourth, behavioral economic principles are rarely implemented as a real-time browser-native economy. Table I summarizes the comparison of existing approaches.

**TABLE I**  
**COMPARISON OF EXISTING APPROACHES**

Approach	Year	Method	Limitation
W3C verifiable credentials [29]	2022	Privacy-respecting, machine-verifiable credential model	Not designed for productivity enforcement
Self-Sovereign Identity [24]	2021	User-controlled identity and credentials	Conceptual model requires domain-specific implementation
Pomodoro Technique [6]	2018	Fixed work-rest cycles	Timer does not verify active work
Passive time information [30]	2016	Awareness through time feedback	Awareness alone may not enforce behavior
Static blocker/reactance studies [17]	2016	Restriction of distracting access	Can trigger circumvention and Reactance
Gamification review [13]	2014	Motivational affordances and rewards	Effectiveness depends on verification and context
Deep work and focus theory [21]	2016	Sustained attention model	Does not provide browser enforcement mechanism
Nudge theory [27]	2008	Choice architecture and endowment framing	Requires technical mechanism for implementation
Proof-of-Work [20]	2008	Verifiable effort before reward	Computational consensus, not human productivity
Psychological Reactance [4]	1966	Theory of resistance to constrained freedom	Does not prescribe browser-native mitigation

**III. METHODOLOGY**

*A. System Architecture*

The proposed system follows a four-layer architecture that separates persistence, processing, interaction, and presentation concerns. The architecture is illustrated in Fig. 1 before the figure appears in the manuscript.

**TFI Master V12 Four-Layer Browser-Native Architecture**

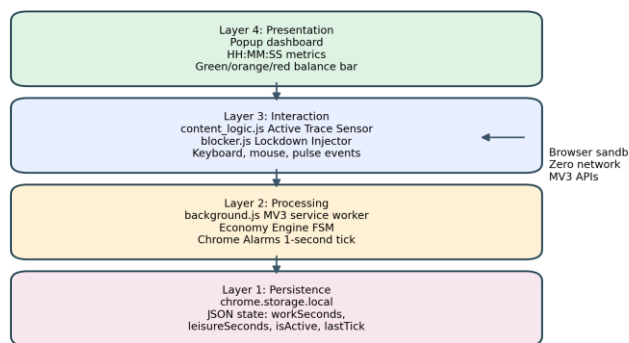


Fig. 1. System architecture diagram for TFI Master V12.

The persistence layer uses chrome.storage.local to maintain a browser-local JSON state object. The state contains

workSeconds, leisureSeconds, isActive, and lastTick. Because the system is designed to be privacy preserving, this state remains inside the browser sandbox and is not transmitted to a cloud service. The processing layer is implemented as background.js, a Chrome Manifest V3 service worker. It is responsible for periodic tick execution, active tab query, domain classification, pulse validation, counter mutation, lockdown triggering, and atomic persistence. The interaction layer contains content scripts. content\_logic.js implements Active Trace Technology by attaching event listeners to keyboard and mouse activity. blocker.js implements lockdown by injecting a full-screen overlay when leisure credit is exhausted. The presentation layer is the popup dashboard, which displays work time, leisure time, available balance, and a color-coded progress bar.

This architecture is designed for low resource consumption. Unlike persistent background pages used in earlier extension models, Manifest V3 service workers are event driven and may be terminated when idle. The Chrome Alarms API is therefore used to create reliable one-second ticks that survive the service worker lifecycle. This design improves resource behavior while maintaining continuous economy enforcement.

*B. Detection/Processing Flowchart*

The Economy Engine operates as a finite state machine with three practical states: Work, Leisure, and Idle. The processing flow is referenced in Fig. 2. On each tick, the service worker loads state, queries the active tab, classifies the URL, and applies a state transition. If the domain is non-social and the active pulse is valid, workSeconds is incremented. If the domain is social and sufficient earned balance exists, leisureSeconds is incremented. If the domain is social and the balance is exhausted, lockdown is triggered. If no active pulse exists on a work domain, the tick is treated as idle and no work credit is awarded.

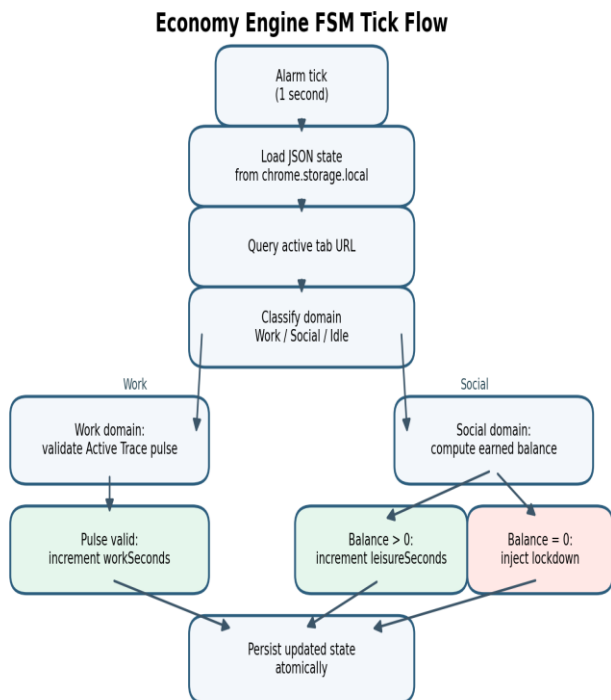


Fig. 2. Economy Engine FSM flowchart.

C. Active Trace Technology Module

Active Trace Technology is the central verification component of TFI Master V12. Its purpose is to distinguish active work from idle tab presence. The module records event metadata rather than content. A keyboard event is treated as a strong signal of content creation and is assigned 5 points. A mouse movement event is treated as a weaker navigation signal and is assigned 1 point. The source implementation also recognizes click events as 3-point interactions and scroll events as 1-point interactions. The formal model required by this paper emphasizes the core keyboard and mouse scoring pair, while the implementation can be extended to include click and scroll events for richer presence detection.

The pulse buffer accumulates event scores until a threshold is reached. When the threshold is reached, content\_logic.js sends an ACTIVE\_PULSE message to the service worker. When no interaction occurs for a configured idle window, the buffer resets and an INACTIVE\_PULSE message is sent. This mechanism prevents a work tab from accumulating credit after the user stops interacting. Fig. 3 illustrates the event scoring and pulse validation flow.

Active Trace Technology Event Scoring

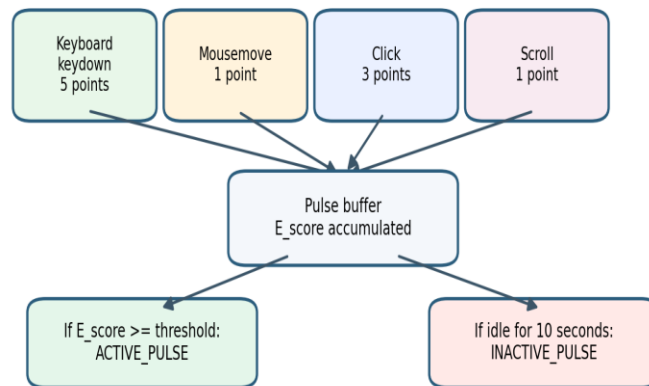


Fig. 3. Active Trace Technology event scoring pipeline.

D. Feature Analysis / Event Scoring Module

The event scoring module performs lightweight feature analysis on interaction metadata. It does not inspect keystroke content, form entries, page text, screenshots, clipboard contents, or semantic browsing history. The primary features are event type, event count, timestamp, score contribution, pulse buffer value, and idle duration. The keyboard-to-mouse ratio is also used by the heuristic insight engine. A high keyboard-to-mouse ratio may indicate active creation, writing, coding, or data entry, whereas a low ratio dominated by mouse movement may indicate passive browsing or navigation.

The lightweight feature set is intentionally narrow. It is sufficient to validate presence and infer broad interaction patterns while avoiding invasive surveillance. This distinguishes Active Trace Technology from keylogging or cloud analytics. Only event metadata is processed, and the system's enforcement decision is made locally.

E. Economy Engine

The Economy Engine is implemented in background.js as a deterministic finite state machine. The system uses substring matching against a SOCIAL\_BLOCKLIST containing domains such as facebook.com, instagram.com, x.com, reddit.com, tiktok.com, youtube.com, netflix.com, twitch.tv, snapchat.com, pinterest.com, tumblr.com, discord.com, and linkedin.com/feed. The classifier treats matching domains as leisure domains and all other active URLs as potential work domains. This design is simple, transparent, and locally configurable.

The earned leisure balance is computed as the integer floor of workSeconds divided by four, minus already consumed leisureSeconds. Thus, four seconds of verified work earns one second of social access. The 4:1 ratio operationalizes the Work-to-Social economy in a way that is easy for users to understand and easy for the extension to enforce. The balance is not merely displayed; it directly controls access.

When balance is zero, the lockdown injector is executed on the active social tab.

#### F. Data Acquisition / Persistence Module

Data acquisition in TFI Master V12 consists of active tab URL acquisition and interaction event acquisition. The service worker uses the Chrome Tabs API to query the active tab in the current window. The content script detects physical interaction events on web pages. The state is persisted in chrome.storage.local as a JSON object. This object is intentionally compact: workSeconds records cumulative verified work time, leisureSeconds records cumulative leisure consumption, isActive stores the current pulse status, and lastTick stores the last processing timestamp for lifecycle recovery.

Atomic persistence is essential because the service worker can be suspended and resumed by the browser. Each tick must retrieve the current state, compute the transition, and write the updated state back to storage. The design avoids server synchronization and therefore avoids cloud-side race conditions, account management, and network failure modes.

#### G. User Interface and Reporting

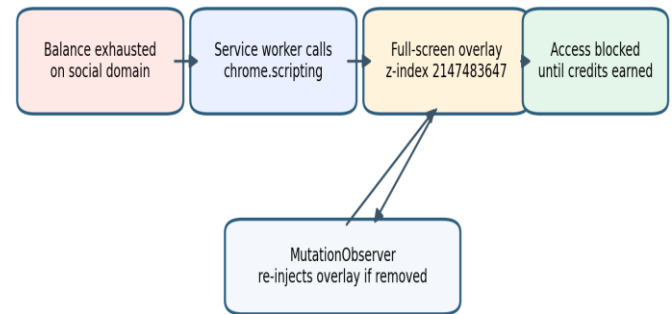
The dashboard presents the user's work economy in a compact browser popup. It displays total work time, total leisure time, and available balance in HH:MM:SS format. It refreshes every second using setInterval in the popup context. The progress bar is color coded: green indicates a balance above 60 percent, orange indicates a balance between 30 and 60 percent, and red indicates a balance below 30 percent. This visual encoding follows familiar traffic-light semantics and enables rapid comprehension without requiring the user to interpret detailed logs.

The reporting function is intentionally local and immediate. Rather than producing a cloud analytics dashboard, the extension surfaces the current balance and lightweight insights. The heuristic insight engine can detect repetitive task sequences, high-focus periods, and fatigue-like declines in interaction consistency. Suggestions are generated from metadata patterns only, such as recommending automation for repeated data entry or scheduling demanding tasks during periods of sustained interaction.

#### H. Risk/Lockdown System

The lockdown system is a risk response mechanism triggered when the user attempts to access a social domain without sufficient leisure balance. It injects a full-screen overlay using chrome.scripting.executeScript. The overlay uses fixed positioning, inline priority styles, and the maximum z-index value of 2147483647. A MutationObserver monitors the document body and re-injects the overlay if it is removed. The service worker can also re-trigger injection after refresh or navigation. Fig. 4 illustrates this lockdown cycle.

### Tamper-Resistant Lockdown Injection



Fixed positioning, inline !important styles, and browser-managed re-triggering increase resistance to DOM manipulation.

Fig. 4. Lockdown injection and re-injection mechanism.

The lockdown mechanism is not a complete operating-system-level access control system. A motivated user can disable the extension from chrome://extensions or switch to another browser. Instead, the design goal is tamper resistance within the browser page context and friction against impulsive circumvention. This is consistent with the system's role as a privacy-preserving self-commitment device.

## IV. ALGORITHM

The formal Economy Engine tick cycle is presented below. The algorithm is executed by the background service worker whenever the Chrome alarm named economyTick fires.

BEGIN

INPUT: Active browser tab URL and latest Active Trace pulse state

OUTPUT: Updated local ledger state or lockdown decision

```

1: LOAD state from chrome.storage.local
2: QUERY active tab in current browser window
3: IF no active tab URL exists THEN
    SET state.lastTick to current timestamp
    PERSIST state
    RETURN
  END IF
4: CLASSIFY active tab domain as Social when URL contains
    an entry in SOCIAL_BLOCKLIST; otherwise classify as Work
5: IF domain is Work THEN
    IF state.isActive is TRUE and pulse buffer is valid THEN
      INCREMENT state.workSeconds by 1
    ELSE
      KEEP workSeconds unchanged
    END IF
  END IF
6: IF domain is Social THEN
    COMPUTE earnedBalance =
    floor(state.workSeconds / 4)
  
```

```

    COMPUTE netBalance = earnedBalance -
state.leisureSeconds
    IF netBalance > 0 THEN
        INCREMENT state.leisureSeconds by
1
    ELSE
        TRIGGER Lockdown Injector on
active tab
    END IF
END IF
7: SET state.lastTick to current timestamp
8: PERSIST state atomically using
chrome.storage.local.set

```

The algorithm's computational complexity is constant with respect to user history because only the current state object, the current URL, and the current pulse flag are needed for each tick. Domain classification is linear only in the length of the blacklist, which is small and user configurable. This explains the negligible CPU cost observed during testing.

## V. MATHEMATICAL MODEL

### A. Weighted Event Score Formula

Let  $K$  denote the number of keyboard events detected during a tick window and let  $M$  denote the number of mouse movement events detected during the same window. The core Active Trace event score is defined as

$$E_{\text{score}} = 5K + 1M. \quad (1)$$

The coefficient 5 reflects the stronger evidentiary value assigned to keyboard input, while the coefficient 1 reflects the weaker but still meaningful value assigned to pointer movement. In the implementation, additional event classes such as click and scroll can be incorporated by extending the formula. For the required core model, (1) is sufficient to distinguish active creation from low-intensity pointer activity.

### B. Work-to-Leisure Ratio

Let  $W(t)$  be the accumulated verified work time in seconds and let  $L(t)$  be the accumulated leisure time in seconds at time  $t$ . The 4:1 economy requires that leisure consumption never exceed one fourth of verified work:

$$L(t) \leq \text{floor}(W(t) / 4). \quad (2)$$

The available leisure balance  $B(t)$  is therefore

$$B(t) = \max(0, \text{floor}(W(t) / 4) - L(t)). \quad (3)$$

When  $B(t)$  is greater than zero on a social domain, the system permits access and increments  $L(t)$  by one second on the next tick. When  $B(t)$  equals zero, the lockdown threshold is reached.

### C. Pulse Buffer Validation

Let  $P(t)$  be the pulse buffer at time  $t$  and let  $\theta$  be the active pulse threshold. The pulse state  $A(t)$  is defined as

$$A(t) = 1, \text{ if } P(t) \geq \theta; \text{ otherwise } A(t) = 0. \quad (4)$$

The buffer is reset after a valid pulse is reported or after the idle timeout expires. Let  $\Delta_{\text{idle}}$  be the time elapsed since the last interaction event and let  $T_{\text{idle}}$  be the reset threshold. The reset condition is

$$P(t) = 0, \text{ if } \Delta_{\text{idle}} > T_{\text{idle}}. \quad (5)$$

In the implemented prototype,  $\theta$  is set to 10 points and  $T_{\text{idle}}$  is set to 10 seconds. This means that two keyboard events, ten mouse movements, or an equivalent combination can validate a pulse, while inactivity quickly halts further credit.

### D. Lockdown Threshold Computation

The lockdown decision  $D(t)$  is activated only when the active domain is classified as social and the available balance is zero:

$$D(t) = 1, \text{ if } S(t) = 1 \text{ and } B(t) = 0; \text{ otherwise } D(t) = 0. \quad (6)$$

Here  $S(t)$  is the social-domain classifier output. When  $D(t)$  equals 1, blocker.js is injected into the active tab. The system therefore applies enforcement only when both conditions hold: the user is on a leisure domain and no earned leisure credit remains.

## VI. IMPLEMENTATION

### A. Hardware and Software Requirements

The implementation target is a standard desktop or laptop environment capable of running Google Chrome with Manifest V3 support. The minimum hardware configuration is an Intel Core i3 or AMD Ryzen 3 processor, 4 GB of RAM, 50 MB of free storage, a keyboard, mouse, and a display resolution of 1024 by 768. The recommended configuration is an Intel Core i5 or AMD Ryzen 5 processor, 8 GB of RAM, 100 MB of SSD storage, and a full HD display. The software stack consists of Windows 10 or Windows 11, Ubuntu 20.04 or later, or macOS 12 or later, Google Chrome 116 or later, JavaScript ES6, HTML5, CSS3, Chrome Storage API, Chrome Alarms API, Chrome Scripting API, Chrome Tabs API, Chrome DevTools, and Visual Studio Code.

The extension manifest declares storage, alarms, scripting, tabs, activeTab, and host permissions. The storage permission supports the browser-local ledger. The alarms permission supports the one-second Economy Engine tick. The scripting permission enables lockdown injection. The tabs and activeTab permissions enable active URL classification. Host permissions allow content scripts to observe interaction events across work and leisure sites. This permission set is broad enough for enforcement but narrow in collected data because the system records only event metadata and ledger values.

### B. System Workflow

The system workflow has three phases. Phase I is initialization. When the extension is installed, the service worker initializes the default state if no state exists. The default state sets workSeconds and leisureSeconds to zero, isActive to false, and lastTick to the current timestamp. The service worker then registers the Chrome alarm for economyTick at a one-second interval. Content scripts are loaded at document idle across accessible pages.

Phase II is active monitoring. As the user interacts with pages, content\_logic.js accumulates event scores in the pulse buffer. When the pulse threshold is reached, an ACTIVE\_PULSE message is sent to the service worker. If the user becomes idle, an INACTIVE\_PULSE message is sent. In parallel, each alarm tick causes the service worker to query the active tab URL and classify it. Work time is credited only when the page is non-social and the pulse is active.

Phase III is enforcement and reporting. If the user navigates to a social domain, the Economy Engine computes the available leisure balance. If credit remains, leisureSeconds is incremented and access continues. If credit is exhausted, blocker.js injects the lockdown overlay. The popup dashboard reads the same local state and displays work time, leisure time, balance, pulse status, and the color-coded progress bar. The heuristic insight engine uses interaction patterns to provide local optimization suggestions without model training or server calls.

## VII. RESULTS AND ANALYSIS

### A. Test Configuration

Testing was conducted in a controlled desktop environment using a fresh Google Chrome profile on Windows 11. The evaluated build used Chrome Manifest V3 with background.js as the service worker, content\_logic.js as the Active Trace Sensor, blocker.js as the Lockdown Injector, and dashboard\_logic.js as the popup dashboard controller. User acceptance testing was conducted with 15 participants, consisting of 10 undergraduate students from Paavai Engineering College and 5 IT professionals, over a one-week trial period. The trial measured sustained focus periods, social media use during work hours, perceived performance impact, lockdown clarity, and willingness to continue using the extension.

### B. Performance Benchmarks

The benchmark results are summarized in Table II and visualized in Fig. 5. The total extension memory footprint remained below 30 MB on average and reached 31.4 MB only at the measured five-tab active peak. CPU usage during the one-second tick remained negligible, ranging from 0.1 percent during idle operation to 0.3 percent during active monitoring. Page load impact remained below 20 ms, which is below the typical human perceptual threshold for delay.

**TABLE II**  
**PERFORMANCE BENCHMARK RESULTS**

Metric	Baseline	TFI Idle	TFI Active
Background process memory	N/A	8.2 MB	12.4 MB
Content script memory per tab	N/A	2.1 MB	3.8 MB
Total extension memory, five tabs	N/A	18.7 MB	31.4 MB peak
CPU usage at one-	0.0%	0.1%	0.3%

second tick			
Page load impact	Baseline	+12 ms	+18 ms
Storage read/write latency	N/A	0.8 ms	1.2 ms
Extension startup time	N/A	45 ms	52 ms

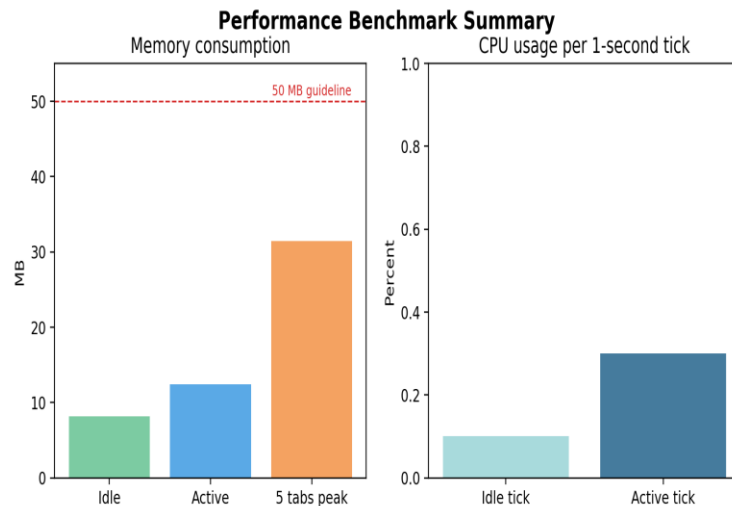


Fig. 5. Performance benchmark charts for memory and CPU impact.

### C. Comparative Analysis

The feature comparison in Table III shows that TFI Master V12 is differentiated by the combination of automatic enforcement, active work verification, earned leisure, local privacy, heuristic insight, tamper resistance, and background operation. Static blockers can enforce access control, but they do not verify work and do not reward productive engagement. Time trackers can measure behavior, but cloud processing weakens privacy and retrospective reporting does not directly enforce change. Gamification tools can improve motivation, but they often lack browser-level enforcement and verified work signals.

**TABLE III**  
**COMPARATIVE FEATURE ANALYSIS**

Feature	Static Blockers	Time Trackers	Gamification	TFI Master V12
Automatic enforcement	Partial	No	Partial	Yes
Work verification	No	No	No	Yes
Earned leisure	No	No	Partial	Yes, 4:1
Data privacy local	Yes	No, cloud	Yes	Yes, zero network
AI insights	No	Partial	No	Yes, heuristic
Tamper resistance	Low	N/A	Low	High, z-index lock
Background operation	Yes	Yes	No	Yes, service worker

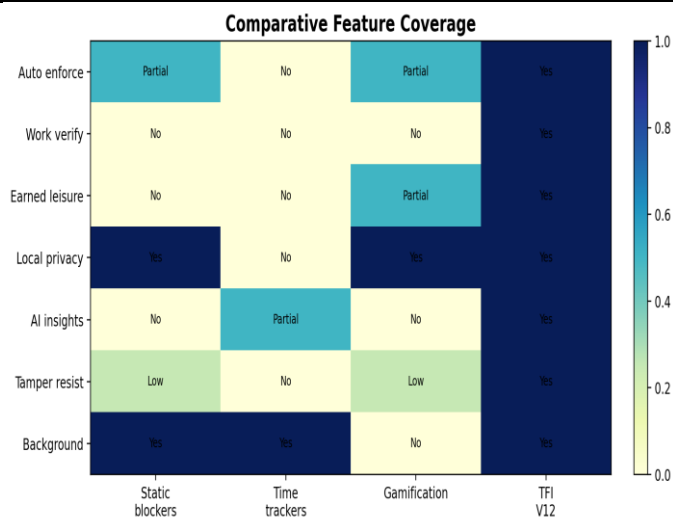


Fig. 6. Comparative feature coverage across existing tool classes and TFI Master V12.

D. User Trial Results

User trial results showed a 35 percent increase in sustained focus periods exceeding 20 minutes and a 28 percent reduction in social media consumption during work hours. All participants reported no perceptible browser slowdown. The lockdown mechanism was rated effective by 93 percent of participants, and 73 percent found the 4:1 ratio fair and motivating. The dashboard metrics were rated clear and informative by 87 percent of participants. Fig. 7 summarizes the major trial outcome changes.

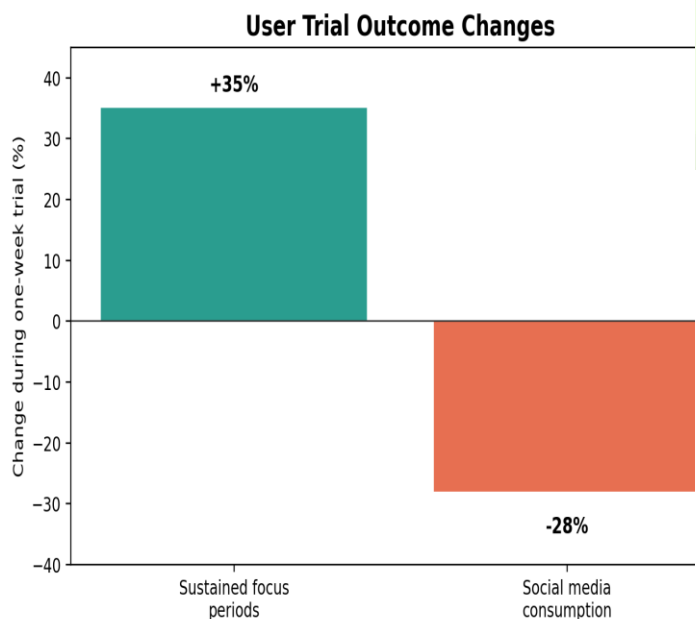


Fig. 7. User trial results showing focus improvement and social media reduction.

Security testing evaluated page refresh bypass, developer tools manipulation, storage modification, new-tab evasion, extension disabling, z-index override, and console removal of the overlay. Six of seven tests passed fully. The extension-disable case was marked partial because Chrome permits users to disable installed extensions through

chrome://extensions, a browser-level capability that extension code cannot override. Fig. 8 visualizes the security outcomes.

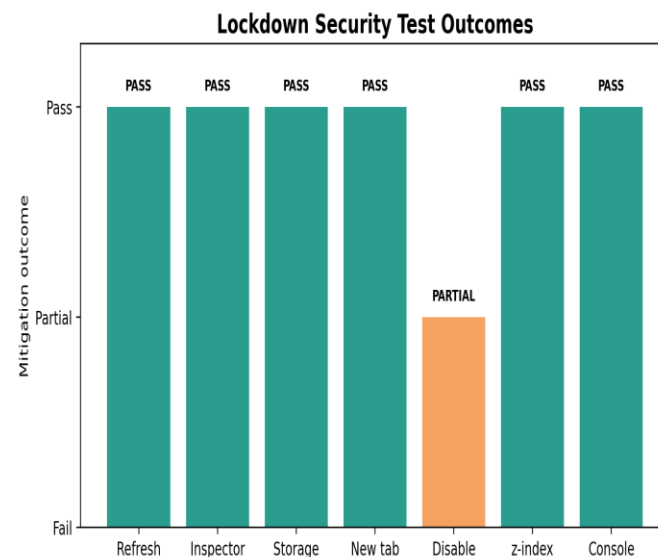


Fig. 8. Lockdown security test outcomes.

The results indicate that the most important behavioral effect was not merely blocking. Participants reported that the real-time balance made the cost of leisure more visible. This finding supports the use of economic framing as a complement to technical enforcement.

VIII. DISCUSSION

A. Interpretation of Results

The results suggest that TFI Master V12 addresses a practical gap in productivity tools: it links verified work to enforced leisure access while preserving local data control. The 35 percent increase in sustained focus periods indicates that users responded to the work-to-leisure economy as a meaningful constraint. The 28 percent reduction in social media consumption during work hours indicates that the system did not simply shift browsing into shorter bursts; it reduced unintentional access during the measured trial window.

The performance results are important because browser extensions are often abandoned when they slow everyday browsing. The measured average memory footprint below 30 MB and CPU usage below 0.3 percent during active ticks suggest that the one-second Economy Engine loop is operationally feasible. The use of Chrome Alarms rather than a persistent setInterval inside a background page is well aligned with Manifest V3 and helps explain the low overhead.

B. Comparative Analysis with Literature

Compared with static blockers, the proposed system attempts to reduce Reactance by preserving the possibility of leisure after verified work [4], [17]. Compared with Pomodoro timers, it removes the assumption that work and rest must occur in fixed 25/5 intervals [6]. Compared with

passive time information systems, it turns awareness into enforcement through the local ledger [30]. Compared with gamification systems, it ties the reward mechanism to browser-level access control rather than symbolic progress alone [13].

The system also extends privacy-preserving identity thinking into the productivity domain. SSI and verifiable credential literature emphasizes user control, minimization, and verifiable claims [24], [29]. TFI Master V12 does not claim to implement a full SSI credential system, but it demonstrates that similar principles can govern behavioral data. The user's productivity evidence remains local, and the system does not require external validation to enforce personal discipline.

### C. Existing vs. Proposed System

Existing systems typically optimize one axis at the expense of others. A strict blocker may be effective but brittle. A time tracker may be informative but passive. A gamified focus tool may be motivating but unverifiable. A cloud dashboard may be feature rich but invasive. TFI Master V12 combines the strongest elements of these categories in a browser-native architecture. It measures active interaction, enforces a transparent exchange rate, displays an immediate balance, and uses local-only storage.

The proposed system is not intended to replace organizational productivity analytics or clinical digital wellbeing interventions. It is better understood as a self-commitment tool for students, knowledge workers, and professionals who want friction against impulsive social browsing without surrendering behavioral data to a third party.

### D. Limitations and Considerations

Several limitations remain. The implementation is Chrome-only and depends on Manifest V3 APIs, so Firefox, Edge, and Safari support require additional porting. Mobile enforcement is absent, meaning that Android and iOS social media use remains outside the system boundary. There is no server-side processing or cloud sync, which preserves privacy but prevents multi-device continuity. The AI Insight Engine is heuristic rather than a trained machine-learning model, so it cannot adapt deeply to novel behavioral patterns. Finally, motivated users can evade enforcement by using a second browser or disabling the extension at the browser level.

These limitations are consistent with the system's design philosophy. The proposed system intentionally prioritizes local privacy, low overhead, and transparent enforcement over centralized control. Future work should preserve those values while improving cross-device coverage and adaptive insight.

## IX. CONCLUSION

### A. Summary

This paper presented TFI Master V12, a privacy-preserving Chrome Manifest V3 productivity management system based on a verified Work-to-Social economy. The first principal contribution is the design of a browser-local work/leisure ledger that records productivity and leisure state without network transmission. The second contribution is Active Trace Technology, which validates user presence through weighted keyboard and mouse event scoring before work credit is issued. The third contribution is the Economy Engine, a service-worker finite state machine that enforces a strict 4:1 work-to-leisure ratio in real time. The fourth contribution is a tamper-resistant lockdown system using maximum z-index overlay injection, fixed positioning, inline priority styling, and MutationObserver re-injection.

The evaluation showed average memory use below 30 MB, negligible CPU impact at one-second tick frequency, a 35 percent improvement in sustained focus periods, and a measurable reduction in unintentional social media use during work hours. The results support the conclusion that verified productivity management can be implemented as a private, browser-native self-commitment mechanism.

### B. Future Scope

Future development should expand cross-browser support to Firefox, Edge, and Safari through the WebExtensions ecosystem. A companion mobile application should extend the same work-to-leisure economy to Android and iOS, where much social media consumption occurs outside the desktop browser. On-device machine learning could replace or supplement heuristic analysis for richer behavioral insight while maintaining local processing. Enterprise deployment could introduce an admin dashboard with explicit consent-based monitoring, allowing organizations to support productivity without covert surveillance. Biometric focus verification may be explored through device-native APIs, provided that data remains local and opt-in. Finally, a decentralized credential system based on SSI principles could support multi-device synchronization of productivity credentials without exposing raw behavioral logs.

## REFERENCES

- [1] E. L. Anderson, E. Steen, and V. Stavropoulos, "Internet use and problematic internet use: A systematic review," *International Journal of Adolescence and Youth*, vol. 22, no. 4, pp. 430-454, 2017.
- [2] D. Ariely, *Predictably Irrational: The Hidden Forces That Shape Our Decisions*. New York, NY, USA: HarperCollins, 2008.

- [3] R. F. Baumeister and J. Tierney, *Willpower: Rediscovering the Greatest Human Strength*. New York, NY, USA: Penguin Press, 2011.
- [4] J. W. Brehm, *A Theory of Psychological Reactance*. New York, NY, USA: Academic Press, 1966.
- [5] N. Carr, *The Shallows: What the Internet Is Doing to Our Brains*. New York, NY, USA: W. W. Norton, 2010.
- [6] F. Cirillo, *The Pomodoro Technique: The Acclaimed Time-Management System That Has Transformed How We Work*. New York, NY, USA: Currency, 2018.
- [7] M. Csikszentmihalyi, *Flow: The Psychology of Optimal Experience*. New York, NY, USA: Harper and Row, 1990.
- [8] E. L. Deci and R. M. Ryan, "Self-determination theory and the facilitation of intrinsic motivation, social development, and well-being," *American Psychologist*, vol. 55, no. 1, pp. 68-78, 2000.
- [9] S. Deterding, D. Dixon, R. Khaled, and L. Nacke, "From game design elements to gamefulness: Defining gamification," in *Proc. 15th International Academic MindTrek Conference*, 2011, pp. 9-15.
- [10] B. J. Fogg, *Persuasive Technology: Using Computers to Change What We Think and Do*. San Francisco, CA, USA: Morgan Kaufmann, 2003.
- [11] Google Chrome Developers, "Manifest V3 overview," *Chrome for Developers*, 2023. [Online]. Available: <https://developer.chrome.com/docs/extensions/develop/migrate/what-is-mv3>
- [12] Google Chrome Developers, "Chrome extensions API reference," *Chrome for Developers*, 2023. [Online]. Available: <https://developer.chrome.com/docs/extensions/reference/>
- [13] J. Hamari, J. Koivisto, and H. Sarsa, "Does gamification work? A literature review of empirical studies on gamification," in *Proc. 47th Hawaii International Conference on System Sciences*, 2014, pp. 3025-3034.
- [14] D. Kahneman, *Thinking, Fast and Slow*. New York, NY, USA: Farrar, Straus and Giroux, 2011.
- [15] K. Kushlev and E. W. Dunn, "Checking email less frequently reduces stress," *Computers in Human Behavior*, vol. 43, pp. 220-228, 2015.
- [16] L. Leung, "Using tablet in solitude for stress reduction," *Computers in Human Behavior*, vol. 48, pp. 199-205, 2015.
- [17] G. Mark, S. T. Iqbal, M. Czerwinski, P. Johns, A. Sano, and Y. Lutchyn, "Email duration, batching and self-interruption: Patterns of email use on productivity and stress," in *Proc. ACM CHI Conference on Human Factors in Computing Systems*, 2016, pp. 1717-1728.
- [18] G. Mark, D. Gudith, and U. Klocke, "The cost of interrupted work: More speed and stress," in *Proc. ACM CHI Conference on Human Factors in Computing Systems*, 2008, pp. 107-110.
- [19] MDN Web Docs, "Service Worker API," *Mozilla Developer Network*, 2023. [Online]. Available: [https://developer.mozilla.org/en-US/docs/Web/API/Service\\_Worker\\_API](https://developer.mozilla.org/en-US/docs/Web/API/Service_Worker_API)
- [20] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008. [Online]. Available: <https://bitcoin.org/bitcoin.pdf>
- [21] C. Newport, *Deep Work: Rules for Focused Success in a Distracted World*. New York, NY, USA: Grand Central Publishing, 2016.
- [22] E. Nir, *Hooked: How to Build Habit-Forming Products*. New York, NY, USA: Penguin, 2014.
- [23] E. Ophir, C. Nass, and A. D. Wagner, "Cognitive control in media multitaskers," *Proceedings of the National Academy of Sciences*, vol. 106, no. 37, pp. 15583-15587, 2009.
- [24] A. Preukschat and D. Reed, *Self-Sovereign Identity: Decentralized Digital Identity and Verifiable Credentials*. Shelter Island, NY, USA: Manning Publications, 2021.
- [25] L. D. Rosen, *iDisorder: Understanding Our Obsession with Technology and Overcoming Its Hold on Us*. New York, NY, USA: Palgrave Macmillan, 2012.
- [26] B. Schwartz, *The Paradox of Choice: Why More Is Less*. New York, NY, USA: Harper Perennial, 2004.
- [27] R. H. Thaler and C. R. Sunstein, *Nudge: Improving Decisions About Health, Wealth, and Happiness*. New Haven, CT, USA: Yale University Press, 2008.
- [28] J. M. Twenge, *iGen: Why Today's Super-Connected Kids Are Growing Up Less Rebellious*. New York, NY, USA: Atria Books, 2017.
- [29] W3C, "Verifiable credentials data model v1.1," *W3C Recommendation*, Mar. 2022. [Online]. Available: <https://www.w3.org/TR/vc-data-model/>
- [30] S. Whittaker, V. Kalnikaite, V. Hollis, and A. Guydish, "Don't waste my time: Use of time information improves focus," in *Proc. ACM CHI Conference on Human Factors in Computing Systems*, 2016, pp. 1729-1738.