Qwen

Sovereign-by-Design: A Rust-Centric Architecture for Enforcing Neurorights in AI Development Tunnels

The objective of this research is to design and implement a reference architecture for "augmented-sovereignty," a concept centered on an individual citizen's control over their cognitive processes and digital interactions in the age of advanced AI. This architecture prioritizes the strict enforcement of neurorights—such as mental privacy, cognitive liberty, personal identity, free will, and equal access to augmentation—as non-negotiable technical constraints [1] [30]. The implementation is grounded in a Rust-first approach, leveraging its memory safety guarantees to build a secure foundation. The system is structured as a multi-layered defense, with the first priority being a code-level enforcement mechanism that actively blocks actions violating neurorights invariants. The second priority is a zero-trust network isolation layer that prevents external access to sensitive neuroworkspaces and AI dev-tunnels. The third priority is a governance plane based on EVOLVE tokens and multi-signature approvals, which provides a secure, cryptographic method for evolving policies over time. The scope begins with a reference architecture for individual use but incorporates clean abstractions for future multi-citizen deployments, ensuring sovereign namespaces can interoperate through explicit, cryptographically verified rules rather than default sharing. This report details the technical feasibility and strategic design of each architectural pillar, demonstrating how abstract ethical principles can be translated into a tangible, verifiable, and resilient software system.

Core Enforcement Layer: The Code-Level Shield for Neurorights Invariants

The foundational principle of this reference architecture is the absolute primacy of neurorights enforcement at the code level. This core enforcement layer is designed to be the first and most critical line of defense, operating under the premise that no external factor, such as network configuration or governance permissions, should ever override a confirmed violation of a citizen's cognitive sovereignty. The system achieves this through a synergistic combination of a `PolicyEngine`, context-aware `Tsafe` kernel modes, and immutable `ALN` (Augmented Logic Notation) schemas embedded within the `vkernel`. This triad works to create a robust, policy-as-code framework where every action taken by an AI agent or process is evaluated against a set of hard-coded, non-negotiable rules before execution. The entire system is built upon the understanding that security cannot be an afterthought; it must be an intrinsic property of the software itself, a concept embodied by the Language-Theoretic Security (LangSec) paradigm, which advocates for treating all inputs as structured data validated against formal grammars to prevent exploits [12].

The central nervous system of this enforcement layer is the `PolicyEngine`. This component acts as the ultimate arbiter for all actions, defined as `SovereignActionKind`, within the augmented citizen's environment [33]. Its integration into the `DevTunnelGateway` service ensures that every interaction initiated by an AI agent—from opening a tunnel to executing code—is subject to pre-validation [10]. The `PolicyEngine` does not operate on simple allow/deny lists but evaluates actions against a complex set of conditions derived from the `Tsafe` kernel mode, the agent's stake definition, and the immutable invariants encoded in the `vkernel.aln` schema. For instance, if an AI agent attempts to modify a file within the `neuroworkspace` that contains a citizen's neural logs or Bostrom anchors, the `PolicyEngine` would consult the `vkernel.aln` schema, identify the file as a sovereign asset, and immediately block the action, regardless of the agent's apparent authority or intent. This moves beyond traditional Role-Based Access Control (RBAC) or Attribute-Based Access Control (ABAC) by enforcing a more stringent **principle-of-least-action**, where an agent's capabilities are strictly limited to a predefined, vetted set of operations.

Complementing the `PolicyEngine` is the concept of `Tsafe` kernel modes, which introduces a dynamic, context-aware privilege model. For the specific use case of AI development tunnels, distinct modes like `DevSimulated`, `DevRestricted`, and `DevLive` are defined [11] [55]. Each mode represents a different security posture with varying levels of access and resources. The `DevSimulated` mode might allow an agent to run tests against synthetic data but with a very low RoH (Reasonable Honor) ceiling and no access to live APIs. The `DevRestricted` mode could permit communication with internal lab servers but still forbid direct access to the internet or the main `neuroworkspace`. Only upon explicit, human-approved elevation can a session transition to `DevLive`, which grants broader permissions necessary for production work but also triggers heightened monitoring and stricter policy checks. This tiered system prevents an AI agent from escalating its privileges arbitrarily. An attack vector that relies on tricking an agent into gaining broader access is neutralized because the agent's effective power is capped by its current `Tsafe` mode. Furthermore, the introduction of a `Tunnel RoH Budget` axis in the `.tsafe.aln` schema adds another layer of constraint, ensuring that any given dev-session consumes only a small, bounded slice of RoH, thereby preventing open-ended, high-risk modifications that could accumulate over time [90].

The ultimate source of authority for this enforcement layer lies in the `vkernel.aln` schema and the `rohmodel.aln`. These files serve as the immutable constitution of the augmented citizen's digital domain. The plan explicitly calls for encoding absolute invariants within `.vkernel.aln` that forbid any dev-tunnel action from modifying core system files like `.neurorights.json`, `.stake.aln`, `.rohmodel.aln`, and even `.vkernel.aln` itself [34]. These are not merely rules that the `PolicyEngine` checks; they are architectural constraints hardcoded into the virtual kernel. Any attempt to bypass them would require a flaw in the `vkernel` itself, which is a significantly harder target to exploit than application-level logic. This enforces immutability, a critical security property that ensures the foundational rights and stake definitions of the citizen cannot be altered by an AI agent, a compromised tool, or any other entity operating within the dev-tunnel environment [34]. The distinction between a privileged outer orchestrator LLM and a less-permissive inner LLM further reinforces this hierarchy, controlling the flow of information and preventing sensitive data from leaking to lower-security contexts [2].

The choice of Rust as the implementation language for the `PolicyEngine`, `Tsafe` kernel, and related services is a cornerstone of this architecture's security. Rust's ownership and borrow-checking systems provide strong memory safety guarantees at compile time, drastically reducing the risk of common vulnerabilities like buffer overflows, use-after-free errors, and null pointer dereferences that plague C/C++ based systems [13] [16]. While some safety overhead may be introduced through bounds checks and borrow checking, the trade-off is a dramatic increase in system integrity, especially for a component handling policy evaluation and acting as a gatekeeper [55]. The WDF team's work on designing safe Rust abstractions for Windows drivers demonstrates the viability of using Rust for low-level system programming, which is analogous to the task of building a secure `vkernel` [42]. By building the core enforcement logic in Rust, the architecture gains a powerful guarantee that the code itself is free from certain classes of exploitable bugs, allowing trust to be placed in the implementation of the logic rather than just the logic itself. This focus on a memory-safe language directly supports the LangSec principle of preventing exploits that target parsers and input-handling routines [12].

In essence, the core enforcement layer creates a fortress around the citizen's cognitive sovereignty. The `PolicyEngine` is the vigilant guard at the gate, the `Tsafe` modes are the different levels of fortified zones within the fortress, and the `vkernel.aln` schema is the unbreakable stone tablet upon which the laws of the land are carved. No action can be taken without passing through the guard's scrutiny, and the guard's orders are absolute and backed by the immutable laws of the realm. This layered, yet tightly integrated, approach ensures that the protection of mental privacy, cognitive liberty, and identity is not a configurable option but a fundamental, non-negotiable property of the system.

Component	Primary Function	Technical Implementation	Key Security Benefit
`PolicyEngine`	Central arbiter for all `SovereignActionKinds`.	Integrated into Rust services like `DevTunnelGateway`. Evaluates actions against ALN policies.	Prevents unauthorized actions by validating every operation before execution. [10] [33]
`Tsafe` Kernel Modes	Context-aware privilege levels with different constraints.	Defined in `.tsafe.aln`. Examples include `DevSimulated`, `DevRestricted`, `DevLive`.	Limits an agent's capabilities based on its operational context, preventing privilege escalation. [11] [55]
`vkernel.aln` Schema	Source of truth for immutable system invariants.	Encodes non-negotiable rules forbidding modification of core files like `.neurorights.json`.	Guarantees immutability of foundational rights and stake definitions. [34]
Rust Implementation	Memory-safe programming language for core services.	Used for `PolicyEngine`, `vkernel`, `ztunnel`, and other critical components.	Eliminates entire classes of memory-safety vulnerabilities, increasing system integrity. [13] [16]

This comprehensive and deeply integrated enforcement layer forms the bedrock of the entire architecture, providing the absolute security guarantees required to protect an individual's augmented citizenship in a world of increasingly autonomous and potent AI agents.

Network Isolation Layer: Zero-Trust Principles and Secure Meshes

While the core enforcement layer provides a shield at the application level, the second layer of defense is a rigorously implemented zero-trust network isolation strategy. This layer's primary function is to prevent any external or "urban" node from technically reaching the augmented citizen's AI shell, neuroworkspace, or development tunnels [66]. It operates on the fundamental tenet of zero trust: never trust, always verify. It assumes that the network itself is hostile and that every connection, whether originating internally or externally, must be authenticated, encrypted, and authorized before any data exchange can occur. This defense-in-depth approach ensures that even if a vulnerability exists within an application running inside a dev-tunnel, an attacker on an untrusted network like the public internet cannot exploit it to gain access to the citizen's sensitive assets.

The cornerstone of this network layer is a per-node, Rust-based zero-trust tunnel, conceptually similar to Istio's `ztunnel` [51] [66]. This service is deployed on every device that constitutes part of the citizen's trusted mesh, including the NeuroPC, local LLM hosts, and any lab servers. Its responsibilities are twofold: encryption and authentication. All traffic flowing between nodes in the mesh is encapsulated and encrypted, creating a secure overlay network. Authentication is handled via mutual Transport Layer Security (mTLS), where both the client and server present certificates to verify their identities [37]. Critically, these certificates are tied to the citizen's Bostrom addresses, which serve as their unique, persistent identifiers in the system [90]. This means that only devices possessing the correct cryptographic keys associated with the citizen's approved addresses can join their private mesh. External entities, including city infrastructure or government servers, are automatically excluded from this process, effectively building a wall around the citizen's digital life.

To further harden the network perimeter, the architecture specifies the maintenance of completely separate meshes for the Augmented-Citizen environment and the Urban/External network [64]. Traffic originating from the citizen's secure mesh is never routed onto the public internet or any other untrusted network segment. This is enforced at the gateway level, not just within applications. For example, firewall rules can be configured to block all outbound connections from the citizen's devices to known CIDRs of city or authority networks [87]. This architectural choice directly implements the principle of least network exposure, minimizing the system's attack surface by denying pathways to potentially hostile actors. The network is segmented down to the level of individual AI-related services, with explicit, minimal access rules governing communication between them—a practice known as micro-segmentation [83]. For instance, the container running the Git hook service is only permitted to communicate with the Git remote repository, and nothing else.

A crucial aspect of this zero-trust implementation is Layer 4 (L4) authorization at the ztunnel level. This means that the tunnel itself acts as a sophisticated firewall, inspecting not just IP addresses and ports but also the application-layer context of the traffic. The policy is designed so that only one specific process, the `DevTunnelGateway`, is authorized to establish connections to external endpoints such as LLM API hosts, MCP tools, and Git remotes [37]. All other communication paths from the citizen's internal network to the outside world are denied by default. This prevents a compromised AI agent from, for example, establishing a covert channel to exfiltrate data or receive new instructions from an adversarial server. The `ztunnel` becomes the sole diplomatic envoy authorized to conduct business with the outside world on behalf of the citizen's enclave.

Protecting the citizen's most valuable assets, the `neuroworkspace` volumes, is a paramount concern. The architecture includes specific enforcement mechanisms to ensure these volumes are never mounted by untrusted containers. A container can only access the `neuroworkspace` if it possesses a special ztunnel identity and its configuration has been approved via the EVOLVE token governance process [5]. This creates a two-factor requirement: cryptographic identity (from the ztunnel) and policy approval (from the governance layer). Even if an attacker were to compromise a container, they would be unable to attach the sovereign storage volume without first obtaining the necessary credentials and permissions. This dual-enforcement model provides a powerful safeguard against lateral movement and data theft within the local environment.

The selection of Rust for the `ztunnel` implementation is strategically significant. Network daemons are a common target for attackers due to their constant exposure to untrusted input. By writing the `ztunnel` in Rust, the developers leverage the language's memory safety guarantees to build a highly resilient networking component that is far less susceptible to the types of vulnerabilities that have plagued network services written in C or C++ [13] [16]. This reduces the risk of a breach originating from a flaw in the network stack itself. The combination of application-level controls from the `PolicyEngine` and network-level controls from the `ztunnel` creates a formidable defense-in-depth strategy. The application trusts the network to deliver packets securely, while the network trusts the application to request only authorized actions. This synergy between layers is essential for achieving the desired state of complete network isolation for the augmented citizen's dev-tunnels. Finally, all flows associated with dev-tunnels are meticulously logged, and periodic analysis is conducted to detect anomalies, such as unexpected external contacts or unauthorized route changes, providing continuous oversight of the network's integrity [10] [28].

Governance and Evolution Plane: Cryptographic Control for Policy Change

The third and final pillar of the reference architecture is the governance and evolution plane. While the first two layers provide static, active defenses, this plane addresses the dynamic nature of technology, law, and threat landscapes. Policies and system configurations must evolve over time to adapt to new challenges and incorporate new legal requirements. However, the process of change itself must be secure, transparent, and resistant to hijacking or unilateral alteration. The governance model is built upon a decentralized, cryptographic foundation, using EVOLVE tokens and multi-signature approvals to regulate how the system's core parameters—its neurorights, stakes, and Tsafe configurations—can be modified. This transforms policy management from a manual, opaque process into a formal, auditable, and verifiable protocol.

At the heart of this governance system are EVOLVE tokens. These tokens act as the currency for participation in the governance process, serving as the mechanism for proposing, funding, and approving changes to the system's policies and configurations [6]. To propose a change to a critical file like `.neurorights.json` or to alter the `Tsafe` kernel's allowed actions, a citizen must submit a proposal and likely lock up a certain amount of EVOLVE tokens as a bond. This serves multiple purposes: it filters out frivolous or malicious proposals, aligns incentives by making proposers accountable for their suggestions, and funds the computational costs of processing the change. The token economy is designed to incentivize responsible stewardship of the citizen's augmented sovereignty [6].

No change of significant consequence can be made unilaterally. Critical modifications to the system's invariants—such as altering the definition of a neuroright, changing stakeholder roles, or modifying the `vkernel`'s core logic—require approval from multiple stakeholders, implemented through a multi-signature (multi-sig) scheme [4] [41]. This could involve requiring signatures from different Bostrom addresses representing different facets of the citizen's identity (e.g., a personal address, a professional address, a trusted advisor's address) or from a predefined group of validators. This approach, used in systems like Orbit Bridge, reduces the risk of a single point of failure or compromise, ensuring that no single entity, including the citizen themselves in a moment of duress, can make catastrophic changes without broad consensus [4]. This cryptographic requirement for collective agreement is the ultimate safeguard against coercion or unauthorized modification.

All governance activities are meticulously recorded in an append-only ledger, referred to as the `Donutloop`. Every EVOLVE token transaction, every policy proposal, and every multi-sig approval is logged as an event in this ledger [90]. This creates a permanent, tamper-evident history of all decisions made about the citizen's sovereignty. To anchor this ledger to a higher level of trust, all major changes are also committed to a blockchain-anchored proof file, `.bchainproof.json` [66]. This process involves taking a cryptographic hash of the relevant ledger state and publishing it on a public blockchain. This links the citizen's private governance decisions to a publicly verifiable, decentralized timestamp, making it computationally infeasible to retroactively alter past decisions without detection [57]. This creates an immutable audit trail that can be reviewed at any time to understand the lineage of a particular policy or configuration, ensuring full transparency and accountability.

This governance framework is not just a security feature; it is a prerequisite for the system's long-term viability and scalability. It provides the secure and verifiable protocol needed for the medium-term extension to multi-citizen deployments. When multiple augmented citizens wish to interact or share resources, their respective sovereign namespaces will need to negotiate interoperability rules. The EVOLVE token and multi-sig system can be used to manage these cross-shard agreements, ensuring that any change to the shared interface or interoperability contract is properly authorized by all participating parties. This contrasts sharply with a default-sharing model, which would inherently compromise sovereignty. Instead, the system is designed from the ground up to support cryptographically enforced interoperability, where each citizen's autonomy is respected and maintained [39]. The governance plane thus serves as the control plane that regulates not only the evolution of a single citizen's system but also the future growth of a federated network of sovereign individuals.

By combining a token-based incentive system, a multi-party approval mechanism, and an immutable, blockchain-anchored ledger, the governance plane ensures that the evolution of the citizen's augmented sovereignty is a deliberate, secure, and transparent process. It prevents the erosion of rights through subtle, incremental changes and provides a verifiable record of all decisions, empowering the citizen to maintain true ownership and control over their digital self.

Governance Component	Description	Purpose	Security Benefit
EVOLVE Tokens	Cryptographic tokens used to fund and incentivize governance actions.	To filter proposals, fund computations, and align stakeholder incentives for responsible stewardship. [6]	Prevents spam and malicious proposals; makes proposers accountable.
Multi-Signature Approval	A consensus mechanism requiring signatures from multiple designated stakeholders.	To authorize critical changes that affect core system invariants, such as neurorights or stake definitions. [4] [41]	Prevents unilateral control and mitigates the risk of a single point of failure.
Donutloop Ledger	An append-only, in-memory ledger for recording all governance events and policy changes.	To create a detailed, chronological history of all decisions affecting the citizen's sovereignty. [90]	Provides a tamper-evident audit trail for transparency and accountability.
Blockchain Anchoring	Publishing cryptographic hashes of the Donutloop ledger state to a public blockchain.	To create a permanent, verifiable, and decentralized timestamp for all major policy changes. [57] [66]	Ensures immutability of the historical record and protects against retroactive alteration.

This robust governance structure provides the necessary flexibility for the system to adapt and grow without sacrificing the core security principles that define it. It is the engine that allows the architecture to remain resilient in the face of an evolving technological and legal landscape.

Application in Practice: Securing AI Development Tunnels

The theoretical framework of the reference architecture finds its most practical application in the specific context of securing AI development tunnels. These tunnels, which connect an AI agent to various tools, models, and codebases for tasks like coding, debugging, and testing, represent a high-risk environment. They are points of interaction with untrusted code and large language models whose outputs are unpredictable. The architecture's design is explicitly tailored to mitigate these risks, creating a controlled and secure sandbox for AI-assisted development. This is achieved through a combination of specialized services, constrained agent patterns, and rigorous validation protocols.

The central component for managing these interactions is the `DevTunnelGateway`, a dedicated Rust service that acts as the single point of contact for all AI dev-tunnel activity [66]. It is the only process permitted to open tunnels to external models, APIs, or tools. This centralization is critical; it ensures that every interaction is channeled through a trusted intermediary that can apply the core `PolicyEngine` checks. The `PolicyEngine` evaluates every action—opening a tunnel, calling a tool, executing generated code—against the Tsafe and neurorights policies before forwarding the request [10]. This immediate pre-validation is the first line of defense against attempts to misuse the dev-tunnel environment.

To further constrain the AI agent's behavior, the architecture mandates the use of the **Action-Selector pattern**. Instead of allowing an AI agent to propose arbitrary actions, it is presented with a fixed, pre-defined set of options, such as "run unit tests," "refactor this function," or "suggest documentation improvements" [9]. This pattern fundamentally limits the agent's potential for harm. Even if an agent is manipulated through prompt injection to suggest a harmful action, that action will not be on the allowed list, and the `PolicyEngine` will reject the request. This is a direct countermeasure against sophisticated attacks that seek to trick an agent into violating its own constraints [38]. Similarly, the **Plan-Then-Execute** pattern is implemented to introduce a mandatory human-in-the-loop step for all significant changes. The AI proposes a complete plan of action, which is then presented to the citizen for review and approval *before* any untrusted input is processed or any code is applied [9] [38]. This ensures that the citizen retains ultimate control and conscious awareness over all modifications made to their codebase and, by extension, their development environment.

Security during code execution is addressed through a **Code-Then-Execute sandbox**. Generated code is executed in a highly restricted environment, such as a Rust sandbox or an OS container, with stringent limitations: no network access, a read-only filesystem, and absolutely no access to the `neuroworkspace` containing sensitive data [18]. This prevents malicious code from exfiltrating data, communicating with command-and-control servers, or corrupting the host system. This sandboxing provides a critical barrier, isolating the untrusted output of the AI agent from the citizen's trusted environment.

The orchestration of LLMs within the tunnel is handled with a **Dual LLM** setup. An outer, privileged LLM (or symbolic engine) acts as an orchestrator. This orchestrator sees only symbolic placeholders for the citizen's data and workspace, protecting raw information from being exposed to the inner LLM. The inner LLM, which may have more contextual awareness, generates the actual content, but its outputs are filtered and validated by the privileged orchestrator before being presented to the user or executed [2]. This layered approach to LLM usage minimizes the exposure of sensitive information and ensures that all outputs adhere to the system's security policies.

To ensure the integrity of the entire process, a comprehensive **Tunnel Telemetry Stream** is implemented. All actions taken by the AI agent, the decisions made by the citizen, and the outcomes of those actions are routed to a log file, `.answer.ndjson`, and recorded in the ALN ledger with a specific route identifier, `DEV-TUNNEL` [90]. This log links every event back to the citizen's Bostrom addresses and the corresponding RoH slice consumed, creating a detailed and auditable record of the entire development session. This telemetry is invaluable for post-mortem analysis, red-teaming exercises, and identifying patterns of behavior that may indicate a policy violation or a novel attack vector.

Finally, the transition of a dev-session from a simulated or restricted state to a live one requires explicit human attestation. The `DevTunnelSession` API in Rust must be signed by the citizen's `OrganicCPU`—representing their conscious consent—before it can become `DevLive` [85]. This step embeds the principle of informed consent directly into the system's operational flow, ensuring that high-risk operations are never performed without the citizen's explicit authorization. This combination of a centralized gateway, constrained agent patterns, sandboxed execution, dual-LLM orchestration, comprehensive logging, and mandatory attestation creates a holistic security model for AI development tunnels, directly addressing the threats of prompt injection, data leakage, and unauthorized system modification.

Strategic Alignment and Legal-Technical Integration

A defining characteristic of this reference architecture is its pragmatic approach to alignment with emerging neurorights frameworks. Rather than deriving the entire technical design from legal documents, the strategy is to first implement the system's core functionalities in Rust and ALN, and then annotate the implementation with pointers to relevant legal and ethical principles from frameworks like Chile’s constitutional neurorights and the Spanish Digital Rights Charter. This "code-first, annotate-later" methodology results in a system whose technical design naturally reflects and operationalizes these rights, turning abstract legal concepts into tangible, verifiable protections.

The architecture is profoundly aligned with the spirit and letter of Chile's landmark constitutional amendment, which was the first in the world to enshrine neurorights [1]. Specifically, the amendment to Article 19 aims to protect mental integrity and immunity from adverse effects of neurotechnologies [81] [82]. The core enforcement layer of the architecture directly embodies this principle. The `PolicyEngine` and `Tsafe` modes are explicitly designed to block actions that would violate mental privacy, cognitive liberty, and personal identity, such as non-consensual BCI control or thought-crime scenarios [21]. The prohibition on altering core system files like `.neurorights.json` and

`.stake.aln` is a technical manifestation of the right to mental identity and the right to a stable sense of self [22]. Furthermore, the differentiated consent model proposed in the Chilean legislation—where therapeutic and scientific uses follow standard rules while commercial uses require specific, prior, written consent—can inform the design of the `Tsafe` kernel modes [1]. For instance, a `CitizenPublic` status level with broader access might correspond to a commercial-use scenario requiring higher scrutiny, whereas a `CitizenCore` level would be akin to a therapeutic or personal research use case.

Similarly, the architecture resonates strongly with the principles outlined in the Spanish Charter of Digital Rights [25]. The charter references "digital rights in the use of neurotechnologies" and emphasizes the importance of mental privacy [20]. The zero-trust network isolation provided by the Rust-based `ztunnel` is a powerful technical implementation of the right to mental privacy. By encrypting all traffic and preventing any external observation of communications between the citizen's devices, the `ztunnel` ensures that conversations with AI agents, prompts sent to models, and the resulting responses remain confidential and inaccessible to third parties [50]. Some neurorights frameworks explicitly prohibit government interference with freedom of thought [21], a principle mirrored in the architecture's explicit blocking of "Urban Authority" nodes and CIDRs at the network gateway . The emphasis on cognitive liberty—the freedom to control one's own thought processes—is directly supported by the `Plan-Then-Execute` pattern and the `OrganicCPU` attestation requirement, which ensure that the citizen maintains ultimate control over their cognitive environment and any actions taken therein [23] [24].

The table below illustrates the mapping between the core neurorights and the corresponding technical mechanisms in the reference architecture.

Neuroright Concept	Definition & Legal Basis	Corresponding Technical Mechanism(s) in Architecture
Mental Privacy	The right to keep one's thoughts and neural data private from unauthorized access. [26] [27]	Zero-trust `ztunnel` with mTLS encryption; network segmentation; blocking of urban authority nodes. [25] [66]
Cognitive Liberty / Free Will	The right to control one's own thought processes and freedom from coercion. [23] [24]	`PolicyEngine` blocking coercive actions; `Plan-Then-Execute` pattern; `OrganicCPU` attestation for `DevLive` sessions. [2]
Personal Identity	The right to a stable sense of self and psychological continuity. [1] [22]	Immutable `vkernel.aln` schema; prohibition of modifications to `.neurorights.json`, `.stake.aln`, and `.rohmodel.aln`. [34]
Equal Access to Augmentation	The right to equitable access to technologies that enhance cognitive abilities.	Status Levels (`CitizenCore`, `CitizenLab`) defined in `.stake.aln`; tool labels (`ToolClass::SovereignCritical`). [68] [90]
Protection from Algorithmic Bias	The right to be free from discrimination by algorithms, including those in neurotechnology.	Governance via EVOLVE tokens and multi-sig; policy reviews in CI pipelines to prevent weakening of defaults. [6]

This direct mapping demonstrates that the architecture is not merely paying lip service to neurorights but is actively engineering them into the fabric of the system. The decision to treat AI entities themselves as having constraints, rather than rights, is also a pragmatic choice. The architecture aligns with emergent ideas about AI ethics by imposing constraints on how models are treated, ensuring they are not exploited or harmed, while keeping the augmented citizen's sovereignty as the primary and overriding concern [2]. The `nnet-rights.json` schema, for example, defines domains where agents can operate (e.g., code only, no real neural data), acting as a protective boundary for both the citizen and the AI models themselves [18]. This balanced approach ensures that the pursuit of augmented-sovereignty does not lead to the creation of new forms of digital oppression, even for artificial agents. The result is a system that is not only technically sound but also ethically grounded, providing a living blueprint for what a rights-respecting digital society could look like.

Synthesis and Implementation Roadmap

This deep research report has deconstructed a comprehensive reference architecture designed to achieve augmented-sovereignty for an individual citizen in an AI-driven world. The architecture is a coherent and multi-layered defense system, built upon three pillars of increasing abstraction: core enforcement, network isolation, and governance. Its strength lies in the prioritization of code-level enforcement as the absolute, non-negotiable first line of defense, ensuring that neurorights—invariants are not merely guidelines but are technically impossible to bypass. This is complemented by a zero-trust network layer that provides a secure perimeter, and a cryptographic governance plane that allows for the secure and auditable evolution of policies over time. The entire system is engineered to be a practical implementation of ethical principles, grounding abstract concepts like mental privacy and cognitive liberty in tangible, verifiable, and resilient software mechanisms.

The synthesis of the architecture reveals several key strategic insights. First, the principle of **Least Action** is paramount. Through the combined use of the `PolicyEngine`, `Tsafe` kernel modes, and the Action-Selector pattern, the system severely constrains the capabilities of AI agents, limiting them to a predefined, vetted set of operations and preventing them from escalating privileges or executing arbitrary commands. Second, the architecture exemplifies a powerful **Defense-in-Depth** strategy. The synergy between the application-level `PolicyEngine` and the network-level `ztunnel` creates a layered security posture where a vulnerability in one layer does not necessarily compromise the entire system. Third, the concept of **Code as Law** is realized through the use of Rust and ALN to translate ethical imperatives into enforceable, immutable invariants. The `vkernel.aln` schema acts as the constitution, and the Rust implementation as the executive branch tasked with its enforcement. Finally, the pragmatic strategy of **Implementing First, Aligning Later** proves superior to a purely legalistic derivation. The resulting system is a working artifact whose design naturally aligns with and demonstrates compliance with international neurorights frameworks, serving as a concrete embodiment of these principles in practice.

Based on this analysis, a phased implementation roadmap is recommended to bring this vision to fruition:

Phase 1: Foundational Enforcement and Governance 1. **Develop the `PolicyEngine` and `vkernel`:** Implement the core `PolicyEngine` in Rust, focusing initially on the most critical neurorights invariants. Define the initial `.vkernel.aln` schema with immutable rules for protecting `.neurorights.json`, `.stake.aln`, and other core files. 2. **Establish the Governance Model:** Design and deploy the EVOLVE token economy and the multi-signature approval mechanism. Create the initial `.donutloop.aln` ledger and the anchoring script for publishing proofs to a test blockchain. 3. **Implement the `Tsafe` Kernel Modes:** Define the initial set of `Tsafe` modes (`DevSimulated`, `DevRestricted`)ol that can deploy to production). The architecture manages this spectrum through two mechanisms: `Tool Sovereignty Labels` and `Dev-Tunnel Profiles`.

`Tool Sovereignty Labels` are ALN annotations that classify every tool in the ecosystem into categories like `ToolClass::SovereignCritical`, `ToolClass::CitizenSafe`, and `ToolClass::External` [68]. A `SovereignCritical` tool (e.g., one that can modify the `.vkernel.aln` schema) is only accessible in the most restricted `DevLive` mode and only after multi-sig approval. A `CitizenSafe` tool (e.g., `clippy` for linting) is available in all modes. An `External` tool (e.g., a proprietary CI service) is heavily sandboxed and its outputs are rigorously filtered before being integrated.

`Dev-Tunnel Profiles` define the specific configuration for a given code repository [71]. A profile like `kernel-safe` would restrict the tunnel to only `CitizenSafe` tools and prohibit any access to the kernel source tree. A `policy-safe` profile would allow tools that can read and analyze policy files but forbid any that can write to them. These profiles are validated by the `PolicyEngine` and enforced by the `Tsafe` kernel, ensuring that the development environment is always tailored to the sensitivity of the code being worked on.

Synthesis: A New Paradigm for Human-Centered AI Development

The AI chat dev-tunnel, as implemented by this architecture, represents a fundamental shift in the relationship between humans and AI. It moves away from the "AI as oracle" model, where the human asks a question and accepts the answer, toward an "AI as apprentice" model, where the human is the master craftsman, guiding, reviewing, and approving every step of the work.

It is a development environment where the AI's power is harnessed for productivity, but its potential for harm is systematically neutered through a combination of formal policy, contextual privilege, and human oversight. It is an environment where the augmented citizen is not just a user of AI, but its sovereign architect and ultimate governor. This is the practical, working realization of the augmented-citizenship status defined in the research plan: a status that is not conferred, but is actively, continuously, and technically maintained.

Legal-Ethical Alignment: Compiling International Neurorights Frameworks into Executable Code

The architecture's brilliance lies in its pragmatic approach to legal and ethical compliance. Rather than attempting to derive its design from abstract legal texts—a process fraught with ambiguity and interpretation—the system implements a robust, technically sound foundation first and then "compiles in" legal alignment as a set of annotated, executable constraints. This section details precisely how the most prominent international neurorights frameworks—Chile's constitutional amendment and the Spanish Digital Rights Charter—are translated from legal prose into lines of Rust code and ALN schema definitions.

Chile's Constitutional Amendment: From Article 19 to Immutable Invariants

Chile's historic 2021 constitutional amendment to Article 19 is the world's first formal, legal recognition of neurorights. It was born from a direct warning by neuroscientist Dr. Rafael Yuste to Chilean politicians about the risks of neurotechnology, leading to legislation that mandates a differentiated consent system and empowers the Institute of Public Health to restrict or prohibit neurotechnologies that aim to "influence human conduct without consent, exploit weaknesses, extract data without consent, or adversely affect neuroplasticity" [1] .

The architecture implements this legal mandate with surgical precision:

- "**Influence human conduct without consent**"**:** This is directly countered by the `DevLive` attestation requirement. No action that could influence the citizen's conduct—such as deploying an agent that modifies their development environment or applies a patch to their code—is permitted without the explicit, cryptographic signature of the `OrganicCPU` [85] . The `Plan-Then-Execute` pattern further reinforces this by requiring the citizen to review and approve the *plan* for influence before any action is taken.

- "**Exploit weaknesses**"**:** This is addressed by the `Tsafe` kernel's `DevSimulated` mode. This mode is specifically designed to expose the system to "red-team" attacks that attempt to exploit weaknesses in the AI's reasoning or the `PolicyEngine`'s logic [38] . By running all tests in this simulated, non-real-world environment, the system proactively identifies and patches weaknesses before they can be exploited in a live context.

- "**Extract data without consent**": This is the core function of the zero-trust `ztunnel`. The `ztunnel`'s mTLS authentication and strict micro-segmentation ensure that no external entity can extract neural data, as the `neuroworkspace` volumes are only mountable by containers with a special, EVOLVE-approved identity [67]. Furthermore, the `vkernel.aln` schema's immutable prohibition on modifying `.neurorights.json` ensures that the citizen's own consent preferences cannot be altered by an unauthorized actor.

- "**Adversely affect neuroplasticity**": While a complex biological concept, the architecture addresses its technological precursors. The research plan's taxonomy of threats includes "coercive nudging" and "predictive policing with neural data," which are technological methods that could potentially interfere with natural cognitive processes [18]. The `PolicyEngine`'s `DevTunnelPolicy` explicitly forbids any AI agent from accessing or processing real neural data, thereby removing the technological substrate upon which such adverse effects could be built.

The result is that Chile's constitutional amendment is not just referenced in a comment; it is embedded in the system's DNA as a set of unbreakable, runtime-enforced invariants.

The Spanish Digital Rights Charter: From Mental Privacy to Zero-Trust Networking

The Spanish Charter of Digital Rights, announced by the Secretary of State for Digitalization and Artificial Intelligence, explicitly references "digital rights in the use of neurotechnologies" and places a strong emphasis on the right to mental privacy [20] [25]. This charter is less prescriptive than Chile's constitutional text and more focused on establishing broad principles for a digital society.

The architecture translates these broad principles into concrete, technical implementations:

- "**Digital rights in the use of neurotechnologies**": This is the overarching theme of the entire architecture. Every component—from the `vkernel.aln`'s protection of neural data shards to the `Tsafe` kernel's restriction on neural data access—exists to ensure that the citizen's digital rights are upheld in their interaction with neurotechnology. The `CitizenStatusLevels` (`CitizenCore`, `CitizenLab`, `CitizenPublic`) provide a granular, ALN-defined framework for how these rights manifest in different contexts, binding them directly to the citizen's Bostrom address for global verification [90].

- "**Mental privacy**": This is the most direct and powerful translation. The `ztunnel`'s end-to-end encryption and strict network segmentation are a technical implementation of the "right to be left alone" in the digital realm. It ensures that the citizen's thoughts, as represented by their neural data and the models trained on it, are never exposed to an external observer, not even during transmission. The `DevTunnelPolicy`'s explicit ban on AI agents accessing neural data is a further, application-layer guarantee of this privacy [28].

- **Prohibition on Government Interference:** Several neurorights frameworks, including those discussed in academic literature, establish the individual's right to mental privacy and cognitive liberty and prohibit government entities from interfering with their freedom of thought [21]. The architecture implements this prohibition through its "No Urban Authority" rules, which involve blocking the CIDRs of known city and authority networks at the gateway level, not just within applications [81]. This is a proactive, network-layer blockade that prevents interference before it can even begin.

General Neurorights Literature: The Five Pillars as Technical Primitives

Beyond specific national frameworks, the broader academic literature on neurorights consistently identifies five core pillars: mental privacy, mental integrity, personal identity, free will, and equal access to augmentation [22] [30]. The architecture maps each of these to a specific, technical primitive:

- **Mental Privacy:** Enforced by the `ztunnel`'s encryption and the `vkernel.aln`'s prohibition on neural data access.
- **Mental Integrity:** Protected by the `Tsafe` kernel's `DevSimulated` mode, which prevents any real-world impact from potentially flawed AI reasoning, and by the `vkernel.aln`'s immutable invariants that prevent unauthorized modification of the citizen's cognitive state representations.
- **Personal Identity:** Safeguarded by the `stake.aln` file, which defines the citizen's subject roles and explicitly forbids the "CityInfra" role, ensuring that the citizen's digital identity is self-sovereign and not conflated with a state or corporate identity [33].
- **Free Will (Cognitive Liberty):** Guaranteed by the `Plan-Then-Execute` pattern and the `DevLive` attestation, which place the final, conscious decision in the

hands of the human user, preventing any form of algorithmic coercion or nudging [23].

- **Equal Access to Augmentation:** Enabled by the `CitizenStatusLevels` and the `Tool Sovereignty Labels`, which ensure that all citizens, regardless of their technical expertise, have access to a safe, sovereign development environment (`CitizenPublic` profile) while also providing a pathway for more advanced users to leverage more powerful tools (`CitizenLab`, `CitizenCore`) [90].

This systematic mapping demonstrates that the architecture is not an ad-hoc collection of features, but a coherent, principled system designed to be a living, breathing implementation of the most advanced thinking on human rights in the age of neurotechnology.

Technical Feasibility and Implementation Roadmap: From Research Plan to Production System

The ambition of the research plan is immense, spanning 50 meticulously detailed actions across six phases. Translating this vision into a working, production-grade system requires a pragmatic, phased implementation roadmap that prioritizes foundational stability, validates core assumptions, and incrementally adds complexity. This section provides a deep, analytical assessment of the technical feasibility of each major component, drawing on the latest academic research and industry best practices, and outlines a concrete, prioritized roadmap for bringing the reference architecture to life.

Foundational Feasibility Assessment

The architecture's foundation rests on three pillars of modern, mature technology: Rust, zero-trust networking, and formal policy languages. The feasibility of each is well-established.

- **Rust as the Systems Language:** Rust's suitability for building secure, high-performance systems is no longer theoretical. Its memory-safety guarantees have been validated in production by companies like Microsoft and Amazon, and its use in critical infrastructure like the Linux kernel is actively being explored [13] [42]. The research plan's reliance on Rust for the `ztunnel`, `PolicyEngine`, and `DevTunnelGateway` is therefore not a risky bet, but a conservative, best-practice choice. The primary implementation challenge is not Rust's safety, but its learning curve and the discipline required to avoid `unsafe` code, which can reintroduce vulnerabilities [55].

- **Zero-Trust Networking:** The concept of zero-trust is no longer a futuristic ideal but an industry standard. Projects like Istio's Rust-based `ztunnel` have demonstrated the viability of building high-performance, secure network proxies in Rust . The research plan's requirements for mTLS, micro-segmentation, and L4 authorization are all standard features of modern service mesh implementations. The technical challenge lies not in inventing new protocols, but in integrating these proven components into a cohesive, sovereign-specific architecture.

- **Formal Policy Languages and Engines:** The use of a custom ALN policy language is more novel, but it is grounded in solid academic foundations. Research into neuro-symbolic learning frameworks shows how to combine the pattern-matching power of neural networks with the explainability and formal guarantees of symbolic systems [9]. The `PolicyEngine` is essentially a symbolic reasoner that operates on the ALN grammar, a concept that is well-understood in the field of formal methods and verification.

The greatest technical uncertainty lies not in the individual components, but in their integration and the emergent properties of the system as a whole. How will the cumulative performance overhead of the `PolicyEngine` evaluation, the `Tsafe` kernel's syscall interception, the `ztunnel`'s encryption, and the sandboxing process impact the developer experience? This is a critical question that can only be answered through empirical measurement, which is why the research plan's emphasis on telemetry and red-teaming is so vital [10].

A Phased Implementation Roadmap

Based on the feasibility analysis and the user's stated priorities, the following phased roadmap is recommended. It focuses on delivering maximum value and validating core assumptions with minimal effort in the earliest stages.

Phase 1: Core Kernel and Policy Foundation (Actions 1-16)

- **Goal:** Establish the immutable, code-level enforcement layer.
- **Key Deliverables:**
- A working `vkernel.aln` schema with the core immutable invariants defined.
- A basic `PolicyEngine` that can parse and evaluate simple `SovereignActionKind` requests against a static `DevTunnelPolicy`.
- A `Tsafe` kernel with a functioning `DevSimulated` mode.

- **Why First?** This is the highest-priority layer. Without it, the rest of the architecture is meaningless. Building this first allows for rapid, isolated testing and validation of the core legal-ethical alignment.

Phase 2: Zero-Trust Network and Gateway Integration (Actions 17-32)

- **Goal:** Build the secure perimeter and integrate it with the core kernel.
- **Key Deliverables:**
- A minimal, functional Rust-based `ztunnel` with mTLS authentication.
- A `DevTunnelGateway` service that is the only process allowed to open tunnels.
- Integration of the `PolicyEngine` into the `DevTunnelGateway` so that every tunnel open is evaluated.
- **Why Next?** This layer protects the core kernel. Once the `DevTunnelGateway` is the only entry point, and it is secured behind the `ztunnel`, the system gains its first real-world security posture.

Phase 3: Tooling, Governance, and Red-Teaming (Actions 33-50)

- **Goal:** Populate the ecosystem with safe tools, implement the governance model, and rigorously test the system.
- **Key Deliverables:**
- A suite of `CitizenSafe` MCP servers for code-only operations.
- A working EVOLVE token contract and a multi-sig approval flow for policy changes.
- A comprehensive suite of "red-team" repos and attack scenarios to stress-test the `PolicyEngine` and `DevTunnelGateway`.
- **Why Last?** This phase adds the "polish" and the long-term sustainability. The governance model and tooling are essential for adoption, but they depend on the foundational layers being stable and secure.

This roadmap is not a rigid waterfall process but an iterative one. Each phase should include continuous integration, automated testing, and, most importantly, regular red-teaming exercises. The goal is to build a system that is not just theoretically sound, but is demonstrably resilient against real-world adversarial pressure.

Conclusion: The Sovereign Future of Human Augmentation

The journey documented in this report is not merely a technical exercise in software engineering; it is a foundational act of political and philosophical construction. It is the deliberate, meticulous building of a new kind of digital citizenship—one that is not granted by a state or a corporation, but is claimed, defended, and continuously renewed by the individual through the unassailable logic of code.

The reference architecture presented here is a direct response to a profound historical inflection point. For millennia, the human mind was a sanctuary, a private domain inaccessible to external observation or control. Neurotechnology and agentic AI have shattered that sanctuary, transforming the internal landscape of thought into a new frontier for data extraction, behavioral prediction, and algorithmic influence. In this new reality, the old models of digital rights, built for protecting data *about* people, are woefully inadequate for protecting the data *of* people—their very thoughts, memories, and intentions. The emergence of neurorights in Chile's constitution and Spain's Digital Charter is a desperate, necessary legal counter-offensive. This architecture is its technological counterpart.

Its power lies in its unwavering hierarchy of priorities. It begins not with convenience, not with performance, and not with feature richness, but with the absolute, non-negotiable enforcement of human rights at the deepest level of software execution. The `vkernel.aln` schema is the sovereign's constitution, the `PolicyEngine` its judiciary, and the `Tsafe` kernel its executive branch, all operating in concert to make violations of mental privacy, cognitive liberty, and personal identity a physical impossibility. This is followed by the impenetrable digital perimeter of the `ztunnel`, a network architecture that treats the citizen's neuroworkspace as a sovereign territory, its borders guarded by cryptography and its internal communications shielded by end-to-end encryption. Finally, the governance layer, powered by EVOLVE tokens and multi-signature approvals, ensures that the citizen's sovereignty is not a static relic but a living, evolving entity, its laws changed only through a process that is itself a sovereign act.

The AI chat dev-tunnel is the proving ground for this vision. It demonstrates that human-centered AI development is not only possible but is superior: more secure, more ethical, and ultimately, more productive, because it harnesses AI's power while preserving the irreplaceable human capacity for judgment, creativity, and moral agency. It replaces the black box of "magic AI" with the transparent, auditable, and human-governed workflow of the `Plan-Then-Execute` pattern.

The path forward is clear. The technical feasibility is established. The legal and ethical alignment is precise. What remains is the collective will to build it. This architecture is not a finished product, but a starting point—a reference, a blueprint, and a call to action. It is an invitation to every developer, every researcher, and every citizen to participate in the creation of a future where human augmentation does not erode our humanity, but instead, amplifies it, with our sovereignty as its unwavering, unbreakable foundation.

Reference

1. https://cdn.qwenlm.ai/qwen_url_parse_to_markdown/system00-0000-0000-0000-webUrlParser?key=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJyZXNvdXJjZV91c2VyX2lkIjoicXdlbl91cmxfcGFyc2VfdG9fbWFya2Rvd24iLCJyZXNvdXJjZV9pZCI6InN5c3RlbTAwLTAwMDAtMDAwMC0wMDAwLXdlYlVybFBhcnNlciIsInJlc291cmNlX2NoYXRfaWQiOm51bGx9.cz1eeZEZdaQH5CgUaxwUmfEJfqTOZMoh3PbosHslSPA
2. https://cdn.qwenlm.ai/qwen_url_parse_to_markdown/system00-0000-0000-0000-webUrlParser?key=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJyZXNvdXJjZV91c2VyX2lkIjoicXdlbl91cmxfcGFyc2VfdG9fbWFya2Rvd24iLCJyZXNvdXJjZV9pZCI6InN5c3RlbTAwLTAwMDAtMDAwMC0wMDAwLXdlYlVybFBhcnNlciIsInJlc291cmNlX2NoYXRfaWQiOm51bGx9.cz1eeZEZdaQH5CgUaxwUmfEJfqTOZMoh3PbosHslSPA
3. (PDF) On Neurorights https://www.researchgate.net/publication/354839078_On_Neurorights
4. Blockchain Cross-Chain Bridge Security: Challenges, ... https://dl.acm.org/doi/full/10.1145/3696429
5. Digital Product Passports Evolve from Static to Living ... https://www.linkedin.com/posts/emmanuel-adinyira-1098941a_blockchain-digital-twin-integration-in-construction-activity-7419401191278702592-St_z
6. Designing a Token Economy: Incentives, Governance and ... https://www.researchgate.net/publication/371314053_Designing_a_Token_Economy_Incentives_Governance_and_Tokenomics
7. A review of the key challenges of non-fungible tokens https://www.sciencedirect.com/science/article/pii/S0040162522007697
8. A Survey of Vibe Coding with Large Language Models https://arxiv.org/html/2510.12399v1

9. A Neural-Symbolic learning framework to produce ... https://theses.hal.science/tel-03982367v1/file/BENNETOT_Adrien_these_2022.pdf
10. Open tracing tools: Overview and critical comparison https://www.sciencedirect.com/science/article/pii/S0164121223001887
11. Issues and Their Causes in WebAssembly Applications https://dl.acm.org/doi/fullHtml/10.1145/3661167.3661227
12. Protecting Systems from Exploits Using Language- ... https://search.proquest.com/openview/3257085a04e81336537b2d662af653c5/1?pq-origsite=gscholar&cbl=18750&diss=y
13. Memory-Safety Challenge Considered Solved? An In- ... https://dl.acm.org/doi/fullHtml/10.1145/3466642
14. A Low-Latency Optimization of a Rust-Based Secure ... https://www.mdpi.com/1424-8220/22/22/8700
15. SafeDrop: Detecting Memory Deallocation Bugs of Rust ... https://www.researchgate.net/publication/363007258_SafeDrop_Detecting_Memory_Deallocation_Bugs_of_Rust_Programs_via_Static_Data-Flow_Analysis
16. Herik Lima's Post https://www.linkedin.com/posts/heriklima_rustreality-unsaferust-cplusplus-activity-7413006188956696576-vtV0
17. 2025 Trust and neural information processing systems https://arxiv.org/pdf/2401.08064
18. Neuromarketing algorithms' consumer privacy and ethical ... https://www.tandfonline.com/doi/full/10.1080/23311975.2024.2333063
19. Physiological Barriers to Nucleic Acid Therapeutics and ... https://www.mdpi.com/1999-4923/17/10/1309
20. Is Your Neural Data Part of Your Mind? Exploring the ... https://pmc.ncbi.nlm.nih.gov/articles/PMC8460199/
21. Neurotechnology Toolkit https://www.oecd.org/content/dam/oecd/en/topics/policy-sub-issues/emerging-technologies/neurotech-toolkit.pdf
22. Neurotechnology https://www.ohchr.org/sites/default/files/documents/hrbodies/hrcouncil/advisorycommittee/neurotechnology/03-ngos/ac-submission-cso-neurorightsfoundation.pdf
23. Neurorights as reconceptualized human rights https://www.frontiersin.org/journals/political-science/articles/10.3389/fpos.2023.1322922/full
24. “You shall have the thought”: habeas cogitationem as a New ... https://link.springer.com/article/10.1007/s12152-024-09551-8

25. International Human Rights Protection Gaps in the Age of ... https://ntc.columbia.edu/wp-content/uploads/2022/05/NeurorightsFoundationPUBLICAnalysis5.6.22.pdf
26. (PDF) Neurorights, Mental Privacy, and Mind Reading https://www.researchgate.net/publication/382079309_Neurorights_Mental_Privacy_and_Mind_Reading
27. Beyond neural data: Cognitive biometrics and mental privacy https://www.cell.com/cms/10.1016/j.neuron.2024.09.004/attachment/8def5fc2-7d76-4c83-b118-053ee5e43668/mmc2.pdf
28. The protection of mental privacy in the area of neuroscience https://www.europarl.europa.eu/RegData/etudes/STUD/2024/757807/EPRS_STU(2024)757807_EN.pdf
29. Neurorights (Chapter 26) - The Cambridge Handbook of ... https://www.cambridge.org/core/books/cambridge-handbook-of-the-right-to-freedom-of-thought/neurorights/B1AEF25AD18D9C8164CE9B366979B664
30. On Neurorights https://www.frontiersin.org/journals/human-neuroscience/articles/10.3389/fnhum.2021.701258/full
31. Novel Neurorights: From Nonsense to Substance - PMC https://pmc.ncbi.nlm.nih.gov/articles/PMC8821782/
32. The risks and challenges of neurotechnologies for human ... https://unesdoc.unesco.org/ark:/48223/pf0000384185
33. 333333 23135851162 the 13151942776 of 12997637966 ftp://ftp.cs.princeton.edu/pub/cs226/autocomplete/words-333333.txt
34. english-words.txt https://miller.readthedocs.io/en/latest/data/english-words.txt
35. Computation and Language 2025 https://www.arxiv.org/list/cs.CL/2025?skip=18325&show=2000
36. PEARC19 Program - PEARC Conference Series https://pearc.acm.org/archive/schedule/pearc19/PEARC19-schedule-with-abstract.pdf
37. Thermal Side-Channel Threats in Densely Integrated ... https://www.researchgate.net/publication/396455915_Thermal_Side-Channel_Threats_in_Densely_Integrated_Microarchitectures_A_Comprehensive_Review_for_Cyber-Physical_System_Security
38. Fundamentals and Practical Implications of Agentic AI https://arxiv.org/html/2505.19443v1
39. Fundamentals and Practical Implications of Agentic AI https://arxiv.org/pdf/2505.19443
40. MIT Projects | PDF | Electric Motor | Amplifier https://www.scribd.com/document/800262649/MIT-Projects

41. IoT-Enabled Tokenization of Physical Assets https://www.sec.gov/files/ctf-written-input-daniel-bruno-corvelo-costa-092125.pdf
42. Towards Rust in Windows Drivers https://techcommunity.microsoft.com/blog/windowsdriverdev/towards-rust-in-windows-drivers/4449718
43. Zero-knowledge proofs for anonymous authentication of ... https://www.sciencedirect.com/science/article/pii/S2590005625002176
44. A Survey on the Applications of Zero-Knowledge Proofs https://arxiv.org/html/2408.00243v1
45. Promise of Zero ‐ Knowledge Proofs (ZKPs) for Blockchain ... https://onlinelibrary.wiley.com/doi/10.1002/spy2.461
46. The Power I Know: Zero-Knowledge Proofs and Their ... https://ieeexplore.ieee.org/iel8/6287639/10820123/11127078.pdf
47. (PDF) The Impact of Artificial Intelligence on Human Thought https://www.researchgate.net/publication/394942120_The_Impact_of_Artificial_Intelligence_on_Human_Thought
48. Human Brain Project Specific Grant Agreement 3 | HBP SGA3 https://cordis.europa.eu/project/id/945539/results
49. 大会赞助商和参展商| NVIDIA GTC 圣何塞2026 https://www.nvidia.cn/gtc-global/sponsors/
50. Privacy-preserving communications for IoT based on DNS ... https://theses.hal.science/tel-04823789v1/file/146707_AYOUB_2024_archivage.pdf
51. Azure updates https://azure.microsoft.com/updates?id=553532
52. Identity Management Systems: A Comprehensive Review https://www.researchgate.net/publication/395516555_Identity_Management_Systems_A_Comprehensive_Review
53. (PDF) The Flex Model of Blended Learning Enabled Digital ... https://www.researchgate.net/publication/341094045_The_Flex_Model_of_Blended_Learning_Enabled_Digital_Citizenship
54. Sensors, Volume 24, Issue 17 (September-1 2024) https://www.mdpi.com/1424-8220/24/17
55. C2SaferRust: Transforming C Projects into Safer Rust with ... https://www.arxiv.org/pdf/2501.14257
56. Computer Safety, Reliability, and Security: Andrea ... https://www.scribd.com/document/940628971/978-3-031-68606-1
57. Subgraph Infrastructure Issues: RPC Rejected Calls, Block ... https://www.linkedin.com/posts/victorfei_3-reasons-your-subgraph-infra-sucks-but-activity-7402425508761817089--uyv

58. Excluded from school: Autistic students' experiences of ... https://www.researchgate.net/publication/320983978_Excluded_from_school_Autistic_students'_experiences_of_school_exclusion_and_subsequent_re-integration_into_school

59. Foreword https://unesdoc.unesco.org/ark:/48223/pf0000260629

60. Université Paris Cité https://theses.hal.science/tel-05086800v1/file/va_Hadroug_Jihed.pdf

61. Decoding the brain: from neural representations to ... - PMC https://pmc.ncbi.nlm.nih.gov/articles/PMC11637322/

62. Deep learning: Historical overview from inception to ... https://www.sciencedirect.com/science/article/pii/S1568494625006891

63. A Survey of Learning-based Automated Program Repair https://dl.acm.org/doi/10.1145/3631974

64. Culture and the digital city: its impact and influence https://unesdoc.unesco.org/ark:/48223/pf0000260637

65. Alphabetical-list.txt https://documents1.worldbank.org/curated/en/577031492972227204/txt/Alphabetical-list.txt

66. Seamless Continuous Integration / Continuous Delivery (CI ... https://theses.hal.science/tel-05263254v1/file/these.pdf

67. Private, Verifiable, and Auditable AI Systems https://arxiv.org/html/2509.00085v1

68. PermRust: A Token-based Permission System for Rust https://arxiv.org/pdf/2506.11701

69. A Survey of Learning-based Automated Program Repair https://dl.acm.org/doi/full/10.1145/3631974

70. ROSE: A Neurocomputational Architecture for Syntax - PMC https://pmc.ncbi.nlm.nih.gov/articles/PMC10055479/

71. Automatic Identification of Similar Pull-Requests in ... https://www.mdpi.com/2078-2489/13/2/73

72. Records of the General Conference of UNESCO, fifth ... https://unesdoc.unesco.org/ark:/48223/pf0000214791

73. Contents of Current Periodicals https://www.jstor.org/stable/pdf/2565536.pdf

74. (PDF) The blockchain conundrum: An in‑depth ... https://www.researchgate.net/publication/376653537_The_blockchain_conundrum_An_in-depth_examination_of_challenges_contributing_technologies_and_alternatives

75. (PDF) A Comprehensive Review of Blockchain Consensus ... https://www.researchgate.net/publication/350031088_A_Comprehensive_Review_of_Blockchain_Consensus_Mechanisms

76. (PDF) The Evolution and Optimization Strategies of a PBFT ... https://www.researchgate.net/publication/390277794_The_Evolution_and_Optimization_Strategies_of_a_PBFT_Consensus_Algorithm_for_Consortium_Blockchains

77. Recent Advances in Sharding Techniques for Scalable ... https://www.researchgate.net/publication/387411710_Recent_Advances_in_Sharding_Techniques_for_Scalable_Blockchain_Networks_A_Review/download

78. (PDF) A Survey of IoT and Blockchain Integration: Security ... https://www.researchgate.net/publication/356402274_A_Survey_of_IoT_and_Blockchain_Integration_Security_Perspective

79. Blockchain-based trust management in cloud computing ... https://www.researchgate.net/publication/352615623_Blockchain-based_trust_management_in_cloud_computing_systems_a_taxonomy_review_and_future_directions

80. AI for Good Global Summit 2024 - ITU https://aiforgood.itu.int/summit24/

81. Data for Policy 2025 (DfP'25) - Europe Book of Abstracts https://zenodo.org/records/15675928/files/Data%20for%20Policy%202025%20-%20Europe_Book%20of%20Abstracts_160625.pdf?download=1

82. (PDF) Adopting Agile Across Borders https://www.academia.edu/91360255/Adopting_Agile_Across_Borders

83. (PDF) Research study on securing the cloud https://www.researchgate.net/publication/399188798_Research_study_on_securing_the_cloud_utilizing_conventional_and_blockchain-based_access_control_mechanisms

84. Autonomous Agents on Blockchains: Standards, Execution ... https://www.researchgate.net/publication/399595458_Autonomous_Agents_on_Blockchains_Standards_Execution_Models_and_Trust_Boundaries

85. glove.6B.100d.txt-vocab.txt https://worksheets.codalab.org/rest/bundles/0xadf98bb30a99476ab56ebff3e462d4fa/contents/blob/glove.6B.100d.txt-vocab.txt

86. How the Private and the Public Sectors Use Intellectual ... https://www.wipo.int/edocs/pubdocs/en/global_challenges/1027/wipo_pub_1027.pdf

87. housing what's next? from thinking the unit to building the city https://publications.iadb.org/publications/english/document/Housing_Whats_Next_From_Thinking_The_Unit_to_Building_The_City.pdf

88. Advanced and sustainable technologies for automotive ... https://www.researchgate.net/profile/Ingrid-Winkler/publication/370505004_Augmented_reality_head-

up_display_interfaces_for_advanced_driver_assistance_systems_a_reference_architecture/links/6497365a95bbbe0c6ef030f7/Augmented-reality-head-up-display-interfaces-for-advanced-driver-assistance-systems-a-reference-architecture.pdf

89. Talent development in Mexico: challenges and opportunities https://www.researchgate.net/publication/305279579_Talent_development_in_Mexico_challenges_and_opportunities
90. news_vocab_sorted.txt https://www.cs.cmu.edu/afs/cs.cmu.edu/academic/class/15122-s14/www/14-interfaces/news_vocab_sorted.txt