

A Technical and Governance Blueprint for the Neuro-XR Ontology Pipeline

Foundations of the Neuro-XR Ontology: From Semantic Formalism to Machine-Readable Constraints

The development of a robust, scalable, and ethically-grounded ecosystem for cybernetic, augmented-human, and wetware-related XR objects hinges on the creation of a foundational ontology [22](#). This document establishes the formal semantics that govern what entities exist, their properties, and the relationships between them, serving as the central nervous system for the entire Neuro-XR Forge pipeline [22](#). Unlike dictionaries, which merely define words, or knowledge graphs without semantic constraints, an ontology provides a rigorous framework that enables reliable reasoning and prevents the kind of statistical guesswork that leads to AI hallucination [22](#). The primary challenge is to design an upper-level ontology that is both expressive enough to capture the complexity of neuromorphic sensors, brain-computer interfaces (BCIs), and biohybrid robotics, and precise enough to serve as a basis for automated schema generation and runtime enforcement. The initial step involves defining a set of core classes that form the bedrock of the domain. The proposal suggests starting with `NeuroSignalChannel`, `NeuroActuator`, `SafetyEnvelope`, and `XRInteractionPrimitive` as these represent the fundamental building blocks of any neuro-augmented XR experience [17](#). Each of these classes must be meticulously defined within a formal language like Web Ontology Language (OWL) or Resource Description Framework (RDF). The choice of methodology is critical; reusing open ontology practices, such as those employed by the OBO Foundry, ensures alignment with community-vetted best practices and facilitates interoperability with existing biomedical and scientific ontologies, providing a proven and robust starting point [18](#) [22](#).

The `NeuroSignalChannel` class, for instance, would serve as the parent class for all data streams originating from a user's nervous system. Its subclasses could include `EEG_Data_Channel`, `fMRI_Data_Channel`, `Peripheral_Nervous_System_Data_Channel`, and `Tactile_Sensor_Data_Channel`. This hierarchical structure allows for the clustering of devices based on their source of information, enabling standardized capability vectors

to be attached to each type [16](#). For example, a `Facial_Emotion_Data_Channel` subclass could be linked to ontologies used in behavior-change studies, providing templates for modeling emotional responses [18](#). Similarly, the `NeuroActuator` class would encompass devices that translate neural signals into physical actions within the XR environment, such as `Virtual_Limb_Controller` or `Haptic_Feedback_Module`. The `SafetyEnvelope` class is a direct implementation of the principle of mental integrity, one of the four pillars of Neurorights, which protects an individual from harmful interference with their brain function [1](#). This class would contain mandatory facets that dictate safety protocols, data handling procedures, and risk mitigation strategies for any object it envelops. Finally, the `XRInteractionPrimitive` class would define the basic ways in which users interact with these objects, leveraging standardized hand interaction primitives that are becoming part of the OpenXR standard, such as those promoted from vendor extensions to `XR_EXT_hand_interaction` [10](#). By grounding the ontology in these core concepts, the pipeline creates a consistent vocabulary that is essential for the subsequent steps of schema generation, conformance testing, and cross-platform adaptation.

Once the conceptual classes are defined in a formal language like OWL, the next critical step is to automate the generation of machine-readable schemas, specifically Abstract Logics Notation (ALN) and JSON Schema, from this ontology backbone [9](#). This automation is paramount for scalability, ensuring that every concrete object definition adheres strictly to the rules and constraints laid out in the upper-level ontology. The process involves using a reasoner to infer all subclass relationships, property domains, and range restrictions from the OWL file and then translating these formal axioms into the syntax of JSON Schema. For example, if the ontology specifies that a `NeuroSignalChannel` must have a `data_sensitivity` property of type string with an enumeration of 'low', 'medium', 'high', and 'extreme', the schema generator will produce a corresponding JSON Schema definition. This generated schema becomes the authoritative specification for any object instance, guaranteeing structural and logical consistency across the ecosystem. To enforce these rules at runtime, a high-performance JSON Schema validator is required. Libraries like Ajv (Another JSON Schema validator for JavaScript) are ideal for this purpose, as they compile schemas into highly optimized executable functions for fast validation [9](#). This compiled validator can check not just basic data types and formats but also complex conditional logic and custom keywords [9](#). This capability is crucial for implementing the governance track of the pipeline, allowing for the embedding of domain-specific safety and ethics checks directly into the schema itself. For instance, a custom keyword could be created to validate that an object's consent flow specification adheres to the requirements of a specific state's neural data law, such as Colorado's mandate for separate consent before third-party disclosure [6](#).

This tight integration of semantic formalism with practical implementation transforms the ontology from an abstract blueprint into a powerful tool for ensuring trustworthiness and compliance at scale.

Ontology Class	Subclasses / Instances	Rationale and Connections
NeuroSignalChannel	EEG_Data_Channel, fMRI_Data_Channel, Peripheral_Nervous_System_Data_Channel, Tactile_Sensor_Data_Channel, Facial_Emotion_Data_Channel	Represents the source of neural and behavioral data. Clustering by modality enables standardized capability vectors and leverages existing behavior-change ontologies as modeling templates 18 . 13 16 19
NeuroActuator	Virtual_Limb_Controller, Haptic_Feedback_Module, Eye_Tracker_Headset	Represents devices that translate neural intent into action in the XR world. This class is fundamental for closed-loop systems used in BCI applications 16 . 13 19
SafetyEnvelope	N/A (Abstract Parent)	Directly implements the "mental integrity" right, protecting against harmful interference with brain function 1 . Contains mandatory facets for data sensitivity, consent, and logging obligations 17 . 1 33
XRInteractionPrimitive	Hand_Grab_Primitive, Voice_Command_Primitive, Gaze_Selection_Primitive	Defines the basic ways users interact with XR objects. Leverages standardized interactions being incorporated into OpenXR, such as XR_EXT_hand_interaction 10 . 3 4

This structured approach ensures that the ontology is not merely a static classification system but a dynamic and evolving foundation for the entire Neuro-XR ecosystem. It provides the formal semantics necessary for reliable AI-driven discovery and synthesis while simultaneously offering the machine-actionable constraints required for verifiable compliance with neurorights and other regulatory frameworks. The success of the pipeline depends entirely on the rigor and clarity of this foundational layer, which must be continuously refined through expert review and feedback loops from real-world implementations 11 .

OpenXR Alignment and Cross-Platform Execution:

Bridging Standards and Engines

For the Neuro-XR Forge pipeline to deliver on its promise of creating portable, interoperable objects, its output must be deeply aligned with the OpenXR standard, the industry's leading effort toward cross-platform compatibility in extended reality [2](#). The cornerstone of this alignment is the use of OpenXR Spatial Entities Extensions (`XR_EXT_spatial_entities`) and its associated components [2](#). These Khronos Group-sanctioned extensions provide a vendor-neutral framework for representing virtual content as persistent, identifiable objects within a shared physical space. They introduce key concepts such as UUIDs for unique identification, anchors for precise spatial anchoring to real-world locations, and persistence mechanisms for retaining context across different sessions [2](#) [8](#). Every new object template generated by the pipeline must be explicitly mapped to these spatial entities, ensuring that a `TactileEskinPatch` or an `EventVisionSensor3D` defined in the ontology has a clear, unambiguous representation within any compliant OpenXR runtime, whether on AndroidXR, PC, or console systems [8](#). This mapping is non-negotiable for achieving true portability and is the primary mechanism through which the pipeline enforces cross-platform consistency. The OpenXR loader discovers the active runtime via platform-specific methods—registry queries on Windows, JSON manifest files on Linux, and ContentProviders on Android—and uses the information within these manifests to bind runtimes to the OpenXR API [8](#). The pipeline's output must therefore include artifacts that correctly configure these manifests, registering the new neuro-XR features and ensuring they are recognized by the underlying driver stack.

A critical component of the OpenXR ecosystem for extending functionality is its feature extension mechanism, which is particularly well-documented in the Unity OpenXR Provider [3](#). This mechanism allows developers to declare custom C# scripts that inherit from the `OpenXRFeature` class and are annotated with attributes specifying their name, version, and, most importantly, the OpenXR extension strings they require (e.g., `XR_UNITY_mock_driver`) [4](#). At runtime, the application can check for the availability of a specific extension using `OpenXRRuntime.IsEnabled()`, allowing for graceful degradation or conditional activation of advanced features [4](#). The Neuro-XR pipeline must leverage this exact pattern to propose and implement new neuro-centric extensions. For example, when a developer wants to use a standardized pose for an EEG cap or access XR-safe "brain state indicators," the pipeline would generate specifications for a new extension, say `XR_KHR_neuro_sensing`, and provide the necessary code to declare and probe for it. This ensures that neuro-augmented capabilities are not treated

as proprietary add-ons but are integrated into the standard OpenXR feature lifecycle, making them discoverable and usable by any conformant application and engine. The recent evolution of the OpenXR specification, with versions 1.1.53 and 1.1.54 introducing new vendor extensions for facial tracking and hand interaction, demonstrates that this is an active area of development and presents an opportunity for the Neuro-XR pipeline to contribute standardized neuro-XR extensions to the official Khronos repository [10](#).

With a solid foundation in OpenXR, the next step is to build the bridge to major game engines like Unity, Unreal, and Godot. Research Action 5 calls for the implementation of engine plugins that map the canonical object spec to engine-specific representations like Unity's prefabs, Unreal modules, and Godot nodes [3](#). This is achieved through a process of abstraction and code generation. The pipeline takes the JSON Schema definition of a Neuro-XR object and generates boilerplate code for each target engine. For Unity, this might involve generating a C# script that inherits from `XRGrabInteractable` from the XR Interaction Toolkit, populating it with the object's properties from the schema, and creating a prefab asset that bundles the script with its visual and audio components [3](#). For Unreal, it would involve generating a Blueprint or C++ class that encapsulates the object's behavior. For Godot, it would create a Node-based scene graph. The key insight here is that the engine adapter acts as a thin translation layer, reading the same object specification and instantiating the appropriate components native to that engine. This approach preserves the integrity of the canonical object definition while abstracting away the vast differences in engine architecture, thus fulfilling the promise of "cross-engine representation" [3](#). The OpenXR loader is the ultimate runtime surface, and the engine adapters ensure that the objects' behaviors are correctly interfaced with the underlying OpenXR runtime, which handles the low-level communication with hardware drivers. This layered architecture—from the canonical schema down to the engine adapter and finally to the OpenXR runtime—is the only viable path to achieving seamless execution across disparate platforms. The success of this strategy is contingent on the quality and completeness of the mappings, requiring deep expertise in each target engine's APIs and best practices for XR development.

Implementing Neurorights by Design: Translating Legislation into Enforceable Schemas

The successful deployment of neuro-augmented XR technology is inextricably linked to the establishment of robust governance and privacy protections. The "neurorights"

movement seeks to codify rights related to mental privacy, cognitive liberty, mental integrity, and fair access to cognitive enhancements into law [1](#) . The first jurisdiction to do so was Chile, which enshrined five neurorights into its constitution, setting a global precedent [1](#) [26](#) . In the United States, this concept is rapidly gaining traction at both the state and federal levels. Colorado's Senate Bill 21-190 (SB21-190) and California's Senate Bill 1223 (SB 1223) have enacted some of the world's first laws specifically governing the collection and processing of neural data [26](#) [34](#) . These laws, along with the proposed Management of Individuals' Neural Data Act of 2025 (MIND Act), provide a rich and detailed legislative framework that must be translated into the very fabric of XR object definitions [7](#) [32](#) . This is the central task of Research Action 6: defining mandatory safety and ethics facets on every object, ensuring that any "unknown" object is born with a safety envelope and audit hooks [17](#) . This requires moving beyond simple metadata tags and embedding machine-enforceable policies directly into the object's JSON Schema.

The legal definitions of "neural data" vary slightly between jurisdictions, and these nuances must be captured in the ontology. Colorado's law defines neural data as "information about bodily or mental functions" and treats it as a distinct category of sensitive personal data [26](#) . California's law, while similar, explicitly excludes data inferred from non-neural sources, such as heart rate or eye movement, which creates a legal distinction that the schema must reflect [5](#) [34](#) . The MIND Act offers an even broader definition, covering any information obtained by measuring the activity of an individual's central or peripheral nervous system, including voice analysis and facial expressions, because it can reveal thoughts, emotions, or decision-making patterns [7](#) . The schema for a `NeuroSignalChannel` must therefore include fields to classify the data source and sensitivity according to these legal distinctions. For example, a field like `neural_data_classification` could be an enumeration with values such as 'direct_cns_measurement', 'direct_pns_measurement', 'inferred_from_non_neural_data', or 'behavioral_signal'. This classification is not merely informational; it triggers different sets of compliance rules. An object collecting direct CNS measurement data under the Colorado law would be subject to its strict opt-in consent requirements for all collection and use [6](#) . The schema must encode these requirements, perhaps through a mandatory `consent_requirements` object that specifies the type of consent needed (e.g., 'opt_in_for_collection', 'separate_opt_in_for_disclosure'), the scope of the data being collected, and the duration of the consent.

Furthermore, the schema must incorporate granular details about data handling and user rights, as mandated by these nascent regulations. Minnesota's Neurodata Bill SF 1110, for instance, mandates unprecedented dynamic consent requirements, where users must be given an independent notice for each BCI connection and a separate, granular consent

form for each planned use and for sharing with each third party [33](#) . This level of detail is achievable through the schema. A `consent_flow_specification` field could define a multi-step process, outlining the disclosures required before each stage of data processing. The schema could also include fields for `user_rights_clauses`, which would specify the rights granted to the user, such as the ability to access, correct, and delete their neural data, mirroring the deletion rights mandated by both Colorado and California laws [6](#) . To make these policies actionable, the schema can be validated against them using a high-performance validator like Ajv [9](#) . Custom Ajv keywords can be written to enforce these complex business and legal rules. For example, a keyword named `validatesAgainstConsentFlow` could inspect the `consent_flow_specification` and compare it against the `allowed_inferences` and `third_party_sharing_policy` fields, flagging any object that attempts to perform a secondary use without the requisite separate consent. This turns the schema from a passive data container into an active compliance instrument, ensuring that every object shipped from the Neuro-XR Forge is inherently respectful of neurorights.

Regulatory Requirement	Corresponding Schema Facet(s)	Enforcement Mechanism
Opt-In Consent for Collection/Processing	<code>consent_requirements.collection_type: "opt_in"</code>	Ajv validator checks that this flag is present and that a <code>consent_flow_specification</code> exists before any processing logic is executed. 5 6
Separate Consent for Third-Party Disclosure	<code>consent_requirements.disclosure_type: "separate_opt_in_required"</code>	Validator enforces that a distinct consent event must be triggered and logged before data is shared with any third party. 6
Right to Deletion	<code>user_rights.deletion_enabled: true</code>	Schema includes a <code>deletion_policy</code> field that defines how and when data is purged, which the runtime must honor upon user request. 6
Dynamic Consent for Multiple Uses	<code>dynamic_consent.enabled: true, use_cases: Array of objects with consent_form_url</code>	The schema defines granular use cases, and the runtime must present a separate consent interface for each one, logging acceptance separately. 33
Data Sensitivity Classification	<code>neural_data_classification: enum('cns', 'pns', 'inferred_non_neural', 'behavioral')</code>	Classification determines which legal framework applies and what default security and privacy measures are enforced by the runtime. 7 34
Logging Obligations	<code>logging_requirements.audit_trail: true, required_log_fields: Array</code>	Schema mandates the recording of specific events (e.g., consent changes, data access requests) with sufficient detail for an external auditor to trace actions. 17

By systematically mapping these legal requirements to schema facets and enforcing them programmatically, the Neuro-XR pipeline creates a "trust-by-design" environment. This approach not only helps developers comply with current and future regulations but also builds consumer confidence in a technology that operates on the most intimate human data: our own minds. The continuous feedback loop from the runtime, where objects

must pass ethical as well as functional checks, ensures that the governance model remains relevant and effective as the technology and the law evolve [27](#).

The Conformance and Curation Framework: Ensuring Trust and Interoperability at Scale

As the Neuro-XR Forge pipeline begins to generate a growing corpus of objects, maintaining coherence, safety, and interoperability becomes a paramount challenge. A purely automated system risks introducing unsafe patterns or redundant object definitions. Therefore, a robust governance and curation framework is essential. This framework is centered around the concept of a Multi-Party NeuroXR Board, a body tasked with reviewing newly proposed objects, ratifying batches of new entries, and deprecating unsafe or obsolete patterns [10](#). The composition of this board must be intentionally multidisciplinary to provide holistic oversight. Drawing inspiration from established protocols like ONRAMP-AI-VRAR, the board should include experts from diverse fields such as clinical psychology, VR/AR software development, bioethics and neuroethics, data privacy and health law, and patient advocacy [11](#). This composition ensures that technical feasibility is balanced with ethical considerations, clinical safety, and user rights. The board's responsibilities would extend beyond mere approval; they would include approving mitigation plans for identified risks, conducting equity audits to prevent bias, and vetting software updates for ongoing legal and ethical alignment [11](#). Regular "ontology sprints" could be organized to handle batches of new proposals, keeping the ecosystem coherent even as it scales to thousands of entries [10](#).

The authority of the NeuroXR Board must be backed by a rigorous conformance testing framework, analogous to the OpenXR Conformance Test Suite (CTS) [10](#). Research Action 9 outlines the necessity of creating CTS-style tests and shared reference scenes for each object family to validate behavior, latency, and safety bounds across different devices and platforms [10](#). However, for neuro-XR objects, the scope of conformance testing must be expanded far beyond traditional functional metrics. The tests must include explicit validation of the object's neurorights compliance profile. This means creating test vectors that simulate privacy and consent flows to ensure that implementations correctly adhere to the ethical as well as the functional checks. For example, a conformance test for an EEG sensor object could include a scenario that verifies it fails to process any data until it receives a valid opt-in consent signal, thereby validating its compliance with Colorado's neural data law [6](#). Another test could simulate a third-party data sharing request and

verify that the object correctly prompts for and logs a separate piece of consent before transmitting the data. These tests would be integrated into the CI/CD pipeline for developers, meaning that an object cannot be certified for distribution unless it passes both its functional and its ethical conformance tests.

To further enhance transparency and trust, the entire process should be linked to formal auditing standards, such as those outlined in the European Union's Digital Services Act (DSA) [27](#) . The DSA mandates annual systemic risk assessments and independent external audits for Very Large Online Platforms (VLOPs), requiring auditors to use sampling methodologies that minimize detection risk and justify their techniques in reports [27](#) . The NeuroXR Board could function as this designated oversight body, and the public registry of objects, where all proposed objects are logged with UUIDs, versioned, and made available for community feedback, would serve as the primary evidence repository for these audits . This registry, combined with mandatory audit trails built into every object's schema, creates an auditable trail of data handling, from collection to storage and potential sharing [17](#) . This level of transparency is crucial for meeting the stringent demands of regulators and for building a sustainable ecosystem where developers and users can have confidence in the safety and integrity of the objects they deploy and use. The combination of a multidisciplinary board for curation and a comprehensive, legally-grounded conformance testing framework provides the dual-layered oversight necessary to manage a complex, rapidly evolving technological landscape safely and responsibly.

Governance Component	Key Features	Purpose and Impact
Multi-Party NeuroXR Board	Multidisciplinary membership (bioethicists, engineers, lawyers, clinicians, patient advocates).	Provides holistic review and ratification of new objects, deprecation of unsafe patterns, and long-term ecosystem stewardship. Ensures a balance between innovation and safety. 10 11
Public Registry	All proposed objects logged with UUIDs, versioned, and open for community feedback.	Creates a transparent, auditable record of all objects. Enables gradual refinement into production-grade standards and fosters community participation in shaping the ecosystem.
Conformance Test Suite (CTS)	CTS-style tests for each object family. Includes functional metrics (latency, accuracy) and ethical checks (consent flows, data handling).	Guarantees interoperability and reliability. Hardens neurorights guarantees by making compliance a mandatory part of certification. Prevents unsafe objects from entering the ecosystem. 10 17
Auditable Trails	Mandatory <code>audit_trail</code> logging in object schemas for key events (consent changes, data access requests).	Provides evidence for independent external audits as required by regulations like the EU's DSA 27 . Builds trust by making data handling transparent and accountable. 17 25
Cross-Vendor App Stores	Gateways for distributing certified Neuro-XR objects.	Creates a powerful incentive for adherence to the NeuroXR standards by controlling access to marketplaces. Centralizes distribution and enforcement.

This comprehensive framework transforms the Neuro-XR Forge from a simple object generator into a trusted ecosystem. It institutionalizes the principles of safety, ethics, and interoperability, ensuring that as the number of objects grows from hundreds to thousands, the overall quality and safety of the environment does not degrade. The continuous feedback loop, where reference scenes and app stores feed back into the ontology forge for expansion, completes the cycle, creating a self-improving system grounded in policy and verified by rigorous testing .

Composable Architectures and End-to-End Proofs-of-Concept

To move beyond theoretical design and demonstrate tangible viability, the Neuro-XR Forge pipeline must embrace composable architectures, adapting concepts like "XR Blocks" into more sophisticated "NeuroBlocks" . A NeuroBlock represents a reusable, self-contained bundle of XR object components, designed to simplify the creation of complex neuro-augmented experiences. Each NeuroBlock would package together a core definition—a neuromorphic sensor, a decoding model, a safety profile, and an interaction contract—into a single, manageable unit . This modular approach is inspired by modern software engineering practices that prioritize composability and reusability, such as those seen in monorepo structures that facilitate collaborative development [24](#) . The power of this architecture lies in its ability to auto-synthesize thousands of variants from a single base schema. For example, a base `TactileEskinPatch` NeuroBlock could be configured with different sensor ranges, varying levels of privacy protection (e.g., anonymized vs. identifiable data streams), and different task profiles (e.g., gaming, teleoperation, medical diagnostics). Each configuration would result in a unique, yet semantically consistent, object instance, drastically reducing the development overhead for creating specialized XR experiences. This synthetic generation capability is crucial for scaling the ecosystem to meet the diverse needs of various application domains, from education to healthcare and professional training [19](#) [24](#) .

The end-to-end proof-of-concept envisioned in the user's prioritization (RA4–RA5) requires integrating several of these components into a working demonstration. The first step would be to adapt the "XR Blocks" idea into composable "NeuroBlocks" that bundle a neuromorphic sensor, a decoding model, a safety profile, and an interaction contract for reuse in Unity, Unreal, and Godot . A prototype NeuroBlock could be created for a simple tri-manual hybrid BCI-VR framework, which integrates eye-tracking, single-channel EEG, and non-haptic controllers to enable control of two biological hands and one virtual third

limb [16](#). The NeuroBlock would contain the URIs for the sensor definitions (from the ontology), a placeholder for the decoding model (which could be a pre-trained ML model hosted elsewhere), a safety profile derived from neurorights principles, and an interaction contract specifying how gaze and attention signals trigger the virtual hand [16](#). This block could then be deployed across different engines using the engine adapters developed in RA5. For instance, in Unity, the plugin would instantiate the NeuroBlock as a prefab using the XR Interaction Toolkit, ensuring it correctly interfaces with the OpenXR runtime via the feature extension mechanism [3](#). The resulting experience would allow a user to activate a virtual hand simply by sustaining their attention on a target for over 300 milliseconds, a feature that demonstrated an 87.5% success rate in trials [16](#).

To demonstrate the full pipeline's discovery and standardization capabilities (RA7), the prototype could be extended to mine academic literature. Using a process akin to LLM-assisted extraction, the pipeline could take a natural language description of a novel device from a recent IEEE VR paper and auto-generate a draft NeuroBlock definition [18](#). This draft would be submitted to the public registry with a "proto-object" status. It would then undergo expert review by the Multi-Party NeuroXR Board, who would assess its novelty, safety, and alignment with the existing ontology. If ratified, the proto-object would be promoted into a fully specified class with a permanent URI and OpenXR bindings, completing the discovery-to-standardization cycle [5](#). This process is vital for ensuring the pipeline remains responsive to cutting-edge research and innovation, feeding richer object candidates into the already-standing system. The final element of the proof-of-concept would be to connect these objects to a larger context, such as a simulated Phoenix smart city environment [15](#). Here, XR objects could be anchored to real-world locations using OpenXR's spatial persistence extensions, and their behaviors could be governed by agent-to-agent protocols for digital twin systems, enabling consent-aware, multimodal interactions across a large-scale urban deployment [14](#). By successfully executing this end-to-end workflow—from LLM-assisted discovery and prototyping to cross-platform deployment and large-scale contextualization—the Neuro-XR Forge pipeline would prove its value as a critical infrastructure for the next generation of immersive, neuro-integrated technologies.

Strategic Synthesis and Actionable Roadmap

In conclusion, the proposed "XR Neuro-Object Forge" pipeline represents a comprehensive and forward-looking solution to the pressing need for standardization, interoperability, and ethical governance in the rapidly expanding field of neuro-augmented XR. My deep analysis of the user's ten research actions, synthesized with a rich body of contextual evidence, reveals that the project's success is contingent on a tightly integrated, dual-track approach. The technical track, focused on creating an interoperable, composable, and verifiable foundation, is inseparable from the governance and policy track, which embeds legal and ethical principles directly into the object definitions. The most significant insight is that neither track can succeed in isolation. The technical architecture, particularly the core ontology and its mapping to OpenXR, cannot be finalized without being explicitly grounded in the specific, machine-readable requirements of emerging neurorights legislation like the Colorado Neural Data Protection Act, California's CCPA amendment, and the federal MIND Act [7](#) [26](#) [34](#) . Conversely, the governance framework, however well-designed, will remain a theoretical construct if it cannot be implemented through concrete schemas, automated validation, and enforceable runtime checks [9](#) . The user's strategic prioritization of actions RA6–RA10 to establish the governance and verification mechanisms before scaling the input from RA1–RA3 is therefore not just sound, but essential for building a trustworthy and sustainable ecosystem.

Based on this comprehensive analysis, I propose the following phased roadmap for immediate deep investigation and prototyping, designed to deliver tangible results while building a secure and scalable foundation.

Phase 1: Establish the Core Foundation (RA2, RA6, RA9) The initial phase must focus on developing the foundational layers of the pipeline, establishing the core ontology and its governance enforcement mechanisms.

- **Develop the Minimal Viable Neuro-XR Ontology:** Begin by defining the upper-level classes (``NeuroSignalChannel``, ``NeuroActuator``, ``SafetyEnvelope``, ``XRInteractionPrimitive``) in OWL/RDF, grounding their axioms and properties in the explicit definitions and requirements from Colorado's SB21-190 and the MIND Act [7](#) [26](#) . This ensures the ontology is immediately relevant to real-world legal constraints.
- **Define Mandatory Safety and Consent Schema:** Translate key neurorights concepts into JSON Schema facets (``neurorights_compliance_profile``, ``consent_flow_specification``, ``data_protection_impact_assessment_summary``).

Use the Ajv library to create a high-performance validator that enforces these rules, acting as a gatekeeper for all object definitions [9](#) .

- **Prototype a Conformance Test Suite:** Select one concrete object class, such as a basic EEG sensor, and develop a reference conformance test. This test must validate both its functional properties (e.g., latency, signal-to-noise ratio) and its ethical properties, such as verifying that it requires an opt-in consent flag before processing any neural data, directly reflecting Colorado's opt-in requirement [6](#) .

Phase 2: Build the End-to-End Proof-of-Concept (RA4, RA5, RA7) With the foundation in place, the second phase focuses on demonstrating the full end-to-end workflow, from concept to a functional, cross-platform object.

- **Adapt the "XR Blocks" Concept:** Develop a prototype for a "NeuroBlock"—a composable bundle containing a sensor definition, a placeholder for a decoding model, a safety profile, and an interaction contract. Use the tri-manual hybrid BCI-VR framework as a case study to ground the prototype in a real-world application [16](#) .
- **Implement Engine Adapters:** Create a simple Unity plugin that maps a NeuroBlock definition to a prefab using the XR Interaction Toolkit. Ensure it correctly interfaces with the OpenXR runtime via the `OpenXRFeature` mechanism, demonstrating seamless integration with a major engine [3](#) .
- **Demonstrate Discovery and Ratification:** Mine a recent IEEE VR paper for a novel BCI metaphor. Use an LLM-assisted process to generate a draft NeuroBlock definition, which is then validated against the Phase 1 schema and reviewed by a mock-up of the Multi-Party NeuroXR Board. This validates the entire discovery-to-standardization pipeline.

Phase 3: Formalize Governance and Scaling (RA8, RA10) The final phase focuses on institutionalizing the governance model and preparing the pipeline for large-scale adoption.

- **Draft a Charter for the Multi-Party Board:** Outline the official composition, responsibilities, and decision-making processes of the NeuroXR Board, drawing on best practices from frameworks like ONRAMP-AI-VRAR and auditing standards from the EU's DSA [11](#) [27](#) .
- **Design a Public Registry:** Define the structure for a public registry where all proposed objects are logged with UUIDs, versioned, and made available for community feedback and validation, ensuring transparency and community involvement .

By following this integrated, three-phase roadmap, the project can systematically address the technical, ethical, and governance challenges inherent in the Neuro-XR domain. This approach moves the initiative from a visionary concept to a tangible, secure, and scalable solution that can guide the development of the next generation of immersive, neuro-integrated technologies, ensuring they are built on a foundation of trust, safety, and respect for human rights.

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. https://cdn.qwenlm.ai/qwen_url_parse_to_markdown/system00-0000-0000-0000-webUrlParser?key=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJyZXNvdXJjZV91c2VyX2lkIjoicXdlbl91cmxfcGFyc2VfdG9fbWFya2Rvd24iLCJyZXNvdXJjZV9pZCI6InN5c3RlbTAwLTAwMDAtMDAwMC0wMDAwLXdlYlVybFBhcnNlciIsInJlc291cmNlX2NoYXRfaWQiOm51bGx9.cz1eeZEZdaQH5CgUaxwUmfEJfqTOZMoh3PbosHslSPA
4. https://cdn.qwenlm.ai/qwen_url_parse_to_markdown/system00-0000-0000-0000-webUrlParser?key=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJyZXNvdXJjZV91c2VyX2lkIjoicXdlbl91cmxfcGFyc2VfdG9fbWFya2Rvd24iLCJyZXNvdXJjZV9pZCI6InN5c3RlbTAwLTAwMDAtMDAwMC0wMDAwLXdlYlVybFBhcnNlciIsInJlc291cmNlX2NoYXRfaWQiOm51bGx9.cz1eeZEZdaQH5CgUaxwUmfEJfqTOZMoh3PbosHslSPA
5. https://cdn.qwenlm.ai/qwen_url_parse_to_markdown/system00-0000-0000-0000-webUrlParser?key=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJyZXNvdXJjZV91c2VyX2lkIjoicXdlbl91cmxfcGFyc2VfdG9fbWFya2Rvd24iLCJyZXNvdXJjZV9pZCI6InN5c3RlbTAwLTAwMDAtMDAwMC0wMDAwLXdlYlVybFBhcnNlciIsInJlc291cmNlX2NoYXRfaWQiOm51bGx9.cz1eeZEZdaQH5CgUaxwUmfEJfqTOZMoh3PbosHslSPA

1cmxfcGFyc2VfdG9fbWFya2Rvd24iLCJyZXNvdXJjZV9pZCI6InN5c3RlbTAwLTAwMDA
tMDAwMC0wMDAwLXdlYlVybFBhcnNlciIsInJlc291cmNlX2NoYXRfaWQiOm51bGx9.cz
1eeZEZdaQH5CGUaxwUmfeJfqTOZMoh3PbosHslSPA

6. [https://cdn.qwenlm.ai/qwen_url_parse_to_markdown/system00-0000-0000-0000-webUrlParser?
key=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJyZXNvdXJjZV91c2VyX2lkIjoicXdlbl91cmxfcGFyc2VfdG9fbWFya2Rvd24iLCJyZXNvdXJjZV9pZCI6InN5c3RlbTAwLTAwMDA
tMDAwMC0wMDAwLXdlYlVybFBhcnNlciIsInJlc291cmNlX2NoYXRfaWQiOm51bGx9.cz
1eeZEZdaQH5CGUaxwUmfeJfqTOZMoh3PbosHslSPA](https://cdn.qwenlm.ai/qwen_url_parse_to_markdown/system00-0000-0000-0000-webUrlParser?key=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJyZXNvdXJjZV91c2VyX2lkIjoicXdlbl91cmxfcGFyc2VfdG9fbWFya2Rvd24iLCJyZXNvdXJjZV9pZCI6InN5c3RlbTAwLTAwMDA
tMDAwMC0wMDAwLXdlYlVybFBhcnNlciIsInJlc291cmNlX2NoYXRfaWQiOm51bGx9.cz
1eeZEZdaQH5CGUaxwUmfeJfqTOZMoh3PbosHslSPA)
7. [https://cdn.qwenlm.ai/qwen_url_parse_to_markdown/system00-0000-0000-0000-webUrlParser?
key=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJyZXNvdXJjZV91c2VyX2lkIjoicXdlbl91cmxfcGFyc2VfdG9fbWFya2Rvd24iLCJyZXNvdXJjZV9pZCI6InN5c3RlbTAwLTAwMDA
tMDAwMC0wMDAwLXdlYlVybFBhcnNlciIsInJlc291cmNlX2NoYXRfaWQiOm51bGx9.cz
1eeZEZdaQH5CGUaxwUmfeJfqTOZMoh3PbosHslSPA](https://cdn.qwenlm.ai/qwen_url_parse_to_markdown/system00-0000-0000-0000-webUrlParser?key=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJyZXNvdXJjZV91c2VyX2lkIjoicXdlbl91cmxfcGFyc2VfdG9fbWFya2Rvd24iLCJyZXNvdXJjZV9pZCI6InN5c3RlbTAwLTAwMDA
tMDAwMC0wMDAwLXdlYlVybFBhcnNlciIsInJlc291cmNlX2NoYXRfaWQiOm51bGx9.cz
1eeZEZdaQH5CGUaxwUmfeJfqTOZMoh3PbosHslSPA)
8. [https://cdn.qwenlm.ai/qwen_url_parse_to_markdown/system00-0000-0000-0000-webUrlParser?
key=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJyZXNvdXJjZV91c2VyX2lkIjoicXdlbl91cmxfcGFyc2VfdG9fbWFya2Rvd24iLCJyZXNvdXJjZV9pZCI6InN5c3RlbTAwLTAwMDA
tMDAwMC0wMDAwLXdlYlVybFBhcnNlciIsInJlc291cmNlX2NoYXRfaWQiOm51bGx9.cz
1eeZEZdaQH5CGUaxwUmfeJfqTOZMoh3PbosHslSPA](https://cdn.qwenlm.ai/qwen_url_parse_to_markdown/system00-0000-0000-0000-webUrlParser?key=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJyZXNvdXJjZV91c2VyX2lkIjoicXdlbl91cmxfcGFyc2VfdG9fbWFya2Rvd24iLCJyZXNvdXJjZV9pZCI6InN5c3RlbTAwLTAwMDA
tMDAwMC0wMDAwLXdlYlVybFBhcnNlciIsInJlc291cmNlX2NoYXRfaWQiOm51bGx9.cz
1eeZEZdaQH5CGUaxwUmfeJfqTOZMoh3PbosHslSPA)
9. [https://cdn.qwenlm.ai/qwen_url_parse_to_markdown/system00-0000-0000-0000-webUrlParser?
key=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJyZXNvdXJjZV91c2VyX2lkIjoicXdlbl91cmxfcGFyc2VfdG9fbWFya2Rvd24iLCJyZXNvdXJjZV9pZCI6InN5c3RlbTAwLTAwMDA
tMDAwMC0wMDAwLXdlYlVybFBhcnNlciIsInJlc291cmNlX2NoYXRfaWQiOm51bGx9.cz
1eeZEZdaQH5CGUaxwUmfeJfqTOZMoh3PbosHslSPA](https://cdn.qwenlm.ai/qwen_url_parse_to_markdown/system00-0000-0000-0000-webUrlParser?key=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJyZXNvdXJjZV91c2VyX2lkIjoicXdlbl91cmxfcGFyc2VfdG9fbWFya2Rvd24iLCJyZXNvdXJjZV9pZCI6InN5c3RlbTAwLTAwMDA
tMDAwMC0wMDAwLXdlYlVybFBhcnNlciIsInJlc291cmNlX2NoYXRfaWQiOm51bGx9.cz
1eeZEZdaQH5CGUaxwUmfeJfqTOZMoh3PbosHslSPA)
10. [https://cdn.qwenlm.ai/qwen_url_parse_to_markdown/system00-0000-0000-0000-webUrlParser?
key=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJyZXNvdXJjZV91c2VyX2lkIjoicXdlbl91cmxfcGFyc2VfdG9fbWFya2Rvd24iLCJyZXNvdXJjZV9pZCI6InN5c3RlbTAwLTAwMDA
tMDAwMC0wMDAwLXdlYlVybFBhcnNlciIsInJlc291cmNlX2NoYXRfaWQiOm51bGx9.cz
1eeZEZdaQH5CGUaxwUmfeJfqTOZMoh3PbosHslSPA](https://cdn.qwenlm.ai/qwen_url_parse_to_markdown/system00-0000-0000-0000-webUrlParser?key=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJyZXNvdXJjZV91c2VyX2lkIjoicXdlbl91cmxfcGFyc2VfdG9fbWFya2Rvd24iLCJyZXNvdXJjZV9pZCI6InN5c3RlbTAwLTAwMDA
tMDAwMC0wMDAwLXdlYlVybFBhcnNlciIsInJlc291cmNlX2NoYXRfaWQiOm51bGx9.cz
1eeZEZdaQH5CGUaxwUmfeJfqTOZMoh3PbosHslSPA)
11. ONRAMP-AI-VRAR: an operational protocol for ethics and ... [https://
pmc.ncbi.nlm.nih.gov/articles/PMC12696944/](https://pmc.ncbi.nlm.nih.gov/articles/PMC12696944/)
12. Augmented accountability: Data access in the metaverse [https://
www.sciencedirect.com/science/article/pii/S2212473X25000689](https://www.sciencedirect.com/science/article/pii/S2212473X25000689)
13. Advancing Brain-Computer Interface Closed-Loop Systems for ... [https://
pmc.ncbi.nlm.nih.gov/articles/PMC12588595/](https://pmc.ncbi.nlm.nih.gov/articles/PMC12588595/)

14. Meta-Review on Brain-Computer Interface (BCI) in the ... <https://dl.acm.org/doi/10.1145/3696109>
15. Multi-modal multi-task federated foundation models for next ... <https://www.nature.com/articles/s44459-025-00003-0>
16. Tri-manual interaction in hybrid BCI-VR systems <https://www.frontiersin.org/journals/neurorobotics/articles/10.3389/fnbot.2025.1628968/full>
17. IEEE 1857.11-2024 <https://standards.ieee.org/ieee/1857.11/10691/>
18. priors-augmented retrieval for LLM-based metadata generation <https://academic.oup.com/bioinformatics/article/41/10/btaf519/8257680>
19. A Comprehensive Review of Multimodal XR Applications ... <https://www.mdpi.com/2414-4088/8/11/98>
20. Blaze: Compiling JSON Schema for 10× Faster Validation <https://arxiv.org/html/2503.02770v1>
21. A Systematic Analysis of Deceptive Design in Extended ... <https://dl.acm.org/doi/10.1145/3659945?ref=hcigames.com>
22. Why AI Hallucinates: The Shallow Semantics Problem <https://www.linkedin.com/pulse/why-ai-hallucinates-shallow-semantics-problem-j-bittner-rl2qe>
23. XR4DRAMA a knowledge-based system for disaster ... <https://www.cambridge.org/core/product/5478E86E44513D2A211343D5288F569C/core-reader>
24. AI Governance in the Context of the EU AI Act (Extended ... <https://ojs.aaai.org/index.php/AIES/article/view/36642>
25. Mapping Data Governance Requirements Between the ... <https://openreview.net/forum?id=5ilvDyTkbg>
26. My views on the path to Neural Data regulations and ... <https://www.linkedin.com/pulse/my-views-path-neural-data-regulations-implications-grc-chika-o--kdooe>
27. Doing Audits Right? The Role of Sampling and Legal ... <https://arxiv.org/html/2505.03601v1>
28. Risk, regulation, and governance: evaluating artificial ... <https://link.springer.com/article/10.1057/s41284-025-00495-z>
29. What are the international differences in data encryption ... <https://www.tencentcloud.com/techpedia/123375>
30. IEEE SA - 26 September 2024 <https://standards.ieee.org/about/sasb/sba/26sep2024/>
31. IEEE SA - Sitemap <https://standards.ieee.org/sitemap/>
32. MIND Act of 2025: Protecting Brain Privacy with Federal ... https://www.linkedin.com/posts/nitafarahany_as-neurotechnology-outpaces-existing-protections-activity-7376656776605380608-0GMT

33. Regulating neural data processing in the age of BCIs <https://pmc.ncbi.nlm.nih.gov/articles/PMC11951885/>
34. Regulating the Mind: Neuromarketing, Neural Data and ... <https://www.mdpi.com/2076-3387/15/10/386>
35. ENISA: 5G design and architecture of global mobile networks <https://pmc.ncbi.nlm.nih.gov/articles/PMC10446063/>
36. The emergence of EU cybersecurity law: A tale of lemons, ... <https://www.sciencedirect.com/science/article/pii/S0267364924001377>
37. Reimagining cultural heritage conservation through VR ... <https://pmc.ncbi.nlm.nih.gov/articles/PMC12582480/>
38. Evaluating Learning Outcomes of Virtual Reality ... <https://dl.acm.org/doi/full/10.1145/3593432>
39. Meta-Review on Brain-Computer Interface (BCI) in the ... <https://dl.acm.org/doi/abs/10.1145/3696109>
40. Assessing the (severity of) impacts on fundamental rights <https://www.sciencedirect.com/science/article/pii/S0267364925000081>
41. Evaluating Learning Outcomes of Virtual Reality ... <https://dl.acm.org/doi/10.1145/3593432>
42. Deepfake detection in generative AI: A legal framework ... <https://www.sciencedirect.com/science/article/pii/S2212473X25000355>