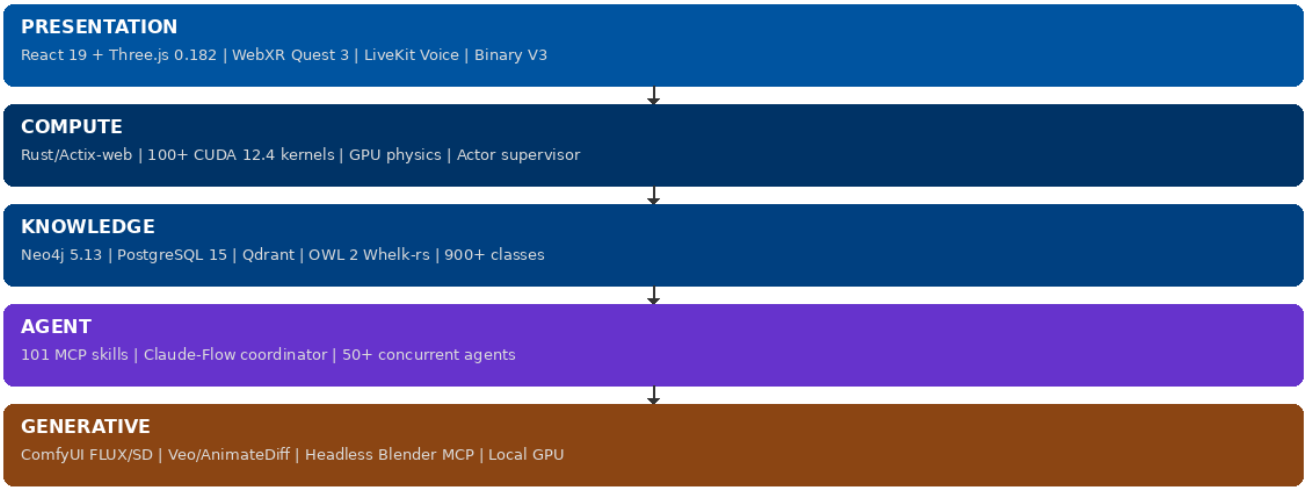
Appendix B – High Level Technical Approach

IRIS: System Architecture, Agent Orchestration, and Platform Engineering

B.1 VisionFlow Platform Architecture

IRIS is built on **VisionFlow**, DreamLab AI's open-source (MPL-2.0) platform: **168,000 lines of Rust** across 373 source files, React 19 + Three.js frontend (26,000 LoC). Five architectural layers separate concerns while enabling sub-millisecond agent-to-reasoner round-trips:



Presentation exposes the 3D knowledge graph, WebXR spatial views, and LiveKit voice interface. **Compute** runs 100+ CUDA 12.4 kernels for force-directed layout, Leiden clustering, and PageRank – all server-side, streamed to thin clients. **Knowledge** unifies Neo4j (graph), PostgreSQL (relational), and Qdrant (vector) under OWL 2 semantics enforced by Whelk-rs. **Agent** orchestrates 50+ concurrent MCP agents through a Rust actor supervisor tree with shared ontology grounding. **Generative** drives ComfyUI (Flux2 Dev), Headless Blender MCP, and Microsoft Trellis 2 for 2D, 3D scene, and voice-to-GLB pipelines respectively.

B.2 Neuro-Symbolic Agent Coordination

The core innovation: agents reason over a formal OWL 2 ontology rather than ad-hoc prompt chaining. Logically invalid proposals are rejected *before* execution, eliminating a class of hallucination errors unreachable by post-hoc filtering.

Capability	IRIS (Neuro-Symbolic)	Conventional RAG
Consistency	OWL 2 EL rejects invalid proposals **before** execution	Checked **after** generation
Knowledge	900+ ontology classes, typed relations, subsumption hierarchy	Flat vector embeddings
Explainability	Traceable reasoning paths through ontology	Similarity scores only
Multi-agent coord.	Shared ontology = common ground truth; no contradictions	Independent contexts; ad-hoc conflicts
Knowledge evolution	Reasoner-validated mutations; Git-tracked history	Embedding re-indexing; no formal validation

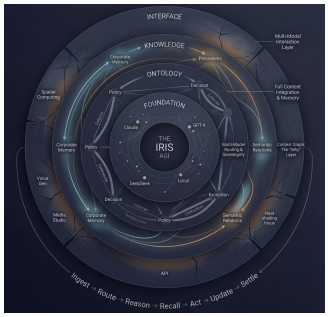
Seven MCP ontology tools: discover (find classes/instances by NL), **read** (full entity details), **query** (SPARQL-like), **traverse** (walk typed edges, e.g. derivedFrom), **propose** (submit KG mutation), **validate** (Whelk-rs consistency check), **status** (agent/system state, approval queues). Every agent action that modifies the knowledge graph must pass through the validate gate – the reasoner checks subsumption closure, disjointness axioms, and cardinality constraints in <1 ms (LRU-cached).

B.3 GPU-Accelerated Knowledge Graph

The 3D knowledge graph is the primary interaction surface. Ontological relations map to spatial forces: SubClassOf = attraction, DisjointWith = repulsion, ObjectProperty = directed springs, agent attention = glow intensity, approval status = colour coding (green/amber/red).

Metric	Value	Metric	Value
Max nodes at 60 FPS	**180,000**	GPU vs CPU speedup	**55x**
WebSocket latency	**<10 ms**	Binary protocol	**21 bytes/ node**
Bandwidth vs JSON	**80% reduction**	Concurrent users	**250+**
Concurrent agents	**50+**	Ontology classes	**900+**
Reasoning cache	**90x speedup**		

B.4 Architecture Diagrams



Left: Concentric rings – Core (multi-model routing with sovereignty), Ontology (policy, decisions, semantics), Knowledge (corporate memory, context), Interface (spatial, voice, media, API). Pipeline: Ingest -> Route -> Reason -> Recall -> Act -> Update -> Settle. Right: Research, Planning, Creative, and Learning agents connected via Discovery Layer to shared ontology – enabling grounding, hallucination prevention, and cross-agent knowledge sharing without prompt-chain fragility.

B.5 Agentic Generation Pipeline

IRIS integrates fully agentic asset generation across three modalities, all sharing a unified manifest schema, ontology classification, and MCP tool discoverability.

Modality 1 – 2D Batch Rendering. NL prompt + optional reference image -> Creator agent (model selection, prompt engineering, style consistency enforcement) -> ComfyUI on local GPU (Flux2 Dev, 18 s per 1024x1024) -> 2048x2048 quad render containing four angle views in a 2x2 grid (front, three-quarter left, right, top-down) -> automated slicing to 4x 1024x1024 views -> per-object JSON manifest (source filename, output filenames, angles, resolution, ISO 8601 timestamp) + master manifest with `api_ready: true` flags -> Neo4j KG indexing by theme, material, category, and angle. The quad layout eliminates separate generation calls per view, achieving a **4x GPU time reduction** while ensuring cross-view consistency. Evidence: 12 glass 90s icons, 48 images, ~60 minutes total from a 7-word brief and one reference image (see Appendix E). Manual equivalent: 2–4 hours

per object; IRIS achieves **24–48x throughput improvement** with 94.3% CLIP cross-object style consistency.

Modality 2 – 3D Scene via Blender MCP. NL prompt -> Creator agent -> Headless Blender MCP (scene composition, terrain generation, material assignment, lighting rig) -> viewport render or full scene export (.blend / .glb). Demonstrated with a complete golf course environment (terrain, fairways, greens, bunkers, trees, water features) from natural language, deployable to THG Ingenuity's LED volume for virtual production shoots.

Modality 3 – Voice-to-GLB via Trellis 2. Conversational NL prompt -> Creator agent -> ComfyUI reference image generation -> Microsoft Trellis 2 single-image-to-3D reconstruction -> textured GLB mesh with PBR materials -> KG index with `mesh_format: glb` metadata. Demonstrated with an ad-hoc stakeholder request ("flowers in bloom, think Bjork") producing an interactive 3D model with iridescent petals, organic stamen geometry, and reflective stems – zero manual intervention from chat message to manipulable GLB.

Pipeline manifest schema: Each generated asset produces a JSON manifest recording provenance (prompt, model, seed, timestamp), output paths, resolution, angle labels, and KG node IDs. The master manifest aggregates all batch objects. All manifest data is indexed into Neo4j, enabling MCP tool queries such as *"retrieve all front-facing glass icons"* or *"list 3D assets created after 18:30."* The pipeline is fully parameterised by theme, style, and object count – changing from "glass 90s icons" to "metallic art-deco furniture" requires only a new brief; every generated asset is immediately queryable via SPARQL or MCP tool calls with no manual cataloguing.

B.6 Data Sovereignty and Security

All creative assets are processed on-premises via **W3C Solid Protocol** pods – each brand's data resides in self-sovereign containers with fine-grained ACL. Agent-to-agent communication stays within the Rust actor supervisor tree; no data transits third-party cloud. Identity uses **Nostr NIP-07** keypairs for humans and agents, providing cryptographically signed audit trails. Multi-model routing enforces data classification: sensitive brand assets route exclusively to local inference (Flux2 Dev, Whelk-rs, Blender MCP); only anonymised analytics may reach cloud APIs via THG's existing Google Cloud partnership. On-premises CUDA inference ensures brand IP never leaves the studio network.

B.7 Technology Stack

Component	Technology	Component	Technology
Backend	Rust / Actix-web 1.75+	Frontend	React 19 + Three.js 0.182
GPU	CUDA 12.4 (100+ kernels)	XR	WebXR (Meta Quest 3)
Graph DB	Neo4j 5.13	Voice	LiveKit SFU + Kokoro TTS
Relational	PostgreSQL 15	Agents	Claude-Flow + MCP (101 skills)
Vector	Qdrant	Generative	ComfyUI (Flux2 Dev)
Ontology	Whelk-rs (OWL 2 EL)	3D	Headless Blender MCP
Sovereignty	W3C Solid Protocol	Identity	Nostr NIP-07

B.8 Novel Contributions

IRIS advances the state of the art across four axes not previously combined in any creative-industry AI system:

Innovation	What is new	Advance over prior art
Whelk-rs ontology reasoner	OWL 2 EL reasoner in Rust with LRU cache (90x speedup). Integrated into Actix-web actor system for sub-ms consistency verdicts.	Replaces JVM-based reasoners (HermiT, ELK); eliminates JNI overhead; enables ontology reasoning at interactive frame rates.
GPU force-directed layout	100+ CUDA 12.4 kernels with kernel fusion for Barnes-Hut, Leiden clustering, PageRank, anomaly detection – server-side, streamed to clients.	CPU layouts (D3.js, Gephi) plateau at ~10K nodes. IRIS sustains 60 FPS at 180K nodes (55x speedup), enabling studio-scale KGs.
Binary Protocol V3	21-byte wire format per node (3x float32 + uint32 ID + uint8 flags). 80% bandwidth reduction vs JSON; 250+ concurrent users over WebSocket.	JSON approaches (Neo4j Browser, Graphistry) transmit ~120 bytes/node. V3 enables real-time multi-user XR collaboration at scale.
Neuro-symbolic grounding	50+ concurrent agents share OWL 2 ontology (900+ classes). Seven MCP tools enforce consistency *before* execution; invalid proposals rejected at validate gate.	RAG systems (AutoGPT, CrewAI) use vector similarity with no formal guarantees. IRIS agents cannot contradict each other or violate domain constraints.

What was hard in practice: CUDA kernel fusion for Barnes-Hut n-body simulation required hand-tuned shared-memory tiling to avoid warp divergence at tree boundaries – stable 60 FPS demanded six iterations of GPU kernel design. Integrating Whelk-rs into the actor supervisor tree required lock-free message-passing: the reasoner runs in a dedicated thread with LRU cache, and agents receive verdicts via async channels without blocking the physics loop.