

# Building a Browser-Based Astronomical Visualization System: Technical Feasibility Report

A SpaceEngine-class universe viewer running entirely in WebGPU is technically achievable today. This report synthesizes cutting-edge research across five critical domains—neural compression, spatial indexing, uncertainty representation, catalog fusion, and browser streaming—revealing that **all core technologies have reached production-ready maturity**, though integration at astronomical scale presents significant engineering challenges.

## Executive summary for decision-makers

The convergence of three technological breakthroughs makes this project viable now: **3D Gaussian Splatting compression** achieving 100× data reduction with minimal quality loss, **WebGPU compute shaders** rendering 2 billion points interactively in browsers, ([kitware](#)) and **HEALPix-based streaming** proven at scale by the Gaia mission's 1.8 billion star catalog. The highest technical risks involve cross-browser WebGPU support (Firefox still experimental) and mobile GPU limitations rather than fundamental feasibility. Estimated development timeline is **8-12 months** to production-ready system, with data preparation pipelines requiring **6-9 months** of parallel development for multi-catalog fusion.

---

## Neural compression enables efficient universe streaming

The state of the art in 3D Gaussian Splatting compression has advanced dramatically since 2023, with **HAC++ achieving 100× compression** versus vanilla 3DGs while actually improving visual quality. This breakthrough from TPAMI 2025 uses hash-grid assisted context modeling with learned entropy coding, compressing typical scenes from 734 MB to just **8.7 MB** at the low-rate setting.

For browser deployment, vector quantization methods offer the optimal tradeoff between compression ratio and decompression speed. **Compact3D** (ECCV 2024) achieves 45× compression using K-means clustering during training, storing small codebooks that enable O(1) GPU lookups at runtime. ([Ucdvision](#)) LightGaussian from NeurIPS 2024 takes a complementary approach, pruning 60% of Gaussians through global significance scoring while distilling spherical harmonics to lower degrees.

Method	Compression	Quality (PSNR)	Browser Feasible
HAC++	<b>100×</b>	27.60-27.82 dB	Partial (CPU decode)
Compact3D	45×	Near-baseline	<b>Yes</b> (GPU lookup)
LightGaussian	15×	Near-baseline	<b>Yes</b> (simple decode)
MPEG G-PCC	10-20×	Variable	No (no browser codec)

WebGPU implementations already exist. **Niedermayr et al.'s Compressed 3DGS** runs in browsers using Rust/WebGPU, achieving 4× higher framerates than compute-only pipelines through hardware rasterization. The Shanghai AI Lab's Visionary engine achieves **2-16ms per frame** on RTX 4090, demonstrating that real-time performance is achievable.

The critical limitation is entropy decoding. While vector quantization decompresses trivially (codebook lookup), advanced entropy-coded methods like HAC++ require CPU-side arithmetic decoding at ~100-500ms for million-scale scenes. For streaming, this suggests a hybrid approach: use aggressive offline compression with simple VQ for browser-side decompression, targeting **20-50× compression** rather than maximum ratios.

---

## HEALPix and octrees provide proven spatial indexing at scale

Indexing billions of astronomical objects requires hierarchical spatial structures that support both 2D sky queries and 3D distance-aware visualization. ([Wallaby-survey](#)) **HEALPix** (Hierarchical Equal Area isoLatitude Pixelisation) has become the de facto standard for astronomical data, underpinning Gaia, Planck, LIGO/Virgo, and the IVOA's HiPS streaming protocol.

HEALPix divides the celestial sphere into 12 base-resolution curvilinear quadrilateral pixels, with each level subdividing into 4 children. At NSIDE=8192, this yields ~800 million pixels with sub-milliarcssecond precision. The NESTED numbering scheme preserves spatial locality, making it ideal for database indexing and progressive streaming.

Gaia DR3 exemplifies production-scale implementation: **source\_id encodes HEALPix level-12 index** in bits 36-63, partitioning 1.8 billion stars into ~3,386 level-8 ranges of ~500,000 sources each. This enables efficient spatial queries through cell intersection and consistent partitioning across all tables.

For 3D visualization with distances, **octrees** complement HEALPix. The Potree system has proven that browsers can render **597 billion points** using octree-based streaming with screen-space error metrics. GPU-accelerated octree construction achieves 1-4 billion points per second, with memory-efficient encoding storing 1 billion points in just 8GB.

The recommended architecture uses **HEALPix for 2D sky tessellation** (leveraging HiPS infrastructure) combined with **octrees for 3D distance-aware LOD**:

### Hybrid Index Structure:

```
|-- healpixIndex: HEALPixTree // For 2D cone searches, catalog streaming
  |-- octreeIndex: Octree    // For 3D navigation, distance-based LOD
```

Performance studies comparing HTM versus HEALPix at billion-object scale show that **hardware I/O throughput dominates over indexing algorithm choice**—focus engineering effort on CDN architecture and compression rather than optimizing index algorithms.

## Uncertainty must drive level-of-detail decisions

Astronomical measurements carry inherent uncertainty that varies dramatically with brightness and distance. Gaia DR3 parallax errors range from **0.02-0.03 mas** for bright stars ( $G<15$ ) to **1.3 mas** for faint sources ( $G=21$ ), meaning fractional distance errors exceed 20% for approximately 80% of the catalog. This uncertainty must inform visualization decisions.

The fundamental challenge is that parallax errors do **not** propagate linearly to distance errors. The inverse relationship  $d = 1/\omega$  creates systematic **Lutz-Kelker bias**: random parallax errors preferentially scatter toward larger distances because more volume exists at greater distances. Simple parallax inversion produces biased distance estimates for most stars.

**Bailer-Jones probabilistic distance estimation** addresses this through Bayesian inference, combining parallax likelihoods with direction-dependent Galaxy model priors. The resulting posterior distributions yield asymmetric confidence intervals ( $r_{\text{lo}}, r_{\text{med}}, r_{\text{hi}}$ ) that properly capture distance uncertainty. These estimates are pre-computed for Gaia DR3 and should be used directly.

Gaia provides a full **5×5 covariance matrix** for astrometric parameters ( $\alpha, \delta, \omega, \mu\alpha^*, \mu\delta$ ) through 10 correlation coefficients. Propagating this to 3D Cartesian coordinates via Jacobian transformation yields an **uncertainty ellipsoid** whose orientation and semi-axes represent positional confidence.

For LOD systems, we propose a **combined importance score**:

$$\text{Importance} = w_1 \times \text{Distance\_score} + w_2 \times \text{Quality\_score} + w_3 \times \text{Brightness\_score}$$

where:

$$\begin{aligned}\text{Distance\_score} &= f(1/d_{\text{screen}}) \quad // \text{Closer} = \text{more important} \\ \text{Quality\_score} &= f(\omega/\sigma_{\omega}) \quad // \text{Well-measured} = \text{more important} \\ \text{Brightness\_score} &= f(20 - G_{\text{mag}})/20 \quad // \text{Brighter} = \text{more important}\end{aligned}$$

Objects with high quality scores ( $\omega/\sigma_{\omega} > 5$ ) receive full geometric detail including uncertainty ellipsoids; uncertain objects ( $\omega/\sigma_{\omega} < 2$ ) display as blurred points or aggregate into density fields. This allocates rendering budget to data that justifies the precision.

Uncertainty visualization research from IEEE VIS provides established techniques: color-opacity mapping (uncertain → transparent), size variation (larger = more uncertain), blur/texture effects, and animated sampling from posterior distributions. **Li et al.'s 2007 TVCG paper** specifically addresses astrophysical data, proposing ellipsoid models with unified color-coding for log-scale distances.

## Mature tools enable multi-catalog fusion

Combining Gaia, 2MASS, WISE, and planetary ephemerides into a unified schema requires cross-matching algorithms that handle positional uncertainty, epoch differences, and heterogeneous photometry. The toolchain is mature and well-documented.

**Positional cross-matching** has evolved from simple nearest-neighbor to probabilistic frameworks. The foundational **Budavári & Szalay (2008)** Bayesian approach computes Bayes Factors comparing the hypothesis that sources originate from the same object versus chance alignment. For Gaussian positional errors, the astrometric Bayes Factor is analytically computable. The **NWAY algorithm** (Salvato et al. 2018) extends this to N-way matching across arbitrary catalog combinations, achieving 94-99% reliability. (ADS)

The CDS X-Match service handles billions of sources using HEALPix indexing (u-strasbg) and Apache Spark, with support for per-row positional errors. **STILTS/TOPCAT**, developed by Mark Taylor over 20+ years, provides comprehensive local processing with epoch propagation via the (epochProp) function. Astropy's coordinate matching uses SciPy k-d trees for efficient nearest-neighbor searches.

**Epoch differences** pose a critical challenge. Gaia DR3 uses J2016.0, while 2MASS observations span J1999-2001. High proper motion stars can move >1 arcsec between epochs. Solutions include:

- STILTS (epochProp) function for Gaia position propagation
- CDS provides Gaia in both J2016.0 and epoch-precessed J2000.0 versions
- Gaia Archive pre-computed cross-matches include proper motion correction

Error propagation when combining measurements follows standard formulas. (Number Analytics) For positional matching with circular errors:  $\sigma_{\text{total}} = \sqrt{(\sigma_1^2 + \sigma_2^2)}$ . Epoch propagation adds proper motion uncertainty:  $\sigma_{\alpha(t)} = \sqrt{(\sigma_{\alpha_0}^2 + (t-t_0)^2 \times \sigma_{\mu\alpha}^2)}$ . Systematic effects include Gaia parallax zero-point ( $\sim 17 \mu\text{as}$ ) and proper motion bias for bright stars.

**Schema design** should follow IVOA standards using VOTable format with UCDs (Unified Content Descriptors) for semantic column identification. Preserve original catalog identifiers, match separation and probability, all photometric bands with errors, and epoch information.

For **solar system objects**, JPL Horizons provides authoritative ephemerides for 1.4M+ asteroids, 4000+ comets, and all planets. (NASA Jet Propulsion Laboratory) Integration requires time-resolved queries: for each observation epoch, query Horizons for predicted positions, then flag as distinct source type in the unified schema.

Tool	Scale	Probabilistic	N-way	Best For
CDS X-Match	Billions	Limited	No	Large catalog pairs (remote)
STILTS/TOPCAT	Millions	No	No	Local processing, custom criteria
NWAY	Large	<b>Yes</b>	<b>Yes</b>	Multi-catalog with priors
Astropy	Moderate	No	No	Programmatic integration

## WebGPU streaming architecture achieves billion-point interactivity

Browser-based streaming of massive point clouds has been proven at scale. **Potree** renders 597 billion points in web browsers using octree-based LOD streaming, and VTK's WebGPU implementation demonstrates **2 billion points interactively** on a single workstation. [\(kitware\)](#)

The Potree 2.0 format uses just three files: metadata.json (structure), hierarchy.bin (octree layout), and octree.bin (all point data with Brotli compression). This 10-50× reduction in file count versus v1.7 dramatically accelerates conversion and reduces HTTP overhead.

**3D Tiles** (OGC Community Standard since 2019) provides a complementary approach for heterogeneous geospatial data. Version 1.1 supports point clouds via glTF with EXT\_mesh\_features for metadata, using geometric error metrics for LOD selection. The screen-space error calculation determines refinement:  $SSE = \text{geometricError} \times \text{screenHeight} / (\text{distance} \times 2 \times \tan(\text{fov}/2))$ .

WebGPU compute shaders enable **10× performance gains** over traditional GL\_POINTS rendering for large point clouds. The technique uses atomic operations (atomicMin with interleaved depth+color) for visibility determination, [\(kitware\)](#) achieving:

Points	OpenGL (ms)	WebGPU (ms)	Speedup
300M	125.7	14.7	<b>8.5×</b>
700M	275.1	27.0	<b>10.2×</b>
1.4B	N/A	59.6	—

**Bandwidth optimization** relies on hierarchical caching and view-dependent streaming. Compressed astronomical data requires 2-4 GB for static billion-point datasets (LAZ compression). Interactive streaming demands 10-100 Mbps depending on navigation speed. HTTP/2 multiplexing reduces latency by 16-71%.

Browser caching uses **Service Worker + IndexedDB** architecture. The Cache API stores HTTP tile responses (50-500MB Safari, 80% disk Chrome), while IndexedDB persists processed geometry and metadata. LRU eviction manages quota constraints.

**Geometry streaming versus pixel streaming** strongly favors geometry for this application:

- Offline capability essential for field astronomy
- Low latency critical for interactive exploration
- CDN scaling is 10,000× cheaper than GPU server rental
- Static astronomical data is highly cacheable

Pixel streaming only makes sense for security-sensitive data or devices without WebGPU support.

---

## Technical risk assessment and maturity levels

Component	TRL	Risk Level	Key Concern
3DGS Rendering	8	Low	Production-ready in WebGPU
VQ Compression	7	Low	Simple GPU decode
Entropy Compression	6	Medium	CPU decode bottleneck
HEALPix Indexing	9	Low	Proven at Gaia scale
Octree Streaming	9	Low	Potree proven at 600B points
Uncertainty LOD	5	Medium	Novel integration required
Cross-Catalog Fusion	8	Low	Mature tools available
WebGPU Browser Support	7	Medium	Firefox still experimental
Mobile Performance	6	<b>High</b>	GPU/memory constraints

**Highest risks** center on browser compatibility (Safari limited, Firefox developmental) and mobile GPU limitations. Mitigation requires WebGL fallback paths and adaptive quality settings.

## Development timeline estimate:

- Phase 1 (2-3 months): WebGPU compute shader renderer with static dataset

- Phase 2 (2-3 months): Tile streaming with LOD and HEALPix integration
- Phase 3 (1-2 months): Service worker caching, offline mode
- Phase 4 (2-3 months): View-dependent optimization, prefetching, uncertainty LOD
- Phase 5 (1-2 months): Mobile optimization, WebGL fallback

## **Data preparation pipeline** (parallel track, 6-9 months):

- Catalog acquisition and epoch normalization
  - Cross-matching via STILTS + NWAY
  - Octree generation via modified PotreeConverter
  - Compression and tile generation
  - Validation against source catalogs
- 

## **Recommendations for project execution**

The technical foundation supports a viable path to production. We recommend:

**Compression strategy:** Start with Compact3D-style vector quantization ( $45\times$  compression, trivial decode), reserving advanced entropy coding for server-side optimization. Target  $20\text{-}50\times$  compression with conservative pruning—astronomical features can be subtle.

**Indexing architecture:** Deploy HEALPix (NESTED) for 2D sky tessellation, leveraging existing HiPS infrastructure and CDS catalogs. Add octree indexing for 3D visualization with distances, using Potree-style streaming.

**Uncertainty integration:** Compute quality scores ( $\omega/\sigma_\omega$ ) and use Bailer-Jones distances. Allocate rendering detail proportional to measurement confidence. Reserve ellipsoid rendering for nearby, well-measured stars; use blur/transparency for uncertain objects.

**Streaming implementation:** Adopt Potree 2.0 format with custom WebGPU renderer using compute shader rasterization. Target 60fps at 100M visible points, with Service Worker caching for offline access.

**Catalog pipeline:** Use pre-computed Gaia cross-matches where available (2MASS, WISE), STILTS for custom regions with epoch propagation, and NWAY for probabilistic multi-catalog fusion. Preserve full provenance in unified schema.

The convergence of 3D Gaussian Splatting compression, WebGPU compute capabilities, and proven astronomical data infrastructure creates a genuine opportunity to build SpaceEngine-class visualization in the browser. The technology is ready; the challenge is engineering integration at scale.

