TECHNICAL CASE STUDY

Interactive Geospatial Intelligence Dashboard Build Handoff

Preserving entity identity while moving from city overview to building records

CLIENT	INDUSTRY	DURATION	TEAM
Independent portfolio	Spatial design	Not documented	Keith Trowbridge

At a Glance

3	1	4	2
Building profiles	Canonical entity ID	View states	Renderer modes

Figure 1. Documented scope, reference values and design targets; labels are not verified production results.

1. Executive Summary

This case study develops the source's concept of a selectable city map that transitions into a detailed building view with contextual records and an internal history archive. The intended audience is visitors to Keith Trowbridge's professional portfolio. The researched design uses a shared entity model, interruptible camera transitions, accessible DOM controls and a resilient rendering fallback. The later build specification provides the integration authority; earlier conversational code remains concept evidence. The source package documents requirements and seed data, not a verified completed application. [SRC; T1–T2; P1]

> KEY TAKEAWAY
>
> **Preserve one entity and one view context across every interaction.**

2. Background & Challenge

2.1 Context

The desired experience begins with named building symbols rather than an always-open detailed model. Selecting a symbol reveals architecture and metadata; opening history should deepen understanding without losing spatial position. The supplied later handoff defines a fictional city, SarDev Metropolitan, with three distinct architectural silhouettes and a local 1,200 m square grid. Ownership, addresses, geographic positions and histories are demonstration records, not surveyed property facts. [SRC]

Illustrative photograph. AI-generated conceptual setting; not an actual facility, client, product screenshot or delivered customer portrait.

The authoritative 6 October 2026 handoff defines an additional PHP portfolio route, scoped local assets, three building profiles, a 1,200 m local grid and a procedural miniature-city view. It supersedes earlier briefs. Existing portfolio/index.php and sitemap.xml are the anticipated baseline changes; shared styling, shell, media and existing demos should remain intact. The package is explicitly a specification and seed dataset, with implementation and deployment evidence still to be supplied.

Source precedence. The later handoff names the shared PHP functions render_site_background, render_site_header, render_site_footer and render_site_scripts. It specifies /portfolio/spatial-intelligence/ as the new route. The present documents describe that integration boundary; the website source itself is not included in this archive. Before implementation, compare the working repository with its authorized baseline and document any adaptation.

2.2 The Problem

Early drafts mix HTML Canvas, CSS 3D, Tailwind CDN styling and prohibitions on graphics libraries. They also risk treating a photograph as a navigable model, losing camera state when opening history, or using separate coordinates for symbols and architecture. The corrected specification resolves those conflicts with procedural Three.js geometry, a Canvas fallback, scoped native styles and one canonical dataset. The challenge is to produce credible depth while keeping controls and records functional when graphics fail. [SRC; T1–T2]

- **Identity drift: selected symbols, detailed models and records can disagree when separate data copies are used.**
- **Context loss: archive dialogs or interrupted camera travel can reset selection, position or focus.**
- **Runtime fragility: GPU failure and third-party dependencies can make a visually rich experience unusable.**

2.3 Goals & Constraints

Goal	Success Criteria	Constraint
Entity consistency	Same ID in map, model, metadata and history	Synthetic records clearly labeled
Predictable transitions	Latest selection wins without stale callbacks	Reduced motion skips travel
Resilient access	Selector and history work without WebGL	Interactive Canvas fallback
Website preservation	Reuse shared PHP shell and scoped assets	No global theme replacement

Table 1. Project objectives and constraints.

3. Solution Overview

The proposed application separates data, interaction state and rendering. A semantic selector offers the same entities as the canvas. The renderer visualizes the currently selected record without owning the source of truth. The archive stores view and focus context, suspends input, and restores that context when closed. Local compiled assets permit ordinary PHP hosting without a persistent Node service, paid map API or runtime account requirement. [SRC; T1–T2; P1]

Visual construction. Build the slender glass tower, stepped Art Deco crown and courtyard campus as distinct silhouettes with facade detail appropriate to each. Warm ivory stone, pale concrete, blue-gray glazing and restrained trees reproduce the reference’s architectural-model language without copying it as a backdrop. Keep target footprints clear of road, park and water geometry. Reuse or instance repeated windows and trees; avoid allocating one unique mesh per decorative element. The initial detail camera must show the full building crown and useful street context. Selection can use an outline or ground ring without recoloring the complete structure.

3.1 Approach

1. **Discover: inventory the existing site shell and source precedence; validate all three building records and local map geometry.**
2. **Design: define CITY, TRANSITION, DETAIL and ARCHIVE states; specify camera targets, focus behavior and reduced motion.**
3. **Build: create procedural architecture, shared ID selection, validated JSON loading and an independent DOM record panel.**
4. **Validate and launch: test repeated selection, interruption, mobile stacking, archive focus and WebGL loss; package only intentional runtime changes.**

3.2 Technology Stack

Layer	Technology	Rationale
Website shell	Existing PHP and scoped native CSS	Retain branding, navigation and background
Rendering	Locally bundled Three.js	Solid architecture and camera control [T1–T2]
Fallback	Interactive Canvas 2D plus semantic DOM	Maintain selection and record access
Data	Validated buildings.json and city-map.json	Canonical IDs and synthetic local geometry

Table 2. Technology stack and selection rationale.

4. Architecture & Implementation

4.1 System Architecture

A same-origin loader validates entity IDs, dimensions, positions, status types and history arrays before state initialization. The interaction controller owns the selected ID, mode, camera pose and archive snapshot. A renderer adapter receives the state and performs picking or drawing; metadata reads the same entity record. Selection through either the canvas or the DOM invokes one command. A failed renderer must not prevent the record panel from loading. [SRC; T1–T2]

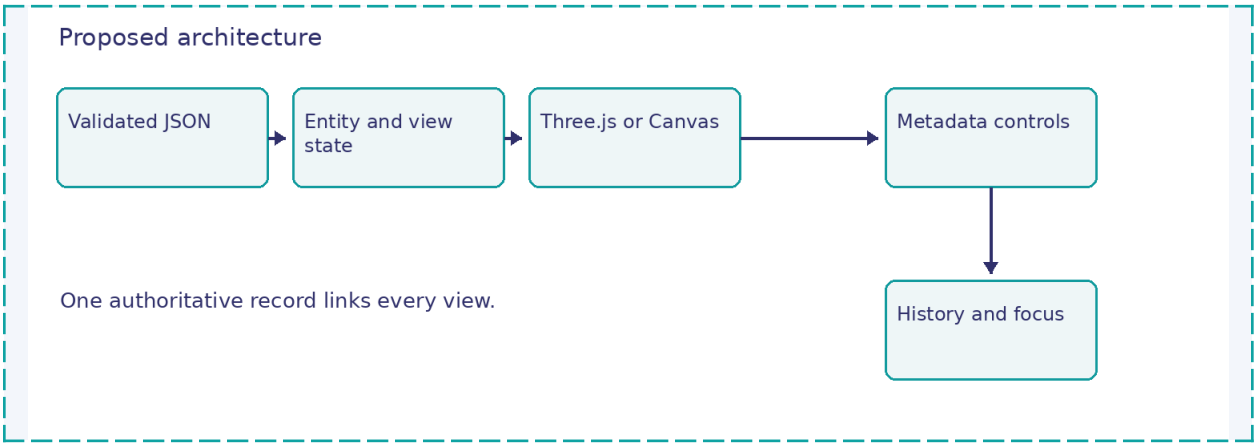

Figure 2. Proposed architecture and information flow; diagram is not deployment evidence.

State transitions. CITY has no selected building. Selection enters TRANSITION, then DETAIL when the current travel completes. Selecting another building during travel starts from the current pose and invalidates the older completion callback. ARCHIVE captures entity, camera, mode and focus context; closing restores the view without reinitializing geometry. Reduced motion completes selection directly. The one controller must own animation scheduling to avoid competing loops.

4.2 Key Technical Decisions

> **DECISION**
>
> **Choose local procedural Three.js geometry over a photographed city background or CSS-only illusion. The first choice provides real spatial depth and controlled camera behavior; the others can be attractive but do not satisfy navigable geometry. Preserve the existing PHP shell and compile dependencies outside the public runtime. Trade-off: geometry optimization, renderer lifecycle and fallback require deliberate engineering. [SRC; T1–T2]**

Local coordinates. Use easting and northing from the 1,200 m square grid as the source of location. The specified normalized overhead coordinates are x = easting/1200 and y = 1 − northing/1200 before camera transformation. The illustrative latitude/longitude origin is synthetic and is not a surveyed projection. A physical scale bar is appropriate only where it correctly represents the view. All owner, address, architecture and history fields remain clearly fictional.

4.3 Code or Configuration Snippet

```
// Illustrative identity-preserving selection
let transitionVersion = 0;
function selectBuilding(id) {
  const entity = records.get(id);
  if (!entity) return showRecordError();
  const version = ++transitionVersion;
  state.selectedId = id;
  updateMetadata(entity);
  animateFromCurrentPose(entity, () => {
    if (version !== transitionVersion) return;
    state.mode = "DETAIL";
  });
}
// DOM selector and canvas picking call this function.
```

Listing 1. Illustrative logic; integration, error recovery and tests remain implementation work.

4.4 Challenges Encountered

CHALLENGE AND RESOLUTION

Picking must account for canvas bounds, resizing and device pixel ratio; visible objects and interaction coordinates must stay aligned. Archive controls need an accessible name, focus containment, Escape handling and restoration to the opening control. Repeated selection must cancel superseded callbacks and avoid duplicate animation loops. Render on demand when idle and dispose resources on teardown so attractive facade detail does not become an avoidable performance burden. [SRC; T1–T2; W3]

Responsive and resilient behavior. The later specification suggests a 70–75% viewport with a 25–30% metadata area on wide desktop screens, stacking when either column becomes unreadable. Controls should wrap and the page should never require horizontal scrolling. A device pixel ratio cap around 1.5–2 is a starting optimization assumption, not a universal setting. ResizeObserver can update projection and picking after container changes. On WebGL context loss preserve entity and records, attempt controlled restoration and offer the simplified interactive view. Pause unnecessary work when hidden; do not rebuild the district on each frame. [SRC; T2]

5. Results & Impact

Evidence status. The source establishes the problem, requirements or described service. No supplied experiment log establishes production uptime, revenue uplift, measured user satisfaction or completed browser acceptance. The table distinguishes source observations from proposed corrections; quantified examples are not substituted for missing client measurements.

Metric	Before	After or Target	Evidence or Change
Entities	Early multiple coordinate forms	One validated canonical dataset	Acceptance requirement
Architecture	Concept/reference image	Three distinct procedural models	Build requirement
History	Context loss risk	Stored view and restored focus	Test requirement
Resilience	GPU dependence	DOM plus Canvas fallback	No test evidence supplied
Frame rate	Unverified promises	Document measured device results	No universal fps claim

Table 3. Source observations and proposed outcomes with evidence status.

Acceptance evidence. Capture initial city, each of the three detail models, history open, mobile layout and fallback mode. Test selection by pointer and keyboard, rapid selection during travel, Back to City during travel, repeated archive use, resize and device pixel ratio changes, loading errors and context loss. Run routes through a local PHP server; static file loading cannot verify the shell. Record Apache routing separately when staging is unavailable.

5.1 Business Impact

- **Efficiency: one data model reduces synchronization defects and makes record updates easier to maintain; no engineering-hour saving has been measured.**
- **Revenue and cost: local compiled assets avoid an always-running Node backend and external map API for this fictional demo; hosting costs remain unquantified.**
- **Experience: overview-to-detail travel helps communicate location and context, while reduced motion and DOM alternatives preserve access for different visitors.**

> **VALIDATION STATUS**
> No attributable client testimonial or independently verified outcome was supplied. The case study is complete as a researched analysis; implementation and commercial claims require the specified project evidence.

6. Lessons Learned

What Worked	What We Would Do Differently
Define source precedence early	Treat the later build specification as integration authority
Separate state from renderer	Keep metadata usable through graphics failure
Store view context explicitly	Test interruption and repeated archive use

Table 4. Retrospective highlights.

Release boundary. Package new runtime assets and intentional edits separately from editable source. Include a dependency lockfile, licenses, changed-files manifest, test report, deployment instructions and rollback instructions. Back up the two expected changed baseline files. A prepared release is not a live deployment; a public deployment claim requires verification of the actual route. A mobile browser viewport simulation also does not establish a physical touch-device test.

7. Next Steps

The roadmap below is a proposed progression, not a commitment or a report of completed releases. Prioritize evidence that resolves the greatest uncertainty before adding visual scope.

- Next 30–90 days: build the three seeded models, state controller, archive and fallback; record browser evidence against the supplied acceptance checks.
- Next six months: evaluate comprehension and navigation with portfolio visitors; refine geometry and rendering budgets from measured device behavior.
- Longer term: reuse the entity-inspection pattern for real datasets only after adding coordinate reference, licensing, privacy and provenance requirements.

- Detailed completion checklist. Verify JSON validity, PHP syntax, local asset loading and absence of new console errors. Capture the shared header, footer and video background before and after integration. Retain the existing heuristic dashboard and its embeds. Test all controls in city and detail modes, including bounded zoom, reset, back navigation, history open/close and focus restoration. Record browser version, viewport, input method and renderer mode for each observation. Package compiled runtime assets separately from editable sources and identify any checks unavailable in the current environment instead of marking them as passed.

Appendix

A. Glossary

Term	Definition
Canonical ID	Single stable key linking all representations of an entity.
Picking	Identifying the rendered object beneath a pointer.
Camera pose	Position, orientation and target of the current view.
DPR	Device pixel ratio affecting drawing resolution and input alignment.
Procedural geometry	Model structures created from parameters rather than remote assets.
Renderer fallback	Simpler interactive view used when the primary graphics path fails.

Table 5. Terms used in this document.

B. Sources and Research Notes

[SRC] Primary source: Case Studies/Geospatial AI Heuristic Loss Dashboard/Codex-Geospatial-Dashboard-Handoff/Codex-Geospatial-Dashboard-Handoff/BUILD-SPECIFICATION.docx. Related documents in the supplied archive inform cross-project context. Research checked 7 October 2026. Additions are proposed designs or validation methods unless explicitly attributed to source evidence.

[T1] Three.js. Picking. https://threejs.org/manual/pages/picking.html

[T2] Three.js. Responsive Design. https://threejs.org/manual/pages/responsive.html

[P1] PHP. JSON Encoding. https://www.php.net/manual/en/function.json-encode.php

[W3] W3C. What Is New in WCAG 2.2. https://www.w3.org/WAI/standards-guidelines/wcag/new-in-22/

Method and limitations. Source assertions, representative simulations, illustrative graphics, researched implementation guidance and proposed acceptance targets are kept distinct. Research can support design choices but cannot establish missing historical client facts, actual delivery dates, measured results or testimonials.

C. About the Author

Keith's life has been shaped by curiosity, adventure, service, and a willingness to step into unfamiliar territory. He is married to his best friend and is the proud father of two Auburn University graduates, both thriving in their respective careers. For all the places he has been to, and the experiences he has gathered, his family remains the center of his life.

His interests span, archaeology and biblical research, skydiving, base jumping, and rock climbing, writing and drawing, and design and development. Across those pursuits runs a common thread: a desire to explore, understand, and create. He is as drawn to questions about the ancient world as he is to the possibilities taking shape in the next generation of technology.

Keith has traveled to every continent except Antarctica, lived in the United States, India, and Israel, and spent more than 20 years immersed in international cultures. Those experiences have deepened his appreciation for different perspectives and the importance of understanding people within the context of their lives. He served in the U.S. Army as a team leader with a special operations unit under Joint Special Operations Command (JSOC) and later supported global anti-human-trafficking organizations through field-based roles.

Today, Keith brings that breadth of experience to his work in AI strategy, design, and development. He sees the present as one of the most extraordinary periods of technological change in human history—and approaches it with the same curiosity that has carried him across the world. His focus is on turning emerging possibilities into useful tools, thoughtful experiences, and businesses that solve real problems. He remains an explorer at heart, with more to learn, more to build, and more to contribute.

Keith Trowbridge

AI Product Designer and AI Strategy Consultant

SarDev AI Labs

keith@keith-trowbridge.com

linkedin.com/in/keith-trowbridge

keith-trowbridge.com