TECHNICAL CASE STUDY

Heuristics Engine From Layout Friction to Review Evidence

Making interface trade offs inspectable through a simulated heuristic score

CLIENT	INDUSTRY	DURATION	TEAM
Independent portfolio	Design evaluation	Not documented	Keith Trowbridge

At a Glance

0.85	0.10	88.2%	4
Legacy example	Optimized example	Illustrative decline	Proposed factors

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

1. Executive Summary

This portfolio simulation makes the gap between visual polish and operational usability tangible by letting viewers adjust a clustering threshold and inspect a heuristic loss. The source contrasts a cluttered marker field with an aggregated view and supplies representative loss values around 0.85 and 0.10. The case study extends that concept into an auditable scoring design, with factor definitions, repeatable scenarios, exception safeguards and operator validation. A loss reduction is an illustrative change in a model score; it is not evidence of faster human cognition, a trained RLHF policy, or improved safety in a deployed system. [SRC; G3; A1]

> KEY TAKEAWAY
> **A useful proxy becomes credible when its definition and limits are visible.**

2. Background & Challenge

2.1 Context

The motivating environments include aerospace dispatch, utility monitoring and defense analysis, where timely exception recognition matters. No named operational client, asset count, validated weights, experiment log or production incident data is provided. The simulation serves portfolio visitors and design reviewers by explaining how rules can influence rendering. It should teach the distinction between a useful proxy and a validated measure rather than present its visual drama as a human-factors experiment. [SRC]

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

The short deep-dive document introduces Heuristic Loss and the AI Detection Threshold to design leaders and technical recruiters. It describes a high-loss cluttered state and a low-loss aggregated state, but does not supply the actual formula or executable engine. This case study completes the methodological design without pretending that missing code or human experiments already exist. Its proposed factor functions and calibration plan are explicitly additions.

Measurement scope. The loss is a proposed proxy for layout defects, not an established measure of cognitive load. The simulation should state its input factors and show their outputs on demand. Operator speed, confidence and error rate are separate observations. A source value above 0.85 or

below 0.10 identifies a modeled state; without a disclosed formula it cannot be independently reproduced.

2.2 The Problem

Aesthetic evaluation alone can favor satellite detail, glowing pins and large decorative components even when they reduce scan clarity. An unstructured critique is difficult to compare across revisions. A single loss value helps discussion, but also creates a risk of false precision if the factors and weights remain hidden. The design challenge is therefore both visual and methodological: make behavior interactive while showing exactly what the number represents. [SRC; A1]

- **Overlapping marker footprints can conceal count, geography and urgent status.**
- **A scalar score without factor definitions cannot be independently interpreted or reproduced.**
- **Aggressive aggregation may improve the displayed score while hiding a critical exception.**

2.3 Goals & Constraints

Goal	Success Criteria	Constraint
Explain trade offs	Display per-factor values alongside total loss	Proxy score must be labeled
Stable aggregation	Conserve all represented asset counts	Protected exceptions remain visible
Reproducible scenarios	Store seed, viewport, thresholds and factor versions	No random baseline drift
Calibrated interpretation	Compare score with operator task performance	Do not imply completed RLHF training

Table 1. Project objectives and constraints.

3. Solution Overview

A proposed scoring engine measures collision, occlusion, contrast failures and record compression, normalizes each factor and computes a bounded weighted loss. A clustering renderer responds to a radius control while preserving critical exceptions and accurate member counts. Each run records its dataset seed, zoom, viewport and configuration so reviewers can compare equivalent states. Human task evidence remains a separate layer that can later calibrate or challenge the proxy. [SRC; G3; A1]

Factor computation should expose denominators. Collision can change merely because a cluster reduces the number of eligible pairs, so report marker count and critical visibility beside the normalized factor. Occlusion sampling should identify what is essential and use a repeatable method. Contrast sampling must include actual label treatments and backgrounds. Compression should distinguish a deliberately dense readable table from clipped content. These definitions are proposed additions because the source does not supply its formula. An educational version can begin with simplified functions, provided it clearly identifies the simplification and stores raw factor outputs.

3.1 Approach

1. **Discover: define the decisions the simulation teaches and identify which source metrics are illustrative.**
2. **Design: specify factor units, normalization, weights, critical failures and protected exception rules.**
3. **Build: connect one threshold control to grouping, renderer and visible factor outputs; persist reproducible scenario settings.**
4. **Validate and launch: check count conservation and monotonic edge cases, then assess operator tasks with counterbalanced scenarios.**

3.2 Technology Stack

Layer	Technology	Rationale
Interface	HTML, CSS and JavaScript	Lightweight portfolio demonstration
Aggregation	Supercluster or equivalent documented method	Configurable screen-space proximity [G3]
Score	Versioned explicit heuristic functions	Transparent proxy; no learned model required
Evidence	Scenario JSON and event records	Repeatable comparison and later calibration

Table 2. Technology stack and selection rationale.

4. Architecture & Implementation

4.1 System Architecture

Synthetic assets enter a validated scenario loader. The threshold and viewport feed a clustering pass, while critical exceptions bypass suppression. The rendered layout is inspected by factor evaluators; a score aggregator displays the total and its components. Scenario and task evidence is logged separately. Feedback from operators can change weights only through a versioned calibration process, so an old result remains interpretable after the scoring design evolves. [SRC; G3; A1]

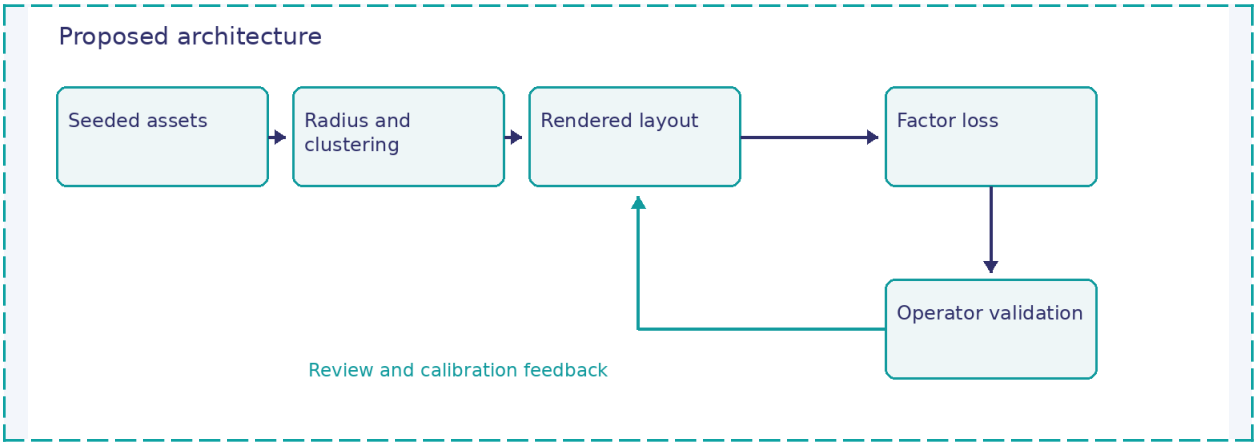

Figure 2. Proposed architecture and information flow; diagram is not deployment evidence.

Factor definitions. A proposed collision factor counts intersecting marker footprints divided by eligible marker pairs. An occlusion factor estimates the share of essential labels or boundaries obscured. A contrast factor records failing sampled label pairs rather than a badge chosen from mode. A compression factor records unreadable or clipped records. Normalize within a documented scenario range, keep raw values, and use a versioned factor schema so the total remains interpretable.

4.2 Key Technical Decisions

DECISION

Use an explicit heuristic before a learned reward model. This makes the first version explainable, inspectable and easier to debug. A trained model would require reliable labels, held-out evaluation and evidence that it generalizes. Keep accessibility and concealed critical assets as acceptance gates: a favorable weighted average must not compensate for a serious individual failure. Trade-off: handcrafted factors need calibration and may not capture the full complexity of a task. [A1; W1–W3]

Suggested weighting. Begin with explicit equal weights only as a demonstrator assumption; do not imply calibration. Report sensitivity when one weight changes, and use failure gates for concealed critical assets or accessibility defects. A lower total cannot compensate for a more dangerous exception. Keep the aggregation radius in screen pixels or clearly document any conversion to ground units. Supercluster's configurable radius illustrates the screen-space approach. [G3]

4.3 Code or Configuration Snippet

```
# Proposed scoring example; weights are not calibrated
def score(factors, weights):
    if abs(sum(weights.values()) - 1.0) > 1e-9:
        raise ValueError("weights must sum to one")
    loss = sum(weights[k] * factors[k] for k in weights)
    gates = factors["critical_hidden"] == 0
    return {"loss": max(0, min(1, loss)), "gates": gates}

# Store seed, viewport, radius and evaluator version.
# Factor values must use documented normalization.
# Human task speed is measured separately.
# A loss score is not a safety certification.
```

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

4.4 Challenges Encountered

CHALLENGE AND RESOLUTION

An expanding radius can reduce visible marker collisions without reducing workload, particularly when a cluster hides the urgent object a user needs. Preserve exception access, expose count and severity composition, and compare error rate as well as completion time. Random particle motion can also change scores independently of the control; use fixed seeds and equivalent layouts. The source's AI Detection Threshold is a clustering radius, not a classifier confidence threshold. [SRC; G3]

Tests that challenge the proxy. Include coincident points, a single urgent member inside a large cluster, zero assets, one asset, dense boundary labels and a very small viewport. Verify that every original asset appears once in a cluster or as an individual marker. Check that increasing the radius does not produce impossible counts or orphaned exceptions. If a lower loss accompanies a missed critical asset, the gate should fail even when the visual score improves. Equivalent seeded scenarios prevent a dramatic random re-layout from being mistaken for an effect of the threshold.

5. Results & Impact

Evidence status. The representative simulation values below are documented examples. 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.

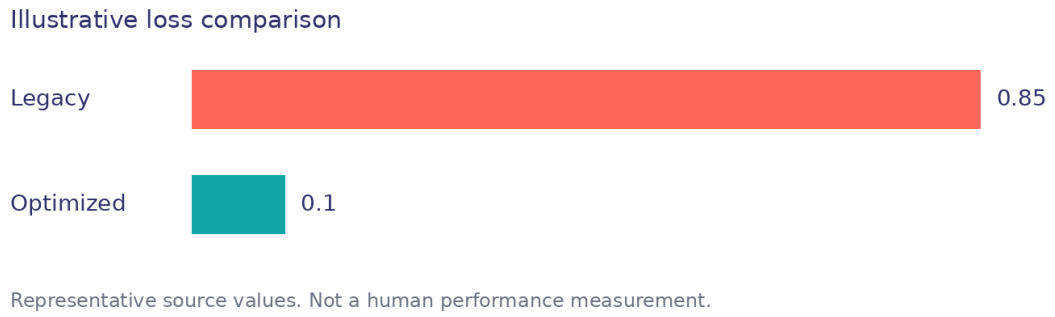

Figure 3. Representative loss values plotted from the source; the 88.2% decline is arithmetic, not a measured human outcome.

Metric	Before	After or Target	Evidence or Change
Loss	0.85 representative	0.10 representative	88.2% illustrative decline
Clustering	Fixed individual symbols	Aggregates with counts	Source concept; not field test
Exception detection	No task measurements	Collect time and miss rate	Proposed validation
Accessibility	Source claims improvement	Test each relevant criterion	No full conformance evidence
RLHF training	Visualization described	Future preference dataset	No trained policy supplied

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

Arithmetic and interpretation. Using representative endpoints 0.85 and 0.10, the relative decline is (0.85 − 0.10) ÷ 0.85 × 100 = 88.235%, rounded to 88.2%. These endpoints are explanatory examples. The result is a change in the proxy score, not an 88.2% reduction in human workload, errors, risk or task duration. The strict source inequalities do not establish precise individual trial results.

5.1 Business Impact

- **Efficiency: the demonstrator explains how rendering rules change information access; actual task-time savings need controlled measurement.**
- **Revenue and cost: a repeatable rubric may reduce ambiguous review cycles, but no measured labor saving or commercial uplift was supplied.**
- **Experience: visitors can explore the mechanism rather than read a static claim; assess comprehension and task success before claiming an educational benefit.**

> **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
Expose the score components	Keep weights, units and normalization inspectable
Make the threshold tangible	Explain screen-space radius and count conservation
Label simulation outputs clearly	Validate the proxy against real task evidence

Table 4. Retrospective highlights.

Calibration plan. Run equivalent exception-finding and region-selection tasks with operators in counterbalanced order. Retain task time, misses, selection errors, subjective effort and scenario configuration. Test whether the factor values correlate with those observations, then evaluate changed weights on held-out scenarios. If a lower loss fails to predict useful behavior, revise the proxy rather than forcing the task evidence to fit the score. [A1]

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: implement factor definitions, stable seeded scenarios and exception safeguards; retain all run configurations.
- Next six months: conduct counterbalanced operator trials and calibrate loss against time, errors and effort ratings.
- Longer term: evaluate human preference labels and held-out scenarios before describing any component as a trained RLHF system.
- Future training boundary. A genuine learning loop would need a collection protocol, independent human judgments, a held-out evaluation set, explicit model inputs and a training objective.

Compare a learned scorer with the transparent rule baseline; document disagreement and failure modes. Do not use a polished visualization as evidence that these steps happened. The first useful product can remain a deterministic review tool if it helps reviewers explain and reproduce design defects. Learning should be added only when it addresses a demonstrated limitation and can be evaluated reliably. [A1]

Appendix

A. Glossary

Term	Definition
Heuristic loss	A rule-based proxy score for a defined layout condition.
Precision radius	Source label for the proximity used to group screen-space symbols.
Normalization	Converting differently scaled factors to comparable bounded values.
Count conservation	Every original asset remains represented exactly once.
Calibration	Testing whether a proxy tracks the intended human outcome.
RLHF	Training using human feedback; visualization alone is not training.

Table 5. Terms used in this document.

B. Sources and Research Notes

[SRC] Primary source: Case Studies/Case Study Deep Dive - Mapping Machine Intent to Operational Reality/Case Study Deep Dive - Mapping Machine Intent to Operational Reality.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.

[G3] Mapbox. Supercluster Documentation. https://github.com/mapbox/supercluster

[W1] W3C. Contrast Minimum. https://www.w3.org/WAI/WCAG22/Understanding/contrast-minimum.html

[W2] W3C. Non Text Contrast. https://www.w3.org/WAI/WCAG22/Understanding/non-text-contrast.html

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

[A1] NIST. AI Risk Management Framework. https://www.nist.gov/itl/ai-risk-management-framework

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