

SurfaceMind and SurfaceDelta

Verified operational surfaces and bounded graph patches for small and large language models

Hayden Lindley · Glyphd Labs

2026-07-27

Contents

SurfaceMind and SurfaceDelta	1
Abstract	1
1. The page as external cognition	2
2. Semantic Surface Graph	2
3. SourceSpan	3
4. SurfaceDelta	3
5. Deterministic rendering	4
6. PromptCapsules and EvidenceCapsules	5
7. Small models as operators	5
8. Truth-state rendering	6
9. Versions and receipts	6
10. SurfaceMind Pocket	6
11. R0 evidence state	7
12. The decisive experiment	7
13. Additional experiments	8
Capsule ladder	8
Prefix reuse	8
Backend phases	8
Foveated project retrieval	8
14. Security	8
15. Limitations	9
16. Conclusion	9
Architecture illustration briefs	9
Source register	9
Public-disclosure and IP boundary	10

SurfaceMind and SurfaceDelta

Abstract

Language models are often asked to produce complete answers, pages, reports, dashboards, or interfaces in one generation. This gives the model too much responsibility: it must select sources, preserve state, choose layout, generate all text, maintain internal consistency, and express the result in a renderable form. Small models suffer most, but large models also produce malformed structures, unsupported claims, and destructive rewrites.

SurfaceMind separates intelligence from rendering and authority. It compiles exact sources and governed state into a versioned **Semantic Surface Graph**. A model operates the graph through small, schema-bound **SurfaceDelta** patches. A deterministic validator checks base version, target existence, citation spans, text budgets, ordering, authority, and surface invariants before the renderer updates the view. The model never generates authoritative pixels and never owns truth.

The architecture externalizes working memory into a visible surface. A compact local model can classify, extract, rank, repair, or patch one region instead of regenerating an entire answer. The system preserves exact source spans, evidence, versions, omissions, and receipts. SurfaceMind's central empirical claim—matched 4B tasks perform better through SurfaceDelta than raw full-answer chat—remains a named go/no-go experiment, not a completed result.

1. The page as external cognition

A long answer is difficult for both the model and the user.

The model must hold:

- source content;
- instructions;
- output format;
- cross-section consistency;
- evidence;
- state of prior revisions;
- typography or layout intent;
- length constraints.

The user must inspect a stream of prose and mentally translate it into:

- priorities;
- decisions;
- evidence;
- actions;
- dependencies;
- risks;
- completion state.

SurfaceMind treats a task-specific page as an external working surface. The model no longer needs to regenerate the whole representation. It operates bounded semantic blocks.

The page is not just presentation. It is a typed state graph whose structure can be validated.

2. Semantic Surface Graph

A surface contains blocks, relationships, source spans, state, and rendering constraints.

A simplified graph:

```
Surface {  
  surface_id  
  surface_type  
  version  
  source_manifest  
  blocks[]  
  edges[]  
  reading_order[]  
}
```

```

    authority_context
    freshness
    omissions[]
    available_actions[]
}

```

A block may be:

- heading;
- claim;
- evidence quote;
- risk;
- decision;
- metric;
- timeline step;
- action;
- comparison;
- diagram;
- alert;
- code;
- status;
- note.

A block is semantic before it is visual. The renderer decides how a RiskCard or EvidenceClaim appears on desktop, mobile, PDF, or accessible list view.

The model can reason about purpose and content without owning CSS or pixels.

3. SourceSpan

Every source-backed statement can point to an exact span:

```

SourceSpan {
    source_id
    source_digest
    normalization_version
    start_offset
    end_offset
    exact_text_digest
    visibility
}

```

The verifier checks that the span exists and that quoted text matches. A source revision or normalization change invalidates affected citations.

This avoids a common failure in generated interfaces: a citation label exists, but the referenced text does not support the claim.

4. SurfaceDelta

A SurfaceDelta is a bounded mutation:

```

SurfaceDelta {
    delta_id
    surface_id
    base_version
    op
}

```

```

    payload
    citations[]
    provenance
    authorizing_refs[]
    expected_invariants[]
}

```

Allowed operations may include:

```

replace_block_text
insert_block
remove_block
move_block
set_block_meta
split_block
merge_blocks
add_citation
compress_zone
mark_uncertain
request_approval

```

A delta validates only if:

- the base version matches;
- target blocks exist;
- the operation is allowed for the surface type;
- every citation resolves;
- quote text matches;
- block budgets are respected;
- reading order remains valid;
- protected blocks are not changed without authority;
- no authoritative text is rasterized;
- expected invariants remain true.

A stale delta is rejected rather than applied to the wrong page.

5. Deterministic rendering

The renderer owns:

- typography;
- spacing;
- layout;
- responsive behavior;
- accessibility;
- visual status grammar;
- print/PDF output;
- static HTML export;
- interaction scaffolding.

The model emits semantic content and bounded metadata. It does not emit arbitrary HTML or execute scripts on the main path.

This creates several benefits:

- predictable accessibility;
- consistent design;
- no hidden remote scripts;

- easier visual regression;
- smaller model output;
- safer user revision;
- cross-platform rendering;
- exact diff and replay.

A surface can be regenerated from graph state. The rendered pixels are not the canonical state.

6. PromptCapsules and EvidenceCapsules

The model receives only the active zone and relevant evidence.

A PromptCapsule can contain:

```
task
schema_id
active_zone
surface_snippet
evidence_spans
constraints
output_contract
token_budget
digest
```

An EvidenceCapsule may contain:

- exact source spans;
- source type and date;
- supported claim candidates;
- contradictions;
- confidence;
- permitted quotation;
- missing evidence;
- disclosure class.

The capsule is a bounded projection over sources and surface state. It can be cached, inspected, and compared across runs.

7. Small models as operators

The architecture changes the unit of intelligence.

Instead of:

Read all sources and write the complete excellent report.

the model receives tasks such as:

```
classify intent
select surface type
extract five claims
attach source spans
rank three risks
replace one block
repair one invalid delta
compress one zone
generate the next action strip
```

This is the central Max-Flex thesis:

We are not making a 4B model bigger. We are making its job smaller, more structured, more spatial, more cacheable, more verifiable, and more reusable.

The system can route different operations to different models or deterministic functions. A tiny classifier may choose the surface type. A 4B model may write a block. A verifier checks citations. A renderer builds the final page.

8. Truth-state rendering

The interface should show epistemic state, not merely confidence language.

One visual grammar is:

- **verified crisp:** exact source and validation passed;
- **derived:** clear but marked as a transformation;
- **speculative haze:** model inference or incomplete evidence;
- **breach/refusal:** failed proof, stale source, or forbidden operation;
- **disputed:** multiple supported states remain;
- **unknown:** information absent.

This allows the user to perceive where the surface is strong or provisional without reading an audit log.

9. Versions and receipts

Every accepted delta creates a new surface version and a receipt:

```
prior_version  
delta  
validator_results  
source_digests  
model_ref  
render_ref  
result_version  
warnings  
acceptance_state
```

A user can inspect:

- what changed;
- why;
- which model proposed it;
- which source spans support it;
- what was rejected;
- whether the rendered view is current;
- how to restore or branch.

A generated page becomes a governed artifact rather than a transient answer.

10. SurfaceMind Pocket

The mobile expression is not “chat on a phone.” It is a visual intelligence appliance.

Reference modes include:

```
Scan → Surface  
Speak → Decision Board
```

Paste → Build Packet
Screenshot → Explain Page
Contract → Risk Strip
Meeting → Action Surface
Project → Memory Atlas

The center of the screen is the surface. Commands such as shorten, show evidence, find risks, turn into tasks, what changed, and repair citations create bounded deltas.

The local model remains behind the interface. Evidence, versions, and export are one tap away.

11. R0 evidence state

The SurfaceMind R0 corpus includes:

- a locked repository skeleton;
- canonical documentation;
- a Zod-based intermediate representation;
- rendering and verification specifications;
- three verified example surfaces;
- an R0 test suite reported green.

This establishes a coherent design and schema foundation. It does not prove the full local-mobile runtime, user benefit, or SurfaceDelta uplift.

The canonical source hierarchy should prefer executable IR and accepted tests over descriptive prose when they conflict.

12. The decisive experiment

The first paper-worthy empirical test is matched:

same source

same task

same 4B model

same hardware

A: raw full-answer generation

B: SurfaceDelta operation

Measure:

- schema validity after bounded repair;
- repair count;
- citation accuracy;
- fabricated citation rate;
- tokens in and out;
- time to first useful verified content;
- full completion latency;
- memory;
- energy where available;
- blinded usefulness;
- user correction count.

The proposed gate is intentionally difficult:

- at least 99% valid deltas after bounded repair;
- mean repairs below the declared threshold;

- every citation verified or flagged;
- lower output tokens;
- non-inferior or better usefulness;
- faster useful verified content.

If SurfaceDelta does not win, the central small-model claim should fail publicly.

13. Additional experiments

Capsule ladder

Compare:

- raw source chunks;
- extractive evidence capsule;
- compressed capsule;
- graph-derived capsule.

Adopt a rung only if it reduces tokens without degrading task score.

Prefix reuse

Keep schema, style, validator rules, and surface contract stable. Swap only the evidence tail. Measure cold versus restored time to first token.

Backend phases

Measure prefill and decode separately across CPU, GPU, NPU, and hybrids. Do not assume an accelerator wins every phase.

Foveated project retrieval

Load the active node, parent, siblings, exact evidence, recent mutation, and salient landmarks rather than the full project. Compare with competent lazy loading.

14. Security

SurfaceMind handles source documents and generated actions. Threats include:

- prompt injection inside source text;
- malicious deltas;
- source-span tampering;
- cross-project leakage;
- stale base versions;
- renderer injection;
- remote scripts in exports;
- hidden authority changes;
- model-generated citations;
- oversized graph or recursive block bombs.

Controls include:

- source/content separation;
- strict operation whitelist;
- optimistic concurrency;
- exact span verification;
- deterministic renderer;

- no remote scripts in standalone exports;
- graph and text budgets;
- authority checks;
- project-scoped capsules;
- receipt chains;
- visible breach states.

A valid schema does not make content trusted.

15. Limitations

Structured surfaces can constrain creativity. Some tasks benefit from long-form synthesis and voice. The architecture should allow explicit prose blocks without making freeform output the canonical transport.

A verifier can confirm exact quote spans but cannot automatically establish every interpretive claim. Source-backed text may still be misleading. Human review remains necessary for high-stakes domains.

Small deltas can create local coherence while degrading the global page. Periodic deterministic or model-assisted whole-surface audits are required.

The page metaphor can become another rigid dashboard if surfaces are generic. The system must generate task-specific structure, not merely fill templates.

16. Conclusion

SurfaceMind does not ask the model to be the page.

It asks the model to operate a page whose truth, structure, and rendering exist outside the model. The unit of work becomes a small, inspectable delta. The unit of evidence becomes an exact span. The unit of presentation becomes a deterministic surface.

The result is a path by which small models can become more useful without pretending to become larger—and by which large models can become more accountable without giving up expressive intelligence.

Architecture illustration briefs

- **P07-F01 — SurfaceMind stack:** Sources → EvidenceCapsules → Semantic Surface Graph → SurfaceDelta validator → deterministic renderer → receipts.
- **P07-F02 — Full answer versus delta:** Large tangled generation path contrasted with one small patch to an existing block, with token and validation surfaces.
- **P07-F03 — Truth-state rendering:** Visual legend for verified, derived, speculative, disputed, stale, and breach states.
- **P07-F04 — Matched E1 experiment:** Same 4B/source/task split into raw prose and SurfaceDelta conditions, converging on common metrics.

Source register

- **P07-S01** — SurfaceMind R0 repository and canonical documentation (2026-07). bohs-electa/surfacemind
- **P07-S02** — SurfaceMind Pocket / 4B Max-Flex research notes (2026-07-13). Pasted markdown.md
- **P07-S03** — Zeke 4B Max-Flex Runtime Master Spec (2026-07-15). ZEKE-4B-MAXFLEX-RUNTIME-MASTER-SPEC-v0.1.md

- **P07-S04** — GlyphCAS Design Report (2026-07-24). `How_we_can_make_CAS_lightning_fast_for_any_LL`
- **P07-S05** — Workpage OS Spec Pack v1.1 (2026-06-18). `Workpage_OS_Spec_Pack_v1_1.md`

Public-disclosure and IP boundary

This paper publishes the public-safe graph/delta architecture and the explicit go/no-go experiment. It excludes private source data, unpublished compiler prompts, restricted mobile implementation details, and patent-claim language. SurfaceDelta uplift is not claimed until a versioned matched run is published.