

Visual-World-First Software

References, absolute locks, production envelopes, and evidence-bearing
visual implementation

Hayden Lindley · Glyphd Labs

2026-07-27

Contents

Visual-World-First Software	1
Abstract	1
1. Visual intent is usually lost between tools	2
2. Visual object model	2
3. Absolute locks	3
4. Preservation ladder	4
Level 1 — live reconstruction	4
Level 2 — hybrid reconstruction	4
Level 3 — sliced or image-backed preservation	4
5. Visual Direction Compiler	4
6. No-execution configuration	5
7. Image-first production	5
8. Omni sequence sheets	5
9. Returned assets and lineage	6
10. Visual verification	6
11. Relationship to software architecture	6
12. Relationship to ImageOS and production systems	7
13. Demonstration	7
14. Evaluation	7
Fidelity	7
Production	7
Human comprehension	8
15. Security, rights, and ethics	8
16. Evidence state	8
17. Limitations	8
18. Conclusion	8
Architecture illustration briefs	9
Source register	9
Public-disclosure and IP boundary	9

Visual-World-First Software

Abstract

Software projects usually treat visual direction as a late presentation layer: a moodboard, a design pass, or a set of images that developers reinterpret into components. In AI-assisted pro-

duction, that separation becomes especially destructive. Models generate plausible but inconsistent interfaces, characters, shots, and products; visual intent is scattered across prompts; approved references lose authority; each regeneration becomes a new interpretation.

Visual-World-First Software (VWFF) treats the visual world as an architecture artifact. References, absolute locks, influence, asset roles, continuity, layout, camera, material, typography, responsive crops, production state, and verification become typed project objects. A **Visual Direction Compiler** converts those objects into a frozen **Production Envelope** or shot/build packet. The envelope is inspectable before execution. A separate explicit action authorizes generation or implementation. Returned assets are reviewed against the envelope, attached to exact lineage, and accepted or revised.

The approach includes a preservation ladder for locked mockups: live reconstruction when faithful, hybrid reconstruction when necessary, and image-backed/sliced assets when code would weaken visual fidelity. The winning output is not the purest CSS. It is the best working system that preserves the approved visual contract.

1. Visual intent is usually lost between tools

A visual project may include:

- a reference image;
- a sketch;
- a finished mockup;
- an approved screenshot;
- brand plates;
- a character sheet;
- camera notes;
- negative constraints;
- a style description;
- generated assets;
- responsive requirements;
- implementation comments.

These artifacts have different authority. A finished mockup may be a locked contract; a mood-board may be only influence. A developer or model often receives them as an undifferentiated folder and makes a new design.

The result is visual drift:

- typography becomes generic;
- composition is “modernized”;
- logos are approximated;
- generated characters change identity;
- panels lose proportion;
- CSS purity overrides fidelity;
- mobile crops destroy hierarchy;
- image generation ignores reference order;
- execution begins before the plan is visible.

VWFF makes the difference between reference, influence, and lock explicit.

2. Visual object model

A project can define:

```
VisualWorld {  
  world_id
```

```

    project_id
    visual_thesis
    references[]
    locks[]
    assets[]
    scenes_or_surfaces[]
    continuity_rules[]
    production_envelopes[]
    revisions[]
    receipts[]
}

```

A reference includes:

```

reference_id
exact_asset_ref
role
influence_scope
strength
allowed_transformations
prohibited_uses
rights
source

```

Roles may include:

- identity;
- composition;
- material;
- lighting;
- typography;
- camera;
- color;
- texture;
- motion;
- continuity;
- negative reference.

3. Absolute locks

A lock is a constraint, not a suggestion.

Examples:

- character facial identity;
- logo geometry;
- hero composition;
- primary silhouette;
- exact aspect ratio;
- title spelling;
- key color relationship;
- spatial orientation;
- no added text;
- no extra subjects;
- approved product form.

A lock record states:

target
property
source_ref
tolerance
verification_method
failure_behavior
authority

The system can allow creative variation around locks while refusing changes to the protected core.

4. Preservation ladder

When a source image is the final intended design, preserve it before reinterpretation.

Level 1 — live reconstruction

Use HTML, CSS, SVG, fonts, gradients, layout, and components when fidelity remains high.

Level 2 — hybrid reconstruction

Use live interaction and text with extracted textures, masks, borders, plates, illustrations, or image-backed sections.

Level 3 — sliced or image-backed preservation

Retain complex visual regions as assets when code reconstruction would visibly weaken the match. Add semantic overlays, accessibility, and responsive behavior around them.

The architecture does not fetishize implementation purity. It optimizes for working fidelity, accessibility, maintainability, and performance together.

5. Visual Direction Compiler

The compiler transforms the approved visual state into a production envelope.

```
ProductionEnvelope {  
  objective  
  target_surface_or_shot  
  exact_references[]  
  reference_order  
  locks[]  
  flexible_regions[]  
  layout_or_camera  
  aspect  
  responsive_rules  
  asset_roles[]  
  text_content  
  negative_constraints[]  
  implementation_or_provider_lane  
  cost_estimate?  
  review_rubric[]  
  expected_return  
}
```

The envelope is deterministic for a given version of the Visual World and compiler.
The compiler may use a model to interpret loose direction, but the output must validate.

6. No-execution configuration

Configuration and execution are separate.

user configures
→ compiler generates frozen envelope
→ user reviews
→ explicit Make/Render action
→ provider or builder executes

Changing a model, quality, duration, reference, or lock updates the estimate and envelope. It does not automatically spend credits or overwrite assets.

This is critical for generative media and coding agents.

7. Image-first production

For video and complex interfaces, the visual plan comes before motion or implementation.

A production sequence can be:

visual bible
→ character/object references
→ timestamped scene list
→ storyboard
→ keyframes
→ continuity review
→ animation/render plan
→ approval
→ render
→ review
→ repair packet

The rule is:

Do not just output prompts. Output production packets.

A packet includes reference order, asset map, shot beats, camera rules, sound posture, negative constraints, cost, and review criteria.

8. Omni sequence sheets

A sequence sheet can place ordered visual states on one 16:9 reference.

The test keeps subject and prompt constant and changes:

- number of panels;
- order;
- layout;
- path;
- final landing image.

Observed behavior can reveal whether a video model follows panel order, camera progression, continuity, and no-text constraints.

These sheets are experimental routing artifacts, not proof that a model understands a formal visual language.

9. Returned assets and lineage

A provider Return contains:

asset_ref
provider
model
settings
input_envelope
reference_refs
cost
output_digest
review_results
limitations

The asset remains a candidate until accepted.

Lineage records:

- source references;
- generated variants;
- crops;
- edits;
- repairs;
- responsive derivatives;
- publication;
- rights.

A new asset cannot silently replace the accepted source plate.

10. Visual verification

Verification can include:

- pixel or perceptual diff;
- layout geometry;
- text extraction;
- lock-specific checks;
- identity review;
- responsive screenshots;
- contrast;
- focus and keyboard behavior;
- performance;
- human visual acceptance.

Automated image similarity cannot decide every lock. The contract identifies which checks require the founder or designer.

11. Relationship to software architecture

VWFF does not replace component architecture. It supplies a visual source of truth.

A component can declare:

visual_contract_ref
supported_states
responsive_behavior
asset_dependencies
accessibility_equivalent
visual_regression_fixture

The implementation remains modular, but components cannot drift from the world merely because a framework encourages generic cards.

12. Relationship to ImageOS and production systems

The Visual Direction Compiler can serve:

- ImageOS;
- VideoBoss;
- X-Splora;
- POLYGLYPH A/V;
- website implementation;
- game art;
- industrial design;
- architecture visualization.

Specialist products retain domain logic. The compiler provides shared visual contracts and production envelopes rather than becoming one universal app.

13. Demonstration

A browser demo can:

1. show three references with different roles;
2. promote one composition to an absolute lock;
3. set aspect and responsive crops;
4. compile a production envelope;
5. compare two returned candidate assets;
6. mark violations;
7. accept a derivative;
8. show lineage and receipt.

No paid generation is required. Fixture assets can prove the contract.

14. Evaluation

Fidelity

- lock violation rate;
- visual diff;
- identity drift;
- layout deviation;
- typography mismatch;
- responsive hierarchy.

Production

- number of regeneration cycles;
- time to accepted asset;

- cost;
- prompt length;
- continuity errors;
- repair specificity.

Human comprehension

- can the director explain what is locked?
- does the builder understand reference roles?
- can a change be made without losing accepted visual state?
- can another provider receive the same envelope?

Compare ordinary prompt folders with compiled envelopes.

15. Security, rights, and ethics

References may contain copyrighted work, real people, brands, or private environments.

The system requires:

- rights metadata;
- consent for likeness use;
- no silent training contribution;
- private/public separation;
- source and influence disclosure where appropriate;
- no embedding secret prompts in public assets;
- safe provider routing;
- removal/revocation;
- no false claim of affiliation.

The compiler cannot “sanitize” rights away through abstraction.

16. Evidence state

The project corpus includes the Visual Mockup Preservation Amendment, brand/source-plate workflows, visual manifests, image-first production prompts, Omni sequence templates, shot-packet systems, and production-skill contracts. These constitute a developed design and workflow architecture.

A unified production compiler with cross-provider benchmark results remains to be completed.

17. Limitations

Locked imagery can constrain responsive design. Sliced assets can harm accessibility or performance. Automated verification may overfit surface similarity. Visual references can contain inconsistencies. Generative providers may ignore envelopes. A highly controlled system can reduce exploration.

The architecture must distinguish exploratory reference from final contract and allow explicit unlocks.

18. Conclusion

Visual direction should not be a pile of inspiration that every downstream model reinterprets.

It should be an inspectable, versioned architecture artifact that compiles into a production envelope and survives provider changes.

The doctrine is:

Preserve the mockup before you reinterpret the mockup.

And the production rule is:

Compile the visual world. Authorize execution separately.

Architecture illustration briefs

- **P19-F01 — Visual world object model:** References and locks compile to a Production Envelope, execution Return, review, accepted asset, and lineage.
- **P19-F02 — Preservation ladder:** Live reconstruction → hybrid → sliced/image-backed, with fidelity/accessibility/performance decision gates.
- **P19-F03 — No-execution configuration:** Configuration changes envelope and cost estimate; only explicit Make/Render crosses the spend/execution gate.
- **P19-F04 — Omni sequence experiment:** Matched prompts with different ordered panel sheets and measured continuity/order outcomes.

Source register

- **P19-S01** — Visual Mockup Preservation Amendment v2 (2026-06-26). VISUAL-MOCKUP-PRESERVATION-AMENDMENT-v2.md
- **P19-S02** — X-Splora Production Skills integration notes (2026-06-30). Pasted mark-down.md
- **P19-S03** — Seedance/C-Dance Omni Sequence Template Packs (2026-07-01/02). README.txt
- **P19-S04** — Glyphd mockups visual provenance and foundry corpus (2026-07). bohse-lecta/mockups
- **P19-S05** — Glyphd/POLYGLYPH audiovisual system canon (2026-07-11). GLYPHD-ZEKE-CORGIVERSE-v4.0-COMBINED-SPEC.md

Public-disclosure and IP boundary

This edition publishes the public-safe visual contract, preservation doctrine, compiler concept, and evaluation program. It excludes private reference images, third-party copyrighted assets not cleared for publication, confidential provider prompts, and patent-claim language. No provider-specific superiority is claimed.