

Chat · Plan · Make

A human-centered AI project environment with one durable Project Core and three role-separated lenses

Hayden Lindley · Glyphd Labs

2026-07-27

Contents

Chat · Plan · Make	2
Abstract	2
1. The mode-collision problem	2
2. Chat	3
3. Plan	3
4. Make	4
5. One Project Core	4
6. Accepted, working, candidate	4
7. ZEKE worker	4
8. Work Orders	5
9. Provider bindings	5
10. Spatial grammar	6
11. Cross-lens operations	6
Chat → Plan	6
Plan → Make	6
Make → Chat	6
Make → Plan	6
Plan → Chat	6
12. Artifact-first, not app-first	6
13. Workpages	7
14. Home and projections	7
15. Cost and authority	7
16. Interaction prototype	7
17. Evaluation	8
18. Security and privacy	8
19. Evidence state	8
20. Limitations	8
21. Conclusion	9
Architecture illustration briefs	9
Source register	9
Public-disclosure and IP boundary	9

Chat · Plan · Make

Abstract

The dominant interface for artificial intelligence collapses exploration, commitment, execution, and review into one conversation. This is convenient for short tasks and cognitively expensive for serious projects. A casual thought can look like a directive. A model proposal can be remembered as a user decision. A worker result can replace an accepted artifact. The user must reconstruct current state across chats, providers, files, devices, and tools.

Glyph Desk is a human-centered project environment built around three permanent synchronized lenses over one durable **Project Core**:

- **Chat:** understand, explore, question, and evaluate;
- **Plan:** decide, constrain, structure, and authorize;
- **Make:** execute, inspect, verify, refresh, and deliver.

The machine unifies state. The interface separates responsibility.

Conversation must cross a boundary before becoming intent. Intent must cross another boundary before becoming consequential action. Accepted and working results remain separate. Models and providers rent scoped project context; the user owns the Project Core. ZEKE can continuously reduce the gap between Plan and working result only within explicit authority, budget, and evidence gates.

Glyph Desk is not differentiated by three columns. It is differentiated by the contracts underneath them.

1. The mode-collision problem

A serious user performs different cognitive acts in nearly identical chat windows:

- brainstorm;
- ask questions;
- make decisions;
- preserve constraints;
- direct agents;
- compare versions;
- review evidence;
- inspect work;
- approve spending;
- publish;
- recover context.

The system gives all of these acts the same visual and operational grammar. The human must remember which sentence had which force.

Mode collision creates:

- accidental authority;
- duplicated decisions;
- obsolete context;
- destructive regeneration;
- unclear completion;
- provider-owned project memory;
- hidden cost;
- weak artifact lineage.

Three lenses externalize the distinction.

2. Chat

Chat is the divergent surface.

It may:

- explore possibilities;
- ask clarifying questions;
- compare artifacts;
- recover prior decisions;
- discuss uncertainty;
- propose candidate decisions;
- explain work;
- attach to exact artifacts and receipts.

It may not:

- silently change Plan;
- authorize spend;
- publish;
- replace accepted work;
- treat a model suggestion as founder intent.

A Chat message can reference:

project
artifact
artifact version
plan item
work order
worker run
evidence
source provider

This lets “Is this good?” resolve to an exact object.

3. Plan

Plan is the declared-intent surface.

It contains:

- objective;
- accepted decisions;
- constraints;
- scope;
- priorities;
- dependencies;
- budgets;
- authority;
- acceptance criteria;
- risks;
- open questions;
- next valid actions.

Promotion from Chat to Plan is explicit. The system shows source and proposed interpretation.

Plan revisions are immutable. The active Plan is one current pointer.

4. Make

Make is the consequential work surface.

It contains:

- artifacts and versions;
- accepted and working results;
- candidates;
- stale state;
- active workers;
- Work Orders;
- estimates;
- previews;
- tests;
- evidence;
- receipts;
- publishing controls.

Make is not an uncontrolled agent or a file explorer with AI decoration.

A result returning to Make is not automatically accepted.

5. One Project Core

All three lenses read and write through one durable Project Core.

Core invariants:

1. material changes are append-only events;
2. current views are projections;
3. accepted results are immutable references;
4. working results may evolve;
5. actions record actor, authority, rationale, inputs, outputs, and time;
6. providers receive scoped context;
7. cross-project access is explicit;
8. state and artifacts are exportable;
9. provider transcripts are not sole memory;
10. invalid candidates cannot replace accepted results.

The Project Core lives outside provider widgets.

6. Accepted, working, candidate

The interface keeps:

```
accepted_result  
working_result  
candidate_results[]
```

ZEKE may work continuously on the working result. Promotion requires the appropriate verification and acceptance.

This preserves progress without sacrificing safety.

7. ZEKE worker

ZEKE asks:

What is the smallest authorized action that would reduce the gap between the active Plan and the working result?

It may:

- organize evidence;
- detect staleness;
- draft Work Orders;
- estimate cost;
- run deterministic checks;
- generate proxies;
- prepare candidates;
- surface contradictions.

Without authority it may not:

- spend;
- publish;
- message;
- change permissions;
- merge protected work;
- expose private data;
- alter canon;
- declare human acceptance.

The worker is visible and bounded.

8. Work Orders

A coding task through Codex illustrates the loop.

1. User selects repository artifact in Make.
2. User discusses change in Chat.
3. Direction is promoted into Plan with constraints and tests.
4. ZEKE creates a Work Order.
5. Codex receives repo, task, authority, and acceptance criteria.
6. Make receives diff, tests, screenshots, limitations, and commit references.
7. Accepted state remains protected until merge/acceptance.

The same pattern applies to media, research, design, and operations.

9. Provider bindings

A provider card is a binding, not a project copy.

```
ChatGPT / Claude / Codex / local model
    ↓ scoped connector
  Project Core
    ↓
Work Orders / Returns / Receipts
```

Changing provider does not recreate the project.

The commercial promise is:

Bring your AI. Keep your projects.

10. Spatial grammar

The visual system gives each responsibility a stable territory:

- Chat: blue/purple;
- Plan: green;
- Make: orange/amber.

All three remain spatially present. Keyboard focus follows the selected lens. Information can be dragged or promoted across boundaries with visible consequence.

The color system is not decoration. It reinforces mode.

Accessible labels, icons, and list alternatives prevent color-only meaning.

11. Cross-lens operations

Examples:

Chat → Plan

Promote a candidate decision with source, constraints, and unresolved questions.

Plan → Make

Approve a bounded Work Order with budget and evidence contract.

Make → Chat

Discuss a returned artifact or failure.

Make → Plan

Revise constraints after evidence.

Plan → Chat

Explore an unresolved assumption without changing the active Plan.

Each operation produces an event.

12. Artifact-first, not app-first

The user browses what they are making:

- video;
- website;
- document;
- code;
- image set;
- research dossier;
- architectural concept.

Specialist products appear as Stations or capabilities attached to artifacts. The user does not need to begin by choosing an app.

Branching, snapshots, diffs, recovery, and provenance happen behind the interface.

13. Workpages

WorkPage OS is the larger generated-interface direction. In Glyph Desk, a Workpage is one artifact capability.

Chat discovers the need. Plan governs the contract. Make generates the Workpage. Evidence returns to the Project Core.

Glyph Desk is the immediate general operator shell; WorkPage OS is a longer-term environment.

14. Home and projections

Glyph Desk can coexist with:

- architectural Home;
- radial Home;
- Universal Workfield;
- Context Corolla;
- mobile;
- desktop;
- TV;
- Glyphd OS.

These are projections over the same state. None becomes a second database.

The architectural Home provides place, ownership, presence, social thresholds, and state declaration. Workfield and Corolla remain protected for speed and context composition.

15. Cost and authority

Before expensive generation, Make shows:

- provider;
- model;
- duration;
- resolution;
- estimated range;
- confidence;
- budget;
- approval requirement.

Changing settings updates the estimate. It does not execute.

Cost receipts return with the artifact.

16. Interaction prototype

The first complete prototype should prove:

- three permanent lenses;
- one keyboard focus;
- cross-lens drag/promotion;
- accepted versus working protection;
- visible bounded worker;
- procedural estimate refresh;
- complete Chat → Plan → Make → review loop.

It should not begin as a Linux distribution or universal agent platform.

17. Evaluation

Compare Glyph Desk with ordinary provider chats across multi-session projects.

Measure:

- time to recover current state;
- accidental plan changes;
- constraint retention;
- accepted artifact loss;
- provider-switch cost;
- duplicated work;
- verification comprehension;
- time to accepted result;
- user cognitive load;
- number of re-explanations.

A three-column layout alone is not the treatment. The Project Core and transition contracts are.

18. Security and privacy

Risks include:

- cross-project context leakage;
- provider transcript capture;
- hidden Work Orders;
- stale approval;
- cost execution;
- worker self-escalation;
- private artifact publication;
- source injection;
- accepted-result replacement.

Controls include scoped bindings, explicit Plan/Make promotion, capability leases, budgets, non-destructive versions, local-first state, receipts, and human acceptance.

19. Evidence state

The Glyph Desk master specification is highly developed, including product boundaries, entities, event examples, connector route, Work Orders, artifact model, cost, design doctrine, and interaction proof.

Related donor systems implement portions: deterministic build engines, process directors, receipt schemas, project stores, prototypes, registry, SurfaceMind, WorkPage OS, and local model runtimes.

A unified finished Glyph Desk product is not claimed by this paper.

20. Limitations

Three persistent lenses consume space. Small screens require focused projections. Users may resist promotion steps. A continuously working agent can create noise. Project Core schemas can become complex. Provider connectors vary.

The system must keep low-risk flow fast and reserve ceremony for consequence.

21. Conclusion

Glyph Desk separates three irreducible human roles while sharing one durable machine state.

Chat is where the user discovers meaning.

Plan is where the user declares intent.

Make is where the system continually attempts to make that intent true.

The product sentence is:

Glyph Desk is the place where a person can think with AI, turn thought into declared intent, and keep an always-working machine moving safely toward a protected result.

Architecture illustration briefs

- **P22-F01 — Three lenses, one core:** Blue Chat, green Plan, amber Make around one Project Core, with explicit transition gates.
- **P22-F02 — Non-destructive result model:** Accepted immutable reference, evolving working result, candidates, evidence, and promotion.
- **P22-F03 — Provider bindings:** Multiple AI/tool providers connect through scoped bindings to one Project Core.
- **P22-F04 — End-to-end accepted act:** Capture → interpretation → Plan → Work Order → Return → Verification → Acceptance → Receipt → Home/Library projection.

Source register

- **P22-S01** — Glyph Desk Master Specification v1 (2026-07-12). GLYPH-DESK-MASTER-SPEC.v1.md
- **P22-S02** — Chat-Plan-Make Approval Package (2026-07-13). CHAT-PLAN-MAKE_Approval_Package_v1.p
- **P22-S03** — Glyphd V1 Canon Recovery and Gap Audit (2026-07-18). ZEKE-V1-CANON-RECOVERY-AND-GAP-AUDIT-v0.1.md
- **P22-S04** — Glyphd V1 Master-Spec Locks and Experiment Plan (2026-07-18). GLYPHD-V1-MASTER-SPEC-LOCKS-AND-EXPERIMENT-PLAN-v0.1.md
- **P22-S05** — Native Accepted Act build directives (2026-07-20). Pasted text.txt

Public-disclosure and IP boundary

This edition publishes the public-safe product architecture and interaction contracts. It excludes private project data, connector secrets, unpublished commercial details, internal provider agreements, and patent-claim drafting. It does not claim that a unified production Glyph Desk has already passed the full accepted-act gate.