

MemoSlide Technical Handover



Product and system overview

MemoSlide is a web application that turns source memos into structured presentation plans. Users review detected source sections, map content to a narrative framework, edit slide cards, connect evidence, collect comments, approve versions, share controlled reviews, and export the result.

The transferred application uses a static browser client with managed authentication, application data, file storage, real-time functions, and subscription billing. The buyer must create and connect buyer-owned service accounts before a public launch.

Main application areas

Public website

The public site contains the home page, feature pages, pricing, narrative frameworks, examples, guides, support entry, and legal pages. Public framework and example content can be managed from the owner area.

Creator workspace

Authenticated creators can complete onboarding, manage projects, import memo text or plain-text files, review parsed sections, plan the narrative, edit slides, manage brand kits, review versions, collaborate, share, and export.

Team review

Workspace roles include Owner, Editor, Commenter, and Viewer. Server-side permission checks protect workspace data. Comments, assignments, due dates, approval states, notifications, presence, share scopes, expiry, and revocation support review work.

Platform owner area

The separate owner area manages users, workspaces, plans, frameworks, slide patterns, public content, support requests, payment events, storage use, and audit records. Replace the transferred bootstrap access during handover and keep owner credentials private.

Architecture summary

Browser client

- React and TypeScript provide the user interface.
- Vite creates a static production build.
- Tailwind CSS and local style files provide the visual system.
- React Router handles public, authenticated, shared-review, and owner routes.

- PptxGenJS creates editable PPTX files in the browser.

Managed backend

- Convex stores application data, provides real-time queries and mutations, runs server actions, stores files, and processes scheduled or HTTP workflows.
- Clerk provides creator authentication.
- Stripe Checkout and Billing handle MemoSlide subscription plans.
- Verified Stripe webhooks update subscription state. Redirects alone do not grant paid access.

Deployment model

The browser build can be hosted as static files on shared hosting that supports HTTPS and route fallback. Convex remains the managed backend and must be deployed separately. Clerk and Stripe remain managed third-party services.

Project structure

- `src/` contains the React client, routes, pages, components, shared client utilities, and export code.
- `src/pages/public/` contains public product and content pages.
- `src/pages/app/` contains creator workspace pages.
- `src/pages/AdminPage.tsx` contains the separate owner interface.
- `src/lib/exportDeck.ts` creates Markdown, JSON, and PPTX exports in the browser.
- `shared/analysis.ts` contains deterministic memo and deck analysis rules shared by the application and tests.
- `convex/` contains the schema, authentication rules, data functions, billing, collaboration, sharing, owner operations, seeding, exports, and HTTP endpoints.
- `public/` contains static assets, fonts, icons, and the shared-hosting route fallback file.
- `.env.example` lists public browser variables and required server variables using placeholders.
- `srs.md` describes the product requirements and boundaries.
- `docs/SRS-COMPLETION.md` maps implemented areas to repeatable verification gates.

Data model

The Convex schema includes these main groups.

Identity and workspaces

- Users store the Clerk identity, profile, status, and onboarding state.
- Workspaces store ownership, plan, status, and usage totals.
- Memberships connect people to workspaces with Owner, Editor, Commenter, or Viewer roles.
- Brand kits store logos, colours, fonts, spacing, footer text, and confidentiality labels.

Planning content

- Projects store the audience, objective, desired action, target slide count, framework, ratio, brand kit, and status.
- Memo versions keep the source text and a parsed-section snapshot.

- Memo sections keep headings, content, section type, importance, and evidence markers.
- Deck versions store immutable version numbers, status, and framework snapshots.
- Slides store order, stage, purpose, headline, takeaway, body points, evidence, visual direction, speaker notes, source references, layout, status, and approval data.
- Recommendations store rule-based analysis results and user approval state.

Collaboration and sharing

- Comments can target a project or slide and include status, assignment, due date, and mentions.
- Presence records provide a light real-time activity indicator.
- Share links store a hashed token, scope, expiry, and revocation state.
- Notifications record review and workspace events.

Billing, content, and operations

- Plans contain pricing, limits, access rules, Stripe mappings, trial settings, and visibility.
- Subscriptions and usage records track plan state and quotas.
- Stripe event records make webhook processing idempotent and auditable.
- Frameworks, framework versions, and slide patterns control story-planning options.
- Public content records support guides, examples, FAQs, testimonials, and legal pages.
- Support requests and audit records support owner operations.

All project and file functions must enforce workspace membership. Public examples use separate curated records. Shared review functions return only the version and actions allowed by the share scope.

Required buyer-owned services

Create new buyer-owned accounts and values. Do not reuse seller credentials.

Public browser variables

```
VITE_CONVEX_URL=https://YOUR_CONVEX_DEPLOYMENT.convex.cloud
VITE_CLERK_PUBLISHABLE_KEY=YOUR_PUBLIC_CLERK_KEY
VITE_STRIPE_PUBLISHABLE_KEY=YOUR_PUBLIC_STRIPE_KEY
VITE_SITE_URL=https://memoslide.com
```

Only public values may use the VITE_ prefix.

Server-only variables

```
CLERK_JWT_ISSUER_DOMAIN=YOUR_CLERK_ISSUER
CLERK_SIGNING_KEY=YOUR_SERVER_ONLY_CLERK_KEY
STRIPE_SECRET_KEY=YOUR_SERVER_ONLY_STRIPE_KEY
STRIPE_WEBHOOK_SECRET=YOUR_SERVER_ONLY_WEBHOOK_SECRET
SITE_URL=https://memoslide.com
APP_ID=memoslide
APP_NAME=MemoSlide
SEED_SECRET=YOUR_LONG_RANDOM_SERVER_SECRET
```

Store server-only values in the Convex deployment environment. Never commit them, place them in browser variables, or include them in client bundles.

Setup and installation

Use pnpm, which matches the transferred lockfile.

1. Install a supported Node.js release.
2. Open the transferred project folder.
3. Run `pnpm install --frozen-lockfile` when dependencies are not already present.
4. Copy `.env.example` to an ignored local environment file.
5. Add the buyer's public Convex, Clerk, Stripe, and site values.
6. Create a buyer-owned Convex deployment and add all server-only values there.
7. Configure Clerk for the production domain and Convex authentication.
8. Start Convex development with `pnpm exec convex dev`.
9. Start the browser client with `pnpm dev`.
10. Sign in with separate test accounts and verify workspace isolation.

The demo seed is manual and idempotent. Use it only in a test environment until its content has been reviewed. Never treat demo users, projects, comments, exports, payments, or testimonials as real activity.

Payment configuration

MemoSlide uses Stripe Checkout and Billing for its own Free, Pro, and Team plans. It does not use Stripe Connect and it does not route payments between customers and third-party providers.

1. Create buyer-owned Stripe products and recurring prices.
2. Map the Stripe product and price identifiers to the matching MemoSlide plans in the owner area.
3. Add the buyer's server secret and public publishable key in the correct environments.
4. Point the Stripe webhook to the deployed application endpoint.
5. Add the buyer's webhook signing secret to the server environment.
6. Test checkout, duplicate webhook delivery, upgrade, downgrade, cancellation, failed payment, customer portal, and plan enforcement in test mode.
7. Move to live keys only after test results are satisfactory.

Paid access must come from verified webhook state, not from the browser redirect after checkout.

Authentication and owner access

Clerk controls creator sign-in. Convex authorization functions connect the authenticated identity to a user, workspace, and role before allowing protected data operations.

The platform owner area uses a separate session boundary. During transfer, change the owner username and password, revoke old sessions, and store the new values in the buyer's password manager. Do not share owner access with normal creator accounts.

Deployment sequence

1. Prepare managed services

Create production projects in Convex, Clerk, and Stripe. Configure the production domain, allowed origins, authentication issuer, Stripe products, prices, and webhook endpoint.

2. Deploy Convex

Add server-only values to the production Convex environment. Run the project's Convex deployment command from the transferred folder. Record the production Convex URL for the browser build.

3. Build the browser client

Add only public production values to the build environment and run `pnpm build`. The command performs TypeScript checks and creates the static output.

4. Upload to shared hosting

Upload the contents of the generated `dist/` folder to the web root for MemoSlide.com. Shared hosting is sufficient when it supports static files, HTTPS, the included route fallback, and the required content types. Confirm that the fallback file is present so direct visits to application routes load the client.

5. Configure the domain

Point MemoSlide.com to the buyer's hosting, enable HTTPS, redirect the unwanted hostname variant, and confirm that authentication and billing callbacks use the final HTTPS address.

6. Run production checks

Create fresh test accounts and complete the functional checklist below. Remove or clearly label demo content before marketing the service.

Security and credential rotation

- Replace every transferred credential, token, webhook secret, owner password, and deployment key.
- Keep server secrets in the managed server environment only.
- Limit owner access and enable multi-factor authentication where the service supports it.
- Review Clerk sessions, allowed origins, redirect URLs, and production domains.
- Verify workspace permission checks for every query, mutation, upload, share, and export.
- Keep share tokens hashed, scoped, expiring when appropriate, and revocable.
- Validate uploaded file types and sizes and keep imported content plain.
- Review rate limits for authentication, shares, comments, exports, support forms, and webhooks.
- Verify that Stripe webhook signatures are checked and events are processed once.
- Review privacy, deletion, export, retention, and backup policies before launch.
- Never publish demo testimonials, revenue, user counts, or payment records as real evidence.

Functional verification checklist

Public and account access

- Public pages load over HTTPS and direct links do not return a hosting error.
- Creator registration, sign-in, sign-out, and session recovery work.
- The owner area uses its separate access flow.

Workspace and planning

- A creator completes onboarding and creates a workspace and project.
- Plain text and Markdown memo import show a safe preview and create a memo version.
- Detected sections can be renamed, merged, split, prioritised, and linked to evidence.
- Slides can be created, edited, reordered, duplicated, split, combined, archived, and restored.
- Planning recommendations show their rule and require user approval.
- Version conflicts do not silently replace newer work.

Collaboration and sharing

- Owner, Editor, Commenter, and Viewer permissions deny unauthorised actions.
- Comments, assignments, due dates, mentions, resolution, approval, and reopening work.
- View and comment share links respect their scope, expiry, and revocation.
- One workspace cannot read or change another workspace's private data.

Export

- Markdown and JSON exports contain the expected project snapshot.
- PPTX opens in a presentation application and contains editable slides and speaker notes.
- Print output has readable page breaks and no clipped content.
- Export validation reports missing headlines, source gaps, unresolved comments, and overflow risks.
- Export records contain the correct version, format, user, time, and validation state.

Billing and owner operations

- Free, Pro, and Team limits match the owner configuration.
- Checkout, webhook updates, portal access, cancellation, and grace handling work in test mode.
- Owner changes to plans, framework versions, slide patterns, public content, and account status are recorded.
- Support, storage, payment events, and audit records are visible to the owner only.

Backup, monitoring, and maintenance

- Set a regular Convex data export and restoration process.
- Record how buyer-owned files and brand assets are backed up.
- Monitor frontend errors, Convex function failures, authentication errors, webhook failures, payment state, and storage growth.
- Review dependency updates in a test branch before release.

- Run focused tests and type checks after code changes, then run one production build before deployment.
- Test the full purchase and cancellation path after billing changes.
- Review privacy and retention rules when adding new data fields or integrations.
- Keep an incident log for access, billing, data, and service failures.

Product boundaries

MemoSlide plans and exports decks. It is not a full PowerPoint editor. It does not currently include generative AI, document sync with Google or Microsoft, live presentation hosting, automatic image generation, meeting transcription, arbitrary PDF or DOCX import, native mobile apps, or live collaborative cursors.

The transferred package does not prove production uptime, active payments, revenue, traffic, customers, or completed deployment under the buyer's accounts. The buyer should complete all configuration and verification steps before public launch.

Support boundary

Codeblix provides up to 30 days of reasonable technical handover support for transfer questions and initial setup guidance. New features, third-party charges, ongoing operations, marketing, legal advice, and unlimited custom work are outside this support unless separately agreed.