

WebinarTo Technical Handover

Operational guidance for the buyer and product team

1. System overview

WebinarTo is a single-page web application for webinar discovery and host operations. It has four main areas:

- Public discovery pages for webinars, topics, hosts, series, guides, pricing, and legal content.
- Authenticated attendee pages for saved events, registrations, schedules, event rooms, replays, notifications, and account actions.
- Authenticated host pages for organizer setup, webinar creation, registrant management, event operations, analytics, billing, and settings.
- A separate protected admin area for approvals, marketplace control, plans, content, reports, privacy requests, and platform settings.

The application does not stream or record video. Hosts provide authorized HTTPS links for live sessions and replays.

2. Architecture

Frontend

- React and TypeScript provide the application interface.
- Vite handles development and the static production build.
- React Router maps public, attendee, host, event-room, replay, and admin routes.
- Tailwind CSS and application CSS provide responsive styling.
- Clerk provides attendee and host authentication.

Backend and data

- Convex stores application records, exposes queries and mutations, handles realtime updates, file storage, scheduled work, and HTTP actions.
- Stripe Checkout and Billing are used for paid host plans and featured event purchases.
- Stripe webhook events are signature checked and stored for idempotent processing.

- The frontend can be deployed as static files. Convex remains the hosted backend and must be deployed separately.

Security boundary

Browser variables may contain public identifiers only. Authentication secrets, payment secrets, webhook secrets, and other privileged values belong in the backend deployment environment. Never copy a server secret into a variable whose name begins with VITE_.

3. Main user roles and application areas

Attendee

Attendees browse approved webinars, register, save events, manage schedules, enter event rooms, ask and vote on questions, answer polls, access approved resources and replays, and submit reviews when allowed.

Host or organizer

Hosts manage organizer profiles, team members, series, speakers, webinars, sessions, registration fields, capacity, waitlists, event operations, resources, replay settings, plan billing, and analytics.

Platform administrator

The separate admin area manages organizer approval, webinar publication, featured placement, plans, content, reports, privacy requests, operational records, and administrator credentials.

4. Project structure

The following relative paths are useful during maintenance:

- `src/` contains the React application, route components, product configuration, shared components, and styles.
- `src/srs/` contains the main public, attendee, host, event operations, settings, privacy, and admin screens.
- `src/home/` contains the public homepage.
- `public/` contains brand files and local webinar cover assets.
- `convex/` contains the data schema, queries, mutations, actions, scheduled jobs, Stripe event handling, file actions, and generated API types.
- `convex/schema.ts` defines the backend tables and indexes.
- `convex/http.ts` exposes the signed Stripe webhook endpoint.

- `convex/seed.ts` contains idempotent demonstration seed actions.
- `.env.example` lists public browser settings and the names of required server settings.
- `vite.config.ts` contains build and development configuration.
- `package.json` contains the `pnpm` scripts and package versions.

5. Data model

Identity and access

- `users` stores the Clerk subject, role state, profile details, timezone, interests, plan state, and account status.
- `adminCredentials` stores the administrator username and a one-way password hash.
- `adminSessions` stores short-lived administrator sessions.

Host and content records

- `organizers` stores the public host profile, approval state, plan code, and usage.
- `organizerMembers` links a host to owner, manager, or moderator members.
- `speakerProfiles` stores reusable speaker details.
- `series` groups related webinars and brand settings.
- `webinars` stores event content, settings, publication state, approval state, and version information.
- `sessions` stores session dates in UTC, display order, host URLs, and link visibility rules.
- `registrationFields` stores custom questions, required flags, consent flags, and publication versions.

Attendee and event records

- `registrations` stores attendee answers, consent version, source information, state, notes, tags, and attendance.
- `waitlistEntries` stores the position and promotion time for capacity-controlled events.
- `questions` and `questionVotes` store moderated event questions and attendee votes.
- `polls` and `pollResponses` store poll settings, state, options, and responses.
- `resources` and `replays` store approved links, visibility settings, access windows, and replay calls to action.
- `reviews` stores post-attendance or replay reviews and moderation state.

Commercial and operational records

- `plans` stores plan codes, prices, intervals, features, limits, active state, and Stripe references.

- subscriptions stores payment customer, subscription, checkout, status, and plan information.
- promotionOrders stores featured event purchases and active periods.
- webhookEvents provides event idempotency and operational history.
- notifications, cmsEntries, contactRequests, reports, privacyRequests, auditLogs, and rateLimits support platform operations.

Public queries must not return host join URLs, private registration information, draft content, or moderation-only values.

6. Required buyer-owned services

Create buyer-owned accounts and values for:

- Clerk application for attendee and host identity.
- Convex project for backend data, functions, storage, and scheduled work.
- Stripe account for host subscriptions and featured event payments.
- Domain registrar and DNS control for WebinarTo.com.
- Static website hosting with HTTPS and single-page route fallback.
- Email delivery if the buyer adds external email notifications.
- Monitoring, backups, and analytics selected by the buyer.

Use placeholders when preparing configuration:

```
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://webinar.to
```

Set the following only in the Convex deployment environment:

```
CLERK_SIGNING_KEY=YOUR_CLERK_ISSUER_OR_SIGNING_VALUE
STRIPE_SECRET_KEY=YOUR_SERVER_ONLY_STRIPE_KEY
STRIPE_WEBHOOK_SECRET=YOUR_SERVER_ONLY_WEBHOOK_SECRET
SITE_URL=https://webinar.to
APP_ID=webinar.to
APP_NAME=WebinarTo
```

Optional webhook forwarding values must also remain server-side. Do not place real secrets in source files, screenshots, documentation, support tickets, or browser variables.

7. Setup sequence

Use pnpm because the transferred package includes a pnpm lockfile.

1. Open the transferred project folder.

2. Copy `.env.example` to an ignored local environment file.
3. Add the buyer's public Clerk and Convex values.
4. Install exact package dependencies with `pnpm install --frozen-lockfile` when dependencies are not already present.
5. Sign in to the buyer's Convex account and link the intended development deployment.
6. Add all server-only values to the Convex deployment environment.
7. Start the backend development process with `pnpm convex:dev`.
8. In a separate terminal, start the frontend with `pnpm dev`.
9. Test public, attendee, host, event-room, replay, billing, and admin routes.

Do not use seller service values in a buyer environment. Do not commit local environment files.

8. Initial data and demonstration content

The seed actions are idempotent and intended for demonstration or a controlled initial catalog. They create sample organizers, webinars, speakers, registrations, polls, questions, reviews, resources, pricing, and dashboard history.

For a development deployment:

```
pnpm seed
pnpm seed:demo
pnpm seed:media
```

For a production deployment, run the matching Convex actions explicitly against the production target only after confirming the selected project. Never run seed commands against an established live database without reviewing their effect.

Demo records are not customers, revenue, traffic, or proof of demand. Remove or clearly label them before public promotion.

9. Stripe setup

Host plans

1. Create or confirm the buyer's Stripe products and recurring prices.

2. Open the WebinarTo admin Billing area.
3. Save the matching product and price references on each paid plan.
4. Confirm the plan is active.
5. Start a checkout as a test host.
6. Complete the payment in Stripe test mode.
7. Confirm the subscription record and user entitlement update.
8. Cancel or change the subscription through the billing portal and confirm the new state.

Featured events

The featured event action creates a one-time Stripe Checkout payment. A completed and verified webhook creates the promotion order and marks the placement active for its configured period.

Webhook

Configure Stripe to send events to:

```
https://YOUR_CONVEX_HTTP_HOST/stripe/webhook
```

Confirm that:

- The signature is rejected when invalid.
- Repeated delivery of the same event does not duplicate the result.
- Checkout completion grants the expected plan or promotion.
- Subscription updates and deletions change the stored entitlement.
- Failures appear in operational logs without exposing secrets.

Keep test mode enabled until every billing check passes. Complete tax, refund, invoice, and legal review for the buyer's market before live payments.

10. Authentication and administrator setup

Clerk protects attendee and host routes. Convex functions must continue to enforce identity and ownership checks on the server, even when the interface hides an action.

The admin area uses a separate credential and session flow. The temporary installer credential is transferred through a secure channel, not in this document. On first access:

1. Sign in to the Admin page.
2. Open Settings.
3. Set a new username with at least four characters.
4. Set a unique password with at least eight characters.
5. Sign out and test the replacement login.
6. Remove seller access and store the new credential in the buyer's password manager.

Admin sessions expire after eight hours. Review this boundary if the buyer's risk requirements are stricter.

11. Deployment guide

Backend

1. Link the source package to the buyer's production Convex project.
2. Add production server values in Convex.
3. Deploy the Convex functions.
4. Run only the reviewed initial data actions required for the new deployment.
5. Configure and test the production Stripe webhook.

Frontend

1. Set the production public values in the build environment.
2. Run one production build near the end of the release process:

```
pnpm build
```

3. Upload the generated static output to HTTPS hosting.
4. Configure the host so unknown application routes return the main `index.html` file.

5. Confirm direct refreshes work for public webinar pages, host pages, attendee pages, event rooms, replays, and the admin page.

Shared hosting is enough for the static frontend when it supports HTTPS and the required route fallback. The Convex backend remains hosted by Convex.

12. Functional verification checklist

Public and attendee

- The homepage, webinar directory, topic, host, series, guide, pricing, contact, and legal pages load.
- Only approved and published webinars appear publicly.
- Search and topic filters return the expected results.
- Dates show a timezone and sessions are stored in UTC.
- An attendee can sign up, choose interests, save a webinar, register, cancel, and view a schedule.
- Capacity, approval, waitlist, required fields, and consent versions behave correctly.
- Join links remain hidden until the configured access time.
- Questions, votes, polls, resources, replay access, and review rules work.

Host

- A host can create a profile and submit it for approval.
- The seven-step builder validates required content before publication.
- A host can add sessions, speakers, fields, consent, capacity, questions, polls, resources, and replay settings.
- Registrant search, notes, tags, attendance, approval, waitlist promotion, and CSV export work.
- Host analytics and plan usage show the expected data.

Owner and payments

- Admin login and credential replacement work.
- Organizer and webinar approvals create audit entries.
- Plan changes appear on the public offer and checkout flow.
- Test subscription and featured event payments produce the correct entitlements.
- Duplicate Stripe events do not duplicate subscriptions or promotions.
- Reports, contact requests, privacy requests, content changes, and maintenance settings work.

Experience and safety

- Keyboard focus is visible and controls are usable without a mouse.
- Reduced motion, responsive layout, readable contrast, and mobile routes are checked.
- External host and replay links use HTTPS and show the intended safety boundary.
- Rate limits apply to registration, questions, votes, reports, and other sensitive actions.
- No browser response exposes server secrets, private registration data, or hidden host links.

13. Backups, monitoring, and maintenance

- Configure backups or exports for the buyer's Convex data and stored files.
- Record the recovery steps and test a restore in a nonproduction environment.
- Monitor frontend availability, Convex errors, Stripe webhook failures, authentication errors, and payment state differences.
- Review dependency updates in a separate branch or copy, then run targeted checks before one final build.
- Review organizer approvals, reports, privacy requests, failed payments, and audit history on a regular schedule.
- Keep domain, HTTPS, Clerk, Convex, Stripe, hosting, and monitoring billing under buyer-controlled accounts.
- Rotate credentials after staff changes or suspected exposure.

14. Product boundaries

The current product does not include native video streaming, video recording, webinar ticket sales, host payouts, Stripe Connect, external calendar sync, built-in email or SMS delivery, native mobile applications, generative AI, or automated certificates.

Hosts are responsible for the rights and safety of the live and replay links they provide. The buyer is responsible for production configuration, legal review, privacy compliance, payment activation, customer support, monitoring, backups, marketing, and ongoing maintenance.

15. Handover and support

Codeblix provides 30 days of reasonable technical orientation after transfer. This covers understanding the application package, connecting the main buyer-owned services, and answering setup questions. It does not include unlimited development, ongoing operations, legal services, marketing, or free future features.