App Composer

Build it.
Keep it running.

A creation and operations platform for people and their agents

Dean Lofts / Technical sales brief

The mission

Make it simple for anyone to create, own and improve useful things with sound engineering underneath.

For the person	An idea becomes a working project they can understand, change and keep.
For Cloudflare	More people adopt Workers and repeatedly use worthwhile native Workers AI capabilities.
For App Composer	Earn trust and optional service revenue by removing continuing work for the owner.

The method behind App Composer

Some parts were worth buying. Some were worth reusing. Some needed custom work.

Music and WordPress

Make something people can actually use.

Systems and DevOps

Understand deployment, ownership and failure.

AI agents

Create more, while retaining judgement and review.

App Composer

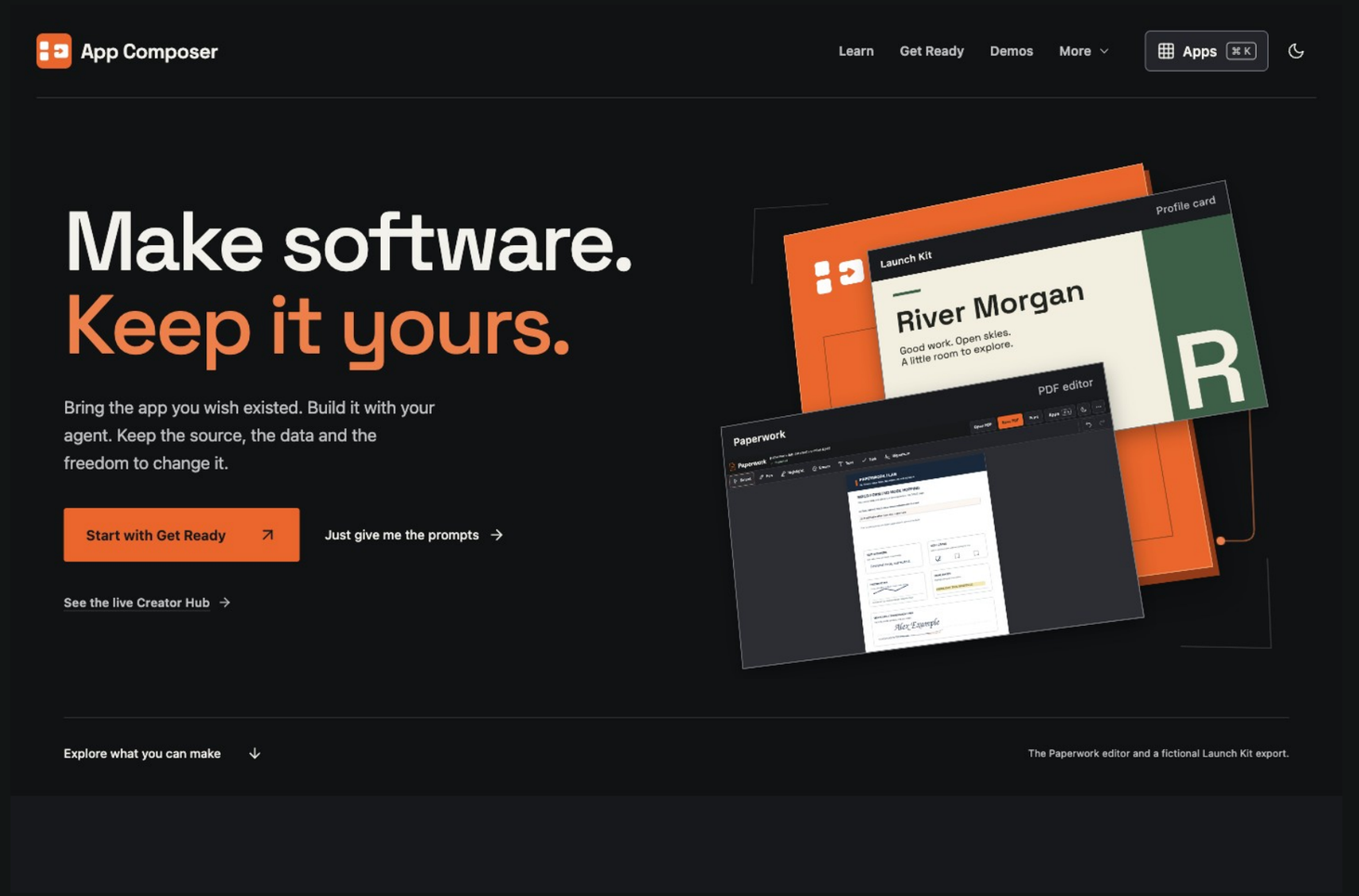
Make that method usable by someone else.

The project needs continuity

A first result still needs an owner, an editor, a release path and a next change.

Existing foundations span these jobs.

The next proof joins them for an independent newcomer.



Repository capture of the public interface. Full independent-owner acceptance remains open.

One project, through its useful life

Moment	What the person does	What the system must preserve
Create	Choose an outcome and make it personal	Intent, references and accepted choices
Own	Launch on their selected accounts	Source, data, identity and permissions
Improve	Edit content or request new behaviour	Context, customisation and working releases
Operate	Understand cost, update and recover	Evidence, budgets and a usable exit path

First proof: an owned Astro site. Fuller proof: the native EmDash Creator Hub.

Where each system belongs

System	Owns	Connects through
App Composer	Intent, project relationships and the next action	Native UI and scoped operation contracts
EmDash	Content, editorial state and native administration	Roles, revisions and installed MCP tools
GitHub + chosen agent	Source, changes and authorised implementation	Repository, API, CLI and supported MCP
Cloudflare	The application and selected runtime services	Workers, D1/R2, Access, AI and operations
Media Studio	Editable originals, media jobs and rights	Versioned assets and scoped capabilities
Optional remote host	Available execution for a qualified job	Existing provider adapter and job identity

Each system remains authoritative for its own records and permissions.

Why Cloudflare could care

Useful demand

Independent owners

Useful live applications

Meaningful return use

Runtime AI that earns repeat use

What Composer contributes

Approachable recipes and setup

A continuous ownership and editing path

Evidence of cost and technical rescue

Specific product feedback from real work

Strategic fit is our hypothesis. No partnership or endorsement is implied.

Cloudflare feedback with less glue

Evidence	Question	Useful output
Observed gap	What does this project actually need?	A source-backed, versioned finding
Current approach	What does our implementation do today?	The workaround or explicitly unbuilt capability
Burden	What extra work or failure does it cause?	Supported calls, restrictions or measured cost
Smallest ask	What would remove that burden?	A precise API, tooling or documentation request

Example: a whole-app preview must name every data and service dependency.

MCP, browser access and compute

Mechanism	Best job	What it does not supply
API / MCP / CLI	Precise source, content and deployment actions	Compute or permission merely by connecting
Browser / computer use	Existing interfaces and visual acceptance	Reliable execution while its host is unavailable
Local / remote runtime	The machine that performs the work	Implicit authority over production

Preferred path: structured operations first, UI interaction where it adds necessary capability.

Hands off for qualified routine work

The agent carries the job

Save intent, target, budget and limits

Discover tools and implement in isolation

Inspect the preview and release with authority

Verify live behaviour and retain the result

The person handles exceptions

Account sign-in, passkeys and 2FA

New spending or broader permission

Unresolved creative choices

Consequential changes outside standing policy

Target: zero routine technical intervention after supported setup.

The first conversation

“What is one thing you have wanted online but never got around to making?”

1. Choose one useful outcome and a relevant example
2. Make their content or taste visible in a draft
3. Explain ownership and connect only required access
4. Publish, then let them make the next change
5. Leave a clear way to reopen and return

A proposed assisted session. Independent newcomer acceptance is still to prove.

Prerequisites and likely friction

Need	Where people get stuck	Our responsibility
A small goal and usable content	Too many choices or uncertain asset rights	A relevant example and guided brief
Supported agent and available host	Entitlement, setup or a sleeping laptop	One qualified path and a resumable next step
GitHub and Cloudflare access	Sign-in, installation scope and billing	Explain each grant and keep target identity
Publication and cost choices	Private source can produce a public site	Separate exposure and show budgets clearly
A way to return	Expired access or forgotten editing steps	Retained project context and a short guide

Usually defer: custom domain, migration, extra model keys and optional runtime AI.

The proof plan

Horizon	Target	Evidence required
3 active delivery days	Owned Astro publication and later edit	Isolated preview, failed candidate and recovery
10 active delivery days	Native Creator Hub slice	Independent newcomer and meaningful functional change
10-person pilot	7 launches and 5 meaningful returns	Each enrollee gets 30 days; all assistance stays visible
90-day stretch	100 owners and 30 qualifying returns	Mature cohorts, useful AI and three creation categories

Planning targets, not achieved traction or guaranteed delivery dates.

How we sell it

Online

Show one useful result and its next change

Link directly to a working example

Explain ownership, prerequisites and cost

Offer one clear “Make it yours” action

In person

Ask about their idea

Let their content appear early

Help only at the actual point of friction

Let them perform the next edit

Minimum launch asset: one useful page, the deck downloads and one supported starting route.

The investment case

Why I would consider a bet

Founder judgement spans making and operating

Ownership is a clear product commitment

A narrow first proof can test a broad vision

Inspectable work and explicit acceptance exist

Why I would wait to scale

Independent retention remains unproven

Willingness to pay is still a hypothesis

Dean may be the hidden support layer

Platforms can absorb overlapping features

A milestone-based early case. Growth capital needs repeatability and unit economics.

Where the plan could fail

Risk	Early signal	Response
Setup overwhelms people	Abandonment at accounts or agent setup	One supported route, retained progress and fewer choices
The founder is indispensable	Dean repairs every successful session	Measure rescue and repeat without him
The result has no lasting value	Launches rise, meaningful returns do not	Improve usefulness and the editing loop
Operations lose trust	Wrong target, exposure or duplicate effects	Scoped authority, isolation and reconciliation
The business cannot sustain use	Support cost rises while payment stays weak	Test a defined service before broad expansion

A wider catalogue cannot repair a broken first journey.

If this works

Someone you have never met makes something useful, returns to improve it, and succeeds without needing you in the room.

Creators keep their projects and contribute useful examples.

Agencies operate client work with clean ownership.

Cloudflare gains retained workloads and worthwhile AI demand.

Dean directs the product and handles the exceptions.

The next ask: prove the journey with ten independent people.

Technical sources and claim boundaries

Product and delivery	PRODUCT.md, ARCHITECTURE.md and the adopted roadmap
Founder narrative	Owner-approved founder-story.md
Human and agent content work	EmDash MCP reference and installed-version discovery
Browser operation	OpenAI browser extension and OpenCode Browse
Cloudflare engineering	Previews, Workflows, rollback and AI Gateway guidance
Platform feedback	Dean’s supplied Cloudflare review prompt and brief

2026/11/20 Australia/Melbourne (AEDT, UTC+11) / 2026/11/20 UTC

Detailed talk tracks and source URLs are in each slide’s speaker notes.