



WHITE PAPER · PRODUCT STORY

CrowdRoll: Every Guest's Photos, One Album, No App

The best photos from a wedding, a reunion, or a trip are spread across every guest's phone. This is the story of the product I built to bring them home: a camera that opens from a QR code, keeps working without a signal, and develops the album when the host says so, engineered and verified the way I build client work.

Status: CrowdRoll is a new product in pre-launch testing. The live site at crowdroll.ptsnet.net runs in a test mode: automated screening of uploads is not switched on yet, paid plans are not open, and the site is not for real events until launch.

Every number in this paper comes from automated tests and load tests run on local hardware in October 2026. They are a baseline, not a production capacity figure; the same tests run again on staging servers before launch.

Every product here starts the same way client work does: with a real problem. This one is familiar to anyone who has hosted an event. Dozens of people take hundreds of photos, and most of them never reach the host. This paper tells how CrowdRoll answers that, what it does that the disposable-camera app I compared doesn't, and how it was built: written requirements, numbered decisions, independent reviews, and a test suite that runs on every change. The product is live for testing, so the result can be *tried*, not just described.

0

photos lost in local load tests:
1,000 of 1,000 shots from 250
simulated guests, 2,000 of 2,000
from 1,000

0.75 s

for the join page to be ready on a
throttled Fast 4G profile with a
phone-speed CPU (target: under 2
s)

1.6 s

from opening the camera page to a
working shutter on the same profile
(target: under 3 s)

2,077

automated tests at the last
milestone: 940 unit, 807 database,
and 330 browser tests

1. The photos that never make it home

The usual ways of gathering event photos all leak. A group text compresses the pictures and buries them under the conversation. A shared cloud album needs everyone to have the right account. A hashtag only works for guests who post publicly. And the leading disposable-camera app I compared requires guests to install it from the app store and then scan a second time. At a reception or on a tour bus, every one of those steps loses people, and every lost guest is a set of photos the host never sees.

So I set the product's first goal around that moment: a guest should go from scanning a QR code to taking a first photo in under 15 seconds, with no app and no account.

2. One scan, no install

CrowdRoll is a progressive web app. The camera runs in the phone's own browser, so the QR code on the table card is the whole onboarding:

- **Scan and type a first name.** Guests read a short join notice, type a first name, and the camera opens. There is no store, no password, and no account. The same phone comes back to the same guest and the same shots left.
- **A camera that feels like film.** Each guest gets a set number of shots, like a disposable camera, with a counter on screen ("17 left today"). Shots don't preview in the camera; guests see their own photos in My Photos at any time. Flip, flash, zoom, and a screen flash for the front camera are there where the phone supports them.
- **Photos already on the phone.** On paid plans, hosts can let guests add photos from their camera roll during the event, after it, or both, against a separate allowance.
- **Four kinds of events.** Personal events (weddings, parties, reunions), multi-day travel, organization staff events, and public organization events each start with their own settings: shot allowances per event or per day, when the album develops, whether location is recorded, and how moderation works.

The design answer: the guest at the party and the host at the desk need different things. Guest screens are built for one thumb on a small phone and pass automated accessibility checks at phone and desktop sizes, in light and dark mode. The host dashboard is built to run an event from a phone during the party and from a desk after it.

3. When the signal drops

Events happen in barns, on boats, in old stone churches, and in other places where phones lose their connection. CrowdRoll treats a lost signal as normal, not as an error:

- **The camera works offline.** After the first visit, a service worker keeps the app on the phone, so the camera opens with no connection at all.

- **Every shot waits safely.** Photos are saved on the phone and upload oldest first when the signal comes back, with automatic retries that back off and try again. A status chip tells the guest how many are still waiting.
- **Travel-friendly choices.** Guests can choose to upload on Wi-Fi only, and long trips offer a home screen shortcut that keeps the guest's photos and shots in place for the whole trip.

One honest limit: on iPhone, the browser does not upload in the background, so queued photos finish uploading when the guest opens the page again. CrowdRoll says so on screen instead of hiding it.

4. The reveal

Half the fun of a disposable camera is the wait. The host chooses when the album develops: at the end of the event, at a set time, a number of hours after the event ends, every evening of a trip, live, or by hand. Public organization events show a photo only after a moderator approves it.

The host sees every photo as it arrives, along with a live count of photos, guests, and uploads still on the way. From the dashboard the host can moderate, post a short announcement to every guest's camera screen, run a live slideshow on a TV or projector from a display link that can be turned off at any time, pick highlights, add alt text, and download everything as a ZIP file with a list of who took each photo and when. The goal is that every event ends with the host owning the photos outside CrowdRoll. Photos stay on CrowdRoll for 30 days on the free plan and 12 months on every paid plan, and the host gets email reminders before anything is deleted (30, 14, and 3 days ahead on paid plans), each with a link to the download.

5. Built like client work

CrowdRoll is my product, but I built it the way I build for clients, because that is how you get software you can trust with other people's photos:

- **Written requirements with IDs.** A product requirements document gives every rule an ID (CAM-3 for photo processing, OFF-2 for the offline queue, SEC-12 for tenant isolation). Code, tests, and reviews point back to those IDs.
- **A numbered decision log.** Every choice that shaped the build, from storage to load testing to pre-launch test mode, is recorded with its reason and consequences.
- **Milestones with a definition of done.** Seven milestones, each planned, built in small commits, and demonstrated. Nothing is done until typecheck, lint, and tests pass, new endpoints have authorization tests, and new screens pass axe-core accessibility checks with zero violations and work by keyboard at 320 px.
- **Independent reviews.** The guest camera milestone and the reveal and gallery milestone each went through two separate code reviews before sign-off. They found 9 and 10 issues, and all were fixed before the milestone closed.
- **Tests that look for trouble.** A cross-tenant suite calls every endpoint with one event's guest, host, and display credentials against another event's data, and every call must be refused. Load tests found real bottlenecks, which were fixed: a photo reservation that made five database calls now makes two, and a join limit that would have let only 10 guests a minute in from a shared venue Wi-Fi address was raised and retested.

6. Privacy and safety as features

A shared camera at a public event is also a way to upload something harmful. I treated trust and safety as part of the product from the first milestone, not as a policy page added at the end. Some of it is in place today; some of it waits for launch, and I want to be plain about which is which.

In place today:

- **No ads, no trackers.** Guests join with a first name. There is no account, no app store, no ads, and no ad trackers.
- **Private by default.** Photos are stored privately and served only through short-lived signed links issued after an authorization check. Location and camera details are removed from every shared copy: the gallery, downloads, and the slideshow.
- **Guests stay in control.** Guests can see their own photos at any time, delete them before the reveal, and ask the host to remove them after.

- **A removal form anyone can use.** A public form, with no account needed, hides the reported photo right away along with exact copies, and tracks a 48-hour removal deadline.
- **Rate limits, checked by a test.** Every route has a limit or a recorded reason for being exempt, and a test fails if a new route has neither.

Waiting for launch: automated screening of every upload for known child sexual abuse imagery, and a nudity classifier, are built behind a provider interface but are not switched on while the site is in test mode. Launch is blocked until they are. The terms of service and privacy policy are placeholders until an attorney reviews them, and a manual screen reader audit (VoiceOver and TalkBack) will follow the automated accessibility checks.

Why the test mode exists: the live site needed real phones on a real network before the screening provider was approved. Rather than quietly relax the rules, one switch turns on a pre-launch mode, shows a “Pre-launch test site” banner on every page, and marks every photo that could not be screened so it can be scanned once screening is on.

7. What the measurements show

The load tests simulate guests doing exactly what the real upload queue does: reserve a slot, send the photo and thumbnail straight to storage, confirm, and retry with the same photo ID after any failure. A separate check then confirms that every completed shot is stored, logged, and marked ready.

MEASURED RESULTS · LOCAL TEST RUNS, OCTOBER 2026

Zero lost photos, with the limits stated

Measure	Target	Result on local hardware
Lost photos, 250 guests	Every completed shot is stored	Pass: 1,000 of 1,000 shots ready
Lost photos, 1,000 guests	Every completed shot is stored	Pass: 2,000 of 2,000 shots ready; 633 temporary errors, each recovered by the automatic retry
Upload slot time, 250 guests	95th percentile under 500 ms	Pass: 93 to 270 ms on a quiet machine
Upload slot time, 1,000 guests	95th percentile under 500 ms	Pass at 88 ms when guests arrive over 10 seconds. Not met at 4.3 s when all 1,000 arrive within 2 seconds, limited by the local database API; to be rerun on staging
Joining from one venue address	Every guest gets in	Pass: 250 of 250 over 10 seconds, and 1,000 of 1,000 over 60 seconds
Join page on Fast 4G	Under 2 s	Pass: 0.75 s with a phone-speed CPU. On Slow 4G: 2.24 s, reported but not a target
Camera ready on Fast 4G	Under 3 s	Pass: 1.6 s with a phone-speed CPU. On Slow 4G: 3.5 s, reported but not a target
Automated tests	940 unit, 807 database, and 330 browser tests (72 end-to-end, 106 accessibility, 149 isolation, 3 performance), green on two consecutive runs with no retries	

These runs used one desktop for the app, the database, and the load generator, so they show where the limits are, not what production will handle. They also leave out the production rate limiter’s network round trip. Rerunning the same scripts on staging, with that rate limiter configured, is a launch requirement.

8. The CrowdRoll path

1. **Create.** The host signs in, picks an event type, and reviews the settings it fills in: shots per guest, when the album develops, uploads from the camera roll, location, and moderation.

2. **Share.** The host prints the table card or shows the QR code full screen. A short link sits under the code for anyone who can't scan.
3. **Shoot.** Guests scan, type a first name, and shoot in the browser. Photos queue on the phone and upload when they can, signal or not.
4. **Reveal.** The album develops on the host's schedule, the host moderates from the dashboard, and the slideshow plays on the big screen.
5. **Take them home.** The host downloads every photo as a ZIP with a list of who took what and when, and the photos leave CrowdRoll on the host's terms.

Sources

- Parker Technology Solutions: CrowdRoll product requirements document, version 2.0, including the comparison of an existing disposable-camera app's site, terms, and privacy policy (October 20, 2026).
- Parker Technology Solutions: CrowdRoll load test and performance budget report, local runs with k6 and Playwright (October 21, 2026).
- Parker Technology Solutions: CrowdRoll milestone records and decision log, including test counts and review findings, milestones 1 to 7 (October 2026).

See it, then picture your event

Visit the product site on your phone, see how the camera, the offline queue, and the reveal work, and send me a note about the event you have in mind. It is a pre-launch test site, so it is for a look around, not for a real event yet.

crowdroll.ptsnet.net · kparker@parkertechsolutions.com

© 2026 Parker Technology Solutions, Inc. CrowdRoll is a new Parker Technology Solutions product in pre-launch testing; the live site shows the current build and will change before launch. Automated upload screening, paid plans, and attorney-reviewed terms are not yet in place, and the site should not be used for real events until launch. Performance and load figures come from local test runs on one desktop computer, not from production servers.