Bioconductor’s web and package infrastructure

Sustainable open research infrastructure: lower cost, less to run, faster to change

Sean Davis
BIOCONDUCTOR TECHNICAL ADVISORY BOARD

2026-10-01

Where we are, in one slide

- **bioconductor.org has been served by the new stack since 2026-09-28.** It's a Cloudflare Worker in front of R2 object storage. `BiocManager::install()` works unchanged: 8 of 8 acceptance checks pass for release 3.23 and devel 3.24.
- **Why we moved:** crawlers overwhelmed one EBS-backed VM, and nobody could see them.
- **Cost:** about $5,000 a month of AWS is being retired. The new stack runs at roughly $240–500 a month (an estimate at list prices).
- **Packages** are still built by BBS and r-universe, and copied hourly from master. The registry built on r-universe is running, but it doesn't serve bioconductor.org yet.
- **Packages and the website are independent:** each publishes on its own path, and site changes go through PRs with previews and one-line rollback.

Speaker notes

(Opening, over this slide.) What I’d like out of the next half hour: the board knows what changed on bioconductor.org, why, what it costs, how it connects to r-universe, and what’s still open. Then four decisions I need help getting made (last slide but one).

Every number in this deck comes from the infrastructure docs site, seandavi.github.io/bioc-infrastructure, where each has a date and a source. Where a number is a projection or approximate, the slide says so.

One day on bioconductor.org

2026-09-30, UTC · FROM THE NEW EDGE'S REQUEST LOGS

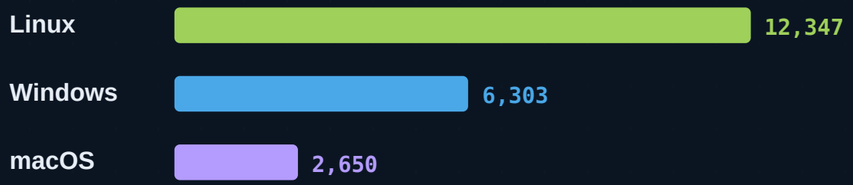
5.8M	8.1 TB	215	21,000
requests	served	countries	R installations

WHO IS ASKING · share of requests

R and package clients · other automation · browser-like · search + AI crawlers · mirrors · CI, monitoring, other

REQUESTS PER HOUR, UTC

R INSTALLATIONS BY OS



R 4.6 is 56% of them

One machine: 162,163 downloads of Bioconductor 3.19, 40% of R installs that day.

Build reports: 22% of all requests, the site's other front door.

Speaker notes

A dashboard view of one full day, the last complete UTC day before this talk, from the request logs the new edge keeps. Production hosts only (bioconductor.org and www).

How to read the numbers:

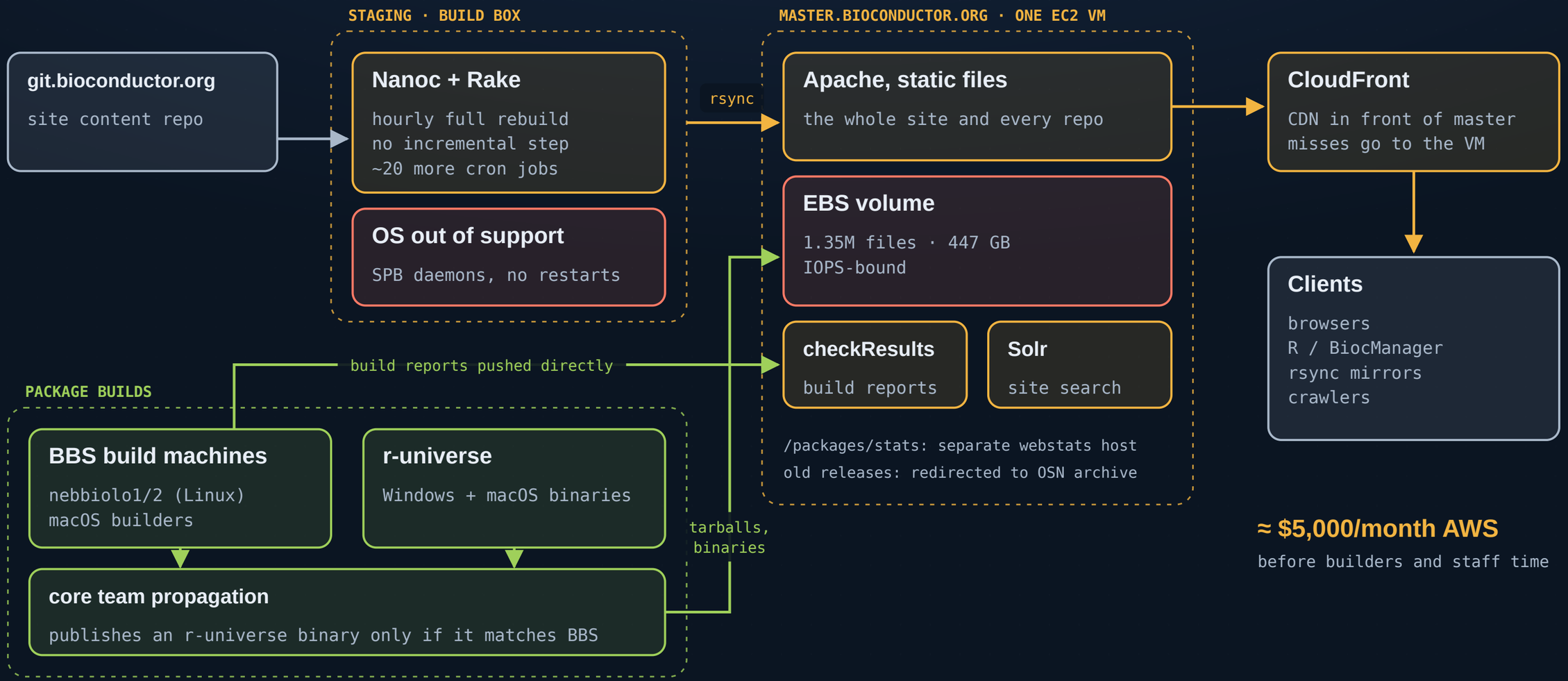
- “Clients” and “installations” are distinct hashed addresses, not people. One office behind one address counts once; one laptop on three networks counts three times.
- R installations: 20,966 distinct addresses sending R’s own user agent. About 24,800 addresses fetched `config.yaml`, which BiocManager reads on every call.
- The breakdown uses bioc-traffic’s first classification (v0), which is deliberately simple. “Browser-like” is what claims to be a browser; the top countries by distinct address there include Vietnam and Brazil, which look more like crawlers on home internet connections than people, so I wouldn’t present it as a count of humans.
- “Other automation” (v0’s `likely_automated`, 1.89M requests) is mostly machines in clouds pretending to be browsers. 53% is about 740 Google Cloud addresses whose browser user agent carries Google’s front-end marker `gzip(gfe)`; 37% is browser user agents from hosting and cloud networks (182,000 addresses; Google Cloud, RapidSeedbox, Datacamp, Azure, AWS, Alibaba). The other 10% is declared tools: HTTP libraries, curl, Python, headless browsers, small bots. Half of it reads build reports, 42% `/packages/`.
- 8.1 TB went out in one day, and 22% of those bytes came from the edge cache. The rest came from storage, where there are no transfer fees.

Factoids:

- One client downloaded 162,163 package files in a day, all for Bioconductor 3.19, which is two years old. That’s 40% of everything R clients downloaded. It is almost certainly a broken script or CI job reinstalling in a loop, and the legacy statistics would have counted every one as a download.
- Build reports (`checkResults/`) took 1.27 million requests from 242,000 addresses, a fifth of all traffic. Some of that is crawlers, but it’s why the raw-results listing (#43) matters.
- Linux is 59% of R installations, which fits CI, containers and servers; Windows 30%, macOS 13%. R 4.6, the current release, is 56%.
- The busiest hour was 09:00 UTC (388,000 requests), when Europe is at work and Asia’s afternoon overlaps.

The legacy estate

WHERE WE STARTED



≈ $5,000/month AWS

before builders and staff time

A package repository with a website attached: every `install.packages()`, every mirror, every build report and every page came off this one VM.

Speaker notes

Frame first: bioconductor.org is a package repository with a website attached. Every `install.packages()` reads `config.yaml`, `PACKAGES` and `VIEWS`; mirror operators rsync the whole tree; maintainers read build reports under `checkResults/`; renv and `install_version()` read `Archive/`. Page views are the smallest share of the load. For scale: about 6.8M requests a day on the first day of the new edge (crawler included), and 5.1 TB in 1.65M objects once loaded into R2 (the 447 GB docroot plus 4.66 TB of old releases).

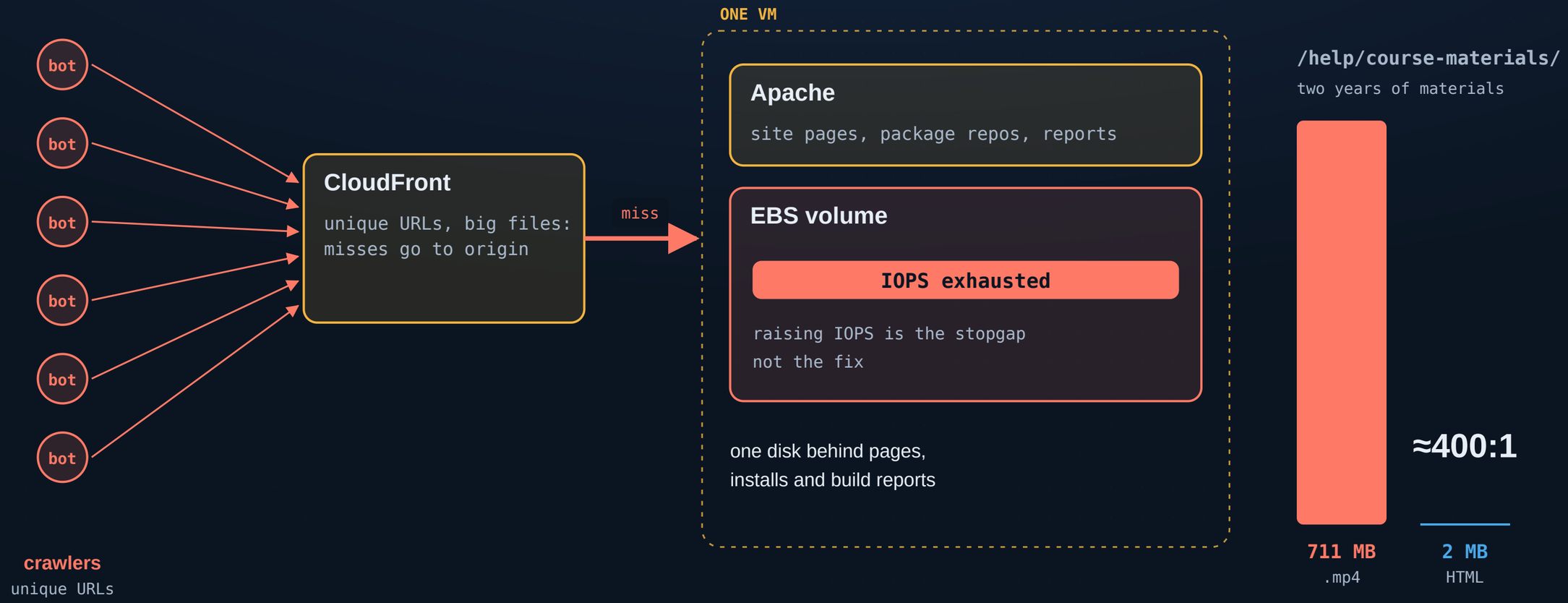
The diagram is drawn from the read-only recon of both boxes on 2026-08-03 (Current State page).

- staging is a build box, not a front end. It does a full Nanoc rebuild every hour with no incremental step, plus about 20 other cron jobs. It runs an Ubuntu release past end of standard support, and the Single Package Builder daemons are started from `@reboot` cron with no supervisor.
- master is one EC2 VM running plain Apache static files: the whole site, every package repository, the build reports, plus Solr for search. `/packages/stats/` lives on a separate webstats host.
- The builders push reports directly into master's `checkResults/`. Packages arrive through the core team's propagation: BBS builds the source tarballs, and r-universe binaries are published only when their version and commit match what BBS built.
- CloudFront is in front, so every cache miss lands on that one VM and its EBS volume.

The ~$5,000 a month is CloudFront, S3, master and staging. It doesn't count the builder hardware or the staff time to keep three operating systems patched.

Security hardening items found on staging were shared privately with the core team and are not in this deck.

The crisis: crawlers against one disk



Source: R2 migration, “Why”

Speaker notes

This is what started the project. The origin runs off an EBS volume, and crawlers exhausted its IOPS.

The requests that hurt aren’t page views. They’re bots pulling large files, like hundred-megabyte lecture videos from `/help/course-materials/`, and walking unique URLs that a CDN can never have cached. So they all go to origin. In that one directory, two years of materials are 711 MB of .mp4 against 2 MB of HTML, roughly 400 to 1.

Because it’s one disk behind everything, a crawler pulling videos slows down `BiocManager::install()` for everyone.

Raising EBS IOPS was the stopgap. It costs more every time and fixes nothing structural.

[Sean: add the dates and user-visible symptoms of the incidents you want to cite here; the docs site records the mechanism but not the incident dates.]

Why not just fix the VM?

- **One VM, one disk, behind everything.** Pages, installs, mirrors and build reports share one failure domain.
- **Nobody could see the traffic.** Request logs fed statistics jobs; nobody read them request by request. The download statistics have never filtered bots: the code that would has been commented out.
- **The build box is fragile.** Its OS is out of support, it does a full rebuild every hour, and its daemons run with no supervisor.
- **Most of the ~$5,000 a month is egress,** and sending tarballs to the world is the whole job.
- **Nothing was written down** in a place a newcomer could find it.

The decision

Move serving to object storage plus edge compute, and replace the legacy pieces one at a time behind the live site (a strangler migration), with every choice recorded as an ADR.

Speaker notes

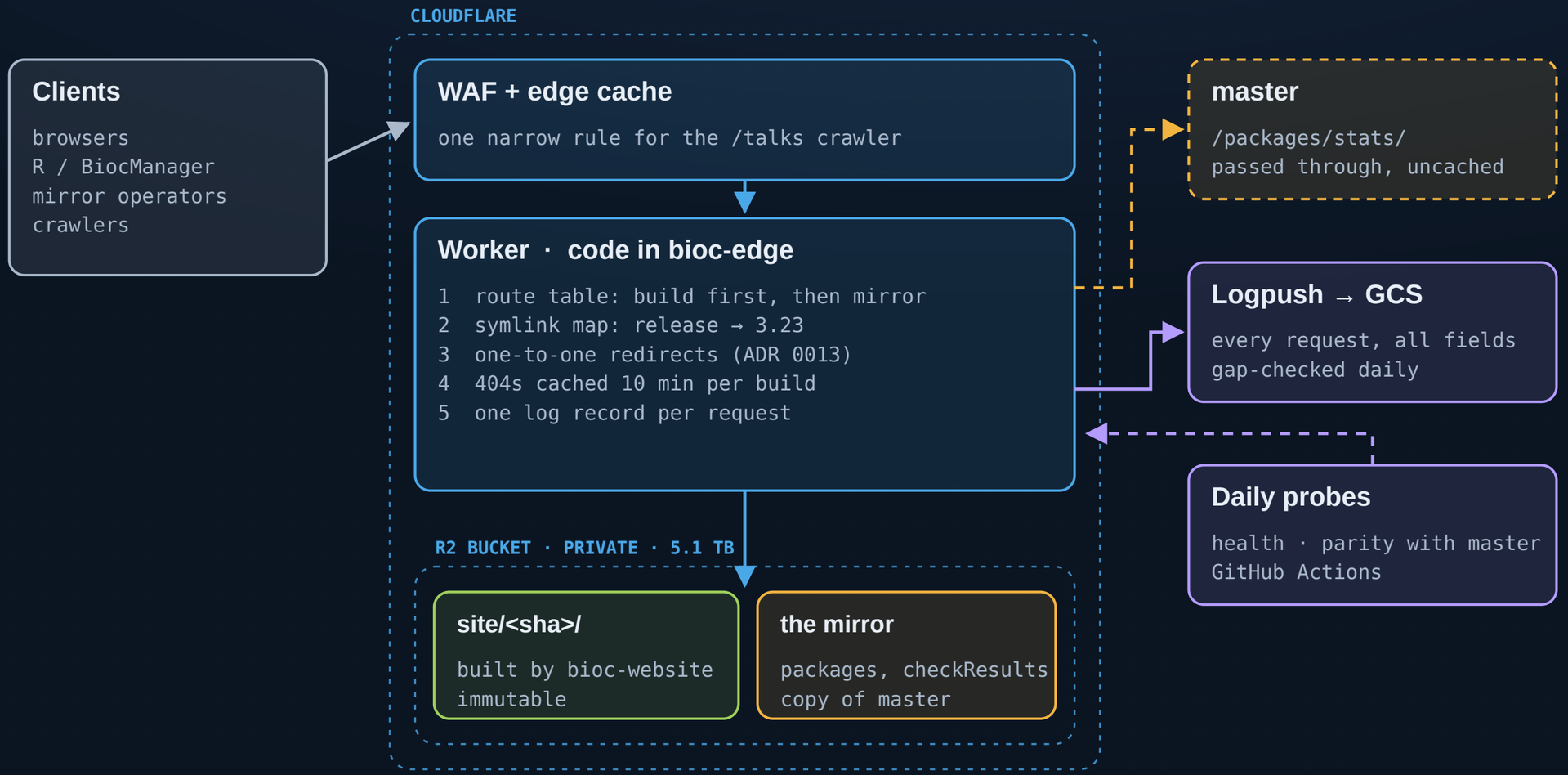
The board will reasonably ask why not a bigger VM or more IOPS. The answer is that the disk was the symptom.

- One failure domain: anything that stresses master stresses installs.
- No visibility: we found out on day one of the new stack that three quarters of traffic was one crawler (slide 12). The old logs almost certainly had it too; nobody was reading them that way. The download statistics page shows what that costs: August and September 2025 downloads roughly quadrupled with no change in distinct clients (CI and container rebuilds), and those months are still inflated in published tables.
- Egress is what a CDN bills for most, and a package repository is mostly egress. R2 charges none.
- Documentation: the docs site exists because none of this was written down.

The strangler approach is ADR 0008: three repos, a route table that says what the new build serves, and the old content as fallback until it’s replaced.

Serving: how a request is answered

THE NEW STACK



No servers.
No EBS.
No egress fees.

HTTP/2, brotli
four hostnames,
one Worker

Details: How bioconductor.org works · Edge cache

Speaker notes

Walk the request path:

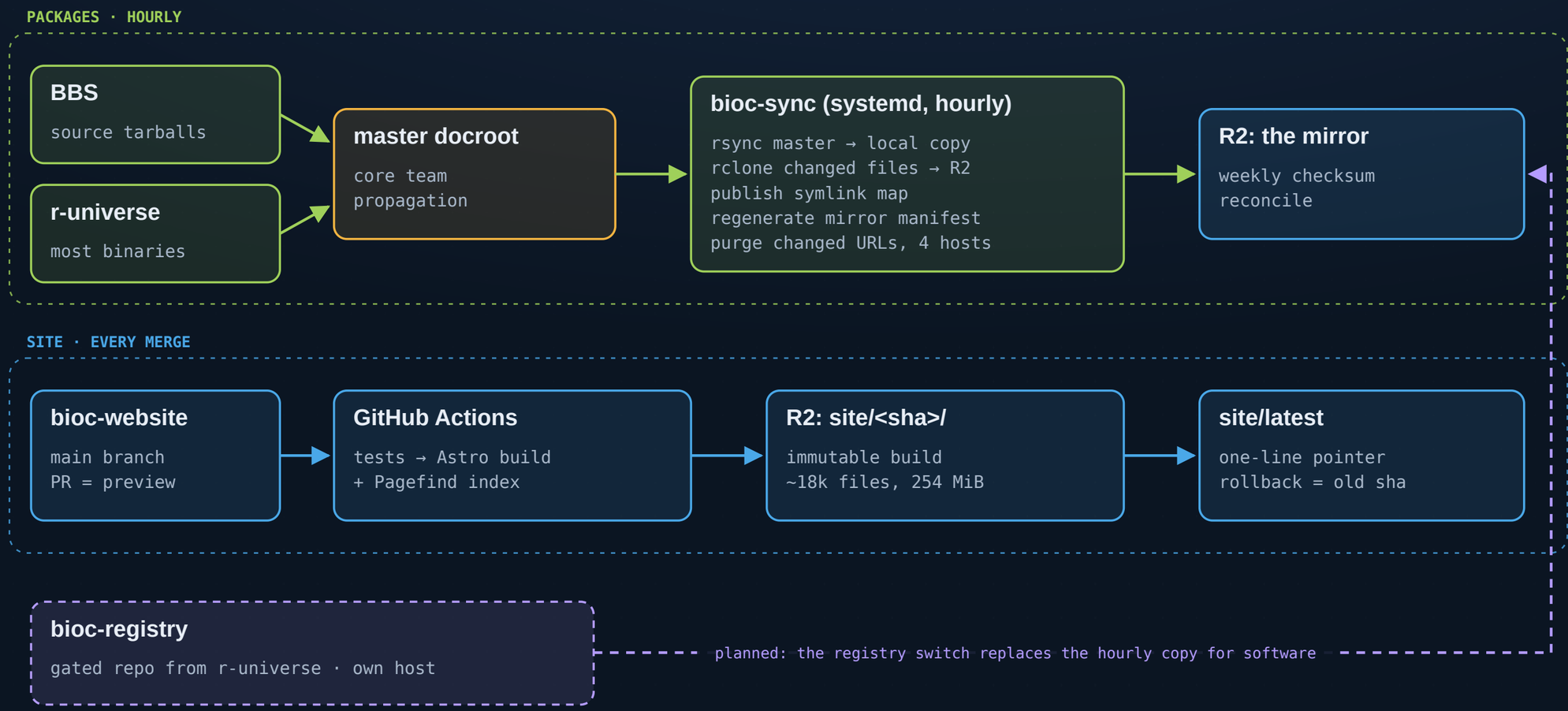
1. Cloudflare’s WAF, with one narrow rule (slide 12), then the edge cache.
2. The Worker, whose code and deploy config live in bioc-edge. (Its deployed name, and the R2 bucket’s, is `bioc-site`; that’s the name the Cloudflare dashboard shows. The pages themselves are built by bioc-website.) It applies the route table (the build first, then the mirror), the symlink map (object storage has no symlinks, so `release` → `3.23` is data the Worker reads), the one-to-one redirects, and 404s cached for 10 minutes keyed by build.
3. R2, a private bucket, 5.1 TB: bioc-website’s immutable `site/<sha>/` and the mirror of master’s docroot.

The one remaining dependency on master at request time is `/packages/stats/`, which is passed through uncached until statistics move to generated files (ADR 0004).

Four hostnames go to the same Worker: bioconductor.org, www, next., and the dev host used for PR previews.

The daily health probe and the daily parity comparison with master both run in GitHub Actions.

Publishing: how content gets there



Packages and the website are independent: neither waits for the other; they meet only in R2.

Speaker notes

Two independent paths, and that’s deliberate: packages and the website are separate systems that share only the storage. A package can be published, or pulled, without building the site; a site change can be merged, previewed or rolled back without touching a single package file. The same holds after the registry switch: it replaces the top lane, and the bottom lane doesn’t change. Neither goes through the registry yet.

Packages: a package update reaches bioconductor.org at the next sync after it lands on master, which is up to about 70 minutes (hourly, with up to 10 minutes of random delay). The sync purges exactly the changed URLs, on all four hostnames. A weekly job checks every object in R2 against the local copy by checksum.

Site: covered on the next two slides.

The prose still comes from Bioconductor/bioconductor.org through a data snapshot that is refreshed by hand. Pages are being ported one at a time; who owns the content long term is a core team decision.

The dashed violet line is the registry switch (slide 20).

Content management: the website as a PR workflow

the response header x-bioc-build names the build that answered

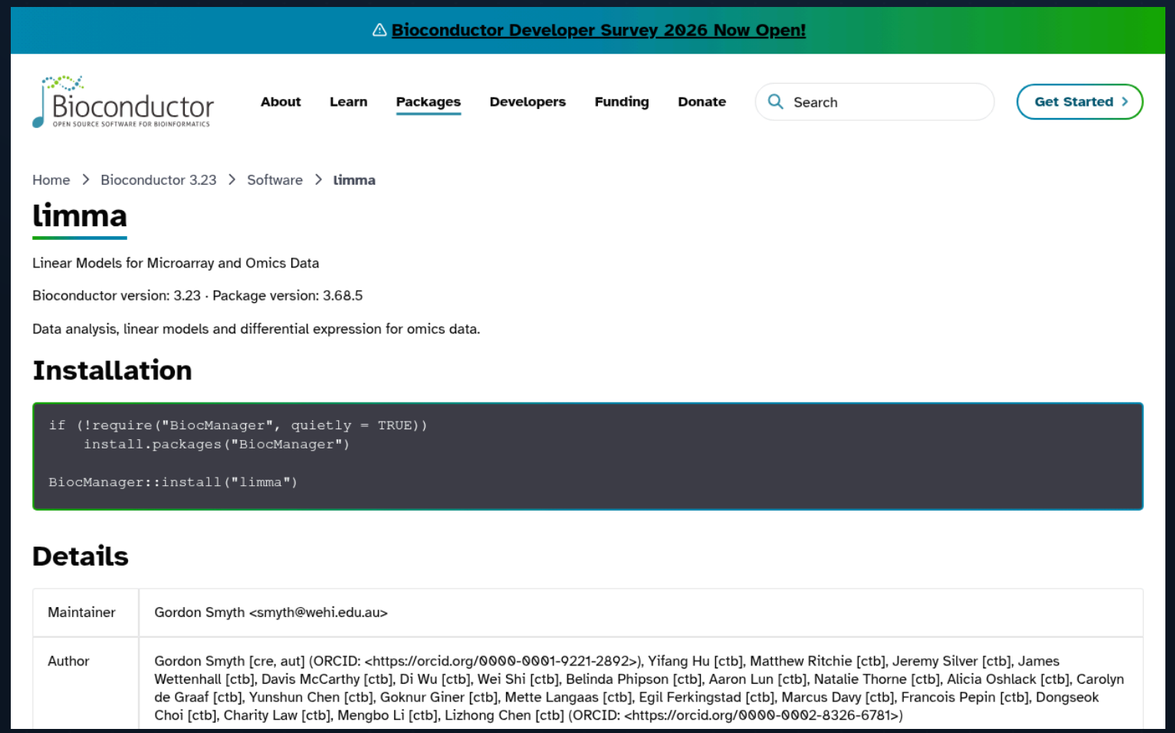
- **Every change is a PR,** with a preview
- **Every build is kept;** rollback is one line
- **Astro build: 27 s;** upload: ~18 min
- **Prose:** still from upstream, by snapshot

Speaker notes

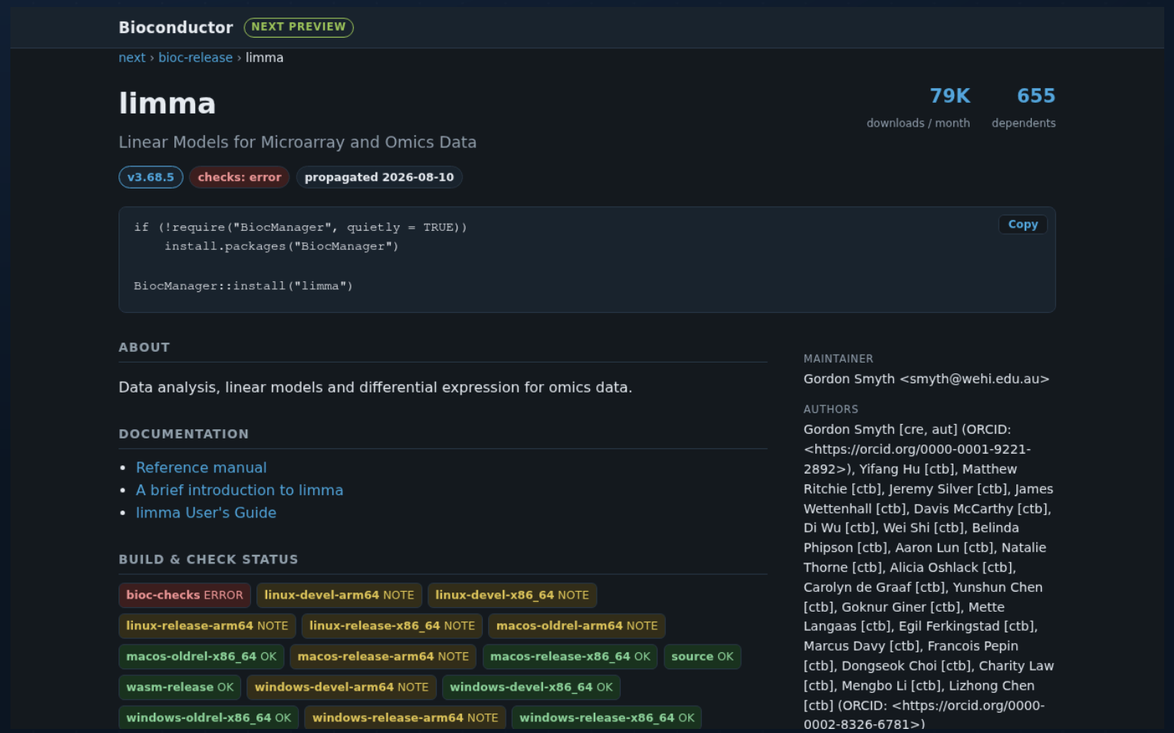
This replaces the hourly Nanoc rebuild on staging.

- PRs: every pull request to bioc-website builds the whole site, runs internal-link and accessibility checks on the templates, and posts a preview URL on the PR (`/_pr/<n>/` on the dev host). The preview is deleted when the PR closes. The caveat: previews need R2 credentials, so a PR from a fork doesn't get one yet. Fixing that is part of M3's goal of an outside contributor getting a preview unaided.
- Versioning: every build of `main` is kept in R2 as `site/<sha>/`, and the Worker serves whichever sha `site/latest` names. Rolling back is writing an older sha there, with no rebuild. Every response carries `x-bioc-build`, so you can tell which build answered.
- Speed, measured on the last main run (2026-09-29): the Astro build plus the Pagefind index took 27 seconds. The whole run took about 20 minutes, and 18 of those were uploading the build to R2: 17,787 files, 254 MiB. Uploading only changed files is the obvious fix; it isn't done yet.
- Content: the prose is still read from Bioconductor/bioconductor.org through a data snapshot refreshed by hand, so an upstream edit reaches the site only after a refresh. Pages are being ported into bioc-website one at a time, and every divergence from upstream is logged in UPSTREAM.md so it can be sent back. Who owns the content long term is a core team decision.

Restyling: bioconductor.org/next/



Today: from BBS, via master's repositories



/next/: from the bioc-registry data plane

Same package, **different data path**: `/next/` renders from r-universe builds through the registry's gate, with no dependency on the legacy AWS servers. New designs can be tried there in public without touching the live pages. Browse it

Speaker notes

`/next/` is a live, public preview area on the production site, built by the same Astro build as everything else, but from different data:

- Today's page (left) takes its version and download links from the repositories' own `PACKAGES` indexes, which BBS and the core team's propagation publish on master and the hourly sync copies to R2.
- The `/next/` page (right) renders from the bioc-registry data plane: the propagation index is the authority on versions, the observation archive on per-platform check status, and its download links point at the registry's own stored artifacts. Nothing there reads master or AWS. (Packages that fail in r-universe were seeded once from BBS builds, so that history traces back to BBS.)
- Downloads, dependents and stars come live from r-universe's API. But r-universe's download figure for Bioconductor packages is itself taken from Bioconductor's legacy stats pages (its API gives the source as bioconductor.org/packages/stats), so the "79K downloads / month" is the legacy, bot-unfiltered number, relayed. It is not a count of installs from r-universe; we don't see those at all today.

In most cases the two agree (limma is 3.68.5 in both), but not always: where a newer version fails in r-universe, the registry holds the older one (the registry-switch slide). So `/next/` is also a working preview of what serving from the registry looks like.

One inconsistency worth knowing if someone looks closely: the install box on `/next/` still says `BiocManager::install()`, which installs from bioconductor.org, not from the registry.

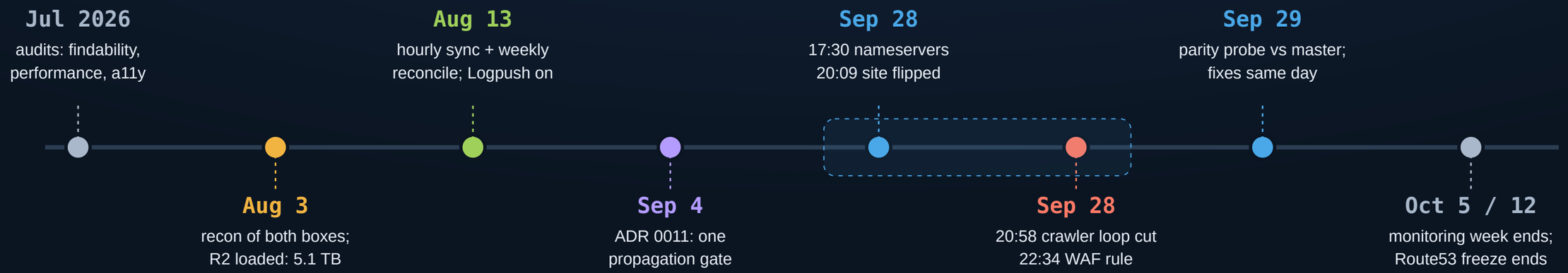
The page follows the reader's light or dark preference; the screenshot is the dark version.

The point for the board is the mechanism, not this particular design. A redesign is a PR: it gets a preview, it can be reviewed on the real site, and it can live under `/next/` until we're happy, then move over by changing a route. That's the M3 "prove the contribution loop" goal.

Two cautions if people go and look:

- The red "checks: error" on limma is BiocCheck (r-universe's `bioc-checks` job), which the gate records but treats as advisory. `R CMD check` is NOTE or OK on every platform.
- The numbers on `/next/` are preview data, and the download statistics have never filtered bots. I wouldn't quote them.

The cutover



- `BiocManager::install()` acceptance passed on release and devel **after** the flip
- Rollback stays one step away: Route53 is kept, with records frozen, until 2026-10-12

Speaker notes

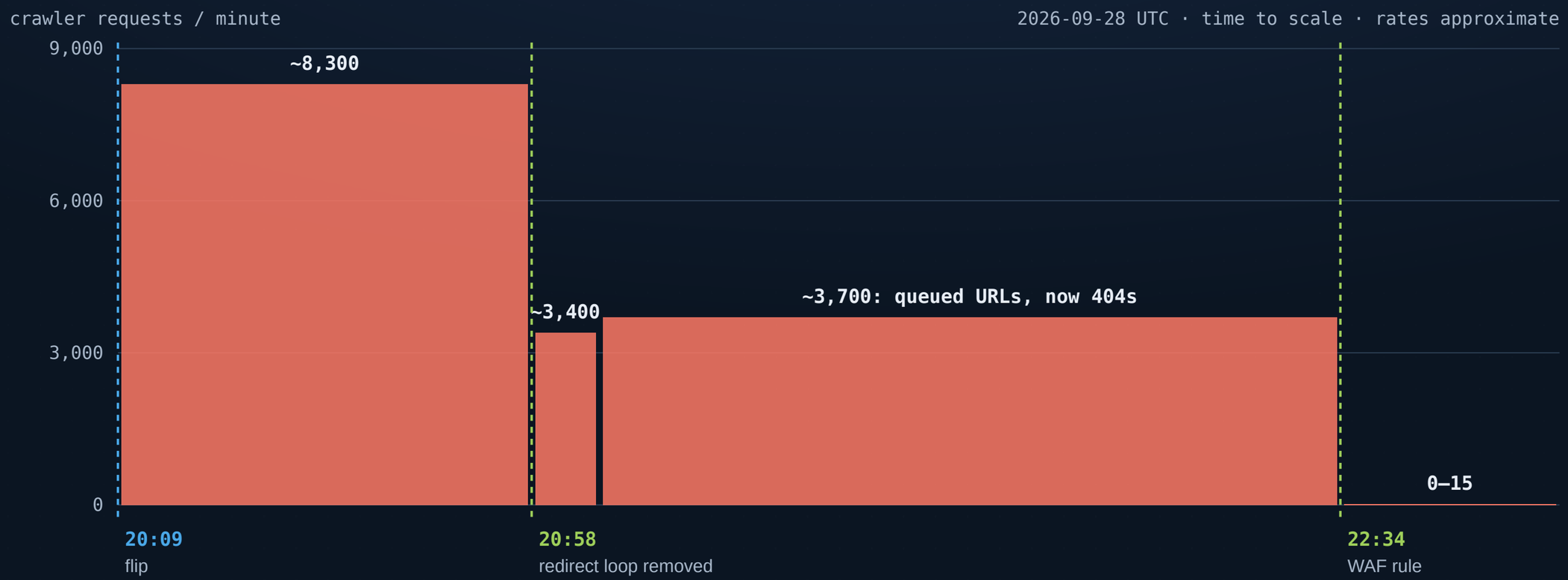
The migration ran for about two months before the flip, and the mirror was live and checked long before DNS moved.

- 2026-08-03: recon, and the R2 load of 5.1 TB, verified against the OSN archive with zero differences.
- By 2026-08-13: the hourly sync and weekly reconcile were running with failure alerts, and request logging was live.
- 2026-09-28: nameservers moved to Cloudflare at 17:30 UTC, and the site flipped at about 20:09 UTC.

The rollback is documented in the DNS runbook and on the Operations page. It is written, not rehearsed; the page says so.

The monitoring week ends 2026-10-05. M1 formally closes when the parity differences (slide 15) are fixed and the core team signs off.

Day one: three quarters of the traffic was one crawler



84,414 of the first 112,632 requests came from 1,130 addresses on two cloud networks, disguised as Mac Chrome. They fed a redirect loop: `/talks/*` sent everything to an index page full of relative links.

Speaker notes

We read the request logs within the hour of the flip. In the first 19 minutes, 75% of requests were one crawler: two cloud-hosting networks in Singapore, 1,130 addresses, rotating three Mac Chrome user agents.

The mechanism: the old site redirected anything under `/talks` to the course-materials index, and we had copied that rule. The index has hundreds of relative links. The crawler resolved them against the URL it had asked for, so every redirect minted hundreds of new `/talks/...` URLs, each redirecting back. No human used that redirect.

Fixes, both on 2026-09-28:

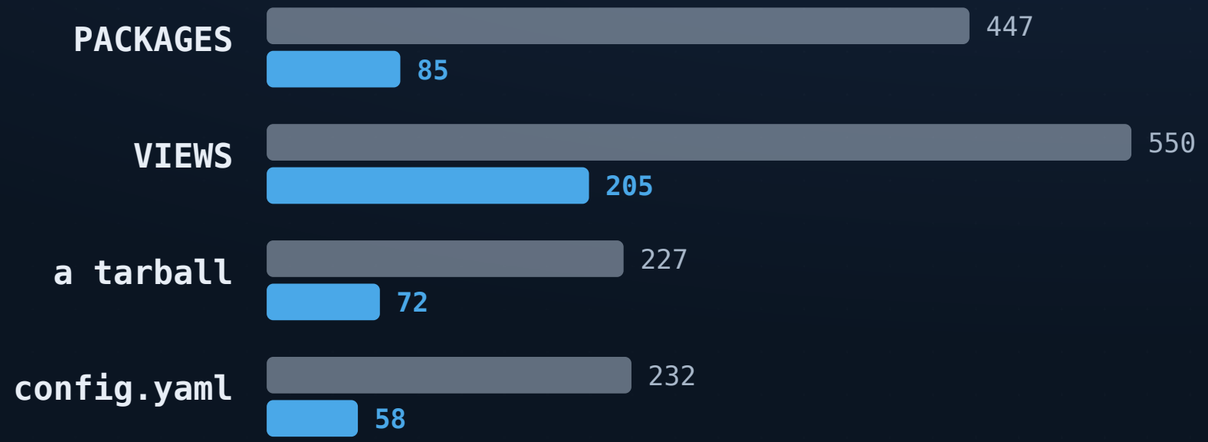
- 20:58 UTC: dropped the catch-all, so those paths now 404. That became ADR 0013: redirects land on content or they go.
- The crawler kept working through its queue at about 3,700 a minute, and each of those 404s cost storage reads. So at 22:34 UTC we added one WAF rule: those two networks, only for `/talks/`. The site has nothing there, so it can’t catch real content.

The old site almost certainly had the same crawler; we just couldn’t see it.

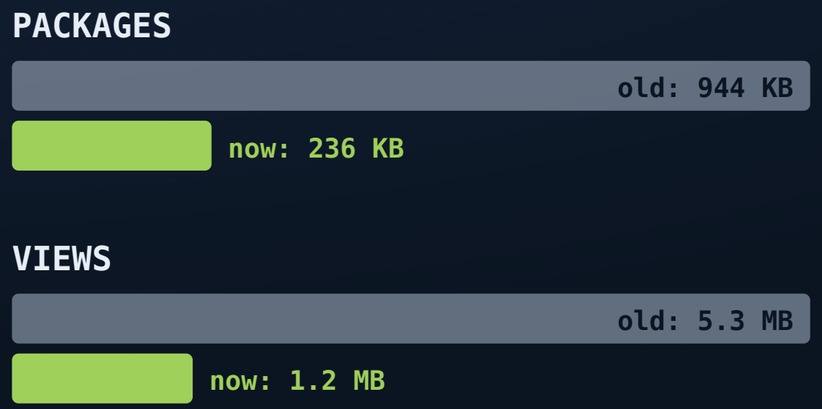
The rates are approximate (read from logs by the minute), and the chart’s time axis is to scale.

Faster, smaller downloads

time to first byte from Denver, ms (before → after the cache fix)



bytes on the wire, compressed



- HTTP/2 and brotli everywhere. The old site spoke HTTP/1.1 and sent the indexes uncompressed.
- 63% of 530,455 requests served from the edge cache (two hours on 2026-09-29)
- Still open: CloudFront was 30–50 ms faster from this host; part of that is Worker time

Speaker notes

Two separate wins.

Compression: `PACKAGES` goes over the wire at a quarter of its old size, and so does `VIEWS`. That matters most for people on slow or metered connections.

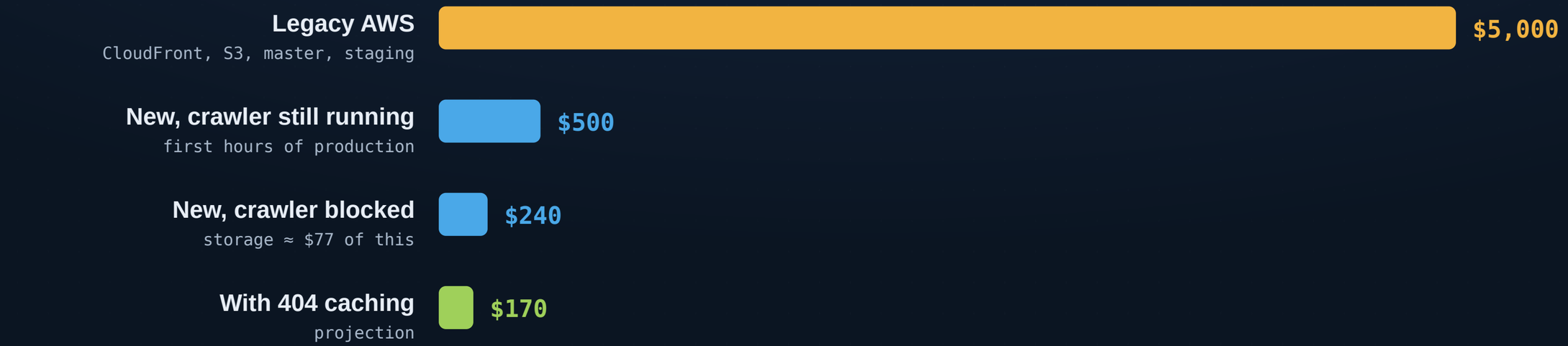
The latency chart is before and after a regression fix on the first evening (bioc-edge#35, fixed in #36). Package files were never being served from the edge cache, because a rule meant for build pages covered everything. These are single measurements from one host in Denver, not a global benchmark.

The honest remainder: from that host, CloudFront was still 30 to 50 ms faster. About a third of that is network distance; the rest is time in the Worker on every request. That’s the next performance item.

Cache hit ratio varies a lot by path: repository metadata is near 100%, package landing pages about 5% (a long tail that rarely repeats in the same data centre).

Costs

USD per month



Cloudflare list prices applied to measured traffic, not a bill. AWS figure excludes builder hardware and staff time.

Speaker notes

Two numbers that are not competing estimates of the same thing:

- ~$5,000 a month is the AWS estate being retired: CloudFront, S3, master and staging. That excludes builder hardware and staff time.
- $240 to $500 a month is the new stack, from first-day production traffic at Cloudflare’s list prices. Storage is about $77; the rest is per-request charges, mostly R2 reads, and the crawler caused most of those.

The difference is mostly egress, which R2 doesn’t charge for.

Two caveats. These are estimates, not a bill. And until M2 retires the AWS side, we pay for both.

What the parity probe found

We compared 13,271 paths from real traffic against master. Fixed the same day:

- 404s returned 9 bytes of text instead of the site’s 404 page
- Package files skipped the edge cache
- Double-slash links, range requests past a file’s end
- Landing pages had lost their Windows and macOS download links

The remaining edge cases (bioc-edge #43–#47) are small, tested code changes, because the web server is now code: routing rules, redirects and directory listings live in one program, with tests, not in server config.

Speaker notes

I want the board to see what went wrong as well as what went right.

The probe: sample paths from about 105,000 post-cutover requests (every 404, every redirect, a sample of the 200s), fetch each from bioconductor.org and from master, and compare status, redirect target, content type and content hash.

The most important open one is #43. Maintainers click “raw results” in a build report and get a 404, because object storage has no directory listings. There were 5,300 requests for those in 70 minutes, and it’s the first thing to fix.

The remaining ones: `checkResults` raw-results directory listings (#43), the container-binary redirect sending 301 where master sends 302 (#44), `/packages/` (#45), a soft-404 under versioned package paths (#46), and text files served as downloads (#47). Each is a change to the Worker plus a test. The Worker already renders directory listings from storage for `Archive/`, so #43 is mostly extending that. The point for the board: on the old server these were Apache configuration; now any routing rule, however specific, is ordinary code that’s reviewed, tested and rolled back like any other change. A sixth, PR previews answering on the production domain (#48), was fixed on 2026-09-30.

Master stays up until these are clean; that’s the gate for M2. `parity.yml` now runs the comparison daily.

The missing landing-page downloads were reported by Lori on 2026-09-29 and fixed the same day.

Where packages come from today

PACKAGES AND R-UNIVERSE

PACKAGE TYPE	BUILT BY	REACHES BIOCONDUCTOR.ORG VIA
Software, source tarballs	BBS	master, then the hourly sync
Software, Windows and macOS binaries	r-universe, published by core team propagation	master, then the hourly sync
Annotation, experiment data, workflows	legacy builders; no replacement running yet	master, then the hourly sync

r-universe already ships most binaries

The core team publishes an r-universe binary only when its version and commit match what BBS built. Across 2,326 packages, r-universe's `R CMD check` agrees with BBS's `nebbiolo1` **96.7%** of the time (counting WARNINGS as clean).

Speaker notes

Windows and macOS users already install r-universe builds; Linux, CI and containers install BBS source tarballs.

The 96.7% is from the legacy propagation page: 2,249 of 2,326 packages, if WARNINGS count as shippable; 94.4% if they don’t. That’s close, but not a drop-in substitute. The remaining few percent is exactly the risk surface of replacing BBS for software packages.

An earlier comparison showed only about 20% agreement, but it used r-universe’s bioc-checks job, which is BiocCheck, not `R CMD check`. Worth knowing if that number comes up.

bioc-registry: one gate for every producer

Speaker notes

The registry is the r-universe integration. It observes r-universe's bioc and bioc-release universes every 15 minutes, and decides which builds are fit to publish.

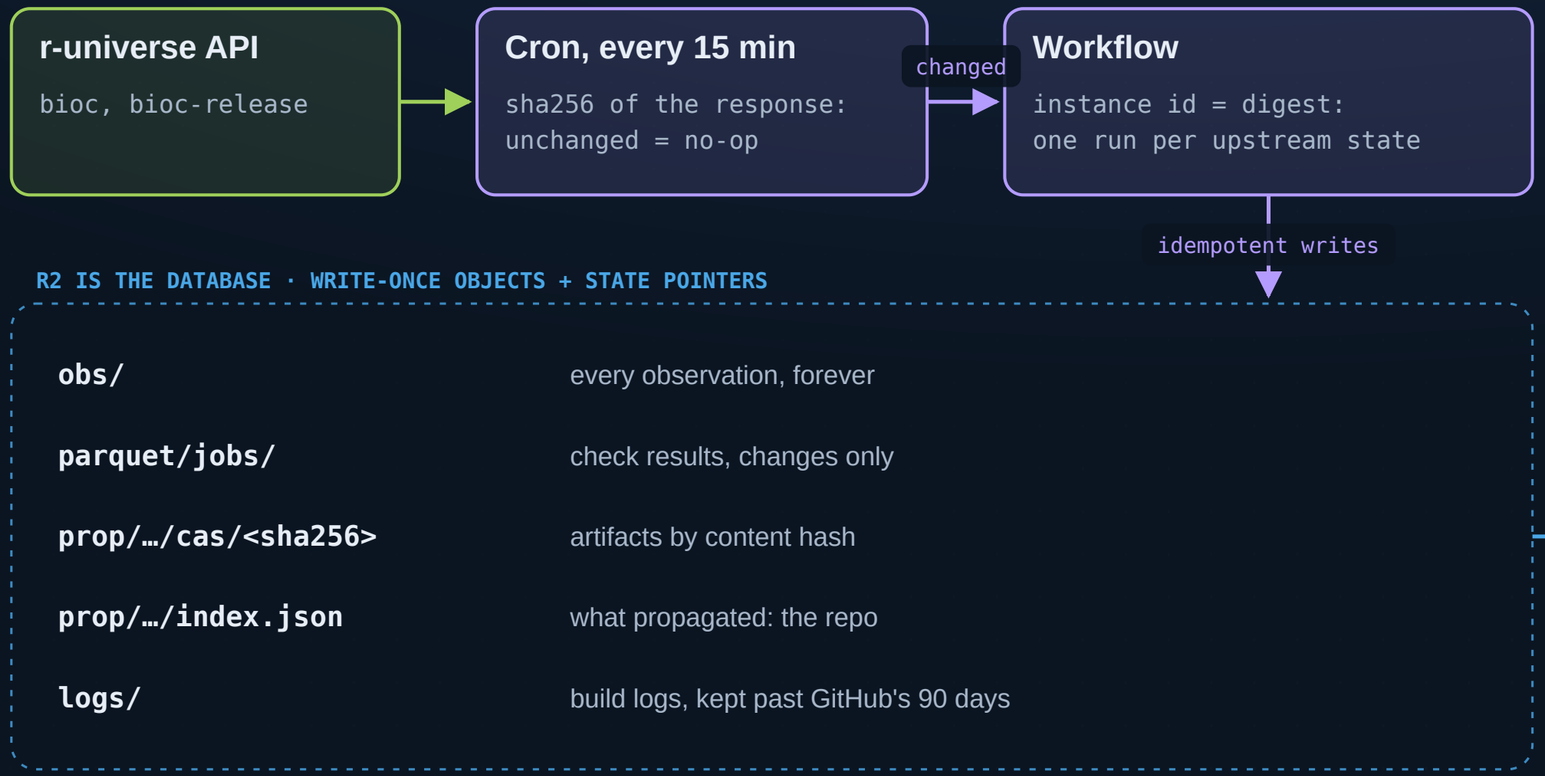
The gate (ADR 0011, one function, three callers):

- Build status is success.
- At least one platform family (Linux, Windows, macOS arm64) has no ERROR or FAIL on the R version Bioconductor uses for that release. NOTE and WARNING pass.
- The version is a strict bump. A push without a bump never propagates.
- bioc-build output must also match its entry in bioc-manifest.
- BiocCheck is recorded but advisory.

Everything is stored by content hash, with a ledger of decisions. Since 2026-09-29 (bioc-registry#50), each wave also re-evaluates published versions, so a binary that r-universe builds later gets added. That recovered 137 missing binaries.

It publishes a real, installable repository today, just not at bioconductor.org.

bioc-registry: serverless, with storage as the database



No servers
No database server
No queue to run

One Worker serves it all

```
/repo    installable repo
/data    any object, Range
/sql     DuckDB in browser
/pkg     history per package
/docs    OpenAPI
```

How it’s built (README) · SQL console · API docs

Speaker notes

The whole registry is one Cloudflare Worker, one R2 bucket, a cron trigger and a Workflow. No VM, no database server, no queue to operate.

- Change detection is a hash. Every 15 minutes the cron fetches r-universe's package list and hashes the response. Same hash, nothing happens. A new hash starts a Workflow whose instance id is that hash, so the same upstream state can never be processed twice, and a retry is always safe.
- R2 is the database. Almost everything is write-once: the raw observations (kept forever), the check results as Parquet, the artifacts keyed by their sha256, the build logs. Only small state pointers are rewritten. The check-results Parquet stores changes, not snapshots, because about 99.9% of consecutive rows are identical.
- Queries happen where the reader is. `/data` serves any object with range requests and CORS, so DuckDB (in R, Python or the browser) can query the Parquet over HTTP without downloading it. `/sql` is a DuckDB-WASM console in the browser over the whole archive, and the per-package pages use the same trick for their history.
- Build logs are captured, so they're still readable after GitHub expires them at about 90 days.

The same Worker serves the installable repository, so the data and the repository can't disagree: both are the same objects.

Using the registry today

```r
# A preview, not for production
reg ← "https://bioc-registry.seandavi.workers.dev/repo/bioc-release"
# devel: .../repo/bioc

install.packages("edgeR",
  repos = c(BioCsoft = reg,
            CRAN = "https://cloud.r-project.org"))
```

⚠ Not through BiocManager

`BiocManager::install(site_repository = reg)` adds the registry *next to* bioconductor.org, and R takes the highest version. BiocManager has no setting that points it at the registry instead.

Works (checked 2026-09-30)

- 2,390 release software packages
- Source, Windows and macOS indexes
- A clean install, at the registry's version

Not yet

- `PACKAGES.rds` (R falls back to `.gz`, so installs still work)
- `Archive/` for older versions
- Annotation and experiment-data packages: add those repositories yourself

Speaker notes

This slide answers “can I use it?” Yes, for software packages, with plain `install.packages()`.

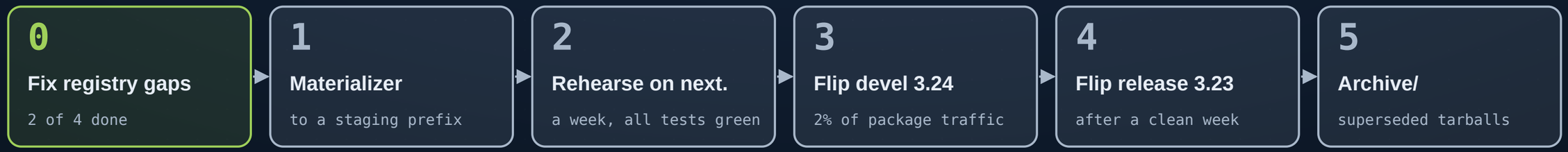
What I checked on 2026-09-30:

- The release repository lists 2,390 packages, and `PACKAGES` and `PACKAGES.gz` return 200 for source, Windows (`bin/windows/contrib/4.6`) and both macOS arm64 paths.
- BiocGenerics 0.58.1 installed from it into an empty library, matching the registry’s index.
- `PACKAGES.rds` is a 404 on both universes. R tries it first and falls back, so it costs a round trip rather than breaking installs. It’s the first blocking item on the switch plan (next slide).

On BiocManager: `site_repository` is meant for an extra repository, so it can’t replace BioCsoft. Making `BiocManager::install()` itself use the registry is really the registry switch: once bioconductor.org serves the registry’s tree at the usual paths, BiocManager needs no change at all. That’s why the plan puts it behind bioconductor.org rather than asking users to reconfigure anything.

If someone tries a package that needs annotation or experiment data, they need to add those repositories (from BiocManager::repositories() for the matching Bioconductor version) alongside.

The registry switch: a plan with tests



each step names its rollback · nothing advances until its tests pass

Compared file by file: 2,374 of 2,384 release source packages are at the same version.

Blocking: no `PACKAGES.rds`, the index R asks for first. Some packages have hard dependencies that can't be installed.

A decision, not a fix: 3 release and 8 devel packages lag. Their newer versions fail in r-universe, so the gate is stricter than BBS here.

Ten regression tests (T1–T10) run on every materialize and block a broken tree.

Speaker notes

Plan and evidence are on the Registry switch page, tracked in bioc-infrastructure#35.

What would regress if we switched today:

- `PACKAGES.rds` is missing from the registry. It's the most requested index: 2,784 requests against about 220 for the other formats, in 70 minutes.
- The registry adds 4 release and 7 devel packages whose hard dependencies can't be resolved. The fix is a gate rule, not a list.
- Lagging versions, for example biomaRt, DOSE and CNVRanger: every one is blocked by a gate rule, because the newer version fails to build or check in r-universe. Accepting that lag, or bridging those versions from BBS, is a policy choice for the core team.

The plan is staged: materialize to a prefix nothing serves, rehearse on next. for a week, then flip devel first (2% of package traffic), then release. Every step names its rollback.

Data, experiment and workflow packages

434
experiment-data packages, 28 GiB in total

67
of them hold three quarters of the bytes

425 / 433
changed size by under 1% across the last release bump

r-universe builds software packages but has a **100 MiB limit**, and many data packages are far bigger. Until something else builds them, the legacy builders can’t be turned off.

bioc-build is a proof of concept:

- a pinned copy of r-universe’s own workflow, on GitHub Actions
- a configurable size limit
- packages authorized by a reviewed manifest (bioc-manifest)
- dependencies resolved from the registry, published through the same gate

The 25 workflow packages fit in 319 MiB together.

Speaker notes

This is the stuck part of M2, and now it has a route.

We measured the corpus instead of guessing (the data packages page). 434 experiment-data packages total 28 GiB, 67 hold three quarters of the bytes, and they almost never change.

bioc-build diverges from r-universe only where it has to. Staying close means a package both systems build should produce the same artifact and the same check result, so we can compare them.

Status: a handful of successful builds as of early September. The build home for annotation, experiment and workflow packages is a decision for the core team and the r-universe track (roadmap notes, slide 22).

Roadmap

WHAT'S NEXT

M0

Operational readiness

public repos, CI, badges
daily health probes

mostly done

→ deploys still from a laptop

M1

Production cutover

flipped 2026-09-28
BiocManager 8/8 on 3.23, 3.24

flipped; closing out

→ parity fixes bioc-edge#43–48

M2

Full disconnect

retire master, staging, BBS
legacy stats, ~$5k/mo AWS

next

→ registry switch; data-pkg builds

M3

Search + redesigns

Pagefind search ADR
redesigns through PRs

can start now

→ prove the contribution loop

M2 needs: parity with master clean (bioc-edge #43–48) · the registry switch · a build home for data and workflow packages · download statistics from the new logs

Speaker notes

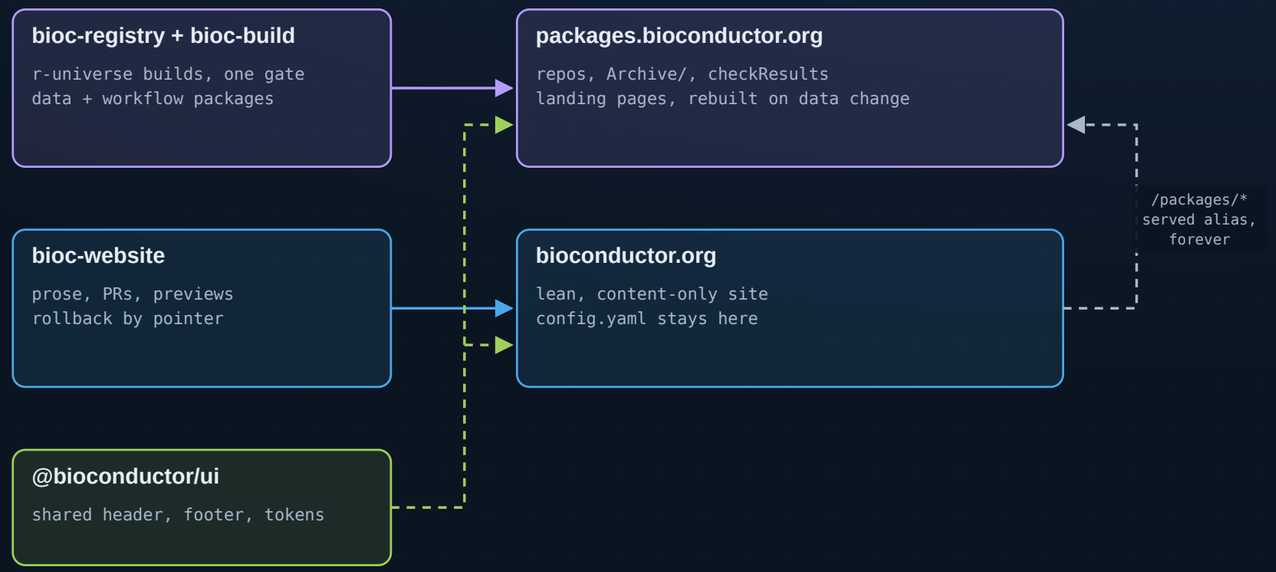
M1 closes with the parity fixes and the core team’s sign-off. M2 is the real disconnect: retire master, staging, the legacy build system and the legacy stats, and stop paying about $5,000 a month for AWS. It needs the registry switch, a build home for data packages, and download statistics generated from the new logs. M3 has no blockers left; production search is already Pagefind, not Solr.

Decisions that need owners, if the board asks (all listed under “Decisions pending” on the Operations page):

1. A backup operator, and an owner for a Bioconductor Cloudflare account.
2. Registry switch policy: accept that versions failing in r-universe wait, or bridge them from BBS?
3. A build home for annotation, experiment and workflow packages.
4. M1 close-out: core team sign-off, the security findings walk-through, and the `www.` / `secure.` hostname decisions.

Where this could go: two sites

UNDER DISCUSSION, NOT DECIDED



- **packages.bioconductor.org** for everything package-shaped; **bioconductor.org** becomes a lean content site
- **bioconductor.org/packages never breaks:** BiocManager hard-codes it, so it stays as an alias
- **One look across both** from a shared Astro package (discussion: #71)

Speaker notes

Where the architecture points once M2 is done. It’s a discussion issue (#71), not a plan, and I’d like the board’s reaction.

- packages.bioconductor.org would hold everything package-shaped: repositories, Archive, build reports, landing pages, the registry’s API. It follows from the registry switch and from splitting the storage buckets (#70), so the website’s credentials can never touch a package file.
- The one hard rule: `bioconductor.org/packages/...` keeps working forever. BiocManager builds every repository URL from `bioconductor.org` and reads `config.yaml` from there, and so do renv lockfiles, mirrors and papers. So it’s a served alias, not a redirect.
- Landing pages stay static, because the long tail makes a cache nearly useless (5% hit rate on landing pages in the logs). Instead of rebuilding the whole site, regenerate just the pages whose package changed when the registry propagates.
- Shared look: Astro components are ordinary modules, so a header, footer and design tokens can be one package both sites import.
- Contributors: previews from forks would close the loop M3 is about.

Order: after the registry switch, and DNS can’t change before 2026-10-12.

One data layer for Bioconductor

WHERE THIS COULD GO

Status, usage, impact and community, joined on the package, open to people **and** agents. Explore today: bioc-intelligence dashboard · bioc-registry API docs

Speaker notes

The idea: Bioconductor already has the pieces of a complete picture of every package, in separate places. The registry knows what built and passed on every platform. The edge logs know who actually uses a package: people, CI or bots. bioc-intelligence links packages to the papers that use them, their citations, and the grants that paid for them. bioc-contrib-intelligence holds the submission and review history from the Contributions trackers. Add community activity (commits, pull requests, maintainers) and you have status, usage, impact and community, all keyed on the package.

Put that in one open, versioned data layer and everything else reads from it:

- package pages that are always current, instead of a snapshot;
- dashboards for the community and funders;
- an API for tools, mirrors and other projects;
- an MCP server, so a person or an AI agent can ask “which packages did this grant support, and how are they used?” or “what cites limma, and does it build on Windows?” in one step;
- the numbers in grant renewals and board reports, from the same source as the website;
- events and hooks: a new version propagates, a build breaks on one platform, a submission gets a review. Agents can subscribe and act, for example triaging a failure, drafting a note to the maintainer, or checking a submission against past reviews.

The pieces exist to different degrees today; bioc-intelligence’s dashboard is a working example of the impact side. If the board wants one principle from this slide: this data should be open by default, with the obvious exceptions (raw client addresses stay private, per ADR 0012).

Acknowledgements

- **Andres Wokaty** and **Lori Kern**: major help with the website and the r-universe builds
- **Hervé Pagès**: explaining the build system and the propagation logic
- **Jeroen Ooms** and **rOpenSci**: r-universe, which most of this stands on

And the **many people who built and ran Bioconductor's infrastructure over the last 25 years**. Everything here is a migration of their work, and it only worked because that work was there to learn from.

Speaker notes

Thank people by name. Andres and Lori helped throughout with the website and the r-universe builds. Hervé walked me through how BBS and propagation actually work, which is what the propagation gate is modelled on. Jeroen and rOpenSci built r-universe, which builds most of what Bioconductor users install on Windows and macOS today.

And the people over 25 years who built the system this replaces. None of this was possible without their work to learn from.

Materials

Docs site: seandavi.github.io/bioc-infrastructure

- How bioconductor.org works
- Operations: runbooks, decisions, risks
- What the new site has changed
- Roadmap
- Registry switch plan
- The propagation gate
- Architecture diagram
- Reporting a problem

Explore

- bioc-registry API docs
- bioc-intelligence dashboard

Repositories

- bioc-edge: the Worker and the sync
- bioc-website: pipeline and Astro build
- bioc-registry: r-universe observer and gate
- bioc-build and bioc-manifest: data and workflow packages
- bioc-infrastructure: docs, ADRs, roadmap issues
- bioc-intelligence: impact data and dashboard

Decisions: the ADRs

Speaker notes

Everything here is public. The docs site is the place to start; “How bioconductor.org works” is the one-page version of this talk. Problems go through the issue forms linked from “Reporting a problem”, and each report is triaged by hand.