Building a new startup on Rails and Starlink



Cole Robertson

CTO & Co-founder

Rebulk · YC S25



SF RUBY CONFERENCE · SFJAZZ · NOV 11, 2026



Upstate New York. January.



A shed in upstate New York. No internet.


THE NORTH FACE

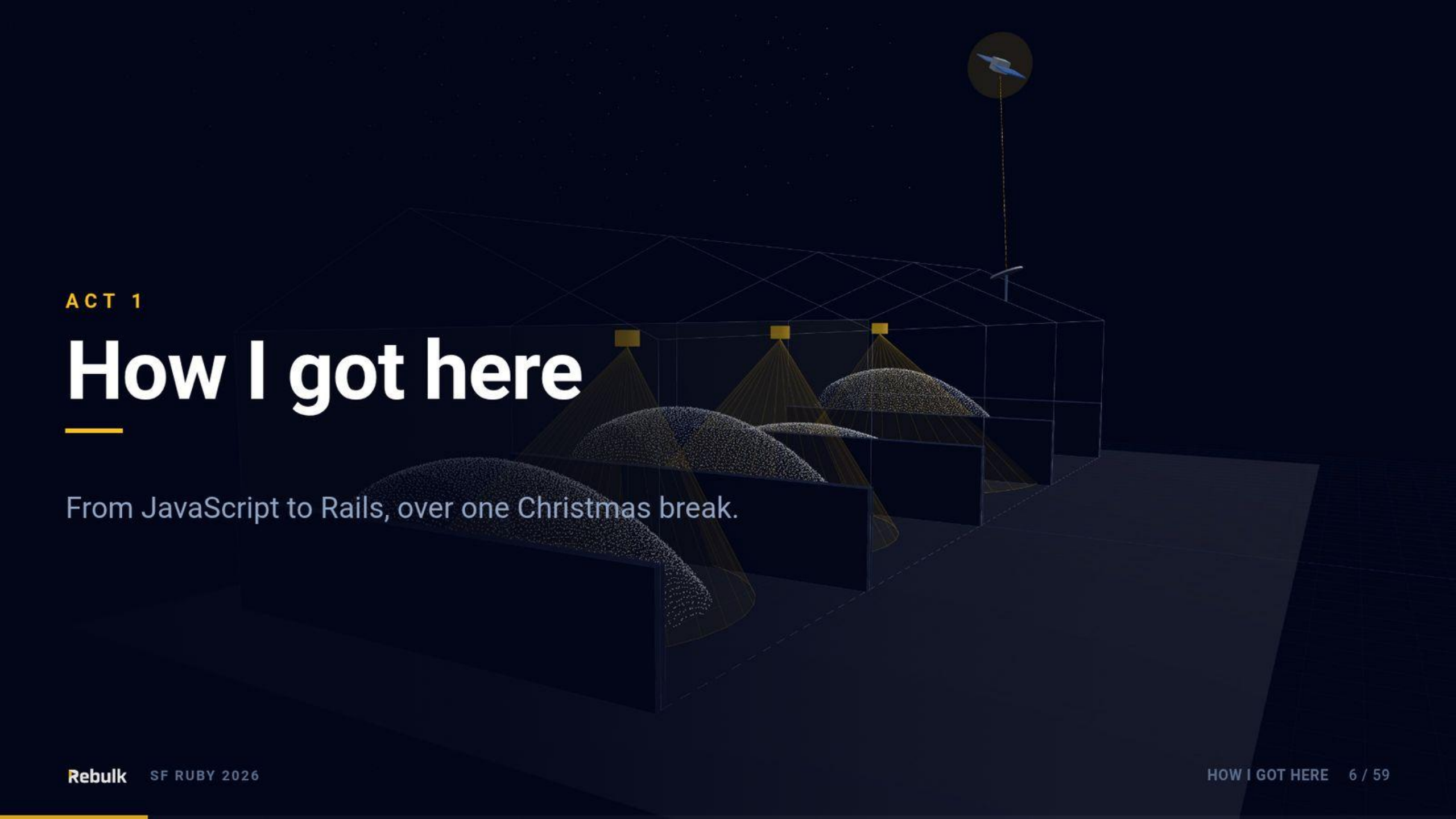


Our internet.

ACT 1

How I got here

From JavaScript to Rails, over one Christmas break.



Always looking for the fastest way to ship

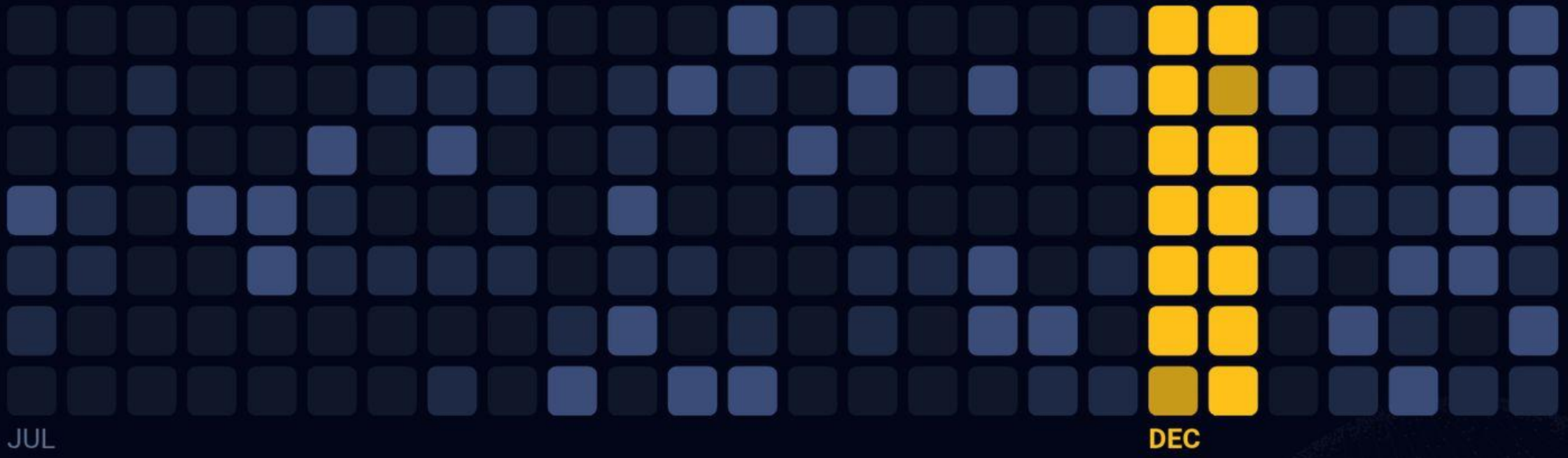
JavaScript → React → Next.js → **Blitz.js**

"The Rails for JavaScript."
I wanted Rails. I just didn't know it yet.

"We're on Next.js, and thinking about moving to Blitz."

A job post on a forum. I joined about two months in.

Christmas break



CTO

Hey, can you talk?

CTO

Hey, can you talk?

CTO

Over the break I rewrote the whole app. Front end and back end. In Rails and Hotwire. I know this isn't what you signed up for.

CTO

Hey, can you talk?

CTO

Over the break I rewrote the whole app. Front end and back end. In Rails and Hotwire. I know this isn't what you signed up for.

ME · NEVER WRITTEN RUBY

This is awesome. Let's do it.

One person rewrote months of work over one holiday.

That's what a framework is for.

The 2026 default stack

Next.js framework	**Vercel** hosting	**Supabase** database	**Clerk** auth	**Prisma** ORM
Upstash Redis	**Inngest** jobs	**Pusher** websockets	**Resend** email	**Sentry** errors

10 vendors

10 bills

10 dashboards

10 status pages

Frameworks matter more when an LLM writes the code.

Rails gave us every primitive from a Jetson in a shed to a per-customer database, so we could spend our time on the business. And stay calm.

ACT 2

Starting Rebulk

A question every bulk site has: how much is in that pile?





How much is in that pile? Take a guess.

Rebulk







The problem

Remote

bad cell, sometimes none

Dusty and dark

a loader driving around

Billed from it

invoices and shipments

Has to keep working

with or without internet

Day one






Hardware. Lasers. A shed.

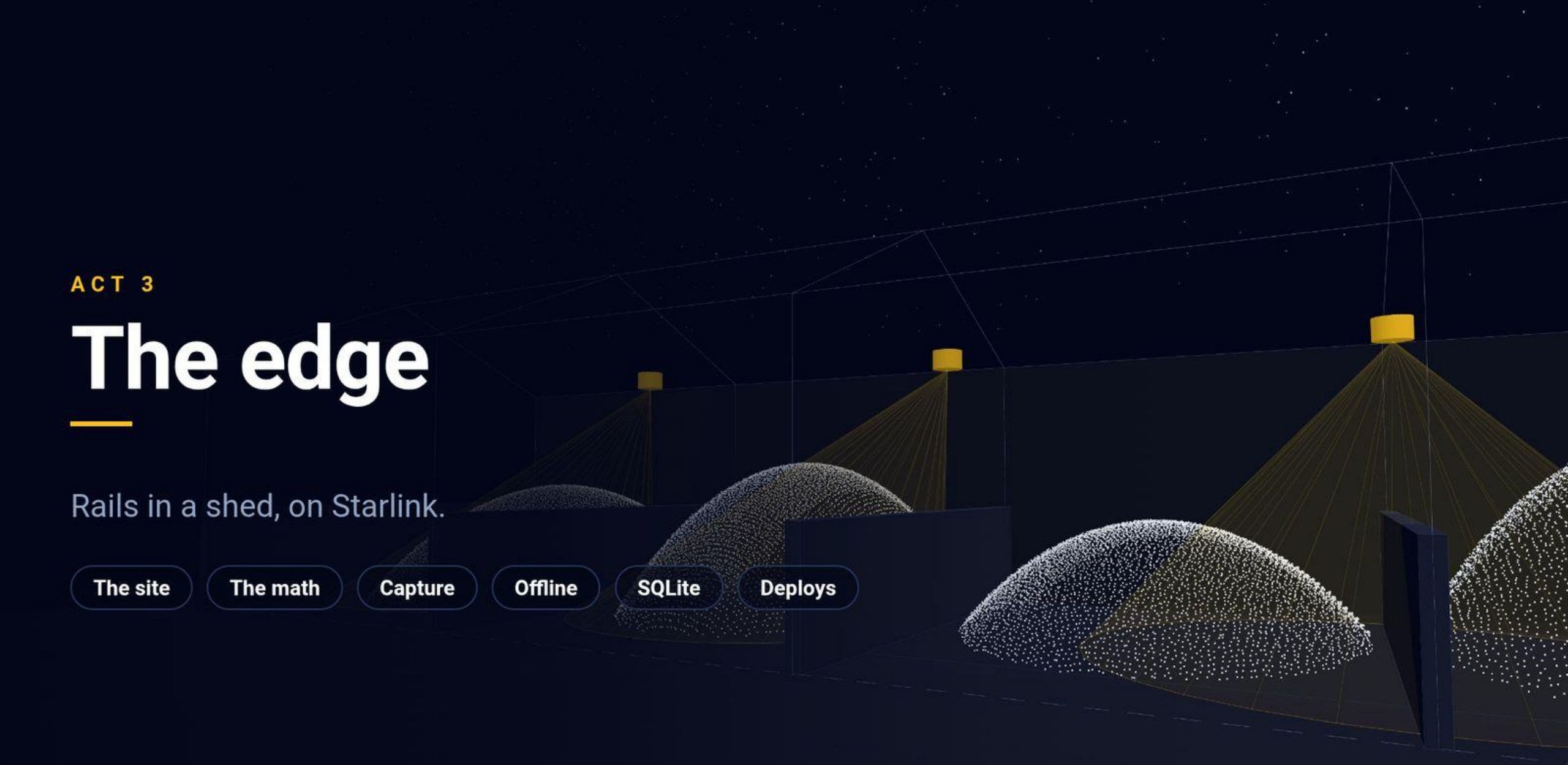
Let's use Rails.

ACT 3

The edge

Rails in a shed, on Starlink.

The site · The math · Capture · Offline · SQLite · Deploys



The site


3× LiDAR on the rafters
1,184 t
2,410 t
962 t
1,977 t
Jetson box
Corn Meal
Canola 1
Canola 2
Wheat Mid

So how do we get the volume?

```
# what everyone in this space uses
import open3d as o3d

pcd  = o3d.io.read_point_cloud("scan.pcd")
down = pcd.voxel_down_sample(0.05)
plane, inliers = down.segment_plane(
    distance_threshold=0.02,
    ransac_n=3, num_iterations=1000)
```

Open3D. In Python.

Open3D, in Ruby

PYTHON · OPEN3D

```python
# what everyone in this space uses
import open3d as o3d

pcd  = o3d.io.read_point_cloud("scan.pcd")
down = pcd.voxel_down_sample(0.05)
plane, inliers = down.segment_plane(
    distance_threshold=0.02,
    ransac_n=3, num_iterations=1000)
```

RUBY · OUR POINTCLOUD GEM

```ruby
# our pointcloud gem
require "pointcloud"

cloud = PointCloud.load("scan.pcd")
down  = cloud.voxel_down_sample(0.05)
plane, inliers = down.segment_plane(
  distance_threshold: 0.02,
  num_iterations: 1000)
```

Same method names. A Ruby gem.

Under the hood: C

Ruby
cloud.rb

```ruby
def voxel_down_sample(size)
  Cloud.new(point_buffer.voxel_downsample(size))
end
```

↓

C binding
pointcloud_ext.c

```c
rb_define_method(rb_cPointBuffer, "voxel_downsample", voxel_downsample, 1);
```

↓

CUDA
same function, on the Jetson

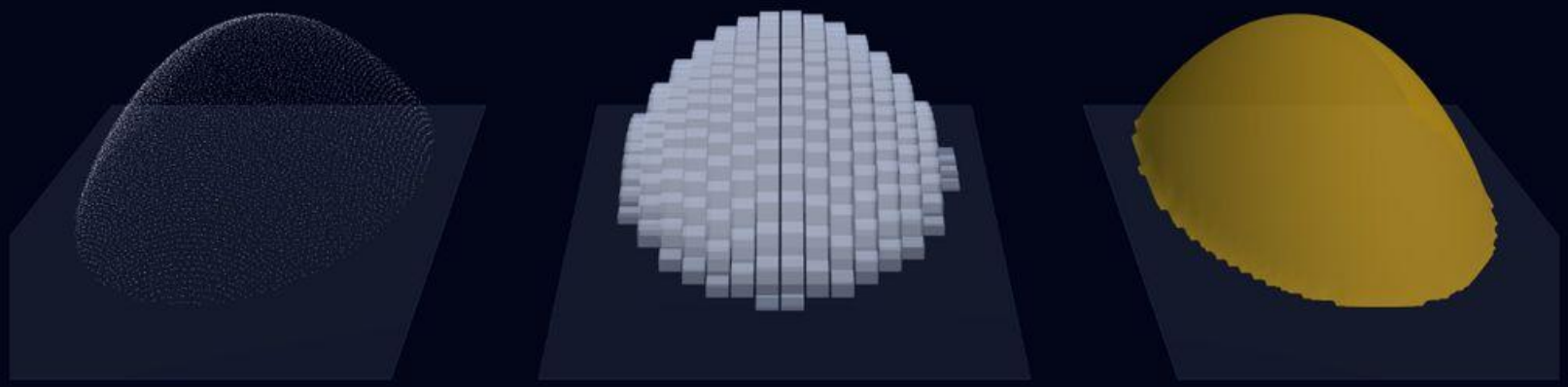
```c
#ifdef HAVE_CUDA
  if (use_cuda()) return cuda_voxel_downsample(points, size);
#endif
```

↓

arm64
extconf.rb

```ruby
$CFLAGS << " -O3 -march=armv8.2-a"   # the Jetson's CPU
$CFLAGS << " -DHAVE_CUDA" if nvcc   # and its GPU
```

From laser points to tons

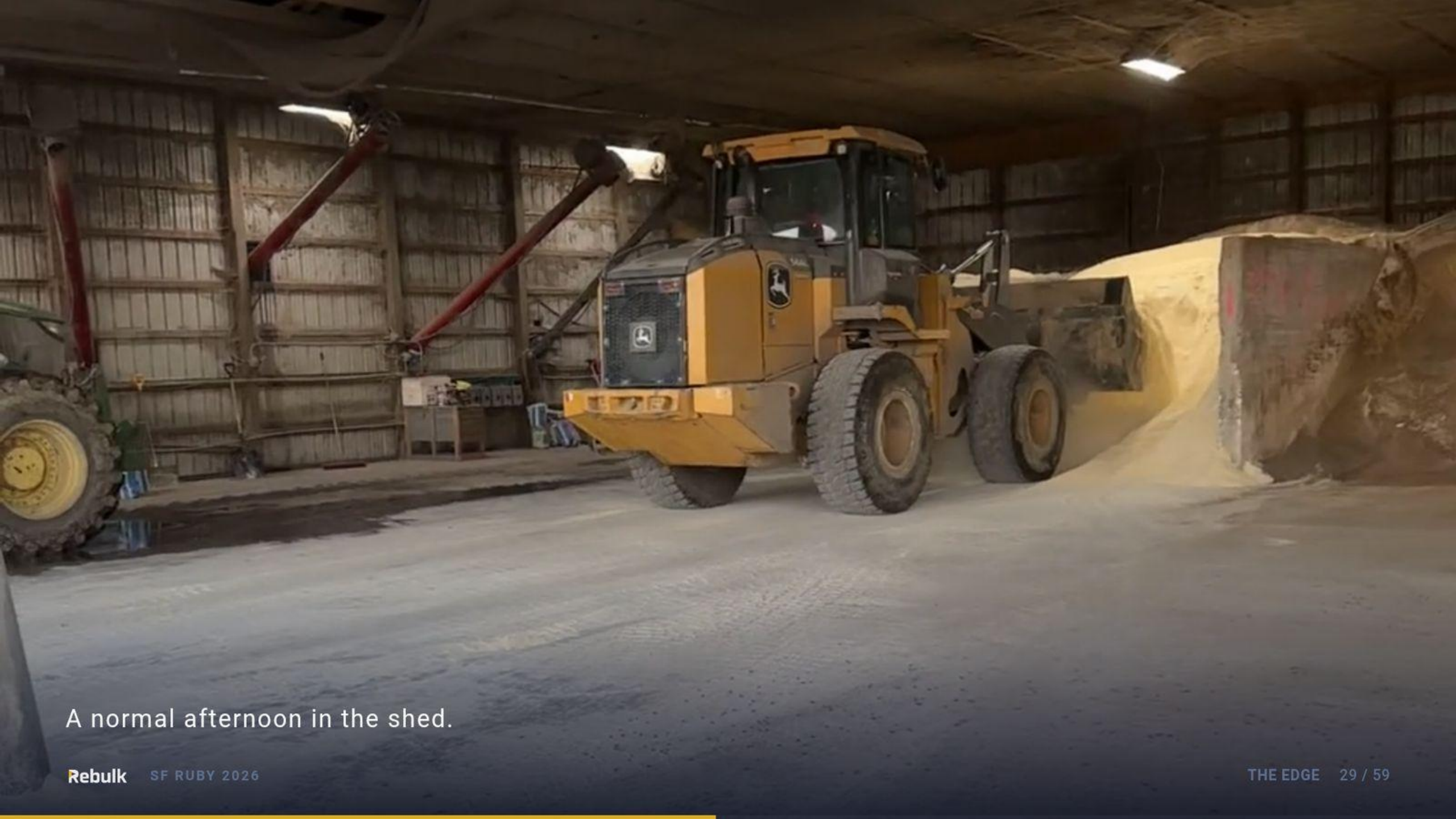


points

a grid of cells

Σ cells × height

A grid of cells. Add up the heights. Ruby turns it into tons, and refuses a number it can't trust.

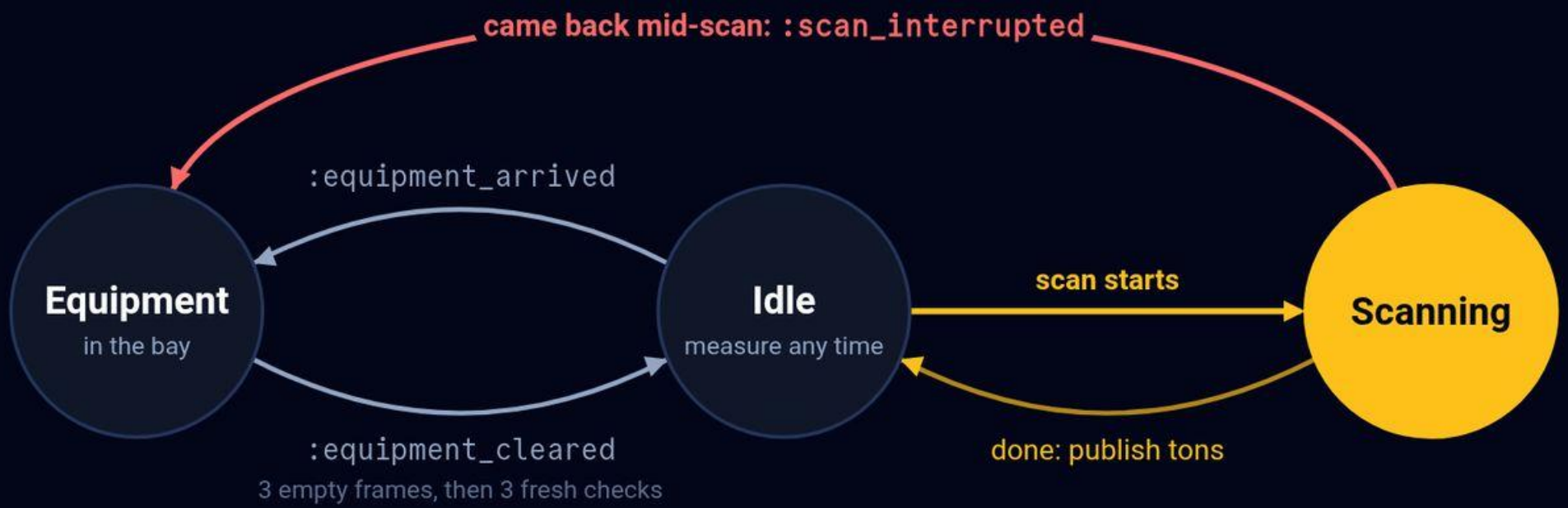


A normal afternoon in the shed.

Nothing measures a pile with a machine in it



A loader decides when we measure




```
gate.on(:equipment_cleared) { @capture_pending = true }                          # the loop starts a scan
gate.on(:scan_interrupted) { @active_cancel_token&.cancel! }                     # LiDAR killed in ~0.5 s, scan dropped
gate.state_machine.on(:transition) { |from, to| EdgeEventPublisher.publish(...) }  # tell the cloud
```

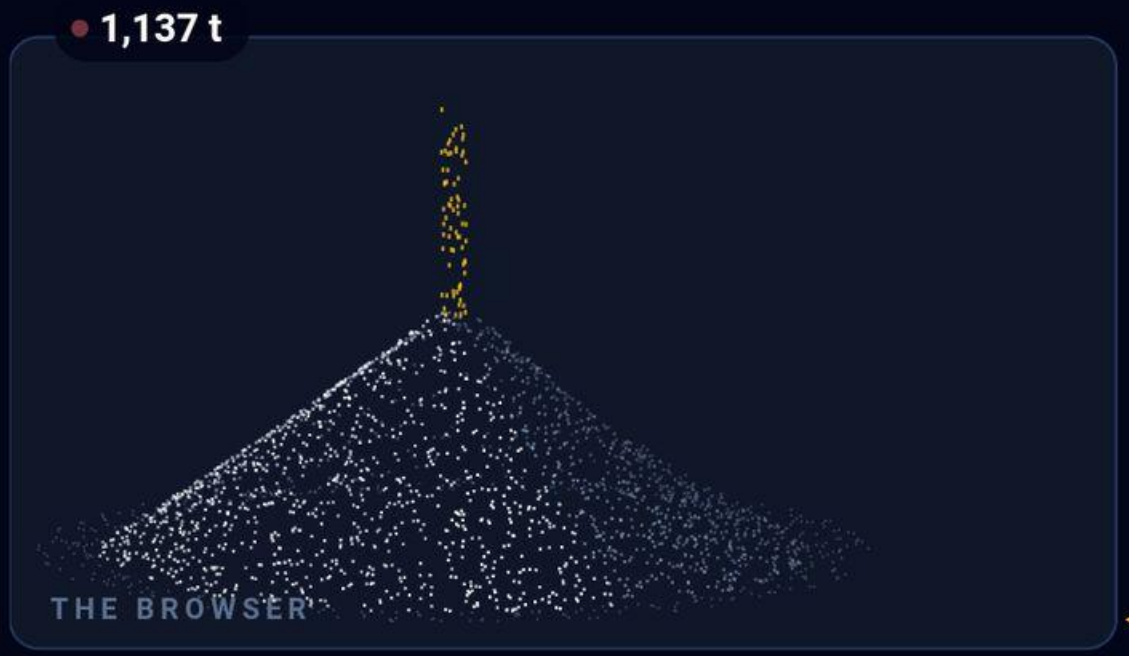
Solid Queue runs the shed

```
# config/recurring.yml (on the Jetson)
capture_scheduler:
  class: ScheduleCapturesJob
  schedule: every minute
edge_config_sync:
  class: EdgeConfigSyncJob
  schedule: every minute
baywatch_outbox_drain:
  class: BaywatchOutboxDrainJob
  schedule: every 2 minutes
```

The same Rails app as the cloud.

Jobs on a schedule keep the capture running, pull new settings and send what’s waiting.

Live, over ActionCable



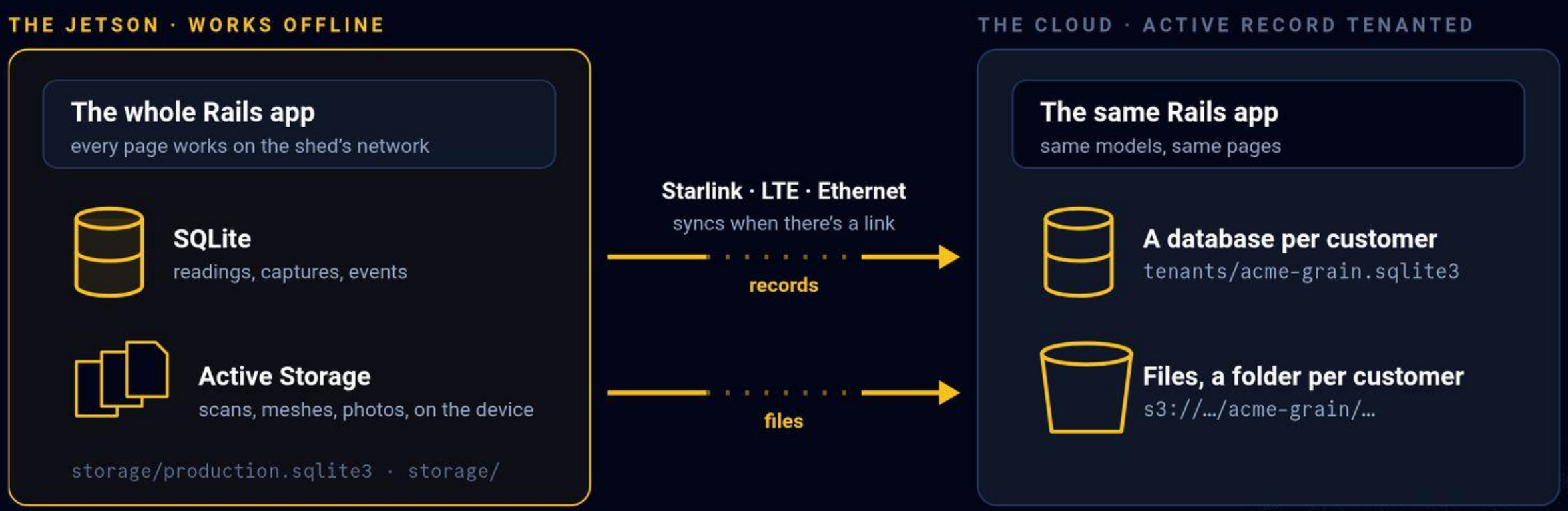
1. **`Rails.cache`**
   "someone is watching"
2. **The Jetson asks the cloud**
   is anyone watching?
3. **Yes: it streams the LiDAR**
   frames over plain HTTPS
4. **`ActionCable.broadcast`**
   Solid Cable, no Redis

`LiveVolumeChannel`
tons, as they change

`LivePreviewChannel`
point-cloud frames

`EdgeEventsChannel`
the gate badge, live

Works offline, syncs when it can



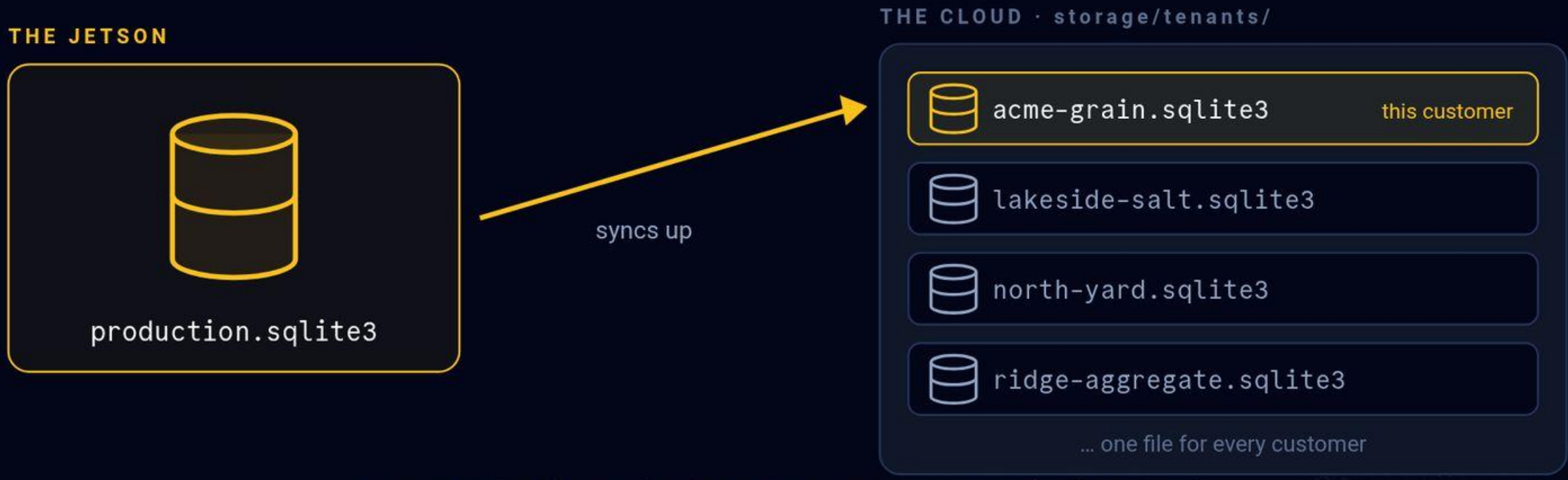
What syncs, and which way



`X-Idempotency-Key: 3f9a…` a retry after a dropped link is counted once

SQLite

SQLite, all the way down



One SQLite file on the box. One per customer in the cloud.

Active Record Tenanted

> “activerecord-tenanted was working great.”
>
> Mike Dalessio, 37signals

Active Record Tenanted



No tenant? It raises.

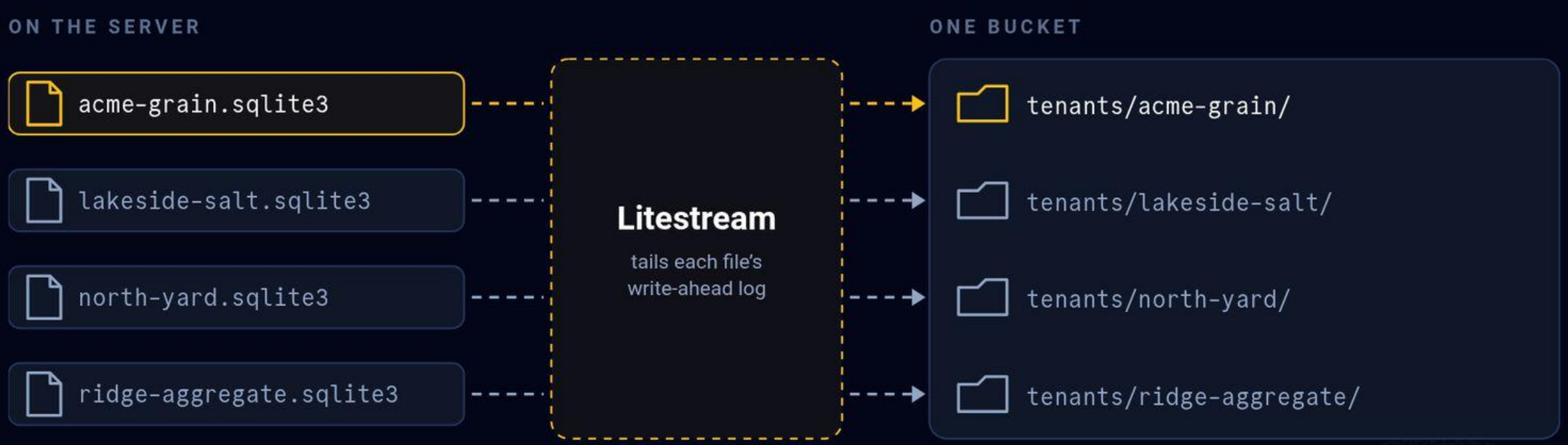
Isolation is a file path.

Restore one customer.

Same code on the Jetson.

Litestream

Litestream: every file, backed up as it's written



Seconds behind.
every change ships as it's written

One customer back.
restore one file, to any second

Nothing to run.
no replica database, no Postgres

Merging is deploying



No release day. Small changes, all day, each one easy to undo.

Deploying to remote edge boxes



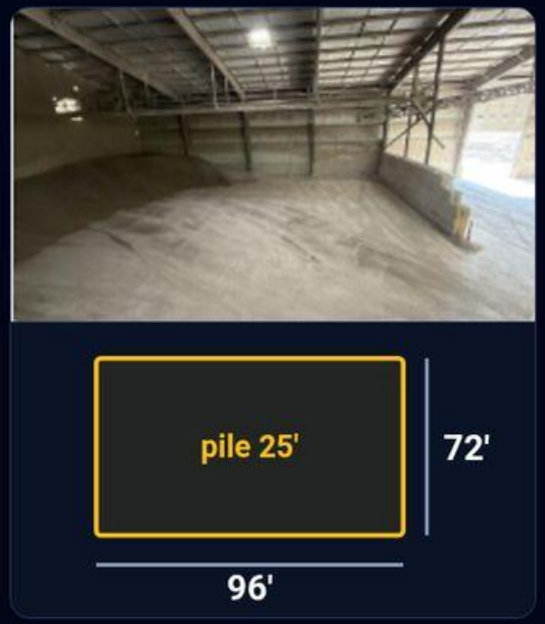
The whole stack

FRONT END	Inertia Rails		React	
RAILS 8	Active Record	Active Job	Action Mailbox	Action Cable
SOLID	Solid Queue jobs	Solid Cache cache	Solid Cable websockets	
DATA	SQLite	activerecord-tenanted a file per customer	Litestream backups	
DEPLOY	Kamal	one Linux box cloud	a Jetson the shed	

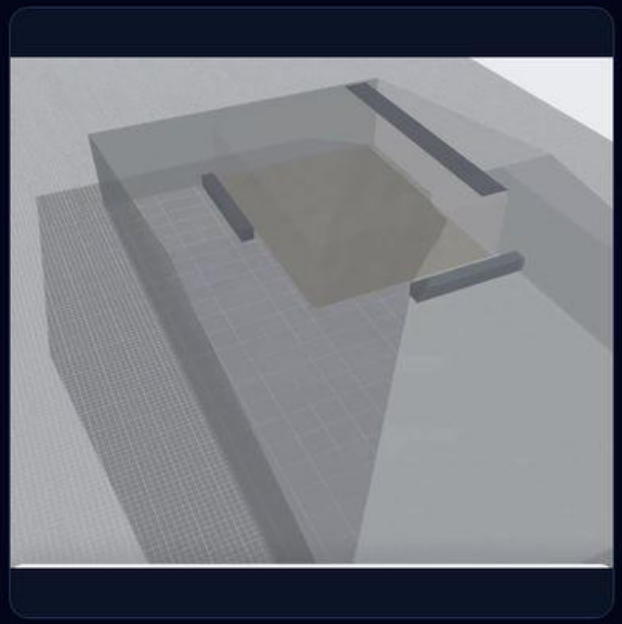
What about the hardware?

Where deployments are now

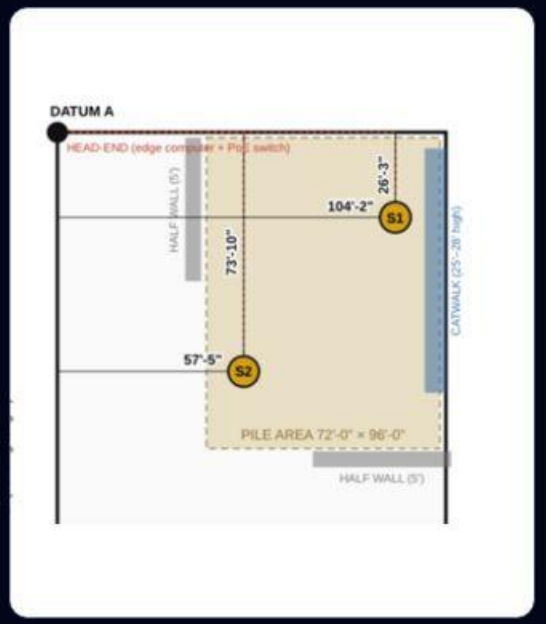
1. Shed dimensions in



2. Modeled in 3D

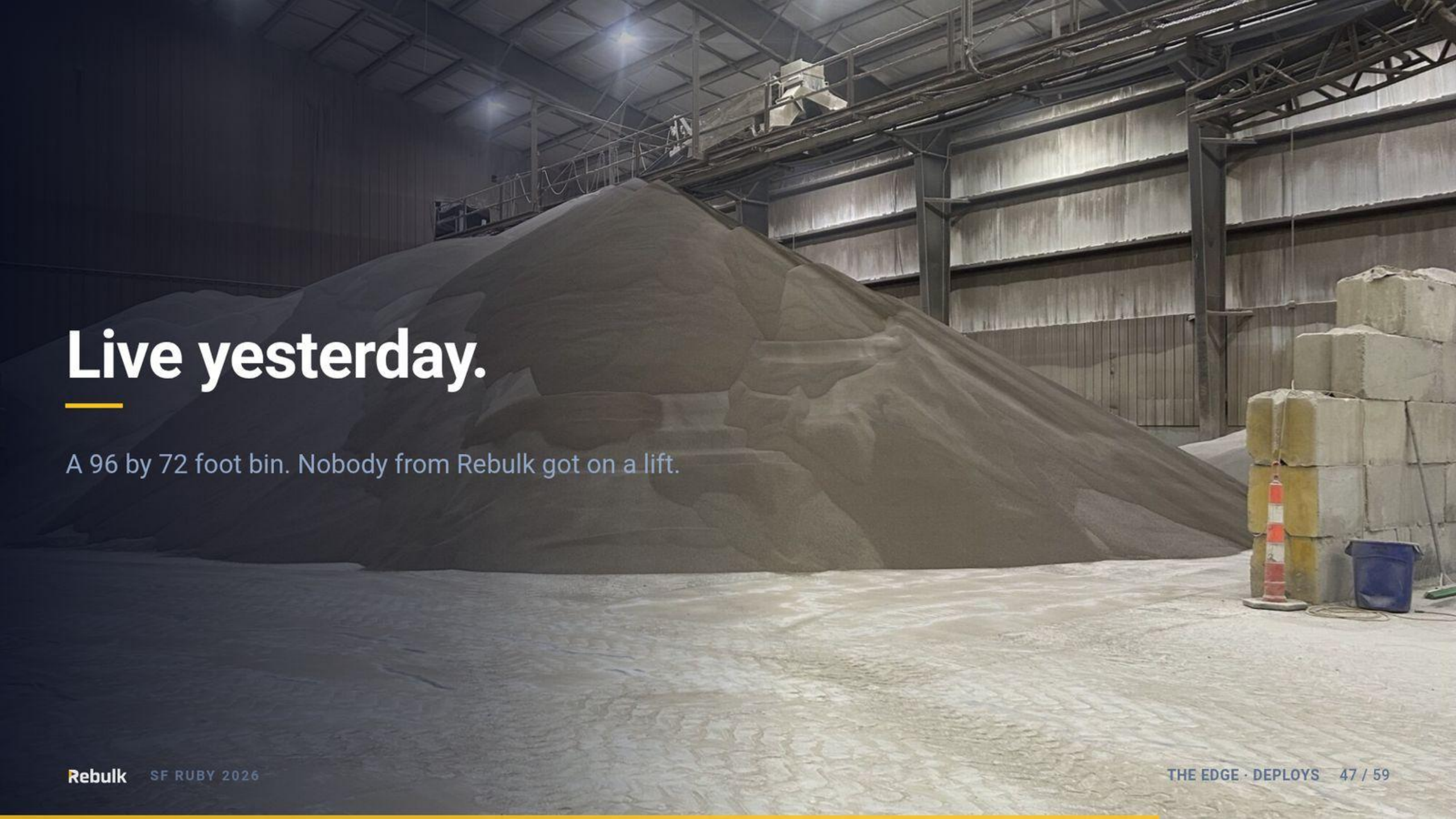


3. Sensor spots out



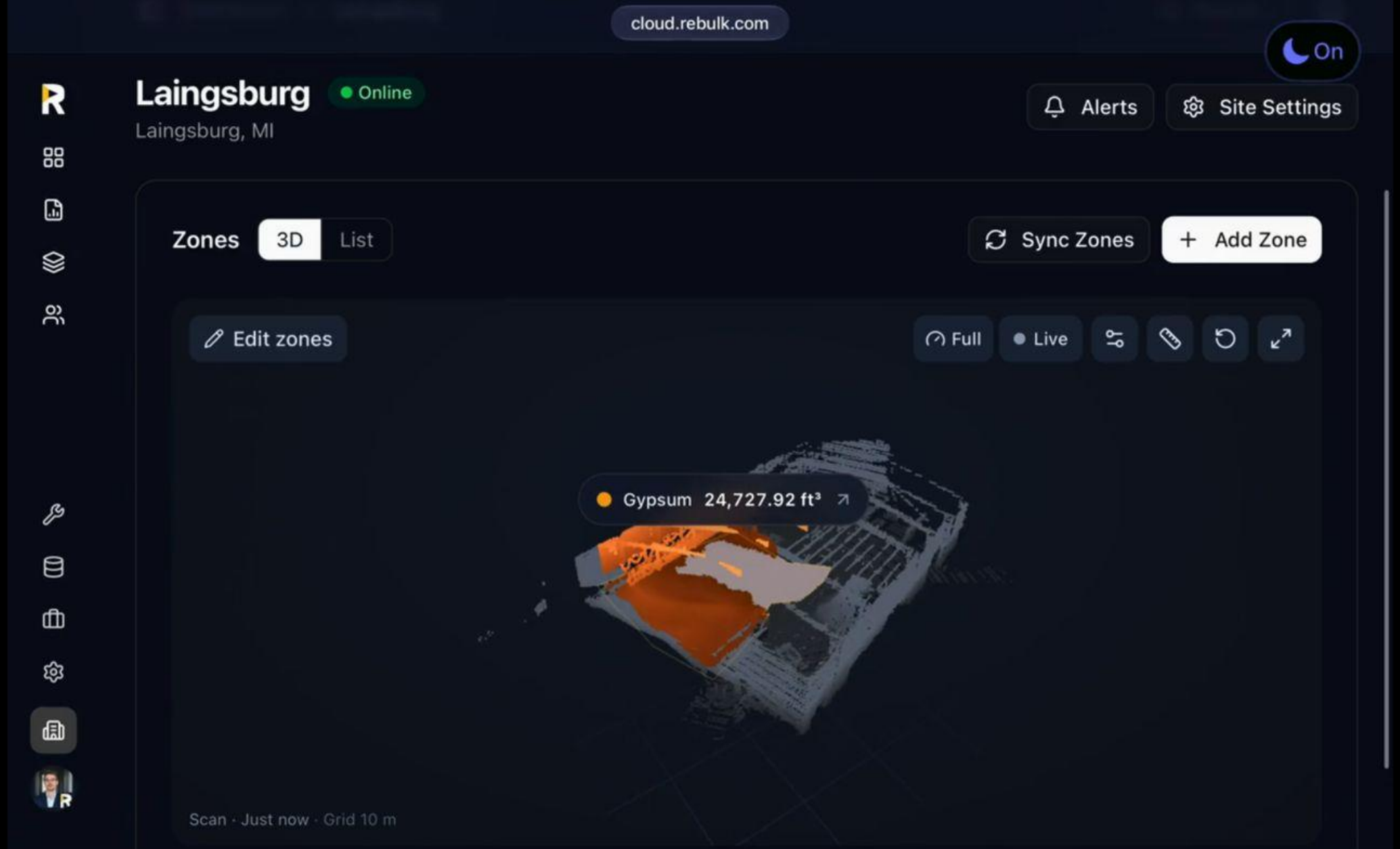
4. The box ships. They install it.





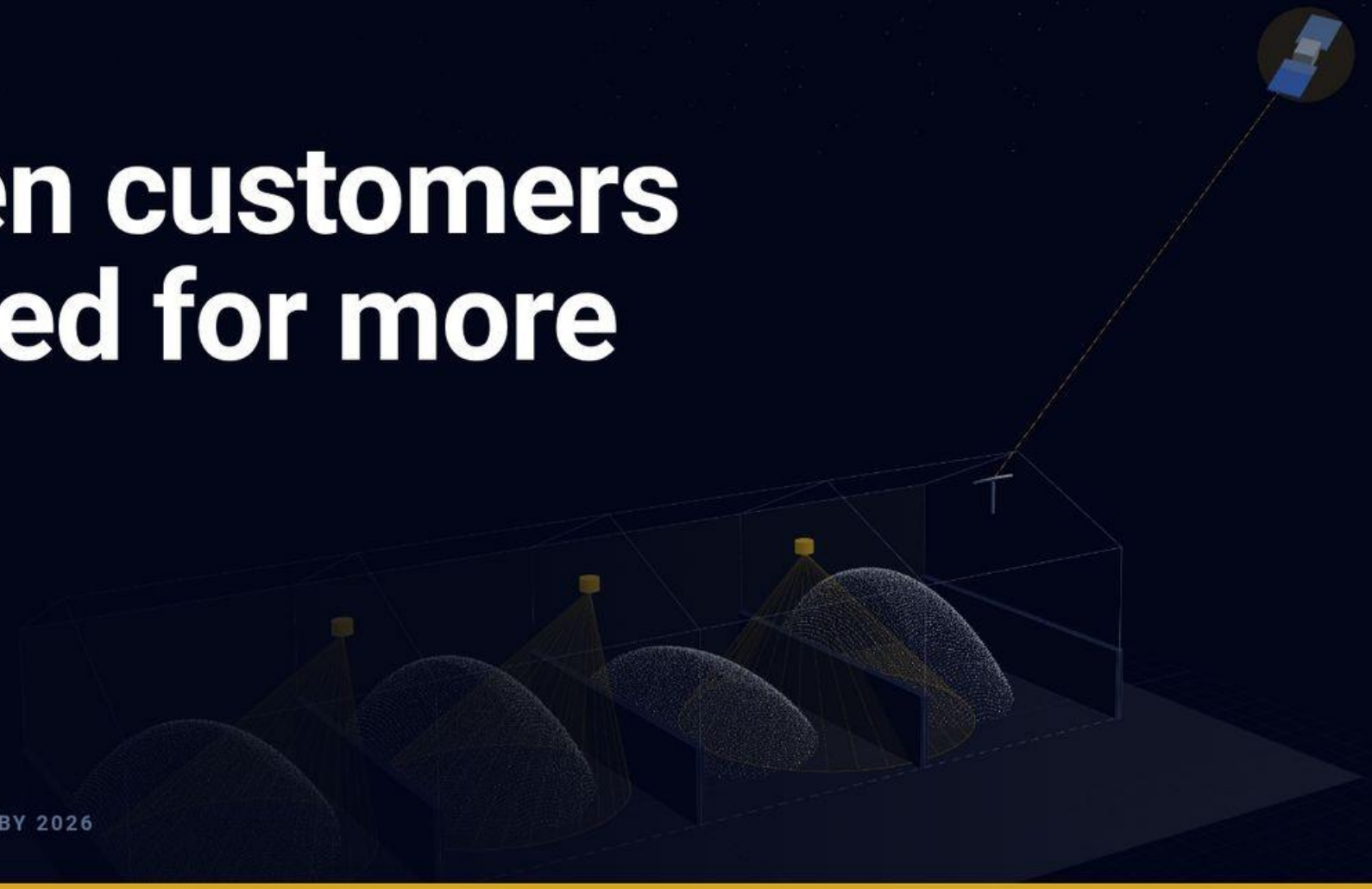
Live yesterday.

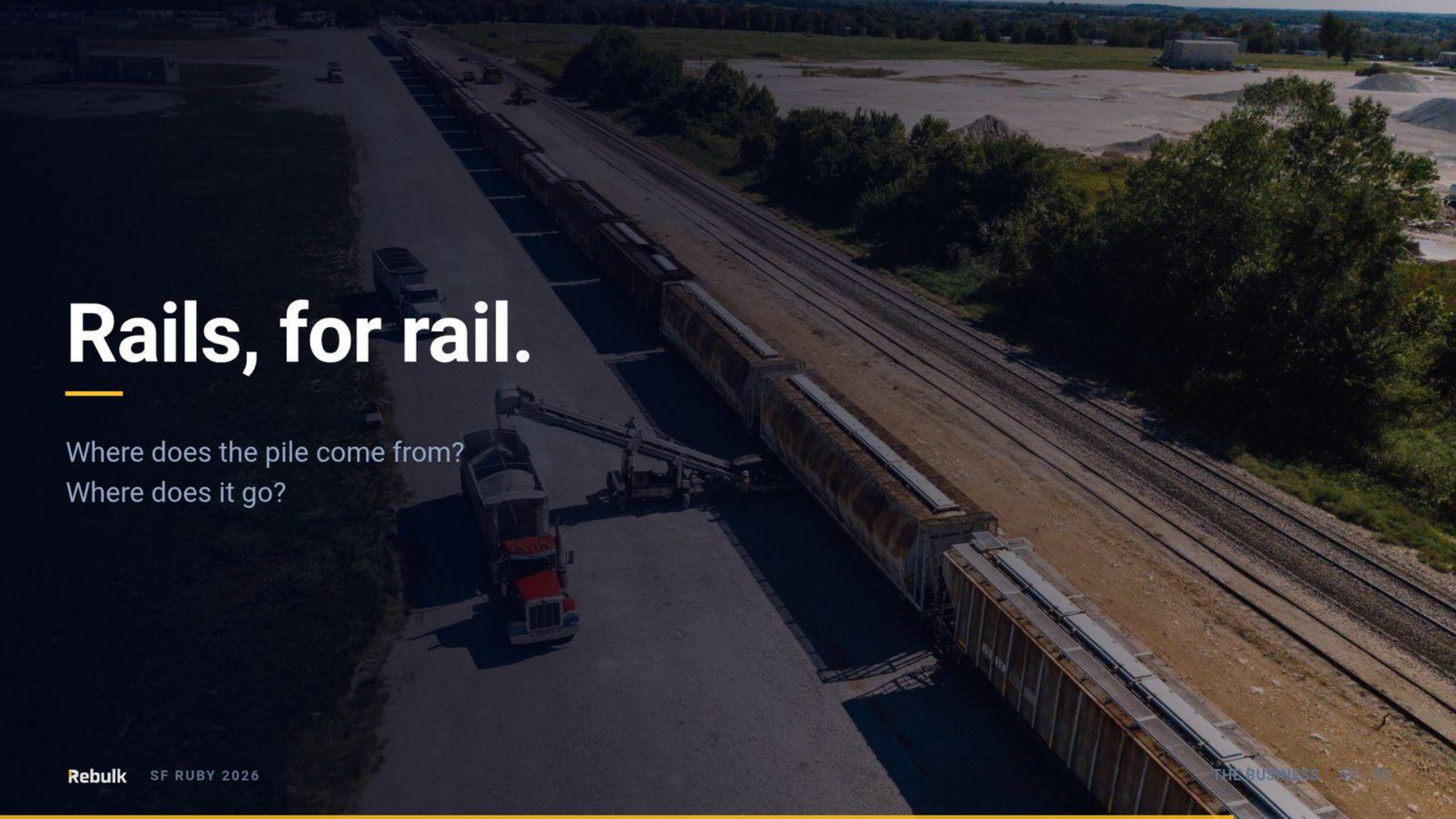
A 96 by 72 foot bin. Nobody from Rebulk got on a lift.


cloud.rebulk.com
On
Laingsburg
Online
Laingsburg, MI
Alerts
Site Settings
Zones
3D
List
Sync Zones
Add Zone
Edit zones
Full
Live
Gypsum 24,727.92 ft³
Scan · Just now · Grid 10 m

ACT 4

Then customers asked for more



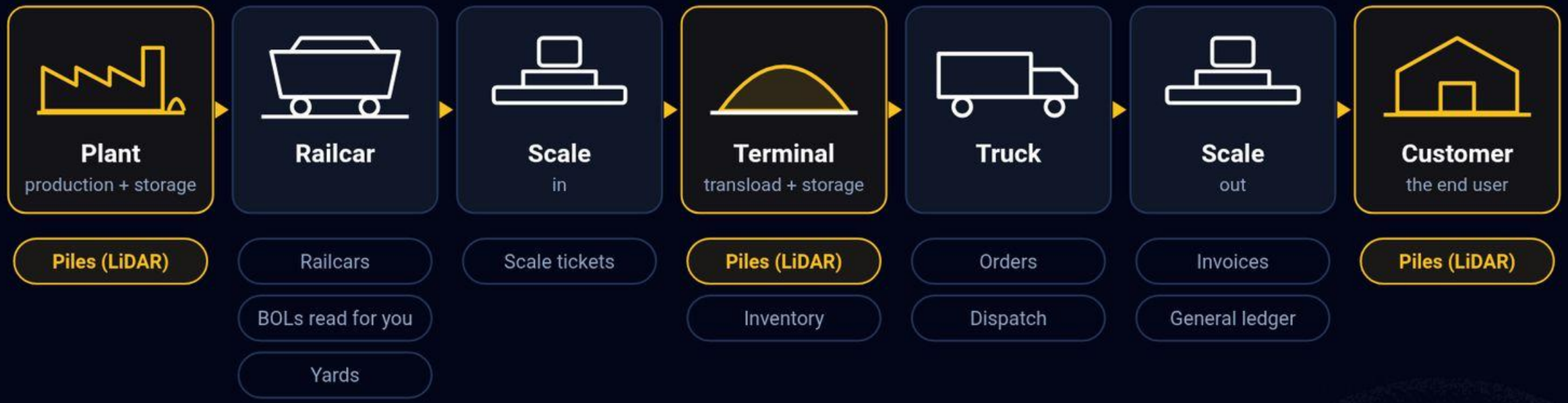


Rails, for rail.

Where does the pile come from?
Where does it go?

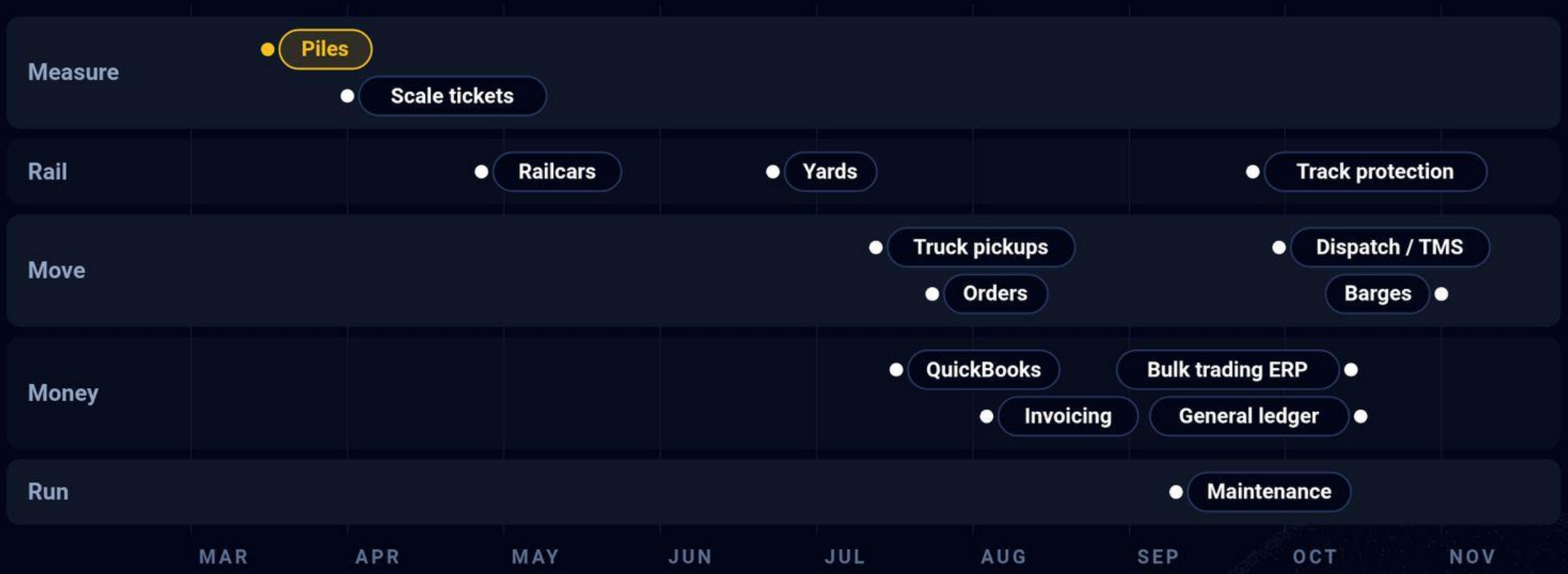
Following the material

WHAT WE EXPANDED INTO



where we started: scanning the piles

Eight months



Same parts, every time



ACT 5

Why it matters now

Frameworks in the age of LLM coding.



A one-person framework.
Two of us.

Why agents work so well on Rails

Conventions

the agent knows where code goes

Primitives

jobs, mail, files, cache, websockets

Tests

the suite tells us it still works

All of it, simplified

~~Supabase~~ ~~Upstash~~ ~~Inngest~~ ~~Pusher~~ ~~Vercel~~

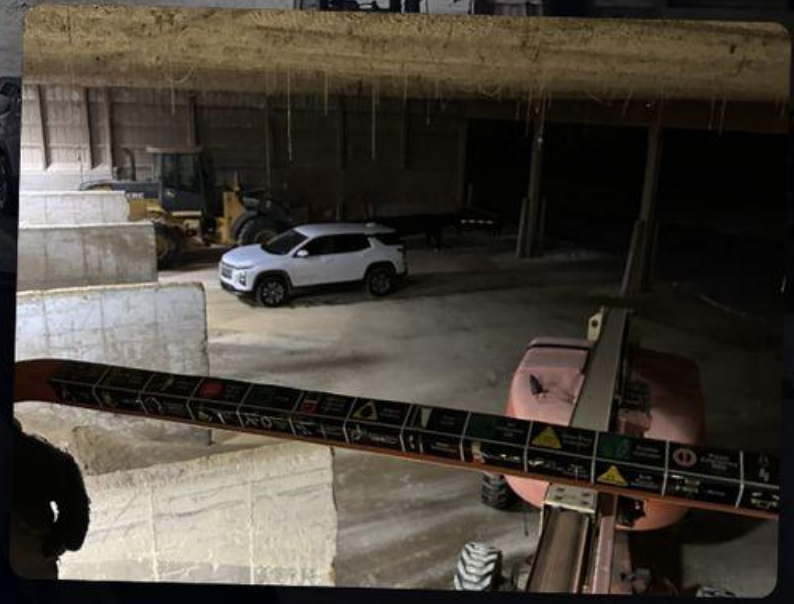
~~Clerk~~ ~~Prisma~~ ~~Next.js~~ ~~Resend~~ ~~Sentry~~

↓

One Rails app

Peace.

Focus on business problems, not stack problems.





Thank you.

Cole Robertson · CTO & Co-founder, Rebulk

rebulk.com

@Cole__Robertson **in/colemrobertson**

Come find me after. Ask me about startups, hardware, SQLite or agentic coding.

rebulk.com