

EXHALE

A Field Guide to the Paycheck Rails

(formerly Palolo)



EWA

Savings

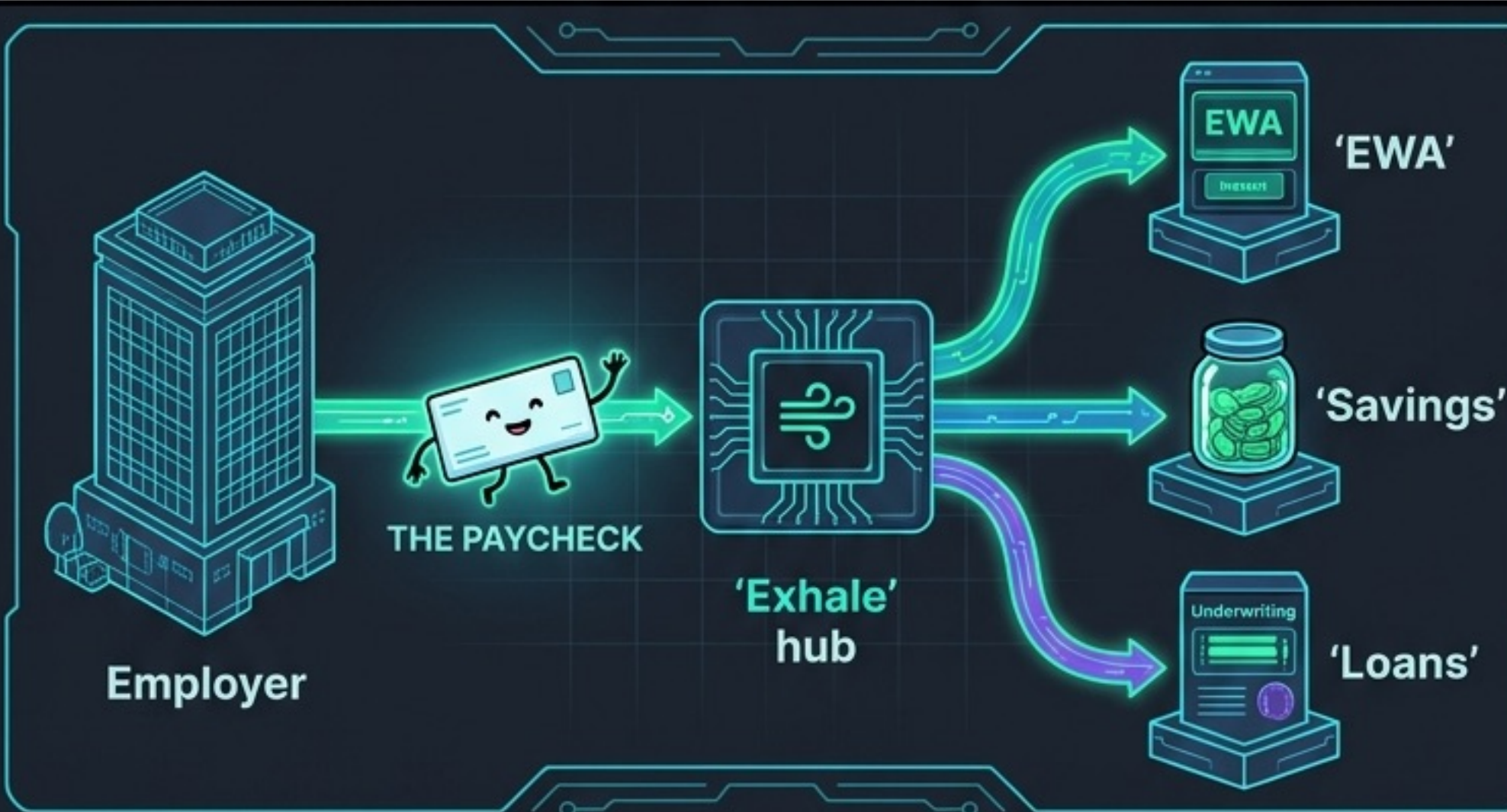
Loan

Follow one paycheck —
from employer to perks.



Our expedition begins between an employer and an employee's money.

Meet **Exhale** — once known as **Palolo**.



Exhale turns paychecks into financial **PERKS**. It's a big **TypeScript monorepo monorepo** — accuracy and compliance matter.

Where do we start?

We follow a single paycheck.

FIELD HUD ▶

Exhale (was Palolo) ·
TS monorepo (pnpm) ·
first reads:
README.md · PERKS.md

"The habitat is one big **monorepo** — nine **package-islands**, **one remote**." (deadpan nature-documentary voice)



``x go`` boots the whole stack — **Vite** client + **Express API** + **DB** + **queue**.

Field note: inside the **Explore container**, ``run-app`` is a convenience **wrapper** around the same thing.

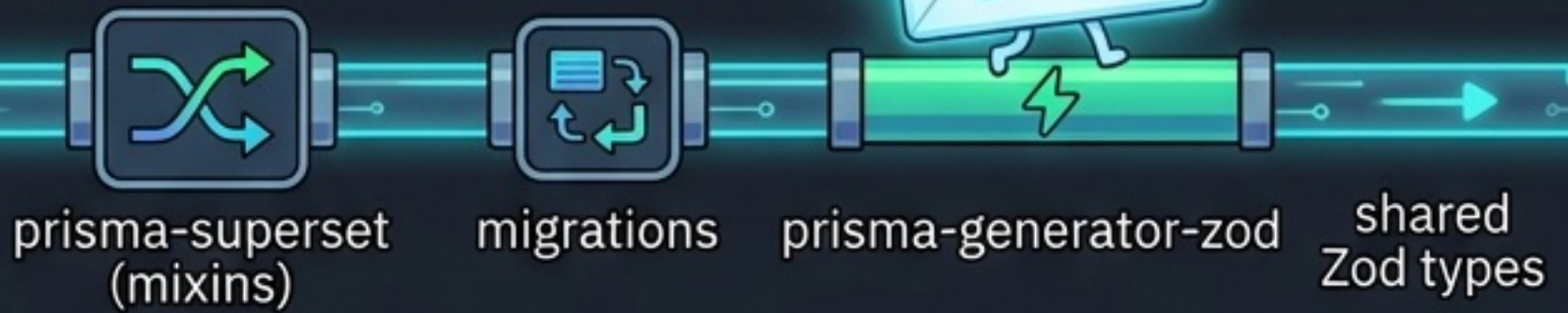
FIELD HUD ▶

ENGINEERING FACTS:

9 packages · client = **Preact** · scripts/x: **x go** · **x db** · **x test** · (Explore wrapper: **run-app**)

One schema feeds the whole habitat — and both of its faces.

models.prisma



Employee consumer app (Preact client)



HQ employer admin



ONE backend, TWO surfaces.

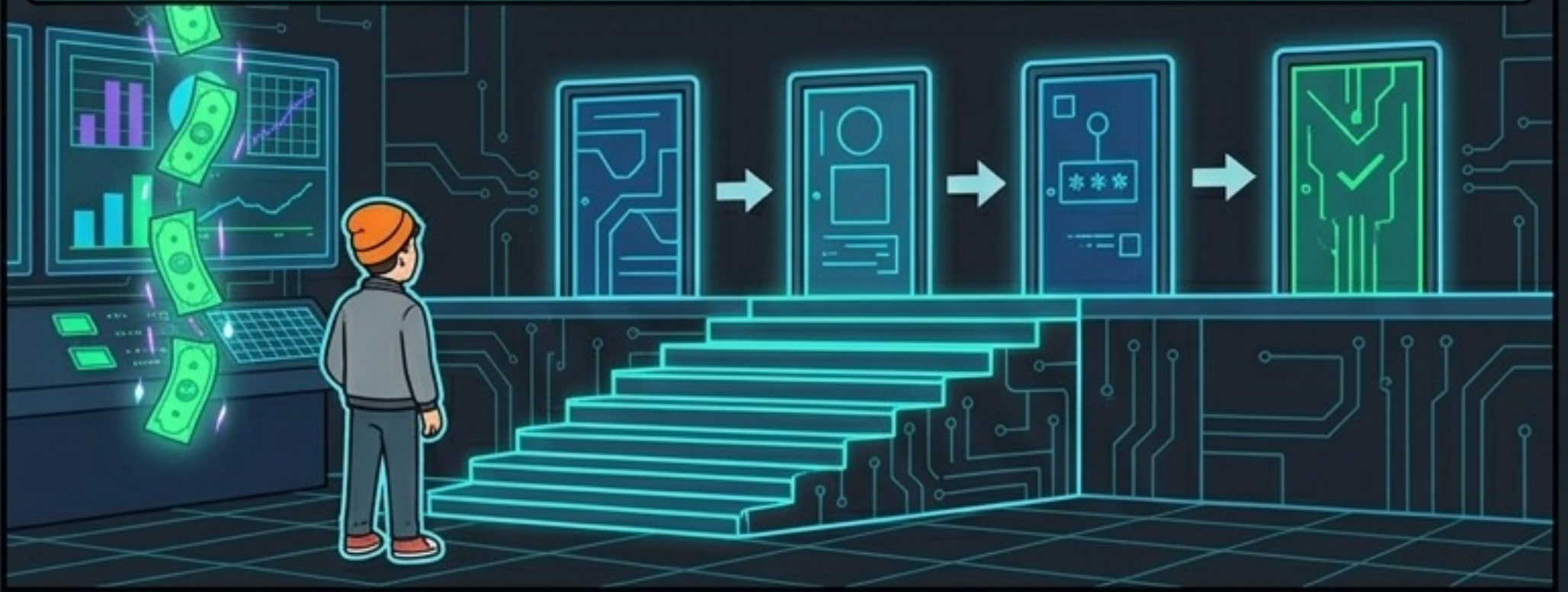
Hand-edit `models.prisma`; the *mixins* de-dup common fields, and the **Zod types** are generated for client and server to share.

So the schema is the single source of truth.

FIELD HUD ▶

```
packages/server/prisma/models.prisma ·  
prisma-superset mixins →  
prisma-generator-zod (shared Zod)  
dual-surface: consumer + HQ admin
```

Before anyone touches money, they climb a staircase of doors.



FIELD HUD ▶

enum SessionAuthStatus

(models.prisma) ·

Unauthenticated → AwaitingOtp
→ AwaitingPin → Authenticated

Our specimen enters the habitat — synced in from payroll.



Payroll providers sync the paycheck into a 'MemberPaycheck'.

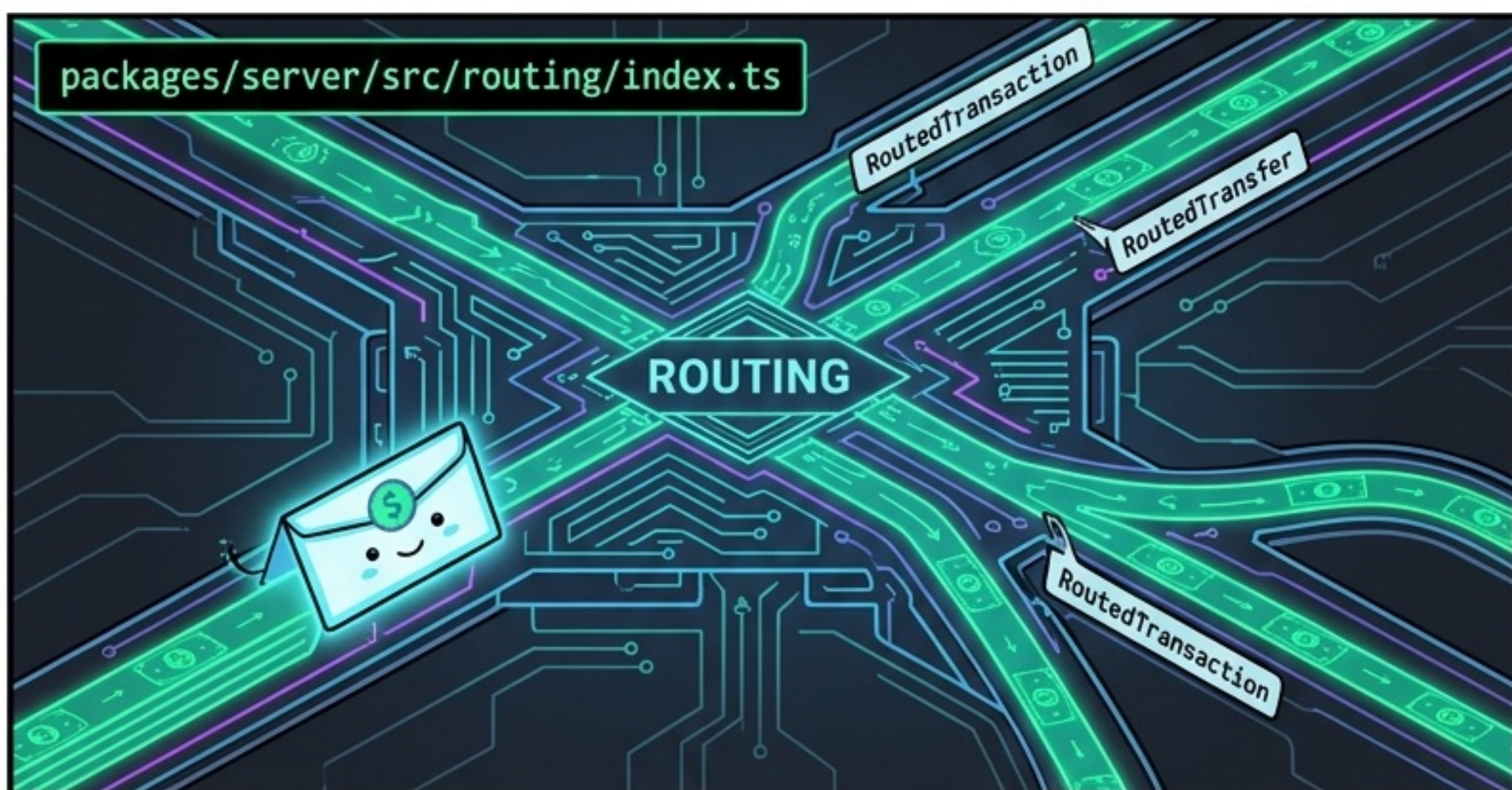
The deposit lands in a **CLEARING** account first.

A waiting room for money. It does not stay long.

FIELD HUD ▶

payroll: Atomic / Finch / Argyle
MemberPaycheck
FinancialAccount type = Clearing

The clearing account is a brief stop — then the split happens.



FIELD HUD ▶

```
packages/server/src/routing/index.ts
RoutedTransaction / RoutedTransfer
clearing window = banking cutoff / effective date
```

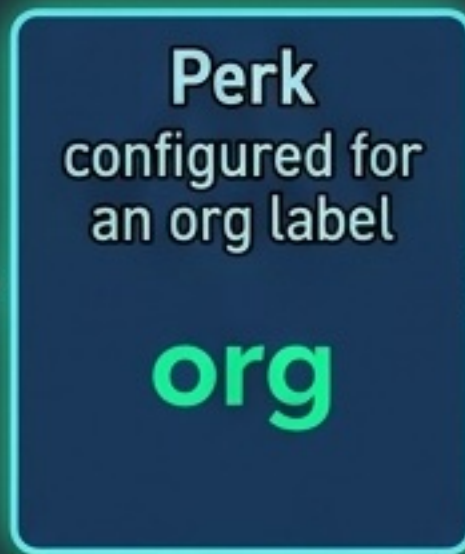
Note the field journal:
the window is a **banking-cutoff rule** — not a tidy round number.



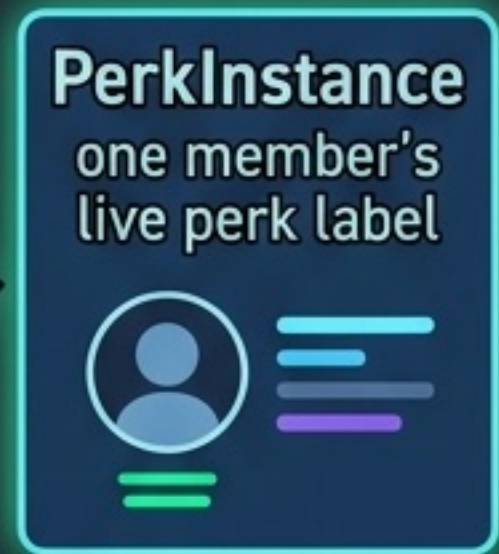
Every perk is grown from a blueprint, per member.



instantiate



configured with
org logo hint



representation/
avatar/status hint

ELIGIBILITY



ELIGIBILITY

OPT-IN



OPT-IN

Eligibility AND opt-in decide who gets what.

A **PerkTemplate** instantiates a **Perk**, which activates as a **PerkInstance**.

So the same template can behave differently per org and per member.

FIELD HUD ▶

PerkTemplate → Perk → PerkInstance
eligibility: Ineligible/PreEligible/Eligible
opt-in: auto/user/approval

Three machines split the paycheck into three kinds of help.

EWA — instant advance

Loans — underwriting

Savings — employer match + vesting



THE PAYCHECK



Each product has its endpoint — **ewa.ts**, **loans.ts**, **save.ts** — coordinated by **perks.ts** and **members.ts**.



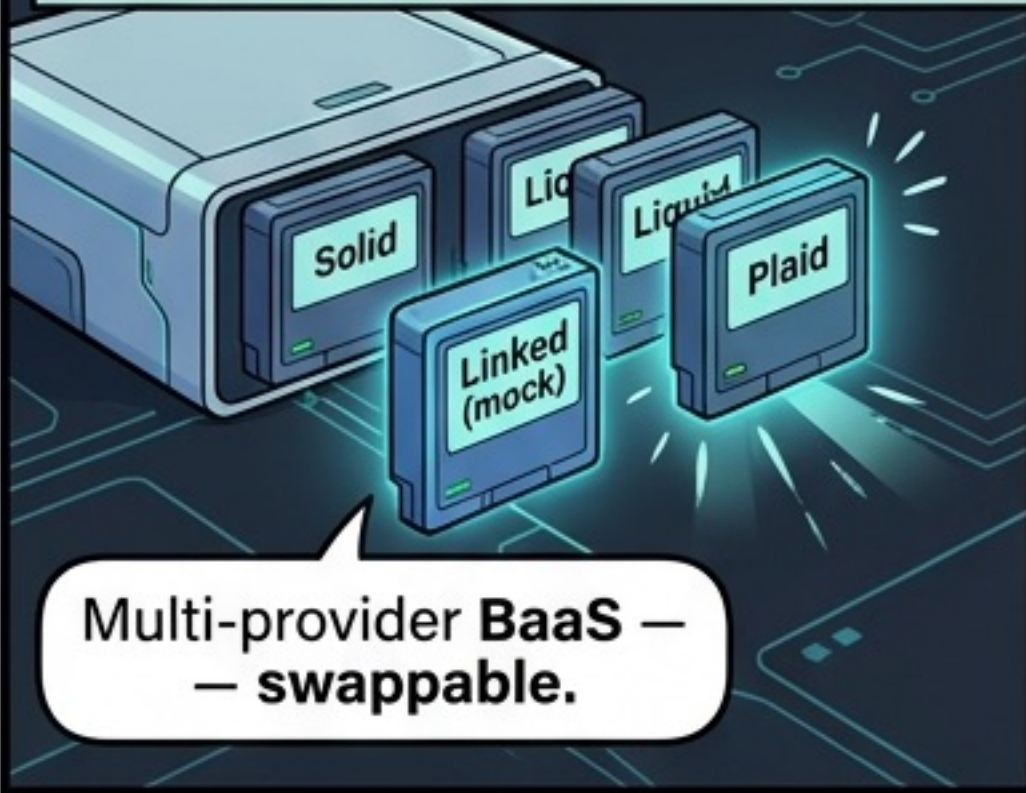
EWA now, a loan if neet, savings that the employer matches.



FIELD HUD ▶

```
endpoints: ewa.ts · loans.ts ·  
save.ts · perks.ts · members.ts ·  
Savings = employer match + vesting
```

To actually move money, the habitat plugs into a bank-as-a-service layer.



Multi-provider BaaS —
— swappable.



Where dev-mode
bench.



Money moves are
IDEMPOTENT —
a retry never
double-pays.



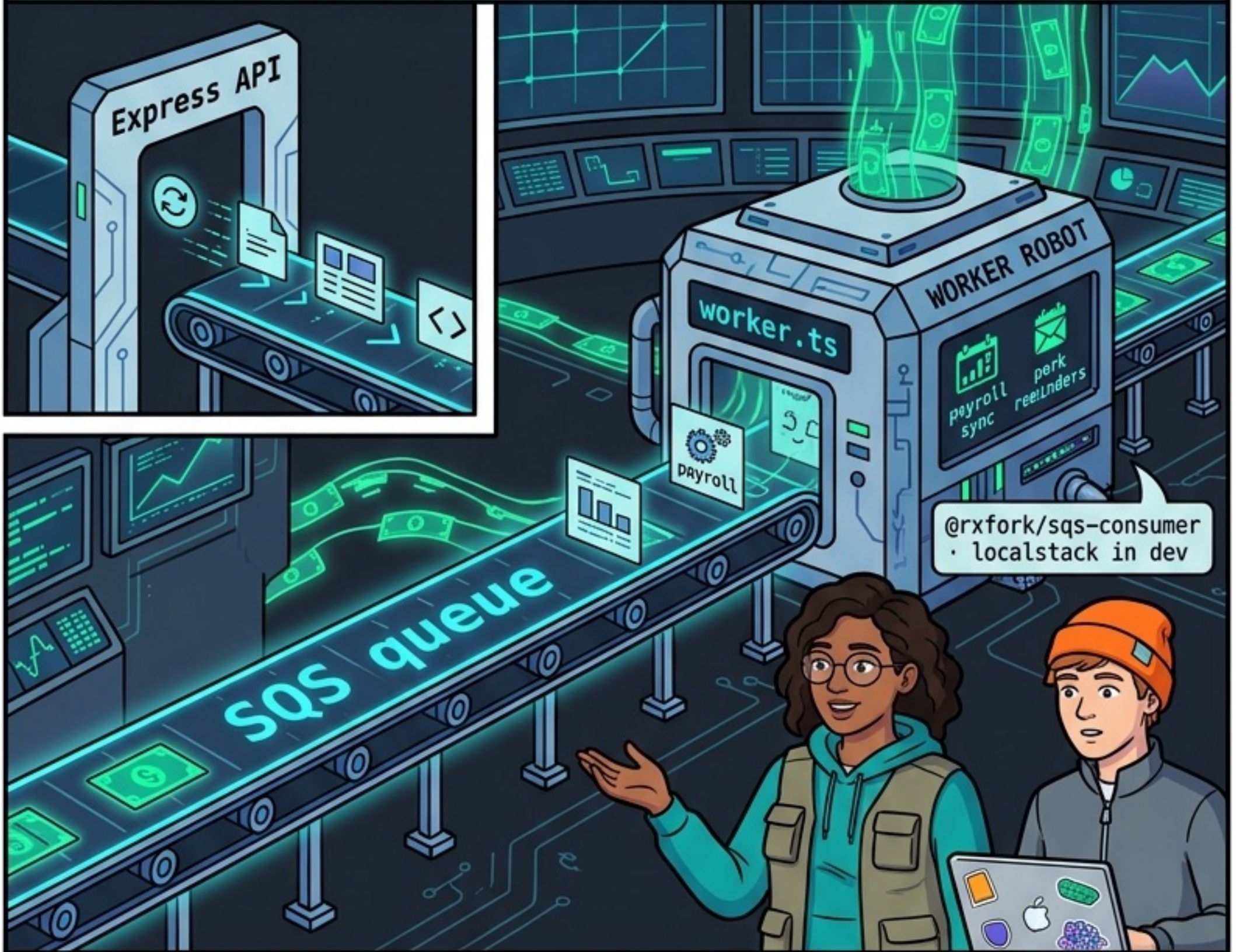
Swap the provider,
mock it in dev, and
never pay twice.



FIELD HUD ▶

services/baas: solid · liquid ·
linked(mock) · plaid · mock
providers in dev · idempotency keys

Slow work doesn't block the paycheck – it goes on a conveyor.



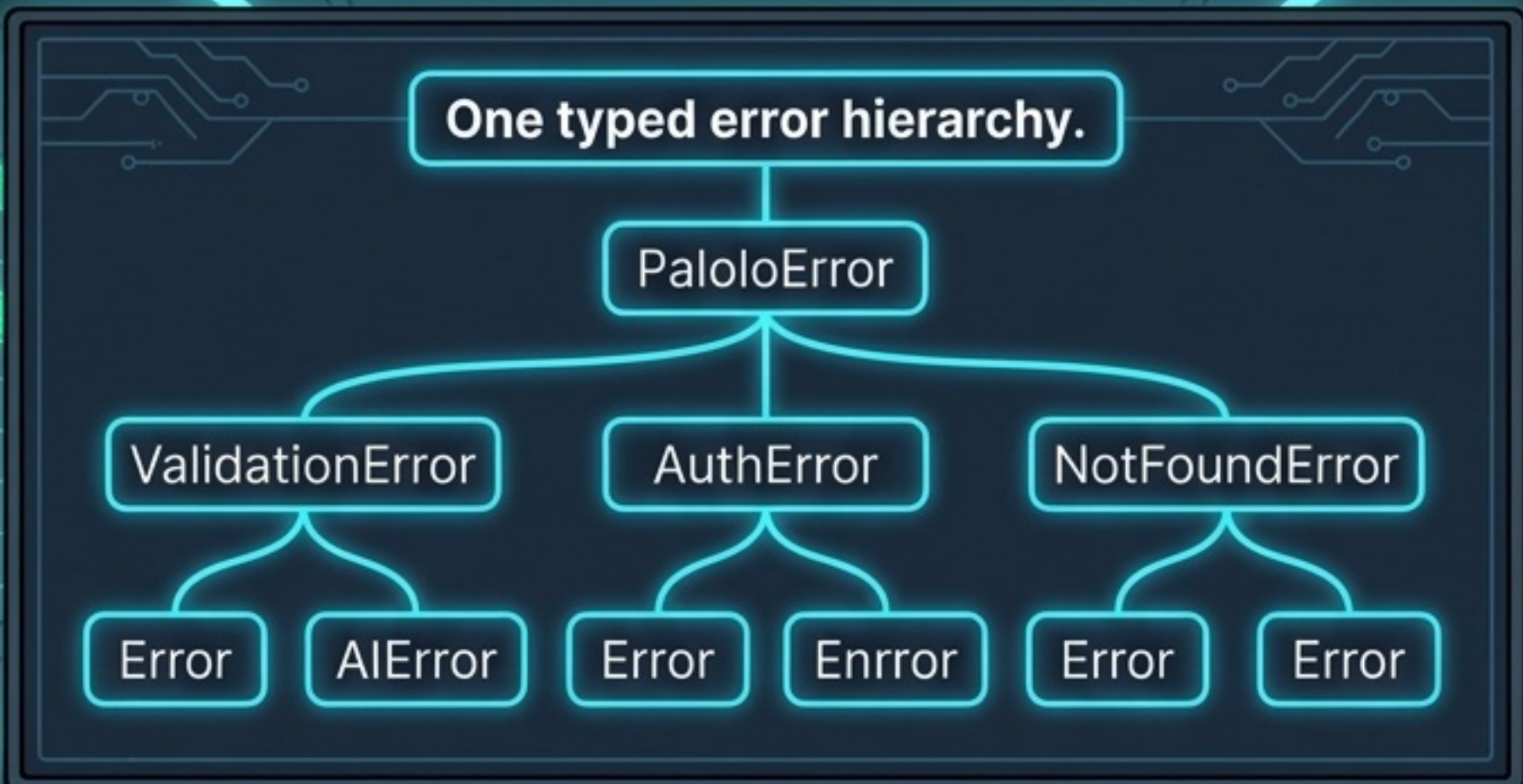
A background **worker** drains the queue. And the **notifications** package **renders** the **emails** and **texts**.



FIELD HUD ▶

Express API → SQS ·
packages/server/src/worker.ts
(@rxfork/sqs-consumer, localstack in dev)
· notifications: Sendgrid/Twilio

When something goes wrong, the habitat fails loudly and on the record.



Every change is recorded — who, what, when — so **HQ** can review it.

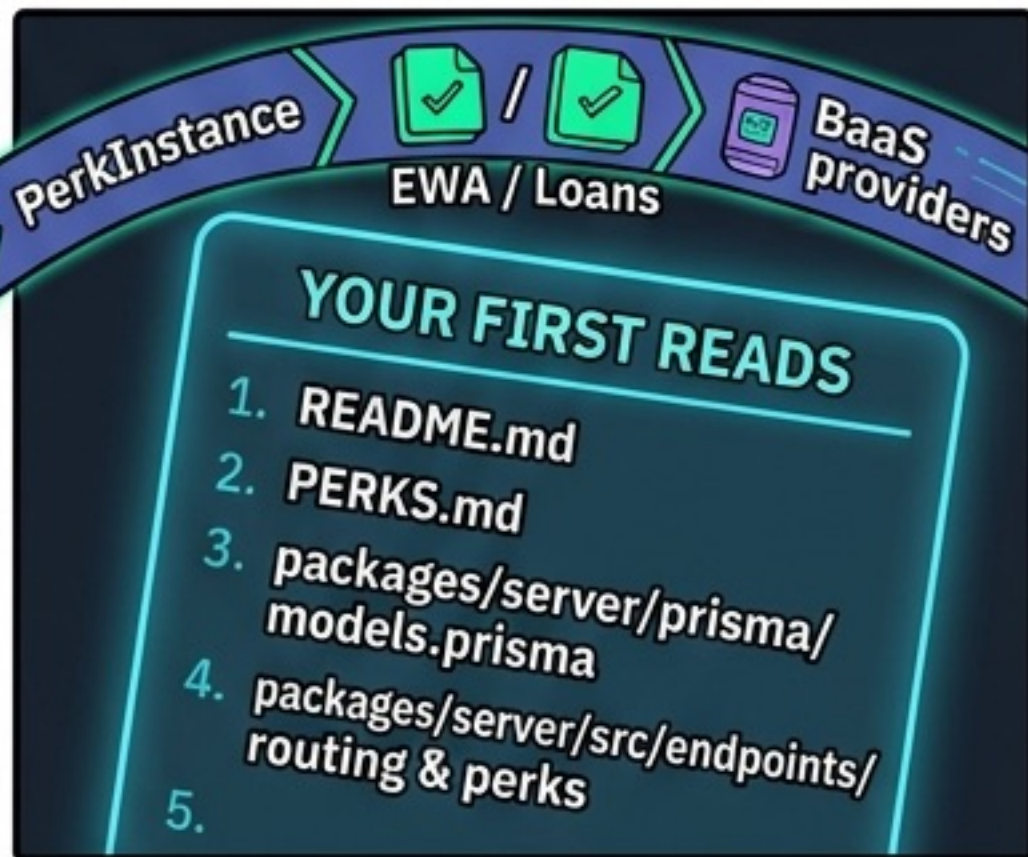
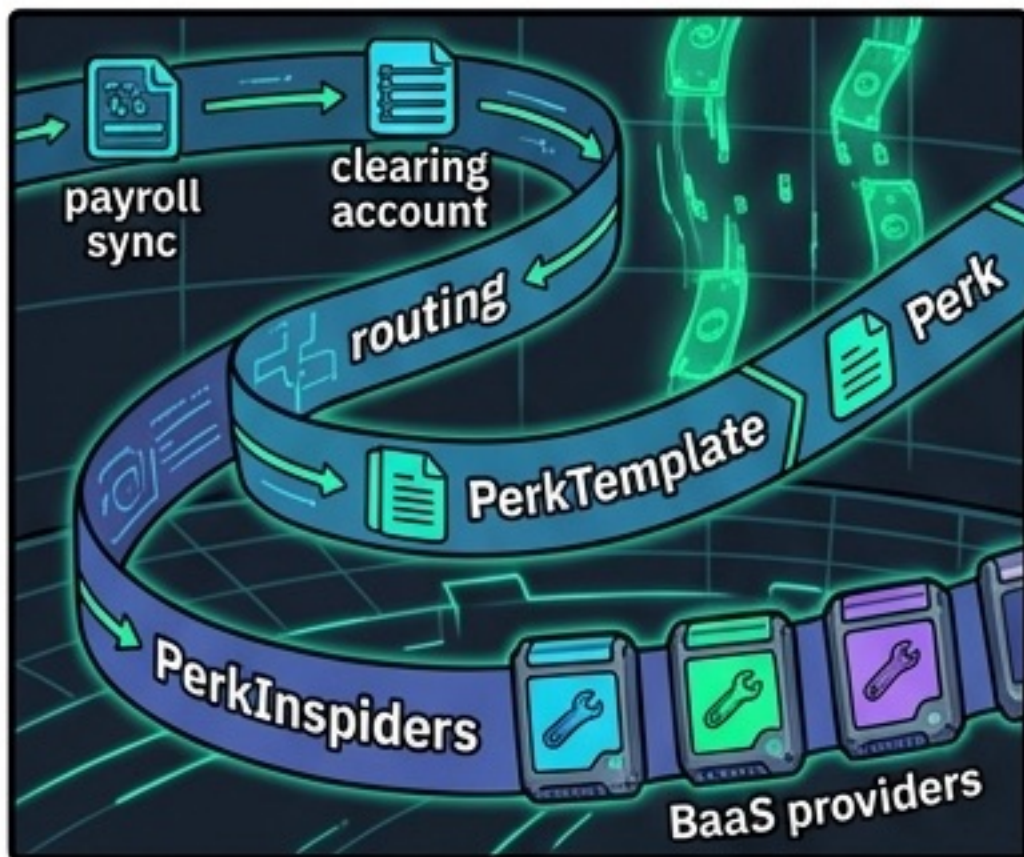
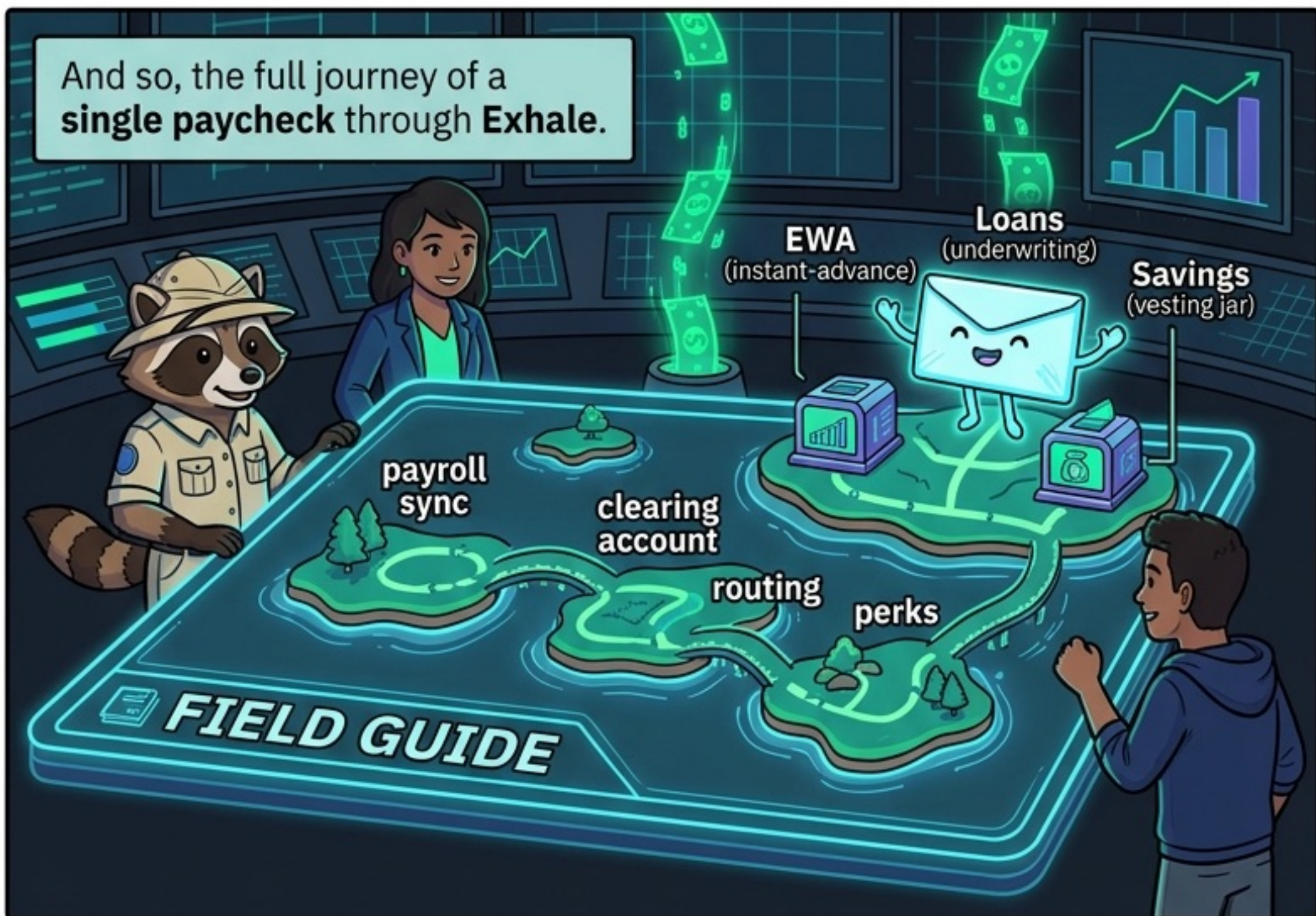


Typed errors, live metrics, a full audit trail. Money likes a paper trail.

FIELD HUD ▶

- packages/shared/errors.ts (PaloloError → Validation/Auth/Not)
- NewRelic metrics/events
- activity/audit log

And so, the full journey of a single paycheck through **Exhale**.



Accuracy and compliance,
all the way down.
Welcome to Exhale.



FIELD HUD ▶

run the suite: **x test**
flow = payroll → clearing →
routing → perks
(EWA/Loans/Savings) → **BaaS**