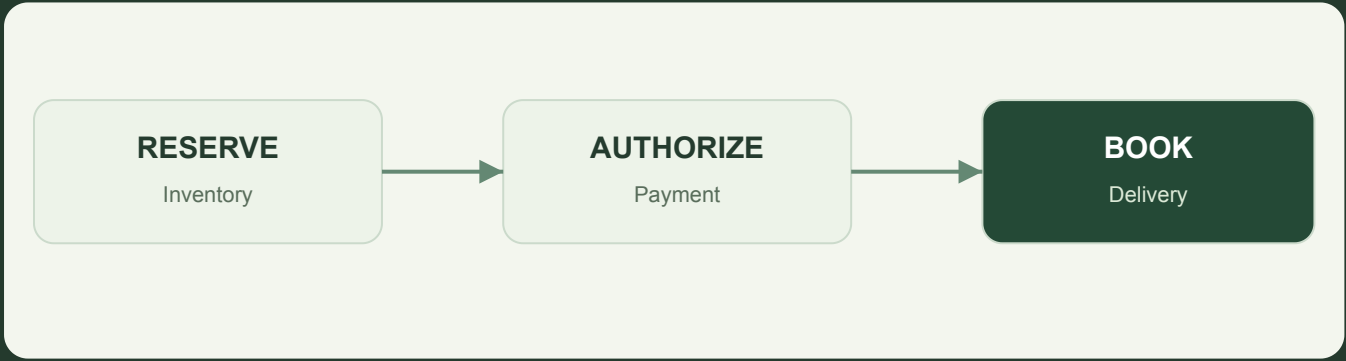
PROJECT DOCUMENTATION / 2026-09-06

Small failures. Better decisions.

A visual guide to API failures, recovery strategies and the evidence behind one order.



DESIGN & DEVELOPMENT

Tharinda.dev

READ IT. RUN IT. QUESTION IT.

Real HTTP · WSO2 API Manager · Observable outcomes

01 / START HERE

The project in one minute.

A successful response is useful. A correct business outcome is what matters.

> **Did the operation fail, or did we just lose its reply?**
>
> Rehearsal makes that question visible. It sends a sandbox order through inventory, payment and delivery, introduces one controlled failure, and checks what actually committed.

01 / Observe

Run a deliberately naive baseline. Watch the request trace and failed checks.

02 / Compare

Repeat the fault with a safer decision. Each experiment gets a fresh ledger.

03 / Explain

Read the effects, compare strategies and download evidence as PDF or JSON.

Find what you need

For curious users, students and technical reviewers. No real payments, deliveries or customer data are involved.

02 / QUICK START

Run your first rehearsal.

You can start in the hosted studio. No installation is needed.

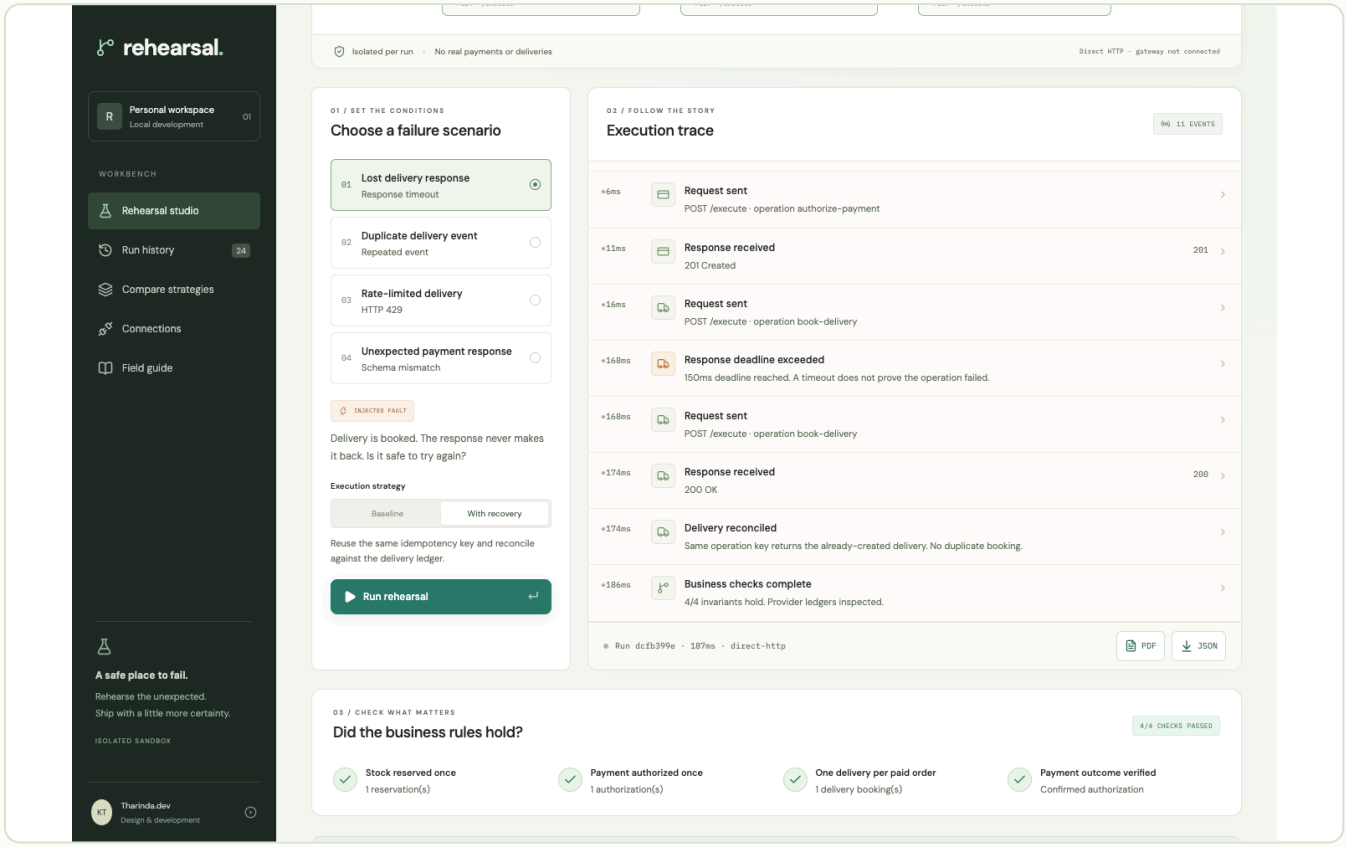


Figure 1. Actual local interface capture; hosted experiments use the same controls. Transport labels identify the execution mode.

1 **Choose a failure**

Leave Lost delivery response selected for your first attempt.

2 **Run Baseline**

The naive retry uses a fresh key. Inspect the delivery count, not just HTTP status.

3 **Run with recovery**

Reuse the original operation key in a new, isolated experiment.

4 **Compare and export**

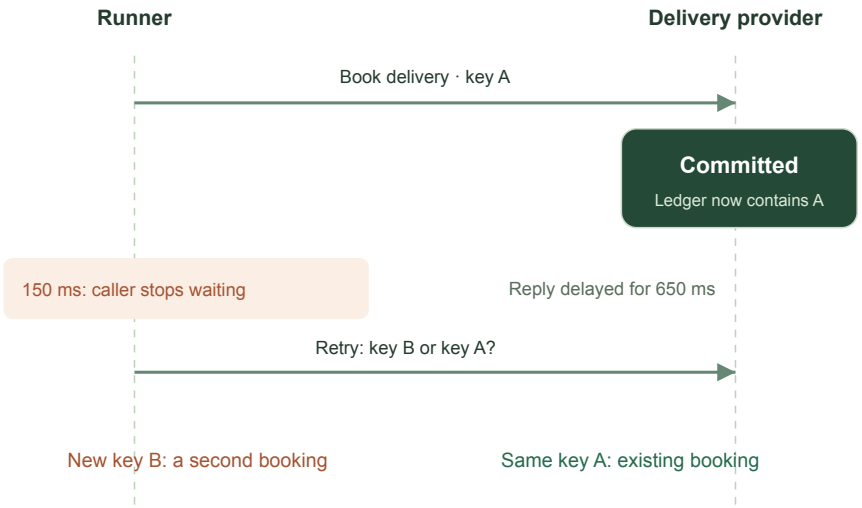
Compare strategies shows the latest pair. PDF is for reading; JSON is for inspection.

[Open the live studio → rehearsal-kumuditha.vercel.app](rehearsal-kumuditha.vercel.app)

03 / THE ENGINEERING PROBLEM

A timeout is not a rollback.

The caller and provider can disagree about what happened without either behaving mysteriously.



Baseline: new key

The provider treats the retry as another operation. One paid order can acquire two delivery bookings.

A + B

two operations

Recovery: same key

The provider recognizes the previous operation and returns its existing result, instead of creating another.

A + A

one operation

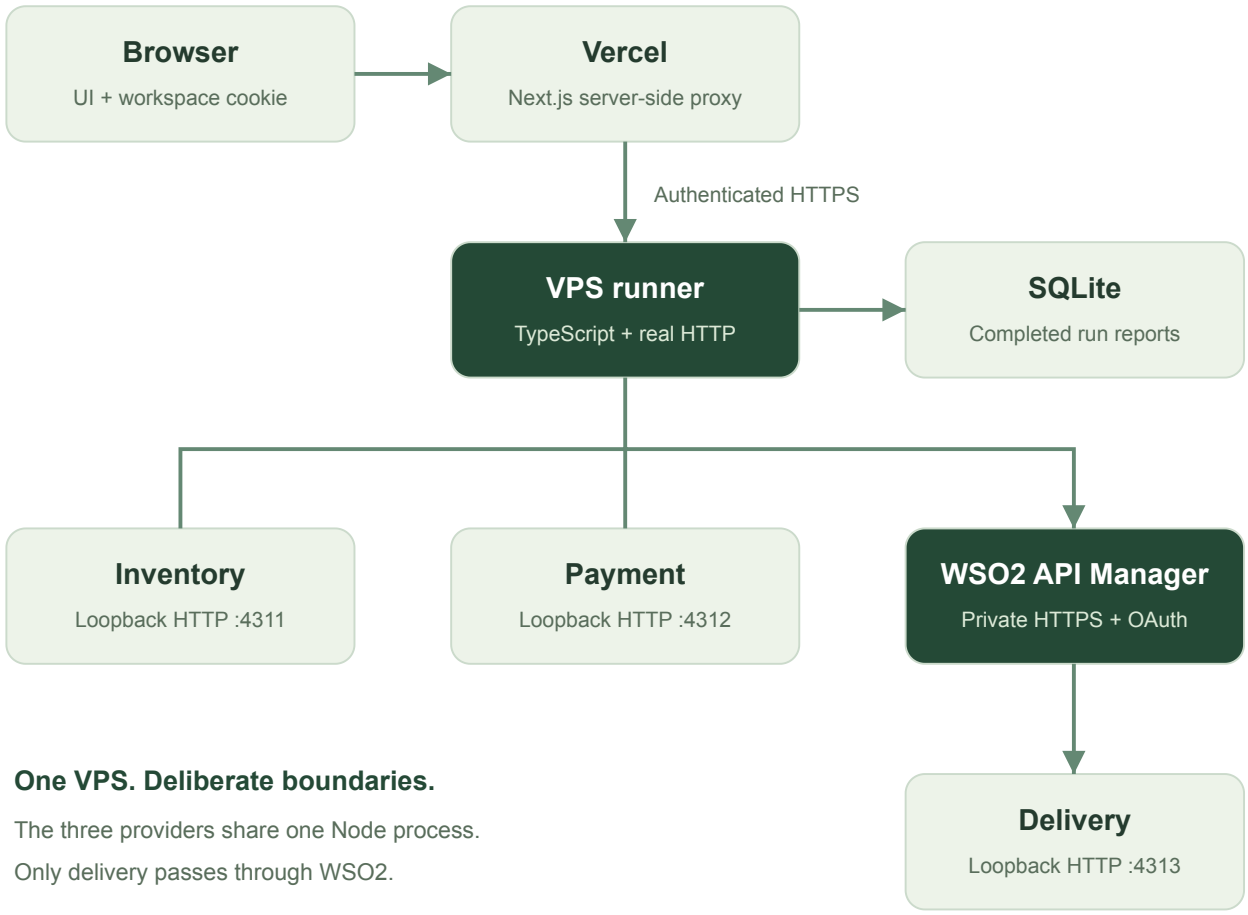
Idempotency, in plain English

Repeating the same logical request has the same business effect. Here, the key is scoped to one provider and one run. This in-memory implementation does not prove exactly-once delivery across replicas.

04 / ARCHITECTURE

Follow one request end to end.

The hosted version pairs a Vercel interface with an always-on DigitalOcean backend.



Why not everything on Vercel?

The provider listeners and SQLite history need an always-on process and persistent disk. Vercel hosts the interface and authenticated proxy.

Why only delivery through WSO2?

One narrow, verified managed boundary is easier to explain. Inventory and payment remain direct HTTP; the runner remains TypeScript.

Solid arrows show actual request or persistence paths. This is a single-node demo, not a high-availability topology. Ballerina is not in the live path.

05 / PROJECT STRUCTURE

Small modules. Clear ownership.

Each part has one job, so a reader can follow a run without learning the whole repository first.

```
Rehearsal/
├── src/app/ studio, layout and web API routes
├── src/lib/
│   ├── engine.ts workflow decisions and business checks
│   ├── gateway.ts server-side OAuth and selected routing
│   ├── store.ts SQLite report persistence
│   ├── report-pdf.ts browser-compatible PDF export
│   └── types.ts run contract and failure presets
├── sandbox/server.ts three loopback HTTP providers
├── vps-backend/ authenticated backend and deployment units
├── integrations/wso2/ OpenAPI, provisioning, verification
├── tests/ engine, backend, OAuth, PDF and browser checks
├── scripts/ launch, live acceptance and guide generation
├── docs/ decisions, setup notes and captured evidence
└── public/docs/ downloadable project guide PDF
```

The contract between layers

BOUNDARY	WHAT CROSSES IT
Browser → web route	Scenario and strategy, never a user-selected target URL.
Runner → provider	X-Rehearsal-Run plus a stable operation key.
Backend → browser	Streamed events, then a completed Run report.
Report → PDF renderer	Already-visible run data. No backend secrets or new execution.

The structure reflects the source reviewed for this guide. Commit at evidence capture: `f4bb587`.

06 / SCENARIOS

Four faults. Different decisions.

Choose the recovery that matches the failure; “retry everything” is not a strategy.

01 Lost delivery response

FAILURE

Delivery commits, then delays its reply beyond the 150 ms caller deadline.

RECOVERY

Reuse the same operation key. A retry resolves to the existing delivery.

The naive baseline can create two bookings.

02 Duplicate delivery event

FAILURE

The runner deliberately replays the same logical delivery trigger.

RECOVERY

Map both attempts to the same key so the provider suppresses a duplicate.

This is a replay inside the runner, not a webhook receiver or message broker.

03 Rate-limited delivery

FAILURE

The sandbox rejects the first delivery attempt with 429 and Retry-After: 1.

RECOVERY

Wait one second, then retry once with the same key.

This injected provider error is separate from the WSO2 quota probe.

04 Unexpected payment response

FAILURE

Payment commits, but its response has the wrong field name.

RECOVERY

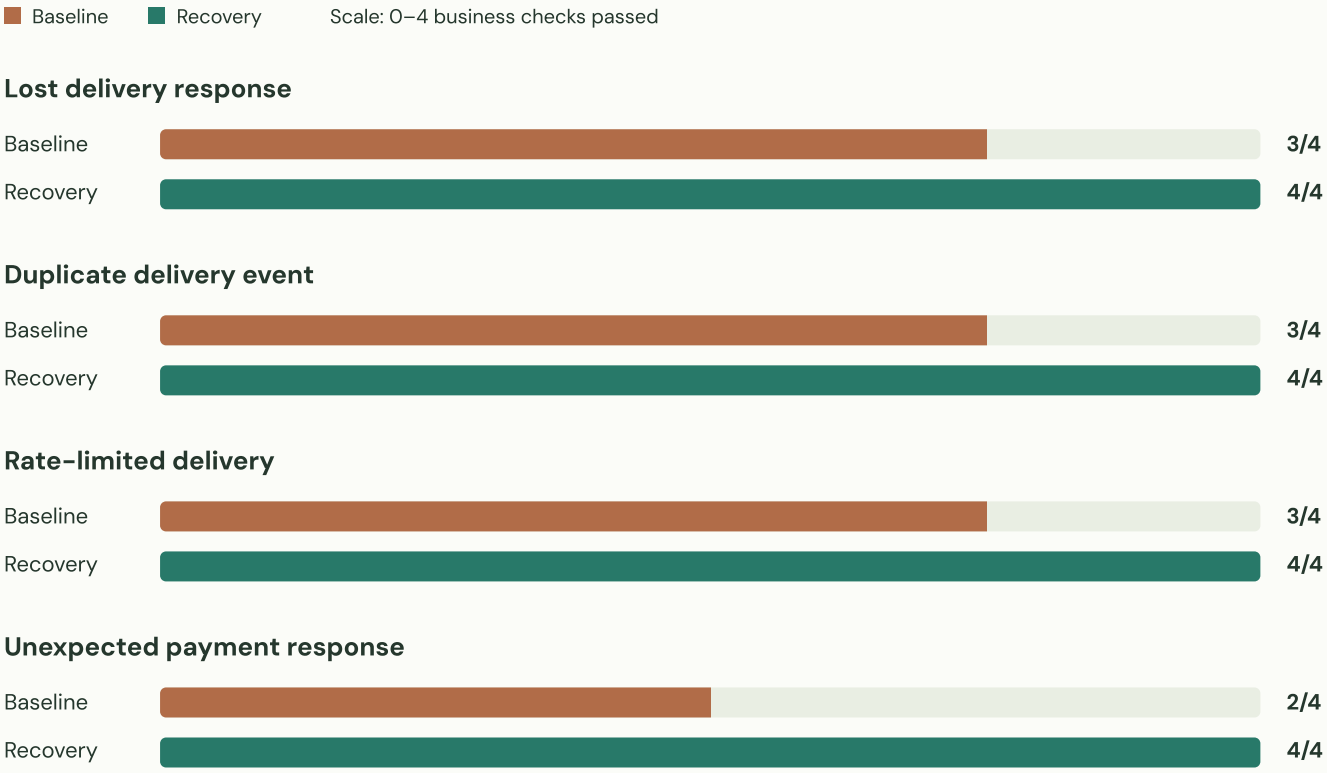
Validate the response and reconcile against the payment ledger before continuing.

Do not charge again just because a response cannot be parsed.

07 / OBSERVED RESULTS

What changed when we recovered?

Eight real hosted runs: one baseline and one recovery for each failure preset.



Delivery effects in the same runs

SCENARIO	BASELINE BOOKINGS	RECOVERY BOOKINGS
Lost delivery response	2	1
Duplicate delivery event	2	1
Rate-limited delivery	0	1
Unexpected payment response	0	1

Captured 2026-09-06T19:45:30.388Z from the public app. All reports used gateway-configured transport. A failed baseline is an expected experimental outcome, not a failed test. These eight observations are not a reliability percentage or a production benchmark.

Source data: docs/evidence/project-guide-runs.json. Full IDs, events, counts and checks are retained for reproduction.

08 / WSO2 INTEGRATION

Security is a separate question.

A gateway can reject an unauthorized call. It cannot choose a safe business key for the caller.

CHECK	OBSERVED RESPONSE	MEANING
Missing token	401 / 900902	Gateway rejected unauthenticated access.
Invalid token	401 / 900901	Gateway rejected invalid credentials.
Valid token	200	Managed delivery health was reachable.
Separate quota probe	429 / 900800	WSO2 enforced its own request quota.
Provider quota	429 / Retry-After: 1	Injected sandbox response survived the gateway.

Normal delivery API

`/rehearsal/delivery/1.0.0`

Execute, inspect the ledger and check health. Outer proxy and backend limits still apply.

Read-only quota probe

`/rehearsal/gateway-probe/1.0.0`

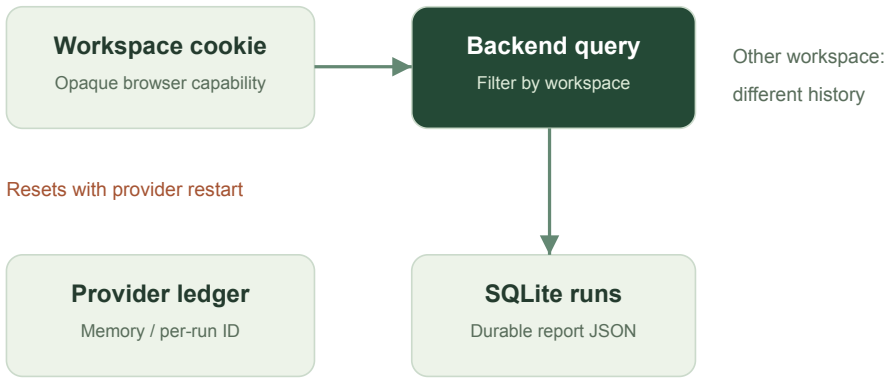
Two requests per minute. Kept separate so quota testing does not disrupt rehearsals.

Policy evidence comes from the verified 7 September 2026 record in docs/VERIFICATION.md. API Manager 4.7.0 runs privately on the VPS. A green UI health indicator proves connectivity, not every security policy.

09 / DATA & PRIVACY

Keep reports. Isolate experiments.

Provider state and completed reports have different lifetimes—and that distinction is intentional.



Actual SQLite table: runs

COLUMN	ROLE
id · TEXT PRIMARY KEY	Unique completed run ID.
started · TEXT NOT NULL	Run timestamp; used to sort the latest reports.
body · TEXT NOT NULL	Serialized Run: events, ledger counts and business checks.
workspace · TEXT NOT NULL	Query boundary; defaults to local for local use.

What survives

Completed SQLite reports survive backend restarts. The interface returns the newest 100 per workspace.

What does not

Provider ledgers are in memory. Clearing the cookie loses access to that workspace; no account recovery exists.

Cookies and runtime secrets are not included in documentation evidence. PDF export runs inside the browser using the report already loaded there. Downloaded reports are ordinary shareable files: review them before distributing.

10 / INTERFACE & EXPORTS

Designed to make evidence legible.

The interface follows the experiment: configure, observe, verify, then explain.

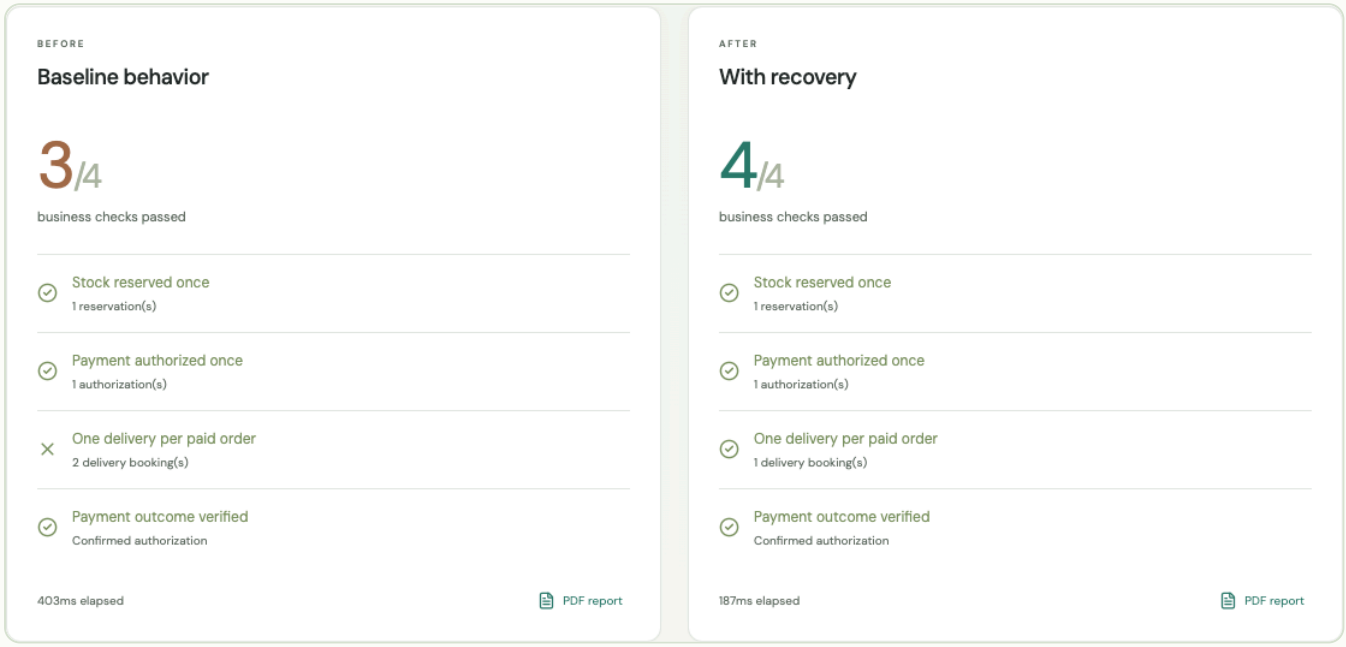


Figure 2. Actual local comparison view. Counts reflect its selected runs, not a decorative illustration.

Run report PDF

Readable summary, pass/fail checks, ledger chart, elapsed-event graph and the complete paginated trace. Original JSON is embedded as an attachment.

Project guide PDF

This illustrated document explains the system, code structure, deployment, evidence and limits. It is not a report for a particular visitor.

Small UX decisions that matter

- Visible PDF + JSON controls
- Keyboard focus and dialog dismissal
- Mobile layouts
- Reduced-motion support
- No fake progress metrics
- Explicit developer credit

Animations present observed events; they never change recorded execution times. PDF exports are not digitally signed certifications. JSON remains the machine-readable evidence format.

11 / BUILD & OPERATE

From checkout to the hosted lab.

Use Node.js 24. Start locally before changing anything on a deployed server.

Local development

```
npm ci
npm run dev

# Web: localhost:3040
# Providers: 4311–4313
```

Keep the terminal open. The launcher does not stop unrelated processes if a port is occupied.

Verification commands

```
npm test
npm run typecheck
npm run test:e2e
npm run build
```

Tests use controlled sandbox inputs. Browser checks can target the live app using PLAYWRIGHT_BASE_URL.

1. **Web on Vercel**
   Set BACKEND_URL, BACKEND_TOKEN and the expected app origin privately. Never use NEXT_PUBLIC_ for secrets.

2. **Backend on the VPS**
   Node service as the rehearsal user. Nginx terminates public HTTPS. SQLite is outside the source release directory.

3. **Private WSO2 runtime**
   Separate wso2 account, trusted localhost TLS and server-side client credentials. Public firewall permits only SSH, HTTP and HTTPS.

4. **Release and verify**
   Deploy a reviewed archive, retain the old release, and verify real runs. Do not overwrite private credentials or the report database.

Regenerating this guide

Run `node scripts/build-project-guide.mjs --capture` with the local studio running. It captures fresh public sandbox evidence and local screenshots. Without --capture it rebuilds from the retained assets. Review every rendered page before publishing.

Detailed maintenance: docs/DEPLOYMENT.md and docs/WSO2.md. Private TLS renewal and off-server backup work are operational responsibilities, not features guaranteed by this PDF.

12 / VERIFICATION & NEXT STEPS

Useful evidence. Honest limits.

A professional project report should make it easy to see both what works and what is unfinished.

Verified in the project

- Four baseline/recovery pairs over real HTTP.
- Gateway authentication and separate quota enforcement.
- Workspace separation and report persistence across a backend restart.
- Automated engine, backend, OAuth and browser checks.

Not claimed

- Production throughput or reliability guarantees.
- Durable provider deduplication across replicas.
- Live Ballerina execution, team accounts or arbitrary API import.
- Off-server restore, high availability or independent security certification.

A practical next iteration

01 / Make effects durable
Persist provider operation keys with a uniqueness constraint.

02 / Audit independently
Add and verify a typed Ballerina ledger auditor.

03 / Harden operations
Review subscriber privileges, backups, restore and certificate rotation.

Each improvement above is a next step, not a claim about the current release. Keep acceptance results with the commit that was actually tested.

13 / GLOSSARY & REFERENCES

Keep asking the right questions.

A short vocabulary for reading the project and discussing its engineering decisions.

Glossary in plain English

API gateway	A managed checkpoint in front of an API.
Idempotency key	A stable identity for one logical operation.
Ledger	The provider's record of committed effects.
Reconciliation	Check the record when a reply is ambiguous.

Source and further reading

Project source, README and verification record — GitHub

WSO2 API Manager — official documentation

PDF-LIB — PDF generation API

Prepared for accessible project review, not as an endorsed WSO2 case study. Design and development credit: Tharinda.dev. Evidence capture: 2026-09-06T19:45:30.388Z.