One AI harness.
A workspace for everyone.

How LibreChat built multiplayer sandboxes
with open-source components



ClickHouse

Danny Avila · LibreChat

The companies building with LibreChat are saying it publicly.

@nejatian

Opendoor

“We are launching a whole bunch of MCPs internally ... all connected to our internal chat system built on LibreChat.”

Opendoor Assistant shipped too.

@brian_armstrong

Coinbase

LibreChat cache hits rose from 5% to 60% while token use climbed and AI spend fell nearly by half.

More usage. Lower cost.

@tobi

Shopify

“Shopify runs an internal fork of LibreChat ... I highly recommend other companies give this project a look.”

“It works very well for us.”

Close the laptop. The work should survive.

Most work needs a result

- quick calculations
- spreadsheet cleanup
- file conversion
- one-off analysis

Some work needs an environment

toolchains

project context

credentials

dependencies

long-lived files

contracts

debugging

repeatable

statefulness

The product requirement

fast by default, durable when needed

Isolation is a lifecycle, not a container flag.

Every execution needs six answers

WHO	tenant · user · agent
WHERE	stateless or stateful
SCOPE	user · agent-user · agent
AUTH	least-privilege credentials
LIMIT	time · memory · concurrency
END	commit · release · expire

Three rules that survived contact with production

1. **Resolve on the server**

   The model can request a tool. The host chooses the environment.

2. **Carry identity end to end**

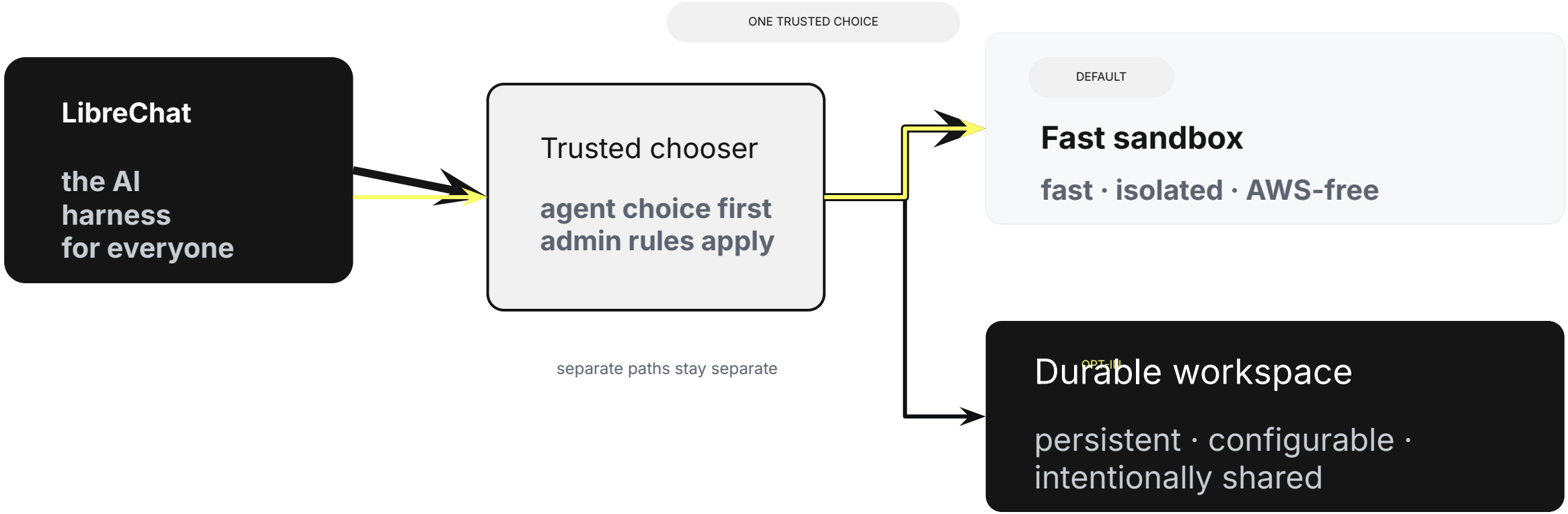
   Jobs, workers, files, and traces carry the same context.

3. **Fail closed**

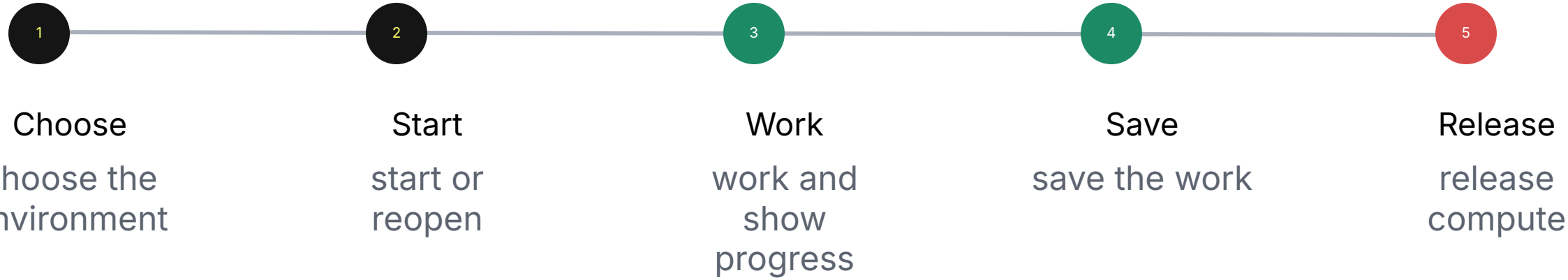
   Never switch workspaces silently.

LibreChat routes execution per agent.

One multi-agent system can safely mix both.



Keep the work. Release the runtime.



Durable

transcript · files · workspace · outcome

Disposable

running process · temporary controls

The bugs lived between components, not inside the sandbox.

01

The choice got lost

The UI offered a choice. The backend still made one choice for everyone.

Carry the choice all the way through.

02

Retries broke

After an upgrade, the same retry could point to something different.

Make every retry mean the same thing.

03

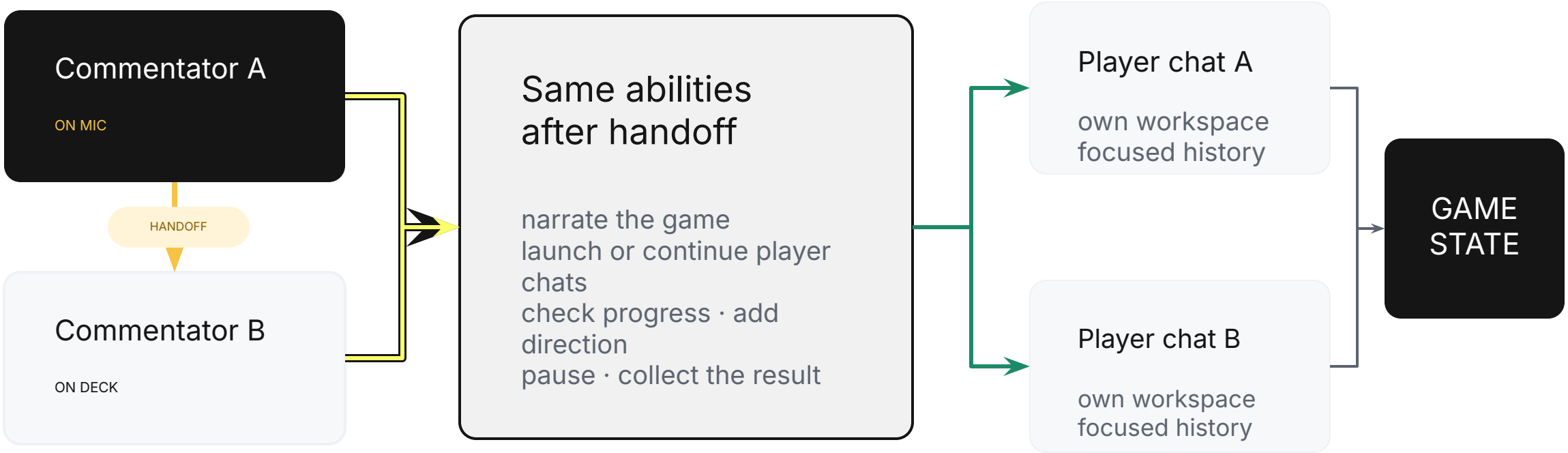
Tests missed reality

Real environments exposed restore and file failures our tests never saw.

Test the real environment before release.

A speed-chess broadcast is a stress test, not a special case.

EMERGING



One harness. Different environments.

1 **Default to speed**

Most work pays only for the execution it needs.

2 **Let work keep its space**

Set up once, continue later, replace compute safely.

3 **Choose per agent**

Mixed teams can safely use both execution profiles.

4 **Keep control with the host**

The host chooses identity, workspace, and infrastructure.

Open source, end to end

LibreChat · @librechat/agents · LibreChat Code Interpreter

github.com/LibreChat-AI

ClickHouse

Connect with ClickHouse

slack

Exclusive for the meetup: $400 cloud credits!

Try ClickHouse for your use case

- ClickHouse Cloud
- Download open source

Learn

- Academy / certifications
- Blogs / YouTube

Engage with our community

- Community Slack
- Monthly Community calls
- Meetups / events

We are Hiring. Come Work with Us!

©2025 CLICKHOUSE INC., CONFIDENTIAL & PROPRIETARY

ClickHouse

ClickHouse

Checkout ClickHouse Agents (Beta)





Exclusive for the meetup:
$400 cloud credits!

©2025 CLICKHOUSE INC., CONFIDENTIAL & PROPRIETARY

ClickHouse