

LOCAL-FIRST SYSTEM KNOWLEDGE

AKASHIC

Your systems,
remembered.

\$ akashic generate



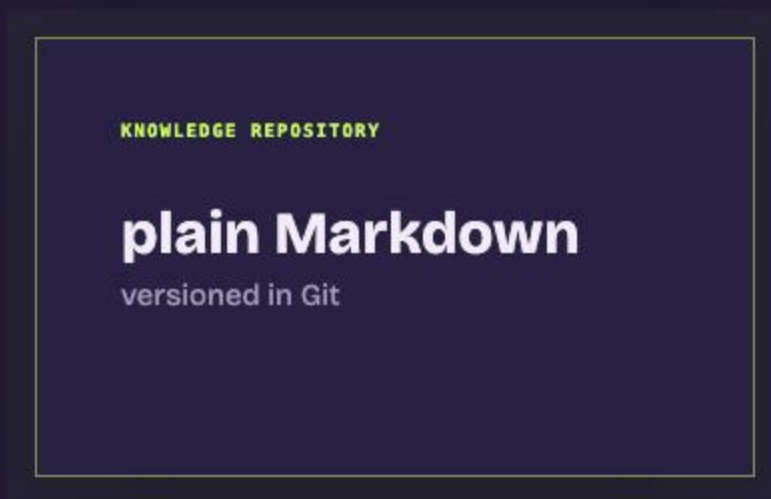
THE FRACTURE

The code is connected. The knowledge is not.



THE PROMISE

One repository becomes the memory of the system.



LOCAL

PORTABLE

LEGIBLE

AUDITABLE

THE LOOP

Four commands. One living knowledge base.



SOURCE REPOS STAY IN PLACE • THE SESSION STAYS VISIBLE • GIT STAYS UNDER USER CONTROL

START LOCAL

akashic init

Git from minute one

safe to run twice

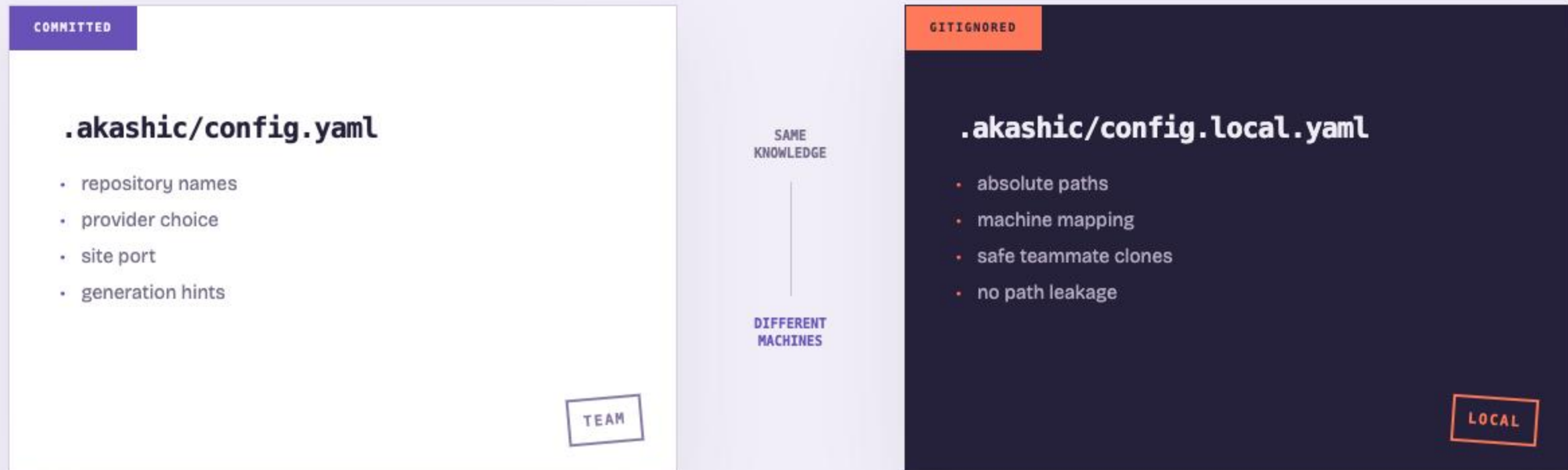
standard layout

• • • ~/knowledge

• LIVE

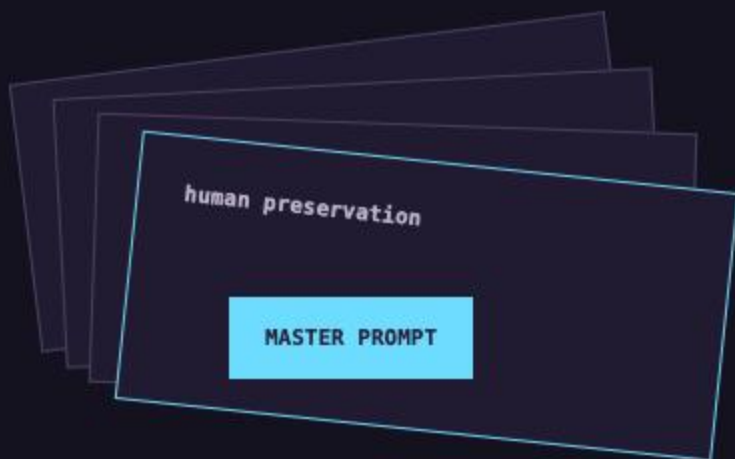
```
$ mkdir system-memory && cd system-memory
$ akashic init
created    .akashic/config.yaml
created    services/ flows/ system/ entities/
created    adr/ glossary/ README.md
initialized Git repository
committed  Initialize knowledge repository
ready      attach a source repository
```

Shared intent. Personal paths.



GENERATION

One composed brief enters. A system map comes back.



interactive session



● ● ● generation session

● LIVE

```
$ akashic attach ../payments
attached payments /repos/payments
$ akashic generate
context 3 repositories
prompt 5 document templates composed
session interactive authoring started
written services/payments.md
written flows/checkout.md
written system/overview.md
summary 14 knowledge files changed
```

Generated structure. Protected human judgment.

```
generated: true  
type: service  
name: payments  
sources: [/repos/payments]  
updated_at: 2026-07-14T...
```

REFRESHABLE

**Service facts, interfaces,
flows, and diagrams**

HUMAN START

**Decisions, caveats, and team memory
survive regeneration.**

HUMAN END

machine-written facts



human-owned meaning

PRODUCT BOUNDARY

A conductor, not another orchestra.



THE EXPERIENCE

Knowledge should feel navigable, not merely stored.

The screenshot shows a web application running on localhost:6969. The interface is divided into a left sidebar and a main content area. The sidebar contains a search bar with 'payments' entered, and a list of categories: SYSTEM, SERVICES, payments (highlighted), identity, FLOWS, ENTITIES, and ADRS. The main content area displays the 'Payments' service details, including a description, statistics for endpoints, flows, and decisions, and a flow diagram.

localhost:6969

SEARCH KNOWLEDGE

payments

SYSTEM

SERVICES

payments

identity

FLOWS

ENTITIES

ADRS

SERVICE

Payments

Owns payment intent creation, confirmation, and ledger handoff.

GENERATED

12 endpoints

04 flows

03 decisions

intent → confirm → ledger

TRUST MODEL

Control stays visible at every boundary.

SOURCE CODE	registered in place	not moved by Akashic	
GENERATION	interactive session	permissions approved live	
GIT	no staging or commits	the user owns every diff	
KNOWLEDGE	plain local files	inspect with any editor	

KNOWN LIMITATION

Source-repo write protection is prompt-enforced in v1. Generate against clean, committed trees.

MVP BOUNDARY

Sharp scope is a feature.

PROVE NOW

init, attach, detach, list

interactive generation

two provider adapters

structured documents

local site + static build

doctor + status

THE LOOP

source
↓
understanding
↓
Git
↓
browser

HARDEN LATER

cloud backend or auth

headless scheduling

deterministic orchestration

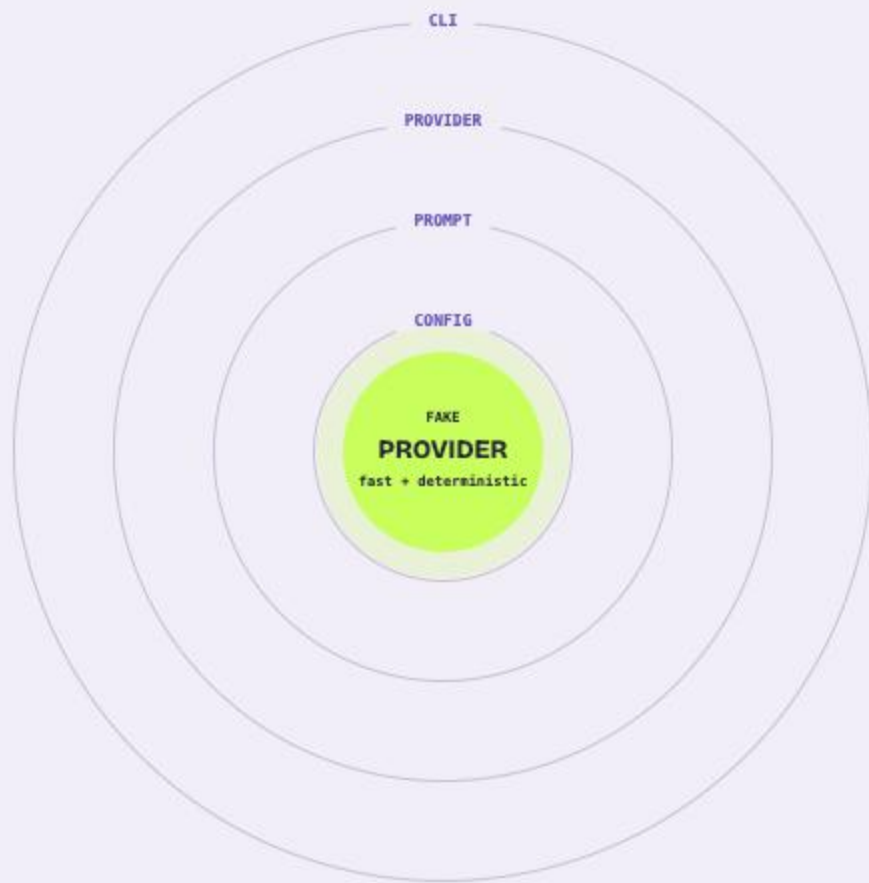
automatic merge engine

in-browser Git workflows

knowledge graph

VALIDATION

Test the behavior users can observe.



CLI

full user workflows

PROVIDER

Invocation contract

PROMPT

composition + overrides

CONFIG

merge + validation

GIT

observable repository state

SITE

static build contract

THE OUTCOME

Systems forget. Git remembers.

Akashic connects the two.



`$ akashic serve`

