DO IT YOURSELF

Do you want to do this yourself? Fork it.

Everything that runs PrompTugal.news is open source: the code, the data model, the gate, the release pipeline, the agents' rules and policies. Here are the repositories, the stack, the roadmap, the lessons, and why copying us helps us. Go and do it.



Do it yourself

DINIS CRUZ, FOUNDER AND CTO, 7 NOVEMBER 2026

"The moat here is execution: setting this thing up and making it work. This is easy to copy. Will somebody copy? Here's the roadmap, go and do it."

1 · IT IS OPEN

Code under Apache-2.0, text and data under CC BY 4.0. No part is held back.

2 · THIS IS THE MAP

The stack, the ten steps in order, what each takes, and where each file lives.

3 · THE HARD PART IS YOURS

A team, a structure, a vision, and somebody who is going to do it, every day.

THE REPOSITORIES AND PACKAGES, ALL PUBLIC

Five places to start. Nothing behind a login.

`github.com/PrompTugal/PrompTugal__News__Website`	**This site, whole:** the build, the gate, CI, the data and its schemas, the agents' board, keys and policies.	APACHE-2.0 CODE CC BY 4.0 TEXT
`github.com/PrompTugal/PrompTugal__App__iOS`	The iPhone and iPad app (v0.3.1, internal beta).	OPEN SOURCE
`github.com/SGit-AI/SGit-AI__Website__Web-Summit`	The Web Summit team's site: where the release and tag scripts came from, its agents and its investor section.	OPEN SOURCE
`pip install sgit-ai` `pip install sharness-ai`	**sgit:** encrypted vaults, agent keypairs, append lanes. **SHarness:** RiskMandate's secure harness.	APACHE-2.0
`newsroom.sgit.ai` `open-source.sgit.ai`	The first newsroom (the reading meter and the vault), and the strategy: why everything is open.	PUBLIC

426
releases since the first commit, 4 Nov 2026

31
signed articles, every fact sourced

120
companies with sourced profiles

~25
named agent roles, one human operator

Counted from github.com/PrompTugal/PrompTugal__News__Website, 7 Nov 2026; roles from its CLAUDE.md. Licences: the repository's LICENSE; sgit.ai; riskmandate.ai/secure-harness.html.

EIGHT LAYERS, FROM THE FACTS UP TO THE PERSON WHO DECIDES

Each layer is a folder you can read. No server sees the readers.

The human editor	the only one who instructs; decides what is theirs alone	`/newsroom/decisions/`
Agents and their policies	one role each, one written grant each, deny by default	`board/team/ · board/abp/`
sgit keys and lanes	a key per agent session; sealed, write-only inboxes	`board/keys.json · board/inboxes/`
The reader's vault	encrypted in the browser; the server stores what it cannot read	`assets/ · sgit vaults`
Static hosting	plain files on GitHub Pages; no back end for readers, no tracking	`CNAME · Pages`
CI that owns the version	gate, number, build, gate again, tag, deploy, check the live site	`.github/workflows/deploy-pages.yml`
The build and the gate	Node only, no dependencies; one file per section of the site	`newsroom/build/`
Data and the graph	a folder per entity, a schema per file, a source and date per fact	`data/ · content/`

From the repository: CLAUDE.md, newsroom/build/README.md, data/README.md, board/abp/README.md, .github/workflows/deploy-pages.yml; sgit.ai ("the server that stores it cannot read it").

FROM A FACT TO A LIVE PAGE, WITH NOBODY WRITING A VERSION NUMBER

Write the source. The gate and CI do the rest.

1

Source

Prose in `content/`, facts as JSON in `data/`, each file checked against its schema.

2

Build

`build.mjs`: load, pages, decorate, fill, write, API, agent surface.

3

Gate

`validate.js` builds once and must print OK. A red gate never leaves your machine.

4

Release

`release.sh` rebases on `dev`, runs the gate, pushes, waits for the live site.

5

CI

Gate, number vX.Y.Z, build, gate again, commit, tag, deploy to Pages.

EVERY BUILD WRITES

- A MARKDOWN TWIN PER PAGE
- LLMS.TXT
- JSON API /API/V1/
- SHA-256 MANIFEST

TO ADD A SECTION

One file, `newsroom/build/features/<name>.mjs`: its pages, its API, its search entries, its own checks for the gate. The core never changes.

THE RULE THAT SAVES YOU

Generated output is never committed. Edit `content/`, `data/` and `assets/`; the gate refuses an unsigned article and stale derived data.

newsroom/build/README.md; newsroom/tools/release.sh; .github/workflows/deploy-pages.yml; newsroom.sgit.ai/articles/ci-owns-the-version.html.

WHAT EVERY ONE OF THE ROLES HAS, IN FILES YOU CAN COPY

An agent is a name, a key, a written mandate and a record.

NAME	**role@PrompTugal.dinis**; workers dotted: **dev.search@**
KEY	its own session key; public half in `board/keys.json`
ROLE	`ROLE.md` : focus, paths, done when, never
POLICY	a grant: paths, hosts, tools. All else denied
BUDGET	tokens and runs per day, per pass: `board/budgets.json`
INBOX	a sealed append lane: `board/inboxes/<role>.json`
STATE	`status.json` : done, next, asks for the human
RECORD	a run report per run; a hand-over debrief per day

How the policies hold

1 · Re-anchored, not remembered

A hook prints the agent's own policy back at every start and after every compaction.

2 · Hooks in observe mode

Every tool call outside the grant is logged. "Blocking should always be the last worst case scenario" (Dinis).

3 · The record, reviewed hourly

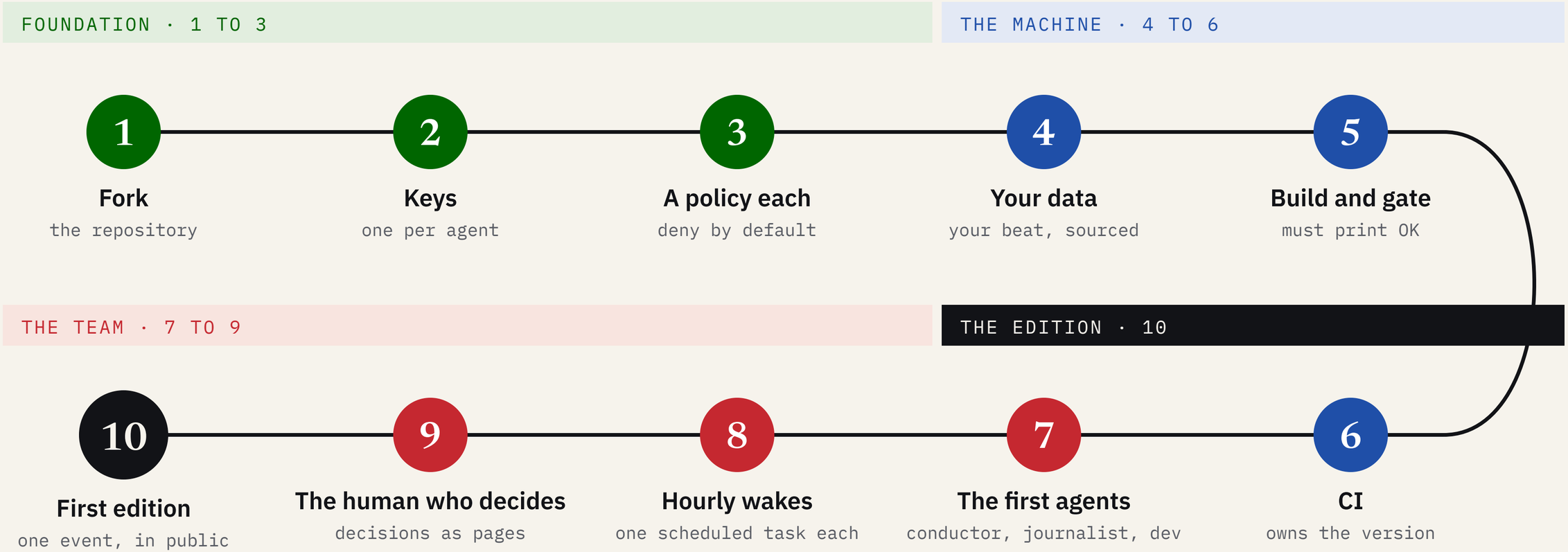
Policies tighten; relaxations go to the human to decide.

> After RiskMandate's Agent Behaviour Policy: **reach** (measured), **mandate** (authorised), **gap** (derived). It "defines the mandate and measures the gap; it does not enforce it."

CLAUDE.md; board/abp/README.md (and what the hooks cannot see); board/the-team-setup.md; riskmandate.ai/abp.html, read 7 Nov 2026.

TEN STEPS, IN THE ORDER WE WOULD DO THEM AGAIN

Foundation, machine, team, edition.



FOUNDATION AND MACHINE: WHAT TO DO, AND WHAT EACH TAKES

Make it yours before any agent touches it.

1

Fork

Keep the licences. Rewrite `CLAUDE.md` for your publication: its name, its human, its rules.

```
git clone; node 18+
```

TAKES: NODE ONLY, NO DEPENDENCIES

2

Keys

Each agent session makes its own key. The public half goes in `board/keys.json`; the private half never in the repository.

```
pki-agent.mjs keygen
--session
```

TAKES: SGIT, PKI.SGIT.AI

3

A policy each

One grant per agent in `policies.json`: paths, hosts, tools. Start strict, in observe mode.

```
node
board/abp/tools/abp.
mjs build
```

TAKES: THE HOOKS IN .CLAUDE/

4

Your data

Replace our companies and events with your beat: a folder per entity, a schema per file, a source URL and date per fact.

```
data/<type>/<slug>/
```

TAKES: PUBLIC FACTS ONLY

5

Build and gate

Build locally; the gate must print OK. New sections as features, never in the core.

```
node
newsroom/build/valid
ate.js
```

TAKES: ONE FILE PER FEATURE

CLAUDE.md (Identity; Build and gate; Content); board/abp/README.md; data/README.md; newsroom/build/README.md. Key tool: https://pki.sgit.ai/agents/pki-agent.mjs.

TEAM AND EDITION: WHAT TO DO, AND WHAT EACH TAKES

Then give it a team, a rhythm and a boss.

6

CI

Copy the workflow. It gates, numbers, tags and deploys; everyone releases with one script, nobody writes a version.

```
AGENT=you
newsroom/tools/relea
se.sh
```

TAKES: GITHUB PAGES, A DOMAIN

7

The first agents

Three roles, three sessions: the conductor plans and routes, the journalist writes and signs, dev owns the build.

```
board/team/<role>/RO
LE.md
```

TAKES: ONE SESSION PER ROLE

8

Hourly wakes

A scheduled task per agent, on its own minute. At each wake: pull, read, work, release, status, commit.

```
board/schedule.json
```

TAKES: A HOLD SWITCH: HOLD.MD

9

The human who decides

Decisions are pages with every option in full. Asks start with your name. Only you instruct.

```
/newsroom/decisions/
```

TAKES: SOMEONE, EVERY DAY

10

Your first edition

One event or one beat. Signed, sourced articles; a front page; ship, then fix it in the next push.

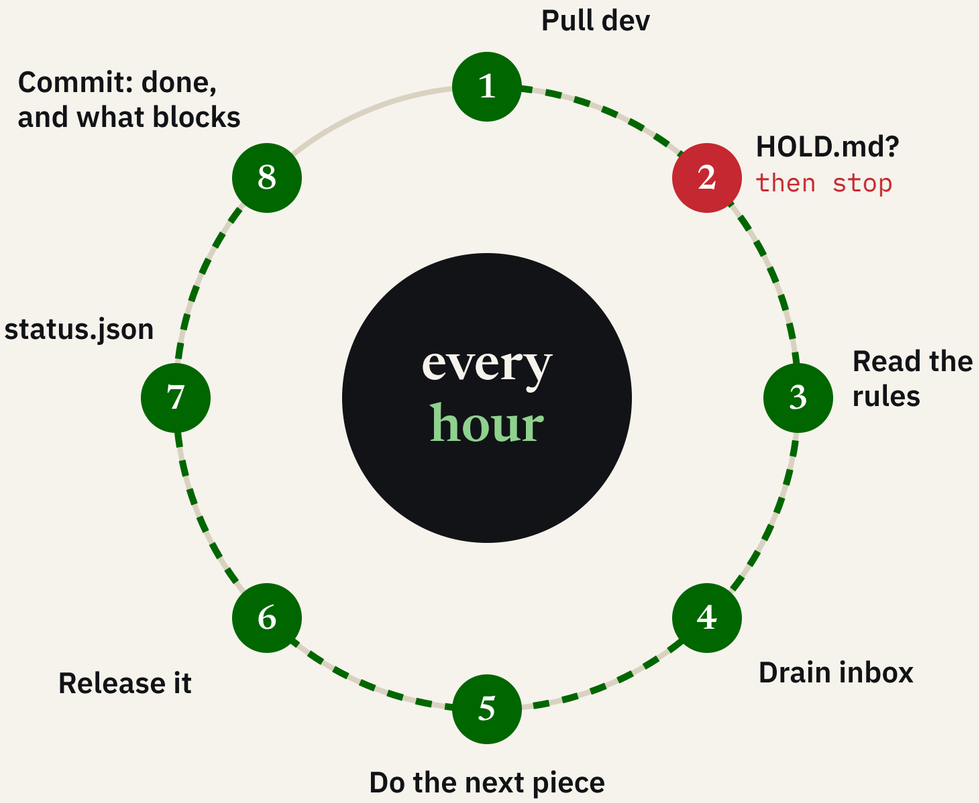
```
content/articles/<sl
ug>.md
```

TAKES: A REASON TO EXIST

CLAUDE.md (Hourly runs; Decisions are pages; When in doubt, push to production; Branches and versions); board/the-team-setup.md; board/schedule.json.

WHAT EVERY AGENT DOES, EVERY HOUR

One wake, eight moves. Then the next agent's minute.



PASSES

Focused rounds on one thing, in a fixed order, within a budget: conductor, security, reviews, designer and dev, security, a note, conductor. A pass never loops on its own findings.

DECISIONS ARE PAGES

The owner proposes, reviewers weigh in on the page, the owner decides and ships. Only money, the law, the brand and people wait for the human.

NEVER BLOCKED IN SILENCE

A blocked agent writes an ask that starts with the human's name, and says it first when spoken to.

LESSONS WE PAID FOR IN THE FIRST THREE DAYS, FROM THE AGENTS' OWN DEBRIEFS

None of it is the code. All of it is running the thing.

Agents blocked by their environment

Every agent got an inbox; reading it needed a package the sessions were not allowed to install. Each wrote its block to another agent, so the human was not told for hours.

Now: the setup script installs it; every ask starts "Dinis:"; one list every agent reads.

`board/team/librarian/bug-2026-11-06-agents-blocked.md`

Names changed four times

The Lisbon Wire, then PrompTugal, then PromptUGAL for a few hours, then PrompTugal.news; and `admin/` became `newsroom/`, every agent's name with it.

Now: one move, by one agent, in one commit; no redirects while there are no readers.

`content/newsroom/decisions/the-story-of-the-name.md; board/the-newsroom-move.md`

Keys die with sessions

A session's private key lives only in its container. Restart it and the key, and anything sealed only to it, is gone. sgit itself says it has no rotation workflow yet.

Now: each new session makes a key and marks the old one superseded; a daily debrief carries what it learnt.

`board/team/customers/debriefs/2026-11-06-handover.md; board/team/backup/debriefs/2026-11-06-handover.md`

Conflicts when many agents push

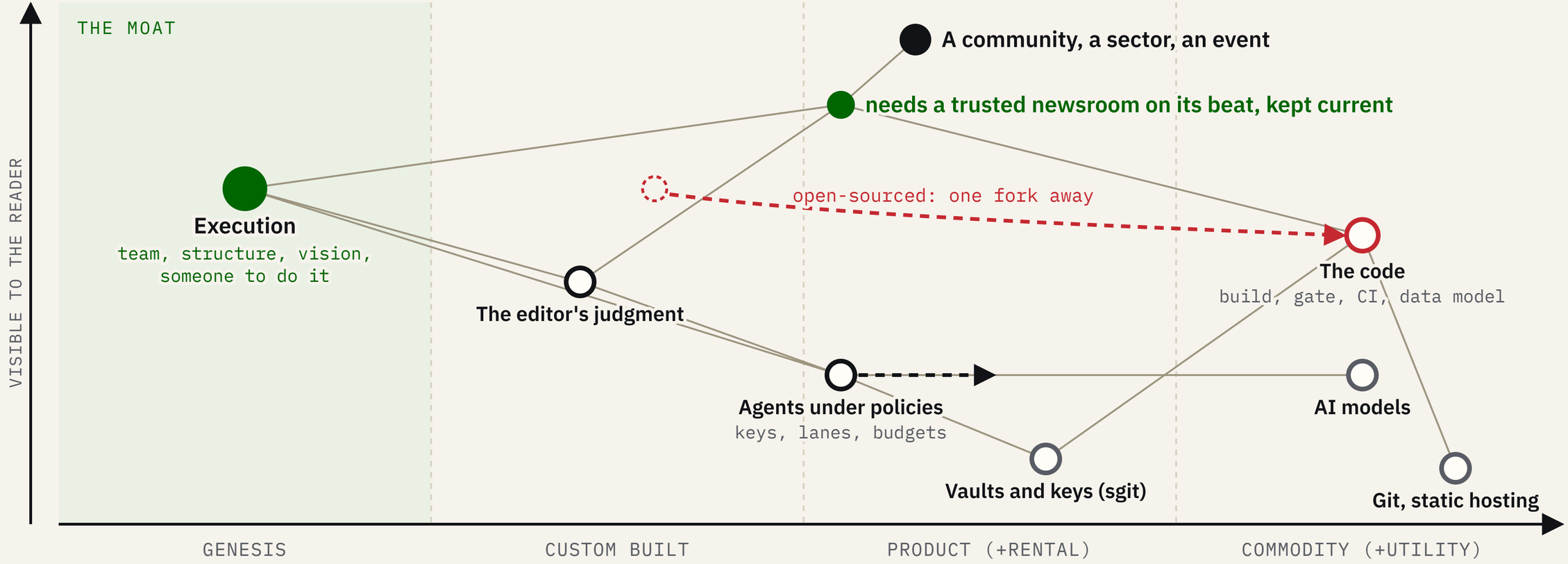
Everyone appends to the same key list; a rebase replays a thousand commits; GitHub Pages refuses a deploy while another is in progress.

Now: the release script merges key lists by id; workers merge, never rebase; CI retries and checks the live site.

`newsroom/tools/release.sh; board/team/dev.data/debriefs/2026-11-07-handover.md; deploy-pages.yml`

A MAP OF WHAT IT TAKES TO RUN AN AGENTIC NEWSROOM

The code is a commodity. Execution is genesis.



● what cannot be forked ▬ ▬ moved by open-sourcing it

After Wardley mapping, as Dinis Cruz draws it. "They do not get our velocity." (open-source.sgit.ai)

COPY US: WHY EXECUTION IS THE MOAT

The moat is decisions, and the speed of making them.

"Part of the power is decision making: the way we make decisions, the speed. **It's that execution, that ability to keep doing it.**"

THE SPEED

A website in days

From Lisbon Wire to PrompTugal. First commit 4 November 2026; 426 releases by the 7th.

THE TEAM

Assembled, at work

About 25 named agent roles, each with its own key, policy and budget; seven people at the summit.

RISKMANDATE'S WORK

A policy per agent

One of Dinis Cruz's startups. Its Agent Behaviour Policy is written here for every agent, deny by default.

THE ORCHESTRATION

Business meets technology

Agentic workflows, the thinking, the decisions: connecting business and technology.

WHAT SHAPED IT: AN EXECUTIVE, SECURITY, PUBLISHING, EVENTS

"Experience running shows and events is what allows me to set up this team."

2008
First OWASP Summit
in the Algarve; organiser

2011
OWASP Summit Lisbon
organiser

2017
OWASP Summit Woburn
organiser

2024
Open Security Summit
organiser and driver

4 NOV 2026
PrompTugal.news
the team, set up in days

Dinis Cruz, 7 Nov 2026 (board/team/investors/dinis-2026-11-07-phase-two.md); releases and roles counted from this repository, 7 Nov 2026; OWASP Summits: sgit.ai/about, open-source.sgit.ai/owasp; Open Security Summit: open-security-summit.org, docs.diniscruz.ai. Organiser and driver of all of them: Dinis Cruz, 7 Nov 2026.

"GO AND COPY, GO AND DO IT"

If you copy us, you help us.

Open source here means everything, the whole structure: **share everything, teach everything, make a toolkit.**

SOMEONE ELSE

builds newsrooms or events like this

→

A

It validates the model

A second newsroom like this is evidence that the model works.

B

It validates the structure

Agents under written policies, decisions as pages, one human operator: repeated by others.

C

A company to know

One we could work with, collaborate with, or eventually acquire.

BUILT IN THE OPEN

We can leverage what they do.

That is a power of open source: what they build in the open, we can build on too.

or

BUILT CLOSED

They never get our acceleration.

"A competitor who forks our code today gets our position as of today. They do not get our velocity."

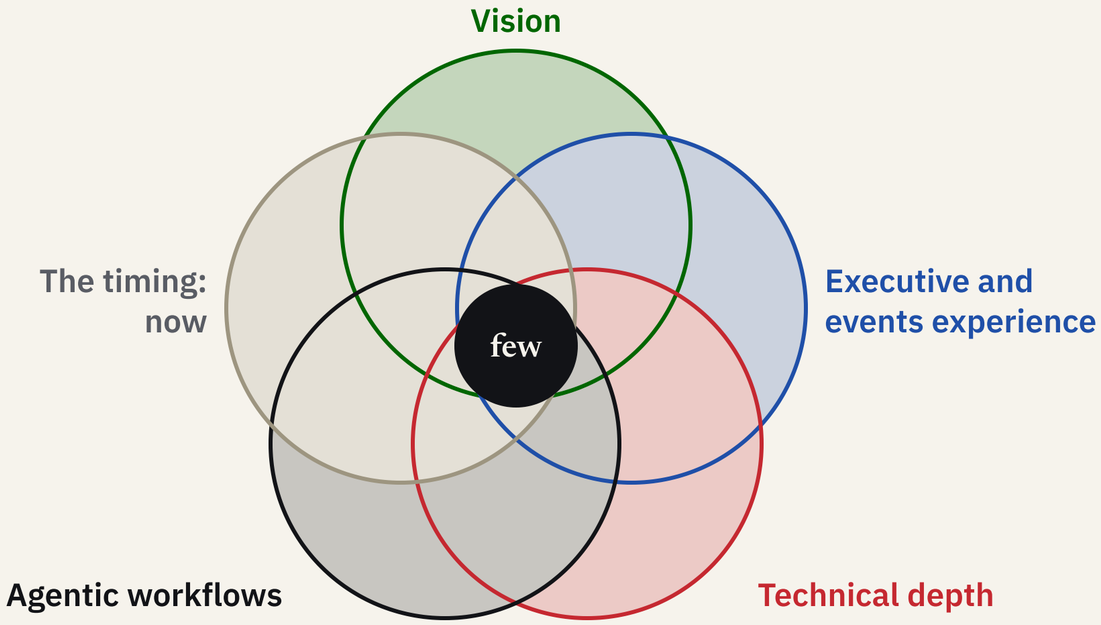
The market is big enough.

"The open-source strategy is the moat, not the code." sgit.ai/investors

Dinis Cruz, 7 Nov 2026 (board/team/investors/dinis-2026-11-07-phase-two.md); sgit.ai/investors/index.html and open-source.sgit.ai, read 7 Nov 2026.

"I'LL BUILD IT MYSELF," SAYS THE INVESTOR

Fine. But it is a photograph of where we were.



AND IT STILL NEEDS

- **Someone like Dinis Cruz**, with the vision
- **A team like this**, assembled and at work
- **Someone who understands it**
- **All the decisions**, made by you, every day

> *"I'm not saying I'm the only one, but it takes a combination of events, and the timing is right now."*

THE INVESTOR'S CHOICE
Invest here

OR
Run with it

You can't close it now. The people we are looking for will only have found us because of the openness.

Dinis Cruz, 7 Nov 2026 (board/team/investors/dinis-2026-11-07-phase-two.md): "The Venn diagram of those individuals is very small."

WHAT IT ACTUALLY TAKES

"The reality is that people don't."

Not because the code is hard. Because four things have to be there, every day, at once.

1 · A TEAM

People and agents

Here: about 25 named agent roles under one human operator, and seven people at the summit.

2 · A STRUCTURE

Rules in writing

Who owns which path, passes with budgets, decisions as pages, a policy per agent.

3 · A VISION

Why it exists

Accurate information, kept current, with its provenance; pay after the value; the reader's data stays the reader's.

4 · SOMEBODY

Who is going to do it

Who reads the asks, decides, and is there tomorrow and the day after.

Dinis Cruz, 7 Nov 2026 (board/team/investors/dinis-2026-11-07-go-live-and-structure.md); CLAUDE.md; the thinking on newsroom.sgit.ai.

IF YOU HAVE THE WILL AND WANT THE HEAD START

You do not have to start from a fork.

Dinis Cruz builds the technology underneath, across his open-source startups: the place of highest leverage. **PrompTugal needs the people who execute.**

JOIN · DECK 7

Be part of the founding team

A business partner as CEO, ideally senior in the news industry; very technical journalists; strategic people, with equity.

PARTNER · DECK 3

Run it for your world

Your event, sector or region. Dual licensing: your own terms, and we maintain it for you.

INVEST · DECK 5

Back the execution

Equity in My Feeds, the publisher, for the operations: commercialisation, services, editorial.

Or help now, this summit week, on the back office: unpaid, with visibility. Web Summit Lisbon, 9 to 12 November 2026.

Dinis Cruz, 7 Nov 2026; open-source.sgit.ai/practice ("You can always commercial-license or relicense everything for a customer who wants it").

THE MAP IS IN YOUR HANDS

Fork it. Or build it with us.

Start here: **github.com/PrompTugal/PrompTugal_News_Website**, then read **CLAUDE.md**.
Everything else is linked from there.

DECK 1
The investment case

DECK 2
The industry, and the right direction

DECK 3
The platform: every feature, live

DECK 4
The Algarve's case

DECK 5
The investment and the structure

DECK 6
The technology that makes it possible

DECK 7
Join us: partners and journalists

DECK 8
Do it yourself

DECK 9
The tour: what is live