

Introducing PYRANA

Powerful, AI-powered enterprise applications you can trust

THREE KINDS OF SOFTWARE, AND WHERE PYRANA SITS



Systems of record

Software that keeps track. People do all the work.



Copilots and chat tools

Software that helps one person at a time. The result depends on that person's skill and on what they thought to ask.



Agentic applications: PYRANA

Software that carries the work. Agents do the analysis inside an application built around the process; people decide inside it.

HOW IT WORKS: EVERY APPLICATION IS CONFIGURATION ON ONE PLATFORM

Month-end close

Contract review

Order exceptions

+ the next one

SHARED ONCE, REUSED BY EVERY APPLICATION

Business knowledge

System connections

Identity and permissions

Approval rules

Audit record

PYRANA HARNESS: ONE RUNTIME FOR EVERY AGENT RUN

Durable

Saves each step. Resumes after a crash or a long approval wait.

Any model

Chosen per application. Your keys, your data, no training.

Governed

Identity, approvals, limits, and the record enforced on every call.

WHAT MAKES IT DIFFERENT



Outcome driven

Applications, not a text box. Shaped around the task, so results depend on the process, not on who is prompting.



Compounding scale

Every application accelerates the next. Foundation and business knowledge are shared, so each one builds on the last.



Risk mitigating

Guardrails outside the model. The platform enforces permissions, approvals, and limits; every action has an audit-ready history.

The hard part is already built. Each application is configuration on top: the process, the pages, and the knowledge, not the plumbing.

PYRANA

Platform architecture

- 1 cortIQ context engine
- 2 Connections
- 3 Agents
- 4 Workflows
- 5 Applications
- 6 Application state
- 7 Security and governance
- 8 Code
- 9 Models and hosting
- 10 Observability
- 11 Build
- 12 Release

YOUR SYSTEMS OF RECORD

ERP

HR

CRM

Commerce

Inventory

Documents

Risk

Tickets

Data lake

SaaS

They stay the record.
Pyrana works on top,
reads and writes back.

CONNECTIONS

- APIs
any REST or HTTP service
- Databases
SQL and ODBC sources
- Tool servers
MCP, both directions
- Files
documents and drives
- Identity
as the person, or a service
- Write-back
results where teams work

PACKAGE, RELEASE, DEPLOY

- Versioned releases
the owner activates each

BUILD

- By conversation
copilot designs and tests
- In code
SDKs and MCP
- Versioned configuration
every change a new version

COLLABORATIVE CONTEXT LAYER

cortIQ CONTEXT ENGINE

what the company knows

- Sourced
every statement cited
- Connected
related knowledge
- Versioned
history kept
- Governed
who may change it

Four kinds of knowledge, shared by every application

- Definitions
- Observations
- Rules
- Procedures

Change the knowledge, the results change. No code change.

AGENTS

- Specialists
own tools and skills
- Finish checks
done means verified

WORKFLOWS

- Hierarchies
level by level, to the top
- Durable runs
wait, resume, retry

CODE

- Tested code
named data calls, scripts
- Numbers by code
agents write words

HUMAN + AI APPLICATIONS

- Pages and copilot
one unit a business owns
- Roles and scopes
each person sees their part

APPLICATION STATE

- Plans and decisions
where no system has a home
- Extensions
add what the business needs

SECURITY AND GOVERNANCE, AROUND EVERY BLOCK

- Company sign-in
your single sign-on
- Fine-grained access
per app, entity, role
- Approval by a person
policy decides when
- Audit trail
who, what, when, why
- Tenant isolation
each client separate

OBSERVABILITY

- Every run and tool call
step by step
- Cost and lineage
of every result

MODELS AND HOSTING

- Models
Right model per agent
from approved providers

Safeguards

- Retries and failover
runs keep going
- Budgets and limits
steps and calls capped

Hosting

- Your environment
managed, or your cloud
- Per-client deployment
own environment and data

PEOPLE

- BUSINESS USERS
- BUILDERS
- OPERATORS
- OTHER ASSISTANTS

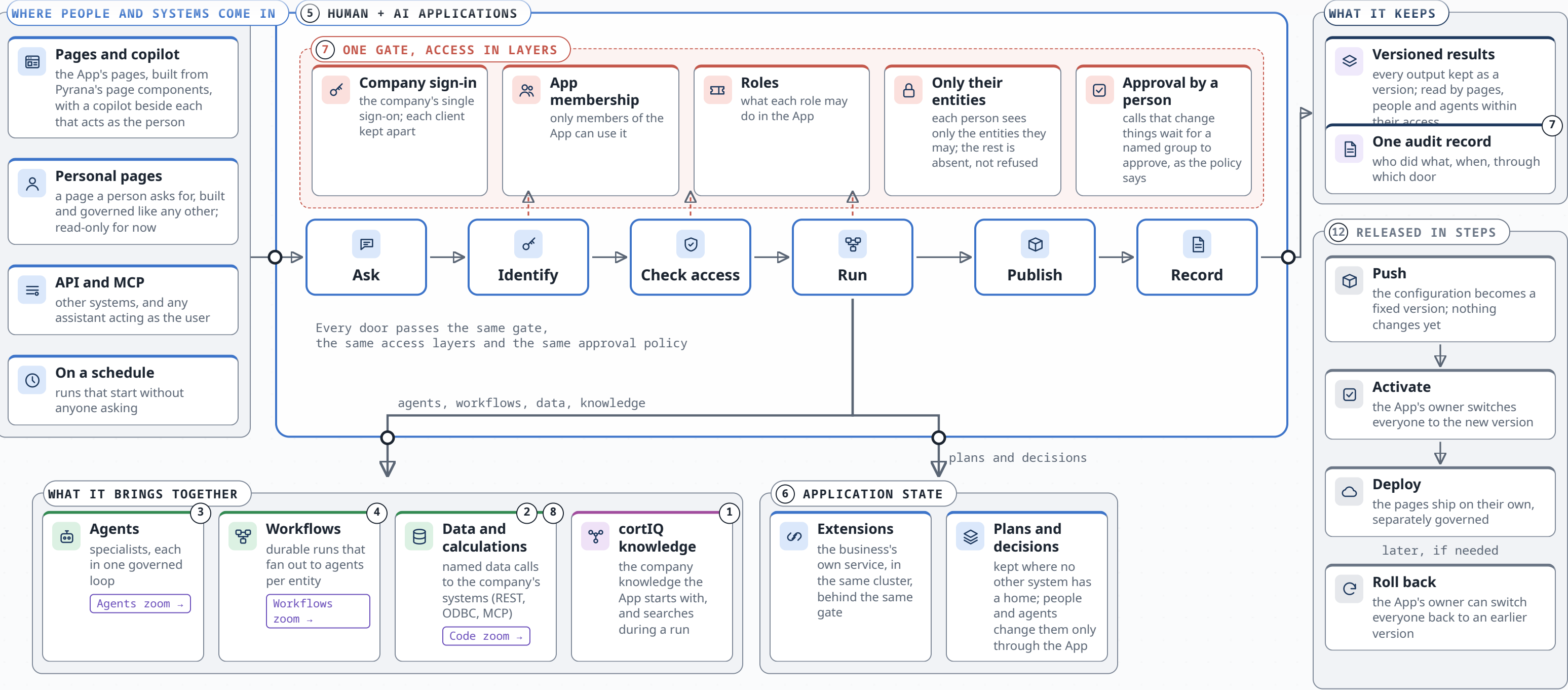
AUTOMATION

- Scheduled
daily, weekly, custom
- API-driven
other systems call
- Background
no prompt needed

Built once. Configured per application.

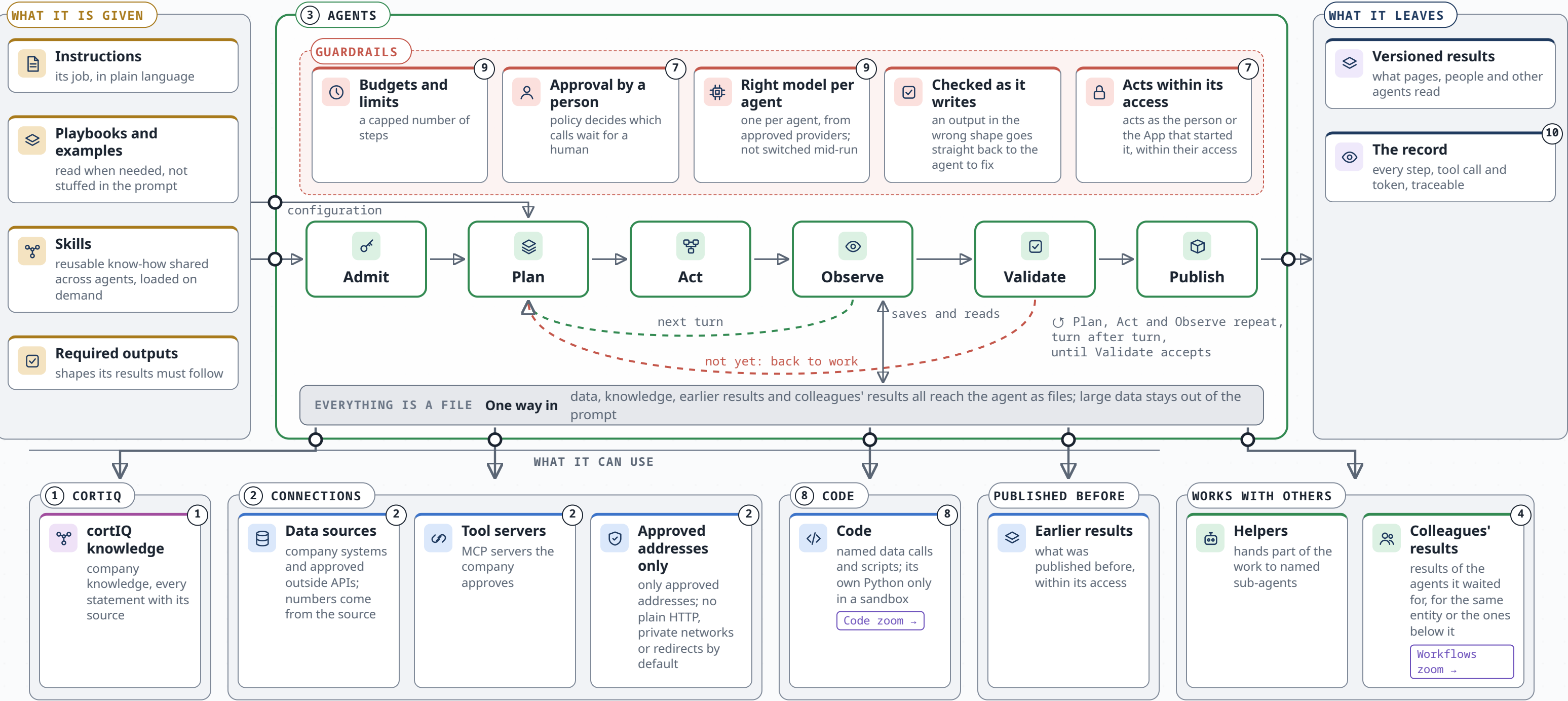
Apps: full-stack applications, not just agents

Pyrana runs whole business applications, not just agents: the pages people work in, who they are, what they may see and do, and the agents, data and results behind them. People and agents work over the same pages, data and results, through one gate.



Agents: specialists that work inside the Pyrana Harness

Every agent runs the same governed loop. Configuration decides what it knows, what it may use and when it is done.

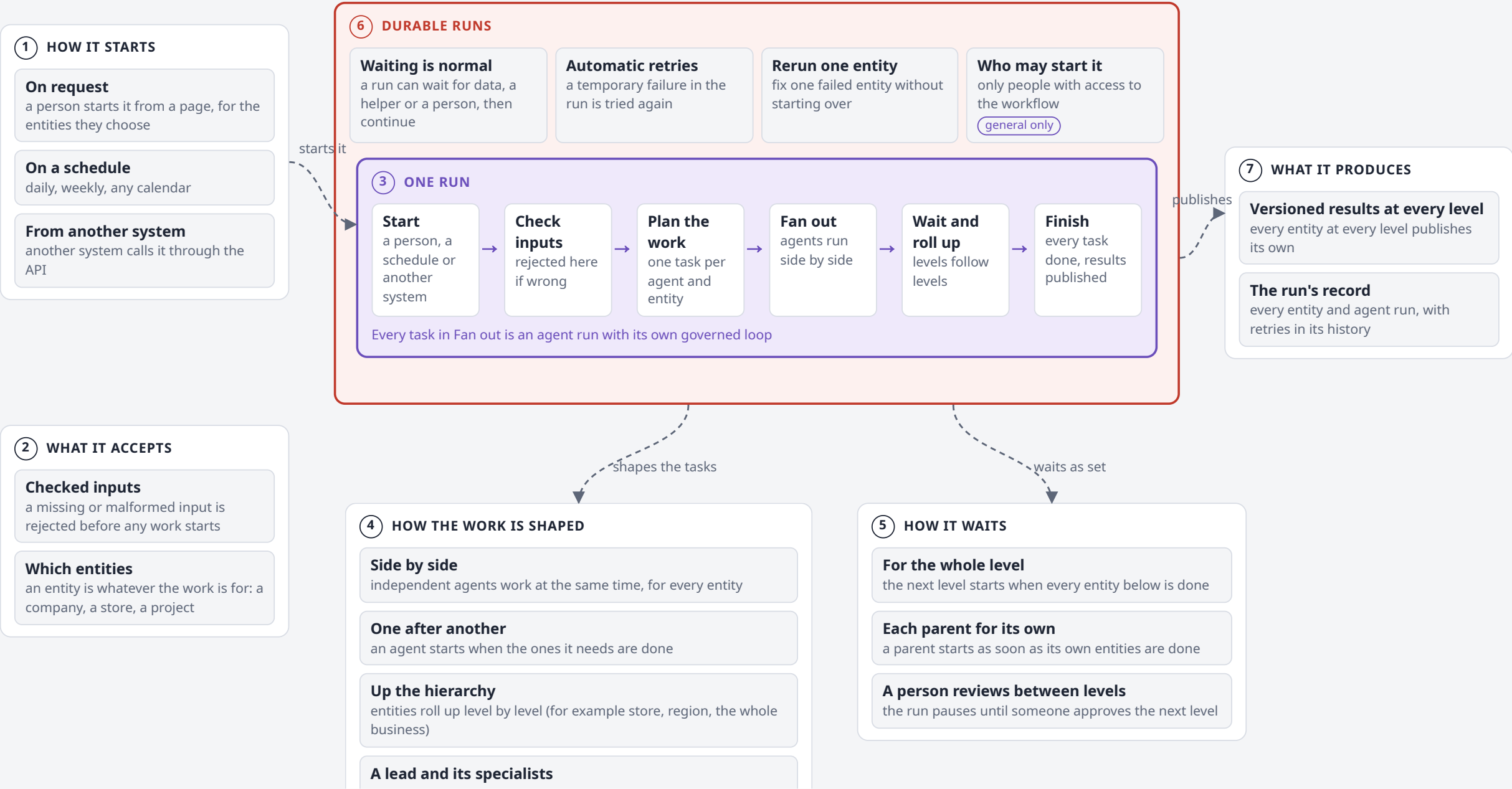


Workflows: durable runs that fan out to agents

A workflow decides who works, in what order, what waits for what, and when a person checks. The Pyrana Harness keeps the run going until it finishes.

- Unpack from the platform picture
- ◀ Back
- Walk a run
- Fold into its configuration
- Show all
- Platform

Unpack grows this view out of block 4 on the platform picture; **Walk a run** goes through the steps one click at a time and lights what each step uses; **Fold** shows it is all one configuration.



The figures people rely on are computed by tested code the agents call: named data calls and scripts. The agents write the words.

[▶ Unpack from the platform picture](#)

[◀ Back](#)

[▶ Follow a number](#)

[▶ Fold into its configuration](#)

Show all

[Platform](#)

Unpack grows this view out of block 8 on the platform picture; **Follow a number** goes through the steps one click at a time and lights what each step uses; **Fold** shows it is all one configuration.

