

AI Sprawl as a Control-Plane Problem



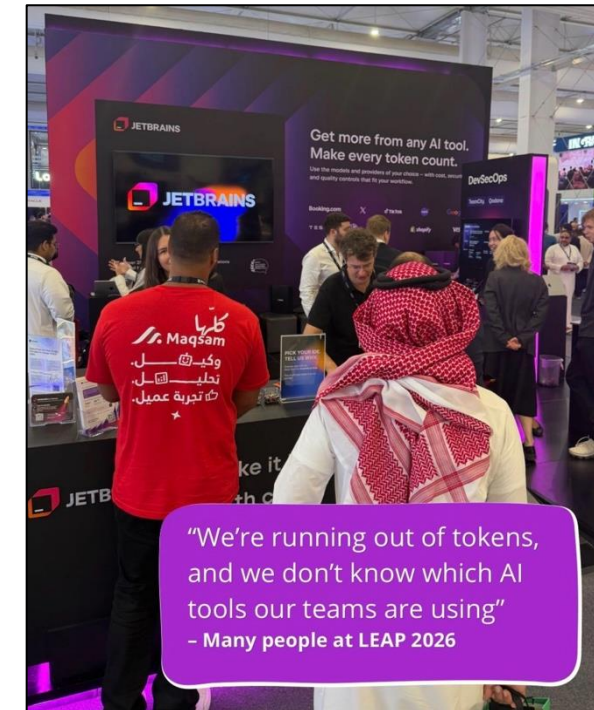
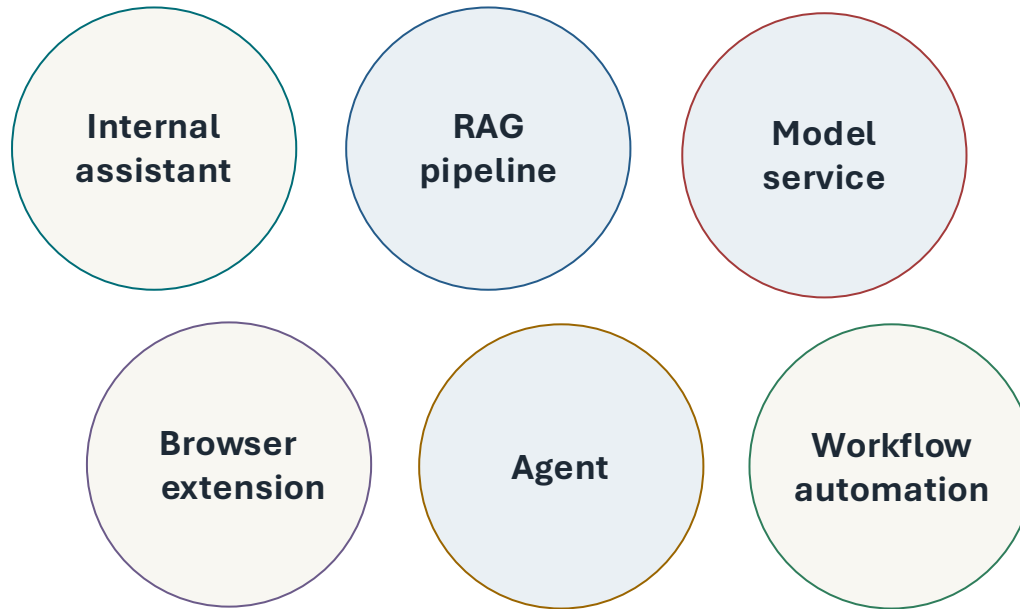
A rule-based perspective

Jacopo Massa · Patrizio Dazzi · Mariarita Pierotti

Department of Computer Science · University of Pisa

September 10th, 2026

AI capabilities outpace governance



Source: JetBrain's LinkedIn Account

many teams + many tools + many sources

Visibility

Accountability

Controls

Lifecycle

Inventories record assets, not confidence

ASSET STATE

Does this system exist?

Who owns it?

Is it approved?

What data does it use?

+

EVIDENCE ABOUT STATE

Which source supports the claim?

How fresh is it?

How confident is the observation?

Do other sources disagree?

Governance state = asset state + epistemic state

Talk structure

**1) Control-plane
framing**

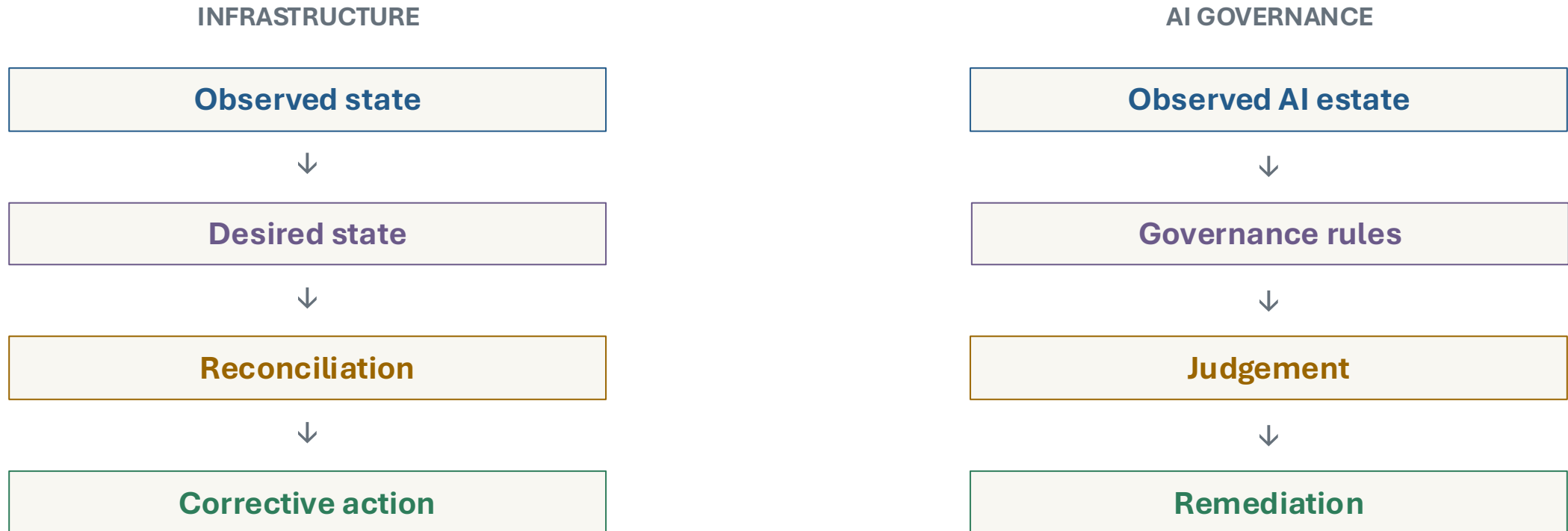
**2) Rule-based
architecture**

**3) Prototype +
running example**

**4) Limits +
production loop**

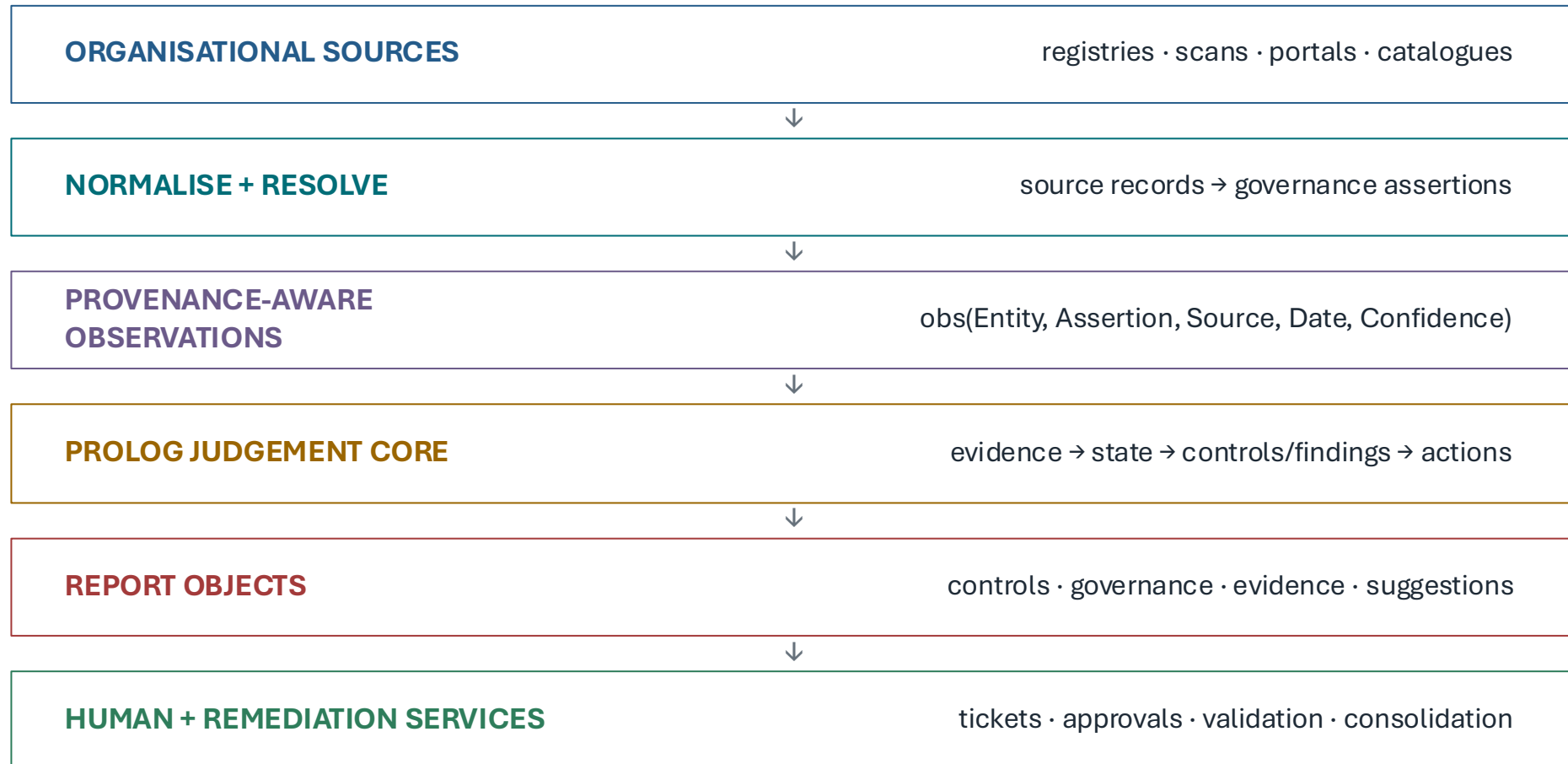
Problem → representation → judgement → action

Governance as continuous reconciliation



systems · models · agents · connectors · data · owners · approvals · safeguards · evidence

Prolog sits between discovery and remediation



feedback: remediation outcomes become new observations

Governance facts begin as claims with provenance

```
obs (  
  marketing_bot,  
  owner(marketing_team),  
  cmdb,  
  date(2026,4,18),  
  0.99  
) .
```

obs.pl

**Raw observation ≠
accepted governance fact**

Evidence policy must run first

1

ENTITY

what the claim concerns

2

ASSERTION

claimed state or relation

3

SOURCE

where evidence came from

4

TIME

when it was observed

5

CONFIDENCE

collection reliability

Evidence policy gates canonical state

```
acceptableObservation(Entity, Assertion, Source, Date, Confidence) :-  
    obs(Entity, Assertion, Source, Date, Confidence),  
    trustedSourceFor(Source, Assertion),  
    fresh(Date),  
    minConfidence(Min),  
    Confidence >= Min.
```

```
accepted(Entity, Assertion) :-  
    acceptableObservation(Entity, Assertion, _, _, _),  
    \+ conflict(Entity, Assertion).
```

main.pl:25-30, 63-66

TRUSTED?

for this assertion

FRESH?

policy hook

CONFIDENCE

≥ 0.90

CONFLICT?

suppress fact

CMDB may be trusted for owner(_), not for approval

Four rule families produce five control states



```
controlStatus(S, owner_assigned, pass) :-
```

```
    system(S), owner(S, _).
```

```
controlStatus(S, owner_assigned, unknown) :-
```

```
    system(S), \+ ownerObservation(S).
```

```
controlStatus(S, owner_assigned, fail) :-
```

```
    system(S), noOwner(S).
```

main.pl:111-120 (selected rules)

PASS

accepted positive fact

FAIL

accepted negative fact

UNKNOWN

no observation

UNRELIABLE

observed, not accepted

CONFLICTING

accepted claims disagree

unknown ≠ fail

Policy failure and evidence failure require different actions

		EVIDENCE QUALITY	
		STRONG	WEAK / ABSENT
POLICY SATISFIED	PASS	Accepted owner exists → no remediation	UNKNOWN / UNRELIABLE No owner evidence, or evidence rejected → validate evidence
	FAIL	Accepted no_owner → assign owner	INVESTIGATE Weak evidence plus risk signal → investigate before acting
POLICY VIOLATED			

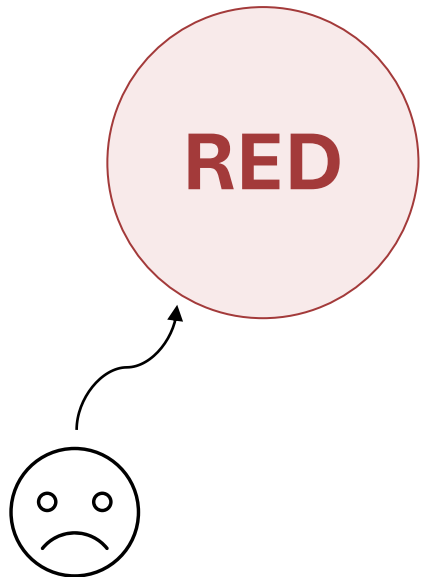
Conflicting accepted claims form a separate evidence state

A browser scan reveals a system, not its governance



DETECTED	INVENTORY
browser_extension_scan	?
CAPABILITY	APPROVAL
summarisation	?
OWNER	DATA USE
?	meeting_notes · 0.89

One report separates governance risk from evidence debt



GOVERNANCE FINDINGS

HIGH | shadow_ai_candidate
LOW | overlaps(marketing_bot,
summarisation)

EPISTEMIC FINDINGS

UNKNOWN | owner · inventory · approval
LOW CONFIDENCE | usesData(meeting_notes)

SUGGESTED ACTIONS

investigate shadow AI

request owner validation

validate inventory

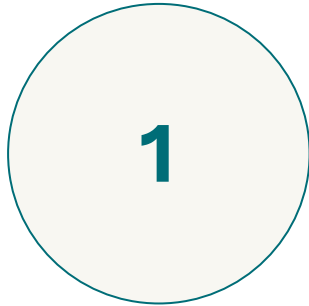
review consolidation

validate approval

review evidence quality

`report(System, OverallStatus, Controls, GovFindings, EvidFindings, Actions)`

What the prototype contributes



INSPECTABLE JUDGEMENT

Explicit declarative rules
Traceable derivations



EVIDENCE-AWARE STATE

Provenance · freshness
confidence · conflicts



ACTIONABLE OUTPUT

Controls · findings
remediation suggestions

AI-estate state + evidence quality + explicit rules + actions

Prototype boundaries

EPISTEMIC

Cannot reason about
observations never collected

Discovery and source
coverage remain upstream
problems

IDENTITY + TIME

No full entity resolution
No source-specific freshness
No supersession
No policy versioning

NORMATIVE

Humans define controls, trust,
safeguards, exceptions,
escalation, and blocking

Inspectable and contestable policy $\neq$ automatically correct policy

What a production loop still needs



Heterogeneous connectors

Identity + asset resolution

Durable observations

Policy metadata + versions

Tickets + approvals

Exceptions + tracking

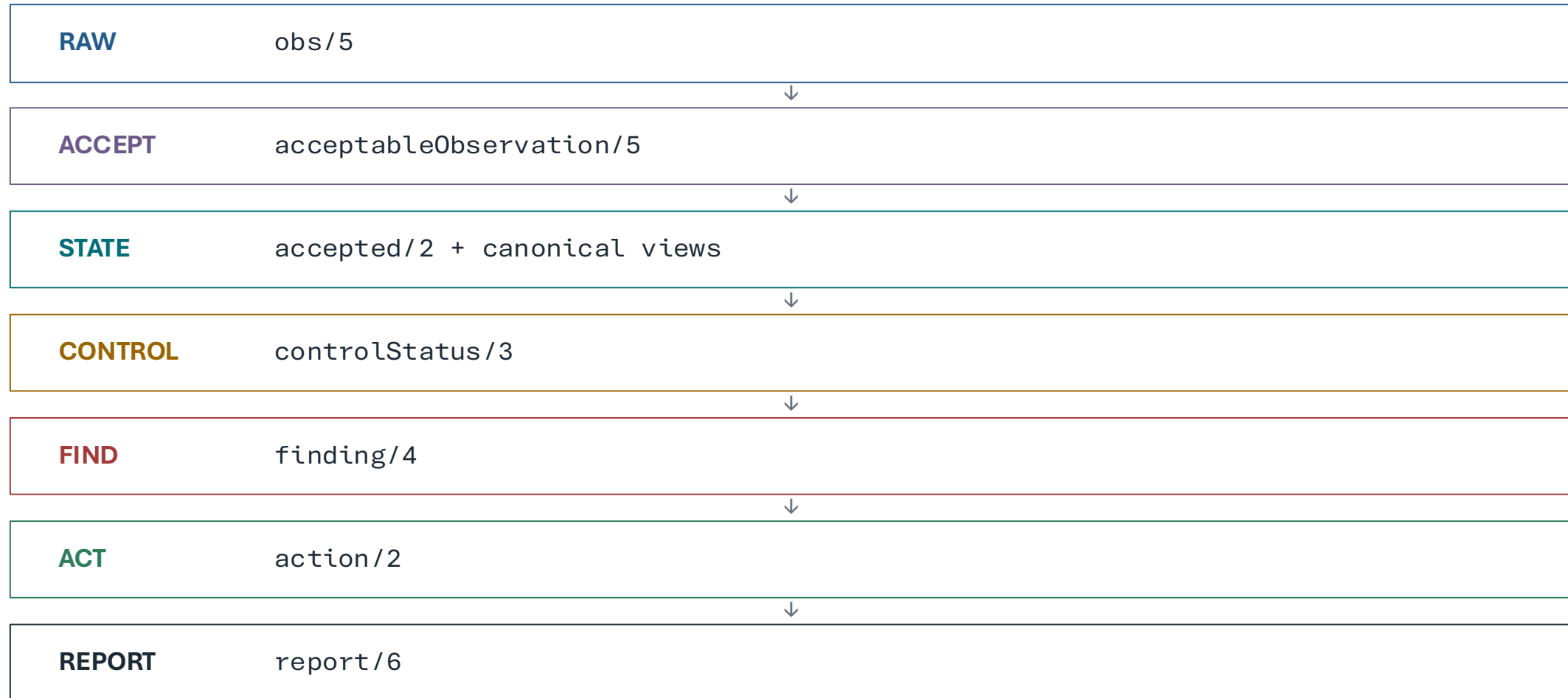
- 1** AI sprawl is state management
- 2** Evidence quality is governance state
- 3** Explicit rules can supply judgement

Questions?



policy failure
≠
lack of reliable evidence

Backup · Full rule pipeline



Backup · Evidence policy configuration

```
minConfidence(0.90).
```

```
trustedSourceFor(cmdb, owner(_)).  
trustedSourceFor(governance_portal, approved).  
trustedSourceFor(ai_registry, in_inventory).  
trustedSourceFor(browser_extension_scan, system).  
trustedSourceFor(browser_extension_scan, capability(_)).  
trustedSourceFor(browser_extension_scan, usesData(_)).
```

```
fresh(_).
```

obs.pl:36-76

CURRENT CEILING

Freshness accepts all timestamps

UPGRADE PATH

Source- and policy-specific windows

Trust applies to assertion shapes, not whole sources

Backup · Control-state semantics

STATE	MEANING	RESPONSE
PASS	Accepted positive assertion	No action
FAIL	Accepted explicit negative assertion	Governance remediation
UNKNOWN	No observation exists	Collect or validate
UNRELIABLE	Observation exists; none accepted	Repair evidence
CONFLICTING	Accepted assertions disagree	Resolve conflict
NOT_APPLICABLE	Control precondition absent	No action

Backup · report/6 assembles an inspectable object

```
report(S, OverallStatus, Controls, GovernanceFindings,  
      EpistemicFindings, Actions) :-  
  system(S),  
  sorted(control(C, St), controlStatus(S, C, St), Controls),  
  sorted(finding(T, Sev, Code), finding(S,T,Sev,Code), Findings),  
  splitFindings(Findings, GovernanceFindings, EpistemicFindings),  
  overallStatus(Findings, OverallStatus),  
  sorted(Action, action(S, Action), Actions).
```

main.pl:3-9



Backup · Prototype estate and observed outcome

marketing_bot

owner ✓ inventory ✓ approved ✓
summarisation · PII without guardrail

meeting_summariser

owner ? inventory ? approved ?
summarisation · low-confidence data use

sales_agent

owner ✓ inventory ✓ approved ✓
lead scoring · PII filtering

hr_helper

owner ✓ inventory ✓ approved ✓
document Q&A · PII without guardrail