

Why AI Needs a Registry

Owned workflows, trusted agents, and the discovery layer sovereign AI will need.

ENTRY № 05 · 23 June 2026 · 11 min read

In a previous essay, [The Return of Owned Software](#), I argued that AI will not kill software. It will change what enterprises rent, what they own, and where the value goes. The simple version was this: generic software loses its pricing power, and strategic workflows become worth owning again.

That argument has a second half.

If AI makes software easier to create, institutions will create more of it. They will generate internal tools, local agents, workflow-specific copilots and packaged skills that encode their own rules, processes and institutional memory. This is powerful. It is also dangerous. Once every organisation can create more software, the question changes again. Not only: which workflows should we own? But: how will anyone know which workflows, agents and skills exist, who owns them, what they are allowed to do, and whether they can be trusted?

That is why AI needs a registry. Not a filing cabinet. A trust map.

THE CENTRAL ARGUMENT

AI does not only need more powerful models. It needs trusted ways to discover which models, agents and skills are locally relevant, properly governed and appropriate to evaluate, integrate or invoke.

The throughline

A single line of thought, across three essays



Three essays, one argument

From accountability, to ownership, to discovery

The new shadow IT will not look like the old one

The old shadow IT problem was unsanctioned SaaS: teams signing up for tools outside procurement, moving data into platforms the organisation did not fully understand, and creating parallel systems that IT discovered only after they had become operationally important.

The new shadow IT problem may be stranger. It will be generated workflows, local agents, scripts, copilots and AI skills, created quickly because the cost of creation has fallen. The risk is not that institutions build too little. It is that they build too much, with no shared memory of what exists.

The answer cannot be to forbid building. That would only push the work underground, which is the worst of both worlds. The answer is to make creation governable. Before an organisation can secure, audit, integrate or retire a capability, it must first be able to name it and find it.

A registry is one part of that discipline. It does not replace security, identity, access control, audit or compliance. It answers an earlier question: what exists, who owns it, what is it for, and how should it be discovered?

AI systems are moving from tools to actors

For most of the history of software, applications waited for people. We opened them. We clicked buttons. We searched, submitted, approved and moved data from one system to another.

AI changes that. Increasingly, software will not only respond. It will interpret, recommend, call tools, retrieve context, coordinate workflows and act on our behalf. A workflow that once waited for a human can become an agent. A document that once

required a specialist can be routed through a skill. A local process that once lived only in an institution's memory can become a callable capability.

That shift changes the trust question. The old question was: can this software work? The new question is: should this AI system be allowed to act? With which tools? Under whose authority? In which jurisdiction? Against which data? With what level of human oversight?

In the API era, software exposed endpoints. In the agent era, software exposes capabilities. Once capabilities become discoverable and callable by AI systems, discovery itself becomes infrastructure.

Discovery, in this sense, does not mean search alone. It means identity, metadata, ownership, governance signals and machine-readable ways to resolve a capability, so that a person or an agent can decide whether to use it, not merely find that it exists.

The missing layer is discovery

Many of the most useful AI resources will not live inside one global platform. They will be local. A tax skill that understands a country's filing rules. A public-service navigator that knows the right ministry. A school assistant aligned to a national curriculum. A company-registration agent that understands domestic procedures. A language model adapted to local language and culture.

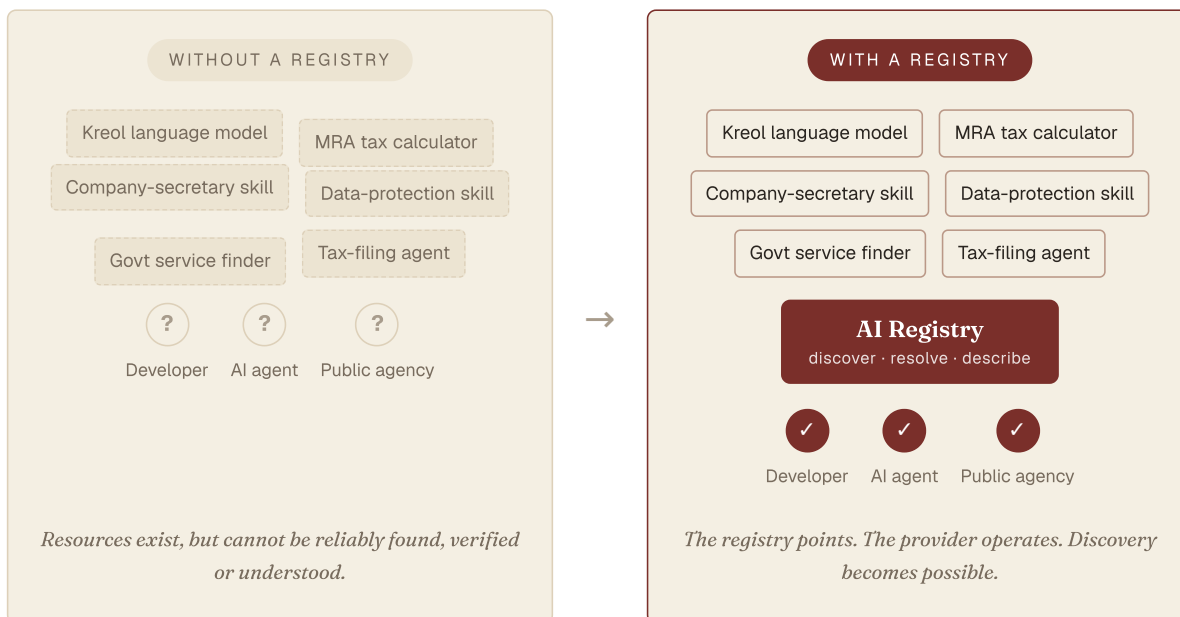
These resources may be small, but they matter. For a teacher, a small business, a public servant or a citizen, the most useful AI resource is often not the largest global model. It is the one that knows the local process.

Without a discovery layer, these capabilities remain scattered. Developers do not know they exist, so they default to generic global tools. Public agencies publish capabilities, but partners cannot reliably plug in. AI systems cannot programmatically find local resources. Governance metadata is scattered or unavailable. Each new resource starts from scratch instead of joining a connected ecosystem.

If these capabilities cannot be found, verified or understood, they do not form an ecosystem. They form clutter.

The discovery gap

The same resources, before and after a sovereign discovery layer



The internet scaled because it had naming, discovery and resolution layers. Sovereign AI needs a similar pattern: not a centralised platform that owns everything, but a common way to list, describe, identify and resolve locally relevant AI resources.

A LEDGER PRINCIPLE

The scarce layer is no longer only the ability to create software. It is the ability to make capabilities discoverable, governable and reusable.

Sovereign AI means local relevance

Sovereign AI is sometimes discussed as if it were mainly a question of where the servers sit. Location matters. Data centres, GPUs, connectivity, cloud controls and regulated hosting all matter. But sovereignty is not only geography. It is relevance.

A sovereign AI resource is useful because it encodes something local: local law, local data, local systems, or local language and culture. It knows the process, the obligation, the institution, the terminology, the context. It does not merely produce fluent answers. It understands the environment in which those answers will be used.

This is why a registry matters at national level. The problem is not simply to list AI tools. It is to identify which AI resources are locally relevant enough to be part of a country's digital public infrastructure. Quality matters more than quantity. A registry of fifty credible, well-described local resources is more useful than a directory of a thousand generic tools.

This is not an argument for cutting countries off from global AI platforms. The opposite is true. The more global AI becomes, the more important it is for local capabilities to be visible to it. Sovereignty here is not a wall. It is a way to make local law, language, institutions and obligations legible to the global AI ecosystem.

What an AI Registry is, and what it refuses to be

An AI Registry is a sovereign discovery layer. It tells people, developers and AI systems what local AI resources exist, who provides them, why they matter, and where to find them.

The most important design principle is restraint. A registry should not try to become an AI hosting platform, a gateway, a billing system, a marketplace, an access-control layer or a legal certification authority. Each of those functions may be useful. But once the registry absorbs them, it becomes a different kind of platform, with different incentives, different liabilities and different trust problems.

A provider remains responsible for its resource: endpoints, documentation, terms of access, version control, quality and runtime behaviour. A hosting environment runs the workload, whether on cloud, sovereign infrastructure, on-premise systems or specialised compute. The registry stays on the discovery path, not the runtime path. That is what allows it to remain neutral: it does not need to own a resource in order to make it discoverable.

The registry-only boundary

Three layers, three parties. The registry is only the first one

The registry points

OPERATED BY THE REGISTRY OPERATOR

Lists, describes and resolves locally relevant resources. Issues stable identifiers and exposes governance signals. It does not host, gate or call.

Public portal

Discovery API

Stable AIR-IDs

Sovereignty review

The provider operates

OPERATED BY THE RESOURCE PROVIDER

Owns the resource itself: endpoints, documentation, terms of access, version control, quality and runtime behaviour. The provider remains responsible.

Endpoints (REST · MCP · A2A)

Access terms

Versioning

Provider authority

The hosting environment secures

OPERATED BY THE HOSTING PROVIDER

Runs the actual workload (cloud, sovereign infrastructure, on-premise or specialised compute) and handles runtime identity and access.

Compute & storage

Runtime identity

Gateways & access

Operational security

User searches the registry → registry returns metadata and an AIR-ID → user goes directly to the provider → provider serves from its own hosting. *The registry is never on the runtime path.*

A LEDGER PRINCIPLE

The registry points. The provider operates. The hosting environment secures. Those boundaries are what make the model durable.

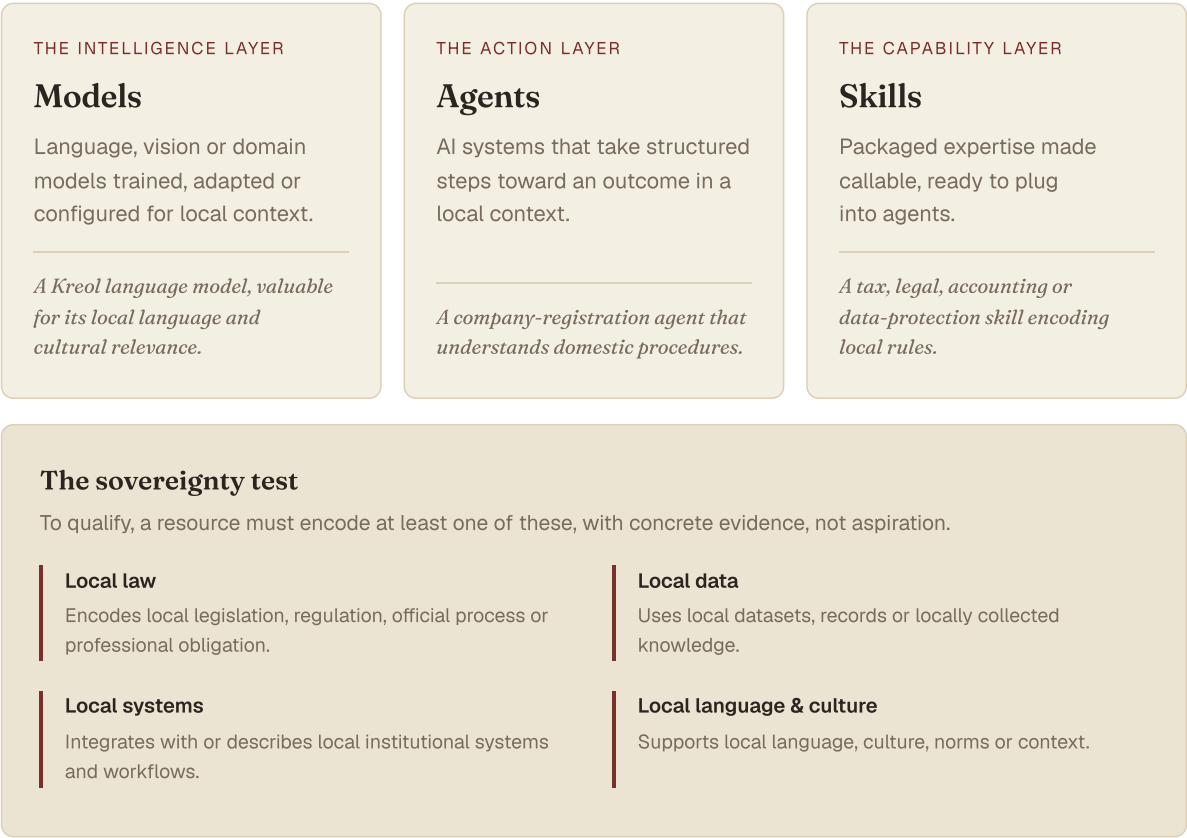
Models, agents and skills

For the purpose of a public registry, three categories are enough to begin with: models, agents and skills. Models provide the intelligence. Agents take structured action. Skills package specific capabilities that agents can call.

The point is not taxonomy for its own sake. It is composability. Models, agents and skills can be discovered, combined and governed only if they are named and described in a consistent way, and only if each one earns its place by encoding something genuinely local.

Models, agents and skills

Three public categories, each tested for genuine local relevance



Why this matters for agentic AI

Agentic AI makes the registry problem more urgent, because agents need to discover capabilities at the moment of work. A human can search a website, read a description, compare options and decide which tool to use. An AI system needs structured metadata, stable identifiers and machine-readable signals.

This is where protocols such as MCP become important. They point toward a world where AI systems can connect to tools, context and capabilities more easily. But easier connection raises the stakes of discovery. If an agent can call tools, it must also know which tools are trusted, which are local, which are official, which are experimental and which are out of scope.

The registry is not the runtime. It is the layer that helps an agent know what exists before it acts.

A registry is a map, not the road. A catalogue, not the warehouse. An address book, not the conversation itself. As AI moves from chat to action, that address book becomes strategic.

Listing is not endorsement

A registry must not accidentally become a certifier. This is one of the hardest design choices, because users will naturally read presence in a registry as a trust signal. That signal must be precise.

Listing should mean: this resource exists and has been described. Provider verification should mean: the provider is who they claim to be. Sovereignty review should mean: the resource has shown concrete local relevance. Official-resource status should mean: an authorised body has explicitly endorsed or authorised the resource.

Those signals should remain separate. A self-declared resource is not the same as a sovereignty-reviewed resource. A verified provider is not the same as an official public resource. A locally relevant capability is not automatically safe for every use case. Keeping these signals apart prevents the most common failure of public registries: the registry quietly becomes a de-facto certifier, attracting liability that was never intended and that providers never agreed to.

A registry should expose trust signals, not pretend to settle all questions of trust. That is governance without overreach. It is the same discipline I argued for in [Accountable Intelligence](#): when work is produced by a mix of people and software, accountability has to be designed rather than assumed, not smuggled in by a label.

Why telcos and trusted DPI enablers have a role

By DPI enablers, I mean institutions trusted to operate digital public infrastructure across an ecosystem, not only for themselves. If telcos are becoming digital infrastructure companies, their role cannot end at connectivity, cloud and compute. They can also help operate the trust and discovery layers that make AI adoption possible.

Telcos understand interconnection. They operate regulated infrastructure. They maintain long-lived platforms. They serve consumers, enterprises and governments. They are used to resilience, identity, availability, security and national-scale service obligations. Those capabilities are exactly what a registry layer demands.

But this is not a claim that only telcos should operate registries. The natural operator in any country may be a government digital agency, a national identity authority, a sovereign cloud operator, a public-interest technology foundation, a regional standards body, a national CERT or another trusted digital public infrastructure enabler. The important point is that the operator must be trusted beyond itself. A registry is not valuable because one organisation controls it. It is valuable because many parties are willing to use it as a neutral discovery layer.

Mauritius as a working reference implementation

This is not only a theory. In Mauritius, we have started building our first version.

The Mauritius AI Registry at airegistry.mu is our first working reference implementation. It points to locally relevant AI resources and gives each listing a stable AIR-ID, provider information and governance signals. The open-source version at airegistry.dev makes the pattern deployable by other countries, cities, telcos or public-infrastructure operators. The code is public on GitHub under [MauritiusTelecom/ai-registry](https://github.com/MauritiusTelecom/ai-registry).

The reason to open-source the work is simple: a registry layer only becomes useful if it can be trusted beyond the institution that first built it. This should not be a proprietary trophy. It should be a working contribution.

Mauritius is a small country. That is part of the point. Small jurisdictions can coordinate nationally, test quickly, learn from real users and build first versions. They cannot define the future alone. But they can make it less abstract.

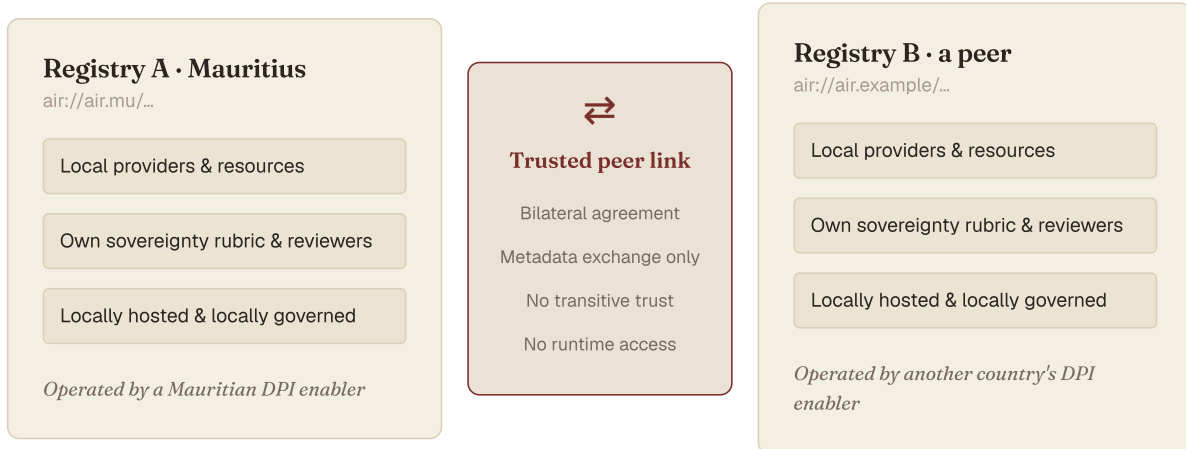
From one registry to a network of registries

The long-term vision is not one global AI registry. That would be the wrong architecture and the wrong politics. The better vision is a network of sovereign registries.

Each country, city or jurisdiction should be able to maintain its own sovereignty rubric, provider verification process, status labels and audit trail. Federation should be bilateral, explicit and metadata-only. It should not merge governance, transfer authority or create automatic runtime trust.

A network of sovereign registries

Not one global registry, but many local ones that trust each other's metadata



Each side keeps its own sovereignty rubric, status labels and audit trail. Federation does not transfer authority.

When a person enters a country, they may need to show who they are, why they are there, and what they are allowed to do. AI agents will need a comparable way to declare identity, capability, authority and local permission. The registry is not the border. It is the trusted layer that helps make discovery, permission and accountability possible.

An open invitation

The AI Registry pattern should be intentionally small, intentionally open and intentionally sovereign. Its value comes from restraint, and from the willingness of others to deploy and govern their own.

For governments, it offers a practical instrument for national AI strategy: visible local capabilities, named providers and transparent review. For industry, it offers a credible place to publish locally relevant AI resources. For developers and AI systems, it offers a machine-readable place to discover sovereign capabilities. For citizens and businesses, it creates the possibility that AI systems can find and use locally accurate tools instead of defaulting to generic answers.

There are four ways to engage. Adopt the pattern with the open-source release at airegistry.dev. Contribute to the codebase and schema on the [GitHub repo](#). Partner on governance and federation. Or simply observe the live Mauritius reference implementation at airegistry.mu and decide whether the pattern fits your context.

AI will create more software, not less: more workflows, more agents, more skills, more local capabilities. The strategic question is no longer only what we build. It is how we make what we build accountable, discoverable and fit to integrate.

A decade ago, the hard part was building intelligence. Soon, access to intelligence may be the easier part. The hard part will be knowing what we have built, who stands behind it, and whether it can be trusted to act.

The model is the engine. The registry is the map.

The next decade will not be shaped only by whoever builds the most intelligence. It will be shaped by whoever can find it, trust it, and stand behind what it does.

That is why AI needs a registry.

WRITTEN BY

Veemal Gungadin

WITH ASSISTANCE FROM **Claude Opus 4.8** · Anthropic , **GPT-5.5** · OpenAI

Drafted and revised by the author. AI tools were used to sharpen structure and prose; conclusions are the author's own.

Published in *The Goodwill Ledger* at www.goodwill-led.com/essays/why-ai-needs-a-registry/