What's the problem?

Right now, if you want AI agents to work together — say one that reads your email, one that manages your calendar, one that handles your finances — they all have to go through a company's cloud server. Your email agent talks to OpenAI. Your calendar agent talks to OpenAI. Your finance agent talks to OpenAI. Everything routes through one company, on their servers, on their terms, billed to your credit card.

If that company goes down, all your agents stop. If that company raises prices, you pay more. If that company decides your agents are doing something it doesn't like, it pulls the plug.

This is exactly what phone networks looked like 30 years ago. One company controlled everything. If you wanted to make a call, you used their equipment, their towers, their cables. Then the industry broke it open. They created a system called O-RAN — Open Radio Access Network — where the software that manages the towers is separate from the towers themselves, and anyone can write the management software. Competition. Interoperability. No lock-in.

We did the same thing for AI agents.

How it works (the simple version)

There are two main pieces, just like in a cell network:

The Manager (SMO). This is the part that knows which agents exist, whether they're healthy, and what they're supposed to be doing. When a new agent comes online, it registers with the manager. Every 30 seconds, it sends a heartbeat — "I'm still here." If the heartbeats stop, the manager knows something is wrong.

The manager also has rules. You set them: "If an agent starts using too much memory, restart it." "If an agent starts behaving strangely, isolate it." "If traffic is too high, spin up more agents." You set the rules once. The

system follows them forever.

The Brain (RIC). This is the part that watches everything and looks for patterns. It uses machine learning — the same kind of pattern recognition that spots a fraudulent credit card transaction or a spam email. It watches the agents' metrics: how fast are they responding? How many errors are they making? How much CPU are they using? Is anything weird?

When it spots something weird, it tells the manager. The manager checks the rules. If there's a match, it acts. No human needed for the day-to-day. Humans set the policy. The system executes it.

What does "something weird" look like?

The brain watches 24 different measurements about each agent. Not just "is it on or off" — it tracks things like:

- How long is it taking to respond? (latency)
- How many requests is it dropping? (packet loss)
- How much of its capacity is it using? (bandwidth utilization)
- Is the response time getting worse over time? (trend)
- Is today a weekday or weekend? (context — patterns differ)

It builds a picture of what "normal" looks like for each agent. When something falls outside normal — a sudden spike in errors, a gradual creep in response time, an unusual pattern of requests — it flags it.

It gets this right 92% of the time. In about 15 milliseconds. Faster than you can blink.

Two examples

The anomaly detector. One of the AI agents in the network starts responding slowly. Packet loss goes up. Latency spikes. The anomaly detector catches it, scores the severity, and publishes an alert. The policy engine sees the alert, matches it against the rule "if anomaly score > 0.8, quarantine the agent," and takes the agent offline before it causes problems for everything else connected to it.

Nobody had to wake up at 3am. Nobody had to look at a dashboard. The system noticed, evaluated, and acted.

The traffic optimizer. The system notices that Route A is handling 70% of the traffic while Route B is handling 40%. The optimizer runs the numbers: if it shifts some traffic from A to B, latency drops by 20%. It publishes a recommendation. If auto-apply is turned on, the system makes the change. If not, it waits for a human to approve it.

Either way, the analysis happened without a human asking for it.

Why not just use the cloud?

You can. The same way you can use one phone company for everything. But:

- When AWS went down in December 2021, it took down Alexa, Ring doorbells, robot vacuums, and Disney+. One failure point, millions of casualties.
- Cloud AI orchestration costs money. Every API call is a billing event. A busy agent network can run up thousands of dollars in monthly fees.
- You don't control the cloud. The cloud controls you. Pricing changes, terms of service changes, capability changes — all at the provider's discretion.

This system runs on your own hardware. A laptop. A Raspberry Pi cluster. A decommissioned office server. It compiles to a single program. It uses about 512MB of memory. It costs nothing per month beyond electricity.

How does this connect to the phones?

(Applebee & Combe, 2026, "*The Invisible Network*") showed that phones can talk to each other over Bluetooth without any cell towers or internet. (Applebee & Combe, 2026, "*Sovereign AI Infrastructure*") showed that AI agents can run without being tethered to a corporation. This paper shows the missing piece: how those untethered AI agents coordinate.

Think of it like this:

- **(Applebee & Combe, 2026, "*The Invisible Network*")**: The walkie-talkies (how devices communicate)
- **(Applebee & Combe, 2026, "*Sovereign AI Infrastructure*")**: The people using the walkie-talkies (sovereign agents)
- **(Applebee & Combe, 2026, "*From Cellular to Neural*")/29**: The switchboard (how the communication is managed)

Except there's no physical switchboard. The management is software. It runs wherever the agents run. If one management node goes down, others keep running. There's no center to attack.

The "but you need the cloud" argument

Every AI agent framework today — LangChain, AutoGen, CrewAI — assumes you'll be sending your data to someone else's computer for processing. Every one of them requires API keys to a cloud provider. Every one of them stops working if that cloud provider decides to revoke your access.

The cell phone industry used to work like that too. One company owned everything. Then O-RAN happened, and now the software that manages the network can run on anyone's hardware, from any vendor, using open interfaces.

We did the same thing for AI agents. Open interfaces. Local hardware. No API keys to someone else's kingdom.

What it looks like in practice

You have a network of AI agents running on machines you control. Maybe they're spread across several computers in your house. Maybe they're running on phones connected over Bluetooth mesh ((Applebee & Combe, 2026, "*The Invisible Network*")). Maybe they're on a cluster of cheap mini-PCs.

The SMO/RIC system runs alongside them. It watches. It learns what normal looks like. When things go wrong, it reacts — in 15 milliseconds. When things could be better, it suggests changes. When a new agent joins, it gets registered and monitored automatically.

You set the rules: how aggressive should auto-scaling be? What anomaly score triggers quarantine? Should route optimizations be applied automatically or require approval?

Then you walk away. The system runs itself. Like a cell network. Except you own it.

The one-sentence version

Your AI agents coordinate the same way cell towers do — automatically, intelligently, and without anyone in a corporate office deciding whether they're allowed to.