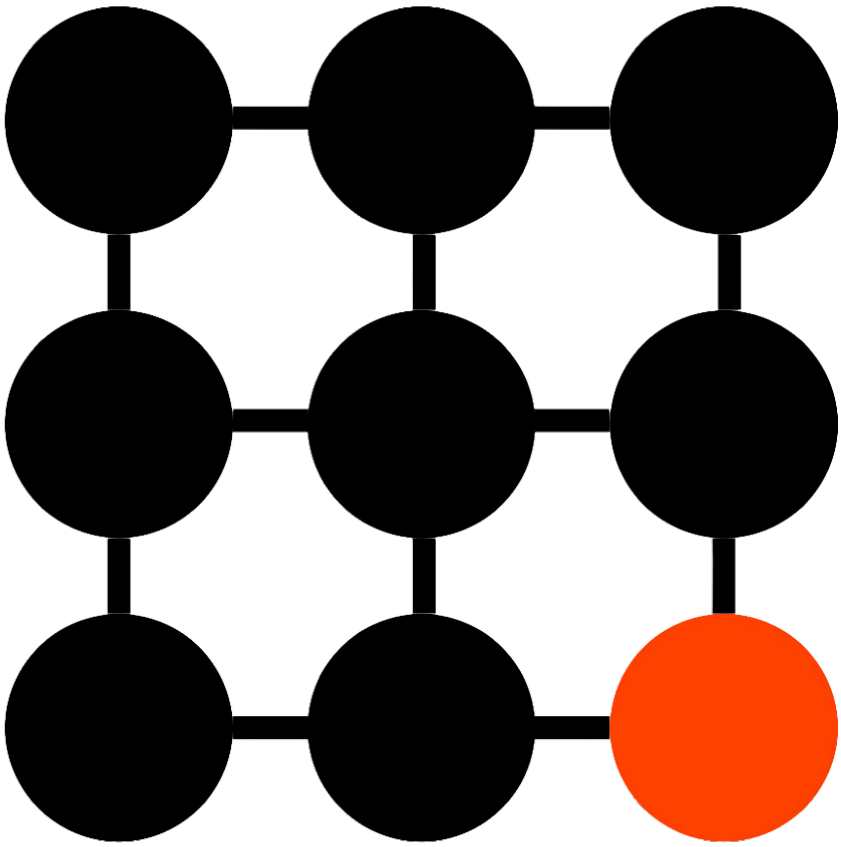
AI Agents: The New Economics of Automation



Barnacle_Labs

An executive guide to AI agents: what they are, where they create value, and how to get started.

Our team has over a decade of experience delivering AI solutions, led by the former European CTO for IBM Watson. We believe in AI for breakthroughs, not buzzwords, so this document distills our practical experience of building agents in production.

DECEMBER 2025

GET YOUR FREE WORKBOOK HERE

Table of contents

The Opportunity 2

What is an agent 4

Workflows vs Agents..... 5

How to Identify Agent Opportunities 8

Getting Started 9

Large or Small Language Models..... 10

Agent Frameworks 11

Designing Tools for Agents 12

Build Evals to Measure Performance 13

Scaling Considerations..... 15

Key Takeaways 16

How Barnacle Labs can help 17

The Opportunity

The Uneconomical Gap

Earlier waves of automation relied on observing existing work and then replicating it in software. AI agents are different, they represent a fundamental shift in the economics of work and can automate:

Work that resisted traditional automation:

Tasks too open-ended or requiring too much judgement for conventional automation software.

Work we never bothered with:

Every business has a backlog of "nice to have" tasks: reviewing every supplier contract for risk, generating personalised outreach for 50,000 prospects, monitoring 200 journals daily. This work is valuable but tedious, high-volume, and requires judgement, so we simply never did it, because the technology couldn't handle it and we couldn't justify the cost of humans doing it.

Agents fill this gap. They handle the "messy middle": work that sits between the low-complexity tasks suited to traditional automation software and the high-value strategic work that still requires human judgement.

Studies suggest the economic potential of automating this work is significant: MIT's Iceberg Index estimates that AI automation spanning administrative, financial, and professional services could represent 11.7% of the US labour market, or approximately $1.2 trillion of economic activity. McKinsey estimate the global opportunity at $2.9 trillion.

The hard part is imagination, we're not used to thinking of this work as automatable, and it's even harder to spot valuable work we've never considered viable to do.

> **The IT department of every company is going to be the HR department of AI agents in the future.**

Jensen Huang,
NVIDIA CEO, January 2025

What is an Agent?

Outside AI:

In philosophy, agency means the capacity to act intentionally – to pursue goals and influence the world. In law, an agent is someone who acts on behalf of another: an estate agent, an insurance agent, a literary agent.

In AI:

Agents are software that represents a user, taking actions on their behalf with a degree of autonomy in how those actions are carried out.

To make this concrete:

An AI agent tasked with researching a competitor might search the web, read the results, decide it needs financial data, query an API, realise the data is incomplete, try a different source, and compile a summary, all without being told how to do this. The agent reasons about how to go about meeting the goal it's been given and creates a plan it then executes on, adapting as more information becomes available.

But it's easy for this excitement to outrun reality. At CES 2025, Jensen Huang of NVIDIA pitched a future where companies manage fleets of AI workers, complete with cartoon faces:

"The IT department of every company is going to be the HR department of AI agents in the future." — Jensen Huang, NVIDIA CEO, January 2025

It's a memorable quote, but a dangerous operating assumption today. If your question is "Can an agent replace my finance department?"– the honest answer is: not yet.

Reality Check:

Today, agents are effective at automating specific, clearly defined tasks, not entire roles or departments.

Workflows vs Agents

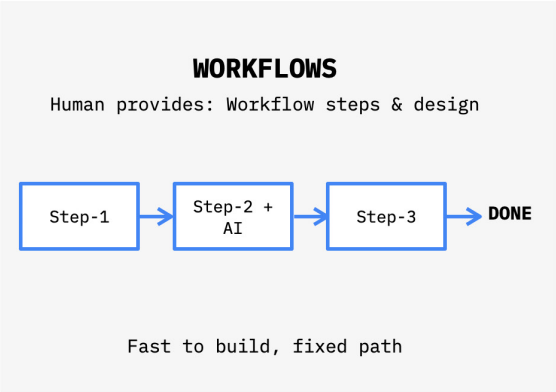
Distinguishing two concepts that often get conflated

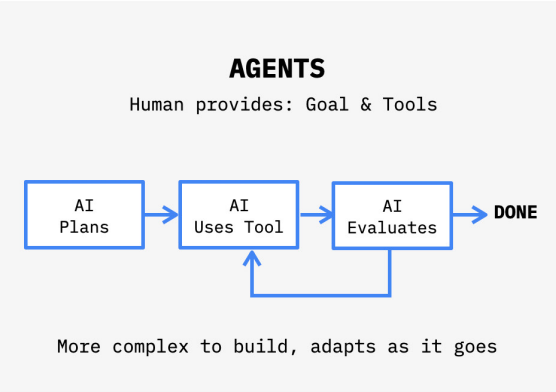
Workflows:

Workflows are step-by-step processes designed in advance by humans. Add AI to one of the steps and some people start calling it an agent. It isn't, it's still running a series of pre-defined steps. Useful for many problems, but limited: these systems can't reason about open-ended problems, construct their own plans, or adapt when something unexpected happens.

Agents:

Agents don't follow pre-defined steps. They're given a goal and access to tools—file systems, APIs, databases, web search and work out how to achieve the goal themselves. Think of tools as the agent's hands: specific capabilities it can use to gather information or take action. The agent decides which to use and when. You define which tools your agent has access to. It’s likely that you will use some generic tools, for example web search. But for enterprises, you’ll also need to build your own tools to give your agents access to your CRM, to query core system records, etc.





How Agents Actually Work

A Concrete Example

1. The Goal: "Find the email for the founder of Barnacle Labs, verify it, and save it to the CRM."
2. The Tools: Web Browser, LinkedIn Search, Email Verifier API, CRM API.
3. How the agent works (see The Agentic Loop):
    - Step 1: Agent plans: "I need to find the name first." → Calls Browser Tool.
    - Step 2: Agent reads results: "The founder's name is [Name]." → Calls Email Verifier Tool.
    - Step 3: Agent verifies: "Email is valid." → Calls CRM Tool to save.
    - Step 4: Task Complete.

If Step 2 fails (e.g., email not found), a standard script crashes. An agent reasons: "I will try a different search query," and loops back.

A note on agent protocols

You might hear discussion about standards that make it easier to connect agents to tools, data sources, and each other, think of them as USB for agents. The one that matters now is MCP (Model Context Protocol), which simplifies connecting tools to agents and is supported by nearly all vendors. Others like Google's A2A and IBM's ACP focus on agent-to-agent communication. If, like most organisations, you only have a small number of agents doing different things that have no need to communicate with each other, this isn't relevant yet.

How real is this?

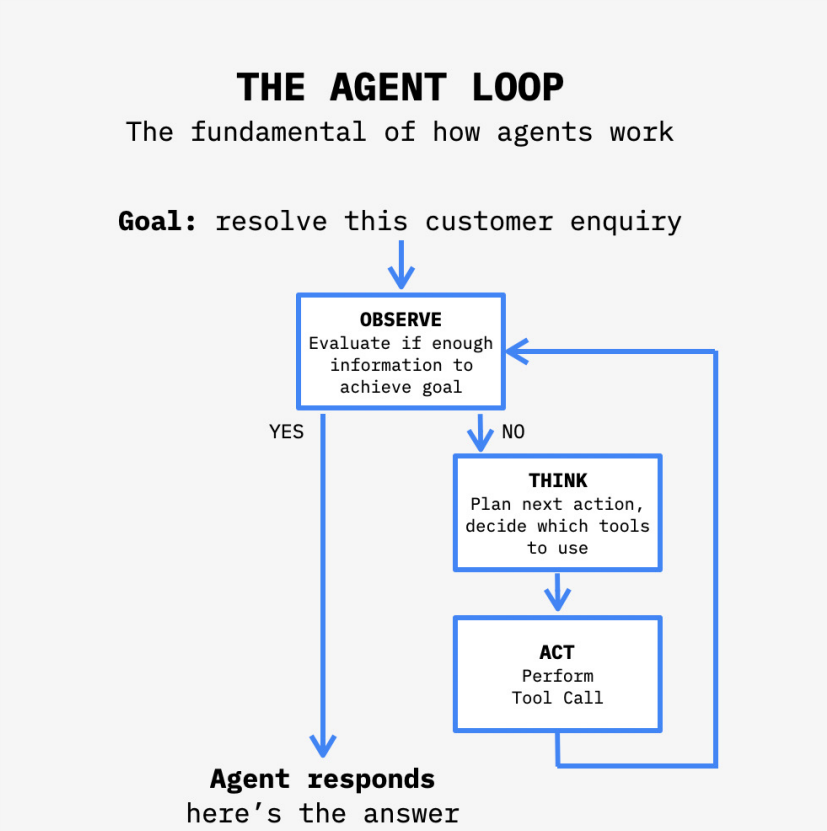
The best proof that agents are enterprise-ready is AI coding. Products like Anthropic's Claude Code and OpenAI's Codex are already used extensively by software developers. The underlying technology is the same for general-purpose agents – the challenge now isn't capability, it's imagination: translating what's working in software engineering to domains like financial services, retail, manufacturing, legal and others.

The Agent Loop

At the heart of any agent is what's known as the agent loop and it's surprisingly simple.

This simple mechanism separates agents from traditional LLM applications. Without the loop, you get a single response limited by what the model can do in one shot. With the loop, the LLM works methodically through complex problems, gathering information and adapting its approach.

Ask a language model for a scientific “literature review” and you’ll get a generic summary. Give it access to databases and let it iterate, and you’ll get structured, citable insights. That’s the difference the loop creates.



In the first pass of the loop, the agent might request a search of an FAQ database, on the second a retrieval of the user’s history from the CRM system, and so on. Each step of the loop allows the agent to reason about the problem and the context it’s accumulated, deciding if further tool calls might help it construct a better response.

How To Identify Agent Opportunities

Walk through your business asking people two questions

What do you wish you never had to do again?

What do you wish you could do, but always assumed you can’t?

Use the Example Case Studies provided here to give inspiration for the types of things agents can do. The answers to your questions will reveal agent opportunities — triaging leads, initial contract reviews, monitoring publications, qualifying inquiries, compiling reports.

The Best Candidates

1. **Skilled people on low-value work:**
   Lawyers reviewing standard contracts. Analysts compiling weekly reports.
2. **Frequent enough to matter:**
   Daily or weekly tasks, not annual reviews.
3. **Requires judgement:**
   Some reasoning needed, not just if/then rules.
4. **Data is accessible:**
   Information lives in systems you can connect to.
5. **Stakes are manageable:**
   Agent drafts, human approves, errors are fixable.

Once you've found a candidate, the next step is to build something.

Getting Started

Pick a concrete problem:

The more defined the task, the easier it is to know if it works. Pilots need to be ambitious enough to demonstrate real impact.

Choose technology:

Pick a major AI provider and a mainstream agent framework. The specific choices matter less at this stage than proving the value that agents can bring. Choices can be optimised, or changed, later.

Build your tools:

Tools are the foundation of any agent and transfer easily between frameworks. Good tool design pays off both strategically and in better agent behaviour.

Build an evaluation framework:

Your agent is only as good as the tests that prove it works. Have a ‘gold set’ of expected inputs and outputs, then use this to identify common errors. The industry calls this evals, which really just means repeatable tests.

Run it manually first:

Trigger the agent, review output, decide whether to act. Don't automate until you trust the results.

Then scale:

Once it works, think about more deliberate technology choices and how you scale the approach to ensure teams have the skills and infrastructure they need.

This whole process – prototype to automated execution should take weeks, not months. If you're spending longer planning, you're overthinking it. Build something small, see if it works, expand if it does.

GET YOUR FREE WORKBOOK HERE

Agents aren't magic, they come with real risks. See the sidebar on security, reliability, cost, and bias.

Large or Small Language Models?

Start with large models GPT, Claude, Gemini. They're more capable, and your first priority is proving agents can deliver value. But large models typically mean sending data to external providers—and you often can't customise (fine-tune) them.

Small models offer alternatives: less capable generally, but deployable anywhere (your cloud, on-premise, specific jurisdictions) and fine-tuneable on your private data. A small model trained on your contracts, your processes, or your domain can match large model performance for specific tasks—with full control over where data is processed. But the cost and effort involved in setting this up means this is worth exploring *after* you've proven the value justifies the investment.

Worried about European data and processing sovereignty? Mistral have good models available through both an EU-hosted API and open source options.

Agent Frameworks: The Scaffolding

You need orchestration logic to manage your agent's state and tool calls, and that requires some form of agent framework. If you're starting today, we recommend:

- Aligned with a major vendor? Use theirs. OpenAI's AgentKit, Google's ADK, Anthropic's Claude Agent SDK, or Microsoft’s Copilot are solid starting points.

- Want independence? LangGraph, Mastra and Pydantic are the three open source options we’d suggest looking at.

Don't overthink this. The framework is scaffolding — your real investment is in tools and business logic, which transfer easily if you switch later.

Designing Tools for Agents

You'll want to expose your existing APIs as tools so your agents can both retrieve information and take action. Many enterprise APIs, however, are highly complex. Top-tier models may be able to handle this, but they often still benefit from extra guidance—for example, clear natural-language documentation the model can read. In other cases, you may need to build a simplified "translation layer" that presents your enterprise APIs in a more model-friendly way, especially if you plan to use smaller models with more limited reasoning capabilities.

Build Evals to Measure Performance

Agents are non-deterministic, they can pass a test ten times, then fail on the eleventh. The only way to trust an agent is to test it across multiple cases and measure the failure rate.
Evals (short for evaluations) are how you do this A mature eval setup includes:

AI-as-judge, led by business experts	AI-as-judge, led by business experts: Start with a senior business or domain expert defining what “good” looks like. They review example interactions and make a simple pass/fail call, with a short note explaining why. Once you have enough of these examples, you can have an AI model apply the same pass/fail rules to new interactions, so your testing reflects real business outcomes rather than engineer guesses or vague 1–5 “quality” scores.
Dashboards built around pass	Show which test cases pass or fail, how many are covered, and how this changes over time. Let people slice results by product, customer segment, or risk level, and click into failed cases to see the full interaction and the expert’s comment. The key question the dashboard should answer is: "Where are we failing, for whom, and why?”
Automated evals with clear guardrails	Evals (evaluations) are the tests that prove your agent works. Every time you change how the AI works (model, settings, tools, prompts), you should be able to rerun the evals automatically and flag any drop in pass rate against thresholds agreed with the business owner. Track the cost of both the AI agents and the evals themselves, so you’re not spending more on evaluation than the use case is worth. Feed important failures back into the eval set so both the product and the evaluation improve over time.

Key Risks

Every risk below is manageable but not automatically. The difference between an agent that works and one that embarrasses you is in the implementation details. These are the risks we've learned to design around.

Security: Agents can do damage

An agent with email access can send messages. Database access means records can be deleted.

Mitigations:

Default to read-only access. Get humans to approve high-stakes actions. Log everything.

Reliability: Human oversight

An agent that invents a fact might then find supporting evidence for it, building an internally consistent but completely wrong conclusion.

Mitigations:

For high-stakes decisions: legal, medical, financial use agents for research, not final decisions. Let them surface recommendations with evidence; humans approve or deny. Often this is where the value lies anyway: most of the effort is in the research, not the decision itself.

Cost: API calls add up

A single LLM call might cost a few cents, but when an agent autonomously makes many calls, the costs compound.

Mitigations:

Monitor spending and implement spend limits. Consider fine-tuned smaller models if volumes and costs are concerning.

Bias: Inherited from training data

Agents use language models, so they inherit the same bias risks.

Mitigations:

Test for bias, especially in high-stakes decisions like hiring or credit—these warrant careful analysis.

From One Agent to Many: Scaling Considerations

Once you've proven one agent works, the natural question is: how do we build ten more? Or fifty?

Scaling isn't just replicating your first success, it requires systematic approaches that weren't necessary for a prototype. Some organisations call this an "Agent Factory": standardisation of tools, infrastructure, and processes around agents.

You'll need:

Reusable tools

A shared library of tools that teams can draw from, rather than rebuilding from scratch each time.

Standardised infrastructure

One agent running manually is fine. Multiple teams running agents on production tasks need a common runtime, eval strategy, agent framework and more that they can all be trained on. Think of agents as the new programming model and aim to empower your teams with common infrastructure that lets them focus on the business problems to solve.

Governance

Security policies, cost tracking, audit logging, approval workflows. But if every change requires committee approval, you'll kill velocity, so balance is critical.

The principles remain the same at any scale: start with real problems, build incrementally, measure results.

Key Takeaways

01. What agents actually are	Software that uses AI to reason and adapt. They can automate work that was always too messy for traditional software — too open-ended, too much judgement required.
02. The real opportunity	Efficiency is only half the story. Agents also make it viable to do work that isn't being done today because it was never practical to have humans do it.
03. How to spot opportunities	Skilled people doing low-value work. Frequent tasks. Needs some judgement. Data you can access. Stakes you can manage.
04. How to succeed	Aim to build a working prototype in a handful of weeks, not spend months in planning. The pattern that works: agents do the grunt work, humans review and approve.
05. Risks to manage	Agents can cause damage with the tools you give them. They can also build confident-sounding but wrong conclusions. For critical uses, human review isn't optional.
06. Be bold	The real risk isn't failure — it's timid pilots that prove modest value and kill momentum. Be ambitious enough to show what's actually possible.

Ready to start? Here's how

How Barnacle Labs can help

We've built agents processing millions of documents for the National Cancer Institute and thousands of daily tasks across a variety of other organisations. We help you focus on what matters and skip the academic debates that slow you down.

Two-Week Discovery	We identify the right problem, build a working prototype on real data, and deliver measured results. By the end, you'll know exactly what agents can deliver because you'll have one working. No six-month consulting engagement. No vague "productivity gains." A working agent and a clear decision: scale it or try something else.
Build Engagements	Already past the first agent? We help you scale: reusable tools, standardised infrastructure, governance that doesn't kill velocity. Same approach, real problems, incremental builds, measured results applied across multiple agents and teams.
Get Started	Email us the problem you want solved and we'll tell you how we can help. sales@barnacle.ai barnacle.ai

Agent Examples built at Barnacle Labs

Contract and Document Analysis

At Barnacle Labs we built an agent for legal and procurement teams who review contracts for risks and non-standard terms. The agent analyses documents against defined frameworks, flags concerning clauses, extracts key terms, and presents findings for review.

Result: Hours of manual review reduced to minutes.

Brief Writing

At Barnacle Labs we built agents for a communications agency that gets requests for new projects from an established client base. One agent mines reference material (brand guidelines, past proposals, etc) to create a short brief for the teams that need to work on the requests.

Result: A day of work turns into a few minutes of review.

Research Monitoring

Users need to stay current with research but can't read everything published. At Barnacle Labs we built an agent that continuously scans new publications, identifies papers relevant to each user's interests, and delivers personalised summaries.

Result: Researchers stay current without spending hours scanning journals.

Genomic Data

At Barnacle Labs we built an agent for a genomics company to speed up their ingestion pipeline. Previously 4+ people had to find new studies, parse them, locate the associated data, upload and normalise it. Our agent replaced most of that work for them.

Result: Now only 1 person is required in the ingestion pipeline. The time to getting a new study ingested has been cut down from 25 - 45 minutes to 2-3 minutes

Customer Support at Scale

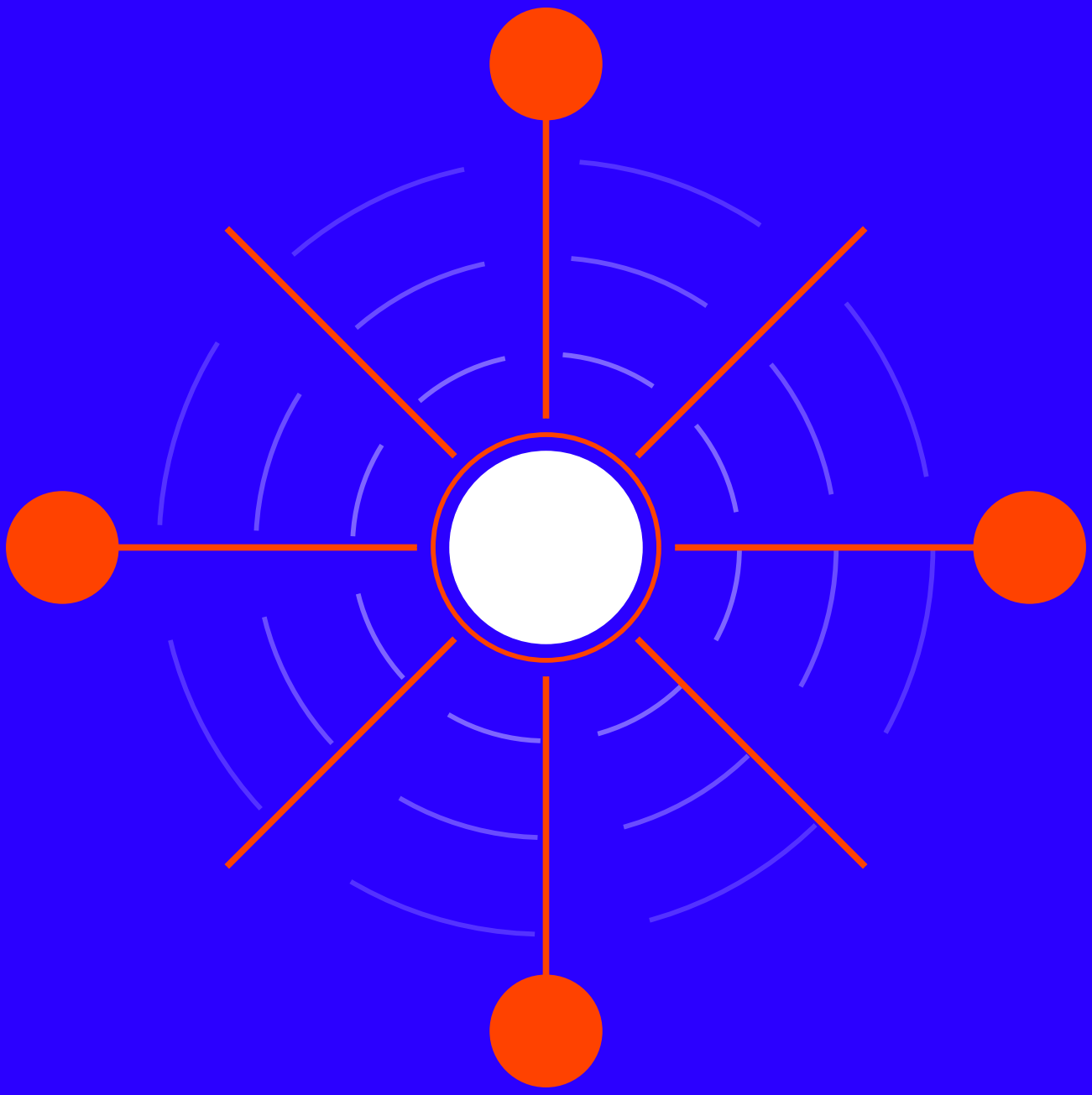
At Barnacle Labs we built an agent for a student accommodation provider operating 70+ halls of residence. Each inquiry triggers an agent that reasons about how to respond—searching FAQs, calling internal APIs for account data, pulling location information.

Result: 100,000 emails eliminated annually.

Newsletter Generation

At Barnacle Labs we built a multi-agent system that researches AI news, spots common themes, writes newsletter copy, and creates social media posts. Our team reviews, adjusts, and approves everything before it's published.

Result: 6 hours of manual work transformed into 30 minutes of review.



About us

Barnacle is an engineering led AI consultancy. We help organisations work out where AI will actually move the needle, not in theory but inside their real constraints. Once the direction is clear, we design and build the systems that deliver it.

We make it easier for senior teams to make good decisions about AI, and we give them the technical capability to turn those decisions into working systems. We've done this for multinationals, high pressure scientific organisations, and fast moving scale ups. The outcome is always the same. A clear plan and software that solves a real problem.