



The Context Layer · Following

 Member-only story Featured

Microsoft Just Quietly Released Amplifier — Here's Why It Might Change How You Automate AI

5 min read · 3 days ago



Jannis



Follow

 Listen Share More

Exploring how Amplifier connects AI agents, memory, and workflows into a compounding automation layer.

Microsoft quietly pushed Amplifier to GitHub. A full system for building and coordinating AI agents that learn from context over time. Let's look at what's actually in the repository, how to set it up and what works right now.



Hi, this is Jannis.

Amplifier is an early-stage research framework designed to explore how specialized agents, memory, and workflows can combine into self-improving automation.

The creators describe it as a “coordinated and accelerated development system.” In practice, it’s an orchestration framework that manages multiple AI agents working on related tasks while maintaining persistent knowledge between runs.

The idea on how to keep context and scoped knowledge for each agent sounds interesting. Each agent has a descriptive name. From *Zen Architect* to *Bug Hunter* to *Security Guardian*. Each one is built for a different workflow domain. These are like modular experts that share a common knowledge base.

Instead of re-prompting from scratch every time, Amplifier stores insights, documents, and reasoning artifacts so that future runs start smarter.

The repo structure makes this clear:

```
agents/  
  zen-architect/
```

```
bug-hunter/  
secu...  
knowle... Welcome back. You are signed into your member account  
workflo... bg****@jaxondigital.com.  
cli/
```

Setting It Up Locally

The setup is straightforward if you're comfortable with modern dev stacks.

You'll need:

- Python 3.11+
- Node.js 18+
- pnpm for package management

Clone the repo and bootstrap the environment:

```
git clone https://github.com/microsoft/amplifier.git  
cd amplifier  
pnpm install  
pnpm run build
```

For Python tasks:

```
python -m venv venv  
source venv/bin/activate  
pip install -r requirements.txt
```

Running `pnpm start` launches the command-line interface. The system then prompts you to choose or define an agent and workflow. Amplifier initializes its workspace under `/amplifier/knowledge`, which becomes the persistent memory for all agents.

Welcome back. You are signed into your member account
bg**@jaxondigital.com**.

It's meant to show what an "AI coordination kernel" could look like — an early vision of tools that connect reasoning, context, and automation.

Current State: Works Best with Claude

The README points out that **knowledge extraction currently performs best in the Claude environment**, meaning Amplifier has been tuned primarily for **Anthropic models**. That's because the framework relies on conversational context and long-form memory — capabilities that Claude handles with high consistency.

You can still experiment with other providers by adjusting your environment variables, but some of the persistent knowledge features and summarization steps are clearly optimized for Claude's API.

In practical terms, that means you can feed Amplifier a collection of files or documents, and its knowledge engine will chunk, summarize, and store those references in its database for reuse in subsequent workflows.

How The Agents Work

Every Amplifier agent has:

- A **specialized goal** (e.g., architecture design, debugging, security review)
- A **prompting strategy** (predefined system and user templates)
- Access to **persistent memory**

- Integration with the central workflow engine

Welcome back. You are signed into your member account

bg**@jaxondigital.com.**

When you loads relevant knowledge, and executes its steps through the workflow runner.

Here's a simplified example from the *Zen Architect* agent configuration:

```
{
  "name": "zen-architect",
  "purpose": "Design and refine system architectures",
  "methods": [
    "analyze_requirements",
    "propose_structure",
    "evaluate_tradeoffs"
  ]
}
```

During execution, Amplifier creates **parallel work-trees** — separate reasoning branches that explore multiple approaches to the same problem. This is one of its most interesting features. Instead of returning a single deterministic answer, it lets multiple solutions evolve concurrently and compares them based on reasoning trails.

That process mirrors what experienced developers already do manually: test alternative solutions, compare pros and cons, and refine.

Building a Practical Workflow

The README includes a set of sample workflows you can study and modify. A typical use case looks like this:

Goal: Summarize a set of research papers and generate a design brief based on their insights.

1. **Load data:** Place your `.txt` or `.pdf` files into `/knowledge/sources/`.

2. **Run the knowledge extraction task:**

```
pnpm run
```

Welcome back. You are signed into your member account
bg**@jaxondigital.com**.

Amplifier parses and chunks your documents, then summarizes them through Claude.

3. Launch an agent to use that knowledge:

```
pnpm run amplify agent zen-architect
```

The agent queries the new knowledge base and generates an architecture plan grounded in what it learned from the documents.

The memory component is visible under `/knowledge/data/`, where JSON files store structured representations of extracted facts and prior runs.

The entire process takes between 10–30 seconds per document, depending on size and model latency. That's slow for production, but fast enough for experimentation — especially when you consider that each run deepens the system's internal knowledge network.

It's not meant for app production or continuous deployment yet. But it already gives a tangible feel for what persistent, context-aware automation could look like.

How This Fits in Microsoft's AI Ecosystem

The repository's design suggests that Microsoft is exploring how to connect multiple reasoning agents into one adaptive system. Imagine combining documentation assistants, bug finders, and research summarizers into a unified pipeline that learns continuously.

While Amplifier is still raw, it extends the same philosophy you see in Claude Skills or MCP servers — a world where automation doesn't restart from zero every time but builds on accumulated understanding.

For now, Amplifier is explicitly marked as experimental. It's not accepting external contributions, and its maintainers caution users to test it "at your own risk."

GitHub - nbg**@jaxondigital.com**

Welcome back. You are signed into your member account

Contributed to 1 project · Last commit 1 day ago · GitHub

github.com/nbg****@jaxondigital.com

0 Issues · 2k Stars · 91 Forks

If you found this article helpful, A few claps 🙌, a highlight 🖍, or a comment 💬 really helps.

If you hold that 🙌 button down something magically will happen, Try it!

Don't forget to follow me to stay updated on my latest posts. Together, we can continue to explore fascinating topics and expand our knowledge.

Thank you for your time and engagement!

• • •

🚀 Stay Ahead with The Context Layer

Your MCP Resource Hub — Get the latest Model Context Protocol news, tutorials, and practical advice delivered straight to your feed.

What You'll Get:

- **Breaking MCP News** — First to know about updates and developments
- **Step-by-Step Tutorials** — From beginner tips to advanced strategies
- **Real-World Applications** — See how MCP boosts productivity and automates workflows
- **Community Insights** — Learn from fellow developers and AI enthusiasts

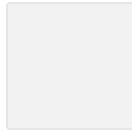
Whether you're just discovering MCP or building advanced solutions, we've got you covered.

👉 [Follow The Context Layer on Medium](#) 👈

Join other developers staying current with MCP innovations.

Microsoft

Welcome back. You are signed into your member account
bg****@jaxondigital.com.



Following

Published in The Context Layer

743 followers · Last published 3 days ago

A journal of real-world experiments, tools, and thinking around Anthropic's Model Context Protocol (MCP)—now established as an open standard for large language model (LLM) context integration.



Follow



Written by Jannis

2.7K followers · 5 following

Product Owner in global telecom, lifelong tech tinkerer, and Mac user. Sharing hands-on hacks, real stories, and the tools that make work (and life) smarter.

Responses (1)



What are your thoughts?



Vishal Goyal
1 day ago

...

I cloned this repository. I'm trying to run it but this error comes up. Could this be all the wrong here?

Welcome back. You are signed into your member account

bg**@jaxondigital.com.**

C:\amplifier>ps

ERR_PNPM_NO_PKG_MANIFEST No package.json found in C:\amplifier



[Reply](#)

More from Jannis and The Context Layer

 Jannis 

This Free OpenAI Alternative Is Exploding: Why LocalAI Belongs On Your Desk

Discover how LocalAI finally brings “OpenAI-level AI” to your own machine. You’ll rethink what’s possible for your next app.

◆ Sep 28

444

6



Welcome back. You are signed into your member account
bg****@jaxondigital.com.

 In The Context Layer by Jannis 

Claude's New "Skills" Show How Anthropic Is Layering Intelligence on Top of MCP

 Oct 17  250  1



...

 In The Context Layer by Jannis 

How Developers Can Auto-Create Claude Skills from Any Framework's Docs

Turn open-source documentation into Claude-ready Skills in minutes, just your terminal and a GitHub repo.

5d ago

Welcome back. You are signed into your member account
bg****@jaxondigital.com.



...

 Jannis 

This Free Microsoft AI Curriculum Is the Secret Weapon for 2025—Don't Miss Out!

Unlock real AI skills—step-by-step, hands-on, and totally free.

5d ago

526

9



...

[See all from Jannis](#)

[See all from The Context Layer](#)

Recommended from Medium

Welcome back. You are signed into your member account
bg****@jaxondigital.com.

In [Dare To Be Better](#) by Max Petrusenko

Claude Skills: The \$3 Automation Secret That's Making Enterprise Teams Look Like Wizards

How a simple folder is replacing \$50K consultants and saving companies literal days of work

 Oct 17  283  4



In [The Context Layer](#) by Jannis 

How Developers Can Auto-Create Claude Skills from Any Framework's Docs

Turn open-source projects into AI with Claude and GitHub Actions. It's like having a terminal and a GitHub repository in one place.

Welcome back. You are signed into your member account

bg**@jaxondigital.com**

5d ago



...

In AI Software Engineer by Joe Njenga

Why Claude Weekly Limits Are Making Everyone Angry (And \$100/Month Plan Will Not Save You)

Yesterday, I finally hit my weekly Claude limit, and I wasn't surprised, since I see dozens of other users online going crazy over these...

5d ago

119

20



...



In CodeToDeploy by TechToFit - Master Your Life with Tech

I Tried Go

Welcome back. You are signed into your member account
bg****@jaxondigital.com.

This is one o



Oct 10

172

3



...



In Stackademic by Somendradev

10 Niche Developer Tools You Didn't Know Existed

Let's be real—the developer world moves fast. Every week, a dozen new tools launch promising to “boost productivity” or “make your life...



Oct 15

120

2



...

Welcome back. You are signed into your member account
bg****@jaxondigital.com.

 In Coding Nexus by Code Coup

OpenSpec: A Spec-Driven Workflow for AI Coding Assistants (No API Keys Needed)

I've been using AI coding tools for a while—Claude, Cursor, even Copilot. They're fast, sometimes magical... but they also love to...

 6d ago  48



...

[See more recommendations](#)