

The background of the image is a dark, futuristic digital space. It features a large, stylized profile of a human head in shades of blue and red, with glowing orange and yellow highlights on the forehead and around the eye area. To the right of the head is a glowing green brain circuit board graphic, enclosed in a red octagonal frame. The overall aesthetic is high-tech and cybernetic.

AI Agents Revolutionizing Software Development

AI Agents Revolutionizing Software Development

From MCP to Multi-Agent Systems



The New Development Stack

AI agents aren't coming - they're here and transforming how we build software

From individual copilots to collaborative multi-agent systems

AI Evolution Timeline



The Transformation

Traditional Tools → Agent Collaboration

Before

Manual integration • Isolated tools •
Weeks of setup

Now

Automated workflows • Connected
systems • Hours to production

What We'll Explore



MCP Protocol: The "USB-C for AI" - universal connectivity



Agent workflows: From prototype to production faster



Developer tools: Real platforms you can use today



Future tech: What's coming for builders and makers

The Developer Experience

-  **Speed:** AI makes development faster - that's a given
-  **Real tools:** Production-ready platforms, not demos
-  **Open source:** European alternatives, no vendor lock-in
-  **Future vision:** Natural language programming by 2027+

About the Author

Sebastian Heußer

Full Stack Developer & Technical Writer

Professional

Currently at STACKIT

 Gov Tech Ambassador

 Hackathon Landing Page - Vibe coded
with passion

Skills & Interests

 Full Stack Developer

 Technical Writer

 Tech Enthusiast

 Nature Lover



Model Context Protocol: The "USB-C for AI"



Universal AI Connectivity

One protocol connects any AI tool to any data source

No more custom integrations for each tool combination

1,900+ servers already available to connect



Standardized connectivity solving the $N \times M$ integration problem

Developer Benefits

What This Means for You

Write once, connect everywhere - MCP server works with any AI tool

Smarter context - 75% fewer tokens through selective queries

Production ready - thousands using it daily already

From prototype to production in hours, not months

Agent-to-Agent Communication



Agents That Talk to Each Other

Beyond human-to-AI interaction

- 👉 **Collaborative intelligence:** Agents working together autonomously
- ⚡ **Dynamic capabilities:** Discovering and negotiating interactions
- 🔄 **Multi-modal support:** Text, audio, video, structured data

Real-World Applications



Development

Code review • Testing • Documentation • Deployment



Critical Systems

Medical flagging • Safety monitoring • Quality assurance



Content

Documentation • Localization • Content generation



Community

100k+ developers • Active open source ecosystem

The Integration Revolution

What Changed for Developers

Before MCP

- Custom integrations for each tool
- Weeks learning APIs
- Brittle, breaking connections

After MCP

- One protocol, any tool
- Minutes to connect
- Reliable, secure standard



Development Ecosystem

⚡ AI Makes Development Faster

That's a given - but there's more to the story

Rapid Prototyping

Ideas become functional
demos in minutes

Team Collaboration

Shared context and
consistent patterns

Iterate Instantly

Change requirements, agents
adapt

Developer Tools & Platforms



n8n: Visual Workflows

230k developers using visual AI workflows

70 AI nodes ready to drag & drop

Open source - self-host or cloud



Hugging Face

1M+ models - easy discovery and deployment

The GitHub for AI models

Deploy anything in minutes



Mistral AI

Apache 2.0 licensing

Devstral specializes in code generation

European alternative - run locally or cloud



Developer Benefits

Open source first - full control

No vendor lock-in - switch anytime

Privacy by design - your code stays yours

What You Can Build Today



Quick Prototypes

n8n • Hugging Face • Replit

Visual workflows, model
testing



Production Apps

MCP • Mistral • Local models

Self-hosted, scalable,
enterprise-ready



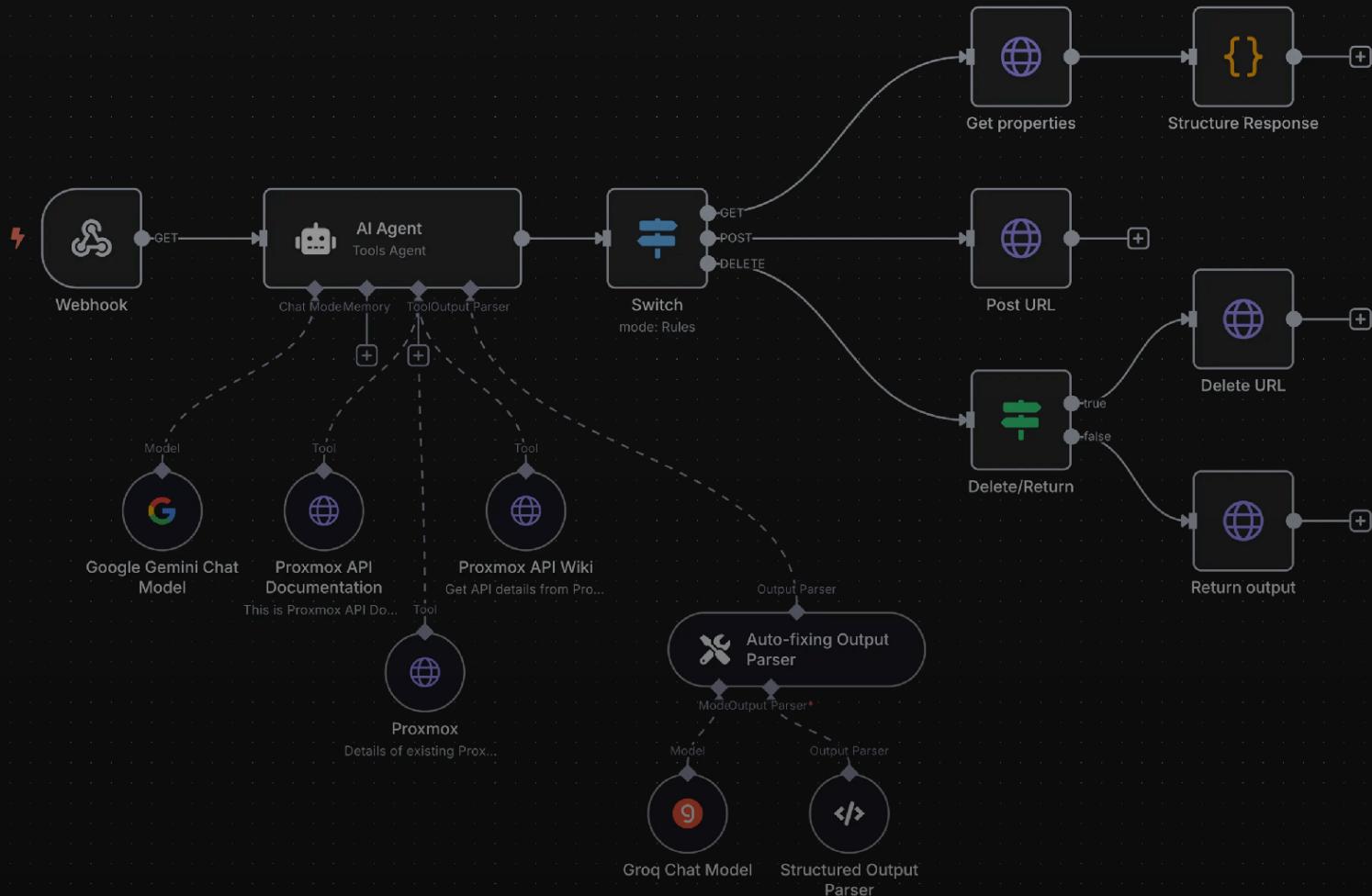
Developer Tools

IDE extensions • Code
assistants

Cursor, Zed, VSCode -
integrate anywhere



European alternatives mean no vendor lock-in



CLI Agents: Terminal-Native AI



AI Agents in Your Terminal

Command-line AI assistants that integrate directly into developer workflows

Claude Code

Official Anthropic CLI

- Direct file access and editing
- Project-wide understanding
- Multi-step task automation

Gemini CLI

Google's multimodal CLI

- Image and document analysis
- Code generation from screenshots
- Natural language queries

Why CLI Agents Matter

⚡ Speed

No context switching

Stay in terminal flow

🔧 Integration

Works with existing tools

Shell scripts & automation

🎯 Focus

Task-specific assistance

No UI distractions

💡 Real Developer Workflow

```
claude "refactor this function for better  
performance"
```

```
gemini analyze-design.png "generate React  
components"
```

```
gh copilot suggest "deploy to  
kubernetes"
```

```
codex "write bash script to backup  
database"
```



Tips for getting started:

1. Ask questions, edit files, or run commands.
2. Be specific for the best results.
3. `/help` for more information.

```
> write a short paragraph about why Gemini CLI is awesome
```

- I will start by searching the web for "Gemini CLI" to understand its main features and purpose. Following that, I'll examine the `README.md` file for a project-specific overview. I will then consult `docs/index.md` and `docs/cli/commands.md` to gain a deeper understanding of its capabilities. Finally, I will write a short paragraph summarizing why Gemini CLI is awesome.

```
↳ GoogleSearch Searching the web for: "Gemini CLI features and purpose"
```

```
: Uncovering Gemini's Awesome (esc to cancel, 21s)
```

```
Using 3 GEMINI.md files
```

```
~/code/gemini-cli (release*)
```

```
no sandbox (see /docs)
```

```
gemini-2.5-pro (99% context left)
```



Let me show you Gemini CLI

Real terminal session, real prompts, real results

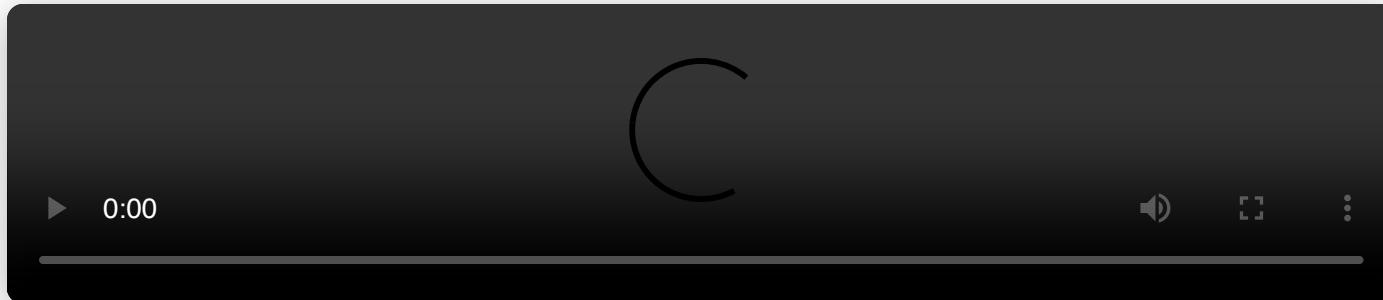


What You'll See

- **Me typing prompts** into Gemini CLI
- **AI generating code** in real-time
- **Iterative improvements** through conversation

The Gemini CLI Session

Me typing, AI generating, game improving



🎯 What Just Happened

✓ First Prompt Success

✗ Issues Found

Two More Prompts to a playable game

2

Graphics Fix

"the graphics are not real,
create own graphics"

3

Polish Game

"the game is currently very hard
to play"

Let's Play!

The polished result after 3 prompts



🎮 What You See

- Smooth gameplay
- Balanced difficulty

📊 The Numbers

- 3 prompts total
- ~10 minutes work

What This Really Shows

The Power of Conversational Programming

For Makers

- Turn ideas into prototypes instantly
- Iterate through natural conversation
- Focus on what, not how

For Developers

- Rapid prototyping capabilities
- Generate boilerplate instantly
- Easy to refine and improve



Near Future: 2025+



Larger Context Windows

Process entire codebases in single conversations



Multi-modal Capabilities

Code from screenshots, diagrams, and voice commands



Real-time Collaboration

Multiple developers + AI agents working together



Industry Standards

Unified protocols for agent communication

The Autonomous Future

AI agents become software architects

Fully Autonomous Development

AI agents handle entire project lifecycles

Natural Language Programming

"Build me a social network" → Working application

Self-improving Systems

AI agents optimize their own capabilities

AI Software Architects

System design and architecture by AI

Start Building Today

 **Pick a tool:** n8n for workflows, Cursor for coding, Hugging Face for models

 **Set up MCP:** Connect your first AI tool to your data

 **Build something:** Even a simple workflow teaches you the patterns

"AI agents are tools, not competitors"

What You Can Build Next



Automated Workflows

Deploy, test, and monitor with AI agents



Code Assistants

Custom tools that know your codebase



Documentation Bots

Always up-to-date project docs



Testing Agents

Generate and run tests automatically

Thank You!



Questions & Discussion

Slides

<https://github.com/huhn511/ai-on-hackathons>



AI Agents Meet the Physical World

Software agents are gaining physical bodies

From code generation to real-world manipulation

Traditional Robots

Programmed instructions • Expert
programming required

AI-Powered Robots

Natural language control • "Clean the table"
commands

\$250 robot arms controlled by GPT-4o • **GitHub repositories** with complete builds

What Developers Can Build

Same Skills, Physical Endpoints

 **Natural language interfaces** - Same as ChatGPT

 **API-first design** - REST endpoints control actions

 **Familiar frameworks** - ROS 2, Python, Node.js

 **Cloud connectivity** - Robots as IoT devices

Open Source Tools

ROS 2: Millions of developers

NVIDIA Isaac: Apache 2.0 license

Programming Shift

Traditional: Complex code

AI-Powered: "Pick up the red cup"

Embodied AI

LLMs for reasoning

Computer vision for environment

