

Google Antigravity unter Linux

Der nächste Schritt der KI-Evolution – Live am Linux Tag

Erlangen

Referent: Franz Mertens

Zielgruppe: Anfänger & Fortgeschrittene

<https://transform2agile.de>

Was ist Antigravity?

Ein Blick hinter den Hype der "Agent-First"
Entwicklung

Die Vision: Agent-First

Vom Co-Piloten zum Kapitän

Bisherige Tools wie Copilot waren Assistenten (Autocomplete). Antigravity ist ein **Akteur**. Es übernimmt komplette "Missionen" – vom Plan bis zum Deploy.

Integrierte Autonomie

Durch direkten Zugriff auf Terminal, Browser und Editor agiert das System als virtueller Junior-Entwickler, der Aufgaben selbstständig validiert.

Kern-Technologien & Features



Gemini 3 Engine

Basiert auf Gemini 3.1 Pro mit einem gigantischen 1-Million-Token Kontext-Fenster.



Multi-Agenten

Spezialisierte Agenten für Backend, Frontend und Testing arbeiten asynchron zusammen.



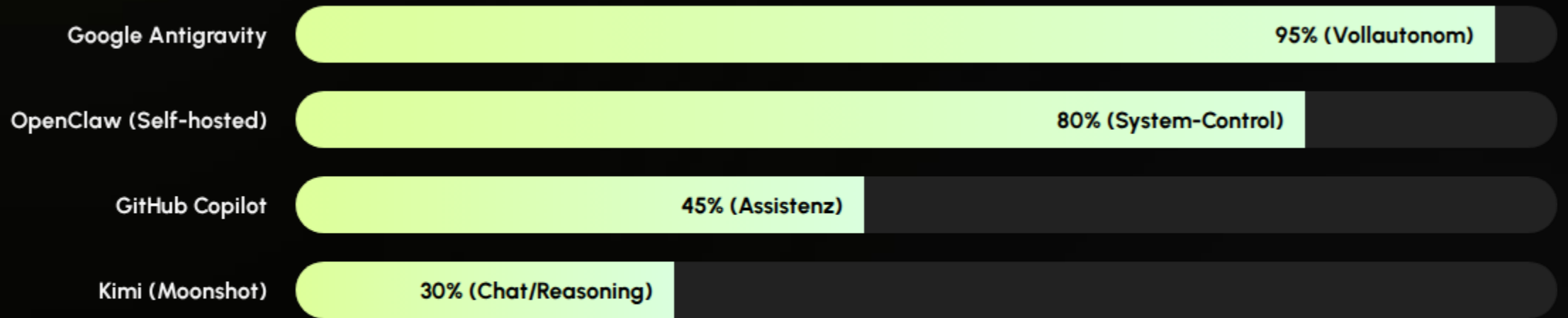
Artifacts

Generiert verifizierbare Deliverables: Task-Listen, Screenshots und Test-Berichte.

Technische Voraussetzungen (Linux)

Komponente	Anforderung	Beispiel / Hinweis
Betriebssystem	64-bit Linux	Ubuntu 20.04+, Fedora 36+, Debian 10+
System-Bibliotheken	glibc >= 2.28	Prüfen via <code>ldd --version</code>
Hardware	Mind. 1GB RAM	4GB+ für flüssige Multi-Agent Workflows empfohlen
Software-Stack	Chrome & Gmail	Account für Preview-Zugang notwendig

Marktvergleich: Autonomie-Level



AntigraVity führt durch die Kombination aus Modell-Intelligenz und Tool-Interaktion.

Stärken & Grenzen

Was es kann

- ✓ Komplexe Refactorings über Repos hinweg
- ✓ Eigenständiges Browsing für Doku-Recherche
- ✓ End-to-End Testing im integrierten Browser

Was es (noch) nicht kann

- ✗ 100% deterministische Logik garantieren
- ✗ Offline-Nutzung (Cloud-basiertes Reasoning)
- ✗ Ersetzen von Software-Architekten

Multi-Agent Orchestrierung

Das "Swarm" Prinzip

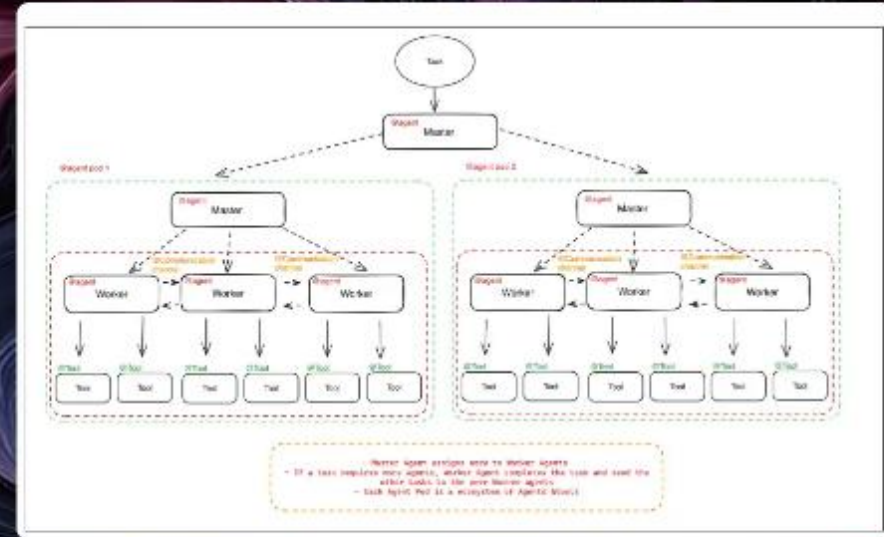
Antigravity nutzt spezialisierte Agenten-Profile:

Backend-Agent: Erstellt APIs und DB-Schemata.

UI-Agent: Implementiert React/Vue Komponenten.

Security-Agent: Scannt Code auf Schwachstellen.

Sie kommunizieren über eine zentrale "Mission Control" und lösen Abhängigkeiten asynchron auf.



Unter der Haube: Technik



- **Gemini 3.1 Pro (40%)**: Das Gehirn & Reasoning
- **LSP & Tool-Interaktion (30%)**: Die Hände im Editor
- **Vektordatenbanken/RAG (30%)**: Das Langzeitgedächtnis

$$A_{\text{performance}} = M_{\text{reasoning}} \times C_{\text{context}} \times T_{\text{tools}}$$

Gefahren & Risiken



Sicherheitslücken

Gefahr durch automatische
Terminal-Befehle oder JS-
Execution im Browser.



Account-Risiko

Verstöße gegen Google Terms
(z.B. via Drittanbieter-Tools)
führen zu Sperren.



Halluzinationen

Kritische Logik-Fehler in
komplexen multi-agenten
Workflows möglich.

Sichere **Nutzung** unter Linux

● Setup

Installation in isolierter VM
oder Container.

● Review Mode

Agent muss jede Aktion im
Terminal bestätigen.

● Safe Mode

Einschränkung externer
Ressourcen & Netzzugriff.

● Full Trust

Vollautomatisierung nach
Workflow-Validierung.

Vielen Dank!

Fragen? Jetzt ist die Zeit!

 antigravity.google

 github.com/antigravity-community

Image Sources

https://cdn.prod.website-files.com/66841c2a95405226a60d332e/682494b9be49b2179a8a7fb4_Agent_Orchestrator_compressed.webp

Source: www.cohorte.co