Von MCP bis Agentic Optimization
Handout zur Präsentation

AI-Agenten in Medienanwendungen

Joshua Aleth

27. April 2026

Inhaltsverzeichnis

1 Was sind AI-Agenten?

AI-Agent AI-Agenten (Künstliche-Intelligenz-Agenten) sind **autonome Softwaresysteme**, die darauf ausgelegt sind, komplexe Ziele zu erreichen, indem sie:

- ihre Umgebung **wahrnehmen**
- eigenständig **Entscheidungen treffen**
- proaktiv **Handlungen ausführen**

ReAct Framework [?] **ReAct = Reasoning + Acting**
LLMs können *reasoning* (denken) und *acting* (handeln) synergistisch kombinieren:

1. **Perceive:** Umgebung beobachten
2. **Reason:** Mit LLM denken/planen
3. **Act:** Tool auswählen und ausführen
4. **Reflect:** Ergebnis bewerten, lernen

Was macht Agenten mächtig?

- **Tool-Use:** Zugriff auf APIs, Datenbanken, Web
- **Planning:** Mehrstufige Aufgaben zerlegen
- **Self-Correction:** Eigene Fehler erkennen
- **Memory:** Langzeit-Kontext speichern

Was macht Agenten gefährlich?

- **Autonomie:** Handeln ohne menschliche Freigabe
- **Skalierung:** Viele Agenten parallel
- **API-Zugriff:** Externe Systeme steuerbar
- **Prompt-Injection:** Manipulation möglich

2 Agenten für Medienanwendungen

Warum Agenten für Medien?

- Generative KI: Video, Musik, Bilder
- Komplexe Pipelines benötigen Orchestrierung
- Agenten automatisieren den gesamten Workflow

Beispiel: ASCII-Video Produktion **Workflow:** Script schreiben → TTS generieren → ASCII rendern → MP4 muxen
Alles **autonom durch Agenten** orchestriert.

3 Agentic Optimization

Problem: Websites sind nicht agenten-freundlich

- Dynamische IDs, JavaScript, Anti-Bot
- Agenten müssen wie menschliche User navigieren
- Mind2Web: Generalistische Agenten für *any* website [?]

API-first	UI-first (HoloTab)
Strukturierte JSON-Daten	DOM-Parsing nötig
Wenig Token-Verbrauch	Hoher Token-Verbrauch
Direkte API-Calls	Viele Interaktionsschritte
Stabil & schnell	UI-Änderungen brechen Agent
Agent-optimiert	Mensch-optimiert

Empfehlungen für Agenten-freundliche Websites

- **Stabile Selektoren:** `data-agent-*` Attribute
- **API-First:** JSON-Endpoints dokumentieren
- **Strukturierte Daten:** Schema.org Markup
- **Rate-Limiting:** Agenten-freundliche Limits

4 MCP Protokoll

Model Context Protocol (MCP) Standardisiertes Protokoll für Tool-Integration – wie **USB für Tools**:

- Client-Server Architektur
- Tool-Discovery (automatisch erkennen)
- Einheitliche Auth
- Teil der "HarnessËxternalization [?]

RAG vs. MCP

	RAG	MCP
Kategorie	Context (Wissen)	Harness (Tools)
Problem	LLM weiß zu wenig	LLM kann nicht handeln
Lösung	Wissen retrieven	Tools aufrufen
Ergebnis	Mehr Kontext	Aktion ausgeführt

5 Risiken und Ethik

Sicherheitsrisiken

- Third-Party API Angriffe [?]
- Prompt-Injection über Tools
- Datenlecks durch externe Integration
- Unautorisierte Aktionen

Ethische Fragen

- **Bias** in generativen Systemen [?]
- **Copyright:** Wer owns generierte Medien?
- **Autonomie:** Was wenn Agenten falsch entscheiden?
- **Transparenz:** Nachvollziehbarkeit von Entscheidungen

6 Fazit

Zusammenfassung

- AI-Agenten = LLMs mit Superkräften (Tools + Autonomie)
- MCP standardisiert Tool-Integration (Harness Layer)
- Websites sollten für Agenten optimiert werden
- Risiken: Security, Bias, Copyright

Ausblick

- Mehr MCP-Server für Medien-APIs
- Bessere Evaluation von Agenten-Fähigkeiten
- Ethik-Richtlinien für autonome Systeme

Literatur

[?] Yao et al.: *ReAct: Synergizing Reasoning and Acting in Language Models.* arXiv:2210.03629, 2022.
[?] Zhou et al.: *Externalization in LLM Agents: A Unified Review.* arXiv:2604.08224, 2026.
[?] Rahman et al.: *LLM-Based Data Science Agents: A Survey.* arXiv:2510.04023, 2025.
[?] Deng et al.: *Mind2Web: Towards a Generalist Agent for the Web.* NeurIPS 2023.
[?] Zhao et al.: *Attacks on Third-Party APIs of Large Language Models.* arXiv:2404.16891, 2024.
[?] Mehta et al.: *Who Gets Heard? Rethinking Fairness in AI for Music Systems.* arXiv:2511.05953, 2025.
[?] Ding & Stevens: *Unified Tool Integration for LLMs.* arXiv:2508.02979, 2025.

Alle Paper verfügbar unter: arxiv.org