Souveräne KI in der Praxis

Modelle, Hardware und Agenten-Workflows vor der Kaufentscheidung evaluieren

Rainer Poisel

Überblick

- Motivation und Zielbild
- Evaluierungsprozess und -design
- Architektur des Inference Lab
- Sicherheit, Souveränität und Grenzen

Rainer Poisel

- DevSecOps für Embedded Systems
- CI/CT/CD für Embedded Systems
- Entwicklungsproduktivität



Automatisierung, Sicherheitstests (IEC 62443, CRA)

Einführung

Large Language Model (LLM)

> A large language model (LLM) is an AI model trained on a vast amount of text for natural language processing tasks, especially language generation.

Quelle: IBM – What are large language models?

- Generative pre-trained transformers (GPTs)
- ChatGPT, Claude, Gemini, Grok, DeepSeek, etc.
- Open/Closed Weights

Agent Harness (I)

> An agent harness is the runtime scaffolding that turns a language model into an agent that can perform work.

Quelle: Microsoft Agent Framework

- Tool calls
- Conversation state + context
- Approval policies

Agent Harness (II)

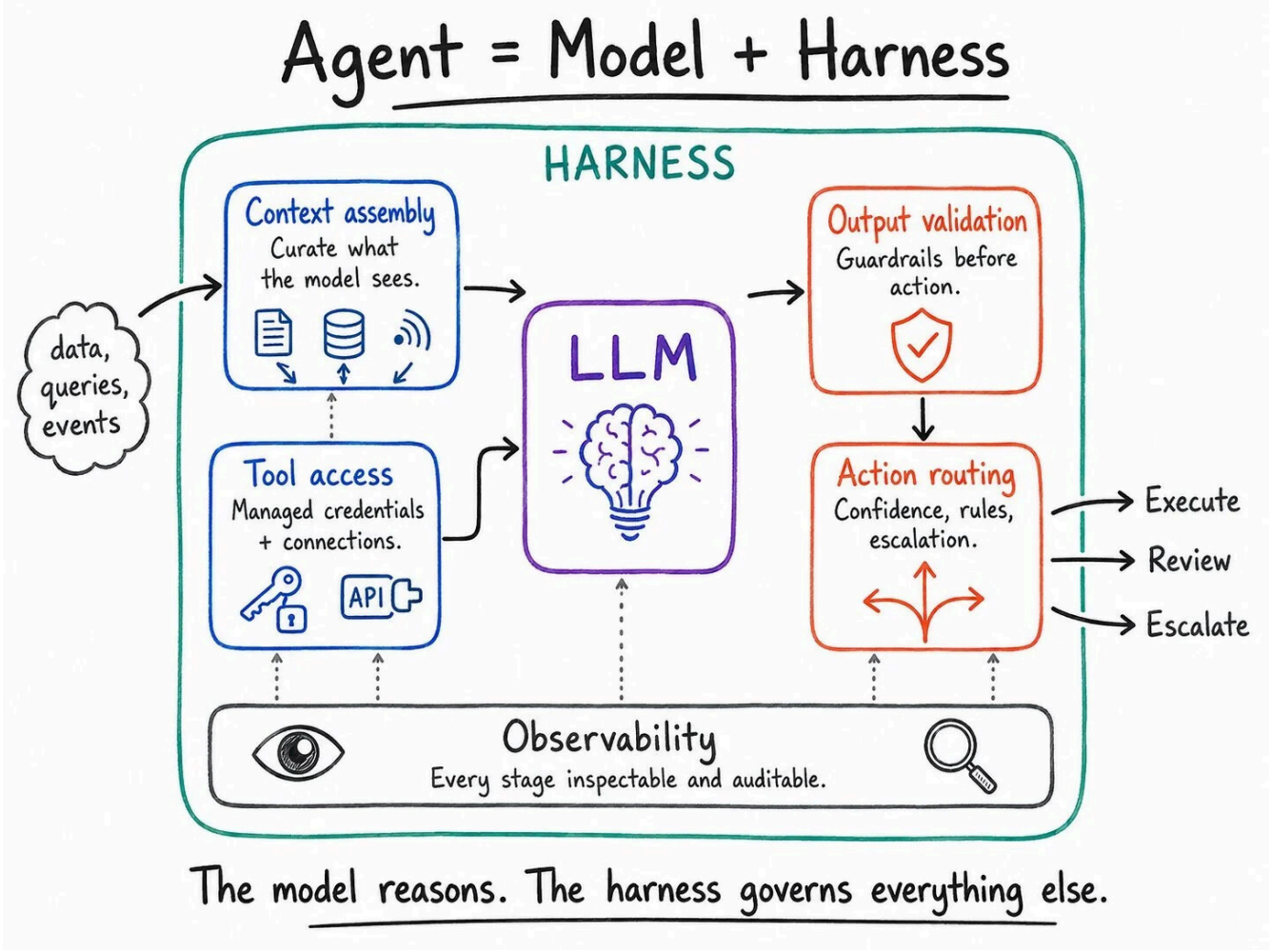


Abbildung 1 — Agent / Model / Harness

Quelle: XMPro – Your AI Agent Needs a Harness

Motivation

Schritt eins

Definition Datensouveränität

> Datensouveränität bedeutet, die Verarbeitung und Weitergabe der eigenen Daten selbst bestimmen und technisch kontrollieren zu können.

Prämisse

Du hast dich für lokale AI Inference entschieden und willst kostengünstig einsteigen.

Variante: Hardwaremiete (I)

- Machbarkeit
  - Ausprobieren vor dem Kauf
- Dimensionierung
  - Hohe Vorlaufkosten
  - Große Unsicherheiten

Variante: Hardwaremiete (II)

- Datenschutz
  - Manche Anwendungsfälle erfordern Air-Gap-Setups
  - Das kann richtig mühsam sein.
- Evaluierung / Lernen
  - Wie funktioniert die Einrichtung?
  - Wie gut würde sich meine Investition auszahlen?

Lösung

- Wir haben das AI Inference Lab erstellt
  - Zentralisierte Bereitstellung
  - Sicherer Rahmen
  - Standardisierte Schnittstellen

Zwei Stufen zur souveränen KI

Stufe 1: Inference Lab

- Gemietet
- Synthetische Daten
- Dimensionierung

Stufe 2: Zielsystem

- Lokal
- Echte Daten
- ✦ **Souverän** ✦

Status: Stufe 1 läuft, heute ein Werkstattbericht

Evaluierungsprozess

1. Ziel und Rahmen klären
2. Aufgaben und Kriterien definieren
3. Hardware und Modelle auswählen
4. Im Inference Lab analysieren
5. Bewerten und entscheiden

Bewertungskriterien

- **Qualität:** Aufgabe zuverlässig lösen
- **Leistung:** Nutzungsprofil bewältigen
- **Kosten:** Investition und Betrieb rechtfertigen
- **Sicherheit:** Daten und Ausführung kontrollieren
- **Datensouveränität:** Abhängigkeiten beherrschen

Evaluierungsdesign

Aufgabenstellungen: Rahmenbedingungen

- Reproduzierbare Ausgangslage
- Objektiv prüfbares Ergebnis
- Realistische Tool-Nutzung
- Genügend Komplexität
- Keine vertraulichen Daten

Aufgabenstellung 1: Test beheben

Vorbereitung:

- Gezielten Fehler einbauen
- Mindestens einen fehlschlagenden Test bereitstellen

Messbares Ergebnis:

- Test besteht
- Keine Regressionen
- Korrekte Behebung (subjektiv!)

Aufgabenstellung 2: CI-Log-Analyse

Vorbereitung:

- Relevanten Fehlschlag auswählen
- Logs extrahieren/bereitstellen

Messbares Ergebnis:

- Richtige Ursache identifiziert
- Symptom und Ursache korrekt getrennt

Aufgabenstellung 3: Sicherheitslücke beheben

Vorbereitung:

- Gut kontrollierbares Beispiel aufbauen
- Codebasis bereitstellen

Messbares Ergebnis:

- Schwachstelle korrekt erkannt
- Regressionstest erstellt
- Ursache behoben

Investitionshypothese A (I)

- Beispiel: Hardware-Set 1
  - Grafikkarte: NVIDIA RTX 5090, 32 GB VRAM
  - 128 GB RAM, AMD EPYC 7452 CPU



Abbildung 2 — NVIDIA RTX 5090
Quelle: TechPowerUp

Investitionshypothese A (II)

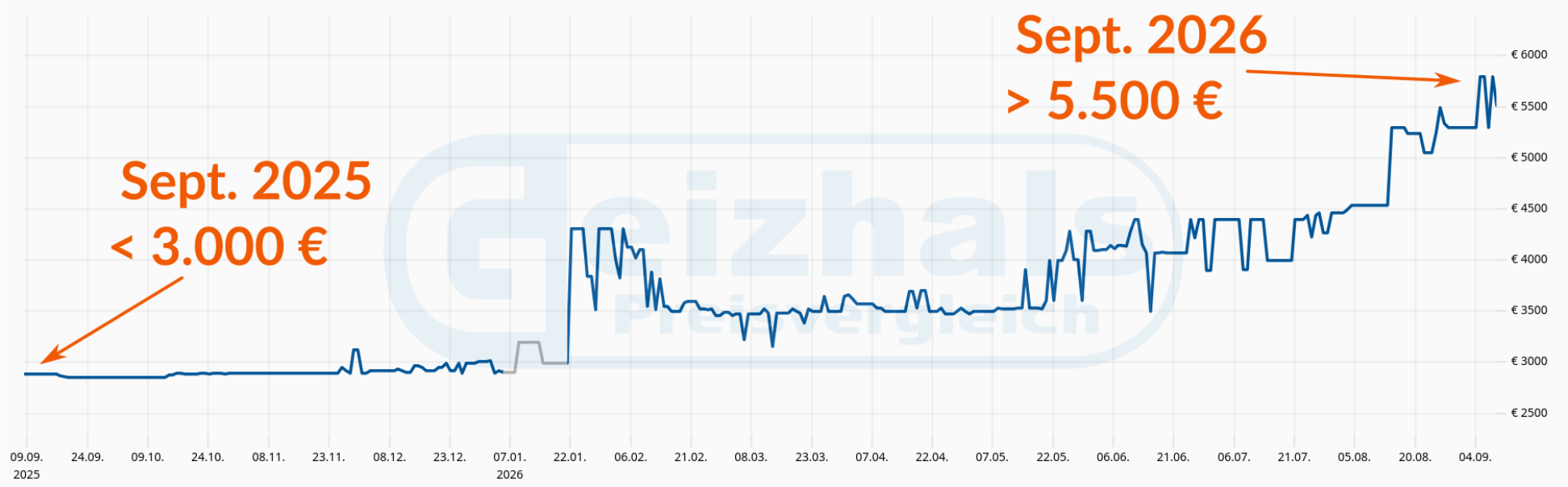


Abbildung 3 — Preisentwicklung: NVIDIA RTX 5090
Daten: Geizhals, Abruf: 09.09.2026

Investitionshypothese B (I)

- Beispiel: Hardware-Set 2
  - Grafikkarte: NVIDIA RTX PRO 6000, 96 GB VRAM
  - 192 GB RAM, AMD Ryzen Threadripper CPU



Abbildung 4 — NVIDIA RTX PRO 6000 Blackwell
Quelle: Leadtek

Investitionshypothese B (II)

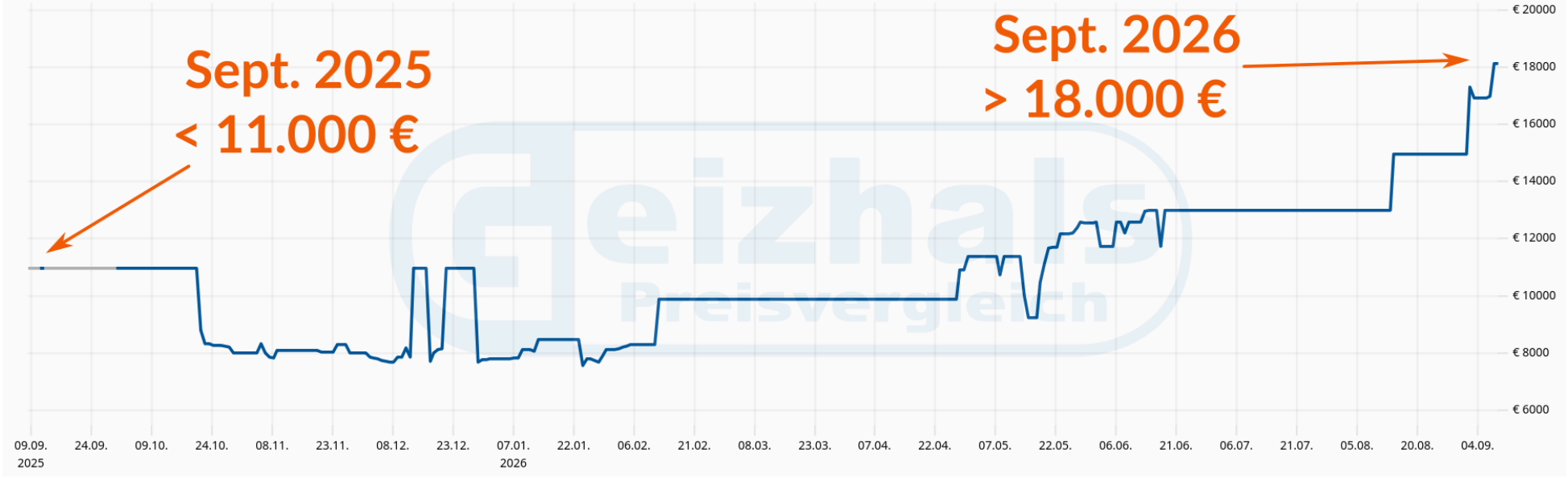


Abbildung 5 — Preisentwicklung: NVIDIA RTX PRO 6000 Blackwell
Daten: Geizhals, Abruf: 09.09.2026

Auswahl der Modelle

- Modellarchitektur und Größe
- Quantisierung
- Kontextlänge und Parallelität
- Agenten-/Inferenzparameter

Konkrete Modellauswahl (I)

Modell	Hardware	Ausschlagende Parameter
gpt-oss-20b	RTX 5090, RTX 6000	21B, 3,6B aktiv, MXFP4
Qwen3-Coder-30B-A3B	RTX 5090, (RTX 6000)	30,5B, 3,3B aktiv, Q4/Q5
Devstral Small 2 24B	RTX 5090	24B, Dense-Modell

Konkrete Modellauswahl (II)

Modell	Hardware	Ausschlagende Parameter
Qwen3-Coder-Next	(RTX 5090), RTX 6000	80B, 3B aktiv, (CPU-Offloading)
gpt-oss-120b	RTX 6000	117B, 5,1B aktiv, MXFP4

Inference Lab

OpenAI-kompatible API

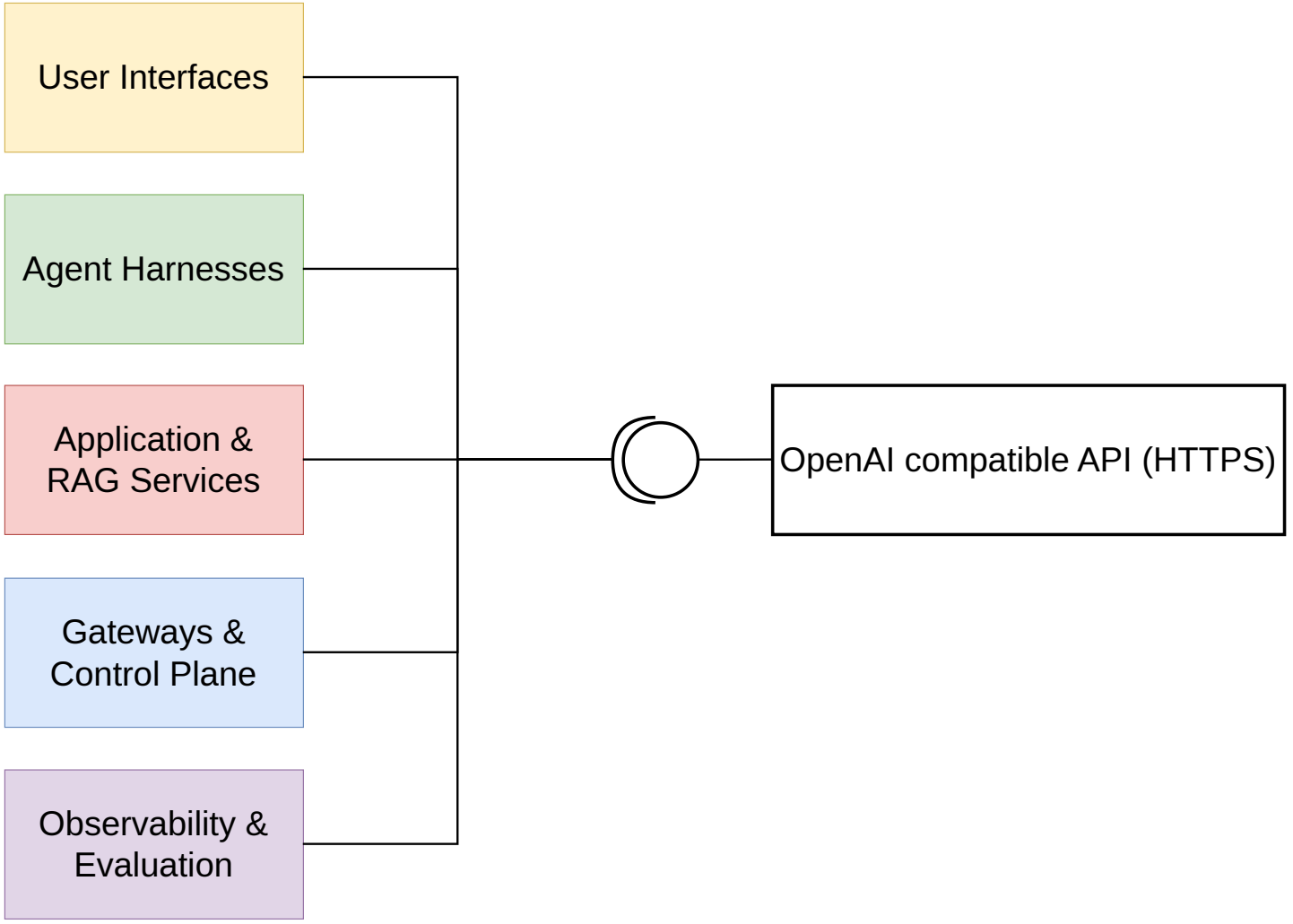


Abbildung 6 — OpenAI-kompatible API

Anbindung LLM (I)

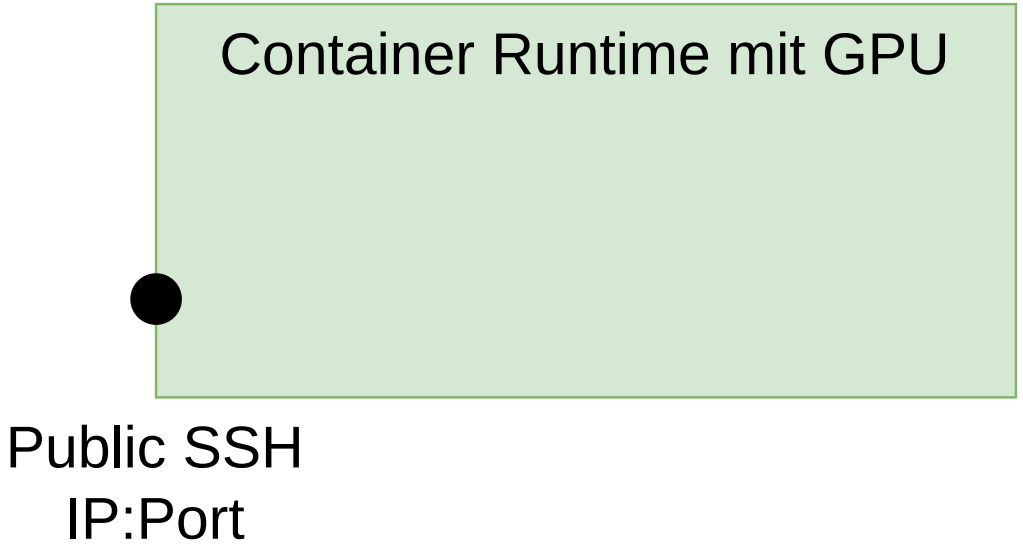


Abbildung 7 — Ausgangslage

Anbindung LLM (II)

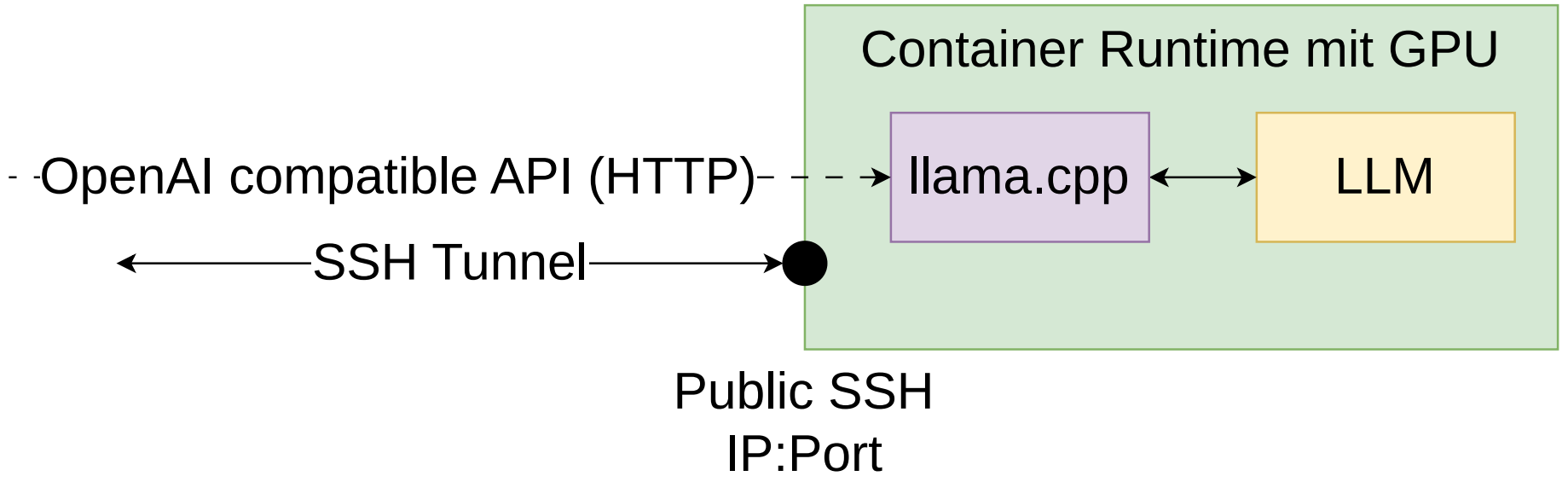


Abbildung 8 — Verbindungspunkte LLM

Anbindung LLM (III)

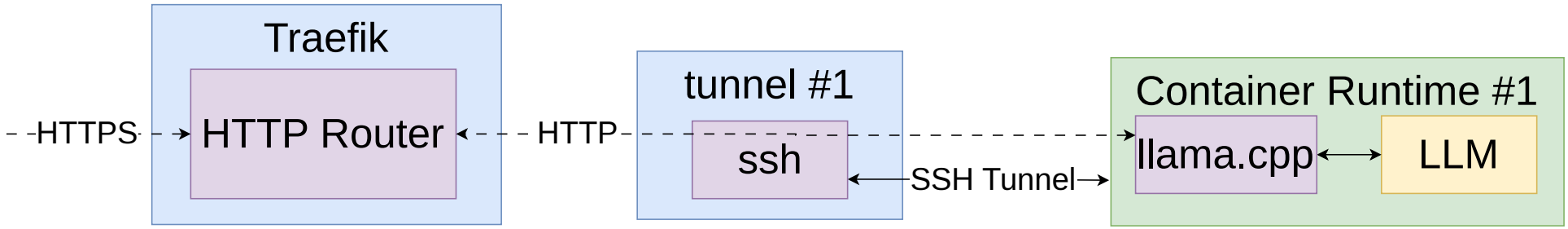


Abbildung 9 — Verbindungspunkte LLM

The Big Picture

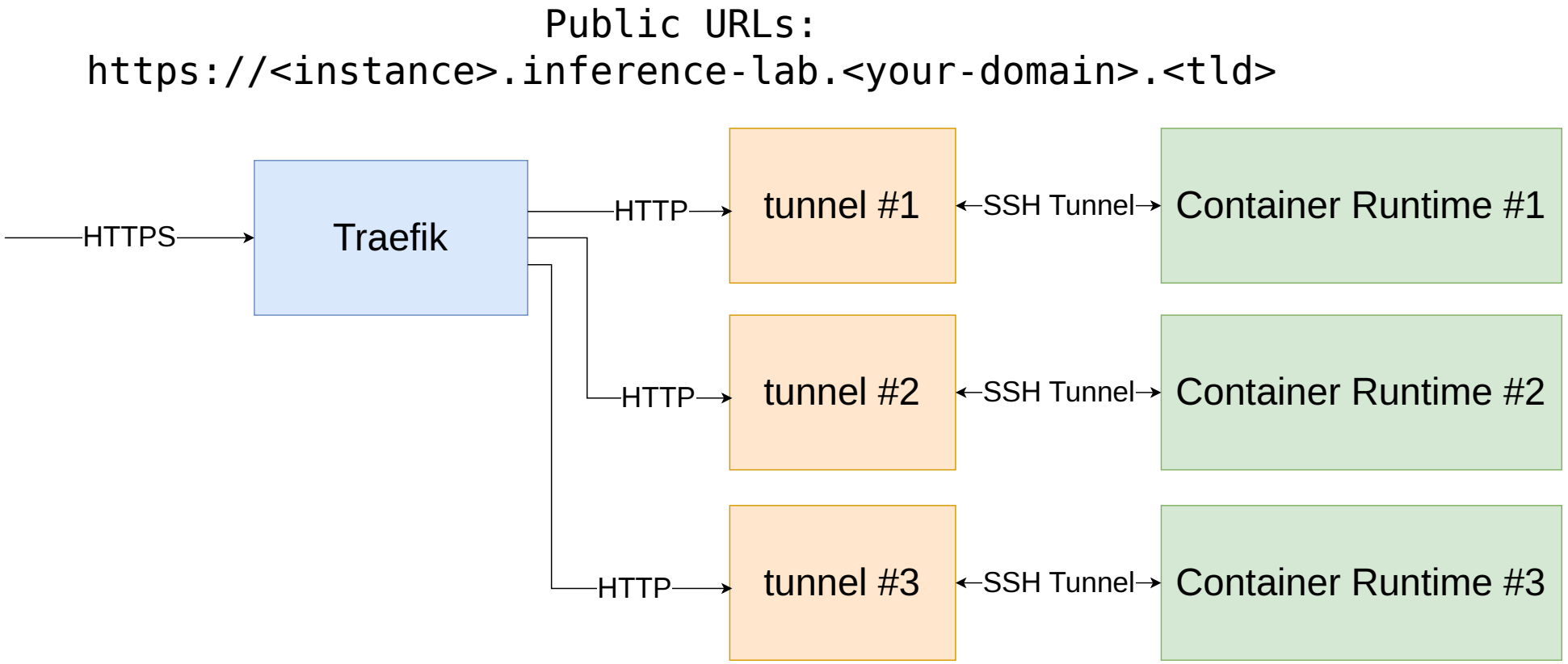


Abbildung 10 — AI-Inference-Lab-Architektur

Details

Infrastrukturanbieter:

- Beispiel: https://vast.ai/
- Zeitbasiertes Preismodell
  - CPU/GPU: nur bei aktiver Verwendung
  - Persistenz: aktiv vs. inaktiv

Proof-of-Concept-Prototyp:

- Einrichtung erfordert manuelle Eingriffe

Grenzen + Auswirkungen

Übersicht

Verwendung von

- User Interfaces
- Agent Harnesses

Einschränkungen

- Organisatorisch
- Technisch

Organisatorisch: Datenschutz

1. **Gemietetes Lab:** synthetische, anonymisierte oder freigegebene Repräsentationen realer Workflows
2. **Kontrollierte Zielumgebung:** vertrauliche Originaldaten lokal oder im Air-Gap validieren

Technisch: Hardwaremodelle

- Hardwaremodelle
  - Dedizierte GPUs
  - Unified Architecture

Technisch: Hardwaredetails

- Details der Hardwarekonfiguration
  - CPU/RAM-Anbindung
  - PCIe-Technologie
  - Storage
  - Treiber und Runtime
  - Kühlung/Power-Limits
  - Parallele Nutzung

Technisch: SRE

- SRE-Perspektive
  - Download der Modelle
  - Knoten könnten verschwinden
  - Knoten könnten nicht erreichbar sein

Hybrid: Privacy (I)

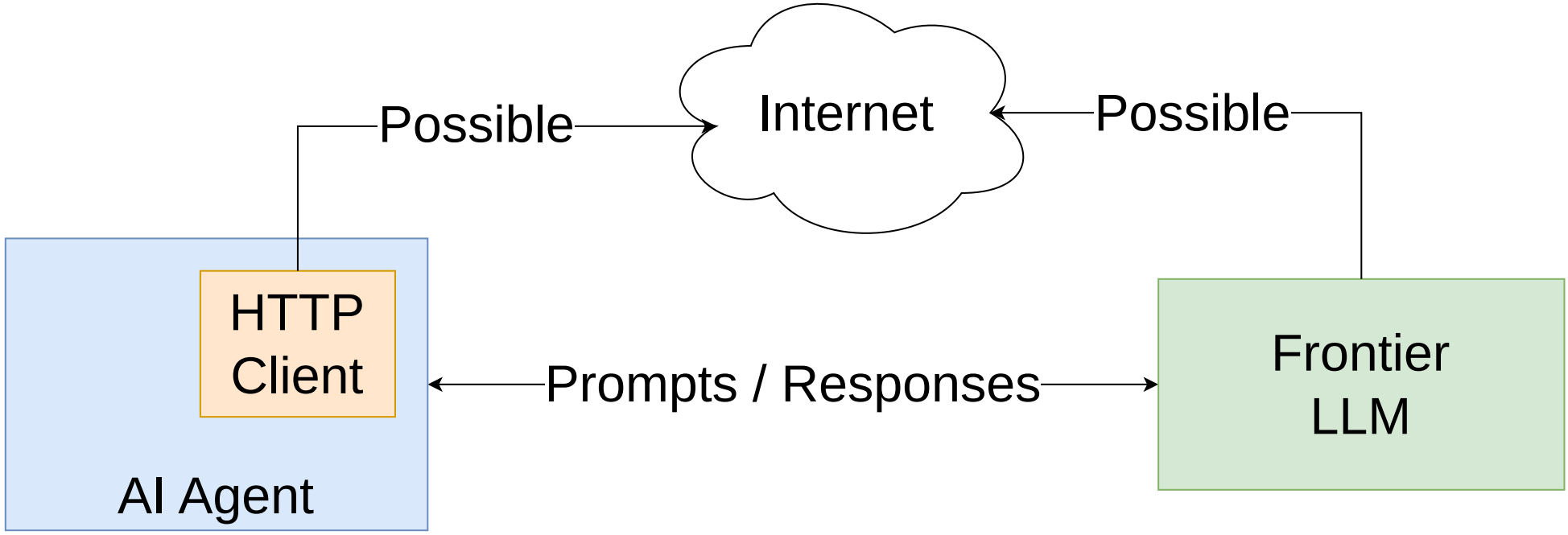


Abbildung 11 — Agent-Harnesses und Internetzugriff

Hybrid: Privacy (II)

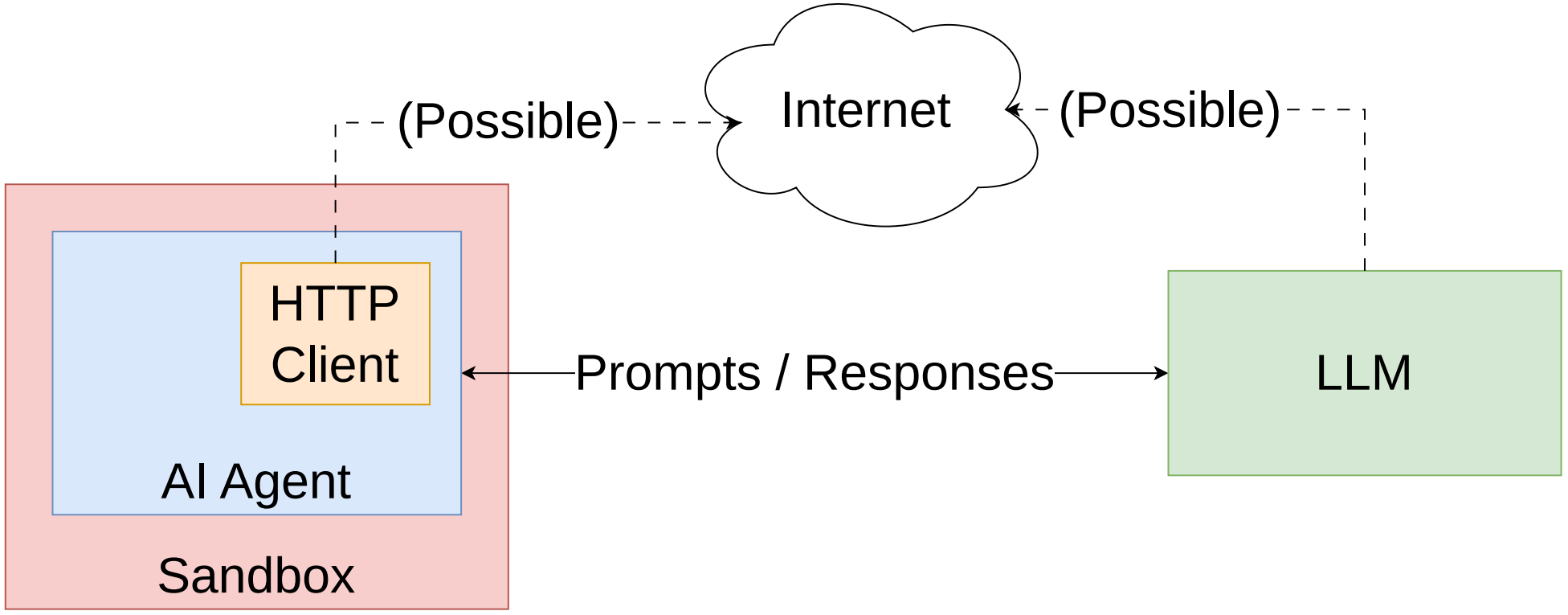


Abbildung 12 — Agent-Harness-Sandbox und Internetzugriff

Sandbox: Agent Circus



GitHub

Abbildung 13 — Agent Circus Sandbox

Agent Circus: Isolationsmechanismen

Isolation

- **Prozessgrenze:** Container als `node`, optional rootless
- **Dateizugriff:** definierte Mounts, read-only oder verborgen

Freigaben

- **Konfiguration:** projektlokale Kopie statt Host-Original
- **Host-Brücken:** Env, SSH, Hosts und Ports nur explizit

> **Grenze:** Container schaffen Isolation – aber keine harte Security Boundary

Zusammenfassung

Kernaussage

- Meine Kaufentscheidung ist noch nicht gefallen
- Gesamtsystem anwendungsspezifisch evaluieren
- Das Inference Lab ermöglicht belastbare Entscheidungen – innerhalb klarer Grenzen

Get in Touch 👋

Sprecht mich an, wenn ihr ...

- AI-Hardware schon evaluiert habt.
- Ideen zu vertraulichen Daten habt.
- selbst vor der Entscheidung steht.

Ich bin noch länger hier. :-) #GernePerDu

Get in Touch 👋



LinkedIn



vCard

Backup-Folien

Nächste Schritte

Agent Sandboxes

- Verbergen von Credentials
- Plugin-System
- Alternative Sandboxing-Ansätze (VMs, MicroVMs, etc.)