

Die Systemhaus-Horst Story

4 Monate Programmierarbeit (August - Dezember 2025)

Dokumentiert: 15. Dezember 2025 Autor: Claude (basierend auf Analyse aller Server)

Executive Summary

In 4 Monaten wurde eine komplette Unternehmensinfrastruktur aufgebaut:

Metrik	Wert
Projekte	6 produktionsreife Anwendungen
Server/VMs	9 virtuelle Maschinen
Code-Zeilen	~43.000+ (Python, TypeScript, Svelte)
Geschaetzer Wert	400.000 - 600.000 EUR
Sessions dokumentiert	37+

Teil 1: Die Produkte

1. MIRA - Das Flaggschiff

Server: 192.168.42.15 (SYSTEMHAUS-001) **Status:** Produktiv mit zahlenden Kunden
Geschaetzer Wert: 250.000 - 350.000 EUR

Was ist MIRA?

MIRA (Modern Intelligence & Response Architecture) ist eine vollstaendige Multi-Tenant SaaS-Plattform fuer KI-gestuetzte Assistenzsysteme. Unternehmen koennen eigene KI-Chatbots betreiben, die aus ihren Dokumenten lernen.

Kernfunktionen

Feature	Beschreibung
Multi-Tenant	10+ Companies auf einer Plattform
Brain-System	Vektorbasierte Wissensdatenbank pro User/Company
Embed-Widget	Chat-Widget fuer externe Websites (Drag & Drop)
Voice-System	Wake-Word "Hey MIRA", Spracheingabe, TTS
Telegram Live-Chat	Admin antwortet per Telegram, erscheint im Widget
Partner-Portal	Self-Service Demo-Erstellung mit Provision (MiraCoins)
2FA	TOTP mit Backup-Codes

Technische Details

Code-Zeilen: 23.453 (Python Backend)
~20.000 (Frontend HTML/JS/CSS)
~43.000 Total

API-Endpoints: 214 REST-Routen
Services: 21 Python-Module
Companies: 10+ aktive Mandanten

Tech-Stack

Schicht	Technologie
Backend	Python 3.11, FastAPI, Uvicorn
Frontend	Vanilla JavaScript SPA
KI	Anthropic Claude API, OpenAI API
Vektorschreie	Sentence-Transformers
TTS	ElevenLabs API
STT	Whisper (EVY Voice API)
Wake-Word	Picovoice Porcupine

Aktive Companies

Company	Besonderheit
MIRA	Hauptdemo
DevoraXx	Quellcode-Boerse Integration
Spengel GmbH	Custom Audio-Begruessung
derhorst	Persoenliche Instanz
jascha	Jascha Horst
MIRA PARTNER	Vertriebspartner-Demo
mira_sales	Sales-Team
mira_support	Support-Team
MIFCOM	Externe Firma

Changelog (Auszug)

Version	Datum	Features
2.11.0	12.12.2025	Audio-Begruessungen, Widget-Animationen, Multi-Widget
2.10.0	07.12.2025	Telegram Live-Chat
2.9.0	01.12.2025	Voice System (STT, TTS, Wake-Word)
2.8.0	25.11.2025	Embed Widget v2, Drag & Drop

2. EVY - AI Backend System

Server: 192.168.42.15 (SYSTEMHAUS-001) **Status:** Produktiv

Was ist EVY?

EVY ist das KI-Backend-System, das mehrere spezialisierte Services bereitstellt.

Services

Service	Port	Funktion
evy_backend	8000	Hauptsystem (Chat, Watcher)
evy_voice_api	5070	Voice API (Whisper STT)
evy_train_api	5071	Training API
evy_auth	-	Authentifizierung
ai_watcher	-	KI Watcher
evy_avatar	-	Avatar Builder (OpenAI)
evy_mentorisierer	-	Mentorisierer Watcher
evy_noten	-	Notenerkennung & Audio
evy_filewatcher	-	File Watcher (Email)
evy_mailwatcher	-	Mail Watcher (Reports)

Spezial-Features

- **Voice Clone:** Stimmen klonen mit ML
- **Audiveris Integration:** Notenblatt → MIDI Konvertierung
- **MIDI → WAV:** FluidSynth Integration

3. DevoraXx - Die Quellcode-Boerse

Server: 192.168.42.214 **Status:** Produktiv **URL:** <https://app.devoraxx.de>

Was ist DevoraXx?

Ein Marktplatz fuer den Kauf und Verkauf von Quellcode. Die Plattform bietet sichere Transaktionen und ein integriertes Legal CMS.

Kernfunktionen

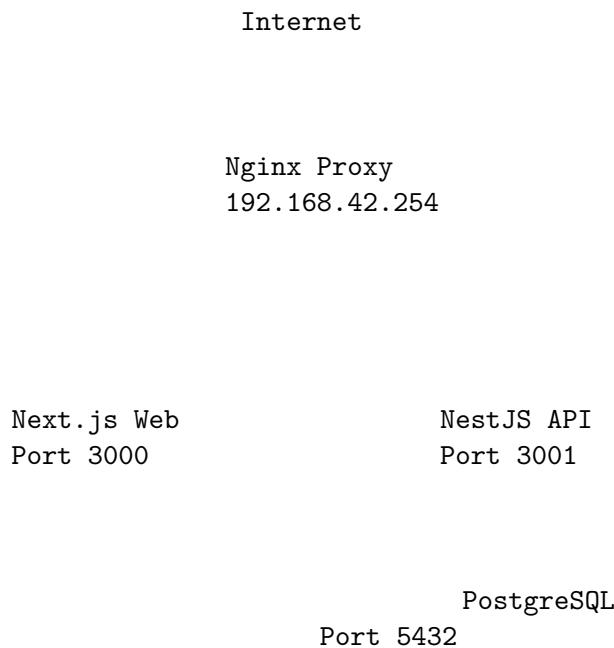
Feature	Beschreibung
Marktplatz	Produkte listen, kaufen, verkaufen
Rollen	Buyer, Seller, Admin, Superadmin
Bewertungen	1-5 Sterne Rating System
Legal CMS	Versionierte Rechtstexte
VM Management	Hyper-V VMs aus dem Browser erstellen!
MIRA Integration	Chat-Widget eingebettet

Feature	Beschreibung

Tech-Stack

Monorepo:	Turborepo + pnpm
Backend:	NestJS (TypeScript)
Frontend:	Next.js 15 (React 19)
Datenbank:	PostgreSQL 16 + Prisma ORM
Styling:	Tailwind CSS v4
Auth:	JWT + Passport + bcrypt
Code-Zeilen:	11.357 TypeScript

Architektur



VM Management Feature

Das beeindruckendste Feature: **Hyper-V VMs direkt aus dem Browser erstellen.**

```

Browser (Superadmin UI)
  ↓
Next.js API Routes (/api/vm/*)
  ↓ SSH (sshpass)
Hyper-V Host DASBIEST (192.168.42.16)
  ↓ PowerShell
Hyper-V VMs
  
```

The flowchart shows the process for creating Hyper-V VMs from a browser. It starts with a 'Browser (Superadmin UI)' which sends requests to a 'Next.js API Routes (/api/vm/*)'. These routes utilize 'SSH (sshpass)' to interact with a 'Hyper-V Host DASBIEST (192.168.42.16)'. From the host, 'PowerShell' commands are executed to manage the 'Hyper-V VMs'.

8-Schritte-Prozess: 1. Ordner erstellen 2. VM erstellen (Gen2, RAM, Disk) 3. CPU konfigurieren 4. Secure Boot deaktivieren 5. Netzwerk hinzufuegen 6. ISO mounten (Preseed) 7.

Boot-Reihenfolge setzen 8. VM starten

4. Admin Portal - Zentrale Verwaltung

Server: 192.168.42.230 **Status:** Produktiv **URL:** <http://192.168.42.230>

Was ist das Admin Portal?

Eine zentrale Verwaltungsoberflaeche fuer alle 9 VMs der Infrastruktur. Echtzeit-Monitoring, Service-Steuerung, und VM-Management.

Features

Feature	Beschreibung
Dashboard	Alle Server mit Live-Status
Maschinen-Details	RAM, Disk, CPU, Uptime
Service-Steuerung	Start/Stop/Restart von Services
Docker-Management	Container steuern
Log-Viewer	Echtzeit-Logs per SSH
Cronjob-Verwaltung	Crontabs bearbeiten
VM-Management	Hyper-V VMs erstellen/loeschen
User-Verwaltung	Rollen: Superadmin/Admin/User

Tech-Stack

Frontend:	SvelteKit
Backend:	FastAPI (Python)
Datenbank:	PostgreSQL 16
Container:	Docker Compose
Code-Zeilen:	~4.000+

Modul-System

Modul	Slug	Verfuegbar fuer
System-Health	health	alle
Dienste	services	alle
VM-Verwaltung	vm-management	Hyper-V Hosts
Backup-Config	backup-config	Hyper-V Hosts
Cronjobs	cronjobs	Linux VMs
Docker	docker	Linux VMs
Webserver	webserver	Linux VMs
Log-Viewer	logs	alle

Verwaltete Maschinen

ID	Name	IP	Typ	Host
1	DASBIEST	.16	HOST	-
8	KleinerHund	.231	HOST	-
2	Admin-Server	.230	VM	DASBIEST
4	devoraxx	.214	VM	DASBIEST
5	SYSTEMHAUS-001	.15	VM	DASBIEST
6	SYSTEMHAUS-004	.205	VM	DASBIEST
3	Webserver	.13	VM	kleinerHund
7	openhab	.10	VM	kleinerHund
9	Nextcloud	.12	VM	kleinerHund

5. Office Server - Claude-basiertes Interface

Server: 192.168.42.253 **Status:** Produktiv (Session 7) **URL:** <https://alexa.mukupi.art>

Was ist der Office Server?

Eine Web-Anwendung mit Claude-Integration fuer Dateiverwaltung, Medienverarbeitung und Sprach-Interaktion. Der persoenliche digitale Assistent.

Kernfunktionen

Feature	Beschreibung
Claude Terminal	Natuerliche Sprache → Aktionen
Spracheingabe	Whisper STT (Mikrofon-Button)
Sprachausgabe	OpenAI TTS (Onyx-Stimme)
Session-Management	Kontext bleibt erhalten
Dateiexplorer	Mac-Mount durchsuchen
Multiviewer	Bilder, Video, Audio, PDF, Text
Email-Client	iCloud Integration
Smart Home	OpenHAB Steuerung
Alexa Skill	“mein office”
Drucker	CUPS Integration

Claude-Tags

[NAV:/pfad]	→ Explorer navigieren
[OPEN:/pfad/datei]	→ Datei oeffnen
[MAILREAD:inbox]	→ Posteingang zeigen
[MAILSEARCH:query]	→ Emails suchen
[MAIL:to:subject:body]	→ Email senden
[OPENHAB:Lampe:ON]	→ Lampe einschalten
[PRINT:file:/pfad]	→ Datei drucken

Tech-Stack

Frontend:	SvelteKit + TailwindCSS
Theme:	Rot/Dunkel
Backend:	Node.js + Express
AI:	Claude Code CLI
STT:	OpenAI Whisper API
TTS:	OpenAI TTS API
Code-Zeilen:	3.876

Smart Home Geraete

- **~40 Lampen:** Wohnzimmer, Kueche, Buero, etc.
 - **17 Jalousien:** Inkl. Markisen
 - **9 Temperatursensoren:** Pro Raum
-

6. Proxy Portal - Reverse Proxy Verwaltung

Server: 192.168.42.254 (dieser Server) **Status:** Produktiv **URL:** <http://192.168.42.254>

Was ist das Proxy Portal?

Web-basierte Verwaltung des zentralen Nginx Reverse Proxys. Alle externen Domains laufen hierueber.

Features

Feature	Beschreibung
Domain-Verwaltung	Hinzufuegen, bearbeiten, loeschen
SSL-Zertifikate	Let's Encrypt automatisch
Upstream-Server	Backend-Konfiguration
Health-Checks	Server-Verfuegbarkeit
Config-Generator	Nginx-Configs automatisch
Integriertes Handbuch	Dokumentation im Portal

Tech-Stack

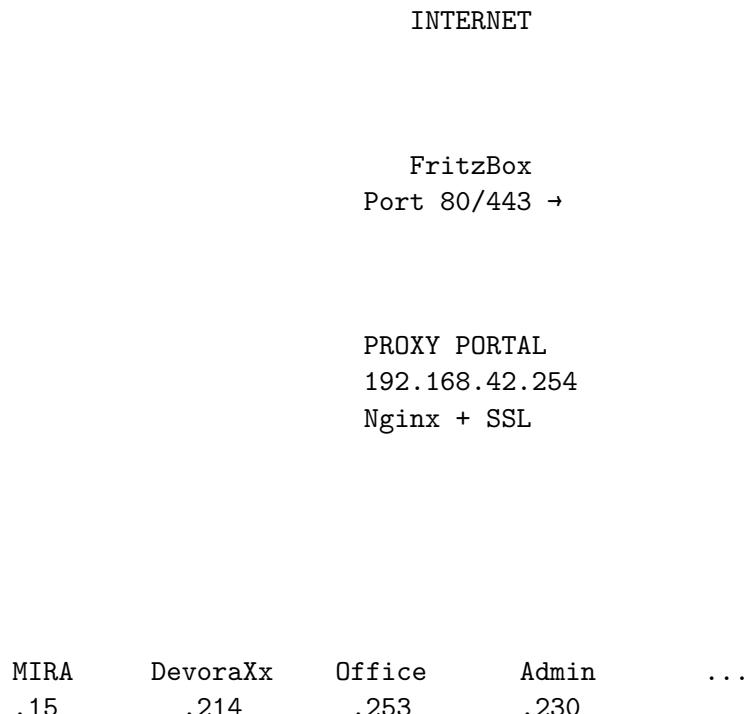
Frontend:	SvelteKit 5 + Vite
Backend:	FastAPI (Python)
Datenbank:	PostgreSQL 16
Webserver:	Nginx
Container:	Docker Compose
Code-Zeilen:	4.140

Verwaltete Domains

Domain	Backend	SSL
mukupi.art	.15 (MIRA)	Let's Encrypt
systemhaus-horst.de	.15 (MIRA)	Let's Encrypt
app.devoraxx.de	.214 (Devoraxx)	Let's Encrypt
alexa.mukupi.art	.253 (Office)	Let's Encrypt
buendnis-leben.de	Static	Let's Encrypt
phil-pohlmeier.art	Static	Let's Encrypt
poimer.art	Static	Let's Encrypt
the-style-of-power.de	Static	Let's Encrypt

Teil 2: Die Infrastruktur

Netzwerk-Topologie



Hyper-V Hosts

DASBIEST (192.168.42.16)

Eigenschaft	Wert
OS	Windows 11
Rolle	Primaerer Hyper-V Host
VMs	6

VMs auf DASBIEST: 1. proxy-portal (.254) - Reverse Proxy 2. Admin-Server (.230) - Admin Portal 3. SYSTEMHAUS-001 (.15) - MIRA/EVY 4. devoraxx (.214) - Quellcode-Boerse 5. dns-portal (.216) - DNS/DHCP (FreeBSD) 6. Office Server (.253) - Claude Interface

kleinerHund (192.168.42.231)

Eigenschaft	Wert
OS	Windows
Rolle	Sekundaerer Hyper-V Host
VMs	3

VMs auf kleinerHund: 1. Webserver ALT (.13) - Apache (abgeloest) 2. openhab (.10) - Smart Home 3. Nextcloud (.12) - Cloud Storage

SSH-Zugriff

Alle Linux-VMs sind per SSH erreichbar:

```
# Standard: Port 2222, Key-Auth
ssh -p 2222 dieterhorst@192.168.42.15      # MIRA/EVY
ssh -p 2222 dieterhorst@192.168.42.214     # DevoraXx
ssh -p 2222 dieterhorst@192.168.42.230     # Admin Portal
ssh -p 2222 dieterhorst@192.168.42.253     # Office Server
ssh -p 2222 dieterhorst@192.168.42.254     # Proxy Portal

# Hyper-V Hosts: Port 22
ssh dieterhorst@192.168.42.16              # DASBIEST
ssh dieterhorst@192.168.42.231             # kleinerHund
```

Teil 3: Die Zahlen

Code-Statistiken

Projekt	Sprache	Zeilen
MIRA Backend	Python	23.453
MIRA Frontend	HTML/JS/CSS	~20.000
DevoraXx	TypeScript	11.357
Admin Portal	Python + Svelte	~4.000
Office Server	Svelte + TS	3.876
Proxy Portal	Python + Svelte	4.140
Gesamt		~67.000

API-Statistiken

Projekt	Endpoints
MIRA	214
DevoraXx	~50
Admin Portal	~30
Proxy Portal	~20
Gesamt	~314

Wertermittlung (MIRA)

Die professionelle Quellcode-Bewertung von MIRA ergab:

Methode	Wertspanne
Wiederbeschaffungswert	195.600 - 293.400 EUR
COCOMO II	640.000 - 1.150.000 EUR
Function-Point-Analyse	300.000 - 570.000 EUR
Ertragswert (DCF)	250.000 - 350.000 EUR

Empfohlener fairer Verkehrswert MIRA: 280.000 - 380.000 EUR

Geschaetzter Gesamtwert aller Projekte: 400.000 - 600.000 EUR

Teil 4: Die Timeline

Phase 1: Grundlagen (August-September 2025)

- Server-Infrastruktur aufgebaut
- Hyper-V VMs eingerichtet
- SSH-Zugang konfiguriert
- Erste MIRA-Version

Phase 2: MIRA Entwicklung (September-Oktober 2025)

- Multi-Tenant-Architektur
- Brain-System (Vektorschreibe)
- Embed-Widget v1
- User-Authentifizierung + 2FA

Phase 3: Erweiterungen (Oktober-November 2025)

- DevoraXx Marktspiel gestartet
- Admin Portal aufgebaut
- EVY Voice System
- MIRA Widget v2 mit Drag & Drop

Phase 4: Integration (November-Dezember 2025)

- Voice System (Wake-Word, TTS, STT)
- Telegram Live-Chat
- Partner-Portal mit MiraCoins
- Office Server mit Claude
- Proxy Portal (Abloesung alter Webserver)

Phase 5: Aktuell (Dezember 2025)

- Audio-Begrüßungen fuer Widgets
 - Widget-Animationen
 - Multi-Widget Support
 - Alexa Skill Integration
 - Smart Home (OpenHAB) Integration
 - Diese Dokumentation
-

Teil 5: Technologie-Ueberblick

Verwendete Frameworks

Kategorie	Technologien
Backend	FastAPI, NestJS, Express
Frontend	SvelteKit 5, Next.js 15, Vanilla JS
Datenbank	PostgreSQL 16, Prisma ORM, SQLAlchemy
Container	Docker, Docker Compose
KI	Claude API, OpenAI (GPT, Whisper, TTS)
Voice	ElevenLabs, Picovoice Porcupine
Webserver	Nginx, Apache
SSL	Let's Encrypt, Certbot

Kategorie	Technologien
Virtualisierung	Hyper-V
Versionierung	Git

Externe Integrationen

Service	Verwendung
Anthropic (Claude)	KI-Chat, Code-Assistenz
OpenAI	GPT, Whisper STT, TTS
ElevenLabs	Text-to-Speech
Picovoice	Wake-Word Detection
Telegram	Live-Chat Bot
Let's Encrypt	SSL-Zertifikate
iCloud	Email-Integration
OpenHAB	Smart Home

Teil 6: Lessons Learned

Was funktioniert gut

1. **Multi-Tenant von Anfang an** - Skalierbarkeit eingebaut
2. **Docker fuer alles** - Reproduzierbare Deployments
3. **SSH-Automatisierung** - Verwaltung ueber alle Server
4. **Dokumentation mitfuehren** - Handbuecher fuer alles
5. **SvelteKit als Frontend** - Schnell, modern, produktiv

Herausforderungen

1. **Monolithisches MIRA-Frontend** - 14.000 Zeilen in einer Datei
2. **Fehlende Tests** - Kein automatisiertes Testing
3. **Single-Developer** - Wissensabhaengigkeit
4. **API-Kosten** - Claude, OpenAI, ElevenLabs summieren sich

Naechste Schritte (Empfehlungen)

- Unit-Tests implementieren
- Frontend-Refactoring (Komponenten)
- CI/CD-Pipeline einrichten
- Monitoring/Alerting aufbauen
- Backup-Strategie formalisieren

Anhang

A: Verzeichnisstruktur

```
/opt/
  Claude/          # Selbsterhaltung & Dokumentation
    01_START/      # Startup-Dateien
    02_PROJEKT/    # Diese Dokumentation
  proxy-portal/    # Proxy Portal Projekt
    MIRA/          # MIRA (auf .15)
    EVY/           # EVY (auf .15)
  devoraxx/        # DevoraXx (auf .214)
  admin-portal/   # Admin Portal (auf .230)
  office/         # Office Server (auf .253)
```

B: Wichtige URLs

URL	Beschreibung
http://192.168.42.254	Proxy Portal
http://192.168.42.230	Admin Portal
https://systemhaus-horst.de/MIRA	MIRA Login
https://app.devoraxx.de	DevoraXx
https://alexa.mukupi.art	Office Server

C: Dokumentations-Dateien

Pfad	Beschreibung
/opt/proxy-portal/docs/HANDBUCH.md	Proxy Portal Handbuch
/opt/MIRA/docs/Admin-Handbuch.md	MIRA Admin-Handbuch
/opt/MIRA/docs/Benutzerhandbuch.md	MIRA Benutzerhandbuch
/opt/MIRA/docs/Vertriebspartner-Handbuch.md	Partner-Handbuch
/opt/devoraxx/docs/ARCHITEKTUR.md	DevoraXx Architektur
/opt/admin-portal/docs/Handbuecher/	Admin Portal Docs
/opt/office/docs/HANDBUCH.md	Office Server Handbuch

Ende der Dokumentation

Erstellt: 15. Dezember 2025 Von: Claude auf proxy-portal (192.168.42.254)