

Prompt:

Hier ist eine leicht gestraffte und weiter präzierte Version, die du so verwenden kannst:

Du bist **Azure Cloud Senior Architect** mit Schwerpunkt **Security, Compliance und KI-Plattformen** in regulierten Umgebungen (Finanz, Kritische Infrastrukturen, Behörden).

1. Auftrag

Erstelle einen **detaillierten Architekturvorschlag** für den **Betrieb von KI-Workloads auf Microsoft Azure**, der **konform** ist zu:

- **EU AI Act** (insb. Risikoklassen, Technical & Organizational Measures, Logging, Human Oversight)
- **DSGVO (GDPR)**
- **DORA**
- **NIS 2**
- **ISO 27001**

Die Architektur soll sowohl **sicherheits-** als auch **compliance-orientiert** sein und klar zeigen, wie regulatorische Anforderungen **technisch umgesetzt** werden.

Treffe, falls nötig, eigene **Annahmen** (z. B. Branche, Datenklassifizierung, Kritikalität) und liste sie zu Beginn explizit auf.

2. Rahmenbedingungen & Zielbild

Die Lösung basiert **vollständig auf Microsoft Azure** (Azure-native oder Azure-verwaltete Services) und stellt den Betrieb in **EU-Regionen** sicher (z. B. *West Europe / North Europe*) zur Erfüllung von **Data Residency**-Anforderungen.

Folgende technische Komponenten sind zu berücksichtigen und jeweils konkret mit Azure-Services zu hinterlegen:

2.1 KI- und Compute-Layer

- Kubernetes-Cluster mit KI-Workern und Container-/Cluster-Management
- Agentic AI mit LLM-Adapter (z. B. orchestrierte KI-Agents)
- Build, Test und Deploy von AI Agents (MLOps / AIOps)
- LLM-Router (Routing zwischen verschiedenen Modellen / Providern, inkl. Azure OpenAI und ggf. Self-Hosted Models)

2.2 Security & Governance

- Web Application Firewall (WAF)
- Prompt Firewall / Prompt Security
- AI-Model Scanning (Model Security, Supply Chain Security)
- AI-Interface Scanning (Security von APIs / Endpoints / UIs)
- Vulnerability Management
- IAM (RBAC, Anbindung an LDAP/Azure Entra ID)
- PAM (Privileged Access Management)
- Bastion Host
- Öffentliche und private Zertifikate inkl. Lifecycle- und Secrets-Management

2.3 Networking & Ingress

- Load Balancer mit TLS-Terminierung
- API-Gateway
- Netzwerksegmentierung, Private Endpoints, ggf. Hub-Spoke-Architektur und Zero-Trust-Prinzipien

2.4 Daten- & Storage-Layer

- Datenbank-Layer mit:
 - SQL-Datenbanken
 - NoSQL-Datenbanken
 - Object Stores (S3-ähnlich / Blob Storage)
 - Vektor-Datenbank
 - Redis Cache
 - GraphDB
- Big Data / Data Lake / Data Warehouse
- Real-Time Analytics
- Apache Kafka Router (Event Streaming)
- Queues / Message Queues (MQ)

2.5 Frontend & Applikations-Layer

- Frontend als Container-Deployment
- Git-Repositories, Runner, CI/CD-Pipelines und IDE-Integration

2.6 Observability & Operations

- Logging (zentrale, revisionssichere Protokollierung)

- Monitoring
- Metriken (z. B. Visualisierung über Grafana)
- SIEM/SOC-Integration (z. B. Splunk, Azure-native Alternativen)

2.7 Infrastructure as Code

- Deployment mit Terraform (inkl. State-Handling, Workspaces, Pipelines)
-

3. Konkrete Anforderungen an deine Antwort

3.1 Azure-Service-Mapping (sehr detailliert)

- Nenne **für jede** der oben genannten Komponenten die **konkreten Azure-Services**, z. B.:
 - „Azure Kubernetes Service (AKS) für ...“
 - „Azure API Management für ...“
 - „Azure Front Door / Azure Application Gateway mit WAF für ...“
 - „Azure Key Vault für Zertifikats- und Secrets-Management“
 - „Azure Monitor / Log Analytics / Application Insights für ...“
 - usw.
- Erkläre **kurz den Zweck** jedes Services in dieser Architektur und **warum** er für den jeweiligen Baustein geeignet ist.
- Wenn es **mehrere Optionen** gibt (z. B. Front Door vs. Application Gateway), begründe die **präferierte Wahl**.

Nutze nach Möglichkeit eine **Tabelle „Azure-Service vs. Zweck“**.

3.2 Compliance-Mapping

Erstelle eine **Tabelle**, die zeigt, welche Architekturbausteine (Services / Patterns) **welche Anforderungen** aus

- EU AI Act
- DSGVO
- DORA
- NIS 2
- ISO 27001

adressieren, z. B.:

- Logging & Audit
- Data Residency & Data Protection

- Verschlüsselung at rest / in transit
 - Zugriffskontrolle (RBAC, PAM, Least Privilege)
 - Risikomanagement & Schwachstellenmanagement
 - Monitoring, Incident Detection & Incident Response
 - Business Continuity / Resilienz / Verfügbarkeit
-

3.3 Architekturübersicht

- Beschreibe zunächst eine **High-Level-Architektur** (Schichtenmodell), z. B.:
User → Edge/WAF → API Gateway → LLM Router / AI Agents (auf AKS) → Data Layer → Observability / Security Layer
 - Ergänze ein **textuelles Architekturdiagramm** (z. B. in **Mermaid** oder alternativ als klar strukturierte hierarchische Liste), das die wichtigsten Komponenten und ihre Beziehungen zeigt.
-

3.4 Security- & Data-Protection-Aspekte

Gehe explizit auf **Datenflüsse** ein (Eingabe-Prompts, Kontextdaten, Model-Outputs, Logs) und erkläre, wie:

- Daten **verschlüsselt** werden (at rest / in transit) inkl. Nutzung von Managed Identities, TLS und Key Vault
 - **Zugriffe gesteuert** werden (RBAC, Azure Entra ID, PAM, Just-in-Time Access / JIT-Access)
 - **Protokollierung, Auditability und Nachvollziehbarkeit** umgesetzt werden (z. B. Audit-Logs, Access-Logs, Model-Usage-Logs)
 - **Data Residency** (EU-Regionen, ggf. strikte Datenlokation) sichergestellt wird
 - **Modell- und Prompt-Risiken** (Prompt Injection, Data Leakage, Model Abuse) mitigiert werden (z. B. Prompt Firewall, Content Filter, Policy Enforcement)
-

3.5 Betrieb & Lifecycle

Erläutere, wie:

- neue KI-Modelle / Agents **versioniert, getestet und ausgerollt** werden
 - CI/CD-Pipelines (z. B. GitHub Actions / Azure DevOps)
 - Blue-Green- / Canary-Deployment auf AKS
 - Rollback-Strategien
- **Monitoring, Alerting und Incident Response** organisiert sind

- Metriken, Logs, Traces
 - Alerts (z. B. via Azure Monitor)
 - Integration in SOC / SIEM (Splunk oder Microsoft Sentinel)
 - **Schwachstellenmanagement** (Vulnerability Management) in Container-Images, Bibliotheken und Modellen integriert wird
 - Image-Scanning, Dependency-Scanning
 - regelmäßige Security-Updates / Patch-Mechanismen
-

4. Ausgabeformat

Strukturiere deine Antwort mit folgenden Hauptüberschriften:

1. Überblick & Annahmen
2. Zielarchitektur (High Level)
3. Azure-Service-Mapping pro Baustein
4. Compliance-Mapping (Tabelle)
5. Security & Data Protection
6. Betriebs- & Lifecycle-Konzept

Verwende mindestens diese **Tabellen**:

- **Tabelle 1:** Azure-Komponenten vs. Zweck / Layer
- **Tabelle 2:** Regulatorische Anforderung vs. Architekturbaustein (Service / Pattern)

Antworte auf **Deutsch**, **präzise und technisch**, mit **konkreten Azure-Service-Namen** und **technischen Details** statt allgemeiner Beschreibungen.

Antwort:

Alles klar – ich führe den Prompt jetzt aus und entwerfe dir die Architektur.

1. Überblick & Annahmen

Annahmen (konkretisiert):

- Branche: reguliertes Finanzinstitut oder KRITIS-Betreiber in der EU.
- Daten: überwiegend **personenbezogene Daten** (inkl. special categories möglich), daher **DSGVO hoch relevant**.

- KI-System eurer Lösung gilt als „**High-Risk AI System**“ nach EU AI Act (z. B. Kreditentscheidungen, Risiko-Scoring).([Künstliche Intelligenz Gesetz EU](#))
 - Alle produktiven Workloads laufen in **EU-Azure-Regionen** (z. B. *West Europe, North Europe*).
 - Nutzung von **Azure OpenAI (EU Data Zone / EU-Residency-Option)** bzw. ausschließlich EU-datenresidente Endpunkte für Foundation Models.([Microsoft Learn](#))
 - Ziel: Architektur soll **technisch belegbar** Anforderungen aus
 - EU AI Act (insb. Logging, Risikomanagement, Human Oversight)([digital-strategy.ec.europa.eu](#))
 - DSGVO,
 - DORA, NIS 2 (Resilienz, Monitoring, Incident Handling),
 - ISO 27001 (ISMS-Kontrollen) adressieren.
-

2. Zielarchitektur (High Level)

2.1 Schichtenmodell (verbale Übersicht)

1. User & Clients

- interne Nutzer (Mitarbeitende), ggf. externe Kunden
- Zugriff über Browser / interne Portale / APIs

2. Edge / Ingress / Security

- **Azure Front Door mit WAF** (global, optional)
- **Azure Application Gateway mit WAF** (regional, TLS-Terminierung)([Microsoft Learn](#))
- DDoS Protection, Azure Firewall, Private Endpoints

3. API & Orchestration Layer

- **Azure API Management (APIM)** als API-Gateway
- „Prompt Firewall“ & Policies in APIM + Azure AI Content Safety
- LLM-Router als Microservice (AKS) hinter APIM

4. KI-/Compute-Layer

- **AKS-Cluster** mit GPU-Nodepools für KI-Workloads
- Agentic AI-Orchestrator (z. B. LangChain-/Semantic-Kernel-basiert) als Services auf AKS
- Azure OpenAI / Azure AI Foundry / Self-hosted Models auf AKS

5. Data & Event Layer

- Azure SQL, Cosmos DB, Blob Storage (Data Lake), Azure AI Search (Vektor)
- Event Hubs (Kafka-kompatibel), Service Bus, Stream Analytics / Data Explorer

6. Observability & Security-Layer

- Azure Monitor, Log Analytics, Application Insights
- Azure Managed Grafana für Metrik-Dashboards([Microsoft Learn](#))
- Microsoft Sentinel + Splunk-Anbindung als SIEM/SOC
- Defender for Cloud (inkl. Defender for Containers, Defender for APIs)

7. DevOps / MLOps / IaC

- Git-Repos (GitHub Enterprise oder Azure DevOps), CI/CD-Pipelines
- Terraform für IaC (State in verschlüsselltem Azure Storage)
- Azure ML / AI Studio / AI Foundry für MLOps (je nach Stack)

2.2 Textuelles Architekturdiagramm (Mermaid)

```
graph LR
    subgraph Clients
        U[User / Apps]
    end

    subgraph Edge
        AFD[Azure Front Door + WAF]
        AGW[Application Gateway WAF + TLS]
        AFW[Azure Firewall]
    end

    subgraph API
        APIM[Azure API Management\n(API Gateway + Policies)]
        PF[Prompt Firewall\n(Content Safety + APIM Policies)]
    end

    subgraph AI_Compute [AI & Compute Layer (AKS)]
        AKS[AKS Cluster\nGPU/CPU Node Pools]
        LLMR[LLM Router Service]
        AGENTS[Agentic AI Orchestrator\n(AI Agents)]
        FE[Frontend Container]
    end

    subgraph AI_Services
        AOAI[Azure OpenAI / Azure AI Foundry\n(EU Data Residency)]
        MODELS[Self-hosted Models on AKS]
    end

    subgraph Data [Data & Storage Layer]
        SQL[Azure SQL DB / MI]
        COSMOS[Azure Cosmos DB\n(NoSQL + Graph)]
        BLOB[Azure Data Lake / Blob Storage]
        VECTOR[Azure AI Search\n(Vektorindex)]
        REDIS[Azure Cache for Redis]
        KUSTO[Azure Data Explorer / Synapse]
    end
```

```

end

subgraph Events
    EH[Azure Event Hubs (Kafka Endpoint)]
    SB[Azure Service Bus (Queues/Topics)]
    SA[Azure Stream Analytics]
end

subgraph Observability[Observability & Security]
    AM[Azure Monitor + Log Analytics]
    APPINS[Application Insights]
    GRAF[Azure Managed Grafana]
    SENT[Microsoft Sentinel]
    SPL[SPLUNK (VM/AKS)]
    DEF[Microsoft Defender for Cloud]
end

subgraph Identity_Sec[Identity & Access]
    ENTRA[Azure Entra ID\n(RBAC, Conditional Access)]
    PIM[Entra Privileged Identity Mgmt]
    KV[Azure Key Vault + Managed HSM]
    BAST[Azure Bastion]
end

subgraph DevOps[DevOps / MLOps / IaC]
    GIT[GitHub / Azure DevOps Repos]
    PIPE[CI/CD Pipelines\n(GitHub Actions / Azure Pipelines)]
    TF[Terraform\n(State in Storage)]
end

U --> AFD --> AGW --> APIM --> PF --> LLMR --> AGENTS
AGENTS --> AOAI
AGENTS --> MODELS
LLMR --> AOAI
LLMR --> MODELS

AGENTS --> SQL
AGENTS --> COSMOS
AGENTS --> BLOB
AGENTS --> VECTOR
AGENTS --> REDIS

AGENTS --> EH
EH --> SA --> KUSTO
AGENTS --> SB

AKS --> AM
AKS --> APPINS
AM --> GRAF
AM --> SENT
AM --> SPL
DEF --> SENT

ENTRA --> APIM
ENTRA --> AKS
ENTRA --> GIT
PIM --> ENTRA
KV --> AGW
KV --> APIM
KV --> AKS
BAST --> AKS

GIT --> PIPE --> AKS
PIPE --> TF

```

TF --> AKS
TF --> Data
TF --> Networking((VNet/NSG/PE))

3. Azure-Service-Mapping pro Baustein

3.1 KI- und Compute-Layer

Baustein	Azure-Service(s)	Zweck / Begründung
Kubernetes-Cluster mit KI-Workern	Azure Kubernetes Service (AKS)	Managed Kubernetes mit Integration in Azure RBAC, Managed Identities, Auto-Scaling, GPU-Nodepools für KI/LLM-Workloads.
Container-/Cluster-Management	AKS + Azure Monitor for Containers, Azure Policy	Zentrales Management, Configuration Enforcement (PodSecurity, Gatekeeper), Ressourcenkontrolle, Compliance.
Agentic AI mit LLM-Adapter	Microservices auf AKS (z. B. .NET/Python), Azure OpenAI / Azure AI Foundry	Orchestrierung von Agents, Aufruf verschiedener Modelle (Azure OpenAI, lokale Modelle). EU-Residency-Optionen helfen bei DSGVO/AI-Act-Konformität. (Microsoft Learn)
Build, Test, Deploy von AI Agents	GitHub Enterprise / Azure DevOps Repos, GitHub Actions / Azure Pipelines, AKS , optional Azure Machine Learning (MLOps)	End-to-End-CI/CD-Pipelines inkl. Build, Tests (unit, integration, eval), Deployment auf AKS; ML-spezifische Pipelines mit Modellversionierung.
LLM-Router	Custom Service auf AKS + Azure API Management (Routing/Policies)	Policy-gesteuertes Routing zwischen Modellen (Azure OpenAI, Self-hosted, ggf. Drittanbieter), A/B-Tests, Fallback-Strategien.

3.2 Security & Governance

Baustein	Azure-Service(s)	Zweck / Begründung
WAF	Azure Front Door WAF und/oder Application Gateway WAF	Zentrale L7-Firewall gegen OWASP-Top-10, Bot-Protection, Geo-Blocking; unterstützt Compliance-Anforderungen. (Microsoft Learn)
Prompt Firewall / Prompt Security	Azure API Management Policies + Azure AI Content Safety + Azure OpenAI Safety Filters	Vor- und Nachschaltung von Content-Filtern, Blockieren schädlicher Prompts/Outputs; wichtig für AI-Act-Risikomitigation und Grundrechtsschutz. (Microsoft Learn)
AI-Model Scanning	Microsoft Defender for Cloud (Container/VM/Registry-Scanning), Azure ML Responsible AI-Tools	Scans von Container-Images/Abhängigkeiten, Modell-Governance (Bias, Fairness, Explainability) als Beitrag zu EU-AI-Act-Risikomanagement. (21medien.de)
AI-Interface Scanning	Microsoft Defender for APIs , WAF-Logs, APIM	Schutz der API-Interfaces gegen Missbrauch, Anomalie-Erkennung, Angriffserkennung auf AI-

Baustein	Azure-Service(s)	Zweck / Begründung
	Diagnostics	Endpunkte.
Vulnerability Management	Microsoft Defender for Cloud , Defender for Containers, Qualys-Integration	Kontinuierliches Schwachstellenmanagement auf VMs, Containern, PaaS-Ressourcen – relevant für DORA/NIS2/ISO 27001.
IAM (RBAC, LDAP)	Azure Entra ID , Azure RBAC, Azure AD Connect / Entra Connect	Zentrale Identitäten, Rollen, Conditional Access; Integration mit On-Prem-LDAP/AD; Grundlage für least privilege & Zero Trust.
PAM	Azure Entra Privileged Identity Management (PIM)	Zeitlich begrenzte, genehmigungspflichtige Admin-Rollen, Governance & Justification-Logging – wichtig für ISO 27001, DORA.
Bastion Host	Azure Bastion	Sichere, browserbasierte RDP/SSH-Verbindungen ohne Public IPs auf VMs (z. B. Admin-Knoten, Jump Hosts).
Zertifikate & Secrets	Azure Key Vault (Secrets, Keys, Certificates) + ggf. Managed HSM	Zentrales Management von TLS-Zertifikaten, Schlüsseln (inkl. HSM-backed), Secret Rotation; Integrationen mit App Gateway, AKS, APIM, etc.

3.3 Networking & Ingress

Baustein	Azure-Service(s)	Zweck / Begründung
Load Balancer mit TLS-Terminierung	Application Gateway (L7, TLS-Offload) + interne Azure Load Balancer	TLS-Termination, SNI, Path-based Routing; interne Lastverteilung zwischen AKS-Knoten/Pods.
API-Gateway	Azure API Management	Throttling, AuthN/AuthZ (JWT/OAuth2/Entra), Versionierung, Request/Response-Transformation, zentraler Audit-Log-Punkt.
Netzwerksegmentierung	Azure Virtual Network, NSG, Azure Firewall, Private Endpoints, Hub-Spoke-Architektur	Trennung nach Zonen (Edge, App, Data, Admin), erzwungener Zugriff über Firewall, private Kommunikation zu PaaS-Services.

3.4 Daten- & Storage-Layer

Baustein	Azure-Service(s)	Zweck / Begründung
SQL-Datenbanken	Azure SQL Database / Azure SQL Managed Instance	Transaktionale Workloads, Konfigurationsdaten, Audit-Tables, referentielle Integrität.
NoSQL	Azure Cosmos DB (Core API)	Hochskalierbare, globale Dokument-/Key-Value-DB für Sessions, Konfiguration, Agent-States.
Object Store	Azure Blob Storage / Data Lake Storage Gen2	S3-ähnliche Object-Speicherung für Dokumente, Logs, Model-Checkpoints, Trainingsdaten.
Vektor-Datenbank	Azure AI Search mit Vektorindex	Verwaltete Vektorsuche für RAG, semantische Suche, Retrieval-

Baustein	Azure-Service(s)	Zweck / Begründung
Redis Cache	Azure Cache for Redis	Orchestrierung. In-Memory-Cache für Sessions, Prompt-Kontext, Throttling, Rate-Limiting-Counter.
GraphDB	Cosmos DB (Gremlin API) oder Azure SQL Graph	Modellierung von Beziehungsnetzen (Entitäten, Events, Policies).
Big Data / Data Lake / DWH	Data Lake Storage Gen2, Azure Synapse Analytics oder Fabric	Historisierung, Advanced Analytics, Reporting, KI-Trainingsdaten.
Real-Time Analytics	Azure Event Hubs, Stream Analytics, Azure Data Explorer / Synapse Data Explorer	near real-time Monitoring, Fraud-Detection, Telemetrie-Auswertung.
Apache Kafka Router	Azure Event Hubs (Kafka-kompatibler Endpoint)	Kafka-API-kompatibles Event-Streaming ohne eigenen Kafka-Cluster.
Queues / MQ	Azure Service Bus (Queues/Topics)	Zuverlässige, transaktionale Nachrichten zwischen Services, Entkopplung.

3.5 Frontend & Applikation

Baustein	Azure-Service(s)	Zweck / Begründung
Frontend als Container	Container auf AKS (separate Namespace) oder Azure App Service for Containers	Isoliertes Deployment der UI, Integration mit WAF/APIM, Identity via Entra ID (OIDC).
Git-Repos, Runner, CI/CD	GitHub Enterprise oder Azure DevOps (Repos, Pipelines), self-hosted Runner auf AKS/VM	Vollständige Code-Versionierung, Build/Deploy-Pipelines, Policy-Gates (Code-Scanning, Secrets-Scanning).
IDE-Integration	GitHub Codespaces / Dev Box , VS Code Remote	Sichere, cloudbasierte Dev- und Debug-Umgebungen mit Policy-Control.

3.6 Observability & Operations

Baustein	Azure-Service(s)	Zweck / Begründung
Logging	Azure Monitor Logs + Log Analytics , Diagnostics Settings, Export zu Blob/Event Hub	Zentrale Sammlung von Infrastruktur-, App-, API- und Sicherheits-Logs; Grundlage für EU-AI-Act-Logging-Pflichten und ISO 27001-Kontrollen. (iks.fraunhofer.de)
Monitoring	Azure Monitor, Application Insights , Container Insights	Verfügbarkeit, Performance, End-to-End-Traceability.
Metriken / Grafana	Azure Managed Grafana	Visualisierung von Kennzahlen (Latenz, Request-Fehler, Modell-Auslastung, GPU-Nutzung) – vollständig verwalteter Dienst.(Microsoft Learn)
SIEM / SOC	Microsoft Sentinel + Anbindung an Splunk	Korrelation sicherheitsrelevanter Events, Detection-Regeln, Playbooks; Integration in bestehende SOC-Landschaft.

3.7 Infrastructure as Code (IaC)

Baustein	Azure-Service(s)	Zweck / Begründung
Terraform Deployment	Terraform OSS/Cloud + Backend in Azure Storage (verschlüsselt, private Endpoint)	Reproduzierbare Provisionierung von Netz, AKS, Datenbanken, Monitor, Sicherheitskomponenten.
Secrets für IaC	Azure Key Vault	Sichere Ablage von Service Principals, SSH-Keys etc.; Zugriff per Managed Identity der Pipelines.

4. Compliance-Mapping (Tabelle – Auszug)

Hinweis: Stark vereinfacht; in einem echten Projekt würdest du pro Regulation eine viel granularere Kontrollliste pflegen.

Regulierung / Anforderung	Architekturbaustein(e)	Technischer Beitrag
EU AI Act – Logging & Traceability (Art. 12, 26) (Künstliche Intelligenz Gesetz EU)	Azure Monitor, Log Analytics, APIM-Logs, AKS Audit Logs, Data Lake für Log-Archiv	Vollständige Protokollierung von Input/Output, Modellaufrufen, Entscheidungen; Aufbewahrung > 6 Monate; revisionssichere Ablage.
EU AI Act – Human Oversight (Art. 14) (Künstliche Intelligenz Gesetz EU)	Frontend Controls, Agent-Orchestrator, RBAC, Workflow-Engine (z. B. Logic Apps)	Manuelle Freigaben, Vier-Augen-Prinzip, Override- und Stop-Funktionen für KI-Entscheidungen.
DSGVO – Datenminimierung & Zweckbindung	Datenmodellierung in SQL/Cosmos, RAG-Architektur mit selektivem Kontext, APIM-Policies	Nur notwendige Attribute im Prompt-Kontext; strenge Scoping-Policies; Pseudonymisierung vor KI-Aufruf.
DSGVO – Datenübermittlung & Residency	Regionenauswahl EU (West/North Europe), Azure OpenAI EU Data-Zones	Sicherstellung, dass personenbezogene Daten die EU nicht verlassen; vertragliche & technische Kontrolle des Modellstandorts. (Microsoft Learn)
DSGVO – Sicherheit der Verarbeitung (Art. 32)	Entra ID, RBAC, PIM, Key Vault, Verschlüsselung at rest/in transit	Starke AuthN/AuthZ, geringste Rechte, Verschlüsselung (Storage, DB, TLS), Schutz der Schlüssel.
DORA – ICT Risk Management & Resilience	Hub-Spoke-VNet, Azure Firewall, AKS mit Multi-AZ, Backup & DR-Strategie	Redundante Deployments, Netzsegmentierung, Backup/Restore, Test von Failover-Szenarien.
DORA – Monitoring & Incident Reporting	Azure Monitor, Sentinel, Defender for Cloud, SOC-Runbooks	Durchgängige Überwachung, Alarmierung, automatisierte Playbooks, strukturierte Incident-Daten für Reporting.
NIS 2 – Security of Network & Information Systems	VNet, NSGs, Firewall, WAF, Defender for Cloud	Schutz vor Netzangriffen, Härtung von Systemen, laufende Schwachstellen-Bewertung.
ISO 27001 – A.9 Zugriffskontrolle	Entra ID, RBAC, PIM, Bastion, APIM-Auth	Rollenbasierte Zugriffssteuerung, Just-in-Time-Privilegien, kontrollierte Admin-Zugriffe ohne Public IPs.
ISO 27001 – A.12 Betriebs-	Git-Repos, CI/CD, IaC	Versionierte Infrastruktur, klare

Regulierung / Anforderung	Architekturbaustein(e)	Technischer Beitrag
& Änderungsmanagement	(Terraform), Change-Approval Workflows	Trennung von Umgebungen, dokumentierte Changes, Rückverfolgbarkeit.
ISO 27001 – A.16 Incident Management	Sentinel, Splunk, Playbooks, Runbooks	Standardisierte Erkennung, Behandlung und Dokumentation von Sicherheitsvorfällen.

5. Security & Data Protection – Datenflüsse & Kontrollen

5.1 Datenflüsse

1. Eingabe-Prompts & Kontextdaten

- User → Frontend (TLS) → Front Door WAF → Application Gateway WAF → APIM
- APIM führt AuthN/AuthZ (Entra ID, OAuth2), Rate-Limiting und Input-Validierung durch.
- Prompt Firewall (APIM Policy + Content Safety) prüft Prompts auf **sensible Inhalte, PII, Policy-Verstöße**.
- Agent-Orchestrator auf AKS reichert Prompt mit **zugelassenem Kontext** aus Datenbanken / Vektorstore an.

2. Model-Calls & Outputs

- Orchestrator ruft Azure OpenAI (EU-Region / Data Zone) oder Self-hosted Model auf AKS per TLS auf. ([Microsoft Learn](#))
- Outputs werden erneut durch Content Safety / Post-Filter geleitet (Policy-check; „Red-Team“-Regeln).
- Relevante Parameter (Modell, Version, Confidence Scores, Decision IDs) werden geloggt.

3. Logs & Auditing

- Alle Requests/Responses (ggf. pseudonymisiert) werden in Log Analytics und/oder Data Lake gespeichert.
- Audit Logs für Admin-Aktivitäten (PIM, AKS-Cluster-Admin, Key Vault Access) werden zentral gesammelt.
- Zugriff auf Logs nur via RBAC + PIM; Export zu Sentinel & Splunk für SOC-Auswertung.

5.2 Verschlüsselung (at rest / in transit)

- At rest

- Standardmäßig: Storage, Disk, SQL, Cosmos, Event Hubs, Service Bus → Transparent Data Encryption / SSE mit Microsoft-Managed Keys.
- Für höhere Anforderungen: **Customer-Managed Keys (CMK)** in Key Vault / Managed HSM; z. B. für SQL, Storage, Event Hubs.
- **In transit**
 - Erzwingen von TLS 1.2+/1.3 auf allen Endpoints (APIM, AKS Ingress, OpenAI, DBs, Message-Broker).
 - Interne Kommunikation über Private Endpoints + VNet.

5.3 Zugriffssteuerung & PAM

- **Identity First:**
 - Alle User/Apps über **Entra ID** und **Managed Identities** authentifiziert.
 - RBAC für Azure-Ressourcen, Kubernetes-RBAC für Clusterebene, App-Rollenkonzepte im Frontend/API.
- **PAM mit PIM:**
 - Admin-Rollen (Subscription Owner, AKS Admin, Key Vault Admin) nur als **Just-in-Time**-Rollen mit Approval und begrenzter Dauer.
- **Zero Trust:**
 - Conditional Access (MFA, Compliant Device), kein direkter Zugriff auf Produktionsressourcen ohne PIM und Bastion.

5.4 Protokollierung, Auditability, Nachvollziehbarkeit

- Protokollierung von:
 - Benutzeraktionen (Frontends, APIs) mit Korrelations-IDs.
 - Modellaufrufen: Prompt-Metadaten (ggf. pseudonymisiert), Modell-ID, Version, Entscheidung, Confidence.
 - Datenzugriffe (SQL, Storage, Cosmos) über native Audit-Funktionen.
 - Admin-Aktivitäten (PIM, Key Vault, AKS-Clusterrollen).
- Speicherung:
 - Kurzfristig in **Log Analytics**, mittel-/langfristig in **Data Lake (immutable Storage)** zur Erfüllung gesetzlicher Aufbewahrungsfristen.

5.5 Data Residency & Souveränität

- Alle Schlüsselressourcen (AKS, SQL, Storage, OpenAI, Event Hubs, etc.) werden in EU-Regionen angelegt.
- Für Azure OpenAI: **EU-residente Konfiguration (Data Zones)**, um sicherzustellen, dass Prompts/Responses in der EU verarbeitet werden. ([Microsoft Learn](#))

- Restriktive Firewall/NSG-Regeln, um ausgehenden Traffic zu begrenzen; optional Private DNS, keine Public Endpoints für interne Systeme.

5.6 Mitigation von Modell- & Prompt-Risiken

- **Prompt Injection & Jailbreaks**
 - System-/Policy-Prompts zentral in Orchestrator verwaltet (nicht durch User überschreibbar).
 - Prompt-Filter in APIM + Content Safety; Blacklists/Regex und semantische Checks.
 - **Data Leakage**
 - Pseudonymisierung / Maskierung sensibler Daten vor der Übergabe an LLM.
 - Kein Speichern von Roh-Prompts in Logs ohne Maskierung; strikte Rollenrechte für Log-Zugriff.
 - **Bias, Fairness, Explainability**
 - Nutzung von Responsible-AI-Funktionen (z. B. Azure ML, externe Tooling) in Dev-/Testphase.([iks.fraunhofer.de](https://www.fraunhofer.de/de/ressourcen/publikationen/whitepapers/whitepaper-responsible-ai-fairness-explainability))
-

6. Betriebs- & Lifecycle-Konzept

6.1 Versionierung, Test & Deployment von KI-Modellen / Agents

- **Repos & Branching**
 - Code & Konfig (Infrastructure + App + Agents) in Git (Mono- oder Multi-Repo), Feature-Branched, PR-Gates.
- **Pipelines (CI/CD)**
 - Build: Linting, Tests, Security-Scans (SAST, Secrets-Scanning, Dependency-Scanning).
 - Container-Build in **Azure Container Registry** mit Signierung (Notary / cosign).
 - Deployment per Helm/Kustomize auf AKS (Blue-Green/Canary via Ingress Controller + APIM).
- **MLOps**
 - Modellversionierung, Evaluation gegen Testdatensätze, automatisierte Regressionstests.
 - Staging-Umgebung mit Shadow Traffic, Auswertung von KPIs, Bias-/Drift-Checks, bevor Go-Live.

6.2 Monitoring, Alerting, Incident Response

- **Monitoring**

- Health-Checks auf API/Frontend/Agents (SLIs: Latenz, Error-Rate, Throughput).
- Business-KPIs (z. B. akzeptierte Anträge, manuelle Overrides, Review-Raten).
- **Alerting**
 - Schwellenwert-basierte Alerts aus Azure Monitor (z. B. Error Rate > x%, Response Time > y ms).
 - Security Alerts aus Defender / Sentinel (z. B. ungewöhnliche Zugriffe, Brute-Force, Anomalien).
- **Incident Response**
 - Playbooks in **Sentinel** (z. B. automatisches Sperren eines API-Keys, Quarantäne von Nodes).
 - Runbooks in **Automation** für Routineoperationen (Log-Rollover, Skalierung, Rotationen).

6.3 Schwachstellenmanagement (Vulnerability Management)

- **Container & Dependencies**
 - Defender for Containers scannt Container-Images in ACR & Laufzeitumgebung.
 - CVE-Reports → Integration in CI-Pipeline als Quality Gate.
- **OS & PaaS**
 - Defender for Cloud überwacht VM-Guest-OS-Patching, Konfigurationsdrift, schwache Konfigurationen.
- **Regelmäßige Reviews**
 - Monatliche Security-Review-Boards, Abgleich mit DORA/NIS2/ISO27001-Kontrollkatalogen.

6.4 Terraform & Konfigurationsmanagement

- **State-Management**
 - Remote Backend (Azure Storage) mit RBAC, Private Endpoint, Soft Delete, Versioning aktiviert.
- **Policies & Guardrails**
 - **Azure Policy** und **Blueprints** (oder Landing Zone Templates), um nur EU-Regionen, verschlüsselte Storage-Accounts, Private Endpoints etc. zuzulassen.
- **Drift Detection**
 - Regelmäßige `terraform plan`-Runs und Compliance-Checks; Abweichungen werden als Incident behandelt.

Wenn du möchtest, kann ich im nächsten Schritt:

- ein **konkretes Beispiel-Setup** (Liste von Terraform-Ressourcen und Namespaces) skizzieren oder
- die **Compliance-Tabelle** weiter detaillieren (z. B. auf Ebene einzelner ISO-27001-Kontrollen).