Leistungsbeschreibung

API-Management Infrastruktur Plattform

Inhalt:

1 Einleitung....2
1.1 Das Unternehmen....2
1.2 Die Ausgangssituation....2
1.3 Zusammengefasste Leistungspflichten eines Auftragnehmers....2
2 Leistungsgegenstand....2
2.1 Proof Of Concept (POC)....3
2.2 Aufbauunterstützung....4
2.3 Training....5
3 Anforderungen an eine neue API-Infrastruktur-Plattform mit API-Management-Funktionalitäten....5
3.1 Funktionale Anforderungen....6
3.1.1 Gateways:....6
3.1.2 API-Policies:....7
3.1.3 Funktionale Erweiterungen von APIs:....7
3.1.4 API-Absicherung:....7
3.1.5 API-Deployment:....8
3.1.6 Hosting, Installation und Integration:....9
3.1.7 API Logging und Logforwarding:....10
3.1.8 API-Monitoring und Analyse:....11
3.1.9 API-Management:....11
3.1.10 API-Versionierung:....12
3.1.11 Developer-Portal Funktionalitäten:....12
3.1.12 Optionale Werkzeuge für API-Design und Implementierung:....13
3.2 B. Nicht funktionale Anforderungen....14
3.2.1 Marktsicherheit und Marktpräsenz des Produktes und des Herstellers:....14
3.2.2 Performance:....15
3.2.3 Skalierbarkeit:....15

3.2.4 Sicherheit, Sicherheitsdokumentation und Zertifizierung: 15
3.2.5 Verfügbarkeit: ..... **Fehler! Textmarke nicht definiert.**
3.2.6 Datenschutz: 16
3.2.7 Nutzungsstatistiken: 16
3.2.8 Support: 16
3.2.9 Training und Ausbildung: 16
3.2.10 Lizensierung: 17

1 Einleitung

1.1 Das Unternehmen

Die Techniker Krankenkasse (TK) ist eine bundesweite Krankenkasse mit rund 9,5 Millionen Mitgliedern und insgesamt rund 12,2 Millionen Versicherten. Als gesetzliche Krankenversicherung ist die TK eine Körperschaft des öffentlichen Rechts mit Selbstverwaltung und die größte Krankenkasse in Deutschland.

1.2 Die Ausgangssituation

Seit Januar 2025 betreibt die TK eine API-Management-Plattform, um Entwicklern einen Support bei der Entwicklung und Bereitstellung von APIs der TK für externe Geschäftspartner und interne Systeme bereit zu stellen. Die Basis dieser Plattform ist das Produkt 3scale von Red Hat. Dieses wurde vom Hersteller abgekündigt. Diese Leistungsbeschreibung dient dafür, die Anforderungen an ein Nachfolgeprodukt zu definieren, welches als neue Plattform-Basis die Migration des Betriebs der bestehenden APIs und eine State of the Art Weiterentwicklung der Funktionsumfangs bietet.

1.3 Zusammengefasste Leistungspflichten eines Auftragnehmers

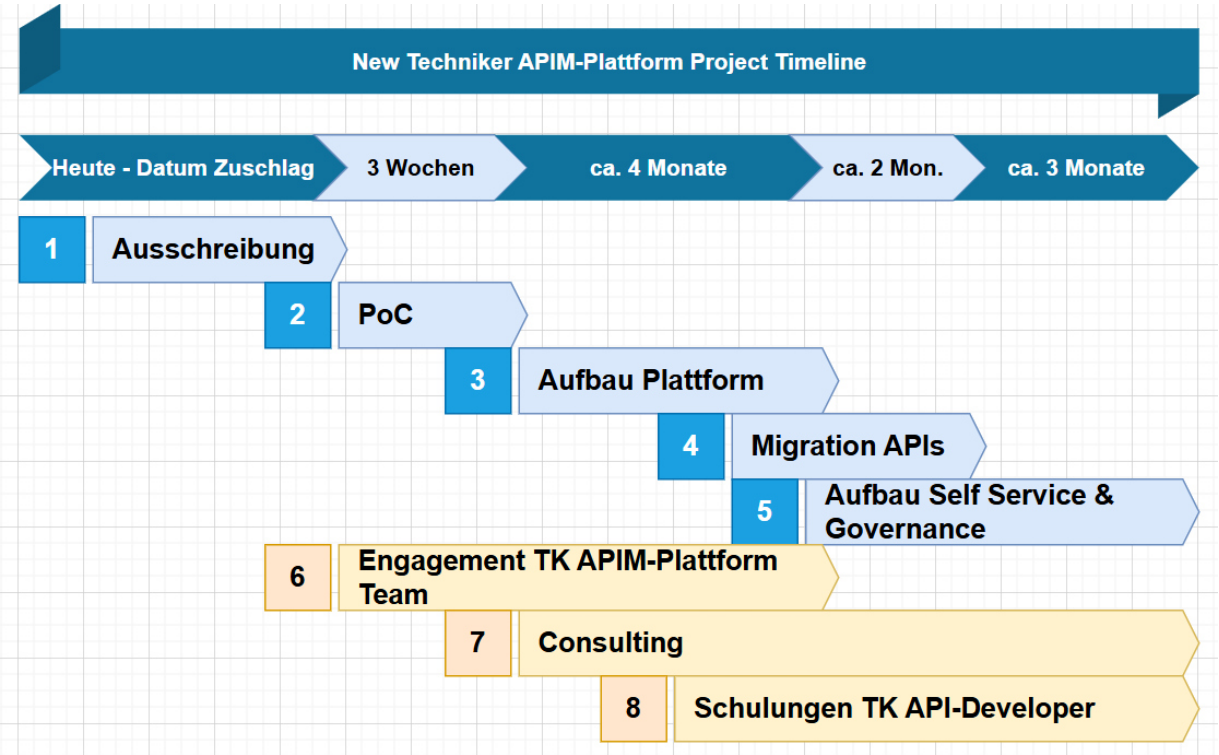
Der Auftragsnehmer (AN) muss ein Anbieter eines Produkts zum Aufbau und Betrieb einer API-Infrastruktur-Plattform mit API-Management-Funktionalitäten [im Folgenden APIM-Plattform-Produkt] sein. Der AN bietet Beratungs- und Supportdienstleitungen zum Aufbau und Betrieb einer Plattform in der TK auf Basis des APIM-Plattform-Produktes an. Die Leistungspflichten, die der AN erfüllen muss, sind im Kapitel 2 und die Anforderungen, welche von den zu beschaffenden Produktkomponenten erfüllt werden müssen, sind in den im Kapitel 3 formulierten Anforderungsdefinitionen beschrieben.

2 Leistungsgegenstand

Der Auftragnehmer (AN) stellt eine Lösung für den Aufbau einer hybriden API-Management-Plattform der TK bereit. Der AN stellt den Management-Anteil des API-Management-Tools als *Software as a Service* (SaaS) bereit und erbringt Beratungs- und Implementierungsleistungen.
Insbesondere erbringt der AN folgende Leistungen für die TK:

- Bereitstellung der Lizenzen
- Bereitstellung des API-Managers als SaaS
- Beratungsleistung für den Aufbau der On-Premise-Komponenten (Gateways)
- Durchführung von Trainings (Online oder Präsenz) für die TK-Mitarbeiter
- Beratungsleistung für die Einrichtung des API-Managements und bei der Migration bestehender APIs.
- Wartung, Pflege, Updates der SaaS Komponenten, sowie Support per Telefon und Mail

Die Sprache, in der alle Leistungen erbracht werden - inklusive Schulungen, E-Mail-Verkehr, Telefonaten, Benutzeroberfläche usw. - ist, soweit nicht anders angegeben, Deutsch. Einzelne Begrifflichkeiten dürfen aus dem Englischen übernommen werden, wie beispielsweise „Operator“, „Call“ oder „Teams“.



2.1 Proof Of Concept (POC)

Im Rahmen eines dreiwöchigen PoC stellt der AN eine funktionsfähige Pilotumgebung für den Aufbau einer hybriden API-Management-Plattform in Zusammenarbeit mit der TK bereit. Ziel des PoC ist es, die technische und organisatorische Machbarkeit der vorgeschlagenen Lösung zu evaluieren und ein gemeinsames Verständnis für den zukünftigen Betriebs- und Integrationsansatz zu schaffen. Der PoC dient der Entscheidungsfindung, ob die TK auf Basis des Produktes des AN die neue API-Management-Plattform aufbauen und betreiben wird.

Der AN stellt hierfür den Management-Anteil des API-Management-Tools als **Software as a Service (SaaS)** bereit und führt ergänzende **Beratungs- und Implementierungsleistungen** durch.

Die im Rahmen des PoC zu erbringenden Leistungen umfassen insbesondere:

- **Bereitstellung der Lizenzen** für die Nutzung der API-Management-Plattform im PoC-Zeitraum.
- **Bereitstellung und Konfiguration des API-Managers als SaaS-Lösung**, inklusive Anbindung an die On-Premise-Komponenten zum Test grundlegender Funktionen.
- **Beratung und Unterstützung** beim **Aufbau der On-Premise-Infrastruktur** zur Realisierung der hybriden Architektur für den Test grundlegender Funktionen.

Der Proof of Concept dient zur Entscheidung der Produkteignung anhand einer Verifizierung der technischen Anforderungen, der Bewertung der Benutzerfreundlichkeit und Integrationsfähigkeit sowie der Identifikation von Optimierungspotenzialen für den produktiven Aufbau der hybriden API-Management-Plattform.

2.2 Aufbauunterstützung

Im Rahmen der Aufbauunterstützung stellt der AN eine umfassende Begleitung zur Einrichtung und Inbetriebnahme einer hybriden API-Management-Plattform für die TK bereit. Ziel ist die Schaffung einer skalierbaren, sicheren und zukunftsfähigen Plattformarchitektur, die zentrale und dezentrale Komponenten integriert und die Verwaltung, Bereitstellung sowie Überwachung von APIs effizient ermöglicht.

Der AN übernimmt dabei den Aufbau des Management-Anteils der Lösung als **Software as a Service (SaaS)** und erbringt ergänzende **Beratungs- und Implementierungsleistungen**, um die TK bei der technischen und organisatorischen Implementierung zu unterstützen.

Die Aufbauunterstützung umfasst insbesondere folgende Leistungen:

- **Bereitstellung, Konfiguration und Inbetriebnahme des API-Managers als SaaS-Komponente**, inklusive Benutzer-, Rollen- und Zugriffsverwaltung.

- **Beratung und Unterstützung beim Aufbau der On-Premise-Komponenten** zur Integration in die bestehende IT-Infrastruktur der TK.

- **Beratung bei der Einrichtung und Anpassung des API-Managements** sowie **Unterstützung bei der Migration bestehender APIs** auf die neue Plattform.

Der AN stellt zu diesem Zweck selbst oder über einen Partner zwei technische Experten für das Produkt im Range eines Architekten und eines Consultants im Zeitrahmen vom Angebotszuschlag bis zum Ende des Projekts bereit. Durch diese Leistungen wird die TK in die Lage versetzt, die Verwendung und den Ausbau der hybriden API-Management-Plattform eigenständig fortzuführen und eine nachhaltige Grundlage für zukünftige Integrations- und Digitalisierungsinitiativen zu schaffen.

2.3 Training

Der AN erbringt ergänzende **Trainings- und Schulungsleistungen**, um die TK bei Betrieb, Optimierung und Verwendung der API-Management-Plattform zu unterstützen.

Insbesondere werden folgende Leistungen erbracht:

- **Durchführung von Trainingsmaßnahmen** (online oder in Präsenz) für die TK-DevOps, um den sicheren Umgang mit der Plattform zu gewährleisten.
- **Durchführung von Trainingsmaßnahmen** (online oder in Präsenz) zur Befähigung der TK-Mitarbeiter bei der Verwendung der Plattform für das Management von APIs.

3 Anforderungen an eine neue API-Infrastruktur-Plattform mit API-Management-Funktionalitäten

Anforderungen werden mit einer eindeutigen ID versehen. Sie sind möglichst kurz und beschreiben fassen soweit möglich keine Anforderung zusammen.
Anforderungen beinhalten immer eines der dem RFC 2119 [RFC2119] entsprechenden, in Großbuchstaben geschriebenen deutschen Schlüsselwörter MUSS, DARF NICHT, SOLL, SOLL NICHT, KANN.

Die Schlüsselwörter MUSS und DARF und deren Pluralform definieren Anforderungen die von einem Produkt zwingend zu erfüllen sind, daneben fließt die Art der Bereitstellung in die Produkt-Bewertung ein. Alle anderen fließen ausschließlich in die Bewertung-Bewertung ein und sind keine Ausschluss-Kriterien.

Anforderungen werden im Dokument wie folgt dargestellt:
<APIM-REQ-THEMENKÜRZEL-ID>
Text / Beschreibung
[<=]
Dabei umfasst die Anforderung sämtliche zwischen ID und Textmarke [<=] angeführten Inhalte.

Die TK geht für die kommenden Jahre von einem Bedarf aus, der wesentlich durch folgende drei Metriken definiert wird:

1.) **Die Anzahl der APIs:** Eine API im Sinne der Anforderungen ist ein Produkt, welches auf Basis einer Spezifikation (z.B. OpenAPI-Specification) eigenständig konfigurierbar ist und durch die Bereitstellung auf einem Gateway abgesichert lauffähig wird. Auf Basis der derzeit produktiven Plattform sind ca. 20 APIs seit Anfang 2025 mit unterschiedlichen Funktionsumfängen entstanden. Es entstehen laufend weitere. Die Annahme des Bedarfs wird von ca. 300 APIs Ende 2026 wachsend bis ca. 1000 APIs im Jahr 2030 geschätzt.
2.) **Anzahl der Gateways:** Ein Gateway im Sinne der Anforderungen ist eine Softwarekomponente, die On-Premise deployed ist und Requests an/von einem OpenShift Cluster annehmen und an Service Backends über konfigurierbare APIs delegieren und Responses zurückgeben kann. Aufgrund der Netzwerkumgebung und der Verwendung betreibt die TK derzeit unterschiedliche feste Instanzen von Gateways. Die Verwendungszecke

der vom Plattform-Team bereit gestellten unterschiedlichen Gateways adressieren zum einen extern erreichbare und intern erreichbare API-Endpunkte. Gateways für Entwicklung und Produktion und Installationen zur Weiterentwicklung der Infrastruktur der Plattform selbst. Darüber hinaus gibt es bereits heute den Bedarf für mehrere gleichartige APIs Im Sinne eines Fedarated-Gateway-Architekturansatzes different konfigurierte Gatewayinstanzen bereit zu stellen. Idealerweise so, dass es dem API-Developer Team selbst möglich ist die Konfiguration an Ihre Bedürfnisse anzupassen. Diese „federated“ Gateway-instanzen müssen keine physischen Gateways sein, sondern können als logisch separierte Konfigurationen bereitgestellt werden. Eine der gateway-api.sigs.k8s.io entsprechende Konfigurierbarkeit durch die Bereitstellung von Instanzen der GatewayClass vom Plattform-Team und Instanzen von Gateway durch die API-Entwickler-Fachteams wird hier als ideale Realisierung gesehen.

3.) **Anzahl der Requests:** Die derzeitige API-Management Plattform ist mit einem Volumen von ca. 18,5 Mrd. Requests p.a. für die produktiven Gateways lizensiert. Die Annahme des Bedarfs wird hier wachsend von 20 Mrd. Requests p.a. im Jahr 2027 auf 100 Mrd. Requests p.a.im Jahr 2030 geschätzt.

3.1 Funktionale Anforderungen

3.1.1 Gateways:

APIM-REQ-GW-001: Das System **MUSS** das „Federated API Gateway“-Architekturprinzip unterstützen, indem beliebig viele API-Gateway-Konfiguration in mehreren On-Premise-OpenShift-Clustern konfiguriert und betrieben werden können. [<=]

APIM-REQ-GW-002: API-Gateways **MÜSSEN** sowohl vom Plattformteam als auch von Entwicklerteams administriert und konfiguriert werden können. [<=]

APIM-REQ-GW-003: Die Spezifikation der Kubernetes Gateway API unter gateway.networking.k8s.io **MUSS** unterstützt werden. Herstellerspezifische Ergänzungen sind zulässig, dürfen aber die Kompatibilität nicht verletzen. [<=]

APIM-REQ-GW-004: Als Protokolle **MÜSSEN** REST (HTTP), GraphQL, gRPC, SOAP via HTTP/WebService unterstützt werden. [<=]

APIM-REQ-GW-005: Das System **MUSS** Event-getriebene APIs und Streaming-APIs über Protokolle wie Webhooks, WebSockets, AsyncAPI und serverseitig gesendete Ereignisse (SSE) unterstützen und MUSS Richtlinien auf diese ereignisgesteuerten und Streaming-APIs anwenden können. [<=]

APIM-REQ-GW-006: Gateways und APIs **MÜSSEN** ohne funktionale Einschränkungen unabhängig weiterarbeiten können, wenn eine angebundene SaaS- oder API-Manager-Komponente temporär nicht verfügbar ist. [<=]

APIM-REQ-GW-007: Das Produkt MUSS AI Gateway-Funktionalität bieten, die als Kontrollpunkt zwischen Anwendungen und KI-Modellen eingesetzt werden kann. OpenAI-kompatible Schnittstellen für die Anbindung von On-Premise/Cloud Modellen sind zu unterstützen. [<=]

APIM-REQ-GW-008: Eine Anbindung an Service Meshes wie z.B. Istio oder Linkerd als Backend für APIs **MUSS** unterstützt werden. [<=]

3.1.2 API-Policies:

APIM-REQ-PO-001: Das Produkt **MUSS** einen Katalog von Standard-Policies anbieten. [<=]

APIM-REQ-PO-002: Die Einbindung benutzerdefinierter Policies (Custom Policies) **MUSS** möglich sein**.**

APIM-REQ-PO-003: Standard- und Custom-Policies **MÜSSEN** in einzelne APIs sowie in API-Gruppierungen eingebunden werden können. [<=]

APIM-REQ-PO-004: Die Erstellung benutzerdefinierter Policies (Custom Policies) **SOLLTE** vorzugsweise mit einer standardisierten Expression Language wie CEL erfolgen. [<=]

3.1.3 Funktionale Erweiterungen von APIs:

APIM-REQ-FEA-001: Pro API **MÜSSEN** mehrere Backends konfigurier- und nutzbar sein (z. B. für Load-Balancing, Failover oder Routing nach Kriterien). [<=]

APIM-REQ-FEA-002: Funktionen für Splitting und Mirroring des API-Datenverkehrs für Blue/Green- und Canary-Tests **MÜSSEN** verfügbar sein. [<=]

APIM-REQ-FEA-003: Integrations-, Steuerungs- und Kontrollfunktionen für AI/ML-Service-APIs **MÜSSEN** im Katalog der Standard-Policies bereitgestellt werden, inklusive AI-Policies wie Guardrails, Prompt-Dekorator, Prompt-Templating, Token-Limitator und Semantik-Caching. [<=]

APIM-REQ-FEA-004: Die Orchestrierung **SOLL** die Koordination und Integration mehrerer APIs in einem einheitlichen, geschäftsprozessorientierten Workflow ermöglichen, der Änderungen und Anpassungen flexibel unterstützt. [<=]

APIM-REQ-FEA-005: Das Produkt **SOLL** die Transformation von Datenformaten in API-Payloads unterstützen, insbesondere von XML zu JSON. [<=]

APIM-REQ-FEA-006: Das Produkt **SOLL** die Möglichkeit bieten, bei OpenAI kompatiblen APIs eine Übersicht der verfügbaren Modelle und deren Status (Health) für die Nutzer bereitzustellen. [<=]

3.1.4 API-Absicherung:

APIM-REQ-ABS-001: Das System **MUSS** Authentifizierung und Autorisierung von API-Zugriffen mittels API-Key, OAuth2/ OpenID Connect (OIDC), JWT-Validierung, Basic Auth sowie weitere gängige Standards unterstützen. [<=]

APIM-REQ-ABS-002: Eine mTLS-Absicherung der bereitgestellten APIs und des Upstreams **MUSS** möglich sein. [<=]

APIM-REQ-ABS-003: Policies für Absicherung, CORS, Zugriffsbeschränkung (Rate-Limiting/Quota), Request- und Response-Manipulation, IP-Black- und Whitelisting, Header-Auswertung und -Modifikation, Zertifikatsprüfungen, DNS-Prüfungen und Claim-Checks bereitstellen **MÜSSEN** im Katalog der Standard-Policies integraler Bestandteil der Lösung sein. [<=]

APIM-REQ-ABS-004: Die integrierten API-Sicherheitsmechanismen **MÜSSEN** Schutz vor gängigen Sicherheitslücken und Bedrohungen gemäß OWASP Top 10 bieten. [<=]

APIM-REQ-ABS-005: Die Plattform **MUSS** entweder eine reibungslose Integration mit der bestehenden Kubernetes-basierten WAF-Lösung (F5 NGINX One) ermöglichen oder alternativ API-Security-Funktionalitäten in einem Umfang bereitstellen, der die externe WAF-Komponente technisch verzichtbar macht. [<=]

APIM-REQ-ABS-006: Es **MUSS** möglich sein, allgemeinen Schutz mandantenübergreifend für mehrere Gateways, Gateway-spezifischen Schutz pro Gateway und individuell konfigurierbaren Schutz pro einzelner API zu definieren. [<=]

APIM-REQ-ABS-007: Statische Prüfungen von API-Payloads gegen eine der verlangten API-Spezifikationen (OpenAPI, AsyncAPI) sind zu unterstützen; das Produkt **MUSS** entsprechende Validierungs-Policies im Katalog der Standard-Policies bereitstellen. [<=]

APIM-REQ-ABS-008: Custom Policies für frei programmierbare Prüfungen auf Basis der Daten der Payload und der API-Requests und -Responses **MÜSSEN** möglich sein. [<=]

APIM-REQ-ABS-009: Es **MUSS** möglich sein, einen Identity Provider (IDP)-Tool wie z.B. Keycloak in das Produkt zur Absicherung der APIs mit OIDC einzubinden. [<=]

APIM-REQ-ABS-010: Es **MUSS** Funktionen zur sicheren Verwaltung von Zertifikaten und privaten Schlüsseln bereitstellen, einschließlich automatisierbarer Erneuerung und sicherer Ablage. [<=]

APIM-REQ-ABS-011: Das Produkt **SOLL** flexible TLS-Terminierung ermöglichen, etwa durch Zero-Knowledge TLS-Konfigurationen. [<=]

3.1.5 API-Deployment:

APIM-REQ-APD-001: Das Produkt **MUSS** ermöglichen, dass eine API inklusive aller zugehörigen Artefakte (z. B. Spezifikationen, Policies, Konfigurationen) über eine REST-API im Rahmen einer CI/CD-Pipeline deployt werden kann. [<=]

APIM-REQ-APD-002: Alle Funktionen der API-Management-Plattform **MÜSSEN** über eine wohldefinierte, versionierte API vollständig automatisierbar sein. [<=]

APIM-REQ-APD-003: Das Produkt **SOLL** einen GitOps-Mechanismus bereitstellen, der auf geeigneten Custom Resource Definitions (CRDs) basiert und deklaratives Deployment von APIs ermöglicht. [<=]

3.1.6 Hosting, Installation und Integration:

APIM-REQ-HOS-001: Das Produkt **MUSS** hybride Deployment-Varianten unterstützen, die eine flexible Betriebsmodellumsetzung erlauben, wobei die Control Plane (API-Administration) als SaaS angeboten werden. [<=]

APIM-REQ-HOS-002: Die API-Gateways (Data Planes) **MÜSSEN** On-Premise deploybar sein. Eine mögliche Cloud-Bereitstellung der Gateways wird positiv bewertet. [<=]

APIM-REQ-HOS-003: Das Produkt **MUSS** ein Multi-Tenant-Konzept zur strukturellen Trennung von Systemen, API-Katalogen und Organisationseinheiten umfassen. [<=]

APIM-REQ-HOS-004: Für die On-Premise-Komponenten **MUSS** eine Kubernetes-native Installation möglich sein, inklusive der automatisierbaren Nutzung von Kubernetes-Fähigkeiten für Resilienz und Skalierung. [<=]

APIM-REQ-HOS-005: Das Produkt **MUSS** ein separates IDP-Tool integrieren können. [<=]

APIM-REQ-HOS-006: Das Produkt **MUSS** Versionskontrolle, Backup, Ausfall- und Wiederherstellungsprozesse bereitstellen, dabei sollten Konfigurationsdaten und Laufzeitinformationen in Kubernetes-nativen Datenspeichern (z. B. etcd) abgelegt und Backups über k8s-Mechanismen realisiert werden. [<=]

APIM-REQ-HOS-007: Funktionen zur Datenextraktion und vollständigen Löschung (Data Eradication) **MÜSSEN** vorhanden sein. [<=]

APIM-REQ-HOS-008: Die Sicherheit für übertragene und gespeicherte Daten **MUSS** mittels Verschlüsselung gewährleistet werden.

APIM-REQ-HOS-009: Nutzer **MÜSSEN** alle Aspekte der Datenfreigabe steuern können, inkl. Deaktivierung bei externem Datenaustausch z. B. mit LLMs oder KI-gestützten Anwendungen. [<=]

APIM-REQ-HOS-010: Monitoring von Datenfreigaben **MUSS** unterstützt werden. [<=]

APIM-REQ-HOS-011: Für SaaS-Komponenten **MUSS** das Hosting in Rechenzentren innerhalb Deutschlands oder der EU gewährleistet sein. [<=]

APIM-REQ-HOS-012: Loadbalancing des Datenverkehrs ausschließlich auf HTTP/HTTPS-Ebene über die OpenShift-Router **MUSS** unterstützt werden. [<=]

APIM-REQ-HOS-013: Das Produkt **MUSS** einfache Integration in OpenShift 4.x (aktuelles Minor-Release und alle folgenden Releases) erlauben für Komponenten, die On-Premise gehostet werden. [<=]

APIM-REQ-HOS-014: Konfigurationen **MÜSSEN** bei On-Premise-Komponenten ausschließlich via Pull aus SaaS-Komponenten erfolgen (kein Zugriff von SaaS auf On-Premise). Jeglicher ausgehender Internet-Zugriff erfolgt ausschließlich über explizite Proxies der TK. Die Proxies führen eine SSL-Inspection durch, d. h. TLS-Verbindungen werden aufgebrochen. Entsprechend müssen Anwendungen

und Base-Images mit der internen CA bzw. den angepassten Zertifikatsketten umgehen können. Direkter Internet-Zugriff aus Pods oder Nodes ist nicht möglich. [<=]

APIM-REQ-HOS-015: Externe Exponierung **MUSS** ausschließlich über die vorgesehenen OpenShift-Mechanismen (Routes/Ingress) erfolgen. [<=]

APIM-REQ-HOS-016: Für die On-Premise-Komponenten **DARF** ausschließlich NFS-basierter Storage verwendet werden. [<=]

APIM-REQ-HOS-017: Für alle On-Premise-Komponenten **MÜSSEN** Request/Limits sauber gesetzt werden. Anpassungen an geltende Resource-Quotas (CPU, RAM, Storage; pro Pod und auch aggregiert) sind nur innerhalb definierter Standard-Node-Größen möglich. [<=]

APIM-REQ-HOS-018: Ephemerer Storage **DARF** nur eingeschränkt genutzt werden und nicht für persistente oder größere Datenmengen eingeplant werden. [<=]

APIM-REQ-HOS-019: Für die On-Premise-Komponenten **DÜRFEN** keine Cluster-Admin-Berechtigungen nötig sein. Berechtigungen sind projektbasiert und folgen dem OpenShift-Standardmodell. Falls erforderlich werden erweiterte Rechte über (Cluster-)Roles vergeben. [<=]

APIM-REQ-HOS-020: Für die On-Premise-Komponenten **DÜRFEN** ausschließlich standardmäßige Security Context Constraints zum Einsatz kommen. [<=]

APIM-REQ-HOS-021: Policy-Enforcement über standardisierte Werkzeuge wie z.B. via OPA/Kyverno und standardisierten Sprachen wie CEL **SOLL** unterstützt werden. [<=]

APIM-REQ-HOS-022: Das Produkt **SOLL** ein Identity Provider (IDP)-Tool als SaaS-Komponente anbieten. [<=]

APIM-REQ-HOS-023: Für SaaS-Komponenten **SOLL** das Hosting in Rechenzentren von EU-souveränen DSGVO-konformen Cloud-Lösungen (z.B.: STACKIT oder vergleichbare Anbieter) möglich sein. [<=]

3.1.7 API Logging und Logforwarding:

APIM-REQ-LOG-001: Spezifisches Verhalten für Logging und die Einhaltung von Compliance-Vorgaben **MÜSSEN** über konfigurierbare Policies definiert werden können. Diese sind im Katalog der Standard-Policies bereitzustellen. [<=]

APIM-REQ-LOG-002: Das Produkt **MUSS** Filterfunktionen anbieten, um sensible Daten vor der Log-Weiterleitung aus den Logs zu entfernen. [<=]

APIM-REQ-LOG-003: Das Produkt **MUSS** umfassende Logging-Funktionen bereitstellen, die alle API-Aktivitäten protokollieren, einschließlich Nutzungsdaten, administrativer Zugriffe und Prozesssteuerungen. [<=]

APIM-REQ-LOG-004: Das Produkt **MUSS** die Weiterleitung von Logs an externe Systeme für Analyse und Archivierung unterstützen. [<=]

APIM-REQ-LOG-005: Das Produkt **MUSS** kundenspezifisches (Custom) Logging unterstützen, um individuelle Anforderungen an die Protokollierung abdecken zu können. [<=]

APIM-REQ-LOG-006: Logdaten **MÜSSEN** gegen unbefugten Zugriff geschützt sein. [<=]

3.1.8 API-Monitoring und Analyse:

APIM-REQ-MON-001: Standard-Policies für Observability-Features **MÜSSEN** im Katalog der Standard-Policies integraler Bestandteil der Lösung sein. [<=]

APIM-REQ-MON-002: Das Produkt **MUSS** eine Überwachung und ein Tracking des API-Traffics in Echtzeit ermöglichen. [<=]

APIM-REQ-MON-003: Für die Analyse der Nutzung und Performance **MUSS** das Produkt Metriken bereitstellen, die definierbar sind und entsprechende Daten liefern. [<=]

APIM-REQ-MON-004: Die Interoperabilität mit dem Grafana-Stack, einschließlich Prometheus und OpenTelemetry (OTEL), **MUSS** für die Data-Plane gewährleistet sein. [<=]

APIM-REQ-MON-005: Das Produkt **MUSS** für die Data- und Control-Plane Alerts und Benachrichtigungen konfigurieren können, die bei ungewöhnlichem Verhalten oder Sicherheitsvorfällen ausgelöst werden. [<=]

APIM-REQ-MON-006: Zur individuellen Überwachung von APIs **MUSS** das Produkt die Definition von Custom-Metriken ermöglichen. [<=]

APIM-REQ-MON-007: Nutzer der APIs **MÜSSEN** ihre Service Levels in Echtzeit verfolgen können, basierend auf Verfügbarkeit, Leistung und Nutzungsmustern. [<=]

APIM-REQ-MON-008: Die Plattform **MUSS** proaktive Alerts bei Systemereignissen und Resolution-Reporting bereitstellen. [<=]

APIM-REQ-MON-009: Vordefinierte OpenTelemetry-Pipelines **SOLLEN** unterstützt werden, um Telemetriedaten effizient zu erfassen und auszugeben. [<=]

3.1.9 API-Management:

APIM-REQ-MAN-001: Das Produkt **MUSS** ein Management-Portal als SaaS bereitstellen, das das Steuern und Verwalten der Prozesse zur Nutzung der APIs ermöglicht. [<=]

APIM-REQ-MAN-002: Das Management-Portal **MUSS** die Definition und Überwachung von Service Level Agreements (SLAs) unterstützen. [<=]

APIM-REQ-MAN-003: Das Management-Portal **MUSS** das Monitoring der API-Nutzung inklusive Metriken und Reporting-Funktionalitäten ermöglichen. [<=]

APIM-REQ-MAN-004: Das Management-Portal **MUSS** eine Funktionalität zur Ausgabe, Verwaltung und Einsatz von Zugangsschlüsseln bereitstellen. [<=]

APIM-REQ-MAN-005: Das Management-Portal **MUSS** eine rollenbasierte Zugriffskontrolle (RBAC) und Identity-Integration für Benutzer und Dienste ermöglichen, um differenzierte Zugriffsrechte abzubilden. [<=]

APIM-REQ-MAN-006: Single Sign-On (SSO) Support für das Management-Portal unter Nutzung von OpenID Connect (OIDC) **MUSS** vorhanden sein. [<=]

APIM-REQ-MAN-007: Bei einer Anbindung von KI-Modellen **MUSS** eine Möglichkeit vorhanden sein, die Berechtigungen der Nutzergruppen für unterschiedliche KI-Modelle zu verwalten. [<=]

3.1.10 API-Versionierung:

APIM-REQ-VERS-001: Das Produkt **MUSS** die Verwaltung und Bereitstellung verschiedener API-Versionen unterstützen. [<=]

APIM-REQ-VERS-002: Das Produkt **MUSS** parallele Nutzung mehrerer API-Versionen ermöglichen und SOLL die Migration zwischen Versionen unterstützen. [<=]

APIM-REQ-VERS-003: Das Produkt **MUSS** die gängigen API-Versionierungsmethoden unterstützen, insbesondere Header-basierte Versionierung.

APIM-REQ-VERS-004: API-Versionen **MÜSSEN** mittels Status wie „published“, „deprecated“ und „deleted“ verwaltet werden können, um den gesamten API-Lifecycle steuerbar zu machen. [<=]

APIM-REQ-VERS-005: Das Produkt **SOLL** ein graduelles Rollout neuer API-Versionen ermöglichen. [<=]

3.1.11 Developer-Portal Funktionalitäten:

APIM-REQ-DEP-001: Das Developer-Portal **MUSS** einen Self-Service-Katalog der APIs bereitstellen sowie eine API-Suche und die Verwaltung von API-Subscriptions ermöglichen. [<=]

APIM-REQ-DEP-002: Das Produkt **MUSS** die Anpassung des Erscheinungsbilds (Look & Feel) des Developer-Portals unterstützen. [<=]

APIM-REQ-DEP-003: Das Developer-Portal-Login **MUSS** rollenbasierte Zugriffskontrolle (RBAC) mit Authentifizierung über OpenID Connect (OIDC) bieten. [<=]

APIM-REQ-DEP-004: Nutzerprofile **MÜSSEN** definierbar sein. [<=]

APIM-REQ-DEP-005: Das Portal **MUSS** Genehmigungsworkflows für die Nutzung von APIs ermöglichen und automatische Notifikationen bei Statusänderungen generieren. [<=]

APIM-REQ-DEP-006: API-Dokumentationen **MÜSSEN** standardkonform, insbesondere auf Basis von z.B. OpenAPI-Spezifikationen, bereitgestellt werden. [<=]

APIM-REQ-DEP-007: Verschiedene Zielgruppen **MÜSSEN** im Developer-Portal abbildbar sein. [<=]

APIM-REQ-DEP-008: Alle Funktionen des Developer-Portals **MÜSSEN** über Automatisierungs-APIs integrierbar sein. [<=]

APIM-REQ-DEP-009: Das Developer-Portal **MUSS** durch andere Produkte (z. B. Backstage) ersetzbar sein. [<=]

APIM-REQ-DEP-010: Das Developer-Portal **MUSS** die Möglichkeit der Generierung von neuen Credentials und das Setzen von historischen Credentials einer API und von bestehenden Credentials von APIs, die bisher auf einer anderen Plattform betrieben wurden, bieten. [<=]

APIM-REQ-DEP-011: Das Developer-Portal **SOLL** Aspekte des API-Lifecycles für Kunden möglichst gut unterstützen. [<=]

APIM-REQ-DEP-012: Das Portal **SOLL** Sandboxes und Trial-Testing als Teil des Portalkonzepts unterstützen. [<=]

APIM-REQ-DEP-013: Das Portal **SOLL** ähnlich einem Content-Management-System (CMS) in allen Aspekten anpassbar sein. [<=]

APIM-REQ-DEP-014: AI-Anwendungen und AI-Agents **SOLLEN** APIs mit MCP-Support auffinden, Zugangsberechtigungen verwalten und die Verwendung steuern können, um maschinelle Nutzung und Integration von TK-APIs zu ermöglichen. [<=]

APIM-REQ-DEP-015: Es SOLL eine Möglichkeit geben, mit der OpenID-Connect-Clients Zugang zu mehreren APIs erhalten können.

APIM-REQ-DEP-016: Es SOLL eine Möglichkeit geben, externe Geschäftspartner im Portal zu verwalten und deren Zugang zu APIs SOLL konfigurierbar sein.

APIM-REQ-DEP-017: Es SOLL eine Möglichkeit geben, Vertragsdaten im Portal verwalten zu können, mit denen der Zugang von externen Geschäftspartnern zu APIs konfigurierbar ist.

APIM-REQ-DEP-018: Der Zugang zu APIs SOLL zeitlich eingrenzbar sein.

3.1.12 Optionale Werkzeuge für API-Design und Implementierung:

APIM-REQ-DSG-001: Das Produkt **SOLL** Werkzeuge zur Erstellung und Implementierung von APIs mitliefern. [<=]

APIM-REQ-DSG-002: Das Produkt **SOLL** Funktionalitäten zur Unterstützung von API-Design, API-Mocking, API-Testing und Deployment von APIs bereitstellen und SOLL kollaboratives Arbeiten an API-Produkten ermöglichen. [<=]

APIM-REQ-DSG-003: Das Produkt **SOLL** ein API-Design-Tool enthalten, mit dem neue APIs und neue Versionen bestehender APIs erzeugt werden können. [<=]

APIM-REQ-DSG-004: Das API-Design-Tool **SOLL** beim API-Design Best-Practices unterstützen und bestehende APIs visualisieren. [<=]

APIM-REQ-DSG-005: Das API-Design-Tool **SOLL** Linting mit benutzerdefinierten Regeln (z. B. mittels Spectral) sowie die Einbindung von Data-Contracts unterstützen. [<=]

APIM-REQ-DSG-006: Das API-Design-Tool **SOLL** API-Spezifikationen in On-Premise-Registries, Git-Repositories und im lokalen Dateisystem verwalten und verwenden können. [<=]

APIM-REQ-DSG-007: Das Produkt **SOLL** einen Code-Editor bereitstellen, der „Design as Code“ für API-Spezifikationen unterstützt und **SOLL** eine KI-Integration zur Unterstützung im Editor bieten. [<=]

APIM-REQ-DSG-008: Das Produkt **SOLL** die Definition und Anwendung von Policies unterstützen, die in dem Produkt verwendet werden können, und zwar bereits in der API-Designphase. [<=]

APIM-REQ-DSG-009: Das Produkt **SOLL** funktionale API-Testszenarien unterstützen und deren Automatisierung ermöglichen. [<=]

APIM-REQ-DSG-010: Tests **SOLLEN** in Test-Suites zusammengeführt werden können und SOLLEN in CI/CD-Pipelines integrierbar sein. [<=]

APIM-REQ-DSG-011: Tests **SOLLEN** aus API-Spezifikationen generiert werden können. [<=]

APIM-REQ-DSG-012: OAuth-Flows **SOLLEN** mit den bereitgestellten Testwerkzeugen testbar sein. [<=]

APIM-REQ-DSG-013: Das Produkt **SOLL** API-Mocking unterstützen, einschließlich der automatischen Mock-Generierung aus API-Spezifikationen. [<=]

APIM-REQ-DSG-014: Das Produkt **SOLL** „Record & Replay“ von Testdurchführungen unterstützen. [<=]

APIM-REQ-DSG-015: Das API-Design-Tool **SOLL** auch Event-Driven-APIs und Streaming-APIs unterstützen. [<=]

APIM-REQ-DSG-016: Das Produkt **SOLL** KI-gestützte Entwickler-Unterstützungsfunktionen enthalten, die Code-Completion, Qualitätsprüfungen und kontextbezogene Hilfestellungen anbieten. Wird eine solche Unterstützungsfunktion angeboten, dann ist die Auswahl der KI-Anbieter konfigurierbar zu gestalten und es sind TK eigene On-Premise-LLMs zu integrieren.[<=]

APIM-REQ-DSG-017: Das Produkt SOLL für Schema Spezifikationen wie AsyncAPI und OpenAPISpec die Übersicht und Verwaltung der verwendeten Schemata mit einem Schema-Repository erleichtern. [<=]

3.2 B. Nicht-funktionale Anforderungen

Grundsätzliche nicht-funktionale Anforderungen können aus den Pflichtenheften und den Cloud AGBs entnommen werden.

3.2.1 Marktsicherheit:

APIM-REQ-MKT-001:

Das Produkt **MUSS** eine Langzeitperspektive über eine Roadmap nachweisen können, um eine stabile Strategie über 2 Jahre hinaus zu gewährleisten.

[<=]

APIM-REQ-MKT-002: Das Produkt **SOLL** auf Open-Source-Code basieren. [<=]

3.2.2 Performance:

APIM-REQ-PRF-001: API-Anfragen **MÜSSEN** eine Latenzzeit unter 50 ms aufweisen. [<=]

APIM-REQ-PRF-002: Alle Logeinträge **MÜSSEN** Zeitstempel enthalten, die maximal 500 ms von einer offiziellen Zeitquelle (z. B. NTP-Server) abweichen. [<=]

APIM-REQ-PRF-003: Caching **MUSS** von den Gateway-Komponenten unterstützt werden. [<=]

APIM-REQ-PRF-004: Policies für Caching, Verteilung an Backends und Traffic-Management **MÜSSEN** im Katalog der Standard-Policies bereitgestellt werden und integraler Bestandteil der Lösung sein. [<=]

APIM-REQ-PRF-005: Performancedaten wie Antwortzeit, Durchsatz und Fehleranzahl **MUSS** das Produkt erfassen und Integration in Berichtssysteme wie Grafana ermöglichen. [<=]

3.2.3 Skalierbarkeit:

APIM-REQ-SKA-001: Kubernetes/OpenShift-native Mechanismen wie Horizontal Pod Autoscaler (HPA) und Vertical Pod Autoscaler (VPA) zur dynamischen Skalierung des Workloads **MUSS** das Produkt unterstützen. [<=]

APIM-REQ-SKA-002: Das Produkt **MUSS** generell und unter optimaler Infrastruktur von 20 Milliarden API-Calls p.a. (2027) und bis zu 100 Milliarden API-Calls p.a. (2030) skalieren können. [<=]

APIM-REQ-SKA-003: Gateways **SOLLEN** clusterübergreifend betreibbar und skalierbar sein. [<=]

APIM-REQ-SKA-004: Die Anzahl der APIs pro Gateway **SOLL** durch das Produkt nicht begrenzt sein, ansonsten **SOLL** die Gateway-Anzahl nicht durch das Produkt begrenzt sein. [<=]

APIM-REQ-SKA-005: Gateways **SOLLEN** hybrid betrieben werden können (On-Premise, Cloud, verschiedene Cluster und VMs) bei zentraler Administration. [<=]

3.2.4 Sicherheit, Sicherheitsdokumentation:

APIM-REQ-SSZ-001: Der Anbieter **MUSS** für alle im Produkt eingesetzten Komponenten bekannte Schwachstellen (CVE) anhand eines anerkannten Bewertungsschemas (z. B. CVSS) bewerten und nach dem jeweils aktuell branchenüblichen Stand der Technik behandeln. [<=]

3.2.5 Datenschutz:

APIM-REQ-DAS-001: Im Falle der Verarbeitung von SGBV-Daten **SOLL** das Produkt zusätzliche Schutzmaßnahmen und Compliance-Vorgaben unterstützen, die den besonderen Schutzbedarf dieser Daten berücksichtigen. [<=]

3.2.6 Nutzungsstatistiken:

APIM-REQ-NST-001: Das Produkt **MUSS** die Erfassung und Analyse von Nutzungsdaten der APIs ermöglichen, um Nutzungsmuster und Performance zu bewerten. [<=]

APIM-REQ-NST-002: Die gesammelten Nutzungsinformationen **DÜRFEN NICHT** an Dritte weitergegeben werden, es sei denn, dies ist gesetzlich oder vertraglich ausdrücklich erlaubt. [<=]

3.2.7 Support:

APIM-REQ-SUP-001: Der AN **MUSS** 24/7 Support mit garantierten Antwortzeiten bieten, mit Unterstützung auf Deutsch und Englisch. [<=]

APIM-REQ-SUP-002: Support **MUSS** über Telefon, E-Mail, Chat und Wissensdatenbanken zugänglich sein. [<=]

APIM-REQ-SUP-003: Bei Prio-1 Vorfällen **MUSS** der Zugriff auf Support-Ingenieure sichergestellt werden. Akzeptanzkriterium: Kritische Vorfälle erhalten sofortige Eskalation und direkten Support durch qualifizierte Techniker. [<=]

APIM-REQ-SUP-004: Die Integration in die ITSM-Prozesse der TK **MUSS** ticketbasiert automatisierbar sein mit einsehbaren Tickets für TK-Mitarbeiter. [<=]

APIM-REQ-SUP-005: Ein Proof of Concept (PoC) **MUSS** kurzfristig mit kostenloser POC-Lizenz möglich sein. [<=]

APIM-REQ-SUP-006: Eine Betreuung durch dedizierte Technical Account Manager (TAM) **SOLL** möglich sein. [<=]

APIM-REQ-SUP-007: In der Beratungsleistung **SOLL** Support für Migrationen, z. B. von Red Hat 3scale möglich sein. [<=]

3.2.8 Training und Ausbildung:

APIM-REQ-TRA-001: Der AN **MUSS** Schulungen für Anwender über entsprechende Medien anbieten. [<=]

APIM-REQ-TRA-002: Die Schulungen **SOLLEN** interaktiv gestaltet sein und praxisnahe Übungen sowie Fallstudien enthalten, um effektives Lernen sicherzustellen. [<=]

APIM-REQ-TRA-003: Der AN **SOLL** flexible Schulungsformate anbieten, darunter offene Kurse, Inhouse-Schulungen und individuelle Trainings, die auf Kundenbedarfe zugeschnitten sind. [<=]

APIM-REQ-TRA-004: Support und Nachbetreuung für Schulungsteilnehmer **SOLLEN** angeboten werden, um nachhaltigen Lernerfolg zu gewährleisten. [<=]

APIM-REQ-TRA-005: Die Trainings **SOLLEN** fokussiert auf API-spezifische Themen sein, z. B. API-Design, API-Sicherheit, API-Testing und API-Management. [<=]

3.2.9 Lizenzierung:

APIM-REQ-LZM-001: Das angebotene Lizenzmodell **MUSS** bei Auswahl einer Lizenz nach oben und unten skalierbar sein. [<=]

APIM-REQ-LZM-002: Bei Abrechnung auf Basis von Zugriffszahlen **DARF KEINE** Limitierung der Zugriffe in Zeitsegmenten unter einer Woche erfolgen. [<=]

APIM-REQ-LZM-003: Die Anzahl der On-Premise laufenden APIs und Tenants **SOLLEN** vom Lizenzmodell her nicht begrenzt werden. [<=]