Technische und organisatorische Maßnahmen „TOM“ nach Art 32 DSGVO für KIWAY

Stand: 06.07.2026

techexx® GmbH
Johann Giefing-Straße 2a
A – 2700 Wiener Neustadt

office@techexx.at
+43 (0) 720 720 020
www.techexx.at

UID-Nr.: ATU77842104
Firmenbuchnr.: FN 574138x
Gerichtsstand: Wiener Neustadt

Wiener Neustädter Sparkasse
BIC: WINSATWNXXX
IBAN: AT86 2026 7020 0015 4035

Technische und organisatorische Maßnahmen (TOM) für KIWAY

Stand: 06.07.2026

1. Zweck und Geltungsbereich

Diese TOM beschreiben die technischen und organisatorischen Maßnahmen für den Betrieb der KIWAY SaaS-Plattform. Sie beziehen sich auf die aktive Plattform mit KIWAYAdminWebApp, KIWAYWebApp, API, ChatAPI, TechexxAPI, MonitorAPI, Models, ServiceFacade und SharedLibrary.

Die Maßnahmen dienen der Gewährleistung von Vertraulichkeit, Integrität, Verfügbarkeit und Belastbarkeit der Systeme im Sinne von Art. 32 DSGVO.

2. System- und Betriebsumgebung

- KIWAY wird als Multi-Tenant-SaaS betrieben.
- Der Kundenstack wird in Microsoft Azure gehostet; die Basisinfrastruktur liegt nach Betreiberangabe in Azure North Europe.
- Der aktive Kundenstack besteht aus: KIWAYAdminWebApp als Admin Center,
- KIWAYWebApp als Blazor WebAssembly App,
- API als HTTP API,
- ChatAPI als SignalR-/WebSocket-Service.
- Supabase Cloud wird als Identity Provider genutzt.
- Nach erfolgreicher Supabase-Authentifizierung stellt die API ein eigenes App-JWT mit Tenant- und Rolleninformationen aus.
- PostgreSQL wird als relationale Datenbank genutzt.
- Qdrant wird als Vector Database genutzt.
- Hintergrundjobs laufen über Hangfire mit In-Memory-Storage.
- Globale Funktionen und zentrales Logging laufen über TechexxAPI und MonitorAPI.

techexx® GmbH
Johann Giefing-Straße 2a
A – 2700 Wiener Neustadt

office@techexx.at
+43 (0) 720 720 020
www.techexx.at

UID-Nr.: ATU77842104
Firmenbuchnr.: FN 574138x
Gerichtsstand: Wiener Neustadt

Wiener Neustädter Sparkasse
BIC: WINSATWNXXX
IBAN: AT86 2026 7020 0015 4035

3. Zutrittskontrolle und Administrationszugriff

- Der Zugriff auf Azure-Ressourcen erfolgt über Azure RBAC.
- Administrative Azure-Zugriffe sind nach Betreiberangabe durch MFA und Conditional Access abgesichert.
- Die Azure-Berechtigungen sind granular über Gruppen gesteuert.
- Privileged Identity Management (PIM) wird aktuell nicht eingesetzt.
- Superadmin-Rechte in KIWAY sind nach Betreiberangabe auf Service Accounts beschränkt.
- Administrative App-Funktionen werden rollenbasiert eingeschränkt; das Admin Center prüft insbesondere auf Admin-, Sectionadmin- und Superadmin-Rollen.

4. Zugangskontrolle in der Anwendung

- Benutzer authentifizieren sich über Supabase Cloud.
- Der Login-Flow unterstützt E-Mail/Passwort und optional Microsoft OAuth via PKCE.
- KIWAY nutzt MFA im App-Flow: TOTP/App Authenticator wird über Supabase MFA angebunden.
- Als Fallback ist ein E-Mail-OTP-Flow vorhanden.
- Nach erfolgreicher MFA-Prüfung wird ein AAL2-Status bzw. MFA-Status in der lokalen Sitzung vermerkt.
- Die Anwendung leitet authentifizierte Benutzer ohne abgeschlossene MFA in den MFA-Flow weiter.
- Passwortrichtlinie im KIWAY-Reset-Flow: mindestens 8 Zeichen,
- mindestens ein Großbuchstabe,
- mindestens eine Zahl,
- mindestens ein Sonderzeichen.
- Das eigentliche Identitäts- und Passwortmanagement liegt bei Supabase Cloud.
- Aktive Sessions werden lokal bereinigt; beim Logout werden App-Session, Supabase-Sessionstatus und MFA-Flags entfernt.

5. Zugriffskontrolle und Rollenmodell

- KIWAY verwendet rollenbasierte Zugriffskontrolle.
- Im Code vorhandene App-Rollen: Superadmin,
- Admin,
- Sectionadmin,
- User.
- Fachliche Zuordnung nach Betreiberangabe: Superadmin: umfassende technische Administrationsrechte, auf Service Accounts beschränkt,
- Admin: Administrationsfunktionen im Tenant,
- Wissensmanager/Sectionadmin: Verwaltung bzw. Upload von Wissen und Chat-/Wissensbereichen,
- User: Nutzung der User App.
- API- und ChatAPI-Endpunkte nutzen Policies wie User, Sectionadmin, Admin, Superadmin und prüfen ein gültiges App-JWT mit tenant_id-Claim.
- Das App-JWT enthält mindestens:
  - Benutzer-ID (sub),
  - Tenant-ID,
  - Tenant-Name,
  - Tenant-Slug,
  - Rolle,
  - Anzeigename,
  - optional E-Mail.
  - App-JWTs werden mit Issuer, Audience, Signatur und Laufzeit validiert.
  - Die Standardlaufzeit des App-JWT beträgt 30 Minuten, sofern keine andere Konfiguration gesetzt ist.

6. Mandantentrennung

- Jeder normale App-Request muss einen Tenant-Kontext besitzen.
- Die API und ChatAPI extrahieren den Tenant-Kontext aus dem App-JWT und setzen ihn requestbezogen.
- Datenbankzugriffe werden über EF-Core-Interceptors kontrolliert: PostgreSQL-Session-Settings app.current_tenant_id und app.current_user_id werden gesetzt.
- Datenbankbefehle ohne Tenant-Kontext werden für tenantbezogene Operationen abgewiesen.
- Tenant-spezifische Dateipfade werden unterhalb eines Tenant-Roots aufgelöst.
- Pfadauflosung verhindert Zugriff außerhalb des jeweiligen Tenant-/Chatflow-Verzeichnisses.
- Qdrant-Collections werden tenant- und chatflow-spezifisch benannt.
- Externe Chat- und Public-Endpunkte prüfen Tenant-ID und Chatbot-Zugehörigkeit gegen den aktiven Tenant.

7. Netzwerksicherheit

- Ausgehende Requests laufen nach Betreiberangabe über eine Firewall mit Default-Deny-Prinzip; nur explizit freigegebene Ziele werden erlaubt.
- Eingehende Requests vom WebAssembly-Frontend zur API laufen nach Betreiberangabe über den Ingress des Azure Container Apps Environments ohne vorgeschaltete Firewall.
- API und ChatAPI setzen CORS-Policies auf die konfigurierten KIWAY-App- und Admin-App-Hosts.
- API und ChatAPI nutzen HTTPS-Redirection und HSTS.
- API und ChatAPI setzen sicherheitsbezogene Header, u. a.: X-Content-Type-Options: nosniff,
- X-Frame-Options: DENY,
- Content Security Policy,
- Referrer Policy,
- X-Permitted-Cross-Domain-Policies: none.
- Globale Rate Limits sind in API und ChatAPI konfiguriert: 100 Requests pro Minute je Benutzer bzw. Host, mit kleiner Queue.

8. Verschlüsselung und Transport

- Webzugriffe erfolgen über HTTPS.
- Datenbankverbindungen werden mit SslMode=Require aufgebaut.
- Qdrant wird für nicht-lokale Hosts über HTTPS angebunden.
- App-JWTs werden serverseitig signiert und validiert.
- Tenant-API-Keys werden nicht im Klartext gespeichert; gespeichert werden Prefix, Metadaten und ein HMAC-SHA256-Hash mit serverseitigem Pepper.
- Passwörter für öffentliche externe Chats werden nicht im Klartext gespeichert; es wird PBKDF2 mit HMAC-SHA256, Salt und 100.000 Iterationen verwendet.

9. Custom Apps, externe Workflows und öffentliche Formulare

- Custom Apps sind externe Workflow- bzw. Web-/Formular-URLs, die in KIWAY per iframe eingebunden werden.
- Custom Apps werden im Admin Center verwaltet und können Nutzern bzw. User-Gruppen zugeordnet werden.
- Das Anlegen bzw. Verwalten von Custom Apps ist im Admin Center auf Superadmin-Funktionen eingeschränkt.
- Custom-App-URLs sind nach Betreiberangabe nicht separat durch KIWAY verschlüsselt oder authentifiziert, können aber mit Passwortschutz gesichert werden.
- Öffentliche Formulare können über eine GUID im Link geschützt sein; dadurch wird Erraten erschwert.
- Für öffentliche externe Chats gibt es eine separate Public-ID (GUID) und optionalen Passwortschutz.
- Wenn ein Passwort für einen externen Chat gesetzt ist, wird es gehasht und vor der Prediction geprüft.

10. Protokollierung, Nachvollziehbarkeit und Monitoring

- Fehler werden zentral über MonitorAPI protokolliert.
- Nutzungsdaten werden über TechexxAPI protokolliert.
- Usage-Logs werden tenantbezogen geschrieben; ohne Tenant-Kontext wird das Schreiben abgewiesen.
- Authentifizierungs-, Token- und Autorisierungsereignisse werden in API und ChatAPI technisch geloggt.
- Rechtliche Akzeptanzen werden über Legal-Modelle und Legal-Controller gespeichert.

techexx® GmbH
Johann Giefing-Straße 2a
A – 2700 Wiener Neustadt

office@techexx.at
+43 (0) 720 720 020
www.techexx.at

UID-Nr.: ATU77842104
Firmenbuchnr.: FN 574138x
Gerichtsstand: Wiener Neustadt

Wiener Neustädter Sparkasse
BIC: WINSATWNXXX
IBAN: AT86 2026 7020 0015 4035

- Fehlerbehandlungs-Middleware reduziert in produktionsnahen Umgebungen Detailinformationen für Clients.

11. Verfügbarkeit und Belastbarkeit

- API und ChatAPI sind als getrennte Services aufgebaut; HTTP API und Echtzeit-/SignalR-Kommunikation sind getrennt.
- Die ChatAPI konfiguriert WebSocket, Server-Sent Events und Long Polling als SignalR-Transporte.
- Rate Limiting schützt APIs vor einfachen Überlastungsszenarien.
- Hangfire wird mit In-Memory-Storage betrieben; eine persistente Hangfire-Storage-Schicht ist bewusst nicht vorgesehen.

12. Datenschutz durch Technikgestaltung und datenschutzfreundliche Voreinstellungen

- Frontend-Projekte rufen Backend-Funktionen über die ServiceFacade auf.
- Geteilte DTOs und Modelle liegen zentral im Models-Assembly.
- Tenantbezogene Backend-Logik nutzt Tenant-Kontexte und Interceptors.
- Datei- und Wissenszugriffe sind tenantbezogen getrennt.
- Public-Endpunkte sind nur für explizit vorgesehene Funktionen vorhanden und prüfen Tenant-/Objektbezug.

13. Regelmäßige Prüfung und Tests

- Im Repository sind Integrationstests für sicherheitsrelevante Isolation vorhanden, u. a. für User-, Settings-, TenantApiKey-, Job- und ChatAPI-Isolation.
- Build- und Deployment-Prozesse laufen laut vorhandener Dokumentation über Azure DevOps.
- Kritische Architekturregeln sollen durch Tests oder Analyzer ergänzt werden, soweit maschinell prüfbar.

14. Unterauftragnehmer und externe Dienste

- Unterauftragnehmer und externe Dienste können im AVV und SaaS eingesehen werden, eine vollständige Liste befindet sich auch im Admin Interface.

15. Organisatorische Maßnahmen

- Regelmäßige Schulungen im Bereich der DSGVO
- Regelmäßige Schulungen im Bereich der KI-Kompetenz

16. Verantwortliche

Verantwortliche im Sinne des Art 4 Z 7 DSGVO ist:

techexx® GmbH

Johann Giefing-Straße 2a, 2700 Wiener Neustadt
Geschäftsführer: Friedrich Schöls
FN 574138x HG Wiener Neustadt
Tel: +43 720 720 020
e-mail: office@techexx.at

techexx® GmbH
Johann Giefing-Straße 2a
A – 2700 Wiener Neustadt

office@techexx.at
+43 (0) 720 720 020
www.techexx.at

UID-Nr.: ATU77842104
Firmenbuchnr.: FN 574138x
Gerichtsstand: Wiener Neustadt

Wiener Neustädter Sparkasse
BIC: WINSATWNXXX
IBAN: AT86 2026 7020 0015 4035