AI Control Tower — Architektur-Arbeitsvorlage

Azure AI Gateway Decision Kit

Entscheidungsmatrix für direkten Modellzugriff, gemeinsames AI Gateway und zentrale Kontrollschicht

Unabhängige Arbeitsvorlage — keine Microsoft-Publikation und keine Produktempfehlung.
Dieses Kit hilft technischen Plattform-, Cloud- und Architekturverantwortlichen, eine zentrale Frage in Azure-Umgebungen fundiert zu strukturieren: Reicht der direkte Zugriff auf Modellendpunkte aus, oder sollten wir ein gemeinsames AI Gateway bzw. eine zentrale Kontrollschicht einrichten?

1. Die drei Muster im Vergleich

Vergleichen Sie die typischen Ausprägungen anhand konkreter Architektursignale:

Kriterium	Direkter Zugriff	Gemeinsames Gateway	Zentrale Kontrollschicht
Clients	Wenige, gut steuerbare Clients	Mehrere Clients mit ähnlichen Wegen	Heterogene Clientlandschaft
Modelle/Provider	Wenige feste Endpunkte	Mehrere Deployments zentral geroutet	Multi-Model / Multi-Provider Steuerungsrahmen
Authentifizierung	Je Client lokal verwaltet	Client- und Backendzugang getrennt	Übergreifende Identitäts- und Rollenmuster
Guardrails	Lokal im Code gepflegt	Wiederkehrende Regeln an einem Punkt	Zentrale Kontrollen mit klarem Owner
Rate Limits	Lokal ausreichend steuerbar	Konsistente Drosselungsreaktion	Bereichsübergreifende Quoten & Evidenz
Routing	Client wählt Endpunkt direkt	Failover / Fallback im Gateway	Dynamisches Routing mit Nachweisbarkeit
Kostenzuordnung	Direkte Zuordnung pro Host	Nutzung je Use-Case sichtbar	Portfolio- & FinOps-Zuordnung
Betrieb	Ein Anwendungs-Team	Dedizierter Gateway-Owner	Plattform-, FinOps- & Security-Rollen

2. Zwölf Entscheidungsfragen zur Selbsteinschätzung

Bewerten Sie jede Frage intern mit: **[ ] Heute klar** · **[ ] Uneinheitlich** · **[ ] Unbekannt**

1. Wie viele Anwendungen greifen heute oder im nächsten Quartal auf Modellendpunkte zu?

[ ] Heute klar geregelt [ ] Uneinheitlich je Team [ ] Unbekannt / Ungeklärt

2. Greifen diese Clients auf dieselbe Bereitstellung oder mehrere Providerpfade zu?

[ ] Heute klar geregelt [ ] Uneinheitlich je Team [ ] Unbekannt / Ungeklärt

3. Werden Zugangsdaten oder API-Keys direkt an mehrere Anwendungen verteilt?

[ ] Heute klar geregelt [ ] Uneinheitlich je Team [ ] Unbekannt / Ungeklärt

4. Kann ein Client einen Modellendpunkt außerhalb des vorgesehenen Kontrollpunkts erreichen?

[ ] Heute klar geregelt [ ] Uneinheitlich je Team [ ] Unbekannt / Ungeklärt

5. Müssen Limits, Inhaltsfilter oder Guardrails für mehrere Anwendungen konsistent gepflegt werden?

[ ] Heute klar geregelt *[ ] Uneinheitlich je Team* *[ ] Unbekannt / Ungeklärt*

6. Wo wird entschieden, welches Modell oder welcher Provider für eine Anfrage verwendet wird?

[ ] Heute klar geregelt *[ ] Uneinheitlich je Team* *[ ] Unbekannt / Ungeklärt*

7. Wie wird reagiert, wenn ein Modellendpunkt drosselt (HTTP 429) oder fehlschlägt?

[ ] Heute klar geregelt *[ ] Uneinheitlich je Team* *[ ] Unbekannt / Ungeklärt*

8. Müssen Gespräche oder zustandsbehaftete Abläufe an dieselbe Bereitstellung gebunden bleiben?

[ ] Heute klar geregelt *[ ] Uneinheitlich je Team* *[ ] Unbekannt / Ungeklärt*

9. Kann die Nutzung einem Use-Case oder Budget zugeordnet werden (auch Multi-Backend)?

[ ] Heute klar geregelt *[ ] Uneinheitlich je Team* *[ ] Unbekannt / Ungeklärt*

10. Wo werden Fehler-, Latenz-, Token- und Kostensignale zentral zusammengeführt?

[ ] Heute klar geregelt *[ ] Uneinheitlich je Team* *[ ] Unbekannt / Ungeklärt*

11. Wer verantwortet Konfigurationsänderungen bei neuen Modellen oder Regionen?

[ ] Heute klar geregelt *[ ] Uneinheitlich je Team* *[ ] Unbekannt / Ungeklärt*

12. Welche Sicherheits- und Compliance-Vorgaben müssen beim Modellzugriff nachweisbar sein?

[ ] Heute klar geregelt *[ ] Uneinheitlich je Team* *[ ] Unbekannt / Ungeklärt*

3. Drei typische Signalmuster

Muster A: Direkter Modellzugriff bleibt vorerst optimal
Signale: Wenige Anwendungen (< 3), stabiler Endpunkt, klare Zuständigkeit im selben Team.
Empfehlung: Keine unnötige Middleware aufbauen. Direkten Zugang sauber dokumentieren.

Muster B: Gemeinsames AI Gateway ist prüfenswert
Signale: Mehrere Teams nutzen ähnliche Endpunkte; Routing, Quoten und Fallbacks entstehen mehrfach.
Empfehlung: Gateway-Architektur (z. B. Azure API Management / AI Gateway) technisch evaluieren.

Muster C: Zentrale Kontroll- und Evidenzschicht erforderlich
Signale: Wachsende Use-Cases, heterogene Provider, FinOps-Transparenz und zentrale Guardrails gefordert.
Empfehlung: AI Control Tower als strukturierte Implementierung im eigenen Azure-Tenant einordnen.

4. Internes Decision Brief (Arbeitsvorlage)

Verwenden Sie diese Struktur für Ihre interne Abstimmung zwischen IT-Leitung, Architektur und Fachbereichen:

Bestehende Use-Cases & Clients:

Aktuelles Zugriffsmuster & Token-Verwaltung:

Größte technische Reibungspunkte (z. B. Quoten, Kosten, Transparenz):

Empfohlener nächster Schritt:

Nächster Schritt — AI Control Tower einordnen
Wenn eine gemeinsame technische Zugangsschicht als nächster Prüfpfad sichtbar wird, zeigt die Produkt-Journey, wie der **AI Control Tower** als begrenzte Implementierungsleistung in Ihrem Azure-Tenant umgesetzt und abgegrenzt wird.

Kontakt & Erstgespräch: hallo@controltowerai.de · https://controltowerai.de