

GEM CORTEX AI

– TECHNICKÁ A PROVOZNÍ SPECIFIKACE

Hlavní princip: Generativní AI je v Cortexu řízená součástí podnikového systému: dostává identitu, oprávnění, zdroje, limity, měření a bezpečnou možnost odmítnout nebo eskalovat.

Manažerské technické shrnutí

- ✓ Každý požadavek se posuzuje v kontextu identity, účelu, datové klasifikace, povolených nástrojů a rozpočtu.
- ✓ Deterministická cesta má přednost tam, kde data-báze, pravidlo nebo API poskytne přesnější, levnější a auditovatelnější výsledek než LLM.
- ✓ Znalosti, modely, nástroje a řídicí vrstva jsou odděleny tak, aby šly měnit, testovat a provozně vlastnit samostatně.
- ✓ Akce s dopadem na podnikový systém používají explicitní oprávnění, limity, záznam a podle rizika lidské schválení.
- ✓ Kvalita se nehodnotí podle plynulosti textu, ale podle přijatého výsledku, citací a jednotkového nákladu.

Minimální produktový kontrakt

- ✓ **Vstup:** Identita nebo workload, účel, klasifikace dat, korelační ID a nákladový kontext.
- ✓ **Rozhodnutí:** Zvolená runtime cesta, model/nástroj, aplikovaná policy.
- ✓ **Výsledek:** Odpověď, strukturovaný draft nebo stav akce včetně zdrojů.
- ✓ **Kontrola:** Odmítnutí, schválení člověkem, limity nástroje, timeout, rollback.
- ✓ **Důkaz:** Auditní stopa, telemetrie, evaluační metrika a nákladové přiřazení.



1. Referenční architektura a rozdělení odpovědností

Architektura je logická a vrstvená. Jejím účelem je oddělit odpovědnosti tak, aby bylo možné nezávisle řídit identitu, rozhodnutí o cestě, práci se znalostmi, volání nástrojů, modelový runtime a provozní kontrolu. To usnadňuje bezpečnostní review, testování změn i smluvní rozdělení provozu.

- **Kanály a konzumenti:** *Workspace, portály, API, batch. Klient nemusí zavádět další chat.*
- **Access:** *SSO, role, uživatelská a workload identita, datový profil. AI dědí kontext oprávnění.*
- **Core & Flow:** *Klasifikace záměru, volba runtime cesty, policy a workflow. Stejný vstup je vyřešen pravidlem, RAG, modelem nebo nástrojem.*
- **Knowledge & Tools:** *Ingest, index, retrieval, citace, konektory a registry nástrojů. Znalosti mají vlastní životní cyklus.*
- **Model Execution:** *Adaptéry poskytovatelů, lokální inference, routing a fallback. Model je vyměnitelná komponenta.*
- **Control:** *Audit, telemetrie, evaluace, náklady a rozpočty. Společný důkaz pro CIO, CISO a finance.*

Datová a řídící rovina: Datová rovina přenáší konkrétní požadavek a výsledek. Řídící rovina spravuje konfigurace, katalogy, policy a rozpočty. Obsah promptů, dokumentů a odpovědí není automaticky vhodný pro běžné aplikační logy; telemetrie je minimalizovaná a klasifikovaná.

2. Runtime cesty a rozhodovací logika

Runtime cesta vyjadřuje, jakým mechanismem vznikne výsledek. Volba se nemá řídit marketingovou atraktivitou modelu, ale povahou úlohy, dostupností podkladů, požadovanou přesností, oprávněním, latencí a jednotkovou ekonomikou. Router vrací nejen výsledek, ale i strojově čitelný důvod zvolené cesty.

- **A – Direct / no LLM:** *Pro známý lookup, pravidlo nebo přesný výpočet. Výstup z DB/API. Kontrola: schéma, oprávnění, timeout. Nejnižší variabilita, testovatelný a auditovatelný.*
- **B – RAG + LLM:** *Pro odpověď vyžadující interní znalost. Text s citacemi. Kontrola: ACL, retrieval eval, absence. RAG není lék na halucinace; kvalitu určuje ACL a evaluační sada.*
- **C – General model:** *Pro obecnou transformaci bez interních dat. Textový výstup. Kontrola: datový profil, filtr. Snižuje riziko záměny obecné odpovědi za podnikové stanovisko.*

- **D – Tool / Workflow:** Pro čtení nebo změnu podnikového systému. Výsledek nástroje. Kontrola: scope, approval, idempotence, audit. Zápis a významné akce používají limit, preview a lidské schválení.
- **E – Batch / Document:** Pro velký objem souborů nebo dlouhou úlohu. Dataset, dokumentový draft. Kontrola: fronta, checkpoint, human review. Škálování dokumentových procesů mimo interaktivní chat.

3. Typy AI a Model Execution

Cortex odděluje požadovanou schopnost od konkrétního modelu. Největší nebo nejdražší model není automaticky nejlepší; klasifikace, extrakce, vyhledávání a syntéza mají odlišné nároky.

- **LLM:** Syntéza, shrnutí, dialog. Flexibilní, ale variabilní výstup.
- **Menší jazykový nebo klasifikační model:** Třídění, extrakce polí. Nižší latence.
- **Embedding + reranker:** Sémantické vyhledávání. Vyšší relevance, ale citlivý na ACL a poisoning.
- **OCR / multimodální model:** Skeny, formuláře, tabulky. Kvalita závisí na obrazu.
- **Speech:** Přepis řeči. Vyžaduje souhlas a bezpečný handoff.
- **Agentní systém:** Vícekrokový plán a použití nástrojů. Pouze bounded autonomy: omezený cíl, kroky, čas, náklad, nástroje a schválení.

Modelový registr a routing: Každý endpoint má vlastníka, verzi, region, účel, povolenou klasifikaci dat a cenový profil. Routing vyhodnocuje minimální potřebnou schopnost, bezpečnostní profil, kvalitu na konkrétním datasetu a náklad. Fallback nesmí tiše změnit datový režim. Změna modelu je produkční změna a podléhá regresní evaluaci.

4. Data a znalostní vrstva (RAG)

Produkční RAG není jednorázové nahrání dokumentů do vektorové databáze. Je to řízený životní cyklus zdroje od schválení přes indexaci až po aktualizaci a odstranění.

- **Onboarding:** Vlastník, účel, klasifikace, právní titul a ACL. Zdroj bez vlastníka nevstupuje do produkčního indexu.
- **Ingest:** Antimalware, parsování, OCR. Chyba vstupu se oddělí od chyby modelu.
- **Chunking:** Dělení podle struktury, významu a revizí. Velikost chunku ovlivňuje recall i cenu.

- **Index:** Lexikální a vektorový index. Hybridní retrieval kombinuje přesné termíny a sémantickou podobnost.
- **Retrieval:** Filtry ACL, tenant, role, dokumentová revize. Nepovolený obsah se nesmí dostat do kontextu modelu.
- **Context build:** Deduplikace, omezení délky, pořadí a citace.
- **Odpověď:** Grounded prompt, strukturovaný výstup, citace a absence.
- **Freshness/delete:** Reindexace změn, mazání a propagace změny ACL.

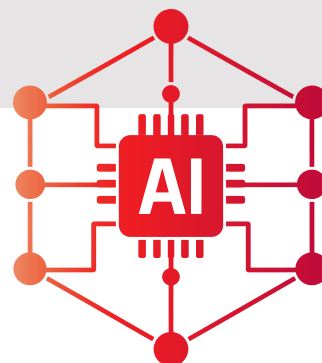
5. Integrace, API a nástrojový kontrakt



Integrace je hlavní rozdíl mezi izolovaným AI demem a podnikovou službou. Cortex podporuje více vzorů podle latence, objemu, transakčnosti a provozního vlastnictví.

- **REST + OpenAPI:** Synchronní čtení, validovaný zápis. Povinné ohraničení: timeout, retry, verze, autorizace.
- **Event / message:** Dlouhé workflow, dávky. Povinné ohraničení: deduplikace, pořadí, dead-letter queue.
- **Batch / file:** Velké dokumentové sady. Povinné ohraničení: manifest, checksum, checkpoint.
- **MCP (Model Context Protocol):** Standardizované vystavení resources a tools. Verze protokolu, scopes, token audience, schema a tool risk.
- **Database read model:** Přesné lookup. Read-only účet, row-level security, allowlist dotazů.
- **Webhook:** Notifikace o dokončení. Podpis, replay ochrana, retry.
- **MCP nuance:** MCP může sjednotit kontrakt zdrojů a nástrojů, ale není sám o sobě identity providerem, datovým katalogem ani bezpečnostní bránou. GEM pinuje podporovanou verzi, ověřuje token audience a odděluje read a write nástroje.

MCP nuance: MCP může sjednotit kontrakt zdrojů a nástrojů, ale není sám o sobě identity providerem, datovým katalogem ani bezpečnostní bránou. GEM pinuje podporovanou verzi, ověřuje token audience a odděluje read a write nástroje.



6. Identity, AI účty a správa tajemství

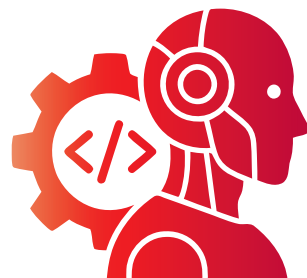
Identity-first model znamená, že Cortex neřeší bezpečnost až po vygenerování odpovědi. Uživatel nebo workload je rozpoznán na vstupu a jeho kontext se propaguje do policy, retrievalu i nástroje. Tím se snižuje vznik sdílených účtů, neznámých API klíčů a přístupů, které přežijí změnu role nebo odchod pracovníka.

- **Uživatel:** SSO přes podnikový IdP. Vlastní firemní účet, žádné sdílené osobní heslo.
- **Workload identity:** Samostatná identita backendu, agenta nebo konektoru. Oddělení člověka od služby.
- **Provider tenant/projekt:** Evidence organizace, projektu, regionu, vlastníka a fakturace.
- **API klíč / secret:** Reference do vaultu, rotace, zákaz uložení v kódu či promptu.
- **Servisní účet cílového systému:** Minimální scope, expiration, vlastník.

Joiner-Mover-Leaver (JML) a recertifikace: Nástup vytvoří jen potřebné role, změna pozice přepočítá oprávnění, odchod odebere session i delegace. Automatizace nesmí konsolidovat osobní hesla. Recertifikace se provádí podle rizika.

7. Bezpečnost, soukromí a governance

Bezpečnostní model kombinuje tradiční aplikační kontroly s riziky specifickými pro generativní AI. Referenční threat model vychází z OWASP Top 10 pro LLM.



- **Prompt injection:** Oddělení instrukcí a dat, sanitizace, tool policy, minimální scopes.
- **Sensitive information disclosure:** ACL před retrievalem, datový profil, DLP/redakce.
- **Supply chain:** Inventář, SBOM, skenování a schválení dodavatele.
- **Improper output handling:** Schéma, escaping, parametrizace a validace.
- **Excessive agency:** Bounded autonomy, budget, approval, kill switch.
- **Vector/embedding weaknesses:** Source approval, metadata, lineage, reindex.
- **Misinformation:** Citace, absence, kritická validace, lidský review.
- **Unbounded consumption:** Rate limit, token budget, timeout, kvóty.

Data profile a režim poskytovatele: Každý endpoint má schválený provider/data profile: účel, region, klasifikaci dat, retenci, použití pro trénování a subdodavatele. Nulovou retenci nebo zákaz trénování lze deklarovat pouze pro konkrétní službu a smluvní konfiguraci.

Governance a regulační souvislosti: Control vede AI system/use-case register, vlastníka, účel, modely, datasety, nástroje, rizika, změny a akceptační důkazy. Evidence může podpořit řízení podle NIST AI RMF, ISO/IEC 42001, GDPR, NIS2 a AI Act. Samotná platforma klienta automaticky necertifikuje; právní klasifikaci potvrzuje odpovědná organizace.

8. *Deployment modely a jejich trade-off*

Deployment se volí podle datového profilu, síťových omezení, provozních kompetencí, latence a ekonomiky. On-premise není automaticky bezpečnější ani levnější.

- **Customer cloud:** *Infrastruktura a služby v tenantovi klienta. Kontrola sítě, identity a nákladů.*
- **GEM Managed cloud:** *Služba provozovaná GEM. Rychlejší start a soustředěná provozní odpovědnost.*
- **Hybrid:** *Data či nástroje interně, inference cloudově. Kombinace interních systémů s pružnou kapacitou.*
- **On-prem / restricted:** *Lokální runtime, modely a data. Vyhoví omezené konektivitě a specifickému datovému režimu.*

Ekonomická souvislost: Lokální inference neplatí cloudové tokeny, ale spotřebovává hardware, energii, kapacitu týmu, licence, aktualizace a rezervu pro špičky. TCO se proto porovnává na přijatý výsledek a úroveň služby.



9. *AI FinOps, rozpočty a ekonomická telemetrie*

AI FinOps propojuje technickou spotřebu s vlastníkem, use casem a přijatým výsledkem. Levný model může vytvářet drahý rework; kvalitnější model může být ekonomičtější, pokud výstup častěji projde kontrolou.

- **Alokační klíč:** Tenant, projekt, cost center, use case, prostředí, model, tým.
- **Rozpočet:** Měsíční soft budget, alert, schvalovací práh a hard stop. Hard stop je bezpečný tam, kde Cortex řídí veškerý provoz.
- **Jednotková ekonomika:** Cena za dokument, dotaz, workflow nebo přijatý draft.
- **Value ledger:** Baseline, použitý objem, míra přijetí, časová změna, rework, TCO.

Řízení rozpočtu: Cortex může před voláním odhadnout velikost vstupu a zkontrolovat budget. Odhad a finální faktura se mohou lišit kvůli cache, dávkovým slevám nebo opožděným datům; finanční uzávěrka provádí reconciliaci. Hard limit odmítne další požadavky; rate limit chrání kapacitu.

Optimalizační páky: Deterministická cesta, model routing, kontext, cache, batch/asynchronní režim. Optimalizace se přijímá pouze tehdy, pokud nesníží akceptační metriku pod stanovený práh.

10. Nefunkční požadavky, SLO a odolnost

Jedna univerzální dostupnost pro celý Cortex by byla zavádějící. Přímý databázový lookup, interaktivní RAG a dávka mají jiné charakteristiky. SLO se stanovuje po runtime cestě a kritičnosti.

- **Dostupnost:** Podíl úspěšně obslužených požadavků.
- **Latence:** P50/P95/P99, time-to-first-token. Měřit odděleně retrieval, model a nástroj.
- **Propustnost:** Souběžné požadavky, dokumenty za hodinu.
- **Kvalita:** Kritická chybovost, groundedness, přijetí draftu.
- **Freshness:** Zpoždění synchronizace a odstranění zdroje.
- **Obnova:** RTO, RPO, záloha konfigurace a indexu.
- **Kapacitní degradace:** Fallback model, fronta. Degradace nesmí změnit datový profil.

Portabilita a exit: Produkční návrh obsahuje řízený odchod: export konfigurace, promptů, evaluačních sad, auditních záznamů a nákladových map. Zároveň popíše, co exportovat nelze kvůli licenci poskytovatele.



11. *Observabilita, změnové řízení a provoz*

Provoz AI služby vyžaduje spojit klasickou observabilitu s kvalitou výstupů a náklady. Jedno korelační ID sleduje požadavek přes gateway, retrieval, model, nástroj i cílovou aplikaci.

- **Metriky:** Požadavky, chyby, latence, tokeny, náklad, cache, fronta, tool success.
- **Trace:** Kroky orchestrace, verze modelu/promptu, rozhodnutí routeru.
- **Log:** Audit přístupu, schválení, změny konfigurace.
- **Eval:** Online feedback, golden set, critical errors.
- **Business event:** Přijatý draft, dokončený případ, rework.

Změnové řízení: Změna modelu, promptu, dat nebo nástroje vyžaduje: vlastníka, offline evaluaci, bezpečnostní review, staged release, schválení a post-release monitoring.

Incidenty a runbooky: Dostupnostní, kvalitativní, přístupový, agentní a finanční incident.

Referenční RACI: Business outcome a proces: Klient (Accountable), GEM (Responsible).
Klasifikace dat: Klient. Aplikační vrstva: GEM. Modelový endpoint: Smluvní provozovatel.
Kvalita: GEM. Incident: Service owner.

12. *Implementační cesta a akceptace*

Implementace postupuje od rozhodnutí, zda má use case smysl, až po důkaz, že jej lze bezpečně provozovat.

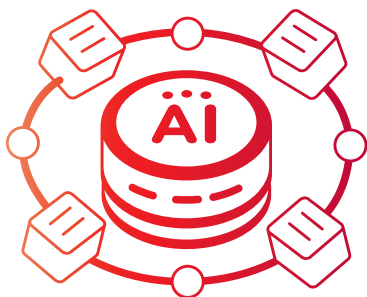
- **0. Qualification:** Business problém, sponsor, datová dostupnost, riziko.
- **1. Discovery:** Baseline, data-flow, integrace, eval dataset.
- **2. Pilot:** Omezený scope, draft/read-only, měření proti baseline.
- **3. Limited production:** Reálné identity, monitoring, support.
- **4. Scale:** Další use cases, automatizace governance.



Akceptační balíček: Funkční, kvalitativní, bezpečnostní, provozní, ekonomická a předávací acceptance. Úroveň autonomie se schvaluje samostatně podle dopadu, nikoli podle průměrné přesnosti.

13. *Limity, předpoklady a rozhodovací checklist*

- **Kvalita vstupů:** Nekvalitní scan a rozporné dokumenty omezí přesnost.
- **Oprávnění zdrojů:** Konektor bez ACL nelze vydávat za permission-aware.
- **Provider/API:** Funkce, limity a ceny se mění; adaptéry snižují, ale neruší dopad.
- **Modelová variabilita:** Stejný vstup nemusí vždy vytvořit totožný text.
- **Autonomie:** Nevratné a právně významné kroky zůstávají pod lidskou kontrolou.
- **Legacy integrace:** Bez dokumentace může integrace určovat většinu pracnosti.
- **Offline/on-prem:** Vyžaduje kapacitu, patching a model lifecycle.
- **ROI:** Ušetřený čas se realizuje až změnou procesu a kapacity.



Solution design checklist:

- Je definována úloha a baseline?*
- Je znám datový tok a klasifikace?*
- Jsou zápisy odděleny od čtení?*
- Existují negativní testy?*
- Je znám SLO profil a exit?*

