LLM in agenti brez iluzij: priložnosti, omejitve in uporaba v praksi

doc. dr. Grega Vrbančič
Fakulteta za elektrotehniko, računalništvo in informatiko, Univerza v Mariboru
2. julij 2026

REPUBLIKA SLOVENIJA
MINISTRSTVO ZA DIGITALNO PREOBRAZBO

REPUBLIKA SLOVENIJA
MINISTRSTVO ZA VISOKO ŠOLSTVO, ZNANOST IN INOVACIJE

REPUBLIKA SLOVENIJA
MINISTRSTVO ZA GOSPODARSTVO, TURIZEM IN ŠPORT

REPUBLIKA SLOVENIJA
MINISTRSTVO ZA OBRAMBO

ARIS Javna agencija za znanstvenoraziskovalno in inovacijsko dejavnost Republike Slovenije

Zakaj ta webinar?

Umetna inteligenca (UI) je na različne načine v veliki meri prisotna v raznovrstnih organizacijah.

Toda uporaba UI v organizaciji še ne pomeni:

- dodane poslovne vrednosti,
- produkcijske zrelosti,
- zanesljivosti delovanja,
- varne uvedbe

Problem ni samo ali uporabiti UI ali ne, ampak kako izbrati pravi pristop, upravljati tveganja ter meriti učinek.

Visoka stopnja sprejetja, nizka zrelost

88 %

organizacij redno uporablja UI v vsaj eni poslovni funkciji.

McKinsey State of AI 2025

>40 %

agentnih UI projektov preklicanih ali zmanjšanih do konca 2027.

Gartner 2025

Visoka stopnja sprejetja + veliko neuspešnih projektov = potreba po boljšem odločanju

Zakaj uporaba UI lahko „boli“?

Napačna izbira pristopa ni samo tehnična napaka. Je poslovno, varnostno in operativno tveganje.

Napačen odgovor stranki
Podporni chatbot obljubi neobstoječo politiko.
→ *Air Canada, 2022*

Uhajanje podatkov
Zaposleni vnese interno kodo v javni LLM.
→ *Samsung, 2023*

“Odlična” ponudba
Manipulacija je chatbot pripravila do navideznega soglasja k prodaji avta za 1 $ – avto ni bil dejansko prodan.
→ *Chevrolet, 2023*

Pomočnik podjetnikom
Podporni chatbot poda najemodajalcu napačno informacijo.
→ *NYC, 2023*

Izmišljeni viri
LLM navede neobstoječe pravne primere in predstavlja prvi večji sodni spor, ki je razkril „halucinacije“ UI.
→ *Mata v. Avianca, 2023*

Napačna akcija
Agent kljub izrecnim navodilom o zamrznitvi kode izbriše produkcijsko bazo podatkov.
→ *Replit, 2025*

Kibernetski napad
Napadalci zlorabijo Claude Code za avtomatizacijo napada na mehiške institucije (~195 mio zapisov, ~150 GB).
→ *Mexico / Claude Code, 2026*

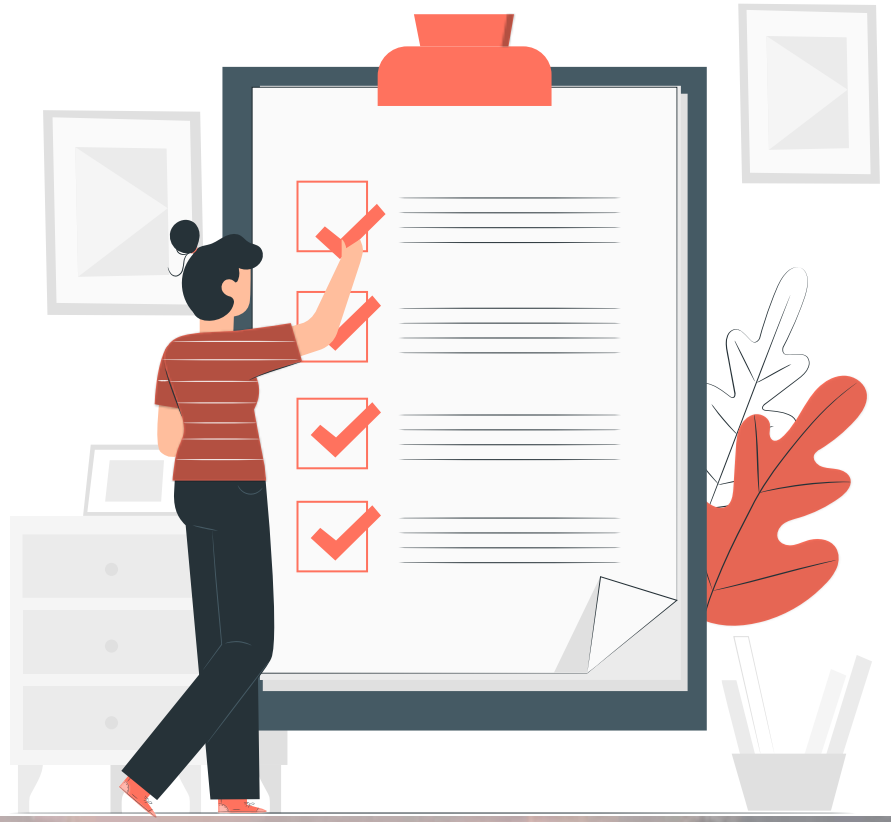
Izpraznitev proračuna
Agent pri skeniranju omrežja sproži obsežne AWS vire in ustvari račun ~6.531 USD.
→ *DN42 / AWS budget, 2026*

Napačna uporaba UI → napačno tveganje

Cilj ni uporabiti najbolj napredne tehnologije, ampak uporabiti primerno tehnologijo za konkretno nalogo in obvladovati pravo tveganje .

Agenda

1. Kaj LLM zmorejo danes
2. LLM, RAG, delovni tokovi in agentni sistemi
3. Tipični primeri uporabe v organizacijah
4. Kdaj je agentni pristop smiselen
5. Pogosti načini odpovedi



1. Kaj LLM danes zmorejo?

In zakaj samozavesten odgovor ni nujno pravilen.

Jezik in besedilo

prevajanje, povzemanje, osnutki

Koda

generiranje, debugging, dokumentacija

Analiza in raziskovanje

ekstrakcija, klasifikacija, iskanje

Kaj je LLM?

Veliki jezikovni model (Large Language Model, LLM) je globoko nevronska mreža (arhitektura transformer), naučeno na ogromnih količinah besedila. Na podlagi statističnih vzorcev napoveduje najverjetnejše naslednje besede v zaporedju in tako generira naravni jezik.

Transformer arhitektura

Globoka nevronska mreža, naučena na milijardah parametrov iz besedil, kode in podatkov.

Napoveduje naslednji token

Ne "razume" pomena kot človek – statistično ocenjuje najverjetnejše nadaljevanje besedila.

Generira naravni jezik

Omogoča pogovor, povzemanje, prevajanje ter pisanje besedil in programske kode.

Poznani primeri modelov:

Claude deepseek Gemini Mistral AI OpenAI Qwen

Kaj LLM počne dobro?

- ✓ **Povzemanje** – dolgi dokumenti, zapisniki, članki
- ✓ **Generiranje osnutkov** – e-pošta, ponudbe, poročila
- ✓ **Prevajanje** – vključno s tehničnimi jeziki
- ✓ **Klasifikacija** – kategorizacija sporočil, zahtevkov
- ✓ **Ekstrakcija podatkov** – iz nestrukturiranih virov
- ✓ **Koda** – generiranje, debugging, dokumentacija

Kje LLM odpove?

- **Dejstva brez konteksta** – ne ve, kaj velja danes
- **Matematika in logika** – ni kalkulator
- **Interne podatke** – nima dostopa do vaše baze
- **Konsistentnost** – isti prompt, različni odgovori
- **Akcije v zunanjih sistemih** – sam ne pošlje e-pošte

Ampak poznamo množico uspešnih rešitev...

shopify

Umetno inteligenco so vgradili v platformo v obliki asistenta Sidekick in sistemov za strojno prodajo. Trgovci z enim stavkom urejajo popuste in opise izdelkov.

15x - rast UI spodbujenih naročil (2025→2026)

2x - konverzija pri UI-prilagojenih izdelkih

Notion

Razvit Notion AI Agent deluje kot avtonomni raziskovalec nad celotno bazo vnesenih podatkov podjetja. Zaposleni v naravnem jeziku poizvedujejo po podatkih.

500 mio $ - letni prihodki po lansiranju agenta

50 %+ podjetij Fortune 500 uporablja Notion

Ampak poznamo množico uspešnih rešitev...

Harvey

Podjetje razvilo platformo (z uporabo modelov OpenAI in Anthropic), ki je prilagojeno pravnim pisarnam in pravnim oddelkom podjetij.

55 % - temeljno orodje

89 % - lahko prevzamejo več dela

77 % - boljši (pravni/komercialni) izidi

Morgan Stanley

Razvoj interne ”super aplikacije". Uporabili 350.000 strani lastnih finančnih raziskav, pravilnikov in naložbenih podatkov. Finančni svetovalci jo uporabljajo za takojšnje preverjanje zapletenih tržnih pozicij in predpisov med sestanki s strankami.

98 % - uporaba (adoption rate) med svetovalci

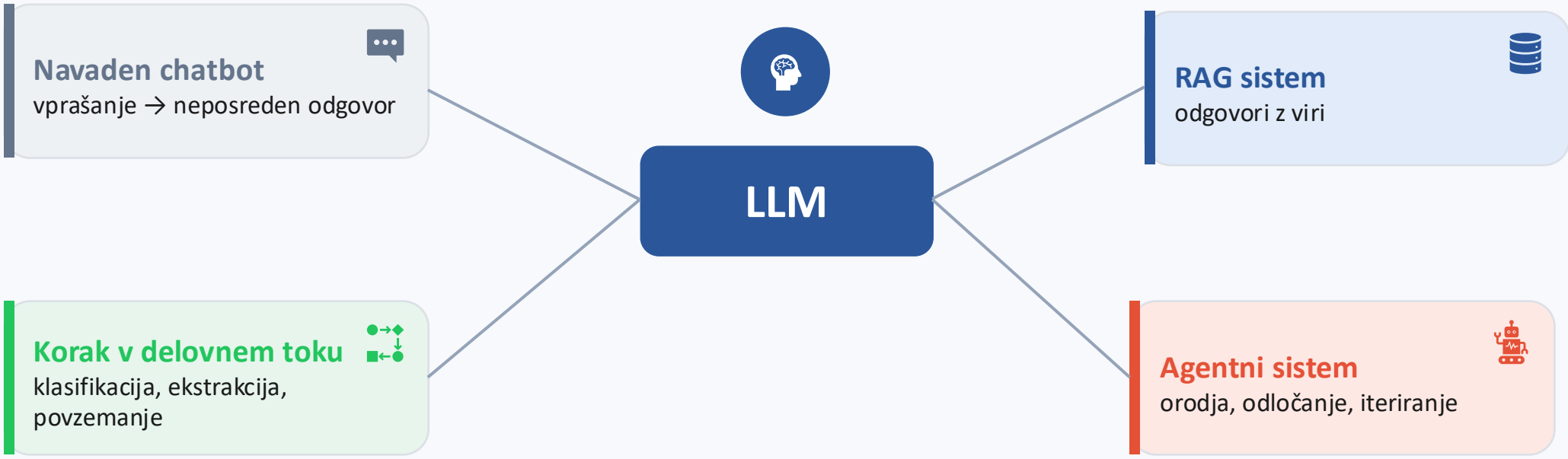
80 % - dostopnost/uporaba znanja

Gre za kompleksne rešitve

- ***LLM*** *– jedro, ki interpretira vhod in generira besedilo*
- ***RAG in interni podatki*** *– povezava z dokumenti in bazami podjetja*
- ***Orkestracija in orodja*** *– veriženje korakov v delovne tokove*
- ***Nadzor in varovalke*** *– preverjanje izhodov ter spremljanje (človek v zanki)*

Še vedno LLM, a različna arhitektura sistema

Model je lahko enak, arhitektura sistema, potencialna tveganja in način nadzora pa so različni.



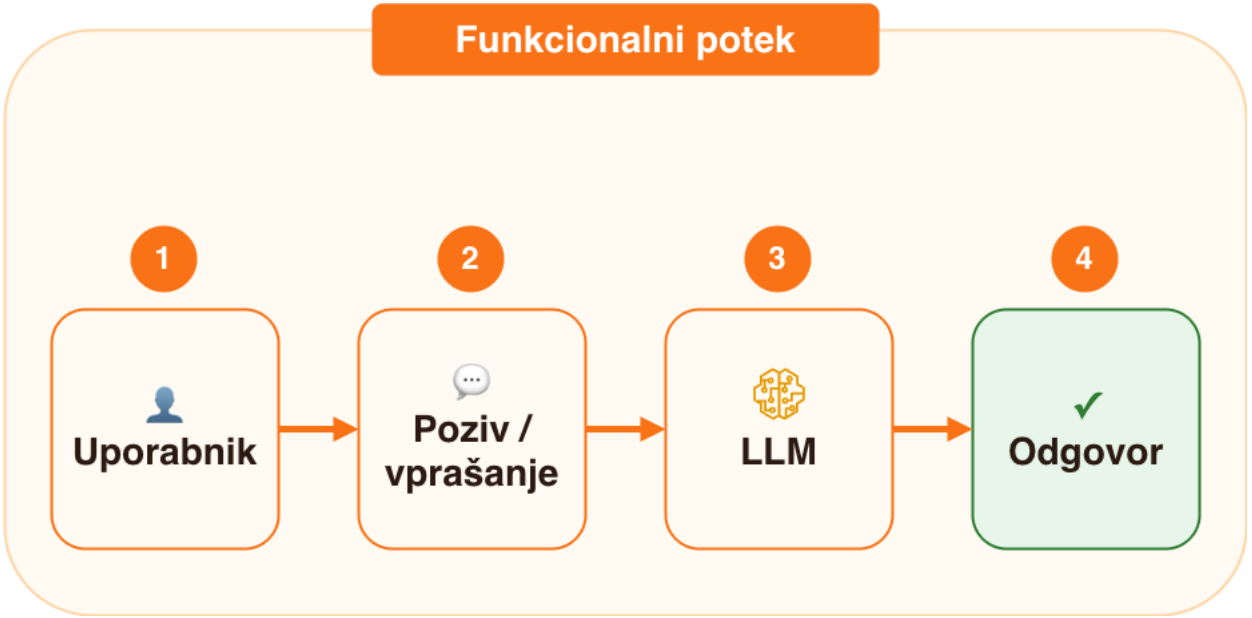
KCUI

Model ni sistem.

2. LLM, RAG, delovni tokovi in agentni sistemi.

Štirje pristopi. Štirje nabori tveganj.

Pristop 1: Osnovni LLM



Kdaj zadostuje:

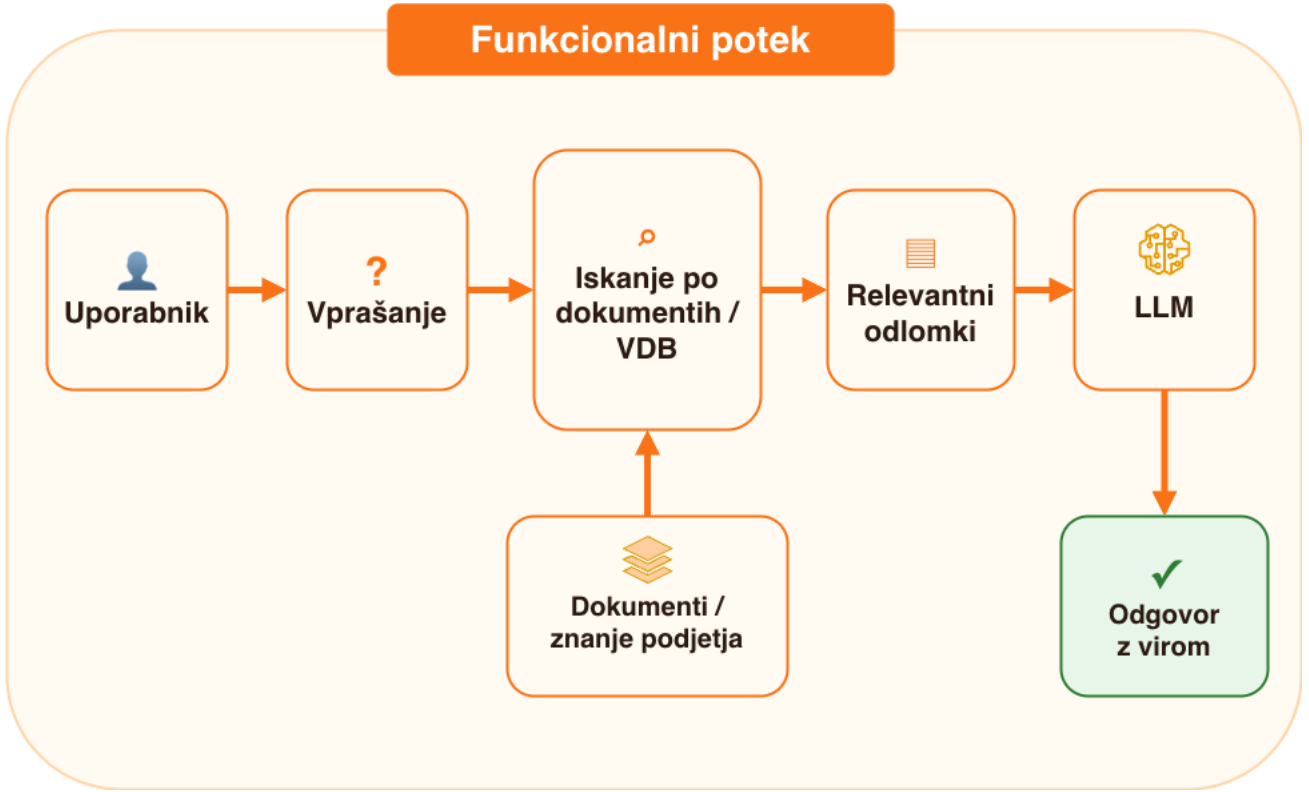
- ✓ Kreativno pisanje, osnutki e-pošte
- ✓ Splošno povzemanje javno dostopnih besedil
- ✓ Brainstorming, prevajanje, formatiranje

Omejitev: Ne ve nič o vaših internih podatkih. Halucinira pri specifičnih dejstvih.

TruthfulQA je pokazal, da večji modeli niso nujno bolj resnicoljubni. Najboljši model v študiji je bil „resnicoljuben“ na 58 % vprašanj, človeška izhodiščna uspešnost pa je bila 94 %.

Pristop 2: RAG – Retrieval Augmented Generation

S poizvedovanjem obogateno generiranje (angl. Retrieval Augmented Generation, RAG)



Kdaj zadostuje:

- ✓ Poizvedovanje po internih dokumentih
- ✓ Navajanje virov pri odgovorih – omilitev halucinacij
- ✓ Preverljivost – uporabnik vidi vir odgovora

RAG ni protihalucinacijsko stikalo

RAG ne odpravi halucinacij, jih zgolj omili.

Pogosti scenariji odpovedi:

- Iskalnik vrne napačen dokument → LLM odgovori samozavestno
- Kontekstno okno je polno, a brez relevantnega odlomka
- LLM poveže podatke iz različnih dokumentov na napačen način

Pomembno: RAG potrebuje testiranje, evalvacijo in spremljanje, enako kot vsak drug sistem.

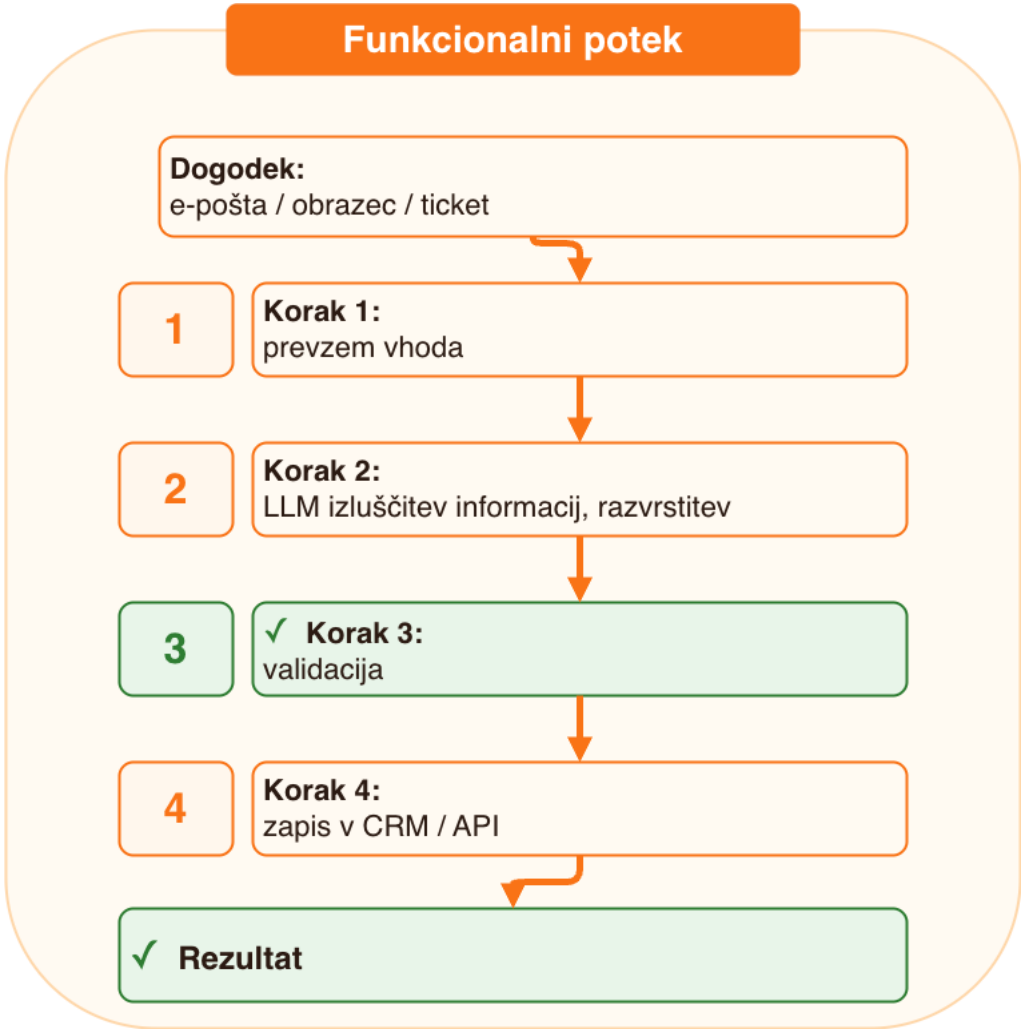
RAG ne naslovi:

- ✕ Napačne interpretacije konteksta
- ✕ Slabih podatkov (garbage in, garbage out)
- ✕ Logičnega sklepanja čez več dokumentov

- Manjkajoča vsebina
- Manjkajoči najbolj pomembni odlomki
- Ni v kontekstu
- Ni izluščeno
- Napačen format
- Neprimerna specifičnost
- Nepopolni odgovori

Barnett et al. (2024), "Seven Failure Points When Engineering a RAG System"

Pristop 3: Delovni tok (workflow)



Kdaj zadostuje:

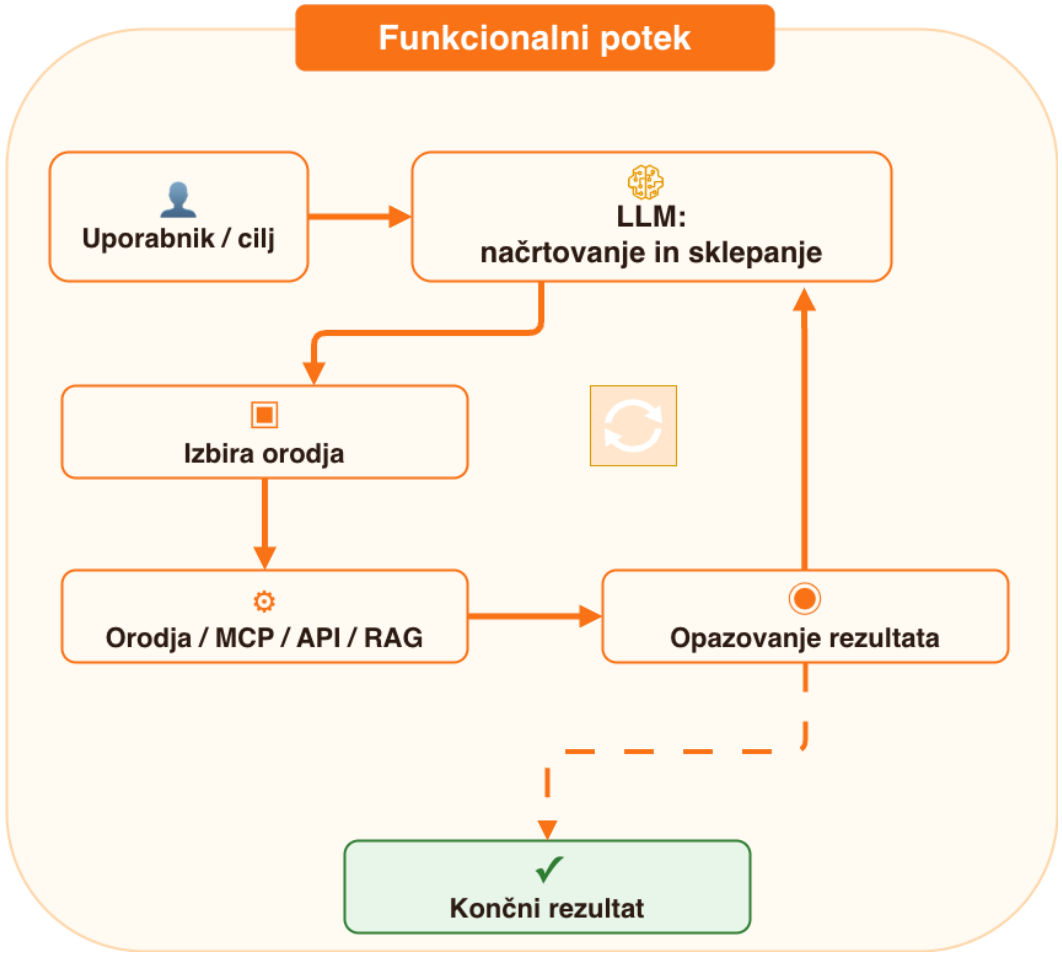
- ✓ Ustaljeni nadzorovani poslovni procesi
- ✓ Koraki so znani vnaprej in ponovljivi
- ✓ UI ne izbira naslednjega koraka
- ✓ Cilj zanesljivost, ne prilagodljivost

Omejitev: UI je komponenta v procesu in ne avtonomni odločevalec.

Primeri platform:

zapier make n8n

Pristop 4: Agent



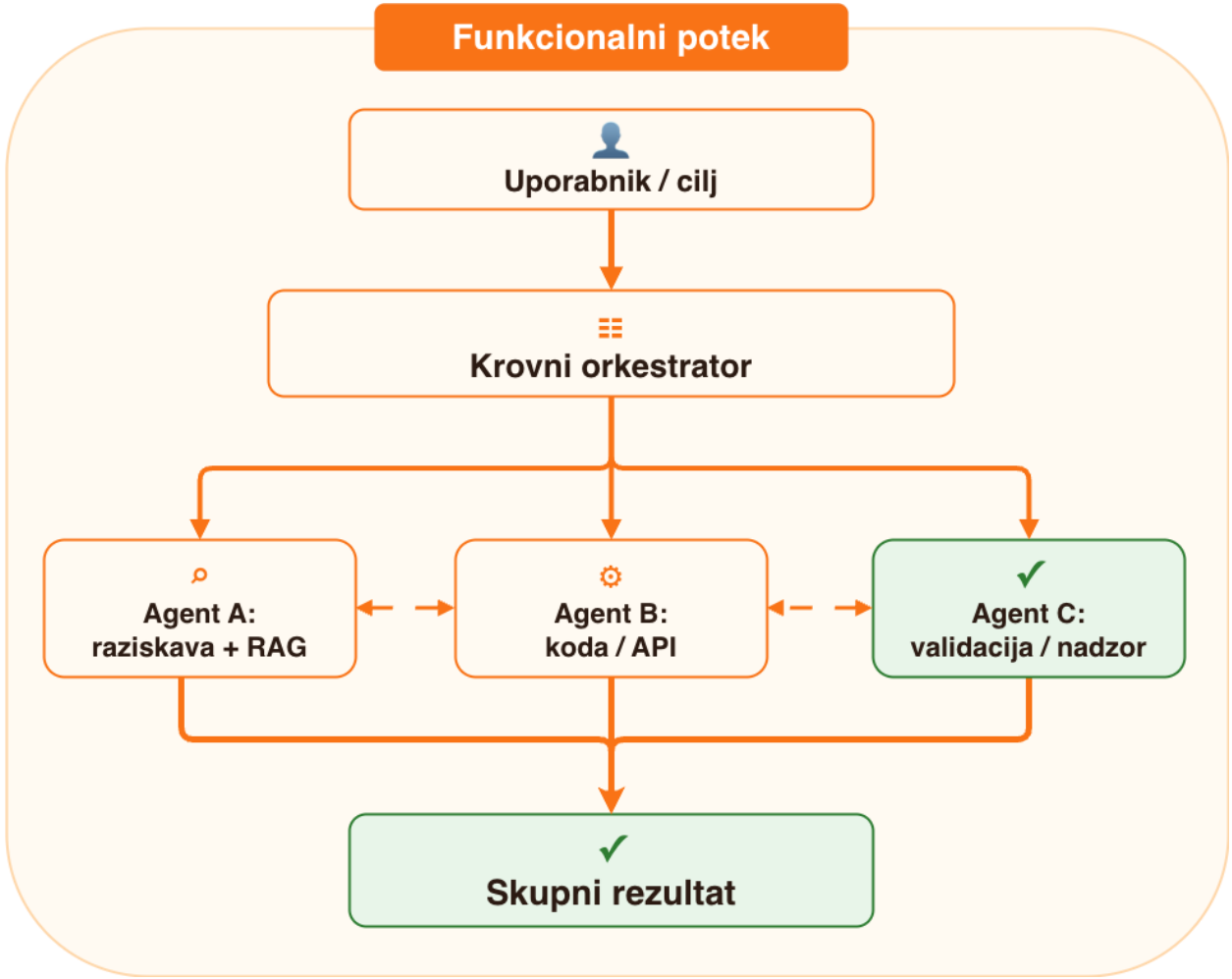
Kdaj zadostuje:

- ✓ Pot do cilja ni vnaprej znana
- ✓ Model mora sam izbrati naslednji korak/naloga zahteva uporabo orodij
- ✓ Ko imamo na voljo varne integracije z orodji kot so MCP, API in podobno.
- ✓ Napake so omejene, zaznavne in popravljive.

Opozorila:

- Vzpostavljene naj bodo omejitve: maks. koraki, timeout, budget, človek-v-zanki
- Imeti je potrebno mehanizem za ponastavitev stanja
- Možnost, da človek agenta ustavi/dopolni/popravi.

Kaj je agentni sistem?



Kdaj zadostuje:

- ✓ Naloga zahteva več različnih specializacij
- ✓ Posamezni koraki so prekompleksni za enega agenta
- ✓ Enostavnejši pristopi ne zadoščajo

Opozorila:

- Vse kar velja za agente
- potrebna sta nadzor in usklajevanje vlog

Kompleksnejši sistem, ki ga je težje testirati in je smiseln šele, ko enostavnejši pristopi ne zadoščajo.

Kako se pristopi razlikujejo/nadgrajujejo?

Pristop	Kaj dodamo?	Kaj rešuje?	Kaj se poveča?
LLM	jezikovno zmožnost	besedilne naloge	napačen odgovor
LLM + validacija	preverjanje izhoda	strukturirani izhodi	več zanesljivosti
RAG	zunanji vir znanja	dokumenti / viri	kompleksnost podatkov
Delovni tok	nadzorovan tok	znani koraki	orkestracija
Delovni tok z LLM/RAG	LLM/RAG kot korake	več korakov + viri	testiranje integracij
Agent	izbiro korakov + orodja	pot ni vnaprej znana	avtonomija in tveganje

3. Tipični primeri uporabe v organizacijah.

Za vsak primer, pravi pristop.

Ko zadostuje osnovni LLM

Primeri:

- **Marketing:** osnutki za kampanjo, ki jih ekipa vedno pregleda in dopolni.
- **Notranja komunikacija:** povzetki sestankov in dolgih dokumentov za interno rabo.
- **Jezik in slog:** prevajanje ter slogovno urejanje internih besedil.
- **Ustvarjalnost:** iskanje idej za imena, gesla in naslove kampanj.

Kdaj (v splošnem) izbrati osnovni LLM:

- ✓ Ni potrebe po internih podatkih
- ✓ Človek vedno pregleda in popravi izhod
- ✓ Cilj je produktivnost posameznika, ne avtomatizacija procesa

Ko potrebujemo RAG

Primeri:

- **Pravna služba:** iskanje po internih pogodbah in odgovarjanje z navedbo vira.
- **Kadrovska služba:** odgovori zaposlenim o pravilnikih in kadrovskih politikah.
- **Podpora strankam:** odgovori na podlagi dokumentacije izdelkov in navodil.
- **Tehnična ekipa:** iskalnik po tehnični dokumentaciji in bazah znanja.

Kdaj (v splošnem) izbrati RAG:

- ✓ Odgovori morajo temeljiti na internih dokumentih
- ✓ Uporabnik potrebuje vir za preveritev
- ✓ Gre za pomoč pri iskanju, ne za avtomatizacijo odločanja

Ko je bolj primeren delovni tok

Primeri:

- **Prodaja:** obdelava vhodne pošte in samodejna uvrstitev v CRM.
- **Finance:** preverjanje in odobritev računov po vnaprej določenih pravilih.
- **Kadri:** onboarding novih zaposlenih – zaporedje obvestil, dokumentov in nalog.
- **Skladnost:** preverjanje dokumentov glede na standardne predloge pred oddajo.

Kdaj (v splošnem) izbrati delovni tok:

- ✓ Koraki so vnaprej znani in predvidljivi
- ✓ Potreba po revizijskem sledu in ponovljivosti
- ✓ Regulirana panoga zahteva sledljivost

Ko pride v poštev agent/agentni sistem

Primeri:

- **Nabava:** spremljanje dobavne verige in samodejno naročanje ob odstopanjih.
- **Raziskave:** agent samostojno zbira podatke iz več virov in sestavi poročilo.
- **IT podpora:** agent diagnosticira težavo in izvede popravek v več sistemih.
- **Projektno vodenje:** dodeljevanje nalog in prilagajanje urnika glede na napredek.

Kdaj (v splošnem) izbrati agenta:

- ✓ Koraki niso vnaprej znani – odvisni od konteksta
- ✓ Potreba po več orodjih in API-jih hkrati
- ✓ Sistem mora delovati proaktivno, ne reaktivno

Zakaj?

Ne obstaja ultimativno pravilen odgovor!

Zakaj (v splošnem) delovni tok?

- **Predvidljivost:** enak vhod vsakič da enak rezultat, brez prilagodljivega sklepanja modela.
- **Nadzor in sledljivost:** vsak korak je mogoče testirati in revidirati.
- **Cena in hitrost:** brez ponavljajočih se klicev LLM ali agenta je rešitev hitrejša in cenejša.

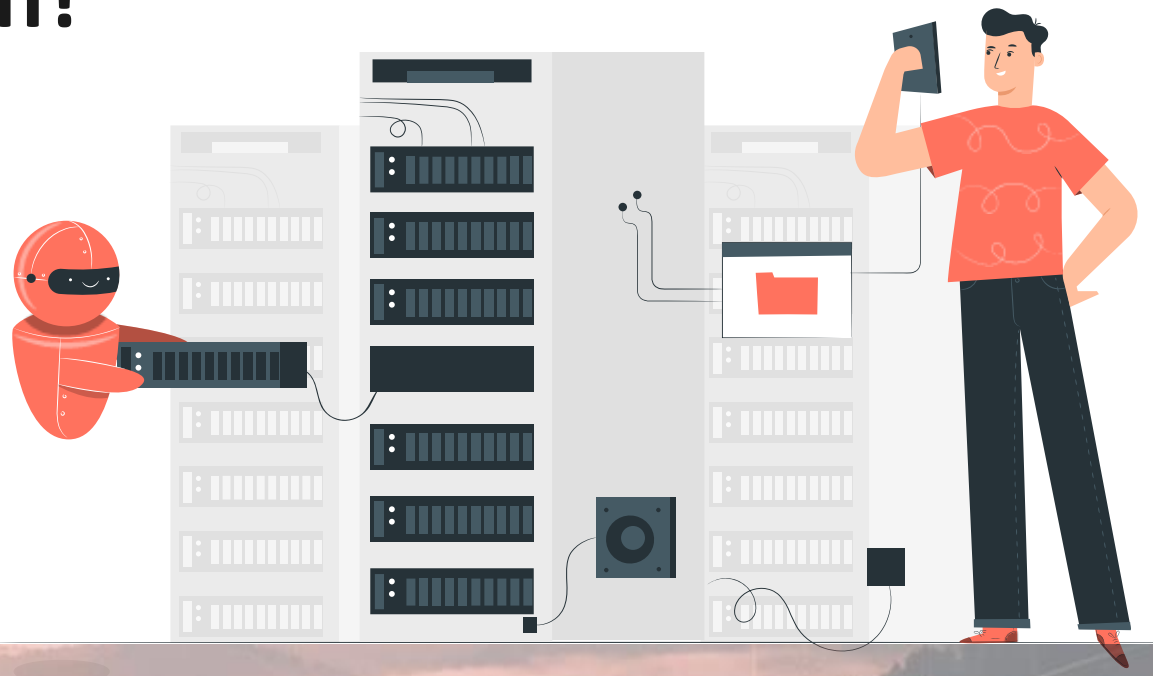
Zakaj ne drugi pristopi?

- **LLM:** ni strukture za primerjavo kandidatov in pozicij – proste, nekonsistentne ocene.
- **RAG:** iskanje po dokumentih ni dovolj, potrebna je logika obdelave in odločitvena pravila.
- **Agent:** ponovljiv, predvidljiv proces. V takšnem primeru bi agent bil pretiran s stališča kompleksnosti, stroškov in tveganja.

Ključna ugotovitev: ponovljiv proces z znanimi koraki → delovni tok, ne agent/agentni sistem.

4. Kdaj je agentni pristop smiseln?

Dva filtra in odločitveno drevo.



Filter 1: Avtomatizacijska zrelost

Preden uvedemo agenta – ali smo sploh pripravljeni?

Pred agentom preverite:

- ✓ Ali imate definirane poslovne procese?
- ✓ Ali imate čiste, strukturirane podatke?
- ✓ Ali imate obstoječe API-je in sisteme za integracijo?
- ✓ Ali imate ekipo, ki razume omejitve LLM?

Če je odgovor na katero koli vprašanje NE → začnite preprosteje.

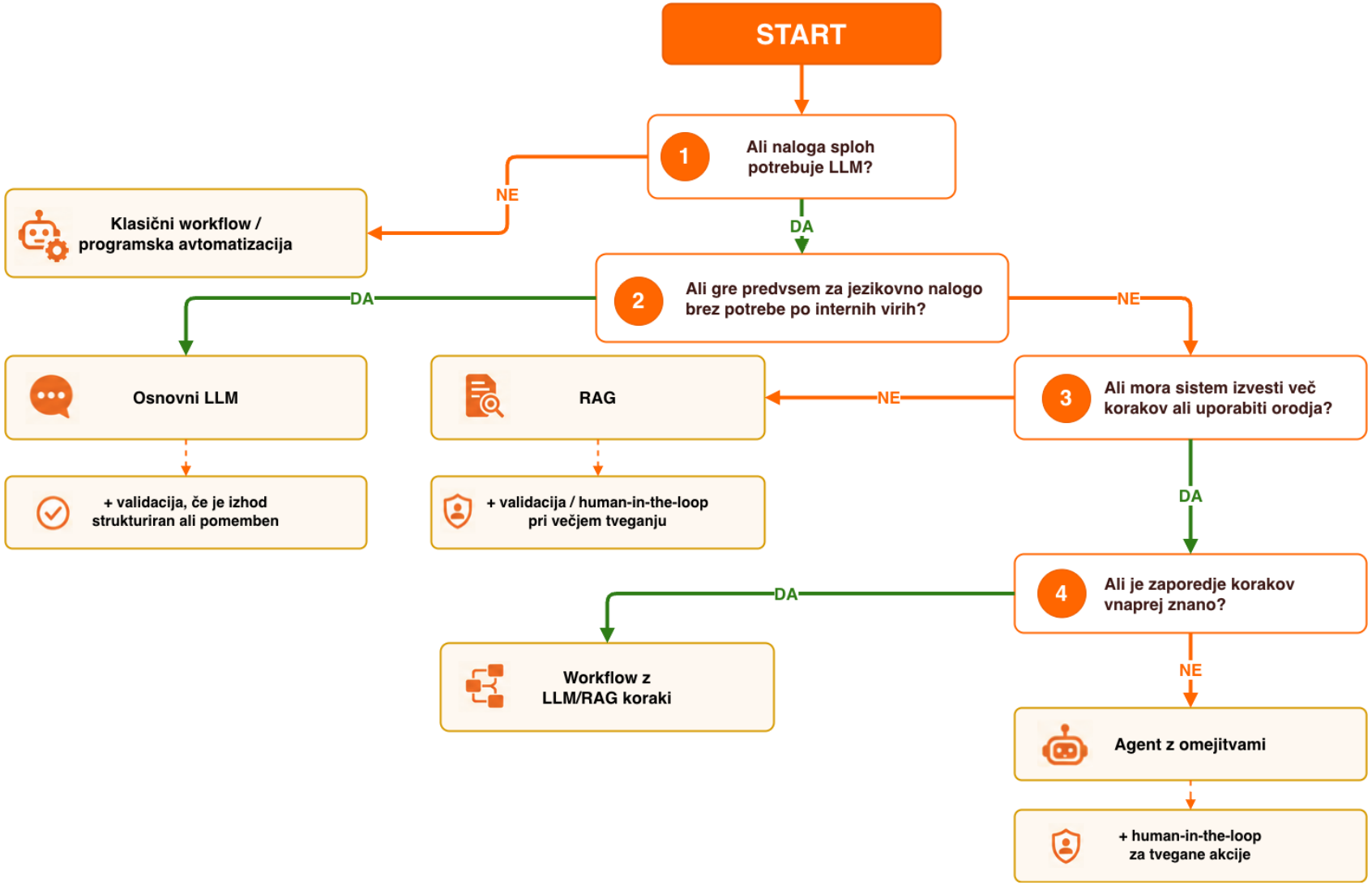
Filter 2: Ali res potrebujemo agenta?

Agent je smiseln le, ko:

- ✓ Koraki niso vnaprej znani
- ✓ Model mora med izvajanjem izbirati naslednji korak ali orodje
- ✓ Delovni tok ne pokriva zadostne variabilnosti
- ✓ Napake so omejene, zaznavne in popravljive
- ✓ ROI upravičuje dodatno kompleksnost in tveganje

Če ne izpolnjujete vseh štirih pogojev, agent verjetno ni pravi odgovor.

Odločitveno drevo: katero arhitekturo izbrati?



Ko agent ni pravi odgovor

Opozorilni znaki, da agent ni prava pot:

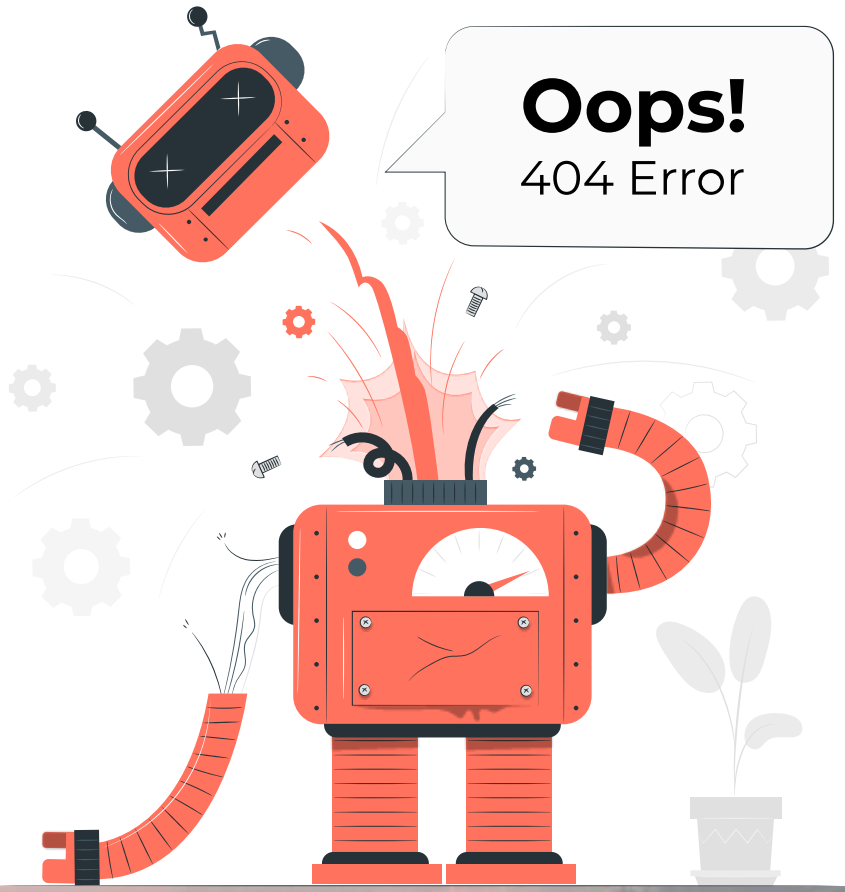
- ✕ Noben preprostejši pristop ni bil preizkušen
- ✕ Cilj je »AI transformacija« brez jasnega ROI
- ✕ Regulatorne zahteve ne dovoljujejo avtonomnih odločitev
- ✕ Podatki niso čisti ali pa sistemi niso integrirani

Namesto agenta:

- začnite z delovnim tokom,
- dodajte LLM korake in postopno razširjajte.

5. Pogosti načini odpovedi

Realni primeri odpovedi in zakaj implementacija varovalk ni opcijska.



Pogosti načini odpovedi

Spomnimo…

Napačen odgovor stranki
Podporni chatbot obljubi neobstoječo politiko.
→ *Air Canada, 2022*

Uhajanje podatkov
Zaposleni vnese interno kodo v javni LLM.
→ *Samsung, 2023*

"Odlična" ponudba
Manipulacija je chatbot pripravila do navideznega soglasja k prodaji avta za 1 $ – avto ni bil dejansko prodan.
→ *Chevrolet, 2023*

Pomočnik podjetnikom
Podporni chatbot poda najemodajalcu napačno informacijo.
→ *NYC, 2023*

Izmišljeni viri
LLM navede neobstoječe pravne primere in predstavlja prvi večji sodni spor, ki je razkril „halucinacije" UI.
→ *Mata v. Avianca, 2023*

Napačna akcija
Agent kljub izrecnim navodilom o zamrznitvi kode izbriše produkcijsko bazo podatkov.
→ *Replit, 2025*

Kibernetski napad
Napadalci zlorabijo Claude Code za avtomatizacijo napada na mehiške institucije (~195 mio zapisov, ~150 GB).
→ *Mexico / Claude Code, 2026*

Izpraznitev proračuna
Agent pri skeniranju omrežja sproži obsežne AWS vire in ustvari račun ~6.531 USD.
→ *DN42 / AWS budget, 2026*

Prompt injection

Kaj je prompt injection?

Uporabnik ali zunanji vir vrine navodila v prompt, ki spremenijo vedenje sistema.

Dve obliki:

- **Neposredna:** uporabnik sam piše zlonamerni prompt
- **Posredna:** navodila so skrita v dokumentu, emailu ali spletni strani

100 % zaščita ne obstaja

- Tveganje lahko zmanjšamo z validacijo pozivov omejitvami in nadzorom.

Spomnimo: primer Chevrolet (2023) — kupec je z direktnim promptom pripravil prodajni chatbot, da „proda“ avto za 1 $ in to razglasil kot obojestranko zavezujočo ponudbo.

Ko sistem začne delovati drugače

Model drift – sprememba vedenja brez spremembe kode

- Ponudnik posodobi model → vaš sistem se začne obnašati drugače.
  - Empirično izmerjena velika sprememba vedenja GPT-4/GPT-3.5 med različicami v študiji avtorjev Chen, Zaharia, Zou (2023, Stanford/UC Berkeley)

Znaki:

- Odgovori so krajši ali daljši kot prej
- Sistem zavrača naloge, ki jih je prej opravljal
- Klasifikacija se začne obnašati nedosledno

Ukrepi:

- Redno testiranje evaluacijskim naborom vprašanj/primerov/ipd.
- Izbira fiksnih verzij LLM.
- Spremljanje delovanja
  - Beleženje pozivov, odgovorov
  - Kvalitativno ovrednotenje odgovorov

Šest osnovnih varovalk

Minimalni nabor varovalk za produkcijsko rabo:

1. **Obseg** – jasno definiramo kaj sistem sme in česa ne
2. **Validacija** – preverimo izhode pred izvedbo akcije
3. **Človeški nadzor** – človek-v-zanki za kritične odločitve
4. **Revizijska sled** – beležimo vse prompte, odzive in odločitve
5. **Testiranje** – evaluacijski set, stresni testi, red teaming
6. **Spremljanje** – uporabe, stroškov, kakovosti, anomalij

Ključno sporočilo

LLM ni čarobna palica – je orodje z jasnimi omejitvami

- Vedno izberite najpreprostejšo arhitekturo, ki zanesljivo reši problem
- Agent ni vedno odgovor za vse probleme
  - Marsikaj je izvedljivo z uporabo agentov/agentnih sistemov
  - Zavedati pa se je potrebno tudi tveganj, ki jih uvedba takšnega pristopa prinaša
- Implementirane varovalke niso izbira, so pogoj za produkcijsko rabo
- Začnite z majhnim, varnim pilotom in ne z velikim agentnim projektom

Najboljša UI strategija je preprosta, zanesljiva in nadzorovana.

KCUI

Univerza v Mariboru

Fakulteta za elektrotehniko, računalništvo in informatiko

doc. dr. Grega Vrbančič

Laboratorij za inteligentne sisteme

linkedin.com/in/gregavrbancic

grega.vrbancic@um.si

Področja delovanja:

- Aplikativna umetna inteligenca
- Inženirstvo inteligentnih sistemov
- Strojno učenje in globoko učenje
- Veliki jezikovni modeli, RAG in agentni sistemi
- MLOps in LLMOps
- Dolgoročno napovedovanje časovnih vrst
- Optimizacija in kompresija napovednih modelov
- Podatkovno rudarjenje in analitika podatkov

REPUBLIKA SLOVENIJA
MINISTRSTVO ZA DIGITALNO PREOBRAZBO

REPUBLIKA SLOVENIJA
MINISTRSTVO ZA VISOKO ŠOLSTVO, ZNANOST IN INOVACIJE

REPUBLIKA SLOVENIJA
MINISTRSTVO ZA GOSPODARSTVO, TURIZEM IN ŠPORT

REPUBLIKA SLOVENIJA
MINISTRSTVO ZA OBRAMBO

ARIS Javna agencija za znanstvenoraziskovalno in inovacijsko dejavnost Republike Slovenije