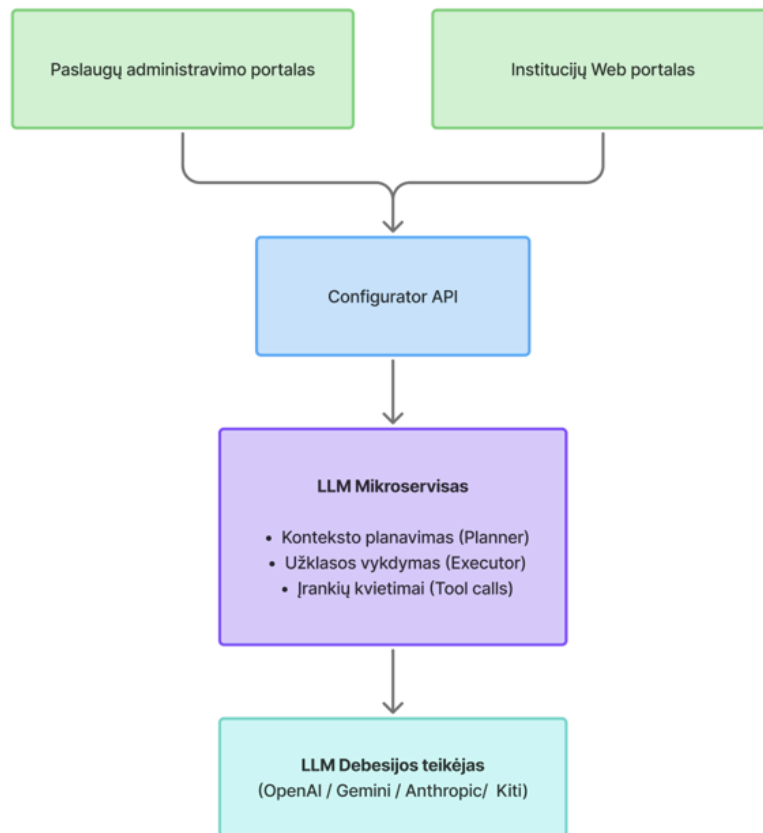


SPP – Dirbtinio intelekto Modulio Architektūros Planas

Architektūra

Aukšto lygio schema



Principai

- Web portalai tiesiogiai nekomunikuoja su LLM teikėju - viskas eina per vidinę infrastruktūrą.
- LLM Mikroservisas yra vienintelis komponentas, turintis prieigą prie išorinio LLM teikėjo.
- API raktai saugomi serverio aplinkoje, niekada nepasiekiami kliento pusėje.
- LLM teikėjas keičiamas per konfigūraciją naudojant bendrą abstrakcijos sluoksnį, be verslo logikos pakeitimų.
- Rezultatai grąžinami kaip struktūrizuoti įrankių kvietimai (tool calls), kuriuos sistema automatiškai pritaiko e-paslaugai.

Žinių bazė

LLM kontekstą sudaro SPP e-paslaugos DSL (Domain Specific Language) struktūra:

| Komponentas | Paskirtis |
|------------------------|--|
| Duomenų modelis | Laukų grupės, laukų tipai, validacijos, numatytosios reikšmės |
| Formos | Vizuali laukų prezentacija, įvedimo tipai, teksto blokai |
| Taisyklės | Sąlyginė logika (rodyti/slėpti/privalomi laukai pagal sąlygas) |
| Žingsniai | Proceso eiga, dalyviai, terminai, būsenų perėjimai |
| Įskiepai (Add-ons) | Išorinių servisų integracijos konfigūracija |
| Standartizuoti sąrašai | Bendrai naudojami pasirinkimų sąrašai |

Užklauso apdorojimo principas

Kiekviena vartotojo užklausa apdorojama dviem etapais:

Konteksto planavimas (Planner)

Prieš vykdant pagrindinę užklausą, greitas ir pigus modelis (pvz. Gemini Flash Lite, Claude Haiku) analizuoja vartotojo žinutę ir nusprendžia:

- Ar užklausa tinkama - jei vartotojas klausia nesusijusių dalykų, užklausa atmetama be pagrindinio modelio kvietimo.
- Kokio konteksto reikia - pagal užklauso pobūdį atrenkamos tik reikalingos e-paslaugos dalys (duomenų modelis, žingsniai, taisyklės, įskiepai, sąrašai). Tai optimizuoja tokenų naudojimą - nereikalingi duomenys nesiunčiami pagrindiniam modeliui.

Užklauso vykdymas (Executor)

Pagrindinis modelis (pvz. Gemini Pro, Claude Sonnet) gauna:

- Sisteminę instrukciją su pilnu DSL struktūros aprašymu
- Atrinktas e-paslaugos kontekstas (pagal Planner planą)
- Pokalbio istoriją
- Vartotojo dabartinės UI būsenos informaciją (kuris žingsnis atidarytas, koks redaktorius aktyvus)

Modelis atsako struktūrizuotais įrankių kvietimais (tool calls), kurie tiesiogiai modifikuoja e-paslaugos konfigūraciją:

| Įrankis | Paskirtis |
|------------------------------|--|
| <code>setDataModel</code> | Sukurti arba atnaujinti duomenų modelio elementą |
| <code>removeDataModel</code> | Pašalinti duomenų modelio elementą |
| <code>setStep</code> | Sukurti arba atnaujinti žingsnį (su formomis, įskiepais) |
| <code>removeStep</code> | Pašalinti žingsnį |
| <code>setRule</code> | Sukurti arba atnaujinti taisyklę |
| <code>removeRule</code> | Pašalinti taisyklę |

Šis metodas užtikrina, kad LLM atsakymai yra struktūrizuoti ir tiesiogiai pritaikomi sistemoje be papildomo rankinio JSON apdorojimo.

Išorinis modelis

Modelių naudojimo strategija

| Modelio tipas | Paskirtis | Pavyzdžiai |
|-------------------------|--|---------------------------------|
| Planner (greitas/pigus) | Konteksto planavimas, užklausos filtravimas | Gemini Flash Lite, Claude Haiku |
| Executor (galingas) | Pagrindinis modelis, struktūrų generavimas, tool calls | Gemini Pro, Claude Sonnet |

Numatomi / orientaciniai modeliai

| Teikėjas | Modelis | Stiprybės |
|-----------|-----------------------------|--|
| OpenAI | GPT-5.2 | 400K kontekstas, mažesnis haliucinacijų lygis pagal viešai prieinamus vertinimus |
| Google | Gemini 3 Pro | 1M tokenų kontekstas, pažangus protavimas (Deep Think) |
| Google | Gemini 3 Flash / Flash Lite | Greitas, pigus, tinka planner užduotims |
| Anthropic | Claude Opus 4 / Sonnet 4.5 | Ilgalaikės užduotys, programinio kodo generavimas, „extended thinking“ režimas |

Tokenų sąnaudos

Pastaba. Visi tokenų įvertinimai yra preliminarūs ir ženkliai priklauso nuo konkrečios e-paslaugos apimties (laukų, formų, taisyklių kiekio) bei vartotojo užklausos sudėtingumo. Tikslesnė apimtis bus aiški po DI modulio sukūrimo ir testavimo su realiomis e-paslaugomis.

Orientacinės tokenų sąnaudos pagal veiksmą

*Pastaba: Skaičiai pateikti **kaip architektūrinės prielaidos**, o ne SLA ar garantijos.*

| # | Veiksmas | Tokenai (preliminariai) |
|---|--|-------------------------|
| 1 | DM ir formų kūrimas iš nuotraukos | ~65 000 |
| 2 | Panašios e-paslaugos kūrimas pagal esamą | ~85 000 |
| 3 | Kūrimas pokalbių roboto pagalba | ~60 000 - 120 000 |
| 4 | Esamos e-paslaugos klaidų tikrinimas | ~50 000 |

Tipinės operacijos tokenų struktūra

*Pastaba: Skaičiai pateikti **kaip architektūrinės prielaidos**, o ne SLA ar garantijos.*

| Dalis | Tokenai (preliminariai) |
|---------------------------------------|-------------------------|
| E-paslaugos kontekstas (JSON) | ~40 000 |
| Sisteminės instrukcijos + schema | ~10 000 – 30 000 |
| Vartotojo užklausa | ~100 – 2 000 |
| LLM atsakymas (generuotas rezultatas) | ~15 000 – 40 000 |

Konteksto planavimo optimizacija

Dėl dviejų etapų architektūros (Planner + Executor) ne kiekviena užklausa siunčia pilną e-paslaugos kontekstą. Planner nusprendžia, kurios dalys reikalingos, todėl tipinė užklausa gali naudoti **30–70% mažiau tokenų** nei pilnas kontekstas.