AKADEMIA E FORCAVE
TË ARMATOSURA

Hyrje në Modele të Mëdha Gjuhësore

Agjentët

Dr. Fiorela Ciroku

Çfarë është një Agjent?

Agjentët janë sisteme inteligjente që përdorin modele gjuhësore për:

- ***të planifikuar,***
- ***të marrë vendime***
- ***të kryer veprime të shumta***

në mënyrë iteruese, duke ndërvepruar me mjedisin dhe mjete të jashtme për të arritur një objektiv të caktuar.

Agjentët në MMGj

Ndryshe nga prompting apo RAG, ku modeli përgjigjet një herë në mënyrë të kushtëzuar nga konteksti, një agjent:

- kryen vendimmarrje iteruese,
- përdor mjete të jashtme (API, kërkim, databaza, kalkulim),
- ruan gjendje (state),
- dhe optimizon progresin drejt një qëllimi.

Pse Agjentë me MMGj?

MMGj-të të vetme janë kryesisht “single-turn generators”. Shumë probleme kërkojnë:

- dekompozim të problemit,
- mbledhje evidencash nga burime të ndryshme,
- verifikim të rezultateve,
- dhe adaptim kur hasen mungesa informacioni.

Agjentët adresojnë këtë duke ofruar një cikël kontrolli, vendimmarrje të kushtëzuar nga gjendja, dhe integrim të drejtpërdrejtë me mjete.

Agjentë të njohur dhe funksionet e tyre

- ReAct - arsyetim + veprim iterativ
- AutoGPT / BabyAGI - ekzekutim autonom i detyrave komplekse
- Toolformer - përdorim mjetesh i mësuar nga modeli
- WebGPT - navigim dhe kërkim i asistuar në web
- Generative Agents - simulim sjelljeje dhe memorie afatgjatë
- Planner–Executor agents - ndarje planifikim / ekzekutim

Agjentë të njohur dhe funksionet e tyre

Agjent / Paradigmë	Karakteristikë Dalluese	Çfarë Problemi Zgjidh
ReAct	Alternon reasoning dhe tool-use në çdo hap	Detyra që kërkojnë verifikim hap-pas-hapi dhe reduktim halucinacionesh
AutoGPT / BabyAGI	Dekompozon qëllimin dhe vepron pa ndërhyrje të vazhdueshme	Automatizim i detyrave të gjata dhe multi-step
Toolformer	Tool-use i inkorporuar gjatë trajnimit	Reduktim i varësisë nga prompt-engineering për mjete
WebGPT	Agjent që përdor browser-in si mjedis	Pyetje që kërkojnë burime aktuale dhe citime
Generative Agents	Memorie episodike + planifikim afatgjatë	Koherencë dhe vazhdimësi sjelljeje në kohë
Planner–Executor Agents	Komponent i veçantë për plan dhe zbatim	Kontroll, auditim dhe siguri në sisteme kritike

Agent Loop: Cikli Observo- Planifiko - Vepro

Cikli agjentik është modeli standard i kontrollit:

- **Observe**: sistematizon inputin dhe rezultatet e mjeteve.
- **Plan**: MMGj propozon veprime (p.sh. “kërko protokollin X”).
- **Act**: ekzekutohet veprimi (tool call), pastaj kthehet rezultat.

Cikli përsëritet derisa të arrihet “stop condition”.

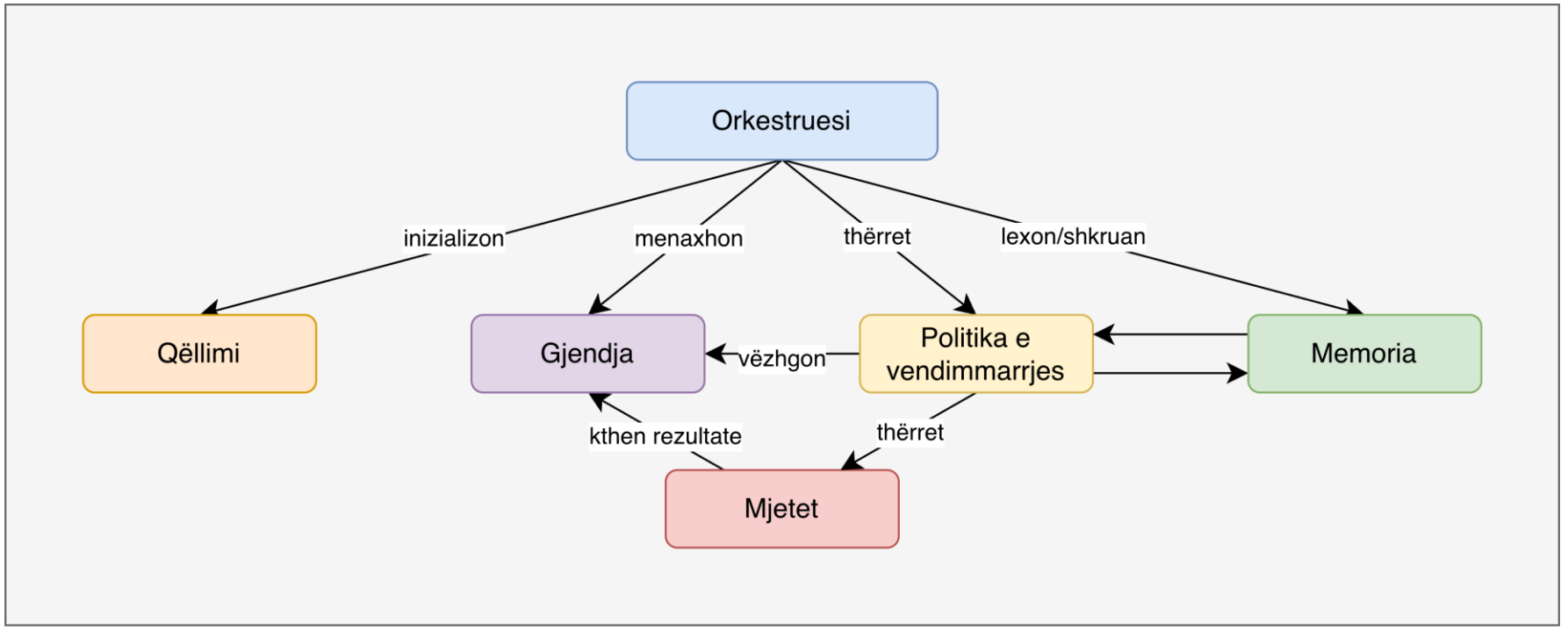
Ky loop është vendimtar sepse e bën sistemin adaptiv, të aftë për korrigjim,
dhe të strukturuar për auditim.

Komponentët bazë të një sistemi agjentik

Një agjent praktik nuk është vetëm MMGj, është një sistem me komponentë:

- ***Qëllimi (Goal)***: specifikim i qartë i detyrës (nga prompt ose sistem).
- ***Gjendja (State)***: çfarë dimë deri tani (fakte të mbledhura, rezultate, mjete).
- ***Politika e vendimmarrjes (LLM Policy)***: MMGj që vendos hapin e ardhshëm.
- ***Mjetet (Tools)***: funksione të jashtme (kërkim, DB, llogaritje).
- ***Memoria (Memory):*** ruajtje afatshkurtër/afatgjatë.
- ***Orkestruesi/Kontrolluesi (Orchestrator)***: logjikë që zbaton kufizime, riprova, limite.

Komponentët bazë të një sistemi agjentik



Qëllimi

Qëllimi është specifikimi formal i asaj që agjenti duhet të arrijë. Ai nuk është thjesht një pyetje, por një deklarim operacional i detyrës, që përfshin:

- çfarë duhet prodhuar,
- në çfarë forme,
- dhe kur detyra konsiderohet e përfunduar.

Një qëllim i paqartë çon në devijim të sjelljes (goal drift), veprime të panevojshme ose cikle të pafundme.

Në kontekstin e mbrojtjes civile, qëllimi duhet të jetë i qartë dhe i kufizuar, p.sh.:
"Gjenero një plan veprimi bazuar vetëm në protokollin zyrtar për përmbytje, me hapa të renditur dhe referenca."

Gjendja

Gjendja përfaqëson çfarë di agjenti deri në një moment të caktuar. Ajo përfshin:

- fakte të konfirmuara,
- rezultate të mjeteve,
- hapa të kryer,
- detyra të mbetura.

Meqë MMGj-të nuk ka memorie të përhershme, gjendja duhet të:

- ruhet në struktura të jashtme,
- jetë e serializueshme (p.sh. JSON),
- dhe e inspektueshme për auditim.

Pa gjendje të qartë, agjentët humbasin kontekst dhe bëhen jo-deterministë.

Gjendja dhe menaxhimi i saj

Në aplikime të mbrojtjes civile, gjendja mund të përfshijë:

- tip incidenti,
- lokacion,
- burime zyrtare të konsultuara,
- vendime të marra dhe arsyetimet.

```
{
  "$schema": "https://json-schema.org/draft/2020-12/schema"
  "title": "AgentState",
  "type": "object",
  "additionalProperties": false,
  "required": [
    "incident_type",
    "location",
    "evidence_ids",
    "actions_taken"
  ],
  "properties": {
    "incident_type": {
      "type": "string",
      "description": "Type of incident."
    },
    "location": {
      "type": "string",
      "description": "Incident location."
    },
    "evidence_ids": {
      "type": "array",
      "items": { "type": "string" },
      "description": "Identifiers of consulted evidence."
    },
    "actions_taken": {
      "type": "array",
      "items": { "type": "string" },
      "description": "Actions executed so far."
    }
  }
}
```

Politika e vendimmarrjes

Politika e vendimmarrjes është roli që luan MMGj në sistemin agjentik. MMGJ:

- analizon gjendjen aktuale,
- merr parasysh qëllimin,
- dhe prodhon një vendim: plan, veprim ose ndalim.

MMGj **nuk duhet të ketë autoritet ekzekutiv direkt**; ai vetëm propozon.

Ekzekutimi realizohet nga shtresa e orkestrimit.

Kjo ndarje është kritike për siguri dhe kontroll.

Mjetet

Mjetet janë komponentët që lejojnë agjentin të:

- ***kërkojë informacion,***
- ***kryejë llogaritje,***
- ***ose ndërveprojë me sisteme të tjera***.

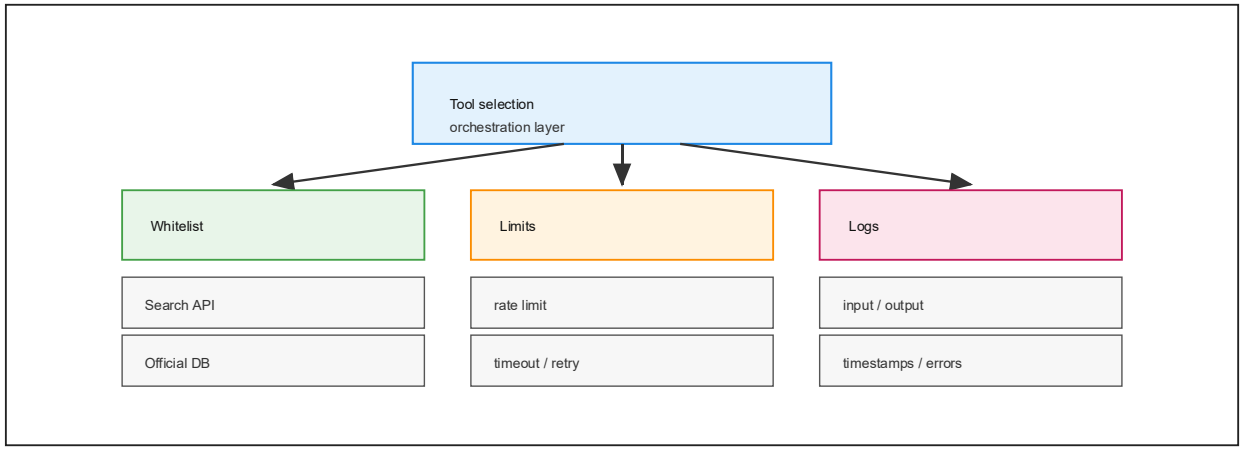
Shembuj: kërkim në databazë dokumentesh, API për gjeolokacion, verifikim inventari, etj.

Mjetet janë burim rreziku nëse nuk kufizohen. Prandaj përdoren whitelist, validim input-I dhe logging i detajuar.

Zgjedhja e mjeteve dhe kufizimi i veprimeve

Agjentët mund të dështojnë nëse:

- zgjedhin mjetin e gabuar,
- e thërrasin shumë herë,
- ose interpretojnë keq rezultatet.



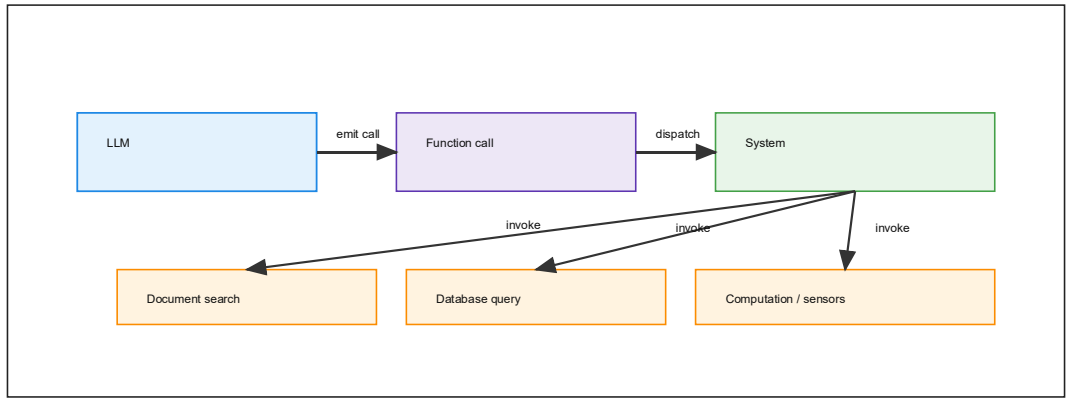
Prandaj sistemet serioze vendosin:

- listë mjetesh të lejuara,
- limite për numrin e veprimeve,
- mekanizma timeouts dhe retry,
- dhe logging të plotë (input/output për çdo tool call).

Përdorimi i mjeteve dhe thirrja e funksioneve

Përdorimi i mjeteve është aftësia e agjentit për të bërë veprime jashtë gjenerimit të tekstit si kërkim dokumentesh, pyetje në databazë, llogaritje, nxjerrje të dhënash nga sensorë.

Thirrja e funksionit formalizon ndërfaqen ku MMGj-ja prodhon një thirrje të strukturuar, sistemi e ekzekuton, dhe rezultati rikthehet.



Memoria

Memoria në agjentë nuk është një koncept unik, por një **arkitekturë**:

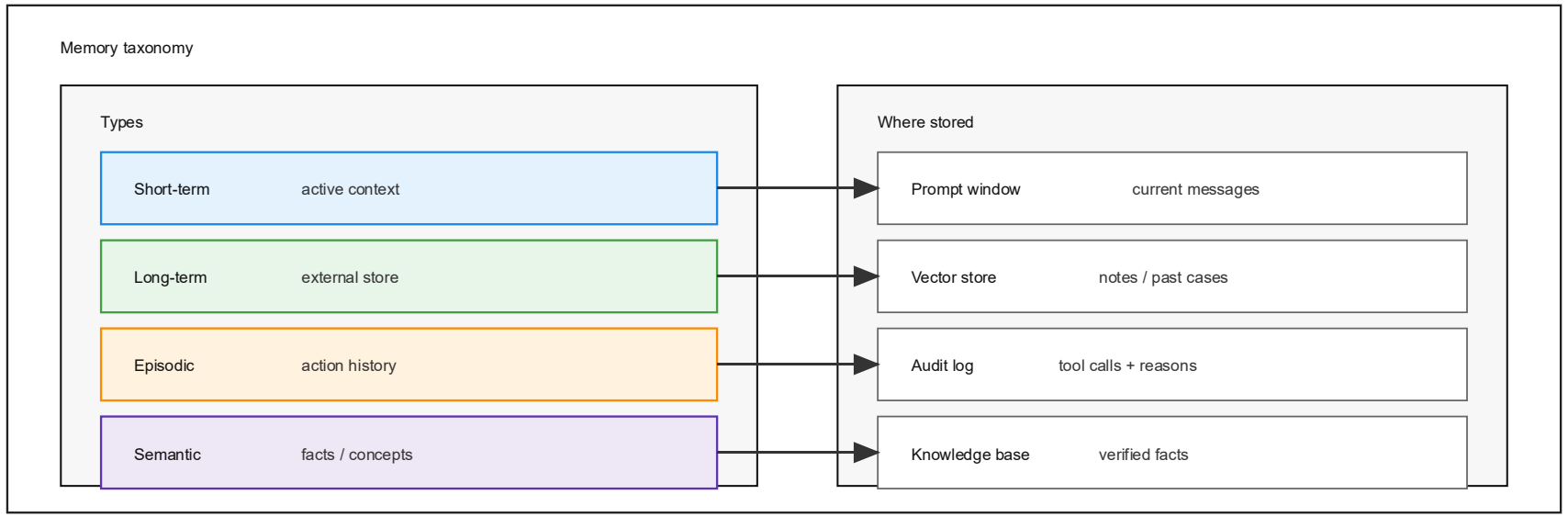
- **Short-term memory**: informacion aktiv në prompt
- **Long-term memory**: ruajtje jashtë (DB, vector store)
- **Episodic memory**: histori veprimesh
- **Semantic memory**: fakte të përgjithshme të nxjerra

Në aplikime të mbrojtjes civile, episodic memory është e domosdoshme për të dokumentuar çfarë vendimesh u morën dhe bazuar në çfarë evidencash.

Memoria në agjentë: llojet dhe funksionet

Memoria është kritike për agjentë pasi konteksti i MMGj është i kufizuar, detyrat mund të zgjasin, dhe duhet konsistencë.

Në mbrojtje civile, episodic log është pjesë e auditability: “çfarë u bë dhe pse”.



Orkestrimi

Orkestrimi është komponenti që:

- vendos kur agjenti vazhdon ose ndalet,
- validon veprimet e propozuara nga MMGj,
- menaxhon retry, timeout, dhe fallback,
- dhe ndërhyn për verifikim njerëzor kur është e nevojshme.

Pa orkestrim, agjentët janë të paparashikueshëm, potencialisht të rrezikshëm dhe të vështirë për t'u certifikuar.

Planifikimi në agjentë

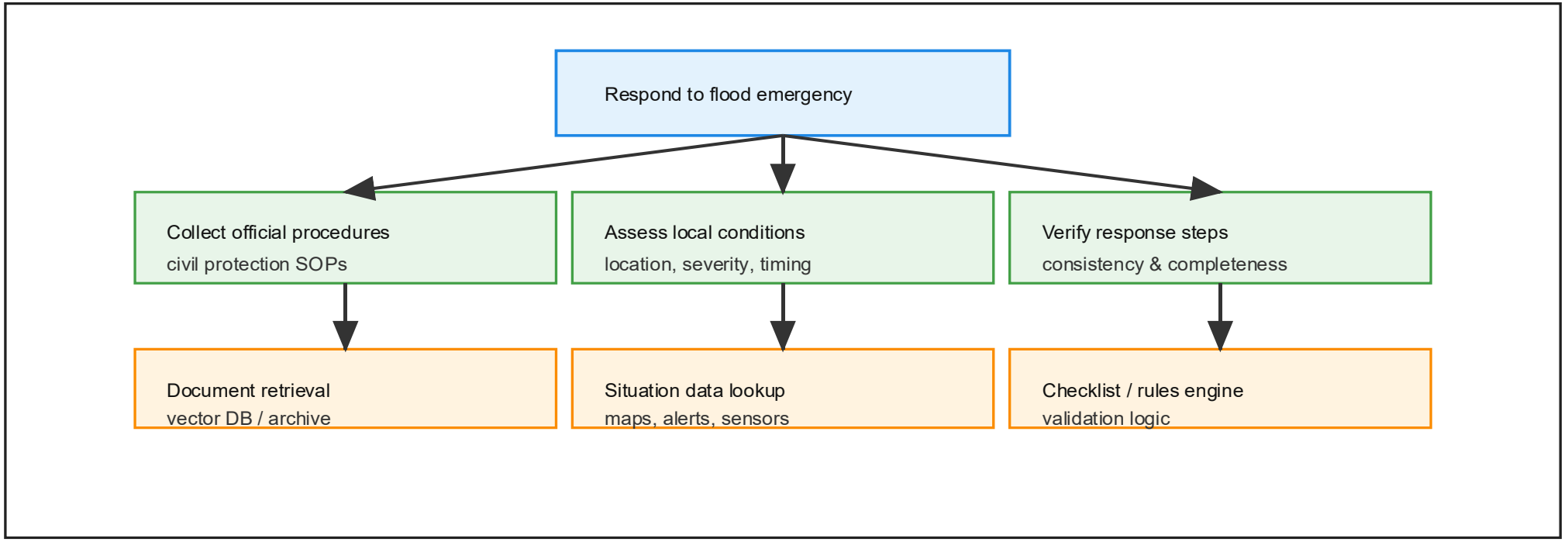
Planifikimi është mekanizëm për të minimizuar gabimet e improvizimit dhe për të strukturuar zgjidhjen multi-step.

Dy qasje:

- **Planifikim i lirë**: MMGj mendon hap pas hapi pa strukturë.
- **Planifikim i strukturuar:** kërkohet format plan (lista hapash, kritere ndalimi).

Në sisteme kritike rekomandohet planifikim i strukturuar, sepse lejon inspektim të planit, ndërhyrje njerëzore (human oversight) dhe vlerësim të detyrueshëm.

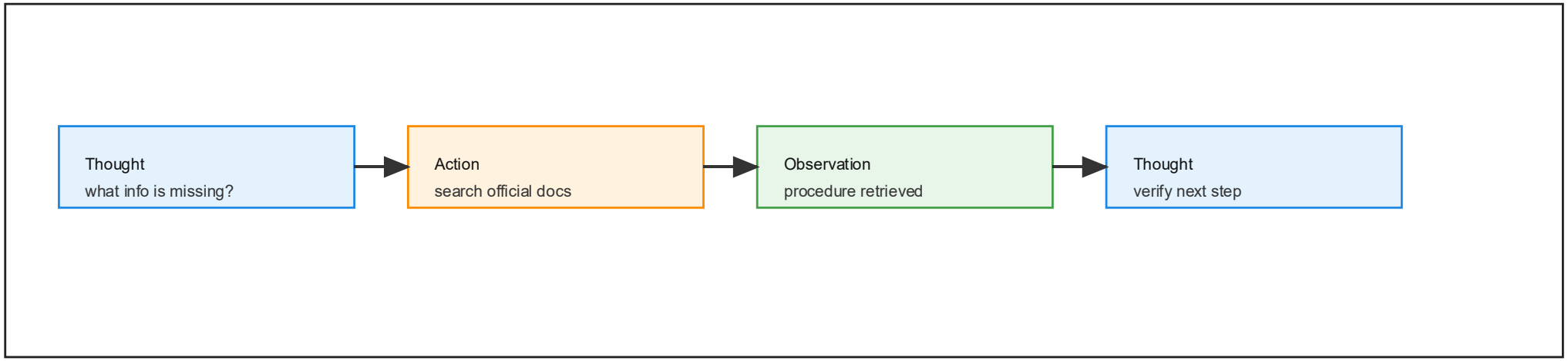
Planifikimi në agjentë



ReAct: ndërthurja e arsyetimit me veprime

ReAct është një paradigmë ku modeli alternon arsyetim të ndërmjetëm me veprime konkrete (tool calls), duke përdorur rezultatet si evidencë.

Kjo e bën agjentin më “evidence-driven” dhe ul nevojën që modeli të “mbushë boshllëqe” me gjenerim të pambështetur.



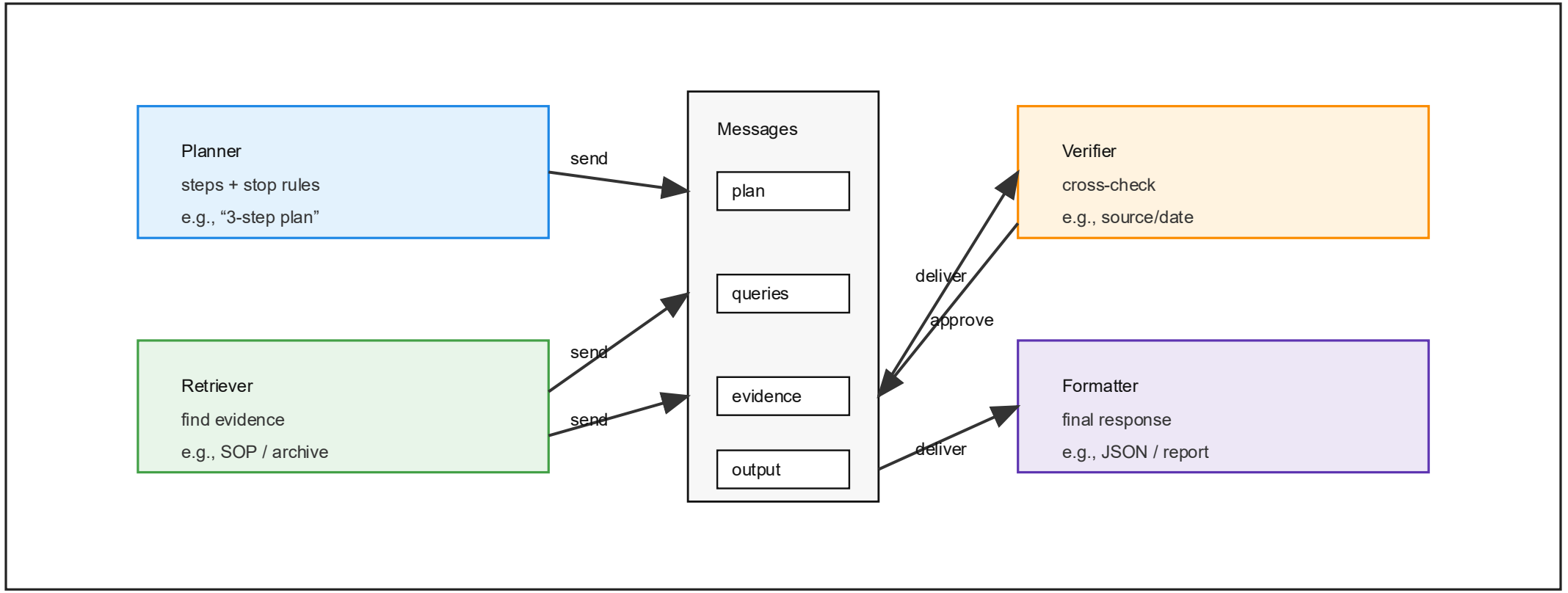
Sistemet me shumë agjentë

Sistemet multi-agjent do të thotë disa agjentë (ose role) që punojnë bashkë:

- një agjent planifikon,
- një agjent kërkon evidencë,
- një agjent verifikon përputhjen me burimet,
- një agjent formaton output-in.

Kjo rrit robustësinë, por sjell overhead komunikimi, mundësi konflikti,
dhe kompleksitet në orkestrim.

Sistemet me shumë agjentë



AI Mbrojtja Civile: Agjent për triage incidentesh

Agjent që merr raport incidenti dhe kryen:

- identifikon tipin (përmbytje/zjarr/tërmet),
- vlerëson prioritet (bazuar në rregulla të parapërcaktuara),
- kërkon protokollin përkatës,
- propozon veprime të standardizuara,
- prodhon një raport të audituar (me referenca).

Ky është use-case tipik ku agjentët tejkalojnë një “chatbot” sepse kërkojnë vendimmarrje të strukturuar,përdorim të mjeteve, dhe auditim.

Failure modes specifike të agjentëve

Agjentët sjellin failure modes të reja:

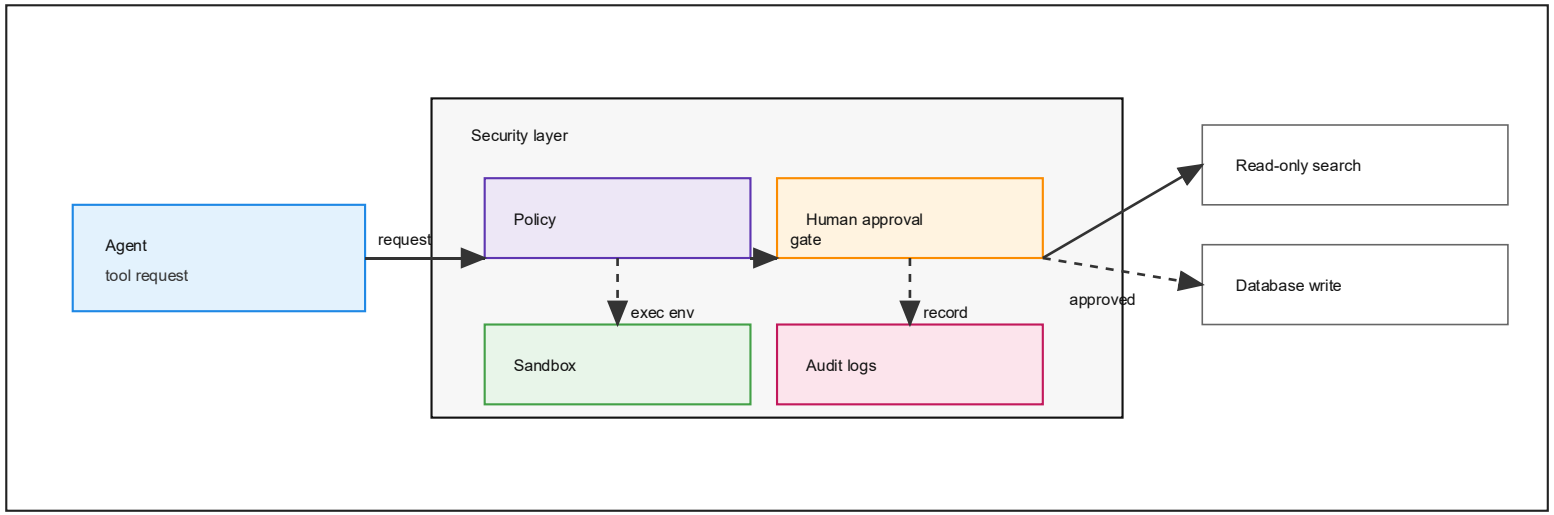
- agjenti nuk ndalet, vazhdon iterimin pa progres,
- thërret mjete pa nevojë (kosto/latencë),
- interpreton gabim output-et e mjeteve,
- devijon nga detyra origjinale,
- ekspozohet ndaj prompt injection nga burime.

Kjo kërkon stop conditions, kufizime, verifikim dhe monitorim.

Siguria në agjentë

Agjentët janë më të rrezikshëm se chatbot-et sepse mund të shkaktojnë veprime reale (p.sh. dërgojnë njoftime, ndryshojnë databaza).

Prandaj kërkohet leje minimale për mjete (least privilege), sandbox për veprime, approval njerëzor për operacione të ndjeshme, dhe logs të detajuar.



Vlerësimi i agjentëve

Ndryshe nga modelet e thjeshta, agjentët vlerësohen si sisteme:

- A e kryejnë detyrën?
- Sa hapa dhe sa kosto?
- A zgjedhin mjetin e duhur?
- A i qëndrojnë detyrës?
- A respektojnë politikat?

Kjo kërkon test-suite specifike dhe simulime (environments).

Benchmarks dhe mjedise testimi për agjentë

Për vlerësim serioz:

- ndërtohen mjedise testimi ku tool calls janë të simuluara,
- përdoren “replay logs” për të riprodhuar skenarë realë,
- dhe krahasohen politika orkestrimi.

Në domene si mbrojtja civile, simulimi është i domosdoshëm për të shmangur rrezikun operacional.

Kur është e arsyeshme të përdorësh agjent?

Një kriter praktik dizajni:

- Nëse detyra është një përgjigje e vetme e bazuar në dokumente → RAG është mjaftueshëm.
- Nëse detyra kërkon vendimmarrje dhe veprim të shumëfishtë → agjent.

Agjentët janë më kompleksë dhe duhet justifikuar përdorimi i tyre.

Çfarë do të trajtojmë në vazhdimësi?

- ✓ Hyrje
- ✓ Tokenizimi & Embeddings
- ✓ Arkitektura Transformer
- ✓ Bazat e të dhënave vektoriale (Vector Databases)
- ✓ RAG (Gjenerim i përforcuar nga kërkimi)
- ✓ Promptimi
- ✓ Agjentët (Agents)
8. Përshtatja e modelit (Fine-Tuning)
9. Vlerësimi (Evaluation)
10. Siguria & Përafrimi (Safety & Alignment)
11. Vendosja e modelit (Deployment)
12. Integrimi i Projektit Final

Bibliografia

1. S. Yao et al., “ReAct: Synergizing Reasoning and Acting in Language Models,” Proc. ICLR, 2023.
2. D. A. Shinn, B. Labash, and A. Gopinath, “Reflexion: Language Agents with Verbal Reinforcement Learning,” arXiv, 2023.
3. E. H. Park et al., “Generative Agents: Interactive Simulacra of Human Behavior,” Proc. CHI, 2023.
4. S. Mialon et al., “Augmented Language Models: a Survey,” arXiv, 2023.
5. P. W. Koh et al., “Toolformer: Language Models Can Teach Themselves to Use Tools,” arXiv, 2023.
6. C. Nakano et al., “WebGPT: Browser-assisted question-answering with human feedback,” arXiv, 2021.
7. M. Chase and H. G. Contributors, “LLM Agents: A Survey of Methods and Benchmarks,” arXiv, 2024.
8. S. Gupta et al., “CALM: Confident Adaptive Language Modeling,” arXiv, 2023.