Joni Juntto

4.4.2026

Kontekstinhallinta ja siihen liittyvät riskit moniagenttisissa tekoälyjärjestelmissä

Johdanto

Generatiivisen tekoälyn ja suurten kielimallien käyttö ohjelmistoarkkitehtuureissa on siirtynyt viime vuosien aikana yksittäisistä kysymys/vastaus-järjestelmistä kohti agenttisia järjestelmiä, joissa tekoälymallit suunnittelevat, käyttävät työkaluja ja toimivat autonomisesti laajemmissakin tehtävissä. Tätä muutosta on kuvattu siirtymänä kohti tavoiteohjautuvaa ohjelmistoarkkitehtuuria, joissa kielimalli ei enää ole vain tekstintuottaja vaan osa laajempaa toiminnallista kokonaisuutta (Alenezi 2026; Wang ym. 2024).

Kehitys on lisännyt kiinnostusta järjestelmiin, joissa tehtävä hajautetaan useille erikoistuneille tekoälyagenteille. Näiden järjestelmien potentiaalia on perusteltu sillä, että työnjako voi tukea ongelmanratkaisua, tiedonhakua ja työkaluohjattua toimintaa (Xi ym. 2025; Wang ym. 2024). Realimaailman havainnot ja tuore tutkimus viittaavat kuitenkin siihen, että vapaasti kommunikoivissa moniagenttijärjestelmissä järjestelmän epäonnistumiset ovat edelleen tavallisia ja koordinaatiokustannukset voivat kasvaa nopeasti agenttien määrän lisääntyessä (Cemri ym. 2025).

Tässä tekstissä tarkastelen sitä, miten konteksti toimii moniagenttisen järjestelmän haavoittuvuuspintana ja mahdollisena heikkoutena. Pääväitteeni on, että moniagenttisten tekoälyjärjestelmien keskeiset luotettavuusongelmat syntyvät siitä, että yksittäisten mallien virheet, epävarmat tulkinnat ja puutteelliset tai synteettiset tulokset pääsevät leviämään, kertautumaan ja vakiintumaan järjestelmätason ongelmiksi.

1. Käsitteet

1.1 Agenttinen järjestelmä

Tässä tekstissä agenttisella järjestelmällä tarkoitetaan ohjelmistokokonaisuutta, joka käyttää suurta kielimallia tai suuria kielimalleja päätöksenteon tai päättelyn ytimenä ja kykenee käyttämään työkaluja, rajapintoja tai muita ohjelmallisia toimintoja (Wang ym. 2024; Xi ym. 2025). Käytännössä tällainen järjestelmä täyttää seuraavat ehdot:

- se käyttää suurta kielimallia päätöksenteon tai ohjauksen ytimenä
- se voi kutsua työkaluja, ohjelmointirajapintoja tai muita toimintoja
- se hyödyntää muistia, tiedonhakua tai ulkoisia tietolähteitä
- se voi toimia osana usean agentin yhteistyörakennetta

Määritelmä on tarkoituksella laaja, koska tutkimuskirjallisuudessa agenttisuus kuvataan usein jatkumona eikä tarkkarajaisena kategoriana. Tässä tekstissä huomio kohdistuu erityisesti niihin agenttisiin järjestelmiin, joissa useat agentit lukevat, tuottavat ja välittävät toisilleen tietoa saman tehtäväkokonaisuuden aikana.

1.2 Konteksti agenttisessa järjestelmässä

Agenttisessa järjestelmässä konteksti ei ole vain käyttäjän viimeisin syöte. Konteksti koostuu dynaamisesta tietokokonaisuudesta, joka ylläpitää agentin tilannetietoisuutta ja ohjaa sen seuraavia toimintoja. Kontekstiin kuuluu tyypillisesti järjestelmätason ohjeistus, vuorovaikutushistoria, työkalukutsujen kuvaukset, työkalujen palauttamat tulokset, mahdollisesti haettu muistisisältö sekä tehtävän aikana syntyneet tulokset (Wang ym. 2024; Mialon ym. 2023; Xi ym. 2025).

Koska suuret kielimallit ovat luonnostaan tilattomia, tämä konteksti joudutaan välittämään mallille yhä uudelleen jokaisessa uudessa päättely- tai toimintavaiheessa. Tämän vuoksi konteksti ei ole agenttijärjestelmässä vain tekninen taustarakenne, vaan tietoperusta, jonka varassa agentti tulkitsee tilannettaan ja muodostaa toimintasuunnitelmansa (Wang ym. 2024).

1.3 Kontekstin saastuminen ja lähikäsitteet

Tässä tekstissä pääterminä käytetään käsitettä kontekstin saastuminen (context pollution). Sillä tarkoitetaan tilannetta, jossa agentin käyttämään kontekstiin päätyy epäolennaista, harhaanjohtavaa, ristiriitaista tai manipuloitua tietoa, joka heikentää myöhempää päättelyä ja voi ohjata toimintaa väärään suuntaan. Huang ym. (2026) tarkastelevat ilmiötä laajemmin mallin oman tuottaman tekstin vaikutuksena myöhempään päättelyyn; tämän tekstin tulkinnassa heidän havaintonsa tukevat ajatusta siitä, että kontekstin laadulla on ratkaiseva merkitys agentin myöhemmälle suorituskyvylle.

Kontekstin saastuminen on hyödyllistä erottaa ainakin kolmesta läheisestä ilmiöstä. Ensinnäkin kontekstin mätäneminen tarkoittaa tässä tekstissä pitkien vuorovaikutusten aikana syntyvää laadun heikkenemistä, jossa olennaiset ohjeet hukkuvat kasvavan kontekstikuorman alle. Toiseksi mallin myrkyttäminen viittaa koulutus- tai hienosäätöaineiston tahalliseen tai tahattomaan kontaminoitumiseen. Kolmanneksi mallin romahtaminen tarkoittaa tilannetta, jossa iteratiivisesti generoitu aineisto rapauttaa myöhempien mallien koulutuspohjaa (Shumailov ym. 2024). Näistä vain ensimmäinen kuuluu suoraan tämän tekstin pääkohteeseen.

Kontekstin saastuminen ei synny aina vahingossa. NISTin taksonomia generatiiviseen tekoälyyn kohdistuvista hyökkäyksistä osoittaa, että kontekstiputkeen voidaan kohdistaa hyökkäyksiä myös tarkoituksellisesti (Vassilev ym. 2025). Zverev ym. (2024) puolestaan tarkastelevat epäsuoria kehoteinjektioita, joissa malli voi tulkita ulkoisesta aineistosta peräisin olevan tekstin toimintaohjeeksi. Näiden lähteiden perusteella kontekstin saastuminen on syytä ymmärtää sekä käytettävyys- että turvallisuuskysymyksenä.

2. Riskimekanismit

2.1 Tilattomuus ja kontekstikuorman kasvu

Moniagenttijärjestelmän ensimmäinen keskeinen riskimekanismi syntyy siitä, että tilattomat kielimallit tarvitsevat yhä uudelleen kertyvän kontekstin voidakseen jatkaa tehtävää, mikä kasvattaa virheiden siirtymisen todennäköisyyttä. Jokainen uusi kierros voi sisältää alkuperäisen järjestelmäohjeen, aiemmat vastaukset, työkalukutsut, työkalujen raakapalautteet ja väliaikaiset virheilmoitukset (Mialon ym. 2023; Wang ym. 2024). Mitä pidemmäksi suoritusketju kasvaa, sitä vaikeammaksi muuttuu olennaisen tiedon erottaminen epäolennaisesta.

Liu ym. (2024) osoittavat, että pitkissä konteksteissa kielimallit eivät käytä kaikkea saatavilla olevaa tietoa tasaisesti, vaan olennaista sisältöä voi jäädä käytännössä huomiotta. Tästä ei seuraa automaattisesti täydellinen epäonnistuminen, mutta havainto tukee sitä tulkintaa, että pelkkä suuri konteksti-ikkuna ei ratkaise agenttijärjestelmien muistiongelmaa. Voidaan siksi esittää, että moniagenttisissa työnkuluissa kontekstin kasvu toimii paitsi kapasiteettiongelmana myös laadunhallinnan ongelmana.

2.2 Hallusinaatioiden vahvistuminen

Toinen riskimekanismi liittyy siihen, että yhden agentin virheellinen oletus voi siirtyä seuraavan agentin lähtötiedoksi ja vahvistua työnkulun edetessä. Jos agentti tulkitsee ohjeen väärin, hallusinoi puuttuvan tiedon tai tekee virheellisen työkalukutsun, tulos ei jää välttämättä paikalliseksi virheeksi, vaan voi muuttua muiden agenttien käyttämäksi kontekstiksi. Tällöin virhe siirtyy pois yksittäisen mallivastauksen tasolta kohti järjestelmätason ongelmaa (Zhou ym. 2025).

Riski kasvaa erityisesti silloin, kun agenteilla on valtuuksia kutsua ulkoisia rajapintoja, päivittää tietokantoja tai käynnistää uusia osaprosesseja. Chen ym. (2025) tarkastelevat lääketieteellisten moniagenttijärjestelmien turvallisuusriskejä ja osoittavat, että virheiden vaikutukset voivat tällaisissa ympäristöissä levitä päätöksenteosta toimintaan. Näiden havaintojen perusteella hallusinaatio on moniagenttisessa järjestelmässä harvoin vain tekstuaalinen virhe; se voi muuttua operatiiviseksi virheeksi, jos järjestelmäarkkitehtuuri ei pysäytä sitä ajoissa.

2.3 Ryhmämuistin muodostuminen

Kolmas riskimekanismi on se, että toistuvat viittaukset samoihin virheellisiin oletuksiin voivat vakauttaa ne koko järjestelmän jaetuksi työoletukseksi. Kun useat agentit lukevat toistensa tuotoksia ja käyttävät niitä uusien päätelmien pohjana, alustavasta tulkinnasta voi muodostua käytännössä ryhmämuisti, vaikka alkuperäinen tieto olisi ollut epävarmaa tai virheellistä. Kulshreshtha ym. (2026) kuvaavat moniagenttijärjestelmissä emergenttejä ja yhteistyötä heikentäviä käyttäytymismalleja; Zhou ym. (2025) puolestaan osoittavat, että suhteet agenttien välillä vaikuttavat siihen, miten virheet leviävät yhteistyöverkossa.

Moniagenttisessa ympäristössä ongelma ei ole vain yksittäinen virheellinen sisältö, vaan virheen sosiaalistuminen osaksi järjestelmän yhteistä muistia. Kun näin tapahtuu, myöhempien agenttien on entistä vaikeampi erottaa alkuperäinen havainto sitä koskevasta tulkinnasta.

2.4 Informaatiokontaminaatio ja tulkintakehysten siirtyminen

Neljäs riskimekanismi liittyy siihen, ettei kontekstin saastuminen koske vain faktoja, lukuja tai teknisiä välituloksia, vaan myös tulkintakehyksiä, joiden varassa myöhemmät agentit jäsentävät tehtävää. Tämä korostuu tehtävissä, joissa agentit arvioivat poliittisia, yhteiskunnallisia, terveydellisiä tai eettisiä sisältöjä. Zhang ym. (2025) raportoivat eroja kielimallien ideologisissa profiileissa, ja Civelli ym. (2025) osoittavat, että ideologisesti roolitetut agentit voivat vaikuttaa sisältömoderoinnin tuloksiin. Näiden lähteiden perusteella on perusteltua todeta, että agentit ja niille annetut roolit voivat tuottaa toisistaan poikkeavia arvottavia tulkintoja.

Moniagenttisen kontekstinhallinnan näkökulmasta keskeinen riski syntyy silloin, kun tällainen tulkintakehys tallentuu jaettuun kontekstiin, muistisisältöön tai välitulokseen ja siirtyy seuraavan agentin lähtöoletukseksi. Tällöin myöhempi agentti ei vastaanota vain yksittäistä väitettä, vaan myös tavan jäsentää ongelmaa, priorisoida tietoa ja rajata hyväksyttäviä ratkaisuja. Voidaan siksi olettaa, että informaatiokontaminaatio näkyy moniagenttisissa työnkuluissa ennen kaikkea tulkintakehysten siirtymisenä, jos vinoutunut kehys jää ilman riippumatonta tarkastusta, se voi vahvistua työnkulun edetessä, vaikka nykyiset lähteet eivät vielä yksin osoita kaikkia tämän leviämismekanismin muotoja.

3. Riskienhallinta-arkkitehtuurit

3.1 Orkestrointi ja roolien eristäminen

Jos moniagenttijärjestelmien keskeiset ongelmat syntyvät agenttien välisten rajapintojen, muistirakenteiden ja viestinvälityksen tasolla, riskienhallinnan on kohdistuttava ennen kaikkea arkkitehtuuriin. Cemri ym. (2025) raportoivat, että vapaasti kommunikoivat moniagenttiasetelmat ovat alttiita koordinaatio-ongelmille. Tästä seuraa käytännöllinen suunnitteluperiaate: agenttien välinen viestintä kannattaa pitää rajattuna, tehtävät roolittaa kapeasti ja päätösvastuut erottaa toisistaan.

Roolien eristäminen tarkoittaa tässä yhteydessä sitä, että jokaisella agentilla on rajattu tehtävä, rajattu työkalupinta ja rajatut oikeudet käyttää tai muuttaa yhteistä tietoa. Wang ym. (2024) ja Xi ym. (2025) kuvaavat agenttijärjestelmiä arkkitehtuurisina kokonaisuuksina, joissa suunnittelu, työkalujen käyttö ja muistinhallinta voidaan erottaa omiksi toiminnoikseen. Juuri tällainen työnjako vähentää riskiä, että yksi saastunut kontekstiketju pääsee dominoimaan koko järjestelmää.

3.2 Riippumaton verifiointi

Pelkkä työnjako ei riitä, jos kaikki agentit jakavat saman virheellisen oletuksen. Siksi moniagenttijärjestelmä tarvitsee riippumattomia verifiointikerroksia, jotka eivät perustu vain edeltävän agentin luonnollisella kielellä muotoiltuun selitykseen. Zhou ym. (2025) esittävät keinoja mallintaa agenttien välisiä suhteita ja tunnistaa riskialttiita vuorovaikutuskuvioita, kun taas Chen ym. (2025) korostavat turvallisuustarkastusten merkitystä korkean riskin ympäristöissä.

Käytännössä riippumaton verifiointi voi tarkoittaa erillistä tarkistusagenttia, sääntöpohjaisia validointeja, rajapintojen palautteiden vertaamista odotettuihin skeemoihin tai ulkoisista lähteistä tehtävää ristivarmistusta. Keskeinen periaate on, että järjestelmä ei luota yhteen päättelyketjuun vain siksi, että se on esitetty johdonmukaisella luonnollisella kielellä.

3.3 Muistinhallinta ja kontekstin kuratointi

Koska pitkät kontekstit eivät itsessään takaa parempaa suorituskykyä, muistinhallinnan on sisällettävä myös aktiivista karsintaa. Liu ym. (2024) osoittavat, että pitkän kontekstin hyödyntäminen on epätasaista, ja Mialon ym. (2023) kuvaavat arkkitehtuureja, joissa ulkoiset tietolähteet ja muistimekanismit tukevat mallin toimintaa. Näiden lähteiden perusteella muistia ei tulisi käsitellä rajattomana säiliönä, vaan kuratoitavana resurssina.

Tämä tarkoittaa käytännössä sitä, että järjestelmässä on erotettava pysyvästi relevantti tieto väliaikaisista työkalulokeista, virheilmoituksista ja ratkaisuhaarojen keskeneräisistä kokeiluista. Jos kaikki tuotettu aineisto syötetään takaisin seuraaville agenteille, kontekstin määrä kasvaa nopeammin kuin sen hyödyllisyys. Muistinhallinta on rakennettava valikoivaksi. Olennaista tietoa säilytetään, epäolennaista poistetaan ja jokaisen muistisisällön alkuperä tehdään näkyväksi.

3.4 Rakenteinen tiedonsiirto agenttien välillä

Luonnollinen kieli on joustava viestintämuoto, mutta moniagenttisessa järjestelmässä juuri joustavuus voi lisätä monitulkintaisuutta. Yao ym. (2022) ja Mialon ym. (2023) osoittavat yleisemmin, että mallin päättely, työkalujen käyttö ja ulkoiset tietorakenteet voidaan kytkeä yhteen useilla tavoilla. Moniagenttisissa järjestelmissä kannattaa suosia tilanteen mukaan myös rakenteista tiedonsiirtoa, jossa agenttien välillä välitetään selkeästi rajattuja kenttiä, tilatietoja ja tuloksia vapaan sanallisen kuvauksen sijaan.

Rakenteinen tiedonsiirto ei poista virheitä, mutta se voi tehdä niistä helpommin havaittavia ja rajattavia. Jos agentti välittää seuraavalle agentille esimerkiksi eksplisiittisen tehtävätilan, lähteen, luottamustason ja vaaditun seuraavan toimenpiteen, myöhemmän agentin on helpompi erottaa havainto tulkinnasta. Tämän vuoksi luonnollisen kielen varaan rakentuva viestintä kannattaa täydentää skeemoilla, rajatuilla kentillä ja tarkistettavilla välituloksilla aina, kun järjestelmän luotettavuus on kriittistä.

Yhteenveto

Moniagenttisten tekoälyjärjestelmien ongelmat näyttävät tämän tekstin perusteella liittyvän ennen kaikkea siihen, miten konteksti muodostuu, siirtyy ja muuttuu agentilta toiselle. Keskeinen kysymys ei ole vain se, tekeekö yksittäinen agentti virheen, vaan se, päästääkö järjestelmäarkkitehtuuri virheen, epävarman tulkinnan tai puutteellisen välituloksen leviämään seuraavien agenttien lähtökohdaksi. Käsitetasolla keskeistä on erottaa kontekstin saastuminen muista läheisistä ilmiöistä, kuten mallin myrkyttämisestä ja mallin romahtamisesta. Riskimekanismien tasolla tilattomuus, kontekstikuorman kasvu, hallusinaatioiden vahvistuminen, ryhmämuistin muodostuminen ja tulkintakehysten siirtyminen tekevät virheistä helposti kertaantuvia.

Näiden havaintojen perusteella voidaan päätellä, että moniagenttijärjestelmien luotettavuus riippuu ennen kaikkea siitä, kuinka hyvin niiden arkkitehtuuri hallitsee viestinvälitystä, muistia ja verifiointia. Orkestrointi, roolien eristäminen, riippumaton verifiointi, valikoiva muistinhallinta ja rakenteinen tiedonsiirto eivät tällöin ole pelkkiä optimointeja, vaan mekanismeja, joilla paikalliset virheet ja vinoutuneet tulkintakehykset estetään muuttumasta järjestelmätason ongelmiksi pitkissä ja turvallisuuskriittisissä työnkuluissa.

Lähteet

Alenezi, M. 2026. From Prompt-Response to Goal-Directed Systems: The Evolution of Agentic AI Software Architecture. arXiv preprint arXiv:2602.10479. DOI: https://doi.org/10.48550/arXiv.2602.10479

Cemri, M., Pan, M. Z., Yang, S., Agrawal, L. A., Chopra, B., Tiwari, R., Keutzer, K., Parameswaran, A., Klein, D., Ramchandran, K. ja Zaharia, M. 2025. Why Do Multi-Agent LLM Systems Fail? arXiv preprint arXiv:2503.13657. DOI: https://doi.org/10.48550/arXiv.2503.13657

Chen, K., Zhen, T., Wang, H., Liu, K., Li, X., Huo, J., Yang, T., Xu, J., Dong, W. ja Gao, Y. 2025. MedSentry: Understanding and Mitigating Safety Risks in Medical LLM Multi-Agent Systems. arXiv preprint arXiv:2505.20824. DOI: https://doi.org/10.48550/arXiv.2505.20824

Civelli, S., Bernardelle, P., Pratama, N. A. ja Demartini, G. 2025. Ideology-Based LLMs for Content Moderation. arXiv preprint arXiv:2510.25805. DOI: https://doi.org/10.48550/arXiv.2510.25805

Huang, J., Choshen, L., Astudillo, R., Broderick, T. ja Andreas, J. 2026. Do LLMs Benefit From Their Own Words? MIT-IBM Watson AI Lab. DOI: https://doi.org/10.48550/arXiv.2602.24287

Kulshreshtha, D., Du, W., Jain, R., Doss, S., Su, H., Swamy, S. ja Qi, Y. 2026. The Subtle Art of Defection: Understanding Uncooperative Behaviors in LLM-Based Multi-Agent Systems. In Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics, Volume 5: Industry Track, 571-585. DOI: https://doi.org/10.48550/arXiv.2511.15862

Liu, N. F., Lin, K., Hewitt, J., Paranjape, A., Bevilacqua, M., Petroni, F. ja Liang, P. 2024. Lost in the Middle: How Language Models Use Long Contexts. Transactions of the Association for Computational Linguistics, 12, 157-173. DOI: https://doi.org/10.48550/arXiv.2307.03172

Mialon, G., Dessì, R., Lomeli, M., Nalmpantis, C., Pasunuru, R., Raileanu, R., Rozière, B., Schick, T., Dwivedi-Yu, J., Celikyilmaz, A. ja Grave, E. 2023. Augmented Language Models: A Survey. arXiv preprint arXiv:2302.07842. DOI: https://doi.org/10.48550/arXiv.2302.07842

Shumailov, I., Shumaylov, Z., Zhao, Y., Papernot, N., Anderson, R. ja Gal, Y. 2024. AI Models Collapse When Trained on Recursively Generated Data. Nature, 631(8022), 755-759. DOI: https://doi.org/10.1038/s41586-024-07566-y

Vassilev, A., Oprea, A., Fordyce, A., Anderson, H., Davies, X. ja Hamin, M. 2025. Adversarial Machine Learning: A Taxonomy and Terminology of Attacks and Mitigations. NIST Trustworthy and Responsible AI, NIST AI 100-2e2025. DOI: https://doi.org/10.6028/NIST.AI.100-2e2025

Wang, L., Ma, C., Feng, X., Zhang, Z., Yang, H., Zhang, J., Chen, Z., Tang, J., Chen, X., Lin, Y. ja Zhao, W. X. 2024. A Survey on Large Language Model Based Autonomous Agents. Frontiers of Computer Science, 18(6), 186345. DOI: https://doi.org/10.48550/arXiv.2308.11432

Xi, Z., Chen, W., Guo, X., He, W., Ding, Y., Hong, B., Zhang, M., Wang, J., Jin, S., Zhou, E. ja Zheng, R. 2025. The Rise and Potential of Large Language Model Based Agents: A Survey. Science China Information Sciences, 68(2), 121101. DOI: https://doi.org/10.48550/arXiv.2309.07864

Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K. R. ja Cao, Y. 2022. ReAct: Synergizing Reasoning and Acting in Language Models. In The Eleventh International Conference on Learning Representations. DOI: https://doi.org/10.48550/arXiv.2210.03629

Zhang, X., Mao, R. ja Cambria, E. 2025. A Systematic Analysis of Biases in Large Language Models. arXiv preprint arXiv:2512.15792. DOI: https://doi.org/10.48550/arXiv.2512.15792

Zhou, J., Wang, L. ja Yang, X. 2025. Guardian: Safeguarding LLM Multi-Agent Collaborations with Temporal Graph Modeling. arXiv preprint arXiv:2505.19234. DOI: https://doi.org/10.48550/arXiv.2505.19234

Zverev, E., Abdelnabi, S., Tabesh, S., Fritz, M. ja Lampert, C. H. 2024. Can LLMs Separate Instructions from Data? And What Do We Even Mean by That? arXiv preprint arXiv:2403.06833. DOI: https://doi.org/10.48550/arXiv.2403.06833