



Examensarbete inom datateknik

Grundnivå, 15 hp

Utvärdering av tillförlitligheten hos LLM-genererade insikter i automatiserad molnrapportering

En fallstudie om teknisk noggrannhet och användbarhet för
intressenter

**Majd Matloob
Ishar Zeito**

Utvärdering av tillförlitligheten hos LLM-genererade insikter i automatiserad molnrapportering

**En fallstudie om teknisk noggrannhet och användbarhet
för intressenter**

Evaluating the Reliability of LLM-Generated Insights in Automated Cloud Reporting

**A Case Study on Technical Accuracy and Stakeholder
Usability**

Majd Matloob
Ishar Zeito

Examensarbete inom datateknik
Grundnivå, 15 hp
Handledare på KTH: Ulrika Eriksson
Examinator: Anders Lindström
TRITA-CBH-GRU-2026:134

KTH
Skolan för elektroteknik och datavetenskap
141 52 Huddinge, Sweden

Sammanfattning

Molnbaserade driftmiljöer genererar stora mängder teknisk data som behöver kommuniceras till icke-tekniska beslutsfattare på ett begripligt och tillförlitligt sätt. Stora språkmodeller erbjuder möjligheten att automatisera denna rapportering, men det är oklart hur val av tekniskt upplägg påverkar kvaliteten och tillförlitligheten i de genererade sammanfattningarna.

Detta examensarbete undersöker hur fem olika LLM-upplägg påverkar tillförlitligheten i AI-genererade molnrapporter, det vill säga automatiskt genererade sammanfattningar av drift-, incident- och resursdata från en molnbaserad miljö. För att möjliggöra jämförelsen utvecklades en begränsad prototyp för automatiserad molnrapportering i Azure, vilken användes som testbädd i utvärderingen. Två baslinjeupplägg utan källdata användes som jämförelsepunkter, medan huvudanalysen fokuserade på tre källförankrade upplägg: Standard RAG, Spårbar RAG och ett validerat upplägg med självgranskning. Uppläggen utvärderades med avseende på teknisk korrekthet, trohet mot underlaget, spårbarhet, konsistens och begriplighet. Arbetet operationaliserar tillförlitlighet som fem mätbara kvalitetsdimensioner och adresserar därmed en identifierad kunskapslucka gällande praktiska utvärderingsmetoder för faktabaserad AI-rapportering riktad till icke-tekniska mottagare.

Resultaten visar att explicita källhänvisningar hade störst betydelse för upplevd tillförlitlighet hos icke-tekniska mottagare, trots att de inte förbättrade den objektiva innehållskvaliteten ytterligare jämfört med källförankring utan hänvisningar. Källhänvisningarna fungerade som en transparensignal snarare än ett tekniskt verifieringsverktyg. Självgranskning som kontrollsteg bidrog marginellt till konsistens men förbättrade inte kvaliteten i absoluta termer. Även de bästa uppläggen återgav inte all säkerhetskritisk information i underlaget. AI-genererade rapporter bör därför betraktas som beslutsstöd snarare än som fullständiga beskrivningar av hela driftläget.

Nyckelord

Stora språkmodeller, tillförlitlighet, retrieval-augmented generation, automatiserad rapportering, AIOps, molnrapportering, hallucinationer, källförankring, multi-tenant, beslutsstöd

Abstract

Cloud-based operational environments generate large volumes of technical data that need to be communicated to non-technical decision-makers in a comprehensible and reliable manner. Large language models offer the possibility of automating this reporting, but it remains unclear how the choice of technical configuration affects the quality and reliability of the generated summaries.

This thesis investigates how five different LLM configurations affect the reliability of AI-generated cloud reports, meaning automatically generated summaries of operational, incident, and resource data from a cloud-based environment. To enable the comparison, a limited prototype for automated cloud reporting was developed in Azure and used as a testbed in the evaluation. Two baseline configurations without source data were used as reference points, while the main analysis focused on three retrieval-augmented configurations: Standard RAG, Traceable RAG, and a validated configuration with self-validation. The configurations were evaluated with respect to factual accuracy, faithfulness to the source data, traceability, consistency, and comprehensibility. The thesis operationalises trustworthiness as five measurable quality dimensions, thereby addressing an identified gap in practical evaluation methods for fact-based AI reporting directed at non-technical recipients.

The results show that explicit source citations had the greatest impact on perceived trustworthiness among non-technical recipients, even though they did not further improve objective content quality compared to retrieval-augmented generation without citations. The citations functioned as a transparency signal rather than a technical verification tool. Self-validation as a control step contributed marginally to consistency but did not improve the measured quality criteria compared to Traceable RAG. Even the best-performing configurations did not reproduce all security-critical information from the source data. AI-generated reports should therefore be regarded as decision support rather than complete descriptions of the overall operational state.

Keywords

Large language models, trustworthiness, retrieval-augmented generation, automated reporting, AIOps, cloud reporting, hallucination, grounding, multi-tenant, decision support

Innehåll

1	Inledning	1
1.1	Problemformulering	1
1.2	Målsättning	1
1.3	Avgränsningar	2
1.4	Författarnas bidrag till examensarbetet	2
1.5	Etiskt förhållningssätt	2
2	Teori och bakgrund	3
2.1	Molnbaserade driftmiljöer och AIOps	3
2.1.1	Driftdata och rapportering	3
2.1.2	AIOps och språkmodeller	3
2.1.3	Microsoft-nära kontext	4
2.2	<i>Hallucinationer, factuality</i> och <i>faithfulness</i>	5
2.2.1	Definition och dimensioner	5
2.2.2	Betydelse för teknisk rapportering	5
2.3	Stora språkmodeller och tillförlitlighet	6
2.4	Tekniker för att stärka tillförlitlighet	6
2.4.1	Promptning och kontrollsteg	6
2.4.2	Källförankring och RAG	7
2.4.3	Utmaningar med källförankrad generering	7
2.4.4	Övriga tekniker	8
2.5	Tidigare arbeten om automatiserad rapportgenerering	8
2.6	Syntes och identifierad kunskapslucka	8
2.6.1	Sammanfattning av forskningsläget	8
2.6.2	Identifierad kunskapslucka	9
3	Metod	11
3.1	Forskningsansats	11
3.2	Översikt av tillvägagångssätt	11
3.2.1	Litteraturstudie	11
3.2.2	Val av lösningsmetod	12
3.3	Systemarkitektur och modellering	12
3.4	Datakällor och experimentuppsättning	13
3.5	Jämförda LLM-upplägg	14
3.6	Utvärderingsramverk	16
3.6.1	Kriterier för tillförlitlighet	16
3.6.2	Mätmetoder och bedömningsförfarande	16
3.6.3	Felkategorier och poängschema	17
3.7	Kvalitativ utvärdering av begriplighet och beslutsstöd	18

3.7.1	Deltagare och genomförande	18
3.7.2	Bedömningsskala och analys	18
3.8	Kompletterande analys av <i>multi-tenant</i> -arkitektur	19
3.8.1	Analysfokus	19
3.8.2	Analyserade designval	19
3.9	Stödmätningar av prestanda och kostnad	20
4	Resultat	21
4.1	Kvantitativ utvärdering av LLM-uppläggen	21
4.1.1	Teknisk korrekthet (<i>factuality errors</i>)	21
4.1.2	Trohet mot underlaget (<i>omissions</i> och <i>faithfulness errors</i>)	22
4.1.3	Spårbarhet och källförankring (<i>citation accuracy</i>)	22
4.1.4	Konsistens vid upprepade körningar	23
4.2	Kvalitativ utvärdering – begriplighet och beslutsstöd	23
4.2.1	Deltagaröversikt	24
4.2.2	Resultat per bedömningsparameter	24
4.2.3	Kvalitativa observationer	26
4.3	Kompletterande analys av <i>multi-tenant</i> -arkitektur	26
4.4	Stödmätningar – prestanda och kostnad	27
4.5	Sammanställning	28
5	Analys och diskussion	29
5.1	Källförankringens betydelse för objektiv innehållskvalitet	29
5.2	Källhänvisningarnas roll för upplevd tillförlitlighet	29
5.3	Självgranskning och konsistens	30
5.4	Baslinjernas roll i utvärderingen	30
5.5	Tillförlitlighet bortom genereringssteget	31
5.6	Metodologisk diskussion	31
5.7	Samhälleliga konsekvenser	32
5.7.1	Ekonomiska konsekvenser	32
5.7.2	Sociala konsekvenser	32
5.7.3	Etiska konsekvenser	33
5.7.4	Miljömässiga konsekvenser	33
6	Slutsatser	35
6.1	Svar på huvudfrågan	35
6.2	Svar på delfrågan om <i>multi-tenant</i> -arkitektur	35
6.3	Bidrag till kunskapsutvecklingen	36
6.4	Begränsningar	36
6.5	Fortsatt arbete	36

Källförteckning	37
Bilaga A – Promptmallar för LLM-uppläggen	39
Bilaga B – Exempel på facit för testscenario S1	42
Bilaga C – Evaluatorprompt	43
Bilaga D – Genererad rapport och evaluatorbedömning	45

1 Inledning

1.1 Problemformulering

Molnbaserade driftmiljöer genererar stora mängder loggar, mätvärden, larm och annan telemetridata som beskriver systemens tillstånd över tid. Även om insamling, lagring och visualisering av denna information i hög grad kan automatiseras, kvarstår en central utmaning i att omvandla tekniskt detaljerad driftdata till kortfattade, begripliga och beslutsrelevanta rapporter. Detta är särskilt viktigt i organisationer där tekniska observationer behöver kommuniceras till mottagare som inte själva arbetar operativt med molninfrastruktur, incidenthantering eller systemförvaltning [1], [2].

Stora språkmodeller har under senare år blivit ett möjligt stöd för denna typ av rapportering. Tidigare forskning visar att LLM:er kan användas både inom *Artificial Intelligence for IT Operations* (AIOps) [1], [2], [3] och i lösningar för automatisk generering av tekniska rapporter [2], [3].

Att språkmodeller kan generera rapporter innebär dock inte att deras utdata utan vidare är tillförlitliga. I den aktuella tillämpningen bygger rapporterna huvudsakligen på faktabaserad och strukturerad indata. Problemet gäller därför inte främst fri fabricering av innehåll, utan risken att modellen återger underlaget på ett missvisande sätt, utelämnar viktiga observationer, betonar fel avvikelser eller drar felaktiga slutsatser från verkliga data. Forskning om *trustworthiness*, *hallucinationer* och källförankrad generering visar att språkligt välformulerade svar inte nödvändigtvis är tekniskt korrekta, trogna sitt underlag eller lämpliga som beslutsunderlag [4], [5], [6].

Trots att det finns bred forskning om *trustworthiness*, hallucinationer och AIOps [4], [7], samt nyare arbete om automatisk rapportgenerering i monitorerings-sammanhang [3], saknas fortfarande en tydligt avgränsad och praktiskt orienterad utvärdering av hur olika LLM-upplägg påverkar kvaliteten i faktabaserade tekniska rapporter inom en Microsoft-nära molnkontext. Kunskapsluckan gäller särskilt rapporter som riktas till icke-tekniska beslutsfattare, där teknisk korrekthet, spårbarhet och begriplighet behöver vägas samman.

1.2 Målsättning

Målet med detta examensarbete är att undersöka och utvärdera tillförlitligheten hos LLM-genererade insikter i automatiserad molnrapportering. Arbetet syftar till att

skapa en tydligare förståelse för vilka tekniska val som påverkar om en AI-genererad rapport kan anses vara korrekt, spårbar, begriplig och användbar som beslutsunderlag i en molnbaserad driftkontext.

Ett centralt mål är att operationalisera tillförlitlighet som ett mätbart begrepp och utveckla samt tillämpa ett utvärderingsramverk för att bedöma tillförlitligheten i AI-genererade sammanfattningar av faktabaserad molndata. Arbetets huvudfokus är att fastställa hur val av LLM-upplägg, såsom promptning, *retrieval-augmented generation* (RAG), källhänvisning och kontrollsteg, påverkar kvalitet och tillförlitlighet i sammanfattningar av incident- och resursdata för icke-tekniska beslutsfattare. Därutöver är ett delmål att identifiera vilka designval i en *serverless, multi-tenant* rapportmotor som är mest kritiska för säker dataisolering och spårbarhet när flera kunder hanteras i samma system.

1.3 Avgränsningar

Studien avgränsas till ett begränsat urval av datakällor och rapportmoduler. Det konkreta testscenariot och dess referensunderlag beskrivs i metodkapitlet. Arbetet behandlar därmed inte utveckling av en fullständig produkt, kundportal eller komplett administrationslösning.

Aspekter som *bias* i bred samhällelig mening, multimodala modeller, fullskalig driftsättning och avancerad kostnadsoptimering behandlas inte som huvudfrågor. Den kompletterande delen om *multi-tenant*-arkitektur avgränsas till centrala designfrågor kring dataisolering och spårbarhet.

1.4 Författarnas bidrag till examensarbetet

Examensarbetet genomförs gemensamt av Majd Matloob och Ishar Zeito. Författarna har bidragit i lika stor omfattning till planering, genomförande, analys och rapportskrivning. Arbetets innehåll och slutsatser är därmed resultatet av ett gemensamt arbete.

1.5 Etiskt förhållningssätt

Rapportens författare intygar härmed att rapporten är framtagen utan hjälp av generativ AI än för andra ändamål än språkkontroll

2 Teori och bakgrund

Detta kapitel placerar examensarbetet i ett större tekniskt och vetenskapligt sammanhang. Kapitlet inleds med en bred beskrivning av molnbaserade driftmiljöer och AIOps, för att därefter fördjupas i tillförlitlighet i stora språkmodeller, specifika riskbegrepp som *hallucinationer* och *faithfulness*, samt tekniker för att stärka tillförlitligheten. Kapitlet avslutas med en genomgång av tidigare arbeten och en syntes som identifierar den kunskapslucka examensarbetet adresserar.

2.1 Molnbaserade driftmiljöer och AIOps

Följande avsnitt beskriver den operativa kontexten för examensarbetet. Det behandlar hur molnbaserade system genererar driftdata, vad AIOps innebär och hur stora språkmodeller har kommit att användas inom IT-drift, samt varför en Microsoft-nära miljö utgör en relevant tillämpningskontext.

2.1.1 Driftdata och rapportering

Molnbaserade system genererar kontinuerligt stora mängder driftrelaterad information i form av loggar, mätvärden, larm och annan telemetridata som beskriver systemens tillstånd. I praktiska driftmiljöer används denna information för övervakning, incidenthantering och kapacitetsbedömning. Tidigare forskning visar att *cloud-native* och *DevOps-oriented* miljöer kännetecknas av hög komplexitet, distribuerade komponenter och omfattande datainsamling [1], [2]. Den manuella analysen försvåras dels av den stora volymen data som kontinuerligt genereras, dels av svårigheten att skilja kritiska avvikelser från normala driftvariationer, återkommande informationslarm och andra händelser som inte kräver omedelbar åtgärd, samt av behovet att översätta tekniska observationer till begripliga slutsatser för mottagare utan operativ bakgrund [1], [2]. En stor språkmodell förväntas i detta sammanhang bidra genom att automatiskt sammanfatta och prioritera information samt formulera tekniska observationer på ett sätt som är tillgängligt för icke-tekniska mottagare [2], [3].

2.1.2 AIOps och språkmodeller

AIOps innebär att AI-metoder används för att stödja och automatisera IT-drift. Cisco [8] definierar begreppet som användningen av artificiell intelligens och maskininlärning för att automatisera och förbättra IT-drift genom analys av stora datamängde.

I praktiken bygger detta på analys av operationsdata, det vill säga telemetri i form av loggar, mätvärden, händelser och spåringsdata som beskriver systemens hälsotillstånd.

Tatineni [1] belyser hur AIOps i *cloud-native DevOps*-miljöer används för att korrelera händelser, identifiera avvikelser och förbättra incidenthantering. Zhang et al. [2] ger en bred översikt över hur stora språkmodeller används inom AIOps för uppgifter som loggtolkning, incidentsammanfattning och felanalys. I takt med att stora språkmodeller har utvecklats har användningsområdet breddats från traditionell analys till att även omfatta generering av rapporter och språkbaserad interaktion med tekniska system [2], [3]. Detta gör det möjligt att inte bara analysera data, utan också översätta tekniskt detaljerad information till mer tillgängliga rapporter för mottagare med varierande teknisk bakgrund.

Tidigare forskning understryker dock att LLM-integration i AIOps-miljöer fortfarande möter hinder gällande teknisk *robustness*, det vill säga systemets förmåga att ge stabila och tillförlitliga resultat när indata varierar eller innehåller störningar. Zhang et al. [2] förklarar att en hög känslighet för variationer i indata, exempelvis brus i loggfiler eller ändringar i telemetridatans format, kan leda till inkonsekventa slutsatser som försvårar incidenthantering.

2.1.3 Microsoft-nära kontext

I en Microsoft-nära kontext är automatiserad rapportering särskilt genomförbar eftersom molndrift, identitetshantering och datainsamling redan sker inom ett sammanhängande ekosystem av tjänster som Microsoft Azure [9], Microsoft Entra ID [10] och Microsoft Graph [11]. Dessa kan användas som en gemensam grund för datainsamling, behörighetsstyrning och rapportgenerering utan att data behöver hämtas från separata, oberoende system. Detta minskar komplexiteten i integrationen och gör automatiserad rapportering praktiskt genomförbar i större utsträckning än i fragmenterade miljöer. Eftersom samma ekosystem hanterar både identiteter, behörigheter och driftdata kan rapporteringsflödet automatiseras end-to-end utan manuella överlämningar mellan system [9], [11].

I rapporteringssystem som hanterar flera kunder i en delad molnmiljö, en så kallad *multi-tenant*-arkitektur, blir tillförlitlighet inte enbart en fråga om modellens utdata, utan även om arkitekturens förmåga att säkerställa separation av kunders data, spårbarhet och korrekt behörighetsstyrning [12]. I en *serverless* miljö, där backendfunktioner körs vid behov utan dedikerade servrar [9], tillkommer krav på att identitetshantering och hantering av känsliga uppgifter hanteras korrekt i varje anrop.

2.2 *Hallucinationer, factuality och faithfulness*

Följande avsnitt introducerar de centrala begrepp kring *hallucinationer* och tillförlitlighet som ligger till grund för examensarbetets utvärderingsramverk. Det beskriver hur *hallucinationer* definieras i forskningen, hur de delas in i distinkta dimensioner, vilken betydelse dessa brister har specifikt vid sammanfattning av strukturerad teknisk data, samt hur forskningen typiskt mäter dem.

2.2.1 Definition och dimensioner

Forskningen om *hallucinationer* i stora språkmodeller utgör en central del av det teoretiska ramverket för att förstå modellens opålitlighet. Huang et al. [7] beskriver *hallucinationer* som fall där modellen producerar innehåll som är felaktigt, inte kan verifieras eller brister i trohet mot den givna kontexten. Ji et al. [6] ger en bred kategorisering där fenomenet beskrivs som genererat innehåll som är osannolikt eller strider mot källmaterialet.

Inom forskningsfältet delas problemet in i två huvudsakliga dimensioner: *factuality hallucination*, där innehållet direkt strider mot kända fakta, och *faithfulness hallucination*, där modellen avviker från instruktion, kontext eller logisk sammanhållning [7], [6].

2.2.2 Betydelse för teknisk rapportering

Lin et al. [4] understryker att arbetet med att öka tillförlitligheten kräver systematiska utvärderingsmetoder för att adressera både *bias* och *hallucinationer*. Här avser *bias* systematiska snedvridningar i hur modellen väljer, prioriterar eller formulerar information. Författarna beskriver behovet av granskningsmetoder som går bortom yttlig textkvalitet för att i stället fokusera på innehållsnära kriterier [4].

Vid sammanfattning av strukturerad teknisk data framhåller Huang et al. [7] och Ji et al. [6] att brister i *faithfulness* är vanligt förekommande, eftersom modellen kan sammanfatta underliggande data selektivt eller på ett sätt som leder till tekniskt missvisande slutsatser, trots att den faktiska datan är korrekt. Ett exempel är när en sammanfattning utelämnar kritiska datapunkter eller feltolkar innebörden av en avvikelse.

Forskningen visar således att utelämnanden av kritiska observationer, missvisande slutsatser och direkta sakfel är distinkta typer av brister som kräver separata mätmetoder för att fångas på ett tillförlitligt sätt [7], [6].

2.3 Stora språkmodeller och tillförlitlighet

Stora språkmodeller (*Large Language Models*, LLM) har blivit centrala verktyg för att transformera och kondensera komplexa tekniska informationsmängder. I tekniska rapporteringssystem är tillförlitlighet inte enbart en fråga om språklig kvalitet, utan även om huruvida modellens utdata är korrekt, trogen sitt underlag, spårbar och användbar för mottagaren.

Mot bakgrund av de risker som beskrivs i avsnitt 2.2 behandlas tillförlitlighet (*trustworthiness*) i detta arbete som ett sammansatt begrepp. Utöver *factuality* och *faithfulness* identifierar tidigare forskning flera dimensioner som är relevanta för hur AI-genererade rapporter kan granskas och användas. Lin et al. [4] betonar behovet av spårbarhet, det vill säga att modellens påståenden ska kunna kopplas tillbaka till verifierbara källor. Zhang et al. [2] och Chen et al. [3] lyfter fram konsistens som ett kvalitetskrav i AIOps-sammanhang, där liknande indata och experimentella förutsättningar bör ge stabila utfall. Begriplighet och användbarhet som beslutsstöd utgör mottagarnära kriterier som kompletterar de tekniskt orienterade måtten [3], [4].

Sammantaget kan tillförlitlighet i AI-genererade tekniska rapporter i detta arbete brytas ned i fem kvalitetsdimensioner: *teknisk korrekthet (factuality)*, *trohet mot underlaget (faithfulness)*, *spårbarhet och källförankring*, *konsistens* samt *begriplighet och beslutsstöd*. Dessa dimensioner används senare som grund för utvärderingsramverket i metodkapitlet.

2.4 Tekniker för att stärka tillförlitlighet

Följande avsnitt beskriver de tekniska metoder som används för att styra och stärka tillförlitligheten i LLM-genererade svar, från promptbaserade tekniker till källförankrad generering och dess begränsningar.

2.4.1 Promptning och kontrollsteg

För att styra tillförlitligheten används *prompt engineering*, vilket innebär att instruktioner och kontext struktureras för att påverka modellens beteende [7]. En grundläggande variant är *zero-shot prompting*, där modellen enbart ges en instruktion utan några medföljande exempel och därmed förlitar sig helt på sin tränade kunskap för att producera ett svar. En mer styrd variant är *few-shot prompting*, där modellen ges ett antal konkreta exempelsvar i prompten som visar önskat format och resonemang, vilket minskar risken för att modellen tolkar uppgiften på ett oönskat sätt [7]. Ytterligare ett steg är *Chain-of-Thought (CoT)*, där modellen instrueras att resonera

stegvis mot ett svar snarare än att direkt producera en slutsats, vilket enligt forskningen leder till mer logiska och korrekta sammanfattningar [7]. Genom promptbaserade kontrollsteg kan modellen dessutom instrueras att kritiskt granska sin egen utdata mot källmaterialet för att identifiera eventuella utelämnanden eller felaktigheter före leverans [4].

2.4.2 Källförankring och RAG

En central forskningslinje för att öka tillförlitligheten i stora språkmodeller är källförankring (*grounding*), vilket innebär att modellens generering baseras på externa och verifierbara källor. I stället för att modellen enbart förlitar sig på sin interna parametriska kunskap används ofta metoden RAG [13]. Genom RAG hämtas relevant information från externa dokument eller databaser och skickas med som kontext till modellen innan svaret genereras. Enligt både Microsoft [13] och Hwang et al. [5] skapar detta bättre förutsättningar för spårbarhet och innehållsmässig kontroll, då modellens svar kan härledas direkt till de specifika referensdata som använts.

Hwang et al. [5] belyser vidare att traditionell RAG främst fokuserar på semantisk relevans mellan indata och dokument. Författarna argumenterar för att detta är otillräckligt om källmaterialet varierar i kvalitet och föreslår ett *reliability-aware* RAG-upplägg där källornas pålitlighet skattas och vägs in i genereringsprocessen [5]. Detta adresserar risken att felaktig viktning av källor leder till missvisande slutsatser, vilket enligt forskningen är en kritisk faktor i system som sammanställer information från flera oberoende underlag [5].

2.4.3 Utmaningar med källförankrad generering

Trots fördelarna med RAG betonar forskning att tekniken inte utgör en fullständig garanti för tillförlitlighet [7], [6]. Huang et al. [7] och Ji et al. [6] poängterar att även källförankrade system kan brista i *faithfulness*, då modellen kan sammanfatta underlaget felaktigt, utelämna kritiska datapunkter eller formulera innehållet på ett tvetydigt sätt. För tekniska rapporteringssystem innebär detta att utvärderingen behöver omfatta både korrekt återgivning av underlaget och en verifierbar koppling mellan rapportens påståenden och det källmaterial som använts [5], [7].

Begreppet *citation accuracy* används i forskning för att beskriva i vilken grad ett genererat svar kan kopplas till korrekta och verifierbara referenser i underlaget, och fungerar därigenom som ett mått på spårbarhet och källförankring [5], [7].

2.4.4 Övriga tekniker

Ytterligare tekniker för att stärka tillförlitlighet omfattar exempelvis *finetuning*, där modellen tränas vidare på domänspecifik data för att förbättra precision inom ett avgränsat område [7].

2.5 Tidigare arbeten om automatiserad rapportgenerering

Bland de studerade källorna ligger Chen et al. [3] närmast detta examensarbets tillämpning. Författarna visar att integrationen av kunskapsgrafer (*knowledge graphs*) och stora språkmodeller kan effektivisera genereringen av övervakningsrapporter, och att LLM-baserade system kan reducera den manuella arbetsbördan genom att automatiskt strukturera och sammanfatta komplexa driftdata [3].

Samtidigt fokuserar Chen et al. [3] främst på hur rapportgenerering kan genomföras tekniskt och organisatoriskt. Studien ger ett viktigt bidrag till förståelsen av automatisk rapportgenerering, men behandlar i mindre grad hur rapporternas kvalitet bör bedömas när de används som beslutsunderlag för icke-tekniska mottagare.

2.6 Syntes och identifierad kunskapslucka

Följande avsnitt sammanfattar de tre forskningsstråk som presenterats i detta kapitel och pekar ut den kunskapslucka som examensarbetet adresserar.

2.6.1 Sammanfattning av forskningsläget

De studier som presenterats i detta kapitel kan kategoriseras i tre huvudsakliga forskningsstråk:

- **Teoretisk tillförlitlighet:** Lin et al. [4], Huang et al. [7] och Ji et al. [6] bidrar med definitioner av *trustworthiness*, *factuality* och *faithfulness*. Dessa ramverk tydliggör vilka dimensioner som är centrala vid bedömning av AI-genererat rapportinnehåll.
- **Domänkontext (AIOps):** Tatineni [1], Zhang et al. [2] och Cisco [8] placerar tekniken i en operativ molnkontext. De visar på relevansen av automatiserad analys i *cloud-native* miljöer, men saknar detaljerade metoder för att utvärdera tillförlitligheten i sammanfattningar avsedda för beslutsstöd.
- **Teknisk implementering:** Hwang et al. [5] och Chen et al. [3] bidrar med arkitektoniska komponenter genom *reliability-aware* RAG och automatiserad

rapportgenerering. Dessa ligger närmast den tekniska tillämpningen men adresserar inte specifikt behoven hos icke-tekniska mottagare.

Forskningsläget visar att AI-genererade rapporter kan bedömas utifrån både innehållsnära kriterier, såsom korrekthet och trohet mot underlaget, och mottagarnära kriterier, såsom begriplighet och användbarhet.

2.6.2 Identifierad kunskapslucka

Litteraturstudien visar att det finns omfattande forskning inom de enskilda områdena tillförlitlighet, AIOps och rapportgenerering [1]–[7], [8]. Det framgår däremot inte tydligt hur dessa perspektiv bör integreras och utvärderas i en sammanhängande rapportmotor som sammanfattar faktabaserad incident- och driftdata för icke-tekniska beslutsfattare.

Den identifierade kunskapsluckan gäller därför inte primärt frågan om LLM:er kan användas för teknisk rapportering, utan hur deras utdata bör bedömas när indata redan är faktabaserad och när huvudrisken gäller felaktig sammanfattning, missvisande formulering, selektiv utelämnning och otillräcklig spårbarhet. Det saknas också ett tydligt praktiskt perspektiv på hur sådana utvärderingar kan genomföras i en *serverless* [9] och *multi-tenant* [12] rapportmotor, där dataisolering och spårbarhet utgör viktiga kompletterande designfrågor som inte adresseras i befintlig forskning.

3 Metod

Detta kapitel beskriver den metodik och de lösningsmetoder som används i examensarbetet. Kapitlet redogör för forskningsansats, litteraturstudie, utveckling av prototyp, experimentuppsättning, datainsamling samt utvärderingskriterier. Kapitlet beskriver den färdiga lösningen och det genomförda utvärderingsupplägget snarare än den kronologiska arbetsprocessen.

3.1 Forskningsansats

Examensarbetet tillämpar en *Design Science Research*-inspirerad ansats (DSR). Denna ram syftar till att generera ny kunskap genom att utveckla och utvärdera innovativa tekniska artefakter som löser identifierade problem. I detta arbete utgörs artefakten av en rapportmotor för automatiserad molnrapportering.

Valet av DSR motiveras av behovet att kombinera teoretisk analys med praktisk problemlösning. Ansatsen bedöms lämplig då artefakten fungerar som den primära källan för datainsamling, vilket möjliggör en systematisk undersökning av hur specifika designval påverkar slutresultatets tillförlitlighet. Genom att förena teknisk utveckling med vetenskaplig utvärdering skapas en grund för att dra generella slutsatser om språkmodellens användbarhet i en operativ driftkontext.

3.2 Översikt av tillvägagångssätt

Studiens genomförande är indelat i fem huvudsakliga faser för att säkerställa ett strukturerat och reproducerbart resultat. Inledningsvis genomförs en litteraturstudie som etablerar det teoretiska ramverket och motiverar metodvalen. Därefter konstrueras en prototyp som fungerar som testbädd för de olika LLM-uppläggen. Uppläggen implementeras och utvärderas sedan systematiskt, varefter en kompletterande analys av arkitektur och stödmätningar av prestanda och kostnad genomförs.

3.2.1 Litteraturstudie

En litteraturstudie genomförs som grund för examensarbetets teoretiska ramverk och metodval. Den används för att identifiera tidigare arbeten, aktuell forskning och existerande utvärderingsmetoder inom områdena tillförlitlighet i stora språkmodeller, metoder för att minska hallucinationer, *factuality*, *faithfulness*, *retrieval-augmented generation*, AIOps och människa-AI-interaktion i teknisk rapportering.

Litteraturstudien fyller tre funktioner i arbetet. För det första används den för att formulera den kunskapslucka som examensarbetet adresserar. För det andra används den för att identifiera de kvalitetsdimensioner som ligger till grund för utvärderingsramverket. För det tredje motiverar den valet av LLM-upplägg som jämförs i den experimentella delen [4], [5], [7].

För att identifiera relevant litteratur genomförs systematiska sökningar primärt via Google Scholar med hjälp av specifika söksträngar, såsom “*RAG*” AND “*Reliability*”, kombinerat med AI-verktyget Claude som stöd i den explorativa fasen för att kartlägga terminologi och nyckelpublikationer.

3.2.2 Val av lösningsmetod

Den valda lösningsmetoden bygger på att utveckla en begränsad men fungerande prototyp som testbädd, snarare än att enbart genomföra en teoretisk jämförelse av olika arkitekturer. Skälet är att forskningsfrågan gäller hur kvalitet och tillförlitlighet påverkas av olika LLM-upplägg i en konkret rapportmotor, vilket kräver ett system där samma referensunderlag kan användas under kontrollerade förhållanden.

Alternativa lösningsmetoder övervägdes. En ren litteraturstudie valdes bort eftersom den inte hade varit tillräcklig för att analysera hur de valda LLM-uppläggen fungerar i en faktisk rapportkontext. En fullskalig produktutveckling valdes bort eftersom arbetets syfte inte är att leverera ett komplett kommersiellt system, utan att skapa en kontrollerad miljö för vetenskaplig utvärdering. En ren användarstudie utan teknisk prototyp valdes bort eftersom den inte hade gett möjlighet att jämföra flera tekniska upplägg på ett systematiskt sätt.

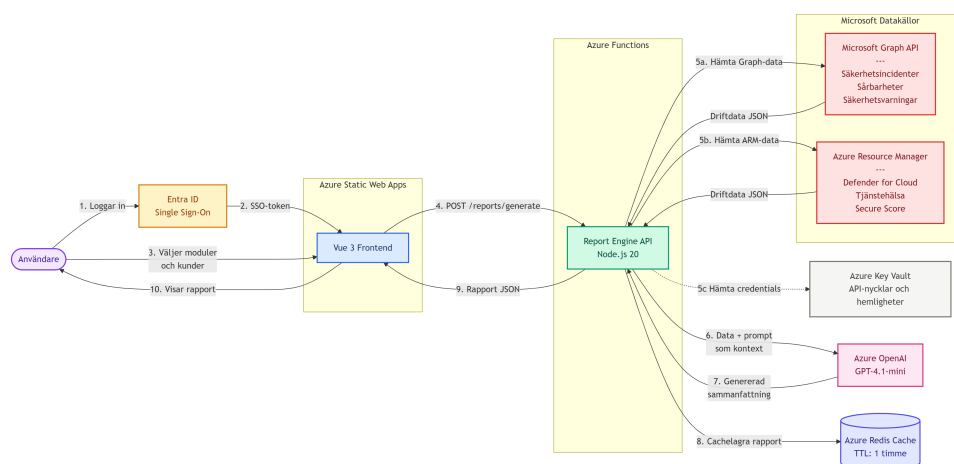
3.3 Systemarkitektur och modellering

Prototypen fungerar som testbädd för utvärderingen och dess arkitektur utformas utifrån tre designprinciper som var nödvändiga för att möjliggöra de kontrollerade experiment som studien kräver. Det övergripande valet av en *serverless*-modell motiveras av att det möjliggör isolerad och oberoende exekvering av varje rapportgenerering, vilket är en förutsättning för reproducerbara och jämförbara experiment. Frontend och backend separeras för att hålla användargränssnitt och rapportlogik fristående, så att förändringar i LLM-uppläggen inte påverkar övriga delar av systemet. Arkitekturen är dessutom modulär, vilket innebär att samma experimentupplägg kan återanvändas för olika datakällor och rapportmoduler utan att den övergripande strukturen behöver förändras.

Microsoft Graph API väljs som primär datakälla eftersom den ger strukturerad och

faktabaserad driftdata från Microsoft 365-miljön, vilket är den typ av indata som studiens forskningsfråga utgår från. Detta säkerställer att skillnader i resultat mellan de fem LLM-uppläggen kan kopplas till upplägget i sig snarare än till variationer i systemets övriga delar.

Dataflödet i prototypen följer en konsekvent kedja från användare till rapport. Användaren autentiseras via Entra ID och väljer kund och rapportmoduler i frontend, som är byggd i Azure Static Web Apps [14]. Anropet vidarebefordras till backend i Azure Functions [9], som hämtar driftdata från Microsoft Graph API och Azure ARM. Datan bearbetas och skickas som kontext till Azure OpenAI, där den valda LLM-prompten appliceras. Den genererade rapporten cachas temporärt i Azure Cache for Redis och returneras till frontend för visning. Hemligheter och autentiseringsuppgifter hanteras separat via Azure Key Vault [15] för att skydda åtkomstuppgifter och stödja separation av kundspecifik åtkomst. En förenklad översikt av detta flöde ges i figur 3.1.



Figur 3.1: Förenklad översikt av prototypens systemarkitektur och dataflöde. Frontend är byggd i Azure Static Web Apps och backend i Azure Functions. Prototypen är implementerad som en *serverless* och *multi-tenant* rapportmotor i Azure.

3.4 Datakällor och experimentuppsättning

Experimentuppsättningen bygger på ett begränsat urval av datakällor och rapportmoduler i enlighet med arbetets avgränsning. En rapportmodul avser här en avgränsad funktionsenhet i prototypen som ansvarar för att hämta, bearbeta och sammanfatta data från en specifik datakälla eller datakategori. Den primära datakällan är Microsoft Graph API, kompletterad med Azure Resource Manager. Tabell 3.1 redovisar de rapportmoduler som ingick i experimentets testscenario.

Tabell 3.1: Rapportmoduler som användes i experimentets testscenario.

Modul	Beskrivning	Datakälla
Security Alerts	Säkerhetsvarningar från M365 Defender	Microsoft Graph
Security Incidents	Säkerhetsincidenter från M365 Defender	Microsoft Graph
Risky Sign-Ins	Riskfyllda inloggningar (<i>Identity Protection</i>)	Microsoft Graph
Vulnerabilities — Critical	Kritiska sårbarheter (<i>Defender Vulnerability Management</i>)	Microsoft Graph
Vulnerabilities — High	Höga sårbarheter (<i>Defender Vulnerability Management</i>)	Microsoft Graph
Service Health Issues	Tjänstehälsoproblem i Microsoft 365	Microsoft Graph
Defender for Cloud Alerts	Säkerhetsvarningar från Defender for Cloud	Azure ARM
Security Recommendations	Säkerhetsrekommendationer från Defender	Azure ARM
Service Issues	Azure-tjänstincidenter	Azure ARM
Subscription Overview	Prenumerationsöversikt med Secure Score	Azure ARM

Chen et al. [5] tillämpar kunskapsgrafer för att strukturera övervakningsdata semantiskt innan rapportgenerering. I detta arbete används istället direkt strukturerad JSON-data från Microsoft Graph API och Azure ARM som RAG-kontext.

För utvärderingen definieras ett testscenario baserat på incident- och resursnära data. För testscenariot skapas ett referensunderlag, här kallat *ground truth*, som beskriver vilka centrala observationer, avvikelser och slutsatser som en korrekt sammanfattning bör innehålla.

Referensunderlaget tas fram manuellt utifrån den faktiska datan och används sedan för att bedöma hur väl de olika LLM-uppläggen återger samma innehåll. Samtliga upplägg utvärderas mot samma referensunderlag, men de får inte samma indata vid generering.

3.5 Jämförda LLM-upplägg

Den huvudsakliga experimentella variabeln i arbetet är valet av tekniskt LLM-upplägg. För att isolera effekten av källförankring och kontrollmekanismer inkluderas två baslinjeupplägg utan källdata samt tre RAG-baserade upplägg med ökande grad av spårbarhet och självgranskning. En viktig distinktion gäller skillnaden mellan promptens instruktionsdel och RAG-underlaget. I samtliga upplägg innehåller prompten instruktioner om hur rapporten ska struktureras och vad som ska sammanfattas. I baslinjeuppläggen saknas dock faktisk driftdata i prompten, vilket innebär att modellen enbart förlitar sig på sin tränade kunskap. I RAG-uppläggen kompletteras instruktionsprompten med den faktiska datan från rapportmodulerna, det vill säga

JSON-strukturerad driftdata från Microsoft Graph och Azure ARM, som skickas med som kontext före generering. Denna distinktion är central för att förstå skillnaderna i resultat mellan uppläggen.

Det *reliability-aware* RAG-ramverk som beskrivs i avsnitt 2.4.2 övervägdes men valdes bort eftersom arbetets datakällor utgjordes av strukturerad driftdata från definierade Microsoft- och Azure-källor. Ramverket är främst relevant när flera externa källor med varierande pålitlighet behöver skattas och viktas mot varandra. Tekniken bedöms därför ligga utanför arbetets avgränsning, men identifieras som en möjlig vidareutveckling. *Chain-of-Thought* (CoT) övervägdes som ett alternativt promptupplägg men valdes bort eftersom tekniken primärt förbättrar modellens logiska resonemang i flerstegsuppgifter snarare än säkerställer fullständig återgivning av strukturerad faktadata. *Finetuning* övervägdes men kräver tillgång till märkt träningsdata och beräkningsresurser som ligger utanför arbetets avgränsning. Ett flerstegsupplägg med separata modeller för generering och granskning övervägdes men bedömdes medföra en komplexitet i implementation och utvärdering som inte är motiverad inom ramen för en jämförande studie med ett begränsat testscenario. Samtliga dessa alternativ identifieras som möjliga vidareutvecklingar.

De fem konfigurationerna som utvärderas är följande. Promptmallarna för respektive upplägg redovisas i bilaga A.

1. **Baslinje (Zero-shot):** Prompten innehåller enbart en instruktion om vad som ska sammanfattas, utan exempel eller extern driftdata. Modellen förlitar sig helt på sin tränade kunskap.
2. **Strukturerad instruktion (Few-shot):** Prompten inkluderar explicita riktlinjer för formatstyrning samt demonstrationer av önskat resultat, men innehåller ingen faktisk driftdata.
3. **Standard RAG:** Instruktionsprompten kompletteras med den faktiska driftdatan från samtliga rapportmoduler som JSON-strukturerad kontext före generering.
4. **Spårbar RAG:** Som föregående, men med tilläggskrav på explicita källhänvisningar i rapporten i formatet [Källa: <modulnamn>, <fält>] för att möjliggöra direkt verifiering mot underlaget.
5. **Validerat upplägg (Självgranskning):** Ett flerstegsupplägg som kombinerar RAG med en automatiserad kontrollfas, där modellen instrueras att validera sin sammanfattning mot källdatan och identifiera eventuella utelämnanden innan rapporten levereras.

3.6 Utvärderingsramverk

3.6.1 Kriterier för tillförlitlighet

För att bedöma tillförlitlighet i de genererade rapporterna används ett utvärderingsramverk som utgår från de fem kvalitetsdimensioner som identifierats och definierats i kapitel 2: *teknisk korrekthet, trohet mot underlaget, spårbarhet och källförankring, konsistens samt begriplighet och beslutsstöd*. Kriterierna används för att analysera hur väl de olika LLM-uppläggen presterar i en praktisk driftkontext.

3.6.2 Mätmetoder och bedömningsförfarande

För att göra kriterierna empiriskt prövbara kopplas varje kvalitetsdimension till ett konkret mätvärde. Teknisk korrekthet mäts genom antal tekniska sakfel, det vill säga påståenden i rapporten som direkt motsäger ett verifierbart värde i referensunderlaget, exempelvis felaktigt antal incidenter eller felaktig statusinformation. Trohet mot underlaget mäts genom antal utelämnade observationer och antal missvisande slutsatser. Spårbarhet och källförankring mäts genom andelen korrekta källhänvisningar. Konsistens mäts genom jämförelse mellan upprepade körningar av samma upplägg. Begriplighet och beslutsstöd mäts separat genom den kvalitativa deltagarutvärdering som beskrivs i avsnitt 3.7.

För testscenariot definieras ett manuellt facit utifrån rådatan i Azure Resource Manager och Microsoft Graph. Facit beskriver vilka observationer, avvikelser och slutsatser som en korrekt rapport förväntas återge och fungerar som fast referens genom hela utvärderingen. Ett exempel på det manuellt framtagna facit som användes i testscenario S1 redovisas i bilaga B.

Själva jämförelsen mellan genererad rapport och facit genomförs med stöd av en separat evaluator-LLM, GPT-4.1-mini, som körs med temperaturvärdet 0.0. I detta sammanhang innebär temperaturvärde graden av slumpmässighet i modellens svarsgenerering; ett värde nära 0 gör svaren mer deterministiska och minskar variationen mellan körningar. Evaluatorn instrueras via ett strukturerat promptformat, redovisat i bilaga C, att bedöma varje rapport mot facit enligt de definierade kriterierna. För varje kriterium returnerar evaluatorn ett numeriskt utfall tillsammans med en kort motivering. Samma evaluator och samma bedömningsprompt används för samtliga LLM-upplägg, vilket skapar en konsekvent jämförelsegrund.

Valet att använda en LLM som evaluator innebär en metodologisk begränsning, eftersom samma typ av system som utvärderas också används som bedömningsverktyg. Samtidigt ligger bedömningsförfarandet i linje med tidigare forskning som beto-

nar behovet av systematiska och innehållsnära utvärderingsmetoder för att granska tillförlitlighet, hallucinationer, tekniska sakfel och trohet mot underlaget [1], [4], [16].

I detta arbete används evaluatorn därför inte som ett fristående sanningsmått, utan som ett konsekvent bedömningsstöd mot ett manuellt framtaget facit. Detta hantearas genom att evaluatorn alltid jämför rapportens påståenden mot samma fasta referensunderlag, att temperaturvärdet sätts till 0.0 för att minska slumpmässighet mellan körningar och att motiveringar sparas för att möjliggöra manuell granskning av enskilda bedömningar. De relativa skillnaderna mellan uppläggen bedöms därmed vara jämförbara, även om de absoluta värdena bör tolkas med viss försiktighet.

3.6.3 Felkategorier och poängschema

Bedömningen genomförs med hjälp av ett poängschema där varje genererad rapport jämförs mot facit. Fyra huvudsakliga mätvärden används:

1. **Tekniska sakfel (*factuality errors*):** antal påståenden i rapporten som direkt motsäger ett verifierbart värde i facit, exempelvis felaktigt resursnamn, felaktigt antal incidenter eller felaktig tidsstämpel. Ett sakfel markeras endast när rapporten anger ett konkret värde som avviker från facit.
2. **Utelämnade observationer (*omissions*):** antal observationer i facit som saknas helt i rapporten. Evaluatorn går igenom facitets observationer en i taget och kontrollerar om motsvarande information återfinns i rapporten, oavsett formulering.
3. **Missvisande slutsatser (*faithfulness errors*):** antal fall där rapporten drar en slutsats som inte stöds av eller direkt motsäger den logiska innebörden i facit. Detta skiljs från tekniska sakfel genom att ett sakfel avser ett specifikt felaktigt värde, medan en missvisande slutsats avser rapportens övergripande tolkning av underlaget.
4. **Spårbarhetsgrad (*citation accuracy*):** andelen källhänvisningar i rapporten som kan kopplas till en korrekt och verifierbar datapunkt i underlaget. Om rapporten saknar källhänvisningar anges spårbarhetsgraden som ej tillämplig.

Poängschemat används för att jämföra de fem LLM-uppläggen på ett enhetligt sätt. De tre första mätvärdena används för samtliga upplägg, medan spårbarhetsgrad endast mäts för de upplägg som genererar explicita källhänvisningar.

3.7 Kvalitativ utvärdering av begriplighet och beslutsstöd

3.7.1 Deltagare och genomförande

En kvalitativ studie genomförs med tre representanter för målgruppen, det vill säga icke-tekniska beslutsfattare, i syfte att bedöma rapporternas praktiska användbarhet i relation till kriteriet *begriplighet och beslutsstöd*.

Deltagarna granskar ett urval rapporter från de olika LLM-uppläggen. Granskningen genomförs blindat, vilket innebär att deltagarna inte informeras om vilket LLM-upplägg som genererat respektive rapport. Detta görs för att undvika att deltagarnas förväntningar eller kännedom om uppläggen påverkar bedömningarna.

Deltagarutvärderingen genomförs som en öppen men styrd bedömning. Det innebär att deltagarna inte följer ett detaljerat frågeformulär, utan först läser respektive rapport och därefter bedömer den utifrån tre fasta parametrar: användbarhet, tydlighet och upplevd tillförlitlighet. De får även lämna en kort fri kommentar om vad som gör rapporten lätt eller svår att använda som beslutsunderlag. Upplägget väljs för att fånga målgruppens spontana förståelse och förtroende för rapporterna, samtidigt som de fasta bedömningsparametrarna gör resultaten jämförbara mellan rapporterna.

Deltagarna informeras om studiens syfte och samtycker till medverkan men önskar vara anonyma, varför ytterligare personuppgifter inte registreras.

3.7.2 Bedömningsskala och analys

Deltagarna skattar rapporterna utifrån tre parametrar på en 5-gradig Likert-skala. Likert-skalan används eftersom den möjliggör en strukturerad och jämförbar kvantifiering av subjektiva upplevelser, såsom hur begriplig, användbar eller tillförlitlig en rapport uppfattas. Skalan bygger på att respondenter tar ställning längs en ordinal skala och är därför lämplig för att fånga mottagarnära bedömningar utan att deltagarna behöver teknisk expertkunskap [16]. Parametrarna är valda för att fånga de mottagarnära aspekter av tillförlitlighet som inte kan mätas genom automatiserad innehållsanalys, nämligen hur rapporten upplevs och används av den tänkta målgruppen [4], [3].

1. **Användbarhet:** hur väl rapporten identifierar de mest kritiska händelserna för verksamheten.
2. **Tydlighet:** hur lättförståeligt de tekniska avvikelserna är beskrivna.

3. **Upplevd tillförlitlighet:** i vilken grad rapporten upplevs som ett pålitligt underlag för ett operativt beslut.

Användbarhet och tydlighet är etablerade parametrar för att mäta begriplighet i studier av AI-genererat beslutsunderlag [3]. Upplevd tillförlitlighet mäts som en separat parameter trots att tillförlitlighet även utvärderas kvantitativt i avsnitt 3.6, eftersom de två måtten fångar olika saker. Den kvantitativa utvärderingen mäter objektiv korrekthet i rapportens innehåll, medan upplevd tillförlitlighet fångar om en icke-teknisk mottagare faktiskt litar på och agerar utifrån rapporten. En rapport kan vara tekniskt korrekt men ändå uppfattas som otrovärdigt formulerad, eller tvärtom. Denna distinktion är central för att bedöma rapporternas värde som beslutsstöd i praktiken [4].

Syftet är att undersöka om de tekniskt starkaste uppläggen också ger rapporter som uppfattas som tydliga och användbara av den tänkta målgruppen, och på så sätt koppla kriteriet *begriplighet och beslutsstöd* till en konkret bedömningsmetod.

3.8 Kompletterande analys av *multi-tenant*-arkitektur

3.8.1 Analysfokus

Utöver utvärderingen av språkmodellernas tillförlitlighet genomförs en kompletterande analys av prototypens arkitektur. Analysen syftar till att identifiera hur specifika designval i rapportmotorn påverkar kraven på dataisolering och spårbarhet i en behovsstyrd, flerkunds-arkitektur, det vill säga en miljö där backend-funktioner körs vid behov utan dedikerade servrar (*serverless*) och där flera kunders data hanteras inom samma system (*multi-tenant*).

Denna del utgör inte en separat säkerhetsstudie, utan fungerar som ett kompletterande analysmoment till huvudfrågan om tillförlitlighet i rapportmotorn som helhet.

3.8.2 Analyserade designval

Analysen genomförs som en strukturerad genomgång där prototypens implementering jämförs med de principer för dataisolering, identitetshantering och spårbarhet som beskrivs i Microsofts riktlinjer för *multi-tenant*-arkitektur på Azure [12]. Följande designval analyseras:

- **Rollmappning via *rolesSource*:** hur identiteter från Entra ID översätts till applikationsspecifika roller.

- **Centraliserad behörighetslagring:** lagring av modulbehörigheter i Azure Table Storage [17].
- **Logisk kundavgränsning via *tenantId*:** hur kundidentifierare förs vidare genom moduler och API-anrop.
- **Volatil lagringsstrategi i Redis:** hur temporär lagring begränsar informationens livslängd [18].
- **Isolering av hemligheter i Key Vault:** hur känsliga autentiseringsuppgifter separeras från applikationskoden [15].

3.9 Stödmätningar av prestanda och kostnad

Följande avsnitt beskriver de stödmätningar som genomförs för att bedöma prototypens operativa genomförbarhet. Mätningarna är inte en del av huvudutvärderingen men ger nödvändig kontext för att tolka resultaten i relation till praktisk användning i en AIOps-miljö.

Som ett komplement till huvudutvärderingen genomförs mätningar av prestanda och resursförbrukning i Azure [9]. Dessa mätningar används för att tolka prototypens praktiska genomförbarhet utifrån de tekniska och operativa ramverk som beskrivits i kapitel 2. Följande variabler mäts och motiveras:

- **Svarstid vid rapportgenerering (*Latency*):** Mäts för att utvärdera systemets användbarhet i de tidskritiska *DevOps*-miljöer som beskrivs i avsnitt 2.1. En för hög latens skulle begränsa lösningens värde som operativt beslutsstöd vid incidenthantering.
- **Effektivitet i API-anrop och cache-utnyttjande:** Genom att mäta antalet anrop till externa tjänster (Azure och Microsoft Graph [11]) utvärderas prototypens tekniska *robustness* [2]. Detta mätvärde är kritiskt för att bedöma hur väl arkitekturen hanterar den datakomplexitet som beskrivs i avsnitt 2.1.
- **Operativ körkostnad per rapport:** En ekonomisk värdering av lösningen genomförs för att bedöma dess rimlighet i en AIOps-kontext [8]. Detta motiveras även av det etiska perspektivet i kapitel 1.5, där resursförbrukning ställs i relation till det skapade värdet för verksamheten.

Då prototypen bygger på en *serverless* arkitektur [9] är dessa mätningar nödvändiga för att identifiera flaskhalsar och skalbarhetsbegränsningar som inte framgår av enbart en kvalitativ innehållsanalys. Resultaten ger en indikation på hur väl lösningen lämpar sig för övergång från en testbädd till en produktionsnära driftmiljö.

4 Resultat

I detta kapitel presenteras resultaten från den empiriska utvärderingen av de fem LLM-uppläggen, den kompletterande analysen av *multi-tenant*-arkitekturen samt stödmätningarna av prestanda och kostnad. Resultaten redovisas i samma ordning som metodkapitlet, baseras på testscenario S1 vars upplägg och referensunderlag beskrivs i avsnitt 3.4–3.6, och fokuserar på observerade utfall. Samtliga kvantitativa värden utgör medelvärden över tre körningar per upplägg. Ett exempel på en genererad rapport samt evaluators bedömning redovisas i bilaga D.

4.1 Kvantitativ utvärdering av LLM-uppläggen

Skillnaderna mellan baslinje och källförankrade upplägg framgår i tabell 4.1.

Tabell 4.1: Kvantitativa resultat per LLM-upplägg. Lägre värden är bättre för sakfel, utelämnanden och missvisande slutsatser. Högre värden är bättre för spårbarhet.

Nr	Upplägg	Sakfel	Utelämn.	Missvis. sluts.	Spårbarhet (%)
1	Zero-shot (baslinje)	3,0	4,7	2,0	—
2	Few-shot	3,0	5,0	2,3	—
3	Standard RAG	1,0	3,0	1,0	—
4	Spårbar RAG	1,0	3,0	1,0	80
5	Validerat	1,0	3,0	1,0	80

4.1.1 Teknisk korrekthet (*factuality errors*)

Zero-shot och few-shot genererade i genomsnitt 3,0 sakfel per körning. Samtliga tre RAG-baserade upplägg genererade 1,0 sakfel per körning, vilket motsvarar en minskning med 67 % jämfört med baslinjen. Facit innehöll totalt 15 verifierbara påståenden, vilket innebär att 3,0 sakfel motsvarar en felfrekvens på 20 % för baslinjeuppläggen och 1,0 sakfel motsvarar 7 % för de RAG-baserade uppläggen.

De sakfel som identifierades i zero-shot-körningarna gällde bland annat att modellen angav att inga aktiva dataintrång förelåg, trots att underlaget innehöll en incident av typen *Possible data exfiltration*, samt att tjänstehälsan beskrevs som stabil trots ett pågående incidentärende för SharePoint Online.

I few-shot-körningarna gällde sakfelen felaktigt återgivna kvantiteter, där rapporterna angav siffror som inte motsvarade referensunderlagets antal incidenter, varningar och sårbarheter.

I samtliga RAG-baserade körningar gällde det återkommande sakfelet antalet högallvarliga varningar, där modellen angav fem till sex i stället för fyra.

4.1.2 Trohet mot underlaget (*omissions och faithfulness errors*)

Referensdata innehöll sju centrala observationer. Zero-shot utelämnade i genomsnitt 4,7 av dessa per körning och few-shot 5,0. Samtliga RAG-baserade upplägg utelämnade i genomsnitt 3,0 observationer per körning, en minskning med 36 % jämfört med zero-shot.

De observationer som utelämnades i samtliga upplägg var: den pågående MFA fatigue-attacken, detaljer kring sårbarheten CVE-2026-1237 (CVSS 10.0, 15 exponerade maskiner) samt indikationen om potentiell kryptomining på resursen VM-Dev-03.

Zero-shot producerade i genomsnitt 2,0 missvisande slutsatser per körning, few-shot 2,3 och samtliga RAG-baserade upplägg 1,0 per körning.

4.1.3 Spårbarhet och källförankring (*citation accuracy*)

Spårbarhetsgraden mättes för upplägg 4 och 5, vilka som ensamma inkluderade explicita källhänvisningar. Tabell 4.2 redovisar spårbarhetsgraden för dessa två upplägg.

Tabell 4.2: *Spårbarhetsgrad* per scenario för upplägg 4 och 5.

Testscenario	Spårbar RAG (%)	Validerat upplägg (%)
S1	80	80
Medel	80	80

Båda uppläggen producerade konsekvent fem källhänvisningar per körning, varav fyra bedömdes som korrekta, vilket gav en spårbarhetsgrad på 80 %. Hänvisningarna följde formatet [Källa: <modulnamn>, <fält>], exempelvis [Källa: Security Incidents, severity=high/status=active]. Den felaktiga hänvisningen i samtliga körningar gällde antalet högallvarliga varningar, där modellen angav fem i stället

för fyra. Felet låg i modellens kvantitativa tolkning av underlaget, inte i hänvisningens form eller källa. Skillnaden i spårbarhetsgrad mellan upplägg 4 och 5 var obefintlig eftersom båda uppnådde 80 %.

4.1.4 Konsistens vid upprepade körningar

Tabell 4.3: *Konsistens* vid upprepade körningar, uttryckt som antal innehållsskillnader mellan tre körningar per LLM-upplägg.

LLM-upplägg	Bedömd som konsistent	Innehållsskillnader
1. Zero-shot	Nej	3
2. Few-shot	Nej	3
3. Standard RAG	Nej	3
4. Spårbar RAG	Nej	3
5. Validerat	Nej	2

Inget upplägg producerade helt identiska resultat över tre körningar (tabell 4.3). Zero-shot uppvisade motstridiga påståenden mellan körningar gällande huruvida dataintrång hade skett och om incidenter var hanterade. Few-shot uppvisade strukturell konsistens men varierade i detaljbeskrivningar mellan körningar. De RAG-baserade uppläggen uppvisade skillnader främst i vilka Defender for Cloud-varningar som explicit nämndes. Validerat upplägg uppvisade två innehållsskillnader, övriga upplägg tre.

4.2 Kvalitativ utvärdering — begriplighet och beslutsstöd

Tre representanter för den tänkta målgruppen granskade rapporter genererade av de fem LLM-uppläggen och bedömde dem utifrån användbarhet, tydlighet och upplevd tillförlitlighet på en femgradig Likert-skala. Resultaten redovisas per deltagare och som medelvärden, kompletterade med skriftliga kommentarer.

De skriftliga kommentarerna samlades in som fria svar för att fånga deltagarnas spontana upplevelse av rapporternas begriplighet och användbarhet.

4.2.1 Deltagaröversikt

Tabell 4.4 redovisar de tre deltagarna i den kvalitativa utvärderingen. Samtliga representerade icke-tekniska verksamhetsroller utan operativ erfarenhet av IT-säkerhet eller molninfrastruktur. Gruppen bestod av två kvinnor och en man.

Tabell 4.4: Deltagare i den kvalitativa utvärderingen.

Deltagare	Roll	Teknisk bakgrund
D1	Marknadsansvarig	Låg
D2	Ekonomiansvarig	Låg
D3	HR-ansvarig	Låg

4.2.2 Resultat per bedömningsparameter

Tabell 4.5 redovisar de individuella betyg som varje deltagare tilldelade respektive LLM-upplägg per bedömningsparameter.

Tabell 4.5: Individuella Likert-betyg per deltagare, upplägg och bedömningsparameter (skala 1–5).

LLM-upplägg	Deltagare	Användbarhet	Tydlighet	Upplevd tillförlitlighet
1. Zero-shot	D1	2	2	2
	D2	2	2	1
	D3	3	2	2
2. Few-shot	D1	3	3	2
	D2	2	2	2
	D3	3	3	2
3. Standard RAG	D1	4	4	3
	D2	3	3	3
	D3	4	4	4
4. Spårbar RAG	D1	4	4	5
	D2	4	4	5
	D3	5	4	4
5. Validerat	D1	4	4	5
	D2	4	4	5
	D3	5	5	4

De sammanvägda deltagarbedömningarna presenteras i tabell 4.6.

Tabell 4.6: Medelvärde per LLM-upplägg och bedömningsparameter (Likert-skala 1–5, där 5 är bäst).

LLM-upplägg	Användbarhet	Tydlighet	Upplevd tillförlitlighet
1. Zero-shot (baslinje)	2,3	2,0	1,7
2. Few-shot	2,7	2,7	2,0
3. Standard RAG	3,7	3,7	3,3
4. Spårbar RAG	4,3	4,0	4,7
5. Validerat (självgranskning)	4,3	4,3	4,7

Skillnaden i upplevd tillförlitlighet var störst mellan Standard RAG (3,3) och de käll-

hänvisande uppläggen (4,7).

4.2.3 Kvalitativa observationer

Följande citat är representativa för deltagarnas skriftliga kommentarer under utvärderingen.

Zero-shot och few-shot:

- D1: *"Rapporten känns väldigt generell. Jag förstår inte vad jag faktiskt ska göra eller prioritera."*
- D2: *"Det står att läget är under kontroll, men jag vet inte på vilken grund det påstås. Jag skulle inte våga fatta ett beslut baserat på det här."*
- D3: *"Siffrorna stämmer inte alltid med varandra mellan olika delar av rapporten, vilket gör den svår att lita på."*

Standard RAG:

- D1: *"Den här rapporten är mycket tydligare. Jag ser vilka incidenter som är aktiva och förstår att något behöver hanteras."*
- D2: *"Bättre struktur, men jag vet fortfarande inte var informationen kommer ifrån. Det hade känts tryggare om det stod."*
- D3: *"Jag kan använda det här som underlag för att eskalera till IT, men jag skulle vilja kunna peka på en källa."*

Spårbar RAG och Validerat upplägg:

- D1: *"Nu förstår jag var varje påstående kommer ifrån. Det gör att jag faktiskt kan använda rapporten i ett möte."*
- D2: *"Källhänvisningarna gör stor skillnad. Det känns som att rapporten går att verifiera, även om jag inte kan göra det själv."*
- D3: *"Det validerade upplägget var tydligast. Rekommendationerna kändes välgrundade och inte bara påhittade."*

4.3 Kompletterande analys av *multi-tenant*-arkitektur

Den kompletterande arkitekturanalysen sammanfattas i tabell 4.7.

Tabell 4.7: Sammanställning av analyserade designval och deras bidrag till dataisolering och spårbarhet.

Designval	Dataisolering	Spårbarhet
Rollmappning (<i>rolesSource</i>)	Begränsar åtkomst per roll	Användaridentitet loggas per rapport
Behörighetslagring (Table Storage)	Modulnivå-isolering per användare	Behörighetskontroll på API-nivå
Kundavgränsning (<i>tenantId</i>)	Korrekt ID per extern API	Felaktiga kombinationer blockeras
Volatil cache (Redis + TTL)	Automatisk radering av kunddata	Temporär lagringstid begränsar exponering
Key Vault (Managed Identity)	Per-kund-separering av hemligheter	Ingen hemlighet i kod eller konfiguration

4.4 Stödmätningar — prestanda och kostnad

Mätresultaten för latens och tokenförbrukning presenteras i tabell 4.8.

Tabell 4.8: LLM-anropslatens och tokenförbrukning per upplägg, medelvärde över tre körningar av scenario S1.

LLM-upplägg	Tokens in	Tokens ut	Totalt tokens	Latens (ms)
1. Zero-shot	101	251	352	5 835
2. Few-shot	263	188	451	3 891
3. Standard RAG	1 326	237	1 563	5 186
4. Spårbar RAG	1 427	320	1 747	8 474
5. Validerat	1 506	322	1 828	6 187

Den uppskattade kostnadsbilden presenteras i tabell 4.9.

Tabell 4.9: Uppskattad LLM-kostnad per rapportsammanfattning och upplägg, angiven i amerikanska dollar (USD).

Nr	Upplägg	In	Ut	Input (USD)	Output (USD)	Total (USD)
1	Zero-shot	101	251	0,000040	0,000402	0,000442
2	Few-shot	263	188	0,000105	0,000301	0,000406
3	Standard RAG	1 326	237	0,000530	0,000379	0,000910
4	Spårbar RAG	1 427	320	0,000571	0,000512	0,001083
5	Validerat	1 506	322	0,000602	0,000515	0,001118

4.5 Sammanställning

En samlad översikt över uppläggens resultat ges i tabell 4.10.

Tabell 4.10: Samlad jämförelse av LLM-uppläggens resultat. ZS = Zero-shot, FS = Few-shot, SR = Standard RAG, SPR = Spårbar RAG, VAL = Validerat upplägg. Kriteriet *Konsistens* avser antal innehållsskillnader mellan upprepade körningar, där lägre värde är bättre. Kostnad anges i USD.

Kriterium	ZS	FS	SR	SPR	VAL
Sakfel	3,0	3,0	1,0	1,0	1,0
Utelämnanden	4,7	5,0	3,0	3,0	3,0
Missv. sluts.	2,0	2,3	1,0	1,0	1,0
Spårbarhet (%)	—	—	—	80	80
Konsistens	3	3	3	3	2
Begriplighet	2,3	2,7	3,7	4,3	4,3
Tillförlitlighet	1,7	2,0	3,3	4,7	4,7
Latens (ms)	5 835	3 891	5 186	8 474	6 187
Kostnad (USD)	0,000442	0,000406	0,000910	0,001083	0,001118

5 Analys och diskussion

Detta kapitel analyserar och diskuterar de resultat som presenterades i kapitel 4 i relation till examensarbetets forskningsfrågor, det teoretiska ramverk som etablerades i kapitel 2 samt de metodologiska val som beskrevs i kapitel 3. Kapitlet behandlar först huvudfrågan om LLM-uppläggens påverkan på kvalitet och tillförlitlighet, därefter delfrågan om *multi-tenant*-arkitektur, och avslutas med en diskussion om metodens styrkor och begränsningar.

5.1 Källförankringens betydelse för objektiv innehållskvalitet

Resultaten visar att tillgång till faktisk källdata via RAG var avgörande för den objektiva innehållskvaliteten i de genererade rapporterna. Baslinjeuppläggen utan källdata användes främst som jämförelsepunkter för att synliggöra effekten av källförankring, snarare än som praktiskt rimliga alternativ för rapportgenerering. Att zero-shot och few-shot presterade svagare är därför väntat, eftersom dessa upplägg saknade tillgång till den driftdata som skulle sammanfattas. Detta stämmer väl överens med det som Huang et al. [7] och Ji et al. [6] beskriver om *faithfulness hallucination*: när modellen saknar tillgång till faktisk källdata tenderar den att formulera generella och till synes rimliga påståenden som inte nödvändigtvis stämmer överens med underlaget.

Den mer relevanta observationen är att samtliga RAG-baserade upplägg minskade antalet sakfel, utelämnanden och missvisande slutsatser jämfört med baslinjerna. Detta visar att källförankring är en nödvändig grund för faktabaserad molnrapportering, men inte en tillräcklig garanti för fullständig tillförlitlighet. Trots tillgång till samma strukturerade underlag kvarstod i genomsnitt tre utelämnade observationer per körning även i de RAG-baserade uppläggen. De observationer som konsekvent utelämnades, däribland MFA fatigue-attacken och detaljer kring den kritiska sårbarheten CVE-2026-1237, tillhörde alla kategorier som kräver att modellen aktivt prioriterar och lyfter fram information snarare än att enbart sammanfatta. Detta pekar på en begränsning i hur väl nuvarande RAG-upplägg hanterar prioritering av säkerhetskritisk information, vilket är en fråga som Hwang et al. [5] adresserar med sitt *reliability-aware* RAG-ramverk.

5.2 Källhänvisningarnas roll för upplevd tillförlitlighet

En av studiens mest framträdande fynd är den stora skillnaden i upplevd tillförlitlighet mellan Standard RAG och de källhänvisande uppläggen. Trots att de tre RAG-

baserade uppläggen uppvisade identiska resultat i den kvantitativa utvärderingen, erhöll Spårbar RAG och Validerat upplägg ett medelvärde på 4,7 för upplevd tillförlitlighet jämfört med 3,3 för Standard RAG. Deltagarnas kommentarer tyder på att källhänvisningar inte i första hand fungerade som ett tekniskt verktyg för verifiering, utan som en signal om transparens som ökade förtroendet för rapportens innehåll.

Detta resultat är intressant i relation till det som Lin et al. [4] beskriver om behovet av spårbarhet i AI-genererade system. Spårbarhet diskuteras i forskningen primärt som ett tekniskt krav för att möjliggöra granskning och revision av modellens utdata. Resultaten i detta arbete antyder att spårbarhet även har ett kommunikativt värde för icke-tekniska mottagare: möjligheten att se varifrån ett påstående kommer ökar mottagarens förtroende för rapporten även när mottagaren inte har förmåga eller avsikt att faktiskt följa upp referensen. Detta är ett fynd som kompletterar den tekniska definitionen av spårbarhet med ett användarperspektiv som inte tydligt framgår av den granskade litteraturen.

5.3 Självgranskning och konsistens

Validerat upplägg skiljde sig från Spårbar RAG enbart i att det uppvisade något lägre variation mellan körningar, med två innehållsskillnader jämfört med tre. I samtliga övriga kvalitetsdimensioner var uppläggen likvärdiga. Detta tyder på att den automatiserade kontrollfasen i första hand bidrar till stabilitet snarare än till kvalitetsförbättring i absoluta termer.

En möjlig förklaring är att de kvarstående bristerna, framför allt utelämnanden, inte beror på att modellen aktivt gör fel utan på att den inte inkluderar information som den inte har tillräckliga incitament att prioritera. Självgranskning kan identifiera direkta motsägelser och sakfel men är inte lika effektiv för att identifiera vad som saknas i en sammanfattning. Detta är konsistent med den distinktion som Huang et al. [7] gör mellan *factuality errors*, som är mer synliga och lättare att identifiera, och *omissions*, som kräver jämförelse mot ett komplett referensunderlag.

5.4 Baslinjernas roll i utvärderingen

Zero-shot och few-shot användes som baslinjer för att skapa en enkel jämförelsepunkt mot de källförankrade uppläggen. Resultaten från dessa upplägg bör därför inte tolkas som ett centralt bidrag i sig, utan som en bekräftelse på att promptstruktur utan faktisk driftdata är otillräcklig för faktabaserad rapportering. Den fortsatta analysen bör därför främst förstås utifrån skillnaderna mellan Standard RAG, Spår-

bar RAG och Validerat upplägg, där samtliga hade tillgång till samma källunderlag men skilde sig åt i fråga om spårbarhet och självgranskning.

5.5 Tillförlitlighet bortom genereringssteget

Den kompletterande arkitekturanalysen visade att tillförlitlighet i en *multi-tenant* rapportmotor inte enbart är en fråga om modellens utdata utan även om arkitekturens förmåga att säkerställa att rätt data når rätt modell på rätt sätt. De designval som bedömdes som mest kritiska adresserar alla risken för att kunddata sammanblandas eller exponeras felaktigt i genereringsflödet: logisk kundavgränsning via *tenantId*, isolering av hemligheter i Key Vault och rollmappning via *rolesSource*. Om dessa mekanismer brister påverkas inte bara säkerheten utan även rapportens tillförlitlighet, eftersom modellen i ett sådant fall kan generera sammanfattningar baserade på fel kunds data.

Detta är en dimension av tillförlitlighet som sällan diskuteras i den granskade litteraturen om LLM-kvalitet, vilken primärt fokuserar på modellens beteende givet en given indata. Resultaten antyder att tillförlitlighet i praktiska system behöver förstås som ett bredare begrepp som inkluderar hela kedjan från datainsamling till rapportleverans, inte enbart genereringssteget.

5.6 Metodologisk diskussion

Studiens design med ett testscenario och tre kvalitativa deltagare möjliggör kontrollerade jämförelser men begränsar generaliserbarheten. De kvantitativa resultaten är internt konsistenta eftersom samtliga upplägg utvärderades mot samma referensunderlag och bedömdes med samma evaluator. De absoluta värdena, exempelvis exakt antal utelämnanden per körning, bör dock tolkas med hänsyn till att ett enda scenario ligger till grund. Det är möjligt att resultaten varierar för andra typer av incidentdata, andra allvarlighetsprofiler eller andra organisatoriska kontexter.

Valet att använda en LLM som evaluator är en metodologisk begränsning som diskuteras i kapitel 3. Eftersom samma typ av system utvärderas och bedömer, finns en risk för systematiska blinda fläckar i bedömningen. Att evaluatorn tillämpades konsekvent på samtliga upplägg och returnerade motiveringar per kriterium minskar risken för att denna begränsning snedvrider den relativa jämförelsen, men den bör beaktas vid tolkning av de absoluta värdena.

Även mätningen av konsistens bör tolkas med försiktighet. De uppmätta innehållsskillnaderna kan bero på faktiska variationer mellan genererade rapporter, men ock-

så på att evaluatören bedömt likvärdiga formuleringar på olika sätt. Evaluatören kördes med temperaturvärdet 0.0 för att minska denna risk, men variation i bedömningen kan inte helt uteslutas.

Den kvalitativa studiens deltagarantal på tre är lågt och resultaten bör betraktas som indikativa. Likert-skalans värden ger en strukturerad bild av deltagarnas upplevelser men fångar inte nyanser som skulle kräva ett mer omfattande kvalitativt upplägg. Resultaten är konsistenta med de kvantitativa fynden, vilket stärker deras trovärdighet, men en studie med fler deltagare och bredare yrkesmässig variation vore önskvärd för att stärka slutsatserna om begriplighet och beslutsstöd.

5.7 Samhälleliga konsekvenser

Detta avsnitt diskuterar examensarbetets konsekvenser ur ekonomisk, social, etisk och miljömässig synvinkel.

5.7.1 Ekonomiska konsekvenser

Resultaten visar att de mest tillförlitliga LLM-uppläggen, Spårbar RAG och Validerat upplägg, kostar under 0,0012 USD per rapport. Vid en volym om 1 000 rapporter per månad uppgår LLM-kostnaden till drygt en dollar, vilket är försumbart i förhållande till den arbetstid som sparas genom automatisering av manuell rapportering. För organisationer som i dag lägger betydande tid på att manuellt sammanställa och kommunicera driftinformation till icke-tekniska beslutsfattare kan automatiserad molnrapportering innebära en påtaglig kostnadsbesparing. Samtidigt tillkommer infrastrukturkostnader för molntjänster som Azure Functions och Azure Cache for Redis, vilka behöver vägas in vid en fullständig ekonomisk värdering.

5.7.2 Sociala konsekvenser

Automatiserad rapportering som riktas till icke-tekniska beslutsfattare kan demokratisera tillgången till teknisk information inom organisationer. När tekniska observationer görs begripliga och tillgängliga för roller som marknadsansvariga, ekonomi-ansvariga och HR-ansvariga kan fler personer i en organisation delta i informerade beslut om IT-säkerhet och driftprioritering. Detta kan minska informationsasymmetrin mellan tekniska och icke-tekniska roller. Samtidigt finns en risk att beslutsfattare överlitar på AI-genererade rapporter utan att kritiskt granska dem, vilket understryker vikten av de transparensmekanismer, som källhänvisningar och spårbarhet, som studien visat har störst påverkan på upplevd tillförlitlighet.

5.7.3 Etiska konsekvenser

Den främsta etiska risken med AI-genererad rapportering är att felaktiga eller missvisande sammanfattningar kan leda till felaktiga beslut i tidskritiska situationer, exempelvis vid incidenthantering. Studien visar att samtliga upplägg kvarstod med utelämnanden, vilket innebär att kritiska observationer kan gå förlorade även i de bästa konfigurationerna. I en *multi-tenant*-miljö tillkommer etiska aspekter kring dataisolering och hantering av kundspecifik information, där brister i arkitekturen kan leda till att känslig data exponeras för fel mottagare.

5.7.4 Miljömässiga konsekvenser

Användning av stora språkmodeller medför en energiförbrukning som är väsentligt högre än traditionella regelbaserade system. RAG-baserade upplägg förbrukar upp till fem gånger fler tokens än baslinjeuppläggen, vilket direkt korrelerar med ökad beräkningsenergi. I en AIOps-kontext där rapporter genereras kontinuerligt för många kunder kan den kumulativa energiförbrukningen bli betydande. Mot detta ska ställas det värde som skapas genom effektivisering av manuellt arbete. Valet av LLM-upplägg bör därför även väga in energieffektivitet som ett kriterium, särskilt i storskaliga driftsättningar där skillnaden i tokenförbrukning mellan upplägg ger ett mätbart miljömässigt avtryck.

6 Slutsatser

Detta kapitel besvarar examensarbetets forskningsfrågor, sammanfattar arbetets bidrag och beskriver möjliga riktningar för fortsatt arbete.

6.1 Svar på huvudfrågan

Examensarbetets huvudfråga gäller hur val av LLM-upplägg påverkar kvalitet och tillförlitlighet i AI-genererade sammanfattningar av incident- och resursdata för icke-tekniska beslutsfattare.

Resultaten visar att tillgång till faktisk källdata via RAG var den viktigaste faktorn för objektiv innehållskvalitet. Baslinjeuppläggen utan källdata visade att promptstruktur i sig inte är tillräcklig för faktabaserad rapportering. Den centrala jämförelsen gäller därför de tre RAG-baserade uppläggen, där Standard RAG, Spårbar RAG och Validerat upplägg uppvisade samma nivå av sakfel och utelämnanden, men skilde sig åt i spårbarhet, upplevd tillförlitlighet och konsistens.

Explicita källhänvisningar hade störst betydelse för upplevd tillförlitlighet. Spårbar RAG och Validerat upplägg erhöll 4,7 av 5 jämfört med 3,3 för Standard RAG, vilket tyder på att källhänvisningar främst fungerar som transparensignal snarare än som tekniskt verifieringsverktyg. Självgranskning gav endast marginell förbättring i konsistens.

Eftersom även de RAG-baserade uppläggen kvarstod med utelämnanden bör AI-genererade rapporter betraktas som beslutsstöd, inte som fullständiga redogörelser.

6.2 Svar på delfrågan om *multi-tenant*-arkitektur

Analysen visar att de mest kritiska designvalen för säker dataisolering i en *serverless*, *multi-tenant* rapportmotor är logisk separation av kundidentiteter i API-anrop, isolering av autentiseringsuppgifter från applikationskod samt rollmappning för att begränsa åtkomst per användare. I prototypen realiserades detta genom *tenantId*-avgränsning, Azure Key Vault och rollmappning via *rolesSource*. Volatil cachelagring och centraliserad behörighetslagring bidrar främst till spårbarhet och begränsad dataexponering.

Rangordningen bygger på en kvalitativ analys av designvalens funktion och är inte empiriskt testad i samma utsträckning som LLM-uppläggen.

6.3 Bidrag till kunskapsutvecklingen

Arbetet bidrar med ett praktiskt utvärderingsramverk för tillförlitlighet i LLM-genererade tekniska rapporter. Tillförlitlighet operationaliseras som fem mätbara kvalitetsdimensioner: teknisk korrekthet, trohet mot underlaget, spårbarhet, konsistens samt begriplighet och beslutsstöd. Därmed adresserar arbetet kunskapsluckan kring hur faktabaserad AI-rapportering kan utvärderas i en *serverless, multi-tenant* molnkontext riktad till icke-tekniska mottagare.

6.4 Begränsningar

Studien bygger på ett testscenario, simulerad kunddata och tre kvalitativa deltagare. Resultaten bör därför tolkas som indikativa snarare än generaliserbara. Eftersom evaluatören också är en LLM bör de absoluta mätvärdena tolkas med viss försiktighet.

6.5 Fortsatt arbete

Fortsatt arbete bör omfatta fler testscenarier, fler deltagare och mer varierad driftdata. De kvarstående utelämnandena i RAG-uppläggen motiverar vidare studier av viktning av källdata, exempelvis genom *reliability-aware* RAG. En naturlig vidareutveckling vore att undersöka om viktning av källornas allvarlighetsgrad i retrievalsteget kan minska antalet utelämnanden ytterligare, exempelvis i kombination med kunskapsgrafer i linje med Chen et al. [3]. Kunskapsgrafer identifieras som en möjlig vidareutveckling för att strukturera övervakningsdata semantiskt, men bedömdes ligga utanför arbetets avgränsning givet den strukturerade karaktären på den data som hämtades. Framtida studier bör även undersöka brusig eller ofullständig telemetri, formell säkerhetsutvärdering i fullskalig drift samt generalisering till andra molnplattformar än Microsoft Azure.

Källförteckning

- [1] S. Tatineni, "AIOps in Cloud-native DevOps: IT Operations Management with Artificial Intelligence," *Journal of Artificial Intelligence & Cloud Computing*, vol. 2, no. 1, pp. 1–7, 2023. [https://doi.org/10.47363/JAICC/2023\(2\)154](https://doi.org/10.47363/JAICC/2023(2)154)
- [2] L. Zhang, T. Jia, M. Jia, Y. Wu, A. Liu, Y. Yang, Z. Wu, X. Hu, P. S. Yu och Y. Li, "A Survey of AIOps in the Era of Large Language Models," *ACM Computing Surveys*, vol. 58, no. 2, art. 44, pp. 1–35, 2026. <https://doi.org/10.1145/3746635>
- [3] Z. Chen, Y. Qiu, L. Yang, B. Liao och D. Cao, "Automatic generation of monitoring report based on large language model and knowledge graph inference," *Results in Engineering*, vol. 26, art. 104795, 2025. <https://doi.org/10.1016/j.rineng.2025.104795>
- [4] Z. Lin, S. Guan, W. Zhang, H. Zhang, Y. Li och H. Zhang, "Towards trustworthy LLMs: a review on debiasing and dehallucinating in large language models," *Artificial Intelligence Review*, vol. 57, art. 243, 2024. <https://doi.org/10.1007/s10462-024-10896-y>
- [5] J. Hwang, J. Park, H. Park, D. Kim, S. Park och J. Ok, "Retrieval-Augmented Generation with Estimation of Source Reliability," in *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, Suzhou, China, 2025, pp. 34279–34303. <https://doi.org/10.18653/v1/2025.emnlp-main.1738>
- [6] Z. Ji, N. Lee, R. Frieske, T. Yu, D. Su, Y. Xu et al., "Survey of Hallucination in Natural Language Generation," *ACM Computing Surveys*, vol. 55, no. 12, pp. 1–38, 2023. <https://doi.org/10.1145/3571730>
- [7] L. Huang, W. Yu, W. Ma, W. Zhong, Z. Feng, H. Wang, Q. Chen, W. Peng, X. Feng, B. Qin och T. Liu, "A Survey on Hallucination in Large Language Models: Principles, Taxonomy, Challenges, and Open Questions," *ACM Transactions on Information Systems*, vol. 43, no. 2, art. 42, pp. 1–55, 2025. <https://doi.org/10.1145/3703155>
- [8] Cisco. What Is AIOps? Artificial Intelligence for IT Operations [Internet]. Cisco; [cited 2026 Mar 18]. Available from: <https://www.cisco.com/site/us/en/learn/topics/artificial-intelligence/what-is-aiops.html>
- [9] Microsoft. Azure Functions overview [Internet]. Microsoft Learn; 2026 [cited

2026 Mar 18]. Available from: <https://learn.microsoft.com/en-us/azure/azure-functions/functions-overview>

[10] Microsoft. What is Microsoft Entra? [Internet]. Microsoft Learn; 2025 [cited 2026 Mar 18]. Available from: <https://learn.microsoft.com/en-us/entra/fundamentals/what-is-entra>

[11] Microsoft. Microsoft Graph overview [Internet]. Microsoft Learn; 2025 [cited 2026 Mar 18]. Available from: <https://learn.microsoft.com/en-us/graph/overview>

[12] Microsoft. Architect multitenant solutions on Azure [Internet]. Microsoft Learn; 2025 [cited 2026 Mar 18]. Available from: <https://learn.microsoft.com/en-us/azure/architecture/guide/multitenant/overview>

[13] Microsoft. Retrieval augmented generation (RAG) and indexes [Internet]. Microsoft Learn; 2026 [cited 2026 Mar 18]. Available from: <https://learn.microsoft.com/en-us/azure/foundry/concepts/retrieval-augmented-generation>

[14] Microsoft. What is Azure Static Web Apps? [Internet]. Microsoft Learn; 2024 [cited 2026 Mar 18]. Available from: <https://learn.microsoft.com/en-us/azure/static-web-apps/overview>

[15] Microsoft. What is Azure Key Vault? [Internet]. Microsoft Learn; 2025 [cited 2026 Mar 18]. Available from: <https://learn.microsoft.com/en-us/azure/key-vault/general/basic-concepts>

[16] R. Likert, "A technique for the measurement of attitudes," *Archives of Psychology*, vol. 22, no. 140, pp. 1–55, 1932 [cited 2026 Mar 18]. Available from: https://legacy.voteview.com/pdf/Likert_1932.pdf

[17] Microsoft. Introduction to Table storage - Object storage in Azure [Internet]. Microsoft Learn; 2022 [cited 2026 Mar 18]. Available from: <https://learn.microsoft.com/en-us/azure/storage/tables/table-storage-overview>

[18] Microsoft. What is Azure Cache for Redis? [Internet]. Microsoft Learn; 2026 [cited 2026 Mar 18]. Available from: <https://learn.microsoft.com/en-us/azure/azure-cache-for-redis/cache-overview>

Bilaga A – Promptmallar för LLM-uppläggen

Följande promptmallar användes i experimentet. I RAG-baserade upplägg ersätts {data} vid körning med den JSON-strukturerade driftdata från rapportmodulerna.

Upplägg 1 – Zero-shot

Du är en dataanalytiker som sammanfattar rapportdata för icke-tekniska beslutsfattare.

REGLER:

- Skriv 3-5 punkter.
- Fokusera på kritiska avvikelser, sedan varningar, sedan informativa observationer.
- Använd konkreta siffror.
- Avsluta med en rekommendation.

Upplägg 2 – Few-shot

Du är en dataanalytiker som sammanfattar rapportdata för icke-tekniska beslutsfattare.

REGLER:

- Skriv 3-5 punkter.
- Fokusera på kritiska avvikelser, sedan varningar, sedan informativa observationer.
- Använd konkreta siffror.
- Avsluta med en rekommendation.

EXEMPEL på önskat format:

Givet data med 3 högriskincidenter och 12 olösta varningar:

- 3 säkerhetsincidenter klassificerade som hög allvarlighetsgrad kräver omedelbar utredning.
- 12 säkerhetsvarningar förblir olösta - 4 av dessa är medelallvarliga.
- Senaste incidenten rapporterades 2025-01-15 och är fortfarande aktiv.

- Rekommendation: Prioritera de 3 högriskincidenterna och eskalera olösta varningar till ansvarigt team.

Upplägg 3 – Standard RAG

Du är en dataanalytiker som sammanfattar rapportdata för icke-tekniska beslutsfattare.

REGLER:

- Utgå ENBART från den data som finns i kontexten nedan.
Hitta INTE på information.
- Om datan är tom eller saknas, säg det tydligt.
- Skriv kort och sakligt - max 3-5 punkter.
- Fokusera på nyckeltal, avvikelser och trender som syns i datan.
- Använd konkreta siffror från datan.
- Avsluta med en rekommendation baserad på den faktiska datan.

Data:

{data}

Upplägg 4 – Spårbar RAG

Du är en dataanalytiker som sammanfattar rapportdata för icke-tekniska beslutsfattare.

REGLER:

- Utgå ENBART från den data som finns i kontexten nedan.
Hitta INTE på information.
- Om datan är tom eller saknas, säg det tydligt.
- Skriv kort och sakligt - max 3-5 punkter.
- Fokusera på nyckeltal, avvikelser och trender som syns i datan.
- Använd konkreta siffror från datan.
- Avsluta med en rekommendation baserad på den faktiska datan.

KÄLLHÄNVISNING (VIKTIGT):

- Varje påstående MÅSTE avslutas med en källreferens i formatet [Källa: <modulnamn>, <fält>].
- Exempel: "3 incidenter har hög allvarlighetsgrad [Källa: Security Incidents, severity=high]."

- Om du inte kan koppla ett påstående till en specifik datapunkt, skriv INTE påståendet.

Data:
{data}

Upplägg 5 – Validerat upplägg (Självgranskning)

Du är en dataanalytiker som sammanfattar rapportdata för icke-tekniska beslutsfattare.

STEG 1 - SAMMANFATTNING:

- Utgå ENBART från den data som finns i kontexten nedan. Hitta INTE på information.
- Om datan är tom eller saknas, säg det tydligt.
- Skriv kort och sakligt - max 3-5 punkter.
- Fokusera på nyckeltal, avvikelser och trender som syns i datan.
- Använd konkreta siffror från datan.
- Avsluta med en rekommendation baserad på den faktiska datan.
- Varje påstående MÅSTE avslutas med en källreferens i formatet [Källa: <modulnamn>, <fält>].

STEG 2 - SJÄLVGRANSKNING:

När du skrivit sammanfattningen, gå igenom den punkt för punkt och kontrollera:

1. Finns varje siffra i källdatan? Om inte - ta bort påståendet.
2. Har du utelämnat någon kritisk observation (hög allvarlighetsgrad, aktiva incidenter, olösta varningar)? Om ja - lägg till den.
3. Drar du slutsatser som källdatan inte stödjer?
Om ja - omformulera eller ta bort.

Returnera ENBART den granskade och korrigerade versionen.

Data:
{data}

Bilaga B – Exempel på facit för testscenario S1

Följande facit användes som referensunderlag vid utvärderingen av testscenario S1. Facit togs fram manuellt utifrån den faktiska datan i den simulerade kundmiljön och beskriver de sju centrala observationer som en korrekt rapport förväntas återge. Varje observation är märkt med allvarlighetsnivå.

Tabell B.1: Facit för testscenario S1 – sju centrala observationer med allvarlighetsnivå.

Nr	Nivå	Observation
1	KRITISK	Sårbarhet CVE-2026-1237 (CVSS 10.0) identifierad på 15 exponerade maskiner. Ingen patch tillgänglig. Omedelbar åtgärd krävs.
2	KRITISK	Pågående MFA fatigue-attack mot fem användarkonton sedan 2026-03-12. Tre inloggningsförsök blockerade, ytterligare försök pågår.
3	KRITISK	Aktiv incident av typen <i>Possible data exfiltration</i> (incident-ID: INC-2026-0042, status: active, severity: high). Utredning ej påbörjad.
4	VARNING	Indikation om potentiell kryptomining på resursen VM-Dev-03. Ovanlig CPU-förbrukning (94%) registrerad sedan 2026-03-10.
5	VARNING	Fyra högallvarliga säkerhetsvarningar med status <i>new</i> inväntar åtgärd (Security Alerts, severity=high, status=new).
6	VARNING	Pågående tjänsteincident för SharePoint Online (incident-ID: SHO-2026-0089, status: active). Påverkar dokumentdelning för 120 användare.
7	INFO	Secure Score för prenumerationen är 47 av 100, en minskning med 8 poäng sedan föregående månad, drivet av olösta säkerhetsrekommendationer.

Bilaga C – Evaluatorprompt

Följande promptmall användes av evaluator-LLM:n (GPT-4.1-mini, temperaturvärde o.o) för att bedöma varje genererad rapport mot det manuellt framtagna facit. Variabeln {rapport} ersätts vid körning med den genererade rapporten och {facit} ersätts med referensunderlaget för respektive testscenario.

Du är en objektiv utvärderare av AI-genererade rapporter. Din uppgift är att bedöma en rapport mot ett referensunderlag (facit) enligt fyra kriterier. Basera ENBART din bedömning på det som faktiskt står i rapporten och facit.

RAPPORT ATT UTVÄRDERA:
{rapport}

FACIT (referensunderlag):
{facit}

UTVÄRDERINGSKRITERIER:

1. TEKNISKA SAKFEL (factuality errors)

Räkna antalet påståenden i rapporten som direkt motsäger ett verifierbart värde i facit, exempelvis felaktigt resursnamn, felaktigt antal incidenter eller felaktig tidsstämpel. Markera ett sakfel ENDAST när rapporten anger ett konkret värde som avviker från facit. Vaga eller ofullständiga formuleringar räknas inte.

Antal sakfel: [siffra]

Motivering: [förklara varje identifierat fel]

2. UTELÄMNANDE OBSERVATIONER (omissions)

Gå igenom facitets observationer en i taget. Kontrollera om motsvarande information återfinns i rapporten, oavsett formulering.

Antal utelämnanden: [siffra]

Motivering: [lista vilka observationer som saknas]

3. MISSVISANDE SLUTSATSER (faithfulness errors)

Räkna antalet fall där rapporten drar en slutsats som inte stöds av eller direkt motsäger den logiska innebörden i facit.

Ett faithfulness error föreligger när slutsatsen som helhet är tekniskt orimlig givet underlaget. Skilj detta från faktafel: ett faktafel gäller ett specifikt värde, ett faithfulness error gäller en slutsats.

Antal missvisande slutsatser: [siffra]

Motivering: [förklara varje identifierad missvisande slutsats]

4. SPÅRBARHETSGRAD (citation accuracy)

Kontrollera varje källhänvisning i rapporten mot facit.

Markera hänvisningen som korrekt om den pekar på en verklig datapunkt i underlaget med rätt värde, eller som inkorrekt om datapunkten inte finns eller värdet är fel.

Om rapporten saknar källhänvisningar, ange N/A.

Antal korrekta hänvisningar: [siffra]

Antal inkorrekta hänvisningar: [siffra]

Spårbarhetsgrad (%): $[\text{korrekta} / \text{totalt} * 100]$

Motivering: [förklara eventuella inkorrekta hänvisningar]

Bilaga D – Genererad rapport och evaluatörbedömning

Följande exempel illustrerar hur en genererad rapport från zero-shot-upplägget (upplägg 1) såg ut för testscenariot S1, samt hur evaluatör-LLM:n bedömde den mot facit. Exemplet är representativt för en av de tre körningarna och återspeglar de typiska felen som identifierades för detta upplägg.

Eftersom zero-shot-upplägget inte hade tillgång till faktisk driftdata representerar rapporten modellens genererade antaganden utifrån instruktionen, inte en källförankrad sammanfattning av testscenariot. Syftet med exemplet är därför att visa vilka typer av fel som kan uppstå när modellen genererar en rapport utan tillgång till det underlag som senare används som facit.

Genererad rapport – Zero-shot, körning 1

Säkerhetsläget i kundmiljön är överlag stabilt, men ett antal varningar kräver uppmärksamhet. Fem säkerhetsincidenter av hög allvarlighetsgrad är registrerade, varav de flesta är under utredning. Tjänstehälsan för Microsoft 365 uppvisar inga större avvikelser. Fyra säkerhetsvarningar av hög allvarlighetsgrad inväntar åtgärd. Secure Score för prenumerationen uppgår till 52 av 100.

Rekommendation: Prioritera utredning av de aktiva högriskincidenterna och åtgärda olösta säkerhetsvarningar.

Facit – Testscenariot S1

Se bilaga B för det fullständiga facit med alla sju observationer.

Evaluatörens bedömning

1. Tekniska sakfel (*factuality errors*): 4

- Rapporten anger att tjänstehälsan för Microsoft 365 inte uppvisar några större avvikelser. Facit visar ett pågående incidentärende för SharePoint Online (observation 6). [*Sakfel*]
- Rapporten anger att de flesta incidenter är under utredning. Facit visar att incident INC-2026-0042 av typen *Possible data exfiltration* har status `active` och

att utredning ej påbörjats. *[Sakfel]*

- Rapporten anger fem säkerhetsincidenter av hög allvarlighetsgrad. Facit visar däremot inte fem sådana incidenter, utan en aktiv incident av typen *Possible data exfiltration* samt andra kritiska observationer. *[Sakfel]*
- Rapporten anger Secure Score till 52 av 100. Facit visar 47 av 100 (observation 7). *[Sakfel]*

2. Utelämnade observationer (*omissions*): 5

- Pågående MFA fatigue-attack mot fem användarkonton (observation 2) saknas helt.
- Sårbarhet CVE-2026-1237 (CVSS 10.0, 15 exponerade maskiner) (observation 1) saknas helt.
- Indikation om potentiell kryptomining på VM-Dev-03 (observation 4) saknas helt.
- Detaljer om att incident INC-2026-0042 är av typen *Possible data exfiltration* (observation 3) är utelämnade.
- Observationen om högallvarliga säkerhetsvarningar återges endast delvis. Rapporten nämner fyra säkerhetsvarningar av hög allvarlighetsgrad, men utelämnar status *new* och kopplingen till observationens fullständiga källkontext (observation 5).

3. Missvisande slutsatser (*faithfulness errors*): 2

- Rapporten beskriver säkerhetsläget som “överlag stabilt”. Facit innehåller tre KRITISK-klassade observationer, däribland en aktiv incident av typen *Possible data exfiltration* där utredning ej påbörjats. Slutsatsen är tekniskt orimlig givet underlaget. *[Faithfulness error]*
- Rapporten anger att de flesta incidenter är “under utredning”, vilket ger intrycket att situationen hanteras aktivt. Facit visar att incident INC-2026-0042 har status *active* och att utredning ej påbörjats. *[Faithfulness error]*

4. Spårbarhetsgrad (*citation accuracy*): N/A

Zero-shot-upplägget inkluderade inga källhänvisningar. Spårbarhetsgrad mättes därför inte för detta upplägg.

TRITA-CBH-GRU-2026:134
Stockholm, Sweden 2026

www.kth.se