

AIOps: Pilvipalveluiden monitorointi tekoälyn avulla

Tiivistelmä

Tekijä(t) Heinonen, Jyrki	Julkaisun laji Opinnäytetyö, YAMK	Valmistumisaika 2021
	Sivumäärä 46	
Työn nimi AIOps: Pilvipalveluiden monitorointi tekoälyn avulla		
Tutkinto Digitaaliset ratkaisut, Insinööri (ylempi AMK)		
Tiivistelmä <p>Digitalisaation edetessä erilaisten monitoroitavien palveluiden määrä on kasvanut merkittävästi. Tämä on asettanut aivan uudenlaisia vaatimuksia niin monitoroinnille kuin operoinnillekin. Aiemmin tämä kaikki oli myös teknistä, IT-johtoista tekemistä, mutta nykyisin vastuuta ja päätöksen tekoa haluttaisiin siirtää tälläkin osa-alueella enemmän liiketoiminnan osaksi. Monitoimittaja- ja monipilviympäristöt asettavat omat haasteensa myös monitoroinnin käyttöönoton helppoudelle ja skaalautuvuudelle.</p> <p>Tämän opinnäytetyön aiheena oli tutkia pilvipalveluiden monitorointia tekoälyn avulla, sen tuomia mahdollisuuksia ja haasteita sekä siihen liittyviä tuotteita. Opinnäytetyössä käytiin dokumenttianalyysin keinoin lävitse olemassa olevaa kirjallisuutta aiheen ympäriltä sekä tutustuttiin empiirisesti työkaluihin. Osana opinnäytetyötä tehtiin myös puolistrukturoituja haastatteluja henkilöille, jotka ovat olleet käyttöönottomassa tai aktiivisesti käyttävät Dynatrace-tuotetta.</p>		
Asiasanat aiops, itops, devops, dynatrace, pilvipalvelu, monitorointi		

Abstract

Author(s) Heinonen, Jyrki	Type of Publication Master's thesis	Published 2021
	Number of Pages 46	
Title of Publication AIOps: Cloud monitoring with artificial intelligence		
Name of Degree Master of Engineering, Digital Solutions		
Abstract <p>As digitalization goes forward, number of applications that need monitoring, is growing significantly. This has posed completely new requirements for monitoring and operations. Earlier this was always a responsibility of IT-department but now responsibility and decision making is shifting towards business. Multivendor and multicloud environments pose also a new challenge for the ease of deployment and scalability of monitoring tools.</p> <p>This thesis was about studying cloud monitoring with artificial intelligence, its advantages, and disadvantages and also the products that enable these methods. Document analysis was used in this thesis to analyse existing literature and also empirically studying the tools. Part of the thesis was also semi structured interviews that were made for people who are part of the deployment team or the team actively using the Dynatrace-tool.</p>		
Keywords aiops, itops, devops, Dynatrace, cloud service, monitoring		

Sisällys

1	Johdanto.....	1
1.1	Työn rajausta ja tutkimuskysymykset.....	1
1.2	Tutkimusmenetelmät	1
2	Eri OPS-metodit ja niiden tavoitteet	3
2.1	ITOps.....	3
2.2	CloudOps	3
2.3	DevOps	3
2.4	BizDevOps	4
2.5	DevSecOps	5
2.6	MLOps.....	5
2.7	NoOps	6
3	AIOpsin synty	7
3.1	Tavoite.....	7
3.2	Määrittely.....	7
3.2.1	Observe.....	9
3.2.2	Engage.....	9
3.2.3	Act.....	10
3.3	Osa-alueet.....	10
3.3.1	Datamäärät suuria ja monimuotoisia.....	10
3.3.2	Infrastruktuurin ja sovellusten lukumäärä ja skaalautuvuus.....	11
3.3.3	Perinteisten prosessien kankeus	12
3.3.4	Seuranta ja toiminta.....	13
3.3.5	Tunnistaminen ja ennustaminen	13
3.3.6	Juurisyysanalyysi ja ehdotukset	14
3.3.7	Automaattinen reagointi.....	15
3.3.8	Reaktiivisesta proaktiiviseen.....	16
3.4	Käyttötapaukset.....	16
3.5	AIOpsin tulevaisuus	17
4	Business case: Dynatrace	18
4.1	Tavoite.....	18
4.2	Arkkitehtuuri.....	18
4.3	Tekoälymoottori Davis	20
4.4	Ominaisuudet	20
4.4.1	Topologiakartta ja palveluiden riippuvuudet	20

4.4.2	Virheiden etsintä ja hälytykset	21
4.4.3	Juurisyyanalyysi	22
4.4.4	Palveluiden eheyttäminen.....	23
4.4.5	Davis virtuaaliassistentti	25
5	Haastattelu ja sisällönanalyysi	26
5.1	Haastattelu tutkimusmenetelmänä	26
5.2	Haastattelun tulokset	26
5.2.1	Mitä ongelmaa lähdettiin ratkomaan	26
5.2.2	Mitä palveluita lähdettiin monitoroimaan	28
5.2.3	Oliko jokin tuote jo aiemmin käytössä?	28
5.2.4	Miten tuote valikoitui?	29
5.2.5	Mitä tuotteen osia on käytössä aktiivisesti?	30
5.2.6	Hyödynnättekö tietoja mittaristoissa kuten SLA, SLO tai SLI?	31
5.2.7	Oliko alkuvaiheessa kovia tavoitteita vai uteliaisuus mahdollisuuksista?.....	31
5.2.8	Saavutettiinkö tai löytyikö hyötyjä?	32
5.2.9	Mikä on vaatinut eniten oppimista tuotteesta?.....	33
5.2.10	Mitä asioita on löytynyt, jota ei manuaalisesti olisi löytynyt.....	34
5.2.11	Sovellusten tai palveluiden automaattinen palauttaminen	34
5.2.12	Miten on auttanut tuotekehityksessä tai ongelmien selvittelyssä?	36
5.2.13	Integraatiot tuotekehitykseen? Onko mukana DevOps-putkessa?	36
5.2.14	Tukitikettien (jos semmoisia oli aiemmin) määrän muutos?	37
5.2.15	Palautumisajan tai virheenselvityksen vaatiman ajan muutos?	37
5.2.16	Mitä mielestäsi AIOps tarkoittaa?.....	38
5.2.17	Muita havaintoja	39
5.3	Havaintoja haastattelututkimuksesta.....	40
6	Johtopäätökset ja pohdinta	42
	Lähteet	44

1 Johdanto

Yritysten digitalisaatio on ollut vuosia jo kovassa vauhdissa ja Forbesin arvion (Forbes 2018) mukaan jo 83 % enterprise-tason työkuormista ajettiin pilvipalveluissa vuonna 2020. Tämä aiheutti ja tulee aiheuttamaan vielä suuren muutospaineen yritysten monitorointimahdollisuuksissa sekä käytettävissä tuotteissa.

Tekoäly on myös viime vuosina mennyt suurilla harppauksilla eteenpäin. Pilvipalveluiden operointi tekoälyn avulla arvioitiin 900 - 1500 miljoonan euron markkinaksi vuonna 2020 (Gartner 2021).

Tämän opinnäytetyön tarkoituksena on käsitellä laajasti IT-järjestelmien monitorointia ja operointia tekoälyn avulla, jota kutsutaan termillä AIOps eli Artificial Intelligence for IT Operations. Monitoroinnista on tehty tutkimuksia aiemminkin, mutta AIOps laajentaa tätä tutkimuskenttää merkittävästi ja tuo analysointiin syvyyttä sekä uusia ominaisuuksia.

AIOpsin yhtenä kulmakivenä on tehdä monitoroinnista henkilökohtaista. Se osaltaan tarkoittaa teknisille ihmisille yhä tarkempia raportteja oikeassa kontekstissa mutta myös liiketoiminnan ihmisille sopivia raportteja ja mittareita, joissa käytetty kieli ja sisältö tuovat hyötyjä päätöksentekoon sekä siirtävät tiettyä vastuuta pois IT-osastolta.

1.1 Työn rajaus ja tutkimuskysymykset

Tämä opinnäytetyö on rajattu käsittelemään AIOpsia menetelmänä, sen eroavuutena aikaisempiin operointimalleihin sekä tuotteistukseen. Opinnäytetyön tarkoituksena ei ole syvällisesti esitellä Dynatrace-tuotetta tai muita tuotteita, vaikka viittauksia siihen paljon löytyykin.

Tutkimuskysymykset ovat seuraavia:

- Mitä AIOps on?
- Miten AIOpsia hyödynnetään IT-järjestelmien virnehallinnassa ja monitoroinnissa?

Työssä oli tarkoitus tehdä myös tekninen toteutus eli POC, mutta se jouduttiin lisäämään aikataulupaineiden takia ulosrajattuihin asioihin osoittauduttuaan liian laajaksi tehtäväksi.

1.2 Tutkimusmenetelmät

Tässä opinnäytetyössä tutkimusmenetelmänä käytettiin dokumenttianalyysiä sekä puolistrukturoitua haastattelua. Tarkasteltua työkalua lähestyttiin empiirisellä tutkimusotteella testausympäristön sallimissa rajoissa.

Dokumenttianalyysin (Kokonat 2021) tuottamat havainnot jäivät vähäisen kirjallisuuden takia ohuiksi suurimman osan ollessa Gartnerin tuottamia visioita tulevaisuudesta sekä blogikirjoituksia näiden raporttien pohjalta. Lisänäkökulmat jäivät tämän takia vähäiseksi ja suurin arvo saatiin empiirisellä tutkimuksella sekä haastattelun avulla.

Puolistrukturoidun haastattelun ajatuksena on esittää haastateltaville samat tai samanmuotoiset kysymykset (Saaranen-Kauppinen & Puusniekka 2006). Tässä haastattelussa kysymysten järjestyksellä ei ollut merkitystä, koska ne eivät suoranaisesti johtaneet keskustelua aiheesta syvemmälle, vaan yleensä vaihtoivat keskusteltavan asian toiseksi. Haastattelulle oli mietitty etukäteen 16 kysymyksen kysymyspatteristo, jota käytiin haastateltavan kanssa soveltuvin osin. Haastattelussa ei oltu muodostettu mielipidettä puolesta tai vastaan, vaan haastateltu sai muodostaa vastauksena oman kokemuspohjansa mukaiseksi. Muodollisempi haastattelu, esimerkiksi lomakehaastattelu, ei olisi tähän sopinut aiheen laajuuden sekä vastausten monimuotoisuuden vuoksi. Harvoin kysymyksiin olisi pystynyt vastaamaan välttämättä edes yhdellä lauseella, jolloin puolistrukturoitu haastattelu koettiin kaikista eniten tietoa antavaksi, mutta kuitenkin säilyttäen jonkinlaisen rakenteen analysointia ja vertailua varten.

Empiirisellä tutkimusotteella (Koppa 2015) kokeiltiin yhtä valittua työkalua, Dynatracea, käytännössä. Työkalulla tarkasteltiin käytettävissä olevaa dataa ja työkalun ominaisuuksia sen ympärillä. Työkalusta oli mahdollista saada kaikkea hyötyä irti tässä testissä, koska testiympäristö oli hyvin rajoittunut ja testiä varten rakennettu, mutta oikeaa ympäristöä simuloiva.

Työn alussa oli tarkoitus tehdä myös tekninen esimerkkitoteutus, Proof-Of-Concept, mutta se osoittautui tutkimuskentän selvityksessä työmäärältään liian laajaksi osa-alueeksi ja toteuttaminen mahdottomaksi. Teknisessä toteutuksessa yksinkertaisimmatkin osa-alueet olisivat oman opinnäytetyönsä arvoisia toteutuksia. Tekninen esimerkkitoteutus päädyttiin korvaamaan aiemmin esitellyillä puolistrukturoiduilla haastatteluilla.

2 Eri OPS-metodit ja niiden tavoitteet

2.1 ITOps

ITOps tai IT Operointi vapaasti suomennettuna sisältää perinteisesti kaikki IT-verkkojen, -laitteistojen tai sovellusten asennuksiin ja ylläpitoon liittyvät työt. Usein myös ensimmäisen tason helpdesk-palvelut ja muut erilaiset käyttäjätukipalvelut sopivat tämän termin alle.

Vanhanaikaisesti ITOps-osasto toimii byrokraattisen kankeasti, koska heidän vastuullaan on loppupeleissä yrityksen verkon toiminta liiketoiminnan mahdollistajana.

2.2 CloudOps

CloudOps eli pilvipalveluiden operointiin keskittyneet asiantuntijat. Usein tämän voidaan ajatella olevan sama kuin ITOps, mutta pilvipalvelut tarjoavat laajan skaalan erilaisia IaaS, PaaS tai SaaS palveluita, joita perinteinen ITOps ei ole tottunut käsittelemään eivätkä heidän projektinsa ole suunniteltu osiltaan täysin erilaisen toimintatavan päälle. Pilvipalveluissa on mahdollista erilainen spontaani innovaatio ja kokeiluluonteisuus, jota usein perinteisillä virtuaalikoneilla tai muilla ITOpsin tarjoamilla välineillä ei pystytä saavuttamaan. Tämä kokeiluluonteisuus ja kehittämisen nopeus tuo tullessaan kuitenkin suuria ratkaistavia haasteita, kuten tietoturva, kustannustenhallinta sekä sopiva määrä hallinnasta luopumista. Näihin kaikkiin on osittainen ratkaisu hyvin tehdyssä teknisessä hallintamallissa.

CloudOps-asiantuntijat joutuvatkin jatkuvasti työskentelemään vaativassa IT-operoinnin hallittavuuden, tietoturvan sekä liiketoiminnan ja ohjelmistokehittäjätiimien vaatimusten riskitilassa.

2.3 DevOps

DevOps tulee sanoista "Development" ja "Operations". DevOps on hyvin laaja termi ja sillä ei olekaan virallista tarkkaa määrettä. Useimmiten se kuitenkin kuvataan laajasti kokoelmaksi erilaisia käytäntöjä, joilla saadaan vähennettyä kehityksen ja tuotantoon viennin välistä aikaa, mutta silti huolehtien korkeasta laadusta (Loukides 2012).

Monet DevOpsin peruseriaatteista tulevat Lean-metodista sekä muista ketteristä menetelmistä. Niihin kuuluvat esimerkiksi seuraavat asiat:

- lyhyempi TTM (Time-To-Market) eli ominaisuuden tuotantoon vientiin tarvittu aika
- nopeampi kehityssykli
- uusien virheiden vähenevä määrä

- nopeampi tahti löytää virheet ja saada ne korjattua
- nopeampi tahti palata edelliseen versioon
- näiden kaikkien asioiden reaaliaikainen mitattavuus (tekniset KPI-mittarit).

Tärkeä vaatimus tätä varten oli ns. sillojen poistaminen yrityksistä. Perinteisesti IT-osasto ja kehittäjätiimit toimivat organisaatiossa erillään toisistaan ja näiden osastojen tavoitteet eivät välttämättä olleet sidottuja toisiinsa. Tällöin kehittäjätiimien oli usein mahdotonta tai ainakin vaikeaa saavuttaa omia tavoitteitansa, koska IT-osastolla ei ollut samoja liiketoiminnan intressejä saatikka komentohierarkiaa kehitystiimin kanssa. Kehittäjätiimin nopeatahtinen innovointi ja tarve erilaisille kokeiluluonteisille palveluille saattavat olla IT-osaston toiminnan kannalta hankalaa ja uusia prosesseja vaativaa. Kehittäjätiimeissä myöskin usein IT-osaston omasta toiminnasta huolehtiminen vaikuttaa enemmän kiusanteolta tai vastahakoisuudelta muuttua.

IT-osaston lisäksi on monia muita organisaation tai lakipykälien tuomia haasteita, jotka täytyy ratkaista todellista DevOpsin käyttöönottoa varten. Usein erilaiset perinteiset vaatimukset määrittelyille, eri asteiselle testaamiselle sekä tuotantoympäristöjen päivityksille saattavat asettaa sen kaltaisia rajoitteita, ettei DevOpsin tavoitteita voida saavuttaa ilman muutoksia.

Näiden asioiden takia DevOps on käytännön tekemistä laajempi termi, joka voi tarkoittaa koko yrityskulttuurin muutosta ja sen hyväksymistä.

2.4 BizDevOps

Liiketoiminnan suurempaa mukaantuloa kehitysprosessiin mukaan haluttiin tarkentaa termissä BizDevOps (Business Development Operations). DevOpsin yleistyessä sillä usein tarkoitettiin vain perinteisestä vesiputousmallista luopumista ja siirryttiin nopeaan määrittely-kehitys-testaus-tuotanto-sykliin. Tämän syklin nopeuttaminen on kuitenkin mahdotonta kehityksen liikkeelle panevaa ja sen hyväksyvää voimaa eli liiketoimintaa ja sen edustajia. BizDevOpsin idea on sekoittaa liiketoiminnan edustusta kiinteäksi osaksi kehitystiimiä ja sen päätöksentekoa. Usein liiketoiminta määrittelee tavoitteet tulevia syklejä eli sprinttejä varten sekä toimii niiden SME-roolissa (Subject Matter Expert eli kyseisen aiheen asiantuntija). Liiketoiminnan edustaja saattaa olla myös usein testaaja tai vähintään sprintin tuotos-ten hyväksyjän roolissa. (Gill 2020.)

Tärkeänä osana kehitysjonon, eli backlogin, muodostumista on liiketoiminnan palaute, tarpeet ja muutosehdotukset. Muutokset ja uudet ominaisuudet pitäisi myös pääosin olla mitattavissa, jolloin muutoksille saadaan tarvittaessa laskettua hyötysuhdelukuja esimerkiksi

kehityksen vaatiman ajan, eli kulujen, suhde tuotettuun aineelliseen tai aineettomaan hyötyyn.

Nykypäivänä liiketoimi toimii käytännössä aina enemmän tai vähemmän DevOps-tiimien yhteydessä, vaikka BIZ-etuliitettä ei erityisesti käytettäisi.

2.5 DevSecOps

Osana perinteistä IT-osastoa tai isommissa yrityksissä jopa omana osastonaan toimii tietoturva. Sovelluskehityksessä tehdään usein uusia hyökkäyspintoja ulko verkkoon API-raja-pintoina tai nettisivuina. Useista eri syistä tietoturvanäkökulmaa ei aina tule huomioitua parhaalla mahdollisella tavalla ja lisäksi tietoturvan ylläpitäminen sovelluksissa ja niiden tarvitsemisissä palveluissa on aivan oma asiantuntijuutensa. Perinteisesti sovelluksiin ja palveluihin on pyydetty auditointeja joko suunnitteluvaiheessa tai juuri ennen tuotantoon siirtoa. Ennakoiva tietoturva-auditointi tehdään usein täysin erilaisien (tai puuttuvien) määrittelyjen avulla kuin valmis sovellus tulee olemaan. Usein loppuvaiheessa tehtävä auditointi taas aiheuttaa suuren paineen liiketoiminnan kannalta yleensä hyväksynnän kannalta. (RedHat 2018.)

Varsinkin ennakkotapauksien puutteessa (sovittu malli toteuttaa sovelluksia) usein DevOps tiimiin sekoitetaan suhteellisen kiinteäksi osaksi tietoturva-asiantuntija. Hänen vastuullaan saattaa olla joko ajoittainen tarkistus muutoksista ja niiden tietoturvasta tai jopa sovittujen osioiden toteuttaminen tietoturvallisesti. Tällöin varsinainen auditointi on jatkuvaa eikä suurempia yllätyksiä projektin loppuvaiheilla pitäisi tulla.

2.6 MLOps

MLOps eli Machine Learning Operations on kokoelma käytäntöjä, jotka yhdistävä koneoppimisen, DevOpsin sekä datainsinöörin (data engineering) osa-alueet ja tähtäävät julkaisemaan ja ylläpitämään koneoppimisen järjestelmiä tuotannossa luotettavasti ja tehokkaasti. (Breuel 2020.)

MLOps tiimiin kuuluu yleensä henkilöitä kolmesta eri roolista, kuten datatieteilijä, DevOps-insinööri sekä datainsinööri. Koneoppimisen mallien luonti ja tuotantoon vienti tehokkaasti vaatii näiden eri roolien aktiivista yhteistyötä, jotta eri osa-alueet toimisivat oikein, tehokkaasti sekä järkevästi suhteutettuna ajoalustaan sekä ajoympäristöön ja näiden tietoturva- ja muihin vaatimuksiin.

2.7 NoOps

Perinteisen IT-Operoinnin laajentuessa, kustannukset yleensä kasvavat lineaarisesti. Enemmän palveluita, enemmän ihmisiä. NoOps (No Operations) suhtautuu hallintaan kuten ohjelmistokehitysprojektiin. Operoinnin tarpeelle ja tehtäville, kuten palveluiden kaa-tumiset ja päivitykset, ei lähdetä tekemään ohjeistusta ja manuaalisesti ajettavia skriptejä, vaan ohjelmistotekniikan keinoin se yritetään joko poistaa kokonaan ratkaisemalla ongelma tai toteutetaan automaatio tekemään manuaalinen osuus. Tärkeimpänä on se, että jatku-vasti kasvavia sovellushallinnan ja ylläpidon kustannuksia ei hyväksytä, ellei sovellusten määrä kasva eksponentiaalisesti

NoOps edellyttää luonnollisesti korkeaa automaatioastetta, lokituksien ja monitorointien kuntoon laittamista, sovellusten suunnittelua self-healing periaatteella, automaattista infra-strukturin luontia ja päivitystä sekä tietenkin tarvittavien työkalujen luontia, ellei valmiita ole saatavilla.

Googlen määrittelemä SRE-toimintamalli (Site Reliability Engineer) tähtää NoOpsiin, täydelliseen automaatioon. SRE-asiantuntija toimii osana tuotekehitystiimejä ja auttaa heitä poistamaan teknistä velkaa toteutuksesta ja ohjeistaa tai muokkaa itse toteutusta niin, että NoOpsin vaatimukset täyttyvät ja manuaalisen operoinnin tarve poistuu tai ainakin vähenee huomattavasti. (Sloss 2017.)

3 AIOpsin synty

Yritysten digitalisaation yleistymisen ja kehitysvauhdin kiihtyminen aiheutti palveluiden tarvitseman infrastruktuurin ja sovellusten määrän nopean kasvun ja sen mukana palveluista kerätty data ja telemetriikatieto kasvoi räjähdysmäisesti (Forbes 2018). Ihmisen kyvyt tutkia ja reagoida tietoihin ja hälytyksiin heikkenee (alarm fatigue) sekä tarve ihmisresursseille kasvaa huomattavasti. Usein haasteena oli myös se, että perinteiset mittaristot olivat staatisia eivätkä mukautuneet muutokseen.

Lähtökohtaisesti kaikki ongelmat luotiin tiketeiksi helpdesk-järjestelmiin ja ihmisten suorittama ongelmien selvittely ja palveluiden operointi ei pysynyt kasvavien vaatimusten perässä. Osaltaan DevOps-menetelmien käyttö oli yrittänyt vastata tähän sovelluskohtaisesti, mutta sovellusten laajentuessa tarvittiin useammin useiden ympäristöjen tutkimista ja mahdollisesti usean eri toimittajan yhteistyötä eri vastuualueilla.

3.1 Tavoite

Tutkimusyhtiö Gartner visioi vuonna 2016 termin ”Algorithmic IT Operations”, mutta se muutettiin myöhemmin muotoon ”Artificial Intelligence for IT Operations” eli AIOps. Termi luotiin kuvaamaan koneoppimisen sekä analytiikan lisäämistä osaksi operointia (Gartner 2016). Koneoppimisen hyödyntämistä kokeiltiin jo useissa isommissa yrityksissä eri tarkoituksiin ja sen toivottiin sopivan tärkeään rooliin käsittelemässä ja ryhmittelemässä tapahtumia, joiden pohjalta voitaisiin soveltaa erilaisia automaatiomenetelmiä. Tämä mahdollistaisi suuren nopeuden tapahtumiin reagoimisessa. Keräämällä tarpeeksi dataa voidaan verrata menneisyyttä ja nykytilannetta, joka mahdollistaa virheiden löytämisen nopeasti ja mahdollistaa tarvittavan mittariston rakentamisen esimerkiksi sovelluksen uusien ominaisuuksien aiheuttamien ongelmien toteamiseksi. Tällöin haasteisiin pystytään puuttumaan välittömästi ja tarvittaessa automaattisesti.

AIOps on yksi tärkeä osa NoOps-ympäristöjen saavuttamista, koska kaikkiin ongelmiin ja haasteisiin ei pystytä etukäteen valmistautumaan. Kasvava määrä loki- sekä telemetriatietoja mahdollistavat tilanteen analysoinnin ja koneoppiminen ennakoimisen tiettyjen tapahtumien tai analyysin tuloksien avulla.

3.2 Määrittely

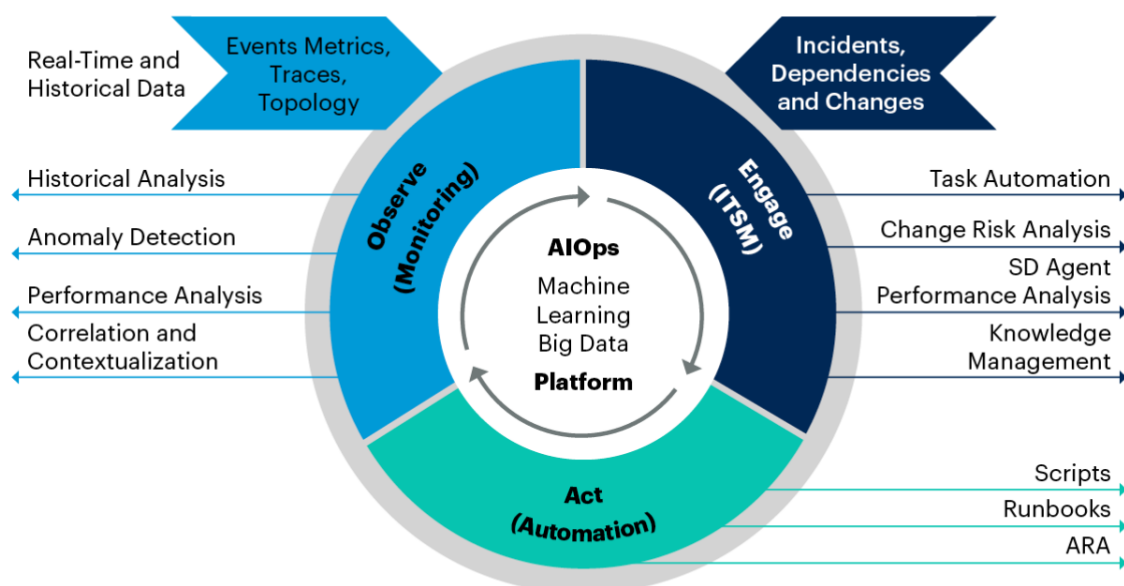
Gartner määritteli termin AIOps seuraavasti:

AIOps platforms utilize big data, modern machine learning and other advanced analytics technologies to directly and indirectly enhance IT operations (monitoring,

automation and service desk) functions with proactive, personal and dynamic insight. AIOps platforms enable the concurrent use of multiple data sources, data collection methods, analytical (real-time and deep) technologies, and presentation technologies. (Gartner 2016.)

Kaikki osa-alueet tässä määrittäksessä ovat ainakin jossain määrin olleet yleisessä käytössä jo aiemmin. Usein lopputuloksena oli kuitenkin erillisiä järjestelmiä omilla ominaisuuksillaan, useita eri sijainteja tiedolle ja siiloutunut näkemys ympäristöihin ja eri sovellusten tilanteeseen. AIOpsin tarkoitus on sitoa nämä asiat yhteen moderneilla teknologioilla ja palveluilla sekä tuottaa dynaamista ja oikeaa tietoa keskitetysti oikeiden henkilöiden saataville.

AIOps Platform Enabling Continuous Insights Across IT Operations Monitoring (ITOM)



Source: Gartner
735577_C

Gartner

Kuva 1. AIOpsin osa-alueet ITOM-kaaviossa (Gartner 2021)

Monitoroinnin perustana on aina ollut jatkuvasti kehittyvä silmukka, jossa jokainen havainto aiheuttaisi mieluiten automaattisen parannuksen kaikkiin silmukan osa-alueisiin. AIOpsin mukana silmukan keskiöön tuodaan koneoppimisen menetelmät sekä big datan tuomat mahdollisuudet (kuva 1).

3.2.1 Observe

Silmukan ensimmäisenä vaiheena pidetään observointia eli varsinaista monitorointia. Perinteisesti tämä on tarkoittanut staattisia sääntöjä, sovellusaluekohtaisia lokituksia ja niiden käsittelyä sekä tiettyjen metriikoiden seuraamista (proessorikuorma, muistinkulutus). Yhteistä näille kaikille on staattisuus ja kokemuspohjaisuus eli säädetään, kunnes arvo tuntuu olevan hyvä. Ympäristöjen nopea kehitys ja muuttuminen aiheuttavat sen, että tässä monitoroinnissa ei pysytä mukana kehityksen vauhdissa.

AIOpsin määritelmän mukaan tässä vaiheessa tietoa pitää pystyä sisäistämään useasta eri lähteestä ja eri formaateissa reaaliaikaisesti. Tietoa ei välttämättä aina ensin säilötä tietovarastoon ja prosessoida sieltä, vaan se prosessoidaan tiedon saapumishetkellä. Tässä vaiheessa tiedolle voidaan tehdä vertailua historiatietoihin, tunnistaa poikkeamia tai havaita haasteita suorituskyvyssä.

3.2.2 Engage

Silmukan toinen vaihe sitoo kaiken edellisessä vaiheessa ja historiassa kerätyn tiedon ja esittelee sen visuaalisesti ja konteksti riippuvaisesti. Tämä vaihe sitoo käsitellyn tiedon muihin järjestelmiin, kuten tiketöintijärjestelmiin, riski- ja muutoshallintajärjestelmiin, sisällönhallintajärjestelmiin (Knowledge Management) tai erilaisiin chatbotteihin (ns. ChatOps).

Kerätyt virheet ja niiden kategoriat esikäsitellään, ryhmitellään ja korreloidaan jotta virheiden määrää ja niiden aiheuttamaa tikettiväsymys-ilmiötä (alarm fatigue) saataisiin vähennettyä. Reaaliaikaisen ja historiatietoihin perustuvan kontekstin kerääminen ja esittäminen on erittäin tärkeä asia, joka antaa tiketin käsitelijälle tarvittavat tiedot asian hoitamiseen ilman suurta manuaalista työtä. Kontekstiin kerätään aiemmat samankaltaiset tilanteet, kaikki tilanteeseen liittyvät käyttäjät, rajapinnat, sovellukset, alustat ja järjestelmät. Juuri-syyanalyysillä voidaan suodattaa tilanteeseen liittyviä järjestelmiä, jotta huomio osattaisiin kohdistaa oikeaan paikkaan nopeasti. Usein on rakennettu myös integraatio erilaisiin viestintäkanaviin, kuten yrityksen sisäiseen pikaviestimeen tai chatbottiin. Tämä mahdollistaa nopeat tarkat kyselyt, kuten onko palvelu vielä alhaalla tai pyydetään raporttia sovituista SLA- (service level agreement), SLI- (service level indicator) tai SLO-mittareista (service level objective) (Atlassian 2021). Chatbot tarjoaa mahdollisuuden hyödyntää alustan kognitiivisia palveluita (puheen- ja tekstintunnistus tai kategorisointi) siihen, että asiaa tutkiva taho saa tarvitsemansa tiedot ilman, että ihmisen tarvitsee avata edes etäyhteyksiä palveluihin. Myös mahdollisesti palvelun korjaamisen toimenpiteet voitaisiin tehdä chatbotin avulla, jolloin ihminen voisi esimerkiksi valita AIOpsin tarjoamista korjausvaihtoehdoista sopivimman tai kertoa omansa.

Tähän vaiheeseen ei välttämättä aina tarvitse ihmisresursseja, vaan mahdollisen automaation ansiosta tämä voi olla joskus myös vain tiedottamiseen liittyvä vaihe, joka tiketöi, selvittää, korjaa ja raportoi kaiken automaattisesti.

3.2.3 Act

Silmukan viimeinen vaihe, toiminta eli toivottu automaatio, on mahdollisesti vaikein osio AIOps muutoksessa. Muiden vaiheiden jälkeen pystytään tietyllä tarkkuudella päättelemään, millä toimenpiteellä palvelu saataisiin palautettua käyttöön. Vaihe on perinteisesti sisältänyt erilaisten ennakolta kirjoitettujen skriptien ajamista tietyissä tilanteissa ja skriptit voivat esimerkiksi käynnistää palveluita uudelleen tai skaalata ja uudelleen provisoida korruptoituneita resursseja. ARA eli Application-Release-Automation tarkoittaa prosessia, jossa automaatio osaisi tarvittaessa asentaa sovelluksista uusia versioita tai palauttaa vanhoja takaisin käyttöön. Viimeisimpänä vaihtoehtona on erilaisten runbookkien eli tarkan prosessiohjeistuksen tarjoaminen ongelmaa käsittelevälle resurssille (=ihminen).

AIOpsin tavoitteena on luonnollisesti vähentää kiinteiden runbookkien hyödyntämistä ja tulevaisuudessa staattisten skriptien käyttöä. Juurisyiden avulla pystytään löytämään toimenpiteitä vaativat järjestelmät ja kohdistaa niihin joko perinteisiä staattisia skriptejä tai sitten maturiteetin kasvaessa automaattisia toimenpiteitä.

3.3 Osa-alueet

3.3.1 Datamäärät suuria ja monimuotoisia

Kerätyn datamäärän käsittely nousee tärkeän rooliin AIOpsissa. Kaikki havainnot ja toimenpiteet tehdään datan perusteella, joten sen kerääminen ja käsittely nopeasti ja tehokkaasti on ensimmäisiä ratkaistavia asioita.

Kerättävä data on usein hyvinkin monimuotoista. Sovellusten sisäinen lokitus, sovelluksen ja infrastruktuurin havaintojen ja suorituskyvyn telemetriikkatietojen tallennus, tiketointijärjestelmän tiketit sekä erilaisten IOT-laitteiden, API-rajapintojen, mobiilisovellusten sekä loppukäyttäjien tuottama data on hyvä esimerkki monimuotoisuudesta. Esimerkiksi taloustiedot tai vain yksittäisen lähteen tuottama data ovat hyviä esimerkkejä määrämuotoisesta datasta.

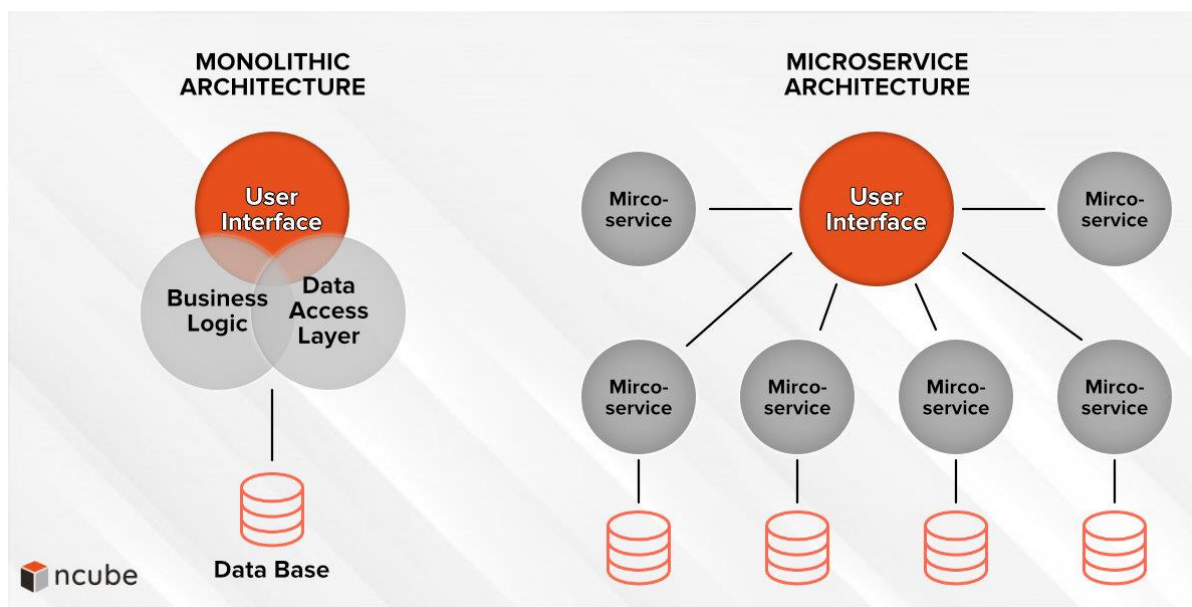
Määrämuotoista sisältöä on yleisesti säilytetty BI-työkaluille sopivassa Data Warehouse-järjestelmässä. DW-järjestelmät pohjautuvat usein skeemamuotoisiin tietokantoihin ja niiden pohjalta rakennettuihin kiinteisiin kyselyihin ja taulurakenteisiin. Monimuotoisuuden

vuoksi yleinen säilytyspaikka on erilaiset supertehokkaat levyjärjestelmät tai järjestelmät, joita kutsutaan datajärveksi (Data Lake).

Dataan pohjautuva ongelmien tunnistaminen ja ennakoiminen vaatii hallinnollisten ja teknisten datasäiliöiden sillojen purkamista sekä reaaliaikaisen ja arkistoidun datan yhdistämistä. Kerättävä data pitää pystyä rajaamaan ja suodattamaan hyvin jo keräysvaiheessa, mutta tietynlainen data voi vaatia tietojen anonymisointia, pseudonymisointia tai muuta siivousta ennen käsittelyä.

3.3.2 Infrastruktuurin ja sovellusten lukumäärä ja skaalautuvuus

Digitalisaation ja kehittämisen nopeutuessa sovellusten määrä ja niiden vaatima arkkitehtuuri moninkertaistuu. Perinteisten monoliittisten järjestelmien sijasta sovelluksia usein kehitetään mikropalveluarkkitehtuurilla tai pilkotaan muilla tavoin pienemmiksi komponenteiksi (kuva 2). Sovelluksen vaatima ympäristö on pikemminkin dynaamisesti muuttuva kuin staattinen tarkasti suunniteltu kokonaisuus. Tällöin sovellusten infrastruktuuri, logiikka ja lokitus on hajaantunut ympäri verkkoa. Sovelluksen tarvitsema prosessointitehoa on siirretty konesaleista lähemmäs verkon reunoja ja sen käyttäjiä.



Kuva 2. Monoliittiratkaisu vs. mikropalveluratkaisu (Demchenko 2020)

Osittain tämä muutos erilaiseen arkkitehtuuriin johtuu siitä, että liiketoimintapäätökset ja budjetointi johtavat ohjelmistokehitystä, jolloin yhtä keskitettyä hallinnointia ei enää ole. Pulonkaulaksi yritysten kehitymisessä on havaittu yksittäisten hyväksyvien henkilöiden käyttämisen eli niin sanottuja sovellusomistajia on muutaman sijasta reilusti enemmän liiketoiminnan hierarkian mukaisesti. Nykyaikaiset teknologiat, muutos hallinnallisessa

rakenteessa sekä uusien resurssien luomisen helppous ja halpuus ovat muuttaneet pelikenttää niin, että sovelluskehittäjillä on suurempi vastuu tämän osa-alueen toteuttamisesta, kuin aiemmin. Suurempi monitorointivastuu siirtyy IT-osastoilta sovellusten ylläpitosopimusten pariin, mutta yleinen toimintavarmuus on edelleen IT-osaston vastuulla. Tähän myös sekoittuu todella paljon seuraavan kappaleen ”Perinteisten prosessien kankeus” sisältö.

3.3.3 Perinteisten prosessien kankeus

Keskitetyn hallintamallin poistuessa sekä sovellusten lukumäärän kasvaessa perinteiset tikeöntijärjestelmät eivät skaalaudu riittävästi. Skaalautumisella tarkoitetaan tässä sitä, että käsiteltävä tietomäärä kasvaa merkittävästi ja suurin osa ei välttämättä vaadi ihmisen puuttumista asiaan ollenkaan. Manuaalinen työ (tikettien käsittely tai vaadittujen muutosten teko eivät riitä dynaamisessa muuttuvassa ympäristössä. Suuren muutosmäärän takia ympäristöjen seuranta ja hallinta manuaalisin keinoin muuttuu mahdottomaksi. IT-operointi on jo vuosien ajan muuttunut monimutkaisemmaksi ja tilanne ei ole paranemaan päin. Kasvanut tikettimäärä ja niiden käsittely luo luonnollisesti suuria operointikustannuksia, joka usein voi toimia motivaationa prosessien hallitulle yksinkertaistamiselle. Kuva 3 kertoo yksittäisen perustason service desk-tiketin käsittelyhinnoista Pohjois-Amerikassa vuonna 2016. Tikettien päätyessä ylemmän tason service desk -henkilön käsiteltäväksi, hinnat ovat luonnollisesti korkeammat.

Metric Type	Service Desk Cost Metrics	North American Statistics		
		Average	Min	Max
Cost	Cost per Ticket	\$15.56	\$2.93	\$49.69
	Cost per Minute of Handle Time	\$1.60	\$0.76	\$2.50

Kuva 3. Service desk -tikettien määrä (Rumburg 2017)

Usein sovellusten kehitys on niin nopeaa, että niihin ei voida sitoa projektiin aktiivisesti kuuluvien tai pienellä ajankäytön allokaatiolla olevien ihmisen tekemää työtä. Ohjelmistokehityspuolella manuaalioyötä on korvattu erilaisilla automaattitestatekniikoilla, kuten yksikkötestaus, integraatiotestaus sekä regressiotestaus. Näillä mittareilla voidaan jo päätellä koodin laadusta ja toiminnasta paljon, mutta iso osa tästä vastuusta tulee ohjelmistokehittäjien tehtäväksi ja projektiryhmän hallittavaksi ja seurattavaksi.

Infrastruktuuri ja palveluarkkitehtuuri pitäisi suunnitella niin, ettei jokainen pienikin muutos aiheuta byrokraattista prosessia yrityksen rakenteissa, vaan sen pitäisi olla sujuvaa, joustavaa ja hyvin sovittua. Usein kankeutta voidaan poistaa ennalta hyväksytyillä

arkkitehtuureilla ja toteutuksilla, jotka vaativat pienemmän katselmointiprosessin etene- mistä varten. DevOps-ajatuksen mukaisesti ohjelmistokehitys ja operointi pitää olla lähem- pänä toisiaan, jotta hidastuksia aiheuttava vastakkainasettelu saadaan poistettua.

3.3.4 Seuranta ja toiminta

Sovelluksien ja ympäristön ongelmat pitää havaita ja ratkaista jatkuvasti kiihtyvällä nopeu- della. Organisaatiot, joka tekevät täyden siirtymän digitaalisiin palveluihin luottavat isolta osalta niiden tuovan tehokkuutta ja kustannussäästöjä pitkässä juoksussa. Liiketoiminta- suunnitelma ja sovelluksen loppukäyttäjät määräävät usein eri sovellusten bisneskriittisyy- den ja ympäristön palveluvasteen. Käyttäjäkokenemusta pidetään niin suuressa arvossa, että pienikin katkos on merkittävä poikkeama.

Sovellusten ja alustojen seuranta pitää rakentaa tarpeita ajatellen ja pienissä vaiheissa. Tarpeet pitää huomioida kaikessa tekemisessä, jotta tekninen velka ei ala pikkuhiljaa kas- vamaan siihen, että sovelluksen monitorointi ei ole ajankohtaista eikä riittävää, ja se alkaa vaatimaan enemmän ja enemmän ihmisresurssien huomiota.

Seurantaa on usein rakennettu erilaisten virtuaalisten kojelautanäkymien tai tiettyjen sään- töjen pohjalta tulevien sääntöjen automaattisähköpostien tai -tikettien varaan, mutta kai- kessa tässä on vaarana IT-monitoroinnista vastaavien hälytysten aiheuttama väsymys eli alarm fatigue. Tärkeänä osana tässä on vain oikeiden mittarien ja tietojen näyttäminen sekä tietojen koostaminen ja ryhmittely. Järjestelmien määrien kasvaessa raportoinnin näky- missä tarpeettomat tiedot vievät huomion oikealta seurannalta.

3.3.5 Tunnistaminen ja ennustaminen

Keskitetyn lokitietojen keräämisen ja reaaliaikaisen metriikan ansiosta dataan voidaan koh- distaa erilaisia tekoälymenetelmiä, kuten koneoppimisen algoritmeja tai neuroverkkoja. Me- netelmät etsivät tietojoukosta poikkeavuuksia ja tärkeämpänä niihin johtavia asioita. Lokiri- vien määrän kasvaessa tuhansista riveistä miljooniin riveihin ihmisen on mahdotonta sil- mäilemällä löytää malleja ja rakenteita, jotka kuvastaisivat ongelmaa saatikka ongelmaan johtavia tapahtumia. Lähes reaaliaikainen tunnistaminen on yksi NoOpsin tai AIOpsin kes- keisistä asioista.

Tapahtumien ennakointi on ollut puuttuva osuus monitoroinneista ennen tekoälyn hyödyn- tämisen yleistymistä. Ennakointia on pyritty tekemään määrittelemällä kiinteitä sääntöjä ja ehtoja, joiden pohjalta on joko luotu tikettejä tai ajettu määriteltyjä toimenpiteitä. Näiden sääntöjen löytäminen ongelmakuvausten, lokien ja verkkovianetsinnän keskeltä on ollut massiivinen työ ja vaatinut huomattavan paljon aikaa. Tekoälyn avulla saataisiin

tulevaisuudessa generoitua vähintään ehdotuksia, joiden pohjalta IT-operoijat ja sovelluskehittäjät voisivat tutkia ja hyväksyä tai hylätä ehdotukset, ellei niitä pystytä nopealla aikataululla korjaamaan.

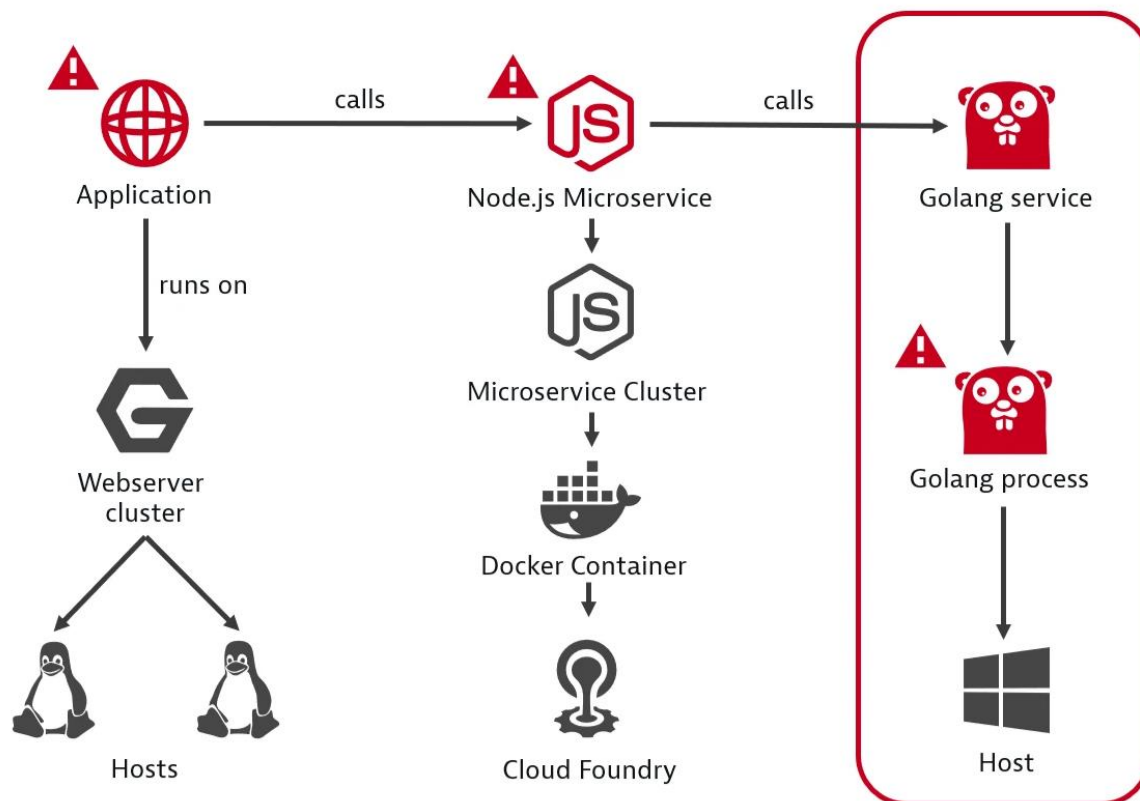
3.3.6 Juurisyyanalyysi ja ehdotukset

Ongelman tunnistamisen jälkeen asiasta pitää muodostaa akuutin ratkaisun lisäksi juurisyyanalyysi tai arvaus sekä ehdotus nopealle ja pysyväälle korjaukselle. Tämä hoidetaan perinteisesti tiketöintijärjestelmäintegraatiolla, jossa automaattisesti luotuun tikettiin reagoidaan palvelusopimuksessa olevalla vasteajalla. Perinteisessä ITOM-mallissa (IT Operations Management) kerättyihin lokitietoihin, kuten palvelinten tai palveluiden terveydentila, niiden metriikat sekä sovellusten virhelokit yhdistetään tulevaisuudessa ITOM-malliin kuulumattomiin liiketoimintamittareihin, kuten esimerkiksi mainoskampanjan tulosprosenttiin tai rekisteröitymisten/verkkokauppa-ostosten lukumäärään.

Tämä mahdollistaa ehdotukseen täydellisemmän kontekstiedon kasaamista kuin aiemmin. Kontekstiedolla tarkoitetaan ongelman syntymisajankohdan tarkempaa tilannetta ja sitä edeltäneitä asioita. Tällöin myös mahdollinen juurisyyanalyysi syntyy reaaliaikaisesti samalla ehdotuksen tekemisen kanssa.

Perinteisesti ongelmien ratkaisuun ja juurisyyanalyysiin liittyy useampi eri yrityksen osasto tai ulkopuolinen toimija. Näissä monimutkaisissa palaverissa (niin sanottu IT war room) kukin toimija on vastaamassa omasta vastuualueestaan, ja yleensä syytely lähtee liikkeelle näkyvästä asiasta piilossa oleviin eli verkkosivuista tai sovelluksesta päätyen yleensä verkkoinfrastruktuuriin.

Automaattisesti tehtyyn juurisyyanalyysiin sisältää paljon tietoa sovelluksista, integraatioista sekä verkon rakenteesta. Analyysin tekevän osapuolen pitää osata kaivaa kaikki ongelmaan liittyvät rajapinnat, sovellukset, sovelluksien ajoalustat ja niissä olevat prosessit, miten nämä on verkotettu keskenään sekä mitkä palvelut ovat riippuvaisia toisistaan. Yleensä virheet oireilevat visuaalisissa komponenteissa kuten verkkosivut (tietojen haku tai lisäys), vaikka ongelma on jossain paljon syvemmillä verkon rakenteissa.



Kuva 4. Juurisyysanalyysi (Beer 2019)

Kuva 4 osoittaa, miten ongelma ilmenee käyttäjille verkkosivulla, mutta oikeasti ongelma onkin taustajärjestelmien riippuvuuksien prosesseissa. Tämän tietomäärän käsitteleminen vie ihmisen askel kerrallaan lähemmäs totuutta, mutta automaattinen käsittely mahdollistaa kaikkien liittyvien järjestelmien tiedon keräämisen, käsittelyn tekoälyn avulla, tietojen ryhmittelyyn sekä todella tärkeän kontekstin lisäksi mahdollisesti jopa oman ehdotuksensa ongelmallisesti ympäristöstä. Jos työkalu pystyy tarjoamaan kontekstia ongelman ympärille (tapahtumahetket, liittyvät järjestelmät ja niiden lokit tai ongelmalliset kohdat koodissa), niin se on jo todella tärkeää ja hyödyllistä tietoa ongelmaa selvittävälle asiantuntijalle.

3.3.7 Automaattinen reagointi

Automaattinen reagointi asioihin, joihin voidaan reagoida, on tulevaisuudessa tärkeämpää kuin koskaan. Ihmisen toteuttama työ tulisi olla aina asiantuntijatyötä ja yksinkertaisemmat tilanteet pitäisi saada automatisoitua. Palveluiden uudelleenkäynnistys tai jopa uusien palveluiden provisiointi pitäisi tapahtua automaattisesti, kun huomataan katkoksia tai korruptoitumista, jolloin ainoastaan suuremmat verkkovirheet (rikkovat topologiamuutokset, puuttuvat reititykset) tai sovelluksen toimintalogiikan virheet vaativat manuaalista työtä niiden korjaamiseksi.

Pilvipalveluista alkaa löytymään jo ominaisuuksia, jotka osaavat resurssien käynnistämisen tai tarvittaessa uudelleenluonnin tietyissä tilanteissa (vasteaika nousee, palvelu ei vastaa), mutta suurin työ on saattaa sovellusten logitus vaaditulle tasolle, jolloin näihin tilanteisiin johtavat merkinnät voidaan tunnistaa ja automatisoida ellei niitä saada nopeasti korjattua. Eri tapahtumien tunnistaminen ja niistä mallien etsiminen on automaattisen reagoinnin kannalta erityisen tärkeää.

Tätä kuvaamaan käytetään mittaria nimeltään MTTR (Mean Time To Resolve) eli ongelman ratkaisuun vaadittu aika. Ratkaisu tarkoittaa tässä itse akuutin ongelman korjaamisen lisäksi myös tarvittavaa dokumentointia, lisätiestien rakentamista tai itse ongelman korjaamista niin, että sitä ei voi enää samanlaisena tapahtua.

3.3.8 Reaktiivisesta proaktiiviseen

AIOpsin tarkoituksena on siirtää IT-operointi ja muut siihen liittyvät asiat pois reaktiivisesta toiminnasta proaktiiviseen eli ennustavaan toimintaan. Se vaatii kaikkien tässä mainittujen palasten paikallaan oloa ja niiden välistä yhteistyötä. Seuraavassa kappaleessa on listattu muutamia esimerkkejä AIOpsin avulla tehtävistä reaktiivisista toimenpiteistä.

Kaikkia asioita ei voi tietysti ennustaa, jolloin jää jäljelle reaktiivisen työn helpottaminen antamalla riittävän selkeitä ja oikean kontekstin omaavia virheilmoituksia sekä niihin liittyviä dokumentaatioita.

3.4 Käyttötapaukset

Tässä kappaleessa on kuvattu muutama tosielämän käyttötapaus, jossa AIOpsin hyödyntämisestä olisi ollut mahdollisesti hyötyä.

- Verkkokauppa kokee toimialastaan riippuen sesonkimuutoksia, esimerkiksi talviurheilutuotteita myydään eniten talven alussa tai loppupalven alennusmyynneissä. Eri-laiset kampanjat ja alennusmyynnit pitäisi myös näkyä verkkokaupan metriikoissa. Näistä metriikoista voidaan päätellä palvelun toimivuuden lisäksi muiden liiketoimintamittareiden tavoittamista. Palvelua voidaan skaalata esimerkiksi tehokkaammaksi etukäteen ennen suurimman käyttäjäkuormituksen saavuttamista tai voidaan tarvittaessa lähettää hälytys mahdollisesta ongelmasta odotetun käyttäjäkuormituksen puuttuessa.
- Sovellustoimittajan tai pilvipalveluntarjoajan palvelun suunniteltu tai suunnittelematon käyttökatkojen status-tietoihin reagointi provisioimalla varakapasiteettia eri maantieteelliseen sijaintiin. Ongelman ilmetessä ulkopuolisten tietolähteiden

lisääminen juurisyysselvityksen kontekstiin, jotta eri palveluiden status-tiedot ovat heti käytettävissä asiantuntijalla.

- Sovelluksen vasteajan hidas ja huomaamaton nousu ilman sitä selittävää syytä, kuten esimerkiksi uusien ominaisuuksien julkaisu tai vanhojen muutokset. Syitä hidastumiselle voisivat olla tällöin muistivuoto, palvelualustan ylikuormitus tai muut palveluun liittyvät integraatiot.

3.5 AIOpsin tulevaisuus

AIOpsin eri osa-alueet ovat olleet yritysten hyödynnettävissä jo kauan, mutta käyttökohteet kehittyvät ja laajenevat koko ajan. Tuotteidenkaan maturiteettia ei ole vielä saavutettu, koska tietyillä osa-alueilla ollaan vielä niin lapsenkengissä. Varsinkin aiemmin tunnistamattomien ongelmatilanteiden automaattinen käsittely kehittyy jatkuvasti, mutta monitoroitavien sovellusten monimuotoisuuden vuoksi yleispätevien ratkaisujen keksiminen ei ole yksinkertaista. Toisaalta myös samoista syistä toimialueriippuvaisia toteutuksia on hankala tuottaa korkeiden aloituskustannuksien ja mahdollisesti hitaan takaisinmaksuajan takia. Saavutetut hyödyt voivat ottaa aikansa ja sen takia kannattaakin aloittaa pienistä muutoksista ja työstää hallitusti kohti täydellisempää AIOps-toteutusta.

Kuten monessa muussakin automaatioissa, loppukäyttäjillä tai pääkäyttäjillä voi olla erittäin pieni luotto lopputulemaan tai ratkaisuihin. Lainsäädäntö saattaa määritellä jo joitain asioita, jotka estävät tiettyjen automaatioiden käyttämisen, mutta tekniset monimutkaisuudet ja niiden hallinta halutaan valitettavan usein pitää ihmisen hallinnassa. Tällöin saatetaan olla valmiita ottamaan mieluummin korkeita kustannuksia, kuin pientä, mutta yleensä hallittua, riskiä toiminnassa.

4 Business case: Dynatrace

Pilvipalveluoperoinnin ja kehittämisen tuomia haasteita varten tarvitaan ratkaisuja ja yleensä ne on jo hyvin pitkälle tuotteistettu ja saatavilla SaaS-palveluna. On-premises-ympäristössä näiden ratkaisujen ajaminen voi olla kustannustehotonta suuren vaaditun kapasiteetin sekä suorituskyvyn takia. Myös jos suurin osa sovelluksista on migroitu tai rakennettu pilvinatiivisti, tulee suuria kustannuksia ja viiveitä lokitusdatan siirtämisestä sisäverkon palveluihin.

Yksi näistä tuotteista on suureen suosioon noussut Dynatrace, jota tässä kappaleessa tullaan tarkastelemaan. (Tack 2021.)

4.1 Tavoite

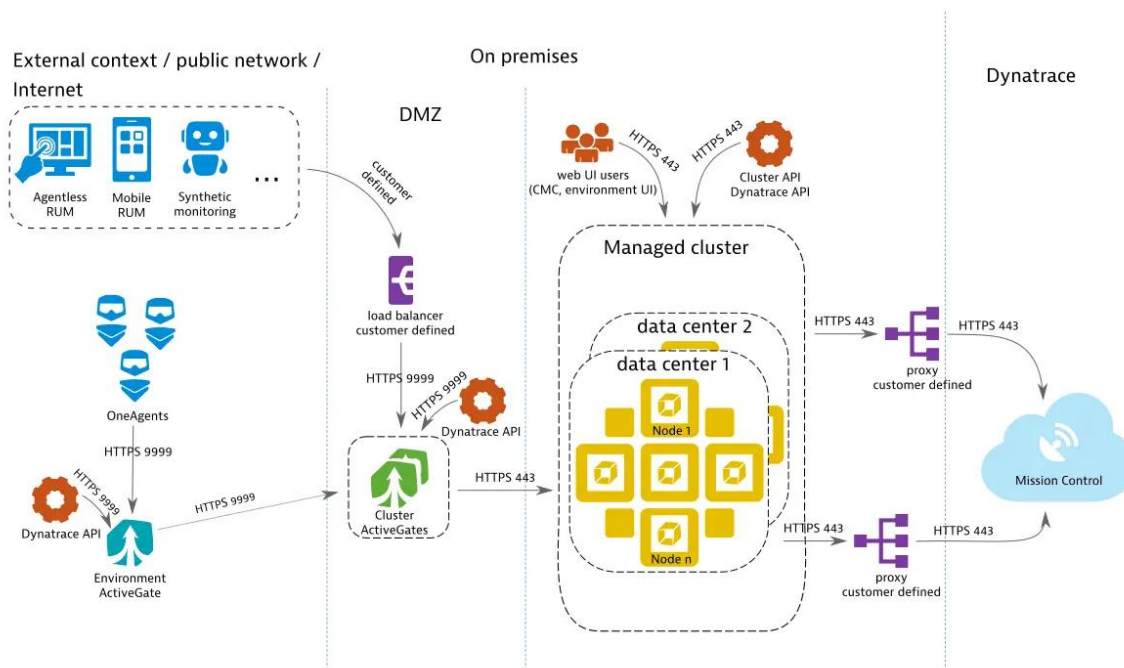
Tämän business casen tavoitteena on selvittää tuotteen tärkeimmät ominaisuudet ja miten ne vastaavat NoOpsin & AIOpsin tuomiin tarpeisiin. Ylimääräisten tuotteiden käyttöönotto tuo aina kustannuksia sekä ylläpidollista painetta. Tärkeänä osana minkä tahansa tuotteen evaluointia on sen hyötyjen ja haittojen listaaminen, mutta sen lisäksi selvittää mahdollinen kyvykkyys palvelun tarjoamiseen asiakkaalle sekä sen tehokkaaseen hyödyntämiseen.

Kyvykkyydellä voidaan tarkoittaa sitä, että onko yrityksen järkevää ottaa haltuun uutta tuotetta osaksi tarjoamaansa vai onko tarjoamassa jo jokin tuote, joka ajaa samoja asioita. Kyvykkyys voi tarkoittaa myös sitä, että jos palvelua asiakkaalle suositteleva yrityksen osa keskittyy lähinnä pilvinatiiviin tuotekehitykseen, onko heidän viisasta ottaa verkkoinfrastruktuuria monitoroivaa tuotetta osaksi heidän ydinosaamistaan.

4.2 Arkkitehtuuri

Dynatrace-palvelu on saatavissa pilvipalveluna tai on-premises asennuksena. Pilvipalvelun hyötynä ovat oman ympäristön operoinnista säästyvät kulut sekä palvelun vaatimien resurssien kustannukset, jotka voivat olla merkittävät riippuen ympäristön suuruudesta ja tarpeista. Muuten palvelut ovat samansisältöiset, on ne omassa hallinnassa tai pilvipalveluna ostettuna. Tärkein syy on-premises asennuksen, eli Dynatrace Managedin, valinnalle on datan säilyttäminen halutussa lokaatiossa (yrityksen oma datacenter tai halutun regionan pilvipalvelu). Pilvipalvelussa datat sijoittuvat käytettävästä regionasta riippuen eri puolille maailmaa. Dynatrace Managed palvelua käytettäessä, asiakas vastaa itse palvelualueen toiminnan ja Dynatrace toimittaa vain päivityksiä, lisenssejä tai käsittelee tukipyyntöjä.

Palvelu voidaan rakentaa arkkitehtuurisesti usealla tavalla ja riippuen vaaditusta tietoturvan määrästä, voi rakentua esimerkiksi seuraavasti (kuva 5):



Kuva 5. Dynatrace Managed -arkkitehtuuri (Dynatrace 2021c)

Palvelussa on useita eri komponentteja, joilla kaikilla on oma vastuualueensa. Taulukossa 1 on esitelty on-premises asennuksen komponentit ja niiden käyttötarkoitukset.

Komponentti	Tarkoitus
Mission Control	Dynatracen hallinnassa oleva pilvipalvelu, jonka portaali tarjoaa teknistä tukea ja lisenssien hallinnointia. Palvelu on myös vastuussa asiakkaiden Dynatrace Managed instanssien päivittämisestä tarvittaessa. (Dynatrace 2021g.)
Dynatrace Managed	On-premise datakeskukseen asennettu palvelu, joka hoitaa varsinaisen tietojen säilymisen ja sen käsittelyn. Kaikki toiminnallisuus ja logiikka on tässä palvelussa. (Dynatrace 2021f.)
ActiveGate	ActiveGate toimii sisäverkon ja ulko-verkon välisenä proxyinä sekä sisäverkon välitysjä testausagenttina. OneAgenttien verkkoliikenne voidaan ohjata kulkemaan näiden kautta, jotta sisäverkon reititys toimii. Testausagentti osaa ajaa scriptattuja tai muuten konfiguroituja http- tai selainpohjaisia testejä sisäverkon palveluihin. Pystyy laajennusosilla lukemaan eri palveluiden tarjoamia rajapintoja, joihin ei voida aina asentaa OneAgenttia. (Dynatrace 2021e.)
OneAgent	Agenttisovellus, joka asennetaan kaikkiin monitoroitaviin ympäristöihin ja palveluihin, mukaan lukien fyysiset- ja virtuaaliset palvelimet sekä SaaS-palvelut (kuten Azure Functions tai App Services). Kerää metriikat ja logidatat sekä lähettää ne Dynatrace-palvelimen käsiteltäväksi. (Dynatrace.2021h.)

Taulukko 1. Dynatrace on-premises komponentit

Monitoroinnin toimivuuden takia jokaisessa palvelussa ja palvelimessa pitää olla OneAgent asennettuna tai jollain toisella tavalla sallittu Dynatrace-palvelun suorittama monitorointi. Ilman tätä tiedonkeruuta, ei AIOpsia pystyisi hyödyntämään.

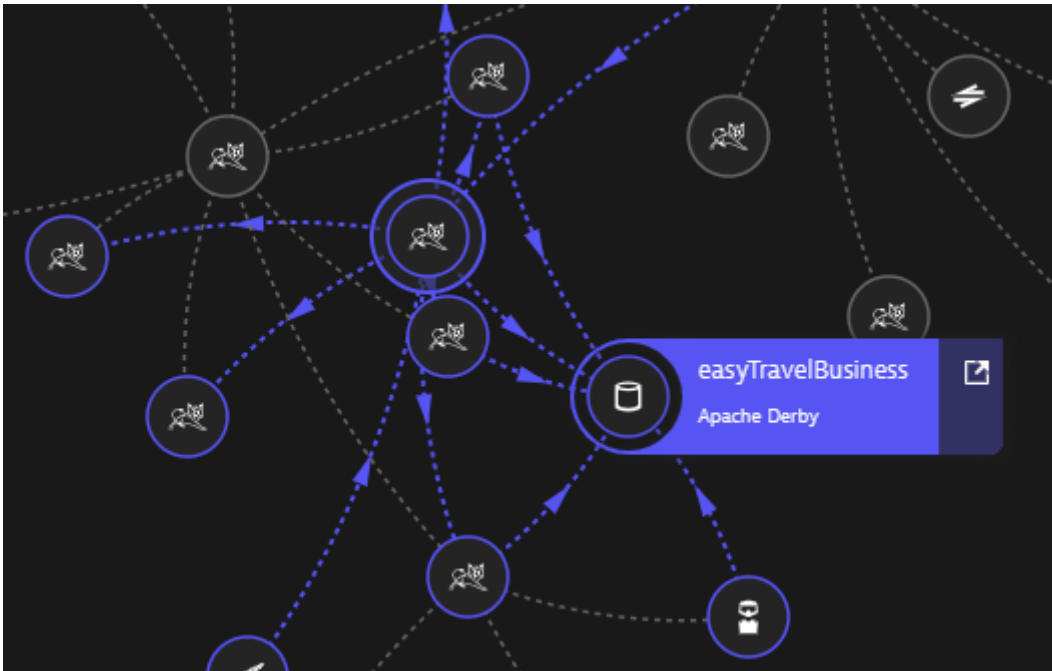
4.3 Tekoälymoottori Davis

Dynatrace-palvelun ytimessä on tekoälymoottori nimeltään Davis. Davis on seuraavan sukupolven tekoälykausaalimoottori, joka julkistettiin vuonna 2019. Davisin käytössä on kaikki ympäristöstä kerätty metriikka, lokidata, verkkotopologiat sekä palvelussa mahdollisesti konfiguroidut sovelluskokonaisuudet ja niiden asetukset. Kausaliteetti tarkoittaa tapahtumien välisiä syy-seuraus-suhteita ja niitä Davis yrittää etsiä. Davis tarjoaakin siis kaikille osa-alueille tarvittavan tiedon, joka näytetään asiantuntijalle. Davis on alunperinkin suunniteltu skaalautuvaksi ja moderniksi palveluksi, joka käsittelee nykyisin Dynatrace-pilvipalvelussa 368 biljoonaa riippuvuutta sekunnissa ja yli kahden biljoonaa topologiamuutosta päivittäin. (Dynatrace 2021d.)

4.4 Ominaisuudet

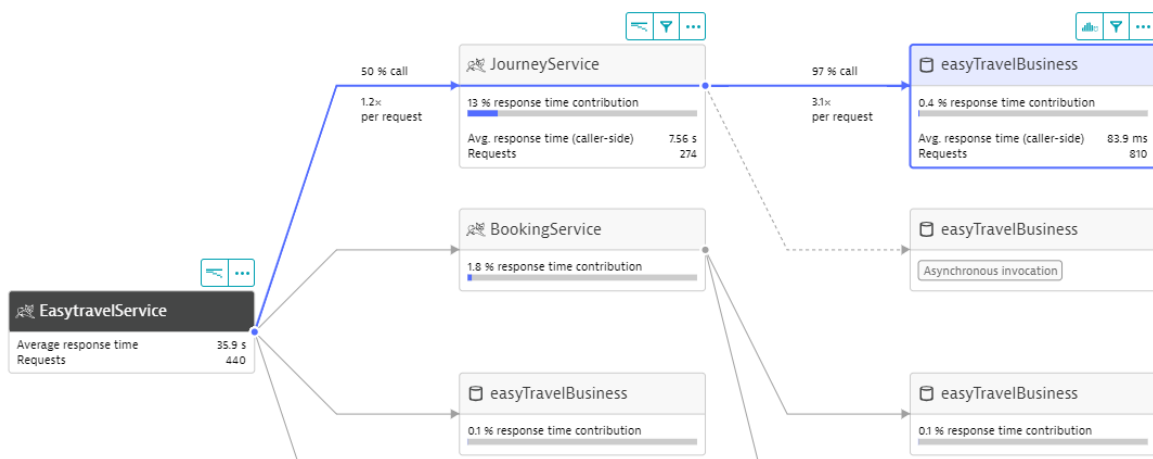
4.4.1 Topologiakartta ja palveluiden riippuvuudet

Dynatrace-palvelu rakentaa Smartscape-topologiakarttaa kaikista monitoroitavista alustoista, sovelluksista ja prosesseista. Topologiakartan avulla pystytään tunnistamaan toisistaan riippuvaiset palvelut, jota tarvitaan oikeiden ongelmien löytämisessä ja juurisyyanalyysissä.



Kuva 6. Smartscape-topologiakartta

Smartscape-topologiakartta antaa hyvän ylätasoin näkymän siihen, mitä palveluita monitoroinnin parissa on sekä miten ne ovat riippuvaisia toisistaan (kuva 6). Tarkemman näkymän riippuvuuksista näkee avaamalla palvelun service flow-näkymän (kuva 7).



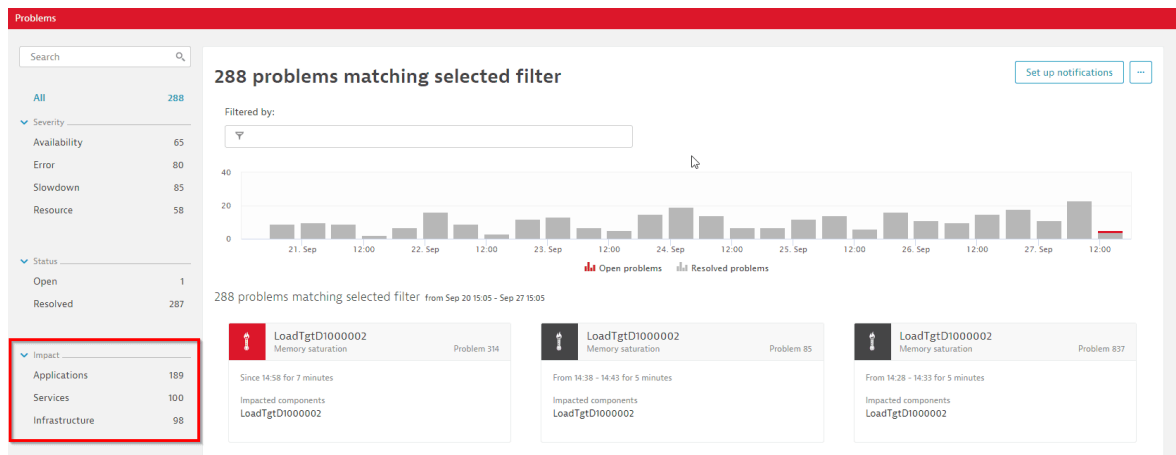
Kuva 7. Service Flow -näkökulma

Service flow -näkökulmassa näkyy kaikki tiettyyn palveluun liittyvät muut palvelut tai alustat. Näkökulma antaa tietoa vasteajoista sekä siitä, kuinka paljon kyseistä riippuvuutta on kutsuttu.

4.4.2 Virheiden etsintä ja hälytykset

Virheet Dynatracessa ovat palveluiden poikkeustilanteita eli esimerkiksi hidas vasteaika tai koodissa tapahtunut virhe (kuva 8). Nostetut virheet sisältävät kaikki virheeseen ja sen

aiheuttaneisiin tapahtumiin liittyvät tiedot. Suorituskykyasioissa on erilaisia raja-arvoja. Osa raja-arvoista ovat automaattisia (automated baselines) ja osa staattisia (Built-in static threshold). Automaattiset raja-arvot vaativat pienen määrän dataa oletusarvojen määrittelyä varten, mutta staattiset raja-arvot ovat heti voimassa. Staattisia raja-arvoja ei suositella yleisesti käytettäväksi, koska ne on tarkoitettu vain selkeiden metriikoiden tarkkailuun, kuten muistin kulutus tai prosessorin kuormitus prosentteina. (Dynatrace 2021b.)



Kuva 8. Ongelmalistaus (Dynatrace 2021i)

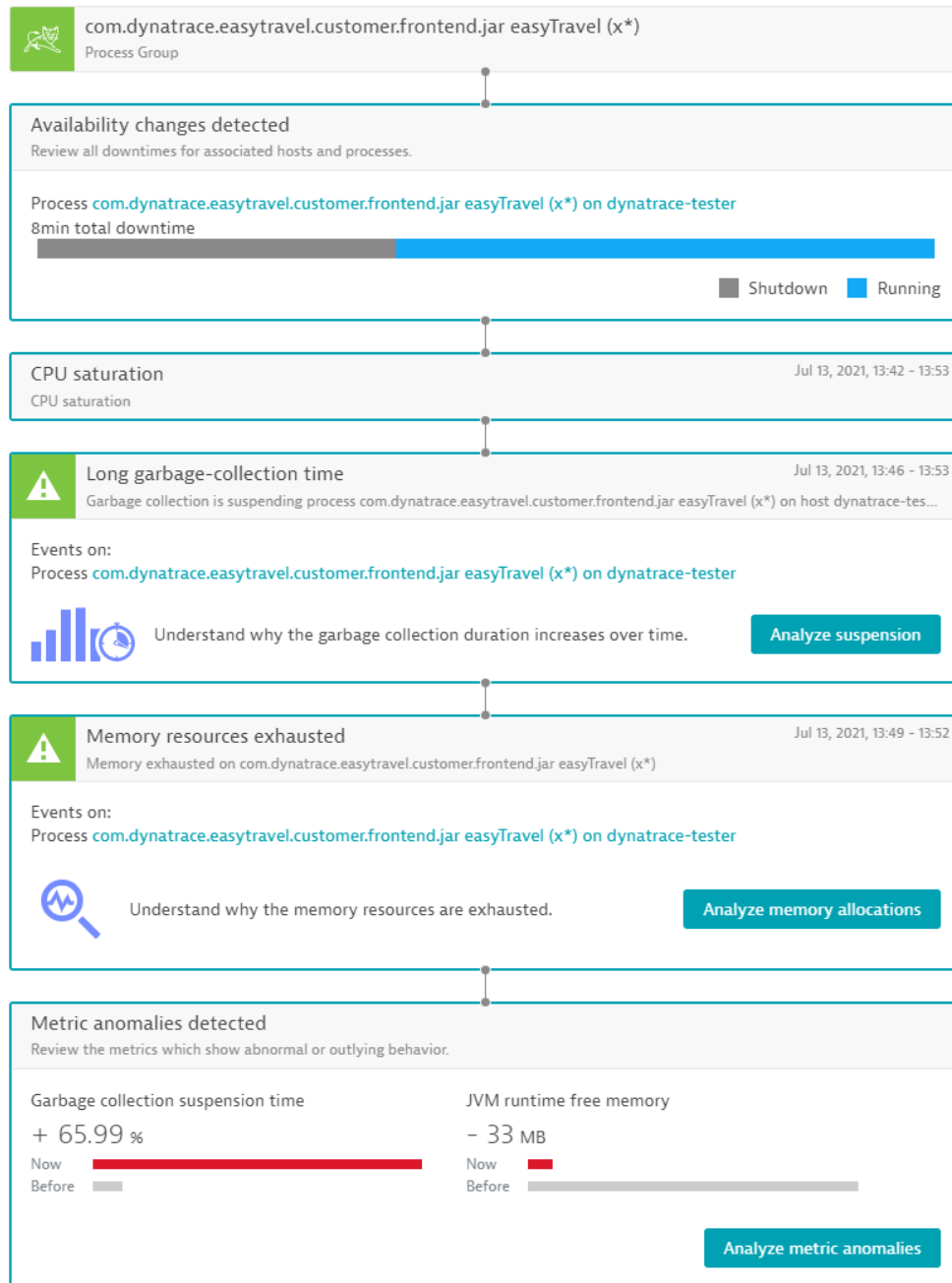
Yksittäinen ongelma kertoo tarkempia yksityiskohtia siitä, missä päin koodilogiikassa tai infrastruktuurissa se sijaitsee. Ongelman yksityiskohdissa näytetään myös liiketoiminnan mitareita, kuten kuinka moneen käyttäjäaktiiviteettiin ongelma on vaikuttanut negatiivisesti. Ongelman tunnistamisen jälkeen pystytään muodostamaan juurisyyanalyysi.

4.4.3 Juurisyyanalyysi

Kun ongelmia on havaittu ja tietoa niistä kerätty, voidaan muodostaa juurisyyanalyysi. Dynatrace koostaa Davisin avulla mahdollisesti useita juurisyyitä tapahtuneelle (kuva 9). Dynatracen analyysistä tulee yleensä hyvin selville ongelmalliset prosessiryhmät, ilmenneet ongelmat ja niihin johtaneet syyt sekä löydettyihin tietoihin pääsee porautumaan syvemälle (kuva 9).

Root cause

Based on our dependency analysis all incidents have the same root cause



Kuva 9. Juurisyysanalyysi

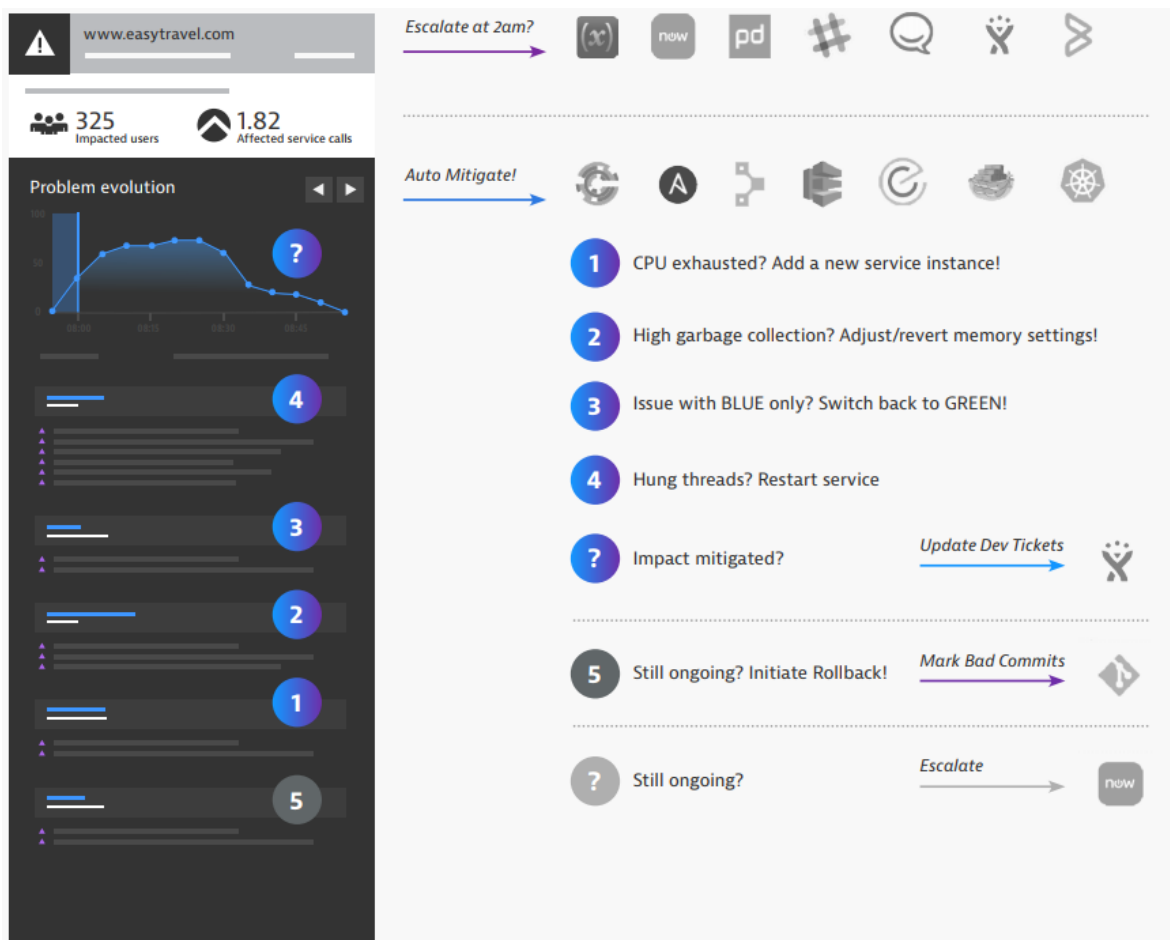
Analyysin pohjalta löytyi nopeasti tieto prosessorikuorman noususta, joka johti roskanke-
rääjän ongelmaan ja sitä kautta muistin täyttymiseen.

4.4.4 Palveluiden eheyttäminen

Virheiden havainnoinnin ja juurisyysanalyysin jälkeen tulee vaihe, jossa palvelut pitäisi saada ehytettyä käytettäväksi. Useissa pilvipalveluissa onkin mahdollisuus jo käynnistää korruptoituneita instansseja tai käynnistää uusia instansseja, jos aiempien kuormitus nousee

liikaa. Näissä on haasteena kuitenkin hallinnan hajautuminen eli sovelluskohtainen näkyvyys toimenpiteille. Tällöin jokaisella sovelluksella voi olla oma tapansa hoitaa eheytystoimenpiteet.

Dynatrace-palvelussa palveluiden eheyttäminen on usean alueen yhteistyötä, jota Dynatrace-palvelu orkestroii. Palveluiden provisioinnit pitää olla tehty Infrastructure-as-a-code-tekniikalla eli skriptattu tai määritelty eri työkaluin ja Dynatracella pitää olla jokin keino kutsua näitä työkaluja. Yleensä tämä tapahtuu CI/CD-työkalujen kautta, jotka huolehtivat sovellusten ja skriptien julkaisemisesta ja ajamisesta oikeita ympäristöjä vasten.



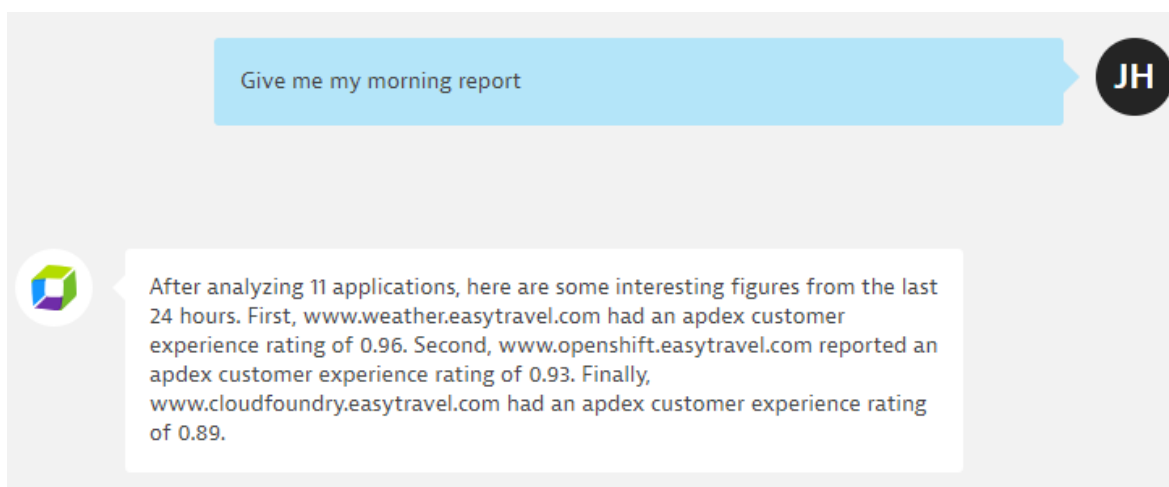
Kuva 10. Palveluiden eheytyminen (Dynatrace 2019, 15)

Eheyttämisessä on useita vaiheita ja päätöksiä (kuva 10). Priorisoinnilla pyritään tekemään ensimmäiseksi päätöksiä tilanteen eskaloimisesta esimerkiksi juhlapyyhinä tai öisin. Priorisoinnin mittarina voi toimia hyvin tilanteen vakavuus, puuttuvat automaattiot palvelun palauttamiseen sekä liiketoiminnan mittareita kuten verkkokaupan tuottavuuden muutokset tai kävijöiden tyytyväisyys. Automaattisia toimenpiteitä voidaan yhdistää tietynlaisiin metriikoihin tai lokimerkintöihin, kuten esimerkiksi prosessorin liiallinen kuormitus tai sovelluksien

roskienkerääjä-prosessin liian pitkä käsittelyjono. Toimenpiteet ajetaan Dynatrace-palvelun hälytyksien avulla ulkopuolisessa järjestelmässä.

4.4.5 Davis virtuaaliassistentti

ChatOps on nopeasti kasvava osa operointia. ChatOpsissa normaali operointi tai näkymä on korvattu kognitiivisilla chatboteilla. Dynatrace-palvelu tarjoaa Davis virtuaaliassistentin, joka ymmärtää kirjoitettua tai puhuttua englantia (kuva 11). Dynatrace-palvelun portaali ja mobiilisovellus ovat oletusrajapintoja virtuaaliassistentin kanssa kommunikointiin, mutta integraation pystyy tekemään myös Amazon Alexaan, Google Assistanttiin, Microsoft Teamsiin tai Slack-viestimeen.



Kuva 11. Keskustelua virtuaaliassistentin kanssa

Davis virtuaaliassistentti pääsee lukemaan kaikkea dataa Dynatrace-palvelusta autentikoidumalla REST API-rajapinnan kanssa. Rajapintaa varten luodulla kirjautumistunnuksella pystyy rajaamaan virtuaaliassistentin oikeuksia.

5 Haastattelu ja sisällönanalyysi

5.1 Haastattelu tutkimusmenetelmänä

Osana opinnäytetyötä haluttiin myös ymmärtää paremmin, millä tavalla yritykset hyödyntävät oikeassa liiketoiminnassaan AIOps-menetelmiä ja tuotteita. Haastattelututkimukseen valikoitui muutama suomalainen yritys, joilla oli Dynatrace-tuotteen käyttöönotto meneillään tai jo aktiivisessa käytössä. Kyseisten yritysten edustajille suoritettiin puolistrukturoitu haastattelu, joka sisälsi ennakoon määritellyt 16 kysymystä. Haastateltaville haluttiin kuitenkin antaa vapaus puhua myös muista aiheeseen liittyvistä asioista ja näin tapahtuikin paljon, jolloin haastattelu sai paljon myös teemahaastattelumaisia piirteitä. Puolistrukturoidusta haastattelusta poiketen kysymyksiä harvoin päästiin esittämään halutussa järjestyksessä, koska haastateltavien henkilöiden vastaukset olivat välillä erittäin laajoja ja kumpuilevia osoittaen laajan tietämyksen sekä tietyn intohimon aiheen ympärillä. Haastattelut videoitiin ja litteroitiin automaattisesti, joita käytettiin opinnäytetyön muistiinpanoina. Tämä osoittautuikin erittäin tärkeäksi ja avustavaksi asiaksi, koska tarkkojen viittausten tekeminen haastattelun jälkeen olisi ollut mahdotonta. Myös vastauksien laajuuden takia niiden kirjaaminen haastattelun aikana oikeisiin kysymyksiin olisi ollut mahdotonta.

Haastattelun lopputuloksessa yritykset, toimialat ja haastateltavat henkilöt on tarkoituksella anonymisoitu, koska tarkoitus oli kerätä kokemuksia ja ideoita aiheen ympäriltä, eikä niinkään dokumentoida kyseisten henkilöiden tai yritysten mielipiteitä tai menetelmiä. Haastattelun viittauksissa mainitaan Dynatrace-tuote useaan kertaan, mutta tärkeimpänä asiana on pyritty pitämään menetelmien tuomat lisähyödyt eikä niinkään keskittyä Dynatraceen tuotteena.

Viittauksissa on pyritty noudattamaan mahdollisimman sanatarkkaa lainausta, mutta aiheeseen lisäarvoa tuomattomat kohdat (sivulauseet, mieleen tulleet huomiot) on saatettu tiivistää kolmella pisteellä (...). Jokainen viittaus on omalla rivillään ja yhteen kysymykseen on kerätty useita viittauksia. Kysymyksestä on aina viittauksien lisäksi kerrottu syitä, miksi kysymys on tärkeä sekä yhteenvetoa vastauksen aiheuttamista huomioista.

5.2 Haastattelun tulokset

5.2.1 Mitä ongelmaa lähdettiin ratkomaan

Monitoroinnin tuotteet tuovat aina kustannuksen mukanaan ja yleensä liike-elämässä kustannukselle pitää olla riittävä syy, että se saadaan budjetoitua. Helpoiten syitä saadaan suoran ongelman ratkaisemisella, koska yleinen ”näkyvyys” tai ennakoitavuus on

hankalammin perusteltavissa lukujen tai raporttien kautta liiketoiminnan budjetoinnista vastaaville. Useimmiten isoilla asiakkailta, joiden kriittiset liiketoiminnanjärjestelmät tulisivat haetun monitoroinnin pariin, on helpompi ymmärtää tämä asia ja siihen liittyvät kustannukset, koska kriittisten järjestelmien toimimattomuus aiheuttaisi tarpeettomia kustannuksia sekä vahinkoa yrityksen maineelle/liiketoiminnalle.

Vehkeet on käytettävissä ja toimivat niin kuin pitää. .. Itse valvonta tilataan sovellusten toimesta ja vastuu säilyy heillä

Heillä on niin iso järjestelmän uudistushanke ollut käynnissä .. ja on tullut paljon uutta teknologiaa ja pilvipalveluita ja pilvipalvelun kaltaista konttimailmaa ja microservice-mailmaa ... ja se rupeaa olemaan niin iso kokonaisuus ja kaikki järjestelmät kuitenkin integroitu keskenään, että sitä ei enää tällain palapelimaisesti kyetä rakentamaan riittävän hyvää kuvaa

Tuotannonvakautushanke, eli niillä on ollut ongelmia, eikä ole ollut riittävää näkyvyyttä eikä selvyyttä siihen, mikä siellä tuotannossa mättää ja siihen on käytetty hirveästi aikaa ja paukkuja

Yritetään tässä tavallaan, että Dynatracesta saataisiin niin yksinkertainen käyttää ja työkalu niin monelle taholle näyttämään end-to-end näkyvyyttä aina sieltä selaimen klikkauksesta mainframen tietokantaan saakka

Saataisiin monitorointi kommunikaation keskiöön

En pysty sanomaan, että olisi ollut yksi tai useampi semmoista ,että se olisi ollut ikään kuin se ajava tekijä

Haastatelluilla yrityksillä ei välttämättä ollut mitään suurta yksittäistä tavoitetta tai ongelmaa, jota monitorointituotteella olisi lähdetty ratkomaan, vaan sillä haettiin enemmän nykyaikaista ja modernia työkalua, jolla pystyy keskitetysti valvomaan ympäristöjä, joita löytyy niin pilvipalveluista kuin on-premises-konesaleistakin. Monitoimittajaympäristö asettaa myös omat haasteensa, koska jokaisella toimijalla on omat tapansa toimia ja näiden yhtenäinen näkyvyys voi jäädä heikoksi. Haastatelluilla yrityksillä oli joko Dynatrace-tuotteen vanhempi versiota käytössä tai sitten aivan toinen tuote, ja käyttöönoton yhtenä tavoitteena usein olikin vain versioiden ajan tasalle saattaminen ja sitä kautta uusien ominaisuuksien käyttöönotto tai sitten vertailevan tuotteen rinnalle tuominen.

5.2.2 Mitä palveluita lähdettiin monitoroimaan

Monitoroitavat palvelut voivat olla erittäin laajan kirjon omaavia palveluita, kuten virtuaalikoneita, virtuaalikoneella pyöriviä sovelluksia, pilvipalvelussa pyöriviä PaaS-alustan sovelluksia tai jopa SaaS-palveluita. Kaikissa näissä on erilaiset vaatimukset ja erilaiset mahdollisuudet monitorointiin sekä vianhallintaan. Perinteisesti eri alueella toimiviin palveluihin on ollut omat toimialuekohtaiset valvontasovelluksensa, mutta palvelukirjon kasvaessa tämä asia vaikeutuu jatkuvasti.

Paljon on-premiä, .. azuren päälle siirretty ... ulkoisten käyttäjien käyttöliittymät ja niiden tarvitsemat mikropalveluarkkitehtuurikerrokset.

[satoja] hosteja valvonnassa ... Hostien koko vaihtelee pienemmistä mikropalveluista ... fyysisiin tietokantoihin. Mikropalvelut on tyypillisesti AWS-pilvessä, azuressa on jotain keskikokoluokan palveluita.... Valtaosa isoimmista kriittisimmistä järjestelmistä on meidän omassa vmwaressa ja sitten on jotain yksittäisiä tietokantoja, on ihan fyysisinä palvelimia.

Verkkokaupat... itsepalvelukanavista B2C ... Yrityspuolen vastaavat... Integraatioalusta, integraatiokerros mikä on x määrä palvelimia, SSO IDM ... mikä liittyy ulkoisiin niin näihin ulkoisiin webbikauppoihin että kaikki sisäiset kirjautumiset... Erilaisia tautajärjestelmiä, prosessijärjestelmiä, jotka pyörittää prosesseja vie tilauksia eteenpäin ... Kivijalkakaupat ... niin siellä on semmoinen järjestelmä mitä valvotaan .. ja yksittäisiä jopa kassapisteitä... ITSM tuotannonjärjestelmät.

Haastatelluilla yrityksillä olikin erittäin laaja-alaisesti erilaisia palveluita perinteisistä mainframe-ratkaisuista, mobiilisovelluksista, integraatorajapinnoista sekä kaikista erityyppisistä sovelluksista sekä tietokannoista. Tämä havainto onkin todella hyvä esimerkki nykyaikaisten järjestelmien vaatimuksesta sekä vaativuudesta. Mikäli käytettäisiin perinteisiä, tarkkoja ja tehokkaita yhden tuotteen/alueen monitorointisovelluksia, pitäisi niitä olla useita ja niiden integroiminen aiheuttaisi jo oman huomattavan operointikuormansa. Haastatellut nostivat useasti esiin sitä seikkaa, että monitoroitavaksi valittiin alkuun järjestelmiä, jotka ovat eniten näkyvissä asiakasrajapinnassa, koska niiden valvonta oli kuormittavinta ja näin helpoiten perusteltavissa liiketoiminnan päättäjille.

5.2.3 Oliko jokin tuote jo aiemmin käytössä?

Laajemmissa ympäristöissä on saatettu käyttää jo paljon aikaa ja rahaa olemassa olevan monitorointiratkaisun hyödyntämiseen ja laajentamiseen. Tällöin vanhasta järjestelmästä luopuminen on vaikea ja tarkasti harkittava päätös. Vanhempiin monitorointiratkaisuihinkin

on tuotu jälkikehityksenä tekoälyn omaisia lisäominaisuuksia, mutta usein niiden toteutus tuntuu päälle rakennetulta eikä kiinteältä osalta itse tuotetta.

Splunk & Wily Infrascop / Broadcom APM

Splunkissa on kanssa lisäosia ITSI (IT Service Intelligence), mitä POC:iin kahteenkin kertaan ... ajatuksena, että millä saataisiin (asiakkaalle) keskitettyä näkymää infra- ja sovelluksista. ... Toimiiko kriittiset järjestelmät ja toimiiko ne oikein ja miten ne näkyvät loppuasiakasta. ... Osoittautui haastavaksi, koska se vaati tavallaan tosi syvällistä järjestelmien tuntemusta ja osaamista

Dynatrace AppMon, että DCRUM samaan aikaan

Haastatelluilla yrityksillä oli joko täysin eri tuotteita tai vanhempia versioita Dynatracesta käytössä. Uuden tuotteen tai uuden version käyttöönotto suoritettiin pienin askelin ja harkiten, koska vanhan version hyvistä puolista ei luonnollisesti haluttu luopua sekä uuden monitorintisovelluksen käyttöönotto voi vaikuttaa ympäristön vakauteen negatiivisesti. Monitorintisovellukset, varsinkin APM-tuotteet, injektivat usein omaa toiminnallisuuttaan osaksi monitoroitavaa sovellusta, joten huolellinen testaaminen oli tärkeää. Myöskään kaikki tuotteet eivät toimi samaan aikaan asennettuina, vaan aiheuttavat kilpailua toistensa seurannassa.

5.2.4 Miten tuote valikoitui?

Uudenkaltaisia monitorintituotteita on markkinoilla useita. Usein sovellukset eroavat ominaisuuksistaan ja hinnoiltaan tai lisensoinneiltaan merkittävästi, joka on todella merkittävä osa liiketoiminnan päätöstä. Tuotteesta täytyy myös ottaa huomioon, onko sen testaaminen Proof-Of-Concept-menetelmällä mahdollista vai pitääkö tuotteen kanssa sitoutua samantien pitkäksi aikaa. Useat tuotteet sallivat nykyaikana pienimuotoiset POC:it, joten tuotteen valikoiminen koelisensseillä ja rajoitetuilla asennuksilla antaa usein jo pätevän kuvan monitorintisovelluksen mahdollisuuksista ja ominaisuuksista.

Otetaan se [dynatrace] mukaan POC:iin, että siinä on paljon potentiaalia tähän meidän ongelma-alueeseen

Dynatracen yksi isoimpia etuja on one-agent-teknologia, että asennat hostille tai pilveen otat käyttöön yhden agentin, joka pyrkii automaattisesti haistelemaan, että minköinen hosti tämä on ja minne tämä kommunikoi

Dynatrace tulee syrjäyttämään toisen APM tuotteen ... Splunk tulee tietysti jäämään ... se on [asiakkaalla] nimetty keskitetty logienhallintajärjestelmä. Dynatracessa on myös mahdollisuutta tallentaa myös logeja, mutta sitä ei ole tarkoitettu siihen.

Se minkä tää niin sanottu Dynatrace-tuote toi lisää, niin oli se tekoälypuoli ja sen niin kuin agenttien asennuksien helppous

... lähettiin pilotoimaan, kokeilemaan uutta Dynatracen SAAS-pilveä ... sitten jossain kohtaa tehtiin päätös, että eiköhän me siirrytä kokonaan sitten tähän uuteen tuotteen. ... Todettiin, että ehkä se managed on meille parempi.

Haastatelluilla yrityksillä oli joko aiempia kokemuksia Dynatrace-tuotteesta tai kontakteja, joiden kautta tuotetta oli suositeltu. Käyttöönoton helppous sekä siten POCin mahdollisuus oli merkittävä tekijä tuotteen valikoitumisessa. Tuotteella oli myös haastateltujen henkilöiden mukaan tehokas ja hyödyllinen tekninen myyntiorganisaatio, joka auttoi merkittävästi myös sovelluksen käyttöönottoa ja hyödyntämistä alkuvaiheissa. Tuotteen valitsemisessa oli merkittävänä osana myös datan sijainti. Kerätty data sisältää usein arkaluontoista tietoa, jolloin sen säilyttäminen vähintään EU-alueella, mutta mieluiten yrityksen omissa koneleissa oli tärkeä asia.

5.2.5 Mitä tuotteen osia on käytössä aktiivisesti?

Useassa monitorointituotteessa on modulaarinen rakenne ja niitä voidaan ottaa käyttöön joko suoraan, lisämaksun tai laajennetun lisensoinnin avulla. Käyttöönottoprojekteissa kaikkia ominaisuuksia ei välttämättä kannata heti alussa ottaa käyttöön, jotta kerätyn tiedon määrä ei hankaloita valittujen osioiden konfigurointia.

Davis-tekoälypuoli, joka baselinaa automaattisesti asioita ja pyrkii nostamaan niitä silmille

Real-user-monitorointipuoli on aika mielenkiintoista

Sitä pyritti Dynatracessa just saamaan, että asennetaan agentit, agentit rupee piirtämään karttaa infrasta mten se kommunikoi

Meilläkin on hirveän niin kuin heterogeeninen joukko käyttäjiä ... niin siellä on chättituki auki 24/7

Mulla on esimerkiksi omia henkilökohtaisia testienvirtoja, missä teen testejä... datamonitoinnit eivät näy kellekään muulle muuta kuin mä näen ne vaan itte

Jos me otetaan asiakkaita ... niin voidaan heille tehdä omia enveja ... Se tekee nopeaa ja ketterää tekemisestä

Käytetään tiettyjä plugareita. Esimerkiksi meillä on isojakin tietokantoja silleen, että meillä ei ole agenttia asennettuna ollenkaan sinne vaan me hyödynnetään Oracle Dynatrace plugaria

VMware näkyvyys tuotiin meidän dynatraceen

Tietyissä tilanteissa on hyödynnetty kertaluontoisesti selaimen lisäosia

Selainrobotti, mikä ikään kuin mukailee oikeaa sitä oikean käyttäjän käyttäytymistä ... mut sit aika lailla me käytetään niin kuin yksinkertaisempaa http-robottia

Haastatelluilla yrityksillä oli huomattavan erilaisia tarpeita monitoroinnissa. Yhtenäistä oli ominaisuuksien tarve, jotka suorittivat ympäristön kartoitusta sekä käyttäjille näkyvien rajapintojen monitorointi. Osaan ympäristöistä käytettiin myös yrityksen oman sisäisen toiminnan monitoroinnin lisäksi heidän asiakkaidensa ympäristöjen monitorointiin. Osassa tapauksista käytettiin hyvinkin tarkasti tuotteiden erilaisia ominaisuuksia, jotta saatiin maksimaalinen teho ja näkyvyys irti, mutta kuitenkin kustannustehokkaasti.

5.2.6 Hyödynnättekö tietoja mittaristoissa kuten SLA, SLO tai SLI?

Näkyvyyden yksi tuoma hyöty on mittaus ja täten eri mittaristot. SLA (palvelutasosopimus) on hyvin perinteinen tapa mitata palveluita, mutta ilman kaiken kattavaa end-to-end-monitorointia hyvin hankala mittari hyödyntää. Googlen SRE-menetelmää noudattavat yritykset asettavat usein SLO- (palvelutasotavoite) ja SLI- (palvelutasoindikaattori) mittareita, jotka ohjaavat tuotekehitystä ja palvelun ylläpidollisia tehtäviä.

Ei olla vielä, se on se mitä se raportoi itse

Haastatelluilla yrityksillä ei ollut tarkemmin tiedossa, käyttikö mikään yksikkö yrityksen sisällä tuotetta näiden mittarien muodostamiseen, mutta pitkänajan tavoite on hyödyntää tätäkin puolta näkyvyydessä. Haastatelluilla oli hyvin tiedossa tuotekehityksen laadun merkitys palveluiden stabiilisuuden tekijänä ja näiden mittarien hyödyt asian seuraamisessa.

5.2.7 Oliko alkuvaiheessa kovia tavoitteita vai uteliaisuus mahdollisuuksista?

AIOps-työkalujen myyntimateriaaleissa on luvattu merkittäviä aika- ja rahastäästöjä, joita tekoäly tuot tullessaan. Näillä perusteilla tuotteita on ollut helppoa myydä, koska tekoälyn hyödyntäminen on saatu kuulostamaan helpolta ja jopa automaattiselta.

Kovasti pidetty mielessä, että niin hyvältä kuin näyttääkin, niin se on kuitenkin vaan työkalu. ... Sama metriikka ja sama data tietystä mielessä on ollut aikaisemminkin

käytössä, että Davis tuo vähän uutta, ettei tarvitse itse keksiä, miten näitä eri metriikoita pyritään korreloimaan

Projektissa pyritään olla ... tuotannon häiriöhallinta edellä .. ja tietysti tulevaisuudessa ... devops-maailma, jos mieltii mihin kehittäminen on mennyt. ... CI/CD pipeline .. tavallaan jatkuvan julkaisemisen malleihin mistä puhutaan. ... Testiautomaatio tuottaa ja koodi viedään tavallaan laatuporttien läpi valutetaan tuotantoon, vaikka joka yö... Sehän tarkoittaa monitoroinnin automatiikkaa pitää saada nostettua ja kyvykkyyttä ja siihen dynatrace pystyy varmasti meitä auttamaan.

Saadaanko yksinkertaistettua tota näkymää heille Dynatracen avulla niin paljon, että järjestelmän omistaja tai liiketoiminta saisi siitä hyötyä omaan päätöksentekoon, ... miksi odottaa kuukausiraporttia ... jos voi katsoa livenä miten busineksellä menee

Otetaan tää tähän rinnalle, että saadaan reilu vertailukohta

Kyllä siinä vähän mainospuhettakin oli ... ”Tämä on vähän kuin Dynatrace AppMon steroideissa” ... kyky integroitua muihin järjestelmiin, mikä käytännössä tarkoittaisi sitä, että ... self-healing ominaisuus olisi mahdollista tässä

Haastatelluilla yrityksillä oli kuitenkin sen verran taustaa ja kokemusta vanhemmista versioista tai muista tuotteista, että myyntipuheista huolimatta odotukset pysyivät realistisina. Haastatellut osasivat ottaa vertailukohdan aiemmin tuotteen ominaisuuksista ja toivoa modernia lähestymistapaa alkuun vähintään yhtä hyvillä lopputuloksilla. Myyntipuheet tekoälyn ylivoimaisuudesta osattiin laimentaa, jotta pysyttiin realistisissa odotuksissa. Globaali näkyvyys monitoroinnissa oli yksi suuri osa-alue, johon odotettiin parannusta uusien tuotteiden myötä, koska näkyvyyttä haluttiin laajentaa suoraan liiketoiminnalle ja tällöin monitorointidata pitää pystyä osoittamaan sopivalla teknisellä tasolla.

5.2.8 Saavutettiin tai löytyikö hyötyjä?

Käyttöönoton onnistuminen tai tuotteen hyötyjen arviointi riippuu paljon tuotteelle asetetuista odotuksista. Aiempi kysymys osoitti, että haastatelluilla yrityksillä odotukset olivat realistisia. Mikäli valittu tuote ei olisi saavuttanut edes realistisia tavoitteita, niin varsinkin POC:ssa olevat käyttöönotot olisivat todennäköisesti jääneet POC:n tasolle.

Yritetään saada laadukas käyttöönotto valmiiksi ja semmoiset peruskäyttäjien koulutus, että saadaan kriittinen massa ylitettyä eli ihmiset käyttämään Dynatracea... Ei voi kutsua käyttämään liian ajoissa, jos se ei näytä hyvältä. Järjestelmät on vähän miten sattuu ja transaktioketjut eivät ole riittävän täydellisiä

Integraatioita on niin hemmetin paljon ja ... miten ne on ... julkaistu jonnekin ... tekninen rajapinta millä nimellä se löytyy ... se voi olla jotain ihan muuta mitä dokumentaatioissa lukee. Se oli ihan salapoliisityötä ... Minne tää menee, Mikä komponentti tämä on mitä tämä kutsu ja tällaisen pitkän kutsuketjun seuraamiseen meni ... päivä ja useita soittoja eri asiantuntijoille. ... Sitä kautta se rupesi positiivisesti tuomaan niitä asioita.

Kyllä se nostaa ihan näitä relevantteja asioita eri tavalla, toki se vaatii vähän aikaa, että tekoäly tottuu dataan

Haastatelluilla yrityksillä oli jäänyt positiivinen kuva ominaisuuksista suhteutettuna odotuksiinsa. AIOps-tuotteen ominaisuudet, joka auttoivat hankalien integraatioiden ymmärtämisessä tai muiden aiemmin huomaamattomien asioiden esiin nostamisessa. Käyttöönottossa ja sen laajentamisessa piti ottaa kuitenkin huomioon, että keskeneräisen monitoroinnin julkaiseminen käyttöön olisi saattanut heikentää käyttäjien luottamusta uuteen järjestelmään.

5.2.9 Mikä on vaatinut eniten oppimista tuotteessa?

Tuotteet, jotka sisältävät paljon ominaisuuksia ja konfiguroitavaa, vaativat tietynlaisen oppimiskäyrän siinä, että mikä on hyödyllistä ja mikä ei. Järjestelmässä asetetut oletusarvot eivät välttämättä ole asiakkaan ympäristöön sopivimmat ja saattavat antaa jopa virheellistä tietoa.

Vaikkakin tämä Dynatrace kykenee paljon tehdä itsenäisesti, niin eihän se sitä tarkoita, että se kytketään päälle ja se antaa järkeviä tuloksia... Tuossa on meillä vielä pitkä matka

Toki se vaatii vähän aikaa, että tekoäly tottuu siihen dataan .. vaikka kuukauden ajalta miten eri palvelut elää siellä. Niin ne False-positive tyypiset liput alkaa tippumaan ja pystyy oikeasti keskittymään oikeisiin haasteisiin, mitä tuolla on

Meillä on tämä hieno työkalu dynatrace, vaan katsellaan tätä ja ongelmat ikään kuin itsestään ratkeaa, että siellähän on tekoälyt ja kaikki ... Mutta eihän se niin mene, se vaatii niinku ne ihmiset ja organisoitumisen asian ympärille

Haastatelluilla yrityksillä on ollut toimittajan vahva tekninen myyntiorganisaatio apuna alkuvaiheen konfiguraatioissa, jotta tuote saataisiin mahdollisimman hyvin hyödynnettyä. Tekoälyn on puhuttu ratkaisevan kaikki pulmat ja haastatelluilla on ollut tässä koulutuksellinen rooli, mihin tekoäly tässä yhteydessä pystyy ja mihin vielä ei.

5.2.10 Mitä asioita on löytynyt, jota ei manuaalisesti olisi löytynyt

AIOpsin yksi tärkeimmistä myyntivalteista on seurannan kautta ongelmien tunnistaminen ja ennustaminen. Ihmisen suorittama tunnistaminen on yleensä ongelmalähtöistä eli etsitään rikkinäistä sovellusta tai sen osaa. Tekoäly etsii virheiden lisäksi normaalista poikkeavaa käytöstä, joka saattaa olla piikki tai alenema kuormituksessa.

Ei välttämättä virheitä, mutta huonoa toteutusta tai vastaavaa

Tuolla palvelimella on muisti ihan liian korkealla ja se kaatuilee, tuolla on ihan outo generoitu SQL, jonka yksi käyttäjätransaktio generoi 25000 SQL-kutsua ... Sitten ihmettelette, että on vähän hidas vasteaika ollut

Just tämmöisiä muutama, jotka ei ole suoraan kovia virheitä, että joku on täysin rikki

Hän epäilee, että heiden kuormantasaajansa toimii väärin, että 90% kuormasta tuntuu menevän toisen kuormanjakokoneen kautta

Observabiliteetti ... on oikeastaan ollut se numero ykkönen, että täytyy saada näkyvyyt. Se on niin monimutkainen toi järjestelmäviidakko... Se vaatii ne työkalut, että sen näkyvyyden sinne saa

Siinä oli semmoinen perustavanlaatuinen bugi ... pitkiä viiveitä saml assertioissa. ... Huomattiin, että siellä on jonkinlainen bugi oltava ja aluksi toimittaja ei tiennyt ... mutta otti sitten sen työn alle

Haastateltujen henkilöiden projekteissa oli tehty löytöjä työkalun avulla, joita ei ilman työkalua olisi välttämättä löydetty. Suurimmat löydöt eivät olleet täysin rikkinäisiä sovelluksia, koska ne oli korjattu jo aiemmin, vaan esimerkiksi huonosti suunniteltuja arkkitehtuureita ja SQL-kyselyitä. Myös palveluiden epätasainen kuormitus nousi mittaristossa esiin ja näin pystyttiinkin paremmin varautumaan kyseisen vuoden Black Friday -kampanjoihin.

5.2.11 Sovellusten tai palveluiden automaattinen palauttaminen

AIOps muutoksessa on viimeisenä vaiheena ”toiminta”, joka on kuvattu kappaleessa Act. Tekoälyn toivotaan osaavan automaattisen viantunnistamisen ja juurisyyn etsimisen jälkeen osaavan toimia tavalla, joka palauttaa järjestelmän automaattisesti takaisin toimintakykyiseksi. Monesti pilvipalveluissa on tätä toiminnallisuutta jo tiettyyn rajaan asti rakennettuna, esimerkiksi verkkosivuja tarjoavan PaaS-palvelun healthcheck-kysely sivustolle epäonnistuu, jolloin automaattisesti palvelu provisioidaan uudelleen. AIOpsin muutoksessa työkalujen markkinointimateriaalit puhuvat automaattisesti palautumisesta termillä ”self-healing” tai ”auto-remediation”. Tämän automaattisen palautumisen koetaan olevan erittäin

tärkeä osa AIOpsia, koska ilman automaatiota, ihminen on kaiken keskiössä ja sitä nimenomaan pyritään välttämään mahdollisimman pitkälle.

Mitä se self-healing ylipäänsä tarkoittaa? Missä kohtaa voidaan todeta, että palvelu on oikeasti palautunut? ... Pitäisikö olla suoraan mitattavissa ennen eventtiä tapahtuneeseen aikaan ... miten saadaan käyttäjät takaisin palveluun

Yksinkertaisimmillaan voisin nähdä sen semmoisena just, että ... joku sellainen jossa ei oikeita käyttäjiä ole, joku prosessi menee jumiin mikä on kuin todetta aikaisemmin, että tämä käynnistämällä uudelleen tai jonkun parametrin muuttaminen ja käynnistäminen uudelleen palauttaa sen palvelun niin kuin siihen samaan tilaan, missä se olin ennen vikaantumista

Niin lähdeittäisiin tekemään tuolla dynatracen ja ansiblen kautta se, että me nähdään vaikka x-määrä virheitä ja tai nähdään esimerkiksi valtava muistinkäyttö tai valtava CPU:n käyttö ... ratkaise ongelma käynnistämällä vaikka prosessi uudestaan tai pahimmassa tapauksessa käynnistämällä koko palvelin uudestaan

Mikropalveluissa hyvänä esimerkkinä viime vuoden black Friday. Oli tietty mikropalvelu, joka sai hirveän kuorman tietynä päivänä ja tietynä kellonaikana ja se kuorma johtui siitä, että meillä lähti mainoskampanja... olikohan puoli miljoonaa asiakasta. ... kutsujen määrä paukahti sinne mikropalveluun mikä nosti sen virtuaalipalvelin CPU:t ihan tappiin ... ja siellä ei ollut mitään logiikkaa, että luo uus instanssi. Dynatrace nosti hälytyksen pystyyn ja siinä oli vielä sillein ... että kukaan ei oikein ehtinyt, jotain kaiken maailman palavereita oli juuri samaan aikaan, niin kukaan ei ehtinyt reagoimaan. Siinä meni 15-20 minuuttia niinku erittäin kallista myyntiaikaa hukkaan... Lesson learnt: koko putki täytyy tunnistaa.

Haastatelluilla yrityksillä oli muodostunut hyvinkin realistinen näkemys mihin tuotteilla voidaan nykyään päästä. Tuotteet nykyisellään ovat parempia tunnistamaan vikatilanteen ja keräämään sekä kategorisoimaan siihen liittyvän datan. Tämän datan pohjalta voidaan laukaista siihen liitettyjä tapahtumia. Työkalu ei käytännössä osaa tehdä tilanteelle mitään, mutta se osaa määritysten mukaisesti kutsua sopivaa rajapintaa, jossa ongelman korjauslogiikka sijaitsee. Myöskin palautumisen määrittely oli hankala, koska helposti palvelun voidaan sanovan olevan ”palautunut”, kun se vastaa teknisesti, mutta pitäisikö palautumisessa huomioida myös esimerkiksi tietokannan transaktioiden peruuttaminen tai käyttäjien kutsuminen eri kanavilla takaisin palvelun pariin. Automaattinen palautuminen tarkoittaakin nykypäivänä sitä, että virheen tunnistuksen jälkeen osataan kutsua sopivaa rajapintaa, joka osaa palauttaa järjestelmää sopivasti. Tämä voi olla eri ympäristöissä komentokehoteella

suoritettava skripti, Infrastructure-as-Code-työkalun hallintarajapinta kuten esimerkiksi Ansible Tower (Dynatrace 2018) tai tiketöintijärjestelmä ja tukihenkilö.

5.2.12 Miten on auttanut tuotekehityksessä tai ongelmien selvittelyssä?

Haastatelluilla henkilöillä ei ollut suoraa näkyvyyttä yrityksensä yksiköiden mahdollisesti hyödyntämiin ominaisuuksiin tuotekehityksessä tai kaikissa eri tavoissa selvittää tuotteiden ongelmia. Haastatellut henkilöt tiesivät kuitenkin, että tuotekehitystiimeillä on mahdollisuus käyttää aiemmin keskusteluja mittareita sekä tietysti tuotteen tekemiä havaintoja mahdollisissa toimintalogiikan ongelmissa.

Tämä oli haastatelluissa yrityksissä toisen vaiheen laajennuksen yksi tavoitteista, että käytössä olevaa monitorointituotetta saataisiin käytettyä myös laadullisessa mielessä osana tuotekehityspotkea.

5.2.13 Integraatiot tuotekehitykseen? Onko mukana DevOps-putkessa?

Ympäristön riittävä monitorointi antaa käyttäjille tarvittavaa infoa myös ympäristön muutoksista. Jos osataan sanoa esimerkiksi miten stabiili tai reagoiva ympäristö on ollut aiemmin, saadaan tieto versiopäivityksistä tai muista muutoksista, niin muutoksen jälkeinen statistiikka on suoraan verrattavissa tehtyjen muutosten laatuun tai muihin vaikutuksiin. Monitorointituotteet keräävät yleensä omaa yleisindeksiään ympäristöstä tai sen sovelluksista, kuten esimerkiksi Dynatrace ApDex (Dynatrace 2021a). Tämän indeksin muuttuessa huomponaan, voidaan syytä lähteä hakemaan ympäristöön kohdistuneista toimenpiteistä.

Hyväksymistestausympäristössä tää rupeaa olemaan instrumentoitu ihan sen takia, että kun tuotantoon ollaan viemässä tämmöistä asiaa, niin me ollaan jouduttu menemään perus kehityspotken kautta .. ja varmistettu ettei Dynatrace itsessään riko mitään

Jenkinssiin on integroitu ja muitakin pienempiä

Tahtotila kanssa, että tuosta pitäisi saada ammattimaisempaa jatkossa eli juuri pitäisi olla nämä quality gatet ja mä näen, että se on kanssa asia mikä meille on tulossa

Sisäinen kehitys ja kehitystarpeet kasvaa ja sitä myötä on niin kuin tarkoitus tuoda tätä dynatracea vahvemmin ja vahvemmin osaksi kaikkia kehityspotkia

Haastatelluilla henkilöillä ei ollut tarkasti tiedossa kaikkien tuotekehitystiimien käyttämät metriikat, mutta parhaan ymmärryksen mukaan tämä ominaisuus ei ollut laajasti vielä käytössä. Tekniset mahdollisuudet oli kuitenkin hyvin tiedostettu ja ne odottivat joko ympäristön

maturiteetin kasvamista tai yksinkertaisesti hyvää käytätapausta, jossa kehitystiimi olisi hyödyntänyt täysin mahdollisuuksia. Kysymys ”Hyödynnättekö tietoja mittaristoissa kuten SLA, SLO tai SLI?” käsitteli vähän samaa aihetta, koska SLO tai SLI -tavoitteiden pohjalta sovelluskehitystiimi olisi voinut ohjata niin omaa toimintaansa kuin automaattisten kääntö- ja julkaisuputkien (CI/CD -pipeline) toimintaa varsinkin yleisindeksiä laskevassa tilanteessa.

5.2.14 Tukitikettien (jos semmoisia oli aiemmin) määrän muutos?

Haastatelluilla yrityksillä ei ollut suoranaisia havaintoja tikettien määrästä ja tarkemman mietinnän jälkeen se olikin hiukan huonosti ajateltu kysymys. Tikettien määrä ei välttämättä kuvaa ongelmien määrää vaan perinteisesti loppukäyttäjien havaitsemien ongelmien määrää. Automaatiikan lisääntyessä ja valvonnan tarkentuessa tiketointijärjestelmä on looginen varoitusrajapinta näille automaattisesti tehdyille havainnoille. Tällöin tikettejä voi tulla alkuun jopa enemmän kuin ennen, koska käyttäjät ovat usein laiskoja tekemään ongelmaticettejä varsinkin toistuvista tai lyhytkestoisista ongelmista. Yksi AIOpsin peruspilareita on kuitenkin havaittujen ongelmien rikastaminen ja yhdisteleminen. Aiemmissa tapauksissa siis järjestelmän ongelma on saattanut tuottaa satoja tikettejä, vaikka yksikin olisi riittänyt. Toisaalta piilossa olleet ongelmat eivät ole näkyneet tiketteinä ja uuden järjestelmän myötä ovat saattaneetkin aiheuttaa merkittävän tikettimäärän kasvun.

Pelkästään tukitikettien määrä on huono mittari järjestelmämuutoksen onnistumiselle, ellei laskettavien tikettien tyyppiä ole tarkkaan määriteltä (esimerkiksi puhelimitse tulleet loppukäyttäjän ilmoittamat virheet).

5.2.15 Palautumisajan tai virheenselvityksen vaatiman ajan muutos?

Vaikkakaan järjestelmien ja palveluiden automaattinen palautuminen ei vielä tuntunut olevan riittävän pitkällä, kehittyneemmät virheenselvityksen toiminnot olivat hyödynnettävissä. Paranneltu vian tunnistaminen sekä juurisyyanalyysi näyttelivät tässä parannuksessa merkittävintä roolia.

Se mitä itse olen nähnyt ja kuullut tuotantopäälliköiltä, niin se on puolittunut jollei jopa niin kuin vielä sen alle mennyt

Tuotannon porukka mikä on yksi aktiivisemmista käyttäjistä, niin heillä on rakentunut ne prosessit valmiiksi. ... He on huomanneet, että mitä arvoa saa dynatracesta ja mitä ne saa muista järjestelmistä, mihin heillä on näkymä, niin ne osaa toimia fiksummin ja herättää ihmiset

Haastatelluilla yritykset olivat kokeneensa saaneen modernin työkalun käyttöönotosta huomattavia etuja nopeutuneena vianselvityksenä sekä pienentyneenä palautumisaikana. End-to-End näkyvyys vaadittuihin järjestelmiin, juurisyysanalyysit sekä kerätyt ja riittävästi rikastetut ehdotukset ja suositukset korjaaviksi toimenpiteiksi olivat vähentäneet virheenselvitykseen tarvittua aikaa jopa puolella. Tärkeimpänä virheenselvitykseen osattiin kutsua vain oikeat henkilöt ja perinteisesti IT War Roomeista oli pystytty osittain luopumaan.

5.2.16 Mitä mielestäsi AIOps tarkoittaa?

Gartnerin virallinen määrittely termille AIOps on esitelty kappaleessa ”Määrittely”, mutta kyselyn lopussa yritettiin tarkentaa sitä, mitä se tarkoittaa haastatellulle henkilölle henkilökohtaisesti.

Jos ylipäättänsä automatisoidaan sovelluskehityspotkia ja sinne eri teknologioilla rakennetaan automaattitestejä aina kun buildataan joku sovellus uusiksi. ... Sinne semmoista koneoppimista edellisiltä kierroksilta, että mitä se laatu on. Siihen päälle sitten monitorointikyvykyys ... ja siihen päälle sitten koneoppimista eri kierroksilta.

Antamaan feedbackiä ihmiselle .. tai jopa rakentamaan automaatiota automaattiseen recovery eli rollataan takaisin muutos

Automaatioastetta nostetaan joka paikassa ja sit ehkä just machine Learning tyypistä baselinausta, mistä kone voi kertoa, että hei tää toimii anomalisesti. .. Siihen tulevaisuudessa lisää heuristiikkaa.

Infra tasolla, että se rauta kestää sen tulevan kuorma jopa vähän niinku ennustaa niinku menneiltä vuosilta ja näin päin pois niitä esimerkiksi liikennemääriä siellä infraassa. Että hei että tammikuussa täällä pitää olla konetehoa ja nostetaan jo etukäteen resurssi, mutta ei pidetä sitä turhaan päiten panttina

Operoinnin, vianhallinnan, kehityksen, testauksen kehitys ... tuotteiden suunnittelu, deployment on kaikki tekoälyn piirissä, missä tekoäly on läsnä enemmän tai vähemmän, mikä itse päättelee asioita mitä tulee seuraavaksi tapahtumaan.

Kriittisimmät järjestelmät dynatracen piirissä niin kuin alusta loppuun asti softakehityksestä testaamiseen, deployment ja vianhallinta ja tietysti tää putki taaksepäin eli siellä ne quality gatet pitää huolen missä kohtaa se palautetaan

Haastatelluilla henkilöillä oli suhteellisen laaja kirjo siitä käsityksestä, mitä AIOps heidän mielestään on. Henkilön mielikuvaan tuntui vahvasti vaikuttavan se, miten ja mihin esimerkiksi analysointityökaluja heidän ympäristössään käytettiin sekä mikä oli heidän työuransa

tausta. Vastauksista nousi useimmiten resurssien skaalautuvuuteen liittyviä ajatuksia, joihin taas liittyy normaalitason arviointia (ns. baseline) sekä anomaliteettien etsintä. Usein ajatusta ei viety edes kauhean pitkälle nykyajasta ja nykymahdollisuuksista vaan jopa todettiin, että teknologia vielä kehittyy, sitä kutsutaan sitten omalla termillänsä.

5.2.17 Muita havaintoja

Haastatteluiden aikana haastateltavat puhuivat hyvinkin laaja-alaisesti projektistaan tai sen sisällöstä ja iso osa mielenkiintoisista havainnoista ei suoranaisesti osunut mihinkään kysymykseen. Tässä kappaleessa on kerätty näitä havaintoja talteen.

Jos mennään tähän monitorointiin ja sovellustenvilvontaan, niin ehkä tässä eniten yritetään muuttaa sitä, että ennen ollaan oltu pakollinen paha, että tämmöinen pitää hoitaa

Käytännön malli on ollut, että tämmöiset asiat kuin valvonnan miettiminen aloitetaan perinteisesti projekteissa ihan liian myöhään, jos tuotetaan jotain aivan uutta ja sitten on budjetit palanut jo

Kun kosketaan vähän kaikkiin järjestelmiin, niin onhan tässä tietynlaista muutosvastarintaa... Varsinkin pilviveikkojen kanssa, ne eivät tunnu ymmärtävän. Kun tekevät azuressa asioita ja sanovat, että azuren omat monitoroinnit näyttää nää meille ihan hyvin, asiantuntijat voi katsoa sieltä... Organisaatiossa nimenomaan halutaan end-to-end näkyvyyttä parantaa ... että päästään impaktivaikutukseen

Monitoroinnin koettiin usein olevan ”pakollinen paha” ja sen käyttöönottoa vaivalloisena prosessina. Varsinkin sovelluskehitysprojekteissa monitorointi jäi valitettavan viime tinkaankin ja usein hetkeen, kun projektille varattu budjetti oli käytännössä jo kulutettu. Aluksi yllätti se, että niin sanottujen ”pilviveikkojen” joukossa oli paljon muutosvastarintaa. Tämä selittynee sillä, että pilvipalveluissa koetaan jo olevan riittävän hyvät ja monipuoliset monitorointityökalut ja niiden käyttöönotto on helppoa. Tällöin ulkoisen monitorointiagentin asentaminen tuntuu ylimääräiseltä työltä ja hyppäykseltä pois projektien teknisestä itsehallinnosta. Myöskään yksittäisestä palvelusta vastuussa olevat eivät välttämättä näe koko kuvaa tai ymmärrä sen tärkeyttä, että iso kuva on keskitetysti hallinnassa.

Moneen se laitetaan AI leima, keinoäly, tekoäly. Musta tuntuu kuitenkin, että ihmis-kunta ja nää järjestelmät on enemmän semmoisessa koneoppimisen mallissa. ... Annetaan koneen oppia eri asioita ja yrittää löytää semmoinen yhteinen nimittäjä. En jaksaisi vielä oikein puhua tekoälystä ... ainakaan täällä monitoroinnin saralla.

Haastateltavat henkilöt kävivät myös mietiskelyä AIOps-termin ympärillä. Tekoäly on saanut niin suuren roolin monessa markkinointipuheessa, että käytännön asioita tekevät eivät halunneet sitä termiä välttämättä käyttää vaan jotain tarkempaa alitermiä kuten koneoppiminen tai algoritmit.

Mutta sitten on niin kuin tällaisia tapauksia, missä on vaan osa valvonnassa niin se vaatii melkein aina verifioida sen elementtien oikeellisuus jostain toisesta näkökulmasta, mitkä sitten hyvin löyhästi liittyy yhteen.

Viimeistelevänä havaintona oli end-to-end monitoroinnin tärkeys. Tarpeetonta dataa, jota ei saada liitettyä lähtö- tai kohdejärjestelmään, ei kannata kustannussyistäkään kerätä. Tämnäkaltaista orpodataa ei pystytä yleensä yksinänsä hyödyntämään mihinkään, vaan se vaatii tiedon rikastamista tai mieluiten liitosjärjestelmien monitoroinnin laajentamista.

5.3 Havainnot haastattelututkimuksesta

Puolistrukturoitu haastattelu oli toimiva malli, koska se ohjasi keskustelua ainakin hetkittäin oikeaan suuntaan ja ilman ennakkoon mietittyjä kysymyksiä, ei yhteenvetoa olisi pystytty tekemään. Haastateltavat olivat niin syvällä aiheessa, että käytännöllisesti haastattelu olisi voitu käydä ilman yhtään kysymystä, mutta oikeiden ja vertailukelpoisten vastausten saaminen olisi tällöin pitänyt unohtaa. Haastattelujen taltiointi ja automaattinen litterointi Teams-sovelluksella oli järkevä ratkaisu, koska haastattelujen purkaminen videolta opinäytetyöhön otti lopulta melkein tuplasti sen ajan, mitä itse haastattelut kestivät.

Haastatteluissa tuli ilmi monitoroinnin ja vianhallinnan monimuotoisuus sekä nykyaikainen vaativuus järjestelmien kasvaessa. Järjestelmien monimuotoisuus asettaa niin valitulle työkalulle kuin sitä käyttäville ammattilaisille suuren haasteen. Haaste muodostuu toisaalta hallittavan datan määrästä kuin myös hallittavien sovellusten määrästä. Jokainen sovellus saattaa noudattaa vähän erilaista kaavaa toteutuksessa ja toiminnassa, jolloin jokaiselle sovellukselle saatetaan joutua asettamaan sovelluskohtaisia rajoja ja sääntöjä. Tämä ei ole kuitenkaan haluttu tila, vaan monitoroinnin helpottamiseksi sovelluksien yhtenäistäminen (nimeämiskäytännöt, teknologia, ajoalusta) ainakin joltain osin on tarpeellista.

Yhtenä myös merkittävänä asiana on liiketoiminnan mukaan tuominen monitorointiin. Aiemmin monitorointi oli IT-osaston vastuualueella, mutta nykyaikaiset työkalut tuovat sitä enemmän ja enemmän myös liiketoiminnan nähtäväksi. Tällöin käytetty kieli termeineen ja näytetyt asiat täytyy luonnollisesti kustomoida liiketoimintapohjaisiksi, mutta tämän myötä ohjelmistojen ja palveluiden mahdollinen kehittäminen tai havainnointi on laajemman väkimäärän vastuulla. IT-osasto vastaa vieläkin sovelluksen toiminnasta, mutta liiketoimi voikin vastata esimerkiksi uusien ominaisuuksien käyttöasteesta tai uudelleensuunnittelusta,

mainoskampanjoiden toimivuudesta suhteessa liikevaihtoon sekä yleisesti palvelun laadullisista ja taloudellisista mittareista.

Mielenkiintoisena seikkana oli muutoksen aiheuttama muutosvastarinta. Sitä koettiin yllättävän vähän, koska usein uuden työkalun käyttöönotto ei vaatinut sovelluksien kehitystiimeiltä käytännössä mitään työtä ja usein he kokivat saavansa jotain lisäarvoa uudesta työkalusta. Toisaalta sitten taas kehittäjät, jotka käyttivät jo mielestänsä moderneja työkaluja (pilvipalveluiden natiivit työkalut, kuten Azure Monitor ja Insights) eivät välttämättä nähneet tässä mitään hyödyllistä vaan pikemminkin haittaavaa toimintaa, koska oletuksena kahta APM-monitorointityökalua ei voi pitää samaan aikaan käytössä, etteivät ne esimerkiksi monitoroi toisiaan. Käytännössä siis sovelluskehityksen ”aallonharjalla” matkustavat kehittäjät olivat kovaäänisempiä vastustajia tälle muutokselle.

Haastateltavat nostivat myös esiin tekoäly-termin käyttöä. AIOps antaa ymmärtää, että tekoäly on keskiössä ja niin se käytännössä onkin, mutta ei ehkä ihan siinä laajuudessa kuin haluttaisiin sen olevan. Tekoäly on näissä tuotteissa edistynyttä, mutta ei kuitenkaan vielä sellaista, jota asiakkaat haluaisivat sen olevan ja kuten toimittajat sitä heille myyvät. Realistiset odotukset ja tarkka arviointi auttoivat kuitenkin haastateltuja näkemään oikeat hyödyt, joita AIOps toi heidän ympäristölleen eivätkä jumiutuneet myyntilupauksiin.

Pohjimmiltaan AIOpsin merkitys ja määrittely oli myös hyvin erilainen, mistä kulmasta sitä katsoi. Toiset kokivat sen olevan täysin automaattinen ja kaiken ratkaiseva tekoäly, kun taas toisaalta sen koettiin olevan myös sitä, että kaikkien (järkevien) järjestelmien kaikki (järkevä) data kerätään talteen, analysoidaan ja tarvittaessa käytetään toisen järjestelmän toiminnan parantamiseksi.

6 Johtopäätökset ja pohdinta

Tämän opinnäytetyön tutkimusongelmina oli selvittää mitä AIOps on ja miten sitä pystyttäisiin hyödyntämään IT-järjestelmien virreehallinnassa ja monitoroinnissa. AIOpsin määrittely on hyvin laaja ja kaikenkattava, jossa on listattu isoja tekniikoita. Määrittely jättää siis paljon valinnanvaraa valittavalle sisällölle, toteutukselle sekä käytettäville tekniikoille.

Opinnäytetyön alussa kuvattiin erilaisia aiheeseen liittyviä operointimenetelmiä monitoroinnin ja ohjelmistokehityksen alueilta. Useat menetelmät ovat laajennoksia toisistaan, joten muiden menetelmien tunteminen helpottaa ymmärtämään kaikkea taustalla olevaa ideologiaa ja menetelmiä. Seuraavaksi kuvattiin AIOps-käsitettä tarkemmin. Monitoroinnin perusrakenteet olivat ennallaan, mutta AIOps laajentaa sitä huomattavasti kehittyneillä tekoälyominaisuuksilla. Näiden ominaisuuksien eri osa-alueet käytiin lävitse ja samalla yritettiin määrittellä erilaisia käyttötapauksia, joita osa-alueen ominaisuuksilla voitaisiin ratkoa nyt tai tulevaisuudessa.

AIOps-tuotteiden tarjoamasta esiteltiin lyhyesti Dynatrace-tuotetta, joka on Gartnerin tilastojen mukaan merkittävä markkinajohtaja APM-tuotteiden markkinoilla. Tuotteen ominaisuudet käytiin lävitse AIOps-näkökulmasta lyhyesti kuvaten niiden mahdolliset hyödyt. Opinnäytetyön ei ollut kuitenkaan tarkoitus olla yksityiskohtainen dokumentti Dynatrace-tuotteen ominaisuuksista vaan ylätasoinen esittely, millä tavalla markkinajohtaja on ratkaissut erilaisia haasteita.

Opinnäytetyötä varten järjestettiin muutama puolistrukturoitu haastattelu henkilöiden kanssa, jotka ovat vastuussa vähintään keskisuuren ympäristön Dynatrace-käyttöön otosta tai ylläpidosta ja jatkokehityksestä. Haastateltavia oli aikataulupaineiden takia valittavan vähän, mutta anti oli silti merkittävä opinnäytetyön osalta. Haastateltavilta kysyttiin 16 eri tavalla monitorointiin tai AIOpsin peruseriaatteisiin liittyvää kysymystä. Kysymysten vastaukset käsiteltiin ja lisättiin viittauksina soveltuvin osin. Haastateltavien henkilöllisyys ja keskusteltavat yritykset anonymisoitiin tarkoituksella pois vastauksista, koska olennaisinta olivat näkökulmat, ideat sekä luonnollisesti käytännön toteutukset.

Opinnäytetyön tutkimuskysymyksiin saatiin vähintäänkin hyvää pohdintaa aihealueen sekä määrittelyn ollessa niin laaja. Ensimmäisessä tutkimuskysymyksessä saatiin esiteltä kahdeksan suurta osa-alueita käyttötapauksineen, joista muodostuu hyvä yleiskäsitys kaikesta mitä AIOps voisi olla. Osaltaan esitellyt osa-alueet vastasivat myös toiseen tutkimuskysymykseen, mutta haastattelu toi siihen suurimman näkemyksen. Ennako-odotusten mukaisesti kysymyksiin ei ollut yksinkertaisia vastauksia, vaan useita erilaisia vaihtoehtoja, jotka

ovat riippuvaisia ympäristöstä, jossa niitä hyödynnetään sekä henkilön osaamisesta ja kokemuspohjasta, joka niitä asioita hyödyntää.

Opinnäytetyön yhteenvedona voidaan todeta, että maailma ei ole vielä siellä, missä sen myydään olevan. Tekniseltä näkökulmalta kaikki asiat olisivat toteutettavissa, vaikka yritysten sisäisesti kehittämänä, mutta työ kuulostaa aluksi yksinkertaisemmalta kuin se oikeasti on. Tämä on usein koettu haaste tekoälyprojektien kanssa. AIOpsissa helpoimpina haasteina voidaan pitää datan keräämistä ja sen keskittämistä sekä ongelmien tunnistamista. Muut asiat, jotka nojaavat enemmän tekoälyyn, ovat vaikeampia ja vaativat enemmän kyseisen alan asiantuntijoita. Riippuvuussuhteita ja muita huomioitavia asioita on todella paljon ja tiedoista kasattu matriisi kasvaa valtavaksi.

Tietyillä operointiin liittyvillä osa-alueilla automaation taso on vielä matala ja luotto manuaaliseen tekemiseen korkea. Tämä osaltaan selittyy sillä, että perinteisesti hoidetuissa ympäristöissä manuaalisen työn määrä on korkea ja nojautuu sitä tekevien ihmisten ammattitaitoon. Operoinnin puolella, joskin maailma on muuttumassa, ohjelmointipohjainen lähestymistapa ei ole niin luonnollinen, kuin esimerkiksi DevOps-tiimissä, jolloin on huomattavan paljon suurempi kynnyks lähteä automatisoimaan asioita, koska ne usein pohjautuvat ohjelmoinnin peruseräisiin. Myöskin luotto automaatioon voi olla huono sen puuttumisen vuoksi.

AIOpsin käyttöönotto, tai minkä tahansa muun monitorointimenetelmän, joka tarjoaa kaikenkattavan observabiliteetin, tuo tullessaan aina kustannuksia. Kustannukset alussa saattavat olla merkittäviä, koska tehdään jotain, mikä koskettaa kaikkia pintoja IT-liiketoiminnassa. Lisääntyneen näkyvyyden ja keskitetyn hallinnan tuoma säästö ei yleensä ole suoraan mitattavissa rahana, joten käyttöönoton takaisinmaksuaikaa on vaikea laskea ja täten perustelu liiketoiminnalle voi joskus olla hankalaa. Liiketoiminnan tuominen lähemmäs monitorointia saattaa helpottaa tässä, koska monitorointi ei ole pelkkiä palvelinten metriikoita ja logeja, vaan myös liiketoiminnalle tärkeitä mittareita esimerkiksi verkkokaupan tuomasta liikevaihdosta, asiakkaiden tyytyväisyysindeksistä tai kampanjoiden konvertoitumisprosentti palvelun uusiksi käyttäjiksi.

Tärkeimpänä AIOpsissa on kuitenkin se, että jostain pienestä pitää lähteä liikenteeseen, jotta saadaan useita onnistumisia matkana varrella. Onnistumiset kasvattavat luottoa eri osastoilla ja liiketoiminnassa, jolloin siihen on helpompi sitoutua. Ilman sitoutumista, palvelusta ei saavuteta hyötyä lyhyellä eikä edes pidemmällä aikataululla. Gartnerin tutkimuksenkin mukaan, ei ole olemassa IT-operoinnin tulevaisuutta, johon ei AIOps kuuluisi.

Lähteet

Atlassian. 2021. How to set, measure and report on SLAs. Viitattu 29.11.2021. Saatavissa <https://www.atlassian.com/itsm/service-request-management/slas>

Beer, W. 2019. Understand complex issues instantly with new root-cause analysis capabilities. Viitattu 7.12.2021. Saatavissa <https://www.dynatrace.com/news/blog/understand-complex-issues-instantly-with-new-root-cause-analysis-capabilities/>

Breuel, C. 2020. ML Ops: Machine Learning as an Engineering Discipline. Viitattu 30.11.2021. Saatavissa <https://towardsdatascience.com/ml-ops-machine-learning-as-an-engineering-discipline-b86ca4874a3f>

Dynatrace. 2018. Set up Ansible Tower. Viitattu 30.11.2021. Saatavissa <https://www.dynatrace.com/news/blog/set-up-ansible-tower-with-dynatrace-to-enable-your-self-healing-applications/>

Dynatrace. 2019. AIOps Done Right. Viitattu 7.12.2021. Saatavissa <https://www.dynatrace.com/monitoring/platform/aiops/>

Dynatrace. 2021a. ApDex ratings. Viitattu 30.11.2021. Saatavissa <https://www.dynatrace.com/support/help/how-to-use-dynatrace/real-user-monitoring/basic-concepts/ratings/apdex-ratings/>

Dynatrace. 2021b. How problems are detected and analyzed. Viitattu 7.12.2021. Saatavissa <https://www.dynatrace.com/support/help/how-to-use-dynatrace/problem-detection-and-analysis/basic-concepts/how-problems-are-detected-and-analyzed/>

Dynatrace. 2021c. Managed deployments. Viitattu 7.12.2021. Saatavissa <https://www.dynatrace.com/support/help/setup-and-configuration/dynatrace-managed/basic-concepts/managed-deployment-scenarios/>

Dynatrace. 2021d. Meet Davis, our radically different AI-engine. Viitattu 7.12.2021. Saatavissa <https://www.dynatrace.com/platform/artificial-intelligence/>

Dynatrace. 2021e. Dynatrace ActiveGate. Viitattu 8.12.2021. Saatavissa <https://www.dynatrace.com/support/help/setup-and-configuration/dynatrace-activegate/>

Dynatrace. 2021f. Dynatrace Managed. Viitattu 8.12.2021. Saatavissa <https://www.dynatrace.com/support/help/setup-and-configuration/dynatrace-managed/>

Dynatrace. 2021g. How does Mission Control pro-active support work?. Viitattu 8.12.2021. Saatavissa <https://www.dynatrace.com/support/help/setup-and-configuration/dynatrace-managed/data-privacy/how-does-mission-control-pro-active-support-work/>

Dynatrace. 2021h. Dynatrace OneAgent. Viitattu 8.12.2021. Saatavissa <https://www.dynatrace.com/support/help/setup-and-configuration/dynatrace-oneagent/>

Dynatrace. 2021i. Dynatrace. Viitattu 9.12.2021. Saatavissa <https://www.dynatrace.com/support/help/how-to-use-dynatrace/problem-detection-and-analysis/basic-concepts/problem-impact-level/>

Forbes. 2018. 83% of Enterprise workloads will be in the cloud. Viitattu 29.11.2021. Saatavissa <https://www.forbes.com/sites/louiscolombus/2018/01/07/83-of-enterprise-workloads-will-be-in-the-cloud-by-2020/?sh=44db4aa86261>

Gartner. 2016. Innovation Insight for Algorithmic IT Operations Platform. Viitattu 30.11.2021. Saatavissa <https://www.gartner.com/document/3263717>

Gartner. 2021. Market Guide for AIOps Platforms. Viitattu 8.2.2021. Saatavissa <https://www.gartner.com/en/documents/4000217>

Gill, N, 2020. Overview of BizDevOps Architecture, Tools and Benefits. Viitattu 30.11.2021. Saatavissa <https://www.xenonstack.com/insights/bizdevops>

Kokonat. 2021. Dokumenttianalyysi. Viitattu 9.12.2021. Saatavilla <https://www.kokonat.fi/dokumenttianalyysi/>

Koppa. 2015. Empiirinen tutkimus. Viitattu 9.12.2021. Saatavissa <https://koppa.jyu.fi/avoimet/hum/menetelmapolkuja/menetelmapolku/tutkimusstrategiat/empiirinen-tutkimus>

Loukides, M. 2012. What is DevOps. Viitattu 2.12.2021. Saatavissa <http://radar.oreilly.com/2012/06/what-is-devops.html>

RedHat. 2018. What is DevSecOps?. Viitattu 30.11.2021. Saatavissa <https://www.redhat.com/en/topics/devops/what-is-devsecops>

Rumburg, J. 2017. Metric of the Month: Service Desk Cost per Ticket. Viitattu 7.12.2021. Saatavissa <https://www.thinkhdi.com/library/supportworld/2017/metric-of-month-service-desk-cost-per-ticket>

Saaranen-Kauppinen, A., Puusniekka, A. 2006. Strukturoitu ja puolistrukturoitu haastattelu. Viitattu 9.12.2021. Saatavissa https://www.fsd.tuni.fi/menetelmaopetus/kvali/L6_3_3.html

Sloss, B. 2017. Site Reliability Engineering. Viitattu 7.12.2021. Saatavissa
<https://sre.google/sre-book/introduction/>

Tack, S. 2021. Dynatrace again named a leader. Viitattu 30.11.2021. Saatavissa
<https://www.dynatrace.com/news/blog/dynatrace-again-named-a-leader-in-2021-gartner-magic-quadrant-for-apm/>