Tampereen yliopisto

Harri Lehtonen

JÄRJESTELMÄ- JA LUOTETTAVUUS-VAATIMUSTEN TOTEUTUMINEN PILVIPOHJAISESSA SCADA-JÄRJESTELMÄSSÄ

Diplomityö
Tekniikan ja luonnontieteiden tiedekunta
Tarkastajat:
professori Matti Vilkko ja
apulaisprofessori David Hästbacka
Helmikuu 2023

TIIVISTELMÄ

Harri Lehtonen: Järjestelmä- ja luotettavuusvaatimusten toteutuminen pilvipohjaisessa SCADA-järjestelmässä
Diplomityö
Tampereen yliopisto
Automaatiotekniikan diplomi-insinöörin tutkinto-ohjelma
Helmikuu 2023

Pilvipalveluiden käyttö perinteisessä IT-maailmassa on viime vuosien aikana lisääntynyt, ja teollisuuden neljännen vallankumouksen myötä pilvipalvelut ovat muodostuneet tärkeäksi osaksi myös teollisuusautomaatiojärjestelmiä. Teollisuusautomaatiossa tuotantoprosessin monitorointiin ja hallintaan käytetyn SCADA (engl. *Supervisory Control and Data Acquisition*) -järjestelmän pilvi-integraatio mahdollistaisi järjestelmälle perinteisen IT-maailman ominaisuuksia kuten skaalautuvuutta ja etäkäyttömahdollisuuden. Teollisuusjärjestelmien vaatimukset ovat kuitenkin perinteistä IT-maailmaa tiukempia, ja yksityisessä verkossa paikallisesti käytettäväksi suunnitellun SCADA-järjestelmän vienti pilvipalveluun vaikeuttaa näiden vaatimusten toteutumista entisestään.

Tässä työssä pyritään kirjallisuustutkimuksen avulla selvittämään toteutuvatko SCADA-järjestelmän pilvi-integraation kannalta oleellisimmat järjestelmä- ja luotettavuusvaatimukset riittävän hyvin pilvipohjaisessa toteutuksessa. Järjestelmävaatimuksista tutkitaan reaaliaikaisuuden ja luotettavuusvaatimuksista järjestelmän saatavuuden sekä tietoturvallisuuden ja kyberturvallisuuden toteutumista. Lisäksi kirjallisuuden pohjalta tutkitaan pilvipohjaisen SCADA-järjestelmän yritykselle tuomia etuja ja mahdollisuuksia.

Työssä todetaan pilveen liikennöinnistä ja pilvessä prosessoinnista aiheutuvien viiveiden olevan pieniä, ja siten reaaliaikavaatimusten toteutuvan riittävän hyvin täyttäen SCADA-järjestelmän kiertokyselyvälin asettamat vaatimukset verkkoyhteyksien normaalitilassa. Merkittäviä viiveitä ei myöskään synny kommunikaatioyhteyksien salauksesta tai pilvipalvelimien maantieteellisestä hajauttamisesta. Saatavuuden osalta todetaan, etteivät suurimmatkaan pilvipalveluntarjoajat vielä yllä korkean saatavuuden vaatimuksiin, ja siten lyhyetkin saatavuuskatkokset voivat muodostua ongelmaksi pilvipohjaisille SCADA-järjestelmille. Saatavuutta voidaan kuitenkin parantaa pilvessä maantieteellisellä hajautuksella, muttei sillä voida kuitenkaan poistaa saatavuuden menetyksen riskiä kokonaan. Tietototurvan puolesta pilviympäristön todetaan altistavan SCADA-järjestelmä uusille tietoturvariskeille ja SCADA-ohjelmistojen tietoturvan olevan riittämätön lisääntyneiden SCADA-järjestelmiin kohdistuvien hyökkäysten edessä. Yhteenvetona työssä todetaan, etteivät vaatimukset toteudu riittävällä tasolla kriittisten tuotantoprosessien SCADA-järjestelmille, mutta ovat riittävät esimerkiksi etävalvomon pilvipohjaiselle toteutukselle. Etuja pilvipohjaiselle SCADA-järjestelmälle työssä todetaan olevan muutamia. Pilvipohjainen toteutus voi esimerkiksi mahdollistaa järjestelmän nopeamman kehityksen, kustannussäästöjä, järjestelmän etäkäytön, vähentyneen ylläpidon tarpeen sekä pilvipalvelun ominaisuudet kuten helpon skaalautuvuuden.

Avainsanat: SCADA, pilvipalvelu, reaaliaikaisuus, saatavuus, tietoturva

ABSTRACT

Harri Lehtonen: Fulfilment of system and reliability requirements in a cloud-based SCADA system
Master of Science Thesis
Tampere University
Master's Degree Programme in Automation Engineering
February 2023

In the recent years, the usage of cloud services has increased in traditional IT and along with the industry 4.0, cloud services have become an integral part of industrial automation systems. The cloud integration of supervisory control and data acquisition systems used for monitoring and controlling the production process in industrial automation would enable the system to adapt the features of traditional IT, such as scalability and possibility for remote access. The requirements for an industrial system are stricter than in the traditional IT and integrating a SCADA system designed for local use in a private network to a cloud service makes it even more difficult to meet these requirements.

In this thesis, through literature research, the aim is to find out whether the most essential system and reliability requirements in terms of cloud integration of the SCADA system are fulfilled in a cloud-based implementation. Regarding the system requirements, the fulfilment of the real-time requirement is studied and from reliability requirements the system availability, the data security and cybersecurity requirements are researched. In addition, based on the literature, the advantages, and opportunities that a cloud-based SCADA system brings to the company are evaluated.

In the thesis, it is concluded that the delays introduced by communication to the cloud and processing in the cloud are small, and thus the real-time requirements are fulfilled sufficiently in normal networking conditions, satisfying the requirements for SCADA system's polling interval. Significant delays are also not caused by encryption of the communication channels or geographical distribution of servers. Regarding availability, it is concluded that even the largest cloud service providers do not yet meet the strict requirements of high availability, and therefore even short outages in cloud services can occur and be a problem for cloud-based SCADA systems. Although, availability can be improved by using geographical distribution it does not guarantee total availability. On behalf of information security and cybersecurity, it is found that the cloud environment exposes the SCADA system to new security threats and that the SCADA system's security features are insufficient in the face of increased attacks targeting SCADA systems. In summary, it is concluded that all the needed requirements for safe and correct operation of the system are not met at a sufficient level for SCADA systems of critical processes. However, it is found that they are sufficient for cloud-based remote monitoring systems. A few advantages to cloud-based SCADA systems are evaluated. Cloud-based implementation for example might enable faster development of the system, cost savings, remote use of the system, reduced need for maintenance and typical cloud service features such as scalability.

Keywords: SCADA, cloud, real-time, availability, cybersecurity

The originality of this thesis has been checked using the Turnitin OriginalityCheck service.

ALKUSANAT

Tämä diplomityö on tehty Insta Automation Oy:lle syksyn 2022 ja alkuvuoden 2023 aikana. Kiitos Installe mahdollisuudesta tehdä heille lopputyö ja Arttu Hanhelalle työn aihepiirin ideoinnista ja rajauksesta. Aihepiiri oli mielenkiintoinen ja työn aikana tuli opittua paljon uutta varsinkin pilvipalveluista. Lisäksi kiitos Henri Borgille neuvoista ja sparrauksesta työn alkuvaiheessa.

Tampereen yliopiston puolelle kiitos työn ohjaajille professori Matti Vilkolle ja apulaisprofessori David Hästbackalle. Heidän asiantuntevasta ja rakentavasta palautteesta tutkimuksen rakenteesta ja sisällöistä sekä ylipäätään aihepiirin tietämyksestä oli suuri apu työtä kirjoittaessa.

Tampereella, 28.2.2023

Harri Lehtonen

SISÄLLYSLUETTELO

KUVALUETTELO

LYHENTEET JA MERKINNÄT

APT	Advanced Persistent Threat
AWS	Amazon Web Services
CPS	Cyber-physical systems
DDoS	Distributed Denial of Service
ECC	Execution Control Chart
ERP	Enterprise Resource Planning
FaaS	Function as a Service
FB	Function Block
HIDS/HIPS	Host-based IDS/IPS
HMI	Human Machine Interface
HSB	Hot Stand By
IaaS	Infrastructure as a Service
IDS	Intrusion Detection System
IoT	Internet of Things
IPS	Intrusion Prevention System
ISA	The International Society of Automation
MES	Manufacturing Execution System
MITM	Man In The Middle
MTU	Master Terminal Unit
NIST	National Institute of Standards and Technology
NIDS/NIPS	Network-based IDS/IPS
PaaS	Platform as a Service
PID	Proportional-Integral-Derivative
PLC	Programmable Logic Controller
PLR	Process Level-Redundancy
QoS	Quality of Service
RTT	Round Trip Time
RTU	Remote Terminal Unit
SaaS	Software as a Service
SCADA	Supervisory Control and Data Acquisition
SOA	Service Oriented Architecture
UDDI	Universal Description, Definition, and Integration
WSDL	Web Services Description Language
XML	Extensible Markup Language

1. JOHDANTO

Teollisuuden neljäs vallankumous, digitalisaatio maailmalla sekä sen myötä kasvaneet datavirrat ovat viimeisen vuosikymmenen aikana vauhdittaneet pilvipalveluiden käyttöä erilaisten IT-sovellusten ja -palveluiden kehitys- sekä julkaisualustana. Pilvipalvelu tarkoittaa laskenta- ja tallennusresurssien vuokraamista palveluntarjoajalta omistamisen sijaan. Pilvipalveluntarjoaja tarjoaa internetin yli abstrakteja palveluita, joita käyttäjät voivat vuokrata omien tarpeidensa mukaan. Pilvipalveluiden tuomia etuja ovat esimerkiksi virtualisoidut resurssit, rinnakkainen prosessointi, pääsynhallinta sekä datapalveluiden integrointi skaalautuvaan datavarastoon. Palveluiden käyttö voi mahdollistaa kustannussäästöjä yritykselle IT-infrastruktuurin ylläpidossa, tehokkaamman järjestelmän hallinnan sekä käyttäjille helpon pääsyn dataan ja resursseihin internetin välityksellä [30].

Toisin kuin pilvipalveluita, käytönohjaus- ja valvontajärjestelmiä (engl. *SCADA, Supervisory Control And Data Acquisition)* on kehitetty jo useita vuosikymmeniä. SCADA-järjestelmät ovat teollisuus- ja tuotantoautomaatiojärjestelmän peruspilareita, ja niitä on käytössä niin kriittisen infrastruktuurin kohteissa kuten ydinvoimaloissa ja vesilaitoksilla, kuin myös vähemmän kriittisissä tuotantolaitoksissa. Nykyajan SCADA-järjestelmät sisältävät ohjelmoitavia logiikoita (engl. *PLC, Programmable Logic Controller*), käyttöliittymän prosessin hallintaan (engl. *HMI, Human Machine Interface*) sekä tietoliikennejärjestelmiä [68]. Nämä yhdessä muodostavat integroidun kokonaisuuden, jolla voidaan hallita ja monitoroida tuotantoprosesseja etänä.

Teollisuus- ja tuotantoautomaatiossa yleinen trendi on adaptoida muutoksia hitaasti perinteistä IT-teknologiasta ja siten seurata sitä perässä kehityksessä. Teollisuuden automaatiojärjestelmien elinkaaret ovat hyvin pitkiä ja sen seurauksena uusien teknologioiden päätyminen tuotantoon on hidasta. Lisäksi teollisuusautomaatiossa teknologian järjestelmävaatimukset ovat tiukempia kuin tavanomaisessa IT-teknologiassa ja toisin kuin perinteiset IT-järjestelmät, teollisuusjärjestelmät ovat suunniteltu käytettäviksi suljetuissa ja yksityisissä verkoissa. Ohjattavat prosessit voivat ongelmien ilmaantuessa aiheuttaa henkilö- tai ympäristövahinkoja tai johtaa yrityksen taloudellisiin tappioihin tuotannon seisoessa. Tämän takia SCADA-järjestelmät vaativat erityisesti luotettavuutta ja järjestelmävaatimusten täyttymistä sekä fyysisiltä laitteistoilta, että ohjelmistoilta ja tietoliikenteeltä.

Tutkimuslaitos Gartnerin mukaan pilvipalvelut ovat siirtyneet ”Slope of Enlightment” -vaiheeseen jo vuonna 2018 [53]. Tässä vaiheessa yleinen ymmärrys kyseisestä teknologiasta laajentuu, sen mahdollisuudet alkavat konkretisoitua ja toisen- tai kolmannen sukupolven tuotteita alkaa ilmestyä markkinoille. Pilvipohjaisten SCADA-järjestelmien osalta tämä on huomattavissa, sillä osa SCADA-järjestelmätoimittajista mahdollistaa jo ohjelmistojen integraation pilviympäristöön ja jotkin toimijat tarjoavat kokonaisia SCADA-järjestelmiä pilvipohjaisina ohjelmistopalveluina. Pilvipohjaisten SCADA-järjestelmien toiminta on kuitenkin vielä vähän tutkittua. SCADA-järjestelmän pilvi-integraatio tuo mukanaan täysin uudenlaisia haasteita esimerkiksi julkisen internetin käytön myötä ja siten SCADA-järjestelmän tiukkojen vaatimusten toteutuminen vaatii uudenlaisia ratkaisuita ja näkökulmia. Mikäli pilveen viedyssä järjestelmässä toteutuvat järjestelmä- ja luotettavuusvaatimukset riittävän hyvin ja järjestelmän oikeellinen käyttö olisi pilvessä turvallista, niin SCADA-järjestelmän vienti pilveen mahdollistaisi yritykselle pilvipalveluiden laajat hyödyt ja mahdollisesti lyhyemmän järjestelmän kehitysajan [31].

1.1 Tutkimuskysymykset

Tämän diplomityön päätavoite on tutkia pilvipohjaisen SCADA-järjestelmän kannalta oleellisimpien järjestelmä- ja luotettavuusvaatimusten toteutumista sekä niiden ohella pohtia pilvi-integraation tuomia etuja yrityksen ja suunnittelun näkökulmasta. Järjestelmävaatimuksista käsitellään reaaliaikaisuutta ja luotettavuusvaatimuksista järjestelmän saatavuutta sekä tieto- ja kyberturvallisuutta.

Työssä pyritään vastaamaan seuraaviin tutkimuskysymyksiin:

1. Toteutuvatko SCADA-järjestelmän järjestelmävaatimuksista reaaliaikaisuus ja luotettavuusvaatimuksista saatavuus sekä tieto- ja kyberturvallisuus riittävän hyvin pilvipohjaisen SCADA-järjestelmän mahdollistamiseksi?
2. Millaisia etuja ja mahdollisuuksia SCADA-järjestelmän pilvi-integraatio mahdollistaa yritykselle?

Työn kontekstissa pilviympäristöillä tarkoitetaan pääasiassa julkisia pilvipalveluita, joiksi tässä työssä on valittu Microsoft Azure ja Amazon Web Services (*AWS*). Vaikka tietoliikenneprotokollat ja tietorakenteet ovat hyvin oleellinen osa SCADA-järjestelmää, ei niiden toteutustapoihin oteta työssä kantaa, vaan keskitytään vain niiden vaikutuksiin järjestelmäarkkitehtuuriin ja vaatimusten toteutumisessa. Vaatimusten pohdinnassa oletuksena pidetään järjestelmää, jossa ohjauslogiikka sijaitsee edelleen kentällä ja pilveen on viety vain HMI, hälytykset ja data.

1.2 Tutkimusmenetelmä ja -aineisto

Työn tutkimusmenetelmänä on kirjallisuuskatsaus. Kirjallisuuskatsauksessa tiettyä ilmiötä tarkastellaan kokoamalla ja analysoimalla ilmiöön liittyvää kirjallisuutta [74]. Tutkimusaiheesta voidaan tarkastella eri näkökulmia, teorioita sekä tuloksia. Näkökulmia voi olla tutkimuksessa useita ja ne voivat olla eri teoriasuuntausten tai tutkimusmenetelmien näkökulmia samasta aiheesta [50]. Kirjallisuuskatsauksen tyyppi tässä työssä on kokoava, jossa tarkoituksena on esitellä aiempaa tutkimusta ja muodostaa sen pohjalta tutkimuskysymyksiin kokoavia johtopäätöksiä [32].

Työssä lähdekirjallisuutta etsitään Tampereen yliopiston kirjaston Andor-palvelusta, josta löytyy kootusti julkaisuja useista eri tietokannoista. Työn kannalta oleellisimmat tietokannat palvelussa ovat SpringerLink, IEEE Xplore, ScienceDirect sekä ProQuest Central. Lähdekirjallisuutta etsitään lisäksi myös suoraan edellä mainituista tietokannoista sekä Googlen tieteellisten julkaisuiden alustalta Google Scholarista. Hakusanoina tietokannoista työssä on pääasiassa käytetty eri englanninkielisiä kombinaatioita ja muunnelmia seuraavista sanoista: pilvi, pilvipohjainen, SCADA, IoT, IIoT, tietoturva, kyberturva, reaaliaikaisuus, vikasietoisuus, hajautettu, saatavuus, SOA, Iaas, PaaS, SCADA-as-a-Service, migraatio, legacy, haasteet, edut, hyödyt, MOM, AWS ja Azure. Lähdemateriaalina käytetään myös SCADA- ja pilvipalveluntarjoajien verkkosivustoja.

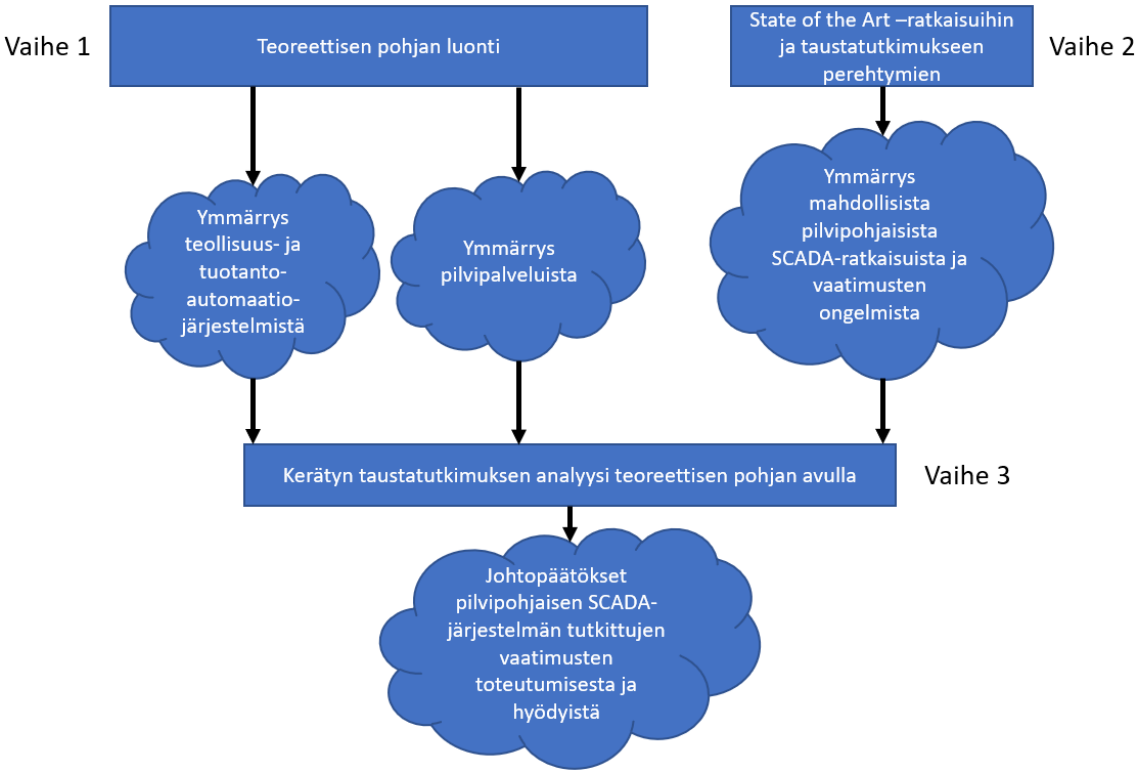
Pilvipalveluiden käytön yleistyessä ja teollisen neljännen vallankumouksen myötä pilvipalvelut ovat lähivuosina vakiinnuttaneet paikkaansa myös teollisuusautomaatiossa. Uudehkon luonteensa vuoksi SCADA-järjestelmän vienti pilveen on vielä vähän tutkittua ja siten lähdemateriaalia on saatavilla rajoitetusti. Työssä käytettävät vanhimmat tieteelliset artikkelit ovat julkaistu noin 10 vuotta sitten. Esimerkiksi hakulauseella ”cloud-based AND SCADA” löytyy yliopiston hakupalvelusta 277 osumaa, joista työn kontekstiin soveltuu alle 10 %. Rajallisesta lähdemateriaalin määrästä johtuen työssä pyritään käyttämään ja soveltamaan lähes kaikkien löydettyjen tutkimusten tuloksia työn kontekstiin soveltuvin osin. Tutkimuksia etsitään myös muilta samankaltaisia ominaisuuksia ja vaatimuksia omaavilta tutkimusaloilta kuten pilvirobotiikasta. Lähdemateriaalista pääosa on konferenssijulkaisuja ja artikkeleita. Artikkeleista pyritään käyttämään vain vertaisarvioituja.

Pilvipohjaisesta SCADA-järjestelmästä tehdyissä tutkimuksissa keskitytään usein natiiviin lähestymistapaan, jossa koko SCADA-järjestelmä rakennetaan uudelleen pilven ominaisuuksien hyödyntämistä varten [31]. Olemassa olevien SCADA-järjestelmien suo-

ran pilvimigraation tutkimus on vähäisempää. Tieteellisessä lähdemateriaalissa keskitytään myös usein vain jonkin tietyn järjestelmä- tai luotettavuusvaatimuksen toteuttamiseen, jolloin toteutus saattaa vaikuttaa muihin tutkittuihin vaatimuksiin.

1.3 Työn vaiheet ja rakenne

Työn tutkimus perustuu kokonaan kirjallisuuteen ja voidaan jakaa karkeasti kolmeen eri vaiheeseen: teoreettisen pohjan luonti, olemassa oleviin pilvipohjaisiin ratkaisuihin sekä tutkimukseen perehtyminen ja johtopäätösten tekeminen. Vaiheet ja niissä tuotettu tieto on esitetty kuvassa 1.1.



Kuva 1.1. **Työn tutkimuksen rakenne.**

Työn ensimmäisessä vaiheessa muodostetaan yleinen teoreettinen pohja teollisuus- ja tuotantoautomaatiojärjestelmistä sekä pilvipalveluista. Ensimmäisessä vaiheessa tuotettu sisältö esitellään luvuissa 2 ja 3. Luvussa 2 esitellään automaatiojärjestelmän vaatimuksia, järjestelmän eri tasot ja tarkemmin SCADA-järjestelmä. Lisäksi esitellään usein automaatiossa käytetyt tekniikat; virtualisointi ja redundanssi. Luvussa 3 puolestaan taustoitetaan pilvimallia käymällä läpi pilven edut ja pääpiirteet, pilven palvelutasot ja käyttöönottomallit sekä lopuksi julkiset pilvipalvelut AWS ja MS Azure. Lukujen tarkoitus on antaa lukijalle teoreettinen pohja SCADA-järjestelmän osista, vaatimuksista ja pilvimallin mahdollisuuksista.

Toisessa vaiheessa teoriapohjan luonnin jälkeen pyritään kirjallisuudesta löytämään aikaisempaa tutkimusta SCADA-järjestelmän viennistä pilveen ja muodostamaan sen pohjalta käsitys mahdollisista järjestelmäarkkitehtuureista ja vaatimusten täyttymisten ongelmista. Toisen vaiheen sisältöä käsitellään luvussa 4. Luvussa esitellään tutkimuksia pilvipohjaista SCADA-järjestelmistä pilven eri palvelutasoilla ja käyttöönottomalleilla. Lisäksi esitellään kaksi kaupallista State of the Art -ohjelmistoa. Esiteltyjä tutkimuksia käytetään viimeisessä vaiheessa omien päätelmien ja oman argumentoinnin tukena.

Viimeisessä vaiheessa analysoidaan löydettyä taustatutkimusta teoreettisen pohjan avulla, ja pohditaan ensimmäisessä tutkimuskysymyksessä esiteltyjen järjestelmä- ja luotettavuusvaatimusten toteutumista pilvipohjaisessa SCADA-järjestelmässä. Lisäksi vaiheessa pohditaan, millaisille järjestelmille vaatimukset toteutuvat riittävän hyvin sekä pilvi-integraation tuomia etuja ja mahdollisuuksia. Viimeisen vaiheen sisältö esitetään luvussa 5, jossa käydään ensin vaatimusten toteutumiset yksitellen läpi, jonka jälkeen lyhyesti pohditaan legacy-järjestelmäkomponenttien vaikutusta vaatimusten toteutumiseen, ja lopuksi esitetään mahdolliset pilvi-integraatiosta löydetyt hyödyt. Luvussa 6 on yhteenveto tutkimuksen tuloksista.

2. AUTOMAATIOJÄRJESTELMÄ

Automaatiojärjestelmässä on integroitu sensoreita, ohjauksia ja toimilaitteita yhdeksi kokonaisuudeksi, jonka tavoite on suorittaa tiettyä tehtävää ja minimoida tai poistaa ihmisen tarve prosessista. Automaatiojärjestelmä voi koostua vain yhdestä loogisesta elementistä tai muodostaa suuriakin kokonaisuuksia, kuten kokonaisen tehtaan ohjausjärjestelmä. Tämän työn kontekstissa automaatiojärjestelmällä tarkoitetaan teollisuusautomaatiojärjestelmiä, jotka voidaan jakaa prosessi- tai tuotantoautomaatiojärjestelmiin.

Automaatiojärjestelmät sopivat vaarallisiin tai toistuviin prosesseihin, ja parantavat prosessin johdonmukaisuutta, toistettavuutta, tarkkuutta ja nopeutta, jotka puolestaan parantavat lopputuotteen laatua. Automaatiojärjestelmät myös parantavat työskentelyoloja ja vähentävät operointikustannuksia. Toisaalta automaatiojärjestelmät vaativat yleensä suuria investointeja, lisäävät kunnossapitoa ja ovat huonoja mukautumaan, mikäli prosessia halutaan muokata erilaiseksi. [65]

Tämän luvun alussa käydään läpi teollisuusautomaatiojärjestelmän perusvaatimuksia: reaaliaikaisuus, saatavuus sekä tieto- ja kyberturvallisuus. Näiden jälkeen esitellään automaatiojärjestelmän laitetasot sekä ANSI/ISA-95 (IEC/ISO 62264) viitekehys ja sen osat. Lopuksi esitellään syvemmin SCADA-järjestelmä ja sen arkkitehtuurit.

2.1 Järjestelmävaatimukset

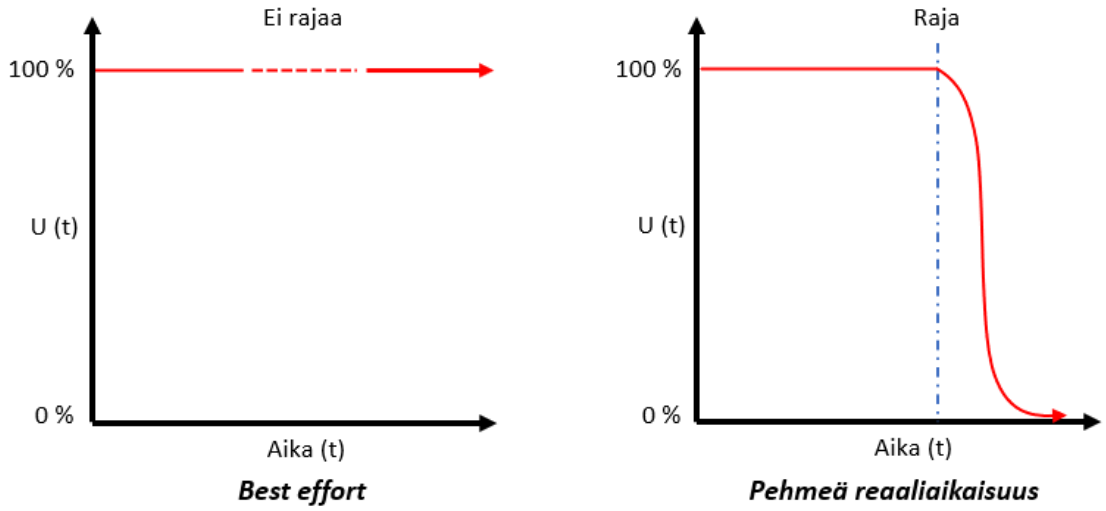
Teollisuusautomaatiojärjestelmillä on huomattavasti tiukemmat vaatimukset kuin tyypillisillä IT-järjestelmillä, jotka yleensä vain manipuloivat dataa ilman fyysisten laitteiden ohjausta. Mikäli järjestelmä ei pysty vastaamaan vaatimuksiin, voi se johtaa häiriöihin teollisuusautomaatiossa ja aiheuttaa vahinkoa järjestelmälle, luonnolle tai pahimmillaan ihmisille sekä aiheuttaa suuria taloudellisia menetyksiä tuotannon seisoessa. Kriittisissä prosesseissa kuten energiantuotannossa tai vedenjakelussa viat voivat vahinkojen lisäksi aiheuttaa ongelmia muille toimialoille ja yhteiskunnan toiminnalle. [40]

Seuraavien lukujen vaatimusten lisäksi automaatiojärjestelmällä on lisäksi useita muita sovelluskohteesta riippuvaisia vaatimuksia, kuten esimerkiksi järjestelmän skaalautuvuus, yhteensopivuus sekä ylläpidettävyys ja konfiguroitavuus. Esiteltävät vaatimukset ovat järjestelmän oikeellisen ja turvallisen toiminnan kannalta olennaisia. Edellä mainitut oheisvaatimukset puolestaan eivät ole järjestelmän toiminnan kannalta pakollisia, mutta asiakkaan ja toimittajan kannalta mahdollisesti tärkeitä vaatimuksia.

2.1.1 Reaaliaikaisuus ja determinismi

Aikariippuvuus tiedonsiirrossa on automaatiojärjestelmän perusvaatimuksen lisäksi myös yksi suunnittelun osa-alue. Ohjaussignaalien sekä mittausdatan tulee kulkeutua järjestelmien välillä tietyssä määritellyssä ajassa, jotta ohjaukset ovat oikea-aikaisia ja data relevanttia. Kommunikaation tulee myös pystyä priorisoimaan esimerkiksi hälytykset muiden toimintojen edelle. Näitä ominaisuuksia kutsutaan reaaliaikavaatimuksiksi. [80]

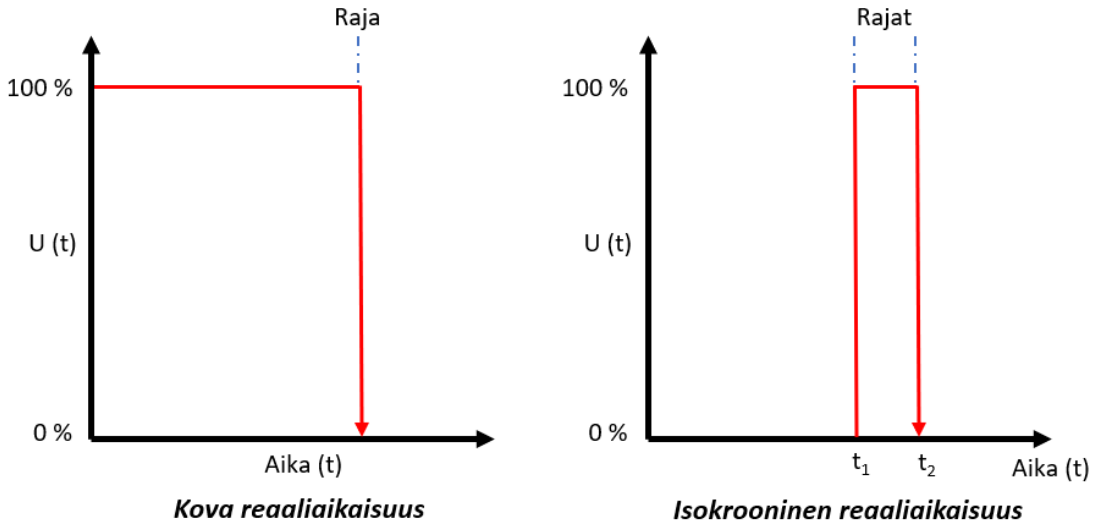
Reaaliaikaisuus määrittelee informaation hyödyllisyyden ajan funktiona. Reaaliaikavaatimukset voidaan jaotella pehmeisiin ja koviin vaatimuksiin. Pehmeät vaatimukset ovat tyypillisiä perinteisissä IT-ympäristöissä, ja kovat vaatimukset järjestelmänohjauksissa ja säädöissä [80]. Kuvassa 2.1 on esitetty pehmeän reaaliaikaisuuden vaihtoehdot.



Kuva 2.1. *Pehmeän reaaliaikavaatimuksen vaihtoehdot. Mukaillen [80].*

Reaaliaikavaatimuksista vapain määritelmä on ”Best-effort” -vaatimuksella, joka ei aseta informaation saapumiselle aikarajaa, jolloin sen hyödyllisyyteen ei vaikuta saapumisaika. Kuvan 2.1 oikeanpuoleisen pehmeän reaaliaikaisuuden tapauksessa on olemassa tietty aikaraja, jonka jälkeen informaation relevanssi pienenee aikariippuvaisen funktion mukaisesti. Pehmeiden aikavaatimusten mukaisten sovelluksien tiedonsiirrossa viiveet tai häiriöt eivät aiheuta ongelmaa järjestelmälle [80]. Tällaisia järjestelmiä automaatiomaailmassa ovat esimerkiksi SCADA-järjestelmien trendien piirtotyökalut, jotka hakevat historiadatan historiaserveriltä ilman reaaliaikavaatimuksia.

Kovat vaatimukset voidaan jaotella itse kovaan reaaliaikaisuuteen ja isokrooniseen reaaliaikaisuuteen. Nämä kaksi vaihtoehtoa on esitetty kuvassa 2.2.



Kuva 2.2. *Kovan reaaliaikavaatimuksen vaihtoehdot. Mukaillen [80].*

Kovassa reaaliaikaisuudessa informaation relevanssi katoaa aikarajan ylittyessä. Tällaisissa järjestelmissä tietoliikenteen viiveet ja viat aiheuttavat ongelmia. Isokroonisella reaaliaikaisuudella puolestaan on tietty aikaikkuna, jonka sisällä saapunut informaatio on relevanttia. Tämä hieman pehmentää viiveestä aiheutuvia ongelmia. Kova reaaliaikavaatimus on tyypillisesti hälytyksillä ja isokrooninen vaatimus ohjausjärjestelmillä [80].

Reaaliaikaisuuden saavuttamiseksi järjestelmän osien tulee olla deterministisiä ja ennakoitavia. Deterministisyys on mitta järjestelmän vasteajan vaihteluun tietyssä tapahtumassa. Teollisuuden automaatiojärjestelmän tulee kyetä tuottamaan esimerkiksi mittausdataa tarkoin aikavälein, jolloin sen maksimi vasteaika on ennustettavissa ja rajattu. [9]

2.1.2 Saatavuus ja luotettavuus

Yksi tärkeimmistä teollisuusautomaatiojärjestelmän ominaisuuksista on saatavuus. Saatavuus tarkoittaa, että tuotantolaitoksen on kyettävä pitämään valmistus tai tuotanto käynnissä minimaalisella häiriöajalla. Häiriöaika järjestelmässä voi aiheuttaa suuria taloudellisia menetyksiä yritykselle ja esimerkiksi SCADA:n tapauksessa saatavuuden menetys voi aiheuttaa vaaratilanteita, kun käynnissä olevaa prosessia ei pystytä monitoroimaan ja hallitsemaan käyttöliittymän kautta. Saatavuuden maksimoinniksi järjestelmän tulee olla myös luotettava. Nykypäivän DCS (Distributed Control System) -järjestelmä voi ääritapauksissa saavuttaa jopa 99.9999 % saatavuuden. [66][88]

Usein saatavuuskatkokset johtuvat jonkin järjestelmän komponentin vikaantumisesta. Isojen automaatiojärjestelmien koon kasvaessa vikojen ja vika-alttiiden komponenttien

määrä nousee. Suurissa järjestelmissä voi olla tuhansia erilaisia laskentaelementtejä ja fyysisiä toimilaitteita, jotka ovat alttiita erilaisille vioille. Vikoja ovat esimerkiksi laitteiden hajoamiset, kommunikaatioyhteyksien katkeamiset sekä muut normaalista toiminnasta poikkeavat tapahtumat. Järjestelmäviat ovat mahdottomia välttää ja ennustaa suurissa dynaamisissa järjestelmissä. Mikäli vikaa ei havaita ja siitä palauduta tarpeeksi nopeasti, voi se johtaa reaaliaikaisen järjestelmän häiriöön ja pahimmillaan pysäyttää koko järjestelmän toiminnan. [64]

Reaaliaikaisten hajautettujen järjestelmien tulee olla edelleen saavutettavissa laitteisto- ja ohjelmistovikojen aikana. Näihin vikoihin voidaan varautua käyttämällä vikasietoisuuden parantamiseen tarkoitettuja tekniikoita, jotka mahdollistavat käyttövarmuuden ja järjestelmän normaalin toiminnan vian ilmaantuessa [64]. Vikasietoisuus voidaan määritellä järjestelmän ominaisuutena, joka mahdollistaa järjestelmän normaalin toiminnan vian aikana [41].

Usean vian vikasietoisuuden toteutuksessa ongelmaksi muodostuvat reaaliaikaisen järjestelmän järjestelmävaatimusten säilyttäminen. Järjestelmän tulee vikasietoisuudesta huolimatta edelleen pystyä normaalitilassa vastaamaan reaaliaikavaatimuksiin, olla skaalautuva ja luotettava sekä säilyttää suorituskykynsä. Tärkeää on myös valita oikeanlainen tekniikka, sillä epäluotettava tai väärä tekniikka voi virheellisesti epäillä oikein toimivan prosessin toimintaa tai harhaisesti luottaa väärin jo vikaantuneeseen prosessiin. [24]

Teollisuusautomaatiossa vikasietoisuutta lisätään tyypillisesti redundanteilla fyysisillä ja ohjelmistopohjaisilla elementeillä sekä turvallisen vikaantumisen (engl. *fail-safe*) metodeja käytetään turvallisuuden lisäämiseen vikatilanteissa. Redundanttisuudessa samoja komponentteja on useampia ja yhden vikaantuessa voidaan ottaa käyttöön redundanttinen ehjä versio. Järjestelmien välisessä kommunikaatiossa on myös erilaisia mahdollisuuksia saavuttaa korkea luotettavuus, kuten pakettien uudelleenlähetys, virheenkorjaus koodit ja reitityskanavien redundanssi. Kuitenkin useat luotettavuutta lisäävät ratkaisut vaikuttavat negatiivisesti deterministisyyteen ja ennakoitavuuteen [66].

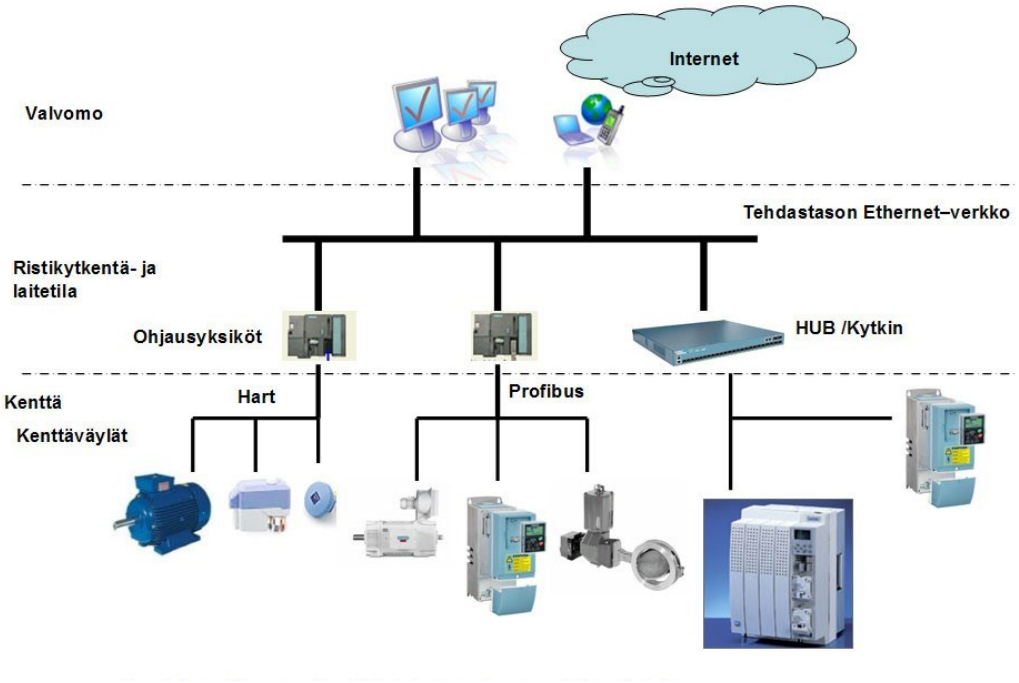
2.1.3 Tieto- ja kyberturvallisuus

Tieto- ja kyberturvallisuus ovat järjestelmän turvallisen ja luotettavan käytön kannalta tärkeäitä vaatimuksia. Tietoturva määritellään informaation luottamuksellisuuden, eheyden ja saatavuuden säilyttämisenä. Sen päätarkoitus on varmistaa liiketoimintaprosessien jatkuvuus uhkatilanteissa mahdollisimman pienin vahingoin ja pienentää niiden vaikutusta. Esimerkkejä tietoturvallisuudesta ovat pääsynhallinta, vaatimuksenmukaisuuksien hallinta sekä tekniset tarkastukset. Kyberturvallisuus on määritelmän mukaan yksi tietoturvallisuuden osa-alueista. Se tarkoittaa informaation turvaamista kyberuhkilta ja -hyökkäyksiltä, kun sitä prosessoidaan, siirretään tai tallennetaan. Kyberturvallisuuden esimerkkejä ovat ohjelmistojen, verkkoyhteyksien ja pilvipalveluiden käyttöturvallisuudet. [89][5]

Järjestelmän turvallisuuden merkitys on korostunut viime vuosina merkittävästi SCADA-järjestelmiin kohdistuneiden verkkohyökkäyksien määrän noustessa [26]. Esimerkkejä tunnetuista hyökkäyksistä ja järjestelmän turvallisuuden tarpeellisuudesta löytyy lähimenneisyydestä; Ukrainassa vuonna 2015 käytetty ”Black Energy 3” on palvelunesto-hyökkäykseen ja datan tuhoamiseen suunnattu troijalainen ja vuonna 2014 laajalle levinnyt SCADA-järjestelmiä vastaan suunnattu toinen troijalainen ”Havex” ovat hyviä syitä pohtia järjestelmän tieto- ja kyberturvallisuutta. Fyysisiä laitteita ohjaavan järjestelmän tieto- ja kyberturvallisuus on myös suoraan sidottu itse järjestelmän fyysiseen turvallisuuteen. [69][42]

2.2 Automaatiojärjestelmän laitetasot

Automaatiojärjestelmät voidaan jakaa hierarkkisiin tasoihin perustuen laitteiden toimintoihin. Kuvassa 2.3 on esitetty automaatiojärjestelmän hierarkia perustuen laitteiden toiminnallisiin ominaisuuksiin. Tyypilliset tasot näin jaotellulle automaatiojärjestelmälle ovat kenttä, ristikytkentä- ja laitetila sekä valvomo. Kuvassa 2.3 mittausdataa kerätään kenttätasolta ja se kulkee kenttäväyliä pitkin ylöspäin ohjauslaitteiden kautta valvomotasolle. Valvomotasolta puolestaan ohjausdata virtaa toiseen suuntaan ristikytkentä- ja laitetilatason ohjausyksiköille, jotka toimintalogiikan pohjalta ohjaavat toimilaitteita.



Kuva 2.3. *Automaatiojärjestelmän tasot perustuen laitteiden toimintoihin [7].*

Moderneissa automaatiojärjestelmissä kuvan 2.3 esittämä hierarkia ei välttämättä toteudu. Automaatiojärjestelmissä esiintyy yhä useammin esineiden internetin (engl. *IoT, Internet of Things*) mahdollistavia älykkäitä laitteita, jotka pystyvät kommunikoimaan keskenään ja tasojen yli. Nämä IoT-laitteet muodostavat kyberfyysisiä järjestelmiä (engl. *CPS, cyber-physical systems*), joissa fyysinen ja virtuaalinen toimintaympäristö yhdistyvät ja siten järjestelmän hierarkiatasot ovat häilyviä. [25]

Seuraavissa alaluvuissa käsitellään kenttätaso sekä ristikytkentä- ja laitetilataso. Valvomotaso käsitellään myöhemmässä SCADA-järjestelmää käsittelevässä luvussa 2.5 laajemmin kuin edellä mainitut tasot.

2.2.1 Kenttätaso

Kenttätaso on hierarkiatasoista alin ja sijaitsee lähinnä tuotantoprosessia. Kenttälaitteet ovat elektronisia laitteita, kuten antureita tai toimilaitteita, jotka ovat suoraan fyysisessä yhteydessä prosessiin. Anturit keräävät prosessista dataa ja lähettävät sitä ylöspäin seuraavan tason ohjausyksiköille kenttäväyliä pitkin. Toimilaitteet puolestaan ovat laitteita, jotka tekevät fyysisiä toimintoja prosessissa, kuten venttiilejä tai moottoreita. [25]

Anturit ja toimilaitteet operoivat tavallisesti < 1 s sykleissä ja suurissa prosesseissa niitä voi olla tuhansia, jolloin dataa liikkuu tasojen välillä suuria määriä. Datan määrän vuoksi teollisuusautomaatiossa kenttätason ja ylemmän tason välissä käytetään kenttäväyliä, jotka ovat suunniteltu nimenomaan suurien datamäärien tehokkaaseen tiedonsiirtoon.

2.2.2 Ristikytkentä- ja laitetilataso

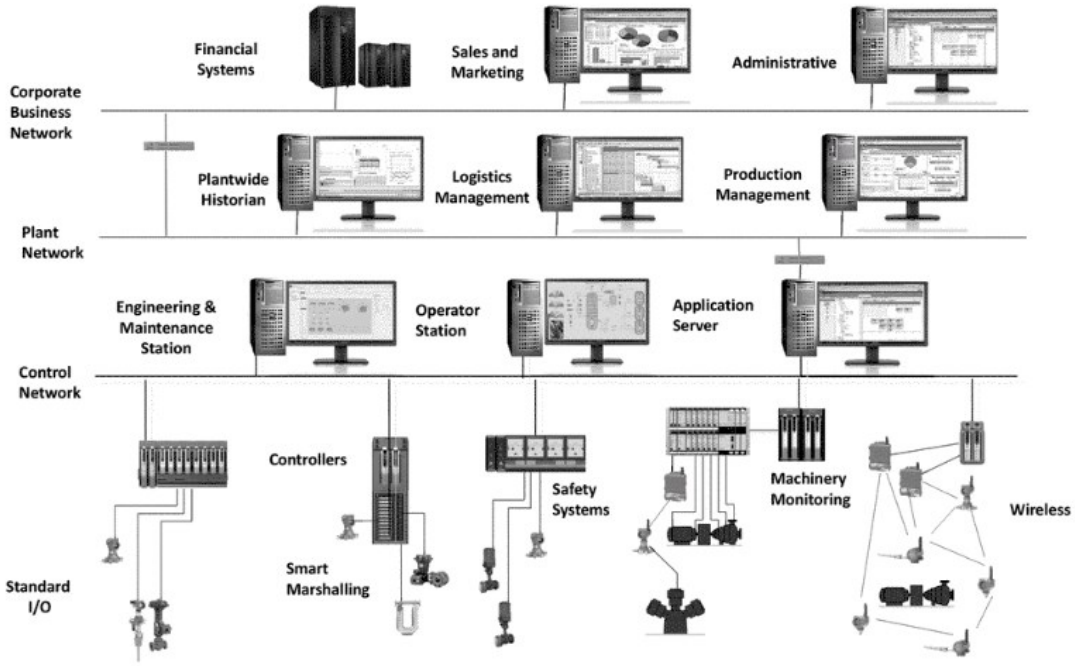
Ristikytkentä- ja laitetilatasolla tapahtuu itse prosessin ohjaaminen. Tälle tasolle siis sijoittuvat teollisuusautomaatiossa usein käytetyt ohjelmoitavat logiikat sekä PID (*Proportional-Integral-Derivative*) -säätimet. Tason tehtävä on kerätä antureilta kenttäväylien välityksellä mittausdataa, ja sen avulla muodostaa kenttälaitteille lähetettävät ohjaukset ohjelmoitavalle logiikalle ladatun ohjelman mukaisesti. Toisinaan, mikäli prosessi on liian monimutkainen ohjelmoitaville logiikoille, voidaan järjestelmä korvata saman tason hajautetulla ohjausjärjestelmällä (engl. *DCS, Distributed Control System*) [77].

PLC-yksiköt ovat laajalti käytettyjä komponentteja automaatiojärjestelmissä. Ne ovat digitaalisia tietokoneita, joita voidaan käyttää sähkömekaanisten prosessien hallinnassa. PLC-yksiköt eroavat tavallisesta tietokoneesta ominaisuuksiltaan; PLC-yksiköllä on liityntämahdollisuudet useiden kenttälaitteiden liitäntään, se kestää lämpötilan vaihteluita, sähköisiä häiriöitä sekä tärinää. Ne voivat käsitellä sekä analogisia, että digitaalisia tuloja ja lähtöjä sekä ovat deterministisiä ja reaaliaikaisia. [65]

2.2.3 DCS-järjestelmä

DCS-järjestelmiä käytetään paljon energialaitoksissa sekä jatkuva-aikaisissa prosesseissa, mutta harvoin diskreeteissä tuotantoprosesseissa. Hajautetut ohjausjärjestelmät tarjoavat laskentatehoa, liitettävyyttä ja infrastruktuurin, joilla linkittää anturit, toimilaitteet, prosessin ohjauksen sekä monitoroinnin. [77]

DCS-järjestelmään kuuluu yleensä prosessiasemia, valvonta-asemia, järjestelmäväylä, ohjelmoitavia logiikoita sekä mahdollinen tiedonhallinta/raportointiasema. Hajautettu arkkitehtuuri DCS-järjestelmässä tarkoittaa, että useat alajärjestelmän ohjaukset ja kommunikointi on hajautettu eri fyysisille laitteille. Reaaliaikainen prosessinohjauksen tietokanta on hajautettu viemällä prosessiasemat lähelle itse prosessia, jossa ne pystyvät hoitamaan mittaustiedon käsittelyn, ohjausten laskennan ja suorittamisen ilman keskustietokonetta. Koko hajautettu laitteiden järjestelmä on linkitetty toisiinsa kenttäväylän avulla, jonka kautta toimivat kommunikointi, koordinointi ja monitorointi. Kuvassa 2.4 on esitetty eräs esimerkki DCS-järjestelmän komponenteista ja niiden linkityksestä. [7][77]

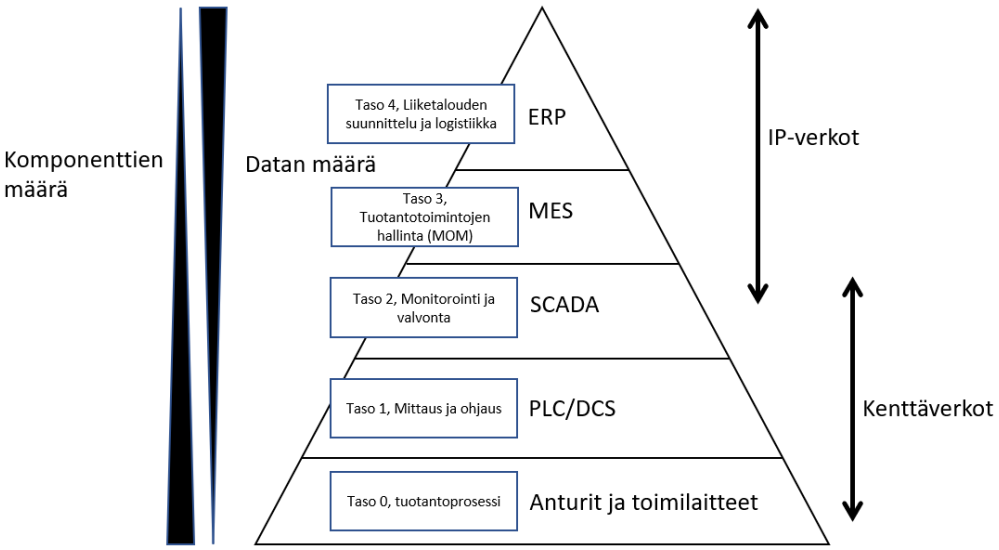


Kuva 2.4. *Modernin DCS-järjestelmän komponentteja [77].*

2.3 ANSI/ISA-95 (IEC/ISO 62264) viitekehys ja tasot

ANSI/ISA-95 on kansainvälisesti hyväksytty standardi yrityksen ja ohjausjärjestelmän integraatioon. Sen on kehittänyt ja sitä ylläpitää The International Society of Automation (*ISA*), joka on kansainvälinen teollisuusautomaatioon ja instrumentointiin keskittynyt yhteisö.

Standardin käyttö voi hyödyttää yritystä antamalla sille koko yrityksen laajuista perspektiiviä järjestelmäintegraatiosta, jonka avulla yritys voi pyrkiä muodostamaan ymmärrettävän viitekehyksen tuhansien eri toimintojen ja datapisteiden ympärille. Standardin tarkoitus on määritellä ja integroida sekä toiminnot yrityksen ja ERP-järjestelmän (engl. *Enterprise Resource Planning*) välillä, että toiminnot MES-järjestelmän (engl. *Manufacturing Execution System)*, tuotantotoimintojen hallinnan sekä toiminnanohjauksen välillä. ISA-95 mallien avulla voidaan päätellä, miten informaation tulisi kulkea myynti-, talous- ja logistiikkajärjestelmien välillä sekä tuotanto-, huolto- ja laatujärjestelmien välillä. [55]



Kuva 2.5. *ANSI/ISA-95 standardin määrittelemät tasot.*

Kuvassa 2.5 on ANSI/ISA-95 standardin määrittelemät järjestelmätasot ja standardi tunnetaan tavallisemmin nimellä automaatiopyramidi. Automaatiopyramidi laajentaa luvun 2.2 kuvausta järjestelmästä yli laitetasojen ja jakaa teollisen yrityksen viiteen hierarkkiseen tasoon. Pyramidin kolme alinta tasoa ovat käytännössä samat kuin luvun 2.2. Tuotantoprosessi (taso 0) vastaa kenttätasoa, mittaus ja ohjaus (taso 1) vastaa ristikytkentä- ja laitetasoa sekä monitorointi ja valvonta (taso 2) vastaa valvomotasoa.

ISA-95 standardi esittelee luvun 2.2 toimintapohjaisen mallin päälle kaksi ylempää tasoa. Tasolla 3 on tuotantotoimintojen ohjaus (engl. *MOM, Manufacturing Operations Management)* ja tasolla 4 on liiketalouden suunnittelu ja logistiikka.

Ylemmillä tasoilla datan määrä kasvaa ja jalostuu alempien tasojen datasta, mutta se ei ole enää yhtä aikakriittistä kuin alemmilla tasoilla. Pyramidin ylimmällä tasolla datan keräys voi tapahtua viikkojen tai jopa kuukausien aikaikkunoilla. Järjestelmäkomponenttien määrä myös putoaa pyramidin tasojen myötä. Alimmat tasot käyttävät kenttäverkkoja ja ylimmät IP-verkkoja. [38]

On hyvä huomata, että ISA-95 on hyvin periaatteellinen lähestymistapa, ja siksi soveltuu parhaiten suuriin järjestelmiin, jotka ovat jokseenkin standardisoituja. Standardin soveltamisesta kaikkiin järjestelmiin on todennäköisesti enemmän haittaa kuin hyötyä, mutta se toimii kuitenkin hyvänä suunnittelun lähtökohtana teollisuusautomaatiojärjestelmää suunniteltaessa. [62]

2.4 Tuotantotoimintojen ohjaus ja ERP

Tuotantotoimintojen ohjausjärjestelmät ottavat kantaa kriittisiin tuotannon toimintoihin, kuten laatuun, turvallisuuteen, luetettavuuteen, tehokkuuteen ja säännösten noudattamiseen. ISA-95 määrittelee MOM-järjestelmälle seuraavat ominaisuudet:

- Tuotantotoimintojen hallinta
- Huoltotoimintojen hallinta
- Laadun hallinta
- Materiaalin käsittely ja varastointi
- Tukitoiminnot, mukaan lukien turvallisuuden, tietojen, konfiguroinnin, dokumentaation, säännöstenmukaisuuden ja tapausten/poikkeamien hallinta.

Nykypäivän MOM-järjestelmät mahdollistavat yrityksille prosessien standardoinnin ja optimoinnin, minimoivat läpimenoaikoja, nopeuttavat markkinoille tuloa sekä parantavat tuotannon näkyvyyttä. [55]

ISA-95 viitekehyksessä MES-järjestelmät ovat MOM-järjestelmien osajoukko. MES-järjestelmä on sidoksissa tuotantoprosessin tapahtumiin ja siihen on sisällytetty joukko prosessinhallinnan työkaluja. MES-ohjelmistojen päätavoite on monitoroida, hallita ja optimoida päivittäisiä prosesseja datan avulla. [70]

ERP-järjestelmät ovat hallinnan laajuudeltaan vielä suurempia kuin MES- ja MOM-järjestelmät. MES- ja MOM-järjestelmät monitoroivat ja hallinnoivat yhtä tuotantolaitosta, kun taas ERP-järjestelmät tarjoavat monitorointia, raportointia ja hallintaa koko konsernin ja automaatiopyramidin laajuudelta. ERP-järjestelmät sisältävät dataa esimerkiksi asiakkaista, toimittajista, sopimuksista, kirjanpidosta ja projektinhallinnasta.

2.5 SCADA

SCADA-järjestelmä löytyy ISA-95:n automaatiopyramidin tasolta 2, ja se määrittelee tason toiminnoiksi fyysisen prosessin monitoroinnin ja kontrolloinnin [55]. Yleisesti SCADA-järjestelmä mielletään luvun 2.2 laitetasojen valvomotasoksi, mutta sen toimintoihin kuitenkin liittyvät vahvasti myös alemmat tasot ja varsinkin automaatiopyramidin 1. tasolla sijaitsevat ohjauslaitteet. Tästä johtuen ISA-95:n standardin ulkopuolella tasojen realistiset rajat ovat häilyvät.

Termiä SCADA-järjestelmä käytetään usein järjestelmissä, joissa hallittava prosessi on maantieteellisesti laajalle levinnyt, kuten energiantuotantoprosessit. SCADA-järjestelmä voidaan määritellä kokoelmaksi laitteita, jotka tarjoavat prosessista etänä olevalle operaattorille tarpeeksi informaatiota päätellä laitteiden tai prosessin tila sekä mahdollisuuden vaikuttaa prosessiin tai laitteisiin ilman, että operaattori on fyysisesti paikalla [90]. SCADA-järjestelmä siis kerää dataa hajautetuilta kenttälaitteilta, siirtää sen eteenpäin ja lopuksi esittää sen operaattorille. Operaattori monitoroi reaaliaikaisesti HMI-käyttöliittymästä tätä loogisesti organisoitua prosessidataa, ja voi halutessaan muuttaa käyttöliittymän kautta prosessin parametreja ja ohjauksia. SCADA-järjestelmä myös tallentaa dataa, esittää trendejä datasta ja hoitaa hälytys- ja varoituskäsittelyn.

2.5.1 SCADA-järjestelmän elementit

SCADA-järjestelmä koostuu yleensä useasta alajärjestelmästä, jotka voivat olla yksinkertaisia tai hyvinkin monimutkaisia. Tyypillinen SCADA-järjestelmä koostuu seuraavista osista: pääpääteyksikkö (engl. *MTU, Master Terminal Unit*), etäpääteyksikkö (engl. *RTU, Remote Terminal Unit)*, kenttälaitteet, HMI, PLC-yksiköt, kommunikaatiomedia ja historiaserveri. [37]

MTU

Pääpääteyksikkö kerää, prosessoi ja tallentaa dataa useilta eri etäpääteyksiköiltä. MTU myös tarjoaa operaattorille käyttöliittymän informaation esittämiseen ja etäasemien hallinointiin. Monimutkaisissa järjestelmissä voi olla MTU- ja RTU-yksiköiden välissä alaasemia, jotka keräävät dataa usean prosessiaseman RTU-yksiköiltä ja lähettävät sen eteenpäin järjestelmän MTU:lle. Kulunutta aikaa siitä, kun MTU aloittaa datan keruun tietyltä RTU-yksiköltä ja palaa tälle samalle RTU:lle uudestaan kutsutaan kiertokyselyväliksi. [37]

Kenttälaitteet (RTU, PLC, IED)

Kenttälaitteet sisältävät luvun 2.2.1 kenttätason ja luvun 2.2.2 ristikytkentä- ja laitetason laitteita ja niiden toiminnallisuuksia. Näiden laitteiden pääasiallinen tarkoitus on kerätä prosessista dataa, muokata se MTU:n ymmärtämään muotoon sekä lähettää ohjaussignaaleja toimilaitteille. Vaikka RTU, PLC ja IED (engl. *Intelligent Electronic Device*) ovat samankaltaisia laitteita, on niillä kaikilla erilaiset toiminnot. RTU kerää dataa ja lähettää sen MTU:lle, joka tekee päätökset ohjauksista, ja lähettää ne takaisin RTU:lle, joka sitten käskyttää toimilaitteita. PLC-yksiköt puolestaan omaavat RTU:n ominaisuudet, mutta pystyvät myös lokaaliin datan prosessointiin ilman MTU:ta. IED:t puolestaan ovat osa

laitteiden sisäisiä ohjausjärjestelmiä kuten muuntajia ja sulakkeita, joita voidaan ohjata PLC-yksiköllä tai RTU:lla. [2]

HMI

HMI tarjoaa operaattorille tehokkaan käyttöliittymän prosessin hallintaan ja monitorointiin. HMI esittää yleensä graafisesti prosessin PI-kaaviota mukaillen prosessin laitteiden, antureiden ja ohjausparametrien tilat sekä mahdollistaa operaattorille prosessin hallinnan. Näytöllä esitetään myös yleensä hälytykset, varoitukset ja muut tärkeät ilmoitukset. [2]

Historiaserveri

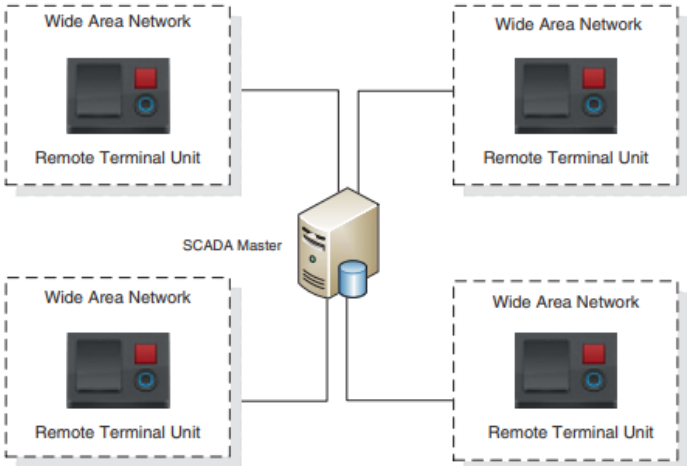
Historiaserveri on tietokanta, jonne tallennetaan kaikki oleellinen järjestelmästä kerätty aikaleimattu data, kuten mittaukset, ohjaukset ja hälytykset. Tätä dataa voidaan myöhemmin käyttää analysointiin, vian etsintään tai trendien esittämiseen HMI:ssä. [2][37]

2.5.2 SCADA-järjestelmän arkkitehtuurit

SCADA-teknologia on ollut olemassa jo yli 50 vuotta, mutta se on vuosien saatossa pysynyt kehityksessä uusien teknologioiden perässä. Jatkuvasti on tarve saada älykkäämpiä, turvallisempia ja tehokkaampia järjestelmiä. SCADA-järjestelmien arkkitehtuurit ovatkin kehittyneet vuosien saatossa monoliittisestä järjestelmästä IoT-pohjaisiin SCADA-järjestelmiin, jotka nojautuvat pilvipalveluiden käyttöön.

Ensimmäinen sukupolvi: Monoliittinen

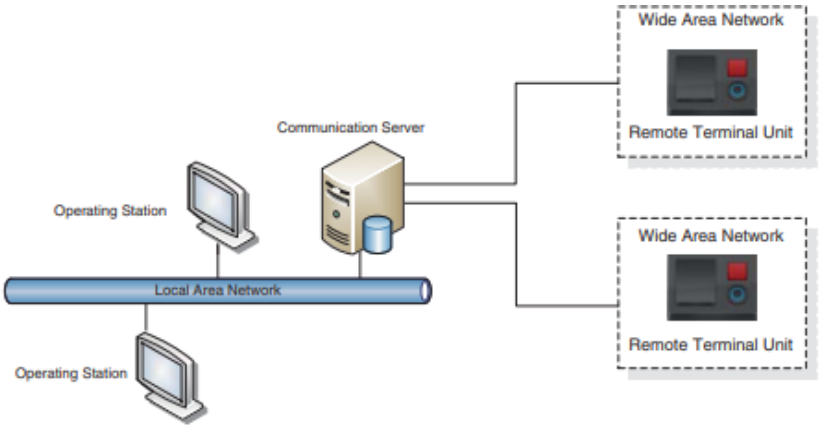
Ensimmäisen sukupolven SCADA-järjestelmät ovat kehitetty aikana, jolloin konseptia tietoverkoista ei käytännössä ollut, ja SCADA-järjestelmät rakennettiin itsenäisiksi ohjausjärjestelmiksi ilman yhteyksiä toisiin ohjausjärjestelmiin. Kommunikaatioprotokollat olivat laitevalmistajien omia ja ne kykenivät skannaamiseen, ohjaukseen ja datan vaihtoon sekä MTU:n ja RTU:n välillä, että RTU:N ja kenttälaitteiden välillä. Tyypillisesti MTU:n ja RTU:n välillä käytettiin laajaverkkoja (engl. *WAN, Wide Area Network*), joka ei kuitenkaan vielä tuolloin ollut yhtä edistyksellinen kuin nykyään. Monoliittisissa järjestelmissä on käytössä järjestelmävikaan varautumiseen kaksi identtistä järjestelmää, joista toinen on ensisijainen ja toinen vian varalle kopioitu. Kuvassa 2.6 on esitetty kyseisen SCADA-järjestelmän arkkitehtuuri. [2][37]



Kuva 2.6. *Monoliittisen SCADA-järjestelmän arkkitehtuuri [2].*

Toinen sukupolvi: Hajautetut järjestelmät

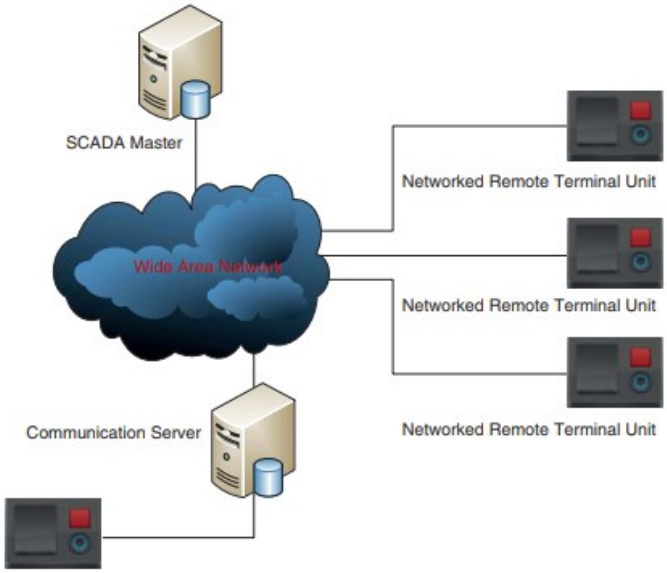
Lähiverkkoteknologian (engl. *LAN, Local Area Network*) kehittyessä kehittyivät myös toisen sukupolven SCADA-järjestelmät. Tämän sukupolven SCADA hajauttaa prosessoinnin usealle asemalle, joista jokaisella on oma roolinsa. Nämä eri asemat ovat yhdistetty LAN-verkolla ja pystyvät sen kautta jakamaan informaatiota reaaliajassa. Tällaisessa järjestelmässä jokin asema toimii MTU:na, toinen historiaserverinä ja kolmas esimerkiksi HMI:nä. Järjestelmätoimintojen hajauttaminen parantaa järjestelmän prosessointitehoa sekä redundanttisuutta, ja siten järjestelmän luotettavuutta. Tässä järjestelmässä järjestelmävian ilmaantuessa voidaan vikaantunut asema korvata LAN-verkossa olevalla toisella asemalla. Toisen sukupolven järjestelmät ovat kuitenkin edelleen riippuvaisia laitevalmistajien omista protokollista eivätkä siksi pysty kommunikoimaan järjestelmän ulkopuolisiin laitteisiin. Kuvassa 2.7 on esitetty hajautetun SCADA-järjestelmän arkkitehtuuri. [2][37]



Kuva 2.7. *Hajautetun SCADA-järjestelmän arkkitehtuuri [2].*

Kolmas sukupolvi: Verkotettu SCADA

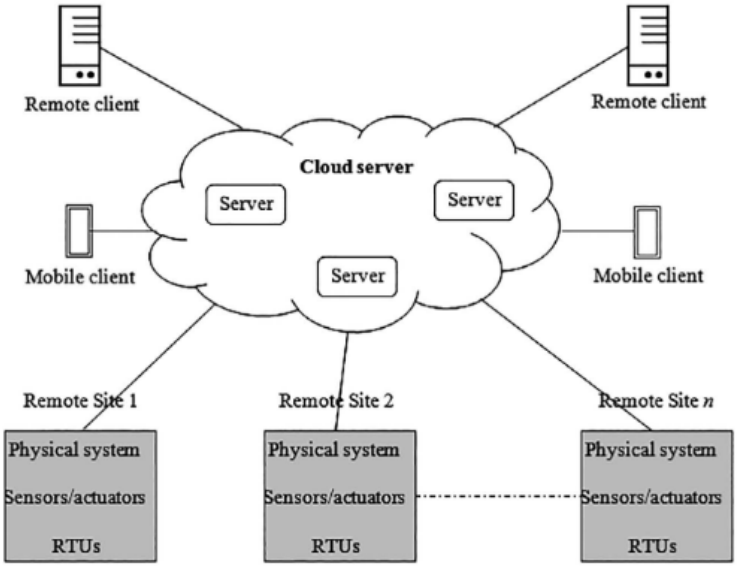
Kolmannen sukupolven SCADA-järjestelmän kehitystä nopeutti automatisoitujen prosessinhallintajärjestelmien kysynnän kasvu. Lisäksi tarve avoimille järjestelmille ilman laitevalmistajakeskeisyyttä on olennaisessa roolissa kolmannen sukupolven järjestelmissä. Suurin ero toiseen sukupolveen on mahdollisuus käyttää avoimien standardien protokollia ja siten mahdollistaa useiden eri valmistajien laitteiden käytön yhden sijaan. Avoimet protokollat mahdollistavat myös hajauttamisen WAN-verkkojen yli pelkän LAN-verkon sijaan. Tämä mahdollistaa SCADA-järjestelmän käytön internetin yli ja siten mahdollisuuden hallita prosessia paikkariippumattomasti. Kuvassa 2.8 on esitetty kolmannen sukupolven järjestelmän arkkitehtuuri. [2][37]



Kuva 2.8. *Verkotetun SCADA-järjestelmän arkkitehtuuri [2].*

Neljäs sukupolvi: IoT-pohjainen SCADA

IoT-innovaation ja taloudellisesti kannattavien pilvipalveluiden myötä SCADA-järjestelmiin on omaksuttu IoT-teknologiaa vähentämään infrastruktuuri- ja käyttöönottokustannuksia sekä parantamaan tehokkuutta, joustavuutta, optimointia, saatavuutta ja skaalautuvuutta. Pilvipalvelut ovat keskeisessä roolissa IoT-pohjaisessa SCADA-järjestelmässä, mahdollistaen etäservereiden linkittämisen pilveen ja siten keskitetyn datan varastoinnin. Pilveen talletettua dataa voidaan jatkojalostaa. Kuvassa 2.9 esitetty IoT-SCADA-arkkitehtuuri. [37][101]



Kuva 2.9. *IoT-pohjaisen SCADA:n arkkitehtuuri [37].*

2.6 Palvelukeskeinen arkkitehtuuri (SOA)

Ohjelmistojen uudelleenkäytettävyys ja niiden tehokkuus ovat muodostuneet tärkeiksi tutkimuskohteiksi teollisuusautomaatiossa [96]. Palvelukeskeinen arkkitehtuuri (engl. *SOA, Service Oriented Architecture*) on ohjelmiston suunnitteluun tarkoitettu malli, joka perustuu erillisiin pieniin ohjelmistomoduuleihin, jotka tarjoavat sovelluspalveluita suuremmille ohjelmistoille [36]. Sovelluskomponentit ovat yhteydessä keskenään erilaisilla viesteillä. Tämä SOA:n mahdollistama sovelluskomponenttien irrallinen yhdistäminen varmistaa yhteentoimivuuden erilaisten sovellusalustojen välillä riippumatta niiden laitteistoista tai ohjelmistoarkkitehtuureista [35]. SOA:n tarjoama suunnittelumalli on yhteensopiva hajautettujen järjestelmien kanssa modulaarisuuden ja kommunikaation puolesta, jolloin sitä on mahdollista soveltaa myös SCADA-järjestelmille [96][103]

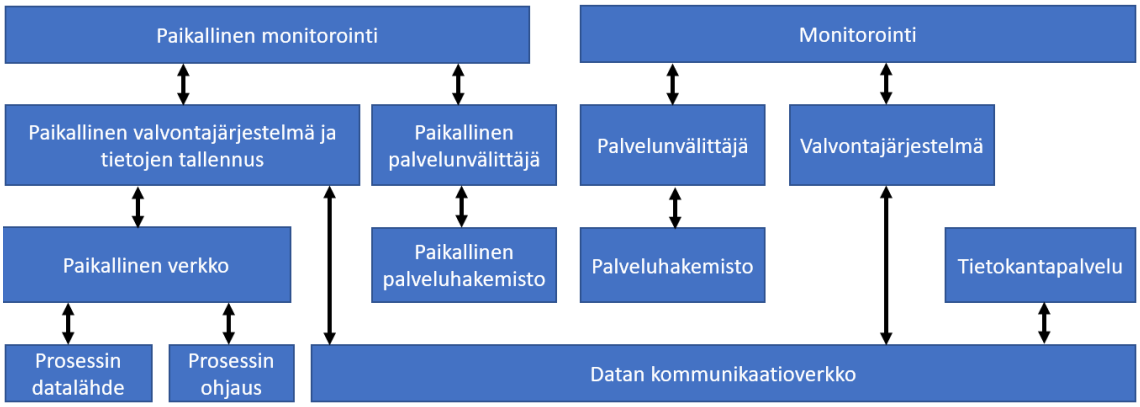
Palvelukeskeisellä arkkitehtuurilla on neljä pääasiallista tunnusmerkkiä [39]:

1. SOA-mallin palveluilla on XML (engl. *Extensible Markup Language*) -pohjaiset itsekuvaavat käyttöliittymät ja palvelut kuvataan WSDL (engl. *Web Services Description Language*) -standardilla.
2. SOA-mallin palvelut kommunikoivat viesteillä, jotka ovat määritelty XML-kuvauksessa. Kommunikaatio komponenttien välillä on mahdollista toteuttaa heterogeenisissä ympäristöissä ilman tarkkaa tietoa itse palvelun toteutuksesta.

3. SOA-mallin palveluita ylläpitää järjestelmässä hakemistolistauksena toimiva rekisteri. Sovellukset etsivät ja kutsuvat rekisterin kautta palveluita. Yleisin palvelurekisterissä käytetty standardi on UDDI (engl. *Universal Description, Definition, and Integration*)
4. Jokaiseen SOA-mallin palveluun on liitetty palvelun laatuvaatimukset (engl. *Quality of Service, QoS*), joissa elementteinä voivat olla esimerkiksi todennukset, valtuutukset, luotettavan viestintä ja käytännöt palvelun käytöstä.

Palvelut voidaan jakaa ydin- ja yhdistelmäpalveluihin. Ydinpalvelut ovat hyvin määriteltyjä ja riippumattomia toisten palveluiden tiloista. Yhdistelmäpalvelut puolestaan ovat yhdistelmiä ydinpalveluista tai toisista yhdistelmäpalveluista, ja ovat riippuvaisia oman toiminnan mahdollistavien palveluiden tiloista. Palveluiden kommunikointi toteutetaan usein verkkopalveluilla (engl. *Web Services*) kuten REST ja HTTP. Verkkopalvelut mahdollistavat prosessien ja laitteiden välisen kommunikaation [67].

SCADA-järjestelmiä on olemassa monia erilaisia ja on käytännössä mahdotonta kehittää universaalia ratkaisua, joka sopisi kaikille SCADA-järjestelmille. Eräs geneerinen ratkaisu on esitetty artikkelissa [43], jossa ideana on tunnistaa SCADA-järjestelmän kaikki toiminnallisuudet ja toteuttaa ne SOA-mallin palveluina. Tällä tavalla muodostetun SOA-malli pohjaisen SCADA:n järjestelmäarkkitehtuuri on esitetty kuvassa 2.10.



Kuva 2.10. *SOA-mallin SCADA-järjestelmäarkkitehtuuri. Mukaillen [43].*

Kuvan 2.10 mukaan paikalliselle tasolle ehdotetaan paikallista palveluhakemistoa. Palveluhakemistosta löytyvät paikalliset palvelut ja tieto siitä, miten niihin päästään käsiksi sekä samat tiedot keskitetyille palveluille. Uuden ohjaus- tai monitorointikomponentin toteutuksessa sen tiedot lisätään paikalliseen palveluhakemistoon. Paikallinen valvontajärjestelmä suunnitellaan datan vaihtoon prosessin datalähteen ja ohjauksien kanssa. Saatu data tallennetaan hetkellisesti paikallisesti ja myöhemmin tietokantapalvelu tallentaa datan tietokantaan, joka voi olla keskitetty tai hajautettu. Paikallinen monitorointi voi

käyttää paikallisen valvontajärjestelmän palveluita, mikäli ne tiedetään ennalta. Jos niitä ei tiedetä ennalta, niin ne tulee ensin hakea paikallisesta palvelinhakemistosta palvelunvälittäjän avulla. [43]

SOA-mallin järjestelmä on toiminnaltaan joustava, sillä järjestelmän palveluiden toimintaa voidaan muuttaa muokkaamalla jo olemassa olevien palveluiden toimintaa tai lisäämällä uusia toimintoja järjestelmään ilman sen toiminnan keskeytystä. Tämä lähestymistapa myös mahdollistaa palveluiden uudelleenkäytettävyyden muissa järjestelmissä, joko suoraan samanlaisina palveluina tai tarpeen mukaan muokattuina.

2.7 Virtualisointi

Virtualisoinnilla tarkoitetaan fyysisten resurssien korvaamista virtuaalisilla replikoilla. Yleinen virtualisoinnissa käytetty tekniikka on serverin virtualisointi, jossa tavallisesti käytetään hypervisor-ohjelmistokerrosta, joka mahdollistaa usean virtuaalisen vieraskoneen käytön isäntäkoneen fyysisellä laitteistolla. Virtualisointia käytetään jo laajasti tavanomaisessa IT-maailmassa, ja sen käyttö on viime vuosien aikana levinnyt myös automaatiomaailmaan. Virtualisoinnilla on useita hyötyjä kuten käyttöönoton helppous, tehokkaampi isäntäkoneen resurssien hyödyntäminen sekä komponenttien eristyksestä johtuva turvallisuuden parantuminen. [46]

Virtualisointitekniikat voidaan jakaa virtuaalikoneisiin ja kontteihin. Hypervisor on laitteiston virtualisointiin tarkoitettu ohjelmisto, jota voidaan käyttää suoraan ohjelmistokerroksena isäntäkoneen käyttöjärjestelmän päällä tai mahdollisesti suoraan laitteiston sisällä. Hypervisorin päällä voidaan ajaa rinnakkain useita virtuaalisia vieraskoneita, jotka voivat käyttää eri käyttöjärjestelmiä ja eri määrän fyysisiä resursseja. Konttitekniikka puolestaan viittaa käyttöjärjestelmätason virtualisointiin. Kontit mahdollistavat samankaltaisia etuja kuin virtuaalikoneet, mutta käyttävät vähemmän resursseja ja tuottavat paremman suorituskyvyn. Kontit eivät kuitenkaan tarjoa mahdollisuutta käyttää muuta kuin Linux-käyttöjärjestelmää vierasjärjestelmänä sekä ovat monimutkaisempia kuin virtuaalikoneet. [46]

Teollisuusautomaation saralla virtualisoinnista tekee kiinnostavan sen ominaisuuksien käyttö redundanssissa ja vikasietoisuudessa sekä taloudelliset hyödyt. Vikasietoisuudessa redundanttisia ohjelmistokomponentteja voidaan viedä eri virtuaalikoneille, jolloin ei ole tarvetta hankkia fyysistä IT-infrastruktuuria redundanssia varten. Tässä riskinä kuitenkin itse isäntäkoneen vikaantuminen tai hajoaminen, jolloin redundanttisista komponenteista ei ole hyötyä. Useat hypervisor ohjelmistot kuten esimerkiksi Hyper-V,

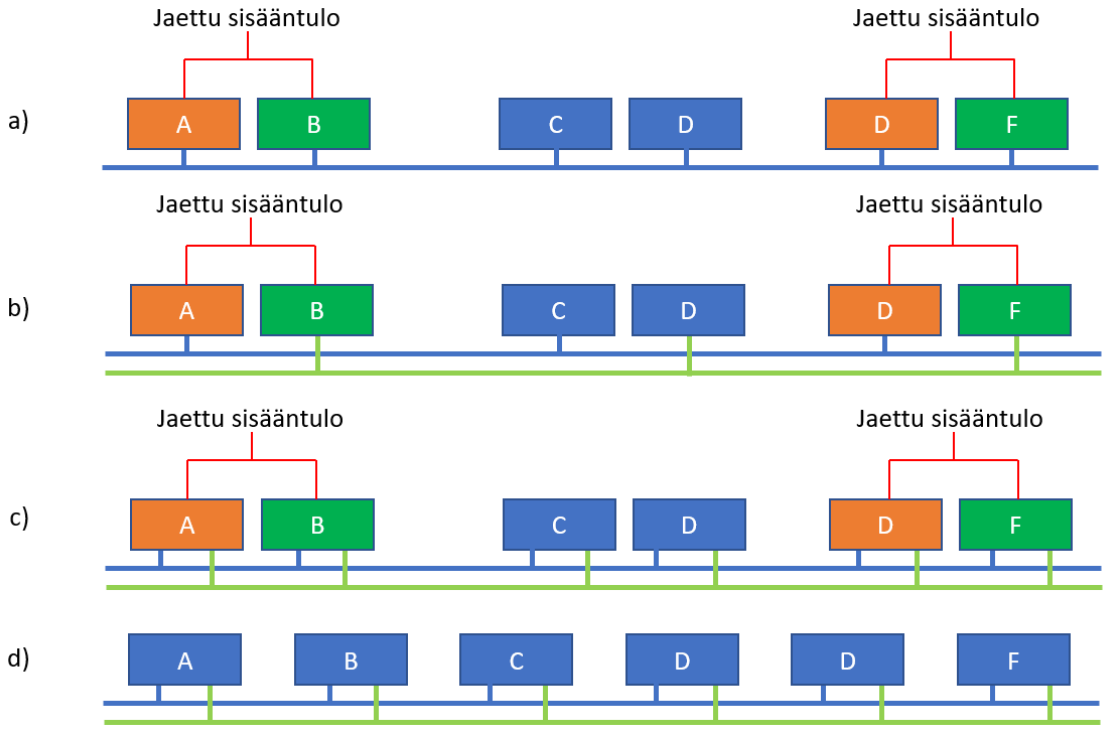
VMware ESXI ja WMware Workstation 12 Pro tarjoavat itsessään hyödyllisiä ominaisuuksia. Edellä mainitut ohjelmistot tarjoavat esimerkiksi mahdollisuuden ajonaikaiseen siirtoon, jossa ajossa oleva virtuaalikone voidaan siirtää fyysiseltä koneelta toiselle, tallennus ja palautus toiminnon, joka mahdollistaa virtuaalikoneen tilannekuvien kopioinnin sekä oman vikasietoisuuden. Vikasietoisuus toimii eri tavalla riippuen ohjelmistosta. Ohjelmisto voi esimerkiksi pitää yllä redundanttista virtuaalikonetta ottamalla pääkoneesta tilannekuvia ja päivittämällä redundanttisen koneen tilaa niiden mukaiseksi [100].

2.8 Redundanssi

Ohjelmistoon perustuvien redundanttisten komponenttien lisäksi lähes aina teollisuuden automaatiojärjestelmissä jotkin fyysiset komponentit ovat vähintäänkin kahdennettuja. Teollisuusautomaatiossa vika mielletään usein jonkin komponentin pysyväksi kokonaisvaltaiseksi vikaantumiseksi. Tällaisessa tilanteessa komponentti lopettaa toimintansa kokonaan, muttei kuitenkaan aiheuta vahinkoa muulle järjestelmälle. Tällaisen vian ilmaantuessa on mahdollisuus käyttää redundanttista komponenttia, jolle vikaantuneen laitteen toiminnot siirretään tietyn ajan kuluttua. Vaatimuksena redundanssille on vian havaitseminen, tilan replikointi ja ohjauksen siirto toiselle laitteelle.

Redundanssin aste ja arkkitehtuurin valinta riippuvat useasta tekijästä. Viasta palautumisajan täytyy olla pienempi kuin järjestelmän armonaika, jonka jälkeen järjestelmä ottaa käyttöön fyysisiä turvamekanismeja, kuten ylipaineventtiilejä tai jarruja. Prosessiautomaatiossa armonaika on tavallisesti < 2 s ja kriittisten kohteiden kovan reaaliajan sovelluksissa jopa < 2 ms. Lisäksi suunniteltaessa pitää päättää redundanssin aste; Kuinka monta ja minkä tyyppistä laitetta voi vikaantua siten, että toiminnot säilyvät. Redundanttisia komponentteja voidaan myös vähentää, mikäli osittainen järjestelmän vikaantuminen on sallittua ja vikaantunut osa on eristetty muusta järjestelmästä. [61]

Fyysinen redundanssi voidaan jakaa laitteiston ja verkkoyhteyksien redundanssiin. Seuraavassa kuvassa 2.11 on neljä eri tavalla redundanttista laitearkkitehtuuria. Tietoverkot ovat esitetty sinisillä ja vihreillä viivoilla, ja piirretty yksinkertaisuuden vuoksi väyliksi. Tässä kontekstissa laitteiden monistuksilla tavoitellaan funktionaalista redundanssia.



Kuva 2.11. *Redundanssiarkkitehtuureja funktionaaliselle redundanssille. Mukaillen [61].*

Kuvan 2.11 a) kohdassa sekä laitteet A ja B, että D ja F ovat keskenään redundanttiset ja jakavat saman sisääntulon. Tällaisia arkkitehtuureja, joissa laitteita on monistettu, mutta tietoverkkoa ei, käytetään usein turvajärjestelmissä. Kohdassa b) ohjausjärjestelmä on monistettu, joka takaa järjestelmälle korkean saatavuuden. Erotetussa tietoverkossa järjestelmä toimii oletuksella, että molemmat ohjausjärjestelmät eivät vikaannu samaan aikaan. Kohdan c) arkkitehtuurissa kaikki laitteet on yhdistetty molempiin tietoverkkoihin, jolloin saavutetaan vielä edeltävää kohtaa parempi järjestelmän saatavuus. Kohdassa d) vain tietoverkko on monistettu. Jokaisen laitteen kohdalla voidaan redundanssia tietoverkkoon lisätä millä tahansa kommunikaatiopinon tasolla. [61]

Järjestelmän jatkuva operointi vaatii funktionaalista redundanssia. Redundanssi metodit voidaan jakaa kahteen eri tekniikkaan, jotka ovat samat sekä laitteistoille, että verkkoyhteyksille. Tekniikat ovat dynaaminen ja staattinen redundanssi, joita yleisesti kutsutaan valmiustilan (engl. *standby*) ja työtilan (engl. *workby*) redundansseiksi. Valmiustilan redundanssissa redundanttiset laitteet ovat joko toimettomia tai suorittavat joitain muita funktionaaliseen redundanssiin liittymättömiä tehtäviä tarpeen mukaan. Valmiustilan tasot voidaan jakaa kylmään, kuumaan ja lämpimään. Kylmässä tasossa redundanttinen laite ei suorita mitään toimintoja ja on hitain kolmesta ohjausvaihdon tapahtuessa. Kuumassa tasossa redundanttinen komponentti ottaa toiminnan operoinnin haltuun vian ilmaantuessa saumattomasti ja lämpimässä lähes saumattomasti. Valinnan

tekee vaihtologiikka, joka päättää käytettävät laitteet ja verkkoyhteydet. Dynaamisen redundanssin hyötyjä ovat jaettu redundanssi, jolloin sama laite voi toimia usean eri komponentin redundanttisena parina sekä sillä voidaan parantaa redundanssin vikasietoisuutta, mikäli valmiustilan komponentti on toimettomassa tilassa. Huonoja puolia dynaamiselle redundanssille ovat sen hitaus vaihdon tapahtuessa, redundanssi ei ole jatkuva-aikaista, jolloin voi ilmetä piileviä vikoja sekä komponenttien välille vaaditaan synkronointi. Redundanttista järjestelmää, jossa on yksi päälaite ja sille yksi varalaite kutsutaan yksi-kahdesta (engl. *1oo2, one-out-of-two*) malliksi. 1oo2-malli on esimerkki MooN-mallista, jossa N komponentin joukosta on M kappaletta aktiivisia komponentteja. [56][61]

Staattisessa redundanssissa kaikki redundanttiset komponentit osallistuvat tehtävän toteuttamiseen rinnakkaisella toiminnalla. Kaikki komponentit ovat aktiivisia ja tuottavat ulostuloja ja siten voivat parantaa saatavuutta ja järjestelmän eheyttä. Logiikka valitsee luotetut kommunikaatioyhteydet ja komponentit perustuen ulostuloihin pohjautuvaan äänestykseen. Mikäli havaitaan jonkin laitteen vikaantuminen, voidaan se eristää pois redundanttisesta järjestelmästä. Staattisen redundanssin etuja ovat vaihdon tapahtuminen sulavasti ja nopeasti, jatkuva-aikainen redundanssi parantaa virheen havainnointia sekä komponenttien keskinäisellä vertailulla pystytään havaitsemaan vääränlaista toimintaa. Toisaalta myös staattinen redundanssi vaatii synkronointia ja on kalliimpi, koska samaa laitetta ei voida jakaa dynaamisesti usean komponentin kesken. [56][61]

3. PILVIMALLI

Pilvimalli, toiselta niemeltään pilvilaskenta (engl. *Cloud Computing*), on viimeisen vuosikymmenen aikana muodostunut keskeiseksi osaksi useaa eri tekniikan alaa. Pilvipalveluita käytetään esimerkiksi tiedostojen tallentamiseen, massadatan analysointiin, varmuuskopiontiin ja arkistointiin sekä ohjelmistojen kehitykseen ja testaukseen. Pilvimallissa palveluntarjoaja tarjoaa internetin välityksellä palveluna tarpeenmukaisesti saatavilla olevia IT-resursseja, kuten tallennustilaa ja laskentatehoa.

Pilvipalveluiden määritelmä on vuosien saatossa muuttunut useasti. Yhdysvaltalaisen NIST:n (*National Institute of Standard and Technology*) formaali määritelmä pilvipalveluille on edelleen laajasti käytetty sekä hyväksytty, ja määrittelee pilvilaskennan seuraavasti [92]:

”Pilvilaskenta on malli, joka mahdollistaa kaikkialta saatavilla olevan, kätevän ja tarpeenmukaisen verkkoyhteyden jaettuun joukkoon konfiguroitavia laskentaresursseja (esim. verkot, palvelimet, tallennustila, sovellukset ja palvelut), jotka voidaan nopeasti valmistella käyttöön minimaalisella hallintatyöllä ja ilman kanssakäymistä palveluntarjoajan kanssa.”

NIST:n määritelmä koostuu viidestä eri pääpiirteestä, kolmesta eri palvelutasosta ja neljästä eri käyttöönottomallista. Seuraavissa alaluvuissa käsitellään näitä eri määritelmän osia sekä pilvipalveluiden pääpiriteitä ja etuja.

3.1 Pääpiirteet ja edut

Sovelluksen ja sen ympärille rakennetun palvelun täytyy omata tietyt piirteet ennen sen määrittelyä aidoksi pilvimalliksi. NIST:n määritelmän mukaan pilvimallissa on viisi pääpiirrettä. Mikäli palvelusta puuttuu jokin näistä viidestä piirteestä, ei kyseessä ole pilvilaskentaa tarjoava palvelu [27][75]. NIST määrittelee pääpiirteet seuraavasti [92]:

1. *Tarpeenmukainen itsepalveluperiaate.* Palvelun käyttäjä pystyy tarpeen mukaan itsenäisesti ottamaan käyttöön pilviresursseja, kuten palvelinaikaa ja tallennustilaa, ilman vuorovaikutusta palveluntarjoajan kanssa.
2. *Kaikkialta saavutettava ja päätelaite riippumaton.* Palvelun tulee olla saavutettavissa tavallisella internetyhteydellä. Lisäksi palvelun ei tule vaatia erillistä asiakasohjelmaa tai asiakasohjelman on oltava hyvin kevyt, eikä se saa vaatia suuria

määriä tiedonsiirtoa palvelun ja itsensä välillä. Palvelun tulee myös olla saavutettavissa useilta erilaisilta päätelaitteilta, kuten esimerkiksi puhelimilta, kannettavilta tietokoneilta ja tableteilta. [75]

3. *Resurssien yhdistäminen ja jakaminen.* Palveluntarjoajan resurssit ovat yhdistetty palvelemaan useaa käyttäjää samanaikaisesti jakamalla fyysisisiä ja virtuaalisia resursseja dynaamisesti käyttäjien tarpeiden mukaan. Käyttäjällä ei lähtökohtaisesti ole tarkkaa tietoa missä tarjotut resurssit sijaitsevat. Palveluntarjoaja voi kuitenkin mahdollistaa käyttäjälle korkeamman abstraktiotason sijainnin määrittämisen, kuten maantieteellisen sijainnin tiettyyn maahan ja datakeskukseen.
4. *Palveluiden nopea joustavuus tarpeen muuttuessa.* Palvelun tulee pystyä nopeasti ja joustavasti mukautumaan käyttäjän resurssitarpeiden muutoksiin. Mikäli käyttäjä vaatii lisää resursseja, pitää ne pystyä tarjoamaan sekä vastaavasti tarpeettomat resurssit tulee voida vapauttaa ja siirtää toisille käyttäjille. Käyttäjän näkökulmasta käytettävissä olevat resurssit näyttävät loputtomilta ja voidaan ottaa käyttöön milloin tahansa.
5. *Käytön monitorointi.* Palvelun täytyy tarjota tapa mitata käyttöastetta. Tämä mittari voi olla esimerkiksi käytetty tallennustila, prosessointi, kaistanleveys tai aktiivisten käyttäjien määrä. Palvelu käyttää tätä tietoa resurssienhallintaan ja optimointiin. Käyttäjä puolestaan maksaa palvelusta perustuen resurssien käyttöön, jolloin monitoroinnin, hallinnan ja raportoinnin tulee olla läpinäkyviä molemmille osapuolille.

Pilvipalvelut ovat monipuolisia järjestelmiä ja mahdollistavat monia erilaisia käyttötarkoituksia. Lisäksi niitä saatavilla monia erilaisia ja riippuen käyttäjän tavoitteista sekä motivaatiosta, voi pilvipalveluiden käyttö mahdollistaa erilaisia hyötyjä. Yleisesti voidaan listata pilvipalveluiden käytölle seuraavanlaisia etuja [75]:

- *Luotettavuus.* Luotettavuuden rakentaminen tavalliseen järjestelmään voi olla kallista, sillä se vaatii yleensä useita samankaltaisia järjestelmiä tai mahdollisesti jopa useita eri palvelinkeskuksia. Useat pilvipalveluntarjoajat mahdollistavat järjestelmän maantieteellisten hajautuksen, jonka avulla voidaan varmistaa saatavuus häiriötilanteissa.
- *Skaalautuvuus.* Pilviympäristö vastaa automaattisesti käyttäjän tarpeita. Ympäristöön voidaan tarvittaessa lisätä resursseja ja vastaavasti vapauttaa niitä tarpeen pienentyessä.

- *Kustannussäästöt.* IT-infrastruktuurin ylläpidon ja omistamisen jäädessä palveluntarjoajan vastuulle siirtyvät myös niistä aiheutuneet kustannukset pois yrityksestä.
- *Suorituskyky.* Palvelun suorituskykyä monitoroidaan jatkuvasti, ja mikäli se putoaa tietyn tason alapuolelle, järjestelmä voi automaattisesti lisätä resursseja järjestelmälle. Palvelu ei kuitenkaan välttämättä takaa suorituskykyä, mutta mahdollisista suorituskyvyn alenemisesta tarjotaan korvauksia.
- *Ylläpidon helppous.* Pilvipalveluita käytettäessä infrastruktuurin ja järjestelmän ylläpito on palveluntarjoajan vastuulla. Palveluntarjoaja vastaa laitteistojen ja ohjelmistojen ajantasaisuudesta ja päivittämisestä. Palveluntarjoaja pystyy myös päivittämään resursseja ilman käyttökatkoa asiakkaan järjestelmään.
- *Turvallisuus ja määräystenmukaisuus.* Pilvipalveluiden tietoturvan ajatellaan olevan parempaa kuin perinteisen IT-ympäristön. Järjestelmän tieto- ja kyberturvallisuutta ylläpitävät tahot voivat keskittyä tiettyihin turvallisuuden ominaisuuksiin tai datatyyppeihin. Tieto- ja kyberturvaongelmia ratkaistaessa ongelmia ratkotaan useille eri asiakkaille organisaation sijaan, jolloin todennäköisesti ratkaisuun käytetään enemmän aikaa ja rahaa. Pilvipalveluiden käyttö voi mahdollisesti helpottaa myös määräystenmukaisuuden toteutumista.

3.2 Käyttöönottomallit

Pilvipalveluiden käyttötarkoitus vaihtelee organisaatioiden välillä. Organisaatioilla on omat vaatimuksensa, mitä palveluita halutaan käyttää ja kuinka hallittava ympäristön tulee olla. Näihin eri tarpeisiin vastataan erilaisilla käyttöönottomalleilla, jotka määrittelevät tavat, kuinka pilvipalvelu voidaan ottaa käyttöön [27]. NIST määrittelee neljä käyttöönottomallia [92]:

1. *Julkinen pilvi.* Pilvipalvelu on rakennettu julkiseen avoimeen käyttöön. Mallin mukaisen pilven voi omistaa ja sitä voi hallinnoida yksityisyritys, akateeminen- tai julkisorganisaatio tai jokin näiden yhdistelmä. Pilvijärjestelmä ja sen resurssit sijaitsevat ulkoisen palveluntarjoajan tiloissa, ja vastuu hallinnasta ja ylläpidosta on palveluntarjoajalla. Julkinen pilvi on käytännössä aina saavutettavissa internetin yli.
2. *Yksityinen pilvi.* Yksityisen pilven mallissa pilvipalvelu hankitaan organisaation sisäiseen käyttöön, jossa sitä voi käyttää suljetut käyttäjäryhmät. Yksityisen pilven voi omistaa ja sitä voi hallinnoida organisaatio, kolmas osapuoli tai näiden

yhdistelmä. Pilvijärjestelmä ja resurssit, joilla itse pilvipalvelu toteutetaan sijaitsevat pilveä käyttävän yrityksen tai organisaation tiloissa. Pilvipalvelun hallinnointi- ja ylläpitovastuu ovat organisaation vastuulla. Käyttöyhteys yksityiseen pilveen on yleensä WAN- tai LAN-yhteys. Etäkäyttäjille yhteys voidaan toteuttaa VPN (engl. *Virtual Private Network*) -yhteytenä internetin yli.

3. *Yhteisöllinen pilvi.* Yhteisöllinen pilvi on laajempi versio yksityisestä pilvestä. Sen käyttäjäkuntana on suljettu joukko käyttäjiä organisaatioista, jotka jakavat samankaltaisia tarpeita ja intressejä. Pilvimallin käyttäjät haluavat julkista pilvimallia enemmän yksityisyyttä, mutta samalla haluavat jakaa pilven ylläpidon vastuun muiden organisaatioiden kanssa. Yhteisöllisen pilven voi siis omistaa ja sitä hallinnoida yksi tai useampi yhteisön organisaatio, kolmas osapuoli tai jälleen jokin näiden yhdistelmä. [76]
4. *Hybridipilvi.* Hybridimallissa pilvi-infrastruktuuri koostuu kahdesta tai useammasta mallista (julkinen, yksityinen tai yhteisöllinen), jotka pysyvät omina kokonaisuuksinaan ja niitä yhdistää standardoidut tai valmistajakohtaiset teknologiat, jotka mahdollistavat datan ja sovellusten tietoliikenteen pilvien välillä. Hybridimalli monimutkaistaa käyttöympäristöä, mutta lisää järjestelmän joustavuutta [75].

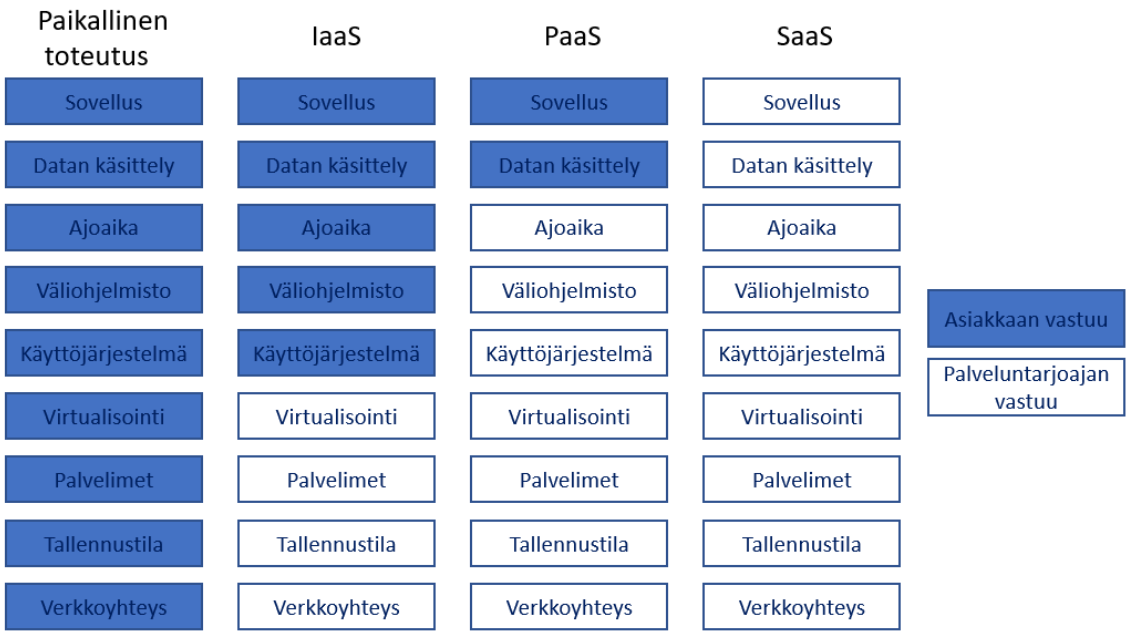
3.3 Palvelutasot

Pilvipalveluiden yleistyessä edellisen alaluvun käyttöönottomallien lisäksi on kehitetty eri palvelutasoja vastaamaan käyttäjien vaatimuksiin. Käyttöönottomallit ja palvelutasot tarjoavat käyttäjälle vaihtoehtoisia tasoja järjestelmän hallintaan, joustavuuteen ja ylläpitoon.

Tietotekniikassa abstraktiolla tarkoitetaan sovelluksen taustalla olevien teknologioiden ja järjestelmän elementtien piilottamista loppukäyttäjältä. Loppukäyttäjän näkökulmasta ei ole aina olennaista tietää miten tai millaisilla komponenteilla järjestelmä toimii, kunhan käyttäjälle on määritelty selkeä rajapinta järjestelmän käyttöön. Tämä voi olla järjestelmän päälle rakennetun sovelluksen suunnittelun kannalta hankalaa, sillä osa käytetyistä teknologioista lukitaan jo järjestelmän abstraktiotason valinnalla. Toisaalta samalla valinnalla osa järjestelmän ylläpitovastuusta voidaan ulkoistaa. [58]

Pilvipalvelut ovat loppukäyttäjän näkökulmasta abstraktioita, joiden eri tasoja kutsutaan yhteisnimellä palvelutasot. NIST määrittelee kolme eri palvelutasoa [92]: IaaS (*Infrastructure as a Service*), PaaS (*Platform as a Service*) ja SaaS (*Software as a Service*) [90]. Pilvipalveluissa palvelutason valinnalla voidaan vaikuttaa abstraktioon. Kaviksen

[58] mukaan on erityisen tärkeää, että pilviarkkitehtuuria suunnittelevat henkilöt ymmärtävät eri palvelutasojen ja käyttöönottomallien erot ja ominaisuudet. Ymmärtämättömyyden seurauksena voi olla, että menetetään edut, joita yrityksen pilvisiirtymällä on lähdetty tavoittelemaan. [58]



Kuva 3.1. *Eri palvelutasojen vastuualueet. Mukaillen [48].*

Kuvassa 3.1 on esitetty pilviarkkitehtuuriin perustuvan sovelluksen teknologiapino sekä miten eri palvelutasoilla pinon osa-alueiden vastuut jakautuvat. Sarakkeiden kolme alinta riviä sisältävät pinon fyysiset kerrokset. Fyysiseen kerrokseen kuuluvat verkkoyhteydet, tallennustila sekä fyysisten palvelimien laskentakapasiteetin mahdollistavat prosessorit ja muistit. Fyysiselle kerrokselle voidaan kuvitella myös kuuluvan tilat, joissa laitteet sijaitsevat sekä niiden käyttöön- ja jäähdytykseen kuluvan energian.

Neljännellä rivillä, fyysisen kerroksen päällä, on virtualisointi, jolla fyysiset resurssit voidaan jakaa usean eri käyttöjärjestelmän kesken. Virtualisoinnin päällä on käyttöjärjestelmä, mahdolliset väliohjelmistot ja ajoaikaiset kirjastot. Datan käsittely viittaa sovelluksen datan käsittelyyn, ja voi olla esimerkiksi tietokanta. Pinon päällimmäisenä on itse sovellus, joka riippuvainen kaikista muista tasoista. Kuvasta 3.1 nähdään, että eri palvelutason valinnalla voidaan vaikuttaa olennaisesti omiin vastuualueisiin ja abstraktion tasoon.

IaaS, Infrastruktuuri palveluna

IaaS tarjoaa käyttäjälle fyysisen tason ominaisuudet aina kuvan 3.1 virtualisointiin asti. Palveluna tarjotaan siis prosessointi, tallennustila, verkkoyhteydet ja muut perustavanlaatuiset fyysisen kerroksen ominaisuudet, joiden päälle käyttäjä voi rakentaa oman ympäristönä. Tyypillisin tapa IaaS-palvelulle on tarjota virtuaalikoneita, joihin käyttäjä voi asentaa haluamansa käyttöjärjestelmän ja sen päälle tarvittavat ohjelmistot. [75][92]

PaaS, Alusta palveluna

PaaS voidaan mieltää vastaavanlaiseksi palveluksi ohjelmistoille, kuin mitä IaaS on infrastruktuurille. PaaS-malli tarjoaa kuvan 3.1 mukaisesti kaikki ohjelmistotasot fyysisestä tasosta aina ajoaikaisiin vaatimuksiin. NIST [92] määritelmän mukaan PaaS tarjoaa käyttäjälle mahdollisuuden viedä pilveen käyttäjän luomia tai hankkimia sovelluksia, jotka on luotu palveluntarjoajan tukemilla ohjelmointikielillä, kirjastoilla ja työkaluilla. Käyttäjä ei voi vaikuttaa arkkitehtuurin abstrahoituihin fyysisiin ominaisuuksiin tai käyttöjärjestelmiin, mutta pystyy hallitsemaan ohjelmistoja ja konfiguroimaan ympäristön asetuksia.

PaaS siis mahdollistaa käyttäjälle web-ohjelmiston rakentamisen ja pilveen sijoittamisen ilman omaa infrastruktuuria. Tämä helpottaa kehittäjien taakkaa, koska PaaS-malli poistaa tarpeen huolehtia resurssien hankinnasta, kapasiteetin suunnittelusta, ohjelmistojen ylläpidosta, korjauksista ja muista sovelluksen pyörittämisen liittyvistä toiminnoista. [93]

SaaS, Ohjelmisto palveluna

Palvelutasoista abstraktiotasolla korkeimmalla on SaaS-palvelumalli. Kuvan 3.1 mukaan SaaS-tason palveluissa kaikki ohjelmistotason elementit jäävät palveluntarjoajan vastuulle. SaaS-palvelutaso tarjoaa käyttäjälle valmiin ohjelmiston, jonka toiminnasta ja hallinasta vastaa täysin palveluntarjoaja. SaaS-tason ohjelmistoa käytettäessä käyttäjän ei tarvitse välittää miten palvelu on rakennettu tai miten sitä hallinnoidaan, vaan vain siitä, miten sitä parhaiten voidaan soveltaa omiin tarkoituksiin. [58]

SaaS-ohjelmistot ovat nopeita käyttöönottaa, sillä ne pääasiallisesti toimivat selainpohjaisesti internetin yli, jolloin päätelaitteille ei tarvitse tehdä muutoksia ja ohjelmiston käyttö ei siten ole päätelaiteriippuvaista. SaaS-ohjelmistoissa käyttäjällä ei ole yleensä juurikaan mahdollisuuksia muokata ohjelmistoa omien tarpeidensa mukaiseksi. SaaS-ohjelmiston käyttäjä on vapautettu kaikesta ohjelmiston ja alustan ylläpidollisista tehtävistä, mutta vastaavasti on silloin täysin riippuvainen ylläpitäjän panostuksesta ohjelmistoon. Tämä voi toisinaan olla huono ominaisuus, koska SaaS-palvelun tapauksessa esimerkiksi tietoturva on kokonaan palveluntarjoajan vastuulla ja siten käyttäjän pitää olla hyvin

varma siitä, että ohjelmisto on toteutettu luotettavasti. SaaS-ohjelmistojen muokkaamattomuus voi myös muodostua ongelmaksi sen ja yrityksen omien tuotteiden välisessä integraatiossa, mikäli SaaS-ohjelmistolla ei ole olemassa yhteensopivaa rajapintaa. [75]

3.4 Julkiset pilvipalvelutarjoajat

Julkisen pilven palveluntarjoajia on markkinoilla useita, ja ne tarjoavat palveluita infrastruktuurin IaaS ja PaaS tasoista PaaS tason ohjelmistoihin saakka. Järjestelmät ovat nopeampia käyttöönottaa julkisessa pilvessä kuin fyysisesti tehdastasossa, ja ne tarjoavat käyttäjän näkökulmasta loputtomiin skaalautuvan alustan [98]. Palveluntarjoajilla on tarjolla myös omia ohjelmistokehitystä avustavia paketteja (engl. *SDK, Software Development Kit*) sekä tekoälyn ja datan ympärille rakennettuja palveluita.

Synergy Researchin mukaan vuoden 2022 tilikauden kolmannen neljänneksen jälkeen Amazon, Microsoft ja Google omistivat 66 % osuuden pilvi-infrastruktuurin (Iaas, PaaS ja yksityisen pilven isännöinti) markkinaosuudesta, jonka kokonaisliikevaihto oli 57 miljardia dollaria [52]. Suurin markkinaosuus on Amazonin palveluilla ja toisena tulee Microsoftin palvelut. Tämän työn kontekstissa pilvipalveluntarjoajiksi valitaan nämä kaksi markkinajohtajaa. Muita tunnettuja maininnan arvoisia palveluntarjoajia ovat Google Cloud Services, joka on investoinut kaksi miljardia Haminaan sijoitettuihin palvelinkeskuksiin sekä IBM ja Alibaba.

3.4.1 Amazon Web Services

AWS (*Amazon Web Services*) on maailman suosituin julkinen pilvialusta, joka tarjoaa yksityishenkilöille, yrityksille ja hallituksille joustavia, luotettavia ja skaalautuvia pilviratkaisuja kaikilla pilven palvelutasoilla. AWS tarjoaa yli 200 erilasta palvelutuotetta laskennasta, datan varastoinnista ja tietokannoista aina koneoppimiseen, analytiikkaan ja esineiden internettiin asti. Alustaa käytetään mm. nettisivujen ja ohjelmistojen isännöintiin, median jakoon, tallentamiseen ja kehitykseen.

AWS:n datakeskukset ovat yhdistetty toisiinsa yksityisillä redundanttisilla kuituyhteyksillä ja ne muodostavat niin sanottuja saatavuusvyöhykkeitä. Yhdellä maantieteellisellä alueella kuten Irlannissa voi olla useita saatavuusvyöhykkeitä. Saatavuusvyöhykkeet ovat kuitenkin maantieteellisesti hajautettu eri luonnonkatastrofialueille eivätkä ole riippuvaisia samoista energiapalveluntuottajista. Usealle saatavuusvyöhykkeelle hajautetut järjestelmät parantavat vikasietoisuutta ja saatavuutta. [51]

Palvelun käyttökustannukset riippuvat käyttäjän valitsemista palveluista, laitteistoista, käyttöjärjestelmistä, ohjelmistoista ja verkkoyhteyksistä [18]. Palveluiden hinnasto on myös riippuvainen käytetystä maantieteellisestä sijainnista [51]. AWS tarjoaa osan palveluista ilmaiseksi tiettyyn käyttörajaan asti ja osalle palveluista tarjotaan 12 kuukauden ilmainen kokeilujakso [18]. Seuraavaksi esitellään muutama AWS:n palvelu sekä niiden kustannusperusteita.

AWS Elastic Compute Service (EC2)

EC2 on AWS:n IaaS-tason palvelu. Sen kautta voidaan käynnistää tarpeellinen määrä virtuaalikoneita ja hallita niiden tieto- ja kyberturvallisuutta, yhteyksiä, tallennustilaa sekä käyttöjärjestelmää. Käyttäjä voi itse valita virtuaalikoneiden teho- ja tallennusominaisuudet sekä skaalata niitä tarpeen mukaan manuaalisesti tai automaattisesti. Käyttäjä maksaa tuntiperusteisesti aktiivisista virtuaalikoneista, joiden hinnat vaihtelevat ominaisuuksien mukaan muutamasta dollarista kuukaudessa aina tuhansiin dollareihin asti. [11]

AWS Lambda

AWS Lambda on serveritön FaaS (*Function as a Service*) -palvelu, joka ajaa käyttäjän ohjelmakoodia läpi tapahtumapohjaisesti. AWS Lambda tarjoaa serverin ja järjestelmän ylläpidon, automaattisen skaalauksen ja kapasiteetin osoituksen, tietoturvan sekä ohjelmakoodin monitoroinnin, jolloin käyttäjälle jää vain itse ohjelmakoodin toimitus. AWS Lambda skaalaa automaattisesti tarvittavat resurssit ohjelman tarpeen mukaan. Tapahtuma voi olla esimerkiksi HTTP-pyyntö tai päivitys tietokantaan. Palvelun kustannukset pohjautuvat toteutuneisiin pyyntöihin ja niiden ajoaikaan. [16]

AWS Application Migration Servcice (MGN)

AWS MGN -palvelulla voidaan automaattisesti siirtää virtuaalialustoilla tai fyysisille servereillä ajossa olevia servereitä suoraan AWS:n palveluihin. Palvelua käyttää ”Lift and shift” -metodia, jossa AWS MGN replikoi lähdeserverin EC2-palveluun uudeksi virtuaaliseksi pilviserveriksi. AWS MGN on palveluna ilmainen käyttää ensimmäiset 99 päivää. Pieniä kuluja kuitenkin aiheutuu, koska migraatioprosessin kuluttamat resurssit ovat maksullisia. [14]

AWS Identity and Access Management (IAM)

AWS IAM on käyttöoikeuksien hallinnan kautta tietoturvallisuutta parantava palvelu. IAM:llä voidaan luoda AWS-käyttäjätilille jaettuja käyttäjiä, rajoittaa näiden käyttäjien oikeuksia palveluihin, resursseihin ja dataan sekä ottaa käyttöön usean askeleen tunnistautuminen. IAM on palveluna ilmainen käyttää. [15]

Amazon Relational Database Service (RDS)

Amazon RDS on hallinnoitu relaatiotietokanta, joka mahdollistaa tunnettujen tietokantamoottoreiden kuten Amazon Auroran, MySQL:n ja Microsoft SQL -serverin käytön. Amazon RDS hoitaa tietokantojen rutiininomaiset asiat kuten päivitykset, varmuuskopioinnin, palautuksen, vikojen havainnoinnin ja korjaukset. RDS mahdollistaa helpomman hallinnoinnin, eritasoisia suorituskykyjä, skaalautuvuutta sekä parempaa tietoturvaa salauksilla. Palvelua käytettäessä maksetaan käytettyjen resurssien mukaisesti. [12]

Amazon S3

Amazon S3 tarjoaa palveluna Object Storage -mallin mukaista tallennustilaa. Amazon S3:een voidaan tallettaa millaisia objekteja tahansa, kuten kokonaisia internet applikaatioita, RDS-palvelun varmuuskopioita, data-arkistoja ja datajärviä. Amazon S3 tarjoaa 8 erilaista varastointitasoa, joissa kestot, saatavuus ja tehokkuus vaihtelevat. Hinta määräytyy käytön ja käytetyn luokan mukaan. [13]

Amazon CloudWatch

Amazon CloudWatch on monitorointiin ja hallintaan tarkoitettu palvelu, joka tuottaa dataa ja päätöksentekoa auttavia tiivistelmiä muista AWS:n palveluista. CloudWatch kerää yhteen paikkaan käytettyjen palveluiden tuottamaan operatiivisen datan lokien ja metriikan muodossa. Sillä voidaan esimerkiksi monitoroida EC2 serverien käyttöasteita. Lisäksi CloudWatch mahdollistaa hälytysten ja ilmoitusten käytön. Pääosa käyttäjistä pystyy käyttämään CloudWatchia AWS:n ilmaisen tilin puitteissa. [10]

AWS IoT Core

IoT Core on AWS:n IoT-laitteiden pilveen yhdistämiseen, turvalliseen datan vaihtoon ja datan prosessointiin tarkoitettu palvelu. Se tarjoaa laitteille pilveen sisääntulopisteen käyttäen protokollana esimeriksi MQTT:tä tai HTTPS:ää, viestinvälittäjänä julkaisu ja tilaus mallin kommunikaatioon, todennuksen ja varmennuksen, laiteen tilan tallennuksen, laiteen metadatan tallennuksen sekä viestien muutokset ja reititykset. Palvelu on osa AWS:n IoT:lle suunnattua palvelumallistoa, ja siihen kuuluu muita palveluita kuten AWS IoT Device Manager. Hintaan vaikuttaa yhteyksien käyttö sekä viestien ja tallennuksien määrät.

3.4.2 Microsoft Azure

Azure on Microsoftin vuonna 2010 lanseeraama yrityskäyttöön suunnattu pilvipalvelualusta. Se tarjoaa hyvin samankaltaisia palveluita kuin AWS ja on tällä hetkellä maail-

malla toiseksi suosituin pilvialusta. MS Azuressa on yhteensä yli 200 palvelua jakaantuneena kaikille kolmelle palvelutasolle pyrkien kattamaan mahdollisimman laajalti erilaiset asiakastarpeet.

Maantieteellisesti Azuren datakeskukset on hajautettu yli 60 Azure-alueelle, joista osa on vielä rakennusvaiheessa. Microsoft julkisti vuoden 2022 alussa aikovansa rakentaa datakeskuksen myös Etelä-Suomeen. Azuren saatavuusvyöhykkeet ovat fyysisesti erotettuja datakeskuksia Azure alueiden sisällä, joiden fyysinen infrastruktuuri on AWS:n saatavuusvyöhykkeiden tavoin rakennettu toisistaan riippumaattomiksi. [19]

Maksupolitiikaltaan Azure on hyvin samankaltainen kuin AWS. Molemmissa maksut perustuvat käytettyihin tunteihin ja maksetaan vain siitä mitä käytetään. Azure tarjoaa AWS:n mukaisen mahdollisuuden virtuaalikoneiden osalta vuokrata niitä pidemmäksi aikaa halvemmalla hinnalla. Azure tarjoaa ilmaisen tason lisäksi uusille käyttäjille 200 dollarin arvosta ilmaista käyttöä sen palveluihin tietyin rajoituksin. [21]

Azuren palvelut jaetaan 13 kategoriaan, joista oleellisimmat SCADA-järjestelmälle ovat analytiikka, laskenta, tietokannat, tietoturva, IoT, varastointi ja migraatio. Nämä kategoriat sisältävät muutamasta 20:een erilaista palvelua, joista monet ovat rinnastettavissa suoraan AWS:n samankaltaisiin palveluihin. AWS:n palveluiden samankaltaisuudesta johtuen esitetään tiivistetysti osa näistä Azuren kategorioiden tarjoamista palveluista ilman syventymistä itse palveluihin.

Azuren laskentapalveluihin sisältyy Virtual Machines -palvelu, joka nimensä mukaan tarjoaa skaalautuvia virtuaalikoneita sekä Azure Dedicated Host -palvelu, joka tarjoaa asiakkaalle varattuja servereitä, joiden fyysiset resurssit eivät ole virtuaalisesti usealla käyttäjälle jaettuja. Azure Functions on myös osa laskentapalveluita ja vastaa Amazonin AWS Lambdan serveritöntä tapahtumapohjaista ohjelmakoodin ajoa. [33]

Tietokantapalveluihin kuuluu laaja tarjonta erilaisia tietokantaratkaisuita sisältäen sekä relaatiotietokantoja, että relaatiomallista poikkeavia tietokantoja. Tunnetuimpia Azuren tarjoamia tietokantapalveluja ovat Azure Cosmos DB, Azure SQL Database, Azure Database for MySQL ja Azure Database for PostgreSQL. Tietokantapalveluista löytyy myös tietokannan pilvimigraatiota helpottava Azure Database Migration Service. [94]

IoT-palveluiden osalta tärkeä palvelu on Azure IoT Hub, jonka voidaan kuvata olevan laitteiden yhdyskäytävä Azuren pilveen. Sen avulla voidaan luoda turvallinen ja luotettava kommunikaatiokanava IoT-palvelun ja laitteiden välille sekä integroida ne osaksi muita palveluita. Erottavana tekijänä AWS:n IoT-palveluihin on, että Azuren laitehallintapalvelu ei ole erillinen vaan kuuluu osaksi IoT Hub -palvelua. Se tarjoaa laitteen tilan monitorointia, laitteiden konfigurointia ja etäpäivityksiä. [20]

4. PILVIPOHJAISEN SCADA:N TAUSTATUTKIMUS JA STATE OF THE ART

Pilvi mahdollistaa useita erilaisia järjestelmäarkkitehtuureja. Arkkitehtuureissa voidaan vaihdella pilven eri palvelutasojen ja käyttöönottomallien välillä, järjestelmään voidaan lisätä redundanttisia komponentteja ja pilveen vietävien SCADA-järjestelmän komponenttien määrää voidaan vaihdella. Lisäksi osa esiteltävistä tutkimuksista esittää teollisuusautomaation kehityksessä yleistyneen SOA-mallin käyttöä SCADA-järjestelmän pilveen viennissä. Teknologian uudehkon luonteensa vuoksi ei ole vielä olemassa vakiintuneita ja hyvin tutkittuja arkkitehtuurimalleja, jonka vuoksi työn kontekstissa State of the Art -ratkaisut käsittelevät tässä luvussa esitettyjä tieteellisessä kirjallisuudessa esiintyneitä ratkaisuita. Taustatutkimusten lisäksi esitellään myös kaksi kaupallistettua ratkaisua. Ratkaisut esitellään, jotta lukijalle muodostuisi käsitys SCADA-järjestelmän pilveen viennin mahdollisista arkkitehtuureista, millaista tutkimusta aihepiiriin liittyen on tehty sekä millaisiin vaatimuksiin tutkimukset keskittyvät. Tutkimusten aihepiirit itsessään kertovat integraation mahdollisista ongelmista ja toisaalta mahdollisuuksista.

Seuraavissa alaluvuissa esitellään eri pilven palvelutasoihin perustuvia arkkitehtuureita sekä kaupalliset ratkaisut Ignition SCADA ja Aveva Insight. Lopuksi esitellään lyhyesti palvelukeskeisen arkkitehtuurin käyttöä. Arkkitehtuurin lisäksi esitetään mihin vaatimukseen kyseisellä arkkitehtuurilla vastataan ja kuinka hyvin on onnistuttu. Esiteltyjä ratkaisuita käytetään argumenttien tukena seuraavan luvun vaatimusten toteutumisen pohdinnassa.

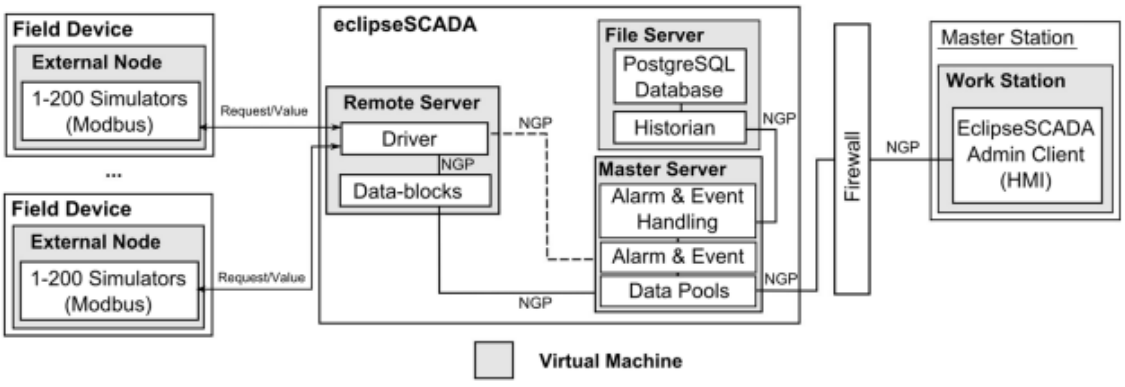
4.1 IaaS- ja PaaS-pohjaiset toteutukset

IaaS- ja PaaS-pohjaiset toteutukset ovat lähellä toisiaan, ja periaatteessa toteutuspohjat voidaan aina vaihtaa päittäin. Toteutuksen kannalta erona näiden välillä on, että IaaS-pohjainen antaa laajemmat mahdollisuudet tehdä muutoksia käytettävään pilviympäristöön. SCADA-ohjelmistoja käytetään ajoittain perinteisessä teollisuusautomaatiossa virtuaaliympäristöissä. IaaS- ja PaaS-tasot tarjoavat tällaisia virtualisoituja ympäristöjä, joten luonnollisesti kyseiset tasot soveltuvat olemassa olevien SCADA-ohjelmistojen pilvimigraatioon. Seuraavissa alaluvuissa esitellään IaaS- ja PaaS-tason taustatukimusta ja lopuksi kaupallinen Ignition SCADA.

4.1.1 Julkinen pilvi ja IaaS-taso

Church et al. tutkivat artikkelissaan [31] SCADA-järjestelmän pilvimigraatiota IaaS-tason pilvipalveluun. Tutkimuksessa tuodaan tuloksena esille toteutuksen aikana ilmenneitä ongelmia ja hyviä suunnittelun tapoja sekä arvioidaan migraation reaaliaikaisuutta. Jotta tutkimustulokset ovat sovellettavissa muihinkin järjestelmiin on tutkimuksen migraation SCADA-järjestelmäksi valittu avoimen lähdekoodin ohjelmisto, joka on ominaisuuksiltaan mahdollisimman samankaltainen kuin laajalti käytetty Siemensin WinCC. Tutkimuksessa päädyttiin SCADA-järjestelmän osalta Eclipse NeoSCADA:n käyttöön ja pilvialustana toimi Australian yliopistojen tutkimukseen tarkoitettu NeCTAR.

Tutkimuksessa migraatio toteutettiin ”lift and shift”-metodilla, jossa ohjelmisto yksinkertaisesti vaihtaa toimintaympäristöä perinteisestä serveristä pilvialustalle. Metodissa ohjelmisto ladataan uudestaan pilviympäristöön ja sen toiminta nojautuu IaaS-tason toiminnallisuuksiin. Tämä migraation metodi on kaikista yksinkertaisin toteuttaa, mutta mikäli pilvipalvelun ominaisuuksia halutaan hyödyntää laajemmin, voidaan ohjelmiston rakennetta joutua muuttamaan paljonkin. Tutkimuksessa NeCTAR pilvialustalle vietiin kenttälaitteiden simulaatio, etäserveri, pääserveri ja dataserveri, jotka kaikki luotiin omille virtuaalikoneilleen. Etäserverin tehtävänä on hakea reaaliaikaisesti dataa sensoreilta, muuttaa se Eclipse NeoSCADA:n sisäisen NGP-protokollan muotoon ja lähettää eteenpäin pääserverille. Pääserveri vastaa hälytysten ja tapahtumien käsittelystä ja dataserveri säilyttää historiadataa relaatiotietokannassa. Mitta- ja toimilaitteiden simuloinnin kommunikaatioon käytetään Modbus/TCP-protokollaa. Kuvassa 4.1 on esitetty tutkitun järjestelmän rakenne.

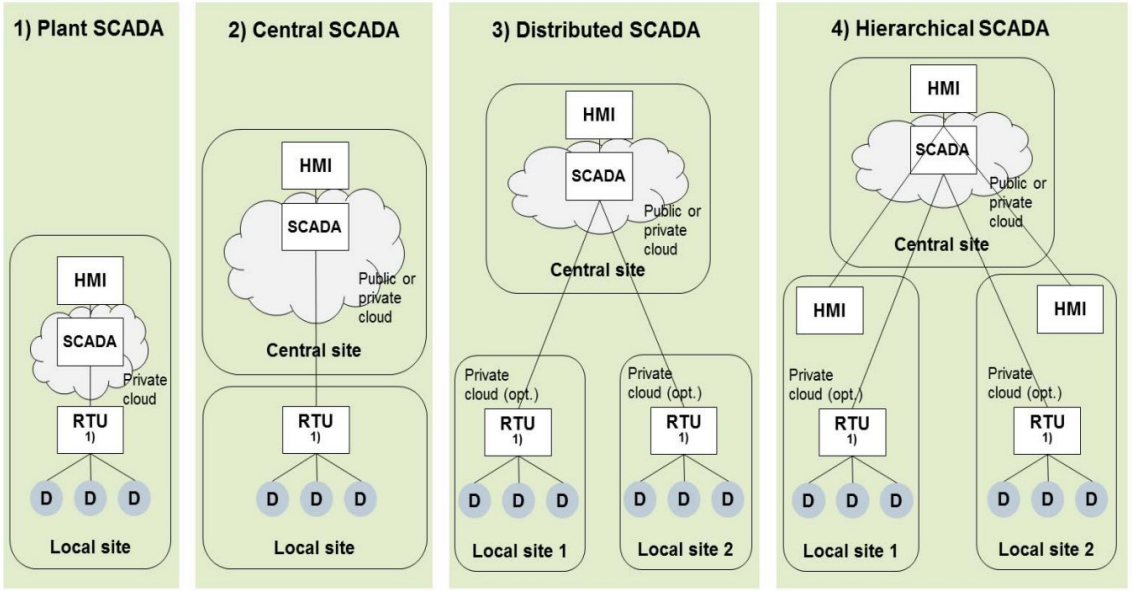


Kuva 4.1. *Tutkimuksen [31] järjestelmän rakenne.*

Tutkimuksessa arvioitiin sekä monitoroitavan ympäristön, että maantieteellisen hajautuksen vaikutusta järjestelmän reaaliaikaisuuteen. Tutkimuksessa kenttälaitteita simuloivat virtuaalikoneet olivat sijoitettu Tasmaniaan ja SCADA-järjestelmän komponentit 430

km päähän Melbourneen. Simuloitujen mittalaitteiden määrä vaihteli 200 ja 800 välillä, ja pyyntöjen intervalli oli 1 ms, jolla pyrittiin aiheuttamaan järjestelmään ylimääräistä kuormitusta simuloimaan suurempaa mittausten määrää. Tutkimuksessa todettiin, että mittauskohtainen RTT (engl. *Round Trip Time*) nousi, mitä enemmän järjestelmässä oli mittalaitteita. Lisäksi tutkittiin hajautuksen vaikutusta siirtämällä ensin kaikki komponentit samalle maantieteelliselle alueelle ja toisella kerralla samalle virtuaalialustalle. Samalla maantieteellisellä alueella sijaitessa RTT laski sensorikuormasta riippuen n. 10 % ja samalla virtuaalikoneella sijaitessa n. 15 %. Maantieteellinen hajautus tutkimuksen mukaan aiheuttaa siis pientä viivettä tiedonsiirtoon, mutta suurin osa RTT:stä kuluu prosessointiin eikä tietoliikenteeseen. Tutkimuksessa todetaankin, että olisi hyvä käyttää tapahtumapohjaisia protokollia mittausdatan tiedonsiirtoon redundantin informaation minimoimiseksi.

Migraatiosta IaaS-tason pilvipalveluun ja reaaliaikaisuuden arvioinnista on tehty myös toinen hyvin samankaltainen tutkimus. M. Yi et al. [102] tutkivat kommunikaatioviiveitä neljässä erilaisessa tyypillisessä pilvipohjaisessa SCADA-järjestelmässä. Tutkimuksessa arvioidaan simuloiduilla mittalaitteilla viiveitä näissä erilaisissa järjestelmissä, tieto- ja kyberturvallisemmassa ratkaisussa sekä eri protokollilla. Pilvipalveluna tutkimuksessa toimii luvussa 3.4.1 esitetty AWS, jonne on pystytetty jokaiselle komponentille oma virtuaalikone. Kuvassa 4.2 on esitetty tutkimuksen neljä erilaista tutkittavaa järjestelmäarkkitehtuuria.



Kuva 4.2. *Tutkimuksen [102] neljä erilaista järjestelmämallia.*

Kuvan 4.2 ensimmäinen malli 1) soveltuu paikallisesti esimerkiksi tehtaaseen tai johonkin muuhun yhden maantieteellisen paikan järjestelmään. Vaikka järjestelmä on kokonaan

paikallinen, voidaan siinä hyödyntää pilveä SCADA-komponenttien virtualisointiin ja siten saada käyttöön pilven hyötyjä ja fyysisten resurssien jakamista. Kuvan toisessa mallissa 2) järjestelmä on jaettu paikalliseen ja keskitettyyn osaan. Paikallinen osa lähettää keskitetylle dataa ja hälytyksiä. Yleensä tällä välillä protokollana käytetään SCADA-järjestelmille tyypillisiä protokollia kuten esimerkiksi Modbus/TCP, mutta SCADA-järjestelmän sisäiset protokollat soveltuvat tiedonsiirtoon paremmin. Tästä johtuen RTU:lla tehdään protokollamuunnos SCADA-järjestelmän sisäiseen protokollaan. Kolmannessa mallissa 3) idea on sama kuin edeltävässä, mutta siinä paikallisia järjestelmiä voi olla useita ja ne voivat sijaita eri maantieteellisissä sijainneissa. Tämä malli sopii hyvin yrityksille, joilla on monitoroitavana useita hajautettuja asemia. Useat nykypäivän suuret SCADA-järjestelmät ovat toteutettu neljännen hierarkkisen SCADA-mallin mukaan. Siinä on edeltävään verrattuna lisänä prosessiasemilla omat käyttöliittymät oman aseman monitorintiin ja ohjaamiseen. Tässä toteutuksessa on tärkeää ottaa huomioon pidentyneet tiedonsiirtoviiveet, koska data siirtyy ensin asemalta keskitettyyn pilveen ja sieltä takaisin aseman käyttöliittymälle sekä sieltä samaa reittiä takaisin RTU:lle.

Tutkimuksen ensimmäisessä vaiheessa tutkittiin virtualisoinnin vaikutusta viiveeseen ensimmäisen mallin tapauksessa, jossa verkkoyhteys ei vaikuta viiveeseen. Siinä verrattiin fyysisillä servereillä toteutettua ja AWS-palveluun virtualisoitua ratkaisua keskenään. Tutkimuksessa todettiin, että tiedonsiirtoviive nousee simuloitujen laitteiden määrän noustessa samalla tavalla kuin edeltävässä tutkimuksessa [31], mutta sen oletetaan johtuvan SCADA-järjestelmän ja simulaattorin toteutuksesta. Pilveen virtualisoinnilla ei havaittu olevan merkityksellistä vaikutusta viiveeseen.

Toisessa kohdassa protokollan vaikutusta verrattiin myös ensimmäisen mallin mukaisella järjestelmällä, joka käytti julkista AWS-pilvipalvelua samalla maantieteellisellä sijainnilla kaikkien järjestelmän osien ollessa samassa aliverkossa. Protokollina vertailussa olivat kyselypohjainen Modbus/TCP ja tapahtumaperusteinen IEC870-5-101. Tuloksena tutkimuksessa todetaan tapahtumaperustaisella olevan huomattavasti pienempi viive pienemmällä vaihtelulla.

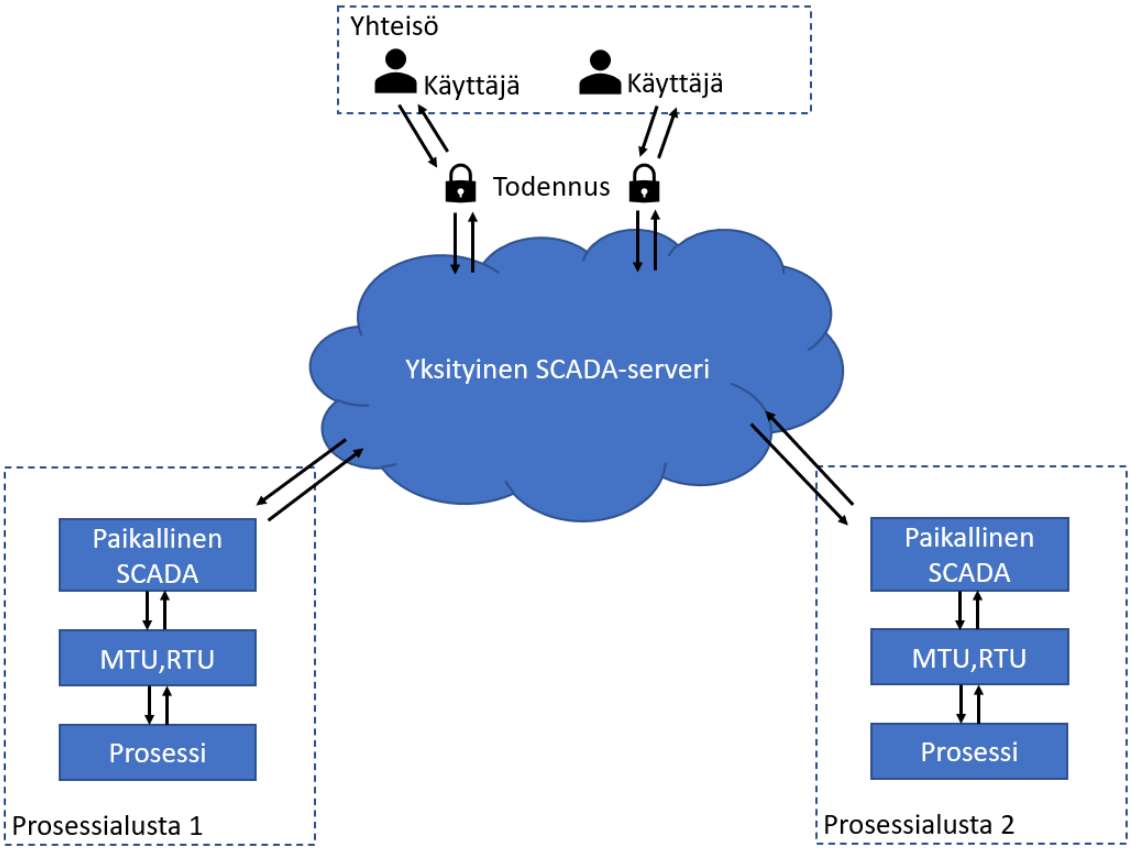
Tutkimuksen kolmannessa vaiheessa tutkittiin tieto -ja kyberturvan parantamista lisäämällä SSL-salattu VPN-tunnelointi kolmannen mallin paikallisen ja keskitetyn aseman välille. Lisäksi paikallisen ja keskitetyn aseman välillä on 1000 km maantieteellinen matka. Tunnelin päättämäinen tehtiin samoilla SCADA-komponentin virtuaalikoneilla. Tuloksena todettiin, että salauksen käyttö ei tuo järjestelmään merkittävää viivettä.

Viimeisessä tutkimuskohteessa vertailtiin kuvan 4.2 mukaisten järjestelmämallien välisiä tiedonsiirtoviive-eroja käyttäen Modbus/TCP-protokollaa. Erona mallien tiedonsiirtoviiveissä on maantieteellisen matkan aiheuttamat viiveet SCADA-komponenttien välillä. Ensimmäisessä mallissa komponentit ovat paikallisia, mutta jo toisessa mallissa elementit ovat hajautetut. Hajautuksen todetaan lisäävän 1000 km maantieteellisessä välillä n. 30 ms viivettä verrattuna ensimmäiseen malliin. Kolmannessa hajautetussa mallissa viiveen jakauma on pienempi kuin toisessa mallissa, joka osoittaa kenttälaitteiden ja etäservereiden jakamisen pienempiin osiin tehostavan tiedonsiirtoa. Hierarkkisessa mallissa viiveet lisääntyivät joitain kymmeniä millisekunteja johtuen edestakaisesta tietoliikenteestä. Yhteenvetona tutkimuksessa todetaan, että SCADA-komponenttien vienti pilveen ei aiheuta kriittisiä ongelmia suorituskykyyn.

4.1.2 Hybridipilvi ja PaaS-taso

V. Nguyen et. al. tutkimuksessa [73] ehdotetaan pilvipohjaista ratkaisua mikroverkkoalustojen yhteentoimivuuden mahdollistamiseksi. Yhteentoimivuus hybridipilvessä mahdollistaa toimijoille merkityksellisen informaation jakamisen, pääsyn jaettuun resurssikantaan sekä mahdollisuuden lainata toistensa resursseja omiin tarkoituksiin. Ehdotuksessa otetaan huomioon tieto- ja kyberturvallisuus, järjestelmän kriittisen ohjausten vaatimat reaaliaikavaatimukset sekä helppous tuoda järjestelmään uusia partnereita. Tutkimuksessa otetaan kantaa myös kommunikaation toimintaan toimijoiden välillä, mutta tässä se sivuutetaan ja keskitytään tutkimuksessa päädyttyyn pilviarkkitehtuuriin.

Tutkimuksessa päädyttiin PaaS-tason toteutukseen, sillä PaaS-taso mahdollistaa käyttäjälle ohjelmistojen hallinnan, mutta estää käyttäjältä pääsyn abstrahoituun IT-infrastruktuuriin ja dataan. Usean eri tahon käyttäjän järjestelmässä on tärkeää, että pääsyä dataan ja eri järjestelmän osiin pystytään rajoittamaan tieto- ja kyberturvallisesta näkökulmasta. Järjestelmä on jaettu paikallisiin SCADA-servereihin ja yksityiseen pilveen vietyyn SCADA-serveriin. Kuvassa 4.3 on esitetty tutkimuksen järjestelmän rakenne.



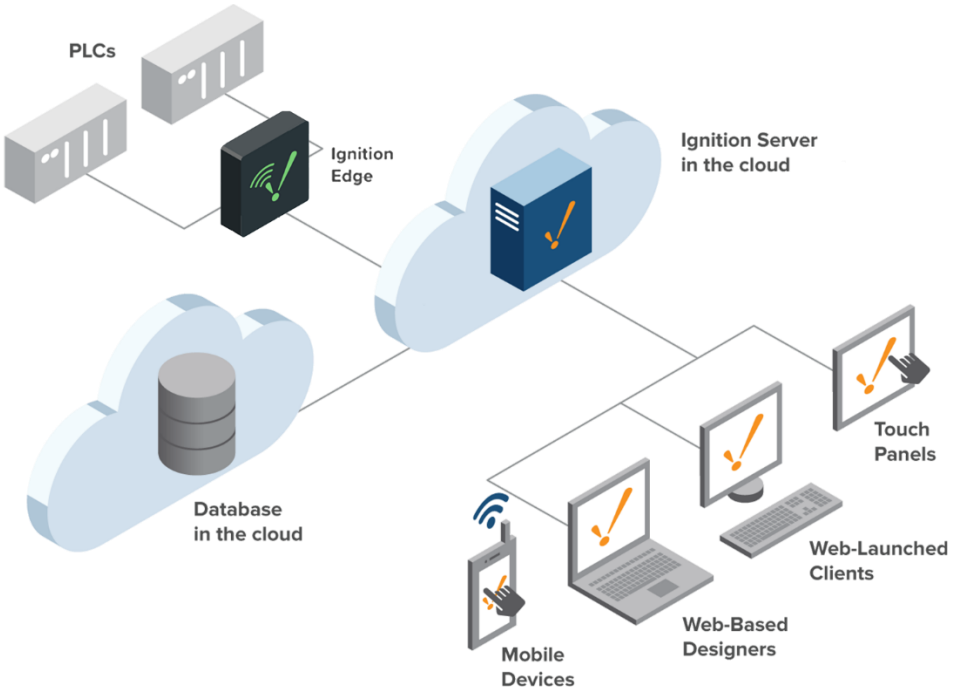
Kuva 4.3. *Tutkimuksen [73] mukainen järjestelmärakenne. Mukaillen [73].*

Paikallinen SCADA hoitaa mikroverkon ohjauksen ja muut kriittiset toimet, jolloin vastataan luotettavuus sekä tieto- ja kyberturvallisuusvaatimuksiin. Paikallinen SCADA toimii lähiverkossa, joka mahdollistaa reaaliaikaiset toiminnot pienillä viiveillä, sekä tietoturvallisesti antaa hyvän suojan, koska informaatiota vaihdetaan vain paikallisesti. Itse yhteentoimivuus tuotetaan yksityisessä pilvessä, josta löytyy käyttäjille yhteiskäyttöiset ohjelmistot, tutkimuksen tapauksessa esimerkiksi virtualisointi ja raportointi etäkäyttäjille, historiadata sekä optimointi mahdollisuuksia. Paikallinen SCADA lähettää tarvittavat tiedot pilven yksityiselle SCADA-järjestelmälle. Yksityisen pilven käyttö usean eri tahon toiminnassa myös parantaa turvallisuutta. Pilvi ei ole avoin, jolloin sinne pääsee vain onnistuneella todennuksella ja se voidaan tehdä tiukasti monitoroiduksi. Lisäksi voidaan lisätä todennus pilven ja paikallisen SCADA:n välille, jolloin voidaan hallita, kenellä on oikeus päästä käsiksi esimerkiksi eri tahojen prosessialustoihin.

4.1.3 Ignition SCADA

Pilvipohjaisen SCADA-järjestelmän State of the Art esimerkkinä voidaan pitää jo yli vuosikymmenen kehitettyä Ignition SCADA -järjestelmää. Ignition SCADA on rakennettu

alusta alkaen Javalla mahdollistaen laajan pilven ominaisuuksien hyödyntämisen. Ignition SCADA:n kerrotaan verkkosivuillaan olevan yhteensopiva lähes kaikkien PLC-komponenttien kanssa, jolloin käyttäjä voi hyödyntää jo olemassa olevaa infrastruktuuria ongelmitta. Ignitionin toiminta perustuu yhden serverin toimintaan, joka voi sijaita pilvessä tai fyysisellä serverillä. Ignition SCADA:n kehittäjä Inductive Automation ei tarjoa pilvipalveluita, mutta suosittelee pilvipalvelun tarjoajaksi Amazonin EC2:sta tai MS Azurea. Asemat voidaan yhdistää serveriin esimerkiksi käyttäen VPN-yhteyttä ja Ignition Edgeä. Asiakkaat yhdistävät päätelaiteriippumattomasti serveriin internetin välityksellä. [34]



Kuva 4.4. *Ignition SCADA:n pilvipohjainen esimerkkiarkkitehtuuri [34].*

Ignition SCADA -järjestelmällä on mahdollista käyttää redundanttisia komponentteja niin pilvessä kuin fyysisesti. Osa pilvipalvelun tuottajista tarjoaa sisäänrakennetun automaattisen redundanssin, jossa alustan vikaantuessa se replikoidaan toiselle toimivalle alustalle. Tässä kestää kuitenkin huomattavasti pidempään kuin käyttämällä Ignitionin omaa redundanssia kahdella redundanttisella serverillä. Ignition SCADA tarjoaa skaalautuvuutta modulaarisella arkkitehtuurilla, jossa järjestelmään voidaan lisätä erilaisia komponentteja myös jälkikäteen. Lisäksi järjestelmän lisensointi on tehty yksinkertaiseksi, jossa yhdellä lisenssillä saa yhden Ignition serverin ja sille loputtoman määrän datapisteitä ja asiakasyhteyksiä.

4.2 SaaS-tason toteutus

SaaS-tason palvelumallissa palveluntarjoajalla on vastuu kaikista ohjelmistotason elementeistä. Palveluntarjoaja tarjoaa käyttäjälle valmiin ohjelmiston, jonka hallinta ja kehitys on tarjoajan vastuulla. Käyttäjälle tämä on helpoin tapa käyttää pilvipohjaista SCADA-järjestelmää. Kääntöpuolena on, että käyttäjällä ei kuitenkaan ole mahdollisuutta vaikuttaa järjestelmän toiminnallisiin ominaisuuksiin, jolloin pitää olla varma, että järjestelmä sopii omaan käyttötarkoitukseen sellaisenaan. Kupalliset ratkaisut ovat pilveen räätälöityjä, ja siten mahdollistavat järjestelmän käyttäjälle pilven ominaisuuksien laajan hyödyntämisen. SaaS-tason kaupallisia SCADA-toteutuksia on markkinoilla vielä vähän ja toisinaan niistä käytetään nimitystä ”SCADA-as-a-Service”. Suuri osa teollisuusautomaatioon liittyvistä SaaS-tason palveluista keskittyy pääosin data-analytiikkaan, tuotannon optimointiin, datan helppoon jakoon, turvallisuuteen ja automaatiopyramidin korkeamman tason elementteihin kuten MOM-järjestelmiin. Joillain palveluntarjoajilla on SaaS-tason SCADA:n lisäksi tarjolla näitä edellä mainittuja pilvipalveluita, jotka voidaan helposti yhdistää osaksi SCADA-järjestelmää. Seuraavissa kappaleessa esitellään lyhyesti Emerson Electricin SaaS-pohjainen SCADA-ratkaisu Zedi SaaS SCADA, Yokogawan SCADA-as-a-Service palvelu ja SKYDA-järjestelmä.

Emerson Electric on suuri monikansallinen amerikkalainen yritys, joka osti Zedi:n ohjelmisto- ja automaatioyrityksen vuonna 2019. Zedi SaaS SCADA on pääasiallisesti tarkoitettu jatkuva-aikaisiin prosesseihin öljyn- ja kaasuntuotantoon, kaasun jakeluun sekä veden ja jäteveden hallintaan. Zedi SaaS SCADA käyttää Microsoftin MS pilvipalvelua alustanaan. Emersonin nettisivun [104] mukaan se tarjoaa intuitiivisen ja skaalautuvan pilvialustan, joka mahdollistaa prosessin turvallisen reaaliaikaisen monitoroinnin ja ohjauksen, sekä data-analytiikkaa. Järjestelmä tarjoaa web-pohjaisen käyttöliittymän sekä puhelinsovelluksen. Järjestelmässä SCADA:n rooli on kuitenkin sivuosassa ja pääasiassa tarjotaan data-analytiikka erilaisina työkaluina, kuten erilaisina raportteina, trendeinä ja tekoälynä. Järjestelmä voi esimeriksi tarjota koneoppimiseen pohjautuvaa raportointia sen havaitsemista vuodoista putkilinjastoissa. [104]

Yokogawa on yli 100 vuotta sitten perustettu sähkötekniikan ja ohjelmistoalan yritys. Yokogawalla on oma tehokas, skaalautuva ja avoimen arkkitehtuurin SCADA-ohjelmisto FAST/TOOLS. Vuonna 2018 Yokogawa alkoi tarjota SCADA-as-a-Service -palvelua Uudessa-Seelannissa ja Australiassa keskittyen pääosin maantieteellisesti laajoihin prosesseihin kuten vedenjakeluun. SCADA-järjestelmänä pilvessä toimii heidän oma

FAST/TOOLS-ohjelmistonsa. Yokogawan Australian datakeskus ei ole yli viiteen vuoteen kokenut suunnittelematonta katkosaikaa ja takaa käyttäjille 99.99 % saatavuuden. [86][87]

SKYDA on tutkimuksessa [44] esitetty kyberturvallinen ja saatavuudeltaan paranneltu SCADA-as-a-Service arkkitehtuuriratkaisu. SKYDA käyttää kahta teknologiaa näiden ominaisuuksien tuottamiseen. Kyberturvassa SKYDA ei tähtää hyökkäysten estoon vaan pyrkii pitämään järjestelmän käyttökelpoisena ulkoisen tahon päästessä käsiksi järjestelmään. SKYDA käyttää useaa SCADA-ohjelmiston kopiota useassa eri pilvipalvelussa, jotka synkronoidaan keskenään tunkeutumista kestävällä replikointimoottorilla. Eri pilvipalveluiksi tutkimuksessa lasketaan myös maantieteellisesti hajautetut saman palvelun-tuottajan datakeskukset. Tämän avulla järjestelmä voi edelleen toimia yhden pilvipalvelun kaatuessa tai jonkin kopioin toiminnan ollessa kyseenalainen. Usean kopion lisäksi SKYDA käyttää kerrosverkkorakennetta poistaakseen internetin rajoitteet tiedonsiirrossa. Kerrosverkkorakenteena SKYDA käyttää Spines-rakennetta, joka mahdollistaa hyppyjen välisen pakettien uudelleenlähetyksen, verkon laajuisen ryhmälähetyksen ja mahdollisuuden määritellä reititykset.

Tutkimuksessa esitetään kolme erilaista redundanssiin perustuvaa konfiguraatiota. Ensimmäisessä konfiguraatiossa on käytössä primäärinen serveri ja kuuman valmiustilan redundanttinen serveri (engl. *HSB, Hot Stand By*), jotka molemmat on sijoitettu eri pilvipalveluun. Mikäli serverin virtuaalinen alusta hajoaa, voidaan serveri siirtää lokaalisti toiselle virtuaalialustalle ilman redundanttiselle serverille vaihtoa. Jos taas koko primäärisen serverin pilvipalvelu ei ole saatavissa, tehdään vaihdos redundanttiselle serverille toiseen pilvipalveluun. Redundanttisten serverien replikointi on tyypillisesti epäsynkronista, jolloin primäärinen serveri ei odota HSB-servereiltä kuittausta ennen oman toiminnan jatkoa.

Toisessa konfiguraatiossa yhden primäärisen sijaan on käytössä 3*f*+1 primääristä serveriä, jotka replikoidaan ja synkronoidaan toiseen pilvipalveluun HSB-servereiksi kuten ensimmäisessä konfiguraatiossa. Muuttuja *f* kuvaa kokonaislukuna määrää, monenko serverin toiminta voi olla samanaikaisesti kyseenalaista. Serverien ulostuloja vertailemalla voidaan päätellä kyseenalaiset serverit. Minimaalisessa konfiguraatiossa, jossa *f* on 1, on 4 primääristä serveriä ja niille 4 replikoitua HSB-serveriä, jolloin näistä voi yksi primäärinen ja siten yksi HSB-serveri olla kyseenalainen. Tunkeutumista vastustava replikaatioprotokolla vaatii jonkin verran lisää tiedonsiirtoa, jolloin viiveet saattavat lisääntyä. Aikakriittiset viestit kuitenkin kulkevat saman pilvipalvelun sisällä replikoilta toiselle ja redundanttisten HSB-servereiden synkronointi on edelleen epäsynkronisena, jolloin viiveet eivät haittaa toimintaa.

Kolmannessa konfiguraatiossa ei käytetä primääri/HSB lähestymistapaa, vaan primääriset serverit replikoidaan ja viedään omiin pilvipalveluihinsa. Nämä replikoidut serverit synkronoidaan samalla tunkeutumista vastustavalla protokollalla ja samalla $3f+1$ kaavalla. Synkronointiviestintä ei tapahdu tässä tapauksessa pilvipalvelun sisällä lokaalisti, vaan palvelusta toiseen, jolloin viiveet lisääntyvät. Tällä konfiguraatiolla voidaan varautua hyvin epätodennäköiseen tilanteeseen, jossa useampi pilvipalvelu kaatuu saman aikaisesti.

4.3 SOA-malli ja pilvi

Luvussa 2.6 esiteltyä palvelupohjaista arkkitehtuurimallia käytettäessä voidaan myös hyödyntää pilvipalveluiden tarjoamia mahdollisuuksia. SOA-mallin hajautetun luonteenmukaisesti palvelut eivät ole keskitettyjä ja ne on mahdollista viedä pilviympäristöön hajautetusti kaikki omille virtuaalialustoilleen tai FaaS-mallin mukaisille palveluille. Luvussa 2.6 esitetyn arkkitehtuurimallin tapauksessa voidaan esimerkiksi tietokanta rakentaa palveluna pilviympäristöön. SOA-mallin tutkimuksessa [43] suositellaan saatavuuden ja turvallisuuden parantamisen vuoksi tietokannaksi Database as a Service -mallista ratkaisua, jossa tietokanta sijaitsee pilvipalvelussa. SOA-mallin yhdistäminen automaatioympäristöön on SCADA:n pilveen viennin ohella uusi trendi, josta johtuen tutkimusta aihepiirien yhdistämisesti on olemassa hyvin niukasti.

WebSCADA esittää erään ratkaisun SCADA:n viennistä pilveen käyttäen SOA-mallia [83]. ICM-AESOP on Euroopan tutkimus- ja kehitysprojekti, jonka tarkoitus on realisoida SOA-mallinen lähestymistapa tulevaisuuden suuriin SCADA/DCS järjestelmiin ja esittää siirtymäpolku legacy-järjestelmistä parhaillaan kehitteillä oleviin [23]. Sen arkkitehtuuri keskittyy objektien yhteistyössä toimimiseen ja tukee näiden objektien ominaisuuksia vahvan integroinnin saavuttamiseksi [23]. WebSCADA-ratkaisuun verrattuna ICM-AESOP vie vielä enemmän automaatiojärjestelmän osia pilviympäristöön. SCADA:n lisäksi viedään esimerkiksi MES-järjestelmä ja ohjaus pilveen.

Lojka et al. tutkimuksessa [67] analysoidaan SOA-mallin käyttöä pilvipohjaisessa tuotantoautomaatiossa. Käytössä oli Windows Azure Pack yksityinen pilvi, jonne luotiin useita ydin- ja yhdistelmäpalveluita. Pilveen toteutettuja palveluita olivat yhteyksienmuodostus, verkkopalvelut, HMI, yksinkertainen MES, ohjaukset, datan prosessointi ja hallinta sekä hälytysten prosessointi ja konfigurointi. Tutkimuksessa todetaan, että heidän SOA-mallinsa käyttö SCADA:n pilvi-integraatiossa on soveltuvampi kuin yksikertainen ”lift and shift” ratkaisu.

5. JÄRJESTELMÄ- JA LUOTETTAVUUSVAATIMUSTEN TOTEUTUMINEN

SCADA-järjestelmän oikeellisen ja turvallisen toiminnan kannalta on olennaista, että sen toiminta täyttää sille asetetut järjestelmä- ja luotettavuusvaatimukset. Nämä vaatimukset eroavat perinteisistä IT-järjestelmän vaatimuksista jonkin verran, sillä SCADA-järjestelmän vääränlainen toiminta voi aiheuttaa vaaraa ihmisille ja ympäristöllä sekä tuottaa suuria taloudellisia menetyksiä. Pilvipalveluiden integrointi osaksi järjestelmää, ja järjestelmän altistaminen julkiselle internetille tuo lisähaasteita vaatimusten täyttymiselle. Kirjallisuudessa lähdetutkimuksissa pilvipohjaisisen SCADA-järjestelmän vaatimuksista on tutkittu paljolti reaaliaikaisuuden sekä tieto- ja kyberturvallisuuden toteutumista. Tässä työssä tutkitaan edeltävien lisäksi myös järjestelmän saatavuutta.

SCADA-järjestelmien pilveen viennissä erilaisia mahdollisia arkkitehtuureja on paljon. Niissä voidaan yhdistellä palvelutasoja ja käyttöönottomalleja eri tavoin, valita eri pilvipalveluntarjoajia, hajauttaa järjestelmiä ja luoda redundanttisia komponentteja. Useiden mahdollisten kombinaatioiden ja toisistaan eroavien tuotantoprosessien vuoksi ei ole olemassa yhtä kaikkialla toimivaa ratkaisua vaatimuksiin vastaamiseen, vaan ratkaisut ovat yleensä tapauskohtaisia. Vaatimusten toteutumista hankaloittaa ratkaisuiden taipumus vaikuttaa positiivisesti yhteen vaatimukseen ja negatiivisesti toiseen. Lisäksi vanhojen legacy-järjestelmien ja -komponenttien käyttö järjestelmissä hankaloittaa vaatimusten täyttymistä entisestään.

Mikäli SCADA-järjestelmän vaatimukset saadaan täytettyä myös pilvipohjaisessa järjestelmässä, toisi se mukanaan laajasti etuja sekä käyttävälle yritykselle, että järjestelmän suunnittelulle. Pilvipohjaisen järjestelmän suunnittelu voi mahdollisesti olla nopeampaa ja tuoda elinkaarensa aikana kustannussäästöjä, pilven käyttö mahdollistaa järjestelmälle luonnollisesti kaikki pilvipalvelun tuomat ominaisuudet kuten skaalautuvuuden, ja datan pakollinen vienti pilveen osana SCADA-järjestelmää alentaa kynnystä datan jatkokäytölle.

Seuraavissa luvuissa käsitellään edellä mainittujen vaatimusten toteutumista ja legacy-järjestelmien vaikutusta niihin pilvipohjaisessa SCADA-järjestelmässä. Kerätään niistä yhteenveto ja sen jälkeen käsitellään mahdollisia pilvipohjaisen järjestelmän tuomia etuja. Vaatimuksien pohdinnassa oletetaan RTU-yksiköiden edelleen sijaitsevan kentällä ja pilveen viedään vain HMI, hälytykset ja data.

5.1 Reaaliaikaisuus

Prosessia monitoroitaessa ja hallittaessa HMI:n kautta, on olennaista, että tiedonsiirtoviiveet eivät ole liian pitkiä oikea-aikaisen datan esittämisen ja sen pohjalta tehtävien ohjausmuutosten lähetysten vuoksi. SCADA-järjestelmän kiertokyselyväli kenttälaitteille on tyypillisesti yhdestä kahteen sekuntia [44]. Tyypillisesti prosessiasemalla sijaitseva SCADA-järjestelmän päätelaiteyksikkö, joka kerää, prosessoi ja tallentaa dataa sekä tarjoaa HMI:n, on yhteydessä etäpääteyksikön tai PLC-yksikön kanssa fyysisellä kenttäväylällä. Kenttäväylät ovat automaatiossa käytettyjä digitaalisia, sarjamuotoisia ja kaksisuuntaisen kommunikaation tietoliikenneprotokollia, jotka mahdollistavat hyvin nopean tietoliikenteen laitteiden välillä. Käytetyimmät kenttäväylät teollisuusautomaatiossa ovat Profibus ja Modbus.

Siirrettäessä päätelaiteyksikkö pilviympäristöön, vaatii se perinteiseen lokaaliin kommunikaatioon verrattuna muutoksia. Tiedonsiirtoa pilveen ei voida toteuttaa fyysisillä kommunikaatiokanavilla, jolloin data täytyy siirtää internetin ylitse IP-protokollilla. SCADA-järjestelmän käyttö pilvessä ei juurikaan vaikuta laskennallisiin prosesseihin kuluvaan aikaan. Tiedonsiirtoviiveiden kannalta olennaista on kuinka paljon ylimääräiset protokolla- ja datamuunnokset lähetys- ja vastaanottopäissä sekä ylimääräiset tietoverkkolinkit pilveen ja pilvestä prosessiasemalle vaikuttavat viiveeseen. Näin ollen tietoliikenteeseen vaikuttaa huomattavasti prosessiaseman ja käytetyn pilven datakeskuksen välisen verkkoyhteyden ominaisuudet ja olosuhteet. [44]

Julkisen pilven käyttö nostaa riskiä, että reaaliaikavaatimukset eivät välttämättä toteudu, koska käyttäjä ei voi hallinnoida julkisen pilven verkkoyhteyksien suorituskykyä. Siten pidentyneet ja odottamattomat viiveet saattavat aiheuttaa reaaliaikaiseen toimintaan ongelmia. Yksityisen pilven tapauksessa tilanne voi olla erilainen. Chen et al. [29] toteuttivat tutkimuksen yksityisen pilven soveltuvuudesta sähköverkon SCADA-järjestelmälle ZTE pilvialustalla. Tutkimuksessa todetaan, että kyseinen järjestelmä on riittävä sähköverkon operointiin ja osa järjestelmän suorituskyvyistä oli parempia kuin perinteisen järjestelmän.

Luvussa 4.1.1 esitellyssä tutkimuksessa [31] tutkittiin virtuaalikoneiden maantieteellisen sijainnin vaikutusta kommunikaatioviiveisiin. Tutkimuksessa käytettiin Modbus/TCP-protokollaa ja hyvin nopeita tietoliikenneyhteyksiä. Melbournen sisäisen verkon TCP-protokollan tiedoston lähetys- ja latausnopeudet olivat molemmat yli 3 Gbit/s ja Melbournesta Tasmaniaan vastaavasti 1,07 Gbit/s ja 1,52 Gbit/s. Näillä verkkonopeuksilla, SCADA-järjestelmän virtuaalikoneen ja mittalaitteiden simulaattorikoneiden maantieteellisen vä-

lin ollessa n. 430 km ja kiertokyselyvälin ollessa 1 ms maksimaalisen kuorman simuloinniksi, RTT yhdelle mittaukselle oli n. 11 ms. Tästä ajasta 4 ms kului pyynnön lähetykseen ja 7 ms datan lähetykseen. Haettavien mittausten määrän kasvaessa myös mittauskohtaisen RTT:n todettiin kasvavan. Esimerkiksi 200 mittalaitteella pyyntöön kului sama 4 ms, mutta lähetykseen 3630 ms. SCADA-järjestelmän komponenttien ja mittalaitteiden simulaattorien vienti samalle virtuaalialustalle tai samalle maantieteelliselle alueelle laski RTT:tä. Tästä voidaan päätellä verkkoyhteyden vaikutuksen olevan hyvin pieni verrattuna datan hakuun liittyvään prosessointiin. Tutkimuksen lopussa todetaan prosessointiin kuluvan suuren ajan johtuvan Modbus/TCP-protokollan toteutuksesta sekä suositellaan käytettävän tapahtumapohjaisia protokollia vähentämään dataliikennettä ja turhan toistuvan datan lähetystä.

Toisessa, samassa luvussa 4.1.1 esitellyssä tutkimuksessa [102] tutkittiin samankaltaisia asioita ja päädyttiin pitkälti samoihin tuloksiin. Ensimmäisessä vaiheessa tutkittiin pilveen virtualisoinnin vaikutusta fyysisiin servereihin verrattuna. Vaiheessa todettiin tiedonsiirtoviiveiden nousevan simuloitujen laitteiden määrän mukaan samalla tavalla kuin edeltävän kappaleen tutkimuksessa ja että virtualisoinnilla ei ole merkityksellistä vaikutusta viiveeseen. Toisessa vaiheessa tukimusta selvitettiin tapahtumapohjaisen protokollan IEC870-5-101 ja kyselypohjaisen Modbus/TCP-protokollan vaikutusta viiveeseen. Tapahtumapohjaisella protokollalla oli useita kymmeniä ms pienemmät viiveet ja huomattavasti pienemmät suurimmat poikkeamat viiveen vaihtelussa. Kolmannessa vaiheessa tutkittiin tieto- ja kyberturvallisuutta parantavan salauksen vaikutusta viiveeseen. SSL-salatun VPN-tunnelin käyttö järjestelmässä prosessiaseman ja pilven välillä ei lisää viivettä juuri yhtään ja sen vaikutusta voitaisiin pienentää entisestään käyttämällä erillistä virtuaalikonetta pelkästään salauksen toteuttamiseen. Tutkimuksessa todetaan maantieteellisen hajautuksen osalta, että 1000 km maantieteellinen matka prosessiaseman ja pilvipalvelun datakeskuksen välillä lisää tiedonsiirtoviivettä n. 30 ms. Lisäksi todetaan, että tiedonsiirtoa voidaan tehostaa jakamalla kenttälaitteet useammalle etäserverille.

Edeltävien tutkimusten pohjalta voidaan sanoa, että SCADA-komponenttien vienti pilveen ei aiheuta kriittisiä ongelmia suorituskykyyn. Pääosa viiveestä johtuu itse SCADA-järjestelmän toteutuksesta, jolloin tiedonsiirtoviiveet eivät ole merkityksellisiä järjestelmän toiminnan kannalta. Protokollan osalta on huomattavasti tehokkaampaa käyttää tapahtumapohjaisia protokollia vähentämään dataliikennettä ja poistamaan redundantin datan lähetys. Myöskään tieto- ja kyberturvan lisääminen salauksella ja VPN-tunneloinnilla ei aiheuttanut merkittäviä muutoksia viiveeseen. Pilven saatavuutta edistävä maantieteellinen hajautuskaan ei tuo järjestelmään merkittäviä viiveitä. Julkisen pilven datakeskusten osalta on olennaista huomauttaa, ettei Suomessa tällä hetkellä ole Azuren tai

AWS:n datakeskuksia. Lähimmät datakeskukset sijaitsevat Ruotsissa ja Keski-Euroopassa, jolloin maantieteellistä matkaa datakeskuksiin tulee väistämättä Suomessa sijaitsevilta prosessiasemilta.

Tutkimuksissa ei kuitenkaan oteta kantaa julkisen internetin mukanaan tuomiin ongelmiin. Julkista internettiä käytettäessä menetetään tietoliikenteen deterministisyys, johtuen verkon nopeuksien vaihteluista ja pahimmassa tapauksessa yhteyden katkeamisesta. Reaaliaikaisuus vaatii kaikkien järjestelmän osien olevan deterministisiä, jolloin julkista internettiä käytettävissä ei voida saavuttaa todellista reaaliaikaisuuttaa [9]. Vaikka edeltävien tutkimusten mukaan normaalitilanteessa viiveet eivät toiminnan kannalta aiheuta ongelmia, tulee järjestelmää suunniteltaessa ottaa huomioon mahdolliset normaalista toiminnasta poikkeavat tilanteet. Monitorointi- ja valvontaohjelmistot eivät ole kovin herkkiä viiveille tai sen vaihtelulle ja sietävät viivettä sekuntien tasolla, mutta esimerkiksi ylemmän tason ohjauksille deterministisyyden puuttuminen voi aiheuttaa ongelmia [105]. Yhteyksien liiallisen hidastumisen tai katkeamisen seurauksena ei saada menettää dataa, ja häiriötilanteessa prosessilaitoksen toiminnan ei tule välttämättä olla optimaalista, mutta sen tulee pysyä vähintäänkin turvallisena [54].

Oleellista ongelmatilanteisiin varautumisessa on pystyä monitoroimaan kentän ja pilven välisen verkkoliikenneyhteyden tilaa. PLC- ja RTU-yksiköille voidaan rakentaa datapuskuri, jolloin verkkokatkosten aikana data tallennetaan muistiin ja yhteyden palatessa siirretään pilveen. Lisäksi prosessin toimintalogiikka tulee rakentaa siten, että yhteyden katketessa prosessin toiminta pysyy turvallisena ja ylemmän tason viivästyneet ohjaukset eivät aiheuta ristiriitoja ohjauksissa.

5.2 Saatavuus ja luotettavuus

Saatavuus on SCADA-järjestelmän turvallisen ja luotettavan käytön kannalta yksi tärkeimmistä vaatimuksista. Pilvipalveluiden osalta kirjallisuustutkimuksen [97] mukaan saatavuus on pilvipalvelun toiseksi tärkein ominaisuus. Saatavuus on järjestelmän eitoiminnallinen vaatimus, joka määritellään prosenttiosuutena järjestelmän saatavilla olemasta ajasta. Tämä prosenttiosuus määrää hyväksyttävän kokonaiskatkosajan tietyllä ajanjaksolla. Korkea saatavuus (engl. *High availability, HA*) on tiukka 99.999 % saatavuusvaatimus, joka sallii enintään viisi minuuttia käyttökatkoja vuoden aikana sisältäen huollot- ja päivitykset. [71]

Useat pilvipalveluntarjoajat väittävät tarjoavansa hyvän saatavuuden palveluihinsa, mutta eivät silti yllä HA:n saatavuusvaatimukseen [72]. Pilvipalveluntarjoajat määrittele-

vät palvelutasosopimuksessa palvelun laadun sekä sitä rikkovat ehdot. Saatavuuden näkökulmasta palvelutasosopimuksessa tärkein elementti on katkosaika. Pilvipalvelujen välillä palvelutasosopimukset eroavat huomattavasti. Tarjoajilla on omat tapansa laskea katkosaika, jonka lisäksi joihinkin sopimuksiin sisältyy poikkeustapauksia, joista johtuvaa katkosta ei lasketa katkosajaksi. Esimerkiksi AWS EC2 poikkeustapauksiksi lukeutuvat internetin saatavuusongelmat ja jotkin ylläpitotoimet. [71]

Palvelutasojen välillä ei ole suurta eroa saatavuuden osalta, sillä kaikki palvelut pohjautuvat loppujen lopuksi palveluntarjoajan infrastruktuuriin ja saatavuutta parantaviin tekniikoihin. Palvelutasot kuitenkin rakentuvat kerroksittain, jolloin niiden saatavuudet ovat riippuvaisia alemmista kerroksista [71]. PaaS on riippuvainen IaaS:sta ja SaaS siten riippuvainen PaaS:sta. Mikäli siis IaaS-tason saatavuudessa on ongelmia, välittyy se usein myös muihin palvelutasoihin. Lisäksi PaaS-tason ohjelmiston laatu on täysin riippuvainen sen tarjoajasta. Mikäli tuote on huonosti tehty, voin sen kanssa ilmetä erilaisista ohjelmistovirheistä aiheutuvia saatavuuskatkoksia.

Käyttöönottomallien osalta saatavuuteen vaikuttaa lähinnä palveluntarjoaja. Suurilla teknologiayrityksillä kuten Amazonilla ja Microsoftilla järjestelmäinfrastruktuurin voidaan olettaa olevan huomattavasti luotettavampaa kuin muilla pienemmillä tarjoajilla suurien investointien vuoksi. Näin ollen niiden tarjoamien julkisten pilvipalveluiden ja yksityisen pilvien isännöintipalveluiden saatavuudet voidaan olettaa paremmiksi.

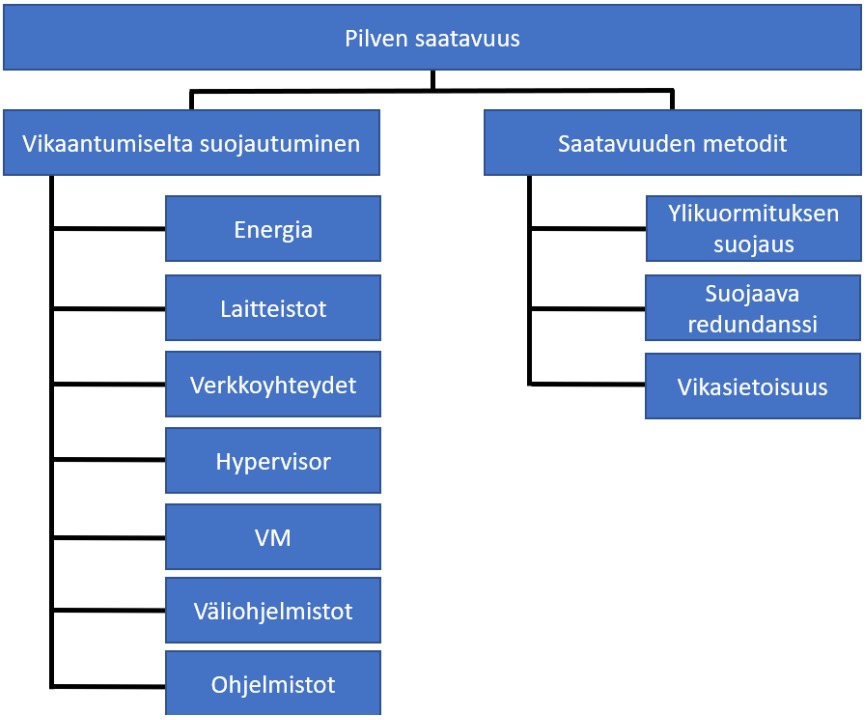
Pilvipalveluiden yksi tärkeimmistä eduista on siis korkean saatavuuden mahdollistaminen, mutta samalla se on myös yksi suurimmista pilvipalveluiden haasteista. Saatavuuden takaamiseksi palveluntarjoajat ovat investoineet paremmin suojattuihin datakeskuksiin, ja käytössä on nykyään myös useita erilaisia saatavuutta parantavia mekanismeja. Seuraavissa alaluvuissa käydään läpi näitä pilvipalveluiden saatavuutta parantavia mekanismeja ja verrataan niitä nykyisiin fyysisiin SCADA-järjestelmiin sekä käydään läpi AWS:n ja Azuren suurimpia käyttökatkoksia.

5.2.1 Pilvipalvelun saatavuus

Pilvipalveluissa voidaan usein soveltaa samoja periaatteita kuin fyysisissä järjestelmissä, ja ne ovat usein perinteistä järjestelmää paremmin suojattuja erilaisilta vioilta. Pilvipalvelun saatavuuteen vaikuttavat tekijät voidaan luokitella kahteen kategoriaan: vikoihin, joilta tulee suojautua sekä mekanismeihin, jotka parantavat saatavuutta. Kuvassa 5.1 on esitetty pilven saatavuuden luokittelu.

Vikaantumiselta suojautuminen fyysisellä tasolla näkyy pilvipalveluntarjoajien datakeskusten rakentamisessa. Pilvipalveluiden infrastruktuuri on rakennettu tavalla, jolla pyritään maksimoimaan saatavuus. AWS:n ja Azuren datakeskukset muodostavat tietyn maantieteellisen alueen sisällä toisistaan eristettyjä saatavuusvyöhykkeitä. Datakeskukset ovat yhdistetty toisiinsa redundanttisilla kuituyhteyksillä. Maantieteellisen alueen saatavuusvyöhykkeet eivät jaa keskenään samoja redundanttisia energianlähteitä ja verkkoyhteyksiä, vaan ne ovat kokonaan eristettyjä ja riippumattomia toisistaan. Saatavuusvyöhykkeisiin jakamisella estetään luonnonkatastrofien kuten ukkosmyrskyjen vaikutukset koko pilvipalvelun toimintaan.

Muita fyysisiä vikoja, joilta palveluntarjoaja voi suojautua ovat laitteistojen kuten muistien tai prosessorien vikaantumiset sekä verkkoyhteyskomponenttien kuten verkkokytkinten, reitittimien tai yhteyskaapelien vikaantumiset. Näihin pilvipalveluntarjoajat käyttävät usein redundanttisia komponentteja ja metodeja, joita käytetään myös perinteisissä SCADA-järjestelmissä takaaman parempi saatavuus. Fyysisten vikaantumisien lisäksi pilvipalveluntarjoajat voivat tarjota vikaantumisen estoa hypervisorille, virtuaalikoneille, väliohjelmistoille ja muille ohjelmistoille. Suojautumisen taso riippuu pilvipalvelussa käytetystä palvelutasosta. Esimerkiksi IaaS-tason palvelun eivät voi välittää ohjelmistotason vikaantumisista, mutta voivat tarjota vikaantumisen estoa hypervisorille. Kuvan 5.1 vikaantumiselta suojautumista AWS ja Azure tarjoavat kaikilla fyysisillä, hypervisor sekä osittain virtuaalikoneiden tasolla. [71]



Kuva 5.1. *Pilven saatavuuden luokittelu. Mukaillen [71].*

Saatavuuden mekanismeista osa on käyttäjän hallittavissa ja osa sisältyy automaattisesti palvelun käyttöön. Vikasietoisuudessa pyritään ylläpitämään järjestelmän normaali toiminta vian ilmaantuessa. Vika tulee ensin havaita, jonka jälkeen pyritään palauttamaan järjestelmä takaisin normaaliin toimintaan. Palauttavina toimina pilvipalveluissa ja yleisesti tietojärjestelmissä voidaan käyttää kolmea erilaista metodia [91]. Automaattisessa vaihdossa varajärjestelmään vikaantuneen osan työkuorma siirretään toimivalle redundanttiselle järjestelmälle. Uudelleen käynnistyksessä vikaantunut elementti pysäytetään ja käynnistetään uudestaan. Tilan palautuksessa (engl. *rollback*) järjestelmän tila palautetaan viimeisimpään tiedossa olevaan virheettömään tilaan. Tilan palautus vaatii järjestelmän tilojen ajoittaisen tallennuksen. Näistä kolmesta vaihtoehdosta AWS käyttää vain automaattista varajärjestelmään vaihtoa ja Azure sen lisäksi myös uudelleen käynnistystä. [71]

Komponenteissa, joiden päälle pilvipalvelut rakentuvat on erilaisia kapasiteettirajoitteita ja näitä rajoitteita rikottaessa voidaan aiheuttaa järjestelmään erilaisia vikatiloja. Suuri ero fyysisen ja pilvipohjaisen SCADA-järjestelmän välillä on automaattinen skaalautuvuus. Pilvipohjaisessa järjestelmässä on käytössä automaattinen resurssien skaalautuminen käyttäjämäärien ja resurssitarpeiden vaihdellessa, jolloin ei muodostu kapasiteettirajoitteita. Fyysistä järjestelmää ei voida yhtä helposti skaalata suuremmaksi, jolloin äkillinen tarpeen muutos voi aiheuttaa ongelmia. Tähän voidaan fyysisissä järjestelmissä vastata mitoittamalla järjestelmien resurssit yli normaalin tarpeen, jolloin järjestelmän käyttö ei ole optimaalista. Automaattisen skaalauksen lisäksi pilvipalvelut voivat tarjota kuormituksen jakamista palvelun tuottoon osallistuvien redundanttisten virtuaaliservereiden kesken. Molempia näistä metodeista käytetään AWS:n ja Azuren palveluissa [71].

Viimeinen saatavuutta parantava metodi on ohjelmistopohjainen redundanssi. Fyysisten komponenttien redundanssin lisäksi voidaan käyttää ohjelmistopohjaista redundanssia ohjelmistoon liittyvien vikojen vaikutusten minimointiin. Pilvipohjaisessa redundanssissa voidaan käyttää samoja redundanssin keinoja kuin mitä luvussa 2.8 on esitelty ja mitä luvussa 4 esitellyissä esimerkeissä on käytetty. Pilven mahdollistama redundanssi on yksi sen suurista eduista verrattuna fyysiseen. Toisin kuin fyysinen redundanssi, pilveen luotu ohjelmallinen redundanssi ei vaadi käyttäjältä investointeja useisiin fyysisiin servereihin, jolloin sen toteutus on huomattavasti halvempaa [44]. Nykyisissä SCADA-toteutuksissa on mahdollista jakaa serverikoneen resurssit hypervisorilla virtuaalikoneiden käyttöön, ja hajauttaa SCADA:n komponentit näille virtuaalikoneille. Näillä virtuaalikoneilla voidaan myös luoda redundanttisia ohjelmistokomponentteja. Tässä kuitenkin ongelmaksi muodostuu kaiken toiminnallisuuden jättäminen yhden isäntäkoneen vastuulle.

Pilvessä puolestaan redundanttiset komponentit eivät ole saman fyysisen laitteen varassa. Lisäksi pilvipalvelut mahdollistavat virtuaalikoneiden maantieteellisen hajautuksen. Maantieteellisessä hajautuksessa käyttäjä voi päättää virtuaalikoneille käytettävät datakeskukset, jolloin toiminnan voi jakaa usealle eri saatavuusvyöhykkeelle. Tällä tavoin hajautetussa järjestelmässä yhden vyöhykkeen kaatuessa koko SCADA-järjestelmä ei lamaannu, koska sen redundanttiset komponentit sijaitsevat toisella vyöhykkeellä. AWS ja Azure molemmat mahdollistavat maantieteellisen hajautuksen [71]. Xiwei Xu et. al. [99] esittävät saatavuuden paranevan yli 6 % käytettäessä kahta saatavuusvyöhykettä yhden sijaan. Tutkimuksen simulaatiossa on käytetty vuoden 2013 AWS:n EC2 palvelutasosopimusta, joten saatavuus ei tänä päivänä välttämättä enää parane yhtä paljoa.

Moni edellä esitellyistä saatavuutta parantavista metodeista on jo käytössä markkinoilla olevissa perinteisissä järjestelmissä. Esimerkiksi Siemensin WinCC:ssä on mahdollista luoda ohjelmiston sisäisiä redundanttisia servereitä ja käyttää näiden välillä sisäistä kuormanjakoa. Mikäli pilvipalveluun migraatiossa käytetään ”lift and shift” -metodia tällaisella kaupallisella palvelulla ja jaetaan serverit eri datakeskuksiin virtuaalikoneille, jää kysymykseksi toimivatko ohjelmiston sisäiset metodit oikein. Varsinkin kommunikaation reititys voi tuottaa haasteita.

5.2.2 AWS- ja Azure-palveluiden käyttökatkokset

Vaikka AWS:n ja Azuren pilvi-infrastruktuuri on rakennettu hyvin kestäväksi ja sillä on pyritty mahdollistamaan korkea saatavuus, on molemmilla silti usein eri mittaisia käyttökatkoksia erinäisistä syistä. Vuoden 2018 tutkimuksessa [82] on raportoitu vuosittaisten pilvipalveluiden käyttökatkoksista johtuvien tappioiden olevan jopa lähes 300 milj. dollaria noin 99.91 % saatavuudella. Taulukossa 5.1 on esitetty AWS:n ja MS Azuren viime vuosien merkittävimpiä käyttökatkoksia ja palvelun laatuun vaikuttaneita ongelmia. Palveluntarjoajat eivät ole velvoitettuja kommunikoimaan katkoksista, jolloin osalle katkoksista ei ole palveluntarjoajan omaa näkökulmaa saatavissa [72]. On siis oleellista huomauttaa, että käyttökatkoksia ja palvelunlaatuun vaikuttavia ongelmia on ollut huomattavasti enemmän ja useammin kuin mitä taulukko esittää, ja niiden luotettavuus pohjautuu osittain kolmansien osapuolien raportointiin.

Taulukko 5.1. *MS Azuren ja AWS:n viime vuosien käyttökatkoksia.*

Vuosi	Palvelun-tarjoaja	Kesto	Tapahtuma
2015	AWS	5 h	US-EAST-1 saatavuusalueella sähkökatkos ja riittämättömät korjaustoimenpiteet. Vaikutti usean palvelun toimintaan. [17]
2016	AWS	4 h	AWS Sydney pois toiminnasta ukkosesta johtuvan sähkökatkon vuoksi [17].
2017	AWS	5 h	Työntekijän väärästä parametrin syötöstä johtuva S3 käyttökatkos. Vaikutti useaan palveluun US-EAST-1 alueella. [17]
2018	AWS	4 h	Yhteysongelmia, jotka estivät asiakkaiden pääsyn EC2 virtuaalikoneinstansseihin [17].
2018	Azure	11 h	Pohjois-Euroopan alueen käyttökatkos johtuen viilennysjärjestelmän ongelmasta [22].
2018	Azure	25–72 h	Jäähdytysjärjestelmän riittämättömästä ylijännitesuojasta johtuva usean käyttöalueen ja palvelun laajuinen käyttökatkos [22].
2019	AWS	6 h	Ohjaus- ja jäähdytysjärjestelmän vikaantumisesta johtuva Tokyon EC2 serverien käyttökatkos [17].
2019	AWS	8 h	Sähkökatkos ja varageneraattorien hajoaminen. Vaikutti EC2 instanssien toimintaan ja sähkösyötön palautuessa osa EBS-palveluun tallennetuista EC2 tiedostojärjestelmistä oli pysyvästi kadotettu. [47]
2019	AWS	8 h	DDoS-hyökkäyksestä johtuvat DNS-ratkaisuvirheet reitityksessä [6].
2021	Azure	16 h	Todennusongelmia Azure Active Directory -ohjelmistoa käyttävien sovellusten kanssa [22].
2021	AWS	1 h	Sähkökatkosta johtuva käyttökatkos ja osittainen datanmenetys [17].
2021	Azure	2 h	Globaali DNS-ongelma. Vaikutti useaan Azuren palveluun. [22]

Taulukosta 5.1 voidaan tulkita, että käyttökatkosten aiheuttajia on monia erilaisia ja niiden kestot vaihtelet tunneista päiviin. Suuri osa taulukossa esitetyistä käyttökatkoksista johtuu sähkökatkoksista, jotka puolestaan johtuvat usein luonnonilmiöistä kuten ukkosmyrskyistä tai tulvista. Sähkökatkojen lisäksi käyttökatkoja aiheuttavat fyysisten laitteistojen kuten viilennysjärjestelmien vikaantumiset, ihmisten tekemät inhimilliset virheet sekä palvelunestohyökkäykset. Yleensä näistä vikatilanteista palvelu palautuu normaalitilaan vian paikannuksen jälkeen nopeasti, mutta toisinaan palvelunlaatu pysyy heikompana jopa päiviä. Taulukon ehkä merkittävin rivi on AWS:n 2019 sähkökatkosta johtunut käyttökatkos, jonka seurauksena osaa ESB-palveluun talletetusta datasta ei pystytty palauttamaan.

5.3 Tieto- ja kyberturvallisuus

Ensimmäisen ja toisen sukupolven SCADA-järjestelmät on rakennettu puhtaasti toiminnallisesta näkökulmasta sivuuttaen täysin tieto- ja kyberturvalliset ominaisuudet. Seuraavien sukupolvien keskittyessä avoimiin standardeihin ja arkkitehtuureihin sekä verkkoteknologioiden käytön yleistymisen myötä, ovat SCADA-ympäristöt omaksuneet perinteisen IT-ympäristön ominaisuuksia, mukaan lukien kääntöpuolena niiden tuomat turvallisuusuhat. Viimevuosina SCADA-järjestelmiin kohdistuneet hyökkäykset ovat yleistyneet ja niiden myötä myös tutkimus turvallisista SCADA-järjestelmistä on kiihtynyt. [81]

Pilveen liittyvät turvallisuusuhat voidaan jaotella palveluntarjoajien ja käyttäjän kohtaamiin ongelmiin ja vastuu tieto- ja kyberturvallisuuden toteutumisesta on vastaavasti jaettu näiden kesken. Palveluntarjoajan vastuulla on varmistaa oman infrastruktuurin turvallisuus ja että asiakkaan ohjelmistot ja data on suojattu. Asiakkaan vastuulle puolestaan jää oman ohjelmiston tietoturvan parantaminen, vahvojen salasanojen käyttö ja tunnistautuminen. Reaaliaikainen datan siirto julkisen internetin yli ja etäyhteydet järjestelmään tekevät pilvipohjaisesta SCADA-järjestelmästä alttiimman kyberhyökkäyksille ja luovat useita aukkoja, joiden kautta järjestelmään voidaan injektoida viruksia tai vakoiluohjelmia [3]. Toisaalta pilvipalvelut mahdollistavat myös useiden erilaisten tietoturvaa lisäävien palveluiden käytön ja palveluntarjoajien omat tieto- ja kyberturvajärjestelyt ovat perinteisiin järjestelmiin verrattuna paremmat, koska palveluntarjoajilla on omat työntekijänsä pelkästään järjestelmän turvallisuudesta huolehtimiseen. Seuraavissa alaluvissa käydään läpi yleisesti pilvipalveluiden tieto- ja kyberturvahaasteet sekä SCADA-järjestelmän aiheuttamat haasteet pilveen integroitaessa. Lisäksi pohditaan mahdollisia ratkaisuja näihin ongelmiin.

5.3.1 Tieto- ja kyberturvahaasteet pilvessä

Verrattuna perinteisiin laskenta- ja tietoverkkojärjestelmiin, pilvipalveluiden käyttö itsessään tuo useita haavoittuvuuksia järjestelmään. Näihin haavoittuvuuksiin kuuluvat [63]: 1) Toisten asiakkaiden hyökkäykset. 2) Jaetun teknologian tuomat ongelmat. 3) Häiriöt palveluntarjoajan tai asiakkaan turvajärjestelmissä. 4) Epäonnistunut palveluntarjoajan ja asiakkaan tieto- ja kyberturvajärjestelmien integraatio. 5) Epävarmat sovellusten ohjelmointirajapinnat. 6) Datan menetykset ja vuodot. 7) Tilin tai palvelun kaappaus.

Haavoittuvuudet riippuvat osittain käytetystä palvelutasosta. IaaS-tason palvelut ovat alttiitta kaikille samoille uhkille kuin perinteiset informaatio- ja kommunikointiympäristöt

[28]. PaaS-tason palvelut puolestaan ovat herkkiä jaettujen resurssien takia, koska tietoturva-asetukset saattavat vaihdella resurssien välillä. Jaetut resurssit ovat myös alttiita datavuodoille. Lisäksi käyttäjäobjektien suojaus on yksi vakavimmista PaaS-tason ongelmista [79]. SaaS-tason palvelut ovat käytössä selaimen ja internetyhteyden välityksellä, jolloin ne ovat alttiita samoille turvallisuusongelmille kuin perinteiset verkkopalvelut. SaaS-tasolla datan tietoturvallisuuteen panostaminen on siten olennaista, varsinkin sen luottamuksellisuuden osalta. Muita tietoturvallisuuden ongelmia ovat datan varmuuskopiot, dataan pääsy, tallennusten sijainti sekä todennus [84].

Edellisten kohtien lisäksi pilvipohjaisesta SCADA-järjestelmästä tekee mahdollisesti perinteistä SCADA-järjestelmää alttiimman erilaisille hyökkäyksille neljä eri tekijää: 1) Infrastruktuurin jakaminen tuntemattomien käyttäjien kanssa luo useita uhkia järjestelmälle [84]. 2) Verkkoyhteydet SCADA-järjestelmän osien ja pilven välillä mahdollisesti lisäävät riskiä koko järjestelmän vaarantumiseen ulkopuolisen hyökkääjän toimesta [78]. 3) Pääosasta SCADA-spesifisistä sovelluskerroksen protokollista puuttuu tarvittavat tietoturvalliset ominaisuudet. Esimerkiksi Modbus ja DNP3 eivät tue todennusta ja salausta [78]. 4) Kaupallisille SCADA-järjestelmille suunnittelemattomien suojaratkaisuiden käyttö voi lisätä kyberturvariskiä [78].

Jaetun infrastruktuurin vuoksi järjestelmät ovat pilvessä alttiimpia kyberturvallisuusuhille kuten komento/vastaus injektioille sekä DoS/DDoS (engl. *Distributed Denial of Service*) - ja MITM (engl. *Man In The Middle*) -hyökkäyksille [84]. Perinteiset SCADA-järjestelmät on rakennettu käyttämään suljettuja yksityisiä verkkoja, ja internetin käyttämättömyys on ollut lähinnä turvamekanismi. Yhdistettäessä järjestelmä pilveen, yhdistetään järjestelmä samalla monimutkaiseen tietoliikenneympäristöön, joka tuo mukanaan uusia haavoittuvuuksia. Pilveen yhdistäminen luo samalla uusia yhteyspisteitä pahantahtoisille osapuolille. IoT-laitteiden kuten erilaisten antureiden lisääntyminen teollisuudessa vaikeuttaa tilannetta entisestään, koska jokainen internettiin yhdistetty laite luo uuden yhteyspisteen järjestelmään ja siten nostaa riskin tasoa. Järjestelmän kannalta suurimman riskin aiheuttavat pahantahtoiset henkilöt, joilla on pääsy järjestelmään syystä tai toisesta. Tällaisia henkilöitä voivat olla esimerkiksi vanhat työntekijät, järjestelmän hallitsijat tai pilvipalveluntarjoajan henkilökunta. Pahantahtoisten henkilöiden pääsy järjestelmään voi aiheuttaa useita uhkia kuten tietomurtoja tai SCADA-järjestelmän tapauksessa pahimmillaan luvattoman järjestelmänohjauksen.

Teollisuuden protokollat ovat SCADA-järjestelmien tavoin suunniteltu käytettäväksi suljetussa yksityisessä verkossa, jolloin niiden tietoturvallisiin ominaisuuksiin ei ole panostettu tai ne ovat riittämättömiä. Tällöin esimerkiksi protokollien heikot todennukset ja salaukset mahdollistavat helpon pääsyn IP-osoitteisiin, käyttäjänimiin ja salasanoihin. [101]

Pilvipohjaisia SCADA-järjestelmiä vastaan voi kohdistua pääasiassa kolmea erilaista tieto- tai kyberturvaa vaarantavaa hyökkäystä. DoS-hyökkäyksen tarkoitus on tehdä järjestelmästä saavuttamaton tarkoitetuille käyttäjille. DoS-hyökkäys voidaan toteuttaa ”hukuttamalla” järjestelmä tietoliikenteeseen tai lähettämällä spesifiä dataa, joka aiheuttaa järjestelmän kaatumisen. Hyökkäyksillä pystytään siis vaikuttamaan järjestelmän saatavuuteen, joten kyberturvallisuuden ominaisuudet ovat myös saatavuuden kannalta olennaisia. Toinen mahdollinen hyökkäys on MITM-hyökkäys, jossa hyökkääjä asettaa itsensä kommunikaatiokanavalle pilven käyttäjän ja palvelun väliin. MITM-hyökkäys voidaan toteuttaa esittämällä toista käyttäjää järjestelmään pääsyn saamiseksi, tai sieppaamalla ja monitoroimalla lähetettyjä datapaketteja, joiden kautta hyökkääjä voi saada haltuunsa sensitiivistä informaatiota kuten salasanoja. Kolmas mahdollinen hyökkäys on APT (engl. *Advanced Persistent Threat*), jossa hyökkääjä onnistuu pääsemään havaitsemattomasti järjestelmän sisään ja keräämään sensitiivistä dataa järjestelmästä ajan kuluessa. [3]

Vaikka pilvipalvelut ovat rakennettu mahdollisimman luotettaviksi ja turvallisiksi käyttää, on niiden käytössä kuitenkin aina pieni riski ongelmille. Taulukon 5.1 vuoden 2019 AWS:n datakeskuksen sähkökatkon seurauksena osa käyttäjistä menetti lopullisesti osan palveluun tallennetusta datasta. Tämä osaltaan korostaa, ettei pilvipalveluiden tietoturvan kannalta ainoita uhkia ole ulkoiset uhat, jolloin oman järjestelmän tietoturvallisuuteen kannattaa panostaa.

5.3.2 Tieto- ja kyberturvahaasteisiin varautuminen

Luvun 5.3.1 perusteella voidaan todeta SCADA-järjestelmän pilvi-integraation kasvattavan tietoturvariskejä huomattavasti, ja että uhkia on monenlaisia, jolloin järjestelmä on hankala turvata kaikkien mahdollisten uhkien varalta. Riskeiltä varautuminen riippuu myös käytetystä pilven käyttöönottomallista, palveluntarjoajasta ja palvelutasosta.

Tutkimuksen [4] mukaan julkista pilvipalvelua käytettäessä on oleellista vastata haasteisiin informaation ja ohjausten kommunikaatiossa, tallennustilan ja laskennallisten resurssien jakamisessa sekä fyysisten datakeskusten infrastruktuurissa. Informaation ja ohjausten kommunikaation turvallisuus riippuu tavasta, jolla pilvipohjainen SCADA on erotettu ohjauslaitteistoista. Toisin sanoen on oleellista, ettei kriittistä ohjausinfrastruktuuria altisteta internetille. Alistaminen voidaan estä esimerkiksi käyttämällä luvun 4.1.2 järjestelmää, jossa paikallinen SCADA hoitaa kriittiset toimet. Lisäksi voidaan käyttää datan siirtämiseen ”push”-tekniikkaa ”pull”-tekniikan sijaan. Jaetun infrastruktuurin osalta on olennaista tietää, kuinka pilvipalveluntarjoaja hallitsee eri ohjelmistojen käytössä olevia

laskennallisia resursseja. Viimeinen kohta fyysisestä infrastruktuurista liittyy fyysisten datakeskusten tietoturvaan. Käyttäjän tulee varmistaa, että palveluntarjoajan serverit ovat tietoturvallisesti oikeellisia sekä niiden kommunikaatioväylät ovat turvattuja.

Pilvipalveluntarjoajaa valittaessa tulee ottaa huomioon seuraavat tärkeimmät kriteerit: 1) käyttäjän turvallinen pääsy palveluun, 2) eri ohjelmistojen tuottaman datan eristäminen toisistaan, 3) käyttäjän hallinnointimahdollisuudet palveluntarjoajan infrastruktuurin muuttuessa, 4) datan salaus, 5) ohjelmistopäivitysten automatisointi, 6) jatkuva reaaliaikainen tapahtumien monitorointi ja analysointi, 7) mahdollisuus luoda lokitiedostoja ja niiden pohjalta havaita tunkeilijoita sekä vastata niiden hyökkäyksiin, ja 8) valmius ryhtyä välittömästi korjaaviin toimenpiteisiin havaittujen haavoittuvuuksien suhteen. Tehokkain tapa varmistua edellä mainittujen ehtojen täyttymisestä on solmia vain palvelutasosopimuksia (SLA), jossa kriteerit ovat selkeästi esitetty ja täytetty. Suurien palveluntarjoajien kuten AWS:n ja Azuren kohdalla voidaan pitkälti olettaa ehtojen täyttyvän, mutta pienempiä toimijoita käytettäessä voi kriteereissä olla mahdollisia puutteita. [84]

Julkista pilveä varmempi tapa varmistua SCADA-järjestelmän turvallisuudesta on käyttää yksityistä oman hallinnonalaista pilveä, jossa tieto- ja kyberturvallisuuden ominaisuudet ovat omistajan vastuulla. Yksityisessä pilvessä on suositeltavaa käyttää usealle tasolle levittyvää turva-arkkitehtuuria, joka minimoi yhden tason vikaantumisen vaikutuksen muihin tasoihin [85]. Yksityistä pilveä käytettäessä vältetään myös jaettuun infrastruktuurin liittyvät ongelmat. Infrastruktuurin jaon ongelmat voidaan välttää myös käyttämällä esimerkiksi Azuren Dedicated Host -palvelua, jossa resurssit eivät ole jaettu muiden käyttäjien kanssa.

S. Alam et. al. [1] mukaan tunkeutumisen tunnistusjärjestelmä (engl. *IDS, Intrusion Detection System*) ja tunkeutumisen estojärjestelmä (engl. *IPS, Intrustion Prevention System*) ovat oleellisia komponentteja kyberhyökkäyksien torjunnassa. IDS on järjestelmä, joka tunnistaa mahdollisen haitallisen toiminnan ja IPS estää tällaiset toiminnot. Molempia järjestelmistä pidetään suojatoimenpiteinä. IDS prosessoi yleensä raakaa verkkoliikennettä OSI-mallin verkkokerroksella, jonka toimintaan pilvessä ei käyttäjä voi juurikaan vaikuttaa. Tällöin IDS:n toteutus on täysin riippuvainen palveluntarjoajasta. Pilviympäristössä käytetään yleisesti kahta eri IDS/IPS yhdistelmää. Isäntäpohjainen järjestelmä HIDS/HIPS (engl. *Host-based IDS/IPS*) monitoroi, analysoi ja ehkäisee poikkeavuuksia isäntäkoneilta kerätyssä datassa. Tiedot kerätään tiedostojärjestelmistä, tietokannoista ja verkon analysoinnista. Havaitessa poikkeavuuksia estomekanismina käytetään hälytystä. Verkkopohjainen NIDS/NIPS (engl. *Network-based IDS/IPS*) puolestaan etsii haitallisia paketteja monitoroimalla verkkoliikennettä. Hyökkäyksen havaitessa järjestelmä ilmoittaa järjestelmänvalvojalle hyökkäyksestä ja estää lähteen IP-osoitteen pääsyn

verkkoon. Pilvipohjaisten SCADA-järjestelmien pääasialliset kyberhyökkäysuhat liittyvät lähinnä verkkoyhteyksiin, jolloin NIDS/NIPS järjestelmää voidaan pitää soveltuvampana estotapana. Tunkeutumisen estämisen lisäksi järjestelmä voidaan rakentaa luvun 4.2 SKYDA-järjestelmän redundanssiarkkitehtuurin mukaisesti sietämään järjestelmän osittaisen tunkeutumisen.

SCADA-protokollien heikkoa tietoturvaa voidaan parantaa käyttämällä verkkoyhteyksissä VPN (engl. *Virtual Private Network*) -tunnelointia, joka tarkoittaa yksityisen virtuaaliverkon muodostamista kahden laitteen välille. Sille ominaista on reitittyminen julkisen internetin yli ja vahva salaus. VPN-tunneloinnin salausominaisuuksien vuoksi tietoliikenteestä saadaan tietoturvallista, vaikka sen sisällä käytettävät protokollat eivät itsessään olisi salattuja. Luvun 4.1.1 tutkimuksessa [102] todettiin, ettei SSL-salatun VPN-tunnelin käyttö lokaalin etäaseman ja pilven välillä aiheuta juurikaan tiedonsiirtoviivettä.

Historia- ja virtuaalikonedatan turvaamiseen tulee myös panostaa. Pilvipalveluilla datan tallennustila on dynaamista ja sinne tallentaminen on tehty helpoksi. Datan katoamisen tai korruptoitumisen varalta on hyvä pitää mahdollisesti eri datakeskukseen sijoitettua varmuuskopiota historiatietokannasta ja virtuaalikoneista. Virtuaalikoneiden useat hypervisor-ohjelmistot mahdollistavat tilannekatsausten (engl. *snapshot*) ajoittaisen ottamisen, jotka voidaan automaattisesti tallettaa pilvipalveluun.

Eri palvelutasoilla käyttäjän mahdollisuus vaikuttaa tietoturvaan vaihtelee. SaaS-tasolla abstraktiotaso on korkeimmillaan, jolloin käyttäjällä ei juurikaan ole mahdollisuuksia vaikuttaa käytetyn palvelun turvallisuuteen. SaaS-tason palvelua käytettäessä koko tietoturva on käytännössä ohjelmisto- ja pilvi-infrastruktuurintarjoajan vastuulla. Abstraktiotason laskiessa mahdollisuus vaikuttaa tietoturvaan nousee. PaaS-tasolla käyttäjä on vastuussa ohjelmistojen suojauksesta, datasta ja käyttäjien pääsystä alustalle, jolloin palveluntarjoajalle jää vain käyttöjärjestelmän ja infrastruktuurin suojaus. Luonnollisesti IaaS-tasolla on käyttäjällä suurin vastuu tietoturvan hallinnasta. Julkisia pilvipalveluita käytettäessä on palveluntarjoajilla kattava kokonaisuus erilaisia tietoturvaa parantavia palveluita. Esimerkiksi AWS tarjoaa palveluita identiteetin- ja pääsynhallintaan, erilaisia raportointityökaluja, palomuureja ja useita muita turvallisuuttaa edistäviä palveluita.

5.4 Legacy-järjestelmien vaikutus migraatioon

Automaatiojärjestelmien pitkien elinkaarien vuoksi järjestelmiä päivitetään hyvin hitaasti, jolloin päivitysten välissä vanhalla teknologialla tehdyt laiteet ja ohjelmistot ovat voineet jäädä kehityksessä uudempia huomattavasti jälkeen. Päivitysten yhteydessä ei aina

voida tai haluta päivittää koko järjestelmää esimerkiksi kustannussyistä, jolloin osa vanhalla teknologialla rakennetuista komponenteista jää osaksi päivitettyä järjestelmää. Esimerkiksi PLC-komponenttien käyttöikä voi ylittää 20 vuotta. Tällaiset käytössä olevat vanhanaikaiset legacy-järjestelmät ja -komponentit eivät välttämättä ole teknologialtaan yhteensopivia uudempien järjestelmien kanssa ja siten niiden integrointi uusiin järjestelmiin voi olla hankalaa.

SCADA-järjestelmät saattavat sisältää kriittisiä Legacy-järjestelmät -ja komponentteja, jolloin pilvimigraatiossa tulee ottaa huomioon niiden vaikutukset. Tämänhetkisiä legacy-järjestelmiä ei ole rakennettu mukautumaan teollisuuden neljännen vallankumouksen visioon, ja vanhempien sukupolvien järjestelmien mukaisesti niistä puuttuvat edistyneet, ja jopa peruslaatuiset tieto- ja kyberturvalliset ominaisuudet. Legacy-järjestelmien käyttöä pidetään perimmäisenä syynä lähimenneisyyden kyberhyökkäysten kuten Ukrainan vuoden 2015 ”Black Energy 3”-hyökkäyksen mahdollistamiseen [60]. Vastaavasti esimerkiksi sähkönjakelun laitteistoissa ei ole kattavaa tietoturvasuojaa, joka tulee ottaa huomioon järjestelmiä modernisoitaessa [59].

R Khan [59] et. al. esittävät erään ratkaisun legacy-järjestelmien mutkattomaan pilvi-integraatioon. Tutkimuksessa esitetään, että kaikki SCADA-järjestelmän ohjelmistokomponentit viedään pilveen kuten muissakin tässä työssä esitellyissä tutkimuksissa, jonka lisäksi jokaisen pilveen kommunikoivan laitteen kommunikaatiokanavaan asennetaan lisäkomponentti. Tämä lisäkomponentti tarjoaa palomuurin, kytkimen ominaisuudet sekä yhteyden monitoroinnin, jolla voidaan varmistaa paikallisen yhteyden turvallisuus laitteelle päin. Välilaite yhdistetään pilvipalveluun salattua VPN-tunnelia pitkin, ja estää sen avulla ohjauksen muualta kuin paikallisesti verkosta. VPN-tunnelien yhteyksiä hallitaan pilvipalvelussa, joihin esimerkiksi Azure ja AWS molemmat tarjoavat palveluita.

Turvallisuuden ominaisuuksia lisättäessä tulee ottaa huomioon niiden vaikutus tietoliikenteeseen. Erilaiset protokollamuunnokset, palomuurit ja välikappaleiden lisäykset lisäävät tietoliikenneviiveitä, jotka voivat aiheuttaa ongelmia reaaliaikajärjestelmässä. Vastaavasti legacy SCADA-järjestelmiä käytettäessä pitää varmistua, ettei itse vanhalla teknologialla toteutettu ohjelmisto aiheuta ylimääräisiä viiveitä hitaamman prosessoinnin takia [31].

5.5 Yhteenveto toteumista

Tutkimusten perusteella reaaliaikavaatimusten ei pitäisi muodostua ongelmaksi. Viiveet tiedonsiirrossa eivät ole pitkiä, ja viiveestä suurin osa johtui käytetystä protokollasta ja

prosessoinnista. Reaaliaikavaatimusten osalta tulee kuitenkin muistaa, että järjestelmään lisättävät osat, esimerkiksi tietoturvalliset ominaisuudet saattavat pahentaa viivettä. Tutkimuksessa kuitenkin todettiin, että esimerkiksi salaus ja VPN-tunnelointi eivät juurikaan vaikuttaneet viiveeseen. Pilvipalveluita käytettäessä järjestelmän tiedonsiirto on julkisen tietoliikenteen vaihteluiden alainen, jolloin voi esiintyä hetkellisiä palvelunlaadun heikkenemisiä, jotka johtavat pidentyneisiin viiveisiin. Lisäksi pitää huomioida legacy-järjestelmistä ja -komponenteista aiheutuvat viiveet sekä julkisten verkkoyhteyksien mukanaan tuomat ongelmat.

Saatavuuden ja luotettavuuden osalta pilvipalvelut eivät vielä täytä korkea saatavuuden vaatimuksia. Suurimmillakin palveluntarjoajilla on taulukon 5.1 mukaan useita käyttökatkoksia vuosittain, joiden kestot vaihtelevat tunneista päiviin. Näistä kuitenkin suuri osa vaikuttaa vain tiettyyn pieneen osaan palveluita ja palveluntarjoajan sisäisen redundanssin avulla vikaantuneiden palvelimien ohjelmistot voidaan hetkellisesti siirtää ajoon toisille palvelimille. Pilvipalveluissa saatavuuden parantamiseen on olemassa useita eri keinoja sekä palveluntarjoajan, että käyttäjän toimesta mahdollistaen esimerkiksi maantieteellisen hajautuksen.

Tietoturvan näkökulmasta pilviympäristön tuominen osaksi SCADA-järjestelmää luo täysin uusia haasteita järjestelmän turvaamiselle. SCADA-järjestelmien suunniteltu käyttö suljetuissa sisäverkoissa, käytettyjen protokollien alkeellinen tietoturva ja internetin käytön myötä lisääntyneet yhteyspisteet järjestelmään luovat haasteita pilvimigraation turvallisuuteen. Pilviympäristö on kuitenkin infrastruktuurin ja käytetyn palvelutason puolesta pääosin turvallinen. Lisähaasteita tietoturvalle tuovat myös legacy-järjestelmien ja -komponenttien käyttö.

Tutkimuksissa SCADA:n viennistä pilvipalveluun käytetyt case-esimerkit ovat pieniä laitoksia ja olemassa olevien SaaS-tason palvelujen esimerkit ovat myös pääasiaa pieniä hajautettuja pumppaamoita tai muita vedenkäsittelyn ja -jakelun laitoksia. Isojen laitosten viennistä ei siis ole olemassa kirjallisuutta tai case-esimerkkejä, joten toimintaa realistisissa isoissa laitoksissa on vaikea arvioida. Suurissa laitoksissa liikkuvan datan määrä lisääntyy nopeasti, joka voi vaikuttaa reaaliaikaominaisuuksiin. Saatavuuden lyhyidenkin katkosten, tiedonsiirtoviiveiden kyseenalaisuuden ja tietoturvapuutosten vuoksi pilviympäristö ei tällaisenaan vielä ole kypsä kriittisten laitosten SCADA:n pilveen vientiin. Katkokset palvelussa tai palvelunlaadussa voivat aiheuttaa vaaratilanteita ja taloudellisia tappioita järjestelmän hallinnan menetyksissä.

Mahdollisesti teknologiaa voisi tällaisenaan hyödyntää ja kehittää etävalvomoissa, joissa kuvan 4.2 mukaisesti on myös paikallinen SCADA-järjestelmä. Kuvasta poiketen paikallisen järjestelmän tulisi olla kokonaan irrallinen pilvipalvelusta, jolloin pilven tai yhteyksien vikaantuessa paikallinen SCADA ei olisi millään tavalla riippuvainen pilvipalvelusta. Tällaisesta järjestelmästä voisi hyötyä laajalle hajautetut laitokset, jolla ei ole jatkuvaaikaisesti päivystävää henkilöstöä. Tällöin laitosten yhteinen SCADA voisi olla yhden henkilön operoitavissa, ja mikäli pilviympäristöstä menetetään hallinta asemille, on niillä oma paikallinen SCADA ja paikalle hälytettävä henkilöstö. Toinen mahdollinen käyttökohde voisi olla kuvan 4.3 mukainen, jossa järjestelmässä on paikallinen SCADA, jonka kautta ohjataan paikallista laitosta, ja pilvipohjainen SCADA, jossa on useiden laitosten toimintaan liittyvää korkeamman tason monitorointidataa esimerkiksi laitosten yhteistehoista ja hälytyksistä.

5.6 Edut yritykselle

Uuden teknologian omaksumista varten tarvitsee sillä olla konkreettisia hyötyjä olemassa olevaan verrattuna. Pilvipohjainen SCADA-järjestelmä tuo mukanaan pilven ominaisuudet skaalautuvuudesta etäkäyttömahdollisuuksiin. Pilven ominaisuuksien lisäksi pilvipalveluiden käyttö voi johtaa mahdollisiin kustannussäästöihin sekä nopeampaan ja tehokkaampaan järjestelmän kehitykseen. Pilvipalveluiden käyttö myös vähentää järjestelmän ylläpitoon kuluvia resursseja ja mataloittaa kynnystä datan jatkokäytölle. Seuraavissa alaluvuissa esitellään karkea kustannustehokkuuden arviointi, pilvipalvelun vaikutus järjestelmän kehitykseen, pilvipalvelun ominaisuuksien hyödyt SCADA-järjestelmälle ja ylläpidolle sekä lyhyesti pohditaan datan jatkokäyttöä.

5.6.1 Kustannussäästöt

Varsinkin järjestelmän kustantavaa tahoa motivoiva tekijä pilvipalveluiden käytössä on mahdolliset halvemmat kustannukset. Tutkimuksen [84] mukaan pilvipohjainen SCADA-ratkaisu voi vähentää loppukäyttäjälle aiheutuvia kustannuksia jopa 90 % verrattuna perinteisiin järjestelmiin. Näin suuret säästöt ovat kuitenkin ääritapauksia.

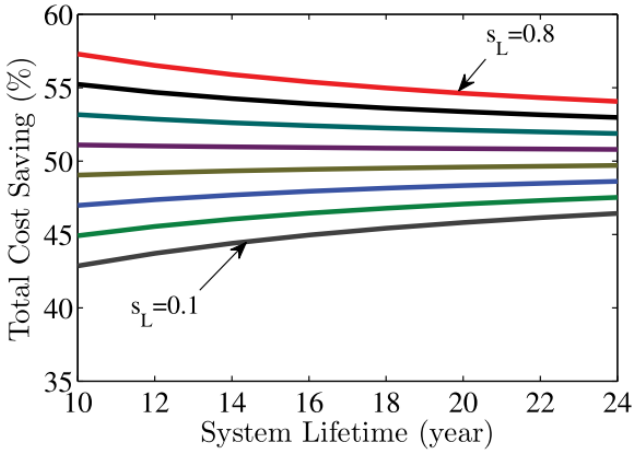
Näistä mahdollisista säästöistä iso osa muodostuu oman infrastruktuurin omistamattomuudesta. Pilvipalveluita käytettäessä ei ole tarpeen investoida omaan IT-infrastruktuurin käytännössä lainkaan. Perinteisessä järjestelmässä SCADA-järjestelmän osat sijaitsevat teollisuuden Rack IPC:llä, jotka vaativat suuria investointeja. Esimerkiksi halvimpien Siemensin teollisuustietokoneiden hinnat vaihtelevat 5000–7000 € välillä ja parempien koneiden 7000–10000 € välillä. Lisäksi Siemens suosittelee halvimmille koneille

käyttöajaksi 3 vuotta ja paremmille 5 vuotta. Tämän ajan jälkeen niihin ei Siemensin puolesta enää saa tukea tai varaosia. Mikäli halutaan käyttää redundanttisia järjestelmiä, nousevat investoinnin kustannukset lisääntyneen infrastruktuurin myötä. Lisäksi, mikäli halutaan käyttää hypervisor-ohjelmistoa ja jakaa teollisuustietokoneen resurssit virtuaalikoneiden käyttöön, on hankittavan tietokoneen hinta miltei tuplasti enemmän verrattuna edeltäviin. Käytettäessä pilvipalveluita ei siis tarvitse investoida infrastruktuuriin ennalta suuria summia vaan käytettävien palveluiden maksut voidaan hoitaa kuukausiperustaisesti. Pilvipalveluita käytettäessä on kuitenkin mahdollisuus myös useamman vuoden kertamaksuihin.

Teollisuustietokoneiden lisäksi säästöjä muodostuu myös muiden laitteistojen ja tilojen osalta. Ilman fyysisiä tietokoneita ei ole tarvetta UPS-järjestelmille, teollisuustietokoneiden kaapeloinneille tai koneiden viemälle ylimääräiselle tilalle. Lisäksi pilvipalveluita käytettäessä ei tarvitse hankkia erikseen esimerkiksi Windows-lisenssejä ja pieniä säästöjä saadaan myös energiankulutuksessa.

Säästöjä syntyy pelkän fyysisen infrastruktuurin vähenemisen myötä myös ihmisten työtuntien määrässä. Fyysisen infrastruktuurin vähentyessä on vähemmän laitetestausta, konfigurointia ja johdotuksia prosessilaitoksella [49]. Jaettaessa osa järjestelmän ylläpidosta ja päivittämisestä pilvipalveluntarjoajien kanssa vähenevät myös IT-resurssien hallintaan kuluvat työtunnit. Järjestelmän elinkaaren aikana tehtävät päivitykset sekä järjestelmän alkuperäinen käyttöönotto voidaan tehdä paikkariippumattomasti ja mahdollisesti perinteistä järjestelmää nopeammin. Paikkariippumattomuudella säästyy kuluja, kun toimistosta käsin töitä tekevä insinööri voi tehdä kolme kertaa kalliimman matkustavan insinöörin työt etänä [49].

Pilvipalveluiden kustannusten vertaaminen on hankalaa, sillä hinta on riippuvainen esimerkiksi SCADA-järjestelmän resurssitarpeista, käyttöjärjestelmästä, datakeskuksen sijainnista, verkkonopeuksista, palveluntarjoajasta, käyttöönottomallista ja palvelutasosta. Halvimmillaan Azuren julkisesta pilvipalvelusta saa virtuaalikoneen käyttöön parilla sadalla eurolla kuukaudessa ja kalleimmillaan kuukausihinnat kipuavat tuhansiin euroihin. Aberdeen yhtiön raportin mukaan yksityisiä pilvipalveluita käyttävät organisaatiot säästävät kustannuksissa n. 12 % enemmän kuin julkisia pilvipalveluita käyttävät [45]. T. Hegazy et. al. esittävät kuvassa 5.2 arvioidut säästöt järjestelmälle, jossa myös ohjauslaitteistot on viety pilvipalveluun ja yhteensä käytössä olevia virtuaalikoneita on useita satoja. Kuvassa viivat kuvaavat säästöjä eri insinöörityön säästöprosenteilla s_L, joka vaikuttaa järjestelmän kehityskustannuksiin.



Kuva 5.2. *T. Hegazy et. al tutkimuksessa arvioidut säästöt [49].*

Arvioista nähdään, että järjestelmän iän pidentyessä (*System Lifetime*) juoksevien kulujen eli pilvipalveluiden maksujen ja ylläpidon kustannusten tuomat säästöt dominoivat alkuperäiskustannuksia. Mikäli alkuperäiskustannusten säästöt ovat pienemmät kuin juoksevien kulujen säästöt, niin kokonaissäästöt (*Total Cost Savings*) nousevat järjestelmän käyttöiän mukana. Vastaavasti esimerkiksi 80 % työaikasäästöjen tapauksessa (s_L = 0,8) alkuperäiskustannusten säästöt ovat suuret ja juoksevien kulujen säästöt pienempiä, jolloin säästöt kokonaiskustannuksissa laskevat. Arvioin mukaan kokonaiskustannussäästöt ovat 10–80 % työtuntisäästöillä 45–55 %.

5.6.2 Käyttöönotto ja kehitys

Pilvipohjaisella SCADA-järjestelmällä on muutamia ominaisuuksia, joiden vuoksi se on perinteistä järjestelmää helpompi ja nopeampi kehittää sekä käyttöönottaa. Pilvipalveluiden virtuaalikoneita käytettäessä suuri etu on järjestelmään pääsy internetin ylitse paikkariippumattomasti. Paikkariippumattomudeen avulla koko kehitys- ja käyttöönottotyö voidaan tehdä etänä. Koneet voidaan asentaa sekä FAT (*Factory Acceptance Testing*) -testata etänä. Samalla jää pois asennettujen koneiden lähetykset asiakkaan tiloihin ja järjestelmän uudelleen pystytys asiakkaan tiloissa. Paikkariippumattomalla kehityksellä voidaan esitellä asiakkaalle kehitystä vaivattomasti, tai asiakas voi itse ottaa osaa kehitykseen. Päätelaiteriippumattomuuden ansiosta järjestelmää voidaan myös ajoaikana monitoroida miltä laiteelta tahansa.

Fyysisten komponenttien pois jäädessä voidaan säästää järjestelmän kehityksessä ajallisesti huomattavasti, koska tarve odottaa tilattujen komponenttien saapumista poistuu. Viime vuosien komponenttipulasta johtuvat toimitusajat ovat venyneet pitkiksi, jolloin on tarvinnut odottaa toimitusajat ennen koneiden asennuksia. Virtuaalialustoilla voidaan

mahdollisesti käyttää olemassa olevien virtuaalikoneiden kopiointia ja niiden sisällön mukauttamista uusissa projekteissa, jolloin osa asennettavista ohjelmistoista on jo valmiiksi asennettu. T. Hegazy et. al. esittävät tutkimuksessaan [49], että aikaväli suuren laitoksen järjestelmän suunnittelun aloituksesta järjestelmän ajossa olemiseen voi lyhentyä 40–70 %, eli järjestelmän suunnittelu ja pystytys voi nopeutua jopa kolminkertaisesti.

5.6.3 Skaalautuvuus ja ylläpito

Pilvipalveluiden käyttö SCADA-järjestelmissä tuo mahdollisten kustannussäästöjen ja nopeamman kehityksen lisäksi helpottavia puolia järjestelmän ylläpitoon ja tulevaisuuden muutoksiin. SCADA-järjestelmien pitkien elinkaarien aikana järjestelmät usein muuttuvat, niitä päivitetään ja mahdollisesti laitoksia laajennetaan tai uudistetaan.

Pilvipalveluiden skaalautuvuus ja uusien palveluiden käyttöönotto tekee järjestelmän laajentamisesta hyvin nopeaa ja helppoa verrattuna perinteiseen järjestelmään. Pilvipalvelua käytettäessä voidaan esimerkiksi uusia järjestelmän osia luoda uusille virtuaalikoneille, tietokantojen ja tallennustilan määrää voidaan tarpeen mukaan lisätä dynaamisesti sekä muuttaa käytössä olevien virtuaalikoneiden resursseja. Tarpeen mukaan voidaan myös skaalata järjestelmää alaspäin esimerkiksi pitkien tuotantoseisakkien ajaksi.

Ylläpidon näkökulmasta vastuun jakaminen järjestelmän ylläpidosta pilvipalveluntarjoajan kanssa on sekä etu, että mahdollinen ongelma. Etuina fyysisen infrastruktuurin ylläpidon vastuu jää kokonaan palveluntarjoajalle ja käytetyn palvelutason mukaan myös osa ohjelmistojen ylläpidosta siirtyy pois käyttäjältä. Toisaalta vastuun siirtäminen voi hankaloittaa tilannetta. Esimerkiksi jotkin SCADA-ohjelmistot ovat hyvin tarkkoja mitä käyttöjärjestelmäpäivityksen paketteja koneelle voidaan asentaa, jolloin PaaS-tason käyttöjärjestelmän ylläpito palveluntarjoajan toimesta voi muodostua ongelmaksi. Ohjelmallisen ylläpidon kuten päivitysten etuna pilvipalveluissa on nopeat virtuaalikoneiden tallennuskuvat ja niiden palautukset. Ennen järjestelmän ohjelmistojen päivityksiä virtuaalikoneesta voidaan ottaa tilannekuva ja tallentaa se pilveen, jonka jälkeen vasta päivitetään järjestelmän komponentit. Mikäli päivityksessä menee jokin pieleen, tai siitä aiheutuu toiminnallisia ongelmia, voidaan järjestelmän tilannekuvan tallenne palauttaa nopeasti takaisin, ja normalisoida toiminta. Virtuaalikoneiden automaattisten tallennekuvien otto on myös vikasietoisuuden kannalta olennaista.

Ylläpitoa helpottaa myös pilvipalvelun mahdollistama järjestelmän etäkäyttö. Etäkäytön myötä järjestelmää ei ole enää tarpeellista mennä päivittämään paikanpäälle vaan se voidaan tehdä kokonaan etänä. Lisäksi pilvipalveluiden mahdollistama käyttöoikeuksien jako järjestelmään helpottaa käyttöoikeuksien hallintaa.

Viimeinen ylläpidollinen kohta liittyy pilvipalveluilla toteutetun etävalvomon omistajan vaihtoon. Perinteisessä järjestelmässä omistajan vaihdon yhteydessä valvomon infrastruktuuri tarvitsee mahdollisesti siirtää uuden omistajan tiloihin, kun taas virtuaalisessa järjestelmässä ei ole fyysistä valvomo-omaisuutta, jolloin riittää vain järjestelmän pääsyn muutos uuden omistajan mukaiseksi.

5.6.4 Datan jatkokäyttö analytiikassa

Joissain järjestelmissä prosessidata saatetaan jo viedä pilviympäristöön ja siellä jatkokäyttöön. Tämä kuitenkin vaatii perinteisen SCADA-järjestelmän rinnalle tavan ja syyn viedä data pilveen. Pilvipalvelupohjaista SCADA-järjestelmää käytettäessä kaikki järjestelmän data kulkee automaattisesti pilveen, jolloin käyttäjän on helpompi halutessaan jatkokäyttää dataa.

Pilvipalveluiden resurssit mahdollistavat tehokkaiden koneoppimis- ja tekoälyalgoritmien käytön data-analytiikassa ja siten tarjoavat monia erilaisia työkaluja datan prosessointiin ja analysointiin. Dataa voidaan jatkojalostaa käytettäväksi päätöksenteon tukena esimerkiksi tuotannon suunnittelussa ja mallintamisessa, tai datasta voidaan esimerkiksi ennustaa järjestelmän kunnossapitotarpeita ja poikkeamia. Esimerkiksi Paula Jyrkösen [57] diplomityössä tutkittu poikkeamien tunnistaminen jätevesipumppaamojen datasta voisi hyvinkin olla jatkuvassa ajossa pilvipalvelussa.

5.7 Tulosten validiteetti

Tutkimuksen validiteetille suurin uhka on aihealueen uudehko luonne ja siitä johtuva lähdekirjallisuuden niukkuus. Tutkittujen vaatimusten ja etujen tulokset pohjautuvat osa-alueittain vain muutamiin lähdetutkimuksiin, jolloin niiden pohjalta tehdyt päätelmät eivät ole yhtä laadukkaita ja luotettavia kuin useampaan saman aihealueen tutkimukseen pohjautuvat. Lisäksi esimerkiksi osa pilvipalveluiden saatavuuskatkoksien materiaalista on peräisin kolmansien osapuolien raportoinnista, eikä se siten ole vertaisarvioitua ja täysin luotettavaa. Samoin työssä käytettyjen kaupallisten ohjelmistojen lähdemateriaalina olevien tuotteiden verkkosivujen asiasisältö ei aina pidä täysin paikkaansa ja toiminnallisuudet saattavat olla liioiteltuja.

Pääosin käytetty aineisto on kuitenkin vertaisarvioitua, luotettavaa ja paljon siteerattua. Lähdemateriaalissa ei myöskään ollut ristiriitaista tietoa, ja lähdetutkimuksissa ongelmien kuvaukset ja tulokset olivat samankaltaisia. Materiaalin niukkuudesta huolimatta työssä on onnistuneesti vastattu asetettuihin tutkimuskysymyksiin.

6. YHTEENVETO

Tämän työn tarkoituksena oli selvittää kirjallisuuden pohjalta pilvipohjaisen SCADA-järjestelmän turvallisen ja oikeellisen toiminnan kannalta tärkeimpien järjestelmä- ja luotettavuusvaatimusten toteutumista. Näitä vaatimuksia olivat järjestelmän reaaliaikaisuus, ja luotettavuuden osalta saatavuus sekä tieto- ja kyberturva. Lisäksi työssä pyrittiin arvioimaan SCADA-järjestelmän pilveen viemisestä saatavia etuja ja mahdollisuuksia. Näitä vaatimuksia ja etuja on pyritty pohtimaan julkisten pilvipalveluiden, AWS:n ja MS Azuren palveluiden kautta sekä järjestelmille, joissa ANSI/ISA-95 standardin mukaisista kolmesta alimmasta tasosta vain taso 2 on viety pilveen.

Reaaliaikaisuus on SCADA-järjestelmän oikeellisen ohjauksen ja monitoroinnin kannalta oleellista. Ilman reaaliaikaisuutta ohjaukset myöhästyvät ja monitoroitava data on vanhaa, jolloin sen pohjalta tehtävät operaattorin päätökset eivät ole oikeellisia. SCADA-järjestelmän pilveen vienti pakottaa järjestelmän käyttämään internetyhteyksiä, ja järjestelmän osien välille voi muodostua maantieteellistä matkaa tuhansia kilometrejä. Internetin käytön seurauksena järjestelmään voi aiheutua viivettä, joka vaarantaa reaaliaikaisuuden vaatimuksen. Tutkimuksessa todettiin, että SCADA-komponenttien vienti pilviympäristöön ei juurikaan lisää prosessoinnista johtuvaa viivettä eikä internetin yli kommunikointi vaikuta viiveeseen merkittävästi. Todettiin myös, ettei tietoturvan lisääminen VPN-tunnelilla ja salauksella tai maantieteellisen hajautuksen käyttö aiheuta merkittäviä ongelmia reaaliaikaisuudelle. Ehdotuksena tiedonsiirron toteutukselle ja viiveen minimoinnille suositeltiin käytettävän tapahtumapohjaisia tiedonsiirtoprotokollia ylimääräisen ja redundantin datan vähentämiseksi. Todettiin myös, että tulee ottaa huomioon järjestelmässä käytössä olevat legacy-komponentit ja -järjestelmät, jotka saattavat aiheuttaa järjestelmään lisäviiveitä. Pilvipohjaisen SCADA-järjestelmän reaaliaikaisuus on siis riittävä toteuttamaan yleisesti SCADA-järjestelmissä käytetyn 1–2 s kiertokyselyn verkkoyhteyksien normaalitilanteessa ilman merkittäviä ongelmia. Lähdetutkimuksissa ei kuitenkaan ole otettu kantaa verkkoyhteyksien ongelmatilanteisiin, jotka voivat aiheuttaa ongelmia esimerkiksi datan käsittelylle ja ylemmän tason ohjauksille.

SCADA-järjestelmän turvallisen toiminnan kannalta on ensiarvoisen tärkeää, että SCADA-järjestelmästä pystytään jatkuvasti monitoroimaan ja ohjaamaan fyysistä prosessia. Pilvipalvelut eivät vielä saatavuuden osalta saavuta kovan saatavuuden 99.999 % rajaa vaan saatavuus on lähempänä 99.9 prosenttia, joka vuositasolla tarkoittaa useiden tuntien käyttökatkoksia. Käyttökatkoset eivät kuitenkaan yleensä vaikuta kaikkiin pil-

ven tarjoamiin palveluihin, koska palveluita tarjotaan useissa eri maantieteellisesti hajautetuissa datakeskuksissa. Lisäksi pilvipalvelut tarjoavat mahdollisuuksia käyttää ohjelmallista redundanssia ja ennen kaikkea mahdollisuuden hajauttaa käytetyt redundanssikomponentit maantieteellisesti eri datakeskuksiin, jolloin voidaan varautua yhden tai useamman datakeskuksen käyttökatkoon. Käytetyllä palvelutasolla ei saatavuuden osalta ole juurikaan merkitystä, koska kaikki palveluntarjoajan palvelut pohjautuvat samaan infrastruktuuriin ja sen saatavuuteen. Saatavuuskatkoksia ei siis pilvipalveluissa ole usein, mutta lyhyetkin katkokset saattavat vaarantaa tuotantoprosessin toiminnan ja turvallisuuden, jolloin saatavuuden osalta pilvipohjaiset SCADA-järjestelmät eivät vielä sovellu kriittisille prosesseille. Pienet katkokset saatavuudessa eivät puolestaan haittaa prosesseissa, joiden monitorointi ja ohjaus ei ole jatkuva-aikaista ja hetkellisen hallinnan menetys ei tuota vaaraa.

SCADA-järjestelmät ovat alun perin suunniteltu käytettäviksi yksityisissä paikallisissa verkoissa mahdollisimman tehokkaasti, jolloin niiden tieto- ja kyberturvaan ei ole juurikaan panostettu. Samoin SCADA-tiedonsiirtoprotokollat on suunniteltu yksityisiin verkkoihin keskittyen nopeuteen, ja sen vuoksi niiden tietoturvalliset ominaisuudet ovat vähäisiä. SCADA-järjestelmän integrointi pilviympäristöön lisää uhkia entisestään altistamalla tieto- ja kyberturvallisesti heikko järjestelmä julkiselle internetille sekä lisäämällä yhteyspisteitä järjestelmään. Työssä todettiin, että riskejä pilvipalvelun käytössä aiheuttavat julkisen internetin yli kommunikointi, jaettu pilvi-infrastruktuuri, heikosti suojatut järjestelmät ja protokollat sekä pahantahtoisten käyttäjien mahdollisuudet päästä käsiksi järjestelmään. Lisäksi olemassa olevien legacy-komponenttien olematon tieto- ja kyberturvallisuus heikentää pilvi-integraation turvallisuutta entisestään. Pilvipalveluiden historiassa on myös esiintynyt käyttökatkoksia, joiden mukana osa pilvipalveluun talletetusta datasta on kadottaa peruuttamattomasti. Pilvipalveluita käytettäessä vastuu turvallisuudesta huolehtimisesta jakaantuu käyttäjän ja palveluntarjoajan välille riippuen käytetystä palvelutasosta. Osa turvallisuudesta huolehtimisesta jää siis pilvessä tieto- ja kyberturva-ammattilaisten hoidettavaksi ja käyttäjälle jää oman käyttöympäristön ja ohjelmiston suojaaminen. Turvallisuuden parantamiselle työssä ehdotettiin yksityisen pilven käyttöä, yhteyksien salausta VPN-tunneloinnilla, pilven omien turvallisuuta edistävien palveluiden käyttöä sekä datan varmuuskopiointia toisiin datakeskuksiin. Turvallisuuden puolesta pilviympäristö on vielä haavoittuvainen ja SCADA-ohjelmistojen turvaominaisuudet liian heikot. Tieto- ja kyberturvan puolesta pilvipohjainen SCADA-järjestelmä ei vielä sovellu kriittisille järjestelmille.

SCADA-järjestelmän viennin pilvipalveluun todettiin mahdollistavan yritykselle erilaisia etuja. Pilvipalveluita käytettäessä fyysisen infrastruktuurin pois jääminen pienentää alkuinvestointeja ja mahdollisesti jopa järjestelmän kokonaiskustannuksia sen elinkaaren aikana. Samalla todettiin ylläpitoon ja suunnitellun kuluvan vähemmän työtunteja, jolloin järjestelmän suunnittelu ja ylläpito elinkaaren aikana halventuu ja nopeutuu. Käyttöönoton osalta etäkäyttö mahdollistaa muutaman perinteisen järjestelmän käyttöönoton vaiheen sivuuttamisen, joka nopeuttaa järjestelmän suunnittelua, ja siten aikaa suunnittelun aloituksesta tuotantoon. Luonnollisesti SCADA-järjestelmän pilveen viennistä saadaan käyttöön pilvipalveluiden yleiset ominaisuudet. Näistä olennaisimmat ovat järjestelmän skaalautuvuus ja helpot konfiguraatiomuutokset tulevaisuudessa, ylläpidon vastuun jakaminen palveluntarjoajan kanssa sekä järjestelmän päätelaiteriippumaton etäkäyttö. Lisäksi todettiin, että pilvipohjaista järjestelmää käytettäessä data kulkeutuu automaattisesti pilveen, jolloin kynnys sen jatkokäytölle on pienempi.

Työssä todettiin, etteivät tutkitut vaatimukset täyty vielä tarpeeksi hyvin kriittisten prosessien SCADA-järjestelmien pilveen vientiin. Ehdotuksia tämänhetkisistä käyttökohteista olivat etäkäyttöiset järjestelmät, joissa hetkellinen käyttökatkos ei aiheuta vaaratilannetta, tai laitoksella on paikallinen pilven toiminnasta riippumaton SCADA-järjestelmä, jonka kautta laitosta voidaan operoida paikallisesti katkoksen sattuessa. Toinen mahdollinen käyttökohde olivat järjestelmät, joissa on prosessilaitoksilla paikalliset SCADA-järjestelmät, ja pilvessä usean laitoksen yhteinen SCADA-järjestelmä, johon on kerätty laitoksille yhteistä monitoroitavaa dataa.

Aihepiirin uudehkon luonteensa vuoksi tutkimusta on tehty vielä vähäisesti ja olemassa olevia pilvipohjaisia SCADA-järjestelmiä on vähän ja ne ovat pieniä. Jatkotutkimusta ajatellen reaaliaikaisuuden osalta voitaisiin tutkia viiveitä oikeilla kenttälaitteilla. Lähdetutkimuksissa kenttälaitteet oli toteutettu simulaattoreilla, joiden viiveet eivät ole suoraan verrattavissa fyysisiin laitteistoihin. Tietoturvan osalta voisi tehdä syvempää tutkimusta pilven turvaominaisuuksista ja niiden yhteensopivuuksista SCADA-järjestelmien kanssa. Hyvä tutkimuksen aihe olisi myös selvitys käytännön tasolla, miten ”lift and shift” -metodilla toteutetun kaupallisen SCADA-järjestelmän pilveen vienti voitaisiin toteuttaa ja vertailla keskenään eri tarjoajien ohjelmistojen soveltuvuutta pilviympäristöön.

LÄHTEET

[1] S. Alam, M. Shuaib, A. Samad, A collaborative study of intrusion detection and prevention techniques in cloud computing, International Conference on Innovative Computing and Communications, Springer, 2019, pp. 231–240.

[2] A. Almalawi, Z. Tari, A. Fahad, X. Yi, SCADA Security: Machine Learning Concepts for Intrusion Detection and Prevention, Newark, John Wiley & Sons, 2020.

[3] F. Alshehry, A. Wali, Analysis of Security Challenges in Cloud-Based SCADA Systems: A Survey, TechRxiv, 2022.

[4] B. Akyol, Cyber Security Challenges in Using Cloud Computing in the Electric Utility Industry, Washington, Pacific Northwest National Laboratory, 2012.

[5] M. Antunes, M. Maximiano, R. Gomes, D. Pinto, Information Security and Cybersecurity Management: A Case Study with SMEs in Portugal, Journal of Cybersecurity and Privacy, vol. 1, no. 2, 2021, pp. 219–238.

[6] N. Arboleda, AWS hit by DDoS attack bringing half of web down, CRN, 2019. Verkkosivu saatavissa (viitattu 22.12.2022): https://www.crn.com.au/news/aws-hit-by-ddos-attack-dragging-half-of-web-down-532842

[7] R. Asp, T. Tuominen, H. Hyppönen, Edu.fi, 2019. Verkkosivu saatavissa (viitattu 27.10.2022): http://www03.edu.fi/oppimateriaalit/kunnossapito/sahkotekniikka_a2_automaatiojarjestelma.html

[8] ABB:n TTT -käsikirja 2000-07, 2007, OAMK. Verkkosivu saatavissa (viitattu 28.10.2022): http://www.oamk.fi/~kurki/automaatiolabrat/TTT/24_Prosessiautomaatio.pdf

[9] Achieving Real-Time Performance on a Virtualized Industrial Control Platform, Intel Corporation, White paper, 2014, pp. 1-7. Saatavissa (viitattu 13.11.2022): https://www.intel.com/content/www/us/en/content-details/330740/achieving-real-time-performance-on-a-virtualized-industrial-control-platform-white-paper.html?wapkw=industrial%20solutions%20real-time

[10] Amazon CloudWatch Features, Amazon. Verkkosivu saatavissa (viitattu 17.12.2022): https://aws.amazon.com/cloudwatch/features/

[11] Amazon EC2 Features, Amazon. Verkkosivu saatavissa (viitattu 17.12.2022): https://aws.amazon.com/ec2/features/

[12] Amazon RDS Features, Amazon. Verkkosivu saatavissa (viitattu 17.12.2022): https://aws.amazon.com/rds/features/

[13] Amazon S3 Features, Amazon. Verkkosivu saatavissa (viitattu 17.12.2022): https://aws.amazon.com/s3/features

[14] AWS Application Migration Service, Amazon. Verkkosivu saatavissa (viitattu 17.12.2022): https://aws.amazon.com/application-migration-service/

[15] AWS Identity and Access Managament (IAS), Amazon. Verkkosivu saatavissa (viitattu 17.12.2022): https://aws.amazon.com/iam

[16] AWS Lamba Features, Amazon. Verkkosivu saatavissa (viitattu 17.12.2022): https://aws.amazon.com/lambda/features/

[17] AWS Post-Event Summaries, Amazon. Verkkosivu saatavissa (viitattu 22.12.2022): https://aws.amazon.com/premiumsupport/technology/pes/

[18] AWS pricing, Amazon. Verkkosivu saatavissa (viitattu 16.12.2022): https://aws.amazon.com/pricing/

[19] Azure Availability Zones, Microsoft, Verkkosivu saatavissa (viitattu 19.12.2022): https://www.azurespeed.com/Information/AzureAvailabilityZones

[20] Azure IoT, Microsoft, Verkkosivu saatavissa (viitattu 19.12.2022): https://azure.microsoft.com/en-us/solutions/iot/#overview

[21] Azure Pricing, Microsoft, Verkkosivu saatavissa (viitattu 19.12.2022): https://azure.microsoft.com/en-us/pricing/#product-pricing

[22] Azure status history, Microsoft, Verkkosivu saatavissa (22.12.2022): https://status.azure.com/en-us/status/history/

[23] T. Baker, M. Mackay, A. Shaheed, B. Aldawsari, Security-Oriented Cloud Platform for SOA-Based SCADA, 15th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, IEEE Press, 2015, pp. 961-970.

[24] S. Bansal, S. Sharma, I. Trivedi, A Detailed Review of Fault-Tolerance Techniques in Distributed System, International Journal on Internet & Distributed Computing Systems, 2011.

[25] S. Bergweiler, Smart Factory Systems - Fostering Cloud-based Manufacturing based on Self-Monitoring Cyber-Physical Systems, International Journal on Advances in Systems and Measurements, vol. 9, no. 2, 2016, pp. 91–101.

[26] E. Byres, H. MacKenzie, Next generation cyber attacks target oil and gas SCADA, Pipeline & Gas Journal, vol. 239, no. 2, 2012, pp. 32–33.

[27] K. Chandrasekaran, Essentials of cloud computing, 1st edition, Boca Raton, CRC Press, 2015.

[28] P. Chavan, P. Patil, G. Kulkarni, R. Sutar, IaaS Cloud Security, Proceedings of the 2013 International Conference on Machine Intelligence and Research Advancement, IEEE, 2013, pp. 549–553.

[29] Y. Chen, J. Chen, J. Gan, Experimental Study on Cloud Computing Based Electric Power SCADA System, ZTE Communications, vol. 13, no. 3, 2015, pp. 33–41.

[30] L. Chih-Wei, H. Chih-Ming, C. Chih-Hung, Y. Chao-Tung, An Improvement to Data Service in Cloud Computing with Content Sensitive Transaction Analysis and Adaptation, IEEE 37th Annual Computer Software and Applications Conference Workshop, IEEE, 2013, pp. 463–468.

[31] P. Church, H. Mueller, C. Ryan, S.V. Gogouvitis, A. Goscinski, Z. Tari, Migration of a SCADA system to IaaS clouds – a case study, Journal of cloud computing: advances, systems and applications, vol. 6, no. 1, 2017, pp. 1–12.

[32] M. Coughlan, P. Cronin, R. Frances, Doing a Literature Review in Nursing, Health and Social Care, Thousand Oaks, 2013.

[33] Compute, Microsoft, Verkkosivu saatavissa (viitattu 19.12.2022): https://azure.microsoft.com/en-us/products/category/compute/

[34] Cloud Based Architecture, Inductive Automation. Verkkosivu saatavissa (viitattu 8.12.2022): https://docs.inductiveautomation.com/display/DOC79/Cloud+Based+Architecture

[35] W. Dai, V. Vyatkin, J. H. Christensen, V. N. Dubinin, Bridging Service-Oriented Architecture and IEC 61499 for Flexibility and Interoperability, IEEE Transactions on Industrial Informatics, vol. 11, no. 3, 2015, pp. 771-781.

[36] E. Demin, S. Patil, V. Dubinin, V. Vyatkin, IEC 61499 distributed control enhanced with cloud-based web-services, 2015 IEEE 10th Conference on Industrial Electronics and Applications (ICIEA), IEEE, 2015, pp. 972–977.

[37] C. Dey, SK. Sen, Industrial automation technologies, 1st edition, Boca Raton, FL, CRC Press, 2020.

[38] R. Dietrich, Industrial Ethernet - from the Office to the Machine, Harting, 2005. Saatavissa (viitattu 27.10.2022): http://schusterusa.com/wp-content/uploads/2012/12/harting_industrial_ethernet_handbook.pdf

[39] T. Erl, Service-oriented architecture: concepts, technology, and design, Pearson Education, 2021.

[40] B. Galloway, G. P. Hancke, Introduction to Industrial Control Networks, IEEE Communications surveys and tutorials, vol. 15, no. 2, 2013, pp. 860–880.

[41] S. Garg, S. Yadav, Fault tolerance in distributed system, International Journal of Innovative Research in Technology, vol. 1, no. 5, 2014.

[42] M. Geiger, J. Bauer, M. Masuch, J. Franke, An Analysis of Black Energy 3, Crashoverride, and Trisis, Three Malware Approaches Targeting Operational Technology Systems, 2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), IEEE, 2020, pp. 1537–1543.

[43] A. Gligor, T. Turc, Development of a Service Oriented SCADA System, Procedia economics and finance, vol. 3, 2012, pp. 256–261.

[44] S. Goose, J. Kirsch, D. Wei, SKYDA: cloud-based, secure SCADA-as-a-service, International transactions on electrical energy systems, vol. 25, no. 11, 2015, pp. 3004–3016.

[45] S. Goyal, Public vs Private vs Hybrid vs Community - Cloud Computing: A Critical Review, International journal of computer network and information security, vol. 6, no. 3, 2014, pp. 20–29.

[46] M. Gundall, D. Reti, HD. Schotten, Application of Virtualization Technologies in Novel Industrial Automation: Catalyst or Show-Stopper?, 2020 IEEE 18th International Conference on Industrial Informatics (INDIN), IEEE, 2020, pp. 283–290.

[47] P. Gumaste, Amazon AWS Outage, Whizlabs, 2019. Verkkosivu saatavissa (viitattu 22.12.2022): https://www.whizlabs.com/blog/amazon-aws-outage/

[48] J. Goldberg, Cloud Computing Basics, CCSI. Verkkosivu saatavissa (viitattu 03.11.2022): https://www.ccsinet.com/blog/cloud-computing-basics/

[49] T. Hegazy, M. Hefeeda, Industrial Automation as a Cloud Service, IEEE Transactions on Parallel and Distributed Systems, vol. 26, no. 10, 2015, pp. 2750-2763.

[50] S. Hirsjärvi, P. Remes, P. Sajavaara, Tutki ja kirjoita, Tammi, 2009.

[51] How AWS Works, Amazon. Verkkosivu saatavissa (viitattu 16.12.2022): https://aws.amazon.com/startups/start-building/how-aws-works/

[52] Huge Cloud Market Still Growing at 34% Per Year; Amazon, Microsoft & Google Now Account for 65% of the Total, Synergy Research Group, 2022. Verkkosivu saatavissa (viitattu 16.12.2022): https://www.srgresearch.com/articles/huge-cloud-market-is-still-growing-at-34-per-year-amazon-microsoft-and-google-now-account-for-65-of-all-cloud-revenues

[53] Hype Cycle for Cloud Computing, 2018, Gartner, 2018. Verkkosivu saatavissa (viitattu 19.10.2022): https://www.gartner.com/doc/3884671/hype-cycle-cloudcomputing

[54] K. Iwanicki, A Distributed Systems Perspective on Industrial IoT, 2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS), 2018, pp. 1164–1170.

[55] ISA 95 Framework & Layers, Siemens. Verkkosivu Saatavissa (viitattu 26.10.2022): https://www.plm.automation.siemens.com/global/en/our-story/glossary/isa-95-framework-and-layers/53244

[56] B. Johansson, Dependable Distributed Control System: Redundancy and Concurrency defects, Mälardalen university, dissertation, 2022. Saatavissa (viitattu 24.11.2022): https://www.diva-portal.org/smash/record.jsf?pid=diva2%3A1700984&dswid=4142

[57] P. Jyrkönen, Poikkeamien tunnistaminen jätevesipumppaamon mittausdatasta, Tampereen yliopisto, diplomityö, 2022. Saatavissa: https://trepo.tuni.fi/handle/10024/138452

[58] M. Kavis, Architecting the cloud : design decisions for cloud computing service models (SaaS, PaaS, and IaaS), 1st edition, Hoboken, New Jersey, Wiley, 2014.

[59] R. Khan, K. McLaughlin, B. Kang, D. Laverty, S. Sezer, A Seamless Cloud Migration Approach to Secure Distributed Legacy Industrial SCADA Systems, 2020 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT), 2020, pp. 1–5.

[60] R. Khan, K. McLaughlin, J. Hastings, D. Laverty, S. Sezer, Inter-Technology bridging gateway: A low cost legacy adaptation approach to secure industrial systems, 2018 IEEE Power & Energy Society General Meeting (PESGM), IEEE, 2018.

[61] H. Kirrmann, D. Dacfey, Selecting a standard redundancy method for highly available industrial networks, 2006 IEEE International Workshop on Factory Communication Systems, IEEE, 2006, pp. 387–391.

[62] J. Kletti, Manufacturing Execution Systems — MES, 1st ed., Berlin, Springer-Verlag, 2007.

[63] B. Krishna, S. Kiran, G. Murali, R. Reddy, Security Issues in Service Model of Cloud Computing Environment, Procedia Computer Science, vol. 87, 2016, pp. 246–251.

[64] A. Kumar, R. Shankar, Y. Ranvijay, A. Jain, Fault Tolerance in Real Time Distributed System, International Journal on Computer Science and Engineering (IJCSE), vol. 3, no. 2, 2011.

[65] F. Lamb, Industrial Automation, 1st ed., New York, McGraw-Hill Publishing, 2013.

[66] T. Lennvall, M. Gidlund, J. Akerberg, Challenges when bringing IoT into industrial automation, 2017 IEEE AFRICON, IEEE, 2017, pp. 905–910.

[67] T. Lojka, M. Bundzel, I. Zolotová, Service-oriented Architecture and Cloud Manufacturing, vol. 13, no. 6, 2016, pp. 25–44.

[68] S.G. McCrady, Designing SCADA Application Software: A Practical Approach, Oxford, Elsevier, 2013, pp. 1–5.

[69] B. Middleton, Havex—2014, A History of Cyber Security Attacks, 1st ed., 2017, pp. 135–138.

[70] MoM vs MES: What's the difference? Matics Manufacturing Analytics Ltd, verkkosivu saatavissa (viitattu 28.10.2022): https://matics.live/blog/mom-vs-mes-whats-the-difference/

[71] M. Nabi, M. Toeroe, F. Khendek, Availability in the cloud: State of the art, Journal of network and computer applications, vol. 60, 2016, pp. 54–67.

[72] M. Naldi, The availability of cloud-based services: Is it living up to its promise?, 2013 9th International Conference on the Design of Reliable Communication Networks (DRCN), IEEE, 2013, pp. 282–289.

[73] V. Nguyen, Q. Tran, Y. Bésanger, SCADA as a service approach for interoperability of micro-grid platforms, Sustainable Energy, Grids and Networks, vol. 8, 2016, pp. 26–36.

[74] Opinnäytteen rakenne: kirjallisuustutkimus, 2021, Aalto yliopisto. Verrkosivu Saatavissa (viitattu 16.2.2023): https://my-courses.aalto.fi/mod/book/view.php?id=688064&chapterid=5302&lang=fi

[75] D. Rountree, I. Castrillo, H. Jiang, The Basics of Cloud Computing: Understanding the Fundamentals of Cloud Computing in Theory and Practice, 1st edition, Rockland, Elsevier Science & Technology Books, 2013.

[76] N. Ruparelia, Cloud computing, Cambridge, The MIT Press, 2008.

[77] P. S. Nicholas, I. Verhappen, Distributed Control Systems, Guide to the Automation Body of Knowledge. 3rd edition. ISA, 2018.

[78] A. Sajid, H. Abbas, K. Saleem, Cloud-Assisted IoT-Based SCADA Systems Security: A Review of the State of the Art and Future Challenges, IEEE Access, vol. 4, 2016, pp. 1375–1384.

[79] M. Sandikkaya, A. Harmanci, Security Problems of Platform-as-a-Service (PaaS) Clouds and Practical Solutions to the Problems, 2012 IEEE 31st Symposium on Reliable Distributed Systems, 2012, pp. 463–468.

[80] J. Seppälä, M. Salmenperä, Towards Dependable Automation, Cyber Security: Analytics, Technology and Automation, Springer International Publishing, 2015, pp. 229–249.

[81] M. Smurthwaite, M. Bhattacharya, Convergence of IT and SCADA: Associated Security Threats and Vulnerabilities, IOP Conference Series: Materials Science and Engineering, 2020, vol. 790, no. 1.

[82] B. Snyder, J. Ringenberg, R. Green, V. Devabhaktuni, M. Alamn, Evaluation and design of highly reliable and highly utilized cloud computing systems, Journal of Cloud Computing, vol. 4, no. 1, 2015, pp. 1–16.

[83] A. Soetedjo, Y. I. Nakhoda, A. Lomi, F. Farhan, Web-SCADA for Monitoring and Controlling Hybrid Wind-PV Power System, Telecommunication Computing Electronics and Control, vol. 12, no. 2, 2014, pp. 305–314.

[84] M. Stojanović, S. B. Rakas, J. Marković-Petrović, Scada systems in the cloud and fog environments: Migration scenarios and security issues, Facta universitatis - series: Electronics and Energetics, vol. 32, no. 2, 2019, pp. 345–358.

[85] K. Stouffer, J. Falco, K. Scarfone, Guide to Industrial Control Systems (ICS) Security, NIST Special Publication 800-82 Rev. 2, 2015.

[86] SCADA as a Service, Yokogawa. Verkkosivu saatavissa (viitattu 28.12.2022): https://www.yokogawa.com/au/solutions/products-and-services/lifecycle-services/scada-as-a-service/#Resources__White-Papers

[87] SCADA Software, Yokogawa. Verkkosivu saatavissa (viitattu 28.12.2022): https://www.yokogawa.com/eu/solutions/products-and-services/control/control-and-safety-system/supervisory-control-and-data-acquisition-scada/fast-tools/#Details

[88] System 800xA Solutions Handbook, ABB AB, 2016. Verkkosivu saatavissa (viitattu 14.11.2022): https://library.e.abb.com/public/e2c8177d884cef5bc1257b4e004c577f/3BSE069330_C_en_System_800xA_Solutions_Handbook.pdf

[89] H. Taherdoost, Cybersecurity vs. Information Security, Procedia computer science, vol. 215, 2022, pp. 483–487.

[90] MS. Thomas, JD. McDonald, D. John, Power system SCADA and smart grids, London, CRC Press, 2015.

[91] M. Toeroe, F. Tam, Service availability: principles and practice, Chichester, John Wiley & Sons, 2012.

[92] The NIST Definition of Cloud Computing, National Institute of Standards and Technology, 2011. Verkkosivu saatavissa (viitattu 04.11.2022): https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf

[93] Types of Cloud Computing, Amazon. Verkkosivu saatavissa (viitattu 03.11.2022): https://aws.amazon.com/types-of-cloud-computing/

[94] Types of Databases on Azure, Microsoft. Verkkosivu saatavissa (viitattu 19.12.2022): https://azure.microsoft.com/en-us/products/category/databases/

[95] O. Vermesan, P. Friess, Internet of Things – From Research and Innovation to Market Deployment, Denmark, River Publishers, 2014.

[96] V. Vyatkin, Software Engineering in Industrial Automation: State-of-the-Art Review, IEEE Transactions on industrial informatics, vol. 9, no. 3, 2013, pp. 1234–1249.

[97] S. Wind, K. Turowski, J. Repschlager, R. Zarnekow, Target dimensions of cloud computing, 2011 13th IEEE Conference on Commerce and Enterprise Computing, IEEE, 2011, pp. 231–235.

[98] What is a public cloud?, Microsoft. Verkkosivu saatavissa (viitattu 19.12.2022): https://azure.microsoft.com/en-us/resources/cloud-computing-dictionary/what-is-a-public-cloud/

[99] X. Xu, Q. Lu, L. Zhu, J. Li, S. Sakr, H. Wada, I. Weber, Availability analysis for deployment of in-cloud applications, ISARCS 2013 - Proceedings of the 4th ACM Sigsoft International Symposium on Architecting Critical Systems, ACM, 2013, pp. 11–16.

[100] Xen Project wiki, 2022. Verkkosivu saatavissa (viitattu 14.12.2022): http://wiki.xen.org/wiki/Main_Page

[101] G. Yadav, K. Paul, Architecture and security of SCADA systems: A review, International Journal of Critical Infrastructure Protection, vol. 34, 2021.

[102] M. Yi, H. Mueller, L. Yu, J. Chuan, Benchmarking Cloud-Based SCADA System, 2017 IEEE International Conference on Cloud Computing Technology and Science (CloudCom), IEEE, 2017, pp. 122–129.

[103] Q. Zhu, Y. Yang, M. Natale, E. Scholte, A. Sangiovanni-Vincentelli, Optimizing the software architecture for extensibility in hard real-time distributed systems, IEEE transactions on industrial informatics, vol. 6, no. 4, 2010, pp. 621– 636.

[104] Zedi SaaS SCADA, Emerson Electric. Verkkosivu saatavissa (viitattu 13.12.2022): https://www.zedisolutions.com/field-technology/field-instrumentation/remote-monitor/zedi-scada

[105] J. Åkerberg, M. Gidlund, M. Björkman, Future research challenges in wireless sensor and actuator networks targeting industrial automation, 9th IEEE International Conference on Industrial Informatics, 2011, pp. 410–415.