

Jarmo Talvivaara & Olli Hatakka

# Lisäarvoa tuottavien kvanttilaskentapalveluiden kartoitus ja esimerkkejä eri toimialueilla



Julkaisusarja Karelia-ammattikorkeakoulun julkaisu C: Raportteja, 112

Tekijät Jarmo Talvivaara, Karelia-ammattikorkeakoulu  
Olli Hatakka, Karelia-ammattikorkeakoulu

Kansikuva starline / Freepick

© Tekijät ja Karelia-ammattikorkeakoulu



Tämä julkaisu on lisensoitu Creative Commons Nimeä-  
EiMuutoksia 2.0 Kansainvälinen -lisenssillä.

ISBN 978-952-275-393-9

ISSN 2323-6914

Karelia-ammattikorkeakoulu 2023

*KvanttiKarelia -Kvanttilaskentaan perustuvat tulevaisuuden liiketoiminta- ja palvelumahdollisuudet -lump sum selvityshanke 2020-2022*

*Quantum Karelia"-Quantum Computing Services - future technology for business and service applications -lump sum 2020-2022*



**Vipuvoimaa**  
**EU:lta**  
2014-2020

# Sisällys

Esipuhe.....	4
Foreword.....	6
1 Johdanto.....	8
2 Kvanttilaskennan mahdollisuuksia, haasteita ja sovelluskohteita.....	11
2.1 Liiketalous, liiketoiminta.....	17
2.2 Materiaalitekniikka, kestävä kehitys, energiankulutus.....	22
2.2 Robotiikka, automaatio, simulaatio.....	24
2.4 Kemia, biokemia.....	26
2.5 Sairaanhoido, terveydenhuolto, lääketiede, farmakologia.....	27
2.6 Kyberturvallisuus, tietoturva, salausten menetelmät.....	31
2.7 Koneoppiminen, analytiikka.....	32
3 Quantum Computing-as-a-Service: kvanttilaskenta pilvipalveluna.....	36
3.1 Johdanto.....	36
3.2 Computing-as-a-Service: perinteiset pilvipalvelut.....	37
3.3 Quantum Computing-as-a-Service: kvanttilaskentaa pilvipalveluna?.....	39
3.4 Perinteistä ja kvanttilaskentaa pilvipalvelussa?.....	43
3.5 Kvanttilaskentapilvipalveluita ja -palveluntarjoajia (QCaaS services and providers).....	45
3.5.1 Amazon Braket.....	46
3.5.2 D-Wave Leap.....	51
3.5.3 Google Quantum AI ja Quantum Computing Service.....	56
3.5.4 IBM Quantum.....	60
3.5.5 Microsoft Azure Quantum.....	64
3.5.6 Oxford Quantum Circuits.....	67
3.5.7 QC Ware Forge.....	68
3.5.8 Xanadu Cloud.....	73
3.6 QCaaS- palvelut – kartoituksen johtopäätöksiä ja pohdintaa.....	78

4 Quantum Application Development: kvanttiohjelmointikieliä, kirjastoja, työkaluja ja ympäristöjä.....	82
4.1 Kvanttiohjelmointiympäristöjen kartoitus (2020–2022) .....	83
4.2 Qiskit.....	88
4.2.1 Qiskit-esimerkkisovellus.....	90
4.3 Microsoft Quantum Development kit ja Q# -kieli.....	93
4.3.1 Q#-esimerkkisovellus .....	95
4.4 D-Wave Ocean SDK.....	97
4.4.1 Ocean SDK-esimerkkisovellus .....	99
5 Quantum Machine Learning: kvanttikoneoppimisen perusteita ja mahdollisuuksia .....	101
5.1 Tekoäly, koneoppiminen ja syväoppiminen.....	101
5.1.1 Koneoppiminen.....	102
5.1.2 Syväoppiminen .....	105
5.1.3 Tekoäly, koneoppiminen ja klassisen laskennan haasteita .....	107
5.2 Kvanttikoneoppiminen (Quantum Machine Learning, QML).....	108
5.3 Esimerkkejä kvanttikoneoppimisessa käytetyistä algoritmeista .....	113
5.3.1 Klusterointi kvanttialgoritmeilla (Quantum Clustering).....	117
5.3.2 Regressio kvanttialgoritmeilla tai kvanttilaskentaa hyödyntäen (Quantum Regression) .....	118
5.3.3 Kvanttineuroverkot (Quantum Neural Networks).....	120
5.4 Ohjelmistoja, kirjastoja, kehitystyökaluja ja -ympäristöjä.....	122
5.5 Quantum Machine Learning as a Service - kvanttikoneoppimISRatkaisujen toteuttaminen QCaaS-pilvipalveluissa .....	124
5.5.1 Lopuksi .....	126

# Esipuhe

Karelia-ammattikorkeakoulun koordinoima KvanttiKarelia-hanke on vastannut haasteeseen, jonka ammattikorkeakoulut kohtaavat kehittäessään ICT-opetusta kvanttilaskennan alueella. Tällä hetkellä ei ole olemassa kattavaa tai soveltuvaa materiaalia, joka auttaisi alan kehityksen seuraamisessa ja sovellusalueiden ymmärtämisessä. Lisäksi tarvittavat kehittämisverkostot ovat puuttuneet.

Haasteen ratkaisemiseksi KvanttiKarelia-hanke on perehtynyt erityisesti pilvipalveluina tarjottaviin kvanttilaskentapalveluihin ja niiden mahdollistamiin sovelluksiin. Hankkeen tavoitteena oli myös verkostoitua kansallisten ja kansainvälisten alan asiantuntijoiden, tutkijoiden, alan yritystoiminnan ja muiden kvanttiteknologiaverkostojen kanssa sekä luoda kuvaa suomalaisesta kvanttilaskentatoimialasta ja ekosysteemin edellytyksistä ja mahdollisuuksista myös maakuntien näkökulmasta.

KvanttiKarelia-hanke on myös pyrkinyt lisäämään tietämystä kvanttilaskennasta ja sen sovelluksista. Hankkeen aikana on järjestetty useita webinaareja ja online-tilaisuuksia, joissa on käsitelty kvanttilaskentaa ja sen mahdollisuuksia. Tarkoituksena on ollut pohtia muun muassa ammattikorkeakoulujen mahdollisuuksia tukea koulutusalojen ja eri toimialojen yritysten perustietämyksen, mahdollisuuksien ja osaamisen hankintaa ja kehittämistä kvanttilaskennan ja -teknologian osalta.

KvanttiKarelia-hanke on luonut hyvän pohjan kvanttilaskentapalveluiden mahdollisimman hyvin tapahtuvan ennakoinnin tietämyksen hankkimiseksi ja sen pohjalta tapahtuvan koulutuksen suunnittelemiseksi ja alueen toimijoiden osaamisen vahvistamiseksi.

Kvanttilaskentaa pidetään ratkaisevan tärkeänä teknologiana kaikkialla maailmassa. Tästä osoituksena suunnaton määrä tutkimus- ja kehittämispanostuksia, joita on käynnissä niin Aasiassa, Euroopassa, Amerikassa kuin muullakin, tavoitteena tämän ilmiön mahdollisuuksien ja potentiaalın selvittäminen. Tämä paradigmaattisesti täysin erilainen, klassisesta tietotekniikasta poikkeava laskentamenetelmä, voi muuttaa radikaalisti kykyämme torjua ilmastonmuutosta köyhyyttä ja sairauksia. Samalla se voi kuitenkin tehdä tavanomaisesta salaustekniikasta vanhentunutta, mutta toisaalta mahdollistaa lähes purkamattoman kvanttilaskentaan perustuvan kryptografian. Kvanttilaskenta tarjoaa uusia taloudellisia mahdollisuuksia jo nyt, ja kiihtyvällä tahdilla maailmanlaajuiseen digitaalitalouteen, energiateknologiaan, monimutkaisten ilmiöiden ymmärtämiseen, tekoälyyn, koneoppimiseen sekä moniin

teollisiin ja muihin erityisesti tehokkaasta laskennallisesta optimoinnista ja mallintamisesta hyötyviin aloihin.

Kehittyviin teknologioihin liittyvien epävarmuustekijöiden ja nopeasti muuttuvan ja hajanaisen kokonaiskuvan hallitseminen on kuitenkin haasteellista. Erilaisten organisaatioiden, yritysten ja julkisen sektorin päättäjien on vaikea ylläpitää riittävää tietoisuutta ja ymmärrystä ja oikea-aikaista päätöksentekokykyä siitä, milloin ja miten reagoida.

Tämän toisen raportin tavoitteena on tarjota mahdollisimman kattava yleiskatsaus teemaan **“Lisäarvoa tuottavien kvanttilaskentapalveluiden kartoitus ja esimerkkejä eri toimialueilla”**. Se pyrkii tarjoamaan perustietoa lisäarvoa tuottavasta kvanttilaskennasta esimerkkien avulla kerrottuna eri toimialueilta, sekä luomaan helposti ymmärrettävän kokonaiskuvan muun muassa yritysjohtajille ja muille päättäjille, sekä kaikille muillekin asiasta kiinnostuneille. Tällä raportilla pyrimme vahvistamaan tietoon ja tosiasioihin perustuvaa ymmärrystä tästä ilmiöstä ja siten antamaan eväitä kvanttilaskennan merkityksen ja mahdollisuuksien ymmärtämiseen.

Kvanttilaskenta saattaa olla avain parempaan huomiseen, mutta tässä raportissa pyritään välttämään sekä siihen liittyvää liiallista innostusta että toisaalta myös mahdollisten vaikutusten vähättelyä. KvanttiKarelia hankkeen tavoitteena on ollut myös vähentää kvanttilaskentaan liittyvää yleistä tietämättömyyttä sekä lisätä kvanttilukutaitoa kaikkien aihealueesta kiinnostuneiden henkilöiden, yritysten, päättäjien ja vaikuttajien keskuudessa.

Toivomme että tämä lyhyt raportti antaa perustietoa kvanttilaskennasta ja luo ilmiöstä helpommin lähestyttävämmän ja mahdollisuuksia avaavan kuvan sekä rohkaisee tutustumaan tähän ilmiömaailman avoimin mielin ja uteliaisuudella. Emme pysty ennustamaan milloin kvanttilaskenta muuttuu valtavirtojen teknologiaksi, mutta siihen varautuminen kannattaa ja antaa eväitä uusille ideoille ja ajatuksille siitä, miten *“minä liityn kvanttilaskentaan ja kvanttilaskenta liittyy meihin”*.

Joensuussa 23.4.2023

Olli Hatakka ja Jarmo Talvivaara

KvanttiKarelia -Kvanttilaskentaan perustuvat tulevaisuuden liiketoiminta- ja palvelumahdollisuudet -hanke, Karelia-ammattikorkeakoulu

# Foreword

The QuantumKarelia project, coordinated by Karelia UAS, has responded to a challenge where universities of applied sciences face a massive challenge to develop ICT education in the field of quantum computing. Currently, there is no comprehensive or suitable material to help follow developments in the field and understand the application areas. Moreover, the necessary development networks are lacking.

In order to address this challenge, the QuantumKarelia project has focused in particular on cloud computing services and the applications they enable. The project also aimed at networking with national and international experts, researchers, industry and other quantum technology networks, and at creating a picture of the Finnish quantum computing industry and the ecosystem conditions and opportunities, also from a regional perspective.

The QuantumKarelia project has also sought to increase knowledge about quantum computing and its applications. During the project, several webinars and online events have been organized on quantum computing and its potential. The aim is to explore, among other things, how universities of applied sciences can support the acquisition and development of basic knowledge, capabilities and competences in quantum computing and technology in educational sectors and enterprises in different industries.

The QuantumKarelia project has provided a good basis for acquiring the best possible knowledge of the anticipation of quantum computing services and for designing training based on this knowledge and strengthening the skills of actors in the region. The aim is to meet the challenges of the applied skills needs of higher education in future technologies, in particular quantum computing.

Quantum computing is considered a crucial technology worldwide. This is evidenced by the huge amount of research and development efforts underway in Asia, Europe, America and beyond to explore the possibilities and potential of this phenomenon. This paradigmatically completely different method of computing, which is distinct from classical computing, could radically change our ability to combat climate change, poverty and disease. At the same time, however, it can render conventional cryptography obsolete, but on the other hand, it can enable near-uncompressible quantum computing-based cryptography, and offer new economic opportunities, already, and accelerating, for the global digital economy, energy technology,

understanding complex phenomena, artificial intelligence, machine learning, and many industrial and other sectors that benefit in particular from efficient computational optimization and modelling.

However, managing the uncertainties associated with emerging technologies and the rapidly changing and fragmented landscape is a challenge. It is difficult for different organizations, companies and public sector decision-makers to maintain sufficient awareness and understanding and timely decision-making ability on when and how to react.

The aim of this second report is to provide a concise mapping of value-added quantum computing services and examples across different domains

It aims to provide basic information on value-added quantum computing, illustrated with examples from different domains, and to provide an easy-to-understand overview for business leaders and other decision-makers, as well as anyone else interested. With this report, we aim to strengthen the knowledge and fact-based understanding of this phenomenon, and thus to provide a basis for understanding the importance and potential of quantum computing.

Quantum computing may hold the key to a better tomorrow, but this report seeks to avoid both over-enthusiasm and underplaying its potential impact. The QuantumKarelia project has also aimed to reduce the general lack of knowledge about quantum computing and to increase quantum literacy among all interested individuals, businesses, policy makers and influencers.

We hope that this short second report will provide basic information about quantum computing and will create an open-minded and curious approach to the phenomenon, its implications, with examples, the possibilities for different domains, now and in the future, and encourage people to explore this phenomenal world. We cannot predict when quantum computing will become a mainstream technology, but being prepared for it is worthwhile and will stimulate new ideas and thoughts on how "I relate to quantum computing and quantum computing relates to us".

Joensuu, 23th of April 2023

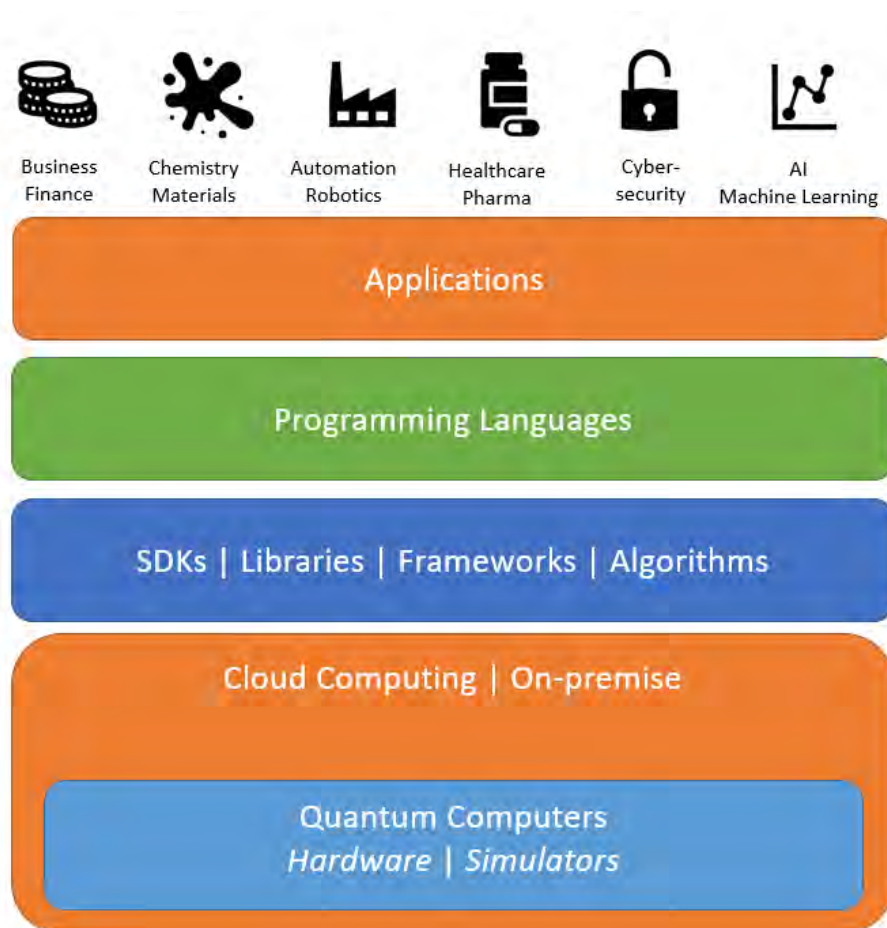
Olli Hatakka and Jarmo Talvivaara

Quantum Karelia - Quantum Computing Services – future technology for business and service applications project, Karelia UAS



# 1 Johdanto

Tässä raportissa esitellään kvanttilaskennan soveltamista erilaisilla sovellusalueilla, erilaisia soveltamisessa hyödynnettäviä pilvipalveluita (*quantum computing as a service*), kvanttietokoneiden ohjelmointiin (*quantum programming*) soveltuvia ohjelmointityökaluja; ohjelmointikieliä, niihin liittyviä SDK- sovelluskehitysympäristöjä (*quantum software development kit*), hyödynnettäviä kirjastoja ja sovelluskehyskiä ja niiden hyödyntämisen peruseriaatteita (*quantum application development*). Lopuksi tarkastellaan tarkemmin todennäköisesti yhtä monikäyttöisintä ja potentiaalisinta kvanttilaskennan sovelluskohdetta; kvanttikoneoppimista (*quantum machine learning*).



Luvussa 2 tarkastellaan kvanttilaskennan hyödyntämistä erilaisilla sovellusalueilla; mitkä sovelluskohteet ja ratkaisut voisivat olla potentiaalisia ja mahdollisia kohteita kvanttilaskennan hyödyntämiseksi.

Luvussa 3 luodaan katsaus hankkeen aikana saatavilla oleviin, julkaistuihin ja vahvimmin kehityksen kohteena olleisiin kvanttilaskentapalveluihin; erityisesti sellaisiin pilvipalveluihin, jotka tarjoavat kvanttietokoneiden käyttömahdollisuuksia tai muuten kvanttilaskentakapasiteettia internetin välityksellä.

Luvussa 4 esitellään erilaisia kvanttiohjelmointiin tarkoitettuja ohjelmointikieliä ja -ympäristöjä sekä niiden soveltamista.

Luvussa 5 tarkastellaan yhtä erityistä kvanttilaskentaa hyödyntävää osa-aluetta; kvanttikoneoppimista. Luvussa käsitellään myös klassisella toteutettavan tekoälyn, koneoppimisen ja syväoppimisen peruskäsitteitä ja -konsepteja, kuten esimerkiksi datan luokittelua (clustering) tai luonnollisen kielen käsittely (NLP), jossa yksi sovel-lusalue on ns. geneeriset kielimallit.

Jokaiseen lukuun ja aihealueisiin on liitetty linkkejä lähteisiin, joiden on tarkoitus auttaa saamaan kyseisestä aiheesta lisätietoa. Liitettyjen lähteiden on tarkoitus tarjota lisätietoja useammalla eri tasolla ja eri käyttötarkoituksia palvellen: lähteissä on mukana niin yleisen tason kuvauksia, teknisiä dokumentaatioita soveltamisesta kuin aihepiiriä tarkemmin tarkastelevia tutkimuksia.

Raportin on tarkoitus luoda yleiskatsaus edellä mainittuihin teemoihin. Tämän vuoksi, esimerkiksi kvanttilaskennan matemaattiseen tai fysikaaliseen taustaan, kvanttilaskennan ja algoritmien kompleksisuusteoreettisiin detaljeihin, kvanttipiiritason toimintaperiaatteisiin tai notaatioihin raportissa ei syvennytä.

Suomeksi näihin em. aiheisiin löytyy kattavasti tietoa etenkin Julian Brownin vuoden 2001 kirjasta "Kvanttietokone". Hankkeen aikana myös Jyväskylän yliopisto (Heinonen 2021) julkaisi monipuolisen raportin otsikolla "*Katsaus kvanttilaskentateknologiaan ja sen sovelluksiin.*".

Kvanttilaskennan matemaattista taustaa käsittelevät esimerkiksi Chen & Brylinskiin (2002) kirjassaan "Mathematics of Quantum Computation" tai Wolfgang Scherer (2019) kirjassaan "Mathematics of Quantum Computing". Laskennan kompleksisuusteoreettisia näkökulmia tutkivat esimerkiksi Bernstein ja Vazirani (1997) sekä Watrous (2008).

Kvanttipiirien ja niihin liittyvien operaatioiden tarkemman tason ohjelmointiin tietoa on löydettävissä esimerkiksi Johnstonin, Harriganin ja Gimeno-Segovian vuoden 2019

kirjasta "Programming Quantum Computers: Essential Algorithms and Code Samples".

Raportissa viitatussa kvanttietokoneiden ns. NISQ- aikakaudesta sekä niiden kehitysnäkymistä tarkempaa tietoa löytyy etenkin John Preskillin vuoden 2018 artikkelista "Quantum Computing in the NISQ era and beyond".

Arviota, ennusteita sekä tietoa panostuksista kvanttilaskennan eri sovelluskohteisiin löytyy hyvin mm. Bob Sorensenin tekemistä markkinakatsauksista.

Kvanttikoneoppimisen näkökulmiin syvällisempää tietoa löytyy mm. Qiskit:in "Quantum Machine Learning"- materiaaleista sekä varsinkin sen teoreettiseen, matemaattiseen ja tietojenkäsittelytieteen näkökulmiin mm. Maria Schuld et al. ja Seth Lloyd et al. artikkeleista.

Eri työkalujen, ohjelmointikielten, kehitysympäristöjen, pilvipalveluiden ja muiden teknisten ympäristöjen dokumentit ohjeistavat tarkemmin niiden käyttöönottoa ja käytön periaatteita.

Erilaiset tekoälypalvelut voivat tukea myös kvanttilaskennan tutkimista ja soveltamista; raporttia kirjoittaessa eri lähteistä tunnistettuja tuloksia, sovelluskohteita ja -esimerkkejä verrattiin esim. Chat GPT:n (v. 3.5) tuottamiin tuloksiin. Monin paikoin tulokset olivatkin varsin yhteneviä ja paikkaansa pitävältä vaikuttavia. Tulokset toki vaikuttivat rajautuvan vuoteen 2021 mennessä tuotettuihin tuloksiin, tunnistettuihin sovelluskohteisiin sekä kehitettyihin työkaluihin. Oletettavaa on, että edelleen kehittyvät - ja mahdollisesti jollakin aikavälillä kvanttilaskentaa hyödyntävät - kielimallit ja muut tekoälypalvelut voivat toimia tehokkaana työkaluna kvanttilaskennan mahdollisuuksien tutkimisessa ja etenkin soveltamisessa.

Linkkejä edellä mainittuihin sekä muihinkin tietolähteisiin löydät raportin "Lisätietoja ja lähteitä" -osioista.

## 2 Kvanttilaskennan mahdollisuuksia, haasteita ja sovelluskohteita

*Jarmo Talvivaara*

Tässä luvussa tarkastellaan kvanttilaskennan soveltamista ja mahdollisuuksia eri sovellusaloilla. Hankkeen alussa valittiin muutamia sovellusalueita, joissa kvanttilaskennalla oletetaan olevan käyttöä ja hyötyä lyhyellä tai pitkällä aikavälillä.



Tarkasteltavat sovellusalueet ovat

1. Liiketalous, liiketoiminta
2. Materiaalitekniikka
3. Robottiikka, automaatio, simulaatio
4. Kemia, biokemia
5. Sairaanhoido, terveydenhuolto, farmakologia
6. Kyberturvallisuus, tietoturva, salaustenmenetelmät.
7. Koneoppiminen, analytiikka

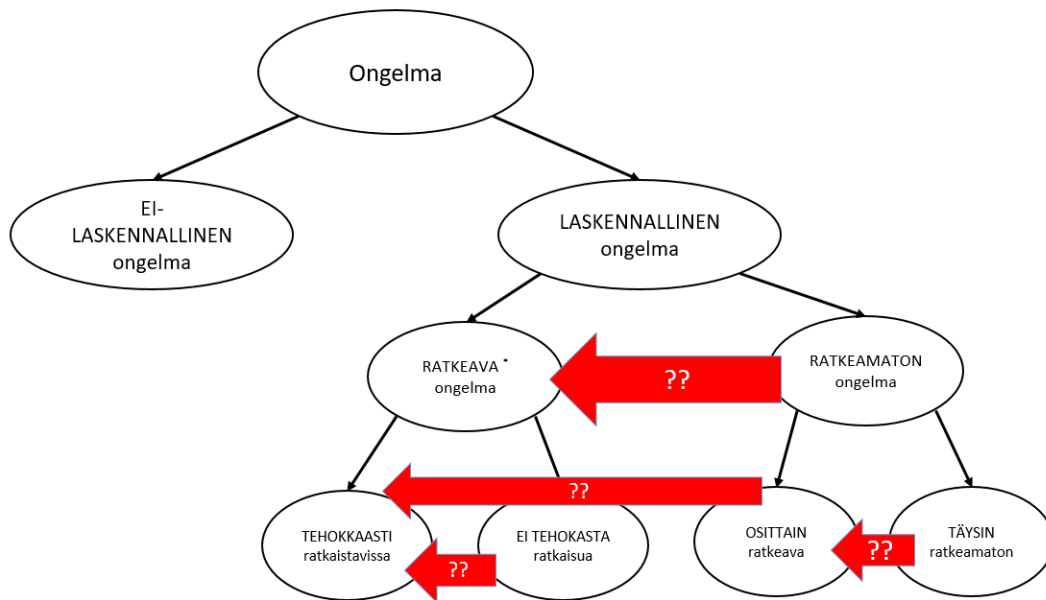
Ennen sovelluskohteiden tarkastelua, voi olla hyvä esittää yleisiä perusteita siitä, millä tavoin kvanttilaskenta voi tuoda etuja sen käyttämisessä erilaisiin sovelluskohteisiin – etenkin perinteisen tietotekniikan hyödyntämiseen verrattuna tai sen tehostamisessa.

### **Quantum speedup – kvanttinopeutus**

Kvanttitietokoneilla voi olla (nyt tai tulevaisuudessa) kyky ratkaista joitakin ongelmia nopeammin, kuin mikä tahansa tähän mennessä kehitetty tehokkain perinteinen tietokone. Tämä nopeutus johtuu kvanttitietokoneen kyvystä käsitellä useita laskentaprosesseja samanaikaisesti, joka on taas mahdollista kvanttien lomittumisen

(*entanglement*) ja superpositioiden avulla. Kun perinteiset tietokoneet suorittavat tehtäviä yksitellen peräkkäisinä sekvensseinä, voivat kvanttietokoneet suorittaa laskentaa kvanttien superpositiota hyödyntäen löytäen useita eri ratkaisuja samanaikaisesti. Tämä mahdollistaa kvanttietokoneiden ratkaistavaksi joitakin ongelmia paljon nopeammin kuin perinteiset tietokoneet.

Tietojenkäsittelytieteen laskennan kompleksisuusteoreettisesti tämä voi vaikuttaa esim. niihin ongelmiin, joihin ei ole aiemmin löytynyt tehokasta ratkaisua – mahdollistaen kvanttilaskennan avulla tehokkaiden ratkaisuiden tuottamisen. Mielenkiintoa kannattaa myös tietenkäynnä suunnata siihen, voiko kvanttilaskenta tarjota ratkaisuja tilanteisiin, joissa a) ongelmaan ei ole aiemmin löytynyt kuin osittainen ratkaisu, b) joissa ongelma on ollut aiemmin täysin ratkeamaton sekä c) onko kvanttilaskennan keinoin aiemmin ratkeamattomiin ongelmiin löytää jollakin aikavälillä tai menetelmällä ratkaisu.



Kuva 1. Ongelmien ratkeavuus ja kvanttilaskennan vaikutuksien tarkastelun kohteita. (Soveltaen Bridge, D.; Harel, D. 2004; Stanford Encyclopedia of Philosophy 2021; Fuller, B., Mandelbaum, R. 2022)

Tietojenkäsittelytieteen laskennan teorian kannalta tämä voi olla sekä merkittävä, paljon mahdollisuuksia tarjoava, mutta myös haasteita aiheuttava tulokulma – jota ei kuitenkaan kannattane jättää tarkastelematta.

Jos kvanttilaskennan etuja voidaan hyödyntää niissä sovelluskohteissa, missä perinteistä tietotekniikkaa käytetään, on oletettavissa, että kvanttilaskenta voi

toimiessaan tarjota kyseisiin käyttökohteisiin merkittäviä etuja. Etuja voidaan oletettavasti saavuttaa sekä jo olemassa oleviin ratkaisuihin, mutta myös luoda jopa uusia, aiemmin tuntemattomia sovellutuksia. Nämä edut voivat olla saavutettavissa esim. korvaamalla tai täydentämällä klassisia algoritmeja tai muita ohjelman osia ja niitä suorittavia perinteisiä tietokoneita kvanttilaskennasta etuja saavilla versioilla (kvanttialgoritmit) ja niitä mahdollistavilla kvanttietokoneilla. Kvanttilaskennan tehostama koneoppiminen on todennäköisesti yksi, ja vieläpä hyvin monipuolisesti eri käyttökohteisiin sovellettava kehittämisen kohde, jota tässä on hyvä seurata.

### **Quantum supremacy - kvanttiherruus**

Quantum supremacy, "kvanttiherruus" eli kvanttietokoneen kyky suorittaa laskentaa, joka on (teoriassa ja jopa käytännössä) mahdotonta suorittaa edes tehokkaimmilla perinteisillä tietokoneilla. Tarkemmin sanottuna "kvanttiherruus" viittaa siihen pisteeseen, jossa kvanttietokone pystyy ratkaisemaan ongelman, joka vaatisi perinteiseltä tietokoneelta epäkäytännöllisen pitkän ajan.

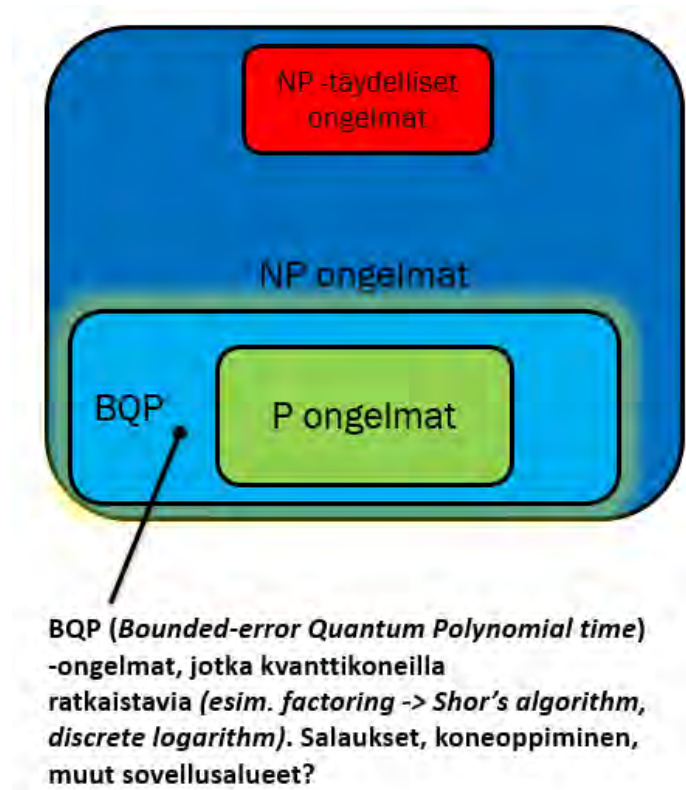
Tämä todistetaan yleensä suorittamalla tietty laskenta kvanttietokoneella ja osoittamalla sen suorituskykyä verrattuna perinteisiin tietokoneisiin. Tämä osoitus tapahtuu usein ajamalla satunnaisia piirejä kvanttietokoneella, jossa on suuri määrä kubitteja ja mitataan tulos. Tulos verrataan siihen, mitä perinteinen tietokone ennustaisi samasta piiristä. Jos kvanttietokone tuottaa tuloksen, jota ei voida käytännöllisesti saavuttaa perinteisillä menetelmillä, silloin on saavutettu kvanttietokoneiden ylivoimaisuus.

Yksi tunnetumpia esimerkkejä tästä on Googlen vuonna 2019 tekemä koe, jossa suoritettavan laskentaoperaation arvioitiin vievän perinteisellä supertietokoneella n. 10 000 vuotta ja kokeessa 53- kubittinen kvanttietokone (Sycamore) suoritti laskennan n. 200 sekunnissa.

Kvanttietokoneiden ylivoimaisuuden saavuttaminen ei kuitenkaan automaattisesti tarkoita sitä, että kvanttietokoneet pystyvät ratkaisemaan kaikki laskennalliset ongelmat tehokkaammin kuin perinteiset tietokoneet. Se kuitenkin osoittaa kvanttietokoneiden potentiaalın ratkaista ongelmia, jotka ovat tällä hetkellä perinteisille tietokoneille käytännössä mahdottomia, kuten monimutkaisten kemiallisten reaktioiden simuloiminen tai tiettyntyyppisten salausjärjestelmien murtaminen.

Yksi siirtymä, minkä kvanttilaskennan on ennustettu aiheuttavan, on ns. ongelmien ratkaisemisen tai ratkaisun oikeellisuuden "vaikeuteen": klassisten P-, NP- ja NP- täydellisten ongelmien laajenuksena esitetään kvanttietokoneille luokkaa BQP, joka

sisältäisi aiemmin vaikeista tai mahdottomista ongelmista ne, jotka olisivat kvanttietokoneilla ratkaistavissa. Näiden ongelmien ratkaisumahdollisuuksista voi syntyä myös aiemmin tuntemattomia ja hyödyntämättömiä ratkaisuja - jopa uusia sovelluksia eri aloilla hyödynnettäväksi.



Kuva 2. Laskennallisten ongelmien luokkia. (Soveltaen Fuller, B., Mandelbaum, R. 2022.)

Tässä raportissa ei keskitytä tarkemmin näihin kompleksisuusteoreettisiin näkökulmiin, vaan kvanttilaskennan tehokkuusetuja tarkastellaan yleisemmällä tasolla.

### Quantum subroutines - kvanttialirutiinit

Kvantti-alirutiinit (*quantum subroutines*) ovat kvanttilaskennan periaattein toteutettuja ohjelman osia, jotka ratkaisevat tietyn ongelman tai suorittavat tietyn tehtävän osana suurempaa kvanttialgoritmia. Näitä alirutiineja voidaan käyttää kokonaisvaltaisen kvanttialgoritmin suorituskyvyn parantamiseen. Sovelluskohteita löytyy esim. kvanttikoneoppimisen alueella.

## “Hitting the moving target?”

Perinteisiin tietokoneisiin pohjautuvat laskentaratkaisut, etenkin erilaiset HPC- eli High-Performance Computing- suurteholaskentaa tarjoavat ympäristöt, ovat jo nykyisin erittäin suorituskykyisiä laskemaan raskaita ja monimutkaisia ongelmia (esim. erilaiset tekoälyn sovellusalueet, mm. syväoppiminen). Nämä teknologiat ja saatavilla oleva, perinteisiin tietokoneisiin pohjautuva laskentakapasiteetti kehittyy jatkuvasti ja tulee yhä laajemmin tarjolle erilaisiin käyttökohteisiin.

Kvanttitietokoneiden osalta eletään ns. NISQ- aikakautta (*Noisy Intermediate-Scale Quantum- era*), jolloin a) ei ole saatavilla yleispätevää (universaalial) kvanttitietokoneita ja b) saatavilla olevat kvanttitietokoneet toistaiseksi koostuvat vielä vähäisistä määristä kubitteja ja/tai kubitit ovat “kohinaisia” (*noisy*). Vähäisellä määrällä, *intermediate scale*, tarkoitetaan yleensä kubitien määrää alkaen muutamasta kymmenistä enimmillään muutamiin satoihin kubitteihin. “Kohinalla” viitataan usein kubitien laadun haasteisiin, eli esim. kubitien matala fideliteetti tai liian lyhyet porttiooperaatioajat eivät mahdollista vielä kovin pitkien kvanttiporttien suorittamista, etenkin ilman toimivaa ja usein hankalasti toteutettavaa virheenkorjausta.

Vielä toistaiseksi, verrattaessa perinteisten supertietokoneiden ja kvanttitietokoneiden aitoa kapasiteettia ja tehokkuusetuja toisiinsa, voidaan kvanttilaskennan nähdä yrittää osuvan “liikkuvaan maaliin”; perinteisten supertietokoneiden kapasiteetti kasvaa nopeaa vauhtia, kun taas kvanttitietokoneiden kehityksessä on vielä omat pulonkaulansa – ja täten kvanttitietokoneita ei voida vielä ainakaan yleispätevästi hyödyntää korvaamaan perinteisiä supertietokoneita.

On kuitenkin hyvä huomioda, että mitä todennäköisimmin a) perinteisiä ja kvanttitietokoneita tullaan hyödyntämään yhteistyössä (quantum-classical hybridit) ja b) kvanttitietokoneiden kehitykseen, nimenomaan kvanttien määrän sekä laadun parantamiseen, kohdennetaan myös jatkossa merkittäviä tutkimus- ja kehityspanoksia. Lisäksi c) NISQ- aikakauden kvanttitietokoneita on mahdollista hyödyntää tiettyjen erityisongelmien (mm. optimointi) ratkaisemiseen – ja tämänkin vuoksi on tärkeää, että teknologioita tutkitaan, kehitetään ja etenkin kokeillaan aktiivisesti eri sovellusalueissa. On todennäköisesti hyödyllisempää tutkia ja tuntea eri sovelluskohteita, tutustua soveltamisen eri mahdollisuuksiin ja soveltamisessa tarvittavien teknologioiden etuihin ja rajoituksiin sekä testata mahdollisia ratkaisuja simulaattoreilla ja rajoittuneemmilla kvanttitietokoneilla – kun pelkästään jäädä odottamaan tilannetta, jossa olisi saatavilla yleiskäyttöinen kvanttitietokone.



## Lähteitä ja lisätietoja:

Aaronson, S., Chen, L. 2016. Complexity-Theoretic Foundations of Quantum Supremacy Experiments. <https://arxiv.org/pdf/1612.05903.pdf>

Arute et al. 2019. Nature. Quantum supremacy using a programmable superconducting processor. <https://www.nature.com/articles/s41586-019-1666-5>

Bernstein, E. ja Vazirani, U. 1997. Quantum Complexity Theory. <https://epubs.siam.org/doi/10.1137/S0097539796300921>

Brown, J. (suom. Pietiläinen, K.). 2001. Kvanttitietokone. Terra Cognita.  
Heinonen, H. 2021. Katsaus kvanttilaskentateknologiaan ja sen sovelluksiin. Jyväskylän yliopisto. Informaatioteknologian tiedekunnan julkaisuja.  
[https://jyx.jyu.fi/bitstream/handle/123456789/74322/88\\_2021\\_Katsaus\\_kvanttilaskentateknologiaan\\_ja\\_sen\\_sovelluksiin\\_\\_VERKKOJ.pdf](https://jyx.jyu.fi/bitstream/handle/123456789/74322/88_2021_Katsaus_kvanttilaskentateknologiaan_ja_sen_sovelluksiin__VERKKOJ.pdf)

Fuller, B, Mandelbaum, R. 2022. What Can a Quantum Computer Actually Do? <https://medium.com/qiskit/what-can-a-quantum-computer-actually-do-4daed0691f6b>

Hodges, A. 2005. Can quantum computing solve classically unsolvable problems? <https://arxiv.org/abs/quant-ph/0512248>

Preskill, J. 2018. Quantum Computing in the NISQ era and beyond. <https://quantum-journal.org/papers/q-2018-08-06-79/pdf/>

Sorensen, B. 2021. The State of Quantum Computing Today: Market Sizing, CAGR Projections, Buyer Expectations. <https://insidehpc.com/2021/09/the-state-of-quantum-computing-today-market-sizing-cagr-projections-buyer-expectations/>

Sorensen, B. 2022. Hyperion: Classical HPC Trajectory Concerns Pushing Quantum R&D. <https://insidehpc.com/2022/12/hyperion-concerns-about-trajectory-of-classical-hpc-helping-to-drive-quantum-rd>

Watrous, J. 2008. Quantum Computational Complexity. <https://arxiv.org/pdf/0804.3401.pdf>

## 2.1 Liiketalous, liiketoiminta



*Olli Hatakka*

Kvanttilaskennan avulla voidaan suorittaa monimutkaisia liiketaloudellisia laskenta-tehtäviä, jotka ovat perinteisille tietokoneille erittäin vaikeita, liian hitaita tai jopa mahdottomia toteuttaa. Tämän vuoksi kvanttilaskenta tarjoaa jo nyt, ja tulevaisuudessa vielä parempia, mahdollisuuksia liiketaloudellisten simulointien ja muiden laskentatehtävien toteuttamiseen. kvanttilaskentaa voidaan hyödyntää liiketaloudellisissa simuloinneissa ja laskentatehtävissä, esimerkiksi seuraavissa tehtävissä:

- **Portfoliosimuloinnit:** Sijoitussalkkujen optimointi on tärkeä tehtävä monille liiketoimintayrityksille, ja kvanttilaskennan avulla voidaan suorittaa nopeita ja tarkkoja portfoliosimulointeja. Kvanttilaskenta voi auttaa löytämään optimaalisen salkun, joka minimoi riskin ja maksimoi tuoton.
- **Optimaalisen reitityksen suunnittelu:** Kvanttilaskennan avulla voidaan ratkaista monimutkaisia reititysongelmia, joita voi esiintyä esimerkiksi logistiikka- tai kuljetusyrityksissä. Kvanttilaskenta auttaa löytämään optimaalisen reitin, joka minimoi matka-ajan ja polttoainekustannukset.
- **Markkinoiden analysointi:** Kvanttilaskennan avulla voidaan analysoida markkinadataa ja ennustaa markkinoiden kehitystä. Tämä auttaa yrityksiä tekemään parempia päätöksiä, kun ne suunnittelevat uusia tuotteita tai palveluita markkinoille.
- **Tietoturvan parantaminen:** Kvanttilaskennan avulla voidaan parantaa tietoturvaa, kun käsitellään suuria tietomääriä. Kvanttilaskennan avulla voidaan kehittää uusia salausten menetelmiä ja tunnistusjärjestelmiä, jotka ovat huomattavasti turvallisempia kuin perinteiset menetelmät.

Nämä siis ovat esimerkkejä siitä, miten kvanttilaskentaa voidaan hyödyntää liiketaloudellisten simulointien ja muiden laskentatehtävien toteuttamisessa. Kvanttilaskenta tarjoaa monia mahdollisuuksia liiketoimintayrityksille, ja sen käyttö tulee todennäköisesti kasvamaan tulevaisuudessa, teknologian ja kvanttietokoneiden kehittyessä. On kuitenkin huomattava, että nykyisissä laitteistoissa ja jopa kvanttisimulaattoreissa käytettävissä olevien kubittien määrä on vielä aivan liian pieni, jotta todellinen liiketoimintasovellus olisi elinkelpoinen. Tämä rajoitus saattaa kuitenkin lieventyä lähitulevaisuudessa, kun kvanttietokoneiden "kohinaa" onnistutaan

vähentämään ja virhetoleranttien kubittien määrää voidaan kasvattaa (Wilkens, S., et al. 2023).

Viimeaikaisten havaintojen perusteella näyttäisi voitavan päätellä, mitkä ovat parhaat tavat ratkaista todellisia portfolio-optimointiongelmia VQE:n avulla kvanttilaitteissa, ja vahvistaa, että niitä voidaan ratkaista nykyisiä menetelmiä tehokkaammin heti, kun kvanttilaitteiston koko on riittävän suuri (Buonaiuto, G. 2023).

## **Los Angelesin satamalogistiikan tehostaminen**

Los Angelesin satama on yksi maailman vilkkaimmista satamista, ja sen toimintojen tehostaminen on erittäin tärkeää sen kilpailukyvyn ja tehokkuuden kannalta. Vuonna 2018 Los Angelesin satama aloitti yhteistyön Kanadan kvanttilaskentayrityksen D-Wave-Systemsin kanssa tutkiakseen kvanttietokoneiden mahdollisuuksia satamatoimintojen optimoinnissa.

Tutkimuksessa käytettiin D-Wave 2000Q -kvanttietokonetta, joka on yksi maailman tehokkaimmista kaupallisista kvanttietokoneista. Tavoitteena oli löytää parempia tapoja optimoida satamatoimintoja, kuten lastausta ja purkamista, kuljetusreittien suunnittelua ja konttien varastointia.

Tutkimuksessa käytettiin kvanttilaskennan menetelmiä, kuten kvanttikilpailutusta (quantum annealing), joka on yksi D-Wave Systemsin erikoisosaamisalueista. Kvanttikilpailutus on erityinen algoritmi, joka on suunniteltu löytämään optimaalisia ratkaisuja monimutkaisiin optimointiongelmiin.

Tutkimuksen menetelmänä käytetty Hyper-Optimized Nodal Efficiency Engine (HONE) virtaviivaisti satamatoimintoja merkittävästi useiden keskeisten suorituskykyindikaattoreiden osalta. Esimerkiksi HONE:n jälkeen terminaali käytti lähes 40 prosenttia vähemmän RTG-nosturiresurssejaan purkamiseen, ja kukin näistä nostureista kulki huomattavasti pienemmän keskimääräisen matkan päivässä, 6 200 metriä 8 900 metrin sijasta. Nosturit lisäsivät myös toimitusten määrää yli 60 prosenttia, ja terminaaliin saapuvat kuorma-autot käyttivät kukin lähes kymmenen minuuttia vähemmän aikaa hyötykuormansa vastaanottamiseen.

METHOD	RESULTS	
Per Day	Before HONE	After HONE
Deliveries Per Crane (Per Day)	60	97
Truck Turn Time (Minutes)	66	58
Crane Utilization	45%	72%
Average Crane Distance	8900 M	6200 M

Kuva 3. Sataman logistiikka numeroina optimoinnin jälkeen. (Kuva: D-Wave/SavantX)

Kun optimoidaan konttien lajittelua, järjestystä ja toimitusta satamassa vain muutamalla prosenttiyksiköllä, voidaan saavuttaa kymmeniä miljoonia dollareita vuodessa. Kun laiturit 300 tuli myyntiin vuonna 2021, se ostettiin 2,3 miljardilla dollarilla - kolminkertaisella hinnalla vuoden 2018 hintaan verrattuna.

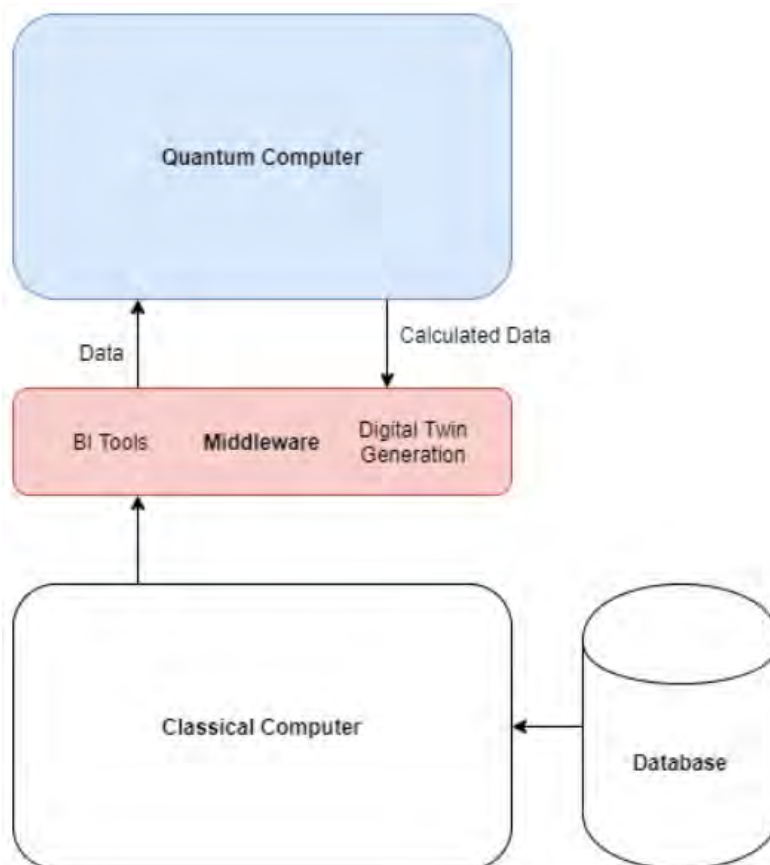


Kuva 4. Los Angelesin satama. (Kuva: SavantX)

## Kvanttilaskenta-avusteinen BI ja reaaliaikainen digitaalinen kaksosen

Kvanttitietokoneiden käyttäminen yhdessä klassisten tietokoneiden kanssa edellyttää liiketoimintatietojen keruun uudelleenmäärittelyä, joka on vasta ensimmäinen vaihe kvanttilaskenta-avusteisessa liiketoimintatietojen hyödyntämisessä ja analysoinnissa. Sen sijaan, että käytettäisiin vain historiatietoja ja regressioalgoritmeja yhden heuristisen ratkaisun määrittelemiseksi yhteen ongelmaan, kvanttilaskenta voi luoda yrityksestä ajantasaisen digitaalisen kaksosen. Tämä taas mahdollistaa, että yritysjohtajat voivat nähdä päätöksen tai olosuhteen vaikutukset koko organisaatiossa käytännössä reaaliaikaisesti ja välittömästi.

Mallilla voidaan simuloida nopeaa kasvua tai katastrofaalisia tapahtumia, mikä mahdollistaa perustellumman päätöksenteon suunnitelmia laadittaessa. Usein päätökset on lisäksi tehtävä nopeasti, ja johto voi analysoida live-dataa ja saada tarkan tilannekuvan liiketoiminnasta välittömästi, sen sijaan, että he analysoisivat hitaiden SQL-kyselyjen pohjalta laadittuja raportteja. Kvanttilaskennan käyttöönoton myötä, BI:n kehitysharppaus tulee olemaan suurin sitten sen tietokoneistamisen.



Kuva 5. Kvanttilaskentaa hyödyntävän Digital Twin- palvelun rakenne (Kuva: Michael Heiner)

Kvanttilaskennasta tulee vuosikymmenen kuluessa välttämätön tehokkaan tietojenkäsittelyn ja analytiikan kannalta. Vuoteen 2025 mennessä datan tuotannon ja kuluksen ennustetaan yli kaksinkertaistuvan nykyiseen verrattuna. Jotta BI-työkalujen avulla voidaan edelleen tuottaa ohjaustietoa tästä alati lisääntyvästä datan määrästä, niihin on integroitava kvanttilaskentaa ja kvanttioptimointia.

Kun kvanttietokoneet ja -laskenta kypsyvät, tämä ongelma saattaa väistyä. Jotkin kvanttietokoneita valmistavat yritykset, kuten IonQ, lupaavat tulevaisuudessa palvelin"räkkeihin" asennettavia kvanttijärjestelmiä, joita organisaatiot voisivat ylläpitää itse. Tämä voi olla tulevaisuudessa mahdollista, koska sekä BI-järjestelmät, että kvanttilaskenta- ja -tietokoneteknologiat jatkavat kehittymistään edelleen. (Tämä lyhyt kuvaus perustuu Michael Hernerin lokakuussa 2022 kirjoittamaan LinkedIn-artikkeliin "Quantum Business Intelligence and Quantum Augmented Digital Twins".

#### **Lähteitä ja linkkejä aiheeseen liittyen:**

Taipale, P. 2023. Käytännön kvanttilaskentaa: Luottoriskimallien optimointia tulevaisuuden menetelmillä. [https://www.op-media.fi/puheenvuorot/luottoriskimallien-optimointia-tulevaisuuden-menetelmilla/?fbclid=IwAR0W0QGvc3EeVzCyYrcObImZ-bo7lppXlpPf5\\_elvSwNQsfoBc6HUyTsHU](https://www.op-media.fi/puheenvuorot/luottoriskimallien-optimointia-tulevaisuuden-menetelmilla/?fbclid=IwAR0W0QGvc3EeVzCyYrcObImZ-bo7lppXlpPf5_elvSwNQsfoBc6HUyTsHU)

D-Wave. 2023. The Only Real-Time Quantum Cloud Service Built for Business. <https://www.dwavesys.com/quantum-cloud-services-built-for-business/>

Wilkens, S., Moorhouse, J. 2023. Quantum computing for financial risk measurement. *Quantum Inf Process* 22, 51 (2023). <https://doi.org/10.1007/s11128-022-03777-2>

Buonaiuto, G., Gargiulo, F., Pietro, G., et al. 2023. Best practices for portfolio optimization by quantum computing, experimented on real quantum devices, 24 February 2023, PREPRINT (Version 1) available at Research Square <https://doi.org/10.21203/rs.3.rs-2601946/v1>

Heinbockel, E., SavantX and D-Wave. 2021. SavantX: Logistics Optimization at the Port of Los Angeles. [https://www.dwavesys.com/media/y3hl22va/dwave\\_port\\_of\\_la\\_case\\_story\\_v7.pdf](https://www.dwavesys.com/media/y3hl22va/dwave_port_of_la_case_story_v7.pdf)

Heiner, M., 2022. Quantum Business Intelligence and Quantum Augmented Digital Twins. <https://www.linkedin.com/pulse/quantum-business-intelligence-augmented-digital-twins-michael-heiner/>

SavantX. 2021. HONE (Hyper Optimization Nodal Efficiency) AI Engine. Youtube. [https://youtu.be/mRoitwgl\\_cg](https://youtu.be/mRoitwgl_cg)

## 2.2 Materiaalitekniikka, kestävä kehitys, energiankulutus



*Olli Hatakka*

Laskennallisia pullonkauloja **materiaalitekniikassa** riittää ja näihin voisi tulevaisuudessa löytyä tehokkaita ratkaisuja kvanttilaskennan avulla. Karelia-ammattikorkeakoulun tarkkuustekniikan laboratorion asiantuntijat ja opetushenkilöstö esittivät useita erilaisia mahdollisia sovelluskohteita tarkkuusteknologioiden alueelle, joista tässä muutama. Asiantuntija ja lehtori Juha Väyrynen nosti esille mahdollisia **tarkkuustekniikkaan liittyviä laskentamenetelmiä**, jotka voisivat hyötyä kvanttilaskennasta. Ensimmäisenä hän toi esille FEM-laskennan, jossa dynaamisia, paljon vapausasteita sisältäviä, paljon laskenta-aikaa ja -kapasiteettia vaativia lujuuslaskentatehtäviä, voitaisiin ehkä suorittaa kvanttilaskennan avulla.

**FEM- tai Femi-laskenta** (finite element method, FEM) on numeerinen menetelmä, jota käytetään monimutkaisten teknisten ongelmien ratkaisemiseen erityisesti rakennepuolijäntä ja mekaniikan alalla. Siinä ongelma jaetaan pienempiin, yksinkertaisempiin elementteihin ja ratkaistaan sitten kukin elementti erikseen matemaattisten yhtälöiden ja tietokoneohjelmoinnin avulla. Nämä elementtien ratkaisut yhdistetään sitten ongelman kokonaisratkaisuksi. FEM-menetelmä on hyödyllinen analysoitaessa rakenteita, jotka ovat liian monimutkaisia tai vaikeita analyttisesti ratkaistavaksi. Sen avulla insinöörit voivat simuloida todellisia tilanteita, ennustaa materiaalien ja rakenteiden käyttäytymistä ja optimoida suunnitteluparametreja. Femi-laskentaa käytetään laajalti esimerkiksi **ilmailu- ja avaruus-, auto-, rakennus-, kone- ja sähkötekniikan** aloilla. Sen sovelluksia ovat muun muassa rakennesuunnittelu, nestedynamiikka, lämmönsiirto ja sähkömagnetismi.

Väyrynen yhdessä laboratorion muiden asiantuntijoiden kanssa, toi esille myös toisen laskentapullonkauloja sisältävän laskentaparadigman, joka liittyy optisten **nanorakenteiden** suunnitteluun ja toteutukseen. Perinteisestä optiikasta poiketen, tämä ns. **Freeform-optiikka** ja siihen liittyvä suunnittelu, mallinnus ja tuotanto ovat erittäin laskentaintensiivisiä ja tehokkaillakin klassisilla tietokoneilla toteutettuna laskenta on hidasta, yhden tehtävän laskeminen saattaa kestää useita päiviä. Tämäkin menetelmä voisi hyötyä hybriditoteutuksessa, jossa osa matriisilaskennasta toteutettaisiin kvanttilaskenta-aliohjelmalla.

Freeform-optiikka on optisen suunnittelun ja valmistuksen ala, joka käyttää vapaita muotoja, eli ei-perinteisiä, vapaita muotoja, kuten vapaamuotoisia pintoja ja vapaita käyriä, perinteisten optisten elementtien, kuten linssien ja peilien sijaan. Freeform-optiikan tavoitteena on suunnitella ja valmistaa optisia elementtejä, jotka ovat muotoilultaan monimutkaisempia ja suorituskykyisempiä kuin perinteiset optiset elementit.

Perinteinen optiikka käyttää yleensä yksinkertaisia geometrisia muotoja, kuten pallo- ja silindrisiä linsskejä, jotka on helppo valmistaa ja joita on käytetty vuosikymmenten ajan. Näiden optisten elementtien suorituskyky ja toimivuus ovat kuitenkin rajallisia, ja ne eivät aina sovi monimutkaisiin sovelluksiin.

Freeform-optiikka tarjoaa uusia mahdollisuuksia optisten elementtien muotoiluun ja suorituskyvyn optimointiin, koska vapaamuotoiset pinnat ja käyrät voivat tuottaa ainutlaatuisia optisia ominaisuuksia, kuten monimutkaisia polttovälejä, laajakulmaisia näköaloja ja vähemmän kromaattista poikkeamaa.

Freeform-optiikan sovelluksia ovat muun muassa laser käsittely, tarkkuusmittaukset, laskennallinen kuvantaminen ja lääketieteellinen diagnostiikka. Freeform-optiikka on myös tärkeä tekijä kehitettäessä uusia optisia järjestelmiä, kuten kamerajärjestelmiä, jotka tarjoavat parempaa kuvanlaatua ja suurempaa tarkkuutta.

Tarkkuustekniikan laboratorion asiantuntijat nostivat tärkeiksi myös optimaalisten työstöratojen, erityisesti tarkkuustyöstettävien kappaleiden **jäännösmateriaalin poistamiseen** liittyvien työstöratojen laskemisen, jota voitaisiin mahdollisesti myös tehostaa, jakamalla osa laskennasta kvanttilaskennan aliohjelmalle.



## Lähteet:

Finite element method. [https://en.wikipedia.org/wiki/Finite\\_element\\_method](https://en.wikipedia.org/wiki/Finite_element_method)

Photonics-journal, special issue "Freeform Optics". <https://www.photonics.fi/event/metrology-for-xr-freeform-optics/>

Metrology for XR & Freeform Optics. <https://www.photonics.fi/event/metrology-for-xr-freeform-optics/>

Quantum Zeitgeist. 2020. Microsoft launches new Quantum Education series on YouTube. <https://quantumzeitgeist.com/microsoft-launches-new-quantum-education-series-on-youtube/>

Quartz. 2019. Quantum computing could change the way the world uses energy: <https://qz.com/1566061/quantum-computing-will-change-the-way-the-world-uses-energy>

## 2.2 Robottiikka, automaatio, simulaatio



*Jarmo Talvivaara*

Kvanttilaskentaa voidaan oletettavasti hyödyntää tehostamaan sekä samoissa että mahdollisesti myös uusissa ja erilaissa robotiikan, automaation sekä simuloinnin sovelluskohteissa, missä perinteisen laskennan ratkaisuja hyödynnetään Kvanttilaskennan käyttö **robotiikassa**: Kvanttilaskenta voi mahdollistaa robotiikan kehittämisen uusille tasoille. Esimerkiksi kvanttialgoritmien avulla voidaan suunnitella robottien liikkeitä ja tehtäviä entistä tehokkaammin ja tarkemmin, jolloin robottien suorituskyky paranee.

**Automaatio**: Kvanttilaskenta voi mahdollistaa entistä paremman automaation. Esimerkiksi kvanttialgoritmien avulla voidaan kehittää uusia menetelmiä, joilla voidaan automatisoida monimutkaisia prosesseja ja optimoida tehtävien suorittamista. Kvanttilaskennan avulla voidaan kehittää uusia menetelmiä robotiikan ja automaation ohjaamiseen. Esimerkiksi kvanttialgoritmien avulla voidaan kehittää uusia ohjausmenetelmiä, jotka mahdollistavat monimutkaisten tehtävien suorittamisen entistä tarkemmin ja tehokkaammin. Näitä etuja voidaan oletettavasti hyödyntää niin perinteisessä fyysisessä automaatiossa (esim. teollisuusautomaatio),

ohjelmistopohjaisessa automaatiassa (esim. ohjelmistorobotiikka) kuin kyberfyysisissä järjestelmissä (esim. älykkäät robotit, itseajavat ajoneuvot).

Älykkään automaation (ts. koneoppimista hyödyntävän automaation) ratkaisuihin kvanttikoneoppiminen tai muu, kvanttilaskennan etuja hyödyntävä koneoppiminen, voi tuoda merkittäviä etuja, jos järjestelmien "älykkyyden" taustalla olevia (esim. neuroverkkopohjaisia) malleja voidaan sekä syventää että monipuolistaa.

**Simulointi:** Kvanttilaskenta voi mahdollistaa entistä tarkemman simulaation eri prosesseista. Esimerkiksi kvanttialgoritmien avulla voidaan simuloida monimutkaisia kemiallisia reaktioita ja kvanttijärjestelmiä, jotka ovat perinteisten tietokoneiden avulla erittäin vaikeita simuloida.

**Kvanttimittaus:** Kvanttilaskennan avulla voidaan kehittää uusia mittausmenetelmiä, jotka mahdollistavat tarkemman tiedon keräämisen. Esimerkiksi kvanttialgoritmien avulla voidaan kehittää uusia mittausmenetelmiä, joilla voidaan mitata esimerkiksi robottien tai automaattisten järjestelmien suorituskykyä ja optimoida niitä entistä paremmin.

#### **Lähteitä ja lisätietoja:**

Bourne, J. Engineering.com. 2022. Quantum Computing Poised to Boost Industrial Automation. <https://www.engineering.com/story/quantum-computing-poised-to-boost-industrial-automation>

Cirac, J. 2020. Quantum computing and simulation - Where we stand and what awaits us. <https://www.degruyter.com/document/doi/10.1515/nanoph-2020-0351/html>

Srivastava, S. 2020. Quantum Robotics: Applications of Quantum Computing in Robotic Science <https://www.analyticsinsight.net/quantum-robotics-applications-quantum-computing-robotic-science/>

## 2.4 Kemia, biokemia



Olli Hatakka

Tietokoneavusteinen kemiallinen suunnittelu on tärkeä ajuri kehitykselle ja välttämätön edellytys ihmiskunnan selviämiseksi. Uusien yhdisteiden ja kemiallisten prosessien suunnittelu laboratoriossa on hidasta ja vaikeaa, joten niitä pyritään simuloimaan ja mallintamaan tietokoneilla. Tämä on erittäin haastavaa ja klassiselle tietokoneelle jopa mahdotonta, sillä kemiallinen sidos on itsessään kvantti-ilmiö, jonka simulointi vaatii kvanttitalan tallentamista tietokoneen muistiin. Tämä johtaa usein (klassisen tietotekniikan myötä) muistiongelmiiin, mutta kvanttietokoneiden avulla voidaan tallentaa kvanttikemiallisten yhdisteiden tilat ilman muistiongelmia. Tämä mahdollistaa uusien yhdisteiden tutkimisen ja suunnittelun.

Kvanttietokoneiden avulla voidaan oppia monia asioita kemiallisista yhdisteistä. Esimerkiksi niiden rakenteesta, kemiallisista reaktioista ja miten ne käyttäytyvät, kun niitä käsitellään tai kemiallisia reaktioita käynnistetään. Tämä on erityisen tärkeää uusien materiaalien katalyyttien käytössä kemiallisissa prosesseissa.

Pitkän aikavälin tavoitteena on käyttää kvanttietokoneita Ferredoxinin kemiallisten reaktioiden täydelliseen ymmärtämiseen. Kyseessä on pieni atomiryhmä (kaksi rautaa ja kaksi rikkiä), jonka mallintaminen klassisilla tietokoneilla ei onnistu. Tämä pieni ryhmä on kuitenkin "iso juttu", nimittäin odotetaan, että sitä voidaan käyttää **typen fiksaatioon pienellä energialla**. Tämä tarkoittaa, että sen avulla voidaan ottaa typpeä ilmasta ja liittää se uuteen molekyyliin. Tämä tulee olemaan avainaskel lannoitteiden tuotannossa, joka tällä hetkellä tapahtuu **Haber-Bosch-prosessilla**, mikä kuluttaa paljon energiaa ja aiheuttaa 2-3% maailmanlaajuisista hiilidioksidipäästöistä. Mikäli tässä onnistutaan, kvanttilaskentaa ja -tietokoneita hyödyntämällä, tällaiset esimerkit tekevät kvanttietokoneiden ja -laskennan kehittämisestä motivoivan tavoitteen ihmiskunnalle.

Kvanttietokone voi siis olla **tutkimustyökalu**, jonka tulokset hyödyttävät ihmiskuntaa, vaikka emme näkisi itse tietokonetta lainkaan. (Tämä lyhyt kuvaus perustuu Quantum Flagship -sivuston artikkeliin "Fertilizer and other Quantum Computer Chemistry".

## Lähteet:

Anderson, M. 2018. VW Solves Quantum Chemistry Problems on a D-Wave Machine. IEEE Spectrum. <https://spectrum.ieee.org/vw-tackles-chemistry-problems-with-a-dwave-quantum-computer>

Quantum Flagship. Fertilizer and other Quantum Computer Chemistry. <https://qt.eu/discover-quantum/applications-of-qt/fertilizer-and-other-quantum-computer-chemistry/>

## 2.5 Sairaanhoito, terveydenhuolto, lääketiede, farmakologia



Jarmo Talvivaara

Kvanttilaskennalla on oletettavissa isoja etuja ja paljon sovelluskohteita sairaanhoitossa ja terveydenhuollossa, lääketieteellisessä diagnostiikassa, lääkkeiden kehityksessä, epidemioiden ja muiden lääketieteellisten ilmiöiden ennustamisessa, simuloinnissa ja muussa tutkimuksessa. Kvanttilaskennan tuottamat tehokkuusedut voivat näyttäytyä esim. personoitujen lääkkeiden kehityksen tai lääketieteellisten diagnoosien merkittävässä nopeuttamisessa. Liu et al. (2022, 17) esittävät algoritmitason ehdotuksia kvanttilaskennan soveltamiseen molekyyli-tieteissä sekä arvioita niiden tuottamia etuja laskennan tehostumisessa eri sovelluskohteissa.

**Lääkekehitys:** Kvanttilaskenta voi auttaa löytämään uusia lääkkeitä ja lääkemolekyylejä nopeammin ja tehokkaammin kuin perinteiset tietokoneet. Kvanttilaskennan avulla voidaan esimerkiksi simuloida molekyylien vuorovaikutuksia ja arvioida eri yhdisteiden tehoa ennen kuin niitä testataan käytännössä.

Yksi esimerkki tällaisesta voisi olla kvanttietokoneille kehitetty VQE- eli Variational Quantum Eigensolver -algoritmi. VQE:llä on potentiaalia olla hyödyllinen lääkkeiden kehityksessä, sillä se voi mahdollistaa monimutkaisten molekyylien rakenteen laskennallisen simuloinnin kvanttietokoneilla. Tämä taas voi auttaa löytämään uusia potentiaalisia lääkeaineita ja katalyyttejä, sekä ennustamaan niiden vaikutuksia ennen kuin niitä testataan laboratoriossa. VQE-algoritmi voi auttaa optimoimaan lääkemolekyylien rakenteita ja ennustamaan niiden vuorovaikutuksia esimerkiksi ihmiskehossa, mikä voi nopeuttaa lääkkeiden kehitysprosessia ja vähentää

kustannuksia. Lisäksi VQE voi auttaa löytämään uusia lääkekohteita ja ymmärtämään paremmin sairauksien biokemiallisia mekanismeja, mikä taas voi johtaa uusien lääkkeiden löytämiseen ja sairauksien parempaan hoitoon. Nämä tavoitteet korostuvat esimerkiksi syöpäsairauksien hoidon ja lääkkeiden kehittämisessä. Esimerkiksi kasvaimien proteiinit ovat usein tärkeitä lääkekohteita syövän hoidossa, ja VQE-algoritmi voi auttaa ennustamaan näiden proteiinien rakenteita ja vuorovaikutuksia eri yhdisteiden kanssa.

Kvanttilaskennassa yhden tunnetuimman algoritmin, Groverin algoritmin, voi olla mahdollista auttaa löytämään nopeasti tietyn määrän tietoja valtavista tietokannoista. Tämä voi olla hyödyllistä löytäessä uusia lääkekohteita ja analysoitaessa lääkkeitä. Toinen tunnettu algoritmi, Shorin algoritmi, joka usein yhdistetään erilaisen salausmenetelmien purkamiseen, voi myös nopeuttaa lääkekehitystä, koska sillä voidaan nopeuttaa perinteisille tietokoneille haasteellista tekijöihin jakamista (factorization). Tämä voi olla hyödyllistä lääketieteellisessä tutkimuksessa, kun tarvitaan nopeaa tekijöihin jakamista esimerkiksi proteiinien rakenteen ratkaisemisessa. Kvanttikoneoppimisen osalta hyödyllinen voisi olla QBM -algoritmi (Quantum Boltzmann Machine), jota voidaan hyödyntää oppimaan todennäköisyysjakaumia datasta. Tästä voisi olla mahdollista hyötyä esimerkiksi lääkkeiden löytämisessä uusille kohdemolekyyleille.

Perinteisillä neuroverkoilla on paljon sovelluskohteita lääkekehityksessä. Tämän vuoksi myös kvanttineuroverkoilla (quantum neural networks) nähdään olevan potentiaalia lääkeaineiden kehittämisessä, esim. molekyylien ja kemiallisten ominaisuuksien ennustamiseen, uusien molekyylien generointiin, lääkeaineiden optimointiin sekä lääkeaineseulontaan liittyen.

**Diagnostiikka:** Kvanttilaskennan avulla voidaan analysoida ja tulkita kuvantamisdataa nopeammin ja tarkemmin kuin perinteisillä menetelmillä. Kvanttilaskentaa voidaan soveltaa tunnistamaan tarkemmin esimerkiksi aivojen toimintahäiriöitä ja diagnosoida sairauksia kuten Alzheimerin tauti ja syöpä.

Seulontatutkimuksissa tai muussa diagnostiikassa esimerkiksi klusterointialgoritmit ja niiden tehostaminen kvanttilaskennalla voivat olla merkittävä mahdollisuus. Kvanttiklusterointi (quantum clustering) voi auttaa tunnistamaan tiettyjä sairauksia tai häiriöitä varhaisessa vaiheessa ja erottamaan ne muista samankaltaisista sairauksista tai häiriöistä, samalla periaatteella mitä perinteiset klusterointialgoritmit, erottamaan potilaan ominaisuuksia, kuten biologisia merkkiaineita, jotka ovat yhteisiä tietyn sairauden kanssa, mutta seuraavin eduin:

- Suurempi tarkkuus: kvanttiklusterointi voi käsitellä monimutkaisempia tietoja ja tunnistaa piileviä rakenteita datasta, mikä voi johtaa parempaan tarkkuuteen diagnooseissa.
- Nopeus: kvanttikoneiden suuremman laskentatehon ansiosta kvanttiklusterointi voi käsitellä suurempaa määrää dataa nopeammin kuin perinteinen klusterointi.
- Monimutkaisten ongelmien käsittely: kvanttiklusterointi voi käsitellä monimutkaisia ongelmia, kuten suuren määrän muuttujia ja erityyppisiä muuttujia, joita perinteiset klusterointimenetelmät eivät kykene käsittelemään tehokkaasti.
- Parempi visualisointi: kvanttiklusterointi voi tarjota paremman visualisoinnin tiedoista ja niiden ryhmittelystä, mikä voi auttaa lääkäreitä ymmärtämään paremmin taudin tai häiriön ominaisuuksia.

Tämä voi auttaa lääkäreitä diagnosoimaan taudin nopeammin ja tarkemmin, mikä voisi johtaa parempaan hoitoon ja ennusteeseen.

Kvanttikoneoppimiseen liittyvistä algoritmeista lääketieteellisessä diagnostiikassa on esitetty esim. seuraavia arvioita:

- QPCA (Quantum Principal Component Analysis), jota voidaan käyttää tiedon esikäsittelyssä ennen kuin data syötetään klassisiin koneoppimisalgoritmeihin. QPCA:lla voidaan poistaa kohinaa datasta ja parantaa sen laatua; tämä oletettavasti voisi sekä parantaa että tehostaa lääketieteellisessä tutkimuksessa käytetyn datan käsittelyä.
- VQC (Variational Quantum Classifier) tai QSVM (Quantum Support Vector Machine), jotka kykenevät erottelmaan ja luokitteluun tietoja tehokkaasti. Näitä mahdollisesti voitaisiin hyödyntää esimerkiksi erottamaan syöpäsoluja normaaleista soluista ja tunnistamaan erilaisia syöpäsoluja ja muita sairauksia.

**Genomitutkimus:** Kvanttilaskennan avulla voidaan tutkia ja analysoida suuria määriä geneettistä dataa, mikä auttaa ymmärtämään paremmin geneettisiä sairauksia ja kehittämään uusia hoitomuotoja. Kvanttilaskennan avulla voidaan esimerkiksi simuloida DNA:n rakennetta ja vuorovaikutuksia.

**Epidemian hallinta:** Kvanttilaskennan avulla voidaan ennustaa epidemian leviämistä ja kehittää parempia strategioita sen torjumiseksi. Esimerkiksi kvanttilaskennan avulla voidaan mallintaa viruspartikkelien liikkeitä ja vuorovaikutuksia, mikä auttaa

ymmärtämään paremmin virusten leviämistä ja kehittämään tehokkaampia rokotteita.

### **Lähteitä ja lisätietoja:**

Castellanos, S. 2017. Quantum Computing May Speed Drug Discovery, Biogen Test Suggests. Wall Street Journal. <https://www.wsj.com/articles/BL-CIOB-12236>

Flöther, F., Murphy, J., Murtha, J. 2020. Exploring quantum computing use cases for healthcare: Accelerate diagnoses, personalize medicine, and optimize pricing. IBM Expert Insights. <https://www.ibm.com/downloads/cas/8QDGKDZJ>

Evers, M., Heid, A., Ostojic, I. 2021. Pharma's digital Rx: Quantum computing in drug research and development. McKinsey & Company. <https://www.mckinsey.com/industries/life-sciences/our-insights/pharmas-digital-rx-quantum-computing-in-drug-research-and-development>

Liu et al. 2022. Prospects of Quantum Computing for Molecular Sciences. Materials Theory. <https://arxiv.org/abs/2102.10081>

Lorenz, J. 2021. How quantum computing could be helpful for medical diagnostics. MedTech. <https://safe-intelligence.fraunhofer.de/en/articles/quantum-computing-in-medical-diagnostics>

Medicalfuturist. 2022. What Can Quantum Computing Do To Healthcare? <https://medicalfuturist.com/quantum-computing-in-healthcare/>

Pharma's Almanac. Quantum Computing will Transform Drug Discovery, Development, Manufacturing, and Supply Chain Management. <https://www.pharmasalmanac.com/articles/quantum-computing-will-transform-drug-discovery-development-manufacturing-and-supply-chain-management>

## 2.6 Kyberturvallisuus, tietoturva, salausten menetelmät



*Jarmo Talvivaara*

Kvanttilaskennalle on olettavasti runsaasti sovelluskohteita kyberturvallisuudessa, tietoturvassa ja salaustekniikoissa. Esimerkkeinä voidaan olettaa olevan seuraavat: Tietoturva: Kvanttilaskennan avulla voidaan kehittää uusia menetelmiä tietoturvan varmistamiseksi. Esimerkiksi kvanttialgoritmien avulla voidaan selvittää, onko järjestelmässä haavoittuvuuksia, jotka voivat mahdollistaa tietojen varastamisen.

Kvanttilaskennan käyttö salauksessa: Kvanttilaskenta voi tarjota mahdollisuuden kehittää entistä turvallisempia salaustekniikoita. Esimerkiksi kvanttialgoritmien avulla voidaan kehittää uusia menetelmiä, joissa tietoa voidaan salata kvanttibiteillä (qubit), joita ei voida kopioida tai varastaa.

Kvanttikryptografia: Kvanttilaskennan avulla voidaan kehittää kvanttikryptografiaa, joka perustuu kvanttitietoihin, jotka voivat lähettää salattua tietoa pitkiäkin matkoja. Tämä tekniikka mahdollistaa tietojen turvallisen siirron, koska salaus avautuu vain vastaanottajalle.

Verkkojen turvallisuus: Kvanttilaskennan avulla voidaan kehittää uusia menetelmiä verkkojen turvallisuuden varmistamiseksi. Esimerkiksi kvanttialgoritmien avulla voidaan kehittää uusia menetelmiä havaita ja torjua tietomurtoja, estää haittaohjelmien leviämistä ja selvittää verkkojen toiminnan häiriöiden syytä.

Kvanttitunnistus: Kvanttilaskennan avulla voidaan kehittää uusia menetelmiä tunnistaa henkilöitä ja varmistaa heidän henkilöllisyytensä. Esimerkiksi kvanttialgoritmien avulla voidaan kehittää uusia menetelmiä, joissa tunnistus perustuu kvanttitietoihin lähettämään tietoon, joka on mahdoton väärentää.

Osa yllä mainittuihin sovelluskohteisiin sovellettavista teknologioista on jo kehityksen alla, osa vaatii tarkempaa suunnittelua. Etenkin kyberturvallisuuden osalta on kuitenkin nähtävissä konsensus, että kvanttikryptografisten menetelmien kehitys sekä luotamuksen tiedon suojaaminen niillä olisi tärkeää aloittaa mahdollisimman aikaisessa vaiheessa; käytännössä jo nyt.



## Lähteitä ja lisätietoja:

Ananthaswamy, A. 2019. The Quantum Internet Is Emerging, One Experiment at a Time. <https://www.scientificamerican.com/article/the-quantum-internet-is-emerging-one-experiment-at-a-time/>

Brown, M. 2022. The Quantum Threat To Cryptography: Don't Panic, But Prepare Now. Forbes. <https://www.forbes.com/sites/forbestechcouncil/2022/01/11/the-quantum-threat-to-cryptography-dont-panic-but-prepare-now/>

Chen, et al. 2016. Report on Post-Quantum Cryptography. National Institute of Standards and Technology. <https://nvlpubs.nist.gov/nistpubs/ir/2016/nist.ir.8105.pdf>

Dames, A. Richuso, E. 2022. What Is Quantum-Safe Cryptography, and Why Do We Need It? IBM. <https://www.ibm.com/cloud/blog/what-is-quantum-safe-cryptography-and-why-do-we-need-it>

Helsingius, M. 2017. Tutkimuskatsaus kvanttilaskennasta ja kyberturvallisuudesta. Puolustusvoimien tutkimuslaitos. <https://puolustusvoimat.fi/-/tutkimuskatsaus-kvanttilaskennasta-ja-kyberturvallisuudesta>

Mavroeidis, V., Vishi, K., Zych, M., Jøsang, A. 2018. The Impact of Quantum Computing on Present Cryptography <https://arxiv.org/abs/1804.00200>

NIST 2022. Post-Quantum-Cryptography Standardization: <https://csrc.nist.gov/Projects/Post-Quantum-Cryptography/Post-Quantum-Cryptography-Standardization>

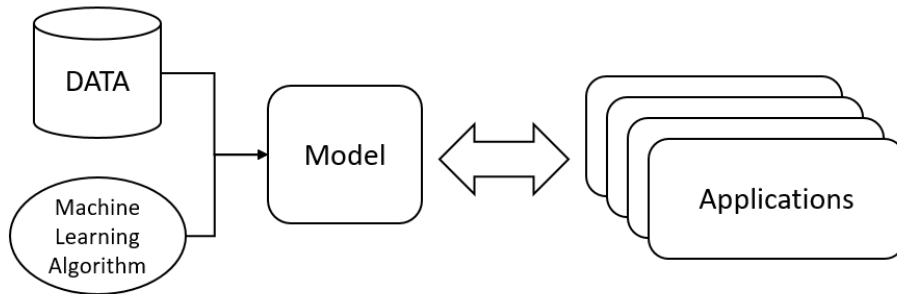
## 2.7 Koneoppiminen, analytiikka



*Jarmo Talvivaara*

Yksi kvanttilaskennan monikäyttöinen sovellusalue on koneoppiminen. Koneoppiminen (eng. machine learning) on tekoälyyn liittyvä tietojenkäsittelyn sovellusalue. Koneoppiminen perustuu datan ja erilaisten algoritmien käyttöön, joiden avulla voidaan rakentaa ohjelmia ilman tarvetta ohjelmoida kaikkia tietojenkäsittelyn sääntöjä etukäteen. Koneoppimisen kautta ohjelmistoihin logiikka rakennetaan opetusdatan avulla, josta koneoppiva algoritmi opettelee datan sisältämät säännönmukaisuudet

ja periaatteet. Yksinkertaistaen, koneoppimisen voidaan ajatella olevan prosessi, jossa koneita opetetaan tunnistamaan kaavoja ja tekemään datan ryhmittelyä, luokittelua, ennusteita tai päätöksiä sen perusteella, mitä opetuksessa käytetystä datasta on pääteltävissä.



Kuva 6. Koneoppimisratkaisun osa-alueita

Koneoppimista hyödynnetään erittäin laajalti monenlaisissa sovelluksissa, esimerkiksi kuvan- ja puheentunnistus (image and speech recognition), luonnollisen kielen käsittely (NLP), ennustava analytiikka (predictive analytics), petostentunnistus (fraud detection), suositusjärjestelmät (recommendation systems). Tämän vuoksi koneoppimisella on erittäin paljon sovelluskohteita ja hyödynnettävää data-analytiikan alueella.

Koneoppimisen avulla pyritään tyypillisesti kehittämään koneita ja ohjelmistoja, jotka oppivat kokemuksistaan ja tekevät parempia päätöksiä ajan mittaan. Tätä tutkitaan ja sovelletaan mm. vahvistusoppimisen (reinforcement learning) ja siirto-oppimisen (transfer learning) alueella.

Kvanttilaskenta voi tehostaa edellä esiteltyä koneoppimisen soveltamista ratkaisemalla ongelmia, jotka ovat perinteisille tietokoneille vaikeita tai jopa mahdottomia. Kvanttilaskentaa hyödyntävästä koneoppimisesta käytetään nimitystä kvanttikoneoppiminen (eng. quantum machine learning, QML).

Esimerkkejä kvanttilaskennan käytöstä koneoppimisessa:

**Kvanttikoneoppimisalgoritmit** (quantum machine learning algorithms): Kvanttikoneoppimisalgoritmit hyödyntävät kvanttimekaniikan periaatteita tehtävien suorittamiseen, kuten luokitteluun, ryhmittelyyn ja regressioon. Esimerkkejä kvanttikoneoppimisalgoritmeista ovat luokitteluongelmissa hyödynnettävä kvanttivektorikone

(Quantum Support Vector Machine), kvanttilaskentaa hyödyntävä, datan ryhmittelyä toteuttavat klusterointialgoritmit (quantum clustering) tai kvanttineuroverkot (quantum neural networks).

**Kvanttilaskennasta vaikutteita saavat klassiset algoritmit** (quantum-inspired classical algorithms): Kvanttilaskennasta vaikutteita saavat - "inspiroituneet" - klassiset algoritmit hyödyntävät kvanttilaskennan periaatteita suunnitellakseen algoritmeja, jotka voivat ratkaista tiettyjä ongelmia tehokkaammin. Esimerkki tällaisesta algoritmista voisi olla esim. kvanttilaskennasta vaikutteita ottanut perinteinen algoritmi lineaaristen yhtälöryhmien ratkaisemiseen.

**"Kvanttiominaisuuskartat"** (quantum feature maps): Kvanttiominaisuuskartat ovat tapa koodata klassinen data kvanttitiloja käyttäen, jotka taas voidaan syöttää kvanttilaskennan koneoppimisalgoritmeihin. Näitä periaatteita voidaan hyödyntää erilaisissa sovelluksissa, kuten esim. lääkeaineiden löytämisessä tai kuvien luokittelussa.

**Kvanttidatan esikäsittely** (quantum data preprocessing): Kvanttidatan esikäsittely sisältää datan valmistelun kvanttilaskennan koneoppimisalgoritmien käyttöä varten. Tämä voi sisältää tehtäviä, kuten datan puhdistus, normalisointi ja ominaisuusvalinta.

Kvanttikoneoppimisen soveltamisia ja teknisiä ratkaisuja tarkastellaan tarkemmin tämän raportin luvussa 5.

### **Lähteitä ja lisätietoja:**

Biamonte, J., & Wittek, P. 2018. Quantum machine learning. <https://arxiv.org/abs/1611.09347>

Beer, K. 2022. Quantum neural networks. <https://arxiv.org/pdf/2205.08154.pdf>

Huang, et al. 2021. Nature. Power of data in quantum machine learning. <https://www.nature.com/articles/s41467-021-22539-9>

IBM. 2022. What is Machine Learning. <https://www.ibm.com/topics/machine-learning>

Jacquier, A., Kondratyev, O., Lipton, A., López de Prado, M. 2022. O'Reilly. Quantum Machine Learning and Optimisation in Finance. <https://learning.oreilly.com/library/view/quantum-machine-learning/9781801813570/>

PennyLane. Key Concepts. <https://pennylane.ai/qml/glossary.html>

PennyLane. What is Quantum Machine Learning? <https://pennylane.ai/qml/whatisqml.html>

PennyLane. Quantum Feature Map. [https://pennylane.ai/qml/glossary/quantum\\_feature\\_map.html](https://pennylane.ai/qml/glossary/quantum_feature_map.html)

Qiskit. 2022. Quantum Machine Learning. <https://qiskit.org/learn/course/machine-learning-course/>

Rebentrost, P., Mohseni, M., & Lloyd, S. (2014). Quantum support vector machine for big data classification. <https://arxiv.org/abs/1307.0471>

Schuld, M. et al. 2014. An introduction to quantum machine learning. <https://arxiv.org/abs/1409.3097>

TensorFlow Quantum. 2021. <https://www.tensorflow.org/quantum>

Wiebe, N., Kapoor, A., & Svore, K. M. 2015. Quantum algorithms for nearest-neighbor methods for supervised and unsupervised learning. Quantum Information & Computation. <https://arxiv.org/abs/1401.2142>

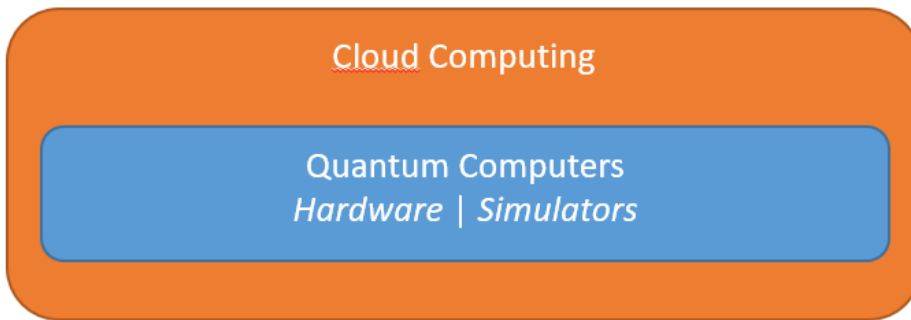
Wikipedia. 2022. Quadratic unconstrained binary optimization. [https://en.wikipedia.org/wiki/Quadratic\\_unconstrained\\_binary\\_optimization](https://en.wikipedia.org/wiki/Quadratic_unconstrained_binary_optimization)

# 3 Quantum Computing-as-a-Service: kvanttilaskenta pilvipalveluna

Jarmo Talvivaara

## 3.1 Johdanto

Tässä raportissa tarkastellaan kvanttilaskennan toteuttamista palveluina (*Quantum-Computing-as-a-Service*, lyh. *QCaaS* tai *QaaS*). Tarkastelussa keskitytään erityisesti verkon kautta saavutettaviin QCaaS –palveluihin eli kvanttilaskentakapasiteetin ja siihen liittyvien teknologioiden tarjoamiseen pilvipalveluna.



QCaaS-mallia tukevissa pilvipalveluissa kvanttilaskentakapasiteetti tarjotaan joko omana, kvanttilaskentaan keskittyneenä ratkaisuna tai osana palvelutarjoajan muuta pilvipalveluinfrastruktuuria. Esimerkkejä ensiksi mainituista ovat esim. D-Waven Leap ja IBM:n Quantum –verkkopalvelut. Esimerkkejä jälkimmäisestä ovat esim. Microsoftin Azure Quantum sekä AWS Braket.

Raportissa on pyritty kartoittamaan ja tarkastelemaan kirjoitushetkellä (kevät 2022 – kevät 2023) yleisimmät kvanttilaskentakyvykkyksiä pilvipalveluina tarjoavat toimijat ja niiden palvelut. Palveluiden osalta esitellään 1) palveluiden yleiskuvaus, 2) tyypillisimmät laitetason kvanttietokone-teknologiat, 3) palvelun käyttöönoton ja käytön perusteet, 3) mahdollisuudet testikäyttöön, 4) hinnoittelun peruserätykset sekä 5) palvelussa tuetut ohjelmointikielien ja muut vastaavat sovelluskehitysteknologiat.

## 3.2 Computing-as-a-Service: perinteiset pilvipalvelut

Pilvipalvelut ovat moderni tapa toteuttaa, tarjota, hankkia ja hyödyntää tietoteknisiä ratkaisuja; esimerkiksi prosessorien tarjoamaa laskentakapasiteettia (compute), tiedon tallennustilaa (storage), verkkoratkaisuja (networks), palvelimia ja sovelluspalveluita (application services), tietokantoja ja tietovarastointia (databases, data warehouses), big data -ratkaisuja sekä muuta datan hallintaan liittyviä palveluita, koneoppimisratkaisuja ja analytiikkaa ja monia muita tietoteknisiä palveluita.

Pilvipalveluista hankittavat laskentaresurssit voidaan jakaa karkeasti ottaen seuraaviin kategorioihin:

- Infrastructure-as-a-Service (IaaS); palvelinten ja muiden tietokoneiden, verkkolaitteiden, tallennuslaitteiden ja muun tarvittavan teknisen infrastruktuurin hankinta omien palvelinsalien sijaan pilvipalveluna. Etuina on mm. nopea infrastruktuurin ja laskentakapasiteetin hankinta ja skaalaus, mahdollisuus rakentaa infrastruktuurin päälle juuri tarkoituksen mukaiset verkot, palvelut ja muut ratkaisut. Lisäksi etuina laitteiden hankinnasta säästetyt kulut sekä laitteiston ylläpitoon kohdistuvan työn säästyminen. Haasteina ovat usein korkeammat käyttökustannukset pitkällä aikavälillä ja tarve oman infrastruktuurin ylläpitoon käyttöjärjestelmätasosta alkaen (esim. palvelinsovellukset, verkkoasetukset). Myöskään laitetason toimenpiteitä ei IaaS- ratkaisuisissa päästä tekemään vaan esim. pilvi-infrastruktuurin luodut tietokoneet ovat tyypillisesti palveluntarjoajan konesalien ympäristöissä ajettavia virtuaalikoneita ja niiden tallennuskapasiteettina käyttämiä massamuistiratkaisuja.
- Platform-as-a-Service (PaaS): erilaisten palveluiden hankinta pilvipalveluna sen sijaan, että toteutettaisiin omilla palvelimilla ja niissä asennetuissa/ylläpidettävissä palvelinsovelluksissa. Esimerkkejä PaaS- ratkaisuisista ovat esim. pilvitietokannat, joissa omat tietokantapalvelimet korvataan pilvipalvelusta suoraan saatavana tietokantana. Muita esimerkkejä ovat esim. erilaiset web- tai sovellus-, koneoppimis-, analytiikka-, internet of things- tai integraatio- palvelut, "alustat".
- Software-as-a-Service (SaaS): yksittäisten tai useampien järjestelmien ja muiden ohjelmistojen hankinta pilvipalveluna. Tässä mallissa niin teknisen laiteinfrastruktuurin, että sen päällä olevien palveluiden mutta myös sovelluksen ylläpidon tarpeet ovat ulkoistettu palveluntarjoajalle. Esimerkkejä tästä voisivat olla esim. yrityksen verkkokauppa-portaali, asiakkuudenhallinta- tai

toiminnanohjausjärjestelmät, selainkäyttöliittymällä varustettu toimistosovellus, sähköpostipalvelu tai julkishallinnon dokumenttienhallintajärjestelmä.

Perinteisesti erityyppiset pilvipalvelut (IaaS, PaaS, SaaS) ovat toteutettu käyttäen ns. klassiseen laskentaan kykeneviä perinteisiä tietokoneita ja niissä olevia laskentayksiköitä (CPU, GPU, TPU).

- **CPU (Central Processing Unit):** CPU on tietokonejärjestelmän ensisijainen prosessointiyksikkö. Se vastaa ohjeiden suorittamisesta sekä aritmeettisten ja loogisten operaatioiden suorittamisesta käsiteltävälle tiedolle. Suorittimet on suunniteltu käsittelemään monenlaisia tehtäviä perustoiminnoista, kuten yhteen- ja vähennyslasku, monimutkaisempiin tehtäviin, kuten sovellusten ja käyttöjärjestelmien suorittamiseen.
- **GPU (Graphics Processing Unit):** GPU on erikoistunut prosessointiyksikkö, joka on suunniteltu käsittelemään monimutkaista grafiikkaa ja muuta kuvan käsittelyssä tarvittavaa laskentaa. GPU:ita käytetään yleisesti peleissä, videoeditointissa ja muissa sovelluksissa, jotka vaativat korkean suorituskyvyn grafiikan käsittelyä. Toisin kuin CPU- prosessoreissa, joissa on tyypillisesti pieni määrä prosessointiytimiä, GPU:ssa on monia pieniä prosessointiytimiä, jotka voivat toimia rinnakkain ja suorittaa täten useita laskentaoperaatioita samaan aikaan. Tämän vuoksi niitä voidaan soveltaa hyvin myös koneoppimisen tarpeisiin, esim. neuroverkkojen opettamiseen.
- **TPU (Tensor Processing Unit)** on Googlen kehittämä erikoistunut käsittely-yksikkö koneoppimislaskelmien suorittamiseen. TPU:t on suunniteltu erityisesti nopeuttamaan koneoppimiseen liittyvää tiedonkäsittelyä ja ne sopivatkin erityisen hyvin syvien neuroverkkojen opettamiseen.

On hyvä huomioida, että pilvipalveluiden taustalla olevat erittäin tehokkaat ja laajat konesalit sekä esim. HPC – eli suurteholaskentaan kykenevät laskentakeskukset eivät välttämättä sisällä oletuksena kvanttilaskentapiirejä tai muuten kapasiteettia kvanttilaskentaa vaativien ohjelmistojen suorittamiseen kvanttietokonelaitteistoilla. Todella mittavasta laskentakapasiteetista huolimatta, ne loppujen lopuksi sisältävät ne samat haasteet, mitä perinteinen tietotekniikka ja klassinen laskenta; riittävän haastavat laskentaoperaatiot muodostuvat ongelmaksi maailman suurimmillekin laskentakapasiteeteille – ja näihin ongelmiin kvanttilaskenta ja kvanttietokoneet voivat myös pilvipalveluiden osalta tuoda ratkaisun.

Nämä edut eivät ole vielä toistaiseksi (-2022) yleispätevästi saavutettavissa ns. NISQ- aikakauden kvanttietokoneiden rajoitusten vuoksi. Kuitenkin

kvanttietokoneet jo tällä hetkellä voivat soveltua tietyn tyyppisten erityisongelmien ratkaisemiseen perinteistä suurteholaskentaa tehokkaammin. Tällöin ratkaistavaksi haasteeksi jääkin se, miten kvanttietokoneita pääsisi käyttämään yhtä joustavasti, kuten perinteisen tietotekniikan avulla tarjottuja ratkaisuja.

### 3.3 Quantum Computing-as-a-Service: kvanttilaskentaa pilvipalveluna?

Kvanttietokoneet hyödyntävät laskentaoperaatioissa QPU- piirejä. QPU (Quantum Processing Unit), kvanttiprozessointiyksikkö, on kvanttietokoneen keskeinen komponentti, joka käyttää kvanttimekaniikkaa laskelmien suorittamiseen. QPU on fyysinen piiri, joka koostuu toisiinsa kytkeytyneistä kubiteista, joiden avulla kvanttilaskentaoperaatiot suoritetaan.

Eri QPU:t (tarkemmin: niiden sisältämät kubitit) voivat olla toteutettu eri teknologioilla, esim. suprajohtavat, ioniloukut, ftoniikka-pohjaiset ratkaisut tai neutraaliatomit. Toisin kuin perinteiset tietokoneet, jotka käyttävät bittejä edustamaan tietoa, kvanttietokoneet käyttävät kubitteja, jotka voivat esiintyä useissa tiloissa samanaikaisesti.

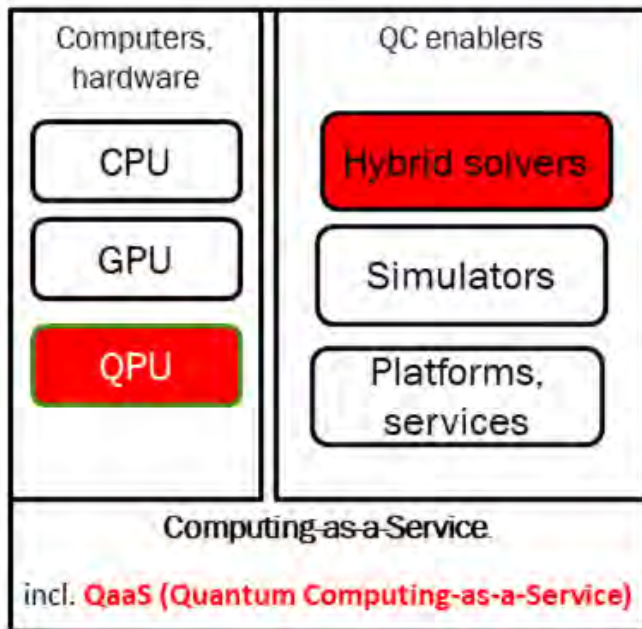
Klassinen laskenta	Kvanttilaskenta
CPU, GPU, TPU	QPU
Laskenta perustuu bittien käyttämiseen. Yksittäinen bitti voi saada arvon 0 tai 1	Laskenta perustuu kvanttibittien, ns. kubitien käyttämiseen. Yksittäinen kubitti voi saada minkä tahansa arvon väliltä 0 – 1
Perustuu klassiseen fysiikkaan	Perustuu kvanttifysiisiin ilmiöihin

QPU- laskentapiirejä sisältävien kvanttietokoneiden kehitys, hankinta ja käyttöön-otto on kallista, joten oman paikallisen kvanttietokoneen ostaminen esim. yritys- tai henkilökohtaiseen käyttöön on usein mahdotonta. Tämän haasteen ratkaisemiseksi on saatavilla pilvipalveluina tarjottavia kvanttilaskentaratkaisuja, joiden avulla erilaiset käyttäjät voivat hyödyntää kvanttilaskentaa ilman tarvetta omistaa ja ylläpitää fyysistä kvanttikonetta. Kvanttilaskentaa tarjoavat pilvipalvelut voidaan yleensä ajatella olevan joko Infrastructure-as-a-Service (IaaS) tai Platform-as-a-Service (PaaS) -tyyppisiä ratkaisuja – riippuen tarjotun palvelun ominaisuuksista;



- mahdollistetaanko pääsy suorittamaan ohjelmia suoraan hardware –laitetalle aidolla kvanttietokoneella?
- tarjotaanko palvelussa pääsy yhteen tai useampaan (eri valmistajien, erityyppiin) kvanttietokoneeseen?
- tarjotaanko fyysisen laitteiston lisäksi simulaattoreita tai muita tapoja ohjelmien testaamiseen?
- sisältääkö pilvipalvelu laitteiston lisäksi muita ohjelmistoja kvanttilaskennan ja kehitystyön tukemiseen? esim.
  - SDK-kehitysympäristöjä, kirjastoja, sovelluskehyskiä, käyttäjähallintaa,
  - suoritusjonoja tai muuta kapasiteetin hallintaa,
  - klassisen ja kvanttilaskennan yhdistelmää tukevia ominaisuuksia (esim. hybrid solvers)

Pilvipalveluna tarjottavista kvanttilaskentapalveluista käytetään usein nimitystä Quantum Computing-as-a-Service. Termi lyhennetään usein muotoon “QCaaS” tai “QaaS”.



Kuva 7. Esimerkki QCaaS- ominaisuuksia tarjoavan pilvipalvelun rakenteesta

Käytännössä QCaaS- palvelut tarjoavat mahdollisuuksia kehittää ja suorittaa kvantti-ohjelmia pilvipalvelussa olevilla QPU- prosessointiyksiköillä tai muilla kvantti-suorittimilla. Tämä mahdollistaa kvanttilaskentakapasiteetin hankkimisen ja käytön eri käyttötarkoituksiin, usein käytön mukaan hinnoiteltuna, ilman mittavia hankinta- tai

muita käyttöönottokustannuksia. QCaaS -palvelut ovat sen vuoksi käytännöllisiä myös siinä tapauksessa, kun halutaan nopeasti kokeilla kvanttilaskennan mahdollisuuksia ilman, että käytössä täytyy olla oma kvanttietokone tai kun halutaan esim. testata useamman erityyppisen kvanttietokoneen toimivuutta ohjelmien suorituksessa.

Voi olla tietenkin tilanteita, joissa oman kvanttietokoneen omistaminen voi olla käytännöllisempi ja myös taloudellisempi verrattuna pilvipalveluna käyttöön; esim. pitkän aikavälin käytön kustannukset voivat olla merkittävät ja joissain tilanteissa haastavampia ennakoita tai jos on tarve tutkia ja kehittää kvanttietokonetta laitetasolla. Myös erilaiset tietosuoja- tai tiedonkäsittelyn säädökset voivat vaikuttaa siihen, miten pilvipalveluna saatavaa kvanttilaskentakapasiteettia voidaan hyödyntää; pilvipalveluna saatavat kvanttietokoneet voivat sijaita fyysisesti jossakin sellaisella alueella (maa, maanosa), jonne esim. luottamuksellisen tiedon tallentaminen ja käsittely ei ole reguloinnin puitteissa mahdollista.

Useampi teknologiavalmistaja ja palvelutarjoaja (esim. Amazon, Microsoft, Google) onkin päätenyt tarjoamaan kvanttilaskentakapasiteettia integroimalla kvanttietokoneita omiin pilvipalveluihinsa. Tällöin pilvipalvelussa saatavilla kvanttilaskentakapasiteetti on saavutettavissa lähes samoin periaattein kuin pilvipalveluissa perinteisesti muuten tarjotut resurssit (esim. virtuaalikoneet, web-palvelut, tietokannat ja muut datan hallintaratkaisut tai koneoppimis- tai analytiikkaratkaisut). Näissä palveluissa voidaan verkon kautta "pilvestä" hankkia vain kulloinkin tarvittava kvanttilaskentakapasiteetti, usein käyttäjän ja kvanttietokoneiden maantieteellisistä sijainneista riippumatta ja maksaen vain kvanttietokoneen laskenta-ajasta (pay-as-you-go). Näitä palveluita esitellään tarkemmin raportissa myöhempanä.

Eri palveluntarjoajat mahdollistavat ainakin yhden mutta usein myös useamman erityyppisten (eri teknologioilla toteutettujen) kvanttietokoneiden käytön.

Esimerkkejä pääsääntöisesti yhden tyyppisen kvanttietokoneen tarjoamisesta ovat esim.

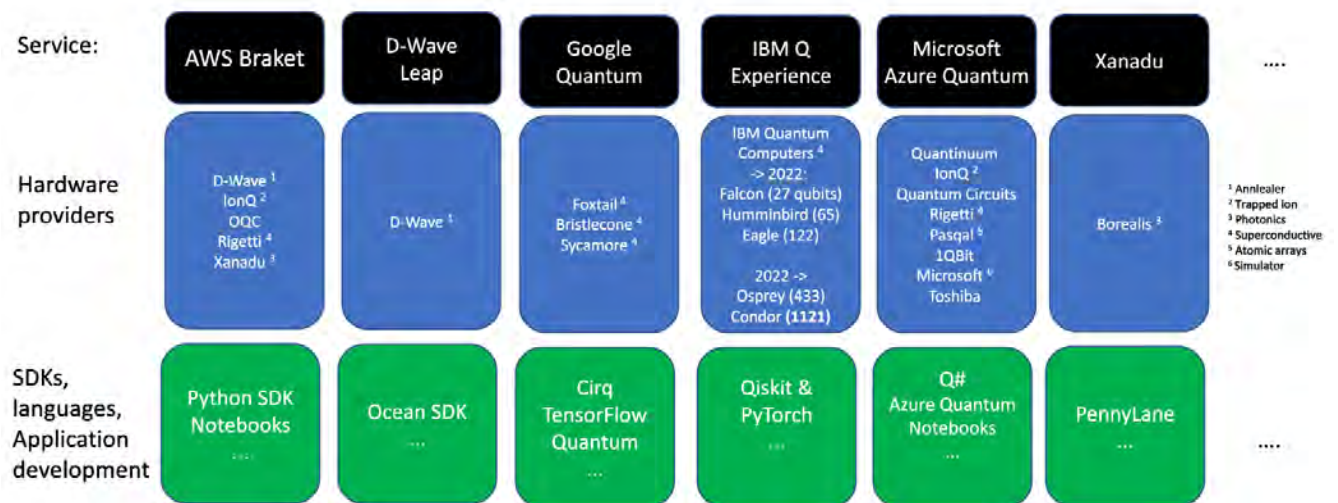
- D-Waven Leap (annealer- tyyppiset koneet)
- IBM Quantum (superconductive QPU eli suprajohtavilla piireillä toteutetut kubitit).

Useampaa eri tyyppistä kvanttietokonelaitteistoa tarjoavat esim.

- Microsoftin Azure-pilvipalvelun Quantum Workspacet mahdollistavat hyödyntää esim. Rigettin superconductive-, IonQ:n trapped ion- tai Pasqal:in neutral atom- tekniikoilla toteutettuja koneita.
- AWS Braket vastaavasti tarjoaa Rigettin ja IonQ:n kvanttietokoneiden lisäksi pääsyn esim. D-Waven annealer- tai Xanadun ftoniikka -tekniikoilla toteutettujen koneiden käyttöön.

QCaaS- palvelut sisältävät usein valmiita kvanttialgoritmeja ja -ohjelmointikirjastoja, joiden avulla käyttäjät voivat kehittää uusia kvanttilaskennan sovelluksia. QCaaS- palvelut käyttävät yhden tai useamman valmistajan tai eri teknologioilla valmistettuja tietokoneita, joilla varsinainen laskentakapasiteetti toteutetaan.

## QaaS (Quantum Computing As a Service)



Kuva 8. Esimerkkejä QCaaS- pilvipalveluista ja niiden ominaisuuksista (AWS 2022, D-Wave 2022, Google AI 2022, IBM Quantum 2022, Microsoft 2022, Xanadu 2022)

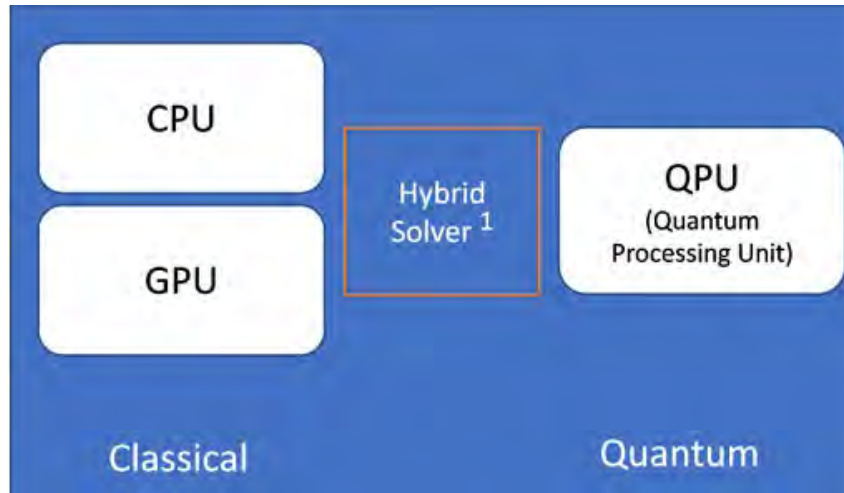
## 3.4 Perinteistä ja kvanttilaskentaa pilvipalvelussa?

QCaaS- pilvipalveluna tarjottavissa kvanttilaskentapalveluissa on usein tarve hyödyntää QPU- piirien lisäksi myös CPU- tai muita perinteisiä, "klassisia" laskentaratkaisuja" (GPU, TPU). Jos CPU- ja QPU- resursseja hyödynnetään yhteistyössä, voidaan tällaisista yhdistelmistä käyttää nimitystä "quantum-classical-hybrid" tai "hybrid quantum computing".

Yhtenä esimerkkinä tästä on annealer -tyyppisiä kvanttietokoneita valmistavan D-Wave:n tarjoaman Leap-palvelun tarjoama ns. "hybrid solver". Hybrid solver on eräänlainen laskenta-algoritmi, joka yhdistää perinteisen tietokoneen ja kvanttikoneen kyvyt ratkaistakseen monimutkaisia laskentaongelmia. Hybrid solver käyttää kvantti- ja klassista laskentaa yhdistämällä kvanttietokoneen ja perinteisen tietokoneen kuhunkin käyttötarkoitukseen parhaiten soveltuvat laskentaominaisuudet.

- Kvanttietokoneet ovat erittäin tehokkaita tietyntyyppisten ongelmien ratkaisemisessa (esim. optimointi- tai simulointiongelmat), mutta vielä tällä hetkellä rajoittuneita monimutkaisempien ongelmien ratkaisemiseen (esim. kubitien vähäisen määrän tai laadun vuoksi).
- Perinteiset tietokoneet kykenevät toistaiseksi vielä monimutkaisempien ja pidempien ongelmien ratkaisuun (algoritmien suorittamiseen), mutta kohtavat väistämättä merkittäviä haasteita tehokkaiden ratkaisujen tuottamiseen niissä ongelmissa, joihin kvanttietokoneet kykenevät.

Hybrid solver- tyyppisillä ratkaisuilla voidaan saavuttaa etuja molemmantyyppisistä tietokoneista: hybrid solver "ohjaa" ongelmanratkaisua niin, että kvanttietokone (QPU) ratkaisee suuren osan ongelmasta (esim. optimointi), jonka jälkeen tulokset tai muu tarvittava tiedonkäsittely välitetään perinteiselle tietokoneelle (CPU) ratkaisun loppuun saattamiseen (esim. tulosten validointi).



Kuva 9. (Soveltaen) D-Wave Leap, hybrid solvers, (D-Wave 2021)

<sup>1</sup>Lisätietoja D-Waven Leap- palvelun Ocean -kehitysympäristön hybrid solver- ominaisuuksista löytyy sivustolta <https://docs.ocean.dwavesys.com/en/stable/overview/hybrid.html>

Hybrid solver -tekniikka tarjoaa myös kehittäjille joustavuutta, sillä he voivat valita, kuinka paljon kvanttilaskentaa haluavat käyttää ongelman ratkaisemiseen ja mitkä osat ongelmanratkaisusta kannattaa suorittaa perinteisillä tietokoneilla.

Edellä esitetty tapa mahdollistaa esim. vain tiettyyn käyttötarkoitukseen soveltuvien kvanttietokoneiden (mm. annealer) tai nykyisen NISQ- aikakauden ominaisuuksiltaan rajoittuneempien koneiden (kubittien vähäinen määrä, laatu tai porttioperaatioiden nopeus) hyödyntämiseen erilaisten kohdennettujen ongelmien ratkaisuun (ks. esim. Preskill 2018)

### Lähteitä ja lisätietoja:

Caulfield, B. 2009. What's the Difference Between a CPU and a GPU. NVIDIA. <https://blogs.nvidia.com/blog/2009/12/16/whats-the-difference-between-a-cpu-and-a-gpu/>

D-Wave. 2020. D-Wave Hybrid Solver Service: An Overview. [https://www.dwavesys.com/media/4bnpi53x/14-1039a-b\\_d-wave\\_hybrid\\_solver\\_service\\_an\\_overview.pdf](https://www.dwavesys.com/media/4bnpi53x/14-1039a-b_d-wave_hybrid_solver_service_an_overview.pdf)

IBM. 2022. IaaS vs. PaaS vs. SaaS. <https://www.ibm.com/topics/iaas-paas-saas>

Thom, M. 2019. Three Truths and the Advent of Hybrid Quantum Computing. Medium. <https://medium.com/d-wave/three-truths-and-the-advent-of-hybrid-quantum-computing-1941ba46ff8c>

Wikipedia. 2022. Tensor Processing Unit. [https://en.wikipedia.org/wiki/Tensor\\_Processing\\_Unit](https://en.wikipedia.org/wiki/Tensor_Processing_Unit)

## 3.5 Kvanttilaskentapalveluita ja – palveluntarjoajia (QCaaS services and providers)

Alla olevassa koosteessa esitellään hankkeen aikana vuosien 2020 – 2022 aikana saatavilla olevia kvanttilaskentapalveluita, joiden käyttöön pääsääntöisesti on ilmainen pääsy tutustumista ja ohjelmien kehittämisen testaamista varten. On kuitenkin hyvä huomioida, että listassa on vain osa hankkeen aikana julkaistuja palveluita.

Lisäksi palveluita julkaistaan ja kehitetään erittäin nopealla vauhdilla, joten etenkin tarkemmat ja ajantasaiset tiedot (palveluiden yleiset ominaisuudet, tuki eri kvanttietokoneille, pääsy palvelun käyttöön, hinnoittelun periaatteet ja kustannusten muodostuminen, tuetut SDK- kehitysympäristöt, ohjelmointikielet ja –kirjastot sekä muut teknisiin detaljit), kannattaa tarkistaa kunkin palveluntarjoajan sivustolta.

Linkit näille sivustoille löytyy alla olevasta listasta sekä kuhunkin kohteeseen erikseen alempana palveluntarjoajittain listattuna.

Palvelut ja palveluntarjoajat on listattu aakkosjärjestyksessä.

1. Amazon AWS Braket, <https://aws.amazon.com/braket/>
2. DWave Leap, <https://cloud.dwavesys.com/leap/>
3. Google Quantum AI ja Quantum Computing Service, <https://quantumai.google/>
4. IBM Quantum, <https://quantum-computing.ibm.com/>
5. Microsoft Azure Quantum, <https://azure.microsoft.com/en-us/services/quantum/>
6. Oxford Quantum Circuits, <https://oxfordquantumcircuits.com/>

7. QC Ware Forge, <https://forge.qcware.com/>
8. Xanadu Cloud, <https://www.xanadu.ai/>

### 3.5.1 Amazon Braket

AWS on yhdysvaltalaisen Amazonin tytäryhtiön, Amazon Web Services Inc.:n tarjoama pilvipalvelualusta. AWS pohjautuu isoon määrään maantieteellisesti hajautettuja konesaleja, joiden avulla toteutettu laskentakapasiteetti on hallittavissa AWS:n verkkoportaalin kautta sekä hyödynnettävissä Internetin yli erilaisia rajapintoja hyödyntäen.

AWS –pilvipalvelun portaali löytyy osoitteesta: <https://aws.amazon.com/>

AWS tarjoaa ison joukon erilaisia palveluita (eng. "services"), esim.

- laskentapalvelut- ja resurssit (compute): esim. virtuaalikoneet (esim. EC2), web-palvelut ja -sovellukset, container- ympäristöt.
- tallennus (storage): virtuaalikoneiden ja muiden palveluiden käyttämät tallennustilat (esim. S3)
- tietokannat ja datan hallinta: esim. tietokannat, tietovarastot, big data- varannot.
- verkot (networking)
- koneoppiminen (machine learning)
- analytiikkaratkaisut (analytics)

Kattavammin lisätietoa AWS:n palveluista löytyy osoitteesta: <https://aws.amazon.com/products/>

Yllä esitetyt AWS:n palvelut ovat valtaosin ns. perinteisellä laskentakapasiteetilla (CPU, GPU) toteutettuja, eivätkä suoraan sisällä kvanttilaskentakyvykkyyksiä.

AWS kuitenkin julkaisi vuonna 2021 Amazon Braket – nimisen kvanttilaskentapalvelun yhdistettynä muuhun AWS- palvelutarjontaan. Amazon Braket- palvelusta tietoa löytyy osoitteesta: <https://aws.amazon.com/braket/>

Amazon Braket tarjoaa erilaisia kehitystyökaluja, mahdollisuuden kehitettyjen ohjelmistojen ajamisen simulaattoreilla tai aidoilla kvanttietokoneilla (vaihtoehdot esiteltä alla tarkemmin), perinteisen ja kvanttilaskennan yhdistelyä hyödyntävien

sovellusten suorittamiseen (ns. Hybrid Jobs), sekä kattavan joukon ohjeita, dokumentaatiota ja muita työskentelyä tukevia resursseja.

Amazon Braket tarjoaa monipuolisesti eri tapoja ja ominaisuuksia kvanttilaskennan soveltamiseksi eri alueilla:

Tyypillisiä käyttötapoja Amazon Braketille ovat muun muassa:

- **Kvanttisimulaatiot:** Tutkijat voivat käyttää Amazon Braketia kvanttijärjestelmien simuloimiseen ja niiden käyttäytymisen tutkimiseen esimerkiksi materiaalitieteessä, kemiassa ja lääkekehityksessä. Ks: <https://aws.amazon.com/braket/quantum-simulators/>
- **Kvanttialgoritmit:** Kehittäjät voivat käyttää Amazon Braketia kvanttialgoritmien toteuttamiseen ja suorittamiseen eri käyttötarkoituksiin, kuten kvanttikoneoppiminen, kvanttioptimointi ja kvanttikryptografia, Ks lisätietoja: <https://aws.amazon.com/braket/quantum-algorithms/>
- **Kvanttipiirien suunnittelu:** Käyttäjät voivat suunnitella, testata ja optimoida kvanttipiirejä Amazon Braketilla tutkiakseen ja kokeillakseen erilaisia kvanttiportteja ja -piirejä sekä niiden vaikutuksia kvanttijärjestelmiin. Ks. lisätietoja: <https://aws.amazon.com/braket/quantum-circuit-design/>
- **Kvantti-klassinen-hybridilaskenta:** Amazon Braket tukee laskentaa, jossa käyttäjät voivat yhdistää klassisia ja kvanttitekniikan laskentaresursseja ratkaistakseen ongelmia, jotka vaativat sekä klassista että kvanttitekniikan avulla toteutettua laskentaa. Ks. lisätietoja: <https://aws.amazon.com/braket/hybrid-quantum-classical-computing/>
- **Kvanttilaskennan tutkimus ja opetus:** Amazon Braket tarjoaa työkaluja ja resursseja opettajille ja tutkijoille kvanttitekniikan opetukseen ja tutkimukseen, mukaan lukien pääsyn kvanttietokoneiden laitteistoon, -ohjelmistoon ja dokumentaatioon. Ks. lisätietoja: <https://aws.amazon.com/braket/education-and-research/>

Amazon Braketia on hyödynnetty mm. autoteollisuudessa, biotekniikan alueella, energiatekniikoiden kehityksessä, vakuutuslalla sekä koneoppimiskäytöiden kehittämisessä.

Tarkemmin Amazon Braketin ominaisuuksia löytyy esiteltynä sivustolta: <https://aws.amazon.com/braket/features/>

Esimerkkejä Amazon Braketin käyttötapauksista eri sovelluskohteisiin: <https://aws.amazon.com/braket/customers/>



## Amazon Braket:in hardware- ratkaisut (tuetut kvanttietokonelaitteistot):

Amazon Braket mahdollistaa useampien erilaisten kvanttietokoneiden käytön kvanttilaskennassa sovellettavaksi:

- IonQ (trapped-ion)
- Rigetti (superconducting qubits)
- OQC/Oxford Quantum Circuit (superconducting qubits)
- QuEra (Rydberg atom qubits)
- Xanadu (photonics based quantum computers)
- Lisäksi lisätoimintojen kautta on saatavilla tuki mm. D-Wave:n annealers-tyyppisten koneiden hyödyntämiseen.

Lisätietoja Amazon Braketin sisältämisestä kvanttietokoneista tarkemmin löytyy osoitteesta: <https://aws.amazon.com/braket/quantum-computers/>

## Palvelun käytön vaihtoehdot, hinnoittelu ja käyttöönotto

AWS tarjoaa Amazon Braket- palvelun käyttöön useita eri hinnoittelumalleja ja -tapoja:

- Käyttöönoton ja tutustumisen kynnystä madalletaan **AWS Free Tier with Amazon Braket** -optiolla, joka mahdollistaa palvelun käyttöön ja sovelluskehitykseen tutustumisen ilmaiseksi. Free Tier sisältää kuukausittaisen yhden tunnin kiintiön laskentakapasiteetin käyttöön. Lisätietoja: [https://aws.amazon.com/braket/pricing/?loc=ft#AWS\\_Free\\_Tier](https://aws.amazon.com/braket/pricing/?loc=ft#AWS_Free_Tier)
- Edellä mainitun ilmaisen kokeilujakson lisäksi palvelua voi käyttää maksullisesti Pay-as-you-go- eli "maksat sen mukaan mitä käytät" -periaatteen mukaan; kustannukset palvelun käytöstä tulee käytettyjen palveluiden, ohjelmien suoritukseen valittujen simulaattoreiden ja aitojen kvanttietokoneiden sekä mahd. muiden tarvittavien resurssien (mm. tallennustila) mukaisesti. Hinnat ja täten käytännön kustannukset voivat vaihdella esim. eri valmistajien (IonQ, Rigetti, Xanadu, jne.) kvanttietokoneiden käytön kesken.
- AWS tarjoaa myös erilaisia optioita mm. Amazon Braketin tutkimuskäyttöön: ks. Lisätietoja <https://aws.amazon.com/braket/quantum-computing-research/>

Amazon Braketin hinnoittelurakenne sisältää kustannuksia kvanttilaitteiden, simulaattoreiden ja muiden palveluiden käytöstä. On tärkeää tarkastella ja ymmärtää

Amazon Braketin hinnoittelutiedot varmistamaan, että on jo etukäteen tietoinen palvelun käytöstä aiheutuvista kustannuksista.

Amazon Braketin hinnoittelusta tarkemmin tietoa on saatavilla sivustolta: <https://aws.amazon.com/braket/pricing/> Sivustolla on myös esimerkkejä erilaisista käyttötapauksista ja niiden kustannusten muodostumisesta, jota voi käyttää pohjana suunniteltaessa Amazon Braketin resurssien käyttöä omiin käyttötarkoituksiin.

Hinnoittelun apuna voi käyttää AWS Pricing Calculator- sivustoa: <https://calculator.aws/>

Kuten yllä mainittu, AWS tarjoaa ilmaisen kokeilumahdollisuuden Amazon Braket – palvelun käyttämiseen. Tästä lisätietoja löytyy osoitteesta: [https://aws.amazon.com/braket/pricing/?loc=ft#AWS\\_Free\\_Tier](https://aws.amazon.com/braket/pricing/?loc=ft#AWS_Free_Tier)

Amazon Braketin käytön aloittamisen ohjeistus on saatavilla sivustolta: <https://aws.amazon.com/braket/getting-started/>

Amazon Braketin käyttöönotto ilmaiseen testaukseen on tehtävissä karkeasti ottaen seuraavilla toimenpiteillä:

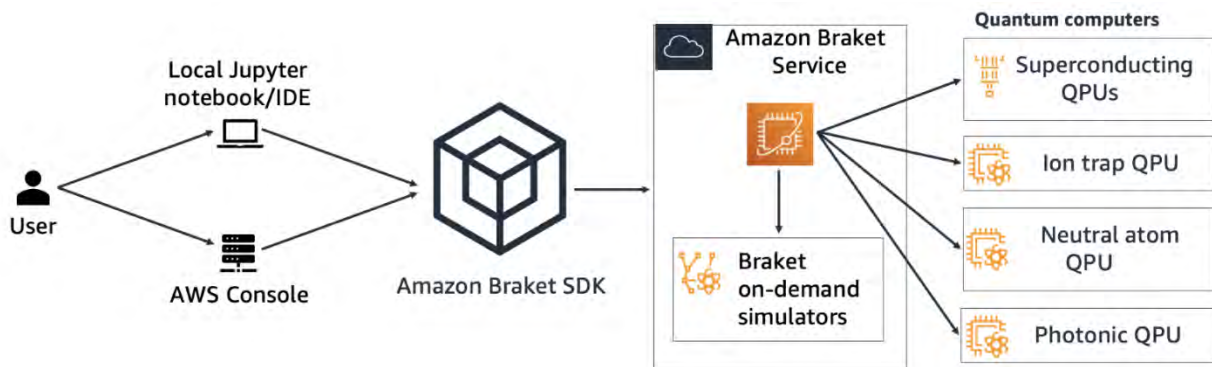
1. Ellei jo käytössä, on luotava tili AWS-pilvipalveluun
  - Tämä onnistuu osoitteessa <https://aws.amazon.com/> ja siellä on tarve antaa tarvittavat tiedot, mukaan lukien maksutiedot.
2. Kun sinulla on AWS-tili, voidaan seuraavaksi siirtyä Amazon Braket – palveluun:
  - Tämä onnistuu joko AWS Management Console – kautta osoitteessa <https://aws.amazon.com/braket/> tai
  - suoraan Braket-konsoliin osoitteessa <https://console.aws.amazon.com/braket/>
3. Otettaessa Amazon Braket – palvelua käyttöön, kannattaa tietenkin tutustua tarjolla olevaan dokumentaatioon, tutoriaaleihin ja muihin resursseihin saadakseen käsityksen palvelusta ja sen ominaisuuksista. esim. Braketin hyödyntämistä erilaisista kvanttilaskennan työkaluista ja palveluista, kuten simulaattoreista, kvanttisuorittimista ja kehitysympäristöistä.

- Dokumentaatiota yllä mainituista asioista löytyy esim. sivustolta: <https://aws.amazon.com/braket/getting-started/> sekä <https://aws.amazon.com/braket/quantum-solutions-built-on-aws/>

4. Seuraavaksi tyypillisesti valitaan kvanttilaskennassa käytettävä laite tai simulaattori:

- Amazon Braket tarjoaa pääsyn erilaisiin kvanttilaskennan laitteisiin eri valmistajilta, kuten D-Wave, Rigetti ja IonQ.
- Valitse kvanttilaskennan laite, joka sopii tarpeisiisi perustuen sen kykyihin, hinnoitteluun ja saatavuuteen.
- Laitteista ja simulaattoreista lisätietoja on saatavilla: <https://aws.amazon.com/braket/quantum-computers/>

5. Amazon Braket –konsolilla (tai käyttäen paikallisia kehitystyökaluja) voidaan luoda ja konfiguroida kvanttitehtäviä, joita voidaan suorittaa valitulla kvanttilaskennan laitteella.



Kuva 10. Amazon Braket –konsolin toiminta. (Kuva: Amazon)

6. Kun kvanttitehtäviä luodaan, voidaan ne lähettää sen suoritettavaksi valitulla kvanttilaskennan laitteella.

- Tehtävän etenemistä voidaan seurata, tarkastella tuloksia ja analysoida tuloksia.
- Esimerkkejä tästä löytyy mm. <https://docs.aws.amazon.com/braket/latest/developerguide/braket-get-started-run-circuit.html>

Amazon Braketin käyttö perustuu joko AWS –portaalin tarjoaman AWS Consolen (selainpohjainen käyttöliittymä) tai paikallisesti asennettujen kehitystyökalujen (Jupyter Notebook) avulla. Molemmat tavat hyödyntävät AWS:n tarjoamaa Amazon Braket SDK -kehitysympäristöä, joka on yhteydessä Amazon Braket- pilvipalveluun. Pilvipalvelu hoitaa asetusten mukaisesti ohjelmien testaamisen ja suorittamisen joko simulaattoreilla tai palvelusta saatavilla olevilla, käyttäjän valitsemilla kvanttietokoneilla.

### **Tuettuja SDK- kehitysympäristöjä ohjelmointikieliä, kirjastoja ja sovelluskehityksiä**

AWS tarjoaa "Amazon Braket SDK"- kehitysympäristön sovelluskehityksen tueksi.

Yksityiskohtainen ohjeistus oman kehitysympäristön pystyttämiseen Amazon Braket SDK:ta sekä pilvipalveluiden käyttämiseksi on saatavilla osoitteessa:


<https://aws.amazon.com/blogs/quantum-computing/setting-up-your-local-development-environment-in-amazon-braket/>

Ohjeessa esitellyn Amazon Braket SDK -kehitysympäristön dokumentaatio on saatavilla osoitteessa: <https://github.com/aws/amazon-braket-sdk-python> Amazon Braket SDK sisältää tuen Python -kielisten kvanttiohjelmien kirjoittamiseen sekä suorittamiseen Braketin tarjoamilla simulaattoreilla sekä aidoilla kvanttietokoneilla.

### **3.5.2 D-Wave Leap**


Leap on D-Wave Systems Inc. -yhtiön vuonna 2018 julkaisema pilvipohjainen alusta D-Wave:n annealer- tyyppisten kvanttietokoneiden etäkäyttämiseen. D-Wave Leap- palvelu on käytettävissä osoitteesta <https://cloud.dwavesys.com/leap> Sivustolla on mahdollista luoda ilmainen tunnus Leap- palvelun testikäyttöön.

## Getting Started DISMISS x




INTERACT WITH OUR DEMOS

Find out how D-Wave quantum computers work by running our brief demos.



EXPLORE OUR EXAMPLES

Learn from example applications or build your own in our Leap IDE.



ACCESS OUR RESOURCES

Demos, documentation, and tools to help you harness the power of quantum computing.

**What's New** >

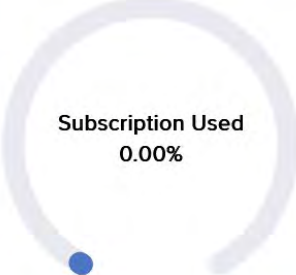
D-WAVE Leap

**CQM SOLVER UPDATES**

Constrained Quadratic Model Solver Update: Now Supporting Weighted Constraints and Presolve Techniques

November 4, 2022

### Monthly Subscription Usage Summary ?



Subscription Used  
0.00%

Usage Details

Hybrid Solvers Total (all regions) ?

**00 h 00 m 00.000 s**

TIME USED

**0** ?

PROBLEMS SUBMITTED

Updated every minute.

GET MORE TIME

### Problem Status ?

FILTER ↻

Status of your last 1000 problems

Problem Label <span style="float: right;">?</span>	Submitted On (UTC) <span style="float: right;">?</span>	Ended <span style="float: right;">?</span>	Status
dwave sample			Completed
dwave ping			Completed

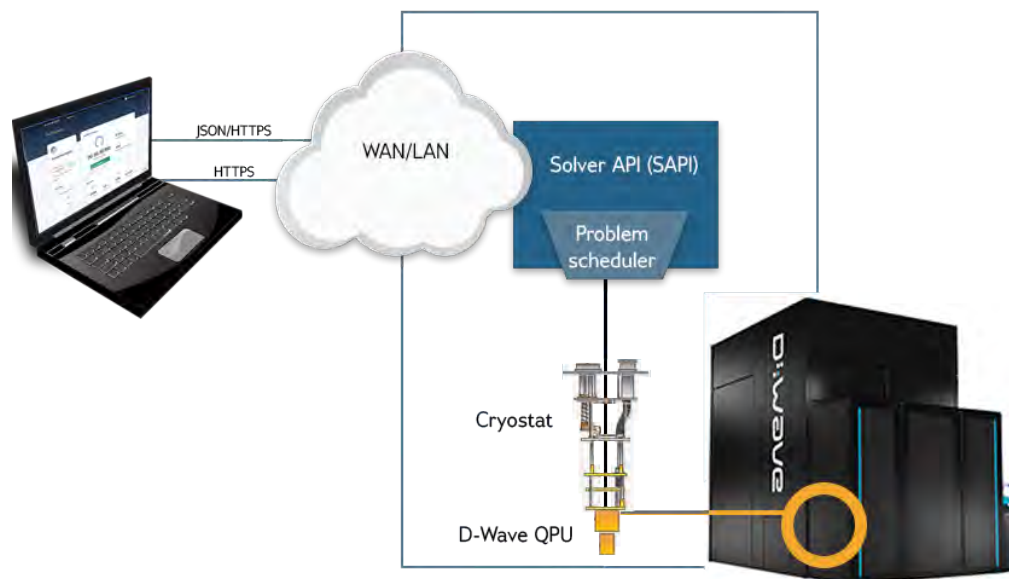
Kuva 11. D-Wave Leapin dashboard-näkymä.

Yllä oleva kuva on esimerkki D-Wave Leapin kirjautumisen jälkeen käytettävissä olevan dashboard- näkymästä, josta löytyy pääsy demonstraatioiden ajamiseen, esimerkkisovelluksiin tutustumiseen, dokumentaatioon, työkalujen käyttöönottoon sekä tietoja liittyen mm. käytettävissä oleviin laitteistoihin, kuukausitason käytettyyn ja jäljellä olevaan laskentakapasiteettiin.

Leap sisältää myös ohjelmistokehitystyökaluja (mm. Ocean SDK), kirjastoja, demoja ja muita esimerkkejä, jotka auttavat käyttäjiä kehittämään ja testaamaan kvanttilaskennan sovelluksia D-Wave:n kvanttitietokoneiden avulla. Lisätietoja näistä löytyy osoitteesta: <https://cloud.dwavesys.com/leap/resources/> (vaatii kirjautumisen palveluun)

D-Wave Leapia hyödyntävät tutkijat, kehittäjät ja yritykset monenlaisiin sovelluksiin, kuten rahoitukseen, logistiikkaan, lääketehtävien ratkaisemiseen ja koneoppimiseen liittyviin optimointiongelmien, joihin annealer-tyyppiset kvanttitietokoneet soveltuvat hyvin.

Leap tarjoaa mahdollisuuden ympäristön ilmaiseen koekäyttöön, kokeilla annealer-tyyppisiä kvanttitietokoneita ja tutkia niiden soveltamista erilaisiin käyttökohteisiin sekä kehittää niille suoritettavia ohjelmia erilaisten tukityökalujen avulla.



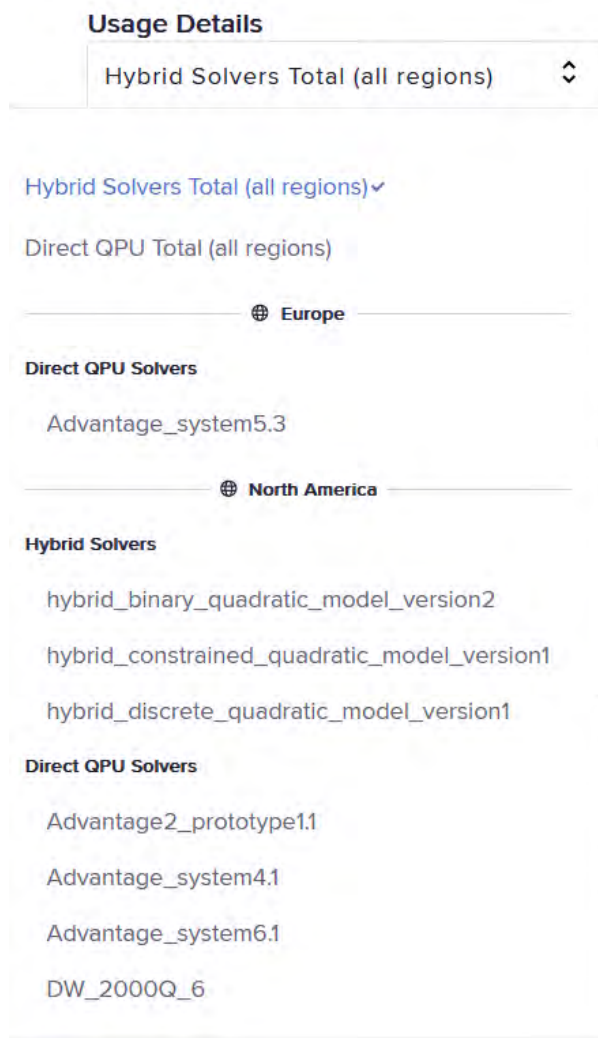
Kuva 12. Leap -palvelun käytön arkkitehtuuri (Kuva: D-Wave)

D-Wave Leap sisältää myös Hybrid Solver Servicen (HSS), joka yhdistää perinteisen tietokoneen ja kvanttitietokoneen laskentaresursseja. Lisätietoja Leapin ja siinä käytettävän Ocean SDK:n hyödyntämisestä hybrid solver-periaatteesta: <https://www.dwavesys.com/resources/white-paper/d-wave-hybrid-solver-service-an-overview/>

Leap –palvelun käyttöönnoton ja käytön tueksi tarkempi dokumentaatio on saatavilla osoitteessa [https://docs.dwavesys.com/docs/latest/guides\\_leap.html](https://docs.dwavesys.com/docs/latest/guides_leap.html)

## D-Wave Leapin hardware- ratkaisut (tuetut kvanttietokonelaitteistot):

Leapin dashboard mahdollistaa pääsyn näkemään, millaisia laitteistoja Leap- ympäristössä on käytettävissä ohjelmien suorittamiseen. Leap –palvelun sisältämät laitteet ovat D-Waven annealer- tyyppisiä QPU –kvanttisuorittimia sekä hybrid solve- reita.



Kuva 13. D-Wave Leap:in dashboard- portaali.

Yllä oleva kuva esittää yhden esimerkin palvelun laitekonfiguraatioista/valinnoista. Lisätietoja tarkemmin laitteista ja niiden ominaisuuksista: [https://docs.dwavesys.com/docs/latest/doc\\_getting\\_started.html](https://docs.dwavesys.com/docs/latest/doc_getting_started.html)

On hyvä huomioida, että D-Waven käyttämä annealer- tekniikka poikkeaa esim. muiden valmistajien käyttämien superconductive – tai trapped ion – tyyppisten tekniikoiden osalta. Tämän vuoksi annealer- koneiden kubittien määrä on usein kerta-luokkia suurempia (100x, 1000x) mitä edellä mainituilla toteutetut kvanttietokoneiden kubittien määrä.

## **Palvelun käytön vaihtoehdot, hinnoittelu ja käyttöönotto**

D-Wave tarjoaa kolmea erityyppistä käyttövaihtoehtoa (subscription, plans) Leap-palvelun käyttämiseksi; Trial Plan, Developer Plan ja Commercial Access and Research & Education Plan.

**Trial Plan** –vaihtoehto tulee käyttöön rekisteröityessä ja kertaa kirjautuessa ensimmäistä Leapin käyttäjäksi. Trial Plan- vaihtoehto sisältää mm. pääsyn Leap- palvelussa oleviin demoihin ja muihin esimerkkisovelluksiin sekä selainpohjaisen IDE- kehitystyökalun käyttöön sekä Ocean -sovelluskehitystyökalujen käyttöönottoon. Lisäksi Leap- palvelu sisältää kattavan dokumentaation sekä Community-kehittäjäyhteisöön.

Lisäksi Trial Plan:issa tarjotaan kuukaudeksi yksi (1) minuutti prosessointiaikaa Leap-pilvipalvelun sisältämien D-Waven annealer- tyyppisten kvanttietokoneiden käyttöön ohjelmien suorittamiseksi. (Minuutti laskenta-aikaa kuukaudessa ei tunnu suurelta, mutta esimerkiksi Leapin sisältämä demo luvun tekijöihin jakamiseksi (factoring) esitti suoritusajaksi 0.025 sekuntia. Minuutissa tällaisia suorituksia voisi ajaa yli 2000 kappaletta). Vaihtoehtoisesti Trial Plan sisältää myös 20 minuutin kuukausittaisen laskenta-ajan palvelun sisältämien hybrid-solverien käyttöön. QPU- ja hybrid solver- laskenta-aikaa voi käyttää rinnakkain.

Trial Plan – periodin päätyttyä käyttäjän on siirryttävä palvelussa **Developer Plan**-vaihtoehtoon, joka tarjoaa mahdollisuuden palvelun käyttämisen jatkamiseksi sekä lisää laskenta-aikaa (1 min QPU -laskenta-aikaa, 20 min hybrid solver-käyttöaikaa).



Tämä vaihtoehto vaatii kuitenkin GitHub- tunnuksien liittämisen Leap- palveluun ja omien ohjelmien tarjoamisen GitHub- versionhallintapalvelun kautta saatavaksi. Kolmas vaihtoehto, Commercial Access and Research & Education Plan:it, tarjoavat palvelun joko kaupalliseen käyttöön tai tutkimus- ja koulutuskäyttöön. Nämä vaihtoehdot tarjoavat QPU- ja hybrid solver- resurssien laajempaan käyttöön. Tämän vaihtoehdon käyttöönotto vaatii yhteydenoton D-Waven myyntitoimintoihin. Tämä on mahdollista tehdä sivustolla <https://cloud.dwavesys.com/leap/plans/>

D-Waven annealer- tyyppisiä koneita tuetaan myös muissa pilvipalveluissa esim. AWS Amazon Braket:issa. Tällaisissa tapauksissa laskentakapasiteetin kustannuksien muodostuminen ja laskutustavat löytyvät kyseisen pilvipalvelun hinnoittelumalleista.

### **Tuettuja SDK- kehitysympäristöjä ohjelmointikieliä, kirjastoja ja sovelluskehityksiä**

D-Wave Leap- palvelun käyttämiseen kvanttiohjelmien toteuttamiseen sekä suorittamiseen simulaattoreilla ja kvanttietokoneilla on tarjolla kehitystyökaluja sekä useampi mahdollinen tapa niiden soveltamiseen.

#### **Ocean SDK**

Ocean SDK ohjelmisto on D-Wave:n tarjoama työkalukokoelma, jota voidaan käyttää rakentamaan Python -kielellä kvanttilaskennan sovelluksia ja suorittamaan niitä D-Waven kvanttietokoneilla, erityisesti yhteistyössä Leap- ympäristön kanssa.

<https://docs.ocean.dwavesys.com/en/stable/index.html>

#### **D-Wave Leap:in selainkäyttöinen IDE- kehitysympäristö**

Selainpohjainen kehitysympäristö sisältää mm. tuen GitHubiin tallennettavien projektien kehittämisen sekä Visual Studio Code- pohjaisen käyttöliittymän ohjelmien toteuttamiseen. IDE- kehitysympäristön osoite on <https://ide.dwavesys.io/workspaces/>

### **3.5.3 Google Quantum AI ja Quantum Computing Service**

Google Quantum AI (tunnetaan myös nimellä Google Quantum Computing) on Googlen kehitysohjelma, joka keskittyy kvanttitekniikan kehittämiseen ja sovellusten luomiseen kvanttitekniikoiden pohjalta.



Kuva 14. Google AI Software (Kuva: Google Quantum AI)

Google Quantum AI tarjoaa sivustonsa kautta monenlaisia työkaluja, resursseja ja alustoja tutkijoille, kehittäjille ja yrityksille, jotka ovat kiinnostuneita kvanttiteknologiasta. Google Quantum AI:n verkkosivusto löytyy osoitteesta <https://quantumai.google/>

Quantum AI –sivustolla tarjottuja resursseja ovat mm:

- **Cirq:** Cirq on Google Quantum AI:n kehittämä avoimen lähdekoodin Python-kirjasto, joka mahdollistaa kvanttipiirien luomisen, simuloinnin ja optimoinnin. Se tarjoaa korkean tason käyttöliittymän kvanttiporttien, piirien ja operaatioiden käsittelyyn.
- **Kvanttitietokoneet:** Google Quantum AI on kehittänyt omia kvanttiprosessoreitaan, kuten Bristlecone ja Sycamore, jotka pohjaavat suprajohdaviin kubitteihin ja niiden avulla toteutettuihin QPU-kvanttiprosessoreihin.
- **Google Quantum Computing Service** antaa käyttäjille mahdollisuuden etäkäyttää Googlen kvanttitietokoneita.
- **Kvanttisovellukset:** Google Quantum AI tutkii aktiivisesti kvanttiteknologian mahdollisia sovelluksia alueilla kuten optimointi, koneoppiminen, kryptografia ja

materiaalitiede, ja tekee yhteistyötä yliopistojen, yritysten ja hallitusten kanssa edistääkseen käytännön edistysaskeleita näillä aloilla.

## **Google Quantum AI:n hardware- ratkaisut (tuetut kvanttietokonelaitteistot):**

Google Quantum AI -kehittämishjelmassa on kehitetty useita kvanttietokoneita.

Merkittävimpiä niistä ovat seuraavat:

- **Bristlecone:** Google Quantum AI:n kehittämä 72-kubitin kvanttiprosessori.
- **Sycamore:** toinen Google Quantum AI:n kehittämä 54- kubittinen kvanttiprosessori, jota käytettiin vuoden 2019 "kvanttiherruus"- kokeilussa. Sycamore on suunniteltu suorittamaan kvanttilaskentaa korkealla tarkkuudella (*fidelity*) ja pienillä virhetasoilla (*error rate*).

Muut prosessorit:

- 22- kubittinen Foxtail
- Bristlecone V2: päivitetty versio alkuperäisestä Bristleconesta, jossa on parannettu kubittien yhteysrakenne (*qubit connectivity*).

Lisätietoja Googlen kvanttietokoneista: <https://quantumai.google/hardware>

## **Palvelun käytön vaihtoehdot, hinnoittelu ja käyttöönotto**

Hintojen ja saatavuuden osalta Google Quantum AI tarjoaa ilmaisen pääsyn Cirq-kirjastoon, avoimen lähdekoodin työkaluihin ja resursseihin kvanttilaskennan tutkimus- ja kehitystyöhön.

Google Quantum AI:n käyttöönotto kokeiluun ja testaukseen on tehtävissä esim. seuraavien toimenpiteiden kautta:

1. Siirry viralliselle Google Quantum AI -verkkosivustolle osoitteessa <https://quantumai.google/>

- Luo tarvittaessa tunnus ja kirjaudu sivuston Sign In- toiminnolla.

2. Tutustu saatavilla oleviin resursseihin, dokumentaatioon ja työkaluihin, kuten Cirq, joka on Googlen kehittämä avoimen lähdekoodin kvanttilaskennan kirjasto tai TensorFlow Quantum, joka on kvanttilaskentaan laajennettu versio koneoppimiseen sovellettavasta TensorFlow -ympäristöstä.

- Educational Resources - itseopiskelumateriaalia: <https://quantumai.google/education>
- Yleistä tietoa kehitystyökaluista: <https://quantumai.google/software>
- Tietoa Cirq- kirjastosta: <https://quantumai.google/cirq>
- Tietoa TensorFlow Quantum -sovelluskehuksesta: <https://www.tensorflow.org/quantum>

3. Tutustu tarjottuihin simulaattoreihin ja kvanttilaitteistoihin ohjelmien suorittamiseksi:

- Laitteet: <https://quantumai.google/cirq/hardware/devices>
- Simulaattorit: <https://quantumai.google/cirq/simulate/simulation>

4. Perehdy kvanttilaskennan käsitteisiin ja periaatteisiin Google Quantum AI:n tarjoaman dokumentaation ja opiskelumateriaalien avulla.

- Cirq'in käyttöönotto: <https://quantumai.google/cirq/start/install>
- Kvanttipiirien suunnittelu Cirq'in avulla: <https://quantumai.google/cirq/build>
- Kvanttipiirien simulointi: <https://quantumai.google/cirq/simulate>
- Ohjelmien suorittaminen Cirq- kirjaston ja Google Quantum Computing Services -palvelun kautta, suoritusten hallinta (jobs) ja tuloksien (results) tarkastelu: <https://quantumai.google/cirq/google/concepts> (Varsinaisilla laitteilla suorittaminen on sallittu vain erikseen tehdyn sopimuksen mukaan.)

### **Tuettuja SDK- kehitysympäristöjä ohjelmointikieliä, kirjastoja ja sovelluskehysä**

Googlen kvanttilaskentapalveluissa hyödynnettävistä ohjelmistoista yleistä tietoa löytyy sivustosta: <https://quantumai.google/software>

Googlen ratkaisuihin yksi keskeinen tekniikka on Python- kieltä tukeva Cirq, avoimen lähdekoodin sovelluskehys. Cirq mahdollistaa kvanttipiirien kirjoittamisen, hallinnan ja optimoinnin sekä niiden suorittamisen sekä simulaattoreilla että kvanttitietokoneilla.

Cirq on pyritty kehittämään korkean tason kieleksi, jolla kvanttiohjelmien kirjoittaminen olisi mahdollista jo tämän päivän NISQ- aikakauden koneille. Lisätietoja Cirq:in käyttöönotosta, käytöstä sekä muu tarkempi dokumentaatio löytyy osoitteesta <https://quantumai.google/cirq>

Toinen mielenkiintoinen Googlen kvanttilaskentapalveluissa hyödyntämä tekniikka on koneoppimiseen tarkoitettun TensorFlow –sovelluskehityksen ”kvanttilaajennus”, **Tensorflow Quantum**. TensorFlow Quantum tukee Python-kieltä ja mahdollistaa kvanttikoneoppimista sekä quantum-classical – hybridityyppisten koneoppimisratkaisujen kirjoittamisen sekä suorittamisen sekä simulaattoreilla että kvanttietokoneilla. Lisätietoja TensorFlow Quantum:in käyttöönotosta ja käytöstä sekä tutoriaaleja ja muita esimerkkejä löytyy osoitteesta <https://www.tensorflow.org/quantum>

Kvanttiohjelmointia Googlen tukemia ohjelmointityökaluja sekä kvanttietokoneita käyttäen voi tehdä esimerkiksi paikallisesti asennetuissa kehitysympäristöissä ja niihin käyttöönotettuja kvanttilaskentakirjastoja (Cirq, TensorFlow Quantum) hyödyntäen. Toinen vaihtoehto on käyttää esimerkiksi Googlen Colabs- verkkopalvelua, johon kvanttiohjelmointilaajennukset ovat samalla tavoin käyttöönotettavissa.

Ohjelmissa voidaan määritellä sovelluksille haluttu ”runtime”- ympäristö, joka yleensä oletuksena on simulaattoripohjainen ympäristö, mutta voidaan konfiguroida myös käyttämään aitoa kvanttietokonetta. Cirq ja TensorFlow Quantum tukevat sekä Googlen mutta myös muiden valmistajien kvanttietokoneita ja palveluja (Azure Quantum, IonQ, Pasqal, Rigetti).

Lisätietoja esimerkiksi Cirq:in tukemista laitteistoista sekä niiden käyttöön tarvittavista asetuksista löytyy osoitteesta <https://quantumai.google/cirq/hardware>

### **Lähteitä ja lisätietoja:**

Google Quantum AI. <https://quantumai.google/>

## **3.5.4 IBM Quantum**

IBM Quantum on IBM:n tarjoama pilvipalvelu kvanttietokoneiden etäkäyttämiseksi verkon kautta. Palvelun käyttäjät voivat käyttää IBM:n simulaattoreita tai aitoja kvanttietokoneita, kokeilla kvanttipiirejä ja suorittaa kvanttialgoritmeja IBM Quantum käyttöliittymän avulla.

IBM Quantum löytyy osoitteesta: <https://quantum-computing.ibm.com/>

Kvanttilaitteiston etäkäytön lisäksi IBM Quantum tarjoaa myös valikoiman erilaisia ohjelmistoja ja ohjelmointikieliä. Yksi keskeisin näistä on Qiskit, avoimen lähdekoodin kvanttilaskennan ohjelmistokehityspaketti (SDK), jonka avulla kehittäjät voivat kirjoittaa ohjelmia kvanttietokoneille Python- ohjelmointikielellä. Qiskit sisältää työkaluja kvanttipiirien (quantum circuits) suunnitteluun, simulointiin ja suorittamiseen IBM:n kvanttietokoneilla.

### **IBM Quantum:in hardware- ratkaisut (tuetut kvanttietokonelaitteistot):**

IBM Quantum perustuu IBM:n omiin, superconducting –teknologialla valmistettuihin kvanttietokoneisiin sekä simulaattoreihin ohjelmistojen kehittämiseksi ja testaamiseksi.

Simulaattorien lisäksi laitetason kvanttietokoneista ovat saatavilla seuraavat:

- 5- ja 7- kubittiset kvanttietokoneet
- 27-kubittinen IBM Falcon,
- 127-kubittinen Eagle
- myöhemmin myös 433- kubittinen Osprey

IBM:n kvanttietokoneista ja niiden sisältämistä kvanttiprosessoreista lisätietoa löytyy kattavammin osoitteesta: <https://quantum-computing.ibm.com/admin/docs/admin/manage/systems/processors>

### **Palvelun käytön vaihtoehdot, hinnoittelu ja käyttöönotto**

Tällä hetkellä IBM Quantum –palvelussa tarjottuja käyttö- ja hinnoittelumalleja ovat seuraavat:

#### **Open Plan**

- Ilmainen, opetus-, opiskelu- ja kehityskäyttöön tarkoitettu malli.
- Käytössä ovat sekä simulaattorit että 5- ja 7- kubittiset kvanttietokoneet.

#### **Pay-As-You-Go Plan**

- Käytön mukaan hinnoiteltu malli, esim. isompien työkuormien suorittamiseen ja tehokkaampien kvanttisysteemien käyttämiseen.

- Käytössä ovat simulaattorit sekä 27- kubittinen Falcon järjestelmä.

## Premium Plan

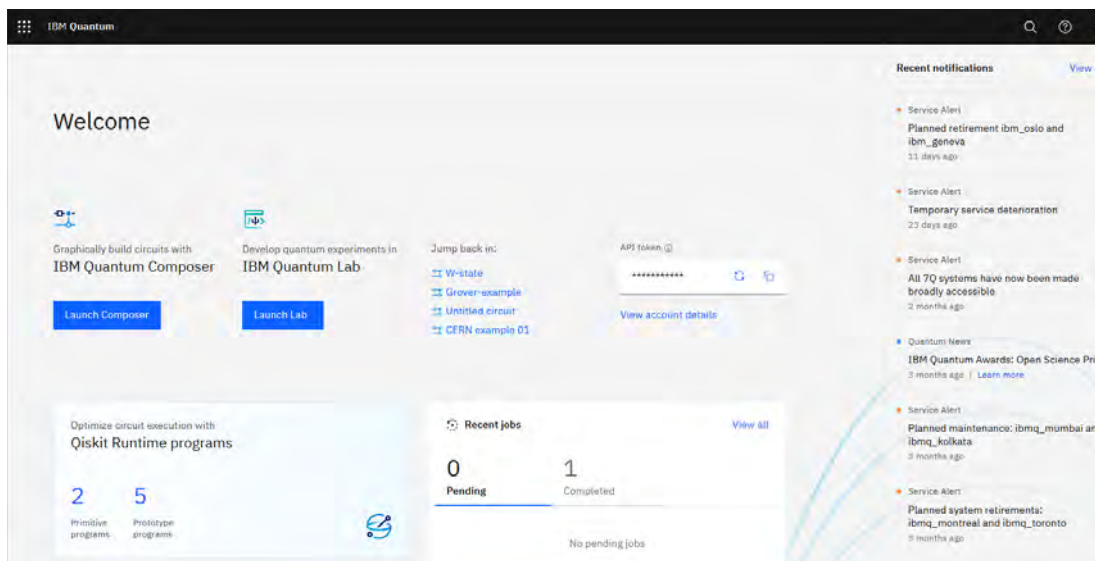
- Erikseen hinnoiteltu, teollisen käytön mittakaavaan tarkoitettu malli
- Käytössä 27-kubittinen Falcon, 127-kubittinen Eagle (myöhemmin, jollakin aikavälillä, oletettavasti vuonna 2023, myös 433- kubittinen Osprey)

Lisätietoja hinnoittelusta, ehdoista yms. tarkemmin osoitteessa:

<https://www.ibm.com/quantum/access-plans>

Kuten yllä hinnoittelun yhteydessä esitelty, tarjoaa IBM Quantum ilmaisen Open Plan-mallin, jonka kautta mahdollistuu pääsy palvelun testaamiseen. Tämän voi tehdä rekisteröitymällä IBM Quantum- palveluun osoitteessa: <https://quantum-computing.ibm.com/>

Palveluun kirjautuminen vaatii IBM ID- tunnuksen, jonka voi luoda ensimmäisen kirjautumisen yhteydessä. Tämän jälkeen palvelun portaali avautuu tarjoten pääsyn erilaisiin selainpohjaisiin sovelluskehitystyökaluihin (esim. Composer- ohjelmointityökalu kvanttipiirien rakentamiseksi ja Quantum Lab –palveluun Python- kielisten kvanttisovellusten toteuttamiseen ja testaamiseen Jupyter Notebook- tyyppisillä kehitystyökaluilla)



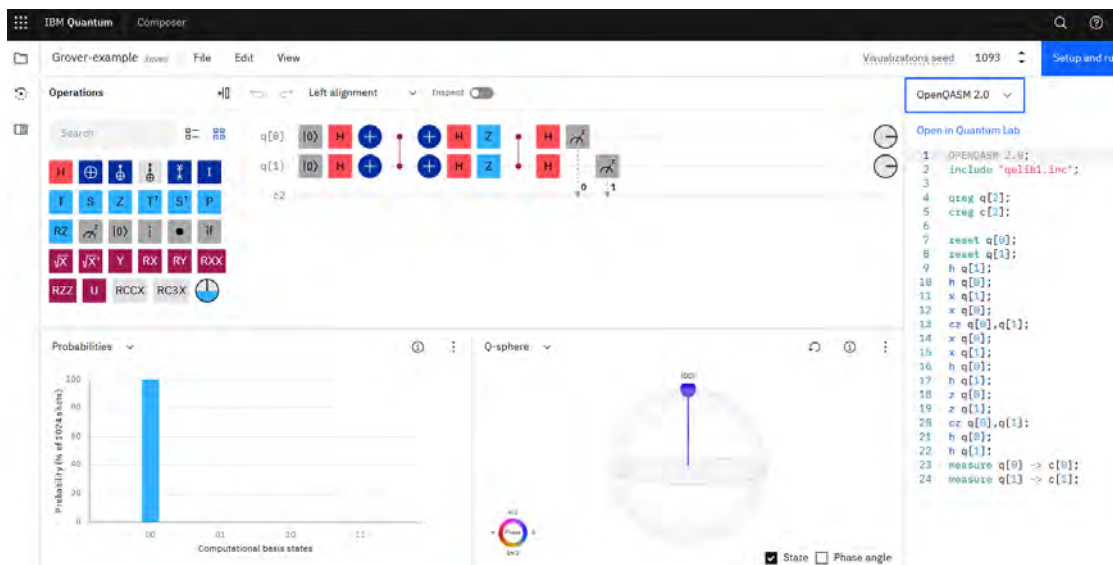
Kuva 15. IBM Quantum –käyttöliittymä.

Käyttöliittymä tarjoaa myös käyttäjäkohtaisen API token- avaimen, jolla esimerkiksi paikallisilla kehitysympäristöillä rakennetut sovellukset voidaan kytkeä IBM Quantum- palvelun sisältämiin kvanttietokoneisiin ohjelmien suorittamiseksi.

## Tuettuja SDK- kehitysympäristöjä ohjelmointikieliä, kirjastoja ja sovelluskehityksiä

IBM Quantum tukee OpenQASM -kieltä ja Qiskit- kirjastoa kvanttipiirien suunnittelemiseksi sekä sovellusten kehittämiseksi.

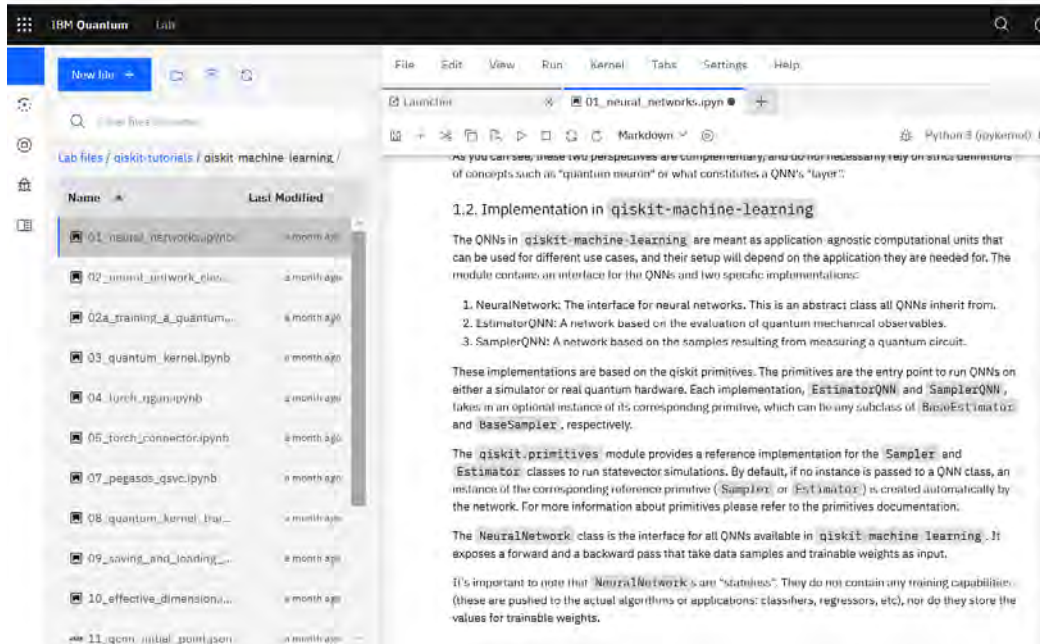
Composer-sovellus tarjoaa visuaalisen kvanttipiirien käyttöliittymän, jossa rakennetut piirit muunnetaan joko OpenQASM- tai Qiskit- kirjastoa hyödyntävään Python- kieliseen muotoon (ks. alla oleva kuva 16).



Kuva 16. Composer-sovelluksen käyttöliittymä.

IBM Quantum Lab tukee Qiskit-kirjaston hyödyntämistä JupyterNotebook-kehitysympäristön tai konsolikäyttöliittymän avulla. Etenkin Notebookien käyttö tukee Qiskitin eri ominaisuuksien kokeilua, toteuttamista, testaamista ja suorittamista palvelun tarjoaman interaktiivisen käyttöliittymän kautta (kuva 17). Palvelu tarjoaa myös runsaan määrän esimerkkisovelluksia, joihin tutustuminen antaa kuvan Qiskit-kirjaston käytämisestä kvanttilaskennan eri sovelluskohteisiin.





Kuva 17. Interaktiivinen käyttöliittymä.

Qiskit Textbook- sivustolta on saatavilla runsaasti ilmaisia itseopiskelumateriaalia <https://qiskit.org/learn/> , alkaen kvanttilaskennan ja - ohjelmoinnin perusteista aina kvanttikoneoppimiseen saakka.

- Introduction to Quantum Computing: <https://qiskit.org/learn/course/introduction-course/>
- Quantum machine learning: <https://qiskit.org/learn/course/machine-learning-course/>

Lisätietoja: <https://www.ibm.com/quantum>

### 3.5.5 Microsoft Azure Quantum

Azure on yhdysvaltalaisen Microsoftin pilvipalvelu erilaisten tietoteknisten resurssien tarjoamiseen verkon kautta käytettäväksi. Azure sisältää yli 200 erilaista palvelua, teknologiaa tai muuta ratkaisua, jotka ovat käytettävissä Azuren –pilvipalvelun kautta, esim.

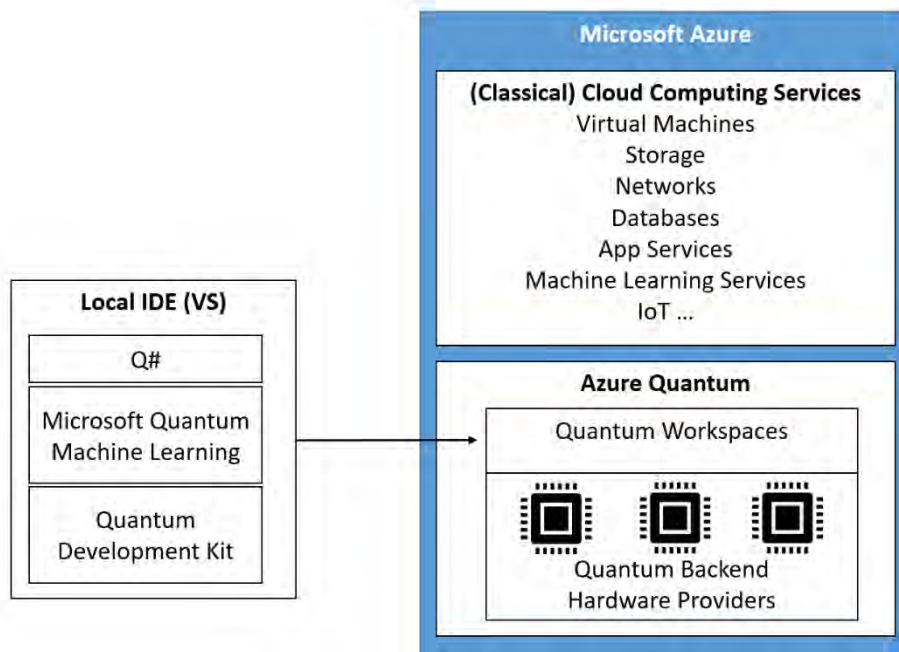
- Virtuaalikoneet
- Pilvitallennustila
- Virtuaaliverkot
- Azure Active Directory- käyttäjähallinta
- Pilvitietokannat ja -tietovarastot
- Dataintegraatio ja muut datan hallintapalvelut
- Tekoäly- ja koneoppimispalvelut
- Analytiikka-ratkaisut
- Container- ympäristöt (esim. Docker, Kubernetes)
- Serverless- teknologiat

Kattava listaus Azure- pilvipalvelun ominaisuuksista löytyy osoitteesta:

<https://azure.microsoft.com/en-us/products/>

Azuren palvelut ovat valtaosin ns. perinteisellä laskentakapasiteetilla (CPU, GPU) toteutettuja, eivätkä suoraan sisällä kvanttilaskentakyvykkyyksiä.

Vuonna 2021 Microsoft kuitenkin julkaisi Azureen kvanttilaskentaa tukevan osion nimeltä **Azure Quantum**. Azure Quantum tarjoaa joustavan mahdollisuuden toteuttaa saman pilvipalvelun sisällä kvanttilaskentaa hyödyntäviä ohjelmia usealle erityyppiselle kvanttietokoneelle. Azure Quantum tukee useita eri ohjelmointikieliä sekä niillä hyödynnettäviä kirjastoja.



Kuva 18. Azure Quantumin toimintaperiaate.

Perustietoja Azure Quantumista löytyy sivustolta: <https://azure.microsoft.com/en-us/services/quantum/>

## **Microsoft Azure:n hardware- ratkaisut (tuetut kvanttietokonelaitteistot):**

Azure Quantum tarjoaa useita eri valmistajien kvanttietokoneita käytettäväksi Azure workspaces- työtilojen kautta:

- Quantinuum (trapped ion)
- IonQ (trapped ion)
- Quantum Circuits Inc. (superconducting)
- Rigetti (superconducting)
- Pasqal (neutral atom)
- IQBit (optimointi, simulaattori)
- Microsoft QIO (optimointi, simulaattori)
- Toshiba SQBM+ (GPU- piireillä toteutettu simulaattori)

(valmistajan nimen jälkeen sulussa koneessa käytettävä teknologia)

## **Palvelun käytön vaihtoehdot, hinnoittelu ja käyttöönotto**

Koska Azure Quantum tarjoaa useampia eri simulaattorivaihtoehtoja sekä eri valmistajien kvanttietokoneita (esim. IonQ, Rigetti, Quantinuum), myös käytön kustannusten muodostuminen tapahtuu hieman eri tavoin riippuen valituista suoritusympäristöistä.

Osa kvanttietokoneista ovat käytettävissä Azuren Credits- mallin mukaisesti sekä käytön mukaan hinnoiteltuna (Pay-as-you-go). Osassa laitteistoissa on eritasoisia valmistajakohtaisia hinnoittelumalleja)

Tarkka tieto hinnoittelusta on löydettävissä sivustolla <https://learn.microsoft.com/en-us/azure/quantum/pricing>

Azure tarjoaa myös erillisen sivuston Azure Quantum- palvelun hinnoista: <https://azure.microsoft.com/en-us/pricing/details/azure-quantum/> Sivustolla on hyvä huomioida, että esimerkiksi kaikki kvanttietokoneet eivät ole saatavilla Azuren kaikilla region- alueilla, vaan voivat olla rajoitettu vain pohjoisamerikkalaisiin alueisiin.

Microsoft tarjoaa Azure Quantum- palveluiden ilmaista testaamisen mahdollisuutta rajoitetun ajan. Tähän löytyy ohjeistus sivustolta <https://azure.microsoft.com/en-us/products/quantum/#get-started>

Sivustolla tarjotaan optioita ilmaisten krediittien (Azure Quantum Credits) hakemiseen eri kvanttietokoneiden käyttämiseksi (krediittejä käytetään laskenta-ajan hankkimiseen). Lisätietoja Azure Quantum Creditseistä: <https://learn.microsoft.com/en-us/azure/quantum/azure-quantum-credits>

### **Tuettuja SDK- kehitysympäristöjä ohjelmointikieliä, kirjastoja ja sovelluskehityksiä**

Azure Quantum sisältää valmiuksia erilaistan kvanttilaskentaa tukevien ohjelmistoratkaisuiden tekemiseen. Tuettuja kehitysyökaluja ja ohjelmointikieliä ovat Cirq, Qiskit ja Microsoftin kehittämä Q#.

#### **Lisätietoja:**

Microsoft Azure Quantum: <https://azure.microsoft.com/en-us/services/quantum/>

## **3.5.6 Oxford Quantum Circuits**

Oxford Quantum Circuits (OQC) on englantilainen kvanttietokonevalmistaja, joka tarjoaa useampaa erityyppistä tapaa kvanttietokoneidensa käyttämiseen pilvipalveluna:

- OQC:n yksityinen pilvipalvelu (private QCaaS platform)
- OQC:n kvanttietokoneiden käyttö Amazon Braket- pilvipalvelun kautta.
- Cyxtera:n datakeskuksiin sijoitetut OQC:n kvanttietokoneet.
- Equinixin datakeskuksiin sijoitetut OQC:n kvanttietokoneet.

<https://oxfordquantumcircuits.com/products>

### **Oxford Quantum Circuits:in hardware- ratkaisut (tuetut kvanttietokone-laitteistot):**

OQC:n pilvipalvelut hyödyntävät OQC:n suprajohtaviin kubitteihin pohjaavia kvanttietokoneita. Tällä hetkellä (2022) saatavilla on 8-kubittinen Lucy- nimellä kutsuttu laite.

Lisätietoja OQC:n laitteistoista: <https://oxfordquantumcircuits.com/technology>

### **Palvelun käytön vaihtoehdot, hinnoittelu ja käyttöönotto**

OQC:n yksityisen pilvipalvelun hinnoittelu määrittyy tapauskohtaisesti ja on tiedusteltava yritykseltä erikseen.

Amazon Braketin kautta käytettäessä hinnoittelu pohjaa Braketin käyttämiin hinnoittelun periaatteisiin. Lisätietoja on saatavilla osoitteessa <https://aws.amazon.com/braket/quantum-computers/oqc/>

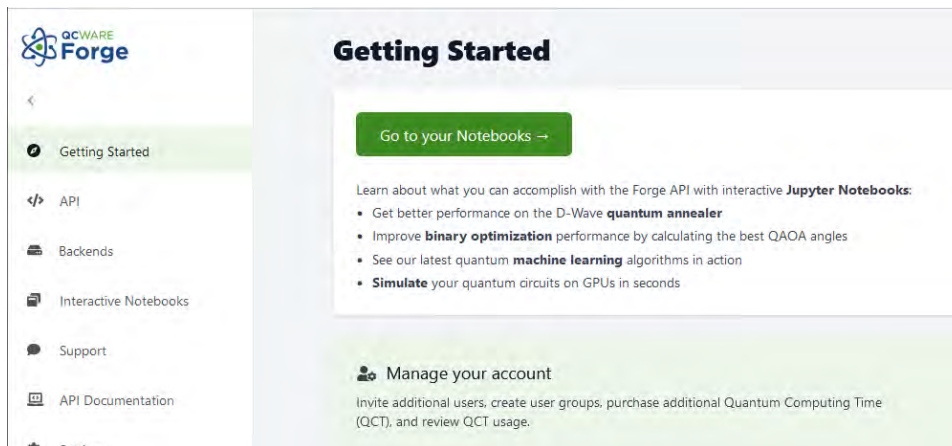
Cyxteran ja Equinixin palveluiden kautta käytettävät hinnoittelut eivät olleet hankkeen aikana saatavilla.

**Lisätietoja:** <https://oxfordquantumcircuits.com/>

### **3.5.7 QC Ware Forge**

QC Ware Forge on pilvipohjainen kvanttilaskenta-alusta. Palvelun kautta on saatavilla pääsy kvanttietokoneisiin, ladattavissa ja käytettävissä erilaisia kehitystyökaluja kvanttiohjelmointiin sekä tarjolla asiantuntijatukea. Forge- alusta on kehitetty ranskalaisen QC Ware -nimisen yrityksen toimesta. QC Ware keskittyy kvanttilaskennan ja kvanttiohjelmistojen kehittämiseen eri toimialojen tarpeisiin.

QC Ware Forge- palvelun hallintaportaali löytyy osoitteesta <https://forge.qcware.com/>



Kuva 19. QC Ware Forgen hallintakäyttöliittymä

QC Ware Forge mahdollistaa käyttäjille kvanttialgoritmien suunnittelun, kehityksen, testauksen ja käyttöönoton erilaisten ohjelmistokehitystyökalujen ja ohjelmointikielten avulla. Alusta tarjoaa myös pääsyn eri valmistajien kvanttilaitteisiin ja tarjoaa optimointityökaluja kvanttilaskentaa suorittavien algoritmien suorituskyvyn parantamiseksi. QC Ware Forge on suunnattu erityisesti tutkijoille ja yrityksille, jotka ovat kiinnostuneita kvanttilaskennan potentiaalin hyödyntämisestä monimutkaisten ongelmien ratkaisemiseen eri aloilla.

### **QC Ware Forge:n hardware- ratkaisut (tuetut kvanttietokonelaitteistot):**

QC Ware Forge- palvelussa tuetaan useita erilaisia kvanttilaitteistoja ja –simulaattoreita (IonQ, Rigetti, NVIDIA) sekä muiden palveluntarjoajien pilvipalveluita (AWS Bracket, IBM Quantum).

## Backends

Select a backend for a more detailed view.

Supported API Method	Provider
Any	Any
awsbraket/ionq Gate-model quantum computer	IONQ
awsbraket/rigetti_aspen_11 Gate-model quantum computer	rigetti
awsbraket/rigetti_aspen_m_1 Gate-model quantum computer	rigetti
awsbraket/sv1 Classical gate-model simulator	amazon Braket
awsbraket/tn1 Gate-model simulator	amazon Braket
ibmq Multi-backend	IBM Quantum Network
qcware/cpu Classical processor	QCWARE
qcware/cpu_simulator Classical gate-model simulator	QCWARE
qcware/gpu GPU-based processor	NVIDIA
qcware/gpu_simulator GPU-based gate-model simulator	NVIDIA

Kuva 20. QC Ware Forge Backends- hallintaportaali,

### Palvelun käytön vaihtoehdot, hinnoittelu ja käyttöönnotto

QC Ware Forge- palvelun hinnoittelussa ja resurssien käytön mittaroinnissa käytetään QCT - eli Quantum Computing Time -krediittejä. Kyseessä on ikäänkuin yksikkö, joka kuuluu käytettäessä palvelua tietyin periaattein. Nämä muodostuvat palveluiden käyttöön välitettyjen API- kutsujen määrän sekä käytetyn ajan ja etenkin laskentaan käytetyn backend:in eli tyypillisesti kvanttietokoneen mukaan. Eri kvanttietokoneet (IonQ, Rigetti) hinnoitellaan esimerkiksi tunti hinnoittelun mukaisesti.

Nämä periaatteet löytyvät yksityiskohtaisesti esiteltynä sivustolla <https://forge.qcware.com/pricing/>

QC Ware Forge tarjoaa useita hinnoitteluvaihtoehtoja (plans), esimerkiksi käyttötarpeiden mukaan mitoittamisen helpottamiseksi ja kustannusten ennakointiin ja muuhun hallintaan.

## Developer Plan

Start for free with our **Developer Plan**, then unlock additional features and **QCT** with one of our enterprise plans.

✓ 1 minute QCT   ✓ 1 user seat   ✓ Algorithm libraries   ✓ Unlimited CPU simulator time

## Enterprise Plans

Pay-As-You-Go	Small	Medium	Large
<p>All the basics to start out with Quantum Computing</p> <p><b>\$2,500</b> /hr</p> <p><a href="#">Buy Pay-As-You-Go</a></p>	<p>Perfect for a small team of researchers</p> <p><b>\$25,000</b></p> <p><a href="#">Buy Small</a></p>	<p>For medium-sized businesses and larger R&amp;D teams</p> <p><b>\$100,000</b></p> <p><a href="#">Buy Medium</a></p>	<p>For large enterprises and corporate/government R&amp;D labs</p> <p><b>\$500,000</b></p> <p><a href="#">Buy Large</a></p>
<p><b>WHAT'S INCLUDED</b></p> <ul style="list-style-type: none"> <li>✓ 1 year term</li> <li>✓ 1 hour QCT</li> <li>✓ Access to all QPUs and simulators</li> <li>✓ Algorithm libraries</li> <li>✓ 1 user seat</li> </ul>	<p><b>WHAT'S INCLUDED</b></p> <ul style="list-style-type: none"> <li>✓ 1 year term</li> <li>✓ 11 hours QCT (10% discount)</li> <li>✓ Access to all QPUs and simulators</li> <li>✓ Algorithm libraries</li> <li>✓ 5 user seats</li> <li>✓ Team features</li> </ul>	<p><b>WHAT'S INCLUDED</b></p> <ul style="list-style-type: none"> <li>✓ 2 year term</li> <li>✓ 50 hours QCT (20% discount)</li> <li>✓ Access to all QPUs and simulators</li> <li>✓ Algorithm libraries</li> <li>✓ 25 user seats</li> <li>✓ Team features</li> </ul>	<p><b>WHAT'S INCLUDED</b></p> <ul style="list-style-type: none"> <li>✓ 3 year term</li> <li>✓ 285 hours QCT (30% discount)</li> <li>✓ Access to all QPUs and simulators</li> <li>✓ Algorithm libraries</li> <li>✓ Unlimited user seats</li> <li>✓ Team features</li> </ul>

Kuva 21. QC Ware Forgen eri hinnoittelumallit.

Alla on yhteenveto tarjotuista (kartoituksen aikaisista) hinnoittelumalleista:

- **Developer plan** on QC Ware Forgen tarjoama ilmainen kokeiluversio, joka antaa käyttäjille mahdollisuuden tutustua alustaan ja sen ominaisuuksiin. Kokeiluversiossa on rajoitettu käyttöaika ja ominaisuudet.
- **Pay-As-You-Go-plan** sisältää vuoden mittaisen sopimuksen ja tarjoaa perustason resurssit kvanttilaskennan hyödyntämiseen QC Ware Forge palvelun kautta. Tämä malli tarjoaa tuntipohjaisen hinnoittelun QCT- laskenta-ajalle, ja kustannukset muodostuvat käytetyn ajan mukaisesti. Tämä malli tarjoaa pääsyn kaikkiin palvelun kautta saatavilla oleviin kvanttietokoneisiin ja simulaattoreihin. Malli on rajoitettu vain yhdelle käyttäjälle.
- **Small plan**:in kerrotaan soveltuvan esimerkiksi pienelle tutkijaryhmälle (max. 5 käyttäjää). Malli tarjoaa kiinteän kustannuksen vuoden mittaiselle sopimusjaksolle sisältäen 11 tuntia QCT- laskenta-aikaa. Malli tarjoaa pääsyn palvelun kaikkiin kvanttietokoneisiin ja simulaattoreihin.
- **Medium plan** -mallia tarjotaan keskisuurille yrityksille tai laajemmille tutkimusryhmille (max. 25 käyttäjää). Sopimuskausi on 2 vuotta, sisältäen 50 tuntia QCT- laskenta-aikaa, jota voidaan edellisten mallien tapaan hyödyntää palvelun sisältämien kvanttietokoneiden ja simulaattorien käyttöön.



- **Large plan** -taso on malleista laajin ja myös kallein. Large-taso on suunnattu suurille yrityksille tai tutkimuskeskuksille. Sopimuskausi on 3 vuoden mittainen, sisältäen huomattavasti laajemman QCT- kapasiteetin (lähes 300 tuntia), jota voidaan hyödyntää kvanttietekoneiden ja simulaattorien käyttöön. Tämä malli ei rajoita palvelua hyödyntävien käyttäjien määrää.
- Palveluun on saatavilla myös paikallisia (on-premise) ratkaisuja, joiden hinnoittelu ja sopimusehdot määräytyvät tapauskohtaisesti.

Ajantasainen ja tarkempi hinnoittelutieto löytyy QC Waren sivustolta

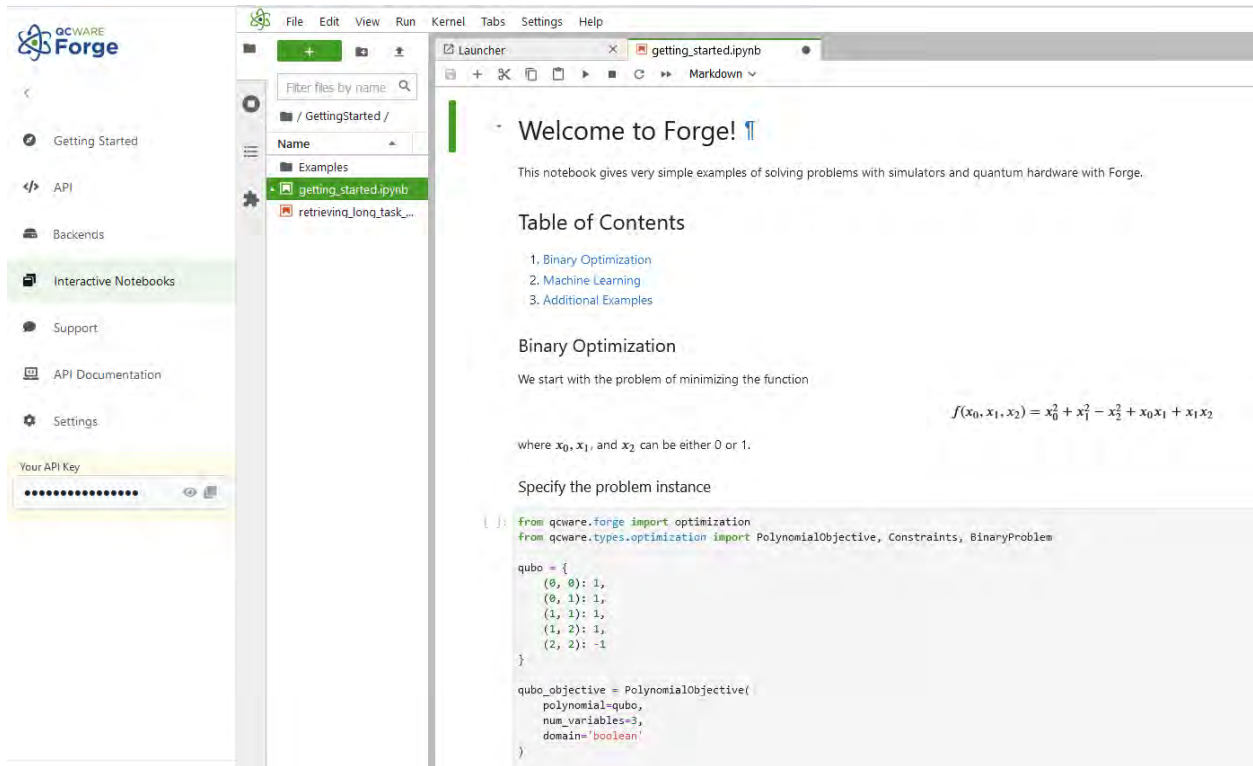
<https://forge.qcware.com/pricing/>

Kuten yllä esitelty, QC Ware Forge- palvelussa on "Start for free" -optio Developer Plan- mallin mukaiseen käyttöön.

Osoitteessa [https://app.forge.qcware.com/sign\\_up/](https://app.forge.qcware.com/sign_up/) pääsee luomaan ilmaisen, Developer plan- tyyppisen tunnuksen, jolla palvelun testaaminen on mahdollista tietyin rajoituksin (mm. rajoitettu QCT- laskenta-aika).

## **Tuettuja SDK- kehitysympäristöjä ohjelmointikieliä, kirjastoja ja sovelluskehityksiä**

QC Ware Forge tarjoaa selainkäyttöisen Interactive Notebooks- kehitysympäristön, jossa sovelluksia kehitetään Jupyter Notebook- ympäristöllä Python- kielellä ohjelmoiden. Ympäristö sisältää esimerkkisovelluksia, joiden tarkoitus on esitellä ja helpottaa palvelun käyttöönottoa ja käyttöä.



Kuva 22. QC Ware Forgen käyttöliittymä.

Palvelu tarjoaa myös Forge client- kirjaston, jolla voi kehittää Python -kielellä sovelluksia paikallisessa kehitysympäristössä. Ympäristö tukee mm. ohjelmistollista kvanttipiirien rakentamista ja suorittamista, optimointiratkaisujen ja kvanttikoneoppimisen ratkaisujen kehittämistä.

**Lähteitä ja lisätietoja:** <https://forge.qcware.com/>

### 3.5.8 Xanadu Cloud

Xanadu Cloud on kanadalaisen Xanadu-nimisen kvanttiteknologiayrityksen tarjoama pilvipohjainen kvanttietokonealusta. Se tarjoaa joukon työkaluja, resursseja ja palveluita, joiden avulla voidaan käyttää, kokeilla ja kehittää kvanttietokoneiden sovelluksia Xanadun ftoniikka-pohjaisilla tekniikoilla toteutetuilla kvanttiprosessoreilla ja niihin liittyvällä ohjelmistolla.

Xanadu Cloud on käytettävissä sivustolta <https://cloud.xanadu.ai/>

Sivustolla pääsee rekisteröitymään uudeksi käyttäjäksi, kirjautumaan sisälle varsinaiseen palveluun ja käyttämään palvelun tarjoamia resursseja (laitteistoit, ohjelmistot, dokumentaatio).

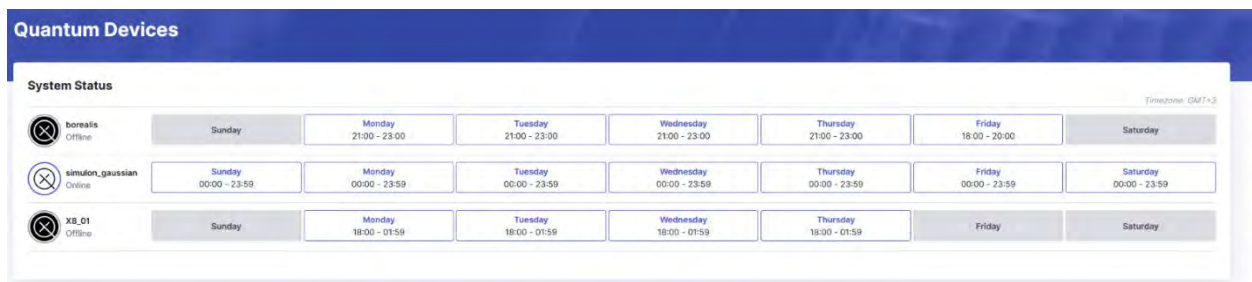
Pilvipohjaisena alustana Xanadu Cloud mahdollistaa kvanttilaskelmien suorittamisen etänä ilman tarvetta omaan kvanttilaitteistoon. Se tarjoaa skaalautuvuutta, joustavuutta ja helppokäyttöisyyttä, mikä tekee siitä helposti saavutettavan erilaisille käyttäjille, aloittelijoista kokeneisiin kvanttietokoneiden käyttäjiin.

Xanadu Cloud tarjoaa käyttäjille käyttöliittymän, jonka avulla voi kirjoittaa kvanttipiirejä, simuloida kvanttilaskelmia ja suorittaa kvanttiohjelmia Xanadun laitteistolla tai virtuaalisilla kvanttiprosessoreilla. Alusta tukee suosittuja ohjelmointikieliä, kuten Pythonia, ja tarjoaa valikoiman kvanttialgoritmeja, kirjastoja ja kehitysympäristöjä (SDK) kvanttitehtävien toteuttamiseen, kuten kvanttilaskentaan, kvanttioptimointiin ja kvanttsimulointiin.

### **Xanadu Cloud:in hardware- ratkaisut (tuetut kvanttietokonelaitteistot):**

Xanadu Cloud tarjoaa Xanadun valmistamien Borealis- ja X-Series- tyyppisiä laitteita.

Palvelun portaalin Quantum Devices -osio näyttää tietoa saatavilla olevista laitteista ja niiden tilasta ja saatavuudesta.



The screenshot shows a 'System Status' table for Quantum Devices. The table has columns for days of the week (Sunday to Saturday) and rows for different device types. The 'Timezone' is GMT+3.

	Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
<b>borealis</b> Offline		21:00 - 23:00	21:00 - 23:00	21:00 - 23:00	21:00 - 23:00	18:00 - 20:00	
<b>gaussian</b> Online	00:00 - 23:59	00:00 - 23:59	00:00 - 23:59	00:00 - 23:59	00:00 - 23:59	00:00 - 23:59	00:00 - 23:59
<b>X8_01</b> Offline		18:00 - 01:59	18:00 - 01:59	18:00 - 01:59	18:00 - 01:59		

Kuva 23. Quantum Devices -osio.

Tarkemmin lisätietoja Xanadun kvanttietokone-laitteistoista löytyy:

- <https://www.xanadu.ai/products/borealis/>
- <https://www.xanadu.ai/products/x-series/>

## Palvelun käytön vaihtoehdot, hinnoittelu ja käyttöönotto

Xanadu Cloud tarjoaa kolmea erityyppistä mallia palvelun käyttämiseen ja hinnoitteluun:

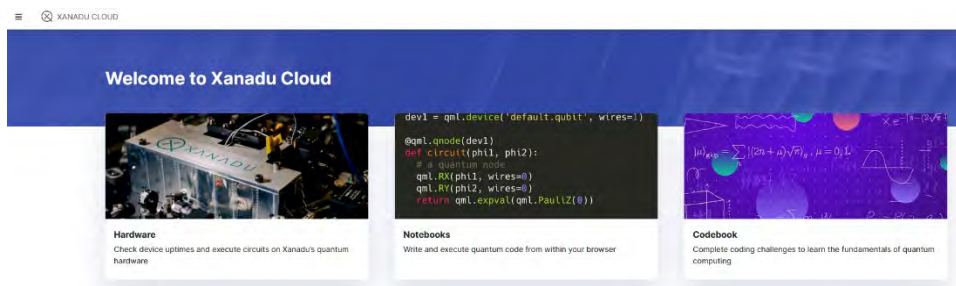
- **Free Tier;** ilmainen, tutustumis- ja kokeilukäyttöön, erityisesti opetus- ja tutkimuskäyttöön tarkoitettu palvelumalli. Mahdollistaa pääsyn Xanadun laitteiston, ohjelmistojen ja opetusresurssien käyttöön.
- **Flexible Tier;** Amazon Braketin kautta tarjottava, maksullinen palvelumalli. Tämän mallin hinnoittelusta lisätietoa löytyy AWS:n dokumentaatiosta osoitteessa <https://aws.amazon.com/braket/quantum-computers/xanadu/>
- **Partner Tier;** maksullinen, laajempia tarpeita varten, dedikoitu laitteiston käyttö

Tarkemmin palveluiden hinnoittelusta ja eri hinnoittelumallien sisällöistä <https://www.xanadu.ai/products/pricing>

Xanadu Cloud- palvelua pääsee testikäyttämään hyödyntämällä ylläesiteltyä Free Tier -mallia.

Tämän voi ottaa käyttöön rekisteröitymällä palveluun käyttäjäksi osoitteessa: <https://cloud.xanadu.ai/>

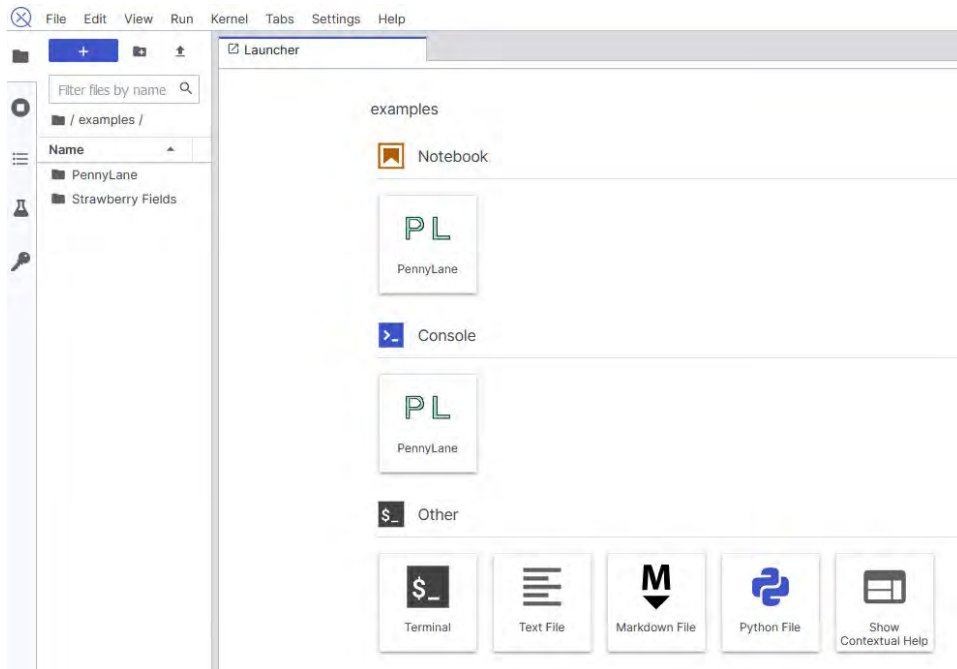
Rekisteröitymisen ja ensimmäisen kirjautumiskerran jälkeen siirrytään portaaliin, jonka kautta palvelun resurssien hallinta tehdään.



Kuva 24. Xanadu Cloud -portaali.

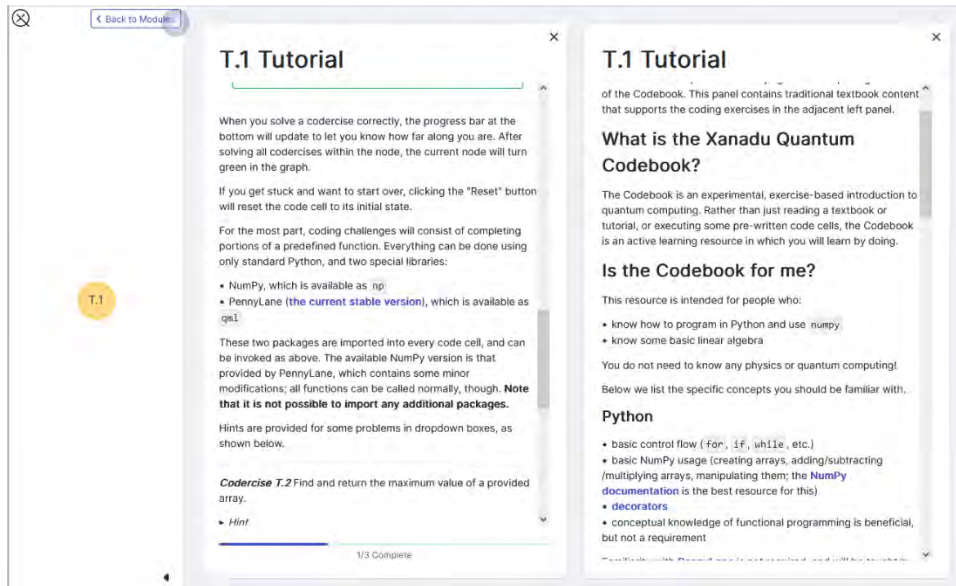
Portaalista on tarkisteltavissa palvelussa käytettävissä olevien kvanttietokoneiden tila ja saatavuus (Hardware-osio).

Portaalista pääsee myös **Notebooks**-osioon, joka on Jupyter Notebook- pohjainen kvanttiohjelmointiympäristö sovellusten rakentamiseksi Xanadun pilvipalveluun. Notebookit käyttävät Xanadun PennyLane -kirjastoa kvanttiohjelmointiin.



Kuva 25. Xanadu Cloud Notebooks.

Portaalin sisältämä Codebook -osio on Xanadun tarjoamien kvanttilaskentateknologioiden opiskeluun tarkoitettu interaktiivinen web-sivusto.



Kuva 26. Xanadu Cloud Quantum Codebook

Xanadu Quantum Codebook:in avulla voi samanaikaisesti sekä perehtyä kvanttiohjelmointiin että tehdä kuhunkin aiheeseen liittyviä esimerkkiohjelmia ja niiden suorituksen testausta.

## Tuettuja SDK- kehitysympäristöjä ohjelmointikieliä, kirjastoja ja sovelluskehityksiä

Xanadu Cloud hyödyntää erityisesti kahta sovelluskehityskirjastoa: PennyLanea ja Strawberry Fieldsiä.

PennyLane on Python-kielelle tarjottu kirjasto kvanttilaskentasovellusten rakentamiseen. Kirjaston idea on "write once – run anywhere" eli tavoite on mahdollistaa kvanttiohjelmien kirjoittaminen siten, että niitä voidaan ajaa useammalla eri alustalla. PennyLane sisältää ominaisuuksia erilaisten kvanttilaskentaa hyödyntävien sovellusten toteuttamiseen, ml. kvanttikoneoppimisen sovellukset.

PennyLanen dokumentaatio on saatavilla osoitteessa: <https://www.xanadu.ai/products/pennylane/>

PennyLane on käytettävissä Xanadu Cloud- pilvipalvelussa Notebook- kehitysympäristön kautta ja myös asennettavissa paikallisesti Python -kehitysympäristöön. Jälkimmäiseen vaihtoehtoon ohjeistus löytyy osoitteesta <https://pennylane.ai/install.html>

**Strawberry Fields** on Python- ohjelmointikielelle saatavilla oleva kirjasto, joka tarjoaa ominaisuuksia kvanttialgoritmien ja muiden kvanttiohjelmien ohjelmointiin, simulointiin ja suorittamiseen fotonikka- pohjaisilla kvanttietokoneilla (esim. Xanadun laitteisto). Strawberry Fields sisältää laajan tuen erityyppisten kvanttiohjelmien kirjoittamiseen, sisältäen mm. ominaisuuksia kvanttikoneoppimiskäytännöiden toteuttamiseen.

Strawberry Fieldsin dokumentaatio on käytettävissä sivustolta <https://www.xanadu.ai/products/strawberry-fields>

Myös Strawberry Fields on käytettävissä sekä Xanadu Cloud- pilvipalvelussa Notebook- kehitysympäristön kautta, että asennettavissa paikallisesti Python -kehitysympäristöön. Jälkimmäiseen vaihtoehtoon ohjeistus löytyy osoitteesta: <https://strawberryfields.readthedocs.io/en/stable/install.html>

**Lisätietoja:** Xanadu Cloud: <https://www.xanadu.ai/>

## 3.6 QCaaS-palvelut – kartoituksen johtopäätöksiä ja pohdintaa

Edellä esitetyn pohjalta ja muun, hankkeen aikana (2020–2022) tehdystä pilvipalveluina tarjottavien kvanttilaskentapalveluiden tarkastelusta voitiin karkeasti koostaa seuraavaa:

### **Tarjottujen palveluiden määrä ja monipuolisuus:**

- Kvanttilaskentaa tarjoavia pilvipalveluita on saatavilla useita kymmeniä erilaisia.
- Mukana on sekä pääsääntöisesti kvanttietokoneiden kehittämiseen keskittyneitä toimijoita (esim. D-Wave, Xanadu) mutta lisääntyvässä määrin muita perinteisiä pilvipalveluita tarjoavia toimijoita joihin kvanttilaskentaresurssien tarjonta on integroitu (esim. Microsoft Azure, AWS). Osalla toimijoista voi olla tarjolla perinteisiä

pilvipalveluita, mutta kvanttilaskentapalvelut ovat tarjolla erillisessä omassa pilvipalvelussa (esim. IBM Quantum)

- Pilvipalvelu voi sisältää mahdollisuuden ottaa kvanttilaskentaresursseja käyttöön samoilla tunnuksilla/tilauksilla, mitä ko. pilvipalvelua muutenkin käytetään (esim. Microsoft Azure, AWS).
- Palvelun käytön hinnoittelussa on jo nyt käytössä useita eri malleja: palveluntarjoajasta riippuen, voidaan valita joko käytön mukaan kustannusten muodostus ("pay-as-you-go") tai varata laskentakapasiteettia kiinteään hintaan (reserved) sekä myös mahdollisuus varata laitetaso kapasiteettia täysin omaan käyttöön (dedicated).
- Palvelut sisältävät laitetaso lisäksi myös kattavan joukon kvanttilaskentaa hyödyntävään sovelluskehitykseen tarvittavia ominaisuuksia; tuen erilaisille SDK-kehitysympäristöille, ohjelmakirjastoille ja sovelluskehittimille sekä ohjelmointikielille.

### **Laitetaso ratkaisut:**

- Palveluissa tarjotaan joko yksittäistä, yleensä palveluntarjoajan itsensä valmistamaa (mm. D-Wave, IBM, Xanadu) tai useampaa erilaista kvanttietokoneita (esim. Amazon Braket, Azure Quantum, QC Ware Forge)
- Eri pilvipalveluiden kautta on mahdollista päästä käyttämään käytännössä kaikkia yleisimpiä erityyppisiä kvanttietokoneita; annealer-, superconductive-, trapped ion-, photonics- sekä neutral atom- tyyppiset koneet löytyvät eri palveluntarjoajien valikoimasta jo nyt.
- Lähes kaikki palvelut tarjoavat myös simulaattoriratkaisuja, joilla ohjelmien testaaminen voi olla edullisempaa varsinaisen kvanttietokoneen käytön kustannuksiin verrattuna. Tämä voi olla merkittävä etu sovelluskehityksen kustannuksen ja joustavuuden kannalta.
- Uusia teknologioita hyödyntäviä kvanttietokoneita lisätään aiemman tarjonnan täydentämiseksi.

### **Pääsy laskentakapasiteetin käyttöön:**

- Valtaosa palveluntarjoajista tarjoaa pääsyn palveluun tutustumiseen ja testaamiseen.
- Tutustuminen on tyypillisesti ilmaista, ainakin rajoitettuun käyttöön (esim. simulaattorit, laskenta-aika, ajallisesti rajattu käyttö).



- Moni palveluntarjoajista (mm. D-Wave, IBM, Xanadu, Azure Quantum, Amazon Braket, QC Ware Forge) tarjoaa testimahdollisuuden myös aitojen kvanttietokoneiden käyttöön.
- Kaupallinen tuotantokäyttö mahdollistaa täysimittaisen, aitoja kvanttietokoneita hyödyntävien palveluiden käytön.

### **Ohjelmakirjastot, kehitysympäristöt, sovelluskehikset:**

- Lähes poikkeuksetta, jokaiseen pilvipalveluun on saatavilla yksi tai useampi kehitystyökalu kvanttilaskentaa hyödyntävien sovellusten rakentamiseen ja suorittamiseen palvelun tarjoamalla kvanttilaitteistolla.
- Työkalut ovat poikkeuksetta ilmaisia ja vapaita ottaa omaan henkilökohtaiseen kehityskäyttöön.
- Työkalut voivat olla palveluntarjoajan omia ratkaisuja (esim. Amazon Braket SDK, Microsoftin Q# & Quantum Development Kit, D-Waven Ocean) ja osa yleiskäyttöisempiä, useassa palvelussa tuettuja (etenkin Qiskit, mutta mm. Cirq, TensorFlow Quantum)
- Kehitysympäristöjen asennus ja käyttöönotto sekä käyttö eri sovelluskohteisiin on pääsääntöisesti kattavasti kuvattu.
- Kvanttiohjelmoinnin luonne poikkeaa toki vielä merkittävästikin perinteisestä sovelluskehityksestä, joten hyvänkin dokumentaation lisäksi peruskäsitteiden, konseptien ja kehityksen periaatteiden opetteluun kannattaa varata aikaa ja käyttää eri resursseja monipuolisesti. Erilaiset palveluiden tarjoavat tutoriaalit tukevat tätä tavoitetta.

### **Tuki eri ohjelmointikielille:**

- Valtaosa kartoituksissa vastaan tulleista kirjastoista sisälsivät tuen etenkin Python kielelle.
- Muita tuettuja kieliä ovat mm. Q# ja OpenQASM sekä joissakin tapauksissa mm. C++.
- Osassa palveluissa esim. kvanttipiirien toteuttamista tuetaan visuaalisilla työkaluilla. Oletettavasti tällaiset low code- tyyppiset ratkaisut voivat olla tuottavuutta parantavia myös kvanttiohjelmoinnissa ja yleistyvät jatkossa useampaan kehitysympäristöön.

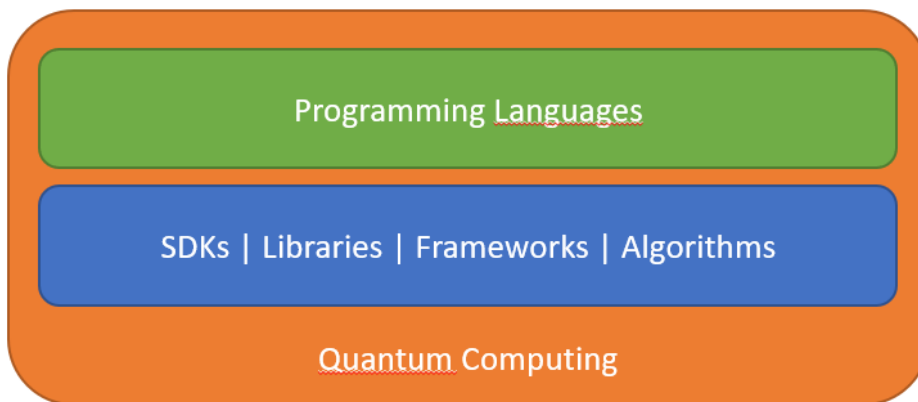
## **Yhteys pilvipalvelun muihin ominaisuuksiin tai muu, perinteisen tietotekniikan ja klassisten laskentaratkaisujen hyödyntäminen:**

- Osassa pilvipalveluissa hyödynnetään perinteisen pilvipalvelun resursseja kvanttilaskentaratkaisujen toteuttamisessa. Esimerkkejä näistä ovat mm. Jupyter Notebook - kehitysympäristöjen tarjoaminen web-palveluna kvanttiohjelmoinnin tukemiseksi tai tiedontallennus, käyttäjienhallinta yms. perinteisten resurssien käyttö kvanttilaskentapalvelun tarjoamisen tukena.
- D-Waven hyödyntämä "hybrid solver"- tekniikka yhdistelee kvanttilaskennan ja klassisen tietojenkäsittelyn periaatteita ja saman tyyppisiä tavoitteita on tunnistettavissa myös muissa palveluissa (esim. Amazon Braket:in hybrid jobs tai kvanttikoneoppimisen quantum-classical hybridit)

Yhteenvetona: pilvipalvelut sisältävät jo nyt kattavasti erilaisia ominaisuuksia esim. kvanttilaskentaa hyödyntävää ohjelmistokehitystä ajatellen, jonka käyttökelpoisuutta ja mahdollisuuksia on sekä erittäin hyödyllistä että myös tarpeellista tutkia, vaikka nykyhetken NISQ- kvanttietokoneiden ominaisuudet tai käytön hinnoittelu ei mahdollistaisi vielä täysimittaista, perinteiseen sovelluskehitykseen verrattavaa ohjelmistotuotannollista toimintaa.

# 4 Quantum Application Development: kvanttiohjelmointikieliä, kirjastoja, työkaluja ja ympäristöjä

Jarmo Talvivaara



Hankkeen aikana pyrittiin luomaan mahdollisimman kattava kartoitus saatavilla olevista ohjelmistokehitystyökaluista, joilla voidaan suunnitella, toteuttaa sekä suorittaa ohjelmia kvanttitietokoneille.

Tiedot ovat koostettu aikavälillä syksy 2020 – syksy 2022. Linkkejä-osiosta löytyy osoitteet verkkosivuille, joista ohjelmointiympäristöistä löytyy lisätietoja, mahdollisuus ympäristöön liittyvien kirjastojen tai muiden työkalujen imurointiin ja asennukseen sekä dokumentaatio ympäristön käyttämiseen.

## 4.1 Kvanttiohjelmointiympäristöjen kartoitus (2020–2022)

Ympäristö	Valmistaja, palvelun-tarjoaja	Tyyppi (pilvipalvelu, SDK, sovelluskehys, kirjasto, muu), lisenssi	Tuetut ohjelmointikiel	Suoritusympäristö:	Linkkejä: lisätietoja, työkalujen imurointi ja dokumentaatio:
<b>Amazon Braket SDK</b>	Amazon AWS	SDK- kehitysympäristö sovelluksien kehittämiseksi AWS:n Amazon Braket- kvanttilaskenta-pilvipalveluun.	Python	Molemmat	Sivusto: <a href="https://docs.aws.amazon.com/braket/index.html">https://docs.aws.amazon.com/braket/index.html</a>
<b>Blackbird</b>	Xanadu	Korkean tason ohjelmien kääntö kvanttiosessorilla suoritettavaksi.  Open Source.	Blackbird  Kirjastot C++ ja Python -kielille.	Molemmat	Sivusto: <a href="https://quantum-blackbird.readthedocs.io/">https://quantum-blackbird.readthedocs.io/</a>  Asennus: <a href="https://quantum-blackbird.readthedocs.io/en/latest/blackbird_python/installing.html">https://quantum-blackbird.readthedocs.io/en/latest/blackbird_python/installing.html</a>  Esimerkki käytöstä: <a href="https://quantum-blackbird.readthedocs.io/en/latest/blackbird_python/example.html">https://quantum-blackbird.readthedocs.io/en/latest/blackbird_python/example.html</a>  <a href="https://github.com/XanaduAI/blackbird">https://github.com/XanaduAI/blackbird</a>
<b>Cirq</b>	Google	Ohjelmakirjasto  Open Source	Python	Molemmat (Quantum Engine API)	Sivusto: <a href="https://quantumai.google/cirq">https://quantumai.google/cirq</a>  Asennus (Linux, Mac, Windows): <a href="https://quantumai.google/cirq/start/install">https://quantumai.google/cirq/start/install</a>  Esimerkki käytöstä: <a href="https://quantumai.google/cirq/start/start">https://quantumai.google/cirq/start/start</a>

<b>cQASM</b>	QCA Lab, TU-Delft	Korkean tason ohjelmien kääntö kvanttiprosessoilla suoritettavaksi.	QASM	Molemmat.	Sivusto: <a href="https://www.quantum-inspire.com/kbase/cqasm/">https://www.quantum-inspire.com/kbase/cqasm/</a>  Asennus: <a href="https://github.com/QuTech-Delft/quantuminspire">https://github.com/QuTech-Delft/quantuminspire</a>  Esimerkkejä: <a href="https://www.quantum-inspire.com/kbase/code-examples/">https://www.quantum-inspire.com/kbase/code-examples/</a>
<b>Forest</b>	Rigetti	SDK- kehitysympäristö, ohjelmakirjastoja  Open Source	Python	Molemmat	Imuointi: <a href="https://qcs.rigetti.com/sdk-downloads">https://qcs.rigetti.com/sdk-downloads</a>  Asennus ja esimerkki käytöstä: <a href="https://pyquil-docs.rigetti.com/en/v2.3.0/start.html">https://pyquil-docs.rigetti.com/en/v2.3.0/start.html</a>  Dokumentaatio: <a href="https://pyquil-docs.rigetti.com/en/v2.3.0/">https://pyquil-docs.rigetti.com/en/v2.3.0/</a>
<b>Intel-QS</b>  (Intel Quantum Simulator)	Intel	Simulaattori.	C++, Python	Simulaattori.	Sivusto, käyttöönotto, esimerkkejä: <a href="https://github.com/intel/Intel-QS">https://github.com/intel/Intel-QS</a>
<b>myQLM</b>	Atos	Osittain Open Source	Python	Simulaattori	Sivusto: <a href="https://myqlm.github.io/">https://myqlm.github.io/</a>  Asennus: <a href="https://myqlm.github.io/myqlm_specific/install.html">https://myqlm.github.io/myqlm_specific/install.html</a>  Esimerkki käytöstä: <a href="https://myqlm.github.io/getting_started.html">https://myqlm.github.io/getting_started.html</a>
<b>Ocean</b>	D-Wave	SDK- kehitysympäristö, ohjelmakirjastoja  Open Source.	Python	Molemmat	Sivusto: <a href="https://docs.ocean.dwavesys.com/en/stable/">https://docs.ocean.dwavesys.com/en/stable/</a>  Asennus ja esimerkkejä käytöstä: <a href="https://docs.ocean.dwavesys.com/en/stable/getting_started.html">https://docs.ocean.dwavesys.com/en/stable/getting_started.html</a>

<b>Open-QASM</b>	IBM	Korkean tason ohjelmien kääntö kvanttiprosessoilla suoritettavaksi.	QASM	Molemmat	Sivusto: <a href="https://openqasm.com/">https://openqasm.com/</a>
<b>PennyLane</b>	Xanadu	Kirjasto, erityisesti koneoppimiseen.  Open Source	Python	Molemmat	Sivusto: <a href="https://www.xanadu.ai/pennylane">https://www.xanadu.ai/pennylane</a>  Asennus ja käyttöönotto: <a href="https://docs.pennylane.ai/en/stable/development/guide/installation.html">https://docs.pennylane.ai/en/stable/development/guide/installation.html</a>  Dokumentaatio, esimerkkejä käytöstä: <a href="https://docs.pennylane.ai/en/stable/development/guide/documentation.html">https://docs.pennylane.ai/en/stable/development/guide/documentation.html</a>
<b>ProjectQ</b>	ETH Institute for Theoretical Physics	SDK  Open Source	Python	Molemmat	Sivusto, dokumentaatio käyttöönottoon ja käyttämiseen: <a href="https://github.com/ProjectQ-Framework/ProjectQ">https://github.com/ProjectQ-Framework/ProjectQ</a>
<b>QC Ware Forge</b>	QC Ware	SDK	Python	Molemmat	Sivusto: <a href="https://qcware.readthedocs.io/en/latest/index.html">https://qcware.readthedocs.io/en/latest/index.html</a>
<b>Qiskit</b>	IBM	SDK  Open Source	Python, Swift, Java	Molemmat	Sivusto: <a href="https://qiskit.org/">https://qiskit.org/</a>  Asennus, käyttöönotto ja esimerkkejä käytöstä: <a href="https://qiskit.org/documentation/getting_started.html">https://qiskit.org/documentation/getting_started.html</a>
<b>Quil</b>	Rigetti (Smith, Curtis & Zeng)	Korkean tason ohjelmien kääntö kvanttiprosessoilla suoritettavaksi (ns. quantum instruction set)	Quil  PyQuil –kirjasto Pythonille	qvm –virtuaalikone	Sivusto, ohjeita käyttöön: <a href="https://github.com/quil-lang/quil">https://github.com/quil-lang/quil</a>

<b>Quantum Development Kit</b>	Microsoft	SDK	Q#	Molemmat	<p>Sivusto: <a href="https://www.microsoft.com/en-us/quantum/development-kit">https://www.microsoft.com/en-us/quantum/development-kit</a></p> <p>Asennus (Jupyter Notebook, Visual Studio): <a href="https://learn.microsoft.com/en-us/azure/quantum/install-overview-qdk">https://learn.microsoft.com/en-us/azure/quantum/install-overview-qdk</a></p> <p>Esimerkkejä käytöstä: <a href="https://learn.microsoft.com/en-us/azure/quantum/user-guide/">https://learn.microsoft.com/en-us/azure/quantum/user-guide/</a></p>
<b>QConvert</b>	Quantastica	Kvanttiohjelmien muuntaminen eri kieliltä toiseen.	Useita		<p>Sivusto: <a href="https://quantastica.com/#converters">https://quantastica.com/#converters</a></p> <p>Lataaminen ja asennus (komentoriviversio): <a href="https://www.npmjs.com/package/q-convert">https://www.npmjs.com/package/q-convert</a></p> <p>Selainversio: <a href="https://quantum-circuit.com/qconvert">https://quantum-circuit.com/qconvert</a></p> <p>Dokumentaatio: <a href="https://quantum-circuit.com/docs">https://quantum-circuit.com/docs</a></p>
<b>QPS (Quantum Programming Studio)</b>	Quantastica	Selainpohjainen, visuaalinen ohjelmointiympäristö	Visuaalinen editori tuottaa OpenQASM -koodia	Molemmat	<p>Sivusto: <a href="https://quantastica.com/#qps">https://quantastica.com/#qps</a></p> <p>Selainkäyttöliittymä: <a href="https://quantum-circuit.com/home">https://quantum-circuit.com/home</a></p> <p>Dokumentaatio: <a href="https://quantum-circuit.com/docs">https://quantum-circuit.com/docs</a></p>
<b>Silq</b>	SRI Lab, ETH Zurich	<p>Korkean tason kvanttiohjelmointikieli</p> <p>Soveltaa useita lisenssejä (lisätietoja)</p>	<p>Silq</p> <p>Silq- laajennus VSCode- ohjelmointiympäristöön.</p>	Simulaattori.	Sivusto: <a href="https://silq.ethz.ch">https://silq.ethz.ch</a>

<b>Strawberry Fields</b>	Xanadu	Sovelluskehys, kirjasto Open Source	Python	Molemmat	Sivusto: <a href="https://strawberryfields.ai/">https://strawberryfields.ai/</a>  Asennus: <a href="https://strawberryfields.readthedocs.io/en/stable/install.html">https://strawberryfields.readthedocs.io/en/stable/install.html</a>  Dokumentaatio: <a href="https://strawberryfields.readthedocs.io/en/stable/">https://strawberryfields.readthedocs.io/en/stable/</a>
<b>pytket</b>	Quantinuum (alkup. t iket> Cambridge Quantum Computing)	Sovelluspino, kääntäjä	Python (pytket)	Tukee useita laitearkkitehtuureja	Sivusto, asennusohjeet: <a href="https://cqcl.github.io/tket/pytket/api/index.html">https://cqcl.github.io/tket/pytket/api/index.html</a>  Dokumentaatio, esimerkkejä käytöstä: <a href="https://cqcl.github.io/tket/pytket/api/getting_started.html">https://cqcl.github.io/tket/pytket/api/getting_started.html</a>
<b>TensorFlow Quantum</b>	Google	Koneoppimiskirjasto (hybrid-quantum-classical), erityisesti koneoppimiseen (" <i>.. for rapid prototyping of hybrid quantum-classical ML model</i> ")	Python	Molemmat	Sivusto: <a href="https://www.tensorflow.org/quantum">https://www.tensorflow.org/quantum</a>  Asennus, käyttöönotto: <a href="https://www.tensorflow.org/quantum/install">https://www.tensorflow.org/quantum/install</a>  Esimerkkejä käytöstä: <a href="https://www.tensorflow.org/quantum/tutorials/hello_many_worlds">https://www.tensorflow.org/quantum/tutorials/hello_many_worlds</a>



Tarkastellaan seuraavaksi tarkemmin muutamia kartoituksen sisältämiä erilaisia kvanttiohjelmointiympäristöjä ja -kehitysympäristöjä. Valinnassa hyödynnettiin eri kehitystyökalujen ominaisuuksia, mahdollisia eroja sekä samanlaisuuksia eri valmistajien, ohjelmointikielten, pilvipalveluiden sekä erityyppisten kvanttietokoneiden kannalta.

Valitut ympäristöt ovat:

- Qiskit
- Microsoft Q# SDK
- Ocean SDK

## 4.2 Qiskit

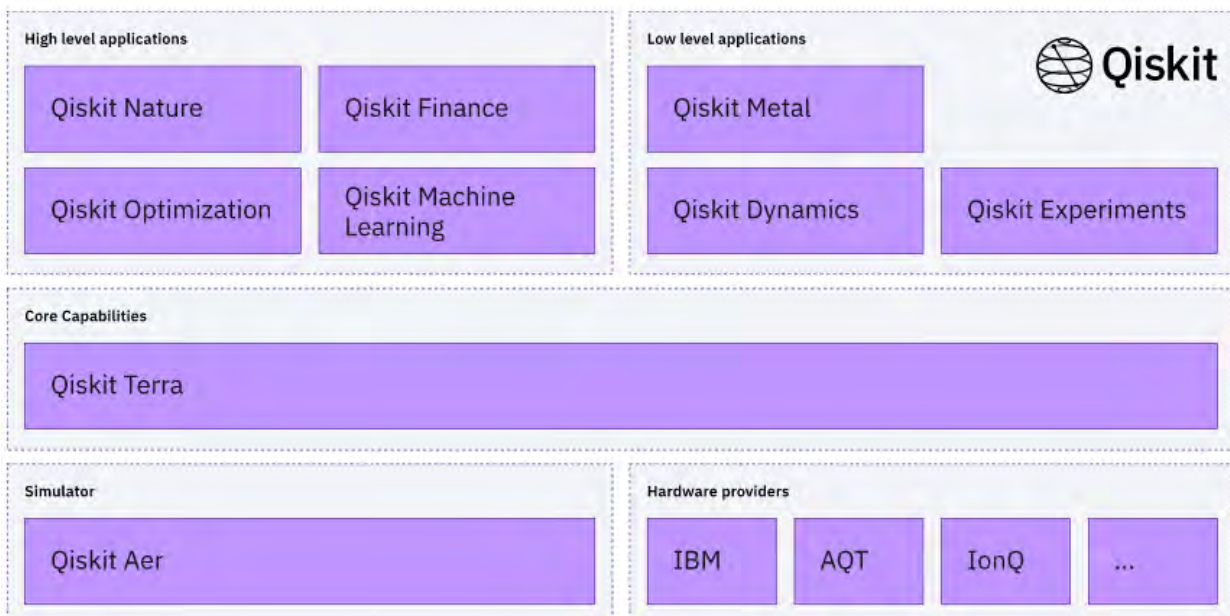
Qiskit on avoimen lähdekoodin (open source) ohjelmistokehitysympäristö kvanttietokoneiden ohjelmointiin ja simulointiin. Qiskit on saatavilla helposti ladattavana ja käytönotettavana ohjelmointikirjastona Python- kielisten kvanttiohjelmien toteuttamiseen. Qiskit onkin yksi suosituimpia kehitysympäristöjä ohjelmien kirjoittamiseen kvanttietokoneille. Tarvittaessa, Qiskitin lähdekoodi on saatavilla GitHub- versionhallintapalvelun kautta.

Qiskit on kehitetty IBM:n toimesta ja tarjoaa käyttäjilleen laajan valikoiman työkaluja kvanttilgoritmiin suunnitteluun, simulaatioon ja suorittamiseen sekä kvanttietokoneiden käyttämiseen esim. pilvipalveluna.

Qiskitin soveltaminen noudattaa usein seuraavia vaiheita:

- Ohjelman (kvanttipiirin) suunnittelu ja rakentaminen, "toteutus" (*Build*): suunnittele kvanttipiiri tai -piirit, jotka kuvaavat ratkaistavaa ongelmaa.
- Toteutetun ohjelman (kvanttipiirin) kääntäminen (*Compile*): käännä kvanttipiirit sisältämä ohjelmakoodi tarkoituksen mukaiselle palvelulle, esim. kvanttietokoneelle tai niiden toimintaperiaatetta noudattaville (klassisille) simulaattoreille.
- Käännetyn ohjelman suoritus (*run*): suorita käännetty koodi tarkoituksenmukaisella kvanttipalvelulla (paikallisella tai palveluna saatavilla olevalla simulaattorilla tai kvanttietokoneella).
- Analysointi (*analyze*): Laske yhteenvedon tilastotiedot ja visualisoi kokeiden tulokset.

Qiskit sisältää laajan tuen erilaisten kvanttilaskentaa hyödyntävien ohjelmien (quantum programming) toteuttamiseen:



Kuva 27. Qiskit -kehitysympäristön osat (Qiskit 2022)

### Ydintoiminnallisuudet (core capabilities)

- **Qiskit Terra:** ohjelmistokehys, joka tarjoaa työkaluja kvanttiohjelmien suunnitteluun, kääntämiseen, optimointiin ja suorittamiseen. Se on yksi osa laajempaa Qiskit-kvanttitietokonekehystä, joka sisältää myös Qiskit Aqua (kvanttialgoritmit), Qiskit Ignis (virheenkorjaus ja mittaus) ja Qiskit Aer (kvanttisimulaattori).

### Korkean tason soveltaminen (High level applications)

- **Qiskit Nature:** avoimen lähdekoodin kehys Qiskitiin, joka tukee kvanttietokonealgoritmien käyttöä luonnontieteellisten ongelmien ratkaisemisessa.
- **Qiskit Finance:** Qiskit- sovelluskehys, joka sisältää ominaisuuksia rakentaa ratkaisuja esim. osake- ja arvopaperiongelmiin, rahoitusongelmien ratkaisemiseksi, kuten salkun optimointi, sekä data-analyysityökaluja reaaliaikaisen tai satunnaisen datan käyttöön rahoituskokeissa.
- **Qiskit Optimization:** korkean tason sovelluskehys ja joukko algoritmeja erilaisten optimointiongelmien ratkaisuun kvanttiohjelmoinnin avulla.
- **Qiskit Machine Learning:** Qiskitin sovelluskehys, jolla voidaan soveltaa kvanttilaskennan menetelmiä ja etuja koneoppimISRatkaisujen toteuttamiseen.

## Matalan tason soveltaminen (Low level applications)

- Qiskit Metal: ohjelmakirjasto kvanttilaitteiden suunnitteluun ja simulointiin.
- Qiskit Dynamics: kvanttijärjestelmien rakentamiseen, muuntamiseen ja ratkaisemiseen Qiskitillä.
- Qiskit Experiments: kirjasto ja sovelluskehys erilaisten kokeilujen/testien ajamiseen kvanttilaitteistolla.

## Simulaattorit ja laitetason ratkaisut (simulator, hardware providers), mm.

- Qiskit Aer (simulaattori)
- IBM:n kvanttietokoneet (superconductive QPUt)
- AQT:n (Alpine Quantum Technologies) simulaattorit ja kvanttietokoneet (trapped Ion QPUt)
- IonQ:n kvanttietokoneet (trapped ion QPUt)
- Muitten kumppanien ratkaisut (mm. Microsoft Azure Quantum)

## Lähteitä ja lisätietoja:

- Qiskit Documentation: <https://qiskit.org/documentation/index.html>
- Introduction to Qiskit. [https://qiskit.org/documentation/intro\\_tutorial1.html](https://qiskit.org/documentation/intro_tutorial1.html)
- Qiskit:in GitHub- repository. <https://github.com/Qiskit>
- Qiskit Quantum Circuits: <https://qiskit.org/documentation/tutorials.html#quantum-circuits>
- Qiskit Nature: <https://qiskit.org/documentation/nature/>
- Qiskit Finance: <https://qiskit.org/documentation/finance/>
- Qiskit Optimization: <https://qiskit.org/documentation/optimization/>
- Qiskit Machine Learning: <https://qiskit.org/documentation/machine-learning/>

## 4.2.1 Qiskit-esimerkkisovellus

Seuraavassa esitellään yksinkertainen Python-kielinen esimerkkisovellus, joka on toteutettu käyttämällä Qiskit kirjastoa.

Ohjelman käyttämät Qiskit:in ominaisuudet ovat seuraavat:

- 1) Tuodaan tarvittavat Qiskit- moduulit
  - i. QuantumCircuit; ohjelmistollisten kubittien ja kvanttipiirien luomiseksi tarkoitettu komponentti.
  - ii. Aer: simulaattori ohjelman suorittamiseksi

- iii. execute: rutiinit kvanttipiirien suorittamiseksi määritellyllä backend-suorittimella (simulaattori tai kvanttietokone)
- 2) alustetaan ohjelmaan kahden kubitin kvanttipiiri
- 3) asetetaan piiriin kubitit ns. Bell State- tyyppiseen lomittuneeseen tilaan
- 4) talletetaan kubittien mittaustulokset kahteen klassiseen bittiin sekä
- 5) tulostetaan tulokset.

Esimerkkihjelman suorittamisen voi tehdä paikallisella Qiskit- kirjaston sisältävässä Python-ohjelmointiympäristössä, Jupyter Notebook -ympäristössä tai IBM Quantum- tai Googlen Colab -pilvipalveluissa. Ratkaisun toteutuksessa on hyödynnetty Qiskit:in tutoriaalia: [https://qiskit.org/documentation/tutorials/circuits/1\\_getting\\_started\\_with\\_qiskit.html](https://qiskit.org/documentation/tutorials/circuits/1_getting_started_with_qiskit.html)

```
# Tuodaan tarvittavat Qiskit-moduulit
from qiskit import QuantumCircuit, Aer, execute

# Alustetaan kvanttipiiri (qc = quantum circuit), jossa on 2 kubit-
# tia ja 2 klassista bittiä
qc = QuantumCircuit(2, 2)

# Asetetaan ensimmäinen kubitti superpositioon Hadamard-portin (Ha-
# damard gate) avulla
qc.h(0)

# Lomitetaan (entangle) kaksi kubittia käyttämällä kontrolloitua-X-
# porttia (CNOT gate)
qc.cx(0, 1)

# Mitataan kahden kubitin tilat ja tallennetaan tulokset klassi-
# sille biteille
qc.measure([0, 1], [0, 1])

# Asetetaan kvanttipiiri suoritettavalle backendille (tässä: simu-
# laattorille) ja suoritetaan ohjelma
simulaattori = Aer.get_backend('qasm_simulator')
job = execute(qc, simulaattori, shots=1024)

# Hae suorituksen tulokset
tulos = job.result()

# Hae määrät kunkin tilan esiintymiselle ja tulosta ne
```

```
lkm = tulos.get_counts(qc)
print(lkm)
```

Kuten ohjelman ensimmäiset rivit osoittavat, tarvitsee Python -suoritusympäristöön asentaa Qiskit'in ohjelmakirjasto. Tämä voidaan tehdä esim. pip -komentoa käyttämällä.

```
pip install qiskit
```

Esimerkkihjelma käyttää suorittamiseen Qiskitin Aer -simulaattoria. Jos ohjelman suoritus halutaan tehdä simulaattorin sijaan esim. IBM:n aidolla kvanttietokoneella (esim. 16 kubitin Melbourne), tarvitsee sovelluksen koodia muuttaa alla olevan koodin mukaiseksi (ks. etenkin lihavoidut kohdat)

```
# Tuodaan tarvittavat Qiskit-moduulit
from qiskit import QuantumCircuit, IBMQ, execute
```

```
# Ladataan IBM Quantum -tili
IBMQ.load_account()
```

```
# Valitaan kvanttikoneen tarjoaja ja laite
provider = IBMQ.get_provider('ibm-q')
backend = provider.get_backend('ibmq_16_melbourne')
```

```
# Alustetaan kvanttipiiri (qc = quantum circuit), jossa on 2 kubit-
tia ja 2 klassista bittiä
qc = QuantumCircuit(2, 2)
```

```
# Asetetaan ensimmäinen kubitti superpositioon Hadamard-portilla
qc.h(0)
```

```
# Kytetään kubitit toisiinsa CNOT-portilla
qc.cx(0, 1)
```

```
# Mittaamme molemmat kubitit ja tallennamme tulokset klassisiin bit-
teihin
qc.measure([0, 1], [0, 1])
```

```
# Suoritetaan kvanttiportti valitulla kvanttilaitteella
job = execute(qc, backend=backend)
```

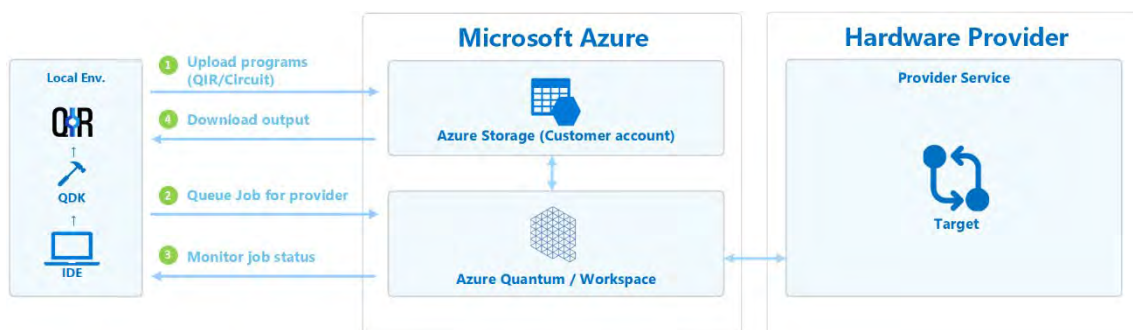
```
# Haetaan työn tulokset
tulos = job.result()
```

```
# Tulostetaan jokaisen tilan lukumäärä mittauksen tuloksena
lkm = tulos.get_counts(qc)
print(lkm)
```

## 4.3 Microsoft Quantum Development kit ja Q# -kieli

Microsoftin Quantum Development Kit (QDK) on sovelluskehitysympäristö, jota käytetään rajapintana ohjelmien kehittämiseen Microsoftin Azure Quantum-pilvipalvelun sisältämille kvanttilaskentaresursseille. QDK tarjoaa tuen kvanttialgoritmien kirjoittamiseen Q# -kieltä käyttäen. QDK tukee myös Pythonia, Cirq- ja Qiskit- kirjastojen tuen kautta.

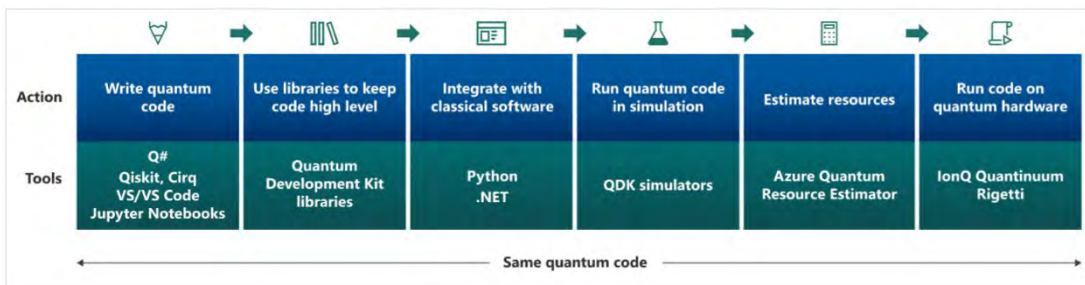
QDK:lla ohjelmat voidaan kirjoittaa paikallisesti asennetussa kehitysympäristössä ja testata simulaattorilla tai välittää Azure -pilvipalvelun Quantum Workspace- palveluun määritellyille resursseille (ts. simulaattoreille tai aidoille kvanttitietokoneille) suoritettavaksi.



Kuva 28. Työnkulku Azure Quantum -palvelussa. (Kuva: Microsoft)

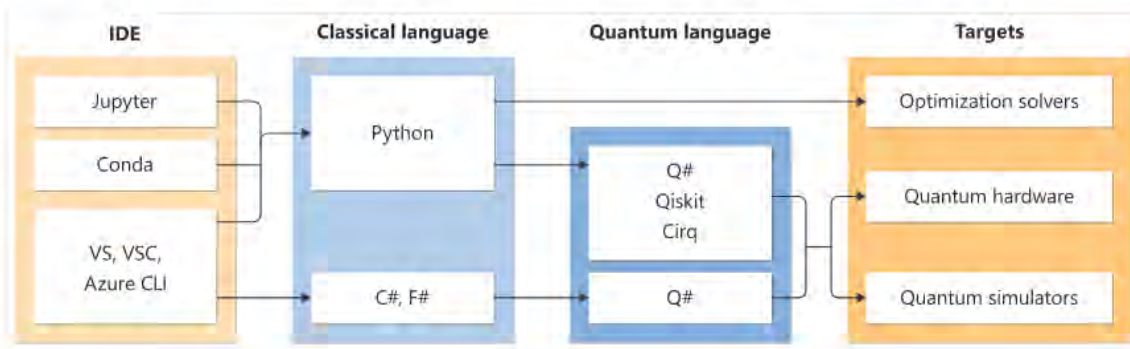
Q# on avoimen lähdekoodin kehitystyöhön pohjautuva, korkean tason ohjelmointikieli, joka on tarkoitettu erityisesti kvanttialgoritmien kehittämiseen ja suorittamiseen. Q# on suunniteltu riippumattomaksi käytettävästä kvanttilaitteistosta, että sillä olisi mahdollista kehittää erityyppisiä sovelluksia, eri kvanttilaskennan käyttökohteisiin sekä optimoida niiden suoritusta. Syntaksiltaan Q# muistuttaa mm. C#, JavaScript ja C++ -kieliä. Vaikkakin Q# -kielen kvanttilaskennasta johtuva erilainen konteksti poikkeaa em. kielten perinteiseen sovelluskehitykseen soveltamisesta, voi Q# -kielen syntaksin samanlaisuus sekä ilmaisuvoimaisuus olla eduksi erilaisten kvanttiohjelmien ohjelmoinnin

opettelussa ja soveltamisessa. Lisäksi Q# -kieli on laajasti dokumentoitu sekä hyvin tuettu kehitysympäristössä monipuolisen soveltamisen mahdollistamiseksi. Toki Q# ei ole läheskään niin yleinen kieli kvanttiohjelmoinnissa eri kehitysympäristöjen lähes poikkeuksetta tukemaan Python-kieleen verrattuna.



Kuva 29. Prosessin eteneminen ja työkalut. (Kuva: Microsoft)

Quantum Development Kit:iä voidaan hyödyntää usein eri tavoin; Visual Studioon tai Visual Studio Codeen käyttöönotettavina lisäosina tai Azureen määritellyssä Jupyter Notebook- kehitysympäristössä.



Kuva 30. Microsoft Quantum Machine learning (Kuva: Microsoft)

Quantum Development Kit ja Q# - kielen käyttöönotto esim. Visual Studioon on tehtävissä seuraavin toimenpitein: <https://learn.microsoft.com/en-us/azure/quantum/install-overview-qdk?tabs=tabid-vs%2Ctabid-conda#use-q-and-python-with-visual-studio-and-visual-studio-code>

**Lähteitä ja lisätietoja:**

Q# and the Quantum Development Kit. <https://azure.microsoft.com/en-us/resources/development-kit/quantum-computing/>

### 4.3.1 Q#-esimerkkisovellus

Q# -kielisen ohjelman kirjoittamiseksi tarvitaan seuraavia työkaluja ja toimenpiteitä:

- 1) Tarvittaessa
  - i. Visual Studio tai Visual Studio Code IDE- kehitysympäristön asennus.
  - ii. Azure- komentoriviohjelmaan (Azure CLI) quantum- laajennuksen asentaminen (tätä tarvitaan, jos ohjelmia välitetään Azureen suoritettavaksi Visual Studion komentoriveiltä)
  - iii. Yhteensopivan .NET -ympäristön asennus (tarkista yhteensopivuus QDK:n dokumentaatiosta)
- 2) Kehitysympäristöön yhteensopivan Quantum Development Kit:in asentaminen.
- 3) Uuden projektin luonti ja sovelluksen kirjoittaminen

Luodaan edellisen Qiskit- esimerkkisovelluksen mukainen kvanttiohjelma, Microsoftin Q# -kieltä käyttäen.

```
namespace Bell {  
  
    // Operaatio Bell-tilan luomiseksi kahden kubitin välille  
    operation BellState(q1 : Qubit, q2 : Qubit) : Unit {  
  
        // Asetetaan ensimmäinen kubitti (q1) superpositioon Hadamard-  
portilla  
        H(q1);  
  
        // Kytetään kubitit toisiinsa CNOT-portilla  
        CNOT(q1, q2);  
    }  
  
    // Testiohjelma, joka luo Bell-tilan ja mittaa siinä kytkettyjen ku-  
bittien tilat  
    operation TestBellState() : Result[] {
```



```

// Alustetaan kaksi kubittia
using (qubits = Qubit[2]) {

    // Luodaan Bell-tila BellState- operaatiolla
    BellState(qubits[0], qubits[1]);

    // Mitataan molempien kubittien tilat ja palautetaan arvot
    return [M(qubits[0]), M(qubits[1])];
}
}
}

```

Ohjelma käyttää Q# -kielen sisältämää Qubit -tietotyyppiä kubittien luomiseen. Alustamisen yhteydessä kerrotaan luotavien kubittien määrä. Esimerkiohjelma luo kahden kubitin sisältämän kokoelman. Qubit- tietotyypistä lisätietoja löytyy Q# -kielen dokumentaatiosta: <https://learn.microsoft.com/en-us/azure/quantum/user-guide/language/typesystem/quantumdatatypes>

Tämän jälkeen ohjelma hyödyntää kahta operaatiota: BellState ja TestBellState. BellState-operaatio ottaa vastaan kaksi kubittia q1 ja q2 ja luo niiden välille Bell-tilan. TestBellState-operaatio luo kaksi kubittia, kutsuu BellState-operaatiota ja mittaa lopuksi molemmat kubitit. BellState-operaatiossa ensimmäinen kubitti asetetaan superpositioon Hadamard-portilla ( $H(q1)$ ), ja kubitit kytketään toisiinsa CNOT-portilla ( $CNOT(q1, q2)$ ), jolloin niiden tila muuttuu lomittuneeksi Bell-tilaksi.

Esimerkiohjelman sisältämästä  $H()$  -kutsulla suorittamasta Hadamard- porttioperaatiosta ja muista operaatioista löytyy lisätietoja dokumentaatiosta: <https://learn.microsoft.com/en-us/qsharp/api/qsharp/microsoft.quantum.intrinsic>

TestBellState-operaatiossa alustetaan kaksi kubittia käyttämällä using-lauseketta, joka varmistaa, että kubitit vapautetaan operaation suorittamisen jälkeen. Tämän jälkeen operaatio kutsuu BellState-operaatiota luodakseen Bell-tilan ja mittaa molemmat kubitit käyttämällä M(qubit)-operaatiota, joka palauttaa mitatun kubitin tilan (1 tai 0). Lopuksi operaatio palauttaa (return) mittauksen tulokset listana.

- Lisätietoja Q# -kielen käyttämisessä: <https://learn.microsoft.com/en-us/azure/quantum/user-guide/>
- Lisätietoja Q# sisältämistä kirjastoista (perustoiminnot, koneoppiminen, jne.): <https://learn.microsoft.com/en-us/azure/quantum/user-guide/libraries/>

- Q#- kielen ja QDK- kehitysympäristön tukemat suoritusympäristöt (Azure Quantum- palvelun simulaattorit ja kvanttietokoneet): <https://learn.microsoft.com/en-us/azure/quantum/qc-target-list>
- QDK:lla toteutettujen ohjelmien suorituksen ohjaus Azuren suoritusympäristöille: <https://learn.microsoft.com/en-us/azure/quantum/how-to-submit-jobs?source=recommendations&pivots=ide-portal>

## 4.4 D-Wave Ocean SDK

Ocean SDK on D-Waven kvanttietokoneiden hyödyntämiseen suunniteltu kehitysympäristö. Ocean SDK sisältää työkaluja, kirjastoja ja dokumentaatiota, jotka on suunniteltu auttamaan kehittäjiä toteuttamaan, suorittamaan ja optimoimaan kvanttietokonesovelluksia D-Waven: annealer- tyyppisillä kvanttietokoneilla. Ohjelmien kehityksessä tuetaan Python-kieltä.

Ocean SDK sisältää erilaisia komponentteja, kuten:

- D-Wave System API: Matalan tason API- ohjelmointirajapinta, joka tarjoaa pääsyn D-Wave:n kvanttiprosessoreihin (QPUs).
- Ocean Software: System API:n päälle rakennettuja Python-kirjastoja, helpottamaan kvanttisovellusten kehittämistä. .
- Ocean Tools: Tukityökalut, jotka auttavat kehittäjiä suunnittelemaan ja ajamaan kvanttikoekteita, mukaan lukien mahdollisuus visualisoida tuloksia ja tehdä vianmääritystä.
- Esimerkit ja demot: Kokoelma esimerkkejä ja demoja, jotka havainnollistavat Ocean SDK:n käyttöä erilaisten ongelmien ratkaisemisessa. Esimerkkejä ovat mm. logistiikkaan, koneoppimiseen ja rahoitukseen liittyvät sovellukset (esim. optimointitehtävät)

Ocean SDK on saatavilla osoitteessa: <https://docs.ocean.dwavesys.com/en/stable/>

Ocean SDK voidaan ottaa käyttöön paikallisesti asennettuun Python -kehitysympäristöön tai tehdä ohjelmien kehitystä Leap- pilvipalvelun Visual Studio Code- pohjaisella, selainkäyttöisellä Leap IDE- kehitystyökalulla.

```

1 from dwave.system import DWaveSampler, EmbeddingComposite
2 import dimod
3
4 # Alustetaan D-Wave-laitteen tiedot
5 sampler = EmbeddingComposite(DWaveSampler(solver={'topology__type': 'chimera'}))
6
7 # Luodaan kaksi qubitia ja asetetaan ne Bell-tilaan
8 qubits = [0, 1]
9 h = {qubits[0]: -1, qubits[1]: 1}
10 J = {(qubits[0], qubits[1]): 2}
11 model = dimod.BinaryQuadraticModel(h, J, 0.0, dimod.SPIN)
12 response = sampler.sample(model, num_reads=1)
13
14 # Tulostetaan lopputulos
15 for sample, energy in response.data(['sample', 'energy']):
16     print(sample, energy)

```

Terminal output:

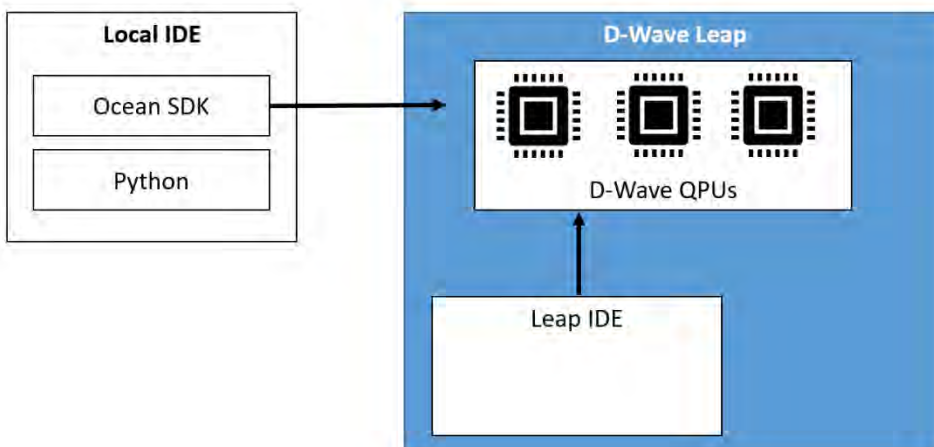
```

Leap IDE /workspace/dwquantumcomputing $ /usr/bin/env /usr/local/bin/python /ide/extensions/ms-python/python/pythonFiles/lib/python/debugpy/launcher 36855 -- /workspace/dwquantumcomputing/Testi_001.py
{0: 1, 1: -1} -4.0
Leap IDE /workspace/dwquantumcomputing $

```

Kuva 31. Ocean SDK.

Paikallisesti kirjoitetut ohjelmat voidaan konfiguroida suoritettavaksi simulaattorilla tai D-Waven pilvipalveluna tarjoamilla kvanttietokoneilla.



Kuva 32. Ocean SDK: toimintaperiaate.

Ocean SDK konfigurointi suoritusten ohjaamiseksi Leap -palvelussa tarjotuilla D-Waven kvanttietokoneilla määritellään D-Wave Solver API- rajapinnan (SAPI) avulla. Rajapinta käyttää kehitysympäristöllä tehdyn ohjelman ja suoritussympäristöjen välillä autentikointiin ja kommunikointiin Leap- palvelussa määritellyjä käyttäjäkohtaisia API Token-avaimia.

SAPIn tarkempi dokumentaatio ja esimerkkejä laitteistojen määrittelyyn löytyy Ocean SDK:n dokumentaatiosta <https://docs.ocean.dwavesys.com/en/stable/overview/sapi.html>

## 4.4.1 Ocean SDK –esimerkkisovellus

Toteutetaan aiempien sovellusten tyylinen kvanttiohjelma Python-ohjelmointikielellä, nyt käyttäen D-Waven Ocean SDK -kehitysympäristöä.

```
# Tuodaan tarvittavat kirjastot
from dwave.system import DWaveSampler, EmbeddingComposite
import dimod

# Alustetaan D-Wave-laitteen tiedot
sampler = EmbeddingComposite(DWaveSampler(solver={'topology__type':
'chimera'}))

# Luodaan kaksi qubittia ja asetetaan ne Bell-tilaan
qubits = [0, 1]
h = {qubits[0]: -1, qubits[1]: 1}
J = {(qubits[0], qubits[1]): 2}
model = dimod.BinaryQuadraticModel(h, J, 0.0, dimod.SPIN)
response = sampler.sample(model, num_reads=1)

# Tulostetaan lopputulos
for sample, energy in response.data(['sample', 'energy']):
    print(sample, energy)
```

Koodi alustaa ensin D-Wave-laitteen tiedot käyttäen `DWaveSampler`-luokkaa ja `EmbeddingComposite`-luokkaa. Tämän jälkeen luodaan kaksi qubittia, joiden tilaa muutetaan Bell-tilaksi asettamalla qubittien välille yhteys ja asettamalla qubittien omille ajurille Hadamard-portit.

Qubittien tilaa hallitaan energiafunktioilla, jonka avulla lasketaan lopputulos. Energiafunktiossa on qubittien ajurien (`qubits[0]` ja `qubits[1]`) painot (`h`) ja kytkennät (`J`) määritelty. Lopuksi käytetään `DWaveSampler`-luokan `sample`-metodia lopputuloksen laskemiseksi.

Lopuksi tulostetaan lopputulos, joka sisältää qubittien tilat ja energia-arvon.

Kuten ohjelman ensimmäiset rivit osoittavat, tarvitsee Python -suoritusympäristöön tällä kertaa asentaa Ocean- ohjelmointiympäristö (D-Waven Ocean SDK). Tämä voidaan tehdä esim. pip -komentoa käyttämällä.

```
pip install dwave-ocean-sdk
```

Esimerkkihjelman suorittamisen voi tehdä Qiskit- esimerkin tapaan mm. paikallisessa Ocean SDK- kehitysympäristön sisältävässä Python-ohjelmointiympäristössä, Jupyter Notebook -ympäristössä tai D-Waven Leap IDE - kehitysympäristössä.

Lisää esimerkkejä ja tarkempi dokumentaatio Ocean SDK:n käyttämisestä D-Waven - kvanttietokoneita hyödyntävien ohjelmien kehittämiseksi löytyy sivustolta:  
[https://docs.ocean.dwavesys.com/en/stable/getting\\_started.html](https://docs.ocean.dwavesys.com/en/stable/getting_started.html)

# 5 Quantum Machine Learning: kvanttikoneoppimisen perusteita ja mahdollisuuksia

Jarmo Talvivaara

Raportin tässä osiossa esitellään perusteita kvanttikoneoppimiseen liittyen. Koneoppiminen on yksi tapa toteuttaa erilaisia tekoälyn sovellutuksia. Koska kvanttikoneoppiminen on yksi tapa soveltaa, laajentaa tai tehostaa perinteistä koneoppimista, käydään aluksi läpi yleisiä perusteita koneoppimisesta. Osa näistä perusteita ovat relevantteja ja siksi hyvä tuntee myös kvanttikoneoppimista tarkasteltaessa.

## 5.1 Tekoäly, koneoppiminen ja syväoppiminen

Tekoäly (eng. artificial intelligence, AI) on tietojenkäsittelytieteen ala, joka tutkii ihmisen älyä ja muita ominaisuuksia simuloivien toiminnallisuuden soveltamista tietokoneille. Tällaisia ominaisuuksia ovat esim.

- päättely ja päätöksenteko ("ajattelu"),
- visuaalinen havainnointi ("näkeminen" ja "tunnistaminen"),
- puheentunnistus ("kuulo")
- kielen käsittely ("tulkinta")

Tekoälyn kehittämisessä hyödynnetään erilaisia tekniikoita kuten koneoppimista, luonnollisen kielen käsittelyä ja syväoppimista mahdollistamaan koneiden oppimisen datasta ja suorituskyvyn parantamisen ajan myötä.

*Tekoälyyn ja sen osa-alueisiin liittyy paljon englanninkielisiä termejä ja lyhenteitä, jotka ovat vakiintuneet käyttöön niin englannin- kuin muun kielisissäkin materiaaleissa; tämän vuoksi raportissa esitellään yleisimmistä termeistä sen englanninkielinen versio sekä tyypillisesti käytetty lyhenne (esim. koneoppiminen, machine learning, ML). Tämän on tarkoitus tukea raportin sisältämien peruskäsitteiden ymmärtämistä ja suhteita, sekä niihin liittyvien lähteiden käyttöä sekä lisätiedon hakemista.*

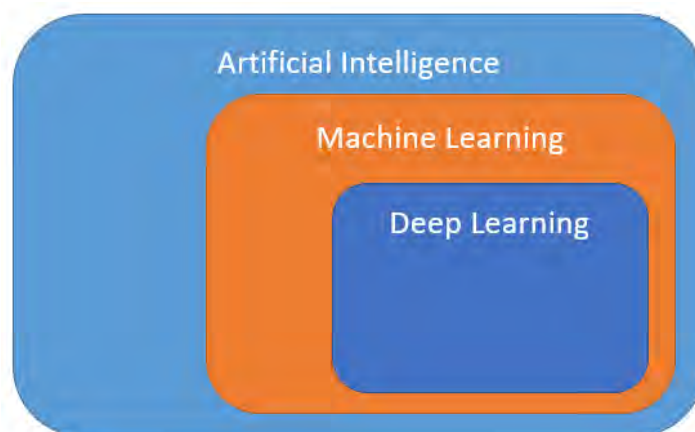
Tekoälyllä on useita alahaaroja, kuten koneoppiminen, syväoppiminen, luonnollisen kielen käsittely (natural language processing, NLP), konenäkö (computer vision, CV), robotiikka (robotics) ja asiantuntijajärjestelmät (expert systems).

**Luonnollisen kielen käsittelyssä** kehitetään algoritmeja, jotka ymmärtävät ja tuottavat ihmiskieltä. Näitä ominaisuuksia hyödyntävät esimerkiksi erilaisissa laitteissa (mm. mobiilipuhelimet) tai ohjelmistoissa olevat ennakoiva tekstinsyöttö, chatbot:it tai Chat-GPT:n tyyppiset tekoälyverkkopalvelut.

**Konenäkö** keskittyy algoritmien kehittämiseen, jotka voivat tulkita, analysoida ja generoida visuaalisia tietoja, kuten kuvia ja videoita. Esimerkkejä tästä ovat teollisuudessa sovelletut konenäköratkaisut materiaalien laadun tarkkailuun, kuvankäsittelyohjelmistojen edistyneemmät ominaisuudet tai vaikka DALL·E2 -tyyliset, luonnollisen kielen tekstistä kuvia tuottavat verkkopalvelut.

**Robotiikka** tutkii sellaisten älykkäiden koneiden luomisen mahdollisuuksia, jotka voivat vuorovaikuttaa fyysisen maailman kanssa. Asiantuntijajärjestelmät keskittyvät sellaisten järjestelmien kehittämiseen, jotka voivat tehdä päätöksiä ja ratkaista ongelmia joukon sääntöjen ja tiedon perusteella.

Tässä raportissa keskitytään tarkastelemaan tekoälyn osa-alueista erityisesti koneoppimista sekä syväoppimista.



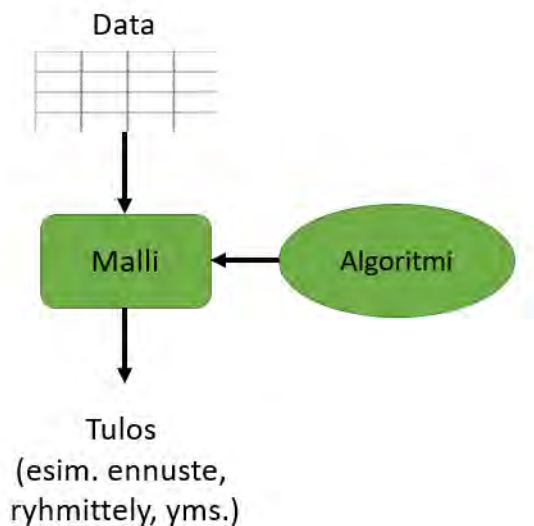
## 5.1.1 Koneoppiminen

**Koneoppiminen** (*machine learning, ML*) on yksi tekoälyn osa-alue, jota sovelletaan kasvavassa määrin erilaisissa ohjelmistoissa, verkkopalveluissa ja muissa tietoteknisissä ratkaisuisissa. Valtaosa koneoppimisen ratkaisuista toteutetaan käyttäen perinteisiä tietokoneita sekä niissä klassiseen laskentaan pohjaavia algoritmeja. Vastaavia

periaatteita voidaan kuitenkin soveltaa myös kvanttilaskentaa hyödyntävässä koneoppimisessa; algoritmit ja tiedon käsittely toki tapahtuu siinä kvanttilaskennan keinoin. Koneoppimisessa keskitytään sellaisten algoritmien kehittämiseen ja soveltamiseen, jotka voivat oppia toimintasääntöjä ja malleja sen opetukseen käytettävästä datasta sekä parantaa suorituskyykyään ajan mittaan ilman, että niitä on tarve ohjelmoida erikseen. Tämän vuoksi koneoppimista hyödyntäviä algoritmeja voidaan soveltaa tuottamaan sellaisia ohjelmistojen osia, joissa kaikkea päätöksentekoa tai muuta logiikkaa ei tarvitse ohjelmoida etukäteen määriteltävällä tavalla tai vain etukäteen tiedettävien vaihtoehtojen ja oletusten mukaisesti.

Tällainen on hyödyllistä esim. silloin, kun rakennetaan analytiikkaa, päätöksentekoa tai automaatiota soveltavia ohjelmistoja, joilla on tarve "sopeutua" monipuolisemmin erilaisiin, esim. päättelyä ja päätöksentekoa vaativiin tilanteisiin mihin perinteinen, etukäteen määriteltä logiikkaa noudattavaa ohjelma ei pysty.

Järjestelmä oppii syötetyn saatavilla olevan datan pohjalta "käyttäytymismalleja" (patterns), jonka pohjalta koneoppivan ratkaisun on tavoite tuottaa ratkaisuja myös tietoon tai tilanteisiin, jotka ovat aiemmin olleet tuntemattomia.

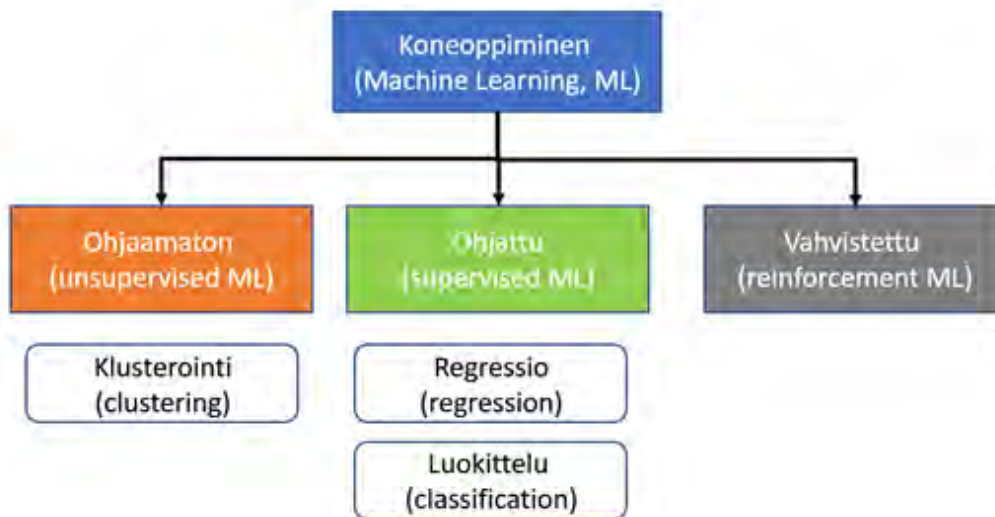


Kuva 33. Koneoppimismallin opettaminen (Kelleher, J. 2019)

Tavoitteita koneoppimiselle voivat olla esim. erilaiset datan ryhmittelyt (*clustering*) ja luokittelut (*classification*), diskreettien tai jatkuvien arvojen ennustaminen (*logistinen ja lineaarinen regressio*) tai tapahtumakulkujen tunnistaminen (esim. päätospuut, decision trees).



Koneoppiminen jaetaan usein kolmeen erityyppiseen luokkaan sen mukaan, miten algoritmit opetetaan ja millaisia tuloksia niiltä odotetaan.



Kuva 34. Koneoppimisen osa-alueita (Kelleher, J. 2019)

Yksinkertaistettuna koneoppiminen toteutetaan hyödyntämällä seuraavia vaiheita:

1. Määritellään **ongelma** tai ongelmat, joihin koneoppimisella on tarkoitus löytää ratkaisu (esim. datan ryhmittely, luokittelu tai ennustaminen)
2. Valitaan kuhunkin ongelmaan sopiva **algoritmi**, joilla ongelmanratkaisu on optimaalisimmin toteutettavissa: esim.
  - a. Datan ryhmittely -> **klusterointi** -algoritmi (*clustering*)
  - b. Datan luokittelu -> **luokittelu**-algoritmi (*classification*)
  - c. Ennustaminen -> **regressio**-algoritmi (*regression*)
3. **Opetetaan** valittu algoritmi aiemmalla, ratkaistavasta ongelmasta tietoa sisältävällä tai sitä muistuttavalla **datalla** koneoppimis-malliksi (machine learning model); malliin liittyvä algoritmi käy opetusdatan läpi, opettelee datasta löytyvien "säännönmukaisuuksia" ja säätää itsensä noudattamaan tästä opittua tapaa datan käsittelyyn. Jos mallin opettaminen onnistuu hyvin, kykenee se sopeutumaan käsittelemään myös sellaista ongelmaan liittyvää tai ilmiötä muistuttavaa dataa, jota opetusdata ei itsessään sisällä.

4. **Sovelletaan** luotua mallia ohjelmiston osana tarvittavissa kohdissa, jossa tarvitaan sen kykyä ratkaista määritelty ongelma.

Kvanttilaskentaa hyödyntävää koneoppimista (QML, Quantum Machine Learning) tutkitaan paljon ja sille on esitetty erilaisia sovelluskohteita, jopa nykyisen NISQ –aikakauden kvanttietokoneille toteutettavaksi. Tämän vuoksi perinteisille koneoppimisalgoritmeille on kehitetty niiden toimintaperiaatetta ja tavoitteita noudattavia, mutta kvanttilaskennalla toteutettavia versioita. Esimerkkejä tällaisista ovat esim. datan ryhmittelyssä käytettävät klusterointialgoritmit, regressioalgoritmit tai kvanttilaskentaa hyödyntävät luokittelualgoritmit (ks. esim. Schuld 2017 tai Khan & Robles-Kelly 2020). Perinteisen ja kvanttikoneoppimisen algoritmeja esitellään ja vertaillaan tarkemmin raportissa myöhemmin.

## 5.1.2 Syväoppiminen

Syväoppiminen (*deep learning, DL*) on miellellävissä koneoppimiseen sisältyvä sovellusalue, joka mahdollistaa yksinkertaisempia koneoppimismalleja huomattavasti monipuolisempia tekoälyratkaisujen toteuttamista. Syväoppimisella on paljon erilaisia tekoälyn käytännön sovelluskohteita, kuten esim. kuvan- ja puheentunnistus (*image and speech recognition*), itsenäiset ajoneuvot (*self-driving vehicles*), petosten havaitsemisessa (*fraud detection*), lääketieteellisessä diagnoosissa (*medical diagnostic*) ja henkilökohtaisissa suosituksissa (*personalized recommendation engines*).

Syväoppimisessa tyypillinen keino oppimisen toteuttamisessa ovat keinotekoiset neuroverkot (*ANN, Artificial Neural Network*) ja niistä kehitetyt erikoistapaukset, esim. takaisinkytketty neuroverkko: (*RNN Recurrent Neural Network*), syvät ns. konvoluutio-neuroverkkot (*CNN, Convolutional Neural Network*) tai GAN –verkot (*generative adversarial networks, GAN*).

Kvanttineuroverkkojen soveltamista tutkitaan jo ns. NISQ- aikakauden kvanttietokoneille. Kvanttilaskennan hyödyntämistä neuroverkkojen soveltamisessa on tutkittu ja kehitetty varsin paljon. Yhtenä lopputuloksena tästä on mm. ”kvanttiversiot” perinteisistä neuroverkoista, kuten esim. QNN eli (basic) Quantum Neural Networks (kvanttiversio perus-neuroverkoista), QCNN eli Quantum Convolutional Neural Networks tai QGAN eli Quantum Generative Adversarial Networks.

### **Lähteitä ja lisätietoja:**

Beer, K. Quantum Neural Networks. 2022. <https://arxiv.org/abs/2205.08154>

Euroopan Parlamentti. 2020. Mitä tekoäly on ja mihin sitä käytetään? <https://www.europarl.europa.eu/news/fi/headlines/society/20200827STO85804/mita-tekoaly-on-ja-mihin-sita-kaytetaan>

Goodfellow, I, Bengio, Y, Courville, A. 2016. Deep Learning. <https://www.deeplearningbook.org/contents/intro.html>

Haikonen, P. 2017. Tietoisuus, tekoäly ja robotit. Art House.

Khan, T., Robles-Kelly, A. 2020. Machine Learning: Quantum vs Classical. IEEE Access. <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9274431>

Kelleher, J. 2019. Syväoppiminen - kuinka tekoäly toimii. Terra Cognita.

Kwak, et. al. 2021. Quantum Convolutional Neural Networks. <https://arxiv.org/abs/2108.01468>

National Institute of Standards and Technology (NIST). Artificial intelligence. 2022. <https://www.nist.gov/artificial-intelligence>

Schuld, et al. 2014. An introduction to quantum machine learning. <https://arxiv.org/abs/1409.3097>

Wikipedia. 2022. Artificial Intelligence. [https://en.wikipedia.org/wiki/Artificial\\_intelligence](https://en.wikipedia.org/wiki/Artificial_intelligence)

Wikipedia. 2022. Machine Learning. [https://en.wikipedia.org/wiki/Machine\\_learning](https://en.wikipedia.org/wiki/Machine_learning)

Wikipedia. 2022. Deep Learning. [https://en.wikipedia.org/wiki/Deep\\_learning](https://en.wikipedia.org/wiki/Deep_learning)

Wikipedia. 2022. Quantum Neural Network. [https://en.wikipedia.org/wiki/Quantum\\_neural\\_network](https://en.wikipedia.org/wiki/Quantum_neural_network)

### 5.1.3 Tekoäly, koneoppiminen ja klassisen laskennan haasteita

Edellä esiteltyjä tekoälyn ja koneoppimisen ratkaisuja toteutetaan vielä toistaiseksi pääosin käyttämällä perinteistä tietotekniikkaa ja klassista laskentaa.

Tekoälyn soveltamiseen käytettävät laitteistot ja ohjelmistot mahdollistavat hyvin korkean tason sovelluskehityksen, ohjelmointikielet ja -kirjastot sisältävät erittäin ilmaisuvoimaisia tapoja käsitellä isoja datamassoja ja monimutkaisia algoritmeja. Kuitenkin; näissäkin tapauksissa laskentaoperaatiot pohjautuvat perinteisen tietojenkäsittelyn tapaan klassisten bittien ja niiden operaatioiden tasolla tietokoneiden toiminnan ohjaimiseen.

Tästä tyypillisesti seuraa se, että tehokkaimmissakin tekoälyratkaisuissa sovelletut klassiset tietojenkäsittelyn ratkaisut kohtaavat samat rajoitukset, kuin sen soveltamisessa muissa ohjelmistokehityksen osa-alueissa; perinteisillä tietokoneilla olevat haasteet riittävän monimutkaisten ja vaativien, esim. suuresta datan määrästä johtuvien operaatioiden toteuttaminen riittävän tehokkaasti voi olla haastavaa ja joissakin tapauksissa jopa mahdotonta. Tehokkuus ei tässä tarkoita pelkästään suoritusnopeutta vaan esim. koneoppivan mallin toteuttamisen riittävää monimutkaisuutta ja kattavuutta tarkasteltavan ilmiön kuvaamisessa. Tällä voi olla esim. lääketieteellisessä tutkimuksessa iso merkitys.

Tietenkin tekoälyn, koneoppimisen ja etenkin syväoppimisen ratkaisuihin on kehitetty ja sovellettu paljon laskentaa tehostavia ratkaisuja; esimerkiksi suurteho- ja rinnakkaislaskentaa (high-performance and parallel computing) tai perinteisten CPU- prosessorien sijaan erityyppiseen laskentaan (matriisien ja tensorien) käsittelyyn tehokkaammin ratkaisut laskevat GPU- ja TPU -piirit.

Näillä teknologioilla saavutetaan merkittäviä etuja esim. syväoppimisessa syvien neuroverkkojen opettamisen tehokkuuden ja monipuolistamisen lisäämisessä.

## 5.2 Kvanttikoneoppiminen (Quantum Machine Learning, QML)

Miten edellä esitetyt perinteisen koneoppimisen periaatteet sitten liittyvät kvanttilaskentaan? Jos liittyvät, niin onko kvanttilaskennalla nähtävissä etuja kone- tai syväoppimisen tai muun tekoälyn osa-alueiden kehittämisessä?

Perinteisiä (klassisia) koneoppimisalgoritmeja voidaan tarkastella kvanttilaskennan kannalta esimerkiksi seuraavista näkökulmista:

1. Korvataan klassiset algoritmit kvanttilaskentaa vastaavilla "kvanttikoneoppimisalgoritmeilla"
2. Yhteiskäytetään klassisia ja kvanttikoneoppimisalgoritmeja (quantum-classical hybrid)
3. Hyödynnetään kvanttilaskentaa tehostamaan tarvittavia vaiheita klassisen koneoppimisen sovellutuksissa.

Kvanttilaskennan tuomia parannuksia ja uusia mahdollisuuksia, verrattuna klassisen laskennan rajoituksiin, voidaan soveltaa tulevaisuudessa entistä laajemmin eri tietojenkäsittelyn osa-alueisiin. Tämän vuoksi on oletettavissa, että nämä edut ovat hyödynnettävissä myös koneoppimisen ratkaisuisissa. Näitä mahdollisuuksia tarkastelee erityisesti kvanttikoneoppimisen otsikon alla tapahtuva tutkimus.

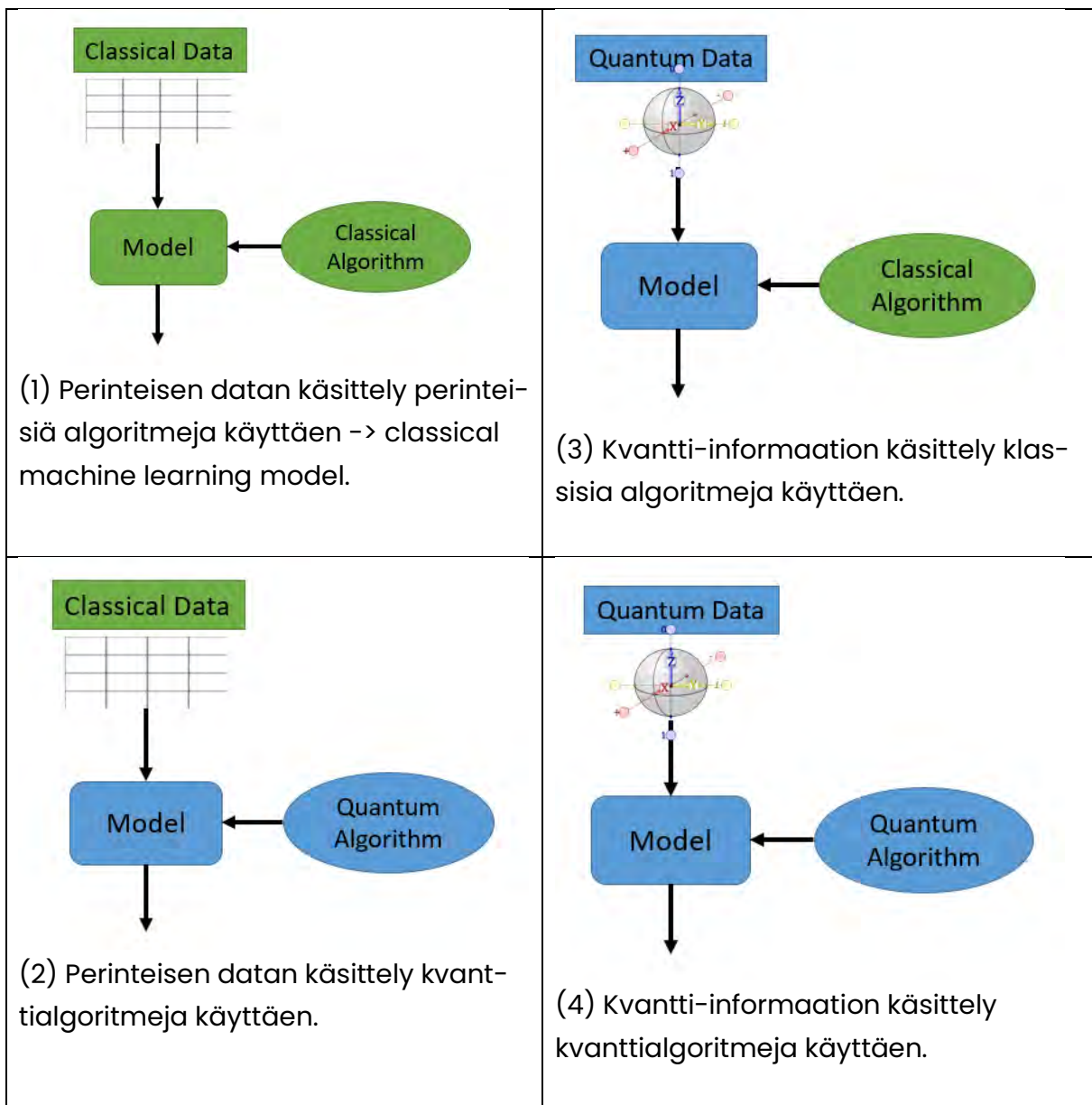
Perinteisessä koneoppimisessa dataa käsittelevät perinteiset tietokoneet, joiden bitteihin perustuva toimintaperiaate mahdollistaa vain yhden operaation suorittamisen kerrallaan, joten koneoppimisen vaatimat eri operaatiot joudutaan suorittamaan peräkkäin.

Kvanttitietokoneissa kubitit voivat olla useassa tilassa samanaikaisesti, mikä mahdollistaa suorittaa monia laskutoimituksia rinnakkain. Tämä kvanttitietokoneiden rinnakkaiskäsitteilykyky voi mahdollisesti johtaa merkittäviin parannuksiin myös koneoppimisalgoritmien nopeudessa ja tehokkuudessa.

Kvanttikoneoppiminen keskittyy erityisesti kvanttitietokoneissa toimivien algoritmien kehittämiseen ja soveltamiseen. Näitä algoritmeja voidaan käyttää monenlaisiin

sovelluksiin, kuten perinteistäkin koneoppimista; esim. luokitteluun, klusterointiin ja regressioon tai esimerkiksi neuroverkkojen avulla toteutettavat syväoppimisen ratkaisut. Esimerkkejä voivat olla kvanttilaskennan hyödyntäminen koneoppimisalgoritmien kehittämisessä ja "kvantti-versioiden" kehittäminen perinteisistä keinotekoisista neuroverkoista (QANN, quantum artificial neural networks), kokonaisten uudenlaisten kvanttialgoritmien kehittäminen (esim. pattern recognition –haasteiden ratkaisemiseen),  
Vielä toistaiseksi, ns. NISQ- aikakauden kvanttietokoneet ovat rajoittuneita yleispätevien kvanttikoneoppimISRatkaisujen tekemiseen. Tähän vaikuttavat mm. kvanttietokoneiden kubittien määrä, laatu (*fidelity*), koherenssi ja porttioperaatioiden kesto riittävän pitkien laskentaoperaatioiden toteuttamiseen ja mahdolliset kohinaan liittyvät haasteet. On kuitenkin tutkittu, että nykyisiäkin NISQ- aikakauden koneita voidaan soveltaa tehostamaan koneoppimISRatkaisuja kvanttilaskennan keinoin.

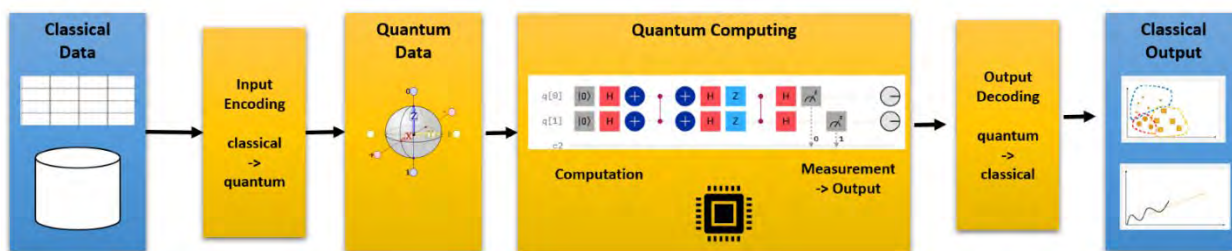
Kvanttikoneoppiminen (quantum machine learning, QML) voidaan nähdä olevan yhdistelmä kvanttiinformatiosta ja kvantti- ja klassisista koneoppimismalleista (soveltaen Qiskit 2022).



Kuva 35. Datun ja algoritmien soveltamisen vaihtoehtoja klassisessa ja kvanttilaskennassa. (Soveltaen Qiskit Textbook – Quantum Machine Learning)

### Klassinen data ja kvanttialgoritmit

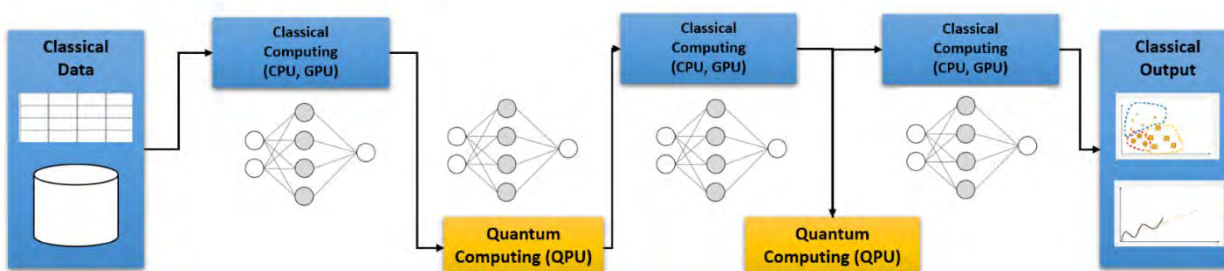
Kuvassa 36 oleva vaihtoehto 2 (perinteisen datan käsittely kvanttialgoritmeilla) voitaisiin toteuttaa esimerkiksi seuraavien vaiheiden mukaisesti:



Kuva 36. Perinteisen datan käsittely kvanttialgoritmeilla. (Soveltaen TheEdgeAi.io 2023)

Ensimmäinen vaihe sisältää klassisessa muodossa olevan datan (tekstimuotoinen, numeerinen, kuva, ääni, jne.) lataamisen. Tämän jälkeen data on koodattava kvantti-informaatiomuotoon (*input encoding*), että seuraavissa vaiheissa dataan kohdistuva kvanttilaskenta (*quantum computation*) sekä tulosten mittaaminen (*measurement*) mahdollistuu. Lopputuloksena saatava kvantti-informaation muodossa oleva tulos (*output*) muunnetaan (*decoding*) klassisen datan muotoon ja on käsiteltävissä perinteisillä tietokoneilla. Vaihtoehdon etuna voi olla kvanttilaskennan mahdollisesti merkittävät edut laskennan tehokkuudessa (*computation*), mutta etenkin datan muuntaminen *encoding* (klassinen->kvantti-informaatio) ja *decoding* (kvantti->klassinen informaatio) -vaiheissa voi muodostaa ison haasteen suorituskyvyllä. Tämä tietenkin vaikuttaa kokonaisuuden laskennan suorituskykyyn.

Yksi vaihtoehto olisi yhdistellä datan käsittelyssä suorittaa osa prosesseista klassisella tietokoneella ja osa kvanttietokoneella; esimerkiksi neuroverkkojen osalta näitä vaiheita voidaan delegoida klassisen ja kvanttilaskennan välillä. Tällaisia malleja nimitetään usein "classical-quantum"- tai "quantum-classical"-hybrideiksi.



Kuva 37. Esimerkki klassista ja kvanttilaskentaa yhdistelevästä koneoppimismallista. (Soveltaen TheEdgeAi.io 2023)

Tällaista mallia voisi olla mahdollista hyödyntää esim. neuroverkkojen opettamisen tehostamisessa tai neuroverkon hyperparametrien optimoinnissa.



Tällaisia hybridi- malleja tuetaan mm. Qiskit -kirjastossa (mm. kvanttilaskennan Qiskit-kirjaston ja klassisen koneoppimisen PyTorch- kirjaston "silta") tai TensorFlow Quantum- sovelluskehyksessä. Jos kvanttilaskennan yhteydessä on muutenkin hyödyllistä käyttää perinteisiä laskentaresursseja, vaikuttaa nämä mahdollisuudet tietenkin käyttökelpoisilta myös kvanttilaskentaa hyödyntävissä koneoppimiseratkaisuissa.

Esimerkkejä kvanttikoneoppimisesta, jossa käsitellään perinteistä dataa kvanttialgoritmien avulla, löytyy mm. Qiskit:in "Quantum machine learning" sekä Microsoftin "Azure Quantum Machine Learning" – opiskelumateriaaleissa. Kvanttidatan periaatteista ja käsittelystä esimerkkejä on saatavilla mm. TensorFlow Quantum -dokumentaatiosta.

### **Lähteitä ja lisätietoja:**

Abbas, A., Sutter, D., Zoufal, C., Lucchi, A., Figalli, A., Woerner, S. 2020. The power of quantum neural networks. <https://arxiv.org/abs/2011.00027>

Coyle, B. 2022. Machine learning applications for noisy intermediate-scale quantum computers. <https://arxiv.org/abs/2205.09414>

Garcia et al. 2022. Systematic Literature Review: Quantum Machine Learning and its applications <https://arxiv.org/abs/2201.04093>

Gupta, Zia. 2001. Quantum Neural Networks. <https://arxiv.org/pdf/quant-ph/0201144.pdf>  
IBM. Quantum Machine Learning. <https://research.ibm.com/topics/quantum-machine-learning>

Mancilla, J., Pere, C. 2022. A Preprocessing Perspective for Quantum Machine Learning Classification Advantage in Finance Using NISQ Algorithms. [https://www.researchgate.net/publication/365402581\\_A\\_Preprocessing\\_Perspective\\_for\\_Quantum\\_Machine\\_Learning\\_Classification\\_Advantage\\_in\\_Finance\\_Using\\_NISQ\\_Algorithms](https://www.researchgate.net/publication/365402581_A_Preprocessing_Perspective_for_Quantum_Machine_Learning_Classification_Advantage_in_Finance_Using_NISQ_Algorithms)

Microsoft. Azure Quantum Machine Learning. <https://learn.microsoft.com/en-us/azure/quantum/overview-quantum-machine-learning>

Peters, et al. 2021. Machine learning of high dimensional data on a noisy quantum processor. Nature. <https://www.nature.com/articles/s41534-021-00498-9>

Qiskit. Quantum Machine Learning. <https://qiskit.org/documentation/machine-learning/>

Schuld, et al. 2014. An introduction to quantum machine learning. <https://arxiv.org/abs/1409.3097>

TensorFlow Quantum. Quantum Machine Learning Concepts. <https://www.tensorflow.org/quantum/concepts>

## 5.3 Esimerkkejä kvanttikoneoppimisessa käytetyistä algoritmeista

Kvanttikoneoppiminen toteutetaan perinteisen koneoppimisen tapaan hyödyntämällä kuhunkin tapaukseen tarkoituksenmukaisia algoritmeja; ohjelmistollista logiikkaa, jolla kukin koneoppimiseen liittyvä ongelma saadaan parhaiten ratkaistua.

Algoritmi yhdistetään opetusdataan ja yhdistelmän tuottamasta lopputuloksesta käytetään tyypillisesti nimitystä malli (*model*). Malli sisältää säännöt, jotka syntyvät lopputuloksena algoritmin sovittamisesta mahdollisimman optimaalisesti opetusdatan sisältämien piirteiden mukaisesti.

Tämän vuoksi niin perinteisessä kuin kvanttikoneoppimisessä algoritmit liittyvät usein samantyyppisten ongelmien ratkaisuun. Esimerkiksi klusterointia käytetään datan ryhmittelyyn ja ryvästämiseen, luokittelualgoritmeja datan luokitteluun tai lineaarista tai logistista regressiota ennusteiden tekemiseen.

On kuitenkin hyvä huomioida, että algoritmien taustalla olevat matemaattiset periaatteet ja laskennalliset menetelmät eroavat perinteisen laskennan ja kvanttilaskennan välillä. Joitakin esimerkkejä samankaltaisista klassisista ja kvanttikoneoppimisalgoritmeista:

**Pääkomponenttianalyysi (PCA):** PCA on klassinen algoritmi, jota käytetään käsiteltävissä tiedossa olevien ulottuvuuksien vähentämiseen (eng. dimensionality reduction). PCA:ta käytetään usein vähentämään ominaisuuksien määrää tietoaaineistossa, samalla kuitenkin säilyttäen mallin kannalta tärkein tieto.

Kvanttilaskentaan pohjaava PCA ("Quantum PCA") on kvanttilaskennallinen algoritmi, joka voi saavuttaa saman tavoitteen tehokkaammin käyttämällä rinnakkaisuutta ja loimitumista.

**Tukivektorikoneet (SVM):** SVM on klassinen algoritmi, jota käytetään etenkin datan luokitteluun ja regressioon.

**QSVM (Quantum Support Vector Machine)** on kvanttilaskennassa sovellettu tukivektorikone, joka pystyy myös suorittamaan luokittelutehtäviä, mutta sillä on lisäetuna kvanttimekaanisen interferenssin käyttö luokittelun tarkkuuden parantamiseen.

**Klusterointialgoritmit:** Klusterointi on klassinen koneoppimistehtävä, joka perustuu datapisteen ryhmittelyyn niiden samankaltaisuuden perusteella. Klassisia klusterointialgoritmeja ovat esimerkiksi k-means, hierarchical clustering ja DBSCAN. Kvanttiklusterointi soveltaa kvanttilaskennan algoritmeja klusterointiin. Esimerkiksi Quantum k-means ja QHCA (quantum hierarchical clustering) ovat tähän käyttötarkoitukseen kehitettyjä algoritmeja.

Yleisesti ottaen kvanttilaskennan koneoppimisalgoritmit voidaan nähdä laajennuksina tai parannuksina niiden perinteisiin vastineisiin verrattuna. Vaikka niissä on joitakin yhtäläisyyksiä, kvanttimekaanisilla koneoppimisalgoritmeilla on mahdollisuus suorittaa tietyt tehtävät paremmin, koska ne voivat hyödyntää kvanttirinnakkaisuutta, kvanttimekaanista interferenssiä ja muita kvanttimekaanisia ilmiöitä.

Osalle klassisista koneoppimisalgoritmeista voidaan nähdä olevan "kvantti-vastinparinsa"; versio algoritmista, joka toteutetaan eri tavoin, kvanttilaskentaa hyödyntäen. Esimerkkejä osasta tällaisista perinteisistä ja kvanttialgoritmeista esitetään alla olevassa taulukossa.

## Koneoppiminen ja kvanttikoneoppiminen:

	Perinteinen (klassinen) koneoppiminen	Kvanttikoneoppiminen
<b>Ohjaamaton oppiminen</b> (unsupervised learning)		
<b>Klusterointi</b> (clustering)	<ul style="list-style-type: none"> <li>• Hierarchical Clustering</li> <li>• K-Means</li> <li>• SOM (Self-organizing maps)</li> <li>• DBSCAN</li> </ul>	<ul style="list-style-type: none"> <li>• QHCA (Quantum Hierarchical Clustering Algorithm)</li> <li>• Quantum k-Means</li> </ul>
<b>PCA-pääkomponentti-analyysi</b> (principal component analysis)	<ul style="list-style-type: none"> <li>• PCA</li> </ul>	<ul style="list-style-type: none"> <li>• Quantum PCA</li> </ul>
<b>Ohjattu oppiminen</b> (supervised learning)		
Luokittelu (classification)	<ul style="list-style-type: none"> <li>• SVM (Support Vector Machine)</li> <li>• k-NN (K-nearest neighbors)</li> <li>• Naive Bayes</li> </ul>	<ul style="list-style-type: none"> <li>• QSVM (Quantum Support Vector Machine)</li> <li>• QkNN (Quantum k-nearest neighbor)</li> <li>• (QNB) Quantum Naive Bayes, Naive Bayes Networks</li> </ul>
<b>Regression</b>	<ul style="list-style-type: none"> <li>• Linear regression</li> <li>• Logistic Regression</li> </ul>	<ul style="list-style-type: none"> <li>• VQR (Variational Quantum Regressor)</li> </ul>
<b>Vahvistusoppiminen</b> (reinforcement learning)	<ul style="list-style-type: none"> <li>• Monte Carlo</li> <li>• Q-Learning</li> <li>• DQN</li> </ul>	<ul style="list-style-type: none"> <li>• VQC (Variational Quantum Circuit)</li> <li>• PQC (Parametrized Quantum Circuit)</li> <li>• Quantum Boltzmann Machine</li> </ul>
Syväoppiminen (deep learning)		
<b>Neuroverkot</b>	<ul style="list-style-type: none"> <li>• ANN, Artificial Neural Networks</li> </ul>	<ul style="list-style-type: none"> <li>• QNN (Quantum Neural Networks)</li> </ul>
<b>Syvät neuroverkot</b>	<ul style="list-style-type: none"> <li>• DNN, Deep Neural Networks</li> </ul>	<ul style="list-style-type: none"> <li>• QDNN (Quantum Deep Neural Networks)</li> </ul>
<b>Takaisinkytketyvät neuroverkot</b>	<ul style="list-style-type: none"> <li>• RNN, Recurrent Neural Networks</li> </ul>	<ul style="list-style-type: none"> <li>• QRNN (Quantum Recurrent Neural Networks)</li> </ul>
<b>”Kerrokselliset” Konvoluutioneuroverkot</b>	<ul style="list-style-type: none"> <li>• CNN, Convolutional Neural Network</li> </ul>	<ul style="list-style-type: none"> <li>• QCNN (Quantum Convolutional Neural Network)</li> <li>• QNN (Quantum Convolutional Neural Network)</li> </ul>
<b>GAN-verkot</b>	<ul style="list-style-type: none"> <li>• GAN, Generative Adversarial</li> </ul>	<ul style="list-style-type: none"> <li>• QGAN (Quantum Generative Adversarial Network)</li> </ul>

Koska kvanttikoneoppiminen hyödyntää kvanttilaskennan tuomia etuja, on todennäköistä, että se kykenee perinteisen koneoppimisen kannalta samantyyppiseen ongelman ratkaisuun, mutta perinteisiä ratkaisuja tehokkaammin. Tehokkuusetuja voivat olla esimerkiksi nopeampi laskenta-aika, laajempi joukko kerralla saavutettavia tuloksia, isojenkin parametrimäärien hyödyntäminen mallin tarkkuuden parantamisessa tai muulla tavoin paremmat tulokset.

Klassisten ja kvanttikoneoppimisalgoritmien välillä merkittäviä eroja voivat olla esim. seuraavat:

- Datan esitys: perinteiset koneoppimisalgoritmit käsittelevät dataa käyttämällä klassisia bittejä, kun taas kvanttikoneoppimisalgoritmit käyttävät kubitteja. Kubittit voivat esiintyä tilojen superpositiossa, mikä sallii kvanttialgoritmien käsitellä suuria määriä potentiaalisia ratkaisuja samanaikaisesti.
- Rinnakkaiskäsitely: kvanttialgoritmit voivat käsitellä useita potentiaalisia ratkaisuja samanaikaisesti kubittien kvanttiominaisuuksien ansiosta, minkä ansiosta ne voivat suorittaa tiettyjä tehtäviä tehokkaammin kuin klassiset algoritmit.
- Algoritminen kompleksisuus: kvanttikoneoppimisalgoritmeilla on erilainen algoritminen monimutkaisuus kuin klassisilla algoritmeilla. Esimerkiksi joillakin kvanttialgoritmeilla on eksponentiaalinen nopeus verrattuna klassisiin algoritmeihin tiettyjen ongelmien ratkaisemiseksi.
- Kvanttilaitteiston rajoitukset: Vielä tällä hetkellä ns. NISQ- aikakauden kvanttilaitteisto on rajoitettu kubittien lukumäärän ja kvanttioperaatioiden tarkkuuden suhteen, mikä voi rajoittaa kvanttikoneoppimisalgoritmeilla ratkaistavien ongelmien kokoa ja monimutkaisuutta.
- Kvanttialgoritmit pohjaavat todennäköisyyksiin: kvanttialgoritmit ovat luonnostaan todennäköisiä kubittien kvanttitilojen mittaamisen vuoksi.

Alla esimerkkejä kvanttikoneoppimisalgoritmeista, joita on kehitetty ja tutkittu viime vuosina. On hyvä huomioida, että osalle algoritmeista ei ole vakiintuneita suomenkielisiä nimiä edes niiden klassisille versioille, ja vielä harvemmin soveltuvia nimiä kvantti-versioille; tämän vuoksi algoritmeista käytetään pääsääntöisesti englanninkielistä versiota sen nimestä.

- **Quantum Support Vector Machines (QSVM):** kvanttilaskennalla toteutettava versio tukivektorikoneesta (SVM). QSVM on kvanttialgoritmi luokittelutehtäviin, jonka tavoitteena on ratkaista klassisten tukivektorikoneiden mahdollisia suorituskykyrajoitteita kvanttilaskennan avulla. QSVM käyttää "kvanttipiirrekarttaa" (quantum feature map) kartoittaakseen syötetiedot kvanttitilaan, joka sitten käsitellään kvanttietokoneella luokittelutuloksen saamiseksi.
- **Quantum Principal Component Analysis (QPCA):** kvanttilaskennalla toteutettava algoritmi datan sisältämien ulottuvuuksien vähentämiseen (*dimension reduction*), mikä on tärkeä esikäsittelevä vaihe monissa koneoppimistehtävissä. QPCA käyttää kubittien kvanttiominaisuuksia löytääkseen tehokkaasti tietyn tietojoukon merkittävimmät pääkomponentit (*principal components*).
- **Quantum Boltzmann Machine Learning (QBML):** kvanttilaskennalla toteutettava versio klassisesta Boltzmann-koneesta, joka on eräänlainen valvoton oppimisalgoritmi. QBML käyttää kubitteja edustamaan solmuja Boltzmann-koneessa ja näiden kubittien kvanttiominaisuuksia oppimisprosessin suorittamiseksi.
- **Quantum-k-Means-klusterointi:** k-Means on suosittu klusterointialgoritmi klassisessa koneoppimisessa, jonka tarkoituksena on ryhmitellä tietty tietojoukko k klusteriin. Quantum k-Means on kvanttialgoritmi, joka käyttää kubittien kvanttiominaisuuksia nopeuttamaan klusterointiprosessia.
- **Kvanttineuroverkot:** luokka kvanttialgoritmeja, joiden tarkoituksena on jäljitellä klassisten neuroverkkojen käyttäytymistä kubittien avulla. Näitä algoritmeja käytetään tehtäviin, kuten kuvantunnistukseen ja luonnollisen kielen käsittelyyn.

### 5.3.1 Klusterointi kvanttialgoritmeilla (Quantum Clustering)

Kvanttikalauklusterointialgoritmeilla on potentiaalia olla tehokkaampia kuin klassiset klusterointialgoritmit, koska ne pystyvät suorittamaan tiettyjä laskelmia rinnakkain.

Kvanttikalauklusteroinnilla on mahdollisuuksia esimerkiksi seuraavissa sovelluskohteissa:

- Isojen datamäärien käsittelemiseen: perinteistä klusterointia sovellettaessa ja datamäärän kasvaessa, laskenta-aika voi kasvaa eksponentiaalisesti, kun taas kvanttietokoneilla voi olla etua suuren datamäärän käsittelemisessä, koska ne eivät noudata samoja laskennan rajoitteita.

- Monimutkaisten tietojen luokitteluun: kvanttiklusteroinnilla voidaan käsitellä monimutkaisia tietoja, joihin liittyy iso määrä muuttujia ja ominaisuuksia, joita voi olla vaikea käsitellä klassisilla menetelmillä.
- Optimaalisten ratkaisujen löytämiseen: kvanttiklusterointi voi auttaa löytämään optimaalisia ratkaisuja monimutkaisiin ongelmiin, joita ei voida ratkaista tehokkaasti perinteisillä menetelmillä.

Perinteiset klusterointialgoritmit vaativat yleensä useita iteraatioita ratkaisun saavuttamiseen. Jokainen iteraatio sisältää datapisteiden ja klusterien keskusten (*centroids*) välisten etäisyyksien laskemisen, päivittämisen ja datapisteiden määrittämisen uudelleen lähimpään klusteriin. Tämä prosessi voi olla laskennallisesti raskas etenkin suurille tietojoukoille tai suuriulotteisille tiedoille; tämä voi rajoittaa esimerkiksi klusteroinnin avulla ryhmiteltävän tiedon muuttujien määrä tai monipuolisuutta, ja siten myös rajoittaa riittävän monimutkaisten klusterointiongelmien ratkaisuja.

Kvanttiklusterointialgoritmit sitä vastoin voivat hyödyntää kubittien kvanttiominaisuuksia suorittaakseen laskelmat rinnakkain. Esimerkiksi kvantti-k-Means-klusterointi voi käyttää kubittien kvanttiominaisuuksia etäisyyksien laskemiseen datapisteiden ja klusterien keskusten välillä samanaikaisesti, mikä voi merkittävästi nopeuttaa klusterointiprosessia.

Lisäksi kvanttiklusterointialgoritmit voivat hyödyntää kvanttiyhdenmukaisuutta optimaalisen klusterointiratkaisun etsimiseksi tehokkaammin. Tämä on erityisen hyödyllistä, kun etsitään optimaalista klustereiden määrää tai optimaalista klusterointikonfiguraatiota tietylle ryhmiteltävälle datalle.

### 5.3.2 Regressio kvanttialgoritmeilla tai kvanttilaskentaa hyödyntäen (Quantum Regression)

Klassiset ja kvanttiiregressioalgoritmit ovat käytössä regressio-ongelmien ratkaisemiseen, joissa tavoitteena on ennustaa jatkuva lähtömuuttuja yhden tai useamman syötemuuttujan perusteella. Näiden kahden tyyppisten algoritmien välillä on joitain yhtäläisyyksiä ja eroja.

#### Yhtäläisyyksiä:

- Sekä perinteiset että kvanttiiregressioalgoritmit pyrkivät sovittamaan mallin annettuun opetusaineistoon uusien datapisteiden ennustamiseksi.

- Sekä perinteiset että kvanttialgoritmit voivat käyttää erilaisia matemaattisia tekniikoita mallin sovittamiseen, kuten lineaarista regressiota, polynomiregressiota tai neuroverkkoja.
- Molempiin algoritmeihin liittyy häviöfunktio (loss function), jotta voidaan arvioida, kuinka hyvin malli sopii aineistoon.

#### Eroja:

- Kvanttiregressioalgoritmit käyttävät kvanttilaskennan periaatteita ja ne suoritetaan kvanttietokoneilla, kun taas klassiset regressioalgoritmit suoritetaan klassisilla tietokoneilla. Niitä ei siis voi suorittaa ristikkäin eri periaatetta noudattavilla koneilla.
- Kvanttiregressioalgoritmit voivat hyödyntää kvanttirinnakkaisuutta, lomittumista ja kvanttimekaanista interferenssiä, jotka voivat tarjota nopeampaa laskentaa ja parannettua tarkkuutta verrattuna klassisiin algoritmeihin.
- Kvanttiregressioalgoritmit vaativat usein suuren määrän kubitteja ja korkean taason kvanttikoherenssia, mikä voi olla haasteellista toteuttaa käytännössä esim. nykyisen NISQ- aikakauden kvanttietokoneilla.
- Kvanttiregressioalgoritmien toteuttaminen vaatii yleensä ohjelmistollisen kvanttipiirin (*quantum circuit*) suunnittelua, mikä on perustavanlaatuisesti erilainen lähestymistapa mitä käytetään perinteisissä regressioalgoritmeissa. Tämä voi tehdä regressiota soveltavien kvanttikoneoppimiskäytännöiden kehittämisestä vaikeampaa tai ainakin merkittävästi erilaista perinteisiin ratkaisuihin verrattuna.

Klassisissa regressioalgoritmeissa mallin parametrit estimoidaan tyypillisesti minimoimalla häviöfunktio, joka mittaa ennustettujen arvojen ja todellisten arvojen välistä eroa. Tämä edellyttää häviöfunktion osittaisten derivaattojen laskemista kunkin parametrin osalta, mikä voi olla laskennallisesti tehotonta etenkin ison määrän ulottuvuuksia sisältäville malleille tai suurille tietojoukoille.

Kvanttiregressio-algoritmit puolestaan voivat arvioida samanaikaisesti useita vaihtoehtoisia malleja ja etsiä optimaalisia parametreja. Esimerkiksi kvanttilineariset regressioalgoritmit voivat käyttää kvanttimatriisin inversiotekniikoita lineaarisen regressiomallin optimaalisten kertoimien löytämiseen yhdellä kertaa, mikä on paljon nopeampaa kuin klassiset menetelmät, jotka vaativat useita iteraatioita tämän tekemiseen. Lisäksi kvanttikoneoppimisalgoritmit voivat tarjota paremman tarkkuuden kuin klassiset algoritmit tietyntyyppisissä ongelmassa. Esimerkiksi kvanttikoneoppimisalgoritmit voivat



käyttää kvantti-Fourier-muunnoksia suorittaakseen tiettyjä operaatioita tehokkaammin, mikä voi parantaa regressiomallin tarkkuutta.

### 5.3.3 Kvanttineuroverkot (Quantum Neural Networks)

Kvanttineuroverkot voivat tuoda etuja, jolla ne ovat tehokkaampia kuin perinteiset neuroverkot, koska ne pystyvät hyödyntämään rinnakkaisuutta laskelmien suorittamiseksi nopeammin ja mahdollisesti tarkemmin.

Klassisissa neuroverkoissa laskennat suoritetaan klassisilla biteillä. Neuroverkon painot ja poikkeamat päivitetään iteratiivisesti ns. gradienttipohjaisilla optimointialgoritmeilla. Nämä optimointialgoritmit voivat olla laskennallisesti raskaita, erityisesti laajoilla neuroverkoilla, joissa on monia kerroksia ja runsaasti parametreja. Kerrosten ja parametrien lukumäärä olisi kuitenkin tapa mahdollistaa neuroverkon kyky monimutkaisempaan tietojenkäsittelyyn – esim. kuvaamaan erilaisia ilmiöitä kattavasti ja korkealla tarkkuustasolla.

Kvanttineuroverkot voivat käyttää kvanttilaskennan ominaisuuksia laskutoimitusten suorittamiseen rinnakkain, mikä voi nopeuttaa esimerkiksi neuroverkon opettamiseen tarvittavia prosesseja. Esimerkiksi kvanttineuroverkot voivat käyttää rinnakkaisuutta arvioidakseen samanaikaisesti useita vaihtoehtoisia malleja ja löytääkseen verkoille optimaaliset parametrit. Tämä voi mahdollisesti vähentää iteraatioiden määrää, joita tarvitaan neuroverkon kouluttamiseen.

Lisäksi kvanttineuroverkot voivat mahdollisesti tarjota paremman tarkkuuden kuin klassiset neuroverkot tietyntyyppisissä ongelmissa. Esimerkiksi kvanttineuroverkot voivat käyttää kvanttilaskentaan sopivia Fourier-muunnoksia suorittaakseen tiettyjä toimintoja tehokkaammin, mikä voi parantaa neuroverkon tarkkuutta.

#### Lähteitä ja lisätietoja:

Amin, et al. 2016. Quantum Boltzmann Machine. <https://arxiv.org/abs/1601.02036>

Beer, et al. 2020. Training deep quantum neural networks. Nature. <https://www.nature.com/articles/s41467-020-14454-2>

Broughton, et al. 2021. TensorFlow Quantum: A Software Framework for Quantum Machine Learning. <https://arxiv.org/abs/2003.02989>

Cong, I., Choi, S., Lukin, M. 2018. Quantum Convolutional Neural Networks. <https://arxiv.org/abs/1810.03787>

IBM Research. The power of quantum neural networks. <https://research.ibm.com/blog/quantum-neural-network-power>

Lloyd, S., Mohseni, M., Rebentrost, P. 2014. Nature physics. Quantum principal component analysis. <https://www.nature.com/articles/nphys3029>

Lloyd, S., Mohseni, M., Rebentrost, P. 2013. Quantum algorithms for supervised and unsupervised machine learning. <https://arxiv.org/abs/1307.0411>

Pennylane. 2022. What is Quantum Machine Learning. <https://pennylane.ai/qml/whatisqml.html>

Qiskit. Quantum-enhanced Support Vector Machine (QSVM). [https://qiskit.org/documentation/stable/0.24/tutorials/machine\\_learning/01\\_qsvm\\_classification.html](https://qiskit.org/documentation/stable/0.24/tutorials/machine_learning/01_qsvm_classification.html)

Qiskit. Quantum Kernel Machine Learning: Clustering. [https://qiskit.org/ecosystem/machine-learning/tutorials/03\\_quantum\\_kernel.html#3.-Clustering](https://qiskit.org/ecosystem/machine-learning/tutorials/03_quantum_kernel.html#3.-Clustering)

Qiskit. Quantum Neural Networks. [https://qiskit.org/ecosystem/machine-learning/tutorials/01\\_neural\\_networks.html](https://qiskit.org/ecosystem/machine-learning/tutorials/01_neural_networks.html)

TensorFlow Quantum. Quantum Machine Learning Concepts. <https://www.tensorflow.org/quantum/concepts>

Xanadu. Quantum Machine Learning 1.0. 2018.. <https://www.xanadu.ai/blog/quantum-machine-learning-1-0>

## 5.4 Ohjelmistoja, kirjastoja, kehitystyökaluja ja – ympäristöjä

Kvanttikoneoppimista hyödyntäviä kirjastoja löytyy useita eri vaihtoehtoja ja hankkeen aikana kehitystyö on vaikuttanut aktiiviselta. Voi jopa olettaa, että kehitystyön nopeus ja laajuus kasvaa entisestään, vaikkakin kaikkia kvanttikoneoppimisen tavoitteita ei laajassa mittakaavassa voi hyödyntää vielä kaikilla saatavilla olevilla kvanttietokoneilla. Tämä ei kuitenkaan estä kvanttikoneoppimisen tutkimista käytännön soveltamisen ta- salla; erilaisia tehtäviä voidaan suorittaa simulaattoreilla sekä joitakin tehtäviä myös aidoilla kvanttietokoneilla.

Esimerkkejä kvanttikoneoppimista tukevista kirjastoista:

- Qiskit Machine Learning: <https://qiskit.org/ecosystem/machine-learning/>
- Microsoft Quantum Machine Learning: <https://learn.microsoft.com/en-us/azure/quantum/overview-quantum-machine-learning>
- TensorFlow Quantum; <https://www.tensorflow.org/quantum>
- Xanadu PennyLane: <https://www.xanadu.ai/products/pennylane/>

Edellä mainittujen kirjastojen tarjonnasta löytyy kvanttikoneoppimisalgoritmeja tyypillisimpiin tehtäviin (alla listattu esimerkkeinä vain osa mahdollisista)

**Classification**–luokittelutehtävät

- Qiskit: QSVM (*Quantum Support Vector Machine*), VQC (*Variational Quantum Classifier*)
- Microsoft QML: **Quantum Sequential Classifier**
- TensorFlowQuantum: luokittelu QNN (Quantum Neural Network) –verkon avulla
- PennyLane: **Variational Classifier**

**Clustering** – datan ryhmittelytehtävät

- Qiskit: Clustering

**Regressio** – esim. ennusteet

- Qiskit: Regressio kvanttineuroverkoilla (EstimatorQNN, Variational Quantum Regressor)

## Reinforcement learning – vahvistusoppiminen

- TensorFlow Quantum: [Parametrized Quantum Circuits for Reinforcement Learning](#)

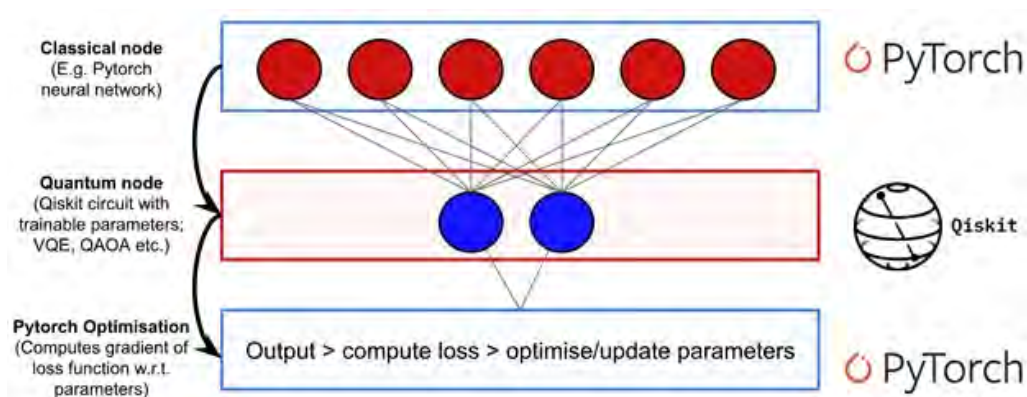
## Neuroverkot

- Qiskit: [QNN](#) (Quantum Neural Networks)
- TensorFlow Quantum: [QCNN](#) (Quantum Convolutional Neural Network)
- PennyLane: [Quantum Convolutional Neural Networks, QGAN](#) (Quantum Generative Adversarial Network)

Osa kvanttikoneoppimista tukevista kirjastoista keskittyy puhtaammin kvanttilaskennan soveltamiseen koneoppimisessa ja osa taas mahdollistaa ns. quantum-classical hybridien soveltamisen; osa koneoppimiseen liittyvistä toiminnoista suoritetaan perinteisillä tietokoneilla ja tarvittavat osat kvanttietokoneilla. Tällä tavoin voidaan pyrkiä saavuttamaan parhaat edut molemmista laskennan tyypeistä.

- Hybrid quantum-classical Neural Networks with PyTorch and Qiskit: <https://learn.qiskit.org/course/ch-applications/hybrid-quantum-classical-neural-networks-with-pytorch-and-qiskit>
- TensorFlow Quantum: <https://www.tensorflow.org/quantum>

Esimerkiksi yllä mainittu kvanttikoneoppimista tukeva Qiskit- ja perinteistä koneoppimista tukeva PyTorch – kirjastojen kyvyt luoda ja käsitellä neuroverkkoja on yhdistettävissä quantum-classical- tyyppiseksi hybridiksi, jossa verkko jakautuu sekä klassisiin että kvanttilaskennan avulla luotaviin ja käsiteltäviin rakenteisiin.



Kuva 38. Qiskit-PyTorch – kvantti-klassinen-hybridi (Qiskit 2022. Hybrid quantum-classical Neural Networks with PyTorch and Qiskit.)

## 5.5 Quantum Machine Learning as a Service – kvanttikoneoppimISRatkaisujen toteuttaminen QCaaS- pilvipalveluissa

Kvanttilaskentaa tarjoavat pilvipalvelut mahdollistavat yleensä erilaisten ohjelmakirjastojen käytön kvanttiohjelmien toteuttamisessa ja suorittamisessa. Käytännössä tämä tarkoittaa myös kvanttikoneoppimista tukevien kirjastojen käyttöä pilvipalveluissa ja parhaimmillaan niillä toteutettujen ohjelmistojen suorittamista pilvipalvelun sisältämällä resursseilla – esim. oikeilla kvanttietokoneilla.

Esimerkiksi useampi pilvipalvelu (mm. IBM Quantum, Amazon Braket, Azure Quantum) tukee Qiskit- kirjaston käyttöä. Qiskit sisältää ominaisuuksia sekä kvanttikoneoppimisen että quantum-classical -hybridejä tukevan koneoppimisen ratkaisujen toteuttamiseen myös pilvipalveluina tarjottavalla kvanttilaskentakapasiteetilla suoritettavaksi.

Toki vielä toistaiseksi pilvipalveluista saatavien koneiden kyky laskea riittävän laajoja kvanttikoneoppimisen algoritmeja sisältää haasteita. Käytännössä, vaikka ohjelmakirjasto mahdollistaisi esimerkiksi kvanttineuroverkkojen toteuttamisen, ei niiden onnistunut tai luotettava suorittaminen vielä toistaiseksi toimi kaikilla kvanttietokoneilla. Toinen huomioitava näkökulma on tietenkin pilvipalvelusta hankittavan kvanttilaskentaajan kustannukset, joka voi jossain tilanteissa nousta esteeksi kehitys- ja testausvaiheissa. Tämän päivän rajoituksista huolimatta kuitenkin periaate ja kehityksen suunta on lupaava pilvipalveluiden kautta saavutettavien kvanttietokoneiden jatkuvasti kehittyessä sekä kubittien määrän ja laadun (mm. fideliteetti, porttioperaatioiden kesto). Esimerkiksi Microsoft Azuren sisältämää Quantum- palveluja voitaisiin hyödyntää kvanttikoneoppimisen toteutuksissa seuraavasti:

Azure tarjoaa palveluita ja työkaluja kvanttikoneoppimisen (quantum machine learning) toteuttamiseen. Tässä on joitakin esimerkkejä Azuren tarjoamista kvanttikoneoppimiseen liittyvistä palveluista ja työkaluista:

1. Quantum Development Kit (QDK)- kehitysympäristö tarjoaa työkaluja kvanttialgoritmien ja kvanttiohjelmien kehittämiseen.

2. Azure Quantum- on Azure pilvipalvelun osa, joka tarjoaa pääsyn kvanttikoneisiin ja mahdollistaa myös kvantti- ja klassisten algoritmien yhdistämisen. Se sisältää myös työkaluja kvanttikoneoppimisen ongelmien ratkaisemiseen.
3. Ratkaisut voidaan toteuttaa esimerkiksi Q#- tai Python - ohjelmointikieliä käyttäen.
4. Microsoft Quantum Machine Learning on osa Microsoftin kvanttilaskenta-kirjastoja, joka tarjoaa algoritmeja kvanttikoneoppimisen soveltamiseen.

### Lisätietoja:

Microsoft Introduction to Quantum Machine Learning. <https://learn.microsoft.com/en-us/azure/quantum/user-guide/libraries/machine-learning/intro>

Microsoft Quantum Machine Learning NuGet library: <https://www.nuget.org/packages/Microsoft.Quantum.MachineLearning>

AWS:n **Amazon Braket** mahdollistaa kvanttikoneoppimisen ratkaisujen toteuttamista pilvipalveluissa varsin pitkälle edellä esitetyn Microsoft Azuren tapaan. Amazon Braket-palvelun kanssa hyödynnettävä **Amazon Braket SDK** tukee natiivisti **Pennylane**- kvanttikoneoppimiskirjastoa, ja sen kautta mm klassista koneoppimista mahdollistavia. PyTorch- ja TensorFlow- kirjastoja.

Amazon Braket:iin on saatavilla **Qiskit-Braket-provider**, jota voidaan hyödyntää Qiskit-kirjastoa Amazon Braket SDK:lla kehitettävien sovellusten kanssa.

Qiskit-Braket -provider voidaan asentaa Amazon Braket SDK:lla tehtyyn projektiin perinteiseen tapaan pip- työkalulla:

```
pip install qiskit_braket_provider
```

Qiskit-Braket-provider mahdollistaa Qiskitillä tehtyjen ohjelmien suorittamisen Amazon Braketin tarjoamilla kvanttietkoneilla.

## Lisätietoja:

Amazon Braket- ominaisuuksia: <https://aws.amazon.com/braket/features/>

Introducing the Qiskit provider for Amazon Braket <https://aws.amazon.com/blogs/quantum-computing/introducing-the-qiskit-provider-for-amazon-braket/>

IBM Quantum –pilvipalvelu tukee natiivisti Qiskit- kirjastoa ja sen quantum machine learning- kyvykkyksiä ja mahdollisuuksia myös quantum-classical –hybridien toteuttamiseen.

Googlen tukeman TensorFlow Quantum- kehitysympäristön soveltamista **Google AI:n Quantum Computing Serviceissä** ei hankkeen aikana ollut harmillisesti mahdollisuutta tutkia tarkemmin. Oletettavasti, TensorFlow:n ollessa yksi laajimpia koneoppimiseratkaisuja ja sen konseptien tukeessa (oletettavasti natiivisti) myös quantum-classical- hybridiratkaisujen toteuttamista koneoppimisessa, on se yksi teknologia, jonka kehittämistä kannattaa tarkkaan seurata.

### 5.5.1 Lopuksi

Vaikka kvanttilaskentaa tukevat pilvipalvelut poikkeavat monin teknisin ja toiminnallisista osin toisistaan, on niissä kuitenkin tunnistettavissa samantyyppisiä käytön periaatteita sekä etenkin potentiaalia mahdollistaa kvanttikoneoppimiseratkaisujen toteuttaminen muun kvanttiohjelmoinnin ja jopa muun, perinteisen sovelluskehityksen käyttämään tapaan. Lisäksi on oletettavissa, että esim. koneoppimisessa kvantti- ja klassisten hybridien toteuttamista ja hallintaa tukevia ominaisuuksia syntyy jatkuvasti lisää.

Yksi ominaisuus, joiden tarve voi korostua etenkin rakennettaessa hybridi- mallin mukaisia kvanttikoneoppimiseratkaisuja, on sovelluksen arkkitehtuurin määrittely ja tehtävien delegointi klassisen ja kvanttilaskennan resurssien välillä, jota esim. D-Waven hybrid solvers- periaate toteuttaa tätä ehkä pisimmälle vietyinä

Toinen tarve voi liittyä toimintoihin, joissa muunnetaan (encoding, decoding) klassista dataa kvantti-informaation muotoon ja toisin päin. Tuki näiden operaatioiden tekemiseen, mielellään mahdollisimman pitkälle automatisoitavana, on oletettavasti tärkeää.

Myös joustavat API- rajapinnat, joilla esimerkiksi perinteisistä sovelluksista voisi kutsua kvanttilaskentaresursseja (alirutiinit, laajemmat algoritmien ja ohjelmien kokonaisuu- det) samoin periaattein kuin muissa nykyaikaisissa verkkopalveluissa, voi olla yksi kvanttikoneoppimISRatkaisujen hyödyntämistä ja jälleenkäytettävyyttä lisäävä mahdol- lisuus.

Vaikka kehittämistarpeita ja ratkottavia pullonkauloja on vielä runsaasti, ovat pilvipal- velut, kirjastot, sovelluskehittimet ja tiettyihin tehtäviin liittyen myös itse kvanttietoko- neet saavuttaneet jo nyt kypsyyden, jossa voidaan tehdä jo ainakin alustavia sovellu- tuksia. Teknologioiden kehittymistä odotellen näiden jo olemassa olevien palveluiden ja teknologioiden saatavuuteen ja hyödyntämisen periaatteisiin on erittäin hyödyllistä tu- tustua. Samalla kehitty näkemystä ja osaamista eri sovelluskohteiden arviointia var- ten – ja samalla rakentaa kyvykkyyttä kvanttikoneoppimista hyödyntäviä käytännön sovellutuksia.