Sami Sihvonen

Vahva tunnistautuminen MFA- ja Web3-teknologioilla sovelluskehi-tyksessä

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Tieto- ja viestintätekniikka

Insinöörityö

28.05.2024

Tiivistelmä

Tekijä(t):	Sami Sihvonen
Otsikko:	Vahva tunnistautuminen MFA- ja Web3-teknologioilla sovelluskehityksessä
Sivumäärä:	31
Aika:	28.05.2024
Tutkinto:	Insinööri (AMK)
Tutkinto-ohjelma:	Tieto- ja viestintätekniikka
Suuntautumisvaihtoehto:	Mobile Solutions
Ohjaaja(t):	Lehtori Janne Salonen

Tämä insinöörityö keskittyy monivaiheisen ja Web3-tunnistautumisen tutkimiseen ja implementointiin web-sovelluksessa.

Opinnäytetyö pyrki selvittämään, miten Web3-tunnistautuminen, joka hyödyntää lohkoketjujen hajautettua rakennetta, voi parantaa tietoturvaa ja käyttäjäkokemusta web-sovelluksessa. Tutkimuksen tarkoituksena oli kehittää ymmärrystä siitä, miten monivaiheinen ja hajautettu todennus toimii ja miten ne voi tarjota paremman turvallisuuden ja käyttäjäystävällisyyden verrattuna perinteisiin tunnistautumismenetelmiin.

Opinnäytetyön tulokset osoittivat, että Web3-tunnistautuminen voi olla merkittävä parannus nykyiseen tunnistautumisprosessiin, tarjoten käyttäjille lisäturvaa ja vähentäen riippuvuutta kolmannen osapuolen todennuspalveluista. Lisäksi tulokset korostivat monivaiheisen todennuksen olevan yksi turvallisimmista vaihtoehdoista nykyisessä internetissä.

Tutkielman yhteydessä toteutin sovelluksen, joka mahdollistaa tunnistautumisen sekä OAuth 2.0- että Web3.0-teknologioilla. Tutkimuksen perusteella havaittiin Web3-tunnistautumisen merkitys ja sen helppo integroituvuus käyttäen valmiita JavaScript-kirjastoja.

Avainsanat: Oauth2, Web3, tunnistautuminen

Abstract

Author(s):	Sami Sihvonen
Title:	Strong Authentication with MFA and Web3 technologies in software development
Number of Pages:	31
Date:	28.05.2024
Degree:	Bachelor of Engineering
Degree Programme:	Information and Communication Technology
Professional Major:	Mobile Solutions
Supervisor:	Lecturer Janne Salonen

The engineering work focused on researching and implementing multifactor- and Web3-authentication in a web application.

The thesis sought to find out how Web3 authentication, which utilizes the decentralized structure of blockchains, can improve data security and user experience in a web application. The purpose of the research was to develop an understanding of how multi-phase and distributed authentication works and how they can offer better security and user-friendliness compared to traditional identification methods.

The results of the thesis showed that Web3 authentication can be a significant improvement to the current authentication process, offering users additional security and reducing dependence on third-party authentication services. In addition, the results highlighted that multi-step authentication is one of the most secure options in today's internet.

At the end of the project, a user-friendly application was created that enables authentication with both OAuth 2.0 and Web3.0 technologies. Based on the research, the importance of Web3 identification and its easy integration using ready-made JavaScript libraries was found.

Keywords: Oauth2, Web3, Web3.0, authentication

Sisällys

Lyhenteet ja käsitteet

2FA: *Two Factor Authentication.* Kaksivaiheinen todennus, jossa käytetään lisäksi toista tekijää, kuten mobiililaitetta.

WebAuthn: *Web Authentication.* Protokolla tarjoaa vahvan tunnistautumisen verkossa ilman salasanoja.

MagicLink: Menetelmä, jossa kirjautuminen tapahtuu erityisen linkin avulla ilman salasanaa.

MFA: *Multifactor Authentication.* Monivaiheinen todennus, jossa käytetään useita tekijöitä käyttäjän henkilöllisyyden varmentamiseksi.

OAuth: *Open Authrization.* Protokolla, joka mahdollistaa sovellusten valtuuttamisen käyttämään rajattua tietomäärää ilman salasanojen jakamista.

OTP: *One-Time-Password.* Kertakäyttöinen salasana, joka lähetetään käyttäjälle ja on voimassa tietyn ajan.

SFA/1FA: *Single-Factor-Authentication.* Yksivaiheinen tunnistautuminen, jossa käytetään käyttäjätunnusta ja salasanaa.

TPA: *Third-party Application.* Kolmannen osapuolen todennus, joka mahdollistaa kirjautumisen käyttämällä muun organisaation tarjoamia tunnistautumistietoja.

GPS *Global Positioning System* on maailmanlaajuinen satelliittinavigointijärjestelmä, joka mahdollistaa sijainnin määrittämisen missä tahansa maapallolla

1 Johdanto

Sähköisen asioinnin yleistyessä digitaalinen tunnistautuminen on tullut olennaiseksi osaksi arkea. Sen kehityksessä on painotettu käyttäjäystävällisyyttä ja turvallisuutta vuosi vuodelta. Perinteinen yksivaiheinen salasanasuojaus ei enää riitä tietovuotojen ja tietomurtojen yleistyessä. Turvallisuuden parantamiseksi tunnistautumispalveluissa on kehitetty useita uusia menetelmiä, joista monivaiheinen tunnistautuminen on yksi turvallisimmista vaihtoehdoista.

Monivaiheinen tunnistautuminen toimii ylimääräisenä suojakerroksena, joka vaikeuttaa hakkerointia, vaikka salasana olisi varastettu. Useat yritykset käyttävät monivaiheista tunnistautumista käyttäjien identiteetin vahvistamiseen ja nopean ja kätevän pääsyn tarjoamiseen valtuutetuille käyttäjille. Väärinkäytöllä voi olla vakavia reaalimaailman seurauksia, kuten liiketoiminnan häiriöitä, yksityisyyden suojan menetyksiä tai taloudellisia ongelmia.

Viimeisen vuosikymmenen aikana on tietoturva-alalla tapahtunut merkittävää kehitystä erityisesti todennus- ja tunnistautumismenetelmissä. Yksi yleisimpiä todennusmenetelmä on nykyään OAuth 2.0 -pohjainen todennus. Tämä menetelmä perustuu kolmannen osapuolen todennuspalvelun tarjoajaan, joka hallitsee täysin käyttäjän tietoja, joita se voi suodattaa tai muokata mielensä mukaan. OAuth 2.0 -tunnistautumisen lisäksi voidaan hyödyntää myös muita todennustekijöitä, kuten esimerkiksi sormenjälkitunnistautumista puhelimella.

Lohkoketju ja hajauttaminen ovat herättäneet paljon kiinnostusta viime vuosina ja hajautettua verkkoa pidetään seuraavana merkittävänä parannuksena maailmanlaajuisessa verkossa. Web3-todennus, joka tunnetaan myös nimellä hajautettu todennus, mahdollistaa käyttäjien turvallisen ja hajautetun todennuksen verkossa. Todennus tapahtuu selaimessa kryptolompakon avulla, mikä poistaa tarpeen erillisille kirjautumisprosesseille palveluissa, jotka hyödyntävät web3-tunnistautumista. Web3 tunnistautumisessa tämä todennus perustuu niihin tietueisiin, mitä käyttäjä on itsestään antanut. Tämä tunnistautumismenetelmä saattaa olla suurin muutos tähän mennessä ja tuo käyttäjille lisäturvaa.

2 Tunnistautuminen

Tunnistautuminen tarkoittaa käyttäjän identiteetin vahvistamista tietyn menetelmän avulla. Perinteisesti tämä on toteutettu fyysisten asiakirjojen avulla, kuten passin, henkilökortin tai ajokortin avulla. Sähköisten palveluiden yleistymisen myötä henkilöllisyyden todentaminen on siirtynyt digitaaliseen ympäristöön, jossa identiteetti voidaan todentaa useilla eri tavoilla. Todennuksen jälkeen käyttäjän valtuudet päästä tiettyyn resurssiin saatetaan tarkistaa.

Kuluttajien tunnistautumisteknologiat ovat kokeneet merkittävää kehitystä tietoturvariskien minimoimiseksi. Salasanoille on asetettu vaatimuksia, kuten vähimmäispituus sekä lisävaatimukset, kuten isot ja pienet kirjaimet sekä erikoismerkit, ja niiden turvallisuutta on parannettu kehittämällä suojausmenetelmiä, jotka vaikeuttavat niiden murtamista.

Digitalisaation vaikutus ulottuu nykyään modernin yhteiskunnan kaikkiin osa-alueisiin. Turvallisuuden varmistamisessa keskeisessä roolissa on todennus. Todennus on olennainen osa digitaalisesta ympäristöä, kuten verkkomaksuissa, viestinnässä ja käyttöoikeuksien hallinnassa. (Aleksandr Ometov ym, 17)

3 Todentaminen

Todennus on prosessi, jossa käyttäjän antamat tiedot vahvistetaan eri todennusmekanismeja käyttämällä. (Farik Mohammed ym., 6)

Todennusmekanismeja on kolme päätyyppiä:

1. *Jotain, mitä tiedät*
    - esimerkiksi salasana tai PIN-koodi
2. *Jotain mitä sinulla on*

- esimerkiksi pankkikortti, älykortti, fyysinen koodigeneraattori, älypuhelinsovellus tai vastaava

3. *Jotain mitä olet*

    - Esimerkiksi henkilön biometriset merkit, kuten ääni, sormenjäljet, kasvojen piirteet, sykkeen, painon tai muiden henkilökohtaisten ominaisuuksien mukaan

    (Johnson Caroline, 8)

Näiden kolmen päätyypin lisäksi käyttäjä voidaan todentaa sijainnin (IP-osoitteen tai puhelimen GPS-sijainnin) mukaan. Sijaintiin perustuvan todennuksen turvallisuustaso on suhteellisen alhainen, joten sitä käytetään yleensä yhdessä muiden todennustapojen kanssa. (Mizrachi Aviad, 13)

3.1 Salasanalla tunnistautuminen

Salasanapohjaisessa todennusmenetelmässä pääsy tiettyyn resurssiin myönnetään ainoastaan, jos käyttäjätunnus ja salasana vastaavat järjestelmässä olevia tietoja. Käyttäjätunnuksen ja salasanan avulla tunnistautuminen on yleisesti käytössä oleva todennusmenetelmä, jolla rajataan käyttäjän pääsyä eri resursseihin. Tämä todennusmenetelmä saattaa kuitenkin olla altis turvallisuusriskille riippuen salasanan pituudesta ja sen ominaisuuksista. (Vapen Anna, 29)

Salasanoja voidaan kuitenkin suojata erilaisin keinoin, kuten hajautusmenetelmillä. Salasanan hajautuksessa syöte muunnetaan matemaattisen algoritmin avulla hajautusarvoksi, joka ei ole helposti luettavassa muodossa. Hajautusarvon laskeminen on helppoa ja käytännöllistä, mutta alkuperäisen syötteen uudelleen luominen on vaikeaa tai mahdotonta, jos vain hajautusarvo tiedetään. (Arias Dan, 2)

3.2 Varmennepohjainen todentaminen

Varmennepohjaisessa todennuksessa käyttäjän henkilöllisyys todennetaan digitaalisen sertifikaatin avulla. Todennus tapahtuu siten, että varmenteen myöntäjän yksityisen avaimen avulla luodaan digitaalinen allekirjoitus, joka perustuu varmenteen myöntäjän julkiseen avaimeen. Digitaalisen allekirjoituksen avulla varmennetaan käyttäjä, tätä prosessia kutsutaan myös digitaaliseksi varmenteeksi. (Ping Identity, 19)

3.3 Biometrinen tunnistaminen

Biometrisessä tunnistamisessa käyttäjä todennetaan biologisten ominaisuuksiensa perusteella. Käyttäjästä voidaan tallentaa biometristä dataa sormenjäljistä, kasvojen tai iiriksen muodoista, joita verrataan aiemmin tallennettuihin tietoihin. Biometrinen tunnistus on käyttäjäystävällinen ja luonnostaan turvallinen tunnistautumiskeino. (Luoma Ossi, 11)

3.4 Token-pohjainen tunnistautuminen

Token-pohjainen todennus on yleisesti käytetty todennusprotokolla, jossa käyttäjä todentaa itsensä kerran ja vastaanottaa henkilöllisyytensä vahvistavan (tokenin) tunnuksen. Niin kauan kuin token on voimassa, käyttäjällä on pääsy verkkosivustolle tai sovellukseen kirjautumatta uudelleen. Token-tunnuksen vanhentuessa käyttäjä tarvitsee uuden tokenin kirjautuakseen uudelleen palveluun. (Mizrachi Aviad, 13)

3.5 Kertakirjautuminen Single-Sing-on (SSO)

Single-Sign-On (SSO) on tapa todentaa käyttäjät kirjautumaan useisiin sovelluksiin yhdellä kirjautumiskerralla. Todennus tapahtuu esimerkiksi kirjautumalla Googlen palveluun, joka tarjoaa pääsyn sovelluksen sisäisiin palveluihin esimerkiksi Gmail:iin tai ulkoisen palvelutarjoajan sovelluksiin, jotka käyttävät SSO-palvelua. (Single sign on, 21)

3.6 Yksivaiheinen tunnistautuminen

Yksivaiheisessa tunnistautumisessa (1FA tai SFA) käytetään ainoastaan yhtä todennustekijää. Tämä prosessi käyttää yleensä yhdistelmänä käyttäjätunnusta ja salasanaa. Menetelmänä tämä on käyttäjälle helppokäyttöinen, mutta se on haavoittuvainen erilaisille turvallisuusuhkille, sillä se nojaa ainoastaan yhteen todennustekijään, joka ei ole riittävä suojaamaan käyttäjän tietoja nykypäivän monimutkaisia kyberuhkia vastaan. (Cisa.gov, 4)

3.7 Kaksivaiheinen tunnistautuminen

Kaksivaiheinen tunnistautuminen (2FA), joka rajoittuu kahteen eri todennustekijään, on todennusmenetelmä, jossa käyttäjältä vaaditaan käyttäjätunnuksen ja salasanan lisäksi ja kaksivaiheisen todennuksen lisäkerrosta, kuten turva-avainta, älypuhelinsovellusta tai vahvistuskoodia joko sähköpostitse, tekstiviestinä tai puhelimitse. (F-secure, 7)

4 Monivaiheinen tunnistautuminen

4.1 Monivaiheisen todennuksen todennustekijät

Monivaiheinen todennus (MFA) on vahva todennusmenetelmä, jossa käyttäjän on todennettava henkilöllisyytensä useammalla kuin yhdellä todennusmenetelmällä kirjautuessaan palveluun tai sovellukseen.

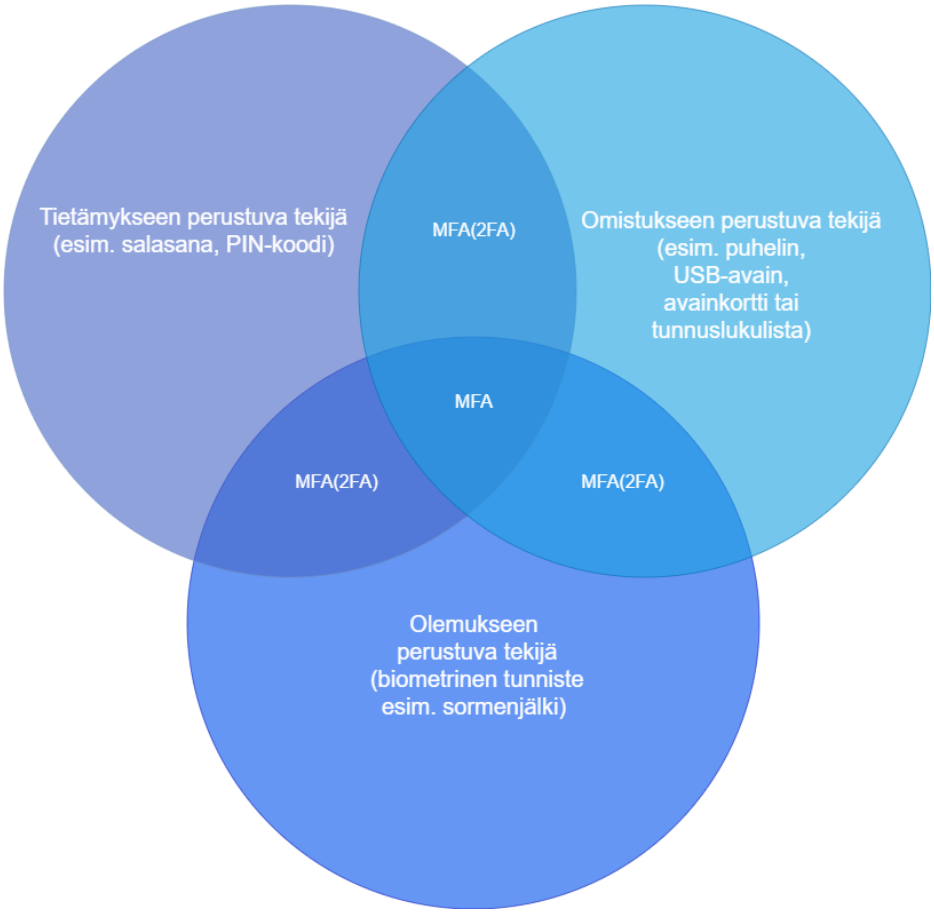
Kaksi- ja monivaiheinen tunnistautuminen koostuu yleensä seuraavista todennustekijöistä tai jostakin niiden yhdistelmistä:

- Jotain minkä tiedät – salasanat, PIN-koodit, turvakysymykset
- Jotain mitä omistat – varmenteet, älykortit, OTP-salasanat, tokenit
- Jotain mitä olet – biometriset tiedot, kuten sormenjälki.

- Jotain missä olet – IP-osoite, maantieteellinen sijainti

- Jotain mitä teet – Käyttäytymisprofilointi, näppäinpainallusten tai hiiren dynamiikka, kävelyanalyysi

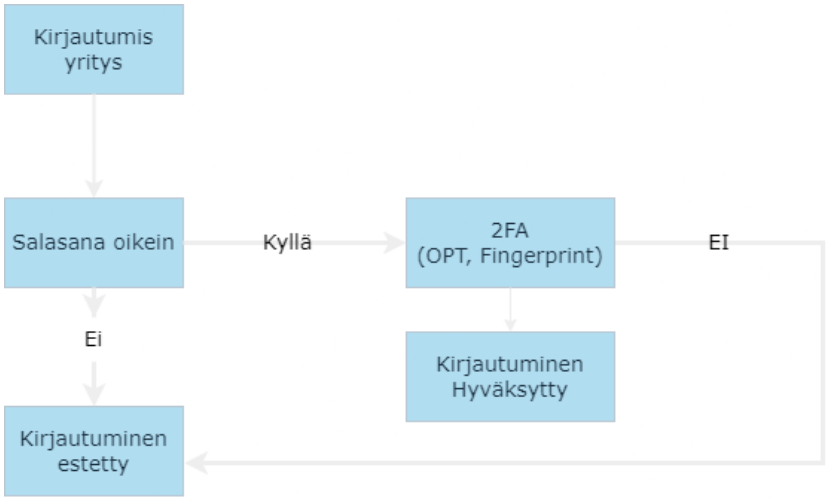
Käytännössä kolme ensimmäistä todennustekijää ovat yleisiä verkkosovelluksissa, mutta monivaiheisessa tunnistautumisessa voidaan käyttää myös muita menetelmiä. Näihin kuuluvat käyttäjän varmentaminen sijainnin perusteella (IP-osoite tai GPS), toiminnan perusteella (näppäinpainallukset, hiiren liikkeet, kävelyanalyysi) tai sosiaalisten tekijöiden perusteella (käyttäjän tuntema henkilö). (Multifactor authentication cheat sheet, 14; Vapen Anna, 29)



Kuva 1. Venn-diagrammissa kolme osaa, jotka kuvaamassa eri tunnistetyyppejä

Monivaiheinen tunnistautuminen edellyttää kahta tai useampaa osajoukkoa eri tunnistetyypeistä ja näiden lisäksi voidaan käyttää myös muita tunnistetyyppejä tunnistuksen vahvistamiseksi.

Kuvassa 2 monivaiheisen tunnistautumisen prosessi kaaviona.



Kuva 2. Monivaiheisen tunnistautumisen prosessi

Jos jokin tunnistustyyppi ei täsmää sovellukseen tai verkkosivustolle tallennettujen tietojen kanssa, kirjautumisyritys estetään. Jos tunnistustyypit vastaavat tallennettuja tietoja, käyttäjä saa valtuutuksen käyttää sovellusta tai verkkosivustoa.

4.2 Kolmannen osapuolen kirjautumispalvelut

TPA-todennus (Third-Party Authentication) on kaksi- tai monivaiheisen tunnistautumisen prosessi, jossa kolmas osapuoli varmentaa käyttäjän henkilöllisyyden tai oikeuden päästä tiettyihin resursseihin tai palveluihin. Tämä auttaa suojaamaan tietojärjestelmiä ja varmistamaan, että vain oikeutetut käyttäjät pääsevät niihin käsiksi.

TPA-todennuksessa käyttäjän tunnistetiedot ja oikeudet tarkistetaan kolmannen osapuolen tarjoaman todennuspalvelun avulla. Todennusprosessi voi sisältää

erilaisia vaiheita, kuten käyttäjän tunnistamisen, salasanan tarkistuksen, monivaiheisen todennuksen tai muunlaisen varmennuksen. (Vapen Anna, 29)

4.3 Todennusmenetelmien riskit

Todennusprosessit, olipa kyseessä henkilöiden, järjestelmien tai laitteiden varmentaminen, sisältävät aina riskejä. Riskien minimoimiseksi on välttämätöntä tunnistaa riskit ja luoda suojakeinoja niitä vastaan, jotta todennusprosessit olisivat luotettavia ja turvallisia. Jatkuva riskienhallinta ja tietoturvan kehittäminen ovat elintärkeitä, jotta todennusmenetelmät pysyvät ajantasaisina ja kestävinä.

Eri todennusmenetelmät tarjoavat erilaisia turvallisuustasoja käyttäjien tunnistamiseen. Mitä useampia todennustekijöitä on käytössä, sitä turvallisempi todennusmenetelmä on. Turvallisuustason noustessa käyttömukavuus ja käyttöönoton helppous saattaa kärsiä. Tämä saattaa nostaa kustannuksia, mutta kuitenkin vähentää tietoturvaan liittyviä riskejä. Todennusmenetelmän valinta riippuu tyypillisesti sovelluksen luonteesta ja sen sisältämän tiedon herkkyydestä. (Ometov, Aleksandr yms., 17)

4.3.1 Yksivaiheisen todennuksen (SFA) riskit

Yksivaiheinen todennus, jossa käyttäjä tunnistautuu vain yhdellä todennustekijällä, kuten salasanalla, tarjoaa heikoimman suojan tietoturvariskejä vastaan. Yleisimpiä SFA:n riskejä ovat:

- Brute-force-hyökkäykset, jossa hyökkääjä pyrkii arvaamaan salasanan kokeilemalla järjestelmällisesti eri yhdistelmiä

- Tietojenkalastelu, jossa hyökkääjä lähettää huijausviestejä ja yrittää houkutella käyttäjän paljastamaan salasanansa tai muita tunnistautumistietoja

- Keylogging (haittaohjelma) tallentaa käyttäjän näppäimistön painalluksia, mukaan lukien salasanat

- Sosiaalinen manipulointi, jossa hyökkääjä yrittää huijata käyttäjää paljastamaan salasanansa tai muita tunnistautumistietoja

- Salasanojen tallennus selkeänä tekstinä, jossa salasanat tallennetaan selkeänä tekstinä, jolloin ne ovat helposti varastettavissa

- Kirjautumistietojen purkaminen, jossa hyökkääjä varastaa käyttäjien kirjautumistietoja tietomurron tai verkkohyökkäyksen yhteydessä

- Verkon nuuskinta, jossa hyökkääjä kaappaa verkkoyhteyden ja sieppaa käyttäjien kirjautumistietoja

- Haittaohjelmia voidaan käyttää varastaessa käyttäjien kirjautumistietoja tai ohittaessa todennusmekanismeja.

Lisäksi jatkuva uusien hyökkäysmenetelmien kehitys ja heikkojen tai uudelleenkäytettävien salasanojen käyttö heikentävät entisestään SFA:n tietoturvaa. Tämän vuoksi on tärkeää harkita monivaiheisen todennuksen (MFA) käyttöönottoa, joka parantaa merkittävästi tietoturvaa verrattuna SFA:han. (Cisa.gov, 4)

4.3.2 Monivaiheisen todennuksen (MFA) riskit

Vaikka monivaiheinen tunnistautuminen parantaa huomattavasti tietoturvaa, ei sekään ole täysin riskitön ratkaisu. Tutkimuksessa havaittiin, että MFA:n käyttö esti 100 prosenttia automatisoiduista bottihyökkäyksistä, 99 prosenttia tietojenkalasteluhyökkäyksistä ja 66 prosenttia kohdennetuista hyökkäyksistä käyttäjien Google-tileihin. (Cisa.gov, 4)

Yleisimpiä riskejä monivaiheisessa tunnistautumisessa ovat:

- Tietojenkalastelussa (phishing) hakkerit voivat käyttää kalastelua hyväkseen päästääkseen kiinni arkaan dataan. Hakkerit voivat lähettää väärennettyjä sähköposteja, tekstiviestejä tai huijaussivustoja, jotka näyttävät aidoilta palveluntarjoajilta.

- Kirjautumistietojen varastamisen (Credential Theft) avulla hakkerit voivat yrittää varastaa käyttäjän tunnistautumistietoja haittaohjelmien tai verkkoliikenteen nuuskinnan avulla. Jos käyttäjä antaa tunnistautumiskoodinsa, niin hakkerit voivat päästä käsiksi tilille.

- SIM-kortin vaihto (SIM-swap), jossa hakkeri siirtää liittymän omistajan puhelinnumeron toiselle SIM-kortille

- SMS-pohjaiset man-in-the-middle-hyökkäykset

- Keyloggerit ovat haittaohjelmia tai fyysisiä laitteita, jotka tallentavat näppäimistösyötteitä, joita käyttäjät kirjoittavat tietokoneella tai muulla laitteella. Ne voivat tallentaa kaikki näppäimistön painallukset, kuten salasanat, luottokorttitiedot, viestit tai muut henkilökohtaiset tiedot.

- Pass-the-cookie-hyökkäyksessä verkkorikolliset käyttävät varastettuja verkkoistunnon evästeitä jäljitelläkseen laillista käyttäjää ja saadakseen pääsyn tietoihin ja järjestelmiin.

  (Farik Mohammed, 6; Cisa.gov, 4)

MFA:n merkittävä haaste ilmenee hallinnollisen monimutkaisuuden kasvuna sekä järjestelmänvalvojien että loppukäyttäjien näkökulmasta. MFA:n käyttö saattaa tuottaa vaikeuksia käyttäjille, joilla ei ole vahvaa teknistä osaamista. Lisäksi on muita yleisiä ongelmia:

- MFA, joka edellyttää käyttäjiltä tietynlaista laitteistoa, saattaa tuoda merkittäviä taloudellisia ja hallinnollisia kustannuksia

- Tilanteessa, jossa käyttäjä menettää tai ei kykene käyttämään muita tunnistustekijöitä, voi käyttäjän pääsy tililleen estyä

- MFA:n käyttöönotto lisää sovelluksen monimutkaisuutta ja voi vaatia lisää resursseja ylläpitoon ja hallintaan

- Monet MFA-ratkaisut lisäävät järjestelmän ulkoisia riippuvuuksia, jotka voivat altistaa järjestelmät tietoturva-aukoille tai yksittäisille vikakohdille.

- Hyökkääjät voivat käyttää hyväkseen käyttöön otettuja prosesseja ohittaakseen tai nollatakseen monivaiheisen tunnistautumisen

- MFA:n käyttöönotto voi rajoittaa tiettyjen käyttäjien pääsyä sovellukseen, joka saattaa vaikuttaa käyttäjäkunnan saatavuuteen ja käyttökokemukseen.

(Multifactor authentication cheat sheet, 14)

5 Web 3.0 ja web3 tunnistautuminen

5.1 Web 3.0

Web 3.0 edustaa internetin kehityksen seuraavaa vaihetta, joka mahdollistaa peer-to-peer-tiedonvaihdon ja poistaa tarpeen välikäsille ja kolmansille osapuolille, antaen käyttäjille mahdollisuuden hallita omia tietojaan. Tämä antaa yksilöille ja organisaatioille mahdollisuuden jakaa dataa tasapuolisesti ja läpinäkyvästi, ilman yksittäisen tahon määräysvaltaa. (Petcu Adrian, 18; Sitra, 22)

Web 3.0 ja Web3-termit sekoitetaan usein, koska internetin seuraava aikakausi yhdistää todennäköisesti molempien konseptien elementtejä, kuten semanttiset verkkosovellukset, linkitetyn datan ja lohkoketjupohjaisen talouden. Ei ole vaikea ymmärtää miksi tälle alueelle tehdään merkittäviä investointeja. (Ma Adrian, 12; Wensheng Gan,28)

Timothy John Berners-Lee esitteli semanttisen webin osana Web 3.0:n visiota, jonka tavoitteena on yhdistää dataa ja hyödyntää konepohjaista tiedon ymmärtämistä. Semanttinen web mahdollistaa koneiden tiedon käsittelyn inhimillisem-

mällä tavalla, joka tarjoaa entistä monipuolisemmat ja mukautuvammat digitaaliset palvelut. Tämä edistää tekoälyn ja koneoppimisen integroimista päivittäiseen verkkoselaamiseen, parantaen käyttökokemusta tarjoten uudenlaisia mahdollisuuksia tiedon hyödyntämiseen. (Wensheng Gan ym, 28; Nilesh Sable ym., 16; Expert.ai Team; Tayyab, Mahmood, 24)

Gavin Wood, yksi Ethereum:in perustajista loi termin ”Web3 heti Ethereum:in lanseerauksen jälkeen vuonna 2014. Kryptovaluuttojen varhaiset käyttäjät olivat huolissaan siitä, että vallitseva verkko vaati liiallista luottamusta. Tämän ongelman ratkaisemiseksi Wood esitti Web3:n, joka tarjoaa hajautetun ja luottamuksettoman vaihtoehdon. (Nilesh Sable ym, 16; Wensheng Gan ym.,28)

Web 3.0 toimii siten, että data tallennetaan lohkoketjuun yritysten ja organisaatioiden hallinnoimien tietokantojen sijaan. Käyttäjillä on oma digitaalinen tallelokeronsa internetissä, joka on julkisesti näkyvillä, mutta vain käyttäjä voi avata sen tai myöntää pääsyn lokerolle. (Blomster Henri, 3)

5.2 Web3

Web3 on Internetin seuraavaa evoluutiota, jossa lohkoketjuteknologia, virtuaalivaluutat ja NFT:tä mahdollistavat käyttäjille suuremman omistajuuden ja autonomian. Sen vahvuudet tulevat esiin hajautetuissa sovelluksissa (dApps), jotka ovat turvallisempia, avoimempia ja tarjoaa vahvemman suojatun vaihtoehdon perinteisiin keskitettyihin palveluihin verrattuna. Tämä mahdollistaa uusia liiketoimintamalleja, jotka eivät olleet toteutettavissa perinteisessä Internetissä. (Nilesh Sable ym., 16)

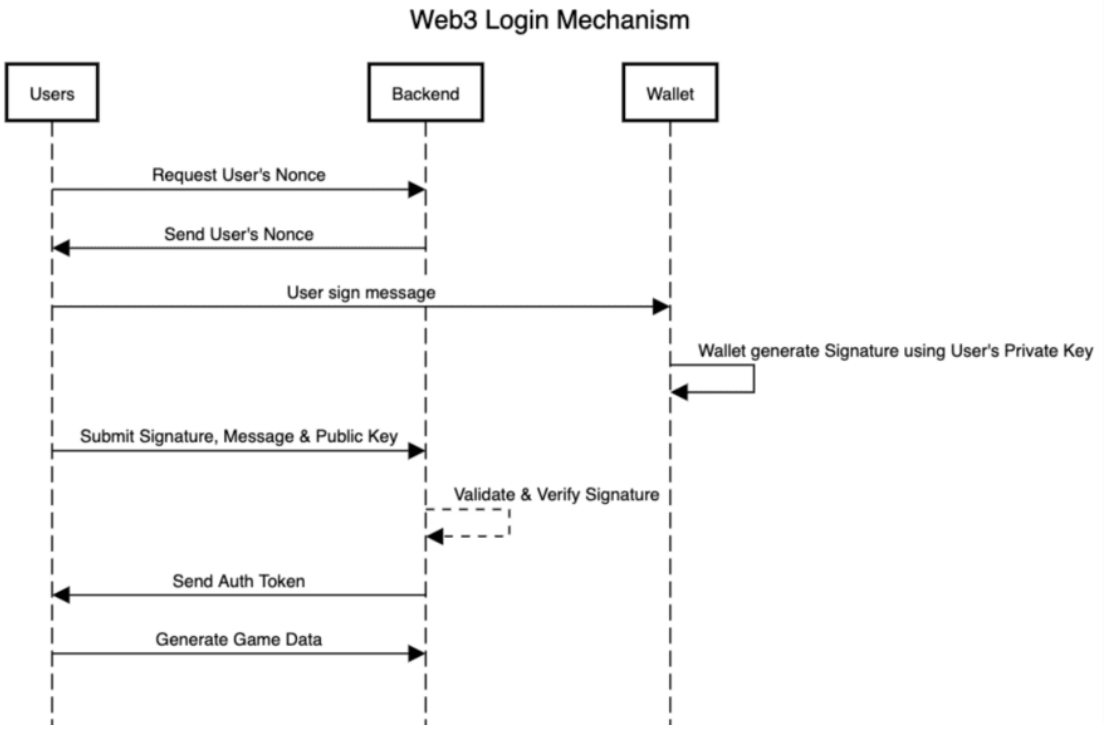
Web3 mahdollistaa sen, että käyttäjät voivat paremmin hallita henkilötietojaan ja voivat halutessaan jakaa vain haluamansa tiedot itsestään. Tämän mahdollistaa sellaiset tekniikat kuin nollatietotodisteet ja hajautetut identiteettijärjestelmät. Lisäksi Web3:lla on potentiaalia mullistaa tapa, jolla suoritamme taloudellisia transaktioita, mahdollistamalla lohkoketjupohjaiset maksujärjestelmät, jotka hoitavat maksut välittömästi ja alhaisemmin kustannuksin kuin perinteiset maksujärjestelmät. (Nilesh Sable ym., 16)

Web3:n keskeiset haasteet ovat skaalautuvuus ja sääntelykysymykset. Nykyiset lohkoketjuteknologiat, kuten Ethereum, kohtaavat rajoituksia tapahtumien käsittelynopeudessa, mikä voi haitata Web3-sovellusten laajentumista. Lisäksi sääntelyviranomaiset ovat kiinnostuneita valvomaan Web3-sovelluksia, mikä voi vaikuttaa niiden kehitykseen ja käyttöönottoon. Ratkaisujen löytäminen näihin haasteisiin edellyttää sekä teknologisia innovaatioita ja yhteistyötä sääntelyelinten kanssa. (Nilesh Sable ym., 16)

5.3 Web3 tunnistautuminen

Tunnistautuminen on keskeistä kyberturvallisuuden kannalta. Monet perinteiset todennusmenetelmät toimivat kuitenkin keskitetyssä ympäristössä, mikä voi aiheuttaa riskejä, kuten identiteettivarkauksia ja henkilötietojen vuotamista. Näiden riskien minimoimiseksi on ehdotettu lohkoketjupohjaista tunnistautumista. Turvallisuusanalyysi tietoturvasta osoittaa, että lohkoketjupohjaiset järjestelmät ovat erityisen tehokkaita torjumaan kyberhyökkäyksiä, jotka kohdistuvat tunnistautumisjärjestelmiin. Käytännön testit vahvistavat tämän, näyttäen lohkoketjuteknologian ylivoimaisen suorituskyvyn verrattuna keskitettyihin tunnistautumisjärjestelmiin. (Li Beibei ym, 10; Sitra, 22)

Hajautetun identiteetin (decentralized identity) tai DID (Decentralized Identifier) -tunnistautuminen, on uudenlainen tapa tunnistautua ja hallita identiteettiä web3-teknologian avulla hajautetuissa järjestelmissä, kuten lohkoketjuissa. Se eroaa perinteisestä keskitetystä tunnistautumisesta, jossa käyttäjät luottavat kolmansiin osapuoliin, kuten sosiaalisen median alustoihin tai palveluntarjoajiin, tunnistautuakseen eri verkkopalveluihin. Web3-todennuksen etuja ovat käyttäjien lisääntynyt tietoturva ja yksityisyys sekä käyttäjien mahdollisuus hallita tietojaan paremmin. (Petcu Adrian, 18)



Kuva 3. Web3 käyttäjän kirjautumismekanismi (AnggaPurz, 1)

Web3-todennus saattaa kuulostaa monimutkaiselta, mutta se on pohjimmiltaan kirjautumismekanismi, joka hyödyntää krypto-osoitteita perinteisen keskitetyn tunnistautumisen sijasta. Tietyt lohkoketjut tukevat Web3-sivustoja, sovelluksia ja muita palveluita. Käyttäjien on pystyttävä turvallisesti yhdistämään näihin kryptoverkkoihin Web3-lompakon avulla. Tunnistautumisen jälkeen käyttäjät voivat liittyä verkkoon ja kommunikoida muiden tunnistautuneiden käyttäjien kanssa. Siksi jokainen Web3-sovellus tarvitsee tämän tunnistautumismenetelmän. (Nilesh Sable ym., 16)

Web3 tunnistautumisen tärkeimpiä ominaisuuksia ovat:

- Omistajuus: Käyttäjät omistavat tunnistautumisavaimensa, jotka ovat yksityisiä ja salaisia. Julkiset avaimet toimivat tunnisteina ja todentavat käyttäjien identiteetin.

- Hajautettu ja itsenäinen: Tunnistautuminen ei perustu keskitettyihin palveluihin, kuten käyttäjätunnuksiin ja salasanoihin. Sen sijaan käyttäjät voivat tunnistautua hajautettujen protokollien ja älysopimusten avulla.

- Yksityisyys: Käyttäjien henkilökohtaiset tiedot ovat turvassa, koska niitä ei luovuteta kolmansille osapuolille. Käyttäjät voivat valita, mitä tietoja he jakavat verkkopalveluiden kanssa.

- Itsenäiset sovellukset: Käyttäjät voivat käyttää eri sovelluksia ja palveluita ilman, että heidän tarvitsee luoda uusia käyttäjätunnuksia ja salasanoja jokaiselle palvelulle erikseen.

6 Oauth2.0 ja Web3-kirjautuminen web-sovelluksessa

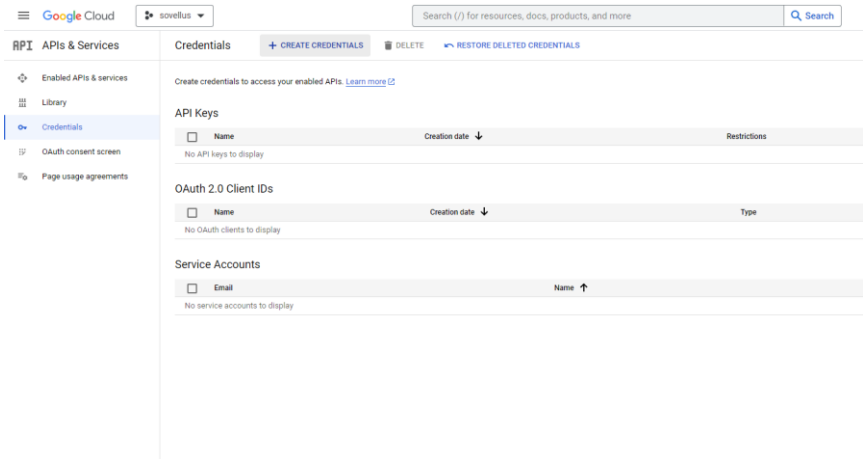
Opinnäytetyön tarkoituksena oli toteuttaa sovellus, jossa käyttäjät voivat kirjautua sekä OAuth 2.0- että Web3-tunnistautumisella. Sovelluksen toteutettiin JavaScript:in React-kirjastoa ja Java:n Spring Boot -kehystä käyttäen. Sovelluksen Frontend:iä varten käytin esimerkkinä muutamia eri oppaita sovelluksen rakentamista varten ja Backend:in toteutin vain esimerkkinä, siitä miten tiedot voitaisiin mahdollisesti tallentaa erilliseen tietokantaan.

6.1 Google cloud console:n käyttöönotto

Sovellusta varten tarvitaan ainutlaatuinen Google OAuth 2.0 -asiakastunnus, joka liittää asiakkaan ja palvelimen OAuth 2.0 -todennuksen kanssa. Tämän saavuttamiseksi Google Cloud Consolessa otetaan käyttöön asiakas-ID, salausavain ja määritetään tarvittavat asetukset, kuten sallitut URI-osoitteet. (Muhammed Sahad, 15)

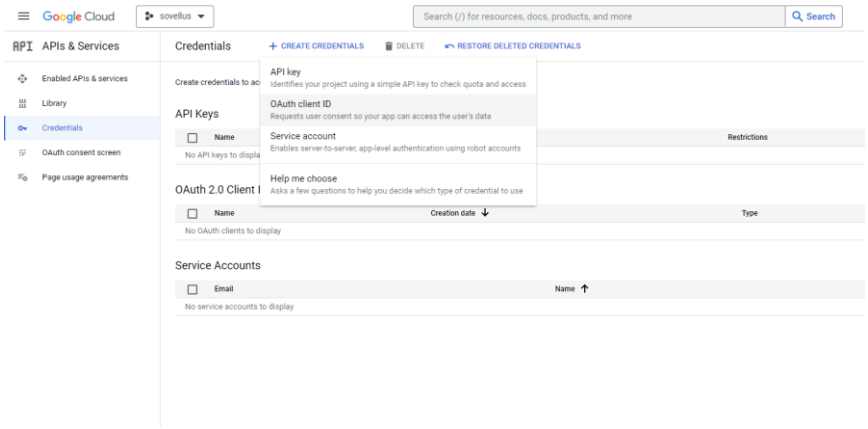
6.2 Uuden projektin luominen Google Cloud Console:ssa

Kuvassa 4 valitaan kohta credentials, josta valitaan create credentials



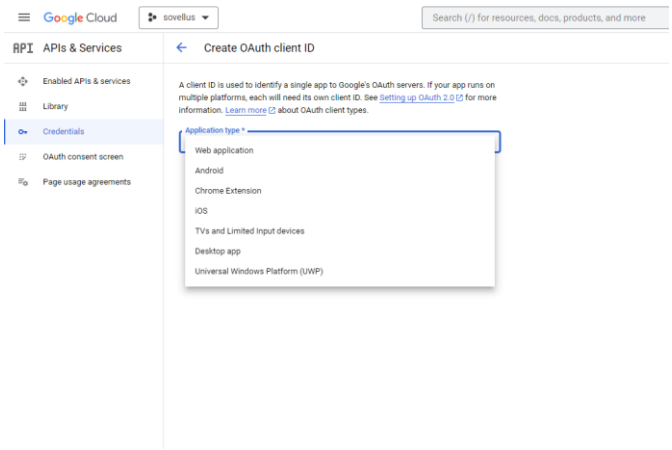
Kuva 4. Google API&Services create credentials

Kuvassa 5 valitaan Create credentials kohdasta OAuth client ID



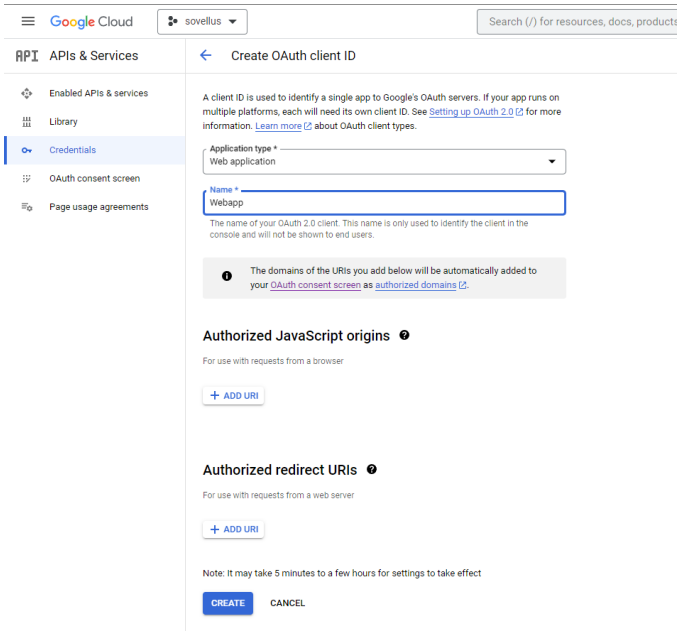
Kuva 5. Google Cloud credentials OAuth client ID

Kuvassa 6 valitaan sovellustyyppi



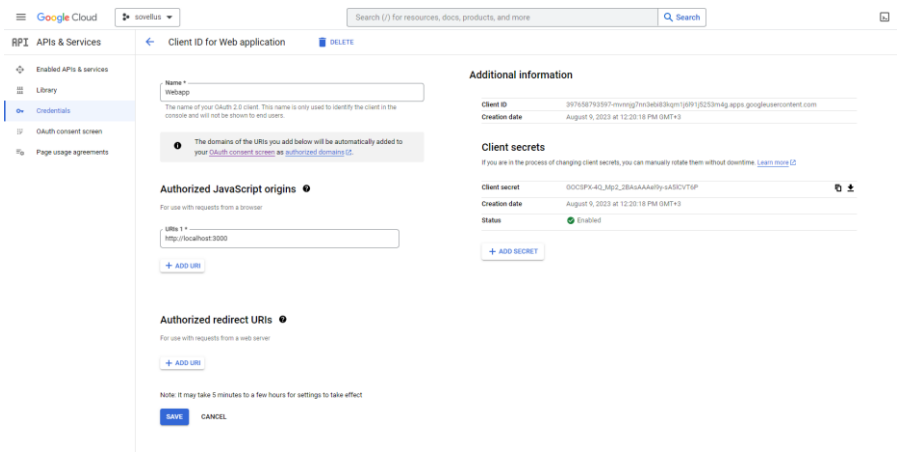
Kuva 6. Google Cloud Console valitaan application type

Kuvassa 7 nimetään sovellus ja valitaan valtuutettu selainosoite ja uudelleenohjausosoitteet, joiden perusteella määritellään sovelluksen nimi ja sallitut osoitteet.



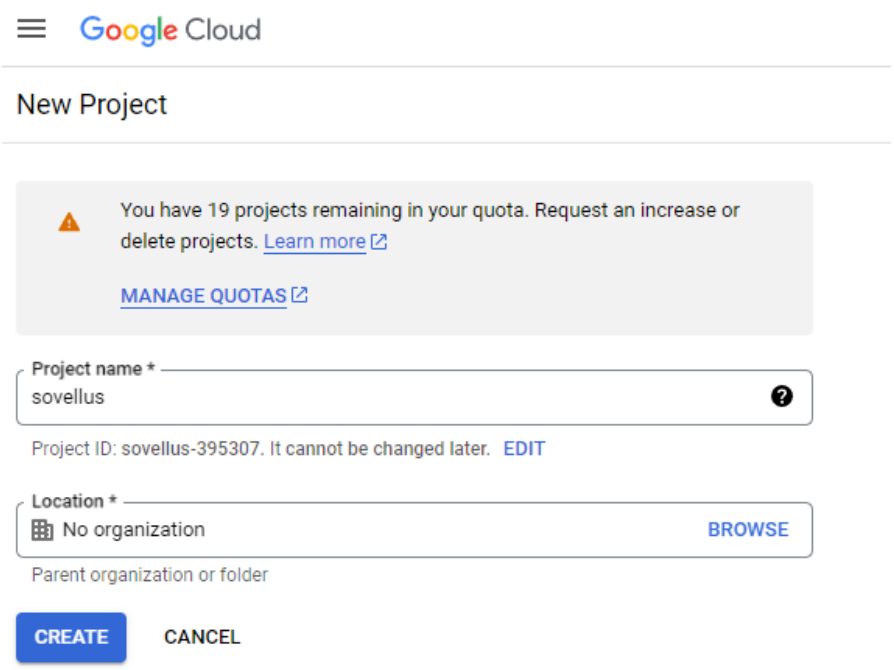
Kuva 7. Sovelluksen nimi ja sallitut uudelleenohjausosoitteet

Kuvassa 8 Googlen Cloud Console luo Client ID:n ja Client Secret:in, joita tarvitaan esimerkkisovelluksessa, jossa ne on hyvä tallentaa ympäristömuuttujiin.



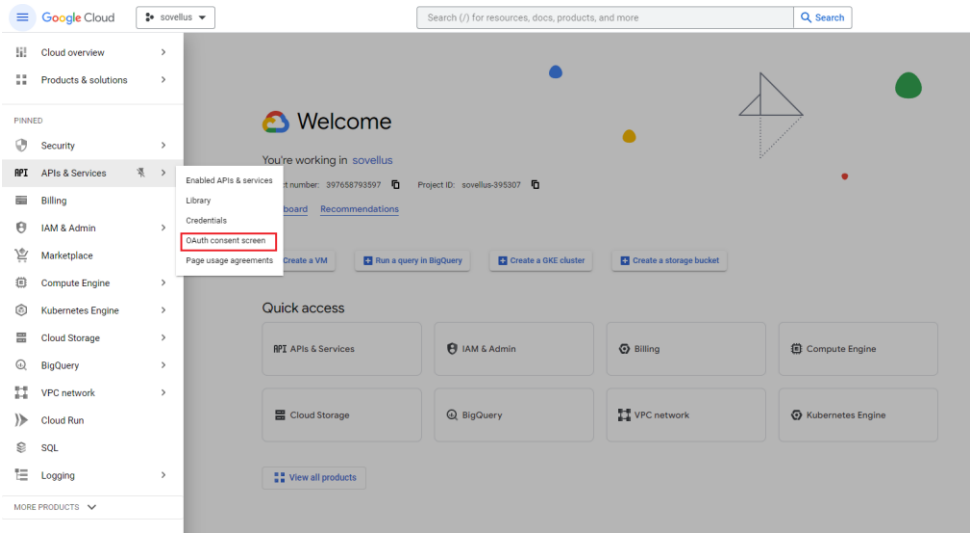
Kuva 8. Client ID ja Client Secret

Kuvassa 9 luodaan ja nimetään uusi projekti



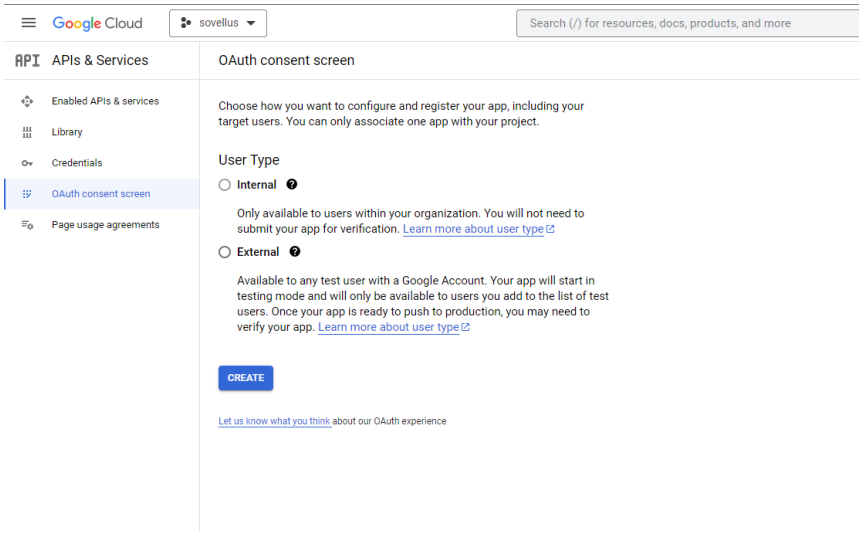
Kuva 9. Uuden projektin luominen Google Cloud-palvelussa

Kuvassa 10 Googlen Cloud Consolen valikosta valitaan APIs & Services alas-vetovalikosta OAuth consent screen



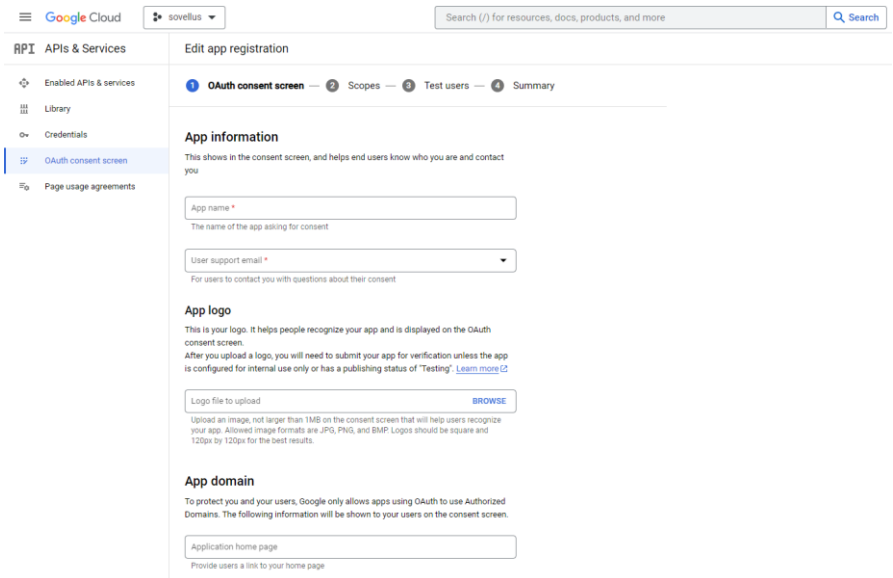
Kuva 10. Google Cloud Console OAuth consent screen

Kuvassa 11 valitaan käyttäjätyyppi riippuen sovelluksen käyttökohteesta. Tämän esimerkin testausvaiheessa valitaan external (ulkoinen).



Kuva 11. Google Cloud Console User type (käyttäjä tyyppi) valinta

Kuvassa 12 rekisteröidään sovellus ja määritellään tarvittavat tiedot, kuten sovelluksen nimi, käyttäjätuen sähköposti, sovelluksen verkkotunnus, kehittäjän yhteystiedot ja mahdollinen logo. Näitä asetuksia voi vielä muuttaa myöhemminkin.

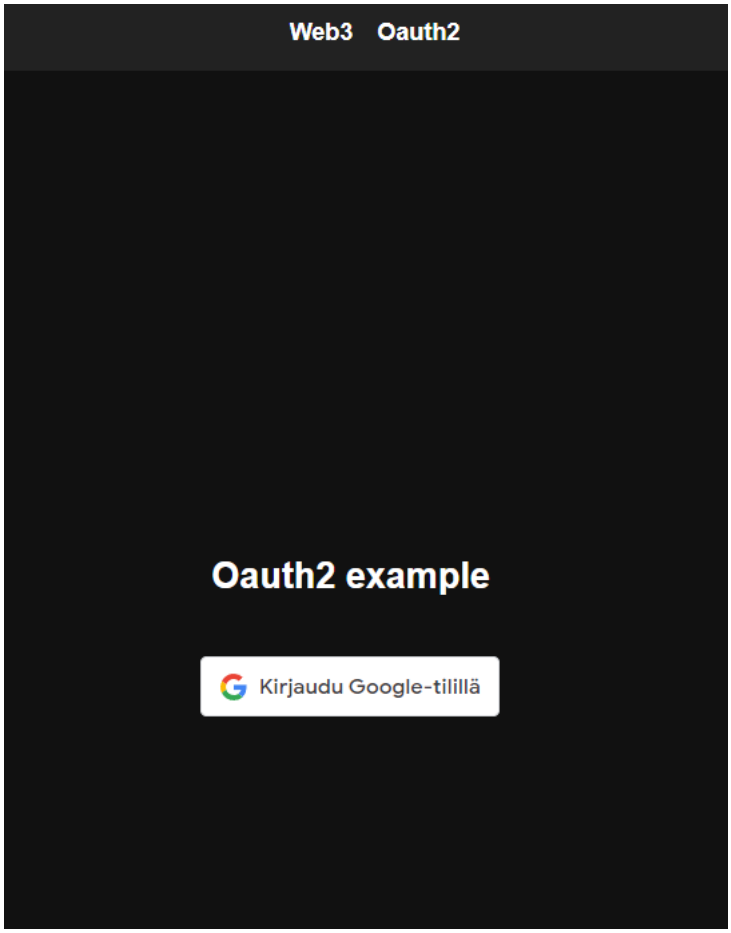


Kuva 12. Google Cloud Console App Information (sovelluksen tiedot)

Googlen OAuth2.0 asetusten jälkeen Google CloudID ja Cloud Secret-arvoja voidaan nyt käyttää sovelluksessa, jotka on hyvä tallentaa ympäristömuuttujiin viimeistään tuotantovaiheessa.

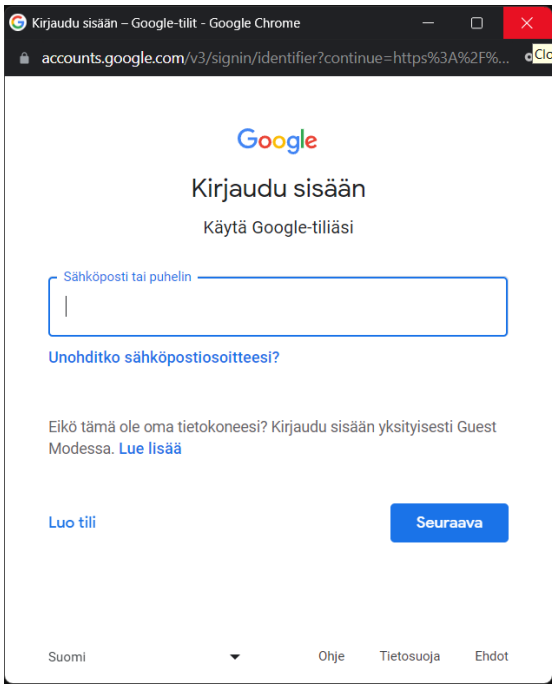
6.3 Esimerkki sovellus OAuth2.0 kirjautumisella

Kuvassa 12 on esimerkkisovelluksen käyttöliittymä, jossa käyttäjä voi painaa nappia avatakseen OAuth2.0-ponnahdusikkunan (kuva 13), jonka kautta voi kirjautua Google-tilillä. Käyttöliittymän ylälaidassa on myös kirjautumisvaihtoehdot web3 ja oauth2-kirjautumisten välillä.



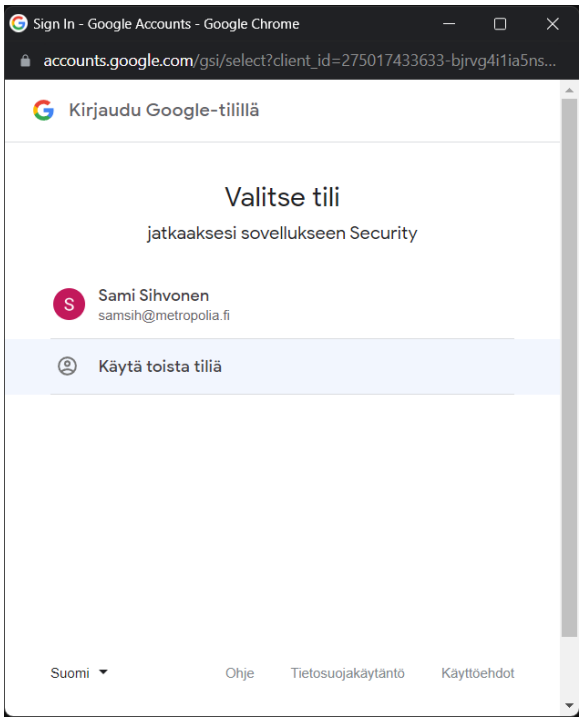
Kuva 12. Sovelluksen käyttöliittymä

Kuvassa 13 sovelluksen OAuth2.0-pohjainen ponnahdusikkuna



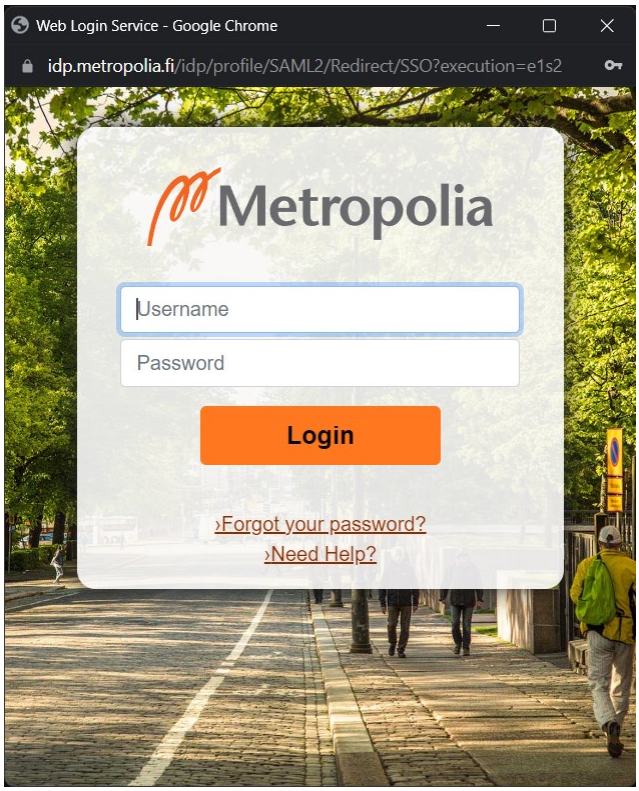
Kuva 13. Google:n sisäänkirjautuminen

Kuvassa 14 käyttöliittymän ponnahdusikkuna pyytää varmentamaan tilin ulkoisesta lähteestä.



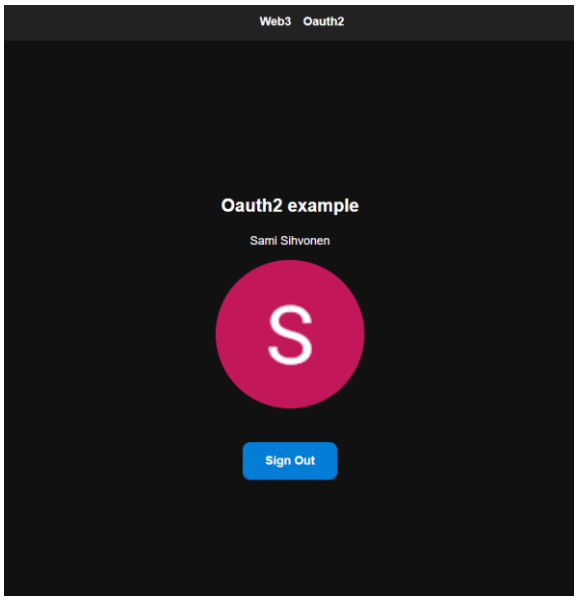
Kuva 14. Tilin valitseminen

Kuvassa 15 varmennus tapahtuu Metropolian kirjautumissivuston kautta.



Kuva 15. Metropolian sisäänkirjautuminen

Kuvassa 16 sovellus hakee käyttäjän tiedot googlen palvelimelta.



Kuva 16. Google-tilin käyttäjätiedot haettuna sovellukseen

6.4 Web3Auth-kirjautuminen esimerkkisovelluksessa

Web3 tunnistautuminen on rakennettu selaimen päälle, jossa käyttäjä kirjautuu selaimessa olevan lompakon avulla. Käyttäjä voi valita oman lompakkonsa ja antaa luvan sovellukselle käyttää lompakon tunnistetietoja. Tämä tapahtuu yleensä kryptografisen allekirjoituksen avulla, joka vahvistaa käyttäjän henkilöllisyyden lohkoketjussa. (Rickard Elsen ym, 20)

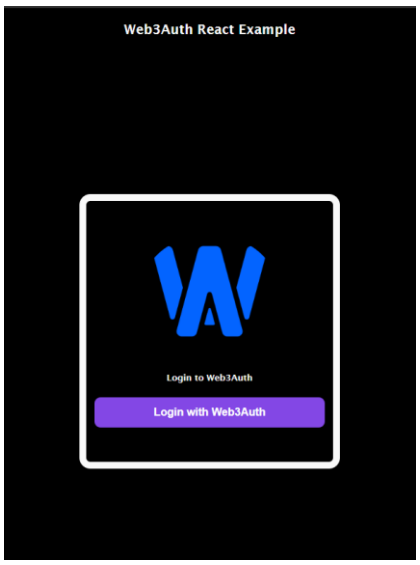
Web3Auth:in sivuston kattava dokumentaatio helpottaa kehittäjiä integroimaan Web3Auth:in kirjautumisen web-sivustolle. Web3Auth on yhteensopiva kaikkien OAuth-pohjaisten kirjautumisjärjestelmien kanssa, oli kyse sitten verkkosivustoista tai mobiilialustoista, joka takaa käyttäjille vaivattoman käyttökokemuksen. (Torus Labs, 25)

Kuvassa 17 Web3Auth:in dokumentaatioon perustuva koodiesimerkki

```
const Web3 = () => {
  const [web3auth, setWeb3auth] = useState<Web3Auth | undefined>(undefined);
  const [provider, setProvider] = useState<IProvider | undefined>(undefined);
  const [address, setAddress] = useState<string[]>([]);
  const [balance, setBalance] = useState<string>("");
  const [chainId, setChainId] = useState<string>("");
  const [userData, setUserData] = useState<User | null>(null);


  useEffect(() => {
    const initializeWeb3Auth = async () => {
      try {
        const web3authInstance = new Web3Auth({
          clientId: import.meta.env.VITE_WEB3AUTH_CLIENT_ID,
          web3AuthNetwork: WEB3AUTH_NETWORK.SAPPHIRE_MAINNET,
          privateKeyProvider: privateKeyProvider,
        });
        await web3authInstance.initModal();
        setWeb3auth(web3authInstance);
        const isConnected = await web3authInstance.connect();
        if (isConnected) {
          setProvider(web3authInstance.provider as IProvider | undefined);
        }
      } catch (error) {
        console.error("Error initializing Web3Auth:", error);
      }
    };
    initializeWeb3Auth();
  }, []);

  useEffect(() => {
    if (provider) {
      getUserInfo();
      getAccounts();
      getChainId();
      getBalance();
    }
  }
  , [provider]);

  const handleLogin = async () => {
    try {
      if (!web3auth) {
        console.log("Web3Auth not initialized yet");
        return;
      }
      const web3authProvider: any = await web3auth.connect();
      setProvider(web3authProvider.provider);
    } catch (error) {
      console.error("Error during login:", error);
    }
```

Kuva 17. VSCoden-koodieditorissa kokeiltu koodi Web3 tunnistautumisesta.

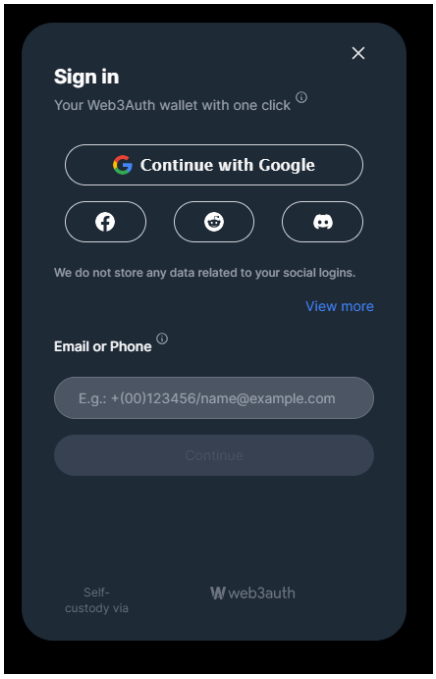
Kuvassa 18 käyttöliittymän ”Login with Web3Auth”-nappi, josta voi valita todennuspalvelun



Kuva 18. Web3Auth-todennuspainike käyttöliittymässä

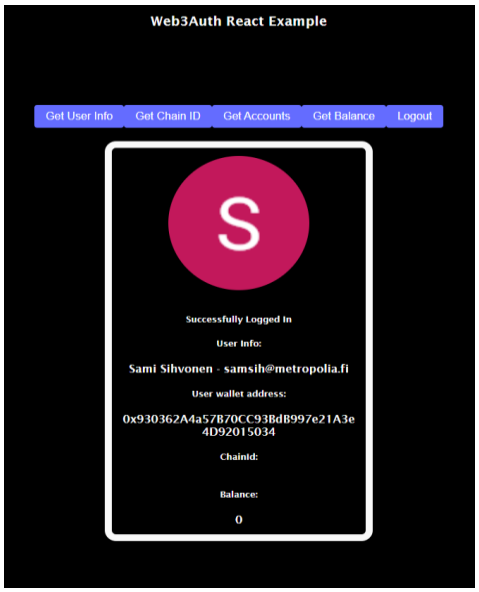
Kuvassa 19 sovelluksen käyttöliittymään tulee esiin Web3Auth-ponnahdusikkuna, josta käyttäjä voi valita mieleisensä kirjautumisvaihtoehdon.

Web3Auth:issa pystyy valita mielensä mukaan eri todennuspalveluita käyttöön ja niiden lisäksi käyttäjä voi tunnistautua sähköpostin tai puhelimen välityksellä.



Kuva 19. Web3Auth:n kirjautumisikkuna

Kuvassa 20 kirjautumisen jälkeen selaimeen tulee käyttäjän tiedot riippuen siitä mitä sosiaalisen kirjautumisen vaihtoehtoa on käyttänyt.



Kuva 20. Käyttäjän tiedot selaimessa

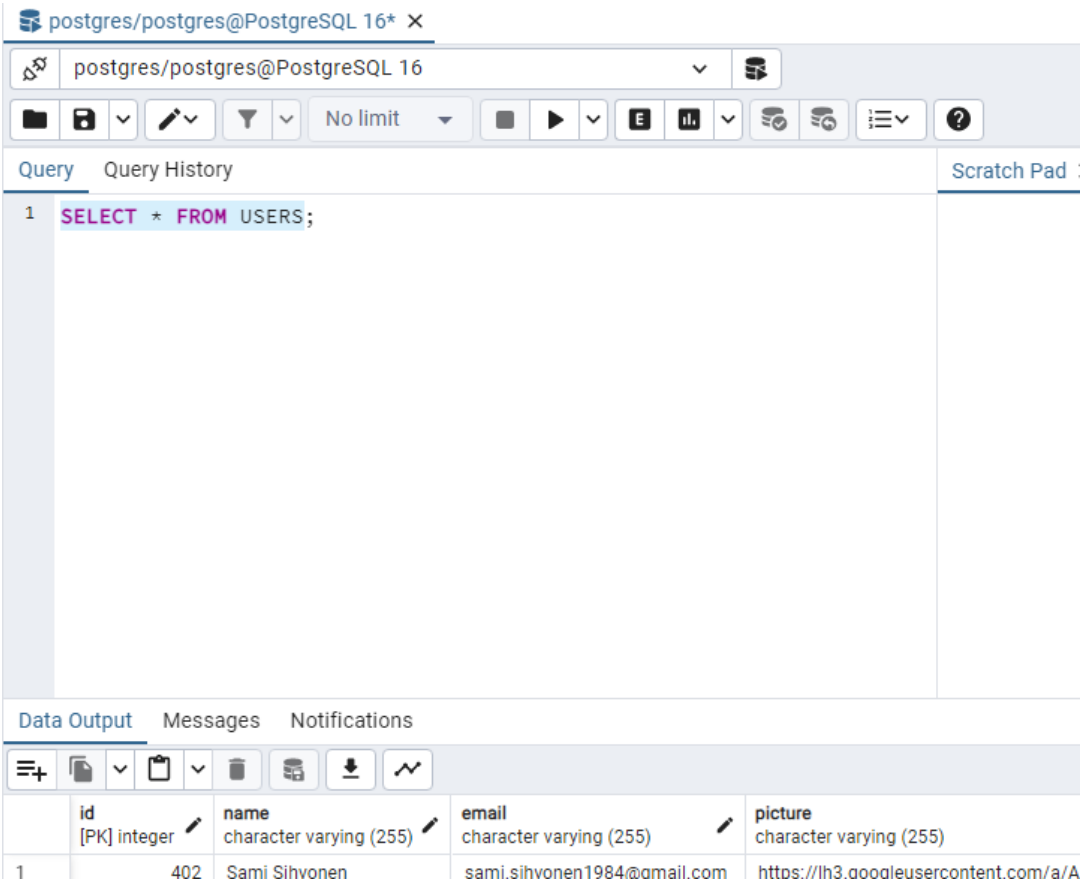
Kolmannen osapuolen ulkoiselta palvelimelta haettuja tietoja voidaan käyttää sovelluksen tarpeiden mukaan. Esimerkissä käytin backendissä Java:n Spring Boot -kehystä ja PostgreSQL-tietokantaa käyttäjätietojen tallentamiseen paikalliseen tietokantaan.
Kuvassa 21 on koodiesimerkki Java:n käyttäjätaulun mallista, jossa valittu tietueet frontendin antamasta responssista.

```java
@Entity
@Table(name = "users")
@Data
public class User implements Serializable {
    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    private Long id;

    private String name;

    private String email;

    private String picture;

    public User() {
    }

    public User(Long id, String name, String email, String picture) {
        this.id = id;
        this.name = name;
        this.email = email;
        this.picture = picture;
    }

    public Long getId() {
        return id;
    }

    public void setId(Long id) {
        this.id = id;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String getEmail() {
        return email;
    }

    public void setEmail(String email) {
        this.email = email;
    }

    public String getPicture() {
        return picture;
    }

    public void setPicture(String picture) {
        this.picture = picture;
    }

}
```

Kuva 20. Käyttäjätaulun malli

Spring Bootin käyttäjätaulun mallin mukaisesti käyttäjän tiedot voidaan tallentaa paikalliseen tietokantaan. Tässä esimerkissä käytetään PostgreSQL-tietokantaa tietojen tallentamiseen.

Kuvassa 21 haettuna tietokantaan tallennetut käyttäjät



Kuva 21. PgAdmin-tietokanta-editori

7 Yhteenveto

Opinnäytetyössä pyrittiin tutkimaan eri tunnistautumismenetelmiä ja luomaan sovellus, jossa käyttäjä voi kirjautua helposti ja käyttäjäystävällisesti. Työn tarkoituksena oli demonstroida OAuth2.0- ja Web3-teknologioiden käyttöä vaihtoehtona perinteisille tunnistautumismenetelmille sekä toteuttaa sovellus, joka hyödyntää näitä teknologioita kirjautumisessa.

Työn toteuttaminen ja aiheiden tutkiminen oli erityisen haastavaa erittäin laajan aineiston vuoksi. Lisäksi Web 3.0:n ja Web3:n määrittely on yhä keskeneräistä monien tutkijoiden keskuudessa.

Tutkimuksessa pohdittiin hajautetun verkon, kuten Web3-tunnistuksen, merkitystä ja sen tarjoamaa turvaa käyttäjille. Lopuksi tutkimus osoitti, että Web3-tunnistautuminen voi olla suuri edistysaskel digitaalisen tunnistamisen alalla, tuoden käyttäjille lisäturvaa ja poistamalla tarpeen erillisille kirjautumisprosesseille.

Projektille asetetut tavoitteet sovelluksen luomisesta sekä monivaiheisen ja Web3-teknologian esittelystä ja vertailusta perinteisiin teknologioihin saavutettiin onnistuneesti. Tavoitteiden saavuttamisen myötä työ auttaa kehittäjiä toteuttamaan monivaiheisen tunnistautumisen ja Web3-teknologian käyttöönottoa web-sovelluksissa. Insinöörityö voi myös auttaa ja inspiroimaan kehittäjiä ja tietoturva-alan ammattilaisia pohtimaan tietoturvakysymyksiä laajemmin ja monipuolisemmin.

Lähteet

1. AnggaPurz. User Authentication with web3. 22.08.2022. <https://dev.to/anggapur/user-authentication-with-web3-4he8>. Luettu 4.4.2024. (Artikkeli verkkoaineistossa, jossa tekijää ei tiedossa)

2. Arias, Dan. 2019. Hashing Passwords: One-Way Road to Security. Verkkoaineisto <https://auth0.com/blog/hashing-passwords-one-way-road-to-security/> Luettu 06.08.2023. (Vieraskielinen blogi)

3. Blomster, Henri. Mikä on Web 3.0?. 23.11.2021. <https://www.salkunrakentaja.fi/2021/11/web-3-0/>. Luettu 20.05.2024 (Verkkoartikkeli)

4. Cisa.gov, America's cyber defence agency. 08.08.2020. Capacity Enhancement Guide: Implementing Strong Authentication <https://www.cisa.gov/sites/default/files/2023-09/CISA_CEG_Implementing_Strong_Authentication_FINAL%20Aug-23%20Revision.pdf>. Luettu 20.04.2024. (PDF-tiedosto, jossa tekijää ei tiedossa)

5. Expert.ai Team. The 8 Defining Features of Web 3.0. 01.2017. <https://www.expert.ai/blog/web-3-0/>. />. Päivitetty 22.08.2024. Luettu 15.3.2024. (Päivitetty verkkoaineisto, jonka kirjoittajaa ei tiedossa)

6. Farik, Mohammed; Nilesh A. Lal; Salendra Prasad. 11.11.2016. A review of authentication methods. <https://www.researchgate.net/profile/Mohammed-Farik/publication/311514269_A_Review_Of_Authentication_Methods/links/584fbed808aed95c250b4915/A-Review-Of-Authentication-Methods.pdf>. Luettu 18.06.2023. (Vieraskielinen artikkeli)

7. F-secure. 2023. What is two-factor authentication?. 12.2023. <https://www.f-secure.com/en/articles/what-is-two-factor-authentication/> Luettu 27.12.2023. (Vieraskielinen artikkeli, jonka kirjoittaja ei tiedossa)

8. Johnson, Caroline. Unveiling the evolution of multi-factor authentication and what's changing next. 2024. <https://www.researchgate.net/publication/378970415_Unveiling_the_Evolution_of_Multi-Factor_Authentication_and_What's_Changing_Next/>. Luettu 25.03.2024. (Vieraskielinen tutkielma)

9. Lehtonen, Kristo; Marja Pirttivaara; Heikki Aura. Web 3.0 ja eteneminen kohti uutta internetiä – Mistä on kyse ja mitä se meille tarjoaa? 28.03.2022. <https://www.sitra.fi/artikkelit/web-3-0-ja-eteneminen-kohti-seuraavan-sukupolven-internetia-mista-on-kyse-ja-mita-se-meille-tarjoaa/>.Luettu 12.03.2024. (Verkkoartikkeli)

10. Li, Beibei; Tao, Li; Shuang, Zou; Yanbin, Xu; Xinya, Jian. Blockchain-Based Authentication Scheme with an Adaptive Multi-Factor Authentication Strategy. 14.11.2023. <https://www.researchgate.net/publication/375660456_Blockchain-Based_Authentication_Scheme_with_an_Adaptive_Multi-Factor_Authentication_Strategy/>. Luettu 08.03.2024. (Tutkimusartikkeli)

11. Luoma, Ossi. 2019. Biometrinen tunnistaminen työelämässä: Euroopan unionin yleisen tietosuoja-asetuksen rajoitukset ja edellytykset. <https://helda.helsinki.fi/items/846687bf-8235-4435-8468-5bf95e7ac359> Luettu 18.07.2023. (Suomenkielinen gradu)

12. Ma, Adrian. What is Web3 and how it could it change the internet? 02.11.2023. <https://www.weforum.org/agenda/2023/03/what-is-web3-and-how-could-it-change-the-internet/>. Luettu 20.04.2024. (Vieraskielinen verkkoartikkeli)

13. Mizrachi, Aviad. Authentication types explained. 23.09.2019<https://frontegg.com/blog/authentication-types> Luettu 18.06.2023. (Vieraskielinen blogi)

14. Multifactor authentication cheat sheet. <https://cheatsheet-series.owasp.org/cheatsheets/Multifactor_Authentication_Cheat_Sheet.html/>. Luettu 02.03.2024. (Vieraskielinen verkkoaineisto, jossa tekijää ja julkaisupäivää ei ole mainittu)

15. Muhammed Sahad. React JS: A Step-by-Step Guide to Google Authentication. 31.12.2023. <https://muhammedsahad.medium.com/react-js-a-step-by-step-guide-to-google-authentication-926d0d85edbd>. (Vieraskielinen verkkoartikkeli)

16. Nilesh Sable; Rahul Sonkamble; Vijay U Rathod; Swati Shirke & Jyoti Yogesh Deshmukh. Web3 Chain Authentication and Authorization Security Standard. <https://www.researchgate.net/publication/374127372_Web3_Chain_Authentication_and_Authorization_Security_Standard_CAA/>. Luettu 03.03.2024. (Vieraskielinen artikkeli verkkoaineistossa)

17. Ometov, Aleksandr;Bezzateev, Sergey; Mäkitalo, Niko; Andreev, Sergey; Mikkonen, Tommi; Koucheryavy, Yevgeni. Multi-factor Authentication: survey. 05.01.2018. <https://www.mdpi.com/2410-387X/2/1/1/>. Luettu 8.5.2024. (Vieraskielinen verkkoaineisto).

18. Petcu, Adrian ym. A secure and decentralized authentication mechanism on web 3.0 and Ethereum blockchain technology. 15.12.2022. <https://www.mdpi.com/2076-3417/13/4/2231>. Luettu 4.4.2024. (Vieraskielinen artikkeli)

19. Ping Identity. Certificate based authentication. 2024. <https://www.pingidentity.com/en/resources/identity-fundamentals/authentication/certificate-authentication.html> Luettu 18.06.2023. (Vieraskielinen artikkeli verkkoaineistossa)

20. Rikard, Elsen; Muhammad, Rikza Nashrulloh; Ade, Sutedi. Wallet-Based Authentication on College Information System. 15.07.2022.

<https://www.researchgate.net/publication/367572129_WALLET-BASED_AUTHENTICATION_ON_COLLEGE_INFORMATION_SYSTEM>. Luettu 03.05.2024. (Vieraskielinen artikkeli verkkoaineistossa)

21. Single sign on. 08.2023. <https://www.logintc.com/types-of-authentication/single-sign-on-sso> Luettu 03.09.2024. (Vieraskielinen artikkeli verkkoaineistossa)

22. Sitra, Seuraavan sukupolven internet kohti web 3.0. 29.03.2022. <https://www.youtube.com/watch?v=o5NRbmAma_0> Katsottu 20.04.2024. (Videotallenne sitran tilaisuudesta)

23. Suomela, Susanna. Mikä on MFA eli monivaiheinen tunnistautuminen? <https://ajankohtaista.emce.fi/mik%C3%A4-on-mfa-eli-monivaiheinen-tunnistautuminen> Luettu 28.10.2023. (Verkkoartikkeli)

24. Tayyab, Mahmood. What is semantic web or web 3.0?. 07.08.2022, <https://medium.com/coinmonks/what-is-semantic-web-or-web-3-0-2bd4093811e8/>. Luettu 13.03.2024. (Vieraskielinen verkkoartikkeli)

25. Torus Labs Private Limited, Web3Auth documentation. 02.06.2023. <https://web3auth.io/docs/what-is-web3auth>. Luettu 12.03.2024. (Vieraskielinen verkkoartikkeli, jossa ei tiedossa tekijää)

26. Udoh, Benjamin. How does Google Authenticator work? 16.1.2023 < https://medium.com/nerd-for-tech/how-does-google-authenticator-work-2696fd5f0764>. Luettu 15.03.2024. (Vieraskielinen verkkoartikkeli)

27. Vapen, Anna. Web Authentication using Third Parties in Untrusted Environments. 2016. <https://www.diva-portal.org/smash/get/diva2:921172/FULLTEXT01.pdf>. Luettu 11.03.2024. (Vieraskielinen opinnäytetyö)

28. Wensheng Gan; Philip S. Yu. Web. 3.0: The Future of Internet. 04.2023. < https://www.researchgate.net/publication/369753421_Web_30_The_Future_of_Internet>. Luettu 20.05.2024. (Vieraskielinen pdf-tiedosto)

29. Wood, Gavin. Why do we need Web 3.0. 12.09.2018. <https://gavofyork.medium.com/why-we-need-web-3-0-5da4f2bf95ab>. Luettu 22.05.2024. (Vieraskielinen verkkoartikkeli)