Karelia-ammattikorkeakoulu
Tietojenkäsittely (AMK)

Vahvan tunnistautumisen käyttöönottosuunnitelma web-sovellukseen

Eetu Pulkkinen

Opinnäytetyö, huhtikuu 2024

www.karelia.fi

Karelia AMMATTIKORKEAKOULU	**OPINNÄYTETYÖ** **Huhtikuu 2024** **Tietojenkäsittelyn koulutus** Tikkarinne 9 80200 JOENSUU +358 13 260 600

Tekijä
Eetu Pulkkinen

Nimeke
Vahvan tunnistautumisen käyttöönottosuunnitelma web-sovellukseen

Toimeksiantaja
Kodia oy

Tiivistelmä

Opinnäytetyön aiheena oli käyttöönottosuunnitelman kehittäminen vahvan tunnistautumisen integroimiseksi web-sovellukseen. Vahva tunnistautuminen muodostaa olennaisen osan modernista tietoturvastrategiasta verkkopalveluissa erityisesti silloin, kun käsitellään arkaluonteisia tietoja.

Työssä tutkittiin erilaisia tunnistautumismenetelmiä ja niiden soveltuvuutta erilaisiin konteksteihin. Tarkastelun alla oli myös vahvan tunnistautumisen vaikutukset käyttäjäkokemukseen sekä sovelluksen tietoturvaan. Lopuksi esitettiin Telian vahvan tunnistautumisen integrointi web-sovellukseen React, JavaScript, TypeScript sekä GraphQL -ohjelmointikielillä. Toteutusvaiheessa keskityttiin yksityiskohtiin, kuten salausavaimen luontiin ja funktionaaliseen toteutukseen.

Lopputuloksena saimme vahvan tunnistautumisen liitettyä toimeksiantajan sovellukseen. Vahvan tunnistautumisen toteutuksessa pyrittiin luomaan mahdollisimman modulaarinen rakenne. Modulaarisen lähestymistavan ansiosta vahva tunnistautuminen on helposti skaalattavissa ja integroitavissa muihin sovelluksen osiin.

Kieli suomi	Sivuja 37 Liitteet 0 Liitesivumäärä 0

Asiasanat
käyttöönottosuunnitelma, vahva tunnistautuminen, web-sovellus, tietoturva, tunnistusmenetelmä, salausavain, modulaarinen lähestymistapa

Karelia
UNIVERSITY OF APPLIED SCIENCES

THESIS
April 2024
Degree Programme in Business Information Technology

Tikkarinne 9
80200 JOENSUU
FINLAND
+ 358 13 260 600

Author
Eetu Pulkkinen

Title
Strong Authentication Deployment Plan for a Web Application

Commissioned by
Kodia oy

Abstract

The topic of the thesis was the development of a deployment plan to integrate strong authentication into a web application. Strong authentication is an essential part of a modern security strategy for web services, especially when sensitive data is handled.

The work investigated different authentication methods and their suitability for different contexts. The impact of strong authentication on user experience and application security was also examined. Finally, the integration of Telia's strong authentication into a web application using React, JavaScript, TypeScript and GraphQL programming languages was presented. The implementation phase focused on details such as encryption key generation and functional implementation.

The end result was the integration of strong authentication into the client's application. The implementation of strong authentication was designed to be as modular as possible. This modular approach makes strong authentication easily scalable and integrable with other parts of the application.

Language
Finnish

Pages 37
Appendices 0
Pages of Appendices 0

Keywords
deployment plan, strong authentication, web application, security strategy, encryption key, modular implementation

Sisältö

Lyhenteet

2FA	Two-factor authentication
eSE	Embedded Secure Elements
FIDO	Fast Identity Online
GDPR	General Data Protection Regulation
JWK	JSON Web Key
JWT	JSON Web Token
NFC	Near Field Communication
OTP	One time password
PIN	Personal identification number
RBA	Risk based authentication
SMS	Short message service
TPM	Trusted Platform Modules
USB	Universal Serial Bus

1 Johdanto

Tietoturva on yksi tärkeimmistä tekijöistä ohjelmistokehityksessä niin ohjelmistokehittäjien kuin sovellusta käyttävien asiakkaiden näkökulmasta. Tässä opinnäytetyössä käyn läpi, mitä vahva tunnistautuminen sovelluksissa tarkoittaa erityisesti tietoturvan näkökulmasta. Ohjelmistokehityksessä vahvan tunnistautumisen menetelmien käyttöönotto on käyttäjien tunnistautumisessa keskeistä. Menetelmä on valittava ja toteutettava harkiten, sillä vahva tunnistautuminen voi heikosti toteutettuna huonontaa sovelluksen käyttäjäkokemusta merkittävästi.

Teknologia sekä tunnistautumisprosessit kehittyvät jatkuvasti, jonka vuoksi syntyy uusia tunnistautumistrendejä. Trendi voi olla esimerkiksi biometristen tunnistautumismenetelmien yleistyminen ja kehittyminen, jota nähdään esimerkiksi laajasti eri älylaitteiden lukitusmenetelmänä. Organisaatioiden on suotavaa olla tietoisia uusista trendeistä, jotta sovelluksen käyttäjäkokemus ja tietoturva olisivat mahdollisimman laadukkaita.

Euroopassa on käytössä EU:n yleinen tietosuoja-asetus (General Data Protection Regulation, GDPR), joka huolehtii käyttäjien henkilötiedoista. Tietosuoja-asetus suojelee siis käyttäjien tietoja esimerkiksi niiden väärinkäyttämiseltä ja tarjoaa käyttäjille oikeuksia heidän tietojaan koskien. Käyttäjillä on muun muassa oikeus poistattaa kaikki tietonsa rekisterinpitäjältä. Organisaatiot, jotka rikkovat tietosuoja-asetusta, voidaan tuomita muun muassa sakkoihin. Maakohtaiset lait voivat olla osittain tiukempia kuin tietosuoja-asetuksen linjaukset.

Tämä opinnäytetyö on tehty Kodia Oy:lle, joka kehittää ja ylläpitää vuokrauksenhallintajärjestelmää. Vuokrauksenhallintajärjestelmällä hallitaan asuntomassojen vuokrausta, esimerkiksi asuntohakemuksia ja vuokrasopimuksia sekä vuokralaisten vuokranmaksuja. Tällä hetkellä Kodialla ei ole vahvaa tunnistautumista käytössä. Työssä kartoitetaan, mihin sovelluksen osiin vahva tunnistautuminen olisi tarpeellinen. Tämän lisäksi toteutetaan vahvan tunnistautumisen prosessi ja liitetään se yhteen ominaisuuteen

sovelluksessa. On huomioitavaa, että työ käsittelee vahvaa tunnistautumista prosessina, eikä niinkään tunnistetun käyttäjän käyttöoikeuksia. Liitettyyn ominaisuuteen vaaditaan kuitenkin jatkossa vahva tunnistautuminen, mikäli asiakkaamme päättää ottaa palvelun käyttöön.

2 Vahvan tunnistautumisen perusteet

2.1 Vahva tunnistautuminen

Vahva tunnistautuminen on tietoturvakäytäntö, jolla voidaan myöntää käyttäjälle pääsy järjestelmään, tiliin tai palveluun vasta sitten, kun vähintään kaksi tekijää tai todennusmenetelmää yhdistetään. Tällä turvataan käyttäjän tunnistautuminen ja vähennetään esimerkiksi luvattoman pääsyn ja tietomurron riskejä. Tästä syystä vahva tunnistautuminen luokitellaankin yhdeksi tärkeimmistä tekijöistä tietoturvassa, erityisesti verkossa ja verkkopalveluissa. (Okta 2023.)

Vahva tunnistautuminen voidaan jakaa kolmeen eri tekijään, joista jokainen tekijä kattaa yhden suuren kokonaisuuden käyttäjän henkilöllisyydestä. Nämä kolme eri tekijää ovat tiedossa oloon perustuva todentamistekijä, hallussapitoon perustuva todentamistekijä ja luontainen todentamistekijä. Näitä kolmea tekijää sekoittamalla ja yhdistelemällä saadaan tulokseksi yhdistelmä, joka voidaan luokitella uniikiksi jokaiselle sovelluksen käyttäjälle. Yhdistelmää hyödyntäen voidaan todeta, että kyseessä on oikea henkilö. Tämän jälkeen voidaan myöntää pääsyoikeus sovellukseen, rakennukseen, huoneeseen tai mihin ikinä käyttäjä tunnistautuukaan. (Authy 2023.)

Tiedossa oloon perustuva todentamistekijä perustuu siihen, että käyttäjä tietää esimerkiksi salasanan tai tunnusluvun. Tiedossa oloon perustuva on näistä todentamistekijöistä yleisin. Tämä edellyttää sitä, että käyttäjällä on tiedossa oma salasana tai tunnusluku, jonka hän syöttää palvelulle. Hallussapitoon perustuva todennus käyttää nimensä mukaan hallussa olevia fyysisiä esineitä. Hallussa olevat esineet voivat olla esimerkiksi älypuhelin, älykortti tai avainlukulaite, joka vaaditaan tunnistautumisprosessissa. Luontainen

todentamistekijä käyttää käyttäjän fyysisiä ominaisuuksia. Fyysinen ominaisuus voi olla esimerkiksi sormenjälki, silmän verkkokalvo tai kasvot. Käyttäjä todennetaan katsomalla, onko hänen tunnistautumiseen vaadittu fyysinen ominaisuutensa yhtenevä kyseiselle käyttäjälle asetettuihin sisäänpääsyvaatimuksiin. Kaikki biometriset tunnistusmenetelmät perustuvat luontaiseen todentamistekijään. (Authy 2023.)

Esimerkiksi pankkipalveluissa tai sähköpostipalveluissa käyttäjät voivat käyttää vahvaa tunnistautumista varmistaakseen, että vain heillä on pääsy omille tileilleen. Tämä auttaa suojaamaan arkaluonteisia tietoja sekä estämään tietomurtoja.

2.2 Vahvan tunnistautumisen tärkeys verkkosovelluksessa

On monta pätevää syytä, miksi vahvan tunnistautumisen liittäminen esimerkiksi verkkosovelluksiin on todella tärkeää. Vahva tunnistautuminen liittyy suoraan tietoturvaan sekä käyttäjäkokemukseen. Vahvan tunnistautumisen voidaan katsoa tuovan yritykselle myös liiketoiminnallisia hyötyjä. Se kasvattaa asiakasluotettavuutta, jonka taas voidaan odottaa tuovan verkkosovellukselle lisää käyttäjiä. (WellDev 2022.)

Harvat verkkosovellukset menestyvät ilman kunnollista luottamusta ja hyvää mainetta. Vahva tunnistautuminen tuo luottamusta käyttäjille, kun heidän tietonsa ja henkilöllisyytensä ovat suojattuja. Myös Locken ja Zieglerin (2016) mukaan vahvan tunnistautumisen tuoma luottamus lisää asiakasuskollisuutta, jonka seurauksena voidaan saavuttaa uusia käyttäjiä käyttämään sovellusta.

Vahvan tunnistautumisen avulla voidaan tunnistaa käyttäjän henkilöllisyys. Tämä on todella tärkeää erityisesti silloin, kun käyttäjillä on mahdollisuus suorittaa kriittisiä toimia sovelluksessa. Kriittisiä toimia voivat olla esimerkiksi rahaliikenteeseen liittyvät toiminnot tai salassa pidettävien tietojen käyttö. Vahvan tunnistautumisen avulla varmistutaan siitä, että kyseisellä käyttäjällä on varmasti käyttöoikeus verkkosovellukseen. Sen ansiosta voidaan myös välttyä monelta haitalliselta hyökkäykseltä. Haitalliset hyökkäykset ovat muun muassa

tietomurtoja ja verkkosovelluksen luvatonta käyttöä. (Kyberturvallisuuskeskus 2023.)

3 Tunnistautumisen teknologiat

Tietoturvateknologiat vahvan tunnistautumisen osalta ovat monimuotoisia, tarjoten paljon erilaisia keinoja vahvistaa käyttäjien henkilöllisyys. Yleisimpiä vahvan tunnistautumisen teknologioita ovat kaksivaiheinen todennus (engl. 2FA, Two-factor authentication), biometrinen tunnistus, älykortit ja avainlukulaitteet, mobiilisovellukset sekä SMS-viestit.

3.1 Käytössä olevat vahvan tunnistautumisen teknologiat

Kaksivaiheinen todennus vaatii kahden eri tekijän (ks. kohta 2.1) käyttämistä ennen kuin voidaan myöntää pääsy tilille tai järjestelmään. Yleensä tämä käsittää salasanan (tiedossa oloon perustuva tekijä) sekä toisen tekijän, kuten yksilöllisen koodin. Kyseinen koodi voi olla esimerkiksi mobiilisovelluksessa tai lähetettynä käyttäjän puhelimeen tekstiviestillä (hallussapitoon perustuva tekijä). (Stanislav 2015.)

Biometrinen tunnistaminen hyödyntää käyttäjän fyysisiä ominaisuuksia tai käyttäytymiseen liittyviä piirteitä, kuten sormenjälkiä, silmän verkkokalvon rakennetta, kasvonpiirteitä, ääntä tai jopa kirjoitusasentoa. Tätä teknologiaa käytetään laajasti älypuhelimissa ja joissakin kannettavissa laitteissa. (Bhattacharyya, Ranjan, Alisherov & Choi 2009.)

Älykortit ja avainlukulaitteet ovat konkreettisia välineitä, joita käyttäjät hyödyntävät tunnistautuessaan. Ne saattavat sisältää mikrosirun tai muita tunnistetietoja. Tällaisia laitteita käytetään yleisesti yritystason järjestelmissä ja hallituissa ympäristöissä. (Blythe 2004.)

Mobiilisovellukset ovat keskeisessä asemassa vahvan tunnistautumisen käyttöönotossa. Niiden kautta voidaan tarjota erilaisia todennusmenetelmiä ja

strategioita, jotka takaavat käyttäjätietojen ja arkaluonteisiin tietoihin liittyvien tapahtumien turvallisuuden. (Nextauth 2023.)

SMS-viestien avulla käyttäjille voidaan lähettää vahvistuskoodi, jonka he syöttävät kirjautumisen yhteydessä. Tämä on yksi yleisimmistä kaksivaiheisen todennuksen menetelmistä. SMS-viestien käyttöä kannattaa kuitenkin arvioida ja käyttää viimeisenä vaihtoehtona ja vain tietyissä tilanteissa, sillä niiden jäännösriskit ovat niin merkittäviä. Jäännösriski on se riski, joka jää jäljelle vaikka kaikki mahdolliset varotoimenpiteet olisi otettu käyttöön riskin minimoimiseksi. (Child 2021.)

3.2 FIDO

FIDO (Fast IDentity Online) -alliance on vuonna 2013 perustettu voittoa tavoittelematon organisaatio, jonka keskeisenä tavoitteena on vähentää käyttäjien todentamisen riippuvuutta salasanoista. FIDO-allianssin kehittämä FIDO-todennus perustuu julkisen avaimen salaukseen ja tarjoaa turvallisen käyttäjäkokemuksen, joka on kestävä tietojenkalasteluyrityksiä vastaan. (FIDO Patentti 2019 & FIDO 2023.)

FIDO tukee monipuolisesti erilaisia todennustekniikoita, kuten biometriikkaa (ks. kohta 3.1.2), sekä olemassa olevia ratkaisuja ja viestintästandardeja. Näitä ovat muun muassa luotettava alustamoduuli (Trusted Platform Modules, TPM) - turvapiiri jota käytetään tietokoneen suojauksen parantamiseen esimerkiksi salausavainten turvalliseen luontiin sekä tallentamiseen (Microsoft 2024.), USB-turvapoletit jotka liitetään tietokoneen USB-porttiin ja toimivat tämän jälkeen fyysisenä avaimena joka vaaditaan tiettyihin tietoihin tai järjestelmiin (Givens 2021.), upotettu turvallinen elementti (embedded Secure Element, eSE) joka varmistaa tietojen turvallisen tallennuksen, antaa tietoja vain valtuutetuille sovelluksille ja henkilöille sekä suorittaa kryptografisia toimintoja joita ovat esimerkiksi todennus ja salaus (Thales 2024.) sekä älykortit ja lähikenttäviestintä (Near Field Communication, NFC) joka on lyhyen kantaman langaton teknologia, mahdollistaen tiedon tai virran vaihdon kahden NFC-yhteensopivan laitteen välillä (Wikipedia 2024.) (FIDO 2023).

4 Vahva tunnistautuminen tietoturvan näkökulmasta

4.1 Vahva tunnistautuminen ja käyttäjäkokemus

Vahvan tunnistautumisen suunnittelussa on olennaista yhdistää tietoturva ja käyttäjäkokemus. Vaikka vahva tunnistautuminen on suunniteltu suojaamaan arkaluonteisia tietoja ja estämään luvattoman pääsyn järjestelmiin, se ei saisi huonontaa käyttäjäkokemusta. (Orr 2019.)

Kun valitaan tunnistusmenetelmiä, on olennaista kiinnittää huomiota niiden käyttäjäystävällisyyteen. Esimerkiksi biometriset tunnistusmenetelmät, kuten sormenjäljen tai kasvojen tunnistus, tarjoavat monesti vahvan tietoturvan ja ovat usein helppokäyttöisiä käyttäjille. (Tucakov 2023). Yksikäyttösalasanatkin (engl. OTP, One time password) voivat olla nopea ja tehokas tapa vahvistaa tunnistus ilman monimutkaisia salasanoja. (Virgillito 2019.)

Vaikka vahva tunnistautuminen saattaakin vaatia hieman enemmän aikaa käyttäjiltä, sen tulisi silti olla sujuva prosessi. Tietoturvallisuus on keskeistä, mutta käyttäjäkokemuksen parantaminen varmistaa myös sen, että vahvan tunnistautumisen edut tulevat mahdollisimman tehokkaasti hyödynnettyä. (Beyond Identity 2021.)

Voidaan siis todeta, että käyttäjäkokemus muodostaa olennaisen osan tunnistautumisprosessia. Selkeät ohjeet ja käyttöliittymä tekevät tunnistautumisesta lähes vaivatonta käyttäjille. Lisäksi on tärkeää tarjota vaihtoehtoja, kuten erilaisia tunnistusmenetelmiä tai vaihtoehtoisia kirjautumistapoja niissä tilanteissa, joissa käyttäjät esimerkiksi unohtavat salasanansa.

4.2 Vahvan tunnistautumisen vaikutus tietoturvaan

Perinteiseen tunnistautumiseen verrattuna, jossa siis käytetään vain salasanaa tai PIN-koodia, vahvan tunnistautumisen käyttöönotto parantaa sovelluksen tietoturvaa huomattavasti. Vahvan tunnistautumisen ansiosta voidaan muun

muassa suodattaa pois joitakin tunnettuja hyökkäystapoja ja niistä johtuvia haittoja. Esimerkiksi, jos hyökkääjä saisi tietoonsa käyttäjän salasanan tai PIN-koodin, ei hän silti pystyisi kirjautumaan palveluun. Salasanan tai PIN-koodin lisäksi hänellä pitäisi olla käyttäjän hallussapitoon perustuva (usein matkapuhelin tai kertakäyttösalasanat) tai luontainen todentamistekijä (usein kasvojen tunnistus tai sormenjälki) (Okta 2023.)

Lisäksi vahva tunnistautuminen voi auttaa suojaamaan käyttäjiä sosiaalisen manipulaation (engl. social engineering) hyökkäyksiltä. Tällaiset hyökkäykset pyrkivät huijaamaan käyttäjiä jakamaan henkilökohtaisia tietojaan tai salasanoja. Vahvan tunnistautumisen käyttö vähentää hyökkäyksien onnistumisen mahdollisuutta, sillä hyökkääjän on huomattavasti vaikeampaa saada kaikki tarvittavat tiedot itselleen, jotta pääsisi järjestelmään. (Fremery 2021.)

Kaiken kaikkiaan vahva tunnistautuminen muodostaa tehokkaan suojan monia tietoturvariskejä vastaan. Se lisää turvallisuuden tasoa huomattavasti ja auttaa pitämään salassa arkaluontoisia tietoja sekä organisaation että käyttäjien tasolla.

4.3 Kehittyvät trendit vahvassa tunnistautumisessa

Riskiperusteinen tunnistautuminen (engl. RBA, Risk-Based Authentication) on turvallisuusominaisuus, joka lisää ylimääräisen suojakerroksen korkean riskin tilanteissa. Se tunnistaa epätavalliset kirjautumisyritykset ja ottaa käyttöön tiukemmat tunnistautumistoimenpiteet. RBA analysoi erilaisia tekijöitä, kuten kirjautumisyrityksiä, laitteen sijaintia ja käyttäytymismalleja. Se pystyy havaitsemaan botti-iskut ja haittaohjelmilla saastuneet laitteet varmistaen, että vain lailliset käyttäjät pääsevät verkkoon. RBA oli merkittävä tunnistautumistrendi vuonna 2023. (ZippyOPS 2023.)

Pankki- ja finanssisektorin laajetessa kosketusmaksutavat yleistyvät, mikä asettaa haasteita tietoturvallisuudelle. Yhtenä haasteena esiintyy identiteettivarkaudet, sillä kosketusmaksutavat liittyvät usein henkilökohtaisiin

maksuvälineisiin. Käyttäjä voi altistua identiteettivarkaudelle, mikäli maksuvälineet joutuvat vääriin käsiin. Perinteinen salasanatunnistautuminen on huomattavasti vähemmän luotettava lisääntyvien verkkouhkien vuoksi. Salasanaton tunnistautuminen vähentää riskejä, kuten tilinpetoksia ja haltuunottoja, mikä tekee siitä ihanteellisen tunnistautumisvaihtoehdon tälle alalle. Se auttaa hallitsemaan käyttäjien tunnistamista ja vähentämään erilaisten hyökkäysten riskejä. (ZippyOPS 2023.)

Ennusteiden mukaan lähiaikoina odotetaan lohkoketjuun (engl. blockchain) perustuvien digitaalisten tunnisteiden yleistyvän. Tämä johtuu esimerkiksi hajautettujen teknologioiden kasvavasta suosiosta sekä enenevästä tarpeesta turvallisille tunnistepalveluille. Vuonna 2022 Apple ja Samsung olivat ainoat lohkoketjuun perustuvien tunnisteiden tarjoajat, mutta nyt yhä useammat yritykset tarjoavat kyseisiä palveluja, mikä antaa kuluttajille enemmän vaihtoehtoja. Tämä mahdollistaa käyttäjille identiteettipalveluntarjoajien valitsemisen ja itsensä todentamisen digitaalisilla tunnisteilla ilman toistuvaa tunnistautumista. (ZippyOPS 2023.)

Fyysisiä ominaisuuksia hyödyntävä biometrinen tunnistautuminen on kasvattanut suosiotaan viime vuosina. Käyttäjän biologisten ominaisuuksien käyttö tarjoaa turvallisen ja saumattoman käyttökokemuksen, mikä vaikeuttaa tietojärjestelmiin murtautuvien henkilöiden pääsyä arkaluontoisiin tietoihin. Monet yritykset pyrkivätkin ottamaan biometrisen tunnistautumisen teknologian käyttöön, jotta he saavuttaisivat optimaalisen turvallisuustason. (ZippyOPS 2023.)

Voidaan katsoa, että vuonna 2023 painopiste tunnistautumisen kehittämisessä oli turvallisissa verkkokauppatapahtumissa sekä yhä turvallisemmassa käyttäjäkokemuksessa. Tämä sisälsi uusien menetelmien tutkimista riskiperusteisen tunnistautumisen osalta, minkä uskottiin esimerkiksi johtavan parempiin liiketoiminnan tuloksiin. (ZippyOPS 2023.)

5 Yleinen tietosuoja-asetus (GDPR)

5.1 Kuinka vahva tunnistautuminen edistää GDPR:n toteutumista

Vahvan tunnistautumisen liittäminen web-sovellukseen voi edistää GDPR:n toteutumista monin tavoin. Se auttaa varmistamaan käyttäjien oikeuden käsitellä esimerkiksi henkilötietoja, joka vähentää tietomurtojen riskiä sekä suojaa käyttäjien yksityisyyttä. Se myös vähentää riskiä tietojen väärinkäytöstä, ja tietoturva on olennaista GDPR:n vaatimusten täyttämiseksi. GDPR korostaa käyttäjän suostumuksen merkitystä henkilötietojen käsittelyssä. Vahva tunnistautuminen voi olla osana suostumuksen antamista, kun käyttäjä hyväksyy tietojen keräämisen tunnistautuessaan. Se myös vahvistaa tietojen läpinäkyvyyttä, kun tunnistautumisprosessissa mainitaan mitä tietoja heiltä käytetään ja mihin tarkoitukseen.

5.2 GDPR:n perusteet

5.2.1 Mikä on GDPR?

Yleinen tietosuoja-asetus (GDPR, General Data Protection Regulation) on Euroopan unionin säädös, joka astui voimaan 25. toukokuuta 2016 ja alettiin soveltaa EU:n jäsenvaltioissa 25. toukokuuta 2018. Yleinen tietosuoja-asetus on suunniteltu suojaamaan esimerkiksi ihmisen yksityisyyttä ja antamaan verkkopalveluiden käyttäjille paremman hallinnan omista henkilötiedoistaan. (Eduskunta 2023.)

5.2.2 GDPR:n tausta ja tarkoitus

Ennen GDPR:n voimaantuloa tietosuojalainsäädäntö oli hajanaista ja vaihteli eri EU-maiden välillä. Tämä loi haasteita yrityksille, jotka toimivat useissa maissa. Lisäksi teknologian nopea kehitys ja digitalisaatio lisäsivät tietosuojaan liittyviä riskejä, kuten tietomurtoja ja tietovuotoja. (European Data Protection Supervisor 2023.)

Euroopan unioni halusi yhtenäistää eri maiden tietosuojakäytäntöjä ja vahvistaa käyttäjien tietosuojaoikeuksia. Tämä johti GDPR:n luomiseen, joka korvasi aikaisemman Tietosuojadirektiivin vuodelta 1995. (European Data Protection Supervisor 2023.)

5.2.3 Miten GDPR vaikuttaa henkilötietojen käsittelyyn?

GDPR vaikuttaa henkilötietojen käsittelyyn monin tavoin. Se määrittelee tiukat säännöt henkilötietojen käsittelylle erityisesti lasten tietojen osalta. Organisaatioiden on selkeästi määriteltävä henkilöiden tietojen käsittelyn tarkoitus ja kerättävä heiltä ainoastaan tarpeelliset tiedot. (GDPR 2023.)

GDPR vahvistaa käyttäjien oikeuksia heidän tietoihinsa, mukaan lukien oikeuden saada tietoa, korjata virheet ja pyytää omien tietojensa poistamista. Se edellyttää myös organisaatioita parantamaan tietoturvakäytäntöjään ja ottamaan enemmän vastuuta henkilötietojen suojelussa. (GDPR 2023.)

Lisäksi GDPR sisältää rangaistuksia niille organisaatioille, jotka eivät noudata sen vaatimuksia, mikä rohkaisee organisaatioita noudattamaan tietosuojasääntöjä tiukasti. Kaiken kaikkiaan GDPR korostaa yksityisyyden suojaa ja organisaatioiden vastuuta henkilötietojen käsittelyssä. (GDPR 2023.)

5.3 GDPR:n tärkeimmät säädökset

5.3.1 Henkilötietojen määritelmä ja käsittelyn periaatteet

Henkilötiedot ovat kaikenlaisia tietoja, jotka liittyvät tunnistettuun tai tunnistettavissa olevaan luonnolliseen henkilöön, tarkoittaen rekisteristä löydettävissä olevaa henkilöä. Tunnistettavissa oleva henkilö on henkilö, joka voidaan suoraan tai epäsuorasti tunnistaa esimerkiksi nimen, henkilötunnuksen, sijaintitiedon, verkkotunnisteen tai fyysisen, geneettisen, psyykkisen, taloudellisen, kulttuurillisen tai sosiaalisen tekijän perusteella. (Information Commissioner's Office 2023.)

CLARINin (2023) mukaan henkilötietojen käsittelyn periaatteita ovat lainalaisuus, rehellisyys ja avoimuus, käyttötarkoituksen rajoitus, tietojen minimointi ja tarpeellisuus, tietojen tarkkuus, säilytysajan rajaus sekä eheys ja luottamuksellisuus.

GDPR linjaa tietojen käsittelyn olevan laillista, kun se perustuu tiettyihin laillisiin perusteisiin, joista merkittävin on suostumus. Suostumus edellyttää vapaaehtoista, tiedotettua ja yksiselitteistä ilmaisua rekisteröidyn tahdosta tietojen käsittelyyn, joka voi olla joko nimenomainen tai oletettu suostumus. Nimenomainen on suostumus, jossa rekisteröity ilmaisee suostumuksensa yksiselitteisesti. Oletettu on suostumus, joka oletetaan annetuksi tietyissä olosuhteissa, mutta se ei ole yhtä yksiselitteinen tai dokumentoitu kuin nimenomainen suostumus. Alaikäisten tietojen käsittely vaatii joko alaikäisen suostumuksen tai huoltajan valtuutuksen, eikä alaikäisen suostumuksen ikäraja voi olla alle 13 vuotta. Herkkien tietojen käsittelyssä suostumuksen täytyy aina olla nimenomainen. Vaihtoehtoisesti käsittely voi perustua muihin laillisiin perusteisiin, mutta tämä vaatii huolellisen etujen tasapainottelun, ottaen huomioon muun muassa käsiteltävien tietojen luokan, rekisteröidyn kohtuulliset odotukset sekä vaadittavat suojatoimet, kuten pseudonymisoinnin. (CLARIN 2023.)

GDPR:ssä ei ole tarkkaa määritelmää "rehellisyydelle", joten sitä tulisi tulkita yleisenä oikeuden ja reiluuden käsitteenä. Käsittely ei ole rehellistä, jos se tapahtuu tavalla, joka voisi johtaa henkilöä harhaan tai uhata hänen yksityisyyttään, edes hänen suostumuksellaan. Tämä periaate pyrkii vähentämään tietojen vilpillistä hyödyntämistä GDPR:n kehyksessä. (CLARIN 2023.)

Avoimuuden periaatteen mukaan rekisteröidylle henkilölle tulee antaa tietoa omien henkilötietojensa käsittelystä riippumatta siitä, mikä oikeudellinen perusta käsittelylle on asetettu. Tietoa tulee antaa myös riippumatta siitä, ovatko tiedot kerätty suoraan häneltä vai hankittu muilla tavoin, esimerkiksi verkkosivujen

kautta tehtävällä tiedonhaulla. Tiedon tulee olla helposti saatavilla ja ymmärrettävää, selkeää ja myös selkeällä kielellä selitettynä. (CLARIN 2023.)

Henkilötietoja on käsiteltävä CLARINin (2023) mukaan "määritellyillä, nimenomaisilla ja laillisilla tarkoituksilla". Tämä tarkoittaa sitä, että tietojen käsittelyn tarkoitus on määriteltävä ennen käsittelyn aloittamista ja tätä tarkoitusta on noudatettava koko henkilötietojen elinkaaren ajan. Vaikka tieteellisen tutkimuksen tarkoituksia ei aina pystytä täysin määrittämään tiedonkeruun yhteydessä, GDPR mahdollistaa sen, että henkilöt voivat myöntää suostumuksensa tietyille tieteellisen tutkimuksen osa-alueille, kunhan ne ovat tunnustettujen eettisten standardien mukaisia. On kuitenkin huomioitava, että jo tiedonkeruuhetkellä tutkimuksella tulee olla selkeästi määritelty tarkoitus, joka voi myöhemmin myös muuttua. Muutoksista täytyy ilmoittaa rekisteröityneelle henkilölle. (CLARIN 2023.)

Kun tiedonkeruun tarkoitus on määritelty, on kiellettyä käsitellä tietoja sellaiseen tarkoitukseen, joka ei ole yhteensopiva alkuperäisen tarkoituksen kanssa. GDPR suo poikkeuksen sallien tieteellisen tutkimuksen aina yhteensopivana tarkoituksena. Tämä tarkoittaa, että esimerkiksi tilastointia tai kirjanpitoa varten laillisesti kerätyt tiedot voidaan uudelleenkäyttää tieteellisessä tutkimuksessa. Käyttö edellyttää, että kaikkia muita tietosuojaperiaatteita kunnioitetaan. (CLARIN 2023.)

Tietojen minimointiperiaate edellyttää, että henkilötietojen kerääminen, säilyttäminen ja käsittely rajoitetaan olennaiseen ja tarpeelliseen suhteessa käsittelyn tarkoitukseen. Tämä tarkoittaa, että sellaisia tietoja, joita ei tarvita tavoitellun päämäärän saavuttamiseksi, ei saa kerätä, säilyttää tai käsitellä laillisesti. Tämä periaate asettuu usein ristiriitaan tietoja käsittelevien intensiivisten toimintojen kanssa, joissa jokainen tieto voi olla arvokas, mutta vain harva on todella tarpeellinen tavoitellun päämäärän saavuttamiseksi. (CLARIN 2023.)

Henkilötietojen tulee olla CLARINin (2023) mukaan "tarkkoja ja tarvittaessa ajan tasalla". Tämä periaate liittyy kiinteästi henkilön oikeuteen tarkastaa ja

tarvittaessa korjata omat henkilötietonsa. Se on myös linjassa tutkimusyhteisön hyväksymien eettisten standardien ja parhaiden käytäntöjen kanssa. (CLARIN 2023.)

Tietojen säilyttämisen rajoittamisen periaate edellyttää, että henkilötietoja säilytetään vain niin kauan kuin se on tarpeen niiden alkuperäisiä käyttötarkoituksia varten. Tiedot tulee säilyttää sellaisessa muodossa, joka mahdollistaa tietojen kohtuullisen tunnistamisen. On olemassa useita maa- ja toimialakohtaisia rajoituksia, jotka määrittävät sallitun ajan henkilötietojen säilyttämiseen tiettyä tarkoitusta varten. Esimerkiksi tietyt sopimukset voivat usein säilyä alkuperäisessä, ei-anonymisoidussa muodossaan viisi vuotta niiden voimassaolon päättymisen jälkeen. (CLARIN 2023.)

Kun tiedot ovat käytössä pelkästään tieteellisiin tutkimustarkoituksiin ja asianmukaisia turvatoimia on otettu käyttöön, esimerkiksi tunnistautumisprosessi, henkilötietoja voidaan säilyttää pidempään GDPR:n mukaan, mutta kansalliset lait voivat kuitenkin olla tiukempia. (CLARIN 2023.)

Henkilötietoja on käsiteltävä siten, että niiden asianmukainen turvallisuus taataan. Tähän sisältyy suojelu luvattomalta tai lainvastaiselta käsittelyltä sekä suoja vahingoilta, eli esimerkiksi vahingossa tapahtuvalta henkilötiedon menetykseltä, tuhoutumiselta tai vaurioitumiselta. Tätä tarkoitusta varten on otettava käyttöön asianmukaisia teknisiä tai byrokraattisia toimenpiteitä. Nämä turvatoimet ovat aina välttämättömiä henkilötietojen käsittelyssä. Lisäksi tietyissä erityistapauksissa saattaa olla tarpeen noudattaa korkeampaa turvallisuustasoa. (CLARIN 2023.)

5.3.2 Rekisterinpitäjän ja käsittelijän velvollisuudet

GDPR asettaa selkeät velvollisuudet rekisterinpitäjille ja -käsittelijöille henkilötietojen käsittelyssä. Rekisterinpitäjän vastuulla on varmistaa, että henkilötietojen käsittely tapahtuu lainmukaisesti, rehellisesti ja avoimessa suhteessa rekisteröityihin. Rekisterinpitäjän on myös kerättävä vain ne tiedot, jotka ovat tarpeellisia tarkoituksen toteuttamiseksi ja varmistettava, että ne

säilytetään vain niin kauan kuin on tarpeen. (Tietosuojavaltuutetun toimisto 2023.)

Käsittelijän rooli on toteuttaa rekisterinpitäjän antamia ohjeita ja tehdä niin ainoastaan sopimuksen puitteissa. Käsittelijän on myös huolehdittava tietoturvan asianmukaisesta toteuttamisesta ja varmistettava henkilötietojen luottamuksellisuus. Käsittelijä on myös velvollinen tukemaan rekisterinpitäjää GDPR:n vaatimusten noudattamisessa ja avustamaan esimerkiksi tietoturvakysymyksissä. (Tietosuojavaltuutetun toimisto 2023.)

5.3.3 Yksilön oikeudet GDPR:n mukaan

GDPR myöntää verkkopalveluiden käyttäjille monia oikeuksia omien henkilötietojensa suhteen. Käyttäjillä on muun muassa oikeus tietää, käsitelläänkö heidän tietojaan ja mihin tarkoituksiin. Heillä on myös oikeus pyytää kopio omista tiedoistaan sekä vaatia mahdolliset virheet tai puutteet korjattaviksi. (Tietosuojavaltuutetun toimisto 2023.)

Käyttäjillä on myös oikeus pyytää tietojensa poistamista tietyissä tilanteissa tai rajoittaa niiden käsittelyä. He voivat myös erityistapauksissa vastustaa tietojensa käsittelyä. Käyttäjän halutessa siirtää tietonsa toiseen organisaatioon tai palveluun, on hänellä oikeus saada tiedot helposti itselleen. (Tietosuojavaltuutetun toimisto 2023.)

Käyttäjillä on oikeus myös vastustaa automaattista päätöksentekoa, mukaan lukien profilointi, joka voi vaikuttaa heihin ja heidän tietoihinsa merkittävästi. Nämä oikeudet ovat asetettu varmistamaan, että käyttäjät voivat hallita omia henkilötietojaan ja että niitä käsitellään oikeuden- ja lainmukaisesti. Rekisterinpitäjien velvollisuutena on vastata näihin pyyntöihin asianmukaisesti ja mahdollisimman pian. (Tietosuojavaltuutetun toimisto 2023.)

6 Vahvan tunnistautumisen käyttöönotto

Vahva tunnistautuminen sovelluksessa edellyttää tunnistautuvan henkilön yhdistämistä oikeutettuun käyttäjään. Käytännössä tunnistautuminen edellyttää lainsäädännön suojaamien henkilötietojen käsittelyä, joihin käyttäjä antaa luvan tunnistautuessaan. Tästä johtuen en voi tässä kappaleessa kertoa kaikkia yksityiskohtia toteuttamastani vahvasta tunnistautumisesta sovelluksessa.

6.1 Sovelluksen kartoitus

Aivan ensimmäiseksi täytyy lähteä kartoittamaan sovellusta, minne vahva tunnistautuminen halutaan. Kartoittaminen alkaa siten, että käyn sovelluksen ominaisuuksia läpi. Tämän jälkeen mietimme teknologiajohtajamme kanssa, onko vahva tunnistautuminen tarpeellinen kyseessä olevaan ominaisuuteen. Koska sovelluksemme on laaja, vahva tunnistautuminen on tarpeellinen vain niissä sovelluksen ominaisuuksissa, joissa se on välttämätöntä tietoturvan näkökulmasta. Vahvan tunnistautumisen käyttö muissa ominaisuuksissa voi vaikuttaa sovelluksen käyttäjien käyttökokemukseen haitallisesti.

Alustavan kartoituksen perusteella liitämme vahvan tunnistautumisen ainakin kolmeen sovelluksemme toimintoon. Nämä kolme ominaisuutta ovat tilitys, sähköinen irtisanominen ja hakemuksen hyväksyminen. Tilitykset ovat kriittinen toimenpide sovelluksessamme, sillä tässä toimenpiteessä asiakkaamme tililtä lähtee tilitysten summan verran rahaa. Meille on tärkeää, että vain asiakasyrityksen haluamat henkilöt voivat tehdä kyseisen toiminnon: tämä tarkoittaa, että toimintoa ei voi suorittaa jokainen asiakkaamme sovelluksen pääkäyttäjä.

Sähköinen vuokrasopimuksen iritsanominen on toiminto, jossa vahva tunnistautuminen on merkittävässä asemassa. Virallinen sopimuksen irtisanominen on lopullinen toiminto, joka suoritetaan vuokralaisen toimesta. Vahvan tunnistautumisen ansiosta voimme olla varmoja, että irtisanominen suoritetaan loppuun asti vain silloin, kun sopimusta irtisanova henkilö on todennettavissa oikeutetuksi käyttäjäksi.

Vahva tunnistautuminen on tärkeä ominaisuus myös esimerkiksi asuntohakemuksissa, sillä hakemukset ovat henkilökohtaisia, jolloin saadaan todennettua hakemuksen aitous. Vahva tunnistautuminen tuo myös sitoutumisen takeen: todennetuilla henkilötiedoilla tehty tunnistautuminen luo oikeutetun käyttäjän hyväksynnän sopimukseen ja sen ehtoihin.

6.2 Palveluntarjoajan valitseminen

Vahvan tunnistautumisen palveluntarjoajia on Suomessa paljon. Minun työtäni helpotti se, että teknologiajohtajamme oli jo tutkinut saatavilla olevia tunnistautumispalveluita ja kokeillut erityisesti Telian tunnistautumispalvelua. Kokeilun seurauksena Telian tunnistautumispalvelu oli valittu sovelluksessa käytettäväksi tunnistautumispalveluksi. Tehdyn pohjatyön avulla käytössäni oli alustava dokumentaatio, joka sisälsi tarvittavat tiedot testiympäristöstä ja testikäyttäjästä. Edellä mainittu ennakkotyö edesauttoi minun osuuttani tunnistautumisen käyttöönotossa, sillä yhteydenpidon aloittaminen Telian kanssa olisi vienyt minulta runsaasti työaikaa.

6.3 Telian tunnistautumispalvelu

Käyttäjän tunnistautuminen alkaa siitä, että hänet uudelleenohjataan Kodian sovelluksesta Telian tunnistautumisportaaliin. Suomessa Telian tunnistautumisessa on monta muotoa. Se tukee pankkitunnistusta kaikilta suomen pankeilta, mobiilivarmennetta kaikilta suomen operaattoreilta sekä henkilökorttia. Tämä tarkoittaa sitä, että käyttäjä voi tunnistautua esimerkiksi pankkitunnuksillaan riippumatta siitä, mikä pankki hänellä on käytössä. Käyttäjän ei siis tarvitse rekisteröityä mihinkään ulkoiseen palveluun tunnistautumista varten.

Kun käyttäjä tunnistautuu valitsemallaan menetelmällä, hän antaa Kodian sovellukselle suostumuksen käyttää tunnistautumispalvelun kautta saatuja henkilötietoja. Näitä henkilötietoja hyödyntäen voidaan varmistaa, että tunnistautunut henkilö on oikeutettu tekemään haluttu operaatio. Kun henkilö on

tunnistautunut, vahva tunnistautuminen on voimassa 60 minuuttia Kodian sovelluksessa. Tämän aikana hänen ei tarvitse tunnistautua uudelleen, eli hän voi suorittaa operaatioita niin kauan, kunnes 60 minuuttia on kulunut ja hänen täytyy suorittaa vahva tunnistautuminen uudelleen.

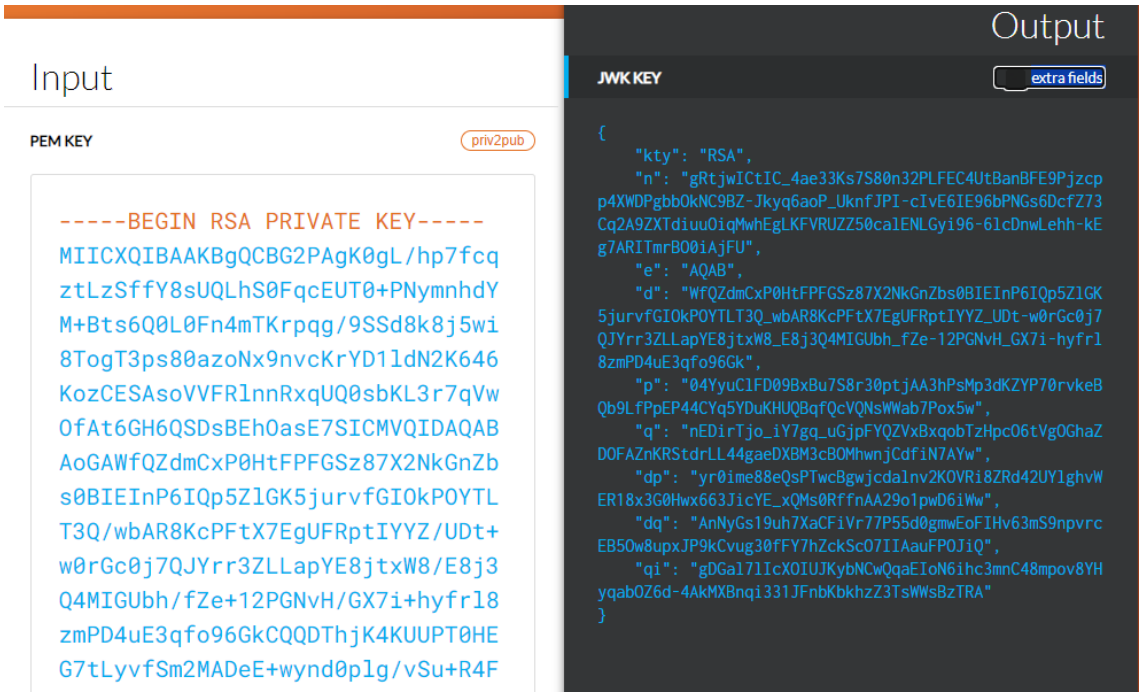
6.4 Testikäyttäjän luonti

Jotta vahvaa tunnistautumista voidaan testata, täytyy Telialta saada testikäyttäjän metatiedot (engl. metadata). Jotta metatiedot saadaan, pitää heille toimittaa JWT (JSON Web Token) sekä halutut uudelleenohjausosoitteet (engl. redirect uri). JWT on turvallinen tapa välittää tietoa osapuolien välillä. JWT:n tiedot voidaan todentaa sekä niihin voidaan luottaa sillä se on digitaalisesti allekirjoitettu. Uudelleenohjausosoitteet ovat niitä osoitteita, joihin käyttäjän halutaan uudelleenohjautuvan. Huomionarvoista on, että tuotannon osoitteen lisäksi on hyvä toimittaa myös testipalvelimen osoite sekä localhost, eli kehitettäessä käytettävä osoite. Osoitteiden avulla uudelleenohjausta voidaan testata tehokkaammin myös kehittäjän tietokoneella tai testipalvelimella.

JWT täytyy luoda itse. JWT:n luomista varten tehdään ensiksi yksityinen avain (engl. private key) PEM tiedostona terminaalin kautta macOS:ssä komennolla ”ssh-keygen -t rsa -b 4096 -m PEM -f [tiedostonimi].key”. Tämän jälkeen komentokehote kysyy salasanaa, mutta on huomioitavaa, ettei salasanaa ole välttämätöntä antaa. Suositeltua kuitenkin on syöttää salasana, jolloin se täytyy kirjoittaa kahdesti komentokehotteeseen. Tämä komento luo tietokoneelle kaksi tiedostoa. Tiedostojen nimet on [tiedostonimi].key ja [tiedostonimi].key.pub. Avain, jossa on pelkästään .key, on yksityinen avain. Tämä avain on tärkeää pitää muistissa, eikä sitä tule levittää eteenpäin. Toinen tiedosto, jonka perässä on .pub pääte, on julkinen avain. Kyseinen tiedosto ei kuitenkaan ole oleellinen meidän toteutuksemme kannalta, sillä luomme toisenlaisen julkisen avaimen käyttäen yksityistä avainta.

Seuraavaksi täytyy generoida RSA-tyyppinen julkinen avain. RSA on laajasti käytetty salausalgoritmi julkisille avaimille, jota käytetään paljon esimerkiksi

elektronisissa kaupankäynneissä. Myös RSA luodaan komentokehotteen kautta. Tämän jälkeen avainten luojan tulee komentokehotteessa avata se kansio, jossa aiemmin luotu yksityinen avain sijaitsee. Tämän jälkeen luodaan RSA-tyyppinen julkinen avain komennolla ”openssl rsa -in [tiedostonimi].key -pubout > [toinentiedostonimi].pub”. Tämän jälkeen [toinentiedostonimi].pub niminen tiedosto on löydettävissä samasta kansiosta. Se on julkinen avain, joka lopuksi käännetään JWK (JSON Web Key) muotoon, joka voidaan toimittaa Telialle. Julkisen avaimen saa käännettyä JWK muotoon osoitteessa https://irrte.ch/jwt-js-decode/pem2jwk.html.



Kuva 1. Vasemmalle liitetään teksti PEM-tiedostosta kokonaisuudessaan. Oikealla puolella on tarpeellinen JWK-avain joka on hyödyllistä ottaa talteen.

Kyseinen JWK-avain toimitetaan sähköpostitse Telialle. Sähköpostiin on hyvä liittää myös halutut uudelleenohjausosoitteet, jotta testikäyttäjä on heti valmis käyttöönotettavaksi. Uudelleenohjausosoitteet voivat olla esimerkiksi https://hyväsovellus.fi/tunnistautuminen ja testipalvelimelle esimerkiksi https://dev.hyväsovellus.fi/tunnistautuminen. Osoitteessa on tärkeää olla tunnistettava polku, esimerkiksi ”/tunnistautuminen”. Tähän polkuun luodaan haluttu tapahtuma silloin, kun tälle sivulle ohjaudutaan. Tunnistautumisen

jälkeen käyttäjä ohjataan ennalta määrätylle sivulle, joten kyseiseen osoitteeseen voidaan tehdä käyttöliittymä tai tapahtuma.

Kun Telia vastaa sähköpostiin ja vahvistaa, että käyttäjä on luotu, Telia lähettää samassa sähköpostissa aiemmin mainitut metatiedot. Metatiedot sisältävät muun muassa asiakastunnuksen (engl. client id), jota tarvitaan pyyntöjä luodessa. Tästä lisää tietoa myöhemmin. Sähköpostissa toimitetaan myös Telian integraatio-ohje, josta löytyvät tarvittavat päätepisteet (engl. endpoint) sekä muita hyödyllisiä ja tärkeitä tietoja. Ohjeen alussa on mainittu jokaisen eri pankin testikäyttäjät, joilla tunnistautuminen voidaan suorittaa. Kun tunnistautumisprosessi suoritetaan, on tärkeää huomioida, että kirjautumistiedot täytetään dokumentista löytyvillä valitun pankin käyttäjän tiedoilla. Dokumentista löytyy myös testiympäristön osoite (engl. preproduction url), minne pyynnöt tulee tehdä.

6.5 Vahvan tunnistautumisen tapahtumaketju

Vahvan tunnistautumisen tapahtumaketju (engl. flow) on seuraavanlainen:

1. Haetaan tunnistautumisosoite Telialta
2. Ohjataan käyttäjä haettuun tunnistautumisosoitteeseen
3. Onnistuneesta tunnistautumisesta kerätään talteen tunnistuskoodi (engl. authorization code), ja epäonnistuneesta osoitetaan käyttäjälle virheilmoitus
4. Tunnistautumisen jälkeen Telia uudelleenohjaa käyttäjän Telialle aiemmin toimitettuun uudelleenohjausosoitteeseen.
5. Haetaan Telialta tunnistuskoodia vastaava pääsykoodi (engl. access token)
6. Haetaan käyttäjän henkilötiedot käyttäen pääsykoodia
7. Kerätään tarvittavat tiedot muistiin ja tarkistetaan, onko kyseinen henkilö oikeutettu tietojen perusteella suorittamaan kyseinen operaatio. Tarvittaessa tunnistautumistieto tallennetaan järjestelmään.

Kun Telialta on saatu tarvittavat metatiedot, voidaan aloittaa tunnistautumisprosessi. Kuten mainitsin aiemmin, ensin käydään hakemassa

Telialta tunnistautumisosoite, jonne käyttäjä voidaan ohjata. Tämä tapahtuu siten, että lisätään openid-client JavaScript-kirjasto projektiimme, jotta voidaan luoda asiakasilmentymä (engl. client instanssin) myöntäjällä (engl. issuer) tunnistautumispalveluun. Tämä vaihe on ehdoton tunnistautumisen tapahtumaketjussa. Ulkoistin tämän toiminnallisuuden funktioon, jota voidaan käyttää monesta paikasta, jotta jatkokehitys ja vahvan tunnistautumisen käyttöönotto olisi mahdollisimman yksinkertaista.

6.6 Tunnistautumisosoitteen haku

```
const getAuthenticationUrl = async (
    redirectUrl: string,
    lang: LanguageCodeType
) => {
    const client = await getTeliaIssuerClient(redirectUrl);

    const authUrl = client.authorizationUrl({
        scope: "openid",
        ui_locales: lang,
    });

    return authUrl;
};
```

Kuva 2. Funktio, joka palauttaa tunnistautumisosoitteen kutsujalle.

Tämä funktio saa parametrikseen uudelleenohjausosoitteen sekä kielen. Uudelleenohjausosoite lähetetään käyttöliittymästä, jotta käyttäjä voidaan palauttaa samalle sivulle, mistä tunnistautumisen tapahtumaketju aloitettiin. Käyttäjän kieli tarvitaan käyttöliittymästä, jotta Telian tunnistautumisportaali voidaan asettaa samalle kielelle, joka käyttäjällä on järjestelmässä.

```
const getTeliaIssuerClient = async (redirectUrl: string) => {
    const teliaIssuer = await Issuer.discover(
        `${teliaApiRoot}/.well-known/oauth-authorization-server`
    );

    const clientMetaData = getTeliaClientMetaData(redirectUrl);

    const secret = await getSecret([redacted], true);

    const teliaJWK = pem2jwk(secret);

    const client = new teliaIssuer.Client(clientMetaData, { keys: [teliaJWK] });

    return client;
};
```

Kuva 3. Funktio, jolla luodaan myöntäjä Telian tunnistautumispalvelimelle.

Kun myöntäjää luodaan, tulee ottaa yhteys dokumentissa mainittuun .well-known endpointtiin. Tästä on maininta openid-clientin dokumentaatiossa. Kun myöntäjä on luotu, kutsutaan toista funktiota, joka hakee Telian toimittamat metatiedot. Seuraavassa metatietoesimerkissä ei ole JWK-avainta mukana tietoturvasyistä.

```
const getTeliaClientMetaData = (redirectUrl: string): ClientMetadata => ({
    issuer: teliaApiRoot,
    redirect_uris: [`${getBaseUrl()}/Teliaresponse?redirect_url=${redirectUrl}`],
    response_types: ["code"],
    grant_types: ["authorization_code"],
    client_id: teliaClientId,
    token_endpoint_auth_method: "private_key_jwt",
    token_endpoint_auth_signing_alg: "RS256",
    id_token_encrypted_response_alg: "RSA-OAEP",
    id_token_encrypted_response_enc: "A128CBC-HS256",
    id_token_signed_response_alg: "RS256",
    userinfo_encrypted_response_alg: "RSA-OAEP",
    userinfo_encrypted_response_enc: "A128CBC-HS256",
    userinfo_signed_response_alg: "RS256",
    request_object_encryption_alg: "RSA-OAEP",
    request_object_encryption_enc: "A128GCM",
    request_object_signing_alg: "RS256",
});
```

Kuva 4. Funktio, jolla haetaan Telian toimittamat metatiedot.

Tämän jälkeen haetaan aiemmin luotu yksityinen avain. Järjestelmään otetaan käyttöön pem2jwk JavaScript-kirjasto, jolla käännetään PEM-tiedoston sisältö JWK-avaimeksi. Kun avain on käännetty, tehdään myöntäjällä uusi asiakasilmentymä ja annetaan sille haetut metatiedot sekä yksityinen avain JWK-muodossa. Tämän jälkeen palautetaan asiakasilmentymä funktion kutsujalle.

Kun asiakasilmentymä on saavutettu (Kuva 2), voidaan tunnistautumisosoite hakea kutsumalla asiakasilmentymän authorizationUrl -funktiota, jolle asetetaan laajuudeksi (engl. scope) ”openid”. Muuttujaan ”ui_locales” asetetaan käyttäjän oma kieli. Tämä osoite palautetaan käyttöliittymälle, jossa käyttäjä uudelleenohjataan suorittamaan tunnistautuminen.

6.7 Tunnistautumisen tallennus Kodiaan

Kun käyttäjä on suorittanut tunnistautumisprosessin, hänet palautetaan Kodian sovellukseen. Palautumisen osoite on lähetetty aiemmin Telialle, jotta Telia osaa palauttaa käyttäjän oikeaan osoitteeseen. Kun loin Telian metatiedot tunnistautumisosoitetta varten (Kuva 4), asetin ”redirect_uris” -muuttujaan sen osoitteen lisäksi oman uudelleenohjausosoitteen kyselymerkkijonoparametriksi, jotta käyttäjä voidaan palauttaa takaisin lähtöpisteeseen. Tämä uudelleenohjausosoite tulee käyttöliittymästä silloin, kun tunnistautumisprosessi aloitetaan.

Renderöidään käyttäjälle haluttu komponentti, kun tämä tulee takaisin tunnistautumisportaalista. Käyttäjä palautetaan osoitteeseen, joka loppuu ”/Teliaresponse”. Loin tälle oman reitin (engl. route).

```
<Route path={"/Teliaresponse"}>
  <TeliaRedirectHandler />
</Route>
```

Kuva 5. Reitti, joka renderöi halutun komponentin.

Loin tämän TeliaRedirectHandler -komponentin sen vuoksi, että sillä voidaan uudelleenohjata käyttäjä takaisin lähtöpisteeseen. Tämän lisäksi komponentti noutaa pääsykoodin tunnistuskoodin avulla, joka tulee onnistuneesta tunnistautumisesta. Komponentti tallentaa myös tunnistautumisen järjestelmään, jotta käyttäjä voi suorittaa vahvaa tunnistautumista vaativia operaatioita.

```
const TeliaRedirectHandler: React.FC = () => {
  const navigate = useNavigate();

  const [getAndSaveAccessToken] = useCreateAndSaveAccessTokenMutation();

  const { setError, setSuccess } = useShowNotification();

  useEffect(() => {
    // Get the URL parameters
    const urlParams = new URLSearchParams(window.location.search);

    // Get the redirectUrl and accessToken from the URL parameters
    const redirectUrl = urlParams.get("redirect_url");
    const authCode = urlParams.get("code");

    if (redirectUrl && authCode) {
      // Handle the redirectUrl and accessToken as needed

      // Define an async function to handle the tokens
      const handleTokens = async () => {
        const tokens = await getAndSaveAccessToken({
          variables: { code: authCode, redirectUrl },
        });

        const accessToken = tokens?.data?.getAndSaveAccessToken?.accessToken;
        const refreshToken = tokens?.data?.getAndSaveAccessToken?.refreshToken;

        if (accessToken && refreshToken) {
          persistTokens(accessToken, refreshToken);
          setSuccess(t("authenticationModal.success"));
        } else {
          setError(t("authenticationModal.failed"));
        }

        navigate(redirectUrl);
      };

      // Call the async function
      handleTokens();
    } else if (redirectUrl) {
      navigate(redirectUrl);
      setError(t("authenticationModal.failed"));
    } else {
      setError(t("authenticationModal.failed"));
    }
  }, [navigate, getAndSaveAccessToken, setError, setSuccess]);

  return null;
};
```

Kuva 6. TeliaRedirectHandler -komponentti.

Kuvassa näkyvä pääsykoodi ei ole noudettu Telialta, vaan Kodian sovelluksen koodi, jolla tunnistetaan käyttäjä. Komponentti tallentaa aluksi verkko-osoitteen kyselymerkkijonoparametrit ”redirect_url” ja ”code” talteen. Mikäli niitä ei löydy, tunnistautuminen on epäonnistunut. Tässä tapauksessa, jos aiemmin asetettu

”redirect_url” löytyy, voidaan uudelleenohjata käyttäjä takaisin lähtöpisteeseen. Tällöin käyttäjälle tulee ilmoittaa, että tunnistautuminen on epäonnistunut. Jos ”redirect_url”:ia ei löydy, näytetään pelkästään virheilmoitus, sillä käyttäjää ei voida uudelleenohjata minnekään.

Onnistuneessa tunnistautumisessa molemmat kyselymerkkijonoparametrit löytyvät ja prosessia voidaan jatkaa. Ensiksi kutsutaan taustapalvelussa (engl. backend) olevaa graphql -mutaatiota koukun (engl. hook) avulla. Mutaatiota voidaan käyttää esimerkiksi tiedon lisäämiseen, muokkaamiseen ja poistamiseen (GraphQL 2023).

```
Mutation: {
    [redacted] async (
        _,
        { code, redirectUrl },
        { dataSources: { userAPI } }
    ) => {
        const ssn = await getSSN(code, redirectUrl);

        const tokens = await userAPI.createStrongAuthentication(ssn);

        return tokens;
    },
},
```

Kuva 7. Mutaatio, jolla haetaan tunnistautuneen henkilön henkilötunnus sekä tehdään sille tarvittavat toimet, joita en voi avata enempää. Kutsujalle palautetaan myös uudet koodit, jotka sisältävät tiedon vahvasta tunnistautumisesta.

Siirryttäessä mutaatioon voidaan löytää kaksi eri funktiota. Ensimmäinen funktio hakee tunnistautuneen käyttäjän henkilötunnuksen ja toinen luo käyttäjälle todennuksen vahvasta tunnistautumisesta järjestelmään. Mutaatio lähettää uudet järjestelmän pääsykoodit käyttöliittymään, jotta se sisältää tiedon onnistuneesta vahvasta tunnistautumisesta.

```
const getSSN = async (code: string, redirectUrl: string) => {
    const accessToken = await getAccessToken(code, redirectUrl);

    const client = await getTeliaIssuerClient(redirectUrl);

    const claims = await client.userinfo(accessToken);

    return claims["urn:oid:1.2.246.21"];
};
```

Kuva 8. Funktio, jolla haetaan käyttäjän henkilötunnus.

Funktion tehtävä on hakea henkilön henkilötunnus, ja tämä kyseinen funktio on jaettu kolmeen alifunktioon. Ensiksi haetaan Telialta pääsykoodi. Pääsykoodilla mahdollistetaan henkilötietojen nouto. Sen jälkeen tehdään asiakasilmentymä, joka tehtiin myös aiemmassa vaiheessa (Kuva 3). Viimeiseksi haetaan henkilötiedot. Tämän jälkeen palautetaan henkilötunnus henkilötiedoista.

```
const getAccessToken = async (code: string, redirectUrl: string) => {
    // // Get access token
    const url = `${teliaApiRoot}/oauth2/token`;

    // Load the private key
    const privateKey = await getSecret([redacted] true);

    // Define the sign options
    const signOptions = {
        algorithm: "RS256",
    };

    const payload = {
        iss: teliaClientId,
        sub: teliaClientId,
        aud: teliaApiRoot,
        exp: (new Date().getTime() + 60000) / 1000,      You, 6 days ago • Authenticat
        jti: randomString(16),
    };

    // Sign the JWT
    const token = jwt.sign(payload, privateKey, signOptions);

    try {
        const params = new URLSearchParams({
            grant_type: "authorization_code",
            redirect_uri: `${getBaseUrl()}/Teliaresponse?redirect_url=${redirectUrl}`,
            code: code,
            client_id: teliaClientId,
            client_assertion_type:
                "urn:ietf:params:oauth:client-assertion-type:jwt-bearer",
            client_assertion: token,
        });

        const response = await axios.post(`${url}?${params.toString()}`);

        return response.data.access_token;
    } catch (error) {
        console.error(error);
        // Handle error accordingly
        throw new GraphQLError("Unauthorized, Failed to get access token");
    }
};
```

Kuva 9. Funktio, joka hakee pääsykoodin Telialta.

Ensimmäiseksi haetaan Telian ja Kodian järjestelmän välinen yksityinen avain. Allekirjoitetaan JWT syöttämälläni tiedoilla ja asetetaan voimassaolon ajaksi yksi minuutti. Kun koodi on allekirjoitettu, voidaan tehdä axios-kutsu Teliaan, joka antaa meille tunnistautumiskoodia vastaavan pääsykoodin.

Asiakasilmentymän luonti ilmenee kuvassa 3. Funktion viimeinen tehtävä on suorittaa henkilötietojen nouto. Tämä onnistuu asiakasilmentymän omalla funktiolla ”userinfo”, jolle annetaan haettu pääsykoodi. Funktio ilmentää tiedonvaihtoa Telian ja kutsujan välillä, jossa kutsuja saa pääsykoodin avulla Telialta oikeutetun käyttäjän tiedot. Saaduista tiedoista palautetaan kutsujalle esimerkiksi oikeutetun käyttäjän henkilötunnus.

Tämän jälkeen kutsutaan funktiota, joka luo todennuksen järjestelmään onnistuneesta vahvasta tunnistautumisesta. Tätä toteutusta en voi avata tietoturvasyistä.

Kun mutaatio on suorittanut sille asetetut funktiot, voidaan palata takaisin käyttöliittymän pariin (Kuva 6). Mutaatiosta noudetaan siellä ilmentyneet Kodian pääsykoodit, jotka sisältävät tiedon vahvasta tunnistautumisesta. On tärkeää, että nämä pääsykoodit tulevat käyttöön käyttäjälle. Kutsumme siis ”persistTokens” -funktiota, jonka tehtävä on poistaa edelliset pääsykoodit käyttäjän selaimen istuntokohtaisista tiedoista sekä paikallisvälimuistista ja korvata ne uusilla pääsykoodeilla.

```
export const persistTokens = (accessToken, refreshToken) => {
  // virheiden vuoksi vanha heti pois
  localStorage.removeItem("accessToken");
  sessionStorage.removeItem("accessToken");

  localStorage.removeItem("refreshToken");
  sessionStorage.removeItem("refreshToken");

  if (localStorage.getItem("rememberMe") === "true") {
    localStorage.setItem("accessToken", accessToken);
    localStorage.setItem("refreshToken", refreshToken);
  } else {
    sessionStorage.setItem("accessToken", accessToken);
    sessionStorage.setItem("refreshToken", refreshToken);
  }
};
```

Kuva 10. Funktio, joka poistaa edelliset pääsykoodit ja päivittää ne uusiin, joissa on mukana myös vahvan tunnistautumisen tieto.

Pääsykoodi on koodi, jossa on tietoa kirjautuneesta käyttäjästä ja tämän oikeuksista. ”Refresh token” on sen sijaan koodi, jolla voidaan tarvittaessa hakea uusi pääsykoodi, mikäli edellinen vanhentuu. Pääsykoodeille annetaan aina vanhentumisaika.

On oleellista huomioida, että vahva tunnistautuminen on voimassa vain rajoitetun ajan. Tästä huolehtii funktio, joka tarkistaa pääsykoodin jokaisella toiminnolla, minkä käyttäjä tekee. Jokainen toiminto joka suoritetaan, menee tietyn funktion kautta. Tämä funktio tarkastaa pääsykoodin käyttöoikeuden ja voimassaoloajan.

```
export default async (
    { body: { token } }: { body: { token: string } },
    res: Response
) => {
    try {
        const payload = await verifyToken(token, "accessToken");

        return res.json({
            ...payload,
            isStronglyAuthenticated: await isStronglyAuthenticated(
                payload.strongAuthenticationId
            ),
        });
    } catch (e) {
        if (e instanceof TokenExpiredError) {
            res.status(400);
            return res.json(expiredToken);
        } else if (e instanceof JsonWebTokenError) {
            res.status(400);
            return res.json(invalidToken);
        } else {
            console.error(e);
            res.sendStatus(500);
        }
    }
};
```

Kuva 11. Funktio, joka tarkistaa koodin.

Kuvan 11 funktiossa ensiksi tarkistetaan pääsykoodi, jonka jälkeen palautetaan pääsykoodi sekä tieto siitä, onko vahva tunnistautuminen voimassa. Tämän

tarkastaa funktio, jonka sisältöä en voi tietoturvasyistä näyttää. Tiivistetysti, funktio tarkastaa milloin vahva tunnistautuminen on suoritettu, kuinka pitkään se on voimassa ja onko tunnistautumisen aikaraja tullut vastaan. Mikäli aikarajaa ei ole saavutettu, palautuu ”isStronglyAuthenticated” -kenttään arvo ”true”, muuten ”false”.

Arvoa voidaan hyödyntää koko taustapalvelussa oikeuksien tarkasteluun. Taustapalvelun kontekstissa (engl. context) on aina tiedossa, onko käyttäjällä voimassaoleva vahva tunnistautuminen vai ei. Tämän ansiosta ominaisuuden käyttöönotto eri puolelle sovellusta onnistuu vaivattomasti.

7 Tulokset

Opinnäytetyön tuloksena saavutettiin valmis vahvan tunnistautumisen prosessi Kodian web-sovellukseen. Toteutusvaiheessa kytkettiin vahvan tunnistautumisen vain tilitystoimintoon, mutta se saadaan helposti myös muihin ominaisuuksiin käyttöön sen modulaarisen rakenteen vuoksi. Modulaarisella rakenteella tarkoitan sitä, että jaottelin funktiot omiin kokonaisuuksiin, jotka ovat helposti vaihdettavissa.

Toteutusvaihe eteni todella hyvin, ottaen huomioon, että aihepiiri oli minulle täysin tuntematon. Sain tarvittaessa tukea työkavereiltani, joiden ansiosta sain tämän ominaisuuden toteutettua valmiiksi määräajassa. Ensimmäinen haasteeni oli selkeän lähtöpisteen puuttuminen, joten etenin tutkimalla dokumenttia sekä kokeilemalla dokumentista löytyviä päätepisteitä eri muuttujilla. Toinen haasteeni oli Telian tunnistautumislinkin auki saamisessa. Olin yhteydessä Teliaan, jolta sain tiedon, että yhteys Kodiaan oli katkennut. Tämän jälkeen Telia aukaisi yhteyden uudestaan Kodiaan, jonka jälkeen Telian tunnistautumissivu avautui normaalisti ja pääsin etenemään toteutuksessa.

Kolmas haaste prosessissa ilmeni vahvan tunnistautumisen tallentamisen suunnittelussa. Tätä en voi avata enempää tietoturvasyistä, mutta sain mielestäni aivan pätevän ratkaisun tähän ongelmaan. Viimeinen ongelmakohta oli, kun vahvan tunnistautumisen tieto tuli saada kirjautuneen käyttäjän

pääsykoodiin. Tämä oli lopulta suoraviivainen toteuttaa, sillä sain tukea tässäkin vaiheessa prosessia. En ollut aiemmin tehnyt mitään JWT:illä. Opin niistä todella paljon myös tämän toteutuksen myötä.

Tulokset vastasi mielestäni hyvin odotuksiani, sekä suunnitellut tavoitteet täyttyivät myös. Suuri osa prosessiin kuluneesta ajasta oli Telian kanssa käytyä sähköpostiviestinvaihtoa, jolloin työssä etenemisen tuli odottaa ja tein muita tehtäviä. Sen aikaa täytyi tehdä muita tehtäviä. Olen todella tyytyväinen tuloksen funktionaaliseen jaotteluun sekä vahvan tunnistautumisen tiedon tallentamiseen Kodiaan.

Vahva tunnistautuminen parantaa Kodian sovelluksen tietoturvaa merkittävästi niissä osissa, joihin se kytketään. Nyt esimerkiksi rahojen tilitys, termi joka tarkoittaa rahan lähetystä asunnon omistajalle, on mahdotonta, mikäli henkilö ei ole asiakkaan ennalta määräämä. Myös sovelluksen muissa ominaisuuksissa pyritään vähentämään virheiden tapahtumisen mahdollisuutta, sekä varmistetaan, että oikeutettu käyttäjä suorittaa esimerkiksi vuokrasuhteen irtisanomisen sekä hakemusten lähettämiset.

Toiminto ei vielä ole tuotantokäytössä ja ennen kuin se kytketään päälle, täytyy asiakkaita opettaa käyttämään uutta ominaisuutta. Uskon, että asiakkaat ovat todella tyytyväisiä toteutukseen. Toiminto tuo lisäkuluja Kodian asiakkaalle, mutta varmistaa sovelluksen ominaisuuksien turvallisuuden ja parhaimmillaan suojaa asiakasta tämän varoihin kohdistuvilta tietoturvauhilta.

8 Pohdinta

Vahvan tunnistautumisen käyttöönotto on suuri askel kohti tietoturvallisempaa sovellusta. Tietoturvallinen sovellus luo luottamusta asiakkaan ja palveluita tarjoavan yrityksen välille sekä tarjoaa turvallisuuden tunnetta. Asiakasluottamus on todella tärkeää jokaisessa organisaatiossa.

Vaikka vahva tunnistautuminen on yksi ylimääräinen ”askel” ennen kuin voidaan tehdä operaatio, turvallisuuden vuoksi se on askeleen arvoinen. Vahva

tunnistautuminen suojaa heidän omaisuuttaan, sekä vähentää sovelluksen väärinkäyttöä, varsinkin käyttäjien määrän ollessa suuri. Jokaisella vuokralaisella, joka on merkattu Kodiaan, on myös oikeus päästä katsomaan omia tietojaan vuokraukseen liittyen. Tietoturvasta on tärkeää pitää huolta, sillä kyberhyökkäysten uhka on läsnä sovelluksesta ja organisaatiosta riippumatta.

Kuten kaikkien uusien ominaisuuksien kohdalla, myös vahva tunnistautuminen täytyy kouluttaa vanhoille asiakkaille, sekä tarvittaessa myös käyttöönotoissa uusille asiakkaille. Kouluttaessa voidaan painottaa muun muassa mihin ominaisuuksiin sitä tarvitaan, kuinka kauan tunnistautuminen on voimassa sekä kuinka tunnistaudutaan Kodiassa.

Vaikka vahva tunnistautuminen ei ole ilmainen organisaatiolle, suosittelen sen liittämistä sovellukseen etenkin silloin, kun sovelluksessa käsitellään arkoja tai liiketoiminnalle kriittisiä tietoja. Esimerkiksi Kodian sovelluksessa on henkilötietoja sekä rahaliikennettä, jolloin pieninkin uhka tietojen väärinkäytölle on minimoitava. Vahva tunnistautuminen auttaa tässä merkittävästi.

Tehdessäni opinnäytetyötä opin todella paljon erilaisista salausavaimista ja pääsykoodeista. Opin käsittelemään pääsykoodeja mielestäni hyvin. Opin myös vahvan tunnistautumisen tapahtumaketjun, eli kuinka vahva tunnistautuminen todellisuudessa etenee. Osaisin mielestäni liittää vahvan tunnistautumisen myös toisiin sovelluksiin. Uskoisin, että eri ohjelmointikielille löytyy myös vastaavanlaiset apukirjastot, joilla voidaan tehdä kriittisiä toimintoja. Esimerkiksi PEM-avaimen muunto JWK:ksi tai Issuerin luonti, jolla voimme olla yhteydessä Teliaan.

Lähteet

Authy. 2023. What is 2FA? https://authy.com/what-is-2fa/ 17.10.2023

Bhattacharyya, D., Ranjan, R., Alisherov, F. & Choi, M. 2009. Biometric authentication: A review. International Journal of u-and e-Service, Science and Technology. https://urly.fi/3hxW 17.10.2023

Beyond identity. 2021. The Importance of User Experience in Customer Authentication. https://www.beyondidentity.com/blog/importance-user-experience-customer-authentication 26.9.2023

Blythe, P. T. 2004. Improving public transport ticketing through smart cards. In Proceedings of the institution of civil engineers-municipal engineer. https://urly.fi/3hxX 17.10.2023

Child, H. 2021. Using SMS for Strong Customer Authentication in 2021. Open Banking Excellence. 15.4.2021. Blogi. https://www.openbankingexcellence.org/blog/using-sms-for-strong-customer-authentication-in-2021-2/ 19.9.2023

CLARIN ERIC. 2023. Principles of Data Processing. https://www.clarin.eu/content/principles-data-processing 17.10.2023

Eduskunta. 2023. EU:n yleinen tietosuoja-asetus. https://www.eduskunta.fi/FI/naineduskuntatoimii/kirjasto/aineistot/kotimainen_oikeus/LATI/EUn-tietosuojauudistus/Sivut/EUn-yleinen-tietosuoja-asetus.aspx 17.10.2023

European Data Protection Supervisor. 2023. History of the General Data Protection Regulation. https://edps.europa.eu/data-protection/data-protection/legislation/history-general-data-protection-regulation_en 17.10.2023

FIDO. 2023. What is FIDO? https://fidoalliance.org/what-is-fido/ 17.10.2023

FIDO Patentti. 2019. Methods and systems for providing FIDO authentication services. https://patents.google.com/patent/US10917405B2/en 7.11.2023

Fremery, R. 2021. How to Protect yourself from Social Engineering Attacks. 18.11.2021. Blogi. https://blog.lastpass.com/2021/11/how-to-protect-yourself-from-social-engineering-attacks/ 26.9.2023

GDPR.eu. 2023. What is GDPR? https://gdpr.eu/what-is-gdpr/ 17.10.2023

Givens, B. 2021. What is a USB security key, and how do you use it? https://www.tomsguide.com/news/usb-security-key 3.4.2024

GraphQL.org. 2023. Queries and Mutations. https://graphql.org/learn/queries/#mutations 5.2.2024

Information Commissioner's Office (ICO). 2023. What is personal information? https://ico.org.uk/for-organisations/uk-gdpr-guidance-and-resources/personal-information-what-is-it/what-is-personal-information-a-guide/ 17.10.2023

Kyberturvallisuuskeskus. 2023. https://www.kyberturvallisuuskeskus.fi/fi/toimintamme/saantely-ja-valvonta/sahkoinen-tunnistaminen 17.10.2023

Locke, L., Ziegler, M. (2016). How to build a stronger reputation with trust and resiliency. https://instituteforpr.org/build-stronger-reputation-trust-resiliency/ 17.10.2023

Microsoft. 2024. Mikä on TPM-turvapiiri? https://support.microsoft.com/fi-fi/topic/mik%C3%A4-on-tpm-turvapiiri-705f241d-025d-4470-80c5-4feeb24fa1ee 3.4.2024
Wikipedia. 2024. Near-field communication https://fi.wikipedia.org/wiki/Near-field_communication 3.4.2024
Nextauth. 2023. Mobile app authentication: 6 best practises. https://www.nextauth.com/mobile-app-authentication/ 17.10.2023
Okta. 2023. Strong Authentication: Definition & Security Factors. https://www.okta.com/identity-101/what-is-strong-authentication/ 17.10.2023
Orr, J. 2019. Great, Secure Experiences Come From Anticipating User Authentication Needs. https://www.cshub.com/security-strategy/articles/great-secure-experiences-come-from-anticipating-user-authentication-needs 19.9.2023
Stanislav, M. (2015). Two-factor authentication (Vol. 4). IT Governance Ltd. 17.10.2023
Thales. 2024. What is an eSE? https://www.thalesgroup.com/en/markets/digital-identity-and-security/mobile/secure-elements/embedded-secure-element 3.4.2024
Tietosuojavaltuutetun toimisto. 2023. Henkilötietojen käsittelijät. https://tietosuoja.fi/henkilotietojen-kasittelijat 17.10.2023
Tietosuojavaltuutetun toimisto. 2023. Henkilötietojen käsittelyn roolit ja vastuut. https://tietosuoja.fi/henkilotietojen-kasittelyn-roolit-ja-vastuut 17.10.2023
Tietosuojavaltuutetun toimisto. 2023. Rekisteröidyn oikeudet eri tilanteissa. https://tietosuoja.fi/rekisteroidyn-oikeudet-eri-tilanteissa 17.10.2023
Tucakov, D. 2023. Biometrics vs Passwords: Which is safer? https://phoenixnap.com/blog/biometrics-vs-passwords 26.9.2023
Virgillito, D. 2019. What are one-time passwords and their pros and cons? https://resources.infosecinstitute.com/topics/security-awareness/one-time-passwords-pros-and-cons/ 26.9.2023
WellDev. 2022. https://www.linkedin.com/pulse/why-two-factor-authentication-necessary-businesses-welldevintl/ 17.10.2023
ZippyOPS. 2023. Top Authentication Trends to Watch Out for in 2023. https://www.zippyops.com/top-authentication-trends-to-watch-out-for-in-2023 26.9.2023