TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Roman Malõšev 206663IADB

Microsoft Office'i pistikprogramm digiallkirjastamiseks

Bakalaureusetöö

Juhendaja: Priit Raspel
MSc

Tallinn 2023

Autorideklaratsioon

Kinnitan, et olen koostanud antud lõputöö iseseisvalt ning seda ei ole kellegi teise poolt varem kaitsmisele esitatud. Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on töös viidatud.

Autor: Roman Malõšev

01.01.2023

Annotatsioon

Antud bakalaureusetöö käsitletavaks probleemiks on digiallkirjastamise, kui protsessi, keerulisus. Probleemi valik tuleneb töö autori praktikakohast, kus tööprotsess on tihedalt seotud dokumentidega ning nende digiallkirjastamisega.

Antud bakalaureusetöö eesmärgiks on muuta digiallkirjastamise protsessi tõhusamaks ning seeläbi vähendada digiallkirjastamisele kuluvat aega. Eesmärgi saavutamiseks luuakse töö raames pistikprogramm Microsoft Office keskkonnale, mille kaudu saab digiallkirjastada faile keskkonda muutmata. Valmis rakendus toetab selliseid Microsoft Office rakendusi nagu: Word, Excel, PowerPoint ja Outlook. Toetatud digiallkirjastamise meetoditeks on Smart-ID, Mobiil-ID ja ID-kaart.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 48 leheküljel, 6 peatükki, 17 joonist, 0 tabelit.

Abstract

Microsoft Office Plugin for Digital Signing

The problem addressed in this bachelor thesis is complexity of digital signing as a process. Problem originates from author's work place, where work process is closely related to documents and their digital signing.

The aim of this bachelor thesis is to make digital signing process more efficient and reduce time needed to give digital signature. Whole bachelor thesis is divided into four stages. First stage is information gathering, that is needed for problem solving. Second stage is analysis, where author consideres different solutions for placed problem. Also, analysis part contains justification for chosen technologies as well as thesis solution comparison with other similar solution for digital signing in Microsoft Office environment. When conduction analysis, author considers company requirements as well as requirements specified in task setup part of this thesis. Third is plan stage, where detailed plan is made, that describes working software functionality as well as database, server and plugin architecture. Last thesis stage is software creation stage. During last stage working software is created based on all the previous stages.

Software that is built as part of this bachelor thesis is plugin for Microsoft Office environment, through which files can be digitally signed without changing environment. Plugin supports such Microsoft Office environments as Word, Excel, PowerPoint and Outlook. In addition, plugin can operate in web environment as well as in standalone applications. In Outlook files, that can be signed are files attached to letters. Plugin supports one or many files signing at once. In Word, Excel and PowerPoint signing file is file that is opened in application. There are three possible digital signature options implemented as signing methods, which are Smart-ID, Mobile-ID and ID-card.

The thesis is in Estonian and contains 48 pages of text, 6 chapters, 17 figures, 0 tables.

Lühendite ja mõistete sõnastik

AAD	Azure Active Directory
AdES	Advanced electronic signature
API	Application programming interface
AWS	Amazon Web Services
AWS S3	Amazon Web Services Simple Storage Service
Azure	Microsoft Azure
DAO	Data Access Object
eIDAS	Electronic IDentification, Authentication and trust Services
Office	Microsoft Office
PDF	Portable Document Format
QES	Qualified electronic signature
SK	SK ID Solutions
SSO	Single sign-on

Sisukord

Jooniste loetelu

1 Sissejuhatus

Digiallkirjastamine on mugav allkirjastamise meetod, mille kasutamisel saab allkirjastada ja hoiustada faile digitaalselt. Teatud taseme digiallkirjad, ei näita ainult digiallkirja andja nõustumist faili sisuga, vaid tõendavad ka faili terviklust. Vaatamata tehnoloogia eelistustele, ei ole digiallkirjastamise protsess optimiseeritud ning sagedasti peab digiallkirja andmiseks kasutama eraldi rakendust. Digiallkirjastamise optimeerimine on antud bakalaureusetöö eesmärk.

Antud bakalaureusetöö eeldatavaks tulemuseks on töötav rakendus, mis funktsioneerib Microsoft Office pisktikprogrammina ning võimaldab digiallkirjastada faile keskkonda muutmata neljas erinevas Microsoft Office rakenduses. Rakendus toetaks kolme digiallkirjastamise meetodit, milleks on Smart-ID, Mobiil-ID ja ID-kaart. Töö lähtetingumusteks on selgelt formuleeritud lahendatav probleem ning eeldatava rakenduse prototüüp, mis tõestab rakenduse loomise võimalust.

Antud töö on jagatud neljaks osaks, kus iga töö osa vastab ühele konkreetsele töö kirjutamise etapile. Töö esimene osa vastab informatsiooni leidmisele ning selle raames kogutakse informatsiooni, mida hakatakse töö kirjutamiseks kasutama. Teine töö osa vastab analüüsi etapile, mille raames analüüsitakse leitud informatsiooni ning valitakse tehnoloogiaid, mida hakatakse rakenduse loomiseks kasutama. Lisaks, analüüsi osas, formuleeritakse nõudeid, millele peab valmiv rakendus vastama. Kolmanda töö osa vastab kavandi koostamise etapile, mille raames koostatakse detailne plaan ning mille järgi luuakse rakendus. Peale plaani koostamise, formuleeritakse antud etapis eeldatava rakenduse funktsionaalsust. Neljas ehk viimane töö osa vastab programmi loomise etapile. Selle raames on kirjeldatud rakendust koos autorile ettenähtamatute piirangutega, mis ilmnesid rakenduse loomisel. Lisaks, on selle etapi raames kirjeldatud rakenduse funktsionaalsus ning millisel määral, see erineb eeldatavast funktsionaalsusest.

2 Ülesande püstitus

Antud töö osa annab ülevaate töö raames valitud probleemist, püstitatud eesmärkidest, probleemi valimise põhjusest, töö piirangutest, bakalaureuse töö etapidest ja töö eeldatavast tulemusest.

2.1 Probleem

Digiallkirja abil saab lisada informatsiooni andmetele, mis on teatud taseme digiallkirjade puhul, samaväärne käega antud allkirjale. Digiallkirja kasutamine on mugavaim viis dokumentide allkirjastamiseks, kuna dokumente ei ole vaja eelnevalt printida. Digiallkirja abil saab neid luua, allkirjastada ja säilitada digitaalselt. Omakorda on funktsionaalsuse kasutamine toonud esile ka mõned probleemid. Antud töös kajastatud probleemiks on digiallkirjastamise, kui protsessi, keerulisus. Failide töötlemiseks on loodud üks kindel programm, kuid failide digiallkirjastamiseks kasutatakse teist programmi. Seetõttu on vajalik ühe faili allkirjastamiseks kahe programmi tööd, mis teebki protsessi aeganõudvaks.

2.2 Eesmärk

Käesoleva bakalaureusetöö eesmärgid on:

- luua pistikprogramm, mis võimaldaks kasutajal faili töödelda ja digiallkirjastada kasutades vaid ühte keskkonda,
- optimeerida digiallkirja andmise protsessi ja seega säästa digiallkirjastamiseks vajalikku aega,
- luua selline lahendus, mida oleks lihtne ja mugav kasutada.

2.3 Probleemi valimise põhjendus

Üheks probleemi valimise põhjuseks on töö autori töökohast tingitud tööprotsess, mis on tihedalt seotud dokumentidega ning nende digiallkirjastamisega. Digiallkirja andmise, kui protsessi, optimeerimine oleks tööandjale kasulik lahendus ning aitaks tõsta märgatavalt kolleegide töötamise efektiivsust. Efektiivsuse tõus kujuneks eelkõige antud töö raames valminud pistikprogrammi abil, mis võimaldaks failide töötlemiseks ja nende allkirjastamiseks kasutada vaid üht keskkonda. On ka teised põhjused. Näiteks, vaatamata sellele, et digiallkirjastamine on Eestis laialt kasutatud, ei ole digiallkirjastamiseks loodud palju lahendusi.

2.4 Piirangud

Töö piiranguks on see, et nii digiallkirjastamine kui ka failiga töötamine peab olema realiseeritud ühe programmi raames. Tulenevalt sellest, et uue keskkonna loomine faili töötlemiseks ei ole mõistlik ning sellest, et Office rakendused toetavad pistikprogrammide loomist, luuakse antud töö raames Office rakendustele pistikprogrammi. [1] Office keskkonda kuuluvad kõige levinumad failidega töötamiseks olevad rakendused nagu Word, Excel ja PowerPoint, mistõttu on valitud Office keskkonna laiendamine. Ajalise piirangu tõttu ei ole töö autor võimeline looma pistikprogramme kahele erinevale keskkonnale. Lisaks, vaadeldakse antud töö raames vaid kahte Office platvormi, milleks on veebipõhine ja eraldiseisev rakendus.

2.5 Etapid

Töö kirjutamine on jaotatud etappideks, kus igal etapil on omad lähteandmed, eesmärgid kui ka tulemused, mida etapi lõpus saavutatakse. Esimese etapi raames leitakse informatsioon erinevatest allikatest, mis on vajalik töö probleemi lahenduseks. Eesmärgiks on koguda informatsioon, mida hiljem analüüsitakse ja kasutatakse. Teiseks etapiks on analüüs, kus analüüsitakse informatsiooni kogumise etapis saadud andmeid. Vaadeldakse erinevaid võimalike viise ülesande lahendamiseks ning kogutud informatsiooni põhjal valitakse tehnoloogiaid, mida kasutatakse püstitatud probleemi lahendamiseks. Selle etapi raames kirjeldatakse nõudeid, millele valmis lahendus peab vastama. Teise etapi tulemuseks on nii põhjendatud loend funktsionaalsusest, mida hakatakse kasutama töö tarkvara kirjutamiseks, kui ka loend nõudeid, millele valmis

lahendus peab vastama. Teisele etapile järgneb plaani koostamine, mille põhjal kirjutatakse tarkvara programm. Etapp kasutab eelmises etapis valitud tehnoloogiaid. Vaadeldakse andmevoogu, andmebaasi arhitektuuri ning kasutajaliidese ja serveri arhitektuuri. Tulemuseks on detailne plaan, mille järgi saab tarkvara programmi koostada. Etapile järgneb programmi kirjutamise etapp, mille käigus koostatakse programm toetudes eelnevalt koostatud programmi kavandile. Realisatsioon, on viimane etapp ning selle tulemusena valmib lahendus töös käsitletavale probleemile.

2.6 Eeldatav tulemus

Töö eeldatavaks tulemuseks on töötav tarkvara programm, mis võimaldaks digitaalselt allkirjastada dokumente. Programm töötaks Microsoft 365 Office pistikprogrammina ja selle abil saaks allkirjastada dokumente järgmistes keskkondades:

- Outlook (nii veebipõhine kui ka eraldiseisev rakendus),
- Word (nii veebipõhine kui ka eraldiseisev rakendus),
- Excel (nii veebipõhine kui ka eraldiseisev rakendus),
- PowerPoint (nii veebipõhine kui ka eraldiseisev rakendus).

Allkirjastamise viisideks oleksid implementeeritud kolm võimalikku digiallkirjastamise võimalust, milleks on Smart-ID, Mobiil-ID ja ID-kaart.

3 Kirjanduse ülevaade

Antud töö osas leitakse erinevaid allikaid, mis on vajalikud probleemi lahendamiseks. Allikatest leitakse vajalikku informatsiooni, mida hakatakse töö probleemi lahenduseks kasutama.

3.1 Digiallkirjastamine

Digiallkiri oma olemuselt on andmed, mida säilitatakse elektroonilisel kujul ning, mis on seotud või loogiliselt ühendatud teiste elektrooniliste andmetega. Antud andmeid kasutatakse allkirja andmiseks. Digiallkiri on võrdväärne käega antud allkirjale ning näitab, et allkirja andja nõustub dokumendi sisuga või on tunnistaja. [2]

3.1.1 Digiallkirjastamise tasemed

Vastavalt eIDAS standardile omab digiallkiri kolm taset. Digiallkirjastamise tasemed on järjestatud selliselt, et iga järgnev digiallkirja tase omab lisaks uutele omadustele ka eelnevate tasemete omadusi. Digiallkirjastamise tasemeteks on järgnevad elektroonsed allkirjad: lihtsad, täiustatud ja kvalifitseeritud. [3]

3.1.1.1 Lihtsad elektroonilised allkirjad

Lihtsaid elektroonseid allkirju iseloomustavad digiallkirjad, mille kaudu ei saa veenduda failide tervikluses ning ei ole võimalik määrata allkirja andjat. Antud liik sisaldab elektroonilisi andmeid, mis on seotud või loogiliselt ühendatud teiste elektrooniliste andmetega. Selle allkirja taseme puhul ei kasutata krüptograafilisi võtteid. Lihtsate elektrooniliste allkirjade üks levinum näide on e-kirja lõpetuseks kirjutatav saatja nimi, mis ei kinnita, et just antud isik on saatnud kirja. [3]

3.1.1.2 Täiustatud elektroonilised allkirjad (AdES)

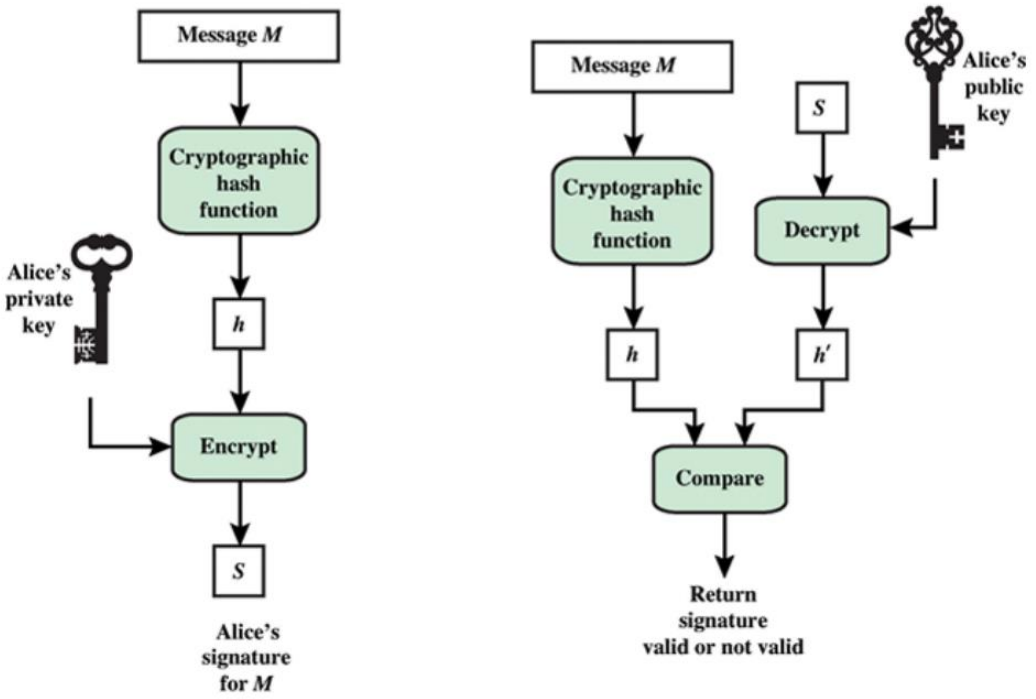
Täiustatud allkirja andmisel kasutatakse krüptograafilisi võtteid, et tuvastada allkirja andjat. Antud allkirjastamise viis aitab tuvastada lisaks allkirjastatud andmete terviklust ehk juhul, kui pärast allkirjastamist on andmed muudetud, on võimalik see tuvastada. Antud allkirjastamise viisi andmiseks kasutatakse asümmeetrilisi krüptoalgoritme, avaliku võtme infrastruktuure ja sertifikaate. [3]

3.1.1.3 Kvalifitseeritud elektroonilised allkirjad (QES)

Kvalifitseeritud elektrooniline allkiri on kõige turvalisem elektroonse allkirjastamise tase. Antud allkirja tase omab analoogseid karakteristikuid täiustatud elektroonsete allkirjadega, kuid lisaks on antud tase moodustatud kasutades kvalifitseeritud allkirja loomise seadet. Lisaks, kvalifitseeritud allkirjad peavad baseeruma kvalifitseeritud sertifikaatidel. Antud tase on ainus tase, mis on samaväärne käega antud allkirjale. [3]

3.1.2 Digiallkirjastamise tööpõhimõte

Digiallkirjastamise protsess toetub nii räsifunktsioonidele kui ka assümeetrilistele krüptoalgoritmidele selleks, et saavutada autentimist, terviklust ning kordumatust. Selleks, et koostada digiallkiri on vaja koguda andmeid, mida hakatakse digiallkirjastama ja teha nendest räsi, kasutades selleks räsifunktsiooni. Digiallkirjadel, mis baseeruvad assümeetrilisel krüptoalgoritmidel, nagu RSA, on räsifunktsioonid vajalikud turvalisuse loomiseks. Selle funktsioonita oleks digiallkirjastamine protsess ebaturvaline selliste küberrünnakute vastu, nagu teadaoleva sõnumi rünne (Ing. known message attack) ja valitud sõnumi rünne (Ing. chosen message attack). Lisaks kasutatakse räsifunktsioone protsessi efektiivsuseks, kuna väikese andmemahu krüpteerimine kestab ajaliselt vähem. Räsi tegemisele järgneb selle krüpteerimine ning, selleks kasutatakse digiallkirja andja privaatvõtit. Krüpteerimise tulemusena valmib digiallkiri ehk teisisõnu krüpteeritud räsi, mis on arvutatud digiallkirjastatavatest andmetest. Digiallkirja verifitseerimiseks on vajalikud digiallkirjastatavad andmed, digiallkirja ja avaliku võtit. Kasutades sama räsifunktsiooni, mis oli kasutatud digiallkirja andmisel, on vaja arvutada räsi digiallkirjastatud andmetest. Sellele järgneb saadud tulemuse võrdlemine dešifreeritud digiallkirjaga. Digiallkirja dešifreerimiseks omakorda kasutakse avaliku võtit. Juhul, kui saadud räsi ja dešifreeritud digiallkiri on võrdsed, saab olla kindel, et digiallkiri on loodud kasutades avaliku võtmega seotud privaatvõtit ning digiallkirjastatud andmed ei ole muudetud. [4] Digiallkirja andmise ning selle verifitseerimise protsess on välja toodud joonisel 1.



Joonis 1. Digiallkirja loomise ja selle verifitseerimise protsess [5]

Kasutades eelnevalt kirjeldatud printsiipi ei ole võimalik autentida digiallkirja andjat. Digiallkirja andjate autentimise probleemi lahendab usaldusväärne kolmas osapool, mis autentib digiallkirja andjat ning väljastab sertifikaate, mis sisaldavad isikuandmeid koos isiku avaliku võtmega. [6]

Sertifikaat on digiallkirjastatud andmed, mis on allkirjastatud usaldusväärse kolmanda osapoolega. Sertifikaadi verifitseerimine on võimalik tänu sertifikaadi teenuse avaliku võtmele, mis on laialt levinud ning, mis eelinstallitatakse seadmetesse. Sertifikaadiga digiallkirjastamisel ei saadeta digiallkirjastud failidega digiallkirja andja avaliku võtit, vaid saadetakse sertifikaati, mis seda sisaldab. Lisaks sisaldab sertifikaat ka digiallkirja andja isikuandmeid selleks, et oleks võimalik digiallkirja andjat tuvastada. [6]

Digiallkirjastamise protsessi turvalisus eeldab, et digiallkirastamiseks kasutatav privaatvõti on digiallkirjastaja ainuvalduses, kuid ei saa välistada juhtumeid, kui see satub kolmandale osapoolele näiteks varastamise tõttu. Kolmanda osapoole poolt ebaõiglast digiallkirja andmise vältimiseks on vajalik sertifikaatide annulleerimine, selleks et digiallkiri, mis ei ole antud privaatvõtme omaniku poolt, ei oleks verifitseeritav ehk teisisõnu oleks kehtetu. Selliste situatsioonide vältimiseks säilitatakse list kõikidest tühistatud sertifikaatidest. Omakorda esineb probleem, kus sertifikaadi annulleerimise järgselt annulleeritakse ka kõik allkirjad, mis olid eelnevalt sertifikaadi kasutamisel

antud. Probleemi lahendamiseks on vajalik teada aeg, millal digiallkirjastamine toimus ja annulleerida vaid neid digiallkirju, mis sooritati pärast sertifikaadi annulleerimist. Digiallkirjastamise aja määramiseks kasutatakse ajatempli teenust. Digiallkirja andja teeb päringu digiallkirja andmise järgselt ajatempli teenusesse, kus väljastatakse ajatempel, mis on üks osa digiallkirjast. Ajatempel näitab, millisel konkreetsel ajahetkel eksisteeris digiallkiri, mis oli saadetud ajatempli teenusesse. [6]

3.1.3 Asümmeetrilised krüptoalgoritmid

Asümmeetrilise krüptoalgoritmi puhul kasutatakse avaliku - ja privaatvõtit. Võtmed on omavahel seotud ning andmed, mis on krüpteeritud kasutades ühte võtit, saavad olla dešifreeritud ainult kasutades teist võtit. Sõltumata, sellest, et võtmepaar on omavahel seotud, ei ole võimalik praktikas ühe võtme olemasolul leida teist võtit. Assümeetriliste krüptoalgoritmide puuduseks on madalam kiirus võrreldes teiste krüptoalgoritmidega. [4]

3.1.4 Räsifunktsioonid

Räsifunktsioonid on krüptograafiaga seotud algoritmid, mis võtavad omavoliliselt pikkuse sisendi ning moodustavad sellest fikseeritud pikkusega bitijada ehk räsi. Räsi tegemine on ühesuunaline protsess, mille puhul ei ole võimalik arvutatud räsist tagasi saada sisendi. Räsi funktsioonid teevad pikemast sisendist väiksema pikkusega räsi, mis tekitab olukorra, kus mitmed sisendid vastavad ühele ja samale räsile. Teisisõnu toimub räside kokkupõrge, kus kaks sisendit annavad samasuguse väljundi. Kuigi räside kokkupõrked eksisteerivad, on praktikas raske leida teist sisendit, mis annaks analoogse räsi. [4]

3.1.5 Smart-ID digiallkirjastamine

Tehnoloogia, mis võimaldab digiallkirjastamiseks ja autentimiseks kasutada nutitelefoni. Smart-ID puhul privaatvõti on jagatud kaheks osaks ning need on salvestatud erinevates kohtades. Üks osa privaatvõtmest lokaliseerub digiallkirja andja ehk privaatvõtme omaniku seadme rakenduses ja teine privaatvõtme osa on Smart-ID teenuse pakkuja serveris. Digiallkirjastamisel privaatvõti ei ühendata vaid räsi krüpteeritakse eraldi privaatvõtme osadega. Seejärel krüpteeritud osad liidetakse kasutades lävi krüptosüsteemi (Ing. Threshold cryptosystem), moodustades sellega digiallkirja. [7]

3.1.6 Mobiil-ID digiallkirjastamine

Mobiil-ID võimaldab teostada nii autentimist, kui digiallkirjastamist. Tehnoloogia põhimõte seisneb selles, et privaatvõtit hoitakse telefoni SIM-kaardi peal. Digiallkirjastamisel allkirjastavatest andmetest tehakse räsi ning saadetakse see digiallkirja andja telefonile. Seejärel, kui digiallkirja andja sisestab õige PIN-koodi, räsi krüpteeritakse. Protsessi tulemusena valmiv digiallkiri saadetakse tagasi. [8]

3.1.7 ID-kaardiga digiallkirjastamine

ID-kaardi kiibil on salvestatud nii sertifikaat, kui privaatvõti. Kiibist saab sertifikaati vabalt lugeda, kuid privaatvõtme lugemine kiibist on võimatu. Selle kasutamine on võimalik ainuüksi PIN-koodide kasutamisel ning vale PIN-koodi kolmekordne sisestus blokeerib ID-kaardi. Tulenevalt sellest, et privaatvõti on kiibil ning seda ei ole võimalik kasutusse saada, on allkirjastatavate andmete räsi krüpteerimine ehk digiallkirjastamine ainuvõimalik lokaalselt allkirja andja seadmes. [9]

Kasutades ID-kaarti digiallkirja andmiseks on laialt kasutatud kaks viisi. Esimeseks digiallkirja andmise võimaluseks on kasutada selleks spetsiaalselt tarkvara programmi ja faile, mis asuvad tarkvara programmiga ühes seadmes. Teiseks aidigallkirja andmise võimaluseks on kasutada brauseri laiendit koos Javascript´iga. [10]

3.1.8 Digiallkirjastamise teenuse pakkujad

3.1.8.1 Dokobit

Digiallkirjastamise teenus, mis võimaldab allkirjastada erinevaid dokumentide formaate kasutades selleks spetsiaalselt valmistatud keskkonda. Lisaks dokumentide allkirjastamisele pakub antud teenus failide säilitamist, e-toetust ning digiallkirjade valideerimist.[11]

Digiallkirjastamise alguses on eelnevalt vaja üles laadida faile dokobit serverisse. Seejärel peab rakendus avama dokobit keskkonna, kas ümbersuunamise teel või olemasoleva programmi osana, kus hakkab toimuma digiallkirjastamine. Digiallkirjastamise järgselt saab programm alla laadida digiallkirjastatud dokumente. [12]

Dokobit on tasuline teenus, mis võimaldab kasutajal tasuta allkirjastada viis dokumenti kuus. Kuulise limiidi tõstmiseks on vaja osta pakett, kus iga digiallkirjastamine maksab kas 0.4€ või 0.3€ sõltuvalt valitud paketist. [13]

3.1.8.2 SK ID Solutions

SK ID Solutions (edaspidi SK) on teenuse pakkuja, mille abil saab digiallkirjastada faile kasutades Smart-ID ja Mobiil-ID. Lisaks, pakub antud teenus ka ajatempliteenust. [14]

Smart-ID ja Mobiil-ID puhul võimaldab antud teenus omandada allkirja andja sertifikaati ja krüpteerida failide räsi, mida teenusele on vaja eelnevalt saata. Teenus ise ei tegele räsi arvutusega ja ei toeta kinnituskoodi arvutamist. Kinnituskood on neljajärguline number, mis saadetakse kasutajale Smart-ID ja Mobiil-ID digiallkirjastamisel nutitelefonile. Teenuse poolt toetatud räsifunktsioonideks, on SHA-256, SHA-384 ja SHA-512. [15, 16]

Ühe digiallkirja andmise hind Mobiil-ID´ga sõltub valitud paketist ja varieerub alates 0.012€ kuni 0.1€.[17] Sarnaselt Mobiil-ID´le, hinna moodustamine sõltub Smart-ID´s valitud paketist ja varieerub alates 0.0088€ kuni 0.1€.[18] Teenusega pakutav ajatempli teenuse hind samuti sõltub valitud paketist ning varieerub alates 0.006€ kuni 0.036€. [19]

3.1.9 Digiallkirjastamise teek

Digiallkirja implementeerimiseks on loodud kaks teeki, millest üks on loodud C++ (libdigidocpp) jaoks ja teine Java (digidoc4j) jaoks. Nii C++ kui ka Java teegid toetavad BDOC-TS ehk .asice, BDOC-TM ehk .bdoc ja DDOC ehk .ddoc formaate.[20] C++ teek on lisaks ka multiplatvormne ja toetab nii Java’t kui ka C#.[21]

Libdigidocpp ja digidic4j teekide funktsionaalsus on sarnane ning need mõlemad aitavad digiallkirjastamist implementeerides järgmiste operatsioonidega:

- digiallkirja konteineri loomine,
- failide konteineri lisamine/kustutamine,
- digiallkirja konteineri lisamine/kustutamine,

- ajatempli saamine ja selle konteineri lisamine,
- digiallkirja andja lisa informatsiooni lisamine digiallkirja juurde,
- digiallkirjade valideerimine,
- failide ja digiallkirjade vaatamine. [22, 23]

3.2 Serveri arhitektuur

Arhitektuuri kasutamine rakenduse loomisel aitab eelkõige luua rakendust kasutades selleks tõestatud võtteid. Arhitektuuri mudeli kasutamine muudab rakenduse arendamist lihtsamaks, kuna tihti paneb mudel piiranguid disainile ning määrab, kuidas kogu rakendus peab töötama. Igal arhitektuuri mudelil on omad eelised ja puudused, millest tulenevad ka lõpliku rakenduse tugevad ja nõrgad küljed. Seetõttu, on arhitektuuri valimine oluline etapp iga rakenduse loomisel, kuna valitud arhitektuur mõjutab valmis rakenduse omadusi. [24]

3.2.1 Monoliitne arhitektuur

See on tarkvara arhitektuur, mille puhul kõik tarkvara komponendid on seotud üksteisega ja on ülesehitatud koostöötamiseks. Komponendid, nagu äri loogika, autentimine, logimine ja andmebaasiga suhtlemine, ei ole eraldatud monoliitses arhitektuuris ning selle tulemusena omab rakendus tihedalt seotud koodi. See arhitektuur sobib väikeste rakenduste moodustamiseks ning selle eelisteks on:

- parem jõudlus. Kogu kood paikneb ühes kohas, mis teeb andmevahetuse koodi erinevate moodulite vahel väga kiireks;
- kogu rakendust on lihtne avaldada. Rakendus kujutab tervikut, mis lihtsustab avaldamise protsessi;
- lihtne testida ja siluda. Kogu rakenduse loogika on ühes kohas, mis vähendab välis komponentide arvu, mida tuleb testimisel ja silumisel arvestada;
- lihtne skaleeruvus. Selleks, et tõsta kogu rakenduse jõudlust on vajalik käivitada veel ühe rakenduse eksemplari. [24]

Selles arhitektuuris esinevad ka mitmed puudused. Näiteks tulenevalt positiivsest aspektist, nagu lihtne skaleeruvus, tuleneb lisaks see, et skaleerida saab vaid kogu rakedust, mitte konkreetset funktsionaalsust. Puuduste hulka kuuluvad ka järgnevad puudused:

- muudatuste tegemise raskus. Tihedalt seotud koodi ühe komponendi muutused võivad tihti põhjustada muutusi ka teistes rakenduse osades;
- avaldamise keerukus ja ressursi nõudlus. Rakenduse avaldamisel tuleb avaldada kogu rakendust, mis võtab rohkem ressursse, sh ka ajalist ressurssi;
- aeglane arenduse tarkvara töö. Rakenduse koodiga töötamisel kasutatakse spetsiaalset tarkvara, mille jõudlus võib langeda suure projekti mahu tõttu;
- kohanemise probleem. Suure projektiga mitte seotud arendajatel võivad tekkida probleemid projektiga kohanemisel;
- ühe tehnoloogia probleem. Tihedalt seotud kood raskendab erinevate tehnoloogiate kasutamist, mille tagajärjeks on monoliitse arhitektuuriga rakendused, kirjutatud kasutades kindlat programmeerimise keelt. [24]

3.2.2 Mikroteenuste arhitektuur

See on arhitektuurne lahendus, mille korral on kogu rakendus jagatud blokkideks ehk mikroteenusteks. Arhitektuuri mõte on jagada suurt rakendust mikroteenusteks, kus igal mikroteenusel on oma konkreetne ülesanne. Mikroteenused suhtlevad omavahel kasutades selleks hästi määratletud liidest (Ing. Well-defined interface). Arhitektuur sobib eelkõige suurtele rakendustele, kuid väikeste rekenduste puhul on see liiga keerulise seadistusega. Arhitektuurse lahenduse puhul saab kasutada erinevaid tehnoloogiaid eri mikroteenuste jaoks. Mikroteenused ei ole tihedalt seotud, mistõttu ei põhjusta ühes mikroteenuses tehtud muudatused, teiste mikroteenuste muutumist, momendini, kuni esialgse mikroteenuse liides ei muutu. Arhitektuuri kõige suurimaks eeliseks on rakenduse tükeldamine, mis vähendab nii programmeerimise, kui ka rakenduse, keerukust. Ühe mikroteenuse muutmise puhul, ei ole vaja avaldada kogu rakendust, vaid saab avaldada ainult mikroteenust, mis muutus, ning seetõttu säästa ressurse. Rakenduse skaleerimise vajadusel, on võimalik skaleerida mikroteenuste ehk funktsionaalsuste kaupa. Näiteks, kui autentimise mikroteenuse koormus pole suuremahuline, ei ole vajalik

teenust skaleerida, vaid saab käivitada kaks või rohkem eksemplari sellest teenusest, mis on koormatud. Lisaeeliseks on ka see, et ühe mikroteenuse sisene viga ei mõjuta otseselt teisi mikroteenusi ja rakenduse töö saab toimuda ka edasi. Teisest küljest omab antud arhitektuur ka puudusi. Üheks puuduseks on silumise raskus, mis on tihti seotud sellega, et vea parandamiseks on vaja jälgida selle esinemist erinevate mikroteenuste vahel. Teiseks puuduseks on rakenduse mikroteenusteks jagamise raskus, ning see, et antud protsess nõuab oskust. Juhul kui mikroteenusi on palju on nende haldamine raskendatud, kui aga mikroteenuseid on vähe, tekib ühel mikroteenusel liiga suur funktsionaalsus. Võib juhtuda ka olukordi, kui mikroteenused on omavahel nii tugevalt seotud, et rakenduse arhitektuur meenutab varjatud monoliidi arhitektuuri. [24]

3.2.3 Serverivaba arhitektuur

Serverivaba arhitektuur on näide tarkvara arhitektuurist, mille puhul rakendus ei tööta pidevalt, vaid käivitatakse see enne töö tegemist ning peatatakse kohe töö lõppedes. Antud arhitektuuri suurim eelis seisneb selles, et arhitektuuri kasutamisel ei ole vajalik serveri seadistamine ja haldamine programmeerija poolt ning saab kontsentreeruda rakenduse koodi kirjutamisel. Serveri seadistamisega, ressursside haldamisega kui ka serveri hooldamisega tegeleb arhitektuuri teenuse pakkuja. Rakendused, mis kasutavad serverivaba arhitektuuri, kasutavad võimalikult vähe arvutusvõimsust ja seega saavad adapteeruda erinevatele koormustele. Serverivaba arhitektuuri eelisteks on:

- kulude kokkuhoid. Maksta tuleb ainult nende ressursside eest, mida rakendus on kasutanud;
- skaleeruvus ja paindlikkus. Serverivaba arhitektuuri teenuse pakkuja ise skaleerib vajadusel rakendust ning selle tulemusena ei teki olukordi, kus rakendusele ei piisa jõudlust või rakendus raiskab ressursse, mida see ei kasuta. [24]

Arhitektuuri puuduste hulka kuuluvad:

- ressursside jagamine teiste rakendustega. Jagatud ressursid seovad rakendusi, mis kasutavad samu ressursse, mistõttu ühe rakenduse töö võib mõjutada teise rakenduse tööd. Näiteks, ühe rakenduse suur ressursside kasutamine võib tekitada olukorda, kus teistele rakendustele ei piisa ressursse normaalseks funktsioneerimiseks. Lisaks, on ühisressursside kasutamisega seotud ka

turvariskid, mis lähtuvad sellest, et ühe rakenduse andmed võivad olla nähtavad teistele rakendustele;

- rakenduse tugev seos serverivaba arhitektuuri teenuse pakkujaga. Erinevad teenuse pakkujad pakuvad erinevaid võimalusi serverivaba rakenduse loomiseks ja olemasoleva rakenduse kohandamine teiseks teenuse pakkujaks võib olla keeruline. [24]

3.3 Office pistikprogramm

Office võimaldab laiendada enda funktsionaalsust kasutades pistikprogramme. Pistikprogrammide kirjutamiseks peab kasutama veebitehnoloogiaid nagu HTML, CSS ja Javascript ning laiendamise toetavateks Office rakendusteks on Outlook, Excel, Word, PowerPoint, OneNode ja Project. Pistikprogrammid saavad töötada erinevates keskkondades, mille koosseisu kuuluvad Windows, Mac, Ipad ja veeb. Pistikprogrammid toetavad enamjaolt sama palju funktsionaalsust, mida toetab tavaline veebilehekülg. Lisaks sellele võivad Office pistikprogrammid suhelda Office rakendusega ning töötada Office andmetega. [1]

Pistikprogrammid koosnevad kahest komponendist, veebileheküljest, ja XML failist. XML failis on märgitud, kuidas pistikprogramm integreerub Office rakendusega. Lisaks on XML failis märgitud ka pistikprogrammiga seotud andmed nagu pistikprogrammi nimi, ID, kirjeldus ja versioon. XML failis on kirjas ka pistikprogrammi andmetele juurdepääsu nõuded. Tähtsaks pistikprogrammi osapooleks on ka veebilehekülg, mis kuvatakse Office rakenduse osana ja mis peab eelnevalt olema hostitud veebiserveris. [1]

3.3.1 Laiendamisvõimalused

Laiendamisvõimalused varieeruvad erinevate office rakenduste vahel. Põhilisi laiendamise võimalusi, mis on toetatud igas office rakenduses, on kaks. Esimeseks põhiliseks laiendamise võimaluseks on lindi nupud (Ing. ribbon buttons). Neid kuvatakse Office tegevusreal ja nende kaudu saab, kas käivitada funktsiooni või avada teist laiendamisvõimalust, milleks on tegumiriba (Ing. task pane). Tegumiriba on laiendamise viis Office rakendustele, mille puhul avatakse rakendusel riba, mis võimaldab kasutajaga suhelda. Riba oma olemuselt on tavaline veebilehekülg, mida avatakse Office rakenduse

sees ja, millel on võimalus suhelda nii Office rakenduse, kui ka kasutajaga. Peale kahe põhilise laiendamisvõimaluse on ka need võimalused, mis eksisteerivad ainult konkreetsetes Office rakendustes, näiteks Outlook lubab lisa laiendamisvõimalust, mille puhul pistikprogramm kuvatakse kontekstiliselt Outlook'i objekti kõrval. Teisisõnu, toetab Outlook pistikprogrammi kuvamist otse kirja sees. Excel ja PowerPoint võimaldavad luua uusi objekte pistikprogrammi kaudu. Loodud objektide kaudu saab teha näiteks andmete visualisatsiooni, meedia näitamist või teist välist integratsiooni. [1]

3.3.2 Tehnoloogiad

Pistikprogramm Office'i jaoks kirjutatakse kahel erineval viisil, mille puhul mõlemad kasutavad HTML, CSS ja Javascript'i. Esimeseks võimaluseks on luua pistikprogramm kasutades selleks Yeoman generaatorit. Yeoman generaator tekitab pistikprogrammile põhja, mille alusel saab kirjutada pistikprogrammi. Genereerimise käigus määratakse raamistik, mida kasutatakse arendamiseks. Toetatud raamistikkudeks on React ja Angular. Generaator toetab muuhulgas ka tavalise HTML, CSS ja Javascript põhja genereerimist. Lisaks raamistikule, saab genereerimise protsessis määrata arendamise keelt, milleks on kas Javascript või Typescript. Yeoman generaator toetab pistikprogrammide loomist järgmistele Office rakendustele:

- Excel,
- OneNote,
- Outlook,
- PowerPoint,
- Project,
- Word. [25]

Muuhulgas saab luua Office'i jaoks pistikprogrammi ka kasutades Visual Studio't. Selle käigus on pistikprogramm loodud Visual Studio lahenduse osana. Microsoft dokumentatsioonis on välja toodud, et kuigi pistikprogrammi saab luua kasutades Visual Studio't, ei ole see soovitatav. Eelkõige soovitatakse Office pistikprogrammide loomiseks kasutada Yeoman generaatorit, mis pakub paremat arendamise kogemust,

rohkem pistikprogramme genereerimise võimalusi ja projekti malle uuendatakse sagedamini. [25]

3.3.3 Liides

Pistikprogrammi suhtlemine Office'iga käib läbi Office objekti. Office objekti kaudu saab pistikprogramm ligipääsu andmetele ja meetoditele, mis erinevad vastavalt kasutatud Office rakendusele. Vaatamata valitud Office rakendusest, pakub Office põhiandmeid ja -meetodeid, mis eksisteerivad igas Office rakenduses. Põhiandmeteks on näiteks andmed sisseloginud kasutajast. Lisaks, omavad sarnased Office rakendused muuhulgas ka sarnaseid meetodeid ja andmeid. Näiteks Word, PowerPoint ja Excel töötavad ühe dokumendiga, mistõttu pakutav funktsionaalsus on nende puhul sarnane ning selle hulka kuulub:

- Dokumendi (erinevates vormingutes) ja sellega seotud andmetele ligipääs,
- kasutaja informatsiooni ligipääs,
- kasutaja poolt valitud informatsiooni ligipääs,
- dokumendis navigeerimine,
- dokumendi muutmine. [26]

Outlook'is funktsionaalsus erineb ning funktsionaalsuse hulka kuuluvad järgmised funktsionaalsused:

- avatud kirja informatsiooni ligipääs,
- kasutaja informatsiooni ligipääs,
- uue kirja loomise protsessi käivitamine,
- kirjade vaheline navigeerimine,
- kirja koostamisel kirja muutmine. [27]

3.3.4 Autentimise võimalused

Pistikprogramm saab autentida kasutajaid kasutades kõiki samasuguseid autentimisvõimalusi nagu näiteks tavaline veebilehekülg. Autentimine saab toimuda

kasutades nii Microsoft`i teenust, kui ka teisi autentimise teenuse pakkujate teenusi. Autentimine saab olla realiseeritud näiteks kasutades järgmisi teenuse pakkujaid: Google, Facebook, LinkedIn, SalesForce või Github. Üheks kasutajat autentimise viisiks soovitatakse Office pistikprogrammides kasutada ühekordset sisselogimise tehnoloogiat (Ing. SSO). Ühekordne sisselogimine on kiire ja mugav viis kasutaja autentimiseks, kuna antud protsess ei nõua kasutajal andmete sisestamist ja ei ava autentimise aknaid. Lisaks, võimaldab ühekordne sisselogimine vahetada saadud tokeni teise vastu, mille abil saab sooritada andmete päringut sellises Microsofti teenuses, nagu Graph API. Vaatamata ühekordse sisselogimise eelistusele, ei ole see toetatud kõikides Office versioonides ning seetõttu kasutades ühekordset sisselogimist peab olema implementeeritud ka varu autentimise viis, mida saaks kasutada juhul, kui ühekordne sisselogimine oleks võimatu. [28]

3.3.5 Avaldamine

Pistikprogrammide avaldamiseks on erinevaid võimalusi. Pistikprogrammi avaldamine võib olla teostatud kasutades järgmisi platvorme:

- AppSource,
- Microsoft 365 Admin keskus,
- SharePointi kataloog,
- vahetusserver (Ing. Exchange server). [29]

AppSource on platvorm, kus iga Office kasutaja saab näha ja lisada pistikprogramme enda rakendustele. Peale AppSource'i on lisaks olemas Microsoft 365 App Store, kus sarnaselt AppSource'ile avaldatakse pistikprogramme. Esitades enda rakendust ühele nendest platvormidest, ilmub see automaatselt teises. [30] Pistikprogrammi esitamine AppSource'i peab olema tehtud spetsiaalsest arendamise kontost ning esitamise käigus peab rakendus läbima sertifitseerimisprotsessi. Sertifitseerimise protsessi käigus testitakse kogu rakenduse funktsionaalsust erinevates keskkondades. Samuti vaadeldakse, et pistikprogramm ei sisaldaks vastuvõetamatut või solvavat materjali ega viiruseid, oleks grammatiliselt korrektne ning kogu rakendus töötaks vastavalt dokumentatsioonile. Peale seda, kui rakendus on võetud AppSource'i, iga rakenduse muudatus peab uuesti läbima sertifitseerimisprotsessi. [31]

Microsoft 365 Admin keskus võimaldab ettevõtete administraatori õigustega kasutajatel anda juurdepääsu pistikprogrammile kõikidel ettevõttes olevatele kasutajatele. Avaldamise viis ei nõua kasutajatelt seadistust, kuna pistikprogramm on kohe kättesaadav esitamise järgselt. [29]

Sharepointi kataloog on veebi lehekülg, mida saab luua Word, Excel ja PowerPoint pistikprogrammi avaldamiseks. Platvorm ei toeta uute pistikprogrammide funktsionaalsust. [29]

Vahetusserver on Outlook'i pistikprogrammide avaldamise viis, mille kaudu saab lisada Outlook'i pistikprogrammi, kas kasutades pistikprogrammi XML faili või url lingi XML failile. [29]

3.4 Pilvandmetöötluse platvormid

Pilvandmetöötluse platvormid on platvormid, mis pakuvad suurehulgalisi teenusi. Pakutavate teenuste hulka kuuluvad teenused nagu: infrastruktuur, platvormid ja tarkvara. Pilvandmetöötluse platvormide kasutamine säästab raharessurse, kuna kõrvaldab ostmise ja enda infastruktuuri hooldamise vajadust ning võimaldab kasutada valmis platvorme ja tarkvara lahendusi. [32]

3.4.1 Microsoft Azure

Microsoft Azure kujutab pilvandmetöötluse platvormi, mis pakub suurehulgalisi teenusi. Platvormi hinna moodustamise süsteem on *pay-as-you-go*, ehk teisisõnu moodustatakse hind vastavalt kasutatud ressurssidele. Platvorm pakub neli erinevat pilvandmetöötluse vorme:

- infrastruktuur teenusena (Ing. Infrastructure as a service),
- platvorm teenusena (Ing. Platform as a service),
- tarkvara teenusena (Ing. Software as a service),
- serverivaba. [33]

3.4.2 AWS

AWS ehk Amazon Web Services on pilvandmetöötluse platvorm. Platvormi eeliste hulka kuuluvad:

- suurehulgalised teenused ja nende suur variatiivsus, näiteks kõige laiem valik andmebaase;
- suur kasutajate kogukond;
- turvalisus, mis on võrdväärne pankade ja sõjaväe tasemetega. [34]

Teenuse hinna moodustamine sõltub konkreetsest teenusest. Hinna moodustamise süsteemid on järgnevad:

- *pay-as-you-go*,
- *save when you commit* ehk süsteem, mille puhul makstakse aja eest, mille jooksul teenust on kasutatud, mitte kasutatud ressursside eest;
- *pay less by using more* ehk mida rohkem teenust kasutatakse, seda väiksem on ressursi ühiku hind. Näiteks AWS S3 teenus, kus suurema andmemahu salvestamise puhul (ühe GB) hind langeb. [35]

3.5 Andmete säilitamine

Andmete säilitamise meetodi valimine on tähtis osa rakenduse loomises, mis otseselt mõjutab selliseid rakenduse aspekte nagu turvalisus, jõudlus või töökindlus. Andmete pärimise kiirus mõjutab kogu rakenduse tööd ning andmete kaotamisel või andmetele ligipääsu puudumisel võivad ettevõttel tekkida probleemid, mille lahendamisel võivad kuluda ulatuslikud rahalised ressursid. Andmete säilitamise viisi valimisel peab eelkõige lähtuma ettevõtte ning rakenduse nõuetest. Erinevatel andmete säilitamise meetoditel on omad tugevad ja nõrgad küljed, millega tuleb arvestada ja millele toetudes tuleb valida sobivate andmete salvestamise lahendust igale konkreetsele olukorrale. [36]

3.5.1 Andmebaas

Andmebaas on struktureeritud informatsiooni kogum, mille puhul andmete salvestamiseks kasutatakse reeglina ridu ja veergusid. Tänu andmete struktuursusele, saab teha kiireid päringuid, muutusi, uuendusi või kustutamisi. Andmebaaside kõige suurem eelis seisneb selles, et see võimaldab salvestada hulgalisi andmeid, mis võimaldab ettevõttel saavutada paremat skaleeruvust ja paindlikkust. Andmebaaside tüüpide koosseisu kuuluvad:

- relatsiooniline andmebaas. Levinuim andmebaasi tüüp, kus andmeid hoitakse tabelites kasutates ridu ja veergusid. Antud tüüp annab kõige paindlikkumat andmetega töötamise viisi;
- NoSQL andmebaas. Andmebaasi tüüp, mis võimaldab andmeid hoida struktureerimata või poolstruktureeritud kujul. Antud tüüp ei oma rangeid reegleid, kuidas andmed peaksid olema struktureeritud. [37]

3.5.2 Failisüsteem

Süsteem, mis hoiab faile lokaalselt arvutis ja vastutab nende säilitamise ja ligipääsu eest. Failisüsteem erineb sõltuvalt kasutatavast operatsiooni süsteemist, kuid selle põhimõte ja eesmärk on eri operatsiooni süsteemides analoogne. [38]

3.5.2.1 Azure failisüsteem

Azure veebirakenduste hostimise teenus (Azure Web App) pakub kaht erinevat võimalust salvestada andmeid failisüsteemi. Esimeseks võimaluseks on failide salvestamine püsivate failidena. Antud failide salvestamise viisil on salvestatud failid jagatud kõikide rakenduste eksemplaride vahel (kui toimub skaleerumine, käivitatakse rohkem rakenduse eksemplare, et koormusega hakkama saada). Salvestatavate failide limiit sõltub valitud teenuse paketist ning tasuta paketi puhul on failide salvestamise limiidiks 1GB. Standardpaket omakorda võimaldab hoiustada faile, mille maht ei ületa 50GB ketta peal. Teiseks viisiks on failide salvestamine ajutiste failidena, mille puhul salvestatud failid ei ole jagatud rakenduse erinevate eksemplaride vahel. Ajutised failid erinevad püsivatest failidest sellepoolest, et rakenduse peatamisel kõik ajutised failid kustutatakse. Ajutiste failide salvestamise limiit sõltub mitte ainult valitud paketist, vaid ka valitud SKU perekonnast. Tasuta või jagatud paketid võimaldavad hoiustada kuni

500MB ajutisi faile ketta peal. Samas, standardpaket koos teise valitud SKU perekonnaga võimaldab hoiustada kuni 15GB ajutisi faile ketta peal.[39]

Azure funktsioonid on serverivaba teenus [40], mis toetab nii püsivate kui ka ajutiste failide salvestamist. Püsivate failide limiit reeglina on 5TB ja salvestatud failid on jagatud kõigi rakenduse eksemplaride vahel. Ajutiste failide salvestamise limiit sõltub valitud paketist ning tarbimisplaani paketis ei saa ajutised failid ületada mahtu 500MB. *App service* ja *premium* paketid võimaldavad hoiustada kuni 11GB ajutisi faile. Lisaks, ei ole ajutised failid jagatud kõikide rakenduse eksemplaride vahel. [41]

3.5.2.2 AWS failisüsteem

AWS veebirakenduste hostimise teenus (AWS EC2) toetab failide salvestamist failisüsteemi ning salvestamise limiit sõltub valitud paketist. Limiit varieerub alates 32GB kuni 60TB. Salvestatavad failid on haavatavad ja eksisteerivad ainult sel juhul, kui rakendus töötab. Juhul, kui rakendus taaskäivitub või peatub, kustutatakse kõik selle rakendusega seotud andmed kettal. [42]

AWS Lambda, mis on serverivaba teenus, võimaldab salvestada kettal ainult ajutisi faile ning salvestamise limiidiks on 512MB. Püsivate failide salvestamine toimub ainult kasutades teisi teenuseid, näiteks AWS S3 või andmebaasi. [43]

3.5.3 Objektide salvestamise teenused

Objektide salvestamine (Ing. Object storage) on andmete salvestamise viis, mille puhul kõik andmed salvestatakse ühetasemelises keskkonnas, mida nimetatakse *storage pool*`iks. Objektidega salvestatakse objektiga seotud andmeid ja unikaalset identifikaatorit, mille abil saab salvestatud objekti salvestamise järgselt üles leida. Sellise salvestamise viisi eeliseks on suur skaleeruvus, mis võimaldab hoida andmeid petabaitides või ka eksabaidides. Lisaks, salvestatakse kõik andmed ühel tasemel ning seetõttu on andmete leidmine kiire ja lihtne. Salvestada saab meediat, audiofaile, veebilehekülgi, varukoopiaid ja teisi objekte, mis struktuuri poolest ei ole võimalik salvestada andmebaasides. Objektide salvestamise teenuse pakkujateks on näiteks AWS, Microsoft ja Google. [44, 45]

3.5.4 Vahemälu

Andmete salvestamiseks kasutatav tehnoloogia, mille raames salvestatud andmeid hoitakse mälus ning ei salvestata kettal. Kuna kõik salvestatud andmed hoitakse mälus, on selle tehnoloogia suurim eelis selle kiirus. Kõik andmetega seotud operatsioonid ei nõua andmete kettalt lugemist ning seetõttu andmetega töötamise operatsioonide aeg väheneb. Vahemälu puuduseks on väike võimalik andmete salvestamise maht, mis tuleneb andmete hoiustamisest mälus. Antud tehnoloogiat kasutatakse eelkõige taaskasutatava informatsiooni salvestamiseks. [46]

3.6 Autentimise teenused

Autentimine on protsess, mille tulemusena määratakse kasutaja. Autentimisteenuste põhimõtteks on autentimisloogika tsentraliseerimine, mille tulemusena muudetakse autentimine kättesaadavaks teenuseks ka teistele rakendustele. Autentimisteenuse kasutamisel toimub kasutaja autentimine rakenduse väliselt, mis lihtsustab rakenduse loomist. [47]

3.6.1 Microsoft Azure Active Directory

Microsoft Azure Active Directory (edaspidi AAD) on Microsofti poolt pakutav teenus, mille kaudu saab autentida kasutajat, kasutades Microsofti töö, kooli või isikliku kontot. Autentimisteenus toetab kasutajate autentimist serveri pärimiseks läbi rakenduse, kuid võimaldab pärida ka Microsoft teenusi nagu Graph API. [48]

Microsoft teenuse kasutamisel toimub autentimine kasutades juurdepääsu tokeneid (Ing. *Access token*). Neid väljastab Microsoft autentimisteenus ning oma olemuselt on need krüpteeritud assümeetrilise krüptoalgoritmiga isikuandmed. Token sisaldab endas erinevaid isikuandmeid, nagu e-maili, isiku täisnime jms. Tokenite verifitseerimiseks on vajalik Microsofti avaliku võtme kasutamine, mida saab pärida Microsoft'i serveritest. Võtmepaare, mida Microsoft kasutab tokenite krüpteerimiseks, püsivalt muudetakse ning Microsofti dokumentatsiooni kohaselt on soovitatav avaliku võtme uuendamine iga 24 tunni tagant. [49]

3.6.2 Amazon Cognito

Amazoni autentimisteenus, mis lisaks kasutajate autentimisele, toetab ka kontode loomist eraldi rakenduste kasutamiseks ning õiguste haldamist. Kasutaja autentimine toimub kas eraldi rakendusega seotud kontoga või kasutades selleks Facebook, Google, Amazon või Apple kontosid. Lisaks, toetab autentimisteenus kaheastmelis autentimist ja võimaldab kohandada logimisakent. [50]

3.7 Sarnane tarkvara

Vaatamata antud töö raames valmiva tarkvara eripäradele, on sarnaseid programme juba eelnevalt loodud. Antud töö raames vaadeldavateks sarnasteks tarkvaradeks on kaks Office keskkonna pistikprogrammi, milleks on Zoho ja DocuSign.

3.7.1 Zoho

Zoho on pistikprogramm Outlook'ile, mis võimaldab faile allkirjastada. Programm kujutab endast tegumiriba, mis avatakse Outlook'i osana ning mis võimaldab pistikprogrammiga suhelda. Pistikprogrammi toetatud platformideks on Outlook veebis, IOS, IPad OS ja Android. Zoho tarkvara Outlook'i jaoks pakub kahte erinevat funktsionaalsust. [51]

Esimeseks pakutavaks funktsionaalsuseks on failide allkirjastamine. Allkirjastamiseks on vajalik eelnevalt avatud kiri, millele on lisatud vähemalt üks fail. Kirjast on vajalik failide valimine, mida soovitakse allkirjastada ning seejärel toimub failide üles laadimine ja allkirjastamine Zoho süsteemis. [51] Seejärel lisab kasutaja käsitsi antud allkirja pildina dokumendile. Allkirja andmise järgselt saab allkirjastatud dokumenete saata meili teel koopiana või neid alla laadida. [52]

Teiseks pakutavaks funktsionaalsuseks on failide saatmine allkirjastamiseks. Funktsionaalsuse kasutamiseks peab olema eelnevalt avatud kiri, millele on lisatud vähemalt üks fail. Kirja failide seas valitakse konkreetsed failid, mida soovitatakse digiallkirjastada. Sellejärgselt sisestatakse isikute andmed, kes peavad faile allkirjastama. Kui kogu informatsioon on sisestatud, üles laetakse valitud failid Zoho süsteemi ja kiri saadetakse valitud isikuteni. Pärast kirja saatmist on võimalik muuta ja korrigeerida saadetud faile Zoho süsteemis. [51]

Enne kirja saatmist on võimalik järgmiste aspektide seadistamine:

- kirja saatmine võib toimuda valitud isikutele üheaegselt või konkreetses järjekorras, kus kiri saadetakse edasi, vaid sel juhul, kui eelmine isik on oma allkirja andnud;
- kirjale on võimalik sõnumi lisamine, mis järgnevalt saadakse kõikidele allkirjastajatele või ainult konkreetsele isikule;
- võimalik on allkirjastaja või lugeja/ülevaataja määramine;
- võimalik turvakoodi seadistamine iga allkirjastajale eraldi, mis saadetakse kirja saatja mobiiltelefonile. Kood sisestatakse Zoho süsteemi, et saada ligipääs failidele. [51, 53]

3.7.2 DocuSign

Office pistikprogramm, mis võimaldab allkirjastada faile nii Outlook'is kui ka Word'is. Antud programm võimaldab failide allkirjastamist kui ka failide saatmist allkirjastamiseks mõlemas keskkonnas. [54, 55, 56]

Outlook'is allkirjastamiseks peab olema eelnevalt avatud kiri, mis sisaldab vähemalt ühte faili. Allkirjastamise protsessi alustamiseks on vaja valida faile kirja manuses olevatest failidest, ning neid üles laadida DocuSign süsteemi. Seejärel lisab kasutaja käsitsi enda allkirja pildina dokumendile. Allkirja andmise järgselt saab allkirjastatud dokumente saata algsele adressaadile või neid alla laadida. [57]

Allkirjastamine Word'is võib olla teostatud keskkonda muutmata. Faili allkirjastamiseks on vaja tõsta allkirjade väljasid pistikprogrammist ning lohistada neid faili sobivatesse kohta. [56]

Failide saatmine allkirjastamiseks on identne nii Word kui ka Outlook keskkondades, kuid ainsaks erinevuseks on allkirjastatavate failide saamise meetod. Outlook keskkonnas valitakse allkirjastatavaid faile kirja manusest olevatest failidest. Word keskkonnas allkirjastatavaks failiks võetakse avatud fail. Failide valimisele järgneb nende üles laadimine DocuSign süsteemi. Failide üles laadimise järgselt on vajalik allkirjastajate ja/või ülevaatajate määratlemine, ning seejärel on võimalik kirja ära saata. [55, 58]

4 Analüüs

Antud töö osas analüüsitakse eelnevalt leitud infoallikaid ning leitud tehnoloogiatest valitakse need, mida hakatakse edaspidi töö probleemi lahendamiseks kasutama. Tehnoloogiate valik on põhjendatud ning tugineb töö rakenduse nõuetel. Lisaks, on antud töö osas detailselt kirjeldatud rakenduse nõuded ning võrreldud eeldatav töö tulemus sarnaste rakendustega.

4.1 Digiallkirjastamine

Vaadeldava töö kirjanduse ülevaates on kirjeldatud erinevad elektroonse allkirja tasemed, kuid antud töö raames kontsentreerutakse vaid kõige turvalisemal allkirja tasemel, milleks on kvalifitseeritud allkirjad. Ülejäänud elektroonsed allkirjad loetakse ebaturvalisteks, mistõttu ei ole need võrdväärsed käsitsi antud allkirjale. Töö raames valminud lahendust hakatakse prognoositavalt kasutama avalikus sektoris, mistõttu on vajalik, et digiallkirjad oleksid väärtuslikud ning analoogsed käega antud allkirjadega.

Kirjanduse ülevaates on vaadeldud kahte võimalust failide digiallkirjastamiseks, mis mõlemad sobivad püstitatud probleemi lahendamiseks. Digiallkirjastamise teenus Dokobit võimaldab programeerijal kontsentreeruda teiste rakenduse funktsionaalsuste loomisel, kuna digiallkirjastamise teenus funktsioneerib iseseisvalt. Teenus on paindlik ja ei nõua eelnevaid teadmisi digiallkirjastamisest, mis teeb teenuse mugavaks lahenduseks tarkvara arendamise vaatenurgast. Teenusel on ka puudused, mille hulka kuuluvad teenuse kasutamise kõrge hind ja digiallkirjastastamise sooritamine vaid Dokobit keskkonnas.

Teiseks vaadeldud digiallkirjastamise võimaluseks on kasutada SK teenuse pakkujat. Teenus nõuab rohkem arendusressursse arendajalt, kuid võrreldes eelmisega on antud viis odavam ning võimaldab digiallkirjastada keskkonda muutmata. Smart-ID`ga ja Mobiil-ID`ga digiallkirjastamise puhul võimaldab SK teenus saada ligipääsu sertifikaatidele ja krüpteerida rakenduselt saadud räsi digiallkirjastaja privaatvõtmega. Lisaks, võimaldab teenus saada ligipääsu ajatemplile. Rakendus, mis kasutab SK teenust, peab teostama mõningaid digiallkirjastamisega seotud ülesandeid. Näiteks, peab rakendus arvutama digiallkirjastatavatest failidest räsi ja koostama digiallkirjastatava konteineri. Rakenduse koosseisus teostatavate digiallkirjastamisega seotud ülesannete

tegemiseks on võimalik kasutada digiallkirjastamise teeki, mis lihtsustab nende ülesannete teostamist. Vaadeldava töö raames plaanitakse edaspidi kasutada SK ID Solutions teenust koos digiallkirjastamise teegiga. Kuigi võrreldes Dokobit teenusega, nõuab valitud digiallkirjastamise viis rohkem arendamise viisile kulutatud ajaressursi, aitab see tuleviku perspektiivis raharessursse kokku hoida. Lisaks, on antud digiallkirjastamise viis paindlikum ning võimaldab digiallkirjastada keskkonda muutmata.

4.2 Pilveteenuse pakkuja

Antud töö raames valitakse pilveteenuse pakkujat Microsoft Azure ja AWS vahel. Mõlemad platvormid sisaldavad suurehulgaliselt pakutavaid teenusi ja omavad ulatuslikku kasutajate baasi, kuid antud töö raames plaanitakse kasutada Azure pilveteenust. Valitud pilveteenuse pakkujaks on Azure, kuna töö autor omab kogemust Azure teenuste kasutamise ja administreerimisega.

4.3 Kasutajaliides

Seoses sellega, et töö tulemusena valmiv lahendus peab võimaldama digiallkirjastada faile Office keskkonnas, on kasutajaliidese valik piiratud. Office toetab pistikprogramme, kuid nende loomiseks on võimalik vaid Yeoman generaator või Visual Studio kasutamine. Seevastu on Microsoft dokumentatsioonis märgitud, et Visual Studio kasutamine Office pistikprogrammi loomiseks on „aegunud“, mistõttu on töö raames kasutatud pistikprogrammi loomiseks Yeoman generaatorit.

4.3.1 Laiendamine

Põhilisteks Office laiendamise võimalusteks, mis on toetatud igas Office rakenduses, on lindi nupud ja tegumiriba, kuid igal rakendusel on ka enda laiendamisvõimalused. Selleks, et antud töö raames valmiv lahendus oleks paindlik ning seda saaks kasutada mitmes Office rakenduses, on vajalik, et valmiv lahedus kasutaks põhilisi laiendamise võimalusi ning, seega töötaks nii Excel’is, PowerPoint’is, Word’is kui ka Outlook’is. Esimeseks põhilaiendamise võimaluseks on lindi nuppud, kuid antud töö raames ei ole mõistlik antud lahendus, kuna lindi nupud ei võimalda kasutajal sisendit saada. Valmiv rakendus aga peaks saama kasutajalt isikuandmeid digiallkirjastamiseks, mis lindi nuppude kasutamisel ei ole teostatav. Teiseks põhilaiendamise võimaluseks on

tegumiriba, mis avaneb Office rakenduse osana ja toetab kasutajalt sisendi saamist. Lähtuvalt varem kirjutatust, hakatakse töö raames valmiva programmi kirjutamisel kasutama tegumiriba, Office rakenduste laiendamiseks.

4.3.2 Tehnoloogia

Yeoman generaatori abil saab luua pistikprogrammi, kas ainult HTML, CSS ja Javascript'i kasutades, kui ka kasutades kahte raamistikku, milleks on React ja Angular. Töö raames valmivat lahendust hakatakse kasutama töö autori töökohal ning seetõttu, toetutakse tehnoloogia valikus töökohas kasutatud tehnoloogiatele. Töökoha tehnoloogia kasutamine on eelkõige vajalik selleks, et valmivat pistikprogrammi saaksid tulevikus muuta ja/või täiendada sealhulgas ka töö autori kolleegid. Sellest tulenevalt, kasutatakse töö raames pistikprogrammi kirjutamiseks React tehnoloogiat, kuna töö autor on sellega eelnevalt kokku puutunud oma töökohal. Raamistiku valik on tehtud tavalise HTML, CSS ja Javascript tehnoloogiate vahel, kuna raamistik lihtsustab rakenduse arendamise protsessi ning hõlbustab mitmeid ülesandeid, mistõttu rakenduse loomisele kuluv aeg väheneb.

4.3.3 Levitamine

Iga allkirja andmisega kaasnevad kulud, mistõttu ei soovi töö autor avalikustada pistikprogrammi piiranguteta. Seetõttu valminud lahenduse levitamine võib olla teostatud, kasutades näiteks AppSource'i. Selle kasutamisel oleks igal kasutajal digiallkirjade limiit, mille värskendamine toimuks iga kindla ajaperioodi tagant. Teiseks pistikprogrammi levitamise võimaluseks võib olla programmi levitamine läbi Admin keskuse. Admin keskus võimaldab administraatori õigustega kasutajatel anda pistikprogrammi kasutamisõigused kõikidele, ettevõttes olevatele kasutajale, mis teeks antud variandi väga mugavaks programmi edasi müümise võimaluseks. Admin keskuse kasutamisel, ei oleks vajalik limiidi määramine, vaid ettevõtte arveldamine toimuks kindla ajaperioodi vältel. Lisaks on ka kolmas võimalus, mis sisaldaks kahe esimese variandi koostööd. Levitamine oleks teostatud AppSource'i kaudu, kuid limiit oleks määratud vaid neile, kes ei maksa pistikprogrammi kasutamise eest. Antud töö raames pistikprogrammi levitamine hakkab toimuma läbi Admin keskuse, kuna AppSource`il on rangeid levitamise nõude kriteeriumid, mida töö autor ei suuda antud töö raames realiseerida.

4.4 Autentimine

Pistikprogramm hakkab töötama Office rakenduse osana, kus sarnaselt pistikprogrammile, toimub kasutaja autentimine. Tulenevalt sellest, et väline rakendus on kasutajad eelnevalt autentinud, oleks eelistatud autentimise lahenduseks, välise rakenduse autentimise andmete taaskasutamine. Antud autentimise viisi nimetatakse ühekordseks sisselogimiseks ning antud töö raames kasutatakse sellist meetodit kasutaja kogemuse tõstmiseks, kuna see ei nõua taaslogimist. Microsoft dokumentatsioonist tulenevalt, ei toeta ühekordset sisselogimist kõik Office rakenduste versioonid, mistõttu peab kasutajate autentimiseks kasutama lisaks varu autentimise süsteemi. Töö raames hakatakse varu autentimise süsteemiks kasutama Microsofti AAD autentimis teenust. Autentimise teenuse valik on tehtud toetudes keskkonnale, milles rakendus hakkab töötama. Microsofti poolt pakutava autentimise teenuse puhul toimub autentimine kasutades kontot, millega kasutaja autentib ennast Office rakenduses. Lisaks, kasutades Microsofti poolt pakutavat teenust, peab antud töö server toetama vaid Microsofti poolt väljastatud tokeneid. Kasutades Amazon Cognito´t oleks vajalik rakenduse töötamine kahe erineva tokeniga, mis omakorda muudaks süsteemi keerulisemaks. Autentimise teenuste kasutamise asemel, võiks kasutajat autentida ka kasutades eraldi rakenduses olevat autentimise süsteemi. Teisisõnu, teha autentimise süsteem rakenduse osana ning nõuda kasutajalt nii konto loomist, kui ka sisselogimist pistikprogrammi. Antud töö raames on valitud autentimise teenuse kasutamine, kuna kasutajale oleks antud viis mugavaim. Valminud pistikprogrammi planeeritav funktsionaalsus ei ole suur ja eraldi konto tegemine ei oleks autori arvates mõistlik. Lisaks, uue konto tegemata jätmine aitaks tagada kasutaja isikuandmete turvalisust ning väldiks lisateabe meeldejätmist.

4.5 Server

Lähtudes ülesande püstitusest ja sellest, et autentimiseks hakatakse kasutama Microsofti teenust, ei ole antud töö raames eeldatava serveri maht suur. Lisaks, toetudes sellele, et kasutajaliidese levitamise viisiks on valitud meetod, mille puhul kasutajate arv on piiratud, hakkab serveri koormus olema alguses väike. Vaatamata esialgsetele tingimustele, tuleb serveri tehnoloogiate valimisel arvestama ka võimalike tingimuste muudatustega tulevikus. Serveri üheks tähtsaks omaduseks, millega tuleb arvestada serveri tehnoloogiate valimisel, on serveris kasutatavad krüptograafilised võtted, mis nõuavad rohkem ressursse võrreldes tavapäraste serveri operatsioonidega.

4.5.1 Arhitektuur

Serveri arhitektuuri valik tuleneb otseselt serveri nõuetest. Serveri väiksest mahust tulenevalt, ei osutu mikroteenuste arhitektuur antud töö kõige tõhusamaks lahenduseks. Kogu server hakkab olema üks teenus mikroteenuste arhitektuuris, mille funktsionaalsuseks on digiallkirjastamine. Monoliitne arhitektuur, lähtuvalt serveri nõuetest, sobiks paremini, kuid antud töö serveri kirjutamiseks hakatakse kasutama serverivaba arhitektuuri, mida põhjendatakse väiksema raharessursi kulumisega. Lisaks, on serverivaba arhitektuuri eelis, et see ei kuluta resursse päringu ootamisele, kuna väikese esiaglse koormuse tõttu võib tekkida olukord, kui monoliitse arhitektuuri kasutamisel hakkab kuluma rohkelt ressursse päringu ootamisele. Samas, serverivaba arhitektuuri kasutamisel ei teki olukorda, mille korral serverile ei piisa ressursse ning selle skaleeruvus toimub automaatselt.

4.5.2 Programmeerimiskeele valik

Toetudes eelnevale, et osa digiallkirjastamise protsessist toimub eeldatavalt serveril, koondub serveri programmeerimiskeele valik vaid nendele keeltele, mis on toetatud digiallkirjastamise raamistikutega. Digiallkirjastamiseks vajaliku funktsionaalsust on võimalik kirjutada iseseisvalt, mis ei ole ratsionaalne, kuna antud viis eeldab sügavaid krüptograafilisi teadmisi ja kulutaks rohkelt ajaressursse. Töös käsitletud digiallkirjastamise teekide toetavateks keelteks on C++, Java ja C#. Toetatud keeltest ei ole valitud C++, kuna keel ei ole levinud ning töö autoril puudub keelega töötamise kogemus. Java ja C# on sarnased keeled, kuid lähtudes sellest, et pilveteenuse platvormiks on valitud Microsoft Azure, hakatakse kasutama C#. Microsoft Azure toetab nii C# kui ka Java't, kuid C# on Microsofti poolt loodud keel ning selle toetus, kui ka kasutus, on suurem Azure teenuste puhul.

4.6 Rakenduse nõuded

Antud töö raames valmiv lahendus peab võimaldama failide digiallkirjastamist nii Outlook'is, Word'is, Excel'is, kui ka PowerPoint'is. Rakenduses toetatud digiallkirjastamise viisid on: Mobiil-ID, Smart-ID ja ID-kaart. Digiallkirjastamine peab toimuma nii veebipõhises keskkonnas, kui ka eraldiseisvas rakenduses. Lisaks, peab olema digiallkirjastamine realiseeritud Office keskkonda muutmata. Digiallkirjastamise lõpus peab olema digiallkirjastatud konteineri alla laadimise võimalus.

Word’is, Excel’is ja PowerPoint’is digiallkirjastatakse avatud dokumenti. Failide digiallkirjastamine peab olema toetatud nii esialgses Office vormingus (.doc, .xlsx, .pptx), kui ka PDF vormingus. PDF vormingu toetus on oluline, kuna PDF vorming on universaalne ning ei sõltu operatsioonisüsteemidest või rakendustest. [59]

Outlook’is digiallkirjastatakse kirjaga seotuid faile. Ühe kirjaga võib olla seotud rohkem kui üks fail, mistõttu peab rakendus toetama nii ühe faili, kui ka failide kogumi digiallkirjastamist. Pärast digiallkirja protsessi lõpetamist Outlook’i keskkonnas peaks lisaks digiallkirjastatud konteineri alla laadimisele, olema võimalus uue kirja alustamiseks, kus digiallkirjastatud konteiner oleks vaikimisi kirja juurde lisatud.

Lisaks tavalisele allkirjastamisele peab Outlook’is olema lisafunktsionaalsus, mis võimaldab uue kirja koostamisel määrata järgmisi aspekte:

- faile, mida hakatakse allkirjastama. Kirja koostamisel saab kirjaga seotud failidest valida konkreetseid faile, mida soovitakse allkirjastada. Piiranguks oleks vähemalt ühe faili valimine;

- isikuid, kes peavad kirja allkirjastama. Kirjal võib olla mitu saajat ning nende seast saab valida konkreetseid isikuid, kes peavad faile allkirjastama. Piiranguks oleks vähemalt ühe isiku valimine. Lisaks eelnevale, oleks piirang, et valitud kasutajal oleks nii Outlook, kui ka selle töö raames valminud pistikprogramm selleks, et kasutaja saaks saadud kirja digiallkirjastada rakenduse raames;

- isikuid, kes saavad digiallkirjastatud faile peale seda, kui kõik isikud, kes pidid digiallkirja andma, on selle andnud. Siinpuhul peab olema võimalus, kus lisatakse uus e-mail, kuhu ei saadeta koostatavat kirja, vaid kuhu saadetakse digiallkirjastatud konteiner. Piiranguks on vähemalt ühe isiku valimine.

Viimases punktis kirjeldatud funktsionaalsust hakatakse edaspidi kutsuma jälitatud kirjaks, kuna server jälgib kirja seisundit ning kõikide digiallkirjade andmise järgselt, saadab server kirja koos digiallkirjastatud konteineriga kõikidele, selleks enne määratud isikutele.

Rakendus salvestab faile digiallkirjastamiseks, kuid rakendus peab failidest vabanema niipea, kui võimalik selleks, et maandada turvariske ning vabastada mälu. Digiallkirjastamise protsessi järgselt peavad failid olema kustutatud, kas kohe pärast

konteineri alla laadimist serverist või teatud aja jooksul, kui digiallkirjastamine on lõpetatud. Juhul, kui failid on seotud jälitatud kirjaga, peab server kustutama antud faile kohe pärast jälitatud kirja digiallkirjastamise lõpetamist.

4.7 Andmete salvestamine

Töö eeldatav lahendus peab salvestama nii faile kui ka andmeid, mis oleksid seotud jälitatud kirjadega. Failide salvestamise võimalusteks on vaadeldud faili süsteemi ja objektide salvestamise teenust. Lähtudes sellest, et valitud pilve teenusplatvormiks on Azure ning serveri arhitektuuriks on valitud serverivaba arhitektuur, hakatakse antud töö rakenduse kirjutamiseks kasutama Azure funktsioone (Ing. Azure functions). Ajutiste failide salvestamine on sobimatu, kuna skaleerumisel ei ole ajutised failid jagatud kõikide rakenduste eksemplaride vahel. Vaikimise faili süsteemi maht püsivate failide salvestamiseks on Azure funktsioonides 5TB. Autori arvates, piisab püsivate andmete salvestamiseks Azure funktsioonide salvestamise ruumist, kuna rakendus hakkab faile kustutama esimesel võimalusel. Seetõttu ei ole üheaegne failide salvestatud maht suur. Teisest küljest, objektide salvestamise teenus toetab suurema andmemahu salvestamist, kuid vajab ka eraldi seadmistamist ja teeb kogu rakenduse struktuuri keerulisemaks. Lähtudes sellest, et rakenduses ei ole üheaegselt salvestatud suur hulk faile, hakkab rakendus kasutama failide salvestamiseks faili süsteemi.

Jälitatud kirjaga seotud andmete salvestamine hakkab toimuma kasutades selleks andmebaasi, kuna jälitatud kirjaga seotud andmed omavad struktuuri, mis võimaldab neid andmebaasi salvestada. Antud töö andmebaasiks hakatakse kasutama PostgreSQL andmebaasi, kuna töö autori töökohal on see kõige kasutatavam andmebaas ning töö autoril on selle administreerimise kogemust.

Lähtudes sellest, et rakenduse autentimise teenuseks on valitud Microsoft`i autentimise teenus, võiks rakendus kasutada vahemälu Microsoft avalikke võtmetete salvestamiseks. Avalikke võtmete salvestamine, kasutades selleks vahemälu, oleks hea viis Microsoft avalikute võtmetega töötamiseks, mis väldiks liigseid päringuid Microsofti serverisse. Lisaks, võimaldaks vahemälu kasutamine salvestada avalikke võtmeid, nende kiireks ligipääsuks. Rakenduse iga protsessi alamprotsessiks hakkab olema autentimine ning seetõttu, autentimise teenuse avalikke võtmete kiire kättesaadavus teeks kogu

rakenduse tööd kiireks. Vaatamata vahemälu eelistustele, on vahemälu tasuline ning selle kasutamine vaid Microsoft võtmete salvestamiseks ei ole autori arvates ratsionaalne.

4.8 Võrdlus sarnase tarkvaraga

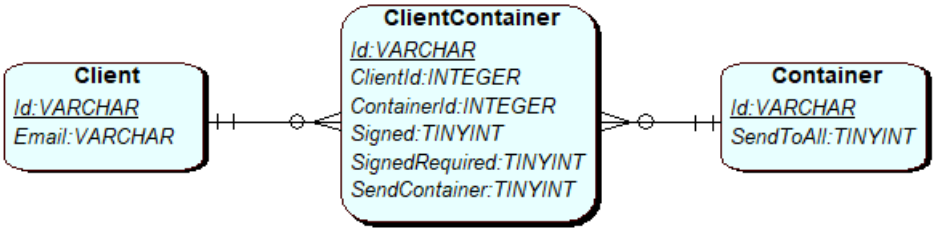
Töö eeldatava lahenduse funktsionaalsus on identne töös kirjeldatud sarnase tarkvara funktsionaalsusega. Nii töö eeldatav rakendus kui ka töös kirjeldatud sarnane tarkvara toevad nii failide allkirjastamist kui ka failide saatmist allkirjastamisele. Töö eeldatava rakenduse kõige suurem eelis, võrreldes teiste programmidega, seisneb antavate allkirjade tasemel. Eeldatavalt hakkab rakendus kasutama kõige turvalisemat digiallkirja taset, milleks on kvalifitseeritud elektroonilised allkirjad. Samas, erinevalt töö raames kirjeldatud sarnasele tarkvarale, hakkab eeldatav pistikprogramm toetama suuremat kogust Office rakendusi. Eeldatav lahendus hakkab töötama nii Outlook'is, Word'is, Excel'is kui ka PowerPoint'is, kuid vaadeldav sarnane tarkvara toetab maksimaalselt kaht Office platvormi. Lisaks, võimaldab töö eeldatav rakendus digiallkirjastada faile keskkonda muutmata. Töö eeldatav rakendus omab ka puudusi võrreldes sarnaste rakendustega. Näiteks, töö eeldatav rakendus ei hakka toetama kõiki emaili platvorme. Sarnane tarkvara võimaldab digiallkirjastada faile nii Outlook keskkonnas kui ka teistes e-maili platvormides. Erinevate platvormide toetus on saavutatud seetõttu, et allkirjastamine ei toimu e-mail platvormi keskkonnas, vaid eraldi keskkonnas. Keskkonna muutmine failide allkirjastamiseks on see konkreetne probleem, mida antud töö proovib lahendada selleks, et allkirjastamise protsessi lihtsustada. Lisaks, erinevalt DocuSign lahendusele, eeldatava pistikprogrammi Word keskkonna funktsionaalsus, ei hakka toetama faili saatmist allkirjastamiseks teistele isikutele.

5 Kavand

Antud töö osas vaadeldakse andmebaasi, serveri ja pistikprogrammi struktuuri ning rakenduse andmevoogu. Samas, antud töö peatükis on detailselt kirjeldatud kogu rakenduse eeldatav funktsionaalsus.

5.1 Andmebaasi struktuur

Antud töö andmebaasi olemi-suhte diagramm on esitatud joonisel 2. Seoses sellega, et kasutajate autentimine hakkab toimuma kasutades Microsofti teenust ning tavalise digiallkirjastamise protsessi puhul ei ole vajadust andmeid andmebaasi salvestada, hakatakse andmebaasis salvestama vaid neid andmeid, mis on seotud jälitatud kirjadega. Andmebaas hakkab koosnema kolmest tabelist:



Joonis 2. Andmebaasi olemi ja suhete diagramm

- Client - tabel, kus säilitatakse informatsioon kõikidest jälitatud kirjaga seotud isikutest. Autentimise protsessi tulemusena saab rakendus ligipääsu kasutaja e-mailile, mille kasutamisel hakkab rakendus eristama erinevaid kliente enda süsteemis. Tabelis väli nimetusega Email on unikaalne ning tabelis salvestatakse igat kasutajat vaid ühekordselt.

- Container - tabel, kus hoitakse infot digiallkirjastatavatest konteineritest. Tabelis salvestatakse digiallkirjastatavad konteinereid koos lisaandmetega. Rakenduse lisaandmete väljaks on väli SendToAll, mis näitab, kas digiallkirjastatava konteinerit saadetakse kõikidele konteineriga seotud isikutele või ainult määratud isikutele.

- ClientContainer - tabel, kus hoitakse viiteid Client tabelile ja Container tabelile. Tabel on vajalik seetõttu, kuna ühe konteineriga võib olla seotud rohkem kui üks isik ning üks isik saab omakorda olla seotud rohkem kui ühe konteineriga. Tabelis,

lisaks viidetele, hakatakse hoiustama andmeid, mis on seotud kliendi ka konteineri seosega. Lisaandmete väljad koos kirjeldusega on järgnevad:

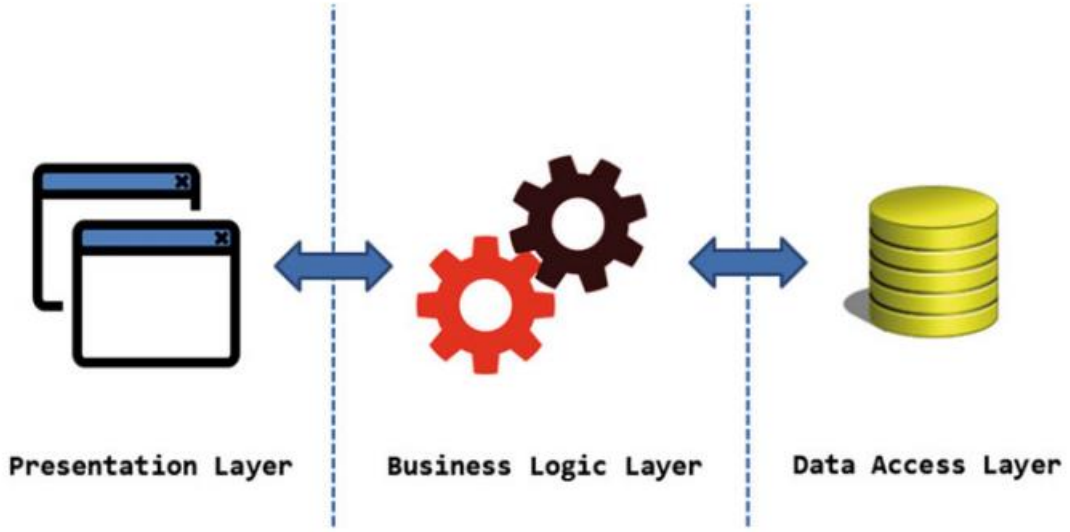
- Signed väli tähistab, kas konteiner on digiallkirjastatud kasutaja poolt või mitte;
- SignedRequired tähistab, kas digiallkirjastamine antud kasutajale on määratud või mitte;
- SendConteriner tähistab, kas konteineri saatmine kasutajale on vajalik või mitte.

5.2 Server

Antud töö serveri arhitektuuriks oli valitud serverivaba arhitektuur. Vaatamata valitud serveri arhitektuurile, mis moodustab ühe terviku, võib serverit loogiliselt jaotada funktsionaalsusteks. Selleks, et server toetaks kõiki analüüsi osas kirjeldatuid rakenduse nõudeid, peab server toetama kõiki funktsionaalsusi, mis on järgnevalt väljatoodud:

- digiallkirjastamine. Peamine rakenduse funktsionaalsus, mis võimaldab failide digiallkirjastamist. Funktsionaalsuse realiseerimiseks hakatakse kasutama SK ID Solutions teenuse pakkujat koos digiallkirjastamise teegiga;
- kirjade saatmine. Funktsionaalsust hakatakse kasutama vaid jälitatavate kirjadega konteinerite saatmiseks.
- failisüsteemiga töötamine. Funktsionaalsust hakatakse kasutama digiallkirjastatavate failide salvestamiseks;
- Andmebaasiga suhtlemine. Funktsionaalsust hakatakse kasutama jälitatavate kirjadega seotud andmete salvestamiseks;
- Päringute saatmine. Funktsionaalsus, mis hakkab moodustama peaaegu kõikide teiste funktsionaalsuste alam funktsionaalsust. Lisaks, kasutades päringute saatmist, hakab server saama ligipääsu Microsoft`i avaliku võtmele, mis on vajalik kasutaja poolt autentimise tokenite verifitseerimiseks.

Antud töös erinevate funktsionaalsuste loogiliseks eraldamiseks kasutatakse kihilist struktuuri, mis on demonstreeritud joonisel 3 ning, mille puhul hakkab server koosnema kolmest kihist. Esimeseks kihiks on kontrollerite kiht, mis võtab vastu päringuid. Teiseks kihiks on teenuse kiht, kus hakkab paiknema rakenduse loogika. Viimaseks kihiks on andmebaasiga suhtlemise kiht.



Joonis 3. Serveri kihiline struktuur [60]

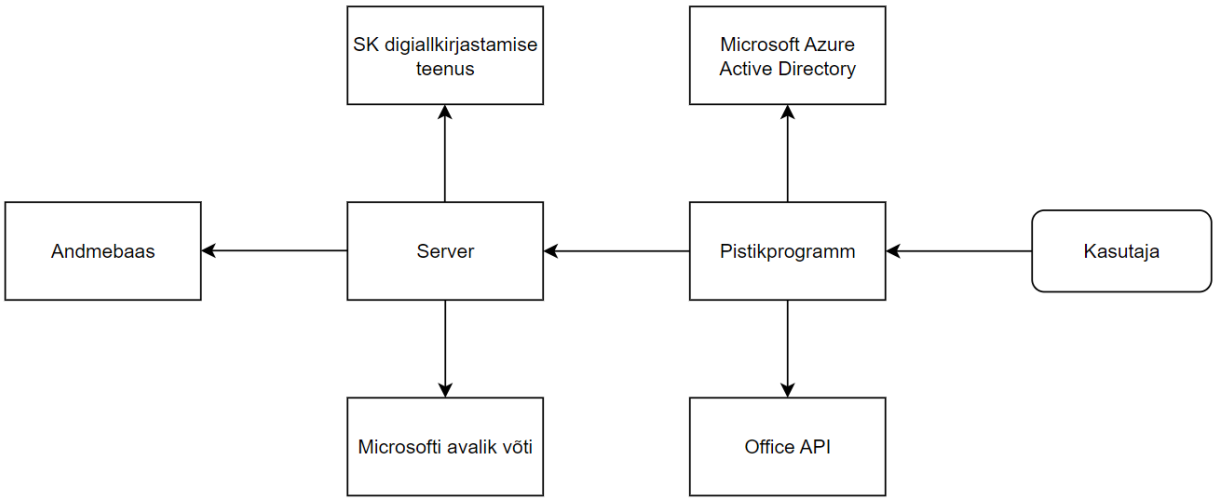
Funktsionaalsusega suhtlemiseks kasutatakse DAO (Data Access Object) mustrit, mille puhul funktsionaalsusega suhtlemine käib läbi staatilist klassi või klassi eksemplari (objekti). Klasside/eksemplaride number sõltub iga funktsionaalsuse puhul funktsionaalsuse keerukusest. Mida keerulisem on funktsionaalsus, seda rohkem klasse kasutatakse funktsionaalsuse realiseerimiseks. Sellest tulenevalt, faili süsteemi kogu loogika on eeldatavalt realiseeritud kasutades vaid ühte klassi, kuna funktsionaalsus ei ole keeruline ning ei nõua palju erinevaid meetodeid selle realisatsiooniks. Teisest küljest, moodustatakse andmebaasiga suhtlemise funktsionaalsuse realiseerimiseks eraldi klass iga andmebaasi tabeli kohta.

5.3 Pistikprogramm

Tulenevalt sellest, et pistikprogrammi raamistikuks oli valitud React, hakkab kasutajaliides koosnema komponentidest. Sarnaselt serverile, peab pistikprogramm toetama teatud funktsionaalsust, selleks, et analüüsi osas kirjeldatud rakenduse nõuded oleksid täidetud. Kogu kasutaja liidese funktsionaalsust saab loogiliselt jaotada järgmiseks funktsionaalsusteks:

- Office API'ga suhtlemine. Funktsionaalsus on vajalik eelkõige digiallkirjastavate andmete ligipääsuks.
- Serveriga suhtlemine. Pistikprogrammi ja serveri vaheline suhtlemine oleks realiseeritud kasutades REST API liidest.
- Microsoft AAD'ga suhtlemine. Teenus oleks kasutatud kasutaja autentimiseks.

Funktsionaalsusega suhtlemine toimub sarnaselt serverile, kuid staatiliste klasside või objektide asemel hakkab funktsionaalsus paiknema funktsioonides. Kogu funktsionaalsusega seotud loogika oleks jaotatud funktsioonidesse ning kõik need funktsioonid, mis on seotud ühe funktsionaalsusega, hakkavad paiknema eraldi failis. Selline funktsionaalsuse eraldamine aitaks loogiliselt eraldada erinevaid rakenduse komponente ning võimaldaks programeerijal fokuseerida ühel funktsionaalsuse implementeerimisel korraga. Töö koosseisus ei rakendata keerulisema funktsionaalsuse eraldamise mudelit, kuna selleks ei nähta põhjust. Eeldatava rakenduse maht ei ole suur ning valitud lihtne funktsionaalsuse eraldamise viis sobib antud töö rakenduse mahtudele. Kogu rakenduse andmevood on skemaatiliselt välja toodud joonisel 4.



Joonis 4. Rakenduse andmevoogud

5.4 Rakenduse funktsionaalsus

Kogu rakendus koosneb serverist ja pistikprogrammist, mille funktsionaalsused on eelnevalt kirjeldatud. Sarnaselt serverile või pistikprogrammile saab ka kogu rakenduse funktsionaalsust loogiliselt eraldada. Antud töö eeldatava lahenduse

funktsionaalsuseks on kasutajate autentimine, failide digiallkirjastamine ning jälitatavate kirjade moodustamine.

5.4.1 Autentimine

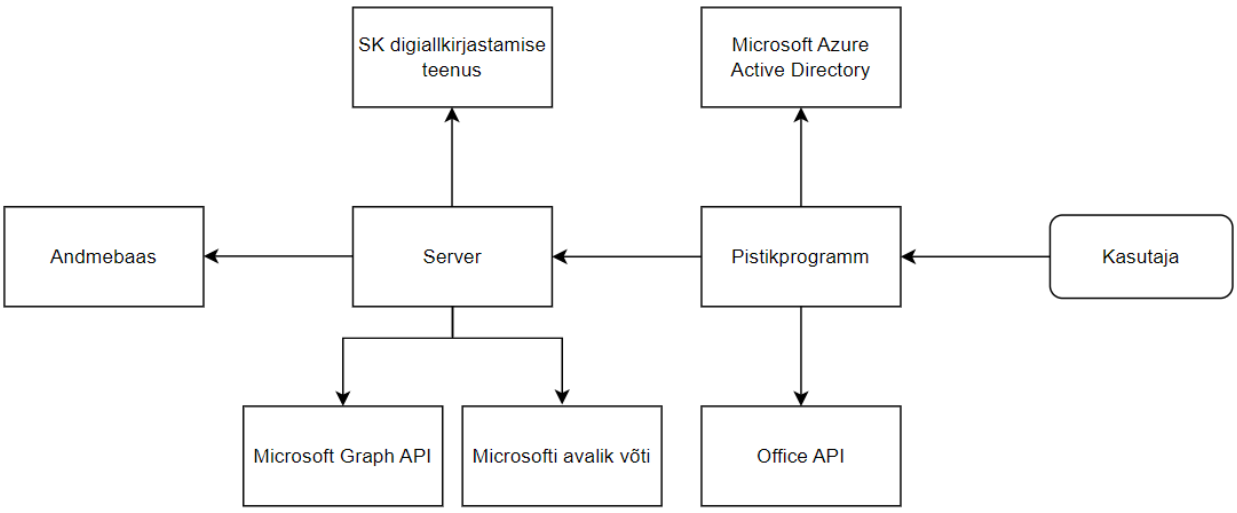
Protsess, mis oleks kõikide rakenduses olevate protsesside alam protsessiks. Iga päring pistikprogrammist serverisse sisaldab autentimise tokeneid, mida pistikprogramm hakkab saama Microsoft AAD'st. Server hakkab kasutama pistikprogrammist saadetud tokeneid päringu saatja autentimiseks. Kasutajate eristamine hakkab serveris toimuma kasutades e-maili aadressi, mida sisaldavad töös valitud autentimise teenuse tokenid.

5.4.2 Digiallkirjastamine

Põhiline rakenduse funktsionaalsus, mis võimaldab failide digiallkirjastamist. Enamus digiallkirjastamisega seotud protsesse hakkavad olema implementeeritud rakenduse serveris. Erandiks, on digiallkirjastamine kasutades ID-kaarti, mille puhul krüpteerimine hakkab toimuma kasutaja seadmel. Seetõttu, et digiallkirjastamise protsess peab olema käivitatav erinevatest Office rakendustest ja toetama erinevaid digiallkirjastamise võimalusi, peab protsess olema piisavalt paindlik. Vaatamata sellele, mis keskkonnast digiallkirjastamist alustatakse, peab digiallkirjastamise päring serverisse sisaldama järgmist infot:

- faile, mida hakatakse digiallkirjastama. Päring võib sisaldada kas faile või kogu vajaliku informatsiooni nende ligipääsuks;
- digiallkirjastatavat viisi, mida tahetakse kasutada digiallkirjastamiseks. Antud töö raames toetatud digiallkirjastamise viisideks on Smart-ID, Mobiil-Id ja ID-kaart;
- digiallkirjastamiseks vajalikke isiklikke andmeid. Sõltuvalt valitud digiallkirjastamise viisist, peab päring sisaldama erinevat isiku informatsiooni. Smart-ID puhul peab päring sisaldama digiallkirja andja isikukoodi. Mobiil-ID puhul peab isikukoodile lisanduma ka digiallkirja andja telefoni number. Id-kaardi digiallkirjastamisel peab digiallkirjastamise päring sisaldama digiallkirja andja sertifikaati;
- autentimiseks vajalikut infot. Microsoft AAD token, mida server hakkab kasutama kasutaja autentimiseks.

Erinevate Office keskkondade ainus erinevus digiallkirjastamise protsessis sõltub sellest, kust saab server digiallkirjastamiseks faile. Kõige lihtsam digiallkirjastamise protsess hakkab toimuma Word, Excel või PowerPoint keskkonnas, kus digiallkirjastamise päring hakkab sisaldama lisaks digiallkirjastatavat faili. Office API võimaldab saada ligipääsu avatud failile nii esialgses Office vormingus (.doc, .xlsx, .pptx), kui ka PDF vormingus. Sellest tulenevalt, sisaldab päring serverisse faili õiget vormingut ning serveri ülesandeks on vaid selle digiallkirjastamine. Outlook keskkond ei võimalda Office API kirjaga seotud failidele ligipääsu saada, seega nende saamine on võimalik kas Vahetus serveri (Ing. Exchange server) või Graph API teenuse kasutamisel. [61, 62] Lähtudes sellest, et kasutaja autentimiseks on valitud Microsoft autentimisteenused, mis mõlemad võimaldavad teha päringu Microsoft Graph API teenusesse, hakatakse kirjaga seotud failidele ligipääsemiseks kasutama Graph API teenust. Seega, Outlook keskkonnast digiallkirjastamise päring ei sisalda faile, vaid sisaldab kõik vajalikku informatsiooni selleks, et faile oleks võimalik saada Microsoft Graph API teenusest. Täpsemalt, hakkab Outlook digiallkirjastamise päring sisaldama kirja unikaalset identifikaatorit ja nimekirja kõikidest failide nimedest, mida tahetakse digiallkirjastada. Seoses sellega, et server peab suhtlema Microsoft Graph API teenusega, on täiendatud rakenduse andmevood skemaatiliselt välja toodud joonisel 5.



Joonis 5. Täiendatud rakenduse andmevood

Lisaks kirjeldatud digiallkirjastatavate failide ligipääsuks, oleks ka kolmas viis, kus digiallkirjastatavad failid oleksid serveris olemas ning päring sisaldaks ainult nende viidet. Olukord, kus failid oleksid juba serveris eelnevalt olemas, oleks võimalik vaid juhul, kui digiallkirjastatakse jälitatavat kirja. Digiallkirjastamiseks vajalike failide ligipääsemisele järgneb nende digiallkirjastamine, mis hakkab erinema vastavalt valitud

digiallkirjastamise viisile. Smart-ID ja Mobiil-ID digiallkirjastamise viisid on enda implementatsioonis sarnased ning nende puhul toimub nii sertifikaadi kättesaamine, krüpteerimine kui ka ajatembeldamine kasutades SK teenust. Pärast failide saamist peab server pääsema ligi digiallkirja andja sertifikaadile, millele järgneb räsi arvutamine ning selle krüpteerimine, mis on teostatud SK teenuse pakkujat kasutades. Smart-ID ja Mobiil-ID digiallkirjastamise protsessis peab kasutaja sisestama enda seadmes PIN koodi ning antud protsessis esineb kinnituskood. Kinnituskood on arvutatav digiallkirjastatavast räsist ning selle edastamiseks kasutajale, peab server pistikprogrammile tagastama vastust koos arvutatud kinnituskoodiga. [15, 16] Esimesele digiallkirjastamise päringu vastuse saamisele järgneb teine päring serverisse, selleks, et lõpetada digiallkirjastamist. Smart-ID´ga ja Mobiil-ID`ga digiallkirjastamisel kahe päringu kasutamine on vajalik vaid selleks, et edastada kasutajale kinnituskood. Digiallkirjastamise lõpus tagastab server linki, mille kasutamisel võib saada digiallkirjastatud konteinerile ligipääsu. ID-kaardiga digiallkirjastamine erineb Smart-ID ja Mobiil-ID digiallkirjastamise protsessidest, kuid sarnaselt nendele, hakkab ID-kaardi digiallkirjastamine koosnema kahest päringust. Esimene päring sisaldab digiallkirja andja sertifikaati koos teiste digiallkirjastamiseks vajalike andmetega. Lähtudes sellest, et ID-kaardi digiallkirjastamine peab toimuma kasutaja seadmes, peab server arvutama räsi digiallkirjastatavatest failidest ning saatma selle krüpteerimiseks tagasi. Seejärel, kui kasutaja krüpteerib serverist saadud räsi, saadetakse valmis digiallkiri tagasi serverisse digiallkirjastamise protsessi lõpetamiseks.

5.4.3 Jälitatav kiri

Rakenduse funktsionaalsus, mille puhul saab luua serveri poolt jälitatavat kirja. Digiallkirja andjateks on kirja saajad ning server jälgib, et ainult kirja koostamisel määratud isikud saaksid anda oma allkirja. Kõikide allkirjade andmise järgselt saadab server digiallkirjastatud faile kõikidele, selleks kirja koostamisel määratud isikutele. Funktionaalsuse nõuded on kirjeldatud analüüsi osas nõuete peatükis.

Jälitatava kirja koostamiseks oleks serveris eraldi funktsioon, mille käivitamiseks oleks vajalik eraldi lingi kasutamine. Jälitatava kirja loomiseks oleks vajalik serverisse järgmiste andmete saatmine:

- digiallkirjastamiseks vajalike faile,
- isikuid, kes peavad digiallkirjastama,

- e-maile, kuhu tuleb saata digiallkirjastatud konteineri seejärel, kui kõik selleks kohustatud isikud on andnud digiallkirja,

- autentimiseks vajalikut infot.

Seejärel kasutajat autentitakse ning kõik jälitatud kirjaga seotud andmed salvestatakse andmebaasi ja digiallkirjastatavad failid salvestatakse faili süsteemis. Jälitatava kirja digiallkirjastamine toimub sarnaselt tavalisele digiallkirjastamise protsessile, kus digiallkirjstatavad failid on eelnevalt serveris olemas. Erinevalt tavalisest digiallkirjastamisest server hakkab valideerima digiallkirja andjaid ning digiallkirjastada saavad vaid selleks ette määratud isikud. Lisaks, jälgib server ka digiallkirjade arvu ning seejärel, kui kõik digiallkirjad on antud, saadab server kirja koos digiallkirjastatud konteineriga kõikidele selleks ette määratud isikutele.

Jälitatava kirja digiallkirjastamiseks, nagu eelnevalt välja toodud, peab saatma failide viite serverisse, mida tahetakse digiallkirjastada. Viide on unikaalne identifikaator, mida kasutades otsib server nii digiallkirjastatavat konteinerit kui ka sellega seotuid andmeid andmebaasis. Digiallkirja andjad saavad unikaalset identifikaatorit saadetud kirjadest. Kogu jälitatava kirja digiallkirjastamine toimub nii, et kirja koostamisel saadetakse serverisse kirjaga seotud valitud faile koos teise informatsiooniga, mis on vajalik jälitatud kirja koostamiseks. Koostatava kirjaga seotud unikaalne identifikaator saadetakse päringu vastuseks. Unikaalse identifikaatori ligipääsu saamisele järgneb selle asetamine koostatavasse kirja selleks, et kirja saajad saaksid sellele ligipääsu ning selle abil saaksid kirjaga seotuid faile digiallkirjastada. Unikaalne identifikaator on peidetud kirja sees selleks, et see ei oleks nähtav ning saadud kiri oleks sarnane tavalise kirjaga. Unikaalse identifikaatori peitmine oleks teostatud kasutades HTML'i ja CSS'i. Unikaalse identifikaatorile ligipääsuks oleks kasutatud vaadeldava töö raames valmiv pistikprogramm. Pistikprogrammi käivitamisel hakkab see otsima avatud kirja sees unikaalset identifikaatorit, selleks, et määrata, kas tegu on jälitatava kirjaga või mitte. Pistikprogramm teeb kindlaks kirja tüüpi, kuna sellest sõltub kuvatud vaade. Juhul, kui pistikprogramm leiab kirjas unikaalset identifikaatorit hakkab pistikprogramm tegema päringu serverisse, et saada andmeid avatud kirjast. Päringu vastuseks saab pistikprogramm järgmisi andmeid:

- Kasutaja digiallkirja andmise vajadus ehk, kas kasutajal on vaja anda digiallkirja või mitte.

- Digiallkirjastatavate failide nimed. Digiallkirjastatavat faili määratakse jälitatava kirja koostamisel ning need moodustavad alamhulga kõigi kirjaga seotud failidest.

- Kasutajaga seotud andmed kirja juures. Antud andmeks on töö raames väli, mis näitab kas kasutajale saadetakse kiri digiallkirjastatud konteineriga digiallkirjastamise järgselt või mitte.

- Kirja digiallkirjastamise staatus. Kas kiri on saanud vajalike isikute digiallkirju või on selle protsessis.

Lähtudes saadetud serveri andmetest kuvatakse kasutajale erinevat informatsiooni. Juhul, kui kirja digiallkirjastamine on lõppenud või kui kasutaja digiallkiri ei ole nõutav, oleks kasutajale kuvatud vaade, kus avatud kirjaga seotuid faile on võimalik digiallkirjastada tavalise digiallkirjastamisega. Samas, kui kasutaja digiallkiri on nõutud avatud kirja digiallkirjastamiseks, oleks kasutajale kuvatud jälitatava kirja digiallkirjastamise vaade.

6 Realisatsioon

Antud töö osas kirjeldatakse valmis rakendust ning analüüsitakse teostatud tööd. Lisaks, kuulub vaadeldava osa koosseisu probleemide kirjeldus, mis tekkisid programmi realiseerimisel ning, mis määral probleemid mõjutasid valmis rakenduse funktsionaalsust võrreldes eeldatava funktsionaalsusega.

6.1 Server

Rakenduse server kasutab serverivaba arhitektuuri ning koosneb funktsioonidest, kus iga funktsioon omab ajendit, vastavalt millele ta käivitub. Kogu serveri koosseisus on kuus funktsiooni, mis omavad erinevaid ajendeid. Viis serveri funktsiooni omavad http ajendit ning nende käivitamiseks on vajalik http päringu tegemine nendega seotud aadressile. Funktsioone http ajendiga moodustavad järgmised funktsioonid:

- digiallkirjastamise alustamise funktsioon,
- digiallkirjastamise lõpetamise funktsioon,
- digiallkirjastatud konteineri ligipääsu funktsioon,
- jälitatava kirja koostamise funktsioon,
- jälitatava kirja informatsiooni saamise funktsioon.

Kuuendaks serveri funktsiooniks, on funktsioon, mille ajendiks on *Queue Storage* sõnumite ilmumine. Serverivaba arhitektuuri kasutamise tõttu, ei ole võimalik loenduri asetamine koodi, kuna server peatatakse kohe pärast vastuse saatmist. Selle tõttu, kasutatakse Azure *Queue Storage* teenust konteineri kustutamiseks kahe minuti möödudes. Azure *Queue Storage* on teenus, mis võimaldab hoida sõnumeid, mille suurus ei ületa 64KB. [63]

6.2 Pistikprogramm

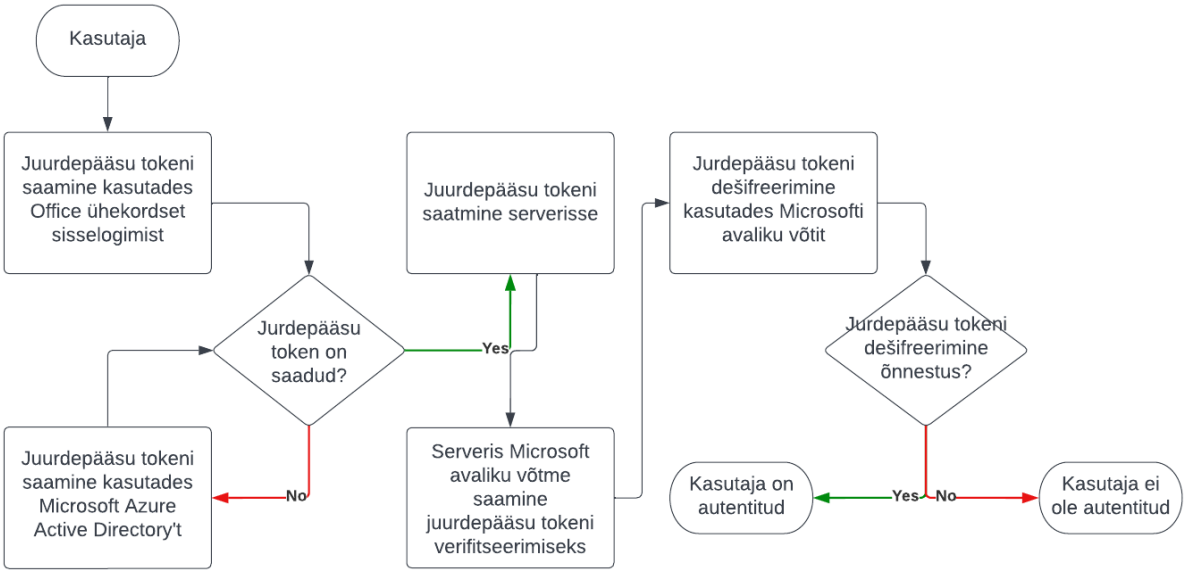
Tulenevalt Office piirangutest, koosneb antud töö rakenduse kasutajaliides kahest eraldiseisvast pistikprogrammist. Üks pistikprogramm töötab Outlook keskkonnas ja teine pistikprogramm Word, Excel ja PowerPoint keskkondades. Vaatamata erinevatele pistikprogrammidele, laiendavad mõlemad pistikprogrammid Office rakendusi kasutades selleks tegumiriba. Lisaks, tulenevalt sellest, et kasutajaliides on loodud kasutades React raamistikku, koosnevad mõlemad pistikprogrammid analoogsetest komponentidest.

6.3 Rakenduse funktsionaalsus

Kõik eeldatav põhifunktsionaalsus on realiseeritud, kuid rakenduse funktsionaalsus erineb eeldatavast funktsionaalsusest eelkõige Office keskkonna piirangute tõttu. Valmis rakendus toetab digiallkirjastamist kõikides eeldatavates keskkondades ning toetab jälitatava kirja koostamist.

6.3.1 Autentimine

Autentimine on realiseeritud kasutades Microsofti autentimise teenust. Autentimise tokenite saamine on võimalik kahel viisil. Esimese viisi kohaselt saab tokeneid läbi Office ühekordse sisselogimise. Juhul, kui esimesele viisile ei ole ligipääsu või ei ole võimalik selle kasutamine teise põhjuse tõttu, kasutatakse teist autentimise meetodit. Teine autentimise meetod autendib kasutajat läbi autentimise akna, kus kasutajal on vajalik enda andmete sisestamine. Mõlemad autentimise meetodid tagastavad autentimistokeneid, mida rakenduse server toetab ning suudab verifitseerida. Autentimise diagramm on välja toodud joonisel 6.



Joonis 6. Autentimise diagramm

6.3.2 Digiallkirjastamine

Digiallkirjastamine on põhiline funktsionaalsus, mida rakendus toetab. Selle funktsionaalsuse kasutamisel on võimalik allkirjastada dokumente, kasutades selleks kõrgemat elektrooniliste allkirja taset, milleks on kvalifitseeritud elektroonilised allkirjad. Funktsionaalsus toetab järgmisi Office platvorme:

- Outlook (nii veebipõhine kui ka eraldiseisev rakendus),
- Word (nii veebipõhine kui ka eraldiseisev rakendus),
- Excel (nii veebipõhine kui ka eraldiseisev rakendus),
- PowerPoint (nii veebipõhine kui ka eraldiseisev rakendus).

Veebipõhistes rakendustes on implementeeritud kolm võimalikku digiallkirjastamise võimalust, milleks on Smart-ID, Mobiil-ID ja ID-kaart. Eraldiseisvates rakendustes ei ole piirangute tõttu toetatud ID-kaardiga digiallkirjastamine. ID-kaardi digiallkirjastamiseks kasutab pistikprogramm brauseri laiendit, millele ei ole ligipääsu eraldiseisvatel Office rakendustel.

Kavandi järgselt koosneb digiallkirjastamine kahest päringust. Esimene päring koosneb digiallkirjastamiseks vajalikest andmetest ja failidest. Päring sisaldab faile või kogu vajaliku informatsiooni selleks, et saada serveris failidele ligipääsu. Mõlemad päringud on vajalikud, kuna kasutades Smart-ID ja Mobiil-ID digiallkirjastamiseks on vajalik digiallkirja andjale kinnituskoodi edastamine serverist. Samas, ID-kaardiga digiallkirjstamisel on vajalik digiallkirja andjale räsi edastamine serverist. Serveri raames kasutatakse digiallkirjastamiseks eelnevalt kirjeldatud SK teenust koos digiallkirjastamise teegiga.

Word, Excel ja PowerPoint keskkonnad töötavad ühe failiga, mistõttu kasutatakse digiallkirjastamiseks avatud faili. Enne digiallkirjastamist on võimalik valida digiallkirjastatava faili vormingut. Toetatud vorminguteks on Office rakenduse esialgsed vormingud (.doc, .xlsx, .pptx) või PDF vorming. Faili konverteerimine valitud vormingusse toimub pistikprogrammis faili saamisel. Outlook keskkonnas võetakse faile kirja manuses olevatest failidest. Seetõttu, et Outlook'is Office liides ei anna kirjaga seotud failidele ligipääsu, saadetakse faile kasutades Microsoft Graph API teenust. Seega, ei saadeta faile digiallkirjastamiseks Outlook keskkonna serverisse, vaid edastatakse kirja unikaalset identifikaatorit koos failide nimedega, mida soovitakse digiallkirjastada. Kasutades kasutajalt saadud autentimistokenit ja kirja unikaalset identifikaatorit, saab server ligipääsu kirjaga seotud failidele kasutades Graph API teenust. Kavandile tuginedes, on võimalik digiallkirjastamise järgselt allalaadida digiallkirjstatud konteinerit kõikides keskkondades. Lisaks, digiallkirjastamise järgselt toetab Outlook uue kirja koostamist, kus digiallkirjastatud konteiner on vaikimisi lisatud koostatavale kirjale. Digiallkirjastamise protsess on lisatud diagrammina Lisa 2. Samuti, on Lisas 3 näidatud digiallkirjastamise protsess valmis rakenduses.

Digiallkirjastatud konteiner kustutatakse kahe minuti möödudes digiallkirjastamise protsessi käivitamisest. Tulenevalt sellest, et *Queue Storage* on Azure teenus ning rakenduse serverivaba arhitektuuri teenuse pakkuja on analoogselt Azure, võimaldab see käivitada Azure serverivaba arhitektuuri vastavalt sõnumite ilmumisele. [64] Konteineri loomisel pannakse sõnum järjekorda, mis varjatakse kaheks minutiks. Kahe minuti möödudes ilmub sõnum, mis oli eelnevalt varjatud ning käivitab rakendust, mis kustutab aegunud konteineri serverist. Kustutava konteineri tuvastamine on võimalik, kuna peidetud sõnum sisaldab konteineri unikaalset identifikaatorit.

6.3.3 Jälitatud kiri

Jälitatud kirja funktsionaalsus on implementeeritud vastavalt kavandile. Kirja koostamisel laetakse serverisse kirjaga seotud faile, mida soovitakse allkirjastada ning saadetakse kirja unikaalsed identifikaatorid. Saadud kirja unikaalne identifikaator peidetakse koostatavasse kirja selleks, et kiri oleks sarnane tavalisele kirjaga ning, et kirja saajad saaksid seda digiallkirjastada. Unikaalse identifikaatori peitmiseks kasutatakse HTML ja CSS tehnoloogiaid. Täpsemalt, lisatakse kirja unikaalne identifikaator html tüübina, millele rakendatakse järgmisi CSS reegleid:

- `display: none, max-height: 0px ja overflow: hidden.`

CSS reeglid teevad unikaalse identifikaatori nähtamatuks kirja saajale, kuid identifikaatorit on võimalik tuvastada Office liidese abil. Identifikaatori leidmiseks teeb pistikprogramm päringu Office liidesesse, et saada ligipääsu kirja sisemusele ning selle sees on võimalik identifikaatori tuvastamine.

Rakenduse koosseisus on kasutatud unikaalne konstant selleks, et valminud pistikprogramm oleks kindel, et kirja sees olev peidetud unikaalne identifikaator on tehtud sama pistikprogrammi kasutades. Unikaalse konstandi kasutamine väldib liigsete päringute saatmist serverisse, kuna pistikprogramm veendub, et tegu on just rakenduse raames kasutatava identifikaatoriga. Seega, salvestatakse kirja sees lisaks kirja unikaalsele identifikaatorile ka unikaalset rakenduse konstanti, milleks on: #$OutlookDigitalSignatureAddIn$#. Kirjas peidetud konstant koos unikaalse kirja identifikaatori näiteks on

- #$OutlookDigitalSignatureAddIn$#5baaa4db-99e1-439f-950e-3278f114a7ef.

Jälitatud kirja digiallkirjastamise protsess on sarnane tavalise digiallkirjastamise protsessiga. Erinevuseks on see, et kasutajale ei kuvata allkirjastatavate failide valiku, vaid kuvatakse informatsiooni, mis on seotud jälitatava kirjaga. Jälitatava kirja avamisel saab allkirja andja teada järgmist informatsiooni:

- kas kirja allkirjastamine on lõppenud või on protsessis,
- kas kasutajal on vajalik kirja allkirjastamine,

- mis failid kirjast allkirjastatakse,
- kas kasutajale saadetakse kiri allkirjastamise järgselt või mitte.

Jälitatava kirja allkirjastamisel ei saadeta allkirjastatavaid faile serverisse, vaid edastatakse jälitatava kirja identifikaatorit, mille järgi leiab server varem salvestatud kirjaga seotuid faile. Kui kõik määratud isikud on jälitatud kirja allkirjastanud, järgneb allkirjastatud konteineri saatmine kõikidele kirja koostamisel määratud isikutele. Konteineri saatmine toimub kasutades selleks C# *System.Net.Mail* paketis olevat funktsionaalsust. Konteineri saatmise järgselt kustutatakse see serverist koheselt, nii turvariski maandamiseks kui ka mälu vabastamiseks. Valmis rakenduse jälitatud kirja loomine ning digiallkirjastamine on välja toodud Lisa 4.

7 Kokkuvõte

Antud töö lahendatavaks probleemiks on digiallkirjastamise protsessi raskus, mille lahendamiseks on loodud rakendus, mis toetab failide digiallkirjastamist Microsoft Office keskkondades. Kõik rakenduse loomisel kasutatud tehnoloogiad, olid valitud lähtudes nii ettevõte nõuetest, kui ka ülesande püstitusest. Lisaks, oli rakenduse loomisele eelnevalt koostatud kavand, mis kirjeldas detailselt eeldatava rakenduse struktuuri, funktsionaalsust ja andmevooge. Valmis rakendus toetab nelja Microsoft Office rakendust, milleks on Word, Excel, PowerPoint ja Outlook. Lisaks, saab rakendus töötada nii veebipõhises keskkonnas, kui ka eraldiseisvate rakenduste osana. Rakenduse raames toetatud digiallkirjastamise meetoditeks on Smart-ID, Mobiil-ID ja ID-kaart. Rakenduse loomise etapil ilmnesid autorile ettenähtamatud piirangud, mistõttu eraldiseisvates rakenduses ei ole toetatud ID-kaardi digiallkirjastamise meetod. Vaatamata ettenähtamatutele piirangutele, toetab lõplik rakendus kogu eeldatavat funktsionaalsust ning kujutab endast üht lahendust antud bakalaureusetöö püstitatud probleemile.

Valminud rakendust kasutatakse ettevõttes, kus töö autor töötab ning rakenduse kasutamine võimaldab tõsta ettevõtte töötajate tõhusust. Rakendus säästab digiallkirjastamisele kuluvat aega ning muudab kogu protsessi ladusamaks, kuna digiallkirjastamiseks ei ole vajalik eraldi rakenduse kasutamine. Lisaks, on töö raames valminud rakenduse kasutamine kiire ja hõlbus viis lepingute allkirjastamiseks, tänu rakenduse jälitatava kirja funktsionaalsusele.

Kuigi töö raames valminud rakendus on võimas tööriist dokumentide digiallkirjastamiseks, on veel võimalusi, kuidas saab rakendust edasi arendada. Näiteks, digiallkirjastamine nutiseadmetel, mis oli antud töö raames väljaspool skoopi. Lisaks, toetab valmis rakenduse versioon digiallkirjastamist vaid Eesti piirides ning üheks rakenduse arendamisvõimaluseks võiks olla rohkemate riikide toetus.

Kasutatud kirjandus

[1] Microsoft, *Office Add-ins platform overview*, [Online]. Loetud aadressil: https://learn.microsoft.com/en-us/office/dev/add-ins/overview/office-add-ins Kasutatud: 06.10.22.

[2] W. Vercruysse, *What is an electronic signature*, 2020, [Online] Loetud aadressil: https://ec.europa.eu/digital-building-blocks/wikis/display/ESIGKB/What+is+an+electronic+signature Kasutatud: 19.10.22.

[3] W. Vercruysse, *What are the levels, simple, advanced and qualified of electronic signatures*, 2020, [Online] Loetud aadressil: https://ec.europa.eu/digital-building-blocks/wikis/display/ESIGKB/What+are+the+levels%2C+simple%2C+advanced+and+qualified+of+electronic+signatures Kasutatud: 19.10.22.

[4] S. Padhye, R. A. Sahu, V. Saraswat, *Introduction to Cryptography.* U.S: CRC Press, 2018. [E-book]. Loetud aadressil: https://www.perlego.com/book/1382562/introduction-to-cryptography-pdf Kasutatud: 08.10.22

[5] W. Stallings, "Cryptologia," *Taylor & Franci*, 37:4, 311-327, 2013. DOI: 10.1080/01611194.2013.797044

[6] A. M. Borzyszkowski, "Electronic signature, the theory and the practice, Poland and Europe" *Institute of Computer Science*, vol 18, 81-825, [Online]. Loetud aadressil: https://citeseerx.ist.psu.edu/pdf/ae4e6a520b379c13408b79e688642ff51d30167c Kasutatud: 09.10.22

[7] Smart-Id, *Smart-ID Technical Overview*, 2017, [Online]. Loetud aadressil: https://www.smart-id.com/wordpress/wp-content/uploads/2017/01/smart-id-technical-overview-v0.6.html Kasutatud: 08.10.22.

[8] S. Kravtšenko, "The Estonian Mobile-ID Implementation on the SIM Card*",* [Bakalaureuse töö], Arvutiteaduse Instituut, teaduse ja tehnoloogia teaduskond, Tartu Ülikool, Eesti, 2022. [Online]. Loetud aadressil: https://comserv.cs.ut.ee/home/files/Kravtsenko_Informaatika_2022.pdf?study=ATILoputoo&reference=B8743FDA5ACEA490D28322527F48186A10F363A0 Kasutatud: 09.10.22.

[9] Arvutikaitse, *ID-kaart*, [Online]. Loetud aadressil: https://www.arvutikaitse.ee/arvutikaitse-algtoed/id-kaart/ Kasutatud: 08.10.22.

[10] A. Parsovs, "Estonian Electronic Identity Card and its Security Challenges*",* [Doktoridissertatsioon], Arvutiteaduse Instituut, teaduse ja tehnoloogia teaduskond, Tartu Ülikool, Eesti, 2021. [Online]. Loetud aadressil: https://dspace.ut.ee/handle/10062/71481 Kasutatud: 08.10.22.

[11] Dokobit, *Digiallkirjastamine teie infosüsteemis*, 2022, [Online]. Loetud aadressil: https://www.dokobit.com/et/lahendused/dokumentide-api Kasutatud: 28.09.22.

[12] Dokobit, *Document signing integration with Documents Gateway*, 2022, [Online]. Loetud aadressil: https://support.dokobit.com/article/543-document-signing-integration-with-documents-gateway Kasutatud: 28.09.22.

[13] Dokobit, *Vali endale sobiv pakett*, 2022, [Online]. Loetud aadressil: https://www.dokobit.com/et/maksumus Kasutatud: 28.09.22.

[14] Sk ID Solutions, *Services*, 2022, [Online]. Loetud aadressil: https://www.skidsolutions.eu/teenused/ Kasutatud: 28.09.22.

[15] Sk ID Solutions, *Mobile ID (MID) REST API*, 2022, [Online]. Loetud aadressil: https://github.com/SK-EID/MID/blob/master/README.md Kasutatud: 28.09.22.

[16] Sk ID Solutions, *smart-id-documentation*, 2022, [Online]. Loetud aadressil: https://github.com/SK-EID/smart-id-documentation/blob/master/README.md Kasutatud: 28.09.22.

[17] Sk ID Solutions, *Mobiil-ID teenuse hinnakiri*, 2022, [Online]. Loetud aadressil: https://www.skidsolutions.eu/teenused/hinnakiri/mobiil-id-teenus/ Kasutatud: 28.09.22.

[18] Sk ID Solutions, *Smart-ID teenuse hinnakiri*, 2022, [Online]. Loetud aadressil: https://www.skidsolutions.eu/teenused/hinnakiri/smart-id/ Kasutatud: 28.09.22.

[19] Sk ID Solutions, *Ajatempliteenuse hinnakiri*, 2022, [Online]. Loetud aadressil: https://www.skidsolutions.eu/teenused/hinnakiri/ajatempliteenus/ Kasutatud: 28.09.22.

[20] Id, *DigiDoc teegid: ülevaade*, 2022, [Online]. Loetud aadressil: https://www.id.ee/artikkel/digidoc-teegid-ulevaade-2/ Kasutatud: 28.09.22.

[21] github, *libdigidocpp*, 2022, [Online]. Loetud aadressil: https://github.com/open-eid/libdigidocpp/blob/master/README.md Kasutatud: 28.09.22.

[22] Libdigidocpp, *Libdigidocpp Programmer's Guide*, 2022, [Online]. Loetud aadressil: https://open-eid.github.io/libdigidocpp/manual.html Kasutatud: 28.09.22.

[23] github, *DigiDoc4j 5.0.0 API*, 2022, [Online]. Loetud aadressil: http://open-eid.github.io/digidoc4j/ Kasutatud: 28.09.22.

[24] J. Ingeno, *Software Architect's Handbook*, UK: Packt Publishing, 2018 [E-book]. Loetud aadressil: https://www.perlego.com/book/799730/software-architects-handbook-become-a-successful-software-architect-by-implementing-effective-architecture-concepts-pdf Kasutatud: 08.10.22.

[25] Microsoft, *Develop Office Add-ins*, [Online]. Loetud aadressil: https://learn.microsoft.com/en-us/office/dev/add-ins/develop/develop-overview Kasutatud: 07.10.22.

[26] Microsoft, *Office.Document interface*, [Online]. Loetud aadressil: https://learn.microsoft.com/en-us/javascript/api/office/office.document?view=common-js-preview Kasutatud: 07.10.22.

[27] Microsoft, *Office.Mailbox interface*, [Online]. Loetud aadressil: https://learn.microsoft.com/en-us/javascript/api/outlook/office.mailbox?view=outlook-js-preview Kasutatud: 07.10.22.

[28] Microsoft, *Overview of authentication and authorization in Office Add-ins*, [Online]. Loetud aadressil: https://learn.microsoft.com/en-us/office/dev/add-ins/develop/overview-authn-authz Kasutatud: 07.10.22.

[29] Microsoft, *Deploy and publish Office Add-ins*, [Online]. Loetud aadressil: https://learn.microsoft.com/en-us/office/dev/add-ins/publish/publish Kasutatud: 07.10.22.

[30] Microsoft, *Why publish your app with Microsoft's app stores*, [Online]. Loetud aadressil: https://learn.microsoft.com/en-us/azure/marketplace/why-publish Kasutatud: 07.10.22.

[31] Microsoft, *Make your solutions available in Microsoft AppSource and within Office*, [Online]. Loetud aadressil: https://learn.microsoft.com/en-us/azure/marketplace/submit-to-appsource-via-partner-center Kasutatud: 07.10.22.

[32] RedHat, *What are cloud service providers?*, 2022, [Online] Loetud aadressil: https://www.redhat.com/en/topics/cloud-computing/what-are-cloud-providers Kasutatud: 07.11.22.

[33] S. J. Bigelow, *Microsoft Azure*, 2022, [Online]. Loetud aadressil: https://www.techtarget.com/searchcloudcomputing/definition/Windows-Azure Kasutatud: 03.10.22.

[34] AWS, *What is AWS*, 2022, [Online]. Loetud aadressil https://aws.amazon.com/what-is-aws/ Kasutatud: 03.10.22.

[35] AWS, *AWS Pricing*, [Online]. Loetud aadressil: https://aws.amazon.com/pricing/?aws-products-pricing.sort-by=item.additionalFields.productNameLowercase&aws-products-pricing.sort-order=asc&awsf.Free%20Tier%20Type=*all&awsf.tech-category=*all Kasutatud: 06.10.22.

[36] R. Ciphertex, *What Is Data Storage?*, 2021, [Online] Loetud aadressil: https://ciphertex.com/what-is-data-storage/ Kasutatud: 20.10.22.

[37] Oracle, *What Is a Database?*, [Online]. Loetud addressil: https://www.oracle.com/database/what-is-database/#:~:text=A%20database%20is%20an%20organized,database%20management%20system%20(DBMS). Kasutatud: 30.09.22.

[38] R. Lavarian, *What Is a File System? Types of Computer File Systems and How they Work – Explained with Examples*, 2022, [Online]. Loetud aadressil: https://www.freecodecamp.org/news/file-systems-architecture-explained/#:~:text=A%20file%20system%20is%20a,system%20to%20manage%20the%20files. Kasutatud: 28.09.22.

[39] Github, *Understanding the Azure App Service file system*, 2021, [Online] Loetud aadressil: https://github.com/projectkudu/kudu/wiki/Understanding-the-Azure-App-Service-file-system Kasutatud: 12.10.22.

[40] Cloudera, *Creating your first Azure Function App*, 2022, [Online] Loetud aadressil: https://docs.cloudera.com/dataflow/cloud/azure-functions/topics/cdf-azure-function-disk-storage-considerations.html Kasutatud: 12.10.22.

[41] Cloudera, *Disk storage considerations*, 2022, [Online] Loetud aadressil: https://docs.cloudera.com/dataflow/cloud/azure-functions/topics/cdf-azure-function-disk-storage-considerations.html Kasutatud: 12.10.22.

[42] AWS, *Amazon EC2 instance store*, 2022, [Online] Loetud aadressil: https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/InstanceStorage.html Kasutatud: 12.10.22.

[43] CloudCasts, *Serverless Limitations with Lambda*, 2021, [Online] Loetud aadressil: https://cloudcasts.io/article/lambda-limitations Kasutatud: 12.10.22.

[44] Google, *What is Object storage?*, 2022, [Online]. Loetud aadressil: https://cloud.google.com/learn/what-is-object-storage Kasutatud: 28.09.22.

[45] G2, *Best Object Storage Software*, 2022, [Online]. Loetud aadressil: https://www.g2.com/categories/object-storage Kasutatud: 28.09.22.

[46] AWS, *Caching Overview*, 2022, [Online]. Loetud aadressil: https://aws.amazon.com/caching/#:~:text=In%20computing%2C%20a%20cache%20is,the%20data's%20primary%20storage%20location. Kasutatud: 03.10.22.

[47] S. Bentotage, *Authentication as a Service for Enterprise Applications*, [Online]. Loetud aadressil: https://medium.com/@sanjayaben/authentication-as-a-service-for-enterprise-applications-828540f74c9b Kasutatud: 27.10.22.

[48] Microsoft, *What is the Microsoft identity platform?*, [Online]. Loetud aadressil: https://learn.microsoft.com/en-us/azure/active-directory/develop/v2-overview Kasutatud: 27.10.22.

[49] Microsoft, *Microsoft identity platform access tokens*, [Online]. Loetud aadressil: https://learn.microsoft.com/en-us/azure/active-directory/develop/access-tokens Kasutatud: 27.10.22.

[50] AWS, *What is Amazon Cognito?*, [Online]. Loetud aadressil: https://docs.aws.amazon.com/cognito/latest/developerguide/what-is-amazon-cognito.html Kasutatud: 27.10.22.

[51] Zoho, *Zoho Sign integration for Outlook*, [Online]. Loetud addressil: https://www.zoho.com/sign/help/microsoft-outlook-add-in-integration.html Kasutatud: 02.10.22.

[52] Zoho, *Sign Yourself*, [Online]. Loetud addressil: https://www.zoho.com/sign/help/send-digital-signature-documents/sign-yourself.html Kasutatud: 02.10.22.

[53] Zoho, *Send for Signatures*, [Online]. Loetud addressil: https://www.zoho.com/sign/help/send-digital-signature-documents/send-for-signatures.html Kasutatud: 02.10.22.

[54] DocuSign, *DocuSign for Outlook*, [Online]. Loetud addressil: https://www.docusign.com/solutions/microsoft/outlook Kasutatud: 02.10.22.

[55] DocuSign, *Send a Document with DocuSign for Word*, 2022[Online]. Loetud addressil: https://support.docusign.com/s/document-item?language=en_US&bundleId=uvt1617134092892&topicId=wjc1617134399151.html&_LANG=enus Kasutatud: 30.10.22.

[56] DocuSign, *Sign a Document with DocuSign for Word*, 2022 [Online]. Loetud addressil: https://support.docusign.com/s/document-item?language=en_US&rsc_301&bundleId=uvt1617134092892&topicId=vsx1617134524229.html&_LANG=enus Kasutatud: 30.10.22.

[57] DocuSign, *Sign attachments*, [Online]. Loetud addressil: https://support.docusign.com/s/document-item?language=en_US&rsc_301&bundleId=ozy1574179776330&topicId=rfx1574179767033.html&_LANG=enus Kasutatud: 02.10.22.

[58] DocuSign, *Send attachments*, [Online]. Loetud addressil: https://support.docusign.com/s/document-item?language=en_US&bundleId=ozy1574179776330&topicId=nyf1574179769893.html&_LANG=enus Kasutatud: 02.10.22.

[59] B. Enos, *7 BENEFITS OF USING PDF FILES*, 2022, [Online]. Loetud addressil: https://www.enostech.com/7-benefits-of-using-pdf-files/ Kasutatud: 14.12.22.

[60] B. Joshi, *Beginning SOLID Principles and Design Patterns for ASP.NET Developers*, Thane, India: Springer Science+Business Media New York, 2016. [E-book].DOI: 10.1007/978-1-4842-1848-8_10

[61] Microsoft, *Get attachment*, 2022, [Online]. Loetud aadressil: https://learn.microsoft.com/en-us/graph/api/attachment-get?view=graph-rest-1.0&tabs=http Kasutatud: 07.11.22.

[62] Microsoft, *Use the Outlook REST APIs from an Outlook add-in*, 2022, [Online]. Loetud aadressil: https://learn.microsoft.com/en-us/office/dev/add-ins/outlook/use-rest-api Kasutatud: 07.11.22.

[63] Microsoft, *What is Azure Queue Storage?*, 2021, [Online]. Loetud aadressil: https://learn.microsoft.com/en-us/azure/storage/queues/storage-queues-introduction Kasutatud: 07.11.22.

[64] Microsoft, *Azure Queue storage trigger for Azure Functions*, 2022, [Online]. Loetud aadressil: https://learn.microsoft.com/en-us/azure/azure-functions/functions-bindings-storage-queue-trigger?tabs=in-process%2Cextensionv5&pivots=programming-language-csharp Kasutatud: 07.11.22.

Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks[1]

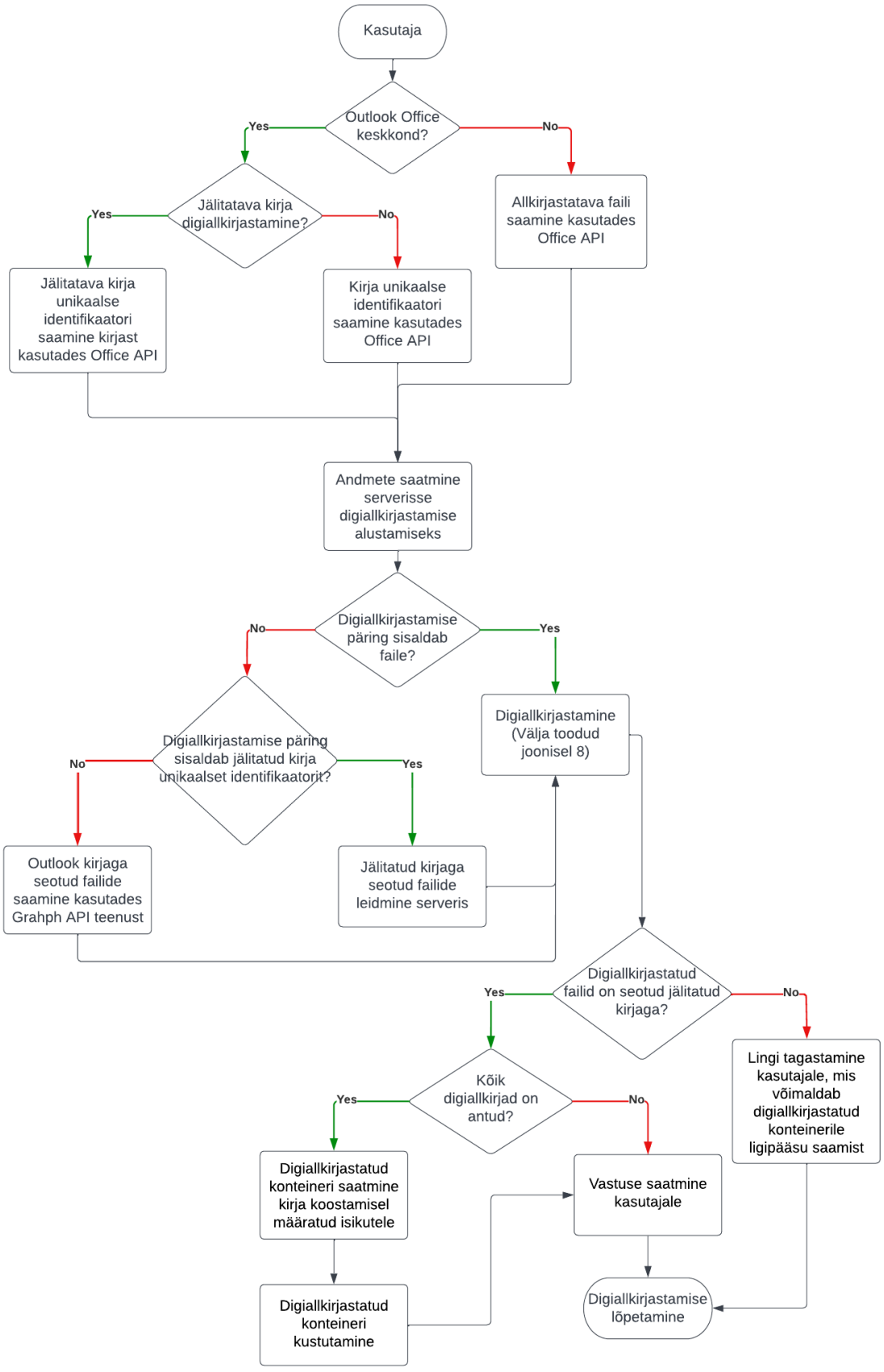
Mina, Roman Malõšev

1. Annan Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose "Microsoft Office'i pistikprogramm digiallkirjastamiseks", mille juhendaja on Priit Raspel
   1.1. reprodutseerimiseks lõputöö säilitamise ja elektroonse avaldamise eesmärgil, sh Tallinna Tehnikaülikooli raamatukogu digikogusse lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;
   1.2. üldsusele kättesaadavaks tegemiseks Tallinna Tehnikaülikooli veebikeskkonna kaudu, sealhulgas Tallinna Tehnikaülikooli raamatukogu digikogu kaudu kuni autoriõiguse kehtivuse tähtaja lõppemiseni.
2. Olen teadlik, et käesoleva lihtlitsentsi punktis 1 nimetatud õigused jäävad alles ka autorile.
3. Kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest ning muudest õigusaktidest tulenevaid õigusi.

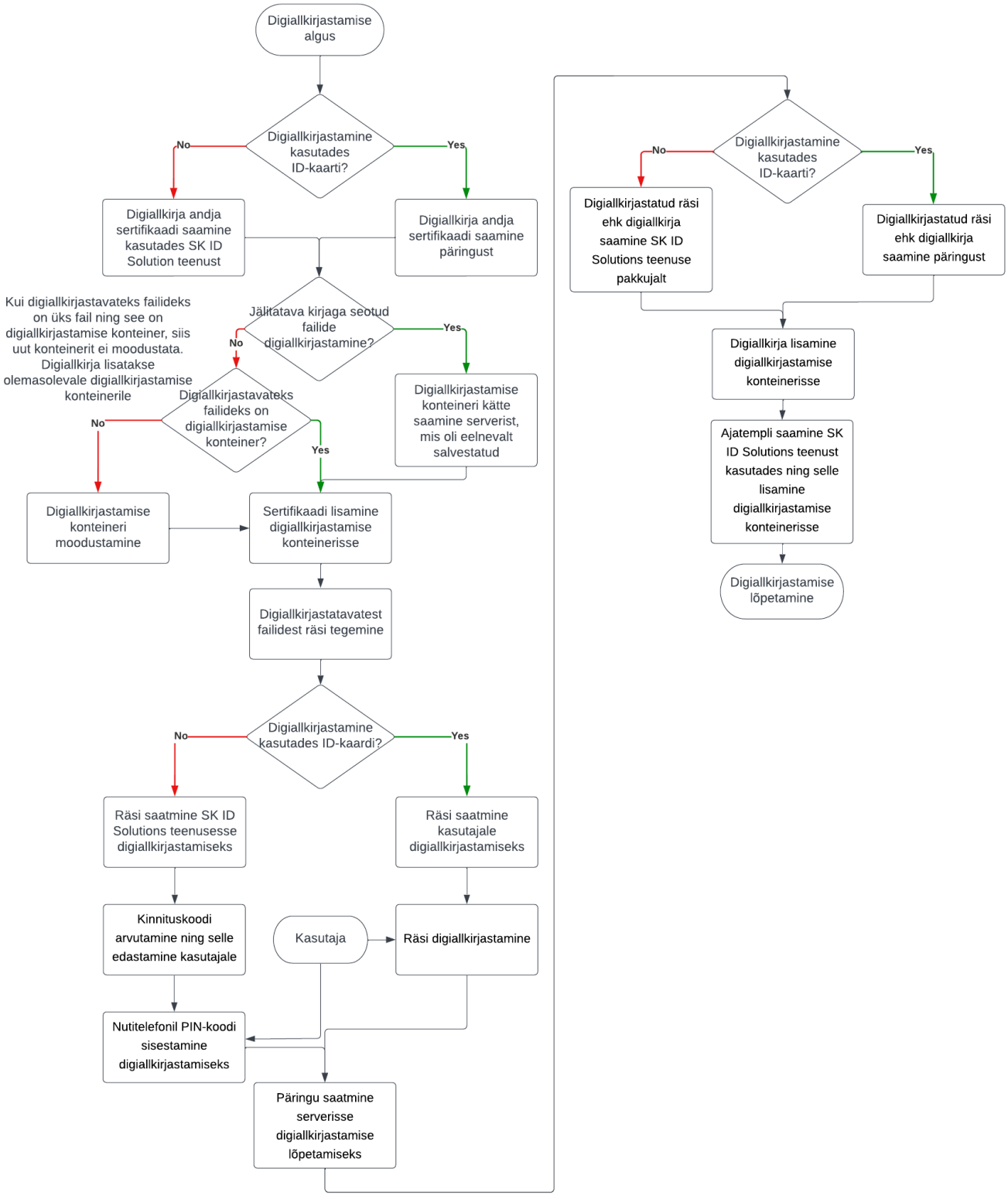
01.01.2023

1 Lihtlitsents ei kehti juurdepääsupiirangu kehtivuse ajal vastavalt üliõpilase taotlusele lõputööle juurdepääsupiirangu kehtestamiseks, mis on allkirjastatud teaduskonna dekaani poolt, välja arvatud ülikooli õigus lõputööd reprodutseerida üksnes säilitamise eesmärgil. Kui lõputöö on loonud kaks või enam isikut oma ühise loomingulise tegevusega ning lõputöö kaas- või ühisautor(id) ei ole andnud lõputööd kaitsvale üliõpilasele kindlaksmääratud tähtajaks nõusolekut lõputöö reprodutseerimiseks ja avalikustamiseks vastavalt lihtlitsentsi punktidele 1.1. ja 1.2, siis lihtlitsents nimetatud tähtaja jooksul ei kehti.

Lisa 2 – Digiallkirjastamise diagramm



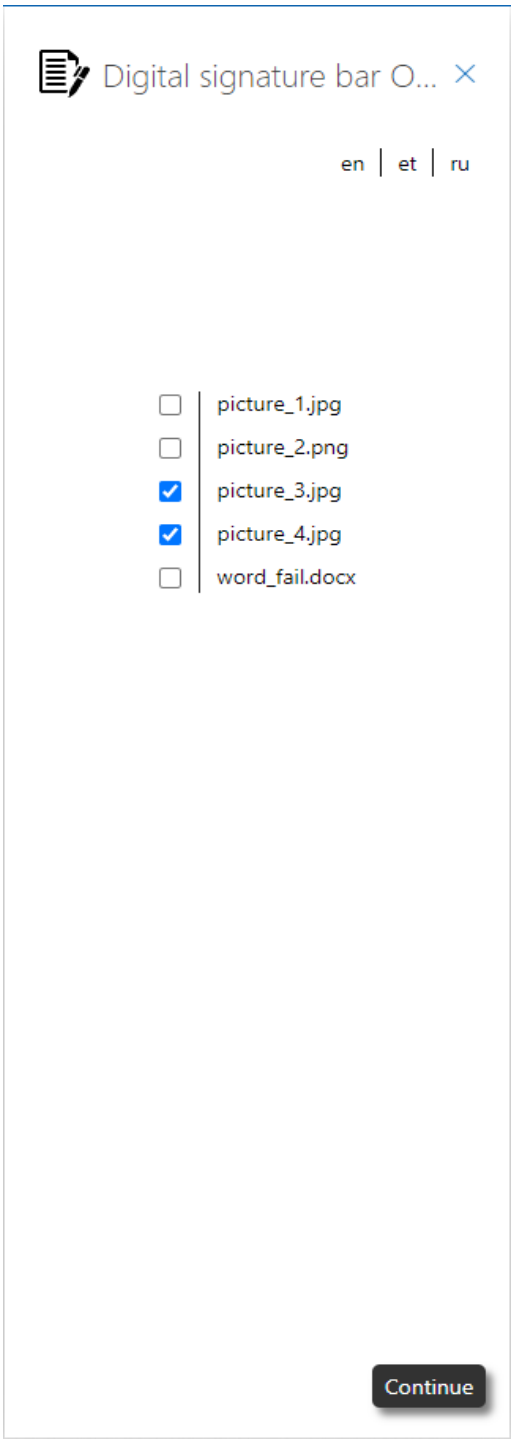
Joonis 7. Digiallkirjastamise protsessi diagramm



Joonis 8. Digiallkirjastamise protsessi diagramm

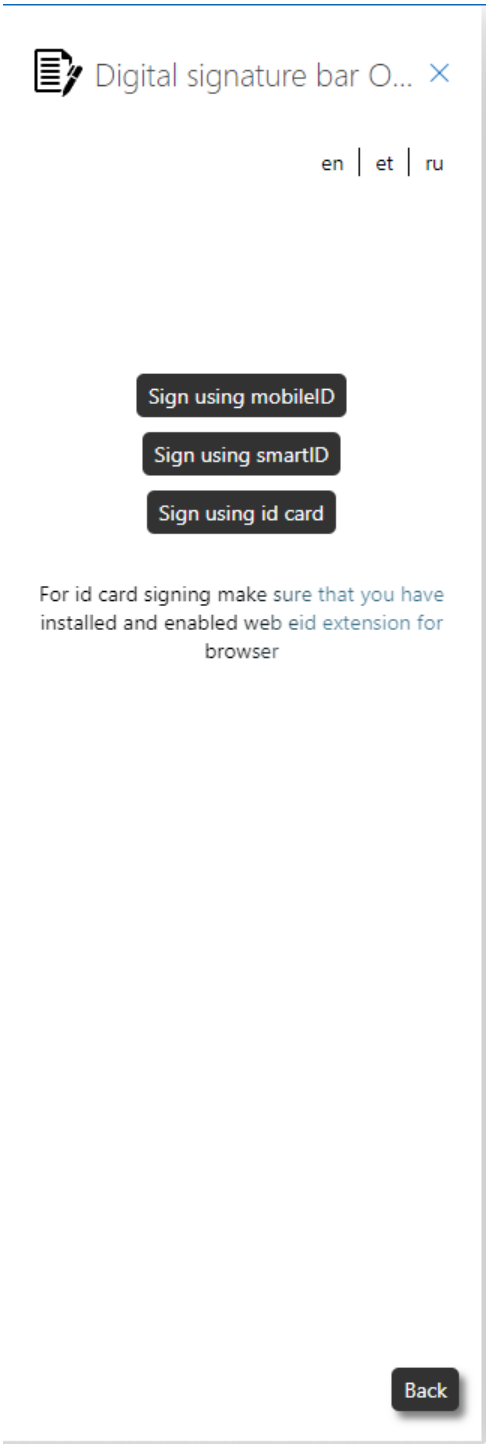
Lisa 3 – Digiallkirjastamise protsess

Joonisel 9 on välja toodud digiallkirjastamise protsessi algus Outlook'is. Antud vaade ei eksisteeri teistes keskkondades, kuna ainult Outlook keskkonnas on võimalik mitme faili digiallkirjastamine üheajaliselt. Nii Word, Excel kui ka PowerPoint keskkondades on digiallkirjastavaks failiks avatud rakenduse fail.



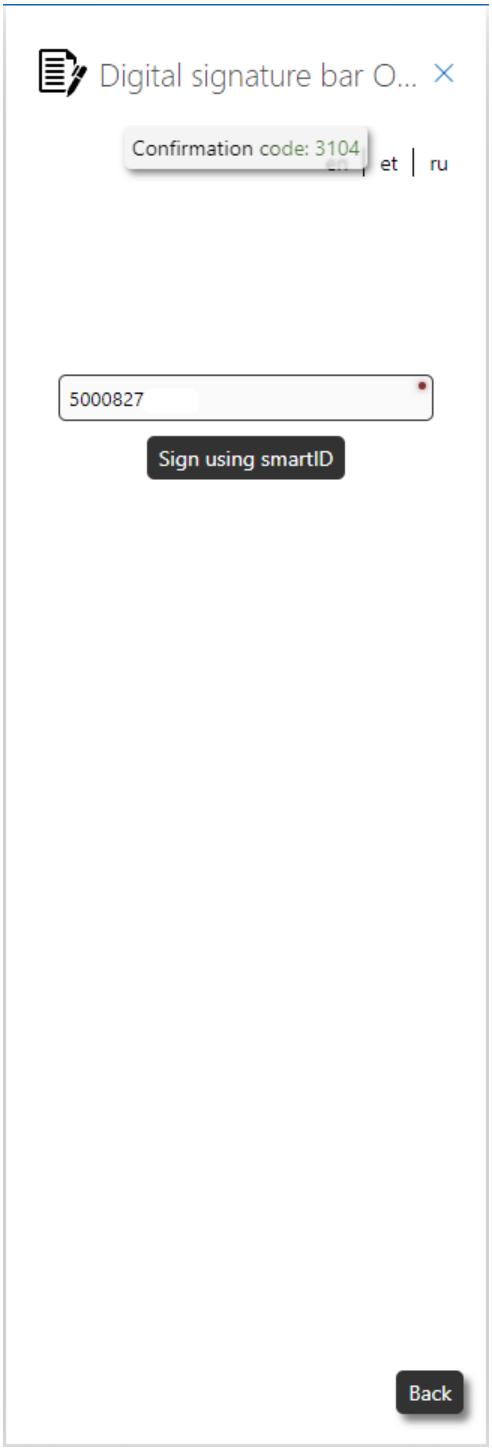
Joonis 9. Outlook digiallkirjastamiseks failide valimise vaade

Joonisel 10 on välja toodud digiallkirjastamise meetodi valimise vaade. Vaade järgneb failide valimise vaatele ning võimaldab valida digiallkirjastamise viisi.



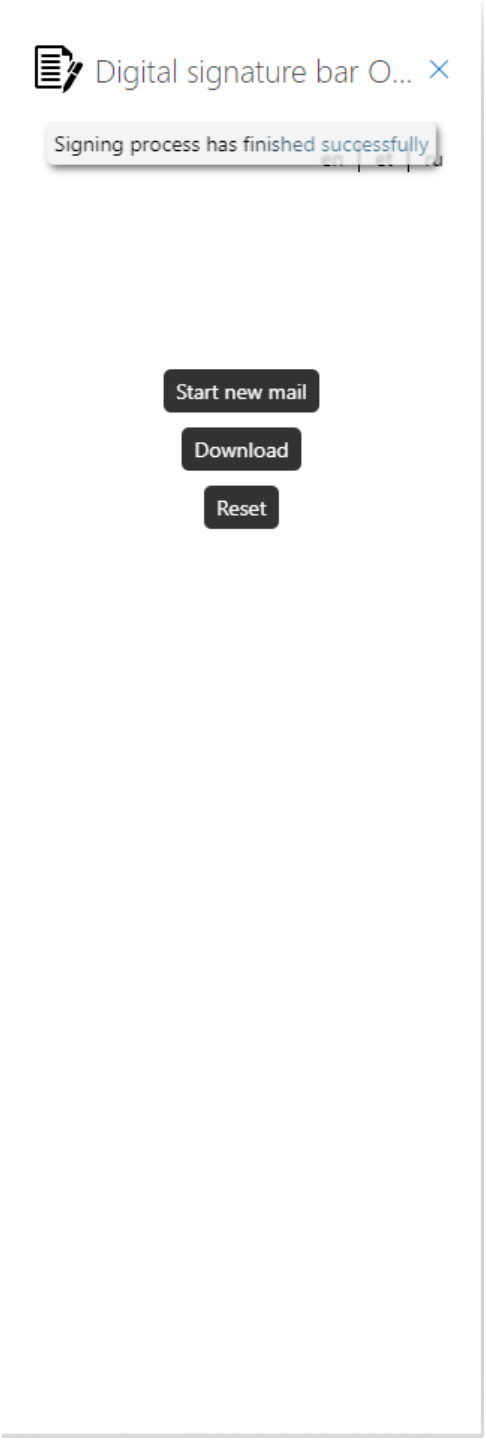
Joonis 10. Digiallkirjastamise meetodi valimise vaade

Joonisel 11 on välja toodud digiallkirjastamise protsessi vaade kasutades Smart-ID meetodit.



Joonis 11. Digiallkirjastamine kasutades Smart-ID vaade

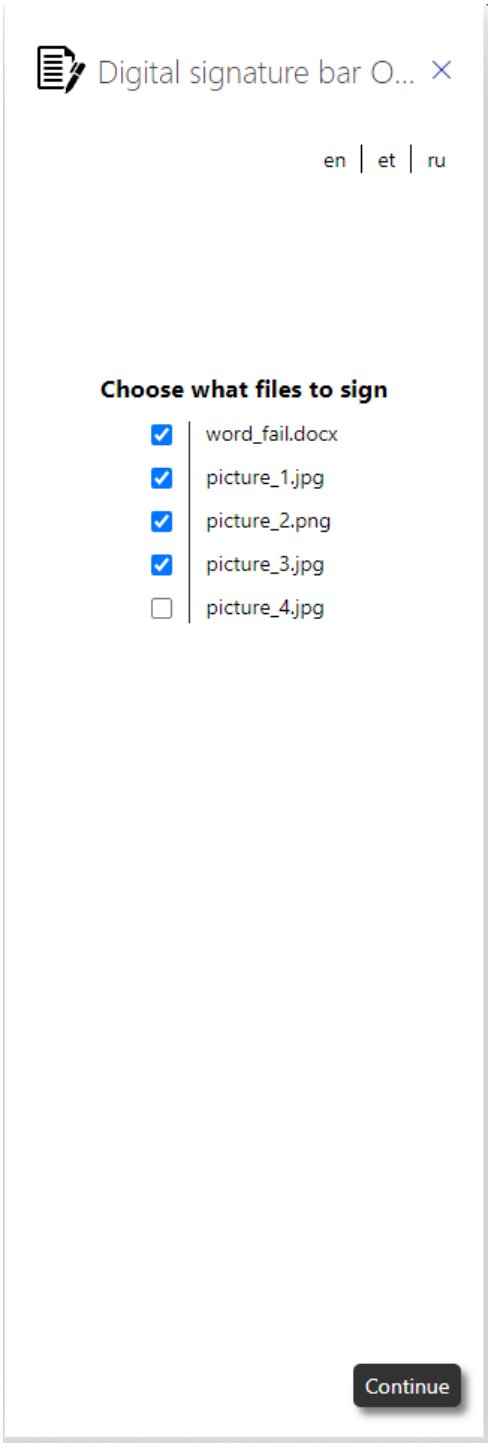
Joonisel 12 on välja toodud digiallkirjastamisele järgnev vaade. Outlook keskkonnas võimaldab pistikprogramm nii alla laadida digiallkirjastatud konteinerit, kui ka koostada uut kirja, kus digiallkirjastatud konteiner on vaikimisi kirja juurde lisatud. Word, Excel ja PowerPoint keskkondades on digiallkirjastamise järgselt võimalik vaid digiallkirjastatud konteineri alla laadimine.



Joonis 12. Digiallkirjastamisele järgnev vaade

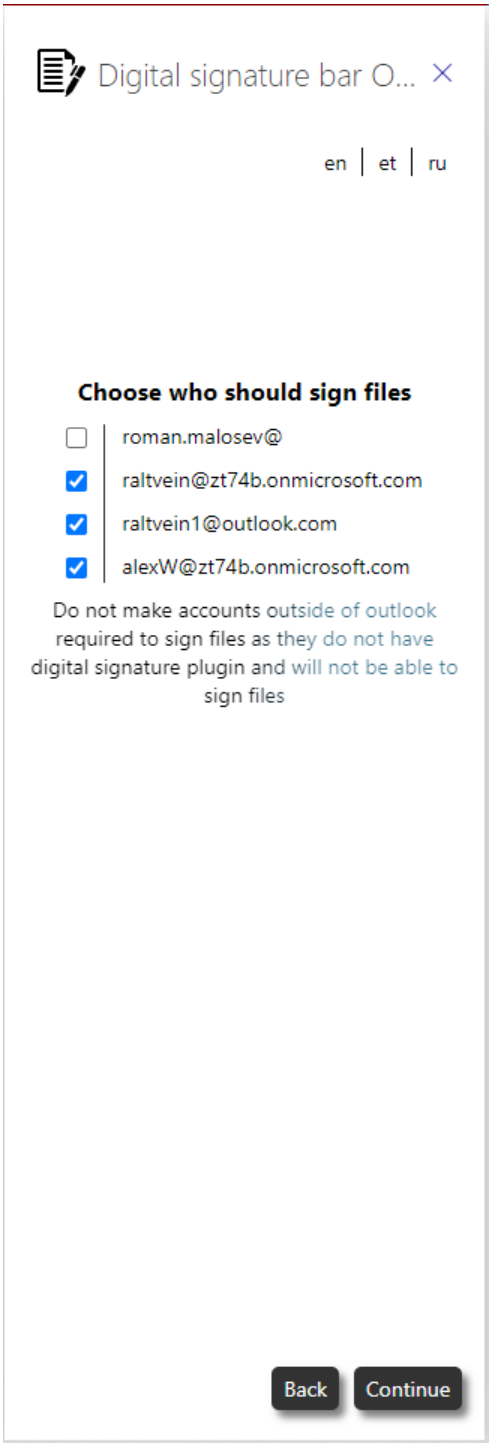
Lisa 4 – Jälitatud kiri

Joonisel 13 välja toodud jälitatud kirja loomise esimene vaade. Vaate raames on vajalik failide valimine, mida hakatakse jälitatud kirjas digiallkirjastama. Faile saab valida koostava kirjaga seotud failide hulgast.



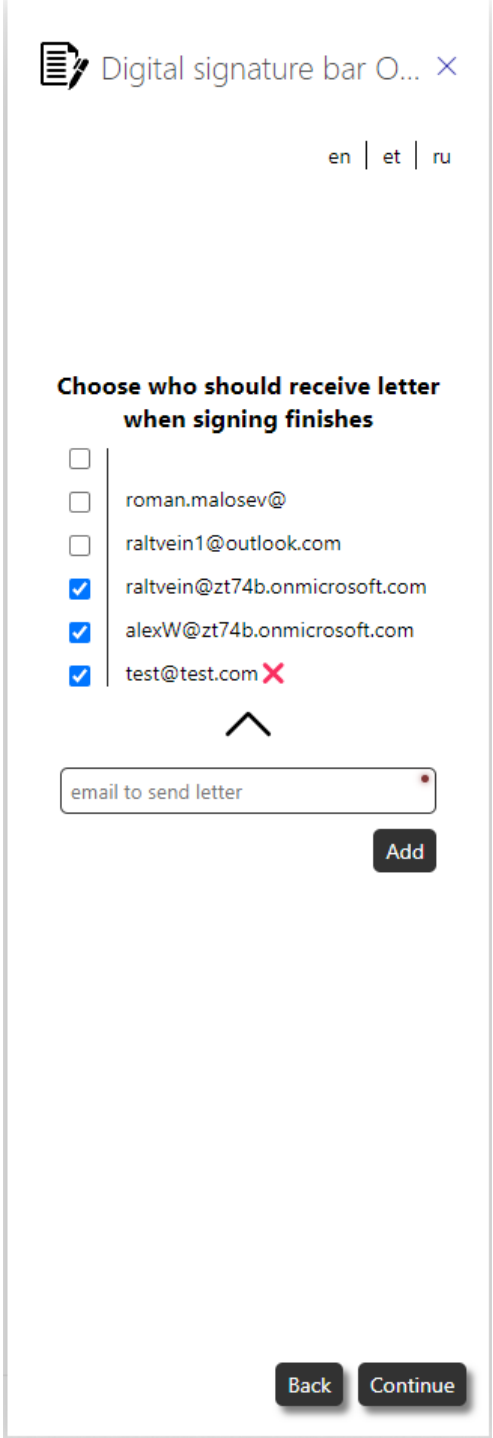
Joonis 13. Jälitatud kirja digiallkirjastamise failide valimise vaade

Joonisel 14 on välja toodud jälitatud kirja loomise teine vaade. Vaate raames on vajalik määrata isikuid, kes peavad koostatava kirja digiallkirjastama. Isikuid saab valida kirja saajate hulgast.



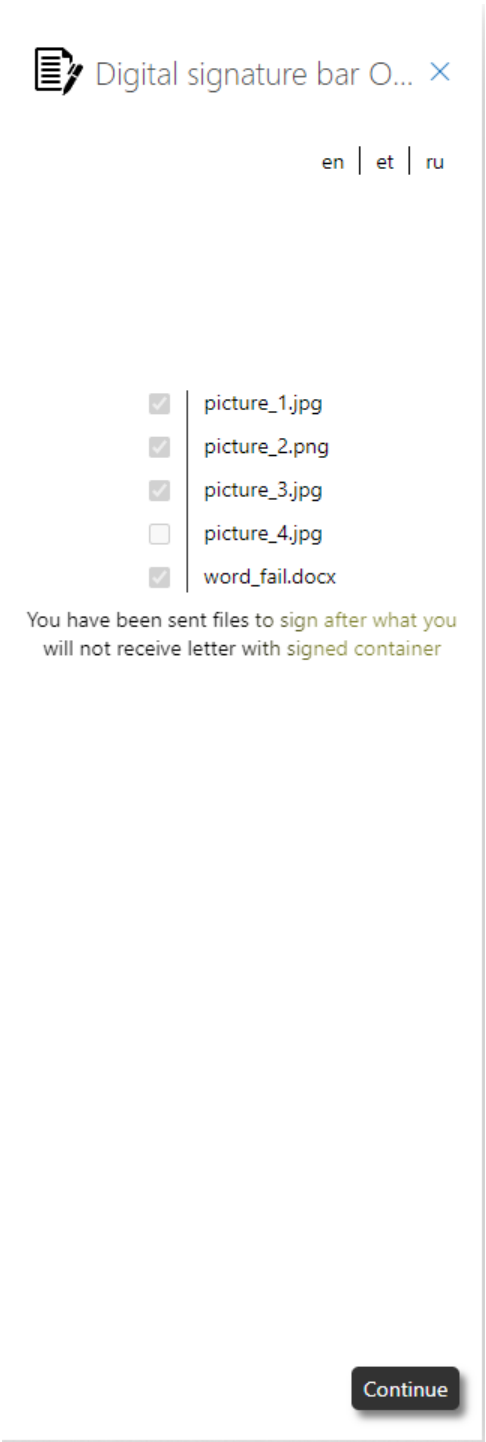
Joonis 14. Jälitatud kirja digiallkirja andjate määramise vaade

Joonisel 15 on välja toodud jälitatud kirja loomise kolmas ehk viimane vaade. Vaate raames on vajalik määrata isikud, kes saavad digiallkirjastatud konteineri pärast, kui kõik selleks enne määratud isikud on oma digiallkirja andnud. Antud etapi raames saab lisada uue e-maili, kuhu ei saadeta esialgset kirja, kuid kuhu saadetakse digiallkirjastatud konteinerit.



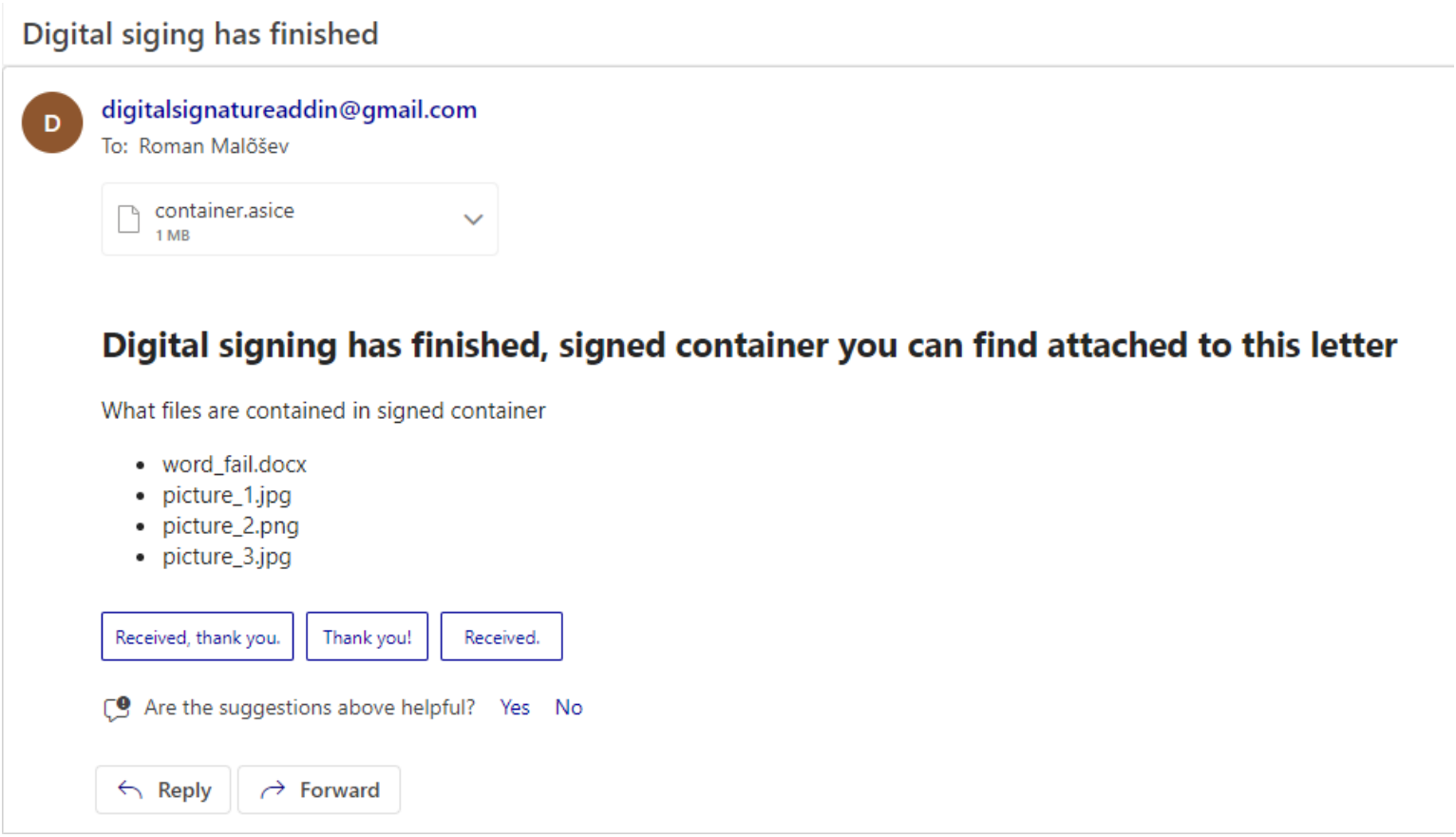
Joonis 15. Jälitatud kirja digiallkirjastatud konteineri saajate määramise vaade

Joonisel 16 on välja toodud jälitatud kirja digiallkirjastamise vaade. Vaate raames ei saa valida digiallkirjastatavaid faile, kuid vaade näitab, mis faile kirjast digiallkirjastatakse. Lisaks, annab vaade informatsiooni, kas digiallkirjastamise lõpus saadetakse digiallkirja andjale digiallkirjastatud konteiner või mitte.Vaatele järgneb digiallkirjastamise meetodi valimise vaade.



Joonis 16. Jälitatud kirja digiallkirjastamise vaade

Joonisel 17 on välja toodud jälitatud kirja digiallkirjastamise lõpus saadetud kiri digiallkirjastatud konteineriga. Kiri saadetakse ainult kirja koostamisel määratud isikutele, seejärel, kui kõik selleks määratud isikud on oma digiallkirja andnud.



Joonis 17. Kiri jälitatud kirja digiallkirjastamise konteineriga