IVXV arhitektuur

Arhitektuuridokument

Versioon 1.0

20. september 2017

36 lk

Dok IVXV-AR-1.0

Sisukord

PEATÜKK 1

Sissejuhatus

Elektroonilise hääletamise infosüsteem IVXV on loodud lähtuvalt e-hääletamise raamistikust *[ÜK2016]* ja riigihanke 171780 tehnilisest kirjeldusest *[TK2016]*. Selles dokumendis kirjeldatakse IVXV arhitektuurne lahendus. Elektroonilise hääletamise infosüsteem koosneb vallasrežiimirakendustest ning sidusrežiimikomponentidest. Täiendavalt sõltub infosüsteem välistest infosüsteemidest ning mõjutab vahetult elektroonilise hääletamiseks/hääle kontrollimiseks kasutatavaid komponente.

Arhitektuuridokument kirjeldab IVXV komponente, nende liideseid nii omavahel kui väliste süsteemidega ning komponentide poolt realiseeritavaid protokolle.

1.1 IVXV kontseptsioon

Üldine kuid terviklik ülevaade elektroonilise hääletamise raamistiku (“IVXV”) tehnilisest ja organisatsioonilisest poolest ning selle rakendamisest Eesti riiklikel valimistel on antud e-hääletamise raamistiku üldkirjelduses *[ÜK2016]*.

IVXV infosüsteemina realiseerib “ümbrikuskeemil” põhinevat e-hääletamisprotokolli. IVXV toimib hääletamiseelsel etapil, hääletamisetapil, töötlusetapil ning lugemisetapil, pakkudes vahendeid elektroonilise hääletamise protsessis osalemiseks Korraldajale, Lugejale, Hääletajale, Kogujale, Töötlejale, Miksijale, Audiitorile, Klienditoele, Valijate nimekirja koostajale ja täiendajale.

Infosüsteemi komponendid on Kogumisteenus, Töötlemisrakendus, Võtmerakendus ning Auditirakendus. Infosüsteemiga on tihedalt seotud Valijarakendus, Kontrollrakendus ning Miksimisrakendus.

Infosüsteem kasutab oma töös väliseid teenuseid - Tuvastusteenus, Allkirjastamisteenus ning Registreerimisteenus.

1.2 IVXV krüptograafiline protokoll

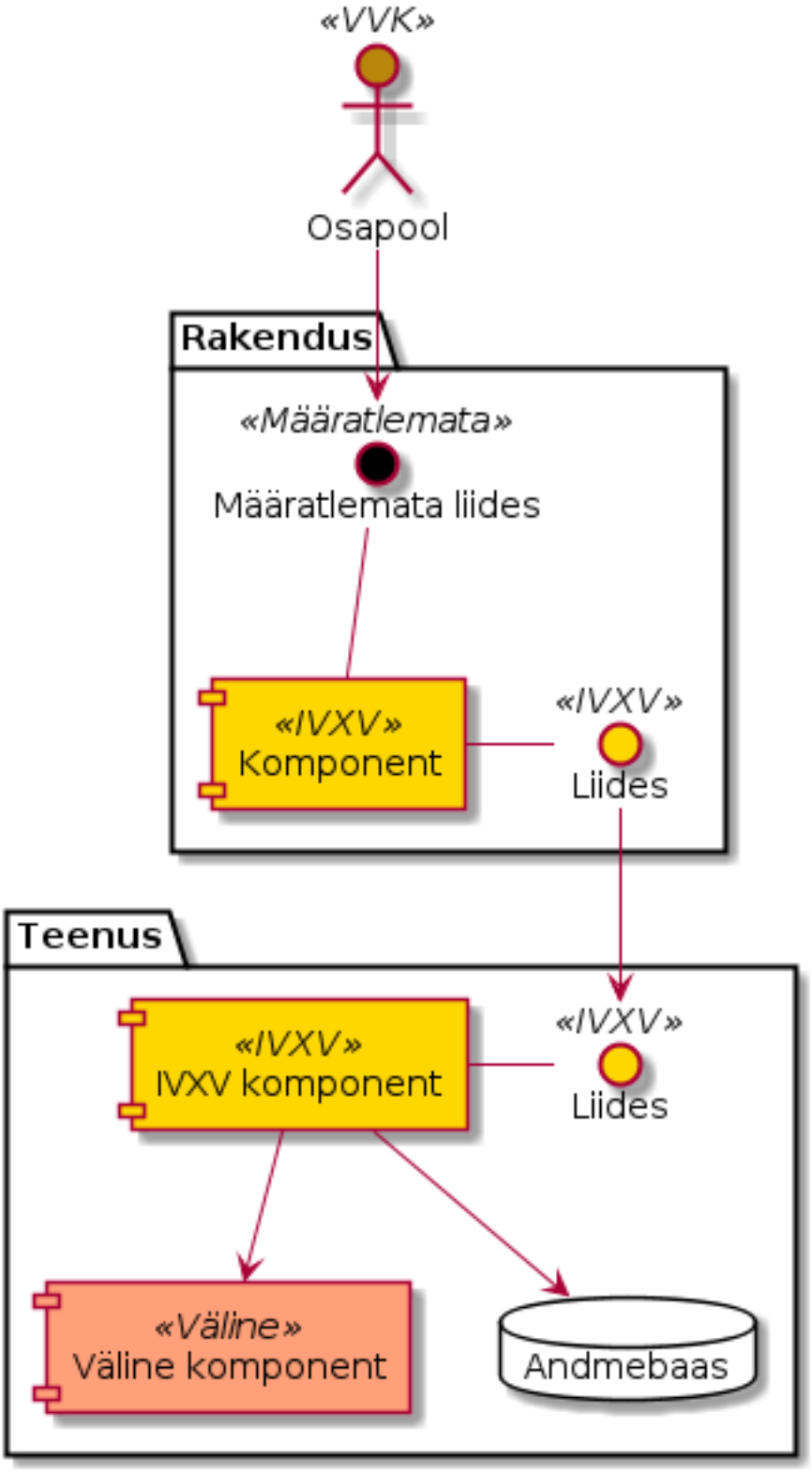
Elektroonilise hääletamise turvalisuse, verifitseeritavuse ning hääletamise salajasuse, hääletamise korrektsuse ja hääletaja sõltumatuse saavutamiseks on rangelt kirjeldatud elektroonilise hääletamise krüptograafiline protokoll *[HMVW16]*. Protokoll annab vajaliku ja piisava ülevaate IVXV ülesehitusest ning selle turvaaspektidest. IVXV komponendid realiseerivad krüptograafilise protokolli alamosi.

IVXV krüptograafiline protokoll on kirjeldatud ka protokolli turvaomaduste süsteemiga *[ProVerif]* formaalset verifitseerimist võimaldavas notatsioonis.

1.3 Notatsioon

Dokumendis kasutame arhitektuurse lahenduse visandi illustreerimiseks UML-skeeme, kus eristame värvide ja <<>> märgenditega kodeeritult olemite – tegijad, liidesed, komponendid – järgmisi aspekte:

- Märgend <<IVXV>> (Kollane) – infosüsteemi liides või komponent defineeritakse/realiseeritakse konkreetse pakkumuse raames tehtavate tööde käigus
- Märgend <<Väline>> (Punane) – infosüsteem sõltub mingi funktsionaalsuse realiseerimisel kolmanda osapoole komponendist või olemasolevast liidesest, mille ümberdefineerimine eeldab ka kolmandate osapoolte tööd.
- Märgend <<VVK>> (Pruun) – sarnane eelmisele, kuid liidese/komponendi omanikuks on VVK.
- Märgend <<Määratlemata>> (Must) – infosüsteemi jaoks oluline liides on määratlemata.



Joonis 1. Näiteskeem

PEATÜKK 2

Kogumisteenus

Üldkirjelduse *[ÜK2016]* põhjal on Kogumisteenus süsteemi keskne komponent, mida käitab Koguja. Teenus abistab Hääletajat e-hääle koostamisel ning registreerib selle enne salvestamist e-urni. Kogumisteenus kasutab väliseid teenuseid (tuvastamine, allkirjastamine, registreerimine). Kogumisteenusel on peale Koguja enda teisigi haldureid (Korraldaja, Klienditugi), kelle jaoks on Kogumisteenusel eraldi halduslиidesed.

Kogumisteenus töötab sidusrežiimis ning vähemalt valija- ja kontrollrakenduse suunalised liidesed on avatud internetile. Seega töötleb Kogumisteenus potentsiaalselt ebausaldusväärsest allikast pärit päringuid. Tulenevalt tarkvarale seatavast turvatasemest, kõrgkäideldavuse, skaleeritavuse, kihilise evitatavuse ning laiendatavuse nõuetest on kogumisteenus omakorda liigendatud ühte konkreetset teenust osutavateks mikroteenusteks, mida on võimalik paindlikult evitada.

Kõik kogumisteenuse komponendid programmeeritakse keeles Go (https://golang.org). Keelel Go on

- staatiline tüüpimine, mis võimaldab tüübivigade avastamist enne programmi käivitamist,
- automaatne mäluhaldus, mis välistab rakenduse vigasest mäluhaldusest tulenevad turvaaugud,
- kompilaator avatud lähtekoodiga ning
- ribastamine/rööprapse, mis võimaldab kasutada paralleelsust mitmetuumalistes süsteemides.

Kogumisteenuse andmeedastusvormingutes kasutatakse üldjuhul JSON-it väljaarvatud olukordades, kus välised asjaolud tingivad mõne muu andmevormingu kasutamist – näiteks BDOC vorming baseerub XML-il.

Kogumisteenus toetab Riigikogu valimisi, kohaliku omavalitsuse volikogu valimisi, Euroopa parlamendi valimisi ning rahvahääletusi.

Kogumisteenuse komponendid arvestavad virtualiseerimistehnoloogiate kasutamisega ning kogumisteenust on võimalik evitada nii ühel virtuaalriistvara instantsil, kui ka

mikroteenuste kaupa erinevatel instantsidel. Kogumisteenuse komponendid on evitatavad Ubuntu LTS 16.04 operatsioonisüsteemil 64-bitisel arhitektuuril.

Andmesäilitus on realiseeritud kasutades võti-väärtus andmebaasi (etcd). Testotstarbel on teostatud ka andmesäilitus failisüsteemi ning mällu, kuid neid ei ole soovituslik kasutada tootekeskkonnas. Lisaks on kogumisteenusel olemas liides uute talletusprotokollide lisamiseks. Lõplik otsus kasutatava lahenduse kohta tehakse kogumisteenuse administraatorite poolt teenust seadistades.

2.1 Mikroteenused

Kogumisteenus on jaotatud põhiteenusteks ja abiteenusteks. Põhiteenused - vahendusteenus, nimekirjateenus, hääletamisteenus, kontrollteenus ning talletamisteenus - on arhitektuuri tehnilise lihtsuse mõttes piiritletud ühe valimisega, kuid ühel riistvaral, ühe operatsioonisüsteemi kontekstis võivad käia mitme valimise mikroteenused. Täiendavalt võib kogumisteenuse juures kasutada abiteenuseid - tuvastusteenust hääletaja isiku tuvastamiseks ning allkirjateenust valijarakenduse poolt hääle allkirjastamise hõlbustamiseks.

Teenuseid on võimalik evitada nii eraldatult kui koos erinevates konfiguratsioonides, mis teeb võimalikuks kihilise arhitektuuri. Lähtudes funktsioonist on otstarbekas hoida Vahendus- ning Talletamisteenused teistest eraldi.

Teenused kasutavad transpordiprotokollina TLS-i, kõik ühendused on mõlemapoolselt autenditud. Rakenduskihi protokoll on JSON-RPC.

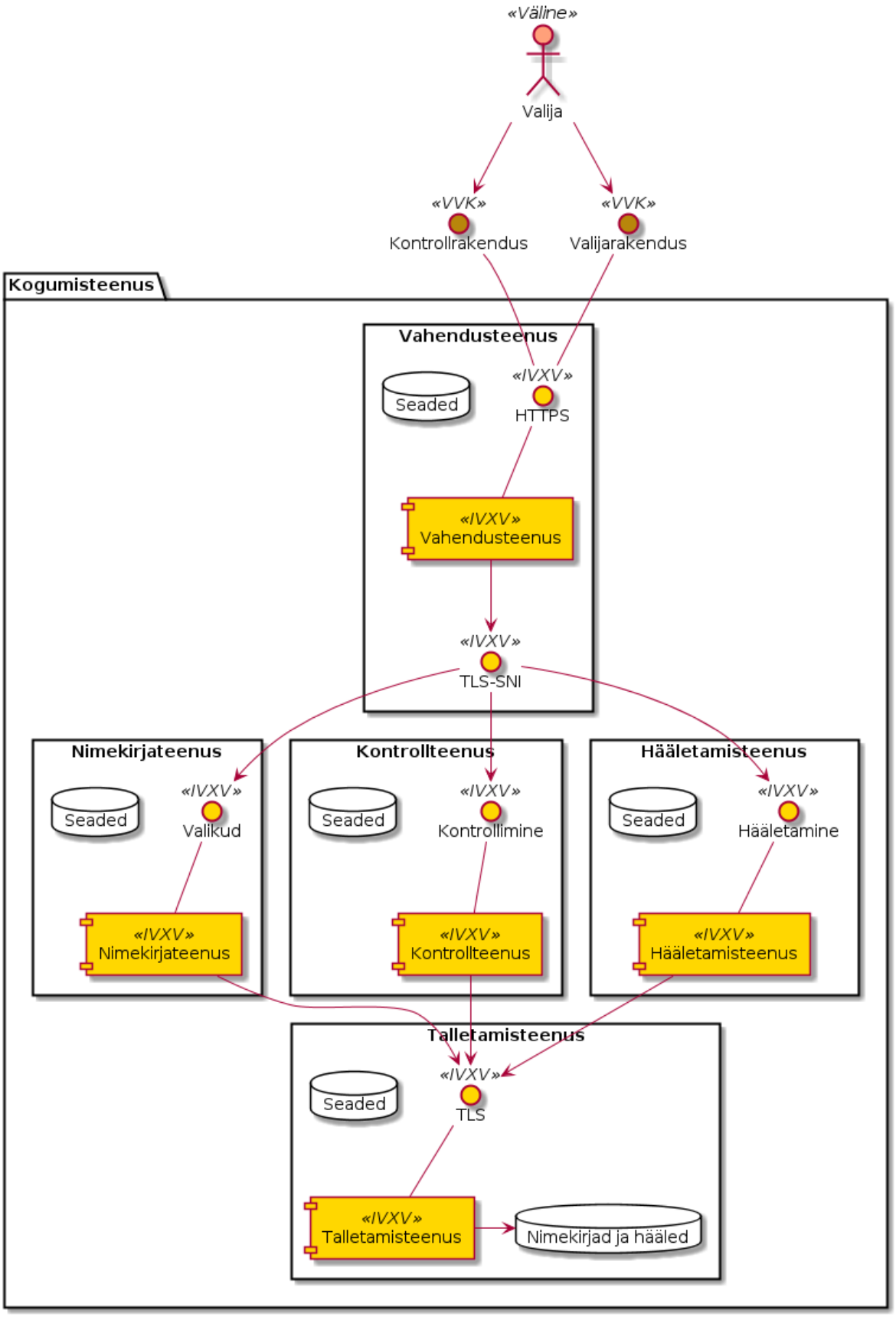
Kõik teenused tekitavad tegevuslogi, mida säilitatakse nii lokaalselt kui logitakse rsyslog liidese vahendusel.

Vahendusteenuse funktsioon ja tehniline liides

Vahendusteenuse põhifunktsiooniks on ühe sisenemispunkti (port 443) pakkumine Valijarakendusele ja Kontrollrakendusele. Vahendusteenus on dispetšerteenus teiste komponentide vahel, mis võimaldab sisemiselt evitada kogumisteenust mikroteenustena, ent omada süsteemi ainult ühte sisenemispunkti. Lisaks suudab see dubleeritud evituse puhul täita koormusjaoturi ülesannet.

Vahendusteenus ei termineeri TLS-ühendust vaid kasutab TLS-i *Server Name Indication* (SNI) laiendust sihtpunkti tuvastamiseks. Kliendid panevad TLS `ClientHello` sõnumisse SNI laiendi, kus avatekstis määravad, millise teenusega soovivad suhelda: vahendusteenus näeb seda, võtab ühendust vastavat teenust pakkuva isendiga ja hakkab kliendi ning teenuse vahelisi sõnumeid vahendama. Vahendusteenus EI termineeri TLS-i ning ei näe sõnumite sisu. Vahendusteenusel on andmed kõigi teiste teenuste asukohtadest (aadress:port) ning teenus vahendab sõnumivahetust kõigi osapoolte vahel.

Vahendusteenus on olekuvaba komponent, mida on võimalik horisontaalselt skaleerida.



Joonis 2. Kogumisteenuse jaotus mikroteenusteks

Vahendusteenuse teostus

Vahendusteenuse teostus kasutab vabavaralist HAProxy serverit, mis on üldlevinud tarkvaraline koormusjaotur ja proksi. Kuna Vahendusteenus on esimene puutepunkt avalikust internetist tulevate ühenduste jaoks, siis on mõistlik kasutada tarkvara, mille töökindlus on juba tõestatud.

Kuigi HAProxyt kasutatakse tihti HTTP-režiimis, kus see analüüsib liiklust, siis vahendusteenuse rollis on see TCP-režiimis ning ei näe vahendatava krüpteeritud TLS-kanali sisse.

IVXV seadistusest genereeritakse HAProxy seadistusfail, mis sisaldab teiste teenuste asukohti, ning ühenduste vahendamise ülesanne jääb viimase kanda. Lisaks on võimalik HAProxyt ka seadistada ühenduste sagedusi piirama lähteaadressi või mõne muu nimetaja põhjal. See aga jääb süsteemihalduri ülesandeks.

Kuigi HAProxy on võimeline ise teostama koormusjaoturi ülesannet, on seda võimalik evitada ka teiste, potentsiaalselt riistvaraliste koormusjaoturite taha, kus see jääb täitma ainult SNI põhjal vahendmise ülesannet.

HAProxy lähtekood on avalik GPL v2 all ning versioon 1.6.3 on pakendatud Ubuntu 16.04 ametlikus hoidlas.

Nimekirjateenuse funktsioon ja tehniline liides

Nimekirjateenuse põhifunktsiooniks on valikute nimekirjade vahendamine Valijarakendusele. Nimekirjateenusesse jõuab informatsioon tuvastatud valija kohta ning Nimekirjateenus väljastab valija ringkonnale vastava valikute nimekirja Talletamisteenusest Valijarakendusse.

Nimekirjateenus on olekuvaba komponent, mida on võimalik horisontaalselt skaleerida.

Kontrollteenuse funktsioon ja tehniline liides

Kontrollteenuse põhifunktsiooniks on kontrollpäringute töötlemine ning kontrollitava hääle väljastamine Talletamisteenusest Kontrollrakendusse.

Kontrollteenus on olekuvaba komponent, mida on võimalik horisontaalselt skaleerida.

Hääletamisteenuse funktsioon ja tehniline liides

Hääletamisteenuse põhifunktsiooniks on hääletamispäringute töötlemine. Hääletamisteenus verifitseerib sissetuleva hääle, registreerib selle Registreerimisteenuses ning talletab Talletamisteenusesse.

Hääletamisteenus on olekuvaba komponent, mida on võimalik horisontaalselt skaleerida.

Talletamisteenuse funktsioon ja tehniline liides

Talletamisteenuse põhifunktsiooniks on valikute ja valijanimekirjade ning häälte pikaajaline talletamine.

Talletamisteenuse horisontaalseks skaleerimiseks tuleb kasutada hajustalletamist võimaldavat säilitustehnoloogiat.

Talletamisteenuse teostus

Talletamisteenus ei ole teadlik IVXV protokollist ega talletatavate andmete spetsiifikast, vaid on üldkasutatav võti-väärtus andmebaas binaarandmete säilitamiseks. Kogu teadmus talletatavate andmete struktuurist ja võtmete hierarhiast on teistes, Talletamisteenust kasutatavates teenustes, mis käituvad nii-öelda “tarkade” klientidena.

Selline lähenemine lubab kasutada ükskõik millist üldlevinud võti-väärtus andmebaasi Talletamisteenusena ilma suurema vaevata: ainsateks ülesanneteks on IVXV seadistuse teisendamine andmebaasi jaoks sobilikku vormingusse ning teenuse käivitamine. Andmebaasi tarkvara peab võimaldama vaid võtme järgi talletamist ja lugemist, võtmete prefiksi järgi loetlemist ning atomaarset võrdle-ja-vaheta (*compare-and-swap*) operatsiooni.

Talletamisteenus on kogumisteenuse töökiiruse oluliseks määrajaks: seetõttu mõjutab seda teenust pakkuv riistvara kogu süsteemi jõudlust ning see tuleks vastavalt kasutatavale andmebaasile dimensioneerida.

Hetkel ainus tooteks mõeldud Talletamisteenuse teostus kasutab hajusat võti-väärtus andmebaasi etcd. Selle puhul tuleks järgida etcd autorite riistvara soovitusi aadressil https://coreos.com/etcd/docs/latest/op-guide/hardware.html.

Tuvastusteenuse funktsioon ja tehniline liides

Tuvastusteenuse põhifunktsiooniks on valija identiteedi tuvastamine. Tuvastusteenus on vajalik näiteks Mobiil-ID autentimise korral.

Allkirjateenuse funktsioon ja tehniline liides

Allkirjateenuse funktsiooniks on Valijarakenduse toetamine hääle allkirjastamisel. Allkirjateenus on vajalik näiteks Mobiil-ID allkirjastamise korral.

Mobiil-ID abiteenuse teostus

IVXV koosseisu kuulub Mobiil-ID abiteenus, mis käitub Mobiil-ID jaoks nii Tuvastusteenusena kui ka Allkirjateenusena. Valijarakendus esitab IVXV päringud Mobiil-ID abiteenusele, mis teisendab need Mobiil-ID päringuteks ning edastab Mobiil-ID teenusepakkujale.

Eduka Mobiil-ID isikutuvastuse korral väljastab abiteenus Valijarakendusele pileti, mille abil on võimalik teistele teenustele valija identiteeti kinnitada. Iga piletiga saab hääletada ainult ühe korra.

Allkirjastamise korral saadab Valijarakendus Mobiil-ID abiteenusele vaid allkirjastatava hääle räsi ning kasutab vastuseks saadud signatuuri samamoodi kui ID-kaardiga loodud signatuuri.

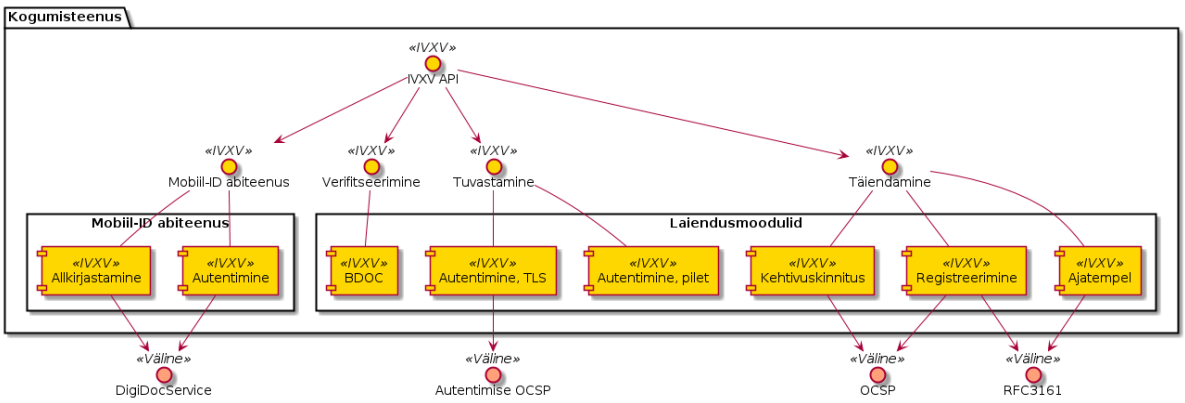
Mobiil-ID abiteenus sisaldab küll olekut pooleliolevate tuvastusseansside kohta, aga muus osas on tegu olekuvaba komponendiga. Tänu sellele on võimalik Mobiil-ID abiteenust horisontaalselt skaleerida, eeldusel et ühe tuvastusseansi kõik päringud edastatakse samale isendile.

Kogumisteenuse mikroteenuste evitamine

Kogumisteenuse mikroteenused sõltuvad välistest pakkidest minimaalselt. Soovitatav on rsyslog teenuse kasutamine.

Kogumisteenuse mikroteenused pakendatakse deb vormingus, neid on võimalik evitada ka docker'i-laadsete konteineritena.

2.2 Välised teenused ja laiendatavus



Joonis 3. Kogumisteenuse laiendusmoodulid ja välised teenused

Kogumisteenuse mikroteenused kasutavad laiendusmooduleid teostamaks erinevaid mehhanisme valija tuvastamiseks, digiallkirjade verifitseerimiseks ja täiendamiseks, sealhulgas hääle registreerimiseks. Laiendusmoodulid võivad teostuse võimaldamiseks kasutada väliseid teenuseid. Mikroteenuste laiendatavuse huvides on defineeritud Go API, mille alusel saab realiseerida ka täiendavaid mooduleid. Hetkel on realiseeritud järgmised moodulid:

- Autentimine TLS sertifikaadiga (ID-kaart)
- Autentimine Tuvastusteenuse piletiga (Mobiil-ID)
- BDOC verifitseerimine

- Kehtivuskinnitusteenus OCSP
- Ajatempliteenus RFC 3161
- Registreerimisteenus OCSP
- Registreerimisteenus RFC 3161

IVXV krüptograafilises protokollis on kesksel kohal Registreerimisteenus, mis osaleb samuti häälte pikaajalisel talletamisel.

Registreerimisteenuse funktsioon

Registreerimisteenuse põhifunktsioon on võtta Hääletamisteenuselt vastu allkirjastatud registreerimispäringuid, kinnitada neid omapoolse allkirjastatud vastusega ning säilitada vähemalt hääletamisperioodi lõpuni, hilisemaks auditeerimiseks.

Auditeerimisel tekkivate võimalike erisuste lahendamiseks on oluline, et

- Registreerimisteenus on võimeline tõestama, et igale tema poolt väljastatud kinnitusele eelnes Talletamisteenuse poolne registreerimispäring
- Talletamisteenus on võimeline tõestama, et iga tema poolt talletatud hääle kohta on olemas Registreerimisteenuse kinnitus

Piisav protokoll sellise tõendamistaseme saavutamiseks on, kus mõlemal osapoolel on olemas võtmepaar allkirjastamiseks, päringud ja vastused on allkirjastatud ning kumbki pool peab registrit teise poole teadete üle. Selline protokoll on realiseeritav näiteks OCSP-põhise Registreerimisteenuse korral. Samas võib esineda juhtumeid, kus näiteks registreerimispäringute allkirjastamine ei ole standardsete vahenditega võimalik - RFC 3161 põhine registreerimine - sellisel juhul tuleb registreerimisteenusele vajalik tõendusmaterjal anda muude organisatsioonilis-tehniliste vahenditega.

Registreerimisteenusel on täna kaks erinevat teostust:

- OCSP liides eeldab Eestis rakendatava OCSP-põhise ajamärgendamisteenuse kasutamist, kus allkirjastatud OCSP-päringu nonsiks on Hääletamisteenuse poolt pandud hääle räsi. Päring on allkirjastatud standardsete OCSP vahenditega.
- RFC 3161 liides, mille korral ebastandardse lahendusena pannakse ajatemplipäringu nonsiks Hääletamisteenuse poolt allkirjastatud hääle räsi.

Kogumisteenuse laiendusmoodulite lisamine

Kogumisteenuse API defineerib kuute tüüpi laiendusmooduleid:

- isikutuvastus (Go pakk `ivxv.ee/auth`, näiteks `tls`),
- tuvastatud isiku sertifikaadist valija identifikaatori tuletamine (Go pakk `ivxv.ee/identity`, näiteks `serialnumber`),
- valija identifikaatorist vanuse tuletamine (Go pakk `ivxv.ee/age`, näiteks `estpic`),

- allkirjastatud konteineri verifitseerimine (Go pakk `ivxv.ee/container`, näiteks `bdoc`),
- allkirja kvalifitseerimine (Go pakk `ivxv.ee/q11n`, näiteks `tspreg`) ja
- andmetalletusprotokoll (Go pakk `ivxv.ee/storage`, näiteks `etcd`).

Uue mooduli lisamiseks tuleb moodulpakki lisada uue mooduli identifikaator ning mooduli teostusega alampakk. Alampaki alglaadimisel tuleb kutsuda välja moodulpaki `Register` funktsioon mooduli registreerimiseks.

Uue mooduli kasutamiseks tuleb selle identifikaator lisada seadistusse vastava moodulitüübi seadistuse juurde koos alammooduli seadistusega. Laiendusmoodulile antakse ette tema identifikaatoriga viidatud seadistusblokk, mida see mooduli-siseselt edasi töötleb.

Moodulpakid ja nende moodulitelt nõutavad liidesed on täpsemalt kirjeldatud dokumendis `IVXV API`. Samuti on iga mooduli kohta olemas vähemalt üks teostus, mida saab kasutada eeskujuna.

2.3 Monitooring

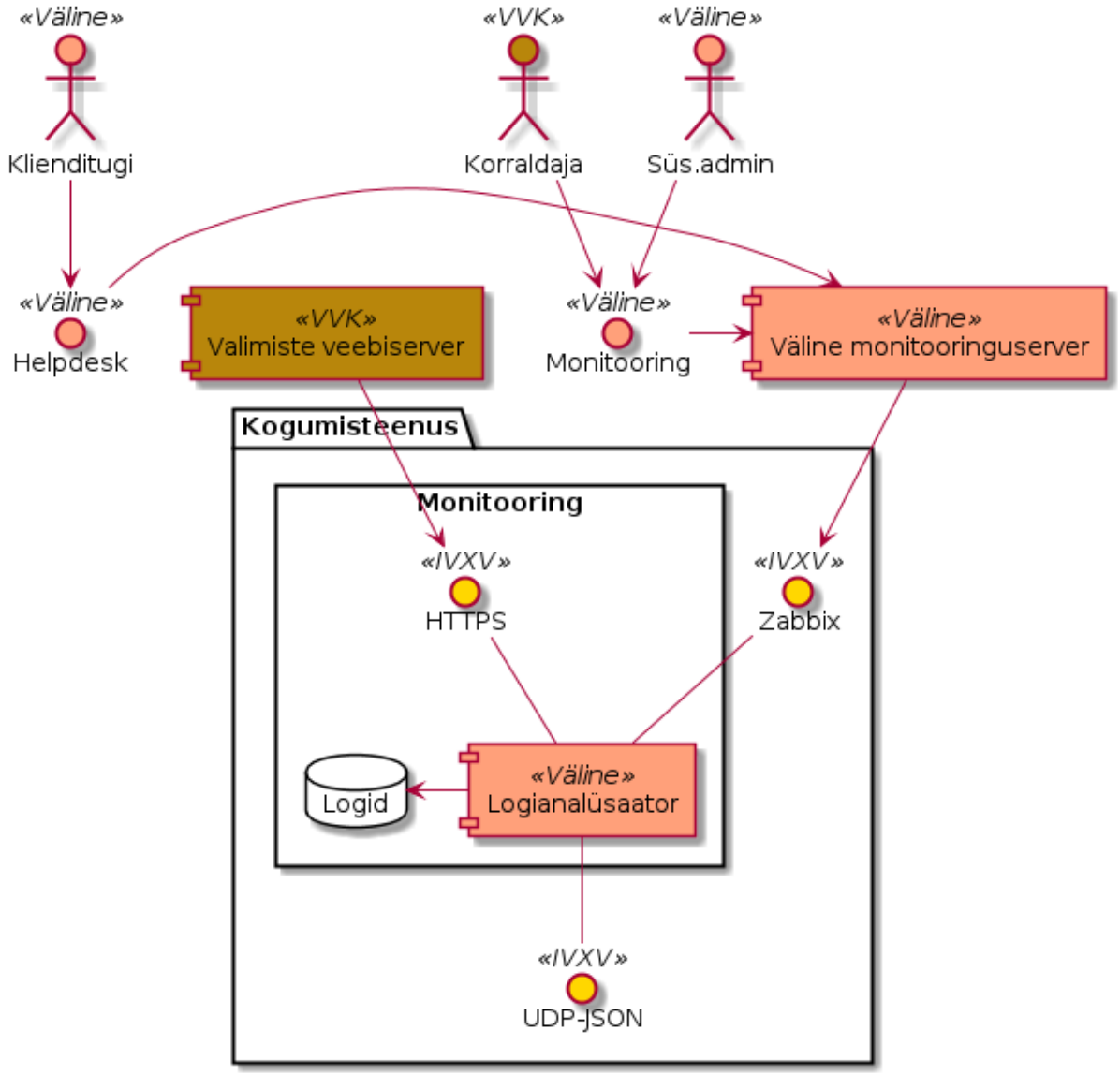
Logimine

Iga mikroteenuse poolt genereeritav logi defineeritakse süstemaatiliselt, lähtudes protokollikirjeldusest ning teenuse osutamise olekudiagrammist. Logitakse minimaalselt:

- iga päringu kättesaamise fakt ning töötlemise algus;
- töötlemise üleandmine välisele komponendile;
- töötlemisjärje naasmine komponenti;
- päringu töötlemise lõpp ning tulemus;
- täiendavalt oluliste etappide läbimine protsessi olekumudelis.

Logimisel järgitakse järgmisi põhimõtteid:

- Logimiseks kasutatakse rsyslog teenust, mis registreerib logiteate kirjutamise hetke millisekundi täpsusega;
- Iga seansi alustamisel genereerib süsteem unikaalse identifikaatori, mida klientrakendus kasutab oma päringutel kesksüsteemi poole pöördumiseks;
- Kõik ühe seansi alla kuuluvad logikirjed sisaldavad sama seansiidentifikaatorit;
- Logikirje on unikaalselt identifitseeritav;
- Iga logitava teate juures on võimalik unikaalse tunnuse abil üksüheselt tuvastada teate tekkimise koht monitooritavas süsteemis;
- Logikirje on JSON vormingus, automaatse monitooringu jaoks on masinloetavus primaarne ning inimloetavus sekundaarne;
- Logisse minev info saneeritakse (urlencode), peale pannakse pikkuse piirang (kogu piirang ja parameetri kaupa);



Joonis 4. Monitooringulahendus

- Süsteemiperimeetrist väljastpoolt pärinevat infot logitakse ainult saneerituna, ainult etteantud pikkuses.

Kuna logimine toimub rsyslog vahendusel, on võimalik Guardtime mooduli kasutamine logide tervikluse tagamiseks.

Üldstatistika

Järgmise statistika jälgimiseks kasutatakse staatilist veebiliidest

- edukalt kogutud hääled/hääletajate hulk;
- hääletajate jagunemine sugude, eagruppide, operatsioonisüsteemide ning autentimisvahendite kaupa;
- edukalt kontrollitud häälte/hääletajate hulk;

- korduvhääletamiste statistika;
- hääletajate jagunemine riigiti IP-aadressi põhjal.

Detailstatistika

Detailstatistika agregeeritakse logide põhjal kasutades SCCEIV logianalüsaatorit, mis analüüsib rakenduste tegevuslogi eeldefineeritud profiili suhtes ning võimaldab seansi-/veatüübipõhist analüüsi.

Detailstatistika on kättesaadav üle HTTPS liidese.

2.4 Haldus

Kogumisteenuse administreerimine toimub digitaalallkirjastatud seadistuspakkide abil.

Kogumisteenus pakub seadistuspakkide laadimiseks kahte liidest:

- Käsurealiides – rakendus verifitseerib allkirja, valideerib korralduste kooskõlalisust ja sobivust kogumisteenuse seisundi suhtes. Korralduse rakendamine toimub eraldi utiliidi abil.
- Veebiliides – veebiliides vahendab seadistuspaki käsurealiidesele ja tagastab kasutajale info laadimise tulemuse kohta. Eduka laadimise korral toimub automaatselt ja samadel põhimõtetel ka seadistuspaki rakendamine.

Veebiliidese funktsioonideks on:

- Kogumisteenuse mikroteenuste seisundi jälgimine;
- Valimiste nimekirjade haldus;
- Statistika kuvamine e-hääletamise kulgemise kohta;
- Haldusteenuse kasutajate haldus;
- Kogumisteenuse halduse logi kuvamine.

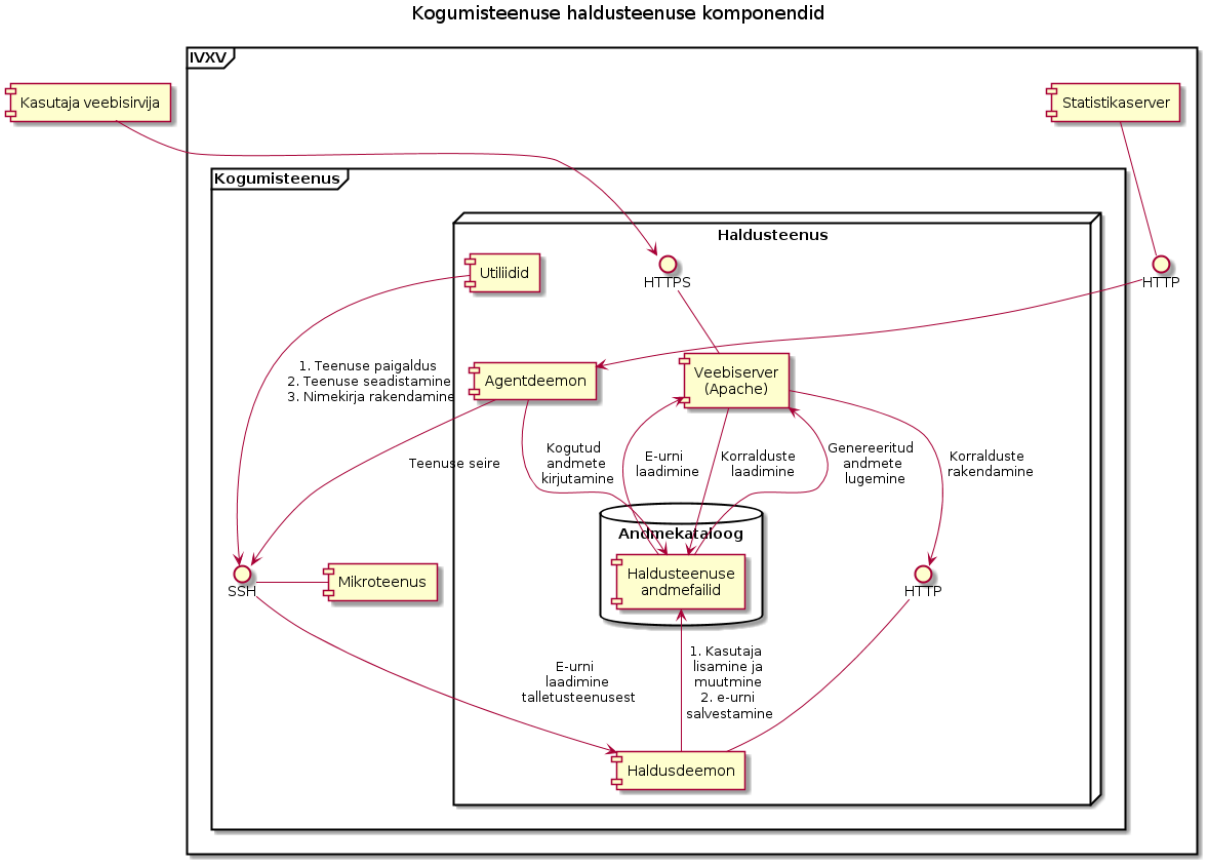
Kõik rakendusele antud korraldused säilitatakse - ka need mida ei rakendatud,

Kogumisteenus võib järgmisi tegevusi teostada automaatselt:

- Talletatud häälte, logide ning seadistuste varundamiseks ettevalmistamine – konteinerisse pakendamine.

Haldusteenuse komponendid

1. **Halduse veebiserver** on süsteemse kasutaja `www-data` õigustes töötav Apache server, mille ülesanded on:
   1.1 Kasutajatelt tulevate HTTPS-päringute esmane teenindamine:
      1.1.1 Haldusteenuse usaldusväärsuse tõestamine (TLS-sertifikaat);



Joonis 5. Kogumisteenuse haldusteenuse komponendid

1.1.2 Kasutajate autentimine;

1.2 Valmisgenereeritud veebilehtede ja andmefailide serveerimine andmehoidlast.

1.3 Üldiste taustaandmete päringu vastuse varustamine sisseloginud kasutaja andmetega (WSGI).

1.4 Üleslaaditavate korralduste esmane valideerimine ja vahendamine haldusdeemonile ning haldusdeemoni sellekohaste vastuste vahendamine kliendile (WSGI).

2. **Haldusdeemon** on kasutajakonto `ivxv-admin` õigustes töötav ja kohalikul (`localhost`) liidesel kuulav veebiserver mille ülesanded on:

   2.1 Üleslaaditavate korralduste valideerimine;

   2.2 Üleslaaditavate korralduste vahetu rakendamine (kasutajate haldus);

   2.3 Üleslaaditavate korralduste salvestamine hilisemaks rakendamiseks (seadistuse ja valimisnimekirjade rakendamiseks teenusele);

   2.4 E-urni allalaadimise vahendamine.

3. **Agentdeemon** on kasutajakonto `ivxv-admin` õigustes töötav deemon, mille ülesanded on:

   3.1 Andmete kogumine ja registreerimine:

   3.1.1 Teadaolevate mikroteenuste seisund;

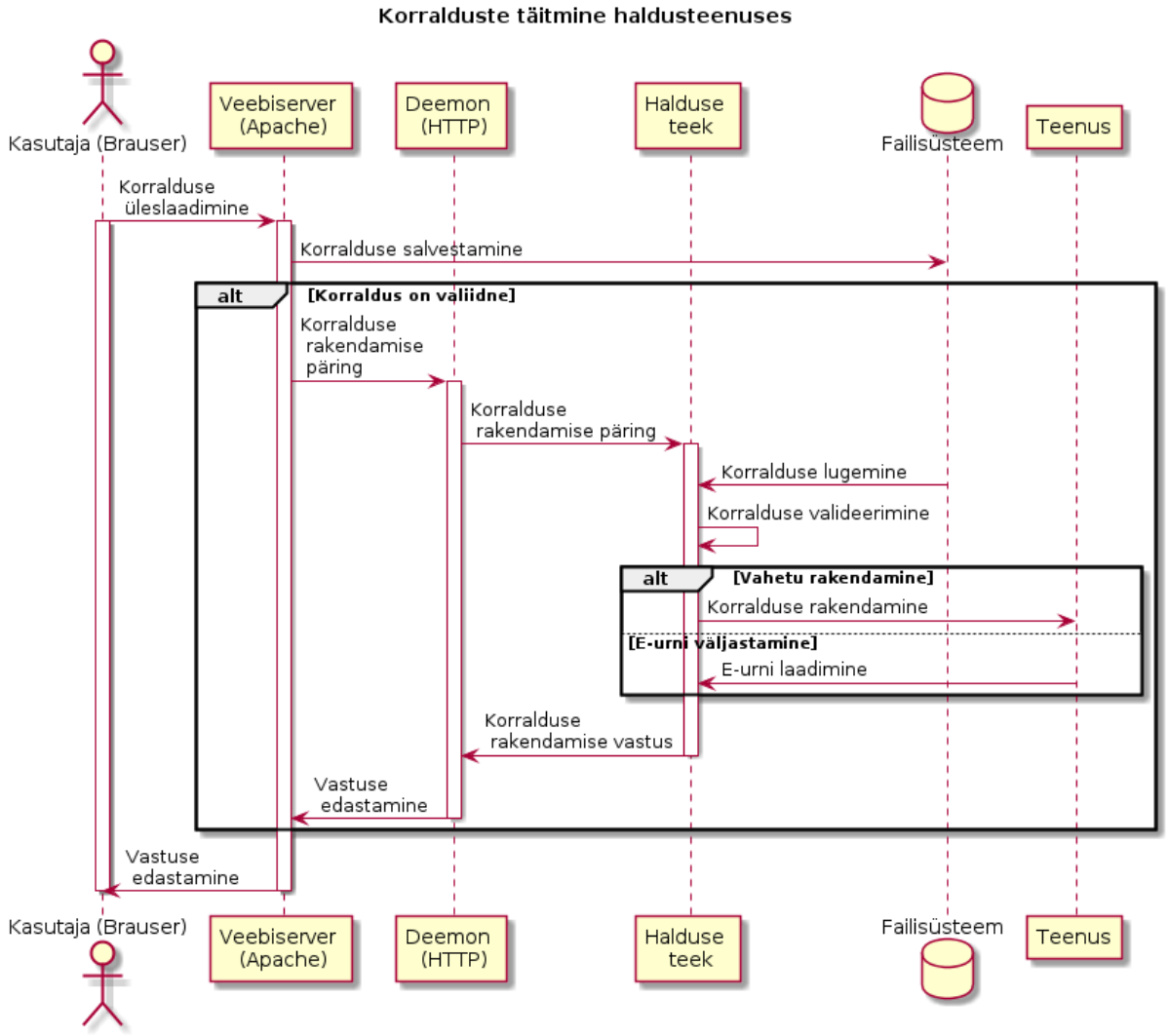
   3.1.2 Tegevusmonitooringu statistika allalaadimine;

4. **Andmehoidla** on failisüsteemis asuv kataloog, kuhu haldusteenuse komponendid hoiavad kogutud ja genereeritud andmeid (vaata üksikasjalist kirjeldust `IVXV kogumisteenuse haldusjuhendi` lisadest);

Välised komponendid, millega haldusteenus kokku puutub:

1. **Kogumisteenuse alamteenused** - paigaldamine, seadistamine ja seisundi andmete kogumine toimub agentdeemoni kaudu (SSH-ühendus teenuse masinasse);
2. **Seireserver** - üldstatistika andmete allalaadimine haldusteenuses kuvamiseks;

Märkus: Veebiserver (WSGI), Haldusdeemon ja Agentdeemon on teostatus Pythonkeeles ja kasutavad ühist **halduse teeki**



Joonis 6. Korralduste laadimine haldusteenusesse

2.5 Kogumisteenuse seisundid

Kogumisteenuse seisund kajastab teenuse kõigi alamteenuste seisundit, kasutusel-olevate väliste teenuste seisundit ja eelneva põhal tuletatud üldseisundit. Kogumisteenuse üldseisundi tuvastamisega tegeleb haldusteenus.

Üldseisundi olekud on:

1. **Paigaldamata** - seisund pärast haldusteenuse paigaldamist kuni kõigi alamteenuste paigaldamiseni;
2. **Paigaldatud** - kõik alamteenused on paigaldatud neile on rakendatud tehnilised seadistused ja teenuse toimimiseks vajalikud krüptovõtmed. Valimiste seadistust pole rakendatud (kuid võib olla laaditud haldusteenusesse);
3. **Seadistatud** - kogumisteenus on seadistatud ja töökorras, sellega on võimalik hääletust läbi viia ja e-urni väljastada.
4. **Osaline tõrge** - kogumisteenus on seadistatud ja osaliselt töökorras, mõned alamteenused pole töökorras, kuid see ei takista kogumisteenuse toimimist.
5. **Tõrge** - kogumisteenuse oluline sõlm pole töökorras, teenuse nõuetekohane osutamine pole võimalik.

Kogumisteenuse alamteenuste seisundid

Kogumisteenuse seisundi muutused

Kogumisteenuse seisund on jälgimav alates haldusteenuse edukast paigaldamisest, algne seisund on **Paigaldamata**.

Paigaldamata

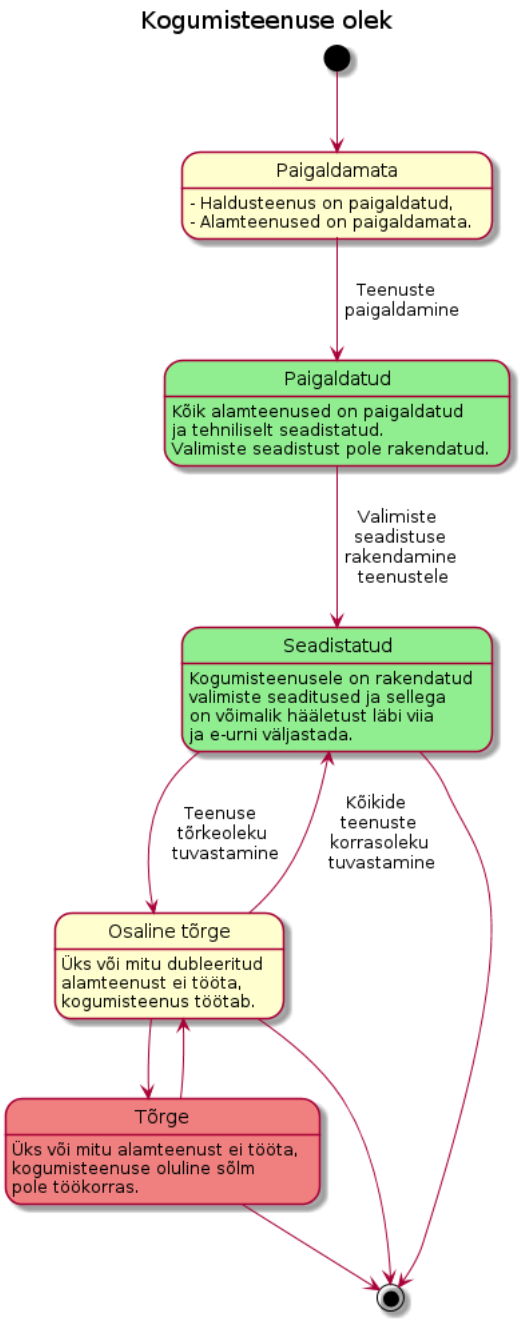
Toimub usaldusjuure ja tehnilise seadistuse rakendamine kogumisteenusele:

1. Seadistuste laadimine kogumisteenusesse;
2. Tehnilises seadistuses kirjeldatud alamteenuste paigaldus;
3. Usaldusjuure ja tehniliste seadistuste rakendamine alamteenustele;

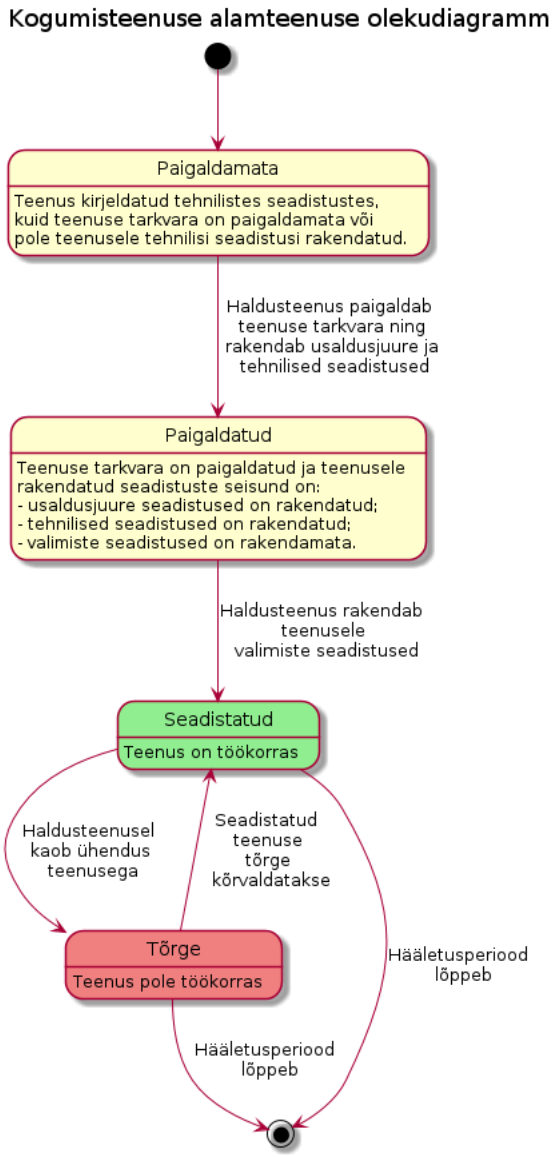
Seadistuste eduka rakendamise tulemusena saab süsteemi uueks seisundiks **Paigaldatud**.

Paigaldatud

Kogumisteenuse seadistused on rakendatud kõigile alamteenustele, valimiste seadistused pole rakendatud. Toimub valimiste seadistuse laadimine haldusteenusesse ja rakendamine alamteenustele.



Joonis 7. Kogumisteenuse olekudiagramm. Olekud vastavalt värvusele: kollane - seadistamisel, punane - viga, roheline - töökorras.



Joonis 8. Haldusteenuse poolt registreeritud alamteenuse olekudiagramm. Olekud vastavalt värvusele: kollane - seadistamisel, punane - viga, roheline - töökorras.

Valimiste seadistuse eduka rakendamise korral saab süsteemi uueks seisundiks **Seadistatud**.

Seadistatud

Kõik kogumisteenuse alamteenused on seadistatud ja töökorras. Haldusteenusel on kõikidest alamteenustest värsked seisundiraportid. Süsteemiga on võimalik hääletust läbi viia ja e-urni väljastada.

Kui süsteemis tuvastatakse tõrge, saab süsteemi uueks **Osaline tõrge**.

Seadistatud olekust ei pöörduta enam kunagi tagasi olekutesse **paigaldamata** või **paigaldatud**, kuigi uute alamteenuste lisamisel (kuni need on olekus **paigaldamata/paigaldatud**) oleks vastavad tingimused täidetud.

Osaline tõrge

Süsteem on seadistatud ja osaliselt töökorras, mõned süsteemi dubleeritud osad pole töökorras, kuid see ei takista süsteemil toimimast.

Rikke süvenemisel piirini, kus süsteem pole võimeline teenust osutama, saab süsteemi uueks olekuks **Tõrge**. Kõigi rikete kõrvaldamise järel saab süsteemi uueks olekuks **Seadistatud**.

Tõrge

Seadistatud süsteemil on tuvastatud rike, mis takistab teenuse osutamist.

Rikete kõrvaldamisel olukorrani, kus süsteemiga on võimalik teenust osutada, saab süsteemi uueks olekuks **Osaline tõrge**.

PEATÜKK 3

Rakendused

3.1 Üldpõhimõtted

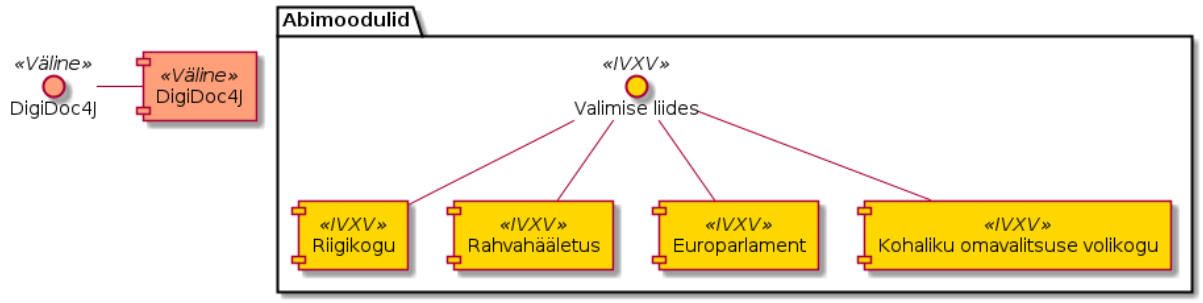
Kõik rakendused on käsurealiidesega rakendused, mis on pakendatud töötama operatsioonisüsteemi Windows 7 (või uuem) keskkonnas. Komponentide kasutajaliidesed on ühekeelsed. Komponendid tarnitakse eestikeelsetena, nende tõlkimine on võimalik tõlkefaili abil.

Rakendused programmeeritakse keeles Java.

Väliste infosüsteemidega suhtlevad rakendused kasutavad maksimaalselt olemasolevaid liideseid/andmestruktuure.

Rakendused saavad oma sisendi rakenduste seadistustest ja seadistustes näidatud failidest failisüsteemis ning salvestavad oma väljundi kasutaja näidatud kausta failisüsteemis. Failid võivad paikneda ka operatiiv-mälukettal.

Relevantsed rakendused toetavad ElGamal krüptosüsteemi täisarvujäägikorpustel ning P-384 elliptkõveral. Lugemistõend realiseeritakse Schnorri nullteadmustõestusel põhineval protokollil.



Joonis 9. Rakenduste abimoodulid

Rakenduste jaoks unifitseeritakse valimise liides, mis võimaldab erinevate valimistüüpide realiseerimist moodulitena. Digiallkirja verifitseerimise funktsionaalsus luuakse di-

gidoc4j teegi (https://github.com/open-eid/digidoc4j) abil. Abimoodulite kasutamist alljärgnevatel skeemidel eraldi välja ei tooda.

Rakenduste seadistamine

Rakendused seadistatakse kas digitaalallkirjastatud konfiguratsioonipakiga või käsureavõtmetega. Hierarhilise struktuuriga seadistuste sisestamist käsureavõtmed ei toeta. Seadistused konfiguratsioonipakis kirjeldatakse keeles YAML:

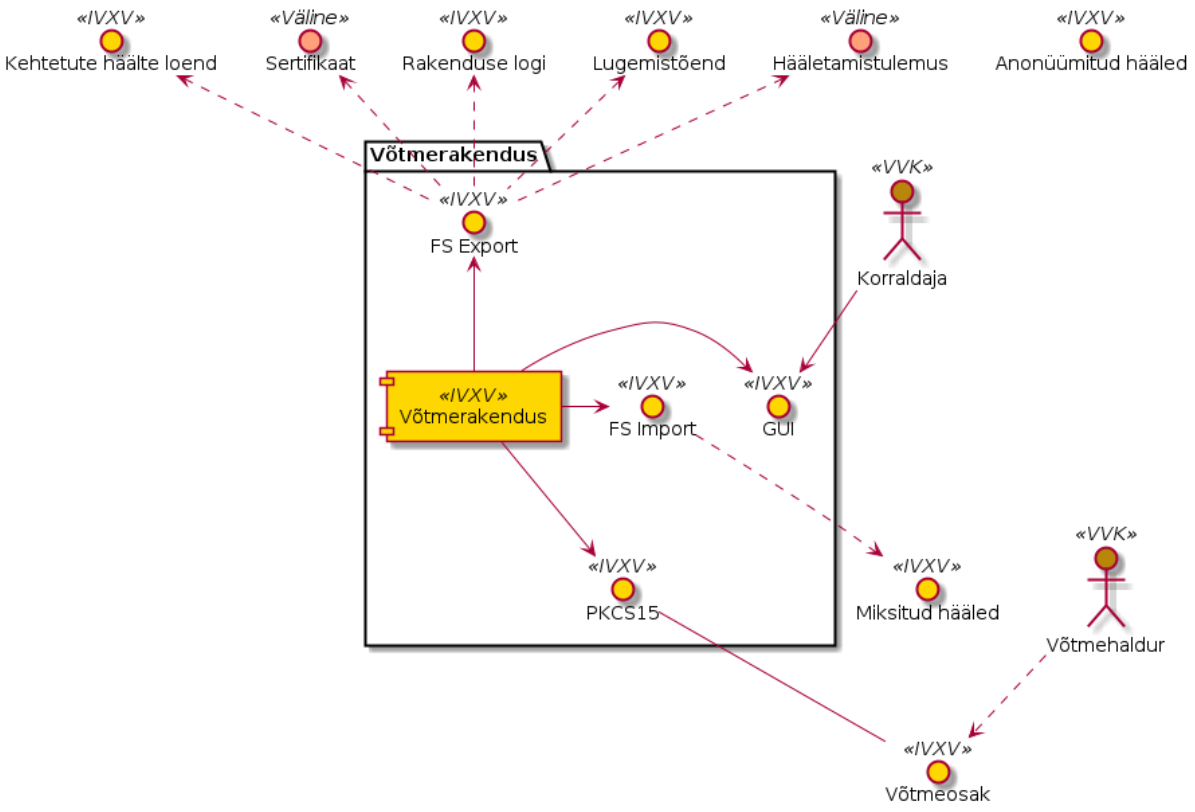
```
check:
  ballotbox: votes.zip
  ballotbox_checksum: votes.zip.sha256sum.bdoc
  districts: TESTKOV2017.districts.json
  registrationlist: register.zip
  registrationlist_checksum: register.zip.sha256sum.bdoc
  tskey: ts.pub.key
  vlkey: test.gen.pub.key
  voterlists:
    -
      path: 00.TESTKOV2017.gen.voters
      signature: 00.TESTKOV2017.gen.voters.signature
    -
      path: 03.TESTKOV2017.gen.voters
      signature: 03.TESTKOV2017.gen.voters.signature
    -
      path: 06.TESTKOV2017.gen.voters
      signature: 06.TESTKOV2017.gen.voters.signature
    -
      path: 09.TESTKOV2017.gen.voters
      signature: 09.TESTKOV2017.gen.voters.signature
  election_start: 2017-05-01T12:00:00+03:00
  out: out-1
squash:
  ballotbox: out-1/bb-1.json
  ballotbox_checksum: out-1/bb-1.json.sha256sum.bdoc
  districts: TESTKOV2017.districts.json
  out: out-2
revoke:
  ballotbox: out-2/bb-2.json
  ballotbox_checksum: out-2/bb-2.json.sha256sum.bdoc
  districts: TESTKOV2017.districts.json
  revocationlists:
    - 12.TESTKOV2017.gen.revoke.json
    - 13.TESTKOV2017.gen.revoke.json
    - 14.TESTKOV2017.gen.revoke.json
    - 15.TESTKOV2017.gen.revoke.json
  out: out-3
```

Sisendite kooskõlalisuse kontroll

Kõik rakendused teostavad konfiguratsioonile sisendite kooskõlalisuse kontrolli vastavalt nende poolt kasutatavale konfiguratsioonile:

1. sertifikaatide konfiguratsiooni laadimine;
2. konfiguratsiooni digiallkirja verifitseerimine;
3. ringkondade nimekirja verifitseerimine;
4. ringkondade nimekirja kooskõlalisuse kontroll;
5. ringkondade nimekirja laadimine;
6. valikute nimekirja verifitseerimine;
7. valikute nimekirja kooskõlalisuse kontroll;
8. valikute nimekirja laadimine;
9. valijate nimekirjade verifitseerimine;
10. valijate nimekirjade kooskõlalisus kontroll;
11. valijate nimekirjade laadimine;

3.2 Võtmerakendus



Joonis 10. Võtmerakenduse liidesed

Võtmerakendus on rakendus, millega genereeritakse iga hääletamise jaoks häälte salastamise ja häälte avamise võti ning mille abil toimub häälte lugemine ja tulemuse väljastamine.

Võtmerakendus kasutab *[DesmedtF89]* läviskeemi, mis baseerub usaldataval osakujagajal ning rakendab Shamiri osakujagamist, mis on informatsiooniteoreetiliselt turvaline t < M osapoole korral, kus M on lävipiir.

Võtmeosakud genereeritakse operatiivmälus ning talletatakse kiipkaardile PKCS15 liidese vahendusel.

Võtmerakenduse sisend võtme genereerimisel on

- Võtmepaari identifikaator
- Krüptosüsteemi ElGamal spetsifikatsioon – täisarvujäägikorpus või P-384 elliptkõver ning võtmepikkus
- M-N läviskeemi spetsifikatsioon, mis peab vastama reeglile N >= 2 * M - 1
- N PKCS15 ühilduvat kiipkaarti

Võtmerakenduse väljund võtme genereerimisel on

- Isesigneeritud sertifikaat
- N võtmeosakut talletatuna kiipkaartidel
- Rakenduse detailne tegevuslogi
- Rakenduse detailne vealogi

Võtmerakenduse sisend häälte lugemisel on

- Miksitud hääled
- Võtmepaari identifikaator
- M võtmeosakut vastavalt läviskeemi spetsifikatsioonile

Võtmerakenduse väljund häälte lugemisel on

- Signeeritud hääletamistulemus
- Kehtetute häälte loend
- Lugemistõend (Schnorri nullteadmustõestusel põhinev protokoll nagu viidatud hankedokumentides)
- Rakenduse detailne tegevuslogi
- Rakenduse detailne vealogi

Lisaks varem defindeeritud liidestele ja sõltuvustele kasutab töötlemisrakendus kolmanda osapoole teeki PKCS15 liidese realiseerimiseks. Konkreetne teek valitakse välja projekteerimisfaasis.

3.3 Töötlemisrakendus

Töötlemisrakendus on rakendus hääletamisperioodil kogutud häälte verifitseerimiseks, tühistamiseks ning anonüümimiseks, mis toimib vastavalt Üldkirjelduse jaotisele 7.6.

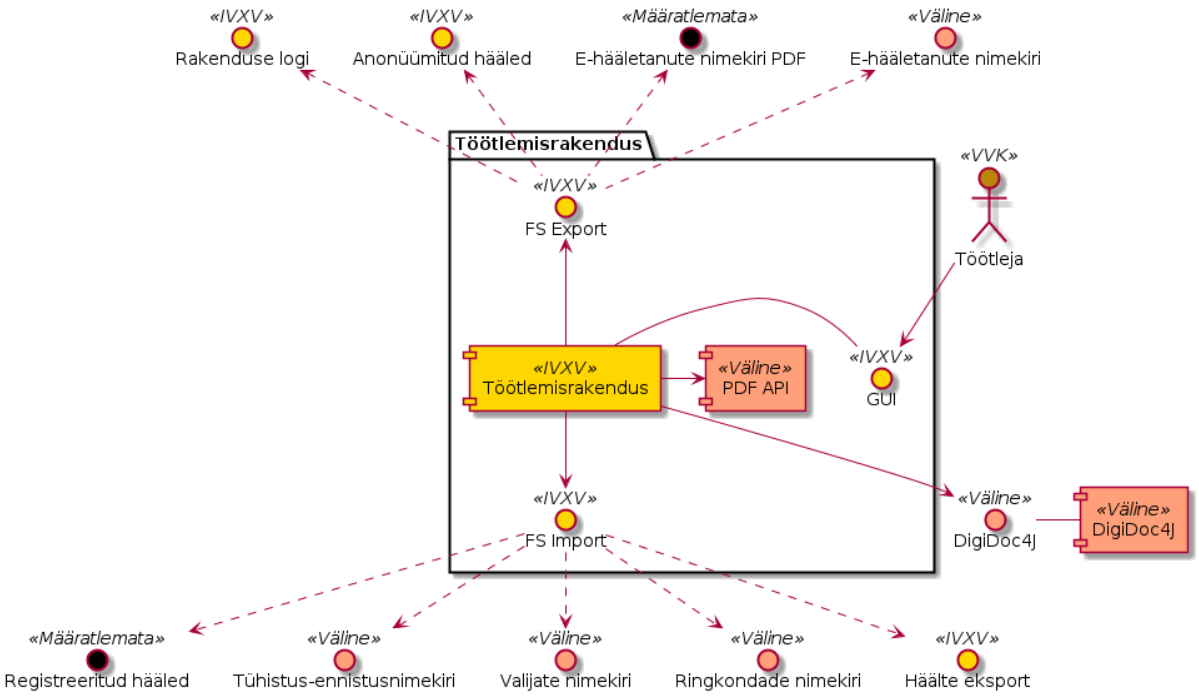
Töötlemisrakenduse sisendid on

- kogumisteenuse poolt talletatud elektroonilised hääled;
- registreerimisteenuse poolt väljastatud ajamärgendid;
- valijate nimekirjad;
- ringkondade nimekiri;
- tühistusnimekirjad;
- ennistusnimekirjad.

Töötlemisrakenduse väljundid on

- rakenduse detailne tegevuslogi;
- rakenduse detailne vealogi;
- e-hääletanute nimekiri PDF vormingus, vastavalt töötlemise etapile;
- e-hääletanute nimekiri masintöödeldaval kujul, vastavalt töötlemise etapile;
- anonüümitud hääled.

Lisaks varem defindeeritud liidestele ja sõltuvustele kasutab töötlemisrakendus kolmanda osapoole teeki PDFide väljastamise funktsionaalsuse realiseerimiseks.



Joonis 11. Töötlemisrakenduse liidesed

Elektrooniliste häälte täielik töötlemine

Elektrooniliste häälte täielik töötlemine on tegevus, mille käigus töötlemisrakendus võrdleb Kogumisteenuse poolt talletatud häältehulka registreerimisteenuse poolt talletatud häältehulgaga, kontrollib talletatud häälte vastavust valimiste konfiguratsioonile, tuvastab loendamisele minevad hääled ning anonüümistab need Võtmerakendusele üle andmiseks.

1. rakenduse seadistuste laadimine
2. elektrooniliste häälte digitaalallkirjade verifitseerimine;
3. registreerimisteenuse kinnituste verifitseerimine;
4. ajamärgendite verifitseerimine;
5. iga valija kohta viimase kehtiva hääle tuvastamine;
6. algse elektrooniliselt hääletanute nimekirja väljastamine PDF-vormingus;
7. tühistus- ja ennistusnimekirjade verifitseerimine;
8. tühistus- ja ennistusnimekirjade kooskõlalisuse kontroll;
9. tühistus- ja ennistusnimekirjade rakendamine;
10. miksimisele minevate häälte nimekirja koostamine, krüptogrammide eraldamine digitaalallkirjadest;
11. lõpliku elektrooniliselt hääletanute nimekirja väljastamine masinloetavas vormingus.

Elektrooniliselt hääletanute nimekirja genereerimine

1. rakenduse seadistuste laadimine
2. elektrooniliste häälte digitaalallkirjade verifitseerimine;
3. algse elektrooniliselt hääletanute nimekirja väljastamine PDF-vormingus;

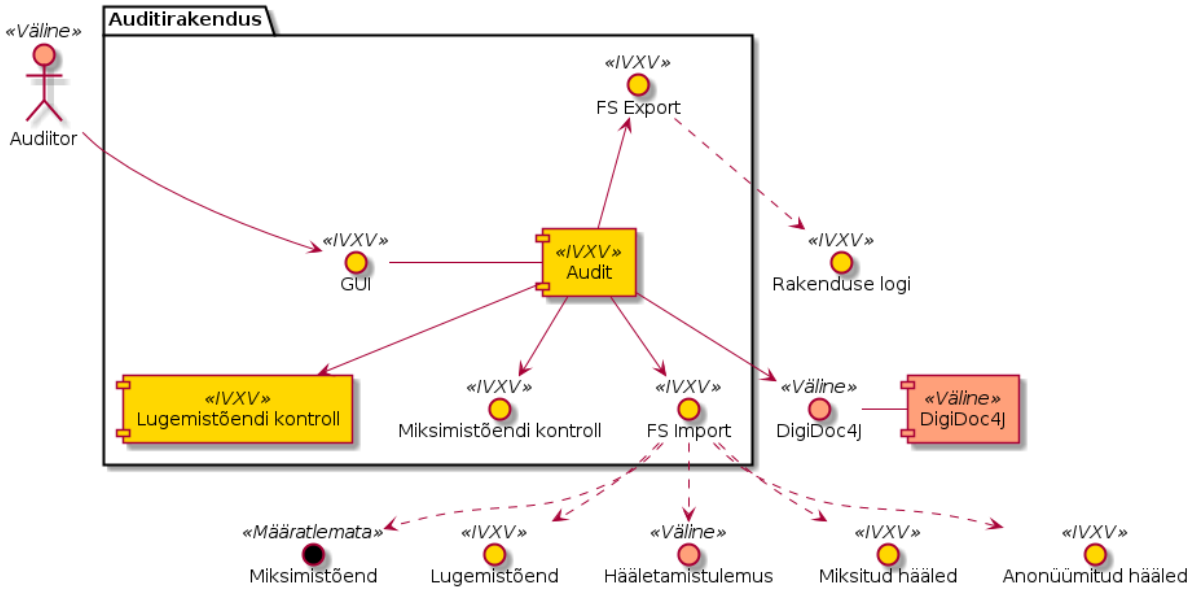
3.4 Auditirakendus

Auditirakendus (joonis 9) on rakendus, mis verifitseerib matemaatiliselt häälte kokkulugemise korrektsust ning miksimise kasutamisel ka miksimise korrektsust.

Auditirakenduse sisendid on

- anonüümitud hääled,
- miksitud hääled,
- miksimistõend (NB! Nii miksimistõend kui konkreetne miksimismeetod on defineerimata),
- hääletamistulemus.

Auditirakenduse väljund on rakenduse detailne tegevuslogi, mis sisaldab ka hinnangut auditi tervikliku õnnestumise kohta. Vajadusel väljastatakse ka rakenduse detailne vealogi.



Joonis 12. Auditirakenduse liidesed

PEATÜKK 4

Kasutatavad tehnoloogiad

4.1 Kogumisteenuse programmeerimiskeel

Kogumisteenuse tuumikfunktsionaalsus on programmeeritud keeles Go, mis vastab järgmistele hanke nõuetele:

- Staatiline tüüpimine;
- Automaatne mäluhaldus;
- Kompilaator avatud lähtekoodiga;
- Ribastamine (rööprapse).

Kogumisteenuse haldusteenus on programmeeritud keeles Python.

4.2 Rakenduste programmeerimiskeel

Rakendused on programmeeritud keeles Java, mis vastab hanke nõuetele keele laia leviku ja jätkusuutlikkuse kohta.

4.3 Projekti sõltuvused

Projektis kasutatavad kolmandate osapoolte komponendid koos nende motiveeritud kasutamisvajadusega on üles loetletud järgnevates tabelites. Eraldi tabelid on raamistiku pakendamiseks ja töötamiseks ning raamistiku arenduseks ja testimiseks.

Kõik IVXV projektis kasutatavad välised teegid asuvad `ivxv-external.git` hoidlas või on saadaval platvormil, kus rakendus tööle hakkab.

Kõik kogumisteenuses kasutatavad komponendid on avatud lähtekoodiga.

Tabel 1. IVXV raamistiku tööks kasutatavad kolmandate osapoolte komponendid

Nimi	Versioon	Litsents	Kasutusvajadus
Bootstrap	3.3.7	MIT	Kogumisteenuse haldusteenuse kasutajaliidese kujundus
Bouncy Castle	1.55	MIT	ASN1 käsitlemine, andmetüübi BigInteger abifunktsioonid
Bottle	0.12.7	MIT	Raamistik kogumisteenuse haldusteenuse veebiliidese teostamiseks
CAL10N	0.7.7	MIT	Mitmekeelsuse tugi, tõlkefailide valideerimine
Digidoc 4j	1.0.6	LGPL	BDoc konteinerite käsitlemine
Digidoc 4j DSS	4.7.RC2.d4j.3	LGPL	Digidoc 4j sõltuvus
Apache Commons (cli 1.4, codec 1.10, collections 3.2.2, io 2.5, lang 2.6, logging 1.2, compress 1.3)	•	Apache License v2.0	Digidoc 4j ja PDFBox sõltuvused
Apache HttpComponents	4.5.3	Apache License v2.0	Digidoc 4j sõltuvus
Apache Santuario	2.0.8	Apache License v2.0	Digidoc 4j sõltuvus
JDigiDoc	3.12.1	LGPL	Digidoc 4j sõltuvus
StaX	1.0-2	Apache License v2.0	Digidoc 4j sõltuvus
log4j	1.2.17	Apache License 2.0	Digidoc 4j sõltuvus
Woodstox	4.4.1	Apache License 2.0	Digidoc 4j sõltuvus
Xalan-Java	2.7.2	Apache License 2.0	Digidoc 4j sõltuvus
Xml Apis	1.3.04	Apache License 2.0	Digidoc 4j sõltuvus
Docopt	0.6.1	MIT	Kogumisteenuse haldusutiliitide käsurealiidese teostus
etcd	3.1.0	Apache License v2.0	Talletusteenusena kasutatav hajus võtiväärtus andmebaas
github.com/ghodss/yaml	73d445a	MIT	etcd klientteegi sõltuvus
gopkg.in/yaml.v2	4c78c97	Apache License v2.0	github.com/ghodss/yaml sõltuvus
github.com/golang/protobuf	224aaba	BSD 2.0	etcd klientteegi sõltuvus
github.com/grpc-ecosystem/go-grpc-prometheus	6b7015e	Apache License v2.0	etcd klientteegi sõltuvus
github.com/grpc-ecosystem/grpc-gateway	6863684	BSD 2.0	etcd klientteegi sõltuvus

Jätkub järgmisel lehel

Tabel 1 – jätk eelmisele leheküljele

Nimi	Versioon	Litsents	Kasutusvajadus
google.golang.org/grpc	1.0.4	Apache License v2.0	etcd klientteegi sõltuvus
golang.org/x/net	f249948	BSD 2.0	etcd klientteegi sõltuvus
Prometheuse klientteek	0.8	Apache License v2.0	etcd klientteegi sõltuvus
github.com/beorn7/perks/quantile	4c0e845	MIT	Prometheuse klientteegi sõltuvus
github.com/matttproud/golang_protobuf_extensions	1.0.0	Apache License v2.0	Prometheuse klientteegi sõltuvus
Gradle	3.0	Apache License v2.0	Java rakenduste ehitamise raamistik
HAProxy	1.6.3	GPL v2	Vahendusteenusena kasutatav TCP proksi
IvyPot	0.4	Apache License v2.0	Gradle ehitusraamistiku laiendus sõltuvuste haldamiseks ja rakenduste ehitamiseks vallasrežiimis
Jackson	2.8.9	Apache License v2.0	JSON vormingus failide lugemine ja kirjutamine
jQuery	3.1.0	MIT	Kogumisteenuse haldusteenuse kasutajaliides
Logback	1.2.3	Eclipse Public License v1.0 või LGPL v2.1	Logimise API SLF4J realisatsioon
Logback JSON	0.1.5	Eclipse Public License v1.0 või LGPL v2.1	Logback logija laiendus JSON vormingus logikirjete koostamiseks Jackson teegi abil
metisMenu	1.1.3	MIT	Kogumisteenuse haldusteenuse kasutajaliides
PDFBox	2.0.6	Apache License v2.0	PDF vormingus raportite genereerimise tugi Java rakendustele

Jätkub järgmisel lehel

Tabel 1 – jätk eelmisele leheküljele

Nimi	Versioon	Litsents	Kasutusvajadus
PyYAML	3.11	MIT	Kogumisteenuse seadistusfailide töötlemise tugi haldusteenusele
Schematics	2.0.1	BSD	Kogumisteenuse seadistusfailide valideerimise tugi haldusteenusele
SLF4J	1.7.25	MIT	Standardne logimise API
SnakeYAML	1.18	Apache License v2.0	YAML vormingus andmete lugemine
SB Admin 2	3.3.7+1	MIT	Kogumisteenuse haldusteenuse kasutajaliidese kujundus

Tabel 2. IVXV raamistiku testide kasutatavad kolmandate osapoolte komponendid

Nimi	Versioon	Litsents	Kasutusvajadus
Hamcrest	1.3	BSD	Loetavam assert-meetodite kasutamine Java ühiktestides
JUnit	4.12	Eclipse Public License v1.0	Java testimisraamistik
JUnitParams	1.1.0	Apache License v2.0	Testide parametriseerimise tugi
Mockito	2.+	MIT	Testitava koodi sõltuvuste mockimise tugi
Byte Buddy	1.6.14	Apache License v2.0	Mockito sõltuvus
Objenesis	2.5	Apache License v2.0	Mockito sõltuvus
libdigidocpp	3.13.0	LGPL	Testandmete genereerimine

Tabel 3. IVXV raamistiku arendamiseks ja/või testimiseks kasutatavad kolmandate osapoolte tööriistad

Nimi	Versioon	Litsents	Kasutusvajadus
Behave	1.2.5	BSD	Regressioonitestide käivitaja (*Behavior-driven development*)
Docker	1.13 (või uuem)	Apache License 2.0	Regressioonitestide läbiviimise keskkond - tarkvarakonteinerid
Docker Compose	1.10.0	Apache License 2.0	Regressioonitestide läbiviimise keskkond - tarkvarakonteinerite haldus

PEATÜKK 5

Lisad

5.1 Kogumisteenuse ehitamine paigatud Go standardteegiga

Eestis on ringluses mitmeid ID-kaarte ja Digi-ID-sid, mille sertifikaadid sisaldavad valesti kodeeritud RSA avalikku võtit. Go standardteek keeldub sellised vigaseid sertifikaate vastu võtmast. Samas on nende arv liiga suur, et selle vastu mitte midagi ette võtta.

Lahenduseks tuleb IVXV kogumisteenuse alamteenused kompileerida kasutades paigatud Go standardteeki. Tarnega on kaasas `ivxv-golang` pakk, mis sisaldab paikasid sellist laadi vigaste sertifikaatide lubamiseks ja vahendeid nende rakendamiseks.

Paigatud standardteegi ehitamine peaks toimuma samas keskkonnas, kus ka IVXV kogumisteenus ehitatakse, st Ubuntu 16.04.

Esimese asjana tuleb paigaldada kõik `ivxv-golang` kaustas olevas `README.rst` failis loetletud sõltuvused. Seejärel `make` käsu andmisel laetakse Ubuntu hoidlatest alla kõige uuem Go 1.7 lähtekood, paigatakse, ehitatakse ning testitakse vigaste sertifikaatide kasutust. Õnnestumise korral on `source/` alamkaustas muuhulgas kaks vajalikku .deb pakki:

- `golang-1.7-src_1.7.1-2ubuntu1_amd64.deb` ja
- `golang-1.7-go_1.7.1-2ubuntu1_amd64.deb`.

Need tuleb paigaldada IVXV kogumisteenust ehitavasse arvutisse enne IVXV kompileerimist: siis kasutatakse valmendamise käigus paigatud Go standardteeki.

PEATÜKK 6

Viited

Kirjandus

[DesmedtF89] Desmedt, Y. & Frankel, Y. Brassard, G. (Ed.) Threshold Cryptosystems Advances in Cryptology - CRYPTO '89, 9th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20-24, 1989, Proceedings, Springer, 1989, 435, 307-315.

[HMVW16] Sven Heiberg, Tarvi Martens, Priit Vinkel, Jan Willemson, Improving the verifiability of the Estonian Internet Voting scheme. In Robert Krimmer, Melanie Volkamer, Jordi Barrat, Josh Benaloh, Nicole Goodman, Peter Y.A. Ryan, Oliver Spycher, Vanessa Teague, Gregor Wenda (Eds.), The International Conference on Electronic Voting E-Vote-ID 2016, 18-21 October 2016, Lochau/Bregenz, Austria, TUT Press, pp. 213-229, ISBN 978-9949-83-022-0

[ProVerif] ProVerif: Cryptographic protocol verifier in the formal model, http://prosecco.gforge.inria.fr/personal/bblanche/proverif/

[TK2016] Tehniline kirjeldus. arenduse hange, Vabariigi Valimiskomisjon, 2016

[ÜK2016] Elektroonilise hääletamise üldraamistik ja selle kasutamine Eesti riiklikel valimistel. Elektroonilise Hääletamise Komisjon, Tallinn 2016