

IVXV arhitektuur

Arhitektuuridokument

Versioon 1.8.0

01.12.2022

39 lk

Dok IVXV-AR-1.8.0

Sisukord

Sisukord	2
1 Sissejuhatus	4
1.1 IVXV kontseptsioon	4
1.2 IVXV krüptograafiline protokoll	5
1.3 Notatsioon	5
2 Kogumisteenus	7
2.1 Mikroteenused	8
Vahendusteenuse funktsioon ja tehniline liides	8
Nimekirjateenuse funktsioon ja tehniline liides	10
Kontrollteenuse funktsioon ja tehniline liides	11
Hääletamisteenuse funktsioon ja tehniline liides	11
Talletamisteenuse funktsioon ja tehniline liides	11
Tuvastusteenuse funktsioon ja tehniline liides	12
Allkirjateenuse funktsioon ja tehniline liides	12
Kogumisteenuse mikroteenuste evitamine	12
2.2 Välised teenused ja laiendatavus	13
Registreerimisteenuse funktsioon	14
Kogumisteenuse laiendusmoodulite lisamine	14
2.3 Monitooring	15
Logimine	16
Üldstatistika	16
Detailstatistika	17
2.4 Haldus	17
Haldusteenuse komponendid	18
2.5 Kogumisteenuse seisundid	20
Kogumisteenuse alamteenuste seisundid	20
Kogumisteenuse seisundi muutused	20
2.6 Valijate nimekirjade olekud haldusteenuses	24
3 Rakendused	26
3.1 Üldpõhimõtted	26
Rakenduste seadistamine	27
Sisendite kooskõlalise kontroll	28
3.2 Võtmerakendus	28
3.3 Töötlemisrakendus	30
Elektrooniliste häälte täielik töötlemine	31
Elektrooniliselt hääletanute nimekirja genereerimine	31
3.4 Auditirakendus	31
4 Kasutatavad tehnoloogiad	33
4.1 Kogumisteenuse programmeerimiskeel	33
4.2 Rakenduste programmeerimiskeel	33

4.3 Projekti sõltuvused	34
5 Viited	38
Kirjandus	39

PEATÜKK 1

Sissejuhatus

Elektroonilise hääletamise infosüsteem IVXV on loodud lähtuvalt e-hääletamise raamistikust [ÜK2016] ja riigihanke 171780 tehnilisest kirjeldusest [TK2016]. Käesolevas dokumendis kirjeldatakse IVXV arhitektuurset lahendust. Elektroonilise hääletamise infosüsteem koosneb vallasrežiimirakendustest ning sidusrežiimikomponentidest. Täiendavalt sõltub infosüsteem välistest infosüsteemidest ning mõjutab vahetult elektrooniliseks hääletamiseks ja hääle kontrollimiseks kasutatavaid komponente.

Arhitektuuridokument kirjeldab IVXV komponente, nende omavahelisi liideseid ja liideseid 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 teostab „ümbrikuskeemil“ põhinevat e-hääletamise protokollil. IVXV toimib hääletamiseelsel etapil, hääletamisetapil, töötlusetapil ning lugemisetapil ja pakkub 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

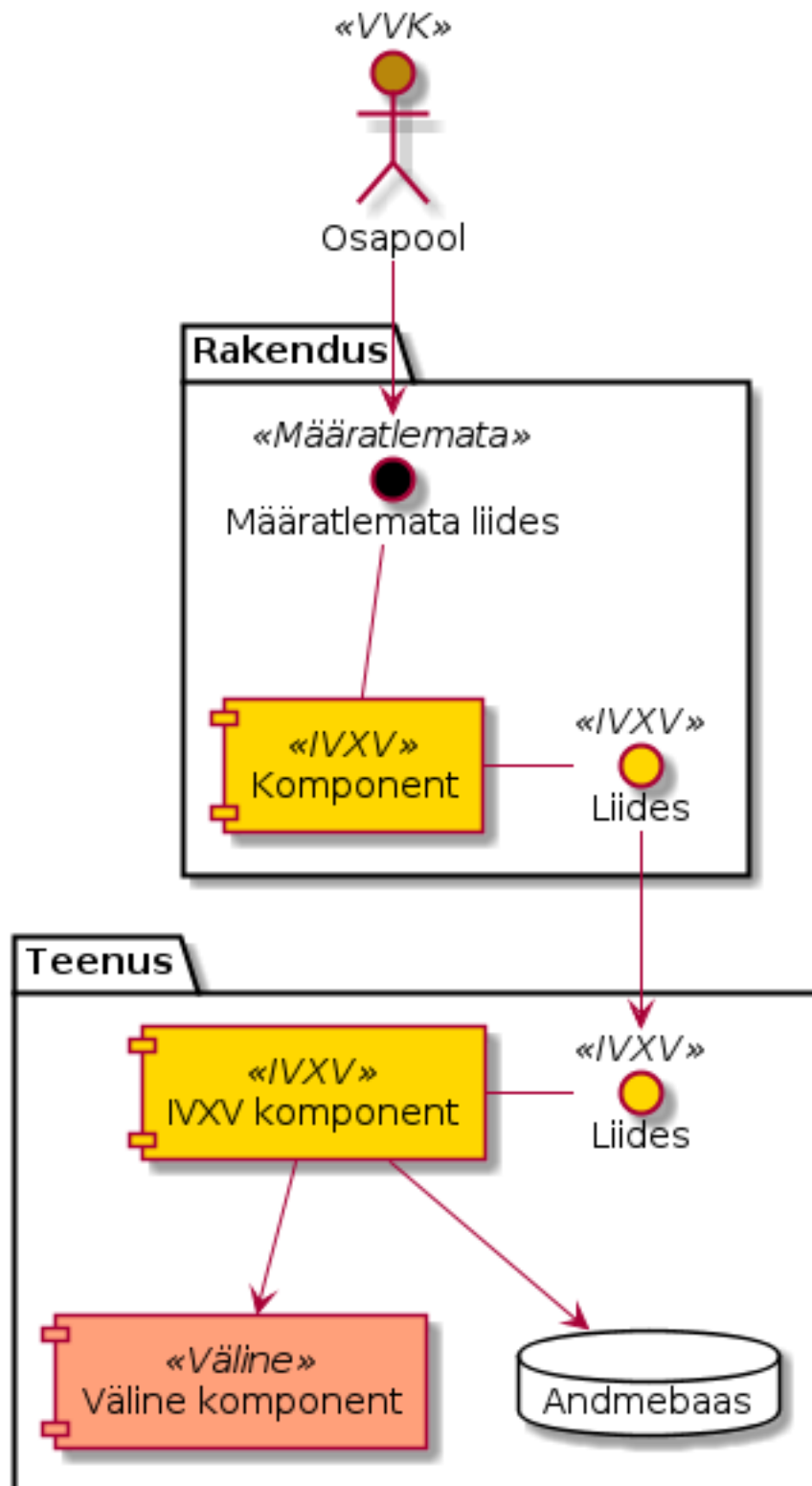
Elektroonilise hääletamise turvalisuse, verifitseeritavuse ning hääletamise salajasuse, hääletamise korrektsuse ja hääletaja sõltumatuse saavutamiseks on rangelt kirjeldatud elektroonilise hääletamise krüptograafiline protokoll [HMVW16]. Protokoll annab vajaliku ja piisava ülevaate IVXV ülesehitusest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

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

1.3 Notatsioon

Arhitektuurse lahenduse visandi illustreerimiseks kasutatakse dokumendis UML-skeeme, kus eristame värvide ja märgenditega <<>> kodeeritud 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 omnikuks on VVK.
- Märgend <<Määratlemata>> (Must) – infosüsteemi jaoks oluline liides on määratlemata.



Joonis 1.1. 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-valimiskasti. Kogumisteenus kasutab väliseid teenuseid (tuvastamine, allkirjastamine, registreerimine). Kogumisteenusel on peale Koguja enda teisi haldureid (Korraldaja, Klienditugi), kelle jaoks on Kogumisteenusel eraldi haldusliidesed.

Kogumisteenus töötab sidusrežiimis ning vähemalt valija- ja kontrollrakenduse suunatud liidesed on avatud internetile. Seega töötleb Kogumisteenus potentsiaalselt ebasaldusväärsest allikast pärit päringuid. Tulenevalt tarkvarale seatavast turvasemest, 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)¹. Keelel Go on:

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

Kogumisteenuse andmeedastuseks kasutatakse üldjuhul JSON-vormingut, välja arvatud olukordades, kus välised asjaolud tingivad mõne muu andmevormingu kasutamist

¹ <https://golang.org>

(näiteks BDOC-vorming põhineb 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 20.04 LTS (Focal Fossa) operatsioonisüsteemil 64-bitisel arhitektuuril.

Andmesäilitus on teostatud 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 haldurite 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.

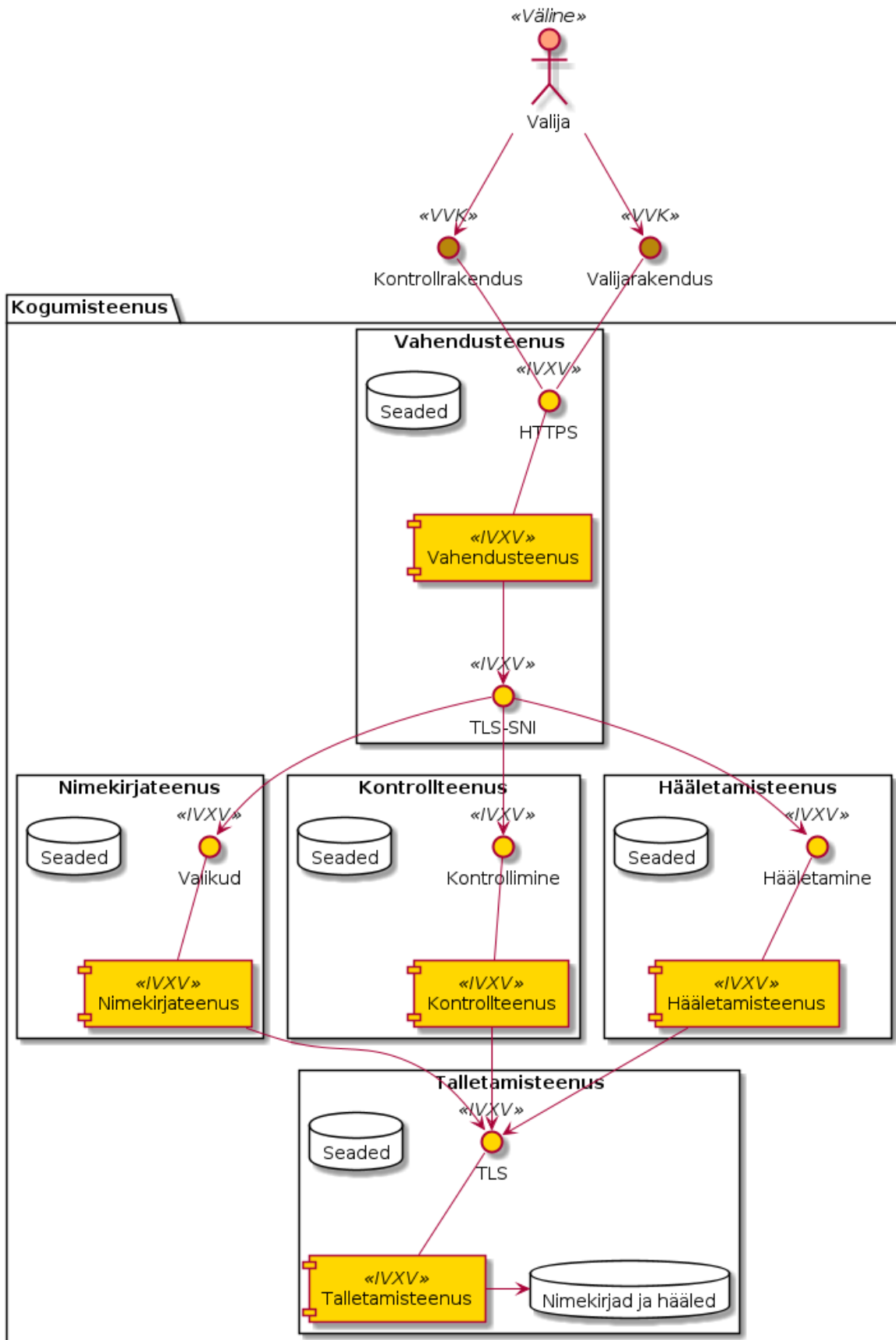
Teenused kasutavad transpordiprotokollina TLS-i, kõik ühendused on mõlemapoolselt autenditud. Rakenduskihi protokoll on JSON-RPC.

Kõik teenused tekitavad tegevuslogi, mida säilitatakse nii lokaalselt kui logitakse syslog protokolliga vahendusel kesksesse logikogujasse.

Vahendusteenuse funktsioon ja tehniline liides

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

Vahendusteenus ei termineeri TLS-ühendust vaid kasutab sihtpunkti tuvastamiseks TLS-i *Server Name Indication* (SNI) laiendust. 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



Joonis 2.2. Kogumisteenuse jaotus mikroteenusteks

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.

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 lähteadressi 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 vahendamise ülesannet.

HAProxy lähtekood on avalik ja sobiva litsentsiga ning pakendatud kogumisteenuse alusplatvormi ametlikus hoidlas (vt. [Kasutatavad tehnoloogiad](#)).

Nimekirjateenuse funktsioon ja tehniline liides

Nimekirjateenuse põhifunktsioon 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 põhifunktsioon on kontrollpäringute töötlemine ning kontrollitava hääle väljastamine Talletamisteenusest Kontrollrakendusse.

Kontrollteenus on olekuvaba komponent, mida on võimalik horisontaalselt skaleerida.

Hääletamisteenuse funktsioon ja tehniline liides

Hääletamisteenuse põhifunktsioon 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 põhifunktsioon on valikute ja valijanimekirjade ning häälte pikaajaline talletamine.

Talletamisteenuse horisontaalseks skaleerimiseks kasutatakse hajustalletamist võimaldavat säilitustehnoloogiat.

Talletamisteenuse teostus

Talletamisteenus ei ole teadlik IVXV protokollist ega talletatavate andmete spetsiifikaad, 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 ilma suurema vaevata kasutada Talletamisteenusena ükskõik millist üldlevinud võti-väärtus andmebaasi: 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, mistõttu 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](https://coreos.com/etcd/docs/latest/op-guide/hardware.html)².

² <https://coreos.com/etcd/docs/latest/op-guide/hardware.html>

Tuvastusteenuse funktsioon ja tehniline liides

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

Allkirjateenuse funktsioon ja tehniline liides

Allkirjateenuse funktsioon 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.

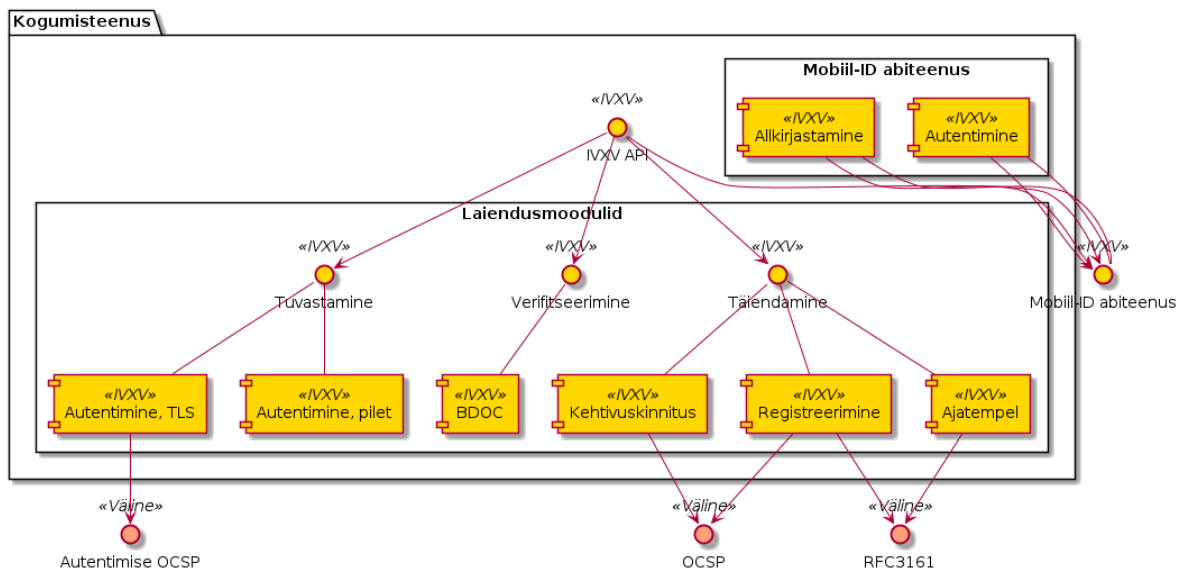
Kogumisteenuse mikroteenuste evitamine

Kogumisteenuse mikroteenused sõltuvad välistest pakkidest minimaalselt. Vajalikud sõltuvused on:

1. SSH-server haldustegevuste läbiviimiseks (seda kasutab mikroteenuste haldamiseks haldusteenus).
2. rsyslog logide kogumiseks logikogumisteenustesse.

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

2.2 Välised teenused ja laiendatavus



Joonis 2.3. 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 teostada ka täiendavaid moduleid. Hetkel on teostatud 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 hääle pikaajalisel talletamisel.

Registreerimisteenuse funktsioon

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

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 praegu kaks erinevat teostust:

1. 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;
2. RFC 3161 liides, mille korral ebastandardse lahendusena pannakse ajatempli-päringu nonsiks Hääletamisteenuse poolt allkirjastatud hääle räsi.

Kogumisteenuse laiendusmoodulite lisamine

Kogumisteenuse API defineerib kuute tüüpi laiendusmooduleid:

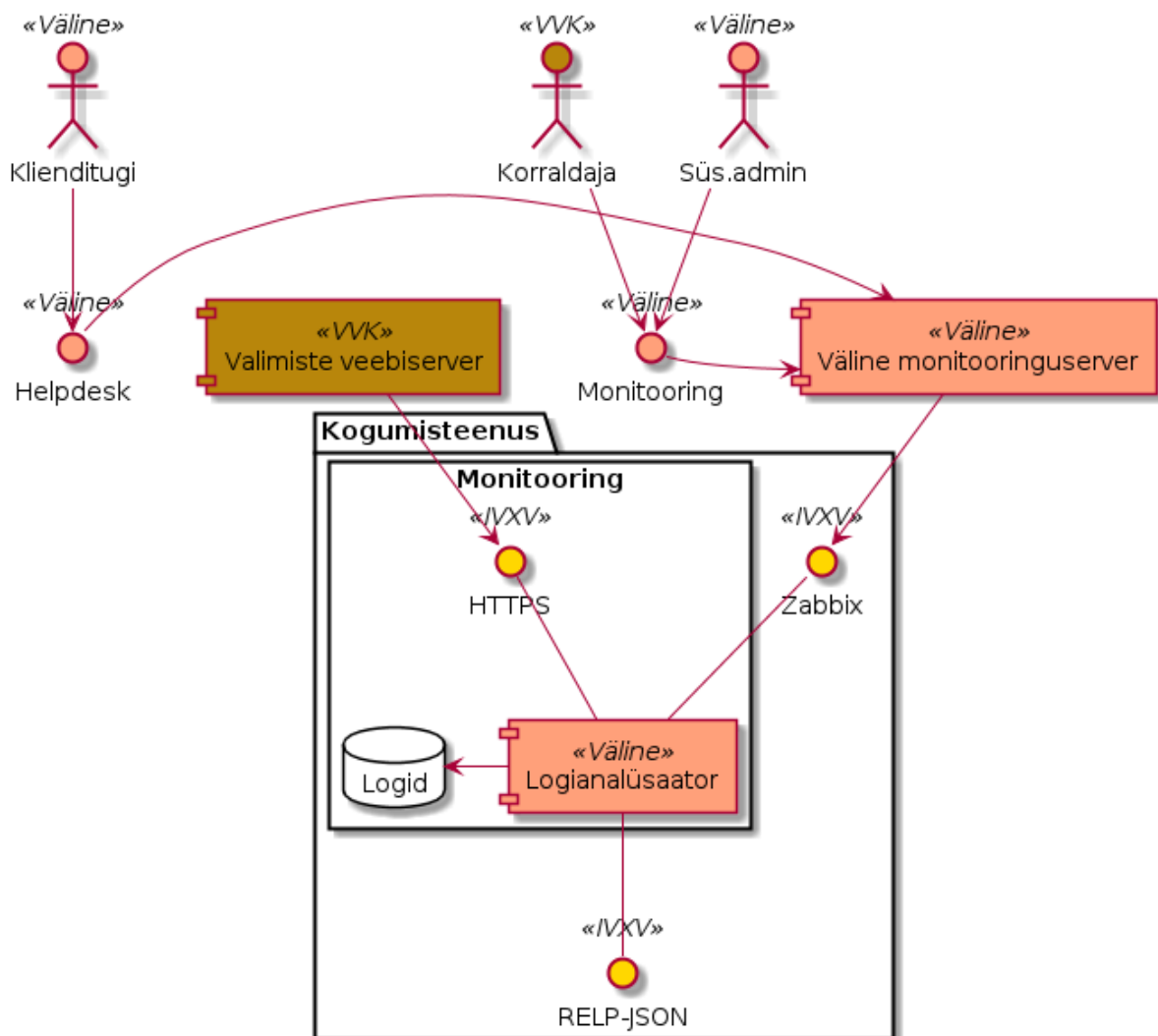
1. isikutuvastus (Go pakk `ivxv.ee/auth`, näiteks `tls`);
2. tuvastatud isiku sertifikaadist valija identifikaatori tuletamine (Go pakk `ivxv.ee/identity`, näiteks `serialnumber`);
3. valija identifikaatorist vanuse tuletamine (Go pakk `ivxv.ee/age`, näiteks `estpic`);
4. allkirjastatud konteineri verifitseerimine (Go pakk `ivxv.ee/container`, näiteks `bdoc`);
5. allkirja kvalifitseerimine (Go pakk `ivxv.ee/q11n`, näiteks `tspreg`);
6. andmetalletusprotokoll (Go pakk `ivxv.ee/storage`, näiteks `etcd`).

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

Uue mooduli kasutamiseks tuleb selle identifikaator lisada seadistusse vastava moodulitüübi seadistuse juurde koos alammoduli 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



Joonis 2.4. Monitooringulahendus

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 klient-rakendus 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) ja sellele rakendatakse pikkuse piirangut (piirang terve logiteate ja samuti parameetri kaupa);
- Süsteemiperimeetrist väljastpoolt pärinevat infot logitakse ainult saneerituna ja 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, vanusegruppide, operatsioonisüsteemide ning autentimisvahendite kaupa;
- edukalt kontrollitud hääle/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-/veatüüpipõhist analüüsi.

Detailstatistika on kättesaadav üle HTTPS-liidese.

2.4 Haldus

Kogumisteenuse haldamine toimub digitaalallkirjastatud seadistuspakkide abil.

Kogumisteenus pakub seadistuspakkide laadimiseks kahte liidest:

- Käsurealiides – rakendus verifitseerib allkirja, valideerib korralduste vormingut, kooskõllalisust 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:

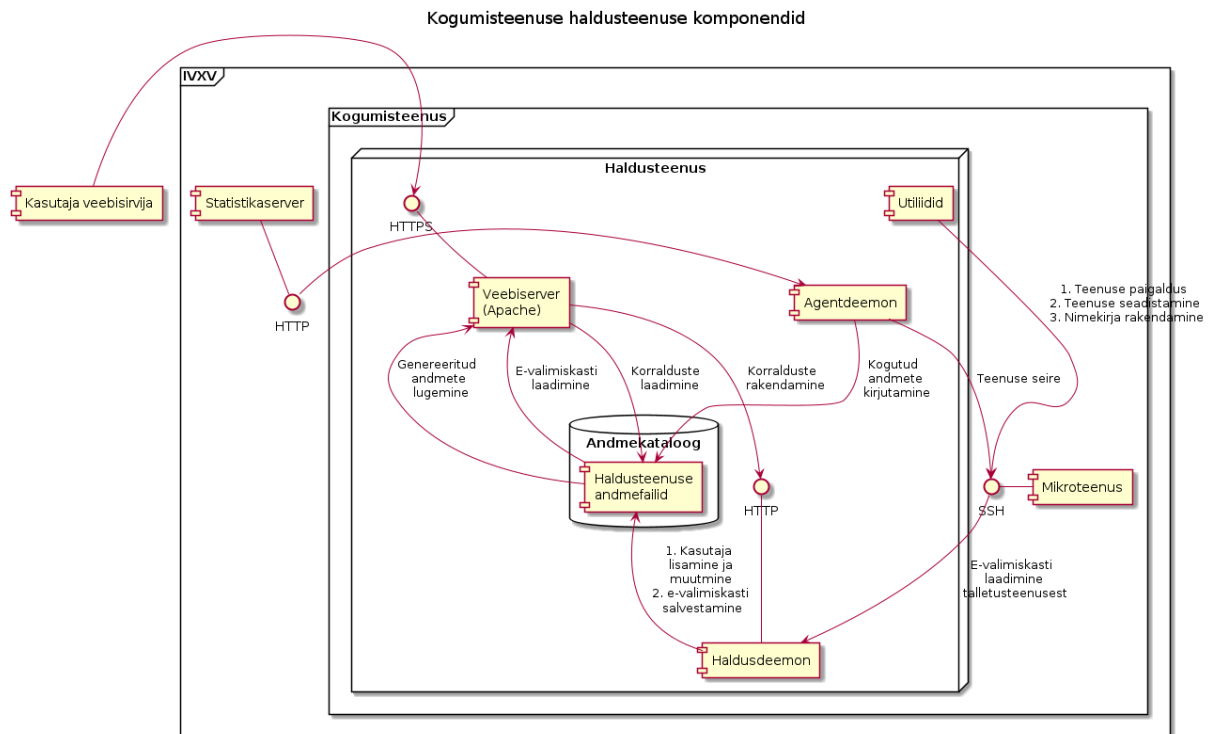
- Kogumisteenuse mikroteenuste seisundi jälgimine;
- Valimiste nimekirjade haldus;
- Statistika kuvamine e-hääletamise kulgemise kohta;
- Haldusteenuse kasutajate haldus;
- Kogumisteenuse halduse logi kuvamine.

Kõik rakendusele antud korraldused säilitatakse - ka need mida ei rakendatud. Viga-seid (mittevalideeruvaid) korraldusi ei säilitata.

Kogumisteenuse haldusteenus sooritab järgmisi tegevusi automaatselt:

1. Valijate nimekirjade muudatuste laadimine Valimiste Infosüsteemist;
2. Hääletamise statistika kogumine hääletusteenusest ja eksportimine Valimiste Infosüsteemi;
3. Talletatud häälte, logide ning seadistuste varundamine varundusteenusesse.

Haldusteenuse komponendid



Joonis 2.5. Kogumisteenuse haldusteenuse komponendid

1. **Halduse veebiserver** on süsteemse kasutaja `www-data` õigustes töötav Apache server, mille ülesanded on:
 1. Kasutajatelt tulevate HTTPS-päringute esmane teenindamine:
 1. Haldusteenuse usaldusväärseuse tõestamine (TLS-sertifikaat);
 2. Kasutajate autentimine;
 2. Valmisgenereeritud veebilehtede ja andmefailide serveerimine andmehoidlast.
 3. Üldiste taustaandmete päringu vastuse varustamine sisseloginud kasutaja andmetega (WSGI).
 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:
 1. Üleslaaditavate korralduste valideerimine;
 2. Üleslaaditavate korralduste vahetu rakendamine (kasutajate haldus);
 3. Üleslaaditavate korralduste salvestamine hilisemaks rakendamiseks (seadistuse ja valimisnimekirjade rakendamiseks teenusele);
 4. E-valimiskasti allalaadimise vahendamine.
3. **Agentdeemon** on kasutajakonto `ivxv-admin` õigustes töötav deemon, mille ülesanded on:

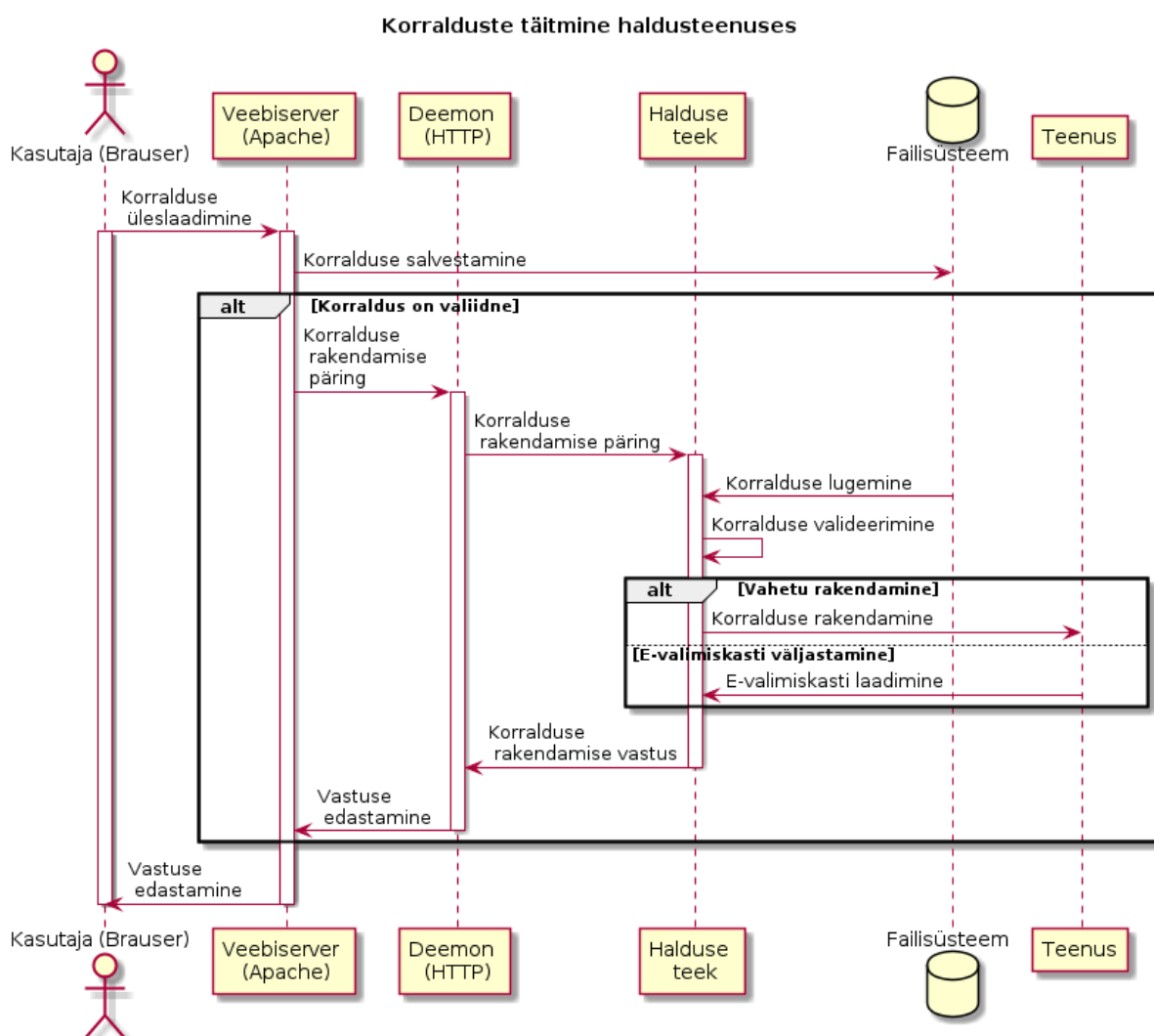
1. Andmete kogumine ja registreerimine:

1. Teadaolevate mikroteenuste seisund;
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;



Joonis 2.6. 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** - alates haldusteenuse paigaldamisest 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 see võib olla laaditud haldusteenusesse);
3. **Seadistatud** - kogumisteenus on seadistatud ja töökorras, sellega on võimalik hääletust läbi viia ja e-valimiskasti 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

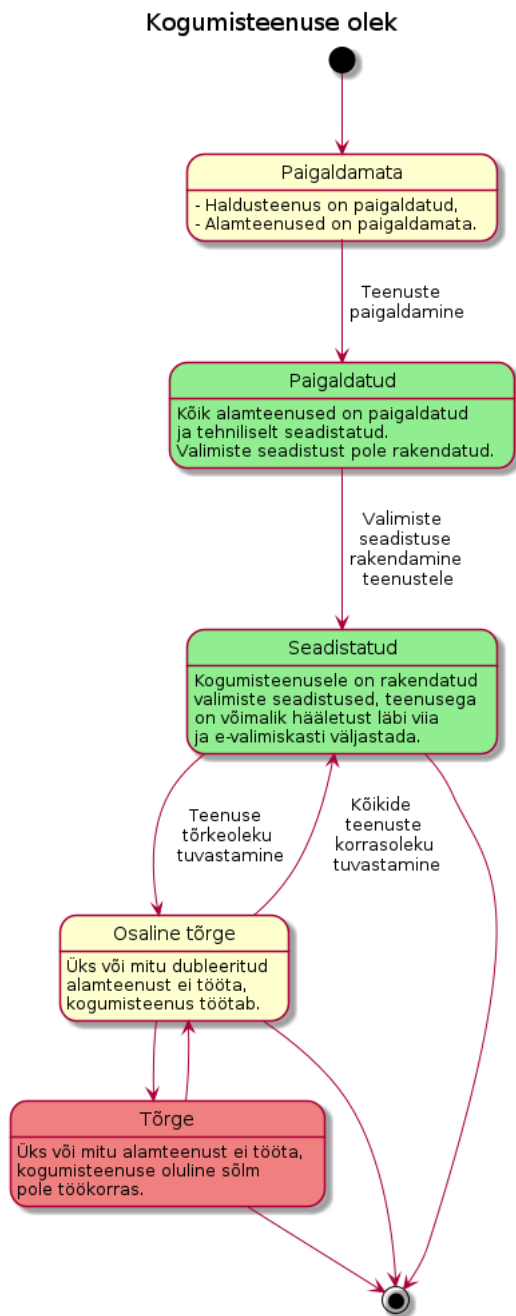
Kogumisteenuse seisund on jälgitav alates haldusteenuse edukast paigaldamisest, algne seisund on **Paigaldamata**.

Paigaldamata

Toimub usaldusjuure ja tehnilise seadistuse rakendamine kogumisteenusele:

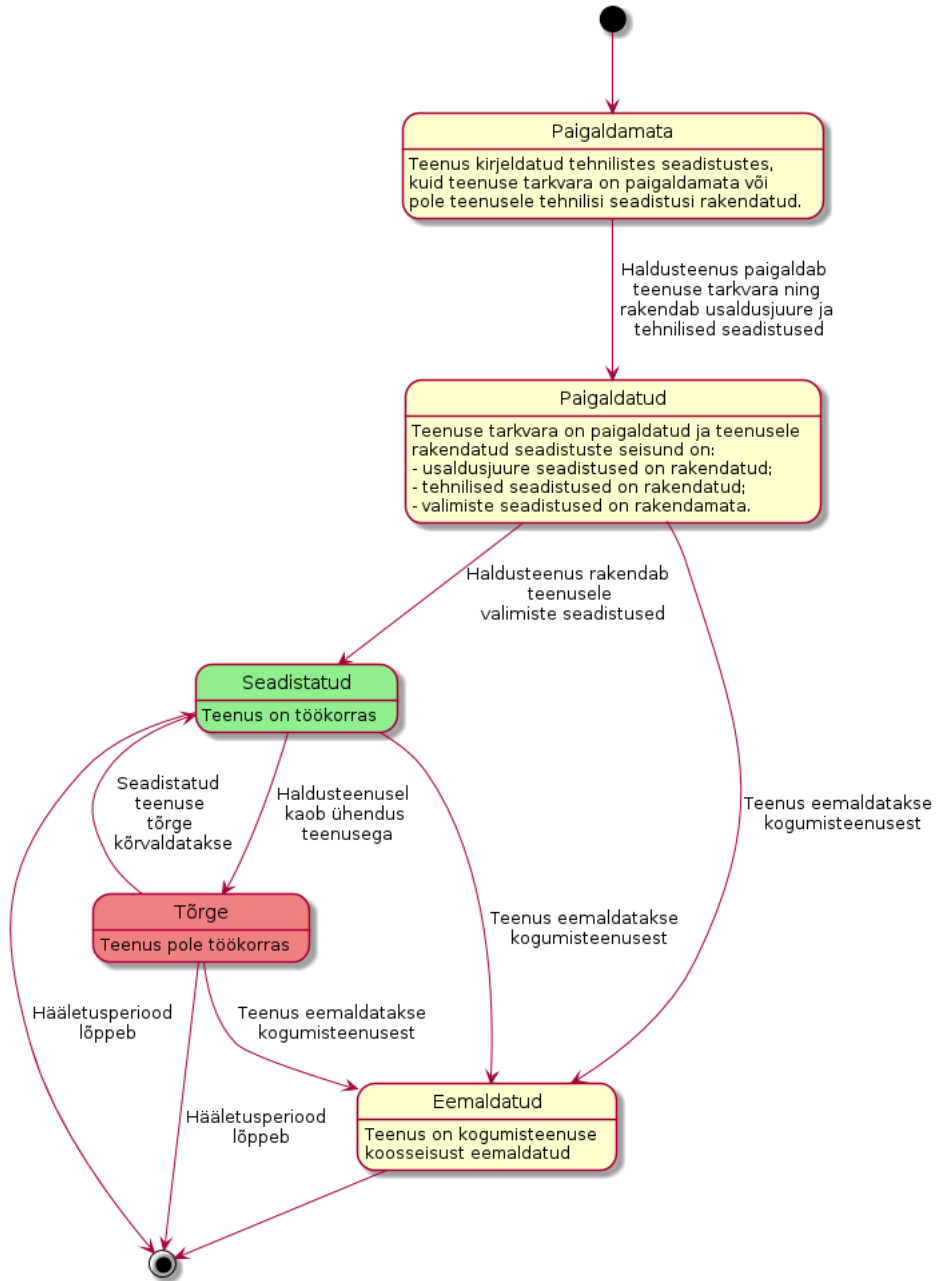
1. Seadistuste laadimine kogumisteenusesse;
2. Tehnilises seadistuses kirjeldatud alamteenuste paigaldus;
3. Usaldusjuure ja tehniliste seadistuste rakendamine alamteenustele;

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



Joonis 2.7. Kogumisteenuse olekudiagramm. Olekud vastavalt värvusele: kollane - seadistamisel, punane - viga, roheline - töökorras.

Kogumisteenuse alamteenuse olekudiagramm



Joonis 2.8. Haldusteenuse poolt registreeritud alamteenuse olekudiagramm. Olekud vastavalt värvusele: kollane - seadistamisel, punane - viga, roheline - töökorras.

Paigaldatud

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

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-valimiskasti 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 olukorran, kus süsteemiga on võimalik teenust osutada, saab süsteemi uueks olekuks **Osaline tõrge**.

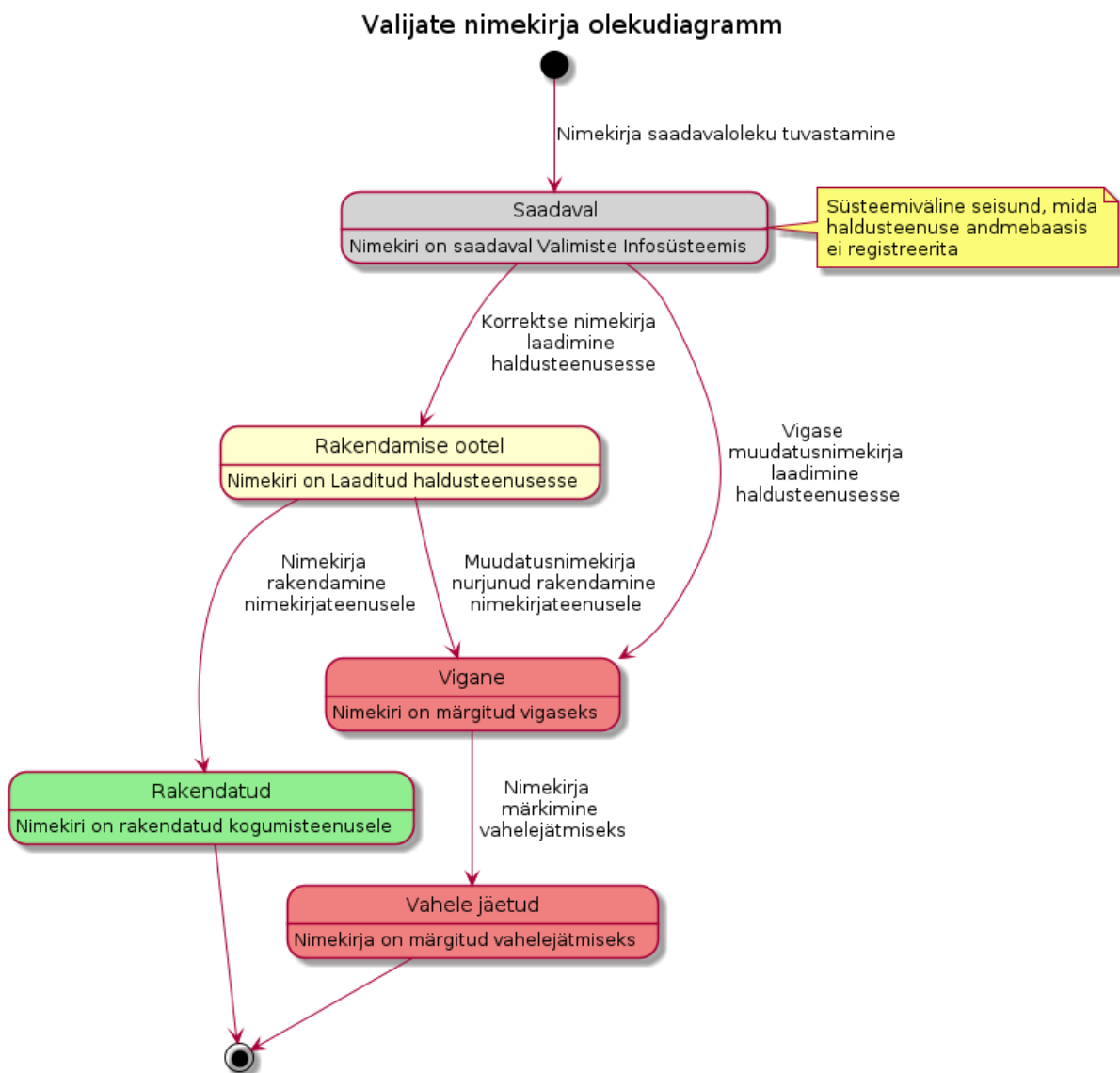
Eemaldatud

Teenus on konfiguratsioonist eemaldatud.

2.6 Valijate nimekirjade olekud haldusteenuses

Valijate nimekirja olek võib olla:

1. **Rakendamise ootel** - nimekiri on laaditud haldusteenusesse;
2. **Rakendatud** - nimekiri on rakendatud kogumisteenusele;
3. **Vigane** - nimekiri on märgitud vigaseks, haldusteenus uusi valijate nimekirjade muudatusi ei laadi;
4. **Vahele jäetud** - vigane nimekiri on märgitud vahelejätmiseks.



Joonis 2.9. Valijate nimekirja olekudiagramm

Siirdeprotsessid:

1. Nimekirja laadimine haldusteenusesse:
Algnimekirja laadib kogumisteenuse operaator, nimekirja olekuks saab **Rakendamise ootel**;

Muudatusnimekirja laadib haldusteenus. Vastavalt valideerimise tulemusele saab nimekirja olekuks kas **Rakendamise ootel** või **Vigane**;

2. **Rakendamine kogumisteenusele**: viib läbi haldusteenus **rakendamise ootel** olekus nimekirjaga. Õnnestumisel määratakse nimekirja olekuks **Rakendatud**, vea korral **Vigane**;
3. **Vahelejätmine**: operaator määrab olekuga **Vigane** nimekirjale oleku **Vahele jäetud**.

3.1 Üldpõhimõtted

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

Rakendused on programmeeritud Java keeles.

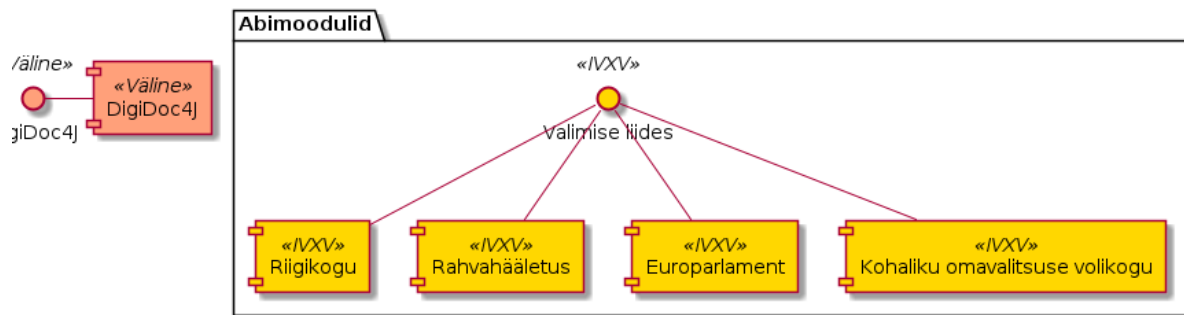
Väliste infosüsteemidega suhtlevad rakendused kasutavad maksimaalselt olemasolevaid liideseid/andmestruktuure.

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.

Relevantsete rakendused toetavad ElGamal krüptosüsteemi täisarvujäägikorpustel ning P-384 elliptilisel kõveral. Lugemistõend on realiseeritud Schnorri nullteadmusestusel põhineval protokollil.

Valimise liides on rakenduste jaoks unifitseeritud, see võimaldab erinevate valimistüüpide realiseerimist moodulitena. Digiallkirja verifitseerimise funktsionaalsus on loodud [digidoc4j](https://github.com/open-eid/digidoc4j)³ teegi abil. Abimoodulite kasutamist alljärgnevatel skeemidel eraldi välja ei tooda.

³ <https://github.com/open-eid/digidoc4j>



Joonis 3.10. Rakenduste abimoodulid

Rakenduste seadistamine

Rakendused seadistatakse kas digitaalalkirjastatud konfiguratsioonipakiga või käsureavõtmetega. Käsurreavõtmed ei toeta hierarhilise struktuuriga seadistuste sisestamist. Seadistused konfiguratsioonipakis kirjeldatakse YAML-keeles:

```

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
  
```

(jätkub järgmisel leheküljel)

```
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

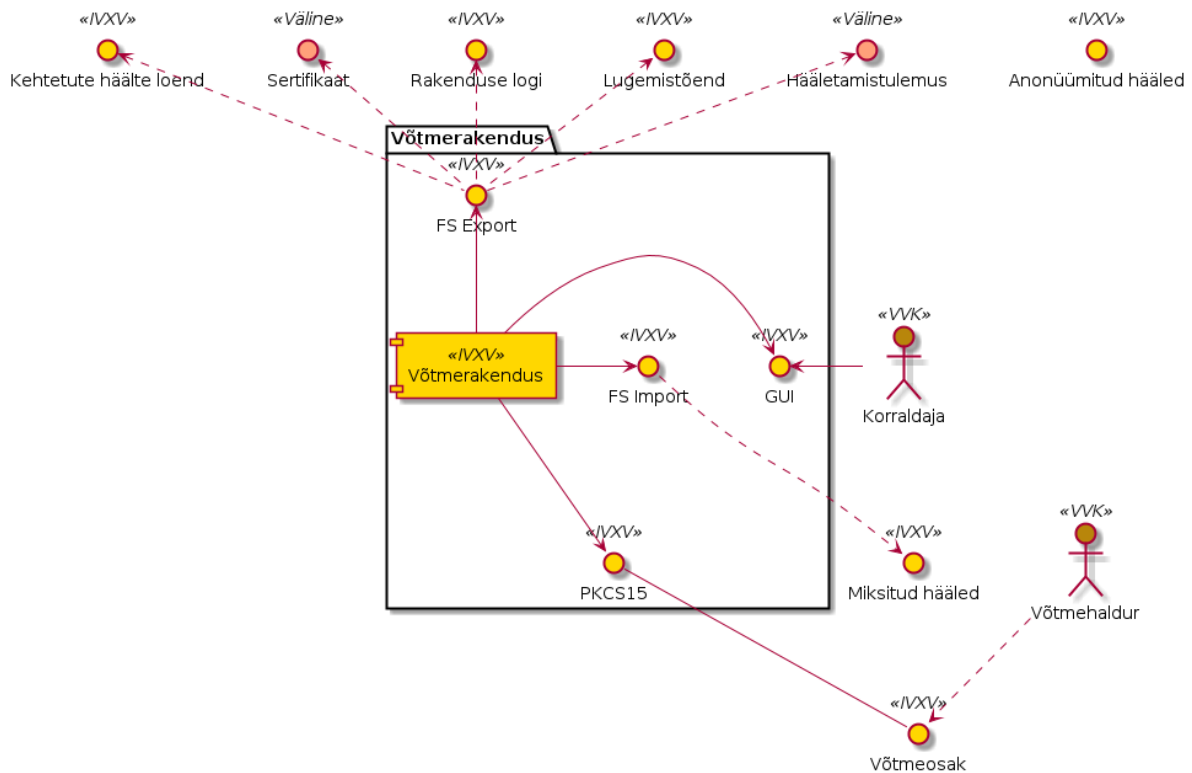
Võtmerakendusega genereeritakse iga hääletamise jaoks hääle salastamise ja hääle avamise võti, samuti toimub selle abil hääle lugemine ja tulemuse väljastamine.

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

Võtmeosakud genereeritakse operatiivmälus ning talletatakse PKCS15-liidese vahendusel kiipkaardile.

Võtmerakenduse sisend võtme genereerimisel on:

- Võtmepaari identifikaator;
- Krüptosüsteemi ElGamal spetsifikatsioon – täisarvujäägikorpus või P-384 elliptikõver ning võtmepikkus;



Joonis 3.11. Võtmerakenduse liidesed

- M-N läviskeemi spetsifikatsioon, mis peab vastama reeglile $N \geq 2 * M - 1$;
- N PKCS15-ühilduvat kiipkaarti;

Võtmerakenduse väljund võtme genereerimisel on:

- Ilesigneeritud 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 vastavalt hanke-dokumentides viidatule);
- Rakenduse detailne tegevuslogi;
- Rakenduse detailne vealogi.

3.3 Töötlemisrakendus

Töötlemisrakendusega verifitseeritakse, tühistatakse ning anonüümistatakse hääletamisperioodil kogutud hääli vastavalt Üldkirjelduse jaotisele 7.6.

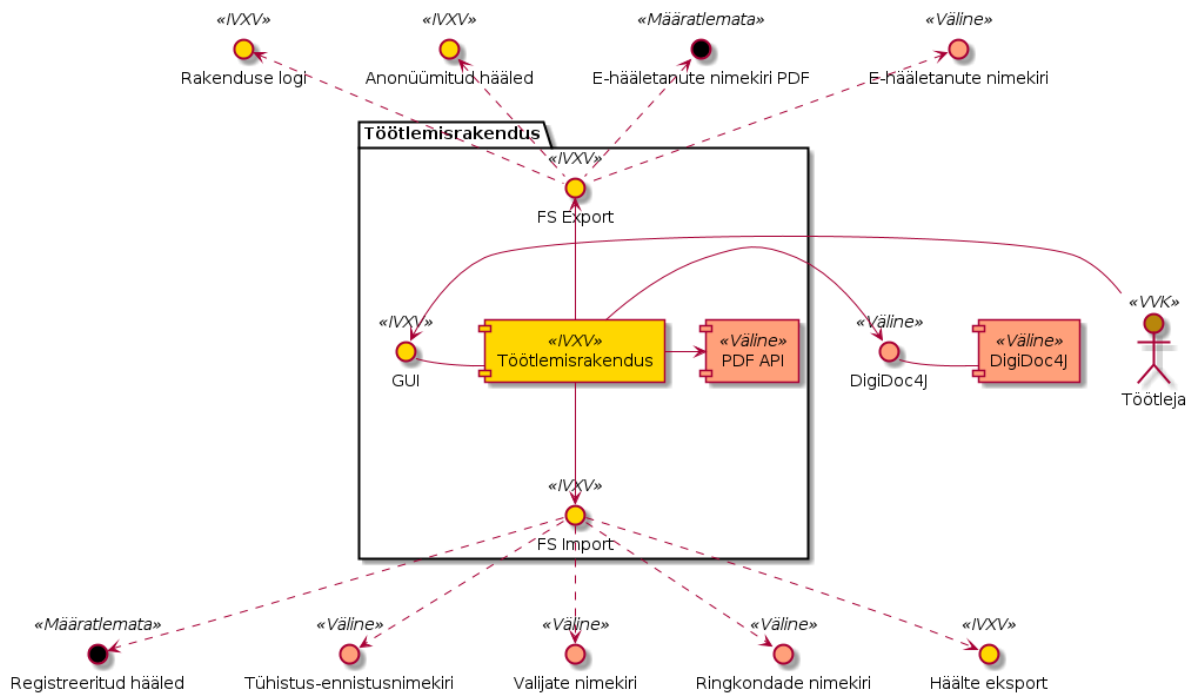
Töötlemisrakenduse sisendid on:

- 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 defineeritud liidestele ja sõltuvustele kasutab töötlemisrakendus kolmanda osapoolte teeki PDF-ide väljastamise funktsionaalsuse teostamiseks.



Joonis 3.12. Töötlemisrakenduse liidesed

Elektrooniliste häälte täielik töötlemine

Elektrooniliste häälte täielik töötlemine on tegevus, mille käigus töötlemisrakendus võrdleb Kogumisteenuse poolt talletatud häälte hulka registreerimisteenuse poolt talletatud häälte hulgaga, 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 kooskõllalisuse 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.

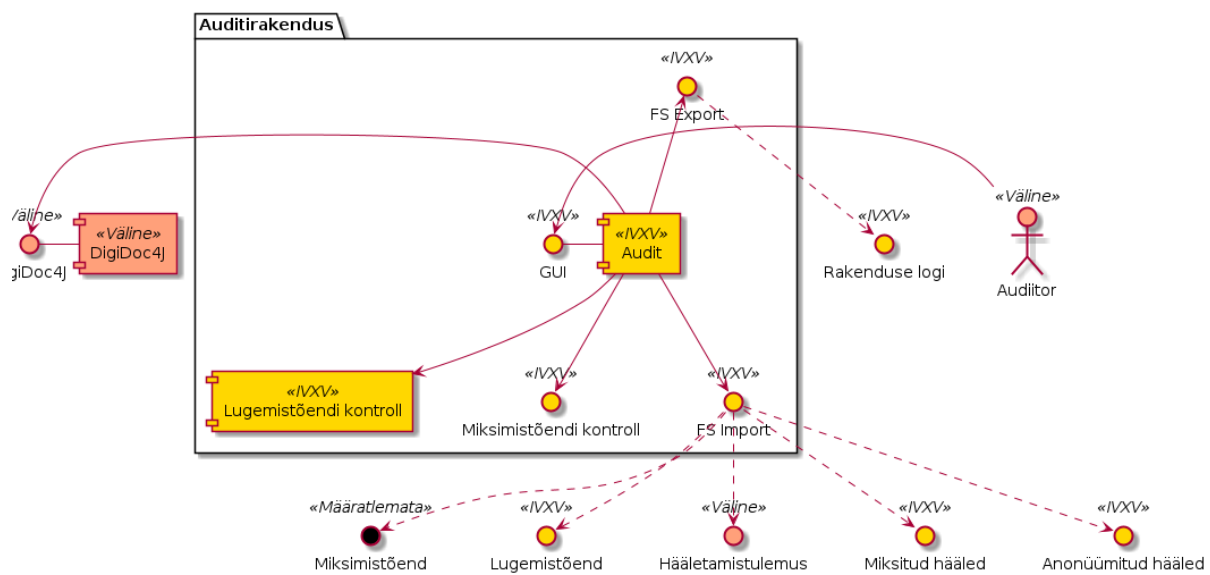
3.4 Auditirakendus

Auditirakendusega (joonis 9) verifitseeritakse matemaatiliselt häälte kokkulugemise korrektsust ning miksimise kasutamisel ka miksimise korrektsust.

Auditirakenduse sisendid on;

- anonüümitud hääled;
- miksitud hääled;
- Verificatum miksimistõend;
- hääletamistulemus.

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



Joonis 3.13. Auditirakenduse liidesed

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 kasutamisevajadusega on üles loetletud järgnevas 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 4.1. IVXV raamistiku tööks kasutatavad kolmandate osapoolte komponendid

Nimi	Version	Litsents (SPDX)	Kasutusvajadus
Bootstrap ⁴	3.4.1	MIT	Kogumisteenuse haldusteenuse kasutajaliidese kujundus
Bouncy Castle	1.58	MIT	ASN1 käsitlemine, andmetüübi BigInteger abifunktsioonid
Bottle ⁵	0.12.15	MIT	Raamistik kogumisteenuse haldusteenuse veebiliidese teostamiseks
CAL10N	0.7.7	MIT	Mitmekeelsuse tugi, tõlkefailide valideerimine
Digidoc 4j	2.1.0	LGPL-2.1-only	BDoc konteinerite käsitlemine
Digidoc 4j DSS	5.2.d4j.3	LGPL-2.1-only or EUPL-1.1	Digidoc 4j sõltuvus
Apache Commons (cli 1.4, codec 1.10, collections4 4.1, io 2.5, lang3 3.6, logging 1.2, compress 1.3)	.	Apache-2.0	Digidoc 4j ja PDFBox sõltuvused
Apache HttpComponents	4.5.3	Apache-2.0	Digidoc 4j sõltuvus
Apache Santuario	2.0.9	Apache-2.0	Digidoc 4j sõltuvus
Google Guava	20.0	Apache-2.0	Digidoc 4j sõltuvus
JDigiDoc	3.12.1	LGPL-2.1-only	Digidoc 4j sõltuvus
StaX	1.0-2	Apache-2.0	Digidoc 4j sõltuvus
log4j	1.2.6	Apache-2.0	Digidoc 4j sõltuvus
Woodstox	4.4.1	Apache-2.0	Digidoc 4j sõltuvus
Xalan-Java	2.7.2	Apache-2.0	Digidoc 4j sõltuvus
Xml Apis	1.3.04	Apache-2.0	Digidoc 4j sõltuvus
Docopt ⁶	0.6.2	MIT	Kogumisteenuse haldusutiliitide käsuterealiidese teostus

jätub järgmisel leheküljel

Tabel 4.1 – jätk eelmisele leheküljele

Nimi	Version	Litsents (SPDX)	Kasutusvajadus
Fasteners ⁷	0.14.1	Apache-2.0	Kogumisteenuse haldusteenuse protsesside lukustus
etcd ⁸	3.2.26	Apache-2.0	Talletusteenusena kasutatav hajus võti-väärtus andmebaas
github.com/golang/protobuf ⁹	1.3.2	BSD-3-Clause	etcd klientteegi sõltuvus
Glassfish JAXB	2.3.3	BSD-3-Clause	Java XML teek
google.golang.org/genproto ¹⁰	58ce757	Apache-2.0	etcd klientteegi sõltuvus
google.golang.org/grpc ¹¹	1.22.1	Apache-2.0	etcd klientteegi sõltuvus
golang.org/x/net ¹²	74dc4d7	BSD-3-Clause	etcd klientteegi sõltuvus
golang.org/x/sys ¹³	fc99dfb	BSD-3-Clause	etcd klientteegi sõltuvus
golang.org/x/text ¹⁴	0.3.2	BSD-3-Clause	etcd klientteegi sõltuvus
Gradle	6.4	Apache-2.0	Java rakenduste ehitamise raamistik
HAProxy ¹⁵	2.0.13	GPL-2.0-or-later	Vahendusteenusena kasutatav TCP-proksi
IvyPot	0.12	Apache-2.0	Gradle ehitusraamistiku laiendus sõltuvuste haldamiseks ja rakenduste ehitamiseks vallasrežiimis
Jackson	2.8.9	Apache-2.0	JSON vormingus failide lugemine ja kirjutamine
jQuery ¹⁶	3.3.1	MIT	Kogumisteenuse haldusteenuse kasutajaliides
Logback	1.2.3	EPL-1.0 or LGPL-v2.1-only	Logimise API SLF4J teostus
Logback JSON	0.1.5	EPL-1.0 or LGPL-v2.1-only	Logback logija laiendus JSON vormingus logikirjete koostamiseks Jackson teegi abil

jätkub järgmisel leheküljel

Tabel 4.1 – jätk eelmisele leheküljele

Nimi	Version	Litsents (SPDX)	Kasutusvajadus
metisMenu ¹⁷	1.1.3	MIT	Kogumisteenuse haldusteenuse kasutaja-liides
PDFBox	2.0.8	Apache-2.0	PDF vormingus raportite genereerimise tugi Java rakendustele
PyYAML ¹⁸	5.3.1	MIT	Kogumisteenuse seadistusfailide töötlemise tugi haldusteenusele
Schematics ¹⁹	2.0.1	BSD-3-Clause	Kogumisteenuse seadistusfailide valideerimise tugi haldusteenusele
SLF4J	1.7.25	MIT	Standardne logimise API
SnakeYAML	1.18	Apache-2.0	YAML vormingus andmete lugemine
SB Admin 2 ²⁰	3.3.7+1	MIT	Kogumisteenuse haldusteenuse kasutaja-liidese kujundus

Tabel 4.2: IVXV raamistiku testide kasutatavad kolmandate osapoolte komponendid

⁴ <http://getbootstrap.com>

⁵ <https://bottlepy.org/>

⁶ <http://docopt.org/>

⁷ <https://github.com/harlowja/fasteners>

⁸ <https://coreos.com/etcd>

⁹ <https://github.com/golang/protobuf>

¹⁰ <https://google.golang.org/genproto>

¹¹ <https://google.golang.org/grpc>

¹² <https://golang.org/x/net>

¹³ <https://golang.org/x/sys>

¹⁴ <https://golang.org/x/text>

¹⁵ <http://www.haproxy.org/>

¹⁶ <https://jquery.org/>

¹⁷ <https://github.com/onokumus/metisMenu>

¹⁸ <http://pyyaml.org/>

¹⁹ <https://github.com/schematics/schematics>

²⁰ <https://github.com/BlackrockDigital/startbootstrap-sb-admin-2>

Nimi	Version	Litsents (SPDX)	Kasutusvajadus
Hamcrest	1.3	BSD-3-Clause	Loetavam assert-meetodite kasutamine Java üksuste testides
JUnit	4.12	EPL-1.0	Java testimisraamistik
JUnitParams	1.1.0	Apache-2.0	Testide parametrizeerimise tugi
Mockito	2.10.0	MIT	Testitava koodi sõltuvuste mockimise tugi
Byte Buddy	1.9.10	Apache-2.0	Mockito sõltuvus
Objenesis	2.6	Apache-2.0	Mockito sõltuvus
libdigidocpp-tools	3.14.5.1404	GPL-2.1-or-later	Testandmete genereerimine
PyTest	6.2.3	MIT	Üksuste testimise tugi Pythonile
Requests	2.25.1	Apache 2.0	HTTP päringute moodul Pythoni testidele

Tabel 4.3: IVXV raamistiku arendamiseks ja/või testimiseks kasutatavad kolmandate osapoolte tööriistad

Nimi	Version	Litsents (SPDX)	Kasutusvajadus
Behave ²¹	1.2.6	BSD-2-Clause	Regressioonitestide käivitaja (<i>Behavior-driven development</i>)
Docker ²²	18.06 (või uuem)	Apache-2.0	Regressioonitestide läbiviimise keskkond - tarkvarakonteinerid
Docker Compose ²³	1.28.6	Apache-2.0	Regressioonitestide läbiviimise keskkond - tarkvarakonteinerite haldus
Sphinx ²⁴	3.4.3	BSD	Dokumentatsiooni genereerimine

²¹ <https://github.com/behave/behave>

²² <http://www.docker.com/>

²³ <http://www.docker.com/>

²⁴ <http://www.sphinx-doc.org/>

PEATÜKK 5

Viited

- [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.
- [HMOV16] Sven Heiberg, Tarvi Martens, Priit Vinkel, Jan Willemsen, 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. Elektroonilise hääletamise infosüsteemi arenduse hange, Vabariigi Valimiskomisjon, 2016
- [ÜK2016] Elektroonilise hääletamise üldraamistik ja selle kasutamine Eesti riiklikel valimistel. Elektroonilise Hääletamise Komisjon, Tallinn 2016