Tallinna Ülikool

Digitehnoloogiate instituut

Camunda platvormil protsessimootori rakendamine Elektrilevi äriprotsessides

Bakalaureusetöö

Autor: Keio Arula

Juhendaja: Jaagup Kippar

Autor: ..... „ ..... „ 2017

Juhendaja: ..... „ ..... „ 2017

Instituudi direktor: ..... „ ..... „ 2017

Tallinn 2017

Autorideklaratsioon

Deklareerin, et käesolev bakalaureusetöö on minu töö tulemus ja seda ei ole kellegi teise poolt varem kaitsmisele esitatud. Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on viidatud.

....

(kuupäev) (autor)

Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks

Mina _____Keio Arula_____ (sünnikuupäev: 03.10.1992)

(*autori nimi*)

1. annan Tallinna Ülikoolile tasuta loa (lihtlitsentsi) enda loodud teose
   Camunda platvormil protsessimootori rakendamine Elektrilevi äriprotsessides

   (*lõputöö pealkiri*)

   mille juhendaja on _____Jaagup Kippar_____,

   (*juhendaja nimi*)

   säilitamiseks ja üldsusele kättesaadavaks tegemiseks Tallinna Ülikooli Akadeemilise Raamatukogu repositooriumis.

2. olen teadlik, et punktis 1 nimetatud õigused jäävad alles ka autorile.
3. kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest tulenevaid õigusi.

Tallinnas/Haapsalus/Rakveres/Helsingis, _____

(digitaalne) allkiri ja kuupäev

Sisukord

Lühendid ja mõisted

DWF	LD toode Domino WorkFlow
ELLI	Elektrilevi liitumiste infosüsteem
BPMN	Äriprotsessi mudel
DMN	Juhtumipõhine mudel
LDAP	Otsustusmudel
SSO	Ühtne autentimine mitmel rakendusel
SQL	Andmebaasi keel
SOAP	Lihtne objektide ligipääsu protokoll
SVN	Versiooni haldussüsteem
Java	Programmeerimiskeel
Spring Boot	Java raamistik
Gradle	Rakenduse ehitamise automatiseerimise süsteem
WAR	Veebirakenduse arhiivi fail
Tomcat	Server

Sissejuhatus

Käesoleval ajal on enamikel ettevõtetel kasutusel tehnoloogiad, automaatikad ning infosüsteemid, mis tõhustavad igapäevaseid äriprotsesse. Infotehnoloogia integreerimine ärikeskkonda on kujunenud väga kasumlikuks ning seetõttu on üha enam organisatsioone otsustanud investeerida infosüsteemidesse.

Elektrilevi üheks olulisemaks ülesandeks on tagada Elektrilevi klientidele kvaliteetne teenindus. Sellega seonduvalt on oluline, et Elektrilevi ametnikud on oma tegevustes väga täpsed, et kogu lepingutega seotud informatsioon oleks kergelt ja kiirelt sisestatav ja kättesaadav ning sisult korrektne. Sellegi poolest on Elektrilevis tegevusi, mida teostatakse käsitsi ning mida ei hoita ühtses infosüsteemis. Elektrileviga liitumiste taotlusi on siiamaani esitatud paberil või PDF faili-formaadis - nende töötlemine on ajakulukas ning nõuab erinevate osakondade ametnike tööd. Samuti on raske jälgida liitumise täit protsessi ning millises äriprotsessi sammus liitumine asub. On tekkinud vajadus Elektirlevi liitumiste sisestamiseks ja jälgimiseks rakendada infosüsteemi - täpsemini protsessimootorit, mis oleks kasulik nii Elektrilevi ametnikule kui ka tulevasele Elektrilevi kliendile.

Käesoleva töö eesmärk on analüüsida ning kirjeldada Elektrilevi äriprotsessides Camunda platvormil oleva protsessimootori rakendamist. Töö annab ülevaate andmetest, funktsioonidest ja arhitektuurist, mis peale projekti lõppu antud infosüsteem peaks sisaldama.

Esimene peatükk annab ülevaate projektist ning projekti taustast, toob välja eesmärgid, mida protsessimootori rakendamine täita aitab. Samuti kirjeldab praeguse taoltuste esitamise äriprotsessi ja sellest tulenevaid ning lahendust vajavaid probleeme.

Teine peatükk käsitleb projekti analüütilise, teoreetilise ning praktilise osa loomiseks kasutatud meetodeid.

Kolmandas peatükis on kirjeldatud töö analüütiline osa, mis sisaldab endas infosüsteemi funktisonaalsete ja mittefunktsionaalsete nõuete kirjeldust, ülevaadet äriprotsessist, mida protsessimootoriga lahendati ning sellega tekkinud probleemidest.

Neljandas peatükis on kirjeldatud lõputöö praktilise osa tulemus, mis annab ülevaate projekti käigus sündinud infosüsteemi arhitektuurist. Peatükis on toodud välja komponentdiagramm, andmemudel, kirjeldus süsteemi integratisoonidest ning ülevaade Camunda arhitektuurist.

Töö lisades on funktsionaalsed ja mittefunktsionaalsed nõuded ning integratsioonid ja autori poolt valminud rakenduse installijuhend.

1. Probleemi püstitus ja projekti eesmärk

Probleemi püstituse ja projekti eesmärgi peatükis kirjeldatakse praegust Elektrilevi äriprotsessi ja tuuakse välja suurimad probleemid. Ühtlasi räägitakse projekti taustast ning projekti eesmärgist.

1.1. Projekti taust

Antud alapeatükis antakse ülevaade Elektrilevist ning protsessimootorist.

1.1.1. Elektrilevi OÜ

Elektrilevi on Eesti suurim võrguettevõtja, kes toob elektri kohale pea kõikidesse kodudesse ja ettevõtetesse üle terve Eesti. Suurematest aladest on teiste võrguettevõtjate käes vaid Läänemaa, Viimsi ning Narva ja selle ümbrus. Sellest hoolimata on Elektrilevil kokku ligi 475 000 klienti.

Selleks, et teenindada kõiki kliente, on Elektrilevil vaja hooldada ja uuendada ligi 61 000 kilomeetrit elektriliine ja üle 22 000 alajaama. Nii Elektrilevi, kui ka kõik teised võrguettevõtjad, haldavad kuni 110 kV pingega elektrivõrku, mis tuuakse põhivõrguseadmetest klientidele.

Tagamaks klientidele kindlat ning sujuvat võrguteenust, on Elektrilevi seadnud endale kaks tähtsat eesmärki:

- elektrivarustuse kvaliteedi parandamine. Eesmärgi saavutamiseks tuleb uuendada võrku ning ehitada uusi alajaamu. Uuendamine on oluline, et saavutada töökindel ja võimalikult väheste katkestustega elektrivõrk, mis vastaks pingekvaliteedi nõuetele;
- teeninduse kvaliteedi parandamine. Eelkõige soovitakse klientide pöördumistele võimalikult kiiret ja kvaliteetset lahenduste leidmist. Teeninduse kvaliteedi ja kiiruse parandamise ühe osana on neil vaja uut liitumiste infosüsteemi, tänu millele saab Elektrilevi oma töid kiiremini ja täpsemalt hallata ning koordineerida [1].

Elektrilevi poolt tellitud liitumiste infosüsteem hõlmab endas nelja taotluse kategooriat:

- uus liitumine. Uue liitumislepingu sõlmimiseks;
- tarbimistingimuste muutmine. Olemasolevate tarbimistingimuste muutmiseks;
- võrgu ümberehitus kliendi soovil. Võrgu ümberehitamine vastavalt kliendi soovidele;
- tehnilised tingimused. Tasuliste või tasuta tehniliste tingimuste dokumentide taotlemiseks

1.1.2. Protsessimootor

Infosüsteemid muutuvad ajas üha keerulisemateks. Sellega seoses on tekkinud vajadus kasutada protsessimootoreid. Protsessimootorid aitavad pikkadel ja keerukatel äriprotsessidel silma peal hoida ning neid lihtsalt hallata. Protsessimootoreid võivad tarkvara arenduse ettevõtted ise arendada ning on olemas ka karbitooteid. Tuntumad karbitooted protsessimootoritest on näiteks Camunda, Activity ja Bonita.

Käesolevas bakalaureusetöös kirjeldatavas projektis on kasutusel protsessimootor. Protsessimootoriks on valitud vabavaraline versioon Camunda protsessimootorist. Camunda protsessimootor osutus valituks, kuna firmas, kus lõputöö autor töötab, oli sellekohane kompetents ning ühtlasi vastas protsessimootor Elektrilevi poolt seatud nõuetele. Elektrilevi poolt seatud nõuded on:

- peab olema vabavaraline;
- peab saama piisavalt kohandada Elektrilevi nõudmistele;
- peab omama administreerimise kasutajaliidest;
- peab olema küllaltki lihtne ja kiiresti selgeks õpitav, et tulevikus saaks Elektrilevi sinna ise arendusi teostada.

1.2. Olemasolev äriprotsess

Elektrileviga liitumine on pikk protsess, mis saab alguse kliendi taotluse esitamisest. Hetkel on kliendil selleks kaks võimalust:

- esimene, kliendile mugavaim, võimalus on esitada liitumistaotlus läbi Elektrilevi veebikeskkonna. Peale iseteeninduses taotluse esitamist, saadetakse taotlus Elektrilevi liitumisspetsialistile, kes vaatab iseteenindusest tulnud taotluse üle ja lisab selle käsitsi liitumiste infosüsteemi. Selline taotluse esitamise viis on kliendile küll mugav ja lihtne, sest klient saab seda teha ükskõik kus, kuid liitumisspetsialistile tähendab see palju käsitööd ja sellest tulenevalt võivad tekkida vead;
- teine võimalus on kliendil leppida kokku aeg liitumisspetsialistiga ja seejärel minna Elektrilevisse kohale ning seal viimasega taotlus ära täita. Sellisel juhul on vigade tekkimine viidud miinimumini, sest taotlust teevad 2 inimest. Kuid tänapäevases infoühiskonnas ei ole see kliendile kuigi mugav lahendus, kuna ollakse harjunud erinevaid taotlusi esitama veebis.

Et liitumine Elektrileviga ei oleks pikk ja keeruline, on kasutusele võetud protsessimootor, mis aitaks ülesandeid automatiseerida ning ühtlasi saaks lihtsamini jälgida, kui kaugel protsess parajasti on. Enne Camunda protsessimootori rakendamist juhitakse Elektrilevis liitumisprotsesse LD tootega Domino WorkFlow (edaspidi DWF).

1.3. Probleemi püstitus

Organisatsioonid arendavad oma infosüsteeme, et saavutada tähtsaid ärilisi eesmärke, milleks on näiteks tööjõu kulude vähendamine ning produktiivsuse ja efektiivsuse kasv. Uus Elektrilevi liitumiste infosüsteem peab lahendama mitmeid olulisi probleeme.

- infosüsteemi ärivajadusele vastavana hoidmine on raskendatud. Elektrilevis puudub DWF keskkonnas arenduste ja muudatuste tegemiseks kompetents. Kompetentsi puudumisest tulenevalt on ka väikearenduste tegemine olemasolevasse infosüsteemi ebapropotsionaalselt aja- ja ressursimahukas;

- vajadus on liitumisprotsessi kaasajastada. Vajadus liitumisprotsessi kaasajastada on püstitanud protsessijuhtimisele lisanõudeid, mida praeguses DWF keskkonnas oleks liiga keeruline või lausa võimatu realiseerida;
- vajadus on vähendada protsessijuhtimise administreerimist. Praegu tekitab DWF keskkonnas tööde menetlemine süsteemitõrkeid, mida peavad süsteemiadministraatorid igapäevaselt lahendama.
- erinevad infosüsteemid ei suhtle omavahel piisavalt. Elektrilevi liitumisprotsess eeldab mitme erineva infosüsteemiga suhtlemist, sest selle käigus on vaja luua arveid, tellimusi ja projekte. Kõike neid ei ole mõistlik hoida ühes infosüsteemis, kuna liitumisprotsess ei ole ainus, mis taolisi teenuseid vajab. Selle tarbeks on Elektrilevis loodud palju erinevaid infosüsteeme, mis kõik täidavad erinevaid ülesandeid. Näiteks üks infosüsteem tegeleb arvetega, teine tegeleb tellimustega ning kolmandas infosüsteemis saab projekti algatada. Elektrilevi infosüsteemide vahel puudub hulk teenuseid. Teenuste puudumise tõttu peab ühest süsteemist andmeid teise käsitsi tõstma;
- vajadus on täpsemini mõõta tegevuste täitmise aega. Praeguses süsteemis tehakse väga palju ära käsitsi, seetõttu ei saa tegevuste kestust täpselt mõõta. Kuna ei hoomata täpselt, kui palju mõni tegevus reaalselt aega võtab, siis see tekitab olukorra, kus puudub ülevaade, mis tegevused on protsessis kitsaskohaks.

1.4. Eesmärk

Elektrilevi liitumiste infosüsteemi arendamise eesmärgiks on äriprotsessi kaasajastamine ning tõhustamine, viies kõik võimalikud tegevused süsteemseks. Sellisteks tegevusteks on näiteks iseteenindusest tuleva taotluse vastuvõtmine, valideerimine ja võimalusel automaatmenetlusse suunamine, et liitumisspetsialist ei peaks sellega ise tegelema. Samuti on eesmärgiks panna infosüsteemid omavahel suhtlema, et andmed liiguksid süsteemide vahel automaatselt.

1.5. Autori roll projektis

Autori roll Elektrilevi liitumiste infosüsteemi (edaspidi ELLI) loomisel oli tarkvara arendamine. Töö autor töötab tarkvaraarenduse ettevõttes Icefire OÜ ning täitis käesolevas bakalaureusetöös kirjeldatavas projektis arhitekti ja tarkvaraarendaja rolli.

Autori poolt valminud rakendus on tellitud Elektrilevi OÜ poolt. Projekt koosnes äriprotsesside kirjeldustest, liidestuste teostamisest vastu teisi Elektrilevi rakendusi ning ELLI rakendusest endast, mis sisaldab administraatori ja tavakasutajate vaadet. ELLI rakendus on loodud vastavalt Elektrilevi OÜ standarditele.

2. Kasutatud metoodika

Autori ülesandeks oli käesolevas projektis Elektrilevi liitumiste infosüsteemi arendamine.

Projekti alustamiseks oli autoril vaja tutvuda hankedokumentidega ning süsteemianalüüsiga. Meeskonnas lepiti kokku projekti osapoolte vastutusalad. Toimus koosolek arendusmeeskonna ja kliendi vahel, et leppida kokku edasine tööprotsess ning pandi paika ka ligikaudsed projekti üleandmise tähtajad.

Arendus teostati nii süsteemianalüüsi, kui ka analüütikute ja tellija sisendi põhjal. Infovahetuseks kasutas autor antud projektis järgmisi infovahetuse võimalusi:

- koosolekud. Lepiti kokku aeg ning koosoleku kava, millistel teemadel on plaanis rääkida. Koosolekud toimusid nii tellija kui teostaja ruumides;
- e-post. Kirjavahetustesse kaasati enamus projektiga seotud inimesed, et tehtud otsused ja vastused jõuaksid kõigi vajalike osapoolteni. Projektijuhid olid kirjavahetustesse alati kaasatud;
- Skype. Skype kasutati kliendiga suhtlemiseks sh kasutati ka videokõnede funktsionaalsust. Skype kasutati peamiselt lühemate ja täpsustavate küsimuste küsimiseks;
- Fleep. Fleep'i kasutati firmasiseseks suhtlemiseks ja info hankimiseks;
- Telefon. Telefoni kasutati vaid juhul, kui oli vaja kiiresti infot saada.

Töö autoril oli Elektrilevi liitumiste infosüsteemi arendamisel kasutusel mitmeid erinevaid tehnoloogiaid, millest põhiliseimad on:

- Java 8 – Programmeerimiskeel;
- Spring boot - Java raamistik, mis kiirendab märgatavalt arendamist;
- Thymeleaf - Kasutajaliidese arendamiseks;
- Bootstrap - Kasutajaliidese disain,
- SOAP - Erinevad integratsioonid Elektrilevi infosüsteemide teenuste kasutamiseks,
- Gradle - Projekti ehitamiseks;
- Camunda - Protsessimootor;

- BPMN - Äriprotsessi mudel. Kogu äriprotsess oli BPMN failis kirja pandud, mis oli sisendiks protsessimootorile;
- CMMN - Juhtumipõhine mudel dokumentide koostamiseks. CMMN mudel oli sisendiks BPMN failile;
- Oracle - Andmebaas.

3. Analüüs

Analüüsi peatükis tuuakse välja Elektrilevi liitumiste infosüsteemile kehtivad erinevad nõuded – funktsionaalsed ja mittefunktsionaalsed nõuded. Analüüsis kirjeldatakse ka loodavat äriprotsessi.

3.1. Nõuded

Elektrilevi liitumiste infosüsteemile kehtivad funktsionaalsed ja mittefunktsionaalsed nõuded olid paika pandud Elektrilevi poolt edastatud hankedokumentides. Loodav infosüsteem peab vastama projekti lepingus sätestatud üldistele ning täiendatavatele funktsionaalsetele (vt lisa 1) ja mittefunktsionaalsetele (vt lisa 2) nõutele.

3.2. Äriprotsess

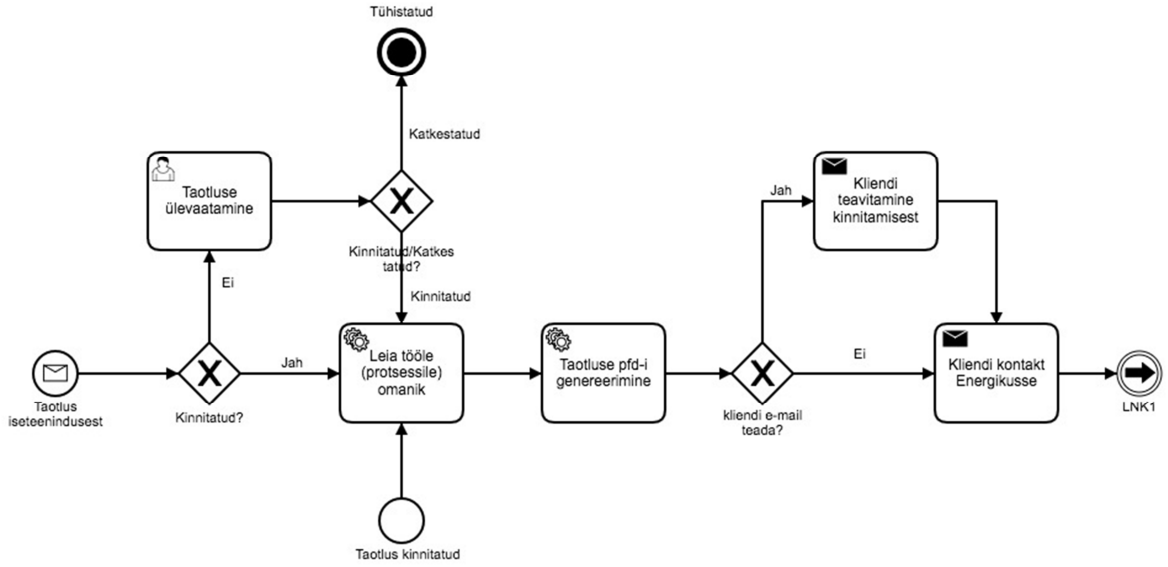
Äriprotsessi alapeatükk annab ülevaate loodavale äriprotsessile ja toob välja tähtsaimad kitsaskohad, mis tegid arendamise keerulisemaks, ja nende lahendused. Lisaks eelnevale implementeeritakse üks funktsionaalne nõue.

3.2.1. Äriprotsessi ülevaade

Elektrilevi liitumiste infosüsteemis on äriprotsess kirja pandud äriprotsesside modelleerimiskeeles BPMN. Lisaks BPMN-le on kasutatud ka juhtumipõhist modelleerimiskeelt CMMN-i. Äriprotsessi põhilisemad punktid on:

- äriprotsess võib alguse saada kahel viisil. Esimene võimalus on taotluse saamine Elektrilevi iseteenindusest. Peale taotluse saamist taotlus valideeritakse ning

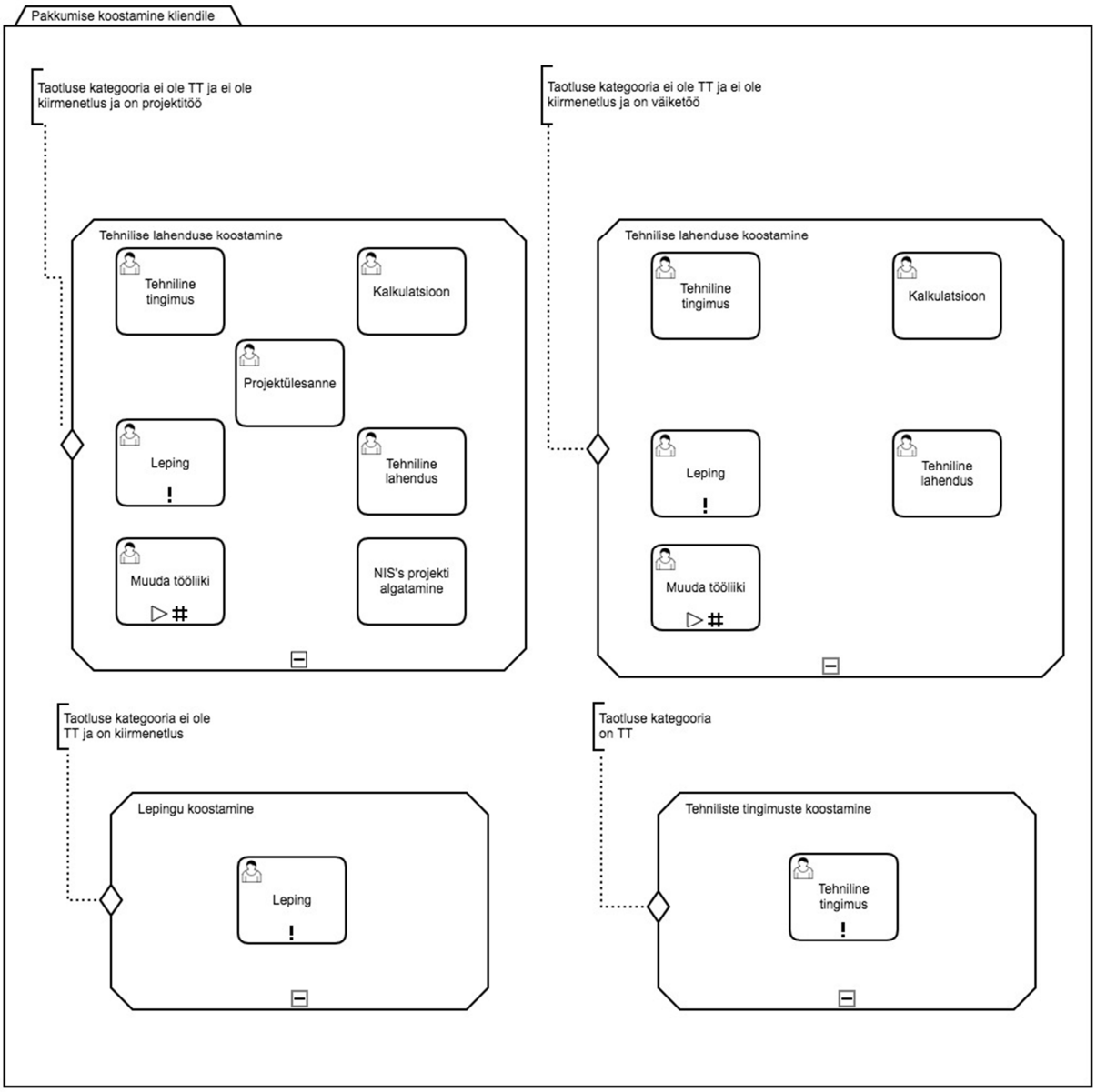
otsustatakse, kas taotlus peab minema liitumisspetsialisti töölauale või liikuda edasi kliendile pakkumise koostamisse. Teine võimalus äriprotsessi algatamiseks on taotluse kinnitamine liitumisspetsialisti poolt Elektrilevi liitumiste infosüsteemis. Peale taotluse kinnitamist liigub äriprotsess kliendile pakkumise koostamise sammu (vt joonis 1);



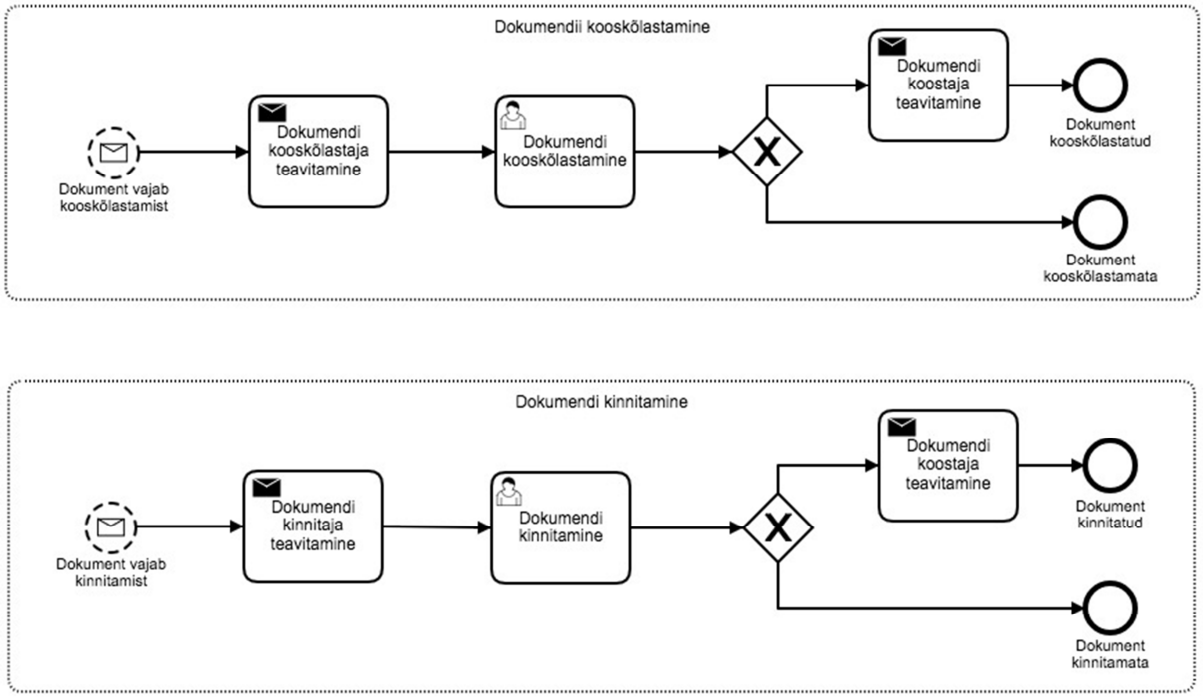
Joonis 1. Äriprotsessi algatamine

- kliendile pakkumise koostamise samm algatab juhtumipõhise äriprotsessi (vt joonis 2). Juhtumipõhine äriprotsess oli vajalik, kuna olenevalt olukorrast on liitumisspetsialistidel vaja täita kas üks või mitu dokumenti. Täidetavateks dokumentideks on:
    - tehniline tingimus;
    - tehniline lahendus;
    - leping;
    - kalkulatsioon;
    - projektülesanne.

  Lisaks dokumentide koostamisele peab olema võimalik lasta dokumente teistel Elektrilevi töötajatel üle vaadata. Ülevaatamine on märgitud infosüsteemis kui dokumendi kooskõlastamine ning kinnitamine (vt joonis 3);

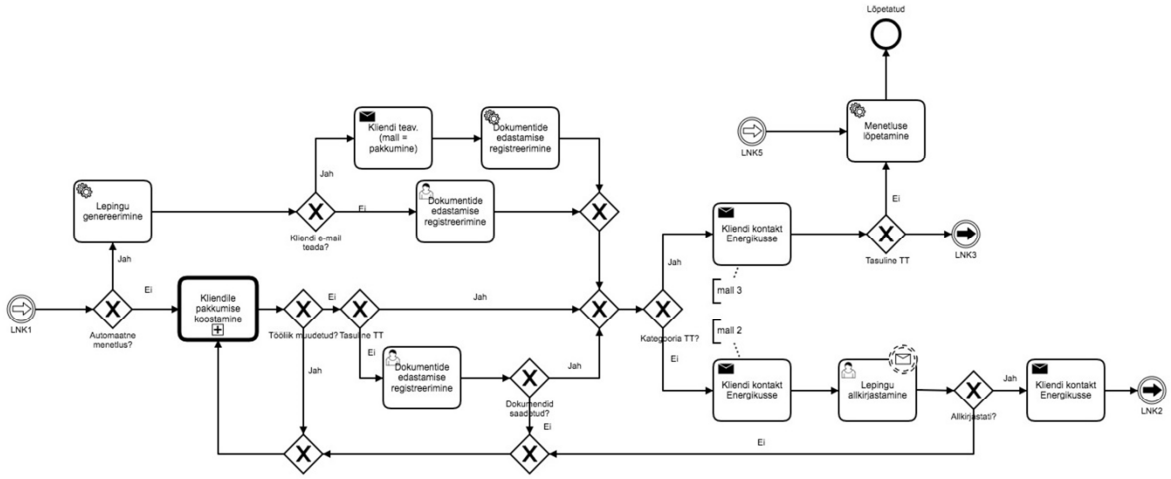


Joonis 2. Kliendile pakkumise koostamine

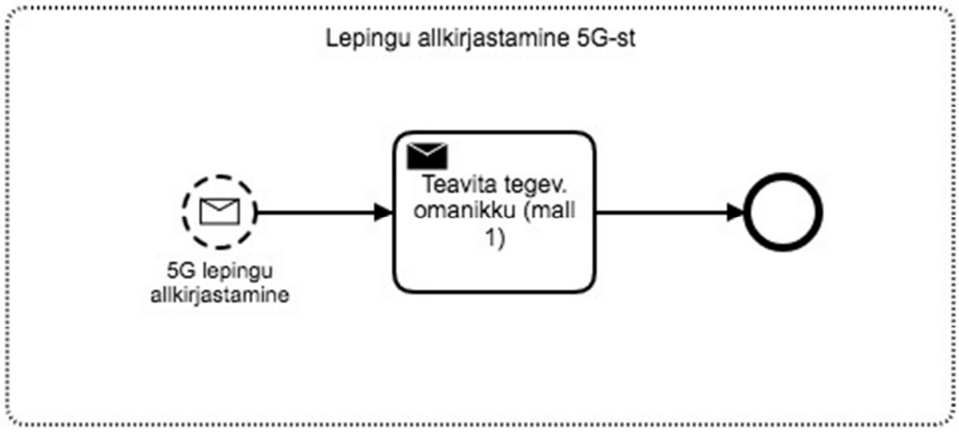


Joonis 3. Dokumentide kooskõlastamine ning kinnitamine

- peale kliendile pakkumise koostamise sammu registreeritakse kliendile saadetud dokumendid (vt joonis 4). Elektrilevi liitumiste infosüsteem ise kliendile dokumente ei saada. Dokumentide saatmine tuleb liitumisspetsialistil teha kas e-maili teel, posti teel või toimetatakse dokumendid kliendile isiklikult kätte;
- järgmiseks tähtsaks sammuks on lepingu allkirjastamine (vt joonis 4). Lepingut on võimalik allkirjastada nii Elektrilevi liitumiste infosüsteemis kui ka iseteeninduses (vt joonis 5);

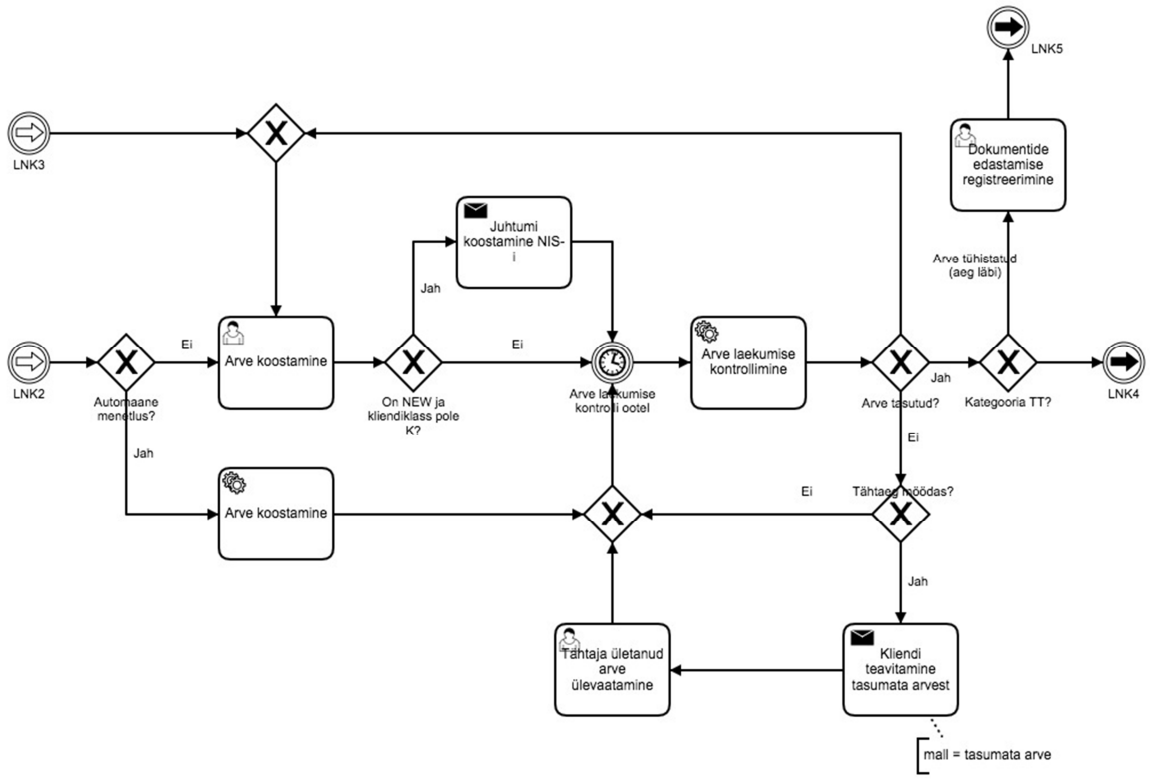


Joonis 4. Dokumentidega töötamine



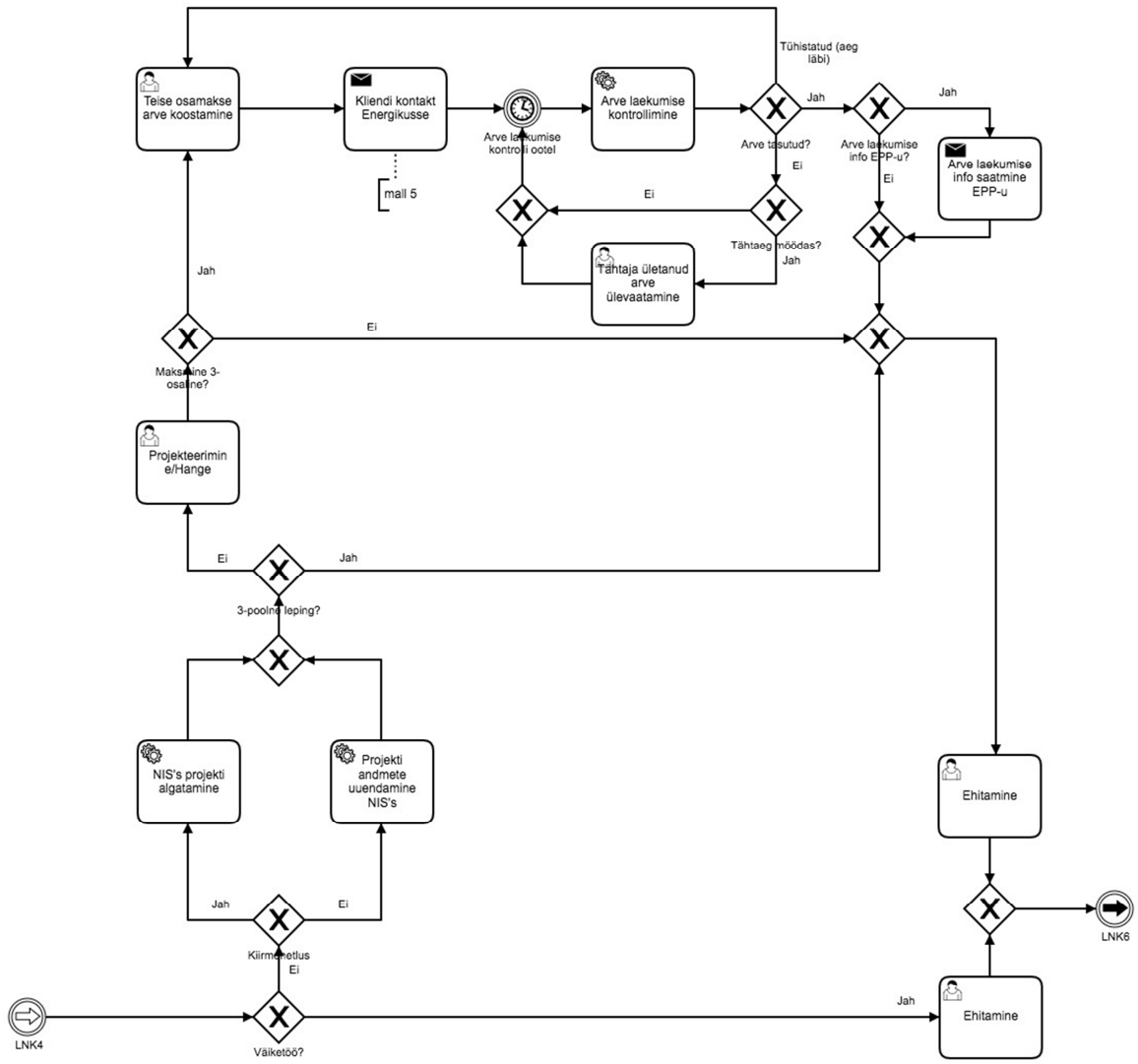
Joonis 5. Lepingu allkirjastamine iseteenindusest

- peale allkirjastamist toimub arve koostamine ning arve laekumise kontrollimine (vt joonis 6). Elektrilevi liitumiste infosüsteem ise arvetega ja arvete laekumistega ei tegele. Arvete koostamise ning laekumiste kontrollimise jaoks on Elektrilevil olemas teine infosüsteem nimega ENERGIK, kus on antud loogika juba olemas. Elektrilevi liitumiste infosüsteem paneb kokku arve koostamiseks vajalikud algandmed, millest tähtsaimad on: isik, projekt, summa ja taotlus. Seejärel saadab süsteem kokku pandud algandmed ENERGIK-usse. ENERGIK koostab etteantud andmetest arve ja tagastab unikaalse arve koodi, millega Elektrilevi liitumiste infosüsteem saab käia ENERGIK'st arve seisu küsimas. Peale arve täielikku maksmist liigub protsess järgmisesse sammu;

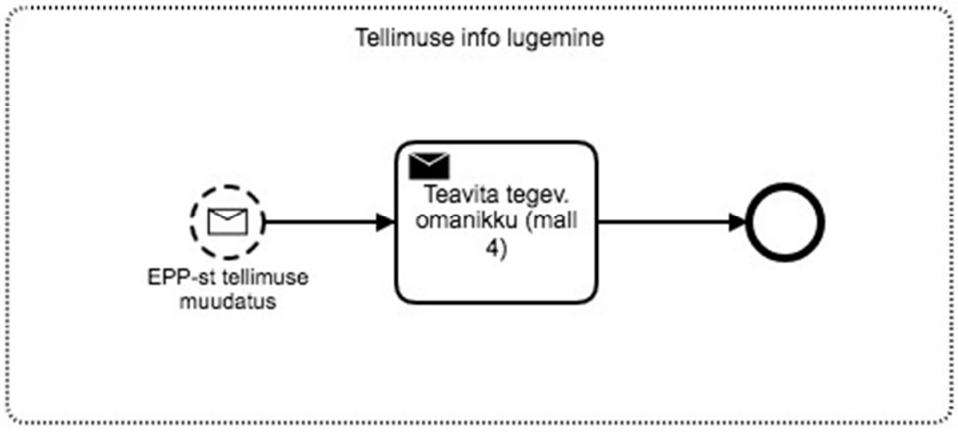


Joonis 6. Arve koostamine ja laekumine

- tellimuse koostamise sammus koostatakse tellimus ning seejärel oodatakse, millal EPP saadab projekti valmimise kuupäeva (vt joonis 7);
- peale EPP-ust tulnud projekti valmimise kuupäeva koostatakse teise osamakse arve ning käiakse laekumist ENERGIK-u infosüsteemis kontrollimas (vt joonis 7). Protsess toimib samamoodi nagu esimese osamakse korral;
- oodatakse, millal EPP saadab ehituse valmimise kuupäeva (vt joonis 7);

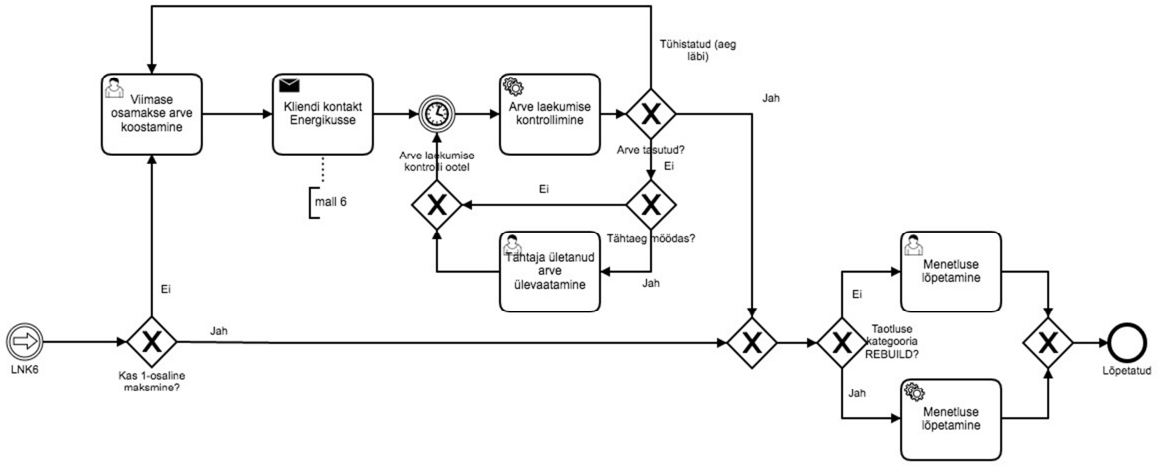


Joonis 7. Teine osamakse ning tellimus



Joonis 8. Tellimuse muudatus

- koostatakse kolmanda osamakse arve ning kontrollitakse laekumist. Protsess toimib identselt esimesele ja teisele osamaksele (vt joonis 9);
- äriprotsessi kõige viimane samm on menetluse lõpetamine, mille käigus muudetakse taotluse staatus lõpetatuks (vt joonis 9).



Joonis 9. Äriprotsessi lõpetamine

3.2.2. Äriprotsessis B1.3 funktsionaalse nõude implementeerimine

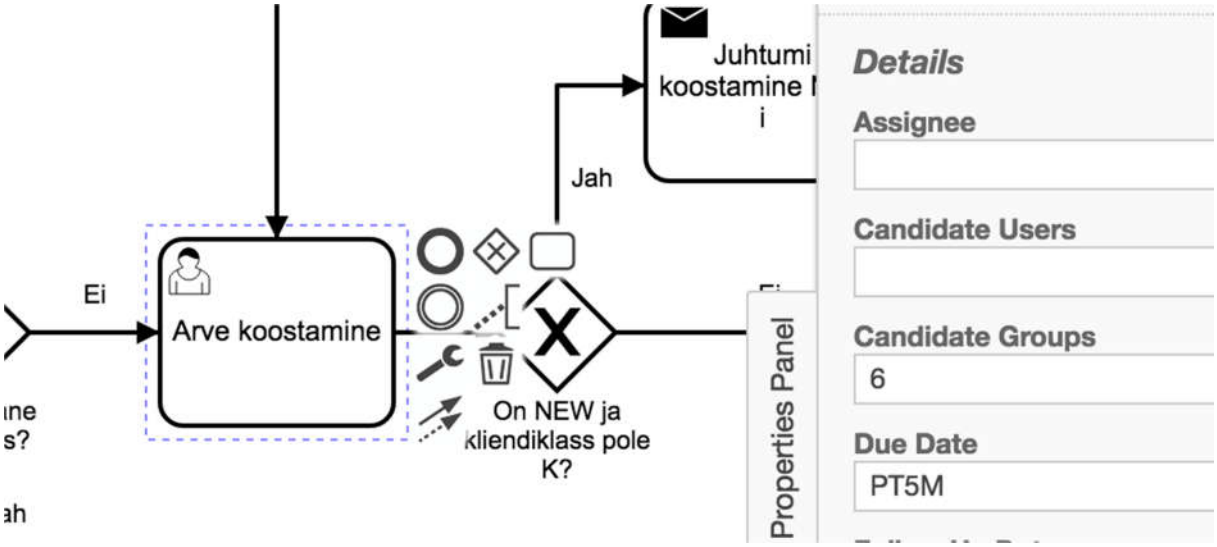
Elektrilevi liitumiste infosüsteemi funktsionaalne nõue B1.3 seadis äriprotsessile kohustuseks leida igale inimtegevusele enne aktiivseks muutumist töö tegija. Tegija on vaja leida inimtegevustele kolmel tasemel:

- protsessipõhine tegija. Protsessipõhine tegija on vajalik, kui liitumisspetsialisti rollis olev inimene on konkreetse taotlusega tegelenud ja soovib ka edaspidi sama taotlusega tegeleda ning protsess liigub sellisesse inimtegevuse sammu, kus on täitja rolliks liitumisspetsialisti roll. Tegevuse täitja leitakse protsessi unikaalse koodi ja rolli koodi järgi;
- tootepõhine tegija. Mõned tooted, mida on võimalik taotluse vormil valida on väga spetsiifilised ning nendega peavad tegelema kindla kompetentsiga inimesed. Tegevuse tegija leitakse toote unikaalse koodi ja rolli koodi järgi;

- piirkonnapõhine tegija. Juhul, kui protsessi ja toote järgi ei suudeta töö tegijat leida, on võimalus otsida tegija ARCOM veebirakendusest piirkonnapõhiselt. Tegevuse tegija leitakse piirkonna unikaalse koodi ja rolli koodi järgi.

Funktsionaalne nõue B1.3 seab kohustuseks igal juhul leida mõni inimene inimtegevustele tegijaks. Kui peaks juhtuma olukord, mil kedagi ei leita, ei tohi protsess edasi minna, vaid peab vea tõstatama ning seejärel teavitama kõiki peakasutaja rollis olevaid inimesi e-maili teel antud veast. Keegi peakasutajatest peab hakkama antud viga uurima ning välja selgitama, miks antud tegevusele täitjat ei leitud ja seejärel süsteemi vastavalt konfigureerima. Peale konfigureerimist peab peakasutaja rollis olev inimene tegevuse taaskäivitama ning vaatama, et tegevus jõuaks taaskordsel käivitamisel kindla inimese kätte.

Kõikidele inimtegevustele tuleb määrata kandidaatide grupp. Kandidaatide gruppe on võimalik lisada inimtegevustele kasutades Camunda Modeler-i (vt joonis 10). Juhul, kui kandidaatide gruppe peab olema rohkem kui üks, siis ei pea piirduma vaid ühe grupiga ja võib grupid eraldada komadega.



Joonis 10. Kandidaatide grupi lisamine

Elektrilevi liitumiste infosüsteemis on kokku 31 inimtegevust. Igale inimtegevusele ei oleks mõeldav panna külge kuularit, mis automaatselt tegevuse aktiveerimisel otsiks tegevusele täitjat. Kuigi saaks kasutada ühesugust kuularit, siis selline lahendus võib tahtmatult viia olukorrani, kus mõnele tegevusele on kogemata kuular ununenud külge panna. Selle tarbeks on võimalik defineerida ühtne kuular, mida ei pea iga inimtegevuse juures eraldi defineerima.

ELLI-s oli antud kuulari nimeks AutomaticTaskAssigningListener ning kuular käivitub automaatselt kõikide inimtegevuste loomise hetkel (vt joonis 11).

```
public class TaskBpmnParseListener extends AbstractBpmnParseListener {

    @Autowired
    private AutomaticTaskAssigningListener automaticTaskAssigningListener;

    @Override
    public void parseStartEvent(Element startEventElement, ScopeImpl scope, ActivityImpl activity) {}

    @Override
    public void parseEndEvent(Element startEventElement, ScopeImpl scope, ActivityImpl activity) {}

    @Override
    public void parseUserTask(Element userTaskElement, ScopeImpl scope, ActivityImpl activity) {
        TaskDefinition taskDefinition = ((UserTaskActivityBehavior) activity.getActivityBehavior()).getTaskDefinition();
        taskDefinition.addTaskListener(TaskListener.EVENTNAME_CREATE, automaticTaskAssigningListener);
    }

    @Override
    public void parseServiceTask(Element serviceTaskElement, ScopeImpl scope, ActivityImpl activity) {}

    @Override
    public void parseIntermediateCatchEvent(Element intermediateEventElement, ScopeImpl scope, ActivityImpl activity) {}
}
```

Joonis 11. Ühtne kuular äriprotsessi tegevustele

Hetkel TaskBpmnParseListener kuular ei oska veel AutomaticTaskAssigningListener-i välja kutsuda, kuna see tuleb lisada Camunda konfiguratsioonis äriprotsessi parsimise kuularina (vt joonis 12).

```
@Configuration
public class CamundaConfig {

    @Bean
    public SpringProcessEngineConfiguration processEngineConfiguration() {
        SpringProcessEngineConfiguration cfg = new SpringProcessEngineConfiguration();
        cfg.setCustomPostBPMNParseListeners(singletonList(taskBpmnParseListener()));

        return cfg;
    }

    @Bean
    public TaskBpmnParseListener taskBpmnParseListener() {
        return new TaskBpmnParseListener();
    }
}
```

Joonis 12. Camunda konfiguratsioon

Automaatselt väljakutsutava AutomaticTaskAssingingListener klassi sisuks on leida üles Camunda Modeler-is inimtegevusele lisatud kandidaatide grupid ning nende järgi leitakse loodavale tegevusele tegija (vt joonis 13). Camunda poolt pakutavat teenust TaskService kasutades on võimalik leitud kasutaja lisada käimasolevale inimtegevusele tegijaks.

```java
@Component
public class AutomaticTaskAssigningListener implements TaskListener {

    @Autowired
    private ElliTaskService elliTaskService;

    @Autowired
    private TaskService taskService;

    @Override
    public void notify(DelegateTask delegateTask) {
        List<String> groupIds = ((TaskEntity) delegateTask).getIdentityLinks().stream()
                .map(IdentityLink::getGroupId)
                .collect(toList());

        Eluser eluser = elliTaskService.findTaskUserByDelegateTask(delegateTask, groupIds);

        taskService.setAssignee(delegateTask.getId(), eluser.getUsername());
    }
}
```

Joonis 13. Inimtegevustele tegija lisamine

3.2.3. Äriprotsessis tekkinud probleemid

Üldiselt läks Elektrilevi liitumiste infosüsteemis äriprotsessi joonistamine ja implementeerimine ilma suuremate probleemideta. Äriprotsessi implementeerimise käigus tulid siiski välja paar takistust:

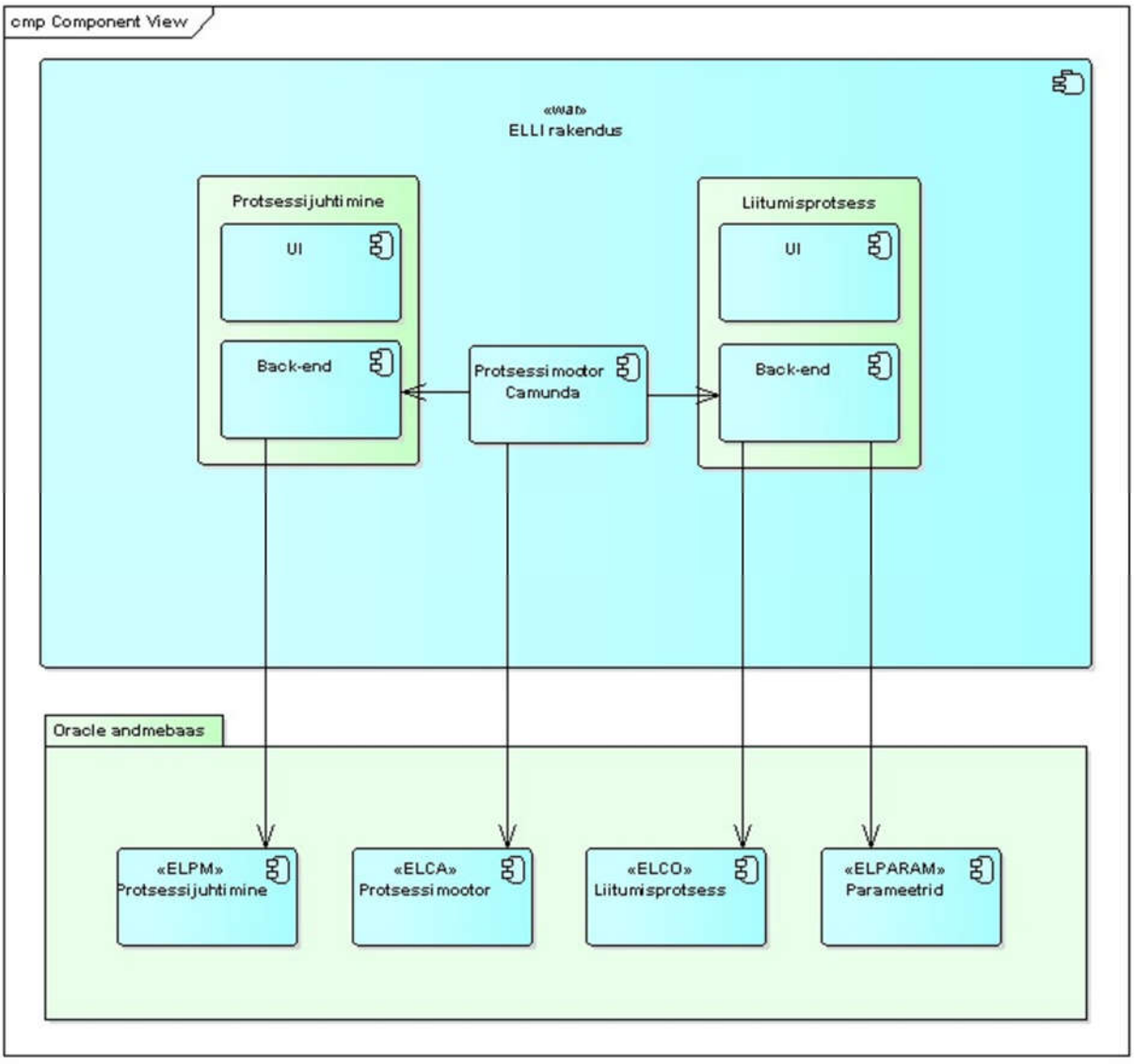
- kliendile pakkumise koostamine. Pakkumise koostamise sammus jõuab protsess kohta, kus käivitatakse juhtumipõhine mudel. Juhtumipõhine mudel andis arendusprotsessi palju juurde, kuid teisalt tõstis arenduse mahtu ja raskusastet. Juurde tuli võimalus eemaldada olemasolevaid dokumente või siis neid juurde lisades. Ühtlasi on olemas võimalus protsess varem ära seadistada juhuks, kui tulevikus peab lisama juurde uue dokumendi, mis hakkab x ajal tööle. Arendustööde raskust ja mahtu lisas ka see, et enam ei piisanud äriprotsessi jaoks loodud meetoditest, mis tegevust valideerivad ja lõpetavad. Tekkis vajadus uute meetodite järele, mis teeksid sisuliselt sama asja, kuid töötaksid juhtumipõhisel mudelil;
- lepingu allkirjastamine. Kui äriprotsess jõuab sammu “Lepingu allkirjastamine”, siis tekib liitumisspetsialistile inimtegevus, kus liitumisspetsialist saab märkida lepingu allkirjastatuks. Lisaks sellele võib klient lepingut ka iseteeninduses allkirjastada.

Iseteeninduses allkirjastamine peaks saatma käimasolevale inimtegevusele sündmuse, mis lõpetab antud tegevuse ning mille peale liigub äriprotsess järgmisesse sammu. Vaikimisi seadistustega protsess selliselt ei töötanud. Lõputöö autor lahendas antud olukorra otsides käimasoleva lepingu allkirjastamise tegevuse ja lõpetas antud tegevuse kasutades TaskService teenuse poolt pakutavat meetodit complete().

4. Arhitektuur

Arhitektuuri peatükis käsitletakse Elektrilevi liitumiste infosüsteemi komponentdiagrammi, Camunda protsessimootori arhitektuuri, andmebaasi ning integratsioone.

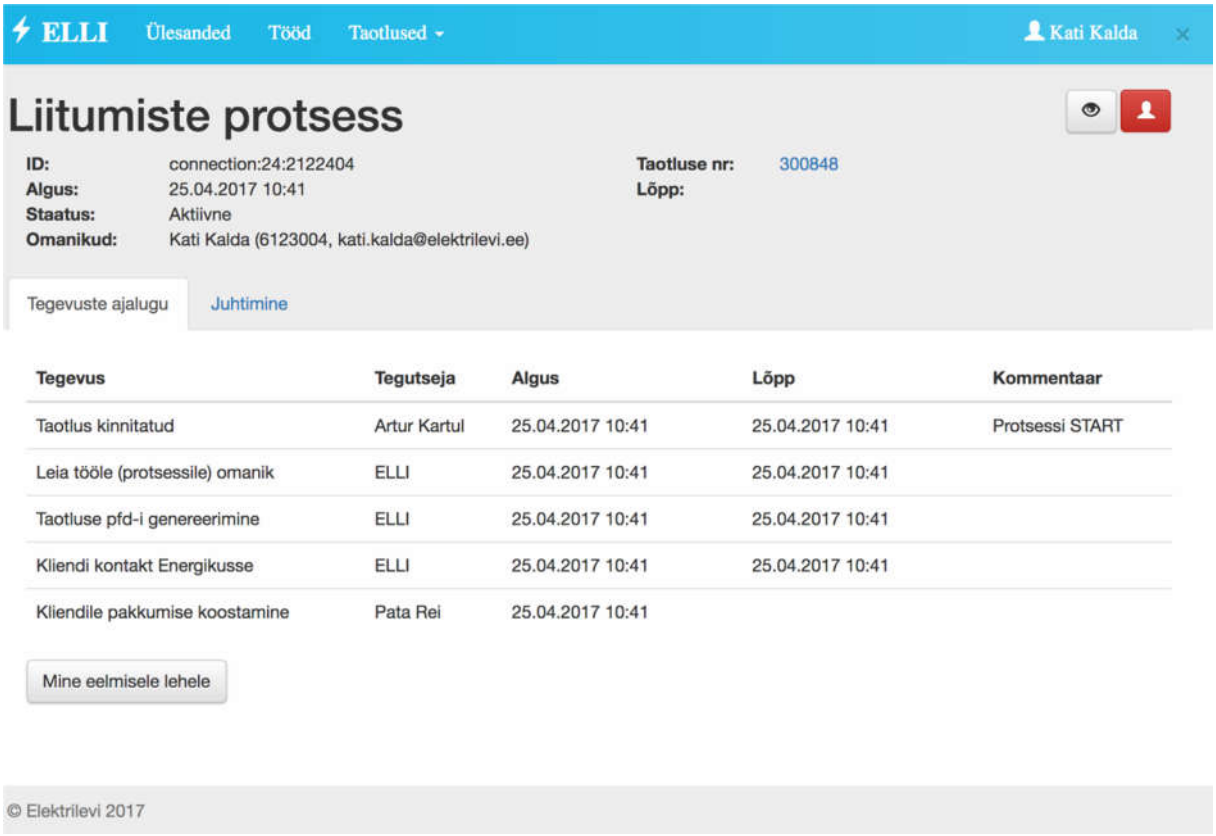
4.1. Komponentdiagramm



Joonis 14. Elektrilevi liitumiste infosüsteemi arhitektuur

Elektrilevi liitumiste infosüsteem on jagatud kolmeks suureks loogiliseks komponendiks, mille keskmeks on Camunda protsessimootor (vt joonis 14). Camunda protsessimootori ülesandeks on protsesside haldamine ning juhtimine. Lisaks Camunda protsessimootorile on olemas veel kaks eraldiseisvat komponenti, mis sisaldavad ka kasutajaliidest:

- liitumisprotsess. Liitumisprotsessi ülesandeks on taotluste, pakkumiste, arvete ning tellimuste koostamine ehk kõik see osa, mida Elektrilevi ametnikud oma töös tegema peavad;
- protsessijuhtimine. Protsessijuhtimise ülesandeks on täiendada Camunda protsessimootorit. Lisaks täiendamisele lisab see funktsionaalsust, mida Camunda protsessimootor ei võimalda või see funktsionaalsus on olemas vaid tasulises versioonis;
    - tegevuste taaskäivitamine. Camunda protsessimootor võimaldab veaga lõppenud tegevusi ükshaaval taaskäivitada. Kuna Elektrilevi liitumiste infosüsteem suhtleb viie erineva infosüsteemiga, siis võib tekkida probleem, et mõni teine infosüsteem ei vasta Seetõttu on oht, et väga paljud tegevused võivad veaga lõppeda. Administreerimine on üks suurimaid probleeme DWF keskkonnas, mida Elektrilevi liitumiste infosüsteem hakkab lahendama. Administreerimise vähendamiseks oli tarvis tekitada nupp, mis võimaldab korraga käivitada kõik vigadesse kukkunud tegevused.
    - tegevuste ajaloo vaatamine. Tasuta versioon Camunda protsessimootorist ei võimalda näha, mis sammud protsess on läbinud. Protsessijuhtimisse oli tarvis arendada lisafunktsionaalsus, mis annaks võimaluse näha, millal tegevust alustati, kes oli tegevuse täitjaks ning millal tegevus lõpetati. Ajaloo vaatamine võimaldab näha nii inimtegevusi, kui ka süsteemi poolt tehtavaid automaatseid tegevusi (vt joonis 15).



Joonis 15. Tegevuste ajaloo vaatamine

Elektrilevi liitumiste infosüsteem on kirjutatud Java 8-s ning kasutatud on Spring Boot raamistikku. Kuna Elektrilevi kõik teised infosüsteemid kasutavad Oracle andmebaasi siis seda teeb ka uus Elektrilevi liitumiste infosüsteem. Kogu Elektrilevi liitumiste infosüsteem ehitatakse Gradle-ga ühtseks war pakiks, mis käivitatakse Tomcat serveri peal.

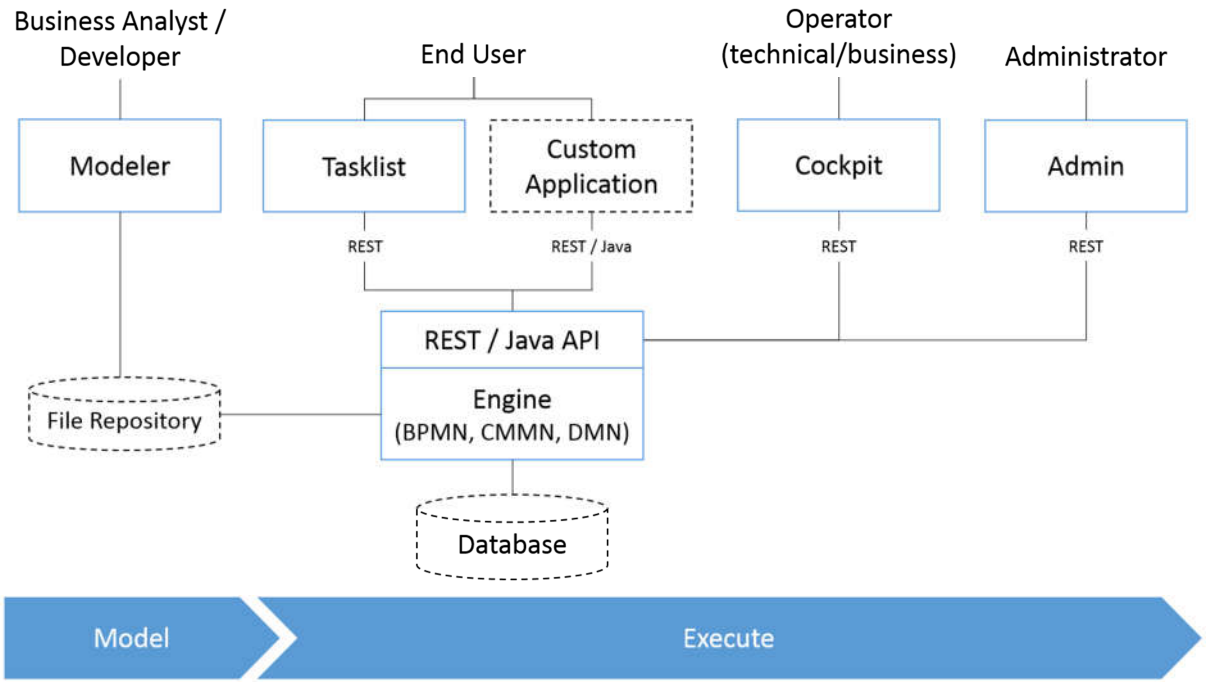
4.2. Camunda arhitektuur

Camunda on Java-s kirjutatud avatud lähtekoodiga protsessimootor. Esimene stabiilne versioon Camunda-st tuli 2012 aasta algul ning kuna Camundat arendatakse jätkuvalt edasi siis lõputöö kirjutamise hetkel on stabiilseks versiooniks juba 7.6 [2].

Camunda-st on olemas kaks versiooni, millest üks on tasuline ja teine on tasuta. Tasuline versioon annab juurde võimaluse suhelda 24/7 Camunda klienditoega, kes pakuvad enda poolset abi ja nõu. Lisaks abile ja nõule on ka protsessi administraatori liideses olemas

erinevaid lisasid. Lisadest põnevaimad ja huvipakkuvaimad on protsessi kuumakaart ja ajaloo vaade. Kuumakaardi vaates on protsessile peale joonistatud värvid, et oleks selgesti näha, kus protsess toppama jääb, et saaks parandada protsessi läbimise kiirust. Ajaloo vaates saab minna ajas tagasi ning seeläbi uurida, kus protsess teatud ajal oli [3]. Antud lõputöös kirjeldatud projektis oli kasutusel vabavaraline versioon Camundast, sest protsessimootori üks kriteeriumitest oli see, et protsessimootor peab olema vabavaraline. Kuigi on valitud Camunda vabavaraline versioon, kus ei ole lisasid, täidab see siiski kõik funktsionaalsed nõuded.

Camunda on protsessimootor, mis vajab töötamiseks sisendina äriprotsessi mudelit (edaspidi BPMN), juhtumipõhist äriprotsessi mudelit (edaspidi CMMN) või otsustusmudelit (edaspidi DMN) (vt joonis 16). Kõiki eelmainitud mudeleid saab luua Camunda Modeler-i abil. Camunda Modeler on eraldiseisev rakendus, mis on mõeldud vaid mudelite joonistamiseks ja defineerimiseks. Antud joonistusvahendit kasutades on võimalik protsessile läbi kasutajaliidese ette anda erinevaid parameetreid, mis mõjutavad protsessi edasist käiku. Nendeks parameetriteks võivad olla viited mõnele Java klassile, mida mõni teenus peab välja kutsuma, kui ka konfiguratsioon taimerile. Kõike seda on võimalik kirjutada ka otse XML-i, kuid üldjuhul kiirendab graafiline liides töö kulgu ning annab protsessist parema ülevaate.



Joonis 16. Camunda protsessimootori arhitektuur

Lisaks vajab Camunda oma tööks Postgre või Oracle andmebaasi. Andmebaasis hoitakse kõike protsessiga seotut ning sealhulgas ka protsessi ajalugu. Põhilisemad protsessiga seotud andmed, mida hoitakse andmebaasis on:

- tegevuste tüübid ning nende alguse ja lõpu ajad. Kui tegemist on kasutaja tegevusega siis ka täitja kasutajanimi;
- muutujad. Äriprotsess liigub edasi tänu muutujatele, tänu milledele leitakse sobiv teekond protsessis, siis ka kõik muutujad koos oma väärtustega on salvestatud andmebaasi;
- versioonid. Ajas võivad äriprotsessid muutuda. Selle tarbeks hoitakse kõiki versioone andmebaasis, et oleks aru saada, mis äriprotsessi versioonil töö käib.

Camunda protsessimootor pakub kolme veebirakendust. Kõik veebirakendused pöörduvad protsessimootori poole REST API teenuste kaudu. Pakutavad veebirakendused on:

- kokpiti veebirakendus. Kokpiti veebirakendus on mõeldud kõikide käimasolevate protsesside monitoorimiseks ning administreerimiseks. Veebirakenduses näeb ära protsesside staatused ning visuaalselt ka koha, kus protsess parajasti on. Juhul, kui protsess peaks saama kusagil tehnilise vea, siis protsess jääb seisma ning kokpitis on näha vea põhjus. Peale vea parandamist on võimalik vea saanud tegevus taaskäivitada. Näiteks, kui Elektrilevi liitumiste infosüsteemis jõuab protsess kasutaja tegevuse sammu, siis peab süsteem leidma alati tegevusele täitja. Kui juhtub, et tegevusele jääb täitja leidmata, siis on kohustus tõstatada viga, mida saab peale konfiguratsiooni muutmist taaskäivitada. Kokpit veebirakendus on Elektrilevi liitumiste infosüsteemis täielikult kasutusel;
- tööde nimekirja veebirakendus. Tööde nimekirja veebirakendus on mõeldud kasutajatele tegevustega töötamiseks. Kasutajal on olemas töölaud, kus ta näeb kõiki töid, mida võib endale võtta ja tegema asuda. Tööde nimekiri saadakse kasutaja töölauale eeldefineeritud filtri abil. Lisaks vormide täitmisele on võimalik tegevusi edasi suunata ning määrata tegevustele täitmise aega. Elektrilevi liitumiste infosüsteem ei kasuta Camunda pakutavat tööde nimekirja veebirakendust, kuna see ei vastanud kõikidele funktsionaalsetele nõuetele. Elektrilevi liitumisspetsialistid peavad koguaeg nägema kõiki taotluste andmeid, dokumente, manuseid ning arveid ja

tellimusi. Tööde nimekirja veebirakenduses oleks sellise võimaluse tekitamine olnud liialt keeruline, mistõttu tuli otsus teha eraldiseisev rakendus;

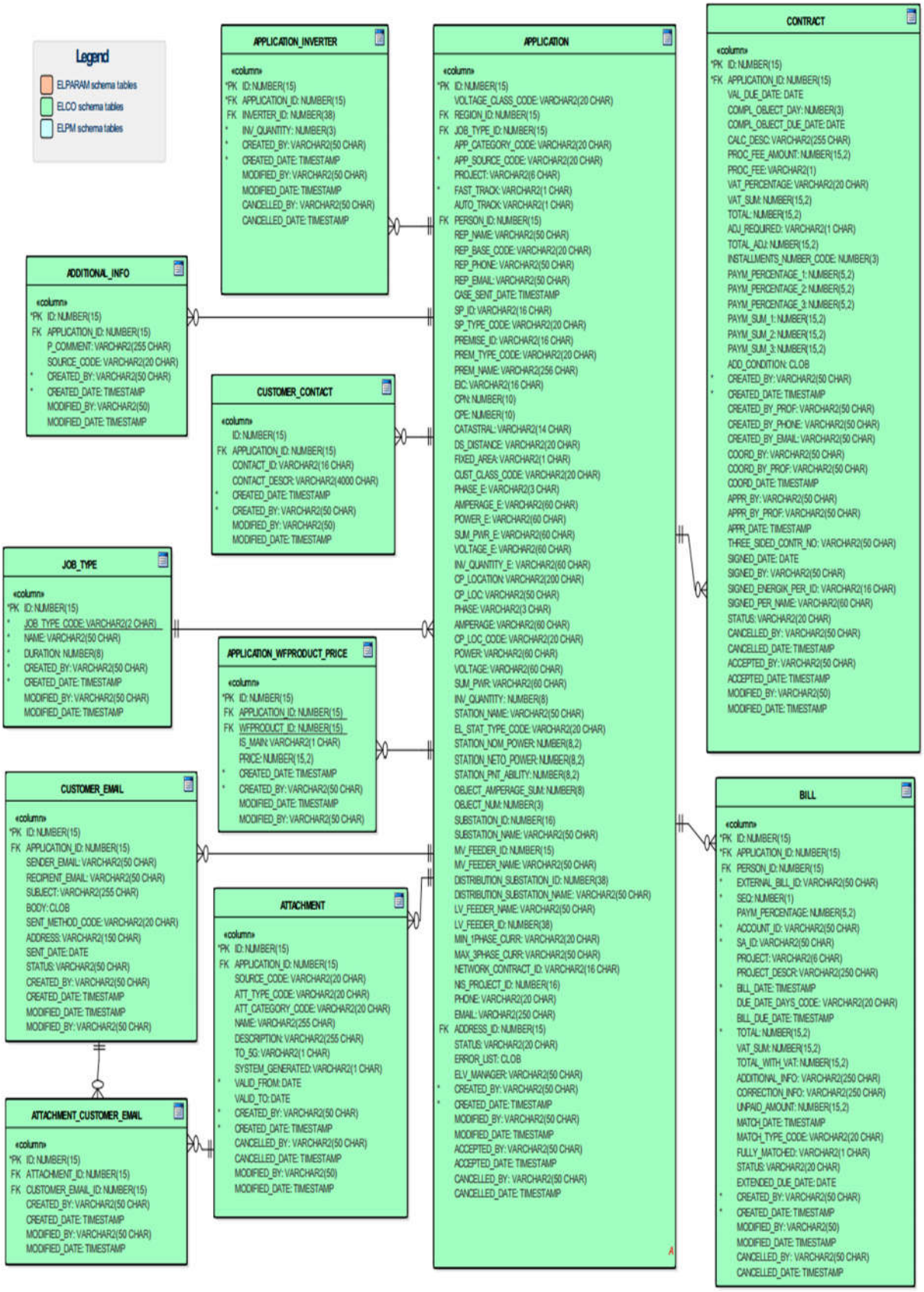
- eritellimusel veebirakendus. Elektrilevi liitumiste infosüsteem on eritellimusel valminud veebirakendus Camunda tööde nimekirja veebirakendusest. Elektrilevi liitumiste infosüsteem suhtleb samuti üle REST API teenuste Camunda protsessimootoriga;
- admininistraatori veebirakendus. Administraatori veebirakenduses saab hallata kasutajaid, kasutajagruppe ning õiguseid. Õiguseid saab anda nii kasutajatele, kui ka kasutajagruppidele. Õigused määravad veebirakendustele ligipääsud ning lisaks eraldi ka protsessid, millega on võimalik töötada. Üldiselt on õigused mõeldud tööde nimekirja veebirakendusele. Administraatori veebirakendus on Elektrilevi liitumiste infosüsteemis kasutusel, et määrata kasutajaid, kes saavad kokpit veebirakendust kasutada [2].

4.3. Andmemudel

Elektrilevi liitumiste infosüsteem vajab oma tööks Oracle andmebaasi. Antud otsus sündis, kuna Elektrilevil on Oracle andmebaaside haldamiseks vajalikud spetsialistid olemas ning ka Elektrilevi teised süsteemid on kõik Oracle peal.

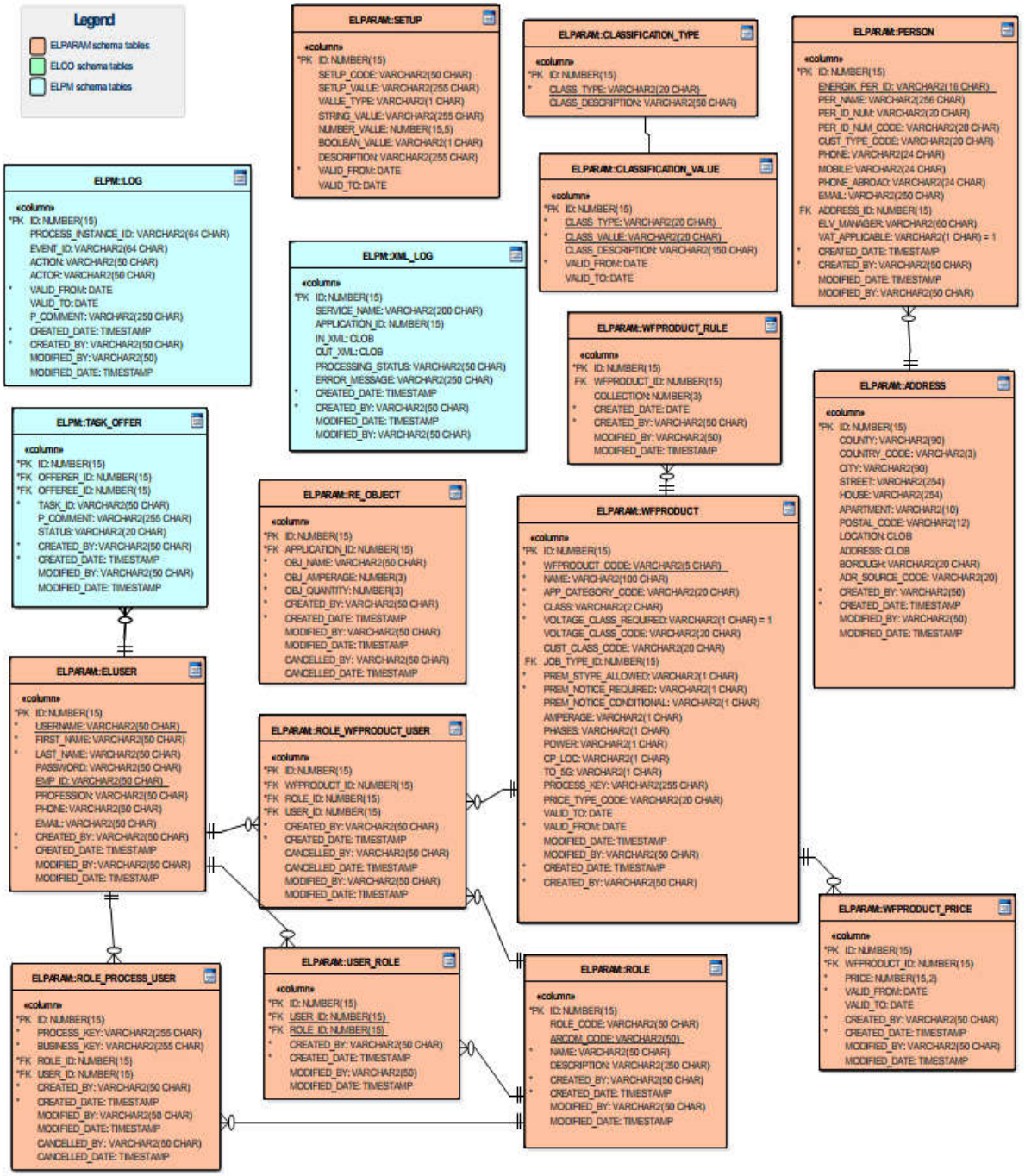
Kokku on vaja nelja kasutajat, kus hoitakse erinevat laadi infot:

- ELCA. Kõik vajalikud tabelid Camunda protsessimootori tööks;
- ELCO. Kõik tabelid liitumisprotsessi tarbeks. Hoitakse kõike infot taotluste, tellimuste, arvete ning lepingute kohta. ELCO kasutajal peab olema juurdepääs teiste kolme kasutajate tabelitele, et sealt lugeda ning sinna kirjutada uut infot, kuna rakendus pannakse käima ELCO kasutajaga (vt joonis 17);



Joonis 17. ELCO andmebaas

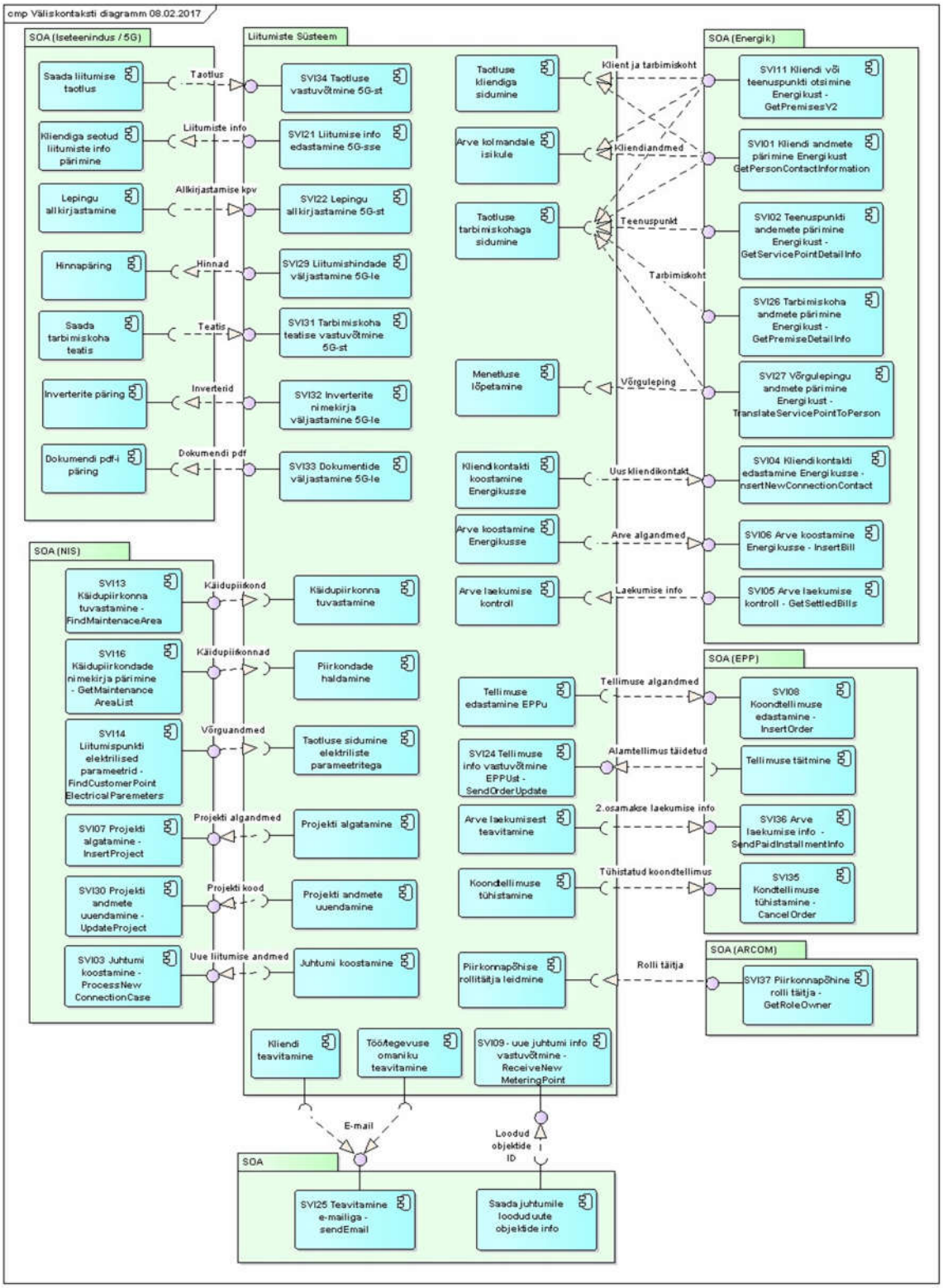
- ELPM. Kõik tabelid protsessijuhtimise jaoks. Selles hoitakse näiteks tegevuste ajalugu (vt joonis 18);
- ELPARAM. Kõik üleüldised andmed, mida liitumisprotsess oma tööks vajab. Hoitakse infot kasutajate ja rollide kohta. Lisaks on klassifikaatorid, tooted, mõned rakenduse seadistused ja ka kirjade mallid (vt joonis 18).



Joonis 18. ELPM ja ELPARAM andmebaas

4.4. Integratsioonid

Elektrilevi liitumiste infosüsteem on seotud viie teise Elektrilevi infosüsteemiga nii teenuseid tarbides, kui ka pakkudes (vt joonis 19). Suhtlus infosüsteemide vahel käib üle SOAP-i. Seotud infosüsteemideks on (vt lisa 3):



Joonis 19. Integratsioonid

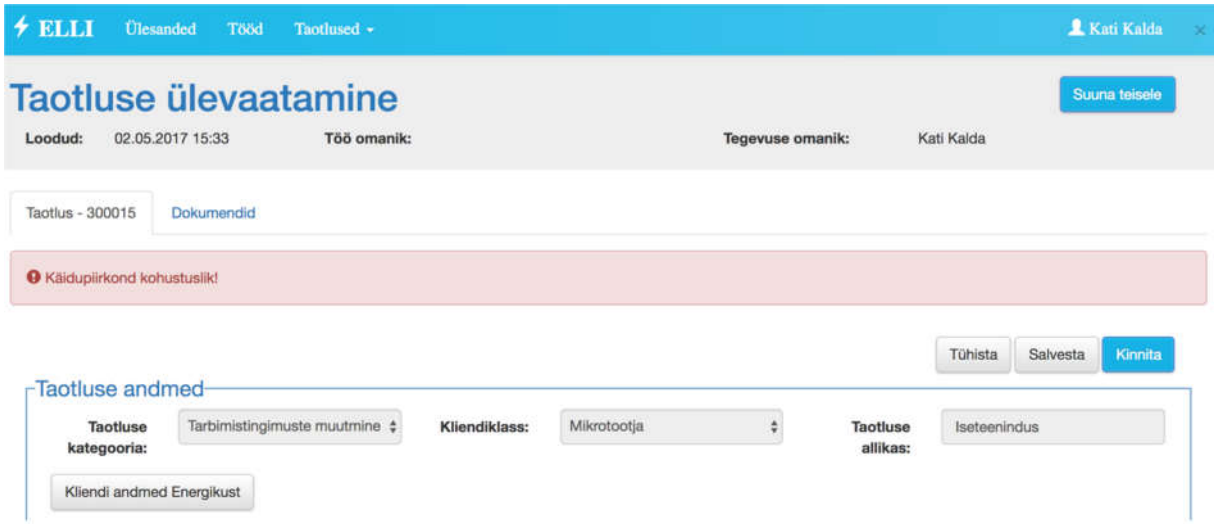
- EPP, mis on Elektrilevi tellimuste infosüsteem;
- NIS, mis on karbitoode ning milles on kirjeldatud füüsiline võrk;

- 5G Iseteenindus;
- Energik, mis on Eesti Energia kliendiinfo ja arveldussüsteem;
- ARCOM, mis hoiab infot töötajate ja käidupiirkondade kohta.

Elektrilevi liitumiste infosüsteem pakub teistele Elektrilevi infosüsteemidele ka omapoolseid teenuseid, millest enamusi kasutab Iseteenindus (vt lisa 4). Samuti on ka suurima teenuse kasutajaks Iseteenindus, milleks on taotluste vastuvõtmine.

Taotluste vastuvõtmise teenus võtab vastu kõik taotlusega seotud andmed ning selle korral on kasutusel kolmekordne valideerimine:

- esimese astme valideerimise peavad läbima kõik Iseteenindusest tulnud taotlused ning selle eesmärgiks on kontrollida, et kas minimaalne hulk kohustuslikke välju on täidetud. Antud kontrollid on ka Iseteeninduse poolel tehtud, et sealt ei tuleks taotluseid, millega ei oleks hiljem midagi teha. Turvalisuse kaalutlusel on need ka Elektrilevi liitumiste infosüsteemi poolel implementeeritud. Juhul, kui tekib mõni viga, siis taotlust ei salvestata ning äriprotsessi ei algatata, vaid antakse juhtunust täpselt Iseteenindusele teada;
- teise astme valideerimiseni jõuavad kõik taotlused, mis on edukalt läbinud esimese astme. Teise astme eesmärgiks on taotlust valideerida, et kas taotlusel on mõned kohustuslikud andmed puudu. Juhul, kui leitakse puuduvaid välju siis kuvatakse kõik tekkinud vead taotlusel, et liitumisspetsialistil oleks kiire ülevaade vigadest (vt joonis 20). Vastasel juhul peaks liitumisspetsialist vajutama „Kinnita“ nuppu, et vigu näha. Olenemata, kas vigu oli või mitte salvestatakse taotlus alati ära ning käivitatakse äriprotsess;



Joonis 20. Taotluse ülevaatamine

- kolmanda astme valideerimisse jõuavad kõik taotlused, mis on edukalt läbinud nii esimese, kui ka teise astme. Kolmas aste teeb vajalikud valideerimised leidmaks, kas Iseteenindusest tulnud taotlus saab olla kiir- või automaatmenetlus.

Kokkuvõte

Bakalaureusetöö kirjeldab Elektrilevi äriprotsesside üle viimist Camunda protsessimootorile ehk Elektrilevi Liitumiste infosüsteemi arendamise projekti, milles töö autor osales kui süsteemi arhitekt ning arendaja. Elektrilevi liitumiste infosüsteemi olulistemaks eesmärkideks on Elektrilevi teenindusosakonna ametnike töö kiirendamine ja lihtsustamine ning pikemas perspektiivis teeninduskvaliteedi tõstmine Elektirlevi klientidele.

Bakalaureusetöö analüütilises osas on kirjeldatud funktsionaalsed ja mittefunktsionaalsed nõuded ning Elektrileviga liitumise äriprotsess. Töö autor andis ülevaate Elektrileviga liitumise äriprotsessi tähtsamatest sammudest ning kitsaskohtadest ning kirjeldas, kuidas töö käigus tekkinud probleemid lahendati.

Bakalaureusetöö praktilises on loodud infosüsteemi arhitektuuri kirjeldus ning ülevaade projektis kasutataud vabavaralise protsessimootori Camunda enda arhitektuurist. Töö autor toob välja Elektrilevi liitumiste infosüsteemis olevad integratsioonid teiste infosüsteemidega ning kirjeldab neid. Praktilises osas on kajastatud ka andmemudel ja komponentdiargamm.

Camunda platvormil oleva protsessimootori rakendamisel Elektrilevi äriprotsessides oli autoril kõige problemaatilisem juhtumipõhine mudel ühendada äriprotsessi mudeliga, mis tekitas kõvasti lisatööd juurde.

Bakalaureusetöös välja toodud ärilisi ja arhitektuurilisi kirjeldusi saab aluseks ning abiks võtta ka teiste äriprotsesside Camunda protsessimootorile üleviimise tarbeks. Samuti saab loodud Elektrilevi liitumiste infosüsteemi lihtsasti arendada juurde funktsionaalsust, mis on eelnevalt skoobist välja jäänud.

Bakalaureusetöö eesmärk, mis oli rakendada Elektrilevi äriprotsessides Camunda protsessimootorit sai täidetud. Töös kirjeldatud arhitektuuri, andmemudelite ning lahenduste alusel käis Icefire OÜ’s süsteemi arendus. Süsteemi lisatööde arendamine ning projekti juurutusjärgse toe pakkumine tellijale olid samuti diplomitöö autori ülesandeks. Projekt on käesolevaks momendiks antud kliendile testimiseks ning süsteemi rakendamine Elektrilevis on hinnanguliselt 17. mai 2017. aastal.

Kasutatud kirjandus

1. Elektrilevi tutvustus. (kuupäev puudub). Elektrilevi. Loetud aadressil: https://www.elektrilevi.ee/et/elektrilevi-tutvustus

2. Core Engine. (kuupäev puudub). Camunda. Loetud aadressil: https://camunda.org/features/engine/

3. Camunda BPM Editions. (kuupäev puudub). Camunda. Loetud aadressil: https://camunda.com/bpm/enterprise/

Summary

Title: Camunda Business Process Model Engine Implementation on the Example of Elektrilevi.

The Bachelor Thesis describes the implementation of the Camunda business procces model engine to Elektrilevi business processes and the development of the Elektrilevi Connections information system. The author participated in the project as the system architect and system developer. The aim of the Elektrilevi Connections information system is mainly to expedite and simplify the work of officials of the service department of Elektrilevi and over the long-term to raise the quality of service for Elektrilevi clients.

In the analytical part of the thesis, the author describes the functional and non-functional requirements, and the business processes of the clients connecting to Elektrilevi. The author gives an overview of the major stages and bottlenecks of the business process of connecting to Elektrilevi, and explains the solutions for problems which occurred during the work.

The practical part of the paper contains a description of the architecture of the created Elektrilevi connections information system and an overview of the architecture of Camunda – the free process model engine used in the project. The author shows and describes integrations with other information systems within the information system of Elektrilevi Connections. In addition, the practical part includes a data model and a component diagram.

The most problematic task for the author in implementing the existing businessprocesses to Camunda business process model engine was connecting the case management model with the business process model, because it involved a lot of extra work.

The business analysis and architectural solutions described by the author can also be used for implementing other business processes onto Camunda process model engine. Further functionality, that was excluded due to narrowing of the scope, can be easily added in the Elektrilevi connections information systems.

The ojective of the Bachelor's Thesis – to implement Camunda process engine in Elektrilevi business processes – was accomplished. The Elektrilevi connections information system development was performed using the architecture, data models and solutions described in the

thesis. The author was also tasked with developing extra functionality within the system and offering the project follow-up support to the customer. The project has been delivered to the client for testing and the date of system implementation at the Elektrilevi is estimated to be 17 May 2017.

Lisad

Lisa 1. Funktsionaalsed nõuded

B 1.1. - Kasutaja peab saama süsteemis defineerida erinevaid protsesse;

F 1.1.1. - Protsessile peab saama määrata unikaalse nimetuse. Antud nõue on oluline, et oleks võimalus antud infosüsteemile lisada uusi äriprotsesse;

F 1.1.2. - Protsessile peab saama lisada kirjeldust, millist töövoogu protsessiga juhitakse;

F 1.1.3. - Protsessiga peab saama defineerida rolli, millega defineeritakse süsteemi töö omanikud;

F 1.1.4. - Protsessile peab saama lisada samme. Protsessis peab olema üks alustav samm ja vähemalt üks lõpetav samm;

F 1.1.4.1. - Sammule peab saama defineerida unikaalset koodi ja nimetust;

F 1.1.4.2. - Peab saama määrata sammu tüübi, kas samm on täitja samm või sammu täidab programm;

F 1.1.4.3. - Sammule peab saama defineerida sammu täitmise juhise;

F 1.1.4.4. - Täitja sammule peab saama määrata sammu täitja rolli;

F 1.1.4.4.1. - Kui täitja roll sõltub töö liigist, peab saama defineerida tööliigi põhised täitja rollid;

F 1.1.4.5. - Sammu täitmiseks peab saama defineerida tähtaegu (päevade arv sammu täitmiseks, päevade arv sammu täitmisse võtmiseks);

F 1.1.4.6. - Peab saama defineerida reegleid meeldetuletuste saatmiseks tähtajast kinnipidamiseks (päevade arv meeltuletuse saatmiseks tähtaja saabumisel/möödumisel, päevade arv, mille järel meeldetuletust korratakse);

F 1.1.4.7. - Peab saama defineerida, kas antud sammu lõpetamisel genereeritakse Energikusse kliendikontakt;

F 1.1.4.8. - Sammule peab saama defineerida sammu lõpetamise otsuseid;

F 1.1.4.8.1. - Sammu lõpetamise otsus peab olema seotud protsessi sammuga, millesse töö järgnevalt suunatakse;

B 1.2. - Kasutaja peab saama süsteemis defineerida erinevaid töös osalemise rolle, protsessi tooteid ja piirkondi;

B 1.3. - Kasutaja peab saama defineerida rollide täitjaid. Rollide täitajaid peab saama defineerida piirkonnapõhiselt, protsessi toote põhiselt ja töö põhiselt;

F 1.3.1. - Tegevuse täitjate määramisel peab süsteem juhinduma tööle/sammule määratud rollist ning leidma rollipõhised täitjad, mis on defineeritud:

- tööle;
- protsessitootele, kui tööle pole defineeritud;
- töö teostamise piirkonnale, kui protsessitootele pole defineeritud.

F 1.3.2. - Võimalusel peab süsteem asendama töötajat puhkusel viibimisel või kui töötaja on endale määranud asendaja töölt eemal viibimise ajaks;

B 1.4. - Kasutaja peab saama algatada tööd, mis põhineb süsteemis defineeritud protsessil;

B 1.5. - Süsteem peab töö algatamisel määrama tööle omaniku(d) ja aktiveerima tegevuse, mis on määratud protsessikirjelduses alustamise sammuga;

F 1.5.1. - Töö omanik peab saama end eemaldada konkreetselt töölt töö omanike loetelust, kui ta ei ole ainuke töö omanik;

F 1.5.2. - Õigustes kasutaja peab saama määrata end mingile konkreetsele tööle töö omanikuks;

B 1.6. - Tegevuse täitmisel peab süsteem saatma välja meeldetuletusi tegevuse lõpetamiseks vastavalt sammule täitmiseks määratud tähtajale;

B 1.7. - Tegevuse lõpetamisel peab süsteem kontrollima tegevuse lõpetamise eeltingimusi ja mitte lubama tegevus lõpetada, kui tingimused on täitmata;

B 1.8. - Süsteem peab suutma tegevuse lõpetamisel suunama töö parameetrite väärtusest automaatselt järgmisse tegevusse;

B 1.9. - Töö käigus peab õigustes kasutaja saama tööd suunata teise täitaja tegevusse;

B 1.10. - Õigustes kasutaja peab saama suunata töö tegevuse täitmise mõnele teisele kasutajale;

B 1.11. - Õigustes kasutaja peab saama tööd peatada kas tähaajaliselt või tähtajatult;

F 1.11.1. - Kui peatatakse tegevust, millele on määratud täitmise tähtaeg, siis nihkub see peatatud aja võrra edasi;

B 1.12. - Õigustes kasutaja peab saama tööd tühistada;

B 1.13. - Kõik kasutajad peavad saama määrata end mingi töö jälgijaks;

B 1.14. - Kõik kasutajad peavad saama lehitseda töös olevaid töid, mille omanikeks nad on, mille tegevuse täitjaks nad on või mille jälgijaks nad on;

B 1.15. - Tegevuste täitmine peab olema tegevuste ja täitjate kaupa ajaliselt mõõdetav.

Lisa 2. Mittefunktsionaalsed nõuded

- turvalisus
    - autentimine peab toimuma Eesti Energia tsentraalses LDAP teenuses;
    - autentimine peab sisaldama SSO funktsionaalsust;
    - õiguseid peab saama anda rolli põhiselt;
    - rakendus peab lõppkasutajale näitama korrektseid veateateid, mis ei tohi sisaldada tehnilist infot;
    - sessiooni pikkust peab saama muuta;
    - rakendus ei tohi sisaldada koodi väljaspoolt Eesti Energiat;
- rakenduse pakendamine ja installeerimine
    - rakendus peab töötama Tomcat serveri peal;
    - kui rakendus on vaja enne Tomcat-i peal käivitamist kompileerida, siis peab ehitamise skript olema rakendusega koos;
    - kui rakendus nõuab sõltuvusi, siis kõik sõltuvused peavad tulema läbi Eesti Energia repositooriumi;
    - kõik SQL skriptid peavad olema UTF-8;
    - kõik skriptid peavad olema lisatud repositooriumisse;
    - rakendus peab kasutama serveri aega;
    - rakenduse versiooni number peab sisaldama SVN revisiooni numbrit;
    - rakenduse versiooni number peab olema kujul X.Y.Z.R, kus
        - X - suuremad rakenduse funktsionaalsed ja/või andmebaasi muudatused;
        - Y - väiksemad rakenduse funktsionaalsed muudatused. Ei tohi sisaldada andmebaasi muudatusi;
        - Z - kasutajaliidese muudatused;
        - R - SVN revisiooni numbrid;
- konfiguratsioon
    - rakenduse uuendamine ei tohi üle kirjutada eelmist konfiguratsiooni;
    - failisüsteemi kõik asukohad peavad olema konfigureeritavad;
    - kõik LDAP ühenduse parameetrid peavad olema konfigureeritavad;

    - muudatused rakenduse konfiguratsioonis ei tohi nõuda rakenduse kompileerimist;
- logimine
    - rakendus peab logima kõiki tehnilisi vigu;
    - logi teadete ajaformaat peab olema muudetav;
    - juhul, kui logi faili või kausta ei eksisteeri peab rakendus need iseseisvalt suutma tekitada;
- monitoorimine
    - rakendus peab näitama oma staatust JSON formaadis;
    - monitooringu lehe laadimine ei tohi olla pikem, kui 20 sekundit;
- andmebaas
    - kõik andmebaasi objektid peavad olema inglise keeles;
    - kõik tabelid peavad sisaldama kirje loomise ja muutmise välju;

Lisa 3. Integratsioonid

Elektrilevi liitumiste infosüsteem kasutab oma tööks jägnevaid teenuseid:

- EPP - EPP saadab tellimuse info uuendamise ELLI-le. ELLI kasutab järgmisi EPP-u teenuseid:
    - SVI08 (InsertOrder) - Koondtellimuse edastamine;
    - SVI35 (CancelOrder) - Koondtellimuse tühistamine;
    - SVI36 (SendPaidInstallmentInfo) - 2. osamakselaekumise info edastamine;
- NIS - NIS ise ei kasuta ELLI poolt pakutavaid teenuseid. ELLI kasutab järgmisi NIS teenuseid:
    - SVI13 (FindMaintenaceArea) - Käidupiirkonna tuvastamine;
    - SVI14 (FindCustomerPointElectricalParameters) - Liitumispunkti elektrilised parameetrid;
    - SVI16 (GetMaintenanceAreaList) - Käidupiirkondade nimekirja pärimine;
    - SVI03 (ProcessNewConnectionCase) - Juhtumi koostamine uue liitumise kohta;
    - SVI07 (InsertProject) - Projekti algatamine;
    - SVI30 (UpdateProject) - Projekti andmete uuendamine;
- 5G Iseteenindus kasutab ELLI pakutavaid teenuseid ning sealhulgas ise teenuseid ei paku;
- Energik - Energik ei kasuta ELLI pakutavaid teenuseid. ELLI kasutab järgmisi Energik-u teenuseid:
    - SVI11 (GetPremisesV2) - Kliendi või teenuspunkti otsimine;
    - SVI01 (GetPersonalContactInformation) - Kliendi andmete pärimine;
    - SVI02 (GetServicePointDetailInfo) - Teenuspunkti andmete pärimine;
    - SVI26 (GetPremiseDetailInfo) - Tarbimiskoha andmete pärimine;
    - SVI27 (TranslateServicePointToPerson) - Võrgulepingu andmete pärimine;
    - SVI04 (InsertNewConnectionContact) - Kliendikontakti edastamine;
    - SVI06 (InsertBill) - Arve koostamine;
    - SVI05 (GetSettledBills) - Arve kontroll;

- ARCOM - ARCOM ei kasuta ELLI pakutavaid teenuseid. ELLI kasutab järgmisi ARCOM teenuseid:
    - SVI37 (GetRoleOwner) - Käidupiirkonna rolli täitja küsimine.

Lisa 4. ELLI pakutavad teenused

Elektrilevi liitumiste infosüsteem pakub järgmiseid teenuseid:

- aadressil: <APPLICATION_CONTEXT>/ws/ApplicationServiceSchema:
    - InsertApplicationRequest - Taotluse vastuvõtmine 5G-st;
    - GetApplicationInfoRequest - Liitumise info edastamine 5G-sse;
- aadressil: <APPLICATION_CONTEXT>/ws/Document:
    - GetDocumentRequest - Dokumentide väljastamine 5G-le;
- aadressil: <APPLICATION_CONTEXT>/ws/InverterListSchema:
    - InverterRequest - Inverterite nimekirja väljastamine 5G-le;
- aadressil: <APPLICATION_CONTEXT>/ws/ReceiveNewMeteringPoint:
    - ReceiveNewMeteringPointRequest - Uue liitumise juhtumi info vastuvõtmine;
- aadressil: <APPLICATION_CONTEXT>/ws/ConnectionApplicationQueryServiceSchema:
    - ConnectionProcessPricesRequest - Liitumishindade väljastamine 5G-le;
- aadressil: <APPLICATION_CONTEXT>/ws/SignContract:
    - SignContractRequest - Lepingu allkirjastamine 5G-st;
- aadressil: <APPLICATION_CONTEXT>/ws/SendConsumptionPlaceNotice:
    - SendConsumptionPlaceNoticeRequest - Tarbimiskoha nõuetelevastavuse treatise vastuvõtmine 5G-st;
- aadressil: <APPLICATION_CONTEXT>/ws/OrderSchema:
    - SendOrderUpdateRequest - Tellimuse info vastuvõtmine EPP-st;
- aadressil: <APPLICATION_CONTEXT>/monitooring:
    - Rakenduse monitoorimine. Monitooringu lehel on näha:
        - VERSION - Kuvatakse rakenduse versioon. Hetkel on rakenduse versiooniks “0.0.0.0”;
        - FILESYSTEM - Testitakse kirjutamist/lugemist/kustutamist kaustas, mis on määratud properties failis documents.path-le. Hetkel on selleks kaustaks /home/tomcat/elli;
        - DATABASES - Tehakse pöördumine baasi poole;

- INTEGRATIONS - Tehakse pöördumine EPP-u, NIS-i, Energik-u, ARCOM-i poole, et teada saada, kas ühendused teistesse süsteemidesse õnnestuvad;
- {“VERSION”:”0.0.0.0”,"FILESYSTEM":true,"DATABASES":{"ELLI":true},"INTEGRATIONS":{"EPP":true,"ENERGIK":true,"NIS":true,"ARCOM":true}}.

Lisa 5. Installijuhend

Projekti nimi: Elektrilevi Liitumised

EE projektijuht: Alar Jõeste

Tellija: Elektrilevi OÜ

Kirjeldus

Liitumisprotsesse juhitakse hetkel LD tootega Domino WorkFlow (edaspidi DWF). On tekkinud vajadus protsessijuhtimine tuua DWF keskkonnast välja ja luua uus infosüsteem protsessijuhtimiseks.

Uueks süsteemiks hakkab olema ELLI, kus hakatakse liitumisprotsesse juhtima Camunda protsessimootoriga. Liitumisprotsess kirjeldatakse ära BPMN-s, mis antakse Camunda-le sisendiks. BPMN faili suureks plussiks on selle lihtsus ja loetavus - sellest saavad aru ka inimesed, kes pole BPMN-ga varem kokku puutunud. Juhul, kui tulevikus tekib vajadus protsesse lisada siis see võimalus on Camunda-l täiesti olemas.

ELLI vastab täielikult protsessijuhtimisele esitatavatele peamistele nõuetele:

- Õigustes kasutaja peab ise saama protsesse kirjeldada
- Protsessis võib sisaldada tegevusi, mida täidetakse programmiliselt.
- Protsessipõhiselt käivitatud tööde menetlejad leitakse programmiliselt.
- Süsteem peab tagama mentelejate õigeaegse teavitamise.
- Menetlejad peavad saama jälgida enda menetletavaid töid.
- Õigustes kasutaja peab saama töö kulgu muuta.

Süsteemi lühikirjeldus

Rakendus (ELLI - ELektrilevi Liitumiste Infosüsteem) on realiseeritud Java-s kasutades Spring Boot raamistikku 1.4.0. versiooniga. ELLI suhtleb teiste süsteemidega veebiteenuste abil (SOAP). Andmebaasiks on Oracle Database 11g. Protsessimootoriks on Camunda 7.6.0. versiooniga. Rakendus ehitatakse valmis gradle-ga 2.6 ning tulemiks on elli-app.war. Ehituseks võib kasutada ka rakendusega kaasas olevat gradle wrapper-t.

Seotud süsteemidega: EPP, NIS, Iseteenindus 5G, Energik, ARCOM.

Andmebaasis kasutatakse ELCO, ELCA, ELPARAM, ELPM skeeme.

- ELCA — Camunda Process Engine ehk Protsessimootor
- ELPM — Elektrilevi Process Management ehk Protsessijuhtimine
- ELCO — Elektrilevi Connections ehk Liitumised
- ELPARAM — Süsteemi üldised parameetrid ja klassifikaatorid

Rakendus koosneb COMMON-PROC, CONNECTION-PROC, ELLI-APP, ELLI-DB, EXTERNAL moodulitest

- COMMON-PROC – Üldine
- CONNECTION-PROC —- liitumised
- ELLI-APP —- rakendus
- ELLI-DB —- andmebaas
- EXTERNAL —- teenuste klassid

Camunda-st kasutatakse BPMN, CMMN ja Cockpit-i funktsionaalsust

- BPMN — Kogu äriprotsess
- CMMN — Juhtumipõhine protsess. ELLI-s pakkumise koostamine kliendile.

- Cockpit — Veebirakendus, mille eesmärgiks on monitoorida ja administreerida käimasolevaid protsesse. Näeb ära kõik käimasolevad protsessid ning nende staatused. Kui mõni automaatne ülesanne peaks ebaõnnestuma siis seda saab cockpit-s uuesti käivitada ning on ka näha, et milles probleem oli.

Rakendus on pakendatud WAR-failiks, mis tuleb paigaldada rakendusserverisse.

Eeldused

- Java 1.8+
- Tomcat 8+

Installeerimine

1. Andmebaas

1.1. Luua andmebaasi kasutajad ELCO, ELCA, ELPARAM, ELPM. Kasutajal ELCO peab olema ELCA, ELPARAM ja ELPM skeemadesse kirjutamise/lugemise/kustutamise õigus. Selleks on loodud skript nimega: baseline.sql

1.2. käivitada gradle :elli-db:update

2. Konfigureerimine

2.1. /opt/conf/elli/logback.xml — ELLI logi häälestamise fail. Toodangu keskkonnas peaks olema piisav, kui logimise level klassidel on INFO tasemel. DEBUG tasemel logitakse välja kõikide exceptionite stact trace-id. Üldjuhul nii palju vaja ei ole.

2.2. /opt/conf/elli/admin.properties — ELLI konfiguratsioonifail.

3. Gradle build käsuga ehitatakse valmis elli-app.war, mis tuleb paigaldada veebikonteinerisse.

4. Rakenduse kasutamise alustamiseks võib minna aadressile "<APPLICATION_CONTEXT>/monitoring", kuna sinna minemiseks pole vaja õiguseid. Kui leht avaneb siis on näha, kas kõik on okei ning ega seal ei esine probleeme. Edasi piisab, kui minna aadressile "<APPLICATION_CONTEXT>", kus peab avanema töölaud ning vastavalt õigustele saab seal erinevaid toimunguid sooritada.

Näidis installeerimine (dev)

1. Checkout and build

1.1. Tehakse checkout svn-st

1.2. Kasutades gradle wrapper-t ehitatakse elli-app.war valmis.

```
chmod a+x gradlew

./gradlew -d -Dorg.gradle.java.home=/opt/java/jdk1.8.0_65 clean build
```

2. Database

2.1. Baas võetakse maha. Kõik tabelid kustutatakse. Jäetakse alles vaid schema-d.

```
chmod a+x gradlew

./gradlew :elli-db:dropAll
```

2.2. Baas pannakse uuesti peale.

```
chmod a+x gradlew

./gradlew :elli-db:update
```

3. Publish to Artifactory

3.1. Fabric-u skriptidega tõstetakse punktis 1.2 ehitatud elli-app.war Artifactory-sse.

4. Deploy artifact to DEV tomcat

4.1. Artifact tõstetakse veebikonteinerisse.

4.2. Tomcat-le tehakse restart.