

TALLINNA TEHNIKAÜLIKOOL  
Infotehnoloogia teaduskond

IT70LT

Mariana Lepassar 177000IAAM

**FUNKTSIONAALSUSTE LIIGUTAMINE  
MONOLIITSEST VEEBIRAKENDUSEST  
MIKROVEEBIDESSE TELIA EESTI AS-I  
NÄITEL**

Magistritöö

Juhendaja: Mart Roost

Magistrikraad

Lektor

Tallinn 2019

## **Autorideklaratsioon**

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

Autor: Mariana Lepassar

20.12.2019

## **Annotatsioon**

Käesoleva lõputöö eesmärk on analüüsida kliendikonto loomise protsessi, tuvastada sealjuures puudujäägid ja püstitada nõuded uuenenud funktsionaalsusele. Lisaks kirjeldada mikroveebide olemust ja tuua välja selle eelised võrreldes monoliitse esitluskihi rakendusega ning kliendikonto loomise liigutamine mikroveebi ja TO-BE arendusprotsessile.

Lõputöö raames on kirjeldatud AS-IS arendusprotsess Liferay platvormil, TO-BE arendusprotsess mikroveebis ning teostatud võrdlus. Lisaks on tehtud äri- ja süsteemianalüüs üle tõstetavale funktsionaalsusele (kliendikonto loomine). Funktsionaalsuste mikroveebidesse üle tõstmise järgset olukorda on töös hinnatud vastavalt arendajatele suunatud küsimustiku tulemustele.

Töö oluliseks järelduseks vastavalt küsitluse tulemuste analüüsile on see, et mikroveebide arhitektuuristiili juurutamine on oluliselt muutnud organisatsiooni tervikuna ja arendaja igapäevast tööd. Uued funktsionaalsused jõuavad kiiremini kliendini ja arendajad tunnevad, et suudavad efektiivsemalt ja kiiremini töötada.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 55 leheküljel, 6 peatükki, 16 joonist, 4 tabelit.

## **Abstract**

### **Moving Functionalities from a Monolithic Web Application into Micro Frontends in the Example of Telia Eesti AS**

The purpose of this thesis is to analyze the process of creating customer account, to find out its drawbacks and to set requirements for the renewed functionality. In addition, describe the nature of micro frontends architecture and point out their advantages over the monolithic presentation layer application. Further the aim is to move the customer account functionality to the micro frontend application and to TO-BE development process and describe how this platform change is done and which preconditions needed to be fulfilled.

The thesis describes the AS-IS development process on the Liferay frontend platform and TO-BE development process on the micro frontend application. Also, the comparison of those two processes is made and the advantages of the new one are highlighted. In addition, business and system analysis has been performed for upgradeable functionality (customer account creation). Detailed analysis diagrams have been modeled, which is a prerequisite for putting together development work and moving the functionality to the micro frontend. The situation after moving the functionalities to the micro user interface has been evaluated in the work according to the results of the questionnaire aimed at developers.

The author of this thesis conducted a survey aimed at understanding whether and how the implementation of the micro frontend architectural style has cognitively changed their daily work. The survey was answered by 15 developers from Telia Eesti AS, mainly from CRM division. An important conclusion of the work, according to the analysis of the survey results, is that the implementation of the micro frontend architectural style has significantly changed the organization as a whole and the developer's daily work. New functionalities reach customers faster and developers feel they can work more efficiently and independently.

The thesis is in Estonian and contains 55 pages of text, 6 chapters, 16 figures, 4 tables.

## Lühendite ja mõistete sõnastik

Andmebaasi trigger	Andmebaasi protseduuride käivitamise reeglid [1]
API	<i>Application Programming Interface</i> Arvuti operatsioonisüsteemige või rakendusprogrammiga määratud reeglistik, mille alusel rakendusprogramm kasutab operatsioonisüsteemi või teise rakendusprogrammi teenuseid [2]
AS-IS	Praegune protsess enne uuendamist
Back end	Tagarakendus, mille ülesandeks on kaudselt toetada eesteenusi ning nad on tavaliselt lähemal vajalikule ressursile ja/või suudavad sellega suhelda [2]
BPMN	<i>Business Process Modeling Notation</i> Standard äriprotsesside graafiliseks kirjeldamiseks
CRM	<i>Customer Relationship Management</i> Kliendihalduse valdkond ja selle süsteemid tervikuna
deploy	haarab kõiki uue tarkvara või riistvara töölepanekuga seotud protsesse - installeerimist, konfigureerimist, käitamist, testimist ja vajalike muudatuste tegemist [2]
Front end	Kasutajaliides, mida näeb iseteeninduse külastaja
Hateos link	<i>Hypermedia as the Engine of Application State</i> Arhitektuuristiil, mis võimaldab kasutada teenuse vastuse sisus olevaid linke, selleks, et klientrakendus saaks läbi linkide navigeerida soovitud ressursi juurde

Jira	Projektijuhtimise töövahend, kuhu kirjeldaks kõik arendustööd, et jälgida nende kulgu. Käesolevas töös käsitletud kui arendustööd
JSON	<i>JavaScript Object Notation</i> Failiformaat, mis koosneb atribuut-väärtus paaridest ja massiivi andmetüüpidest
jwt	<i>JSON Web Token</i> Avatud standard (RFC 7519), mis määratleb kompaktse ja iseseisva viisi teabe turvaliseks edastamiseks osapoolte vahel JSON objektina [3]
Liferay	Tarkvaraplattform veebisaitide ja veebirakenduste loomiseks
Portaal	Veebisait, mis on kasutusel Telia iseteeninduses ja võimaldab juurdepääsu teatud ressurssidele
Portlet	Veebipõhine komponent, mis töötleb pöördumisi ja genereerib dünaamilist sisu [4]
queue	Ootejärjekord, kuhu publitseeritakse andmemuudatused
Rabbit MQ	Tarkvara asünkroonseks sõnumiedastuseks
Reliis	Rakenduse uue versiooni tootmiskeskonda paigaldamine
REST	<i>Representational State Transfer</i> Veebiteenuste arendamise arhitektuuristiil
TO-BE	Tulevane protsess pärast funktsionaalsuste kolimist uuele rakendusele

## Sisukord

1 Sissejuhatus .....	11
1.1 Taust ja probleemistik .....	11
1.2 Töö eesmärgid .....	12
1.3 Metoodika.....	12
1.4 Töö ülesehitus.....	13
2 Arhitektuur ja arendusprotsess .....	14
2.1 Liferay arhitektuur .....	14
2.2 Mikroveebide arhitektuur .....	17
2.2.1 Mikroveebide arhitektuuri puudujäägid .....	21
2.3 Tänapäevane (AS-IS) arendusprotsess Liferay veebiplatvormil .....	23
2.4 Arendusprotsess (TO-BE) mikroveebide puhul .....	27
2.5 Arendusprotsesside erinevused.....	28
3 Kliendikonto loomine.....	30
3.1 Tänapäevane kliendikonto loomine Liferay veebiplatvormil .....	37
3.2 TO-BE kliendikonto loomine iseteeninduses .....	38
3.2.1 UC01 – Loo kliendikonto.....	40
3.2.2 Järgnevusdiagrammid.....	43
4 Kliendikonto loomise üleviimine mikroveebi .....	47

4.1 Meeskond ja töökorraldus .....	47
4.1.1 Tseremooniad .....	48
4.2 Eeldustööd funktsionaalsuse liigutamiseks mikroveebi .....	49
4.2.1 Ärilised eeldustööd .....	49
4.2.2 Tehnilised eeldustööd .....	51
4.3 Kliendikonto loomise funktsionaalsuse üleviimine jagatud teeki .....	53
4.4 Järeldused .....	53
5 Küsitlus ja tulemuste analüüs .....	55
5.1 Küsitluse tulemused.....	55
5.2 Lahendusettepanekud .....	63
5.3 Küsitluse tulemuste kokkuvõte.....	64
6 Kokkuvõtte .....	65
Kasutatud kirjandus .....	66
Lisa 1 - Kliendikonto loomise 1. samm.....	69
Lisa 2 - Kliendikonto loomise 2. samm.....	70
Lisa 3 – Uus ühesammuline kliendikonto loomine .....	71
Lisa 4 – Uus kliendikonto loomine koos andmetega.....	72
Lisa 5 – Küsitlus .....	73



## Jooniste loetelu

Joonis 1. Liferay komponentdiagramm [6]. .....	14
Joonis 2. Monoliitsed veebid [12]. .....	18
Joonis 3. Mikroveebid [12].....	19
Joonis 4. Mikroveebi arhitektuur.....	20
Joonis 5. Reliisitsükkel.....	23
Joonis 6. AS-IS arendusprotsess.....	25
Joonis 7. TO-BE arendusprotsess.....	27
Joonis 8. UML klassidiagramm - kliendikonto loomine. ....	31
Joonis 9. Kliendilepingu olekudiagramm.....	33
Joonis 10. Nõusoleku olekudiagrammi erinevus.....	34
Joonis 11. Komponentdiagramm - kliendikonto loomine. ....	35
Joonis 12. Kliendikonto loomine kasutaja vaates.....	40
Joonis 13. Järgnevusdiagramm - uus klient.....	44
Joonis 14. Järgnevusdiagramm - olemasolev klient. ....	45
Joonis 15. Domeenimudel. ....	51
Joonis 16. <i>Frontend</i> komponentide täiendatud loogika. ....	52

## **Tabelite loetelu**

Tabel 1. Reliisitsükli tähtajad.....	23
Tabel 2. Arendusprotsesside võrdlus.....	28
Tabel 3. Liferay arendusi puudutavad väited - vastused.....	56
Tabel 4. Mikroveebide arendusi puudutavad väited - vastused.....	59

# 1 Sissejuhatus

Käesoleva lõputöö teemaks on „Funktsionaalsuste liigutamine monoliitset veebirakendusest mikroveebidesse Telia Eesti AS-i näitel“. Autor käsitleb töös ühte iseteeninduse funktsionaalsust, mille vanalt veebiplatvormilt uuele tõstmisega tegeleb tema tiim. Töös on esitatud teemad, millesse autor on enim panustanud Telia CRM-i valdkonna *team lead*-i ja analüütikuna.

## 1.1 Taust ja probleemistik

Telia Eesti AS-i (edaspidi Telia) näol on tegemist telekommunikatsiooniettevõttega, mis on osa rahvusvahelisest Telia Company grupist. Käesoleval ajal (detsember 2019) töötab ettevõttes ligi 1800 töötajat ning era- ja äriklientidele pakutavate toodete arv ulatub 500-ni. Telia IT süsteemide arendamise ja ülalhoiu eest vastutab tehnoloogiaüksus, kus töötab kokku 500 inimest. Antud lõputöö autor kuulub vastava üksuse struktuuris kanalite- ja kliendihalduse osakonna koosseisu, mille peamiseks eesmärkideks on CRM-i valdkonna rakenduste, sh kliendi iseteenindus, teenindustegevusteks mõeldud süsteemid, arendus ja haldus.

Kui Elioni ja EMT-i ühinemisel võeti valdkondade ülese veebiplatvormina kasutusele Liferay rakendus, siis nüüdseks on ettevõtte arhitektuuriliseks suunitluseks valdkondade põhised mikroveebid, mis on üksteisest sõltumatud ja seega ka eraldiseisvalt arendatavad ja hallatavad. Selleks, et lõplikult Liferay platvormist (ja sellega kaasnevatest litsentsitasudest) vabaneda, tuleb igal valdkonnal luua oma vastutusala mikroveebid ning tõsta funktsionaalsused sinna üle. Funktsionaalsuste üle tõstmisega üksteisest sõltumatutele rakendustele saab tagada kiirema ja efektiivsema arendusprotsessi. Kuivõrd uued rakendused ei kuulu kesksesse reliaanshaldusesse, on versioonimuudatusi võimalik tootmiskeskonda paigaldada igal ajahetkel. Seeläbi saab tagada kliendile väärtust pakkuvad lahendused ning kasutajakogemust negatiivselt mõjutavate vigade paranduste kiirema jõudmise kliendini kui varasemalt.

## 1.2 Töö eesmärgid

Lõputöö eesmärgid on:

1. kirjeldada tänane arendusprotsess nii Liferay veebiplatvormil kui mikroveebides. Võrrelda neid kaht ja püstitada probleem;
2. analüüsida kliendikonto loomist, mis on ehitatud Liferay veebiplatvormile. Teostada tehniline analüüs, mis on eelduseks platvormi vahetamiseks;
3. analüüsida, millised on tänase kliendikonto loomise puudujäägid ning kaardistada selle funktsionaalsuse uued ärivajadused;
4. kirjeldada, kuidas toimus kliendikonto loomise funktsionaalsuse liigutamine mikroveebi ning millised eeldustööd selleks tuli teostada;
5. viia läbi küsitlus hindamaks, kas ja kuidas on mikroveebide arhitektuuri stiili juurutamine muutnud arendaja igapäevast tööd ning millised mured/rõõmud neil seoses sellega on tekkinud;
6. teha ettepanekuid, kuidas parandada küsitluse tulemustes ilmnenuid murekohti.

## 1.3 Metoodika

Autor selgitas koostöös äritellijaga, millised on uuenenud nõuded ületõstatavale funktsionaalsustele. Autor koostab praegustele funktsionaalsustele detailanalüüsi ja paneb kokku arendustööd.

Nõuded uuenenud funktsionaalsustele esitatakse kasutuslugudena (ingl k *User Story*). Funktsionaalsed nõuded kirjeldatakse kasutusjuhtudena (ingl k *Use Case*) ning parema ülevaate saamiseks modelleeritakse need UML (ingl k *Unified Modeling Language*) kasutusjuhtude diagrammil. AS-IS ja TO-BE arendusprotsessi modelleerimiseks kasutatakse BPMN notatsiooni. Teiste detailanalüüsi UML-keelsete diagrammide koostamiseks kasutab autor tarkvaralahenduste modelleerimise vahendit *Enterprise Architect*.

Mikroveebide arhitektuuri juurutamise osas viib autor arendajate hulgas läbi küsitluse, et aru saada, kuidas mikroveebide juurutamine mõjutab arendaja igapäevast tööd.

## 1.4 Töö ülesehitus

Lõputöö jaguneb neljaks sisupeatükiks. Esimeses peatükis kirjeldab autor praegust arendusprotsessi, kui funktsionaalsused on vanal valdkondade ülesel veebiplatvormil. Lisaks tutvustab autor mikroveebide olemust, arhitektuuri ja TO-BE arendusprotsessi.

Teises peatükis kirjeldab autor kliendikonto loomise funktsionaalsust ning toob välja praeguse protsessi peamised murekohad. Lisaks teeb autor detailanalüüsi, mis on eelduseks arendustööde kirjeldamiseks.

Peatükis „Kliendikonto loomise üleviimine mikroveebi“ selgitab autor, kuidas tiim liigutas funktsionaalsuse mikroveebi. Kirjeldatud on meeskonna koosseisu, töökorraldust ning funktsionaalsuse üleviimiseks vajalikke eeldustöid. Samuti on analüüsitud tööprotsessi ja välja toodud järeldused.

Neljandas peatükis viib autor läbi küsitluse arendajate seas. Peatükk sisaldab küsitluse tulemuste analüüsi ja ettepanekuid küsitluses välja toodud murekohtade leevendamiseks.

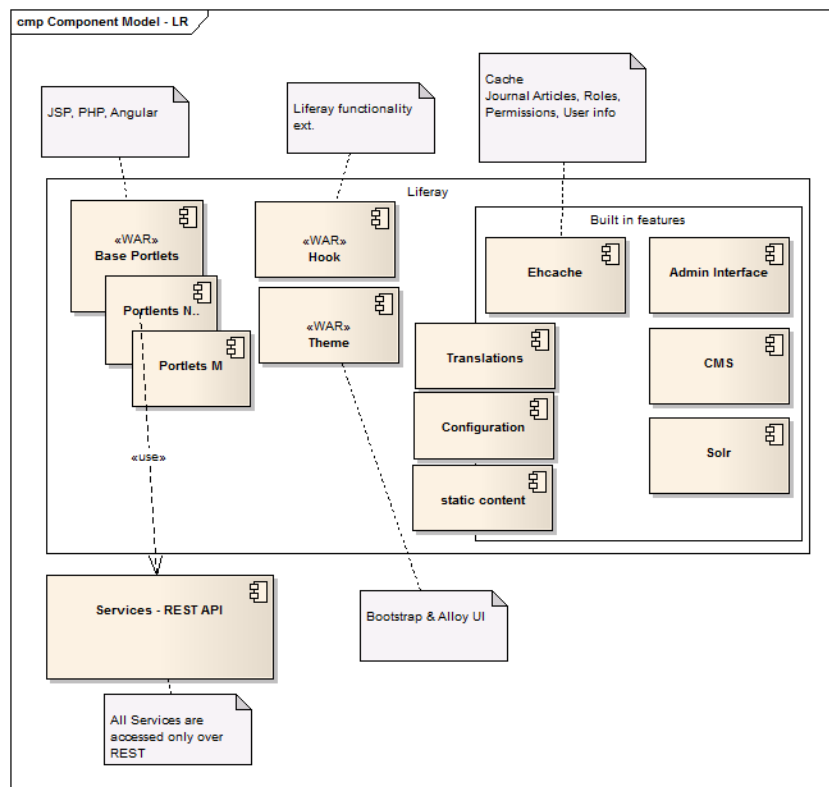
## 2 Arhitektuur ja arendusprotsess

Käesoleva peatüki eesmärk on kirjeldada nii tänane kui ka uus veebirakendus(t)e arhitektuur. Sealjuures tuua välja, milline on praegune arendusprotsess Liferay veebirakenduse näitel ning kaardistada selle kitsaskohad. Lisaks kirjeldada TO-BE arendusprotsess mikroveebides.

### 2.1 Liferay arhitektuur

Liferay on avatud lähtekoodiga ettevõtetele mõeldud veebiplatvorm. Portaali platvorm võimaldab kasutajatel veebisaitide loomist, koondades *portlet*-id ja vidinad portaalilehele. Portaali veebisait kombineerib teema (päis/jalus ja ühtne vaade), lehtede kogumiku, navigatsiooni (menüüriba jms) ning *portlet*-ide ja vidinate komplekti [5].

Järgnev komponentdiagramm kirjeldab, millistest osadest veebiplatvorm koosneb:



Joonis 1. Liferay komponentdiagramm [6].

Diagrammil olevad komponendid võib suures osas jagada kaheks:

- Liferay enda funktsionaalsus;
- Telia spetsiifiline lisatud funktsionaalsus.

Põhilised Liferay komponendid, mida Telia iseteeninduse juures kasutatakse [6]:

1. *EHcache* – Liferay kasutab *ehcache*-i staatilise sisu, kasutajate, õiguste jne enda sisemiste andmete cache jaoks;
2. Administreerimise liides (*Admin Interface*) - Liferay administreerimise liides, mis võimaldab serveri administraatoritel Liferay seadistamist ja sisuhalduritel sisu hallata;
3. CMS (ingl k *Content Management System*) - kõik sisuhalduseks vajalik;
4. Solr - Liferay poolt vaikimisi kasutatav otsingu implementatsioon.

Mõned Liferay-sse sisseehitatud funktsionaalsused on kohandatud vastavalt Telia spetsiifilistele nõuetele. Näiteks vaikimisi on kõik Liferay tekstid failisüsteemis. Selleks, et sisuhalduritel oleks võimalik tekste ja tõlkeid muuta, on lisatud tugi tekstide küsimiseks andmebaasist. Samalaadselt on lahendatud ka konfiguratsiooni osa ehk ka seda talletatakse andmebaasis, mitte failisüsteemis.

Peamised põhjused, miks Liferay portaal Telia kontekstis ei sobinud, on hästi kokku võetud Kai Wähneri poolt kirjutatud artiklis „*Pros and Cons – When to use a Portal and Portlets instead of just Java Web-Frameworks*“. Artiklis on välja toodud nii Liferay eelised kui ka puudused võrreldes tavaliste Java veebiraamistikega.

Puudused portaali kasutamise juures [7]:

- suurem keerukus – täiendav konfiguratsiooni vajadus, piirangud *portlet*-ide arendamisel;
- mahukam testimine;
- lisakulud – avatud lähtekoodiga ettevõtetele mõeldud versioon, mis sisaldab tugiteenuseid (Liferay);
- portaali peab kohandama ja rakendusi integreerima;
- arendajad vajavad lisaks veebiraamistiku kasutamisele täiendavaid oskusi.

Üheks portaali kasutamise eeliseks on välja toodud see, et Liferay pakub erinevaid lisasid nagu otsing, sisuhaldus, dokumentide haldus, web 2.0 tööriistad, koostöökomplektid (nt meeskonna lehed), analüüs ja aruandlus [7].

Artikli autori sõnutsi on mõistlik portaal kasutusele võtta üksnes siis, kui Liferay poolt pakutavaid lisasid saab päriselt ära kasutada. Samas toob ta välja, et palju lihtsam on luua ise lihtne veebirakendus, mida saab integreerida teistega [7].

Käesoleva lõputöö autor on uurinud mitmetelt Telia töötajatelt, kes on Liferay arendustega kokku puutunud, peamisi põhjuseid selle ebasobivuse osas. Üldiselt on välja toodud järgmiseid aspekte:

- Liferay pakub palju võimalusi, mida Telia kontekstis ei ole vaja kasutusele võtta;
- portaal on liiga suureks kasvanud ja seda on üha keerukam hallata;
- iga-aastased litsentsitasud;
- arendajad peavad omama Liferay spetsiifilisi teadmisi.

Mõningad peamised negatiivsed küljed monoliitse veebirakenduse puhul [8]:

- suuremahulist monoliiti on selle keerukuse tõttu keeruline hooldada ja arendada. Vigade otsimine nõuab oma koodibaasi läbinisti tundmist;
- monoliidid kannatavad nn „sõltuvuspõrgu“ [9] all, milles tarkvara teekide lisamine või värskendamine viib ebajärjekindla süsteemini, mis ei kompileeru/käita või käitub valesti;
- monoliidi ühe mooduli muudatused vajavad kogu rakenduse taaskäivitamist. Suuremahuliste projektide jaoks kaasnevad sellega seisakud, mis takistavad arendamist, testimist ja haldust;
- monoliit on tehnoloogia lukustus arendajate jaoks, kes peavad kasutama originaalrakenduse keelt ja raamistikke.

Monoliitne lähenemine suurele esitluskihi rakendusele muutub ajapikku kohmakaks. Vaja on viisi, kuidas see jagada väiksemateks mooduliteks, mis saaksid iseseisvalt tegutseda [10].



## 2.2 Mikroveebide arhitektuur

Tänane Telia iseteenindus on Liferay veebiplatvormil. Tegemist on monoliitrakendusega, mida arendavad valdkondade üleselt eri tiimid. Rakenduse reliisimine toimub kaks korda kuus, mistõttu on versioonivahetused mahukad, sõltuvused on suured ning seega risk vigade tekkeks on kõrge. Lisaks on vea tekkimisel selle paranduse reliisimine aeglane.

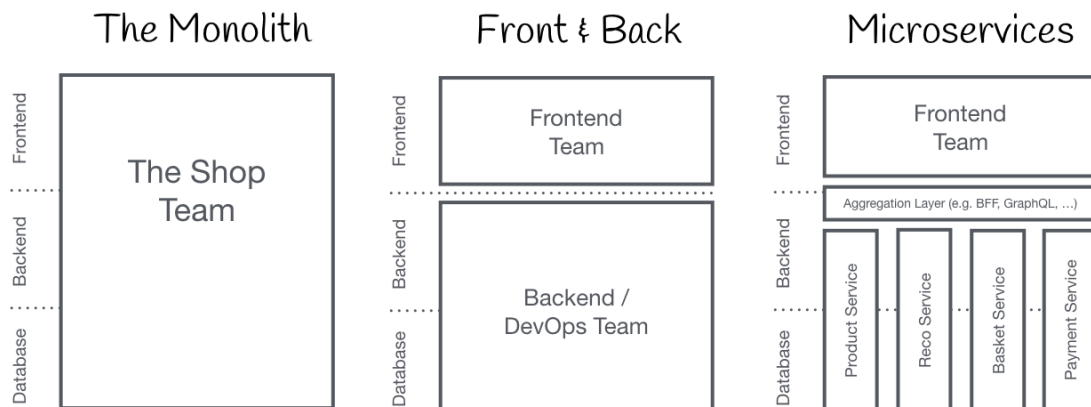
Viimastel aastatel on mikroteenuste populaarsus plahvatuslikult kasvanud ning paljud organisatsioonid on selle arhitektuurstiili kasutusele võtnud vältimaks suurtest monoliitrakendustest tulenevaid piiranguid. Kuigi palju kirjutatakse sellest stiilist serveripoolse tarkvara ehitamisel, siis paljud ettevõtted on hädas just monoliitsete esitluskihtide (ingl k *frontend*) koodibaasidega [11].

Viimasel ajal pööratakse üha enam tähelepanu üldisele arhitektuurile ja organisatsioonilistele struktuuridele, mis on vajalikud keerukaks ja kaasaegseks veebiarenduseks. Eelkõige näeme, et ilmnevad mustrid monoliitrakenduste jagamiseks väiksemateks lihtsamateks osadeks, mida saab iseseisvalt arendada, testida ja reliisida, jättes sealjuures kliendile mulje, et tegemist on ühe ühtse tootega. Me nimetame seda tehnikat mikroveebideks, mida me defineerime järgnevalt: „Arhitektuurstiil, kus iseseisvalt reliisitavad (ingl k *deliverable*) veebirakendused moodustavad suurema terviku“ [11].

Mikroveebide idee on mõelda veebirakendusele kui funktsioonide kogumile, mis kuuluvad sõltumatutele tiimidele. Igal tiimil on kindel ärivaldkond või missioon, millest ta hoolib ja millele spetsialiseerub. Tiim on ristfunktsionaalne (ingl k *cross functional*) ja arendab oma funktsioone otsast lõpuni – andmebaasist kasutajaliideseni [12].

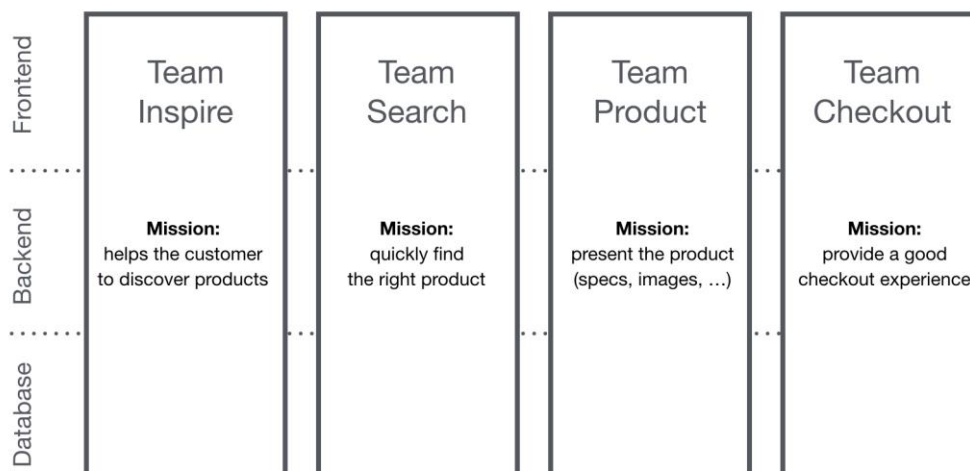
Mikroteenuste (saab paralleeli luua mikroveebidega) arhitektuuri lähenemisviis on ristfunktsionaalse meeskonna organiseerimine konkreetse teenuse ümber. Sellises meeskonnas on kõik arenduseks vajalikud oskused: kasutajakogemus, andmebaas ja projektijuhtimine [13]. Seda lähenemist tuntakse ka kui „sina ehitad, sina käivitad“ põhimõttena, mida esmakordselt tutvustas Amazoni CTO (Chief Technology Officer) Werner Vogels [14]. Selle lähenemisviisi kohaselt vastutavad meeskonnad tervikuna teenuse toe ja arendamise eest kogu selle elutsükli vältel [8].

Telia vaates ei ole veel jõutud päris olukorrani, mis on kujutatud joonisel 3. Seda seetõttu, et eksisteerib eraldi osakond, kus tehakse valmis disain ja esitluskihi komponendid, mida hiljem mikroveebi meeskonnad saavad kasutusele võtta. See on Liferay arendustest nii jäänud ja seetõttu tekib arenduste koha pealt pudelikael just veebikomponentide koha peale. Samas on hetkel käimas ettevõttes muutused veebitiimi tükeldamise osas, mis tähendab, et mikroveebide tiimid saavad enda meeskonna liikmeteks veebidisaineri ja veebikomponentide arendaja. Sellised sammud viivad lähemale õigele mikroveebide arhitektuurile ja vertikaalsetele tiimidele, mis on kujutatud joonisel 3. Liferay vaates asub CRM-i valdkond joonise 2 teise (*Front ja Back*) ja kolmanda (*Microservices*) joonise vahele.



Joonis 2. Monoliitsed veebid [12].

## End-to-End Teams with Micro Frontends



Joonis 3. Mikroveebid [12].

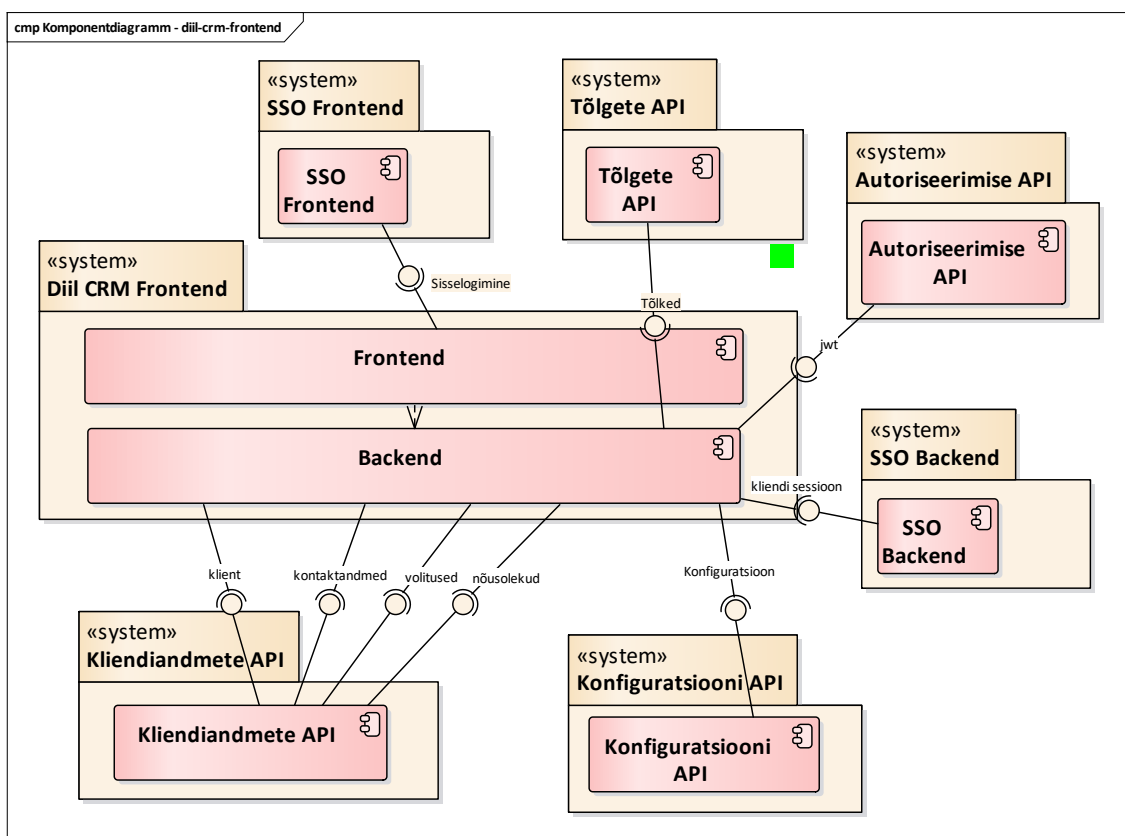
Mikroveebide arhitektuuri peamised eelised monoliidi ees on [15]:

1. **annab meeskondadele autonoomsuse** – enam ei pea tiimid, kes ei soovinud midagi reliisida, olema valves, kuna teiste tiimide arendused võivad eksikombel nende funktsionaalsusi lõhkuda. Iga tiim saab sõltumatuna arendada ja reliisida, valides sealjuures endale sobiva tehnoloogilise aluse;
2. **hoiab insenerid õnnelikena** – tervel koodibaasil mitte põhinemine vähendab sõltuvust ja ulatust, mis võimaldab meeskondadel kiiresti sisse elada ja kiiremini tarnida. See loob aega innovatsiooni jaoks, kartmata teiste meeskondade funktsioonide lõhkumist;
3. **skaleeruv ja paremini toimiv veebirakendus** – nõrgalt sidestatud arhitektuur teeb lihtsamaks uute funktsioonide lisamise ja tiimide loomise vastavalt vajadusele. Kui üks mikroveeb on aeglane, siis see ei mõjuta kogu rakenduse kui terviku toimimist.

Telia kontekstis ei viida funktsionaalsusi Liferay portaalilt mikroveebidele üks ühele. Vastupidi – monoliidi tükeldamine väiksemateks loogilisteks osadeks annab võimaluse teha asju paremaks, st uuendada koodibaasi, teha lihtsustusi ja kahelda varasemalt tehtud otsustes nii äri loogika kui koodi osas.

Sageli kujutatakse ette, et teenuse ekstraktsioon või monoliidi tükeldamine on eksisteeriva implementatsiooni kasutamine ja selle jagamine eraldi teenusteks. Osaliselt seetõttu, et meil on kognitiivne kallutatus koodi suunas, mida me disainime ja kirjutame. Kahjuks see kallutatus hoiab monoliidi tükeldamise saavutust tagasi. Alternatiivse variandina on tiimidel võimalik kood ümber kirjutada ja vana koodibaas kustutada. Ümber tegemine annab võimaluse vaadata uuesti peale ärireeglitele, algatada läbirääkimised äriga *legacy* protsessi lihtsustamiseks ja viia kahtluse alla vanad eeldused ja piirangud, mis on süsteemi loodud [16].

Järgnev komponentdiagramm kirjeldab Diili iseteeninduse näitel, millistest osadest koosneb mikroveeb (ülejäanud mikroveebid samalaadselt üles ehitatud):



Joonis 4. Mikroveebi arhitektuur.

Diili uue iseteeninduse rakendus (Diil CRM *frontend*) koosneb *frontend* ja *backend* kihist. *Frontend* osas on kõik see, mis on kliendile iseteeninduses nähtav – UI komponendid (*React*) ja funktsionaalsus (*typescript*). *Backend*-is on realiseeritud suhtlus teiste rakendustega (üle REST-i).

Autentimiseks ja sessiooni loomiseks on rakenduste üleselt kasutusel SSO rakendus (SSO *frontend* ja *SSO backend*). Autoriseerimiseks on kasutusel Autoriseerimise API, mis võimaldab SSO sessiooni alusel küsida jwt, mis tuleb edaspidi iga teise API poole pöördumisel kaasa anda.

Tõlgete API väljastab rakendusele tõlked. Tegemist on 2018. aastal loodud mikroveebide ülese tõlkerakendusega, kuhu on võimalik sisestada vene-, eesti- ja inglisekeelseid tõlkeid. Uue Diili iseteeninduse puhul on kogu sisu võimalik näha nii eesti- kui venekeelsena.

Kliendiandmete API väljastab Diili iseteenindusele kliendiandmebaasist vajalikku informatsiooni: kliendi volitused, kontaktandmed, isikuandmed, nõusolekud jne.

Konfiguratsiooni API võimaldab keskkonnapõhiselt valitud funktsionaalsuse peita. Kui hiljem on vaja see funktsionaalsus kliendile kasutatavaks teha, siis saab konfiguratsiooni muuta ilma rakenduse versioonivahetusega.

Anonüümse veebi (sisse logimata kasutaja diili avalehel) sisu saab hallata Wordpress-is. Viimast ei ole antud joonisel välja toodud, kuna diil-crm-frontend rakendus otse WORDPRESS-API-ga ei suhtle. Küll aga teeb seda diili avaleht. Ehk sisuliselt on Liferay CMS asendunud kahega: staatilise sisu haldamiseks Wordpress, iseteeninduse tekstide haldamiseks Tõlgete API.

### **2.2.1 Mikroveebide arhitektuuri puudujäägid**

Mikroveebide arhitektuuri rakendamine monoliitrakenduste tükeldamisel väiksemateks osadeks ei ole kindlasti nn hõbekuul. Sellel on omad puudujäägid, mida tuleb kindlasti enne vastava arhitektuuristiili juurutama hakkamist läbi mõelda.

Cam Jackson on enda 2019. aastal kirjutatud artiklis „Micro Frontends“ toonud välja paar puudujääki mikroveebide osas [11]:

#### **1. Tegevuse ja juhtimise keerukus**

Negatiivne külg, mille puhul saab paralleeli luua ka mikroteenuste arhitektuuriga. Rohkem hajutatud arhitektuurile omaselt viivad mikroveebid paratamatult rohkemate hallatavate asjade juurde – rohkem koodihoidlaid, tööriistu, pakiehituse/reliisimise kanaleid (ingl k *build/deploy pipelines*), servereid, domeene jne. Ehk peamiselt on

vaja enne mikroveebide juurutamist selgeks teha, kas sellise lähenemisviisi kasutuselevõtmiseks ilma kaost tekitamata on olemas vajalik tehniline ja organisatsiooniline küpsus.

## 2. Suurenenud koormus

Sõltumatult ehitatud *JavaScript*-i kimbud võivad põhjustada üldkasutatavate sõltuvuste dubleerimist, suurendades baitide arvu, mida tuleb lõppkasutajani üle võrgu viia. Näiteks kui iga mikroveeb sisaldab *React*-i koopiat, siis me sunnime klienti *React*-i alla laadima  $n$  korda, mis omakorda mõjutab lehe toimimist.

Samas - dubleerivate sõltuvuste probleemi eirates, on võimalik, et iga leht laadib ikkagi kiiremini, kui oleksime ehitanud monoliit esitluskihi. Klassikalises monoliittrakenduses laetakse tihti ühe lehe laadimisel kõikide lehtede lähtekood ja sõltuvused samaaegselt. Iseseisvate rakenduste puhul laetakse üksiku lehe laadimisel üksnes selle konkreetse lehe läte ja sõltuvused. Selle tulemusena võib lehe alglaadimine olla kiirem, kuid aeglasem hilisem navigeerimine, kui kasutajad peavad samad sõltuvused uuesti alla laadima.

Lisaks eelpool mainitud probleemidele on autori poolt läbi töötatud artiklites tulnud negatiivsena veel välja ka see, et iga mikroveebi tiim saab ise valida tehnoloogia, mida enda rakenduse loomisel kasutada. Kuid kas see on alati halb?

„Mikroteenused ei kehtesta erinevate tehnoloogiate kasutust. See, et need seda lähenemisviisi võimaldavad, ei tähenda, et me peaksime seda järgima. Sarnaselt mikroteenuste maailmale ei kasuta ka meie samas süsteemis 20 erinevat keelt. Kuna igauks neist on erinev ja toob süsteemi uue visiooni, on sellise ökosüsteemi haldamine väga kulukas ja potentsiaalselt segadust külvav, kui see just ei paku väga palju kasu [17]. Seega võib öelda, et väited, mida paljud toovad välja mikroveebide miinusena, võivad hoopis kujuneda kasulikuks. Praegusel juhul saab võimalikuks uute raamistike ja lähenemiste katsetamine saamaks aru, kas see võib mõjuda positiivselt rakenduste koos toimimisele.

Ülal mainitud mikroveebide võimalikud probleemid on võetud artiklitest ja autoril puudub selge ülevaade, kas need esinevad ka Telia kontekstis. Et selles küsimuses selgusele jõuda, on autor koostanud küsitluse Telia arendajatele (peamiselt CRM

valdkonna), et aru saada, millised mured ja/või rõõmud on neil seosed monoliit esitluskihilt mikroveebide lähenemisele üleminekuga. Küsitluse ja selle tulemuste analüüsiga on võimalik täpsemalt tutvuda peatükis „5 Küsitlus ja tulemuste analüüs“.

### 2.3 Tänapäevane (AS-IS) arendusprotsess Liferay veebiplatvormil

Liferay veebiplatvormil on tänapäevane Telia kliendi iseteenindus ning teenindusveeb, mida kasutavad Telia esinduste ja kõnekeskuse töötajad klientide teenindamiseks. Paljud funktsionaalsused nagu näiteks iseteeninduse avaleht, kliendi põhiaandmete leht, suurem osa Diili iseteeninduse funktsionaalsusest, teenindaja funktsionaalsused on juba üleviidud uutele mikroveebidele. Antud lõputöö raames liigutatakse veel üks oluline funktsionaalsus Liferay pealt mikroveebi.

Liferay rakendust reliaasitakse (viiakse muudatused tootmiskeskonda) 2 korda kuus – korralise reliaasi või täiendava reliaasi koosseisus. Järgnev joonis kirjeldab reliaasitsükli ehk reliaasi ettevalmistamist ja paigaldamist kõikidesse keskkondadesse:



Joonis 5. Reliaasitsükkel

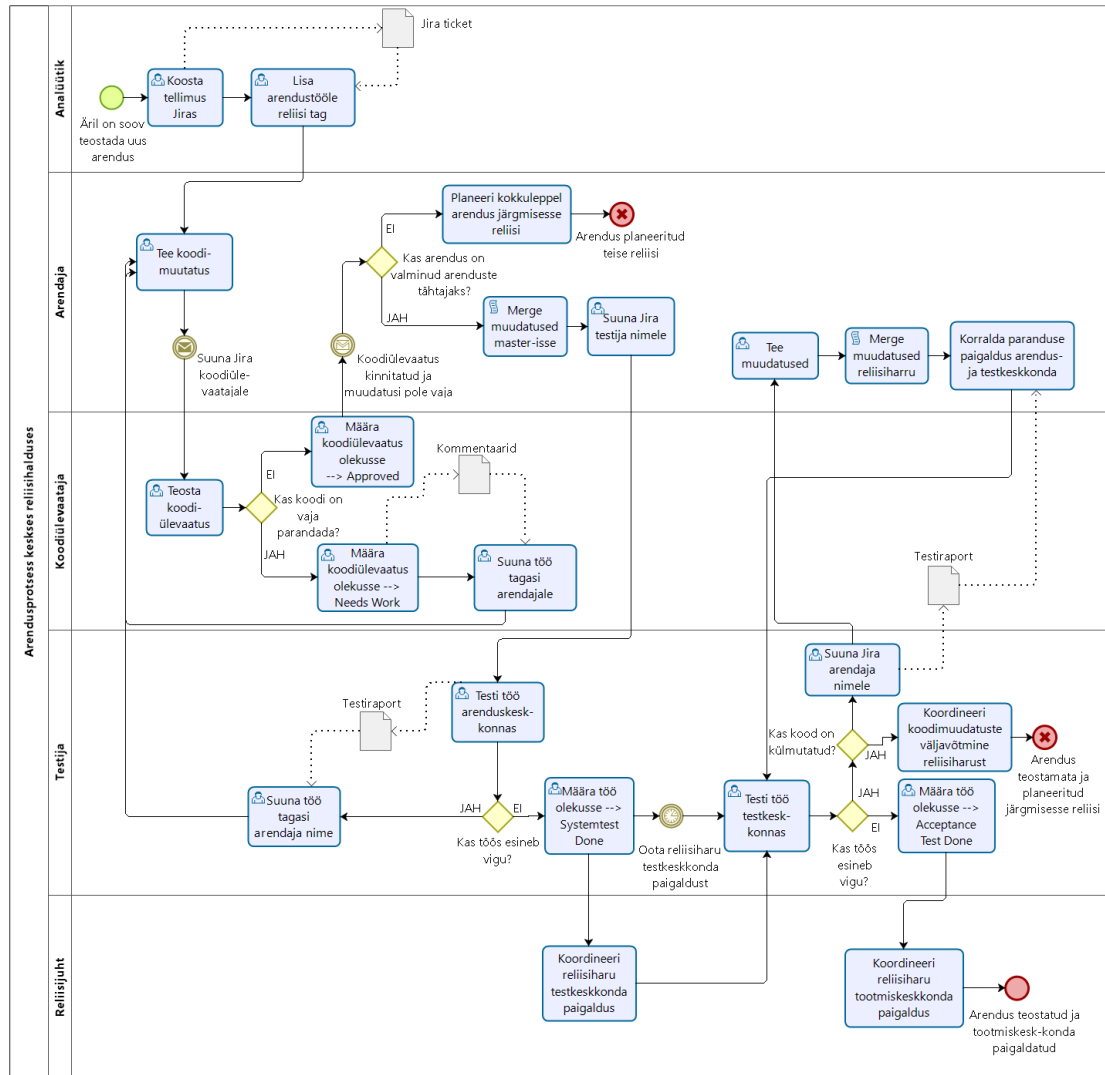
Arendustiimid on kohustatud kinni pidama reliaasitsükli läbimise eelduseks olevatest tähtaegadest. Selleks on igal arendustiimi liikmel ligipääs reliaasihalduse lehele, kus on võimalik vastavate kuupäevadega tutvuda. Järgnev tabel kujutab näidet korralise reliaasitsükli tähtaegadest:

Tabel 1. Reliaasitsükli tähtajad.

<b>R200211</b>	<b>Algus</b>	<b>Lõpp</b>	<b>Kommentaar</b>
Skoobi tähtaeg	17.01.2020	17.01.2020	Tööd reliisi planeeritud (reliisi <i>tag</i> Jiras lisatud)
Arenduste tähtaeg	24.01.2020	24.01.2020	Kõik arendused ja live vead olekus „Systemtest done“
Testkeskkonda paigaldus	27.01.2020	27.01.2020	Reliisipakkide testkeskkonda paigaldamine
Vastuvõtutestimine (ingl k <i>acceptance testing</i> )	27.01.2020	04.02.2020	Kõik arendused ja live vead olekus „Acceptance test done“
Kood külmutatud	05.02.2020	05.02.2020	
Paigalduskindluse test	10.02.2020	10.02.2020	Testkeskkonnas
Regressioonitestid	05.02.2020	07.02.2020	Testkeskkonnas, aruanded saadetud 15:00
Tootmiskeskonda paigaldus	11.02.2020	12.02.2020	

Järgnev BPMN protsessidiagramm kajastab arendusprotsessi tänase iseteeninduse näitel, mis on Liferay platvormil:





Joonis 6. AS-IS arendusprotsess.

Protsessi osapoolteks on:

- **Analüütik** – arendustiimi liige, kes kirjeldab soovitud arendustöö vastavalt äritellija soovidele;
- **Arendaja** – arendustiimi liige, kes teostab vastavas projektis vajalikud koodimuudatused;
- **Koodiülevaataja** (ingl k *code reviewer*) – arendaja, kes vaatab üle, et koodis ei esineks vigu ja et see vastaks ettevõttes kehtivatele koodistandarditele;

- **Testija** – arendustiimi liige, kes testib antud arendustöö vastavust analüüsi käigus seatud nõuetele;
- **Reliisijuht** – arendustiimi väline töötaja, kes omab tervikvaadet kogu reliisist ja vastutab skoobi, kvaliteedi ja õigeaegsuse eest. Jälgib, et arendustiimid peavad kinni kehtestatud reeglitest.

Arendusprotsessi sõnaline kirjeldus korralisse reliisitsükklisse kuuluva muudatusvajadusega rakenduse jaoks:

Protsess algab sellega, et äril on tekkinud uus arendusvajadus. Analüütik koostab vastava arendustöö (edaspidi Jira) ning lisab sellele külge korralise reliisi *tag*-i vastavalt sellele, kui prioriteetse arendusega on äri jaoks tegemist. Jira koostamise eelduseks on analüütiku eel- ja detailanalüüs, mida antud protsessis kajastatud ei ole.

Pärast seda, kui *Jira* on kirjeldatud, saab arendaja alustada koodimuudatustega. Kui vajalik funktsionaalsus on arendatud, siis suunab arendaja oma töö varasemalt kokkulepitud koodiülevaatajale. Kui ülevaataja on koodiga rahul ja puudujääke ei esine, siis märgib ta ülevaatuses „*Approved*“ olekusse, mille peale läheb arendajale teavitus, et tal on võimalus töö testimisse suunata. Kui ülevaataja leiab koodist puudujääke, siis ta lisab vastavate koodiridade juurde kommentaarid ning märgib ülevaatuses olekusse „*Needs work*“. Sellisel juhul läheb samuti arendajale teavitus, et kood vajab täiendamist. Kui täiendused on tehtud, laseb arendaja need koodiülevaatajal uuesti kontrollida ning antud protsess kordub kuni „*Approved*“ olek on saadud.

Pärast edukat koodiülevaatuset suunab arendaja töö testijale, kes testib antud muudatus arenduskeskkonnas. Kui testimisel puudujääke ei tuvastata, märgib testija vastava *Jira* olekusse „*Systemtest done*“. Puudujääkide avastamisel koostab testija vigadest raporti ning suunab *Jira* arendajale tagasi, kes teeb vajalikud täiendused koodis. Siinkohal võib tekkida olukord, kus töö ei jõua arenduste tähtajaks valmis ja testitud, mistõttu võib kerkida vajadus töö liigutamiseks järgmisesse korralisse reliisi.

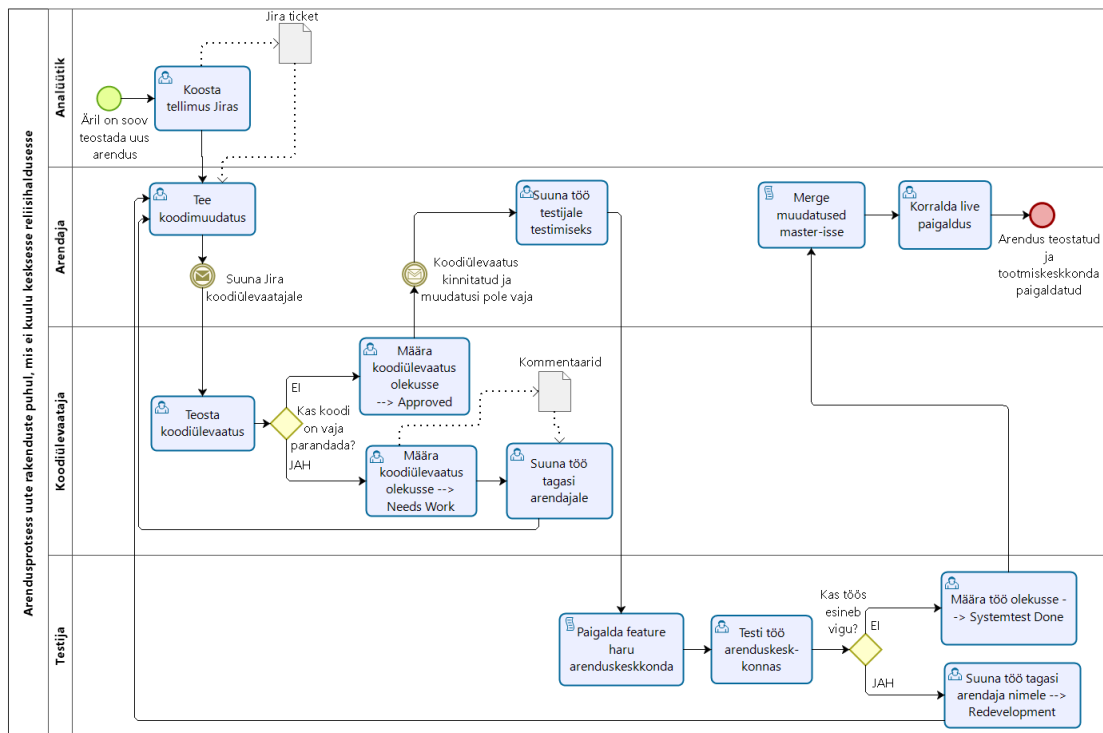
Juhul, kui töö jõuab olekusse „*Systemtest done*“ enne arenduste tähtaega, jätab testija töö enda nimele ja ootab kuni antud muudatus on tõstetud testkeskkonda, et viia läbi vastuvõtutestimine (ingl k *acceptance testing*). Kõikide korralise reliisi rakendustes olevate muudatuste testkeskkonda paigaldust koordineerib reliisijuht.

Kui testija vastuvõtutestimisel vigu ei tuvasta, paneb ta Jira olekusse „Acceptance test done“ ning muudatus jõuab toomiskeskonda reliisihalduse poolt koordineeritud tegevuste kaudu kokku lepitud kuupäeval.

Juhul, kui testija avastab vastuvõtutestimisel vea, mis arenduskeskkonnas testimisel ei ilmnenud, täiendab ta testiraportit ja vastavalt sellele, kas kood on külmutatud või mitte, korraldab muudatuste eemaldamise reliisiharust või suunab töö arendajale tagasi ja palub teha muudatused. Seejärel teeb arendaja täiendused ja korraldab muudatuste jõudmise arendus- ja testkeskkonda. Järgneb uuesti vastuvõtutestimise faas.

## 2.4 Arendusprotsess (TO-BE) mikroveebide puhul

Järgnev BPMN diagramm kirjeldab arendusprotsessi uutes mikroveebides, mida on võimalik reliisida igal ajahetkel:



Powered by bizagi

Joonis 7. TO-BE arendusprotsess.

### Arendusprotsessi sõnaline kirjeldus:

Protsessi esimene pool, st analüüs, tellimuse kirjeldamine, arendus ja koodiülevaatus toimib samaselt eelmisele protsessile (vt Joonis 2).

Kui koodiülevaatus on tehtud ja aktsepteeritud, suunab arendaja töö varasemalt kokku lepitud testijale. Testija paigaldab arendaja poolt loodud *feature*-haru arenduskeskkonda ning testib.

Vigade leidmisel koostab testija testiraporti ning suunab töö paranduste tegemiseks arendajale tagasi. Seejärel kordub kogu eelnev protsess kuni vigu ei esine.

Kui testija arenduskeskkonnas vigu ei leia, märgib ta töö olekusse „*Systemtest done*“. Antud protsessi erinevus võrreldes keskse reliisi rakendustega on see, et siin vastuvõtutestimist ei tehta, vaid kontrollitakse, kas uue versiooni paigaldus testkeskkonda on edukas või mitte. Seejärel saab arendaja enda *feature*-haru muudatused ühendada *master*-haruga. Viimaks tagab arendaja, et tema poolt tehtud arendus jõuaks tootmiskeskonda. Mõne rakenduse puhul läheb viimase tegevuse peale automaatselt käima versioonivahetuse protsess, mõnel juhul tuleb vastava rakenduse arhitektilt paluda reliisimist.

## 2.5 Arendusprotsesside erinevused

Järgnev tabel kirjeldab eelnevalt tutvustatud arendusprotsesside erinevusi, millest joonistuvad välja AS-IS arendusprotsessi puudujäägid ja vastupidi.

Tabel 2. Arendusprotsesside võrdlus.

	<b>AS-IS</b>	<b>TO-BE</b>
<b>Arenduste mastaap</b>	Korraga reliisitavate rakenduste arv suur, mistõttu versioonivahetuses palju muudatusi	Võimalikult vähe arendusi ühes versioonivahetuses
<b>Versioonivahetuste sagedus</b>	2 korda kuus	Nii sagedasti kui vaja

<b>Arendusprotsessi ajaline kestvus</b>	Umbes 1 kuu	Sõltuvalt arenduse mahust, võib olla ka 1 h
<b>Hõivatud osapoolte arv</b>	Suurem	Väiksem
<b>Testimine</b>	Kahes keskkonnas	Ühes keskkonnas
<b>Sõltuvused</b>	Palju sõltuvusi – paljud arendustiimid arendavad sama rakendust	Vähem sõltuvusi – iga valdkond arendab enda domeeni mikroveebi
<b>Üleüldine protsessi keerukus</b>	Suurem	Väiksem
<b>Agiilset arendusmetoodikat toetav</b>	Ei	Jah

Ülal väljatoodud tabelist selgub, et funktsionaalsuste liigutamine mikroveebidesse kiirendab arendusprotsessi, ühtlasi väheneb sõltuvus kesksele reliisitsüklile omastest tähtaegadest. Uus protsess võimaldab oluliselt kiiremini rakendusi reliisida, mistõttu funktsionaalsused jõuavad kliendile kasutusse kohe, kui need on valmis.

Lisaks toimub rakenduste reliisimine tihti ja seega on versioonivahetuses vähe töid. See omakorda aitab vea tekkimisel kiiremini aru saada, millises arenduses on viga tekkinud. Seevastu on Liferay platvormi reliisis üldiselt palju erinevate valdkondade arendusi, mis kõik reliisitakse korraga. Kui peaks juhtuma, et pärast reliisi tuvastatakse viga, mis takistab kliendil iseteenindust kasutada, tuleb kogu reliisieelne seis taastada.

### 3 Kliendikonto loomine

Kliendikonto loomine on üks funktsionaalsustest, mis on täna Liferay veebiplatvormil ja tuleb üle viia mikroveebi ehk teisisõnu viia eelmises peatükis kirjeldatud TO-BE protsessile. Kliendikonto loomine viiakse üle eksisteerivasse mikroveebi, kus hetkel on iseteeninduse avaleht. Tegemist on CRM-i valdkonna mikroveebiga, kuhu on tulevikus koondatud kõik sellised kliendiga seotud funktsionaalsused ja vaated, mis on teenindajale ja kliendile samasugused.

Telia Eesti AS-i kliendiks saamiseks tuleb eraisikul või juriidilise isiku esindajal luua kliendikonto. Konto loomine on eelduseks Telia teenuste tellimisele ja kauba ostmisele e-poest kliendi hinnaga ning personaalsete pakkumiste saamisele. Samuti on konto loomine oluline teenusepakkujale, kuna protsessis sõlmib isik ka kliendilepingu ning nõustub lepingust tulenevate tingimuste ja kohustustega.

Kliendikontot on võimalik luua kolmel erineval viisil:

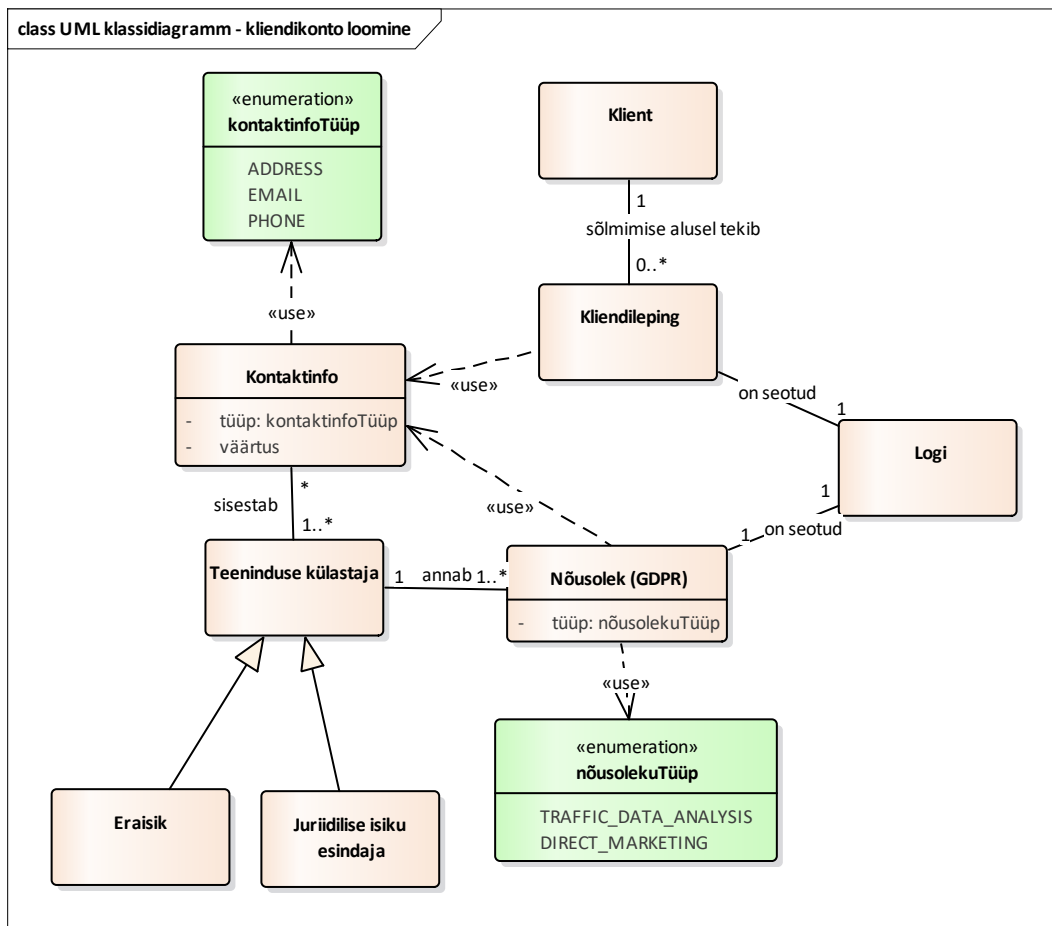
- Telia iseteeninduses;
- Telia esinduses ekraanil allkirjastades;
- Telia esinduses paberil allkirjastades.

Antud lõputöö keskendub üksnes kliendikonto loomisele iseteeninduses, kuna see on esimene osa, mida mikroveebidesse üle tõstetakse.

2016-2017. aastal sõlmiti 55% kliendilepingutest iseteeninduses ning 45% teenindusbüroos paberil. 2018. aasta mais loodi juurde uus kliendikonto loomise viis, mille puhul klient peab küll teenindusse kohale minema, kuid leping allkirjastatakse ekraanil. Viimasel aastal (okt 2018 – okt 2019) on sõlmitud 36% lepingutest teeninduses ekraanil, 20% paberil ning 44% iseteeninduses, mis teeb neist variantidest viimase populaarseimaks klientide liitumise allikaks [18].

Seega on väga oluline, et kliendikonto loomine iseteeninduses oleks võimalikult mugav ja et klient saaks teenuste tellimise eelduse täita võimalikult vähese ajakuluga.

Järgnev UML klassidiagramm kirjeldab konto loomisel osalevaid subjekte, nende tegevusi ning protsessi tulemusena tekkivaid objekte:



Joonis 8. UML klassidiagramm - kliendikonto loomine.

### UML klassidiagrammi tekstiline kirjeldus:

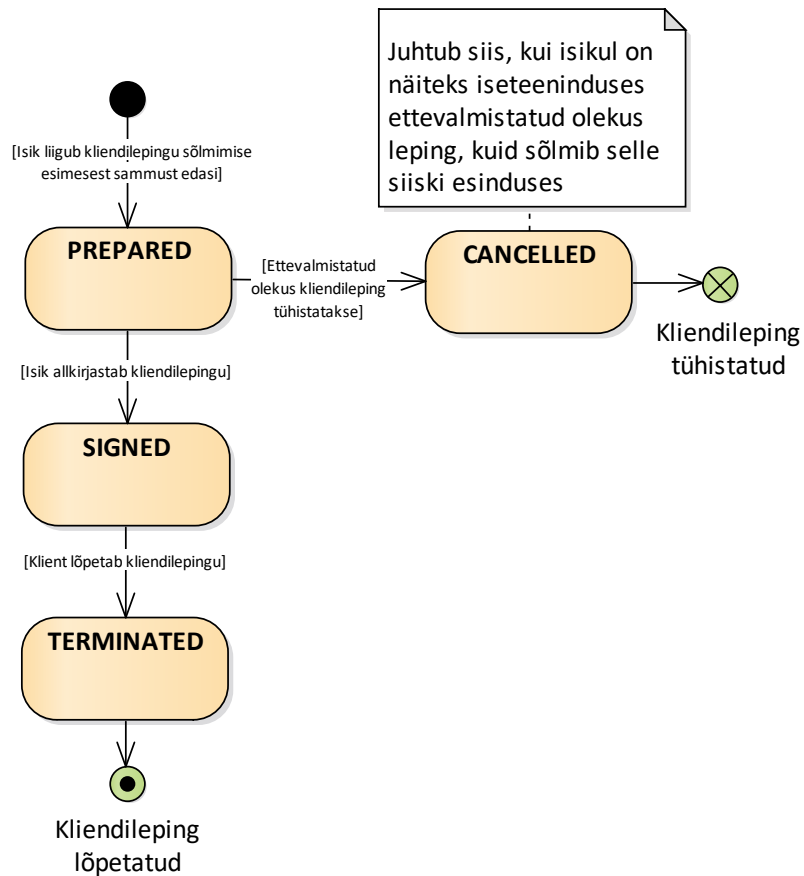
Kliendikonto looja võib olla kas eraisik või juriidilise isiku seaduslik esindaja. Neist viimase esindusõigus tuleneb Eestis registreeritud ettevõtete puhul äriregistrist, välismaa ettevõtteid on võimalik vajalike dokumentide alusel registreerida üksnes esinduses. Käesoleva lõputöö raames vaatleme stsenaariumi, kui iseteenindusse on sisse loginud eraisik ja ta esindab iseennast.

Kliendikonto loomisel tuleb isikul anda enda kontaktinfo. Kohustuslikus korras tuleb isikul sisestada enda aadress, e-mail või telefon ning suhtluskeel. Seejärel tuleb kasutajal valida, kas ta soovib edaspidi saada turunduspakkumisi ning kas lubab kasutus- ja arukohaandmete töötlemist teenusepakkuja poolsete turunduspakkumiste genereerimiseks (GDPR-ist tulenevad nõuded). Kliendikonto loomise teises sammus on isikul võimalik tutvuda ettevalmistatud kliendilepinguga ning sõlmida varem antud nõusolekud ja kliendileping. Sõlmimise peale tekitatakse logi, millele lisatakse digitempel ja seda saab hiljem kasutada võimalike juriidiliste vaidluste puhul teenusepakkuja ja kliendi vahel.

Kliendikonto loomise puhul kehtib juriidiline nõue, et klient peab enne lepingu allkirjastamist saama seda dokumenti näha. Sellest tulenevalt on vaja genereerida 3 erinevat ettevalmistatud *pdf* formaadis dokumenti: neist kaks on nõusolekud ja üks kliendileping. Nüüd avaneb kliendil võimalus need alla laadida ja dokumentide sisuga tutvuda.

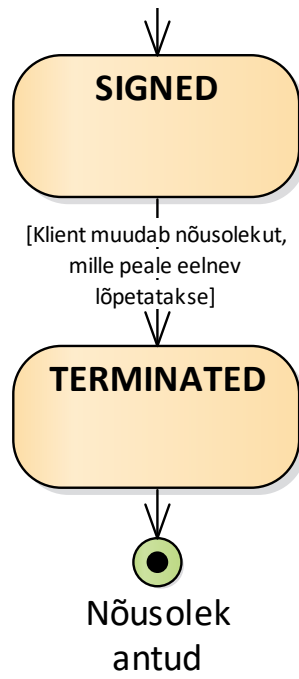
Järgnevad diagrammid kirjeldavad kliendilepingu ja nõusolekute võimalikke olekuid ning sündmusi, mis olekumuutusi põhjustavad:





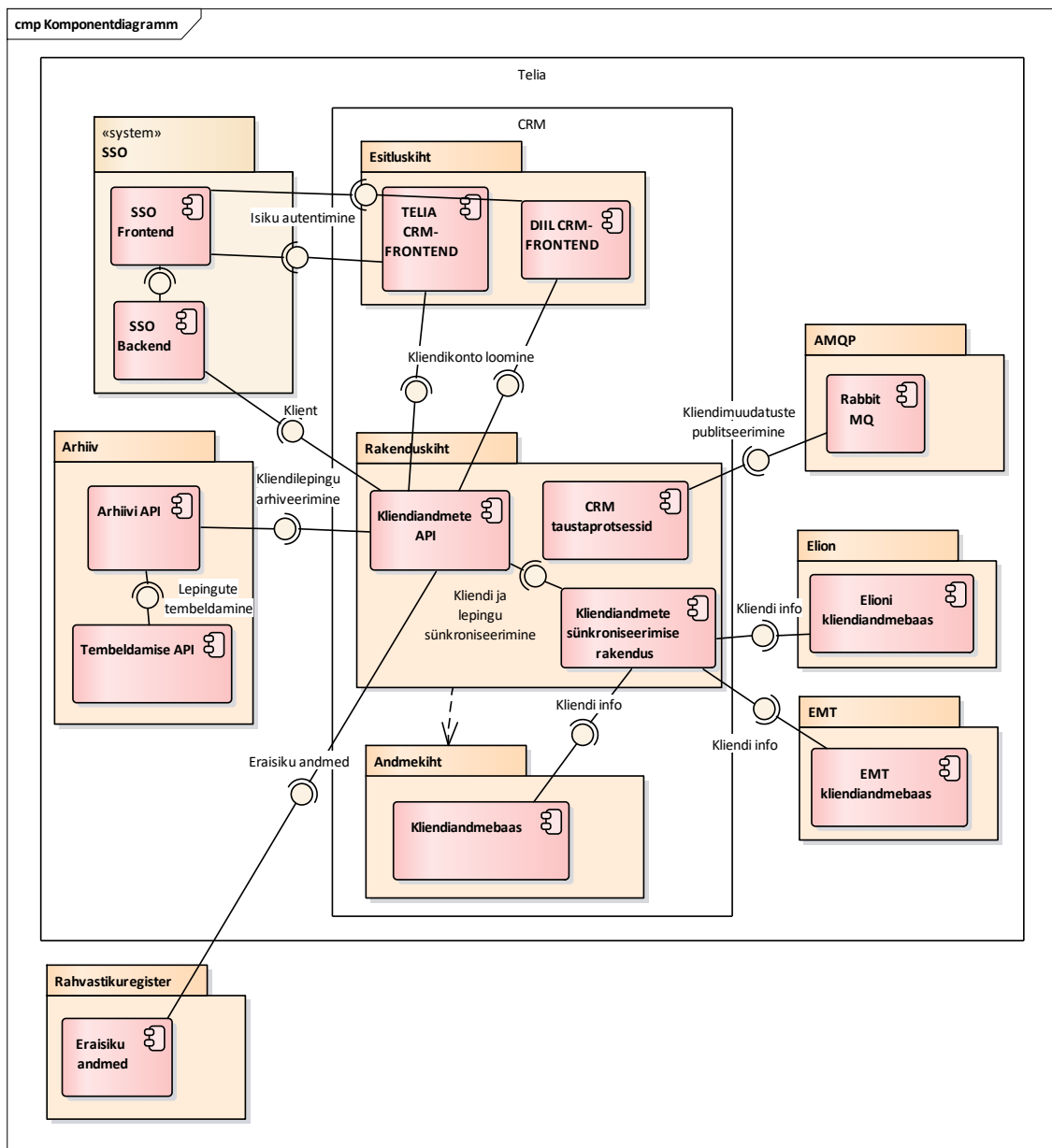
Joonis 9. Kliendilepingu olekudiagramm.

Nõusoleku olekudiagramm on kliendilepingu diagrammiga väga sarnane. Ainus erinevus on see, et nõusolekud liiguvad lõpetatud olekusse iga kord, kui neid muudetakse vastavalt (Ei → Jah või Jah → Ei).



Joonis 10. Nõusoleku olekudiagrammi erinevus.

Järgnev komponentdiagramm kirjeldab, millised Telia sisesed ja välised rakendused kliendikonto loomise protsessis osalevad:



Joonis 11. Komponentdiagramm - kliendikonto loomine.

Kliendikonto loomise protsessiga on seotud palju rakendusi. Antud lõputöö keskendub üksnes erakliendi konto loomise protsessile, ärikliendi vaates on komponente veelgi rohkem.

CRM valdkonna komponendid:

- Esitluskiht
  - Telia crm-frontend – Telia iseteeninduse osa (mikroveeb), mis koosneb sellistest vaadetest, mis on kliendile ja teenindajale täpselt samad või mõningate erinevustega (iseteeninduse avaleht, kliendi andmed, lepingud, kliendikonto loomine);
  - Diil crm-frontend – Diili iseteenindus (mikroveeb).
- Rakenduskiht
  - Kliendiandmete API – *REST API*, mis võimaldab väljastada, sisestada uuendada ja kustutada kliendiandmebaasi andmeid;
  - Kliendiandmete sünkroniseerimise rakendus – rakendus, mis tagab, et kolmes klientide andmeid säilitavas andmebaasis on andmed samad. Korjab ühes või teises andmebaasis tehtud muudatused üles ja sünkroniseerib neid andmebaaside vahel. Elioni ja EMT-i puhul on tegemist vanade monoliittrakendustega ning samal ajal sulguvate pärandüsteemidega;
  - CRM taustaprotsessid – taustaprotsesside rakendus, kust muuhulgas publitseeritakse vastavalt andmebaasis defineeritud trigeritele kliendiandme muudatused Rabbit-i *queue*-sse, kust tarbijatel on võimalik muudatusi kuulata.
- Andmekiht
  - Kliendiandmebaas – andmebaas, mis koondab endas järgmiseid kliendiandmeid: isikuandmed, kliendileping, kontaktisikud, volitused, pöördumised, kliendihaldurid, kontaktandmed jne.

#### Arhiiv:

- Arhiivi API – *REST API*, mis võimaldab lepingud digitaalsel kujul arhiveerida;
- Tembeldamise API – rakendus, kus toimub kliendilepingute tembeldamine, st Telia Eesti AS-i poolse digitempli lisamine.

Digitaalne tempel e. digitempel on teenus, mille abil saavad juriidilised isikud (nt ettevõtted) anda digiallkirju dokumentidele. Seeläbi lisatakse dokumendile kinnitus, et dokument pärineb allkirjastanud ettevõttest (s.t digidokumendi kinnitab asutus, mitte selleks volitatud füüsiline isik -- inimesed vahetuvad, asutus jääb) ning et dokumenti ei ole vahepeal muudetud. Allkirjastada (sh mass-

allkirjastada) saab nt arveid, maksekorraldusi, kinnitusi, tunnistusi, väljavõtteid (nt SEB pank pakub automaatselt digitembeldatud kontoväljavõtet) jne. Teenuse tellimisel väljastab SK ettevõttele USB-krüptopulgal X.509-sertifikaadi (mille kasutusala määratakse sertifikaadi nimetusega) ning analoogiliselt digiallkirjaga luuakse templi kasutamisel allkirjastatavatest andmetest DigiDOC-vormingus konteiner [19].

SSO:

- (SSO *Frontend* + SSO *Backend*) – rakendused, mis võimaldavad isikutel logida iseteenindusse. Samuti vajalik autoriseerimiseks erinevate API-de poole pöördumisel.

Rahvastikuregister:

- Eraisiku andmed – saame kliendikonto loomisel isikukoodi alusel kontrollida sisse loginud isiku olemasolu ja andmeid rahvastikuregistrist.

### **3.1 Tänane kliendikonto loomine Liferay veebiplatvormil**

Tänases kliendikonto loomise protsessis (Liferay portaalis) tuleb potentsiaalsel kliendil konto loomiseks teha järgmised tegevused:

SAMM 1 (vaata täpsemalt Lisa 1 – Kliendikonto loomise 1. samm):

- Kontaktandmete sisestamine (kontaktaadress kohustuslik, lisaks kohustuslik – e-posti aadress või telefoninumber);
- Suhtluskeeles valik – vajalik selleks, et kui teenindusel on mingil teatud põhjusel vaja kliendiga hiljem ühendust võtta (pöördumine, võlateavitus vms), siis suheldakse soovitud keeles (eesti, vene ja inglise keel);
- Nõusolekute andmine – vajalik GDPR regulatsiooni jõustumise järgselt selleks, et ettevõttes oleks teadmine, kas antud kliendile tohib saata personaalseid pakkumisi ning kas klient lubab enda teenuste kasutust töödelda selleks, et teenusepakkuja saaks edaspidi teha personaliseeritud pakkumisi.

SAMM 2 (vaata täpsemalt Lisa 2 – Kliendikonto loomise 2. samm):

- kliendil on võimalik tutvuda ettevalmistatud kliendilepingu ning nõusolekutega;
- klient peab kinnitama, et on tutvunud kliendilepingu ning Telia Eesti AS üldtingimustega. Selles sammus on võimalik kasutajal kliendilepingu sõlmimisest veel loobuda.

Peamised murekohad tänases kliendikonto loomise protsessis:

1. protsess on kahesammuline ja isik peab allkirjastama kolm lepingut. Kuna kliendikonto loomise protsess peaks olema nii lihtne kui võimalik, siis tuleb kasutajate tegevuste arvu vähendada;
2. kontaktandmete plokis on kohustuslik üks kahest – e-posti aadress või telefon. Kuna peamiseks kliendiga suhtlemise kanaliks on e-post, siis on vajalik, et see oleks täidetud kõikidel klientidel;
3. tegemist on Liferay sisulehega, mida ei saa teistes protsessides kasutusele võtta. Tulevikus on oluline, et kui isik satub näiteks iseteeninduses teenuse tellimise protsessi, saab ta kohe selle sisemiselt luua kliendikonto. Teisisõnu pole vaja suunata klienti mujale ja pärast teenuse juurde tagasi;
4. arendusprotsess on keerukas ja muudatuste tegemine aeganõudev.

### **3.2 TO-BE kliendikonto loomine iseteeninduses**

Oluliseks eelduseks kasutaja vajadustele vastava tarkvara projekteerimisele on arusaam sellest, mida kasutaja kavatseb sellega teha. Keskendudes kasutajatele ja nende oodatavatele tegevustele on võimalik välja selgitada oluline funktsionaalsus ning vältida ebavajalike osade realiseerimist [20].

Kasutajakesksed nõuded on võimalik kirja panna kasutuslugudena ja kasutusjuhtudena. Kasutusjuht kirjeldab suhtluse jada süsteemi ja välise tegutseja vahel, millest tegutseja saab talle mingit väärtust pakkuva resultaadi. Kasutuslugu seevastu on lihtne kirjeldus süsteemi kasutaja vaates, kes soovib uut pakutavat funktsionaalsust. Kasutuslood kirjutatakse üldjuhul järgnevas formaadis [20]:

<Kellena> soovin <eesmärk> selleks, et <põhjus> [20].

Antud lõputöös kirjeldab autor funktsionaalsed nõuded uuele kliendikonto loomise protsessile kasutuslugudena (ingl k *User Story*).

Funktsionaalsed ja mittefunktsionaalsed nõuded uuele kliendikonto loomise komponendile on järgmised:

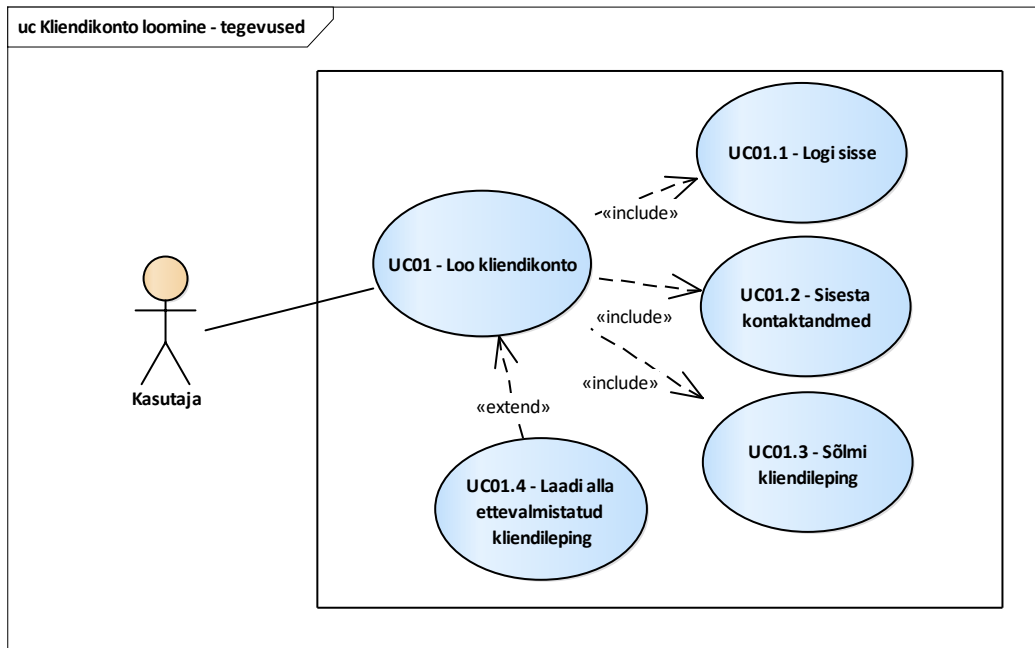
### **Funktsionaalsed nõuded**

1. kasutajana soovin, et mulle kuvataks minu kontaktandmed võimaliku varasema seose põhjal Teliaga, et ei peaks andmevälju nende valiidsuse korral uuesti täitma;
2. kasutajana soovin, et saaksin tutvuda kliendilepingu sisuga enne selle allkirjastamist, et olla kindel selle sisus ja andmete korrektsuses;
3. kasutajana soovin, et saaksin allkirjastada kliendilepingu selleks, et jätkata teenuste tellimise tegevusega.

### **Mittefunktsionaalsed nõuded**

1. kliendikonto loomine peab olema võimalikult kiire ja lihtne, et potentsiaalne klient saaks edasi liikuda teenuste tellimisega;
2. kasutaja jaoks peab olema koheselt protsessi jõudes hästi arusaadav, millised väljad tuleb täita kohustuslikus korras;
3. kliendikonto loomise protsess peab olema läbitav ühes sammus, et saaks kiiremini personaalsete pakkumiste ja teenuste tellimise juurde asuda;
4. kliendikonto loomise funktsionaalsus peab olema lahendatud jagatava komponendina. Seda peab olema võimalik kasutusele võtta ka muudes protsessides, mille eelduseks on aktiivse kliendilepingu olemasolu.

Kliendikonto loomise protsessi kasutajapoolsed tegevused (funktsionaalsused) on valitud lähtuvalt sellest, millist tagasisidet oleme vana protsessi kohta kasutajatelt kogunud. Eesmärk on teha kliendikonto loomine ühesammuliseks ja jätta sellest protsessist välja nõusolekute andmine, kuna need ei ole konkreetselt teenuste tellimise eelduseks.



Joonis 12. Kliendikonto loomine kasutaja vaates.

Kliendikonto loomiseks tuleb kasutajal esmalt iseteenindusse sisse logida (võimalikud sisselogimise viisid: ID-kaart, Mobiil-ID, Smart-ID, pangad). Telia rakenduste ülene autentimine ja autoriseerimine oli lahendatud enne käesoleva lõputöö kirjutamist ja seda kasutatakse ka uue kliendikonto loomise puhul samalaadselt Liferay-le. Seega selle osa analüüsi lõputöö ei sisalda, kuid selguse huvides on kasutusmallide diagrammil välja toodud.

### 3.2.1 UC01 – Loo kliendikonto

Uue kliendikonto loomise protsessi visuaalse vaatega on võimalik tutvuda lõputöö lisades (vaata Lisa 3 - Uus ühesammuline kliendikonto loomine ja Lisa 4 - Uus kliendikonto loomine koos andmetega)

**ID:** UC01

**Kirjeldus (ingl k *Description*):**

Kasutaja loob kliendikonto ja saab seeläbi Telia Eesti kliendiks

**Roll (ingl k *Actors*):**



- a. Kasutaja/Isik – loob kliendikonto
- b. Süsteem – lisab taustal kliendiandmebaasi vajalikud andmed, loob lepingu arhiivi

**Eeltingimused (ingl k *Preconditions*):**

- a. Isik on sisse loginud Telia iseteenindusse
- b. Autenditud kasutajal puudub allkirjastatud kliendileping ja tegemist on täisealise isikuga
- c. Isik on valinud esindatavaks iseenda (mitte mõne teise era- või ärikliendi)

**Standard protsess (ingl k *Standard path*):**

1. Iseteenindusse sisse loginud isik, kellel puudub allkirjastatud kliendileping on sattunud kliendikonto loomise protsessi
2. Kasutaja täidab kohustuslikud kontaktandmete väljad (aadress, telefon, e-post, suhtluskeel)
3. Kasutaja kinnitab sisestatud kontaktandmete õigsust tehes vastavalt linnukese „Kinnitan andmete õigsust“
4. Süsteem loob kliendiandmebaasi isiku, kontaktandmed, volituse, ettevalmistatud olekus kliendilepingu ning saadab viimase arhiivi
5. Kasutaja tutvub ettevalmistatud olekud kliendilepinguga (laeb alla *pdf* formaadis dokumendi)
6. Punkt 5 kliki peale pärib süsteem arhiivist ettevalmistatud leping *pdf* kujul
7. Kasutaja kinnitab, et on nõustunud Telia üldtingimustega ning kliendilepingu sisuga
8. Kasutaja klikib „Kinnitan ja jätkan tellimist“
9. Süsteem muudab taustal kliendilepingu oleku allkirjastatuks

**Alternatiivsed protsessid (ingl k *Alternate paths*):**

a) Kasutaja soovib pärast kontaktandmete sisestamist ja nende kinnitamist neid muuta.

1. (Standard protsess 1-4)

2. Kasutaja muudab kontaktandmeid

3. Kasutaja kinnitab sisestatud andmete õigsust

4. Süsteem uuendab taustal varem loodud ettevalmistatud olekus lepingut ning uuendab lepingu metaandmed ka arhiivis. Lisaks uuendab vastava isiku kontaktandmed kliendiandmebaasis

5. (Standard protsess 5-9) jätkub

b) Kasutaja otsustab ümber ja ei soovi kliendikontot luua

Võib mis tahes hetkel välja logida või mõnele muule lehele suunduda ning kliendikontot ei looda.

**Lõpptulemused (ingl k *Post conditions*):**

a) Edukas protsess (ingl k *Successful path*)

Kliendikonto on loodud:

Kontaktandmed on sisestatud;

Kontaktandmete õigsus on kinnitatud;

Kliendi andmed on süsteemi sisestatud;

Kliendileping on sõlmitud.

b) Katkestatud protsess (ingl k *Unsuccessful path*)

Kliendikontot ei looda, st kliendileping jääb ettevalmistatud olekusse või seda ei looda süsteemi poolt üldse.

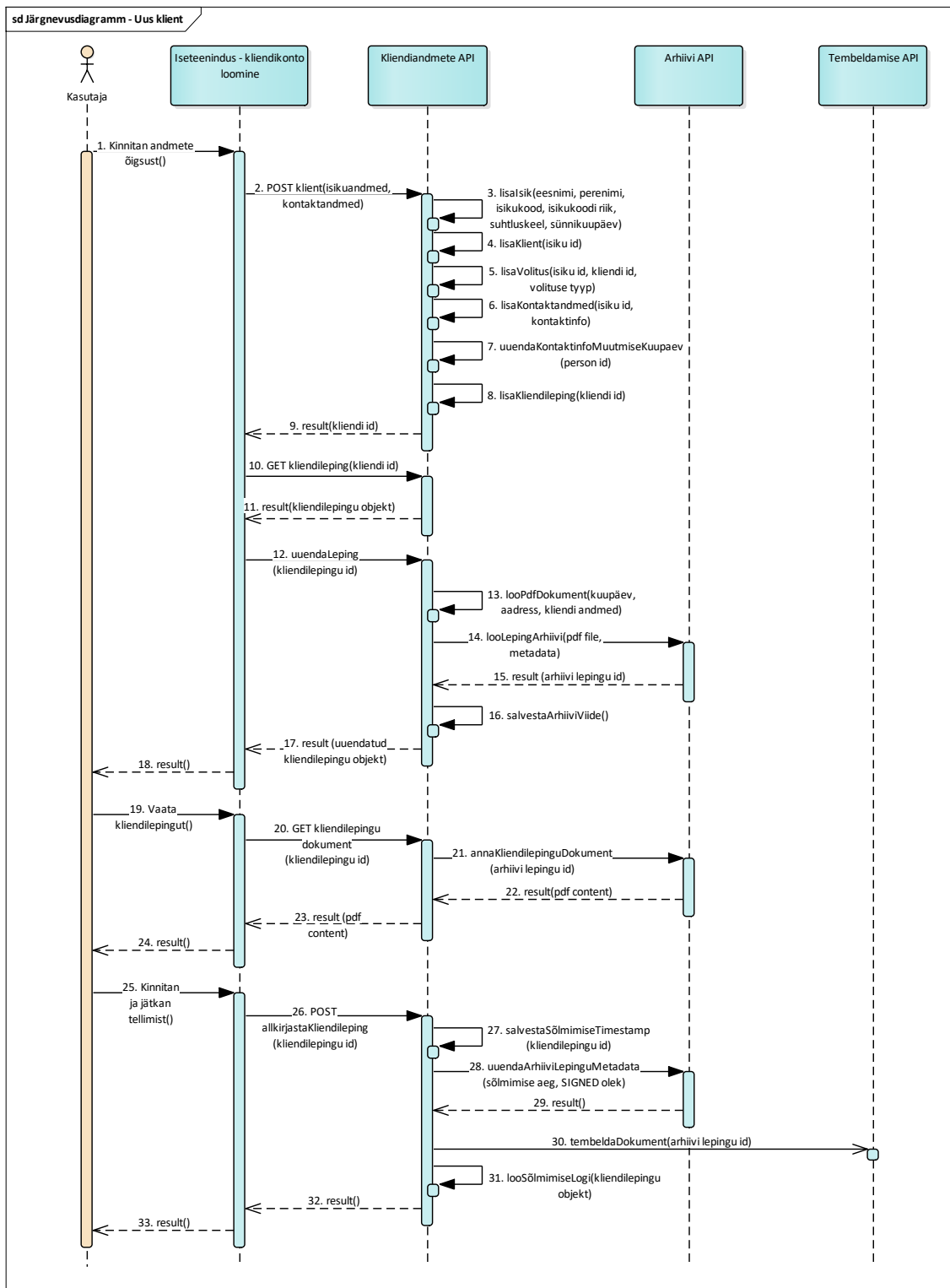
### **3.2.2 Järgnevusdiagrammid**

Telia CRM süsteemide vaates saab erakliendi kliendikonto loomise protsessi sattuda kahte tüüpi isikuid:

- a. isikud, kes ei ole kunagi varem Teliaga seotud olnud, st kliendiandmebaasis puudub nende kohta info;
- b. isikud, kes eksisteerivad kliendiandmebaasis kliendina, kuid puudub allkirjastatud kliendileping.

Sellest tulenevalt on defineeritud ka süsteemi käitumine, st teenuste väljakutsed ja seega ka andmete salvestamine. Selleks, et joonistuks paremini välja, milliseid CRM-i teenuseid tuleb mis ajahetkel iseteenindusest välja kutsuda, modelleeris autor järgnevusdiagrammid.

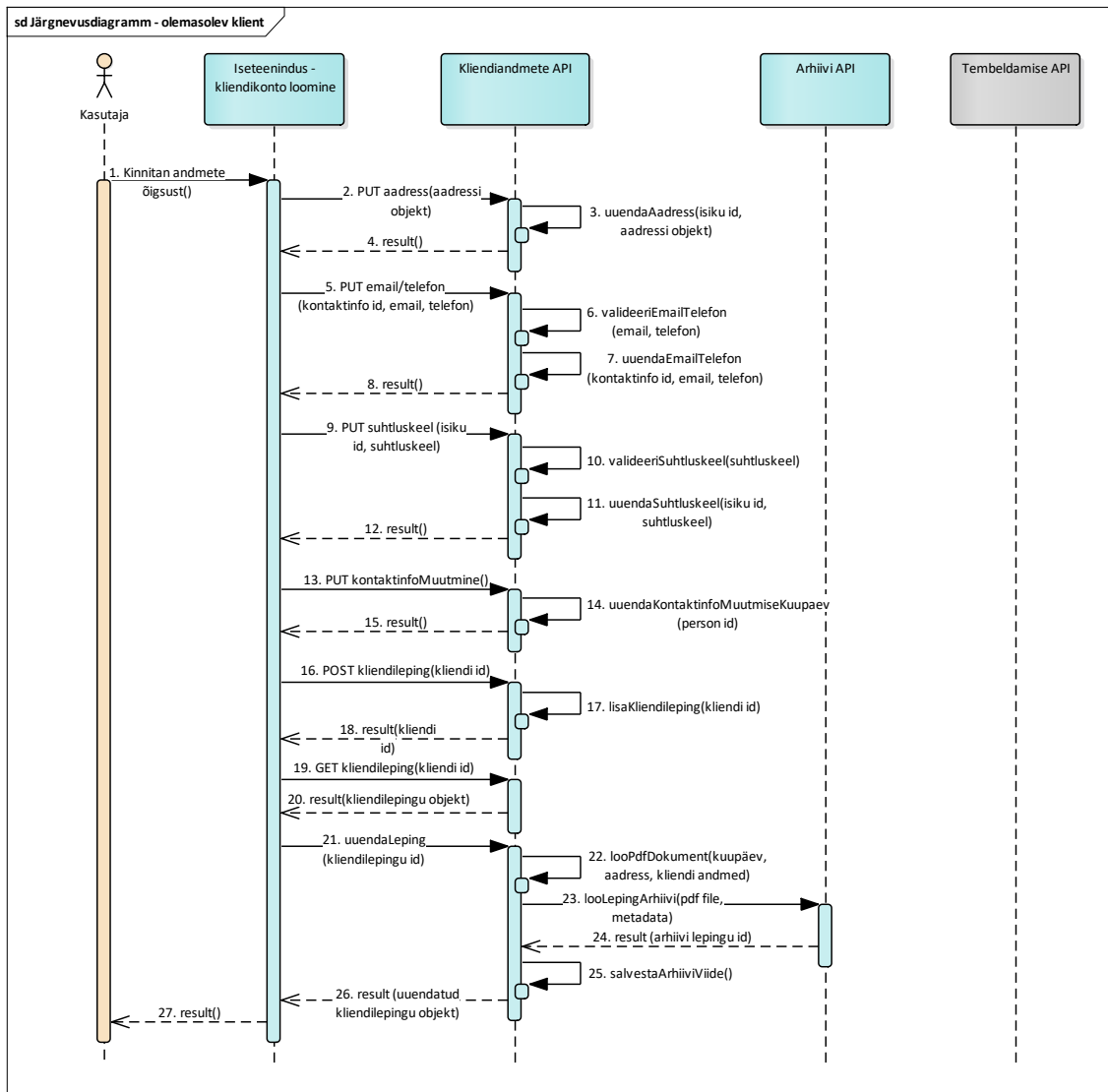
#### **Järgnevusdiagramm täiesti uue kliendi puhul**



Joonis 13. Järgnevusdiagramm - uus klient.

## Järgnevusdiagramm kliendiandmebaasis eksisteeriva kliendi puhul

Näitena on modelleeritud kliendikonto loomine sellise kliendi puhul, kellel on olemas kõik kohustuslikud kontaktandmed (telefon, e-post, aadress, suhtluskeel) ning ta muudab neid kõiki. Antud kliendil puudub andmebaasis kliendilepingu kirje.



Joonis 14. Järgnevusdiagramm - olemasolev klient.

Alates 27ndast punktist algab sama tegevuste jada, mis eelmisel järgnevusdiagrammil algab 19ndast punktist „Vaata kliendileping()“. Tembeldamise API on toodud joonisel hallina välja, kuna sõlmitud lepingud sinna hiljem jõuavad. Dubleerimise vältimiseks kahel diagrammil jättis autor selle osa välja.

Järgnevusdiagrammide alusel kirjeldas autor tiimile *back end* tööd liidestamaks Diili iseteenindus CRM API-ga vajalike andmete kätte saamiseks. Funktsionaalsetest ja mittefunktsionaalsetest nõuetest lähtuvalt kirjeldas autor esitluskihi (*front end*) arendustööd.

## 4 Kliendikonto loomise üleviimine mikroveebi

Käesolevas peatükis kirjeldab lõputöö autor, kuidas kliendikonto loomise funktsionaalsus mikroveebi liigutati, milliseid eeldustöid selleks tuli teha ja kuidas käis kogu ülemineku planeerimine.

### 4.1 Meeskond ja töökorraldus

Meeskonnas, mis tegeles kliendikonto funktsionaalsuse liigutamisega mikroveebi ja TO-BE protsessile, oli 5 arendajat, tooteomanik ja analüütik. Neist viimane oli siinkohal ka tiimijuhhi rollis. Testija rolli eraldi ei eksisteerinud ja seda pidid vajadusel suutma täita kõik tiimi liikmed. Käesolev osa laiendab alampeatükkides 2.3 ja 2.4 rollide jaotust arendusmeeskonna vaates. Meeskonna liikmete vastutus projekti vaates oli järgmine:

**Tooteomanik (ingl k *Product Owner*)** – on sise- ja väliskliendi esindaja ja funktsionaalsuse mikroveebi liigutamise tellija. Tema otsustab ärioluliselt, milliseid muudatusi, kui üldse, ärioloogikasse tehakse ja milline on uue funktsionaalsuse visuaal mikroveebis. Tooteomanik vastutab selle eest, et uus protsess oleks kliendile võimalikult lihtne ja arusaadav. Järjestab tiimi töid (ingl k *backlog*) koostöös tiimijuhhiga selliselt, et kõrgema prioriteediga tööd saaksid eelisjärjekorras valmis, et parimal viisil saavutada vajalikud eesmärgid ja kindlustab, et arendusmeeskond saab planeeritavatest töödest vajalikul määral aru.

**Analüütik/tiimijuhh** – suhtleb tooteomaniku ja kasutajatega selgitamiseks välja nõuded kliendikonto loomisele mikroveebis. Vastavalt äri- ja süsteemianalüüsi tulemustele kirjeldab analüütik vajalikud arendustööd ja määrab nendevahelised sõltuvused, on arendajatel tekkivate küsimuste esimene pidepunkt. Viib läbi tseremooniat arendustööde ülevaatamiseks, hindamiseks, töösse planeerimiseks ja tagasiside kogumiseks. Annab aru tiimi käekäigu kohta ja juhib igapäevaseid tegevusi. Seda rolli täitis kliendikonto mikroveebi liigutamise projektis käesoleva lõputöö autor.

**Arendaja** – vastutab toimiva ja nõuetele vastava lahenduse projekteerimise ja ehitamise eest. Töötab tihedas koostöös teiste arendajate ja analüütikuga. Vastavalt vajadusele

täidab koodiülevaataja rolli teistele tiimisisestele ja -välistele arendajatele. Testib teise arendaja poolt loodud funktsionaalsust, annab tagasisidet.

Tiimiväliste osapooltena on kaasatud veebidisainer, veebikomponentide arendaja, CRM-i mikroveebide arhitekt ja CRM-i valdkonna lahenduse arhitekt.

#### 4.1.1 Tseremooniad

- Igapäevane *stand-up* kestvusega 15 minutit, kus meeskonna liikmed vastavad kolmele küsimusele:
  - Mida tegin eile?
  - Mida on plaanis täna teha?
  - Millised takistused töös esinevad?
- Iganädalane koosolek tööde ülevaatamiseks ja hindamiseks. Arendustööde hindamiseks kasutatakse *planning poker* tehnikat, kus igale arendajale jagatakse kaardipakk. Arendajad hindavad iga töö keerukust skaalal 1, 2, 3, 5, 8, 13. Kogu kohtumine koosneb järgmistest sammudest:
  - Analüütik teeb lühiülevaate töödest, mis hindamisele tulevad. Arendajatel on võimalik esitada täpsustavaid küsimusi, millele analüütik ning koosolekul osalev tooteomanik vastused üritavad leida.
  - Arendajad annavad tööle keerukuse hinnangu ja asetavad vastava kaardi tagurpidi enda ette.
  - Arendajad keeravad enda kaardid ümber.
  - Arutelu, kus vähim ja enim punkte andnud arendajad selgitavad, kuidas nad antud punktideni jõudsid. Lõplike punktide määramiseks tuleb jõuda konsensusele.
  - Pärast arendustööde hinnangut pannakse kokku hinnatud tööde järjekord, mille alusel arendajad töid hakkavad lahendama. Järjekord võib ajas muutuda, kuid suuremad muutused (mõne prioriteetsema arendustöö esiletõstmine) kommenteeritakse jooksvalt kogu tiimile.



- Iganädalane CRM-i valdkonna ülene demo, kus arendajad näitavad äritellijale ja teistele meeskondadele, mis nädala ajaga valmis on saanud.
- Vastavalt vajadusele (tüüpiliselt iga 2 nädala tagant) retrospektiiv, kus tiimi liikmed saavad välja tuua, mis on halvasti läinud, mis hästi ja millised sammud tuleb astuda, et probleemid lahendada.

## 4.2 Eeldustööd funktsionaalsuse liigutamiseks mikroveebi

Kliendikonto loomise funktsionaalsuse liigutamise eeldustööd saab suures pildis liigitada kaheks: ärilised eeldustööd ja tehnilised eeldustööd.

### 4.2.1 Ärilised eeldustööd

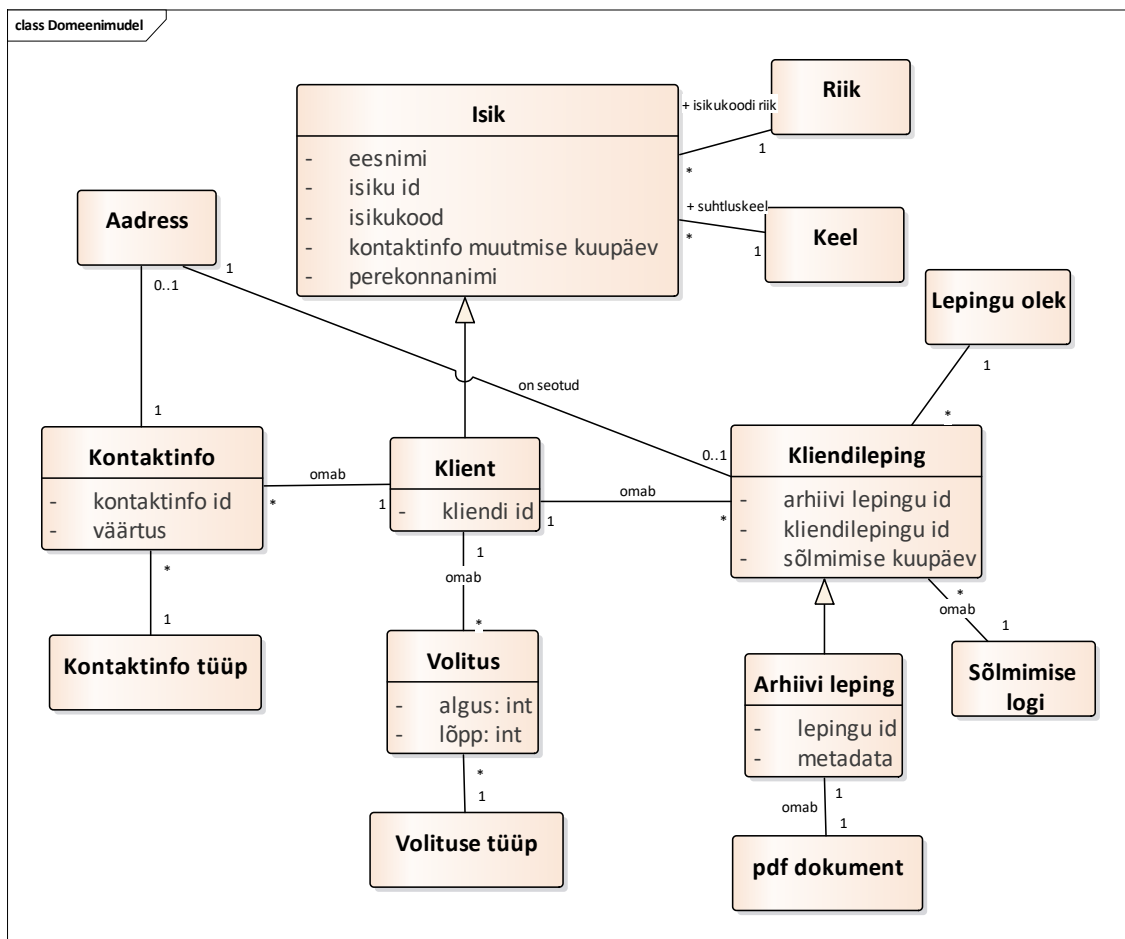
Ärianalüüsi käigus selgitas lõputöö autor välja järgmised kliendikonto loomise funktsionaalsusega seotud aspektid:

- Kuna kliendikonto loomist on vaja võimaldada mitmes mikroveebis (Diili iseteenindus, Telia iseteenindus, E-pood), võeti vastu otsus realiseerida kliendikonto loomise funktsionaalsus jagatava komponendina teegis. Nii vähendatakse koodi dubleeritust. Samuti on lihtsam viia läbi muudatusi kliendikonto loomise loogikas ning hoida funktsionaalsust mikroveebides *up-to-date*.
- Liferays realiseeritud kliendikonto loomise funktsionaalsuses oli lisaks kliendilepingu loomisele eraldi sammuna ka andmete töötlemise ja personaalsete pakkumiste saamise nõusolekute andmine. Kuna kliendikonto loomine ei sõltu eelpool mainitud nõusolekute olemasolust, sai nende küsimine kliendikonto loomise funktsionaalsusest välja tõstetud ning nõusolekuid küsitakse nüüd eraldi tegevusena.
- Olles tegeletud andmebaasis asuvate andmete kvaliteeti kirjeldavate kontrollpäringutega saadi aru, et suur osa kontaktaadressi salvestamise keerukusest tuleneb halvast andmekvaliteedist. Seega selleks, et vähendada kontaktaadressi lisamise keerukust, võeti esmalt ette kliendiandmebaasis asuvate kliendi kontaktaadresside parandustööd:

- Esmalt võeti vastu otsus, et arveaadresse kliendiandmebaasis ei hoita. Seda eelkõige seetõttu, et arveaadresside näol pole tegemist kliendi juriidilise aadressiga, vaid mõne konkreetse arvega seotud aadressiga. Nende puhul on tegemist arvelduse valdkonna andmetega ja seega CRM-i valdkonna protsessides neid ei kasutata.
- Seejärel tehti analüüs, et kas on olemas tarbijaid, kes küsivad ja kasutavad arveaadresse just kliendiandmebaasist. Lõputöö autor suhtles erinevate valdkondade süsteemide arhitektidega ja koostöös jõuti järeldusele, et arveaadresse kliendiandmebaasist ei kasutata.
- Pärast analüüsi kustutati ära kliendiandmebaasis asuvad arveaadressid. Sealjuures tuli tagada, et aadresside kustutamist ei sünkroniseeritaks Elioni ja EMT-i kliendiandmebaasidesse, kuna seal toimivad vanad arvelduse protsessid. Lõputöö autor kirjutas arveaadresside kustutamiseks vajalikud skriptid ning käivitas need. Enne skriptide käivitamist oli kliendiandmebaasis 1 527 231 aadressi ning pärast kustutamist 1 108 078, seega sai andmebaasist eemaldatud 30% aadressidest [18].
- Nüüd leiti, et erinevad iseteenindused käitusid kontaktaadresside salvestamisel valesti. Kui kliendil oli olemas aadress, mis ei olnud märgitud peamiseks (põhiliseks) aadressiks, lisati iseteeninduses aadressi muutmisel andmebaasi uus aadress juurde. Tehti täiendused, et olemasoleva aadressi puhul ei lisataks juurde mitte uut aadressi, vaid uuendatakse olemasolevat ning märgitakse see peamiseks. Lõputöö autor kirjeldas arendustööd *frontend* rakenduste loogika muutmiseks aadresside osas.
- Kui kliendil oli baasis kaks valiidset mitte-peamist kontaktaadressi, otsustati alles jätta see aadress, mis oli ettevõtete ühinemisel migreeritud Elioni kliendiandmebaasist, kuna erinevalt EMT-i poolsest kliendiandmebaasist oli see liidestatud aadressiregistriga. Lõputöö autor kirjutas skriptid aadresside viidete ümber tõstmiseks ja ilma viiteta jäänud aadresside kustutamiseks.

- Planeeritud said *backend* täiendused, et *frontend* peaks arvestama ainult aadressi olemasoluga ning ei peaks tegelema aadressi peamiseks märkimisega.
- Tooteomanikuga koostöös võeti vastu otsus, et kui Liferays oleva kliendikonto loomise puhul on kohustuslikud väljad üks kahest: kas e-mail või telefoninumber, siis uue funktsionaalsuse puhul on nii e-mail kui ka telefoninumber kohustuslikud väljad.

Kliendikonto loomisel tekkivaid objekte ja nendevahelisi sõltuvusi kirjeldab järgnev domeenimudel:



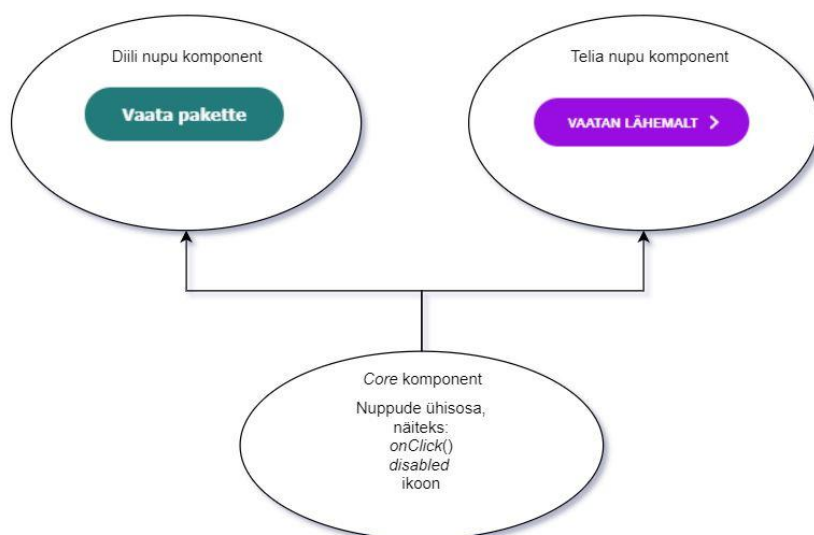
Joonis 15. Domeenimudel.

#### 4.2.2 Tehnilised eeldustööd

Kõige mahukamaks tehniliseks eeldustööks kujunes ärianalüüsist ilmnenu vajadus teostada kliendikonto loomise funktsionaalsus jagatava komponendina. Selle tarbeks sai

loodud uus teek, millega autori juhitud tiim polnud varem kokku puutunud. Loodud teeki oli vaja seadistada sõltuvused, *deploy* loogika, põhiline abistav funktsionaalsus (näiteks API poole pöördumise konfiguratsioon).

Arvestades, et kliendikonto loomise funktsionaalsust pidi saama kasutada erinevates mikroveebides, millel võib olla erinev välimus, pidi loodud teek olema nõ „stiilivaba“. See tähendab, et loodud teegis kasutusel olevad *frontend* komponendid ei oma stiili ning rakendus, mis võtab teegist mõne loogilise komponendi kasutusele, peab teegile ütlema, milline see komponent oma stiililt on (milline välja näeb). See eeldas veebikomponentide arendajate poolt refaktooringut olemasolevatele stiilidele, mille tulemusena koondati ühisosa ühtsesse *core* koodihoidlasse.



Joonis 16. *Frontend* komponentide täiendatud loogika.

Järgmisena tehti olemasolevale mikroveebile refaktooring. Ettevõttes võeti vastu arhitektuuriline otsus, et uute arenduse puhul *hateoas*-i linke ei kasutata. Seetõttu täiendati olemasolevat teenuste poole pöördumise loogikat nii, et *hateoas* linkide asemel kasutataks pöördumiste tegemisel rakendusse sissekirjutatud teenuste URL-e, kus muutuv osa asendati baasikirjete identifikaatoritega. Lisaks täiendati mikroveebi *backend*-i kliendisessiooni hoidmise loogikat, kuna selle puhul oli varasemalt arvestatud, et kliendikonto loomine toimub Liferays.

Veel täiendati kliendisessiooni loomise loogikat ning kliendiandmete API teenuseid lisaandmetega.

### **4.3 Kliendikonto loomise funktsionaalsuse üleviimine jagatud teeki**

Kliendikonto loomise loogika viidi jagatava komponendina teeki peatükis 3 kirjeldatud süsteemianalüüsi alusel. Funktsionaalsuste arendus jagati kolmeks suuremaks tükiks:

1. Kontaktaadressi komponendi loomine. Kontaktaadressiks saab olla nii Eesti aadress kui ka välismaa-aadress, kusjuures Eesti aadressi puhul on olemas liidestus aadressiregistriga.
2. Kliendiandmebaasis juba eksisteeriva kliendi kliendilepingu allkirjastamine, mis kasutas punktis 1 mainitud kontaktaadressi komponenti.
3. Uue, kliendiandmebaasis puuduva kliendi konto loomine ja kliendilepingu allkirjastamine, mis kasutas samuti punktis 1 välja toodud kontaktaadressi komponenti.

Sellise funktsionaalsuste tükeldamise põhjenduseks on kontaktaadressi loogika keerukus, mis on juba varasemalt mainitud ja olemasoleva kliendi ning uue kliendi puhul taustal olevate teenuste erinevus.

### **4.4 Järeldused**

Olles viinud kliendikonto loomise loogika mikroveebi on näha, et Liferays olevatel funktsionaalsustel ühisosa pigem puudub. See tähendab, et iga uus, loodav komponent on erinev ja mingit üldist protsessi, kuidas funktsionaalsust üle tuua, kirjeldada ei saa.

Samas polnud suuremad tehnilised eeldustööd kuidagi seotud konkreetse üle toodava funktsionaalsusega, seega neid võiks pidada ka tulevaste platvormi vahetust nõudvate funktsionaalsuste eeldustöödeks.

Kliendikonto loomise loogika liigutamisel jagatavasse teeki sai selgeks sellise komponentide ülesehituse positiivne pool. Koodi ei dubleerita ning muudatused äriloogikas on hallatavad ühes koodihoidlas. Tulevikus on eesmärk tõsta samuti kõik

funktsionaalsused, mis võiksid olla vähegi kasutusel mitmes rakenduses, jagatavasse teeki.

Funktsionaalsuste liigutamine monoliitrakendusest mikroveebidesse annab peale kõige muu veel võimaluse kahelda varasemalt tehtus ning optimeerida ja parendada nii koodi- kui ka ärioloogikat

## 5 Küsitlus ja tulemuste analüüs

Autor koostas Telia arendajatele küsitluse, et uurida, kuidas mikroveebide arhitektuuri stiili juurutamine ja Liferay veebiplatvormi järk-järguline sulgemine on mõjutanud nende igapäevast tööd ning millised on nende mured/rõõmud sellega seoses.

Küsitluse saab jagada kolmeks (kõikide esitatud väidete ja küsimustega on võimalik tutvuda käesoleva lõputöö lisas Lisa 5 - Küsitlus):

1. Liferay veebiportaali arendust puudutavad väited;
2. mikroveebide arendust puudutavad väited;
3. mikroveebide suurimad plussid ja miinused arendajate vaates ning muud tähelepanekud.

Esimese kahe punkti väidetele tuli arendajatel anda hinnang järgmisel skaalal:

- 1 – ei ole üldse nõus;
- 2 – pigem ei nõustu;
- 3 – ei oska öelda;
- 4 – pigem nõustun;
- 5 – olen täiesti nõus.

### 5.1 Küsitluse tulemused

Küsitluse esimesele poolele, mis puudutab Liferay arendusi, vastas 15 arendajat. Küsitluse teisele poolele, mis puudutab mikroveebide arendusi, nende plusse ja miinuseid ning muid mõtteid seoses selle arhitektuuriga, vastas 10 arendajat.

Esimene väide, mille toetus autor palus 5 punkti skaalal hinnata oli „Ma tean, mis on mikroveebid ja kuidas see arhitektuuri stiil puudutab Teliat“. Selle küsimuse kaalutud

keskmiseks hindeks 15 vastaja hulgas tuli 4.25, seega võib öelda, et kõikidel vastanud arendajatel on ülevaade mikroveebide olemusest ja nende juurutamisest Telias.

Teise küsimusena palus autor arendajatel anda hinnang Liferay arendusi puudutavatele väidetele (Hinda järgmiste Liferay arendusi puudutavate väidete tõesust 5 palli skaalal.Liferay puhul:) ja tulemused on järgmised:

Tabel 3. Liferay arendusi puudutavad väited - vastused.

	<b>Ei ole üldse nõus</b>	<b>Pigem ei nõustu</b>	<b>Ei oska öelda</b>	<b>Pigem nõustun</b>	<b>Olen täiesti nõus</b>	<b>Vastajate arv</b>	<b>KK</b>
võtab rakenduse lokaalne püstipanek oluliselt kauem aega võrreldes mikroveebidega	1	2	2	3	7	15	<b>3.87</b>
on arendusprotsess kiire ja lihtne	5	8	2	0	0	15	<b>1.8</b>
on testimise maht suurem võrreldes mikroveebidega	1	2	5	6	1	15	<b>3.27</b>
on vea tekkimisel tootmiskeskonnas (live) selle paranduse reliisimine kiire	7	5	3	0	0	15	<b>1.73</b>
on erakorralise reliisi korraldamine lihtne ja valutatu protsess	7	4	4	0	0	15	<b>1.8</b>
on võimalik funktsionaalsused kiiresti kliendini viia	4	9	2	0	0	15	<b>1.87</b>



on portlet-ide arendus mugav ja lihtne	2	9	4	0	0	15	<b>2.13</b>
ei nõua arendamine minult lisateadmisi (st pole spetsiifilisi oskusi vaja)	4	6	2	3	0	15	<b>2.27</b>
on mul arendajana rohkem stressi, kuna vea tekkimisel on selle paranduse reliisimine keerukas ja aeganõudev	1	1	3	8	2	15	<b>3.6</b>
on võimalik kiiresti tuvastada, milles viga, kui tootmiskeskonnas (live) midagi ei toimi	3	9	2	1	0	15	<b>2.07</b>
on kommunikatsioonivajadus erinevate valdkondade vahel suurem, kui mikroveebide puhul	0	2	4	6	3	15	<b>3.67</b>

KK- kaalutud keskmine

Tabeli 3 põhjal võib öelda, et väited, millega arendajad olid kõige vähem nõus, puudutasid eelkõige arendus- ja reliisimisprotsessi. Tulemused võib suures osas jaotada kahte gruppi:

### **Kiirus**

- Suurem osa arendajatest nõustub, et Liferay rakenduse lokaalne püstipanek võtab oluliselt kauem aega, kui mikroveebi puhul. Seega tekib kiiruse osas langus juba enne arendama hakkamist.

- Liferay uut versiooni ei ole võimalik igal ajahetkel toomiskeskonda paigaldada ja selle reliisimise toimub läbi keskse muudatuste halduse protsessi. Seega on vastanud arendajad arvamusel, et arendus- ja reliisimisprotsess on aeglane, vea tekkimisel erakorralise reliisi planeerimine keeruline ja arendused ei jõua piisavalt kiiresti kliendile kasutusse.
- Arendajad on pigem seisukohal, et Liferay arendamine nõuab neilt teatud spetsiifilisi oskusi, mis võib kaudselt mõjutada negatiivselt arenduste kiirust. Vastajad on arvamusel, et Liferay *portlet*-ide (vaadete) arendamine ei ole mugav ja lihtne.
- Testimise mahu erinevuse osas on arendajad neutraalsel seisukohal. See tähendab, et üldiselt ei näe nad, et testimise maht on Liferay rakenduse puhul suurem võrreldes mikroveebidega. Põhjus võib seisneda selles, et Liferay puhul viivad regressioonitestimise (ingl k *regressioon testing*) läbi vastava valdkonna testijad ja see võib jääda suures osas arendajale märkamatuks tegevuseks.

## **Vabadus**

- Suurem osa vastajatest tunneb, et neil on Liferay veebirakenduse arendamisel rohkem stressi, kuna esiteks on veakoha tuvastamine selle ilmnemisel keeruline ja teiseks ei ole võimalik sõltumatult versiooni vahetada.
- Autori üllatuseks on vastajad pigem nõus sellega, et kommunikatsioonivajadus erinevate valdkondade vahel on Liferay arendamisel suurem. Üldiselt on ka Liferay puhul jaotatud, milliste konkreetsete lehtede eest milline osakond vastutab. Ka mikroveebide puhul on teatud juhtudel vaja teiste valdkondadega kommuniqueerida (näiteks, kui on äril soov kuskil kuvada teise valdkonna andmestikku). Samas ei anna küsimuse püstitus ülevaadet sellest, mida vastajad täpselt selle vastusega mõtlesid. Võib oletada, et Liferay puhul on protsessi üheks osapooleks reliisihaldus, mistõttu kokkupuude sealsete inimestega on arendajatel küll lisaks, võrreldes mikroveebide arendamisega.

Kokkuvõttes võib öelda, et Liferay puhul ei pärsi arenduste kiiresti kliendini jõudmist mitte ainult aspekt, et rakendust reliisitakse selle suurte sõltuvuste tõttu harva, vaid

arendused nõuavad arendajatelt teatud portaali spetsiifilisi teadmisi. Lisaks on toodud välja ka suurenenud kommunikatsioonivajadus erinevate osapoolte vahel.

Kolmanda küsimusena palus autor vastajatel anda hinnang mikroveebide arendusi puudutavatele väidetele (Hinda järgmiste väidete tõesust 5 palli skaalal. Tunnen, et seoses mikroveebide (*micro services to the frontend world*) implementeerimisega:) ja tulemused on järgmised:

Tabel 4. Mikroveebide arendusi puudutavad väited - vastused.

	<b>Ei ole üldse nõus</b>	<b>Pigem ei nõustu</b>	<b>Ei oska öelda</b>	<b>Pigem nõustun</b>	<b>Olen täiesti nõus</b>	<b>Vastajate arv</b>	<b>KK</b>
lisavad uued arendajad kiiremini väärtust	0	1	1	4	4	10	<b>4.10</b>
on rakendused rohkem kapseldatud ja loogikat on palju vähem	0	1	1	6	2	10	<b>3.90</b>
on probleemide ( <i>bug</i> ) lahendamine ja reliisimine kiirem	0	0	2	3	5	10	<b>4.30</b>
on esile kerkinud suurem <i>full-stack</i> arendajate vajadus	0	2	6	1	1	10	<b>3.10</b>
on mul arendajana võimalik panustada <i>end to end</i>	0	2	0	4	4	10	<b>4.00</b>

on võimalik suhteliselt sõltumatuna arendada ja reliisida	0	1	0	4	5	10	<b>4.30</b>
saavad arendused kiiremini valmis	0	1	1	4	4	10	<b>4.10</b>
on võimalik funktsionaalsused kiiremini kliendini viia	0	1	0	4	5	10	<b>4.30</b>
on mul arendajana rohkem autonoomiat, vabadust ja vastutust	0	0	0	6	4	10	<b>4.40</b>
on mul arendajana vähem stressi, kuna alati on võimalik eelnev versioon kiiresti tootmiskeskonda paigaldada	0	0	0	6	4	10	<b>4.40</b>
olen arendajana produktiivsem	0	0	2	6	2	10	<b>4.00</b>
on võimalik kiiremini tuvastada, milles on viga, kui tootmiskeskonnas ( <i>live</i> ) midagi ei toimi	0	0	1	6	3	10	<b>4.20</b>
on mul arendajana võimalik rohkem kaasa rääkida võimalike lahenduste osas	0	1	1	5	3	10	<b>4.00</b>

KK – kaalutud keskmine

## **Kiirus**

- 80% vastanud arendajatest ütleb, et nende hinnangul lisavad uued arendajad kiiremini väärtust. Ilmselt on see tingitud sellest, et arendajal tuleb töötada väiksema koodibaasiga ja ärioloogikat, mille peale peab mõtlema, on vähem. Näiteks CRM-i valdkonnas on loodud kolm erinevat meeskonda, kes vastutavad tervikuna teatud ärioloogika ja sellega seotud andmete eest. Seetõttu on meeskonnad väiksemad ja teemad paremini kapseldatud.
- Enamik vastajaid on nõus, et vigade parandamine ja tootmiskeskonda paigaldamine on kiirem võrreldes Liferay arendustega. 80% arendajatest ütleb, et arendused saavad kiiremini valmis. Põhjus võib olla selles, et domeeni ärioloogika tundmine meeskondades on parem ja puudub Liferay arendustega kaasnev lisakeerukus.
- Samuti on vastajad nõus, et vigade tekkimisel on mikroveebide puhul võimalik kiiremini aru saada, kus viga konkreetselt tekkinud on ja mida tuleb parandada.

## **Vabadus**

- Suur osa arendajatest tunneb, et seoses mikroveebide arhitektuuri juurutamisega on neil võimalus lahenduste osas rohkem kaasa rääkida. Iga mikroveebi eest vastutab meeskond, seega on ka lahendused suuresti tiimi välja mõelda.
- Kõik vastajad on nõus, et mikroveebi arendamisega on neil tunnetuslikult vähem stressi, kuna alati on võimalus vea tekkimisel kohe eelmine versioon või vea parandus tootmiskeskonda paigaldada.
- Lisaks on arendajad nõus, et mikroveebide arendamisel on neil rohkem autonoomiat, vabadust ja samas ka vastutust. Võimalik on arendada teistest tiimidest sõltumatult ja võtta vastu otsused, milliseid tehnoloogiaid kasutada.

Kokkuvõttes on arendajad mikroveebide arendamise ja reliisimise suhtes oluliselt positiivsemalt meelestatud, kui Liferay arenduste puhul. Nad tunnevad, et arendus- ja reliisimisprotsess on kiirem ja tänu mikroveebidest tulenevale autonoomiale on neil vähem stressi ja rohkem vabadust.

Järgmiseks esitas lõputöö autor kolm avatud vastustega küsimust:

1. Mis on Sinu arvates mikroveebide suurim miinus?
2. Mis on Sinu arvates mikroveebide suurim pluss (kasu)?
3. Kui Sul on muid mõtteid (mured/rõõmud) seoses Telia valdkonnapõhiste mikroveebidega, siis kirjuta sellest siia.

**Mikroveebide suurimateks miinusteks** toodi välja järgnev:

- Igäühe arhitektuur kipub erinev kujunema ja ühtlust on vähevõitu. Samalaadsetel rakendustel võib tekkida ebavajalikke erisusi sama loogika implementeerimisel.
- 4 vastajat tõid välja, et mikroveebides esineb palju dubleeritud koodi (logimine, *deploy*, monitooring).
- Mitu vastajat on esile toonud, et rakenduste üleste muudatuste koordineerimine on keeruline. Näitena on toodud Telia rakenduste ülese menüü uuendamine, kus iga rakendus peab uue versiooni vastavalt kasutusele võtma, mistõttu peab iga rakenduse vastutavat tiimi teavitama, et seda tehtaks.
- Lisaks on välja toodud, et keeruline on näha tervikut ja aru saada, kuidas erinevad mikroveebid omavahel integreeruvad. Sealjuures on 1 vastaja maininud, et iga meeskond keskendub konkreetse mikroveebi arendustele ning puudub lai silmaring kogu süsteemi arendustest.

Mikroveebide arhitektuuri stiili **suurimaks kasuks** toodi välja järgnev:

- Väikesed domeenipõhised rakendused, mille sisu on ühe arendustiimi vastutada. Vastutav tiim on süsteemiga kursis ja oskab kiiremini vigu parandada või uusi funktsionaalsuseid arendada.
- Väikesed kiired reliisid ja kiire vigade lahendamine.
- Arenduse poolest oluliselt lihtsam võrreldes monoliitse rakendusega. Koodi on vähem, vajalikud failid leiab kiiresti üles.
- Arenduste kiirus ja vabadus. Lihtsam on arendada ja hallata.

Viimase küsimuse osas, mis puudutas muid muresid/rõõme seoses mikroveebidega, toodi kaudselt välja samu aspekte, mis eelmise kahe küsimuse vastustest juba esile tulid. Lisaks toodi veel välja, et osades valdkondades on olukord, kus esitluskiht on küll mikroveeb,

kuid *API* on endiselt keskses reliisitsükklis. See tähendab, et vana rakenduse reliisihaldus pärsib mikroveebi igal ajahetkel reliisimist.

## 5.2 Lahendusettepanekud

Vastavalt küsitlusest välja tulnud murekohtadele seoses mikroveebidega, pakub autor välja võimalikud lahendused.

### Erinev arhitektuur mikroveebides

Lahendus on luua mikroveebi näidisrakendus (ingl k *template application*), kus on kõik vajalik konfiguratsioon, liidestused jne. Selleks, et näidise sisu kokku leppida, on vaja uurida arendajatelt, mida nende arvates see sisaldama peaks. Näidisrakendus peaks koosnema sellest, mis on kõikides mikroveebides ühine ja ei sõltu sinna ehitatavatest funktsionaalsustest.

### Dubleeritud kood

Üks ühele dubleeritud koodi aitab vältida jagatud teekide (ingl k *library*) loomine. Näiteks antud lõputöös käsitletud kliendikonto loomine sai selliselt realiseeritud, kuna erinevates rakendustes on vaja sama funktsionaalsus kasutusele võtta. Koodi dubleerimise ja jagatud teekide ehitamine on muidugi suur dilemma, kuna ühelt poolt ei soovita sama koodi mitmesse rakendusse kirjutada, kuid teisalt suureneb jagatud teegiga rakenduste vaheline sõltuvus. Näiteks kui kliendikonto loomise jagatud komponenti kasutab 3 rakendust ja selles komponendis tuleb teha muudatus, siis kõik tarbijad peavad kasutatava teegi versiooni uuendama.

### Arendajal on keeruline näha tervikut

Selle probleemi juures on võimalik panna arendajat nägema enda meeskonna arendatavast mikroveebist ja selles sisalduvast funktsionaalsusest kaugemale:

1. Koodiülevaatuste jagamine üle meeskondade. Lõputöö autori tiimis seda jõudumööda juba katsetatakse ja see mõjub positiivselt mõlemale osapoolle – üheltpoolt saavad ühe tiimi arendajad ülevaate teise meeskonna arendustest ja teisalt on võimalik koodiülevaate soovijal õppida ja saada oma kirjutatud koodile tähelepanekuid uue nurga alt.

2. Valdkonnaülene demo arenduste esitlemiseks erinevate meeskondade vahel. See on midagi, mida käesoleval ajal juba CRM-i valdkonnas tehakse iganädalaselt. Lõputöö autor pakub siia juurde äriliste demode vaate. See tähendab, et lisaks sellele, et arendajad näitavad arendusi, mis viimase paari nädala jooksul tehti, räägivad ka äri poole esindajad, mida on lähiajal tiimidesse arendusse oodata. See aitab erinevatel sama valdkonna meeskondadel mõista suuremat pilti ja näha seeläbi süsteemset tervikut.

### **Rakenduste üleste muudatuste koordineerimine**

Täna on mikroveebid üles ehitatud selliselt, et jagatud teegi uue versiooni puhul tuleb igas rakenduses see vastavalt kasutusele võtta. Lahenduseks võiks olla uue versiooni kasutuselevõtu automatiseerimine. On võimalik kasutusele võtta *latest* parameeter ja sellisel juhul tõmmatakse rakenduse käivitamisel alla viimane versioon jagatud teegist [21].

### **5.3 Küsitluse tulemuste kokkuvõte**

Küsitlusele vastajate poolt välja toodud mikroveebide arhitektuuri stiili positiivsed ja negatiivsed küljed kattuvad suures osas käesoleva lõputöö peatükis 2 välja toodud kirjandusest leitud. Mikroveebide arendamine annab arendustiimile teatava autonoomsuse – meeskond vastutab selle arendamise, reliisimise ja ülalhoiu eest, sõltumata sealjuures teiste tiimide koodibaasist. Arendajad tunnevad, et neil on võimalik rohkem kaasa rääkida, kuidas lahendused on üles ehitatud ning panustada sealjuures *end to end*. Lisaks on tunnetuslikult kiirem uute arendajate sisseelamine, kuna äri loogika ja koodibaas mikroveebi puhul on paremini hoomatav võrreldes monoliitse veebirakendusega.



## 6 Kokkuvõtte

Käesolev lõputöö käsitleb funktsionaalsuste liigutamist monoliitset esitluskihi rakendusest valdkonnapõhistesse mikroveebidesse. Sellega saavutatakse olukord, kus kliendile väärtust pakkuvad funktsionaalsused jõuavad senisest kiiremini kättesaadavaks, arendustiimi liikmed tunnevad suuremat vastutust, iseseivust ja sõltumatust teistest arendusmeeskondadest.

Töös on analüüsitud Liferay veebiplatvormi arendusprotsessi ning mikroveebi oma. Kahele protsessile on teostatud võrdlus ning toodud välja põhjused, miks üleminek vajalikuks osutus. Lisaks on selgitatud mikroveebide olemust ja kirjeldatud selle arhitektuuri stiili juurutamise tulemusena saadud kasu nii organisatsiooni kui ka arendusmeeskonna vaates.

Magistritöö käsitleb kliendikonto loomise funktsionaalsust iseteeninduses ning selle näitel on kirjeldatud, kuidas mikroveebi liigutamine praktikas toimub. Autori poolt on antud funktsionaalsuse vaates teostatud äri- ja detailanalüüs, mis on eelduseks arendustööde kirjeldamiseks, eeldustööde vajadusest aru saamiseks ja funktsionaalsuse liigutamiseks mikroveebi.

Töö peamiseks tulemuseks on vastavalt analüüsi tulemustele ja kirjeldatud arendustöödele mikroveebi liigutatud funktsionaalsus, mida on võimalik lihtsasti kasutusele võtta ka muudes mikroveebides. Tähtsaimaks järelduseks vastavalt arendajate hulgas läbi viidud küsitluse tulemustele on asjaolu, et mikroveebide kasutuselevõtt on muutnud arendajate igapäevast tööd positiivses suunas. Nad tunnevad, et töötavad efektiivsemalt, arendused saavad kiiremini valmis ja mikroveebi kapseldatud äri loogika tõttu on neil vähem aspekte, millesse peavad süvenema. Vastanute sõnul pakuvad ka uued arendajad kiiremini väärtust.

Vastavalt küsitluse tulemustes välja tulnud probleemidele seoses mikroveebidega, on autori poolt välja pakutud lahendustepanekud nende leevendamiseks. Lahendustepanekuid on valdkonnasiseselt arutatud, need on arvesse võetud ja tegeletakse nende probleemide lahendamise suunas.

## Kasutatud kirjandus

- [1] „3. Andmebaaside erinevad kontseptid ja arhitektuurid,“ [Võrgumaterjal]. Available:  
[http://enos.itcollege.ee/~priit/1.%20Andmebaasid/1.%20Loengumaterjalid/03/69678909001\\_3.htm](http://enos.itcollege.ee/~priit/1.%20Andmebaasid/1.%20Loengumaterjalid/03/69678909001_3.htm). [Kasutatud 11 12 2019].
- [2] „E-teatmik,“ [Võrgumaterjal]. Available: <http://vallaste.ee/>. [Kasutatud 20 11 2019].
- [3] „What is JSON Web Token?,“ [Võrgumaterjal]. Available: <https://jwt.io/introduction/>. [Kasutatud 18 12 2019].
- [4] „Portlet,“ [Võrgumaterjal]. Available: <https://www.webopedia.com/TERM/P/portlet.html>. [Kasutatud 17 12 2019].
- [5] „What is a Portal?,“ [Võrgumaterjal]. Available: <https://web.liferay.com/products/what-is-a-portal/web-platform>. [Kasutatud 15 11 2019].
- [6] M. Mürk, „Arhitektuur,“ [Võrgumaterjal]. Available: Telia Eesti AS ettevõttesisene Confluence wiki. [Kasutatud 11 10 2019].
- [7] K. Wähner, „Pros and Cons – When to use a Portal and Portlets instead of just Java Web-Frameworks,“ [Võrgumaterjal]. Available: <https://dzone.com/articles/pros-and-cons-%E2%80%93-when-use>. [Kasutatud 20 11 2019].
- [8] N. Dragoni, S. Giallorenzo, A. L. Lafuente ja M. Mazzara, „Microservices: yesterday, today, and tomorrow,“ 20 4 2017. [Võrgumaterjal]. Available: <https://arxiv.org/pdf/1606.04036.pdf>. [Kasutatud 1 12 2019].

- [9] D. Merkel, „Docker: Lightweight Linux Containers for Consistent Development and Deployment,“ 19 3 2014. [Võrgumaterjal]. Available: <https://www.linuxjournal.com/content/docker-lightweight-linux-containers-consistent-development-and-deployment>. [Kasutatud 2 12 2019].
- [10] T. Söderlund, „Micro frontends—a microservice approach to front-end web development,“ 6 7 2017. [Võrgumaterjal]. Available: <https://medium.com/@tomsoderlund/micro-frontends-a-microservice-approach-to-front-end-web-development-f325ebdad16>. [Kasutatud 10 11 2019].
- [11] C. Jackson, „Micro Frontends,“ [Võrgumaterjal]. Available: <https://martinfowler.com/articles/micro-frontends.html>. [Kasutatud 30 11 2019].
- [12] M. Geers, „Micro Frontends,“ [Võrgumaterjal]. Available: <https://micro-frontends.org/>. [Kasutatud 12 11 2019].
- [13] J. Lewis, „Microservices,“ [Võrgumaterjal]. Available: <https://martinfowler.com/articles/microservices.html>. [Kasutatud 10 12 2019].
- [14] J. Gray, „A conversation with Werner Vogels,“ *ACM Queue*, 4(4), pp. 14-22, 30 6 2006.
- [15] M. Felix-Jayachandran, „Slaying the UI AngularJS Monolith Using Micro Frontends Architecture,“ [Võrgumaterjal]. Available: <https://medium.com/capital-one-tech/slaying-the-ui-angularjs-monolith-using-a-micro-frontend-architecture-f6c6332652d3>. [Kasutatud 14 11 2019].
- [16] Z. Dehghani, „How to break a Monolith into Microservices,“ [Võrgumaterjal]. Available: <https://martinfowler.com/articles/break-monolith-into-microservices.html>. [Kasutatud 04 11 2019].

- [17] „I don't understand micro frontends.“ [Võrgumaterjal]. Available: <https://medium.com/@lucamezzalira/i-dont-understand-micro-frontends-88f7304799a9>. [Kasutatud 05 12 2019].
- [18] „Telia Eesti AS kliendiandmebaas“.
- [19] „eID lühitutvustus,“ 2017. [Võrgumaterjal]. Available: [https://eid.eesti.ee/index.php/EID\\_1%C3%BChitutvustus](https://eid.eesti.ee/index.php/EID_1%C3%BChitutvustus). [Kasutatud 31 10 2019].
- [20] K. Wiegers ja J. Beatty, Software requirements. 3rd ed., Redmond, Washington: Microsoft Press, 2014.
- [21] „CLI documentation > Configuring npm,“ [Võrgumaterjal]. Available: <https://docs.npmjs.com/files/package.json>. [Kasutatud 1 12 2019].
- [22] „UML Use Case Extend,“ [Võrgumaterjal]. Available: <https://www.uml-diagrams.org/use-case-extend.html>. [Kasutatud 01 12 2019].
- [23] „Functional and Nonfunctional Requirements: Specification and Types,“ [Võrgumaterjal]. Available: <https://www.altexsoft.com/blog/business/functional-and-non-functional-requirements-specification-and-types/>. [Kasutatud 20 11 2019].
- [24] „Unified Modeling Language (UML) | Sequence Diagrams,“ [Võrgumaterjal]. Available: <https://www.geeksforgeeks.org/unified-modeling-language-uml-sequence-diagrams/>. [Kasutatud 10 11 2019].

# Lisa 1 - Kliendikonto loomise 1. samm

Eraklient **Ariklient** Russий TEST ISIK

**Telia** Mobiil TV Internet Lisateenused E-pood Pakkumised Abi | **Iseteenindus** Q

## KLIENDIKONTO LOOMINE

Hea Test!

Palume sul iseteeninduses Telia teenuste tellimiseks ja personaalse sisu nägemiseks luua Telia kliendikonto ja sõlmida kliendileping. Konto loomine võtab vaid hetke: sisesta või kontrolli oma andmeid ja täpsusta andmete töötlemise reegleid.

### TÄPSUSTA OMA KONTAKTANDMEID

Nimi	Isikukood
TEST ISIK	49410120091

Postiaadress

⚠ Email on nõutud väli ⚠ Mobiil on nõutud väli

### KAS SOOVID PAREMAID PAKKUMISI JA PERSONAALSET TEENINDUST?

Sinu parimaks teenindamiseks ning sulle personaalsete pakkumiste loomiseks on vajalik sinu nõusolek sinu teenuste kasutamise andmete töötlemiseks.

Nõustun minu interneti-, mobiilside-, telefoni-, televisiooni ja muude sideteenuste kasutus- ja asukohaandmete töötlemisega (sh profileerimisega) turunduslikul eesmärgil personaalsema teeninduse, pakkumiste ja teenuste saamiseks.

Soovin saada turunduspakkumisi e-posti, SMS-i ja teiste e-kanalite kaudu:

i Personaalsete pakkumiste puhul on lubatud kõik kanalid. Soovi korral saad neid hiljem muuta.

i Vajutades nupule "Jätkan" salvestatakse sisestatud andmed ja tehtud valikud ka siis kui kliendilepingut ei sõlmita.

Joonis 17. Kliendikonto loomise 1. samm

## Lisa 2 - Kliendikonto loomise 2. samm

Eraklient Ariklient Russkii TEST ISIK

Telia Mobiil TV Internet Lisateenused E-pood Pakkumised Abi | [Iseteenindus](#) Q

# KLIENDIKONTO LOOMINE

Hea Test!

Enne kliendilepingu allkirjastamist kontrolli üle sisestatud andmed ja enda antud nõusolekud. Samuti saad enne allkirjastamist soovi korral lepingutega tutvuda. Parast kontrollimist vajuta nuppu „Sõlmin lepingud“ ja kliendikonto ongi loodud.

### KONTAKTANDMED

Nimi	Isikukood	
TEST ISIK	49410120091	
Postiaadress	Korter	
Tormi tn 13, Pärnu linn, 80023 Pärnu linn, Pärnu maakond		
E-mail	Mobiil	Suhtluskeel
test.isik@gmail.com		Eesti

### KAS SOOVID PAREMAID PAKKUMISI JA PERSONAALSET TEENINDUST?

Sinu parimaks teenindamiseks ning sulle personaalsete pakkumiste loomiseks on vajalik sinu nõusolek sinu teenuste kasutamise andmete töötlemiseks.

Nõustun minu interneti-, mobiilside-, telefoni-, televisiooni ja muude sideteenuste kasutus- ja asukohaandmete töötlemisega (sh profileerimisega) turunduslikul eesmärgil personaalsema teeninduse, pakkumiste ja teenuste saamiseks.

[Laadin alla PDF-faili](#)

Soovin saada turunduspakkumisi e-posti, SMS-i ja teiste e-kanalite kaudu:

! Personaalsete pakkumiste puhul on lubatud kõik kanalid. Soovi korral saad neid hiljem muuta.

[Laadin alla PDF-faili](#)

### Palun tutvuda kliendilepingu sisuga

- Kliendileping on kliendisuhete raamleping, mis loob Kliendile võimaluse lihtsal viisil liituda Telia Teenustega, kasutada e-Keskkondi, osta kaupu ja sõlmida muid Lepinguid Teliaga.
- Telia osutab Kliendile Teenuseid ja müüb kaupu tingimustel, mis on toodud Üldtingimustes, Hinnakirjas, Teenuse kohta kehtivates Teenusetingimustes ja muudes Tingimustes. Telia viitab Tingimustele Kliendiga Lepingu sõlmimisel või talle Teenuse, kauba või e-Keskkonna pakkumisel ja võimaldab Kliendil Tingimustega tutvuda. Tingimused on Lepingu osaks ja Pooled kohustuvad neid järgima.
- Kliendilepingu sõlmimisega Klient kinnitab, et ta on Üldtingimuste ja Andmete kasutamise põhimõtetelega tutvunud ning kohustub neid täitma.
- Teenuste ja ostetud kaupade eest peab Klient maksma Teliale tasu vastavalt esitatud arvele sellel märgitud maksetähtajaks.
- Kliendileping kehtib alates selle sõlmimise kuupäevast tähtajatuna. Kliendilepingut on võimalik üles öelda või lõpetada Üldtingimuste kohaselt.

[Laadin alla PDF-faili](#)

Olen tutvunud ja nõustun Kliendilepingu ja Üldtingimustega

[< Tagasi](#)  [>](#)

Joonis 18. Kliendikonto loomise 2. samm

## Lisa 3 – Uus ühesammuline kliendikonto loomine

Koduleht Iseteenindus E-pood Русский TEST ISIK2 Väljun iseteenindusest

**diil** Teenused Arved Konto

### KLIENDIKONTO LOOMINE

EMT, Elioni ja Eesti Telekom'i ühinemisel sai meie uueks ärinimeks Telia Eesti AS. Selleks, et saaksid meie teenuseid ka edaspidi mugavalt kasutada, palume registreerida kasutajakonto ja sõlmida uue kasutajalepingu.

Ees- ja perekonnanimi Isikukood  
TEST ISIK2 39001010044

**Sisesta oma kontaktandmed**

Riik

Address  Korter

Telefon

E-post


Suhtluskeel

Kinnitan andmete õigsust

---

#### Kliendileping ja tahteavalduse sisu

- Kliendileping on kliendisuhete raamleping, mis loob Kliendile võimaluse lihtsal viisil liituda Telia Teenustega, kasutada e-Keskkondi, osta kaupu ja sõlmida muid Lepinguid Teliaga.
- Telia osutab Kliendile Teenuseid ja müüb kaupu tingimustel, mis on toodud Üldtingimustes, Hinnakirjas, Teenuse kohta kehtivates Teenusetingimustes ja muudes Tingimustes. Telia viitab Tingimustele Kliendiga Lepingu sõlmimisel või talle Teenuse, kauba või e-Keskonna pakumisel ja võimaldab Kliendil Tingimustega tutvuda. Tingimused on Lepingu osaks ja Pooled kohustuvad neid järgima.
- Kliendilepingu sõlmimisega Klient kinnitab, et ta on Üldtingimuste ja Andmete kasutamise põhimõtetele tutvunud ning kohustub neid täitma.
- Teenuste ja ostetud kaupade eest peab Klient maksma Teliale tasu vastavalt esitatud arvele sellel märgitud maksetähtajaks.
- Kliendileping kehtib alates selle sõlmimise kuupäevast tähtajatuna. Kliendilepingut on võimalik üles öelda või lõpetada Üldtingimuste kohaselt.

 [Kliendileping ja tahteavaldus](#)

Olen tutvunud ja nõustun **Telia üldtingimuste** ja kliendilepinguga.

**KINNITAN JA JÄTKAN TELLIMIST >**

Joonis 19. Ühesammuline kliendikonto loomine Diili iseteeninduses.

## Lisa 4 – Uus kliendikonto loomine koos andmetega

Koduleht Iseteenindus E-pood Русский TEST ISIK2 Väljun iseteenindusest

**diil** Teenused Arved Konto

### KLIENDIKONTO LOOMINE

EMT, Elioni ja Eesti Telekomis ühinemisel sai meie uueks ärinimeks Telia Eesti AS. Selleks, et saaksid meie teenuseid ka edaspidi mugavalt kasutada, palume registreerida kasutajakonto ja sõlmida uue kasutajalepingu.

Ees- ja perekonnanimi: TEST ISIK2 Isikukood: 39001010044

**Sisesta oma kontaktandmed**

Riik: Eesti

Address: Tormi tn 13, Pärnu linn, 80023 Pärnu linn, Pärn Korteri:

Telefon: 5543587

E-post: testisik2@gmail.com


Suhtluskeel: **Eesti** Inglise Vene

Kinnitan andmete õigsust

---

#### Kliendileping ja tahteavalduse sisu

- Kliendileping on kliendisuhte raamleping, mis loob Kliendile võimaluse lihtsal viisil liituda Telia Teenustega, kasutada e-Keskkondi, osta kaupu ja sõlmida muid Lepinguid Teliaga.
- Telia osutab Kliendile Teenuseid ja müüb kaupu tingimustel, mis on toodud Üldtingimustes, Hinnakirjas, Teenuse kohta kehtivates Teenusetingimustes ja muudes Tingimustes. Telia viitab Tingimustele Kliendiga Lepingu sõlmimisel või talle Teenuse, kauba või e-Keskkonna pakkumisel ja võimaldab Kliendil Tingimustega tutvuda. Tingimused on Lepingu osaks ja Pooled kohustuvad neid järgima.
- Kliendilepingu sõlmimisega Klient kinnitab, et ta on Üldtingimuste ja Andmete kasutamise põhimõtetele tutvunud ning kohustub neid täitma.
- Teenuste ja ostetud kaupade eest peab Klient maksma Teliale tasu vastavalt esitatud arvele sellel märgitud maksetähtajaks.
- Kliendileping kehtib alates selle sõlmimise kuupäevast tähtajatuna. Kliendilepingut on võimalik üles öelda või lõpetada Üldtingimuste kohaselt.

 [Kliendileping ja tahteavaldus](#)

Olen tutvunud ja nõustun [Telia üldtingimuste](#) ja [kliendilepinguga](#).

**KINNITAN JA JÄTKAN TELLIMIST >**

Joonis 20. Uus kliendikonto loomine näiteandmetega.



## Lisa 5 – Küsitlus

### Valdkondadepõhised mikroveebid

#### Valdkondadepõhised mikroveebid

Viimasel ajal on meie peamine ettevõtteülene eesmärk iseteeninduse ja teenindusveebi funktsionaalsuste liigutamine Liferay veebiplatvormilt valdkonnapõhisesse mikroveebidesse. Selle küsitluse eesmärk on aru saada, mil moel on mikroveebide ja ka mikroteenuste arhitektuuri stiili implementeerimine mõjutanud arendajate igapäevast tööd ning millist kasu/kahju antud juurutus on Teliale kui organisatsioonile kaasa toonud.

Vastamiseks läheb aega maksimaalselt 3 minutit.

\* 1. Ma tean, mis on mikroveebid ja kuidas see arhitektuuri stiil puudutab Teliat

Ei tea midagi	Pigem ei tea	Ei oska öelda	Pigem tean	Tean
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

\* 2. Hinda järgmiste Liferay arendusi puudutavate väidete tõesust 5 palli skaalal.

Liferay puhul:

	Ei ole üldse nõus	Pigem ei nõustu	Ei oska öelda	Pigem nõustun	Olen täiesti nõus
võtab rakenduse lokaalne püstipaneik oluliselt kauem aega võrreldes mikroveebidega	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
on arendusprotsess kiire ja lihtne	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
on testimise maht suurem võrreldes mikroveebidega	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
on vea tekkimisel tootmiskeskonnas ( <i>live</i> ) selle paranduse relüisimine kiire	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
on erakorralise relüisi korraldamine lihtne ja valutatu protsess	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
on võimalik funktsionaalsused kiiresti kliendini viia	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
on <i>portlet</i> -ide arendus mugav ja lihtne	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
ei nõua arendamine minult lisateadmisi (st pole spetsiifilisi oskusi vaja)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
on mul arendajana rohkem stressi, kuna vea tekkimisel on selle paranduse relüisimine keerukas ja aeganõudev	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
on võimalik kiiresti tuvastada, milles viga, kui tootmiskeskonnas ( <i>live</i> ) midagi ei toimi	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

on kommunikatsioonivajadus erinevate valdkondade vahel suurem, kui mikroveebide puhul

Next

\* 3. Hinda järgmiste väidete tõesust 5 palli skaalal.

Tunnen, et seoses mikroveebide (*micro services to the frontend world*) implementeerimisega:

	Ei ole üldse nõus	Pigem ei nõustu	Ei oska öelda	Pigem nõustun	Olen täiesti nõus
lisavad uued arendajad kiiremini väärtust	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
on rakendused rohkem kapseldatud ja loogikat on palju vähem	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
on probleemide (bug) lahendamine ja relüüsimine kiirem	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
on esile kerkinud suurem <i>full-stack</i> arendajate vajadus	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
on mul arendajana võimalik panustada <i>end to end</i>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
on võimalik suhteliselt sõltumatuna arendada ja relüüsida	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
saavad arendused kiiremini valmis	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
on võimalik funktsionaalsused kiiremini kliendini viia	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
on tagatud parem kvaliteet, sest vähem on testida	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
on mul arendajana rohkem autonoomiat, vabadust ja vastutust	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
on mul arendajana vähem stressi, kuna alati on võimalik eelnev versioon kiiresti tootmiskeskonda paigaldada	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
olen arendajana produktiivsem	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
on võimalik kiiremini tuvastada, milles on viga, kui tootmiskeskonnas ( <i>live</i> ) midagi ei toimi	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
on mul arendajana võimalik rohkem kaasa rääkida võimalike lahenduste osas	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

\* 4. Mis on Sinu arvates mikroveebide suurim miinus?

\* 5. Mis on Sinu arvates mikroveebide suurim pluss (kasu)?

6. Kui Sul on muid mõtteid (mured/rõõmud) seoses Telia valdkonnapõhiste mikroveebidega, siis kirjuta sellest siia..

Prev

Done

Joonis 21. Küsitlus