



Wisercat

# Projekti „Ettevõtjate ühtse kontaktpunkti infotehnoloogilise arhitektuuri analüüs” lõpparuanne



Euroopa Liit  
Euroopa  
Regionaalarengu Fond



Eesti  
tuleviku heaks

23.07.2021

## Sisukord

1	Kokkuvõtte.....	6
2	Lühendid ja mõisted.....	8
3	Hetkeolukord.....	12
3.1	Riigiportaali arhitektuur AS-IS .....	12
3.2	Riigiportaali AS-ISi arhitektuuri ülevaade.....	12
3.3	Turvalisus .....	15
3.3.1	Autentimine.....	15
3.3.2	Autoriseerimine.....	15
3.3.3	Parandusettepanekud.....	16
3.4	Andmete haldus .....	16
3.4.1	Parandusettepanekud.....	17
3.5	Suhtlus komponentide vahel .....	17
3.5.1	Ruuter + andmemuundur + TIM + XTR .....	17
3.5.2	Parandusettepanekud.....	18
3.6	Kasutajaliidesed .....	21
3.6.1	Menüüd .....	22
3.6.2	Artiklid ja portaali tekstiline sisu.....	23
3.7	Artiklivaramu SEO.....	23
3.7.1	Parandusettepanekud.....	24
3.8	Logimisvajaduse analüüs ja logimislahenduse kirjeldus (Süsteemide jälgitavus).....	24
3.8.1	Logide kirjutamine ja koondamine .....	24
3.8.2	Staatuslehed .....	25
3.8.3	Trace ID-d.....	25
3.8.4	Parandusettepanekud.....	25
4	Olemasolevad protsessid.....	25
4.1	Riigiteenuste haldus .....	25
4.1.1	PH.01. Artiklivaramu kirje (artikli) toimetamine .....	26
4.1.2	PH.02. Eesti.ee portaali struktuuri täiendamine.....	26
4.1.3	PH.03. Eesti.ee portaali sisu administreerimine .....	27
4.1.4	PH.04. Avaliku teenuse kirjeldamine .....	28

4.1.5	PH.05. Eesti.ee portaali teenuste arendus/administreerimine.....	28
4.2	Riigiteenuste kasutaja protsessid.....	28
4.2.1	Ettevõtja protsessid.....	29
4.2.2	Eraisikute protsessid.....	35
5	Erasektori, avaliku sektori äriühingute ja sihtasutustest teenusepakkujate analüüs.....	36
5.1	Erasektori teenusepakkujate kirjeldus.....	37
5.2	Teenuste kasutamise võimalused sündmusteenuste pakkumisel.....	38
5.3	Liidestamise võimaluste kasutamine.....	39
5.4	Liidestamise protsessi variandid.....	40
5.4.1	Liidestamise näidised.....	40
5.5	Riskid.....	41
6	Optimeerimise ettepanekud.....	42
6.1	Optimeerimist vajavad kohad.....	42
6.2	Optimeerimise põhimõtted.....	43
6.3	Protsesside optimeerimise ettepanekud.....	44
6.4	Vajalikud muudatused ja seotud väljakutsed.....	45
6.4.1	Teenuste kataloogi muudatusvajadused.....	46
6.4.2	Artiklivaramu muudatusvajadused.....	47
6.4.3	Tehnilised optimeerimised.....	47
7	Kavandatav lahendus (TO BE).....	49
7.1	Kavandatavad protsessid.....	49
7.1.1	TOBE.01. Artiklivaramu kirje toimetamine.....	49
7.1.2	TOBE.02 Eesti.ee portaali struktuuri täiendamine.....	49
7.1.3	TOBE.03. Sündmusteenuse kirjeldamine.....	50
7.1.4	TOBE.03.1 Sündmusteenuse arendus.....	50
7.1.5	TOBE.03.2 Teenuse kasutamine lingina.....	51
7.1.6	TOBE.03.3 Eesti.ee portaali sisu administreerimine.....	52
7.1.7	TOBE.04 Avaliku teenuse kirjeldamine.....	52
7.1.8	TOBE.05. Sündmusteenuse planeerimine.....	52
7.2	Andmemudel ja andmevood.....	53
7.2.1	Loogiline andmemudel.....	53
7.2.2	Andmevood.....	57
7.3	Visiooni dokumendi kasutusjuhtumid.....	58
7.3.1	Ettevõtte kohustused, teenused ja toetused / Info ettevõtjale.....	58

7.3.2	Ettevõtte sündmuskalender .....	58
7.3.3	Ettevõtte riiklik postkast .....	58
7.3.4	Ettevõtte andmekaart .....	59
7.3.5	Andmetele juurdepääsu andmine / volitused, rollid ja pääsuõigused .....	59
7.3.6	Reaalajas andmete esitamine .....	59
7.4	Vastutusmudel .....	59
7.4.1	Osapooled .....	59
7.4.2	Tegevused .....	60
7.4.3	Vastutusmaatriks .....	62
7.5	Tehnilised nõuded .....	63
7.6	Tulevikulahenduse arhitektuuri kirjeldus .....	64
7.6.1	Töökindlus .....	64
7.6.2	Skaleeritavus .....	65
7.6.3	Hooldatavus .....	66
7.6.4	Turvalisus .....	67
7.6.5	Komponentdiagramm .....	68
7.6.6	Muutmist/arendamist vajavad komponendid .....	68
7.6.7	Küsimärgiga komponendid .....	69
7.6.8	Printsiibid .....	69
7.6.9	Skaleeritavus, tõrkekindlus .....	70
7.6.10	Paigaldus .....	71
7.6.11	Ressursside planeerimine .....	72
7.7	Pilveteenused .....	73
7.7.1	Võrdlustabel .....	74
7.8	Innovatsiooni stimuleerimine .....	75
8	Pääsuõigused .....	75
8.1	Visioon .....	75
8.2	Lahendus .....	76
8.2.1	Lahendus A .....	76
8.2.2	Lahendus B .....	77
8.2.3	Kokkuvõtte .....	79
9	Arhitektuurilahenduste võrdlus .....	80
9.1	Mikroarhitektuuri teemad .....	81
9.1.1	Turvalisus .....	81

9.1.2	Andmete haldus .....	82
9.1.3	Suhtlus komponentide vahel.....	82
9.1.4	Service discovery .....	84
9.2	Automaattestide kasutus.....	84
9.3	Süsteemide jälgitavus .....	85
9.4	Kasutajaliidesed .....	85
9.4.1	Serveripoolne UI kokkupanek .....	85
9.4.2	Kliendipoolne UI kokkupanek.....	86
9.5	Rakenduse paigaldamise variandid .....	87
9.5.1	Eraldiseisvad virtuaalsed masinad (VM).....	87
9.5.2	Konteinertehnoloogia.....	88
9.5.3	Konteinerite orkestreerimine .....	88
9.5.4	Pilveteenused .....	88
9.6	Seadistatavuse ja konfigureeritavuse analüüs .....	89
10	Liidestamise variantide analüüs.....	90
10.1	Lihne link .....	90
10.1.1	Lahenduse kirjeldus .....	90
10.1.2	Arendusvajadused .....	91
10.1.3	Halduslahendus .....	91
10.2	Süvalink.....	91
10.2.1	Lahenduse kirjeldus .....	91
10.2.2	Arendusvajadused .....	92
10.2.3	Halduslahendus .....	92
10.3	Vormihaldusvahendiga kirjeldatud vormi kasutamine .....	92
10.3.1	Lahenduse kirjeldus .....	92
10.3.2	Arendusvajadused .....	93
10.3.3	Halduslahendus .....	94
10.4	Domeeni jagamine - asutuse lehe integreerimine riigiportaali.....	94
10.4.1	Lahenduse kirjeldus .....	94
10.4.2	Arendusvajadused .....	96
10.4.3	Halduslahendus .....	96
10.5	Microfrontend lahendused.....	96
10.5.1	Lahenduse kirjeldus .....	96
10.5.2	Arendusvajadused .....	97

10.5.3	Halduslahendus .....	98
11	Masinõppe kasutamise analüüs .....	98
11.1	Tehisintellekti määratlus .....	98
11.1.1	Tehisintellekti lahenduse tunnused .....	99
11.2	Masinõppe kasutamise võimalused .....	99
11.2.1	Masinõppe/tehisintellekti kasutamise ülevaade .....	100
11.3	Tehnoloogilised aspektid .....	102
11.3.1	Andmete kasutamine .....	102
11.3.2	Masinõppe juurutamine .....	103
12	Kontaktpunkti ja sündmusteenuste arhitektuuri projektide vastutuse jaotus .....	104
13	Riskide analüüs ja maandamise meetodid .....	106
13.1	Äririskid .....	106
13.2	Tehnoloogilised riskid .....	110
14	Arendusplaan ja prioriteedid .....	112
14.1	Täiendused liidestatud süsteemides .....	117
15	Halduskulude hinnang .....	117
15.1	Riigipilve kasutamise maksumus .....	117

# 1 Kokkuvõtte

Ettevõtjatele suunatud avaliku sektori teenused on killustunud mitme sarnase eesmärgiga platvormide ja lahenduste vahel, puuduvad ühtsed alused ja komponendid teenuste pakkumiseks ühest kohast. Nii peabki inimene ettevõtte asutamisel ja juhtimisel orienteeruma mitme infotehnoloogiliselt erineva keskkonna vahel.

Käesolev arhitektuurianalüüs on osa ettevõtjatele ühtse kontaktpunkti loomise projektist.

Projekti eesmärk on leida riigiportaalis realiseerimiseks jätkusuutlikud ning skaleeritavad tehnoloogilised lahendused, nõuded veebipõhise platvormi IT-arhitektuurile, valida välja nende realiseerimiseks sobivad tehnoloogiad. Projekti raames on valminud uue ettevõtjate ühtse kontaktpunkti arhitektuurianalüüs ja liidestamise prototüüp, vastavalt tehnilises kirjelduses esitatud nõuetele.

Projekti raames on uuritud ja proovitud erinevaid tehnilisi võimalusi uute, tehnoloogiliselt innovaatiliste ja asjakohaste tööriistade kasutuselevõtmiseks ja nende rakendamiseks sündmusteenuste puhul.

## **Hetkeolukorra ülevaade**

Portaali sisu on hallatud CMS tarkvara abil – tänasel päeval on valitud Grav CMS, portaali sisu hoitakse Git-is. Andmevahetust tagab komponent Ruuter, pääsuõiguste kontrollimisega tegeleb TIM. Infoartikleid hallatakse riigiportaali CMS süsteemis ja riigiportaali halduri poolt, kuid nende sisu tuleb teenusepakkujalt.

Puuduvad paindlikud tööriistad teistes süsteemides paiknevate teenuste kasutamiseks riigiportaali osana. On olemas vormigeneraatori lahendus Orbeon Forms platvormil, kuid see katab ainult osa integreerimisvajadustest ja vajab edasiarendamist kasutajakogemuse parendamiseks.

## **Prototüübi funktsionaalsus**

Võrreldes "Ettevõtja jaoks ühtse veebipõhise kontaktpunkti visioon" projekti tulemustega on prototüübimise käigus tehtud rida muudatusi ja täiendusi.

Täislahendus riigiportaalis ei ole otseselt realiseeritud, selle asemel on arendatud *microfrontend* - moodulite integreerimise lahendus. Lahenduse juures ei ole oluline kes on integreeritud *microfrontend* mooduli ega komponendi omanik, see võib olla teenuseosutaja või teenusepakkuja asutus, kuid võib olla ka riigiportaali haldur. Sel juhul võib lahendust nimetada täislahenduseks riigiportaalis.

Projekti käigus on arendatud järgmised liidestamise variandid:

**Lihtlink** – link teisele süsteemile, mis ei edasta süsteemide vahel kasutaja ega teenuse andmeid ja ei eelda lisaarendusi.

**Süvalink** – link otse vormile või teenuse lehele, lahendus võimaldab andmete edastamist ja tagasikutsumise funktsionaalsuse kasutamist. Parim tulemus saavutatakse SSO olemasolul ja teenusepakkuja süsteemi võimekusega kasutada edastatud tagasikutse aadressi.

**Vormilahendus** – formly.dev formaadi alusel tehtud vormigeneraatori lahendus. Võimaldab kokku panna lihtsamaid vorme, nii otse riigiportaali CMS süsteemis, kui ka kasutades olemasolevaid valmis konfiguratsioone.

**Microfrontend** mooduli integreerimine – teise süsteemi osa integreerimine. Projekti raames on tehtud kaks lahendust: teise süsteemi osa integreerimine täislehena ja teise süsteemi osa integreerimine portaali lehe osana. Prototüüpimise käigus on arendatud integratsioon angular teeki moodulite jaoks, kuid on võimalik integreerida ka teiste teekidega valmistatud komponente.

**Domeeni jagamine** teise asutusega – teenusepakkuja süsteemi integreerimine riigiportaali domeeni. Kasutajakogemuse säilimiseks peab teenusepakkuja kasutama riigiportaali poolt publitseeritud menüüd ja kasutama Veera raamistiku komponente. SSO lahenduse implementeerimine on samuti vajalik kasutajakogemuse parendamiseks.

Iga liidestamislahenduse jaoks on valminud liidestamisjuhend, mis kirjeldab riigiportaali seadistamist, vajalikke eeldusi ja arendusvajadusi riigiportaali ja liidestatud süsteemides.

Arendatud lahendusi on mõistlik edasi arendada, lisades nendele mugavuse ja kasutajakogemuse parendavaid muudatusi. Edasiarenduste nimekiri on antud prioritseeritud tööde nimekirjana.

## 2 Lühendid ja mõisted

Lühend / Mõiste	Kirjeldus
AAR	Autoriseerimisteenus
ABAC	Andmepõhine ligipääsuhaldus ( <i>Attribute-based Access Control</i> )
ACL	Pääsupiiramisloend ( <i>Access Control List</i> )
API	Rakendusliides ( <i>Application Programming Interface</i> )
CDN	Sisuedastusvõrk ( <i>Content Delivery Network</i> )
CMS	Sisuhaldussüsteem ( <i>Content Management System</i> )
COTS	Valmis tarkvaralahendus ( <i>Commercial off-the-shelf</i> )
CPSV-AP	<i>Core Public Service Vocabulary Application Profile</i> . Andmemudel, mis käsitleb põhilisi avalike asutuste poolt pakutavate teenuste tunnuseid ning mille eesmärk on standardiseerida sündmusteenuste semantikat. Sisuline laiem eesmärk on pakkuda kasutajakeskset informatsiooni teenuste kasutamise kohta vastavalt isikute elus aset leidvatele elu- või ärisündmustele.
CPU	kesktötlusseade ehk keskprotsessor ( <i>Central Processing Unit</i> )
CVE	Levinud haavatavused ja riskipositsioonid ( <i>Common Vulnerabilities and Exposures</i> )
DevOps	Tarkvaraarenduse kultuur, mille eesmärk on ühendada tarkvaraarendus ( <i>Dev</i> ) ja tarkvaraoperatsioonid ( <i>Ops</i> ).
Elusündmus	Füüsilise isiku elus toimuv sündmus
Erandteenus	Sündmusteenuse skooopi kuuluv osateenus, mida kaalutletult ei osutata sündmusteenuste osutamise käigus. Erandteenuse osutamist on võimalik siiski eraldiseisvalt algatada.
EÜKP	Ettevõtjate ühtne kontaktpunkt
Git	Hajutatud versioonihaldustarkvara
HTTP	Hüpertexti edastusprotokoll ( <i>Hypertext Transfer Protocol</i> ), protokoll teabe edastamiseks arvutivõrkudes.

Lühend / Mõiste	Kirjeldus
IAAS	Infrastruktuuri pakkuv teenus (pilveteenuse alamliik) <i>Infrastructure-as-a-Service</i>
ISKE	Eesti riigi infosüsteemide kolmeastmeline etalonurbe süsteem, mille eesmärk on tagada töödeldavatele andmetele piisava tasemega turvalisus.
JAR	Pakitud failiformaat, tavaliselt kasutatud Java klasside ja seotud ressursside kokkupakkimiseks nende jagamise lihtsustamiseks
JSON	Lihtsustatud andmevahetusvorming, mis põhineb JavaScripti programmeerimiskeele alamhulgal
JWT	Vaba standard JSON baasil (RFC 7519) ( <i>JSON Web Token</i> )
NPM	JavaScripti programmeerimiskeele jaoks loodud paketi haldur (Node Package Manager)
Metaandmed	Teenust või muu objekti kirjeldavate andmete kogum
MFN	Mittefunktsionaalsed nõuded
Microfrontend	<p><a href="https://micro-frontends.org/">https://micro-frontends.org/</a></p> <p><i>Microfrontend</i> on kasutajaliidese tükeldamise kontseptsioon (arhitektuuriline lähenemine, mitte konkreetne tehnika). <i>Microfrontend</i> vastab järgmistele kriteeriumitele:</p> <ul style="list-style-type: none"> <li>• <b>eraldi paigaldatav;</b></li> <li>• vearisk isoleeritud konkreetsele väiksemale kasutajaliidese alale;</li> <li>• piiratud skoobiga, mis on kergemini mõistetav;</li> <li>• väiksem koodibaas, mida on kergem arendada ja välja vahetada;</li> <li>• ei jaga oma seisu ja sõltuvust teiste komponentidega;</li> <li>• iseseisev, saab käsitleda eraldi rakendusena, ehk sisaldab kõike, mis on tööks vajalik.</li> </ul>
MTÜ	Mittetulundusühing
Mugavusteenus	Vabatahtlik mugavusteenus on põhiteenus, mille osutamine sündmusteenuse käigus toimub vaid teenuse saaja soovi korral.
OIDC	OpenID Connect, autentimise kiht OAuth2.0 protokollil peal.
Osateenus	Osateenus on üksik otsene avalik teenus, mille läbiviimine täidab elusündmusega seotud vajaduse. Osateenused jagunevad sündmusteenuse kontekstis põhiteenusteks, vabatahtlikeks mugavusteenusteks ja erandteenusteks.
Pattern	Muster, tarkvara arendamisel kasutatud tehnoloogiate või koodi kirjutamise mall või muster
PoC	Tehnilisi võimalusi demonstreeriv prototüüp ( <i>Proof of Concept</i> )

Lühend / Mõiste	Kirjeldus
PoD	Kubernetes-e väikseim paigaldatav ja hallatav jagatud ressursside ühik
Põhiteenus	Sündmusteenuse skooopi kuuluv osateenus, mida osutatakse alati sündmusteenuse osutamise käigus.
Proaktiivne teenus	Riigiportaali või muu teenuse osutaja või vahendaja poolt kasutaja andmete, käitumise või muu tegurite alusel suunatud osateenus.
RACI	Vastutusmaatriks, tabel, mille veergudele kirjutatakse projekti tegevused ja ridadele projektis osalejad.
RBAC	Rollipõhine ligipääsuhoodus ( <i>Role-based Access Control</i> )
Repositoorium	Andmebaas, kus säilitatakse hallatava projekti kõiki redaktsioone.
REST	Andmevahetuse arhitektuuri standard ( <i>REpresentational State Transfer</i> )
Riigiportaali haldur	Riigiportaali Eesti.ee toimimise ja sisu eest vastutav asutus
SA	Sihtasutus
SEO	Otsingumootoritele optimeerimine ( <i>search engine optimization</i> )
SLO	Ühekordne väljalogimine ( <i>Single Sign-Out</i> )
SSO	Ühekordne sisselogimine ( <i>Single Sign-On</i> )
SQL	Relatsioonilise andmebaasi päringute keel ( <i>Structured Query Language</i> )
Sündmusteenus	Otsene avalik teenus, mida mitu asutust osutab ühiselt, et isik saaks täita kõik kohustused ja kasutada kõiki õigusi, mis talle tekivad ühe sündmuse või olukorra tõttu. Sündmusteenus koondab mitu sama sündmusega seotud teenust (edaspidi osateenus) kasutajale üheks teenuseks.
Sündmusteenuse staatuspaneel	Sündmusteenuse osutamise ja hetkeseisu salvestamise süsteem
TARA	Riigi autentimisteenuse tarkvara tehniline nimi
Teenus	Käesoleva dokumendi raames on teenusena arvestatud otseseid teenuseid, mida juriidiline isik osutab füüsilisele või eraõiguslikule juriidilisele isikule.  Teenuseid võivad ettevõtjatele osutada nii avalikud asutused kui ka teised ettevõtted. Käesoleval dokumendil teenuse osutaja all peetakse silmas nii avalikke asutusi kui ka erasektori ettevõtteid.
Teenuse majutaja	Teenuse ligipääsu pakutava ressurssi haldur. Teenus ise võib asuda teises kohas ja olla teise avaliku asutuse või muu juriidilise isiku vastutusel. Majutaja vastutab ainult teenuse kättesaadavuse eest oma ressurssi kaudu.
Teenuste kataloog	Teenuste metainfo halduse süsteem, praegu selleks on riigiteenused.ee süsteem
Token	Objekt mis väljendab õigust teostada teatud tegevusi

Lühend / Mõiste	Kirjeldus
UI	Kasutajaliides ( <i>User Interface</i> )
URL	Veebiressurssi unikaalne aadress ( <i>Unified Resource Locator</i> )
vCPU	Virtuaalne keskprotsessor, kasutusel pilveteenuse süsteemides, sest klientidele pakutakse ekvivalentjõudlust
VPN	Virtuaalvõrk andmevahetuse privaatsuse tagamiseks ( <i>Virtual Private Network</i> )
WSDL	XML-vorming, mis kirjeldab veebiteenust ( <i>Web Service Description Language</i> )
Ärisündmus	Ettevõtte elu käigus toimuv sündmus
Ühe lehe rakendus	<i>Single-page application</i> , SPA. Veebirakendus või veebisait, mis suhtluse kasutajaga kirjutab käesoleva veebilehe dünaamiliselt üle, selle asemel et serverist tervet uut lehte alla laadida
Üksikteenus	vt Erandteenus
XML	W3C välja töötatud ja soovitatud standardne üldotstarbeline märgistuskeel ( <i>Extensible Markup Language</i> )
X-tee	Tehniline ja organisatsiooniline keskkond Eestis, mis võimaldab turvalist ja tõestusväärtust tagavat internetipõhist andmevahetust riigiasutuste vahel ja erasektoriga
YAML	Andmevormingu formaat

### 3 Hetkeolukord

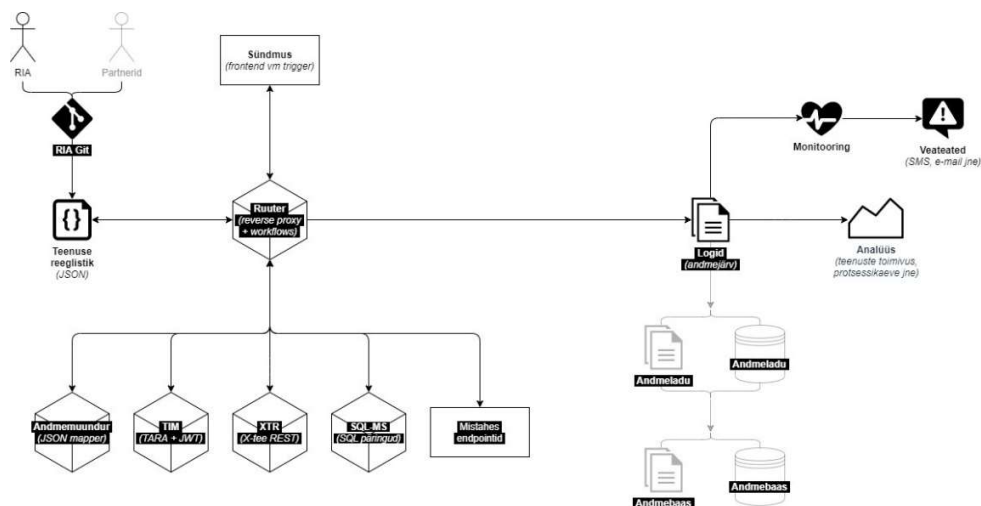
#### 3.1 Riigiportaali arhitektuur AS-IS

Ettevõtjatele ühtse kontaktpunkti (EÜKP) arhitektuuri koostamisel lähtume sellest, et see peab ühilduma uue (sinise) Eesti.ee portaali AS-ISi arhitektuuriga. Arhitektuurianalüüs sisaldab Eesti.ee portaali AS-ISi olukorda ja muudatusettepanekuid.

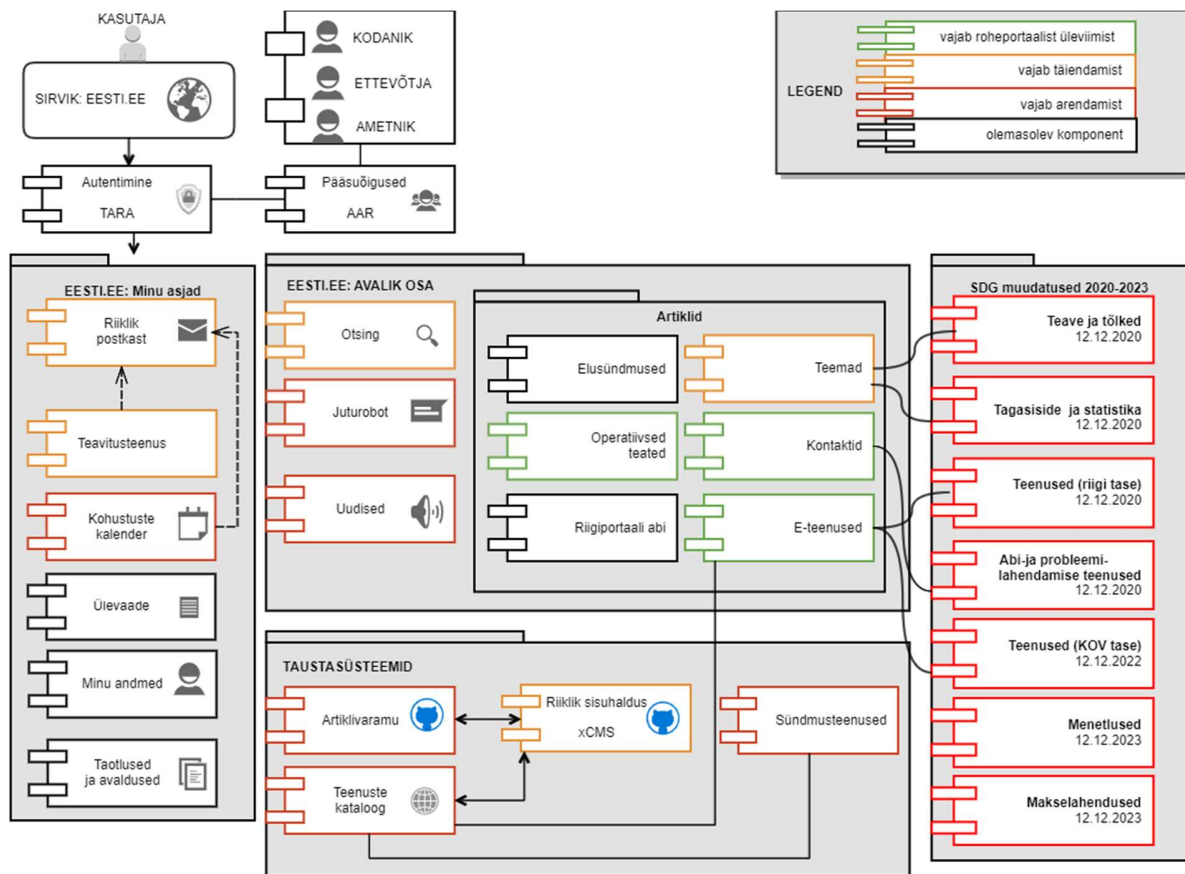
Sarnane arhitektuur riigiportaaliga lihtsustab süsteemi haldamist RIA poolt. Rohelise portaali arhitektuur on vananenud ja seda pole antud analüüsis käsitletud.

#### 3.2 Riigiportaali AS-ISi arhitektuuri ülevaade

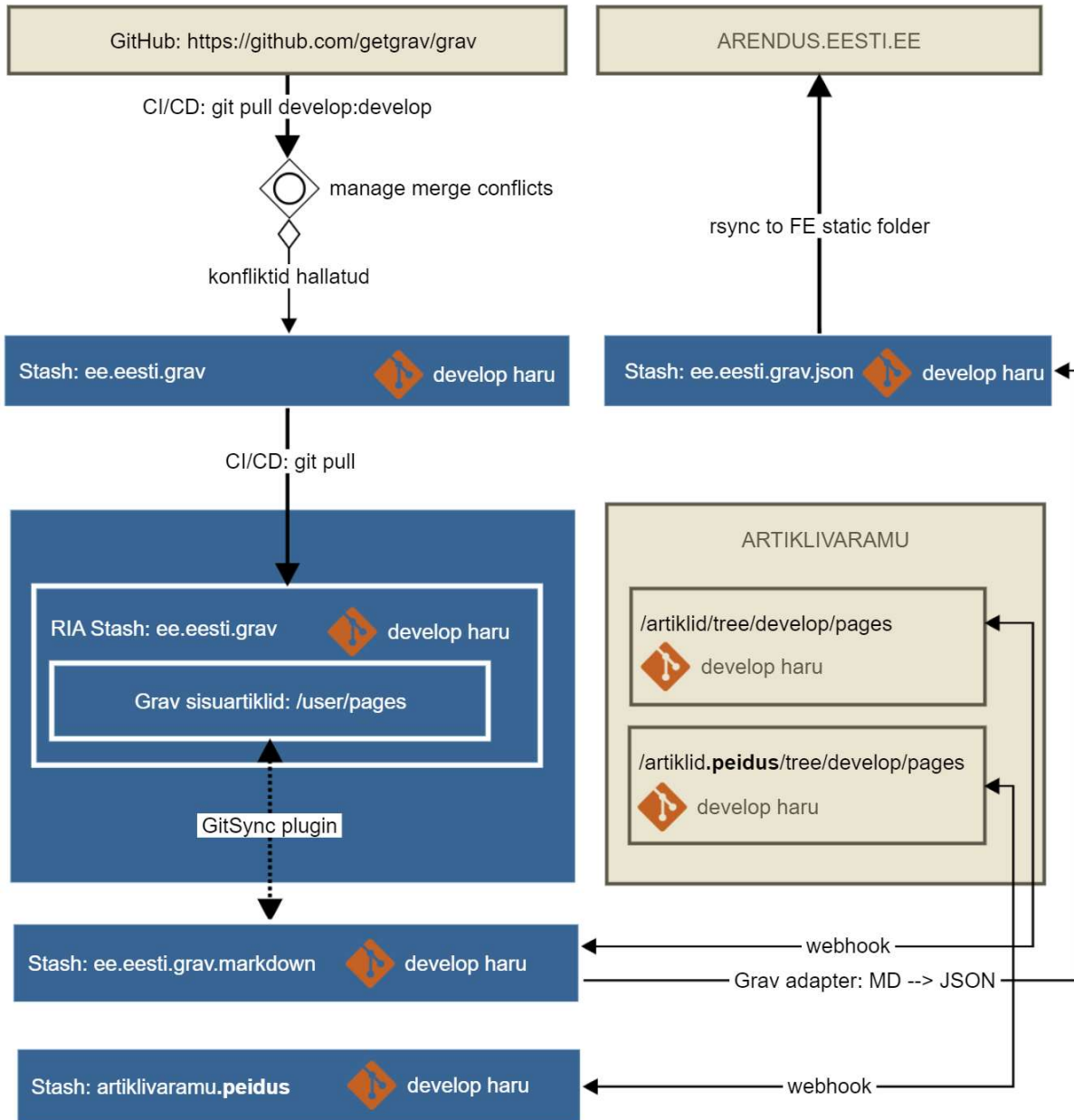
##### Põhikontseptsioon



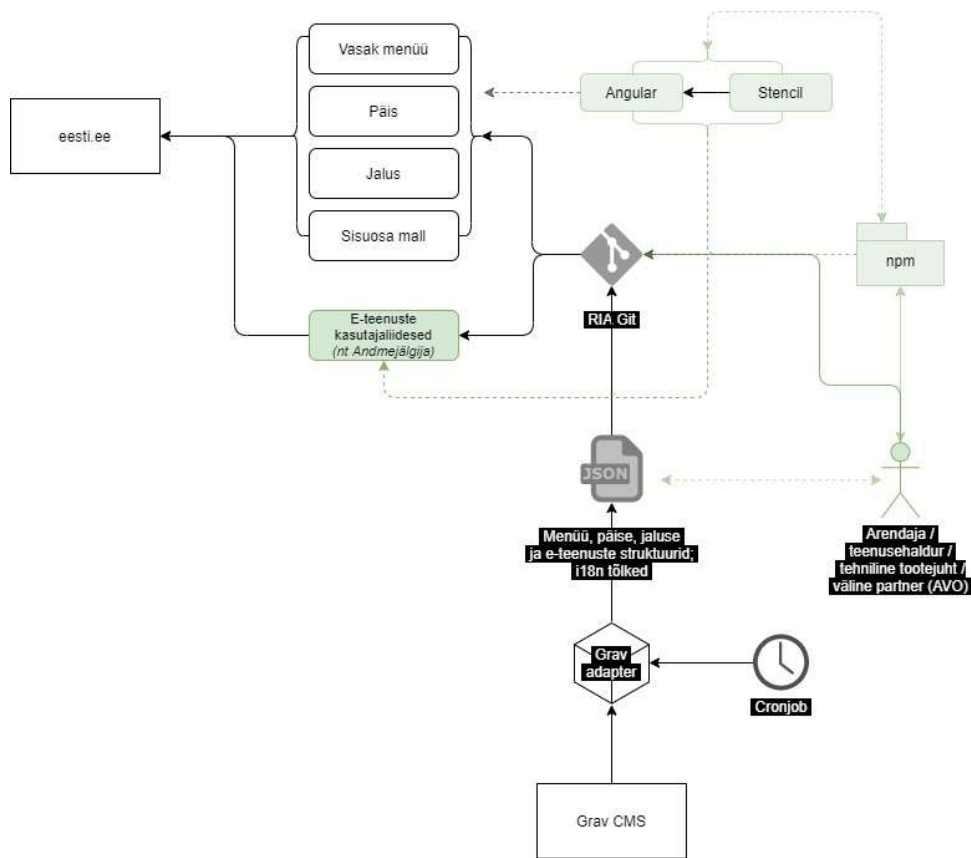
## Arhitektuuri detailssem vaade



## Artiklivaramu haldamise töövoog



## Kasutajaliidese ülesehitus



### 3.3 Turvalisus

#### 3.3.1 Autentimine

Eesti.ee portaalil kasutatakse autentimiseks riigi TARA autentimisteenust. Tehniliselt TARA toetab *OpenID Connect* protokoll. (<https://e-gov.github.io/TARA-Doku/>)

#### 3.3.2 Autoriseerimine

Vanas (Eesti.ee) riigiportaalil kasutatakse ettevõtte õiguste kontrollimiseks AARi teenust. Uues riigiportaalil ei ole see veel kasutuses (vähemalt ei ole toodangus).

### 3.3.3 Parandusettepanekud

Nagu oli mainitud hanke dokumentides, on vaja SSO (*Single-Sign-on*) lahendust. Koos SSO lahendusega ei tohi unustada SLO (*Single-Log-out*) vajadusi.

Üks variant oleks kasutusele võtta TARA SSO, mille projekti dokumentatsioon on siin (<https://e-gov.github.io/TARA-Doku/SSO+projektlahendus>)

TARA SSO kasuks räägib asjaolu, et see annab võimaluse kokku viia erinevate asutuste portaalid, nii et nende vahel saaks navigeerida.

Kui võtta mingi muu RIA-keskne SSO lahendus, siis asutustevahelist navigeerimist, mille puhul pole autentimist, tuleb vaadelda iga üksikjuhtumi kaupa.

Mis puutub autoriseerimist, siis RIAI võiks olla ligipääsuõiguste süsteem, millega saab käsitleda mitte ainult seaduslikke õigusi, vaid ka neid, mille puhul üks inimene on volitanud kedagi teist.

Hetkel kasutatav AAR-teenus on aegunud, ainukeseks kasutajaliideseks on vana roheline Eesti.ee portaal, mida ei ole plaanis edasi arendada.

Oleks mõistlik kasutada üht järgmistest variantidest:

- Kui esindamisõiguste kohta teha päring äriregistrist, siis sellise variandi puhul jääb lahendamata probleem volitustega.
- Uuendada AAR-teenus või luua täiesti uus pääsuõiguste süsteem, et see oleks kasutatav uues portaal.

## 3.4 Andmete haldus

Riigiportaal võtab andmeid mitmest allikast. Isikustatud informatsioon valitud isiku kohta tuleb X-tee kaudu teistest süsteemidest, päringute tegemise eest vastutab XTR-komponent.

Artiklivaramu tekstid tulevad sisuhaldusest (Grav CMSist), need tekstid on oma olemuselt staatilised ja neid hoitakse GITis versioonihalduses.

Samuti GIT-ist tulevad kõik võimalikud konfiguratsioonid.

Arhitektuur, mis põhineb enamjaolt kas teiste asutuste teenustel (X-tee) või staatilisel sisul.

### 3.4.1 Parandusettepanekud

Kui tekib vajadus kasutada SQL või no-SQL andmebaase, siis peab jälgima printsiipi „üks teenus = üks baas“, arvestades valitud tehnoloogia nõudeid.

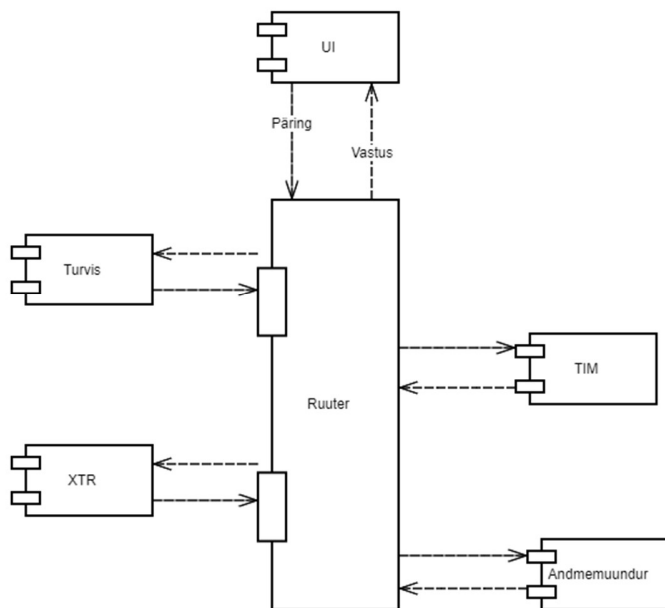
Nii nagu artiklivaramu, võiksid ka portaali sisuartiklid ja tekstid tulla Grav CMSist, nagu staatiline sisu. See lihtsustaks portaali haldamist, vajadusel toimetajad saaksid muuta tekste ilma lisa arenduseta.

## 3.5 Suhtlus komponentide vahel

### 3.5.1 Ruuter + andmemuundur + TIM + XTR

Kõiki päringuid teeb UI (kasutajaliides) **ruuteri** kaudu, mis võtab päringu vastu, töötleb vastavalt teenuse konfiguratsioonile ja annab vastuse tagasi.

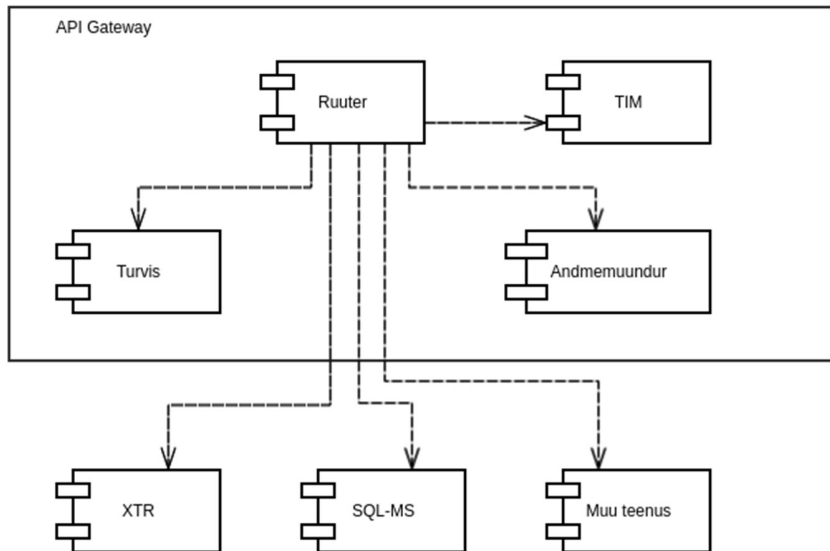
Ruuter on kasutajaliidese jaoks n-ö sisenemispunkt (*entrypoint*).



### 3.5.2 Parandusettepanekud

- Kaasajastada ja ühtlustada terminoloogiat, et see oleks maailmas levinud kasutusega ühtemoodi. See annab võimaluse arendajatega paremini suhelda ning leida ühist keelt. Näiteks,

- Ruuter, andmemuundur, TIM – võib võtta kasutusele koondnimetust (*API Gateway*). Iga komponent *API Gateway*'s täidab küll oma rolli, kuid arendaja jaoks see peab olema läbipaistev.
- Määrata *API Gateway* jaoks funktsioonid, näiteks
  - teenuste ühise turvalisuse tagamine ehk päringute autentimine, sessioonide hoidmine (TIM);
  - päringute logimine (ruuter);
  - päringute koondamine (ruuter);
  - päringute formaatide teisendamine (andmemuundur);
  - *API Gateway pattern*'i kohta vt: (<https://microservices.io/patterns/apigateway.html>)



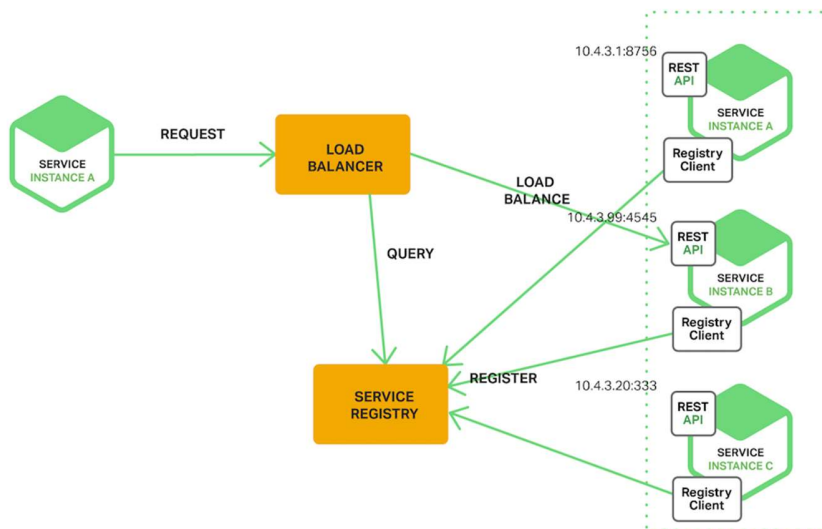
- Hetkel on **/userinfo endpoint** kasutajale kättesaadav läbi TIMi domeeni. Kui me lähtume *API Gateway pattern*’ist, siis antud *endpoint* peab olema ka publitseeritud läbi **ruuteri**.
- API Gateways ehk ruuteris peab olema võimalik seadistada erinevaid autentimise meetodeid või komponente, olgu see siis TIM või muu autentimisteenuse pakkuja (nt Apareocas või otse TARA SSO). Näiteks OIDC protokollis kasutatakse tavaliselt **/introspect endpointi**, mis on mõeldud *access token*’i valideerimiseks. Paljudes kohtades kasutatakse **/userinfo endpointi** valideerimiseks, kuid spetsifikatsiooni järgi need peavad tagastama erinevat informatsiooni. *Userinfo* tagastab informatsiooni kasutaja kohta, *Introspect* tagastab informatsiooni, mis on seotud konkreetse *token*’iga.
- Teenusteregistri kasutuselevõtt (*Service discovery*). Annab ülevaate komponentidest, mis on antud keskkonda paigaldatud ja kus (mis URLi või nime taga) need asuvad. Lihtsustab infrastruktuuri haldamist, näiteks automaatne koormusjaoturi konfigureerimine. Ühendused komponentide vahel peavad toimima komponendi nime järgi (mis peab olema

sama kõikides keskkondades), mitte URL-i või IP järgi, mis võivad erinevates keskkondades olla erinevad. See lihtsustab komponentide/rakenduste haldamist. Enamasti kasutatakse kaht põhilist printsiipi: serveripoolne tuvastus ja kliendipoolne tuvastus. Keskkondades, kus suhtlus komponentide vahel peab olema keskselt kontrollitav, kasutatakse nn serveripoolset tuvastust (*server-side service discovery*).

- Ettepanek kasutada standardiseeritud autentimise/autoriseerimise protokolle ja nende realisatsioone, näiteks OIDC. Hetkel on näiteks kasutuses n-ö isetehtud protokoll. Selline lähenemine lihtsustaks arendust ja haldust.

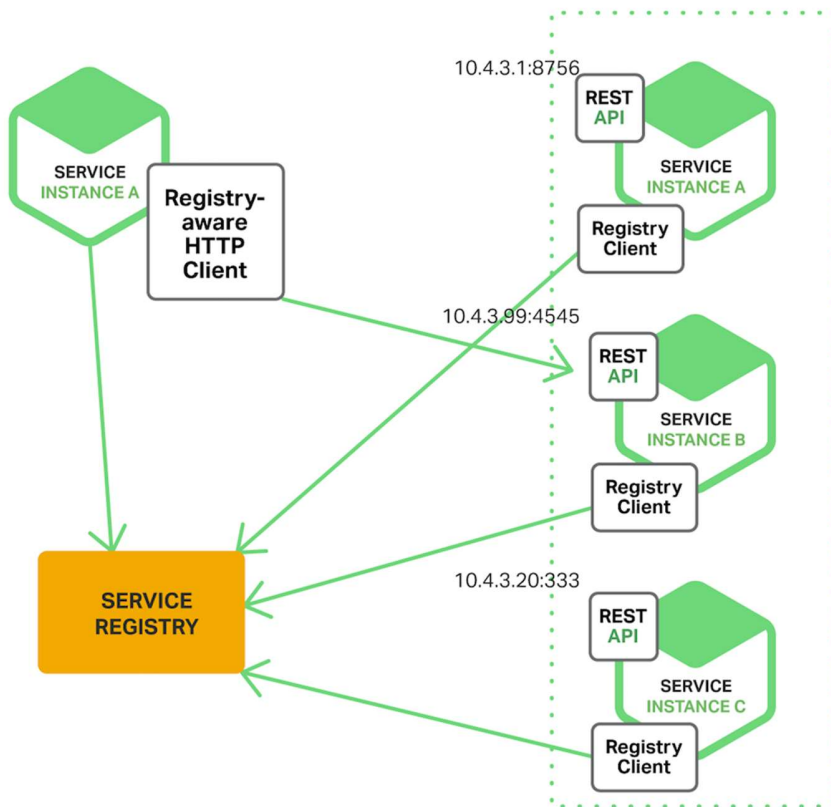
### 3.5.2.1 Server-Side Service Discovery

Arvestades olemasolevat arhitektuuri, kõige mõistlikumaks variandiks jääb serveripoolne tuvastuse (*server-side service discovery*) variant. Kuna komponentide vaheliseks suhtlemiseks näeme ette *reverse proxy* kasutust, siis komponent ise ei saa pöörduda otse teenuse poole.



### 3.5.2.2 Client-Side Service Discovery

Antud variant on toodud võrdluseks eelmise variandiga.



### 3.6 Kasutajaliidesed

Riigiportaali arhitektuuri ja esitluskihi visioonis on kasutatud *microfrontend*'i mõistet. Kindlasti on mõistlik arendada ja taaskasutada kasutajaliidese ühiseid komponente, nagu menüüd, jalust ja päist, kuid tuleb vaadata erinevaid aspekte.

Esiteks, ühiste komponentide arendamiseks on vähemalt kaks varianti:

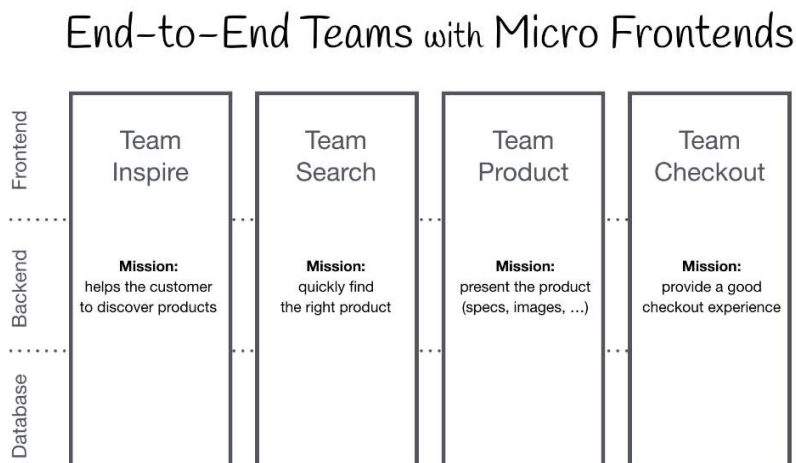
- Arendamiseks kasutada populaarsemaid JavaScript raamistikke, näiteks Angulari, Reacti, Vue'd ehk menüüd Angularis ja menüüd Reactis.
- **Eelistatuim variant:** Arendamiseks kasutada universaalset vahendit, näiteks StencilJS ehk üks universaalne menüü, mida saab kasutada nii Angularis kui ka Reactis.

Esimese variandi puhul rakendatakse sama funktsionaalsust mitu korda, mis on ajaliselt ja rahaliselt kallim, kuid enamasti arendajad eelistavad seda varianti rohkem.

Teise variandi puhul luuakse funktsionaalsust ühe korra, kuid komponendid peavad olema tehtud ja testitud arvestades kõiki kasutuses olevaid raamistikke. Organisatoorselt on selline lähenemine keerulisem.

### 3.6.1 Menüüd

Kui portaal koosneb mitmest komponendist, kus funktsionaalsus on vertikaalselt jagatud, siis menüü mängib süsteemis olulist rolli. Menüü peab olema ühine ja peab kõik komponendid omavahel kokku viima.



AS-ISi olukorras menüü muutmine on arendus, kus arendaja muudab konfiguratsiooni GITis ja see jõustub koos paigaldamisega. Oli idee avaldada menüüd (sisu osa, JSONi kujul) CDNi (*Content Delivery Network*) kaudu teistele osapooltele staatilise failina. Kood, mis ehitab menüüd andmete pealt, publitseeritakse avalikult NPM pakettidena.

Nende artiklite jaoks, mis tulevad GRAV-ist, menüü koostatakse *grav adapteri* poolt, kui toimub transformatsioon Markdown formaadist JSON formaati. See puudutab ainult 2. taseme menüüsid.

Korduvalt on käinud läbi ettepanek hallata kõiki eesti.ee menüüsid GRAV CMS-is, et need oleksid ühes kohas.

#### 3.6.1.1 Paranduste ettepanek

- Tekitada võimalus 1 taseme menüü haldamiseks.
- Võtta kasutusele dünaamilised menüüd, mis oskavad aadressi (PATH-i) ehitada sõltuvalt kontekstist.

#### 3.6.2 Artiklid ja portaali tekstiline sisu

Artikleid koostatakse sisuhalduse lahenduses, mille nimeks on Grav CMS. Grav CMS töötab Markdowni peal ja tulemuseks on Markdowni failid. Edasi neid faile konverteeritakse JSONi failideks, kasutades Grav Adapteri komponente.

Artikleid hoitakse GITis ja kasutaja saab need kätte staatiliste failidena. Artikleid haldavad RIA toimetajad käsitsi.

#### 3.6.2.1 Paranduste ettepanek

GRAV-i pääsuõiguste süsteem peab toetama ACL-e, et saaks õigusi määrata üksikute artiklite kaupa. Siis oleks võimalik lubada artiklite omanikele ise hallata tekste ja kirjeldusi.

### 3.7 Artiklivaramu SEO

Hetkel on eesti.ee portaali tekstid keeruliselt leitavad otsingumootoriga. Näiteks, <https://www.eesti.ee/et/koroonainfo/abi-ja-info-koroonaviirusega-seotud-kuesimustes/>

Ainukene info, mida otsingumootorid antud lehelt leiavad, on metainfo *description* 'i välja peal ja see on:

<meta data-n-head="1" name="description" content="Sisukord

Arstiabi  
Piiriületus, reisimine, konsulaarabi  
Sõidueksam, juhiluba  
Ajutised töökohad, abipakkujad

Kõige ajakohasemat ametlikult kinnitatud infot ja j">

Olukorda raskendab ka see, et lehed on tehtud *Single-Page* rakendusena, see tähendab, et tuleb rakendada eraldi meetmeid, et leht oleks indekseeritav otsingumootorite poolt.

Artiklivaramu hallatud infotekstide kasutamisega peab ka metainfo olema hallatud tekstiga koos ja edastatud riigiportaali. Riigiportaali peab oskama artiklivaramust edastatud metainfot kasutada.

### 3.7.1 Parandusettepanekud

- Peab järgima Google'i ja teiste mootorite juhiseid.
- *Single-Page* lehed peavad sisaldama korrektset metainfo välja.
- Võib kasutada *server-side rendering* tehnoloogiat. **Tänase info järgi seda varianti juba testitakse.**

## 3.8 Logimisvajaduse analüüs ja logimislahenduse kirjeldus (Süsteemide jälgitavus)

### 3.8.1 Logide kirjutamine ja koondamine

Iga komponendi logid transporditakse rsyslog abil RIA kesksesse logihaldusesse, kust igal projektil on võimalik neid andmeid tekstifailide kujul lugeda ja endale sobival moel tarbida.



Logid asuvad keskses kohas, muutmata kujul. Administraatorid ei saa neid muuta. Esmased vajadused mikroteenuste arhitektuuris on sellise lähenemisega tagatud.

### 3.8.2 Staatuslehed

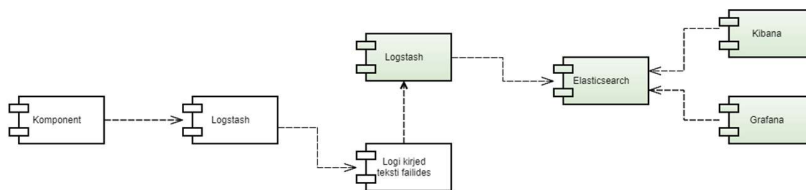
Kasutatakse jälgimissüsteemi Zabbix, mis jälgib põhiliselt infrastruktuuri. Samuti sisaldab iga komponent **/healthz endpointi**, mille poolt jagatud info alusel Zabbix jälgib teenuse olekut – kas teenus on töökorras või mitte ja vajadusel informeerib administraatoreid tekkinud probleemidest. Lisaks annab see infot rakenduse versiooni ja ehitamise aja kohta, kuid kahjuks selline lähenemine ei anna ülevaadet selle kohta, kas äriline teenus toimib või mitte.

### 3.8.3 Trace ID-d

Ruuter saadab päisega igale *backend* komponendile kaasa unikaalse ID, mille maha logimisel (meie komponentide puhul on vastav võimekus juba loodud) saab kergesti siduda eraldi rakendusserverites asuvate ruuteri ja teiste *backend*'i komponentide vahelise liikluse.

### 3.8.4 Parandusettepanekud

- Logide analüüsi läbiviimiseks peaks logid indekseerima, näiteks kasutades *elasticsearch*'i. Samuti kiire indeks annab võimaluse ehitada ärilist monitoorimist logide peal. *Elasticsearch*'i indeksit on võimalik integreerida nii Kibana logide analüüsimise kui ka Grafana hoiatuste tegemise või meetrikate visualiseerimisega.



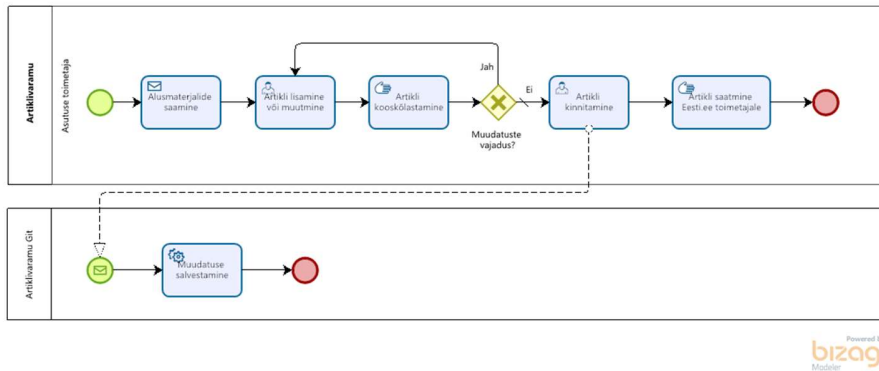
## 4 Olemasolevad protsessid

### 4.1 Riigiteenuste haldus

Kuna kontaktpunkt tulevikus asub eesti.ee portaalis, uurime portaali halduse protsesse, selleks et aru saada, millised on nende muutmise ja parendamise võimalused.

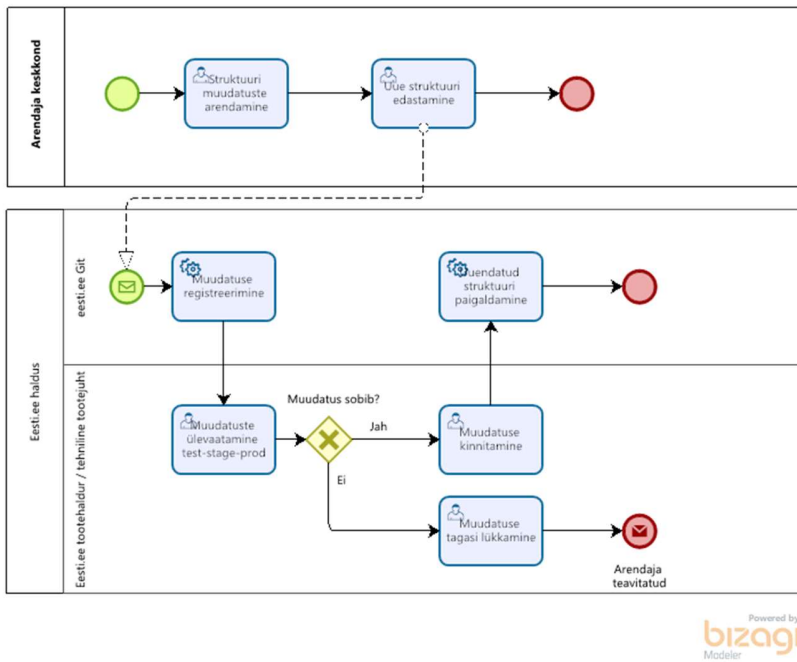
Riigiteenuste kirjeldusi haldavad mitmed erinevad süsteemid. Tulevikuvaate uurimise vajadustest lähtudes kirjeldame olemasolevaid portaali halduse protsesse ja ka muude seotud süsteemide protsesse.

#### 4.1.1 PH.01. Artiklivaramu kirje (artikli) toimetamine



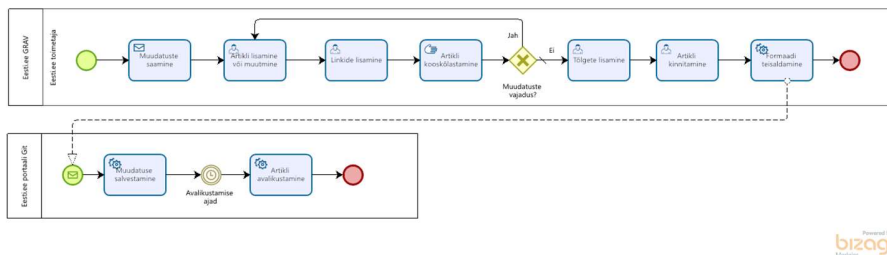
#### 4.1.2 PH.02. Eesti.ee portaali struktuuri täiendamine

Portaali struktuuri täiendamine hõlmab menüü struktuuri muutmist ja täiendamist. Lehtedel on võimalik seadistada oma menüü alamstruktuuri.



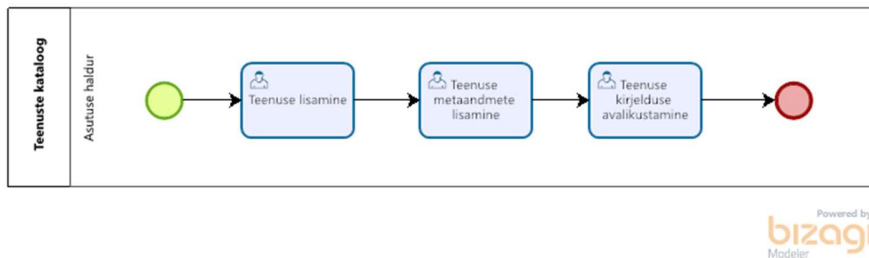
#### 4.1.3 PH.03. Eesti.ee portaali sisu administreerimine

Portaali sisu administreerimine toimub GRAVi sisuhaldussüsteemi kaudu ja artiklivaramu portaali kaudu.



#### 4.1.4 PH.04. Avaliku teenuse kirjeldamine

Avaliku teenuse kirjeldamine toimub teenuste kataloogi süsteemis ja hetkel see ei mõjuta eesti.ee portaali toimimist. Protsess on lihtne ja otsene, teenuste info lisatakse kataloogi, mis ei ole seotud teiste infosüsteemidega.



#### 4.1.5 PH.05. Eesti.ee portaali teenuste arendus/administreerimine

Portaali iseteeninduses teenuste lisamine, muutmine ja muu administreerimine.

## 4.2 Riigiteenuste kasutaja protsessid

Peatükis on toodud ülevaade varasema projekti käigus analüüsitud ja kirjeldatud protsessidest. Käesoleva peatüki eesmärgiks on:

1. anda ülevaade iseteeninduse protsessidest ettevõtja ja kodaniku vaatenurgast;
2. loetleda üles sündmusteenuste uuringu käigus identifitseeritud olemasolevate protsesside probleemid;
3. kirjeldada halduse vaatenurgast vajalikke muudatusi, et lahendada probleemseid protsesse.

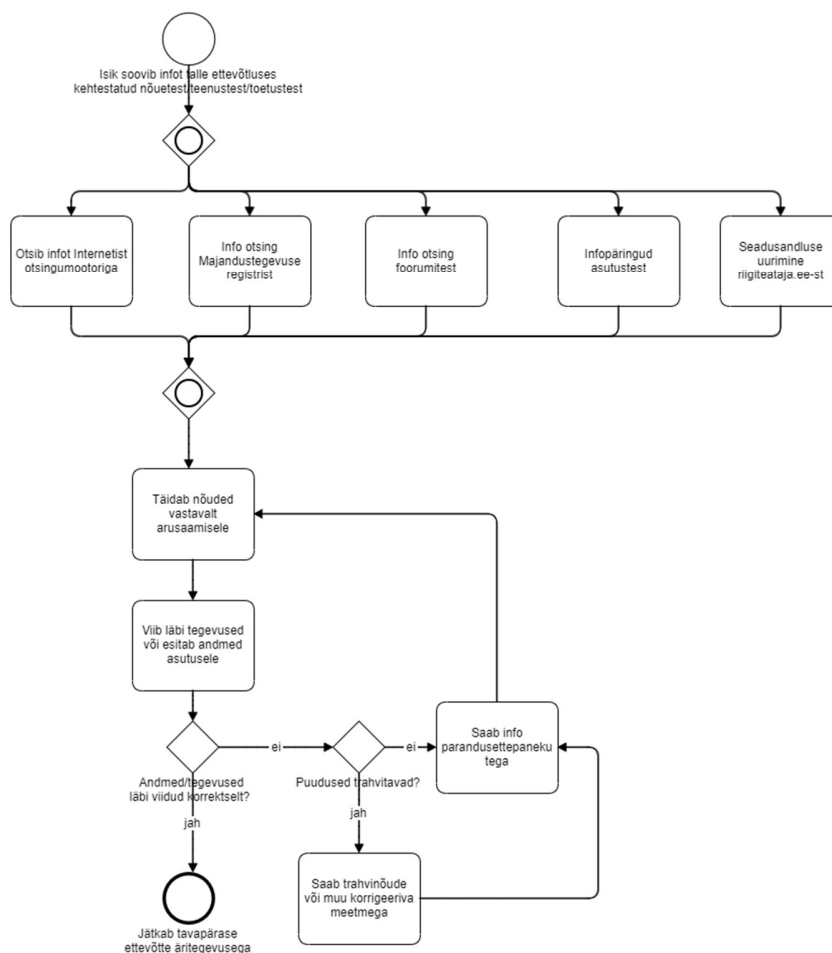
Allpool on toodud sündmusteenuste uuringu käigus kirjeldatud protsessid (protsessi identifikaatorid on lisatud parema viitamise eesmärgil, algdokumentidel protsesside koodid puuduvad). Protsesside kirjeldused sisaldavad soovitatud eesmärgipüstitust.

## 4.2.1 Ettevõtja protsessid

### 4.2.1.1 EP.01. Ettevõtte kohustuste otsing

Protsess on kirjeldatud dokumendis [Lisa 2.3.1 Ettevõtte kohustused, teenused ja toetused.pdf](#)

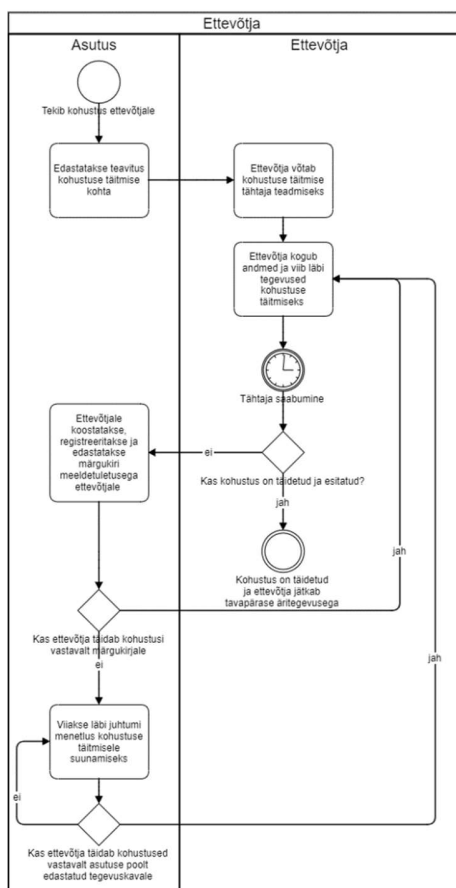
Ettevõtjana, ettevõtlust planeeriva või tegevusala laiendava isikuna saan ülevaate minule kehtivatest riiklikest nõuetest, lubadest ja muudest kohustustest ning nende varasemast täitmisest. Samuti, kuna mulle on toodud välja avalikud mittekohustuslikud teenused ja toetused ja nende kasutamiseks vajalikud nõuded ning tegevused, saan olla kindel, et olen saanud võimaluse kasutada riiklikku tuge oma ettevõtte arendamiseks.



#### 4.2.1.2 EP.02. Ettevõtte riigikohustuse täitmine

Protsess on kirjeldatud dokumendis [Lisa 2.3.2 Ettevõtte sündmuskalender.pdf](#)

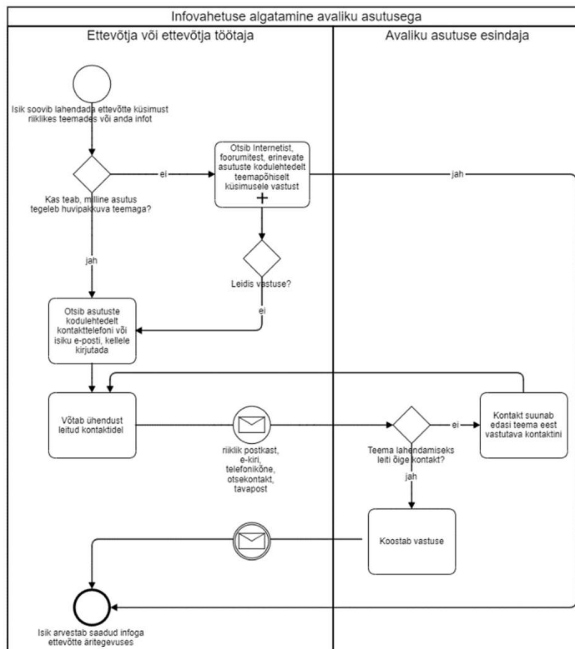
Ettevõtjana näen enda äritegevusele vastavat kohustuste nimekirja koos kohustuse täitmise kuupäevaga ning saan alustada kohustuse täitmist. Aruande esitamise valimisse sattumisel näen info olemasolul ka seda, millistel põhjustel olen valimisse sattunud või millisel eesmärgil kogutavat statistikat kasutatakse, mis motiveerib esitama sisukaid andmeid õigeaegselt.



#### 4.2.1.3 EP.03.1 Ettevõtte infovahetuse algatamine avaliku asutusega

Protsess on kirjeldatud dokumendis [Lisa 2.3.3 Ettevõtte riiklik postkast.pdf](#)

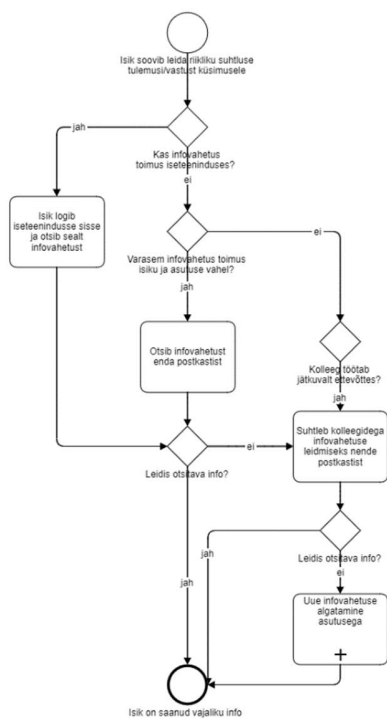
Ettevõtjana, ettevõtte töötajana ja ettevõtte volitatud isikuna saab kontaktpunktis suhelda riigiga huvipakkuvatel teemadel ning oman ülevaadet varasemalt toimunud suhtlusest.



#### 4.2.1.4 EP.03.2 Varasema infovahetuse otsimine

Protsess on kirjeldatud dokumendis [Lisa 2.3.3 Ettevõtte riiklik postkast.pdf](#)

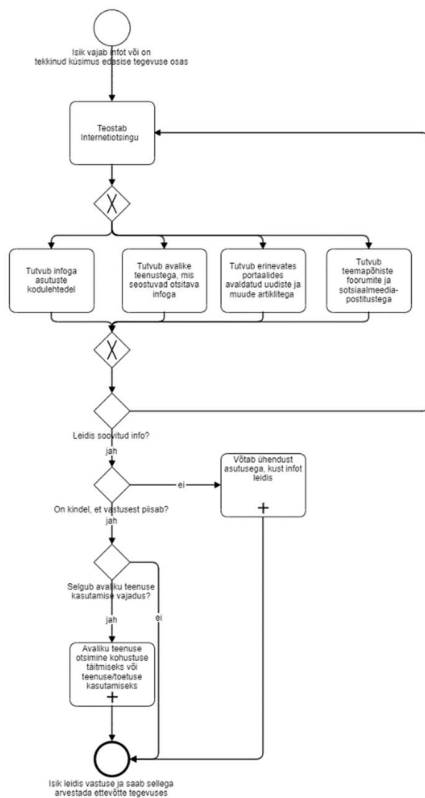
Ettevõtjana, ettevõtte töötajana ja ettevõtte volitatud isikuna saab kontaktpunktis suhelda riigiga huvipakkuvatel teemadel ning oman ülevaadet varasemalt toimunud suhtlusest.



#### 4.2.1.5 EP.04. Infomaterjalide otsing teenuste, võimaluste ja seotud teemade kohta

Protsess on kirjeldatud dokumendis [Lisa 2.3.4 Info ettevõtjale.pdf](#)

Ettevõtjana saan kontaktpunkti enda jaoks olulise fookuseeritud, ajakohase ja süstematiseeritud infomaterjali ettevõtluskeskkonna, pakutavate teenuste, kohalduvate kohustuste ja toetuste kohta ning ei pea otsima infot erinevatest kanalitest (nt asutuste kodulehtedelt, infokirjadest, Riigiteatajast jne). Ettevõtjana soovin leida infot nii enda tegevusala kohta kui ka ettevõtlust üldiselt mõjutavatest teemadest ja muudest mind huvitavatest tegevusaladest.



#### 4.2.1.6 EP.05. Ettevõtte andmete vaatamine

Protsess on kirjeldatud dokumendis [Lisa 2.3.5 Ettevõtte andmekartaat.pdf](#)

Ettevõtjana näen ettevõtte andmeid erinevatest riiklikest registritest.

#### 4.2.1.7 EP.06. Andmetele juurdepääsu andmine

Protsess on kirjeldatud dokumendis [Lisa 2.3.6 Andmetele juurdepääsu andmine.pdf](#)

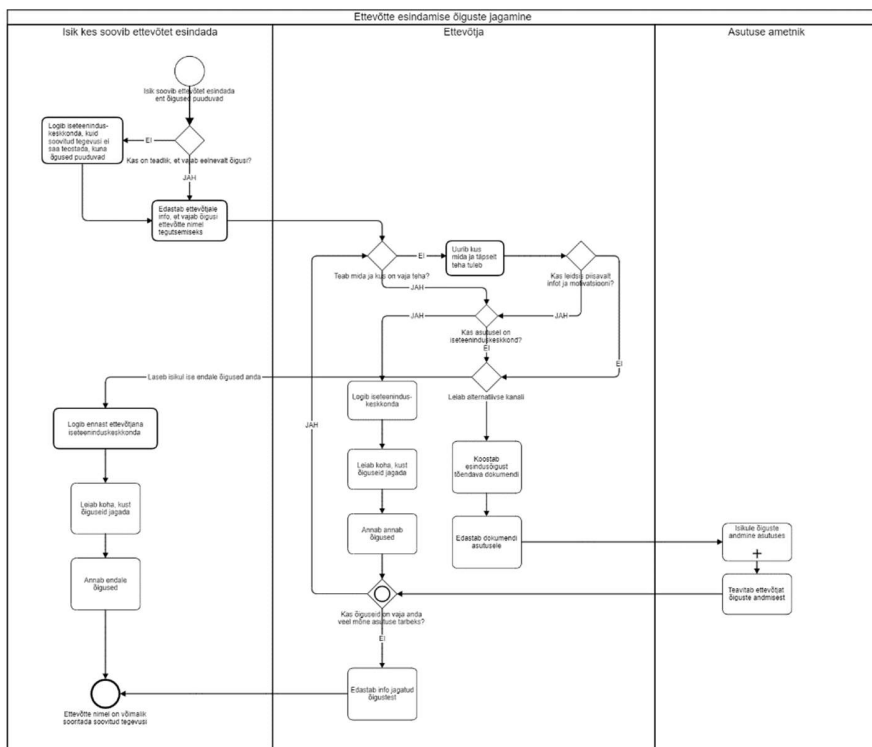
Ettevõtjana saan anda nõusoleku või kutse kolmandatele osapooltele minu ettevõtte andmetega kontaktpunktis või asutuste süsteemides tutvumiseks, et lihtsustada asjaajamist ja anda mulle ettevõtjana kontroll enda andmete üle.

Ettevõtjana saan ennetavalt loobuda riigi pakutavatest proaktiivsetest teenustest.

#### 4.2.1.8 EP.07. Ettevõtte esindamise õiguse jagamine

Protsess on kirjeldatud dokumendis [Lisa 2.3.7 Volitused, rollid ja pääsuõigused.pdf](#)

Ettevõtjana saan hallata enda ettevõttega seotud isikute õiguseid erinevate asutuste süsteemide kasutamiseks ühest kohast ja korraga.



#### 4.2.1.9 EP.08. Majandustegevuse andmete esitamine reaalajas

Protsess on kirjeldatud dokumendis [Lisa 2.3.8 Reaalajas andmete esitamine.pdf](#)

Ettevõtjana ei pea ma tegelema aruannete koostamise ja andmete sisestamisega riikliku aruandluse tarbeks, vaid andmed liiguvad minu majandustarkvarast riigini.

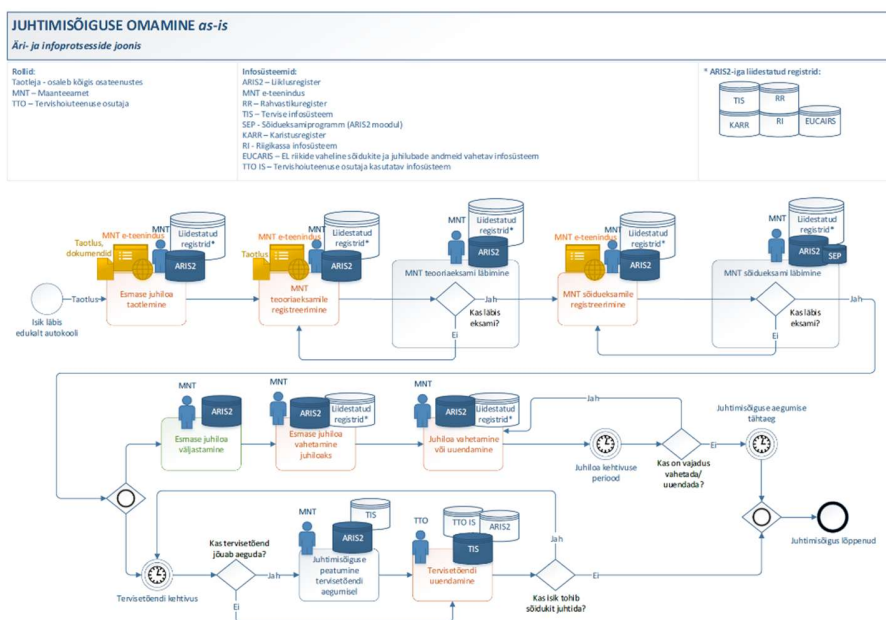
Pilt puudub

## 4.2.2 Eraisikute protsessid

Eraisikute protsessid on kirjeldatud sündmuspõhiselt PWC poolt läbiviidud uuringu käigus.

Protsessid on kirjeldatud vajalike teenuste ja kaasatud osapoolte vaatepunktist iga uuringus käsitletud konkreetse sündmuse kohta. Üldprotsess on välja toodud ainult infovahetuse ja pakutud tehnilise arhitektuuri kujul.

Protsessi kirjelduse näidis:



On kirjeldatud järgmiste sündmuste protsessid:

[SP.01. Abiellumine](#)

[SP.02. Juhimisõiguse omamine](#)

[SP.03. Kuriteo ohvriks langemine](#)

[SP.04. Kutseõppesse õppima asumine](#)

[SP.05. Lahutamine](#)

[SP.06. Lapse hoidmine](#)

[SP.07. Lapse saamine](#)

[SP.08. Liiklusõnnetusse sattumine](#)

[SP.09. Lähedase surmaga tegelemine](#)

[SP.10. Pensionile jäämine](#)

[SP.11. Sõiduki omamine](#)

[SP.12. Töötü olemine](#)

[SP.13. Töövõime vähenemine](#)

[SP.14. Üldhariduses õppimine](#)

## 5 Erasektori, avaliku sektori äriühingute ja sihtasutustest teenusepakkujate analüüs

Erasektori ettevõtted pakuvad teenuseid nii ettevõtete kui ka eraisikute sündmuste raames. Samas on olemas ka rida avaliku sektori äriühinguid ja sihtasutusi, kes pakuvad ettevõtetele sarnaseid teenuseid. Käesoleva analüüsi raames on vaadeldud esmajärjekorras ettevõtete sündmustega seotud teenuseid.

Riigiportaali liidestamine võib eraettevõtetele kaasa tuua lisakulusid. Kõige lihtsam integreerimise viis on tava- või süvalingi lisamine ja seda saab kasutada kõikide teenuste puhul, kuid selline integreerimine toob ka kõige väiksema kasu riigiportaali kasutajale. Analüüsi eesmärgiks on uurida paremaid integreerimisvõimalusi.

## 5.1 Erasektori teenusepakkujate kirjeldus

Alustuseks on mõistlik kaasata riigiosalusega või riigiasutuste poolt loodud avaliku sektori äriühinguid ja sihtasutusi, sel kujul saab riik kontrollida integreerimisprotsessi mõlemat poolt. Esimeses etapis on võimalik kaasata ka kõrge IT pädevusega ja küpsusastmega eraettevõtteid, näiteks telekomi ettevõtteid ja pankasid. Antud kirjeldus ja välja toodud teenusepakkujate nimekiri ei ole lõplik ega täiuslik, see näitab kõige olulisemaid ja tõenäolisi esmajärjekorras liidestatavaid partnereid.

Riigi poolt asutatud äriühingud: [https://www.eesti.ee/est/kontaktid/sihtasutused\\_1](https://www.eesti.ee/est/kontaktid/sihtasutused_1)

Riigi osalusega äriühingud: [https://www.eesti.ee/est/kontaktid/riigi\\_osalusega\\_ariuhingud\\_2](https://www.eesti.ee/est/kontaktid/riigi_osalusega_ariuhingud_2)

Allpool olevas tabelis on välja toodud kõige olulisemad riigiosalusega (avaliku sektori äriühingud ja SA-d) teenusepakkujad ja eraettevõtted, kelle teenused on vajalikud ärisündmuste teostamiseks. Tabelis on toodud riigi poolt loodud sihtasutused, riigi osalusega ja osaluseta ettevõtted. Siin peatükis on käsitletud vaid neid avaliku sektori asutusi (avaliku sektori äriühingud, SA-d, MTÜ-d, avalik-õiguslikud juriidilised isikud), mis pakuvad ka eraettevõtete poolt osutatavaid teenuseid (nt konsultatsioonid, laenud)

Teenuse pakkuja(d)	Teenused	Ärisündmus (CPSV-AP kohandatud sõnastiku alusel)	Elusündmus
Ettevõtluse Arendamise SA	Abi toetuste saamisega	ettevõtte finantseerimine, eksport	-
SA Tartu Teaduspark	Konsultatsioonid, ruumide rent	ettevõtte asutamine	-
SA Tallinna Teaduspark Tehnopol	Konsultatsioonid, ruumide rent	ettevõtte asutamine	-

Maaelu Edendamise SA	Toetused	ettevõtte finantseerimine	-
SA KredEx	Toetused, laenukäendused, kindlustus	ettevõtte asutamine, ettevõtte finantseerimine, eksport	Elukoha otsimine
Pangad	Konto avamine, finantseerimistooted	ettevõtte asutamine, ettevõtte finantseerimine, makseraskused kreditoridele	Tööle asumine, pensionile minek
Telekomi ettevõtted	Sideteenused, veebimajutus	ettevõtte asutamine, töötajatega seotud toimingud	-
E-arve pakkujad (Omniva, Fitek, E-arveldaja)	Arvete vaheldamine	avalikes hangetes osalemine, äritegevuse laiendamine olemasolevas ettevõttes	-
Eesti Liikluskindlustuse Fond	Sõidukite kindlustamise abi	ettevõtte asutamine	Sõiduki juhtimine

Tabelist on näha, et kõige perspektiivsemad eelkäsitletud teenusepakkujad, keda saaks esmajärjekorras liidestada, on SA KredEx ja Ettevõtluse Arendamise SA.

## 5.2 Teenuste kasutamise võimalused sündmusteenuste pakkumisel

Erasektori ettevõtted (ja avaliku sektori asutused, mis pakuvad sarnaseid teenuseid) pakuvad enamasti mugavusteenuseid ja üksikteenuseid.

Vaadates visiooni analüüsidokumendis toodud kasutusjuhtumite vaatepunktist on võimalik jaotada teenusepakkujaid järgmiselt:

Kasutusjuhtum	Teenusepakkujad
Ettevõtte kohustused, teenused ja toetused	Ettevõtluse Arendamise SA, Maaelu Edendamise SA, SA KredEx
Ettevõtte sündmuskalender	vajab täiendavat analüüsi
Ettevõtte riiklik postkast	-
Info ettevõtjale	Kõik, sh e-arve pakkujad
Ettevõtte andmekaart	Eesti Liikluskindlustuse Fond, pangad, SA KredEx
Andmete juurdepääsu andmine	vajab täiendavat analüüsi
Volitused, rollid ja pääsuõigused	vajab täiendavat analüüsi
Reaalajas andmete esitamine	vajab täiendavat analüüsi

### 5.3 Liidestamise võimaluste kasutamine

Erasektori liidestamiseks võib kasutada samu võimalusi, nagu avaliku sektori teenuste liidestamisel.

**Lihtlink.** Võib liidestada kõik teenusepakkujad.

**Süvalink.** Võib liidestada enamus teenuseid, suureks plussiks on, et see viib otse vajaliku teenuse juurde.

**Asutuse (teenuse osutaja) lehe integreerimine (domeeni jagamine).** Sobib eelkõige teenusepakkujatele, kelle infosüsteem kasutab autentimiseks eID vahendeid ja kelle infosüsteemis on olemas kindlad eristatavad teenuste lehed ja vormid.

**Vormilahendusega kirjeldatud vormi kasutamine.** Sobib teenusepakkujatele, kelle teenused on arendatud vormihalduse lahenduse alusel.

**Teenuse täislahendus riigiportaalis.** Riigiportaali lahenduse loomisega võib kaasnedä suurem investeering ja arenduse teostajaks on sel juhul riik. Riigiportaali lahendus sobib ainult pikaajaliselt ja harva muutetavate teenuste arendamiseks. Ülaltoodud põhjuste tõttu enamus erasektori teenustest ei vasta sellele kriteeriumile.

Liidestamise variantidest on eelistatumad: süvalink, teenusepakkuja vormi integreerimine ja vormilahenduse kasutamine. Need variandid pakuvad teenuse kasutajale lisaväärtust kaasnevate infomaterjalide kujul ja samaaegselt ei tekita üleliigset kulu riigiportaali haldusele ja arendusele.

#### 5.4 Liidestamise protsessi variandid

1. Välja arendada mugavusteenuste pakkumise osa sündmusteenuste lehtede jaoks. Teenuste lisamine võib toimuda teenuste kataloogi andmete alusel.
2. Teenuste edendamiseks saab kasutada tarka otsingumootorit (nt pakkuda teenuseid vastava sündmuse või seotud tegevuste otsingu tulemuste esilehel).
3. Saab otsida ja käsitsi lisada teenuseid vastavate sündmusteenuste juurde, sh kasutades teenuste kataloogi ja selle metaandmeid.

Esimese ja teise variandi saab vajadusel ja õigusbaasi olemasolul monetiseerida riigiportaali halduskulude katmiseks.

##### 5.4.1 Liidestamise näidised

###### 5.4.1.1 *Stsenarium 1. Portaali haldur lisab teenuseid sündmusteenuste kirjeldusse*

Ettevõtte, arendaja või portaali haldur koostöös ettevõttega lisab valmislahenduse sündmusteenuse kirjelduse juurde. Ettevõtte liidestamise jaoks on vaja tekitada võimalus lisada teenuste kataloogi eraettevõtete osutatavaid teenuseid.

Võimalikud liidestatavad teenused:

1. **Ettevõtte loomisel konsultatsioonid.** SA Tallinna Teaduspark Tehnopoly ja SA Tartu Teaduspargi teenuseid võib pakkuda süvalingi kujul. Pangad ja teised konsultatsioone

pakkuvad juriidilised isikud, sh SA KredEx, Ettevõtluse Arendamise Sihtasutus liidestatakse süvalinkide või vormihalduse abil.

2. **Pangakonto avamine juriidilise isiku registreerimisel.** Pankade teenused liidestatakse süvalinkide või vormihalduse abil.
3. **Laenu saamine ettevõtte finantseerimiseks.** Pangad ja teised konsultatsioone pakkuvad juriidilised isikud, sh SA KredEx, Ettevõtluse Arendamise Sihtasutus liidestatakse süvalinkide või vormihalduse abil.

#### 5.4.1.2 *Stsenaarium 2. Teenuseid pakutakse automaatselt targa otsingu alusel*

Riigiportaali haldur treenib teenuste leidmiseks otsingumootorit. Teenuste registreerimine toimub sarnaselt esimese variandiga.

Selleks et kasutada süvalinki, tuleb teenuste kataloogis vastav teenus registreerida. Vormilahenduse ja/või integreeritud teenuse pakkumiseks peab teenusevormi registreerima riigiportaali arenduskeskkonnas ja otsingumootor peab kasutama otsingu aluseks ka vormihalduse repositooriumi.

#### 5.5 Riskid

Erasektori (siin ei käsitleta erasektoriga sarnaseid teenuseid pakkuvaid avaliku sektori asutusi) liidestamise riskid on kirjeldatud allolevas tabelis.

	<b>Risk</b>	<b>Kirjeldus</b>	<b>Maandamise meetodid</b>
1	Teenused võivad etteteatamiseta muutuda	Erasektori ettevõtete teenused ei ole otseselt seadustega või määrustega reguleeritud, seega mõned teenuse osutamise tingimused võivad muutuda ilma, et teenuste kasutajaid oleks sellest teavitatud.	Jälgida tehniliselt teenuse versiooni

	Risk	Kirjeldus	Maandamise meetodid
2	Teenuse osutamise lõpust ei teavitata	Eraettevõtted ei ole kohustatud avalikkust teavitama teenuste peatamisest või teenuste osutamise lõpetamisest. Seega tekib risk, et teenus ei ole enam kättesaadav; ja risk, et teenus on tehniliselt kättesaadav (IT lahenduse mõttes), kuid äriliselt ei ole osutatav ja pöördumised jäävad vastamata.	Jälgida tehniliselt teenuse kättesaadavust (nt kasutada Zabbixi vms süsteemi)
3	IT lahendust asendatakse/muudetakse	IT lahenduse uuendamise tõttu teenus ei ole enam kättesaadav, ei ole stabiilne või ei ole kasutatav.	Jälgida tehniliselt teenuse versiooni
4	Teenusepakkuja juriidiline isik likvideeritakse	Eraettevõtete likvideerimisel mõned tegevused võivad tegemata jääda, sh teenuse lõpetamine, teenuse muutmisest ja/või lõpetamisest teavitamine.	Jälgida teenusepakkuja juriidilise isiku olekut äriregistrist Jälgida tehniliselt teenuse kättesaadavust
5	Teenusepakkuja süsteem ei ole stabiilne	Integreeritud teenuse või riigiportaali loodud teenuse puhul on risk, et teenusepakkuja süsteemi vead tekitavad maineprobleeme portaali jaoks.	Jälgida tehniliselt teenuse kättesaadavust Teenuse muutmine probleemsete süsteemide välistamiseks

## 6 Optimeerimise ettepanekud

### 6.1 Optimeerimist vajavad kohad

Vastavalt varasemate uuringute tulemustele vajavad optimeerimist järgmised protsessid:

1. Sama informatsiooni sisestus mitmete erinevate süsteemide puhul.
2. Sama informatsiooni erinev kirjeldus asutuse portaali ja riigiportaali jaoks. Taaskasutamine tihti ei ole otstarbekas või ei ole võimalik.
3. Teenused ei ole omavahel seotud, informatsioon teenusevaheliste seoste kohta puudub või on raskesti kättesaadav.
4. Asutuste süsteemid ja protsessid on erinevatel tasemetel, mistõttu teenuseid ei saa tihti ühtse tervikuna osutada.

5. Teenuse kirjeldused on jaotatud mitme sidumata süsteemi vahel (nt teenuste kataloog, RIHA, riigiportaal, asutuse veebileht), seega sama teenust kirjeldatakse mitmel viisil erinevatel lehtedel.

Sündmusteenuste süsteemi loomisega saab mõjutada süsteemide omavahelist sidumist ja teenuste leidmise parendamist. Teenuse kirjeldamist ja abimaterjalide kättesaamist saab parendada ainult teenuste parema kirjeldamise kaudu. Artiklite taaskasutamise ja kanali spetsiifikaga arvestamise probleemi lahendamine ei kuulu käesoleva projekti alla.

## 6.2 Optimeerimise põhimõtted

Teenuste pakkumise protsesse saab optimeerida erinevate põhimõtete alusel:

1. informatsiooni sisestamise korrastamine,
2. informatsiooni standardiseerimine,
3. infovahetuse automatiseerimine,
4. mittekohustuslike osade ärajätmine.

Käesoleva projekti raames informatsiooni sisestamise korrastamist ja mittekohustuslike osade ärajätmist ei ole võimalik saavutada, sest see nõuab mitmete teenuste ja alamsüsteemide analüüsi.

On vaja täiendavalt silmas pidada, et infotehnoloogiline arhitektuur peab toetama visioonianalüüsis välja toodud kasutusjuhtumeid (ettevõtte kohustused, teenused ja toetused, ettevõtte sündmuskalender, ettevõtte riiklik postkast, info ettevõtjale, ettevõtte andmekaart, andmetele juurdepääsu andmine, volitused, rollid ja pääsuõigused, reaajas andmete esitamine) ja sealt tulenevaid protsesse.

Teenuse osutamise vaatepunktist saab protsesse parandada järgmiselt:

1. Teenus on paremini leitav.
2. Teenuse kasutamine on kasutajale lihtsam: toimub (kasutaja vaatest) ühes süsteemis, ei nõua juba registreeritud ja vahetulemuste andmete sisestamist.

3. Teenusega seotud informatsioon on täiuslik, kasutaja saab informatsiooni erinevatest vajalikest teenustest ja nende variantidest.

On olemas kolm põhivarianti teenuste struktuuri ja informatsiooni haldamiseks:

1. Automaatne paigaldamine ja kirjeldamine teenuste kataloogi alusel (siin mõeldakse teenuse kataloogi all ükskõik millist teenuste metainfo haldussüsteemi, praegu on selleks riigiteenused.ee).
2. Käsitsi vormistamine kasutades Teenuste kataloogi kirjeldusi ja metaandmeid, et hoida teenuse informatsiooni kvaliteedi teenuse osutaja kontrolli all.
3. Käsitsi kirjeldamine Teenuste kataloogi kasutamata.

Igal variandil on oma plussid ja miinused.

Esimese variandi plussideks on hea hallatavus ja teenuse osutamise automatiseerimine. Miinuseks on kasutajakogemuse kehvem tase – automaatsüsteemides on võimalik implementeerida lõplik (ja mitte väga suur) nimekiri kirjeldamise ja vormindamise variante.

Teise variandi puhul on plussideks teenuste vormindamise kõrgem paindlikkus, mis tagab kasutajakogemuse parema taseme, suurema paindlikkuse teenuste kasutamisel, parem teenuse osutamise juhitavus. Miinuseks on keerulisem teenuse uuendamise protsess.

Kolmas variant on hetkel kasutusel ja see ei aita eespool kirjeldatud eesmärke saavutada.

Soovitav on kasutusele võtta teine variant paindlikkuse, kasutajakogemuse parema juhitavuse ja arenduskulude kokkuhoiu tõttu.

### 6.3 Protsesside optimeerimise ettepanekud

1. Artiklivaramu kirjed ja nende uuendused peavad olema vajadusel automaatselt edastatud teistele süsteemidele (nt teenuste kataloogi, riigiportaali), juhul kui see süsteem peab kasutama oma sisuhaldussüsteemi. Parim edastamise variant on muudatuste või uuendatud

versioonide publitseerimine teistele süsteemidele kättesaadavas kohas. Artiklivaramu poolel on vajalik API muudatuste nimekirja ja/või artiklite muudatuste saamiseks.

2. Teenuse kirjeldamise käigus ja teenuse lehe kokkupanemisel peab olema võimalik kasutada artiklivaramu materjale.
3. Pärast teenuse arendamist portaalis, peab teenuste kataloogi kirje olema uuendatud vastavalt teenuse osutamise kanali kirjeldusele.
4. Teenuse kirjeldamine peab olema piisavalt lihtne, et lihtteenuse kirjeldamisega saaks hakkama inimene, kel pole tehnilist tausta.
5. Portaali teenuse arendamine võiks olla seotud teenuste kataloogis kirjeldatud infoga ja portaali kaudu osutatud teenused peavad olema teenuste kataloogis kirjeldatud. See tagab teenuste kirjelduse korrektsuse ja osutatavate teenuste parema leitavuse, kuna kanalite kirjeldused on olemas.
6. Teenuste konfigureerimisel peab olema võimalik kasutada teenuste kataloogi metaandmeid ja need peavad olema algallikate muutmisel portaalis automaatselt uuendatud.
7. Teenuse osutamise ja kirjeldamisega (portaaliga liidestamisega) seotud optimeerimissetepanekud selguvad liidestamise variantide analüüsi käigus.

#### 6.4 Vajalikud muudatused ja seotud väljakutsed

Eesti.ee portaali puhul on tegemist teenuste osutajate jaoks korraldatud edastamispunktiga, mis tähendab lõppkasutaja jaoks teenuste kasutusliidest. Riigiportaal on sel juhul teenuse majutaja rollis. Seega eesti.ee portaali protsessid peavad toetama olemasolevate teenuste pakkumist teistes süsteemides kirjeldatud info- ja andmeobjektide alusel. Teenuste kataloog sisaldab peamiselt teenuste osutamise kanalite kirjeldusi, ehk elektroonilise teenuse juures selle teenuse vormi, alguslehe või API aadressi. Samas on vaja kirjeldada selle teenuse osutamiseks vajalikud protsessid, eriti need, mis puutuvad mitmesammulisse teenusesse või infovahetust vajavatesse teenustesse. See tehniline kirjeldus on alati pakutava süsteemi spetsiifiline ja vajab üldjuhul arendust või eriteadmisi vajavat konfigureerimist. Sündmusteenuste juhtija/teenuse oleku jälgija kui eraldiseisev sündmuste olekuid haldav infosüsteem, mis ei ole riigiportaaali osa ja seega ei too kaasa muudatusi (eraldiseisev süsteem on vajadusel integreeritav välisteenusena).

Teenuste kvaliteedi ja leitavuse parendamiseks on vaja teha järgmised muudatused:

1. CMS süsteem peab kasutama teenuste kataloogis oleva sündmusteenuste struktuuri ja metaandmeid, sh teenustevahelisi seoseid, et pakkuda teenuseid portaali kaudu.
2. Riigiportaal kasutab artiklivaramu süsteemis hoitavaid andmeid. Artiklivaramu süsteemis sisalduv informatsioon on põhiosa teenuste viimistlemiseks.
3. Teenuste kataloog on teenuste kirjelduste ja teenuste avalikustamise ühispunktiks. Uue teenuse pakkumiseks tuleb esmajärjekorras kirjeldada uus teenus ja seejärel arendada seda, kasutades loodud kirjeldust.
4. Linkide hoidmiseks rakendada keskne linkide kogu, mis on kohustuslik, viitamaks riigisüsteemidele ja portaalidele.

Uute protsesside elluviimisel on vaja arvesse võtta teenuste majutaja nõudmisi:

- Teenuste kohaldamine muudatuste järel.
- Teenuste saadavuse kontroll ja mittesaadavaloleku perioodidest (ette)teavitamine.
- Teenuste kirjelduste eelkontroll ja põhjalik testimine.

#### 6.4.1 Teenuste kataloogi muudatusvajadused

1. Teenuste kataloogi andmekogumit on vaja oluliselt täiendada (vt 7.2.1 Loogiline andmemudel)
  1. Olulised täiendused:
    1. teenuste omavaheline seos (teenus-osateenus, seotud teenused),
    2. õigusaktide viited,
    3. kirjelduste viited (artiklivaramu artiklid),
    4. teenuse unikaalne identifikaator,
    5. teenuste atribuudid (nt teenuse sihtgrupi tunnus)
    6. tegevusvaldkonnad
  2. Väheolulised täiendused (riigiportaaali vaatepunktist):
    1. andmekogumid,
    2. piirangud,

### 3. muud atribuudid

2. Teenuste kirjeldamise kasutajaliides peab võimaldama ärisündmuste jaoks defineerida seda, mitmest osateenusest koosnevad sündmusteenused või seda, mis on üksikteenused.
3. Lisada teenuste metaandmete edastamise API, mis jagab teiste süsteemidega sündmusteenuste metaandmeid.
4. Lisada sõnumiruumi, et teavitada teenusepakkujaid teenuse kirjelduse muudatustest.

#### 6.4.2 Artiklivaramu muudatusvajadused

1. Teenuste artiklite haldus (lisaks olemasolevale sisuartiklite haldusele)
2. Artiklite uuendamise sündmustest teavitamine

#### 6.4.3 Tehnilised optimeerimised

Tehnilise optimeerimise ettepanekud on kirjeldatud peatükis [3.1 Riigiportaali arhitektuur AS-IS](#)

Lühidalt võib välja tuua järgmised punktid:

- Hea oleks kasutusele võtta konteinertehnoloogia ja Kubernetese süsteem selle korraldamiseks.
- Autentimiseks kasutusele võtta TARA SSO. TARA SSO peab võimaldama sujuvat üleminekut (ilma nõusolekut küsimata) ühest rakendusest teise, juhul kui andmete töötlemise volituse küsimine ei ole kohustuslik.
- Kaasajastada terminoloogia ja ühtlustada nimetused vastavalt sellele, kuidas neid mujal maailmas kasutatakse. See annab võimaluse arendajatega paremini suhelda ja leida nendega ühine keel. Näiteks
  - ruuter, andmemuundur, TIM – võib võtta kasutusele koondnimetuse (API Gateway). Iga komponent API Gateways täidab küll oma rolli, kuid arendaja jaoks see peab olema läbipaistev.
  - Määrata API Gateway jaoks funktsioonid, näiteks
    - teenuste ühise turvalisuse tagamine ehk päringute autentimine, sessioonide hoidmine (TIM);

- päringute logimine (ruuter);
  - päringute koondamine (Ruuter);
  - päringute formaatide teisendamine (andmemuundur);
  - *API Gateway pattern*'i kohta vt  
(<https://microservices.io/patterns/apigateway.html>)
- Hetkel **/userinfo endpoint** kasutajale kättesaadav läbi TIM-i eraldi domeeni. Kui me lähtume *API Gateway pattern*'ist, siis antud *endpoint* peab olema ka publitseeritud läbi **ruuteri**.
  - API Gateways, ehk ruuteris peab olema võimalik seadistada erinevaid autentimise meetodeid või komponente, olgu see siis TIM või muu autentimise teenuse pakkuja (nt Apareocas või otse TARA SSO). Näiteks OIDC protokollis kasutatakse tavaliselt **/introspect endpointi**, mis on mõeldud *access token*'i valideerimiseks. Paljudes kohtades kasutatakse valideerimiseks **/userinfo endpointi**, kuid spetsifikatsiooni järgi need peavad tagastama erinevat informatsiooni. *Userinfo* tagastab informatsiooni kasutaja kohta, *Introspect* tagastab informatsiooni, mis on seotud konkreetse *token* 'iga.
  - Teenuste registri kasutuselevõtt (*Service discovery*). Annab ülevaate komponentidest, mis on paigaldatud antud keskkonda ja kus (mis URL-i või nime taga) need asuvad. Lihtsustab infrastruktuuri haldamist, näiteks automaatne koormusjaoturi konfigureerimine. Ühendused komponentide vahel peavad toimima komponendi nime järgi (mis peab olema sama kõikides keskkondades), mitte URL-i või IP järgi, mis võivad erinevates keskkondades olla erinevad. See ka lihtsustab komponentide/rakenduste haldamist. Enamasti kasutatakse kaht põhilist printsiipi: serveripoolne tuvastus ja kliendipoolne tuvastus. Keskkondades, kus suhtlus komponentide vahel peab olema keskselt kontrollitav, kasutatakse n-ö serveripoolset tuvastust (*server-side service discovery*). Kui otsustatakse kasutada Kubernetes, siis klastrisisene Service Discovery tuleb kaasa.
  - Ettepanek kasutada standardiseeritud autentimise/autoriseerimise protokoll ja nende realiseerimise, näiteks OIDC. Hetkel näiteks kasutuses n-ö isetehtud protokoll. Selline lähenemine lihtsustaks arendust ja haldust.
  - Sisu ja tekstide osas peab järgima Google'i ja teiste otsingumootorite juhiseid. Hetkel tekstid ei ole hästi indekseeritavad.

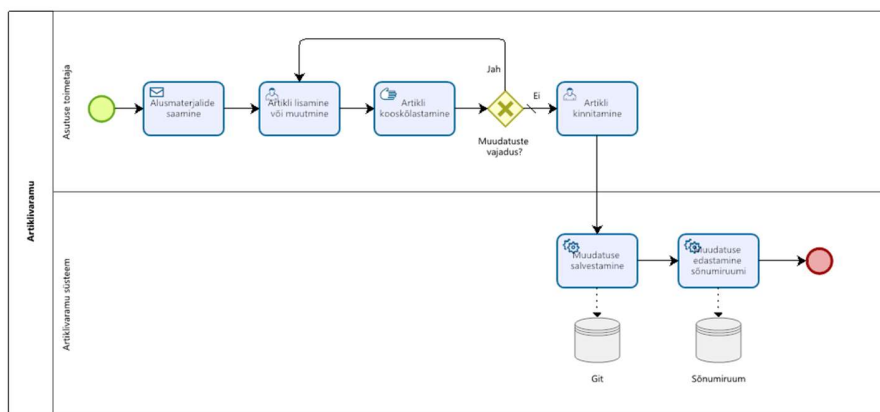
- *Single-Page* lehed peavad sisaldama korrektset metainfo välja.
- Võib kasutada *server-side rendering* tehnoloogiat. **Tänase info järgi seda varianti juba testitakse.**
- Logide analüüsi läbiviimiseks peaks logid indekseerima, näiteks kasutades *elasticsearch*'i. Samuti kiire indeks annab võimaluse ehitada ärilist monitoorimist logide peal. *Elasticsearch*'i indeksit on võimalik integreerida nii Kibana logide analüüsimise kui ka Grafana hoiatuste tegemise või meetrikate visualiseerimisega.

## 7 Kavandatav lahendus (TO BE)

### 7.1 Kavandatavad protsessid

#### 7.1.1 TOBE.01. Artiklivaramu kirje toimetamine

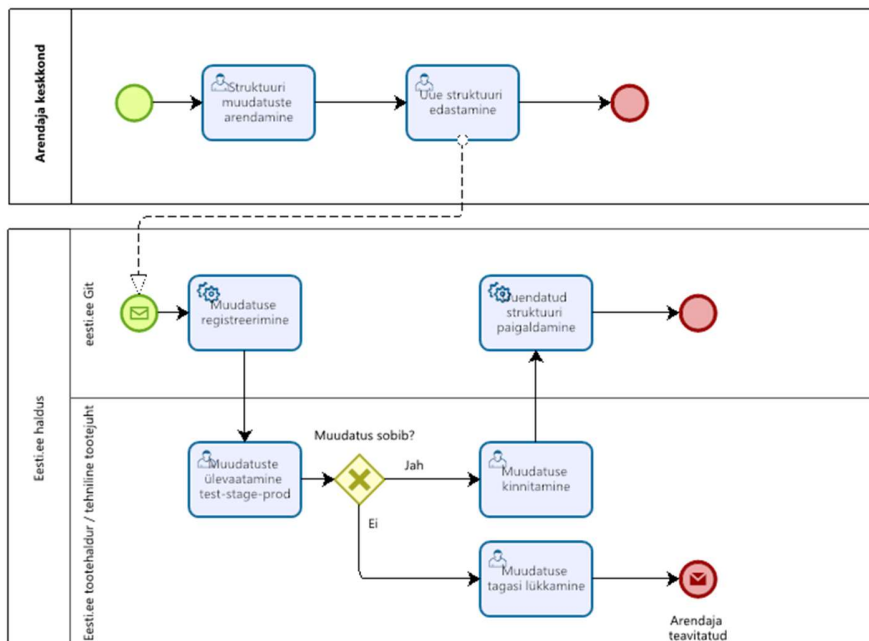
Artiklivaramu kirjete lisamine ja muutmine ei muutu, tulevikus edastatakse artiklid vajadusel automaatselt sõnumiruumi. Artikleid kasutavad süsteemid otsustavad iseseisvalt, kas muudatus puudutab neid ja küsivad vajadusel muudatused artiklivaramu süsteemist. Tulevikus on artiklivaramu sisu ja muutelugude ajalugu kättesaadav API otspunktide kaudu.



Powered by  
**bizagi**  
Modeler

#### 7.1.2 TOBE.02 Eesti.ee portaali struktuuri täiendamine

Portaali struktuuri täiendamise protsessi pole vaja muuta.



Powered by  
bizagi  
Modeler

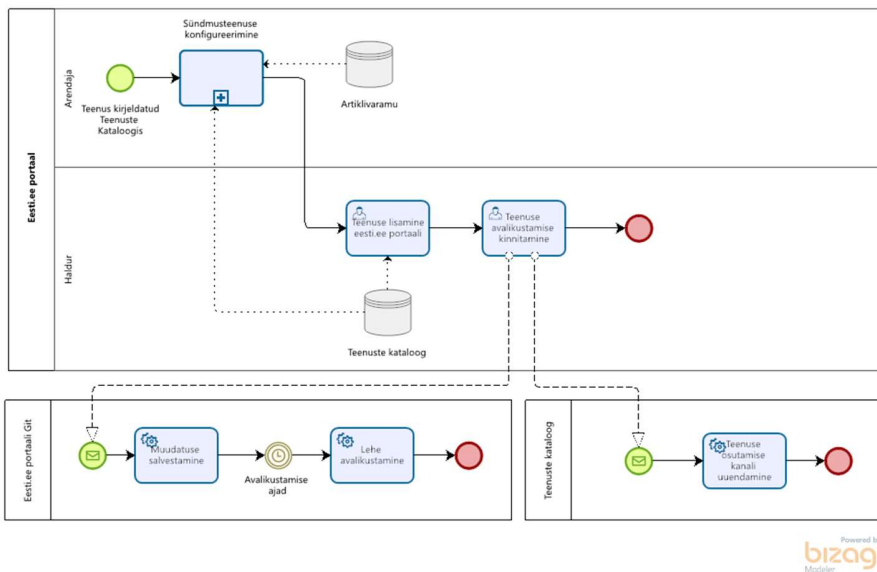
### 7.1.3 TOBE.03. Sündmusteenuse kirjeldamine

Teenuste kirjeldamine algab teenuste kataloogis, portaalis ei saa lisada ühtegi teenust, kui see ei ole kataloogis kirjeldatud. Arendus algab teenuste kataloogi viite lisamisest.

Pärast teenuse arendamist portaalis, peab teenuste kataloogi kirje olema uuendatud vastavalt teenuse osutamise kanali kirjeldusele.

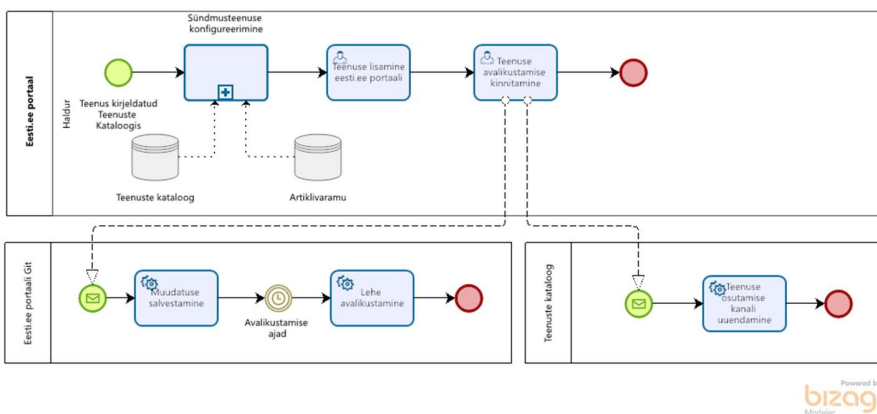
### 7.1.4 TOBE.03.1 Sündmusteenuse arendus

Sündmusteenus hõlmab mitut osateenust. Teenuse konfigureerimise eelduseks on teenuste kataloogi süsteemi lisatud teenuse kirjeldus. Teenuse arendamisel peab kasutama kataloogis kirjeldatud teenuse identifikaatorit ja metaandmeid.



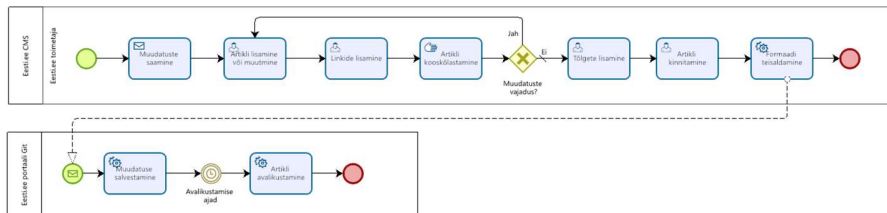
### 7.1.5 TOBE.03.2 Teenuse kasutamine lingina

Lingi (või süvalingi) kujul teenuste kirjeldamine peab olema piisavalt lihtne ka portaali halduri jaoks, teenus peab olema teenuste kataloogi lisatud ja avaldatud enne portaali konfigureerimise alustamist, teenuse arendamisel peab kasutama kataloogis kirjeldatud teenuse identifikaatorit ja metaandmeid.



### 7.1.6 TOBE.03.3 Eesti.ee portaali sisu administreerimine

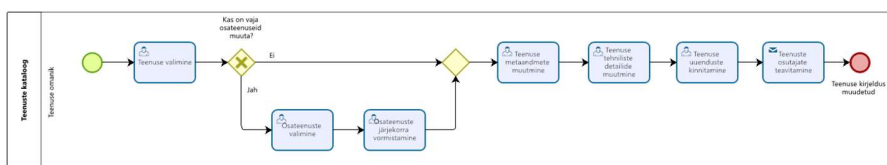
Teenustega mitteseotud portaali sisu haldus ei muutu.



Powered by  
**bizagi**  
Modeler

### 7.1.7 TOBE.04 Avaliku teenuse kirjeldamine

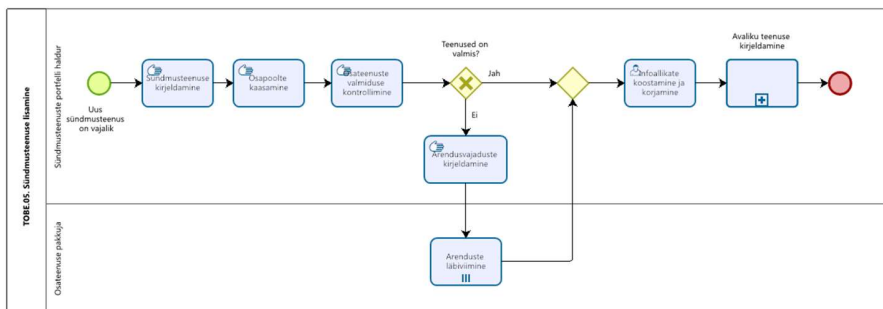
Teenuse omanik on kas sündmusteenuse (osateenustest koosneva teenuse) või üksikteenuse omanik ehk teenuse eest vastutav osapool. Avaliku teenuse kirjeldus peab olema täiendatud, et kompleksteenust oleks võimalik kirjeldada. Avaliku teenuse kirjelduse lisamisel või muutmisel peab sellest teavitama kõiki teenuse osutamist pakkuvaid osapooli.



Powered by  
**bizagi**  
Modeler

### 7.1.8 TOBE.05. Sündmusteenuse planeerimine

Protsess kirjeldab sündmusteenuse planeerimist ja ettevalmistust.



## 7.2 Andmemudel ja andmevood

### 7.2.1 Loogiline andmemudel

Eesti.ee portaali kaudu kättesaadavad teenused on osa Euroopa ühtsest avaliku sektori teenuste raamistikust, seetõttu on loogilise andmemudeli aluseks võetud [CPSV-AP raamistiku](#) mudel. Eesti.ee portaali jaoks on aga vaja lisaandmeid, mida CPSV-AP raamistik ei nõua.

Käesoleva projekti raames on fookus Eesti.ee portaali kaudu kättesaadavatel teenustel, kuid mudelis on toodud ka planeeritava teenuste kataloogi ja artiklivaramu süsteemides hallatavad olemid.



Olem	Ingliskeelne vaste	Kirjeldus
Atribuut	Attribute	Kogum teenust kirjeldavaid lisaandmeid, saab vabalt lisada uusi andmeid. Osa atribuutidest ühilduvad CPSV-AP mudeliga.
Elukaare etapp	Stage	Ärielu või inimese elu suuremad etapid.
Infoallikas	Source of Information	Mittesiduva informatsiooni allikad: abiinfo, foorumid, KKK ja muud taolised ressursid.
Kontaktpunkt	Contact Point	Teenuse osutamise käigus kasutatav infovahetuskanal, eeskätt teenuse osutamise jaoks.
Sündmusteenus	Event Service	Teenus, mis eeldab teiste teenuste osutamist kindlas järjekorras või kindlas mahus. Sisaldab teenuste jada kirjeldust.
Lehe vorming	Page Template	Eesti.ee või muu ressursi kasutatud teenuse või sündmuse esitluse vorming.
Piirang	Criterion	Teenuse saamise eeltingimused.
Sündmus	Event (Lifeline Event)	Äritegevuse või eraelu käigus tekkiv sündmus, mis eeldab riigi- või erasektori teenuste tarbimist. Teenus võib olla eraldiseisev või sündmusteenus. Sündmusteenus eeldab teiste teenuste osutamist kindlas järjekorras või kindlas mahus, seoste ja järjekorra kirjeldamiseks on teenustevahelised seosed.

Olem	Ingliskeelne vaste	Kirjeldus
Sündmusteenus	Event Service	Otsene teenus, mida mitu asutust või eraettevõtet osutab ühiselt, et isik saaks täita kõiki kohustusi ja kasutada kõiki õigusi, mis talle tekivad ühe sündmuse või olukorra tõttu. Sündmusteenus koondab mitu sama sündmusega seotud teenust (edaspidi osateenus) kasutajale üheks teenuseks.
Üksikteenus	Separate Service	Otsene teenus, mida osutab asutus või eraettevõtte, et isik saaks täita ühe sündmuse või olukorra kaudu tekkinud kohustusi ja kasutada neid õigusi. Tehniliselt võib kasutada teiste teenuste tulemusi, kuid olemuselt lõppkasutaja jaoks see on üks teenus.
Osateenus	Part-service	Avaliku sektori teenus või erasektori teenus, mida pakutakse ka Eesti.ee keskkonnas.  Teenus võib olla: 1) <b>erandteenus</b> , mis ei käivitu koos teiste sündmusteenustega; 2) <b>põhiteenus</b> , mis osutatakse sündmusteenuse osana; 3) <b>mugavusteenus</b> – teenus, mis osutatakse ainult teenuse saaja soovil.
Teenus	Service	Osateenuse implementatsiooni ühik, osateenuseid võib olla üks või mitu.  Teenused võivad olla omavahel seotud (seos „seotud teenus“).
Tegevusala	Field of Activity	Hierarhiline tegevusalade klassifikaator, võib kasutada äritegevuste klassifikaatorit, nt EMTAKi või koondklassifikaatorit, mis hõlmab nii äri- kui ka füüsilise isikute vajadusi.



### 7.3 Visiooni dokumendi kasutusjuhtumid

#### 7.3.1 Ettevõtte kohustused, teenused ja toetused / Info ettevõtjale

On võimalik korraldada erinevalt.

Üks variantidest on luua riigiportaalis loogiline ja struktureeritud infoartiklite ja muu materjali struktuur. [Eesti.ee](http://Eesti.ee) portaali sisu administreerimiseks kasutatakse protsessi TOBE.03.3.

Teiseks variandiks on luua artiklivaramu materjalide otsinguteenus, mis kuvab valitud kontekstiga seotud artiklid.

Kolmandaks variandiks on väliste süsteemide teenuste (kohustuste nimekiri, teenuste soovitusel, toetuste pakkumine) koondamine riigiportaali lehele. Integreerimiseks sobib kõige paremini süvalink ja *microfrontend*.

#### 7.3.2 Ettevõtte sündmuskalender

Kompleksteenuse, millel on kaks lahendust:

- eraldiseisev rakendus, mis koondab sündmusi teistest süsteemidest, tulemuse võib riigiportaali integreerida *microfrontend*-lahenduse abil.
- riigiportaali lisatakse leht, mis küsib teistest süsteemidest infot sündmuste kohta ja kuvab selle kasutajale (eelistatum ja jätkusuutlikum variant). Tulemuse võib riigiportaali integreerida *microfrontend*-lahenduse abil.

#### 7.3.3 Ettevõtte riiklik postkast

Eraldiseisev rakendus, mis kuvab riikliku postkasti koos kõikide vajalike tegevustega.

Teenuse kirjeldamine ja integreerimine toimub vastavalt TOBE.03.1 kompleksteenuse arenduse protsessile.

Integreerimiseks sobib tehniliselt domeeni jagamise lahendus.

#### 7.3.4 Ettevõtte andmekaart

On võimalik lahendada kahel viisil:

1. Leht mitme teenusega, mis korjab kõik vajalikud andmed erinevatest registritest kokku.
2. Kompleksteenuse, mis pärib ettevõtte andmeid ja kuvab nende alusel andmekaardi õiges järjekorras ja õiges mahus.

Esimene variant on paindlikum ja lubab lisada paremini struktureeritud infomaterjale. Teine variant arvestab ettevõtte eripäraga andmete pärimisel ja kuvamisel. Mõlemal juhul kasutatakse teenuse kirjeldamiseks materjale, mis sisaldavad teenuse kirjeldust, portaali struktuuri muutusi ja artiklivaramut.

#### 7.3.5 Andmetele juurdepääsu andmine / volitused, rollid ja pääsuõigused

Selleks et teenust saaks osutada riigiportaali kaudu, tuleb teenusepakkujate pääsuõiguste süsteemid integreerida riigiportaali. Integreerimiseks sobib kõige paremini *microfrontend* ja domeeni jagamine.

#### 7.3.6 Reaalajas andmete esitamine

Teenuse ei ole riigiportaalis kuvatud, selle kohta võib olla infoartikkel liidestamise infoga. Selleks et teenust saaks osutada riigiportaali kaudu, tuleb teenusepakkujate süsteemid integreerida riigiportaali. Integreerimiseks sobib kõige paremini *microfrontend* ja domeeni jagamine.

### 7.4 Vastutusmudel

#### 7.4.1 Osapooled

#### **Sündmusteenuste portfelli haldur**

Vastutab sündmusteenuse portfelli arendamise ja juhtimise eest. Omab ülevaadet kõikidest hallatud sündmusteenustest ja tal on volitus nende juhtimiseks.

### **Sündmusteenuse omanik**

Omab ülevaadet konkreetse sündmusteenuse osutamise spetsiifikast ja tal on volitus seda kujundada. Vastutab konkreetse sündmusteenuse arendamise eest.

### **Riigiportaali haldur**

Vastutab riigiportaali toimimise eest, sh selle kaudu osutavate teenuste toimimise eest. Vastutab teiste osapoolte pakutud teenuste eest, mis osutatakse ka riigiportaali kaudu ainult teenuse toimimiseks vajaliku info edastamise osas ja ei vastuta kolmandate osapoolte infrastruktuuris asuvate lahenduste eest.

### **Osateenuse omanik**

Tagab osateenuse arendamise ja vastutab teenuse kvaliteedi eest, omab volitusi ja kohustusi avaliku teenuse korraldamiseks.

### **Osateenuse osutaja**

Tagab osateenuse osutamise, sh teenuse tehnilise kättesaadavuse. Osateenuse osutajaks on sageli osateenuse omanik, kuid esineb ka olukordi, kus teenuse omanik on riigiasutus, kuid sisuline teenus on delegeeritud teisele asutusele või erasektorile.

#### 7.4.2 Tegevused

**Teenuse vahendamise haldamine portaalis**, sh portaali struktuuri haldus. Struktuuri arendamine ja haldus toimub nii arenduse käigus (peamenüü tase, üldine navigatsiooni struktuur) kui ka teenuse konfigureerimise ja portaali sisu arendamise käigus. Üldine vastutus on riigiportaali halduril, kuid arendamine võib olla teostatud nii osateenuse osutaja kui ka kaasatud arenduspartneri poolt (ei ole käesoleva analüüsi raames oluline).

**Artiklivaramu teenuse artikli haldamine – osateenused.** Osateenuseid ja osateenuste osutajat kirjeldavate tekstide haldus. Tegevuse eest vastutab ja seda teostab osateenuse omanik. Tegevuse teostamisel konsulteeritakse osateenuste osutajaga.

**Artiklivaramu teenuse artikli haldamine – sündmusteenused.** Sündmusteenuseid kirjeldavate tekstide haldus. Tegevuse eest vastutab ja seda teostab sündmusteenuse omanik. Tegevuse teostamisel konsulteeritakse osateenuse omanikuga.

**Sündmusteenuste metaandmete (kataloogi) haldamine.** Sündmusteenuse koosseisu, osutamise parameetrite ja muu vajaliku informatsiooni haldus. Tegevuse eest vastutab ja tegevust osutab sündmusteenuste omanik. Tegevuse teostamisel konsulteeritakse sündmusteenuste portfelli halduriga ja informeeritakse osateenuste omanikku ja osutajat.

**Sündmusteenuse konfigureerimine.** Sündmusteenuse tehniline koostamine vastavalt teenuse kirjeldusele. Eeldab arendatud osateenuste, sündmusteenuse kirjelduste ja infotekstide olemasolu. Tegevust osutab riigiportaali haldur ja selle eest vastutab sündmusteenuse omanik, vajadusel konsulteeritakse sündmusteenuste portfelli halduriga ja informeeritakse sündmusteenuse osutamisel kaasatud osateenuste omanikke ja osutajaid.

**Osateenuse arendamine.** Osateenuse tehnilise võimekuse või teenuse arendamine nii ärilisest kui ka tehnilisest poolest (vajadusel arenduspartneri kaasamisega), sh teenuse parameetrite ja muu vajaliku informatsiooni haldus teenuste kataloogis. Tegevuse eest vastutab osateenuse omanik. Tegevuse teostamisel konsulteeritakse sündmusteenuste omanikuga, vajadusel ka sündmusteenuse portfelli halduriga, riigiportaali halduriga ja informeeritakse osateenuste osutajat.

**Osateenuse osutamine.** Tegevuse eest vastutab osateenuse omanik. Tegevust teostab osateenuse osutaja ning juhul, kui teenus on osutatud riigiportaali kaudu, siis on kaasteostaja riigiportaali haldur.

### 7.4.3 Vastutusmaatriks

Vastutusmaatriks ehk RACI mudel on risttabel, mille ridadele on kirjutatud tegevused ning veergudele nendes tegevustes osalevad rollid. Tegevuse vastutused määratakse iga rolli osas vastutustasemena ristuvatesse lahtritesse. Vastutustasemed jagunevad nelja järgmisesse kategooriasse:

**R** – Teostaja (inglise k *responsible*) täidab tegevusega seotud tööülesandeid. Igale ülesandele määratakse vähemalt üks isik, kes selle ülesande eest vastutab. Vastutust võidakse ka jagada. Vastutuse täitjate hulga määrab vastutaja.

**A** – Vastutav (inglise k *accountable*) osapool vastutab tegevuskava täitmise eest. Tema ülesanne on juhtida ja kontrollida teostajate tööd. Ühe ülesande jaoks ei tohiks olla rohkem kui üks vastutaja.

**C** – Nõustaja (inglise k *consulted*) on osapool, kes jagab teavet ja/või annab nõu vastutajatele ja teostajatele. Nõustajad on tavaliselt eksperdid või isikud, keda töötulemused võivad otseselt mõjutada, ning nendega suheldakse kogu tööprotsessi vältel.

**I** – Informeeritavad (inglise k *informed*) on need, kellel ei ole tööprotsessis kindlat ülesannet, kuid keda teavitatakse otsustest ja töötulemustest. Teabevahetus toimub üldjuhul pärast tegevuse lõpetamist. Informeerijad on teostajad või vastutajad.

#### 7.4.3.1 Vastutustabel

	Sündmusteenuste portfelli haldur	Sündmusteenuse omanik	Riigiportaali haldur	Osateenuse omanik	Osateenuse osutaja
Artiklivaramu teenuse artiklite haldamine - osateenused				RA	C
Artiklivaramu teenuse artiklite haldamine – sündmus-teenused		RA		C	

	Sündmusteenuste portfelli haldur	Sündmusteenuse omanik	Riigiportaali haldur	Osateenuse omanik	Osateenuse osutaja
Teenuse vahendamise haldamine portaalis		C	RA	C	
Sündmus-teenuste metaandmete (kataloogi) haldamine	C	RA		I	I
Sündmus-teenuse konfigureerimine*	C	A	R	I	I
Osateenuse arendamine, sh IT võimekuse arendamine	C	C	C	RA	I
Osateenuse osutamine			R*	A	R

\* Riigiportaali täislahenduse puhul

## 7.5 Tehnilised nõuded

Lahenduse loomisel lähtume RIA poolt seatud MFN nõuetest:

1. Lahenduse nõuded on Github-is: <https://e-gov.github.io/MFN/>
2. DevOps poolt seatud nõuded on RIA Infra MFN dokumendis.
3. Riigipilve nõuded on kokku korjatud peatükis [7.7 Pilveteenused](#)

Kasutades Kubernetesit vaja täiendada MFN nõuded:

- <https://e-gov.github.io/MFN/#18.6> - antud nõue ei ole enam pädev.
- Infra MFN vaja täiendada, kuna hetkel see ei sisalda ühtegi reeglit selleks.

**Lisaks võtame arvesse visiooni:**

- Ärifunktsionaalsust mida saab lahendada konfiguratsiooni kaudu (näiteks Ruuteri ja Andmemuunduri kaudu) ei tohi Java koodi sisse kirjutada.
- Kui tekib vajadus uute ühiste komponentide arendamist, siis need komponendid peavad olema taaskasutatavad ja vajadusel lihtsasti väljavahetatavad.
- *Frontend* peab olema tehtud kasutades Veera raamistikku disaini mõttes ja Stencili komponentidena kasutatavaid tükke.
- Kuna meie pakkumine oleks kasutada riigipilve, siis me analüüsis ei käsitle nõuded riistvarale.

### POC-i jooksul tehnilised kitsendused:

- POC-i teeme *frontend*-i osas kasutades Angular-i
- Võtame kasutusele Veera StencilJS komponente ja nende jaoks loodud Angular-i direktiive
  - <https://stash.ria.ee/projects/RIG/repos/ee.eesti.microfrontends/browse>
  - <https://stash.ria.ee/projects/RIG/repos/ee.eesti.microfrontends.angular/browse>
- POC-i backend komponente ja MOCK-e teeme kasutades JAVA Spring Boot

## 7.6 Tulevikulahenduse arhitektuuri kirjeldus

Arhitektuuri kirjelduse koostamisel lähtume sellest, et süsteem peab olema töökindel, skaleeritav ja hooldatav. Samuti peab süsteem olema turvaline.

Erinevate lahenduste võrdlus on peatükis [9. Arhitektuurilahenduste võrdlus](#)

### 7.6.1 Töökindlus

Töökindlus tähendab seda, et süsteem peab olema võimeline tööd jätkama isegi siis, kui süsteemis tekib riistvaraline või tarkvaraline probleem, või kui inimene eksib.

Töökindluse tõstmiseks kasutatakse näiteks klasterdamist. Iga komponent peab süsteemis võimaldama klasterdamist (dubleerimist).

Teise meetmena kasutatakse infrastruktuuri loomiseks ja konfigureerimiseks automaatikat, mis vähendab inimese tekitatud vigu.

Kolmandaks, kõik komponendid ja andmete liiklus komponentide vahel peavad olema jälgitavad, selleks on logimise ja monitoorimise süsteemid.

Antud arhitektuuris eeldame, et süsteemi iga komponent võimaldab klasterdamist (dubleerimist) ja on automaatselt konfigureeritav.

Komponentide dubleerimisel, väja tähelepanu pöörata andmebaaside kasutamisele. Kui tegemist komponentidega mis suhtlevad andmebaasidega otse, siis komponent peab veenduma, et komponendi teised koopiad ei tee samaaegselt sama tööd topelt. Selleks näiteks kasutatakse kirjade lukustamist baasis. Töökindluse tõstmiseks, andmebaase on võimalik replitseerida. Selleks on 2 klassikalist variant: "*master-master*" ja "*master-slave*" lähenemist. Kui andmebaasid replitseeritud "*master-master*" skeemi järgi, siis mõlemad baasi koopiad mängivad süsteemis aktiivset rolli ja suudavad vastu võtma päringuid nii lugemisele kui ka kirjutamisele. Kui andmebaasid replitseeritud "*master-slave*" skeemi järgi, siis üks baasi koopia (*slave*) enamustel juhtudel ei osale süsteemi töös. Kui juhtub, et *master* koopia enam ei saa tööd jätkata, siis süsteem lülitakse ümber *slave* koopia peale.

### 7.6.2 Skaleeritavus

Töökindluse tõstmiseks näiteks, kasutatakse klasterdamist.

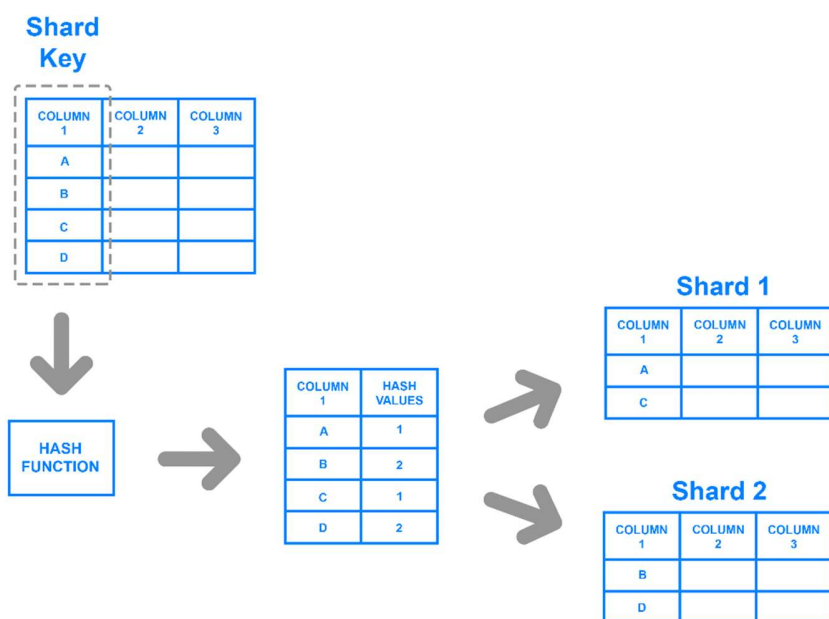
Skaleeritavus on põhiliselt vertikaalne ja/või horisontaalne. Vertikaalne skaleeritavus tähendab seda, et ressursside lisamisel (CPU, Mälu) komponent suudab rohkem tööd teha, näiteks teenindada rohkem paralleelseid kliente.

Horisontaalne skaleerimine tähendab seda, et komponenti on võimalik panna klastrisse ja klastris töötades suudab komponent teenindada rohkem paralleelseid kliente.

Vertikaalne skaleerimine on oluline siis, kui ühe ja sama protsessi jooksul on vaja töödelda suuremat hulka informatsiooni, mida ei saa teha paralleelselt.

Kuna antud süsteemis on enamjaolt tegemist paralleelsete päringutega kasutajatelt, siis meie jaoks on hästi oluline horisontaalne skaleerimine. Lahendus ei tohi sisaldada *stateful* komponente.

Andmebaase on võimalik samuti skaleerida, tihti kasutatakse vertikaalset skaleerimist, ehk lisatakse andmebaasidele ressursse juurde. Teine variant, mis töötab hästi on nõ "*sharding*", ehk andmeid baasis tükeldatakse ja hoitakse eraldi baasides.



### 7.6.3 Hooldatavus

Süsteemiga hakkavad töötama erinevad inimesed (arendajad, administraatorid, haldurid, väliskasutajad jne). See töö peab olema nende jaoks mugav ja kindel.

Süsteem peab olema lihtsasti edasiarendatav, kergesti hallatav administraatori poolt ning arusaadav uutele inseneridele.

Üheks meetmeks, kuidas saab hooldatavust tõsta, on tükeldada süsteem sõltumatuteks komponentideks, et neid saaks lihtsasti välja vahetada. Suhtlus komponentide vahel peab olema lihtne ja standardiseeritud. Nii on võimalik süsteemi järk-järgult edasi arendada ja uuendada.

Teiseks meetmeks on virtualiseerimise taseme tõstmine, näiteks konteinertehnoloogia kasutamine. See annab võimaluse lihtsamini organiseerida alumiste kihtide haldamist, nt välja vahetada riistvara, uuendada operatsioonisüsteeme, migreerida komponente ühest serverist teise jne.

#### 7.6.4 Turvalisus

Süsteem peab olema turvaline. Turvalisus tagatakse erinevate meetmetega.

Esiteks, me kasutame kesksel komponenti, mis annab võimaluse tagada kõikide päringute turvalisus (Turvis). Kui muutuvad turvalisuse nõuded või reeglid, siis see lahendatakse ja testitakse keskselt, vähendades riski, et mingi komponent jääb käsitlemata.

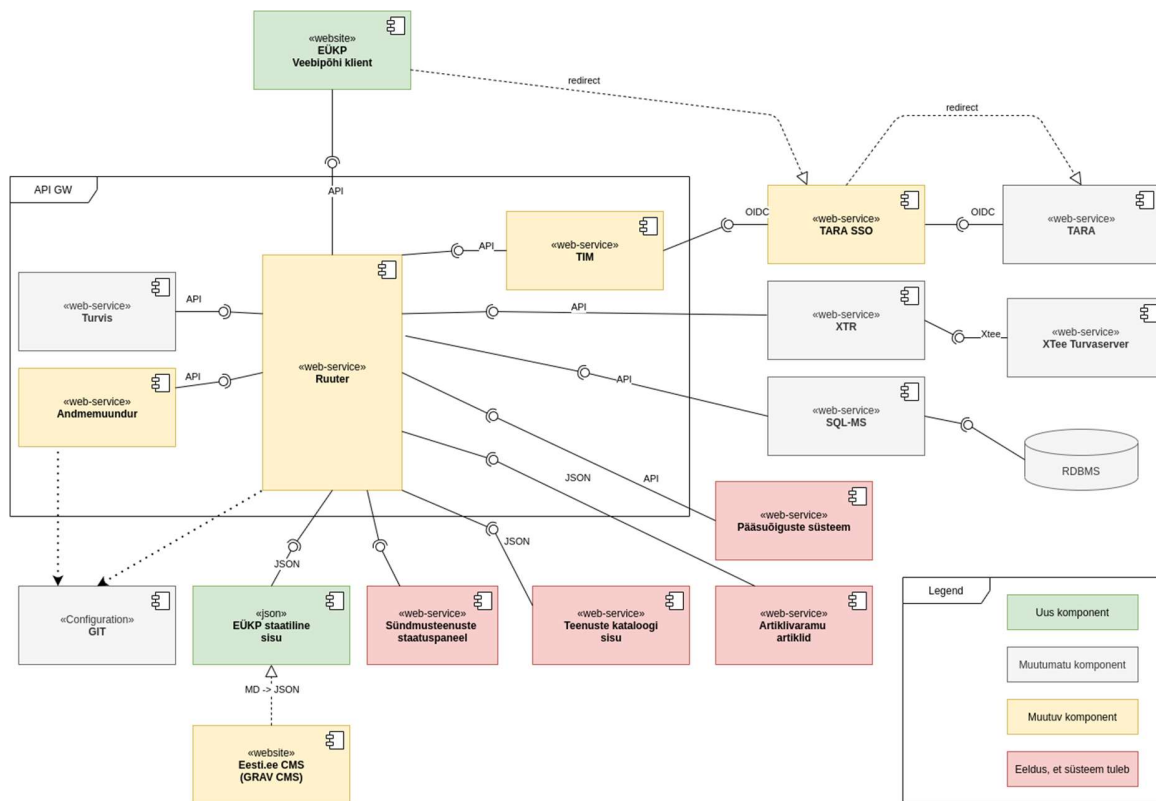
Teiseks, piirangud infrastruktuuris erinevad keskkonnad on eraldatud ja ligipääsud piiratud õiguste ja vajaduste järgi.

Kolmandaks, rakendused mis pakuvad kasutajatele kasutajaliidest, rakendavad kasutajaõiguste kontrolli.

Lisaks kõikidele rakendustele läbi viiakse turvatestimine.

EÜKP projekti raames vaja suuremat tähelepanu pöörata GRAV CMS-i õiguste kontrollile. GRAV CMSi õiguste kontroll peab võimaldama erinevatele asutustele anda õiguse kohandada erinevate lehtede sisu ja seadistusi.

## 7.6.5 Komponentdiagramm



## 7.6.6 Muutmist/arendamist vajavad komponendid

- EÜKPi veebipõhine klient
  - Uus komponent, mis toetab funktsionaalseid nõudeid.
- Eesti.ee CMS
  - EÜKPi veebilehtede sisu haldus
  - Hetkel kasutatakse riigiportaali sisu haldamiseks Grav CMSi.
- Andmemuundur
  - Vaja tekitada uued konfiguratsioonid uute teenuste jaoks.
- Ruuter
  - Vaja tekitada uued konfiguratsioonid uute teenuste jaoks.
  - Muuta konfiguratsioon JSONist YAMLi kujuks, et see oleks kergemini loetav.
- TIM

- Suunata vastu TARA SSO.
- TIMi väljund peab vastama OIDC standardile, hetkel ei vasta.
- Tegeleb lisaks TARA integratsioonile ka sessioonihalduse ja kohandatud JWT-de loomise, signeerimise, valideerimise, pikendamise ja *blacklist*'i haldamisega.

#### 7.6.7 Küsimärgiga komponendid

- Pääsuõiguste haldus
  - Analüüsiprojekti ei ole veel alustatud (hankemenetlus väljas).
  - EÜKPi raames saame ajutiselt kasutada ruuteri konfiguratsiooni, selleks et pärida andmeid otse registritest.
- Artiklivaramu
  - Esialgne versioon on olemas. See sisaldab teabeartikleid. Ei ole selge, mis tingimustel oleks seal võimalik hoida ja sinna lisada sündmusteenuste artikleid.
  - Ajutise lahendusena saame hoida EÜKPi sündmusteenuste artikleid EÜKPi CMS-is.
- Teenuste kataloog
  - Riigiteenused.ee süsteemis teenuste tehnilised kirjeldused ei ole piisava kvaliteediga ja ei sisalda kõiki vajalikke metaandmeid.

#### 7.6.8 Printsüübid

- Kasutajaliides teeb alati päringuid API GWilt (Ruuterilt).
- Ruuter saab konfiguratsiooni abil suunata oma päringu TIMile, et see saadaks kasutaja tunnused. Selle info alusel ruuter tuvastab, kas kasutaja on autenditud või mitte.
- Ruuter koondab andmeid selle kohta, kui mitu teenust on vaja korraga välja kutsuda
- Andmemuundur tegeleb päringuandmete formaadi teisendamisega.
- TIM tegeleb lisaks TARA integratsioonile ka sessioonihalduse ja kohandatud JWT-de loomise, signeerimise, valideerimise, pikendamise ja *blacklist*'i haldamisega.
- Üksikuid komponente peab olema võimalik vajadusel välja vahetada

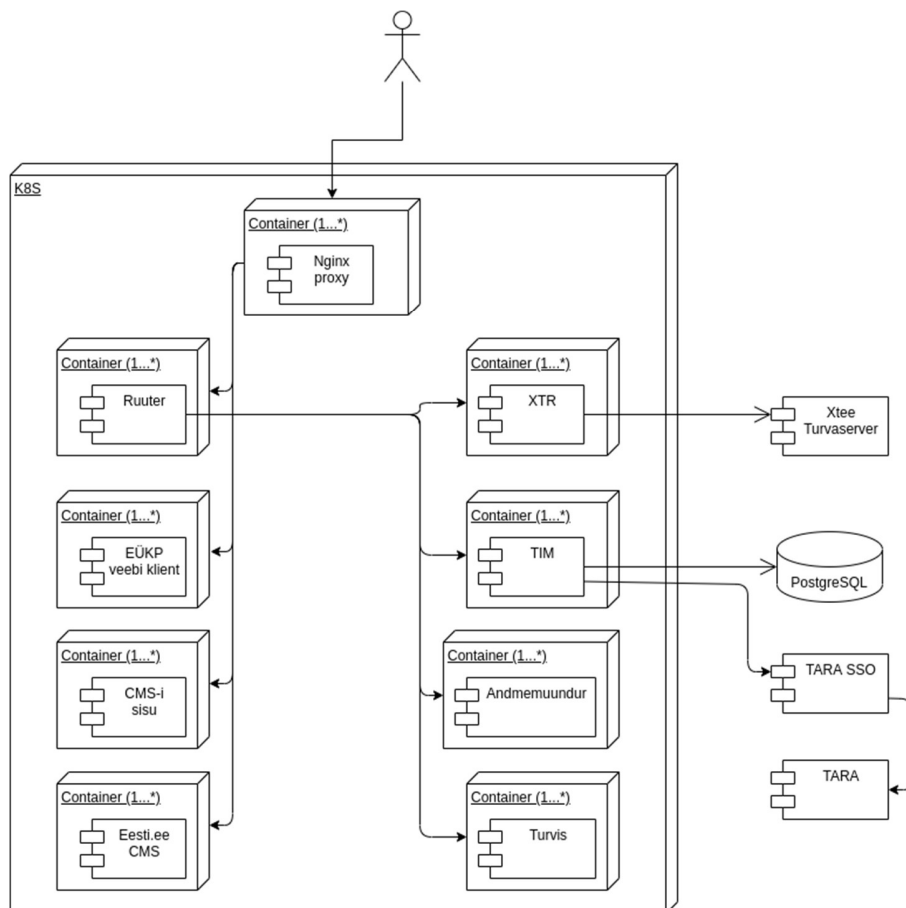
### 7.6.9 Skaleeritavus, tõrkekindlus

- Kasutatavad komponendid on hästi skaleeritavad (nii vertikaalselt kui ka horisontaalselt).
- Arhitektuur on hästi hajutatud mitte ainult komponentide vahel, vaid ka asutuste vahel. Ühe teenuse mittetoimimine ei takista teiste teenuste toimimist.
- Autentimist ja autoriseerimist tagavad juba valmis arendatud komponendid, kasutades selleks standardseid protokolle (TARA, TIM, RUUTER).

Komponent	Tehnoloogia	Kirjeldus
EÜKPi veebipõhine klient	Angular	Staatiline sisu, mis laetakse kasutajate veebilehitsejasse. Võib publitseerida Nginx'i kaudu.
Ruuter	Java, Spring Boot	Hästi skaleeritav komponent, mis töötab konfiguratsiooni alusel, ise ei oma andmebaasi ja ei hoi tegevuse olekut. Paralleelselt saab käivitada mitu koopiat, mis tõstab tõrkekindlust. Komponendi töökiirus sõltub väljakutsutavate komponentide kiirusest.
Andmemuundur	Java, Spring Boot	Hästi skaleeritav komponent, mis töötab konfiguratsiooni alusel, ise ei oma andmebaasi ja ei hoi tegevuse olekut. Paralleelselt saab käivitada mitu koopiat, mis tõstab tõrkekindlust.
XTR	Java, Spring Boot	Hästi skaleeritav komponent, mis ise ei oma andmebaasi ja ei hoi tegevuse olekut. Paralleelselt saab käivitada mitu koopiat, mis tõstab tõrkekindlust.
TIM	Java, Spring Boot	Hoiab kasutajate sessioone PostgreSQL andmebaasis. Tõrkekindluse osas andmebaas on üks koht, mis võib tekitada tõrkeid. Rakendust saab vajadusel horisontaalselt skaleerida.
TARA SSO	x	Ei ole plaanis projekti käigus seda käsitleda.
TARA	x	Ei ole plaanis projekti käigus seda käsitleda.
Xtee turva server	x	Ei ole plaanis projekti käigus seda käsitleda.
Turvis	x	Ei ole plaanis projekti käigus seda käsitleda.
Eesti.ee CMS	Grav CMS	Lehtede sisu ja seaded on staatilised. CMSi maasolek ei tohi mõjutada veebi toimimist.
EÜKPi staatiline sisu	JSON	Staatiline sisu JSONi kujul, mis laetakse kasutajate veebilehitsejasse. Võib publitseerida Nginx'i kaudu.

## 7.6.10 Paigaldus

Idee järgi saab kõiki komponente paigaldada k8s klastrisse, Dockeri konteineritesse.



Paigaldamise osas on ettepanek kasutada pilveteenust (võrdlus on eraldi välja toodud peatükis [7.7. Pilveteenused](#)) ja Kubernetese klastrit.

EÜKPi veebi kliendi Docker *image* võib olla ehitatud, kasutades NodeJs baas *image*'it, et oleks võimalik organiseerida SSR (*Server-Side Rendering*).

Ruuteri, XTRi, TIMi, andmemuunduri komponendid on Javas kirjutatud ja võivad kasutada vastavad Java baas *image*'it.

Kuna CMSi sisu on staatiline JSON, siis seda saab publitseerida, kasutades näiteks Nginx'i.

Kui valida suunaks, et kõik komponendid on k8s klastris, siis Grav CMSi saab ka panna sinna, kuid see ei ole otseselt vajalik.

#### 7.6.11 Ressursside planeerimine

Hetkel on teada kaheksa komponenti, mida on plaanis paigaldada.

Igast komponendist vaja vähemalt kaht koopiat, et tagada liiasus (*redundancy*), ehk kokku 16 PoD-i k8s klatri sisse.

Iga PoD tahab keskmiselt saada 2GB mälu ja 1vCPU, ehk kokku 32GB mälu ja 16vCPU.

Peab arvestama, et Kubernetes vajab enda protsesside käivitamiseks ka ressursse, keskmiselt 2GB mälu ja 1vCPU iga NODE kohta.

Üheks keskkonnaks võib planeerida k8s klatri, mille sees on 3 NODEi ehk iga VM on järgmise konfiguratsiooniga: vähemalt 16GB mälu ja vähemalt 6 vCPU.

Kuna üleval toodud komponendid ei ole kunagi olnud sellise konfiguratsiooniga, siis prognoos on hästi umbkaudne.

Riigiportaali kasutatavuse analüütika prototüüpimise projekti analüüsist saab välja lugeda, et keskmiselt on riigiportaalis 12 000 unikaalset sessiooni päevas.

Võttes arvesse, et päevas on umbes 12 aktiivset tundi, kus inimesed toimetavad, siis see teeb umbes 1000 unikaalset sessiooni tunnis.

Selleks, et tuvastada, kas sellise konfiguratsiooniga süsteem suudab ära teenindada kõik kasutajad, on vaja läbi viia koormusteste.

## 7.7 Pilveteenused

Tehnoloogiast, mida RIA on valmis kasutusele võtma (näiteks Kubernetes), sõltub lahendus, mida on EÜKPi projekti raames mõistlik soovitada.

Edasi tuleb võrdlustabel ja innovatsioon.

Hetkel on teada järgmised nõuded, mis tulevad ärivajadustest.

- X-tee turvaserver on olemas või VPNi ühendus võrku, kuhu on paigaldatud X-tee turvaserver.

## 7.7.1 Võrdlustabel

	Kohapeal	Riigipilv	Muu pilv
Kirjeldus	Riistvara on paigaldatud majasiseselt, oma või renditud serveriruumides. Asutusel on täielik kontroll oma füüsilise riistvara üle. Serveriruumid võivad asuda seal, kus seda nõutakse, näiteks Eesti territooriumil. <i>Enamus asju RIAs praegu kasutab antud varianti.</i>	Riigipilv on infrastruktuuri teenus (enamjaolt IAAS), mida pakutakse riigiasutustele riigisüsteemide majutamiseks. Riigipilve nõuded on avalikult kättesaadavad siin: <a href="https://www.riigipilv.ee/files/riigipilve_alusno_uded_v.2.pdf">https://www.riigipilv.ee/files/riigipilve_alusno_uded_v.2.pdf</a> Järelevalvet teostab RIA. Asukoht on Eesti territooriumil. <i>Mõned teenused juba kasutavad riigipilve.</i>	Kõige suuremad pilved on Microsoft Azure, Amazon AWS ja Google Cloud. Suuremad teenusepakkujad ei asu Eesti territooriumil, kuid kõik nendest pakuvad teenuseid ELi territooriumil. Riigiasutuste jaoks seda valikut on raske põhjendada, kui on teada, et süsteem töötleb isikuandmeid.
X-tee turvaserver	x	x	-
VMide kasutamise võimalus	x	x	x
Kubernetesi teenusena kasutamise võimalus	-	x	x
Ei pea haldama riistvara	-	x	x
Asukoht on Eesti territooriumil	x	x	-
Võimalik rakendada ISKE turvameetmeid	x	x	-
RIA strateegiline suund	-	x	-
Võimalus ühendada vana infrastruktuuriga	x	x	x
Maksa-väljudes ( <i>Pay-as-you-go</i> ) hinnastamise mudel	-	x	x
Ei pea tegelema infrastruktuuri teenuste standardiseerimisega ja dokumenteerimisega	-	x	x

## 7.8 Innovatsiooni stimuleerimine

Kuna riigi jaoks on riigipilv strateegiline suund, siis kindlasti peaks kasutama riigipilve teenuseid. Riigipilv pakub alates 2020. aasta algusest teenusena Kubernetese klatri kasutamise võimalust.

Kuna Kubernetes on riigi teine strateegiline suund, siis oleks mõistlik alustada vähemalt ühe projektiga, mida saab piloodina käivitada riigipilves, kasutades Kubernetese teenust.

Selline samm annab võimaluse ka teistele projektidele, et nad saaksid kasutada tänapäeval levinud toiminguid ja tehnoloogiaid.

# 8 Pääsuõigused

## 8.1 Visioon

Pääsuõiguste teenus peab olema EÜKPi süsteemi jaoks nagu iga teine teenus, API Gateway (ruuteri) kaudu konfigureeritud ja publitseeritud API. EÜKPi süsteemis ei ole tarvis pääsuõigusi hoida ega hallata.

Pääsuõigused võivad olla seadusest tulenevad õigused, näiteks ettevõtte juhatuse liige, või õigused antud volituse alusel, näiteks juhatuse liige volitas raamatupidajat esitama Maksu- ja Tolliameti süsteemis deklaratsiooni.

Tänapäeval iga asutus hoiab õigusi enda süsteemides, keskset üleriigilist pääsuõiguste süsteemi hetkel ei ole. Olemas on AARi lahendus (<https://www.eesti.ee/est/autoriseerimine>), mille eluiga on lõppemas ja mis ei täida keskse süsteemi rolli.

Kasutaja mugavuse saavutamiseks peavad pääsuõigused olema asjakohased ja ühest kohast paika pandud. Näiteks õigused, mis on antud konkreetsele teenusele Maksu- ja Tolliameti süsteemi kaudu, peavad rakenduma ka samale teenusele riigiportaali kaudu.

Kasutaja ei pea riigiportaalis eraldi õigusi seadistama. Pääsuõiguste seadistamisel peaks lähtuma *Once-Only* printsiibist.

EÜKPi süsteemi kontekstis piisab rollipõhisest autoriseerimisest (RBAC), kuid üksikud teenused EÜKPi portaalis võivad sisaldada ka täpsemat õiguste kontrolli, näiteks atribuudipõhist (ABAC).

## 8.2 Lahendus

Visiooni lahendamiseks on vähemalt kaks varianti, millel on oma plussid ja miinused. Üks võimalik variant (**Lahendus A**) oleks luua või kasutusele võtta üleriigiline keskne õiguste süsteem, nagu vana AARi lahendus (<https://www.eesti.ee/est/autoriseerimine>).

Teine variant (**Lahendus B**) oleks pärida õiguste informatsiooni otse (või läbi *proxy*) asutuste süsteemidest, näiteks Maksu- ja Tolliametite teenuste kasutamiseks pärida õigusi EMTA süsteemist.

### 8.2.1 Lahendus A

RIA hoiab keskselt suurt pääsuõiguste andmebaasi, kuhu teised asutused edastavad informatsiooni, et oleks võimalik selle asutuse teenuseid kasutada. RIA vastutaks pääsuõiguste teenuse ülevõtmise eest ning asutused vastutaks informatsiooni eest, mis hoitakse selles süsteemis.

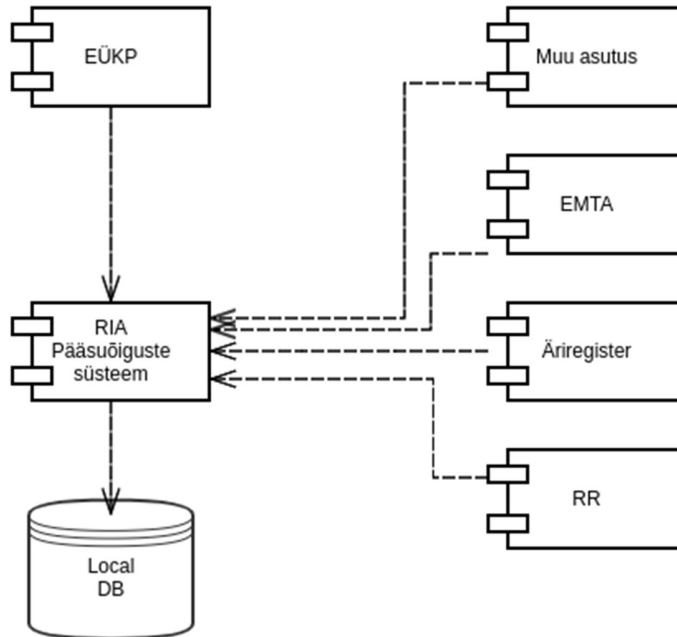
Tehniliselt võiks selline lahendus välja näha nii, et iga kord, kui asutuse süsteemides kasutajatel õigused muutuvad, siis neid muudatusi edastatakse RIA keskele süsteemile. Asutused hakkaksid vastutama mitte ainult enda süsteemi toimimise eest, vaid ka andmete sünkroniseerimise toimimise eest.

Tänapäevane AAR-lahendus võimaldaks tehniliselt seda korraldada. Asutused saavad ise hallata informatsiooni AAR-süsteemis ning luua klassifikaatorid vastavalt asutuse nõuetele.

Selle lahenduse probleemiks on see, et asutused ei ole huvitatud selle informatsiooni edastamisest ja integratsioonide arendamisest. Enamjaolt asutused võtavad kasutusele oma pääsuõiguste süsteemi. See on organisatoorselt lihtsam ja kasutajatele mugavam. Informatsiooni haldamine ja kasutamine enda süsteemis on palju lihtsam. Juurutatud süsteemid võivad täita erinõudeid, mis on

asutusele eriomane. Asutuste jaoks on enda juurutatud süsteem töökindlam, tootlikum ja riskivabam variant, kui AARi taolise teenuse kasutamine.

Antud lahenduse puhul peab enamik asutusi olema huvitatud lahenduse kasutamisest.



### 8.2.2 Lahendus B

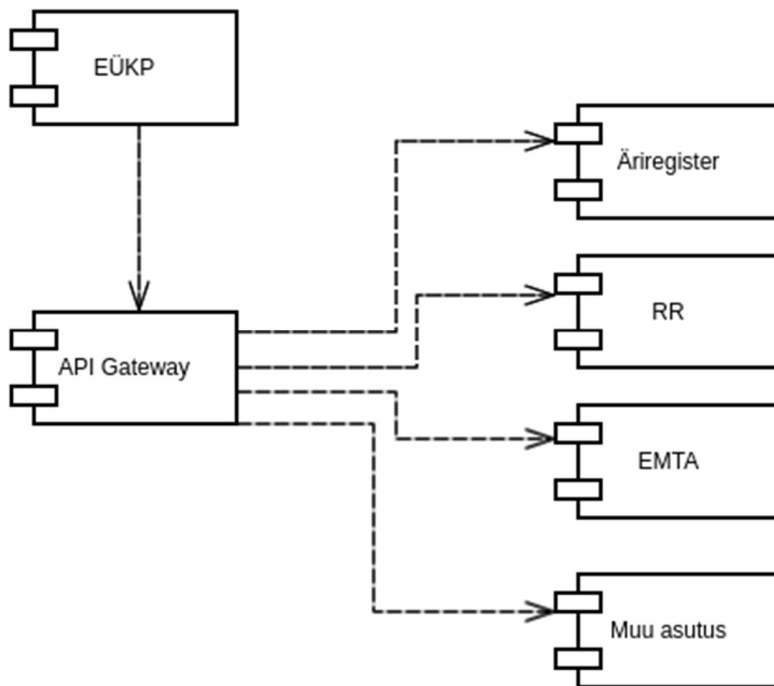
PoCi raames on lahendatud võimalus pärida õigusi AARist , äriregistrist, rahvastikuregistrist, koondada vastused ja tagastada kasutajale informatsioon, mis rollis võib kasutaja süsteemis olla.

[https://bitbucket.ria.ee/projects/RIG/repos/ee.eesti.java.ruuter/browse/configurations/PERMISSIONS/SET\\_ROLE.json](https://bitbucket.ria.ee/projects/RIG/repos/ee.eesti.java.ruuter/browse/configurations/PERMISSIONS/SET_ROLE.json)

[https://bitbucket.ria.ee/projects/RIG/repos/ee.eesti.java.ruuter/browse/configurations/PERMISSIONS/SET\\_ROLE\\_RESTRICTIONS.json](https://bitbucket.ria.ee/projects/RIG/repos/ee.eesti.java.ruuter/browse/configurations/PERMISSIONS/SET_ROLE_RESTRICTIONS.json)

Selle PoCi (SET\_ROLE\_RESTRICTIONS.json) raames on lahendatud vaid üks kasutusjuhtum, kus kasutajal on võimalik võtta kas enda laste või ettevõtte esindaja roll, et lugeda nende e-posti. Teiste kasutusjuhtumite rakendamiseks on vaja luua sarnased konfiguratsioonid, mis pärivad andmeid teistest andmekogudest, näiteks EMTA süsteemist.

Kohandatud JWT-de tasemel luuakse N arv mistahes nime, struktuuri, sisu ja kehtivusajaga *token*'eid, mille eesmärgiks on vähendada pidevalt väliste registrite poole pöördumist ning hoida tulemusi signeeritud kujul. PoCi raames hoiti vastust n-ö vahemälus, see realiseeriti kohandatud JWT-de abil.



Selle lahenduse puhul peab RIA arvestama, et iga uue pääsuõiguste informatsiooni pakkuja puhul on vaja teada asjaolusid:

- et pakkuja (RR, Äriregister jne) oleks võimeline seda informatsiooni välja andma;
- et RIA tekitaks uue integratsiooni konfiguratsiooni faili/loogika;

- Süsteempäringud võtavad aega ja mida rohkem süsteeme on ühes jadas, seda ajakulukam see on. Lahenduseks võib olla taustal pärimine ja *cache*'i kasutamine.
- Osa päringuid võivad olla vigased. Lahenduseks võib olla see, et kasutajale näidatakse ainult osa informatsiooni või kuvatakse teade, et osa teenuseid ei ole hetkel kättesaadavad.
- Informatsioon erinevates süsteemides võib kattuda või olla vastuolus, see tähendab, et iga uue pakkuja liitumiseks on vaja koostada analüüs ja plaan.

### 8.2.3 Kokkuvõtte

Päasuõiguste osas peab aru saama, et kõikide asutuste süsteemid on erinevad, neid on arendatud erinevatel aegadel ning et neid kasutavad erinevad tehnoloogiad ja COTSi (*Commercial off-the-shelf*) komponendid. Asutustel on ka erinevad vajadused ja lahendused, tagamaks ärilisi vajadusi.

Hajutatud süsteemid on iseenesest õige valik töökindluse ja tootlikkuse kontekstis. Teisest küljest, hajutatud süsteemid tekitavad probleeme, kui teenuseid on vaja koonda ühte kohta, nagu riigiportaal seda teha üritab.

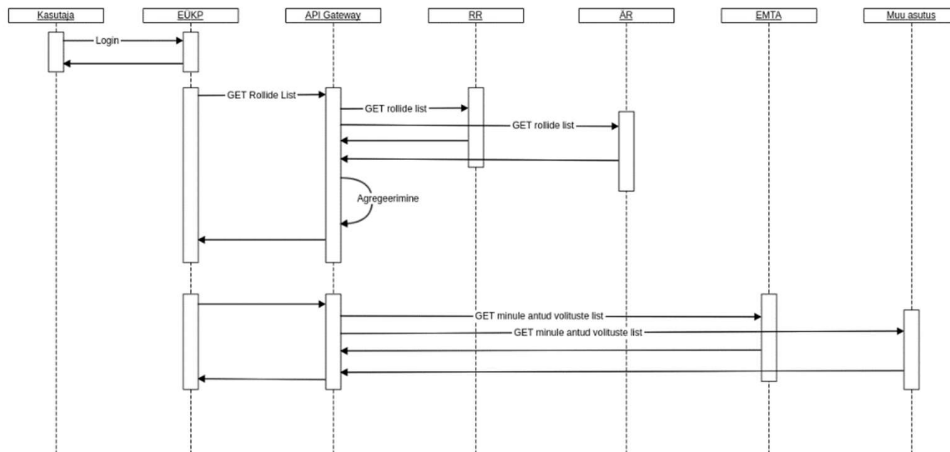
Riigis võiks olla üks teenus, mis koondaks kokku seadusest tulenevad õigused erinevatest registritest, näiteks äriregistrist, rahvastikuregistrist jne. Teenus peab koondama informatsiooni reaajas ja hoidma lühiajaliselt mälus, kuid informatsiooni põhiallikas (*master*) peab olema ikka vastav register. Teisest küljest, volituse alusel tulenevad õigused peavad tulema lisapäringuga teenusepakkuja süsteemist.

Portaal peab võimaldama sündmusteenuste käivitamist vastavalt rollile, kuid teenusepakkuja peab kontrollima õigusi oma poole pealt.

EÜKPi toimimiseks on vaja kolme teenust:

- rollide loetelu (esindatavate isikute loetelu);
- rolli valimine;
- kasutaja õiguste loetelu valitud rolli kontekstis. Antud teenus ei ole kohustuslik, kui teenusepakkuja ise kontrollib õigusi või annab võimaluse seda teha lisapäringuga.

Olukorras, kus portaal on mõeldud ainult info koondamiseks, mitte hoidmiseks, on kõige mõistlikum kasutada **lahendust B**.

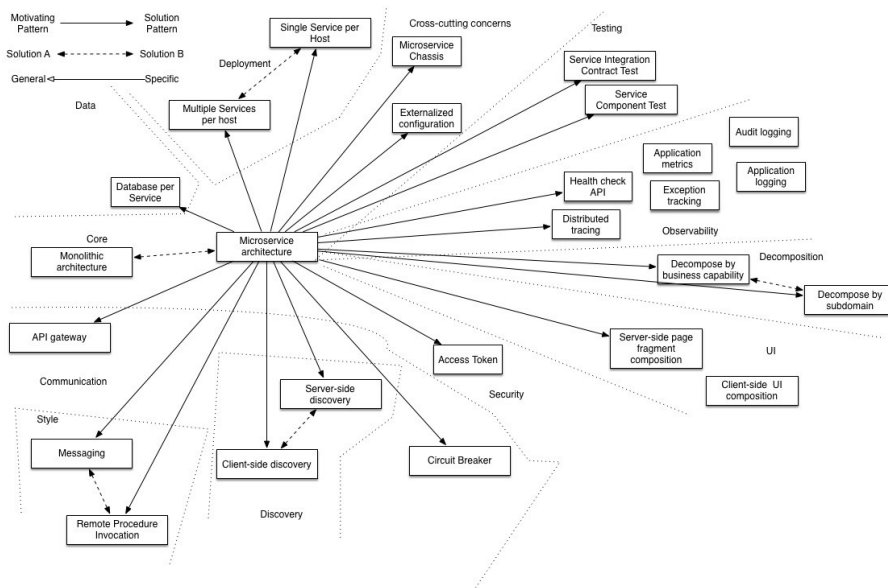


## 9 Arhitektuurilahenduste võrdlus

Mikroteenustel ja mikroteenuste arhitektuuril leidub erinevaid definitsioone ja erinevaid kasutusmustreid (*pattern*).

Sõltuvalt organisatsiooni tüübist, rahastamise skeemidest, meeskondade komplekteerimisest ja muudest faktoritest valitakse oma sobivamaid *pattern*-e.

Sõltumata sellest, milline neist osutus valituks, on käsitlemist vajavad teemad mainitud järgmisel joonisel (allikas <http://microservices.io/>).



## 9.1 Mikroarhitektuuri teemad

### 9.1.1 Turvalisus

#### 9.1.1.1 Autentimine

Riigiportaalides kasutatakse autentimiseks riiklikku autentimisteenust (TARA - <https://e-gov.github.io/TARA-Doku/>). Tehniliselt TARA kasutab OpenID Connecti (OIDC) protokollit ning pakub mitut autentimismeetodit, nagu ID-Kaart, MobileId, SmartId ja eIDAS.

Selleks et süsteemide vahel sujuvalt navigeerida, oleks mõistlik kasutada SSO lahendust. Kuna TARA juba kasutab OIDC protokollit, siis oleks loogiline, et SSO lahendus kasutaks ka OIDCd.

Esimeseks variandiks oleks TARA SSO (<https://e-gov.github.io/TARA-Doku/SSO%20tehniline%20spetsifikatsioon>). Kui võrrelda TARA SSOd klassikalise SSOga, siis tekib sellesse isikuandmete töötlemise volituse samm, mis kasutusmugavuse vaatenurgast ei ole hea.

Teine variant oleks võtta kasutusele eraldi komponent, näiteks Keycloak või Aparent, mis toetavad OIDC protokollit. Kui valida RIA-keskne SSO lahendus, siis palju keerulisem oleks

kaasata väliseid teenuseid. Iga välise teenuse kaasamisel tuleb SSO lahendust käsitleda eraldi iga üksikjuhtumi lõikes, seega eelistatud variant on ikka TARA SSO.

#### 9.1.1.2 Autoriseerimine

Pääsuõiguste eest vanas eesti.ee portaalis vastutas AAR-nimeline teenus, mis on plaanis kinni panna koos vana eesti.ee portaaliga.

EÜKPi jaoks on oluline informatsioon:

1. Nimekiri isikutest, keda saab esindada (firma A, firma B jne).
2. Mis õigused on kasutajal valitud isiku suhtes, näiteks info vaatamine või info muutmine. Täpsemad reeglid peavad selguma AAR2 detailanalüüsi käigus.

EÜKPi kontekstis on mõni variant autoriseerimise jaoks. Need on kirjeldatud eraldi peatükis [8. Pääsuõigused](#)

Integreerimine vana AAR-teenusega, mis oleks nagu lõpplahendus, ei ole mõistlik, kuna selle tegevus on lõppemas.

#### 9.1.2 Andmete haldus

Üks printsiipidest andmehalduse osas on üks-andmebaas-teenuse-kohta. Arvestades AS-IS arhitektuuri ja andmehoidlate omapära, siis selle printsiibi kohta saab öelda: üks andmehoidla teenuse kohta.

Andmehoidla võib olla nii SQLi baas kui ka GITi repositoorium.

#### 9.1.3 Suhtlus komponentide vahel

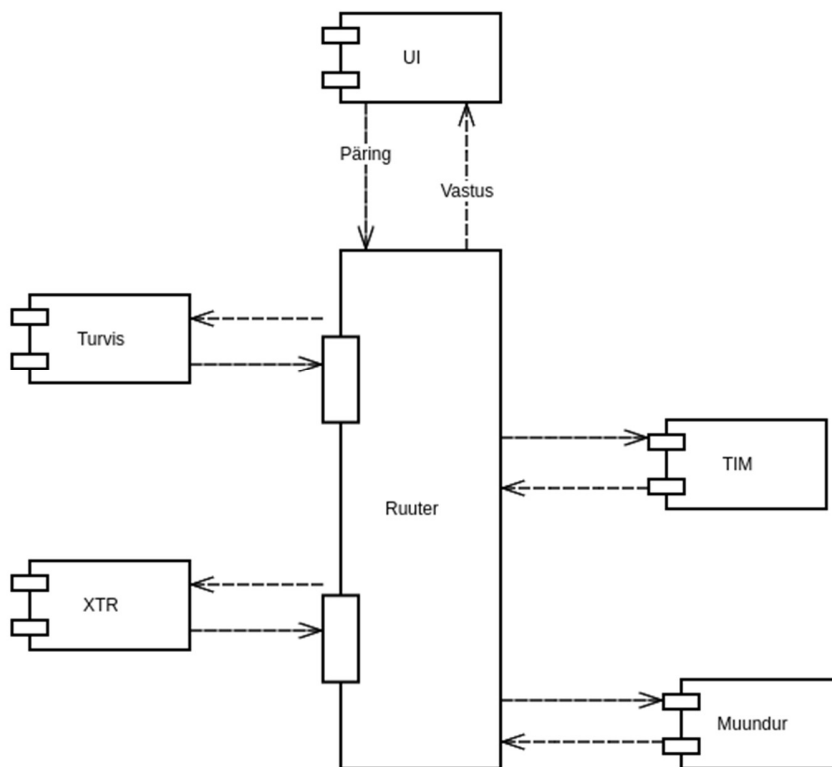
##### 9.1.3.1 Ruuter + andmemuundur + TIM + XTR

Kõiki päringuid teeb UI (kasutajaliides) **ruuteri** kaudu, mis võtab päringu vastu, töötleb vastavalt teenuse konfiguratsioonile ja annab vastuse tagasi.

Ruuter on kasutajaliidese jaoks n-ö sisenemispunkt (*entrypoint*). Ruuter, andmemuundur, TIM – võib võtta kasutusele koondnimetuse (API Gateway).

Iga komponent API Gateways täidab küll oma rolli, kuid arendaja jaoks see peab olema läbipaistev. Määrata API Gateway jaoks funktsioonid, näiteks:

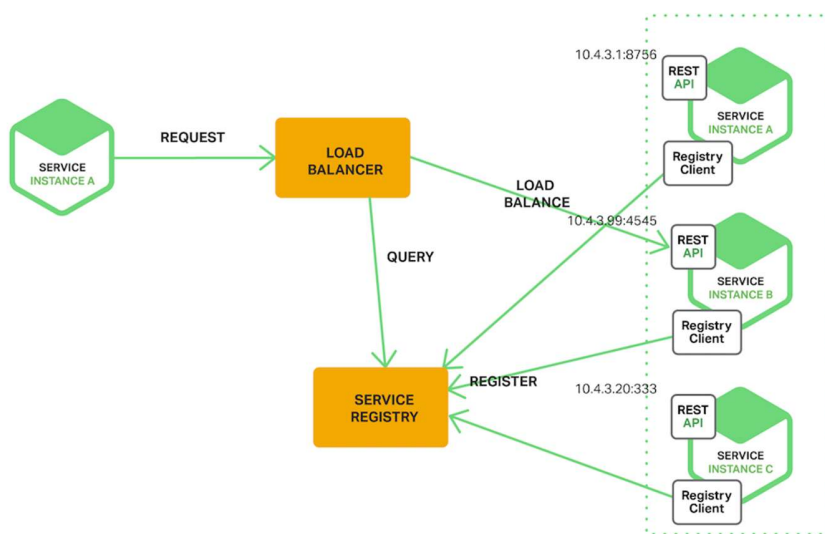
- teenuste ühise turvalisuse tagamine ehk päringute autentimine, sessioonide hoidmine (TIM);
- päringute logimine (ruuter);
- päringute koondamine (ruuter);
- päringute formaatide teisendamine (andmemuundur);
- *API Gateway pattern*'i kohta vt <https://microservices.io/patterns/apigateway.html>



#### 9.1.4 Service discovery

Teenusteregistri kasutuselevõtt (*Service discovery*) annab ülevaate komponentidest, mis on paigaldatud antud keskkonda ja kus (mis URLi või nime taga) need asuvad. Lihtsustab infrastruktuuri haldamist, näiteks automaatne koormusjaoturi konfigureerimine. Ühendused komponentide vahel peavad toimima komponendi nime järgi (mis peavad olema samad kõikides keskkondades), mitte URL- või IP järgi, mis erinevates keskkondades võivad olla erinevad. See ka lihtsustab komponentide/rakenduste haldamist. Enamasti kasutatakse kaht põhilist printsiipi: serveripoolne tuvastus ja kliendipoolne tuvastus. Keskkondades, kus suhtlus komponentide vahel peab olema keskselt kontrollitav, kasutatakse n-ö serveripoolset tuvastust (*server-side service discovery*).

##### 9.1.4.1 Ettepanek kasutada serveripoolset tuvastust (*Server-Side Service Discovery*)



## 9.2 Automaattestide kasutus

Automaattestide kasutus arenduse käigus on oluline, et minimeerida tulevikus probleemide tekkimist uue funktsionaalsuse lisamise puhul. Minimaalselt süsteemi vaja katta komponenttestidega (*unit* testid) ja integratsioonitestidega. RIA MFN #6.1 - #6.5 järgi automaattestid on nõutud kuid ei ole määratud nõutud kattuvus. Enamasti kattuvuse mõistlikkuse

piir jookseb 60-70% kogu koodist. Testidega kattuvust mõõdavad CI/CD vahendid, ehk iga kord kui rakendus ehitatakse kokku, käivitatakse ka automaatsete ja määratakse kattuvust.

Komponenttestide eesmärk on kontrollida, et süsteemi komponendid töötavad erinevates olukordades ootuspäraselt. Integratsioonitestide eesmärk on kontrollida, kas integreeritud komponentide omavaheline koostöö toimib (näiteks integratsioon Java rakenduse ja baasi vahel).

Nii komponenttsete, kui ka integratsioonitsete kirjutatakse süsteemi arendamise käigus arendaja poolt.

### 9.3 Süsteemide jälgitavus

Hetkeolukord on kirjeldatud AS-ISi peatükis [3.8 Logimisvajaduse analüüs ja logimislahenduse kirjeldus](#).

EÜKPi arhitektuuris ei ole otsest vajadust seda muuta. Hea oleks realiseerida parandusettepanekud.

### 9.4 Kasutajaliidesed

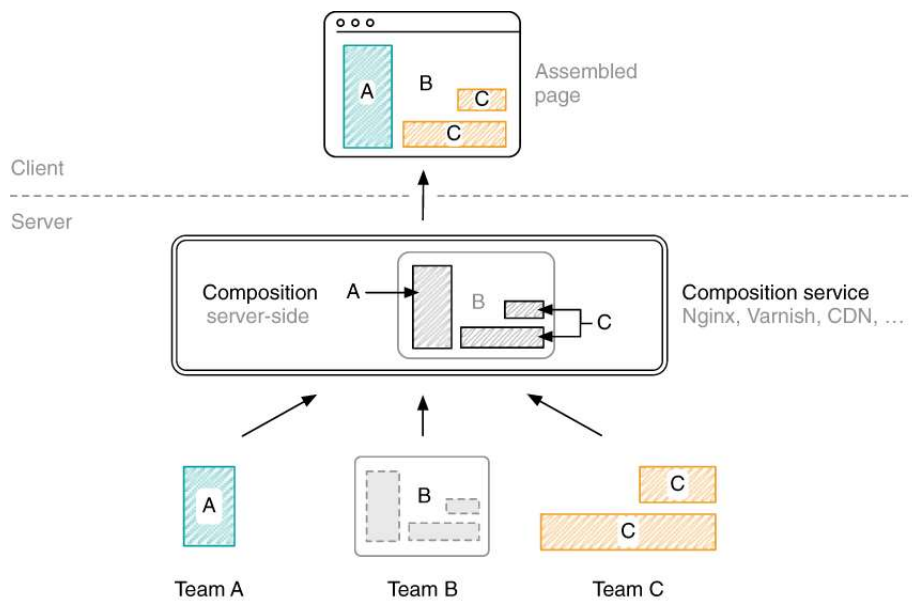
Hetkeolukord on kirjeldatud AS-ISi alamosas [3.6. Kasutajaliidesed](#).

Mikroteenuste puhul käsitletakse kasutajaliidese osas kaht erinevat printsiipi: serveripoolne UI kokkupanek (*server-side composition*) ja kliendipoolne UI kokkupanek (*client-side composition*).

#### 9.4.1 Serveripoolne UI kokkupanek

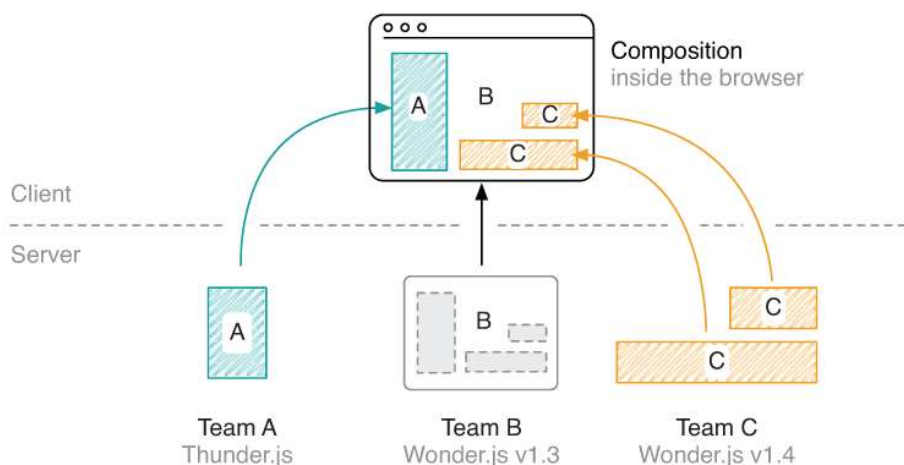
Server paneb kokku veebilehe ja selle struktuuri ning genereerib HTMLi *markup*'i. Tavaliselt vastutab selle eest eraldi komponent, mis asub kliendi ja reaalse teenuse vahel. Kasutatakse juhtudel, kui esmane laadimine on kliendi jaoks oluline.

Keskseks komponendiks võib olla ka näiteks Nginx, mis kasutab SSI (*Server Side Includes*) tehnoloogiat.



#### 9.4.2 Kliendipoolne UI kokkupanek

Veebileht, selle struktuur, ning HTMLi *markup* genereeritakse kliendi lehitsejas. Esmane laadimine ei ole oluline ja klient peab tükk aega ootama, kui kõik lehe elemendid jõuavad kohale. Erinevate rakenduste vahel navigeerides toimub ümbersuunamine ühelt rakenduselt teisele, mis põhjustab uuesti rakenduste laadimist, mis on klientidele märgavat.



Kui on vaja optimeerida lehed, et need oleksid indekseeritavad otsingumootorite poolt, siis serveripoolne UI kokkupanek on eelistatum variant. Angularis on see võimalik Angular Universal SSRi kaudu, ReactJSis saab sama asja teha Next.js serveri kaudu. Samuti enamikku JS raamistikke saab käivitada Node.JS Express serveri kaudu. Selline lähenemine ei välista ka seda plaani, et UI komponendid on publitseeritud NPM pakkidena.

## 9.5 Rakenduse paigaldamise variandid

Tänapäeval kasutavad erinevad asutused rakenduste paigaldamiseks järgmisi variante:

### 9.5.1 Eraldiseisvad virtuaalsed masinad (VM)

VMidele paigaldatakse rakenduse käivitamiseks vajalik tarkvara, näiteks serverid Java jaoks või PHP interpretaatorid PHP rakenduste jaoks. Eraldi paigaldatakse andmebaasimootorid jne.

VMide kõige suurem probleem on hallatavus. Pidevalt on vaja jälgida, et paigaldatud tarkvara oleks uuendatud, varukoopia tehtud ja jälgitav. Uuendamise käigus tekivad teenustes tihti katkestused.

Tänapäeval jookseb VMides enamjaolt n-ö Legacy süsteem, mis oli loodud ajal, kui tekkis konteineritehnoloogia. Teine põhjus VMide kasutamiseks on majasisene kompetents või siis kompetentsi puudus.

Selline lähenemine on jätkusuutlik nii kaua, kui süsteemi pole vaja oluliselt skaleerida. Tavaliselt süsteemi tootlikkus on loomise/muutmise käigus fikseeritud ning skaleerimiseks vajalik oluline muudatus on infrastruktuuris ja rakenduses.

***Täna on see kõige kindlam variant.***

### 9.5.2 Konteinertehnoloogia

Järgmine tase on konteinerid. Ühe VMi sees käivitatakse mitu erinevat konteinerit koos rakendustega, mis on loodud eri tehnoloogiatega. Oluline muudatus võrreldes VMidega on see, et konteiner peab sisaldama kogu vajaliku tarkvara rakenduse käivitamiseks, näiteks Java käivitamiseks on konteineri sees juba olemas kindla versiooni Java Virtual Machine. Tõstes konteinerit ühest kohast teise või tekitades teist koopiat on kindel, et rakendus töötab edasi samamoodi.

Konteinerite jooksutamiseks kasutatakse enamjaolt Dockerit. Konteinerid ilma orkestreerimise võimaluseta tekitavad samu probleeme, nagu eelmises punktis. Nende haldamine on keeruline, ülevaade, milline konteiner millises masinas jookseb, vajab lisapingutust.

### 9.5.3 Konteinerite orkestreerimine

Tänapäeval on populaarsemaks lahenduseks konteinerite orkestreerimisel Docker Swarm ja Kubernetes. Orkestraator ise hoolitseb selle eest, et piisavalt konteinereid oleks elus ja mis VMides need käivad. Docker Swarmi või Kubernetesi paigaldamine vajab põhjalikku planeerimist ja oskust, kuid see oluliselt lihtsustab rakenduste haldamist. Kubernetesi jaoks on olemas palju erinevaid tehnoloogiaid, mis aitavad administraatoritel enamus rutiine automatiseerida. Näiteks TLS sertifikaatide eest hoolitseb Cert Manager, rakenduste skaleerimise eest Autoscaler, võrgu turvalisuse eest Envoy või Istio.

Kuna klasteri paigaldamine on keeruline ja paljude detailidega, siis enamikul juhtudel oleks otstarbekam kasutada pilveteenuse pakkujaid, kes pakuvad näiteks Kubernetesi teenust.

***Kubernetes on see suund, kuhu RIA tahab liikuda.***

### 9.5.4 Pilveteenused

Oma infrastruktuuri saab majutada nii enda serveriruumides kui ka [pilveteenuseid](#) kasutades.

## 9.6 Seadistatavuse ja konfigureeritavuse analüüs

Antud analüüs käsitleb sündmusteenuste seadistamist ja konfigureerimist erinevatest komponentidest.

Kui sündmusteenus on keskselt hallatav RIA poolt, siis järgmine tabel kirjeldab, milliseid komponente ja kuidas hallatakse:

Komponent	Kirjeldus
EÜKP-i veebipõhine klient	PoCi raames peame välja selgitama, kuidas see konfiguratsioon välja näeb.
Ruuter	Konfiguratsioon on eraldi rakendusest. Kogu juhtimine peab olema seadistatud JSONi failide abil. JSONi formaat ei ole kergesti loetav, hea oleks konfiguratsioon üle viia YAMLi formaadi peale.  Nii kaua, kui ei ole Sandboxi keskkonda, kus iga arendaja saab katsetada oma konfiguratsiooni, tegelevad sellega RIA spetsialistid.
Andmemuundur	Konfiguratsioon on eraldi rakendusest.  Nii kaua, kui ei ole Sandboxi keskkonda, kus iga arendaja saab katsetada oma konfiguratsiooni, tegelevad sellega RIA spetsialistid.
XTR	Uute teenuste jaoks on vaja genereerida uus JAR uue WSDLi põhjal. Sellega tegelevad RIA spetsialistid.
TIM	Vajalikud ainult infrastruktuursed seadistused. Praegu ei ole teada, et oleks vaja midagi eraldi seadistada sündmusteenuste jaoks.
Eesti.ee CMS	Kasutajaliidese kaudu on võimalik seadistada lehtede sisu ja struktuur.

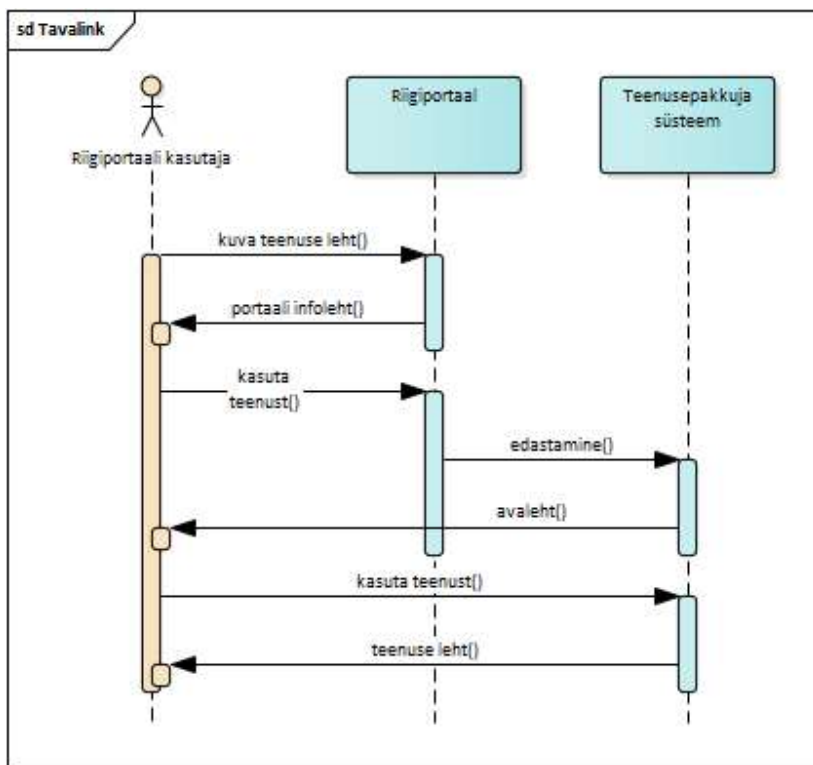
Kui tegemist on sündmusteenusega, mida asutus ise osutab ja haldab, siis RIA vajab seda, et asutuse leht integreeritakse riigiportaali, kuid teenuse konfigureerimine jääb asutusele ja selle asutuse arenduspartneritele.

## 10 Liidestamise variantide analüüs

### 10.1 Lihtne link

#### 10.1.1 Lahenduse kirjeldus

Tavaline veebilink, mis viib teisele lehele, mis ei ole otseselt teenuse leht. Tavalingi kasutamisel ei ole vaja kaasa anda teenuse osutamise seotud andmeid.



Tavalingi kasutamisel riigiportaal ei vastuta teenuse toimimise eest. Kõik vajalik kontroll tuleb teha teenusepakkuja süsteemi poolel.

### 10.1.2 Arendusvajadused

See on kõige lihtsam variant. Portaali ega teenuse osutajal pole vaja midagi juurde arendada.

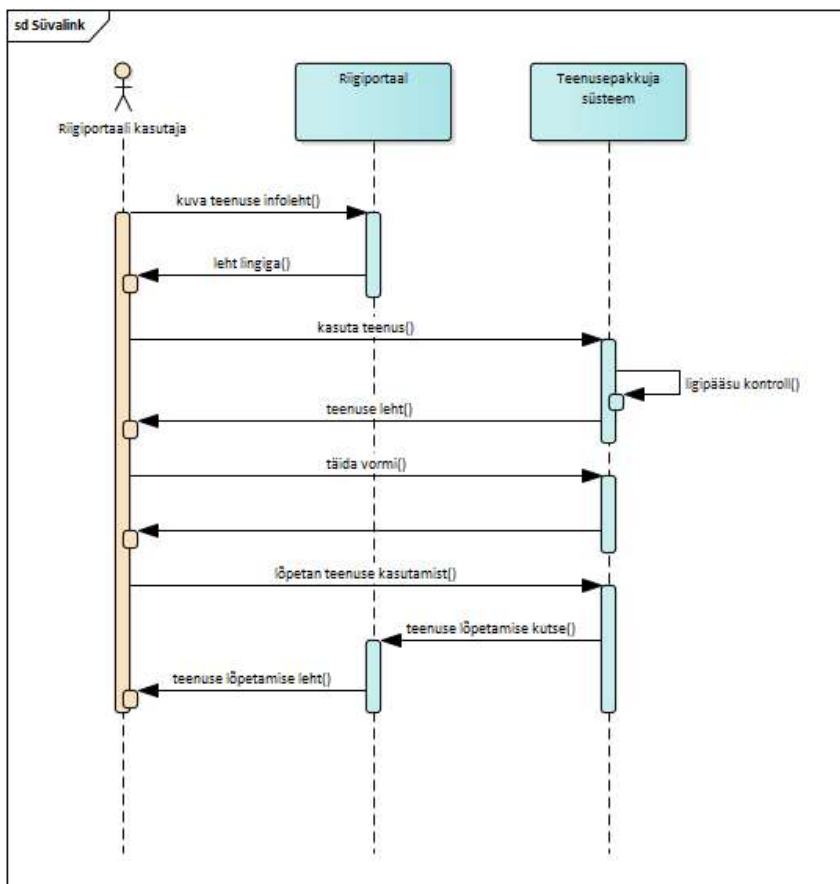
### 10.1.3 Halduslahendus

Teenust saab lisada portaali CMSi süsteemi kaudu.

## 10.2 Süvalink

### 10.2.1 Lahenduse kirjeldus

Link, mis viib otse teenuse juurde asutuse portaalis. Süvalink võib edastada osa andmetest ja süvalingi kasutamisel võib (aga ei ole kohustuslik) kasutaja suunata pärast teenuse kasutamist tagasi riigiportaali.



### 10.2.2 Arendusvajadused

Antud variandi puhul peab teenusepakkujal olema link/teenus, mis:

- kontrollib ligipääsu või vajadusel autendib kasutaja.
- Ideaalis link/teenus võiks toetada tagasisuunamise (*callback*) funktsionaalsust.
- Kasutusmugavuse tõstmiseks peab teenusepakkuja süsteem olema liidestatud TARA SSOga, et kasutaja ei peaks ennast mitu korda autentima.

### 10.2.3 Halduslahendus

Süvalingiga osutatud teenust saab lisada portaali CMSi süsteemi kaudu.

Teenuse tagastamislingi (*callback*) jaoks võib pakkuda kahte lahendust:

1. Tagastamislink genereeritakse automaatselt, näiteks lisades sellele tagastamistunnuse.
2. Tagastamislink lisatakse käsitsi ja tagastamiseks kasutatakse kasutaja poolt defineeritud lehte (jätkusuutlikum ja paindlikum lahendus).

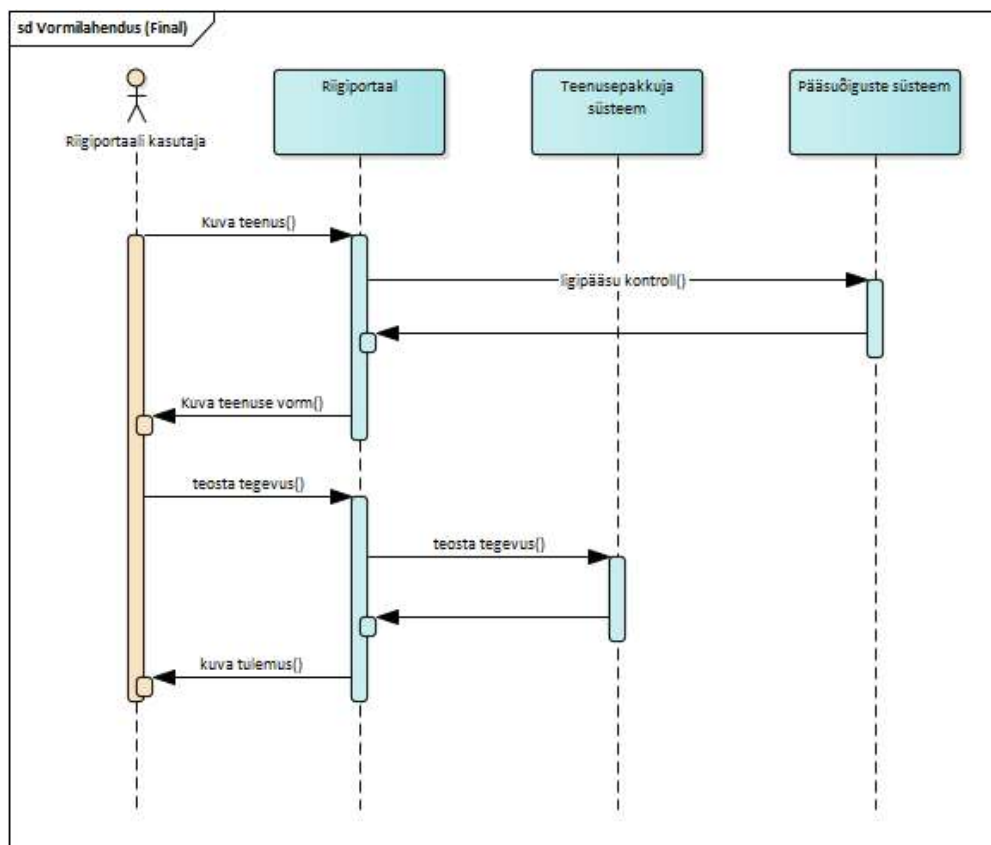
Mõlemal juhul kasutaja lisab riigiportaali lehele süvalingi objekti ja konfigureerib lingi sihtaadressi ja vajadusel ka tagastamislingi.

PoCi raames on välja arendatud teine variant.

## 10.3 Vormihaldusvahendiga kirjeldatud vormi kasutamine

### 10.3.1 Lahenduse kirjeldus

Välises vormihalduse süsteemis kirjeldatud vormi kasutamine riigiportaalis.



PoCi raames valminud lahendus võimaldab lisada vormikirjeldust otse portaali lehele või konfigureerida vormi käsitsi.

Vormi teostamine toimub vormis kirjeldatud aadressil, eeldatavalt teenusepakkuja süsteemis. Vormi loogika on sisse ehitatud vormi kirjeldusse ja riigiportaal tegeleb ainult kirjelduse töötlemisega.

### 10.3.2 Arendusvajadused

PoCi raames on portaalil välja arendatud UI komponendid, mis joonistavad konfiguratsiooni alusel vastava vormi.

PoCi raames on kasutusele võetud <https://formly.dev> vormikirjelduse formaat, mis hoiab andmeid JSONi kujul.

Edasiarenduste käigus võib täiendada PoCi raames valminud lahendust stiilide ja täiendavate väljatüüpide töötlemisega.

Selline variant tundub olevat sarnane praeguse Orbeon Formsi lahendusega, mis tänapäeval tekitab suurt halduskoormust RIAle. On otsustatud, et Orbeon Forms lahendusi ei arendata ümber ja see jääb lähitulevikus kasutusele. Mõlema vormilahenduse edasiarendamine ei ole kulude kokkuhoiu vaatepunktist mõistlik. Soovitus on edasi arendada ainult ühte lahendust.

PoC projekti raames valminud lahendus on vabavaraline, piisavalt küps ja laia kasutusega. Orbeon Forms lahendus on litsentseeritud lahendus, millega kaasneb iga-aastane litsentsitasu.

### 10.3.3 Halduslahendus

Vormide konfigureerimine toimub kas eraldiseisvas haldussüsteemis, riigiportaal kasutab haldussüsteemis loodud vormi kirjeldust, mis hõlmab vormi väljade ja ülesehituse kirjeldust, andmemudeli ja tegevuste kirjeldust.

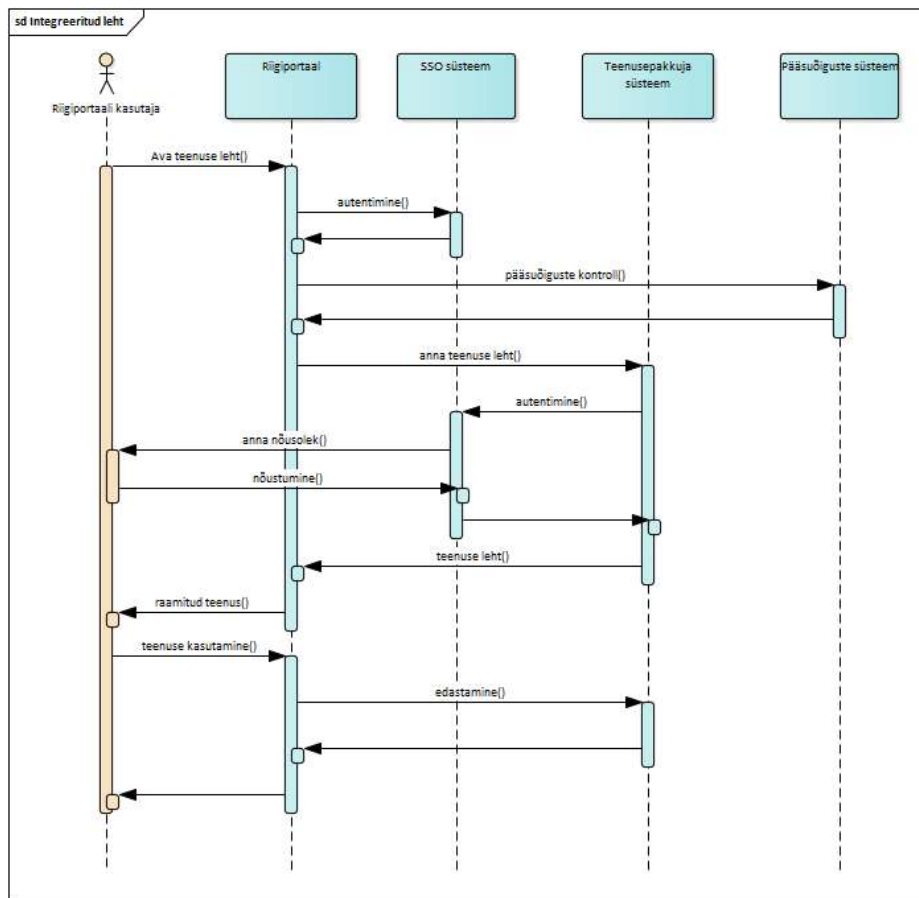
Samuti saab vormi konfigureerida otse riigiportaalis, kasutades selleks PoCi raames valminud malli.

Teenuse lisamiseks CMS süsteemis tuleb lisada leht vastava malli alusel ja konfigureerida vormi.

## 10.4 Domeeni jagamine - asutuse lehe integreerimine riigiportaali

### 10.4.1 Lahenduse kirjeldus

Teenusepakkuja serveris asuvate lehtede kuvamine riigiportaali osana. Kasutaja suunatakse teenusepakkuja süsteemi milles kasutatakse riigiportaali päis, jalus ja menüü.



Riigiportaali integreeritud lahenduse puhul toimub teenuse osutamine, andme- ja ärioloogika töötlus teenusepakkuja süsteemis.

Pääsuõiguste kontrollimine võib toimuda nii riigiportaalis kui ka teenusepakkuja süsteemis, või see võib olla jagatud, nt pääsuõigused kontrollitakse teenusepakkuja süsteemis ja autentimine toimub portaalis.

Teenust osutab teenusepakkuja süsteem, riigiportaal vahendab teenusepakkuja ja kasutaja suhtlust, edastab teenusepakkuja ja kasutaja tegevusi ja kuvab teenusepakkuja süsteemis asuvad vormid ja muud lehed.

## 10.4.2 Arendusvajadused

### **Portaali poolt:**

Peavad olema arendatud ühised UI komponendid, nagu menüüd, päis, jalus ja publitseeritud avalikult kättesaadavates repositooriumites.

RIAI on teenusepakkujaga sama domeen ja sessiooni küpsised (*cookie*).

Menüüde konfiguratsioon on publitseeritud avalikult ja teenusepakkujate süsteemid saavad seda konfiguratsiooni kasutada.

Teenusepakkuja ja portaali on ühendatud ühise SSO lahendusega.

### **Teenusepakkuja poolt:**

Arendatakse kogu teenus valmis, kasutades ühiseid UI komponente ja CDNi kaudu publitseeritud portaali menüüd.

## 10.4.3 Halduslahendus

Teenuse konfigureerimine toimub eraldiseisvas lahenduses, mis salvestab masinloetava ja portaali poolt kasutatava konfiguratsiooni faili, või siis toimub see otse arenduskeskkonnas.

Valmislahendust võib kasutada CMSi süsteemis täislehena. Teise teenusepakkuja lehte ei saa kasutada eraldi lehena portaali struktuuris, selleks tuleb defineerida portaali leht ja sellele lisada vastav komponent, lisades sellele teenuse konfiguratsiooni.

## 10.5 Microfrontend lahendused

### 10.5.1 Lahenduse kirjeldus

Lahendus kasutab valmiskomponente, mis on arendatud kas riigiportaali halduri või teenusepakkuja poolt. Lahenduse eeliseks on võimalus kasutada samu komponente, mis on



Seda varianti ei ole soovitatav kasutada, kui ärivajaduste katmiseks piisab vormihalduse võimalustest (vt [10.3. Vormihaldusvahendiga kirjeldatud vormi kasutamine](#)).

### 10.5.3 Halduslahendus

Teenuse konfigureerimine toimub eraldiseisvas lahenduses, mis salvestab masinloetava ja portaali poolt kasutatava konfiguratsiooni faili, või toimub see otse arenduskeskkonnas.

*Microfrontend* lahendust võib kasutada:

- eraldi lehena,
- lehe osana.

Nii eraldi lehena kui ka lehe osana kasutamisel CMSi süsteemis tuleb lisada vastav leht (PoCi raames on valminud CMSi süsteemi mall) ja panna paika teenuse konfiguratsioon.

## 11 Masinõppe kasutamise analüüs

Masinõppe on viimasel ajal näidanud märkimisväärset kasvu ja arengut. Tänapäeval on see võimas tööriist kasutajakogemuse parendamiseks ning nii ajalise kui ka rahaliste kulude kokkuhoiu saavutamiseks. Masinõppe all mõeldakse nii kitsa tähendusega kohandatavaid süsteeme kui ka laia tähendusega tehisintellekti süsteeme. Käesoleva analüüsi raames masinõppe lahenduste all mõeldakse tehisintellekti lahendusi.

### 11.1 Tehisintellekti määratlus

Tehisintellekti süsteemid on inimeste projekteeritud tarkvaralised ja võimalik et ka riistvaralised süsteemid, millele antakse keerukas eesmärk ja mis toimivad füüsilises või digitaalses mõõtmes, tajudes oma keskkonda andmehõive abil, tõlgendades kogutud struktureeritud ja struktureerimata andmeid, tehes teadmuse põhjal järeldusi või töödeldes kõnealustest andmetest saadud teavet ja otsustades, millised on antud eesmärgi saavutamiseks parimad toimingud. Tehisintellekti süsteemid võivad kasutada kas sümbolreegleid või õppida selgeks arvmudeli ning samuti saavad

nad kohandada oma käitumist, analüüsisides seda, kuidas nende varasemad toimingud mõjutavad keskkonda.

Teadusvaldkonnana hõlmab tehisintellekt mitut lähenemisviisi ja meetodit, näiteks masinõpet (mille konkreetseteks näideteks on süvaõpe ja stiimulõpe), masinloogikat (mis hõlmab kavandamist, plaanimist, teadmuse esitamist ja järelduste tegemist, otsimist ja optimeerimist) ning robotikat (mis hõlmab juhtimist, tajumist, andureid ja täiturseadmeid ning ka kõigi muude meetodite lõimimist küberfüüsikalistesse süsteemidesse).

#### 11.1.1 Tehisintellekti lahenduse tunnused

1. Iseõppiv või vähese inimsekkumisega õppiv lahendus – süsteem kohandab oma käitumist uute andmete ja juhtumite teadasaamisega nii, et paremini vastata tuleviku olukorrale.
2. Otsustamisvõimega – rakendus oskab iseseisvalt leida lahenduse mitte ainult ettekirjutatud situatsioonides, vaid ka sarnastes ja mittesarnastes situatsioonides, samas otsused on enamasti õiged.
3. Andmepõhine – lahendus kasutab otsuste langetamiseks kõiki kättesaadavaid andmeid. Andmed võivad olla nii aluskujul kui ka töödeldud kujul.

#### 11.2 Masinõppe kasutamise võimalused

Ettevõtjate jaoks ühtse kontaktpunkti loomise visiooni dokumendis välja toodud masinõppe kasutamise võimalused:

- teenuste proaktiivne pakkumine ettevõtte profiili arvestades ehk avalik sektor kui partner ettevõtja võimalike ärisündmuste edukaks ellukutsumiseks;
- soovitusüsteem tegevusvaldkonna valimiseks või laiendamiseks ja kohese ülevaate andmine sellega kaasnevatest kohustustest ja võimalustest (sh teiste sarnase valiku teinud ettevõtjate arengust);
- esitatud aruannete põhjal tegevusvaldkonna muutmise soovitamine, mis omakorda toetab kontaktpunkti eesmärki pakkuda infot ja teenuseid vastavalt ettevõtte tegelikule tegevusalale;

- kohustuste täitmise meeldetuletamine, kus vastavalt ettevõtja profiilile näidatakse talle antud ajahetkel kohalduvaid ja olulisi kohustusi. Seejuures meeldetuletuste põhimõtted ja ajakava tulenevad ettevõtjate varasematest käitumismustritest;
- ettevõtte teavitamine riigihangete avaldamisest juhul, kui ettevõtte on varem sarnastel hangetel osalenud või hanke pakkumuses esitatud andmed võiksid sobida ka uuel hankel osalemiseks. See tõstaks ka hangetel osalejate omavahelist konkurentsi, mis hankija jaoks tähendab optimaalsemat pakutava hinna ja teenuse/toote kvaliteedi suhet;
- pöördumistest tuleneva asutuste koormuse vähendamiseks pakub esmase pöördumislahenduse enimlevinud probleemidele/küsimustele tehisintellektlahendus. Seejuures on oluline, et pakutud vastuste tõesus ja usaldusväärsus peab olema selline, mis võimaldab ettevõtjal seda edasistes otsustes kasutada võrdväärset asutuse töötaja poolt antavate vastustega;
- avalike asutustega suhtluse analüüsimine, et tuvastada teenuste või suhtlusega rahulolematust või ootustele mittevastavat kasutajakogemust. Nende mitteisikustatud andmete analüüsitulemuste põhjal on võimalik teha järeldusi tagasiside küsimiseks valitud teenuste või asutuste osas, et saada infot parendamise jaoks vajalikeks juhtimisotsusteks;
- menetluste üldine lihtsustamine, kus rohkem tähelepanu nõudvate probleemsete juhtumite valimiga tegeleb masinõppe algoritm. Valimivälistele juhtumitele on aga võimalik välja töötada lihtsustatud menetlusprotsess, mis vähendab nii asutuste kui ettevõtjate halduskoormust.

#### 11.2.1 Masinõppe/tehisintellekti kasutamise ülevaade

Tehisintellekti, kitsamas mõttes masinõpet on võimalik mitmekülgsest rakendada, seda on võimalik efektiivselt kasutada riigiportaali kaudu teenuste osutamise hõlbustamiseks. All on toodud masinõppe kasutusjuhud, mis on saavutatavad olemasolevate taaskasutatavate komponentide baasil ja mis toovad reaalselt kasu lõppkasutajatele.

## **Otsing**

Otsingu kvaliteedi tõstmiseks võib kasutada erinevaid meetmeid. Esiteks võib lisada häälotsingu ja/või isikliku assistendi, otsingu täpsustamiseks saab kasutada ka juturobotit. Teiseks, otsingu tulemusi aitab täpsustada otsinguobjektide automaatne kategoriseerimine ja sobitamine kasutaja profiili andmetega. Tulemuste saavutamiseks on vaja rakendada kategoriseerimise algoritme, keeletuvastuse ja keeletötluse (transkribeerimine ja tekstitöötlus) algoritme.

## **Teenuste pakkumine**

Proaktiivne teenuste pakkumine võib põhineda kategoriseerimise algoritmil, mis võtab aluseks kasutaja ja tema valdkonna andmeid. Teenuste pakkumine võib toimuda soovitusüsteemi abil. Soovitusüsteem on eraldiseisev moodul või riigiportaali osa, mis kasutab masinõppe algoritme sobivate objektide leidmiseks.

Teenuste soovitamine või seotud informatsiooni pakkumine võib toimuda nii otsingu tulemuste juures kui ka teistes riigiportaali osades. Üks variantidest on lehtede ja töölaudade osaline muutmine vastavalt kasutaja andmetele ja tema tegevustele: lehe alamosi võib dünaamiliselt muuta selliselt, et kasutaja jaoks saadav info katab konkreetse olukorra vajadusi. Tehniliselt saab kasutada soovitusüsteemi, mis baseerub kasutaja andmete, tegevuste ja teenuste klassifitseerimisel või kategoriseerimisel.

## **Kättesaadavus**

Teenuste kättesaadavuse tõstmiseks võib rakendada automaattõlget, mis tagab selle, et teenust saab osutada võõrkeelsetele inimestele. Infoartiklid ja taotluse vormid tõlgitakse tavaliselt inglise ja vene keelde, aga automaattõlge teeb teenuse kättesaadavaks ka teistes keeltes suhtlevatele inimestele, samuti tehakse ka ametlikud vastused (mis on tavaliselt ainult eesti keeles) arusaadavaks teistes keeltes suhtlevatele inimestele.

Tagasiside töötlemise lihtsustamiseks võib kasutada kliendipöördumiste klassifitseerimise süsteemi. Lisaks kategoriseerimisele, mis lihtsustab pöördumise suunamist, võib rakendada ka

anomaaliate tuvastamist halduri teavitamisega. Samas tagasiside või pöördumine võib olla teostatud juturoboti abil, sh hääljuhtimisega.

### 11.3 Tehnoloogilised aspektid

Riigiportaali kontekstis on soovituslik kasutada juhendamata või vahelise juhendamisega algoritme arenduskulude vähendamiseks. Juhendamata õppe (*unsupervised learning*) puhul sisendandmed ei ole tulemi osas märgistatud ja mudel peab leidma andmetest sarnaseid struktuure või looma üldistamise reegleid. Vahelise juhendamisega õppe (*semi-supervised learning*) puhul sisendandmed on segu märgistatud ja märgistamata näidetest ning mudel peab andmetest leidma nii sarnaseid struktuure kui ka tegema ennustusi. Vahelise juhendamisega õpet võib kasutada klientide tagasiside algoritmi iseõppimiseks.

Iga juhtumi jaoks parima algoritmi valik peab toimuma arendusprojekti käigus mudeli hindamise alusel. Mudelite hindamiseks tuleb defineerida mõõdikud iga juhtumi jaoks ja kontrollida ennustusjõu koefitsienti – õigete tulemuste arvu protsent kogu juhtumite arvust. Lisaks on vaja jälgida ka eksimuste maatriksit – valepositiivsete ja valenegatiivsete tulemuste arvu. Sobivad mudelid peavad maksimeerima ennustuskoeffitsienti (õigepositivsete osakaalu) ja minimeerima eksimuste arvu (valepositiivsed ja valenegatiivsed tulemused).

#### 11.3.1 Andmete kasutamine

Masinõpe tegutseb andmete põhjal. Mida suurem ja täiuslikum on andmete kogum, seda paremat tulemust on võimalik saavutada.

Ettevõtjate jaoks ühtse kontaktpunkti loomise visiooni dokumendis välja toodud eeldused:

- analüüsitavaid, ühtlustatud ja masinloetavaid andmeid peab olema võimalikult palju;
- eelduseks on kvaliteetsed ja asjakohased andmed;
- lisaks korrastatud äriprotsessidega seotud andmetele kogume ka kasutajate käitumismustrite, tegevusajaloo infot, mis kirjeldavad ettevõtja vajadusi ja valikuid. Selline info on sisendiks järgmiste tegevuste ja info pakkumiseks;

- oluline on erinevate asutuste kogutavate andmete kooskasutuse juurutamine.

Selleks, et kontaktpunkt saaks kasutada masinõppe eeliseid, on vaja lahendada järgmised andmekorje probleemid:

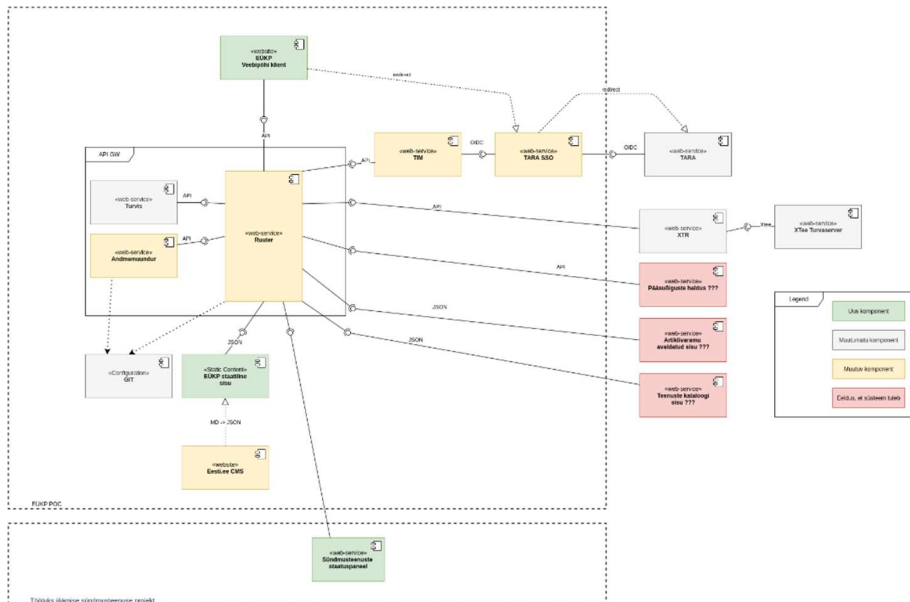
1. Andmed on töödeldud ja salvestatud erinevates süsteemides, kontaktpunkt ei oma ligipääsu sellistele andmetele. Masinõppe algoritmid peavad omama ligipääsu või korjama selliseid andmeid eraldiseisvas süsteemis.
2. Andmete ühiseks töötlemiseks (sõltumata andmetest) peab olema tagatud juriidiline põhjendus, mis paneb paika reeglistiku ja võimaliku kasutuse ulatuse.

### 11.3.2 Masinõppe juurutamine

Masinõppe juurutamiseks on vaja tagada rida nõudeid.

1. Peab salvestama anonümiseeritud või isikustatud (siin „isikustatud“ tähendab kasutajaga ühendataval moel salvestatud andmed, kasutaja identifitseerimistunnused, nagu isikukood, nimi jne võivad olla teadmata) käitumisandmed.
2. Masinõppe süsteemil peab olema ligipääs avalikele andmetele, nt äriregistri andmed, avalikustatud aastaaruanded.
3. Masinõppe süsteem peab omama ligipääsu teiste portaalide ja teenuste kasutajate koond- ja käitumisandmetele.
4. Andmed peavad olema hästi klassifitseeritud, kasutama keskselt hallatud kontrollitud sõnastikke, nt tegevusvaldkondade klassifikaator.
5. Andmed peavad olema masinloetaval kujul kirjeldatud metaandmete süsteemi abil, mis muuhulgas tagab ka sõnastikkude kasutamise kirjeldamist. Abiks võib olla kavandatav RIHAKE alamsüsteem.

## 12 Kontaktpunkti ja sündmusteenuste arhitektuuri projektide vastutuse jaotus



EÜKPi arhitektuuri analüüsi projekt tegeleb osateenuste ja/või sündmusteenuste liidestamise lahendustega riigiportaali jaoks. Projekti mõjualasse kuulub halduslahenduse pakkumine osateenuste haldamiseks riigiportaalis.

EÜKPi arhitektuuri analüüsi projekti vastutusala:

1. Osateenuste liidestamise arhitektuur.
2. Osateenuste liidestamise tehnilised näited ja juhendid.
3. Tehnilised nõuded osateenuseid osutavatele süsteemidele.
4. Arendussoovid teenuste kataloogi, artiklivaramu ja muude süsteemide suhtes liidestamise lahenduse toimimiseks.

5. Sündmusteenuse lehe vormistamise variantide ettepaneku tegemine.

Kuna töötuks jäämise sündmusteenuse projekti üheks eesmärgiks on sündmusteenuse oleku jälgimine ja olekust teavitamine, projekt esitab nõudeid (nt osateenuste oleku raporteerimise suhtes) ja pakub lahendust sündmusteenuse oleku jälgimise ja oleku kuvamise komponentide jaoks. Eeldame, et komponendid on kasutatavad riigiportaali CMSi lahenduses.

Töötuks jäämise sündmusteenuse projekti vastutusala:

1. Sündmusteenuse oleku jälgimise komponendid teenuse osutatavate süsteemide jaoks ja/või sellele esitatavad nõuded. Arvestada juriidiliste isikute sündmusteenustega.
2. Sündmusteenuse oleku salvestamise lahendus ja/või sellele esitatavad nõuded. Arvestada juriidiliste isikute sündmusteenustega.
3. Elusündmuste jaoks vajalikud liidestamised, mis ei kuulu EÜKPi projekti vastutusalasse.
4. Arhitektuuriliselt peavad jälgimise komponendid olema ühildatavad EÜKPi liidestamise ja halduse arhitektuuriga (nt jälgimise komponenti peab saama riigiportaali CMSi kaudu kasutada)
5. Arendussoovid teenuste kataloogi, artiklivaramu ja muude süsteemide suhtes, et sündmusteenuste oleku jälgimise lahendus toimiks.

# 13 Riskide analüüs ja maandamise meetodid

## 13.1 Äririskid

Halduse ja sündmusteenuse pakkumise riskid

	<b>Risk</b>	<b>Kirjeldus</b>	<b>Riski tõenäosus</b>	<b>Mõjuhinnang</b>	<b>Maandamise meetodid</b>
1	Osateenuse/sündmusteenuse osutamise muutmisest või lõpetamisest ei teavitata	Risk, et teenus ei ole enam kättesaadav ja risk, et teenus on tehniliselt kättesaadav (IT lahenduse mõttes), kuid äriliselt ei ole osutatav ja pöördumised jäävad vastamata.	Madal	Keskmine	<ul style="list-style-type: none"><li>• Jälgida tehniliselt teenuse kättesaadavust (nt kasutada zabbix jms süsteeme)</li></ul>
2	IT lahendust asendatakse/muudetakse	Osateenuse või sündmusteenuse osutaja muudab kontseptuaalselt IT lahenduse ülesehitust - kasutab uut teeki, lähenemist või arhitektuuri	Madal	Keskmine	<ul style="list-style-type: none"><li>• Jälgida tehniliselt teenuse versiooni või kasutada kindlat versiooni</li><li>• Jälgida tehnoloogiate kasutamist teenuste arenduste käigus, täiendada mittefunktsionaalsete nõuete nimekirja</li><li>• Kasutada mikroteenuste arhitektuuri</li></ul>

	<b>Risk</b>	<b>Kirjeldus</b>	<b>Riski tõenäosus</b>	<b>Mõjuhinnang</b>	<b>Maandamise meetodid</b>
3	Sündmusteenuse osutamise protsessis osalevad süsteemid ei ole stabiilsed	Integreeritud teenuse või riigiportaali loodud teenuse puhul on risk, et teenuse osutaja süsteemi vead tekitavad probleeme portaali jaoks.	Keskmine	Keskmine	<ul style="list-style-type: none"> <li>Jälgida tehniliselt teenuse kättesaadavust(nt kasutada zabbix jms süsteeme)</li> <li>Teenuse muutmine probleemsete süsteemide välistamiseks, näiteks õigusaktide, vajalike dokumentide muutmine, töökorralduse muutmine jne.</li> </ul>
4	Sündmusteenuse osutamise IT süsteemide koostoime ei ole saavutatud	Teenuse osutamise protsessis osalevad süsteemid ei ole omavahel kooskõlas, nt. Artiklivaramu ei edasta tekstide uuendusi, Teenuste kataloog jagab valesid identifikaatoreid või ei jaga neid üldse jne	Keskmine	Keskmine	<ul style="list-style-type: none"> <li>Süsteemide arendusprojektide koordineerimine</li> <li>Sündmusteenuste haldamise protsessi jälgimine</li> <li>Alternatiivide analüüs teenuste muutmisel ja arendamisel: analüüsi etapil uurida erinevaid teenuse osutamise variante ja valida kõige töökindlam variant</li> </ul>

	<b>Risk</b>	<b>Kirjeldus</b>	<b>Riski tõenäosus</b>	<b>Mõjuhinnang</b>	<b>Maandamise meetodid</b>
5	Portaalis implementeeritud sündmusteenuse koosseisu muudetakse	Portaal pakub vananenud lahendust, mis ei ole kooskõlas sündmusteenuse portfelli halduri poolt koostatud kirjeldusega.	Madal	Madal	<ul style="list-style-type: none"> <li>• Sündmusteenuste väljalülitamise võimaluse tekitamine</li> <li>• Sündmusteenuse koosseisu jälgimise komponendi implementeerimine</li> <li>• Sündmusteenuste haldamise protsessi jälgimine</li> </ul>
6	Teenuse osutamine liigutakse teise haldusalasse	Teenuse osutamine viiakse üle uude haldusalasse ja uus teenusepakkuja ei ole valmis seda teenust pakkuma endise lahendusega.	Madal	Madal	<ul style="list-style-type: none"> <li>• Jälgida tehniliselt teenuse kättesaadavust (nt kasutada zabbix jms süsteeme)</li> </ul>

	Risk	Kirjeldus	Riski tõenäosus	Mõjuhinnang	Maandamise meetodid
7	Sündmusteenuse arendamisel ei arvestata teenuse majutaja soovidega	Sündmusteenuse lahendus ei sobi riigiportaalis kuvamiseks või tekitab haldus- ja tehnoloogilisi raskusi.	Keskmine	Keskmine	<ul style="list-style-type: none"> <li>• Kasutada domeeni jagamise lahendust;</li> <li>• Jälgida tehnoloogiate kasutamist</li> <li>• Kasutada koodi ja tehnoloogiliste lahenduste tehnilisi automaatkontrolle: <ol style="list-style-type: none"> <li>1.SonarQube;</li> <li>2.BitBucketi (ja selle alternatiivide) automaatika CVE nimekirjade vastu kasutatavate seoste (<i>dependency</i>) kontrollimiseks;</li> <li>3."npm audit" käsu kasutamine, kus sobilik; jne</li> </ol> </li> </ul>
8	Sündmusteenused ei arvestata pääsuõigustega	Riigiportaali kaudu pakutud sündmusteenused ei kasuta või ei kasuta õigesti pääsuõiguste süsteemi	Madal	Suur	<ul style="list-style-type: none"> <li>• Korralik testimine liivakastis</li> </ul>

## 13.2 Tehnoloogilised riskid

Ametlik OWASP TOP 10 nimekiri sisaldab mitte ainult rakenduse (tehnoloogia) spetsiifilisi punkte, kuid ka organisatoorseid. Allpool on kirjeldatud tehnoloogia riskid ja maandamismeetodid vastavalt igale punktile OWASP Top 10-st.

	Owasp risk	Risk	Maandamise meetodid
1	<i>Injection</i>	Päringusse sisestatakse vööras SQL-i osa	<ul style="list-style-type: none"> <li>• puhastada (rakendada escapemist jt lahendusi andmete salvestamisel) ja kontrollida kasutajate sisendi</li> <li>• kasutada teeki, mis koostab SQL päringu kasutades nõ <i>SQL Bind Variables</i></li> </ul>
2	<i>Broken Authentication</i>	Autentimise funktsioon on vigane või poolik	<ul style="list-style-type: none"> <li>• kasutada OAuth2 protokollil põhineva autentimise viisi</li> <li>• kasutada tugeva autentimisviisi</li> <li>• piirata autentimise valekatsete arvu</li> </ul>
3	<i>Sensitive Data Exposure</i>	Lahtiste andmete saatmine	<ul style="list-style-type: none"> <li>• andmete transportimisel kasutada krüpteerimist</li> <li>• rakendada kliendi sertifikaadi kontroll, kui on olemas oht, et keegi suudab endale hankida usaldusväärse sertifikaadi teise nime alt</li> </ul>
4		Andmete ebaturvaline hoidmine	<ul style="list-style-type: none"> <li>• kasutada krüpteerimist failisüsteemi, andmebaasi või konkreetsete väljade tasemel</li> </ul>
5	<i>XML External Entities (XEE)</i>	XML failide töötlemisel koodi käivitamine serveril	<ul style="list-style-type: none"> <li>• liideste testimine rünnaku vastu</li> <li>• java teekide kontroll</li> <li>• kasutada XML-i asemele näiteks JSON-it</li> </ul>
6	<i>Broken Access Control</i>	Ligipääsu halduse puudused	<ul style="list-style-type: none"> <li>• kasutada rakenduses Spring Security vms teeki, mis vähendab eksituste võimalust arendajate poolt</li> </ul>

7	<i>Security Misconfiguration</i>	Lõppkasutajaga üleliigse informatsiooni jagamine vigade ja põhjuste kohta	<ul style="list-style-type: none"> <li>• kasutada inimloetavad vead mis ei sisalda tundlikku informatsiooni</li> </ul>
8	<i>Cross-Site Scripting</i>	Süsteemi mõjutamine sedapidi, et kasutajale edastatakse ründaja koodi (murdskriptimine)	<ul style="list-style-type: none"> <li>• kasutajate sisendite kontroll</li> <li>• rakendada escapemist jt lahendusi andmete salvestamisel</li> </ul>
9	<i>Insecure Deserialization</i>	Süsteemi töös oleku mõjutamine pahatahtlikult moodustatud andmete edastamisega	<ul style="list-style-type: none"> <li>• kasutajate sisendite kontroll</li> <li>• rakendada escapemist jt lahendusi andmete salvestamisel</li> <li>• kasutada programmeerimiskeeli ja teeke mis on selle riski vastu eelnevalt kontrollitud</li> </ul>
10	<i>Using Components With Known Vulnerabilities</i>	Kasutatud komponendid sisaldavad turvaauke ja muid probleeme	<ul style="list-style-type: none"> <li>• ehitamise (<i>build</i>) käigus käivitada sõltuvuste kontrolli vastu liste, mis sisaldavad turvaaukude raporteid. Kuid selle kontrolli tulemusega peab olema ettevaatlik, sest mitte iga sõltuvus ei põhjusta probleemi</li> </ul>
11	<i>Insufficient Logging And Monitoring</i>	Tegevused ei ole piisaval määral logitud või logi eripäradele ei ole õigeaegset reaktsiooni	<ul style="list-style-type: none"> <li>• luua korralik organisatoorne protsess: kes, mis hetkel, kuidas vaatab ja reageerib logidele</li> <li>• täiendada MFN logimise nõuetega</li> </ul>

## 14 Arendusplaan ja prioriteetidid

Prioriteetide hindamisel rakendatakse MoSCoW meetodit:

**M** - nõue peab kindlasti kavandatavas süsteemis olema olemas.

**S** - nõue on kriitiline, kuid äärmisel juhul saab hakkama ka ilma selle nõude täitmiseta.

**C** - oleks hea, kui nõue saab kavandatavas süsteemis lisatud. See tehakse juhul, kui ressursi jätkub.

**W** - neid nõudeid kavandatava süsteemi raames ei vaadelda. Jäädavad realiseerimiseks hilisemasse tulevikku.

	Etapp	Arendus	Süsteem	Kirjeldus	Prioriteet
1	Portaali ettevalmistamine sündmus- ja osateenuste liidestamiseks	Liivakasti (sandbox) loomine teenusepakkuja liidestuste iseseisvaks testimiseks	Devops	Liivakasti keskkonna loomine Keskkonna konfigureerimine <i>Pipeline</i> -ide ettevalmistus Turvaaspektide hindamine	M
2		Teenuse artiklite haldus	Artiklivaramu	Teenuse artiklid on artiklite eriliik osateenuste ja sündmusteenuste vormidel ja teenuseid kirjeldavatel lehtedel kasutamiseks. Artiklivaramu peab võimaldama eristama teenuse artikleid ja võimaldama nende haldust.	M
3		Riigiportaali täiendavad pääsuõiguste konfiguratsioonid	Ruuter	vt peatükk <a href="#">8. Pääsuõigused</a> Ruuteri konfiguratsioonid	M
4		Menüü koostamise algoritmi täiendamine	CMS	Menüü koostamine CMS lehtede nimekirja alusel CMS mallide täiendamine: lehe määratlemine menüüpunktina Menüü publitseerimise funktsiooni täiendamine	M
5		Domeeni jagamise komponent	Devops	Menüü uuendamine ja publitseerimine CDN-i Ettevalmistada Jenkins <i>pipeline</i> -id (Grav menüüde publitseerimiseks CDN-i)	M

	Etapp	Arendus	Süsteem	Kirjeldus	Prioriteet
6		Sündmusteenuse staatuse jälgimise komponent	CMS	Sündmusteenuse jälgimise võimaldamine riigiportaali CMS süsteemis uued mallid ( <i>blueprint layout</i> , 2 malli) uute mallide töötluste lisamine front-end-i	M
7		<i>Microfrontend</i> lahendus	Devops	<i>Microfrontend</i> lahenduse paigaldamise protsessi täiendamine Ettevalmistada Jenkins pipeline-id Repositooriumite ettevalmistustööd	M
8		Vormilahenduse komponent	CMS	Vormilahenduse välja tüüpide lisamine riigiportaali CMS mallidele ja front-end komponendile. 4 uut tüüpe (nt numbri-, e-maili väljad, textarea, combo välisklassifikaatoriga). Stiilide käsitlemine. Vormi tegevuste ( <i>actions</i> ) täiendamine. Veatöötlus. Kinnituse modaalid, edu- ja veateated.	M
9		Paranduste sisseviimine turvatestide tulemuste alusel			

	<b>Etapp</b>	<b>Arendus</b>	<b>Süsteem</b>	<b>Kirjeldus</b>	<b>Prioriteet</b>
10	Sündmus- osateenuste liidestamise lihtsustamine ja	Teenuse artiklite kuvamine	CMS	Artikli stiilide ja <i>layout</i> variantide (3 varianti) kasutamine  Grav mallide koostamine  Front-end komponentide täiendamine	S
11		Süvalingi komponent	CMS	Süvalingi lahenduse <i>callback</i> funktsiooni täiendamine (2 lisatüüpi)	S
12		Sündmusteenuse metaandmete täiendamine	Teenuste kataloog	Metaandmete täiendamine vähemalt järgmises mahus:  teenuste omavaheline seos (teenus-osateenus, seotud teenused),  teenuste atribuudid (nt teenuse sihtgrupi tunnus),  teenuse unikaalne identifikaator	S
13		OIDC implementeerimine	TIM	Autentimise ja autoriserimise muutmine OIDC standardile ( <i>OpenID Connect</i> )	S
14		Domeeni jagamine	CMS	Riigiportaali viite (defaultPrefix) parameetri lisamine grav_adapter poolt tagastatav menüü struktuuri	S
15		Paranduste sisseviimine turvatestide tulemuste alusel			
16	Halduse lihtsustamine	Ruuteri konfiguratsiooni üleviimine YAML formaadi peale	Ruuter	Ruuter komponendi konfiguratsiooni üleviimine YAML formaadi  Konfiguratsiooni interpretaatori muutmine uue formaadi lugemiseks	C

	<b>Etapp</b>	<b>Arendus</b>	<b>Süsteem</b>	<b>Kirjeldus</b>	<b>Prioriteet</b>
17		Sündmusteenuse metaandmete täiendamine	Teenuste kataloog	Metaandmete täiendamine vähemalt järgmises mahus: õigusaktide viited, kirjelduste viited (Artiklivaramu artiklid), tegevusvaldkonnad	C
18		Vormilahendus	CMS	<i>Grid layout</i> vormi jaoks (võimalus kasutada 12-veeru <i>grid</i> ) Plokkide kasutamine vormidel	C
19		Vormilahendus	CMS	Baasloogika vormidele (kontrollide tingimuslik kuvamine, tingimuslik väljade eeltäitmine)	C
20		Metaandmete päring	CMS	Teenuste kataloogis registreeritud metaandmete kasutamise võimalus	C
21		Paranduste sisseviimine turvatestide tulemuste alusel			

## 14.1 Täiendused liidestatud süsteemides

Kasutajakogemuse parendamiseks soovitame enne riigiportaali liidestamise protsessi alustamist teha liidestavates süsteemides järgmised täiendused:

1. TARA SSO tugi
2. Riigiportaali teekiga valmistatud lehed domeeni jagamise võimaluse kasutamiseks

Lisaks sellele on mõistlik arendada ka masinõppe tehnoloogiatel põhinevat soovitusüsteemi, mis aitab parandada informatsiooni ja vajalikke teenuste otsingut.

## 15 Halduskulude hinnang

Pakutud arendused ja PoC projekti raames valminud liidestamise variandid ei mõjuta oluliselt olemasolevaid halduskulusid.

Pääsuõiguste konfiguratsioonide halduse suurus sõltub liidestatavate süsteemide arvust ja Pääsuke süsteemi olemasolust, kuid käesolevad halduskulud oluliselt ei muutu. Pääsuke süsteemi valmimine vähendab vajadust jälgida pääsuõiguste konfiguratsioonide muudatusi.

PoC projekti käigus valminud Artiklivaramu artiklite kasutamise võimalus. Olemasolevad artiklid Artiklivaramust saab integreerida otseselt andmevahetuse teenuse kaudu. Otseselt Artiklivaramust kasutatud artiklite haldus riigiportaalil ei ole enam vajalik.

### 15.1 Riigipilve kasutamise maksumus

Riigipilve kasutamise maksumuse arvutamise aluseks on võetud peatükis [7.5 Tulevikulahenduse arhitektuuri kirjelduses](#) tehtud arvutused ja riigipilvi kasutamise hinnakiri <https://riigipilv.ee/hinnakiri>. Arvutuses oli kasutatud tavaketta maht 100 GB.

K8S klasteri igaaastane majutuse hind on 1361.45 EUR.