

## 1. Teostatavad arendustööd

Häirekeskuse 112 teenuse IT-süsteemi arendustööd viiakse läbi vastavalt Euroopa Liidu ligipääsetavuse direktiivile (Directive (EU) 2019/882) ja NG eCall EL-i määrusele 2024/1180. Arendustööd keskenduvad uuele kõnelahendusele, mis on SMIT-is arendatav kõnejuhtimise, kõnehalduse ja agendihalduse tarkvara. Lahendus läheb kasutusse novembris 2025 ning asendab hetkel Häirekeskuses kasutusel olevad lahendused. Uus kõnelahendus on osa suuremast platvormi uuendusest, mille käigus toimub üleminek browseripõhisest monoliitsest lahendusest modulaarsemale desktop-lahendusele. Desktop-lahendust tervikuna nimetatakse HKSOS-ks.

### Ligipääsetavuse direktiiv (Directive (EU) 2019/882)

**Eesmärk:** Parandada ligipääsetavust kõigile kodanikele, sh puuetega inimestele, tagades, et digitaalsed teenused ja sidevahendid vastaksid kõrgematele ligipääsetavuse standarditele.

#### Tehniline kirjeldus:

- Direktiiv määrab kindlaks miinimumnõuded info- ja kommunikatsioonitehnoloogia (IKT) toodetele ja teenustele, sealhulgas hädaabikõnede süsteemidele.
- Nõuab, et hädaabiteenused toetaksid mitut sideviisi, sealhulgas videokõnesid, RTT-d (reaalajas tekst), tekstitelefone ja muid alternatiivseid suhtlusviise.
- Eesmärk on tagada, et puudega isikud, näiteks kuulmis- ja kõnepuudega inimesed, saaksid võrdselt kasutada hädaabiteenuseid ilma vahendajata.
- Standardite kohaselt peavad kasutajaliidesed olema ligipääsetavad ja sisaldama näiteks kõrge kontrastsusega visuaale ning hääljuhiseid.

### Olulised arendused

#### Videokõne lahendus

- Videokõne teenuste arendamine .NET WPF raamistikul (HKPHOS)
- Videokõne kasutajaliidese arendamine erinevate nõuete põhjal .NET WPF raamistikul (HKPHONUI)
- Videokõne järelvaatamisteenuse arendamine .NET WPF raamistikul (HKCALSS, HKCALS)
- Videokõne järelvaatamise kasutajaliidese arendamine erinevate nõuete põhjal .NET WPF raamistikul (HKCALLUI)

#### RTT (Real-Time Text) lahendus

- RTT teenuste arendamine .NET WPF raamistikul (HKPHOS)
- RTT kasutajaliidese arendamine .NET WPF raamistikul (HKPHONUI)

### NG eCall EL-i määrus 2024/1180

**Eesmärk:** NG (Next Generation) eCall süsteem on uue põlvkonna hädaabikõnede teenus, mis tagab kiire ja täpse info vahetamise sõidukite ja häirekeskuste vahel, parandades liiklusõnnetuste korral reageerimisvõimet ja ohutust.

#### **Tehniline kirjeldus:**

- Määrus 2024/1180 laiendab eCall süsteemi võimalusi, et toetada IP-põhiseid sideprotokolle ja andmeedastust.
- Nõuab, et NG eCall rakendaks Advanced Mobile Location (AML) teenuseid, mis saadavad automaatselt täpse asukohainfo häirekeskusele.
- Võimaldab andmevahetust sõidukite ja päästekeskuste vahel, sh sõiduki täpne asukoht, tüüp, kütuseliik, turvapatjade olek ja muud avariiga seotud parameetrid.
- Toetab nii automaatseid kui ka käsitsi algatatud hädaabikõnesid, mis edastavad lisaks hääljuhitud kommunikatsioonile ka andmepõhiseid teateid.
- NG eCall toob kaasa ülemineku 3G võrgult 4G/5G-põhisele sidele, mis võimaldab kiiremat ja usaldusväärsemat andmeedastust ning paremat kõnekvaliteeti hädaolukordades.

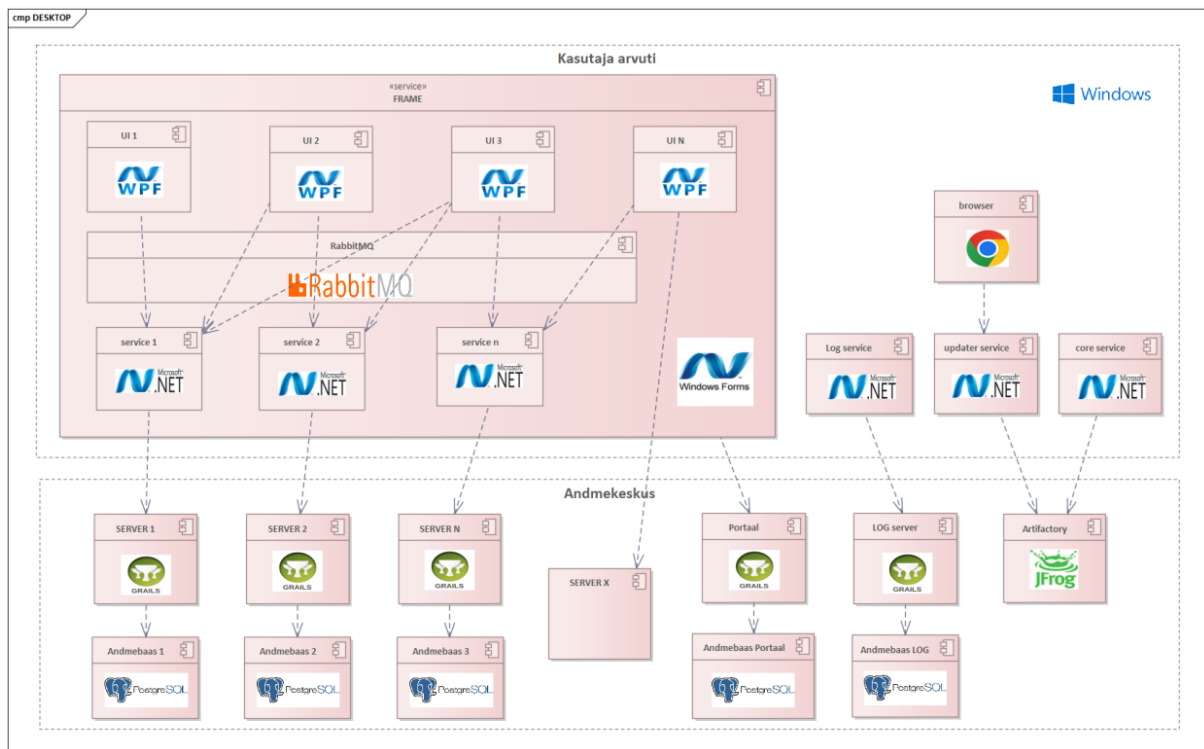
#### **Olulised arendused**

##### **NG eCall lahendus**

- Kõneteenuse kohandamine(uues kõnelahenduses) NG eCall kõnede .NET WPF raamistikul
- Kõnelahenduse kasutajaliidese kohandamine NG eCall andmetele .NET WPF raamistikul

Arendused toimuvad kahes põhikomponendis: front-end (kasutajaliideseid) ja back-end (teenused). Eesmärk on tagada, et Häirekeskuse 112 teenused oleksid kooskõlas EL-i nõuetega, parandades ligipääsetavust ja reageerimisvõimet kriitilistes olukordades.

## 2. HKSOS tehniline lahendus



### 2.1. Üldine

HKSOS on uus Häirekeskuse hädaabiteadete ning abi- ja infoteadete haldamiseks loodud modulaarne lahendus.

Lahendus võeti kasutusele alates 09.2019 riigiinfo telefoni 1247 töökohtades ja alates 03.2022 osaliselt 112 töökohtades.

Lahendus võimaldab lähtuvalt kasutaja tööst (rollist) käivitada konkreetses arvutis, vaid vajalik hulk mooduleid sh vastu soovitud andmekeskuse keskkonda. Vastavalt loodava funktsionaalsuse korduvkasutatavusele on kavas lahenduse juurutamine ka Häirekeskuse väliselt (SMIT, PPA, SiM jne).

Lahendus koosneb andmekeskuses olevatest serveritest ning kasutaja arvutis asuvast modulaarsest DESKTOP lahendusest (edaspidi "Desktop"). Kasutaja arvutis olevad moodulid (Frame, UI komponendid ja Serviced) on kõik eraldi tarkvarad.

Moodulid arendatakse .NET core platvormil, UI komponentides kasutatakse WPF-i. Moodulite tööd orkestreerib Frame (autentimine, vajalike UI komponentide käivitamine/sulgemine ja asukohad/paiknemine frames, Servicete äratamised/uinutamised jne).

Andmekeskuses olevate serveritega suheldakse üldjuhul Servicete kaudu. Andmevahetus moodulite vahel toimub läbi lokaalse RabbitMq.

Erijuhtudel suhtlevad UI komponendid andmekeskuses olevate serveritega otse (Frame kasutaja autentimiseks, UI moodulid SMIT sisevõrgus avalike n: GEO teenuste tarbimiseks jne).

Logi edastavad moodulid otse lokaalsele logi kogumise ja edastamise moodulile (log servicele), mis pakub localhost-is logide kogumise teenuseid.

Mooduleid paigaldatakse ja uuendakse eraldi uuendaja mooduli (updater service) kaudu.

## 2.2. Rakendusserverid

Andmekeskuses olevad serverid (**backend**) on realiseeritud Grails-is (ver 6.\*). Grails on vabavaraline Spring Boot-il põhinev veebirakenduste arendusplatvorm, mis kasutab Apache Groovy objektorienteeritud programmeerimise keelt (põhineb omakorda Java platvormil). Rakendusserverite arenduses kasutatakse üldkasutatavates funktsioonides SMIT-i arendatud Grails pluginaid (igas uues rakenduses ei pea uuesti realiseerima üldkasutatavaid funktsioone: autentimine, autoriseerimine, andmebaasi notificationide kuulamine jms).

Andmebaasina kasutatakse PostgreSQL andmebaasi (ver 15.\*) ja PostgreSQL teavitusi (NOTIFY/LISTEN).

Rakendusserveritel on haldus-andmete administreerimise kasutajaliidesed ning sisu-andmete päringu/loetelu kasutajaliidesed operatiivtöö „väliste“ kasutajagruppide jaoks. Sisu-andmete aruandlus realiseeritakse andmelaos (ärandab/haldab eraldi SMIT meeskond). Kasutajaliidese vaated realiseeritakse Grails-i vahenditega, kasutades SMIT-i Grails pluginaid (lihtsustavad jqGrid-i kasutamist). Kasutajaliideses kasutajate tegevuslogi edastamiseks kasutatakse vastavat SMIT-i Grails pluginit.

Rakendusserverid töötavad samaaegselt mitmel õlal, toetavad mitut erinevat andmebaasi ühendust (*ala* primary/secondary), andmete muudatustest saavad teada läbi andmebaasi teavituste. Rakendused paigaldatakse docker konteineritena, mille sees on rakendus paketeeritud BootJar'na koos Tomcat'ga.

Töölaua (desktop frontend) lahendusega suhtluseks pakutakse REST teenuseid ja SSE-d (Server-Sent Events). Andmeid vahetatakse JSON formaadis. Iga loodud REST teenuse osas tuleb koostada dokumentatsioon Spring REST Docs-iga.

Rakendusserverid on loodud/luuakse põhiliste andmeobjektide/domain-i (kõne, sms, raadioside, 112 sündmus, 1247 sündmus, ressurss, arhiiv jne) või funktsionaalsus-gruppide (e-posti automaatteavitused, andme-edastus x-Tee-le, kasutajate haldus, klassifikaatorite/reeglite haldus, kaardi põhifunktsioonid jne) kohta. Rakendusserverite vaheline suhtlus käib REST teenuste või sõnumiserveri (RabbitMQ) kaudu.

## 2.3 RabbitMQ

Andmevahetus töölauda moodulite vahel toimub läbi lokaalse RabbitMQ. Moodulid edastavad iga erineva sõnumi struktuuri vastava kanali sees kasutatavas ühes queue-s või topic-us.

## 2.4 Frame

Frame on keskne moodul, mis orkestreerib kogu töölaua tööd. Hetkel realiseeritud .NET Windows Forms rakendusena.

Frame põhilised funktsioonid:

- Äratab konkreetsele kasutajale (profiilile) vajalikud teenused/Service-d (arvutis võib olla Service-id, mida ei äratata/pole hetkel vajalikud);
- Kujundab UI komponentidest kasutaja vaate;
- Edastab UI komponentidele vaike stiili ja režiimi;
- Korraldab sisse logimise (autentimise) ning levitab saadud 'SmitTokenId' Service-tele, et viimased saaksid luua autenditud ühenduse andmekeskuses olevate serveritega;
- Esitab Service-te olekud (foor), UI-de uute andmete teke/loendurid ning vajadusel Service-te või UI-de töös esinevad probleemid (veateated);
- Võimaldab töö käigus vahetada kasutaja režiimi ja stiili;
- Sulgeb väljumisel UI moodulid ning uinutab Service-d (laseb lõpetada ühendused andmekeskuses asuvate serveritega ja oma mälud tühjendada);
- Edastab oma logid lokaalsele logi teenusele.

Frame suhtleb Service-tega ning UI komponentidega läbi RabbitMQ kasutades FrameMessaging nuget-is kirjeldatud andmestruktuure.

## 2.5 Serviced

Service-te ülesandeks on hoolitseda andmevahetuse eest ühe konkreetse andmekeskuses oleva rakendusserveriga. Realiseeritud .NET core rakendustena (Windows Service-tena). Service-te tööd juhib (äratab, annab serveriga ühenduseks vajalikud parameetrid sh 'SmitTokenId' ning uinutab) Frame.

Andmekeskuses olevate rakendusserveritega ühendustes kasutatakse sertifikaadi põhist krüpteerimist ning suhtlus käib üldjuhul pordil 443 (REST ja SSE). Telefonijaamadega suhtluses kasutatakse SIP protokoll.

Service-te põhilised funktsioonid (sõltuvalt Service-st võivad need olla erinevad):

- Teenuse käivitamisel Windows-i poolt RabbitMQ-ga ühendumine ja sõnumite kuulama hakkamine;
- Ärkamise (wake up) käsu peale, sõnumis edastatud rakendusserveriga ühenduse võtmine ja autentimist mittevajavate andmete (n: enum-id) pärimine;
- Ühenduse (connect) käsu peale, sõnumis edastatud 'SmitTokenId'-ga rakendusserveriga ühenduse loomine (nö autoriseeritud sessioon), sh vajadusel:
  - algandmete pärimine ning oma cache-s laadimine;
  - SSE kanali loomine ning edasine üleval hoidmine;
  - varem saatmata jäänud andmete edastamine.
- Rakendusserverist saadud algandmete edastus UI-dele või UI-de algpäringule vastamine peale algandmete laadimist/cache-i täitmist;
- Rakendusserverist SSE (server-sent-events) kaudu laekuvate andmete edastus UI-dele;

- UI-delt saadud sõnumite (päringud, sisuandmete haldus) edastus serverile ja vastuse tagastus, sh vajadusel serverile edastamata jäänud sõnumi edasiseks saatmiseks salvestamine;
- UI-delt saadud päringutele lokaalsest cache-st vastamine;
- Uinumise (close) käsu peale serveriga ühenduse lõpetamine ja nõ mälu puhastamine;
- Oma logide edastus lokaalsele logi teenusele.

Eelnimetatud funktsioonid on realiseeritud SMIT-i universaalsetes komponentides (nuget-ites).

Service suhtleb Frame-i ning UI komponentidega läbi RabbitMQ kasutades selleks vastavalt FrameMessaging või oma ServiceMessaging nuget-is kirjeldatud andmestruktuure.

Service töö ei sõltu teistest Service-test.

## 2.6 UI moodulid

UI moodulid käivitab Frame, mis ühtlasi seadistab kasutaja profiilist sõltuva vaate (paigutab UI moodulid ekraanile).

UI moodul kuvab oma elemendid dünaamiliselt (responsive), st mooduli suuruse muutumisel ei tohi midagi kaduda 'nurga taha' või tekkida kerimisriba. Rakendust kasutatakse FHD (1920x1080) ja QHD (2560x1440) monitoridel, sh näiteks 480 ja 640 pikseli laiuste appidena. Kasutajaliidese värvilahenduse edastab Frame UI mooduli käivitamisel (näiteks 112 ja 1247 kujundus või päeva ja öö režiim).

UI moodulid suhtlevad Service-tega läbi RabbitMQ. UI moodul kuulab üksnes neid Exchange ja nendest Service-test, mis on talle olulised.

Juhul kui päring ei vaja autentimist, võib päringu teostada otse UI moodulist andmekeskuses asuva rakendusserveri poolt pakutavasse teenusesse (näiteks aadressiteenus).

Funktsionaalsuste puhul, mis vajavad klassifitseeritud andmeid (näiteks klassifikaatorite esitatavate väärtuse näitamine) peab UI moodul kasutama arvuti failisüsteemis olevaid JSON kujul olevaid klassifikaatorite faile.

## 2.7 Updater service

Windows service, mis tegeleb moodulite paigaldamise ning uuendamisega. Pakub 'localhost' - is teenust brauseri vaatele, milles kasutaja valib soovitud keskkonna ja profiili ning teenus paigaldab/uuendab vajalikud moodulid ning konfiguratsiooni failid.

## 2.8 Log service

Windows service, mis pakub kõigile lokaalsetele moodulitele logide edastamise teenust. Moodul võtab vastu tehnilise ning kasutajate tegevuste logid ning edastab need keskssele logide rakendusserverile.

## 2.9 Classifier service

Windows service, mis pärib klassifikaatorite andmed ning salvestab need JSON kujul arvuti failisüsteemi, mida teised moodulid (UI-d ja Service-d) saavad vajadusel kasutada.

# 3 Arendusnõuded

## 3.1 Arenduse juhtimine

Arendus dokumenteerida ja arendust juhtida SMIT-i Confluence-s ning JIRA kaudu.

- JIRA arendusülesanded koostada struktuurset, st jagada suuremateks osadeks ja viimased omakorda väiksemateks töödeks (Epic, Story, Task). Võimalusel siduda Confluence-s koostatud analüüsiga ja Figma UI disainiga;
- arenduspiletite täitmisel kasutada JIRA töövoogusid (In Progress, Blocked, Code Review, In Review) ning kommentaare (küsimused analüütikule, arenduse täpsustused testijale jms);

## 3.2 Arenduse protsess

- Tarkvara tarnitakse SMIT-i GIT-i;
- Tulemit JIRA arenduspiletite kaupa tarnides arendaja:
  - laeb koodi ülesse JIRA pileti nimelisse harusse, sh viitab commit-is JIRA piletile. Commit-e soovitage teha korra päevas, näiteks tööpäeva lõpetades;
  - funktsionaalsuse valmimisel teeb pull request-i JIRA pileti nimelisest harust develop harusse ning muudab JIRA pileti olekusse 'Code Review' (pull request-i peale käivitub automaatne protsess, mille käigus tehakse koodile staatiline analüüs ja käivitatakse automaattestid);
  - pull request-i järgselt kontrollib, et automaatne protsess oleks edukalt läbitud, ja ootab teiste arendajate poolt koodi ülevaatuse tulemit. Vajadusel kõrvaldab esinenud puudused;
  - mergeb koodi develop harusse;
  - töölaua moodulite muudatuste korral kontrollib merge järgselt, kas koodist ehitus automaatselt exe, mis on testijale/vastuvõtjatele kättesaadav. Kui ehitus toimis, siis muudab JIRA pileti olekusse 'In Review'. Kui ei toimunud, siis lahendab iseseisvalt või koostöös hankijaga ehitusel tekkinud probleemid.

## 3.3 Nõuded koodi ülesehitusele

- Rakendus tuleb üles laadida SMIT-i GIT-i järgmise struktuuriga:
  - ROOT kataloogis peab olema mooduli nimeline projekti fail (Smit<moodul>.sln);
  - Rakendus peab olema ROOT kataloogis olevas mooduli nimelises kataloogis Smit<moodul>;
  - Testid peavad olema ROOT kataloogis olevas eraldi kataloogis Smit<moodul>.Tests;
  - ROOT kataloogis peab olema kataloog nimega 'doc', kus peab olema mooduli kirjeldus ning juhend. Kui moodul vajab välist konfifaili, siis ka kommenteeritult näidis konfifail.
- Andmevahetuses esinevad andmestruktuurid (näiteks serverist saabuval JSON-id või RabbitMQ kaudu vahendataval JSON-id) tuleb kirjeldada eraldi nugeti projektina, mille nimi on <mooduli nimi>-service-messaging. Igal Service-l on oma andmestruktuuridega

kirjeldatud nuget. Antud nuget kaasatakse nii Service-le kui ka UI moodulitele, mis suhtlevad konkreetse Service-ga.

- UI moodulites tuleb hoida eraldi andmete mudel ning esitlusmudel, st kasutada Model View ViewModel (MVVM) mustrit.
- Moodulites ei tohi kasutada kolmandate osapoolte teekes. Mooduli sõltuvused võib laadida üksnes SMIT-i sisemisest repositooriumist. Juhul kui kolmandate osapoolte teekide kasutamine oleks 'mõistlik', tuleb see iga kord eraldi tellijaga kokku leppida.
- Kood tuleb kokku pakkida 'publish' käsuga üheks versioneeritud exe failiks.
- Kasutada tuleb SMIT-is seni loodud järgmisi universaalseid komponente:
  - Nuget, mis korraldab RabbitMQ ühenduse ning konkreetsete queue/topic-ute kuulamise;
  - Nuget, mis tegeleb FRAME integratsiooniga;
  - Nuget, mis tegeleb andmekeskuses oleva serveri sessiooniga ning SSE kanali hoidmisega;
  - Nuget, mis tegeleb logimisega;
  - Nuget, kus on kasutajaliidese elemendid.

### 3.4 Nõuded koodi kvaliteedile

Kood peab olema inglise keelne, meetodite ning muutujate nimest peab olema aru saada milleks neid kasutatakse.

Olulised klassid peavad olema kommenteeritud.

Kood peab läbima standardse SONAR analüüsi. SMIT-is kasutatavad kvaliteedimõõdikud ja nende lubatud väärtused:

- Coverage - mõõdik määrab mitu protsenti kogu lahenduse lähtekoodist on kaetud automaatsetest testidega. Alampiir on 80%;
- Duplicated Lines (%) - mõõdik määrab mitu protsenti lähtekoodi koodist võib olla dubleeritud. Ülempiir on 3%;
- Maintainability Rating - mõõdik määrab mitu protsenti kogu rakendusele kulunud ajast on tekitanud tehnilist võlga (Code Smell) koodi stiili ja standarditele vastavuse osas. Lubatud maht on alla 5%-i (tase A);
- Reliability Rating - mõõdik näitab palju on koodis erinevaid vigu. Uues ja refaktoiritud koodis ei tohi olla ühtegi viga (tase A);
- Security Rating - mõõdik näitab palju on turvanõrkuseid rakenduses, mis on Sonari poolt tuvastatud. Ei ole lubatud ühtegi turvanõrkust (tase A).

### 3.5 Nõuded logimisele

Kasutada tuleb SMIT-is loodud spetsiaalset logimise nuget-it, kus on defineeritud logi kirje muster, kohad kuhu logi kirjutatakse ning ka log objekt. Iga äriprotsessi alguses tuleb luua uus log objekt, kuhu pannakse äriprotsessi alguse aeg ning sisendiks saadud info (näiteks moodulisse saabunud JSON). Log objekti tuleb kaasas kanda kuni äriprotsessi lõpuni, kus logi kirje luuakse. Protsessi käigus saadud vea korral kirjutatakse log objektist võetud sisend veateate juurde.

- Moodulid kirjutavad logi INFO tasemel.
- INFO tasemel logitakse üksnes olulisi, konkreetsetes arvutis toimuvaid tegevusi.

- Perioodilisi tegevusi üldjuhul ei logita INFO tasemel.
- Üks äriprotsess peab olema logitud ühe logi reana. Logi peab olema informatiivne, ühelt realt peab olema aru saada mida moodul tegi.
- Vea korral peab olema logi kirje alusel võimalik viga korrata, st logis peab olema sisend, mis põhjustas vea.

Tehnilisele logile lisaks tuleb SMIT-iga kokkulepitud UI funktsioonide kasutamine logida nõ täiendavate kasutaja tegevuslogi kirjetena. Tegevuslogi kirje peab sisaldama viidet hallatavale objektile, tegevuse nime/tunnust, teostamise aega ning tegevuse teostanud kasutaja andmeid. Tegevuslogi koostamisel tuleb kasutada vastavat nuget-it ning selles kirjeldatud logimise mustrit.

Tehniline logi ja tegevuslogi edastatakse kirjete kaupa lokaalsele Log Service-le, mis edastab need Log rakendusele.