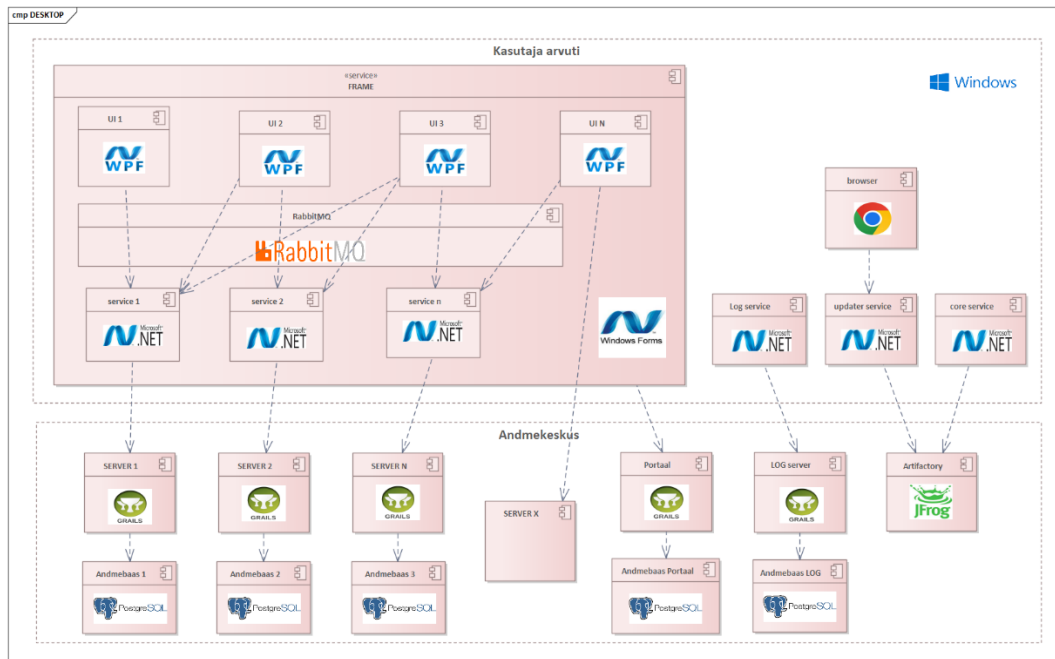


HKSOS tehniline lahendus



1.1. 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/sulgumine 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 uuendatakse eraldi uuendaja mooduli (updater service) kaudu.

1.1.1. RabbitMQ

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

1.1.2. Frame

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

Frame põhilised funktsioonid:

- Äratab konkreetsele kasutajale (profiilile) vajalikud Serviced (arvutis võib olla Serviceid, mida ei äratata/pole hetkel vajalikud);
- Kujundab UI komponentidest kasutaja vaate;
- Korraldab sisse logimise (autentimise) ning levitab saadud 'SmitTokenId' Servicetele, et viimased saaksid luua autentitud ühenduse andmekeskuses olevate serveritega;
- Esitab Service olekud (foor) ning vajadusel Service või Ui-de töös esinevad probleemid (veateated);
- Võimaldab töö käigus vahetada kasutaja profiili;
- Sulgeb väljumisel UI moodulid ning uinutab Serviced (laseb lõpetada ühendused andmekeskuses asuvate serveritega ja oma mälu tühendada);
- Edastab oma logid lokaalsele logi teenusele.

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

1.1.3. Serviced

Service ülesandeks on hoolitseda andmevahetuse eest ühe konkreetse andmekeskuses oleva serveriga. Realiseeritud .NET core rakendustena (Windows Servicetena). Service töö juhik (äratab, annab serveriga ühenduseks vajalikud parameetrid sh 'SmitTokenId' ning uinutab) Frame.

Andmekeskuses olevate serveritega ühendustes kasutatakse sertifikaadi põhise krüpteerimist ning suhtlus käib üldjuhul pordil 443 (rest ja sse). Service põhilised funktsioonid (sõltuvalt Servicest võivad need olla erinevad):

- Teenuse käivitamisel Windows-i poolt RabbitMQ-ga ühendumine ja sõnumite kuulama hakkamine;
- Ärkamise (wake-up) käsu peale, sõnumis edastatud serveriga ühenduse võtmine ja autentimist mittevajavate andmete (n: enum-id) pärimine;
- Ühenduse (connect) käsu peale, sõnumis edastatud 'SmitTokenId'-ga serveriga ühenduse loomine (nõ autoriseeritud sessioon) sh vajadusel:
 - algandmete pärimine;
 - SSE kanali loomine ning edasine üleval hoidmine;
 - varem saatmata jäänud andmete edastamine.
- Serverist saadud algandmete edastus UI-dele;
- Serverist SSE (server-sent-events) kaudu laekuvate andmete edastus UI-dele;
- UI-delt saadud sõnumite edastus serverile ja vastuse tagastus sh vajadusel serverile edastamata jäänud sõnumi edasiseks saatmiseks salvestamine;
- 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 Servicetest.

1.1.4. 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 Servicetega läbi RabbitMq. UI moodul kuulab üksnes neid topic-uid ja nendest Servicetest, mis on talle olulised.

Juhul kui päring ei vaja autentimist võib päringu teostada otse UI moodulist andmekeskuses asuva serveri poolt pakutavasse teenusesse (n: aadressiteenus).

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

1.1.5. 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.

1.1.6. Log service

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

1.1.7 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.

1.2. Nõuded service'itele

HKSOS desktop lahenduse olulised komponendid on windows serviced. Windows serviced on ette nähtud andmeid vahendavateks komponentideks UI-de ning andmekeskuses asuvate serverite vahel. Üks service = üks andmekeskuses olev server rakendus (BackEnd ehk BE). Seega üks service saab pöörduda ainult ühe BE poole. Kogu süsteemis on iga BE kohta üks eraldi töötav service.

Service ülesanneteks on:

- 1) hoida lokaalses arvutis süsteemi tööks vajalikke andmeid (näiteks enumid, operatiivtöös vajalikud sisulised algandmed jne);
- 2) optimeerida UI-de päringuid BE-sse (kui service mälus on operatiivseks tööks vajalikke andmed ning UI päringu parameetrid võimaldavas siis vastatakse UI-le nõ cache-st, mitte ei pärita andmeid BE-st);
- 3) luua vajadusel BE-ga SSE ühendus, hoida seda üleval (taastada katkemise korral) ning võtta vastu BE-st saabunud operatiivandmed ning edastada need UI-dele sh vajadusel uuendada enda operatiivandmete cache;
- 4) vahendada UI-de andmete salvestusi ja päringuid BE-sse;
- 5) tagada vajadusel ajutised (BE puudumisel) salvestused;
- 6) logida oma töö.

Service töötab tavalise windows servicena, mis käivitatakse Windowsi poolt. Service paigaldamisel peab olema märgitud tema sõltuvus servicest - Rabbit MQ. Seega Rabbit MQ service peab olema enne käima läinud. Service käivitamisel Windowsi poolt loob service ühenduse Rabbit MQ-sse ning jääb sealt käske ootama.

Service haldab faile "c:/ProgramData/SMIT/" kataloogi sees oma nimelises kataloogis. Service poolt hallatavad failid on:

1. enumite fail (kõik BE-st saadud enumid ühes failis) või muude sarnaste st staatilised andmete failid (n: call service-l serviceGroups andmed);
2. ajutiste salvestuste draft failid;
3. mõnel servicel ka nõ algandmete failid (näiteks classifier service-l classifiers, classifierClassifiers ja aikEventDataRules);

Service loeb esmasel käivitamisel failidest endale mällu enumite ja algandmete andmed. Draftidega käitumine on servicetes erinev.

Service haldab enda sees 4 olekute foori: **enumid**, **algandmed**, **SSE ühendus** ja **autentimine**. Igal fooril on kolm seisundit - null/false/true. Foori seisundid kuvatakse Frame ui-s värviliselt (väärtustamata-hall [kui teenus pole foori väärtustanud], null-sinine, false-punane, true-roheline). Lisaks on servicel mälus draftide arv.

1.2.1. Service'te standardtegevused

1.2.1.1. Windows käivitab service

Service käivitub sõltuvalt Rabbit MQ servicest ehk siis peale Rabbit MQ service tööle hakkamist. Service käivitub vaikumisi logimise tasemega INFO. Logida tuleb käivitumine INFO tasemel, kirjutades logisse 'Service started' (patternist tuleb see, mis servicega on tegemist). Käima minekul võtab ta ühenduse Rabbit MQ-ga ning hakkab kuulama FRAME-st saabuvat topicut (seda kuulavad kõik serviced) ning UI-dest saabuvat queue (seda kuulab alati konkreetne service üksinda). Käima mineku järgselt kontrollib service kas enumite ja sarnased staatiliste andmete failid eksisteerivad arvutis. Kui kõik konkreetse service poolt hallatavad enum jm failid on olemas ja ei ole tühjad, siis paneb service "enumid" oleku false. Kui kasvõi üks enum failidest puudub või on tühi, siis paneb service "enumid" oleku NAN ehk väärtustab tühjana.

Mõned serviced - näiteks classifier service käitub sarnaselt ka algandmetega. See on väga konkreetne/erandlik juhtum st kus algandmed on BE-st vabalt kätte saadavad kõigile (a'la üldised reeglid ja klassifikaatorid). Kui sellisel servicel on algandmete failid arvutis olemas ja mitte tühjad, siis paneb service "algandmed" oleku false. Kui esineb probleem kasvõi ühe algandmete failiga, siis märgib service "algandmed" NAN ehk väärtustab tühjana. Enamikel servicetel, mis algandmete laadimiseks vajavad autentimist, paneb service "algandmed" oleku NAN ehk väärtustab tühjana.

Servicete käima mineku järgselt pannakse paika ka teised olekud. Kui service ei kasuta SSE-d ning autentimist - näiteks classifier service, siis määratakse koheselt "autentimine" oleku väärtus true. Draftide arv ehk siis failide arv loetakse kokku "c:/programData/SMIT" kataloogis konkreetse service ajutiste salvestuste kataloogist.

Kui kõik olekud on seatud sh draftide arv on kokku loetud, tuleb logida INFO tasemel logi kirje, kus on muutujana kirjas kõik seatud olekute väärtused ning draftide arv.

Service ei tohi veel käivitada ühtki cron-i ega muud arvutit koormavat tegevust (see kehtib servicete kohta, mida juhib FRAME). Selliselt jääb service ootama käskke FRAME-lt ja ka päringuid UI-st.

Sõnumid framelt:

1. Frame töötamisel hakkab ta servicetele saatma iga 5 sek tagant "keepalive" sõnumit, et FRAME on olemas. See sõnum on to ALL, mitte konkreetsele servicele.
2. Kui FRAME poolt saabub "keepalive" sõnum ning service EI OLE ÄRATATUD (üksnes Windowsi poolt tööle pandud), siis service logib:
 1. TRACE tasemel terve JSON-i.
 2. DEBUG tasemel - "received reason (muutuja) but sleeping".
 3. INFO tasemel sellise sõnumi saabumist ei logita.

Päringud UI-delt:

1. Kui UI-st saabub päring ja service EI OLE ÄRATATUD, siis service ei vasta sellele ning logib
 1. TRACE tasemel terve JSON-i.
 2. WARN tasemel - "received request from: (muutuja kellelt) reason: (muutuja), but sleeping." Mõlemad väärtused on sõnumi päises ning JSON sõnumit ei pea parsima.

1.2.1.2. Service äratamine

Tavapäraselt äratatakse serviced enne UI päringuid. Äratamiseks saadab FRAME konkreetsele servicele äratamise st "**wakeup**" sõnumi Rabbit MQ kaudu (sama topic, kust saabuvad "keepalive" sõnumid). FRAME saadab äratamise sõnumi alati vaid konkreetsele servicele, mitte to ALL. Sõnumis on ka BE url, kuhu peab service pöörduma. Lisaks on sõnumis logimise tase. Kui sõnumis puudub logimise tase, peab logimise tase olema INFO. Kui sõnumis on näiteks DEBUG, siis peab service lülituma logimise tasemele DEBUG.

Logida tuleb:

1. TRACE tasemel received wakeup ning terve saabunud JSON sõnum.
2. INFO tasemel received wakeup, logging - (muutuja), url (muutuja).

Service peab panema enda sees tööle cron-i, mis hakkab iga 5 sek tagant saatma enda olemasolu kohta sõnumit FRAME-le, mis koosneb service olekutest.

NB! - varasemalt saatsime seda FRAME "keepalive" sõnumi vastusena, nüüd teeme selle autonoomseks ning eraldame frame tööst.

Sõnumi järgselt teab Frame, et konkreetne service töötab.

NB! kõiki serviceid ei äratata alati, seega äratamine peab olema service põhine.

Peale äratamist pöördub service saadud url-ile ning küsib BE-st enimid ja kõik muud andmed, milleks ei ole vaja autentimist (näiteks classifier service küsib oma algandmed ehk classifiers, classifierClassifiers ja aikEventDataRules, call service-l serviceGroups andmed jne). Kui

enumite teenus (call service-l ka serviceGroup teenus) vastab, siis service uuendab salvestatud faili ning kirjutab selle kohta INFO tasemel logi kirje. TRACE taseme korral tuleb logida terve päring ja vastus. Lisaks paneb service "enumid" oleku true. Kui vastust ei saadud või faili salvestamine ei õnnestu siis tuleb logida ERROR ja "enumid" olekut ei muudeta.

Erand:

1. Classifier service uuendab ka algandmed (classifiers, classifierClassifiers ja aikEventDataRules jne), sest neid saab ilma autentimata pärida.
2. Kui päringud õnnestub siis tuleb uuendada failid ning panna "algandmed" olek true ning kirjutada INFO tasemel logi kirje üldiselt. TRACE tasemel terve päring ja vastus.
3. Kui vastust ei saadud või faili(de) salvestamine ei õnnestu siis tuleb logida ERROR ja "enumid" olekut ei muudeta.

Service hakkab iga 5 sekundi tagant FRAME-le saatma olekute sõnumit, kus on 4 olekut ja draftide arv. Seda tegevust tuleb logida TRACE tasemel terve saadetav sõnum, DEBUG tasemel üldiselt, "sent message ...", INFO tasemel seda ei logita.

1.2.1.3. Service ühendumine

Ühendumiseks saadab FRAME kõigile servicetele Rabbit MQ kaudu (sama topic, kust saabuvad "keepalive" ja "wakeup" sõnumid) ühendumise st "connect" sõnumi, kus on token, millega saab BE-s ennast autentida. Ühendumise käsk saabub to ALL, mitte konkreetsele servicele. Kui varasemalt ei ole saanud konkreetsele servicele äratamise st "wakeup" käsku, siis service ei tea BE aadressi/asukohta ning ignoreerib FRAME poolt antud ühendumise korraldust. See on normaalne olukord, sest konkreetne service ei ole hetkel kasutuses. Saabunud sõnum tuleb logida TRACE tasemel terve JSON-ina. Kui service oli varem äratatud siis ühendub ta "connect" ühendumise sõnumis saadud tokeniga wakeup/äratamise sõnumis saadud aadressil/asukohaga BE-ga. Kui autentimine õnnestub ja BE-st EI TULE vastuseks 40X viga, siis service:

- paneb "autentimine" oleku true;
- saadab kõigile UI-dele (to ALL) Rabbit MQ kaudu sõnumi, ühenduse loomise kohta (peale mida saavad UI-d servisele päringuid saata);
- kirjutab logisse INFO tasemel - received reason connect, authenticated = true.
- Kui autentimine EI ÕNNESTU, siis tuleb vaadata miks ei õnnestunud. Kui põhjuseks on võrgu puudumine, siis tuleb logida WARN tasemel üldiselt mis juhtus. Kui BE ei lase tokeniga sisse, tuleb logida ERROR. Mõlemal juhul peab service jätma sh vajadusel muutma "autentimine" oleku false-ks.
- Seoses tokeni vahetamisega saabub Framelt ka juba autentitud servicele regulaarselt uus "connect" käsk, kus on uus token. Kui servicele on ette nähtud siis loob ta BE-ga peale autentimist ka SSE ühenduse/socketi. Kui see õnnestub, siis logitakse see INFO tasemel ja pannakse "SSE ühendus" olekuks true. Kui ei õnnestu, siis ERROR ja olekut ei muudeta.
- Service jätkab iga 5 sekundi tagant FRAME-le olekute sõnumite saatmist, kus on 4 olekut ja draftide arv. Tegevust tuleb logida TRACE tasemel terve saadetava sõnumina, DEBUG tasemel üldiselt, "sent message ...", INFO tasemel seda ei logita.

1.2.1.4. Service ühendumise lõppemine

HKSOS Desktopis on olemas ka ühendumise lõpetamise ehk väljalogimise võimalus. Ühendumise lõpetamiseks saadab FRAME kõigile servicetele Rabbit MQ kaudu (sama topic, kust saabuvad "keepalive", "wakeup", "connect" sõnumid) ühendumise lõpetamise st "**disconnect**" sõnumi. Ühendumise lõpetamise käsk saabub to ALL, mitte konkreetsele servicele. Sõnumi peale tuleb servisel unustada varasemalt saabunud token ja lõpetada SSE ühenduse, kui see eksisteeris. Enam ei tee service ühtki päringut BE-sse. Frame 'dicconnect" sõnum tuleb logida: TRACE tasemel - terve saabunud JSON sõnum. INFO tasemel üldiselt - received reason disconnect, all disconnected (logi pattern paneb kõik service üldised andmed). Sellised serviced, millel on SSE ühendus ja autentimine, panevad mõlemad olekud false ning edastavad uued olekud sõnumiga FRAME-le. Serviced, kus SSE ühendus ja/või autentimine ei ole kasutusel, jätkuvad olekud jätkuvalt true-ks (need pandi true juba service käivitamisel), sest need pole antud service juures kasutusel. Kui mingi olek muutus ehk siis sellisel servicel, kus oli näiteks SSE ühendus kasutusel ja see pandi false-ks siis tuleb muutus logida INFO tasemel näiteks new flags ning sinna järgi kõikide fooride olekud.

1.2.1.5. Service magama minek close sõnumiga

Servise magama minekuks saadab FRAME kõigile servicetele Rabbit MQ kaudu magama mineku st "**close**" sõnumi. Service katkestab selle peale SSE ühenduse ja ei tee enam BE-sse päringuid. Isegi siis mitte kui UI-st tuleb salvestuse päring. Close sõnumi järgselt läheb service magavasse olekusse, paneb kõik KASUTUSEL olevad olekud false, sest ühendusi enam pole. Kui pole ka enumeid ja ei ole neid kunagi saanud, siis enumite olekud ei muutu ehk jäävad NAN. Sellised olekud, mis EI OLE KASUTUSEL on alati, juba service käivitamisest alates true. Olekud peavad olema paigas, sest kohe võib saabuda uus äratus ning siis peab service ka kohe frame-le vastama. Magav olek tähendab sama seisundit nagu service on kohe peale windows käivitamist. Magama minemisel tuleb eranditult ka kõik cron-id ja timerid sulgeda! Service ei saada enam FRAME -le iga 5 sek tagant olekuid.

Magama minek tuleb logida:

1. TRACE tasemel terve saabunud JSON,
2. INFO tasemel üldiselt received close, going sleep.

Magavas olekus tuleb käituda nagu peale service käivitamist windowsi poolt.

1.2.1.6. Service magama minek kui FRAME ei saada enam iga 5 sek tagant sõnumeid

Service peab jälgima kas FRAME saadab talle ikka iga 5 sek tagant "keepalive" sõnumit st et ta on alles. Selleks võiks olla services timer, mis nullitakse iga FRAME-st saabuva "keepalive" sõnumi peale. Kui FRAME sõnumid lakkavad, siis peab service eeldama, et FRAME on sulgunud või lihtsalt kadunud. Kui 12 sek ei ole FRAME-lt tulnud sõnumeid, siis läheb service magama nagu ka close sõnumi korral. Antud olukord tuleb logida WARN tasemel - keepalive stopped, going sleep. Jälgida tuleb, et mitte ühtki protsessi ei jääks tööle. Üksnes FRAME-st ja UI-dest sõnumite kuulamine jääb tööle.

2. Tarkvaraarenduse nõuded

2.1 Arenduse protsess

Tarkvara tarnitakse SMIT-i GIT-i;

Tulemit jooksvalt st JIRA arenduspiletite kaupa tarnides:

- laeb arendaja koodi ülesse JIRA pileti nimelisse harusse sh viitab commit-is JIRA piletile. Commit-e soovitame teha korra päevas, näiteks tööpäeva lõpetades;
- funktsionaalsuse valmimisel teeb arendaja pull requesti JIRA pileti nimelisest harust develop harusse ning muudab JIRA pileti olekusse 'code review';
- kontrollib arendaja pull request järgselt koodile tehtud staatilise SONAR analüüsi tulemi vastavust nõuetele;
- ootab arendaja teise/teiste arendaja(te) poolt koodi ülevaatuse tulemit (märkuste korral lahendab need);
- aktsepteeringu(te) korral mergeb commiti teinud arendaja koodi develop harusse;
- kontrollib arendaja merge järgselt, kas koodist ehitus automaatselt exe, mis on testijale/vastuvõtjatele kättesaadav. Kui ehitus toimis siis muudab JIRA pileti olekusse 'in review', kui ei siis lahendab iseseisvalt või koostöös Hankijaga ehitusel tekkinud probleemid.

2.2. 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 konffaili, siis ka kommenteeritult näidis konffail.

Andmevahetuses esinevad andmestruktuurid (näiteks serverist saabuavad JSON-id või RabbitMQ kaudu vahendatavad JSON-id) tuleb kirjeldada eraldi nugeti projektina, mille nimi on <mooduli nimi>-service-messaging. Igal servicel on oma andmestruktuuridega kirjeldatud nuget. Antud nuget kaasatakse nii servicele kui ka UI moodulitele, mis suhtlevad konkreetse servicega.

UI moodulites tuleb hoida eraldi andmete mudel ning esiflusmudel st kasutada Model View ViewModel (MVVM) mustrit.

Moodulites ei tohi kasutada kolmandate osapoolte teekide. 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 sh vajadusel neid täiendada:

- 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 UI komponendid.

2.3. Nõuded koodi kvaliteedile

- Kood peab olema inglise keelne, meetodite ning muutujate nimest peab olema aru saada milleks neid kasutatakse.
- Kõik klassid ning meetodid peavad olema kommenteeritud.
- 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 automaatsetestidega. 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 refaktoorigitud 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).
- Vähemalt pooled (arvuliselt) koodi automaatsetestidest peavad olema testid, kus läbitakse äriprotsess tervikuna. Mõte seisneb selles, et ühe meetodi teste oleks vähe, kogu protsess moodulis tuleb testimisel läbida. Näiteks tuleb simuleerida UI mooduli testis JSON sõnumi saatmist ning seejärel kontrollida kas ViewModel-isse on andmed kohale jõudnud.

2.4. 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-i funktsioonide kasutamine logida nõ taidendavate kasutaja tegevuslogi kirjetena. Tegevlogi 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 kirjade kaupa lokaalsele logiteenusele, mis edastab tehnilise logi kesksele logi serverile ja error logi ning tegevuslogi eraldi seisvale HKSOS teenuse log rakendusele.