Sisukord

1. Üldnõuded

Käesolev dokument koos lisadega määratleb mittefunktsionaalsed nõuded Keskkonnaministeeriumi Infotehnoloogiakeskuse (KeMIT) halduses ja vastutusel olevatele infosüsteemidele, nende poolt osutatavatele teenustele ning seotud dokumentatsioonile.

Nõuded on kohustuslikud ka olemasolevate infosüsteemide lisaarendustele ja versiooniuuendustele mahus, mis on lisaarenduse ja versiooniuuenduste käigus võimalik.

Alternatiivsete tehnoloogiate, vahendite ja metoodikate valikul, mis käesolevas mittefunktsionaalsete nõuete dokumendis puuduvad, peab hankija rakendama järgmisi kriteeriume: sobivus olemasoleva infrastruktuuriga, platvormi tehnoloogilised omadused ning kompetentsi olemasolu ja kättesaadavus tööjõuturul. Juhul kui pakkuja leiab, et mõnda konkreetselt kohalduvat nõuet ei ole võimalik või otstarbekas täita, peab ta selle mittetäitmise fakti ja põhjenduse pakkumuses selgesõnaliselt välja tooma. Kõik erikokkulepped peavad olema fikseeritud tellijaga KeMIT kirjalikku taasesitamist võimaldaval viisil ning saama KeMITi arhitekti kinnituse.

2. Nõude kirjeldus koos märkusega

Nõude kirjeldus	Märkus
Vastavus standarditele ja seadusandlusele	
Lahendus peab vastama „Aadressiandmete süsteem" määruses kehtestatud aadressiandmete nõuetele.	Määrus on leitav https://www.riigiteataja.ee/akt/118062021032?leiaKehtiv
Lahenduses tuleb tegevusalade määramiseks kasutada Eesti Majanduse Tegevusalade Klassifikaatorit (EMTAK).	Klassifikaator on leitav: https://ariregister.rik.ee/est/emtak_search
Lahendus peab vastama „Avaliku teabe seaduses" kehtestatud teabe avalikustamise ja juurdepääsu võimaldamise nõuetele.	Seadus on leitav: https://www.riigiteataja.ee/akt/114032011019?leiaKehtiv
Lahendus peab vastama „Riigi infosüsteemi haldussüsteem" (edaspidi RIHA) esitatud nõuetele ja reeglitele.	Määrus on leitav: https://www.riigiteataja.ee/akt/12933746?leiaKehtiv

Nõude kirjeldus	Märkus
Lahendus peab minimaalselt vastama Eesti infoturbestandardi (E-ITS) etalonturbe kataloogis sätestatud lahendusele kohalduvate moodulite põhi- ja standardmeetmetele.	Standard on leitav: https://eits.ria.ee/
Loodav lahendus peab olema disainitud ja arendatud kooskõlas isikuandmete kaitse põhimõtetega (Data Protection by Design and by Default).	**IKÜM (GDPR):** o Euroopa Parlamendi ja nõukogu määrus (EL) 2016/679 füüsiliste isikute kaitse kohta isikuandmete töötlemisel ja selliste andmete vaba liikumise ning direktiivi 95/46/EÜ kehtetuks tunnistamise kohta. **Siseriiklik seadus:** o Isikuandmete kaitse seadus (IKS) – (Märkus: IKS võtab Eesti õigusesse üle ülaltoodud direktiivi 2016/680 sätted).
Lahenduses peavad X-tee teenused olema realiseeritud vastavalt RIA poolt kirjeldatud nõuetele.	Nõuded on leitavad: https://www.ria.ee/et/riigi-infosusteem/andmevahetuskiht-x-tee.html https://www.ria.ee/amet-uudised-ja-kontakt/uudised-pressikontakt/juhendid
Lahendus peab vastama "Infosüsteemide andmevahetuskiht" määruses kirjeldatud põhimõtetele.	Määrus on leitav: https://www.riigiteataja.ee/akt/127092016004?leiaKehtiv
ID-kaardi allkirjastamislahenduse kasutamisel eelistatakse SiGa veebiteenuse kasutamist.	Lisainfo leitav: https://github.com/open-eid/SiGa/wiki
Kuupäeva ja aja (kuupäev, kellaaeg, ajaintervall) talletamisel teksti kujul tuleb aluseks võtta ISO 8601 standardis kirjeldatud põhimõtted.	Põhimõtted on leitavad: https://www.w3.org/TR/NOTE-datetime
Veebirakenduse kasutajaliides peab vastama WCAG 2.2 tasemele AA	https://www.w3.org/TR/WCAG22
Veebipõhine kasutajaliides peab ühilduma standarditega HTML 5 ja CSS 3.	Valideerimiseks kasutatakse vastavaid validaatoreid: http://validator.w3.org/ Kui on tegu olemasoleva süsteemi edasiarendusega, siis tuleb järgida olemas olevat HTML ja CSS versiooni.
API nõuded ja standardid	
API dokumentatsioon peab olema OpenAPI Specification 3.1.0 või uuemas formaadis.	https://spec.openapis.org/oas/v3.1.0.html
Dokumentatsioon peab olema automaatselt genereeritud lähtekoodist (code annotations).	

Nõude kirjeldus	Märkus
HTTP meetodite (GET, POST, PUT, PATCH, DELETE) ja olekukoodide (2xx, 4xx, 5xx) kasutamine peab vastama RFC 9110 (HTTP Semantics) standardile.	https://www.rfc-editor.org/rfc/rfc9110
API veateated peavad järgima RFC 7807 (Problem Details for HTTP APIs) struktuuri.	Kohustuslikud väljad: type, title, status, detail. Näide: {"type": "about:blank", "title": "Not Found", "status": 404, "detail": "Kasutajat ei leitud"} Viide: https://www.rfc-editor.org/rfc/rfc7807
RESTful API peab vastama Richardson Maturity Model tasemele 2 või kõrgemale (ressursi-, mitte toimingupõhine).	https://martinfowler.com/articles/richardsonMaturityModel.html
RESTful nimetamisreeglid (naming conventions).	https://restfulapi.net/resource-naming • Ressursid URL-is mitmuses: /users, /orders, /invoices • Hierarhilised suhted: /users/{userId}/orders/{orderId} • Verbide vältimine URL-is (vale: /getUser, /createOrder) • Ainult väiketähed ja sidekriipsud: /user-profiles (mitte /userProfiles ega /user_profiles) • Erandid peavad olema põhjendatud ja dokumenteeritud.
API versioneerimine URL-i osas on kohustuslik.	/api/v1/users, /api/v2/users
Uue API versiooni loomisel peab eelmine versioon püsima vähemalt 12 kuud.	
Aegunud API versioonid peavad tagastama HTTP 410 Gone koos viitega uuele versioonile.	
CORS (Cross-Origin Resource Sharing) poliitika peab olema määratletud. Lubatud päritolud (origins) peavad olema eksplitsiitselt loetletud. Wildcard (*) kasutamine toodangukeskkonnas on keelatud.	https://www.rfc-editor.org/rfc/rfc6454.html
Arhitektuur	
Realiseeritud lahendus peab tellija poolt kinnitatud arhitektuurile vastavalt töötama tellija poolt nõutud funktsionaalsete ja mittefunktsionaalsete nõuete ulatuses.	

Nõude kirjeldus	Märkus
Kõik infosüsteemis kasutatavad komponendid peavad olema tuvastatavad, põhjendatud ja dokumenteeritud. Komponentidel peab olema teadaolev päritolu ja versioonihaldus.	Arhitektuuridokument XYZ
Kõik lahenduses kasutatavad kolmanda osapoole komponendid (nt välised süsteemid, teegid) peavad olema tuvastatavad ja dokumenteeritud.	
Liidesed väliste süsteemidega peavad olema standardsed (allutatud sarnastele reeglitele) ja liidestamise detailid peavad olema dokumenteeritud.	
Rakenduse liidesed peavad olema tõrkekindlad.	Lõppkasutaja peab saama jätkata rakenduse kasutamist ulatuses, mis on protsessiliselt võimalik. Süsteem peab tõrke korral võimalikult lühikese aja jooksul väljastama asjakohase veateate.
Infosüsteemide platvormid (nt rakendusserver, andmebaas) ja topoloogia peavad olema tellijaga kooskõlastatud enne reaalse tarkvaraarenduse algust.	Detailne infrastruktuur on kirjeldatud KliM haldusala tehnoloogilses profiilis.
Kõik arendamisel kasutatud komponendid peavad vastuvõtmise hetkel olema viimased stabiilsed versioonid.	

Nõude kirjeldus	Märkus
Infosüsteemi ülesehitus peab põhinema kolmekihilisel arhitektuuril, kus andmekiht, äriloogika ja esitluskiht on üksteisest lahutatud või Äriloogika kiht peab olema täies ulatuses tarbitav masinloetava API kaudu (REST/OpenAPI 3.0+ või GraphQL). Rakenduse arendamisel ja juurutamisel tuleb järgida 12-Factor App põhimõtteid, tagades protsesside olekuta (stateless) toimimise ja konfiguratsiooni lugemise keskkonnamuutujatest.	**Kihtide suhtlus ja vastutus** 1. Esitluskiht on andmekihist täielikult isoleeritud ja suhtleb äriloogikaga AINULT API kaudu. 2. Otsepäringud esitluskihist andmebaasi ega välistele süsteemidele (sh X-tee) on keelatud. 3. Esitluskiht on AINULT andmete esitaja, mitte töötleja - kogu andmetöötlus (filtreerimine, transformeerimine, agregeerimine, arvutused, valideerimised) toimub backend'is. 4. Backend tagastab esitluskihile valmis andmestruktuurid AS-IS kuvamiseks a. Näide: kui X-tee päring tagastab 50 välja, millest UI vajab 5 välja teises struktuuris, teeb backend kogu transformatsiooni. **Sessioonihaldus ja autentimine:** 1. Autentimine põhineb JWT (JSON Web Token) standardil (RFC 7519, RFC 9068) 2. Rakendusserveri protsessid ei tohi hoida püsivat olekut - JWT on stateless. 3. Kui on vaja serveri poolset olekut, kasutada välist andmehoidlat (Redis või andmebaas). **Viited** Kolmekihiline arhitektuur Masinloetav API põhimõtted 12-Factor App põhimõtted
Infosüsteem peab olema üles ehitatud nii, et eessüsteemid (inglise k front end) ja tagasüsteemid (inglise k back end) on arhitektuuriliselt selgelt lahutatud.	
Andmebaasid ja rakendused peavad kasutama UTF-8 kodeeringut ja UTC aega, sealhulgas logimine.	
Andmebaasi objektide nimed peavad olema sisulised ja andma aimu nende otstarbest.	
Lahendused peavad olema projekteeritud laiendatavana ja edasi arendatavana.	

Nõude kirjeldus	Märkus
Komponendid peavad olema sõltumatud ja taaskasutatavad.	Tuleb kasutada KEMITi olemasolevaid komponente (nt RabbitMQ, GEOSERVER jne.). Detailne infrastruktuur ja komponendid on kirjeldatud: KliM haldusala tehnoloogilises profiilis. Erandjuhud tuleb kooskõlastada tellijaga.
Rakendust peab saama liigutada ilma ümberprogrammeerimiseta erinevate domeenide, domeenisaitide ja keskkondade vahel.	
Rakendusel peab olema haldusliides.	Haldustoimingute tegemine otse andmebaasis peab olema viidud miinimumini. Peakasutajal peab olema selge ülevaade kasutajate õigustest.
Kui rakendused saadavad e-kirju, siis peavad nad selleks kasutama välist e-posti serverit.	
Rakendus peab kasutama keskkonnamuutujaid.	
Arhitektuur peab olema modulaarne, teenustepõhine.	
Rakendus peab pakkuma konfiguratsiooni, mis piirab ebaõnnestunud sisselogimiste arvu määratud ajavahemikus	Muudatusi (logimiste arv, ajaühik) peab saama seadistada konfiguratsiooniga keskkonnamuutujate abil.
Klientrakendus ei tohi teostada otsepöördust andmebaasi poole.	Peab kasutama rakendusserverit või adapterit.
Rakenduse failid, mis ei tohi olla kasutajale nähtavad, peavad olema kaitstud (rakenduse kasutajale mittekättesaadavates) kaustades.	
Sorteerimisreeglistik peab vastama eesti tähestikule, tõstutundlikkus peab olema väljalülitatud.	
Lihtsamatele päringutele (nt ühe konkreetse andmeobjekti otsing) peab loodav lahendus vastama maksimaalselt 2 sekundi jooksul. Keerulisemate päringute (nt nimekirja filtreerimine) puhul on ajaline piirang 5 sekundit.	

Nõude kirjeldus	Märkus
Loodav lahendus peab võimaldama serveri poolt lõppkasutajale tagastatavate andmeobjektide arvu piirangut ja/või mahukate andmekomplektide lehekülgjaotust (inglise k pagination või lazy load).	Eelistatud: RFC 5988 Link päis koos rel="next", rel="prev", rel="first", rel="last". Alternatiiv: offset/limit või cursor-based pagination. Vastus peab sisaldama metainfot: kokku kirjeid, lehekülgede arv, praegune lehekülg. Viide: RFC 5988
Kogu filtreerimis- ja töötlemisloogika peab olema backend'is.	
Virtuaalmasinasse paigaldatava rakenduse arendamisel peavad arvestama platvormi turvanõuetega.	Nõuete standart on leitav: https://en.wikipedia.org/wiki/Security-Enhanced_Linux KEMITis on vaikimisi kasutusel järgmised turbemoodulid: SELinux.
Kui rakenduse toimimiseks on vajalik autentimine, tuleb selleks kasutada RIA TARA.	https://www.ria.ee/et/riigi-infosusteem/eid/partnerile.html, https://e-gov.github.io/TARA-Doku/
Turvalisus, sh infoturve	
Arendatav rakendus peab vastama üle andmise hetkel OWASP ASVS 4, tase 2-le.	
Rakenduse autentimise jõustamine peab toimuma serveri poolel, põhinema JWT-l (JSON Web Token) vastavalt RFC 7519, RFC 9068, ning saama alguse kasutaja tuvastamisest TARA kaudu.	
Ebaõnnestunud autentimise korral ei tohi rakendus avaldada infot süsteemi sisemise loogika, kasutaja või konfiguratsiooni kohta.	
Rakendus (sh konteineris käitatav) ei tohi töötamiseks vajada root/administraatori õigusi.	
Autentimist võimaldav informatsioon (nt autentimissaladused, API võtmed, salasõnad) ei tohi sisalduda lähtekoodis.	
Rakenduse kasutajale kuvatavad URL-id ei tohi sisaldada isikuandmeid.	

Nõude kirjeldus	Märkus
Rakenduse autoriseerimine peab olema rollipõhine (RBAC). Kõik API endpoint'id peavad kontrollima kasutaja rolli/õigust. Õiguste haldus jms peab olema dokumenteeritud.	
Rakendusel peab olema konfigureeritav kasutajaseansi aegumise aeg.	
Süsteemist väljumine peab toimuma kasutajale üheselt arusaadaval ja turvalisel viisil. Seansist väljumine toimub kahel viisil: kasutaja seansi lõpetab süsteem, sest seanss on olnud pikem kui süsteemile seadistatud vaikimisi limiit või kasutaja lõpetab seansi omal soovil.	Rakendus peab võimaldama kasutajale nähtavat ja turvalist väljalogimise funktsiooni. Seanss tuleb automaatselt lõpetada, kui see ületab süsteemis määratud aegumistähtaega.
Süsteem peab teavitama kasutajat sessiooni lõppemisest.	
Kasutajal peab olema igal süsteemi kasutamise ajahetkel võimalik seanss omal soovil lõpetada.	
Iga eduka süsteemi sisselogimise (autentimise) korral tuleb alati luua unikaalne seansi identifikaator (inglise k session ID).	
Seansi identifikaator ei tohi kajastuda ressursilokaatoris (URL-is), veateadetes ega logides.	
Seansi identifikaator peab olema piisava pikkusega, juhuslik ja unikaalne kogu aktiivse seansi jooksul.	Rakenduse seansihaldus peab kasutama identifikaatoreid, mis on vastupidavad ennustus- ja jõurünnetele (brute-force). Identifikaator peab olema piisavalt pikk (soovitavalt min 128 bitti) ja genereeritud turvalise algoritmi abil, välistamaks kokkulangevusi teiste aktiivsete seanssidega. • **Vältida UUID v1**: see sisaldab ajatemplit ja MAC-aadressi, mis võib olla turvaoht. • **UUID v4 on sobiv**: kasutab juhuslikke bitte, aga spetsiaalsed session ID generaatorid on eelistatavad. • **Lisada täiendavad kontrollid**: IP-aadress, User-Agent, aegumisaeg. • **Vahetada ID-d sensitiivsete toimingute järel**: näiteks peale sisselogimist.

Nõude kirjeldus	Märkus
Rakendus ja selle komponendid peavad võimaldama keskkondade lahusust (nt arendus-, test- ja toodangu keskkond). Toodanguandmeid ei tohi kasutada testkeskkonnas.	
Andmebaasis olevate rakenduste kontod peavad omama ainult minimaalselt rakenduse tööks vajalikke õigusi.	
Kõik autentimise privaatandmed ning isiku- ja muud tundlikud andmed peavad olema krüpteeritud nii salvestamisel kui edastamisel, kasutades andmete kaitsevajadusest tulenevalt sobivaid ja dokumenteeritud krüptoalgoritme.	
Krüptoalgoritmide ja räsifunktsioonide kasutamisel tuleb järgida RIA veebilehel avaldatud krüptograafiliste algoritmide elutsükli uuringu värskeimas versioonis toodud soovitusi ja põhimõtteid.	Lisainfo: https://www.ria.ee/amet-uudised-ja-kontakt/uudised-pressikontakt/uuringud-ja-analuusid#kruptouuringud
Rakenduses peab olema tagatud võimekus välja vahetada aegunud ja ebaturvalisi krüptoalgoritme.	
Kõik võtmed ja salasõnad peavad olema asendatavad ning need tuleb toodangu keskkonna installatsiooni ajal luua või asendada.	
Võrguliikluse krüpteerimiseks peab olema TLS 1.3 valmidus.	
Tarkvara iga versioon peab läbima koodikvaliteedikontrolli süsteemi SonarQube nii, et pole kõrgeid ja kriitilisi tüüpi vigu.	
Kasutatavas tarkvaras ei tohi paigalduse ajal esineda teadaolevaid kõrgeid ja kriitilisi turbenõrkusi. Selle jälgimiseks liidestatakse Giti projekt KeMITi Dependency Track teenusega, ja arenduste tarnimisel skaneeritakse ehitamiseks kasutatud tarkvaralisi komponente (raport nähtav 1-2 tunnise viitega ka arendajale).	Dependency Track teenus on leitav: https://dependencytrack.sise.envir.ee/. Teadaolevate nõrkuste allikad: NVD, GitHub Advisory Database, OSV, Sonatype OSS Index, VulnDB.
Ligipääs rakendusele ja andmetele peab olema võimalik ainult dokumenteeritud liideste ja autentimisprotseduuride kaudu. Varjatud või dokumenteerimata ligipääsuteid ei tohi olla.	

Nõude kirjeldus	Märkus
Kui rakendus kasutab brauseri küpsiseid (inglise k session cookie) või muid tehnoloogiaid (nt local storage jm), mis salvestavad kasutaja arvutisse informatsiooni, tuleb kasutajat sellest eelnevalt teavitada.	
Rakendus ei tohi teostada X-tee päringut otse kasutaja arvutist.	
Rakendusserver ja andmebaas peavad olema võimelised töötama eraldi serveritel või konteineritel.	
Veebirakenduse kõik viited failidele ja kataloogidele peavad olema ilma absoluutse failiteeta.	
Lahendus peab minimaalselt vastama Eesti infoturbestandardi (E-ITS) etalonturbe kataloogis sätestatud lahendusele kohalduvate moodulite põhi- ja standardmeetmetele. Meetmete rakendamine peab olema kooskõlas E-ITS rakendusjuhistega.	Turvameetmed E-ITS rakendamisel - E-ITS portaalis. Kõrgmeetmetele vastavuse vajadus sätestatakse lahenduse tellimisel.
Rakendus peab logima kõikide kasutajate (sh administraatorite ja haldurite) poolt tehtud andmete vaatamised, loomised, muutmised ja kustutamised.	
Välistele kasutajatele mõeldud veebilehega rakendused peavad olema kaitstud keelatud päringute eest.	
Kui rakendus võimaldab mitteautenditud kasutajal edastada andmeid, tuleb need andmed puhastada XSS-filtriga.	
Kui on nõutud andmete jälgimise rakendamine, tuleb selleks kasutada RIA poolt pakutavat andmejälgijat.	
Kõik veebiteenused peavad sisaldama korrektset robots.txt faili, mis määratleb otsingurobotite ligipääsu reeglid. Otsingurobotite juurdepääs veebisisule on robotikeelu protokolliga (Robot Exclusion Protocol) kitsendatud.	robots.txt peab asuma teenuse juurkataloogis (https://<teenuse_domeen>/robots.txt).
Juhul, kui rakendus loob faile või võimaldab failide üles laadimist, tuleb failide salvestamiseks kasutada S3 tehnoloogiat.	

Nõude kirjeldus	Märkus
Lähtekood	
Kõik rakenduse versioonid, alates arenduse algfaasist kuni toodangusse viimiseni, peavad olema salvestatud KEMIT-i koodihoidlasse. See tagab, et igal hetkel on võimalik juurdepääs rakenduse lähtekoodile, ka arendusprotsessi käigus.	Lähtekoodis ei tohi sisalduda ligipääsupiiranguga infot (nt salasõnad, privaatvõtmed jmt), kuna peab arvestama, et lähtekood viiakse avalikku koodivaramusse.
Koodihoidla failide vaikimisi kodeering on UTF-8 ilma BOM signatuurita.	Välja arvatud juhtudel, kui tehnilises kirjelduses on tehnoloogiliste eripärade või muude piirangute tõttu sätestatud teisiti.
Kõik koodihoidla failid kasutavad LF (Line Feed) realõppe, mis on Unix/Linux standard.	
Koodihoidlasse üleslaaditud lähtekood peab olema kompileeritav ilma muudatusteta.	
Koodihoidlasse üleslaaditud lähtekood peab olema täielik ehk sellest saab koostada ja paigaldada täisfunktsionaalse rakenduse.	
Koodihoidlasse tuleb korraga laadida kõik iga muudatusega seotud materjalid.	Enne materjalide üleslaadimist tuleb koodihoidlast võtta viimane uuendatud koodi seis.
Materjalide üleslaadimisega peab kaasnema sisuline üleslaetavat materjali kirjeldav kommentaar.	
Rakenduse lähtekood peab olema kommenteeritud detailsusega, mis võimaldab erialast ettevalmistust omaval tarkvaraarendajal teostada süsteemi edasiarendust.	
Rakenduse lähtekood, kommentaarid, muutujate, tüüpide ja funktsioonide nimed peavad olema inglise keeles, sisulised ja andma aimu nende otstarbest.	
Koodis kasutatavad konstandid ja lühendid tuleb kirjutada suurtähtedega.	
Rakenduse lähtekood ei tohi sisaldada pöördumispunktide aadresse, paroole, sertifikaate ega võtmeid (ka siis kui need on koodist väljakommenteeritud).	

Nõude kirjeldus	Märkus
Rakenduse lähtekoodis peab olema võimalik tuvastada muudatuse teinud konkreetne füüsiline isik.	
Lähtekood peab olema kompileeritav tellija poolt määratletud arenduskomplekti ja versioonidega.	Tellija poolt aktsepteeritud Java ehitustööriistad on Maven ja Gradle. Eelistatud on React.js, kuid teiste raamistike kasutamine tuleb Tellijaga eelnevalt kooskõlastada.
Kasutuses mitteolev kood tuleb rakendusest eemaldada.	
Tarkvara versioonimine	
Tarkvara versiooninumbrid peavad vastama semantilisele versioneerimisele.	Standard leitav: https://semver.org - MAJOR: murrangulised muudatused (breaking changes). - MINOR: uus funktsionaalsus (tagasiühilduv). - PATCH: veaparandused.
Tarkvara publitseerimisel kasutatakse eelväljalaske versioone.	1.0.0-alpha.1, 1.0.0-beta.2, 1.0.0-rc.1
Rakenduse lähtekoodis on kohustuslik pidada muudatuste logi faili (CHANGELOG.md), mis järgib „Keep a Changelog“ standardit. Kõik tarkvara väljalasked (*releases*) peavad olema versioonihalduses tähistatud (*git tags*) ning versiooninumbrid peavad vastama Semantic Versioning standardile.	Keep a Changelog 1.1.0 standard. Standard on leitav - https://keepachangelog.com/en/1.1.0
Iga tarkvara väljalase tähistatakse Git tag'iga.	Formaat: vX.Y.Z, näiteks v2.3.1
API versioon (URL-is /api/v1/) ei pea vastama tarkvara versioonile.	
Andmebaas	
Andmebaasi tabelid ja väljad peavad olema kommenteeritud. Kommentaarid peavad olema inglise keeles ja sisulised.	
Andmebaasi väljapikkused peavad olema väljendatud sümbolites (tähemärkide arv).	
Andmebaasi objektide nimetused peavad olema inglise keeles ning andma selget aimu otstarbest (va ajutised rakenduse poolt genereeritavad tabelid nn temp).	

Nõude kirjeldus	Märkus
Andmebaasi tabelite nimetused tohivad sisaldada ladina tähestiku väiketähti „a-z", numbreid „0-9", alakriipsu "_". Andmebaasi objekti nimetus ei tohi alata numbriga.	
Igas andmebaasi tabelis peab olema defineeritud primaarvõti (inglise k Primary Key).	
Andmebaasi objektide loomiseks tuleb kasutada andmebaasi migratsioonivahendeid.	KeMIT poolt heaks kiidetud migratsioonivahenditeks on Liquibase ja Flyway.
Igas schema-põhises kataloogis peab olema ühine andmebaasi muudatuste paigaldamise skript, mis järjest käivitab vajalikke käske õiges järjekorras.	
Ühest andmetabelist teise viitamisel tuleb kasutada võõrvõtit (inglise k Foreign Key). Võõrvõtme nimi peab seostuma tabeli ja väljaga, millele see viitab.	
Kõik andmebaasi võõrvõtmed peavad olema indekseeritud, et tagada päringute efektiivsus.	
Infosüsteemi osades, kus töödeldavate andmete tervikluse nõue on kõrge (E-ITS osakaal I2 või kõrgem), on kohustuslik realiseerida andmekirjete versioneerimine, et tagada andmete muudatuste täielik jälgitavus ja taastevõime.	
Kui rakenduse versioon nõuab andmebaasi muudatusi, peavad üleantava koodiga kaasas olema andmebaasi paigalduse migratsiooniskriptid.	
Logimine ja monitooring	
Süsteemi muudatused ja rakenduse ning kasutajate tegevused logitakse seostatuna muudatuse/tegevuse teostanud konkreetse füüsilise isiku ja tema rolliga.	
Logiteadete sisu peab olema kirjutatud inglise keeles.	
Logidesse ei tohi salvestada tundlikke isikuandmeid (nt paroole, tokeneid jms).	

Nõude kirjeldus	Märkus
Logimine peab olema konfigureeritav ning kasutada tuleb standardseid logiformaate, et võimaldada hilisem logianalüsaatorite kasutamine.	Eelistatud on JSON ja vastavalt ECS standardile https://www.elastic.co/guide/en/ecs/current/index.html
Rakenduse logid tuleb edastada Elasticu ECS standardile vastavas formaadis.	ECS väljade viide on leitav: https://www.elastic.co/docs/reference/ecs/ecs-field-reference
Samasugused veateated logitakse maha eksponentsiaalselt. Logimine peab olema optimeeritud.	Informatsiooni dubleerimist logides tuleb vältida, kui ole nõutud teisiti.
Andmebaasi logidest saadetakse reaalajas koopia failisüsteemi logisse, mis peab sisaldama ka logimisfunktsionaalsuse aktiveerimise ja deaktiveerimise infot (nt aeg, kasutaja jm).	
Failisüsteemi logimisel peavad logid olema katalogiseeritud, üldlevinud faililaiendiga (nt .log, .txt, .xml) ja roteeruvad.	
Rakendus ei tohi väljastada kasutajale veateateid või aktiivsusjälgi, mis sisaldavad seansi identifikaatorit või isikuandmeid.	
Sisselogimise mehhanismid peavad olema võimelised logima nii õnnestunud, kui ka ebaõnnestunud sisselogimise katseid.	
Rakendus peab omama sisemist meetrikat ja infot sündmuste kohta ning jagama seda vastavalt prometheus standardile.	Juhul kui rakendus on arendatud Java Spring Boot raamistikus, tuleb kasutada **Spring Boot Actuator** moodulit koos **Micrometer Prometheus** registriga ja jätta vaike-seadistused (defaults), välja arvatud turvapiirangud. Viide: Observability
Rakenduse ärilogi(auditlog) andmebaas peab olema sõltumatu rakenduse tööbaasist.	
Kõik logid peavad olema kaitstud rakenduse kasutaja poolse lubamatu ligipääsu ja muutmise eest.	

Nõude kirjeldus	Märkus
Logimise tasandid on: • DEBUG – arendamise etapis süsteemi olekut kirjeldav informatsioon (kasutatakse ainult arenduskeskkonnas) + Live keskkonnas debug tasand erandkorras kooskõlastatult tehnoloogia valdkonnaga. • INFO – kasutaja päringute informatsioon, kasutaja infoteated (nt „Andmed salvestatud", „Andmed muudetud"). • WARNING – kasutajale kuvatav valest sisendist tekkiv oodatud viga (nt „Vigaselt sisestatud andmed"). • ERROR – süsteemi vead, mis tekivad kasutaja sisendist (nt vigase andmebaasipäringu tekkimine). • FATAL – rakenduse toimimise kriitilised vead, mis takistavad rakenduse tavapärast toimimist (nt vigane konfiguratsioon). • TRACE - rakenduse koodi käigu jälitamiseks (kasutakse ainult arenduskeskkonnas ja tõrkeotsingul) + Live keskkonnas trace tasand erandkorras kooskõlastatult tehnoloogia valdkonnaga.	
2.8 Konfiguratsioon	
Kõik komponendid peavad olema ajakohase turvakonfiguratsiooniga ja versiooniga.	
Komponentide vaheline (nt rakendusserveri ja andmebaasi serveri) suhtlus peab olema krüpteeritud.	
Komponentide vahelise ühenduse jaoks tuleb kasutada minimaalselt vajalike õigustega kontot.	
Rakenduste omavahelisel suhtlemisel tuvastatakse üksteist oAuth2 abil.	

Nõude kirjeldus	Märkus
Rakenduse konfiguratsiooniparameetrid ei tohi muutmisel vajada uuesti kokku kompileerimist. Konfiguratsiooniparameetrid muudetakse keskkonnamuutujate abil ja peale rakenduse restarti on uus parameeter sisseloetud.	
Konfiguratsiooniparameetrite nimed peavad olema ingliskeelsed ja sisulised.	Kui sisulist nime ei ole võimalik kasutada, siis peab kasutatava nime kõrval olema seletus.
Samasisulisi konfiguratsiooni parameetreid ei tohi korduvalt kasutada.	

Nõude kirjeldus	Märkus
Rakendusi jooksutatakse konteinerites ning need peavad vastama vähemalt alljärgnevatele turvanõuetele. • **Multi-stage build** Konteineri kujutise loomisel peab kasutama multi-stage build'i, et eraldada ehituse ja käitamise faasid ning vähendada lõppkujutise mahtu. • **Mitte-root kasutaja** Konteiner peab käivituma mitte-root kasutajana, mis on määratud USER direktiiviga Dockerfile'is. • **Minimalistlik baaskujutis** Konteineri baaskujutis peab olema võimalikult minimalistlik (eelistatud: Distroless, Alpine või slim variandid). • **Haavatavuste skaneerimine** Konteineri kujutised peavad läbima haavatavuste skaneerimise CI/CD pipeline'is (nt Trivy või Grype) • **Lubatud haavatavuste tase** Konteineri kujutises ei tohi olla haavatavusi, mis ületavad taseme KESKMINE (MEDIUM). • **Saladuste käsitlemine** Saladused ei tohi olla konteineri kujutise kihtides ega Dockerfile'is. • **Kujutise allkirjastamine** Konteineri kujutis soovitud viis on allkirjastada (nt Cosign), et tagada päritolu ja terviklikkus. • **Dockerfile kinnitamine** Dockerfile'i sisu peab saama KeMIT tehnoloogiavaldkonna kinnituse enne tootmiskasutusse lubamist.	Konteinerite platvormiks on Docker.

Nõude kirjeldus	Märkus
Rakendused paigaldatakse **Kubernetes klastrisse** ja peavad vastama allpooltoodud nõuetele. • **Horisontaalne skaleerimine** Rakendus peab toetama horisontaalset skaleerimist ning klastris peab olema konfigureeritud HorizontalPodAutoscaler. • **Sessioonihaldus** Rakenduse autentimine peab kasutama JWT-tokensüsteemi, mis tagab sessioonihalduse sõltumatult pod'ide arvust ja olekust. • **Elusoleku ja valmisoleku kontrollid** Rakendus peab defineerima elusoleku (liveness) ja valmisoleku (readiness) kontrollid, mis on kättesaadavad endpoint'idena /health/live ja /health/ready. • **Ressursipiirangud** Iga pod peab omama määratud CPU ja mälu ressursipiiranguid (requests ja limits), et vältida ressursi ülekasutust. • **Graatsiline sulgemine** Rakendus peab korrektselt käsitlema SIGTERM signaali ning lõpetama pooleliolevad päringud maksimaalselt 30 sekundi jooksul enne sulgemist. • **Konfiguratsiooni ja saladuste haldus** Rakenduse konfiguratsioon peab olema hallatud ConfigMap-ide kaudu ning saladused peavad olema talletatud Secret-ites.	
Kasutajaliides	
Kasutajaliidese kõik disainiotsused peavad olema tellijaga kooskõlastatud.	
Kasutajaliideste realiseerimiseks kasutatavad UI raamistikud ja komponendid tuleb tellijaga eelnevalt kooskõlastada.	

Nõude kirjeldus	Märkus
Kasutajaliidese arenduse tehnoloogia määrab tehnoloogiline profiil.	
Kasutajaliidese loomisel tuleb kasutada TEDI disainisüsteemi komponente.	Tedi komponendid on leitavad: https://tedi.tehik.ee/
Ühe veebirakenduse realiseerimiseks kasutatavate erinevate UI raamistike ja komponentide arv peab olema minimaalne.	
Rakenduse kasutajaliides peab olema täies ulatuses eestikeelne, sealhulgas kõik menüüd, navigeerimiselemendid, abitekstid, vormide väljad ja süsteemiteated (sh veateated ja kinnitused). Kasutatav keel peab olema grammatiliselt korrektne ja terminoloogiliselt ühtne.	
Peale kasutaja sisselogimist rakendusse kuvatakse sisseloginud kasutaja nimi ja rolliinfo. Kui ühele kasutajatunnusele on määratud mitu rolli, kuvatakse kasutajale rollivalik.	
Kasutajaliides peab alati küsima kinnitust andmete kustutamise ja massmuutmise kohta.	
Kasutajal peab olema võimalik rakenduses tegevus pooleli jätta ning hiljem jätkata samast kohast, ilma et oleks kohustus alustada algusest.	
Kui kasutatakse brauserit, mis ei ole toetatud, peab kasutaja tõrke korral saama vastavasisulise teavituse.	
Veebilehitseja navigatsiooninupud peavad käituma rakenduses analoogiliselt klassikalise veebilehitsemisega (nt veebilehitseja „Tagasi" nupp navigeerib kasutaja eelmisele kuvatud lehele).	
Kasutajaliideses navigeerimine peab lähtuma äriloogikast ning võimaldama andmete sisestamist ja kasutamist ainult klaviatuuri abil.	
Kasutajaliidese toiminguni navigeerimiseks peab kehtima kolme klõpsu printsiip, väljalogimiseks ühe klõpsu printsiip.	

Nõude kirjeldus	Märkus
Interaktiivse vormi puhul ei tohi lehe värskendamisega tegevust korrata (nt faili teistkordselt laadida, saadetud andmeid uuesti saata).	
Kui vorm koosneb mahukatest andmeväljadest, peab kasutajaliides eeldefineeritud ajavahemike järel salvestama välja sisu, et vältida sisestatud andmete kadumist.	Mahukad andmeväljad täpsustatakse/lepitakse kokku detailanalüüsi käigus arenduse teostamisel. Salvestamine puudutab vaid sisestamise vormi kohta.
Vormide puhul peab tellija poolt nimetatud väljal olles kuvama kasutajale juhised, mis kujul informatsiooni väljale sisestada tuleb.	
Andmete sisestamisel peab rakendus kontrollima nii esitluskihis kui tagasüsteemis (backend), et sisestatud andmed vastavad välja tüübile.	
Rakendus peab võimalikult palju informatsiooni automaatselt eeltäitma (nt kirje sisestamise kuupäev).	
Kasutajaliides peab olema tõlgitav teise keelde ilma rakenduse lähtekoodi muutmata.	
Vältida tuleb kuvasid, mis eeldavad info lugemiseks kerimist paremale-vasakule.	TEDi raamistik.
Kui rakenduses teostatav päring on pikem kui kolm sekundit, peab kasutajat sellest visuaalselt teavitama (nt ekraanil on liivakella kujutis; kuvatakse teade, et päringut teostatakse).	
Rakenduse esilehel peab olema võimalus halduri poolt lisada kasutajale mõeldud teavitusi ja informatsiooni.	
Kasutajaliides peab teavitama kasutajat ette seansi aegumisest.	
Veateade peab olema kirjutatud lühidalt, selges ja lõppkasutajale arusaadavas keeles. Süsteemi rikete tehnilisi üksikasju ei tohi välja näidata.	
Veateade peab sisaldama probleemi kirjeldust, vea koodi ja lahendust või infot, mis juhendab kasutajat edaspidiseks vea vältimiseks.	

Nõude kirjeldus	Märkus
Süsteem peab asendama vaikimisi veateate lehekülje, kuid säilitama algse HTTP vastuskoodi.	
Testimine	
Rakenduse kõik üleantavad versioonid peavad olema enne tellijale üleandmist täies mahus testitud: testitakse kõiki funktsionaalseid ja mittefunktsionaalseid nõudeid. Tellija nõudmisel tuleb arendajal koos rakenduse üleandmisega esitada testitulemuste raport.	
Rakenduse igakordsel versiooni üleandmisel tellijale, peab kaasas olema skript analüüsi käigus kokkulepitud jõudlustestide teostamiseks.	
Jõudlustestid peavad olema läbiviidud vähemalt kahekordse eeldatava koormuse varuga.	
Rakenduse koormuse testimiseks tuleb luua testandmete kogum.	
Kriitilise tähtsusega funktsionaalsus oleks kaetud automaattestidega (ühiktestid, integratsioonitestid, E2E testid.	
Enne tellijale üleandmist ei tohi loodav rakendus sisaldada viimases kehtivas OWASP Top 10 nimekirjas kirjeldatud turvanõrkusi.	Testimistulemuste raport tuleb esitada tellijale rakenduse üleandmisel.
CI/CD ja DevOps praktikad	
Lähtekoodi kompileerimine peab toimuma automatiseeritud CI/CD protsessis, mis tagab reprodutseeritava ja jälgitava build'i. Build-protsess ei tohi sisaldada käsitsi tehtavaid samme.	
Build-protsess peab sisaldama staatilist koodianalüüsi (nt SonarQube), mis kinnitab, et turvavigu ning Blocker ja Critical taseme vigu ei tohi esineda	Nt SonarQube - Security, Blocker, Critical vigade puudumine.
Ühiktestide koodikate peab olema vähemalt 80%, kriitilise funktsionaalsuse puhul vähemalt 90%. Testid peavad jooksma automaatselt CI/CD protsessis.	

Nõude kirjeldus	Märkus
Build-protsess peab sisaldama turvaskanneerimist, mis tuvastab teadaolevad nõrkused sõltuvustes (nt Dependency Track) ja konteineripiltides (nt Trivy või Grype). Kõrge ja kriitilise taseme haavatavustega artefakte ei tohi tootmisesse paigaldada.	• Dependency Track teadaolevate nõrkuste tuvastamiseks. • Konteineri haavatavuste skaneerimine (Trivy/Grype).
Iga build peab olema automaatselt paigaldatav test- või arenduskeskkonda CI/CD pipeline'i kaudu.	
Igal paigaldusel peab olema võimalik teostada kiire tagasipööramine eelmisele stabiilsele versioonile mitte rohkem kui 15 minuti jooksul.	
Dokumentatsioon	
Kogu rakenduse dokumentatsioon peab olema kirjeldatud korrektses eesti keeles.	Erandiks võib olla kolmanda osapoole komponentide dokumentatsioon (dokumentatsioon, mis pole kirjutatud tellija jaoks). Erandina käsitletakse ka väliste partneritega seotud projektdokumentatsiooni. Kõik erandid tuleb kirjalikku taasesitamist võimaldaval viisil kooskõlastada tellijaga enne dokumentatsiooni koostamist.
Dokumentatsioon peab olema eesti keeles ja sisaldama versiooni numbrit, muutmise kuupäeva, autori nime ja olema koostatud selge struktuuriga ja peab olema piisavalt selge, et Tellija iseseisvalt suudab selle järgi toimida.	Iga dokumendi versiooni kõik uuendused (võrrelduna eelmise kehtinud versiooniga), peavad olema visuaalselt eristatavad.
Lahenduse dokumentatsioon peab sisaldama RIHA määrusest tulenevat kohustuslikku informatsiooni.	RIHA määrus on leitav: https://www.riigiteataja.ee/akt/13147268?leiaKehtiv
Lahenduse dokumentatsioon peab sisaldama E-ITS'is nõutud dokumente.	
Lahenduse dokumentatsioon peab sisaldama E-ITS'is nõutud dokumente.	Nõuded on leitavad (CON.8.M12 Detailne tarkvara dokumentatsioon): https://eits.ria.ee/et/versioon/2024/eits-poohidokumendid/etalonturbe-kataloog/con-kontseptsioonid-ja-metoodikad/con8-tarkvaraarendus/3-meetmed/33-standardmeetmed/con8m12-detailne-tarkvara-dokumentatsioon/
Kõik tellijale esitatud dokumendid peavad olema redigeeritavad enamlevinud redaktoritega (nt Microsoft Office).	

3. Dokumendi muudatuste register

1. Standardid ja seadusandlus (ISKE asendumine E-ITS-iga)

Kõige suurem põhimõtteline muudatus puudutab infoturbe raamistikku.

- 2024: Nõudis vastavust infosüsteemide turvameetmete süsteemile (ISKE) ja turbeastmele M.
- 2025: ISKE on asendatud Eesti infoturbestandardiga (E-ITS). Nõutakse vastavust E-ITS etalonturbe kataloogile ja rakendusele kohalduvatele moodulitele.
- Ligipääsetavus: WCAG nõue on tõstetud versioonilt 2.1 (2024) versioonile 2.2 (2025).

2. API ja liidestamine

2025. aasta versioonis on lisatud peatükk "API nõuded ja standardid".

- Dokumentatsioon: Nõutud OpenAPI Specification 3.1.0 (või uuem).
- Standardid: HTTP meetodid (RFC 9110) ja veateated peavad vastama RFC 7807 (Problem Details) struktuurile.
- Disain: Nõutud on Richardson Maturity Model tase 2 (ressursipõhine).
- Reeglid: Kehtestatud ranged nimekonventsioonid (mitmus, kebab-case), URL-i versioonimine on kohustuslik.
- Paginatsioon: Eelistatud on RFC 5988 (Link header) standard.

3. Arhitektuur ja kihilisus

- Lisandus valik 12-Factor App: 2025. aasta versioon nõuab eksplitsiitselt 12-Factor App põhimõtete järgimist.
- Eraldatus: Rõhutatakse, et Frontend ja Backend peavad olema täielikult eraldatud. Esitluskiht ei tohi teha otsepäringuid andmebaasi ega X-teesse.
- Autentimine: Sessioonihaldus peab põhinema JWT (JSON Web Token) standardil ja olema olekuta (stateless).
- Loogika asukoht: Kogu äriloogika, filtreerimine ja valideerimine peab toimuma backend'is, mitte frontend'is.

4. Turvalisus ja DevSecOps

- Tarneahela turvalisus: 2025 lisab nõude kasutada Dependency Track teenust (KEMITi Giti kaudu) teadaolevate nõrkuste tuvastamiseks.
- Koodianalüüs: Täpsustatud on SonarQube nõuded – ei tohi olla Security, Blocker ega Critical vigu (2024 oli üldsõnalisem).
- Testimine: 2025 määratleb konkreetsed numbrid – ühiktestide (Unit tests) koodikate peab olema minimaalselt 80% (kriitilisel osal 90%). 2024 seda ei täpsustanud.

5. Konteinerid ja Kubernetes (Docker)

- 2024 mainis Kubernetesesse paigaldamist, kuid 2025 läheb süvitsi Dockerfile'i nõuetesse:
- Turvalisus: Keelatud on root kasutaja, nõutud mitmeastmeline (multi-stage) build.
- Baaskujutis: Eelistatud on minimalistlikud (Distroless, Alpine).
- Skaneerimine: Kohustuslik haavatavuste skaneerimine (Trivy või Grype) CI/CD torudes.
- Elusoleku kontroll: Täpsustatud liveness ja readiness otspunktide nõuded.

6. Monitooring (Java spetsiifika)

- 2025 täiendus: Kui rakendus on Java Spring Boot raamistikus, on kohustuslik kasutada Spring Boot Actuator'it koos Micrometer Prometheus registriga.

Kokkuvõte

Peamine erinevus seisneb standardsuses ja detailsuses. Vana (2024) dokument kirjeldas mida on vaja (turvalisust, logimist, skaleerimist), kuid uus (2025) dokument kirjutab ette kuidas seda tehniliselt teha (mis RFC standardid, mis arhitektuurimustrid, mis konkreetsed turvaskannerid ja mõõdikud).

Olulisemad märksõnad üleminekul:
- Monoliit/Üldine -> Mikroteenused/API-kesksus (API ja eraldatuse reeglid).
- CI/CD rangus (Konkreetsed protsendid testkatvusele ja automaatsed turvakontrollid konteineritele).