RmIT üldised ristfunktsionaalsed, mittefunktsionaalsed ja tehnilised nõuded 2.9.0

Käesolev dokument määrab kvaliteedi-, rist- ja mittefunktsionaalsed nõuded RmIT tellitavatele või hallatavatele infosüsteemidele ning nende dokumentatsioonidele.

Kui mõnda nõuet pole otstarbekas täita, tuleb selle mittetäitmine kooskõlastada tellijaga hiljemalt enne teostuse algust.

Nõudeid tuleb järgida ka olemasolevate infosüsteemide versiooniuuendustel nii palju kui versiooniuuenduse käigus võimalik.

Nõude liik	Nr	Sisu	Täpsustused	Tehnoloogia spetsiifika
Üldine	1	Lahenduse X-tee teenused peavad vastama RIA nõuetele. Aluseks tuleb võtta tööde tellimise hetkel kehtiv versioon.	Kehtiv versioon nõuetest on kättesaadav RIA veebilehtelt.	X-tee
Üldine	2	Rakendus peab olema arendatud arvestades selle rakenduse poolt töödeldavatele andmetele määratud CIA (confidentiality/konfidentsiaalsuse, integrity/terviklikkuse, availability/käideldavuse) nõudeid.	Üldised CIA nõuded annab ette tarkvara tellija (E-ITS või ISO/IEC 27001 standardist juhinduva organisatsiooni puhul nõutud protsesside raames loodud infoturbe nõuded, seadusandlusest tulenevad nõuded, jne). Arendatavates rakendustes realiseeritavad kaitsemeetmed täpsustatakse analüüsi käigus arendaja ja tellija koostöös. Arvestada, et kui rakenduse osadel andmetel on kõrged CIA nõuded, ei tähenda see automaatselt, et selle rakenduse kõikidel andmetel on kõrged CIA nõuded. Näiteks, kui andmebaasi mõne tabeli read tuleb kõrge terviklikkuse nõude tõttu versioneerida, ei tähenda see seda, et tuleb versioneerida valimatult selle andmebaasi kõikide tabelite read.	
Üldine	4	Rakendused ja nende poolt genereeritud sisu peavad olema digiligipääsetavad, milleks tuleb järgida EN 301 549 standardit (Euroopa Liidus kehtestatud ligipääsetavuse nõudeid), sealhulgas WCAG (Web Content Accessibility Guidelines) juhiseid, vastavalt avaliku teabe seaduse paragrahvis 32 ja ministri määruses nr 20 "Veebilehe ja mobiilirakenduse ligipääsetavuse nõuded ning ligipääsetavust kirjeldava teabe avaldamise kord" kehtestatud ligipääsetavusnõuetele. Rakenduse kasutajaliides peab vastama vähemalt WCAG 2.1 tasemele AA.	Euroopa Liidu nõuded baseeruvad WCAG juhistel. Lisainfot: • https://digital-strategy.ec.europa.eu/en/policies/web-accessibility • https://eur-lex.europa.eu/eli/dir/2016/2102/oj • https://www.w3.org/TR/WCAG21/ • https://www.w3.org/WAI/WCAG21/Understanding/ • https://www.w3.org/WAI/test-evaluate/ Abimaterjale: • https://webaim.org/standards/wcag/checklist • https://ttja.ee/eraklient/tarbija-oigused/kaubandus-teenused/digiligipaasetavuse-tagamine • TTJA koolitusvideod, nt https://www.youtube.com/watch?v=B8cqmxdRKDg	

Üldine	6	E-tembeldamiseks tuleb kasutada olemasolevaid RmIT-i teenuseid.		
Üldine	7	Veebipõhine kasutajaliides peab ühilduma täielikult HTML ja CSS viimaste kehtivate standarditega ning kõikide nõutud veebilehitsejate versioonidega.	Kood peab olema valideeruv (https://jigsaw.w3.org/css-validator/). Vigu (error) ei tohi esineda. Hoiatuste korral tuleb nende parandamata jätmine kooskõlastada RmIT-ga. Lehitsejate versioonid tulenevad IT-profiilist.	Veeb
Üldine	8	Infosüsteem peab olema projekteeritud arvestades rakenduse käideldavuse taset.	Süsteemi hooldamine ja varundamine peab olema võimalik SLA-s lubatud katkestuste pikkust ja tihedust arvestades.	
Platvorm	9	Lahendus peab vastama RmIT IT-profiilile.		
Platvorm	10	Loodava või muudetava infosüsteemi platvorm (rakendusserver, andmebaas, kolmanda osapoole komponendid) ja topoloogia peavad olema enne reaalse arenduse algust RmIT hooldusosakonnaga kooskõlastatud.	Kasutuses olevate infosüsteemide platvormide ja topoloogiaga ning RmIT arengusuundadega arvestamine on vajalik selleks, et tulevased komponendid ühilduks olemasolevate süsteemidega (nt Oracle DB või RHEL konkreetse versiooniga). Platvormi muudatused ei pruugi olla alati jõudnud veel kehtivasse IT-profiili. Süsteemi jõudlus peab vastama kokkulepitud topoloogiale (k.a mälumaht ja samaaegsete baasiühenduste piirmäär), eelanalüüsi ja lähteülesande käigus välja toodud jõudlusnäitajatele.	
Platvorm	11	Rakenduse, andmebaasi ja kolmanda osapoole komponentide platvorm(id)/versioon(id) peavad olema sellised, mille eluea lõpp (EOL) pole teadaolevalt vähem kui 2 aasta pärast ning mis ei ole alles arenduse algusjärgus (alfa, beeta või snapshot staatuses). Samuti ei tohi kasutada komponentide taunitud (deprecated) funktsionaalsust.	Uue rakenduse loomisel või uue sõltuvuse kasutuselevõtmisel peab kasutama sõltuvuste (nt teekide, ka raamistike, platvormide) viimaseid versioone (mis ei ole alles arendusjärgus). Kui sõltuvusel on LTS (long term support) versioone, tuleb mitte LTS versioonid vahele jätta.	
Platvorm	12	Rakendusserver peab võimaldama töötamist andmebaasiserverist eraldi serveril.		
Platvorm	13	Kõik rakenduse komponendid peavad võimaldama töötamist kõrgkäideldavalt.	Tarnitud rakenduse komponente (kasutajaliides, teenused jne) ja funktsionaalsust saab kasutada aktiivklastris. Andmebaasi aktiivklasterdamise (active-active cluster) toetamine ei ole üldjuhul nõutud.	
Platvorm	14	Rakenduse kasutajasessioon ei tohi olla klastri node'i põhine.	Kasutaja peab saama edasi tegutseda ka siis kui tema sessioon mingil põhjusel teisele node'ile suundub.	

Platvorm	15	Rakendust peab saama ilma ümber programmeerimata liigutada erinevate domeenide ja domeeni saitide vahel.	Ei tohi kasutada absoluutseid URL-e.	
Platvorm	16	Rakendused peavad lähtuma pehme halvenemise (graceful degradation) ja stabiilsuse (robustness) põhimõttest. Väljuvatel päringutel peab olema määratud aegumine. Rakenduse sõltuvuse taastumisel peab ka rakendus automaatselt taastuma.	Rakenduse äriloogiliselt kriitilised liidesed peavad olema tõrkekindlad. Liidestatud (nii süsteemi sisese kui ka välise) komponendi tõrgete korral ei tohi rakendus hanguda, vaid peab väljastama mõistliku aja jooksul ajakohase veateate. Vajadusel tuleb töökindluse tõstmiseks kasutada asünkroonseid liideseid. Mõistlik aeg peab olema süsteemi konfiguratsioonist seadistatav. Rakendused peavad tulema toime juhuvigadega (transient error), näiteks peab rakendus ise taastuma, kui võrgu- või baasiühendus korraks ära kaob.	
Platvorm	17	Andmevahetus riigi infosüsteemi kuuluvate andmekogudega ja riigi infosüsteemi kuuluvate andmekogude vahel toimub läbi riigi infosüsteemi andmevahetuskihi X-tee (Avaliku teabe seaduse § 43 (9) lõige 5).		X-tee
Versioonihaldus ja paigaldamine	18	Paigaldusühik (näiteks war, docker image) peab olema muutmata kujul paigaldatav kõikidesse keskkondadesse. Rakenduse või rakendust sisaldava rakendusserveri käivitamisel peab saama käsurea parameetriga "confRoot" või keskkonnamuutujaga "CONF_ROOT" määrata failitee rakenduse configuratsioonifaile sisaldavale kataloogile. Rakendus loeb konfifailid kataloogist {{confRoot}}/{{appName}}, kus {{appName}} on teenuste registrisse registreeritav rakenduse nimi. {{appName}} kataloogi sisu peab olema dokumenteeritud rakenduse paigaldusjuhendis.	{{confRoot}}/{{appName}} on kataloog, kuhu on koondatud rakenduse konfiguratsioonifailid (nt application.properties). Ka logimise seadistus (nt logback.xml) peab olema etteantav selles kataloogis (rakenduse pakke väliselt).	
Platvorm	20	Rakenduse käivitus (saidi taaskäivitus, konfiguratsiooni muutmine vms) peab toimuma mõistliku aja jooksul.	Maksimaalselt 30 sek. Rakendus peab selle aja jooksul serveris running-staatusesse jõudma. Pikemad tegevused tuleb vajadusel teha peale seda.	
Platvorm	21	Andmebaas ja rakendus peavad kasutama UTF-8 kodeeringut.		

Platvorm	22	Failid tuleb katalogiseerida kokkulepitud tunnuste alusel.	Tunnused tuleb valida selliselt, et ühte kausta ei tekiks üle 1000 faili.	
Platvorm	24	Vigastele meiliaadressidele ei tohi kirju saata. Meiliaadress peab vastama RFC5322 ja/või RFC6854 standardile.	Enne saatma hakkamist tuleb teha formaadi kontroll isegi siis, kui andmete sisestusel seda juba tehakse.	
Konfiguratsioon	25	Konfiguratsiooniparameetrite nimed peavad olema sisulised. Kui see ei ole võimalik, siis peab kõrval olema seletus.	Näited: X-tee_Turvaserver, mitte XTTS; viitenumber, mitte vk_seb	
Konfiguratsioon	26	Konfiguratsioonifailid ei tohi olla lõppkasutajale nähtavad.		
Konfiguratsioon	27	Erinevaid sama tähendusega parameetreid ei tohi konfiguratsioonis eksisteerida.	Kõiki parameetreid tuleb konfiguratsioonis kirjeldada vaid ühel korral, mitte nii, et mitmes failis või faili lõigus kirjeldatakse samu parameetreid uuesti.	
Platvorm	28	Kliendi ja andmebaasi suhtlus peab toimuma läbi rakendusserveri.	Klientrakendus (sh mobiilirakendus) ei tohi pöörduda otse andmebaasi poole.	
Konfiguratsioon	29	Kõik keskkonnapõhised parameetrid peavad olema konfiguratsioonist seadistatavad.	IT-profiilis on välja toodud (mitte täielik) loetelu levinumatest parameetritest, mis peavad olema muudetavad.	
Platvorm	30	Rakendus ei tohi kasutada meetodeid, mis välistavad shared hostingu kasutamise.		
Platvorm	31	Uniform resource identifier (URI) pikkus ei tohi ületada ühegi toetatava brauseri maksimaalset lubatud väärtust.		
Turva	32	Erinevat tüüpi kasutajatele mõeldud teenused peavad olema üksteisest eraldatavad loogiliselt (samas paigaldatavas komponendis URLi path'i alguse järgi) või füüsiliselt (eraldi komponentidena paigaldatavana). Teenuste grupeerimine ning eraldamise viisid tuleb enne arenduse algust tellijaga kokku leppida.	Teenuste grupeerimisel ning loogilise ja füüsilise eraldatuse vahel valides tuleb arvestada, et teenuste käideldavuse, seadistuse ja turva nõuded võivad olla erinevad. Reeglina peavad rakenduse inimkasutajatele mõeldud teenused olema füüsiliselt eraldatavad masinatele mõeldud teenustest. Inimkasutajate hulgas tuleb eristada väliskasutajaid (nt tavakodanikud ja firmad), sisekasutajaid (nt ametnikud) ning administraatoreid.	
Platvorm	33	Rakendus ei tohi teostada X-tee päringut otse kasutajaarvutist.	Kasutajaarvutitest otse X-tee päringute tegemine on arvutivõrgu tasemel suletud.	X-tee

Platvorm	34	Arendus peab olema orienteeritud toodangukeskkonnas toimimiseks.	Toodangukeskkonna rakendus ei tohi sisaldada osiseid, mis on toodangukeskkonnas ebavajalikud või segavad (näiteks mõeldud testimiseks testkeskkonnas, arendusabiks arenduskeskkonnas) või annavad rakenduste lõppkasutajatele üleliigset infot süsteemi toimimise kohta (nt koodi kommentaarid JavaScriptis).	
Platvorm	35	Rakendus peab serverist kustutama kõik ajutised failid koheselt, kui neid enam ei kasutata.		
Seire	36	Rakendusel peab olema seire otspunkt (URL), mis annab rakenduse staatuse ja versiooni, ning eraldi seire otspunkt iga olulise sõltuvuse staatuse testimiseks. Seire otspunktid asuvad TEENUSE_URL + "/info" otspunkti all, kui ei ole kokku lepitud teisiti.	Seire ressursid (HTTP ressursid) on minimalistlikud (nii suuruse kui serveri koormuse osas) ja peavad sisu uuendama (mitte vahemälust võtma) otspunkti poole pöördumisel. Seire otspunktide URL-i /info* kuju ei tohi muul otstarbel kasutada. Erandid seireteenustes ja formaadis lepitakse kokku tellijaga. Detailsem seire otspunktide kirjeldus: https://confluence.rmit.ee/x/WyKBBQ	
Platvorm	37	Rakenduse kasutamine ei tohi nõuda kasutaja arvutisse lisatarkvara paigaldamist.		
Platvorm	38	Kõik java veebirakenduse sessiooni salvestuvad objektid peavad implementeerima java.io.Serializable liidest.	Vajadusel tuleb kasutada transient märgendit.	Java
Kood	39	Rakendus tohib kasutada ainult suhtelisi aadresse/teid (relative URL/path).	Kui on vajalik täispikk tee, siis tuleb see kirjeldada konfiguratsiooniparameetrina ühes kohas ja sealt seda infot kasutada.	
Platvorm	40	Rakendusel on keelatud kirjutada rakendusserveri standardväljunditesse.	RmIT keskkondade (või vaikimisi) konfiguratsioon peab tarnitaval rakendusel suunama kogu logi roteeruvatesse failidesse.	
Platvorm	41	Kasutajate autentimiseks tuleb kasutada haldusalas kasutuses olevaid teeke või rakendust.	Kasutuses olevad lahendused on kirjeldatud IT profiilis.	
Kasutajaliides	42	Kasutajaliidese vaate kuvamiseks vajalike ressursside (nt HTML koodi, skriptide, stiililehtede, piltide) suurus kokku (arvestatuna nende allalaadimiseks kuluva võrguliikluse mahuna) ei tohi ületada 2 MB.	Kasutajaliidese vaate (eriti esmase vaate) laadimine peab olema võimalikult kiire, et kasutaja ei oleks teadmatuses, mis toimub. Kuvatud vaade peab vajadusel indikeerima jätkuvat laadimist. Kasutajaliidese vaadete kuvamiseks vajalike ressursside laadimine tuleb tükeldada nii, et kasutaja ei kogeks vaadete laadimiste ajal liiga pikka ooteaega.	

Platvorm	43	Kui rakendus eeldab kasutaja veebibrauseris skriptide käivitamist, kuid kasutatav skriptimiskeel pole kasutaja brauseris lubatud või toetatud, tuleb kuvada lõppkasutajale arusaadav veateade.	Teate näide JavaScripti eelduse puhul: • Käesolev veebirakendus eeldab JavaScripti, kuid teie veebibrauseris ei ole JavaScript kasutatav või lubatud. • This web application requires JavaScript but JavaScript is disabled or not supported in your web browser.	JavaScript
Rakendus	44	Rakendusel peab olema haldusliides või -liidesed, mille kaudu saavad rakenduse administraatorid ja peakasutajad teha vajalikke ärilisi haldustoiminguid.	Kasutajate haldus, menüüde haldus jne. Kui selliste tegevuste vajadus puudub, pole ka liides nõutud. Vajalikud tegevused ja liidesed tuleb analüüsi etapis kokku leppida.	
Rakendus	45	Kui rakenduse tellija soovib, peab rakenduse kasutajaliidesest olema võimalik näha rakenduse versiooni numbrit (näiteks jaluses, päises või pealkirjas ehk <title> elemendis).	Mõeldud on spetsiaalrakendusi, mis on konkreetsele rakenduse tellijale arendatud (custom built). Nõue ei kehti laialt kasutatava tarkvara (nt Apache HTTP server, Weblogic, Spring raamistik), andmebaaside jms kohta. Nende versioonid tuleb pigem varjata, et teadaolevate turvanõrkuste ärakasutamise võimalusi vähendada. Spetsiaaltarkvara puhul on versiooni näitamine vähem riskantsem kui laialt levinud tarkvara puhul, aga tuleb arvestada, et ka spetsiaaltarkvara puhul toob versiooni näitamine endaga kaasa väikese täiendava turvariski.	
Rakendus	46	Kui rakendus kasutab haldusalast väljas olevaid teenuseid (nt digitembeldamine, ID-autentimine, pangalingi kasutamine jms), peab ta suutma arvestust pidada vastavate teenuste kasutamise mahu üle.	Kasutusinfo tuleb talletada kõikide väljastpoolt kasutatavate teenuste kohta minimaalselt logi tasemel. Oluliste (nt tasuliste) teenuste puhul peab vastav info olema tavakasutajale rakendusest iseseisvalt kättesaadav. Selliste teenuste puhul tuleb see osa täpsemalt lahti kirjutada ka funktsionaalsetes nõuetes.	
Rakendus	47	Aadressiandmete sisestamisel, kuvamisel ja hoidmisel tuleb lähtuda kehtivast Vabariigi Valitsuse määrusest "Aadressiandmete süsteem".	Rakendused liidestatakse RmIT sisemise vastava Maa-ameti andmeid vahendava süsteemiga. (https://www.riigiteataja.ee/akt/113102015002)	
Turva	48	Kui rakendus tuvastab kasutajaid ise, peab olema võimalik piirata ebaõnnestunud logimisi ajaühiku kohta (mobiil-ID, paroolid, ID-kaart) ühelt IP-aadressilt.	Vastavalt tellijaga kokkuleppele kasutatakse captcha-t, konto lukustamist või viiteaega.	
Turva	49	Faili asukohta (path) süsteemis ei tohi edastada kliendi poolele.		

Turva	51	Kliendi ja serveri vahel peab autenditud kasutajasessioonide korral olema sessioon krüpteeritud HTTPS-protokolli kasutades.	Vaikimisi tuleb kasutada HTTPS-protokolli. Koormusjaoturi ja rakendusserveri vahel võib kasutada HTTP protokolli.	
Turva	52	Tellija nõuete järgi piisavalt kõrge (vt nõude nr 2 täpsustust) terviklikkuse nõuetega andmekirjed tuleb versioneerida.	Kõik andmemuudatused peavad baasis säilima. Andmete muutmisel andmeid ei kustutata, vaid tehakse uus kirje uute andmetega. Vana muudetakse kehtetuks. Iga uus kirje peab sisaldama järgmist informatsiooni: • viide kirjele, mille ta kehtetuks muutis (kui on); • kasutaja, kes kirje lõi; • kirje loomise aeg; • sessiooni-ID (kui on olemas). Iga kehtetuks tunnistatud kirje peab omama järgmist informatsiooni: • kasutaja, kes kirje kehtetuks tunnistas; • kirje kehtetuks tunnistamise aeg.	
Turva	53	Rakendus ei tohi kasutada andmebaasi tegevusi, mis eeldavad DDL õiguseid.		
Turva	54	Rakendusse ja andmetele tohib olla ligipääs vaid dokumenteeritud ja tellimuses kirjeldatud teid mööda ning dokumenteeritud autentimisprotseduure kasutades.	Rakendustes ega andmebaasides ei tohi olla ligipääsemiseks teisi võimalusi.	
Turva	57	Sessioonide lõpetamine tuleb teostada serveri poolel ja kõigil rakendustel peab olema konfigureeritav kasutajasessiooni aegumise aeg.	Aeg peab olema muudetav koos teiste konfiguratsiooniparameetritega. Kui kliendilt pole etteantud aja jooksul ühtegi päringut tulnud, tuleb sessioon serveri enda algatusel lõpetada.	
Turva	58	Rakendusse kasutajate poolt sisestatud või muul viisil edastatud andmed tuleb puhastada XSS-filtriga või eemaldada HTML-tag´id.	Soovitatavalt rakendada enne andmebaasi salvestamist, kuid kindlasti enne väljakuvamist.	
Turva	60	Krüpteerimise ja/või räside arvutamise korral tuleb kasutada tugevaid algoritme.	Vastavalt AES-256, RSA-2048, SHA-2 või tugevamaid.	

Turva	63	Rakenduste autentimisvormidel ei tohi salvestada (meelde jätta) kasutajainfot.	Rakendus ei tohi autentimise infot püüda meelde jätta ja annab standardsete meetmete abil (nt HTMLis autocomplete="off") sellest brauserile teada. Mittestandardsete lisameetmete kasutuselevõtt brauserite poolt autentimisinfo meeldejätmise tõkestamiseks (et vähendada brauserite isetegevust kõrgemate turvanõuete puhul) lepitakse eraldi kokku. Erandina võib kliendi seadmes meelda jätta viimati kasutatud autentimismeetodi. GDPR-i kohaselt tohib selle salvestada ainult juhul, kui see ei ole kasutajaga seotav info (või kasutaja on selleks nõusoleku andnud).	
Turva	65	Veebirakendus peab järgima kõiki ASVS (Application Security Verification Standard) tase 2 reegleid vastavalt arendusetapi alguses kehtivale OWASP (Open Web Application Security Project) standardi versioonile.	Rakendus peab OWASPi turvatestid ka info tasemel vigadeta läbima. Leitud vead tuleb parandada. https://owasp.org/www-project-application-security-verification-standard/ https://www.owasp.org/index.php/Cheat_Sheets	Veeb
Turva	66	Rakendus ei tohi lubada ühe kasutajaga mitut samaaegset sessiooni.		
Turva	67	Rakendus tohib aktsepteerida vaid sessioonivõtmeid, mida ta ise on väljastanud. Sisselogimisel peab kasutaja saama uue sessiooni võtme ning endine võti tuleb kehtetuks tunnistada.		
Turva	68	Kui rakendusse laetakse faile väliskasutaja poolt, peab kontrollima faili tüüpi (faili nime laiendi järgi) ja failid peavad läbima viirusetõrje.	Viirusetõrje lahenduse pakub välja RmIT hooldusosakond. Failide üleslaadimiseks kasutatakse failihoidlat. Lubatud failide tüüpide nimekiri peab olema konfigureeritav. Rakenduse sisekasutaja on kasutaja, kes töötab asutuses, kes omab või haldab antud rakendust. Väliskasutaja on kasutaja, kes ei ole sisekasutaja. Näiteks kui tegemist on MTA rakendusega, siis väliskasutaja on kasutaja, kes ei ole MTA ega RmIT ametnik või töötaja.	
Turva	69	Kui rakendusse laetakse faile kasutaja poolt, siis peab faili nimi salvestamisel sisaldama juhuslikku (random) komponenti nii, et faili tee ei ole lihtsalt ära arvatav.	Siin peetakse silmas reaalset faili, mis serverisse salvestatakse.	
Turva	70	Kui rakendusse laetakse faile kasutaja poolt, siis tuleb need failid valideerida.	Näiteks ei tohi üles laetav fail viidata (include) otse failisüsteemis teistele ressurssidele.	

Turva	72	Kasutaja poolt sisestatud andmed tuleb enne välja kuvamist filtreerida. Kasutada võimalusel blacklisting-u asemel whitelisting-ut.	Erisümbolid tuleb filtreerida vastavalt keskkonnale, kus neid andmeid vaadatakse (html; console jne).	
Turva	73	Rakendus ei salvesta andmeid kliendi arvutisse (sh küpsiseid).	Erandid: session cookie, viimane autentimismeetod, mitmekeelse süsteemi puhul keelevalik, küpsistega nõustumise märge. Nõue ei laiene rakendusest sõltumatu veebilehitseja käitumisele (nt sessiooni mitte lõpetamine brauseri sulgemisel).	
Turva	74	Iframe-i kasutamine pole lubatud.	Lubatud ei ole nii rakenduse kasutajaliidese näitamine võõra rakenduse iframe sees kui ka rakenduse iframe-s võõra rakenduse kasutajaliidese näitamine. Mõlematpidi tekivad infoturbe vaates probleemid.	Veeb
Dokumentatsioon	75	Rakenduse tehnilisel dokumenteerimisel tuleb keskenduda rakenduse kesksele teenuskihile, kus peab katma dokumentatsiooniga kõigi teenuskihi liideste avalikud meetodid. Tarkvarasüsteemist väljapoole teistele rakendustele pakutav teenus tuleb dokumenteerida OpenAPI (https://swagger.io/specification/) või WSDL kujul. See dokumentatsioon peab sisaldama ka teenuse (mitte üldisemalt rakenduse) muutelugu (changelog) vastavalt https://keepachangelog.com/ juhistele või sisaldama viita teenuse muuteloole. OpenAPI puhul kasutada selleks päises asuvat "description" välja.	Keskse teenuskihi täiendav kirjeldamine annab kiire ülevaate rakenduse põhifunktsionaalsusest API dokumentatsiooni põhimõttel. Näiteks võib dokumentatsioon asuda OpenAPI (Swagger) spetsifikatsioonis. Dokumentatsiooni lugedes peab aru saama, mida meetodid teevad ja kuidas neid kasutada. Minimaalselt on skoobis kõik liidesed, mis on kasutatavad väljaspoolt liidest pakkuvat moodulit.	
Kood	76	Lähtekoodi mõistetavuse tõstmiseks tuleb läbivalt eelistada "Self-Documenting Code" lähenemist massiivsele kommenteerimisele. Isedokumenteeriva koodi realiseerimisel tuleb eelkõige kasutada läbivalt tähenduslikke nimetusi ja suuremad koodiplokid tuleb eraldada selgemõttelistesse abimeetoditesse. Lisakommenteerimist tuleb kasutada keerulisemate algoritmide täiendavaks selgitamiseks.	"Self-Documenting Code" põhimõtet on väga detailselt kirjeldatud raamatus "Clean Code: A Handbook of Agile Software Craftsmanship"	

Kood	77	Muutujate, tüüpide ja funktsioonide nimed koodis peavad olema inglise keeles, sisulised ja andma aimu nende otstarbest.		
Kood	78	Kasutades sama (ja mitte juhuslikult vaid sisuliselt sama) väärtust erinevates kohtades, tuleb defineerida konstant üks kord ja kasutada seda, mitte korduvalt sama väärtus välja kirjutada.		
Andmebaas	79	Andmetabelites sisalduvad võõrvõtmed peavad nime järgi seostuma tabeli ja väljaga, millele need viitavad.		
Kood	80	Tarnitav lähtekood tuleb enne üleandmist valideerida ja seal ei tohi esineda kriitilisi vigu.	Valideerimiseks kasutatakse koodi analüsaatoreid SonarQube, PMD, Findbugs või olenevalt keelest nendega samaväärset üldkasutatavat tööriista. Ka mittekriitilised leiud tuleb üle vaadata ning (mõistlikkuse piires) nendega arvestades koodi kvaliteeti järjepidevalt parandada ja parendada.	
Kood	81	Kasutuses mitteolev kood tuleb rakenduse lähtekoodist kõrvaldada.	Koodi saab vajadusel taastada versioonihalduse abil.	
Kood	82	Lähtekood peab vastama üldlevinud standarditele.	Java: Google Java Style Guide (https://google.github.io/styleguide/javaguide.html). Mõistlikkuse piires, nt plokkide 2-tühikulise taande asemel sobib taandeks ka 4 tühikut. PHP: PHP Framework Interop Group'i standardid ja soovitused (https://www.php-fig.org/psr/). Drupali arenduste puhul tuleb lähtuda Drupali poolt pakutud kodeerimisstandardist (https://www.drupal.org/docs/develop/standards). SQL, PL/SQL: ISO/IEC 9075-* sari. Eesmärk on lihtsustada migreerimist andmebaasimootorite ja nende versioonide vahel. Andmebaasimootori-spetsiifilise koodi kasutamine on lubatud vajaliku jõudluse või ühilduvuse saavutamiseks. Koodi analüüsimise tööriist (nt Checkstyle) ei tohi anda vigu ega olulisi märkusi. Süsteemseks lähtekoodi stiili määramiseks kasutada ka .editorconfig faili. Üldlevinud standardi kasutamisest tähtsam on koodibaasi piires ühtse lähtekoodi stiili kasutamine, mis võib pärandvara (legacy) koodibaasis kaasaegse standardi kasutamist piirata.	

Kood	83	Tarkvara tellijale loodud lähtekood ja tehised peavad asuma tellijale (mitte arenduspartnerile) viitavas grupis, nimeruumis jne.	Näiteks Java puhul peab lähtekood asuma "ee.{tellija-asutus}.{tarkvarasüsteemi-nimetus}" nimeruumis (package), näiteks "ee.emta.fidek", ning Maven tehiste gruppide nimed peavad samuti viitama tellijale (mitte arenduspartnerile). Üldine kood, mida pole loodud tellijale tehtud arenduste raames, võib paikneda ka mujal.	
Arhitektuur	84	Rakenduse arhitektuur peab olema kooskõlastatud RmITiga.	Sealhulgas tuleb RmIT arhitektiga üle arutada rakenduse jõudlust oluliselt mõjutavad aspektid. Näiteid: • ORM-i (object-relational mapper), nt Hibernate, kasutamine. Keeruliste andmemudelite puhul pigem mitte kasutada. • Mahukate UI kuvade realiseerimisel, kas nt vaja materialiseeritud andmebaasi vaateid (view). Rakenduse kihid (nt veebikiht, teenuskiht, andmevahetuse kiht), komponendid ning komponentide poolt pakutavad ja kasutatavad liidesed peavad olema selgelt defineeritud ja dokumenteeritud. Kooskõlastada tuleb ka töö käigus tekkivad arhitektuurilised muudatused.	
Kood	85	Rakenduse äriloogika realiseerimine andmebaasi vahenditega (nt PL/SQL-ga) ei ole lubatud, välja arvatud kokkulepitud konkreetsel vajadusel.	Näiteks saab erandi kokku leppida kõrgjõudluse saavutamiseks.	
Kasutajaliides	89	Kasutajaliidese kõik disainiotsused peavad olema kooskõlastatud tellijaga.	Kasutajaliidese loomisel tuleb lähtuda tellija UI/UX nõuetest (nt stiiliraamat/brandbook). Näiteks võib tellija lähtuda üleriigilisest VEERA-st ("veebide raamistik"). MTA puhul tuleb lähtuda värskeimast eMTA disainisüsteemist. Rohkem infot: https://confluence.rmit.ee/x/kGAnFg	
Kasutajaliides	90	Veebipõhine kasutajaliides peab olema kasutatav enamlevinud veebibrauseritega.	Loetelu ja versioonid on toodud IT-profiilis.	
Kasutajaliides	91	Kasutajaliides peab alati küsima kinnituse andmete kustutamise ja massmuutmiste kohta, kui pole kokku lepitud teisiti.		
Kasutajaliides	92	Süsteemis esinevate tehniliste vigade sisu ei tohi lõppkasutajale kuvada. Koos veaga peab kuvama unikaalse vea tunnuse mis on lihtsalt logidest leitav.	Kasutajale tuleb kuvada võimalikult täpne teade, missugune toiming ebaõnnestus kasutaja vaatest. (Nt mitte „andmebaasi ühenduse viga“, vaid „salvestamine ebaõnnestus“). Vea tunnus võimaldab konkreetse kasutaja probleemi siduda tehnilise veaga logides.	

Kasutajaliides	93	Rakenduse kasutajaliideses nähtavad tekstid peavad olema koodist ja kujundusest eraldi.	Mitmekeelse rakenduse puhul peab uue keele lisamine olema teostatav konfiguratsioonifailist või administreerimisliidesest.	
Kasutajaliides	94	Rakenduse kasutajaliides peab teavitama kasutajat ette sessioon aegumisest.	Etteteavitamise aeg peab olema konfigureeritav ilma rakendust uuesti ehitamata.	
Kasutajaliides	95	Kui vorm koosneb paljudest väikestest andmeväljadest (nt taotlus), siis jagatakse vorm etappideks ning salvestatakse vastava etapi lõpus.		
Kasutajaliides	97	Interaktiivsete vormide puhul (näiteks faili üleslaadimine) ei tohi lehe värskendamisel andmeid/olekut muutvat tegevust korrata (faili taas üles laadida, avaldust topelt esitada jne).	Andmeid lisavate või muutvate operatsioonide korral peab toimima andmete terviklikkust tagav operatsioonide idempotentsus.	
Kasutajaliides	98	Päringutel, mille kestvuseks on lubatud rohkem kui 3 sekundit, peab kasutajale näha olema, et päringut töödeldakse.	Indikatsioon on vajalik, et vältida kasutaja asjatut tegevuse kordamist. Päringutele, mis peavad rakenduse normaalse töö korral vastuse saama kiiresti (alla 3 sekundi), ei ole vaja töötlemise indikaatorit realiseerida.	
Kasutajaliides	99	Veebilehitseja navigatsiooni nupud peavad töötama rakenduses, kuid ei tohi korrata andmeid muutvaid tegevusi.	Vähemalt üksteise järel toimuvate andmeid mitte muutvate päringute vahel peab saama liikuda edasi/tagasi (näiteks otsingutulemuste leheküljed).	
Kasutajaliides	100	Rakendus ei tohi avada uusi lehitseja aknaid.	Erandina võivad rakendusest välja suunduvad lingid avaneda uues aknas (kasutades target="_blank"). Vajadusel võib kooskõlastatult tellijaga kasutada modaalseid (modal) brauseri dialoogiaknaid.	Veeb
Kasutajaliides	101	Kasutaja tegevuste kohta peab süsteem esitama selge tagasiside.	Nt "Edukalt salvestatud", "Tegevus pole lubatud", selge mittetekstiline graafiline märguanne.	
Kasutajaliides	102	Serverit koormavate päringute puhul peab rakendus piirama sama tegevuse asjatut kordamist.	Kui päringu ajaks on lubatud >10sek (koormustestide plaanis) siis tuleb piirata näiteks nupu uuesti vajutamist kuni lehe uuesti laadimiseni. Nupu saab näiteks blokeerida, kasutajaliidesest eemaldada, kasutaja eest varjata.	

Kasutajaliides	103	Peab järgima adaptiivse (adaptive) ja/või muganduva (responsive) kasutajaliidese printsiipe.	Veebilehed peavad olema klientseadmele kohanduvad (kasutatavad nii lauaarvutites kui ka mobiiltelefonides ja tahvelarvutites). Toetatud vaadete parameetrid (laius, puutetundlikkus) võib kitsendada kokkuleppel tellijaga. Tuleb vältida elementide paigutust (layout), mis vajavad info lugemiseks kuva nihutamist horisontaalselt. Horisontaalset nihutamist võib lubada alates tellijaga kokkulepitud parameetritest.	
Kasutajaliides	104	Loetelu kuvamisel tuleb kasutada paginatsiooni, kus ühel lehel kuvatakse korraga kokkulepitud arv kirjeid. Kokkuleppel tellijaga võib kasutada ka "infinite scrolling" lahendust.	Kirjete arv lehel peab olema konfigureeritav.	
Andmebaas	105	Andmebaasis peab kasutama indekseid ja muid meetmeid, et nõuded rakenduse jõudlusele oleksid täidetud ka tulevikus.	Vastavalt planeeritud kasutusajale ja andmete hulgale.	
Andmebaas	106	Tuleb kasutada päringumuutujaid (Parameter Binding).	SQL päringute väljakutsumisel väljastpoolt andmebaasi peab kasutama päringumuutujaid, et vältida SQL vahemälu fragmenteerumist ja SQL injection ründeid. Andmeanalüütika korral ei pruugi jõudluse vaates päringumuutujate kasutamine olla mõistlik. Kui päringumuutujatest loobuda, peab see olema tellijaga kokku lepitud ning turvalisus peab olema tagatud teiste meetmetega.	
Andmebaas	107	Andmebaas peab toetama nii külm- kui ka kuumvarundamist (peegeldamist) teise serviruumi.	Ei tohi kasutada teenuseid, mis välistavad andmebaasi peegeldamist (nt "filestream").	
Andmebaas	108	Rakenduse operatiivbaas tuleb hoida võimalikult väike. Suurte andmemahtude korral, kui äriprotsess seda võimaldab, tuleb kasutada andmete arhiveerimist väljapoole põhibaasi.	Andmete pikaajalise (10+ aasta perspektiivis) säilitamise vajadused peavad olema analüüsitud. Pikaajalist säilitamist operatiivbaasis üldjuhul ei tehta. Äripool võib otsustada operatiivandmebaasist andmete välja arhiveerimise asemel andmed kustutada, kui ärireeglid seda võimaldavad.	
Andmebaas	109	Andmebaasi objektide nimed tohivad sisaldada ainult ladina tähestikku, numbreid ja alakriipsu.	Ei ole lubatud kasutada täpitähti.	

Dokumentatsioon	110	Rakenduse dokumentatsioon peab üldjuhul olema kirjutatud eesti keeles. Eelkõige arendajatele suunatud (nt OpenAPI või WSDL kujul) dokumentatsioon võib olla kirjutatud inglise keeles. Ühe dokumendi piires peab kasutama ühte keelt.	Erandiks võib olla tellijaga kokkuleppel kolmanda osapoole komponentide (mis pole kirjutatud tellija jaoks) dokumentatsioon.	
Dokumentatsioon	111	Rakenduse dokumentatsioon peab sisaldama tabelite, andmete, logide mahu kasvu hinnangut rakenduse sihipärase kasutamise korral ettenähtud arvu kasutajate poolt (MB/GB kuus/aastas).	Esialgne kirjete mahu hinnang peab tulema lähteülesandest, ning täpsustuma eel- ja detailanalüüsi käigus. Mahuhinnang peab sisaldama ka logide säilitamise ja arhiveerimise tähtaegu. Arvestada tuleb ka potentsiaalsete arhiveerimise võimalustega, mis vastukaaluks jooksvalt vähendavad operatiivandmebaasis asuvate andmete mahtu.	
Dokumentatsioon	112	Iga uue versiooniga peab alati välja tooma versiooni muudatuste kirjeldused muuteloos (changelog).	Muutelugu peab kajastama kõiki muudatusi eelmise ja uue versiooni vahel. Rohkem infot: https://confluence.rmit.ee/x/jWqBDw	
Versioonihaldus ja paigaldamine	113	Arendaja peab veenduma, et teeb muudatusi aktuaalsesse koodi.	Enne muudatuste tegemist võetakse viimane seis RmIT koodirepositooriumist.	
Versioonihaldus ja paigaldamine	114	Paigaldusjuhendi alusel valmiv paigalduspakett tohib sisaldada ainult minimaalse rakenduse käitamiseks vajamineva failikomplekti.	Näiteks ei tohi paigalduspakett sisaldada lähtekoodi (kompileeritavate keelte puhul), arendus- või testkeskkonna spetsiifilisi komponente, andmebaasi muudatuste paigaldamise koodi ja vahendeid (nt Liquibase) jne.	
Versioonihaldus ja paigaldamine	115	Paigaldusjuhendi alusel ehitatud paigalduspaketti peab olema võimalik paigaldada erinevatesse keskkonadesse ja masinatesse.	Näiteks ei tohi rakenduse paigaldamine uude serverisse vajada rakenduse paigalduspaketi uuesti ehitamist.	
Turva	116	Rakenduse moodustavad komponendid (sh. CSS, skriptid) peavad asuma tellija taristus ja neid tuleb sealt ka kasutada. Lähtekoodi kompileerimine ja paigalduspakke ehitamine peab olema teostatav ka välisvõrguühenduse puudumise korral.	Rakendust peab saama välisvõrgu (Interneti) ühenduseta taristus ehitada ja paigaldada ilma koodihoidlas olevaid faile muutmata. Äriliselt välisvõrgust sõltumatud funktsionaalsused peavad rakenduses korrektselt toimima ka välisvõrgu ühenduse puudumisel. Vajalikud sõltuvused peavad olemas olema RmIT võrgus (RmITi hallatavates paketihoidlates, nt Java puhul RmITi hallatavas Maven Repository's). Skriptid peavad asuma RmIT infras. Lähtekoodis asuvates vaikimisi kasutatavates ehitamise seadistustes ei tohi olla viiteid arenduspartneri infrastruktuurile. Näiteks tuleb vältida olukordi, kus tuleb RmIT infras ehitamiseks üle kirjutada arenduspartneri enda Maven repository URL.	

Versioonihaldus ja paigaldamine	117	Kogu arendatud lähtekood peab asuma RmIT lähtekoodihoidlas (RmIT GitLab-is). Arendatud tarkvaras kasutatavad kolmandate osapoolte komponendid (nt raamistikud ja teegid) peavad asuma RmIT tehiste hoidlas (artifact repository). Maven-i, npm-i, Docker-i jt tehnoloogiate tehiste jaoks on RmIT Nexus Repository Manager.	(Lähte)koodid ja tehised tuleb RmIT (lähte)koodi ja tehiste hoidlatesse laadida hiljemalt RmIT hallatavatesse keskkondadesse paigaldamise ajaks. JavaScript bundle (ühte faili kokku pakendatud ja potentsiaalselt minimeeritud JavaScripti komponendid) ei liigitu RmIT vaates lähtekoodiks, olgugi et mõnel juhul sobib bundle tarnida lähtekoodihoidlas. Kui arenduse tulemusena tarnitakse JavaScript bundle, tuleb lisaks tarnida ka bundle-sse kuuluvate arendatud komponentide lähtekoodid ja bundle ehitamise juhised. Seejuures peab olema selgelt arusaadav, millised komponendid bundle-sse kuuluvad ning kust leiab nende komponentide lähtekoodid (ka õiged lähtekoodi versioonid), mille põhjal on bundle ehitatud.	
Versioonihaldus ja paigaldamine	118	Repositooriumisse laetakse üles ainult lähtekood, kujunduselemendid (pildid, ikoonid, mallid) ja andmebaasi skriptid.	Suuremahuliste failide (teegid, andmebaasi dump-id jms) laadimine repositooriumisse on keelatud.	
Versioonihaldus ja paigaldamine	119	Andmebaasi muudatuste tarnimise vahend peab võimaldama automaatselt korraga paigaldada kõik andmebaasist puuduvad muudatused (vahend tuvastab andmebaasi veel paigaldamata tarned/muudatused ning paigaldab need). Kasutatavaks vahendiks on vaikimisi IT profiilis toodud andmebaasi muudatuste propageerimise vahend.	Viidad juhtfaili ja skriptide vahel peavad olema suhtelise teega, nii nagu need versioonihalduses paiknevad.	
Versioonihaldus ja paigaldamine	120	Andmebaasi uuendusi peab saama paigaldada andmebaasi administraatori (DBA) õigustes.	Andmebaasi skeemi nimi peab olema seadistatav. Liquibase puhul käivitatakse uuendused skeemi või andmebaasi omaniku õigustes.	
Versioonihaldus ja paigaldamine	121	Baasiuuenduste paigaldus peab tekitama logi.	Oracle puhul spool file.	
Versioonihaldus ja paigaldamine	122	Tarnet peab olema võimalik ilma andmeid rikkumata korduvalt paigaldada.	Commiti kasutada nii vähe kui võimalik ja nii palju kui vajalik, et oleks vigast tarnet võimalik rollbackida. DML korduvkäivitamine peab constraintide abil välistama duplikaatkirjete tekke	
Versioonihaldus ja paigaldamine	123	Kõik andmebaasi skriptid peavad olema UTF-8 kodeeringu (without BOM) formaadis.		

Versioonihaldus ja paigaldamine	125	Andmebaasi muudatused peavad olema käsurealt käivitatavad ja ei tohi olla pakendatud rakenduse sisse. Muudatused peab olema võimalik eksportida skriptina (ka tagantjärele).	Nii saab baasimuudatusi eraldi paigaldada ja kontrollida. Lisaks välistab see klastri puhul võimalikud paralleelsuse probleemid. Kui kasutatav lahendus nõudeid ei täida, tuleb baasimuudatused edastada skriptidena.	
Logimine	127	Rakendus peab logima sessiooni algamise ja lõppemise (kui rakenduses sessioon tekib ehk kui rakendus ei ole stateless), kasutaja IP aadressi (mille koormusjaotur edastab rakenduse õlale vastavas HTTP päises), autentimismeetodi (ID-kaart, mobiil-ID vms, kui autentimisteenus selle identsustõendis rakendusele edastas), eduka autentimise puhul tuleb logida ka kasutaja isikukood ja mobiil-ID puhul telefoninumber (kui autentimisteenus need väärtused identsustõendis rakendusele edastas).	Turvalogi, mis võib lisaks eelnevalt nimetatud väärtustele sisaldada ka muud kasutajate autentimisega, õiguste kontrolliga ning sessioonidega seotud infot, peab olema võimalik suunata eraldi faili. Ainuüksi logimise jaoks teistest teenustest andmeid pärida ei tule (logida saab andmeid, mis on rakendusel olemas). Info ja sellest kõrgema (nt Warning ja Error) taseme logi kirjed ei tohi sisaldada andmeid, mis võimaldavad sessioone üle võtta. Trace tasemel sellist piirangut ei ole.	
Logimine	128	Rakendus peab võimaldama logida kõiki väljaminevaid ja sissetulevaid (ka X-tee teenuste kaudu liikuvaid) päringuid. Sõnumi sisu logimist peab saama eraldi sisse-välja lülitada, st sõltumatult muust sama taseme logist. Vastav juhis peab sisalduma rakenduse paigaldusjuhendis.	Eesmärk on lihtsustada tarkvara silumist ja toodangukeskkonna probleemide lahendamist. TRACE taseme logi peab sisaldama ka andmebaasi päringute parameetrite väärtuseid.	
Logimine	129	Rakendus peab logima kõiki rakenduses tekkivaid tehnilisi vigu kas faili või andmebaasi.	Logi peab sisaldama minimaalselt (toodud järjekorras) vea tekkimise aega, veakoodi, veakirjeldust (stack trace, traceback vms), võimalusel kasutaja andmeid (nimi, ID, IP ja URL), HTTP-, GET- ja POST-parameetrid ja nende väärtusi.	
Logimine	130	Logitabelid (sh krüptoaheldatud logid) peavad olema arhiveeritavad operatiivbaasist välja nii, et rakendus jääb vanade kirjete eemaldamisel tööle.	Näiteks on oluline, et logi struktuur võimaldab vanu kirjeid eemaldada ja selle peale ei ehitata äriloogikat, mis takistab vanade kirjete eemaldamist.	

Logimine	131	Kasutajakontode puhul peab olema tagatud funktsioonide lahusus. Konkreetse kasutajatunnuse alt infosüsteemis tehtud andmeuuendused ja -muudatused peavad olema üheselt seotavad selle kasutajatunnusega.		
Logimine	132	Logi peab kajastama rakenduse poolt tehtud tööd. Seadistuste, võrguligipääsude ja muude probleemide põhjused peavad olema logist selgelt väljaloetavad.	Rakenduse loomisel tuleb lähtuda IT-profiilis toodud logitasemete kirjeldustest.	
Testimine	133	Rakenduse kõik üleantavad versioonid peavad enne tellijale üleandmist olema testitud.	Testplaan ja testitulemused tuleb edastada tellijale koos rakenduse üleandmisega.	
Testimine	134	Rakendusega peab olema kaasas skript jõudlustestide tegemiseks, mis võimaldavad tuvastada rakenduse poolt talutava koormuse ülempiiri (stress test) ning selgitada, kuidas rakendus käitub piiripealse koormusega (load test). Testid peavad võimaldama simuleerida kasutajate kasutusmustreid, sealhulgas ka erineva pikkusega pause kasutajate tegevustes, et saaks tuvastada probleeme, mis tekivad pikalt ootele jäävate ühenduste puhul. Koormusega tuleb katta ka olulised liidestused teiste süsteemidega (mitte ainult rakenduse sisemine toimimine). Olemasoleva rakenduse uuendamisel tuleb uuendada ka vastavat skripti.	Jõudlustestide täpne kirjeldus ja kasutatavad töövahendid tuleb kokku leppida detailanalüüsi käigus. Arendaja peab koos rakendusega tarnima skripti ja vajalikud tarkvaralised vahendid kokkulepitud jõudlustestide läbiviimiseks. Jõudlustestide läbiviimine ei tohi nõuda tellijalt omapoolset tarkvara arendamist, skriptide kirjutamist või litsentside ostmist.	
Turva	135	Rakendus ei tohi vastata http-päringutele, mida rakendus oma töös reaalselt ei kasuta.	Nt kui rakendus kasutab GET päringut urlile /hello, siis ei tohi rakendus vastata samale POST päringule, kui seda rakenduse töös ei kasutata.	
Turva	136	Rakenduse logidesse ei tohi kirjutada erisümboleid, mis võivad moonutada logi formaati või väljanägemist. Selleks tuleb kasutada struktuurset logimist, mille puhul erisümbolid (reavahetus, backspace jne) vajadusel kodeeritakse (escape).	Kasutada tuleb võimalikult palju üldlevinud logimise raamistike võimalusi, et vältida rakendustes erilahenduste realiseerimist. Näiteks Logback-i puhul saab kasutada vastavaid lisasõltuvusi/teeke (https://www.baeldung.com/java-structured-logging).	

Turva	137	Mobiil-ID rakendamisel tuleb lisaks mobiilinumbrile küsida ka isikukoodi.		
Andmebaas	138	Kui tarne käigus lisatakse uusi objekte baasi, siis tuleb tarne kirjelduses eraldi uued objektid välja tuua.	Vajalik rakenduse administraatoritele uute tabelite vaatamise õiguste tellimiseks.	
Kasutajaliides	139	Rakendus peab kasutajat hoiatama, kui kasutatavas brauseris ei ole rakenduse funktsionaalsus täismahus kasutatav.	Mitte toetatud brauseri tuvastamiseks saaks kasutada näiteks "feature detection" põhist lahendust (nt Modernizr baasil).	Veeb
Andmebaas	140	Rakendus peab olema loodud ja tarnitud selliselt, et rakenduse andmete omanik baasis ja rakenduse andmebaasi ühenduse kasutaja saab määrata erinevad. Õigused, mida läheb vaja rakendusele määratud andmebaasi kasutajal, peavad olema minimaalsed (ainult rakenduse tööks vajalikud) ja dokumenteeritud.	Nt andmebaasi skeem ehk andmete omanik baasis on RAKENDUS1 ja rakenduse andmebaasi ühenduse kasutaja on RAKENDUS1_APP_USER.	
Andmebaas	141	Binaar- ja suuremahulisi objekte (BLOB) ei salvestata rakenduse relatsioonilisse operatiivandmebaasi.	Sobivad alternatiivid lepitakse kokku projekti käigus. Näiteks salvestatakse failid failihoidlasse väljaspool andmebaasi.	
Andmebaas	142	Objektide sõltuvused andmebaasis hoitakse võimalikult lihtsana selliselt, et puudub vajadus kasutada WITH GRANT OPTION privileege.		
Arhitektuur	143	Rakenduste vaheline suhtlus tuleb realiseerida läbi teenuste.	Sobivad liideste formaadid lepitakse kokku projekti käigus (REST, SOAP, JMS).	
Turva	144	Krüptograafiliste algoritmide ja meetodite valimisel tuleb lähtuda kehtivast krüptograafiliste algoritmide elutsükli uuringust	Kehtiv uuringu versioon on kättesaadav RIA veebilehtelt (https://www.ria.ee/).	
Andmebaas	148	Andmebaaside skeemid tuleb hoida läbivalt ühes keeles. Vanades skeemides kasutatakse olemasolevat keelt ja uued skeemid luuakse läbivalt inglisekeelsetena.	Uutest inglisekeelsetest skeemidest vanale eestikeelsele viitamise vajadusel kasutatakse välja kommentaari, mis on hiljem lihtsasti muudetav.	

Testimine	150	Kasutajaliidese automaattestimise võimaldamiseks tuleb HTML elementidel eesmärgipõhiselt kasutada identifikaatoreid (atribuut "id"), klasse (atribuut "class") ja nimetusi (atribuut "name").	Kõigil unikaalsetel interaktiivsetel või muutuvatel elementidel peab olema lehel unikaalne id väärtus. Samalaadsetel elementidel (nt korduv vormi element / loetelu element) peab olema sisu kirjeldav klassi nimetus. Vormi elementidel peavad olema nimetused nii nagu (X)HTML standard neid ette näeb.	
Andmebaas	151	Kõik andmebaaside tabelid ja tabelite väljad tuleb loomisel ja muutmisel kirjeldada sisulise ja ajakohase infoga andmebaasi vastavas kommentaaris nii, et see seletab lahti nende otstarbe.	Andmeobjektide kommentaaride kood tuleb tarnida koos DDL-iga.	
Üldine	152	Kui rakenduse kasutajad on väljaspool RmIT hallatavat võrguressursside domeeni, peab rakendus võimaldama autentimist muuhulgas ID-kaardi ja Mobiil-ID abil. Kui selline pakutav rakendus ei toeta ID-kaardi või Mobiil-ID abil autentimist, tuleb pakkumises välja tuua alternatiivne samaväärne hinnastatud lahendus koos pakkujapoolse realiseerimise, juurutamise, ülalhoiu jt kaasnevate kuludega.	Toetab üldist eesmärki, et pakutav lahendus tuleb sobitada olemasolevasse keskkonda nii, et see tervikuna antud keskkonnas toimib.	
Turva	154	Kui rakendus töötleb tellija nõuete järgi piisavalt kõrge (vt nõude nr 2 täpsustust) konfidentsiaalsuse tasemega andmeid, peab rakendus kasutajate toimingute auditeerimise eesmärgil kirjutama auditlogi. Auditlogi tuleb kirjutada faili (mitte andmebaasi).	Auditeeritavad kasutajate toimingud määratakse analüüsi käigus. Auditlogi üldiste nõuete dokumentatsioon: https://confluence.rmit.ee/pages/viewpage.action?pageId=77612807	
Turva	155	Rakendus, kuhu laetakse dokumente, peab võimaldama konfigureerida lubatud faili laiendite nimekirja (whitelist).		
Üldine	156	Automaatselt käivituvaid taustatöid peab saama käsitsi (taas)käivitada ja peatada.	Käivitamiskatse ebaõnnestumisel tuleb kasutajale sellest teada anda ja logida käivitamist takistav põhjus. Käsitsi käivitamise viis tuleb tellijaga kokku leppida. Taustatööde käivitamine ja peatamine ei tohi eeldada rakenduse taaskäivitamist.	

Kasutajaliides	157	Avaliku veebi avalikud materjalid peavad olema otsingumootorite jaoks indekseeritavad ja otsinguga leitavad.	SEO-le (search engine optimization) peab pöörama piisavalt tähelepanu, et kasutajad pakutavad e-teenused populaarsetes otsingumootorites (nt Googles) üles leiaks (https://developers.google.com/search/docs/fundamentals/seo-starter-guide). Avalikud andmed peavad olema kättesaadavad ka JavaScriptita.	
Dokumentatsioon	159	Lähtekoodi selgelt eristatav peamine haru (vaikimisi "master" nimeline haru) peab sisaldama dokumentatsiooni vähemalt järgmises ulatuses: README: • Projekti üldine kirjeldus. • Lähtekoodi struktuur, kaasa arvatud harudeks jaotamise loogika lähtekoodihoidlas. • Tarkvara nõuded keskkonnale. • Tarkvara konfigureerimise, ehitamise ja paigaldamise juhendid. CHANGELOG: • Tarkvarakomponendi muudatuste ajalugu keepachangelog.com juhiste järgi. LICENCE: • Sisu annab ette RmIT. Vähemalt üks README (juurkataloogis) on kohustuslik. Moodulite puhul kaaluda ka moodulite varustamist READMEdega. CHANGELOGi tuleb pidada iga eraldi versioneeritava komponendi või mooduli kohta eraldi. Dokumentatsiooni failide laiend (ja vastavalt vorming) võib olla .md, .adoc või .txt.	Vajadusel võib sisu jagada mitme faili vahel, kasutades failide vahel navigeerimise võimaldamiseks viitasid.	

Versioonihaldus ja paigaldamine	160	Tarkvarakomponendid on versioneeritud semantilise versioneerimise põhimõttel MAJOR.MINOR.PATCH kujul, kus PATCH on veaparandus, MINOR on funktsionaalne tagasiühilduv uuendus ja MAJOR on tagasiühilduvust potentsiaalselt lõhkuv uuendus.	https://semver.org/ Ei tohi luua sisutühjasid komponentide versioone. Näiteks põhjustab sisutühjasid versioone rakenduse ja selle klienditeegi versioneerimise sidumine ehitamise juhtfailides nii, et klienditeegi uue versiooni loomisel luuakse automaatselt ka rakenduse uus versioon ja vastupidi. Põhjused rakenduse uue versiooni tegemiseks ja klienditeegi uue versiooni tegemiseks võivad olla erinevad ja üksteisest sõltumatud.	
Arhitektuur	161	Rakendus peab olema üles ehitatud nii, et see toetaks teenuste versioneerimist teenuse tüübile omaste vahenditega. Vaikimisi versioneeritakse teenused URLi põhiselt.	Näiteks veebiteenuste puhul tuleb toetada versioneerimist URLi, API või andmestruktuuri (schema) põhiselt, sõnumijärjendite korral tuleb toetada versioneerimist URLi, nimemustri või andmestruktuuri (schema) põhiselt jne.	
Arhitektuur	162	Rakendused peavad olema projekteeritud nii, et tarneid saab üldjuhul äriteenuste katkestuseta paigaldada.	Levinud meetod katkestuseta paigaldamiseks on paralleelpaigaldus (side-by-side). Paralleelsus ehk rakenduse erinevate versioonide samaaegne töötamine võib toimuda klastri tasemel (erineval õlal erinev versioon) või rakendusserveris (erinev versioon korraga samas rakendusserveri instantsis). Tarne dokumentatsioon peab kajastama paralleelpaigalduse võimalikkust.	
Arhitektuur	163	Rakendust projekteerides ei tohi eeldada, et sõnumite edastuskanal (nt JMS queue) tagab sõnumite kohalejõudmise (reliable delivery).	Sõnumi saatja peab vajadusel saama kinnituse vastuvõtjalt.	
Konfiguratsioon	164	Võimalusel peab konfiguratsiooni parameetril olema vaikeväärtus, aga see ei tohi olla selline, mis ei sobi kõikidesse keskkondadesse.	Kui kõikidesse keskkondadesse sobivat vaikeväärtust ei ole võimalik määrata, peab süsteem kohustama administraatori seda määrama keskkonnaspetsiifilise parameetrina (annab käivitamisel vea, kui vaikeväärtust mitte omav parameeter on väärtustamata). Tarkvara tüüpseadistus ja parameetrite vaikeväärtused peavad soodustama süsteemi turvalist käitamist.	
Arhitektuur	165	Tagasüsteem (backend) peab enda poolt pakutavad teenused registreerima teenuste registrisse ning kasutatavate teenuste URLid küsima teenuste registrist. Kasutada tuleb ETCD baasil põhinevat RmIT teenuste registrit, mille kohta leiab infot arenduspartnerite teabelehelt: https://confluence.rmit.ee/x/9zu1Bg	Kui kasutatava platvormi jaoks on olemas teenuste registri klienditeek, peab seda teeki (täpsemalt selle teegi viimast versiooni) kasutama. Java klienditeegi infoleht arenduspartnerite teabelehel: https://confluence.rmit.ee/x/3jm1Bg	

Kasutajaliides	166	HTML (ka genereeritud) ei tohi sisaldada JavaScripti ega CSSi.	Eelkõige on mõeldud serverist serveeritavat HTMLi. Tuleneb otstarbe lahususe (separation of concerns) põhimõttest.	
Andmebaas	167	Andmebaasiühenduste connection pool peab olema konfigureeritav parameetritega: • max ühenduste arv • jõude ootavate ühenduste min arv (minimum idle) • aeg, mille möödudes jõude olev ühendus suletakse (idle timeout)	Need konfigureerimise võimalused peavad olema dokumenteeritud.	
Andmebaas	168	Andmebaasi ühendused peab olema võimalik seadistada kasutama TLS-i.		
Turva	169	Kõik X-tee turvaserveri poole pöördumised peab olema võimalik seadistada käima üle turvalise kanali (TLS) ja autenditult.	Autentimine toimub kliendi sertifikaadiga.	X-tee
Turva	171	HTML lehe juurelement peab sisaldama tõlkimist keelavat atribuuti (<html translate="no">). Mitteavalikke andmeid sisaldada võival sisul peab olema õigekirjakontroll keelatud (spellcheck="false").	Eesmärgiks on tõkestada sensitiivsete andmete leket. Näiteks võib kasutaja brauser olla seadistatud külastatavaid lehti Google Translate kaudu automaatselt tõlkima, mille puhul, kui ei ole määratud translate="no", saadab brauser veebilehel asuvad tekstid Google serveritesse.	Veeb
Konfiguratsioon	172	Rakendus peab käimaminemisel sisseloetud konfiguratsiooni koheselt valideerima ja logima. Konfiguratsioon tuleb Logida INFO tasemel. Logida ei tohi sensitiivseid väärtuseid, nt paroole ja privaatseid võtmeid.	Valideerimisel tuvastatud vead tuleb logida ERROR tasemel ning logi sõnum peab selgelt väljendama vea põhjust.	

Seire	173	Rakendustes peab olema realiseeritud hajus jälgitavus (distributed tracing), et hajusa süsteemi mitmes komponendis tehtud toimingud, mis moodustavad loogiliselt ühe terviku, oleks omavahel seostatavad. Selleks peavad rakendused olema valmis sissetulevast päringust korrellatsiooni ID vastu võtma, selle maha logima ning päringu töötlemiseks vajalikesse väljaminevatesse päringutesse sama korrellatsiooni ID kaasa panema (edasi levitama). Kui sissetulev päring korrellatsiooni ID-d ei sisalda või väljaminev päring on päringute jadas esimene (nt kui päring tehakse rakenduse enda algatusel), tuleb väljaminevasse päringusse panna kaasa uus unikaalne korrellatsiooni ID.	Nõutav hajusa jälgitavuse tehnoloogia on toodud IT-profiilis.	
Platvorm	174	Parajasti siis (ja ainult siis), kui X-tee päringu põhjustab autenditud kasutaja toiming (mitte infosüsteem ise nt taustaprotsessis), peab X-tee päringu päises olema selle kasutaja identifikaator (elemendis Envelope/Header/userId).	Vajalikest salastatuse nõuetest tulenevalt võib kasutaja identifikaatori jätta täitmata.	X-tee
Turva	175	Kõik rakenduse poolt kasutatavad ressursid, mis asuvad teistel domeenidel (rakendusest erineval domeenil), peavad olema dokumenteeritud.	Eelistada tuleb samal domeenil asuvate ressursside kasutamist, kuna see on turvalisem. Seega peab domeeniväliste ressursside kasutamine olema ka põhjendatud.	Veeb

Turva	176	Rakendus peab töötama korrektselt, kui rakenduse ees olev komponent (nt reverse proxy) määrab HTTP vastustele järgmised päised: • Content-Security-Policy: default-src 'self'; object-src 'none'; base-uri 'self'; frame-ancestors 'none' • Strict-Transport-Security: max-age=31536000; includeSubDomains • X-Frame-Options: DENY • X-Content-Type-Options: nosniff Rakendus peab töötama korrektselt, kui rakenduse ees olev komponent (nt reverse proxy) määrab HTTP vastuses Set-Cookie päistele atribuudid Secure, HttpOnly ja SameSite=Lax. • Set-Cookie: <cookie-name>=<cookie-value>; Secure; HttpOnly; SameSite=Lax	Kui rakendust ei ole võimalik tööle saada siin kirjeldatud CSP (Content Security Policy) päisega, tuleb vajalikud CSP lõdvendused põhjendada ja RmITiga kooskõlastada. Maksimaalselt range CSP, millega rakendus korrektselt töötab, tuleb rakenduse paigaldusjuhendis dokumenteerida (koos lõdvenduste põhjustega).	Veeb
Rakendus	177	Nõuded mitmekeelsetele tekstiressurssidele (i18n): • UTF-8 kodeering. • Iga keele jaoks eraldi fail(id). • JSON, YAML või Java properties formaat. • Loogiliste moodulite failid eraldi. • Faili nime muster: [common|<module>]_<ISO 639-1>[-<ISO 3166-1 alpha-2>].[properties|json|yaml] • Failide asukoht vastavalt kasutatava tehnoloogia tüüpmustrile. • Failide siseselt võtme (key) struktuuri loomiseks eraldaja "." (nt "sub-module.object.propety") või vastav JSON|YAML struktuur. • Ainult asutusesiseseks kasutamiseks mõeldud ressursid ei tohi olla kättesaadavad avalikult.	Täpsustusi: • Loogiliste moodulite kattuv osa võib olla ühises failis nimega "common_<keel>.[properties|json|yaml]". • Faili nime muster on IETF BCP 47 alamhulk. Regex: "^.+_[a-z]{2}(-[A-Z]{2})?\.(properties|json|yaml)$". Näiteid: backoffice_et.properties, backoffice_en-GB.json, frontend_en.yaml. • Failide asukoha tüüpmustrite näiteid: [<module>/]/src/main/resources/i18n/ (Java), /src/assets/i18n/ (frontend). • Failide sisemiste võtmete näiteid: sub-module.object.property=something, "sub-module.object.property":"something". • Asutusesiseseks kasutamiseks mõeldud ressursside kaitseks peavad backoffice ressursid olema frontoffice ressurssidest eraldi, va ühisosa (common).	

Turva	178	OIDC/OAuth2 autentimisvoogudest tuleb inimkasutaja puhul kasutada voogu "Authorization Code Flow". Eelistatult tuleb kasutada "confidential" klienti (st OAuth2 teostab tagasüsteem). Kui on vajalik kasutada "public" klienti, siis peab kasutama voogu "Authorization Code Flow with PKCE" ning juhul kui rakendus ei suuda kontrollida koodi saatnud serveri identiteeti (nt tegemist on veebilehitsejas töötava rakendusega), peab "refresh_token" saamine viima autentimise vigasesse olekusse (st tuleb järgida standardit, eeldada, et server on ründe all ja mitte aktsepteerida saadud autentimisinfot). Masin-masin autentimisvoona tuleb kasutada "Client Credentials" autentimisvoogu. Toetatud peavad olema nii mTLS autentimine kui ka "Client-secret" autentimine (kasutatav autentimismeetod peab olema seadistatav rakenduse konfiguratsioonist).	• Referents OIDC/OAuth2 IdP: Keycloak (https://www.keycloak.org/) • OIDC: https://openid.net/developers/specs/ • OAuht 2.0: https://www.rfc-editor.org/rfc/rfc6749 • OAuth 2.0 Authorization Flow with PKCE: https://www.rfc-editor.org/rfc/rfc7636 • OAuth 2.0 mTLS Client Authenticaton: https://www.rfc-editor.org/rfc/rfc8705	
Kasutajaliides	179	Veebilehitsejas töötav rakendus peab taustapäringute (XMLHttpRequest, fetch) kasutamise korral tuvastama võrguühenduse katkemise või puudumise ning sellest tingitud vigade puhul esitama kasutajale selge sellekohase veateate käitumisjuhistega. Veebilehitsejas tekkinud vead peavad olema kasutajaliideses selgelt eristatavad serveripoolsetest vigadest.	Ei ole aktsepteeritav, et võrguühenduse katkemise korral esitatakse veateade "Tehniline viga" või "Timeout". See ei ole kasutajale abiks ja teeb probleemi diagnoosimise keeruliseks. Veebilehitsejas tekkinud vigade eristamine serveris tekkinud vigadest on oluline selleks, et kasutajatugi teaks veebilehitsejas tekkinud vea korral küsida kasutajalt veebilehitseja konsooli logi.	
Platvorm	180	Rakendustes kasutatavad kolmandate osapoolte poolt loodud komponendid peavad olema üldjuhul vabalt kasutatavad. Ilma kooskõlastuseta ei tohi kasutada komponente, mis seavad kasutamisele piiranguid või panevad rakenduse omanikule või käitlejale täiendavaid kohustusi.	Piirangute näiteid: tohib kasutada ainult heategevuse eesmärgil, tohib kasutada ainult teatud mahtude piires, peab levitama tasuta, peab avaldama lähtekoodi. Kohustuse näide: kohustus maksta litsentsitasu. Mõnd tasulist komponenti võib olla otstarbekas kasutada ning neid tasub uurida ja välja pakkuda, aga enne kasutamisele võtmist tuleb kooskõlastada RmITiga.	

Üldine	181	Andmebaasides tuleb ajatemplid salvestada koos ajatsooniga, mis peab olema UTC. API päringute vastustes peab ajatemplil olema küljes ajatsooni märge.	Teisisõnu, andmebaasides tuleb ajatemplid salvestada ajatsoonitundlikuna, kusjuures salvestuv väärtus peab olema UTC ajatsoonis. Niimoodi käitub "timestamp with time zone" andmetüüp PostgeSQL-s. PostgreSQL-s võtavad ajatempli andmetüübid ajatsooniga ja ajatsoonita sama palju salvestusruumi. Normaalseks koostoimeks legacy-ga saab kaaluda erandeid. Andmebaasi piires tuleb kasutada ühtset lähenemist. Näiteks tuleb samas andmebaasis vältida ajatemplite salvestamist erinevate põhimõtete järgi.	