Versioon: 2.3
Dokument määrab kvaliteedi- ja mittefunktsionaalsed nõuded (MFN) uutele infosüsteemidele ning nende dokumentatsioonile. Dokumenti hoiavad ajakohasena Tervise ja Heaolu Infosüsteemide Keskuse (TEHIK) arhitektid.
Dokumenti tuleb vaadata kui arenduste kvaliteedi- ja mittefunktsionaalsete nõuete põhidokumenti. Põhidokumendi ja viidatud dokumentide erisuste puhul tuleb lähtuda põhidokumendis kirjeldatust. Põhidokumendis viidatud TEHIKu koostatud dokumentide ja kolmandate osapoolte koostatud dokumentide erisuste puhul tuleb lähtuda TEHIKu dokumentides kirjeldatust.
Kui mõnda nõuet ei ole võimalik või otstarbekas täita, tuleb selle mittetäitmise fakt ja põhjendus välja tuua pakkumuse esitamisel.
Nõudeid tuleb järgida ka olemasolevate infosüsteemide versiooniuuendustel nii palju kui versiooniuuenduse käigus võimalik.
Erandid tuleb kooskõlastada TEHIK'u vastutava arhitektiga kirjalikult taasesitataval kujul projektidokumentatsiooni juures.

Nõude nr	Nõude sisu	Seletused	Digiriigi CFR	Iseteenindus	Veebid	Karbitoode	Koostamise eest vastutaja	Testimise läbi viib või kinnitab
1. Vastavus üldistele standarditele								
1.1	Lahendus loomisel peab arvestama digiriigi ristfunktsionaalsed nõudeid.	Kohustus- ja ootustasemel olevad nõuded tuleb rakendada. https://koodivaramu.eesti.ee/e-gov/cfr Erisused tuleb kokku leppida Tellija lahenduse arhitektiga.		V	V		Arendaja	**Projektijuht** **Arhitekt** **Administraator** **Testija** **Turvatestija** **Infoturbe spetsialist** **Standardija**
1.2	Lahenduse X-tee teenused peavad vastama nõuetele.	https://X-tee.ee/docs/live/xroad/		V	V	V	Arendaja	**Testija**
1.3	Lahendus peab vastama Sotsiaalministeeriumi IT-profiilile.	Tulevase ja olemasolevate infosüsteemide platvormid (rakendusserver, andmebaas, kolmanda osapoole komponendid) ja topoloogia peavad olema loodud kooskõlas hankes viidatud IT-Profiil versioonile.		V	V	V	Arendaja	**Projektijuht** **Arhitekt** **Administraator** **Testija** **Turvatestija** **Infoturbe spetsialist** **Standardija**
1.4	Lahenduse kasutajaliides peab vastama dokumendis "Front-end arendusreeglid" kirjeldatud reeglitele.	https://tehik.ee/arendusjuhendid		V	V		Arendaja	**Arhitekt** **Testija**
1.5	Rakendus peab olema kirjutatud arvestades selle lahenduse äriprotsesside ja andmete E-ITS ja ISKE turvaklassi nõudeid.	https://eits.ria.ee/		V	V	V	Arendaja	**Turvatestija** **Arhitekt**
1.6	Veebirakenduse kasutajaliides peab vastama vähemalt WCAG 2.2 tasemele AA.	https://www.w3.org/TR/WCAG22/ CFR selgitus: WCAG 2.2 alates 05.10.2023	#39	V	V	V	Arendaja	**Testija**
1.7	Veebipõhine kasutajaliides peab ühilduma täielikult standarditega HTML 5 ja CSS 3.	Valideerimiseks kasutatakse vastavaid validaatoreid: https://validator.w3.org/ Kui on tegu olemasoleva süsteemi edasiarendusega, siis tuleb järgida olemas olevat HTML ja CSS versiooni.	#38	V	V		Arendaja	**Testija**
1.8	Allkirjastamisel tuleb kasutada Tellija SiGA/SiVa vahendusteenust.	https://www.tehik.ee/arendusjuhendid		V		V	Arendaja	**Testija** **Arhitekt**
1.9	Rakendus peab probleemideta läbima OWASP ASVS baasil põhineva testi.	Kui pole arenduses eraldi kokku lepitud teisiti, siis on OWASP ASVS tasemeks 2 (https://owasp.org/www-project-application-security-verification-standard/). Kinnise lähtekoodiga kommertstoote kasutamisel ei eeldata ligipääsu kinnisele lähtekoodile. Tellijapoolset turvatestimist teostab kolmas sõltumatu osapool. Selline esmane kolmanda osapoole turvatestimine tellitakse Tellija finantseeringul. Ilmnenud vigade korral ja peale nende parandamist peab järeltestimise rahaliselt kompenseerima arendaja, kui Tellija vastava nõudmise esitab.	#9 #10	V	V	V	Arendaja	**Turvatestija**
1.10	Krüptoalgoritmite ja räsifunktsioonide kasutamisel tuleb kasutada turvalisi algoritme ja võtmepikkuseid.	Krüptoalgoritmite ja räsifunktsioonide kasutamisel tuleb järgida uusimat RIA kodulehel avaldatud krüptograafiliste algoritmide kasutusvaldkondade ja elutsükli uuringut. Värskeima uuringu leiab aadressilt https://www.ria.ee/amet-uudised-ja-kontakt/uudised-pressikontakt/uuringud-ja-analuusid#kruptouuringud Arendaja loodud lahenduse dokumentatsioonis (nt detailanalüüs vms) tuleb välja tuua kasutatavad krüpto- ja räsialgoritmid, nende võtmepikkused, kasutuskohad, sh sertifikaatide kasutuskohad.	#46	V	V	V	Arendaja	**Arhitekt** **Administraator** **Turvatestija**
1.11	Andmete edastus peab olema kaitstud kasutades krüpteeritud ning vajadusel autenditud ja autoriseeritud kanalit.	Autentimist ei ole vaja ainult avalike andmete edastamisel (nt avaandmed).		V	V	V	Arendaja	**Turvatestija** **Arhitekt**

		Autentimise mehhanism tuleb kokku leppida Tellijapoolse arhitektiga.						
1.12	Infosüsteem peab kasutama serveri kellaaega.	Kõik mahakirjutatavad ja talletatavad kellaajad tuleb salvestada UTC ajatsoonis koos ajatsooni infoga. Kasutajatele mõeldud kuvades tuleb kasutada sirviku ajatsooni. Aja esitamisel tekstikujul lähtuda standardist ISO 8601.	#51	V	V	V	Arendaja	**Arhitekt** **Administraator**
1.13	Süsteemi edasiarendamisel/loomisel peab arvestama selle võimaliku laiendamisega nii andmemahtude kui ka kasutajate arvu osas.	Süsteemi jõudlus peab vastama kokkulepitud topoloogial eelanalüüsi ja lähteülesande käigus välja toodud jõudlusnäitajatele.		V	V	V	Arendaja	**Arhitekt** **Testija**
1.14	Rakendus peab olema tehniliselt tükeldatud vastavalt loogilisele jaotusele. Saadud osised peavad olema eraldi versioneeritavad ja paigaldatavad.	Lahenduse arhitektuuris kasutada domeenist juhitud disaini ja mikroteenuste põhimõtteid. Näiteks kui rakendus on eraldi turvakontekstidega liidesed ametnikule ja kodanikule, peab rakendus olema jagatav kaheks eraldi liidesekomponendiks ning nende mõlema poolt kasutatavaks andmebaasiks. https://learn.microsoft.com/en-us/archive/msdn-magazine/2009/february/best-practice-an-introduction-to-domain-driven-design https://microservices.io/	#21	V	V		Arendaja	**Arhitekt**
1.15	Avalike e-teenuste loomisel peab arvestama valitsusasutusele kehtestatud visuaalse identiteedi stiilijuhiseid.	https://riigikantselei.ee/valitsuslogo	#36	V	V	V	Arendaja	**Testija**
1.16	Avalike e-teenuste loomisel peab arvestama Veera disainisüsteemiga.	https://veera.eesti.ee/ Näiteks kasutada ja laiendada E-Gov CVI projekti komponente: https://e-gov.github.io/cvi/	#36	V	V		Arendaja	**Arhitekt**
1.17	Lahenduse loomisel peab arvestama riiklikku koostoimeraamistikuga.	https://digiriik.eesti.ee/koostoimeraamistik/		V	V	V	Arendaja	**Arhitekt** **Administraator**
1.18	Aadressiandmete sisestamisel, kuvamisel ja hoidmisel tuleb lähtuda Vabariigi Valitsuse määrusest "Aadressiandmete süsteem".	Liidestatakse Maa-ameti ADS teenusega. Esitluskihis on lubatud liidestada In-ADS teenusega otsingu tarvis. Taustsüsteemides toimub liidestamine X-tee teenusega. https://www.riigiteataja.ee/akt/115072023005?leiaKehtiv	#53	V	V		Arendaja	**Arhitekt** **Standardija** **Testija**
1.19	Tegevusalade andmete sisestamisel, kuvamisel ja hoidmisel tuleb lähtuda Vabariigi Valitsuse 10. jaanuari 2008. a määrusest nr 11 "Klassifikaatorite süsteem" ja kasutada EMTAK infosüsteemis kehtivat klassifikaatorit.	https://www.riigiteataja.ee/akt/12910889?leiaKehtiv https://emtak.rik.ee/EMTAK/	#54	V	V		Arendaja	**Standardija** **Testija**
2. Nõuded rakenduse arhitektuurile								
2.1	Rakenduse, andmebaasi ja kolmanda osapoole komponendid peavad olema sellised, mille turbe uuenduste eluea lõpp (Security Support) pole teadaolevalt vähem kui 2 aasta pärast.	Arendaja loodud lahenduse dokumentatsioonis (nt analüüs vms) peab olema välja toodud kasutatavate komponentide nimetused ja versioonid. Lubatud on kasutada ka tarkvara materjalide loendit (SBOM). Versiooni eluea lõppu ei loeta võrdseks terve komponendi eluea lõpuga, st versiooni tugi võib aeguda, kui uus versioon on välja lastud. Jätkuarenduse puhul tuleb kaardistada eelneva arendusperioodi komponentide kaardistus. EOL komponentide kasutamisest tuleb teavitada Tellijapoolset arhitekti. CFR selgitus: Kahjuks täna paljud komponendid vaatavad tulevikku 1.5 - 2 aastat. Võimalusel valida komponendid, mis omavad pikka elueatuge, nn LTS versioon.	#7	V	V	V	Arendaja	**Arhitekt**
2.2	Rakendusserver peab võimaldama töötamist andmebaasiserverist eraldi serveril.			V	V	V	Arendaja	**Administraator**
2.3	Rakendusserver peab olema kõrgkäideldav ja horisontaalselt skaleeruv.	Kasutaja sessioonid ei tohi olla rakenduserveri klastri õla põhised.		V	V	V	Arendaja	**Administraator**
2.4	Rakendust peab saama ilma ümberprogrammeerimata liigutada erinevate domeenide ja domeeni saitide vahel.	Lahenduses ei tohi olla sisse kompileeritud absoluutseid URI-sid.		V	V	V	Arendaja	**Administraator**
2.5	Rakenduse komponentide konfiguratsiooni peab olema võimalik ette anda käivitamisel. Konfiguratsiooni muudatus peab olema teostatav ilma rakendust kompileerimata.	Rakendus peab neid sealt ka kasutama (mitte kopeerima parameetreid käivitamisel kolmandatesse kohtadesse), logimise seaded võivad olla rakenduse konfiguratsioonifailist eraldi ühes lisakonfiguratsioonifailis (nt Log4j). Samuti on tungivalt soovituslik eraldi konfiguratsioonifailis hoida arendaja ja administraatori vastutusala parameetreid. Infosüsteem peab olema seadistatav konfiguratsiooniparameetrite(de) abil. Konfiguratsioonifailiks ei saa lugeda faili, kus hoitakse lisaks konfiguratsioonile ka muud programmikoodi.		V	V	V	Arendaja	**Administraator** **Testija**

		Näiteks konteinerlahenduste puhul peab kasutama keskkonnamuutuja põhiseid konfiguratsiooni parameetreid.						
2.6	Rakenduse taaskäivitus, konfiguratsiooni muutmine vms peab toimuma mõistliku aja jooksul.	Tavaline käivitusaeg ei tohi ületada 30 sekundit. Kui rakendus vajab indekseeritud sisu ja see pole kättesaadav, siis peab rakendus väljastama selle kohta selge teate.		V	V	V	Arendaja	**Testija**
2.7	Lahenduse väliste osapoolte komponentide konfiguratsioonid peavad olema puhverdatud.	Kui rakendusel või mõnel selle komponendil on tihti kasutatav teenus ning sellel teenusel on laetav konfiguratsioon, siis tuleb: • laadida konfiguratsioon ühekordselt ja seda korduvkasutada; • konfiguratsiooni automaatselt ja regulaarselt värskendada; regulaarsuse tarbeks peab saama määrata intervalli, millise aja järel või täpsed kellaajad, millal konfiguratsiooni värskendatakse; • luua võimalus värskendada konfiguratsiooni käsitsi. Näiteks: • digidoc4j teegi korral laetakse TSL nimekiri välisvõrgust puhvrisse, et vähendada koormust kolmandale osapoolele. • OpenID konfiguratsioon		V	V		Arendaja	**Arhitekt** **Testija** **Administraator**
2.8	Kõik andmed, andmebaasid, SQL skriptid, lähtekood ja rakendus peavad kasutama UTF-8 või UTF-16 kodeeringut.		#50	V	V	V	Arendaja	**Administraator** **Testija**
2.9	Rakendusserveri failisüsteemi ei tohi salvestada midagi püsivaks kasutamiseks.	Näiteks objektide talletuseks kasutada objektide talletamise lahendust (nt MinIO).		V	V		Arendaja	**Administraator**
2.10	Ühest relatsioonilise andmebaasi andmetabelist teise viitamisel tuleb kasutada väliseid võtmeid (Foreign key).	Erinevate skeemide vahelised ühendused on keelatud. Peab kasutama REST/SOAP/AMQP liidestust.		V	V		Arendaja	**Arhitekt**
2.11	Kõik välised võtmed (Foreign Key) peavad olema indekseeritud.	Andmebaasis peab kasutama indekseid ja/või muid meetmeid, et nõuded rakenduse jõudlusele oleksid täidetud ka tulevikus. (ühe, kolme, viie või 10 aasta pärast – vastavalt planeeritud kasutusajale).		V	V		Arendaja	**Arhitekt**
2.12	Tuleb kasutada päringumuutujaid (Parameter Binding).	SQL päringute väljakutsumisel väljastpoolt andmebaasi peab kasutama päringumuutujaid, et vältida SQL vahemälu fragmentseerumist (When calling SQL code from outside the database, Parameter Binding should be used to prevent SQL cache fragmentation).		V	V		Arendaja	**Arhitekt**
2.13	Kõigis andmebaasi tabelites peab olema defineeritud üks primaarvõti	Kasutada tuleb vastava andmebaasisüsteemi nimetamise parimaid praktikaid.		V	V		Arendaja	**Arhitekt**
2.14	Andmebaasi objektide nimetused peavad olema sisulised ja andma aimu nende otstarbest.	Kasutada tuleb vastava andmebaasisüsteemi nimetamise parimaid praktikaid.		V	V		Arendaja	**Arhitekt**
2.15	Andmebaasis defineeritakse üldjuhul kaks või enam kasutajat: • Rakenduse peakasutaja, kellena luuakse objektid ja skeemid. • Rakenduse piiratud õigustega kasutaja, kellena pöördub rakendusserver/rakendus. Objektide loomiseks vajalikud õigused ja ressursid on loetletud rakenduse dokumentatsioonis.	Need õigused, mis on vajalikud ainult rakenduse baasi loomiseks, on eraldi välja toodud ja tuleb peale installeerimist ära võtta. Karbitoodete puhul tuleb erisused läbi arutada Tellija arhitektiga. Lahenduse puhul, milles kasutatakse andmebaasi versiooneerimist, tuleb kasutada mitut andmebaasi ühendust. Ennem rakenduse käivitumist teostatakse andmebaasi skeemi muudatused eraldi kasutajaga. Rakendus kasutab enda põhitööks kasutajat, kellel puudub õigus andmebaasi skeemis muudatusi teostada.		V	V		Arendaja	**Administraator**
2.16	Failide hoidmise asukoht lepitakse iga kord kokku, kuid failid ja failide indeks peavad olema replikeeritavad teise asukohta.	Failide hoidmine klassikalises andmebaasis on kulukas ja seab kõrgendatud nõudmised ja piirangud andmebaasiserveritele. Lahenduse dokumentatsioonis tuleb ära tuua failide hoidmise asukoht. Näiteks objektide talletuseks kasutada objektide talletamise lahendust.		V	V		Arendaja	**Arhitekt**
2.17	Peab olema miinimumini viidud vajadus, et haldur teeb haldustoiminguid otse baasis. St rakendusel peab olema haldusliides, mille kaudu rakenduse haldur saab teha tavapäraseid haldustoiminguid.	Halduri haldustoimingud lepitakse Tellijaga kokku detailanalüüsi käigus.		V	V	V	Arendaja	**Testija**
2.18	Andmebaas peab toetama nii külm- kui ka kuumvaru (peegeldamist) teise asukohta.	Ei tohi kasutada teenuseid, mis välistavad andmebaasi peegeldamist (nt "MSSQL filestream").		V	V		Arendaja	**Arhitekt**
2.19	Sorteerimisreeglistik peab olema Eesti tähestikule vastav. Tõusutundlikkus peab olema välja lülitatud. Diakriitiline (Accent) peab olema sisse lülitatud.	Näiteks PostgreSQL puhul et_EE.		V	V	V	Arendaja	**Testija**
2.20	Kui infosüsteemid saadavad e-kirju, peavad nad kasutama välist e-maili serverit. Kirja saatmisel peab rakendus veenduma, et e-posti server võttis kirja vastu. E-kirjade vormindamine peab järgima interneti standardeid (RFC 5322).	Saatja ja adressaadid, pealkiri ja sisu ei tohi olla rakendusse kodeeritud, vaid on muudetavad konfiguratsioonifaili kaudu. Genereeritud kirjade puhul peab tagama kirjade jälitatavuse (näiteks lisada X-päise kodeeritud kirje, milles on kirjeldatud, mis protsess/skriptifail/kasutaja kirja genereeris jms abistav info).		V	V	V	Arendaja	**Administraator**
2.21	Konfiguratsiooniparameetrite nimed peavad olema sisulised. Kui see ei ole	Näiteks : X_TEE_TURVASERVER, mitte XTTS või viitenumber, mitte vk_seb jne		V	V		Arendaja	**Administraator**

	võimalik, siis peab kõrval olema seletus.							**Testija**
2.22	Infosüsteemides on eessüsteemid (front end; presentatsiooni kiht) ja tagasüsteemid (back end; äriloogika kiht) arhitektuuriliselt selgelt lahutatud ja eraldi paigaldatavad. Rakenduse äriloogika tuleb realiseerida andmebaasist eraldi sõltumatus rakenduskihis.	Koostöövõime raamistik 2011. Punkt 3.1. Tagasüsteemide ülesanneteks on andmete haldamine ja võrguteenuste pakkumine. Tagasüsteemid ei tegele lõppkasutaja autentimise ja autoriseerimisega. Lõppkasutaja autoriseerimise tagavad eessüsteemid. Välise süsteemi tõrge tohib mõjutada ainult sellest otseselt sõltuvate kasutuslugude toimimist. Välise süsteemi taastumisel peab süsteem olema suuteline oma tööd jätkama taaskäivitamata. Andmebaas ei tohi sisaldada äriloogikat, mis muudab andmetabelites olevaid/sinna kirjutatavaid andmeid, va trigerid, mis tekitavad logi.	#23 #29 #64	V	V		Arendaja	**Arhitekt**
2.23	Konfiguratsioonifailid peavad olema vastavalt rakendusserveri tüübile vaikimisi kaitstud failid/objektid.	Näiteks IIS: *.config , *.resources Apache: *.conf, .htaccess. Arendaja peab välja tooma konfifailide listi, kui neid on mitu.		V	V	V	Arendaja	**Administraator**
2.24	Rakenduse failid, mida kasutaja näha ei tohi, peavad olema vaikimisi kaitstud kaustades ja ei tohi olla veebi juurkaustas.	Näiteks: IIS: Bin,App_Code, App_Data, App_Browsers, App_GlobalResources, App_LocalResources, App_Themes, App_WebReferences, .git		V	V	V	Arendaja	**Administraator**
2.25	Konfiguratsiooniparameetrite taaskasutus. Erinevaid sama sisuga parameetreid ei tohi konfiguratsioonis eksisteerida.	Kõiki parameetreid tuleks konfiguratsioonis kirjeldada vaid korra, dubleerimine on keelatud.		V	V		Arendaja	**Administraator** **Testija**
2.26	Esitluskihist ei tohi pöörduda otse andmebaasi poole.	Tuleb rakendada vähemalt 3 tasandilist arhitektuuri (three-tier architecture). Lõppkasutaja lokaalsesse seadmesse paigaldatud tarkvara loeme esitluskihiks. Ka sellisel juhul ei tohi teha lõppkasutaja seadmest otse ühendust andmebaasi.		V	V	V	Arendaja	**Administraator**
2.27	Keskkonnapõhised muutujad peavad olema konfiguratsiooniparameetritega seadistatavad.	Näiteks WSDL ei tohi sisaldada viiteid arendusserveritele.		V	V	V	Arendaja	**Administraator** **Testija**
2.28	Eelistada tuleb tsentraalseid autentimislahendusi (nt Tellija SSO lahendus).	Kui rakendus realiseerib ise autentimist, siis peab olema võimalik piirata ebaõnnestunud logimisi ajaühiku kohta (mobiil-ID, paroolid) ühelt IP-aadressilt. • Eelistama peaks IP-aadressipõhist blokeeringut. Erandina Tellijaga kokkuleppel võib kasutada captchat või konto lukustamist. Blokeeringute ajavahemikku ja logimiskatsete arvu peab saama konfiguratsioonifailist muuta. • Rakenduses realiseeritav autentimise lahendus peab olema põhjendatud ja omama kirjalikku taasesitatavat kokkulepet projekti dokumentatsiooni juures	#30	V	V	V	Arendaja	**Arhitekt** **Testija**
2.29	Relatsioonilises andmebaasis võib kasutada vaid ISO/IEC 9075 standardiga kaetud funktsionaalsusi. Lisaks ei tohi kasutada ka sama standardi osas 13 kirjeldatud funktsionaalsusi.	• Ei ole soovitav kasutada mingit platvormispetsiifilist lahendust, mille üleviimine mõnele muule andmebaasiplatvormile ei ole võimalik. • ISO/IEC 9075 osa 13 spetsifitseerib Javas kirjutatud programmimoodulite kasutamist andmebaasis.		V	V		Arendaja	**Arhitekt**
2.30	Uniform resource identifier (URI) pikkus ei tohi ületada ühegi lahenduse poolt toetatava sirviku maksimaalset lubatud väärtust.	Harilikult on piiriks 2048 tähemärki, kuid iga lahenduse puhul tuleb seda eraldi järele uurida sõltuvalt lahenduse komponentidest. Asjakohased viited: RFC 3986 ja RFC 7239.		V	V	V	Arendaja	**Administraator** **Testija**
2.31	Veebiteenuseid (REST, SOAP) pakkuv rakendus peab olema üles ehitatud nii, et see toetaks teenuste versiooneerimist URL-i ja/või skeemi tasemel.	Näiteks WSDL puhul: Alajaotis definitions/types/schema: • complexType defineerimisel tuleb sellele lisada any element.		V	V	V	Arendaja	**Arhitekt**
2.32	Rakendus peab olema võimeline töötama koormusjaoturitega varustatud taristul.	Koormusjaoturi peal kasutatakse järjestikplaanurit (Round Robin) päringute suunamisel. Samuti võidakse teostada koormusjaoturil TLS ühenduse lahtivõtmist ja uuesti kokkupanemist (SSL offload).		V	V	V	Arendaja	**Administraator**
2.33	Sidusinfosüsteemide mittekättesaadavus ei tohi segada rakenduse töötamist. Sidusinfosüsteemidega andmevahetamisel tekkinud vead logitakse ja kasutajat hoiatatakse.	Sidussüsteemi tõrge tohib mõjutada ainult sellest otseselt sõltuvate kasutuslugude toimimist. Sidussüsteemi taastumisel peab süsteem olema suuteline oma tööd jätkama rakendust taaskäivitamata. Väliste liidestatud süsteemide tõrke korral ei tohi süsteem hanguda, vaid peab väljastama mõistliku (võimalikult lühikese) aja jooksul asjakohase veateate. Võimalusel tuleb kasutada asünkroonseid liideseid.		V	V	V	Arendaja	**Administraator** **Testija**
2.34	Automaatselt käivituvaid taustatöid peab saama käsitsi (taas)käivitada.	Vajalik juhul, kui automaatsel käivitumisel on tekkinud viga ja/või taustatöö on pooleli jäänud. Pärast vea põhjuse korrigeerimist peab saama taustatöö uuesti käivitada. Lahendusse on vaja luua taustatööde haldamiseks ja juhtimiseks võimekus, et võimaldada vastavates rollides olevatel inimestel lahendust hallata (nt peakasutaja või rakenduse administraator).		V	V	V	Arendaja	**Testija**
2.35	Kui ajastatult käivitatav taustatöö ei ole mõeldud käima paralleelselt, peab selles olema realiseeritud kontrollmehhanism, mis tagab, et sama taustatööd ei ole	Kui lahendus töötab mitmel õlal, ei tohi tööd, mis ei ole mõeldud paralleelselt käima, käivituda korraga mitmel õlal. Peab rakendama *lukustus* põhimõtet.		V	V		Arendaja	**Administraator** **Testija**

	võimalik käivitada uuesti enne, kui eelmisena käivitatud instants on oma töö lõpetanud.							
2.36	Uue toote arenduse ja olemasolevate infosüsteemide versiooniuuendustel kasutusele võetavate tehnoloogiate ja standardite valik tuleb kooskõlastada Tellijapoolse arhitektiga.			V	V	V	Arendaja	**Arhitekt**
2.37	Rakenduse ühenduste (s.h. andmebaasi ja sidusinfosüsteemide ühendused) realiseerimisel tuleb kasutada ühenduste puulimist (connection pooling).	Implementeeritud peab olema vähemalt maksimaalsete ühenduste arvu piirang, päringu aegumise aeg (request timeout) ja ühenduse elususe periood (keepalive). Rakenduse ühenduste tõrge tohib mõjutada ainult sellest otseselt sõltuvate kasutuslugude toimimist. Ühenduste taastumisel peab rakendus olema suuteline oma tööd jätkama taaskäivitamata. Tekkinud vead logitakse ja kasutajat hoiatatakse.		V	V	V	Arendaja	**Arhitekt**
2.38	Rakenduse uuendustega kaasnevad andmebaasi muudatused tuleb automatiseerida ja versioneerida.	Näiteks Liquibase või Flyway.		V	V		Arendaja	**Administraator**
2.39	Mitterelatsioonilises mudelis andmete hoiustamine tuleb eraldi kokkuleppida.	Näiteks kui soovitakse kasutada NoSQL lahendusi püsivaks andmete talletuseks, tuleb see kokku leppida Tellijapoolse arhitektiga.		V	V	V	Arendaja	**Arhitekt**
2.40	Mikroteenuste arhitektuuris vältida ebavajalikku andmete dubleerimist.	Iga teenus vastutab enda valdkonna andmete eest. Kui on vajadus lisaandmete jaoks, on tal võimalik pöörduda teise teenuse poole.		V			Arendaja	**Arhitekt**
2.41	Infosüsteemide vaheline andmevahetus toimub üle X-tee.	https://www.ria.ee/riigi-infosusteem/andmevahetuse-platvormid/andmevahetuskiht-X-tee https://www.riigiteataja.ee/akt/106082019017?leiaKehtiv Erandiks on lubatud päringud sama andmekogu raames. Sellisel juhul tuleb turvalisus tagada lahenduse loojate poolt. Kasutada kas mTLS või tõendipõhist autentimist.	#22	V	V	V	Arendaja	**Arhitekt**
3. Turvalisuse tagamisega seotud nõuded								
3.1	Asutusesiseseks kasutamiseks mõeldud rakenduse kasutajate autoriseerimist peab saama teha vastu TEHIKu keskset autoriseerimisteenust.	TEHIKu haldusala kasutajad ja nende rollid on kirjeldatud Active Directory's. Võimalik on kasutada rollide pärimiseks TEHIK SSO teenust. Täpsem tehniline lahendus leida koos TEHIKu poolse arhitektiga.		V	V	V	Arendaja	**Turvatestija**
3.2	Kliendi ja serveri vahel peab autenditud kasutajasessioonide korral olema sessioon krüpteeritud HTTPS-protokolli kasutades.			V	V	V	Arendaja	**Turvatestija**
3.3	Rakendus tohib kasutada vaid sessiooni küpsiseid (cookies). Muude küpsiste kasutamine tuleb kokku leppida Tellijapoolse arhitektiga.			V	V		Arendaja	**Turvatestija**
3.4	Kui andmebaasis olevate andmete E-ITS tervikluse (I ehk *integrity*) turvaosaklass on S või VS, siis tuleb kõik andmebaasi kirjed/tabelid versioneerida.	St 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) • X-tee ID (kui on olemas) Iga kehtetuks tunnistatud kirje peab omama järgmist informatsiooni; • kasutaja, kes kirje kehtetuks tunnistas; • kirje kehtetuks tunnistamise aeg. Täpne realisatsioon tuleb kokku leppida Tellija arhitektiga.	#55 #49	V		V	Arendaja	**Turvatestija** **Arhitekt**
3.5	Rakendusega peab kaasas olema lahendus, mis suudab toota toodangu andmetest testandmed, mis ei võimalda siduda konfidentsiaalset informatsiooni päris andmesubjektiga.	Testandmed peavad säilitama kõik toodangu andmete omadused (pikkuse, tüübi) ja omavahelised suhted. Täpsem vajadus ja tegevusplaan tuleb koostada Tellija arhitekti ja tooteomanikuga.		V			Arendaja	**Arhitekt**
3.6	Rakendus ja selle komponendid peavad võimaldama kasutada keskkondade lahusust.	Arendaja arendab arenduskeskkonnas ja annab tarne üle Tellijale paigalduspakkidena. Tellija paigaldab selle testkeskkonda ja testib ning seejärel paigaldab tarne toodangu keskkonda. Reaalseid andmekogu andmeid tohib töödelda üksnes toodangu keskkonnas. Üldjoones on kõik keskkonnad majutatud Tellija majutuses.		V	V	V	Arendaja	**Turvatestija**
3.7	Rakendusse ja andmetele tohib olla ligipääs vaid dokumenteeritud ja tellimuses kirjeldatud teid mööda ning dokumenteeritud autentimisprotseduure kasutades.	St rakendustes ega andmebaasides ei tohi olla ligipääsemiseks teisi võimalusi.		V	V	V	Arendaja	**Turvatestija**
3.8	Rakendus ei tohi teostada X-tee päringut otse kasutajaarvutist.	Kasutajaarvutitest otse X-tee päringute tegemine on arvutivõrgu tasemel kinni.		V	V	V	Arendaja	**Turvatestija**
3.9	Veebipõhised välise veebilehega rakendused peavad kasutama vahendeid,	IIS puhul peab kasutama näiteks URL scan, apache puhul modsecurity või vastavat tööriista. Lubamatud		V	V	V	Administraator	**Turvatestija**

	kaitsmaks rakendust lubamatute päringute eest.	päringud on kõik päringud, mis ei ole detailanalüüsi käigus vastavalt kasutusjuhtudele ette nähtud. Blacklistingu asemel tuleb kasutada whitelisting põhimõtet.						
3.10	Kasutaja peab saama soovi korral veenduda, kas keegi pole tema nime all vahepeal sisse loginud.	Rakendus peab sisenemisel näitama pärast õnnestunud sisselogimist eelmise õnnestunud sisselogimise aega. Kui on toimunud ebaõnnestunud sisselogimise katseid, siis peab ka kuvama, millal need toimusid, mitu neid oli ja mis IP-aadressilt pöörduti. Ebaõnnestunud logimiste katsete kuvamise nõue kehtib juhul, kui autentimine ja autoriseerimine lahendatakse rakenduses lokaalselt.		V	V	V	Arendaja	**Turvatestija**
3.11	Kõigil rakendustel peab olema konfigureeritav kasutajasessiooni aegumise aeg.	Aeg peab olema muudetav koos teiste konfiguratsiooniparameetritega. Nõue kehtib juhul, kui kasutatakse lahenduse sisest sessiooni haldust.		V	V	V	Arendaja	**Turvatestija**
3.12	Lahenduses kasutatavate küpsiste sisu peab olema krüpteeritud	Eesmärk on kaitsta kasutaja andmeid, mis on talletatud sirviku küpsiste hulka.		V	V		Arendaja	**Turvatestija**
3.13	LDAP lahenduse (nt Active Directory) kasutamisel peab rakendus kasutama kontoga kaasnevaid piiranguparameetreid.	Näiteks: konto on lukus, parool aegunud, konto aegunud, paroolipoliitika jne.	#30	V	V	V	Arendaja	**Turvatestija**
3.14	Tagada tuleb rakenduse rollide lahusus.	Peakasutajal ja tavakasutajal on erinevad tööülesanded. Rollide/õiguste kirjeldus peab lähtuma detailanalüüsist ja kasutusjuhtudest.		V	V	V	Arendaja	**Turvatestija**
3.15	Arendus peab olema orienteeritud toodangukeskkonnas toimimiseks.	Toodangukeskkonnas mittevajalikud funktsionaalsused peavad olema eraldi juhitavad ja tavakäivitusel väljalülitatud. Näiteks eraldiseisva profiiliga Java arenduste puhul. (kasutuseta funktsionaalsus ja komponendid, mis on mõeldud testimiseks testkeskkonnas ja arendusabiks arenduskeskkonnas)		V	V	V	Arendaja	**Turvatestija**
3.16	Kui rakenduse tervikluse turvaosaklass on T3, peavad tõestusväärtust omavad andmed olema kas ajatembeldatud, digiallkirjastatud või digitembeldatud ning krüptoaheldatud.	See tagab, et tõestusväärtusega andmeid ei saaks märkamatult kustutada. Konkreetne lahendus tuleb kokku leppida Tellija arhitektiga.		V	V	V	Arendaja	**Turvatestija**
3.17	Kui rakenduses on S3 salastatuse astmega andmeid, peavad need olema nii transpordi ajal ja ka salvestatult alati krüpteeritult.	Konkreetne lahendus tuleb kokku leppida Tellija arhitektiga.		V	V	V	Arendaja	**Turvatestija**
3.18	Rakendus peab võimaldama hõlpsalt välja vahetada aegunud ja ebaturvalise krüptoalgoritmi.	Krüptograafiat kasutav rakenduskood ei tohi nimeliselt välja kutsuda krüptograafilisi algoritme, vaid peaksid seda tegema vahendavate vaheteekide kaudu üldiste funktsioonide järgi (nt krüpteerimine, dekrüpteerimine, signeerimine, signatuuri verifitseerimine jne). Dokumentatsioon peab kajastama üldist kirjeldust, kuidas vajadusel ebaturvaline krüptoalgoritm välja vahetada. Lisaks peavad eksisteerima vahendid juba olemasolevate krüpteeritud andmete ümberkrüpteerimiseks.		V	V	V	Arendaja	**Turvatestija**
3.19	Rakenduse andmebaasi krüpteerimisega seotud andmeväljad peavad olema muudetava pikkusega.	Andmebaasides kasutatavad krüpteerimisfunktsioonidest tingitud lisaväljad peaksid olema muudetava pikkusega, et formaati muutmata saaks kasutada teistsuguste parameetritega krüpteerimisalgoritme.		V	V	V	Arendaja	**Turvatestija**
3.20	Lahendus peab olema kaitstud HTMLi süstimiste eest.	OWASP soovitab kasutada DOMPurify lahendust HTMLi saneerimiseks	#44	V	V	V	Arendaja	**Arhitekt** **Turvatestija**
			4. Logimine					
4.1	Logimiseks tuleb kasutada standardseid komponente kogu logiahela ulatuses.	Näiteks Java raamistikku log4j, SLF4J, logback; transpordiks syslog, Elastic Beats; logi formaadiks JSON. Logi peab olema loetaval tekstilisel kujul, et logikirjeid saaks töödelda masinmõistetavalt ja inimloetavalt.		V	V	V	Arendaja	**Arhitekt** **Administrator**
4.2	Peab kasutama logikomponenti ja peab olema võimalik juhtida logikomponendi seadistusi.	Seletus: Näiteks peab saama muuta logimise taset ja logimise formaati.		V	V	V	Arendaja	**Arhitekt** **Administraator** **Testija**
4.3	Logisündmused peavad olema loogiliselt eristatavad.	Auditlogi (Seansilogi, tegevuslogi) - info sisselogimiste, väljalogimiste ja seansi aegumiste kohta. Vigased sisselogimise katsed. Info õiguste suurendamise kohta. Peab olema logitud ka tühja või puuduvate parameetritega logimise katsed. Kogu informatsioon kasutajate tegevuste kohta koos tegevuse tüübi, seansi parameetrite (korreleerimaks seansi- ja tegevuslogi) ja kasutaja poolt esitatud sisendparameetritega (sh. väliste ressursside kasutamise kohta). Logida tuleb nii õnnestunud kui ka ebaõnnestunud tegevusi. Tehniline logi - rakendusserveri poolt loodud logi Vealogi - erinevate veaolukordade info Silumislogi - arendajate jaoks vajalik debug info	#41	V	V	V	Arendaja	**Arhitekt** **Administraator**
4.4	Logimine peab olema optimeeritud.	Informatsiooni dubleerimist logides tuleb vältida, kui ei ole nõutud teisiti.		V	V	V	Arendaja	**Arhitekt** **Testija**

4.5	Logides peab olema maksimaalselt üks sündmus ühel real.			V	V		Arendaja	**Administraator** **Testija**
4.6	Logikirje peab olema JSON formaadis.			V	V		Arendaja	**Arhitekt** **Testija**
4.7	Logiväljade nimed peavad olema normaliseeritud ja tuleb rakendada Elastic Common Schema spetsifikatsiooni.	Samatüübilised logiväljade nimed peavad olema ühtsed üle logi. https://www.elastic.co/guide/en/ecs/current/ecs-reference.html		V	V		Arendaja	**Arhitekt** **Infoturbespetsialist**
4.8	Rakendus peab logima kasutaja edukat ja ebaedukat autentimist ja sessiooni lõpetamist, kasutaja IP-d ja autentimismeetodit.	Logima peab ka autentimise ebaõnnestumise koos põhjusega (vale juurdepääsumandaat, aegunud konto jne). Logida tuleks IP-aadress, meetod ja kui võimalik kasutajatunnus (mobiil-ID puhul telefoni number; ID-kaardi või Smart-ID puhul isikukood). Kui rakendus kasutab kasutajate autentimiseks välist autentimise/autoriseerimise vahendit, siis leppida eraldi kokku autentimise detailsus ehk mida kajastatakse autentimise/autoriseerimise vahendis ja mida rakenduses.		V	V	V	Arendaja	**Arhitekt** **Testija**
4.9	Üle terve logi peab olema kasutaja sessiooni käigus tehtud tegevusi või sama sündmust võimalik siduda loogiliselt kokku.	Tegevuste sidumiseks peab olema võimalik logikirjeid siduda ühise välja abil. Selleks ei sobi kellaaeg, IP ega isikukood. Sobib näiteks unikaalne ID, mis ei tohi olla sessiooni ID, sest seda saaks logist välja lugeda ja rünnakuks ära kasutada. Võib olla sessiooni ID räsi koos transaktsiooni ID'ga. Konkreetne lahendus tuleb kokku leppida Tellija arhitektiga.	#17	V	V	V	Arendaja	**Arhitekt** **Testija**
4.10	Andmete loomise/vaatamise/muutmise/kustutamise tegevused peavad olema kajastatud logides. Logida tuleb ka päringud, mille vastus on puhverdatud.	Logikirjes peab sisalduma piisavalt informatsiooni, et vastata küsimustele kes?, mida?, kus?, kust?, millal?, kuidas? ja tulemus. Konkreetne detailsus ja tehniline lahendus tuleb kokkuleppida Tellija arhitektiga. Näiteks on mõistlik luua audit teenus, mis annab vastavale rollile võimaluse näha ja auditeerida tegevusi.		V	V	V	Arendaja	**Testija** **Turvatestija** **Infoturbespetsialist**
4.11	Administraatorite ja haldurite poolt tehtavaid andmete vaatamised, muutmised sh kustutamised (ka otse baasis) tuleb logida. Muutmise puhul tuleb logida nii uus kui ka vana väärtus.	Lahendus peab tagama, et administraatorid/haldurid ei saa andmete vaatamise, muutmise logimist ise (ka tavakasutajate logimist) deaktiveerida või logisid kustutada/muuta. Konkreetne detailsus ja tehniline lahendus tuleb kokku leppida Tellija arhitektiga. Näiteks on mõistlik luua audit teenus, mis annab vastavale rollile võimaluse näha ja auditeerida tegevusi.		V	V	V	Arendaja Administraator	**Infoturbespetsialist** **Testija**
4.12	Süsteemsed logid ei tohi sisaldada otseseid isikuandmeid.	Tulenevalt GDPRist ja logi sündmuse subjekti õigustest, ei tohi logide igapäevane analüüsimine ja jälgimine riivata sündmuse subjekti õigusi. Näiteks kasutada kasutaja nime ja tunnuse asemel tema süsteemset ID'd. Lahendus peab sisaldama võimalust ID ümberpööramist reaalseteks andmeteks. Antud tegevus peab olema auditeeritav. Konkreetne detailsus ja tehniline lahendus tuleb kokkuleppida Tellija arhitektiga.		V	V	V	Arendaja	**Arhitekt** **Testija** **Turvatestija** **Infoturbespetsialist**
4.13	Kui parameetri väärtus on tühi, tuleb see logis märkida asendusväärtusega.	Näiteks NULL		V	V		Arendaja	**Testija**
4.14	Logis tuleb kõik mittekuvatavad (non-printable) sümbolid kodeerida.	Näiteks reavahetused -> \n, non-printable sümbolid - 0x00..0x1f, 0x7f..0xff.		V	V		Arendaja	**Testija**
4.15	Rakendus peab logima kõiki rakenduses tekkivaid tehnilisi vigu.	Logi sisaldab minimaalselt vea tekkimise aega, veakoodi, veakirjeldust (stack trace, traceback vms), võimalusel kasutaja andmeid, HTTP-, GET- ja POST-parameetreid ja nende väärtusi. Logimise detailsusrežiimi (info, warning, errog, debug) peab saama muuta.		V	V	V	Arendaja	**Administraator** **Testija**
4.16	Rakendus ei tohi X-tee päringuid salvestada rakenduse logis. Logis peab olema X-tee tunnus (ID), et saaks siduda X-tee logiga.			V	V	V	Arendaja	**Administraator** **Testija**
4.17	Rakenduse funktsionaalsuse kirjeldusega tuleb luua logimise dokumentatsioon ja loginäidised. Koos funktsionaalsuse arendamisega tuleb luua ka loodava funktsionaalsuse logimine ja selle dokumentatsioon. Dokumentatsioon peab sisaldama logis kasutatud klassifikaatorite kirjeldusi.	Mida logitakse, kuidas sündmused on logis jagatud, logiridade näited.		V	V	V	Arendaja	**Arhitekt** **Administraator** **Infoturbespetsialist**
			5. Testimine					
5.1	Rakenduse kõik üleantavad versioonid peavad enne Tellijale üle andmist olema testitud.	Testitulemused tuleb edastada Tellijale koos rakenduse üleandmisega. Vaata lisaks nõuet 5.2 ja 5.3. Testid peavad olema käivitatavad Tellija pideva integreerimise (CI) keskkonnas (nt Gitlab) ning olema dokumenteeritud, kuidas teostada testide seadistamist ja manuaalset käivitamist.	#12	V	V	V	Arendaja	**Testija**
5.2	Lahendus peab olema minimaalselt 75% ulatuses kaetud automaatsete komponenditestidega (unit test).	Käivitatakse Tellija pideva integreerimise (CI) keskkonnas (nt Gitlab) ja kaetust raporteeritakse lähtekoodi analüsaatoris (nt SonarQube).	#12 #35	V			Arendaja	**Arhitekt** **Testija**
5.3	Lahendus peab olema minimaalselt 50% ulatuses kaetud automaatsete	Käivitatakse Tellija pideva integreerimise (CI) keskkonnas (nt Gitlab).	#12 #33	V			Arendaja	**Arhitekt** **Testija**

	vastuvõtutestidega.							
5.4	Rakendusega peab olema kaasas skript jõudlustestide tegemiseks.	Jõudlustestide täpne kirjeldus 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. Jõudlustestid peavad olema käivitatavad Tellija pideva integreerimise (CI) keskkonnas (nt Gitlab) ning olema dokumenteeritud, kuidas teostada testide seadistamist ja manuaalset käivitamist. Konkreetne detailsus ja tehniline lahendus tuleb kokku leppida Tellijapoolse testija esindajaga.	#12 #34	V	V		Arendaja	**Arhitekt** **Administraator** **Testija**
5.5	Testimine toodangu andmetega on keelatud.	Testimiseks tuleb luua vastavad andmekooslused, et tagada tervik voo testimise võimekus.		V		V	Arendaja	**Arhitekt** **Administraator** **Testija**
5.6	Enne lahenduse esmast tootesse lansseerimist peab olema teostatud turbetestid ja seal välja toodud probleemid lahendatud.	Turbe testide teostamist ja tellimist koordineerib Tellijapoolne testija esindaja. Samuti tuleb kokku leppida põhimõtted, millistel juhtudel turbe testi tuleb uuesti teostada. Parendused ja lahendused tuleb kokku leppida Tellija arhitektiga.	#9 #10	V		V	Arendaja	**Arhitekt** **Administraator** **Testija**
6. Monitooring								
6.1	Rakendusel peab olema masinloetav tervise testleht (*health check*) JSON kujul.	Testlehe kättesaadavus erinevatest arvutivõrkudest peab olema konfigureeritav. Testleht peab uuendama ennast lehe pärimisel. Testleht peab sisaldama custom built rakenduse versiooni numbrit, standardsed komponendid (veebiserver, andmebaas, CMS'id jms) ei tohi oma versioone reeta. Samuti peab testlehel olema infot rakenduse (vajadusel tema erinevate osade) ja tema kõigi väliste liideste staatuse kohta (töötab, ei tööta). Rakenduse, andmebaasi ja liideste töökorda kontrollitakse testpäringute teel, mis tuleb Tellija arhitektiga kokku leppida. Testleht peab oma konfiguratsiooni võtma rakenduse üldisest konfiguratsioonist (baasistring, välised ühendused). Näiteks java Spring raamistiku puhul kasutada actuatori võimekust.	#16	V	V		Arendaja	**Arhitekt** **Administraator**
6.2	Rakendusel peavad olema elususe ja tööks valmiduse otspunktid.	Näiteks java Spring raamistiku puhul kasutada actuatori võimekust. Konteinerite orkestraatori kiht teostab nende järgi otsuseid. Antud lehekülgede sisuline poole peab kajastuma ka tervise testlehel.	#16	V			Arendaja	**Arhitekt** **Administraator**
6.3	Rakendus peab pakkuma monitooringu lehte, kus leidub informatsioon rakenduse funktsionaalsuse toimimise kohta.	Tuleb rakendada OpenMetrics spetsifikatsiooni. Monitooringu leht peab välja kuvama ka testlehel kuvatud komponentide olukorda. Näiteks kui testleht kuvab infot, et andmebaasi ühendusega on probleeme, peab see kajastuma ka monitooringu lehel. Monitooringu lehte kasutame lahenduse jälgimiseks. https://openmetrics.io/	#16	V	V		Arendaja	**Arhitekt** **Administraator**
7. Nõuded rakenduse lähtekoodile								
7.1	Lähtekoodi kommentaarid peavad kõigis lahenduse kihtides (rakenduse enda kood, andmebaas jne) olema kirjutatud inglise keeles.	NB! Nõuet ei arvestata arendustarkvara poolt automaatselt genereeritavate koodilõikude puhul – neid ei ole vaja tõlkida. Samuti ei rakendata nõuet kolmandate osapoolte poolt toodetud lähtekoodile – nt igasugu erinevad lahtise koodiga koodilõigud jms. Kui tegu on olemasoleva süsteemi edasiarendusega, siis peaks kommentaarides kasutama eelnevalt kasutatud keelt.	#2	V	V		Arendaja	**Arhitekt**
7.2	Lähtekoodi genereeritud dokumentatsioonid peavad olema selged, arusaadavad ja sisuliselt kirjeldama vastavat koodi, mille juures nad on. Lähtekoodist genereeritava dokumentatsiooniga tuleb katta kõik avalikud (public) meetodid ja funktsioonid.	Rakenduse kood peab olema piisavalt hästi dokumenteeritud, et erialast haridust omav tarkvaraarendaja on võimeline süsteemile jätkuarendusi teostama. Rakendama peab dokumenteerimisel programmeerimiskeele parimaid praktikaid. Näiteks tarkvara, mis on kirjutatud Java keeles, peab kasutama javadoc põhimõtteid ja võimekust.	#2	V	V		Arendaja	**Arhitekt**
7.3	Muutujate, tüüpide ja funktsioonide nimed peavad olema sisulised ja andma aimu nende otstarbest.	Tuleb rakendada Clean Code põhimõtteid. Kui muutuja nimetus vajab kommentaari, siis pigem muuta muutuja nimetust kommenteerimise asemel. Näiteks muutujate nimed peavad olema selged ja arusaadavad. Hea näide muutujast: elapsedTimeInDays Halb näide muutujast: etid	#2	V	V		Arendaja	**Arhitekt**
7.4	Koodis kasutatavad konstandid ja lühendid tuleb kirjutada suurte tähtedega, lähtudes kasutatava programmeerimiskeele parimast praktikast.	Nt Javas identifikaator --> ID		V	V		Arendaja	**Arhitekt**
7.5	Koodis kasutatavaid konstante ei tohi selle kasutamise kohta väärtusena			V	V		Arendaja	**Arhitekt**

	*hardcode'*da – need tuleb defineerida muutujatena ja kasutada läbi nende.							
7.6	Koodis defineeritud andmetüübid peavad olema nimetava käände ainsuses. Kõik andmemassiivid tuleb nimetada nimetava mitmuses (st igasugu collectionid, arrayd, jms).	N:Isik; Menetlus; jne. Andmebaaside struktuurikirjeldustes/andmemudelis ei tohi kasutada täpitähti.		V	V		Arendaja	**Arhitekt**
7.7	Andmetabelites sisalduvad võõrvõtmed peavad nime järgi seostuma tabeli ja väljaga millele need viitavad.	Kasutada tuleb konkreetse andmebaasisüsteemi nimetamise parimaid praktikaid. Nt kui tegu on tabelitega 'Isikud' ja 'Autod', siis seos 'isiku autod' oleks: Isikud.ID=Autod.Isik_ID		V	V		Arendaja	**Arhitekt**
7.8	Andmebaasi väljade pikkused tuleb kirjeldada sümbolites, mitte baitides.	Selle asemel, et eraldada väljale x baiti, tuleb eraldada x tähemärki. (Instead of allocating x bytes of storage for the field, x chars of storage must be allocated).		V	V		Arendaja	**Arhitekt**
7.9	Kui kokku pole teisiti lepitud, siis rakenduse kood peab olema kirjutatud vastavalt Google stiili juhendile.	https://google.github.io/styleguide/ Kui tarkvara keelel puuduvad Google stiili juhised, siis tuleb need kokku leppida Tellija arhitektiga enne kodeerimist.		V			Arendaja	**Arhitekt**
7.10	Koodi valideerimiseks kasutatakse minimaalselt Tellija lähtekoodi analüsaatorit.	Üleantavas koodis ei tohi olla kriitilisi ja kõrgemaid probleeme. IT profiil: Lähtekoodi analüüs	#10 #40	V	V		Arendaja	**Arhitekt**
7.11	Kasutuses mitteolev kood tuleb rakenduse lähtekoodist kõrvaldada.	Erandina on lubatud koodi osad, mis on valmis tehtud, aga ei ole veel kasutusse rakendatud ja on peidetud nn. funktsionaalsuste lippude (feature flag) taha.		V	V		Arendaja	**Arhitekt**
7.12	Arendamisel kasutatakse DRY ja SOLID printsiipe.	http://en.wikipedia.org/wiki/Don%27t_repeat_yourself http://en.wikipedia.org/wiki/SOLID_(object-oriented_design)		V	V		Arendaja	**Arhitekt**
7.13	Üleantavas koodis ei tohi olla paroole, mida on kasutatud arenduse käigus.	Kehtib ka siis, kui need on välja kommenteeritud. Kõik sellised paroolid tuleb asendada fraasiga "<password>".	#28 #63	V	V		Arendaja	**Arhitekt**
7.14	Üleantavas koodis ei tohi olla komponente, mille CVSS punktid on 7 ja kõrgemad ning CVE on rakendatav.	Kui CVE ei ole lahenduses rakendatav ehk tegu on lahenduse mõistes vale-positiivsega, siis võib komponendi uuendus lükkuda edasistesse etappidesse. Eeldusel, et tegu ei ole viimase üleantava tarne versiooniga. Partneri viimane üleantav versioon peab olema turbevigade vaba. Mõistlik on kõik CVE'd omavad komponendid uuendada või välja vahetada. Ajas võib mitme madalama punkti koosmõjul avalduda kriitiline turbeprobleem.	#10	V	V		Arendaja	**Arhitekt**
7.15	Üleantava lahendusega peab olema kaasas viis SBOM genereerimiseks.	SBOM ehk tarkvara materjalide loend. Mõistlik on SBOM genereerimine viia üheks järjepideva integratsiooni voo sammuks. SBOM publitseerimine ja analüüsimine tuleb kokku leppida Tellijapoolse arhitektiga. https://www.sbom.com/		V	V		Arendaja	**Arhitekt**
7.16	Tehniliste komponentide API'del eksisteerib automaatselt genereeritud dokumentatsioon.	Näiteks REST API'de puhul kasutada OpenAPI spetsifikatsiooni. https://www.openapis.org/	#20	V			Arendaja	**Arhitekt**
			8. Andmekvaliteet					
8.1	Andmekorjel kasutatakse klassifikaatoreid ja loendeid, kus need on olemas.	Vabatekstivälju tuleb vältida.	#61	V	V	V	Arendaja	**Arhitekt** **Testija**
8.2	Tekstiväljad on mõistliku suurusega.	Varchar N tähemärki, kus N on ratsionaalne kaalutlus, kui suur lahter võib olla; vältida text/long-varchar kasutust, kui see pole hädavajalik.		V		V	Arendaja	**Arhitekt**
8.3	Rakendus peab automaatselt eeltäitma kõik võimalikud andmeväljad, kui need andmed on varem riigile esitatud või kui nende väärtused on võimalik automaatselt arvutada.	Välja arvatud logimisvormi lahtrid autentimisel. Näiteks: kirje sisestamise kuupäev, kasutaja nimi, sünnikuupäev jne		V	V		Arendaja	**Testija**
8.4	Lahenduses peab olema tagatud idempotentsus.	Kasutaja sama tegevuse kordamisel ei tohi tekkida lahendusse andmeid topelt. Näiteks "salvesta" nupu korduval vajutusel ei tohi tekkida dubleeritud andmeridu.		V	V		Arendaja	**Arhitekt** **Testija**
8.5	Analüüsi tulemusena ja enne esimese arendusetapi algust peab infosüsteemi kohta olema koostatud kontseptuaalne andmemudel olemi-suhte diagrammi (*Entity Relationship Diagram*, ERD) või klassidiagrammina olemite ja nende semantika kirjeldusega: teenuse nimi ja selle ärilne kirjeldus, tabeli nimi ja selles talletatavate andmete semantika ehk äriline kirjeldus.	Peab olema dokumenteeritud projekti põhi dokumentatsiooni juures. Dokument tuleb hoida ajakohane.	#58	V			Arendaja	**Arhitekt**
8.6	Enne igat arendusetapi algust peab infosüsteemi kohta olema koostatud loogiline andmemudel ehk olemi-suhte diagramm (ERD) koos kirjeldusega: skeemi ja olemite ehk tabelite nimi ja semantika, atribuutide ehk tabeli veergude kirjeldus, sh primaar- ja välisvõtmete kirjeldus: veeru nimi, andmetüüp, kohustuslik või	Peab olema dokumenteeritud projekti põhi dokumentatsiooni juures. Dokument tuleb hoida ajakohane.	#58	V			Arendaja	**Arhitekt**

	mittekohustuslik (NULL/ NOT NULL), semantika ehk andmete tähendus.							
8.7	Füüsiline andmemudel peab iga iteratsiooni lõpus või tarne tähtajaks selle iteratsiooni või tarne ulatuses sisaldama lisaks ajakohasele kontseptuaalsele ja loogilisele andmemudelile ka andmekirjeldusi andmebaasis.	Igal skeemil, tabelil ja veerul on kommentaar andmekirjeldusega, mis vastab loogilise andmemudeli kirjeldusele.	#58	V			Arendaja	**Arhitekt**
9. Kasutajaliides								
9.1	Kasutajaliidese kõik disainiotsused peavad olema kooskõlastatud Tellijaga enne nende realiseerimist.			V	V	V	Arendaja	**Projektijuht**
9.2	Veebipõhine kasutajaliides peab olema kasutatav enamlevinud veebibrauseritega, sh nutiseadmetel (Android, IOS).	Minimaalselt Microsoft Edge, Mozilla Firefox, Chrome ja Safari arenduse testimise hetkel tootja poolt toetatud versioonid. Täpsemad nõuded dokumendis "Front-end arendusreeglid".		V	V	V	Arendaja	**Testija**
9.3	Rakenduse värviskeemi ja logo kasutamine peab vastama Tellija ametlikule visuaalsele identiteedile (CVI) ja disainijuhistele (UIG).	Kui tegemist on struktuurfondide projektiga, on lisaks nõutud ka vastav SF sümboolika. Tellija ametlikud CVI esitluspõhjad, logo kasutusjuhend ja kõik logod (ka jpg-na) küsida Tellijalt.		V	V		Arendaja	**Testija**
9.4	Kasutajaliidese kõik osad ja teated peavad olema eestikeelsed.	Kui soovitakse juurde eraldi ka muid keeli, siis see on spetsifitseeritud hankedokumentides.		V	V	V	Arendaja	**Testija**
9.5	Sisemiseks kasutamiseks tehtav rakendus peab olema graafiliselt skaleeruv ja mugavalt kasutatav Tellija töökohaprofiilis loetletud resolutsioonides.	Toetatud peavad olema töökohaprofiilis loetletud resolutsioonid. Ühegi nimetatud resolutsiooni korral ei tohi tekkida horisontaalset kerimisriba.		V	V		Arendaja	**Testija**
9.6	Kasutajaliideses toiminguni (põhi- ehk enamkasutatavad tegevused) navigeerimiseks peab kehtima 3 kliki printsiip, väljalogimiseks 1 kliki printsiip.	Kõik rakenduse kasutajaliidesest tehtavad toimingud tohivad üksteisest olla maksimaalselt 3 hiirekliki kaugusel. Toimingut ei pea nende 3 klikiga tehtud saama. Väljalogimise nupp/link peab olema ühe kliki kaugusel ja arusaadavas/intuitiivses kohas.		V	V		Arendaja	**Testija**
9.7	Kasutajaliides peab alati küsima kinnituse andmete kustutamise ja massmuutmiste kohta kui just teisiti kokku pole lepitud.			V	V	V	Arendaja	**Testija**
9.8	Rakenduse kasutamisel tekkinud veale peab kasutajaliides vastama kasutajale eestikeelse kasutajasõbraliku veateatega, mis sisaldab ka vea koodi. Veateated peavad olema hallatavad.	Veateated peavad olema sellised, mis võimaldavad IT-abil võimalikult lihtsalt tuvastada vea olemuse ja asukoha.		V	V	V	Arendaja	**Testija**
9.9	Kasutajaliides peab olema ilma rakenduse koodi muutmata tõlgitav teise keelde, v.a kui ei ole teisiti kokku lepitud.	Uue keele lisamine peab olema teostatav konfiguratsiooni failist või administreerimisliidesest. Konkreetne lahendus tuleb kokku leppida Tellija arhitektiga.		V	V		Arendaja	**Testija**
9.10	Rakenduse kasutajaliides peab teavitama kasutajat ette sessiooni aegumisest.	Etteteavitamise aeg peab olema konfigureeritav.		V	V	V	Arendaja	**Administraator** **Testija**
9.11	Kui vormile sisestatakse mahukaid andmevälju, peab kasutajaliides kokku lepitud ajavahemike järel salvetama välja sisu, et sessiooni aegumisel või võrgu katkestuse korral juba sisestatud andmed ei kaoks.	Näiteks kui vorm koosneb paljudest väiksest andmeväljadest (nt taotlus), siis jagatakse vorm etappideks ning salvestatakse vastava etapi lõpus.		V	V		Arendaja	**Testija**
9.12	Interaktiivsete vormide puhul (näiteks faili üleslaadimine) ei tohiks lehe värskendamisega tegevust korrata (faili taas üles laadida, andmeid saata, avaldust esitada).			V	V		Arendaja	**Testija**
9.13	Esilehel (sisselogimata) ja pärast kasutaja sisselogimist peab olema lihtne võimalus teavitada kasutajat muudatustest või probleemidest. Teavitus peab olema halduri poolt lihtsasti lisatav ja kasutajale märgatav.	Näiteks võimalikud teavitused: mingi süsteemi osa on vigane, tuli mingi uus funktsionaalsus, hetkel on hooldus, uuendage isikuandmeid jne. Mõistlik on hooldusteate võimekiust juhtida taustteenuse abil. Näiteks läbi esitluskihi jaoks loodud seadete REST liidese.		V	V		Arendaja	**Testija**
10. Dokumentatsioon								
10.1	Lõppkasutajatele ja avalikkusele suunatud rakenduse dokumentatsioon peab olema kirjutatud eesti keeles.	Erandiks võivad olla kolmanda osapoole komponentide (mis pole kirjutatud Tellija jaoks) dokumentatsioon. Samuti võib erandiks olla väliste osapooltega seotud projektid. Erandid tuleb kooskõlastada Tellijaga enne dokumentatsiooni koostamist.		V	V		Arendaja	**Projektijuht** **Arhitekt** **Administraator** **Testija** **Infoturbe spetsialist**
10.2	Lahendus kirjeldatakse RIHA määruse nõuete kohaselt.	https://www.riigiteataja.ee/akt/12933746?leiaKehtiv#para6	#57	V			Arendaja Projektijuht Tellija RIHA haldur	**Projektijuht**
10.3	Rakenduse dokumentatsioon peab vastama dokumendis "Nõuded infosüsteemi dokumentatsioonile" kirjeldatud nõuetele.	Dokumentatsioon peab olema versioneeritud, muutmiskuupäevadega, autori nimedega, korrektse keelekasutusega, selge struktuuriga. Dokumentatsiooni detailsus peab olema piisav, et sõltumatu kolmas tehnliste IT baasteadmistega isik		V	V		Arendaja	**Projektijuht** **Arhitekt** **Administraator**

		suudaks dokumendist vajalikke järeldusi teha (st dokument peab olema arusaadav sellele isikule, kuid näiteks paigaldusjuhise järgi toimetades ei pea ta ebaõnnestunud tarnele teostama veaanalüüsi). Täpsemad nõuded dokumendis "Nõuded infosüsteemi dokumentatsioonile".						**Testija** **Infoturbe spetsialist**
10.4	Rakenduse dokumentatsioon peab sisaldama tabelite-andmete-logide mahu kasvu arvestuslikku 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, arhiveerimise tähtaegu.		V	V		Arendaja	**Projektijuht** **Arhitekt** **Administraator** **Infoturbe spetsialist**
10.5	Iga uue versiooniga peab alati välja tooma versiooni muudatuse kirjeldused (release notes).	Release notes peab kajastama kõiki muudatusi eelmise ja uue versiooni vahel.		V	V	V	Arendaja	**Projektijuht**
10.6	Arendaja loodud lahenduse dokumentatsioonis (nt detailanalüüs vms) tuleb välja tuua kasutatavad krüpto- ja räsialgoritmid, nende võtmepikkused, kasutuskohad, sh TLS sertifikaatide kasutuskohad.			V	V		Aremdaja	**Arhitekt** **Administraator**
11. Versioonihaldus								
11.1	Kogu rakenduse testimiseks, koolituseks või implementeerimiseks üle antav lähtekood ja tarkvarapaketid peavad olema versioneeritud. Kasutama peab Tellija versioonihalduse ja tehiste (artifaktide) repositooriumi.	Arendajale antakse selleks õigused Tellija versioonihalduse repositooriumi, kus ta peab hoidma oma erinevaid versioone. Versioonihalduse repositooriumi juurdepääsutaotlus esitatakse Tellija kasutajatoele läbi projektijuhi.	#1	V	V		Arendaja	**Arhitekt** **Administraator**
11.2	Arendaja peab veenduma, et teeb muudatusi aktuaalsesse koodi.	Hea tava on, et paralleelse arendamise puhul võetakse igal hommikul versioonihalduse repositooriumist viimane seis koodist.		V	V		Arendaja	**Arhitekt** **Administraator**
11.3	Nii arendamisel kui ka hooldusle pingute korral kasutatakse Tellija tööde ja veahalduse keskkonda.	Arendajale antakse selleks õigused Tellija tööde ja veahalduse keskkonda. Veahalduse keskkonda juurdepääsutaotlus esitatakse Tellija kasutajatoele läbi projektijuhi.		V	V		Arendaja	**Projektijuht**
11.4	Versioonihaldusesse muudatuste üleslaadimisel kasutada üleslaadimissõnumis *Conventional Commits* stiili.	https://www.conventionalcommits.org/		V			Arendaja	**Arhitekt**
12. Paigalduspaketi kooste								
12.1	Rakendus on versioneeritud kasutades semantilise versioneerimise põhimõtet.	A.B.C kujul, kus C on veaparandus, B on funktsionaalne uuendus, mis töötab ka vanematel integratsioonidel ja A on integratsioone potentsiaalselt lõhkuv uuendus. Versiooni suurt numbrit A kasutatakse ka API versiooni defineerimiseks. Lisasoovitus: Kui major versioon saab uuenduse, siis peavad vanema versiooniga teenused hakkama tagastama päises teavitust, et versioon on deprecated (nt. X-API-Deprecated). https://semver.org/	#15	V			Arendaja	**Arhitekt** **Administraator**
12.2	Juhul kui versioonihalduse keskkond ei paku paigalduspaketile kontrollsumma (checksum) automaatset koostamist, siis koostatakse kontrollsumma arendaja poolt ja pannakse eraldi .sum failina tarnele kaasa.	Räsialgoritmiks tuleb kasutada SHA256. Linuxi käsurealt kontrollkoodi koostamiseks: $ sha256sum filename [filename2] ... > kontrollkood.sum.		V	V		Arendaja	**Administraator**
12.3	Tarnitava lahenduse koosseisus üleantava lähtekoodiga peavad kaasas olema kirjeldused sellest paigalduspaketi koosteks.	Näiteks võib lahenduse paigalduspaketi koosteprotsess ette näha, et käivitada tuleb rida shell-käske või võivad lahenduse koosseisus olla valmis (ant, ..) koosteskriptid või mistahes muu moodus paigalduspaketi tekitamiseks. Eelistatud on kasutada Dockerfile ja Gitlab töövooge.		V	V		Arendaja	**Projektijuht** **Administraator**
12.4	Kooste kirjelduste alusel valmiv paigalduspakett tohib sisaldada ainult minimaalse rakenduse käitamiseks vajamineva failikomplekti.	Näiteks: kompileeritavate keelte puhul ei tohi sisaldada lähtekoodi, kui see pole vajalik rakenduse käitamiseks.		V	V		Arendaja	**Arhitekt** **Administraator**
12.5	Kooste kirjelduste alusel valmivat paigalduspaketti peab olema võimalik liigutada erinevate masinate vahel.	Näiteks ei tohi tekitada olukorda, kus rakenduse jooksutamiseks uues serveris tuleb see tingimata just sealsamas kokku kompileerida.		V	V		Arendaja	**Administraator**
12.6	Rakenduse kõik sõltuvused peavad olema kompileerimisel saadavad Tellija tehiste repositooriumist.		#4	V	V		Arendaja	**Arhitekt**
12.7	Andmebaasi paigalduse skriptid ei tohi olla kompileeritud.	Administraator tahab veenduda skripti sisus.		V	V		Arendaja	**Administraator**
12.8	Rakenduse lähtekoodi juures peab leiduma skriptid rakenduse keskkonnast sõltumatult (konteinerlahenduses) kokku kompileerimiseks.	Tellijal peab olema võimalik suuri pingutusi tegemata ja keskkonna erinevusi vältides teha rakendusest paigaldatav pakk.		V	V		Arendaja	**Arhitekt**
12.9	Rakenduse lähtekoodi juures peab leiduma skriptid rakenduse lokaalselt	Vajadusel peab konteinerlahendus käivitama ka rakenduse muud sõltuvused (näiteks andmebaas).		V	V		Arendaja	**Arhitekt**

	mõnes konteinerlahenduses (Docker) käivitamiseks.	See on Täitjale uue meeskonnaliikme liitumise lihtsustamiseks ja Tellijale võimalus suuri pingutusi tegemata süsteemi testimiseks.						
12.10	Paigalduspakett koostatakse Tellija pideva integratsiooni (continuous integration - CI) ja paigaldus (continuous deploy - CD) arendus keskkonnas.		#13	V	V		Arendaja	**Arhitekt**
12.11	Kubernetesel (K8s) orkestreeritavate lahenduste paigalduste jaoks tuleb luua *Helm chart*.	https://helm.sh/ Helmi jaoks kasutatava malli annab Tellijapoolne arhitekt.		V	V		Arendaja Arhitekt	**Arhitekt** **Administraator**
12.12	Kubernetesel (K8s) orkestreeritavate lahenduste paigalduste jaoks tuleb luua vajalikud automaatsed laienemise reeglid.	https://kubernetes.io/docs/tasks/run-application/horizontal-pod-autoscale/		V	V		Arendaja Arhitekt	**Arhitekt** **Administraator**