

E-Riigi ristfunktsionaalsed nõuded

Leht on ainult informatiivne ja arhiivi eesmärgiga, tegelikud nõuded, millest lähtuda on [Tehnilised nõuded tarkvaralahendustele](#)

NB! kõik kohustusliku märkega nõuded on SMIT-i arendustes kohustuslik järgida.

i E-riigi ristfunktsionaalsed nõuded (*cross-functional requirements*) on mittefunktsionaalsete nõuete laiem vaade. Käesolevad nõuded on aluseks igale haldusalale IT-arenduste põhinõuetele. Igale IT arendusele ja infosüsteemile rakenduvad ainult need nõuded, mida konkreetse tarkvara iseloomu, ülesehituse ja kasutatavate komponentide kontekstis on mõistlik rakendada. IT-majadele jääb vabadus kohandada ja laiendada nõudeid vastavalt enda maja spetsiifikale, võttes arvesse aga **KOHUSTUS** ja **OOTUS** märgistustega nõudeid eesmärgipäraselt.

- Nõuete dokument peab olema kasutatav nii välisel, kui sisemisel auditeerimisel näiteks RIA's, et hinnata SF või muu tarne tulemust.
- Lõpptulemus peab olema kujul, kus iga punkti saab hinnata kas **✓** või **✗**. Kui nõue teenusele ei kohaldu, siis hinnata seda **N/A**.
- Kui teenus saab **✗** nõudel, mis on **KOHUSTUS**, siis tuleb viga parandada.
- Kui teenus saab **✗** nõudel, mis on **OOTUS**, siis tuleb viga parandada, kui see on ka asutuse enda ootus.
- Nõuded, mis on märgitud kui **SOOVITUS** on puhtalt soovituslikud. Need on head täita, kuna võivad muutuda ootuseks või kohustuseks tulevikus.
- Nõudeid, mis on märgitud kui **MUSTAND** võib ignoreerida.

Lisaks on eesmärk kõik masinloetavad nõuded dokumendina plokiahelasse panna, mis võimaldaks hiljem valideerida nõudeid vastu projekti kokkuleppeid. Projekti kokkulepetes on kajastatud spetsiifiline versioon ristfunktsionaalsetest nõuetest, mis on valideeritav ja taasleitav koodivaramust. *Nõuetest peaks eksisteerima ka ingliskeelne versioon.*

Nõuded on jaotatud loogiliselt viite kategooriasse (**Arendus, Tarne, Arhitektuur, Kvaliteet, Turvalisus ja Andmehaldus**), aga need nõuded pole teineteisest iseseisvad. Näiteks mitmed turvalisuse nõuded eksisteerivad ka tarneprotsessis, sest on otsemad osad tarnest.

Arendus

nõue	alamnõuded ja soovitus	laiendatavus	SMIT kommentaar
Rakenduse kood on versioneeritud kasutades Git'i.	https://en.wikipedia.org/wiki/Git Lisaks on ülimalt tähtis, et SSL võtmeid ei talletataks koodihoidlas. Lisasoovitus: GIT kasutamiseks soovitame Git-flowd. Lisasoovitus: Andmebaasi versiooniuuenduste puhul oleks olemas ka andmebaasi migratsiooniskriptid, mis varasemaid versioone uuele versioonile tõstavad, et ei toimuks 'versiooni lukustamist'.	KOHUSTUS	Koodi halduses lähtume sellest dokumendist: L ähtekoodi halduse ja ehitamise nõuded arendajale .
Rakenduse lähtekood on kirjutatud selgusega, mis võimaldab erialast ettevalmistust omaval tarkvaraarendajal süsteemi edasi arendada.	Lisasoovitus: Rakenduse lähtekood ja kommentaarid peavad olema inglise keeles. Rakenduse ärilised muutujad aga eesti keeles, kui neile pole mõlemapoolset loogilist vastet. Lisasoovitus: Võimalusel arvestada ka Clean Code (<i>ISBN 978-0-13-235088-4</i>) printsiipe.	KOHUSTUS	Reeglina kasutada inglise keelt.
Rakenduse kood on e-riigi koodivaramus (koodivaramu.eesti.ee).	Lisasoovitus: sõltuvused on majasisesed, ehk kood koodivaramusse ainult peegeldada ja ehitamiseks kasutada kohalikku majasisest koodihoidlat.	OOTUS	Võimalusel jagada kood välja.
Rakenduste ehitamine pole väliselt sõltuv.	Tarkvara ehitamine peab olema võimalik ka siis, kui välised sõltuvused (<i>näiteks GitHub või koodivaramu</i>) on maas ja teekide ning muud sõltuvused ehitatakse rakendusse kohalikust koodihoidlast, failihoidlast või <i>artifactory</i> 'st. Lisasoovitus: Arendustes on integratsioonide puhul kasutusel <i>mock'id</i> , et võimaldada paralleelset arendust ja tuvastada arhitektuuri võimalikke vigu ja kitsaskohti. Lisasoovitus: Rakendus ei tohi eeldada operatsioonisüsteemi komponentidelt spetsiifiliste teekide olemasolu?	OOTUS	
Rakenduse arenduses ei kasutata programmeerimiskeeli, mis pole Top 25 TIOBE index'is.	https://www.tiobe.com/tiobe-index/ https://en.wikipedia.org/wiki/TIOBE_index Nõude eesmärk on minimeerida olukordi, kus arendatakse uusi komponente tehnoloogiates, millele on raske leida arendajaid, kui komponendid vajavad muutmist. Täpsustusvajadus: kas eksisteerib statistikat Eestis kasutatavate programmeerimiskeelte ja ulatuste kohta?	OOTUS	

Rakenduse tarkvara markeerida litsentsiga.	Teose autoriõigused tuleb selgelt välja tuua. Standardseks vahendiks selleks on litsents. Litsents esitatakse ühel või mõlemal alljärgnevatest viisidest: 1) LICENSE-faili repos; 2) litsentsi tekst iga faili päises. MKM ja RIA eeldavad vaba litsentsiga koodi litsentseerimist. Erandid turva- jm õigusega pandud piirangute korral. Soovitatav on kasutada MIT litsentsi - nii tagatakse paremini tarkvarade litsentsiline ühtesobivus. Alternatiiv on EUPL .	KOHUSTUS	Vajab kokkulepet
Kõikide arendamisel kasutatavate komponentide (<i>rakenduse, andmebaasi, kolmanda osapoolse omand</i>) eluea lõpp (<i>inglise k End-of-Life, EOL</i>) ei tohi teadaolevalt olla vähem kui 5 aastat.		OOTUS	
Teenuse rakendustele eksisteerivad ajakohased taasteplaanid.		OOTUS	

Tarne

nõue	alamnõuded ja soovitus	laiendatavus	SMIT kommentaar
Rakendus peab olema läbinud enne toodangusse minemist turvestimise.	Vastavalt rakenduse olemusele ja riskianalüüsile rakendada meetmed OWASP ohuetaabelites (Top 10) jm. tekstides antud soovitude järgimiseks. Link: OWASP_Top_Ten_Project	KOHUSTUS	Välisele rakendustele kindlasti partneri pool tehtud pen-testid. Siserakendustel Top10 käisiti. Ootus SECile, et saaks automaatse tööriista pipeline osaks. Testimisel kasutatakse OWASP v3.0 tase2 nõudeid.
Rakendus ei lähe toodangu-keskkonda, kui selles on avastatud turvalisuse vigu.	Vigade tuvastamine on käsitsi või automatiseeritud, näiteks koodianalüüsi tulemusel. Lisasoovitus: Rakendus ei lähe toodangu-keskkonda, kui selle koodis on avastatud koodikvaliteedi vigu.	KOHUSTUS	Kriitilised vead peavad olema parandatud.
Toodangusse ei lähe kood, mis pole läbinud koodi ülevaatamist (<i>code review</i>) minimaalselt nelja-silma printsiibi põhimõttel.	Koodi ülevaatamist saab teostada ja vigasid vähendada näiteks Atlassian BitBucket töövahendiga. Lisasoovitus: Välise partneri koodi ülevaatamine toimub majasiseste kompetentside poolt (<i>arhitekt või vanemarendaja</i>), kui see olemas.	OOTUS	
Toodangu-keskkonda ei lähe kood, mille testid ebaõnnestuvad.		KOHUSTUS	Kriitilised testid ei tohi ebaõnnestuda. Kõik testitulemused tuleb enne toodangusse panemist tuleb meeskonna poolt üle valideerida.
Keskkondadesse tarkvara paigaldamine on automatiseeritud.	Kasutades näiteks Bamboo, Jenkins või muid CI /CD vahend.	OOTUS	
Rakenduse tarne toodangu-keskkonda toimub sinine-roheline (<i>blue-green</i>) põhimõttel, ehk rakendusest sõltuva teenuse töö ei katke versiooniuuenduse puhul.	Blue-green eeldab ka mitmeid arhitektuurilisi ettevalmistusi. Link: https://martinfowler.com/bliki/BlueGreenDeployment.html	SOOVITUS	
Rakendus on versioneeritud kasutades semantilise versioneerimise põhimõtet.	https://semver.org/ A.B.C kujul, kus C on veaparandus, B on funktsionaalne uuendus, mis töötab ka vanematel integratsioonidel ja A on integratsiooni 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 <i>deprecat ed (nt. X-API-Deprecated)</i>	SOOVITUS	
Toodangu keskkondades on rakendused on automaatselt monitooritud.	See eeldab ka monitooringus tekkivate vigadest automaatset teavitamist. Lisasoovitus: Monitooringut teostada ka arendus ja test-rakenduskeskkondades. Lisasoovitus: Monitoorida ka teenuse käitumismustreid, näiteks kiirust ja arenduste mõju rakenduse kiirusele.	OOTUS	

Arhitektuur

nõue	alamnõuded ja soovitus	laiendatavus	SMIT kommentaar
Rakenduse tehnilised komponendid logivad korrelatsiooni ID'd või genereerivad selle ise. Korrelatsiooni ID'd saadetakse iga edasise päringuga kaasa.	Korrelatsiooni ID'd ei eksisteeri veel standardit. Soovitus on päise parameeter X-Correlation-ID https://en.wikipedia.org/wiki/List_of_HTTP_header_fields#Common_non-standard_request_fields	MUSTAND	
Kasutajate autentimine käib läbi RIA TARA.	https://e-gov.github.io/TARA-Doku/ Lisasoovitus: Hajusa infosüsteemi teenused saavad teiste teenuste komponentidega suhelda ilma, et peaks tekitama omaenda kasutajasessioone. Kaaluda JWT'd, OAuthi. Lisasoovitus: Seansi identifikaator peab olema piisava pikkusega, juhuslik ja unikaalne kogu aktiivse seansi jooksul.	OOTUS	Kasutame UAA-d.
Rakenduse tehnilised komponendid kaitsevad iseennast ja valideerivad nende poole pöördunud kasutaja või tehnilise teenuse õiguseid.	Kui autentimine on soovituslikult tsentraalne, siis autoriseerimine on rakenduste enda tagada.	OOTUS	
Tehniliste komponentide API'del eksisteerib automaatselt genereeritud dokumentatsioon.	Näiteks kasutades Swagger lähenemist.	OOTUS	
Rakendused on disainitud ja arendatud domeenist juhinduva disaini (<i>domain driven design</i>) ja mikroteenuste arhitektuuri põhimõtteid jälgides.	https://martinfowler.com/tags/domain%20driven%20design.html http://martinfowler.com/articles/microservices.html <i>See nõue läheb 'paksemaks', kui defineerida ära autonoomsus ja muud mikroteenuste true/false tingimused.</i> Lisasoovitus: Komponentid suhtlevad omavahel asünkroonse sõnumivahetuse teel väljaarvatud, kui komponentide vaheline suhtlus peab toimuma sünkroonselt. Lisasoovitus: Süsteemi arhitektuur peab olema ehitatud kujul, mis võimaldab nii selle tehnoloogiamigratsiooni, kui ka paindlikkust edasises arenduses. V.t. Conway Seadus https://en.wikipedia.org/wiki/Conway%27s_law Lisasoovitus: Rakendus peab olema välise süsteemide tõrgete suhtes vastupanuvõimeline (<i>resilient</i>). Välise süsteemi tõrge tohib mõjutada ainult sellest otseselt sõltuvate kasutuslugude toimimist.	OOTUS	
Infosüsteemide vaheline andmevahetus toimub üle X-tee.	https://www.ria.ee/et/riigi-infosusteeim/andmevahetuskiht-x-tee.html	OOTUS	
Kasutajaliides ja teenuse funktsionaalsus on loogiliselt eristatud kihid ja suhtlevad üle API. Ärioloogilised tehnilised komponendid on kasutatavad erinevate kasutajaliideste poolt.	Näiteks võib kaaluda Headless CMS põhimõtteid https://en.wikipedia.org/wiki/Headless_content_management_system Lisasoovitus: Kaaluda ka mikrofrontendide põhist lähenemist https://martinfowler.com/articles/micro-frontends.html	OOTUS	

<p>Tehnilised komponendid avaldavad taaskasutuseks REST API</p>	<p>Loe veel: https://en.wikipedia.org/wiki/Representational_state_transfer</p> <p>HTTP päised ja HTTP seisundi-koodid (https://en.wikipedia.org/wiki/List_of_HTTP_header_fields ja https://en.wikipedia.org/wiki/List_of_HTTP_status_codes) on osa RESTful API teenuse nõuetest.</p> <p>Lisasoovitus: REST API URL on kujul https://[teenuse-domeen-ja-keskkond]/api/[versioon]/[ressurs]/[identifikaator]/[jms...]</p> <p>Lisasoovitus: RPC lähenemine peab olema välistatud nii palju, kui (Remote Procedure Call, ehk 'sisendparameetriteta funktsioonide käivitamine') võimalik. See tähendab, et sama API päring ei tohiks üldjuhul anda erinevaid vastuseid. Ehk GET /api/v1/getMyContactInformation asemel oleks GET /api/v1/contact/23/ (kus 23 on kontakti ID).</p> <p>Lisasoovitus: Mitte unustada 'idempotentsust' https://restfulapi.net/idempotent-rest-apis/</p> <p>Lisasoovitus: Toetada HTTP protokollide vahemälu standardeid. See tähendab, et teenustes peaks olema tugi <i>Cache-Control If-Modified-Since, If-Unmodified-Since</i> jms päistele.</p> <p>Lisasoovitus: Teenus defineerib omaenda veakoodid. Veakoodid ise peavad olema kategoriseeritud HTTP standardile vastavalt.</p> <p>Lisasoovitus: Tehniliste komponentide API'del on /status päring, millega saab kontrollida teenuse tööd ja tehniline komponent saab ise läbi kontrollida ka omaenda tervise.</p> <p>Lisasoovitus: Nii veebilehtede, kui ka API'de URL'id on loetavad ja inimõistatavad</p> <p>Lisasoovitus: API teenused peavad toetama JSON andmeformaati. Accept: application/json</p> <p>Lisasoovitus: CamelCase peaks olema kasutusel API teenuste keeles nii palju, kui võimalik. Fikseeritud väärtused, nagu BOOLEAN ja ENUM tüüpi väärtused peavad olema suurte tähtedega ja snake case stiilis. Näiteks https://something.ria.ee/api/v1/person/personalCode</p>	<p>OOTUS</p>	
<p>Kasutajaliidese olekut (state) hoitakse kliendi liidese poolel ja teenused on ise olekuta.</p>	<p>Brauseri kohalik hoidlas, küpsistes vms. See on vajalik, kui teenused on mitmel õlal koormusjaoturi taga.</p>	<p>SOOVITUS</p>	
<p>Rakendus on pilvekõlbulik.</p>	<ul style="list-style-type: none"> Rakendus on skriptiga paigaldatav ning paigaldamine ja taastamine toimub automatiseeritava skripti käivitamisega. Rakendus koosneb mitmest sõltumatu instantsist. Rakendus on automaatselt skaleeritav ja vajadusel kõrgkaideldav kahe asukoha vahel. Rakenduse andmed on varundatavad ja varundatud. <p>Lisasoovitus: Rakenduse enda mälu või failisüsteemis talletatakse maksimaalselt käimasoleva päringu andmeid. Kõik andmed, mida vaja talletada pikemalt, peavad olema talletatud teenuse instantsist väljaspool.</p> <p>Lisasoovitus: Teenus suudab püsti jääda või end täisfunktsionaalselt taastada ka Chaos Monkey eksperimenteeritumise tulemusel. https://en.wikipedia.org/wiki/Chaos_engineering</p> <p>Lisasoovitus: Etalon-ideaal on 12-faktorilised rakendused https://en.wikipedia.org/wiki/Twelve-Factor_App_methodology ja https://www.nginx.com/blog/microservices-reference-architecture-nginx-twelve-factor-app/</p>	<p>OOTUS</p>	
<p>Rakenduse tehnilised hajuskomponendid ei jaga sama andmebaasi, mälu ega failisüsteemi.</p>	<p>Ehk komponendi andmed ja failid on ligipääsetavad näiteks ainult läbi kontrollitud API.</p>	<p>OOTUS</p>	
<p>Rakenduse konfiguratsioon (<i>sh. paroolid</i>) ei ole osa tarkvara koodist vaid paigaldatakse külge rakenduse tarnel keskkonda läbi keskkonnamuutujate.</p>	<p>Lisasoovitus: Ära talleta paroole tekstina konfiguratsioonides või andmebaasides. Paroolide taaskasutus ainult läbi Vault-sarnase lahenduse.</p>	<p>KOHUSTUS</p>	<p>Kui hoida konfiguratsiooni failis, siis peavad failiõigused olema minimaalsed.</p>
<p>Andmebaas ei tohi sisaldada ärioloogikat (<i>triggerid, protseduurid jms</i>).</p>	<p>Täpsustusvajadus: Seda ei saa tõenäoliselt kunagi defineerida kohustuseks. Kahjuks.</p>	<p>OOTUS</p>	
<p>Rakendus ei tohi luua uut identiteedisüsteemi. Tuleb tugineda olemasolevatele riiklikele (ID-kaart) või põhilistele op-süsteemide süsteemidele (Kerberos jms).</p>		<p>KOHUSTUS</p>	<p>Kasutame UAA-d</p>
<p>Tehnilised komponendid peavad saatma tehnilise manifesti RIHA X'i</p>	<p>täieneb</p>	<p>MUSTAND</p>	

Kvaliteet

nõue	alamnõuded ja soovitus	laiendatavus
Süsteemi funktsionaalne skoop on defineeritud ja dokumenteeritud selgesõnaliste kasutuslugudega. Nimetatud kasutuslood peavad olema toote ärilises dokumentatsioonis.	See on ülimalt tähtis, et oleks võimalik toetada automaat-testimist. Kasutuslood on hea alus ka spetsiifilise SLA defineerimiseks, monitoorimiseks ja süsteemi kvaliteedi tagamiseks. Vaata veel: ISO9000 kvaliteedisüsteem	SOOVITUS
Kasutajafunktsionaalsus peab olema kaetud automaattestidega.	See hõlmab nii kasutajaliidest, kui ka API integratsioone.	OOTUS
Rakendused on läbinud koormustestid vähemalt kahekordse eeldatava kasutajamahuga enne toodangusse minekut.	Lisasoovitus: Koormusteste teostada regulaarselt.	OOTUS
Keerulisemad sisemised tarkvaralised funktsioonid on kaetud unit-testidega.	Keeruline sisemine tarkvaraline funktsioon on funktsioon või protsess, mille hoomamine võtab rohkem, kui 30 minutit erialast ettevalmistust omavalt tarkvaraarendajalt.	OOTUS
Kasutajaliides kasutab Riigi Infosüsteemide Ameti pakutatavat e-riigi stiiliraamatut.	Leitav https://koodivaramu.eesti.ee/explore/projects	SOOVITUS
Rakendus peab töötama veebisirviijates, mis toetavad eID baastarkvara kaht viimast versiooni.	Peamiselt siis veebisirviijad <i>Chrome, Firefox, Internet Explorer, Edge</i> Lisasoovitus: Brauseri mitteühilduvusel tuleb kasutajale kuvada inimloetavat ja arusaadavat veateadet.	OOTUS
Veebipõhine kasutajaliides järgib HTML5 ja CSS3 standardeid.	https://validator.w3.org/ ning https://jigsaw.w3.org/css-validator/	OOTUS
WCAG 2.0?	https://www.w3.org/TR/WCAG20/	MUSTAND

Turvalisus

nõue	alamnõuded ja soovitus	laiendatavus	SMIT kommentaar
Tarkvara koodile rakendatakse staatilise koodi analüüsi.	Näiteks SonarCube	OOTUS	
Rakenduse tarkvara tööd logitakse ning audit-logi talletatakse rakendusest ning selle baasist eraldi.	Lisasoovitus: Logitakse aeg, kontekst (kes, kus, mida, mis tulemusega) ja kategooria (DEBUG, INFO, WARNING, ERROR, FATAL). DEBUG kategooria talletamine toodangu keskkonnas pole kohustus. Täpsustusvajadus: Logide maskimine? GDPR? Lisasoovitus: Rakenduse funktsionaalsus lukustada, kui tuvastatakse, et logitalletamine ebaõnnestub. Võimalusel tuleks arvestada logipuhvriga, et mitte iga võrguhäire puhul teenuseid maha võtta.	KOHUSTUS	Kasutame kesket logisüsteemi.
URL ei tohi sisaldada isikuandmeid või sessioonivõtiti.		OOTUS	
Kui ei ole määratud teisiti, peab rakendus olema kasutatav ISKE klassile K2T2S2 vastavate süsteemide loomisel.	Turvameetmetega tutv ISKE portaal is.	KOHUSTUS	Ei. Vastavalt vajadusele.
Kasutaja või rakenduste (sealhulgas ka enda omade) sisendit kontrollitakse ja puhastatakse rakenduse poolel vastu rakenduse enda siseseid ootuseid.	See tähendab, et misiganes sisend kasutaja või teise rakenduse poolt vajab puhastamist ja valideerimist oodatud kujule.	OOTUS	
Rakenduse väljundit (sealhulgas ka enda oma) tuleb alati puhastada ja filtreerida enne kasutamist.	Näiteks kasutajaliidestest.	OOTUS	
Krüptoalgoritmide ja räsifunktsioonide kasutamisel tuleb järgida RIA veebilehel avaldatud krüptograafiliste algoritmide elutsükli uuringu värskemais versioonis toodud soovitusi ja põhimõtteid.	Täpsustusvajadus: Link RIA blogist katki?	OOTUS	
Kasutajaliides ja tehnilised komponendid suhtlevad üle TLS /SSL'i	See eeldab ka sisevõrkude rakendusi.	KOHUSTUS	Jah. Uutel asjadel kindlasti.
Rakenduse tehnilised komponendid suhtlevad omavahel üle TLS/SSL'i		OOTUS	

Andmehaldus

nõue	alamnõuded ja soovitus	laiendatavus	SMIT kommentaar
Andmete kustutamine ei tohi kustutada tegelikke kirjeid andmebaasist, välja arvatud, kui kirjed pole mittefunktsionaalsed, ehk ei oma äriolist seost.	Kustutamine peaks olema kustutamise lipuga või juhul, kui andmed peavad päriselt kaduma, siis need obfuskeerida taastamatult. Unikaalne ID, mis erinevates teistes logides võib olla talletunud, peab aga jääma alles.	OOTUS	
Andmed peavad olema talletatud vähemalt UTF-8 enkoodingus.	UTF-16 või UTF-32 on soovitatavad, eriti kui on vajadus talletada ka kasutajate sisendina emoji'sid.	KOHUSTUS	Jah
Aja esitamisel tekstikujul lähtuda standardist ISO 8601.	<ul style="list-style-type: none"> Kuupäevad kirjutatakse kujul AAAA-KK-PP. Näide: 2. juuni 2012 kirjutada kujul 2012-06-02. Kellaajad kirjutatakse kujul hh:mm:ss, kus hh järgib 24-tunnist kellaajaformaati. Millisekundi täpsusega kellaaja teksti kujul esitamisel kasutatakse formaati hh:mm:ss,nnnn, kus nnnn on millisekundite arv. Kuupäev ja kellaeg samas andmeväljas esitatakse kujul, kus kuupäevavormingu ja kellaajavormingu vahele lisatakse täht T. Ajaintervallide kirjeldamiseks kasutatakse kahte sama täpsusega (kuupäev, kellaeg või kuupäev ja kellaeg) ajavormingut (algus ja lõpp), mis eraldatakse sümboliga /. Vt ka RFC 3339 Date and Time on the Internet: Timestamps, kus määratletakse internetiprotokollidele sobiv ISO 8601 profiil. Märkus. Mõned standardid nõuavad aja esitamist Unix epochvormingus, nt: 1) nt OpenID Connect ja 2) W3C veebileided. Märkus. Aja esitamisel inimkasutajale kasutajaliidestest lähtuda vastavas kultuuris omaksvõetud vormingutest. 	OOTUS	
Objektid identifitseerida registrikoodide abil.	Riiklikesse registritesse kantavad objektid (isikud, katastriüksused jne) kantakse andmebaasi nende registrikoodiga, mida täiendab riigiprefiks vastavalt ISO3166-1 Alpha 2 standardile. Näiteks isikute sidumiseks süsteemi kasutajakontoga peab kasutama isikukoodi rahvastikuregistrist. Eesti Vabariigi kodanik identifitseeritakse Eesti Vabariigi poolt väljastatud eIDga. Igasuguse muu identifitseerimisevahendi kasutamine peab olema selgelt põhjendatud. Mittekodanike isikuidentifikaator saadakse järgmisel viisil: riigikood + sookood + sünniaeg + [dok_nr id_riigis], kus riigikood - kolmekohaline ISO 3166-1 Alpha-3 standardile vastav riigi kood sookood - soo identifikaator nii nagu Eesti Vabariigi isikukoodis sünniaeg - sünniaeg formaadis YYYYMMDD id_riigis - kui see on olemas, tuleb kasutada isiku koduriigi isikuidentifikaatorit. 16 kohta, 0-polsterdatud vasakult dok_nr - kui isiku koduriigis isikuidentifikaatorit ei ole, siis kasutatakse isiku dokumendi numbrit. Dokumendi number, 16 kohta, 0-polsterdatud vasakult.	OOTUS	
Rakendada aadressiandmete süsteemi nõudeid.	Eesti aadressiandmete sisestamisel, kuvamisel ja hoidmisel lähtuda Vabariigi Valitsuse 8. oktoobri 2015. a määrusest nr 103 „Aadressiandmete süsteem“.	OOTUS	
Rakendada klassifikaatorite süsteemi nõudeid.	Eesti tegevusalade andmete sisestamisel, kuvamisel ja hoidmisel lähtuda Vabariigi Valitsuse 10. jaanuari 2008. a määrusest nr 11 „Klassifikaatorite süsteem“ ja kasutada EMTAK infosüsteemis kehtivat klassifikaatorit.	OOTUS	
Kui andmebaasis olevate andmete ISKE tervikluse klass on 2 või kõrgem, siis tuleb kõik klass 2 infot sisaldavad andmebaasi kirjed versioneerida.		KOHUSTUS	Ei. Vastavalt äriolisele vajadusele.
Kasutatakse Riigi Infosüsteemide Ameti andmejälgijat.	https://www.ria.ee/et/riigi-infosusteem/x-tee/andmejalgija.html	OOTUS	
Andmebaaside metaandmete kirjeldused peavad olema saadetud manifestina "RIHA"sse	täieneb seoses RIHA X arendustega	MUSTAND	