

# Tehnilised nõuded tarkvaralahendustele

Need nõuded määratlevad millele peaksid loodavad tarkvara komponendid vastama, et nad ei oleks infrastruktuurist jäigalt sõltuvates, töötaksid SMIT majutusplatvormil, nende skaleerimist ja paigaldamist saaks automatiseerida ning tehnoloogiline võlg oleks hallatav ja mõõdetav.

## Arhitektuurinõuded ja infrastruktuuriotsused

### 1. Üldine

Nr	Nõue	Kohustuslik	Märkused ja lingid
1.1	Rakendused luuakse "bounded context" põhimõttel - ühte komponenti ei panda sisult erinevat funktsionaalsust	<b>KOHUSTUS</b>	Komponentideks jaotamine toimub sisulistest, mitte tehnilistest funktsionaalsustest lähtudes v.a kasutajaliideste eraldamine. Vältida hajust monoliiti. Vt: <a href="#">Microservices</a> , <a href="#">Bounded Context</a> , <a href="#">Microservices Pitfalls</a>
1.2	Komponent peab jooksmas vähemalt 2 instantsi peal platvormi katkestuste vältimiseks	<b>KOHUSTUS</b>	Konteinerplatvormile paigaldatud rakenduste puhul minimaalselt 2 instantsi toodangus. Eelistada komponentide hulga tõstmist, mitte ressursikasutuse tõstmist
1.3	Ehitamine SMIT CI/CD-ga (Bamboo), staatilise koodi analüüs SonarQube-ga	<b>KOHUSTUS</b>	Lähtekood ei tohi sisaldada analüsaatori poolt leitud vigu. Minimaalne vaikeprofiil. Vt: "Tehnoloogilise võla mõõtmine (SonarQube)" dok
1.4	Rakenduse lähtekood kirjutatud selgusega, mis võimaldab erialast ettevalmistust omaval tarkvaraarendajal süsteemi edasi arendada	<b>KOHUSTUS</b>	Rakenduse lähtekood ja kommentaarid peavad olema inglise keeles. Rakenduse ärilised muutujad aga eesti keeles, kui neile pole mõlemapoolset loogilist vastet
1.5	Rakenduse lähtekoodi haldus toimib SMIT kesksete versioonihalduspõhimõtete järgi	<b>KOHUSTUS</b>	Kehtib nii loogika kui kasutajaliidese komponendi puhul. Lisainfo leiab dokumendist: "Lähtekoodi halduse ja ehitamise nõuded"
1.6	Keskkondadesse rakenduste paigaldamine on automatiseeritud SMIT CI/CD mehhanismidega	<b>KOHUSTUS</b>	Kasutatakse Bamboo paigaldusplaane: <a href="#">Bamboo Deployment Projects</a>

Nr	Nõue	Kohustuslik	Märkused ja lingid
1.7	Rakendustel puudub ligipääs avalikku internetti (sh klientidel)	<b>KOHUSTUS</b>	Kui on vajalik ligipääs äriandmetele väljaspool SMIT võrku, tuleb tellida "webproxy" ligipääs
1.8	Kõik välised sõltuvused tuleb rakenduse ehitamisprotsessi käigus läbi skaneerida Artifactory Xray töövahendiga	<b>KOHUSTUS</b>	Mitte kasutada kõrge kriitilisusega turvanõrkuseid sisaldavaid teeke/komponente. <a href="#">JFrog CLI for Xray</a>
1.9	Rakenduse arhitektuuri koostamisel tuleks arvestada, et komponente peab saama uuendada iseseisvalt	Soovituslik	Soovituslik kasutada "Blue-Green" paigalduspõhimõtteid
1.10	Rakendustevahelisi integratsioone teostatakse kokkulepitud põhimõtete järgi	<b>KOHUSTUS</b>	Võimalikud variandid: XTEE, HTTP otse või MQ. Integratsioonide disainis lähtuda dokumendist: "Rakendustevahelised integratsioonipõhimõtted"
1.11	API spetsifikatsioonid peavad olema kättesaadavad kolmandatele osapooltele	<b>KOHUSTUS</b>	OpenAPI/Swagger spetsifikatsioon peab olema eraldi ligipääsetav välispartneritele integratsioonide jaoks. Versiooninumber võib olla kas päises või URI-s. Teenuse tehnilises dokumentatsioonis peab olema viide OpenAPI spetsifikatsioonile (SwaggerUI). API disainimisel tuleb lähtuda dokumendist: "Rakendustevahelised integratsioonipõhimõtted"
1.12	Kui loodav lahendus koosneb mitmest komponendist, on need lahus arendatavad, pakendatavad, versioneeritavad ja paigaldatavad	<b>KOHUSTUS</b>	Võimaldab erinevate komponentide iseseisvat arendamist ja paigaldamist
1.13	Rakendused automaatselt monitooritud toodangus	<b>KOHUSTUS</b>	Toodangu keskkondades on automaatne monitooring kohustuslik
1.14	Kasuta SMIT raamistike põhjasid uute komponentide jaoks	<b>KOHUSTUS</b>	Erandid nõuavad arhitektuurinõukogu heakskiitu. Kasuta SMIT Examples starterkit malle. Kui sobilikku põhja ei eksisteeri, tuleb selle minimaalne versioon projekti käigus tekitada

Nr	Nõue	Kohustuslik	Märkused ja lingid
1.15	Rakendus on versioneeritud kasutades "semantic versioning" põhimõtet	KOHUSTUS	A.B.C kujul, kus C on veaparandus, B on funktsionaalne uuendus mis töötab vanematel integratsioonidel ja A on integratsioone potentsiaalselt lõhkuv uuendus. Versiooni suurt numbrit A kasutatakse ka API versiooni defineerimiseks. Vt: <a href="#">Semantic Versioning</a> . Soovitus: major versiooni uuendamisel tagastada vanema versiooniga teenused päises X-API-Deprecated
1.16	Ehitamise, reliisimise ja paigaldamise etapid on rangelt eraldatud	KOHUSTUS	Ehitamine toodab muutumatu artefakti, reliis laeb artifakti ülesse, paigaldamine kombineerib artifakti konfiguratsiooniga ja käivitab keskkonnas. Iga etapp peab olema iseseisev ja korratav. Vt: <a href="#">"12-Factor Build Release Run"</a>
1.17	Komponendid peavad käivituma kiiresti ja sulguma korrektselt (graceful shutdown)	KOHUSTUS	Käivitusaeg maksimaalselt 60 sekundit, korrektne sulgumine SIGTERM signaali peale. Komponendid peavad olema "disposable" - saab igal hetkel peatada ja käivitada. Toetab kiiremat skaleerimist ja paigaldamist
1.18	Üks lähtekood repositoorium komponendi kohta - kõik keskkonnad paigaldatakse samast koodist	KOHUSTUS	Iga komponent peab omama ühte Git repositooriumit. Erinevad keskkonnad (arendus, test, toodang) kasutatavad sama lähtekoodi, kuid erinevat konfiguratsiooni. Git "branching" strateegia peab toetama sellist lähenemist
1.19	Kõik sõltuvused tuleb deklareerida spetsfiiliselt ja isoleerida	KOHUSTUS	Komponendid ei tohi sõltuda süsteemi-laadsetes teekidest. Kasutada "dependency manifest" faile ja lukustatud versioone reprodutseeritavate ehituste jaoks. Isoleerida sõltuvused konteinerite või virtuaalsete keskkondadega

Nr	Nõue	Kohustuslik	Märkused ja lingid
1.20	Administratiivsed ülesanded käivitatakse eraldi protsessidena	<b>KOHUSTUS</b>	Andmebaasi migratsioonid, ühekordsed skriptid ja muud administratiivsed toimingud peavad töötama eraldi komponent põhiprotsessidest. Admin ülesanded kasutavad sama koodi baasi, kuid käivitatakse eraldi
1.21	Komponendid peavad olema iseseisavad ja eksportima teenuseid pordi sidumise kaudu	<b>KOHUSTUS</b>	Komponendid ei tohi sõltuda välisest runtime serverist (Apache, Nginx). Iga komponent peab sisaldama oma HTTP serverit ja siduma end konfigureeritava pordi külge. Toetab mitme instantsi käivitamist samal masinal
1.22	Kasutamata kood ja välja kommenteeritud kood tuleb koodibaasist eemaldada	<b>KOHUSTUS</b>	Hoia kood puhas: eemalda "dead code" ja välja kommenteeritud plokid; koodi taastamiseks kasuta Git ajalugu, mitte kommenteeritud fragmente. Vastavus SonarQube "commented-out code" ja kasutamata koodi reeglitele

## 2. Testimine

Nr	Nõue	Kohustuslik	Märkused ja lingid
2.1	Minimaalne testandmete genereerimise võimekus	Soovituslik	Peamiselt koormustestide või "smoke" testide keskkondade jaoks
2.2	Test- ja toodangukeskkonnad eraldatud	<b>KOHUSTUS</b>	Testandmed ei tohi olla toodanguandmed (välja arvatud avalikud andmed nagu aadressid)
2.3	Kasutajaliidese automaattestide (E2E) kirjutamine on arendusmeeskonna otsustada	Soovituslik	Testide loomisel on soovitatav kasutada ilma kasutajaliideta sirvijat ("headless" brauserid)
2.4	Rakenduse automaattestide mahtu tuleb analüüsida CI/CD mehhanismidega ja publitseerida SMIT SonarQube keskkonnas	<b>KOHUSTUS</b>	Testide mahu miinimum määratakse SonarQube kvaliteedivärvaga (reeglina 80% kogu koodist)

Nr	Nõue	Kohustuslik	Märkused ja lingid
2.5	Testikattuvus peab olema vähemalt 80% kogu koodist	<b>KOHUSTUS</b>	Testikattuvuse miinimum jõustatakse SonarQube kvaliteedivärvaga. Erandid või madalama läve kasutamine tuleb arhitektuurinõukogus kooskõlastada. Staatilised HTML ja stiilifailid (CSS/SCSS/SASS) võib testikattuvuse arvutusest välja jätta, kuna nende eraldi testimine on raskendatud; soovitatav on kasutada SonarQube <code>sonar.coverage.exclusions</code> sätteid
2.6	Turvatestid avalike teenuste jaoks	<b>KOHUSTUS</b>	Turvatestide juhendid ja nõuded leiab dokumendist: "Turvatestimine"
2.7	Rakenduste poolt toodetavale logi väljundile tuleb kirjutada automaattestid	<b>KOHUSTUS</b>	Testid peavad veenduma, et rakenduses genereeritud logi vastab "Logimise nõuded" dokumendis kirjeldatud nõuetele

### 3. Kasutajaliides

Nr	Nõue	Kohustuslik	Märkused ja lingid
3.1	Kasutajaliides võib korraga suhelda mitme loogikakomponendiga	Soovituslik	Võimaldab paindlikku arhitektuuri disaini
3.2	Kasutajaliides on soovitatav arendada õhukese kliendina	Soovituslik	Thin client arhitektuuri eelistamine
3.3	Avalikud kasutajaliidesed peavad järgima VEERA disainistandardit ning omama WCAG tuge	Soovituslik	Vajadusel kohaldatav avalike teenuste puhul

### 4. Infrastruktuur ja paigaldamine

Nr	Nõue	Kohustuslik	Märkused ja lingid
4.1	Rakendused peavad jooksmas vähemalt 2 instantsi peal	<b>KOHUSTUS</b>	Platvormi muudatuste käigus katkestuste vältimine. Konteinerplatvormil minimaalselt 2 instantsi toodangus

Nr	Nõue	Kohustuslik	Märkused ja lingid
4.2	Komponendid peavad olema iseseisvalt arendatavad, pakendatavad, versioneeritavad ja paigaldatavad	<b>KOHUSTUS</b>	Võimaldab sõltumatut arendust ja paigaldamist
4.3	Komponendid suhtlevad omavahel ainult üle HTTP/HTTPS või JMS/AMQP protokoll	<b>KOHUSTUS</b>	Soovitus kasutada protokolle üle turvalise kanali (HTTPS/AMQPS)
4.4	Iga komponent peab omama oma andmebaasi, kui vajab andmete salvestamist	<b>KOHUSTUS</b>	Andmebaasidevahelised integratsioonid on keelatud. Mitu komponenti ei tohi kasutada sama andmebaasi
4.5	Kasutada eraldi failide hoidmise teenust binaarfailide jaoks	<b>KOHUSTUS</b>	SMIT-is on selleks olemas sisemine S3 objektihoidla
4.6	Toetada horisontaalset skaleerimist	<b>KOHUSTUS</b>	Suurendada instantside arvu, mitte ressursse instantsi kohta
4.7	Komponent käivitub ilma ühendusega liidestetavate süsteemidega	<b>KOHUSTUS</b>	Nõrgalt liidestatud, välja arvatud andmebaasid
4.8	Komponent töötab osaliselt ka liidestuste katkestuste puhul	<b>KOHUSTUS</b>	Kasutada Circuit Breaker või retry mehhanisme. Healthcheck peab andma teada probleemist
4.9	Komponentide käivitusaeg peab olema alla 60 sekundi	<b>KOHUSTUS</b>	Kiire käivitamine skaleerimise ja taaskäivitamise jaoks
4.10	Kasutada SSL kõigile UI-backend suhtlustele	<b>KOHUSTUS</b>	SSL termineeritakse koormusjaoturis
4.11	Kasutada distribueeritud vahemälulahendusi	<b>KOHUSTUS</b>	Vältida vahemälu replikatsiooni, kasutada nt. Redis või andmebaasi
4.12	Konfiguratsioon keskkonna muutujatena või platvormi võimalustega	<b>KOHUSTUS</b>	Puudub eraldi väline konfiguratsioonifail. Kasutada Kubernetes secrets/configmap. Rakenduse saladused, mida ei genereerita teenuste poolt automaatselt, panna SMIT saladuste hoidlasse. Saladused liiguvad rakendusse ainult läbi CI/CD

Nr	Nõue	Kohustuslik	Märkused ja lingid
4.13	Komponendi pakendamine Docker konteinerina SMIT baasimageidega	Soovituslik	Soovituslik pakendamisformaad on Docker konteiner. Kasutada ainult SMIT dockerhubis asuvaid baasimageid: SMIT DockerHub
4.14	Failid tuleb skaneerida enne salvestamist	<b>KOHUSTUS</b>	Kasutada SMIT MetaVault teenust

## 5. Sõltuvuste haldamine

Nr	Nõue	Kohustuslik	Märkused ja lingid
5.1	Kõik sõltuvused tuleb lahendada SMIT sisesest repositooriumist	<b>KOHUSTUS</b>	SMIT Artifactory. Ei tohi kasutada väliseid repositooriume (Maven Central, NPM, PyPI) otse
5.2	Konfigureerida ehitusvahendid kasutama SMIT repositooriumi	<b>KOHUSTUS</b>	Kõik välised sõltuvused peavad olema Artifactory kaudu kättesaadavad

## 6. Äriloogika

Nr	Nõue	Kohustuslik	Märkused ja lingid
6.1	Komponentide vaheline andmevahetus peab olema turvaline TLS-iga, sertifikaate verifitseeritakse	Soovituslik	Autentimata ja/või krüpteerimata protokollide kasutamisel rakendatakse täiendavaid konfidentsiaalsust ja terviklust tagavaid turvameetmeid
6.2	Kasuta SMIT keskset tuvastamise teenust (UAA)	<b>KOHUSTUS</b>	UAA integreeritud RIA Taraga ID-kaart, mobiil-ID, Smart-ID jaoks. <a href="#">UAA API Docs</a> . Väliste kasutajate puhul kasuta e-posti põhiseid OTP-d
6.3	Kasutajate ja nende gruppide/rollipõhiste õiguste tuvastamine keskest Active Directory-st	<b>KOHUSTUS</b>	Vähemalt üks AD roll peab asuma AD-s (ja vastav UAA skoop). AD gruppidest võetakse info ja lisatakse UAA tokeni skoopi. Andmetepõhised õigused (ACL) asuvad rakenduse andmebaasis
6.4	Iga komponent peab omama oma kontot	<b>KOHUSTUS</b>	Ei tohi taaskasutada kontosid, mis on väljastatud teistele rakendustele

Nr	Nõue	Kohustuslik	Märkused ja lingid
6.5	Komponendid peavad saama omavahel autoriseerituna andmeid vahetada kasutades tokeneid	<b>KOHUSTUS</b>	Kahe komponendi vaheliseks andmevahetuseks, mis ei toimu lõppkasutaja kontekstis, tuleb kasutada SVC tüüpi süsteemikontosid ADs lisatud vajalike skoopidega
6.6	Tuvastatud kasutajal peab eksisteerima serveripoolne sessioon keskses hoidlas	<b>KOHUSTUS</b>	Kasutada ainult OAuth2 Authorization Code mehhanismi. Keskse seansi hoidlana Redis või PostgreSQL. Klient peab saama tokeniga seotud info kustutada. Mitmest komponendist koosneva rakenduse puhul kasutada API gateway sessioonihaldajana
6.7	Komponendi versiooniuuendused tehakse ilma katkestusteta teenuse töös	Soovituslik	Soovitav "Blue-Green deployment" mehhanism. <a href="#">Blue-Green Deployment</a> . Andmebaasi muudatuste tegemisel tagada töötamine ka eelmiste versioonidega
6.8	Töövoogude koordineerimisel kasutada protsessimootorit	Soovituslik	Tagab protsesside parema läbipaistvuse ja modelleerimisvõimekuse. Näiteks <a href="#">Camunda</a> või <a href="#">Flowable</a>
6.9	Kasutada välist ACL süsteemi õiguste haldamiseks	Soovituslik	Keskne õiguste haldus ja auditeeritavus. Näiteks <a href="#">Permify</a> või sarnased lahendused

## 7. Andmebaas

Nr	Nõue	Kohustuslik	Märkused ja lingid
7.1	Eraldi operatiiv- ja arhiiv andmebaasid	<b>KOHUSTUS</b>	Kasuta eraldi arhiivandmebaase või spetsialiseeritud archiveerimislahendusi
7.2	Andmebaaside vahelised integratsioonid ei ole lubatud	<b>KOHUSTUS</b>	Erinevad andmebaasid ei tohi omavahel integreeruda
7.3	Andmebaase komponentide integratsioonivahendina ei tohi kasutada	<b>KOHUSTUS</b>	Mitu erinevat komponenti ühe andmebaasi poole pöörduda ei ole lubatud. See ei tähenda, et sama komponenti skaleerides ei võiks kasutada sama andmebaasi



Nr	Nõue	Kohustuslik	Märkused ja lingid
7.4	Migratsiooniskriptide käivitamine paigaldusprotsessi osana	<b>KOHUSTUS</b>	Migratsioone on soovitatav käivitada paigaldusprotsessi ühe osana, mitte iga rakenduse restardiga. Kõige lihtsam variant on määrata CI/CD keskkonna muutujaga. Mõistlik on migratsiooni teha ühe instantsiga, siis vajadusel uuesti paigaldus mitme instantsiga
7.5	Andmebaasi ärioloogikat vaikimisi ei kirjutata (protseduurid ja triggerid)	<b>KOHUSTUS</b>	Kõrvalekalded tuleb defineerida ja põhjendada konkreetse komponendi arhitektuuridokumendis

## 8. Jõudlus

Nr	Nõue	Kohustuslik	Märkused ja lingid
8.1	Sisemised komponendid kasutavad asünkroonset suhtlust	Soovituslik	Blokeeruvate lõimede vältimine ja efektiivne protsessorituumade kasutamine. Erand: kasutaja transaktsioonisisene suhtlus. Vt: <a href="#">12 Factor Concurrency</a> , <a href="#">Reactive Manifesto</a>
8.2	Jõudluse kasvamisel horisontaalne skaleerimine (instantside hulga tõstmine)	<b>KOHUSTUS</b>	Ei tohiks kasutada "scale-up" lahendust, kui seda ei määra konkreetne toote pakkuja
8.3	Mitte-operatiivsed lugemispäringud suunata võimalusel andmebaasi replikatele	Soovituslik	Vähendab peaandmebaasi koormust ja parandab jõudlust. Rakendada read-only operatsioonide jaoks

## 9. Monitooring ja jälgimine

Nr	Nõue	Kohustuslik	Märkused ja lingid
9.1	Komponentide monitooringu info kättesaadavus kokkulepitud formaadis ja protokolliga	<b>KOHUSTUS</b>	Soovitatavalt REST formaadis. Koormusjaotur peab aru saama, kas suunata liiklus konkreetsele nodele. Näide: Spring Boot Actuator endpoint /health. <a href="#">Spring Boot Health</a>
9.2	Rakendused automaatselt monitooritud toodangus APM tarkvara agentidega	<b>KOHUSTUS</b>	Monitooringu seadistamine ja haldamine. Toetatud tehnoloogiad: <a href="#">AppDynamics</a> <a href="#">Supported Environments</a>

## 10. Failid

Nr	Nõue	Kohustuslik	Märkused ja lingid
10.1	Failide skaneerimine enne salvestamist andmebaasi või objektihooldlasse	<b>KOHUSTUS</b>	Kasutada tuleb SMIT-i poolt pakutavat MetaVault teenust

## 11. Taaskasutus

Nr	Nõue	Kohustuslik	Märkused ja lingid
11.1	Kontrollida enne uue komponendi arendamist SMIT-is olemasolevaid lahendusi	<b>KOHUSTUS</b>	Eelistada olemasolevaid lahendusi teenusena või paigaldatava tootena
11.2	Olemasoleva teenuse koodi eraldamine eraldi projektiks vajab arhitektuuripaneeli heakskiitu	<b>KOHUSTUS</b>	Eelnev diskussioon olemasoleva tiimiga soovitatav
11.3	Teenuste puhul väljaspool konkreetset infosüsteemi luua iseseisev tehnilise liidestuse võimekus	Soovituslik	Nt Kubernetes Operator: <a href="https://operatorframework.io">operatorframework.io</a>

## 12. Arvutitöökoht

Nr	Nõue	Kohustuslik	Märkused ja lingid
12.1	Uued seadmed peavad olema Windows või iOS põhised	<b>KOHUSTUS</b>	Toetatud seadmetüübid organisatsiooni IT-poliitika kohaselt
12.2	Rakendus võimeline väikeste versioonide iseuuendamiseks	<b>KOHUSTUS</b>	Võimaldab automaatseid väiksemaid uuendusi ilma IT-toe sekkumiseta
12.3	Paigaldus ei tohi keelata arvuti teenuseid	<b>KOHUSTUS</b>	Näide: DocCheck keelas ID kaardi sertifikaaditeenuse
12.4	Toetada kõiki heakskiidetud brausereid, mitte seotud konkreetse versiooniga	<b>KOHUSTUS</b>	Kõik brauserid tehnoloogiavaliku lehelt peavad olema toetatud
12.5	Tarkvara ja konfiguratsiooni tarne ATKH-le töökoha paigaldamiseks	<b>KOHUSTUS</b>	Arendus peab andma tarkvara + konfiguratsiooni ATKH-le
12.6	Domeeni valideerimine standard/erilises ATK profiilis	<b>KOHUSTUS</b>	Domeen peab valideeruma standard ATK profiilis või SMIT erilises profiilis

Nr	Nõue	Kohustuslik	Märkused ja lingid
12.7	Rakendus peab olema käsurealt paigaldatav	<b>KOHUSTUS</b>	Võimaldab automatiseeritud paigaldamist ja masshaldust
12.8	Rakendus ei tohi nõuda administraatoriõigusi kasutamiseks	<b>KOHUSTUS</b>	Tavakasutaja peab saama rakendust kasutada ilma kõrgendatud õigusteta
12.9	Veebirakendused peavad toetama Chrome'i ja Firefox'i sisevõrgus	<b>KOHUSTUS</b>	Minimaalne brauserite tugi organisatsiooni sisevõrgus