

Tehnilised nõuded tarkvaralahendustele

Käesolevad nõuded/põhimõtted spetsifitseerivad millele peaksid loodavad tarkvara komponendid vastama, et nad ei oleks infrastruktuurist järgalt sõltuvuses, töötaksid SMIT majutusplatvormil, nende skaleerimist ja paigaldamist saaks automatiseerida ning tehnoloogiline võlg on hallatav ja mõõdetav. Dokument on abimaterjal arendajatele ja arhitektidele kes konkreetset tarkvaralahendust disainivad.

Kategooria	Nõue	Laiendatavus	Kommentaar
Üldine			
	Rakendused luuakse põhimõttel, kus ühte paigaldatavasse komponenti ei panda kokku sisult väga erinevat funktsionaalsust, vaid pigem jaotatakse vajadusel erinevate komponentide vahel. Komponentideks jaotamine toimub sisulistest, mitte tehnilistest funktsionaalsustest lähtudes v.a kasutajaliideste eraldamine eraldi rakendusteks. (vt. Bounded Context)	KOHUSTUS	Komponentideks jaotamisel mitte lähtuda tehnilistest mõõdikuteist nagu koodiridade arv või et tõstame näiteks tehnilise funktsionaalsuse nagu logimise eraldi komponendiks. Kui üks komponent on jäigas sõltuvuses teisest (ei saa tööprotsessi lõpuni viia või ei saa paigaldada, kui teine ei vasta samal ajal), siis on tegemist hajusa monoliidiga ja tuleks need komponendid kokku tõsta koodi tasemel. Täiendavat lugemist: http://martinfowler.com/articles/microservices.html https://martinfowler.com/bliki/BoundedContext.html https://microservices.io/microservices/general/2018/11/04/potholes-in-road-from-monolithic-hell.html
	Komponent peab jooksma vähemalt 2 instantsi peal, et vältida platvormi muudatuste käigus tekkida võivaid katkestusi.	KOHUSTUS	Konteinerplatvormile paigaldatud rakenduste puhul peaks vähemalt toodangu keskkonnas minimaalselt 2 instantsi jooksma, et vältida katkestusi. Eelistada ka jõudluse vaates komponentide hulga tõstmist, mitte tõsta olemasolevate komponente ressursikasutust. Näiteks mitte teha VM-e suuremaks (mälu/CPU vaates).
	Rakendust ehitatakse ja pakendatakse SMIT CI/CD lahendusega (Bamboo) ning komponendile kohandatakse automaatset staatilise koodi analüüsi vastavalt SonarQube profiilile, mille tulemused on kättesaadavad SMIT SonarQube keskkonnast. Lähtekood ei tohi sisaldada vigu mis on analüsaatori poolt leitud.	KOHUSTUS	Kehtib nii loogika kui kasutajaliidese komponendi puhul. Minimaalne on vaikeprofiil, kui arendajad ei lepi kokku täiendavates nõuetes. Täiendav info: Tehnoloogilise võla mõõtmine (SonarQube)
	Lähtekood on UTF-8 formaadis ning tekstilised väärtused tuleb liigutada tõlkefailidesse (i8n), lisaks mitte kasutada Deprecated meetodeid.	KOHUSTUS	
	Rakenduse lähtekoodi haldus toimib SMITI kesksete versioonihalduspõhimõtete järgi.	KOHUSTUS	Kehtib nii loogika kui kasutajaliidese komponendi puhul. Lisainfo leiab dokumentid: " Lähtekoodi halduse ja ehitamise nõuded arendajale "
	Keskcondadesse rakenduste paigaldamine on automatiseeritud SMIT CI /CD mehhanismidega.	KOHUSTUS	Kasutatakse Bamboo paigaldusplaane: https://confluence.atlassian.com/bamboo/deployment-projects-338363438.html
	Kokku ehitatud rakendust peab saama paigaldada erinevatesse keskkondadesse, ilma et seda peaks uuesti kokku ehitama.	KOHUSTUS	Konfiguratsioon määratakse keskkonna muutujatena või muude rakenduseväliste võimalustega (näiteks Kubernetes configmap)
	Auditlogi tuleb markeerida ära logikirjetes, et oleks pärast otsingus eristatav	KOHUSTUS	
	Rakendustel puudub ligipääs avalikku internetti (sh klientidel). Kui on vajalik ligipääs äriandmetele väljaspool SMIT võrku, tuleb tellida webproxy ligipääs.	KOHUSTUS	
	Rakenduste sõltuvuste (maven,npm jne) allikana tuleb kasutada SMIT sisest kesksel repositooriumi (Artifactory) .	KOHUSTUS	Kõik välised sõltuvused peavad olema seal kättesaadavad.
	Kõik välised sõltuvused tuleb rakenduse ehitamisprotsessikäigus läbi skaneerida Artifactory Xray töövahendiga ja mitte kasutada kõrge kriitilisusega turvanõrkuseid sisaldavaid teeke/komponente.		https://jfrog.com/help/r/jfrog-cli-for-jfrog-xray
	Kasutajaliidese sõltuvused/osad nagu JS,CSS, FONTS,GIF jms, mida kasutatakse kasutajaliidese kuvamisel, peavad rakenduse käivitusel tulema samast allikast.	KOHUSTUS	Välise sõltuvuste ja repode kasutamine ei ole lubatud (nagu välised cdn-id, google fonts, github jne). Arendamisel tuleb kontrollida browseri võrguliikluse lehel, et välisperimeetrisse ei mindaks päringuid tegema.
	Väliste teenuste nagu ReCaptcha, Google Analytics jms kasutamine ei ole lubatud.	KOHUSTUS	Tuleb kasutada SMIT siseseid lahendusi.

	Rakenduse arhitektuuri koostamisel tuleks arvestada, et rakenduse koosseisu kuuluvaid komponente peab saama uuendada iseseisvalt.		Soovituslik kasutada "Blue-Green" paigalduspõhimõtteid.
	Rakenduse loogikakomponentide üldine sisemine arhitektuur peaks järgima MVC mustrit.		
	Rakendustevahelisi integratsioone teostatakse kokkulepitud põhimõtete järgi (võimalikud variandid on XTEE, HTTP otse või MQ).	KOHUSTUS	Kasutatavatest protokollidest olenemata teostatakse integratsioone, lähtudes nõuetedokumentidest: " Rakendustevahelised+integratsioonimustrid "
	Kui loodav lahendus koosneb mitmest komponendist, on need lahus arendatavad, pakendatavad, versioneeritavad ja paigaldatavad.	KOHUSTUS	
Testitavus			
	Komponendil on olemas minimaalne genereeritav testandmete komplekt.		Peamiselt koormustestide või suitsutestide läbiviimise keskkonna jaoks.
	Test- ja toodangukeskkonnad peavad olema üksteisest lahus ning testandmed ei tohi olla toodanguandmed (v.a juhul kui tegemist ei ole avalike andmetega - näiteks aadressid).		
	Loogikakomponendil on olemas integratsioonitestid veebiteenuste või sõnumivahetusteenuste jaoks ning vajadusel täiendavad ühiktestid spetsiifilise äriloogika valideerimiseks.	KOHUSTUS	Ühiktestide maht ja olemasolu on vabatahtlik.
	Kasutajaliidese automaatsete kirjutamine, mis simuleerivad kasutaja käitumist põhivõrgude taseme, on arendusmeeskonna otsustada.		Testide loomisel on soovitatav kasutada ilma kasutajaliidestast sirvijat, et oleks võimalik kergesti testida ilma graafilist keskkonda omamata.
	Rakenduse automaatsete mahtu tuleb analüüsida CI/CD mehhanismide abil ning tulemused peab publitseerima SMIT SonarQube keskkonnas.	KOHUSTUS	
	Avaklikke teenuseid tuleb täiendavalt turvestada.	KOHUSTUS	Turvestuste juhendid ja nõuded leiab dokumentidest: " Turvestamine "
	Rakenduste poolt toodetavale logi väljundile tuleb kirjutada automaattestid	KOHUSTUS	Testid peavad veenduma, et rakenduses genereeritud logi vastab " Logimise nõuded " dokumentis kirjeldatud nõuetele
Kasutajaliides			
	Kasutajaliides tuleb täielikult eraldi arendada loogikakomponendist ning ei eelda selle implementatsiooni olemasolu.	KOHUSTUS	
	Kasutajaliides võib korruga suhelda mitme loogikakomponendiga.		
	Kasutajaliides ja loogikakomponent suhtlevad omavahel üle HTTP /WEBSOCKET protokollide ning ainult läbi eeldefineeritud liidestuse (vahetatakse ainult andmeid).	KOHUSTUS	Võimalikud integratsioonitehnoloogiad: REST, GraphQL, STOMP.
	Kasutajaliidese olekut hoitakse kliendi poolel.		Kas siis veebilehitsejas (sessionstorage) või sessioonis mis asub kasutajaliidest pakkuvas serverikomponendis.
	Kasutajaliides vahetab loogikakomponendiga ainult andmeid, visuaalset sisu (nn. html-i javascripti) sellest komponendist ei laeta.	KOHUSTUS	Vaikimisi on andmete formaat JSON
	Kasutajaliides suhtleb loogikakomponentidega üle SSL kanali (SSL termineeritakse koormusjaoturis, kust kõik kasutaja päringud läbi lastakse).	KOHUSTUS	
	Kasutajaliides on soovitatav arendada õhukese kliendina.		
Äriloogika ja õigused			
	Komponentide vaheline andmevahetus peab olema turvaline või kaitstud kasutades TLS-i, mille sertifikaate verifitseeritakse. Autentimata ja/või krüpteerimata protokollide kasutamisel rakendatakse täiendavaid konfidentsiaalsust ja terviklust tagavaid turvameetmeid.		
	Nii sisemised kui välimised süsteemid peavad kasutaja tuvastamiseks kasutama SMIT keskest tuvastamise teenust (UAA).	KOHUSTUS	UAA on tuvastamislahendus, mis on integreeritud lisaks sisemisele parooliga tuvastamisele ka RIA Taraga, et saaks kasutada ID-kaart, Mobiil-ID või SmartID tuvastamisvõimalusi. Rakendustega integratsiooniks kasutame Open-ID protokollid. UAA liidestamise dokumentatsioon: https://docs.cloudfoundry.org/api/uaa/version/76.13.0/index.html#overview
	Kasutajaid ja nende grupi või rollipõhiseid õiguseid tuvastatakse keskest Active Directory andmebaasist. Andmetepõhised õigused (ACL) asuvad rakenduse juures andmebaasis.	KOHUSTUS	Vähemalt üks roll peab asuma AD-s (ja selle küljes vastav UAA skoop), mis määratleb kas rakendusele saab ligi. Kehtib kõigi uute arenduste puhul. AD gruppidest võetakse info ja lisatakse see UAA tokeni skoopi. Eeldus on et AD gruppide külge lisatakse skoopi info ning rakendusel ei ole vaja minna AD-sse eraldi grupinfot küsima.

Komponendid suhtlevad omavahel ainult üle HTTP või JMS/AMPQ protokollide. Suurema jõudluse saavutamiseks on eelistatud sõnumivahetus.	KOHUSTUS	Soovitus on kasutada protokolle üle turvalise kanali (HTTPS/AMQP).
Komponent on sessioonivaba - kasutaja andmete hoidmiseks ei kasutata HTTP sessiooni, vaid iga päringuga valideeritakse kliendist "tokeni" kehtivust. Tokenile vastav kasutaja info peab asuma keskses tokeni hoidlas, kus on märgitud ka aegumistähtaeg.	KOHUSTUS	Tokenite haldamise mehhanismina tuleb kasutada SMIT keskest tuvastamise teenust.
Komponendid peavad ka omavahel saama autoriseerituna andmeid vahetada analoogselt kasutajaliidesele ilma sessioone tekitamata.	KOHUSTUS	
Komponentide omavahelises integratsioonis peab iga komponent oma oma kontot, ei tohi taaskasutada kontosid, mis on väljastatud teistele rakendustele.	KOHUSTUS	Kasutatakse AD-st väljastatavaid teenuskontosid läbi keskse tuvastamise teenuse.
Igal komponendil on oma andmebaas mille skeemi ja süsteemsete andmete muudatusi hallatakse komponendiga koos, kui komponent vajab andmete salvestamise võimekust.	KOHUSTUS	Haldab kas rakendus ise või hallatakse eraldi automaatikaga.
Komponendi versiooniuuendusi teostatakse reeglina ilma katkestusteta teenuse töös (tehakse nn. instantsi haaval), andmebaasi muudatuste tegemisel tuleb tagada, et muudatus töötab ka eelmise komponendi versioonidega (kohustuslike väljade mitmeetapiline sisseviimine jms)		Üks võimalikke lahendusi on näiteks "Blue-Green deployment" mehhanism.
Komponent käivitub ka ilma ühenduseta liidestatavate süsteemidega ehk on nn. nõrgalt liidestatud (v.a andmebaasid)		
Komponent töötab osaliselt edasi ka liidestuste katkestuste puhul ja taastab töö peale katkestuste lõppemist, kõik ühendused on olekuta. "Stateful" ühendused nagu RMI ei ole lubatud kasutada.	KOHUSTUS	Kui rakendusel mingi väline liidestus on maas, siis kas kasutatakse "Circuit Breakerit" või näiteks "retry" mehhanismi teatud pikkusega. Kui võrguühendus taastub, taastab komponent oma töö. Kui ootamine ei ole võimalik, peaks kindlasti "healthcheck" teada andma probleemist, mis võimaldab antud rakenduse vajadusel eemaldada teenindavate komponentide hulgast ja hiljem taastada.
Komponendil puudub eraldi väline konfiguratsioonifail - vajalik konfiguratsioon määratakse kas keskkonna muutujatena või kasutatakse muid platvormi pakutavaid võimalusi (Kubernetesi <i>secrets/configmap</i> näiteks).	KOHUSTUS	Rakenduse saladused, mida ei genereerita teenuste poolt automaatselt (näiteks andmebaas või s3) tuleb panna saladuste hoidlasse. Saladuste hoidlast saab saladus liikuda rakendusse ainult läbi CI/CD mehhanismi. Saladuste hoidla jaoks on SMIT-is olemas vastav teenus.
Suurte koormuste teenindamiseks vähemuutuvate andmete puhul kasutatakse rakendusserveris vajadusel vähemälusid mis on kesksed, kiired, kõrgkäideldavad ning ei sõitu konkreetse rakendusserveri instantsist. Ei kasuta mitte cache'de replitseerimist vaid distributeeritud lahendust.	KOHUSTUS	Võimalik kasutada näiteks Redis või olemasolevat andmebaasi selleks ettenähtud tabelite abil.
Komponendil on tööks kõik vajalikud teegid kaasa pakendatud, aliolevalt operatsioonisüsteemis mingite teekide olemasolu eeldada ei tohi.		Soovituslik komponendi pakendamisformaad on Docker konteiner, mis baseerub SMIT sisese Artifactory teenuse sõltuvustel.
Komponent on olekuta ehk kõik mis vaja hoida kauem kui üks süsteemiväline päring (request), salvestatakse kas andmebaasi või mõnda teise hoidlasse.	KOHUSTUS	Oleku hoidmiseks kasutatakse väliseid teenuseid.
Komponent ei tohi eeldada failisüsteemi olemasolu, kus andmed säilitatakse. Mälus võib hoida ühe päringu sees opereeritavate andmete olekuid või andmebaasist taastatavaid cachesid.	KOHUSTUS	
Talletamiseks mõeldud binaarfailide jaoks tuleb kasutada eraldi failide hoidmise teenust, mis pakub vastavat veebipõhist teenust.	KOHUSTUS	SMIT-is on selleks olemas eraldi sisemine S3 objektihoidla.
Komponentide poolt publitseeritavad REST teenused on versioneeritud, dokumenteeritud ning veahaldust tuleb teostada HTTP veakoodidega. Teenused on peavad olema dokumenteeritud OpenAPI spetsifikatsioonile vastavalt ning spetsifikatsioon peab olema vajadusel eraldi kättesaadav kolmandatele osapooltele.	KOHUSTUS	Versiooninumber võib olla kas päises või URI-is. Teenuse tehnilises dokumentatsioonis wikis peab olema link OpenAPI spetsifikatsioonile (SwaggerUI). API disainimisel lähtuda dokumendist: " Rakendustevahelised integratsioonipõhimõtted "
Andmebaas		
Andmebaaside vahelised integratsioonid ei ole lubatud.	KOHUSTUS	
Andmebaase komponentide integratsioonivahendina ei tohi kasutada (mitu erinevat komponenti ühe andmebaasi poole pöörduda ei ole lubatud).	KOHUSTUS	See põhimõte ei tähenda et ühte sama loogika komponenti jõudluse mõistes skaleerides ei tohiks kasutada sama andmebaasi.
Andmeobjektide muutmisel tuleb luua ka migratsiooniskriptid mis teisendavad automaatselt olemasolevad andmed uuele kujule. Migratsiooniskripte on soovivat käivitada paigaldusprotsessi ühe osana ja mitte määrata neid käivitatavaks iga rakenduse restardiga (see võib tekitada ebavajalikke lukke rakenduste restartimisel, kui neid on mitu instantsi)		Kõige lihtsam variant on määrata CI/CD keskkonna muutujaga, kas migratsioon käivitada. Mõistlik on migratsiooni teha ühe instantsiga, kui see tehtud, siis teha vajadusel uuesti paigaldus mitme instantsiga. Alternatiiv on kasutada "job/task" tüüpi paigaldusmehhanisme tavalise rakenduse paigaldamise asemel.

	Andmebaasi ärioloogikat vaikimisi ei kirjutata (protseduurid ja triggerid).	KOHUSTUS	Kõrvalekalded tuleb defineerida ja põhjendada konkreetse komponendi arhitektuuridokumendis - lisaks tuleb ära näidata mehhanismid, kuidas funktsionaalsus töötab andmebaasis, kui kasutusel on mitu andmebaasi instantsi.
	Andmebaasi pöördutakse ainult rakenduse jaoks eraldatud süsteemsete kasutajatena.	KOHUSTUS	Mõeldud on, et andmebaasiga suheldakse läbi rakendusele eraldatud kasutaja. SYS ja POSTGRES tüüpi superuser kontodega baasis käia ei tohi.
	Operatiiv- ja arhiivi andmebaasid on eraldi lahendused, kasutatakse kas eraldi arhiivibaase või mõnda muud spetsiaallahendust.	KOHUSTUS	
	Tekstiotsingute jaoks kasutatakse ainult täisteksti indekseid (Lucene või andmebaasi sisemine täistekstiotsing).	KOHUSTUS	
	Rakendusse tuleb luua võimekus leida vajadusel mitme andmebaasi instantsi seast peamine, kuhu on võimalik kirjutamispäringuid saata.	KOHUSTUS	Eesmärk on, et saab vajadusel andmebaase maha võtta hooldustöödeks, ilma katkestusi tegemata.
Jõudlus			
	Komponendi sisemised alamosad suhtlevad omavahel võimalusel sõnumivahetuse või muude asünkroonsete meetodite abil, et vältida blokeeruvaid lõimesid ja tagada et iga alamtöö töötab eraldi lõimes, kasutades efektiivselt virtuaalmasinate mitut protsessorituumat - see soovitus ei kehti kui komponentide suhtlus peab toimuma ühe kasutaja transaktsiooni sees (mitte segamini ajada andmebaasi transaktsiooniga).		Lisainfo: http://12factor.net/concurrency , http://www.reactivemanifesto.org/
	Pikalt töötavad operatsioonid tuleb viia eraldi taustaprotsessideks, mis toimivad ka mitmete instantside puhul, ehk taustatööde info peab olema salvestatud.	KOHUSTUS	
	Komponent peab peatumisel lõpetama käimasolevad protsessid ning pikad taustaprotsessid peavad pooleli jääma ja käimasolev töö tuleb tagastada tööde nimekirja (näiteks "queue").	KOHUSTUS	Lisainfo: http://12factor.net/disposability
	Komponendis loodud protsessid peavad arvestama, et võib toimuda ootamatu rakenduse seiskumine ning selle tulemusel peavad samuti käimasolevad tööd minema järjekorda tagasi.	KOHUSTUS	
	Jõudluse kasvamisel saab automaatselt komponentide instantside hulka tõsta (horisontaalne skaleerimine) ning koormus jaotatakse instantside vahel laiali.		Ei tohiks kasutada nõ. "scale-up" lahendust, kui seda ei määra konkreetne toote pakkuja.
	Komponent peaks käima minema (olema valmis esimeste päringut teenindamiseks) kuni 60 sekundi jooksul	KOHUSTUS	
Monitooritus			
	Komponentide poolt publitseeritavad teenuste monitoorimiseks vajalik info on kättesaadav kokkulepitud formaadis ja protokolliga - soovitatavalt REST formaadis.	KOHUSTUS	Koormusjaotur peab aru saama, kas suunata liiklus konkreetsele nodele või mitte. Kõige lihtsam näide või näidis selleks on Spring Boot Actuator endpoint <code>/health</code> mis annab infot, kas rakendus on korras või mitte. Seda saab vastavalt täiendada ka lisainfoga. https://docs.spring.io/spring-boot/docs/current/reference/html/production-ready-endpoints.html .
	Komponenti on võimalik monitoorida APM tarkvara agendiga, mis oskab komponendi seest kõikide lähtekoodis realiseeritud protsesside kohta statistikat (sh. kasutamise sagedus, töötuse aeg) väljastada.		Toetatud tehnoloogiad leiab siit: https://docs.appdynamics.com/display/PRO45/App+Server+Agents+Supported+Environments
	Komponent logib enamus tegevused erinevatel tasemetel ning suunab need "stdouti" või sõnumitena sõnumiserverisse (auditlogi). Komponent ei tegele logifailide haldamisega.	KOHUSTUS	Lisainfo: http://12factor.net/logs
Failihaldus			
	Kui teenus võtab vastu faili, siis tuleb seda skaneerida enne baasi või objektihoidlasse salvestamist.	KOHUSTUS	Kasutada tuleb SMIT-i poolt pakutavat MetaVault teenust.
Logimine			
	Erindite (Exception) kinnipüüdmisel tuleb logisse salvestada kogu stacktrace, mitte ainult veateade.	KOHUSTUS	
	Logimisel tuleb arvesse võtta Infoturbeosakonna poolt kehtestatud logimise nõudeid.	KOHUSTUS	Lisainfot leiab dokumendist " Logimise nõuded "
Taaskasutus			
	Uut komponenti arendades, tuleks eelnevalt kontrollida, ega sellist pole juba eelnevalt SMIT-is välja töötatud. Eelistatud on kasutada olemasolevat lahendust teenusena ning äärmisel juhul paigaldada endale toode.	KOHUSTUS	

<p>Olemasoleva teenuse koodi eraldamine eraldi projektiks ja selle iseseisev edasiarendus peaks saama SMIT arhitektuuripaneeli heakskiidu, sellele võiks eelne diskussioon olemasoleva tiimiga, et vajalikud muudatused nemad viiks sisse või aksepteeriksid muudatusettepanekuid.</p>	<p>KOHUSTUS</p>	
<p>Kui arendatakse teenust, millele hakkavad olema kliendid väljaspool konkreetset infosüsteemi/lahendust, tuleks luua sellele iseseisev tehnilise liidestuse võimekus.</p>		<p>Üks võimalik tehniline lahendus on Kuberneteses Operator (https://operatorframework.io)</p>
<p>Arvutitöökoht</p>		
<p>Uued seadmed/lahendused peavad olema Windows või IOS põhised</p>	<p>KOHUSTUS</p>	
<p>Rakendus/lahendus peab olema suuteline vähemalt minor versioone ise uuendada.</p>	<p>KOHUSTUS</p>	
<p>Rakenduse paigalduse või uuendamise käigus EI TOHI rakendus arvuti SERVICEID kinni "Disabled" panna (Näide... DocCheck pani ID kaardi serti edastuse teenuse kinni arvutites)</p>	<p>KOHUSTUS</p>	
<p>Rakendus/lahendus ei tohi olla seotud ühe konkreetse lehitseja või selle versiooniga. Kõik Tehnoloogia valikute lehel olevad lehitsejad peavad olema toetatud ja testitud.</p>	<p>KOHUSTUS</p>	
<p>Kui soovitakse arvutitöökohta paigaldada mingit rakendust/lahendust, siis tarkvara koos konfiguratsiooniga tuleb arenduselt ATKH-le</p>	<p>KOHUSTUS</p>	
<p>Valdkond peab kontrollima enda lahendust standard ATK profiilis või eriprofiilis SMIT tehnikupoolt väljastatud seadmes ja kinnitama kas paigaldus on edukas ja vastab teenuse pakkumise nõuetele. Kui ei ole, siis annab infot mida on vaja konfiguratsioonis muuta.</p>	<p>KOHUSTUS</p>	
<p>Rakendus peab olema käsurealt paigaldatav.</p>	<p>KOHUSTUS</p>	
<p>Rakendus ei tohi nõuda administreerimise õigust kasutamiseks</p>	<p>KOHUSTUS</p>	