

Tehnilised nõuded

Nõue	Täiendav selgitus
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)	Komponentideks jaotamisel mitte lähtuda tehnilistest mõõdikutest 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.
Komponent peab jooksma vähemalt 2 instantsi peal, et vältida platvormi muudatuste käigus tekkida võivaid katkestusi.	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.	Kehtib nii loogika kui kasutajaliidese komponendi puhul. Minimaalne on vaikeprofiil, kui arendajad ei lepi kokku täiendavates nõuetes. Lisainfo on dokumendis Tehnoloogilise võla mõõtmine (SonarQube).
Rakenduse lähtekood on kirjutatud selgusega, mis võimaldab erialast ettevalmistust omaval tarkvaraarendajal süsteemi edasi arendada.	Rakenduse lähtekood ja kommentaarid peavad olema inglise keeles. Rakenduse ärilised muutujad aga eesti keeles, kui neile pole mõlemapoolset loogilist vastet.
Rakenduse lähtekoodi haldus toimib SMITi keskkete versioonihalduspõhimõtete järgi ja SMIT-i GIT-is.	Kehtib nii loogika kui kasutajaliidese komponendi puhul. Lisainfo on dokumendis Lähtekoodi halduse ja ehitamise nõuded.
Keskcondadesse rakenduste paigaldamine on automatiseeritud SMIT CI/CD mehhanismidega.	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.	Konfiguratsioon määratakse keskkonna muutujatena või konfiguratsioonifailidena või kombinatsioonis, kusjuures infot hoitakse Kubernetes objekti „ConfigMap“.
Auditlogi väljastamiseks tuleb kasutada SMIT poolt pakutavat JAVA teeki.	
Rakendustel puudub ligipääs avalikku internetti (sh klientidel). Kui on vajalik	

ligipääs äriandmetele väljaspool SMIT võrku, tuleb tellida webproxy ligipääs.	
Rakenduste sõltuvuste (maven,npm jne) allikana tuleb kasutada SMIT sisest keskest repositooriumi (Artifactory) .	Kõik välised sõltuvused peavad olema sealt kättesaadavad. Komponentide piketeerimiseks ja ehitamiseks CICD/-ga Bamboos saab sõltuvusi kätte ainult SMIT repositooriumist.
Kõik välised sõltuvused tuleb rakenduse ehitamisprotsessikäigus läbi skaneerida Artifactory Xray töövahendiga ja mitte kasutada kõrge kriitilisusega turvanõrkuseid sisaldavaid teeke/komponente.	
Kasutajaliidese sõltuvused/osad nagu JS,CSS, FONTS,GIF jms, mida kasutatakse kasutajaliidese kuvamisel, peavad rakenduse käivitusel tulema samast allikast.	Välise sõltuvuste ja repode kasutamine ei ole lubatud (nagu välised cdn-id, google fonts, github jne). Arendamisel tuleb kontrollida browseri võrguliikluse lehelt, et välisperimeetrisse ei mindaks päringuid tegema.
Välise teenuste nagu ReCaptcha, Google Analytics jms kasutamine ei ole lubatud.	Tuleb kasutada SMIT sisesesest lahendusi.

Testimine

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.	
Peab olema tarnitud end-to-end testid kasutades Playwright raamistikku.	
End-to-end testid peavad olema vastuvõtutestide väärilised ja võtma arvesse vastuvõtukriteeriume.	
Rakenduse automaattestide mahtu tuleb analüüsida CI/CD mehhanismide abil ning tulemused peab publitseerima SMIT SonarQube keskkonnas.	Testide mahu miinimum määratakse Sonari kvaliteedivärvavaga (reeglina 80% kogu koodist).

Kasutajaliides

Kasutajaliides tuleb täielikult eraldi arendada loogikakomponendist ning ei eelda selle implementatsiooni olemasolu.	
Kasutajaliides ja loogikakomponent suhtuvad omavahel üle HTTP protokollile ning ainult läbi eeldefineeritud liidestuse (vahetatakse ainult andmeid).	

Kasutajaliides vahetab loogikakomponendiga ainult andmeid JSON vormignus, visuaalset sisu (nn. html-i javascripti) sellest komponendist ei laeta.	
Kasutajaliides suhtleb loogikakomponentidega üle SSL kanali (SSL termineeritakse koormusjaoturis, kust kõik kasutaja päringud läbi lastakse).	
Kasutajaliidese arendamisel tuleb rakendada micro-frontends arhitektuurimustrit, kasutades single-spa raamistikku.	
Kasutajaliides peab jälgima SMIT stiiliraamatus sätestatut.	

Andmebaas

Andmebaaside vahelised integratsioonid ei ole lubatud.	
Andmebaase komponentide integratsioonivahendina ei tohi kasutada (mitu erinevat komponenti ühe andmebaasi poole pöörduda ei ole lubatud).	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 soovitatav 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.

Logimine

Logimisel tuleb arvesse võtta Infoturbeosakonna poolt kehtestatud logimise nõudeid.	Lisainfot leiab dokumendist Logimise nõuded.
Rakenduse tehnilised komponendid logivad korrelatsiooni ID'd või genereerivad selle ise. Korrelatsiooni ID'd saadetakse iga edasise päringuga kaasa.	
Logi väljund peab olema JSON formaadis.	Java puhul võib kasutada näiteks: <code>net.logstash.logback.encoder.LogstashEncoder</code>