Lisa 2 Nõuded arendusele

Väljavõte SMITi nõuetest arendusele, mis on olulised käesoleva hanke skoobist lähtuvalt.

Üldnõuded

Kohustuslik:

1. 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)
2. Komponent peab jooksma vähemalt 2 instantsi peal, et vältida platvormi muudatuste käigus tekkida võivaid katkestusi.
3. 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 peab pull-requesti kaudu läbima SonarQube Quality Gate-i, enne koodi põhiharusse mergeda ei saa. Täpsemalt dokumendi lõpus „Tehnoloogilise võla mõõtmine (SonarQube)“

   (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.)
4. Lähtekood on UTF-8 formaadis ning tekstilised väärtused tuleb liigutada tõlkefailidesse (i8n), lisaks mitte kasutada Deprecated meetodeid.
5. Rakenduse lähtekood on kirjutatud selgusega, mis võimaldab erialast ettevalmistust omaval tarkvaraarendajal süsteemi edasi arendada. (Rakenduse lähtekood ja kommentaarid peavad olema inglise keeles.)
6. Rakenduse lähtekoodi haldus toimib SMITi kesksete versioonihalduspõhimõtete järgi.
7. Keskkondadesse rakenduste paigaldamine on automatiseeritud SMIT CI/CD mehhanismidega (Bamboo paigaldusplaanid).
8. Kokku ehitatatud rakendust peab saama paigaldada erinevatesse keskkondadesse, ilma et seda peaks uuesti kokku ehitama. (Konfiguratsioon määratakse keskkonna muutujatena või muude rakendusevälistе võimalustega (näiteks Kubernetes configmap).
9. Auditlogi tuleb markeerida ära logikirjetes, et oleks pärast otsingus eristatav.
10. Rakendustel puudub ligipääs avalikku internetti (sh klientidel). Kui on vajalik ligipääs äriandmetele väljaspool SMIT võrku, tuleb tellida webproxy ligipääs.
11. Rakenduste sõltuvuste (maven,npm jne) allikana tuleb kasutada SMIT sisest keskset repositooriumi (Artifactory).

12. Kõik välised sõltuvused tuleb rakenduse ehitamisprotsessikäigus läbi skaneerida Artifactory Xray töövahendiga ja mitte kasutada kõrge kriitilisusega turvanõrkuseid sisaldavaid teeke/komponente.
13. Kasutajaliidese sõltuvused/osad nagu JS,CSS,FONTS,GIF jms, mida kasutatakse kasutajaliidese kuvamisel, peavad rakenduse käivitusel tulema samast allikast. (Väliste 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.)
14. Väliste teenuste nagu ReCaptcha, Google Analytics jms kasutamine ei ole lubatud.
15. Rakenduse arhitektuuri koostamisel tuleks arvestada, et rakenduse koosseisu kuuluvaid komponente peab saama uuendada iseseisvalt.
16. Rakenduse loogikakomponentide üldine sisemine arhitektuur peaks järgima MVC mustrit.
17. Rakendustevahelisi integratsioone teostatakse kokkulepitud põhimõtete järgi (võimalikud variandid on XTEE, HTTP otse või MQ).
18. Kui loodav lahendus koosneb mitmest komponendist, on need lahus arendatavad, pakendatavad, versioneeritavad ja paigaldatavad.
19. Rakendus on versioneeritud kasutades semantilise versioneerimise põhimõtet.
20. Toodangu keskkondades on rakendused on automaatselt monitooritud.

Testitavus

Kohustuslik:

21. Komponendil on olemas minimaalne genereeritav testandmete komplekt.
22. Test- ja toodangukeskkonnad peavad olema üksteisest lahus ning testandmed ei tohi olla toodanguandmed (v.a juhul kui tegemist ei ole avalike andmetega - näiteks aadressid).
23. Loogikakomponendil on olemas integratsioonitestid veebiteenuste või sõnumivahetusteenuste jaoks ning vajadusel täiendavad ühiktestid spetsiiflise äriloogika valideerimiseks.
24. Kasutajaliidese automaattestide kirjutamine, mis simuleerivad kasutaja käitumist põhivoogude taseme, on arendusmeeskonna otsustada.
25. Rakenduse automaattestide mahtu tuleb analüüsida CI/CD mehhanismide abil ning tulemused peab publitseerima SMIT SonarQube keskkonnas.
26. Avaklikke teenuseid tuleb täiendavalt turvatestida.
27. Rakenduste poolt toodetavale logi väljundile tuleb kirjutada automaattestid.

Kasutajaliides

Kohustuslik:

28. Kasutajaliides tuleb täielikult eraldi arendada loogikakomponendist ning ei eelda selle implementatsiooni olemasolu.
29. Kasutajaliides võib korraga suhelda mitme loogikakomponendiga.
30. Kasutajaliides ja loogikakomponent suhtelvad omavahel üle HTTP/WEBSOCKET protokolli ning ainult läbi eeldefineeritud liidestuse (vahetatakse ainult andmeid).
31. Kasutajaliidese olekut hoitakse kliendi poolel.
32. Kasutajaliides vahetab loogikakomponendiga ainult andmeid, visuaalset sisu (nn. html-i javascripti) sellest komponendist ei laeta.
33. Kasutajaliides suhtleb loogikakomponentidega üle SSL kanali (SSL termineeritakse koormusjaoturis, kust kõik kasutaja päringud läbi lastakse).
34. Kasutajaliides on soovitav arendada õhukese kliendina.
35. Avalikud kasutajaliidesed peavad vajadusel järgima VEERA disainistandardit ning omama WCAG tuge.

Äriloogika ja õigused

Kohustuslik:

36. 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.
37. Nii sisemised kui välimised süsteemid peavad kasutaja tuvastamiseks kasutama SMIT kesksest tuvastamise teenust (UAA).
38. Kasutajaid ja nende grupi või rollipõhiseid õiguseid tuvastatakse kesksest Active Directory andmebaasist. Andmetepõhised õigused (ACL) asuvad rakenduse juures andmebaasis.
39. Komponendid suhtlevad omavahel ainult üle HTTP või JMS/AMPQ protokolli. Suurema jõudluse ja sõltumatuse saavutamiseks on eelistatud sõnumivahetus.
40. Kasutaja sessiooni hoidmiseks ei kasutata mälus olevat sessiooni, vaid iga päringuga valideeritakse päringu teostaja "tokeni" kehtivust. Tokenile vastav kasutaja info peab asuma keskses hoidlas, mis peab välistama mitmekordse sisselogimise ja aeguma vastavalt kokkulepitud nõutele.
41. Komponendid peavad ka omavahel saama autoriseerituna (kasutades tokenit) andmeid vahetada analoogselt kasutajaliidesele ilma sessioone tekitamata.
42. Komponentide omavahelises integratsioonis peab iga komponent omama oma kontot, ei tohi taaskasutada kontosid, mis on väljastatud teistele rakendustele.

43. Igal komponendil on oma andmebaas mille skeemi ja süsteemsete andmete muudatusi hallatakse komponendiga koos, kui komponent vajab andmete salvestamise võimekust.

44. Komponendi versiooniuuendusi teostatakse reeglina ilma katkestusteta teenuse töös (tehakse nn. instantsi haaval), andmebaasi muudatuste tegemisel tuleb tagada, et muudatus töötaks ka eelmise komponentide versioonidega (kohustuslike väljade mitmeetapiline sisseviimine jms)

45. Komponent käivitub ka ilma ühenduseta liidestetavate süsteemidega ehk on nn. nõrgalt liidestatud (v.a andmebaasid)

46. **Komponent töötab osaliselt edasi ka liidestuste katkestuste puhul ja taastab töö peale katkestuste lõppemist. Stateful ühendused nagu AMPQ peavad automaatselt taastama oma ühendused.**

47. Komponendil puudub eraldi väline konfiguratsioonifail - vajalik konfiguratsioon määratakse kas keskkonna muutujatena või kasutatakse muid platvormi pakutavaid võimalusi (Kubernetese *secrets/configmap* näiteks).

48. Suurte koormuste teenindamiseks vähemuutuvate andmete puhul kasutatakse rakendusserveris vajadusel vähemälusid mis on kesksed, kiired, kõrgkäideldavad ning ei sõltu konkreetse rakendusserveri instantsist. Ei kasuta mitte cachede replitseerimist vaid distributeeritud lahendust.

49. Komponendil on tööks kõik vajalikud teegid kaasa pakendatud, allolevast operatsioonisüsteemis mingite teekide olemasolu eeldada ei tohi.

50. Komponent on olekuta ehk kõik mis vaja hoida kauem kui üks süsteemiväline päring (request), salvestatakse kas andmebaasi või mõnda teise hoidlasse.

51. Komponent ei tohi eeldada failisüsteemi olemasolu, kus andmed säilitatakse. Mälus võib hoida ühe päringu sees opereeritavate andmete olekuid või andmebaasist taasloodavaid cachesid.

52. Talletamiseks mõeldud binaarfailide jaoks tuleb kasutada eraldi failide hoidmise teenust, mis pakub vastavat veebipõhist teenust.

53. 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 kolmandatale osapooltele.

54. URL ei tohi sisaldada isikuandmeid või sessioonivõtit.

Andmebaas

Kohustuslik:

55. **Andmebaasi ühenduse probleemide puhul tuleb andmete muutmise/lugemise päringutele rakendada kordust (kirjutamisel vähemalt 30s), et päringud õnnestuks, kui baasi funktsionaalsus taastub.**
56. Andmebaaside vahelised integratsioonid ei ole lubatud.
57. Andmebaase komponentide integratsioonivahendina ei tohi kasutada (mitu erinevat komponenti ühe andmebaasi poole pöörduda ei ole lubatud).
58. Andmeobjektide muutmisel tuleb luua ka migratsiooniskriptid mis teisendavad automaatselt olemasolevad andmed uuele kujule. Migratsiooniskripte on soovitav käivitada paigaldusprotsessi ühe osana ja mitte määrata neid käivitatavaks iga rakenduse restardiga (see võib tekitada ebavajalikke lukke rakenduste restartimisel, kui neid on mitu instantsi)
59. Andmebaasi äriloogikat vaikimisi ei kirjutata (protseduurid ja triggerid).
60. Andmebaasi pöördutakse ainult rakenduse jaoks eraldatud süsteemsete kasutajatena.
61. Operatiiv- ja arhiivi andmebaasid on eraldi lahendused, kasutatakse kas eraldi arhiivibaase või mõnda muud spetsiaallahendust.
62. Tekstiotsingute jaoks kasutatakse ainult täisteksti indekseid (Lucene või andmebaasi sisemine täistekstiotsing).
63. Objektid identifitseerida registrikoodide abil.

Jõudlus

Kohustuslik:

64. Komponendi sisemised alamosad suhtlevad omavahel võimalusel sõnumivahetuse või muude asünkroonsete meetodite abil, et välistada blokeeruvaid lõimesid ja tagada et iga alamtöö töötab eraldi lõimes, kasutades efektiivselt virtuaalmasinate mitut protsessorituuma - see soovitus ei kehti kui komponentide suhtlus peab toimuma ühe kasutaja transaktsiooni sees (mitte segamini ajada andmebaasi transaktsiooniga). Andmebaaside vahelised integratsioonid ei ole lubatud.
65. Pikalt töötavad operatsioonid tuleb viia eraldi taustaprotsessideks, mis toimivad ka mitmete instantside puhul, ehk taustatööde info peab olema salvestatud.
66. Komponent peab peatumisel lõpetama käimasolevad protessid ning pikad taustaprotsessid peavad pooleli jääma ja käimasolev töö tuleb tagastada tööde nimekirja (näiteks "queue").
67. Komponendis loodud protsessid peavad arvestama, et võib toimuda ootamatu rakenduse seiskumine ning selle tulemusel peavad samuti käimasolevad tööd minema järjekorda tagasi.
68. Jõudluse kasvamisel saab automaatselt komponentide instantside hulka tõsta (horisontaalne skaleerimine) ning koormus jaotatakse instantside vahel laiali.

69. Komponent peaks käima minema (olema valmis esimeste päringut teenindamiseks) kuni 60 sekundi jooksul.

Monitooritavus

Kohustuslik:

70. Komponentide poolt publitseeritavad teenuste monitoorimiseks vajalik info on kättesaadav kokkulepitud formaadis ja protokolliga - soovitavalt REST formaadis.
71. Komponenti on võimalik monitoorida APM tarkvara agendiga, mis oskab komponendi seest kõikide lähtekoodis realiseeritud protsesside kohta statistikat (sh. kasutamise sagedus, töötluse aeg) väljastada.
72. Komponent logib enamus tegevused erinevatel tasemetel ning suunab need "stdouti" või sõnumitena sõnumiserverisse (auditlogi). Komponent ei tegele logifailide haldamisega.

Failihaldus

Kohustuslik:

73. Kui teenus võtab vastu faili, siis tuleb seda skaneerida ennem baasi või objektihoidlasse salvestamist. Kasutada tuleb SMIT-i poolt pakutavat MetaVault teenust.

Logimine

Kohustuslik:

74. Erindite (Exception) kinnipüüdmisel tuleb logisse salvestada kogu stacktrace, mitte ainult veateade.
75. Logimisel tuleb arvesse võtta Infoturbeosakonna poolt kehtestatud logimise nõudeid.
76. Rakenduse tehnilised komponendid logivad korrelatsiooni ID'd või genereerivad selle ise. Korrelatsiooni ID'd saadetakse iga edasise päringuga kaasa.
77. Logi peab olema JSON formaadis.