

DigiDoc4J kasutuselevõtu kogemuslugu Maa-ameti näitel

Villem Vannas

18.05.2016



Projekti algseis

- eKataster toodangukeskkonnas aastast 2014 - Allkirjastamine DDOC, kasutusel oli JDigidoc, lisandus ka BDOC, mida see teek toetas (BDOC-TM)
- eKataster peab võimaldama laadida arhiivi digiallkirjastatud dokumente, neid lahti pakkida ja menetleda
- eKataster väljastab digiallkirjastatud dokumente menetlusprotsessi tulemusena
- eKataster väljastab ka digitembeldatud dokumente



Projekti ärilised vajadused

- Vajalik oli vastu võtta igas formaadis BDOC dokumente ja ka neid välja andma (BDOC-TS, ASIC-E)
- Vajalik oli väljastada BDOC digitembeldatud dokumente
- Vajalik oli tagada mobiil-ID allkirjastamine kõikide veebilehitsejatega (probleem oli seoses Chrome uuendustega)



Projekti tehnilised vajadused

- JDigidoc teegi arendus lõppeb, asemele tuleb DigiDoc4J
- RIA poolt tungiv soovitus oli kõigil üle minna uuele teegile



Olukord projekti alguses

- Kui alustasime olid kahtlused: viimased arendused aastast 2013, teek tundub olevat alfa või beeta seisus
- Valida oleks kas oma lahenduse tegemine või kasutada DigiDoc4J
- RISK: Hoiatasime tellijat Maa-ametit, et me ei pruugi selle projektiga tähtaegselt lõpetada ega toodangusse saada
- Maa-ameti otsus: Võtame riski
- Digitembeldamise osas polnud vajalik muud teha kui uuendada tarkvara



Arendus

- Probleem: teek omab palju sõltuvusi teistest teekidest
- Osad teegid on kasutusel ka eKataster tarkvaras kuid versioonide erinevused
- Mida teha?
- Otsus: Digiallkirjastamise jaoks luuakse omaette teenuskomponent, et eKataster ja digiallkirjastamine „ei läheks sõlme“

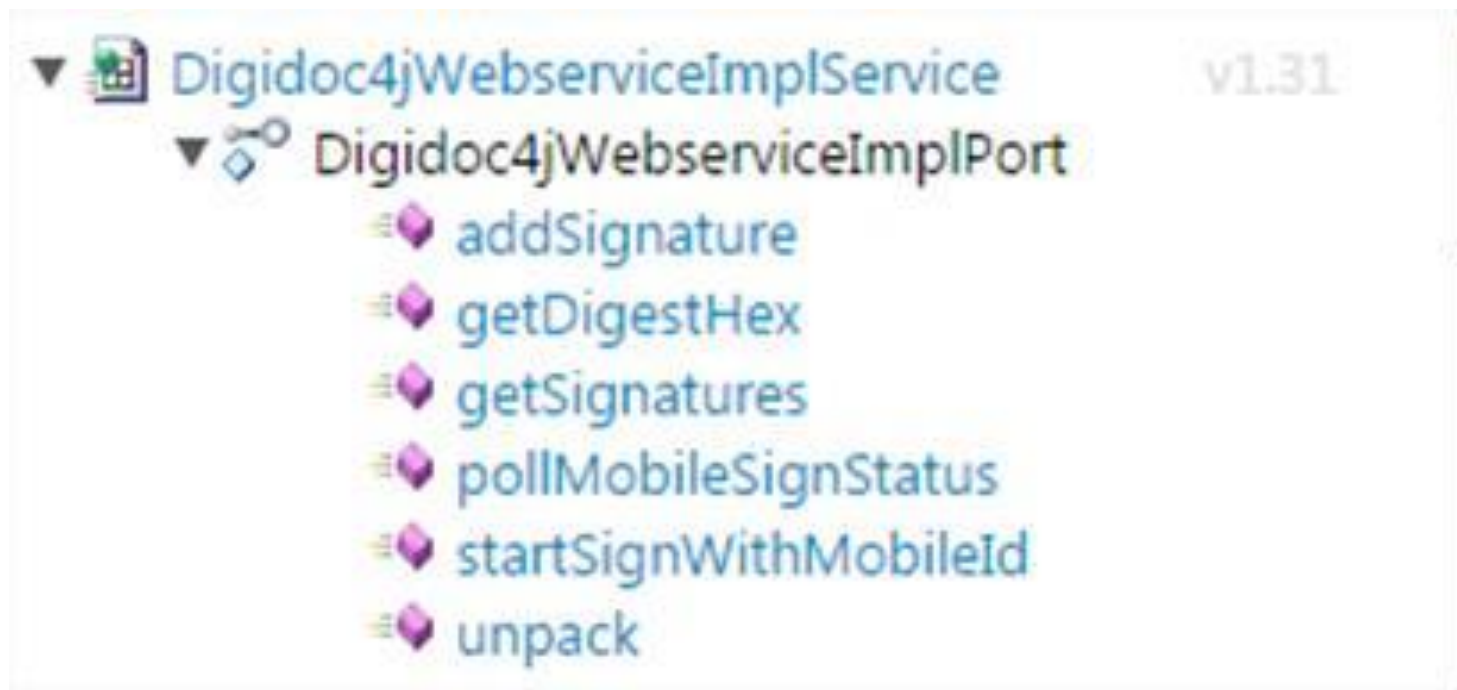


Arendus

- Loodav teenuskomponent **digidoc4jws** – SOAP liidesega komponent, liidese kirjeldab WSDL
- Sarnaste ülesannete puhul võimalik kasutada ka teistes projektides
- Tehtud teenuskomponent ja ka liideskomponent, mis eKataster projekti koosseisus **digidoc4jwsclient**

Komponent

Piirdusime esialgu järgmiste meetoditega:





Meetodid

- **unpack** – pakib konteineri lahti ja valideerib – tagastab leitud vead, vajalik allkirjastatud dokumendi vastuvõtmisel ja dokumentide laadimiseks menetlusmoodulisse või edaspidi digiarhiivi edastamiseks
- **getSignatures** – tagastab dokumendi allkirjad, vajalik registris metaandmetes

Meetodid

- **getDigestHex** – tagastab kliendi poolt allkirjastatava konteineri räsi (sisend – nimekiri failidest, säilitab järgmiseks allkirjastamiseks)
- **addSignature** – lisab dokumentide konteinerile allkirja (sisend – nimekiri failidest, leiab eelmise serialiseeritud objekti mälust)
- **startSignWithMobileId** – algatab konteineri allkirjastamise Mobiil-ID-ga (kasutab väliseid teenuseid)
- **pollMobileSignStatus** – kontrollib, kas Mobiil-ID allkiri on antud või on sellest tegevusest loobunud, või oodatakse endiselt kasutaja tegevuse lõppemist

Töövoog

- Allkirjastatavad dokumendid või avatav allkirjastatud fail paigutatakse failisüsteemi kausta
- Kutsutakse välja vajalik teenus
- Saadakse tulemus failisüsteemi kaustast
- Kui midagi läheb valesti, siis tagastab teenus veearaporti



Probleemid testimisel

- Varasem eKataster funktsionaalsus võimaldas allkirjastamata konteineri kokku panna enne allkirjastamist ning kasutaja selle üle vaadata. DigiDoc4J seda ei võimalda, pidime rakenduse äriloogikat muutma
- Varasem JDigidoc kasutasime ka mobiil-ID allkirjastamisel (addRawSignature), siin teegis kasutamisel esines „Not yet implemented“ Exception - tuli teha asendus, et neid tegevusi asendada
- Lõpuks saime testimisse üle anda kliendile, esmased testid edukad
- Peaaegu oleks isegi 2015 aasta lõpus toodangukeskkonda levitatud



Probleemid jätkutestimisel

- Ei õnnestu üles laadida ja avada välistes süsteemides allkirjastatud mõningaid konteinereid
 - Postipoiss „\“ probleem: vigane mime tüüp „application\rtf“ näiteks
 - XML metaandmete fail oli UTF-8-bom alguses
 - Erinevalt vanast teegist, valideerib teatud mime tüüptidega failide sisu – kas dokument on terviklik: näiteks PDF dokumente, piirab kasutusjuhte
- Vajalik DigiDoc4J arendaja tegevus – meie jaoks ootamine

Veel probleeme

- Vanas teegis sertifikaadid pakendatud samasse rakenduse konteinerisse, uus kasutab vaikimisi sertifikaatide listisid ja tõmbab need kohale
- Esines probleem: Belgia sertifikaatide server ei vastanud, vastuse ootamine serverilt põhjustas nii suure viite, et toimingud teegiga muutusid ebamõistlikult aeglaseks. Päringud aegusid
- Sertifikaatide laadimine oli aeglane, cache probleem
- Vajalik oli DigiDoc4J arendaja tegevus – meie jaoks ootamine
- SK õnneks parandas need vead nüüd ruttu
- On aeglasem kui vana teek, siin oleks veel arenguruumi



Veel probleeme

- uue teegi arendusjuhendis soovitati kasutada SHA256 räsi, mis pole toetatud vanemate ID-kaartide poolt, inimesed ei saanud allkirjastada
- Lahendus – teha oma äriloogikas: kontrollida ID-kaardi versiooni ja saata vajadusel lahjem räsi - SHA192



Toodangukeskkonda

- Märtsis – õnnestus lõpuks toodangu keskkonnas juurutada, vead olid parandatud
- Hiljem selgunud veel
 - .asic laiendiga failid (vaja teha täiendus meie süsteemis)
 - Tulevikus kui peaksime võtma uue versiooni DigiDoc4J: Meie lahenduse jaoks on tehtud eraldi DigiDoc4J haru, loodame, et tulevikus uues versioonis on ka need täiendused, mis tehti



Kokkuvõte

- Projekt venis üle tähtaja 3 kuud
- Õnneks oli tellija see, kes võttis selle riski, aitas kaasa, et vead parandatud saaks
- SK parandas vead lõpupoole kiirelt
- Nüüdseks on DigiDoc4J saanud paremaks ja toodangukeskkonnas kasutusel kaks kuud – lahendus töötab
- Tehtud on eraldi SOAP server, mida saab kasutada komponendina ka mujal projektides



Aitäh kuulamast!