



JY:N CASE EXAM-INSTANSSIN AUTOMATISOIDUSTA YLLÄPIDOSTA.

Keskitetyn ja korkeakoulukohtaisen
ylläpitoratkaisun vertailua

Markku Närhi, JY



Teemoja

- Miksi JY oma ylläpito ja integraatiot
- Korkeakoulun tietojärjestelmäkokonaisuus ja Exam osana OPI arkkitehtuuria
- Exam sovelluksen tekninen ylläpito JY
- Oma vai yhteinen ylläpito?
- Pohdintaa ja kommenttipuheenvuoro



Lähtötilanne 2015

- JY:lla kiire korvata vanha (sofittutor) eTentti-järjestelmä
- Tarve tehdä migraatio vanhasta datasta (tentit) uuteen järjestelmään
- Muilla ei vastaavaa integraatiomallia, ei synergiaetuja
- CSC palvelumalli ei vielä valmis ja yhtään tuotantoasennusta ei ollut valmiina
- EXAM versio 1.3? 1. asennus, tuotantoon 2.0 versiolla, softa voimakkaassa kehitysvaiheessa, jolloin paljon päivityksiä ja ongelmia joiden nopea ratkaiseminen keskeistä
- Softa piti 1.kerralla kääntää uudelleen jotta saatiin käyntiin
- Päivitetty lähes jokainen uusi versio ja välillä otettu myös suoraan repositosta väliversio sekä korjattu bugeja itse
- Nyt ajetaan uusinta versiota muutamalla omalla fixillä



Sovellus ja mihin kokonaisuuteen se liittyy?



FI SV EN

Markku Närhi

Työpöytä

Kysymysluonnokset

Tentit

Varaukset

Kirjautu ulos



Yleinen tentti

PERUUTA

Tentin nimi:

Testi

Tallenna

Tentin perustiedot

Tenttiperiodi alkaa:

11.02.2016



Tenttiperiodi päättyy:

12.03.2016



Opintojakso:

Opintojakson koodi:

KANS135



Opintojakson nimi:

Kansalaiskasvatus



Opintojakson laajuus (op):

5



Organisaatio:

Jyväskylän yliopisto



Linkki ilmoittautumiseen ja paikanvaraukseen:

<https://exam.app.jyu.fi/#/enroll/KANS135/exam/451>

Tentaattorit: ?

Lisää tentaattori

Markku Närhi

Tentin arvioijat: ?

Lisää arvioijia

Markku Närhi



Kysymysluonnokset

Opintojakso ▾

Tentti ▾

Avainsana ▾

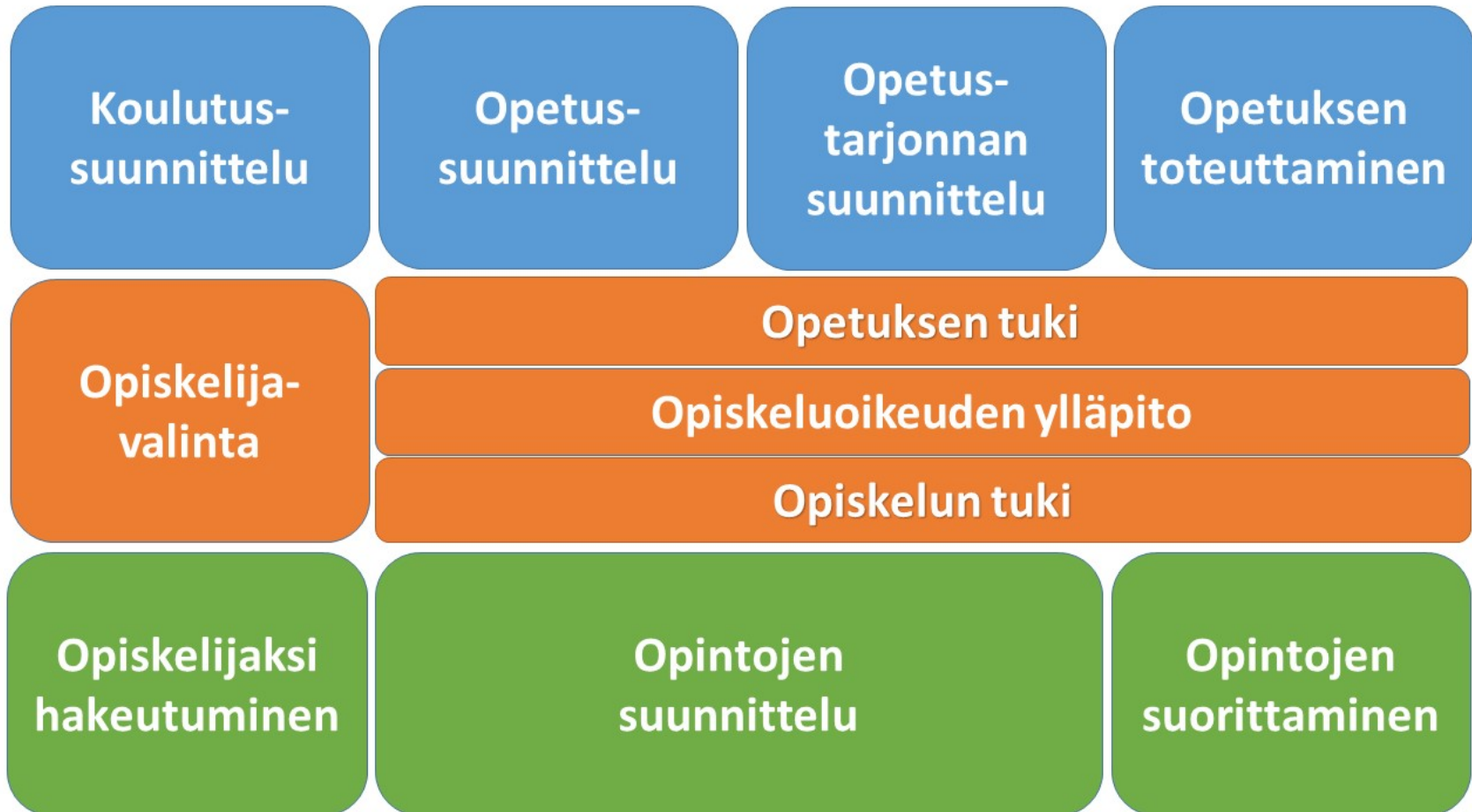
Kasvatus- ja tieteenfilosofia

Hae



L
i
s
ä
ä
k
a
i
k
k
i

OPI Viitearkkitehtuuri



Palvelujen koostamisen skenaario, palvelut

Opiskelijan ja opettajan työpöytä, PLE - näkymä

Opetuksen resurssi-suunnittelu ja OPS palvelut

OPS- ja opintojen vuosisuunnittelujärjestelmä
Korppi- järjestelmä
www-julkaisujärjestelmä

Opintopolku,
koulutustarjonta, opintoihin hakeutuminen, opiskelijavalinta

Opintojen suunnittelun ja seurannan palvelut

Korppi- järjestelmä/OTM
ePortfolio
eHOPS- järjestelmä/OTM
Opintopalaute (webropol)
Mobility Online

Opetuksen palvelut
Kurssihallinta, materiaalit, opiskeluprosessi, tuotokset

Optima
Moodle
Koppa
Moniviestin (viideojakelu)
Kirjasto (Nelli, Finna Jyx)
Korppi- järjestelmä/OTM
Exam (eTentti)
Connect (verkkokokous)
TIM
Pedanet

Muu palvelut

0365-palvelut
Google Apps
Yammer
CRM (mm. alumni)
SOME liitännäiset

Tausta- palvelut
Talous, HR, tietovarasto, verkkomaksut

Opintohallinnon palvelut , (ROTI /OTM)
Opinto-oikeudet, rekisterit, tutkinnot, AHOT, liikkuvuus, KV

Kansalliset hallinnonalan palvelut, Opintopolku, OILI, tunnistus, UAF, Vipunen , kurssikoodistot, sanastot

Kansalliset peruspalvelut, palveluväylä mm VTJ ,tunnistus, VIRTATA, OID , sanastot, koodistot

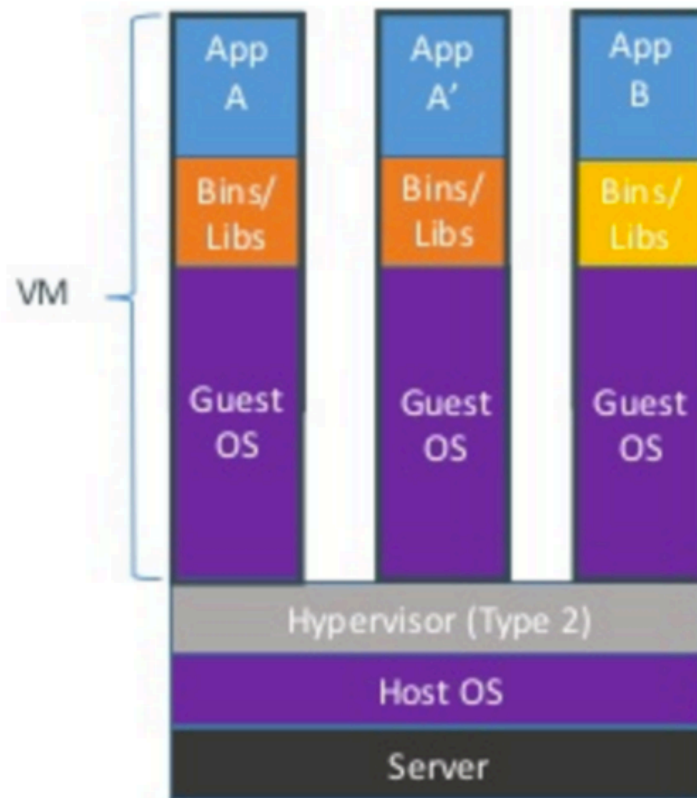
EXAM tekninen ylläpito

- Oma konesali ja domain alla
 - VMware klusteri ja levyjärjestelmät sekä palvelinvirtualisointi
 - RHEL stack, alusta ylläpito standardilla tavalla
- Sovelluspalvelujen ylläpito DevOps tiimillä
 - www-palvelut, LMS , CMS,
 - Koodaajat, ylläpitäjät samassa, Agile kehitys
 - Docker paketointi ja Ansible orkestrointi
- Pääkäyttäjä ja käyttäjätukipalvelu lähellä tiimiä
 - Palvelutukijonot, ohjeet ja koulutus
 - Yhteys opetuksen kehittämiseen ja opintopalveluihin



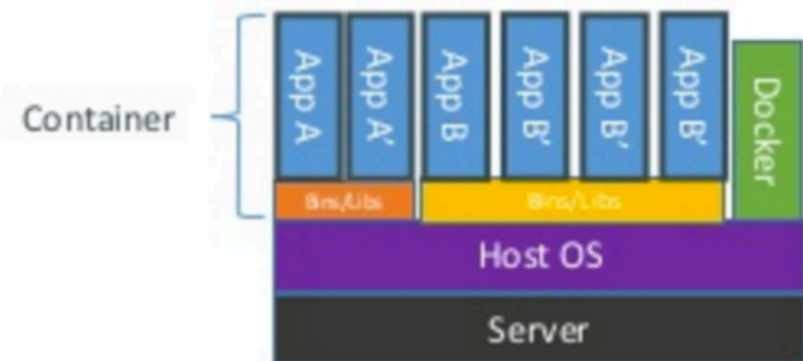
Docker vs perinteinen virtualisointi

Containers vs. VMs



Containers are isolated, but share OS and, where appropriate, bins/libraries

...result is significantly faster deployment, much less overhead, easier migration, faster restart



kuvalehde: Docker Inc.

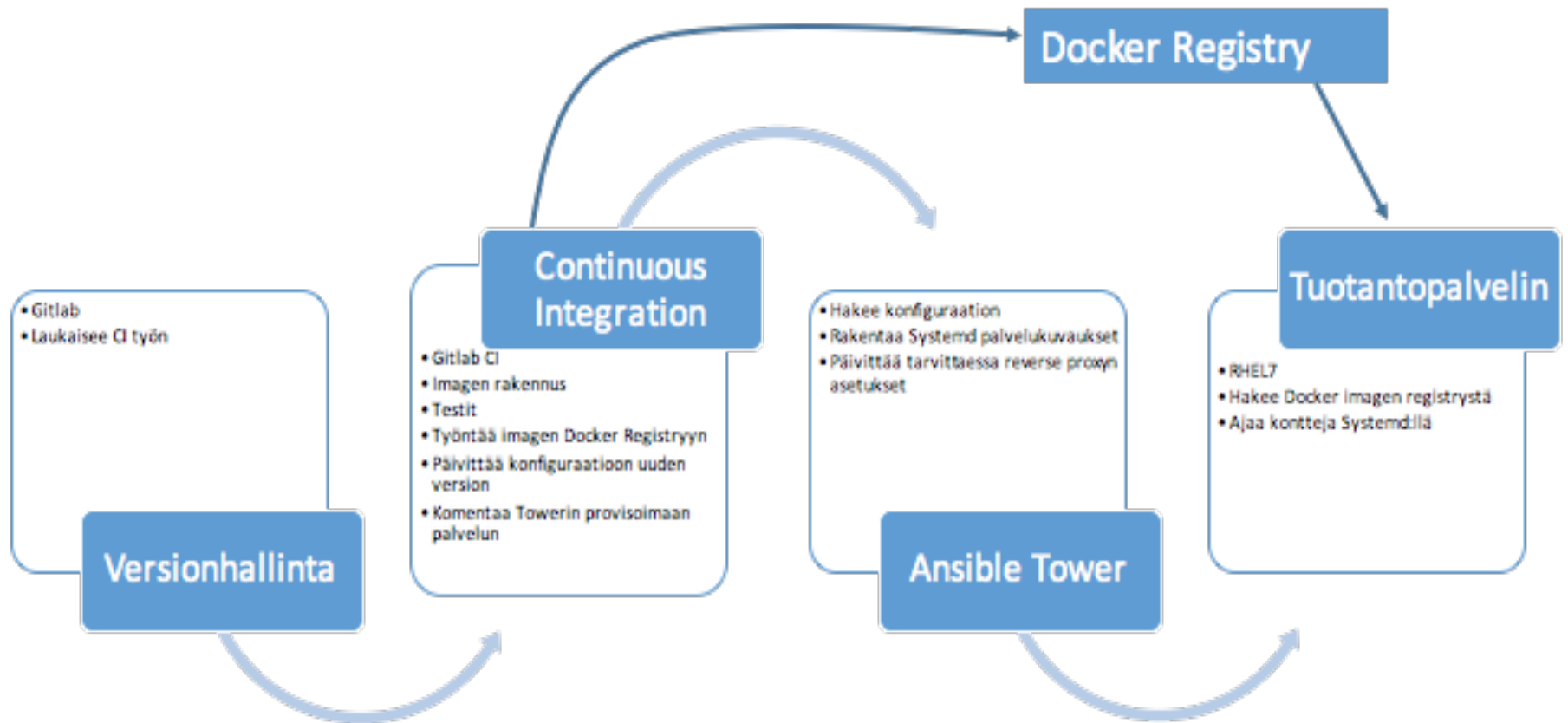


Exam-Docker-malli:

- Exam käännetään ja paketoidaan täysin itsenäiseen Docker-pakettiin jolla ei ole riippuvuuksia sitä ajavaan palvelimeen
- Exam ei sisällä Docker-paketointia, mutta omaa koodia vaadittiin kymmenkunta riviä (Dockerfile)
- Examin tietokanta sijaitsee perinteisellä erillisellä monen sovelluksen kesken jaetulla tietokantapalvelimella
- Examin levyille kirjoittama tallessa pidettävä data (liitetiedostot) kirjoitetaan Dockerin ulkopuolelle (atm suoraan palvelimelle, toivottavasti tulevaisuudessa erilliseen tallennuspalveluun)
- Dockerissa ajettu Exam käynnistyy aina täysin puhtaana, koska se tuhoutuu pysyvästi sammuessaan
- Kaikki käännetyt Exam-docker - paketit ovat säilössä sisäisessä rekisterissämme, josta voidaan valita haluttu Exam-versio ajoon käytännössä Examin uudelleenkäynnistyksellä



Tekninen ylläpitomalli



EXAM ylläpito

■ Oman Exam-ylläpidon etuja:

- Kysytään käyttäjiltä suora palaute
- Lyhyet, onnistuneet päivityskatkot
- Oma sovellus- ja pääkäyttötuki lähellä ylläpitoa

■ Oman Exam-ylläpidon haittoja:

- Ongelmatapauksien selvitys on ollut vaikeampaa kun Exam-kehittäjät eivät pääse tutkimaan paikallista tilannetta/tilaa, eivätkä he aina ole onnistuneet toistamaan ongelmaa omassa ympäristössään
- Ongelmatapaukset kiertävät oman talon sisällä muutaman, ehkä turhan, välihypyn ennen varsinaisille kehittäjille päätymistä
- Paikallisten kehittäjien täytyy selvittää ongelmakohtaisesti onko vika Examissa vai ympäristössä, välillä lukemalla Examin koodeja
- Jokainen versiopäivitys vaatii jonkin verran paikallista testausta ja

18.2.2016 työtä



Yhteistyö Examissa

- Flowdock on hyvä kanava nopeaan viestintään
- Ongelmiin on vastattu nopeasti ja henkilökohtaisesti, realiaikainen kommunikaatio on tuntunut tehokkaammalta kuin sähköposti yksinkertaisissa kysymyksissä
- Muutama nopea korjaus on saatu tunneissa ajoin nimenomaan JY:lla ilmenneisiin ongelmiin
- Lähdekoodin käsilläolo on hyvä (julkinen lähdekoodi olisi vielä parempi) (kehittäjien mielipide)
- Lähdekoodi oikeasti avoimeksi olisi kaikille parasta. Toimiminen avoimen lähdekoodin projektien periaatteella. Nyt on liian yksisuuntaista. (kehittäjien mielipide)



Pohdintaa

■ Yhteiset palvelut

- Kun palvelu on stabiilissa tilassa ja tuotteistettu
- Osaaminen erikoistunutta ja skaalautuvaa
- Organisaatioiden yhteiskäyttöisyys keskeistä
- Standardit vakiintuneet integraatiot
- Ylläpidon yksikkökustannukset keskeiset

■ Omat palvelut

- Paljon omia integraatioita
- Kehitys nopeaa , paljon päivityksiä joilla keskinäisiä riippuvuuksia
- Omia räätälöintejä, koostettuja profiilinäkymiä
- Ylläpidon ja kehittämisen nopeuden suhde

kokonaispalveluun ja käyttäjäkokemukseen/palautteeseen

18.2.2016





KIITOS

- Kysymyksiä?
- Kommentteja?

