

## Sovellusprojekti Kepler, Korppi-palaveri

Aika Tiistai 10.2.2015 klo 9:15–10:30  
Paikka Jyväskylän yliopisto, Agora, Sovellusprojektien kokoushuone C226.1

### Läsnä

Atte Rätty, puheenjohtaja  
Sakari Juutinen  
Anu Koskela  
Mikko Kuhno  
Henrik Paananen  
Petri Partanen  
Kari Patana, saapui kohdan 9 lopussa  
Panu Rahkila  
Jukka-Pekka Santanen  
Panu Wetterstrand  
Joonas Konki, sihteeri

## Pöytäkirja

Pöytäkirja laadittu: 12.2.2015  
Muokattu: 17.2.2015

### 1 Palaverin avaus

Rätty ehdotti puheenjohtajaksi itseään ja sihteeriksi Konkia. Ehdotus päätettiin hyväksyä, ja puheenjohtaja Rätty avasi palaverin.

Projektiryhmä jakoi osallistujille esityslistan.

### 2 Esityslistan hyväksyminen

Esityslista hyväksyttiin ilman muutoksia.

### 3 Osallistujien esittäytyminen

Osallistujat esittäytyivät lyhyesti.

### 4 Organisaation esittely

Puheenjohtaja Rätty aloitti kuvailemalla sovellusprojektin ja palaverin tavoitteita lyhyesti. Tämän palaverin tavoitteena on keskustella fysiikan laitoksen oppilaslaboratorion laboratoriotöiden varausjärjestelmän toteuttamisesta. Vaihtoehdot ovat joko toteuttaa se osaksi Korppi-järjestelmää tai siitä erillään. Molemmat ratkaisut vaativat yhteistyötä Korppi-kehittäjien ja IT-palveluiden kanssa.

## 5 Sovelluksen tarpeet

Rahkila kertoi sovelluksen pääasiallisten tarpeiden olevan varausjärjestelmän käyttäjien autentikointi JYU-tunnuksilla, perustietojen haku (mm. nimi ja sähköposti) autentikoinnin jälkeen ja varaustietojen syöttäminen käyttäjien Korppi-kalenteriin näkyville (feed). Nykyinen varausjärjestelmä ei ole yhteydessä Korppi-kalenteriin.

## 6 Mahdollinen Korppi-yhteistyö

Wetterstrand kertoi, että Korpissa on käytössä vanhanaikainen SyncML ja jonkinasteinen iCal-vienti kalenterin synkronisointiin. SyncML-ominaisuuksien päivittäminen ja korjaaminen on hankalaa, ja iCal-tuontia ei ole toteutettu tällä hetkellä. Jos kalenteritapahtumia halutaan lisätä tai poistaa, niin on parempi kehittää jokin uusi ratkaisu.

Autentikointirajapinnaksi ollaan ottamassa lähiaikoina käyttöön uusi OAuth2.0, jolla käyttäjä valtuuttaa sovelluksen käyttämään omia tietojaan, mutta suurempia oikeuksia sovelluksella ei suoraan ole. Sovellukselle oikeuksia voidaan tarvittaessa antaa erikseen.

OAuth2.0:n ideana on, että sovellukselle annetaan oikeuksia käyttäjän tunnuksella ja salasanalla. Sovellus saa luvan kutsua 3. osapuolen ja omia palveluita, joihin käyttäjä antoi luvan, kuten käyttäjätietojen haku ja käyttäjän kalenterin muokkaus HTTP API:n yli (iCal-feed yleisesti käytetty). Yhdellä käyttöliittymälle yhteisellä yläpalkilla hoidettaisiin näin pääsy yliopiston useisiin erillisiin sovelluksiin, kuten Optima, Koppa ja kalenteri.

Rahkila tiedusteli, miten kalenteritapahtumat saataisiin näkyviin sekä työparien varaukset ja peruutukset hoidetaan. Tähän sovellukselle tarvitaan käytännössä järjestelmän ylläpito-oikeudet. Wetterstrandin mukaan kalenteritapahtumien poistamisessa varausjärjestelmän pitäisi tehdä paljon erilaisia toimintoja automaattisesti. Tämä on yksinkertaista, kunhan käyttäjältä ei tarvitse kysyä mitään. Synkronoiduissa laitteissa (puhelimet ym.) ei ole tarvittavia tietoja näiden toimintojen toteuttamiseksi. Korppia on muokattu niin, että tapahtumia ei välttämättä oikeasti poisteta, vaan vain niiden näyttäminen synkronoidun laitteen kalenterissa häviää, ja tämä muistetaan järjestelmässä. iCal-feedin tuonti kalenteriin näyttäisi vain tapahtumat, mutta tätä tukea ei tällä hetkellä ole Korpissa.

Wetterstrandin näkemyksen mukaan varausjärjestelmän toteuttaminen osaksi Korppia ei ole kehittäjien ja käyttäjien kannalta helpoin vaihtoehto. Korppia ollaan purkamassa useiksi erillisiksi palveluiksi. Nykyään Korppiin lisätään mielellään vain todella nopeita ja helppoja muutoksia. Lisäksi uusien ominaisuuksien, kuten käyttöliittymäpäivitysten, tuotantoon saaminen vaatii järjestelmän alasajon ja uudistusten ottamisen käyttöön seuraavassa huoltoikkunassa, joka saattaa olla vasta kahden viikon päästä. Toteutettavan sovelluksen tuotantoon saaminen ja testaaminen hidastaisivat kehitystä tällöin merkittävästi.

Erillisenä sovelluksena toteutettaessa sovellusta voidaan päivittää ja testata useammin Korpin päivitysykleistä riippumatta. Santasen mukaan IT-palvelut ovat aiemmin tarjonneet sovelluksen käytön ja toimivuuden testaukseen virtuaalipalvelimia. Konki huomautti lisäksi, että projektiryhmän käytettävissä oleva aika puoltaa sovelluksen toteuttamista erillisenä sovelluksena.

## 7 Kehitysmahdollisuuksien läpikäynti

Varausjärjestelmän pääsyoikeuksia tulisi pystyä rajoittamaan autentikoinnin lisäksi esimerkiksi fysiikan kurssien opinto-oikeuksien perusteella. Toisaalta ei ole ehkä tarpeellista näyttää OAuth2.0:aan liittyvässä yläpalkissa oppilaslaboratorion varausjärjestelmän linkkiä kaikille yliopiston palveluita käyttäville. ROTIsta (opinto-oikeuksia ja -suorituksia hallinnoiva järjestelmä) saataisiin tarvittavat opinto-oikeustiedot. Tietokanta erillisessä järjestelmässä tulisi olla erillinen.

Kalenteri-feedit halutaan toimimaan molempiin suuntiin (tuonti ja vienti). Korppiin ei tallennettaisi tapahtumatietoja, vaan siirreltäisiin pelkkiä näkymiä. Wetterstrandin mukaan Korpista saanee feedin ulos jo nyt, mutta se pitää toteuttaa vielä sisään päin. Tälle ominaisuudelle on ilmennyt kysyntää, joten se pyritään toteuttamaan.

Wetterstrand ehdotti, että ryhmätiedot kannattaa toteuttaa erillisen sovelluksen sisällä, koska ne eivät ole kovin staattisia. Ryhmät kannattaa sijoittaa koodausteknisesti rajapinnan taakse.

Wetterstrand mainitsi useita yliopiston sovelluksissa käytettyjä ja tutustumisen arvoisia kehitystyökaluja: Mercurial-versiohallinta, Go-ohjelmointikieli, käyttöliittymien toteutuksessa hyödynnetty React JS -kirjasto, Node.js ja Plone-järjestelmissä käytetty Python-ohjelmointikieli. Docker-sovellus on käytössä uusien pienten moduulien käyttöönotossa, mihin tutustuminen voisi auttaa varausjärjestelmän käyttöönotossa ja erityisesti sovelluksen jatkokehityksessä.

Wetterstrand korostaa kiinnittämään huomiota lähdekoodin laatuun. Vaikka esimerkiksi Java-ohjelmointikieli olisi tuttu lukijalle, niin 50000 vaikeasti ymmärrettävää riviä tekevät ohjelmasta vaikeasti ylläpidettävän. Kommentit ja versiohallinnan commit-tiedot hän suositteli kirjoittamaan projektille yhtenäisellä kielellä. Jos käytetään englannin kieltä, niin domain-alueen termit, kuten luokat ja metodien nimet, kannattaa kirjoittaa suomeksi kriittisten aihealueen termien osalta, kuten varaus, getVaraus, opintojakso ja henkilötunnus. He ovat käyttäneet englannin kieltä, mutta siitä ei ole ollut merkittävää hyötyä, koska kehittäjät ovat olleet suomenkielisiä. Kehitystyökalujen valinta riippuu lähinnä kehittäjistä.

### Päätökset:

- Tapahtumien näyttäminen toteutetaan molemmissa järjestelmissä (Korppi ja Kepler) kahdensuuntaisilla ICal syötteillä.
- Korppiin ei tallenneta tapahtumatietoja, vaan siirretään pelkkiä kalenterinäkymiä ja -tietoja.

## 8 Jatkokehitysmahdollisuudet

Wetterstrand mainitsi alustavasti jatkokehityksen olevan mahdollista Kehittämispalveluiden kehittäjien toimesta. Tähän vaaditaan kuitenkin päätös ylemmältä taholta.

## 9 Muut esille tulevat asiat

Sovelluksen tarvitsemista rajapinnoista on sovittava Patanan ja Wetterstrandin kanssa. Dokumentaatiota saatavilla olevista ja tulevista rajapinnoista on tulossa jossain vaiheessa.

Vaitioloovelvollisuuksia ei ole toteutusten kannalta, mutta mitään tahallisesti tai tahattomasti kohdattuja henkilötietoasioita ei saa tietenkään levittää.

Wetterstrand esitteli IT-palveluilla käytössä olevaa infrastruktuuria uusien sovellusmoduulien käyttöönoton osalta tussitaululla (Docker).

Tässä kohdassa palaveriin saapuneen Patanan mielestä sovelluksen toteuttaminen Korpin osaksi ei olisi mielekästä, mikäli Korppiin tarvitaan suuria muutoksia.

Projektiorganisaatio päätti, että sovellus kehitetään Korpista erillään omana sovelluksena. Lisäksi päätettiin, että projekti toteutetaan mielellään mahdollisimman yleisessä muodossa niin, että muut yliopiston laitokset voivat hyödyntää kehitettyä sovellusta pienin muokkauksin varausjärjestelmään. Varattavien asioiden määrittely onärkevintä toteuttaa rajapinnan taakse.

#### **Päätökset:**

- Kepler-projekti kehittää Korpista erillisen sovelluksen.
- Sovelluksesta pyritään tekemään yleiskäyttöinen varausjärjestelmä myös muiden laitosten hyödynnettäväksi.
- Vaitioloovelvollisuuksia ei ole, mutta tahallisesti tai tahattomasti kohdattuja henkilötietoasioita ei saa levittää.

## **10 Läsnaolijoille sovitut toimenpiteet**

Projektiryhmän jäsenet:

- Tekevät taustatutkimusta kehitystyökalujen valitsemiseksi seuraavassa projektiryhmän virallisessa palaverissa.
- Selvittävät muiden yliopiston laitosten varausjärjestelmätarpeita.
- Sopivat tarvittavista rajapinnoista Patanan ja Wetterstrandin kanssa.

Kari Patana ja Panu Wetterstrand:

- Toimittavat projektiryhmän jäsenille dokumentaation saatavilla olevista ja tulevista rajapinnoista.

## **11 Seuraavan palaverin aika ja paikka**

Seuraava palaveri on tiistaina 17.2.2015 klo 9:15 kokoushuoneessa Agora Ag226.1, mikäli se on vapaana. Paikka ilmoitetaan myöhemmin.

## **12 Palaverin päättäminen**

Puheenjohtaja päätti palaverin.

## **Jaettu materiaali**

- Ryhmä jakoi kaikille läsnäolijoille palaverin esityslistan.
- Santanen jakoi Patanalle ja Wetterstrandille tilaajan toimittaman aihekuvauksen ja sitä tarkentavat sähköpostit.