Sopimuksen xx liite 1.1

# Suomen Hippos
# HEPPA-NYKYJÄRJESTELMÄN KUVAUS

# SISÄLTÖ

# 1. JOHDANTO

## 1.1 TAUSTAA

Suomen Hippos ry on laajentamassa Heppa-järjestelmän (jäljempänä pelkkä Heppa) tarjoamia sähköisiä palveluita. Kehitettävät palvelut ovat:

- jalostustietojärjestelmä
- hevosen tunnistus- ja rekisteröintiprosessi
- hevosen lääkeainekirjanpito

Palveluiden toteutus edellyttää sekä täysin uusien liiketoimintakomponenttien toteutusta, että muutoksia olemassa oleviin komponentteihin. Paikoin nykyiset komponentit voivat olla käyttökelpoisia (lähes) muutoksitta.

Hepan toteutusprojekti on käynnistynyt syksyllä 2005 ja se on mennyt tuotantoon helmikuussa 2008. Käyttöönotosta lähtien siihen on tehty Hippoksen toimesta jatkokehitystä. Järjestelmän tekninen dokumentaatio on pääosin ajan tasalla.

Järjestelmää ylläpidetään edelleen aktiivisesti ja käynnissä on hanke, jossa ajoalusta päivitetään uudempaan. Samassa yhteydessä luodaan pohjaa jatkokehitykselle, jossa erityisesti käyttöliittymän osalta voidaan käyttää tuoreempaa teknologiaa kuin mitä tähän saakka on käytetty, kun kyseessä on kokonaan uusi toiminnallisuus. Sellainen jatkokehitys, jossa tehdään muutoksia nykyiseen käyttöliittymään, joudutaan pitäytymään vanhassa esityskerrosteknologiassa.

## 1.2 TARKOITUS

Tämän dokumentin tarkoituksena on kuvata Heppa sitä ennestään tuntemattomalle lukijalle siten, että lukija saa kuvan järjestelmän yleisestä toiminnallisuudesta, laajuudesta ja toteutusteknologiasta huomioiden myös käynnissä oleva teknologiauudistus. Erityinen kohderyhmä tälle dokumentille ovat tarjouskilpailuun osallistuvien yritysten tekniset asiantuntijat.

# 2. JÄRJESTELMÄN YLEISKUVAUS

## 2.1 TOIMINNALLISET KOKONAISUUDET

Uusien palveluiden kannalta keskeisimmät käyttöliittymän toiminnalliset kokonaisuudet ovat:

- Hevoset
    - Hevosen perustiedot
    - Suku ja jälkeläiset

    - Ravikilpailuhistoria
    - Astutus/Tiineystiedot
    - Omistus- ja hallintahistoria
    - Väliaikaiset tuonnit
    - Hevosen näyttelyhistoria
- Oma talli
    - Omat hevoset

Yläpuolinen lista on vain hyvin pieni osajoukko Hepan täydestä toiminnallisuudesta.

Osa Heppa-järjestelmän toiminnallisuudesta on käytettävissä anonyymisti ilman kirjautumista. Järjestelmän osoite on http://heppa.hippos.fi/heppa/app.

### 2.2 KÄYTTÄJÄT, ROOLIT JA KÄYTTÖOIKEUDET

Sovelluksessa on roolitettu JAAS-pohjainen käyttöoikeushallinta. Rooleja käyttäen sovelluksessa voi antaa katselu ja/tai muokkausoikeuksia sovelluksessa toteutettuihin käyttöliittymäsivuihin ja toimintoihin. Käyttöoikeushallintaan ei ole nyt tehtävän jatkokehityksen yhteydessä nähtävissä välitöntä jatkokehitystarvetta.

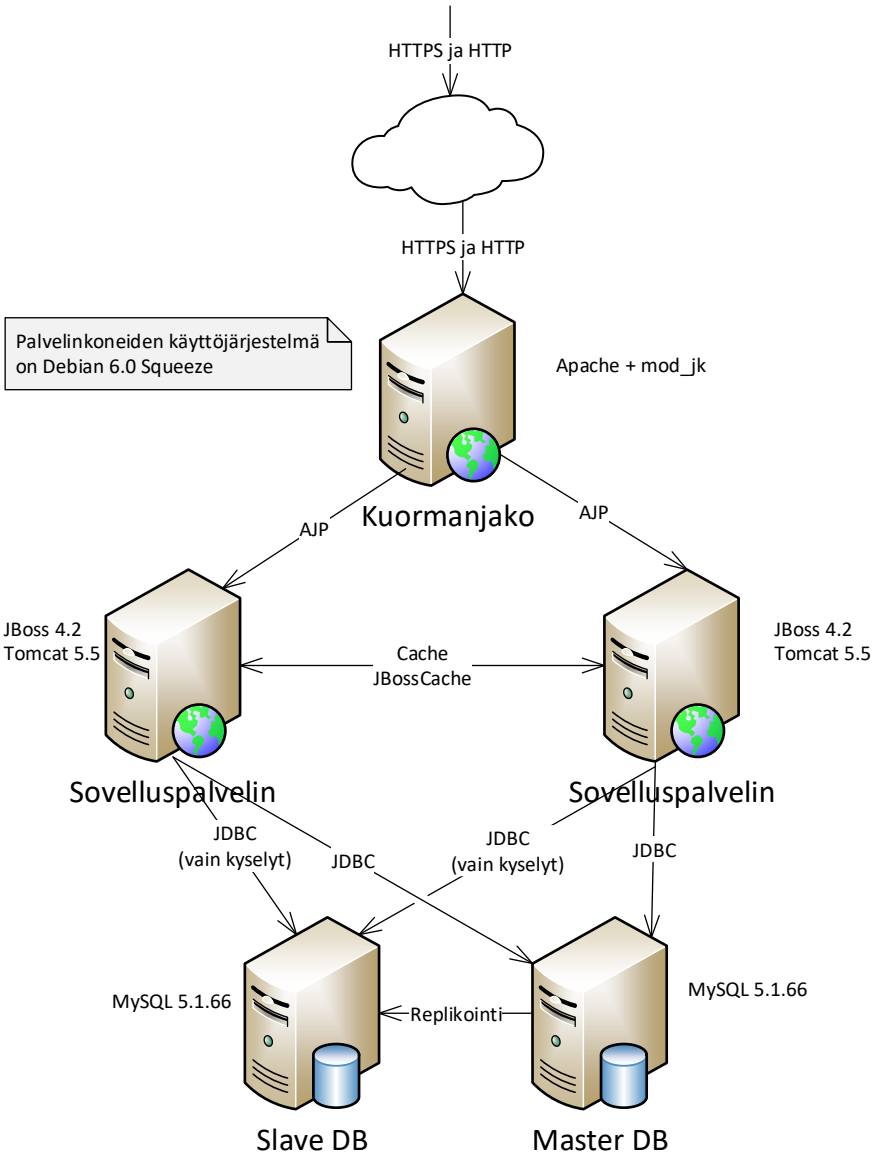
## 3. TEKNINEN KUVAUS

Tämä osio kuvaa keväällä 2016 tuotannossa olevan alkuperäisen sovellusarkkitehtuurin, sekä sen jälkeen toteutetun ajoalustapäivityksen, sekä järjestelmäarkkitehtuurin, johon uudet sähköiset palvelut on toteutettu.

Alkuperäisen sovellusarkkitehtuurin tunteminen on tärkeää myös uusia palveluita tehtäessä, koska vanhan arkkitehtuurin mukaista toteutusta ja toiminnallisuutta ei hylätä ajoalustapäivityksen yhteydessä. Ajoalustapäivitys mahdollistaa modernimman esityskerrostoteutuksen, kun tehdään täysin uutta käyttöliittymää. Liiketoimintakerros säilyy ytimeltään ennallaan.

## 3.1 ALKUPERÄINEN SOVELLUSARKKITEHTUURI

### 3.1.1 PALVELIMET




Kuva 1. Yleiskuva nykyjärjestelmän palvelinarkkitehtuurista.

Palvelimet ovat fyysisiä Linux-palvelimia Hippoksen omassa konehuoneessa.

Kuormanjakopalvelimena on Apache, johon on asennettu mod_jk-plugin. Kuormanjako-palvelimen ja JBossin web-palvelimen välissä on käytössä AJP-protokolla.

Sovelluspalvelimena on JBoss ja web-palvelimena on JBossin Tomcat.

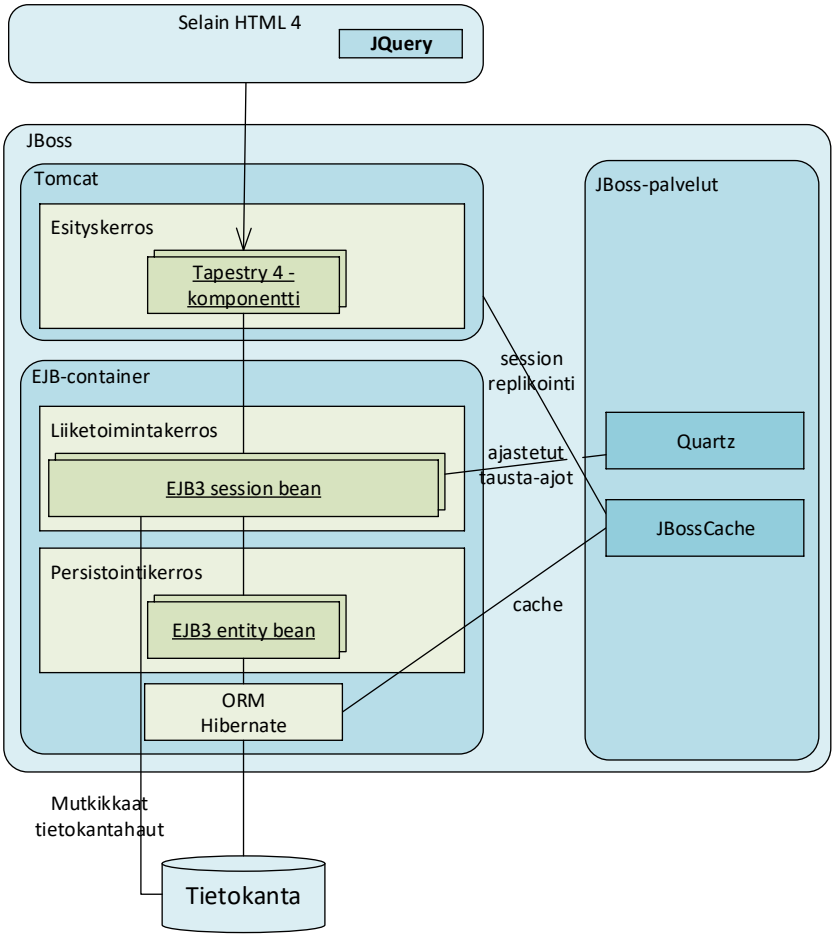
Sovelluspalvelimet ovat samassa JBoss-klusterissa. Välimuistissa olevat tiedot ja sessiot replikoituvat JBossCachen toimesta.

Tietokantapalvelimissa on käytössä master-slave –konfiguraatio. Automaattista failover-mekanismia ei ole toteutettu.

### 3.1.2 TOTEUTUSTEKNOLOGIAT JA -VÄLINEET

| Nimi | Kuvaus |
|---|---|
| Tapestry 4 | Esityskerroksessa käytetty sovelluskehys. |
| Java | Java 8 |
| JBoss | Sovelluspalvelin |
| Tomcat | Web-palvelimena käytetään JBossin Tomcat:ia |
| Quartz | Tausta-ajojen ajastus |
| JBossCache | EJB3 entity-komponenttien välimuisti ja web-session replikointi palvelinten välillä. |
| Altova Stylevision | PDF-tulosteiden xslt-tiedostojen (xsl:fo) muokkaamisessa käytetty kaupallinen tuote. Jatkokehityksen aikana FOP:in edellyttämiä xslt-tiedostoja on tehty myös ilman tätä tuotetta manuaalisesti muokkaamalla. |
| Apache FOP | PDF-tulosteiden muodostamiseen ajonaikaisesti käytetty xsl:fo –prosessori. |
| JavaScript | Sovellus sisältää pienessä mittakaavassa selaimessa suoritettavaa JavaScript-koodia. |
| Subversion | Versiohallinta |
| Ant | Sovelluksen buildit |

### 3.1.3 SOVELLUKSEN RAKENNE




Kuva 2. Sovelluksen sisäinen rakenne ennen uudistusta.

Esityskerroksessa on käytetty Tapestry 4:ää. Tapestryn versio on vanha eikä se sovellu responsiivisen käyttöliittymän toteuttamiseen. Muutokset, jotka kohdistuvat olemassa olevaan käyttöliittymään, tarkoittavat muutoksia Tapestry-komponentteihin. Esityskerros käyttää liiketoimintakerroksen komponentteja.

Web-sessio replikoituu sovelluspalvelinten välillä JBossCachen toimesta.

Sekä liiketoimintakerroksen, että persistointikerroksen EJB-komponentit on konfiguroitu sovelluskoodissa olevilla annotaatioilla. XML- konfiguraatiota ei ole käytetty.

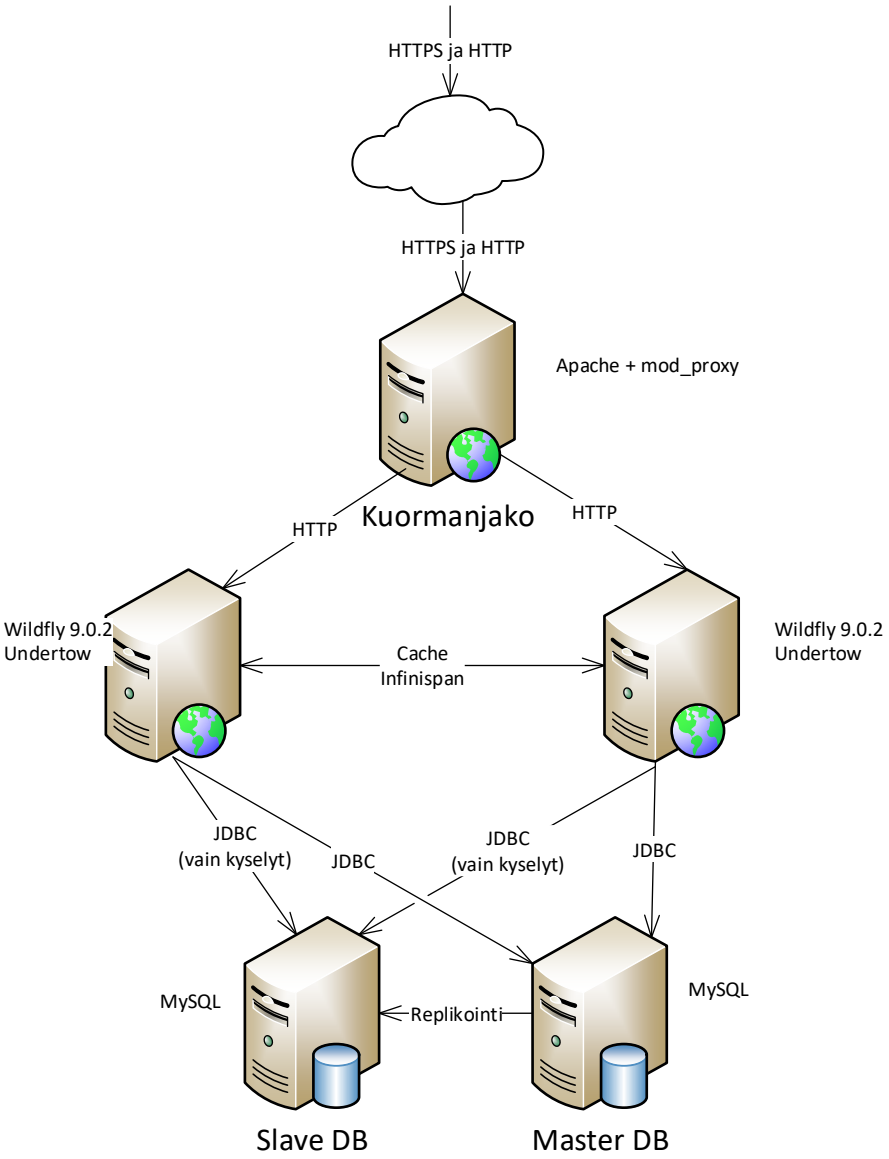
Liiketoimintakerros on toteutettu EJB 3 stateless session bean –komponentteina. Liiketoimintakerros suorittaa tietokakantakäsittelyn pääsääntöisesti persistointikerroksen komponentteja käyttäen. Poikkeuksen muodostavat monimutkaiset haut, jotka on toteutettu suorina SQL-kyselyinä.

Persisointikerros on toteutettu EJB3 entity bean –komponentteina, joiden tietokantakäsittelystä vastaa Hibernate. Hibernate on konfiguroitu käyttämään JbossCachea välimuistina.

Järjestelmän eräajojen ajastuksiin on käytetty Quartz-ajastinta, joka toimii JBoss-palveluna. Eräajot on toteutettu liiketoimintakerroksen stateless bean –komponentteina. Kukin tausta-ajo on näin ollen

## 3.2 AJOALUSTAPÄIVITYS

### 3.2.1 PALVELIMET




Kuva 3. Yleiskuva palvelinarkkitehtuurista uudistuksen jälkeen.

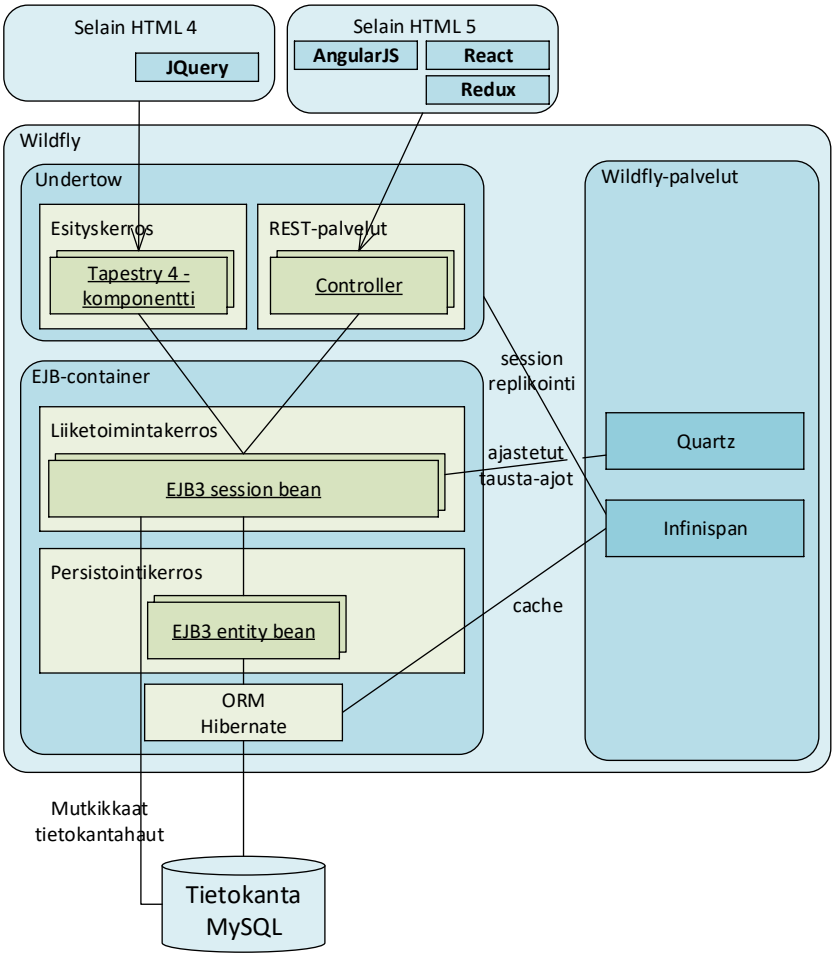
Merkittävin tekninen uudistus palvelinarkkitehtuurissa on vanhan ajoalustan päivittäminen JBoss-sovelluspalvelimesta Wildfly-palvelimeksi. Web-palvelimena on Wildfly:n Undertow.

### 3.2.2 TOTEUTUSTEKNIIKAT JA VÄLINEET

| Nimi | Kuvaus |
|---|---|
| Tapestry 4 | Olemassa olevassa esityskerroksessa käytetty sovelluskehys. Nykyisen käyttöliittymän toteutustekniikkaa vaihdetaan vain, jos muutokset ovat niin suuria, että koko muutoksen alaisen osuuden tekeminen kokonaan uudestaan on kannattavaa. |
| AngularJS 2.0<br>React + Redux | Uudet käyttöliittymäkokonaisuudet toteutetaan käyttöliittymän osalta nojaten selaimessa ajettavaan JavaScript-sovelluskehykseen. Lopullista valintaa sovelluskehyksen osalta ei vielä ole tehty, mutta vahvimmin ovat esillä AngularJS ja React+Redux –yhdistelmä. Näistä React+Redux on erityisesti AngularJS:n versiotilanteen vuoksi vahvempi ehdokas.<br><br>AngularJS:n osalta olennainen tekijä on sen 2.0 –version valmistumisaikataulu. Toteutustekniikkaa ei tulla pohjaamaan sen 1.x versioon.<br><br>Käyttöliittymän toteutuksessa tulee noudattaa Hippoksen valitsemaa teknologiaa. Jatkokehitys ja ylläpito mutkistuvat tarpeettomasti, jos käytössä on useita eri sovelluskehyksiä. |
| Java 8 | Sovelluskoodi päivittyy Javan versiosta 5 versioon 8. Tämä koskee koko koodipohjaa, eli myös vanha koodi käännetään ajettavaksi JVM 8:ssa.<br><br>Vanhan sovelluskoodin osalta tämä on vain eräänlainen ”ajoalustan päivitys”. Se pysyy ennallaan eikä sille tehdä yleistä refaktorointia, jossa otettaisiin käyttöön uudempien java-versioiden mahdollistamaa kehittyneempää tekniikkaa. Esimerkkejä näistä ovat java 6:n tuoma generics tai java 8:n tuoma mahdollisuus käyttää myös funktionaalista ohjelmointia.<br><br>Uuskehityksessä edellytetään täysipainoista Java 8:n mahdollisuuksien hyödyntämistä ja myös funktionaalisen ohjelmointityylin käyttö on suositeltavaa. |
| Wildfly 9.0.2 | Sovelluspalvelin |
| Undertow | Web-palvelimena käytetään Wildfly:n Undertow:ta. |
| Quartz | Tausta-ajojen ajastus |
| Infinispan | EJB3 entity-komponenttien välimuisti ja web-session replikointi palvelinten välillä. |
| Altova Stylevision | Vanhojen PDF-tulosteiden xslt-tiedostojen (xsl:fo) muokkaamisessa on käytetty tätä kaupallista tuotetta. Muutoksien tekeminen näihin tulosteisiin ilman Stylevisionia voi olla työlästä. |
| Apache FOP 0.20.5 | PDF-tulosteiden muodostamiseen ajonaikaisesti käytetty xsl:fo –prosessori. |
| iText 2.0.8 | Uusien PDF-tulosteiden toteutuksessa voidaan käyttää tätä kirjastoa. Tukeutuminen FOP:iin ei ole välttämätöntä. |
| JavaScript ja/tai<br>TypeScript | Vanha käyttöliittymä sisältää pienessä mittakaavassa selaimessa suoritettavaa JavaScript-koodia.<br><br>Uuden käyttöliittymän myötä JavaScriptin ja/tai TypeScriptin määrä kasvaa merkittävästi. Jos käyttöliittymän sovelluskehys on AngularJS, tulee pääasiallinen selaimen toteutuskieli olemaan TypeScript. |
| Git | Versiohallinta |

| Nimi | Kuvaus |
|---|---|
| Maven | Sovelluksen buildit |

### 3.3 UUSI ARKKITEHTUURI




Kuva 4. Yleiskuva sovellusarkkitehtuurista uudistuksen jälkeen.

Uusi sovellusarkkitehtuuri eroaa vanhasta käyttöliittymätoteutuksen osalta. Web-palvelimessa olevan vanhan käyttöliittymän esityskerroksen rinnalle tulee uuden käyttöliittymän tarvitsema REST-palvelut tarjoava kerros.

Liiketoimintalogiikkakerrosta ei ole mielekästä uusia jo toteutettujen komponenttien osalta. Wildfly sisältää Java EE 7 version, joten kokonaan uuden toiminnallisuuden toteutuksessa voidaan hyödyntää Java EE 7 tarjoamia mahdollisuuksia, mutta tässä kuvatusta kerrosrakenteesta ja EJB-komponentteihin pohjaavasta toteutuksesta ei luovuta tässä vaiheessa.

# 4. TIETOKANTA

Tietokannan taulurakenne on järkevälle tasolle normalisoitu hyviä suunnitteluperiaatteita noudattava SQL-tietokanta. Taulut ja sarakkeet on nimetty suomeksi käyttäen Hippoksen terminologiaa.

Heppa-järjestelmän laajuuden vuoksi taulumäärä on kohtuullisen suuri ja taulujen välillä on runsaasti riippuvuuksia. Tietokannassa ei ole käytetty triggereitä eikä proseduureja. Näkymiä on määriteltynä vain muutamia.

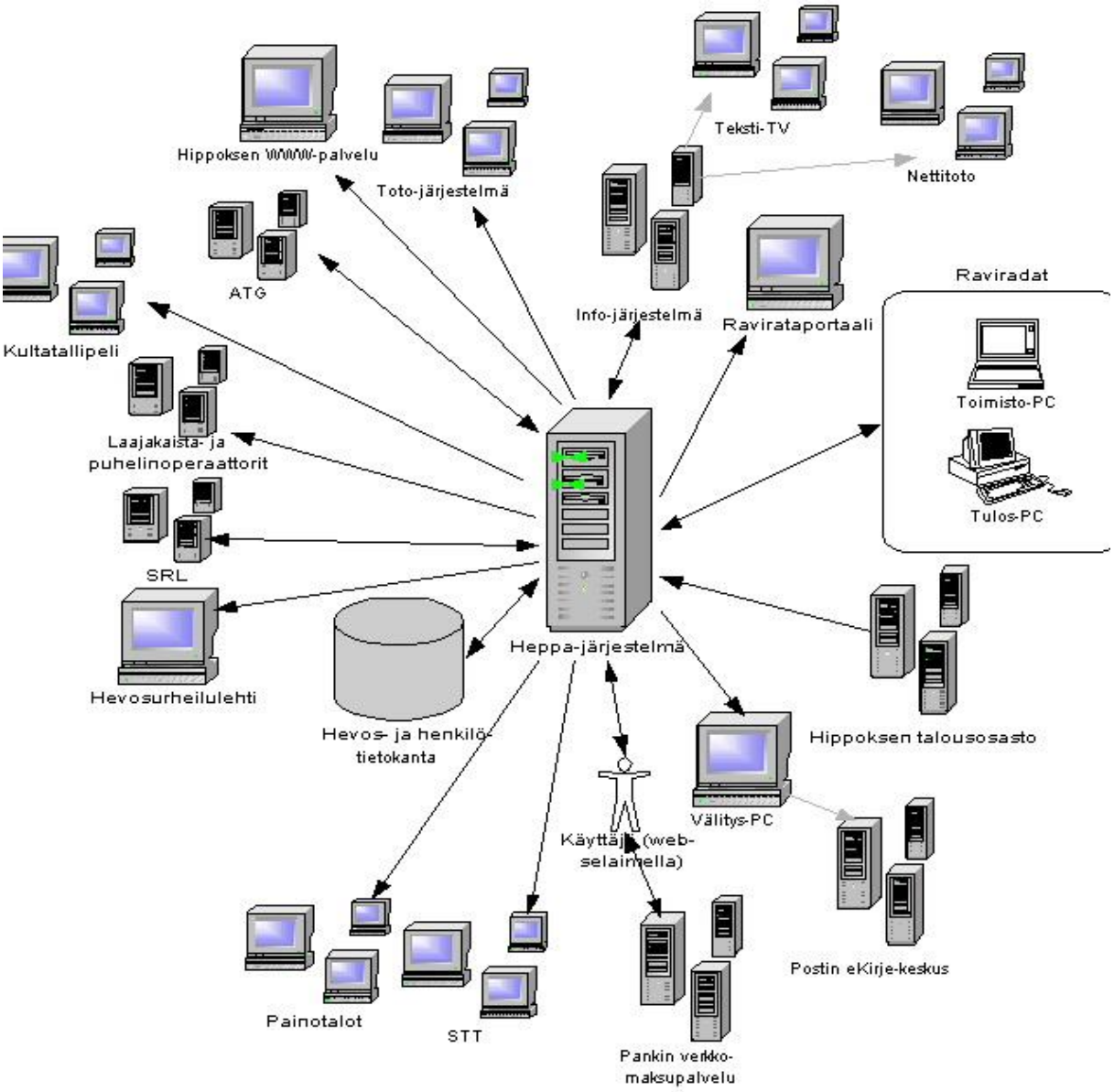
Sarakkeiden nimeämisessä on käytetty seuraavia yleisperiaatteita:

- taulun pääavaimen sisältävä sarake on nimeltään id
- lukumäärätietoa sisältävän sarakkeen nimen loppu on _lkm.
- totuusarvoisen sarakkeen nimen loppu on _ke
- foreign key –sarakkeen nimi sisältää kohdetiedon nimen (taulun nimi, jos mahdollista) ja sen loppu on _id
- päivämääräsarakkeen nimen loppu on pvm.

Tietokannan monimutkaisuutta kuvaavia lukumäärätietoja:

- tauluja on noin 130 kpl
- foreign key –määrittelyitä on noin 220 kpl
- indeksejä pääavaimen lisäksi on noin 120 kpl

# 5. ULKOISET LIITTYMÄT




Kuva 5. Järjestelmän liittymät käyttöönottohetkellä 2008.

Järjestelmällä on lukuisia ulkoisia liittymiä. Tämän kappaleen tarkoituksena on vain tuoda esille se, että liittymiä on ei-vähäinen määrä ja on mahdollista, että uusilla palveluilla on sidoksia johonkin liittymistä. Yläpuolinen kuva havainnollistaa visuaalisesti liittymien laajuutta, mutta sen yksityiskohtiin ei pidä kiinnittää huomiota, koska kuva kertoo liittymätilanteen järjestelmän käyttöönottohetkellä.

## 6. TILASTOTIETOA SOVELLUKSEN LAAJUUDESTA

Alapuolinen taulukko kuvaa sovelluskoodin laajuutta keväällä 2016.

| Tieto | Arvo |
|---|---:|
| Java-luokkien lukumäärä | 1 504 |
| Rivimäärä yhteensä Java-tiedostoissa | 453 071 |
| Lähdekoodirivien lukumäärä Java-tiedostoissa | 308 135 |
| Java-luokkia kpl, joissa rivimäärä < 10 | 17 |
| Java-luokkia kpl, joissa rivimäärä 10 – 50 | 290 |
| Java-luokkia kpl, joissa rivimäärä 50 – 100 | 328 |
| Java-luokkia kpl, joissa rivimäärä 100 – 200 | 348 |
| Java-luokkia kpl, joissa rivimäärä 200 – 500 | 322 |
| Java-luokkia kpl, joissa rivimäärä 500 – 1000 | 119 |
| Java-luokkia kpl, joissa rivimäärä 1000 – 2000 | 49 |
| Java-luokkia kpl, joissa rivimäärä > 2000 | 31 |

## 7. KÄYTETTÄVISSÄ OLEVA DOKUMENTAATIO

Sähköisten palveluiden toteutusprojektin käytettävissä on Heppa-järjestelmän toteutusprojektin tuottama dokumentaatio. Järjestelmää on tämän jälkeen aktiivisesti jatkokehitetty kahdeksan vuoden ajan, mutta dokumentaatiota ei ole jatkokehityksen aikana ylläpidetty. Ylläpitotöistä on olemassa Trac-tikettijärjestelmässä tikettejä, mutta kokonaiskuvan saaminen siitä miten järjestelmä on muuttunut jatkokehityksen aikana voi olla vaikeaa.

Alapuolinen taulukko kuvaa vain osajoukon dokumentaatiosta. Siihen on koottu dokumentit, joista todennäköisimmin on todellista hyötyä järjestelmään perehtyville henkilöille, jotka eivät sitä ennestään tunne.

| Dokumentti | Kuvaus |
|---|---|
| Heppa-järjestelmä, Tekninen suunnittelu, Versio 0.33.doc | Word-dokumentin sivumäärä on 119 ja se sisältää varsin kattavan kuvauksen koko järjestelmän teknisen toteutuksen periaatteista.<br><br>Jatkokehityksessä on noudatettu toteutusprojektin teknisen dokumentaation antamia suuntaviivoja, joten alkuperäinen dokumentaatio on tältä osin edelleen validia ja sen avulla perehtyminen järjestelmään helpottuu huomattavasti. |
| tk_heppa(1.5).xls | Excel-tiedosto sisältää kommentoidut tietokannan taulujen luontiskriptit, primary key –määrittelyt, foreign key –määrittelyt ja näkymämäärittelyt.<br><br>Tiedosto palvelee edelleen melko hyvänä referenssimateriaalina, mutta se ei ole kattava, koska sisältöä ei ole ylläpidetty alkuperäisen toteutusprojektin päätyttyä. |
| Käyttöoikeuksien_hallinta_työversio.doc | Dokumentti sisältää tarkan kuvauksen järjestelmän käyttöoikeushallintatoteutuksesta. |
| Heppa_käyttöliittymämäärittely_v1.6.doc | Tämä on 206-sivuinen dokumentti, joka voi joissain tilanteissa olla hyödyllinen tehtäessä muutoksia olemassa olevaan käyttöliittymään. Tämän dokumentin lukijan on syytä muistaa, että se on monin paikoin pahasti vanhentunut. |
| Heppa_toiminnallinen_määrittely_v1_0_34b.doc | Tämä on 262-sivuinen dokumentti, joka voi edelleen paikoin toimia hyvänä referenssinä ja antaa hyvää taustaymmärrystä järjestelmän toiminnasta. |

# Heppa, Heila, Mobiiliheppa ja Omatalli – sovelluksissa käytettyjä tekniikoita

**Sovellukset:**

- Heppa: https://heppa.hippos.f
- Omatalli: https://heppa.hippos.fi/omatalli
- Mobiiliheppa: https://heppa.hippos.fi/mobiili
- Heila (jalostus): https://heppa.hippos.fi/jalostus
- Nykyinen alusta: Jelastic (PaaS): https://jelastic.com/
- https://jelastic.cloud/
- Tietokanta: nykyinen alusta on MariaDb (10.2.11/10.4.13): https://mariadb.org/
    - Yksisuuntainen 'Master –> Slave' replikointi

**Heppa tekniikat**

- Backend: Java(Jakarta EE/ Java EE / J2EE):
    - Buildattu ja paketoitu standardi Jakarta EE application .ear
    - Toimii WildflyApplication Server 19.1.0.Final / Java 11
- 2 nodea clusterissa:
    - Transaktio cache (EJB) replikointi
    - Hibernate ORM(5.3.15.Final)
- Frontend: Apache Tapestry(4.0.x) web-framework
- Asiakkaan alusta: web-selain(PC)

**Omatalli tekniikat**

- Backend: Java(Jakarta EE/ Java EE / J2EE):

    - Pakattu samaan moduliin kuin (.ear) Heppa
- Frontend: Java Script (React) web-rajakerros
- Asiakasalusta: web-selain (tabletti, älypuhelin, PC)

- Applications: ‘Mobiiliheppa’

**Mobiiliheppa tekniikat**

- Backend: Java(Spring-boot):
    - Toimii erillisessä Jelasticnodessa (Docker)
    - Paketoitu Java 11.jarembedded Dockerimagessa joka perustuu adoptopenjdk/openjdk11:alpine-jre
    - Spring-boot JDBCTemplate/ mysql-connector-java
    - Käyttää Slave DB read-only tilassa
- Frontend: Java Script (React) web-kerros
- API (Swagger-UI)
- Asiakasalusta: web-selain(älypuhelin, tabletti, PC)

**Heila tekniikat**

Backend: **Java**(Spring-boot):

- Oma dedikoitus Jelasticnode (Docker)
- Paketoitu Java 11.jarembedded Dockerimagessa debian:buster
- Spring-boot JDBCTemplate/ mysql-connector-java(+ muut java libs)
- Oma tietokanta metadatalle
- Käyttää Omatallibackend (REST) tunnistautumiseen

- Frontend: Java Script (React) web-kerros
- Asiakasalusta: web-selain (PC, tabletti, älypuhelin)

**Frontend**

- Apache webpalvelin
- oma Jelastic node
- Docker container perustuu Debian Linux
- IP-taulut perustuvat palomuuriin(portít, blokatut IP:t jne)
- Rajoitettu vain HTTPS
- Kuormanjako Wildfly nodeille
- Staattiset tilastot caching

**Integraatiot**

- XML / TEXT / PDF / Image files served / keskitetty käyttö (S)FTP (lähtölistat, tulokset, maalikamerakuvat totokertoimet, DNA-laboratorio, SRL, STT jne.)
- HTTPS (java Servlet): rokotukset
- Postitusintegraatio (eKirje)
- Maksujärjestelmäintegraatio (ropo24)
- Confluence dokumentointi