

1. Vuokraohjelmiston yleiskuvaus

Focusa vuokraohjelmisto on uusinta web-tekniikkaa hyväksi käyttävä web-ohjelmisto. Vuokraohjelmiston rakenne on kevyt ja yksinkertainen. Focusa vuokraohjelmistoa kehitetään jatkuvasti. Se varmistaa ohjelmiston ajantasaisuuden pitkälle tulevaisuuteen.

Vuokraohjelmistolla on, kuten kaikilla Diantin rakentamilla web-ohjelmistoilla, standardien mukaiset rajapinnat, joilla ohjelmisto voidaan kytkeä muihin ohjelmistoihin. Tässä suhteessa web-palvelut ([Web service](#)) ja [mikropalvelut](#) ovat erityisen mielenkiintoisia. Aihetta on käsitelty yleisesti kohdassa 2. (Ohjelmiston arkkitehtuuri) ja tarkemmin kohdassa 8. (Tulevaisuus).

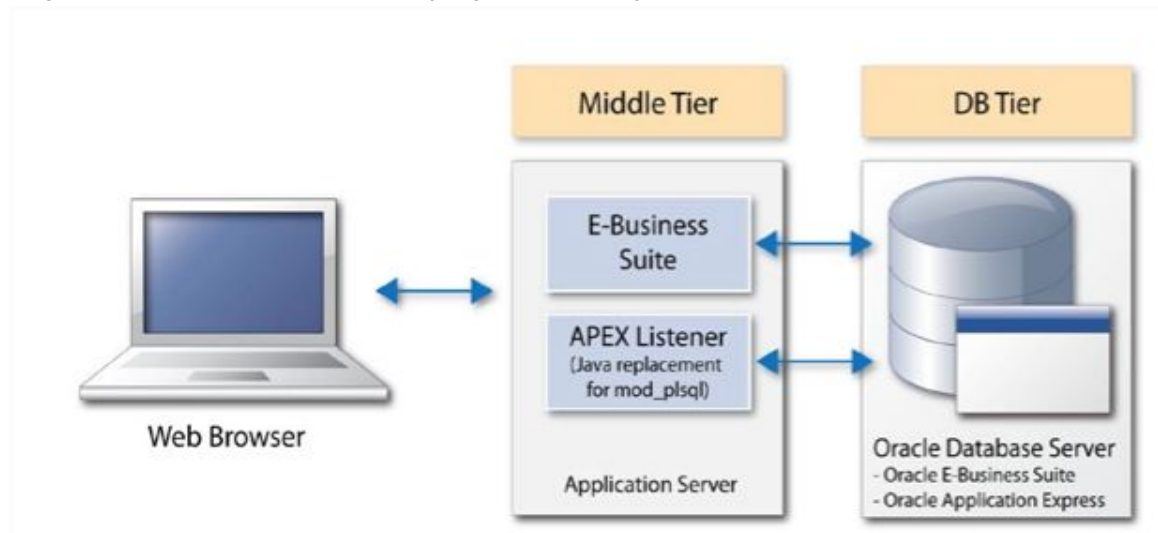
Kuvauksessa käsitelty ohjelmiston tekniikkaa, sisältöä, toimintaa ja tulevaisuutta -- tässä järjestyksessä. Dokumenttiin on lisätty runsaasti linkkejä alan kirjallisuuteen. Dianti on kiitollinen kaikesta palautteesta ja pyrimme ottamaan sen huomioon kuvauksen uusissa päivityksissä.

2. Ohjelmiston arkkitehtuuri

Kaikki Focusa ohjelmistot ovat web-ohjelmistoja. Sovelluksessa on kolme kerrosta (kuva 1): Web-selain (Web Browser), sovelluspalvelin (Middle Tier, Application server) ja tietokanta (DB Tier). Sovelluspalvelin ja tietokanta voivat olla omassa organisaatiossa (jopa samalla koneella) tai pilvessä internetin takana.

Web-selain

Ohjelmiston operointia varten käyttäjä tarvitsee työasemassa tai mobiililaitteessa olevan,



Kuva 1 Focusa web-sovelluksessa sovelluspalvelimella on Apex kuuntelija (Apex listener) (tai pois käytöstä jäävä mod_plsql), joka hoitaa yhteydenpidon selaimen ja Oracle tietokannan välillä. Sovelluspalvelimena voi olla mm. ilmainen Apache Tomcat, Oraclen WebLogic tai Oraclen Glassfish. E-Business suite on joukko Oraclen Javalla rakentamia sovelluksia, kuten CRM (asiakasohjelmisto).

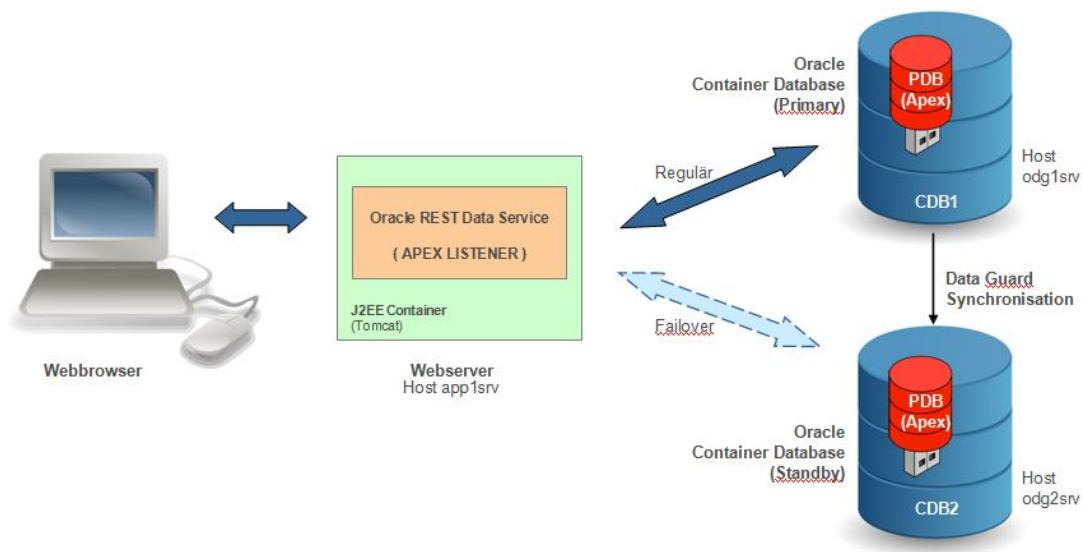
JavaScript tulkilla varustetun selaimen, jolla on http-protokollaa käyttävä yhteys sovelluspalvelimeen eli normaali internet yhteys.

Sovelluspalvelin (Middle Tier)

Palvelin toimii muuntajana selaimen http-kutsujen ja Oracle tietokannan käyttämän SQL tai plsql kielen välillä. Tulkkina voi olla http-server (mod_plsql-plugi) tai nykyään uudempi Apex kuuntelija (kuten kuvassa). Uusin tulokas on Oraclen suosittelema Oraclen ilmainen ORDS palvelu. Tarkempia kuvauksia löydät Diantin kotisivuilla (www.dianti.fi) ja niillä olevista hyperlinkeistä.

ORDS palvelu (Oracle REST Data Service, kuva 2)

Kehittyneempi ja Oraclen suosittelema ratkaisu kuvassa 1 esitetylle Middle Tier kerrokselle on Oraclen [ORDS](#) palvelu, joka avaa uusia portteja Focusa web-ohjelmistoille (kuva 2)



Kuva 2. Ords (Oracle REST Data Service) Apex kuuntelijana (Apex Listener). Ratkaisu mahdollistaa REST arkkitehtuurin mukaisen rajapinnan. Kaaviossa on kaksi tietokantaa, joista toinen on varatietokanta häiriötilanteessa (Failover).

Se mahdollistaa mm. [web.palveluiden](#) ja [mikropalveluiden](#) liittämisen [REST](#) arkkitehtuurin mukaisesti osaksi sovellusta. Diantilla on hyvät valmiudet nimenomaan uusien, kevyempien mikropalveluiden toteuttamiseen. Mikropalveluilla voidaan rakentaa helposti laajennettavia tietojärjestelmiä, joiden komponentit viestivät REST rajapintojen kautta.

Mikropalveluista on kerrottu enemmän kohdassa 8. Vuokraohjelmiston tulevaisuus. Tietoa ja linkkejä palveluista löydät myös Diantin [kotisivuilta](#). Syvempää tietoa kaipaavalle Oracle on julkaissut [PDF-dokumentin](#) REST arkkitehtuurin toteutuksesta Oraclen WebLogic sovelluspalvelimella.

Tietokanta (DB Tier)

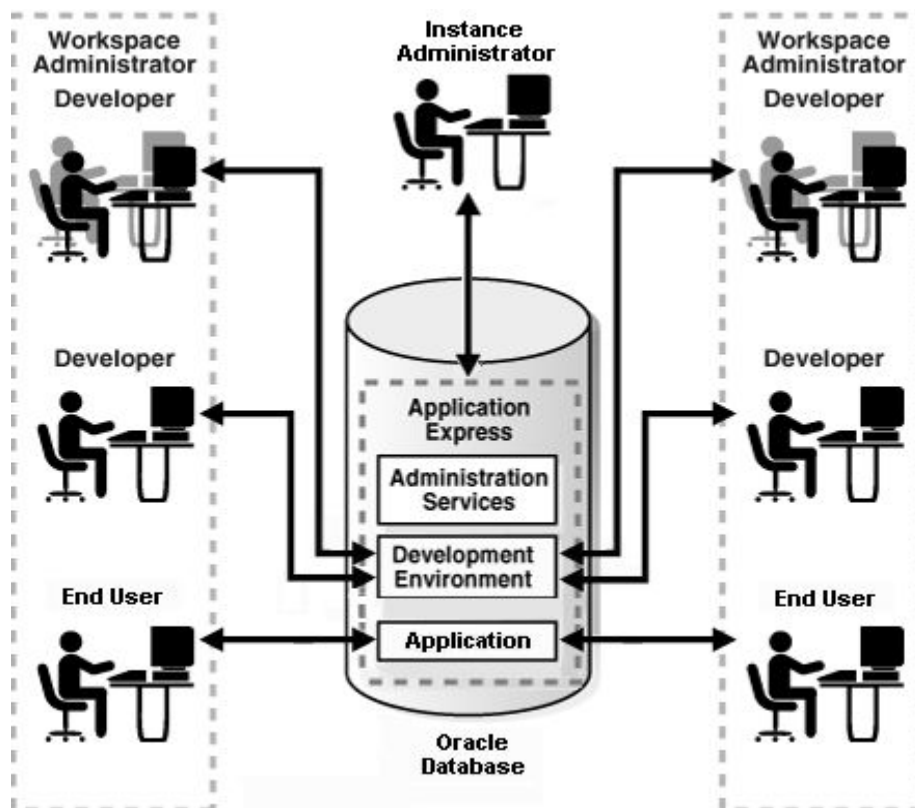
Oracle tietokannassa on vuokraohjelmiston data, näkymät, eri tarkoituksiin tehdyt tietokanta-ohjelmat, paketit (package) sekä osa itse vuokraohjelmistoa (Kuva 3).

Apexissä on myös kehitinosa, jolla Dianti rakentaa kaikki ohjelmistonsa.

Oracle tietokannan suojausten sisällä on myös [Oracle Application Express \(Apex\)](#) kehitinohjelmisto, jonka ajoaikaisessa osassa pääosa vuokraohjelmistoa sijaitsee.

Alla on kaavio (Kuva 3) Oracle tietokannassa olevasta Apex (Application Express) kehitinohjelmiston instanssista eli ilmentymästä. Instanssin hoitaja (Instance Administrator) on kuvassa ylhäällä keskellä. Kaaviossa on kaksi Apexin työaluetta (Workspace), joilla on omat kehittäjät (Developer) ja käyttäjät (End user).

Kuvasta selviää myös, että käyttäjät operoivat vuokraussovelluksella (Application) ja ohjelmiston kehittäjät (Developer) kehitysalueella (Development Environment).



Kuva 3. Application Express (Apex) kehittimen kehitysalue (Development Environment) ja sovellus (Application) sijaitsevat Oracle tietokannan suojausjärjestelmien sisällä. Se lisää web-sovelluksen turvallisuutta ja nopeutta. Käyttäjät (End user), kehittäjät (Developer) ja hallinnoijat (Administrator) operoivat omilla erillisillä alueillaan.

3. Ohjelmiston vaatimukset

Focusa vuokraohjelmiston ympäristövaatimukset ovat suhteellisen helposti toteutettavissa. Alla esitetään ohjelmiston selaimelle, sovelluspalvelimelle ja tietokannalle asettamat vaatimukset.

Selain voi olla mikä tahansa yleisessä käytössä oleva selain, jossa on JavaScript ja HTML5 kielet sekä CSS3 tyyliohjeet. Ne ovatkin tällä hetkellä vakiovarusteena lähes kaikissa selaimissa. Dianti suosittelee käytettäväksi Chrome, Firefox tai Internet Explorer selaimia.

Sovelluspalvelimena voi olla Oraclen Weblogic, Oraclen Glassfish tai ilmainen Apache Tomcat sovelluspalvelin. Alustana voi olla esimerkiksi Unix palvelin tai Windows Server.

Tietokannan on oltava Oracle 10g tai sitä uudempi. **Oracle Application Developer** kehittäjän versio täytyy olla 5.0 tai uudempi.

4. Vuokraohjelmiston vasteajat

Vuokraohjelmiston virhe- ja ongelmatilanteiden korjausten vasteajat on selvitetty asiakaskohtaisissa sopimuksissa Focusa vuokraohjelmiston ylläpidosta. Focusa ohjelmiston toiminnallinen vasteaika, joka kuvaa ohjelmiston nopeutta vastata käyttäjän käynnistämiin tapahtumiin, riippuu useista tekijöistä laiteympäristössä ja tietoliikenneyhteyksissä eikä sitä käsitellä tässä.

Toiminnalliset vasteajat ovat kuitenkin sovelluksen keveydestä johtuen muihin vastaaviin ohjelmistoihin verrattuna erinomaiset.

5. Rajapinnat muihin ohjelmistoihin

Yleisesti käytettyjen standardien mukaiset rajapinnat muihin ohjelmistoihin on kaikkien Focusa web-ohjelmistojen eräs suurimpia vahvuuksia. Kohdassa 2 mainittiin Oraclen kehittämän uuden ORDS (REST) palvelun rajapinnat (Kuva 2). Ne avaavat tien kokonaisten tietojärjestelmien luomiseen pienistä itsenäisistä, REST rajapinnan omaavista sovelluksista LEGO palikoiden tapaan.

Mikropalveluista odotetaan paljon tällä saralla. Diantilla on tieto-taito niiden rakentamiseen.

Teklan Trimble Locus tietokanta

Tällä hetkellä tietojen haku Trimble Locus tietokannasta vuokraohjelmistoon tapahtuu Diantin valmistamien ns. näkymien kautta, jotka tässä tapauksessa ovat kyselyyn soveltuvia "virtuaalisia tauluja". Käyttäjä ei pysty niiden kautta päivittämään, lisäämään tai poistamaan tietoja. Reaaliaikaiset näkymät ovat toimineet hyvin.

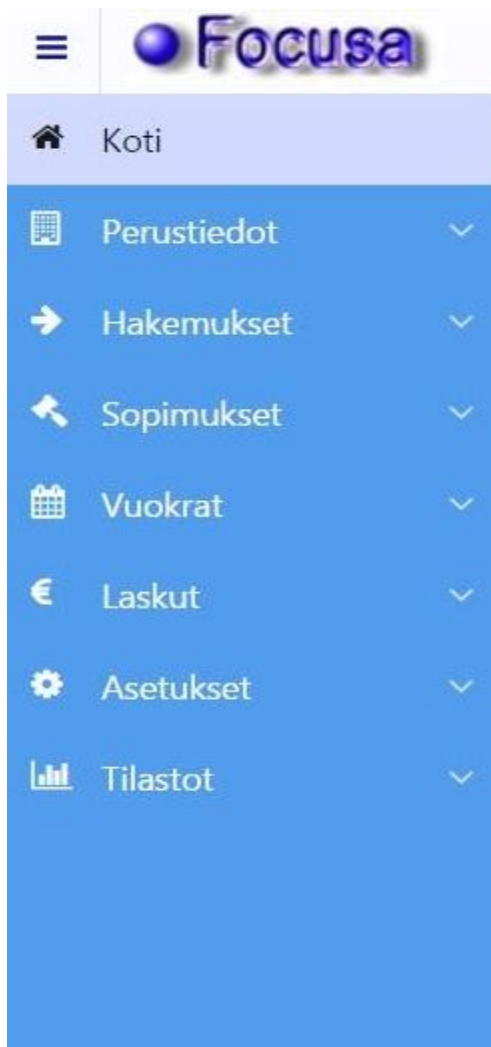
Toinen "rajapinta" Trimble Locus tietokantaan on Diantin rakentama suojattu taulu, johon Focusa vuokraohjelmisto tallentaa vuokrasopimukseen liittyvää Trimble Locus tietokannan mahdollisesti tarvitsemaa tietoa, ja josta Trimble Locuksella on oikeus hakea tiedot.

CGI:n talousohjelmisto Rondo

Tietojen siirto CGI:n Rondo talousohjelmistoon tapahtuu Focusa vuokraohjelmiston generoiman määrämuotoisen tiedoston avulla, josta CGI tulostaa vuokralaisille

lähetettävät laskut. Vuokraohjelmistossa on myös valmius vastaavien laskujen tulostamiseen.

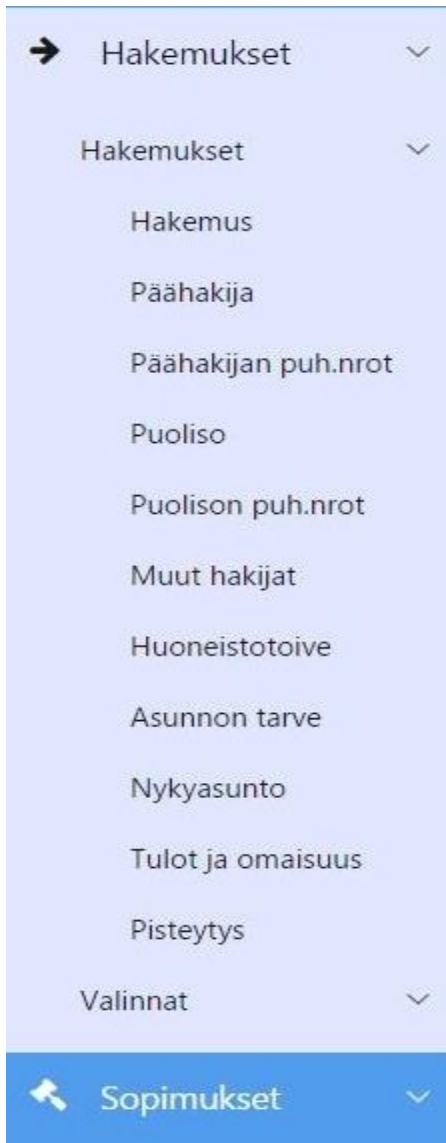
6. Vuokraohjelmiston sisältö



Focusan pääkansiohakemisto:
Seuraavissa kuvissa esitetään järjestyksessä kunkin pääkansion alikansiot. Kunkin kuvan alaosassa on viittaus seuraavaan pääkansiokuvaan. Asetukset ja Tilastot kansioiden alikansioita ole avattu tässä.

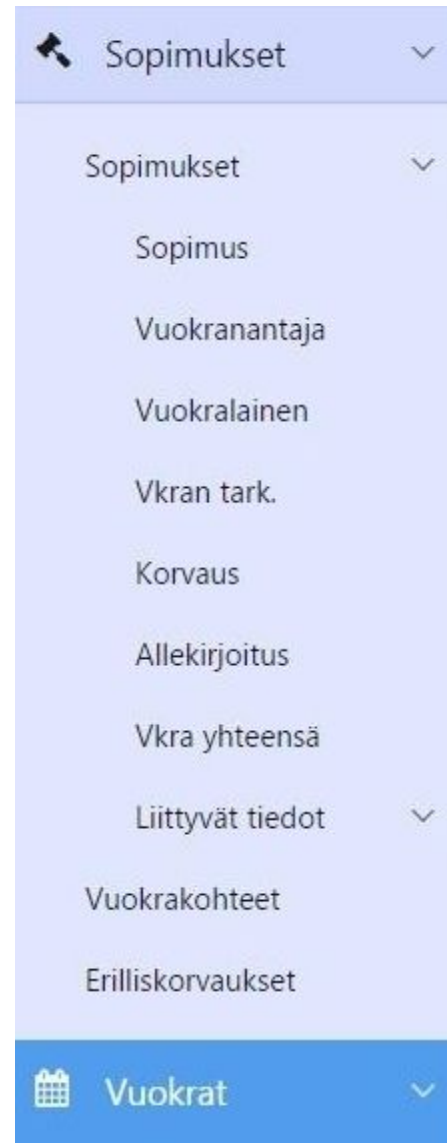


Perustiedot: Tiedot kiinteistöistä ja osapuolista.
Osapuoli: Vuokranantajan ja vuokralaisen tiedot.
TL = Trimble Locus tietokanta
"Hakemukset" alaosassa viittaa seuraavaksi esitettävään pääkansioon.



Hakemukset: Asunnon tai maa-alueen vuokrauksen hakijatiedot ja niiden käsittely. Hakemukset alikansiossa on mahdollisuus hakijoiden neuvoa antavaan vuokraohteen tarvepisteytykseen

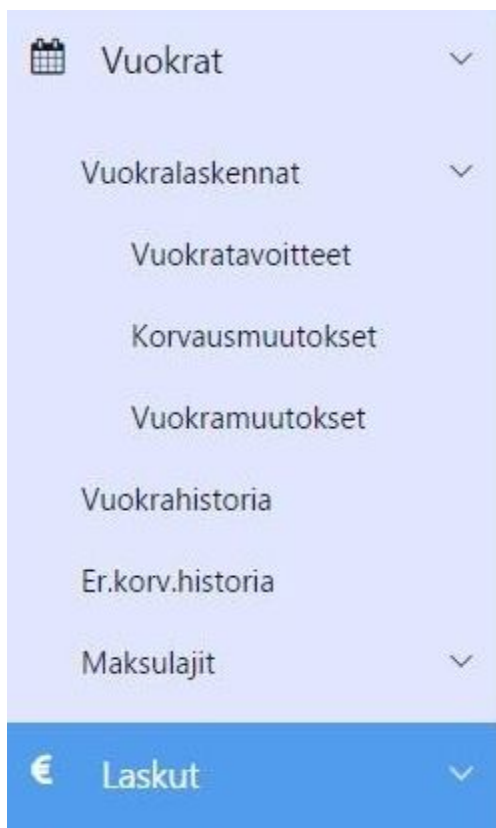
Valinnat-kansiossa on valintakoukset ja vuokraohteiden ja hakijoiden yhdistämiset, joiden pohjalta tapahtuu vuokrasopimusten automaattinen luonti.



Sopimukset: Kansio sisältää vuokrauksen keskeiset tiedot. Vuokrantarkistus voi tapahtua mm. elinkustannusindeksin mukaisesti.

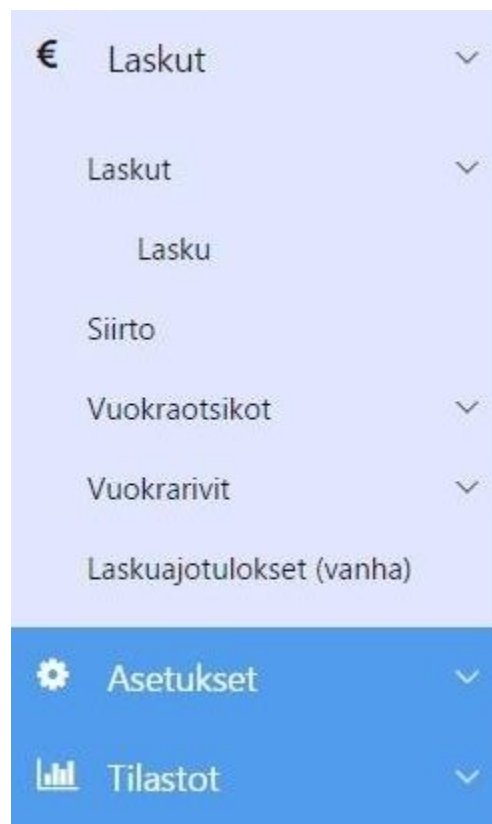
Vuokrakohteita voivat olla rakennukset, huoneistot, solut ja vuokra-alueet.

Sopimukseen voidaan liittää erilliskorvauksia, kuten siivouspalveluja. Sopimukseen voi sisältyä tilapäisesti muuttuneita vuokria.



Vuokrat: Vuokrien laskenta tapahtuu Vuokratavoitteet-ohjelman avulla. Vuokrassa voi olla monia siihen vaikuttavia parametrejä, kuten vuosi- tai kuukausivuokra.

Vuokria tai erilliskorvauksia (esim. siivouskulut) voi muuttaa monin tavoin. Muutetut vuokrat tallennetaan vuokrahistoriaksi



Laskut: Monipuolisin ehdoin valitulle vuokrasopimusjoukolla voidaan laskea vuokrat yhdellä Vuokratavoitteet-ohjelman avulla.

Vuokrat voidaan tulostaa suoraan Focusasta tai siirtää johonkin talousohjelmistoon tulostettaviksi -- Salossa CGI:n Rondo talousohjelmistoon.

7. Vuokrausohjelmiston toimintaperiaate

Kohtaa 7. täydennetään kuvilla ja kaavioilla.

Focusa vuokrausohjelmiston toiminnot voidaan jakaa 1) hakuihin + analytiikkaan ja 2) tallennuksiin.

Haut ja analytiikka

Focusa vuokrausohjelmiston hakuraportit hakevat tietokannasta joukon tietueita käyttäjän antamien hakuehtojen mukaisesti. Hakuraportit tunnistaa nimen monikkomuodosta, esimerkiksi Sopimukset on hakuraportti, jolla haetaan vuokrasopimuksia. Raportin hakutyökalut ovat monipuolisia ja niiden toiminnoissa avautuu jatkuvasti uusia ulottuvuuksia kokeneellekin käyttäjälle.

Esimerkinä yksinkertainen hakuehto: Käyttäjän halutessa hakea Kalle Tuiskusen

vuokrasopimuksen hän kirjoittaa raportin hakukenttään "tuis", jolloin kaikki ne tietueet, joissa on tämä merkkijono jossakin sarakkeessa, tulevat hakutulokseen. Spesifisempiä hakuehtoja ovat mm. suodattimet ja sarakeotsakkeet. Seikkaperäisen ohjeen hakumahdollisuudesta saa verkossa valmisteilla olevista vuokraohjelmiston ohjeista, jotka ovat aina ajantasaiset.

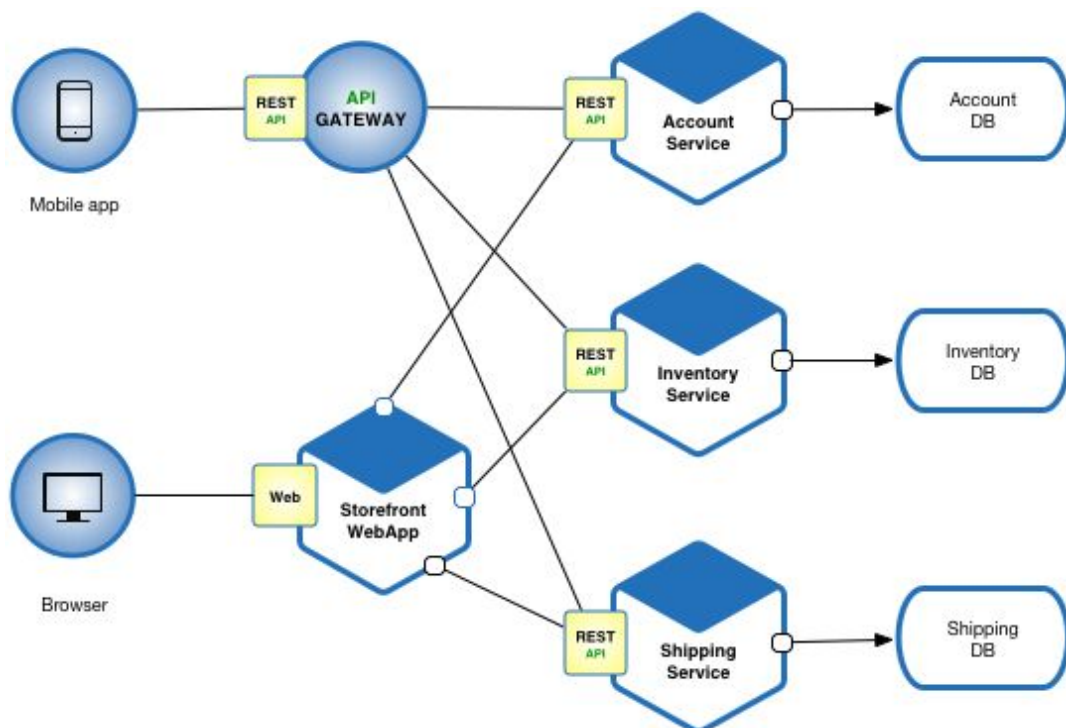
Hakuraportin hakuja voi jalostaa monilla työkaluilla. Raportin rivejä voi ryhmitellä käyttäjän antamalla parametreillä, niistä on helppo laatia hakutulokseen perustuvia diagrammeja ja yhteenvetoja suoraan myös selaimella. Esimerkiksi kaupungin asuntojen jakautumisesta pinta-alojen tai kaupunginosien mukaan voisi laatia piirakkakuvion -- ja monia muita. Lopuksi raportin voi lähettää sähköpostilla halutuille henkilöille.

RaporttiaHakuraporteista voidaan ajaa monipuolisia paperitulosteita, joissa on mm. näyttäviä kaavioita.

Tallennukset

Uusien tietueiden tallennukset, päivitykset ja poistot tapahtuvat tallennuslomakkeilla, jotka avautuvat hakuraporteista napauttamalla kynä-ikonia. Tallennuslomakkeiden nimet ovat yksikössä, esimerkiksi Vuokrasopimus.

8. Vuokraohjelmiston tulevaisuus



Kuva 4. Yksinkertainen verkkokauppasovellus, jossa neljä mikropalvelua viestii REST rajapinnoilla. Mobiililaitte on liitetty sovellukseen yhdyskäytävän (Gateway) ja REST rajapinnan avulla. Selain on yhteydessä verkkokaupan esittelyohjelmistoon web-palvelimen kautta.

Mikropalvelut (Kuva 4)

Dianti Oy:n kehitysstrategia perustuu useiden asiantuntijoiden tukemaan ennusteeseen, jonka mukaan sovelluskehitys nojaa yhä enemmän jo edellä mainittuihin [mikropalveluihin](#).

Sen mukaan suuretkin sovellukset rakennetaan pienistä yhteisesti sovituilla rajapinnoilla toisiinsa liimatuista ohjelmistoista, jotka yhdessä muodostavat sovelluksen (kuva 4).

Mikropalveluiden esiinmarssia on helpottanut suuresti ns. kontti-käsite (Container), joka "käärii" mikropalvelun paketiksi, jossa on kaikki tarvittava käyttöjärjestelmästä alkaen sovelluksen itsenäistä toimintaa varten. Konttia, jossa [Docker](#) on vahvin toimittaja, voi verrata mustaan laatikkoon, josta ulospäin näkyy vain rajapinta (esim. REST = REpresentational State Transfer). Kontit ovat täysin itsenäisiä ja kontti voi toimia omana sovelluksena.

Mikropalveluista on teoriassa helppo luoda kokonaisia tietojärjestelmiä. Se pitääkin paikkansa kun "konttien" lukumäärä on pieni. Suurten satoja tai tuhansia mikropalveluja sisältävien järjestelmien kehittäminen voi olla hyvinkin haasteellista. Mikropalvelujen hyvien puolien katsotaan voittavan kuitenkin siihen sisältyvät heikkoudet. Sillä tehdyt tietojärjestelmät ovat skaalautuvia, helposti laajennettavia ja joustavia.

Oracle kuten kaikki suuret IT talot panostaa voimakkaasti mikropalveluihin ja [kontteihin](#) pilvessä. [Tässä](#) linkki kattavaan diaesitykseen Oraclen mikropalveluista ja konteista. Oraclen ja usean muun toimittajan mukaan pilvi tai oman organisaation yhteinen palvelin on mikropalveluiden välttämätön kehitysalusta.

Tämän ymmärtää helposti, kun ajattelee uutta mikropalveluihin liittyvää DevOps kehitysarkkitehtuuria, jossa organisaation kaikki kehitettävään sovellukseen liittyvät tahot osallistuvat omalla panoksellaan kehitystyöhön. Kehitysympäristön täytyy olla reaaliaikaisesti kaikkien saavutettavissa. Jos kehittäjät ovat maantieteellisesti eri paikoissa esimerkiksi yksityispilvi on hyvä ratkaisu.

REST vai SOAP rajapinta

Edellä viitattiin useissa kohdissa REST rajapinnan käyttöön hajautetuissa järjestelmissä (SOA = Service-oriented Architecture). Muitakin arkkitehtuureja on, vaikka tällä hetkellä REST rajapinta on uusissa projekteissa eniten käytetty. Monet suuret yhtiöt (Netflix, Google, Amazon) ovat valinneet REST pohjaisen ratkaisun.

Hieman vanhempi SOAP (Simple Object Access Protocol) on edelleen laajassa käytössä. Se on monissa, varsinkin ehdotonta turvallisuutta vaativissa tilanteissa sopiva arkkitehtuuri. SOAP:in varjopuolena on sen raskaus, monimutkaisuus ja sitoutuminen XML formaattiin.

Useat ohjelmointiyrietykset valitsevat REST (RESTful) arkkitehtuurin, jossa viestityksen formaatti on JSON tai XML. JSON (JavaScript Object Notation) pohjautuu selaimissa yleisesti käytettyyn JavaScript kieleen. JSON tiedostot ovat ihmiselle ymmärrettävässä muodossa. REST'issä käytetyt komennot ovat tuttuja http-protokolla käskyjä kuten GET, POST jne. REST'in turvallisuus on kiinni internet-yhteydestä, jota voidaan parantaa

monin tavoin kryptauksella, autentikoinnilla jne.

Olemme Diantissa valinneet REST arkkitehtuurin edellä kerrotuista syistä. REST ratkaisussa on, -- vähän töitä tehden mahdollista lukea myös SOAP viestejä (XML) . Dianti haluaa olla edelläkävijä kohdissa 1 ja 8 (Mikropalvelut) kuvatuissa mikropalveluissa.

IoT

Kokonaan oman lukunsa mikropalvelujen yhteydessä ansaitsisi Esineiden internet ([IoT](#)), jossa Dianti on mukana eräiden asiakkaiden projekteissa. Diantin [kotisivuilla](#) tullaan käsittelemään aihetta tiiviisti.