DigiBiogasHubs

Digitaaliset alustat joustavan ja skaalautuvan biokaasutoiminnan mahdollistajina

Digitaalisen yhteistyöalustan kehitys ja pilotointi

Jyri Mäkelä

Risto Hietala

27.11.2025
Ryhmähanke R-00483

Euroopan unionin osarahoittama

Sisällys

1 Johdanto

DigiBiogasHubs-projektin työpaketissa neljä tehtiin itse alustan toteutus. Alustaa kehitettiin työpaketissa kolme tehdyn vaatimusmäärittelyn pohjalta. Alustan kehityksessä pyrittiin ottamaan huomioon projektin jälkeisen kehityksen helppous. Alustassa käytetyt teknologiat valittiin sellaisiksi, että ne ovat mahdollisimman yleisesti käytössä olevia ja helppoja käyttää. Teknologioiden tulee kuitenkin tukea mahdollisimman laaja-alaista kehitystä. Alusta on pääosin rakennettu JavaScript-ohjelmointikieleen pohjautuvilla työkaluilla.

Alusta on saatavilla avoimena lähdekoodina GitHubista osoitteesta https://github.com/CentriaUniversityOfAppliedSciences/DigiBioGasHub

2 Alustan rakenne

Alusta koostuu viidestä eri osasta. Nämä ovat käyttöliittymä (frontend), palvelin (backend), avoin rajapinta, verkostointipalvelin ja analytiikkatyökalu. Näiden lisäksi tiedon tallennukseen on kolme eri tietokantaa. Palvelin ja analytiikkatyökalu on kontitettu käyttäen Docker-kontitusalustaa. Alustalla käytettiin i18n lokalisaatio kirjastoa, jonka avulla on helppoa luoda eri kieliversioita alustasta. Projektin aikana tehtiin suomen ja englannin kielen kieliversiot.

2.1 Käyttöliittymä

Käyttöliittymä rakennettiin käyttäen Vue framework:ia. Vue on JavaScript-pohjainen työkalu käyttöliittymien rakentamiseen. JavaScript:iin pohjautuminen helpottaa alustan julkaisemista avoimena lähdekoodina, sillä JavaScript on yleisesti käytetty ohjelmointikieli. Vue:n tukena käyttöliittymän rakennuksessa käytettiin Ionic framework nimistä työkalua. Ionic:n avulla voidaan käyttöliittymästä tuottaa yhtä aikaa web-pohjainen käyttöliittymä sekä Ios ja Android sovellukset.

2.2 Palvelin

Käyttöliittymän apuna toimii palvelin (backend), joka hoitaa käyttöliittymän yhteyttä tietokantoihin. Palvelin toteutettiin NodeJS-alustalla. NodeJS on JavaScriptiin pohjautuva suoritusympäristö. NodeJS:n rinnalla ajettiin ExpressJS web framework:ia. ExpressJS:n avulla rakennettiin palvelimen osoitteitus. Palvelin hoitaa suurimman osan alustan tietoturvasta, siellä tarkistetaan esimerkiksi, että käyttäjillä on tarvittavat oikeudet suorittaa komentoja. Palvelimen ja PostgreSQL-tietokannan välistä yhteyttä hoitamaan valittiin Sequelize niminen NodeJS-kirjasto. Tämän kirjaston avulla tietokannan rakenne voitiin luoda malleina (models) ja kirjasto myös loi ja ylläpiti tietokannan tauluja.

2.3 Avoin rajapinta

Avoin rajapinta on toteutettu REST-rajapintana. Siellä voi lähettää ja vastaanottaa alustaan liittyvää dataa käyttäen käyttöliittymässä yritykselle luotua rajapinta-avainta. Pilottivaiheessa rajapintaa testattiin lähettämällä sinne arvoja RaspBerry Pi-laitteeseen kytketystä sensorista.

2.4 Verkostointipalvelin

Verkostointipalvelin on NodeJS-alustalla toimiva keskustelupalvelu. NodeJS:n Socket-kirjaston avulla luotiin reaaliaikainen keskustelupalvelu, joka tallentaa viestit MongoDB-tietokantaan. Käyttöliittymä kytkettiin verkostointipalvelimeen, jolloin käyttöliittymässä voitiin käydä reaaliaikaista tekstipohjaista keskustelua käyttäjien välillä.

2.5 Analytiikkatyökalu

Analytiikkaan käytettiin CubeJS (https://github.com/cube-js/cube) semantic layer:iä. CubeJS:n avulla on helppo muodostaa analytiikkaan tarvittavia hakuja tietokannasta ja sen tuloksista on helppo muodostaa kuvaajia. Analytiikasta voidaan muodostaa myös erilaisia raportteja tekstimuodossa käyttöliittymään tai ladattavina tiedostoina.

2.6 Tietokannat

Tietokantoina käytettiin PostgreSQL-tietokantaa yleiseen tiedon tallennukseen, MongoDB-tietokantaa verkostointipalvelimen tietokantana sekä MinIO-tietokantaa tiedostojen säilytykseen. PostgreSQL-valittiin käytettäväksi, koska sillä on helppo toteuttaa palvelun skaalausta. Siihen on sisäänrakennettuna klusterointi eli kuormituksen kasvaessa liikennettä voidaan jakaa usealle tietokantaprosessille. PostgreSQL soveltuu mainiosti myös aikapohjaisen tiedon tallennukseen eli sillä on tehokasta käsitellä esimerkiksi sensoridatan tallennustiedon hakua aikaleiman perusteella.

MinIO auttaa järjestelmää käsittelemään sinne ladattavia tiedostoja. Mikäli tiedostot tallennettaisiin normaaliin tietokantaan, niin on olemassa vaara, että niitä ei enää saataisi palautettua alkuperäiseen muotoon. MinIO säilyttää tiedoston alkuperäisenä, joten tätä vaaraa ei ole ja tietokannalla voi muodostaa latauslinkkejä tiedostoihin, jolloin suurten tiedoston käsittely alustalla on helpompaa.

MongoDB-tietokanta on ei-relationaalinen eli NoSQL-tietokanta ja se soveltuu erinomaisesti alustalle rakennetun keskustelupalvelun tietokannaksi. Etuina ovat esimerkiksi suuri yhdenaikaisten yhteyksien määrä ja tietorakenteen suora JavaScript-tuki.

3 Pilotointi

Alusta oli julkisesti testattavissa osoitteessa biokaasuhubi.fi ja tämän lisäksi alustalla oli Azure-palvelimella testiversio, jossa testattiin uusia ominaisuuksia pienemmällä testaajamäärällä. Alustan ominaisuuksista pidettiin työpajoja sidosryhmille, joissa he saattoivat testata alustaa ja antaa palautetta. Työpajoissa tuli ilmi vielä joitakin alustalta puuttuvia ominaisuuksia, joita ei olla vaatimusmäärittelyssä otettu huomioon. Tälläisiä huomioita oli esimerkiksi kuljetuksissa mahdollisesti tarvittavien lupien lisääminen tarjottaviin tuotteisiin sekä jätteitä vastaanottavien kunnallisten toimijoiden käyttämä porttimaksu, jossa jätteiden tuoja maksaa jätteiden hävityksestä.

Alustan pilottiversio rakennettiin Tieteen tietotekniikan keskuksen (CSC) ylläpitämälle Linux-pohjaiselle palvelimelle. Linux-pohjainen palvelin tarjosi helpon tavan rakentaa palvelimen rakenteet. Reititystä tukemaan palvelimelle asennettiin Nginx http-palvelin ohjelmisto, jolla voidaan hoitaa reitit ykset alustan eri osiin sekä tarvittaessa lisätä kuormituksen tasausta (load balancing).

Pilotoinnin liiketoimintamalliksi valittiin tilauspohjainen malli. Alustan perustoimintoja voi käyttää rekisteröitymällä alustalle ilmaiseksi, mutta alustalla on myös edistyneempiä toimintoja, joiden käyttöön tarvitaan maksettu tilaus. Tätä varten alustaan tehtiin kevyt integraatio Stripe-maksualustan kanssa. Pilottialustassa maksavan käyttäjän toiminnot olivat pääasiassa erilaisia filttereitä tarjouksien haussa, mutta järjestelmä tukee myös muiden osa-alueiden lisäämistä tilauspalvelun alle.

Alusta suunniteltiin niin, että ylläpitäjä voi käyttöjärjestelmän kautta lisätä helposti uusia myyntikategorioita ja -materiaaleja.

4 Alustan tilanne projektin päättyessä

Alusta julkaistiin avoimena lähdekoodina projektin loppuvaiheessa GitHub-alustalla. Alusta on saatavilla MIT-lisenssin alla, mikä mahdollistaa myös kaupallisen toiminnan alustaa käyttäen. Alustan päätoiminnallisuudet ovat valmiina, mutta siitä jäi vielä muutamia osia pois ajanpuutteen takia. Näitä olivat esimerkiksi hakutoiminnot ja sopimuksien muodostaminen. Alusta on rakennettu pohjaksi, jota tulevat toimijat voivat käyttää rakentaessaan omaan liiketoimintaansa sopivan järjestelmän.

Kehitystarpeina ovat ainakin seuraavat:

- Hakutoiminnot
- Sopimuspohjat
- Logistiikkayritysten kuljetusalueet ja hinnat
- Logistiikkaan tarvittavien lupien kartoitus ja lisäys alustaan (ainakin jätekuljetukset)
- Negatiivisen hinnan tarjous (esim. jätteiden vastaanotossa pitää biomassan tuojan maksaa porttimaksu)
- Koordinaatit kartalta kohdan lisäys tarjouksiin. Tällä hetkellä koordinaatit katsotaan tarjouksen osoitteesta