

# Avoimen lähdekoodin sovelluksien hyödyntäminen verkkopalvelussa

Case World Water Hub

Tiia Rantanen

Opinnäytetyö  
Elokuu 2014

Mediatekniikan koulutusohjelma  
Tekniikan ja liikenteen ala



JYVÄSKYLÄN AMMATTIKORKEAKOULU  
JAMK UNIVERSITY OF APPLIED SCIENCES



Tekijä(t) Rantanen, Tiia	Julkaisun laji Opinnäytetyö	Päivämäärä 01.08.2014
	Sivumäärä 63 + 1	Julkaisun kieli Suomi
		Verkojulkaisulupa myönnetty: ( X )
Työn nimi <b>Avoimen lähdekoodin sovelluksien hyödyntäminen verkkopalvelussa</b> Case World Water Hub		
Koulutusohjelma Mediatekniikan koulutusohjelma		
Työn ohjaaja(t) Niemi, Kari		
Toimeksiantaja(t) Trimedia Oy, Saksola Arto		
Tiivistelmä <p>Opinnäytetyön toimeksiantajana toimi jyväskyläläinen ICT-alan palveluita tarjoava yritys Trimedia Oy. Toimeksiantaja halusi saada tietoa nykyaikaisista sovelluskehitykseen soveltuvista sisällönhallintajärjestelmistä. Toimeksiantajan tilaajalle toteutettiin opinnäytetyön pohjalta verkkopalvelu World Water Hub, jossa tarjotaan tietoa järvien ja jokien veden puhtaudesta ja lämpötilasta.</p> <p>Opinnäytetyössä tutkittiin sisällönhallintajärjestelmiä ja alustoja sekä niiden mahdollisuuksia toimia sovelluskehitysalustana. Järjestelmiä vertailtiin ja pisteytettiin eri kriteerien pohjalta ja lopullisena tuloksena valittiin järjestelmä ja toteutettiin verkkopalvelu kyseisellä alustalla.</p> <p>Vertailussa olivat mukana Craft, Keystone.js, Statamic, October, WordPress ja Concrete5. Vertailun mittareina käytettiin hintaa, kehittäjäverkostoa, ohjeita ja tukipalveluita, jatkokehitystä, vaatimuksia ja käytettyjä tekniikoita, hakukoneoptimointia, käytettävyyttä, suorituskykyä ja nopeutta sekä tietoturva.</p> <p>Kehitysympäristöjen sijoittuminen eniten pisteitä saaneesta vähiten pisteitä saaneeseen oli Craft, Statamic, WordPress, Concrete5, October ja Keystone.js. Valittujen vertailukriteereiden pohjalta Craft ja Statamic sijoittuivat lähes ylivoimaisesti muiden edelle. Lopulta kuitenkin projektin resurssien rajallisuuden ja muutoskestävyyden vuoksi toteutuslueksi valittiin WordPress.</p>		
Avainsanat ( <a href="#">asiasanat</a> ) Sisällönhallinta, sovelluskehitys, verkkopalvelut, WordPress, Statamic, Craft, October, KeystoneJS, Concrete5		
Muut tiedot Liite 1: projektin aikajana		



Author(s) Rantanen, Tiia	Type of publication Bachelor's/Master's thesis	Date 01.08.2014
	Number of pages 63 + 1	Language of publication Finnish
		Permission for web publication: ( X )
Title of publication <b>Open Source Software in software development for a web service</b> Case World Water Hub		
Degree programme Media Engineering		
Tutor(s) Niemi, Kari		
Assigned by Trimedia Ltd, Saksola Arto		
Abstract <p>Trimedia Ltd, a Jyväskylä based ICT company, assigned this thesis. World Water Hub, a web service offering information about the quality of river and lake water, needs a new web service and platform. Trimedia gets a good view of the modern content management systems.</p> <p>The Bachelor's Thesis studies different content management systems and their suitability as a platform in software development. The systems were compared based on a certain pre-defined criteria and in the end one system was selected to be the platform for the World Water Hub web service.</p> <p>The comparison included Craft, KeystoneJS, Statamic, October, WordPress and Concrete5. The systems were compared according to pricing, development network, documentation and support, software development, requirements, search engine optimization, usability, performance and security.</p> <p>The systems were ranked from highest to lowest score as follows: Craft, Statamic, WordPress, Concrete5, October and Keystone.js. Even though WordPress was not the highest ranking system it was chosen as the platform due to the limited amount of resources and the changes in requirements of the planned service.</p>		
Keywords/tags ( <a href="#">subjects</a> ) Content management, software development, web service, WordPress, Statamic, Craft, October, KeystoneJS, Concrete5		
Miscellaneous Appendice 1: project timeline		

## Sisältö

Käsitteitä.....	5
1 Opinnäytetyön lähtökohdat.....	10
1.1 Tausta ja toimeksiantaja .....	10
1.2 Tavoitteet .....	10
2 Vaatimusmäärittely ja elinkaari .....	11
2.1 Esiselvitys sivustosta World Water Hub (worldwaterhub.fi) .....	11
2.1.1 Yleistä .....	11
2.1.2 Keskeiset ongelmat aikaisemmassa toteutuksessa .....	11
2.2 Vaatimusmäärittely.....	13
2.2.1 Tavoitteet ja vaatimukset .....	13
2.2.2 Toteutusvälineet ja ympäristö .....	17
2.2.3 Järjestelmäarkkitehtuuri .....	17
2.2.4 Projektin riskit .....	18
2.3 Projektin vaiheistus .....	19
3 Teknologiavertailu.....	20
3.1 Valintaperusteet- ja kriteerit.....	20
3.1.1 Epäsuora hinta .....	20
3.1.2 Kehittäjäverkosto, ohjeet ja tuki.....	21
3.1.3 Jatkokehitys.....	21
3.1.4 Vaatimukset ja tekniikat.....	21
3.1.5 Hakukoneoptimointi .....	22
3.1.6 Käytettävyys .....	22
3.1.7 Suorituskyky ja nopeus .....	22
3.1.8 Tietoturva.....	23
3.2 Testiympäristö.....	23
4 Vertailtavat järjestelmät .....	23
4.1 Yleistä .....	23

4.1.1	Millä perusteella järjestelmät valittiin? .....	23
4.1.2	Järjestelmillä tehdyt toteutukset .....	25
4.2	Concrete5 .....	25
4.2.1	Yleistä .....	25
4.2.2	Ominaisuudet .....	25
4.2.3	Vaatimukset .....	28
4.2.4	Lisäosat .....	28
4.2.5	Jatkokehitys .....	28
4.2.6	Yhteenveto .....	29
4.3	Craft .....	30
4.3.1	Yleistä .....	30
4.3.2	Ominaisuudet .....	30
4.3.3	Vaatimukset .....	33
4.3.4	Lisäosat .....	33
4.3.5	Jatkokehitys .....	34
4.3.6	Yhteenveto .....	35
4.4	KeystoneJS .....	36
4.4.1	Yleistä .....	36
4.4.2	Ominaisuudet .....	36
4.4.3	Vaatimukset .....	37
4.4.4	Lisäosat .....	38
4.4.5	Jatkokehitys .....	38
4.4.6	Yhteenveto .....	39
4.5	October .....	39
4.5.1	Yleistä .....	39
4.5.2	Ominaisuudet .....	40
4.5.3	Vaatimukset .....	41
4.5.4	Lisäosat .....	42
4.5.5	Jatkokehitys .....	42
4.5.6	Yhteenveto .....	43

4.6	Statamic.....	44
4.6.1	Yleistä .....	44
4.6.2	Ominaisuudet.....	44
4.6.3	Vaatimukset .....	46
4.6.4	Lisäosat.....	46
4.6.5	Jatkokehitys.....	46
4.6.6	Yhteenveto .....	47
4.7	WordPress .....	48
4.7.1	Yleistä .....	48
4.7.2	Ominaisuudet.....	48
4.7.3	Vaatimukset .....	52
4.7.4	Lisäosat.....	53
4.7.5	Jatkokehitys.....	53
4.7.6	Yhteenveto .....	54
5	Tulokset ja pohdinta.....	55
5.1	Järjestelmien vertailun yhteenveto .....	55
5.2	Toteutuksesta.....	58
5.3	Pohdinta .....	58
	Lähteet.....	61
	Liitteet .....	64
	Liite 1. Projektin aikajana .....	64

## Kuviot

Kuvio 1.	World Water Hubin alkuperäinen toteutus .....	12
Kuvio 2.	World Water Hubin uusittu etusivu .....	13
Kuvio 3.	Navigaatio avattuna .....	14
Kuvio 4.	Jokilistausnäkyvä.....	14
Kuvio 5.	Karttanäkyvä .....	15
Kuvio 6.	Lisätietonäkyvä .....	15

Kuvio 7. Yksittäinen jokinäkymä.....	16
Kuvio 8. Näkymä veden laatuparametreista .....	16
Kuvio 9. Järjestelmäarkkitehtuuri .....	18
Kuvio 10. Concrete5:n hallintapaneeli, jonka ulkoasu hyödyntää käytöstä poistettua Bootstrap 2.0 -sovelluskehystä .....	26
Kuvio 11. Mukautettua tietosisältöä (page attributes) joki-tyyppisellä sivulla .....	27
Kuvio 12. Reaaliaikainen editointi (in-context editing) .....	27
Kuvio 13. Uuden artikkelin lisääminen. Kuvassa näkyvissä myös Bootstrap 3.0 - sovelluskehukseen pohjautuva käyttöliittymä.....	37
Kuvio 14. Asennus .....	38
Kuvio 15. Hallintapaneeli.....	39
Kuvio 16. Octoberin koodieditori .....	40
Kuvio 17. Julkaisun editointinäkymä, jossa nähdään HTML ja taitettu HTML .....	41
Kuvio 18. Statamicin hallintapaneeli .....	45
Kuvio 19. WordPress WYSIWYG-editori .....	49
Kuvio 20. Hallintapaneeli, jossa näkyvisä valittuna mukautetu tietotyyppi (custom post type) .....	49
Kuvio 21. Mukautettua tietosisältöä (custom post meta data).....	50
Kuvio 22. WordPress Codex eli dokumentaatio kehittäjille.....	51

## Taulukot

Taulukko 1. Osa-alueiden painoarvojen jakautuminen .....	20
Taulukko 2 Tulokset.....	55

## Käsitteitä

### Ajax

Ajax (Asynchronous JavaScript and XML) on sovelluskehityksen tekniikka, jolla voidaan tehdä dynaamisia HTTP-pyyntöjä palvelimelle ilman sivunlatausta. Vaikka Ajaxin nimi viittaa XML-syntaksiin, on nykyään yleisempää käyttää JSON-syntaksia sen keveyden vuoksi. Asynkronisuus viittaa siihen, että pyyntöjä voidaan tehdä enimmillään 4-6 kerrallaan riippuen standardista ja selaimesta. Pyyntöt voidaan tehdä myös synkronisesti, jolloin järjestelmä odottaa Ajax-pyyntöltä vastauksen, ennen kuin se käsittelee uusia pyyntöjä (Treehouse Blog 2014).

### Apache

Apache on Internetin eniten käytetty HTTP-palvelin. Avoimen lähdekoodin Apachella on suuri kehittäjäyhteisö. Elokuussa 2014 Apachea käytettiin 60.2 %:ssa aktiivisista verkkopalveluista (W3 Techs 2014).

### Client-side (front-end)

Client-side tarkoittaa, että tapahtumat suoritetaan käyttäjän laitteella esimerkiksi tietokoneella tai mobiililaitteella. Tässä työssä clientilla viitataan selaimen.

### CSS (cascading style sheets)

CSS on web-taiton muotoiluun käytetty syntaksi. Uusin versio on CSS3, jota tuetaan kaikissa moderneissa selaimissa.

### Debug mode

Debug mode on tila, joka voidaan asettaa päälle, jolloin nähdään kaikki järjestelmässä syntyneet virheet. Monesti samalla käytössä ovat myös virhelokit. Tiloja on yleensä eritasoisia sen mukaan mitä virheitä halutaan seurata.

### Express.js

Express.js on Node.js-alustan suosituin palvelinpuolen MVC-kirjasto. Express tarjoaa kai-



ken perustoiminnallisuuden reitittämisestä tietokantojen hallintaan. Expressin kanssa voidaan käyttää useita eri tietokantatyyppejä kuten MongoDB tai Redis sekä sivupohjamootoreita kuten Jade tai Dust.js.

### **Flat-file CMS**

Sisällönhallintajärjestelmä, joka pohjautuu dynaamiseen kansiorakenteeseen, eli se ei käytä tietokantaa.

### **GitHub**

GitHub on Git -versionhallintaan pohjautuva verkkopalvelu, joka tarjoaa versionhallintaa ja tilastoja projektin aktiivisuudesta. Pääosin GitHubia käytetään projektien versionhallintaa, kehittäjien verkostoitumiseen ja tiedon jakamiseen. Nykyisin monella avoimen lähdekoodin sovelluksella on oma GitHub-tilinsä.

### **HTML (Hypertext Markup Language)**

Merkintäkieli, jolla voidaan määrittää verkkosivuston rakenne. Uusin versio on HTML5, jota tuetaan kaikissa moderneissa selaimissa. HTML-taitto vaikuttaa suuresti hakukoneoptimointiin esimerkiksi Google-hakukoneessa.

### **I18n (internationalization) ja gettext**

I18n tarkoittaa lokalisointi- ja kansainvälisyystoiminnallisuutta kuten kielipaketteja. Gettext on lokalisointijärjestelmä, jolla sisällöstä voidaan tehdä käännöstiedostot.

### **Jade**

Jade on palvelinpuolelle suunnattu Node.js –sivupohjamoottori. Sitä voidaan käyttää myös muissa ympäristöissä.

### **JavaScript client- ja server-side -toteutuksissa**

Single-page application (SPA, joskus myös SPI single-page interface) on verkkosovellus tai -palvelu, jota ajetaan yhdeltä sivulta asynkronisesti. Pääosin data ladataan osissa taustalla tapahtuvilla HTTP-pyyntöillä tarpeen mukaan, yleensä reaktiona käyttäjältä tulleelle komennolle.

## **Laravel PHP Framework**

Laravel PHP Framework on yksi uusimmista MVC PHP -sovelluskehysistä, joka on julkaistu helmikuussa 2012. Joulukuussa 2013 tehdyn tutkimuksen mukaan Laravel oli eniten käytetty PHP -sovelluskehys.

## **LEMP/LAMP-ympäristö**

LEMP on kokoelma avoimen lähdekoodin sovelluksia, jotka yhdessä muodostavat palvelinarkkitehtuurin, jossa voidaan suorittaa verkkopalveluita. LEMP koostuu Linuxista, nginx:sta, MySQL:stä, MariaDB:stä tai MongoDB:stä sekä nykyisin PHP-ohjelmointikielestä. LAMP-ympäristö on samanlainen LEMPin kanssa sillä erolla, että käytössä on Apache-palvelinohjelma.

## **Markdown**

Markdown on työkalu ja syntaksi, jolla voidaan konvertoida tekstiä HTML-muotoon.

## **MEAN-ympäristö**

MEAN koostuu avoimen lähdekoodin sovelluksista ja kirjastoista, joilla voi rakentaa verkkopalveluita. MEAN koostuu MongoDB-tietokannasta, Express.js-verkkopalvelukirjastosta, AngularJS MVC -kirjastosta sekä Node.js-sovellusalustasta.

## **MongoDB**

MongoDB on avoimen lähdekoodin NoSQL-tietokanta, joka pohjautuu dynaamisiin JSON-dokumentteihin. MongoDB-tietokanta koostuu kokoelmista (vrt. taulu) ja kokoelmat dokumenteista (vrt. tietue). MongoDB on skeematon eikä sisällä tukea kokoelmien välisille transaktioille.

## **Mongoose**

Mongoose on MongoDB-tietokantojen skeemanhallintatyökalu. Mongooseella voidaan määrittää erilaisia tietokantarakenteita, joita ei voi muuttaa enää dataa käsiteltäessä tai lisätessä.

### **MVC (model, view, controller)**

MVC on arkkitehtuuri, jossa logiikka jaetaan kolmeen eri alueeseen. Mallit (model) ovat yhteydessä tietokantaan ja tekevät sille abstraktin rajapinnan. Näkymiin (view) kuuluvat käyttöliittymän visuaaliset komponentit sekä datan tulostus. Kontrollerit käsittelevät käyttäjän pyyntöjä ja hakevat tietoja malleista päivittäen sitä näkymille, jotka tulostavat sivun.

### **MySQL**

MySQL on avoimen lähdekoodin relaatiotietokantajärjestelmä.

### **Node.js**

Node.js on Googlen V8 JavaScript-moottorin päälle rakennettu sovelluskehitysalusta, jolla voi toteuttaa skaalautuvia verkkopalveluita, jotka perustuvat asynkronisiin tapahtumiin.

### **NPM**

NPM on Node.js:n mukana asennettava tehokas paketinhallintatyökalu. NPM huolehtii myös sovelluksiin määritetyistä riippuvaisuuksista ja niiden automaattisesti asennuksesta.

### **REST-rajapinta ja RESTful verkkopalvelut**

REST (Representational state transfer) on Internetin järjestelmäarkkitehtuuri. RESTin periaatteita noudattava rajapinta ei vaadi, että client tietää miten rajapinta toimii. Palvelimen tulee tarjota tiedot, joilla client voi keskustella palvelun kanssa. Kaikki tiedonsiirto tapahtuu HTTP-pyyntöillä ja URL -osoitteilla. Lisäys-, muokkaus-, poisto- ja päivitys -toiminnot toteutetaan HTTP-pyyntöillä GET, POST, PUT ja DELETE. Näitä toimintoja voidaan soveltaa resurssien tulostamiseen tai muokkaamiseen palvelimen mukaan.

### **Server-side (back-end)**

Server-side tarkoittaa toiminnallisuutta, joka suoritetaan palvelimella käyttäjän huomaamattomissa.

## **SQLite**

SQLite on tiedostopohjainen relaatiotietokantajärjestelmä, joka ei tarvitse omaa palvelinta.

## **Static-site generaattori**

Generaattori, joka tekee dynaamisista sivuista staattisia sivuja. Suosituin generaattori lienee Jekyll, jossa on mukana samoja kehittäjiä kuin GitHub-versionhallintapalvelussa. Dynaamisten ja staattisten sivujen raja saattaa olla hyvinkin häilyvä.

## **Twig Template Engine**

Sivupohjamoottori, joka toimii siten, että sivupohjiin voidaan kirjoittaa selkeäkielisempää syntaksia, jonka Twig kääntää PHP:ksi. Twigilla ei ole tarkoitus toteuttaa sovellusloogiikkaa, vaikka sekin olisi mahdollista, vaan toiminnallisuus pysyy edelleen MVC:n mukaisesti kontrollereissa.

## **YAML**

YAML on merkintäkieli ja standardi kaikille ohjelmointikielille, jolla voi muuntaa esimerkiksi taulukoita YAML-muotoon. Siinä on yksinkertainen helposti ymmärrettävä syntaksi. YAML -syntaksi koostuu avain-arvo pareista.

## **Yeoman generaattori**

Yeoman generaattori on sovellus, joka voi asentaa kokonaisia sovelluskehitysalustoja tai niiden osia Node.js -ympäristöön. Generaattoreita voi tehdä myös itse nopeuttamaan sovelluksien kehitystä esimerkiksi tiimeissä. Yeoman yhdistää useat sovelluskehityksessä käytetyt työkalut (Grunt, Gulp jne.) sekä niiden riippuvaisuudet (Jade, Less, Sass, Angular.js, Underscore, Bootstrap jne.) yhdeksi paketiksi.

## **Yii**

Yii on PHP:lle vuonna 2008 kehitetty avoimen lähdekoodin MVC -sovelluskehys.

## **Zend PHP Framework**

Zend on avoimen lähdekoodin MVC PHP -sovelluskehys, joka on julkaistu BSD lisenssillä.

# 1 Opinnäytetyön lähtökohdat

## 1.1 Tausta ja toimeksiantaja

Työn toimeksiantajana oli jyvaskyläläinen media-alan yritys Trimedia Oy, jonka ydinliiketoimintaan kuuluvat internetmarkkinointistrategiat, konseptointi ja kampanjasuunnittelu. Trimedia Oy:n tuotteisiin kuuluvat myös Automatik-verkkokauppa ja sen asiakasrajapinnan sovellustukipalvelut.

Trimedia Oy sai tilauksen NRM Oy:ltä uudistaa World Water Hub -verkkopalvelun yleisilmeen ja teknisen toteutuksen palvelun laajentuessa moninkertaiseen kokoon. World Water Hubin aikaisempaa verkkototeutusta ei ollut optimoitu mobiililaitteille eikä se kestäisi uutta datan määrää.

## 1.2 Tavoitteet

Opinnäytetyön tavoitteena oli vertailla eri sisällönhallintajärjestelmiä ja kehitysympäristöjä, jotta pystyttiin valitsemaan sopivin järjestelmä verkkopalvelua varten. Vaatimusmäärittelyn pohjalta toteutettiin alustava prototyyppi, jonka pohjalta uusi verkkopalvelu rakennettiin.

## **2 Vaatimusmäärittely ja elinkaari**

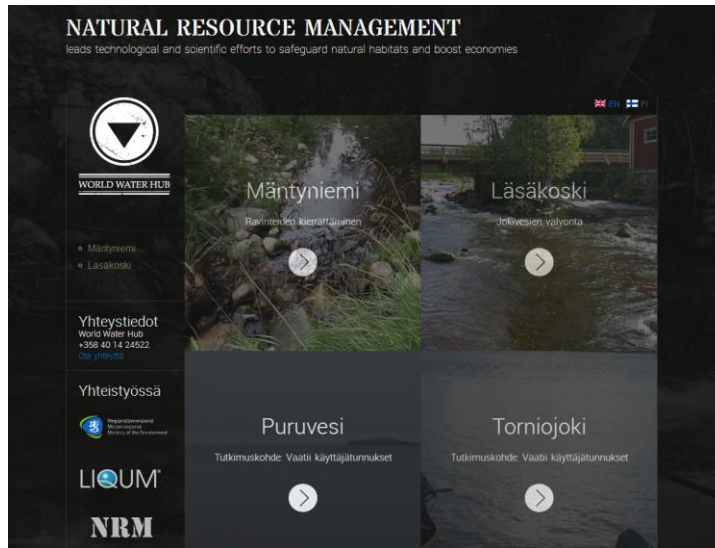
### **2.1 Esiselvitys sivustosta World Water Hub (worldwaterhub.fi)**

#### **2.1.1 Yleistä**

Natural Resource Management seuraa Suomen jokien ja järvien veden laatua veteen asennetuilla antureilla, jotka lähettävät tietoa joen tai järven tilasta reaaliaikaisesti. Mittattavia tietoja ovat muun muassa veden pinnan korkeus, veden laatu ja lämpötila. Vedestä otetaan säännöllisin väliajoin näytteitä, jotka analysoidaan laboratorioissa. Näistä tuloksista nähdään lisääntyvän maankäytön, kuten kaivosteollisuus, turvetuotanto tai maa- ja metsätalous, vaikutus veden laatuun ja vedessä eläviin lajeihin. Tällä hetkellä antureita on asennettuna Mäntyniemessä ja Läsäkoskessa, mutta tarkoituksessa on kattaa valtaosa Itämereen laskevista joista ja järvistä. Näitä tietoja halutaan jakaa maailmanlaajuisesti World Water Hub -verkkopalvelussa.

#### **2.1.2 Keskeiset ongelmat aikaisemmassa toteutuksessa**

Kuviossa 1 näkyy World Water Hubin vanha verkkopalvelu, joka on julkaistu vuonna 2010. Vaikka sivustoa nykyaikaistettiin syksyllä 2013 pienillä käytettävyyssuutoksilla, ei sivusto siltikään vastaa nykyajan vaatimuksia.



Kuvio 1. World Water Hubin alkuperäinen toteutus

Verkkopalvelu laajentuu sisällöllisesti moninkertaisiin mittoihin, sillä sitä kautta aletaan seuraamaan useampia Suomen jokia ja järviä, joten nykyinen tekninen toteutus ei suorituskvyyllisesti riitä vastaamaan uusia tarpeita. Tämän hetkinen palvelu on toteutettu WordPress -julkaisujärjestelmällä, mutta siinä ei ole otettu huomioon laajennettavuutta tai jatkokehitystä. Sivustoa on hankala hallita eikä se sisällä riittäviä ylläpitotyökaluja.

Yksi suuri ongelma sivustossa on responsiivisuuden puute, sivusto ei skaalaudu järkevästi mobiililaitteissa. Tämä on tärkeä osa uudessa toteutuksessa. Palvelun tulee mahdollistaa käytettävyys moderneilla selaimilla ja päätelaitteilla siten, että käyttö on mukavaa ja helppoa.

Sivuston käyttäjät koostuvat pääosin kalastajista ja muista vesillä liikkujista, jotka katsovat niin veden laatuparametreja kuin myös säätietoja sivustolta. Näiden tietojen löytäminen sivustolta on tällä hetkellä hieman hankalaa, koska sivuston rakenne ja ulkoasu eivät ole yhdenmukaisia.

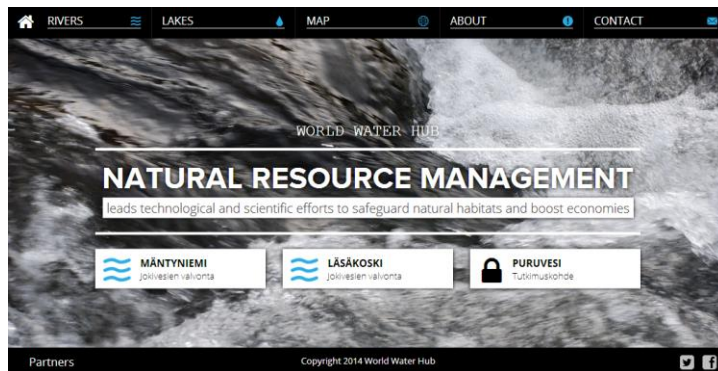
Sivuston päätarkoituksena on jakaa tietoa myös globaalisti, eli tukea tarvitaan eri kielille. Nykyinen toteutus ei skaalaudu dynaamisesti, joten jokainen sivu tarvitsee myös teknisiä muutoksia, jotta kieliversiot saadaan toimimaan.

## 2.2 Vaatimusmäärittely

### 2.2.1 Tavoitteet ja vaatimukset

Verkkopalvelu sisältää viisi pääsivua, joista koostuu sivuston olennaisin rakenne. Palvelun toteutuskieli on englanti. Määrittelyvaiheessa sivustosta toteutettiin ensimmäiset ulkoasut, joissa näkyy ominaisuuksien visualisointi.

Ensimmäisestä näkymästä (ks. kuvio 2) löytyy palvelun nimi ja slogan sekä muokattavat järvi- ja jokiosot, joita voi olla kolme kappaletta. Myöskin sivuston nimen ja sloganin tulee olla muokattavissa suoraan julkaisujärjestelmästä. Etusivun yhteyteen voidaan laittaa myös yhteistyökumppaneiden ja some-linkkien hallinta. Taustalle tulee voida määrittää joko staattinen kuva tai video. Videon voi linkittää halutessaan suoraan YouTubesta, tai järjestelmään voidaan ladata video, jota taustalla käytetään. Ääniä tässä videossa ei toisteta.



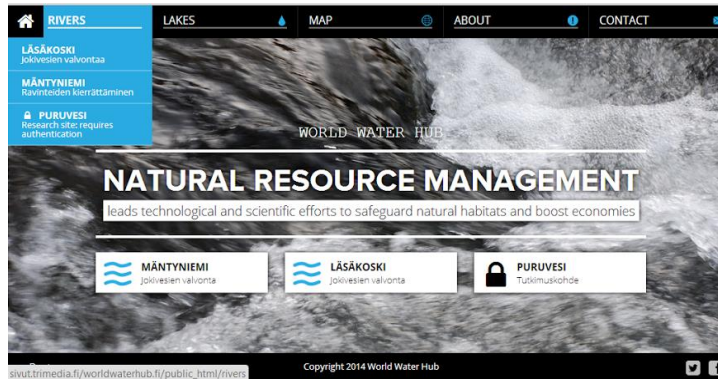
Kuvio 2. World Water Hubin uusittu etusivu

Joen tai järven voi myös asettaa tutkimuskohteeksi, jolloin yksittäinen joki- tai järvinäkymä sekä niiden mittaustulokset ja parametrit ovat sallittuja vain kirjautuneille käyttäjille. Myöhemmin järjestelmää laajennetaan kattamaan eritasoisia käyttäjiä, jolloin kaikki käyttäjät eivät enää näe kaikkea sisältöä.

Verkkopalvelun navigaatio toteutetaan kuvion 3 mukaan siten, että valikko aukeaa kursorin ollessa rivers- tai lakes- painikkeen päällä. Mobiilioptimoidussa versiossa valikko

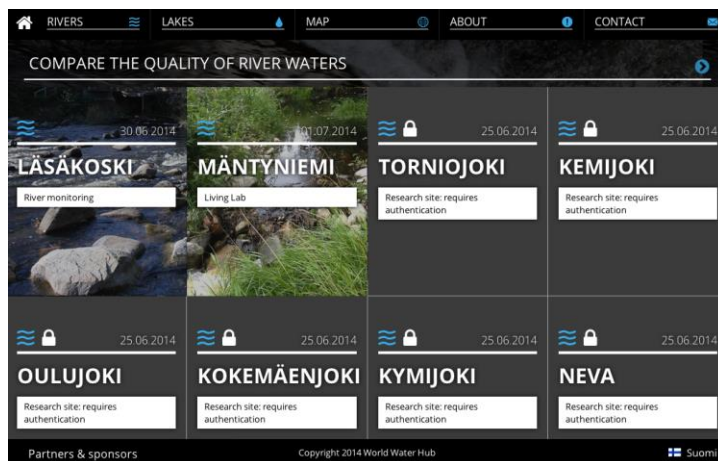


piilotetaan ja käytetään joki- ja järvi-listausnäkyymiä (ks. kuvio 4). Palvelun ensimmäisessä versiossa navigaatio pidetään kapeana siten, että se kasvaa aina alaspäin jokien ja järvien lukumäärän kasvaessa. Myöhemmin tämä muunnetaan koko sivun levyiseksi pudotusvalikoksi johon sisällytetään myös hakutoiminnot ja rajaus, jotta liikkuminen sivustolla helpottuu. Nämä toiminnot otetaan mukaan myös joki- ja järvilistaukseen.



Kuvio 3. Navigaatio avattuna

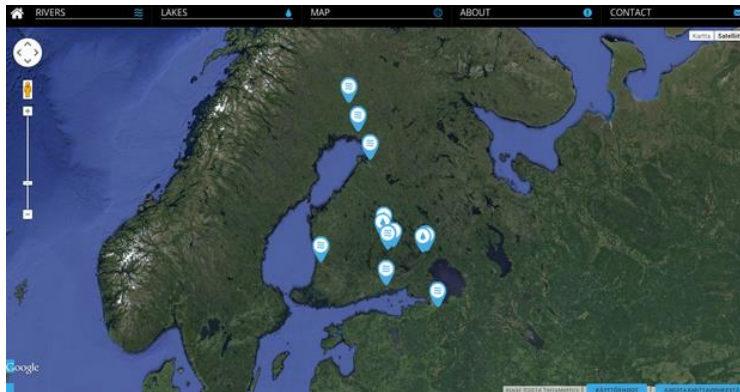
Rivers- ja lakes-painikkeista päästään joki- tai järvilistaukseen. Listaus-näkymästä näkee selkeästi kaikki tällä hetkellä julkaistut joet ja järvet sekä niiden edelliset päivytyspäivämäärät.



Kuvio 4. Jokilistausnäkyymä

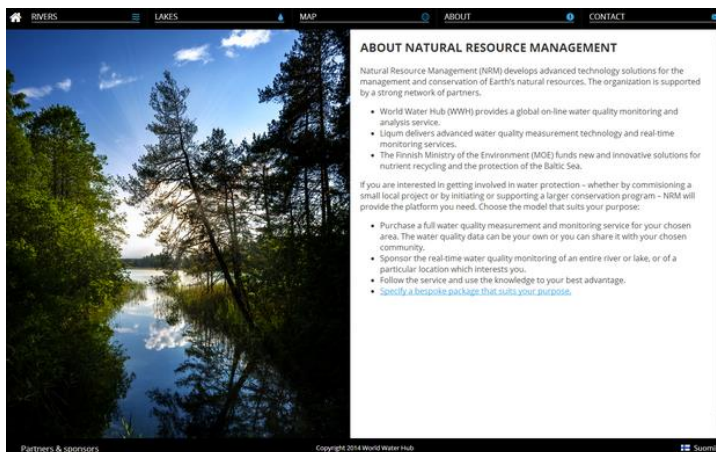
Kuviossa 5 näkyvä kartta-näkymä toteutetaan Google Maps-rajapintaa käyttäen. Karttaan piirretään kaikkien jokien ja järvien sijainnit. Kartalta voi siirtyä suoraan yksittäiseen

joki- tai järvinäkymään, josta näkyy kohteen lisätiedot. Google Mapsin käyttämät sijain-  
timerkit räätälöidään halutunlaisiksi. Ilmaisen version palvelupyyntöjen määrä ei välttä-  
mättä ole riittävä palvelulle, joten tulevaisuudessa saatetaan siirtyä maksulliseen versi-  
oon.



Kuvio 5. Karttanäkymä

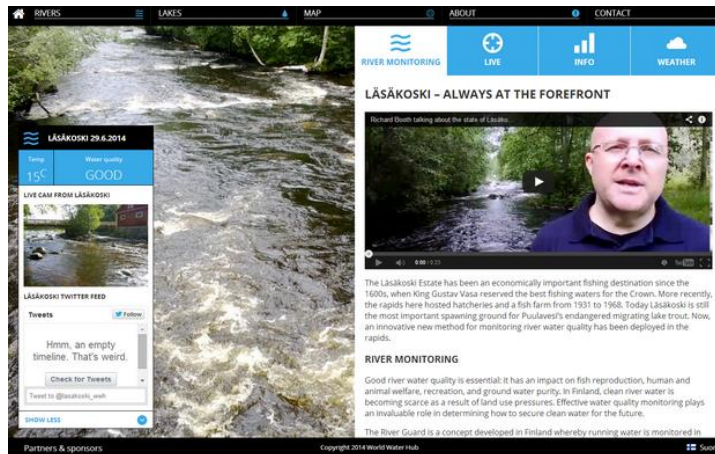
Lisätieto-näkymä (ks. kuvio 6) sisältää World Water Hubin perustiedot sekä tarkemman  
palvelun kuvauksen. Yhteystieto-sivulta käyttäjä löytää palvelun ylläpitäjän yhteystiedot.  
Näihin molempiin näkymiin tulee voida määrittää haluttu taustakuva tai video kuten en-  
simmäisessä näkymässäkin.



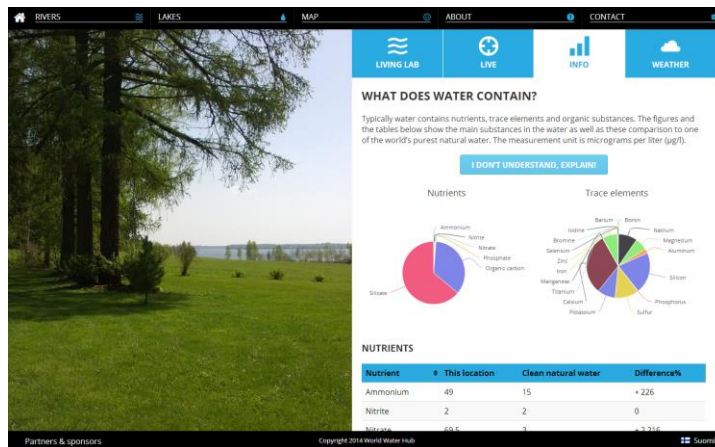
Kuvio 6. Lisätietonäkymä

Kuviossa 7 on yksittäinen jokinäkymä. Jokainen joki tai järvi sisältää neljä päävälilehteä.  
Ensimmäisellä välilehdellä on joen tai järven perustiedot. Toisella välilehdellä on mit-

tauspisteiden tulokset veden laadusta. Kolmas välilehti sisältää laatuparametrit, jotka tulee voida tuoda järjestelmään suoraan Excel-taulukosta (visualisointi kuviossa 8). Neljäs välilehti koostuu kohteen sääennusteesta.



Kuvio 7. Yksittäinen jokinäkymä



Kuvio 8. Näkymä veden laatuparametreista

Säätiedot tuodaan automaattisesti joen tai järven sijaintitietojen perusteella. Yksittäiseen joki- tai järvi-näkymään tulee voida vaihtaa taustaelementti julkaisujärjestelmässä siten, että se voi olla kuva, kartta tai video. Mikäli tähän valitaan kartta, käytetään Google Maps-karttapalvelua ja järjestelmään syötettyjä sijaintitietoja. Sijaintitietoja ei oteta automaattisesti mittausanturilta, jotta voidaan välttää antureihin kohdistuva mahdollinen ilkeä.

Jokaisella joella tai järvellä on oma Twitter-tili, jota päivittää valittu aktiivi. Näiden tilien uusimmat päivitykset tuodaan suoraan yksittäiseen näkymään, jotta saadaan reaaliajassa lisätietoa siitä, mistä johtuen esimerkiksi veden laatu on äkillisesti romahtanut.

### 2.2.2 Toteutusvälineet ja ympäristö

Käytännössä tälle sovellukselle on kaksi vaihtoehtoa, joko LEMP- tai MEAN-pohjainen verkkosovellus. LEMP-toteutusta suositaan LAMP-toteutuksen ohi siksi, koska nginx on palvelimena tehokkaampi kuin Apache. Tämä pohjautuu siihen, että kuten myös Node.js, nginx käyttää asynkronista tapahtumapohjaista pyyntöjenkäsittelyä, jolloin http-pyyntöjä voidaan käsitellä useampi kerrallaan kuin Apachen säie-pohjaisessa käsittelyssä (SimplerCloud 2014). LEMP-pohjaista ympäristöä tässä opinnäytetyössä edustavat WordPress-, October-, Craft- ja Concrete5-julkaisujärjestelmät, sekä osittaisesti myöskin Statamic. MEAN-pohjaisesta ympäristöstä mukana on KeystoneJS-julkaisujärjestelmä.

### 2.2.3 Järjestelmäarkkitehtuuri

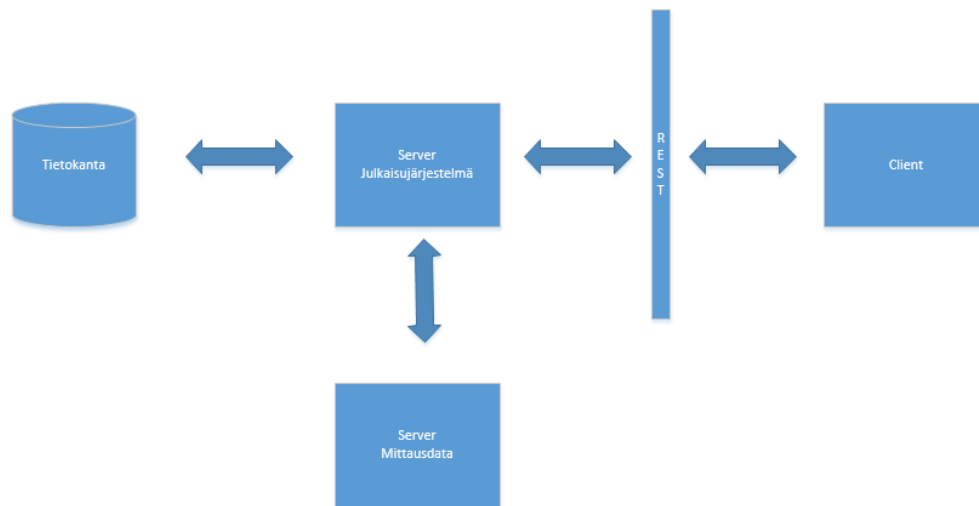
Järvet ja joet edustavat samaa tietotyyppiä, joista jokainen sisältää samantyyppistä tietoa. Ainoa poikkeus tähän on se, että joissain joissa tai järvissä saattaa olla useampia mittauspisteitä tai mitattavat tiedot saattavat vaihdella, toisin sanoen joistain joista tai järvistä mitataan perustietojen lisäksi myös erilaisia mineraalipitoisuuksia. Tämän datan käsittely tulee suunnitella hyvin julkaisujärjestelmää valitessa, jotta tiedonsiirrosta saadaan automaattista ja dynaamista.

Mittaustuloksien toimittaja toimittaa datan palvelimelle XML-muodossa, josta se noudetaan, muotoillaan ja tallennetaan tietokantaan. Tälle tietokannalle toteutetaan rajapinta, josta tietoa voidaan hakea HTTP-pyyntöjen avulla.

Verkkopalvelusta haluttiin tehdä moderni ja nykyaikainen käytettävyys huomioiden, joten järkevintä on hyödyntää REST-rajapintaa (ks. Kuvio 9). REST-rajapinnan valinta poh-

jautui siihen, että tästä samasta järjestelmästä voidaan viedä dataa myös mahdollisiin natiiveihin mobiilisovelluksiin.

Mittausdata-palvelimelta tulevaa dataa ei ole järkevää tallentaa julkaisujärjestelmään, koska sitä ei ole tarpeellista muokata tai hallinnoida.



Kuvio 9. Järjestelmäarkkitehtuuri

## 2.2.4 Projektin riskit

Verkkopalvelun tiedon määrä moninkertaistuu järvien ja jokien laatuparametrien ja muiden tietojen myötä, joten suorituskyky on tärkeä kriteeri toteutuksessa. Mikäli suorituskyky ei riitä ja sivustossa joudutaan käyttämään erilaisia välimuisteja se voi aiheuttaa ongelmia jatkossa. Jokien ja järvien data tulee rajapinnan kautta sivustolle päivittyen aina noin 15 minuutin välein, tässä tulee huomioida mahdollisen välimuistin käyttö ylimääräisen tietoliikenteen välttämiseksi.

Uuden verkkopalvelun applikaatiomaisen toteutuksen räätälöiminen eri selaimille voi osoittautua haasteelliseksi erityisesti vanhemmilla selaimilla ja laitteilla. Palvelu tulee sisältämään paljon erilaisia kuva-animaatioita, videoita ja karttaelementtejä. Nämä tulee ottaa huomioon käyttöliittymän teknistä toteutusta tehtäessä ongelmien välttämiseksi.

## 2.3 Projektin vaiheistus

Ajallisesti projekti aloitettiin 1.4.2014 ja päätettiin 31.6.2014. Tarkempi aikajana löytyy liitteestä 1.

### **Suunnitteluvaihe 1.4.2014 - 9.5.2014**

Suunnitteluvaiheessa projektista tehtiin esiselvitys, jossa oli sovelluksen alustavat vaatimusmäärittelyt sekä ensimmäiset visualisoinnit.

### **Prototyyppi 5.5.2014 - 16.5.2014**

Verkkopalvelusta toteutettiin alustava HTML-prototyyppi, josta näki järjestelmän ominaisuudet ja toiminnallisuuden. Tässä vaiheessa käytiin uudestaan läpi vaatimusmäärittely ja tehtiin viimeiset muutokset ennen järjestelmän teknistä toteutusta. Samalla käytiin myös läpi käyttöliittymäsuunnitelmat ja tehtiin niihin lopulliset muutokset.

### **Tekninen toteutus 19.5.2014 - 15.6.2014**

Järjestelmän tekninen toteutus tehtiin päivitettyjen visualisointien ja vaatimusmäärittelyn pohjalta. Katselmoinnit pidettiin 2-3 viikon välein, jolloin varmistettiin projektin eteneminen.

### **Lopetusvaihe 16.5.2014 - 30.6.2014**

Ennen projektin lopetusta projektin tilaajalle luovutettiin käyttöohjeet ja järjestettiin tarvittava opastus sovelluksen käyttöön. Tekninen toteutus dokumentoitiin, jotta mahdollistettiin jatkokehitys.

## 3 Teknologiavertailu

### 3.1 Valintaperusteet- ja kriteerit

Julkaisujärjestelmiä arvioitiin eri osa-alueiden perusteella. Osa-alueita olivat jatkokehitys, kehittäjäverkosto, ohjeet ja tuki, epäsuora hinta, hakukoneoptimointi, vaatimukset ja toteutustekniikat, suorituskyky ja nopeus, käytettävyys sekä tietoturva. Jokainen osa-alue sai oman painokertoimen sen perusteella, miten tärkeä se on tämän projektin kannalta. Kaikki osa-alueet pisteytettiin asteikolla 1-100. Tämän jälkeen pisteet kerrottiin vastaavilla painokertoimilla. Taulukossa 1 on näkyvissä jokaisen osa-alueen painokerroin. Painokertoimien jakautuminen pohjautuu osa-alueen ratkaiseviin ominaisuuksiin projektissa.

Taulukko 1. Osa-alueiden painoarvojen jakautuminen

Osa-alue	Painokerroin
Epäsuora hinta	1
Kehittäjäverkosto, ohjeet ja tuki	2
Jatkokehitys	3
Vaatimukset ja tekniikat	3
Hakukoneoptimointi	4
Käytettävyys	4
Suorituskyky ja nopeus	5
Tietoturva	5
(asteikko 1 ei tärkeä - 5 tärkeä)	

#### 3.1.1 Epäsuora hinta

Epäsuoralla hinnalla tässä tapauksessa tarkoitetaan sitä tuleeko järjestelmän käyttöönotosta ylimääräisiä kustannuksia esimerkiksi vaatimusten tai tekniikoiden osalta. Yllättäviä kustannuksia saattaa tulla myös lisäosista ja moduuleista mikäli järjestelmä ei sisällä vaadittuja perustoiminnallisuuksia ja niiden toteuttaminen itse on projektin laajuuteen

nähden liian työlästä. Hinta ei tässä projektissa ole ratkaiseva tekijä, sillä jos ostetuilla ominaisuuksilla tai lisäosilla saadaan nopeutettua projektin toteuttamista, on siitä paljon hyötyä.

### **3.1.2 Kehittäjäverkosto, ohjeet ja tuki**

Julkaisujärjestelmiä arvioidaan myös niiden elinkaaren ja kehittäjäverkon pohjalta. Verkoston tulee vaikuttaa luotettavalta, jotta voidaan määrittää miten kauan järjestelmä tullaan jatkossa päivittämään. Tässä projektissa omaa kustomointia joudutaan joka tapauksessa tekemään suhteellisen paljon. On tärkeää, että ongelmatilanteissa saadaan riittävästi tukea, mikäli sisällönhallintajärjestelmän ytimeistä tai lisäosasta löytyy vaikka jokin virhe, jota itse ei voi järkevästi korjata.

### **3.1.3 Jatkokehitys**

Järjestelmän pitää olla modulaarinen ja helppo jatkokehittää myös muille, kuin projektin tämänhetkiselle toteuttajalle. Tästä syystä järjestelmän jatkokehittäminen ja millä tavoilla voidaan jatkokehittää sekä siihen liittyvä ohjeistus on tärkeä osa järjestelmän räätälöintiä. Jatkokehitykseen liittyy myös kuinka paljon järjestelmässä on valmiita hyödynnettäviä lisäosia ja ominaisuuksia sekä kuinka paljon toiminnallisuutta joudutaan toteuttamaan itse.

### **3.1.4 Vaatimukset ja tekniikat**

Valitun julkaisujärjestelmän tulee toimia normaalissa palvelinympäristössä. Mikäli se vaatii ylimääräisiä kirjastoja tai yllättävän tehokasta palvelintä tästä voi tulla yllättäviä kustannuksia. Tekniikat ovat ratkaisevassa osassa projektin toteutuksessa. Onko ohjelmointiin käytetty sovelluskehys kehitystiimille jo ennestään tuttu vai joudutaanko sisältämään jokin uusi tekniikka. Uusien tekniikoiden käyttö vie huomattavasti enemmän resursseja kuin jo ennestään tunnettu sovelluskehys. Myöskin ohjelmointityyli on ratkaisevaa sekä käytetäänkö järjestelmässä MVC -mallia.



### 3.1.5 Hakukoneoptimointi

Nykypäivän verkkopalveluprojekteissa hakukoneoptimointi on avainasemassa jo kehitysvaiheessa. Monet julkaisujärjestelmät huomioivat hakukoneoptimoinnin jo itsessään. Hakukoneet ovat merkittävä sivuston kävijämäärään vaikuttava tekijä. Verkkopalvelu Googlen hakukonesijoitukseen vaikuttaa tekniseltä puolelta esimerkiksi käyttääkö sivusto SSL-sertifikaattia, onko sivusto optimoitu mobiililaitteille, sivuston latausnopeus, AJAX-toimintojen ja rajapintojen toteutus sekä linkkirakenne (Backlinko 2014).

### 3.1.6 Käytettävyys

Käytettävyys ratkaisee miten sivuston hallinnoija kokee kyseisen järjestelmän hallintapaneelin ja sen käytön. Kyse ei tässä tapauksessa ole client-puolen käytettävyydestä, jonka sivuston käyttäjä näkee vaan se osa, jota sivuston ylläpitäjä joutuu hyödyntämään. Käytettävyyteen vaikuttavia tekijöitä ovat käytön helppous, kuinka selkeästi hallintapaneeli on rakennettu, pystyykö hallintapaneelia kustomoimaan tai tiettyjä ominaisuuksia priorisoimaan esimerkiksi ylimääräisiä ominaisuuksia piilottamalla tai järjestystä vaihtamalla. Julkaisujärjestelmän hallinnan tulee olla helppoa, vaivatonta sekä nopeaa oppia. Myös visuaalisuus ja mobiilioptimointi ovat tärkeitä kriteereitä. Käytettävyyteen vaikuttavia tekijöitä ovat käytön helppous, kuinka selkeästi hallintapaneeli on rakennettu, pystyykö hallintapaneelia ja sen ulkonäköä kustomoimaan tai tiettyjä ominaisuuksia priorisoimaan esimerkiksi ylimääräisiä ominaisuuksia piilottamalla tai järjestystä vaihtamalla.

### 3.1.7 Suorituskyky ja nopeus

Järjestelmän tulee olla nopea ja tehokas. Järjestelmän tulee pysyä riittävän nopeana myös suurilla kävijämäärillä. Suorituskyky vaikuttaa kriittisesti palvelun käytettävyyteen vaikka käyttökokemusta voidaankin parantaa teknisen toteutuksen optimoinnilla.

### 3.1.8 Tietoturva

Tietoturvan saralla tutkitaan järjestelmän luotettavuutta, sekä löytyykö jo lähdekoodia selattaessa jotain epämääriä. Onko olemassa ennakkotapauksia, jonka perusteella voidaan päätellä julkaisujärjestelmän tietoturvassa olevan poikkeamia tai heikkouksia. Tietoturva vaikuttaa sivuston luotettavuuteen ja siltä osin bisneslogiikkaan. Sivuston tulee olla aina saavutettavissa sijainnista ja ajasta riippumatta.

## 3.2 Testiympäristö

Kaikki järjestelmät testattiin Apache- tai nginx-palvelinohjelmalla nopeuksien vertaamiseksi. JavaScript-pohjaiseen Ghost-julkaisualustaan käytettiin Node.js HTTP-kirjastoa palvelinympäristön ja palvelinpuolen ohjelmoinnin hallintaan. Palvelinsovellukset asennettiin Digital Oceanin 512MB / 1 CPU pilvipalvelimelle, jossa oli 20 gigatavun SSD-kiintolevy. Palvelimen asetukset pidettiin oletusasetuksilla, jotta kaikista järjestelmistä saatiin mahdollisimman realistinen käyttökokemus. Käyttöjärjestelmänä oli Ubuntu 14.04 x64. Tietokantoina käytettiin PHP-pohjaisissa sovelluksissa MySQL-tietokantaa ja JavaScript-pohjaisessa MongoDB-tietokantaa.

## 4 Vertailtavat järjestelmät

### 4.1 Yleistä

#### 4.1.1 Millä perusteella järjestelmät valittiin?

Vertailtavia järjestelmiä olivat Concrete5, Craft, Keystone.js, October, Statamic ja WordPress. Nämä järjestelmät valittiin sen takia, koska ei haluttu, että kaikki järjestelmät olisivat niitä yleisimmin käytettyjä valtavirtajärjestelmiä, vaan haluttiin tutkia myös vaihtoehtoisia nopeasti kasvavia järjestelmiä ja niiden tuomia mahdollisuuksia.

Näistä järjestelmistä Concrete5, Keystone.js, October ja WordPress löytyvät GitHub-verkkopalvelusta. Järjestelmien saama tähtien määrä GitHubissa kertoo miten suosittuja ne ovat kehittäjien keskuudessa tällä hetkellä. Tähtiä voisi verrata Facebookin ”tykkää” – toiminnallisuuteen, eli tähdet kertovat mikä on tällä hetkellä eniten nosteessa.

WordPressillä on GitHubissa 4027, Octoberilla 1377, Keystonella 1318 ja Concrete5:lla 590 tähteä (GitHub 2014). Siitä huolimatta vaikka Concrete5:lla on paljon vähemmän ”tykkääjiä” kuin esimerkiksi Keystonella, on Concrete5 silti huomattavasti enemmän käytetty kuin Keystone. Tämä kertonee siitä, että Concrete5:n ympärillä ei vanhempana järjestelmänä ole enää niin paljon nostetta eikä sitä kehitetä yhtä aktiivisesti.

WordPress ja Concrete5 ovat yleisimmin käytettyjä pienempien web-projektien sisällönhallintajärjestelmiä, mutta niillä on paljon eroavaisuuksia, jotka tarvitsevat lisäselvitystä. Craft on suhteellisen uusi järjestelmä, joka yrittää vähentää sisällönhallintajärjestelmissä olevia turhia ominaisuuksia ja keskittyä vain sisällöllisiin ominaisuuksiin. Statamic on julkaisujärjestelmä, joka ei käytä julkaisussa tietokantaa ollenkaan. Statamicin suosio on kasvussa sillä vuotta 2014 on povattu flat-file järjestelmien vuodeksi (Type & Grids 2013). October on todella suosittuun Laravel – sovelluskehukseen perustuva järjestelmä, joten sen käyttäjäkunta löytynee hyvinkin pitkälti Laravelin käyttäjistä. Siitä voisikin päätellä, että October tulee kasvamaan merkittävästi tulevien vuosien aikana niin kehittäjäyhteisöltään kuin myös ominaisuuksiltaan. October valittiin vertailtaviin järjestelmiin siksi, että Laravel on projektin kehitystiimille tuttu sovelluskehys. Keystone on täysin vaihtoehtoinen niin tekniikaltaan kuin palvelinympäristöltään. Keystone on mukana Node.js:n suosion ja yleistymisen vuoksi. Ghost on yksi Node.js:lle tehty julkaisuuteen päässyt julkaisujärjestelmä. Ghostia ei kuitenkaan otettu tähän työhön mukaan sen takia, että se käyttää SQLite – tietokantaa. Työhön valittiin MongoDB:tä hyödyntävä Keystone, jotta saatiin jotain oikeasti erilaista ja uutta.

### 4.1.2 Järjestelmillä tehdyt toteutukset

Järjestelmät käytiin läpi yksitellen ja arvioitiin niiden ominaisuuksien sopivuus World Water Hub – verkkopalveluun. Jokaisella järjestelmällä toteutettiin yksi joki-tyyppinen mallisivu. Sivu sisälsi kaiken tiedon, mitä vaatimusmäärittämissä oli vaadittu. Tällä menettelyllä saatiin hyvin esille järjestelmien väliset erot ja jokaisen järjestelmän soveltuvuus verkkopalvelun alustaksi. Samalla järjestelmän jatkokehittämisestä ja sen vaatimuksista saatiin jonkinlaisen käsitys ja pystyttiin edes jollain tasolla arvioimaan tarvittavia resursseja mikäli World Water Hub – verkkopalvelu olisi toteutettu kyseisellä järjestelmällä.

## 4.2 Concrete5

### 4.2.1 Yleistä

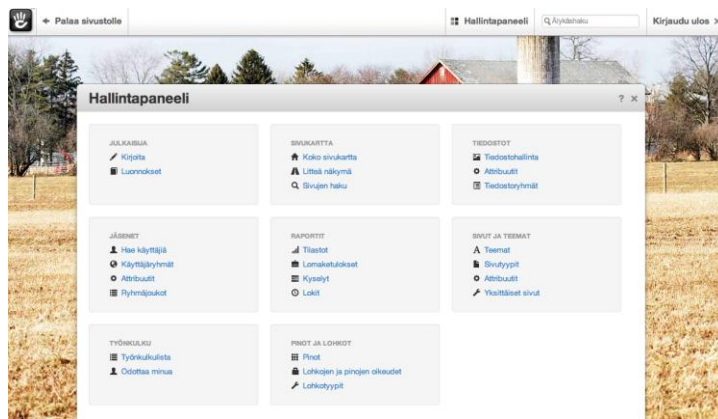
Concrete5 on MIT-lisenssillä julkaistu avoimen lähdekoodin sisällönhallintajärjestelmä ja sovelluskehys, joka alun perin oli julkaistu helpoksi alustaksi jatkokehitystä varten muille web-kehittäjille. Nykypäivänä se on kuitenkin laajentunut käyttäjäystävällisemmäksi alustaksi, jonka käyttöön ei tarvitse teknistä osaamista.

Concreten ensimmäinen versio julkaistiin vuonna 2003. Sen suunnittelivat ja toteuttivat Franz Maruna ja Andrew Embler tarkoituksena luoda järjestelmä, joka olisi helppo, joustava ja kestävä (SourceForge 2008). Uusin versio 5.6 on julkaistu huhtikuussa 2014. Järjestelmä pohjautuu osittain PHP Zend Frameworkiin, joka on yksi käytetyimmistä PHP-sovelluskehyksistä. Noin puoli miljoonaa verkkopalvelua käyttää Concrete5-järjestelmää (W3 Techs 2014).

### 4.2.2 Ominaisuudet

Concrete5 on perusasennukseltaan hyvin pelkistetty sisältäen vain perusominaisuudet, joihin kuuluu sisällön-, oikeuksien- ja käyttäjienhallinta sekä mediakirjasto.

Tiedostonhallinta on samaa pelkistettyä ja minimalistista linjaa kuin WordPressissäkin, sillä poikkeavaisuudella, että kuvia ei voi muokata järjestelmän kautta. Sen sijaan Concrete5 sisältää laajennetun haun, jotta oikeat tiedostot löytyvät helpommin. Sama puute yhdisti Concrete5- ja WordPress järjestelmiä, kansioita ei voi luoda tiedostojenhallintaan oletuksena. Tiedostoista voidaan tehdä erilaisia tiedostoryhmiä, jotka vastaavat omia kansioitaan, mutta ovat hieman monimutkaisempia. Hallintapaneelista huomaa, että käyttöliittymä ei ole kovinkaan nykyaikainen (ks. kuvio 10).



Kuvio 10. Concrete5:n hallintapaneeli, jonka ulkoasu hyödyntää käytöstä poistettua Bootstrap 2.0 -sovelluskehystä

Järjestelmän räätälöinti tapahtuu erityyppisillä sisältölohkoilla (block), joita saa lisää lisäosilla (add-on), joista harmittavan moni on maksullisia. Myös käyttäjähallintaa voidaan toteuttaa lohkoilla. Lohkoihin voidaan määrittää, millä käyttäjällä käyttäjärhmillä on oikeudet nähdä sisältö.

Erityyppisiä sivuja voidaan julkaista helposti ilman koodiosaamista järjestelmän sisällä. Julkaisut voi halutessaan jättää myös luonnoksiksi. Sivujen muokattavien sisältökenttien lisääminen tapahtuu sivuattribuuteilla (ks. kuvio 11), joita saa helposti lisättyä suoraan hallintapaneelista ilman teknistä osaamista. Sivuattribuutit sisältävät kaikki perustyytit kuten teksti ja kuva tai tiedosto. Myös käyttäjillä ja medially voi olla omia attribuutteja. Esimerkiksi kuva-tiedostoon voidaan lisätä missä kuva on otettu tai käyttäjälle voidaan laittaa lisätiedoksi osoite tai luottokortin numero.

**Lisää Joki**

**Yleiset ominaisuudet**

Nimi

URL-osoitteen päätte

Julkinen pvm/aika   :

Kuvaus

**Omat attribuutit**

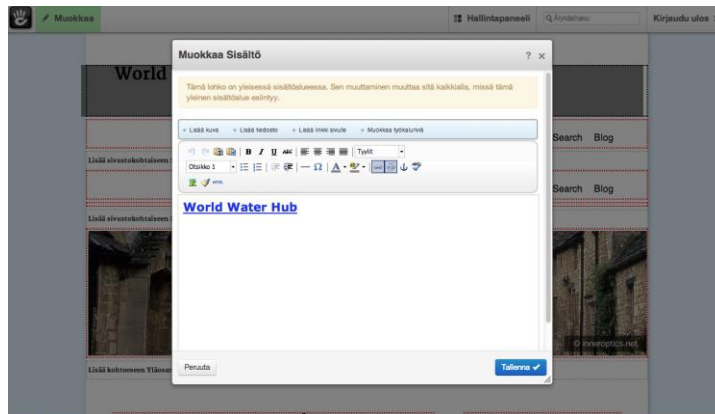
Tutkimuskohde  Kyllä

Lyhyt kuvaus

**Rich text editor toolbar:** Tyylit, Kappale, Kirjasinkoko, Kirjasin, Bold, Italic, Underline, Text color, Background color, Bulleted list, Numbered list, Indent, Outdent, Undo, Redo, Link, Unlink, Image, Table, Source, Preview, Fullscreen, Help.

Kuvio 11. Mukautettua tietosisältöä (page attributes) joki-tyyppisellä sivulla

Concrete5:n edistynein toiminnallisuus on sisällön reaaliaikainen editointi (in-context editing, ks. kuvio 12), joka mahdollistaa sivun lohkojen muokkaamiseen samassa selainnäkyvässä kuin mikä loppukäyttäjälle näkyy.



Kuvio 12. Reaaliaikainen editointi (in-context editing)

Concrete5 tallentaa automaattisesti myös lokitietoja ja статистиikkaa sivuston toiminnasta ja käytetyistä resursseista. Statiistikasta näkyy esimerkiksi viimeaikaisen sivujen katselut, rekisteröitymiset ja viimeisimmät lataukset.

Muista järjestelmistä poiketen Concrete5:lla voi tehdä työnkulkulistoja, joilla voi rajoittaa käyttäjien ja käyttäjäryhmien pääsyä hallintapaneeliin esimerkiksi päivämäärän ja ajan mukaan.

Hakukoneoptimointia varten Concreteen on sisäänrakennettu työkalut, joilla julkaisuihin voi lisätä metatietoa.

Concreteen ydin on eriytetty muusta ohjelmointilogiikasta. Olemassa olevia luokkia voi ylikirjoittaa siten, että päivittäminen onnistuu silti helposti ja räätälöidyt luokat säilyvät.

### **4.2.3 Vaatimukset**

Concrete5 vaatii toimiakseen vähintään PHP 5.2.4, suositus on kuitenkin 5.3, sekä PHP moduulit cURL, zip, mcrypt, openssl, GD, mysql, mbstring ja iconv. Muistia tulee olla käytettävissä ainakin 64 megatavua. MySQL version on oltava vähintään 5.0. Palvelimeksi suositellaan Apachea, järjestelmä toimii kuitenkin myös nginx:lla.

### **4.2.4 Lisäosat**

Concreteen virallisilta verkkosivuilta löytyy kauppapaikka, josta voi ladata maksullisia ja maksuttomia lisäosia. Kaikki maksulliset lisäosat pohjautuvat sivustokohtaiseen kertamaksuun. Kauppapaikassa olevat lisäosat testataan ennen niiden lisäämistä, jotta varmistetaan niiden toimivuus ja yhteensopivuus. Myös Concreteen kehittäjät myyvät lisäosia kauppapaikassa, jolloin avoimen lähdekoodin periaate toteutuu vain itse järjestelmässä.

### **4.2.5 Jatkokehitys**

Lisäosien ja teemojen kehitys on kattavan dokumentaation osalta vaivatonta ja helppoa.

Concrete5:n jatkokehitys noudattaa MVC-mallia, joka tekee teemojen ja lisäosien toteutuksesta helppoa omaksua.

Uusia sisältöalueita ja kenttiä voidaan luoda tekemällä objekti halutusta luokasta. Näitä olemassa olevia luokkia, kuten GlobalArea, voidaan periyttää tai laajentaa uusilla luokilla. Näiden kenttien sisältöjä voidaan muokata esteettömästi hallintapaneelin kautta.

Teemoihin voidaan lisätä halutut kentät ohjelmallisesti ja ne tulostuvat automaattisesti hallintapaneeliin, tai voidaan myös käyttää hallintapaneelissa olevia työkaluja.

Concrete5:lla voidaan toteuttaa yksittäisiä sivuja (single pages), mikä tarkoittaa sivua joka sisältää yksilöllisen sisällön. Sivun sisältö ei ole monistettavissa muille sivuille, vaan ainoastaan tämä yksittäinen sivu sisältää kyseisen toiminnallisuuden, kuten esimerkiksi kirjautumissivu. Yksittäisten sivujen toteuttaminen on yksinkertaisempaa kuin sisältölohkon toteuttaminen, myöskin kehittäjällä on vapaammat kädet sisällön suhteen. Jokaisella yksittäisellä sivulla on oma kontrollerinsa, joka MVC-mallin mukaisesti sisältää sivuun kuuluvan toiminnallisuuden.

Työkaluilla (tools) on pääsy Concrete5:n ympäristöön ja ne pystyvät keskustelemaan rajapintojen kanssa, mutta niillä ei kuitenkaan ole omia sivuja. Työkaluilla voidaan toteuttaa erilaisia avustavia sovelluksia.

Concrete5:ssä on automaattinen loki-toiminto, joka kirjaa ylös kaikki poikkeustilanteet ja ilmoitukset. Tämä on kehitystä huomattavasti helpottava tekijä. Myöskin suorituskyvyn varmistamiseksi käytetyn välimuistin saa asetuksista kehityksen ajaksi helposti pois päältä.

#### **4.2.6 Yhteenveto**

Concrete5 olisi World Water Hubia ajatellen varteenotettava vaihtoehto. Pääosa kaikista ominaisuuksista voidaan toteuttaa järjestelmän ominaisuuksilla. Kaikkia ominaisuuksia järjestelmästä ei kuitenkaan löytynyt. Osalle löytyy maksullinen lisäosa, mutta jonkin verran ominaisuuksia jouduttaisiin toteuttamaan itse, kuten esimerkiksi sähköpostilla julkaistavat riistakameran kuvat.

Concrete5:n jatkokehittäminen on MVC-mallin myötä nopeasti sisäistettävissä ja järjestelmän ohjeista löytyy kattavasti tietoa. Concrete5 toimii kaikissa yleisimmissä palvelinympäristöissä, joten siltä osin ongelmaa ei ole. Hakukoneoptimointi on huomioitu siten, että julkaisuihin voidaan lisätä omia avainsanoja. Järjestelmällä voidaan lisäksi seu-



rata sivuston kävijöitä sekä käytettyjä hakusanoja, joilla sivuille on tultu. Myöskin ulkopuolisten kävijäseuraajien, kuten Google Analyticsin koodin lisääminen onnistuu helposti suoraan järjestelmän hallintapaneelistä. Teknisten ominaisuuksien, kuten Ajaxin haku-koneoptimointiin Concrete5 ei kuitenkaan ota kantaa.

Käytettävyyttä Concrete5:ssa ei ole huomioitu riittävästi. Hallintapaneeli on epäjohdonmukainen ja sieltä on hankala löytää etsimäänsä. Useasti reaaliaikainen editointi vain pahentaa tilannetta. Suorituskyvyltään Concrete5 toimii sutjakkaasti. Sen nopeudessa tai suorituskyvyssä ei ole havaittavia pullonkauloja. Internetissä ei ole tapauksia Concrete5:n tietoturvaongelmista, joten voidaan olettaa, että tietoturva on hyvällä tasolla (Concrete5 2011).

## **4.3 Craft**

### **4.3.1 Yleistä**

Craft sisältää yhden tämän opinnäytetyön parhaiten toteutetuista responsiivisista hallintapaneelin käyttöliittymistä. Craftin saa perusominaisuuksilla ilmaiseksi, mutta lisäosat ovat kaikki maksullisia. Lisäosia on kattava valikoima, jolla järjestelmää saa nopeasti laajennettua tarpeita vastaavaksi. Craftista on tarjolla myös maksullinen PRO versio, joka sisältää huomattavasti kattavammat ominaisuudet, kuten laajennetun käyttäjähallinnan ja lokalisointimahdollisuudet (Craft Pricing 2014).

Craft on tuore projekti, joka on suunniteltu nimenomaan erityyppisten tietojen hallintaan. Sen kehittämisessä on huomioitu erityisesti joustavuus ja sisällönhallinnalliset ominaisuudet. Craft pohjautuu Yii PHP -sovelluskehikseen.

### **4.3.2 Ominaisuudet**

Craftissa, kuten WordPressissä ja Concretessäkin, pystyy luomaan mukautettuja kenttiä. Craft on vienyt tämän mallin hiukan pidemmälle. 16 erityyppisestä mukautetusta ken-

tästä voidaan muodostaa sisältölohkoja ja niistä edelleen matriisiin (matrix field type), jonka voi lisätä sisältöön. Tällä toiminnallisuudella sisältöä voi niputtaa yhteen tai luoda toistuvaa sisältöä.

Yhdessä matriisissa voi olla rajaton määrä sisältölohkoja, joista ylläpitäjä voi valita sisältöä lisätessään. Jokaisella loholla on omat mukautetut kenttensä. Matriisien käyttökohdetta ei ole rajattu, vaan niitä voi käyttää missä vain kuten käyttäjien tiedoissa ja tiedostoissa. Matriiseja on helppo tulostaa sivupohjiin, jotka käyttävät samaa moottoria kuin October, eli Twigia.

Craftissa on kattava tiedostojenhallinta, jossa on hyvin rakennettu käyttöliittymä. Toisin kuin esimerkiksi WordPressin oletustiedostojen hallinta, Craft sisältää kattavasti ominaisuuksia resurssien hallintaan. Tiedostojenhallinnassa voidaan luoda kansioita, siirtämään tiedostoja drag & drop -toiminnolla, uudelleen nimeämään sekä valitsemaan missä näkymässä tiedostoja haluaa selata.

Kuvien kokoja voidaan muuttaa helposti sisäänrakennetulla työkalulla, joka muuttaa esikatselukuvat ja kuvien suurennot uusiin mittoihin. Esimerkiksi WordPress ei osaa automaattisesti tehdä uusia esikatselukuvia, vaan tätä varten on räätälöityjä lisäosia, jotka skannaavat kansiorakenteen.

Craftin erikoisuuksiin kuuluu mahdollisuus luoda riippuvaisuuksia eri sisältöjen ja sisältötyyppien välille. Tällä ominaisuudella voitaisiin World Water Hubissa toteuttaa sivuna esimerkiksi jokilistaus-sivu ja linkittää sinne kaikki alueella olevat joet ja järvet.

Sivuston rakenteen hallintaan Craftissa on kolme erilaista osiotyyppiä. Yksittäiset sivut (single), kanavat (channels), sekä rakennelmat (structures).

Yksittäisillä sivuilla, kuten esimerkiksi yhteystietosivulla, on omat uniikit sisältökenttensä ja oma yksittäinen url-osoitteensa, joka voi olla erimuotoinen kuin tavallisilla julkaisuilla. Yksittäisellä sivulla voi olla vain yksi yhdistetty julkaisu. Yksittäisen sivun ja julkaisun ero

on sama kuin WordPressissä, eli yksittäisellä sivulla on oma hierarkia ja riippuvuudet, kun taas julkaisulla ei.

Kanavia käytetään sellaisien julkaisujen ryhmittelyyn, joilla ei ole minkäänlaisia julkaisujen välisiä suhteita, kuten esimerkiksi blogikirjoitukset tai uutiset. Samaa kanavaa voidaan käyttää useammassa sijainnissa.

Rakennelmia käytetään, kun julkaisujen välillä on suhteita. Rakennelmia voidaan lajitella ja niillä voi olla myös hierarkia. Käytännössä julkaisut tässä tapauksessa ovat yksittäisiä sivuja.

Julkaisu voivat olla mitä tyyppiä tahansa, ei esimerkiksi pelkkä blogikirjoitus, vaan jokin itse lisätty tyyppi, kuten järvi tai joki. Craftissa on käytössä eri taksonomiat, kuten muisakin järjestelmissä.

Mielenkiintoinen yksityiskohta on myös, että jokainen käyttäjä voi kustomoida oman aloitussivunsa sisällön. Tämä ominaisuus on valitettavasti World Water Hubiin turha, mutta sitä ei muista järjestelmistä oletuksena löydy.

Kuten WordPressissä ja Octoberissa, päivittäminen onnistuu hallintapaneelin kautta parilla klikkauksella. Craftin muista poikkeava ominaisuus on tietokannan varmuuskopiointi hallintapaneelin kautta, WordPressiin tämän toiminnallisuuden saa vasta lisäosana.

Craft on julkaistu kolmella eri lisenssillä, yksityiskäyttöön tarkoitettulla ilmaisella lisenssillä sekä kahdella kaupallisella lisenssillä (Craft Pricing 2014).

Craft Personal on tässä työssä käytetty Craftin ilmaisversio. Se sisältää kaikista pelkistetyimmät ominaisuudet. Sisältölohkojen kanavat (channels) ja yksittäiset sivut (singles) ovat määrällisesti rajattu siten, että kanavia voi olla 1 ja yksittäisiä sivuja 5 kappaletta. Rakennelmia Personalissa ei ole ollenkaan. Sisällöllä voi kuitenkin olla rajaton määrä räätälöityä sisältöä ja kenttiä. Personal ei sisällä versionhallintaa eikä luonnoksia. Käyttäjänhallintaa ei myöskään ole, käyttäjien määrä on rajattu yhteen pääkäyttäjään.

Craft Client ja Craft PRO sisältävät molemmat rajattoman määrän kanavia, yksittäisiä sivuja ja rakennelmia. Molemmissa on myös kattava versiohallinta ja julkaisuluonnokset, sekä kustomoitu kirjautumis- ja sähköpostinäkyvä. Myöskin sähköpostien sisällön saa itse kirjoittaa oletussisältöjen sijasta. Craft Clientissa käyttäjien määrä on rajattu yhteen pääkäyttäjään ja yhteen normaalikäyttäjään. Craft Prossa käyttäjiä voi olla rajattomasti sekä niillä on ryhmiä, oikeuksia ja mahdollinen julkinen rekisteröityminen.

Lisäksi Craft PRO:n ominaisuuksiin kuuluu tiedostojen tallentaminen suoraan pilveen, kuten Amazon S3:een tai Google Cloudiin. PRO-versiosta löytyy myös lokalisointi vaihtoehtoja, käyttäjien oikeuksia voidaan rajata lokalisoinnin mukaan sekä sisältöjä voidaan kääntää vapaasti eri kielille.

Craftiin on kehitetty noin satakunta erilaista ilmaista lisäosaa sivuston välimuistin tyhjentämisestä sähköpostilistaan.

### **4.3.3 Vaatimukset**

Craft vaatii toimiakseen PHP version 5.3.0 tai uudemman, eli vanhempia ympäristöjä ei tueta. Tietokannan tulee olla MySQL 5.1.0 tai uudempi. PHP ei tarvitse kuin 32 megatavuun muistia Craftia suoritettaessa, mikä on todella vähän verrattuna esimerkiksi WordPressin suositettuun 256 megatavuun. Craft myöskin ottaa voimakkaasti kantaa ilmoittamalla, että hallintapaneeli ei toimi kuin moderneilla selaimilla. PHP laajennuksista Reflection, PCRE, SPL, PDO, PDO MySQL, Mcrypt, GD, OpenSSL, Multibyte String, cURL ja Crypt tulee olla asennettuina (Craft Requirements 2014).

### **4.3.4 Lisäosat**

Craftin lisäosille ei ollut vielä olemassa yhtenäistä kauppaa, josta lisäosia voisi ostaa, mutta niitä löytyi joitakin kymmeniä netistä. Osa lisäosista ovat ilmaisia, mutta laajemmat lisäosat kuten verkkokauppa- tai käyttäjien jäsenyyslisäosa ovat maksullisia (Straight Up Craft, 2014). Craftiin julkaistaan tulevaisuudessa monia kiinnostavia lisäosia kuten

esimerkiksi REST Easy, joka nimensä mukaisesti toteuttaa REST-rajapinnan (Github rest-easy 2014).

### 4.3.5 Jatkokehitys

Craft sisältää kattavat rajapinnat ja dokumentaation lisäosien kehittämiseen. Lisäosat toteuttavat oliopohjaista MVC-mallin mukaista ohjelmointia. Jokaisella lisäosalla on oma luokkansa ja kontrollerinsa. Ohjelmointilogiikka on palveluissa (service) ja kontrollerit toimivat pyyntöjen käsittelijöinä. Craftin erityispiirteinä ovat normaalien MVC-mallin mukaisien mallien (model), jotka toimivat datan esittelijöinä, lisäksi tietueet (record), jotka voivat myös tallentaa ja muokata tietokannan dataa. Olemassa olevia tietotyyppejä voidaan laajentaa tai luoda kokonaan omia.

Craftin tietokantakäsittely tehdään kyselyrakentajan (Query Builder) kautta, jotta saadaan yhtenäinen tietokantarajapinta. Tämä mahdollistaa sen, että myöhemmin voidaan tukea myös muita relaatiotietokantoja kuin MySQL.

Kuten WordPressissä, myös Craftissa on koukkuja (hooks) ja tapahtumia (events). Koukuilla voidaan muuttaa tai laajentaa ytimeen kuuluvaa toiminnallisuutta, kun taas tapahtumissa suoritetaan toiminnallisuutta silloin kun jokin tietty tapahtuma on toteutunut. Koukkuja on 6 kappaletta erilaisiin tarkoituksiin. Tarkoitukset ovat Twig -laajennuksen lisäys, välimuistin polkujen lisäys, lisäosan URL-osoitteiden reititys, front-end osoitteiden reititys, sähköpostiviestien lisäys ja käyttäjäoikeuksien hallinta. Tapahtumien aikana voi laukausta erilaista lisätoiminnallisuutta. Toiminnallisuuksia on yhteensä 36 kappaletta, kuten esimerkiksi kategorian lisäys tai poisto, sekä lisäosan lataus.

Craftissa ei tule mukana teemoja, eikä siihen juurikaan löydy niitä valmiina. Käytännössä kaikki näkymät ja sivupohjat on tehtävä itse. Periaatteessa tämä ei ole ongelma, koska Twig on yksi parhaimmista kehitetyistä sivupohjamootoreista. Craft on sen verran pitkälle jalostettu, että se osaa automaattisesti hakea kaikki teematiedostot sekä ottaa ne käyttöön kansiorakenteesta.

### 4.3.6 Yhteenveto

World Water Hubin perustoiminnallisuuden voisi toteuttaa melko vaivattomasti, mutta kaikille vaativammille ominaisuuksille, kuten esimerkiksi säätiedoille pitäisi toteuttaa jokin oma rajapinta. Myös laatuparametrien hallinta pitäisi toteuttaa itse. Valmiiden ominaisuuksien puute tekee toteuttamisen Craftille resursseja kuluttavaksi. Vaatimuksiltaan Craft toimii kaikissa yleisimmissä palvelinympäristöissä ongelmitta, joten tästä ei aiheudu ylimääräisiä kustannuksia.

World Water Hubia varten lisensseistä tuli ottaa PRO-versio, joka maksaa 299 dollaria. Hinta on kuitenkin sen verran matala, että se ei ole ongelma. Tätä työtä tehtäessä käytettiin Craftin personal-versiota, joten ei voida sanoa varmaksi onko Craftin käyttäjienhallintaan ja oikeuksiin liittyvät ominaisuudet varmasti tarpeeksi kattavalla tasolla.

Ongelmatilanteissa apua voi olla hankala saada, sillä kehittäjäverkosto ei ole kovin suuri. Tämä voi tuotantoympäristöissä luoda suuriakin ongelmia.

Jatkokehitys saattaa olla Craftille työlästä, koska lähdekoodista ei ole löydettävissä juuri-kaan olemassa olevia esimerkkejä vaan ainoastaan pelkät ohjelmoinnin yleislinjaukset dokumentaatiossa.

Hakukoneoptimointia ei ole huomioitu järjestelmässä itsessään, mutta Craftille ollaan julkaisemassa omaa hakukoneoptimointi-lisäosaa (Sprout 2014). Craftin hallintapaneeli on perusominaisuuksillaan helppo omaksua ja käyttää. Ominaisuuksien lisääntyessä epäilen pystyykö järjestelmä nykyisellä käyttöliittymällään sisältämään niin paljon ominaisuuksia.

Craftin suorituskyky ja tietoturva ovat korkealla tasolla. Tietoturva riskejä ei käytännössä ole, jos sivuston hallinta pidetään erillään itse sivustosta ja kansiodien oikeudet kunnossa.

## 4.4 KeystoneJS

### 4.4.1 Yleistä

KeystoneJS on sisällönhallintajärjestelmä ja sovelluskehitysalusta, joka on rakennettu Express.js- ja Node.js-kirjastoilla. Tietokantana KeystoneJS käyttää MongoDB -tietokantaa ja Mongoose -työkalua skeeman hallintaan.

Keystone on suhteellisen tuore projekti, joka on vasta versiossa 0.2.20, jonka on kehittänyt Jed Watson (KeystoneJS 2014).

### 4.4.2 Ominaisuudet

Keystoneen saa halutessaan asennuksen yhteydessä asennettua tuen blogille, kuvagallerialle ja yhteydenottolomakkeelle. Näillä valmiiksi asennetuilla moduuleilla saa selkeämmän käsityksen järjestelmän toimimisesta ja lähdekoodista.

Poikkeuksellisesti Keystone sisältää oman WYSIWYG-editorin (ks. kuvio 14), joka on uniikki vain Keystoneessa, toisin kuin Concreten TinyMCE -oletuseditori ja WordPressin räätälöity TinyMCE.

Käyttäjähallinta on oletuksena hyvin pelkistetty, järjestelmään voi lisätä käyttäjiä, joita on käytännössä kahta tyyppiä. Kirjautunut käyttäjä, joka voi käyttää hallintapaneelia sekä käyttäjä, joka voi vain kirjautua.

Kuvio 13. Uuden artikkelin lisääminen Keystoneilla. Kuvassa näkyvissä myös Bootstrap 3.0 -sovelluskehikseen pohjautuva käyttöliittymä

#### 4.4.3 Vaatimukset

Keystone vaatii toimiakseen Node.js-kirjastokokoelman ja MongoDB -tietokannan.

Keystone on tämän opinnäytetyön ainoa järjestelmä, jonka lähdekoodin joutuu itse kääntämään. Keystone ei sisällä vielä varsinaista asennusohjelmaa. Asennus on kuitenkin hyvinkin yksinkertainen ja onnistuu npm -paketinhallintatyökalulla ja Yeoman -generaattorilla.

Keystonen asennuksen (ks. kuvio 14) jälkeen järjestelmän asetukset tulee määritellä itse käyttöä varten. Järjestelmän asetuksien määrittelyyn kuuluu muun muassa MongoDB:n asetukset sekä yhteiskäyttöasetukset. Keystone ei ota kantaa siihen mitä sivupohjielua projektissa käytetään. Valitsin tähän työhön Jaden, koska se kuuluu yleisimmin käytettyihin sivupohjamootteihin.



```

Welcome to KeystoneJS.

[?] What is the name of your project? World Water Hub
[?] Would you like to include a Blog? Yes
[?] Would you like to include an Image Gallery? Yes
[?] Would you like to include a Contact Form? Yes
[?] Would you like to include gulp or grunt? (enter: gulp / grunt / default: none)
[?] Would you like to include gulp or grunt? (enter: gulp / grunt / default: none)
e) gulp
-----
[?] -----
KeystoneJS integrates with Mandrill (from Mailchimp) for email sending.
Mandrill accounts are free for up to 12k emails per month.
Would you like to include Email configuration in your project? (Y/n)

```

Kuvio 14. Asennus

#### 4.4.4 Lisäosat

Keystone.js ei sisällä tukea asennettaville lisäosille, eikä niitä juurikaan löydy valmiina. Keystone:ssa ei ole valmista rajapintaa, jolla lisäosia voisi toteuttaa. Kaikki ominaisuudet ja moduulit tulee toteuttaa itse.

#### 4.4.5 Jatkokehitys

Keystone on oletusasennuksellaan melko raakile sisällönhallintajärjestelmänä, mutta sovelluskehitysalustana se on ketterämpi. Keystone:n lähdekoodi on hyvin modulaarista ja kattavasti kommentoitua.

Kaikesta huolimatta Keystone tarjoaa hyvin vähän valmiita työkaluja sillä sovellus ei sisällä valmiita rajapintoja tai funktioita räätälöidyn sisällön toteutukseen, kuten esimerkiksi Concrete5 tai WordPress.

Jokaisella moduulilla tulee olla oma malli (model), reititys ja sivupohjat. DAL (Data Access Layer) on erillään front-endistä, eli ohjelmoitavaa riittää ihan perustoiminnallisuuttakin lisätessä.

MongoDB -tietokanta ja sen Mongoose -skeema on kattavasti dokumentoitu. Tietokanta sisältää valmiiksi luotuja tietotyyppejä normaalien tietotyyppien lisäksi joita sovelluskehityksessä voi hyödyntää, kuten esimerkiksi sijainnin koordinaatit.

#### 4.4.6 Yhteenveto

Keystone ei sovellu World Water Hubin alustaksi, sillä olemassa olevia ominaisuuksia on liian vähän. Myöskään olemassa olevia ostettavia lisäosia ei juurikaan ole. Keystoneissa ei ole huomioitu hakukoneoptimointia ja hallintapaneelin käytettävyys on melko alkeellisella tasolla.

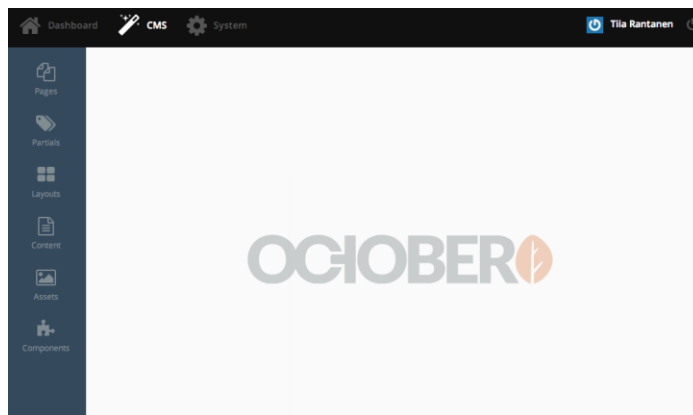
Keystone on nopea ja sen tietoturva hyvällä tasolla. Käytetyt tekniikat ovat tällä hetkellä vielä niin uusia, ettei niistä juurikaan ole etsitty olemassa olevia heikkouksia.

Järjestelmään ei kuulu omaa asennusohjelmaa, joten sen asentaminen eri ympäristöihin, kuten esimerkiksi Windows-palvelimelle saattaa olla työlästä.

### 4.5 October

#### 4.5.1 Yleistä

October on PHP-pohjainen avoimen lähdekoodin sisällönhallintajärjestelmä, joka pohjautuu Laravel PHP -sovelluskehikseen. October on toukokuussa 2014 beta-testaukseen julkaistu hyvin tuore projekti (OctoberCMS 2014).



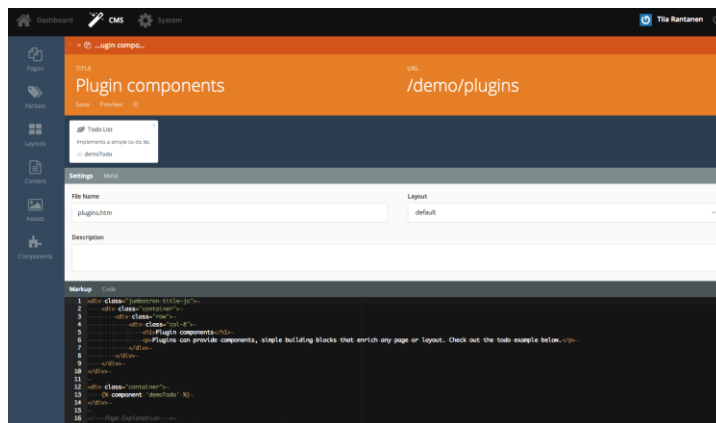
Kuvio 15. Hallintapaneeli

Octoberin tarkoitus on yksinkertaistaa sovelluskehitystä ja tehdä kehityksestä ohjelmoijille helpompaa palaamalla perusteisiin. Järjestelmän käyttöliittymä ja visuaalisuus on huomattavasti muita järjestelmiä parempi, sillä October on selkeä ja nykyaikainen (ks. kuvio 15). Octoberin käyttökohdetta ei ole rajoitettu, se skaalautuu pienestä verkkosivustosta tarvittaessa suurimpiin verkkopalveluihin.

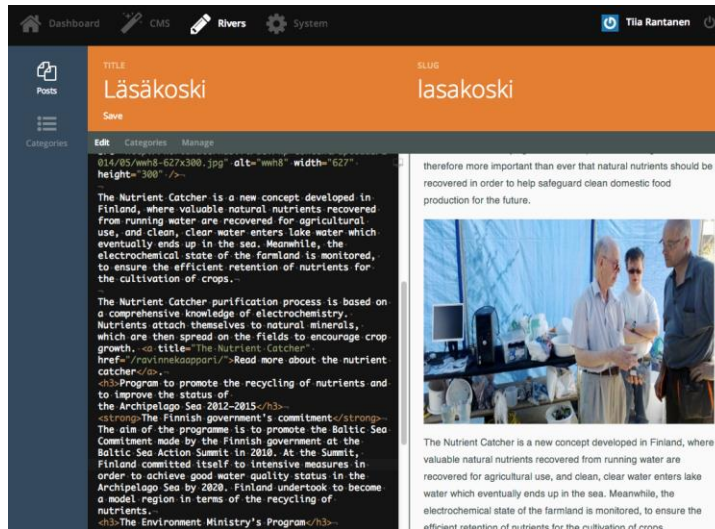
Järjestelmän pääkehittäjiä ovat Alexey Bobkov ja Samuel Georges (October 2014).

## 4.5.2 Ominaisuudet

October ei ole vielä kovinkaan kypsä projekti, sisältäen oletusasennuksella pelkästään lisättäviä sivuja. Ilman ohjelmointitaitoja järjestelmää ei voi käyttää, koska mukana ei ole esimerkiksi visuaalista tekstieditoria vaan ainoastaan reaaliaikainen koodieditorin (ks. kuvio 16 ja 17). Koodieditoria ei tule kuitenkaan aliarvioida kehittäjän näkökulmasta, sillä se on tehokas työkalu, jossa voidaan tallentaa esimerkiksi Twig- tai PHP -syntaksia.



Kuvio 16. Octoberin koodieditori



Kuvio 17. Julkaisun editointinäkymä, jossa nähdään HTML ja taitettu HTML

Sivuihin voidaan lisätä komponentteja (components), joilla sivuun saadaan erilaista toiminnallisuutta. Komponentteja voi kehittää itse tai ladata lisäosina (plugins). Järjestelmää voidaan laajentaa lisäosilla, jotka noudattavat Laravelin MVC-mallin mukaista syntaksia, joka tekee jatkokehityksestä helppoa.

October sisältää sisäänrakennetun AJAX-sovelluskehityksen, jolla AJAX-pyyntöjä voi sitoa helposti sivupohjiin ja käsitellä komponenteissa tai kirjoittaa käsittely suoraan koodieditoriin samalla kun muokkaa sivua.

Octoberin hallintapaneelin etusivulle voi ladata tai tehdä itse vimpaimia (widgets), jotka näyttävät esimerkiksi palvelimen tilan ja muistin käytön. Hallintapaneeli käyttää useimpiin toimintoihin drag & drop -toiminnallisuutta, jolla lisäosia voidaan lisätä sivuille hiirellä vetämällä.

### 4.5.3 Vaatimukset

Octoberin toteutuksessa on otettu moderni ote, jonka vuoksi October toimii vain ympäristössä, jossa on vähintään PHP 5.4, PHP PDO-tietokantarajapinta sekä cURL-, MCRYPT-, ZipArchive- ja GD -laajennukset (Github OctoberCMS 2014).

Tietokannaksi sopii MySQL, SQLite, PostgreSQL tai SQL Server (Laravel Documentation 2014).

#### 4.5.4 Lisäosat

Octoberiin on toteutettu noin 40 erilaista lisäosaa, joilla käyttöön saa esimerkiksi visuaalisen tekstieditorin, gallerian tai yksinkertaisen käyttäjähallinnan. Jokainen lisäosa sisältää kattavan dokumentaation, jotta lisäosia on helppo muokata omiin tarkoituksiin sopiviksi. Lisäosien asentaminen on helppoa, sillä omasta asennuksesta voi luoda Octoberin sivustolla oman projektin, jolloin lisäosat asentuvat hallintapaneelin kautta itsestään.

Tällä hetkellä lisäosia oli niin maksullisia kuin ilmaisiaikin, mutta jatkossa suunnitteilla on kauppapaikka maksullisille lisäosille, jossa lisäosia myydään minimissään 10 dollaria kappale (October Blog 2014).

#### 4.5.5 Jatkokehitys

Järjestelmän teemat koostuvat html-tiedostoista (layout), jotka pohjautuvat Twig-sivupohja moottoriin. Sivupohjat koostuvat kolmesta osasta, asetuksista, PHP-koodista ja Twig-sivupohjan tulostuksesta. Asetukset sisältävät sivupohjan parametrit. PHP -koodi voi sisältää lähes mitä tahansa koodia, joka halutaan suorittaa ennen sivupohjan tulostamista. Sivupohjia voidaan muokata suoraan hallintapaneelin koodieditorilla tai millä tahansa muulla editorilla. Sisällöt tallennetaan suoraan tiedostoihin kansiorakenteeseen, mikä helpottaa myös projektin versionhallintaa ja tietokannan kuormitusta. Octoberin kanssa ei välttämättä tarvita välimuistia, vaikka jotkin lisäosat käyttävätkin dynaamista sisältöä.

Teemat voidaan myös koostaa sivupohjan palasista (partials), jotka voivat sisältää lähes mitä tahansa ohjelmointilogiikkaa. Palasia voidaan lisätä sivuille suoraan hallintapaneelin kautta. Vaihtoehtoisesti voidaan käyttää sisältölohkoja (content block), jotka ovat toiminnallisuudeltaan huomattavasti yksinkertaisempia, koska ne voivat olla vain HTML-

txt (teksti)- tai md (markdown)-tiedostoja. Sisältölohkoja ei pysty käsittelemään hallintapaneelin kautta vaan ne tulee kutsua ohjelmallisesti suoraan sivuun, teema-tiedostoihin (layout) tai sivupohjan palaseen (partial).

Komponentit ovat kustomoitavia elementtejä, jotka useasti sisältävät omat asetuksensa. Käytännössä komponentti vastaa asennettua lisäosaa. Komponentteja voidaan lisätä hallintapaneelistä sivulle tai teema-tiedostoon. Komponenttien voivat esimerkiksi tulostaa HTML-sisältöä, käsitellä AJAX-pyyntöjä, käsitellä lomaketietoa ja määrittää sivupohjien tulostusjärjestyksen.

Octoberiin kuuluva sisäänrakennettua AJAX-rajapintaa voidaan käyttää kahdella eri tavalla. Ensimmäinen vaihtoehto on käyttää data-attribuutteja, jolloin Ajaxin käyttöön ei tarvitse kirjoittaa JavaScriptia, vaan funktiot sijoitetaan teeman asetuksiin PHP:lla. Toinen vaihtoehto on käyttää tehokkaampaa JavaScript-rajapintaa, jossa datan käsittely ja kutsujen tekeminen on huomattavasti vapaampaa. Octoberin Ajax-rajapinta on huomattavasti kattavampi ja modulaarisempi kuin esimerkiksi WordPressin sisäänrakennettu AJAX-luokka.

#### **4.5.6 Yhteenveto**

Octoberin käytettävyys oli erikoista ja innovatiivista. Tämä saattaa olla kaksiteräinen miekka siltä osin, että käyttäjä ei välttämättä sisäistä toiminnallisuutta kovinkaan äkkiä vaan se vaatii totuttelua. Octoberin hallintapaneeli oli näistä järjestelmistä miellyttävin ja kiinnostavin. Hakukoneoptimointia ei ole viimeistelty, jokaiselle julkaisulle voidaan laittaa meta-tietoina vain kuvaus ja otsikko.

Octoberilla voitaisiin toteuttaa kaikki World Water Hubin perusominaisuudet. Joitakin lisäominaisuuksia saataisiin ostamalla lisäosia, mutta kuitenkin suuri osa toiminnallisuudesta tuli toteuttaa itse. Suurin puute lienee lokalisointi, jonka toteuttaminen voi olla hyvinkin työlästä.

Jatkokehitys on nopeaa sisäistää, mutta dokumentaatio tuntuu paikka paikoin pintapuoliselta. Myöskin kehitysyhteisö on tällä hetkellä vielä suhteellisen pieni, joten tuotantoverkkopalveluiden ongelmatilanteissa apua ei välttämättä saa heti.

October toimii yleisimmissä palvelinympäristöissä ja lisäksi Laravel tuo mukavaa vapautta tietokannan valinnassa. Octoberin suorituskyky on hyvällä tasolla, eikä siinä ole havaittavia pullonkauloja. Järjestelmä on vielä niin uusi, ettei sillä ole havaittuja tietoturva-aukkoja.

## **4.6 Statamic**

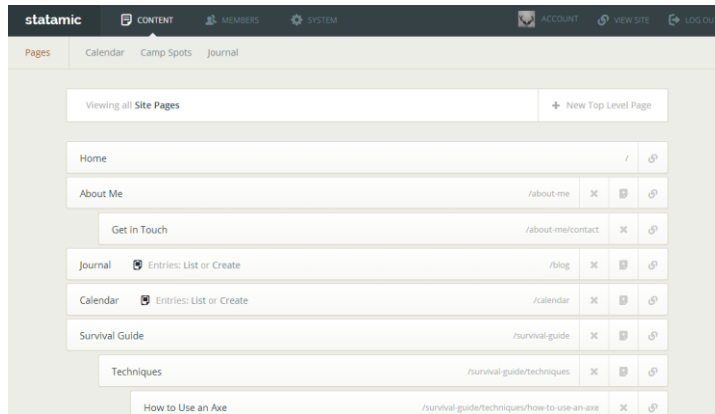
### **4.6.1 Yleistä**

Statamic on flat-file -sisällönhallintajärjestelmä, joka on rakennettu PHP -versiolla 5.3. Statamicin kehittäjinä toimii entisiä ExpressionEngine kehittäjiä. Statamicin ensimmäinen versio on julkaistu vuonna 2012 (Statamic 2014), joten kyseessä on melko uusi projekti.

Tämä järjestelmä on ainut maksullinen tämän työn järjestelmistä. Statamicilla on kahta erilaista lisenssiä, ammattikäyttöön tarkoitettu kaupallinen 99 dollarin lisenssi sekä yksityiskäyttöön tarkoitettu henkilökohtainen 29 dollarin lisenssi (Statamic Store 2014). Lisenssit olivat domain-kohtaisia. Molemmilla lisensseillä sai käyttöön samat ominaisuudet.

### **4.6.2 Ominaisuudet**

Statamicin hallintapaneelin käyttöliittymä on selkeä ja nopea oppia, ja koska se on responsiivinen, voi sitä käyttää myös mobiililaitteella optimoidulla käyttöliittymällä.



Kuvio 18. Statamicin hallintapaneeli

Vaikka Statamic on maksullinen, on siinä hyvä huomioida, että rahalla saa jatkuvasti kehittyvän sovelluksen ja hyvän käytön tuen. Koska palvelusta maksetaan, on myös lisäosien laatu kohdallaan.

Statamic ei ole static-site -generaattori vaikka se ei tietokantaa käytäkään. Järjestelmä tallentaa dataa tiedostoina suoraan kansiorakenteeseen, jolloin jokainen tiedosto vastaa julkaisua järjestelmässä. Tämä on etu esimerkiksi versiohallinnassa sekä nopeassa vasteajassa. Käytännössä tiedostoja voitaisiin jakaa ja synkronoida pilvestä, jolloin päivittäminen on todella helppoa.

Statamicissa tulee mukana myös hallintapaneeli, vaikka sivuja voidaankin päivittää suoraan millä tahansa tekstieditorilla. Statamicin hallintapaneeliin kuuluu yli 20 erilaista tietotyyppiä sisällönhallintaan, joten hallintapaneeli on ihan kätevä, mikäli ei muista näitä kaikkia ulkoa.

Järjestelmän asetuksia ei voi muuttaa hallintapaneelistä, vaan asetukset, reititys ja kehitysoasetukset, kuten debuggaus, löytyvät erillisestä YAML-tiedostosta.

Statamic on näistä järjestelmistä ainut, jossa ei ole julkaisujen kommentointimahdollisuutta. Statamicilla ei myöskään voi hallita suuria käyttäjäryhmiä.



Järjestelmän päivittämiseen ei ole valmista työkalua, vaan Statamic tulee päivittää korvaamalla vanhat lähdetiedostot uusilla, jotka saa ladattua Statamicin nettisivuilta kirjautumalla sisään, tämä on kuitenkin nopea operaatio.

Järjestelmässä on sisäänrakennetut lokit, jotta nähdään, mikäli sivustossa on virhetilanteita. Samalla nähdään myös onko tiedosto-oikeudet oikein ja käyttäjien salasanat kryptattu.

Statamicin tietoturva muihin järjestelmiin nähden huomattavasti paremmalla tasolla, koska järjestelmä ei sisällä murrettavaa tietokantaa. Mikäli tiedostorakenteen oikeudet ovat oikein, on murtautuminen palvelimelle Statamicin kautta käytännössä mahdotonta.

### **4.6.3 Vaatimukset**

Statamic vaatii toimiakseen vähintään PHP version 5.3.6 sekä siihen mcrypt -laajennuksen. Statamicin mukana tulee myös testitiedosto, jolla voidaan varmistaa, että ympäristö on sopiva järjestelmälle.

### **4.6.4 Lisäosat**

Statamic sisältää joitakin valmiita lisäosia, mutta pääosin lisäosat keskittyvät sosiaaliseen mediaan, joten niiden hyöty jää kehittäjälle kovin löyhäksi siinä suhteessa, että some-jaot yleisimpiin palveluihin on helppoa toteuttaa myös itse. Statamicin lisäosista ei vielä löydy mitään mullistavia laajennuksia. Lisäosat ovat tasalaatuisia ja ne tarkistetaan mahdollisten tietoturva-aukkojen varalta. Kolmansien osapuolten maksullisissa lisäosissa on jonkin verran valinnanvapautta. Tarjolla on muun muassa Bison-verkkokauppalaajennus.

### **4.6.5 Jatkokehitys**

Statamicin tiedostot muodostuvat YAMLista ja Markdownista. YAMLia käytetään esimerkiksi Statamicin sisäisten, lisäosien ja teemojen asetuksien ja reitityksen määrittämiseen. YAMLilla voidaan tehdä myös globaaleja muuttujia, joita voidaan käyttää tageina

teemoissa. Samantyyllisesti kuin Octoberissa teemojen toteuttamiseen ja komponenttien lisäämiseen käytetään erilaisia aaltosuluilla ympäröityjä tageja. Jokainen sisältötiedosto alkaa loholla, joka koostuu YAMListasta. Tähän lohkoon voidaan määrittää myös tageja sekä muita asetuksia.

#### **4.6.6 Yhteenveto**

Statamicin suurin puute on käyttäjien oikeuksien hallinta, tämä käytännössä tarkoittaa sitä, että World Water Hubia varten tämä ominaisuus tulisi toteuttaa itse. Tästä saattaa tulla yllättävän työlästä. Myöskään julkaisujen kommentointimahdollisuutta Statamicissa ei ole. Statamic ei oletuksena sisällä juurikaan mitään ominaisuuksia mitä tarvittaisiin, räätälöityä sisältöä saa toteutettua suhteellisen helposti, mutta Statamicin ongelmana on puutteellinen dokumentaatio, jota ei ole helppo sisäistää.

Statamicin kaupallinen versio maksaa 99 dollaria. Pääosin myös kaikki lisäosat ovat maksullisia. Näistä saattaa äkkiä kerääntyä yllättävän paljon kustannuksia. Hakukoneoptimointia varten Statamiciin saa lisäosan, vaikka järjestelmä ei itsessään sisällä näitä ominaisuuksia.

Järjestelmän hallintapaneeli on miellyttävän näköinen ja helppo käyttää. Ominaisuuksien lisääntyessä hallintapaneelissa piilee ongelma sen selkeydessä.

Suorituskyky ja tietoturva ovat Statamicissa huomattava etu. Ilman tietokantaa ei tietoturva-aukkoja juurikaan ole ja sivuja voi halutessaan hallita paikallisesti ja julkaista erilleen. Käytännössä ainoat tietoturvariskit ovat palvelimeen liittyviä.

## 4.7 WordPress

### 4.7.1 Yleistä

WordPress on ilmainen avoimen lähdekoodin blogialusta ja sisällönhallintajärjestelmä, joka pohjautuu PHP-ohjelmointikielen ja MySQL-tietokantaan. WordPressia käyttää noin joka viides verkkosivusto maailmassa, sivustoja on yhteensä yli 60 miljoonaa (W3 Techs 2014). WordPressistä on tarjolla blogipalvelu, jonne voi perustaa oman sivustonsa ilman omaa palvelinta, sekä ladattava stand-alone versio, jonka voi asentaa itse. WordPressia käyttävät esimerkiksi The New York Times-blogi (WordPress Showcase 2014) ja The New Yorker (WordPress Showcase 2014).

WordPress julkaistiin toukokuun 27, 2003 perustajien Matt Mullenweg ja Mike Littlen toimesta GPLv2 Free Software Foundationin lisenssillä. (Siobhan WcKeown 2014).

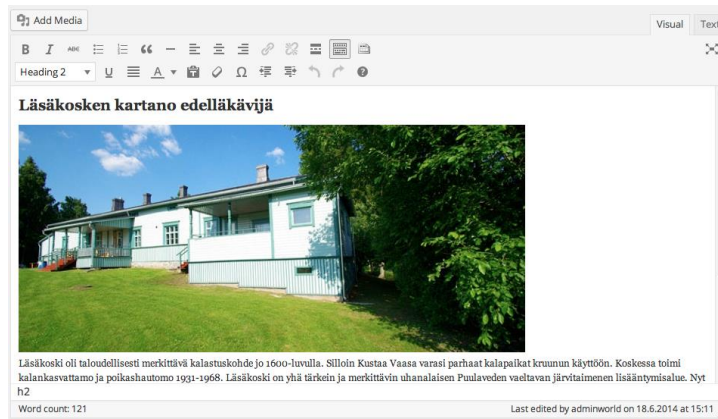
Tulevaisuudessa uusiksi ominaisuuksiksi on tulossa sisäänrakennettu REST-rajapinta (WP Tavern 2014) ja Front-End editointi (Make WordPress UI 2014).

### 4.7.2 Ominaisuudet

WordPress on tunnettu viiden minuutin asennuksesta, joka on selkeä ja ohjattu.

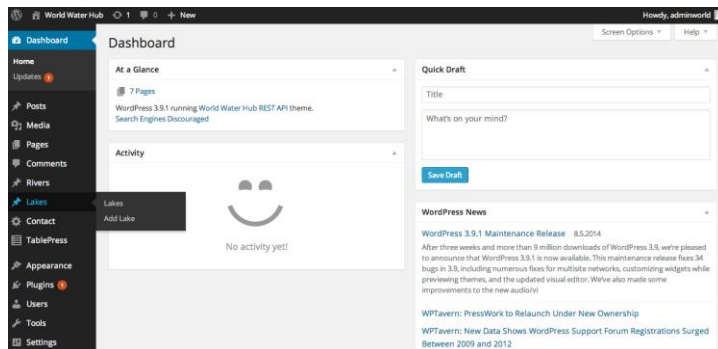
WordPress on kasvattanut huomattavasti sen alkuaikojen blogialusta ajoista. WordPressin suurin mullistus tapahtui versiossa 3.0, jolloin järjestelmään tuli tuki mukautetuille tieto-tyypeille. Tämä mahdollisti, että WordPressillä voidaan toteuttaa laajempia ja skaalautuvampia verkkopalveluita kuin aikaisemmin.

WordPressin eniten erottuvin ominaisuus verrattuna muihin tässä työssä tutkittuihin järjestelmiin oli sen vaikuttavan näköinen hallintapaneeli sekä WYSIWYG-editori (ks. kuvio 19), jossa korostuu käytettävyys. (Sarah Gooding 2014)



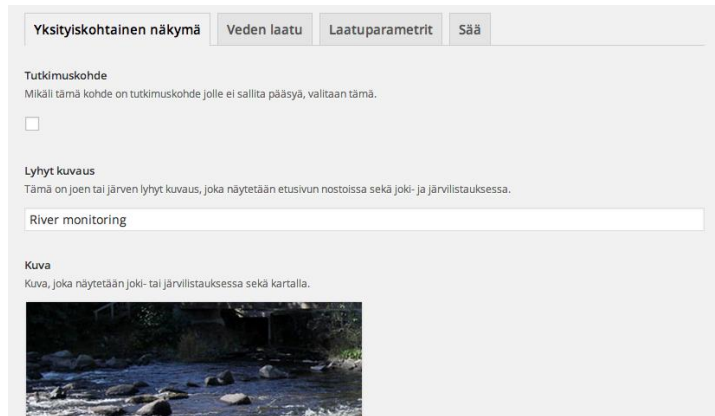
Kuvio 19. WordPress WYSIWYG-editori

WordPressin keskeisiin ominaisuuksiin kuuluvat artikkelit ja sivut, sekä niihin liittyvät kommentointi- ja mediakirjasto-ominaisuudet. WordPress mahdollistaa myös mukautettujen tietotyyppien lisäämisen (ks. kuvio 20), jotka kuitenkin pohjautuivat artikkeleihin tai hierarkiallisiin sivuihin. Jokaisella tietotyypillä on omat taksonomiansa, eli niille voitiin vapaasti luoda tageja ja kategorioita.



Kuvio 20. Hallintapaneeli, jossa vasemmalla on valittuna mukautettu tietotyyppi (custom post type)

Käyttäjä-, artikkeli- ja sivu-metadataa (ks. kuvio 21) pystyy luomaan itse ohjelmallisesti, tähän tarkoitukseen on myös useita lisäosia, joista tunnetuin on Advanced Custom Fields (WordPress Most Popular Plugins 2014). Tällä ominaisuudella WordPressiä voidaan laajentaa hyvinkin pitkälle sisältämään minkälaisia sivuja, artikkeleita tai julkaisuja tahansa.



Kuvio 21. Mukautettua tietosisältöä joki-julkaisun hallinnassa (custom post meta data)

Jokaisella artikkelilla ja sivulla, sekä niiden mukautetuilla tietotyypeillä on kattavasti näkyvyysasetuksia. Julkaisu voi olla julkaistu, luonnos, yksityinen tai salasana-suojattu. Julkaisun näkyvyyden voi myös ajastaa. Julkaisulla on myös automaattinen versiohallinta sekä tallennus tietyin väliajoin, jota ei valitettavasti suoraan hallinnasta saa pois käytöstä. Automaattinen versionhallinta ja tallennus täyttävät tietokantaa hyvinkin nopeasti. Tämä voi pahimmillaan joillain palvelimilla johtaa suorituskyvyn heikkenemiseen.

Käyttäjillä on useita eri rooleja. Oletusrooleja ovat tilaaja, avustaja, kirjoittaja, päätoimittaja ja pääkäyttäjä. Näitä rooleja voi pienellä kustomoinnilla luoda itse lisää ja eri käyttäjäryhmien oikeuksia voidaan rajata tarkemmin, mitä kukin käyttäjäryhmä voi tehdä. Tähän tarkoitukseen on myös tarjolla paljon eri lisäosia. Järjestelmään voidaan määrittää, että pelkästään kirjautuneet käyttäjät voivat kommentoida julkaisuja, tai julkaisujen kommentointi voidaan jättää avoimeksi. Kommenttien hyväksynnän saa halutessaan myös käyttöön, eli pääkäyttäjän tulee hyväksyä kommentti ennen sen julkaisua.

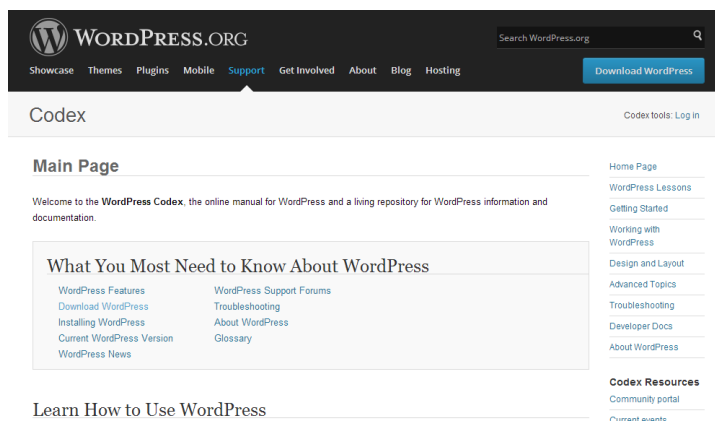
WordPress pystyy itse lataamaan, päivittämään ja poistamaan lisäosia ja teemoja joko palvelimen oikeilla tiedosto-oikeuksilla tai ohjelmallisesti määritetyillä FTP-tunnuksilla, tämä tekee päivittämisestä todella helppoa. Teeman ja lisäosien yhteensopivuus tulee varmistaa ennen päivitystä, jotta vältetään räätälöidystä ohjelmointikoodista johtuvista virheistä lähdekoodissa.

Hakukoneoptimointi on huomioitu esimerkiksi re-writeissa, joiden muodon saa itse päättää vapaasti. Hakukoneoptimointiin on myös useita lisäosia, joilla saa määriteltyä tarkemmin otsikoita ja hakusanoja. Hakukonenäkyvyyden saa myös otettua pois päältä, jolloin hakukoneet eivät indeksoi sivustoa, tämä on toimiva ominaisuus kehitysvaiheessa oleville verkkopalveluille.

WordPressin hallintapaneeli on käännetty yli 70 eri kielelle. Itse sivuston front-endin kääntämiseen oli saatavilla muutama varteenotettava lisäosa. I18n-ominaisuudet on rakennettu gettextin avulla pot- ja po-tiedostoilla, joten kielikäännöksiä oli yksinkertaista tehdä myös itse. Myös omien teemojen ja lisäosien kielikäännökset onnistuvat samalla logiikalla.

WordPressin ympärille on perustettu myös erilaisia vuosittain järjestettäviä epävirallisia tapahtumia kuten WordCampit (WordCamp Central 2014) ja Meetupit. Tukifoorumeilla toimii aktiivisia vapaaehtoisia vastaamassa käyttäjien kysymyksiin. Lisäksi jokaisella lisäosalla on oma tukifooruminsa, josta saa useimmiten tukea ongelmiin.

Järjestelmän dokumentaatio (ks. kuvio 22) on kattava ja selkeä, sisältäen kaikki tarvittavat tiedot laajempaankin sovelluskehitykseen.



Kuvio 22. WordPress Codex eli dokumentaatio kehittäjille

WordPress itsessään voidaan tulkita tietoturvalliseksi. Mikäli siinä huomataan jokin haavoittuvuus, se paikataan aina nopeasti muutaman päivän sisällä. WordPress on avoimen

lähdekoodin sovellus, joten sille voi tehdä lisäosia kuka tahansa. Lisäosa-rajapinta sisältää valtavan määrän hyödynnettäviä funktioita ja sen dokumentaatio on kattava ja selkeä. Kehittäjien kynnyks toteuttaa lisäosia on matala. Monesti tämä aiheuttaa yllättäviä ongelmia. WordPress sisältää valmiin asennustyökalun lisäosia varten, jolloin niiden asentaminen on helppoa ja vaivatonta. Useimmiten lisäosat, varsinkin sellaiset, jotka sisältävät jonkinlaisen tiedostonlataus-rajapinnan, sisältävät itsessään tietoturvaheikkouksia. Heikkouksia sisältäviä lisäosia varten on toteutettu erilaisia automaattisia botteja, jotka skannaavat Internetin WordPress asennukset ja etsivät niistä suoralla URL-osoitteella heikkouksia sisältäviä tiedostoja. Mikäli tällainen lisäosa on asennettu oletushakemistoon, voi botti asentaa palvelimelle vaikka roskapostipalvelun heikkoutta hyödyntämällä.

WordPressin tietoturvaa saa kuitenkin parannettua usealla eri tavalla, näistä tehokkain on teemakansiot ja lisäosat sisältävän wp-content kansion uudelleennimeäminen joksikin muuksi (Wpitude 2014). Näin botit eivät enää löydä etsimiään tiedostoja palvelimelta. Vaikka tämä täyttääkin virhelokeja, on se silti pienempi paha tietoturvavuotoon verrattuna. Palvelimen tiedosto-oikeuksien kanssa tulee olla tarkkana. WordPress vaatii, että tiedostojen ja kansioden omistaja on sama käyttäjä, jolla ajetaan palvelinsovellusta. Tällöin ei tarvitse laittaa kaikkia oikeuksia edes tiedostolatauskansioille. WordPress on myös altis brute-force hyökkäyksille. Tietoturvan parantamiseen on olemassa monia lisäosia, jotka esimerkiksi poistavat brute-force mahdollisuuden sekä suojelevat tärkeimpiä hallintapaneelin tiedostoja.

### **4.7.3 Vaatimukset**

WordPressin ainoat tekniset vaatimukset ovat PHP, jonka tulee olla vähintään versio 5.2.4 ja MySQL, jonka tulee olla vähintään versio 5.0. WordPress tukee siis hyvinkin vanhoja palvelinympäristöjä.

#### 4.7.4 Lisäosat

WordPressin lisäosia on olemassa yli 30 000 kappaletta (WordPress Plugins Directory 2014). Lisäosia löytyy useisiin eri tarkoituksiin kuten muutamina esimerkkeinä verkko-kauppa sekä siihen kuuluvat maksurajapintalaajennukset, käyttäjähallinta, mukautetut sisällöt, erilaisia rajapintoja kuten REST ja LDAP sekä säälajennukset. WordPressin omilta sivuilta löytyvät lisäosat ovat aina ilmaisia, WordPress on avointa lähdekoodia ja olettaa, että myös sen lisäosat julkaistaan samalla lisenssillä, mitään krediittiä ei vaadita. Tästä säännöstä poiketen moni lisäosa toimii freemium-periaatteella. Tästä hyvänä esimerkkinä on esimerkiksi WooCommerce-verkkokauppalisäosa, jonka laajennukset ovat maksullisia.

#### 4.7.5 Jatkokehitys

WordPress sisältää kattavan rajapinnan lisäosien toteuttamiseen ja hallintapaneelin muokkaamiseen. Vaikka WordPress ei sisälläkään minkäänlaista sivupohja-moottoria (template engine), joka tekee WordPressistä aika ajoin todella epäloogisen, tulee hyödyt vastaan siinä, että samat funktiot ja toiminnallisuudet ovat käytettävissä niin sivupohjissa kuin lisäosissa.

Alustan sovelluskehitys pohjautuu MVC-mallin sijaan koukkuihin, joita ovat suodattimet ja toiminnot (filters, actions). Koukuilla voidaan muokata tai ylikirjoittaa WordPressin olemassa olevia funktioita, kuten esimerkiksi `admin_bar_menu`-toiminnon, joka tulostaa järjestelmän ylläpitäjälle selaimen yläreunassa näkyvän valikon tai `the_content`-suodattimen, joka tulostaa sivun tai artikkelin pääsisällön. Toiminnoilla voidaan lisätä sisältöä tai toiminnallisuutta kun taas suodattimilla voidaan muokata olemassa olevaa sisältöä. Koukkuja on versiossa 3.9 olemassa 1650 kappaletta ja niitä voidaan lisätä joko lisäosiin tai teeman sisäisiin funktioihin. Koukuissa piilee myös ongelmia. Käytännössä kaikki räätälöinti tapahtuu koukuilla tai WordPressin omien luokkien ylikirjoittamisella. Koska WordPress ei pohjautu MVC-malliin, on lähdekoodin räätälöinti sekavaa jos pelkät koukut eivät riitä.



WordPressin teemojen kehitys perustuu silmukkaan (The Loop), jota järjestelmä pyörittää. Silmukka sisältää oletuksena avatun sivun tai artikkelin kaikki tiedot. Silmukkaa voidaan muokata tai monistaa tulostamaan mitä sisältöjä tahansa. Silmukka on yksi WordPressin tehokkaimmista työkaluista.

Järjestelmään kuuluu myös sisäänrakennettu Ajax-rajapinta, joka mahdollistaa AJAXin käytön suoraan WordPressin omilla funktioilla. WordPressin teemaan tai lisäosaan tulee määrittää funktioita joita käytetään kutsujen käsittelyyn. Itse funktiot voivat sisältää mitä tahansa ohjelmointilogiikkaa, pois lukien järjestelmän sisäinen ohjelmointilogiikka, joka vaatii tietoturvan vuoksi lisäkustomointia.

WordPressin MySQL-tietokanta koostuu poikkeuksellisen vähäisestä 11 taulusta, joista keskiössä ovat posts ja users taulut. Posts-tauluun tallennetaan kaikkiin julkaisuihin liittyvät perustiedot. Postmeta-taulu sisältää kaikki julkaisuihin liittyvät lisätiedot avain-arvo -pareina. Users-taulu sisältää käyttäjien perustiedot ja usermeta-taulu käyttäjien lisätiedot avain-arvo -pareina. Useasti meta-taulut ovat WordPress-pohjaisissa järjestelmissä suurimman kuormituksen alaisena, koska suurin osa tiedosta on useasti meta- eli lisätietoa.

#### **4.7.6 Yhteenveto**

WordPressillä voidaan toteuttaa pääosa kaikista World Water Hubin ominaisuuksista ilman jatkokehitystä. Pääosin jokaiselle puuttuvalle toiminnallisuudelle löytyy lisäosa, jota järjestelmässä voidaan hyödyntää.

WordPressin kaikki lisäosat ovat pääosin ilmaisia, joten ylimääräisiä kustannuksia ei synny. WordPress ei itsessään sisällä lokalisointivaihtoehtoja, mutta nämä voidaan toteuttaa lisäosalla, joka maksaa 79 dollaria.

WordPressin dokumentaatio on kattava, jokaiselle toiminnolla löytyy hyvinkin pitkälti omat sisäänrakennetut funktiot, joita jatkokehityksessä voidaan hyödyntää. Järjestelmän

suurimpia heikkouksia ovat sivupohjamoottorin ja MVC-mallin mukaisen ohjelmoinnin puuttuminen.

WordPress on vaatimuksiltaan hyvin kevyt. Järjestelmä toimii vanhoissakin palvelinympäristöissä, mutta myös uusissa.

Hakukoneoptimointia ei ole huomioitu WordPressin perusasennuksessa juurikaan millään tavalla. Ainoa hakukoneoptimointia parantava ominaisuus on osoiterakenteen melko vapaa hallinta sekä sivustolle asetettava otsikko ja kuvaus. Hakukoneoptimointia voidaan parantaa erilaisilla asennettavilla lisäosilla. WordPressin käytettävyys ja hallintapaneelin selkeys ovat ensiluokkaisella tasolla, vaikka välillä tuntuu, että käyttöliittymän ominaisuuksista on tingitty ulkonäön vuoksi.

WordPressin suurimpia ongelmia ovat sen suorituskyky ja tietoturva. WordPress on niin tunnettu järjestelmä, että sen versioiden heikkoudet ovat löydettävissä helposti Internetistä. Myöskin lisäosien lähdekoodeissa huomaa huolimattomuutta erityisesti tiedostonlatausrajapintojen toteuttamisessa. WordPressin tietokanta on yksi sen suurimpia heikkouksia. WordPress tallentaa dataa hyvin keskitetysti aiheuttaen kovaakin kuormitusta tietokannalle. Tähän ongelmaan on omat lisäosansa, jolla voidaan toteuttaa erilaista välimuistin hallintaa, jotta tietokantakyselyiden määrää voitaisiin laskea.

## 5 Tulokset ja pohdinta

### 5.1 Järjestelmien vertailun yhteenveto

Taulukossa 2 (sivu 56) näkyy eri järjestelmien sijoittuminen vertailussa. Järjestyksessä vahvimasta heikoimpaan näillä mittareilla mitattuna tulos oli seuraava: Craft, Statamic, WordPress, Concrete5, October ja KeystoneJS. Tärkeimpiä mittareita pisteytyksen kannalta olivat suorituskyky ja nopeus sekä tietoturva.

Taulukko 2. Vertailun tulokset

Järjestelmä tai alusta	Craft	Statamic	WordPress	Concrete5	October	KeystoneJS	Maksimi
Epäsuora hinta	50	10	100	30	10	100	100
Kehittäjäverkosto	240	90	240	120	120	90	300
Ohjeet ja tuki	140	80	180	240	120	80	200
Jatkokehitys	270	240	270	270	90	30	300
Vaatimukset ja tekniikat	270	300	240	240	300	270	300
Hakukoneoptimointi	40	400	200	240	180	40	400
Käytettävyys	320	200	360	140	160	120	400
Suorituskyky ja nopeus	500	450	200	400	400	450	500
Tietoturva	500	500	250	350	450	450	500
<b>Yhteensä</b>	<b>2330</b>	<b>2270</b>	<b>2040</b>	<b>2030</b>	<b>1830</b>	<b>1630</b>	<b>3000</b>

KeystoneJS jäi näillä kriteereillä mitattuna viimeiseksi. KeystoneJS on kaikista tässä työssä verratuista projekteista nuorin ja vähiten kypsä. Keystone ei sisällä riittävästi ominaisuuksia, jotta sen voisi ottaa sellaisenaan muukaan asiakasprojektiin. KeystoneJS vaatii huomattavan määrän jatkokehitystä. Tämä voi olla JavaScript-pohjaisessa ympäristössä työlästä, sillä se ei ole vielä niin suosittu tekniikka kuin vanhempi PHP ja sen palvelintoteutukset. Siinä missä Keystone vetää vertoja muille järjestelmille on kaikesta huolimatta sen suorituskyky ja laajennettavuus.

Octoberin sijoitus pohjautuu siihen, että se on hyvinkin pitkälti kehittäjille suunnattu järjestelmä, World Water Hub-verkkopalvelun toteuttaminen Octoberilla on käytännössä mahdotonta. Asiakas vaatii helposti ymmärrettävän käyttöliittymän ja Octoberissa ei oletuksena ole edes visuaalista editoria. Octoberin ominaisuudet ovat tälle projektille puutteelliset ja ominaisuuksien toteuttaminen vaatii huomattavaa jatkokehitystä, vaikkakin Octoberin käyttämä Laravel-sovelluskehys on yksi tämän hetken suosituimmista PHP-kehyksistä ja helposti jatkokehitettävissä. Yksi Octoberin vahvuuksista on Laravelin mukainen abstrakti tietokantarajapinta, vaikka se ei Octoberin sijoitusta parannakaan.

Concrete5 jäi tuloksissa keskiverroksi, koska esimerkiksi tietoturvan saralla Concreten on hankala kilpailla järjestelmien kanssa, jotka eivät käytä tietokantaa ollenkaan. Concre-

te5:n käytettävyyttä verotti huomattavasti vanhentunut käyttöliittymä, joka olisi pakko uusia, mikäli järjestelmää käytettäisiin asiakasprojekteissa. Jatkokehitys oli yksi Concreten vahvuuksista, sillä dokumentointi oli kattavaa ja kehitys MVC-mallin mukaisesti järjestyksellistä. Hakukoneoptimointia ei mielestäni huomioitu riittävästi, se on kuitenkin tärkeä osa sivuston näkyvyyttä.

WordPressin sijoittuminen kolmanneksi pohjautuu sen laajennettavuuteen ja suureen lisäosien määrään. WordPress sisältää oletuspohjaisesti valtaosan kaikista ominaisuuksista, jotka WWW-verkkopalveluun tarvitaan, ainoastaan muutama ominaisuus joudutaan toteuttamaan käsin. WordPressin jatkokehitys ei kattavasta dokumentoinnista huolimatta ole normaalin ohjelmointilogiikan mukaista ja teemoista tuleekin helposti PHP:n ja HTML:n sekoituksia, jotka ovat todella vaikea lukuisia ja haluttua toiminnallisuutta ei löydy teematiedostojen viidakosta. WordPressin tietoturva on sen suosioista johtuen muihin järjestelmiin verrattuna huonompi. WordPressin suorituskykykin monesti kärsii lisäosien määrästä ja niiden huonosta toteutuksesta. WordPressillä voidaan toteuttaa suurehkojakin järjestelmiä nopealla aikataululla ja pienillä resursseilla, mutta kyse pääosin on kuitenkin aina kertaluontoisista projekteista, joita ei enää jatkossa juurikaan kehitetä, vaan koko sivusto uusitaan muutaman vuoden välein.

Statamic päätyi sijalle kaksi, johtuen sen hyvästä käytettävyydestä ja erinomaisesta tietoturvasta. Statamic on yksi tämän vuoden kovimmassa nosteessa olevista järjestelmistä. Statamic on loistava valinta asiakasprojekteihin pienempiä sivustoja varten. Mikäli WWW toteutettaisiin Statamicilla, vaatisi se jonkin verran räätälöintiä ja jatkokehitystä.

Craft on tämän opinnäytetyön varmin valinta verkkopalvelua varten, mikäli oletettaisiin, että projektilla olisi käytettävissä riittävästi resursseja. Craftin huippuunsa viritetty suorituskyky on ihan eri luokkaa kuin muissa työssä vertailuissa järjestelmissä. Nopeutta voidaan lisätä vielä välimuistilla, jolloin vasteajat putoavat melko lähelle nollaa. Vaikka Craft onkin jo versiossa 2.1, on se osittain vielä keskeneräinen. Myös Craftin käyttäminen vaatisi jonkin verran jatkokehitystä erityisesti Google Maps- ja muissa rajapinta integraatioissa.

## 5.2 Toteutuksesta

Lopulta resurssien rajallisuuden ja nopean aikataulutuksen vuoksi World Water Hub-verkkopalvelu toteutettiin WordPress-sisällönhallintajärjestelmällä. WordPress tarjosi riittävän hyvän kompromissin suorituskyvyn ja sisäänrakennettujen ominaisuuksien välimaastosta.

WWH:n käyttöliittymä on rakennettu käyttäen Semantic UI HTML5-sovelluskehystä ja sen tarjoamaa responsiivista gridiä pienillä muutoksilla. Laatuparametrien ja muiden ulkopuolisten tietojen siirto järjestelmään on toteutettu REST-rajapinnan avulla. Muuta sisältöä hallinnoidaan WordPressin hallintapaneelin kautta. WordPressiin saatiin suhteellisen helposti automatisoitua jopa useat Twitter-tilit sekä riistakameroiden kuvat, jotka lähettävät kuvia muutaman kerran päivässä sähköpostilla.

Järjestelmään toteutettiin joki- ja järvi-sisältötyypit, sekä niille omat muokattavat sisältökenttensä. Järjestelmä sisältää taulukoita varten oman lisäosansa, jolla taulukoita voidaan muokata helposti visuaalisella editorilla. Taulukot ovat aina samanmuotoisia, jotta niistä voidaan piirtää sivustolla käytettävät ympyrädiagrammit.

Verkkopalvelussa on Google Maps-integraatio, johon piirretään kaikkien jokien ja järvien sijainnit, sekä niiden lyhyet kuvaukset ja painikkeet, joista voidaan siirtyä kohteen tarkempiin tietoihin. Sää tiedot tuodaan lisäosalla karttaa varten syötettyjen koordinaattien avulla.

Järjestelmässä on käytössä W3 Total Cache lisäosa vähentämään tietokantapyyntöjä ja sivuston vasteaikoja. Jatkossa mikäli sivustolle tulee kirjautuneita käyttäjiä, joiden sisältö rajataan oikeuksien mukaan, voidaan harkita pirstaloitua välimuistia tai Varnishia.

## 5.3 Pohdinta

Työn tavoitteena oli valita soveltuva sisällönhallintajärjestelmä tai alusta, jonka päälle verkkopalvelu World Water Hub rakennettiin. Tavoitteena oli myös käydä läpi verkko-

palvelun vaatimukset, ja perustella eri järjestelmien soveltuvuus vaatimusten ja ennalta määriteltyjen kriteerien pohjalta. Työtä aloittaessa yhtenä työn motiivina oli myös, että tämän työn pohjalta joitakin järjestelmiä voitaisiin käyttää myös tulevilla projekteilla toimeksiantajayrityksessä.

Työn tuloksena saatiin toimeksiantajaa helpottava selvitys nykypäivän sisällönhallinta-järjestelmätarjonnasta sekä niiden perusominaisuuksista ja jatkokehittämisestä. Lisäksi saatiin kustannustehokkain menetelmä toteuttaa toimeksiantajana ollut World Water Hub -verkkopalvelu.

Tämän opinnäytetyön avulla saatiin toimeksiantajan näkökulmasta tehty vertailu eri sisällönhallintajärjestelmistä peilaten ja painottaen valintakriteerejä juuri tämän hetkisille markkinoille ja asiakkaiden yleisesti vaatimiin ominaisuuksiin, jotka ovat tällä hetkellä tarpeellisia. Käytännössä tämä tarkoittaa sitä, että tehty vertailu on osittain puolueellinen eikä vastaa yleistä näkökulmaa. Toimeksiantajalle työ on hyödyllinen, mutta ei välttämättä muille yrityksille riippuen heidän harjoittamastaan liiketoiminnasta ja strategias- ta. Tämä työ painottuu vain tällä hetkellä trendikkäimpiin PHP- ja JavaScript – ohjelmoin- tikielillä toimiviin järjestelmiin sulkien pois esimerkiksi useissa suurissa verkkokauppa- teutuksissa käytetyn Java EE:n ja siihen pohjautuvat suuremmat järjestelmät kuten Magnolia ja .Net -kieleen pohjautuvan EPiServer -järjestelmän.

Valitut järjestelmät ovat toisistaan hyvinkin paljon poikkeavia. Tämä asettaa vertailut järjestelmät eriarvoiseen asemaan, koska ominaisuudet ja käyttötarkoitukset eriävät niin suuresti toisistaan. Rärkeimmin eron huomasi tietokannallisissa ja tietokannattomissa järjestelmissä. Uskon, että flat-file -järjestelmät tulevat yleistymään niiden erinomaisen suorituskyvyn ansiosta tietomäärien verkossa moninkertaistuessa. Tämä luo paljon paineita kehittäjille, jotka kehittävät julkaisujärjestelmiä, mutta myös niille, jotka kehittävät näiden järjestelmien avulla verkkopalveluita. Nykypäivän vaatimukset monimutkaistuvat kokoajan. Pelkkä perustoiminnallisuus ei enää riitä edes yksinkertaiselle verkkosivustolle saati sitten isompiin palveluihin.

Työssä vertailut järjestelmät ovat kohderyhmältään ja soveltuvuudeltaan eritasoisia. Siinä missä WordPress painottaa käyttäjää ja helppokäyttöisyyttä, muut järjestelmät keskittävät viestintänsä kehittäjille, jotka palveluita luovat. Tämä aika pitkälti selittää eroja siinä, miten WordPress on niin paljon käytetympi kuin saman koko luokan sisällönhallintajärjestelmät. Tilaajat ja asiakkaat tietävät WordPressin jo entuudestaan ja osaatkin vaatia sitä jo suoraan. Muita järjestelmiä he eivät tunne, kun pääosin jokaisen järjestelmän verkkosivut ovat toteutettu niin, että peruskäyttäjä ei ymmärrä niistä mitään eikä osaa lukea teknistä sisältöä.

Työn tekeminen tarjosi hyvän mahdollisuuden tutkia useita järjestelmiä ja sitä, että löytyykö tulevaisuudessa WordPressille haastajaa ja milloin tämä mahdollisesti voisi tapahtua kehittäjän näkökulmasta. Tällä hetkellä tulos on raju. WordPress vie ja muut vikisee keskikokoisissa ja pienissä sovelluskehitysprojekteissa, jotka tehdään verkkoon palvelinympäristöön PHP-kielellä.

## Lähteet

Backlinko. 2014. Google's 200 ranking factors: The Complete List. Päivitetty 8.8.2014. Viitattu 15.8.2014. <http://backlinko.com/google-ranking-factors>

Craft Pricing. 2014. The price is right for every site. Viitattu 29.7.2014. <http://buildwithcraft.com/pricing>

Craft Requirements, 2014. Requirements. Viitattu 29.7.2014. <http://buildwithcraft.com/docs/requirements>

Concrete5. 2011. Security and Concrete5. Artikkelijulkaistu huhtikuussa 2011. Viitattu 17.8.2014. <http://www.concrete5.org/documentation/how-tos/editors/security-and-concrete5>

GitHub. Tähtien tiedot otettu jokaisen järjestelmän oman repositoryn sivuilta 15.7.2014. <http://github.com>

GitHub rest-easy, REST Easy readme. 2014. Viitattu 29.7.2014. <https://github.com/wesrice/rest-easy>

GitHub OctoberCMS. 2014. Installation Wizard for October. Päivitetty 25.7.2014. Viitattu 29.7.2014. <https://github.com/octobercms/install>

Gooding, S. 2014, JSON Rest API Slated For WordPress 4.1 release. Artikkelijulkaistu 28.5.2014. Viitattu 1.7.2014. <http://wptavern.com/json-rest-api-slanted-for-wordpress-4-1-release>

KeystoneJS. 2014. Viitattu 29.7.2014. <http://keystonejs.com/>

Laravel Documentation. 2014. Basic Database Usage. Viitattu 29.7.2014. <http://laravel.com/docs/database>

Make WordPress UI. 2014. Front-end Editor Meeting April 22. Viitattu 29.7.2014. <http://make.wordpress.org/ui/2014/04/23/font-end-editor-meeting-22-april/>

October. 2014. Viitattu 29.7.2014. <http://octobercms.com/>

October. 2014. About Us. Viitattu 29.7.2014. <http://octobercms.com/about>

October Blog. 2014. Upcoming changes in the marketplace. Julkaistu kesäkuussa 2014. Viitattu 29.7.2014. <http://octobercms.com/blog/post/upcoming-changes-marketplace>



- SimplerCloud. 2014. What is the difference between LAMP stack and LEMP stack. Julkaistu elokuussa 2014. Viitattu 15.8.2014.  
<http://simplercloud.wordpress.com/2014/08/06/what-is-the-difference-between-lamp-stack-and-lem-p-stack/>
- SourceForge. 2008. Project of the Month, October 2008. Julkaistu lokakuussa 2008. Viitattu 29.7.2014. <http://sourceforge.net/blog/potm-200810/>
- Sprout. 2014. All-in-One Search Engine Optimization Plugin for Craft CMS. Viitattu 17.8.2014. <http://sprout.barrelstrengthdesign.com/craft-plugins/seo>
- Statamic. 2014. Changelog. Viitattu 29.7.2014. <http://www.statamic.com/changelog>
- Statamic Store. 2014. Buy a licence. Viitattu 29.7.2014. <https://store.statamic.com/>
- Straight Up Craft. 2014. Craft Plugins. Viitattu 29.7.2014.  
<http://straightupcraft.com/craft-plugins>
- Treehouse Blog. 2014. What is Ajax and Where to Use It? Julkaistu kesäkuussa 2014. Viitattu 17.8.2014. <http://blog.teamtreehouse.com/what-is-ajax>
- Type & Grids. 2014. Goodbye WordPress: 2014 will be the year of flat-file CMS. Viitattu 17.8.2014 <http://www.typeandgrids.com/blog/goodbye-wordpress-2014-will-be-the-year-of-flat-file-cmses>
- W3 Techs. 2014. Market share Trends for content management systems for websites. Viitattu 1.7.2014.  
[http://w3techs.com/technologies/history\\_overview/content\\_management](http://w3techs.com/technologies/history_overview/content_management)
- W3 Techs. 2014. Usage statistics and market share of Apache for websites. Viitattu 17.8.2014. <http://w3techs.com/technologies/details/ws-apache/all/all>
- WcKeown, S. 2014. WordPress Chapter 3 - On forking WordPress, Forks in General, Early WordPress and the Community. Viitattu 1.7.2014.  
<http://wordpress.org/about/history/chapter3.pdf>
- WordCamp Central. 2014. Schedule. Viitattu 1.7.2014.  
<http://central.wordcamp.org/schedule/>
- WordPress Most Popular Plugins. 2014. Advanced Custom Fields löytyy ensimmäisenä aiheeseen liittyvistä lisäosista. Viitattu 29.7.2014.  
<https://wordpress.org/plugins/browse/popular/>
- WordPress Notable Users. 2014. Viitattu 29.7.2014. <http://en.wordpress.com/notable-users/>
- WordPress Plugins Directory. 2014. Viitattu 1.7.2014. <https://wordpress.org/plugins/>

WordPress Showcase. 2014. Viitattu 29.7.2014. <https://wordpress.org/showcase/>

WP Tavern. 2014. JSON REST API Slated For WordPress 4.1 Release. Viitattu 29.7.2014. <http://wptavern.com/json-rest-api-slated-for-wordpress-4-1-release>

Wpmudev. 2013. 10 wp-config Tweaks To Improve Your WordPress Site. Julkaistu marraskuussa 2013. Viitattu 29.7.2014. <http://premium.wpmudev.org/blog/10-wp-config-tweaks-to-improve-your-wordpress-site/>

## Liitteet

### Liite 1. Projektin aikajana

