

# Seminar TIE-11406: Backend as a Service (BaaS)

Tampere University of Technology

05.11.2013

<http://www.deployd.com/>

application example

Juha Nurmi, Timo Kokko

# Deployd - technology demonstration

1. Introduction
2. Overview of the system
3. Design principles
4. Walkthrough of a complete code example
5. Evaluation (benefits, drawbacks, usefulness)
6. Summary

# Introduction

# Motivation

We used these technologies to create map application based ticket system for everyone:

<http://www.apps4finland.fi/kilpailutyö/kunnossapitoa-ja-korjaamista-kommunikoiden/>

Apps4Finland contests is producing practical solutions to everyday problems. This year the ambitious goal of the contest is to make Finland one of the top countries in open data utilization by increasing the degree of collaboration and commitment between the various interest groups.

*"Kaupunkilaiset voivat ilmoittaa kunnossapitopyyntöjä karttapohjaisella sovelluksella, joka on kytkettävissä valmiisiin tikettijärjestelmiin."*

# Deployd demo

- Deployd backend as a service system
- HTML5 and JavaScript
- RESTful
- Node.js
- JIRA

# **Overview of the system**

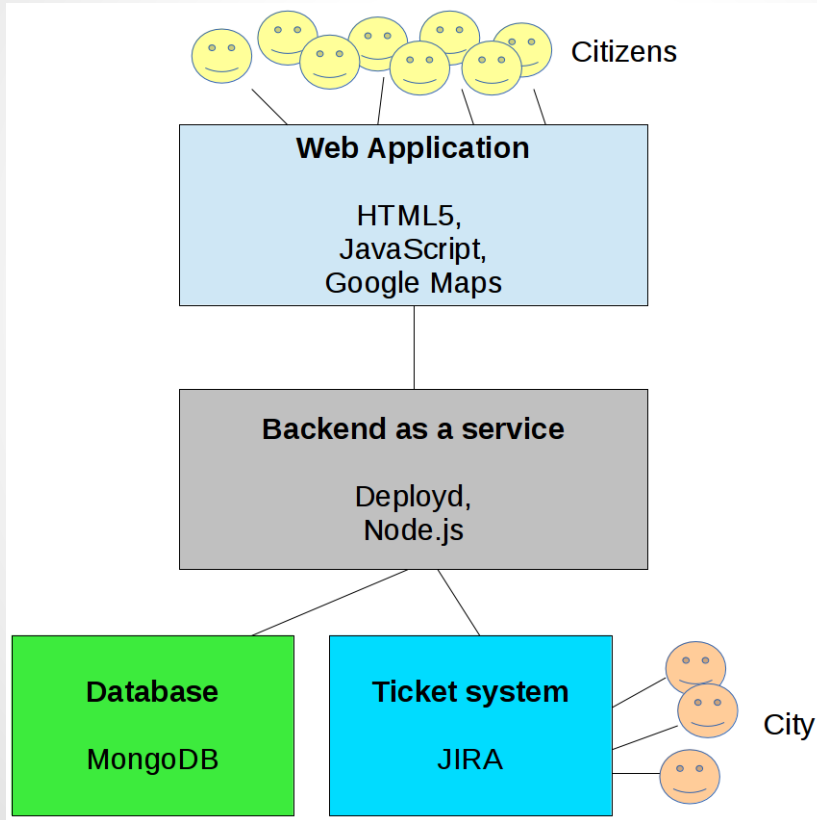
# Issue tracking system for Tampere

- <http://baas.cloud.tilaa.com:7070/>
- JIRA integration <http://baas.cloud.tilaa.com/>

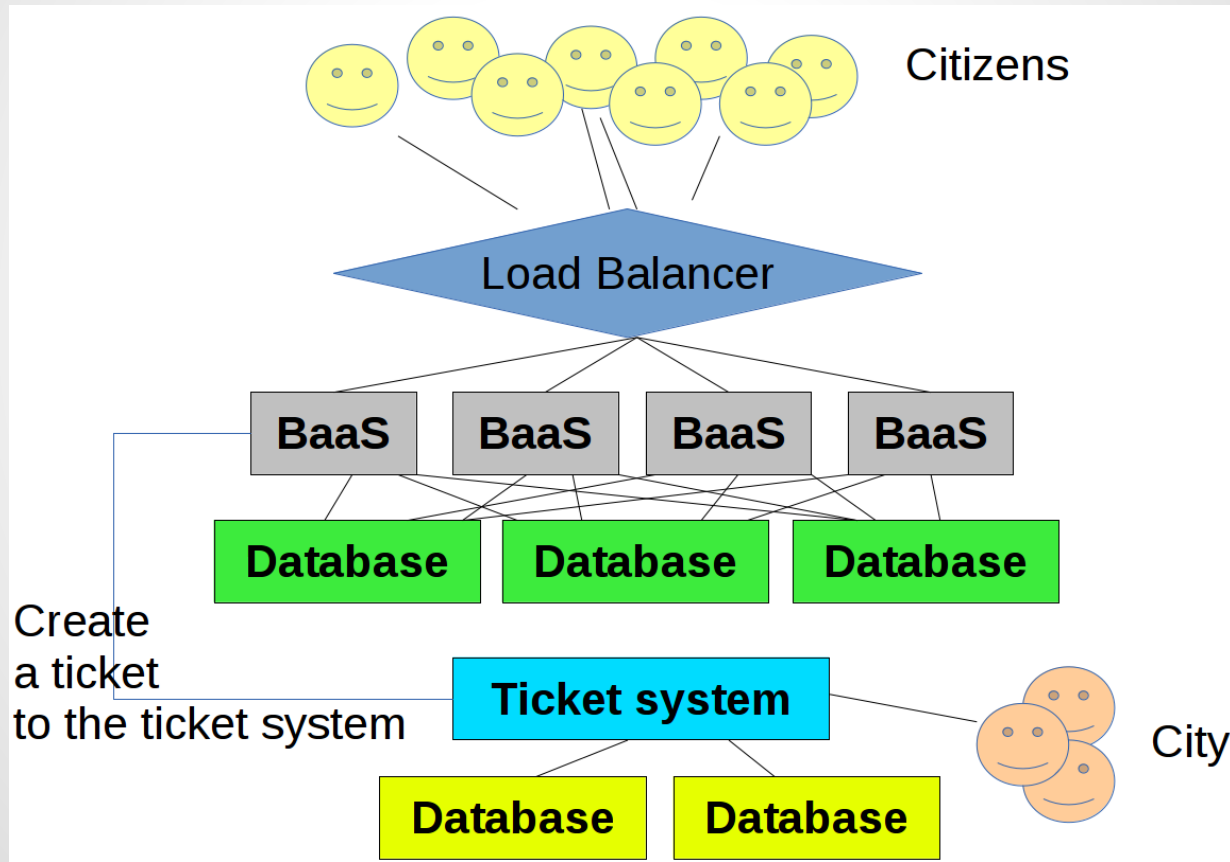
# **Design principles**



# Application Architecture

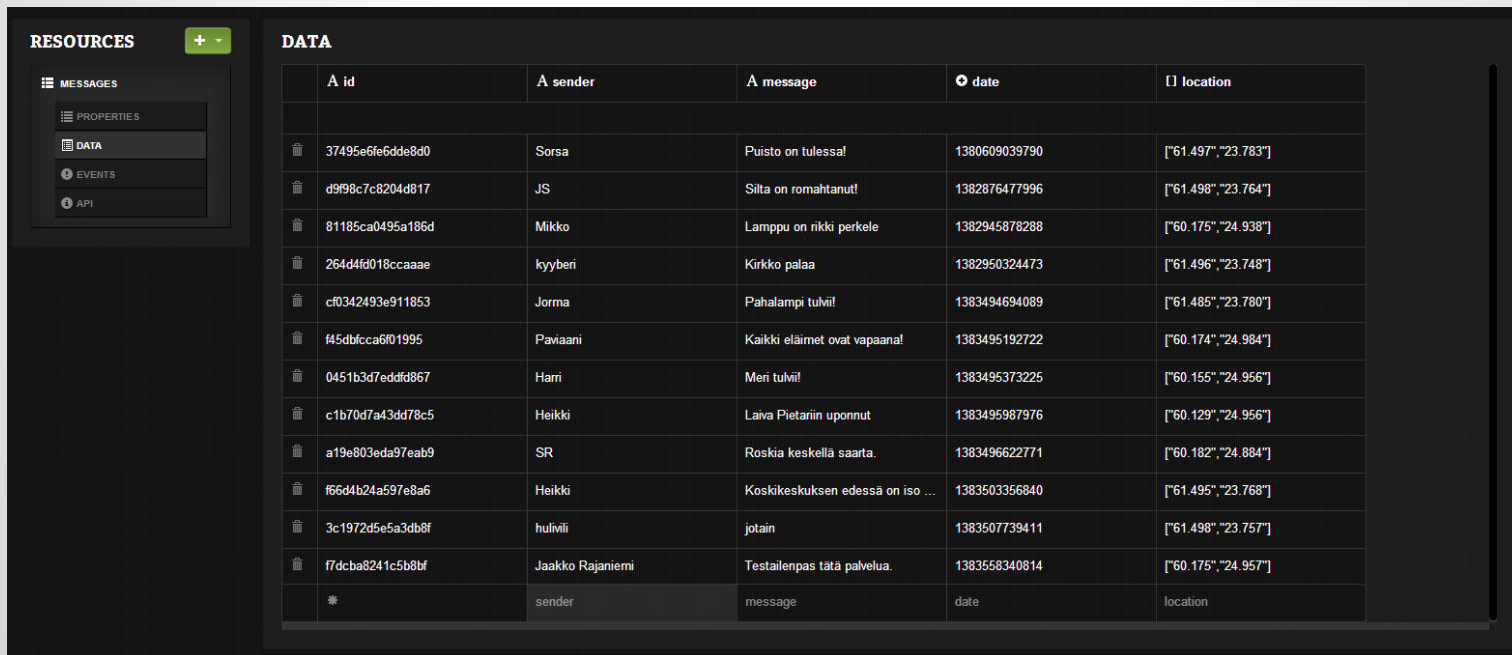


# Application Scalability



# Designing RESTful APIs

- The user created messages from the UI are stored into a CRUD collection in Deployd using REST calls

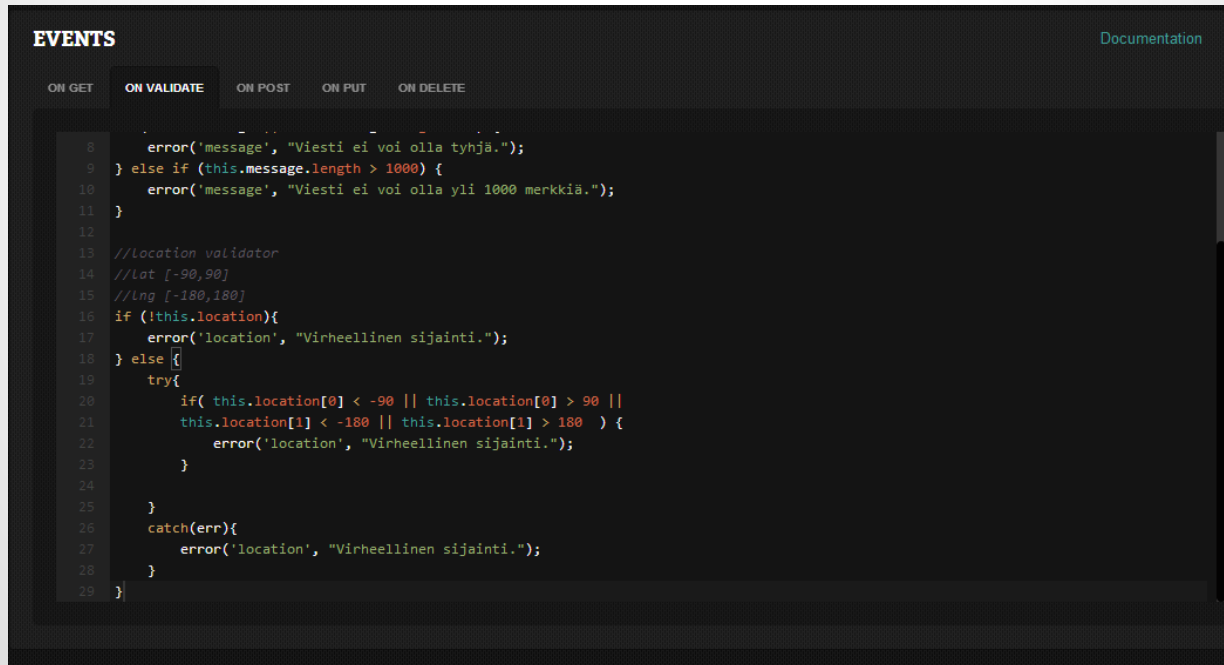


The screenshot shows a REST client interface with a sidebar on the left and a main data table on the right. The sidebar has a 'RESOURCES' section with a '+ -' button and a list of resources: 'MESSAGES', 'PROPERTIES', 'DATA', 'EVENTS', and 'API'. The 'DATA' resource is selected, and the main area displays a table of message data.

	A id	A sender	A message	🕒 date	📍 location
🗑️	37495ef6fe6dde8d0	Sorsa	Puisto on tulessa!	1380609039790	["61.497", "23.783"]
🗑️	d9f98c7c8204d817	JS	Silta on romahtanut!	1382876477996	["61.498", "23.764"]
🗑️	81185ca0495a186d	Mikko	Lamppu on rikki perkele	1382945878288	["60.175", "24.938"]
🗑️	264d4fd018ccaaae	kyyberi	Kirkko palaa	1382950324473	["61.496", "23.748"]
🗑️	cf0342493e911853	Jorma	Pahalampi tulvi!	1383494694089	["61.485", "23.780"]
🗑️	f45dbfcca6f01995	Paviaani	Kaikki eläimet ovat vapaana!	1383495192722	["60.174", "24.984"]
🗑️	0451b3d7eddf8867	Hari	Meri tulvi!	1383495373225	["60.155", "24.956"]
🗑️	c1b70d7a43dd78c5	Heikki	Laiva Pietariin uponnut	1383495987976	["60.129", "24.956"]
🗑️	a19e803eda97eab9	SR	Roskia keskellä saarta.	1383496622771	["60.182", "24.884"]
🗑️	f66d4b24a597e8a6	Heikki	Koskikeskuksen edessä on iso ...	1383503356840	["61.495", "23.768"]
🗑️	3c1972d5e5a3db8f	hulvili	jotain	1383507739411	["61.498", "23.757"]
🗑️	f7dcb8241c5b8bf	Jaakko Rajaniemi	Testailenpas tätä palvelua.	1383558340814	["60.175", "24.957"]
*		sender	message	date	location

# Scripted Logic in Events

- Events are used to intercept REST calls and inject business logic



```
EVENTS Documentation  
  
ON GET ON VALIDATE ON POST ON PUT ON DELETE  
  
8     error('message', "Viesti ei voi olla tyhjä.");  
9 } else if (this.message.length > 1000) {  
10    error('message', "Viesti ei voi olla yli 1000 merkkiä.");  
11 }  
12  
13 //Location validator  
14 //Lat [-90,90]  
15 //Lng [-180,180]  
16 if (!this.location){  
17    error('location', "Virheellinen sijainti.");  
18 } else {  
19    try{  
20       if( this.location[0] < -90 || this.location[0] > 90 ||  
21          this.location[1] < -180 || this.location[1] > 180 ) {  
22          error('location', "Virheellinen sijainti.");  
23       }  
24    }  
25    }  
26    catch(err){  
27       error('location', "Virheellinen sijainti.");  
28    }  
29 }
```

# Using DPD Library

- Build-in DPD library can be used to access collections in Deployd database and script business logic

## API

HTTP

JAVASCRIPT

Deployd generates a [browser JavaScript API](#) for easily accessing data from the `/messages` collection. To use it you first need to include the generated script.

```
<script src="dpd.js"></script>
```

Insert data into the `/messages` collection:

```
dpd.messages.post({"sender":"foobar","message":"foobar","date":123}, function(result, err) {  
  if(err) return console.log(err);  
  console.log(result, result.id);  
});
```

# Using Node.js Modules

- Deployd can be easily extended using node.js modules that are automatically loaded when placed under directory node\_modules
- Process.server can be used to reference Deployd server object in the code
- Many deployd modules available including sending email and getting files from Amazon S3
- Modules are Node Packages and can be reused using Node.js package manager (NPM)

```
jira_rest.js  
root@baas:~/deployd/citymapper/node_modules# █
```

# **Walkthrough of a complete code example**

# Walkthrough of a complete code example

1. Running the code: **dpd --open --port 7070**  
or **node mapper.js**
2. See mapper.js



# Walkthrough of a complete code example

1. Client side code (HTML, JavaScript, CSS)
2. See `/public/index.html`, `/public/js`,  
`/public/img/`, `/public/css/`

# Walkthrough of a complete code example

1. Server side code in Deployd
2. <http://baas.cloud.tilaa.com:7070/dashboard/>
3. See validators in the dashboard

# Walkthrough of a complete code example

1. Server side code in Node.js
2. `/node_modules/jira_test.js`
3. Coding itself is easy, the problems where elsewhere
4. Eventually we got a very short and clean solution how to attach tickets to JIRA
5. We had to guess how to catch these messages from Deployd
6. And use a lot of time to sysop duties

# Evaluation

# Pros

- Easy and fast to get started
- Good documentation and example projects available
- Real-time capabilities
- Multi-platform support
- Supports web and mobile apps
- Good for prototyping
- Easy to extend with Node.js modules
- Saves from boilerplate code
- Being Open Source can be developed further

# Cons

- No datetime type
- No support for MongoDB replication or sharding
- Authentication system needs to be augmented for production use
- Encryption only for cloud version which is not available

# Deployd and Create Event

- We had some problems figuring out how to catch this create event in Node.js

```
dpd.on('messages:create', function(req) {  
  if (req.method === 'POST') {  
    console.log("POST");  
  }  
});
```

# Further Development Ideas

- Further development is possible
- Can be integrated to <http://dev.helsinki.fi/apis/issuereporting>
- We can select any ticket system
- JIRA supports email integration and automated user account creation
  - the user could follow his/hers tickets



# Summary

# Summary

- Deployd delivers what it promises and removes a lot of boilerplate code
- Deployd has to be developed further for more complex scenarios
  - Stronger authentication and encryption
  - Failover capabilities using Forever scripts
- Deployd is open source → easy to extend with Node.js
- Although, we had to guess how Deployd actually worked with events
- Easy to use and extend