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/kilpailutyo/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

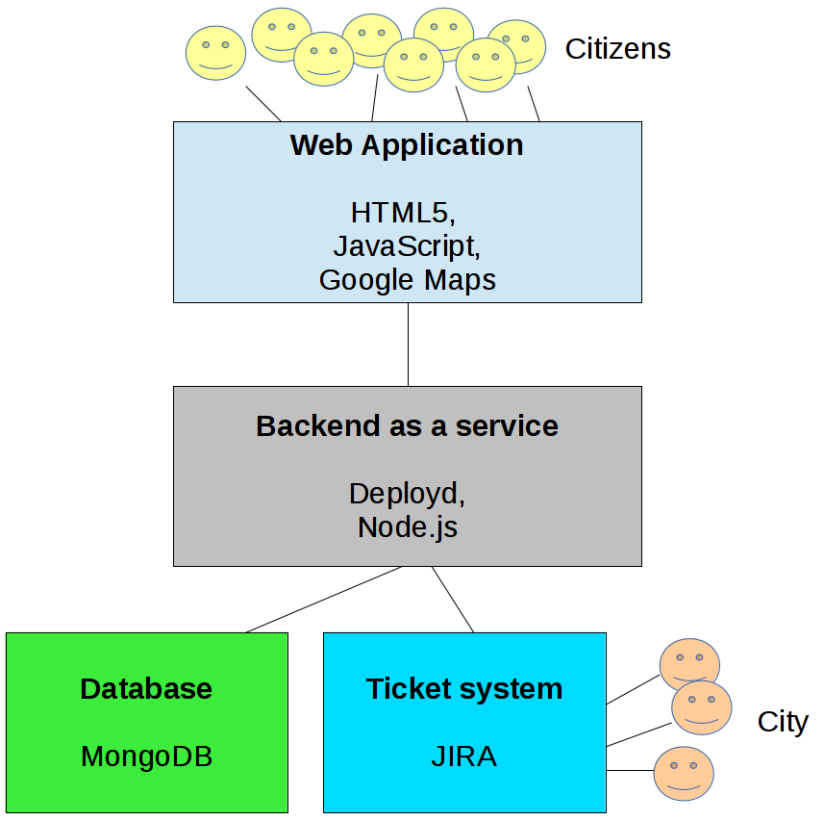
Overview of the system

Issue tracking system for Tampere

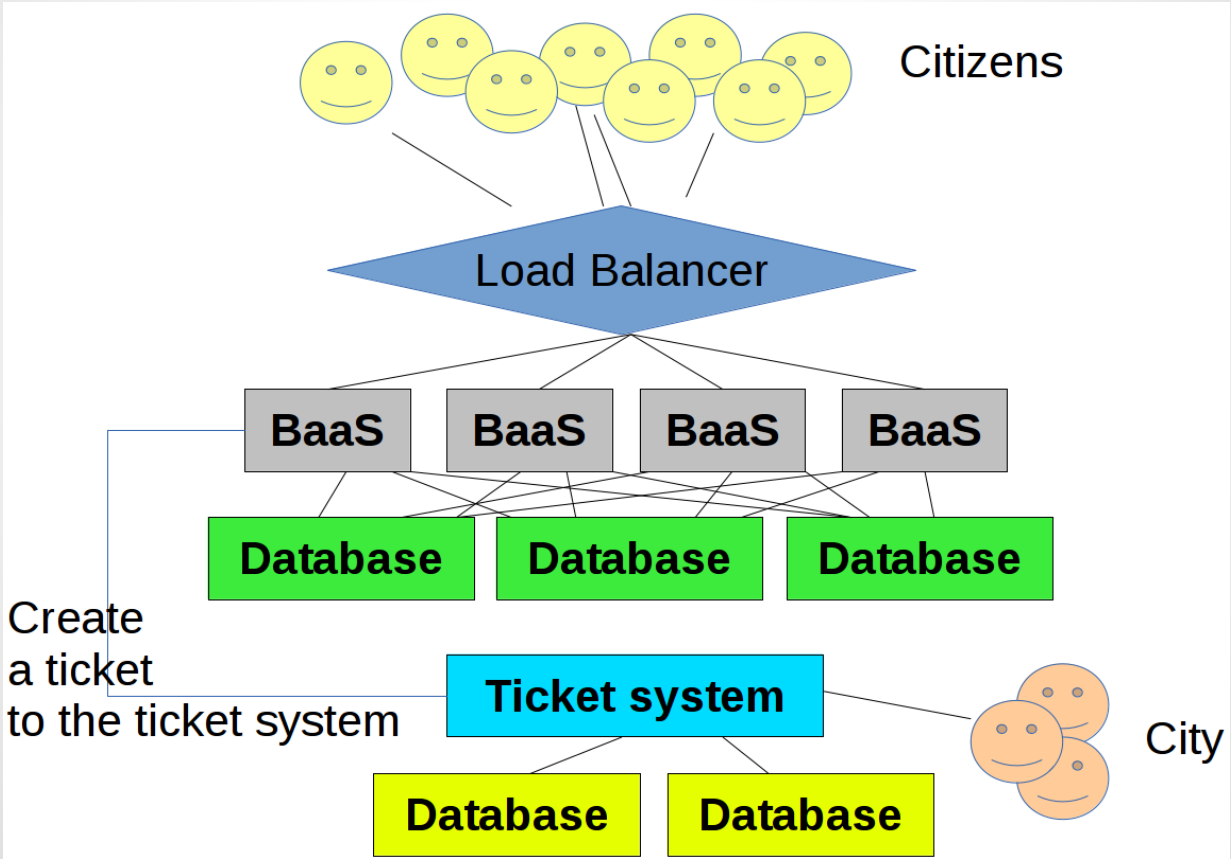
- http://baas.cloud.tilaa.com:7070/
- JIRA integration http://baas.cloud.tilaa.com/

Design principles

Application Architecture

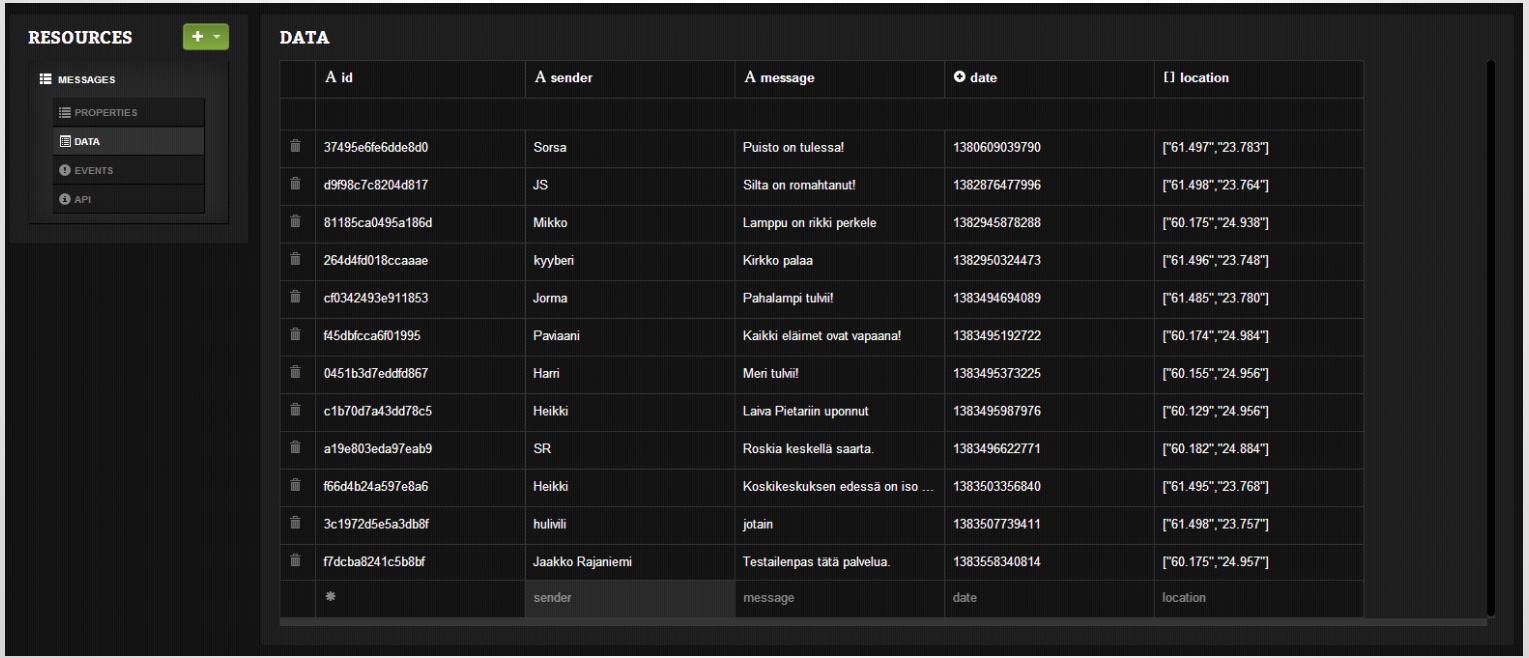


Application Scalability



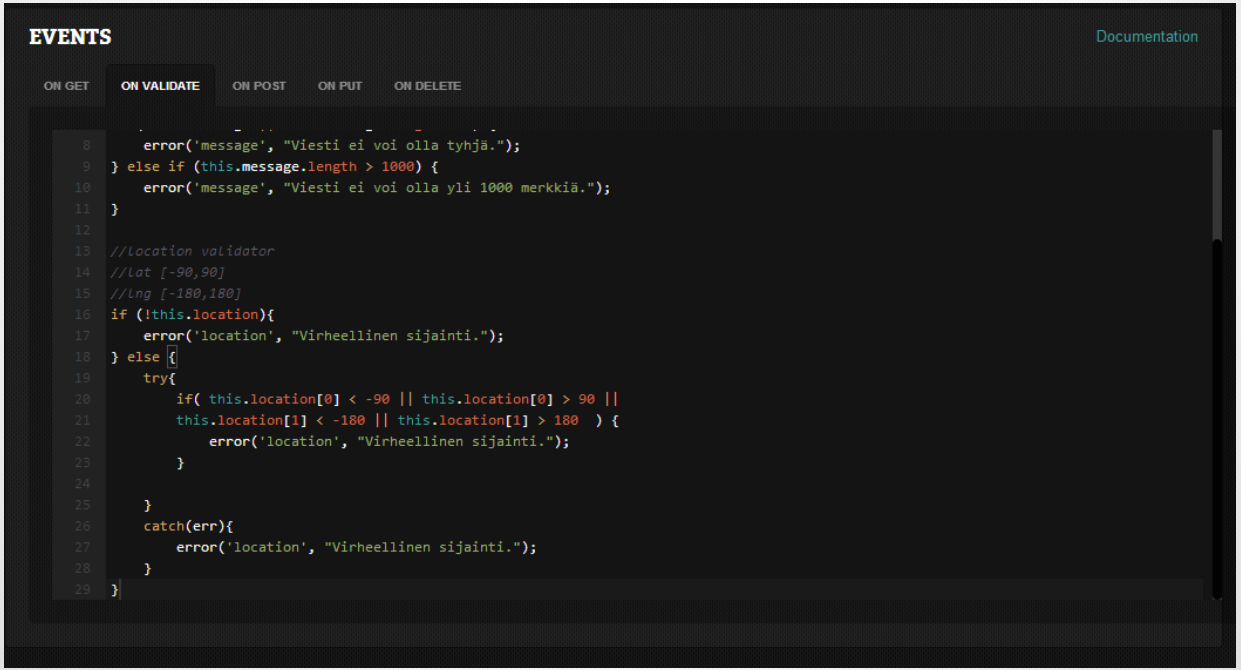
Designing RESTful APIs

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



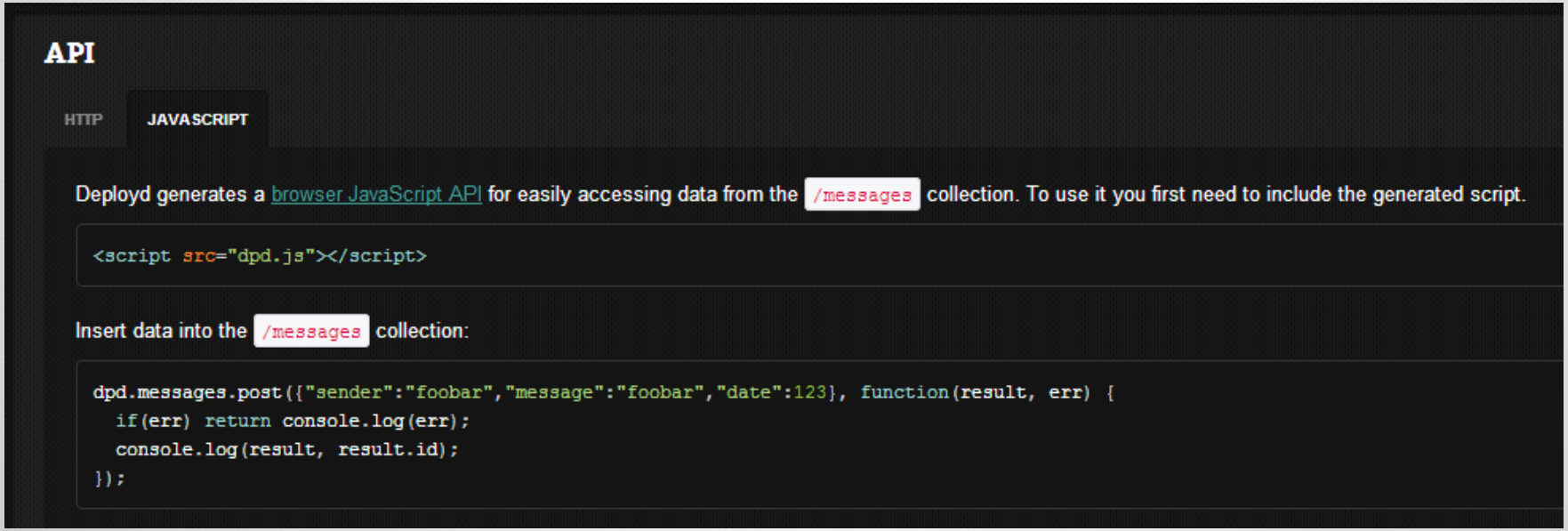
Scripted Logic in Events

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



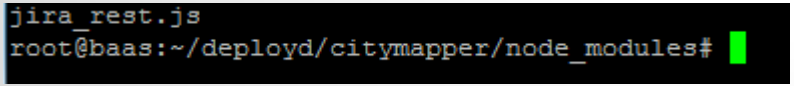
Using DPD Library

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



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)



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 integraded to http://dev.hel.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