



UNIVERSITY *of* LIMERICK  
OLLSCOIL LUIM NIGH

## Project Log

### Spring Semester

#### Week 4

**MEng Information & Network Security**

**Thomas Flynn**

**16117743**

**Project Supervisor: Sean McGrath**

**13/02/17- 19/02/17**

# **1 Log Entries**

## **1.1 Entry 15/02/17**

Today I struggled to get the HAProxy container working. I spent the first hour and a half just trying to attach an “override config” as a volume to the container. Once “docker logs <haproxy container id>” stopped reporting errors, I knew at least the container was building successfully. The next step is to export HAProxy logs in order to find out more information as to what the problem might be.

## **1.2 Entry 16/02/17**

Today I researched both Kafka and Redis, trying to understand the difference between them and why/where you would use one over the other.

## **1.3 Entry 17/02/17**

Today I started my Spring report, this consisted of laying out the template and putting in section headings. I also research Amalgam8 microservices framework.

## **1.4 Entry 18/02/17**

Today I worked on “section 2 Literature survey” of my spring report. I also researched container monitoring and logging using the ELK stack.

## **1.5 Entry 19/02/17**

Today I continued work on my Spring report, finishing off the literature survey section.

## 2 Tasks completed

- Attempted to implement HAProxy container
- Redis research
- Kafka research
- ELK Stack research
- report draft
- Literature survey spring report

## 3 GIT Repositories

### 3.1 INS-Thesis-Docmentation

Commits on Feb 17, 2017

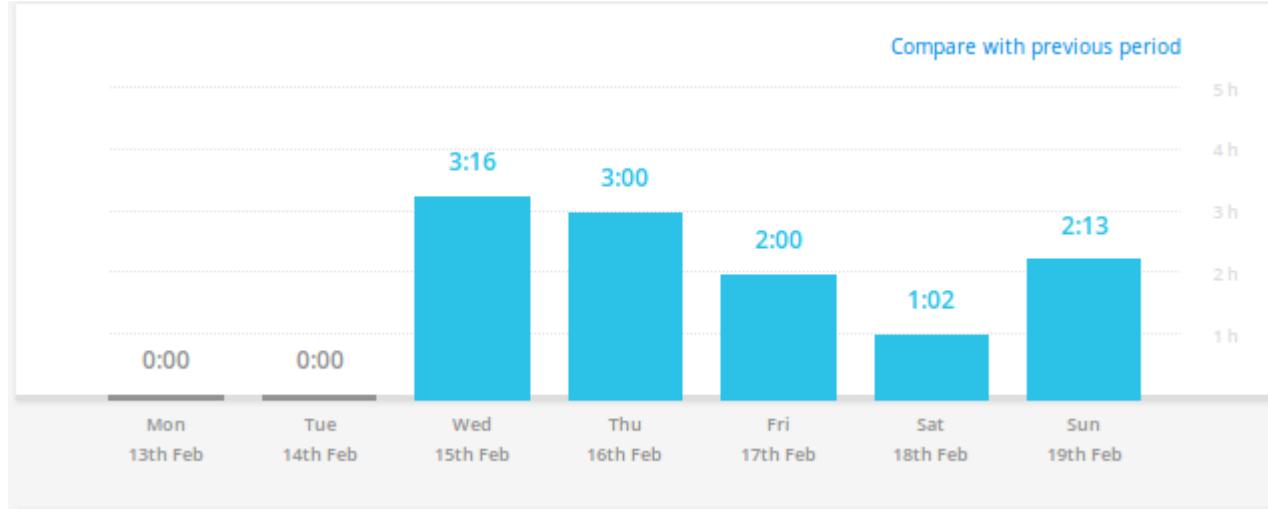
 report draft v1	<a href="#">16117743 committed 3 days ago</a>	 <a href="#">1773454</a>	
---	---	---	---

Commits on Feb 12, 2017

 week 3 log	<a href="#">16117743 committed 8 days ago</a>	 <a href="#">ef8f976</a>	
--	---	---	---

## 4 Toggl Time Logs

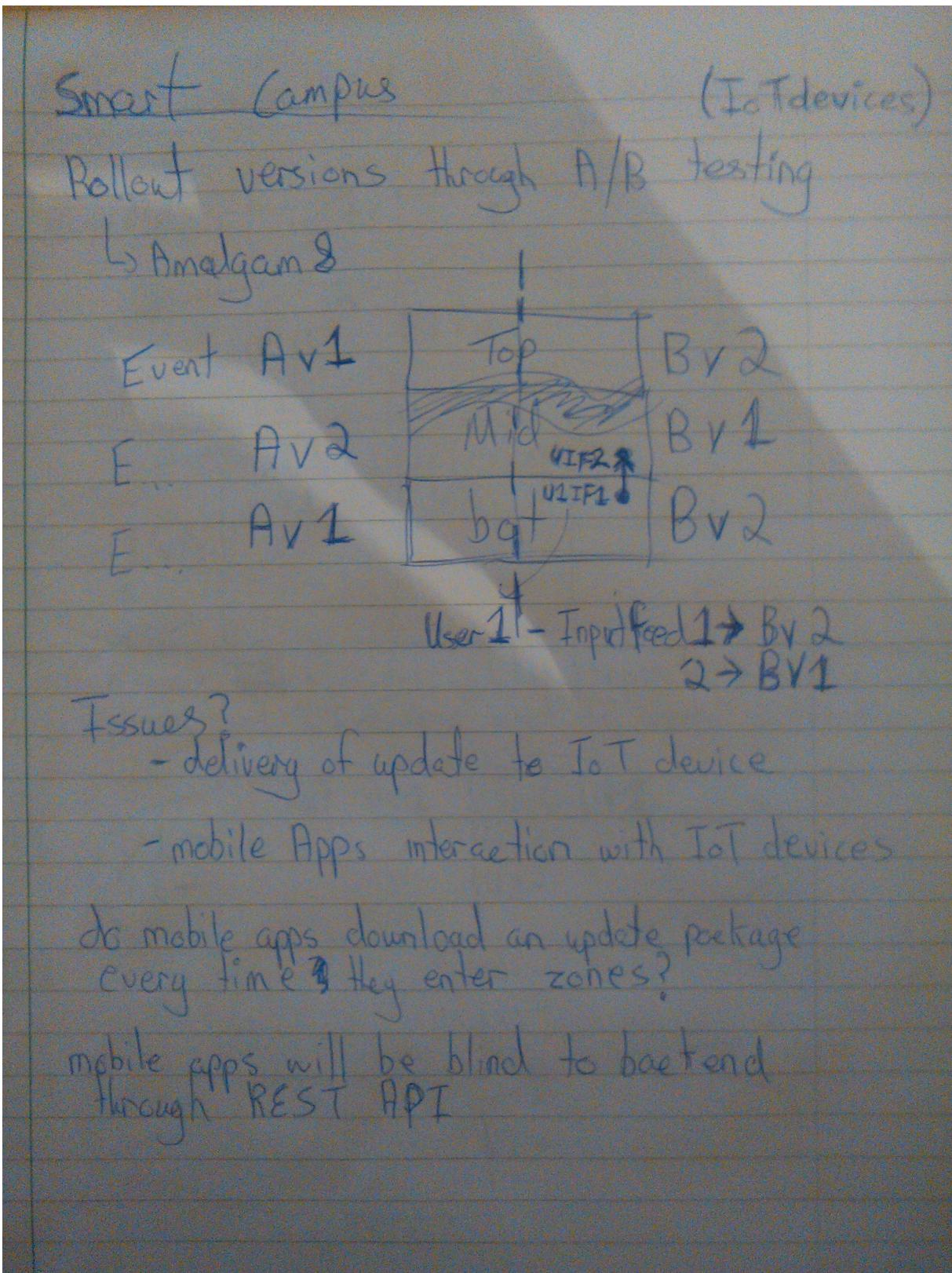
### 4.1 Weekly time Log bar chart



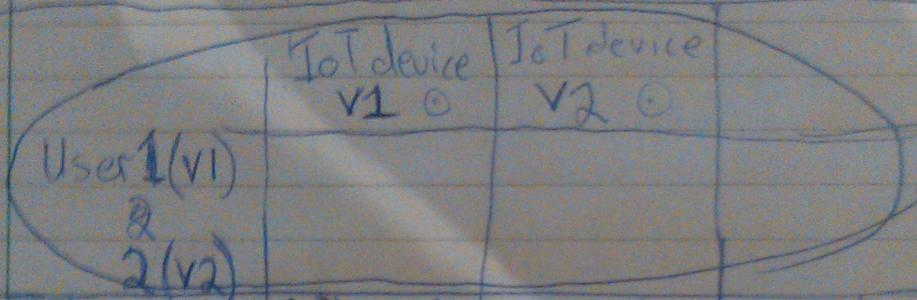
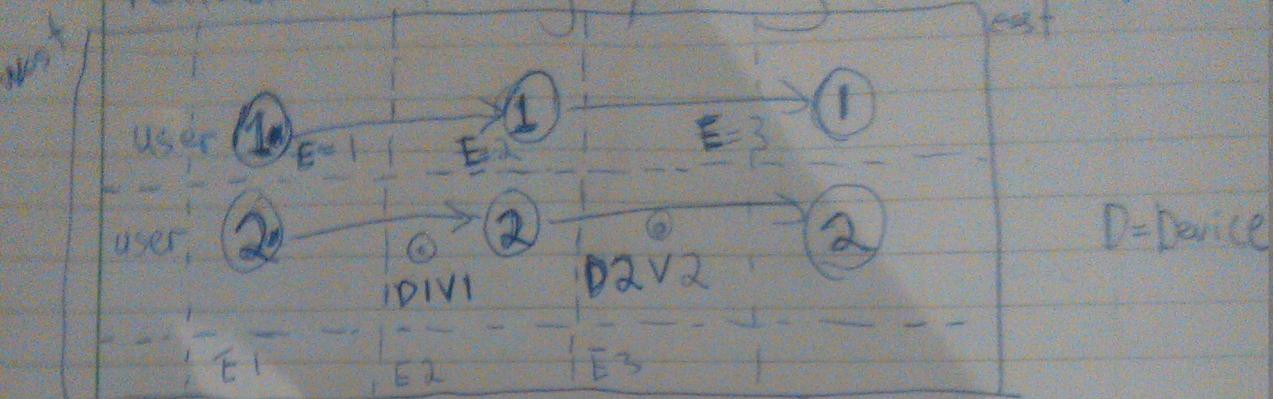
### 4.2 Weekly time log

Yesterday		2:13:42	
report draft	● General	5:40 PM - 7:59 PM	2:13:42
Sat, 18 Feb			1:02:23
report draft	● General	1:46 PM - 2:48 PM	1:02:23
Fri, 17 Feb			2:00:00
report draft	● General	4:44 PM - 6:44 PM	2:00:00
Thu, 16 Feb			3:00:00
Kafka research	● Database	7:07 PM - 8:07 PM	1:00:00
Redis research	● Database	5:07 PM - 7:07 PM	2:00:00
Wed, 15 Feb			3:16:15
HAProxy	● Back end	2:23 PM - 5:39 PM	3:16:15

## 5 Pictures

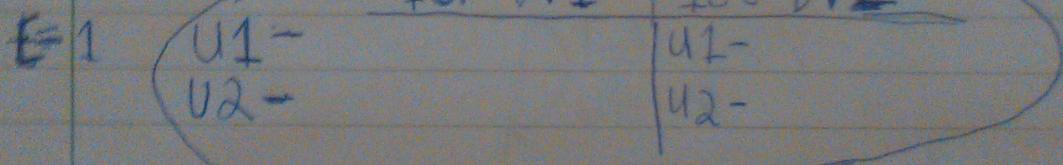


## Pilot out versions through A/B testing (mobile Apps)



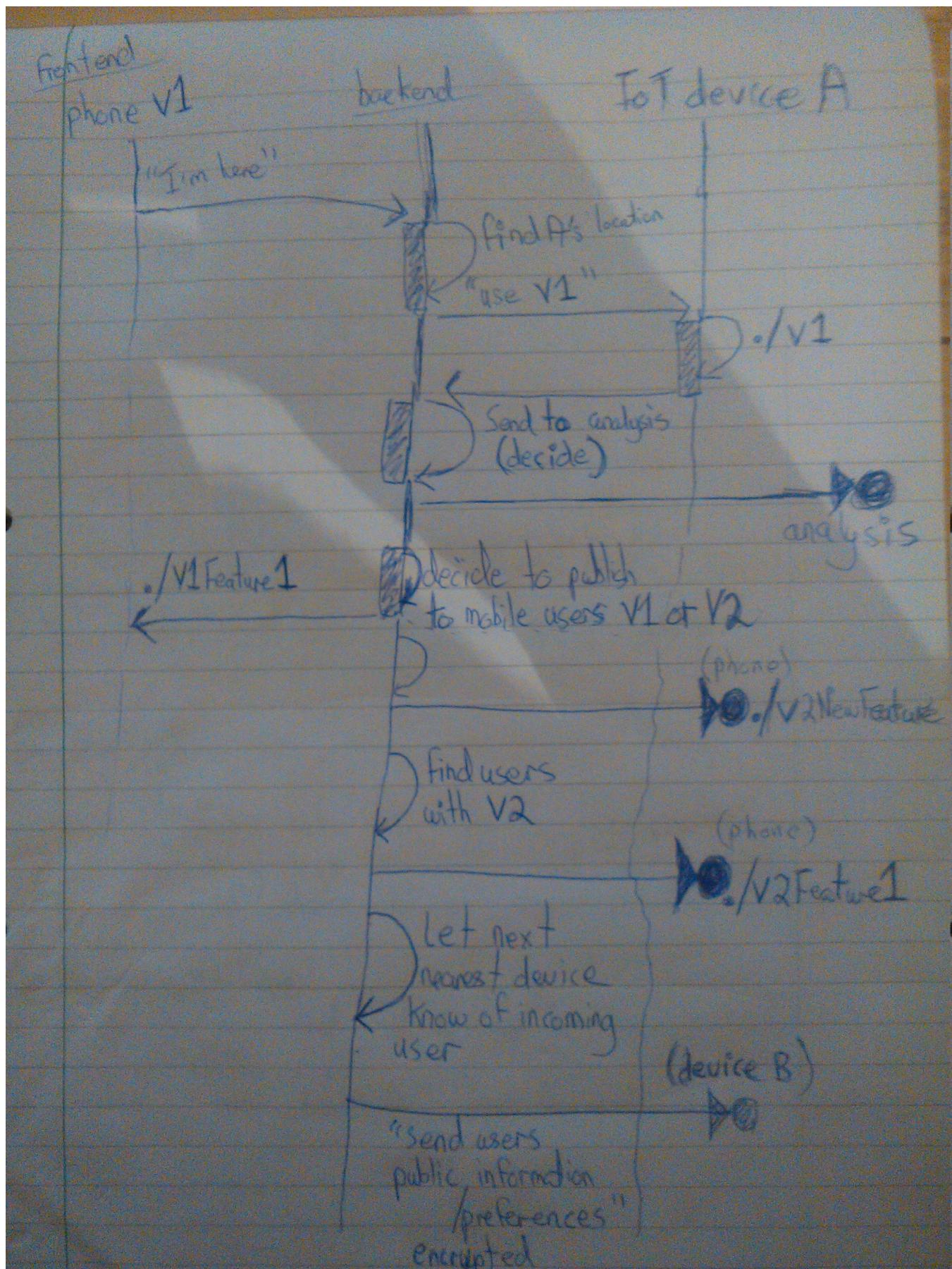
Note: front ~~is~~ backend updated Separate to IoT devices  
 ie phones and IoT devices react to "events" differently

= Event  
= user version



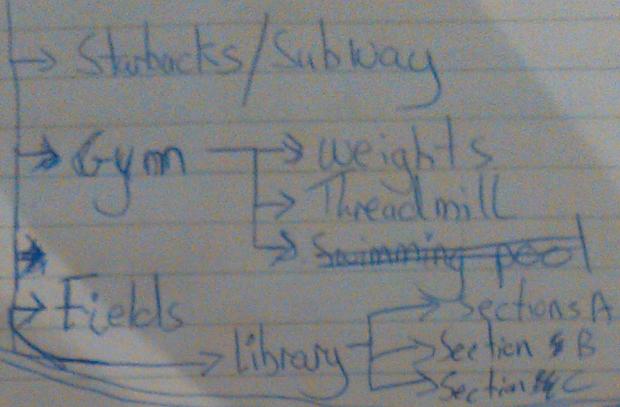
• 19.3

• 19.3



(msgs)

backend facilitates high frequency GPS updates  
from individual users who...  
are in designated areas...



Q Why?

A: Realtime info for these areas ~~is not~~  
is more critical.

Q Out of HA?

↳ Users who are outside "designated areas"  
msg less frequently and their caching  
of data is of lower priority

Q Message failure?

msgs that fail to send for users in designated  
spots...  
are transmitted more frequently

Q Other ideas?

users leaving campus transmit a "I might be leaving"  
tag.

If we don't hear from them in 20 mins,  
presume dead gone

(meets better)

~~Is it~~ "EXPOSE" or  
↳ service in container is only  
accessible from other containers

Flag "-p"  
↳ accessible from outside docker  
-P without "EXPOSE" does implicit "Expose" anyway

Problem?

Container (single) works, but exposing ports  
in group ~~isn't~~ for non http traffic can't be done

"integrated" domain routing is only available for container groups

## Container Groups

Fields - host:

Domain:

↳ System domain already provides SSL certificate, so accessible through HTTPS

To make custom domains secure, upload SSL cert

Field    Http port

specify public port to publish

Drop down list is auto populated with ports "EXPOSED" on "Dockerfile".

For container groups you cannot specify multiple ports

When you bind a route, containers in your group must listen for HTTP traffic on the group's exposed port!

Non-HTTP ports cannot be exposed publicly.

When HTTPS traffic arrives on the exposed port, the (Go) router completes the HTTPS termination.

MQTT

BB = TLS

Kafka consumer - runs in thread

consumer connects to Zookeeper server,

which distributes the consumers to the kafka servers active in the cluster or group.

Circuit breaker pattern provides improved stability / availability by...

Shielding the caller from calling the real operation when...  
reaching a specified failure count

### Bulkhead stability pattern

- failures in one system should not cause failures in another
- kafka as msg broker allows for multiple instances of each service
- Bulkheads help prevent "hidden linkages" by replicating services and make each system use its own dedicated service pool.

Heartbeat timer for microservices  
ping every 15 seconds

- uses msg Queue provider to handle the load and fan out the heartbeat msgs to clients

Reservation pattern model

- Request
  - Response
  - Confirmation
- } events

if request is to modify data then  
a reservation is performed.

Let Service that owns the data tell you  
when there are changes.

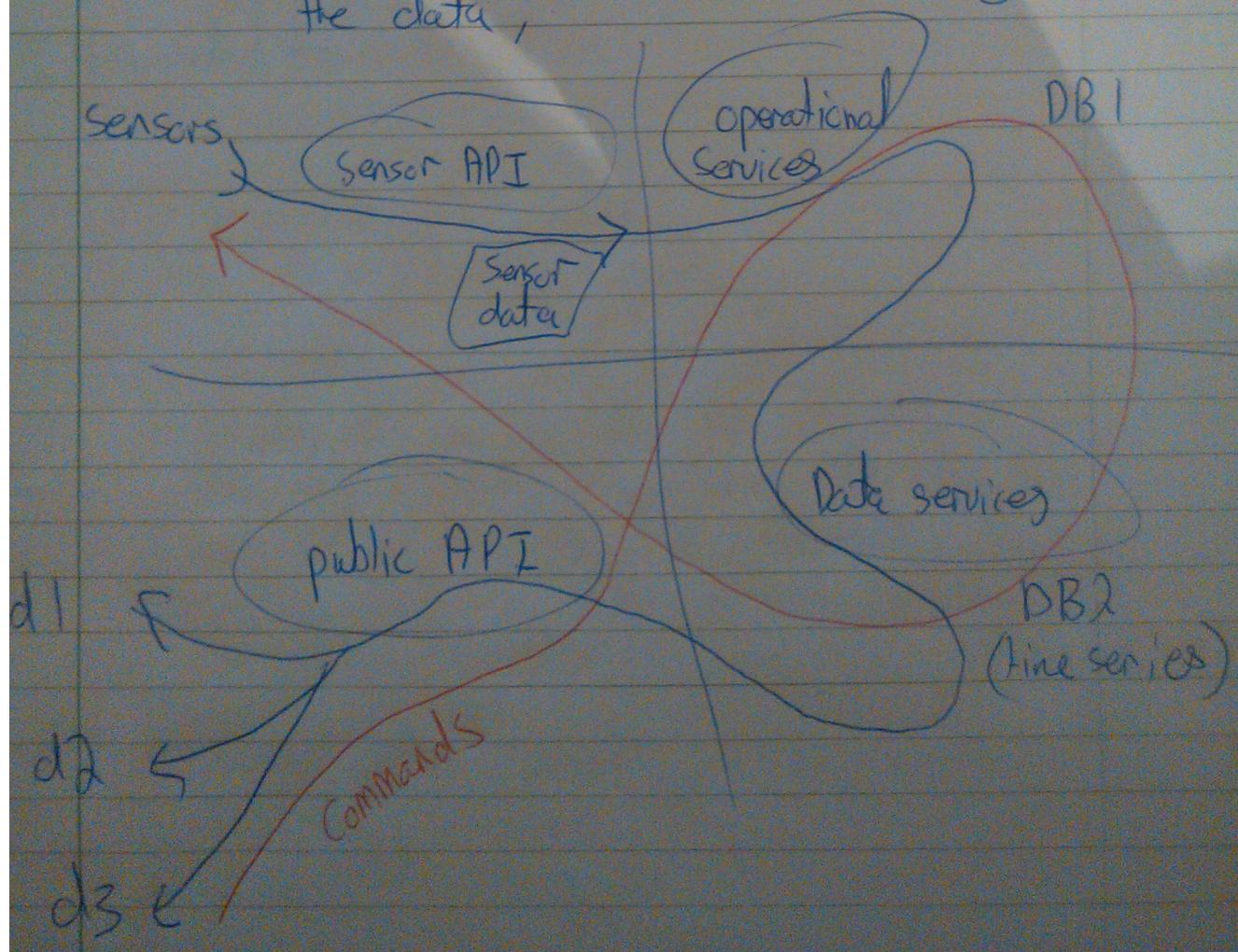
Even if DB can prevent all but one node  
to succeed in writing the activity record,  
this is wasteful because all the nodes  
are attempting the same task.

Sensor API → called to deliver data readings and receive commands

Public API → called by sensors to retrieve real-time data, historical data, manage devices

Operational services → responsible for authentication and authorization, these services manage their own DB

Data services → responsible for storing & analyzing the data,



## 6 Bookmarks

<https://www.altoros.com/blog/continuous-integration-and-continuous-delivery-in-ibm-bluemix/>

[https://www.ibm.com/blogs/bluemix/2016/07/amalgam8-framework-for-microservices-orchestration/?S\\_TACT=M16103KW](https://www.ibm.com/blogs/bluemix/2016/07/amalgam8-framework-for-microservices-orchestration/?S_TACT=M16103KW)

<http://addendum.e.altervista.org/tag/nginx-websocket-proxy-configuration/>

<https://www.rabbitmq.com/features.html>

<https://github.com/mcollina/mosca>

<https://aucouranton.com/2015/10/08/a-breakdown-of-layers-and-tools-within-the-container-and-microservices-ecosystem/>

<http://qiita.com/giwa/items/5da37c3aca0aee75a628>

<https://docs.docker.com/compose/overview/>

<https://www.amalgam8.io/docs/demo-helloworld.html>

<http://microservices.io/patterns/server-side-discovery.html>

<https://www.linkedin.com/pulse/how-secure-data-kafka-robert-fuller>

<http://www.bigdataeverywhere.com/files/atlanta/BDE-KafkaRedisTools-CURTIN.pdf>

[https://console.ng.bluemix.net/docs/containers/monitoringandlogging/container\\_ml\\_getting\\_started\\_tutorial.html#container\\_ml\\_gettingstarted\\_tutorial](https://console.ng.bluemix.net/docs/containers/monitoringandlogging/container_ml_getting_started_tutorial.html#container_ml_gettingstarted_tutorial)

[https://www.ibm.com/developerworks/library/ba-bluemix-elklog/#1.4.1.ProvisionaSoftLayerServer\(optional\)](https://www.ibm.com/developerworks/library/ba-bluemix-elklog/#1.4.1.ProvisionaSoftLayerServer(optional))

<https://www.ibm.com/developerworks/library/ba-bluemix-elklog/#N1013F>

<https://developer.ibm.com/cloudarchitecture/docs/service-management/elasticsearch-logstash-kibana-analytics-bluemix/>

<http://logz.io/learn/complete-guide-elk-stack/>

<http://logz.io/blog/kafka-vs-redis/>

<https://devops.profitbricks.com/tutorials/creating-a-grafana-and-influxdb-docker-container/>

[http://air.imag.fr/index.php/Developing\\_IoT\\_Mashups\\_with\\_Docker,\\_MQTT,\\_Node-RED,\\_InfluxDB,\\_Grafana](http://air.imag.fr/index.php/Developing_IoT_Mashups_with_Docker,_MQTT,_Node-RED,_InfluxDB,_Grafana)

[http://cs.au.dk/fileadmin/site\\_files/cs/AA\\_pdf/ClausDNielsen\\_rapport.pdf](http://cs.au.dk/fileadmin/site_files/cs/AA_pdf/ClausDNielsen_rapport.pdf)



