

# Microservices, MQ Light and Space Probes

[j0nnymac](#)

Published on 14/05/2015 / Updated on 25/05/2015

0

One of the activities we really enjoy doing is working with the academic community to do something innovative and fun. In the theme of the democratisation of innovation, we are working with students from the Imperial College in London, to create an edge of space probe. The probe, will send back telemetry and using BlueMix technologies such as MQ Light, Node-RED, Cloudant we will then attempt to represent (rather than replicate) the conditions encountered by the probe, both virtually and physically.

The team from Imperial have started work already and have created a blog post around their first experiences with BlueMix, MQ Light and microservices...

*Yesterday we were treated to a talk from the lovely Steve Upton, who stopped by Imperial on his way to a talk he was giving in London. Steve is very knowledgeable in the world of microservices, and kindly came to tell us all about modern software architecture.*

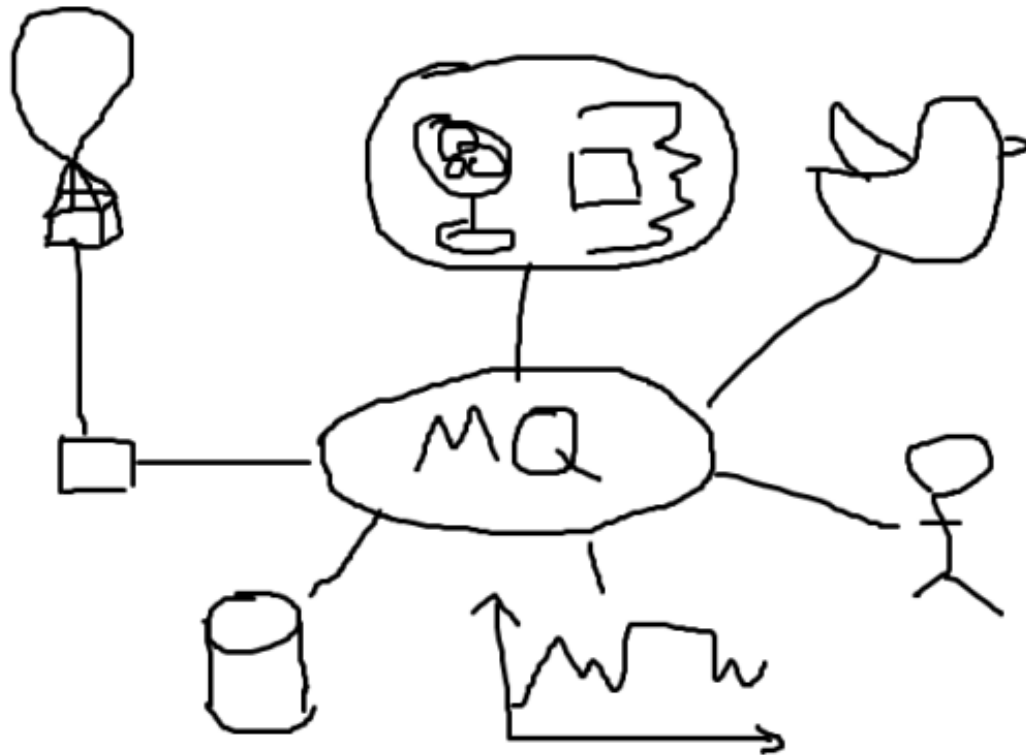
*He explained to us that old architectures were usually self contained, existing as a single monolith of a project...*

*This is great for small projects, but it isn't scalable because if one component breaks, the whole thing goes down. If you want to just, for example, change a logo, you have to update the entire thing at once. It quickly becomes unsuitable as the size of the team working on the project gets bigger, as parallelising work is impossible when the processes are dependent on each other.*

*Microservices counteract these flaws by division and separation. Each service is an individual component, and even though it requires a bit more effort to modularise the different parts, the benefits outweigh this negative very quickly as the project grows in size.*

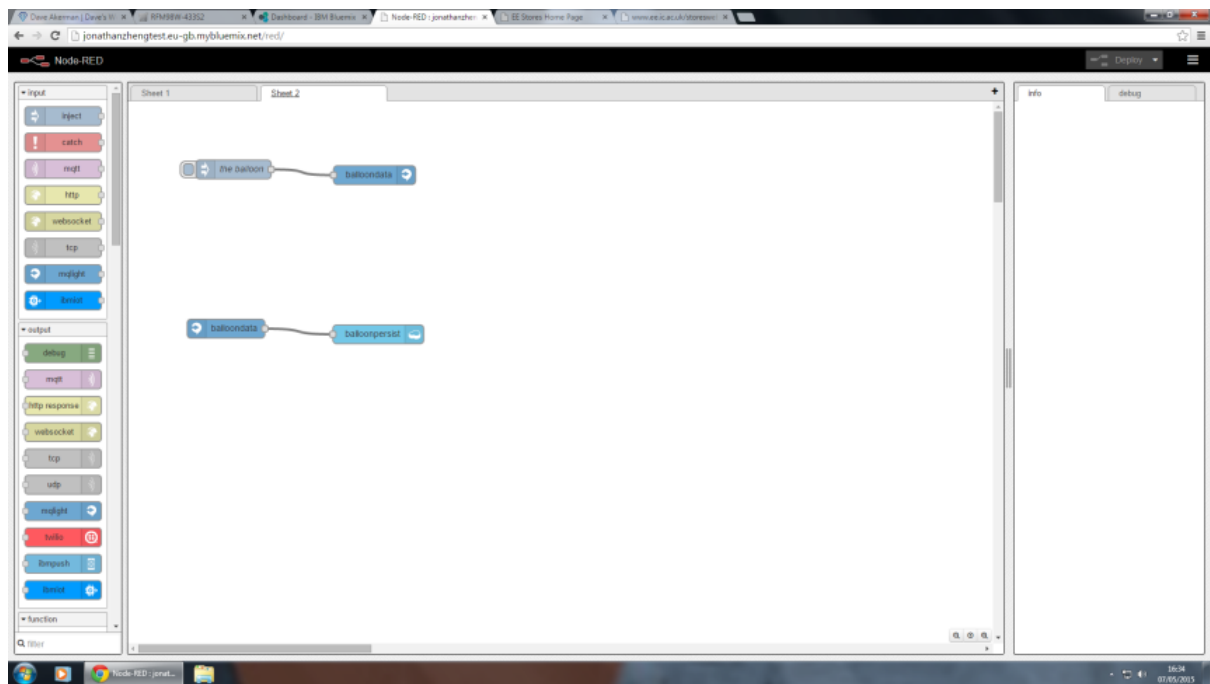
*This allows updating of services separately, compatibility allowing. It also means that if one service goes down, the others can adapt and manage. The case study of Netflix was often used! For example, If the microservice giving 'Top Ten Recommended For You' fails, this line on the website is simply absent. It can do this because every time it needs something it simply asks the microservice dedicated to that job; if it's down then the rest of the website can go on.*

*This architecture fits right in with our balloon. Each use case of our data can be mapped to a microservice. For example (clockwise from top of the diagram) our simulation room, twitter updates, online virtual simulation, data visualisation, database storing:*

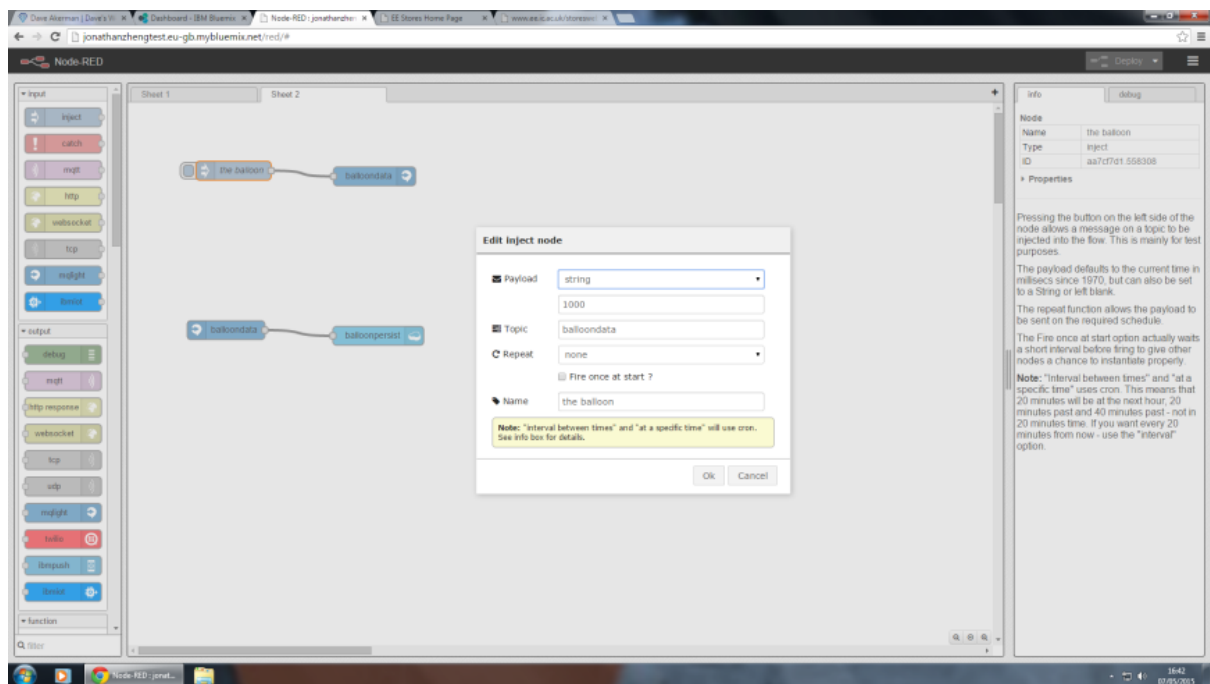


*The arbiter for the data in this architecture is messaging (with us using either MQLight or MQTT integration in Bluemix). MQ is a super reliable data handling system that uses publish/subscribe. This means that one microservice can publish data to a topic, and others who are subscribed to this topic receive the data. It's simple and removes the hassle of working out how different bits of your project query others.*

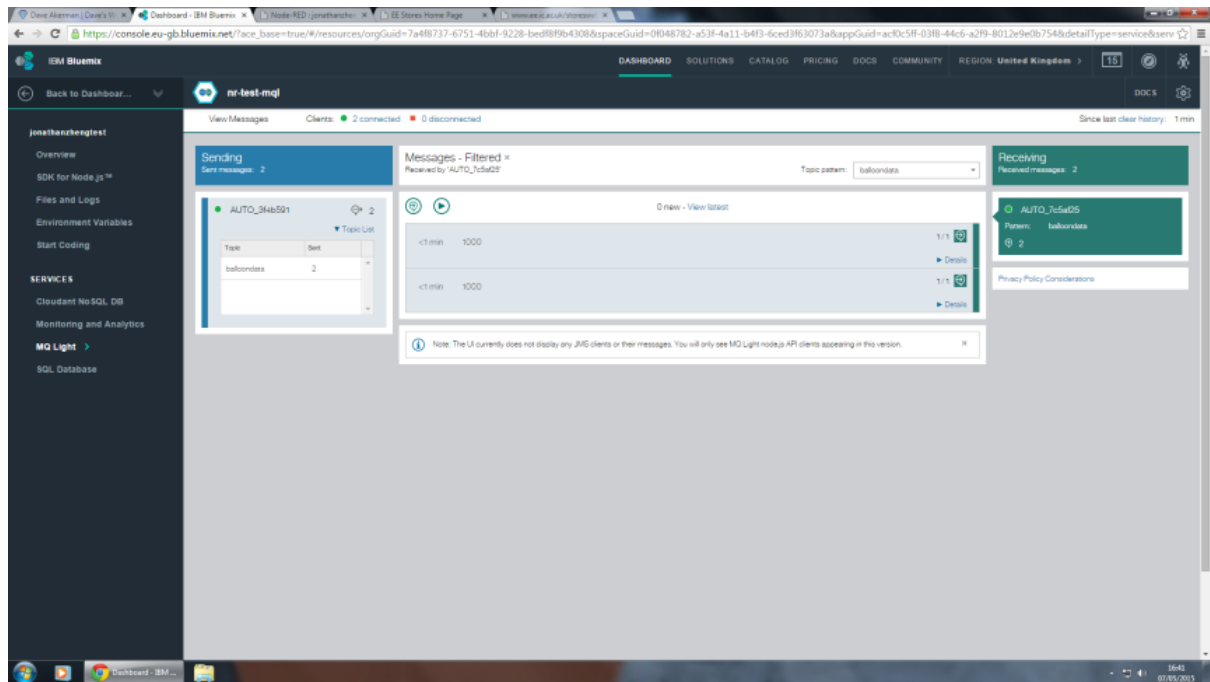
*Steve then showed us the real power of Bluemix (we were admittedly a bit sceptical beforehand) by throwing together a quick demo using node-red (Bluemix's node.js platform). He quickly made an MQLight service, and showed us how to simulate packets coming from a balloon and being stored in a database:*



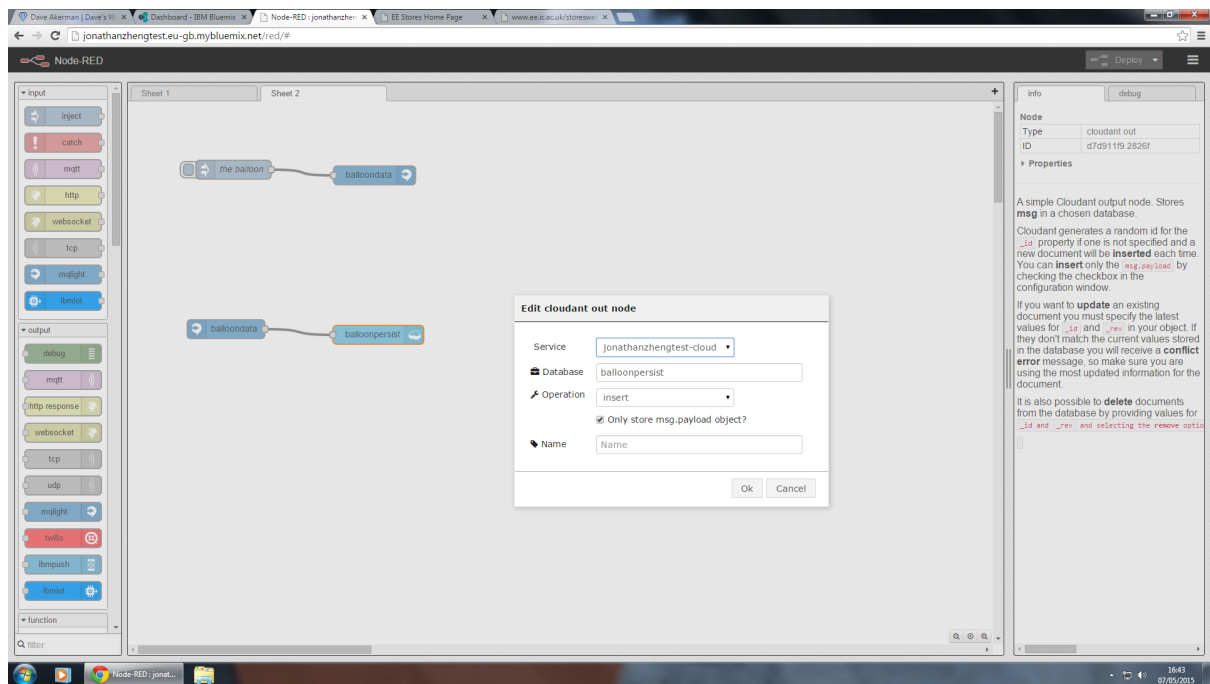
*With an inject node he gave a message a payload containing the string '1000' and the topic 'balloondata'. This is connected to an MQLight output node publishing to the same topic:*



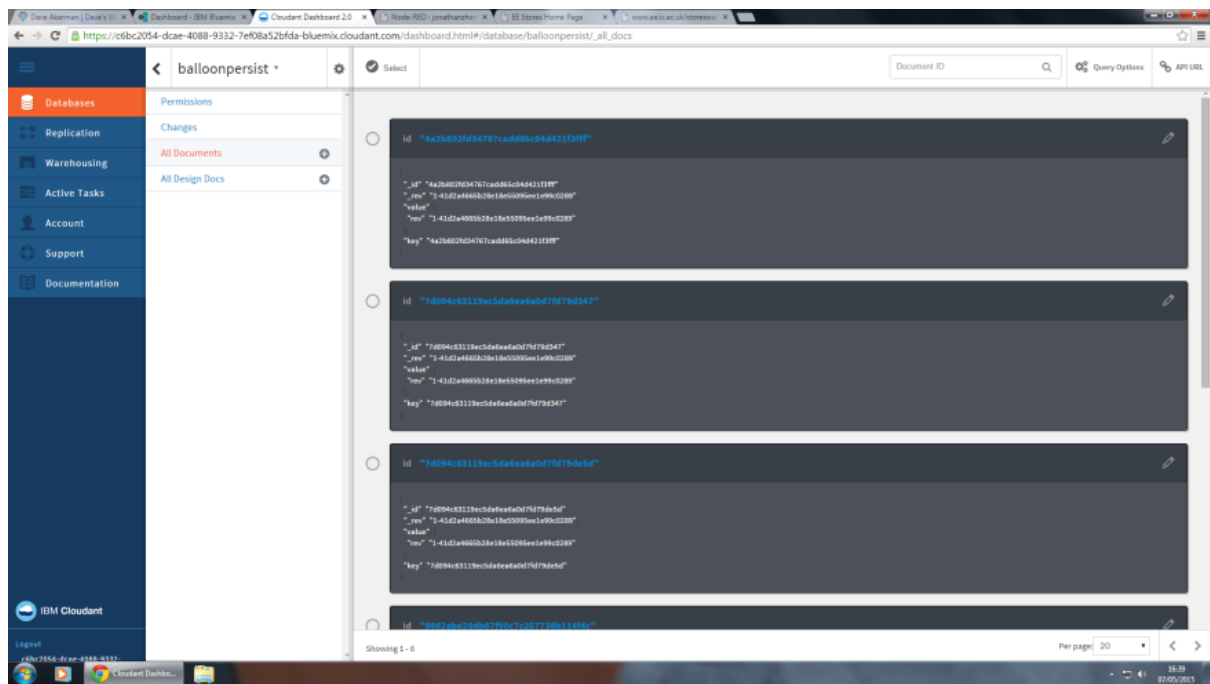
*When data is injected, you can view the messages in from the bluemix dashboard by viewing the MQLight service used:*



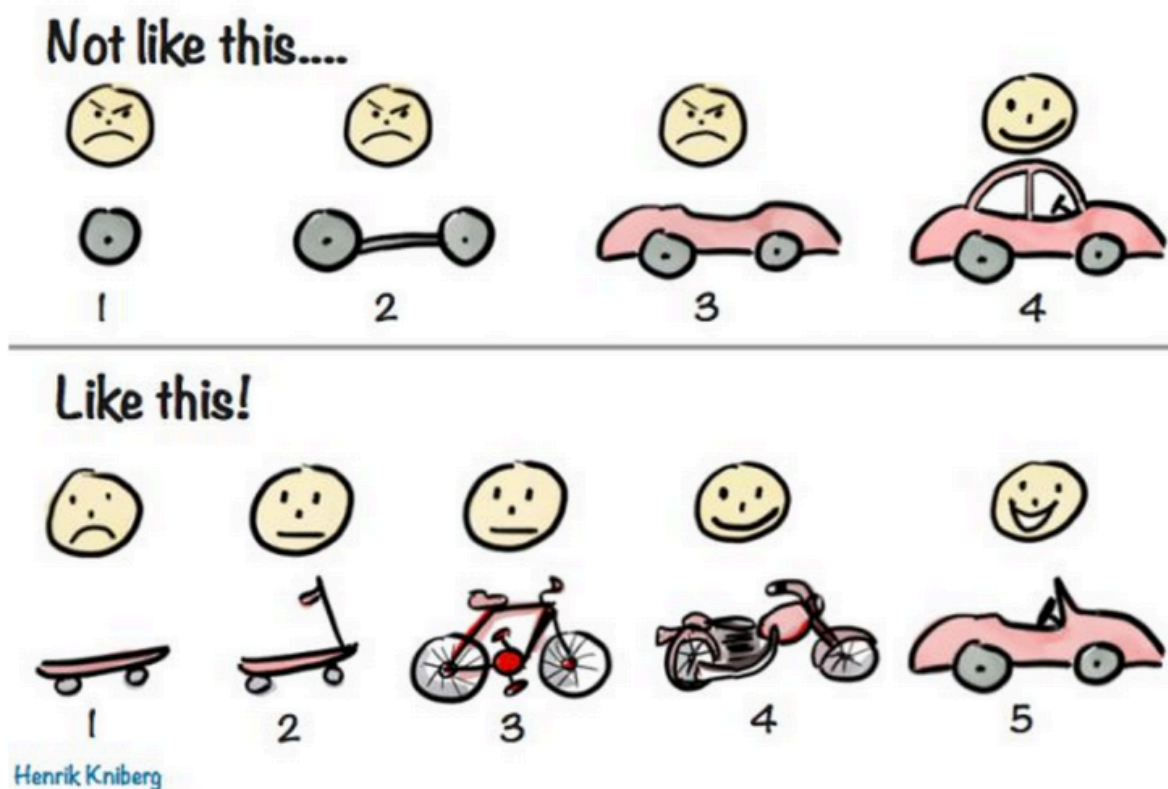
*An MQLight input node is then used to subscribe to these messages, and stored in a Cloudant database:*



*Finally, you can also view your beautiful messages from the dashboard:*



*So in 20 minutes he created a working template of the backbone of our project. No biggie. This shows how quickly you can get something up and running (however simple) on platforms like this. This is the embodiment of agile development. Get something working at each stage of development instead of spending hours and hours planning something that might not be what you even want in the end. This picture Steve showed us sums this up pretty well, and will impact our mindset throughout the course of this project.*



*Thanks for reading, our next aim is to get the Pi in the Sky sending us the data we need to make use of these tools.*

*-Jonny from theEdge of Space team.*

Try it for yourself here – <http://bit.ly/lightmessaging>

The team from Imperial are going to recount their adventures as well as their ‘how to’ collateral on their blog. Which I heartily recommend that you check out and follow...

<http://bit.ly/EdgeofSpace>

Check back here soon for more serialization of their posts.