

A photograph of a mountain landscape at twilight. In the foreground, a glowing orange tent is pitched on a rocky ground. The background features a large, rugged mountain peak with patches of snow, set against a sky with vibrant purple and blue hues. The overall scene is serene and adventurous.

MULE + TWITTER + LED



ALL ON AN IOT DEVICE CALLED A RASPBERRY PI

WELLINGTON MULESOFT
USER GROUP

2 NOVEMBER 2016

ABOUT ME



- Blockchain Integration + Training
- Founder of talkcrypto.org
- Co-organiser Wellington Blockchain Meetup
-  cloudnthings  Sean Au

ME AND MULESOFT

- Participated in community
- Trained Mulesoft in JIRA



IT ALL STARTED WITH A PI



#1.TURNING ON AN LED



- <https://www.anintegratedworld.com/how-to-turn-on-a-led-with-a-raspberry-pi/>

THEN

- Involved in a project to evaluate Mule ESB



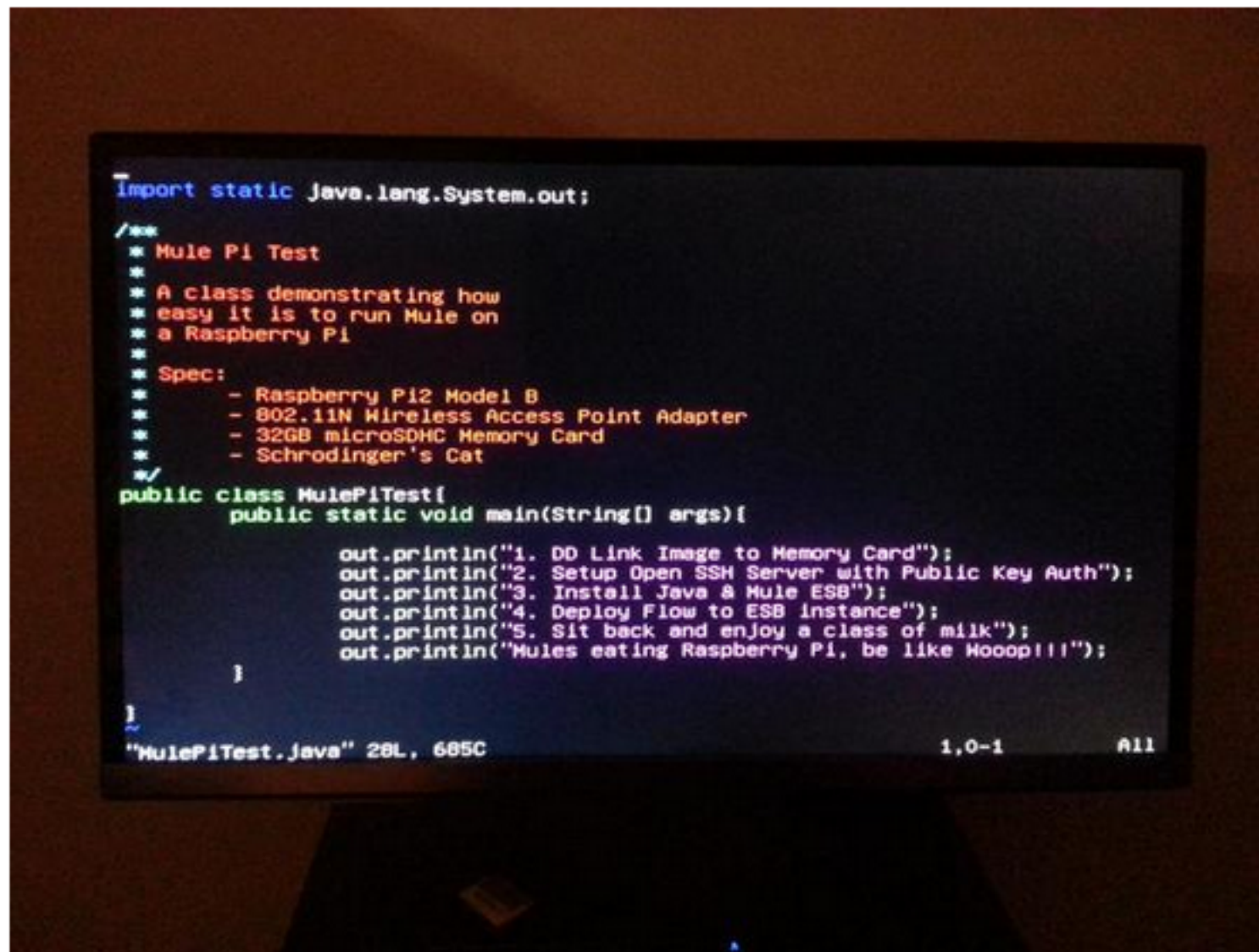
Keaton Victor

@keaton_victor



+ Follow

@MuleDev Mules eat Raspberry Pi.
Beautiful.

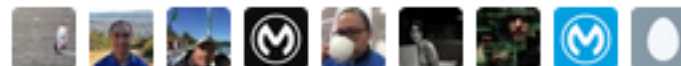


RETWEETS

10

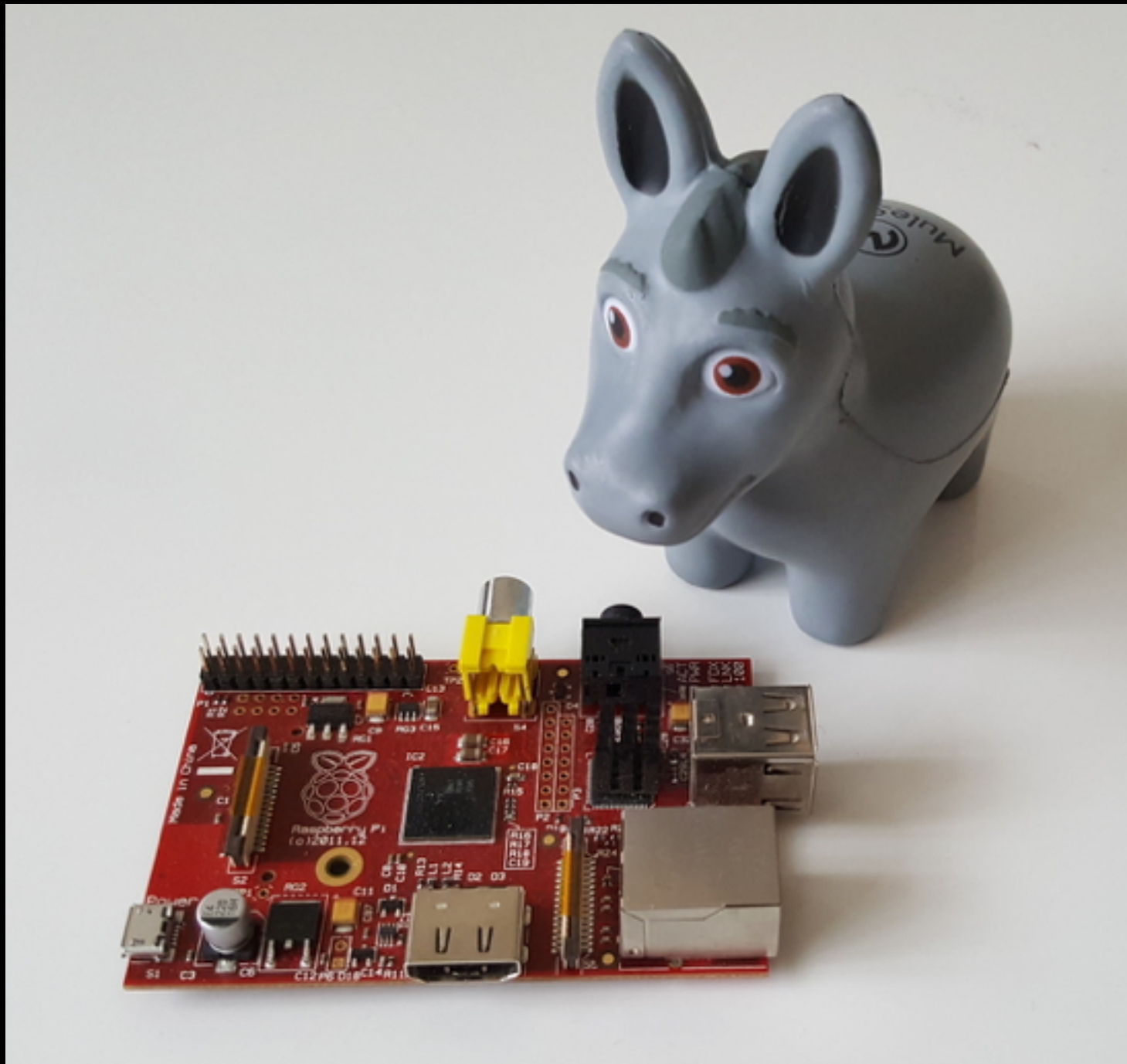
FAVORITES

4



11:07 AM - 11 Jul 2015





- <https://www.anintegratedworld.com/how-to-get-mule-esb-going-on-a-raspberry-pi-2/>

LIGHT BULB MOMENT

- 2 projects done, what next?



MY NEXT PROJECT

- Came across an article by Ryan Carter from MuleSoft

[Home](#) > [Blogs](#) > Polling and Watermarking in Mule



POLLING AND WATERMARKING IN MULE

Polling and Watermarking in Mule

by Ryan Carter Jan 28, 2015

[f0](#) [t5](#) [ln0](#) [g+0](#)

When it comes to synchronizing data between many systems, polling an API resource is an unfortunate inevitability. This results in developers calling the API over and over again to get updates, only to find out nothing has changed. This process constantly uses up resources and is not acceptable to either the API consumer or the API provider. In order to most efficiently poll an API, you keep track of where you last left off so we don't process the same data over and over again.

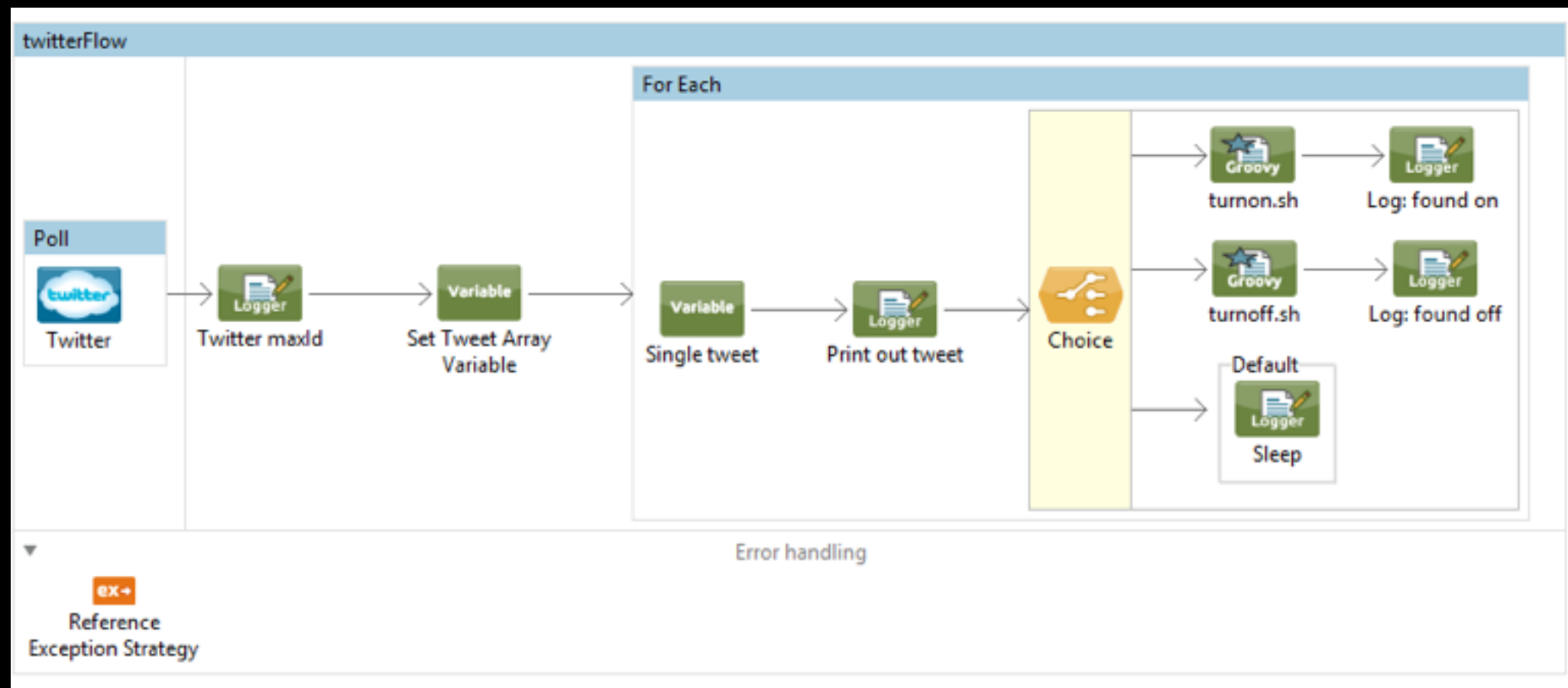
The term 'watermarking' is borrowed from floods, whereby you measure the watermarks on a surface to see how high the water rose. This resonates with synchronization when you need to measure how much of a particular dataset has already been processed. Watermarking allows us to pick up from where we left off without having to reprocess and filter out the old data that we do not care about anymore.

Let's look at an example using Twitter.

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <mule xmlns="http://www.mulesoft.org/schema/mule/core"
3   xmlns:http="http://www.mulesoft.org/schema/mule/http"
4   xmlns:twitter="http://www.mulesoft.org/schema/mule/twitter"
5   xmlns:spring="http://www.springframework.org/schema/beans"
6   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
7   xsi:schemaLocation="
8     http://www.mulesoft.org/schema/mule/http
9     http://www.mulesoft.org/schema/mule/http/current/mule-http.xsd
10    http://www.mulesoft.org/schema/mule/twitter
11    http://www.mulesoft.org/schema/mule/twitter/current/mule-twitter.xsd
12    http://www.springframework.org/schema/beans
13    http://www.springframework.org/schema/beans/spring-beans-3.0.xsd
14    http://www.mulesoft.org/schema/mule/core
15    http://www.mulesoft.org/schema/mule/core/current/mule.xsd ">
16
17   <twitter:config name="twitter" consumerKey="${twitter.consumer.key}"
18     consumerSecret="${twitter.consumer.secret}"
19     accessKey="${twitter.access.key}"
20     accessSecret="${twitter.access.secret}" />
21
22   <flow name="twitterWatermarking" processingStrategy="synchronous">
23     <poll frequency="30000">
24       <watermark variable="lastId" default-expression="#{0}"
25         update-expression="#{payload.sinceId}" />
26       <twitter:search query="mule" sinceId="#{flowVars['lastId']}" />
27     </poll>
28     <logger message="#{payload}" level="INFO" />
29   </flow>
30 </mule>
```

- <https://www.appnovation.com/blog/polling-and-watermarking-mule>

MULE FLOW



- <https://www.anintegratedworld.com/use-twitter-mulesoft-to-turn-on-an-led-on-a-raspberry-pi/>

Turning on an LED using Twitter and Mule ESB on a Raspberry Pi



- <https://www.youtube.com/watch?v=wDYRe07HztQ>

WHERE TO NEXT?

- Let's containerize it!



docker



MuleSoft®

SUMMARY

- Pi with an LED
- Mule on Pi
- Twitter -> Mule -> LED via Pi
- Let's containerise it!