How to morph a Raspberry Pi into a Powerful Sensor Data Cache and IoT Gateway

Session B06

Alexander Koerner (akoerner@de.ibm.com)

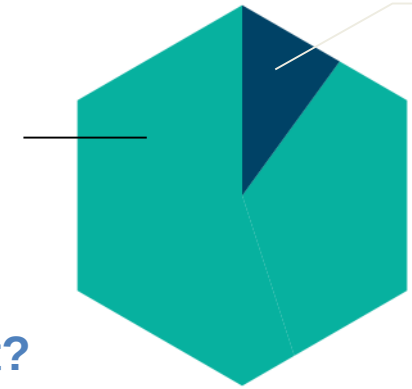
Agenda

- Introduction
- Why a Raspberry Pi?
- How to build a sensor data cache with a RPi
- Sensor data analytics in motion and at rest
- The cloud connection
- Sensor data visualization
- Summary

And we're only beginning

Extabytes of data are generated by the IoT every day.

88 percent



Think of all we've accomplished using only 12 percent of the data generated by the IoT.

What will we achieve as we unlock the rest?

Some IoT-/Edge-Gateway Use Cases...

Industrial Production

- Local Shop Floor Sensor Data (Pre-)Processing / Cleansing / Monitoring (Predictive)
- Shop Floor Communication Protocol consolidation (Industrial Cloud Gateway with Analytic Capabilities)

Automotive

- On-Vehicle Data Buffering / Processing / Monitoring (Predictive)

Smart Facility-/Building-Management

- In-Building Sensor and Actor Data Collection and Control
- Local Energy Optimization Capabilities with Cloud Access for more Complex Optimizations

Sports Statistics / Fan Experience

- In-Stadium Sensor based Statistics (Ball, Player and Referee Positions, Movements, Goals/Touchdowns)

Wind Turbine Monitoring

- Off-Site or Off-Shore locations with limited communication bandwidth require local operation data monitoring to optimize preventive maintenance

A typical „IoT-/Industry 4.0-Gateway“ Scenario

Industrial Shop Floor Sensors

Internet of Things
IBM

IBM Bluemix™

Time Series Database
IBM

Raw Sensor Data
via industry specific
automation protocols
(e.g. PROFINET, BACnet,
OPC UA, Modbus etc.)

Aggregated / Filtered Sensor Data
via standard communication protocols
and formats (e.g. MQTT/JSON) into
the Cloud (IBM Bluemix, IBM IoT Foundation),
IBM PMQ plus on premises solutions etc.

Industrial IoT-/Edge-Gateway(s)
with specific industry communication stacks
and local Analytic capabilties (in motion / at rest analytics)

Agenda

- Introduction
- Why a Raspberry Pi?
- How to build a sensor data cache with a RPi
- Sensor data analytics in motion and at rest
- The cloud connection
- Sensor data visualization
- Summary

Why a Raspberry Pi? 1/2

- Pros:
  - Very reasonably priced: between about 5 USD (RPi Zero) and about 40 USD (RPi 3)
  - Widely available
  - Very strong and active WW community
  - Different OS choices and lots of software available
  - Lots of HW extensions available (e.g. for sensor data processing and specific communication protocols)
  - Low power consumption.
    - Can be easly powered by a portable battery pack
  - Informix 12.10 is available on the Raspberry Pi!!! ☺

Why a Raspberry Pi? 2/2

- Cons:
  - Out-of-the-box lacks certain industry certfications (e.g. for automotive use) → Industry certifications are expensive!
  - Currently not RAM extendable (even not on the 64bit RPi3)
  - SD card storage is not reliable in an industry context

Available Raspberry Pis

- Rapberry Pi Zero
  - 1000 MHz single-core ARM ARM1176JZF-S , 512 MB RAM

- Raspberry Pi 2
  - 900 MHz quad-core ARM Cortex A7, 1 GB RAM

- Raspberry Pi 3
  - 1200 MHz quad-core ARM Cortex-A53 64bit, 1 GB RAM, Built-In Bluetooth & WiFi

Not listed is the original RPi 1 (Models A&B)

Agenda

- Introduction
- Why a Raspberry Pi?
- How to build a sensor data cache with a RPi
- Sensor data analytics in motion and at rest
- The cloud connection
- Sensor data visualization
- Summary

How to build a sensor data cache with a Raspberry Pi

- Choice of communication protocol(s)
  - Sensor-to-Pi communication
  - Pi-to-Network communication
- Sensor HW considerations
- Operating System(s)
- Additional Software requirements
  - Sensor Data Cache (aka Informix ☺)
  - Flow / Streams Processing
- Additional building blocks
  - LED / LCD Displays

Communication protocols: Sensor-to-Pi

- **Bluetooth**
  - Suitable for limited range sensors (e.g. Ti SensorTag CC2650)
  - Typically BT 4.0 LE
  - RPi Zero, RPi 1 and RPi 2 need an BT 4.0 LE dongle. RPi 3 has one built-in!
- **ZigBee**
  - Low cost, low power wireless network
  - ZigBee modules for RPi available
- **EnOcean**
  - Supports batteryless wireless sensors and controllers. Some with solar cells.
  - EnOcean communication module(s) are available for the RPi
- **Homematic / BidCos**
  - Quite popular propriatry protocol for smart home applications in Germany (868.3 Mhz band)
- **Realtime Ethernet**
  - Industry protocols like PROFINET, EtherNet/IP or EtherCAT
  - Realtime Ethernet adaptor for RPi available from Hilscher (Germany)
- **And many more...**

Communication Protocols: Pi-to-Network

- The Raspberry Pi 1 (B) and 2 only have on-board Ethernet
  - You would need an USB WiFi dongle for WLAN connectivity
- The Raspberry Pi 3 has on-board Ethernet, WiFi and Bluetooth 4.0 LE
- Tip for the Raspberry Pi Zero (who has no on-board network): get a combi USB dongle w/ built-in WiFi

Sensor HW considerations

- Choose your sensor HW based on your use cases, sensor HW availability and cross vendor protocol compatibility
  - Environmental monitoring / Weather station(s)
  - Smart home monitoring / control
  - On-board, on-vehicle monitoring
- To get started, for demo environments and/or to simply play with the technology
  - Get an integrated, combined sensor like the Ti SimpleLink SensorTag CC2650

  - or get e.g. the Raspberry Pi Sense HAT (w/ integrated LED display)

Operating System(s)

- There is a very large selection of OS for the Raspberry Pi family
- The majority is **Linux** based, but there is also a free **Windows 10 IoT Core** from Microsoft available
  - Requires at least an RPi2
- The most popular choice is **Raspian** which also seems to have the broadest community support
- Choose your RPi OS based on the best support for your desired communication protocols, RPi HW and use case(s)

Software Requirements: Informix

- A sensor data cache needs a optimized sensor database: Informix 12.10
  - Built-in time series data (aka sensor data) support
  - Low install, memory and operation footprint
  - Free Developer Edition available for ARM v6 → Raspberry Pi w/ Raspian
  - Download:
    - https://www-01.ibm.com/marketing/iwm/iwm/web/reg/pick.do?source=ifxids&S_TACT=109HF16W
- Also supports the REST API and JSON documents
  - Both features are key for a sensor data cache

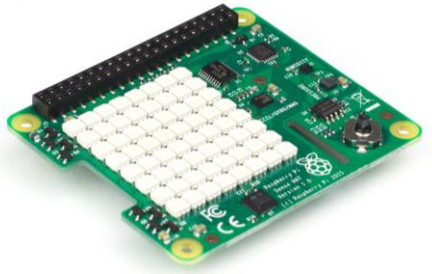
Flow-/Streams-Processing

- Sensor data often arrive in a continous stream
- That data stream might need some on-the-fly cleansing, format conversion, time stamp additions and/or aggregation
- A very easy to use option for flow processing on the RPi
  - Node-RED (based on the MQTT/JSON concepts, Open Source, by IBM)

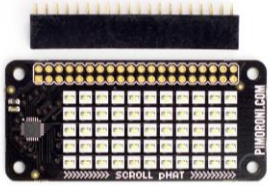
- And a very powerful option for streams processing on the RPi
  - Quarks (similar to IBM Streams, Open Source, by IBM)

Additional Building Blocks

- If the Raspberry Pi will be operated in a headless mode (no regular screen connected), then a small LCD/LED display can be helpful
  - Display e.g. the IP address, sensor data etc.
- Two suggestions (I am personally using both):
  - Raspberry Pi Sense HAT (8x8 multi color LED matrix display with integrated sensors and a "joystick")



  - Scroll pHAT (11 x 5 white LEDs, perfect size for the RPi Zero)



Using the Sense Hat to display weather data

Scroll pHAT on a Raspberry Pi Zero with Informix 12.10.UC6

Agenda

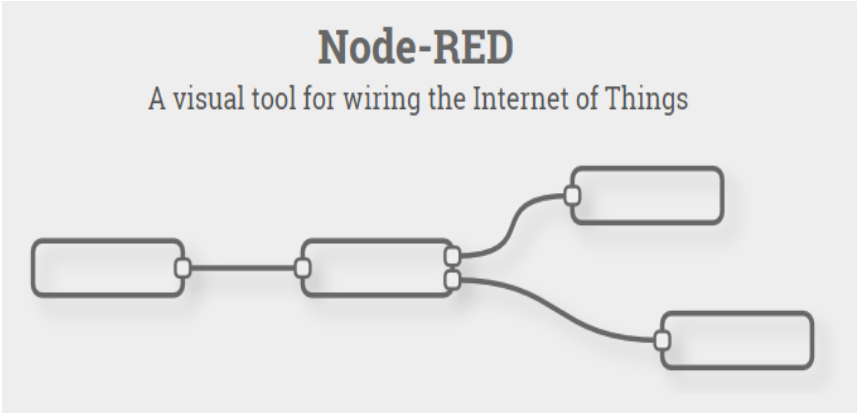
- Introduction
- Why a Raspberry Pi?
- How to build a sensor data cache with a RPi
- Sensor data analytics in motion and at rest
- The cloud connection
- Sensor data visualization
- Summary

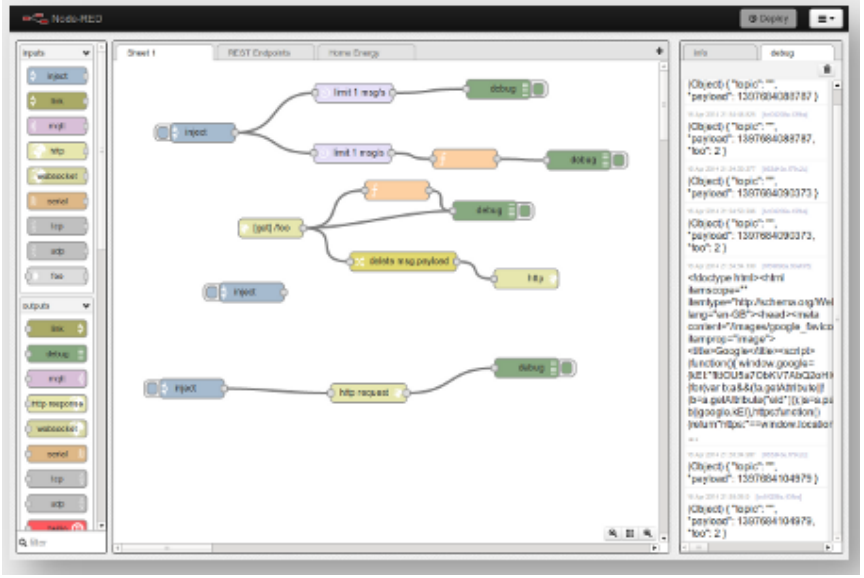
Sensor data analytics in motion and at rest

- In-Motion Sensor Data Analytics
  - Node-RED
  - Quarks
  - Spark Streaming (not covered)
- At-Rest Sensor Data Analytics
  - Informix 12.10 ☺

What is Node-RED?

- nodered.org
- A visual tool for wiring the Internet of Things
- Node.js application. Uses building blocks of reusable code to control flow of data

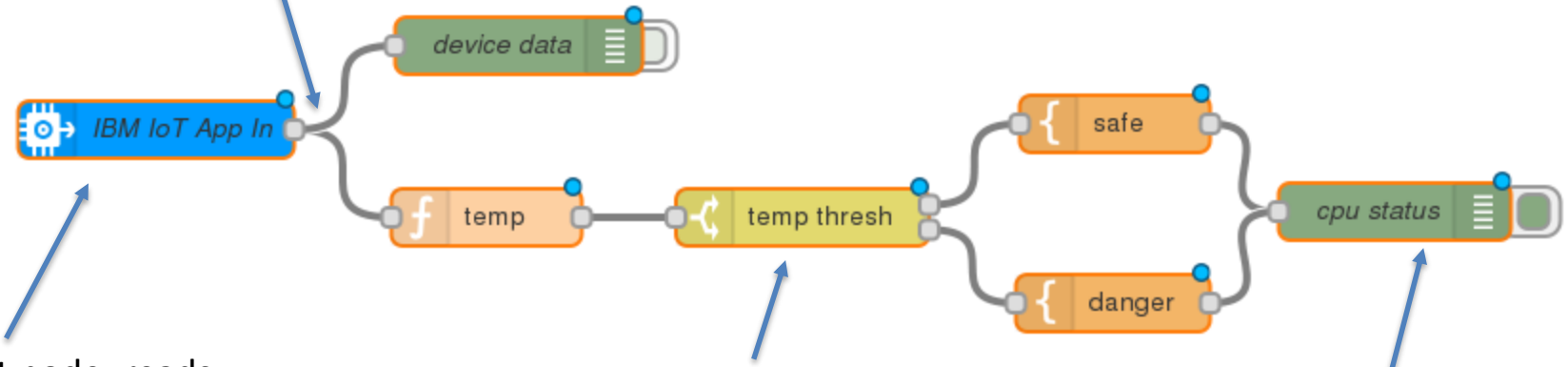




- "Flows" are created to process data.
- Example: Take data in, transpose, process and store in a database and/or publish to a message broker
- Collaboration among node developers

Node-RED: Flows and Nodes

These wires are attached to the same terminal, so output from "IoT App in" is copied to both receiving nodes



Input node: reads events from IoTF

Switch node: routes incoming messages to one or more of its output terminals

Debug node: writes output to debug window

Node-RED implements a graphical Data Flow language.
Flows can be imported or exported as JSON objects

Introducing QUARKS

- Quarks: Programming model and runtime for analytics at the edge
  - Quarks programming SDK
  - Quarks lightweight and embeddable runtime
  - Quarks initially built by IBM based on 12 years experience with IBM Streams
  - Quarks is Open Source, available on github
  - Proposal submitted for Quarks to Apache Incubator project

Features

- Java APIs for developing streaming applications
- Windowing support aggregation
- Micro-kernel style runtime for execution
- Connectors for MQTT, HTTP, JDBC, File, Apache Kafka and IBM Watson IoT Platform
- Simple analytics and pattern detection device sensors
- Web-console to view the graph of running applications
- Integration with assertion based testing systems like JUnit
- Android support, e.g., streaming a phone's sensor events
- Multi-Platform support including Java 8, Java 7 and Android

Edge Language Support

- Initial support

  Java

  •Easy to get running, demonstrate the concept on many platforms
  - Java 8 running on Raspberry Pi B & Pi2 B today (ARM chip)

  Android

  •Smart phones and increasing adoption for devices

- Will support multiple languages

  •Integrated vision, not a specific edge language

- Other options for the future

  C, **C++** – Generate code for customer to compile to support custom kernels
  Open Swift - iOS
  Python – Python runs on Raspberry Pi, popular language on Pi
  Streams Processing Language – Simpler way to declare a flow and produce C/C++ code

Language priorities driven by Open Source Community

Informix 12.10 on the Raspberry Pi

- Informix is the perfect sensor data store for edge gateways based on the Raspberry Pi
  - Small footprint (installation, memory & cpu utilization, sensor data storage size)
  - Optimized for sensor data handling (time series)
  - REST and JSON support out-of-the-box
  - Available on Raspberry Pi CPUs (ARMv6)
  - Does integrate well with Node-RED and Quarks

Informix on the RPi – How To 1/6

- Use a Raspberry 3 or 2 since they have 1GB memory built-in
  - The Raspberry Pi 3 has built-in WiFi and BT 4.0 LE!
- For special gateway requirements you can also choose a Raspberry Pi Zero
  - But you would need some micro USB network dongle to connect to the outside
- Download and transfer the latest Raspian relase ("Jessie") to a microSD card (minimum 8GB, recommendation 16GB+)
  - Refer to Jean George‘s great microSD benchmark article to choose an optimal card
    - http://jgp.net/microsd-card-performance-raspberry-pi/

Informix on the RPi – How To 2/6

- For the first start up connect your Pi via LAN cable to a network which is ideally connected to the Internet
- Update Raspian with the latest binaries
  - `sudo apt-get update`
  - `sudo apt-get upgrade`
- Download the Informix Developer Edition for ARMv6 and transfer it to your RPi
  - https://www-01.ibm.com/marketing/iwm/iwm/web/reg/pick.do?source=ifxids

Informix on the RPi – How To 3/6

- Install and configure Informix on the Raspberry Pi
  - Simply follow my article series on www.raspberrypi.org:

Informix Introduction and Installation on the RPi

  - https://www.raspberrypi.org/forums/viewtopic.php?f=37&t=97199&p=674959

- That first part will create a very basic Informix installation on the RPi w/o any database schema

Informix on the RPi – How To 4/6

- Now let's create a database with a schema suitable for sensor data

  An Informix Sensor DB - Part 1

  – https://www.raspberrypi.org/forums/viewtopic.php?f=37&t=97772&p=678497

- For full schema flexibility, I would recommend to choose the option with a JSON based time series

  – You'll never know how your sensor data will look in the future ☺

Informix on the RPi – How To 5/6

- Optimize the sensor data storage by introducing a round robin time series container
- The "rolling window" container(s) limit the physical storage requirements by automatically deleting the oldest sensor data entries → Important on a limited resources Raspberry Pi!

An Informix Sensor DB - Part 2

- https://www.raspberrypi.org/forums/viewtopic.php?f=37&t=100029&p=693935

Round robin sensor data storage with IBM Informix

- https://www.raspberrypi.org/forums/viewtopic.php?f=37&t=140398

Informix on the RPi – How To 6/6

- Finally we should install the Informix REST listener
- Node-RED requires the REST API in order to access the sensor data in Informix

  The Informix REST API

  https://www.raspberrypi.org/forums/viewtopic.php?f=37&t=137392&p=912405

- Please note that I am explicitly limiting the REST listener resources in my article (above)
  - The Informix Developer Edition only supports 20 concurrent sessions!
  - The Raspberry Pi has (again) limited resources

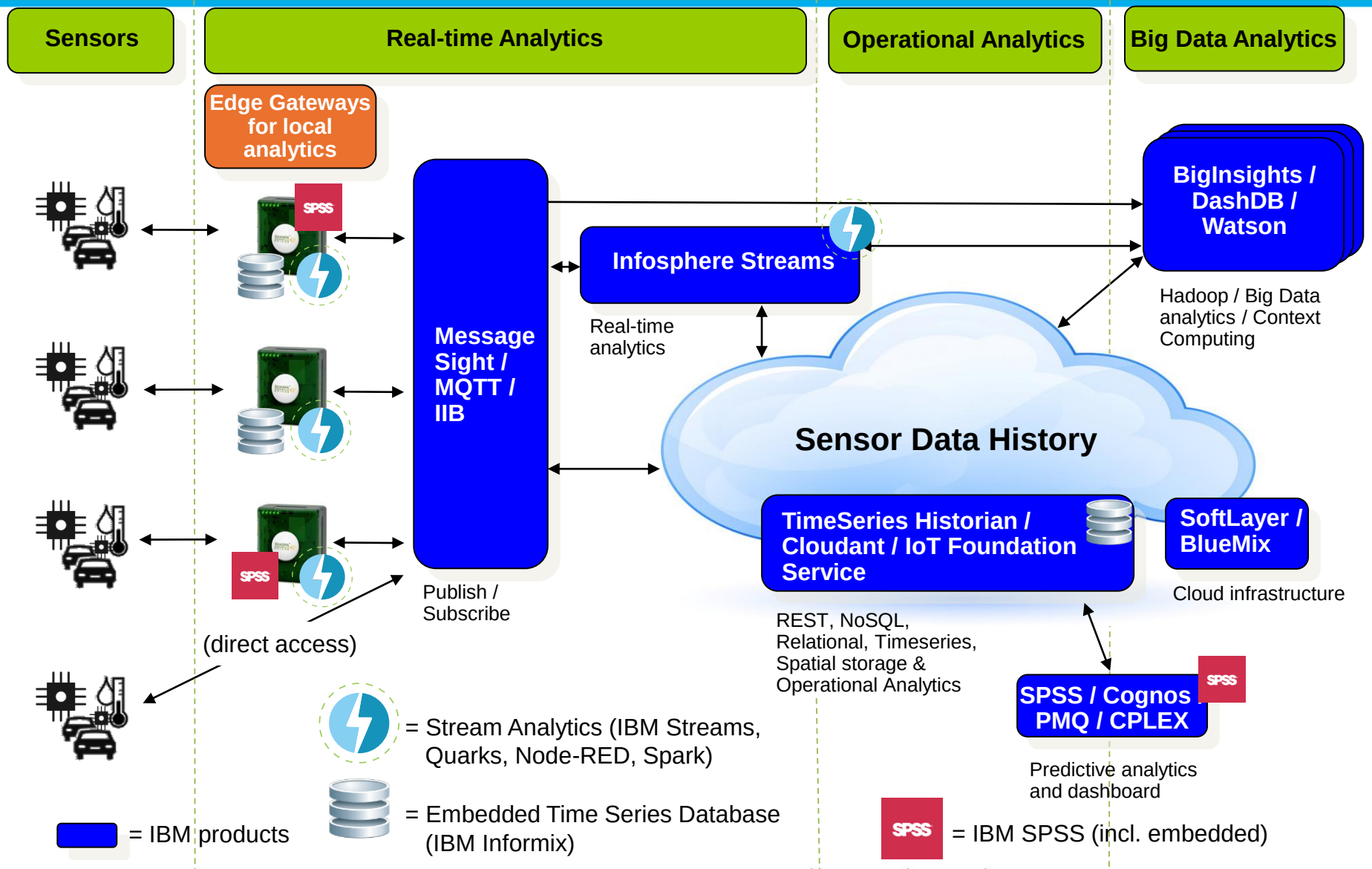
Node-RED & Informix Node-RED Nodes

- Node-RED comes pre-installed with Raspian "Jessie"
  - http://nodered.org/docs/hardware/raspberrypi
- Install the Node-RED package installer "npm"
  - `sudo apt-get update`
  - `sudo apt-get install npm`
- Install the Informix Time Series Node-RED nodes
  - `npm install node-red-contrib-timeseries`
    - (local install)
  - `sudo npm install -g node-red-contrib-timeseries`
    - (global install: the nodes will be accessible to all users on the Raspberry Pi)
- Start/Stop Node-RED
  - `node-red-start`
  - `node-red-stop`
- Access Node-RED
  - http://localhost:1880 (or instead of localhost the IP address of your Raspberry Pi)

Agenda

- Introduction
- Why a Raspberry Pi?
- How to build a sensor data cache with a RPi
- Sensor data analytics in motion and at rest
- The cloud connection
- Sensor data visualization
- Summary

IoT Architecture Annotated with IBM Products



The Watson IoT Platform – IBM's next generation IoT platform

IBM IoT Foundation Offerings

IBM IoT Foundation Connect
Attach, Collect & Organize, Device Management, Secure Connectivity, Visualization

IBM IoT Foundation Information Management
Storage & Archive, Metadata Management, Reporting, Streaming data, Parsing and Transformation, Manage unstructured data

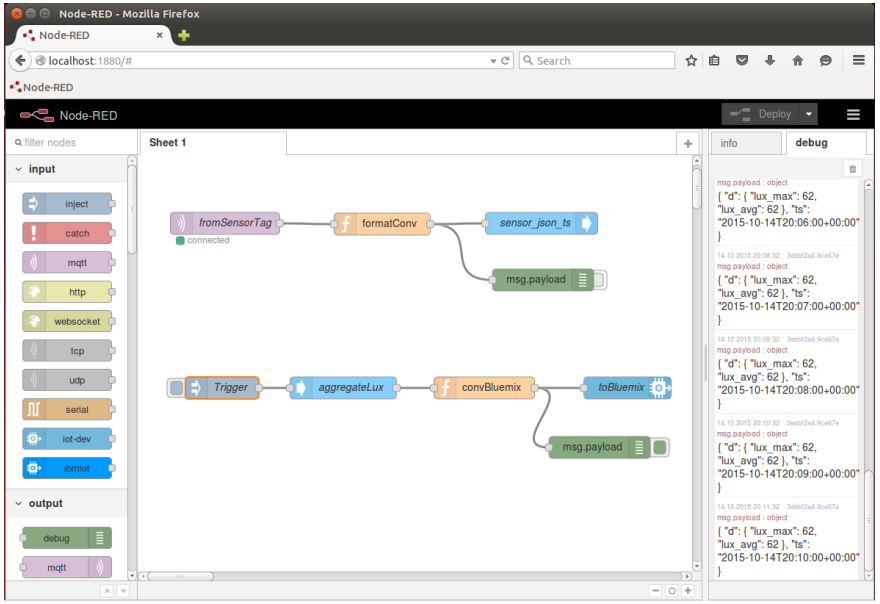
IBM IoT Foundation Analytics
Predictive, Cognitive, Real-time, and Contextual

IBM IoT Foundation Risk Management
Security Analytics, Data Protection, Auditing/Logging, Firmware Updates, Key/Cert Mgmt, Org Specific Security

How to connect the RPi to the IBM Cloud

- Protocol of choice: MQTT
  - **M**essage **Q**ueue **T**elemetry **T**ransport
  - Light weight IoT communication protocol

- Tool of choice: Node-RED with an 'iot-dev' node
  - Node-RED is centered around MQTT
  - The 'iot-dev' node connects to IBM's IoT Foundation



Some IoT Foundation specifics

- The IoT Foundation requires the following payload format (in JSON):

```
{"d":
    { "lux_max": 49,
      "lux_avg": 49 },
 "ts": "2015-10-13T13:10:30+00:00"}
```

- Install the 'iot-dev' nodes

```
sudo npm -g install node-red-contrib-iotclouddev
```

Agenda

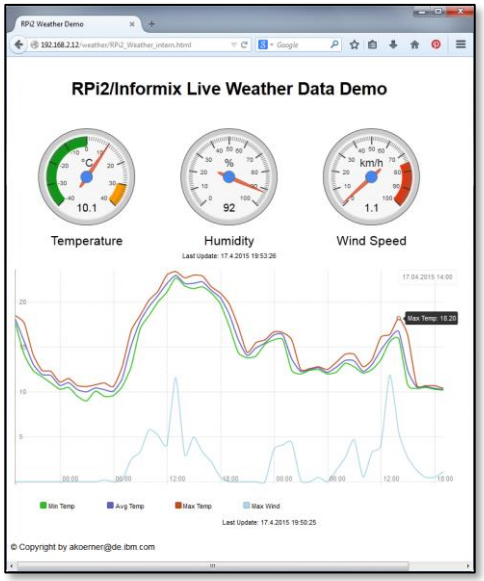
- Introduction
- Why a Raspberry Pi?
- How to build a sensor data cache with a RPi
- Sensor data analytics in motion and at rest
- The cloud connection
- Sensor data visualization
- Summary

Sensor data vizualization

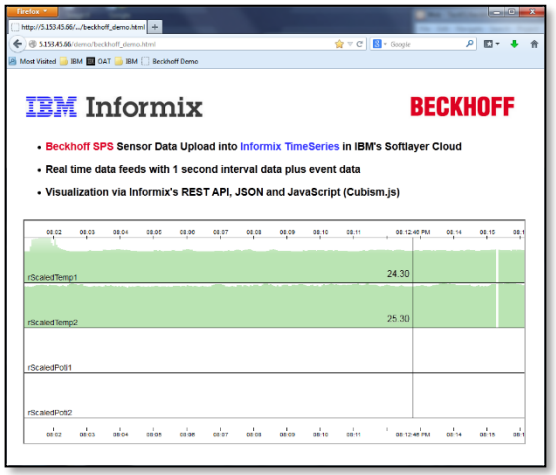
- JavaScript Based Vizualizations
- Grafana

JavaScript based Visualizations

- D3.js framework
  - https://d3js.org/
  - Very popular JavaScript library to vizualize data
  - Graph: rickshaw.js
    Gauges: Google Charts



- Cubism.js
  - Specialized Time Series 'real time' vizualization based on D3.js
  - https://square.github.io/cubism/

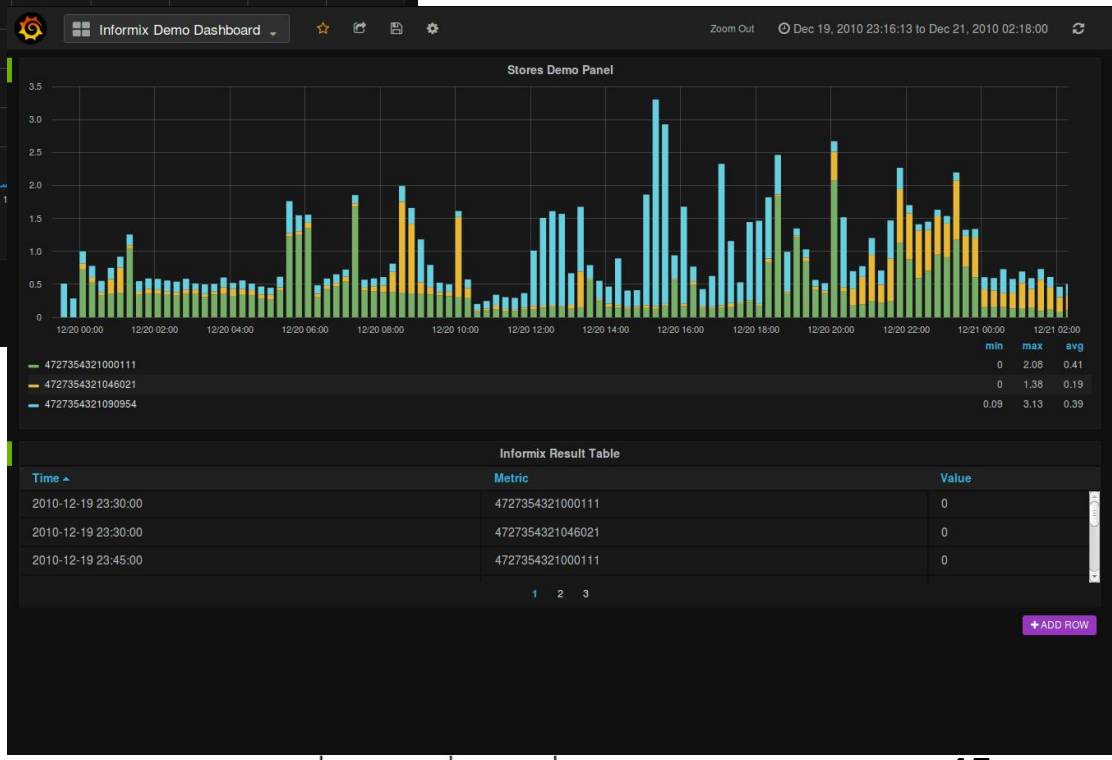


Grafana 1/3

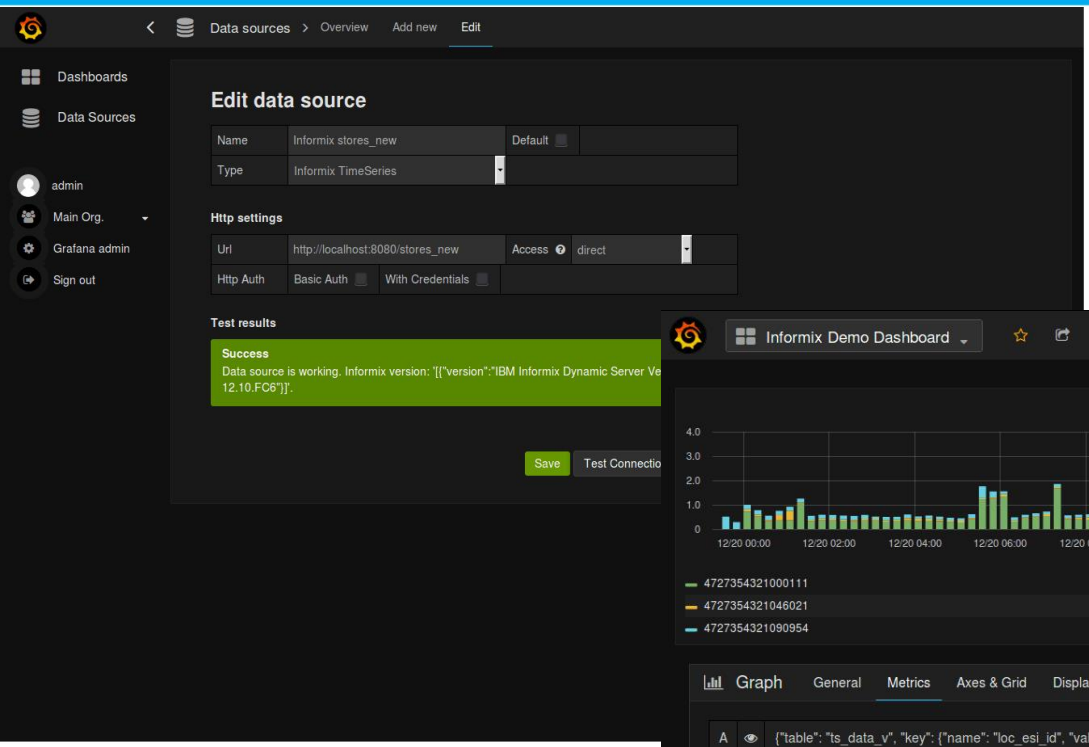
- Very popular open source tool to vizualize sensor data
  - http://grafana.org/
- Easy to create sensor data dash boards
- Currently out-of-the box support for very specialized databases like Graphite, InfluxDB, Prometheus etc.
- Currently no support for generic SQL databases
- Plug-in architecture for custom extensions (like other data sources)
- I started to work on a simple Informix TS plug-in for Grafana 2.6 based on Informix's REST API... ☺

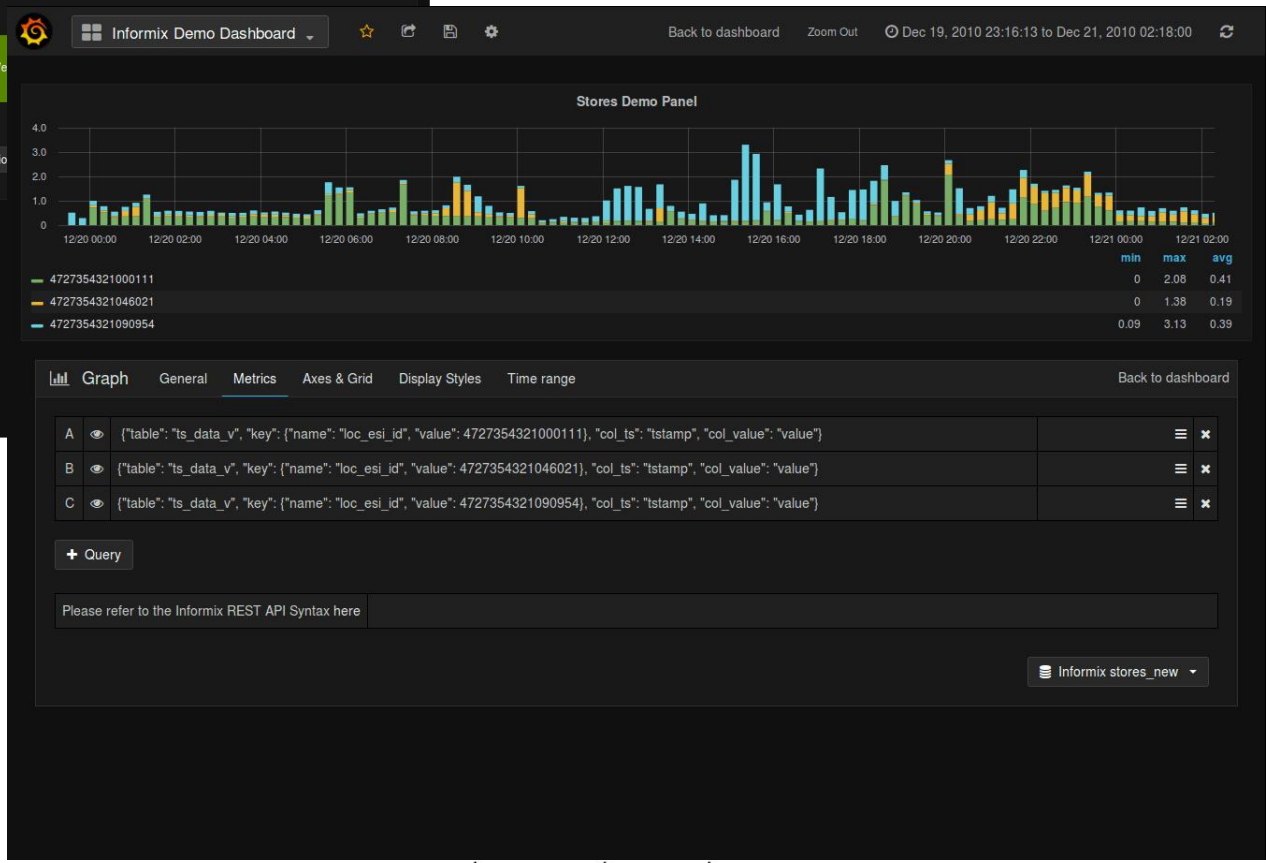
Grafana 2/3


Informix Munich Weather Data
Zoom Out
Jan 11, 2016 13:04:02 to Jan 25, 2016 18:25:00
Informix Munich Weather Data (Temperature)
Temperature
Wind Speed
15 °C
10 °C
5 °C
0 °C
-5 °C
-10 °C
-15 °C
1/12
1/13
1/14
1/15
1/16
1/17
1/18
1/19
1/20
1/21
1/22
1/23
1/24
1/25
min
max
avg
ELV_Weathersensors/t9
-12 °C
13 °C
-1 °C
Informix Munich Weather Data (Wind Speed)
Wind Speed
50 km/h
40 km/h
30 km/h
20 km/h
10 km/h
0 km/h
ELV_Weathersensors/w9


Informix Demo Dashboard
Zoom Out
Dec 19, 2010 23:16:13 to Dec 21, 2010 02:18:00
Stores Demo Panel
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0
12/20 00:00
12/20 02:00
12/20 04:00
12/20 06:00
12/20 08:00
12/20 10:00
12/20 12:00
12/20 14:00
12/20 16:00
12/20 18:00
12/20 20:00
12/20 22:00
12/21 00:00
12/21 02:00
min
max
avg
4727354321000111
0
2.08
0.41
4727354321046021
0
1.38
0.19
4727354321090954
0.09
3.13
0.39
Informix Result Table
Time
Metric
Value
2010-12-19 23:30:00
4727354321000111
0
2010-12-19 23:30:00
4727354321046021
0
2010-12-19 23:45:00
4727354321000111
0
1
2
3
ADD ROW

Grafana 3/3


Data sources
Overview
Add new
Edit
Dashboards
Data Sources
admin
Main Org.
Grafana admin
Sign out
Edit data source
Name
Informix stores_new
Default
Type
Informix TimeSeries
Http settings
Url
http://localhost:8080/stores_new
Access
direct
Http Auth
Basic Auth
With Credentials
Test results
Success
Data source is working. Informix version: "[{"version":"IBM Informix Dynamic Server Ve
12.10.FC6"}]".
Save
Test Connectio


Informix Demo Dashboard
Back to dashboard
Zoom Out
Dec 19, 2010 23:16:13 to Dec 21, 2010 02:18:00
Stores Demo Panel
4.0
3.0
2.0
1.0
0
12/20 00:00
12/20 02:00
12/20 04:00
12/20 06:00
12/20 08:00
12/20 10:00
12/20 12:00
12/20 14:00
12/20 16:00
12/20 18:00
12/20 20:00
12/20 22:00
12/21 00:00
12/21 02:00
min
max
avg
4727354321000111
0
2.08
0.41
4727354321046021
0
1.38
0.19
4727354321090954
0.09
3.13
0.39
Graph
General
Metrics
Axes & Grid
Display Styles
Time range
Back to dashboard
A
{"table": "ts_data_v", "key": {"name": "loc_esi_id", "value": 4727354321000111}, "col_ts": "tstamp", "col_value": "value"}
B
{"table": "ts_data_v", "key": {"name": "loc_esi_id", "value": 4727354321046021}, "col_ts": "tstamp", "col_value": "value"}
C
{"table": "ts_data_v", "key": {"name": "loc_esi_id", "value": 4727354321090954}, "col_ts": "tstamp", "col_value": "value"}
+ Query
Please refer to the Informix REST API Syntax here
Informix stores_new

Agenda

- Introduction
- Why a Raspberry Pi?
- How to build a sensor data cache with a RPi
- Sensor data analytics in motion and at rest
- The cloud connection
- Sensor data visualization
- Summary

Summary

- The Raspberry Pi family is a great foundation to build an IoT edge gateway
- It might not be suitable for tough production environments
- The requirements for sensor data in-motion and at-rest can be easily addressed on the Pi
  - In-motion: Quarks, Node-RED, Spark Streaming
  - At-rest: Informix
  - For complex analytics: SPSS 'Nuggets'
- Consider using Docker to deploy the components!

Thank You! ☺


IBM
Alexander Koerner
Analytics Architect
IBM Analytics
Industrie 4.0 & IoT
Informix SME
Hollerithstrasse 1
D-81829 Muenchen
Mail: Postfach 820364
D-81803 Muenchen
Phone +49-89 4504 1423
Fax +49-160 90737756
Mobile +49-160 90737756
akoerner@de.ibm.com