

INF5120 – Spring 2019

Oblig 3 Group Delivery – Smart Home – Support services for Elderly being able to stay longer at home"

Node-RED Design – ArchiMate Application/Technology layers/views, also with UML/BPMN– and Demonstration 10%

Delivery and group presentations May 13th, 2019

Oblig 3 focus:

Extend from your group work from Oblig 1 on "Smart Home - Support services for Elderly being able to stay longer at home " Strategy/Motivation models, use cases and Business Models – to create a Node-RED design of your proposed system (You can focus on a very minimum subset (MVP – Minimum Viable Product) for your demonstration, although your total design might be larger).

Note that for the implementation of Node-RED with connected services/sensors/actuators it is suggested to use Node-RED associated services that are already available through the Node-RED libraries and/or from the Node-RED connected IBM Cloud environment with connection to PC sensors like camera, microphone, speaker – or to the provided El-Watch Neuron wireless sensors that we have integrated into Node-RED (See description below). Others might also be considered for the MVP (Minimum Viable Product) demonstration – depending on the focus alternatives considered by the group.

Application and Technology layers with Structure/Behavior/Information aspects – and cross layer views – Application Usage view, Information view (UML Class Model) and Process view (BPMN)

Oblig 3 Group part: Provide a design model (in Node-RED) realisation for a minimum part of your product design of the "Smart Home" derived / specified from Oblig-1 with, together with further ArchiMate Application and Technology layer models/views and related UML/BPMN models, and any relevant updates to Oblig 1 for consistency. You are allowed to make changes/adaptations to your specification from Oblig 1 as you find suitable – and to update this accordingly (See information about this in 8) below).

Description for Oblig 3 – Group delivery

Overview

The focus of Oblig 3 (Group part) is to create relevant **models** for the design and implementation of the "Smart Home" product line from the "SmartX" startup company.

- 1) Create a Node-RED design model for your Smart Home product/service value proposition. Consider the usage of relevant IBM Cloud Watson IoT, TJBOT and Cognitive services with a minimum set of sensors and/or Watson services involved.
 1B) Consider the use of actual Node-RED Nodes and Flows available through the Node-RED library. These might include those that are connected with IBM Watson services, and also with Nodes for Image and Text/Language processing and Chatbots/Conversation – as well as the EI-Watch Neuron wireless sensors – described at the end of this document.
 1B) Create a working demonstration of your Node-RED design (a minimal subset/MVP is sufficient)
- 2) Create an ArchiMate Architectural layer model for your system for the supporting Node-RED system design – and Create an ArchiMate Application usage diagram showing the relationship (supports) between the relevant parts of the ArchiMate Business layer (updated from Oblig 2) and the ArchiMate Application layer.
- 3) Create an ArchiMate Technology layer model for your system design (typically well related to the Nodes, Functions and Flows in your Node-RED design, and associated sensors etc.) and Create an ArchiMate Technology usage diagram showing the relationship (supports) between the relevant parts of the Application layer and the Technology layer
- 4) Create a UML Class diagram detailing as you find it appropriate – the ArchiMate business objects in the Business layer, the data objects in the Application layer and the artifacts in the Technology layer.
- 5) Create a UML 2.x Composition/Composite structure diagram for the system architecture – with port/connectors/interfaces for the Smart Office system design – and comment on how this compares to parts of the ArchiMate Application and Technology views.
- 6) Create a BPMN diagram detailing as you find it appropriate – business process flow in the ArchiMate Business layer and/or application flow in the ArchiMate Application layer - and comment on how this compares to parts of the ArchiMate Application and Technology views.
- 7) Create any additional diagrams/models – that you find to be relevant and useful for your system architecture/design. Comment on why you find these models useful
- 8) Update Oblig 1 models/documents as you find necessary and submit a separate document Oblig-1-final – so that your revised Oblig 1 document becomes consistent with the new Oblig 3 models/specifications/decisions – i.e. including from Oblig 1: . Lean Startup Canvas for SmartX (in VDMBee) , Business Model Canvas for SmartX ArchiMate 3.0 Models (Using the Modelio tool) with Motivation, Strategy and Business Models (in Modelio), User stories and use cases model. Comment at the end what has changed between Oblig 1 and Oblig 3 – and any challenges related to how to handle the evolution of models/specifications over time.

RelevantLinks

Modelio with ArchiMate 3.0, UML and BPMN: <https://www.modeliosoft.com/en/products/modelio-ba-archimate-enterprise-architect.html>

Node-RED: <https://nodered.org/>

EIWatch Neuron sensors: <https://www.el-watch.com/en/>

IBM Watson IoT for connection of sensors and analytics,

<https://console.bluemix.net/catalog/services/internet-of-things-platform>

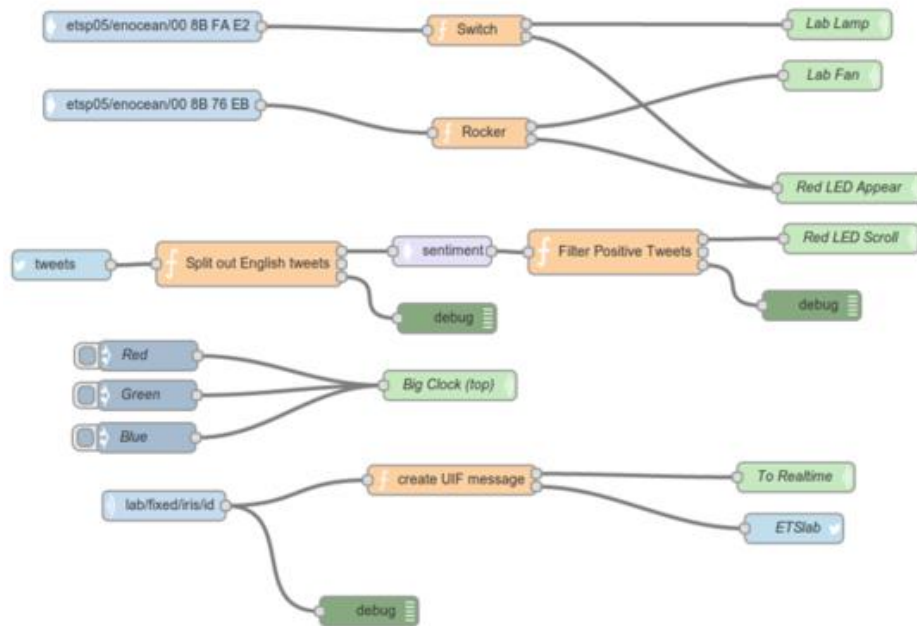
Consider also available sensors etc. from the Node-RED libraries and other sources – as suitable related to your initial Smart Home design - i.e. camera, microphone, speaker etc.

Submission

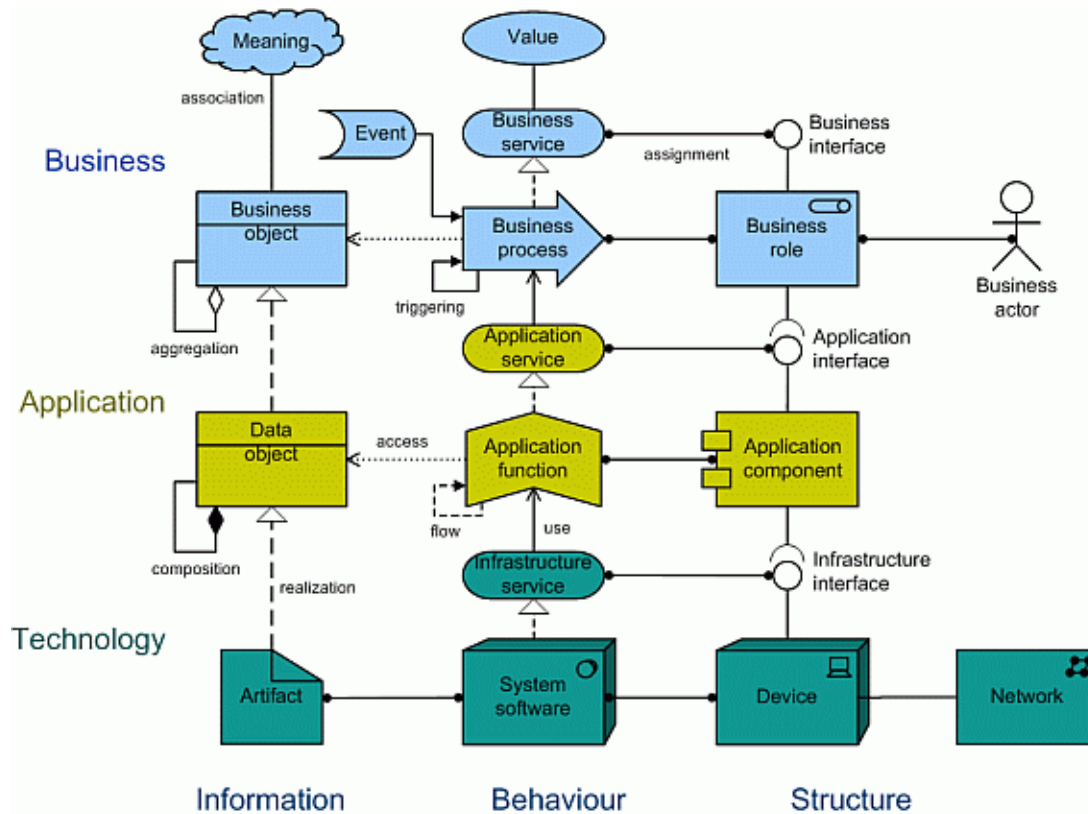
1. Submit a document with the models with descriptions, and also the model files.
2. Submit it through Devilry until May 13th at noon, (1159am)
3. Prepare a short Group demonstration/presentation (10 minutes) of your Oblig 3 submission for May 13th in the INF5120 lecture/exercise time slot between 1415 and 1700

Example models

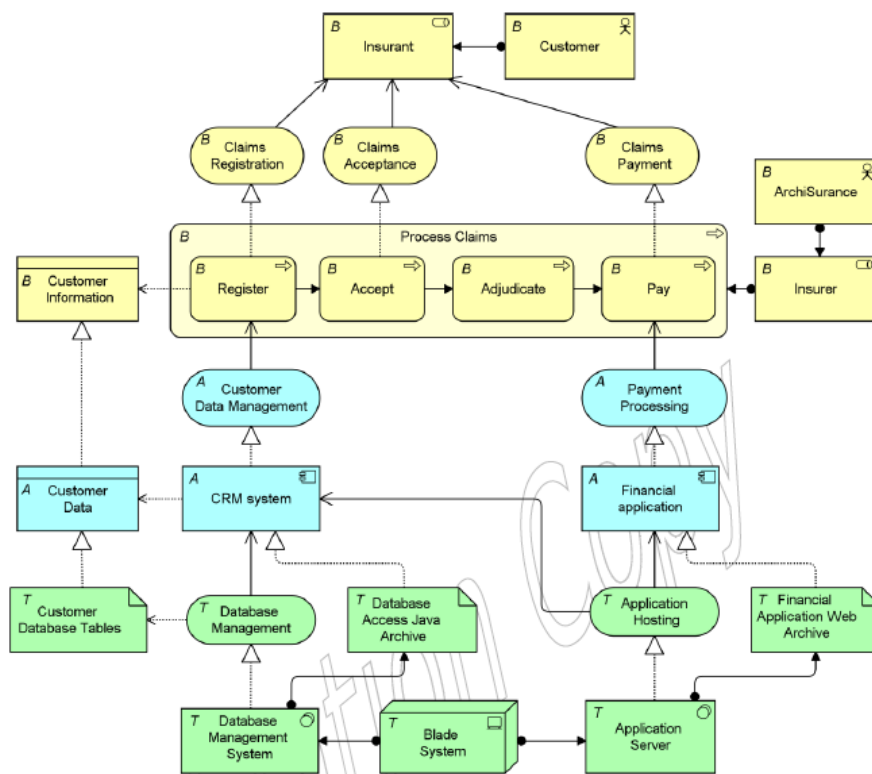
The following shows some examples of the kind of models that shall be provided as part of Oblig 3.



Example Node-RED diagram/model



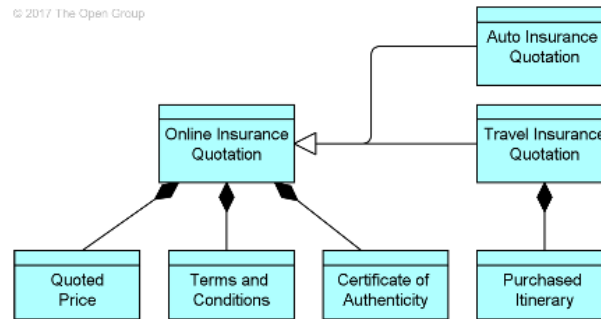
Example of ArchiMate models in the Business, Application and Technology layers



Example of usage views across ArchiMate layers

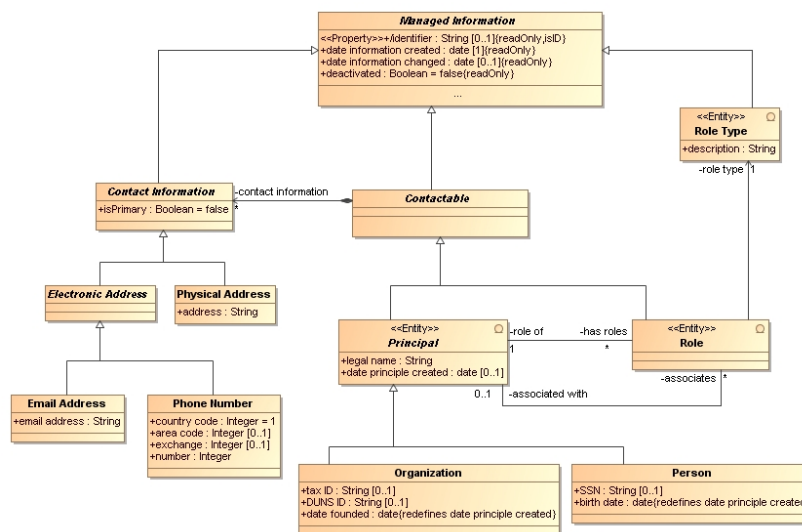
Example Application usage view between Business layer (yellow) and Application layer (blue) and Technology usage view between Application layer (blue) and Technology layer (green)

Business Objects in the Business layer are related to Data Objects in the Application layer and further to relevant Artifacts in the Technology layer.



Example of ArchiMate Data Objects

The Data Objects in particular, but potentially also Artefacts can be documented in more detail by UML class models as shown in the following.

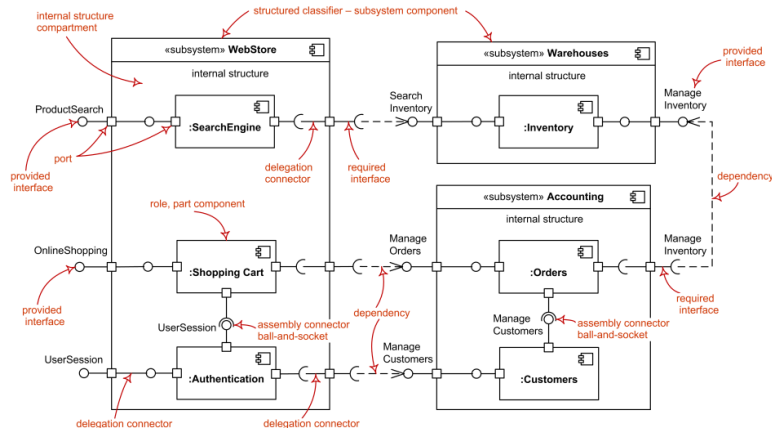


Example UML Class diagram

Information Models can be detailed with UML Class diagram – typical with less attributes for the Business Object in the Business layer, and with more attributes, data types and cardinalities etc. for the Data Object in the Application layer.

In a Node-RED design the Information structure is partially hidden in the input/output to and from nodes and functions. It is, however, there in the associated details and can be further documented through UML class diagrams.

UML component/composite diagram example (with ports, connectors and interfaces)



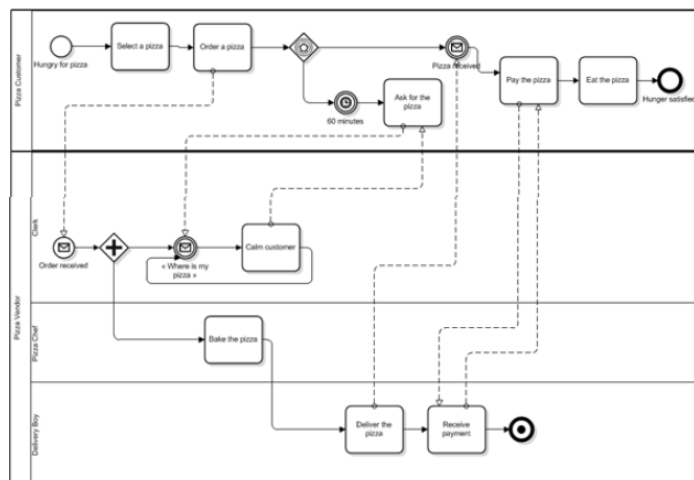
Example: UML Composite diagram

More details of UML diagrams here:

<https://modeling-languages.com/best-uml-cheatsheets-and-reference-guides/>

BPMN model (potentially used for process/flow models on both Business and Application layers)

<http://www.bpmnquickguide.com>



Example BPMN Diagram

Node-RED integration of El-Watch Neuron sensors

The following describes how to connect to wireless Neuron sensors from Node-RED.

1. El-watch API for Sensors

We have already implemented a Node-RED access to the Neruon sensor API available from here:
<https://www.el-watch.com/neuronsensors-data-api/>

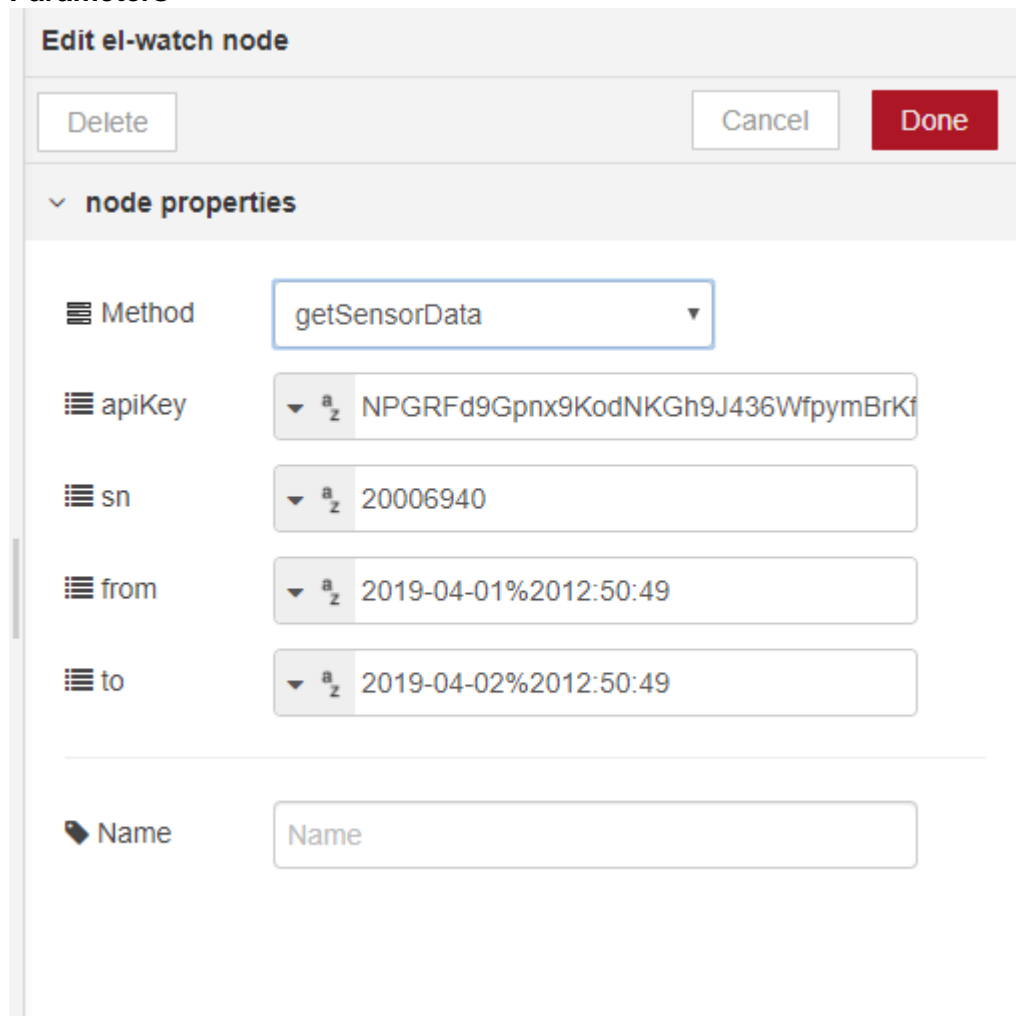
This can be accessed as described below:

2. Node-red custom node for el-watch api

node-red-contrib-el-watch

Use command “npm install node-red-contrib-el-watch” to install custom node package. Then restart Node-Red server then you will find el-watch node in nodes list on left side.

Parameters



The screenshot shows the 'Edit el-watch node' configuration window in Node-RED. At the top, there are three buttons: 'Delete', 'Cancel', and 'Done'. Below these is a section titled 'node properties' with a dropdown arrow. Inside this section, there are five configuration fields, each with a list icon on the left: 1. 'Method' is a dropdown menu currently set to 'getSensorData'. 2. 'apiKey' is a text field containing 'NPGRFd9Gpnx9KodNKGh9J436WfpymBrKf'. 3. 'sn' is a text field containing '20006940'. 4. 'from' is a text field containing '2019-04-01%2012:50:49'. 5. 'to' is a text field containing '2019-04-02%2012:50:49'. Below these fields is a 'Name' field with a tag icon on the left and the text 'Name' inside.

Move that node on canvas/design area and double click on el-watch node to see details, parameters.

Method: Choose method to get sensor data (Currently only one method available)

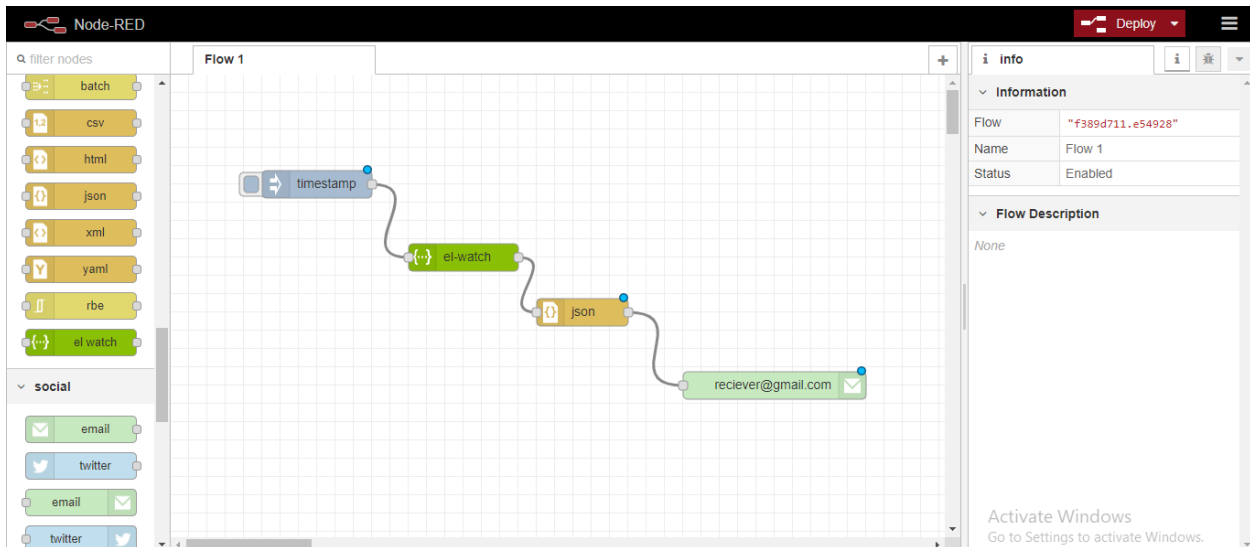
From: Enter start time from which point and further you want to get measured data

To: Enter time to retrieve data from sensors before that.

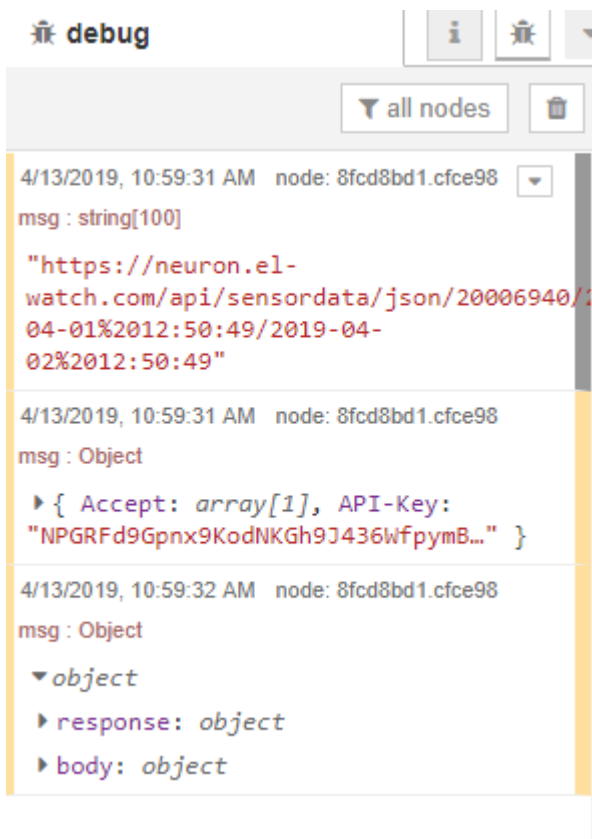
Sn: Enter serial number of sensor from which you want to get data. It can be found on el-watch web portal after login.

This will return measured data by a single sensor between from and to time.

3. Example of a flow to send email containing json data of sensor response.



4. You can use debug view to see raw output of el-watch API



5. You can further explore all different nodes of node-red and even install other node publicly available to create new functionality. For integrating other API's, HTTP node can be setup in Node-Red.

5. SINTEF has now created a test setup with some relevant Neuron sensors – that are available for access and use. We might also move these into a relevant test setup if needed. We should be able to share these between the groups.

Here is the login information to see these sensors:

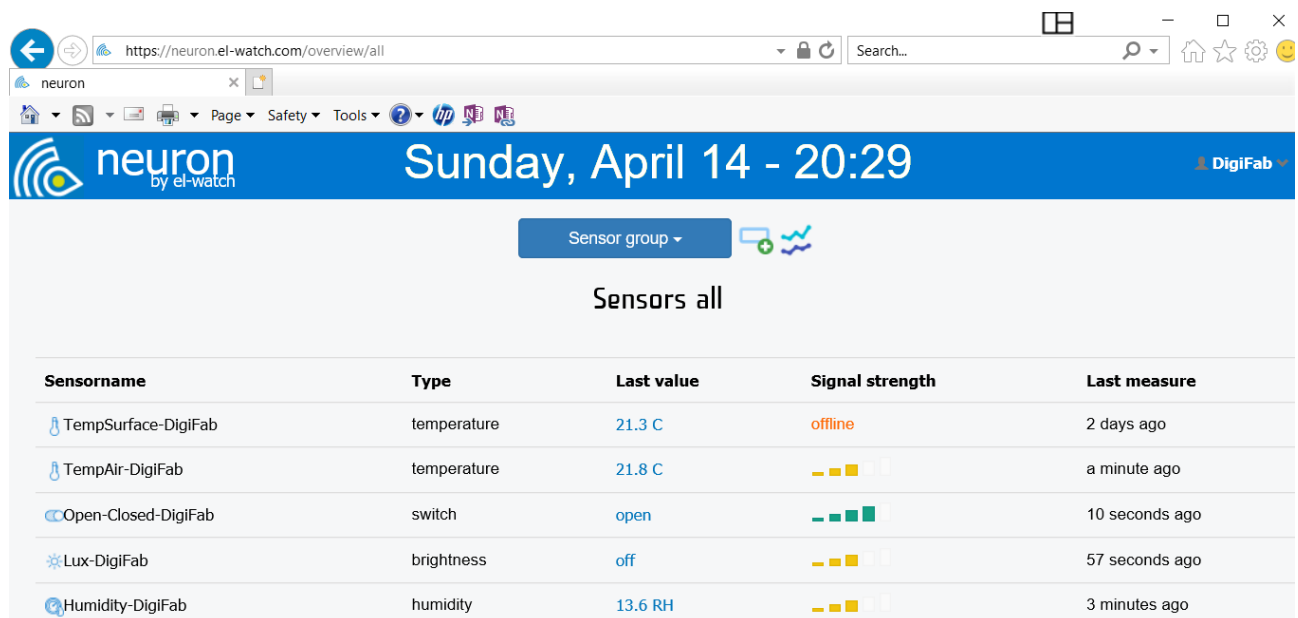
<https://neuron.el-watch.com/signin>

Username: DigiFab

Password: DigiFab2018

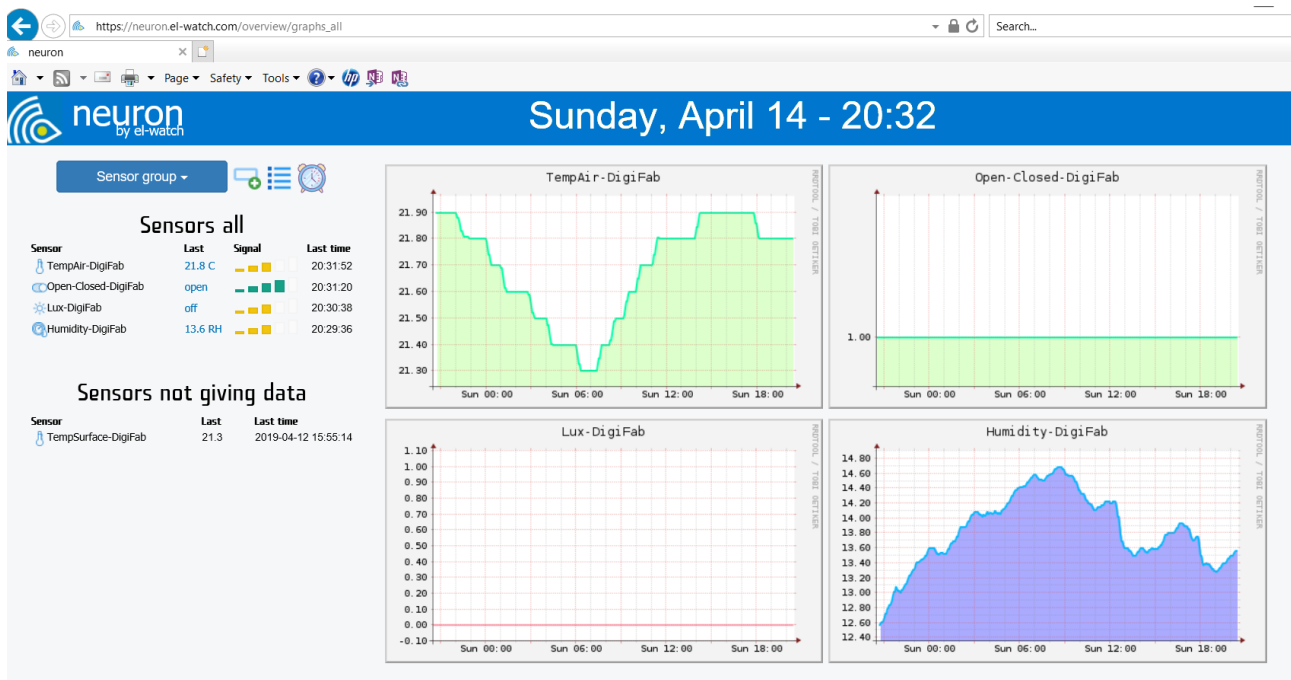
User guide: <https://www.el-watch.com/en/start-up-instructions/>

Access API: <https://www.el-watch.com/neuronsensors-leverer-data/>



The screenshot shows a web browser window with the URL <https://neuron.el-watch.com/overview/all>. The page has a blue header with the 'neuron by el-watch' logo, the date and time 'Sunday, April 14 - 20:29', and the user 'DigiFab'. Below the header, there is a 'Sensor group' dropdown menu and a 'Sensors all' title. A table lists five sensors with their names, types, last values, signal strengths, and last measurement times.

Sensorname	Type	Last value	Signal strength	Last measure
TempSurface-DigiFab	temperature	21.3 C	offline	2 days ago
TempAir-DigiFab	temperature	21.8 C	<div><div></div><div></div><div></div><div></div><div></div></div>	a minute ago
Open-Closed-DigiFab	switch	open	<div><div></div><div></div><div></div><div></div><div></div></div>	10 seconds ago
Lux-DigiFab	brightness	off	<div><div></div><div></div><div></div><div></div><div></div></div>	57 seconds ago
Humidity-DigiFab	humidity	13.6 RH	<div><div></div><div></div><div></div><div></div><div></div></div>	3 minutes ago



By selecting a sensor above one will find the required sensor id for the API as shown below (Edit sensor <sensor id>).

Edit sensor 20006414

Alias

TempAir-DigiFab

Type

temperature

Sensorgroup

Trendgraph

ON

OFF

Graph order

99

Valuerange

-40,100