on-line slides: http://behouddeparel.nl/MySense status#januari2022

MySense: Sensor Kits
How Did They Do It

- **Measurement Data Exchange Format (MDEF first implementation)**
  - **meta status information exchange**
  - **measurements data exchange**
  - **marshalling (data linearisation) in:**
    - **JSON**
    - **Python pickle ?**

teus hagen
email: mysense@behouddeparel.nl

vereniging Behoud de Parel
http://behouddeparel.nl

measurement data exchange format

dream two

MDEF
dream one
Koppelting 25th Jan 2020

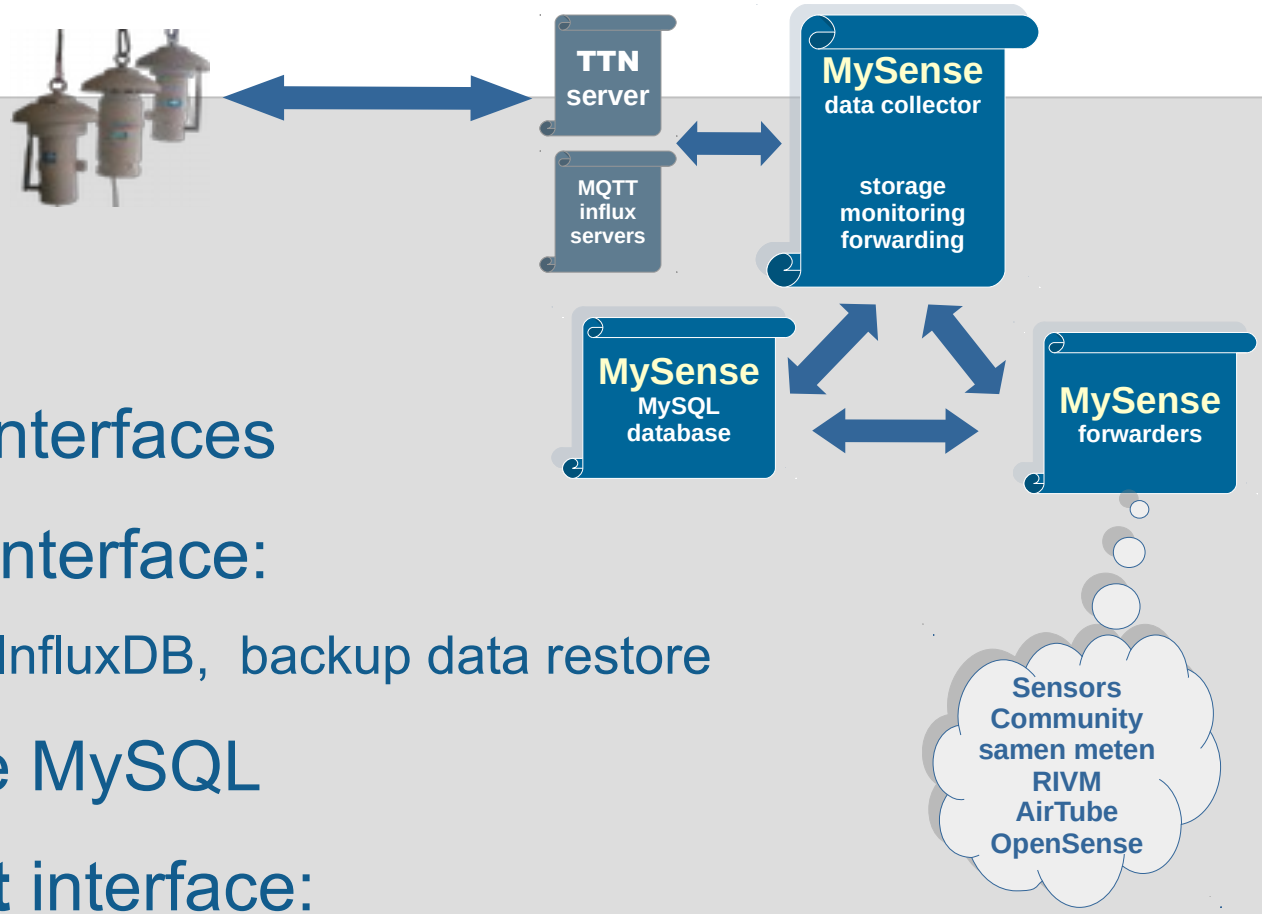
- **we all do data stream communication**:

  (s)http post, json, xml, python pickle (security issue!)
- **lessons learned**:

  timestamps, version, sensor type, data unit, meta data (defaults, etc.), flexible, use of state?, human readable, sample implementations, naming, independent, version, scalable, more?
- **MySense data collector**

  data acquisition, data check, monitoring and forwarder

Python data format how it is done



- **internal** modules interfaces
- data record **input** interface:

  Mosquitto (MQTT), InfluxDB, backup data restore
- **database** interface MySQL
- data record **output** interface:

  HTTP(s), Mosquitto, InfluxDB, monitor, debugging

JSON example (uses state, defaults)

```
{
    'meta': [ { 'id': '123456ABCDE',
                'label': 'BdP-1234',
                'GPS': [ 52.123456, 6.123456], // home static
                'match': { r'bme([26]80)': 'BME\1', r'temp.*': 'temperature'},
                'sensors': [ {'BME280': {'temp': 'C', 'RH': '%' }, } ]
              } ],
    'data': [ {'date': 123456789,
                'BME280': {'temperature': 21.3, 'pressure': 1023 },
                'SPS30': { 'date': 123456790, 'PM2.5 #': 23.4 },
                'GPS': [ 52.123454, 6.123454] // dynamic
              }]
}
```

meta info configuration

names of ‘pollutants’

- **"translate"**: {

"*pm25*": {"pm25","pm2.5","PM2.5"},

"*pm10*": {"pm10","pm","PM"},

"*O3*": {"O3","ozon"},

"*temp*": {"temp","temperature"},

"ws": {"ws","windspeed","windsnelheid"},

"*wr*": {"wr","windrichting","winddirection","direction"},

"*geohash*": {"geohash",},

"*altitude*": {"altitude","alt","hoogte","height"},

"*longitude*": {"longitude","long","lon","lengte graad"},

. . .

meta info information

default info sensor types

MySQL database table ‘SensorTypes’

an example:

timestamp: 2021-11-07 15:25:39 CET

product: SDS011

matching: (SDS|sds)011 regular expression

producer: Nova

category: dust

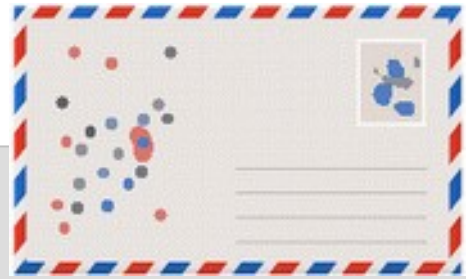
fields: list of name, unit, calibration function (type [Taylor sequence]) . . .

pm25, ug/m3, PMSx003 [1.6190,1.5450] SPS30 [2.1630,0.7645] BAM1020 [5.7590,0.3769]

pm10, ug/m3, PMSx003 [3.7600,1.1570] SPS30 [1.6890,0.6322] BAM1020 [1.4370,0.4130]

Measurements Data Exchange Format

record envelope example



```
{ // MDEF data record envelope

 // if timestamp, version, or id is not defined in data record
 // the key of child in the tree is taken.
 // timestamp default: timestamp of receive data

   "version":     0.02,          // version of exchange format Nov 2021
   “id”:          { “project”: “SAN”, “serial”: “78CECEA5167524” },
   "timestamp": 1621862416,   // or “2021-05-24T15:20+02:00”,

   // meta data is state information of a measurement kit
   "meta": …

   “data”: …

   “net”: ...
}
```

Measurements Data Exchange Format

meta record example

```
// optional, meta data is state information of a measurement kit, default undefined
"meta": {              // meta data (re)definitions, kit state sensor type in use definitions
    "version":     0.2,                  // firmware version, optional
    "timestamp":   1621862400,           // meta info timestamp, optional

    "dust":        "PMSx003",            // dust sensor type
    "meteo":       [ "BME680", "SHT31" ],          // more as one type present in kit
    "energy":      { "solar": "5W", "accu": "Li-Ion" }, // energy type: dflt "adaptor"
    "gps":         "NEO-6",                        // sensor type

    "geolocation": { "geohash": "u1hjjnwhfn", "alt": 18.2  },   // static location
    "GeoGuess":    True,          // optional if geolocation geohash is gateway location

    "event": 13                                    // measurement event
},
```

Measurements Data Exchange Format

‘net’ record example

```
"net": {
    "timestamp":    1621862950,  // or “2021-05-24T13:29:10+00:00”
    “TTN_id”:    “kipster-k1”,
    “TTN_app”: “201802215971”,
    “type”:        “TTNV2”,
    “gateways”: [
     { “gateway_id”: “eui-ae01c16”, “rssi”: -94,  “snr”: 9.5, “geohash”: None }
     ]
},
```

Measurements Data Exchange Format

‘data’ record example

```
"data": {                          // measurements, only those active at that moment
    "version": 0.2,                // data version, optional
    "timestamp": 1621862400,       // measurement timestamp, optional
    // internal use: 'sensor type':   [ (“field name”, value [, unit[, calibration]] ), ...]
    "NEO-6":      { "geohash": "u1hjjnwhfn", "alt": (23.2, None , [-1, 1, 0.01] ) },
    "BME680":     { },             // present but undefined or invalid
    “SHT31”:      [ { “temp”: 20.1, “rv”: 70.1 }, { “temp”: 20.3, “rv”: None } ],
    "PMSx003": {                   // _cnt items are PM count up to upper bound bin!!
      "pm05_cnt": 1694.1, "pm10": 29.4, "pm25_cnt": 2396.9, "pm03_cnt": None,
      "grain": (0.5, ”mu” ),  // average grain size
      "pm1_cnt": 2285.7, "pm25": 20.4, "pm10_cnt": 2.4, "pm1": 13.0 },
    “accu”: ( 89.5,”%” )
  }
```