SenML - Too **Little** Meta Data

Consider

```
    {"e":[{ "n": "urn:dev:ow:10e2073a01080063", "v":23.5, "u":"Cel" }]}
```

We know the globally unique name of the sensor and it is a temperature of 23.5 degrees Celsius

We don’t know the device type, OS version, manufacturer, which data model it uses or the access control lists. Yes, you might need that for some management but you don’t need it for most use cases for a sensor

Goal is to cut it down to the bare minimum that still accomplishes many (but not all) use cases and is really simple to understand and use

The web, like IP, is successful because it started simple and easy

SenML - Too **Much** Meta Data

One might say:

> The name of the sensor and unit is duplicated meta data so why send it in every measurement? Get rid of the meta data and just send 23.5

Keeping the name and time allows the data to be stored in a schemaless DB and still processed. It allows many cache, aggregation, and filters to be applied.

It can increase performance for servers receiving millions of measurements to be able to handle the measurement in a stateless way

SenML tries to balance the meta data to make it easy for small simple devices with limited connectivity while being easy for large servers using current big data style tools