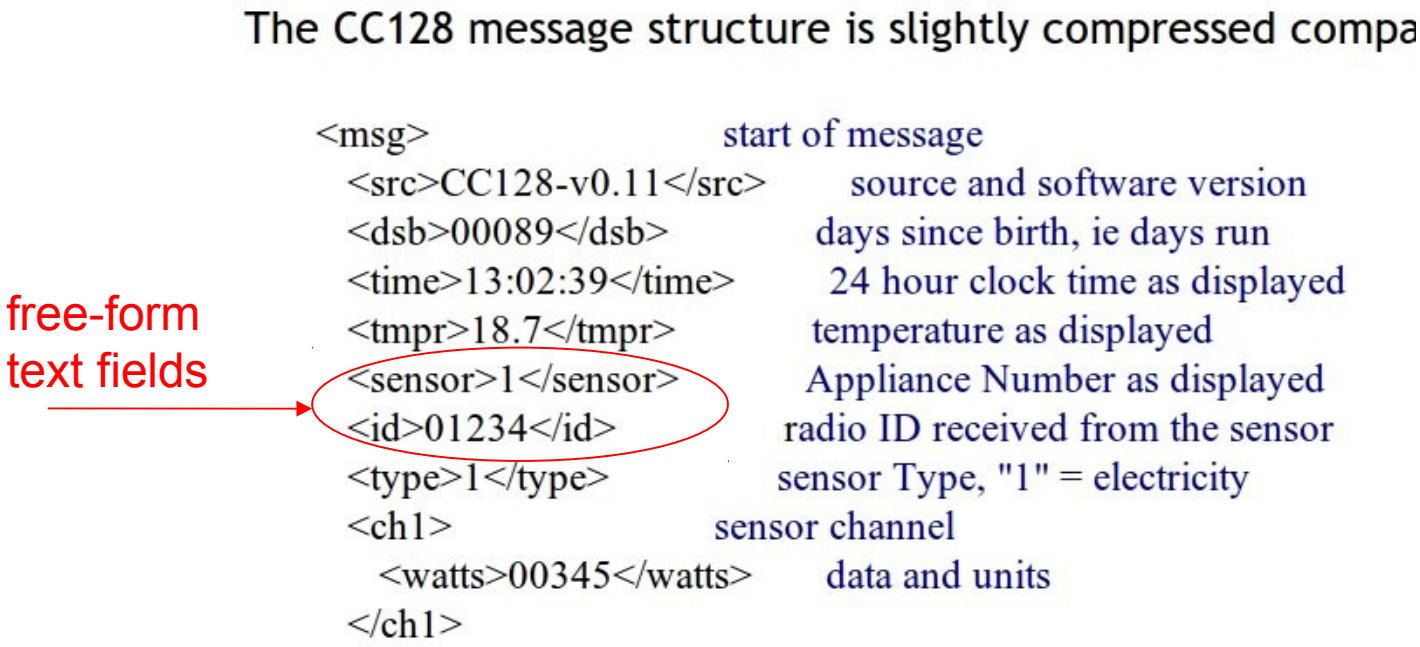
ERICSSON

Device URN

Jari Arkko, Cullen Jennings, Zach Shelby

What's the Problem?

1. Google for ”XML sensor data format”
2. Take the first search result
3. Go to the first example on the page

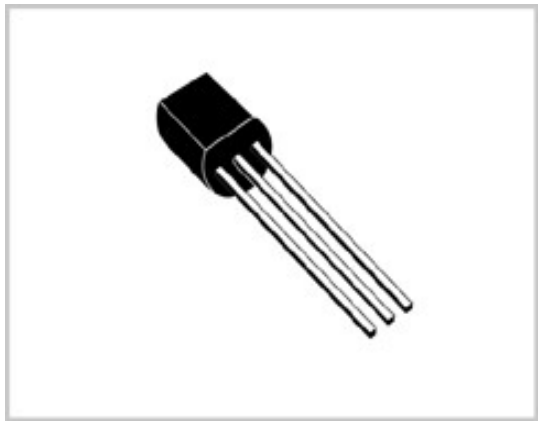


Text-Based vs. Uniform Identifiers

- Cannot make any use of the text identifiers beyond exact match
- Text identifiers do not have clearly defined scope or uniqueness properties
- Uniform, formally defined identifiers can be passed around more easily:
  - They are self-describing
  - Merging data from different sources easier
  - No coordination needed across types
- Conclusion 1: use URNs or URIs as identifiers

Identifier Types

- Semantics-based ”sensor for the oven”
- Name-based ”my_sensor_3”
- Location-based ”coordinates X,Y”
- Address-based ”http://[2001:db8::1]”
- Device ID-based (”mac=...” or ”serial=...”)



How do you configure this device to send a name or location?

Conclusion 2: Device IDs are attractive for many deployment cases – e.g., identifying specific devices in sensor data streams, storage servers and equipment inventory applications. Names are obviously needed too, but can exist at higher layers

The Specification for ”dev” URNs

urn:dev:mac:0024befffe804ff1 (my laptop's MAC address)

- Device identifiers based on EUI-48/64 MAC addresses
  - Similar to UUIDs but requires no real-time clocks, stable storage, and has easier process on the manufacturing side
- Device identifiers based on 64-bit 1-Wire addresses
- Device identifiers based on cryptographic identifiers – related to the security discussion from yesterday
- Extension rules for new types

ERICSSON

SenML

draft-jennings-senml

Cullen Jennings

Zach Shelby

Jari Arkko

```
{"e":[
      {"v":"23.5", "t": "0"},
      {"v":"23.4", "t": "10"},
      {"v":"23.4", "t": "20"},
      {"v":"23.3", "t": "30"},
      {"v":"23.2", "t": "40"},
      {"v":"23.0", "t": "50"},
      {"v":"22.0", "t": "60"},
      {"v":"19.3", "t": "70"},
      {"v":"17.21", "t": "80"},
      {"v":"17.03", "t": "90"},
      {"v":"16.9", "t": "100"}
    ],
 "bt":"1276020076",
 "bn":"urn:dev:ow:10e2073a01080063"
}
```

Why?

- Smart objects need common data format(s)
- JSON is an easy, relatively compact format
- Properly designed base format helps use a generic data container for typical smart object applications – no need to design a scheme just to represent temperature measurements
- Right design helps keep size down even on textual format
- JSON, XML, EXI mappings

```
root@weather:/home/jar/OneWire/History# ls -l 26.*
-rw-r--r-- 1 root root 50436604 2011-11-17 21:44 26.2B4DF5000000.history.dat
-rw-r--r-- 1 root root 75752642 2011-11-17 21:44 26.2D5FE7000000.history.dat
-rw-r--r-- 1 root root 50438850 2011-11-17 21:44 26.4437F5000000.history.dat
-rw-r--r-- 1 root root 95495758 2011-11-17 21:44 26.80A3CD000000.history.dat
root@weather:/home/jar/OneWire/History#
```