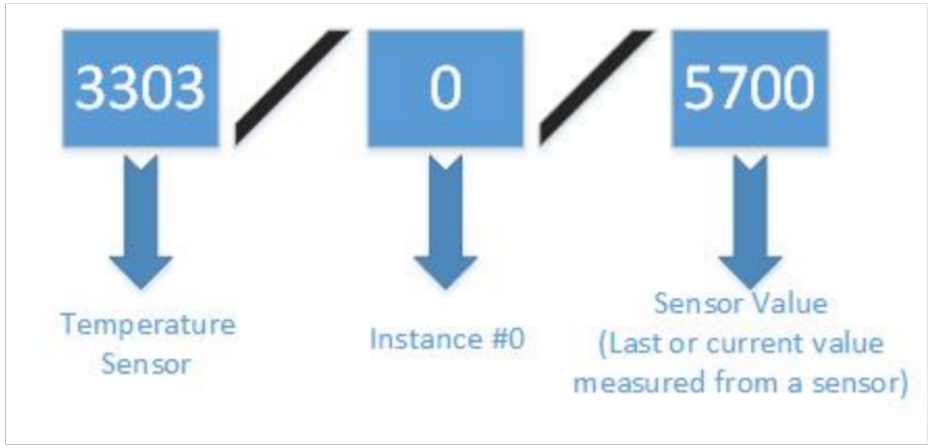
# Data model component re-usability: Re-usable Resources

WISHI 5.2.2019

Ari Keränen

# IPSO information and data model

- Set of resources that are grouped into objects
  - Devices/things modelled as a set of objects
  - Numeric IDs for objects and resources
  - Instance IDs for having multiple instances of objects and resources




- Composite object option for grouping objects

# Resources

- The atomic values (string, float, Boolean, time, etc.)
- Defined name, data type, operation, description
- Optionally also defined range, unit, is this mandatory/multi-instance
- Can be object specific or reusable
  - different numeric ranges for each
  - Re-usable resource examples: current value, min value, max value,…
- Re-usable resource characteristics are fixed after registration
  - E.g., if range is defined, it can’t be re-defined in new object

# Why Reusable Resources (RRs)?

- Use automatic logic in back-end/UI by understanding the semantics
  - "using current, min, and max values, and reset resource, set up an UI widget for showing/adjusting/resetting values, independent of what the object/value is actually about”
  - Data and input validation
- Possibility to add RRs in any object and therefore extending the object capabilities without defining a new object (version)?
  - RR semantics would need to make sense in the context (“avg. speed” vs. “description”)
  - Making up new objects by using only RRs?
  - Requires discovery; which could be the only way
- Re-use for specification: don't need to define details every time you're using RR in a new object
- What else?

# Evolvability strategy

- Best/only way: discovery. Multiple levels:
  - Simplest way: one ID that tells what remote system doing (GitHub API etc.)
  - Version number (enables simple evolution)
  - Put more details from the spec to the wire
    - Resources
    - Operations
    - Parameters
- Cost: implementations have to deal with variety of systems
  - Hypermedia is (high) potential solution
  - Bigger representations
    - Splitting to parts and efficient encoding help

K. Hartke: ”Hypermedia for Long-Term Semantic Interoperability”