

Eclipse Unide

eclipsecon Europe 2016, Unconference (24.10.2016)



Production Performance Management Protocol Overview

Through central collection of machine's vital signs, plant staff effectively monitors the production condition

Sensors broadcast measurements and alert messages

PPMP

Production Performance Management Protocol

Payload definition and sample implementations facilitates the adaptation of further IoT technologies

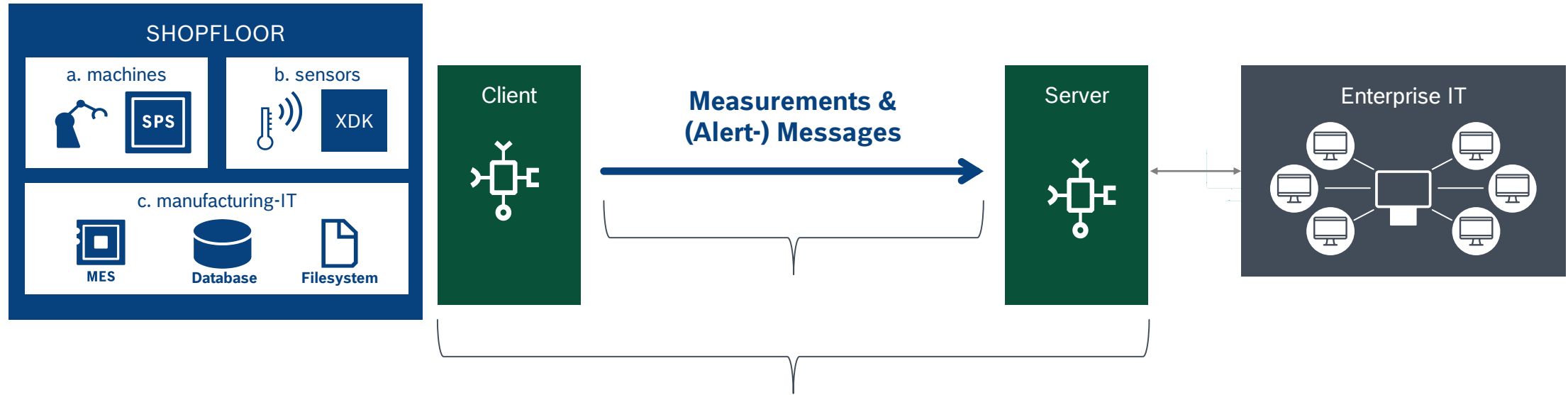
Temperature

Vibration

Energy consumption

Abrasion

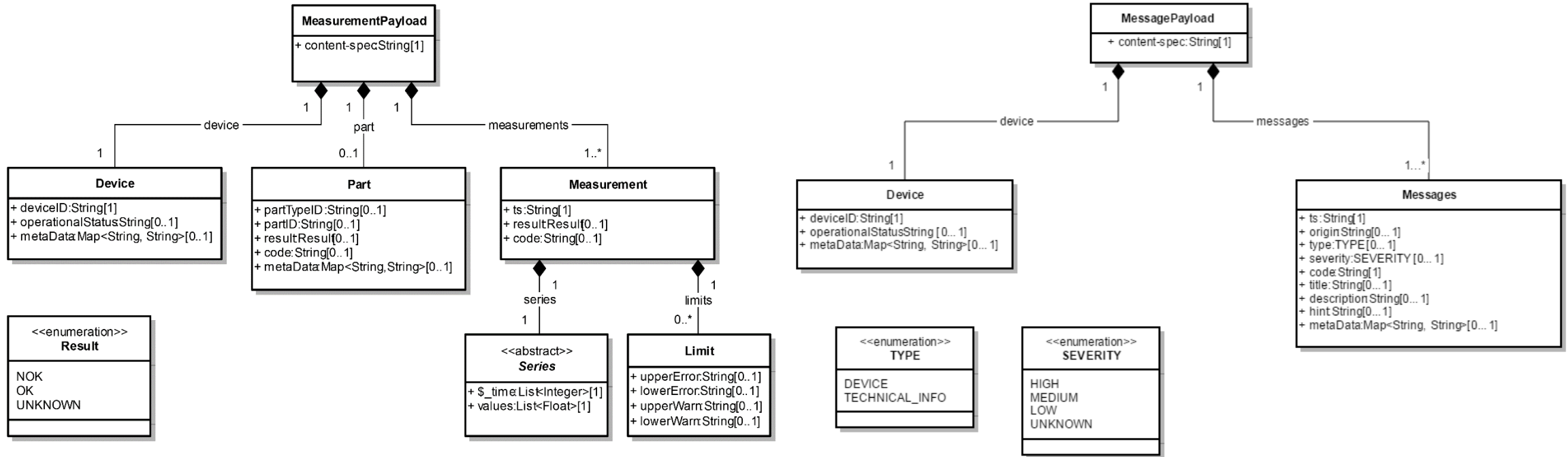
Eclipse Unide / PPMP Project Scope



- Definition of Payload structure
- Sample Implementations
 - Client
 - Server
- Reusing existing Protocols

Eclipse Unide / PPMP

Machine Measurement / Messages



```

{
  "content-spec": "urn:spec://eclipse.org/unide/machine-message#v2",
  "device": {
    "deviceID": "2ca5158b-8350-4592-bff9-755194497d4e",
    "operationalStatus": "normal",
    "metaData": {
      "swVersion": "2.0.3.13",
      "swBuildID": "41535"
    }
  },
  "messages": [
    {
      "origin": "sensor-id-992.2393.22",
      "ts": "2002-05-30T09:30:10.123+02:00",
      "type": "DEVICE",
      "severity": "HIGH",
      "code": "190ABT",
      "title": "control board damaged",
      "description": "Electronic control board or its electrical connections are damaged",
      "hint": "Check the control board",
      "metaData": {
        "firmware": "20130304_22.020"
      }
    },
    {
      "ts": "2002-05-30T09:30:10.125+02:00",
      "type": "TECHNICAL_INFO",
      "severity": "HIGH",
      "code": "33-02",
      "title": "Disk size limit reached",
      "description": "Disk size has reached limit. Unable to write log files."
    }
  ]
}

```

Eclipse Unide

Differentiator to other protocols

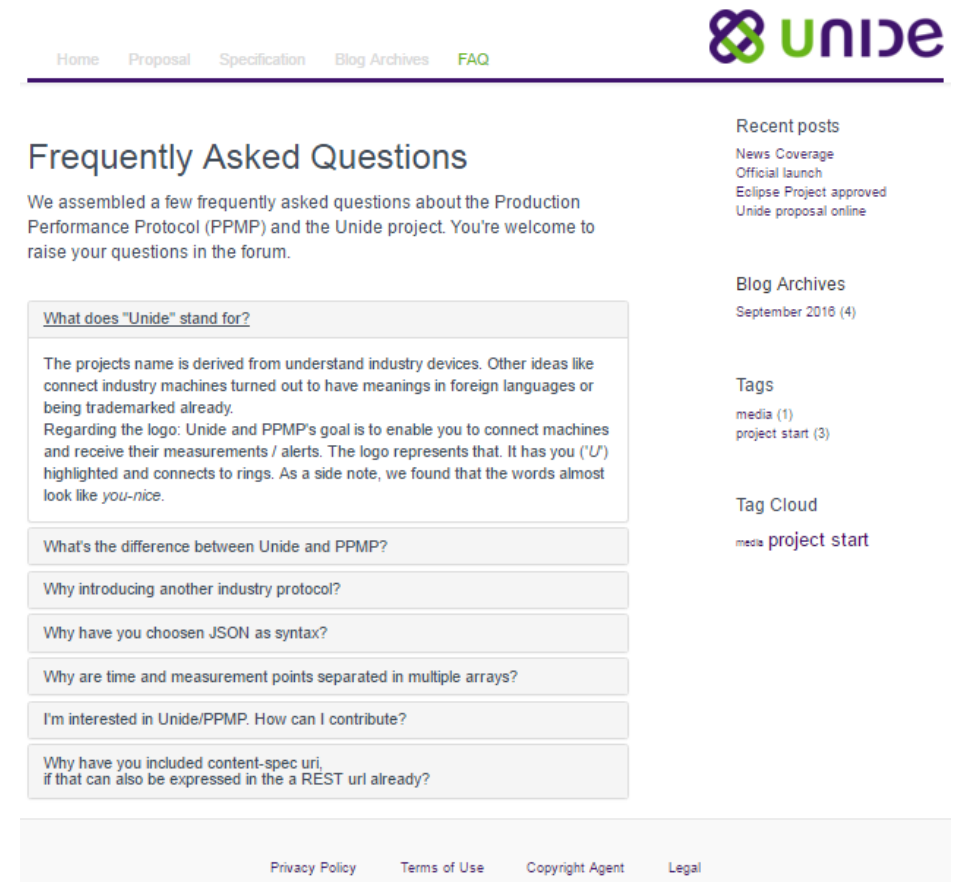
- Sample implementations for Connected Industry
- Small Payload structure
- Industry focus (part info., operational status)
- Reusing existing / small Hardware
- Reusing existing IoT technology



Eclipse Unide

Initial Progress (08/16-now)

- Website with payload definition launched
- Advertising started
 - Press releases backing PPMP/Unide
 - Blog posts
- Motivated interested (industry) parties
SAP, Tata Consultancy Services, Dassault Systèmes, Balluff, CADIS Service GmbH, EGT AG, KLW Karl Lutz GmbH, RAMPF Machine Systems GmbH & Co. KG, J. Schmalz GmbH
 - Expectations: simple, flexible, easy to use
- Seeking collaboration with other projects

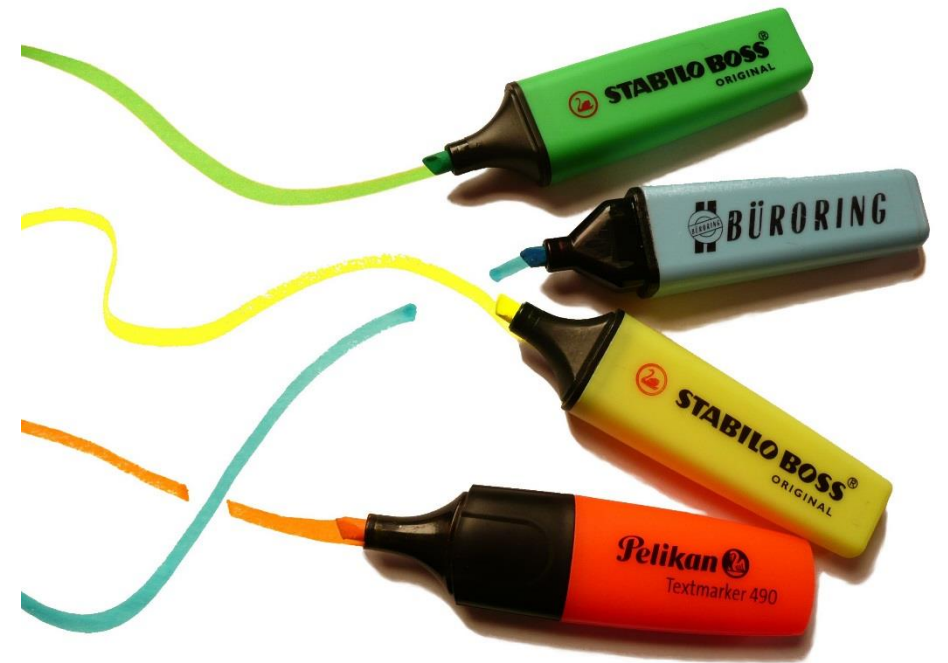


The screenshot shows the Eclipse Unide website's Frequently Asked Questions (FAQ) page. The page has a navigation bar at the top with links for Home, Proposal, Specification, Blog Archives, and FAQ. The Unide logo is in the top right corner. The main content area is titled "Frequently Asked Questions" and includes an introductory paragraph: "We assembled a few frequently asked questions about the Production Performance Protocol (PPMP) and the Unide project. You're welcome to raise your questions in the forum." Below this is a list of FAQ items, each in a light gray box with a question in italics. The first item is "What does 'Unide' stand for?" with a detailed answer explaining the name's origin and the logo's meaning. Other questions include "What's the difference between Unide and PPMP?", "Why introducing another industry protocol?", "Why have you chosen JSON as syntax?", "Why are time and measurement points separated in multiple arrays?", "I'm interested in Unide/PPMP. How can I contribute?", and "Why have you included content-spec uri, if that can also be expressed in the a REST url already?". On the right side, there are sections for "Recent posts" (listing News Coverage, Official launch, Eclipse Project approved, and Unide proposal online), "Blog Archives" (September 2016 (4)), "Tags" (media (1), project start (3)), and "Tag Cloud" (media, project start). At the bottom, there is a footer with links for Privacy Policy, Terms of Use, Copyright Agent, and Legal.

Eclipse Unide

Next Steps

- Driven by Manufacturing
- Client / Server Implementations (2016)
 - (Chrome-) App, Raspberry Pi Clients
 - (REST-) Node.js / Python / Camel Server
 - Reusable bindings (java general, Hono)
- Evolving payload definition (2017)



Picture: photosteve101 (CC BY 2.0)