Smart Manufacturing
Tech Overview

January 2020
CESMII SM Platform™ “Markitecture”

Smart Manufacturing API - Streaming & Historical Data via Model
Data Store
Data Contextualization - Federate all Profiles & Assemble Data Model
Data Management - Connect, Monitor, Manage

Connect, Ingest, Buffer... (OSI, OPC, CIP, MQTT...)
CESMII SM Platform™ Core Capabilities

- API
- Apps
- Profiles
- CESMII Marketplace
- Orchestration Workflow Capability
- Contextualization Capability
- Ingestion & Connectivity
- Provisioning Of SM Platform
- Hyper scale Cloud Provider
What is the CESMII SM Profile?

- SM Profiles will **capture the key attributes** of a device, asset, flow-path etc in a structured contextualized data model.

- These profiles will be **collaboratively developed** by subject matter experts from CESMII membership.

- The SM Profile will **enable** components and applications in the SM platform to **interoperate** and allow **information to flow freely** between them without losing context.

- The SM Platform, SM Profiles and SM Apps will all be **provisioned from the SM Marketplace** for use by CESMII’s members.
SM Profiles are an extension to the OPC/UA Information Model that can be distributed to automate the creation of information value throughout a software architecture.

**SM Profiles™**
**TYING IT ALL TOGETHER**

- **“Edge”**
  - Gathers raw data
  - Determines what data to collect and how
  - Defines the object model and interface

- **Data and Object Model**
  - Contextualizes, relates and stores data in a model
  - Uses the interface for type-awareness. Provides Profile extensions to extend the data.

- **App**
  - Uses and modifies data to create value
  - Profile

- **Manufacturing System**
  - Profile
Profile Design Personas

**Personas**

*Machine or Process Designer:* this persona has significant knowledge about a specific machine or process, and the general class to which that entity belongs. In the case of a machine, this persona might be the machine builder or designer, deeply familiar with the particular machine and other similar types of machines. In the case of a manufacturing process, this persona will be well versed in the process, its behaviors, physics and commonly used data points.

*Machine or Process Implementer/Integrator:* this persona has knowledge about how a machine or process is implemented, including specific communication technologies for a given instance. They would likely be involved in the programming of the machine or process, specifically applying their knowledge to cause the machine or process to be useful in a given situation, or to integrate with other manufacturing systems.

*Application Developer or Data Scientist:* this persona wants to work with the data produced by a machine or process – either extending that data or leveraging it to solve a problem – possibly both. This persona may not have instance-specific machine or process knowledge, rather domain knowledge around an application, algorithm or research effort that create new value on top of the existing Profile.
## COMPARISONS WITH COMPLEMENTARY EFFORTS

<table>
<thead>
<tr>
<th>OPC UA</th>
<th>MT Connect</th>
<th>CESMII Profile Effort</th>
</tr>
</thead>
<tbody>
<tr>
<td>An XML based “language”</td>
<td>A specific XML based “interface” <em>in the OPC UA language</em></td>
<td>(Probably) XML based class definitions and interfaces <em>in the OPC UA language</em></td>
</tr>
<tr>
<td>General semantic for <strong>information modeling</strong> (can be used for anything)</td>
<td>Specific semantic for <strong>machine modeling</strong> (pre-defined models) initially focused on CNCs</td>
<td>Multi-part semantic for distribution, protocol binding and extensible object creation</td>
</tr>
<tr>
<td>Type-safe, infinitely extensible at <strong>design time</strong></td>
<td>Type-safe, monolithic as defined at the <strong>time of specification</strong></td>
<td>Type-safe, specific extensions and polymorphism <strong>at run-time</strong></td>
</tr>
<tr>
<td>Primarily focused on making data flow generally upward (although can be used horizontally and bi-directionally) where the implementer selects the data set.</td>
<td>Primarily focused on machine-to-machine status and control (although can be used vertically) where the specification defines the data set.</td>
<td>Primarily focused on shifting the responsibility for selection of data and automating acquisition (take the burden off the customer)</td>
</tr>
<tr>
<td>Specification includes the behavior of the client and server (and publisher and subscriber). <strong>Type discovery implicit.</strong> Implementation and protocol translation are up to the vendor.</td>
<td>Specification defines an Agent that abstracts the machine to another layer that exposes data (and commands) on behalf of the machine. <strong>Type discovery via pre-definition.</strong> Translation up to the vendor.</td>
<td>Implementation will require member contributions to support protocol adaption. Reference implementations will be made available, but open for other vendor implementations. Type discovery through Profile distribution.</td>
</tr>
</tbody>
</table>