IoT for Industrial Controls

Design Exploration, SDK, and Standards

Michael J Koster

November 26, 2020

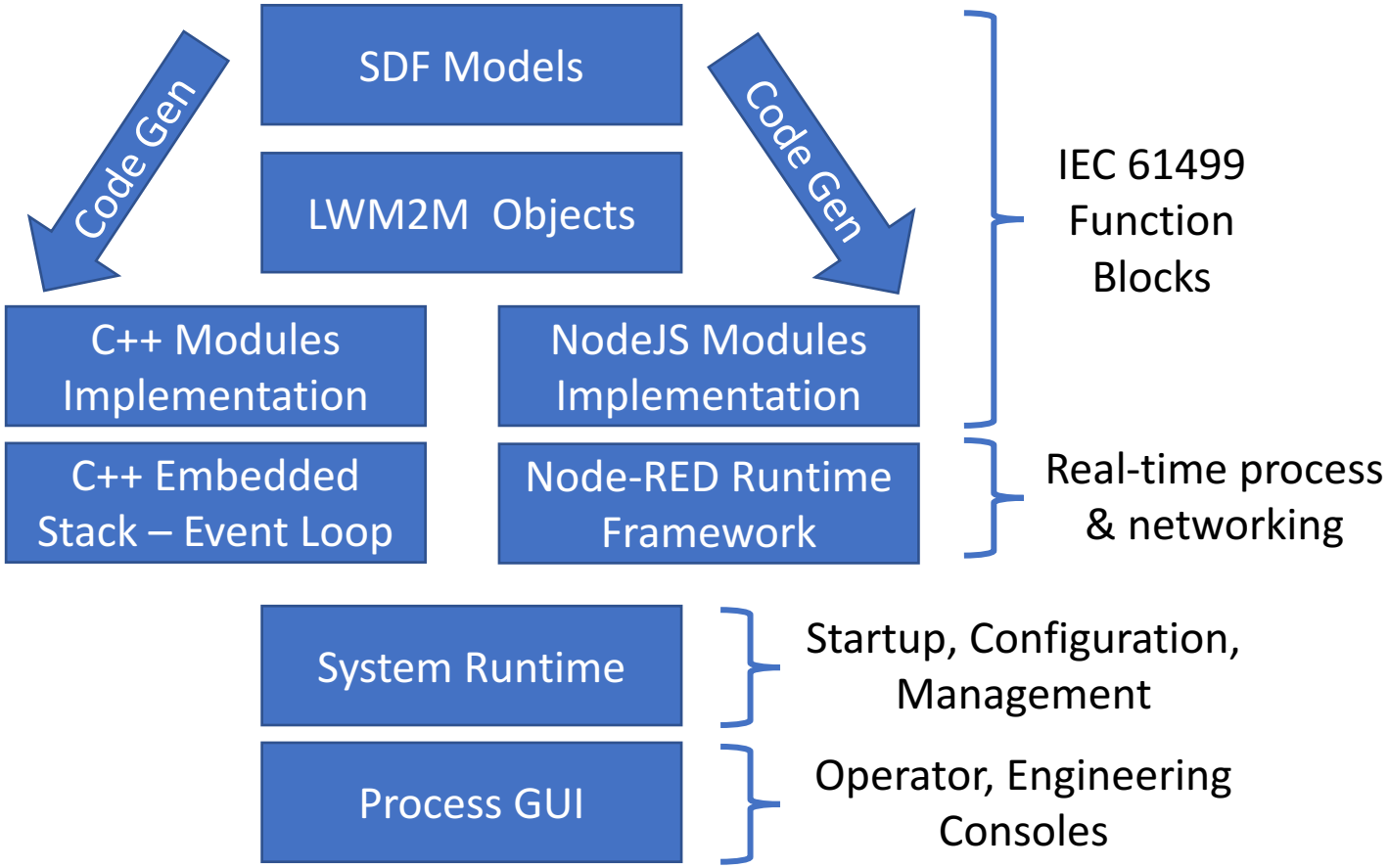
Scope

- Apply IoT industry learning to Industrial Controls
- Abstraction of Communication Protocols using IoT patterns and technology - OneFieldBus
- Model-driven design using SDF with existing standards
- Map PLC programming patterns onto a distributed control fabric
- SDK Approach to deliver a reference platform
- Open Standards and Open Source Runtime

End-to-end Model Driven Design

- Use SDF and SDF extensions to model system elements – architecture neutral and developer friendly
- Map to RDF and connect to existing ontologies
- Process Model – P&ID data, Digital Twin Modeling
- Instrumentation and Control Points
- Control networks – Control Loop Diagrams
- Function blocks – IEC 61499
- Application logic – Inside Function Blocks

SDK Outline


SDF Models
LWM2M Objects
Code Gen
Code Gen
C++ Modules Implementation
NodeJS Modules Implementation
C++ Embedded Stack – Event Loop
Node-RED Runtime Framework
IEC 61499 Function Blocks
Real-time process & networking
System Runtime
Process GUI
Startup, Configuration, Management
Operator, Engineering Consoles

Modeling in SDF and Extensions

- Function Block – Properties, Actions, Events
- Internal Logic– Application Logic, State Machines
- Common Data Types; Quantities, Units, Scales
- Alarms and Reports, Logging
- Fieldbus Message formats and Protocols
- System Logic – Brokers, Servers, Descriptors
- Process Modeling – P&ID and Digital Twin Models
- Control Points and Control Networks

Standardization Opportunities

- Function Blocks for common applications, PID controllers, sensors and actuators, sequencer controllers, programmed logic
- SDF language extensions for control systems + RDF mapping and integration with common ontologies
- Mapping LWM2M Objects to PubSub protocol
- Models for system elements

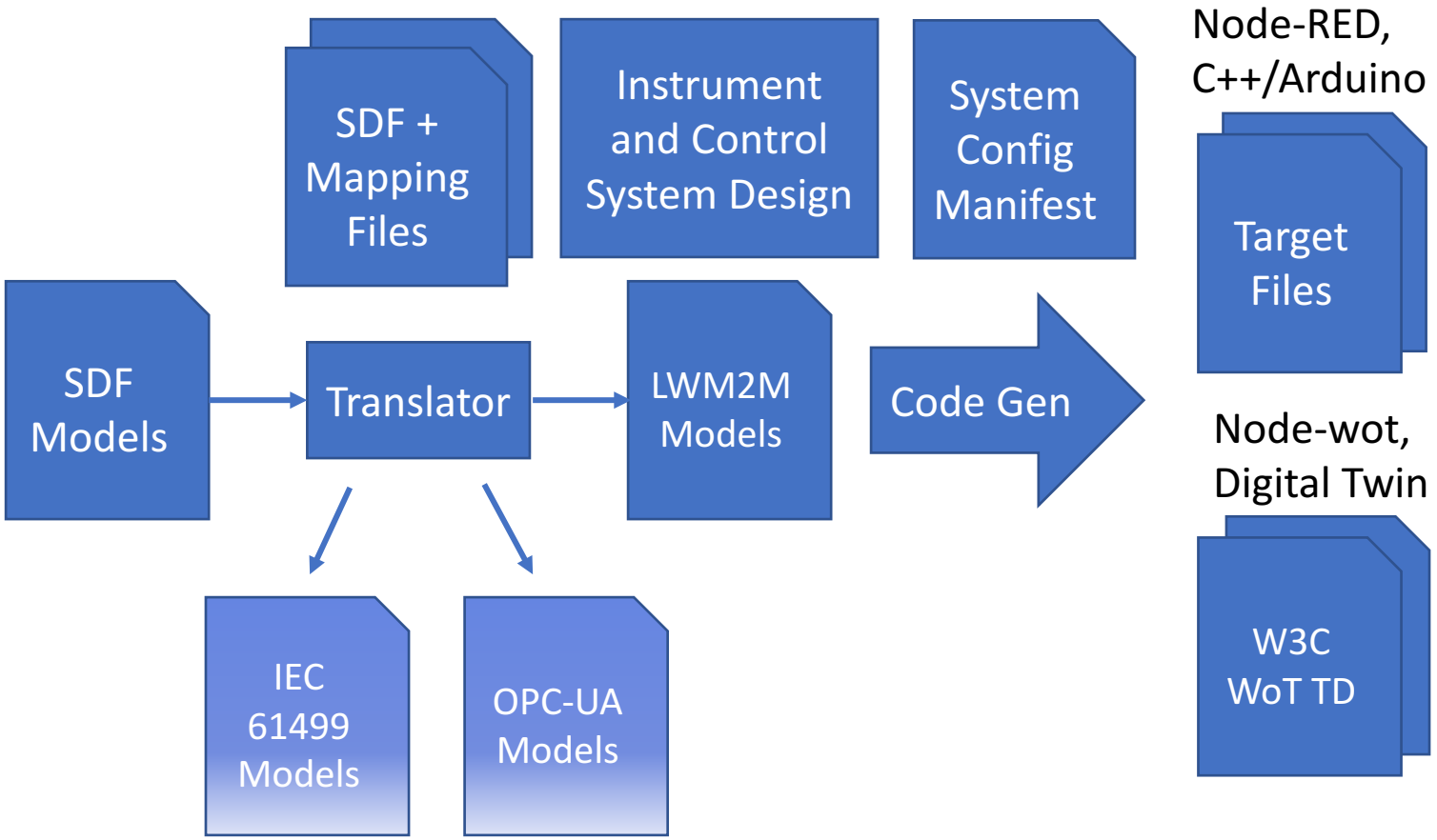
Physical Testbed and Demo

- Industrial/medical Gas Separator
- Air compressor, Air Dryer, and Pressure Swing Adsorption units
- Continuous and sequence controls
- Safety system
- Also represents an integrated product/SKU level configurations

SDK

- Tool flow
- System tools and components
- Models
- Control System Architecture

SDK Tool Flow



System Tools and Components

- System design tools and configuration manifest
- Control loop and sub-system modeling
- Centralized logging and alarms
- Digital Twin framework

Models

(SDK Developed)

- Control system models for PLC functionality
- Continuous and batch process control elements
- State machines, Programmers, Sequencers
- HMI models

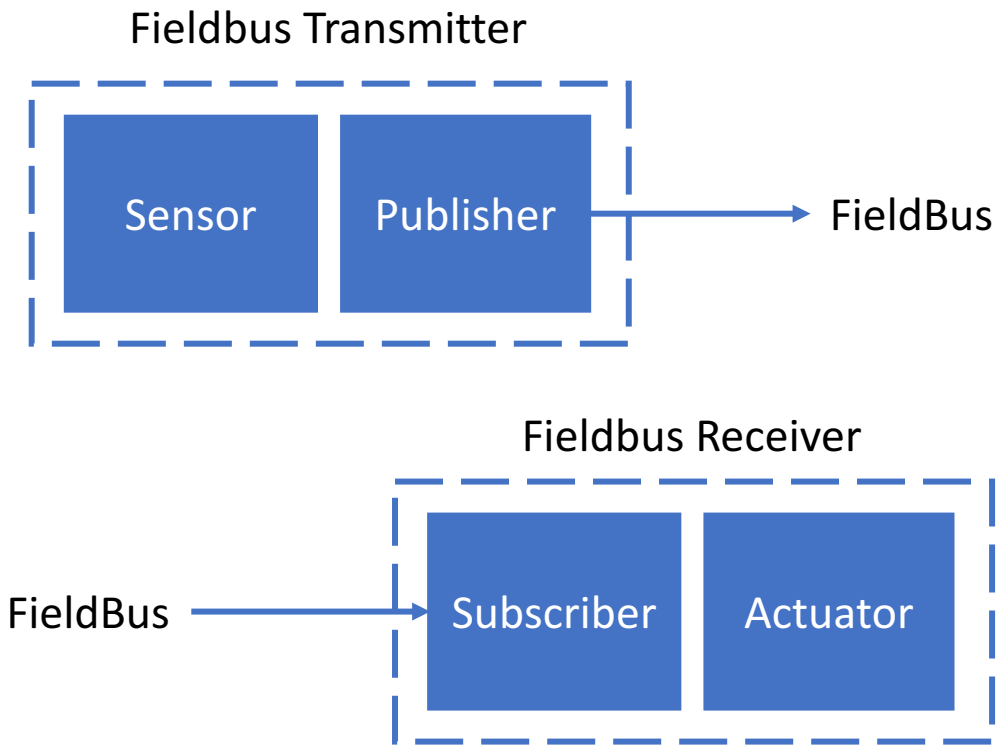
(Externally sourced)

- P&ID process models => digital twin
- Process ontologies and vocabularies
- Control algorithm abstract models
- Quantities and Units

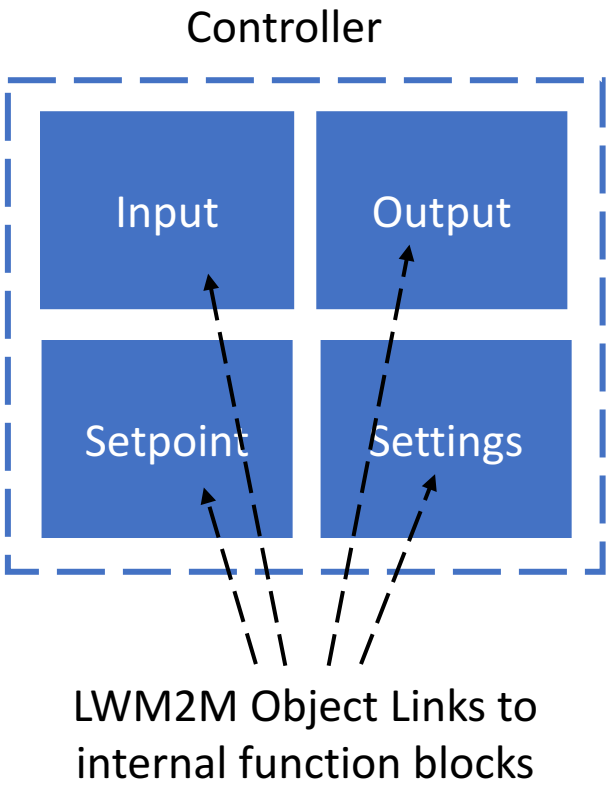
Control System Architecture

- Embedded module integrations use LWM2M Object links
- Pub/Sub - Serial and MQTT bearers initially
  - LoRA and USB through serial pub/sub bearer
  - Publisher and Subscriber Modules to customize the communication stack and adapt to Field Buses
- Example Transmitter as Sensor + Publisher
- Example Controller
- Local and FieldBus integration patterns
- System Architecture Example

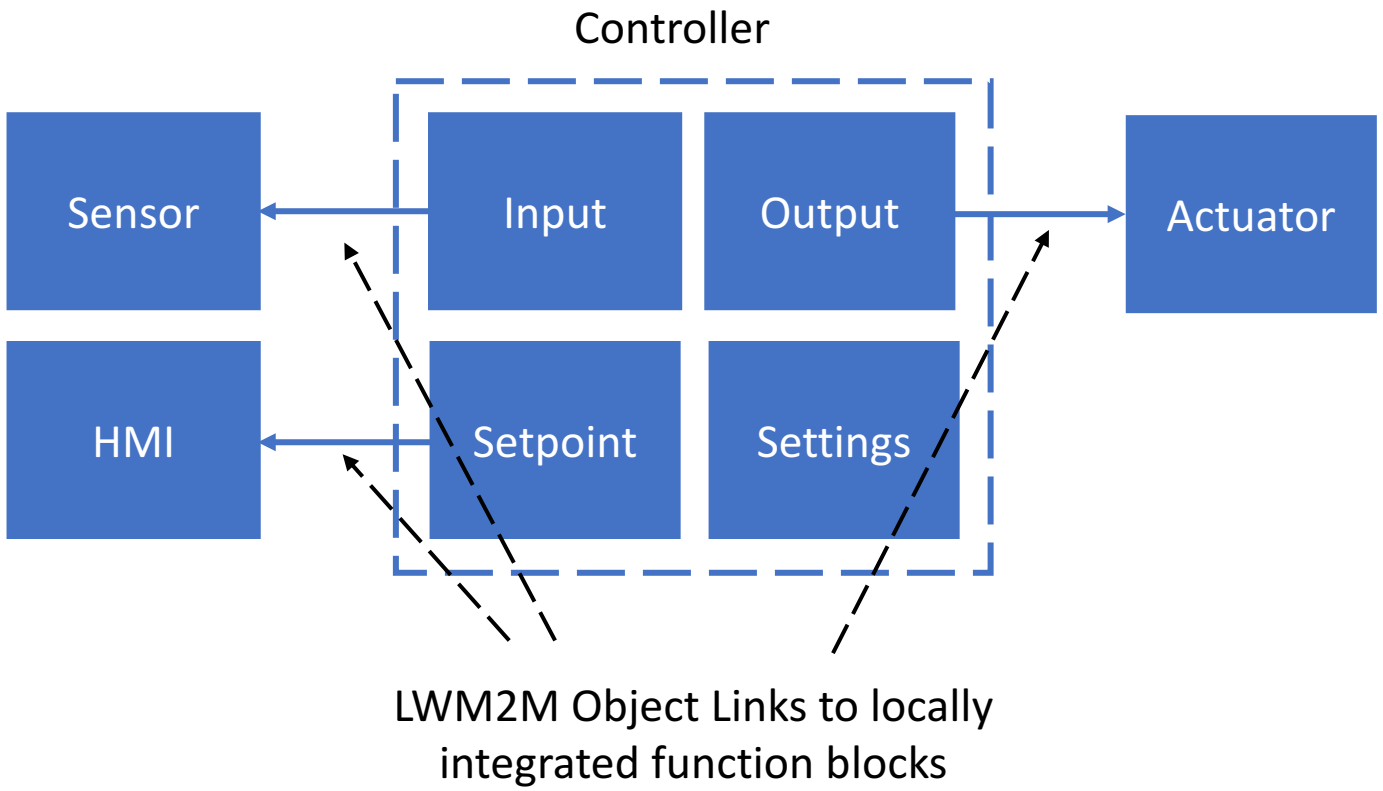
Integration Patterns – FieldBus



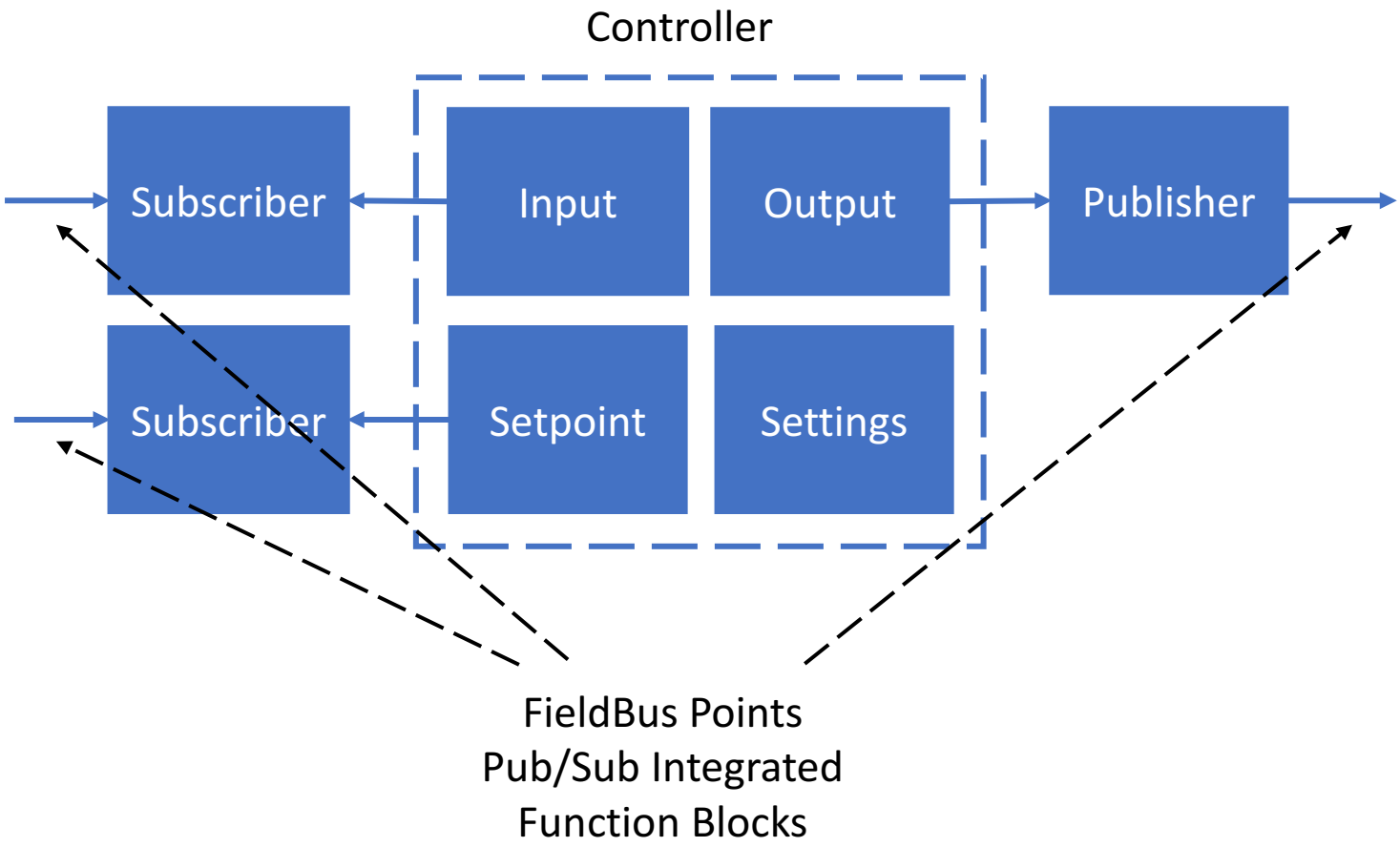
Integration Patterns - Modular Controller Example



Integration Patterns – Local Controller Integration



Integration Patterns – FieldBus Controller Integration



System Architecture Example