ATLAS Reference Architecture

Stefan Rilling
Fraunhofer IAIS
Interoperability in digital Agriculture

- Very heterogeneous landscape of machines, sensors and data platforms
- Exchange of data between all entities is a key-capability
Farmers have lots of things to manage
Complex Farms with multiple Sites

- Example from an ATLAS project partner, this is a normal farm setup
Multiple Software Tools in Use

- Example: One Farm, 7 different Software Systems
- This will probably increase in the future
Multi-Vendor Fleet Operations
Typical connected ISOBUS System

Interface: ISO-XML and EFDI
ATLAS Concept and Goals

- **Interoperability between**
  - Agricultural machines, sensors and data services
- **Standardized and extensible interfaces for data exchange**
- **Full control over the data through the farmer**
**ATLAS Interoperability Architecture**

- **High-level reference architecture**
  - Collaborative development process between industry partners, software developers, agricultural service providers

- **Designed along concrete use-cases**

- **Two basic concepts complementing each other:**
  - Data-platform based data exchange and processing
  - On-board / on-site computing and processing capabilities
ATLAS Service Mesh Network

- Each participant stays autonomous
  - Responsible for implementing and providing services
- Central components kept to a minimum
ATLAS Network Participants

- Participants are defined through
  - Own software and proprietary services
  - Identity Provider (IDP) service to store and authenticate service identities
  - Consent management system
  - Data storage capabilities
Service Registry

- **Central Component serving as a trusted directory**
  - Identified participants can register services upon request
  - Provided and required service capabilities are part of the request

- **Service capabilities are granular endpoint accesses**
  - Type of resource and operation (CRUD)
  - Service verifies that requested operation is within the capability scope
Data Services

- **Data Service Instances as the central participant component**
  - Data- and transport-technology agnostic
- **Multiple Layers defined by ATLAS or (optionally) by participant**
Data Services - Layers

- **Service Interface**
  - Establish connection and data transport

- **Service Layer Mapping**
  - Defines a generic exchange protocol

- **Supported Format**
  - Technical message type (e.g. JPEG image, EFDI live telemetry message)
  - Information message type (purpose of the information transported)
Data Service Pairing

- **Service Registry as the only component known to each participant**
ATLAS AppEngine for on-board Computing

- Provides a platform for executing applications
  - with little or no internet connectivity
  - apps that require very low latency when processing nearby sensor data to actuate adjustments in real-time on local devices
ATLAS Apps

- Apps run within the AppEngine on an onboard computer
- Different type of Apps: Real-time apps, job-apps, utility apps, platoon apps
- Apps come with a cloud-based companion service registered to the Service Registry
AppCenter provides a user interface to
- browse and search for apps
- install, update and uninstall apps on the respective AppEngines.

App publication via companion services
- AppCenter connects to Service Registry to retrieve information on available apps
**AppEngine SDK**

- **Mediation layer for apps**
  - Interact with the environment
  - Abstracting low level technologies and transport

- **Sensor API**
  - Abstracts the low-level connectivity layers to and from sensors

- **Tractor and Implement (T&I) API**
  - Retrieve information or send controls to the machinery based on existing standards (ISOBUS, TIM, Steering and Sequence Control)

- **App2App API**
  - Abstracts the low-level connectivity layers (e.g. various 802.11 variants) for inter communication between sibling platoon apps
AppEngine SDK – T&I API

- **API to access machinery specific functionality**
  - Read from / write to the ISOBUS

- **Access control and flow control layers for safety and security**
  - No direct access to the ISOBUS from the AppEngine
Certification and Safety

- Different AppEngine instances may offer different type of features
  - AEF ISOBUS certification required for AppEngine implementations destined to be installed on tractors and connecting to ISOBUS, TIM or Steering/Sequence Control
- Some app functions can impact machinery operations in a potential hazardous way
  - Example: setting tractor’s speed, unfolding implements
- Apps requiring high safety class permissions will have to undergo a stricter certification process before being approved
Summary

- **ATLAS interoperability reference architecture with two basic concepts:**
  - Data platform based data exchange and processing
  - On-board computing and processing capabilities through a self-contained computing platform

- **Only two central components**
  - Service Registry and AppCenter

- **Implementations of the architecture will be conducted along concrete use cases**
Thank you!

WP3 - ATLAS Reference Architecture

Stefan Rilling
Fraunhofer IAIS

stefan.rilling@iais.fraunhofer.de