ROS2 for Autonomous Agriculture Applications

Darcy Cook, Jorge Viramontes,
JCA Technologies, Winnipeg, Canada

The state of the industry for autonomous machines for agricultural applications is in the process of transitioning from what has been mostly concept machines and research projects to real production solutions in a wide variety of applications. This is happening with growing momentum that is moving at a fast pace driven by the market needs for increased agricultural production and efficiencies and evolving technologies that make these solutions possible. The market needs are driven from a combination of factors that include skilled labour shortages, limits of machine size growth trends, sustainable farming methods, all under the pressures of higher food production needs globally. Applications where the business cases make the most sense with the current technology are emerging across a wide diversity of areas with specific machine functions, and these will continue to expand as the technology matures and lowers the barrier to autonomy in new applications. Autonomous machines in agriculture has been enabled by broader advances in robotics technology that spans multiple technologies (e.g. vision systems, controls, navigation and guidance). Integrating these technologies requires systems that can manage the complexity of integration with architecture and tools that allow for development efforts to be focused on the application-specific areas. ROS (Robot Operating System) has emerged as the de facto standard platform for autonomous applications in agriculture (as well as many other industries). Its wide adoption is a result of its flexibility and capability of supporting wide varieties of systems, its accessibility through the open source community, and the tool set that facilitates rapid development of complex systems. Just as the autonomous agricultural machine industry is undergoing this transition from research systems to production system, ROS has evolved to make this leap from a research platform to a production platform, and ROS2 is the major initiative in this step change. This paper provides high-level overview and benefits of ROS in general, the new key features of ROS2 that enable scalable autonomous systems, the implications for autonomous agricultural machines, and how JCA has integrated and applied these technologies within the JCA Autonomous Framework (AFW).
1 Background: Integration of ROS with Agricultural Mobile Machine Controls

1.1 ROS Overview

ROS (Robot Operating System) is a set of software libraries and tools that can be used to build robotics applications. It had evolved out of robotics efforts from Stanford University in the mid-2000 and originally took shape in 2007 in the robotics incubator, Willow Garage (ROS History, 2020). The developers of ROS found that they had spent much of their time in working through technology integration and infrastructure problems with each new robot they worked on, which limited the time available to work on the core robotics problem unique to the specific robot. ROS was developed with the goal of minimizing the effort on infrastructure and integration, allowing for rapid development of new robots through reuse of core software libraries, and a defined communication architecture. This approach has become widely successful as ROS has been deployed across a wide variety of different robotic applications. The primary strengths of ROS that have facilitated its adoption are:

- **Open-Source with commercially friendly licensing** – ROS software libraries are open-source, which has resulted in a large community contributing to the development and progress of the platform. Additionally, the licensing is friendly to be used in commercial and closed-source products, which allows commercial-focused organizations to build their unique offerings on a ROS platform, while maintaining a competitive edge in their unique offerings.

- **Robot Geometry Library** – The ROS robot geometry library provides a framework for combining sensors and actuators in a common frame of reference that represents a 3-dimensional world. The ROS geometry library performs the transfer function math in the background that allows sensors to be described by a physical location on the robot/machine and the data produced by the sensors/actuators to be associated with a physical location relative to the 3D model that makes up the complete system. This drastically simplifies integrating multiple sensor systems that are mounted in different physical locations into a common model.

- **Distributed/Modular Architecture** – ROS provides a communication infrastructure that allows ROS applications to span multiple processing modules with the concept of ROS node and a pub/sub message passing system. This allows data from sensors, actuators, and communication interfaces to be interfaced to different processing modules within a system and the information to be shared throughout the system in a simple way. This facilitates edge computing and multi-processing systems with a common machine application.
• **ROS Packages** – ROS has a large number of packages that consist of commonly used functions and features needed in robotics, such as localization, pose estimation, mapping, navigation, and perception system processing.

• **Integration with other libraries** – ROS integrates well with third-party libraries that are also commonly used in advanced automation applications, such as OpenCV (OpenCV, 2020) used for vision system processing, Gazebo (Gazebo, 2020) used for simulation, and Point Cloud Library (Point Cloud Library (PCL), 2020) use for image and point cloud processing.

• **ROS Tools** – Several highly capable tools have been developed as part of the ROS ecosystem that have proven to be extremely useful in development of complex robotics systems. These include tools such as Rviz, a visualization tool that can be used to visualize multiple sensors in the ROS frames of reference, and ROS Bag, a logging tool for logging and playback of data collected across the entire system.

Fig. 1: Example output from ROS Rviz Tools
These key features provided a strength in development of complex systems need for advanced automation and robotics systems that had not been prevalent in simpler mobile machine control systems.

While ROS provides strong benefits to adapting to the technologies and complexity needed for robots and autonomous machines, it was originally developed for very different use cases than what is needed in agricultural mobile machines. The use cases that guided the original development of ROS (Gerkey, 2020) were:

- A single robot
- Workstation-class computational resources on board
- No real-time requirements (or, any real-time requirements would be met in a special-purpose manner)
- Excellent network connectivity (either wired or close-proximity high-bandwidth wireless)
- Applications in research, mostly academia

These use cases allowed for focused development of ROS to meet its goals of developing infrastructure that facilitates rapid development of complex robotic systems, without being handcuffed by ensuring the systems were scalable for production-intent systems. This has been very useful in the development of concept and prototype machines, but has some barriers when the goal is scalable robotics systems. It is useful to contrast this with the evolution of agricultural machine controls systems, which has been driven from a need to meet practical and business needs for agricultural machines on a large scale.

1.2 Agricultural Mobile Machine Controls

Agricultural mobile machine controls have developed over several decades on a very different path than the robotics world from which ROS emerged. Through the 1980s and 1990s machine controllers for agricultural systems developed on a path towards ruggedized and reliable ECUs that had heavy overlap in technology to that of the automotive electronics market (Stone, Benneweis, & Van Bergeijk, 2008). The division of separate machines between the powertrain (the tractor) and the machine application (the implement) is more common agriculture than in other off-highway industries (such as mining, construction, and forestry), which has resulted in the need for standard methods of communicating between the tractor and the implement controls. This drove the development of the ISOBUS standard (ISO11783, 2017) with its Universal Terminal (UT) concept that could be used in the tractor as an interface to variety of different
implement ECUs. This has developed an approach of standardization and collaboration within agriculture that sets it apart for many other similar industries.

Another core technology that has heavily influenced the evolution of agricultural mobile machine technology is GNSS/GPS satellite systems, which has been in the catalyst to what is known as precision ag systems, and has been applied heavily in planting, seeding, spraying, and harvesting applications. These systems have all evolved in parallel with connectivity and mobile device technology fueled the evolution of farm management information systems (FMIS) that integrate with machine controls. Each of these key technologies and innovations have resulted in complex tapestry of offerings that have caused continual integration challenges for agricultural systems. To combat these integration challenges, the industry has pushed towards standardization efforts, driven by organization such as AEF (AEF ISOBUS, 2020), with the goals of providing defined functions and interfaces that allow multiple parties to integrate systems together. This has been supported with initiatives such as Plugfests (AEF ISOBUS, 2020) used to facilitate collaboration and conformance to these standards.

This progression of mobile machine controls technology in agriculture has resulted in real-time control systems, rugged and robust electronics, and structured, defined, and standardized module interfaces built around CAN-based communication networks.

While this model has worked well for robust, reliable, and scalable systems it does not adapt well to the complexity needed for advanced autonomous machines. Autonomous machines have an order of magnitude complexity compared to traditional machine controls, as there are many more functions and subsystems required. This is where the strengths of managing complexity offered by ROS world need to combine with the robustness and reliability of the agricultural machine controls world to result in effective and scalable agricultural autonomous systems.

1.3 Early Adoption of ROS in Agricultural Machines

The initial use of ROS in agricultural machine applications has come through research and concept machine development, largely for autonomous machines. The technology and systems available through traditional agricultural machine systems could not handle the complex systems needs for advanced perception systems, simulation, guidance and navigation needed for autonomous machines. ROS has been used in the development of many different autonomous agricultural machines, but most often on hardware platforms that are of a prototype nature, and with glue-technology that works for a one-off demonstration, but would not be sustainable in a production environment. This has been in alignment with the original use cases.
of ROS (i.e. research systems), but as the technology and industry for autonomous agricultural machines matures, it is no longer sufficient to produce machines that cannot be extended to production systems.

2 ROS2 – Expansion to Scalable Robotic Platforms

As ROS has gained in momentum, and robotics technology in general has matured, there has been a need to expand beyond the original use cases for ROS, which is ultimately driven from the need for robotics systems to emerge from research and academic environments towards scaled production systems. ROS2 has been developed with a focus on new use cases that enable scale robotic systems, which are defined (Gerkey, 2020) as:

- **Teams of multiple robots** – Expansion from the single robot use case to a system that considers multiple robots.
- **Small-embedded platforms** – Consideration of microcontroller-based processing platforms integrating into the ROS ecosystem.
- **Real-time systems** – Support of real-time systems considering inter-process and inter-machine communication.
- **Non-ideal networks** – Robust operation in networks that may suffer from intermittent and poor-quality communication.
- **Production environments** – Use of ROS beyond lab and prototype environments, to be expanded to real-world production applications.

These use cases are very exciting in the context of scaled autonomous agricultural systems, as they address many of the concerns with ROS related to its suitability beyond prototype and research systems.

The following sections describe, at a high-level, some of the key changes implemented in ROS2 that help to achieve the targeted use cases. This is not an exhaustive or in-depth description of ROS2, but rather a highlighting of some of the key components that are significant in the consideration of autonomous agricultural machines.

2.1 ROS2 Architecture

In order to address the targeted use cases effectively, it was determined that breaking changes to ROS were needed, which prompted the development of ROS2 (as opposed to further development of ROS1 to achieve these goals). Some of these changes are realized in the underlying architecture of ROS2. **Fig. 2** shows key components of the ROS2 architecture as compared to ROS1.
Some of these key differences are:

- **OS Support** - ROS1 is supported mainly by the Linux operating system, where ROS2 has more operating system support including Linux, Windows, Mac, and can be expanding to other real-time operating systems in the future. This enables the ability to expand the processing platforms for ROS, with potential to expand to real-time OS, supporting the embedded and real-time use cases for ROS2.

- **Use of DDS** – DDS (Data Distribution Service) is a communication transport middleware that supports a publish-subscribe transport for messaging (Woodall, 2020) that was introduced in ROS2. DDS is based on a defined standard and has a strong technical credibility as it been implemented in mission critical applications, such as space and flight systems, locomotive systems, financial systems, and has multiple implementations that are supported from different vendors. The change to DDS maintains the publish-subscribe messaging model that was used in ROS1, but also provides additional reliability, quality of service, security, and modularity that allows ROS2 to be effective with non-ideal communication networks and in production environments (two of the key ROS2 use cases).

- **No ROS Master** - ROS1 was developed with a concept of a ROS master that acts as a server for multiple nodes within the system. Communication between ROS nodes is routed through the ROS master. This is a centralized communication system, so while distribution of ROS nodes can be across multiple processing systems (and even multiple machines), one processing module (or one machine) has to run the ROS master, and as a result acts as the master for the system. This means that the ROS master must always be present in the system for other ROS nodes to communicate with each other, which drives the need for a master or hub in multi-machine systems. This architecture change drastically improves the potential to support the teams of robots use case.
2.2 ROS2 Communication
The ROS1 communication model is based on topics and services. Topics allow ROS nodes to communicate to each other in a publish/subscribe method, where one ROS node publishes a topic, and other ROS nodes may subscribe to that topic if they wish to receive information associated with that topic. Services are an alternative method of communication to topics, where instead of the publish/subscribe method, a service is defined by a pair of request-response messages between ROS nodes. This method is more appropriate when specific information is to be exchanged between ROS nodes. In general, the ROS1 communication model is dependent on the ROS master within the architecture to route messages between nodes, which limits the use in multi-machine systems, as well as is more susceptible to network communication reliability issues.

In contrast, the ROS2 communication architecture adopts a DDS-based communication model that enables a dynamic, distributed, and robust data transport. This introduces the concept of “domains” in which ROS nodes belong to and can communicate within this domain. Independent domains may exist within the same processing module, and domains can extend across multiple machines as well. The communication model applied within ROS2 consists of (Maruyama, Kato, & Azumi, 2016):
• **Domain Participants:** The domain participants are ROS nodes that can publish and/or subscribe to a global data space within a defined domain.

• **Publisher:** The publisher is responsible for data issuance and manages one or more data writers that send data to topics within the domain.

• **Subscriber:** The subscriber is responsible for receiving data and making the data available, acting on behalf of one or more data readers.

• **Data Writer:** The data writer publishes data of a specific type through a publisher.

• **Data Reader:** The data reader reads data of a given type associated with a subscriber.

• **Topic:** Similar in concept to ROS1, a topic identifies a data-object between a data writer and a data reader. A topic has a specific name and data type.

• **Quality-of-Service (QoS) Policy:** All entities have a QoS policy that defines the data transport behavior. It specifies communication parameters that define reliability, continuity, and message timing deadlines for each participant. This is an important addition to ROS2 that has a major impact in reliability of communication in networks that have less than ideal connection reliability.
2.3 ROS2 Security

A key reason for the adoption of DDS as communication transport middleware in ROS2 is to take advantage of the security benefits possible with DDS. The DDS Security Specification (Object Management Group, 2018) is an expansion to the DDS specification that defines a Service Plugin Interface (SPI) for a specific set of security functions. There are five SPIs defined (Fazzari, 2020), which are:

- **Authentication** – Verifies the identity of each domain participant
- **Access Control** – Enforces restrictions on the DDS-related operations that can be performed by an authenticated domain participant
- **Cryptographic** – Handles all encryption, signing, and hashing operations
- **Logging** – This is an optional SPI that provides the ability to audit DDS security related events
- **Data Tagging** – This is also an optional SPI that provides the ability to add tags to data samples

An architecture diagram of how these security SPIs fit together is shown in **Fig. 4**.

![DDS-Security Architecture](image)

**Fig. 4: DDS-Security Architecture from (Object Management Group, 2018)**

The details of implementation of the security components are beyond the scope of this paper, but the important takeaway in understanding enhancements offered by ROS2 is that secure communications have been a focus, these have been implemented through the DDS standards, which are proven and mature technologies. This is a critical component of expanding ROS2 based systems to production environments, especially in safety-critical applications like autonomous agricultural machines.

### 3 Scalable Autonomous Agricultural Machines

Autonomy in agricultural applications are a subset of the wider applications where ROS can be applied, and similar to maturation of the robotics industry as a whole is going through, the next step in autonomous machines in agricultural systems is to make the leap from concept and prototype systems to scalable deployed machines. The approach taken by the ROS
community in defining the relevant use cases that are necessary for achieving scalable autonomous systems. Use cases that apply to autonomous agricultural machines extend beyond the specific technologies needed to achieve these use cases, but rather define some of the core characteristics that are needed to be achieved in broad terms for truly scalable autonomous systems in agriculture. We have defined these use cases based on JCA’s experience across a wide variety of applications to be:

1. **Multi-machine missions** – Mirroring the ROS2 “teams of robots” use case, scalable autonomy for ag machines requires the heterogeneous machine types from working together in a common mission. This may include a mixture of autonomous machines and operator-driven machines.

2. **Real-time controls** – Closed-loop real-time machine controls is a key part of agricultural machines. These are applied in drivetrain functions (steering and propulsion), as well in a wide variety of implement control applications (typically hydraulic and/or electric systems). Robust controls are a critical element of scalable agricultural autonomous machines.

3. **Integration with existing vehicle networks technology (CAN / J1939 / ISOBUS)** – CAN-based machine communication networks are standard in most agricultural machines, and this long history of use has proven this to be an effective and robust method of communication for a wide variety of machine messaging. Limits to this technology are reached as higher data rates are needed for more advanced systems and new wired communication systems (primarily Ethernet networks) have been evolving to meet these needs. However, these higher data rates will not replace CAN networks, but rather augment them. The depth of technologies that have developed in the mobile machine industry around CAN bus technologies means that autonomous agricultural systems need to be CAN compatible.

4. **Environmentally ruggedized processing platforms** – Agricultural machines operate in harsh outdoor environments, the “workstation-class computational resources” required by ROS are typically not available in processing platforms that are made to survive high vibration, wide temperature, dusty and wet, and electrically noisy environments, which is where agricultural machines are used. Any system that depends on processing platforms that do not meet this need will not be scalable in agricultural applications.

5. **Operation in areas without Internet connectivity** – Agricultural machines are often used in areas where there is not dependable Internet connectivity. Agricultural
machines that can be scaled need to be able to operate effectively in these areas. While certainly there are major efforts underway to expand connectivity to remote areas, reliable Internet connectivity in remote areas will not be a given still for many years to come, so developing autonomous agricultural machines that depend on this connectivity may not work for many use cases.

6. **Connected machines over wide work areas** – There is the need for machines to communicate with each other over a wide working area, which can be especially challenging in areas without Internet connectivity. Machines that are working on a common mission over areas that are more than a few kilometers away are common in agricultural applications, so the ability to exchange key information related to a common mission over these areas are needed for scalable autonomous agricultural machines.

7. **Integration of tasks to mission management** – Agricultural machines exist for the purpose of execution of a task, whether this is spraying, seeding, harvesting, transporting, baling, mowing, or any number of different agricultural machine tasks. The management of these tasks includes planning, deploying, executing, monitoring, and analyzing the task results. Autonomous missions include management of both the task, as well as management of the path that the machine will drive to execute this path. The concept of the task is not new to agricultural systems, and has a long history of development towards standard methods of managing and communicating these tasks, much of this defined by the ISOBUS Task Controller of the ISO 11783 specification (ISO11783, 2015). There has been a large ecosystem of technology developed around these standards, and these will not be displaced suddenly by any one new autonomous machine, so integrating task control that is compatible with existing systems and standards is an important part of scalable agricultural autonomous machines.

8. **Integration with Farm Management Information Systems (FMIS)** – As an extension to the task management, there has been a rapid development of farm management software systems that interact with machines that are executing farming tasks. These FMIS enable analytics that agronomic and operational efficiencies, and integration with the ag machines are a key component. These FMIS are an industry unto themselves, and common interfaces are developing to allow for farmers to have independent choices across FMIS and machines that best fit their needs. Integration with wide variety of FMIS are a key component of scalable autonomous agricultural machines.

9. **Secure platforms** – The connectivity that is needed to enable much of the multi-machine, mission management, and FMIS connectivity comes with potential of security concerns that could open up machines to have unauthorized use, interfere with
authorized use, or expose application data. Scalable agricultural autonomous machine systems need to apply robust and proven security systems to prevent any of these potential security breaches.

10. **Functionally safe systems** – Functional safety has developed significantly in the mobile machine industries over the last few decades. Several standards have been developed to define levels of functional safety, and provide guidelines for the development of functionally safe machines. Autonomous machines bring new challenges with safety systems, in particular, this brings challenges in applying existing standards to more complex and advanced systems that are needed for perception systems that are capable of object detection and recognition. Safety is a key concern, and considerations and solutions to address safety for autonomous machines are needed for scalable systems.

Even with a cursory examination of these use cases, it is clear that ROS2 alone does not address the needs of platforms for autonomous agricultural machines, however it can play a significant role in meeting many of these needs. The JCA AFW is a platform of technologies aimed at addressing all of these use cases, the role of ROS2 within the AFW provides an indication of how ROS2 is applied for autonomous agricultural machines.

4 **JCA Autonomous Framework (AFW) Overview**

The JCA Autonomous Framework (AFW) (Cook, 2020) is a set of technologies that serves as technology building blocks that enable the rapid development of autonomous agricultural machines. These technology building blocks consists of hardware and software systems that span all core subsystems of autonomous machines, including perception systems, power and drivetrain, localization and mapping, mission management, human-machine interfacing (HMI), communication and data management, and safety systems.
Similar to the philosophy that spurred the development of ROS, the JCA AFW has been inspired by the idea of reducing the development effort of the underlying infrastructure needed for autonomous agricultural machines to allow effort to be spent on the areas that make each machine unique in its application. This is achieved through providing these building block technologies that can be adapted and used for customized autonomous machines. The development of the JCA AFW has been guided by meeting the needs of the use cases defined in the previous section towards a platform that is suitable for scalable autonomous agricultural systems. Key components of the JCA AFW are:

- **Advanced processing platforms** – Ruggedized computing platforms made for execution of autonomous master, perception, guidance and localization, and mission management
- **ROS-node controllers for real-time machine controls** – Controllers capable of real-time closed-loop machine controls, interfaced to ROS-based processing systems
- **Mission management system** – Mission planning, deployment, monitoring, execution, and analysis of multi-machine autonomous missions
- **Perception platform** – Object detection, object recognition, and advanced sensor interfacing software systems
- **Multi-machine communication** – Communication system for coordinated multi-machine autonomous systems with a common mission
- **Flightpath cloud platform** – Task management software, including map management, task progress, and data management across multiple machines. Interfacing with FMIS and task controller compatibility.

- **Guidance and localization software** – Autonomous machine guidance, navigation, and localization software

- **Safety module and alarm management** – Safety monitoring, watchdog, and alarm management

- **Simulation systems** – System-level simulation for verification

The complete scope of the JCA AFW is beyond the scope of this paper, however highlights of key components of the JCA AFW that specifically relate to the use of ROS2 are highlighted in the next sections.

### 4.1 Advanced Processing Platforms

The JCA Eagle (JCA Eagle Feature Sheet, 2020) serves as processing platform for autonomous agricultural machine systems. It has been designed specifically for surviving the harsh environmental conditions encountered in agricultural applications (wide temperature, high vibration, sealed for dust and water ingress, electrical mobile machine conditions). It features the NVIDIA Jetson Xavier SOM for processing power that provides advanced edge-computing capabilities, dual RTK-GPS for orientation and localization, multiple high-speed (Gigabit ethernet), wireless communication interfaces, cell modem capabilities, up to 8 camera interfaces, 4 CAN/J1939 interfaces, and inputs/outputs made for machine control applications.
This platform can run both ROS/ROS2 (simultaneously) on a Linux-based software platform. It provides a key hardware component that addresses several of the ag autonomous use cases, most significantly the environmentally ruggedized processing platforms capable of ROS2 execution.

JCA also has other linux platforms, such as the JCA Hummingbird controller that are suitable for simpler applications that don’t require some of the sophisticated equipment needed for autonomous machines but would still benefit from the use of ROS/ROS2.
4.2 ROS-Node Controllers – Real-Time Machine Control

The JCA ROS node controllers includes a family of machine controllers that are capable of real-time closed-loop controls needed for machine controls through configuration and commanding from a ROS system (JCA ROS Node Controllers, 2020). This is enabled through the combination of ROS nodes that provide an interfacing to commanding over a J1939 bus of controllers that have a defined API that allow for I/O control, PID control loops, and variety of other functions. The application function of the real-time controllers are defined in the ROS application, eliminating the need for custom application software on the real-time controllers, and direct interfacing through a ROS application.
This JCA AFW technology addresses the real-time controls and integration with existing vehicle networks use cases for scalable ag autonomous machines.

4.3 JCA AFW Multi-Machine Communication System

JCA has also developed a communication system that addresses several of the defined ag autonomous use cases, specifically the multi-machine missions, operating in work areas without reliable Internet connectivity, connected machines over wide work areas, and secure platforms.

To address the challenges of needing both high bandwidth communication, and long-distance communication, multiple frequency bands are included in the communication system, with definition of messaging types that include:
• **Persistent** – Persistent communication is used to send mission critical information between all machines in the system. This communication is over a long range, and has a lower data rate that proximity communication. Persistent messages include critical health and status messages, RTK corrections, and low-resolution mission progress updates. This is implemented over a physical layer that operates in 868-900 MHz range.

• **Proximity** – Proximity communication is used to transmit high-data rate information to machines that are in close proximity to each other. This communication has shorter range, but with a high data rate. Proximity messages includes mission deployment, detailed mission progress and task information, video feedback, manual machine controls, and machine-to-machine coordination. This is implemented over a physical layer that operates in 2.5 GHz and 5GHz ranges.

The JCA AFW architecture supports a multi-machine system that includes a mission control center that coordinates missions across multiple machines in a given work area, multiple machines, and multiple users with HMIs as an interface to the systems.
Each of the machines within the system, as well as the mission control center run ROS, so this communication system allows for broad multi-machine communication using ROS infrastructure. The key features of ROS2 communication that consider security and non-ideal networks allow for this communication system to expand ROS-based multi-machine systems reliably for agricultural autonomous machines. Specifically, secure communications and quality-of-service attributes implemented with ROS2 provide a solid base in which to build the communication systems that address the specific needs for agricultural systems.

The mission control center in the multi-machine system has capability for Internet connection when available (through cell connections), that allow for cloud system connections for integration with mission control functions that include remote planning, monitoring, and analysis.

![Diagram Showing Communication of Systems within a Work Area](https://doi.org/10.51202/9783181023747-103)

*Fig. 10: Diagram Showing Communication of Systems within a Work Area*
5 Summary of ROS2 Applicability to Autonomous Agricultural Applications

ROS2 provides a key step towards scalable autonomous agricultural machines. The table below summarizes the applicability of both ROS2 and the JCA AFW components have in addressing the key use cases or scalable autonomous agricultural systems. It can be seen from this that while ROS2 does not address all areas needed for these machines, it provides a core technology component that takes a meaningful step on the direction of increasing the reliability, robustness, and scalability of autonomous agricultural machines. The JCA AFW builds on ROS2 as a core technology, and further bridges the gaps between robotics and agricultural machine controls, providing a platform that can be used to develop customized agricultural autonomous controls.
Table 1: Summary of ROS2 and JCA Components Relative to Autonomous Agricultural Machine Use Cases

<table>
<thead>
<tr>
<th>Autonomous Agriculture Machine Use Case</th>
<th>ROS2 Applicability</th>
<th>JCA AFW Components</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multi-machine missions</td>
<td>Direct applicability – Expansion to “teams of robots”</td>
<td>AFW Mission management, communication system, FlightPath cloud</td>
</tr>
<tr>
<td>Real-time controls</td>
<td>Indirect applicability – ROS2 is moving towards supporting embedded MCU platforms (with limited functionality), a more complete set of real-time features is available if used in conjunction with JCA ROS node controllers</td>
<td>ROS node controllers provide the bridge between ROS systems and real-time controls</td>
</tr>
<tr>
<td>Integration with existing vehicle networks technology (CAN / J1939 / ISOBUS)</td>
<td>Not addressed</td>
<td>JCA Eagle and Hummingbird platforms provide this capability of ROS-based systems with CAN interfaces</td>
</tr>
<tr>
<td>Environmentally ruggedized processing platforms</td>
<td>Not addressed</td>
<td>JCA Eagle and Hummingbird platforms provide rugged hardware platforms for ROS systems</td>
</tr>
<tr>
<td>Operation in areas without Internet connectivity</td>
<td>ROS2 adoption of DDS transport provides reliable expansion of communication that can be used for this.</td>
<td>JCA multi-machine communication system designed specifically for this use case, enabled with ROS2 communication</td>
</tr>
<tr>
<td>Connected machines over wide work areas</td>
<td>ROS2 adoption of DDS transport provides reliable expansion of communication that can be used for this.</td>
<td>JCA multi-machine communication system designed specifically for this use case, enabled with ROS2 communication</td>
</tr>
<tr>
<td>Integration of tasks to mission management</td>
<td>Not addressed</td>
<td>JCA FlightPath Cloud system integrates mission management and task management compatible with ISO11783 Task Controller</td>
</tr>
<tr>
<td>Integration with Farm Management Information Systems (FMIS)</td>
<td>Not addressed</td>
<td>JCA FlightPath Cloud system integrates provides APIs for connection with farm management systems</td>
</tr>
<tr>
<td>Secure platforms</td>
<td>ROS2 provides security build on DDS-security. This provides secure communications within local work area communication</td>
<td>JCA multi-machine communication system designed specifically for this use case, enabled with ROS2 communication. JCA FlightPath Cloud systems also provide security (unrelated to ROS) for cloud systems.</td>
</tr>
<tr>
<td>Functionally safe systems</td>
<td>Not addressed directly, but reliable and deterministic communication provides strong base to build on for functionally safe systems.</td>
<td>JCA safety module and alarm management systems provide safety systems tailored to the needs of autonomous machine systems</td>
</tr>
</tbody>
</table>
6 The Road Ahead

ROS2 has significant improvements that have built on the success of ROS systems, and grown through the adoption within the open-source community. There is still more evolution of this technology to come. Groups such as ROS Agriculture (ROS Agriculture, 2020) as well as many universities, OEMs, and technology companies continue to push the boundaries of the technology with new use cases and applications, which are collectively driving improvements in all areas. It is clear that any organizations that are serious about autonomy in agriculture are beginning to adopt ROS in a big way, and concerns of a lack of suitability of ROS to production systems are becoming outdated. There is no single organization that will be able to develop similar core technologies at the pace the ROS continues to develop. It is also clear that ROS alone will never meet the complex needs of autonomous agricultural machines, as this is not its intended scope. The JCA Autonomous Framework provides this bridge between the available technologies, such as ROS, that facilitate autonomous machine development, and the specific needs of the agricultural machine industry, with a platform that can be adaptable to needs of any unique machine application.
References

**AEF ISOBUS.** (2020, 09 07). Retrieved from AEF: https://www.aef-online.org/about-us/isobus.html


JCA *ROS Node Controllers.* (2020, 09 08). Retrieved from JCA Electronics: http://docs.jcaelectronics.ca/jcaroscontroller/


OpenCV. (2020, 09 07). Retrieved from OpenCV Website: https://opencv.org/

*Point Cloud Library (PCL)*. (2020, 09 07). Retrieved from Point Cloud Library (PCL) Website: https://pointclouds.org/


*ROS History*. (2020, 09 07). Retrieved from ROS Official Website: www.ros.org/history/

