Advanced Research + Invention Agency ARIA

POSITION PAPER

Revolutionising the robotics ecosystem through enhanced modularity and interoperability

Acknowledgements

The members of ARIA's external expert committee for the Robot Modularity, Dexterity, and Standardisation programme extend their thanks to Dr Jenny Read (Programme Director), Dr Radhika Gudipati (Technical Specialist), and Paul Brown (Programme Specialist) for their valuable support and feedback, which significantly shaped the recommendations in this report.

They also express their gratitude to all participants in the ARIA workshop held on 13 March 2025 for their valuable contributions and the Manufacturing Technology Centre for hosting the event. Special thanks go to the keynote speakers and panel members: Dr Aron Kisdi (Autodiscovery), Dr Rob Buckingham (UKAEA), Takiyah Williams (BSI), Nikos Pronios (Innovate UK), Jeremy Hadall (Satellite Applications Catapult), and Keith Mallinson (WiseHarbor).

Thanks to David Bisset (euRobotics aisbl), Karol Janik (MTC), Eirini Malliaraki (Renaissance Philanthropy) for independently reviewing and revising the document. David Bisset acted as an editor to refine the report recommendations.

Foreword

As artificial intelligence continues to advance in areas such as language, perception, and reasoning, the gap between what machine intelligence can achieve *in silico* and what robots can *physically do* is becoming increasingly apparent. This gap represents more than a technical challenge; it is a profound strategic opportunity. For a nation like the UK with a rich history of industrial innovation and a pressing need to boost productivity, the ability to design, build, and deploy advanced robotic systems is not a luxury—it is essential for our future economic prosperity and sovereign capability. ARIA's Robot Dexterity programme aims to unlock this by making robots that can handle objects with unprecedented precision and skill.

A key reason why robotics has not yet progressed as rapidly as AI is the problem this paper seeks to address: a fragmented ecosystem. Progress is hampered by proprietary systems that don't communicate, by teams reinventing the wheel on every project, and by a lack of the shared frameworks needed to build safe, secure, reliable and scalable innovation. This siloed approach is inefficient, costly, and unsustainable. For ARIA, it means that the breakthroughs we are fostering in hardware and design risk being stranded, unable to achieve widespread impact.

To identify a practical route forwards, in December 2024 I convened a group of experts tasked with making practical recommendations that would foster robotics innovation. To inform their work, we launched a national survey in February 2025, gathering insights from developers, engineers, researchers, and system integrators (see results in the survey report: *Perspectives on the Future of Robot Modularity, Interoperability, and Standards*). In March 2025, we brought together over 100 stakeholders to a workshop at the Manufacturing Technology Centre in Coventry, where we explored strategic priorities alongside technical debates.

This paper is the result of their work. It does not offer easy answers, but it does provide a clear, actionable roadmap. Its recommendations are designed to build a national advantage by creating shared, pre-competitive assets—like a common design library and development resource—that will benefit the entire ecosystem. I would like to thank everyone who contributed to this endeavour, whether by completing the survey, participating in the workshop, or offering informal input. I am especially grateful to the authors for their leadership of this process and dedication in delivering this report.

The UK now has a unique opportunity to lead in robotics and automation. The AI revolution, combined with our world-class research base, puts us in a strong position to shape the future. By building the right ecosystem – with the supporting assets and standards – we can not only accelerate domestic adoption, which has been slow to date, but also influence the global robotics landscape for decades to come. I invite you to engage seriously with the ideas in this report, and to help realise a future where sustainable, safe and secure robots enable humans to flourish.

Dr Jenny Read, Programme Director

Authors

Robert Richardson is Professor of Robotics, at the University of Leeds where he heads up the Real Robotics Lab. His research interests include robots in the real world, in the air, on the ground and underwater. He is a member of the UK Robotics Growth Partnership Committee and Ex-Chair of the EPSRC UKRAS network.

Ipek Caliskanelli is a Principal Research Engineer at the UK Atomic Energy Authority. Her work focuses on interoperability, coordination, and collaboration among large-scale robotic systems in high-consequence environments, with a strong emphasis on advancing technology readiness levels and integrating state-of-the-art robotics into nuclear applications.

Mini Rai, a UK All-Party Parliamentary and Scientific Committee member, is the founding Director of Orbit Rise Ltd. She is a space engineer specialising in robotics, automation and control for extreme environments. Previously. Mini was a Chair Professor at the University of Lincoln and an Associate Professor at the Surrey Space Centre, UK.

Having previously successfully floated a technology business on the London Stock Exchange, Dominic Keen launched Britbots to support best-in-class UK-based robotics, AI and automation ventures, capitalising on British technical expertise in these areas. To-date Britbots has backed over fifty companies. Dominic is a member of the UK Robotics Growth Partnership and also sits on the investment committee of New Anglia Capital, a public-sector venture capital fund.

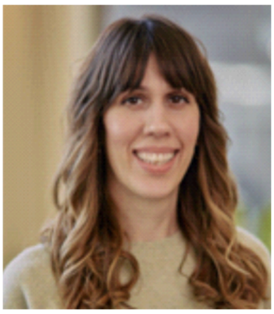
Aaron Prather is the Director of Robotics and Autonomous Systems at ASTM International, leading global efforts in robotics standards, workforce development, and safety. He bridges industry, academia, and government to advance responsible robot deployment across sectors.

Other Contributors

David Bisset (euRobotics aisbl), Radhika Gudipati (ARIA), Karol Janik (MTC), Paul Brown (ARIA), and Eirini Malliaraki (Renaissance Philanthropy).







Executive Summary

Robotics and Autonomous Systems (RAS) are poised to revolutionise society by significantly boosting productivity, efficiency, and adaptability. Yet, widespread adoption remains constrained by several factors, including high implementation costs, fragmented architectures, and a lack of unified standards and frameworks. These challenges are rooted in a lack of modular design guidance and interoperability, with industry uptake slowed by vendor lock-in, technical fragmentation, and insufficient incentives.

To address these challenges, this paper advocates for a coordinated strategy centred on three foundational recommendations. The first is to conduct a wide-scale analysis of the current use of modularity and interoperability in RAS to create an open resource for developers. The second is to establish a comprehensive RAS Development Resource, which would promote best practices, provide guidance on standards, and build a lasting ecosystem for modular design. The third involves creating a RAS Design Library with detailed, step-by-step instructions for building sophisticated robotic systems based on these best practices, thus reducing duplicated effort and fostering a new, more efficient design methodology.

Together, these will promote the standardisation of key technical areas, including the development of common protocols for inter-module communication to allow different robots to work together seamlessly, unified conventions for human-robot interaction, a unified robot API to streamline development across diverse platforms, and standard formats for specialised data such as from tactile and other sensing.

Safety and security are paramount for gaining public trust and achieving commercial viability. Given its importance, we make a further three recommendations in this area specifically. We propose a unique Robot Identification Number to ensure traceability and verify ownership; a standardised "black box" module to record operational data, providing an auditable log for incident investigations; anomaly detection techniques to identify hardware faults, software errors, or malicious interference.

Looking ahead, embracing these principles could enable a future "RAS gig economy," where a dense network of modular, interoperable robots from different owners could collaborate, negotiate roles, and trade tasks in a decentralised digital marketplace. To realise this long-term vision, we finally recommend investment in simulation platforms and real-world testbeds to explore the opportunities and risks.

Together, these seven recommendations aim to overcome fragmentation and foster a collaborative ecosystem built on modularity, interoperability, and robust safety and security. This will enable the UK to scale innovation, reduce costs, and become a leader in the responsible deployment of advanced robotics.

Summary of recommendations

Recommendation	Summary
1. Wide-scale analysis of modularity and interoperability.	Establish the extent and scale of the use of modularity in current RAS design, development and deployment. Map and identify applications and systems where interoperability is being used to enhance RAS operation. Curate and disseminate the findings of this work as an open resource for RAS developers.
2. RAS Development Resource	Establish a comprehensive Development Resource that promotes the development of interoperability and modularity in RAS design and deployment. Assess, develop and work towards publishing standards for interoperability and modularity so that designers and developers have a clear reference point. Build a platform structured to be extensible and accessible and used to develop a long lasting ecosystem of best practice in RAS.
3. RAS Design Library	Develop a RAS Design Library offering step-by-step instructions for constructing sophisticated robotic systems using modules and components that adhere to the best practices of the RAS Development Resource. Provide ongoing support for extending the design base, updating, curating and creating new designs. Ensure that the guide is sufficiently well used to change the design practices of the RAS community.
4. Robot Identification Number	Develop a requirement for robots, and/or autonomous sub-systems, to have a unique Robot Identification Number and to share this when requested.
5. Black box module	Develop a requirement specification for 'black box' modules to be applied to RAS technology, to inform standards and regulation bodies and provide industry with the confidence to develop the right technology.
6. Anomaly detection	Develop techniques to standardise the detection of failures, errors and anomalies in real-world RAS systems and their environments and develop a layered set of standardised responses to failure, error and anomaly. Develop tools to support detection and response processes within modules and systems and in post failure analysis.
7. Driving a RAS Gig Economy	Develop techniques and tools to model and explore the opportunities and risks that a RAS gig economy might offer. Conduct virtual simulations and experimental trials in real 'living labs' where robot behaviours can be safely observed.

Contents

1. Opportunities, Gaps and Barriers

1.1 Introduction

The rapid rise of Robotics and Autonomous Systems (RAS) is transforming industries worldwide, offering major opportunities and challenges. Robots now play vital roles in healthcare, energy, automotive, manufacturing, space, defence, and agriculture sectors, to name a few. In 2024, the International Federation of Robotics reported over 4.2 million industrial robots operating in factories globally - an increase of 10% [1], and an annual increase of some 30% in service robot installations. This growth is set to accelerate as industries seek greater efficiency, safety, and automation amid labour shortages. Looking ahead, robots are expected to integrate seamlessly into everyday life, from delivery drones and robotic caregivers to smart farming systems and collaborative robots. With advances in AI, robots will become ever more present in smart cities, education, and homes, making them essential to the fabric of modern society.

RAS have immense potential to drive industrial transformation. However, progress is often constrained by high costs and by system inefficiencies. One significant factor in these high costs and inefficiencies is a lack of openly specified modularity and interoperability in the systems that constitute RAS devices. Most current RAS systems are not modular or easily interoperable, limiting flexibility, scalability and increasing the likelihood of rapid obsolescence.

Adopting well-defined modularity and interoperable systems will lower costs and ease design, development and deployment bottlenecks, thereby raising efficiency both in development and in the supply chain. In addition, improved modularity supports the staged upgrading of installations over time, thereby reducing long term operational costs and enabling a transferrable skills pipeline between industries. Despite these tangible advantages there are multiple barriers to the widespread use of modularity and interoperability that need to be overcome.

The purpose of this paper is to examine how well-specified interoperability and defined modularity can be more widely adopted to create RAS systems that are quicker and cheaper to design, install and maintain. It makes a series of recommendations as to how best practice can be identified and then coordinated across the RAS ecosystem. The ultimate goal is to develop common approaches to the design and development of RAS systems built using a modular approach that can enhance interoperability. If this can be achieved it will usher in a new generation of RAS devices that are more cost effective and quicker to deploy, stimulate a cross-sector supply chain and simplify the operation and maintenance of RAS systems.

1.2 Understanding modularity and interoperability

Both modularity and interoperability are generally well understood within a range of engineering disciplines and operate across different levels. In understanding how to bring them more fully within the RAS ecosystem it is important to understand how they manifest in RAS. In abstract terms they can be described in the following ways:

> ***Modularity at a component level*** is about the architectural division of function between modules that compose a system. Implementing modularity requires agreement on the architecture and on the interfaces between modules, even if this is proprietary and unique to a particular RAS device. Here the architecture and the module interfaces provide a standard approach to design but do not necessarily support extensibility or composability.
>
> ***Modularity at a system level*** allows the interconnection of subassemblies through a standard framework where the subassemblies could come from different vendors. This relies on published and agreed interfaces to modules that support some level of composability, even if this is proprietary, for example only within a vendor based product range.
>
> ***Interoperability at a component level*** is about being able to compose and reuse modules in a variety of ways to make different systems, such that two modules of the same type and interface, but with different internal characteristics, can be seamlessly interchanged. This requires modules to share a common interface and implies that modules can be reconfigured to interact at some level by the exchange of data and information about the module using the module's interface.
>
> ***Interoperability at a system level*** refers to the ability of independently functioning RAS devices to work together to achieve a common function, based on interaction standards which include standards for interaction with human operators and inter-vendor standards. These standards may still be proprietary and there are strong commercial pressures to own market segments by owning the commonly used standard.

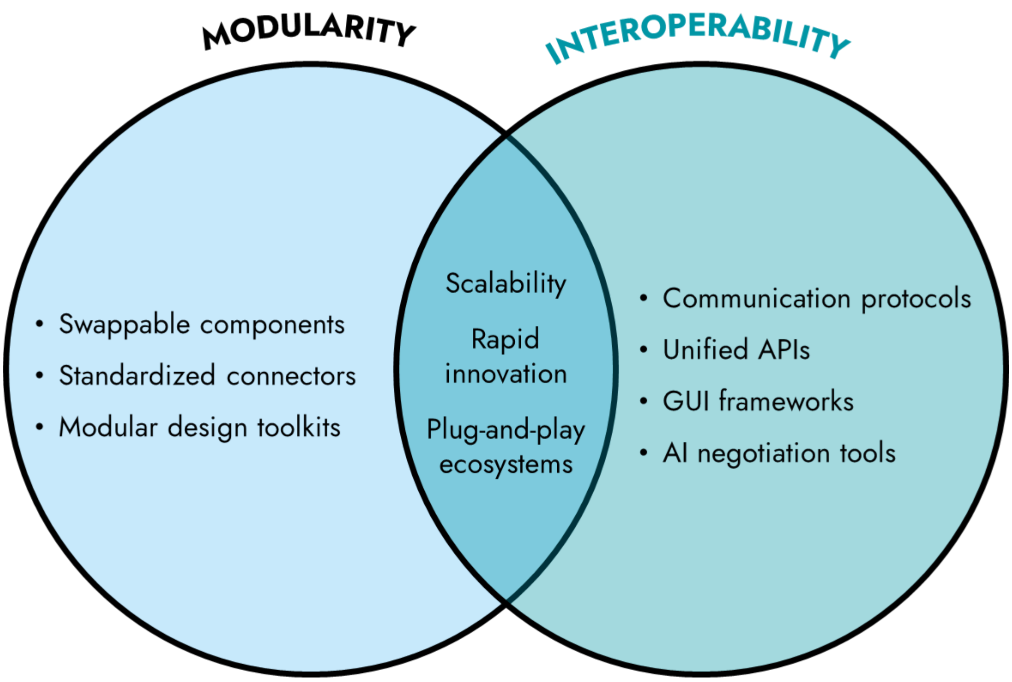
Each of these different forms of modularity and interoperability are critical to RAS. Critical to achieving modularity is the embedding of it as a design methodology, and critical to achieving interoperability is the definition of standard interfaces. These standards can be built in various ways and do not need to be formal. Widespread adoption and clear definitions are more important than formalisation.

Both modularity and interoperability rely on the definition of a common architecture or framework in which the modules operate and a common understanding of how modules and systems connect and interact. These commonalities can be created through conventions, best practice or through formal or informal standards. The nature of these commonalities depends on the function and character of the system or module; software or hardware, electronic or mechanical; design pattern or library etc. Modules may also constitute entire products in their own right that fit within a wider ecosystem that offers wide-scale interoperability.

In between modularity and interoperability is the concept of interchangeability. This applies to hardware and software components and describes the circumstance where one component can be swapped for another without impact on the surrounding system, because the components are functionally identical, even though internally they may be different. For example, AA batteries from different manufacturers are interchangeable, even if their internal chemistry is different. At a systems level, interchangeability means that modules or even whole RAS devices can be swapped for equivalents without the need to reconfigure or recalibrate. Systems that are not interchangeable may still be interoperable if they are able to reconfigure themselves. For example, when you plug a new webcam into your computer, interoperability means that the two communicate automatically and the computer loads the correct driver for that webcam, ensuring that the system continues to operate correctly without action required from the user.

Modularity and interoperability are widely considered, from a design and technical standpoint, to be an advantage in RAS, however after 70 years of robotics development there has been no natural emergence of a modular standard for RAS and support for interoperability is weak. Thus in making a case for modularity and interoperability in RAS, it is important to analyse why they have not been widely adopted to date. We now consider this separately, first for modularity and then for interoperability.

MODULARITY and INTEROPERABILITY



1.3 Opportunities and barriers in modularity

Why Modularity Matters

Component-level modularity is a basic engineering design principle that is used in all forms of engineering. Systems are divided into loosely coupled modules with well defined functions, interfaces and dependencies. When these formalised dependencies are standardised, published and maintained, even internally within a project or company, this can enable interchangeability, extensibility, reusability and maintainability. Modularity speeds development and reduces costs through:

- faster iteration cycles as modules can be swapped or improved independently,
- component reuse across multiple product lines,
- concurrent engineering, where teams develop different modules simultaneously,
- easier identification of faults and bugs through testing of sub-elements in isolation.

In software engineering there are numerous design paradigms that enforce modularity, such as the Universal Modelling Language (UML) that is widely used to design and test complex software systems. In hardware engineering, design tools enable designs to be modularised into a hierarchy of assemblies and sub-assemblies. They also provide standardised components to be used through libraries of parts. These libraries often include both physical and functional information, including 3D models and design parameter lists, that can be used in both design and in simulation; for example modeling the physical shape of an electronic circuit board to ensure fit. While this type of modularity works well at the detailed design stage, the lack of standard architectures in RAS means that modularity is often re-invented with every design.

Systems-level modularity, driven by common standards, enables the composition of systems from interchangeable building blocks. This in turn stimulates supply chains and spurs the growth of a market for components and sub-system, where vendors compete on cost and performance. For example, the power supply for a desktop PC can be swapped for a replacement made by a different manufacturer without reconfiguring the power supply or the PC. As modularity becomes embedded in RAS development, it will reshape the ecosystem, unlocking significant economic, environmental, and technological benefits.

Barriers to Modularity in Practice

To understand why this has not yet happened requires a deeper investigation to fully understand the complex interplay between software, electronics, and mechanical systems in RAS, particularly regarding their functional, behavioural, and structural integration. At an individual company level, design-based modularity is typically well understood as an accelerant to development. However at a system level, modularity that delivers a component or sub-system market for RAS is essentially non-existent; although there are some limited examples of open single-vendor standards that have enabled markets clustered around that specific vendor's products.

Large robot manufacturers have typically protected their systems and created lock-in by using proprietary interfaces which, although modular, do not adhere to a published standard that would allow competitors to replicate and create substitute components. Instead manufacturers have expanded their product range, and service offering, to include the variants that their primary customers need. This is now a well-established business model supported by a "system integrator" market that encapsulates the expertise needed to integrate systems between vendors. As a consequence, integration costs are embedded as a significant part of total installation cost.

For medium-size companies who are manufacturing and selling components or systems, modularity is critical to being able to rapidly develop and deliver bespoke systems. Success is built on a back-catalogue of designs that are continuously being refined and updated. However since this catalogue represents the IP that creates their primary value, it is highly protected. There are exceptions when it comes to Open Source Software. Companies, and small companies in particular, invest in Open Source because it saves them time and allows the community to use their products. Underneath there may be libraries and modules that are not covered by open source agreements but which do have open interface definitions that encapsulate their IP.

In academia, the story is similar. Research teams complete a project and move on, shelving designs rather than building them into future modular frameworks. Research systems are built for proof of concept and demonstration with no intent to scale or any expectation of further use. As a result new students often prefer, or are encouraged, to learn by building from scratch with existing open tools, rather than adopt designs and software that a previous student created.
This is a significant wasted opportunity to train future designers in the benefits of modularity and interoperability.

In addition to these IP and commercial barriers to modularity in RAS, there are also technical barriers. Firstly, the range of physical forms that RAS devices necessarily adopt, because of their "form follows function" nature, makes it harder to develop "one size fits all" standards. A second barrier is the perception that the rate of technology change will outpace the rate of adoption of any particular standard.

Finally there is a cost barrier to system level modularity. RAS systems designed around system level modularity take longer to design in the first instance and only provide a scale advantage in the long term. Unfortunately when products are conceived and costed as one-off systems, or at least as ultra-low-volume systems, mostly made by hand, there is nothing to gain from spending time on making the modularity shareable.

Ironically these barriers are all lowered by adopting modular design processes. Developing novel forms of robots becomes quicker if all the building blocks fit together in a standard way around a well-defined architecture and innovation is accelerated if it can be rapidly integrated into existing standard architectures because extensibility is built in. **Design reuse needs to take priority in the RAS ecosystem.**

Achieving Modularity in Practice

While modularity is widely recognised as valuable, achieving it in RAS is difficult due to the inherent complexity of integrating diverse software, hardware, and mechanical systems across vastly different scales and functional requirements. To overcome this, RAS needs a deeper understanding of common design patterns, reliable off-the-shelf components, and, crucially, well-defined, multi-faceted interface specifications that integrate physical, electrical, and functional requirements.

1.4 Opportunities and barriers in interoperability

Why Interoperability Matters

According to the IEEE Standard 610.12-1990 [2], Glossary of Software Engineering Terminology, interoperability is defined as "the ability of two or more systems or components to exchange information and to use the information that has been exchanged." This definition was later adopted by the National Institute of Standards and Technology (NIST) [3] in the 1990s. The ISO/IEC 2382:2015 standard [4], Information Technology—Vocabulary, similarly defines interoperability as "the capability to communicate, execute programmes, or transfer data among various functional units in a manner that requires the user to have little or no knowledge of the unique characteristics of those units."

To appreciate what interoperability offers beyond modularity, imagine a robot arm made up of sections. If the design is fully *modular*, a section of the arm can be swapped for a new section simply by unplugging and replugging. If the old and new sections are not interchangeable, e.g. the new one is longer, then the arm will need to be reconfigured to operate with the longer section. If there is also full *interoperability* then the arm will be able to self-configure to the new longer section and obey external instructions exactly as before. Provided that the interface between the arm and the section are standardised, then the new longer section could be provided by a different supplier. The expectation is that there is sufficient abstraction so that the rest of the system doesn't need to notice that the arm just got longer.

Interoperability at a systems level

At a systems level, interoperability refers to the ability of RAS devices to work together or to work with other devices in their working environment. So mixed sets of robots from different manufacturers could work together to achieve a task. For example a drone might survey a building site that needs leveling and it is able to instruct the earth mover robot about the terrain.

The earth mover may then instruct a dump truck and digger to remove the earth it has scraped up in the leveling process, with the drone overhead monitoring progress and feeding back information about the ground state. In addition each of these devices can be generating and using external information about the site and the project being undertaken. They may even tap into weather information to shut down if a storm is approaching.

In general terms interoperability enables different devices, systems, components, or software modules to work together seamlessly. The benefits of interoperability include:

- Easier integration of and with third-party systems.
- Greater cross-sector technology and skill exchange.
- Faster innovation through reusable systems and components.
- Lower development and lifecycle costs through improved use, maintenance and reuse.
- Enhanced flexibility and operational agility

Large-scale systems, in particular, require flexible, scalable interoperability to manage complexity and avoid inefficiencies. However in more complex systems with multiple layers of functionality the ability to provide interoperability at each layer becomes more complex.

Tolk et al. [5, 6] address these levels of complexity in their "Levels of Interoperability Model" that defines six stages of interoperability, progressing from no interoperability to full conceptual alignment. Level 1 (Technical) ensures basic connectivity and data exchange, Level 2 (Syntactic) addresses standardised data formats and structures, and Level 3 (Semantic) ensures a shared understanding of the data's meaning. The higher levels include Level 4 (Pragmatic), which considers the context in which data is used. Level 5 (Dynamic) supports understanding systems' behaviours over time, and Level 6 (Conceptual) aligns underlying assumptions, models, and conceptual frameworks between systems. Achieving conceptual interoperability where systems fully understand and correctly interpret exchanged information is the most challenging, remaining a longer-term goal, demanding shared models, deep system-level collaboration, and persistent standardisation efforts.

Barriers to Interoperability in Practice

Despite its advantages, interoperability is still rare in RAS. Developers often build isolated proprietary solutions that create vendor lock-in through proprietary protocols and closed architectures, making cross-platform integration difficult, increasing costs, and slowing innovation.

Key barriers include [5] [6] [7]:

- Easier integration of and with third-party systems.
- Lack of sector-wide standards across RAS disciplines.
- Siloed, non-collaborative development.
- Technical complexity of real-time, cross-platform integration

- Proprietary systems and vendor lock-in.
- Limited uptake by suppliers and buyers.
- High upfront costs and funding constraints.
- Resistance to change from established workflows.

Robots are complex integrations of multiple core technologies and there are few standardised ways of designing, building and installing such systems. Robotic systems typically encounter interoperability challenges due to the use of proprietary platforms, software, and hardware from different manufacturers, resulting in vendor lock-in that restricts integration and limits flexibility. Although frameworks like the Robot Operating System (ROS) promote modular software architectures enabling broad reuse and integration across diverse robotic platforms, its use is still not widespread in industry and its components lack enforced standardisation. While in industrial automation, the OPC Unified Architecture (OPC-UA) [8] provides a standardised means of data exchange, offering a degree of cross-device compatibility and interoperability in manufacturing environments, it has limited uptake outside of industrial automation and can only address the lowest layers of data exchange. In the UK defence sector, the Generic Vehicle Architecture (GVA) provides standardised hardware and software interfaces between vendors, enhancing integration and interoperability in autonomous vehicle systems but is niche to that market. In summary, these frameworks lack the necessary cohesion to pave the way for more open, flexible, and scalable robotic ecosystems. They do provide ideas and elements that can be built on, but do not provide a unified approach.

There is a strong argument that interoperability in robotics fails not simply due to a lack of standards, but because there are no strong economic or technical incentives to align. The core issue isn't the absence of standards—it's that no one is working towards shared integration targets. Most robotics efforts are geared towards short-term outcomes such as demonstrations, pilots, academic publications, or one-off stand-alone products, rather than flexible deployment within a wider ecosystem where interoperability would add enormous value. Vendor lock-in remains a deliberate strategic choice, which makes it even more difficult to create incentives for openness and alignment. The lack of a common reference architecture means that teams are left to reinvent interfaces across mechanical, electrical, and software domains. There is no dominant platform, no mandate from funders, and no supply chain pressure to drive convergence.

Where there are claims of interoperability these often go untested due to the lack of modular, real-world testing infrastructure. Most crucially, there is no supplier and buyer uptake — no procurement incentive, downstream demand, or community validation loop that encourages or rewards interoperable design. As a result, despite significant technical progress, the field continues to be shaped by bespoke, siloed systems, and the promise of true interoperability remains unfulfilled.

These problems of duplication and a proprietary approach to systems design are caused and compounded both by the lack of widely accepted modularity and interoperability standards and a long standing design methodology geared to bespoke approaches. This makes it difficult to ensure compatibility across components and systems and thereby stimulate a cross-sector, cross-application supply chain [7] [9].

There are multiple challenges to the widespread adoption of interoperable design best practice at the scale needed to create economic benefit. Significant challenges include limited uptake by suppliers and buyers, resistance to change, and the need for substantial investment to establish common best practice.

Achieving Interoperability in Practice

Interoperability is driven by a strong functional and commercial need in combination. In sectors where modularity and inter-vendor operability have occurred, for example in the communications sector, there has been an underlying business model driving collective behaviour. This has resulted in global standards such as WiFi, Bluetooth and USB. In industrial automation the PLC has become a standard building block and in computing the business model underlying the ARM processor core has created a global supply chain. In robotics, pressure to achieve better interoperability is most likely to come from major end-users of robotics; for example in the construction industry or in logistics. As RAS use becomes more widespread the push for greater interoperability will grow.

However, the creation of a more connected, efficient, and future-ready robotics ecosystem will require coordinated efforts to overcome adoption barriers, improve standardisation, and invest in scalable, interoperable solutions. We now turn to what these efforts should address.

2. Harnessing Modularity and Interoperability in RAS

The first step to harnessing modularity and interoperability in RAS is to develop a common approach to these two design paradigms that can be readily adopted and used by RAS developers. This needs to start with an extensive analysis of what is already in use and adopted. To proceed it will be necessary to gather sector and robotics expertise together with technology providers able to "blueprint" existing ideas around modularity and interoperability. Such an activity will require a clear industrial remit coupled to academic excellence in the analysis of systems modularity, tools to support its development and an analysis of design complexity in RAS. Academia has a role to play here by supporting the wide scale adoption of design patterns and standards in research and in joint support with end user industry and vendors of demonstrators built and deployed at scale. This work should also identify opportunities for interoperability at component and system level. In particular it should examine common use cases where interoperability, supported by standards, can have a significant functional impact.

Such work is also likely to highlight the need for tools and frameworks that enable modularity at the design and deployment stages and highlight the need for standards both at a module level and in the exchange of information across the levels defined by Tolk et-al [5] [6]. This review of what already exists will seed the second step.

The second step is to communicate and disseminate how modularity can be used in RAS to advantage designers and developers. This involves gathering expertise in the development of RAS systems to create a canon of best practice for modularity and interoperability, building on the investigations in the first step. This canon of expertise should address the lowest levels of modularity and interoperability so that there is clear guidance on best practice for those who are starting out developing their first RAS system. It should identify tools and platforms that can be used as starting points and explore how common RAS functions can be implemented. In this regard there are multiple resources already available that can be channeled to create a comprehensive design guide (for example see [10]).

The third step is to build a community of best practice that shares examples and creates an ongoing momentum to improve RAS through increased modularity and interoperability. This work should provide comprehensive design examples of RAS systems and sub-systems to act as starting points for new designs. These should be of sufficient detail and quality to show how to design, construct, test and approve the design and provide detailed design notes on the choices and directions taken at each design step.

The material generated in each of these steps will be of no value if it fails to be used widely. An integral part of this work is therefore the establishment of a design ecosystem that can uplift the use of modularity and interoperability in RAS. This can start with academic work on robotics both contributing to and drawing from these guides, but requires the engagement of industry to assure quality and relevance that is targeted at promoting reuse.

Each part of this work must also consider how such collections can be assembled, curated, maintained and most importantly validated in the long term. Their value lies in the long term maturation of the content and the maintenance of open access.

In the following sections, we consider each step in more detail.

2.1 Step 1: Modularity and Interoperability Analysis

It is proposed that work is undertaken to investigate the current state of the art in the use of modularity and the current state of interoperability in RAS. This would look at key use cases and sectors where RAS is being deployed and examine the supply chain modularity that currently exists. It should examine current practice in terms of design methodologies for RAS and examine architectures for different kinds of RAS device across different scales and use cases.

The work should then seek to catalogue and curate its findings and identify patterns and common elements, assisted by AI wherever possible. It should assess current design processes and document how modularity and interoperability are treated.

The work should also investigate novel methods of design analysis and decomposition and catalogue where there are existing tools that can assist during design and development.
The work should encompass hardware and software design and consider all the common robot modalities, especially where these alter or constrain the modularity or interoperability of devices. It should also encompass the interoperability of RAS devices with common support systems, such as communications networks, cloud storage and human interface, and review all currently relevant standards that are in use within RAS.

The work should, as a baseline, explore the lower levels in the interoperability hierarchy covering hardware interfaces, software data exchange methods and command and control interfaces. It should also reference both formal and informal standards that are in use. Here it should also perform a gap analysis to identify key areas of development for new standards.

This work will need to examine deployed RAS applications and current RAS systems to analyse interoperability and modularity. It will need to develop a means of categorising and classifying that can be used to curate the gathered information. This may need to work both by sector and by function, for example in healthcare there are many different types of RAS at work with different forms of modularity and different types of interoperability at a component and system level. Capturing and classifying these will allow comparison to similar functions and applications in another sector such as manufacturing. Thereby building a more structured picture of modularity and interoperability across multiple sectors and applications.

The outcome of this work should provide a state of the art analysis of modularity and interoperability in RAS. The findings should be curated and disseminated as a living, open-access resource for RAS developers, for example, through a dedicated portal or a managed GitHub repository, to ensure it is both accessible and continuously updated by the community. This work acts as a precursor to Step 2.

Recommendation 1
Wide-scale analysis of modularity and interoperability.

Establish the extent and scale of the use of modularity in current RAS design, development and deployment. Map and identify applications and systems where interoperability is being used to enhance RAS operation. Curate and disseminate the findings of this work as an open resource for RAS developers.

2.2 Step 2: A Robotics Development Resource

System- and component-level modularity and interoperability in RAS rely on a consistent approach being adopted by researchers, industry professionals, and academic institutions. Consistency in design is crucial for ensuring that modularity and interoperability maximise the benefits of design reuse and trigger a "network effect" that builds trust in a common approach.

The core of Step 2 is the creation of a good-practice resource that can assist developers in creating RAS devices and services. The primary objective is to reduce the level of reinvention and encourage the emergence of a more widely adopted in-common approach to the development and deployment of RAS through improved modularity and interoperability. For example, it could act as a central reference point for interface standards, or guides to choosing sensors or actuators.

This requires the identification of common design patterns and architectural frameworks based on the outcomes of Step 1. From this work it is reasonable to expect that standards will emerge or be adopted from related areas into RAS and this work should support and encourage their emergence and validation. The resource should address the standardisation of module interfaces and address the design approaches needed to create composability and extensibility of these interfaces and the architectures that they fit within. To achieve this the underlying resource framework for curating best practice needs to be well defined and allow for its own extendibility to new and emerging technologies, methods and use cases that will inevitably occur during the lifetime of the resource. It is not unreasonable to expect that the resource has the potential to create a standard in its own right.

The resource should also address the development of interoperability between systems developed by different teams and ultimately between different vendors. Whether this involves standardised software function headers, uniform physical connectors, or even consistent wire colour conventions, aligning on common practices allows diverse components and platforms to work together seamlessly. This cannot be a static endeavour and the resource must build in the ability to review, revise and evolve over time. Such a resource would not only reduce duplicated engineering effort but also build trust among developers by providing validated pathways for modular design.

With respect to standardisation, the resource should operate at a practical level drawing together existing standards and developing or contributing to the development of new ones. In RAS, standardisation needs to address interoperability at all levels, component and system modularity, reference architectures, functional performance, human factors, safety, security and interchangeability. The resource needs to provide real world "good practice" guidance to define "good design" that can speed up the development of RAS devices and the deployment of systems within a framework of well defined standards, either formal or informal. Importantly, standards could evolve alongside the resource, eventually forming a *de facto* specification for composable RAS architecture.

To provide a strategic framework for this, Standards Development Organisations (SDOs) should work together to develop a RAS modularity and interoperability maturity model. This model, hosted and maintained within the Development Resource, would help developers, vendors, and end users assess where they stand in terms of design reuse, module interchangeability, and system composability. It would also map current and in-progress standards across SDOs such as BSI, ISO, IEEE, ASTM, and others to a common development roadmap, highlighting gaps and helping coordinate future work. By providing a strategic framework, the maturity model would encourage convergence and benchmarking, supporting wider ecosystem alignment.

The goal is to produce an evolving body of work that captures and spreads the best practice of interoperability and modularity for RAS that can be adopted globally. While this may start out at the lower levels there has to be an expectation that it will, in the longer term, progress to establishing modularity of function at the higher levels of interoperability. There are parallels here with the functions of the WWW consortium (W3C) [11] that oversees web standards, and with the curation processes used by Linux to develop and manage software compatibility.

Recommendation 2
RAS Development Resource

Establish a comprehensive Development Resource that promotes the development of interoperability and modularity in RAS design and deployment. It should assess, develop and work towards publishing standards for interoperability and modularity so that designers and developers have a clear reference point. The resource should be built into a platform structured to be extensible and accessible and used to develop a long lasting ecosystem of best practice in RAS. This work builds on the survey outcomes of Recommendation 1 and seeks to capture, document and curate all elements of good practice in the design of RAS systems.

2.3 Step 3: A RAS Design Library

As robotics research advances at pace, there is a growing need for tools and resources that can streamline development and remove barriers to innovation. One of the most significant challenges facing researchers today is the duplication of effort in constructing robotic platforms that are the foundations of further research. Too often, valuable time and resources are consumed by building bespoke systems from the ground up; efforts that could be more effectively directed toward exploring new concepts, testing algorithms, or developing innovative applications. To overcome this issue, there is a clear need for a practical, standardised resource that offers reliable starting points for complex robotic builds. Initiatives such as robot-manipulation.org [12] and Partcad [13] are excellent steps in the right direction, but there is further to go to build a comprehensive resource. An initial task in constructing this is to identify and collect all the relevant design resources that already exist into one place. This will highlight gaps and create opportunities to unify existing resources.

The proposed RAS Design Library then aims to fill these gaps. It would serve as a detailed, accessible guide designed to support RAS designers and developers. The library would document the design and construction of a range of robot devices in sufficient detail to allow the reader to replicate the design or utilise parts of the design in another design. These designs would act as illustrations of the development guidelines in Step 2 and would exemplify the modularity and interoperability approaches that have been identified as best practice.

The library would offer step-by-step instructions for constructing sophisticated robotic systems in a range of modalities and use cases with the aim of providing a comprehensive guide. Where appropriate, this should also reference the relevant commercial supply chain for parts and modules. The goal is to reduce the time wasted by developers spent reinventing modules and designs that already exist at a sufficient level for their needs. A standardised approach to modularity and interoperability, expressed across designs in the library, should seed the RAS ecosystem with a new approach to design and development. This approach would ensure consistency and reproducibility, essential qualities in both academic and industrial settings, encouraging collaboration through common frameworks.

For the library to be adopted by the community at large, both research and industrial designs must be high quality and comprehensively tested. Designs would need to be peer-reviewed, built, tested and if necessary put through a basic certification process to assure potential users that they can be made CE/UKCA compliant before inclusion in the library. For example, ROS is not widely adopted in industry because it is not certified. This is vital to improving industry academia collaboration and the acceleration of design sharing between research and industry.

Responsibility for developing this library should lie with a collaborative group drawn from leading robotics research institutions, academia, industry partners, and standardisation bodies. This consortium would ensure technical rigour and broad applicability and keep the content current and reflective of emerging best practices. While this is a substantial undertaking, the potential benefit of more rapid design and development progress across the spectrum of RAS deployment is seen as significant. However this benefit will only accrue if the design library is utilised, maintained, extended and disseminated. It is therefore important to couple the development of the library with a programme of dissemination and adoption processes that promote widespread engagement with the designs particularly in an industrial context. For this propagation to be effective, all designs must be provided via an open hardware license allowing commercial reuse and modification.

Recommendation 3
RAS Design Library

Develop a RAS Design Library that serves as a detailed, accessible resource to support world-class robotics research and development. Designs would offer step-by-step instructions for constructing sophisticated robotic systems using modules and components that adhere to the best practices established through Recommendation 2 utilising the existing resources identified in Recommendation 1. This work should also encompass the need to provide ongoing support for extending the design base, updating, curating and creating new designs as well as undertaking activities to ensure that the guide is sufficiently well used to change the current design practices of the RAS community and thus to speed up design, development and deployment.

2.4. Content examples for the Development Resource and Design Library

In order to ground the above recommendations, it is important to illustrate the types of work that could be undertaken against the actions proposed by each recommendation. The following four sections act as examples of the type of work that could be included in the Development Resource and the Design Library. These represent detailed issues raised by the workshop undertaken as a part of the work of the committee. These descriptions are of work that could be undertaken today and so are rooted in the "here and now". The resource and library must evolve over a longer time span so these are snapshot illustrations of the sort of action that could be undertaken. Ultimately each of these suggestions will need to fit within a larger framework of knowledge capture and curation.

2.4.1 Standardising inter-module and device communication

One of the main areas of standardisation that underpins modularity is the standardisation of interfaces, both mechanical, electronic and data between modules and between devices. As such, developing design and implementation best practice around these interfaces is likely to form a major part of the Development Resource and they will be widely used in the Design Library.

In today's robotics landscape, when two robots meet, their primary communication protocols may not be compatible and they typically can't communicate out of the box. Their data formats, content schemas, and communication protocols often differ—sometimes subtly, sometimes fundamentally. Engineers must manually rewrite code, translate messages, or build middleware bridges just to make basic interoperability possible. This not only slows down deployment but also constrains real-world collaboration, especially in dynamic, high-stakes environments like disaster response, defence, or manufacturing.

To fix this, first, individual robots must use a recognised standard of communication. There are good examples of communities coming together to create such standards, e.g. the SiLA standards [14] used in laboratory automation. Where there is a strong commercial incentive from end users to lower the cost of integration and deployment, there will be pressure on RAS manufacturers to conform to standard communication protocols.

There is therefore a strong need to provide a well-defined low-level data-exchange mechanism appropriate to multiple real time heterogeneous RAS devices with various connection strategies and methods. This base layer needs to be extensible and adaptable while maintaining backward compatibility. This is especially critical as the lifetime of some RAS devices is likely to be long, especially in manufacturing or in safety critical environments where devices may be out of reach; for example in space or in nuclear environments. In manufacturing there is a healthy second-hand market for older robot arms that are still fully functional. Critical to this market, which supports circularity, is the continued ability to integrate older devices.

The layers of module communication are highly likely to align with a RAS based version of the "Tolk layers" [5] [6] for interoperability. In this case only the lower layers can currently be defined however, provided extensibility is encoded sufficiently well, the later layers can be outlined and defined over time. These specifications should then form the basis of module interactions where modules are interoperable and/or interexchangeable.

Second, robots must be able to communicate even if they are using different standards. Here it is possible to imagine protocol negotiations at higher layers to accommodate different RAS modalities. This could involve the development of an AI-based negotiation toolkit for communication across diverse robotic systems, regardless of hardware or software differences. Rather than relying on rigid, pre-defined interfaces, robotic agents could use this system to interpret metadata, reconcile schema mismatches, and generate on-the-fly compatibility layers. This would dramatically reduce the time and complexity involved in multi-robot system integration, making deployments more scalable and robust.

Imagine, for instance, a drone from Company A and a ground robot from Company B rapidly deployed to assist in a disaster zone. Today, differences in communication protocols would require engineers to manually intervene. With a standardised approach to capability discovery and data interpretation, the two robots could autonomously negotiate a shared communication protocol, enabling seamless collaboration. For example one device might need to transfer an object that it cannot handle to another location. Interactions that request assistance from other devices can refer to the generic need without having to communicate the skill required. The drone could transmit aerial imagery and GPS coordinates, while the ground robot navigates to target locations and reports back status updates—all without external coordination.

This long-term vision builds directly on today's needs and gaps. It complements efforts around reference interfaces, standards, and testing infrastructures, while offering a fallback solution when systems don't align out of the box. By enabling robotic platforms to "speak" to one another on demand, a standardised approach lays the groundwork for a truly interoperable and collaborative robotics ecosystem.

This vision builds on, rather than bypasses, near-term progress: it complements the development of reference architectures, standardised data models, and protocol testbeds, by enabling intelligent fallback mechanisms where those standards are absent or mismatched.

To be viable in safety-critical and real-world settings, the proposed communication toolkit must integrate robust mechanisms for safety, security, and traceability, discussed in Section 4 below. Rather than replacing standardisation efforts, this approach enhances adaptability and resilience, serving as a dynamic layer of interoperability that bridges today's fragmented systems and tomorrow's modular, collaborative robotic ecosystems.

This is a long-term ambition—but it's one that emerges naturally from today's pain points. The proposed AI-based negotiation toolkit not only aims to streamline current multi-robot collaboration but also intends to create a standard framework for future robotic ecosystems. By continuously adapting to technological advancements, this toolkit can facilitate interoperability across diverse robotic systems and become a universal solution for collaborative robotics, industrial automation, and autonomous systems. This adaptability will empower robots to function effectively in a rapidly evolving technological landscape, ultimately driving more efficient, scalable, and autonomous robotic operations across industries.

2.4.2 Standardised conventions for robot/human interaction

Familiarity with technology is built on good user interfaces, the ubiquity of "Buttons", "Menus" and objects such as pull-down lists means that we can quickly navigate around unfamiliar applications on computers and on the web. No such standardisation exists in RAS; devices with exactly the same function offer different and incompatible user experiences. The Design Guide should set out what "good design" looks like, feels like and what behaviour RAS devices should exhibit, especially in public spaces.

A simple example is that many robots use lights to communicate information to the user, but there is currently no standardisation about what these mean. In robot vacuum cleaners, for example, on the Eufy 11S [15], an orange light means that the unit is returning to dock, whereas on the Roborock S7 [16], an orange light means an alert and docking is indicated with a green light. Coming up with an agreed convention covering standard situations like "running normally", "under tele-operation", "low charge", "malfunction" etc would be extremely helpful.

Users and developers often interact with robots through a graphical user interface (GUI) that provides information on robot performance and internal state while providing access to control and task parameters in order to direct and alter the behaviour of RAS systems. This has particular relevance to tele-operated and tele-monitored systems and to the interaction developers need when integrating RAS devices into an operating environment and when proving functionality.

Although the wide variation in robot modalities and architectures may preclude a single robot GUI, there is still considerable scope for standardisation. A standard convention for displaying and setting parameters and other information about the robot would reduce the training needed to interact with robots and provide a common framework for interaction. This should not be limited to graphical interfaces but extend to, for example, haptics and hand controllers of different forms.

The development of standardised guidelines/conventions for robotics will enhance the user experience, reduce the learning curve and facilitate system integration and robot deployment. Aligning the UI design with key standards such as ISO 11064 (ergonomic design of control centres) [17], ISO 9241 (human-system interaction) [18], and ISA 101 (human-machine interfaces for process automation) [19], along with widely accepted principles of accessibility and usability, will enhance usability and improve safety.

Key features of robot APIs, such as sensor data visualisation, real-time status updates, and remote control capabilities, should be integrated seamlessly into the UI. This will allow operators to engage with robots through visual representations, making complex processes more manageable and reducing reliance on command-line interfaces. A significant feature of the UI will include standardised error handling and debugging tools. These features will enable users to quickly identify and resolve issues, reducing downtime and improving operational efficiency.

Ideally the UI can be configured for use by people with sector or use-case oriented skills rather than robotics experts. It should provide the capability to interrogate decision making and communicate issues by providing real-time feedback to simplify troubleshooting, even for users with limited technical knowledge.

2.4.3 A Unified Robot API

An underlying fundamental to modularity and interoperability is a unified API. These can be found in all communication devices, in plug-and-play peripherals and in emerging standards for smart electricity devices such as heat pumps, electric car chargers and battery storage systems [20]. The Development Resource should provide a platform for cataloguing and developing APIs.

A significant barrier to seamless integration and innovation in RAS is the lack of standardised application programming interfaces (APIs). Robot manufacturers typically provide a unique API tailored to their specific platform, leading to a fragmented ecosystem. This diversity in APIs complicates system integration, increases development time, and creates unnecessary obstacles for developers who wish to work across different robotic systems. As a result, integration efforts often require significant customisation, hindering scalability and the rapid evolution of robotic technologies.

Despite the existence of widely used middleware such as ROS, there is no recognised standard for APIs. If interoperability and module exchange and substitution are to scale, we will require standards for interoperability that encompass a wide range of implementation paradigms. While it may be a complex and difficult task there is merit in developing a unified approach to providing RAS APIs with the goal of providing a unified API at one level to enable greater hardware interoperability.

To address this challenge, we recommend the development of a standardised robot API that facilitates easier interoperability and accelerates the integration process across various robotic platforms. This effort could begin with defining APIs for common, self-contained components. For instance, the community could develop and validate standard APIs for lidar, force-torque sensors, or simple grippers, ensuring a standard format for naming and a common approach to semantics. Success with these foundational components would build momentum and provide the building blocks for standardising more complex systems, creating a clear and practical pathway toward a more unified, top-level robot API. A unified API would provide a consistent interface for controlling and interacting with robots, regardless of the manufacturer or underlying technology, reducing complexity and streamlining development workflows.

The proposed unified robot API should support multiple widely used programming languages in the robotics, automation, and AI fields. Python, C++, JavaScript, and Java are key languages that should be incorporated into the standard API. Python is ideal for AI and machine learning due to its simplicity and strong libraries. C++ supports real-time control and high-performance robotics. JavaScript enables web-based and cloud automation, while Java offers platform-independent, robust integration for real-time robotics control.

By incorporating support for these languages, the unified API would cater to a broad range of use cases, from AI-powered robotics and advanced machine learning applications to real-time control systems and web/cloud-based automation. This combination of programming languages ensures the API would be accessible to a diverse set of developers and integrators, fostering widespread adoption and innovation across the robotics community.

2.4.4 Standardised Tactile Data Representation

RAS devices generate significant amounts of data from their various hardware systems; both sensing and actuation require new standards to be developed to enable interoperability and create device independence. The Development Resource should provide a home for the development of data standards that relate to the different types of data that are unique to RAS while the Design Library should provide examples of how they are used and designed into systems.

An important part of delivering module and interoperability standards is the creation of standards that relate to data types that are commonly encountered in RAS, just as there are data standards for common data types in communications. One example area is in the formatting of tactile data.

Tactile sensing is critical for enabling the next generation of dexterous manipulation and safe physical interaction in robotic systems. However, despite advances in tactile sensor technologies, there remains a significant gap in the standardisation of data representations for tactile interactions. Current efforts, such as ROS-based tactile/wrench messages, and some frameworks like GelSight [21], Skinware [22], and OpenTouch [23], have contributed valuable developments but remain limited in scope and interoperability.

The absence of a universally accepted file or transmission format hinders cross-platform integration, restricts the reusability of tactile data for research and simulation, and complicates the development of generalised algorithms for tactile perception and control.To address these challenges, we propose the creation of an open, standardised file and transmission format for tactile data. The format should be sensor-agnostic and morphology-agnostic, accommodating diverse sensing technologies and surface geometries. It would define consistent structures for contact events, force distributions, temporal sequences, and metadata, while supporting extensibility for new modalities. Such a standard would foster interoperability, promote data sharing, and support multi-system integration in tactile robotics across academic research and industrial applications.

3. The Criticality of Safety and Security

Why Safety and Security Matter

Robotic systems introduce new risks to the environments in which they operate. Perils include the direct risk of injury to humans and critical infrastructure. They also increase the potential impact from cybersecurity breaches by bringing hacking into the physical world. For autonomous systems to achieve mainstream acceptance, it will be important for modular, interoperable systems to adhere to levels of safety and security appropriate to the operating environment. Safety and security must be treated as core design, development and deployment priorities for these systems to gain public trust and reach commercial viability. Public acceptability and trust are critical to wide scale deployment. Because this is so crucial, we make a further 3 recommendations for future work in this area.

Barriers to Safety and Security in Practice

For RAS to safely and successfully integrate into public and private domains, interoperable and modular systems must evolve alongside emerging cybersecurity and safety standards, and frameworks should be designed to manage risk effectively without constraining innovation. Over the coming years, it is both necessary and inevitable that a combination of formal (*de jure*) and informal (*de facto*) standards coupled to regulation will emerge to govern key aspects of public RAS deployment. These standards will need to address areas such as implementing cryptographic security and secure communication protocols, reinforcing internal threat mitigation at the level of each individual robot, and developing external threat detection systems that monitor environmental signals.

Establishing a safe and secure RAS ecosystem will hinge on a clear set of enforceable principles that promote reliability, accountability, and public confidence. Systems must be underpinned by explainable decision-making frameworks prioritising human safety, particularly in failure scenarios.

To address these challenges, we propose three innovations we think will be needed to ensure the safety and security of interoperable RAS systems:

- A unique **robot identification number** for each autonomous unit
- Standardised **black-box modules** to provide clear, accessible records of operational history for use in the independent investigation of incidents and to standardise the recording of operational data to enhance safety, security and reliability.
- **Anomaly detection systems** that are able to assess the status of a robot and its environment and communicate faults, errors and anomalies in behaviour and the environment.

Together, these components represent foundational building blocks in creating safer, more reliable autonomous systems that scale responsibly across industries and environments. As safety and security standards rise, and robots are increasingly used without supervision or in hazardous or critical environments these technologies will provide guarantees for manufacturers and users.

3.1 Robot Identification Number for Traceability

Each robot or module should have a unique identity, similar to the IMEI number of a mobile communication device, or the vehicle identification number [24] on every car. This would enable the provenance and ownership of robotic systems to be reliably verified, regardless of where or how they are deployed. Such a capability is particularly valuable in supply chain contexts, modular robotics, or environments involving multiple stakeholders. Robots would share their identification numbers during encounters with other robots. The system would support traceability across complex deployments while guarding against identity spoofing or unauthorised access. If in the future robots incur national usage charges, for example a system of usage based charges or taxation, the robot identification number could be used to identify the owner and therefore make charges appropriately.

> ***Recommendation 4***
> ***Robot Identification Number***
>
> Develop a requirement for robots, and/or autonomous sub-systems, to have a unique Robot Identification Number and to share this when requested.

3.2 Black Box for Robotic Transparency and Accountability

As robots become increasingly embedded in critical infrastructure, logistics, healthcare, and public spaces, there is a growing need for transparent and verifiable records of their actions, decisions, and operational history. To address this, we propose a robotic "black box", similar to an aircraft's flight recorder but adapted for the unique challenges of autonomous robotic systems.

This secure module would record operational data while providing an auditable digital identity anchored in a secure and reliable framework. It would continuously capture key metrics such as movement, task execution, sensor inputs, decision logs, and system states. If data were captured directly from the main robot control system, manufacturers might argue that it reveals their trade secrets, and thus that disclosure to an investigator or in a court would breach their rights to protect their inventions. To avoid this, the black box module may need to capture specific auxiliary information from the main controller via a well defined interface.

It would also capture key actions of other robots, logged with their robot identification number. This ongoing log would be stored securely, using the latest techniques [25] to ensure the data remains immutable and tamper-evident.

The auditable data within the black box would enable detailed analysis of any untoward incidents involving robots. Effectively, it provides a way of asking the robot what happened and ensuring an accurate and helpful answer. A black box would provide investigators with essential data and the evidence to make recommendations following a technical fault, a security breach, an operational failure or simply a complaint. This could support regulatory compliance, simplify investigations, and accelerate iterative improvements in autonomous performance. Integrating secure and reliable digital technologies may offer a decentralised mechanism for ensuring integrity and transparency.

By coupling this with onboard logging and identity systems, the robotic black box could play a foundational role in the future of autonomous system governance, enhancing safety, security and accountability across the field.

Recommendation 5
Black box module

Develop a requirement specification for 'black box' modules to be applied to RAS technology, to inform standards and regulation bodies and provide industry with the confidence to develop the right technology. This will use the Robot Identification Number developed in Recommendation 4.

3.3 RAS Anomaly Detection in the Real World

Detecting when a RAS technology deviates from expected norms and behaviours is an important and useful capability, particularly when those deviations may stem from a wide range of causes. Hardware faults, software errors, and even malicious interference can all lead to unpredictable outcomes, and distinguishing between these factors is complex. Such errors may be identified by external sensor measurements or communicated directly from the faulty system itself.

In any moderately complex system there will be an internal fault diagnosis and response system able to detect hardware failures such as a motor not working, or a sensor producing anomalous outputs. The system may be complex enough to have sufficient redundancy to allow it to continue working, or it may need to fail safely. Similarly software, particularly realtime software that controls hardware, has a built-in failsafe system based on the design paradigm of "make it work but assume it will fail". These software systems may also be able to correct by restarting processes or in the last resort by rebooting. Software also needs to be resilient to malformed commands, data synchronisation errors and interrupt storms.

Where systems are composed of modular units, errors and failures often occur through misalignment in specifications between modules and in unforeseen gaps in specifications. Similarly failures in interoperability can arise, particularly at the higher levels where unique combinations of modules or devices are in use. While many such mismatches and failures will be ironed out in the development and deployment process they are likely to also manifest in operation.

Where the behaviour of a robot is simple and repetitive it is possible to detect abnormal motion either internally or externally, either by automatic monitoring via safety systems external to the robot or by anomalies in the internal data streams; for example the position reported for a motor being at odds with the current control position, or by the detection of rapid changes of position. However as robots operate in more complex environments and with more complex functions, more advanced anomaly detection is required. As well as detecting their own anomalous behaviours, robots should also detect and report suspicious behaviour from other robots (e.g. failure to respect priority protocols). This ability could be an important defence against malicious deployment.

Reliable anomaly detection will be extremely challenging especially given the need for both false positive and false negative rates to be extremely low. With potentially millions of robots deployed in the future, even a tiny false positive rate would result in an unacceptably high level of false positive reports, overwhelming any oversight system and critically undermining public trust. Conversely, even a single false negative, where a genuine fault goes undetected, could be disastrous.

In the industrial manufacturing sector, the ESPRC-funded Circular Robot 5.0 R&D project [26] will make a start on these challenges by harnessing AI-powered predictive maintenance, blockchain-based traceability, and comprehensive life-cycle assessments, aiming to extend the operational life of industrial robots.

Where large language models (LLMs) are used to interpret data, issue instructions, or drive decision-making the issues become even more complex. Processes and techniques are needed to determine whether they produce reliable and contextually appropriate outputs during operation and assess whether they could be manipulated to generate unsafe or misleading commands. Indeed, arguably data-driven AI models such as LLMs cannot be used in safety-critical control tasks unless it is possible to make guarantees about their behaviour [27].

The ultimate goal is to create practical tools and methodologies for detecting and diagnosing anomalous behaviour and failures in the environment. It is also important to have a standard architecture for detecting and reporting internal failures. Because of the physical nature of robots and their interaction with the real world, it is also necessary to develop a standard hierarchy of responses to failures. This is particularly important in safety-critical environments.

Standardised approaches to failure, error and anomaly detection and a standard set of responses to different levels and kinds of failure are critical to both system and module design and constitute an essential part of the interoperability "catalogue" of functions. Developing this capability as a standard design pattern will help engineers and developers design more robust, secure and reliable systems capable of maintaining safe operation, even with unforeseen challenges or adversarial conditions.

Recommendation 6
Anomaly detection

Develop techniques to standardise the detection of failures, errors and anomalies in real-world RAS systems and their environments and develop a layered set of standardised responses to failure, error and anomaly. Develop tools to support detection and response processes within modules and systems and in post failure analysis using the black box developed in Recommendation 5. Such work is critical to establishing robust interoperability and in the certification of RAS in complex, hazardous and safety critical environments.

4. A Future Gig Economy for Interoperable and Modular RAS

Many in the RAS community consider that modularity and interoperability will be transformative, that including these design paradigms in all of RAS will unlock new functionality and opportunity in the delivery of services. As part of a foresight process, it is valuable to consider how this might look in the future.

Why a RAS gig economy matters

Imagine a world where robots do not just work in isolation, but collaborate. In this vision, robots from different manufacturers and sectors — for example, delivery bots, cleaning units, industrial arms, and mobile surveyors — can communicate, adapt, and collaborate without bespoke integrations. Much as USB transformed personal computing, interoperable robotics will enable plug-and-play compatibility across systems. Robots will identify themselves, negotiate roles, and adjust behaviours in real-time based on shared protocols and mission goals.

A dense network of modular robots working together also redefines operational models. In this world, robots not only perform tasks but also trade them. Picture a delivery robot weaving through traffic when it suddenly receives a new request: a package needs to be picked up a few blocks away, but the robot is already en route to another job. Instead of ignoring the task, it posts it to a digital marketplace accessible by all nearby robots. Within milliseconds, two other robots—one closer and the other more efficient at navigating dense areas—see the task, assess their own availability, and submit bids. The task is awarded to the most cost-effective option, and the original robot continues on its way. No human stepped in. No dispatcher issued orders. It all happened peer-to-peer, powered by digital tokens that serve as a kind of robotic currency.

The real innovation lies in what this decentralisation unlocks. Instead of central systems assigning jobs based on fixed rules, the market optimises everything dynamically. Specialised robots (e.g. one excels at navigating stairs or another at handling fragile items) are naturally selected for the jobs they are best at. If one robot breaks down, others can redistribute its tasks in real-time. If a rare event causes a spike in demand, prices adjust automatically, ensuring the most critical tasks get priority.

Barriers to a RAS Gig economy

This vision requires shared frameworks, safe and secure standards, and secure communications. Encryption, digital identities, and policy-based access controls must be baked into the infrastructure to ensure safe, secure and reliable robot-to-robot interaction. Interoperability at scale alters the financial, environmental and sustainability operation models for these multi-function RAS.

Still, such a vision is not without complications. For one, how do robots trust each other? In a world where machines exchange tasks and payments without central oversight, mechanisms for trust and verification [25] become essential. Smart contracts (self-executing agreements written into code) could help. For example, a robot might only release payment once it has verified that the job was completed properly. Logs, location data, and sensor records can serve as proof of work, and independent auditing bots or decentralised review systems could resolve any disputes.

There are also broader questions. What happens if a robot manipulates market prices for its own benefit? For example, could powerful owners flood areas with their robots, undercutting prices and pushing others out? Just as in human economies, rules and regulations must be established (only in this case, the "laws" will likely be enforced algorithmically).

Ethically, there are also challenges. To what extent should humans intervene if robots are making autonomous decisions about work, cost, and collaboration? If a robot refuses a task due to low payment, who is to say it is wrong? These are more than just philosophical puzzles, they are practical concerns in designing systems that are fair, accountable, and aligned with human goals.

Such systems also pose broader socio-economic questions. In future societies where the working-age population is an unprecedentedly small fraction of the whole, the increased use of robots in this way may allow the economy to flourish. But without a universal means of wealth distribution from the work of these robots, society may become even more divided. How we use and deploy the technology we create in the face of the changes we expect in the decades ahead will shape society and the nature of work.

4.1 Modularity, Interoperability, Safety and Security as Foundations of the Robot Gig Economy

To explore this future, we propose undertaking detailed simulations, e.g. of virtual cities filled with autonomous robot agents that perform real-world tasks under various economic rules. These environments will test different bidding strategies, secure and reliable protocols, and ownership models. Over time, the most promising designs can transition into 'living lab' environments and then into real-world trials using physical robots in controlled environments: warehouses, smart buildings, and perhaps even neighbourhoods.

The vision of a densely networked, decentralised ecosystem of autonomous robots—trading tasks, managing resources, and collaborating without central control – rests on the foundational pillars of modularity, interoperability and security.

For example, robots cannot be easily adapted, upgraded, or repurposed without modular design to meet changing needs across varied environments. Modularity ensures that mechanical, electronic, or software components can be swapped or improved without rebuilding entire systems. It makes flexibility affordable and scalability achievable.

Interoperability, meanwhile, is what allows these robots to interact within a shared digital economy. It enables real-time communication, task negotiation, and seamless integration across manufacturers, platforms, and owners.

Modularity and interoperability are only achievable when there is safety, security, and reliability that interconnected and interacting elements will behave as intended; if they do not, there is a clear mechanism for accountability.

Recommendation 7

Driving a RAS Gig Economy

Develop techniques and tools to model and explore the opportunities and risks that a 'Gig economy for RAS' might offer. Research, development and innovation should include virtual simulations and experimental trials in real 'living labs' where robot behaviours can be safely observed.

5. Conclusions

This position paper presents a set of practical recommendations for realising the transformative potential of RAS, highlighting the foundational roles of modularity, interoperability, safety, standards and security. Despite notable technological progress, development remains constrained by fragmented system architectures, proprietary platforms, and the absence of consistent standards and enabling regulatory frameworks. Rapidly tackling these systemic challenges is essential to unlock scalable innovation, reduce costs, and allow the safe and effective deployment of **RAS** across various sectors.

Following a national engagement programme, including surveys and a stakeholder workshop, the team has identified a clear path forward: innovation-driven frameworks, community-driven resources, and targeted interventions that can accelerate progress. The proposed **RAS Development Resource** and **RAS Design Library** aim to establish shared engineering practices and reduce duplication of effort. Interoperability will be advanced by developing standardised interfaces, AI-infused communication tools, and standardised data representations. Crucially, these technical enablers must be paired with robust safety and security systems, including black box modules and real-time anomaly detection, to build public trust and resilience.

The idea of a **decentralised RAS gig economy** presents a vision of autonomous collaboration among modular and interoperable robots—exchanging tasks, adapting in real-time, and functioning through shared protocols. Realising this vision will depend on coordinated investment in simulation platforms, real-world testbeds, and policy frameworks that support innovation while ensuring accountability.

This position paper presents seven recommendations, summarised on page 7. The UK has an opportunity to contribute meaningfully, drawing on its research strengths and innovation capacity, but progress in this area will ultimately depend on global coordination and shared ambition.

References

1. World Robotics 2024 Report, International Federation of Robotics, Frankfurt, September 2024.

2. 610.12-1990 - IEEE Standard Glossary of Software Engineering Terminology [Online]: https://ieeexplore.ieee.org/document/159342/definitions#definitions [Accessed: 4/8/25]

3. National Institute of Standards and Technology (NIST) [Online]: https://www.nist.gov/ [Accessed: 4/8/25]

4. ISO/IEC 2382:2015 - Information technology — Vocabulary [Online]: https://www.iso.org/standard/63598.html [Accessed: 4/8/25]

5. A. Tolk and J. A. Muguira, "The Levels of Conceptual Interoperability Model," presented at the Fall Simulation Interoperability Workshop, Orlando, FL, USA, Sep. 2003.

6. A. Tolk, C. D. Turnitsa, and S. Y. Diallo, "Ontological Implications of the Levels of Conceptual Interoperability Model," presented at the World Multi-Conference on Systemics, Cybernetics and Informatics (WMSCI), Orlando, FL, USA, Jul. 16–19, 2006.

7. Smart Machines Strategy 2035, The Robotics Growth Partnership / DSIT, February 2025

8. OPC Foundation - The Industrial Interoperability Standard [Online]: https://opcfoundation.org/ [Accessed: 4/8/25]

9. UK Cross-Sector RAS Development Task Force: National Collaboration Transforming the Route to Market [Online]: https://uk-ras.org.uk/publications/white-papers/uk-cross-sector-ras-development-task-force/ [Accessed: 4/8/25]

10. Robot Kinematics and Dynamics [Online]: https://u0011821.pages.gitlab.kuleuven.be/pdf/2009-HermanBruyninckx-robotics-textbook.pdf [Accessed: 4/8/25]

11. World Wide Web Consortium [Online]: https://www.w3.org/ [Accessed: 4/8/25]

12. Robot-manipulation.org [Online]. https://www.robot-manipulation.org/hardware [Accessed: 6/5/25]

13. Partcad [Online]. https://partcad.org/ [Accessed: 6/5/25]

14. Standards in Lab Automation (SiLA) [Online]: https://sila-standard.com/ [Accessed: 4/8/25]

15. Eufy 11S - RoboVac Owner's Manual [Online]: https://salesforce-knowledge-download.s3.us-west-2.amazonaws.com/000012766/en_US/000012766.pdf [Accessed: 4/8/25]

16. Roborock S7 Robotic Vacuum Cleaner User Manual [Online]: https://gzhls.at/blob/ldb/6/2/2/1/06cb3efd778d3755764595fe568312113c2c.pdf [Accessed: 4/8/25]
17. ISO, "ISO 11064-4:2000, Ergonomic design of control centres - Part 4: Displays and controls," International Organisation for Standardisation, 2000.

18. ISO, "ISO 9241-171:2008, Ergonomics of human-system interaction — Part 171: Guidance on software accessibility," International Organisation for Standardisation, 2008.

19. ISA, "ISA-101.01-2015, Human-Machine Interfaces for Process Automation Systems," International Society of Automation, 2015.

20. New Smart Appliance Standards Empower UK Consumers to Cut Energy Bills and Embrace Flexibility [Online]: https://www.techuk.org/resource/new-smart-appliance-standards-empower-uk-consumers-to-cut-energy-bills-and-embrace-flexibility.html [Accessed: 4/8/25]

21. GelSight Inc., "DIGIT Tactile Sensor," [Online]. Available: https://www.gelsight.com/products/. [Accessed: 30/4/25].

22. Skinware, "Middleware for tactile information acquiring and processing," *Software Development*, [Online]. Available: https://www.cyskin.com/software-development/. [Accessed: 30/4/25].

23. LASR Lab, "OpenTouch interface," *GitHub*, [Online]. Available: https://github.com/lasr-lab/opentouch-interface [Accessed: 30/4/25]

24. Vehicle Identification Number (VIN) [Online]: https://www.rac.co.uk/drive/advice/buying-and-selling-guides/vin-number/ [Accessed: 12/08/2025].

25. Trust Everything, Everywhere, ARIA's Opportunity Space [Online]: https://www.aria.org.uk/opportunity-spaces/trust-everything-everywhere [Accessed: 4/8/25]

26. The ESPRC funded Circular Robot 5.0 R&D project [Online]. https://www.techuk.org/resource/unlocking-interoperability-in-robotics-with-self-sovereign-digital-wallets.html [Accessed: 30/4/25].

27. Safeguarded AI, ARIA's Programme [Online]: https://www.aria.org.uk/opportunity-spaces/mathematics-for-safe-ai/safeguarded-ai [Accessed: 4/8/25]

Authors

Robert Richardson is Professor of Robotics, at the University of Leeds where he heads up the Real Robotics Lab. His research interests include robots in the real world, in the air, on the ground and underwater. He is a member of the UK Robotics Growth Partnership Committee and Ex-Chair of the EPSRC UKRAS network.

Ipek Caliskanelli is a Principal Research Engineer at the UK Atomic Energy Authority. Her work focuses on interoperability, coordination, and collaboration among large-scale robotic systems in high-consequence environments, with a strong emphasis on advancing technology readiness levels and integrating state-of-the-art robotics into nuclear applications.

Mini Rai, a UK All-Party Parliamentary and Scientific Committee member, is the founding Director of Orbit Rise Ltd. She is a space engineer specialising in robotics, automation and control for extreme environments. Previously. Mini was a Chair Professor at the University of Lincoln and an Associate Professor at the Surrey Space Centre, UK.

Having previously successfully floated a technology business on the London Stock Exchange, Dominic Keen launched Britbots to support best-in-class UK-based robotics, AI and automation ventures, capitalising on British technical expertise in these areas. To-date Britbots has backed over fifty companies. Dominic is a member of the UK Robotics Growth Partnership and also sits on the investment committee of New Anglia Capital, a public-sector venture capital fund.

Aaron Prather is the Director of Robotics and Autonomous Systems at ASTM International, leading global efforts in robotics standards, workforce development, and safety. He bridges industry, academia, and government to advance responsible robot deployment across sectors.