ROS-M Summary
NDIA GRCC 2017

Open Source Benefits: DoD Policy for Open Source promotion

- Seeking data deliverables and rights in technical data and computer software sufficient for competition throughout the life cycle as an objective;
- Continuous competition throughout the life cycle;
- Increasing capability to the warfighter on a faster development timeline;
- Reducing lifecycle costs;
- Shared risks with other programs;
- Minimizing duplication for technology development investments, shared life cycle costs;
- Collaboration promotion through peer reviews.

Acquisition incentives

- Higher return on investment across the entire life-cycle
- Lower Verification & Validation and certification costs
- Standard contracting language with many of the relating issues already resolved or accounted for will decrease the time required to develop acquisition packages.
- Lower technical and schedule risk resulting from use of mature/stable ROS-M infrastructure
- Increased supplier base for both development and sustainment

Industry Incentives

- Decreased schedule and technical risk
- Small business access to larger opportunities
- Increased partnering ability
- Increased capability (at lower cost) increases market size
- Workforce recruitment and training

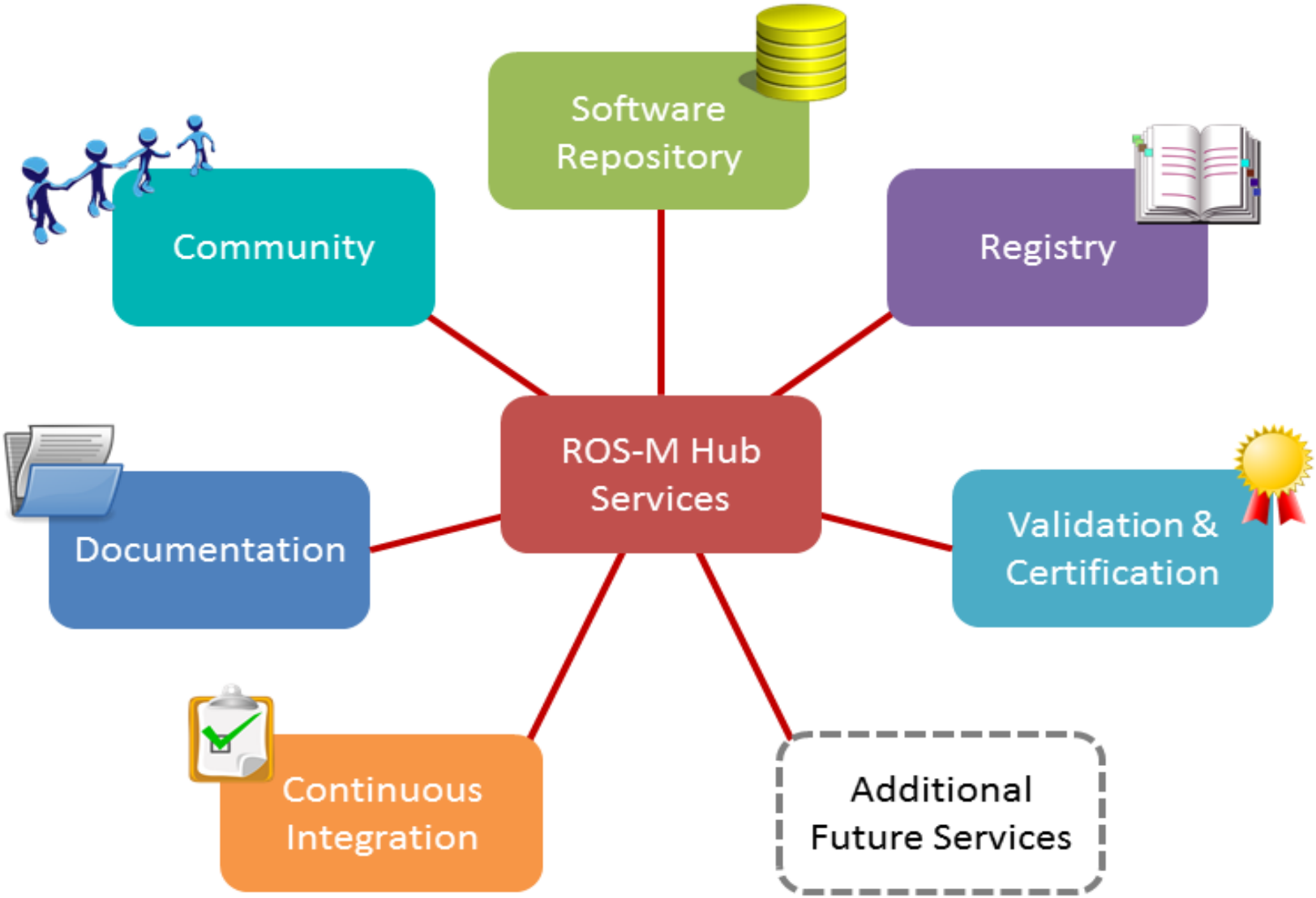
Researcher Incentives

- Government labs and academia
- Leverages benefits of current ROS usage
- Transition to product
- Focus on capability development

ROS-M Working Groups

- **Software Stack**
  - Demonstration Scenario
  - ROS 2 modules to be incorporated or developed
  - ROS 1 modules to be bridged
  - Tool and OS requirements
- **Software Process**
  - Repository and Access Control
  - Registry
  - Configuration Management
  - Metadata
  - Wiki/shared documentation
  - Software maturity and coding standards
- **Security**
  - The mil-standards associated with open source software
  - Ros-M consortium IA responsibilities versus acquisition body's responsibilities
  - Security best practices
- **Business Process**
  - Marketing and educational material requirements
  - DFARS and Licensing analysis
  - Community standards
  - DoD open source community best standards

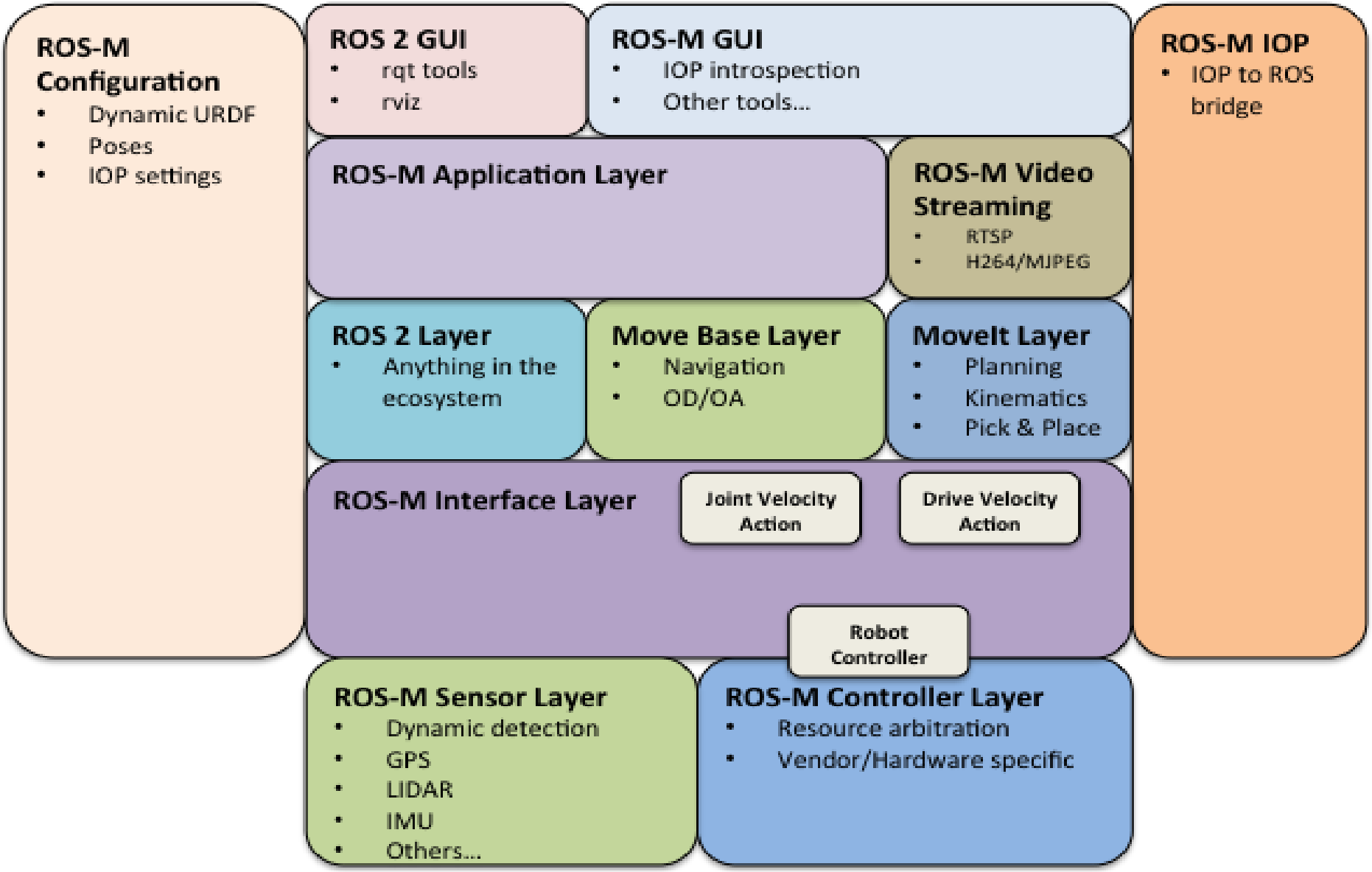
ROS-M Organization



Metadata contents

- **Package name** – The name used within the development environment for the ROS-M package. Example: “cv_bridge”
- **Level of Maturity** – The state of the package (experimental, development, stable, deprecated)
- **Last Updated Date** - The date when this package was last modified.
- **Maintainer(s)** – The names and contact info for the current maintainers of this package. This can be a company name or individuals.
- **Author(s)** – The names of the original authors of this package. This can be a company name or individuals.
- **License** – The license associated with this package. Example: “BSD”
- **Link to source and/or binaries** – Links pointing to where a developer can find the source code, binaries, or an associated store/site where they can be purchased
- **Package description** – A short description of what the package offers in terms of functionality.
- **Package compatibility** - What versions of ROS-M does this package work with?
- **Metatags** – Metatags associated with this package.
- **Link to tutorial(s)** – Tutorials may take up several pages and should be separate from the main package page
- **Link to bug/feature tracker** – A link to where bugs and features are tracked for this particular package. This may be an external link.
- **Overview section** – a more in-depth explanation of the package (as compared to the “package description” section). This may include discussion of various algorithms or approaches.
- **Example usage section** – A short, basic example use of the package.
- **Link to the code API** – link to the API documentation, such as a Doxygen page. This might be an external link.
- **Link to a FAQ/QA** – Link to the QA site with tags relevant to this package (if possible)
- **Changelog** – This lists any changes that are made for each revision for this package
- **Change list** – This lists any changes that are made in each stack release for this package
- **Reviews** – API or code review meeting notes as well as the ability to request a new review of this package
- **Dependencies** – A list of packages that this package depends upon
- **Used by** – A list of packages that are known to use this package
- **Projects and Programs** - a list of military projects and programs that are known to have used this package.
- **Version** - specify which version of package is being used

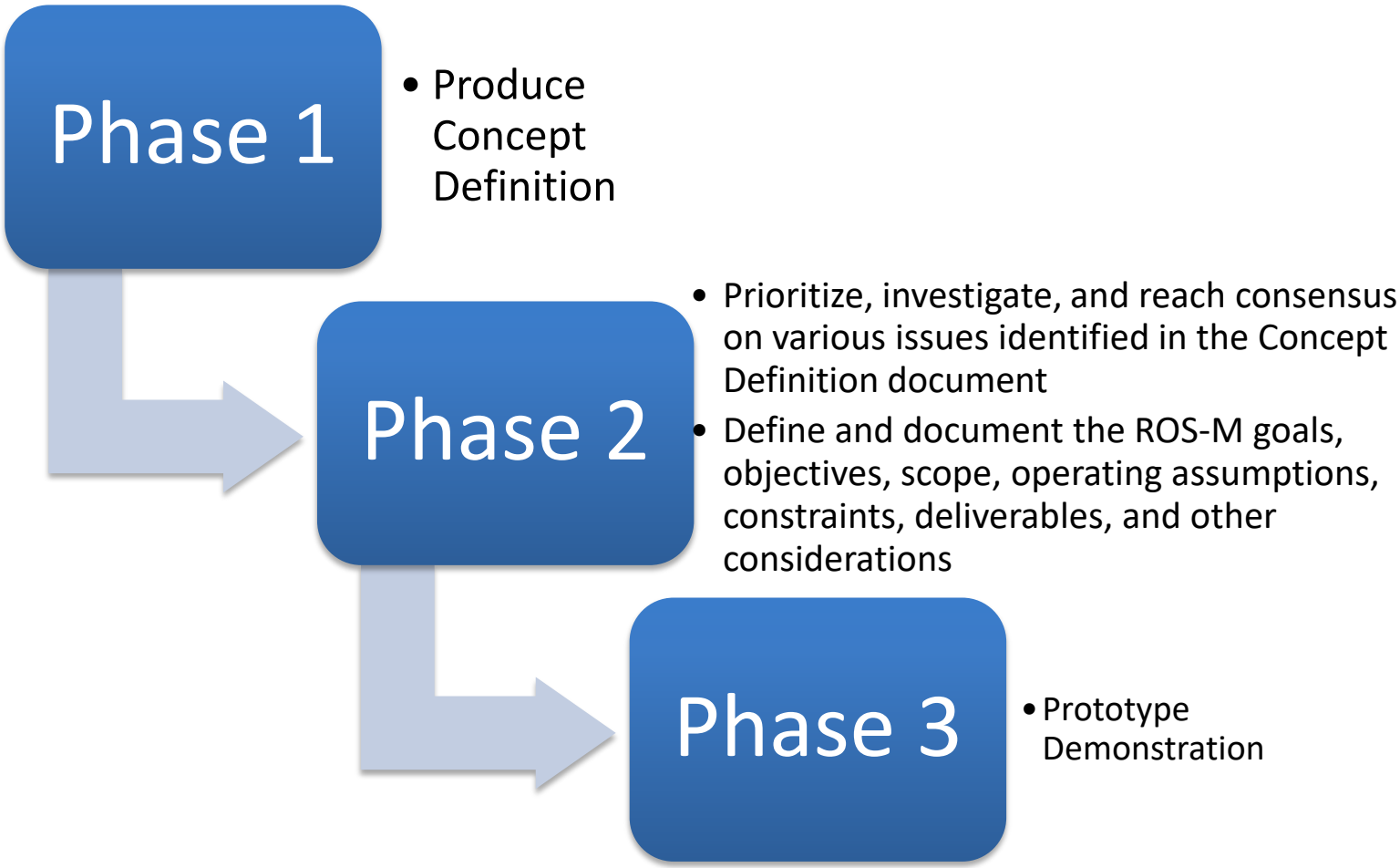
Conceptual Model



Core ROS-M components

- ROS to JAUS/IOP bridge
- Audio/video streaming
- Sensor drivers
- Hardware abstraction interface
- Resource arbitration / behavior system
- Application Debugging tools

Next Steps


Phase 1
• Produce Concept Definition
Phase 2
• Prioritize, investigate, and reach consensus on various issues identified in the Concept Definition document
• Define and document the ROS-M goals, objectives, scope, operating assumptions, constraints, deliverables, and other considerations
Phase 3
• Prototype Demonstration