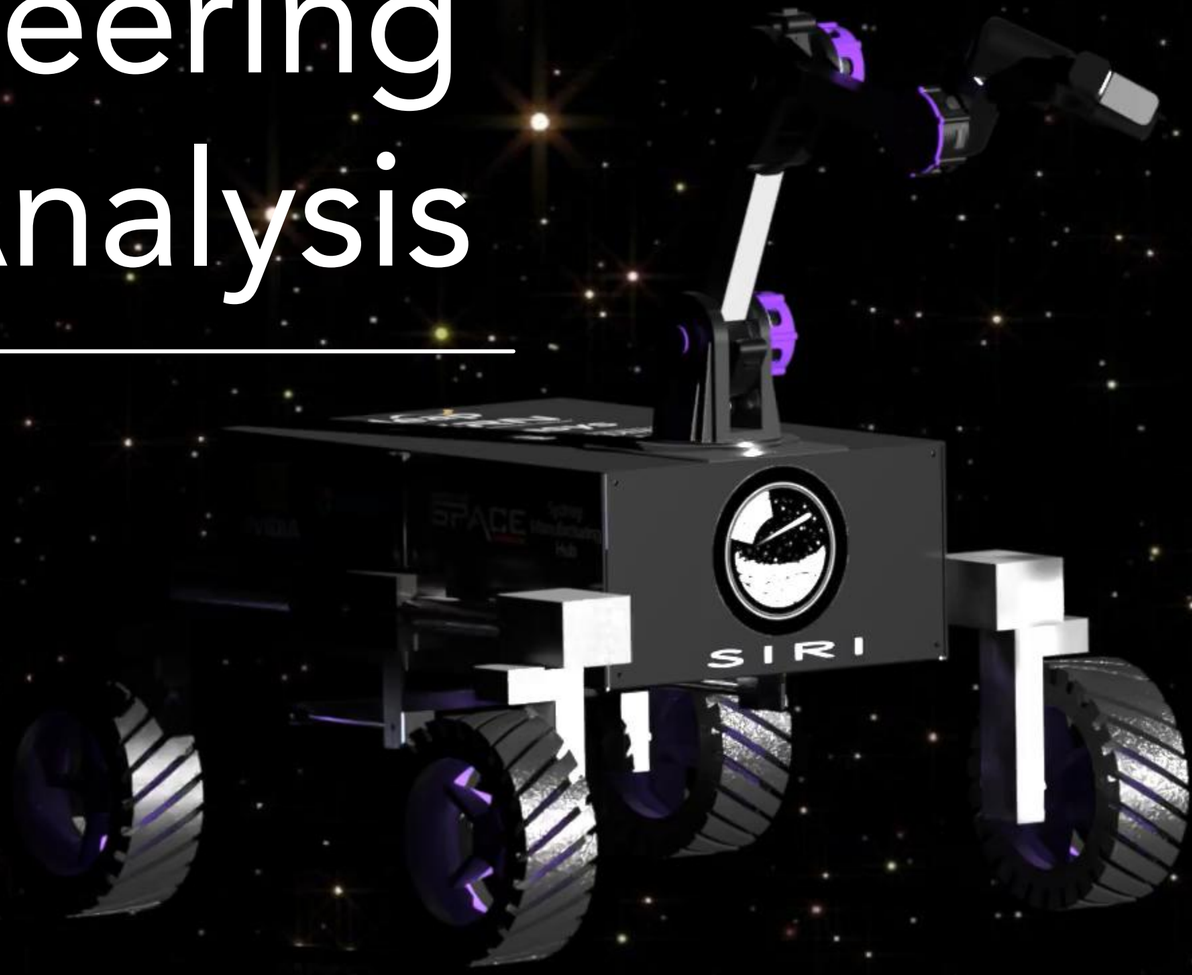
The University of Sydney

Sydney Interplanetary Rover Initiative (SIRI)

Systems Engineering Modelling & Analysis



Kaylee Li
Founder & Project Lead | SIRI

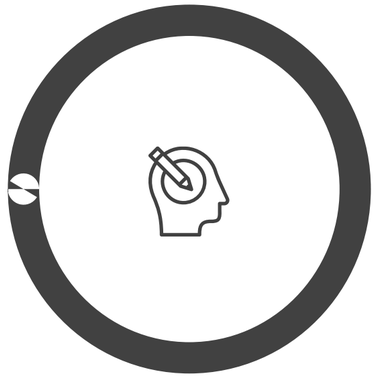
Arukshan Pavalachandran
Lead Systems Engineer | SIRI

February 18, 2025

Why Do We Need SysEng @ SIRI ?

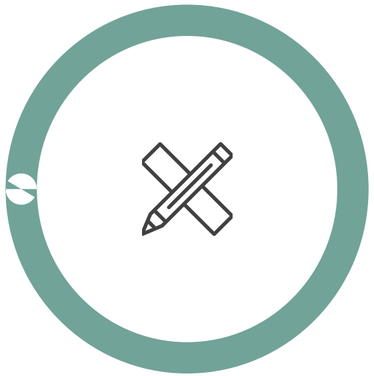
Stakeholder Alignment

Stakeholder needs and expectations are defined at the forefront. Ensures the final system meets its intended operational goals



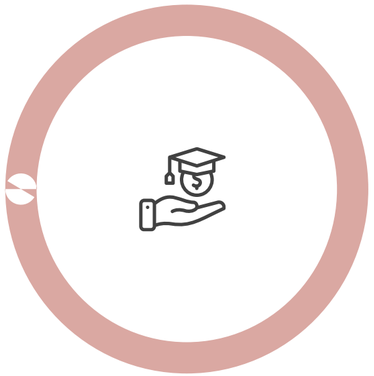
Manage Complexity

This process breaks down complex problems into manageable parts, making it easier to understand, design, and integrate



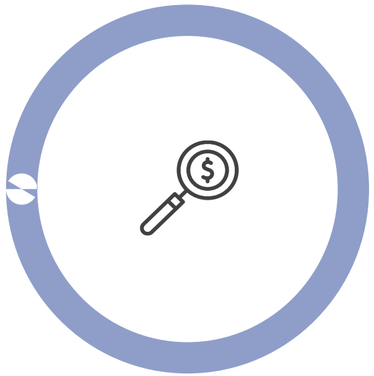
Reduce Risks

This process proactively identifies potential issues by avoiding invalid assumptions and validating concepts through trade studies.



Improves Quality

Designed for its purpose with minimal errors and iteration time, thus maintaining a high quality system



Controls Costs

The project is more likely to be on schedule and within budget due to minimal errors and iterations

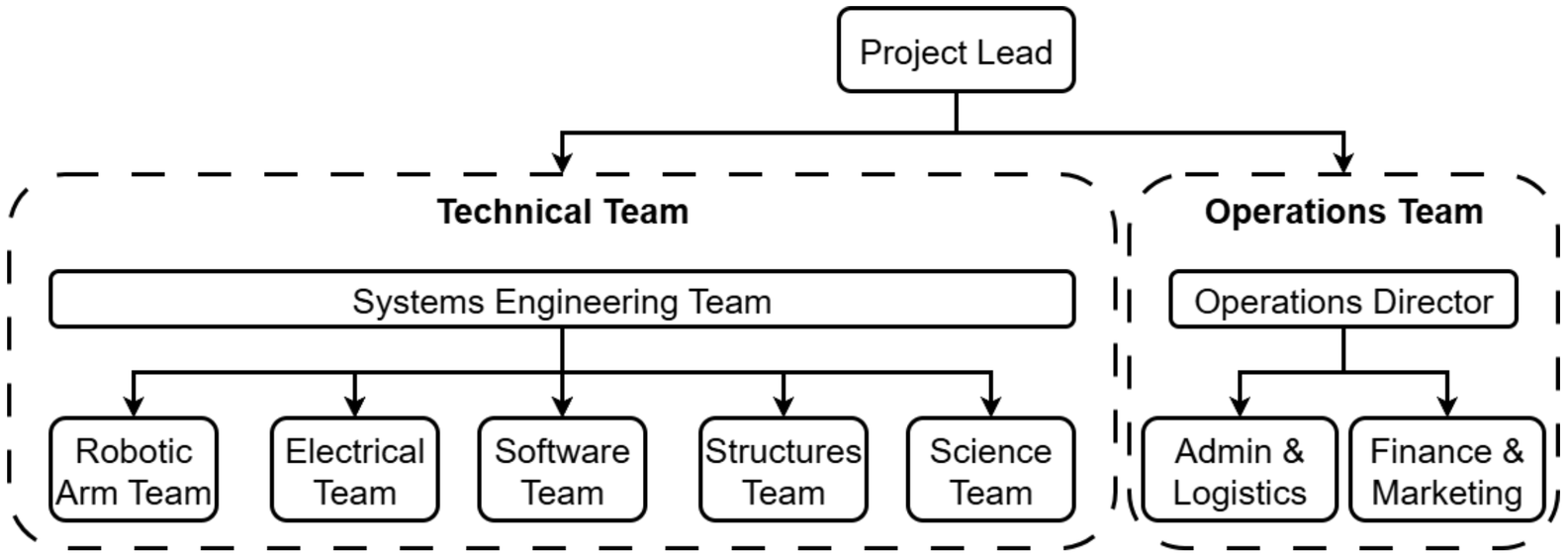
Motivation

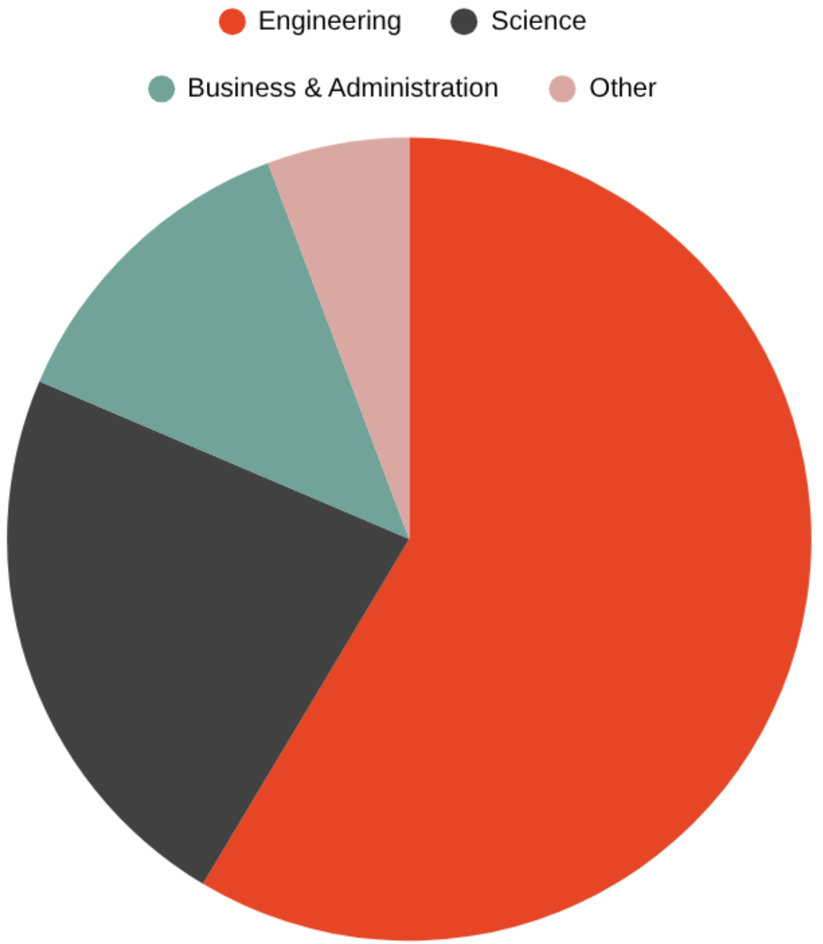
Although leading organizations in several industries are adopting MBSE practices, its application in projects of a smaller scale is minimal.

The Feasibility Of Implementing MBSE In A Student-Led Rover Team

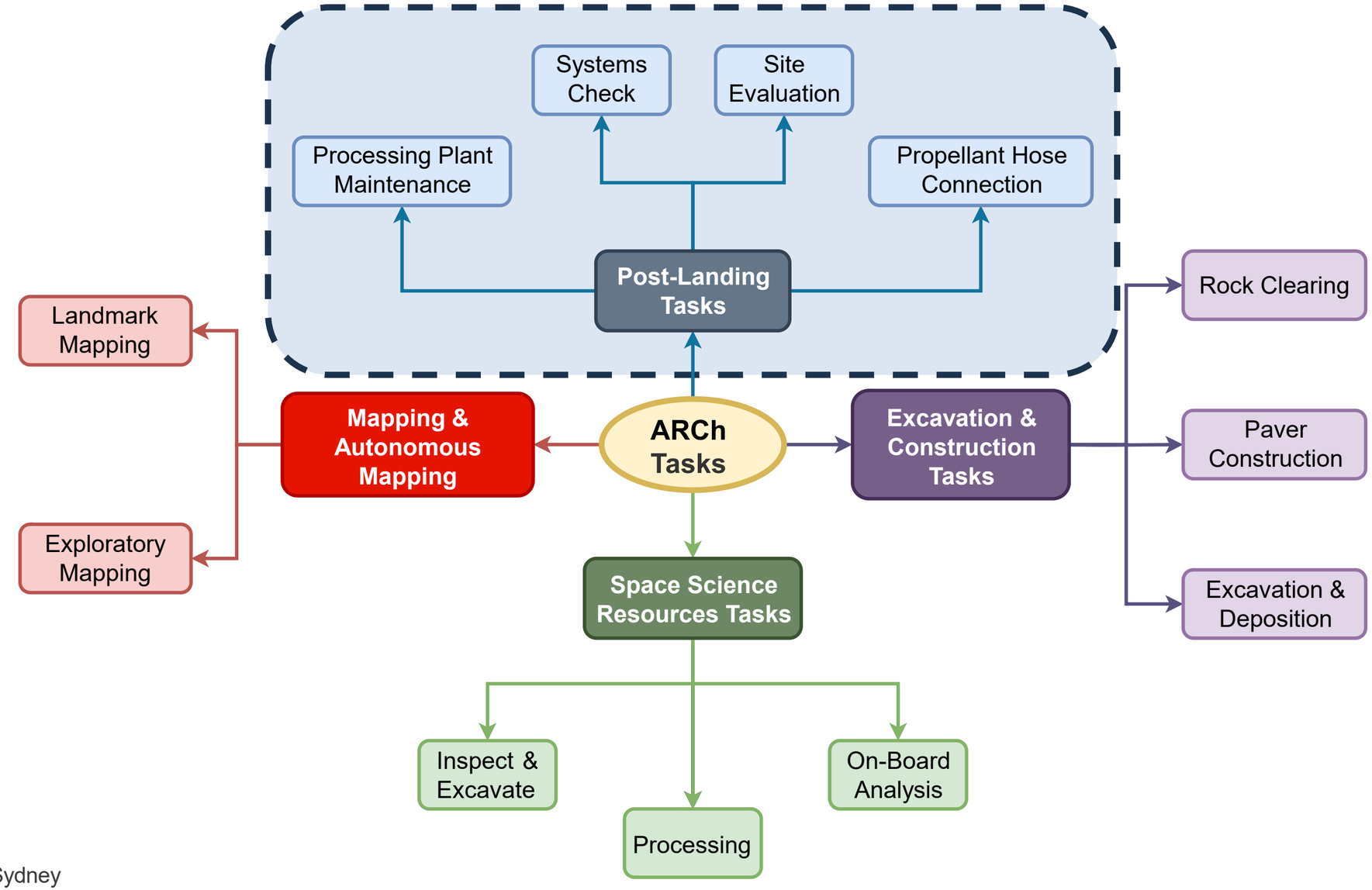
CONTEXT

The SIRI Project Team





The Australian Rover Challenge - Overview



ARCh : Post-Landing Tasks

1

Systems Check & Circumnavigate Lander

2

Navigate to and relay status readouts

3

Wireless communication from the processing plant

4

Perform a series of processing plant maintenance jobs

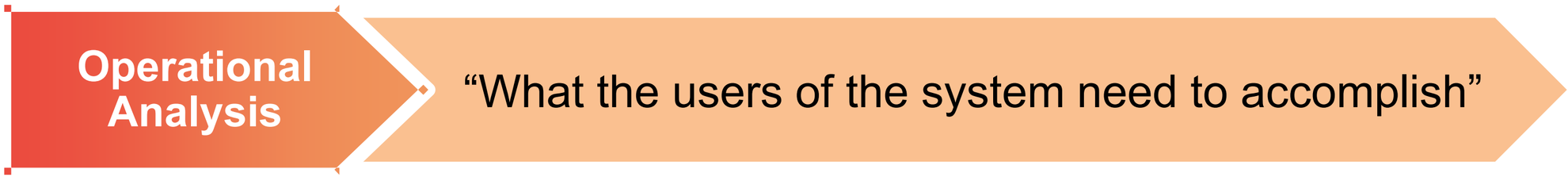
5

Connecting the lander to a gas line

Implementation

The Arcadia Method

The Arcadia Method



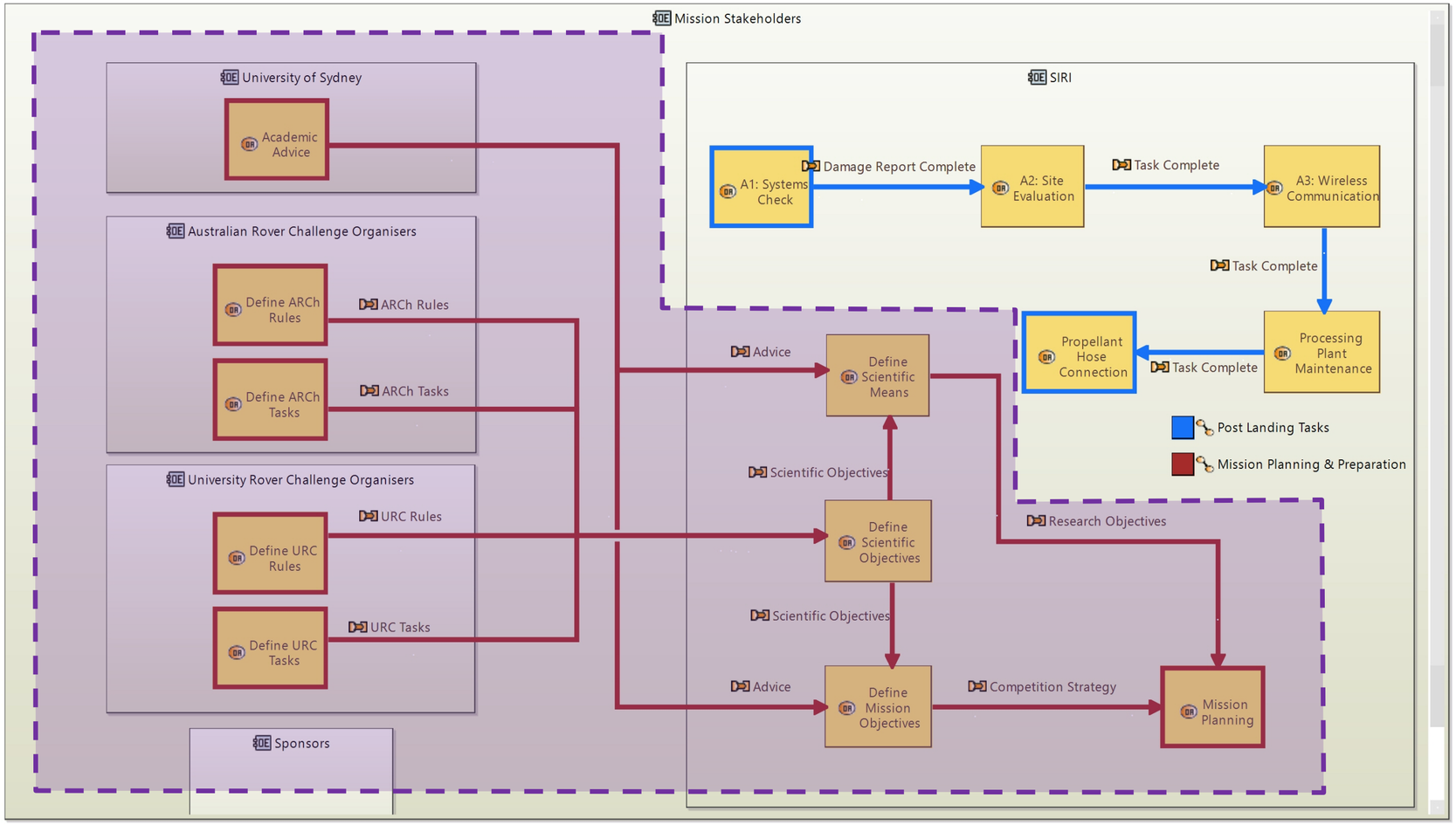


Figure 1.1: Operational Architecture Diagram – 1st iteration

The "Mission Planning & Preparation" chain is not useful

- in setting the context or
- to design the rover

Only the "Post Landing Tasks" chain is suitable for this diagram

- activity allocation to the actors
- the interactions among the activities

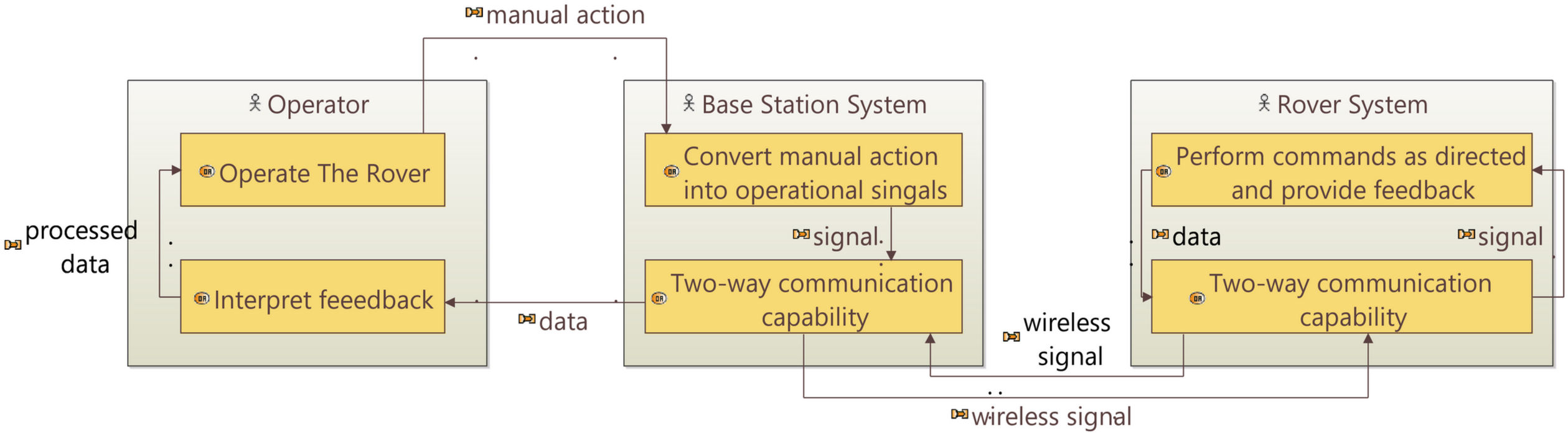


Figure 1.2: Operational Architecture Diagram – 2nd iteration

The Arcadia Method

System Analysis

“What the system must accomplish for the users”

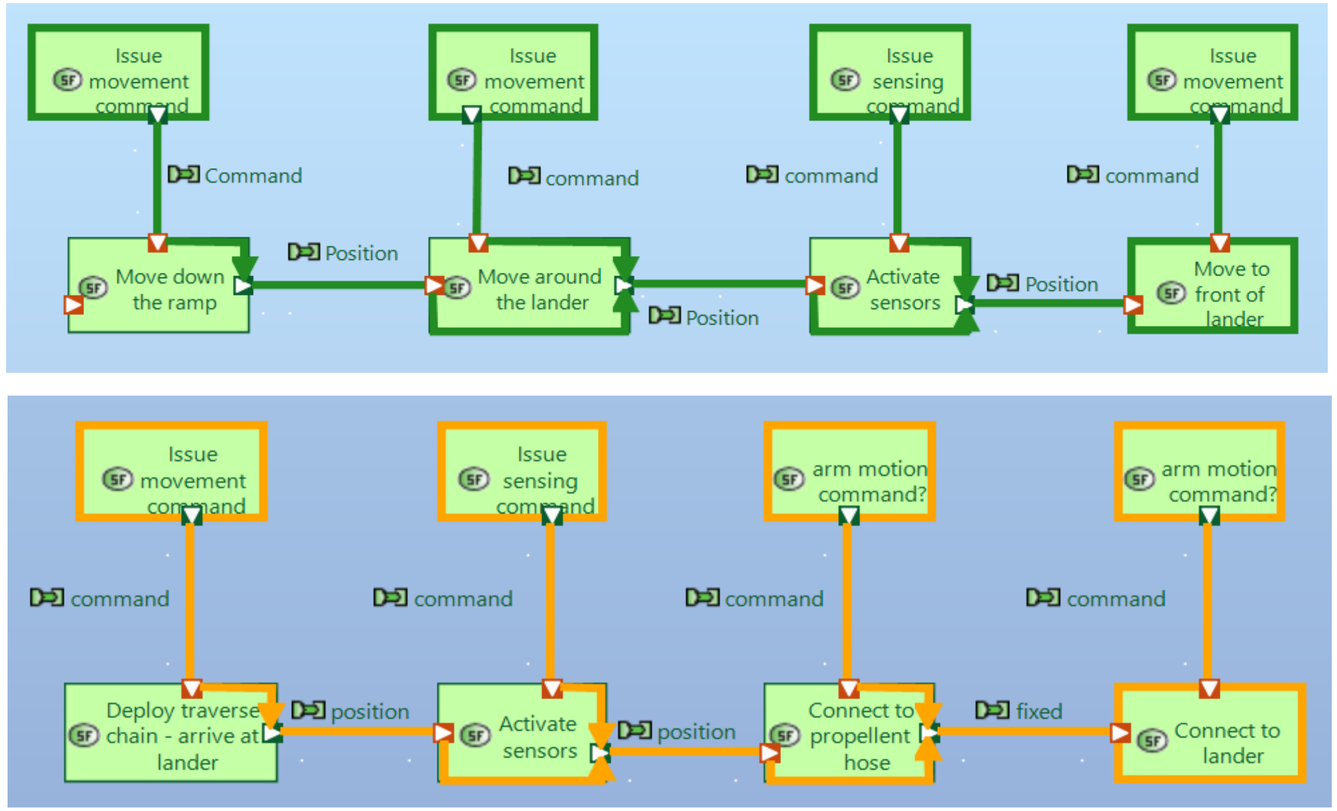
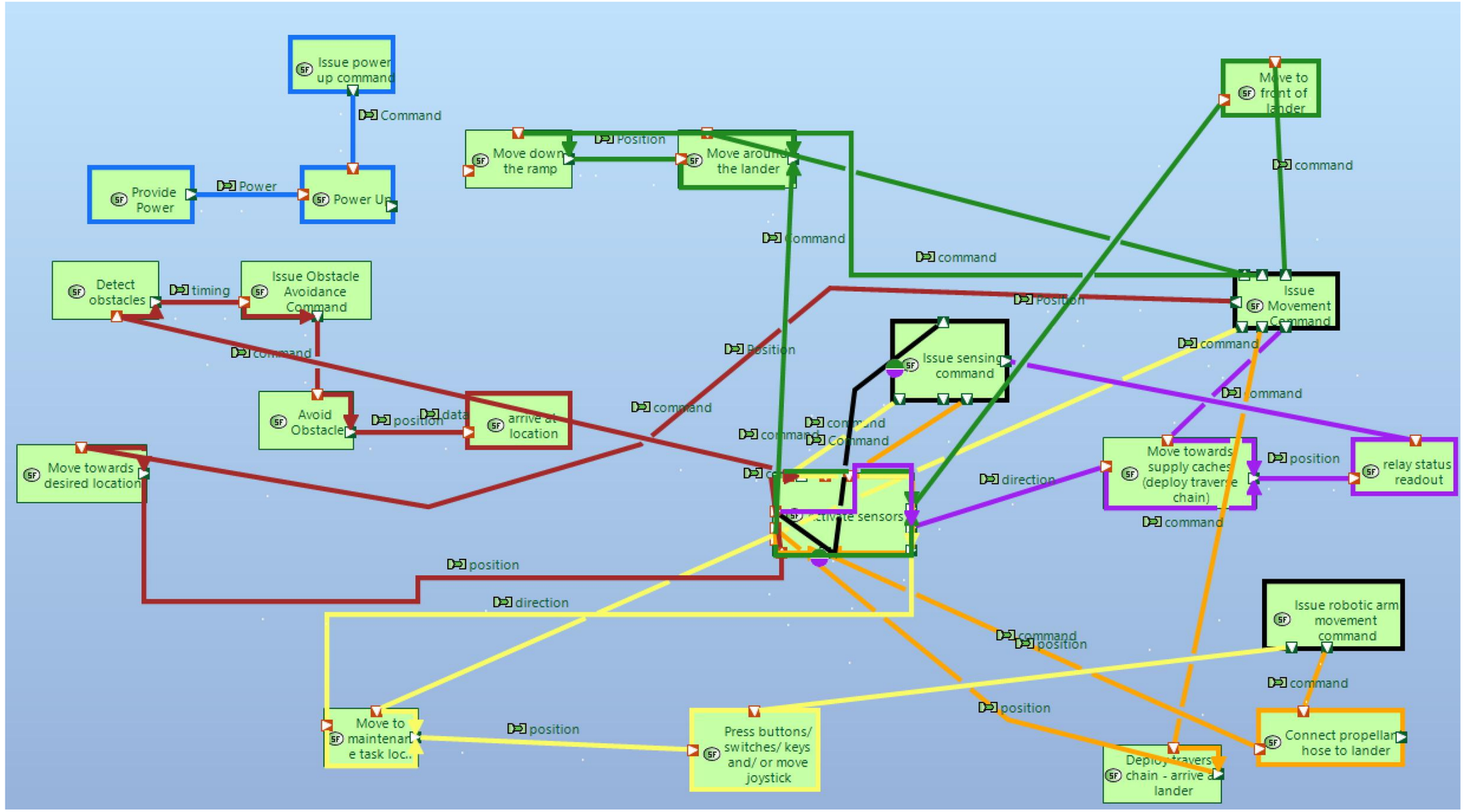


Figure 2.1: System Architecture Diagram –
Top: Traversal functional chain,
Bottom: Propellant Hose Connection Functional Chain

- The layout represented **scenarios**, where;
  - functions were a **series of tasks,**
  - rather than **specific tasks** the system can perform
- The aim was to model a **dependency diagram**
- System functions were **repeated** in the different functional chains



Power Up

Traverse

Systems Check

Site Evaluation

Plant Maintenance

Propellant Hose Connection

Figure 2.2: System Architecture Diagram – 2nd iteration – **Eliminating repeated functions**

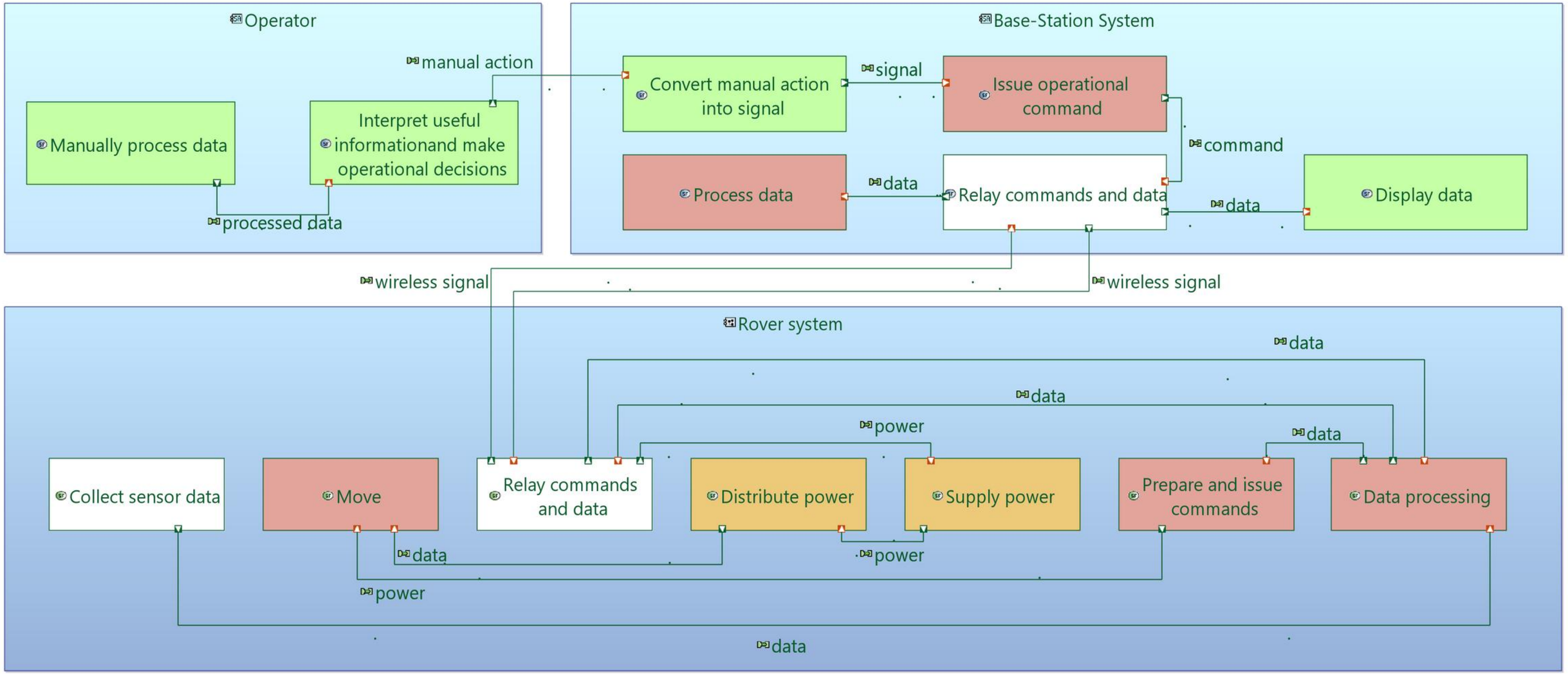
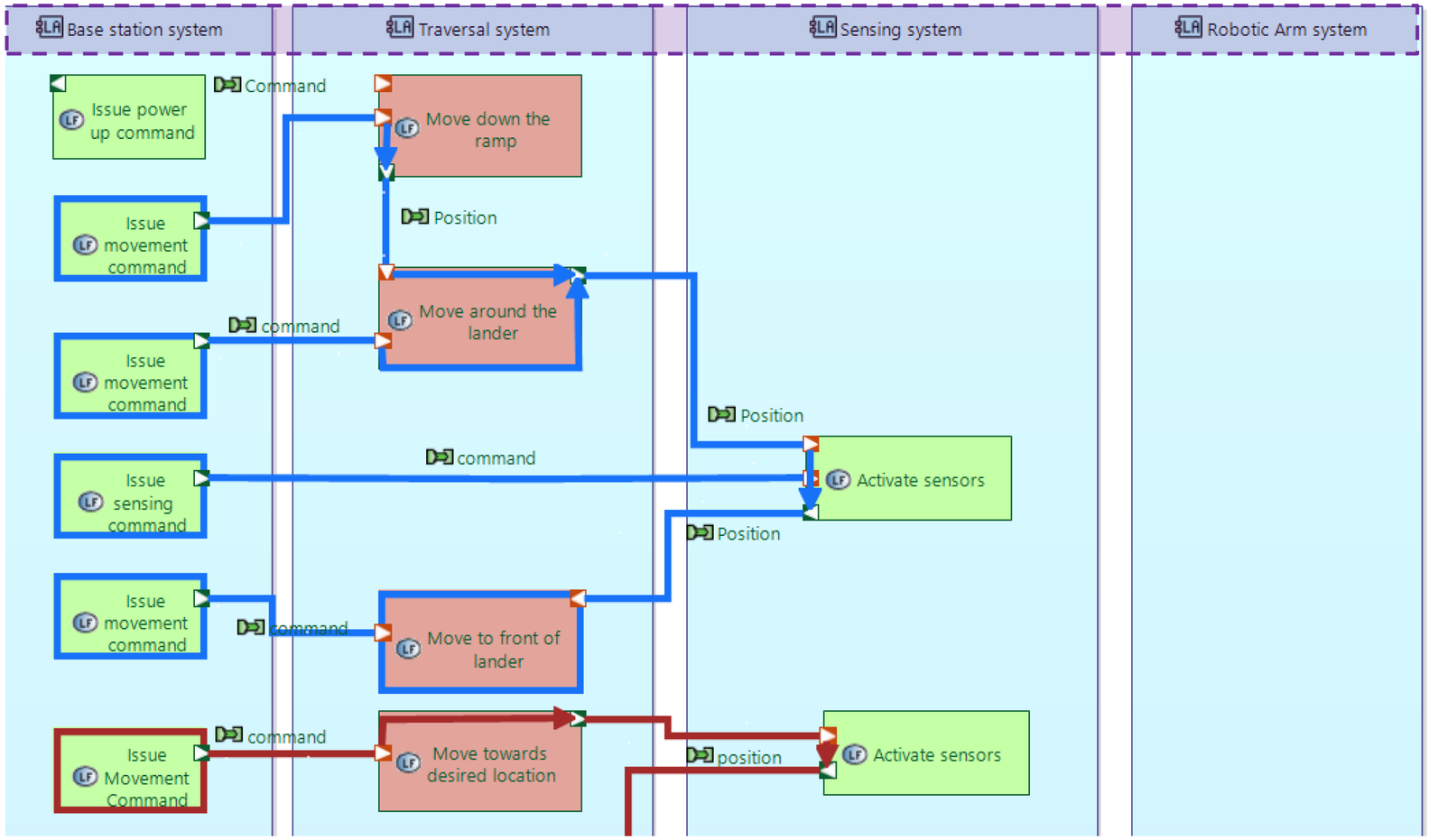


Figure 2.3: System Architecture Diagram – 3rd iteration –
Revised System Functions mapped to System Actors

The Arcadia Method

Logical Architecture

“How the system will work to full-fill expectations”



Subsystems modelled as **logical actors** instead of **logical components**

- Certain 1st iteration functions were "**scenarios**" (i.e. series of tasks)
- Must be tasks the **subsystems can perform**

Figure 3.1: Logical Architecture Diagram – 1st Iteration – **Transitioning from the System Analysis layer**

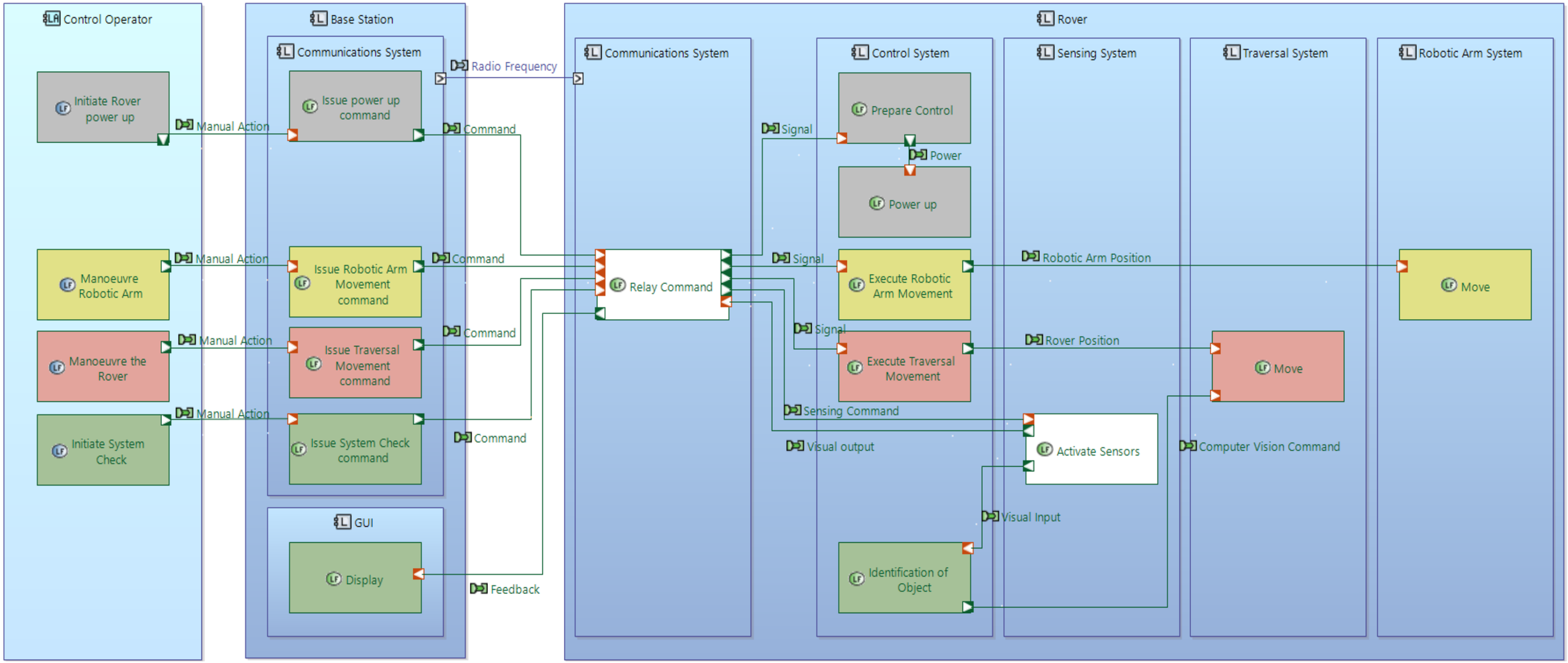


Figure 3.2: Logical Architecture Diagram – 2nd Iteration –
Eliminated repeated functions

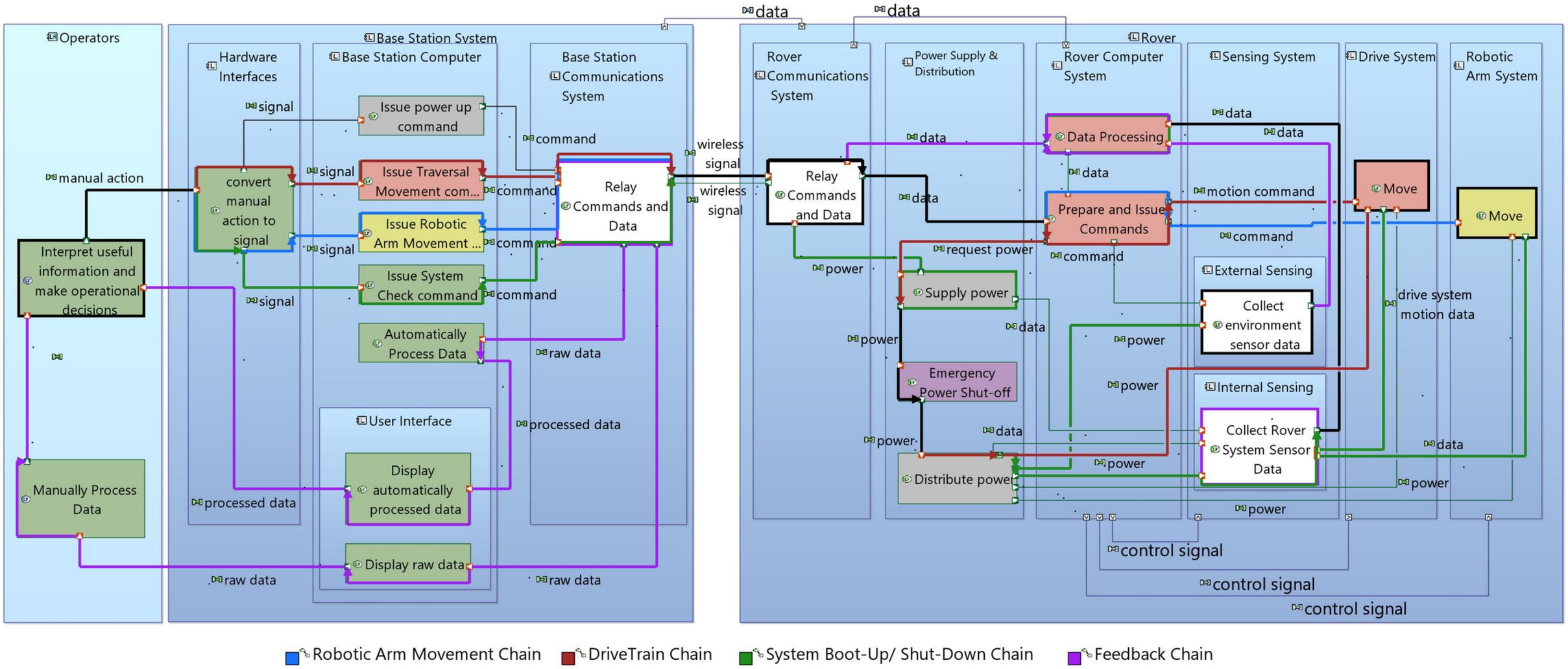


Figure 3.3: Logical Architecture Diagram – 3rd Iteration – **Revised functions**

The Arcadia Method

Physical Architecture

“How the system will be developed and built”

- All actors, functions and their exchanges established in the logical phase undergo a phase transition to appear in the physical architecture layer.
- This model helps visualise all the components on the potential solution system, along with its physical and logical links.
- All components are mapped to their suitable logical functions that trace back to the operational analysis layer.

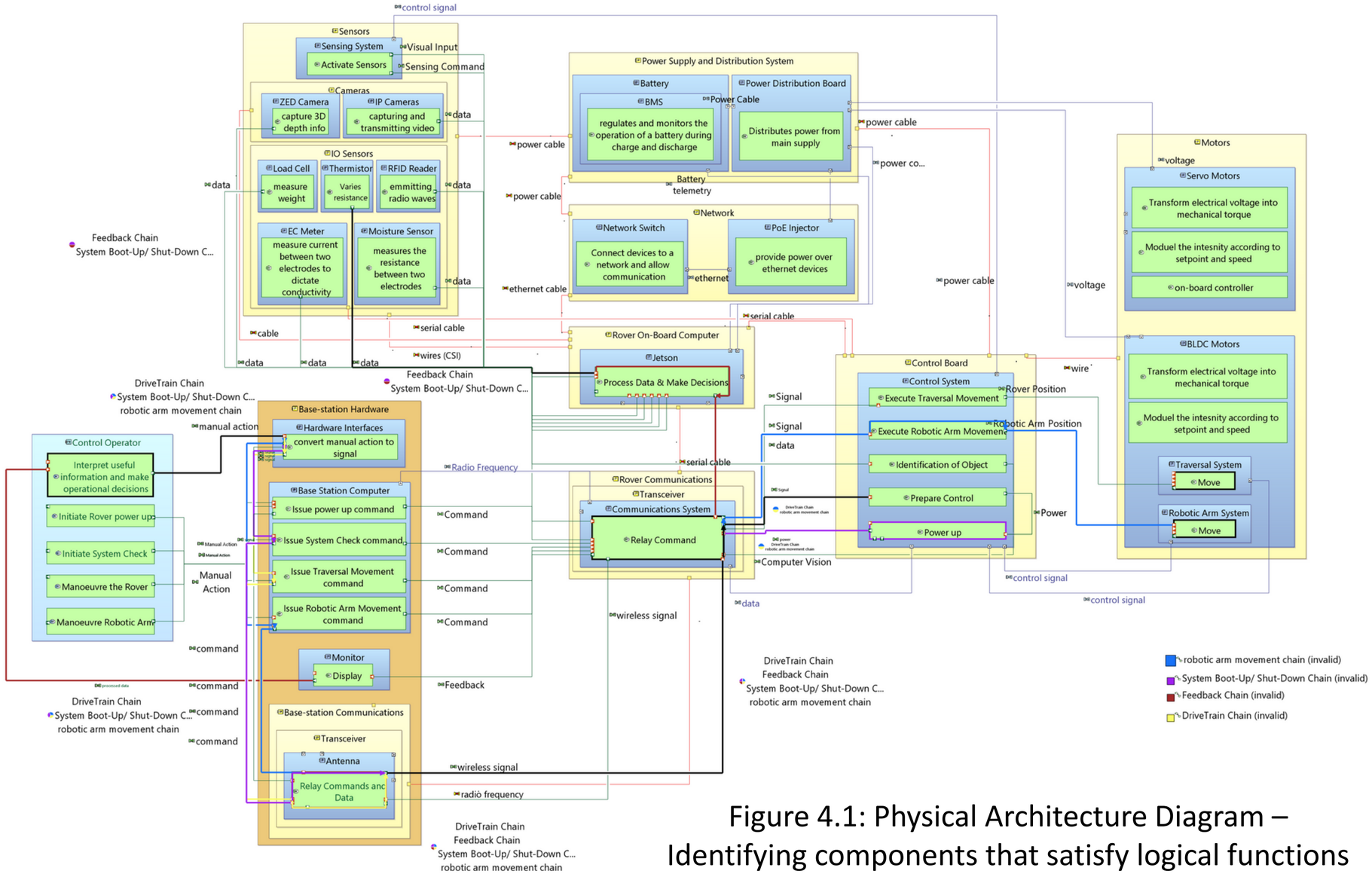


Figure 4.1: Physical Architecture Diagram – Identifying components that satisfy logical functions

Modes Machine

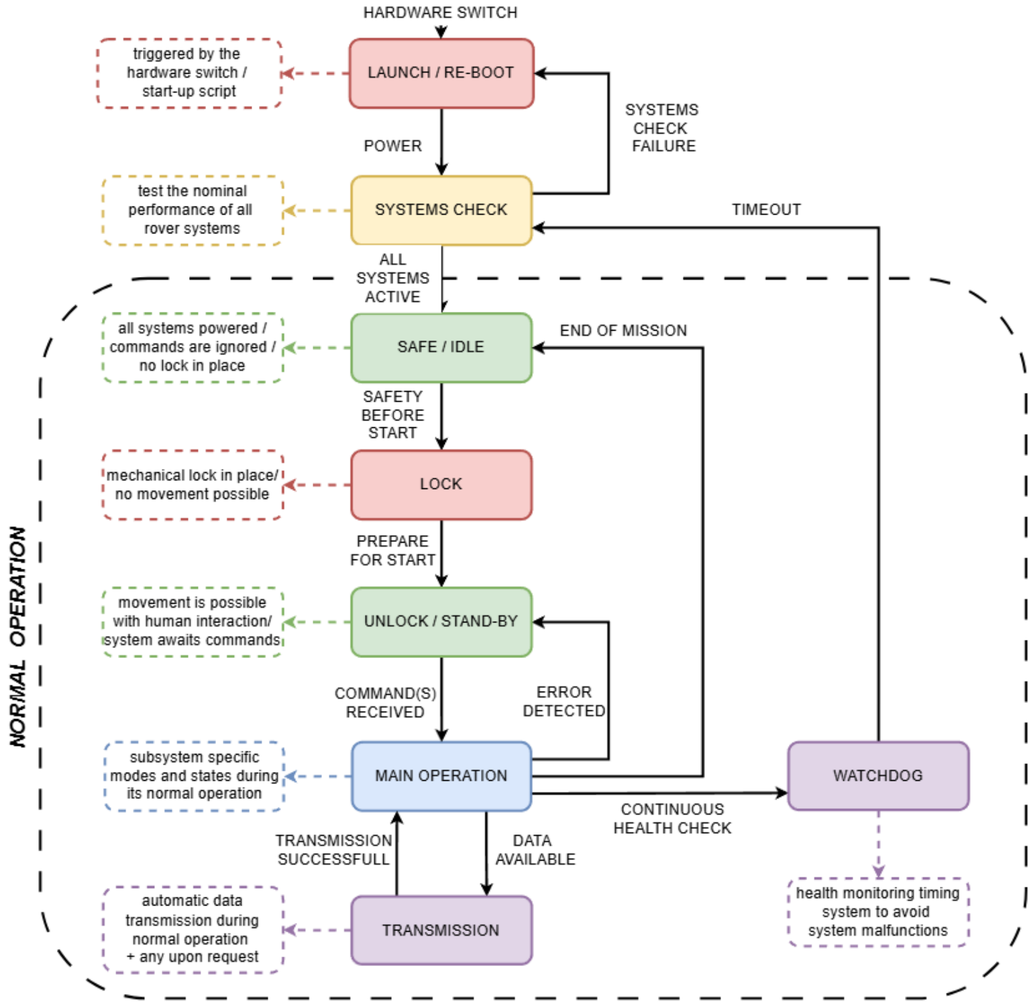


Figure 5.1: Modes Machine of Rover system (general)

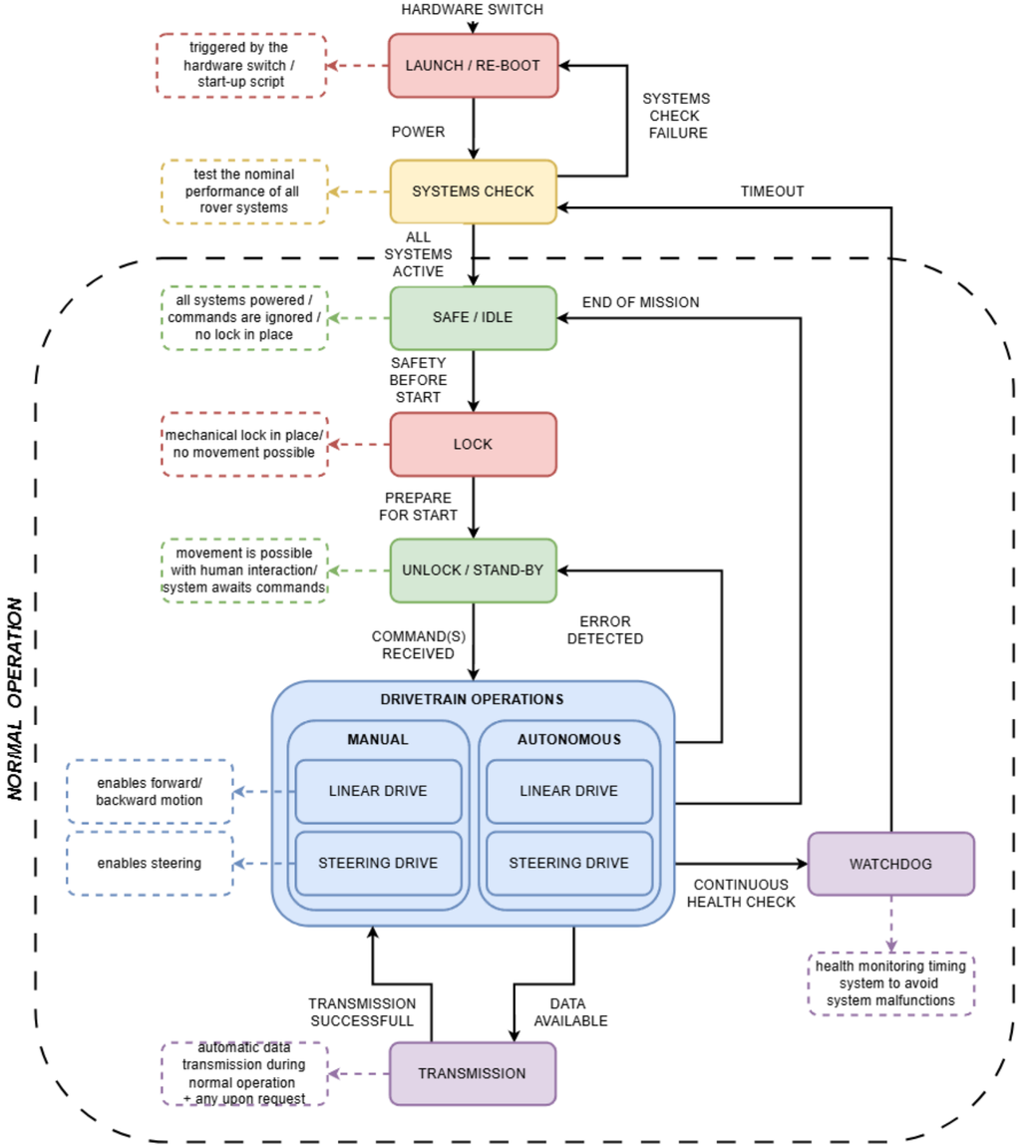
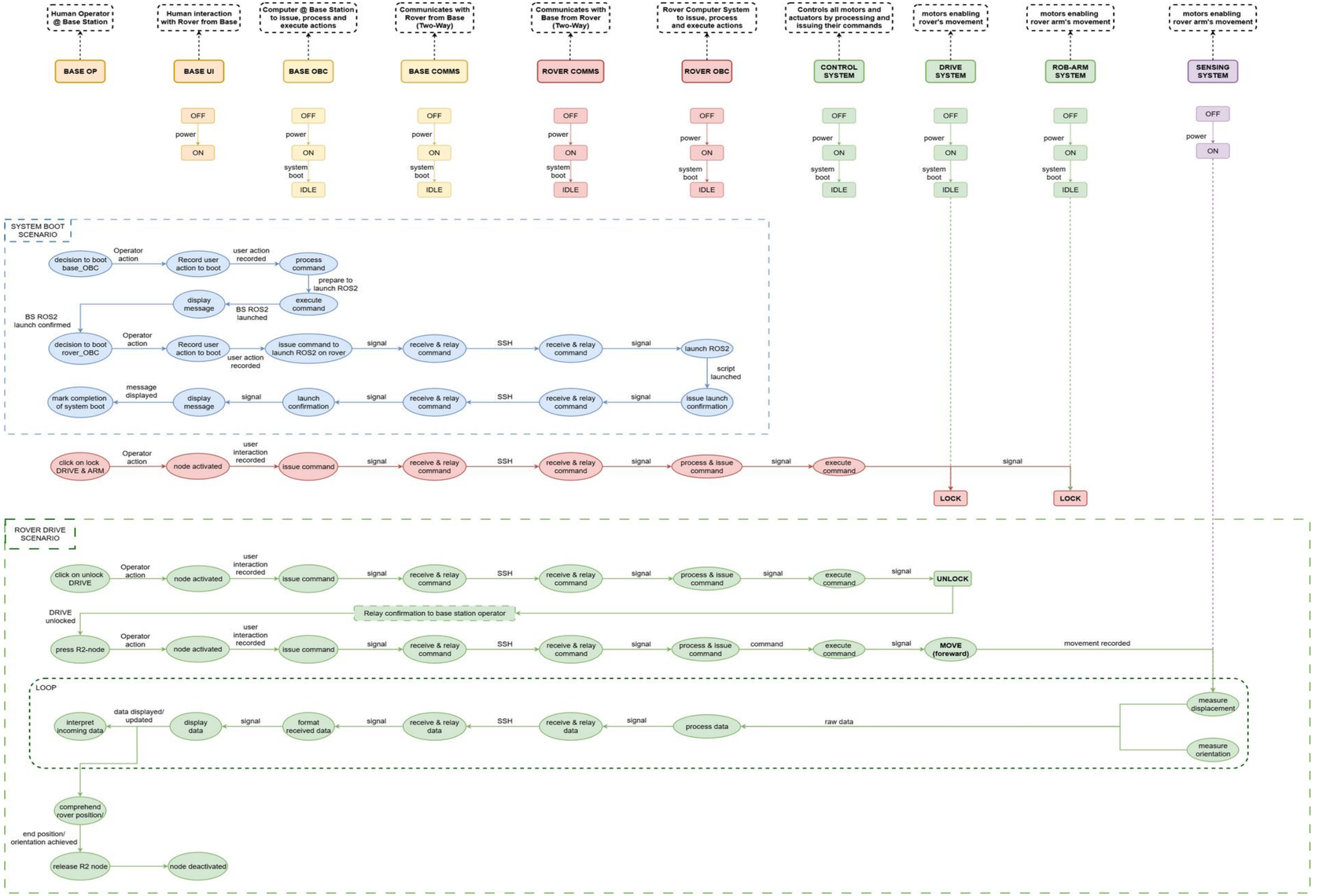


Figure 5.2: Modes Machine of Drivetrain subsystem (specific)

Exchange Scenario



- Shows the **trajectory** from A-B through time as opposed to the previous models that are static diagrams
- Helps **identify gaps when modelling the process** of performing tasks using the functions defined in the model
- Helps visualise the **subsystems** involved and their **interactions** and inputs in performing tasks.

Figure 6.1: Exchange Scenario Diagram

Conclusion

Summary

- Modelled the ‘Post-Landing’ tasks in Capella, producing;
  - Operational Architecture - identified user goals
  - System Architecture - defined system functions and interactions
  - Logical Architecture - identified functions to satisfy system functions
  - Physical Architecture - identified components to satisfy logical functions
  - Modes Machine - defined the modes available in the system and the actons that trigger such modes
  - Exchange scenarios - helped visualise the order of operation in time
- This procedure is replicated to model all other remaining tasks in ARCh

Future Work

Implement the **Modes Machine** and **Exchange Scenario** diagrams to integrate with the project in Capella.

Integrate our **project requirements** with the project models in Capella

- Early validation and feasibility
- Strong traceability

Recommendations & Best Practices

- Form a subsystem of members specifically for Systems Engineering, who will commit to learning and implementing the practices.
- Introduce MBSE training very early on in the project life-cycle through agile workshops with a specific focus.
- Following tutorials and example Capella projects can be a great exercise and starting point to get accustomed to the software and its features.
- Create modelling templates that can be used for future iterations of the project by a different team or for the adoption of future projects

Thank You!

Special Mention To:

- ***Mr. Tim Carter – MBSE Mentor***
- ***Ms. Kaylee Li – Project Lead @ SIRI***
- ***Systems Engineering Team @ SIRI (Past & Present)***