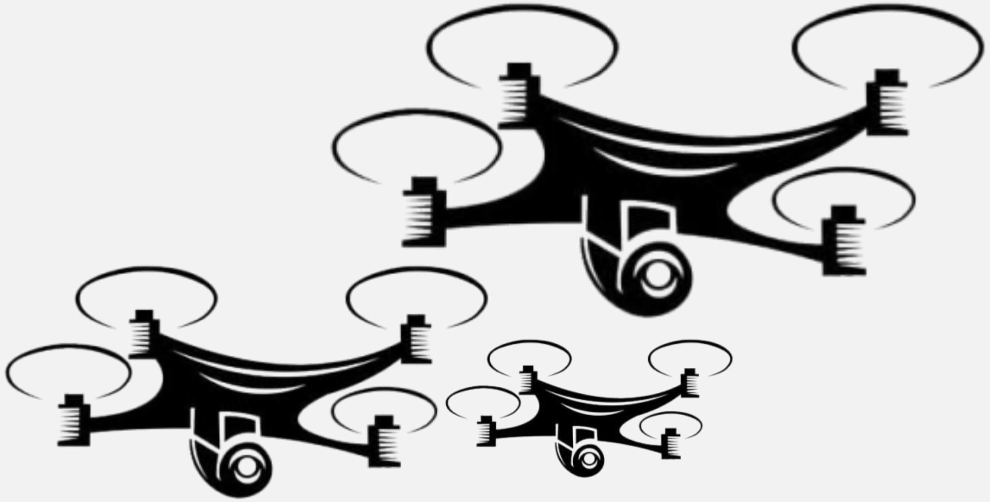
Autonomous Drones System for Light Show

SysML v2 (MBSE Approach)



Presented by

Ye Thura Oo (Moris)

4.9.2025

Introduction

Drone light shows are an emerging alternative to fireworks, using swarms of drones to create synchronized visual patterns in the night sky. Unlike delivery drones or camera drones that usually work one by one, a light show is a complex system that needs hundreds of drones to fly at the same time. Each drone must know its position, follow the timing, and keep safe distances from others. This requires high accuracy and strong coordination to make the show successful.

Main Challenges :

- Many drones must fly together without collisions
- Light and movement must be synchronized
- The whole system must follow air safety rules

Why MBSE is used?

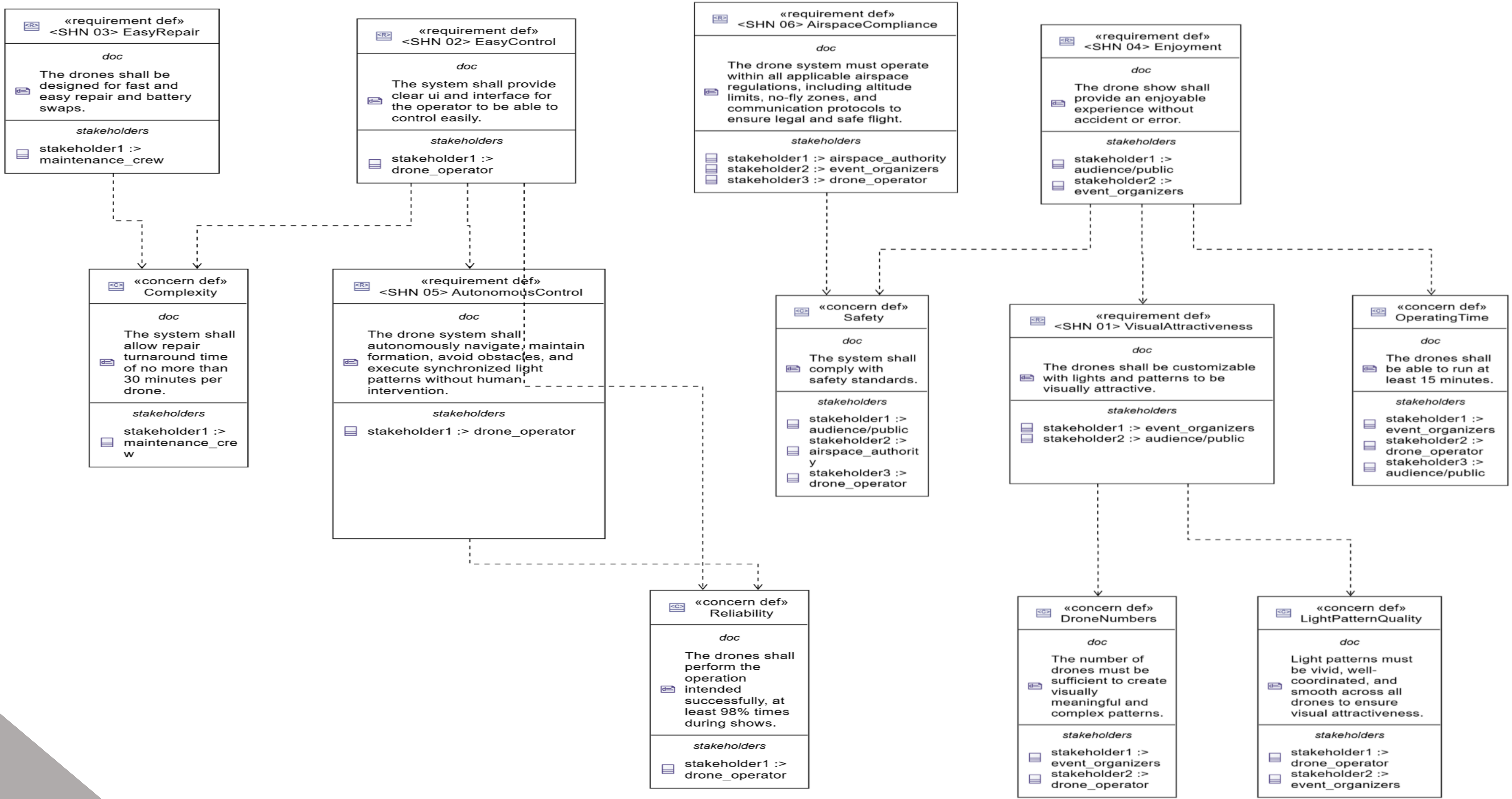
- Handle complexity by showing how many drones and subsystems connect and work together.
- Keep traceability from stakeholder needs to system requirements and design.
- Reduce errors by using diagrams and models instead of only text documents.
- Support verification and validation by linking requirements to test cases.
- Improve communication between engineers, managers, and stakeholders with clear models.

Stakeholders and Their Needs

The drone light show system must meet the expectations of multiple stakeholders, each with different priorities. Understanding these needs helps us define clear system requirements and ensures the system delivers value to everyone involved.

Stakeholder	Needs
Event Organizers	<SHN 01> Visual Attractiveness <SHN 04> Enjoyment <SHN 06> Airspace Compliance
Drone Operator	<SHN 02> Easy Control <SHN 05> Autonomous Control <SHN 06> Airspace Compliance
Maintenance Crew	<SHN 03> Easy Repair
Audience / Public	<SHN 01> Visual Attractiveness <SHN 04> Enjoyment
Airspace Authority	<SHN 06> Airspace Compliance

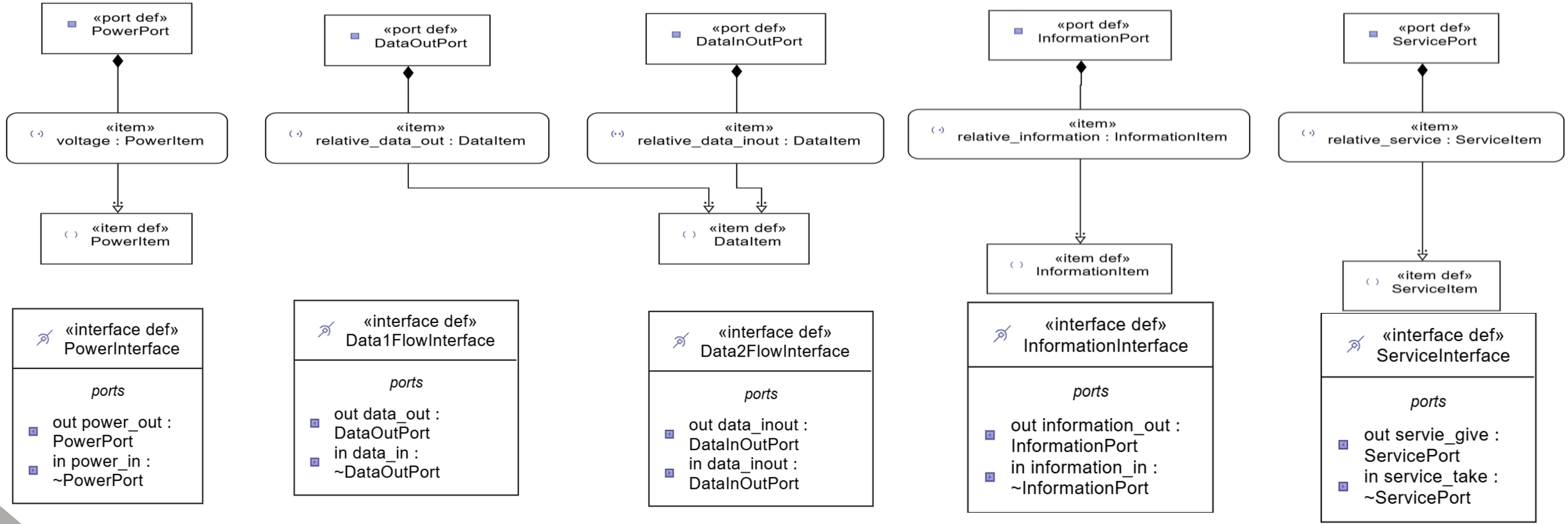
Stakeholders Needs and Concerns in SysML v2



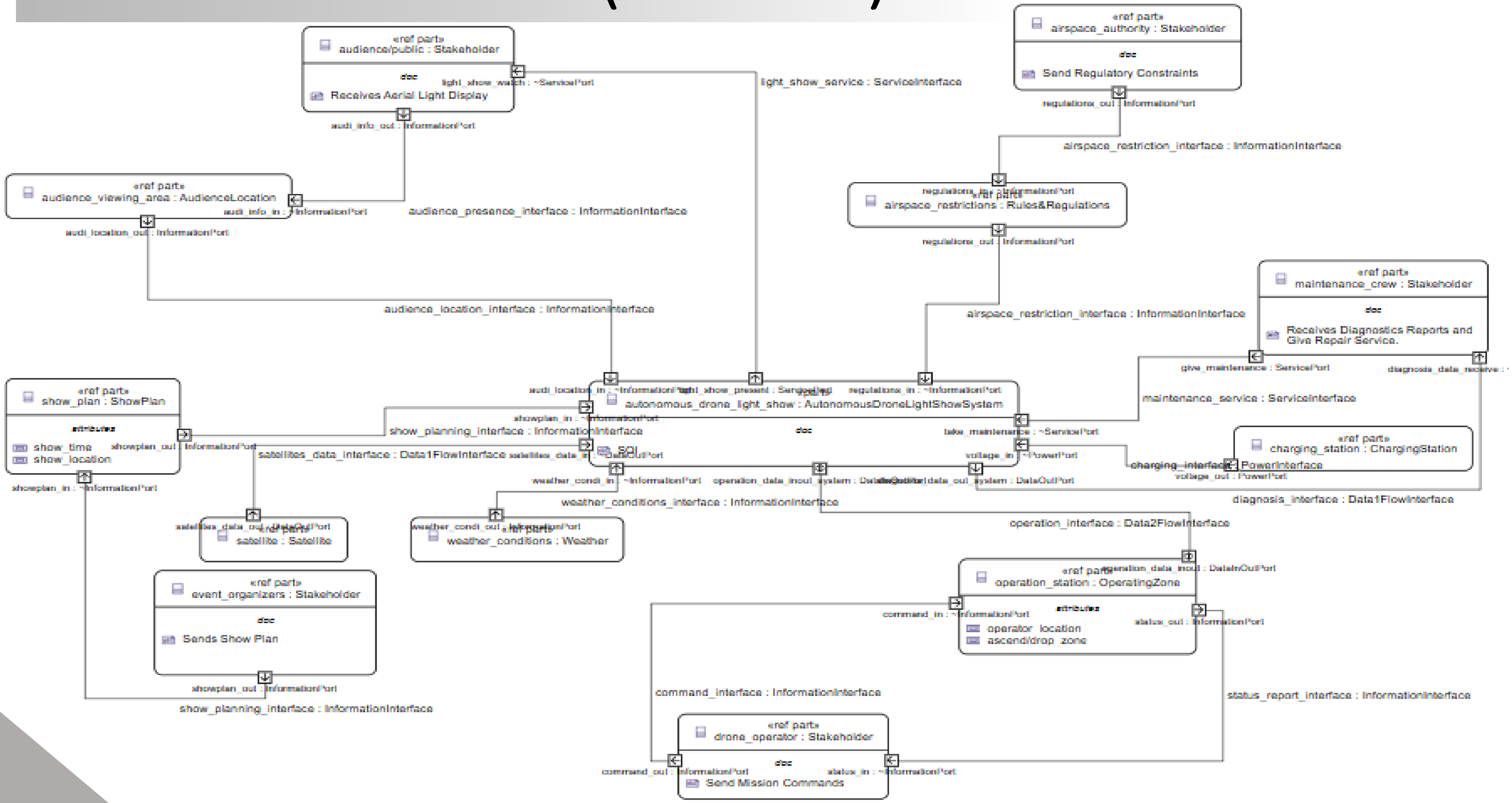
Ports and Interfaces Definition

The System of Interest (SoI) is the Drone Light Show System, which includes hundreds of drones working together to create synchronized patterns and images in the sky. The performance of the Drone Light Show System depends not only on the drones themselves but also on how the whole system interacts with external factors.

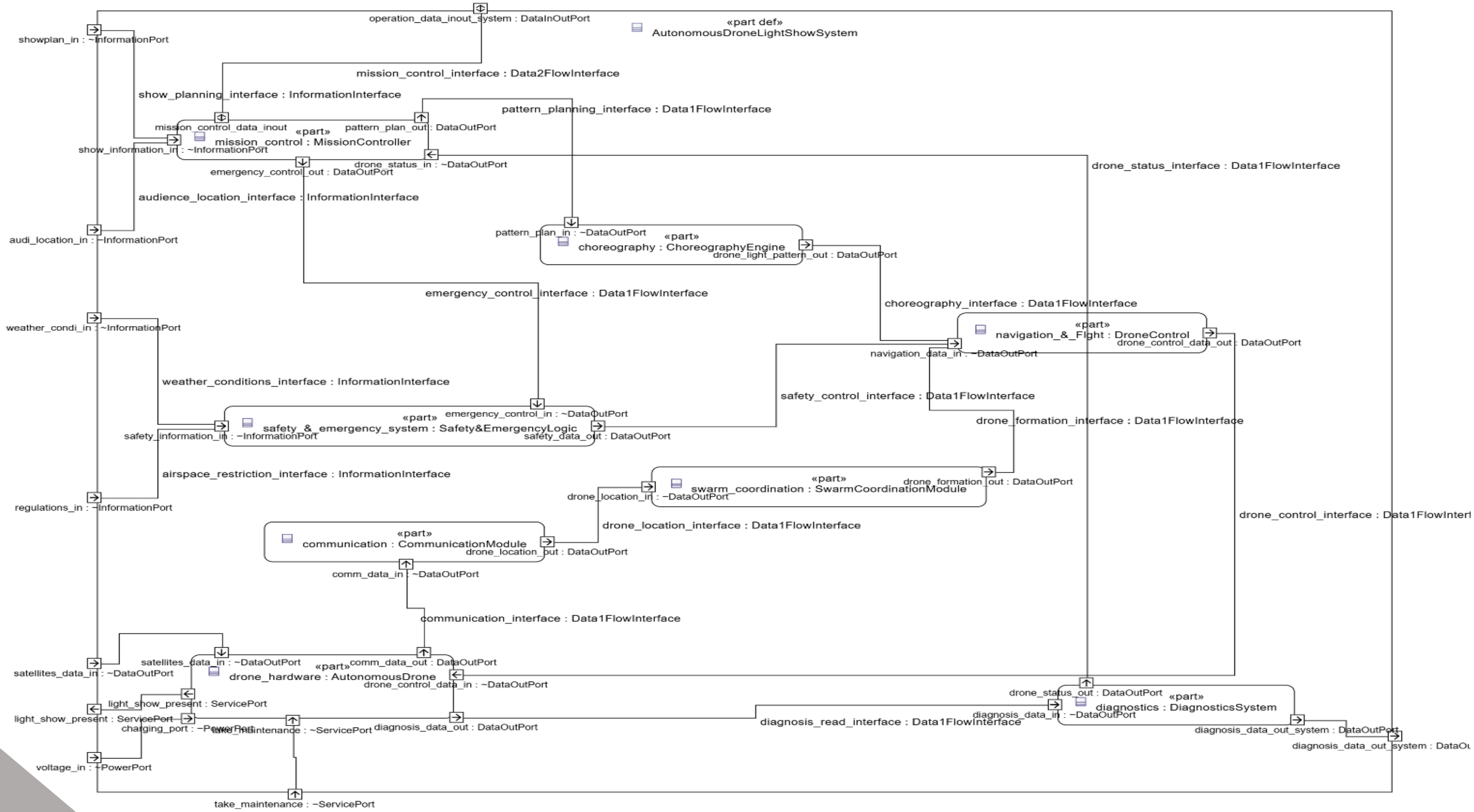
In SysML v2, ports and interfaces are used to show those interactions between the SOI and the external factors as well as the interfaces between internal components of the SOI. The following shows how the ports and interfaces are defined before being used.



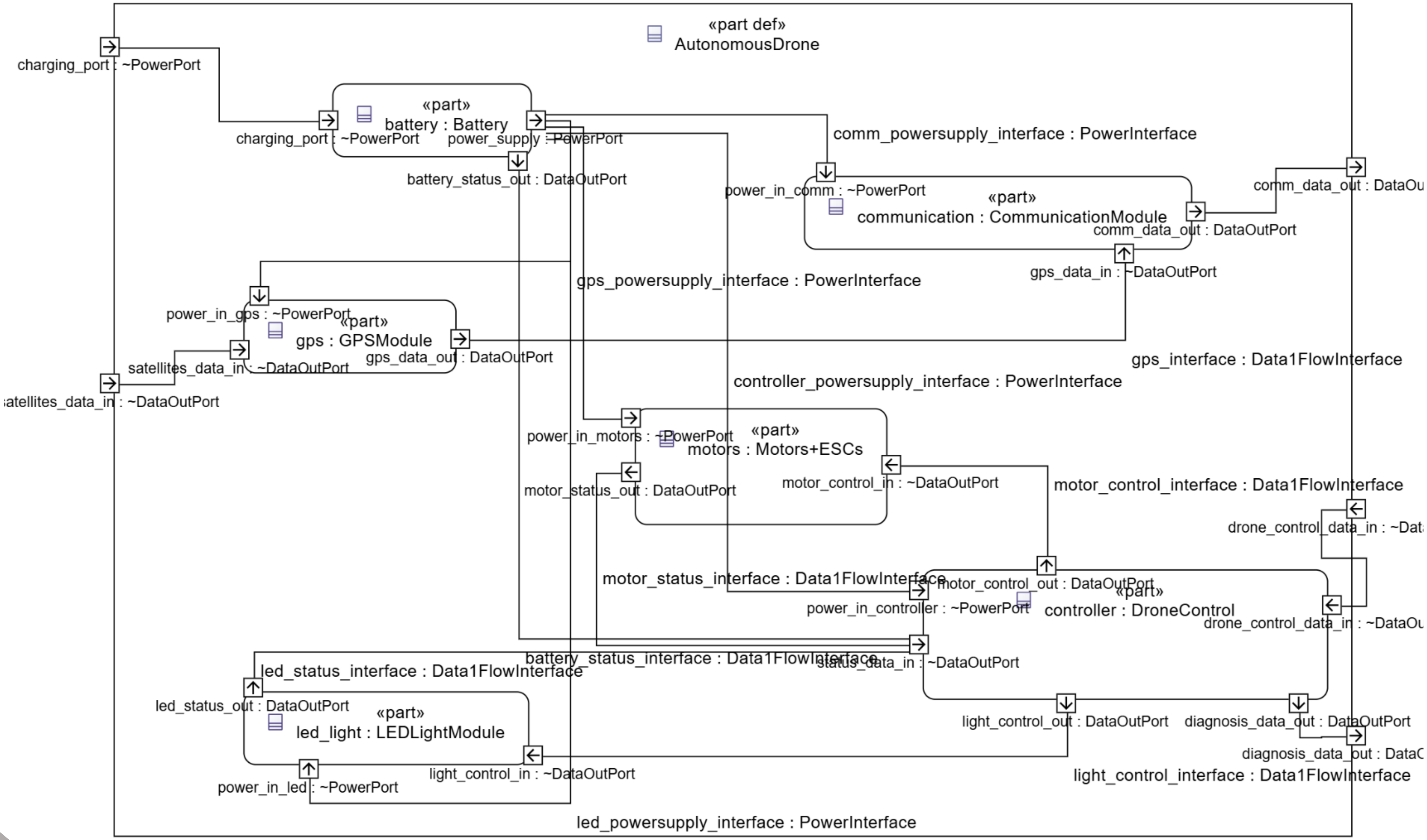
Performance Context (Black Box)



Interconnection View of SOI (White Box)

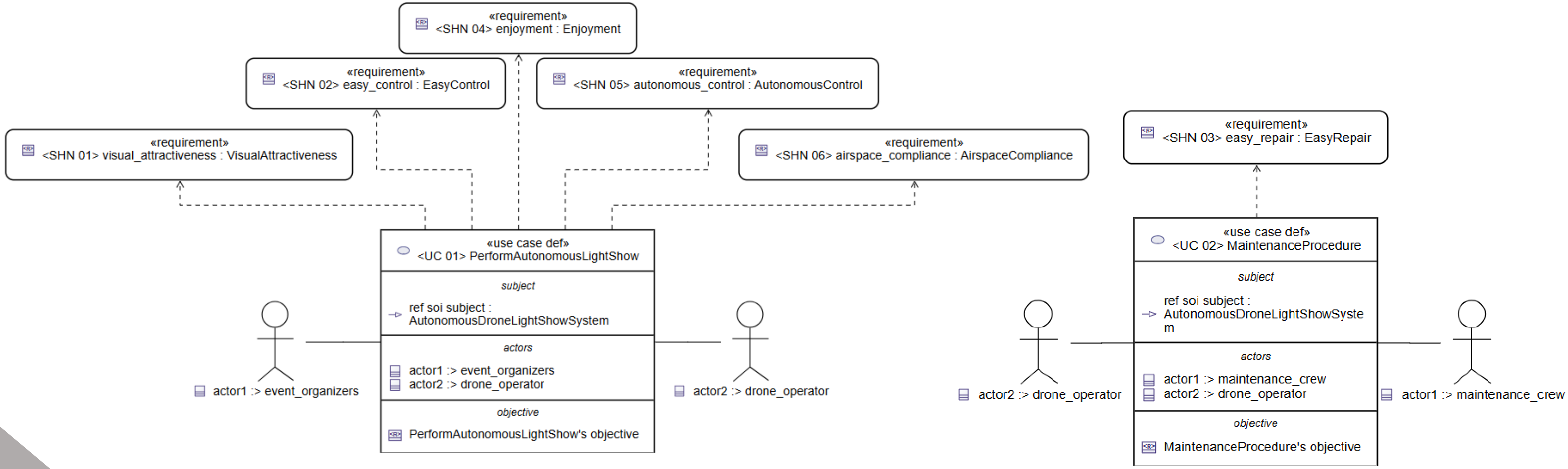


Interconnection View of Individual Drone



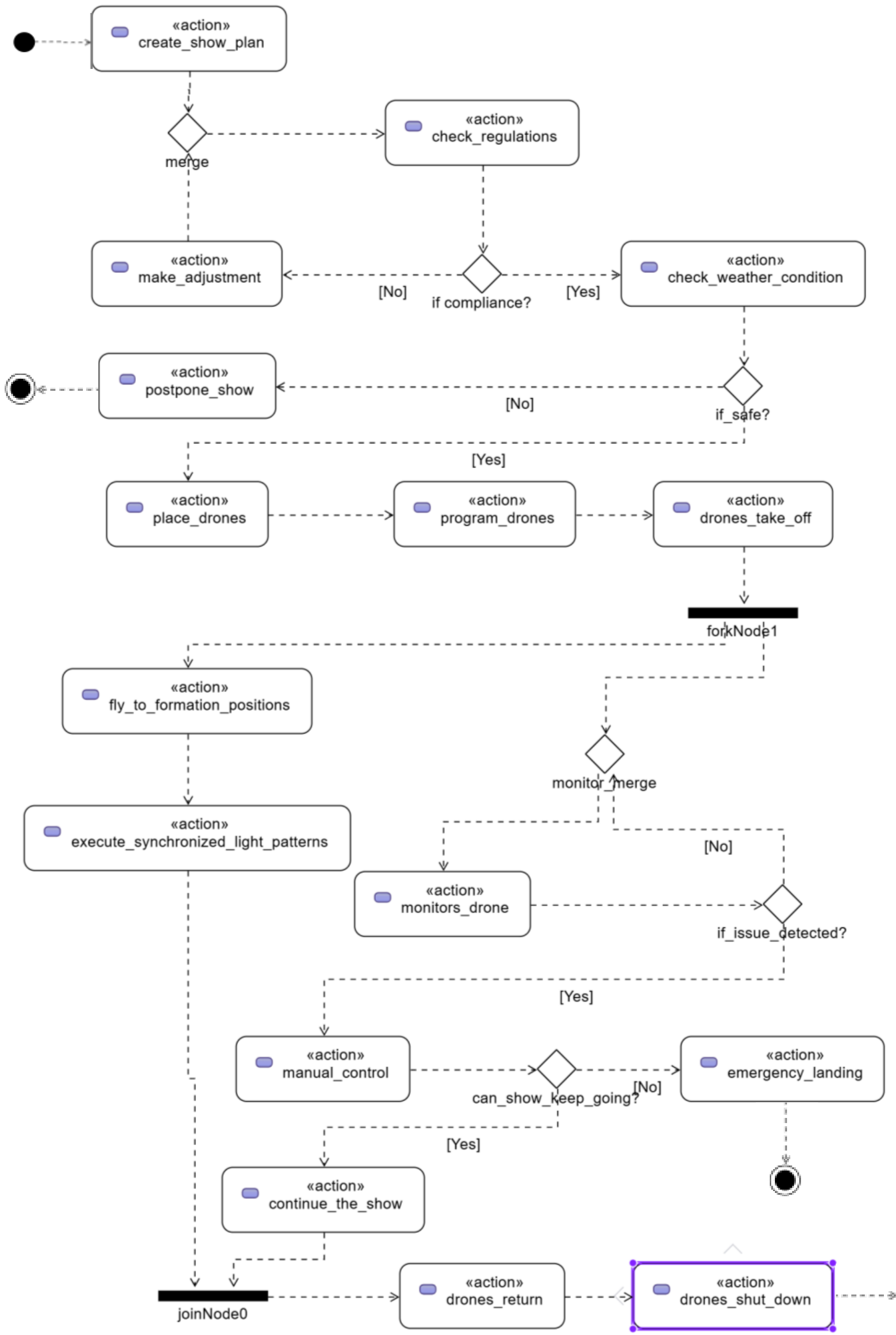
Use Cases

Use cases describe the main functions of the Drone Light Show System from the perspective of stakeholders. Each use case connects stakeholder needs to system behavior, showing how the system provides value and can be expanded into a scenario showing the sequence of actions. In this system, two example use cases have been defined, which are autonomous light show and maintenance procedure.



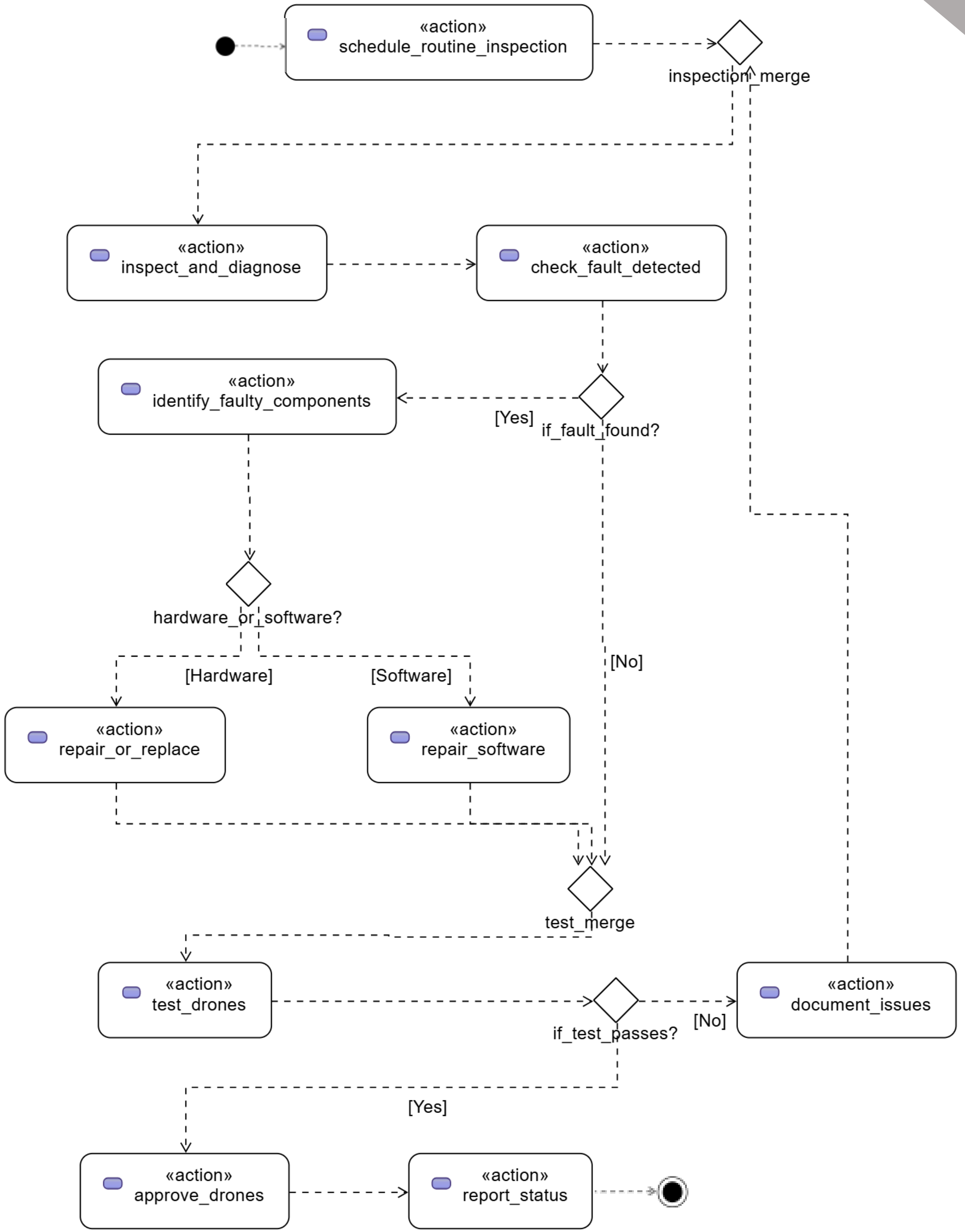
Scenario of Performing Autonomous Light Show

The autonomous light show starts with planning, preparing and programming the drones, followed by take-off and flying into formation. During the show, the drones perform synchronized movements and light patterns while being monitored. If needed, operators can take manual control for safety. After the show, the drones return and land in the designated zone.



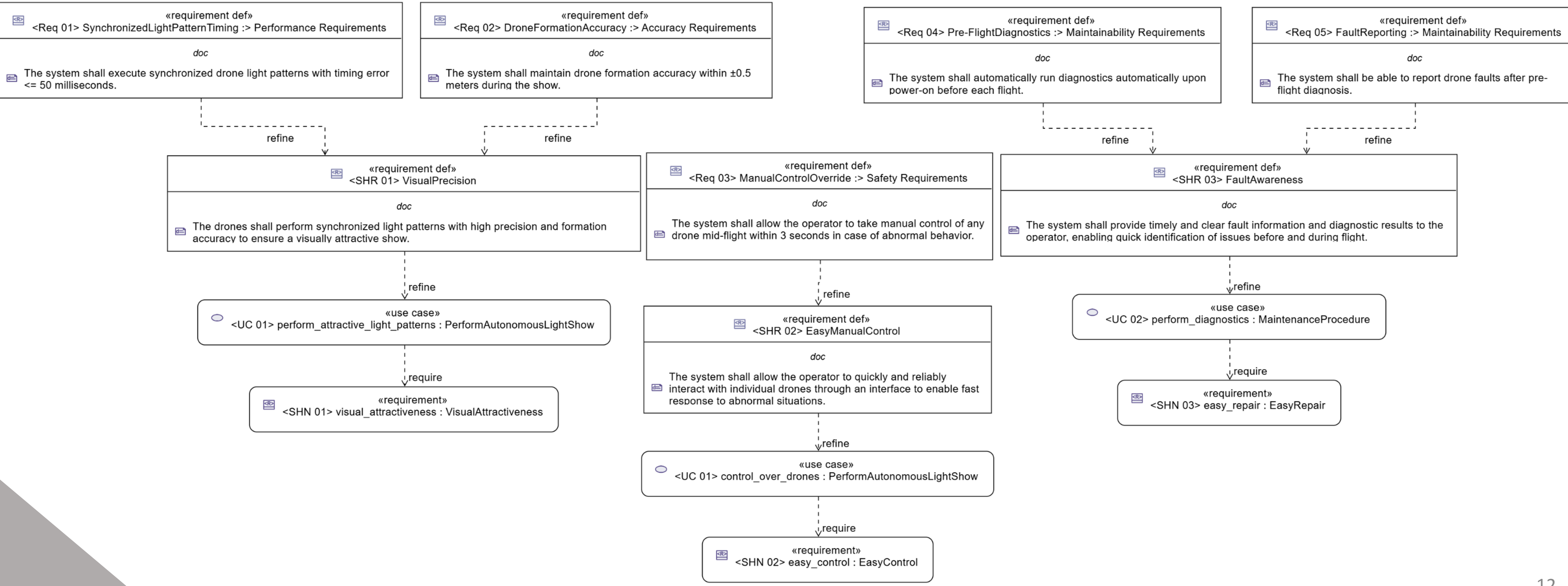
Scenario of Maintenance Procedure

The maintenance process begins with scheduling and inspecting the drones before and after a show. Each drone is diagnosed for faults, and if issues are found, software is repaired or hardware is replaced. The repaired drones are then tested, approved for future use, and their maintenance status is documented for traceability.

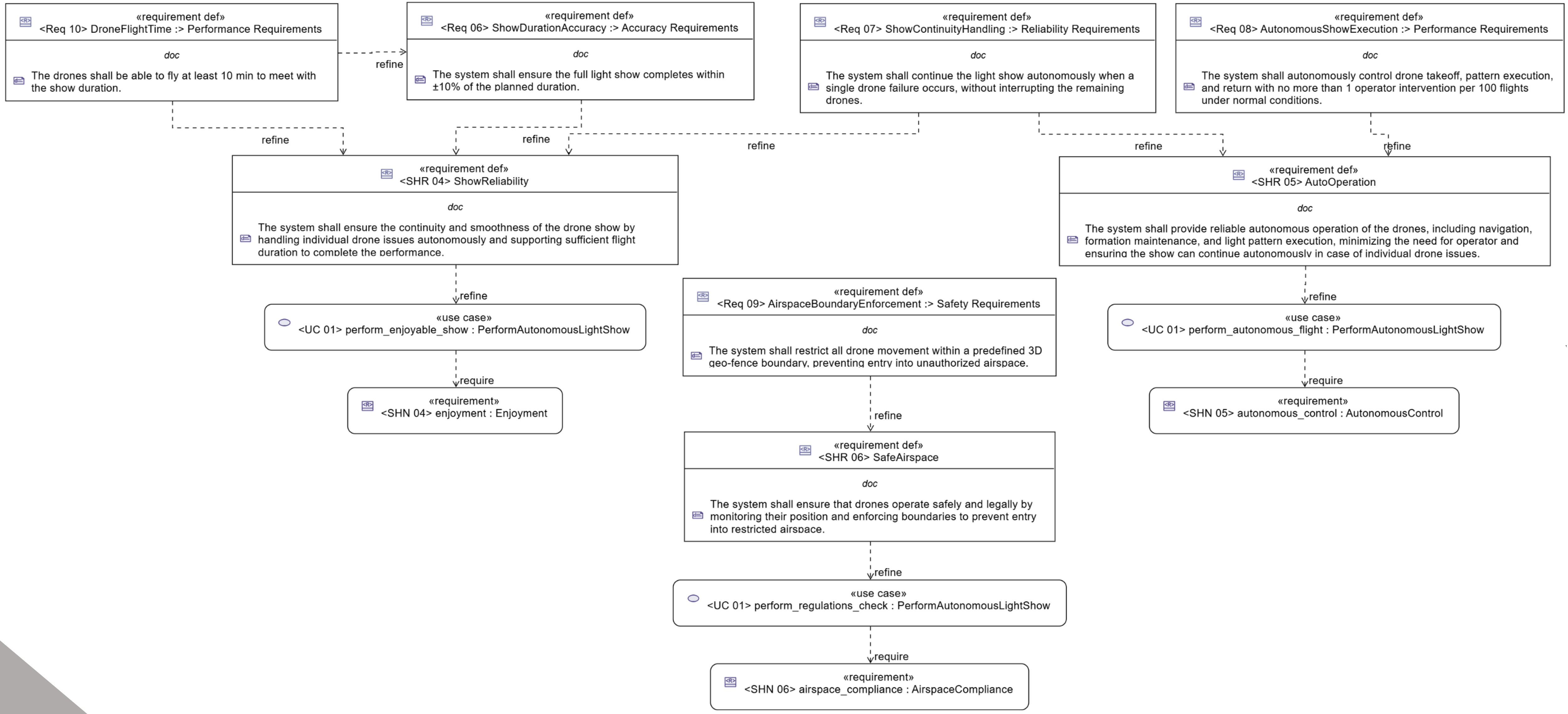


Requirements

Stakeholder needs are refined into clear requirements that the Drone Light Show System must satisfy. Each stakeholder requirement is mapped to one or more system requirements, ensuring traceability from needs to requirements.



Requirements



System Requirements Categorization

System requirements can be grouped into categories based on their purpose. This helps manage complexity, makes traceability easier, and ensures that all aspects of the Drone Light Show System are addressed, from performance to safety and compliance.

Categories	System Requirements	Functional or Non-Functional
Performance Requirements	<Req 01> Synchronized Light Pattern Timing	Non-Functional Requirements
	<Req 10> Drone Flight Time	
	<Req 08> Autonomous Show Execution	Function Requirement
Accuracy Requirements	<Req 02> Drone Formation Accuracy	Non-Functional Requirements
	<Req 06> Show Duration Accuracy	
Maintainability Requirements	<Req 04> Pre-Flight Diagnostics	Function Requirements
	<Req 05> Fault Reporting	
Reliability Requirements	<Req 07> Show Continuity Handling	Functional Requirement
Safety Requirements	<Req 03> Manual Control Override	Functional Requirements
	<Req 09> Airspace Boundary Enforcement	

State Machines

The drone light show system can be described using a state machine, which shows how drones move between different states during the show. Typical states include preparation, take-off, flying into formation, performing synchronized light patterns, and returning to land. The state machine also includes safety-related transitions, such as switching to manual control or emergency landing when needed. This helps us clearly understand drone behavior throughout the entire light show.

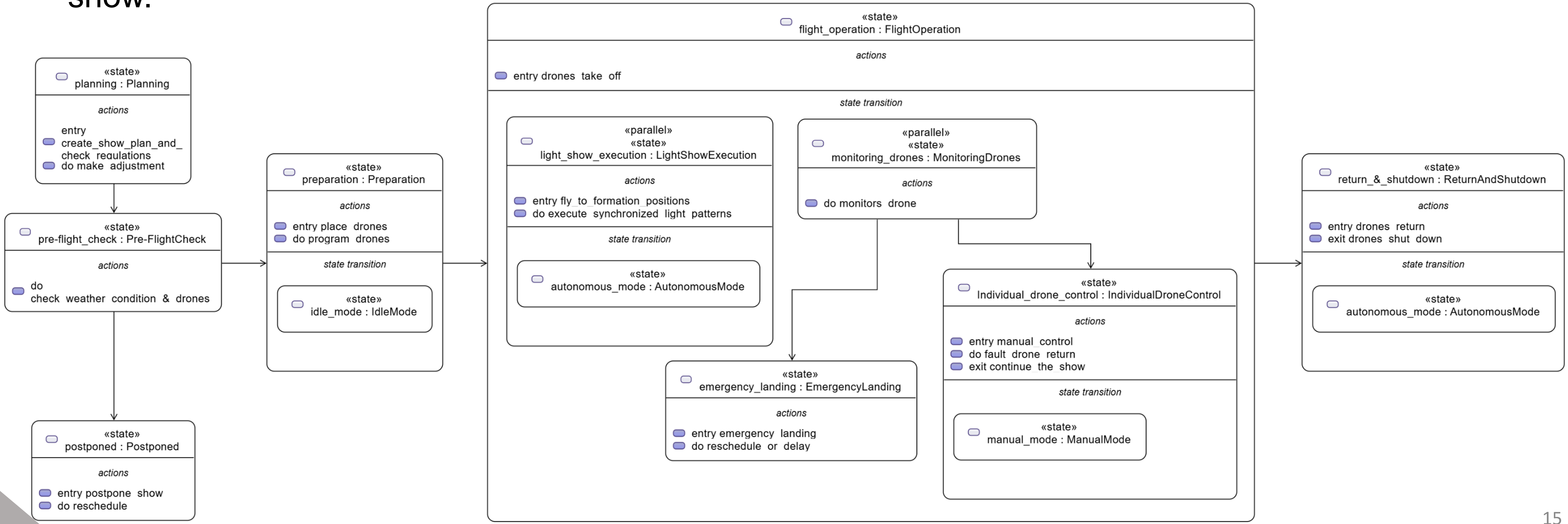
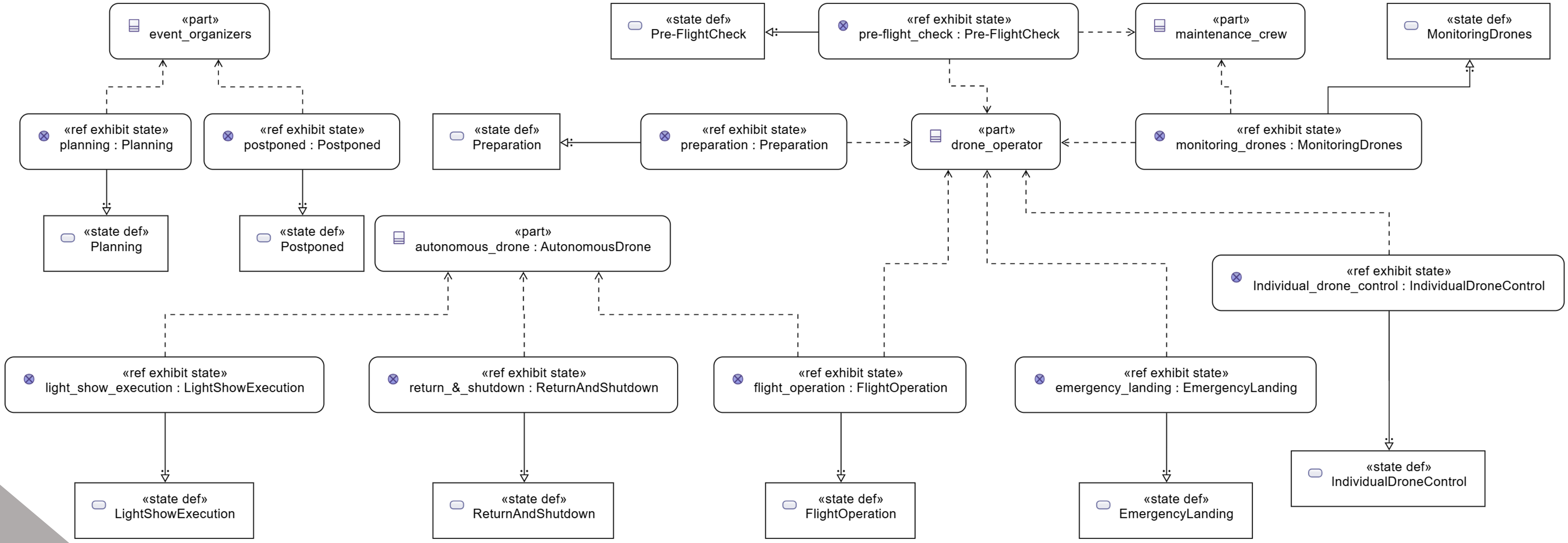


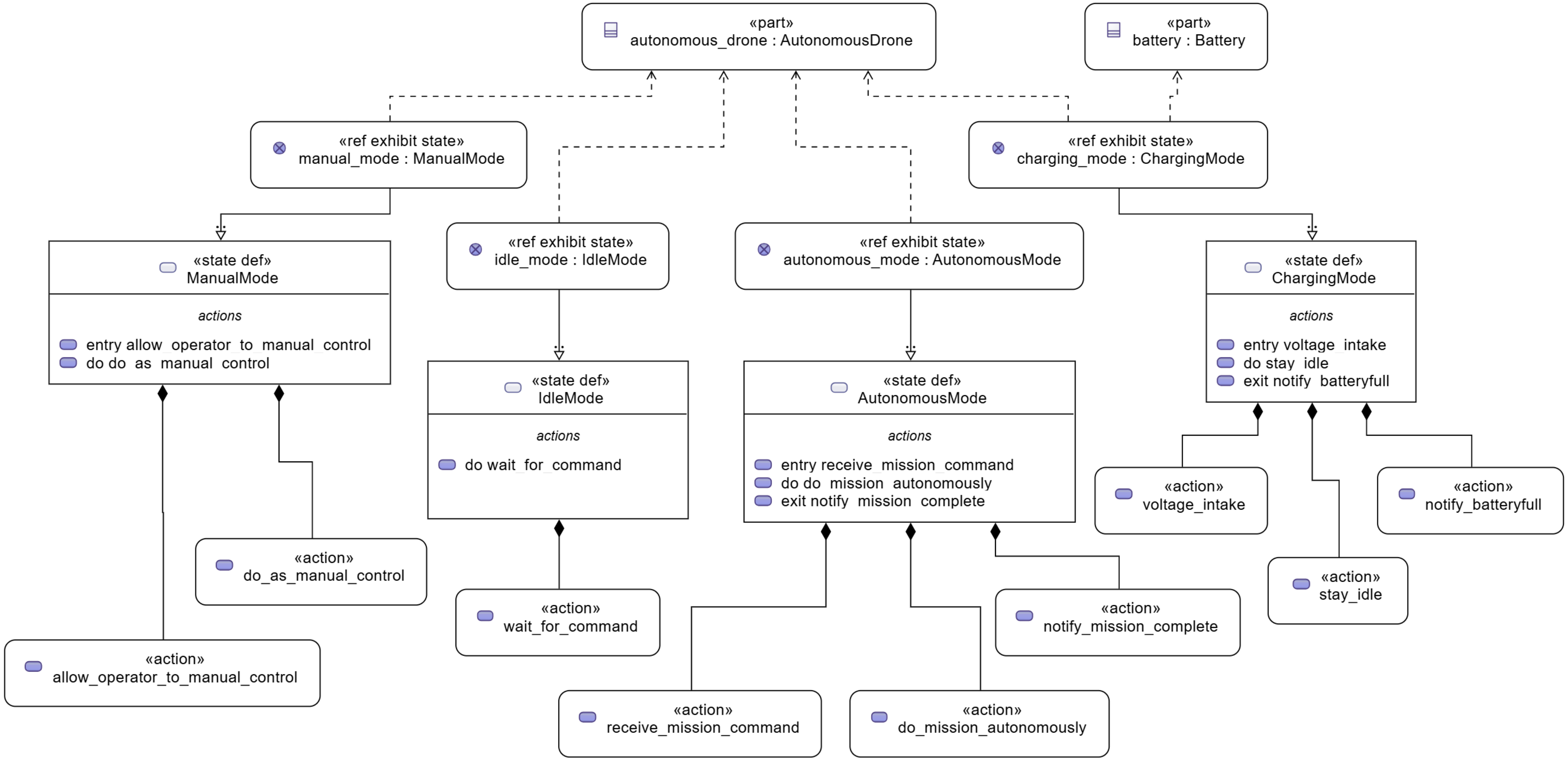
Exhibit States

Exhibit states link the states to the system parts showing which part of the system exhibits a given state as part of its behavior. For example, the drone hardware is active in flight-operation, the light modules are active during show performance. By mapping exhibit states to system parts, we can see how different components contribute at each step of the operation.



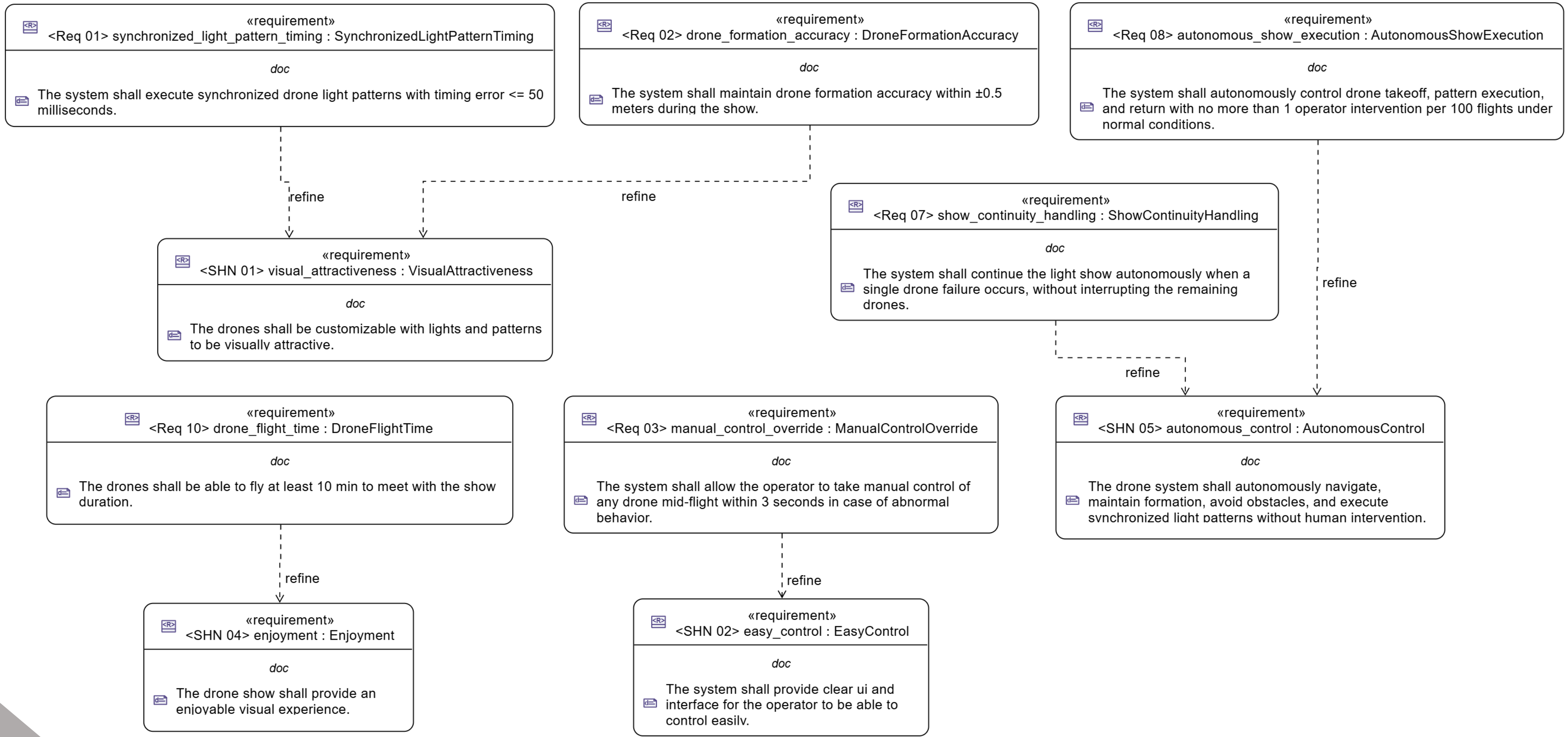
Drone Modes

State machines can also be used to represent the different operational modes of each drone. They show how drones behave in modes such as charging, autonomous flight, manual control, and idle.



Logical Layer

The logical layer of the Drone Light Show System shows how stakeholder needs are translated into system requirements. Using the refine relationship, we ensure that every need is addressed by one or more system requirements, supporting traceability and structured design.

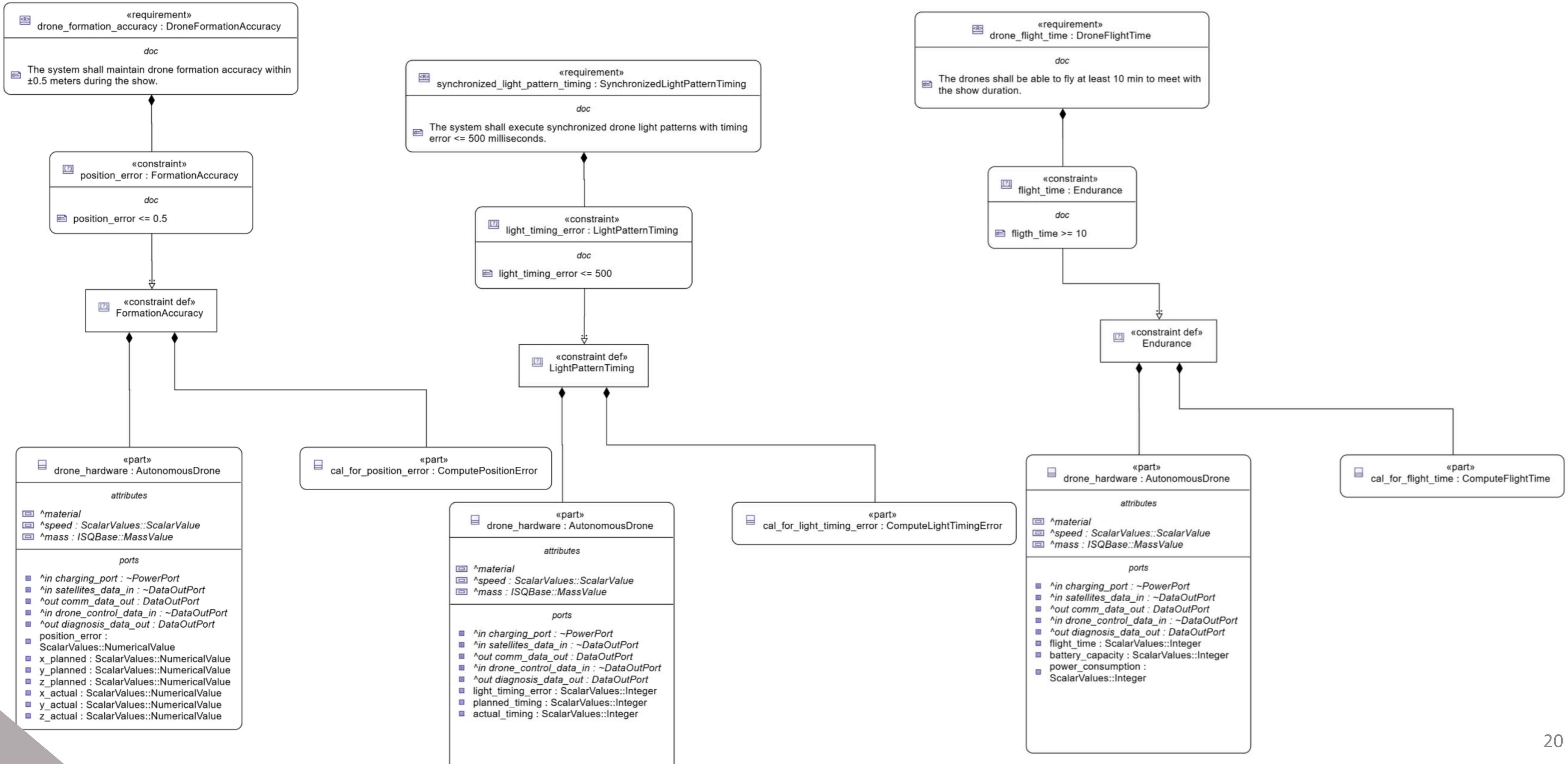


Constraints and Calculations

Constraints define limits on system behavior, performance, and operations, ensuring safety, compliance, and reliability. Calculations provide measurable values to guide system design. These constraints and calculations support analysis, optimization and verification of the system to improve performance and are also linked to non-functional requirements, ensuring that quality attributes like safety, reliability, and efficiency are met.

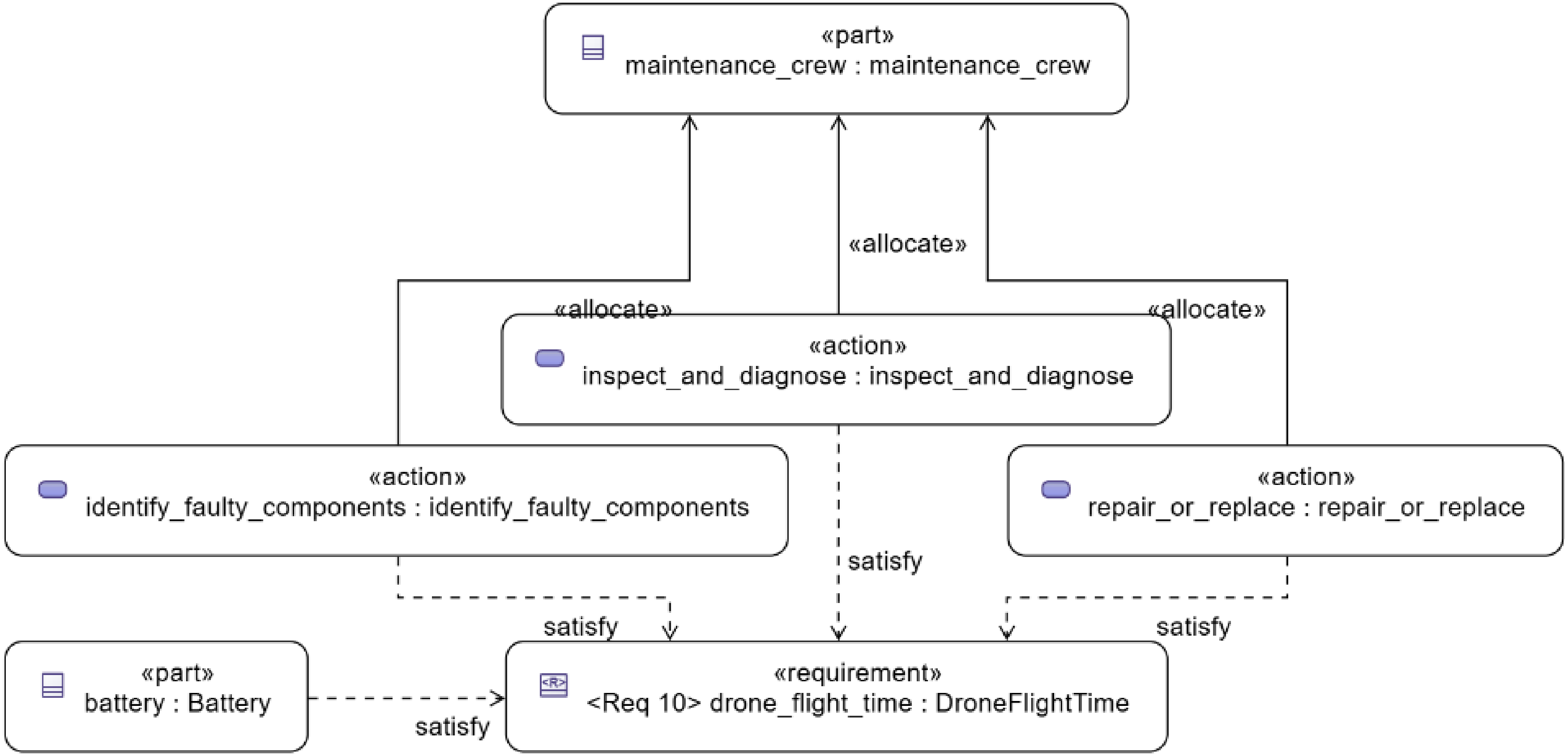
Constraint Name	Refined System Requirement	Input	Output	Mathematical Equation	Constraint
Light Pattern Timing	<Req 01> Synchronized light pattern timing (Non-Functional)	(p_t) planned timing (a_t) actual timing	Light timing error	light timing error = | a_t - p_t |	Light timing error <= 50 ms
Formation Accuracy	<Req 02> Drone formation accuracy (Non-Functional)	(x_p) x planned (y_p) y planned (z_p) z planned (x_a) x actual (y_a) y actual (z_a) z actual	Position error	position error = sqrt ((x_a - x_p)^2 + (y_a - y_p)^2 + (z_a - z_p)^2)	Position error <= 0.5 m
Endurance	<Req 10> Drone Flight Time (Non-functional)	(b_c) battery capacity (p_c) power consumption	Flight time	flight time = (b_c / p_c) * 60	Flight time >= 10 min

Constraints Definition

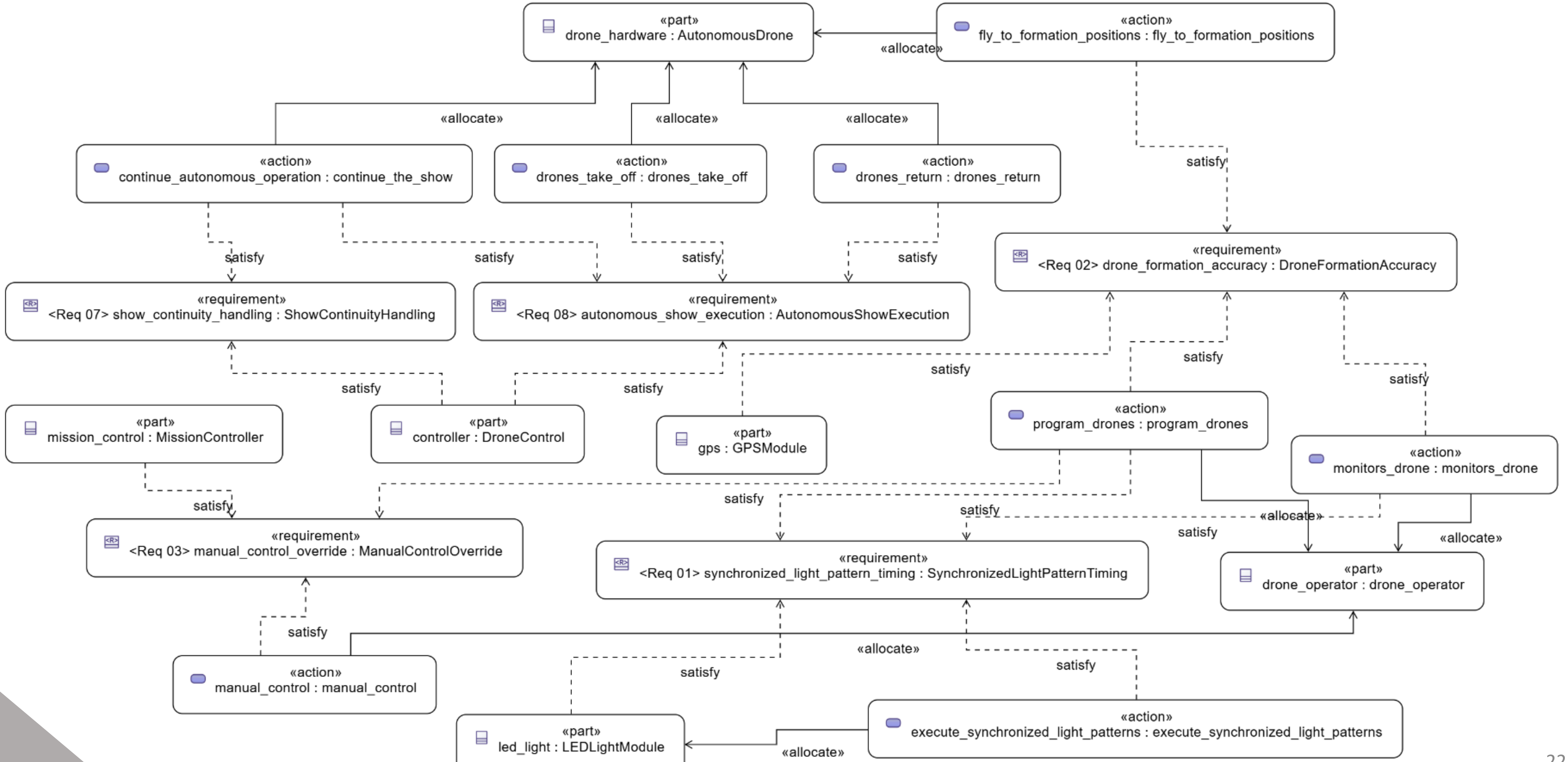


Satisfaction and Allocation

Satisfaction and allocation show how system requirements are fulfilled by the components and their actions. Each requirement is linked to the part and the action that satisfies it, supporting clear traceability.

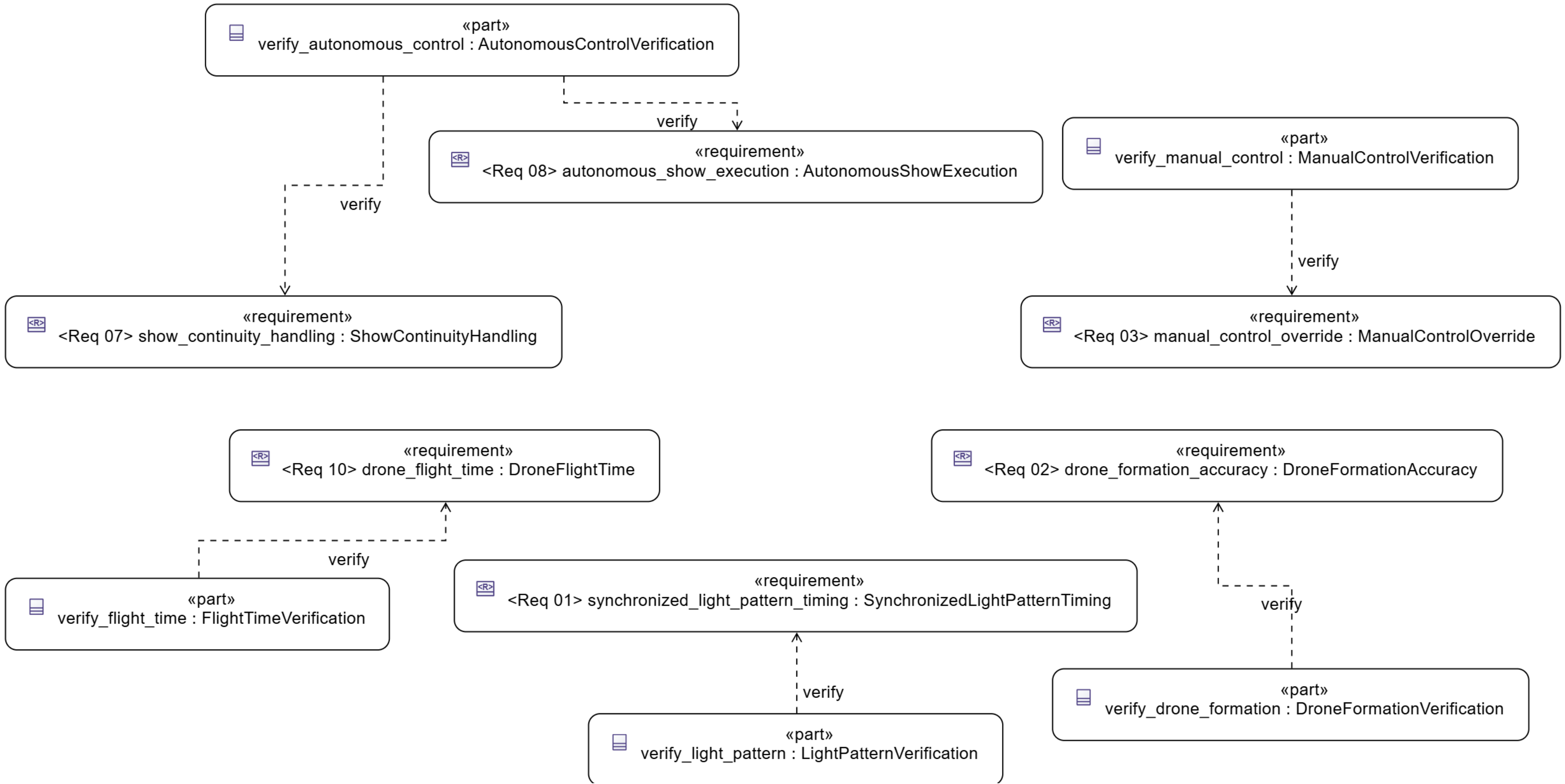


Satisfaction and Allocation



Verification Cases

Verification cases define how each system requirement will be tested to ensure it is correctly implemented. They link requirements to specific tests or actions, supporting traceability, and verification. Both functional and non-functional requirements can be verified using these cases.



Art takes flight on the canvas of the sky

Thank you for Your Attention

Any Question !