



# Reference Missile Functional Architecture (RMFA)

An architecting toolset to drive design and modelling

*Contributors: Andrea Caroni, Jason Gollaglee, Giulio Telleschi, Ivan Mactaggart, Pierre-Henri Pradel, Ed Willingham*

Presented by

Jason Gollaglee

Technical Lead – Systems Capability | RMFA UK Technical Lead

**MBDA**  
MISSILE SYSTEMS



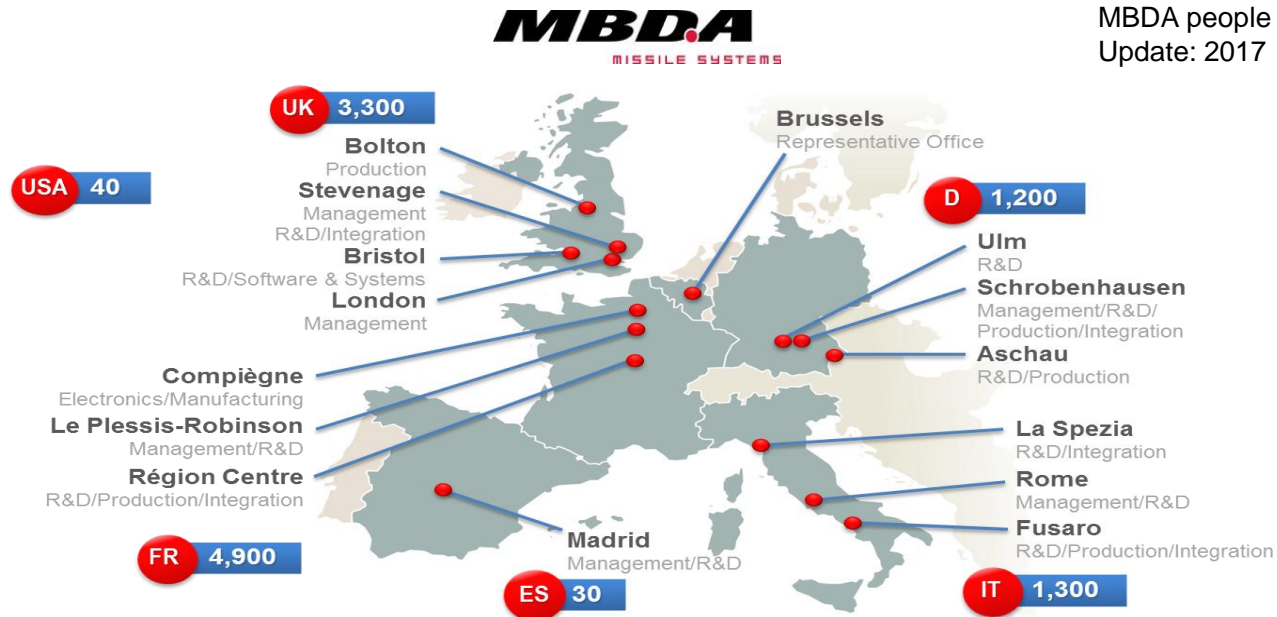
- **Context: Who are we?**
- **Needs: Why do we need it?**
- **Goals: What should it solve?**
- **Toolkit: What is it?**
- **Tuning: How do you use it?**
- **Conclusions**

## Context: Who are we?



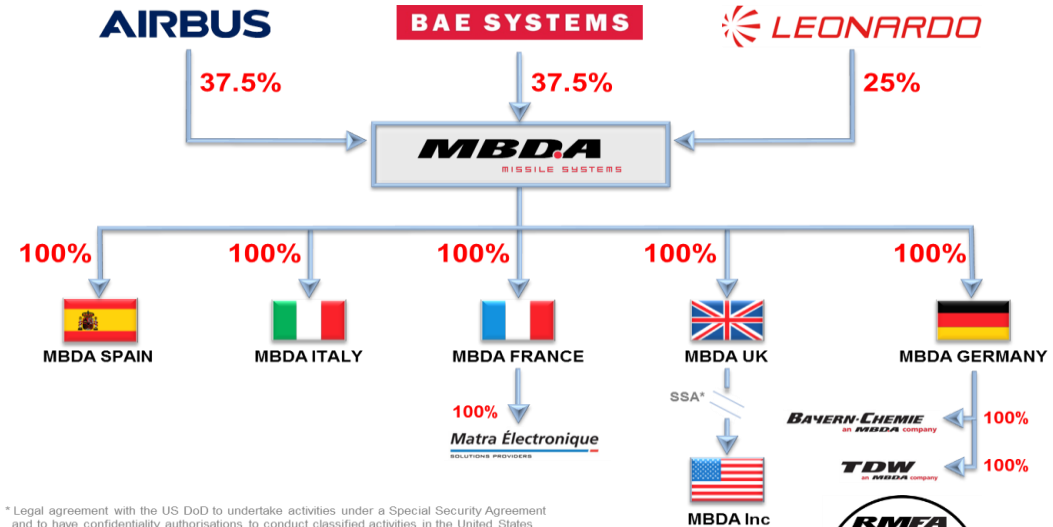
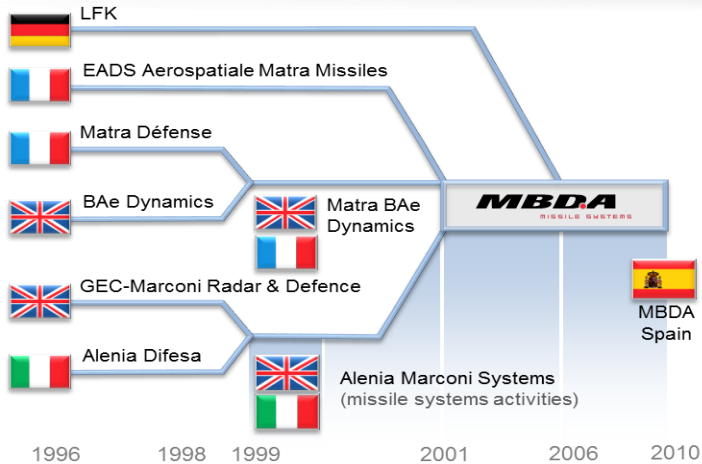
## MBDA, a European integrated defence company

MBDA is the first truly integrated defence company in Europe and the only European one able to provide missiles and missile systems for each branch of the armed forces, whether in the air, at sea or on land.





# MBDA's heritage mirrors the history of tactical missiles in Europe



\* Legal agreement with the US DoD to undertake activities under a Special Security Agreement and to have confidentiality authorisations to conduct classified activities in the United States





## Promoting and delivering cooperative programmes is deeply rooted in MBDA's DNA



STORM SHADOW / SCALP



METEOR



ASTER Family



CAMM, CAMM-ER



TAURUS KEPD 350



SEA VENOM / ANL



FC/ASW





- The authors form part of an international working group which develops the RMFA for use on MBDA projects.
- RMFA team members are system design & architecting practitioners coming from UK, France and Italy.
- Some responsibilities of the team within MBDA for RMFA are:
  - Architecture development;
  - Capability development;
  - Support and train engineers;
  - Promote RMFA companywide;
  - Provide feedbacks to RMFA from missile systems development.
- RMFA has been presented at INCOSE Italy Conference, and is included in CIISE 2018 proceedings.



## Needs: What drives it?





**R** eference  
**M** issile  
**F** unctional  
**A** rchitecture

- The Defence world is facing a big change → Processes to facilitate the transition
- Cooperative multinational programs → Protect information, enable sharing
- MBDA wide portfolio and number of customers → Many variants to be managed
- Need to manage increasing complexity in missile design → Systems engineering is more and more relevant. Risk reduction is a must





**R** eference  
**M** issile  
**F** unctional  
**A** rchitecture

- Unprecedented demand of capabilities and flexibility → Empower communication
- Missile interoperability is leading to a wider-ranging requirement set, Concepts of Operations (ConOps) and Concepts of Use (ConUse) → Move from a document-based approach to a model-based approach (broad meaning)
- Limited budget constraints → Provide modular, interoperable and cost-efficient solutions





- MBSE in MBDA empowers the advantages provided by cross-sharing and model consistency;
- MBDAAF, a legacy MBDA DAF;
- International working groups (e.g. RMFA, MBSE, Architecting);
- National Capability Teams to foster MBSE within each National Company (NatCo);
- Legacy guidelines and procedures.

**Limited sharing across NatCos due to restricted or classified information,  
therefore MBSE is tailored for each project-specific solution**



## Goals: What should it solve?





## Can we develop missile “building bricks”???



© LEGO





Missile....1 single word for many technical aspects

*When you read "Missile", what do you think of?*





## The role of the RMFA to act as a “reference” is:

- To share a common way of thinking for missile functional architecture across the company
  - Identify the required missile functional architecture;
  - Support the capture of the non-functional viewpoints like performances, safety or security in order to select the appropriate missile physical architecture;
  - Refine the equipment functional requirements and associated behavior and performances;
  - Avoid an excess of project-specific modelling;
  - Share internationally non-restricted architecture design best practices.
- To define a common approach for functional analysis;
- To share a common taxonomy for missile functional analysis and architecting;
- To enhance Knowledge Management.

**The RMFA is open, modular, robust, interoperable and cost-efficient**



RMFA has to encompass current and future missile developments in all fields that are part of MBDA portfolio, including access to platforms and customer engagement:

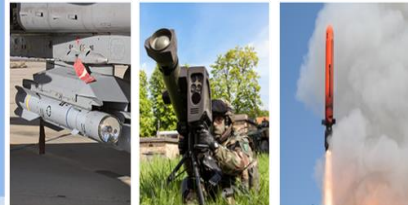
*How many missile-platform-customer combinations can you imagine?*

**Air defence**



Extended Air Defence/BMD	Ground Based Air Defence	Naval Based Air Defence	Airborne Air Defence
ASTER Block 1 NT	VL MICA	PAAMS/SEA VIPER	METEOR
SAMP/T	EMADS/ LAND CEPTOR	ASTER 15/SAAM	MICA
(PAAMS)	SPADA 2000	SEA CEPTOR	ASRAAM
TLVS	MCP/MPCV	VL MICA	MISTRAL ATAM
	MISTRAL	ALBATROS	AIRBORNE COUNTERMEASURES
	MANPADS, ALBI, ATLAS RC	SIMBAD RC	

**Surface attack**



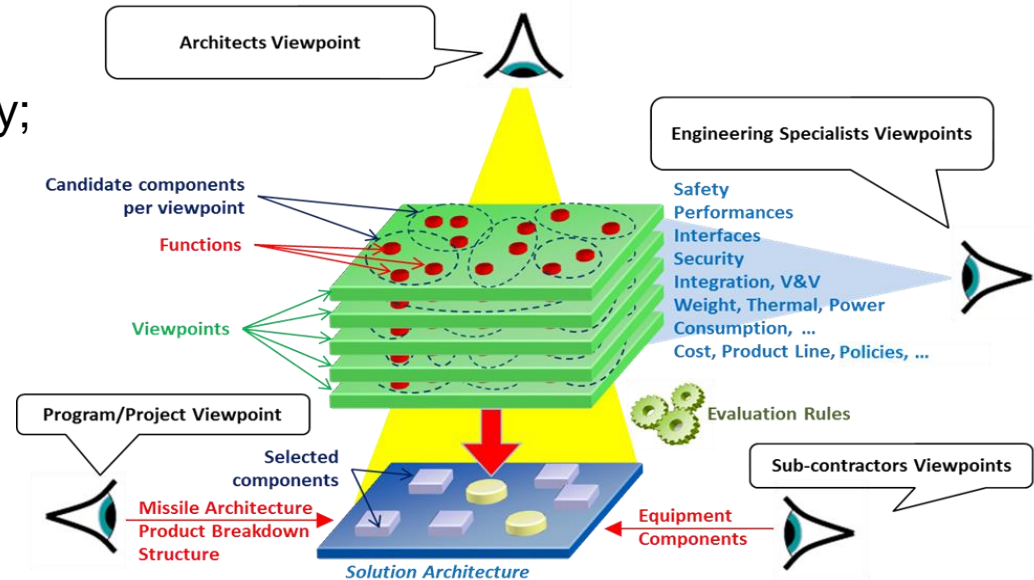
Airborne Surface Attack	Land Combat	Naval Combat
ASMP-A	MMP	NCM/MdCN
STORM SHADOW/ SCALP	MILAN ER	EXOCET TESEO/OTOMAT MARTE MILAS
TAURUS KEPD 350	ERYX	COASTAL BATTERIES
SPEAR	ENFORCER	SEA VENOM/ANL
BRIMSTONE	BATTLEFIELD COUNTERMEASURES	
PARS 3 LR		





## The system architecture design has to face many challenges:

- Multi-viewpoint engineering activity;
- Functions are cross-viewpoints;
- 1 source of truth;
- Early and robust validation of the solution architecture.



The main goal of the RMFA and its joint methodology is to facilitate this multi-viewpoint architecture selection process

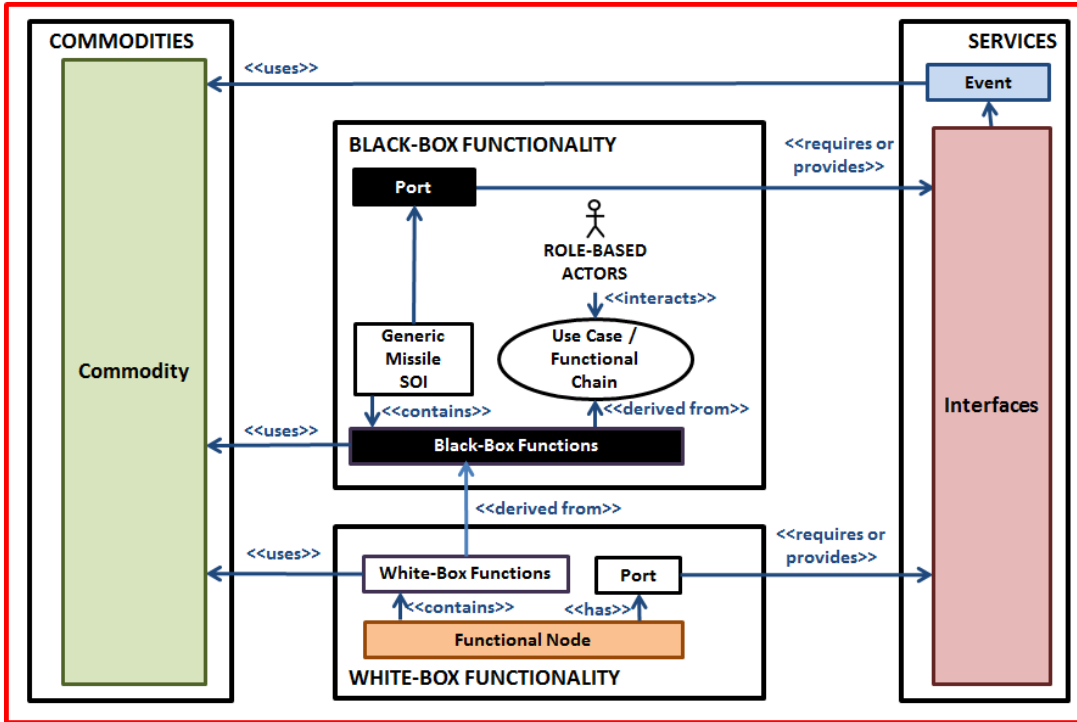
## Toolkit: What is it?



Context

RMFA Missile





- RMFA uses a small set of model entities and the relationships between them are described within a meta-model.
- The framework is based on:
  - Missile Use Cases;
  - Functional Chains Analysis;
  - Missile Functions;
  - Missile Services (Interfaces);
  - Information/Data Model;
  - Missile Sub-System (Equipment) Functions captured within Functional Nodes;
  - Architecture Examples.





High level use cases to embrace the wide portfolio of products:

- Goals and boundaries of the generic missile
- Full-lifecycle (operational and non-operational use cases)
- Operational use cases: 4 main use cases, around 15 total
- Standardized format:
  - Goal;
  - Actors;
  - Pre-conditions;
  - Trigger;
  - Perceived functionality (described with the RMFA Functional Chains);
  - Success Post-Conditions;
  - Alternative Paths.





Functional analysis is an activity that is frequently squeezed within projects, the RMFA provides a quick and validated tool:

- Group missile Functions with a goal-driven approach;
- Compact information;
- Strong link with Use Cases and equipment;
- Available at Black-Box e White-Box levels, with components that can be reused at both missile and equipment levels.

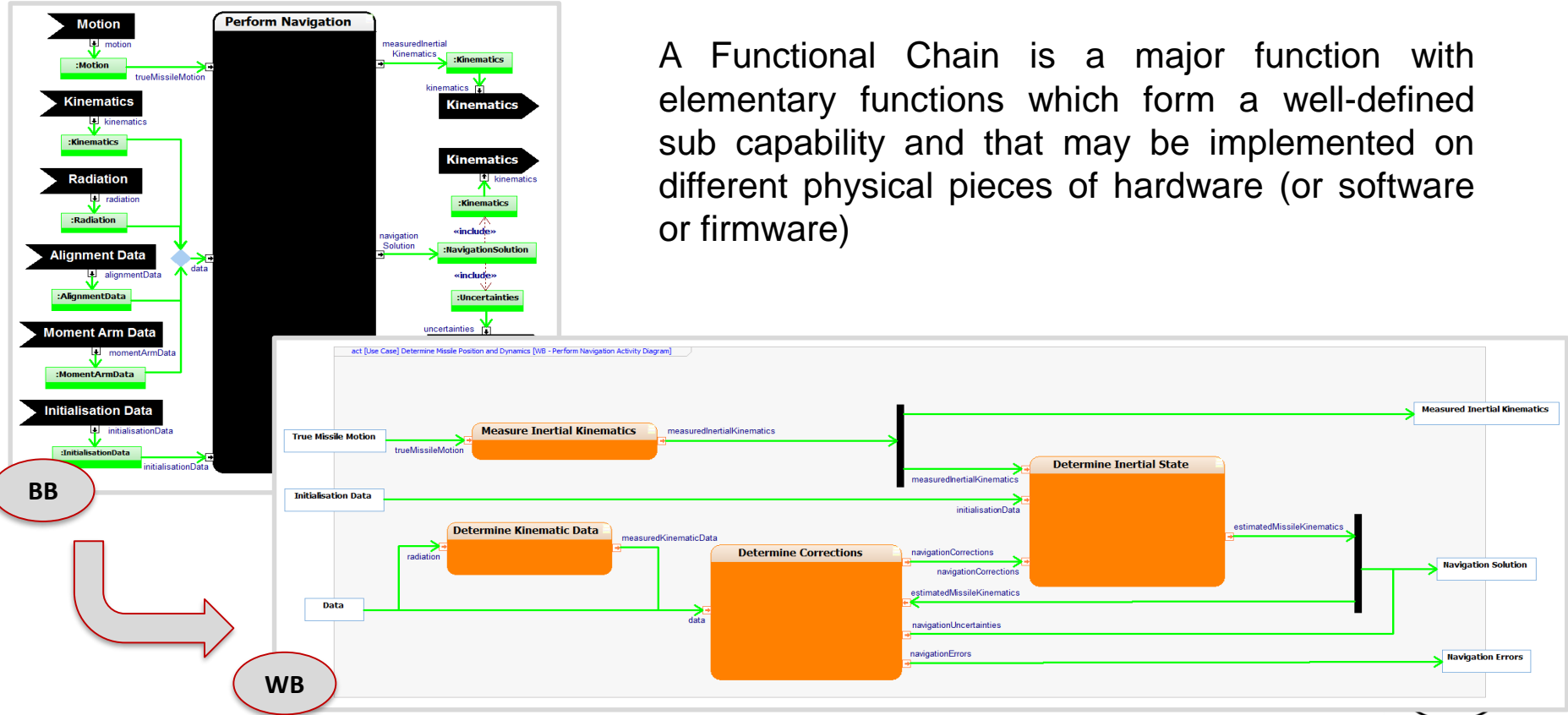
**It has been created a super-set of missile functionalities.  
It is valid in any missile domain, ranging from air-to-air to anti-ship missiles**





# The Toolkit - Functional Chains

A Functional Chain is a major function with elementary functions which form a well-defined sub capability and that may be implemented on different physical pieces of hardware (or software or firmware)

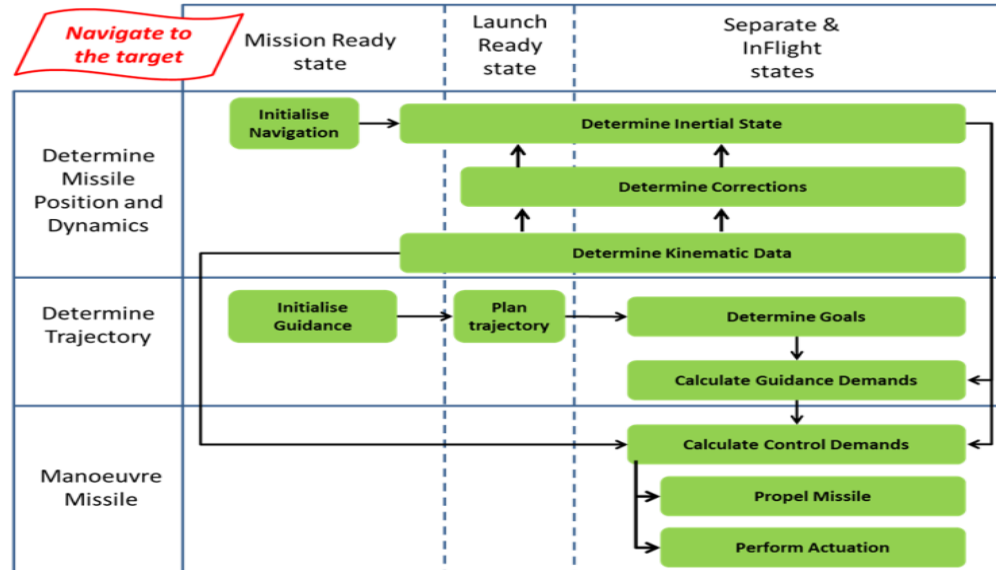


BB

WB



Functional Chains activation depends on time (i.e. mission evolution) and is mapped through the allocation of Functions to Missile States for each Use Case



Challenge: not natively supported by the modelling tool in use







The data model includes the abstraction of the following elements, to be tailored for each specific project:

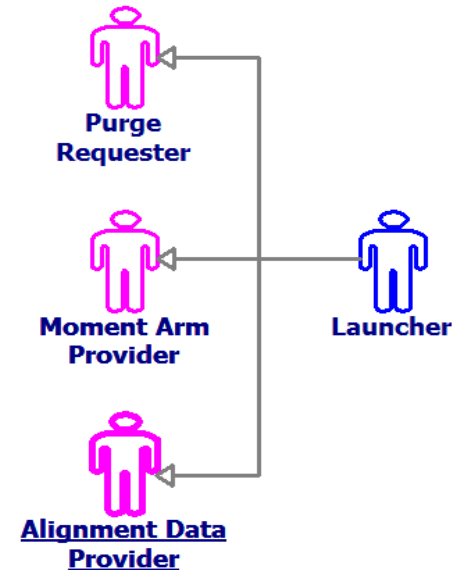
- Functional Interfaces;
- Commodities:
  - Standardized data/information to be exchanged.

Commodities are passed across interfaces and processed by functions



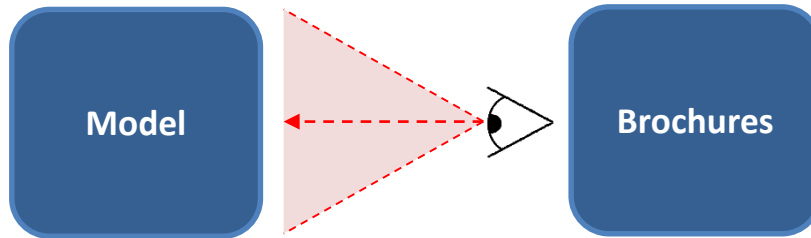
## Role-based actors are defined in order to maximize the correct allocation of interfaces and commodities:

- Each Actor (in blue) will inherit the properties of the associated role-based actors (in purple);
- Interfaces and commodities are allocated straightforwardly for each tailored project;
- Generalization allows to easily manage different configurations, providing consistency within your product-line architectures;
- Easily maintainable solution: changing the relationship, the functionality is reallocated to another Actor or removed.





## Reference Architecture





### Content:

- Additional context
- View into RMFA
- Methodology
- Architecture patterns (Examples)

### How to Brochures:

- Introduction
- RMFA Use Cases
- How To Use RMFA Actors

### Functional Chain

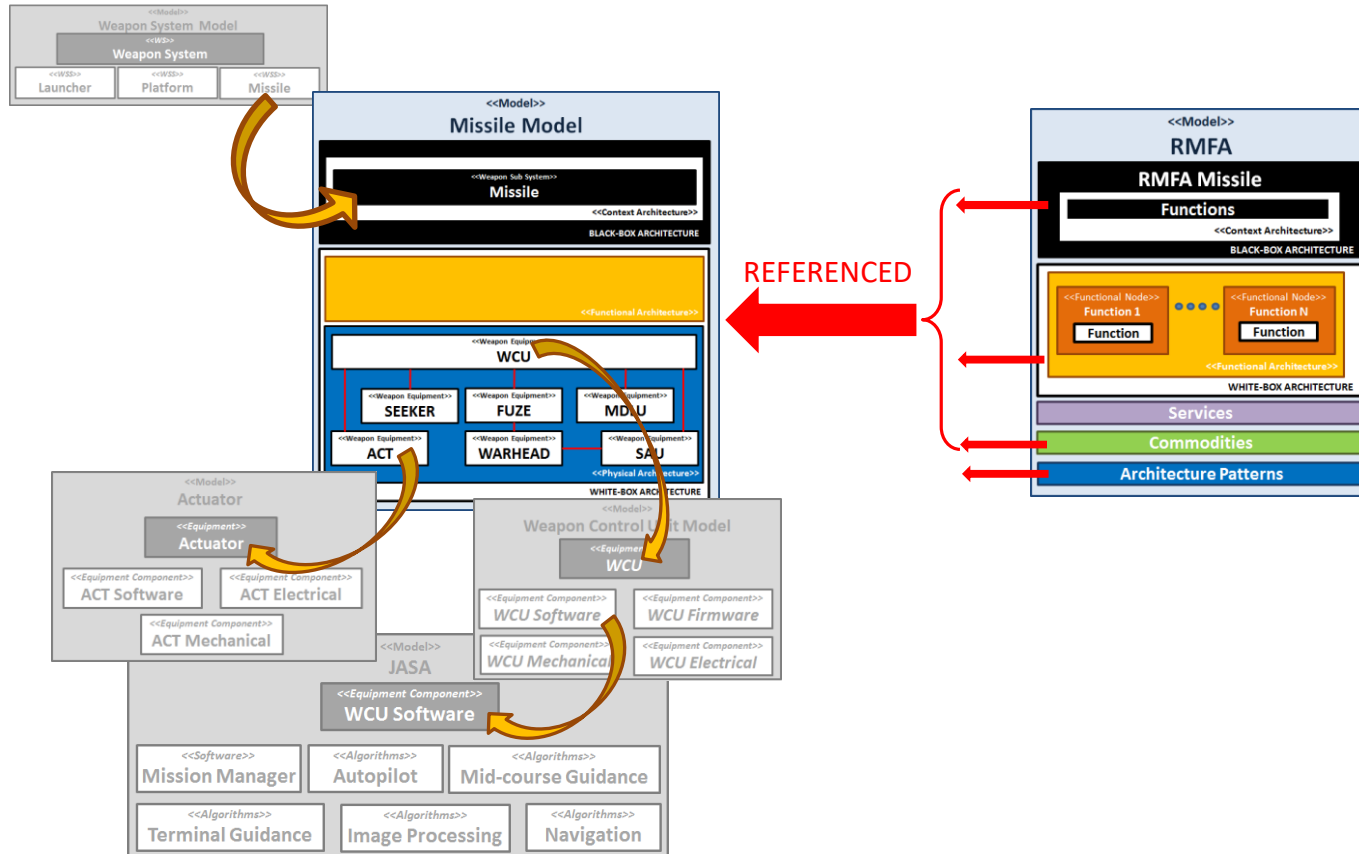
- Total of 10 brochures
- One for each functional chain

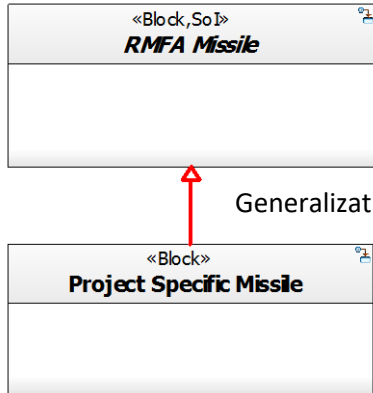
# Tuning: How do you use it?





# Reference Architecture





Generalization

**Inheritance**

Block : Project Specific Missile in Project Specific Missile

General | Description | Attributes | Flow Properties | Operations | Ports | Flow Ports | Full Ports | Proxy Ports | Constraints | Relations | Tags | Properties

Show Inherited

Name	Visibility	Return Type	Feature Direction	Defined In
performNavigation	Public	void		GMFA Missile
performGuidance	Public	void		GMFA Missile
performFlightControl	Public	void		GMFA Missile
determineEffectTrigger	Public	void		GMFA Missile
armEffect	Public	void		GMFA Missile
deliverMissile	Public	void		GMFA Missile
propelForObjects	Public	void		GMFA Missile
findAndSelectTrack	Public	void		GMFA Missile
trackAndHomeTarget	Public	void		GMFA Missile
erase	Public	void		GMFA Missile
purge	Public	void		GMFA Missile
power	Public	void		GMFA Missile
identification	Public	void		GMFA Missile
performSecurityMana...	Public	void		GMFA Missile

Locate OK Apply

**Override**

```

classDiagram
    class Project_Specific_Missile["«Block»  
Project Specific Missile"]
    class Identification["identification(identificationRequest,identificationRequest,identificationStatus):void"]
    class LoadMissionData["loadMissionData(engagementPlanningData,engagementPlanningData,missionData):void"]
    class Power["power(powerDemand,powerStatus):void"]
    Project_Specific_Missile --> Identification
    Project_Specific_Missile --> LoadMissionData
    Project_Specific_Missile --> Power
  
```

Implement Base Classes

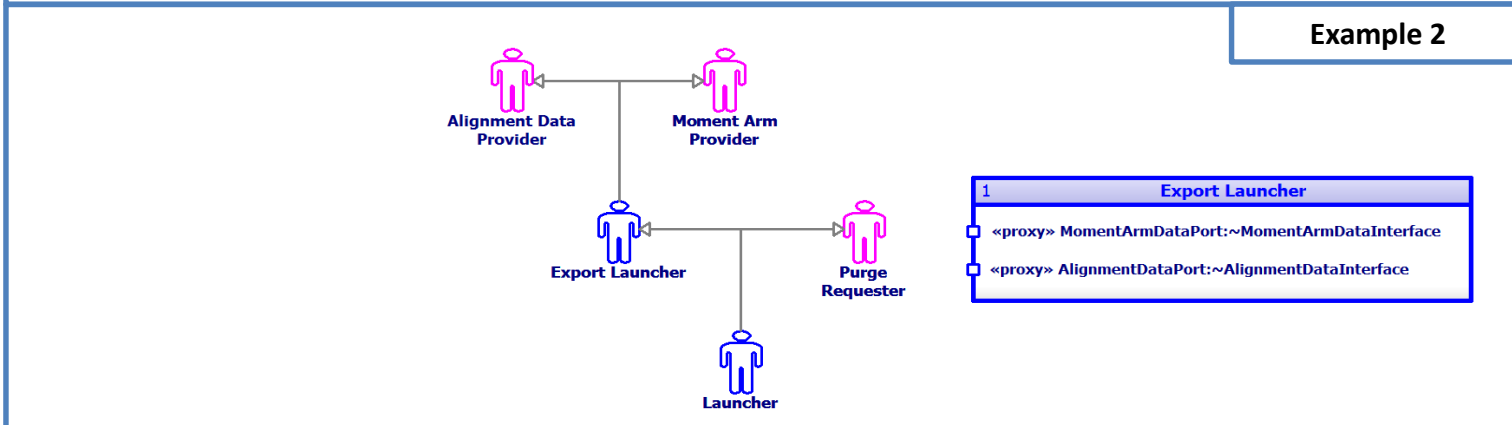
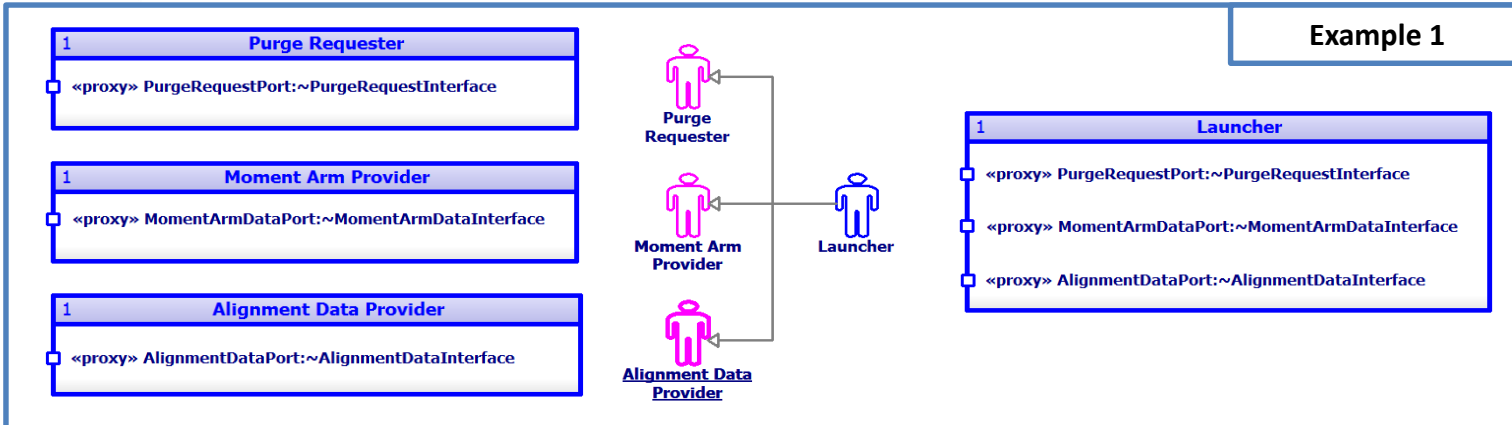
Select the elements to be implemented by class Project Specific Missile :

Filter View :  Required  Optional  Implemented

Automatically show this window

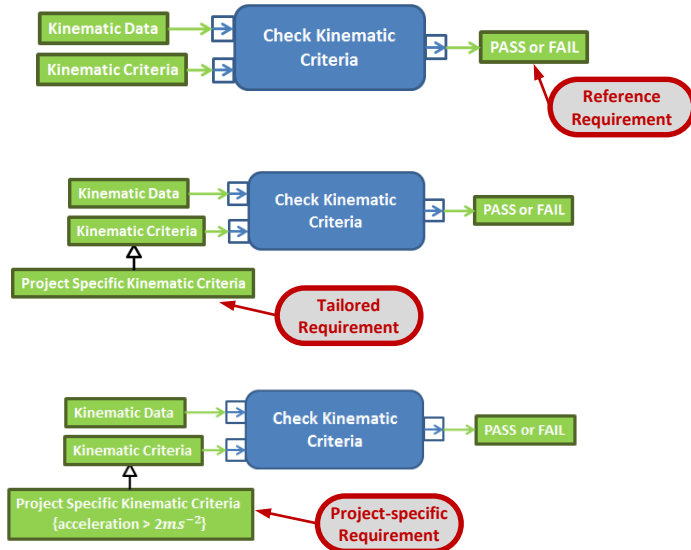
OK Edit Code... Cancel Help







RMFA functionality has been designed to allow the tailoring of the functionality through the tuning of the RMFAs function parameters:



- 1) Original function within RMFA
- 2) The project references the function and commodities from RMFA. Any commodities that need to be 'tuned' are specialised using a generalisation relationship
- 3) The project completes the tuning by populating the detail of any specialised commodity

No modifications to RMFA shall be needed when tuning to specific project. Requirements may come from RMFA or from the tailoring process.

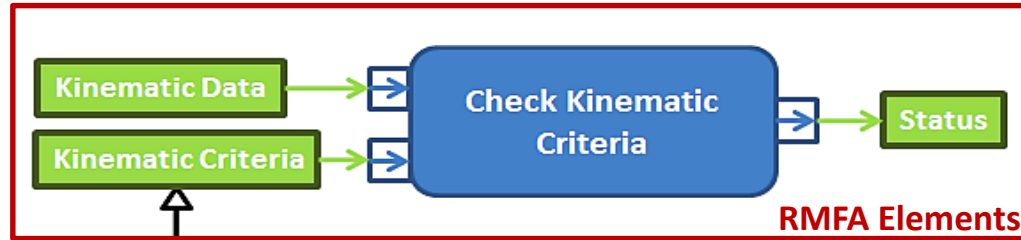




## Allows to focus on design rather than modelling techniques

- Contains elements which can be referenced by projects and ‘tuned’ to create a specific architecture

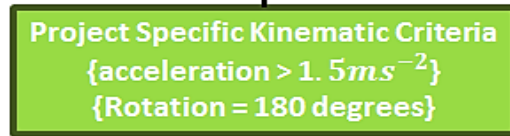
150% model from which a subset is selected for each application



Project-specific Requirement



**Project 1  
Specialisation**



**Project 2  
Specialisation**

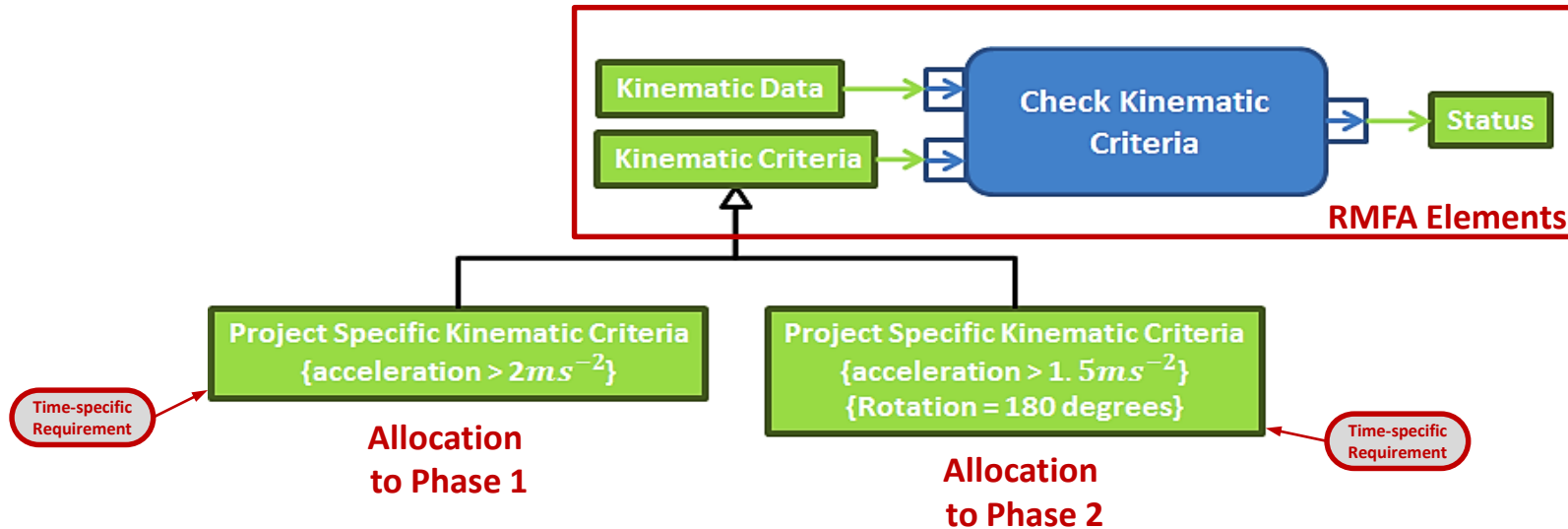
Project-specific Requirement

Requirements are tuned for each project



## Requirements allocation with respect to time

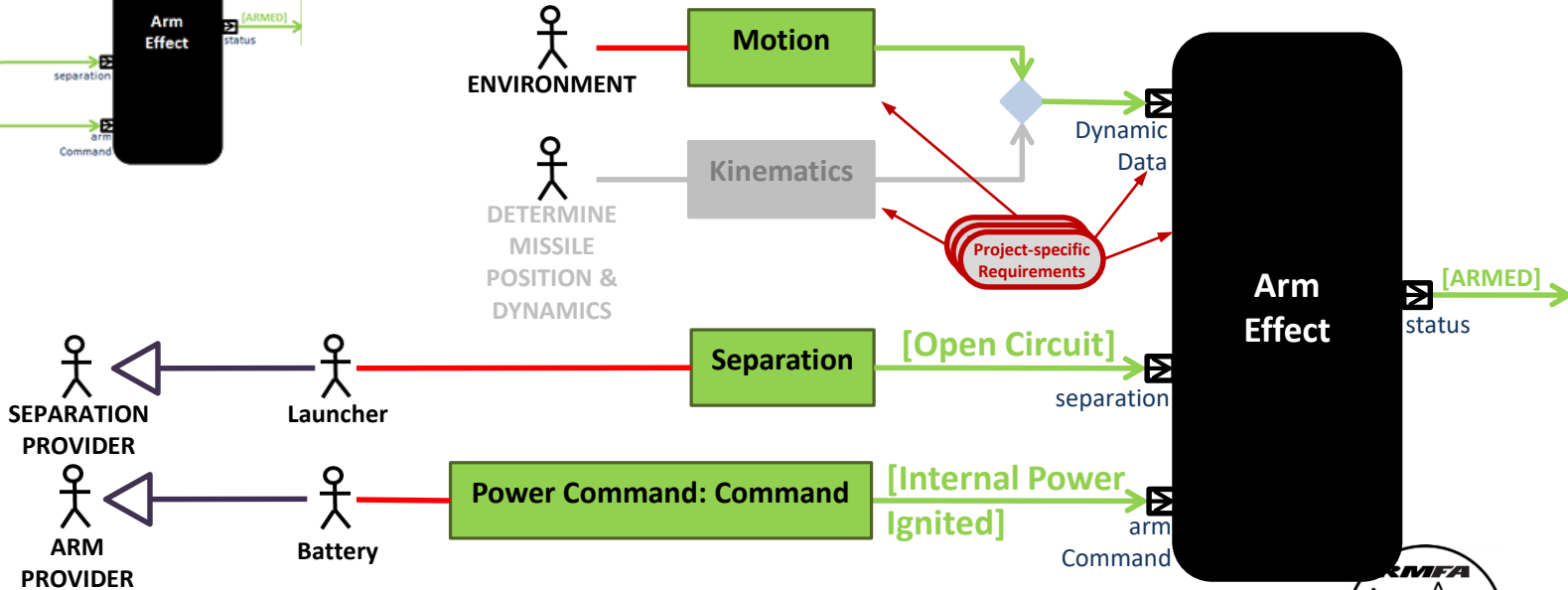
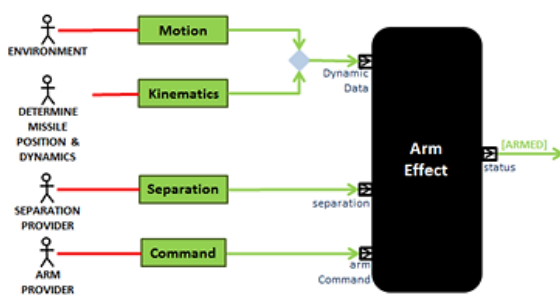
- Allocation of requirements to phases in a compact view



Compact allocation of requirements respect to time/phases

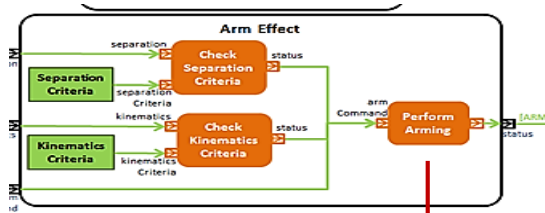


# Tuning to Project - Example



Removing a function → requirements allocation to functions, interfaces, equipment, ...





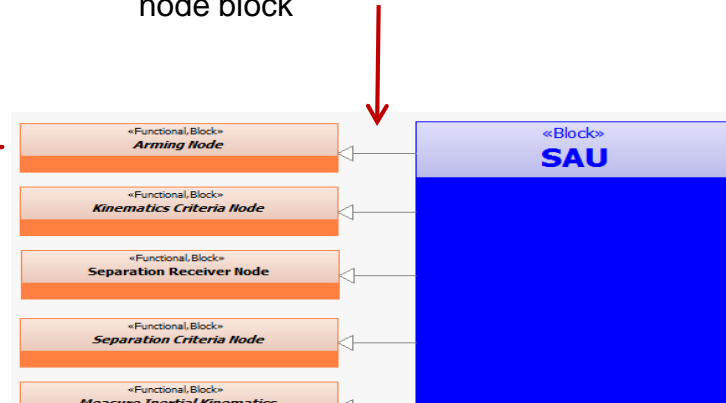
Use a 'wrapper' - a functional node. The functional node wrapper provides a means to allow a physical component to 'inherit' the functionality.



Tailored Requirement

Reference Requirement

The SAU 'inherits' the arming functionality via the functional node block

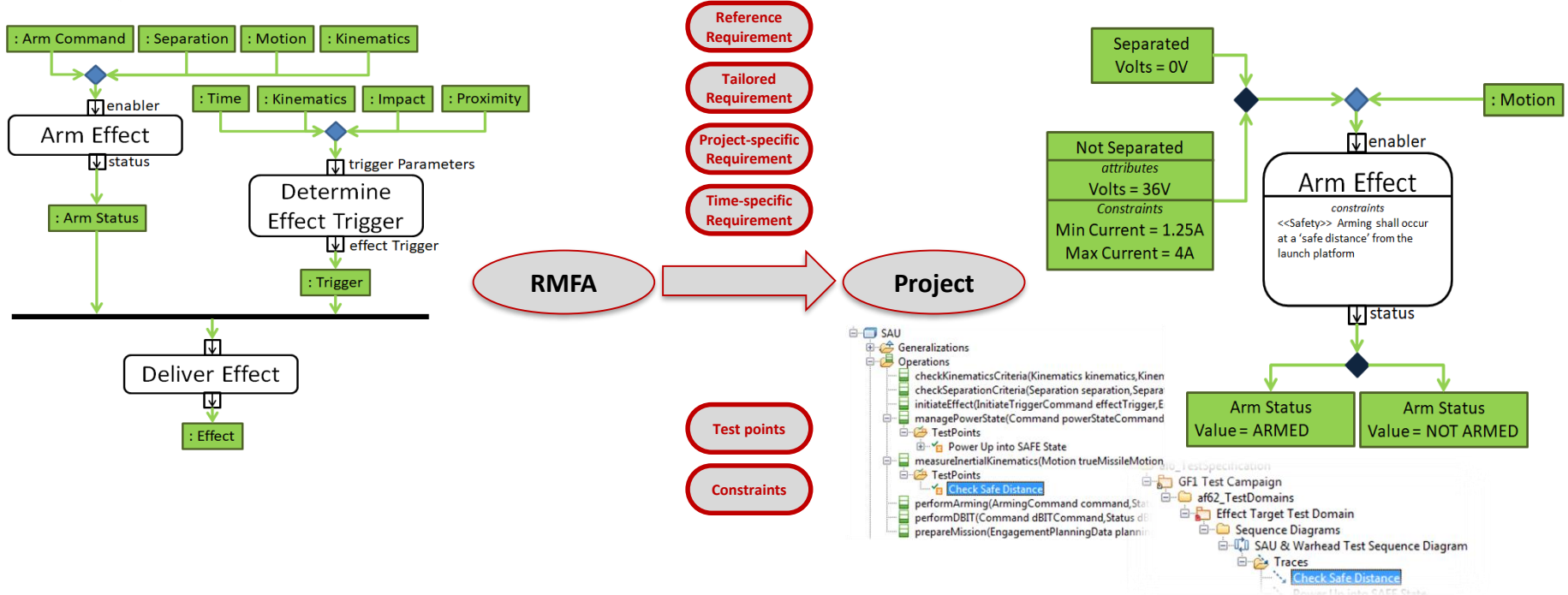


## Requirements allocation to equipment





# Tuning to Project - Example

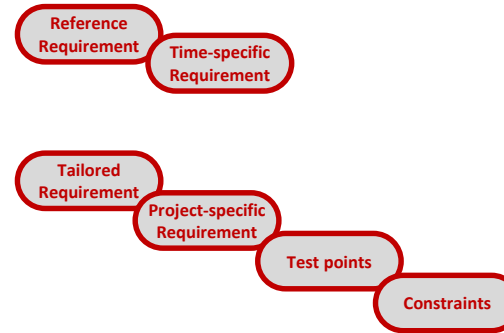


Once the architecture has been completed, constraints and test points can be added → support to V&V



## Tuning to Project – Requirements VS Model

- The **RMFA holds requirements in context**
  - Reference Requirements;
  - Time-specific Reference Requirements.
- **Tailoring RMFA to a model** allows for:
  - Tailored Requirements;
  - Project-specific Requirements (variant management);
  - Test Points;
  - Constraints.
- **Applying RMFA to Project** allows to:
  - Refine requirements;
  - Categorize and derive requirements;
  - Avoid designing from scratch, requirements refinement starts from a validated set of functions;
  - V&V response to requirements starts from the very beginning of the process.



**RMFA enables early validation and reduces risks**



# Conclusions







RMFA winning factors can be summarized as:

- The RMFA has been developed by system design practitioners rather than tool experts, focusing on **systems thinking** instead of MBSE techniques;
- The model that has been developed allows **quick deployment** (it is not just a collection of guidelines or studies) even if it's **not a plug-and-play tool**;
- The RMFA has been **tested on real projects** and collected many **feedbacks** before it has been released;
- It is a «**toolkit**» **for modellers**, that empowers their design activity (constrain but not constrict);
- The RMFA is a **live model**, that is enriched by feedbacks from projects;
- **Little maintenance** required on the model.





RMFA adoption gives advantages to projects in terms of:

- **Risk reduction** – functional analysis already validated;
- **Consistency** – documentation generated directly by the model;
- **Scalability** – broad approach, applicable from small to big developments;
- **Modularity and Reuse** – commonalities within the different project models;
- **Communication** – different teams have common taxonomy;
- **Sharability** – no restrictions with respect to national security;
- **Knowledge Transfer** – different projects share common views and way of modelling;



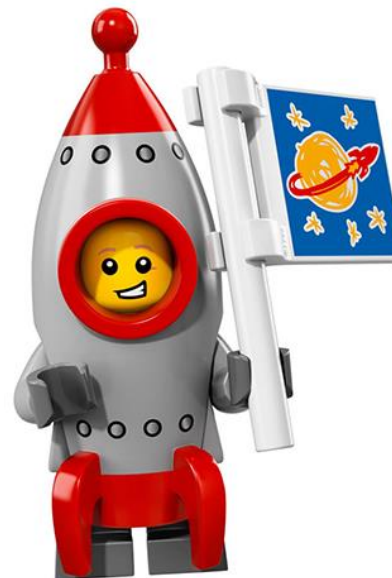


Even if the field of application of RMFA is deeply specific, its experience can be useful in other domains:

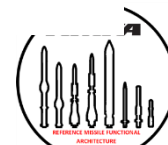
- « **Yes, we can** » message. Functional Architectures are daily-serves for the systems engineering community but examples of broad abstractions are limited;
- Requirements cannot always be harmonized/shared across projects, a **higher level of abstraction** allows to gain common paths;
- A **live model** is more useful than static guidelines and policies in helping modellers and the development process;
- RMFA approach ensures critical functions are captured and provided to a design team, **de-risking and reducing the effort of requirements engineering activities** (identify 100% functions/behaviors);
- The **concept of functional chains can be applied to any Complex System design problem** considering a complex set of behaviors/functions as a set of 'chains' (patterns).



Questions?



©LEGO



**MBDA**

MISSILE SYSTEMS