PYRAMID Exploiter's Pack Version 4.1

This document has been prepared, as part of the PYRAMID Exploiter’s Pack, in order to set out a generic approach to implementation of the PYRAMID Architecture. The PYRAMID Reference Architecture has not been created for any specific system. It is the user’s responsibility to ensure that any article created using this document meets any required operational, functional and safety needs. The Author accepts no liabilities for any damages arising due to a failure of the user to verify the safety of any product produced using this document, nor for any damages caused by the user failing to meet any technical specification.

For further information regarding how you can exploit PYRAMID on your project, provide feedback following your review of the PYRAMID Exploiter’s Pack V4.1, or have a technical query that you would like answering, please contact the PYRAMID Team using the following email address. PYRAMID@mod.gov.uk

This publication is available at https://www.gov.uk/government/publications/pyramid/pyramid-programme

EXECUTIVE SUMMARY

The MOD’s PYRAMID programme introduces a change to the current method of avionic systems design and procurement, aiming to make the next generation of air systems affordable, capable and adaptable by the adoption of an open architecture approach and systematic software reuse.

The PYRAMID Reference Architecture (PRA) is an open, air system, reference architecture aimed at software implementation that is both Exploiting and Execution Platform independent. It will support the realisation of the PYRAMID Key User Requirements (KURs) when fully instantiated within a complete mission system or air vehicle system.

The PYRAMID Exploiter’s Pack comprises:

- An introduction to the PYRAMID Exploiter’s Pack including reader guidance, overview information, the applicability of the KURs and the scope of the PRA.
- The PYRAMID Reference Architecture in both model and document form.
- A Deployment Guide, outlining the key processes that should be followed, including rationale for the use of PRA artefacts.
- A Compliance Guide, outlining the requirements for compliance declaration reporting by which any component or deployment of the PRA should be measured.
- A Glossary of terms, abbreviations and acronyms.

This PYRAMID Exploiter’s Pack document primarily acts as an introduction to the PYRAMID Exploiter's Pack and includes supporting information; the detailed content is provided in the PYRAMID Exploiter's Pack Annex documents.

CHANGE HISTORY

Date	Issue	Description of Changes
N/A	1	Issue 1 was never produced due to contractual reasons.
December 2020	2	This document, the PYRAMID Exploiter’s Pack, is formed from some of the content that was previously part of the PYRAMID Reference Architecture Description Document, BAES-IPAS-TIKAL-ENG-DOC-33159, Issue 1.1, January 2020. New content provided in this document includes: • Document Structure • Reader Guidance • Appendix A: Reader Guidance Examples New structure to the this document is as follows: • Annex A: PYRAMID Reference Architecture Description Document [2] • Annex B: Deployment Guide [3] • Annex C: Compliance Guide [4] • Annex D: Glossary [5]
December 2021	3	Issue 3 has been updated to include: • A new PRA Principles section (3.2.3). • Changes resulting from answering queries from both internal and external stakeholders and to improve maturity. • Updates to reflect changes to other PYRAMID Exploiter’s Pack documents.
October 2022	3.1	Issue 3.1 is a UK OFFICIAL version of Issue 3. No changes to the content of the document have been made other than non-technical changes due to the up-issue of the PYRAMID Exploiter’s Pack from Version 3 to Version 3.1 (headers/footers, references, etc.).
December 2022	4	Issue 4 has been updated to include: • Updates to reflect changes to other PYRAMID Exploiter’s Pack documents. • Changes resulting from answering queries from both internal and external stakeholders, and to enhance clarification and improve maturity. • The introduction of new and renamed components to the PRA component set, Fig.12. • Clarity improvements made to sections 3.3, PRA principles, and 3.6, Key Concepts and Common Features for Components.

Date	Issue	Description of Changes
		- Numbering changes to the PYRAMID Exploiter's Pack documents governance artefacts in line with external comments.
September 2023	4.1	The document has been updated due to now being released via Open Government License v3.

List of Effective Pages

62 pages UK OFFICIAL

62 pages in total

TABLE OF CONTENTS

ANNEXES

TABLE OF FIGURES

TABLE OF TABLES

TERMS AND ABBREVIATIONS USED IN THIS DOCUMENT

Definitions of project terms, the meaning of acronyms and the meaning of abbreviations used in this document can be found in the PYRAMID Glossary Ref. [5].

REFERENCES

The reference numbers are consistent across all the documents in the PYRAMID Exploiter's Pack. This means that in this document, when a reference is not used, the corresponding reference number will not appear in the reference list.

Project Related Document References:

For the avoidance of doubt, all documents referenced below which form part of the PYRAMID Exploiter's Pack are subject to the terms of DEFCON 703.

Reference	Title, Document Number, Issue & Date
[2]	PYRAMID Exploiter's Pack Annex A: PRA Description Document, RCO_FUT_23_005, Issue 4.1, September 2023.
[3]	PYRAMID Exploiter's Pack Annex B: Deployment Guide, RCO_FUT_23_006, Issue 4.1, September 2023.
[4]	PYRAMID Exploiter's Pack Annex C: Compliance Guide, RCO_FUT_23_007, Issue 4.1, September 2023.
[5]	PYRAMID Exploiter's Pack Annex D: Glossary, RCO_FUT_23_008, Issue 12.1, September 2023.
[6]	PYRAMID Reference Architecture Model, RCO_SYS_0025 Version 4.0.0, December 2022.
[7]	PYRAMID Reference Architecture Model v4.0.0 Model Version Description, BAES-FCAS-TIK-ENG-VD-101285, Issue 1, December 2022.
[60]	Dstl, Security Guidance for PYRAMID Exploiters, DSTL/TR111125, Issue 1.0, October 2021.

Non-Project Related References:

Non-project related references below contain information which is proprietary to that referenced third party. Any information from this source is subject to separate rights and terms and is not subject to the terms of DEFCON 703 or DEFCON 705.

Reference	Title, Document Number, Issue & Date
[8]	J. Borky and T. Bradley, Effective Model-Based System Engineering, 2019.
[9]	Raistrick et al, Model Driven Architecture with Executable UML, 2004.
[10]	Object Management Group, Model Driven Architecture (MDA) Guide, OMG, 2014.
[11]	T. Erl, Service-Oriented Architecture: Concepts, Technology and Design, 2005.
[12]	ISO/IEC 18384, Reference Architecture for Service Oriented Architecture (SOA RA), 2016.
[13]	C. Raistrick, Land Data Model Methodology and Modelling Standard, 2019.
[14]	UK Air and Space Power, Joint Doctrine Publication 0-30, 2nd Edition, 2017.

1 Introduction

1.1 PYRAMID and PYRAMID Reference Architecture

The MOD's PYRAMID programme introduces a paradigm shift to the current method of avionic systems design and procurement, aiming to make the next generation of air systems affordable, capable and adaptable by the adoption of an open architecture approach and systematic software reuse.

A PYRAMID Exploiter's Pack has been developed that defines: a reference architecture in the form of a set of coherent, reusable and well-bounded functional components; guidance for developing a PRA deployment; a set of compliance rules; and definitions of terms used.

Figure 1: PYRAMID Exploiter's Pack Context illustrates the context within which the PYRAMID Exploiter's Pack has been developed.

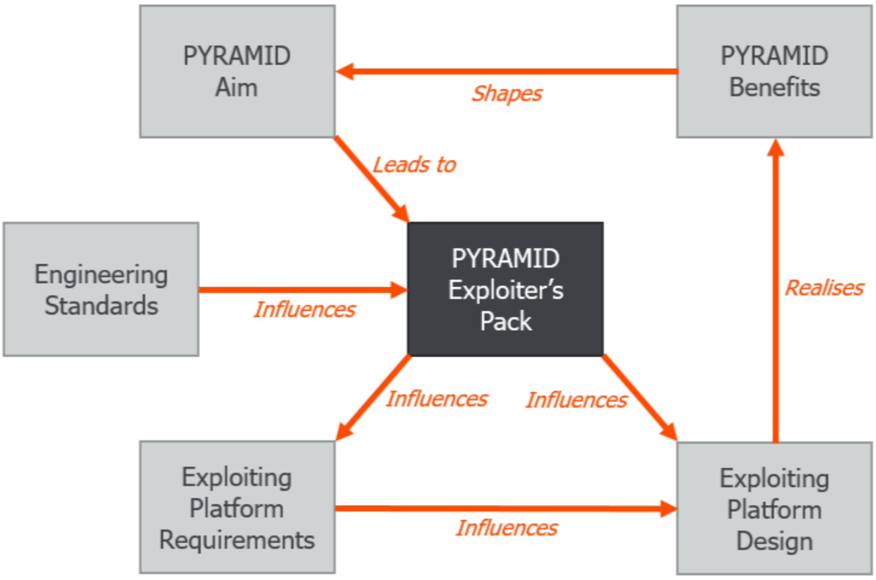


Figure 1: PYRAMID Exploiter's Pack Context

1.2 Scope

This is the main document of the PYRAMID Exploiter's Pack. It primarily acts as an introduction to the PYRAMID Exploiter's Pack, whereas the detailed content is provided in the PYRAMID Exploiter's Pack Annex documents. This main document comprises:

- A definition of the PYRAMID Exploiter's Pack document structure.
- Introductory information and guidance on reading the PYRAMID Exploiter's Pack.
- A definition of the PYRAMID KURs and how the PRA contributes to them.
- A definition of the architecture scope of the PRA.

1.3 Purpose

The purpose of this document is to introduce and define the scope of the PRA, as well as describe the PRA contribution to the fulfilment of the PYRAMID KURs. It also acts as the link between the PYRAMID Exploiter's Pack Annex documents, by providing reader guidance and greater context to help the reader understand the PRA.

2 Document Structure

Figure 2: Exploiter's Pack Structure represents the different documents that the Exploiter's Pack is composed of.

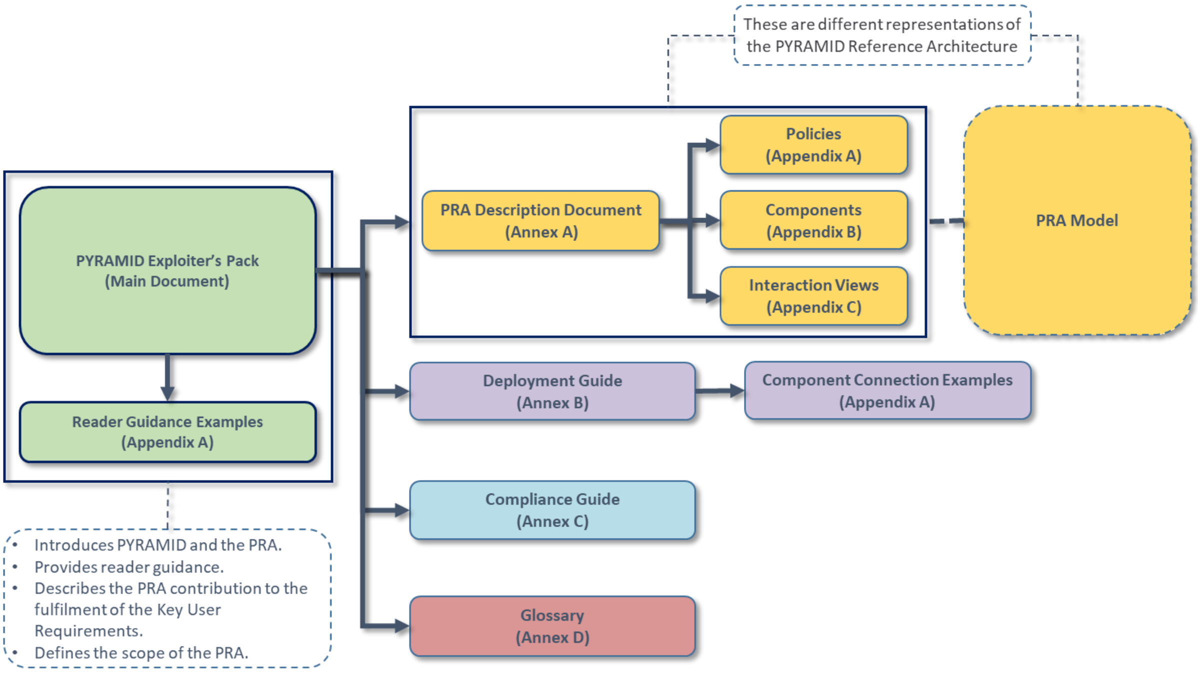

These are different representations of the PYRAMID Reference Architecture
PYRAMID Exploiter's Pack (Main Document)
Reader Guidance Examples (Appendix A)
PRA Description Document (Annex A)
Policies (Appendix A)
Components (Appendix B)
Interaction Views (Appendix C)
PRA Model
Deployment Guide (Annex B)
Component Connection Examples (Appendix A)
Compliance Guide (Annex C)
Glossary (Annex D)
Introduces PYRAMID and the PRA.
Provides reader guidance.
Describes the PRA contribution to the fulfilment of the Key User Requirements.
Defines the scope of the PRA.

Figure 2: Exploiter's Pack Structure

2.1 PYRAMID Exploiter's Pack (Main Document)

This document.

2.2 PRA Description Document + Model

The PRA Description Document Ref. [2] is generated from the PRA Model Ref. [6]. These contain the functional components that underpin the PRA. They also contain architectural policies and interaction views (IVs).

The PRA Description Document Ref. [2] provides these in a document format, whereas the PRA Model Ref. [6] provides these as a set of UML models.

2.3 Deployment Guide

The PYRAMID Deployment Guide Ref. [3] outlines how PYRAMID artefacts can be used to enhance a traditional system design process such that the PYRAMID KURs can be realised.

2.4 Compliance Guide

The PYRAMID Compliance Guide Ref. [4] outlines the requirements for compliance by which any component or deployment derived from the PRA should be assessed.

2.5 PYRAMID Glossary

The PYRAMID Glossary Ref. [5] defines a common set of terms and abbreviations relevant to the PYRAMID Exploiter's Pack.

3 Reader Guidance

The aim of this guide is to provide guidance and advice for reading the PYRAMID Reference Architecture (PRA) Exploiter's Pack, as well as showing:

- How the PRA and PYRAMID Exploiter's Pack fit into the wider PYRAMID programme.
- Some of the key concepts of the PRA and deployments thereof.

Whilst some of the content of this guide describes technical concepts, it is still intended to be accessible to less technical readers and such readers should be able to appreciate the essence of the PRA by the end of this guide. To aid with the understanding of first-time or non-technical readers key terms from the Glossary Ref. [5] are introduced when required.

It must be stressed that since this guide simplifies some concepts to make them more easily readable, the relevant parts of the PYRAMID Exploiter's Pack should be read to fully understand the concepts.

3.1 General Reader Guidance and Advice

3.1.1 How to Read the Exploiter's Pack

The PYRAMID Exploiter's Pack is not necessarily intended to be read in its entirety by any single reader.

Figure 3: Recommended PYRAMID Exploiter's Pack Reading and Order provides recommendations for the parts of the PYRAMID Exploiter's Pack that different readers may wish to read and a recommended order in which to read them. An overview of the different parts of the PYRAMID Exploiter's Pack is provided in section 2 - Document Structure.

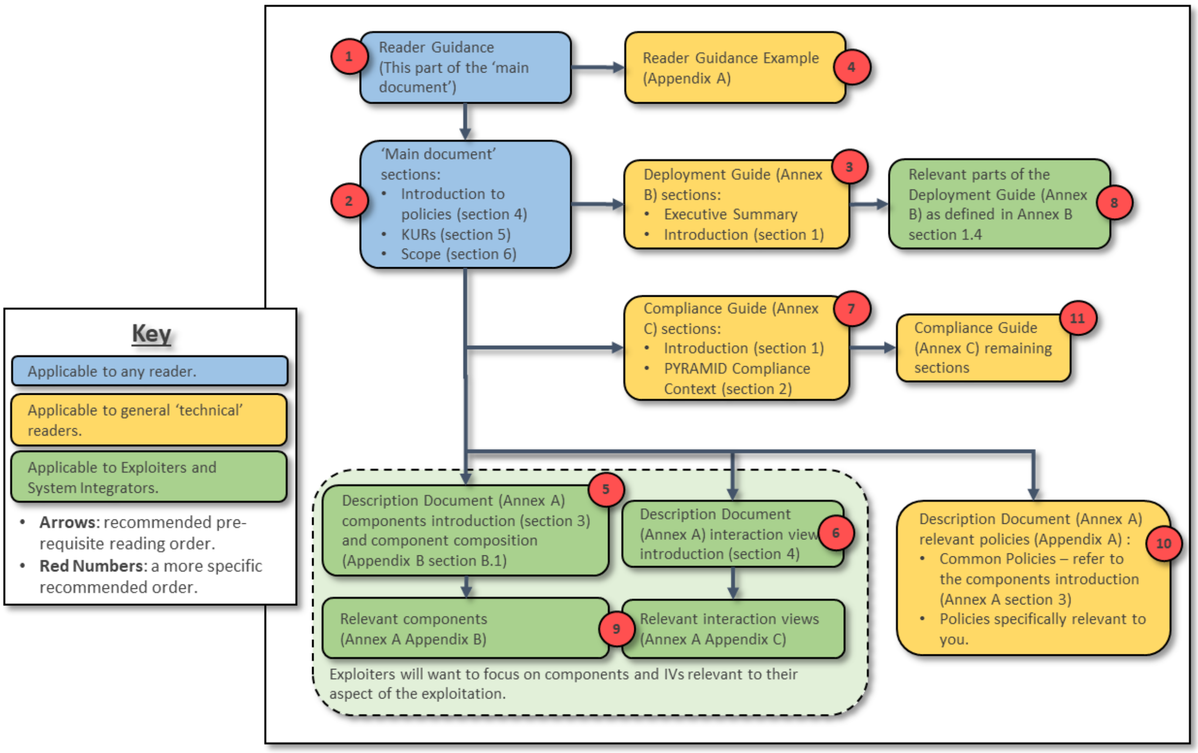

1
Reader Guidance (This part of the 'main document')
4
Reader Guidance Example (Appendix A)
2
'Main document' sections:
• Introduction to policies (section 4)
• KURs (section 5)
• Scope (section 6)
3
Deployment Guide (Annex B) sections:
• Executive Summary
• Introduction (section 1)
8
Relevant parts of the Deployment Guide (Annex B) as defined in Annex B section 1.4
7
Compliance Guide (Annex C) sections:
• Introduction (section 1)
• PYRAMID Compliance Context (section 2)
11
Compliance Guide (Annex C) remaining sections
5
Description Document (Annex A) components introduction (section 3) and component composition (Appendix B section B.1)
6
Description Document (Annex A) interaction view introduction (section 4)
9
Relevant components (Annex A Appendix B)
Relevant interaction views (Annex A Appendix C)
Exploiters will want to focus on components and IVs relevant to their aspect of the exploitation.
10
Description Document (Annex A) relevant policies (Appendix A) :
• Common Policies – refer to the components introduction (Annex A section 3)
• Policies specifically relevant to you.
Key
Applicable to any reader.
Applicable to general 'technical' readers.
Applicable to Exploiters and System Integrators.
• Arrows: recommended pre-requisite reading order.
• Red Numbers: a more specific recommended order.

Figure 3: Recommended PYRAMID Exploiter's Pack Reading and Order

The generic roles from Figure 3: Recommended PYRAMID Exploiter's Pack Reading and Order can be summarised as:

- **Any Reader:** Includes somebody who requires a broad top level understanding of the PRA. This may include someone who will have no need to use the PRA.
- **General 'Technical' Reader:** Somebody interested in understanding what the PRA is. This may include somebody not directly involved in the development of components or systems that are compliant to the PRA. Such readers may have management responsibilities involving the PRA for example.
- **Exploiter:** Somebody involved in the design and development of components or design of systems that are compliant with the PRA.
- **System Integrator:** Somebody involved in the wider integration of PYRAMID compliant systems.

When reading components and Interaction Views (IVs), Exploiters will want to focus on components and IVs relevant to their aspect of the exploitation.

The Deployment Guide Ref. [3] further defines Exploiter roles (in section 1.4) and points Exploiters and System Integrators to the content of interest within the Deployment Guide.

3.1.2 Recommended Experience and Training

Most of the PYRAMID Exploiter's Pack content is aimed at engineers and assumes a level of knowledge in:

- The subject matter areas covered by the PRA or a deployment of the PRA
- Systems engineering processes
- Safety and security-related aspects of engineering
- Model-Based Systems Engineering (MBSE)
- Reading and understanding Unified Modelling Language (UML) notation

The PRA uses a UML model-based architecture. Although it is designed to be useful whatever the chosen development approach, the system based upon the PRA is expected to be developed using an MBSE approach. As such, Exploiters and System Integrators will benefit from training or experience in MBSE and Model Driven Architecture (MDA) in order to fully appreciate or use the PYRAMID Exploiter's Pack. When exploiting the PRA an understanding of Service Oriented Architecture (SOA) will also be of benefit.

The following texts could aid understanding in the above areas:

- Model-Based Systems Engineering texts:
    - Text book - Effective Model-Based System Engineering Ref. [8]
- Model Driven Architecture texts:
    - Text book - Model Driven Architecture with Executable UML Ref. [9]
    - Guide - Object Management Group, Model Driven Architecture (MDA) Guide Ref. [10]
- Service Oriented Architecture texts:
    - Text book - Service-Oriented Architecture: Concepts, Technology and Design Ref. [11]
    - Standard - Reference Architecture for Service Oriented Architecture Ref. [12]
- Other texts that may be of interest:
    - Land Data Model Methodology and Modelling Standard Ref. [13]

3.2 PYRAMID and PRA Overview

This section provides a brief introduction to the PRA and how it fits into the wider PYRAMID programme. The PYRAMID overview is not intended to be comprehensive and focuses on the engineering aspects of PYRAMID.

This section also provides an overview of the content of the PYRAMID Exploiter's Pack, including how the different parts of the pack are related, as well as the evolution of the PRA.

The key terms introduced in the following section are:

> **Deployment:** A set of hardware and software elements forming a system (or part thereof) and used to support its system requirements.
>
> **Exploiting Programme:** A programme, e.g. Typhoon or TEMPEST, incorporating a deployment of the PRA.

3.2.1 PYRAMID Overview

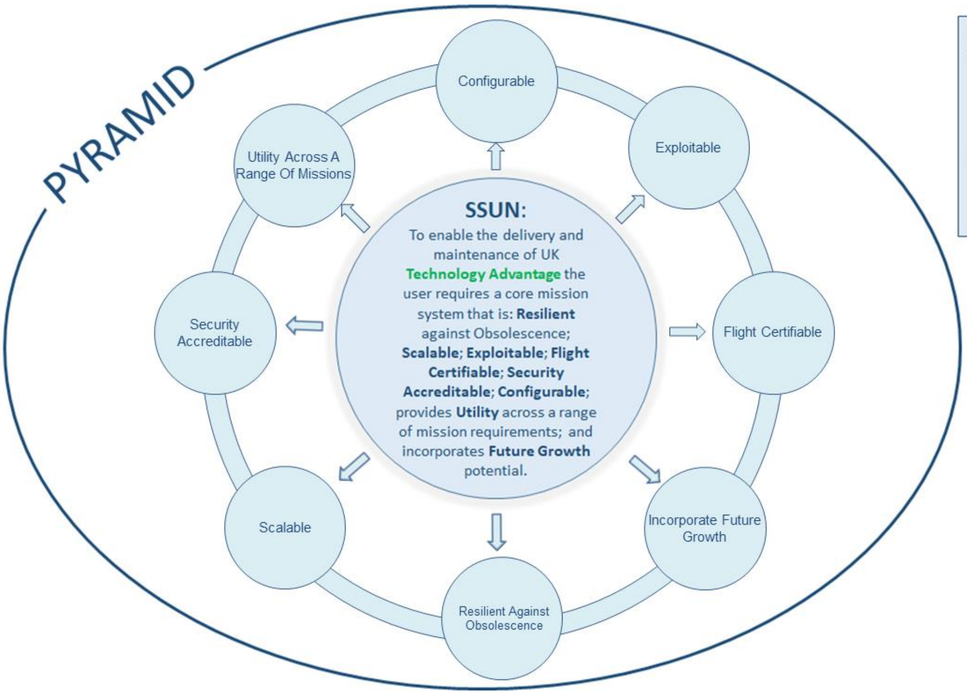
The MOD's PYRAMID programme introduces a paradigm shift to the current method of avionic systems design and procurement, aiming to make the next generation of air systems affordable, capable and adaptable by the adoption of an open architecture approach and systematic software reuse.

Generation by generation, aircraft capability is increasingly being delivered through avionics software which has become more and more complex and difficult to manage and upgrade through life. To maintain technology advantage through life, avionics systems must be able to rapidly respond to evolving threats and exploit emerging technologies in an incremental way.

The user requires a solution that offers rapid and affordable adaptability, but that also provides opportunities for reuse within aircraft platforms and across aircraft types. To ensure wide exploitation, the solution developed must be: inherently resilient to Obsolescence; Scalable; Exploitable; Flight Certifiable; Security Accreditable; Configurable; provide Utility across a range of mission requirements; and incorporate Future Growth potential.

A Single Statement of User Need (SSUN) summarises the MOD PYRAMID goal and is supported by 8 Key User

Requirements (KURs). These are central to the objectives of PYRAMID and emphasise an ethos that should be applied to all work relating to PYRAMID. Therefore, not only have they significantly influenced how the PRA has been developed, but they should also drive how a PYRAMID deployment should be developed. Figure 4: PYRAMID Single Statement of User Need and Key User Requirements shows the SSUN, 8 PYRAMID KUR titles and provides information on what technology advantage entails.


PYRAMID
Configurable
Exploitable
Flight Certifiable
Incorporate Future Growth
Resilient Against Obsolescence
Scalable
Security Accreditable
Utility Across A Range Of Missions
SSUN:

To enable the delivery and maintenance of UK **Technology Advantage** the user requires a core mission system that is: **Resilient** against Obsolescence; **Scalable**; **Exploitable**; **Flight Certifiable**; **Security Accreditable**; **Configurable**; provides **Utility** across a range of mission requirements; and incorporates **Future Growth** potential.

Where **Technology Advantage** entails:

- **Operational Advantage**: the underpinning capability to overcome your adversaries in military operations.
- **Freedom of Action**: the ability to develop, adapt and field operational capability at the time of our choosing.

Figure 4: PYRAMID Single Statement of User Need and Key User Requirements

Section 5 Key User Requirements provides the detailed definition of the PYRAMID KURs and explains how the PRA contributes to the fulfilment of these.

Figure 5: The MOD's PYRAMID Aims indicates the MOD's aims for the PYRAMID programme.

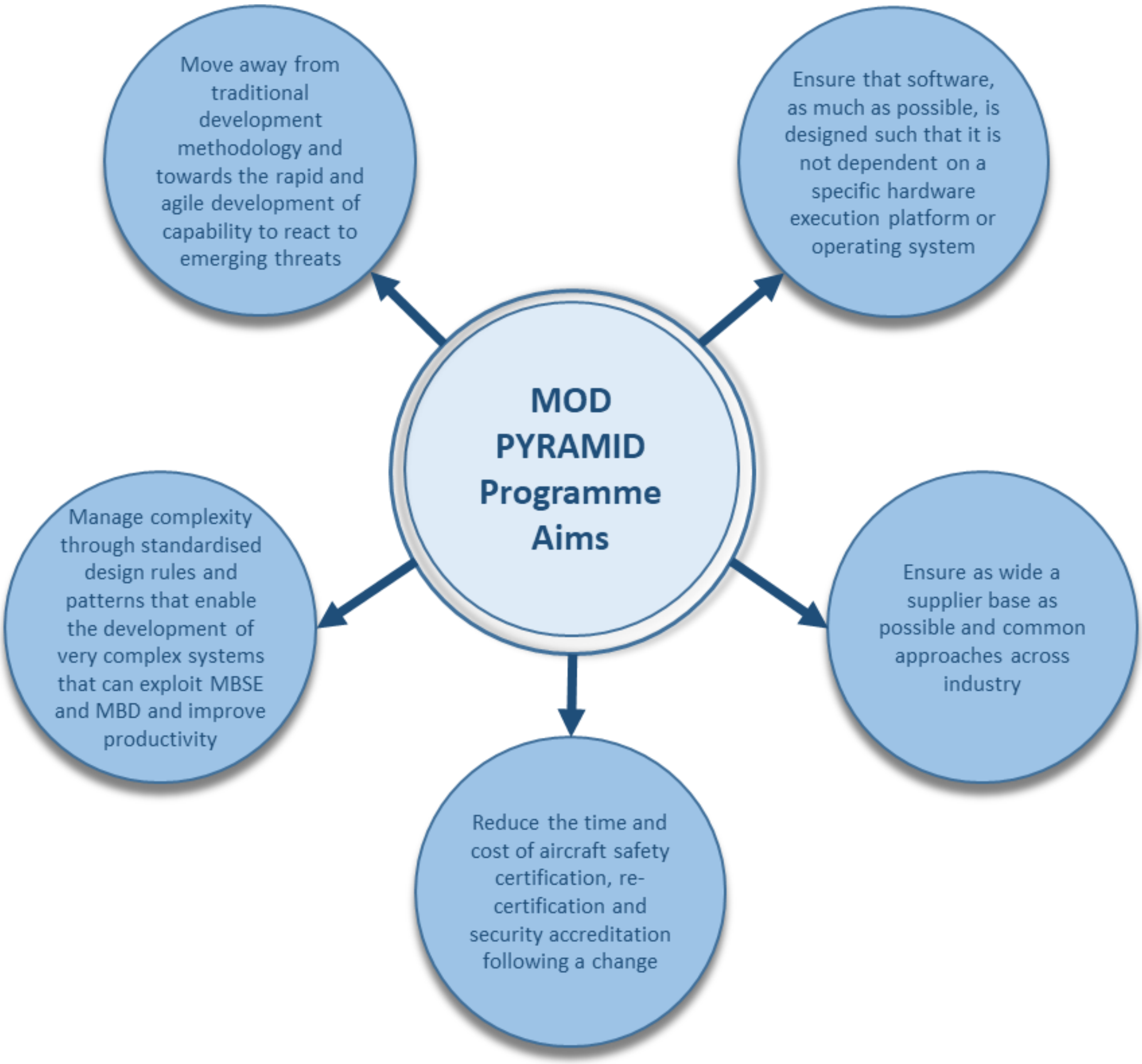

MOD PYRAMID Programme Aims
Move away from traditional development methodology and towards the rapid and agile development of capability to react to emerging threats
Ensure that software, as much as possible, is designed such that it is not dependent on a specific hardware execution platform or operating system
Manage complexity through standardised design rules and patterns that enable the development of very complex systems that can exploit MBSE and MBD and improve productivity
Ensure as wide a supplier base as possible and common approaches across industry
Reduce the time and cost of aircraft safety certification, re-certification and security accreditation following a change

Figure 5: The MOD's PYRAMID Aims

The PYRAMID programme is wider in scope than the PRA or the PYRAMID Exploiter's Pack. The PRA only defines a reference architecture and does not, for example, define further developed versions of components incorporating Exploiting Programme specific requirements.

The benefits of PYRAMID are potentially far reaching, but include:

- **Potential for Reuse:** The PRA components are not based on a specific aircraft and therefore can be reused across multiple aircraft. This underpins wider PYRAMID strategies for reuse, where components matured through a PYRAMID deployment also have the potential for reuse providing that the Exploiting Programme defining their requirements specifically supports this.

- **Common Approaches over Multiple Aircraft:** The use of a common approach across aircraft programmes is underpinned by the standardised PRA component set and policies.
- **Rapid Capability Update:** Better management of system development complexity, partly made possible by the standardised PRA component set and policies, combined with the potential for reusable components means that capability can be developed faster. Therefore, in conjunction with approaches to more rapid qualification, certification and accreditation, capability can be updated much more quickly.
- **Adaptable Systems:** The speed and ease through which systems can be adapted to perform different or specialised roles is enhanced through the increased ease of adding, or updating components. It also enables the adaptation of specific component behaviour without the need to replace the component. This also aids customisation of information and capability for different market environments.

3.2.2 PRA Overview and Summary of Scope

The PRA is a reference architecture for the software aspects of functionality of an air system, enabling software components to be developed and integrated into an Exploiting Platform. At its core is the decomposition of the software aspects into different areas of functionality. The decomposition consists of a number of standardised ‘building-blocks’ called components. The functionality of each component can be accessed via its services, which facilitate components interaction by allowing them to place or receive requirements for action or knowledge.

These component building blocks are supported by guidance in the form of:

- Section 3.3 - PRA Principles and how the PRA is intended to be used in the process of creating a deployment of the PRA. This information is contained within a series of policies. These direct people using the PRA to develop systems with common approaches that help enable systems to realise the SSUN and PYRAMID KURs.
- Examples (called Interaction Views) of how the components could be connected together, as part of a system, to achieve specific system level behaviour.
- More information about components and services can be found in Section 3 - Components in Appendix A of the Description Document Ref [2].

Figure 6: What is the PYRAMID Reference Architecture? breaks down the definition of the PYRAMID Reference Architecture.

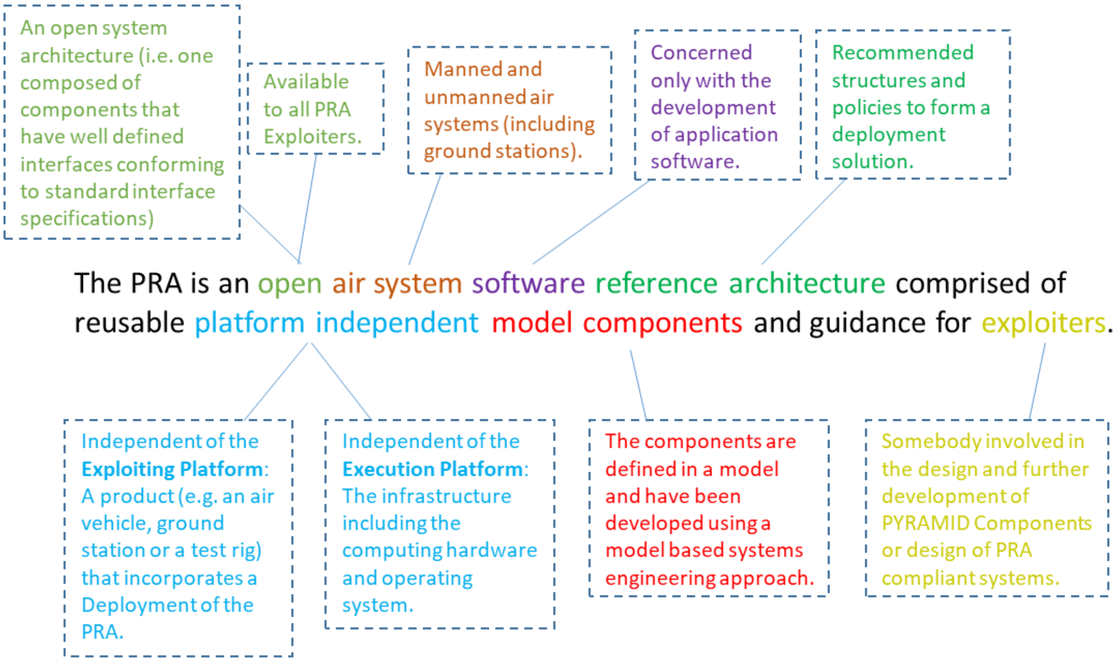

An open system architecture (i.e. one composed of components that have well defined interfaces conforming to standard interface specifications)
Available to all PRA Exploiters.
Manned and unmanned air systems (including ground stations).
Concerned only with the development of application software.
Recommended structures and policies to form a deployment solution.
The PRA is an open air system software reference architecture comprised of reusable platform independent model components and guidance for exploiters.
Independent of the Exploiting Platform: A product (e.g. an air vehicle, ground station or a test rig) that incorporates a Deployment of the PRA.
Independent of the Execution Platform: The infrastructure including the computing hardware and operating system.
The components are defined in a model and have been developed using a model based systems engineering approach.
Somebody involved in the design and further development of PYRAMID Components or design of PRA compliant systems.

Figure 6: What is the PYRAMID Reference Architecture?

It is intended to be used as the reference architecture from which PYRAMID compliant software components can be developed and integrated as part of a system. These are intended to be used within the scope summarised in Figure 7: The Scope of the PYRAMID Reference Architecture and more rigorously defined in section 6 - Architecture Scope; however, this does not exclude their use in other applications.

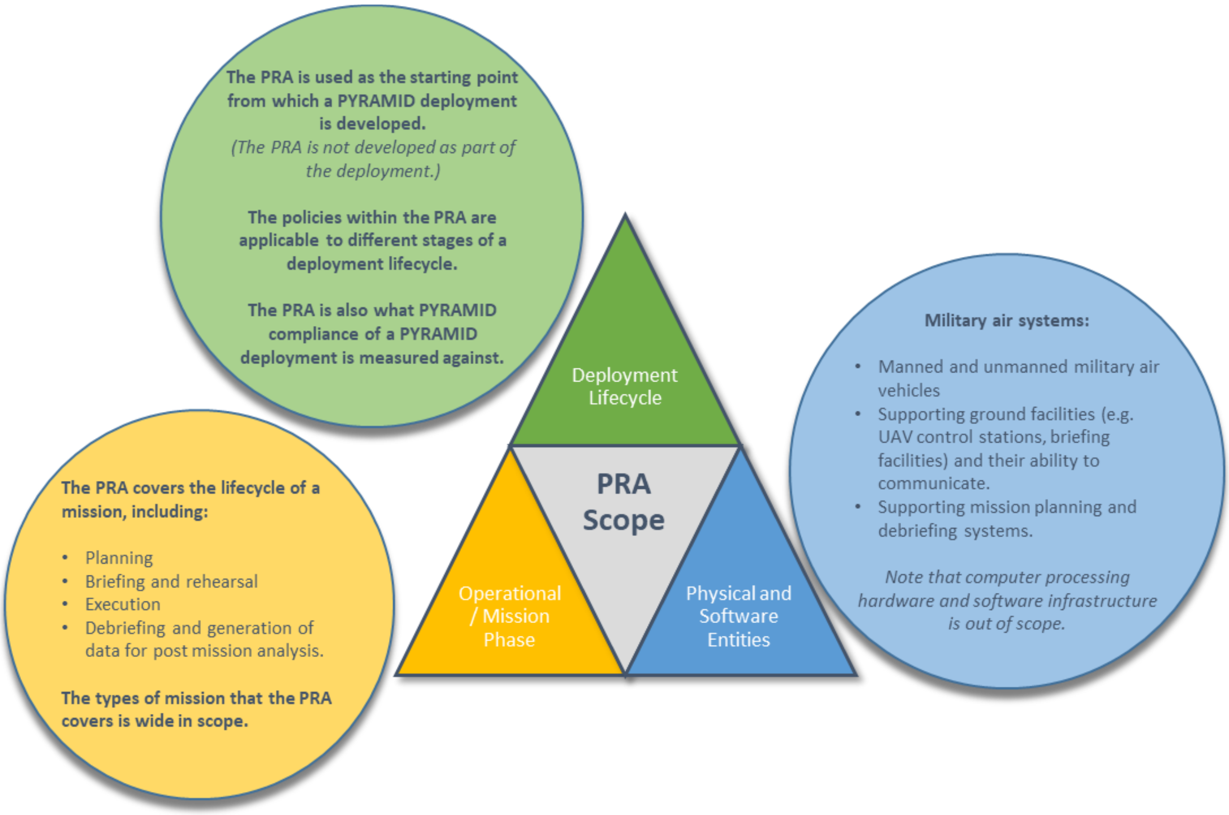

The PRA is used as the starting point from which a PYRAMID deployment is developed.
(The PRA is not developed as part of the deployment.)
The policies within the PRA are applicable to different stages of a deployment lifecycle.
The PRA is also what PYRAMID compliance of a PYRAMID deployment is measured against.
Deployment Lifecycle
PRA Scope
Operational / Mission Phase
Physical and Software Entities
Military air systems:
Manned and unmanned military air vehicles
Supporting ground facilities (e.g. UAV control stations, briefing facilities) and their ability to communicate.
Supporting mission planning and debriefing systems.
Note that computer processing hardware and software infrastructure is out of scope.
The PRA covers the lifecycle of a mission, including:
Planning
Briefing and rehearsal
Execution
Debriefing and generation of data for post mission analysis.
The types of mission that the PRA covers is wide in scope.

Figure 7: The Scope of the PYRAMID Reference Architecture

The deployment lifecycle refers to activities involved in developing PYRAMID compliant components as part of a deployment. This can include deployments supporting through-life updates for an Exploiting Programme.

Platform independence is an important concept within the PRA. Platform independence means that it is not dependent on the details of any particular infrastructure or product as highlighted in Figure 6: What is the PYRAMID Reference Architecture?.

3.2.3 Evolution of the PRA

The PRA has evolved from prior work in the context of an Unmanned Air System (UAS) platform. Figure 8: Prior Work Informing the PRA provides a high-level illustration of the evolution sequence, showing:

- Identification of UAS requirements based upon an assumed operational context and conservative safety and security analyses.
- Analysis of subject matter domains based upon the UAS conceptual design and requirements analysis.
- Identification of PRA policies, components and Interaction Views.

Whilst this body of work has informed the PRA definition, the PRA places no reliance on it and does not directly refer to it.

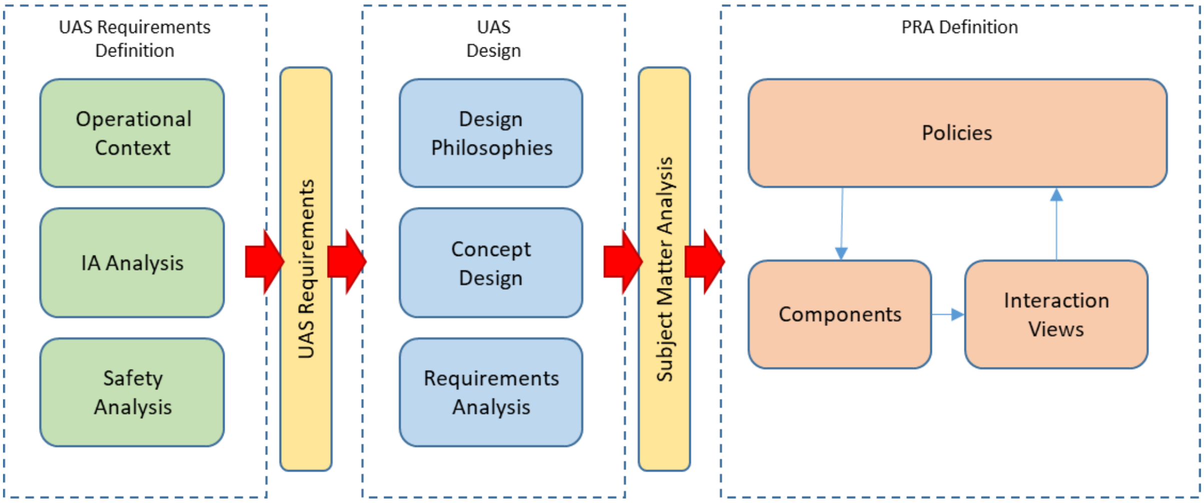


Figure 8: Prior Work Informing the PRA

Issue 1.1 of the PRA was the initial baseline release used to gather industry feedback and serve as a basis for the trial deployment work being undertaken.

Issue 2 of the PRA added further consideration of mission planning and power and cooling functionality, matured the understanding of interacting with equipment, began to expand the component definitions and addressed industry feedback on issue 1.1.

Issue 3 of the PRA completed the initial expansion of the component definitions for all components, added service definitions to some components, refined the existing policies and introduced some new policies, and addressed industry feedback on previous issues of the PRA.

Issue 3.1 of the PRA was produced as a consequence of a reassessment of the classification of its contents, with no technical changes to the PRA necessary.

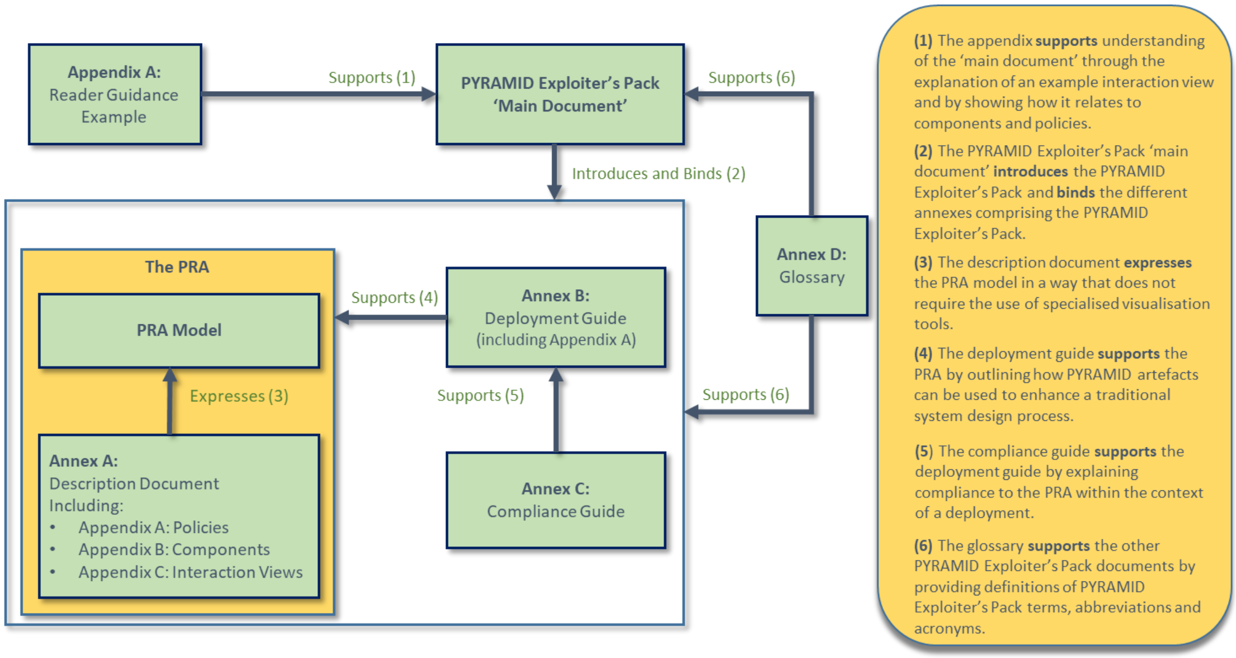
Issue 4 of the PRA (in this issue of the PYRAMID Exploiter's Pack) completes service definitions for every component (with the exception of Tasks Extensions), introduces a new Trajectory Prediction component, and provides further maturation and clarifications to the PRA, including in areas identified through industry feedback and validation activities. The model version description document Ref. [7] provides a more detailed view of the changes incorporated in issue 4.

3.2.4 PYRAMID Exploiter's Pack Overview

The PYRAMID Exploiter's Pack is built around the PRA model (which defines the PRA) as well as providing some supporting information. The structure of the PYRAMID Exploiter's Pack is described in Section 2 - Document Structure.

The PRA model uses UML but it is not a UML software design. It therefore does not adhere rigidly to all UML rules or common conventions. Furthermore, some UML artefacts are sometimes used to represent something different to what they would normally be used to represent within a UML software design. It must be emphasised that these deviations are deliberate in order to articulate information in a clear way.

Figure 9: The Relationship between PYRAMID Exploiter's Pack Contents shows the relationships between the PRA model and the documents that make up the PYRAMID Exploiter's Pack.


Appendix A: Reader Guidance Example
Supports (1)
PYRAMID Exploiter's Pack 'Main Document'
Supports (6)
Introduces and Binds (2)
The PRA
PRA Model
Supports (4)
Annex B: Deployment Guide (including Appendix A)
Annex D: Glossary
Expresses (3)
Supports (5)
Supports (6)
Annex A: Description Document Including:
• Appendix A: Policies
• Appendix B: Components
• Appendix C: Interaction Views
Annex C: Compliance Guide

(1) The appendix **supports** understanding of the 'main document' through the explanation of an example interaction view and by showing how it relates to components and policies.

(2) The PYRAMID Exploiter's Pack 'main document' **introduces** the PYRAMID Exploiter's Pack and **binds** the different annexes comprising the PYRAMID Exploiter's Pack.

(3) The description document **expresses** the PRA model in a way that does not require the use of specialised visualisation tools.

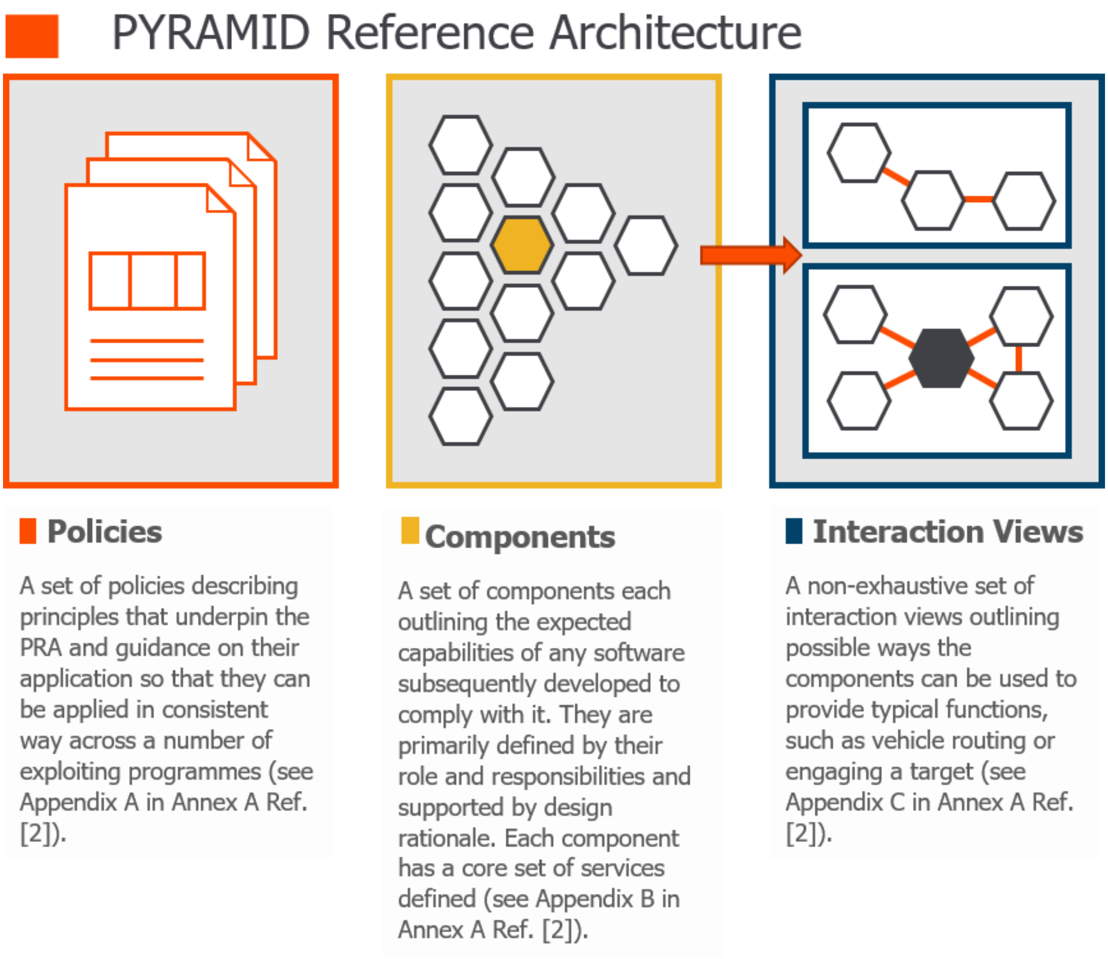
(4) The deployment guide **supports** the PRA by outlining how PYRAMID artefacts can be used to enhance a traditional system design process.

(5) The compliance guide **supports** the deployment guide by explaining compliance to the PRA within the context of a deployment.

(6) The glossary **supports** the other PYRAMID Exploiter's Pack documents by providing definitions of PYRAMID Exploiter's Pack terms, abbreviations and acronyms.

Figure 9: The Relationship between PYRAMID Exploiter's Pack Contents

The PRA itself is made up of policies, components and Interaction Views as shown in Figure 10: PYRAMID Reference Architecture Content. (The Description Document Ref. [2] provides a more detailed overview of these.)



PYRAMID Reference Architecture

Policies

A set of policies describing principles that underpin the PRA and guidance on their application so that they can be applied in consistent way across a number of exploiting programmes (see Appendix A in Annex A Ref. [2]).

Components

A set of components each outlining the expected capabilities of any software subsequently developed to comply with it. They are primarily defined by their role and responsibilities and supported by design rationale. Each component has a core set of services defined (see Appendix B in Annex A Ref. [2]).

Interaction Views

A non-exhaustive set of interaction views outlining possible ways the components can be used to provide typical functions, such as vehicle routing or engaging a target (see Appendix C in Annex A Ref. [2]).

Figure 10: PYRAMID Reference Architecture Content

Figure 11: The Relationships between Policies, Components and Interaction Views shows the relationship between policies, components and interaction views.

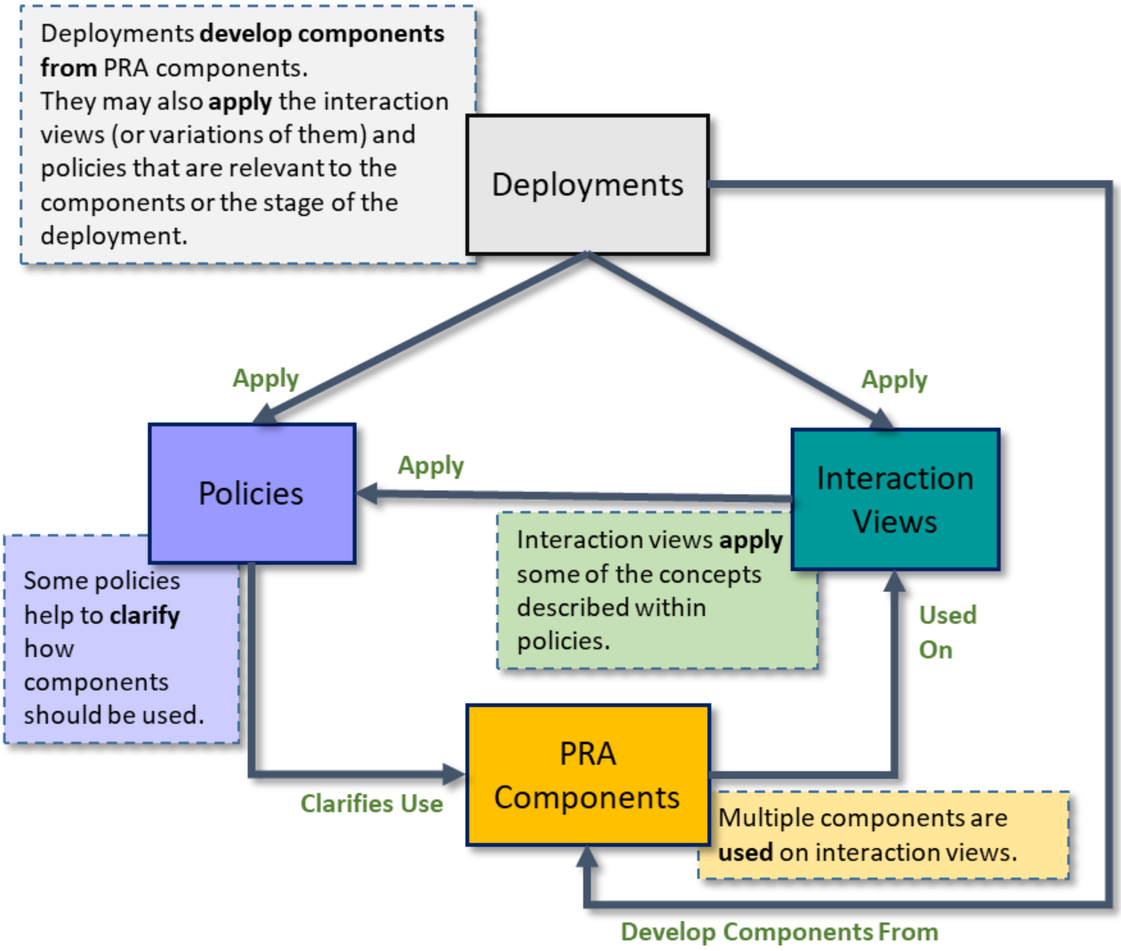

Deployments develop components from PRA components.
They may also apply the interaction views (or variations of them) and policies that are relevant to the components or the stage of the deployment.
Deployments
Apply
Apply
Policies
Apply
Interaction Views
Interaction views apply some of the concepts described within policies.
Some policies help to clarify how components should be used.
Used On
PRA Components
Clarifies Use
Multiple components are used on interaction views.
Develop Components From

Figure 11: The Relationships between Policies, Components and Interaction Views

Figure 12: PRA Component Set shows the components and, for illustrative purposes to aid reader orientation, how they broadly relate to traditional systems/sub-systems. Note that there is no particular meaning behind where components are relative to other components on the diagram, nor does the colouring imply any restriction on how components may be used.

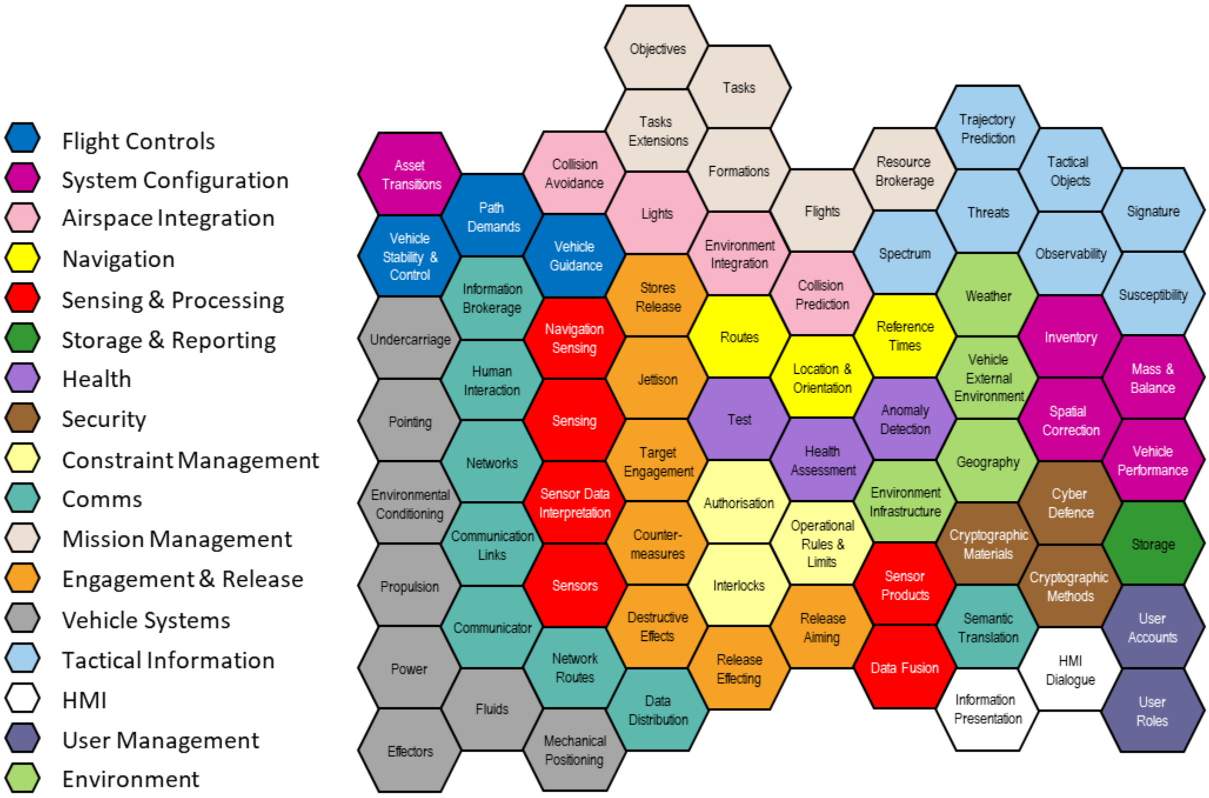


Figure 12: PRA Component Set

Components and Interaction Views are developed so that they can apply to multiple phases of a mission or operation (within the PRA scope). This means that instances of components can be deployed on different systems to support all required mission/operation phases. Figure 13: Component and Interaction View Applicability to Mission/Operation Phases provides a view of this.

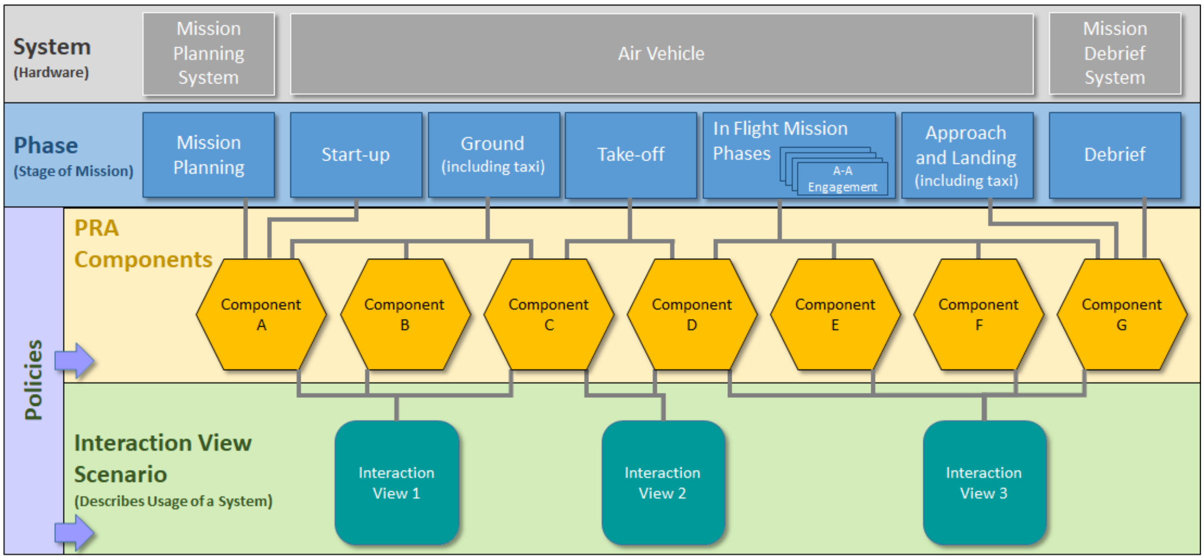


Figure 13: Component and Interaction View Applicability to Mission/Operation Phases

Appendix A contains an example from the PRA of an Interaction View and its components in order to demonstrate:

- The relationship between policies, components and Interaction Views
- The applicability of these components and Interaction Views to multiple mission/operational phases

3.2.5 Security Guidance for PYRAMID Exploiters

A Security Guidance for PYRAMID Exploiters document, Ref. [60], is available from Dstl on request. This guidance document provides additional insight on a number of security aspects that will likely apply to Exploiting Programmes using PYRAMID-based deployments and will need to be addressed in order to achieve security accreditation and capability assurance sign-off for MOD acquisition projects.

To request this document, please contact PYRAMID@MOD.GOV.UK.

3.3 PRA Principles

This section describes the key principles behind the design of the PRA and the way in which it is expected to be used. Other principles are described elsewhere within the PYRAMID Exploiter's Pack, in the PRA policies and the Deployment Guide Ref. [3].

3.3.1 PRA Design Principles

3.3.1.1 Separation of Concerns

The most significant principle used to design the PRA is the separation of concerns. In system design, it is useful to make a separation between the functional needs of a mission system (what something does) and the needs of a deployment (how it does it). As a reference architecture, the PRA was established on the needs of functional design, as non-functional design is dependent on a specific deployment. This focus on functional independence from an exploiting platform promotes clarity of understanding of the PRA. How the functions are achieved by the exploiting programme is a separate concern which the PRA does not address.

3.3.1.2 Separation of Subject Matter

Domain modelling has been used to separate the PRA into components that focus on an area of knowledge. Each component represents a discrete area of subject matter that includes behaviour and all the data needed to achieve it. A component's roles and responsibilities are defined using terms appropriate to its subject matter, and provides a common language for users, developers and exploiters of the component.

The separation of subject matter supports effective and efficient reuse of components between deployments without undue constraint on design, when a change is made, the impact is minimised according to the scope of the subject matter. This enables upgrades to be potentially isolated to one component and its associated functions, supporting the PRA's goal to be rapidly adaptable.

3.3.1.3 System and Environment Independence

Components are agnostic of their environment: they are defined in ways that make minimal assumptions about how they will be used or how they will be connected. In software terms, they are highly decoupled. Components are therefore independent of deployment factors such as spatial separation (where two PRA components in different platforms interact, e.g. ground station and air vehicle) and temporal separation (covering different phases of the mission, e.g. pre-planning, execution and post-mission analysis).

3.3.1.4 What, not How

A component definition is a requirement specification; it says what a component must do (primarily in terms of its services). It does not say how a component should be implemented. This has been an important consideration in deciding what content should be included in the PRA.

3.3.2 PRA Exploitation Principles

3.3.2.1 Component Connections

An exploitation of the PRA will combine components to support system functionality through the use of bridges. As each PRA component is defined in the language of its subject matter , bridges are used to close the semantic gaps (acting as translators or intermediaries) between deployed components. This enables components, which may have been developed independently, to exchange information seamlessly.

3.3.2.2 Design Around Static Data

Each component contains the data relevant to its subject matter. Where components understand the same concept in the real world, but from the view of different subject matters, those objects or concepts should be linked by counterpart transformations provided by bridges.

As relevant data is embodied in components, it does not "flow through" the system in the traditional way. In particular, there is no need for data to "pass through" components where no value is added.

3.3.2.3 Design for Change

A driver behind PYRAMID is that future systems must be much more receptive to change. The PRA expects components to be specialised and configured for an exploitation by the use of data driving (see Data Driving). PRA components can also be extended to add specialised behaviour (see Component Extensions).

3.3.2.4 Avoid Component Duplication

A component represents behaviour and data. An exploitation should use a single instance of a component, tailored via extensions and data driving for multiple uses, rather than using multiple instances. However, there are number of specific situations described in the Deployment Guide Ref. [3] where a deployment would benefit from the use of multiple component instances.

3.3.2.5 Interaction at PRA Boundaries

The PRA acknowledges that Exploiters may develop systems, where parts of the system capability that could have been provided by the PRA, use non-PYRAMID components instead. For example, a legacy system may be used to provide datalink functionality rather than create a software defined radio built from PYRAMID components. The boundary of the PRA based part of the system (or ‘PRA system’) is therefore flexible and specific to a deployment. This means that the ‘PRA system’ will interact with the wider system at different levels of abstraction (or detail) based on the nature and capabilities of the wider system. For example, this could range from low level instructions to a sensor on how to make a measurement, to high level instructions to a sensor equipment assembly to track a target.

Furthermore, PYRAMID components are expected to be used in conjunction with the capabilities provided by the computing infrastructure which includes other software such as middleware and operating systems. Therefore, PYRAMID components can, and should, interact with capabilities provided by the infrastructure, such as accessing mathematical functions, managing security and safety partitions, and managing storage media and storage implementation.

These interactions are not shown in the PRA as they will be specific to the deployment of the PYRAMID components. Access to any of these capabilities must never be for a reason that is beyond the scope of the component’s subject matter. This means that these capabilities should not be used in a deployment in this way if they duplicate any part of the subject matter of another PRA component. However, where the PRA is not used for all aspects of the system development, they may still be accessed via the services defined within the PRA as if interacting with another PYRAMID component.

3.4 Deployment Lifecycle

This section outlines how PRA components might be developed to produce PYRAMID compliant software components and how these may be deployed as part of a system.

The key terms introduced in this section are:

Component Behaviour: The behaviour required from a component in order to fulfil its responsibilities within the system and provide its services.

Execution Platform: The infrastructure supporting the execution, communication, etc. of application functionality, e.g. ECOA, ARINC 653, Linux, Windows, and the computing hardware.

Exploiting Platform: A product (e.g. an air vehicle, ground station or a test rig) that incorporates a deployment of the PRA.

Platform Independent Model (PIM): A representation of a system that is independent of the Execution Platform.

Platform Specific Model (PSM): A representation of a system that incorporates the Execution Platform.

Figure 14: Bringing Together PYRAMID Components and Other Software shows how PRA components might come together with other software and be developed into a platform independent design. This includes the reuse of previously developed PIM components (see **Platform Independent Model (PIM) Components**), and how these are subsequently developed into something that can be run on a chosen Execution Platform, again with possible reuse of developed PSM components.

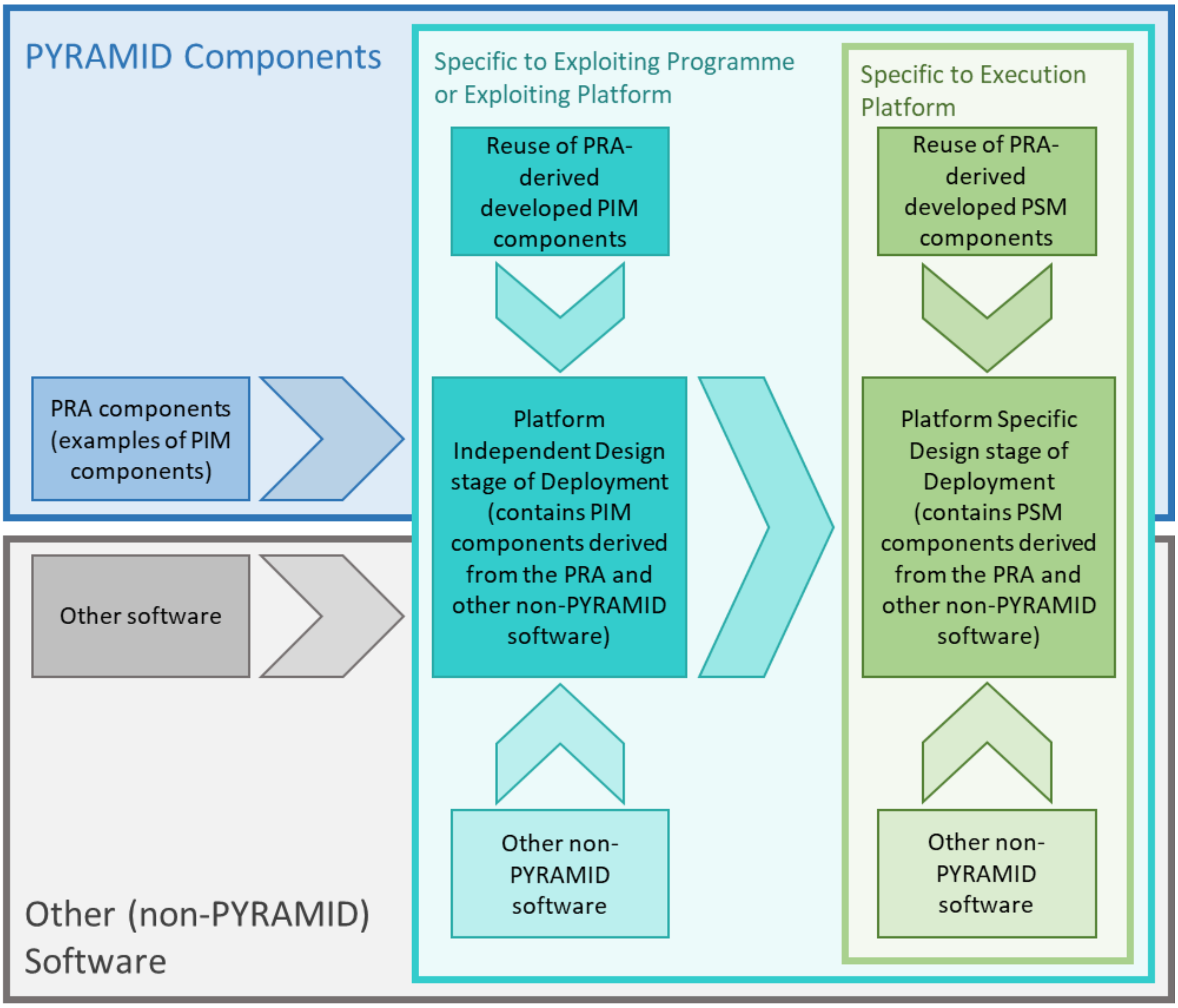

PYRAMID Components
PRA components (examples of PIM components)
Other software
Other (non-PYRAMID) Software
Specific to Exploiting Programme or Exploiting Platform
Reuse of PRA-derived developed PIM components
Platform Independent Design stage of Deployment (contains PIM components derived from the PRA and other non-PYRAMID software)
Other non-PYRAMID software
Specific to Execution Platform
Reuse of PRA-derived developed PSM components
Platform Specific Design stage of Deployment (contains PSM components derived from the PRA and other non-PYRAMID software)
Other non-PYRAMID software

Figure 14: Bringing Together PYRAMID Components and Other Software

Figure 15: Example PYRAMID Deployment Evolution provides a pictorial representation of the PRA and how its contents can be built upon through a deployment process. This section explores the three parts of this figure:

- **The PRA:** As defined by the PYRAMID Exploiter's Pack.
- **A Platform Independent Design:** A component and/or system design process based on PRA components that does not incorporate Execution Platform considerations (e.g. is independent of any computation hardware or operating system). A platform independent system design may also incorporate other non-PYRAMID software designs.
- **A Platform Specific Design:** A component and/or system design process developing the Platform Independent Design to incorporate the relevant Execution Platform considerations into the design. Again, the Platform Specific Design may incorporate other non-PYRAMID software designs.

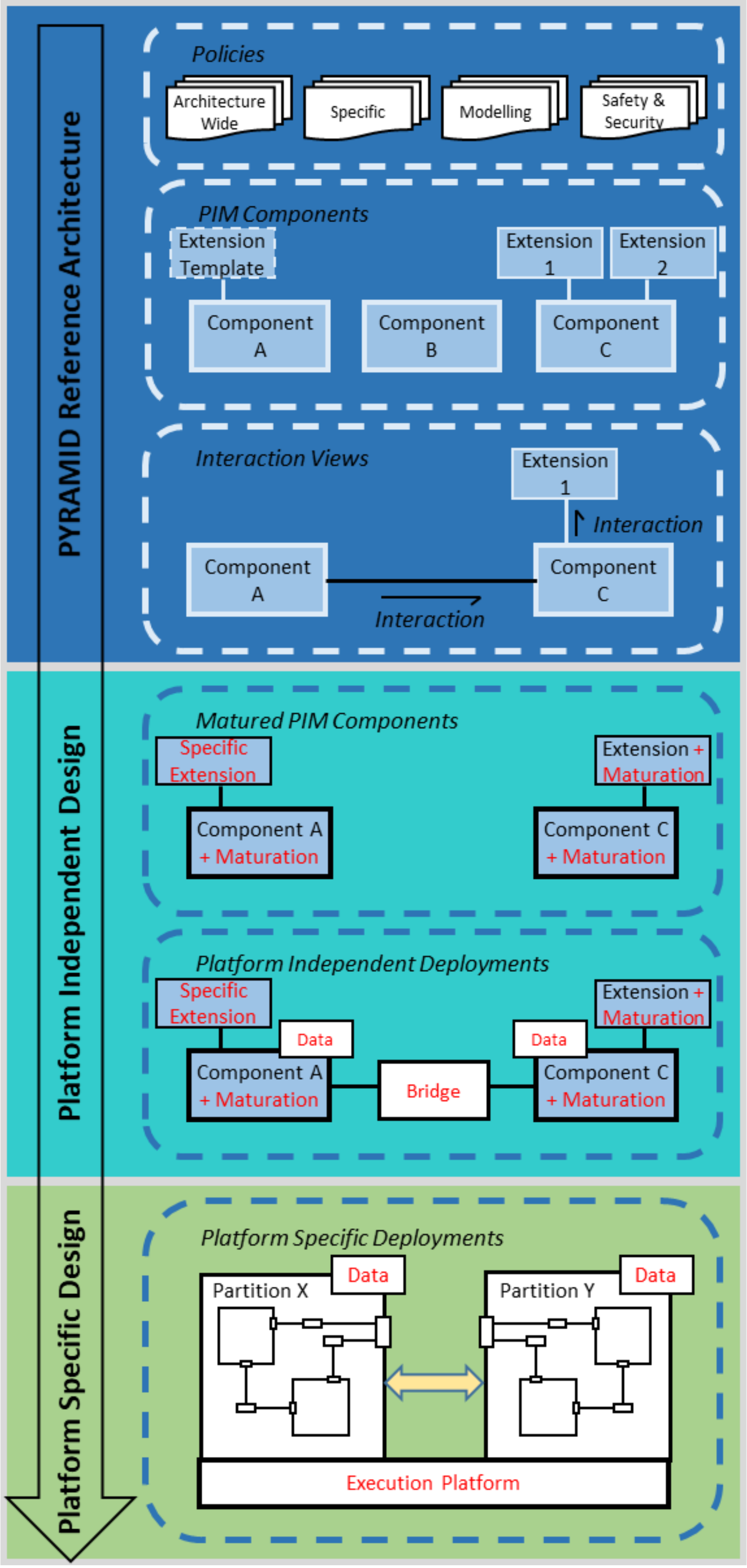


Figure 15: Example PYRAMID Deployment Evolution

3.4.1 The PYRAMID Reference Architecture

The PRA is the starting point for development activities. Developers cannot modify the PRA; however, they can provide feedback for future consideration, via the PYRAMID Query Management System (PQMS).

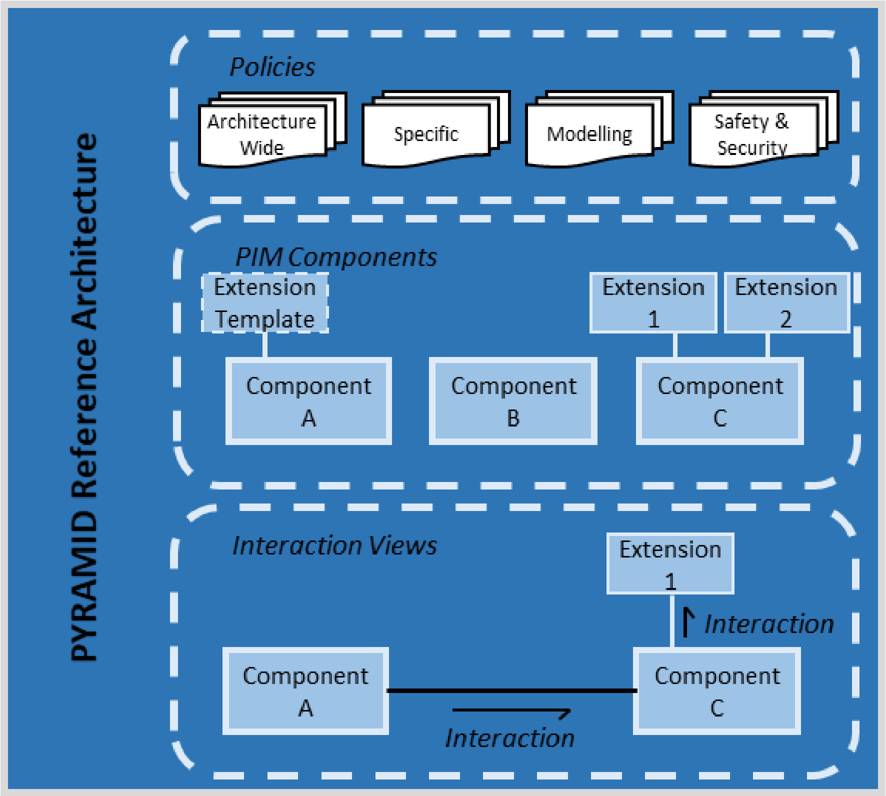


Figure 16: The PRA – used by a Deployment

A description of policies, components and interaction views has already been provided. However, Figure 16: The PRA – used by a Deployment introduces the following concepts:

- **Platform Independent Model (PIM) Components:** The components in the PRA are examples of PIM components, meaning that they are independent of any specific Execution Platform. The term "PIM component" is used when describing the development of a component from a PRA component in the platform independent design process, but not all PIM components will be derived from the PRA.
- **Component Extensions:** These are optional to allow for specialisation of component subject matter and are discussed further in section 3.4.2 - Platform Independent Design.

An Exploiter should identify:

- Components relevant to the deployment.
- Policies relevant to the components and to the stage in the deployment lifecycle. Relevant policies are listed within the component design considerations in Appendix B of the Description Document Ref. [2].

3.4.2 Platform Independent Design

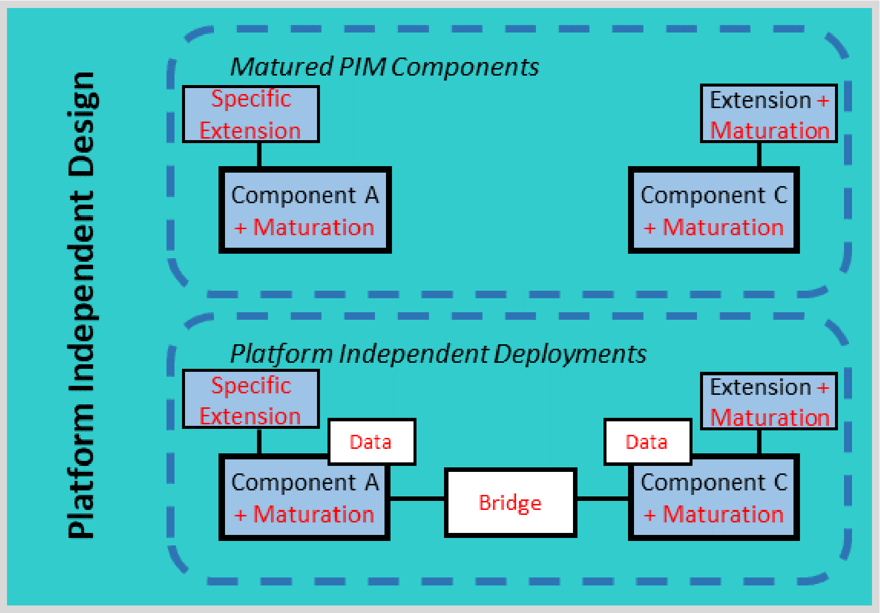


Figure 17: Platform Independent Design

Figure 17: Platform Independent Design shows how this design process adds further detail and definition to the PIM components derived from the PRA and develops a PIM deployment.

Maturing the PIM component design involves adding the following and in addition Exploiting Programme specific context could be added at this time:

- Class, state and service dependency definitions. (Service dependencies define the relationship between the services required by the component and the services provided by the component). This includes maturing Extension Components.
- Additional component behaviour detail.
- Subject matter specific data.

A PIM Deployment involves:

- Specifying which PIM components derived from the PRA to use in the PIM deployment.
- Adding Bridges and further deployment specific configuration data.

Components within a deployment are independent of each other. This improves the scope for component reuse, minimising the impact of change and easing development maintenance overheads. Simple, self-contained and deterministic bridges are used to connect components together and allow interaction between different parts of a system.

The PRA encourages various methods to provide flexibility in component use and effectiveness, allowing greater adaptability, exploitability, reuse and configurability:

- **Component Extensions:** This allows functionality provided by a component to be split across a parent component and one or more extension components, with the parent responsible for providing services to other components. This allows component extensions to be developed with a degree of independence, for example allowing for specialised or enhanced subject matter capability.
- **Data Driving:** A method for allowing a component's specific behaviour to be specified or optimised through the use of data files. This reduces the need for component redesign.
- **Multiple Component Instances:** Multiple instances of a component can be used within a deployment in order to allow components to be tailored for specific purposes (e.g. to support different sensor types) or to distribute/compartmentalise behaviour (e.g. for safety or security reasons).

Sections 2 and 3 in the Deployment Guide Ref. [3] provide more detail on how PRA artefacts can be used in platform independent design.

3.4.3 Platform Specific Design

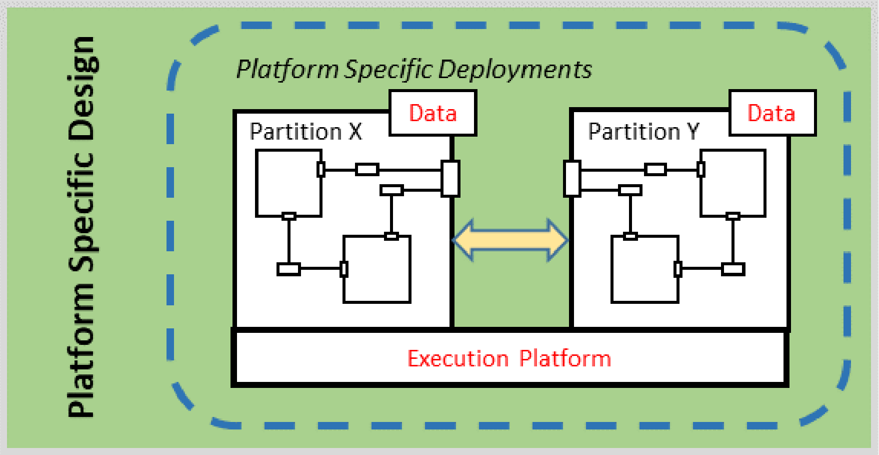


Figure 18: Platform Specific Design

Figure 18: Platform Specific Design shows the result of how this design process might transform the PIM deployment into a Platform Specific Deployment by taking account of the Execution Platform (i.e. being compatible with the software and computing infrastructure that is outside of the component scope).

Development activities potentially include satisfying non-functional requirements (such as latency, safety and security), adding implementation specific data and software implementation requirements (such as data driving, partitioning and coding languages). Partitioning is the separation of software in some way, often via some physical means.

Sections 4 and 5 in the Deployment Guide Ref. [3] provide more detail on how PRA artefacts can be used in platform specific design.

3.5 Compliance

It is recommended that developers assess the attained compliance of each component or deployment against the PRA definition.

Compliance declaration is intended to provide a consistent manner of collecting and retaining the information that is required to support and improve:

- Ease of integration of components and deployments.
- The ability to reuse components between differing deployments.
- The ability to insert and update capability of a deployment rapidly, and with reduced risk.

It is likely that the component's or the deployment's attained compliance will be required to be understood and declared by the Exploiting Programme at its key programme milestones.

It is probable that compliance reviews would be held alongside an Exploiting Platform's design reviews.

The Compliance Guide Ref. [4] describes compliance in detail.

3.6 Key Concepts and Common Features for Components

This section describes some key concepts and common features of how components are able to work together in the PRA to achieve mission objectives.

3.6.1 Requirement Breakdown and System Interaction

The PRA is an architecture containing loosely coupled components, where in general any component can interact with any other. The Control Architecture policy defines layers in which a component sits, with the components in each layer having different roles in the achievement of mission objectives. This is achieved through components receiving requirements to be satisfied by the provision of a service, which the component can then break down into further derived requirements for other components to satisfy where necessary. This provides a more consistent control mechanism that can be understood by all Exploiters. Refer to the Control Architecture policy in the Description Document Ref. [2] for detailed information.

Information from other components may be required to help plan or execute a solution in order to fulfil the provision of a service. This may include, for example, a capability assessment which is dependent on current resource information, or an aiming function that has a dependency on missing positional information. The solution dependencies may also include a component tasking other components when supporting activity is required. The PRA specifies how these dependencies between components are managed in a flexible manner in order to respond to changes in the situation that require changes to planned or currently executing solutions. Refer to the Dependency Management policy in the Description Document Ref. [2] for detailed information.

This layered structure is loosely comparable to the management structure of an organisation where executive level management set goals based on the managing directors' objectives. This is analogous to the objectives being handled by the Objective Layer issuing task requirements to the Task Layer. Departmental management take account of these and derive more specific goals (analogous to a Tasks component issuing action requirements) to be achieved by the various employees within the department, with various roles and responsibilities, who create products by using the necessary resources.

Figure 19: Component Applicability to the Control Architecture Layers shows the four control architecture layers, plus the Service components, and which components sit within each layer.

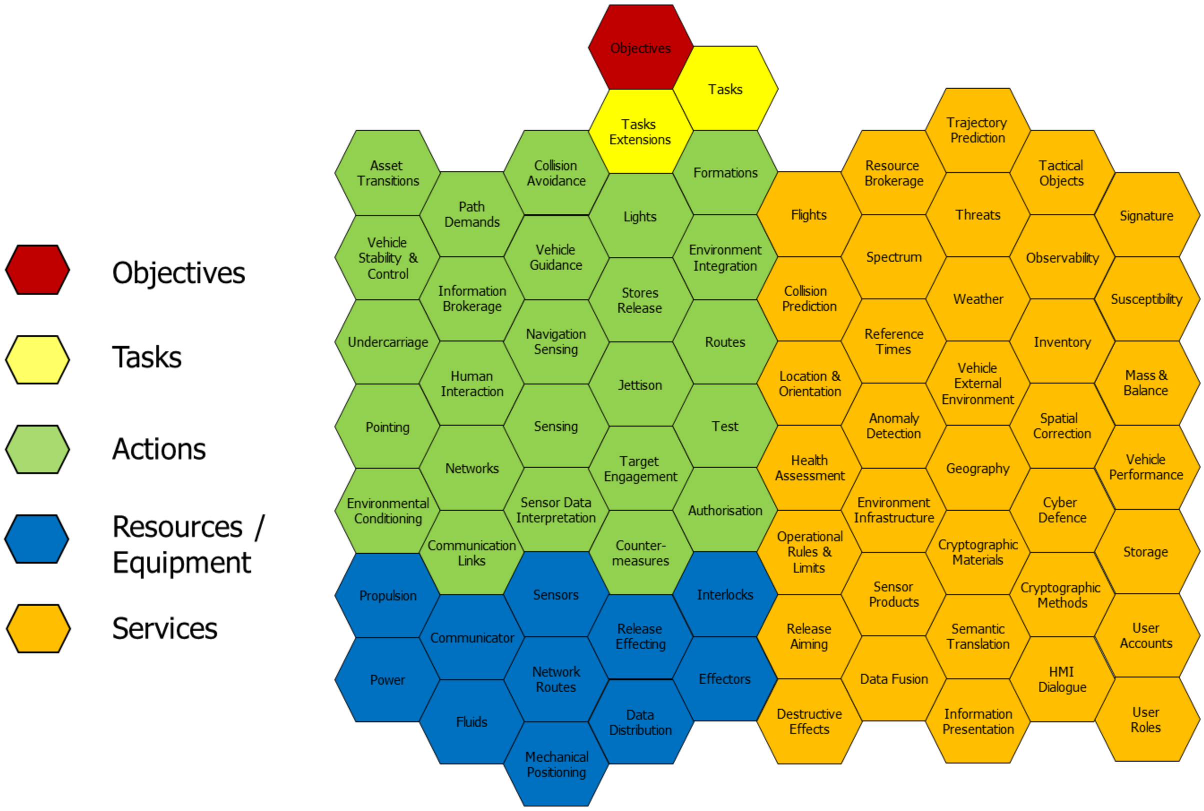


Figure 19: Component Applicability to the Control Architecture Layers

Figure 20: Control Architecture Layer Requirement Breakdown for a Full System Entirely Based on the PRA shows how components across different layers typically interact specifically when flowing down requirements. The figure represents a system that is developed entirely using PYRAMID components and so interactions with the most basic equipment that observes and affects the environment are supported using resource components. Whilst requirement interactions between components in the same layer are not shown they are still allowed and expected.

Requirements at any level of abstraction can be placed on the system and so not all of the control architecture layers will be used in all circumstances.

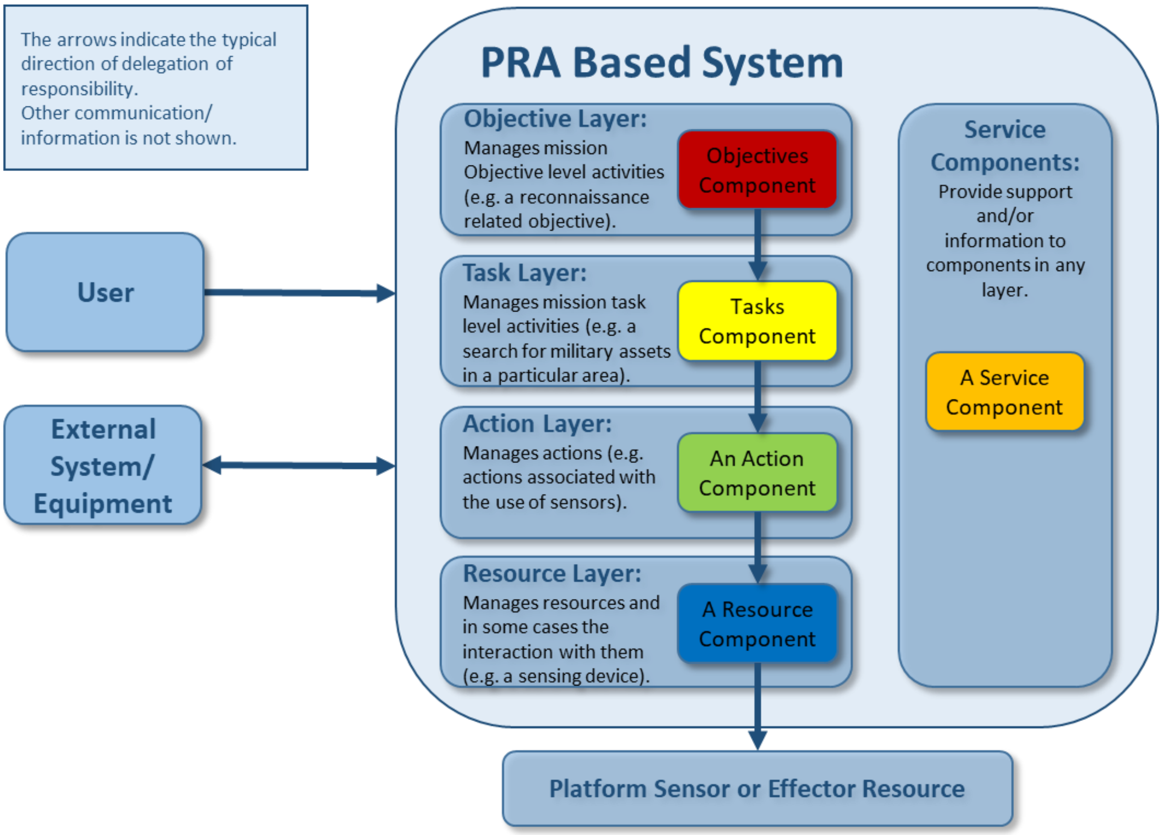

The arrows indicate the typical direction of delegation of responsibility.
Other communication/ information is not shown.
User
External System/ Equipment
PRA Based System
Objective Layer:
Manages mission Objective level activities (e.g. a reconnaissance related objective).
Objectives Component
Task Layer:
Manages mission task level activities (e.g. a search for military assets in a particular area).
Tasks Component
Action Layer:
Manages actions (e.g. actions associated with the use of sensors).
An Action Component
Resource Layer:
Manages resources and in some cases the interaction with them (e.g. a sensing device).
A Resource Component
Service Components:
Provide support and/or information to components in any layer.
A Service Component
Platform Sensor or Effector Resource

Figure 20: Control Architecture Layer Requirement Breakdown for a Full System Entirely Based on the PRA

In many real world cases a system will be developed from a combination of PRA based parts and non PRA based parts. For example, it may include re-hosted legacy software, or software on off-the-shelf equipment. Furthermore, different parts of a system may all be PRA based, but may be developed separately and could therefore be treated as separate systems. Therefore, a key concept of the PRA is that any component can interact with the software on other equipment or systems. In other words, the external equipment or systems may fulfil the role of one or more PRA components and so it would be expected to see other PRA components interacting with these external equipment or systems in a similar way to how they would interact with the equivalent PRA component whose role is being fulfilled. This includes the breakdown of requirements through the system(s).

The resource components do not provide an interface with the execution platform since any PYRAMID component can interact with the middleware or operating system that provides access to the hardware on which the component is loaded and the general software services that it provides. Likewise, not all resource components are responsible for the direct interaction with their associated resource since the physical interaction is achieved through the use of other resources. For example, a component responsible for the movement of a fuel resource may achieve the movement through the use of valve and pump resources which are handled by a different resource component. Refer to the Resource Management policy in the Description Document Ref. [2] for detailed information.

The PRA also specifies a separation of Human Machine Interface (HMI) specific components from non-HMI components so that HMI parts of a system can be developed, updated, or reused separately from the non-HMI part of a system. Refer to the Human-Machine Interface policy in the Description Document Ref. [2] for detailed information.

3.6.2 Component Interactions

The PRA components have been developed to be independent of each other and so the subject matter of each component is distinct from any other component. This means that whilst different components will reason about the same objects in the real world, they will view them purely from the perspective of their subject matter. For example, a missile loaded onto a wing is understood in different ways by different components; to one component it is a mass that can be detached, to another it is something that can provide a destructive effect, and to a third it is a system that can be communicated with.

These different views of the same object or concept are referred to as being different counterparts of the object or concept. When data is transferred between components it may be interpreted differently by different components, often requiring some form of data translation which is performed by a bridge. The bridge ensures that the data is transferred to the required place and therefore implements the so called 'counterpart relationship'; i.e. the association between one component's counterpart and another component's counterpart.

This approach is essential to maintain independence between components.

3.6.3 Common Features of Components

The Component Composition section in Appendix B of the Description Document Ref. [2] describes patterns of interactions between components that support the policies, shown as a series of use cases and service patterns. The Component Composition section expresses the common features of components that support these patterns, including:

- Responsibilities
- Subject matter semantics
- Services

The behaviour exhibited by individual components is defined by their responsibilities and services. The Component Composition responsibilities and services provide a level of commonality across all components, ensuring they can work together. Most components will have many, if not all, of the features listed there, specialised in terms of their subject matter.

4 Introduction to Policies

The architectural policies are split into four categories:

- **Architecture Wide Policies:** These describe ways of developing system-wide aspects of a PRA based system in a way that supports the PYRAMID KURs.
- **Specific Policies:** These describe how the components are intended to be integrated to provide specific functionality in a way that supports the PYRAMID KURs.
- **Modelling Principles:** These support a deployment of components in meeting the PYRAMID KURs.
- **Safety and Security:** These describe how safety and security have been considered during the development of the components.

Since policies are concerned with architectural principles, some technical details are left out for clarity, specific technical implementation detail of policies is the responsibility of Exploiting Programmes to define and apply. Diagrams in policies generally abbreviate how service connectivity is illustrated. In particular, although connections between component services are always implemented by bridges, these are not shown except where they are the specific focus of the policy.

The components capture design decisions or specific design statements resulting from the application of these policies.

4.1 Scope Summaries for Architecture Wide Policies

- **Control Architecture:** Describes a control architecture that has been embodied in the PRA, enabling an Exploiting Platform to implement system-wide control through the use of a layered architecture.
- **Constraint Management:** Explains how a PRA-based system can keep within the constraints that limit its behaviour.
- **Dependency Management:** Describes how the PRA has been designed to manage dependencies between components so that mission objectives can be achieved.
- **Autonomy:** Explains what is meant by autonomy within PYRAMID and how it applies to the PRA, including the relationship with authorisation for carrying out actions.
- **Health Management:** Explains what is meant by health and how the PRA enables a system to manage situations of reduced health.
- **Capability Assessment:** Explains how a PRA-based system assesses its ability to perform its designed functions as internal system factors change.
- **Multi-Vehicle Coordination:** Describes how different vehicles can interact in a coordinated manner through the use of components, including instances of the same component, deployed across different vehicles.
- **Interaction with Equipment:** Explains how the PRA has been designed to enable an exploiting system to interact with equipment, including determination of equipment capability and control of equipment resources.
- **Resource Management:** Explains what is meant by resources and how resources can be managed when there are multiple demands on them.

- **Operational Support:** Explains how the PRA can be used for purposes beyond the execution of a mission. It describes a range of such uses, for instance mission preparation, and the components that would support these.
- **Storage:** Describes the mechanisms provided by the PRA to enable storage of data, and the interactions of the components with storage facilities provided within a deployment of the PRA.
- **Recording and Logging:** Defines the means by which components identify information that is needed for current or future use and therefore the information that needs to be retained through recording or logging. It discusses how a component knows how long information should be retained for and if it is no longer required.

4.2 Scope Summaries for Specific Policies

- **Cyber Defence:** Describes how the PRA can be used to provide the system with a level of security monitoring and protection from unauthorised interactions, including how components should work together to protect against different types of cyber attack.
- **Human-Machine Interface:** Introduces the HMI components and explains how the HMI components can be used together to support interaction between human users and system (i.e. machine) elements within an Exploiting Platform.
- **Interfacing with Deployable Assets:** Deployable assets are hardware which can be deliberately separated from the Exploiting Platform during a mission. The policy describes interfacing with deployable assets, before and after separation, from a PYRAMID compliant Exploiting Platform.
- **Tactical Information:** Explains how the PRA supports the handling of sensor data and associated tactical information.
- **Test:** Defines the ability to support testing which is provided by the PRA, including how components support different types of test at various levels of system capability. The policy scope is restricted to self-testing of a PRA deployment.
- **Use of Communications:** This policy explains how communications capability may be used by PRA components and is agnostic to components deployed on the same or different platforms, communicating directly or through the use of the communication infrastructure. This includes how components with differing levels of communications ‘awareness' interact with components which provide communications capability.
- **Data Exchange:** Explains the exchange of data and information between systems, at least one of which is PYRAMID compliant, and the interactions of the components which provide distribution facilities to support this.

4.3 Scope Summaries for Modelling Principles

- **Component Connections:** Explains how components can be connected in a deployment of the PRA to combine components to produce complex systems.
- **Component Extensions:** Defines what is meant by a component extension, considers their benefits and provides guidance on their appropriate usage. Extensions provide a method of enhancing, specialising, or extending the subject matter, in a way which reduces the size of system changes supporting rapid capability update.

- **Data Driving:** Explains how configuration data could be utilised in a deployment of the PRA to modify component behaviour to cater for different conditions; for example, to cater for different role fit equipment or operational requirements.

4.4 Safety and Security Policies

- **Safety Analysis:** Explains how safety analysis has been applied to the PRA and why. Note that the safety analysis does not place any requirement on an Exploiting Programme.
- **Security Approach:** Explains how security analysis has been applied to the PRA and why. Note that the security analysis does not place any requirement on an Exploiting Programme.

5 PYRAMID Key User Requirements

This section shows how the PRA supports the PYRAMID Key User Requirements (KURs). For each PYRAMID KUR, the PRA Architectural Policies which support the fulfilment of these goals are outlined. Refer to the Description Document Ref. [2] for detailed descriptions of the Policies and Components referred to in the fulfilment section for each PYRAMID KUR.

The PRA does not provide a complete fulfilment of the PYRAMID KURs at the Platform Independent level defined within this iteration of the PRA.

Complete fulfilment of the PYRAMID KURs requires additional elements outside the scope of the PRA, which are dependent on the end user (e.g. UK MOD) and the party responsible for developing a specific deployment of the PRA.

5.1 Configurable

5.1.1 Definition

The User shall be able to configure instances of the PYRAMID System such that mission capability exploits available and emerging hardware, software and data services with minimal impact on the qualified system. Configurable items include - System Behaviour with varying levels of autonomy, during both planning and airborne phases, across all areas associated with mission/flight management; sensor, weapon and defensive aids employment; data handling and appropriate equipment interfaces. Additionally, the user shall be able to configure necessary elements of the Operator-Mission Interface (OMI), including Human-Machine Interface (HMI) aspects.

The PRA supports this PYRAMID KUR by providing the ability to change the component behaviour at deployment or operational time.

5.1.2 Fulfilment

- The PRA can be configured for vehicles with different capabilities, behaviour, equipment and HMI interfaces.
- A deployment of PRA components can be structured to support different types of task, action and resource, taking into account what can and cannot be done at different times during a mission. The Control Architecture and Dependency Management policies describe how different types of component can be configured to achieve the mission goals, with changes to applicable rules covered in the Constraint Management policy, and flexible resource sharing in the Resource Management policy.
- The components can be used in different configurations to meet the needs of a specific implementation. The Component Connections policy describes how different configurations can be connected based on the needs of the system. The PRA uses data-driven design principles, in accordance with the Data Driving policy, which allow the properties of a specific implementation to be incorporated as data rather than by modifying the function of components. As such it supports modifying properties associated with specific role fit equipment on a mission-by-mission basis with mission data loads. The use of data-driven elements to control recording and software logging in accordance with the Recording and Logging policy further supports this.

- The Autonomy policy describes how different levels of autonomy can be configured, including potential limits on when and how each level of autonomy can be utilised under a deployment of the PRA.
- The components can be configured to coordinate multiple independent vehicles in accordance with the Multi-Vehicle Coordination policy, and different Deployments of the PRA may require different methods of achieving coordination between vehicles.
- Separation of non-HMI from HMI logic, in accordance with the Human-Machine Interface policy, allows changes (e.g. reconfiguration, re-porting, scaling) to be made to the HMI and non-HMI parts of the system with minimal impact on other components. It also allows reuse of non-HMI components in Exploiting Platforms with different HMI requirements.
- PRA components can be used in different stages of a mission, including pre- and post-sortie. Their use within an operational support environment is described in the Operational Support policy. The Tactical Information policy covers how components can be configured to adapt to information about the objects encountered during a mission.
- The components can be set up to communicate with other components or systems, regardless of hardware platform, as detailed within the Use of Communications policy.
- The components involved in defending against cyber threats can be configured depending on the threats likely to be encountered by the Exploiting Programme, as per the Cyber Defence policy.
- Systems that exploit PRA components will have different resource requirements and resource availability. The Resource Management policy describes how the problem of resource availability, allocation and conflict resolution is handled within systems of differing characteristics.

5.2 Exploitable

5.2.1 Definition

The User shall be able to deploy the PYRAMID System architecture and associated software components across multiple National, Collaborative and Export Equipment Programmes, utilising the open reference architecture to gain maximum leverage for the UK supplier base and maximum return on investment on this re-usable capability.

The PRA supports this PYRAMID KUR by providing the ability to integrate and utilise the PRA on various Exploiting Platforms.

5.2.2 Fulfilment

- The Mission Context Scope defines the scenarios and mission environments which have been considered during the development of the PRA. These cover a wide range of scenarios and environments enabling the PRA to be used on a wide range of platforms and meeting the needs of national, collaborative and export Exploiting Programmes.
- The PRA has been developed in a way that allows PYRAMID deployments to be incorporated into Exploiting Platforms that do not exclusively use the PRA, such as legacy platforms, thereby maximising the opportunities for its use. The Deployment Guide acknowledges the potential to use a PYRAMID deployment in a legacy architecture.

- The Deployment Guide also describes considerations for legacy code to be reused within or in support of a PYRAMID deployment.
- PRA-based systems will interact with other platforms. The Operational Support policy covers how this is done in pre-mission and post-mission timescales. Coordination with other platforms to achieve mission objectives is the subject of the Multi-Vehicle Coordination policy.
- The Resource Management policy explains how the PRA separates out the problem of resource availability, allocation and conflict resolution, contributing to the ability to design and tailor systems that are suited for a wide variety of UK, collaborative and export programmes.
- The PRA can be tailored for use on various Exploiting Platforms and to meet different end user needs:
    - The PRA can be tailored for vehicles with different capability, including performance characteristics, different (vehicle) equipment and different operating philosophies (i.e. manned or unmanned). The Interaction with Equipment, Interfacing with Deployable Assets and Use of Communications policies explain how the components of the architecture have been defined to allow the PRA to interact with the environment.
    - The PRA can be tailored for use in different physical environments, allowing different Exploiting Programmes to tailor components such as Weather, Vehicle External Environment and Geography to a particular region relevant to a given Exploiting Programme.
    - The Tactical Information policy shows how a PRA component can be tailored to achieve different goals depending on the requirements of the Exploiting Programme
- The PRA contains guidance to support tailoring activities:
    - Extension Components, in accordance with the Component Extensions policy, separate out deployment-specific concerns from general system behaviour. This allows the PYRAMID components to be tailored for multiple Exploiting Programme needs without modification to core system behaviour, with elements of a component bespoke to a particular Exploiting Programme customer added to the architecture as an extension.
    - Data Driving, in accordance with the Data Driving policy, provides the ability to integrate and utilise the PRA on various Exploiting Platforms by allowing tailoring for vehicles with different capability (including performance characteristics), different (vehicle) equipment and for use in different physical environments.
- The Deployment Guide describes tailoring opportunities in the development process, while briefings and training for Exploiters also support this.
- The components can be configured to operate in accordance with varying rules and constraints in accordance with the Constraint Management policy, such as the different operational doctrine of an export customer.
- The Safety Analysis policy provides flexibility in order to be successfully exploited under differing international safety standards, adapting to whatever legislative environment is applicable to a deployment of the PRA.

5.3 Flight Certifiable

5.3.1 Definition

The user shall be able to have instances of the PYRAMID System certified for Flight (against Civil and Military regulations) as part of a complete system deployment.

The PRA supports this PYRAMID KUR by supporting the ability to create and provide evidence in support of the certification for an Exploiting Platform. The PRA does not provide evidence for a specific implementation but its structure supports carrying out certification (and re-certification after system change) in a structured and straightforward manner.

5.3.2 Fulfilment

- Components within the PRA have been subjected to an initial safety assessment in accordance with the Safety Analysis policy.
- Risks which can be understood and addressed at an architecture level are defined in the Safety Analysis policy and in the safety analysis of specific components. However risks which are associated with a specific Exploiting Platform or technology are part of a specific implementation and fall within the deployment domain.
- The component structure includes tightly scoped roles and responsibilities, which supports the segregation and separate certification of subject matters with different safety requirement levels. Formally modelling the relationship between components, as described in the Component Connections policy, can contribute to modular safety cases.
- The ability to test during maintenance and during missions supports certification. The Test policy supports providing Exploiting Platforms with the ability to determine and prove limits of capability and the Recording and Logging policy supports the capture of performance data and safety-critical logs.
- The Storage policy provides a framework which enables conformance to legislation, e.g. to enable storage of specific aircraft performance parameters on a Crash Survivable Memory Unit (CSMU).
- The Tactical Information policy allows Exploiting Platforms to be more easily certifiable by separating the control of sensor data handling and the sensor data handling itself.
- Different levels of autonomy may be needed during the early stages of development/deployment to allow automation technology to be matured. The Autonomy policy describes the flexibility needed to allow a human user to take back control from a system with high autonomy under certain circumstances, thus enabling systems that would otherwise not be flight certifiable, or very hard to certify, to be certified.
- The PRA defines the concept of extensions within the Component Extensions policy. Extensions allow functionality within a PRA component to be separated in ways that minimise the impact on certification and recertification, for example by segregating safety critical functions from non-safety critical functions, functions that are updated frequently from those that are not, or functions that need to be certified by different authorities.

- The component security considerations, provided in response to the Security Approach policy, highlight where security measures for continued airworthiness might need to be applied and where an attack on security attributes may cause a safety concern.

5.4 Incorporate Future Growth

5.4.1 Definition

The User shall be able to further develop the PYRAMID System reference architecture and deployed instances of the PYRAMID System through evolutionary changes in response to operational and non-operational drivers with the minimum of resource and time overhead.

The PRA supports this PYRAMID KUR by providing for the ability to implement changes to the PRA design, and by providing concepts and the flexibility for deployed systems to adapt to evolutionary changes. Note that other PYRAMID KURs (Scalable, Resilient against Obsolescence & Configurable) also address adaptability of the architecture in an Exploiting Programme.

5.4.2 Fulfilment

- The PRA can be adapted by deleting components or adding new components. Interaction Views illustrate the top-level interactions between components in specific use cases, allowing understanding of the impact of restructuring or removing components, and the components needed to provide a given element of functionality.
- Through the Control Architecture policy, the PRA provides a framework within which different types of task, action and resource can be added with minimal impact. When new functionality is introduced, the Constraint Management and Dependency Management policies show how this can be done without affecting the interpretation of rules or dependencies.
- The potential use of component extensions in accordance with the Component Extensions policy, in a deployment of the PRA, allows components to be further extended or specialised hence minimising impact on the component and its behaviour.
- The Interaction with Equipment policy provides a framework through which PRA components can be used to interact with widest possible variety of equipment of any complexity level. It describes how the PRA has been designed to accommodate adding a new item of equipment to an Exploiting Platform with minimal modification.
- The Resource Management policy describes how the problem of resource availability, allocation and conflict resolution is kept apart from other areas of knowledge, allowing new functionality to be added without requiring changes to the component that deals with brokering resources.
- The Recording and Logging policy framework facilitates ease of updating which data is to be retained, for example, to account for new components in a deployment of the PRA.
- The use of data driving by a deployment of the PRA, in accordance with the Data Driving policy, supports ease of modification, minimising impact to existing parts of an exploiting platform, for example, to account for new configurations.
- The architectural elements that perform the handling of sensor data can be extended to support many different processing algorithms, as per the Tactical Information policy.

- The Storage policy provides a framework within which Storage Media can be added or removed to cater for changing data retention requirements.
- As PRA components are communications agnostic (excluding the comms components), the Use of Communications policy allows for a simple and efficient introduction of new capabilities.
- Through application of the Component Connections policy, components and their relationships can be more easily changed to incorporate future growth.

5.5 Resilient Against Obsolescence

5.5.1 Definition

The User shall be able to deploy the PYRAMID system onto a range of underlying computing platforms, and therefore it shall not be tied to the Original Equipment Manufacturer's computing solution.

The PRA supports this PYRAMID KUR by supporting the ability to port the system software on to different hardware and operating systems with minimal rework.

5.5.2 Fulfilment

- The components as defined are independent of Execution Platform.
- The Health Management policy defines how implementations of the PRA can handle run-time errors in the underlying Execution Platform, providing a capability to efficiently identify and resolve issues.
- The Recording and Logging policy includes data-driven elements described in the Data Driving policy, thus allowing the PRA to be easily adapted between different Execution Platform capabilities for the logging and recording of activities or events.
- The PRA separates concerns between PIM and PSM such that it is Execution Platform agnostic, and the Deployment Guide provides appropriate guidance in relation to the deployment lifecycle. This supports the porting of components running on obsolete hardware to new Execution Platforms with minimal impact on the system and so minimal rework.
- The Resource Management policy supports resilience against obsolescence through separating assigning resources from other concerns and catering for different resourcing strategies appropriate to different hardware and technology.
- A clear definition of the relationship between components and the use of bridges, as proposed within the Component Connections policy, insulates components from the underlying communications mechanism, which further supports portability from the underlying Execution Platform.

5.6 Scalable

5.6.1 Definition

The User shall be able to deploy complete (or selectable part) instances of the PYRAMID System into integrated system solutions that form part of a range of families of systems that span the full range of platform system classes.

The PRA supports this PYRAMID KUR by providing the ability to use varying numbers of components to produce system deployments.

5.6.2 Fulfilment

- The PRA can be deployed in simple systems using a small subset of the available components or in complex systems using a large number of the available components. Interaction Views illustrate the top-level interactions between components in specific use cases, allowing understanding of the impact of restructuring or deleting components, to provide a given element of functionality.
- The PRA provides a framework which allows different numbers of components to be deployed flexibly through application of the Control Architecture, Capability Assessment and Dependency Management policies.
- The Multi-Vehicle Coordination policy supports scalability by allowing capability to be split between participating vehicles, e.g. laser designating on one, weapon release on another.
- The components can be configured to operate in accordance with varying rules and constraints in accordance with the Constraint Management policy.
- Extension Components, in accordance with the Component Extensions policy, allow for the simplification and elimination of resourcing overheads by making it possible to omit capabilities or support improved algorithms more suited to the needs and resources available to a given deployment.
- Separation of non-HMI from HMI logic, in accordance with the Human-Machine Interface policy, allows standard interfaces and controls to be used independently of the scope or complexity of the rest of a particular deployment.
- The PRA, in accordance with the Health Management policy, allows systems with different numbers of components to be able to assess and manage their health in a consistent way that is not tied to a wider system architecture.
- The components control recording and software logging locally in accordance with the Recording and Logging policy and are not dependent on a wider architecture for these activities.
- The Storage policy provides a framework that allows any number of components to store data on any storage media available.
- By allowing equipment to connect at different levels of the control architecture as per the Interaction with Equipment policy, the PRA supports scalability by allowing alternate designs where functionality can be held within the PRA-based system or in the external ‘smart’ equipment.

- As PRA components are by default communications agnostic (they are not aware of where comms signals originate or go, or the route via which those comms signals travel), application of the Use of Communications policy allows for systems of varying complexity and numbers of components to be designed.
- The Resource Management policy provides a variety of management strategies which can be employed depending on the types of resources being managed, including multiple strategies within the same deployment. This facilitates variable levels of complexity across different deployments.
- The Cyber Defence policy accommodates scaling to cover the necessary elements of detection, defence and recovery for a deployment.

5.7 Security Accreditable

5.7.1 Definition

The user shall be able to have instances of the PYRAMID CMS security accredited by Defence Assurance and Information Security (DAIS) against MOD security policy and with security risks mitigated to a level acceptable to the relevant security risk owner.

The PRA supports this PYRAMID KUR by supporting the ability to create and provide evidence in support of the security accreditation of an Exploiting Platform. The PRA does not provide evidence for a specific implementation but its structure supports carrying out accreditation in a structured and straightforward manner.

5.7.2 Fulfilment

- Components within the PRA have been subjected to an initial security assessment in accordance with the Security Approach policy.
- Typical security risks that can be understood and addressed at an architecture level are defined in the Security Approach policy and Cyber Defence policy and in the security considerations for specific components. However, risks that are associated with a specific platform or technology are part of a specific implementation and fall within the exploitation domain, and as such are out of scope of the PRA security analysis.
- Formally modelling the relationship between components, as described in the Component Connections policy, will allow for clearer vulnerability analysis and security case design by the Exploiting Programme.
- The component structure includes tightly scoped roles and responsibilities, which supports the segregation and separate accreditation of functions with different security requirement levels.
- The PRA is accompanied by the Deployment Guide and by briefings and training for Exploiters, which support tailoring the PRA to specific implementations and provide advice on how to address security requirements within a deployment (for example through the use of multiple component instances).
- The ability to test during maintenance and during missions supports security accreditation. The Test policy ensures that Exploiting Platforms have the required testing capability, and the Recording and Logging policy and Storage policy allow software logging and test results to be securely captured.

- The HMI architecture defined in the Human-Machine Interface policy supports partitioning for safety and security reasons whilst providing a means to achieve a consistent and seamless user interface.
- The components can be configured to operate in accordance with varying rules and constraints in accordance with the Constraint Management policy.
- The Data Exchange policy caters for the application of Confidentiality, Integrity and Availability (CIA) handling rules to important information.

5.8 Utility Across A Range Of Missions

5.8.1 Definition

Dependent upon the core capabilities of the host air vehicle, the User shall be able to participate in a broad gamut of operations including: Intelligence, Surveillance and Reconnaissance (ISR); Command and Control (C2); Attack; and Control of the Air. The User shall also have the ability to conduct communications relay and where present operate any associated defensive aids systems.

The PRA supports this PYRAMID KUR by providing the ability to create a system deployment for various mission scenarios and organisational structures using the components.

5.8.2 Fulfilment

- Exploitations of the PRA can be used in different types of missions and to support different mission behaviour. The Mission Context Scope defines the scenarios and mission environments which have been considered during the development of the PRA.
- The components can be used in different configurations to meet the needs of a specific implementation in accordance with the Control Architecture policy. The PRA uses data-driven design principles, in accordance with the Data Driving policy, which allows the properties of a specific implementation to be incorporated as data rather than by modifying the function of components. As such it supports modifying properties associated with specific role fit equipment on a mission-by-mission basis with mission data loads.
- The Capability Assessment policy provides a framework for establishing a PRA system that has awareness about its abilities to carry out different missions, objectives and actions.
- In accordance with the Dependency Management policy, PRA components are able to manage their dependencies and work together to adapt to and fulfil a variety of mission types.
- The Operational Support policy shows how deployments of the PRA can be set up to carry out specific missions, and how organisational structures and processes can be accommodated. Additionally the User Roles component defines the various user roles that different users can be assigned.
- The Autonomy policy describes how levels of autonomy available to implementations of the PRA and limits on each level of autonomy can be utilised.
- The PRA can be used to support different legislative requirements. The Operational Rules and Limits component interprets operational rules for an implementation of the PRA. The Environment Infrastructure component defines the operational environment including both civil and military aspects.

- Extension Components, in accordance with the Component Extensions policy, enable the PRA to be utilised for multiple operational scenarios.
- Different mission scenarios may require different methods of achieving coordination between vehicles. The Multi-Vehicle Coordination policy enables different coordination behaviour of multiple independent vehicles on different types of mission.
- Being able to integrate different deployable asset allows an Exploiting Platform to be utilised for multiple operational contexts as described in the Interfacing with Deployable Assets policy.
- Effective communications is the foundation to achieve any operational scenarios and is the subject of the Use of Communications policy.
- Application of the Test policy supports deployments of the PRA to be used in different types of mission, where it is necessary to determine and prove that the limits of system capability meet the requirements of the mission.
- HMI in accordance with the Human-Machine Interface policy enables human oversight of system operation.
- Some mission scenarios may be more susceptible to certain attack vectors than others. The cyber attack examples covered in the Cyber Defence policy apply to a range of operational scenarios.

5.9 Additional Considerations

As well as supporting the PYRAMID KURs there was an additional requirement, titled ‘supportable’, which was considered desirable for the PRA to support.

5.9.1 Supportable

5.9.1.1 Definition

NOTE: This is an additional requirement on the PRA, not a PYRAMID KUR.

The PRA in its potential deployments can be supported efficiently and effectively throughout the Exploiting Programme's lifetime, this can be summarised as the ability of the PRA to support reliability, maintainability and fault diagnosis.

5.9.1.2 Fulfilment

- The PRA can be used to create maintainable systems. The Anomaly Detection component, and Capability Assessment, Recording and Logging, Storage, Test and Cyber Defence policies show how an implementation of the PRA can identify anomalies and potential cyber threats, monitor and predict a loss of capability, record safety and security-critical data and support the implementation in taking corrective action.
- The PRA can be used to create reliable systems. For example, the Asset Transitions component is responsible for determining the system hardware and software configuration(s), and the Health Management policy defines how the health of system elements is monitored and managed. The Operational Support policy shows how deployments of the PRA can be used for maintenance and support.

- The reliability and maintainability of systems developed using the PRA contribute to their supportability. The Test policy allows for diagnostic testing during maintenance, as well as the testing of physical elements during a mission.
- HMI in accordance with the Human-Machine Interface policy enables human oversight of system operation, including during maintenance.
- The Resource Management policy explains how the problem of resource availability, allocation and conflict resolution benefits from subject specific knowledge about activity chains, allowing development and maintenance of a system in accordance with that knowledge.

6 Architecture Scope

6.1 PRA Scope

The PRA is a reference architecture for the software aspects of functionality of an air system. It enables software components to be developed and integrated into an air system, including air vehicles, supporting ground facilities (e.g. UAV control stations, mission planning and briefing facilities), and their ability to communicate.

The PRA covers areas that have traditionally been thought of as mission systems and vehicle systems.

The PRA covers the lifecycle of a mission, including:

- Planning
- Briefing and rehearsal
- Execution
- Debriefing and generation of data for post mission analysis

The PRA's components have been designed to support the PYRAMID KURs, and they have the potential for use in many applications. However, it is not expected that components will be used in high level strategic planning and tasking systems.

6.1.1 Mission Variant Scope

Figure 22: Operational Context Scope and Figure 23: Air Platform Scope show the variants of air platform type, and their operational context that were considered during the design of the PRA. Possible deployments of the PRA are not necessarily limited to the types in the diagrams. Any Exploiting Programme would however, be responsible for impact assessment and potential rework.

The UML notation for these figures is relatively simple:

- **Part Association:** represented by a black diamond. As shown in Figure 21: UML Terminology, A is composed of B and C.
- **Generalisation:** represented by a white triangle. As shown in Figure 21: UML Terminology, E and F are similar classes within the category D.

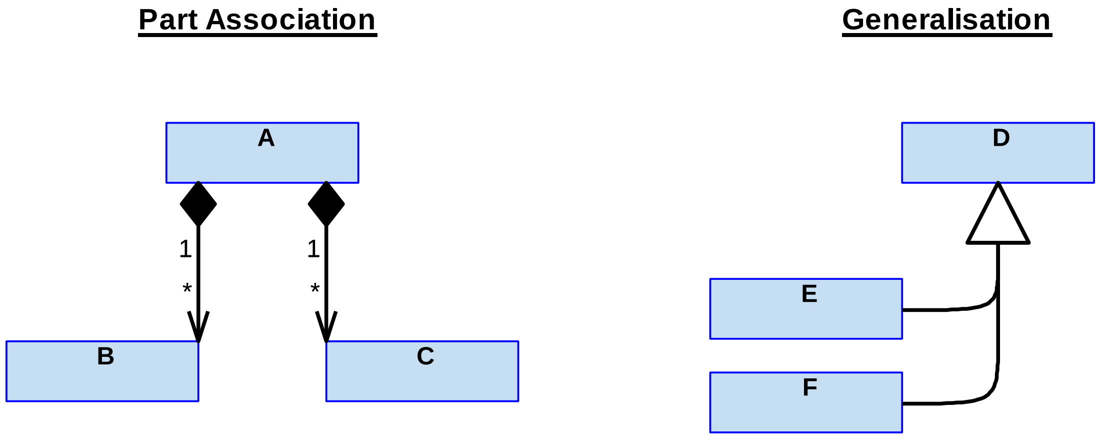


Figure 21: UML Terminology

6.1.1.1 Operational Context Scope

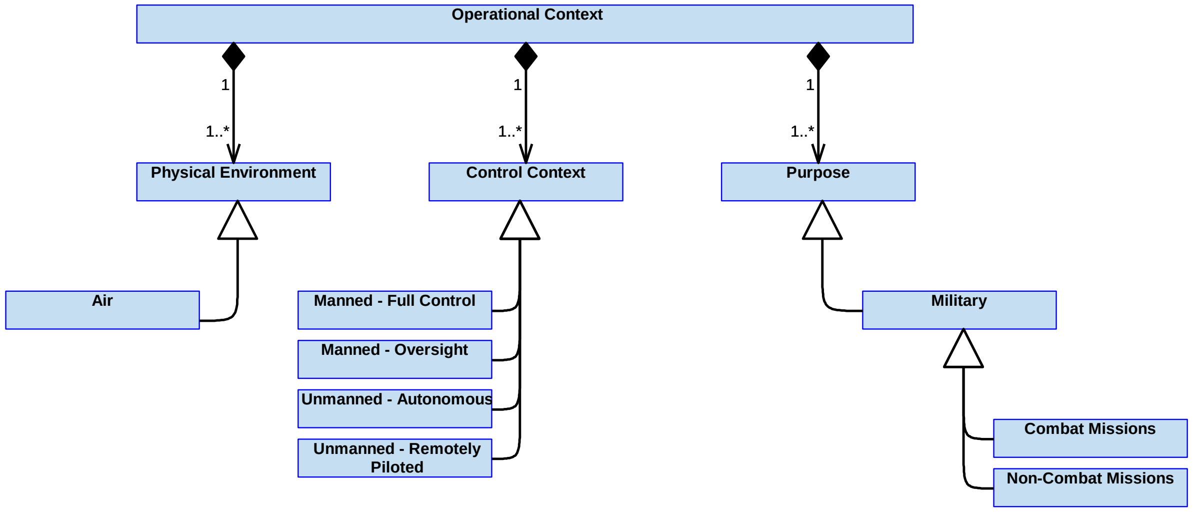


Figure 22: Operational Context Scope

Figure 22: Operational Context Scope identifies that military applications within the air environment, for both manned and unmanned platforms, have been considered during development of the PRA.

6.1.1.2 Air Platform Scope

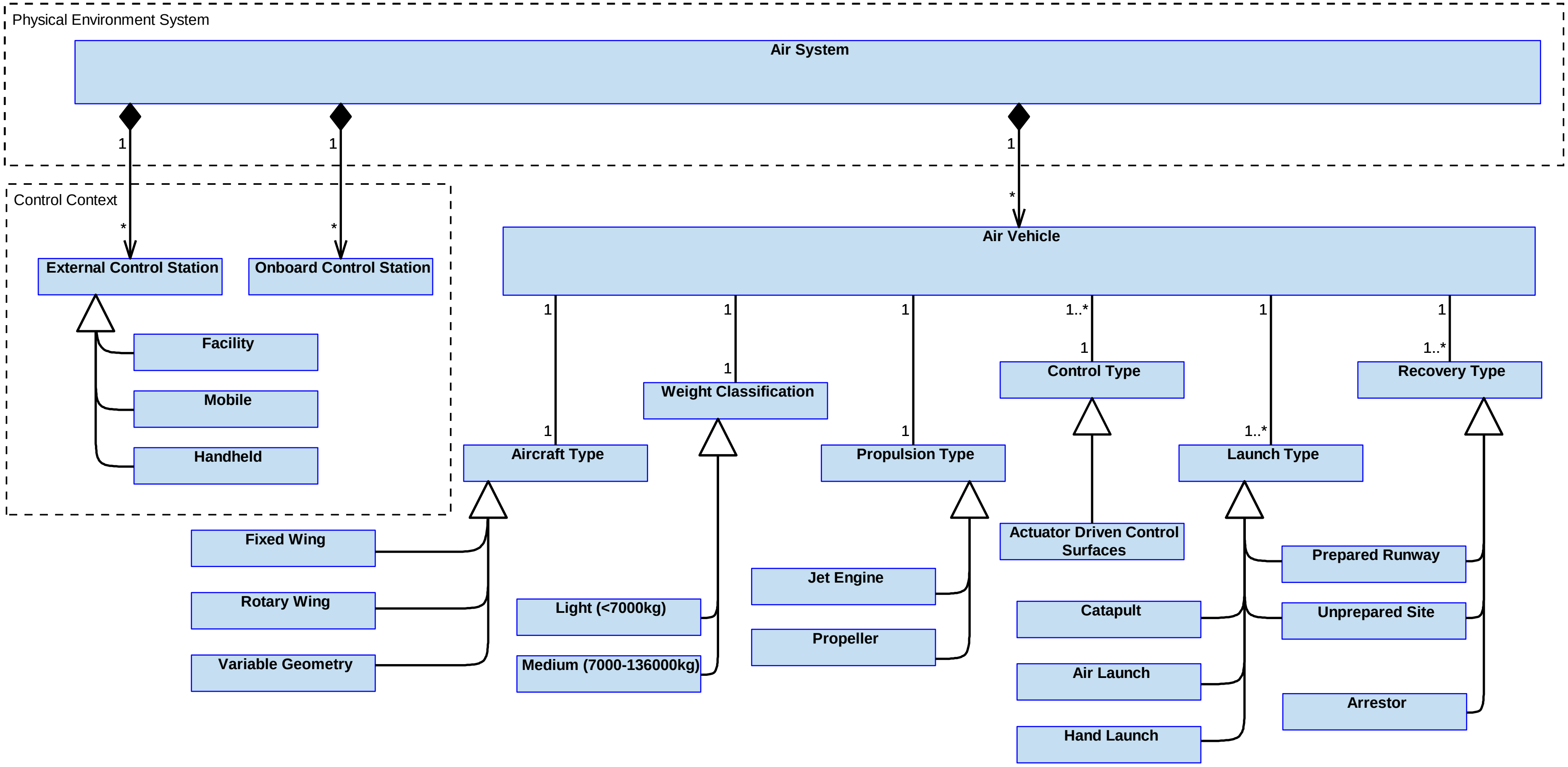
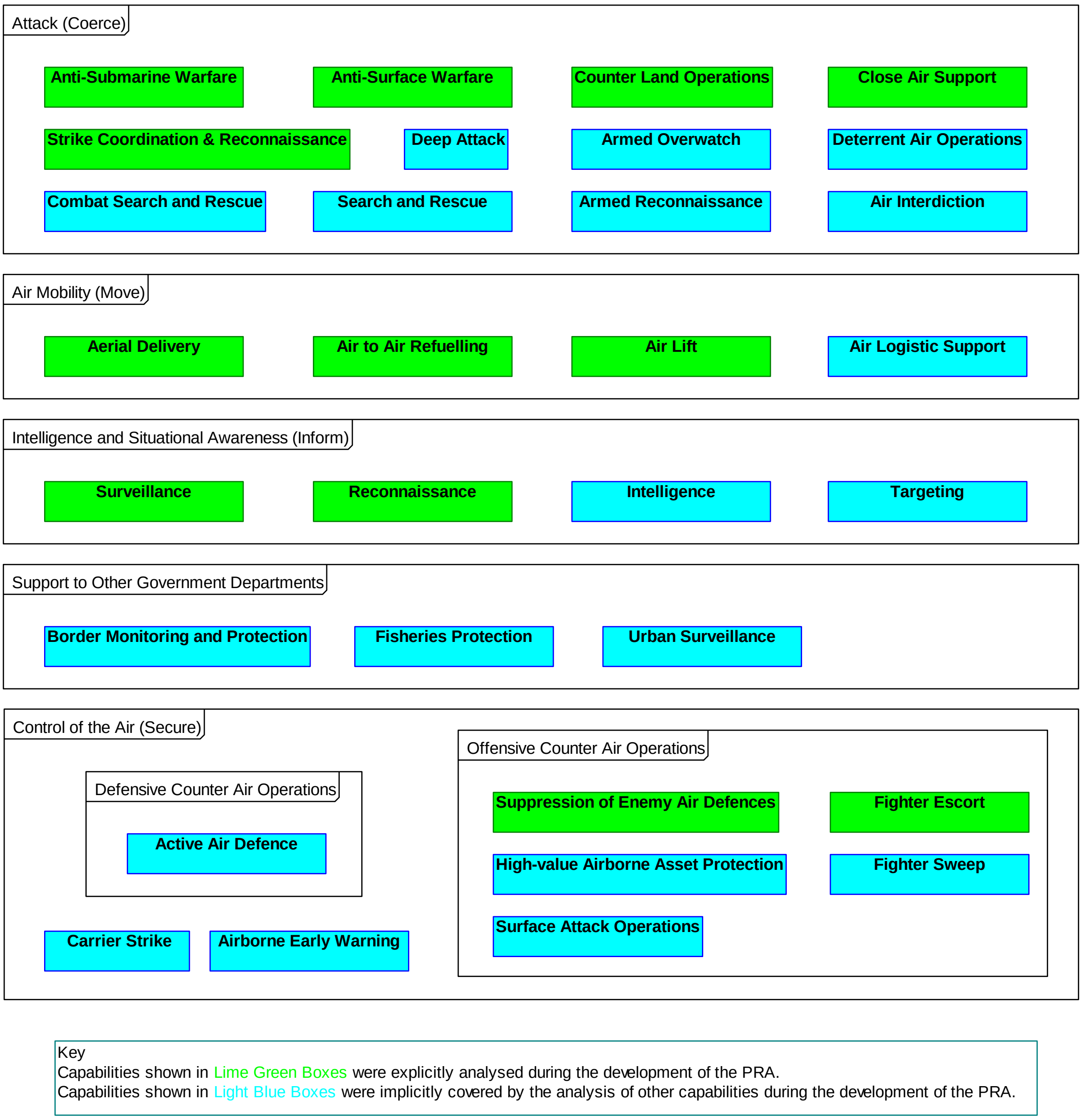


Figure 23: Air Platform Scope

Figure 23: Air Platform Scope identifies variants within the scope of an air system which have been considered during development of the PRA.

6.1.2 Mission Context Scope

Figure 24: Mission Context Scope identifies the military capabilities that have been considered in developing the PRA. These capabilities are based on the roles and missions defined in MOD UK Air and Space Doctrine Ref. [14]. Possible deployments of the PRA are not necessarily limited by these capabilities. Any Exploiting Programme would however, be responsible for impact assessment and potential enhancement.



Key
Capabilities shown in Lime Green Boxes were explicitly analysed during the development of the PRA.
Capabilities shown in Light Blue Boxes were implicitly covered by the analysis of other capabilities during the development of the PRA.

Figure 24: Mission Context Scope

6.1.3 Boundary

6.1.3.1 Mission and Vehicle Equipment Hardware

The PRA is a reference architecture aimed at software implementation. Any equipment function that takes the form of application software is within scope of the PRA, but equipment hardware, firmware, drivers or middleware are not. This potentially includes any software that handles data once it has been converted to take a digital format, as this could be handled by application software. For any Exploiting Programme there will be a boundary beyond which the PRA does not apply. It is for the designers to determine exactly where the boundary lies, and whether any equipment control software is within the PRA boundary or is part of the installed equipment. For example, the software for an Inertial Navigation System (INS) could be built from PYRAMID components, or an off-the-shelf INS solution could be used.

6.1.3.2 Processing Infrastructure

Being platform-independent means that PRA components can be used to develop components that run on any suitable computer processing hardware and software infrastructure (i.e. Execution Platform). Such infrastructure is therefore not within the scope of the PRA. The PRA expects basic infrastructure functions to be provided, such as:

- Storage media and storage implementation
- Computing management functions, including latency management
- Managing security and safety partitions
- Mitigating (as far as possible) infrastructure errors and faults

The PRA design does not preclude components from having control over these functions where required by the Exploiting Programme.

6.1.4 Safety and Security

Safety and security analysis are specific to each Exploiting Programme and will be expected to take account of the particular safety and security targets, vehicle type and operating scenarios of that programme. The PRA only provides indicative safety and security analysis. Therefore, the Exploiting Programme will be entirely responsible for demonstrating that the Exploiting Platform meets the safety and security targets applicable to the Exploiting Platform. Within the PRA, safety and security have been considered and observations recorded. However:

- The PRA does not place safety and security requirements on an Exploiting Programme.
- The safety and security considerations in the PRA are not expected to directly contribute to the Exploiting Programmes safety or security case.

See the Safety Analysis and Security Approach policies for further information.

6.2 Potential Changes

In order to make a PYRAMID based system easier to upgrade, different ways in which a system could change were identified. These change scenarios were used to identify features which would provide resilience to changes during the design phase of the PRA and are explained below.

Change scenarios identify possible ways in which Exploiting Programmes may change through life and affect how the system is designed. The scenarios are similar to use case scenarios except that they focus on requirements to develop and upgrade the system rather than how it will be used. For example, when a new sensor is acquired by an Exploiting Programme, how is the programme expected to change the system to accommodate it?

A set of possible scenarios, framed within the context of an Unmanned Air System, was collated by domain specialists (for example, weapon specialists, security specialists, safety specialists and the Dstl customer). This set was down-selected to provide a random subset spanning different areas of the system. During the design of the components, these scenarios were considered to identify possible strategies to mitigate the impact of system change in response to each scenario with the resulting design decisions or considerations captured in the component design rationale.

Appendix A: Reader Guidance Example

This appendix provides a contextual example, based on an air vehicle navigation use case, of how the PRA elements work together, to bring the various concepts discussed in the Reader Guidance section together.

The Routing Interaction View (IV) shows how an air vehicle can plot a route through the use of PRA components, taking account of specified constraints (e.g. air volumes, terrain and speed restrictions) and limits (e.g. vehicle capability).

Figure 25: Routing Interaction View shows the IV. The full details, including the associated use case, can be found in the Routing IV section of Appendix C of the Description Document Ref. [2]. The use case provides a description of the IV and defines the preconditions, sequence of events and post conditions.

The point of showing the Routing IV here is not to concern the reader with the details, but to illustrate that the IV is made up of multiple components (i.e. the Tasks, Routes, Path Demands, Environment Integration, Operational Rules and Limits, Vehicle Performance, Vehicle Guidance, Location and Orientation, Weather and Geography components). These not only feature on this IV, but many other IVs as well.

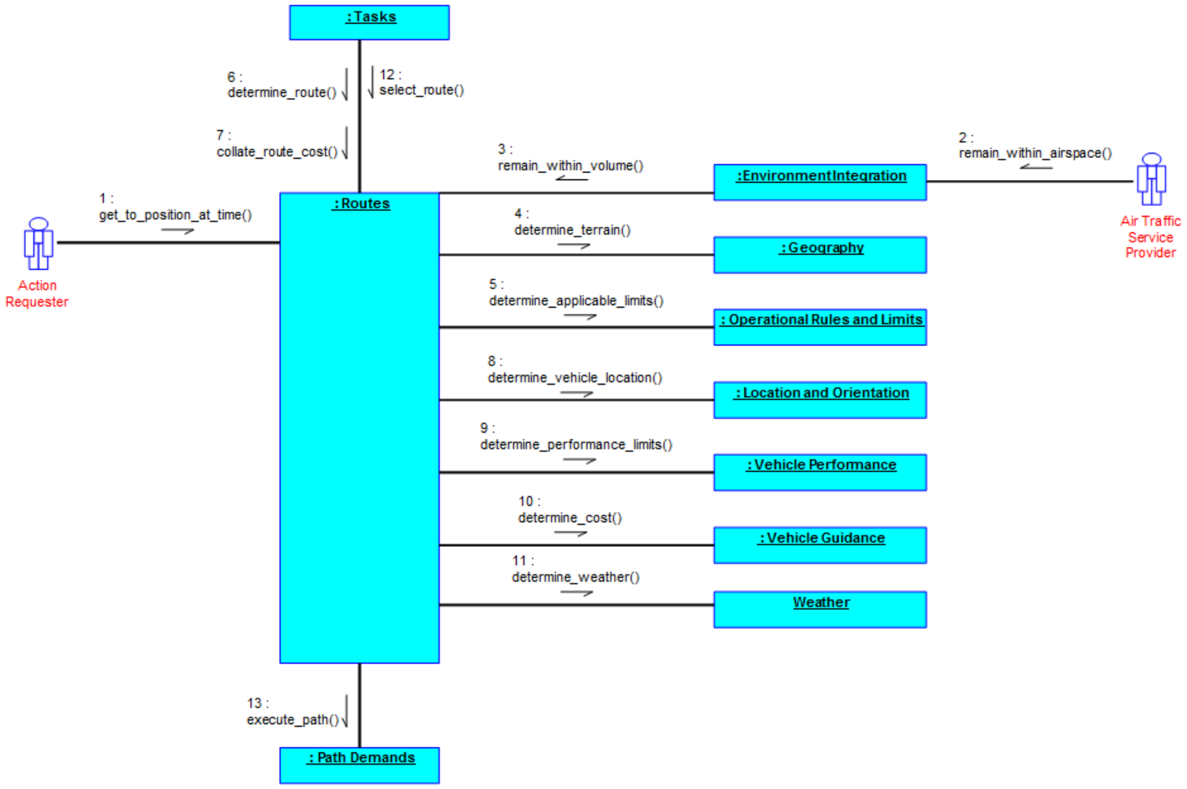


Figure 25: Routing Interaction View

The definitions of each of these components can be viewed in detail in Appendix B of the Description Document Ref. [2].

Table 1: Policy Applicability to the Routing Interaction View and its Components shows the policies relevant to the Routing IV. While all architecture wide policies are relevant, which policies are especially pertinent to the scenario and components in the IV is derived from the use case description, the component design considerations, and the allocation of components in different control architecture layers. Some policies are not explicitly identified in the component design considerations since they are applicable to all components. These are listed in the components introduction (section 3 of the Description Document Ref. [2]).

The Operational Support policy is applicable since this describes how PRA deployments on systems such as mission planning and debriefing systems can be considered, as illustrated later in Figure 26: The Routing Interaction View and its Component's Relevance to Mission/Operational Phases.

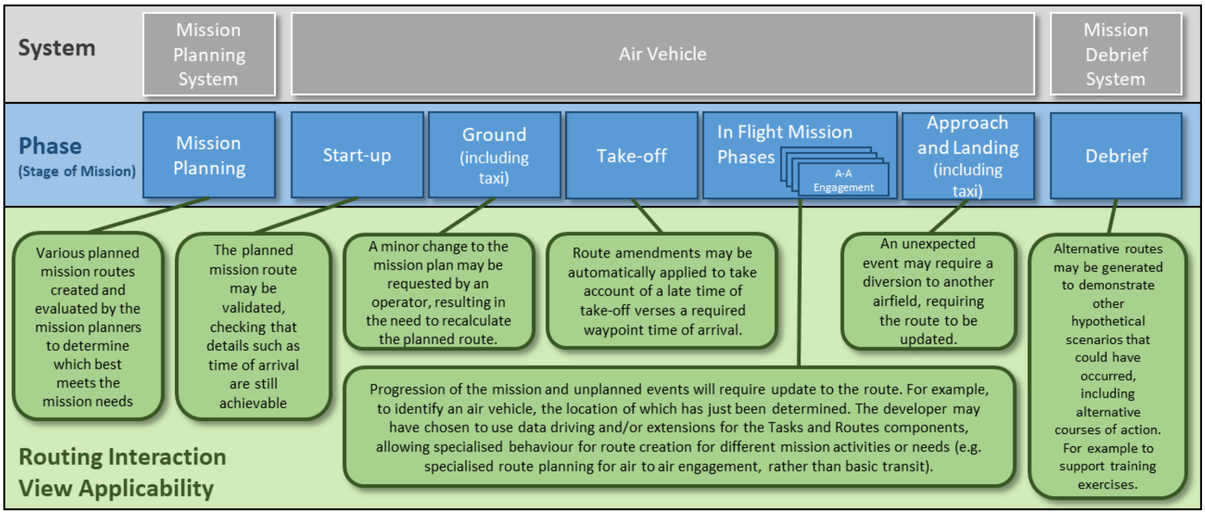
Policy Type		Component										
	Architecture Layer	Task	Action	Action	Action	Action	Service	Service	Service	Service	Service	
	Policy	Task	Routes	Path Demands	Environment Integration	Vehicle Guidance	Operational Rules and Limits	Location and Orientation	Vehicle Performance	Geography	Weather	Routing Interaction View
Architecture Wide Policies	Control Architecture	Common to all components										X
	Constraint Management	X					X					X
	Dependency Management											X
	Autonomy											
	Health Management											
	Capability Assessment	Common to all components										
	Multi-Vehicle Coordination	X	X									
	Interaction with Equipment											
	Resource Management	X										
	Operational Support	Applicable in this context through the use of mission planning and debrief										
	Storage	Common to all components										
	Recording and Logging			X								
Specific Policies	Cyber Defence											
	Human-Machine Interface											
	Interfacing with Deployable Assets											
	Tactical Information											
	Test											
	Use of Communications											
	Data Exchange											
Modelling Principles	Component Connections											X
	Component Extensions	X	X		X		X	X	X	X	X	
	Data Driving	X	X	X	X	X	X	X	X	X	X	
Safety and Security	Safety Analysis	Common										
	Security Approach	Common										

Table 1: Policy Applicability to the Routing Interaction View and its Components

The Deployment Guide Ref. [3] describes which policies should be considered at different stages of a deployment.

Within Figure 26: The Routing Interaction View and its Component's Relevance to Mission/Operational Phases it can be seen that, like other IVs, the Routing IV can be applied to various phases of a mission or operational timeline.

It is worth noting that some IV use cases may be specific to a particular mission/operational context and therefore phase. However, in most cases it can be intuitively seen how the use case could be adapted to apply the IV, or one very similar, to additional contexts and phases.



Various planned mission routes created and evaluated by the mission planners to determine which best meets the mission needs

The planned mission route may be validated, checking that details such as time of arrival are still achievable

A minor change to the mission plan may be requested by an operator, resulting in the need to recalculate the planned route.

Route amendments may be automatically applied to take account of a late time of take-off verses a required waypoint time of arrival.

An unexpected event may require a diversion to another airfield, requiring the route to be updated.

Alternative routes may be generated to demonstrate other hypothetical scenarios that could have occurred, including alternative courses of action. For example to support training exercises.

Progression of the mission and unplanned events will require update to the route. For example, to identify an air vehicle, the location of which has just been determined. The developer may have chosen to use data driving and/or extensions for the Tasks and Routes components, allowing specialised behaviour for route creation for different mission activities or needs (e.g. specialised route planning for air to air engagement, rather than basic transit).

Figure 26: The Routing Interaction View and its Component's Relevance to Mission/Operational Phases

Clearly since the IV is applicable to different phases so too are the associated components. In fact, since the components are also applicable to other IVs the ways in which they can be applied exceeds that described here.