



# IMAGE

## Interoperable Management of Aeronautical Generic Executive software

PROPOSAL N° GRD1-2001-40137  
PROJECT CONTRACT N° G4RD-CT-2002-00850

### D 2.3: Generic Architecture requirements

# Table of contents

<b>1</b>	<b>INTRODUCTION .....</b>	<b>3</b>
1.1	IMAGE .....	3
1.2	TASK WP 2.3 .....	3
1.2.1	Objectives.....	3
1.3	RESEARCH METHODOLOGY .....	4
1.4	SIMULATION CLASSIFICATION .....	4
1.4.1	Levels of Interoperability (LISI) Classification.....	4
1.4.2	Domain classification.....	4
1.4.3	Image target.....	5
1.5	TRACEABILITY .....	6
1.5.1	WP 2.1 Requirements .....	6
1.5.2	WP 2.2 Requirements .....	9
<b>2</b>	<b>MIDDLEWARE &amp; INTEROPERABILITY.....</b>	<b>13</b>
2.1	INTEROPERABILITY .....	13
2.2	MIDDLEWARE.....	13
2.3	APPLICABLE REQUIREMENTS FROM WP2.1 TO THE TASK WP23 .....	13
2.3.1	Interoperability .....	13
2.3.2	Heterogeneity.....	14
2.3.3	Flexibility .....	14
2.3.4	Compatibility.....	14
2.3.5	Optimisation of the data exchange.....	14
2.3.6	Administrative support.....	14
2.3.7	Norms and standards .....	14
2.3.8	Cost reduction.....	15
<b>3</b>	<b>PROPOSED ARCHITECTURE.....</b>	<b>16</b>
3.1	SIMULATION COMPONENT. ....	16
3.2	SIMULATION COMMERCIAL COMPONENT .....	16
3.2.1	Wrappers.....	16
3.3	IMAGE MIDDLEWARE.....	16
3.3.1	Communications Service.....	17
3.3.2	Time management Service.....	17
3.3.3	Data Management Service .....	17
3.3.4	Supervision Service .....	17
3.3.5	Analysis & Statistics Service .....	17
3.3.6	Administration Service .....	17
3.3.7	System Configuration Service .....	17
3.3.8	User Management Service .....	17
3.3.9	Security Service.....	17
3.3.10	Repository.....	17
3.4	IMAGE BRIDGE COMPONENT.....	18
3.5	IMAGE UTILITIES COMPONENT.....	18
3.5.1	Supervision Service. ....	18
3.5.2	User Management Service. ....	18
3.5.3	System Configuration Service. ....	18
3.5.4	Analysis & Statistics Service. ....	18
3.6	IMAGE TOOLS COMPONENT.....	18
3.6.1	Testing and controlling tools.....	18
3.6.2	Learning and experimentation support .....	18
3.6.3	Post-processing results and visualisation. ....	18
<b>4</b>	<b>CONCLUSION.....</b>	<b>19</b>
<b>5</b>	<b>DOCUMENTS.....</b>	<b>20</b>
5.1	APPLICABLE DOCUMENTS .....	20
5.2	REFERENCE DOCUMENTS .....	20

# 1 Introduction

## 1.1 IMAGE

IMAGE project aims at implementing a generic and universal environment in order to make complex, distributed and heterogeneous integrated simulators cooperate. IMAGE focuses on such themes as interoperability, modularity, flexibility, cost reduction, etc.

The object of this document (output of WP 2.3) is to precise a wide range of requirements and needs expressed by users of real time applications that the IMAGE API should meet.

## 1.2 Task WP 2.3

This sub-task specifies the generic architecture for the IMAGE system.

CORBA and HLA implementations are the basic mean for inter-process communication.

- CORBA advantage: (100% standard, free, portable).
- OKTAL, IRIT, LBEIN point-out architecture requirements.
- LBEIN designs the distributed architecture fitting with.

### 1.2.1 Objectives

The technological objectives of the project raise themes such as:

- Interoperability
- Heterogeneity
- Generic environment
- Optimisation of the data exchange
- Control
- Administration
- Normalisation

We will mainly explore the axes:

- The system's distributed architecture allowing to optimise the distribution of all the simulation resources
- The simplification of the communications between the systems and the reuse of the existing modules
- The use of more generic platforms fulfilling the users' expressed needs.

Items associated to systems interoperability appearing as crucial for the users:

- To allow cooperation between heterogeneous simulation.
- To handle the constraints of synchronous/asynchronous modules through a simulation supervision module

- To remain coherent: participants must be aware of each other's actions on objects concerning them
- To exchange data among simulation modules with the minimum of manual interaction
- To keep an acceptable level of performance

### 1.3 Research methodology

Each Task Document will refer to current state of the art, by analysing sources or communications found on each subject.

### 1.4 Simulation Classification

- Levels of Interoperability (LISI)
- Domain

#### 1.4.1 Levels of Interoperability (LISI) Classification

The LISI Reference Models translate interoperability levels into requisite capabilities. LISI considers the ability to exchange and share services between systems. The specific capabilities needed to achieve each level are described in four attributes (procedures, applications, infrastructure and data). The LISI Reference Model defines five levels:

- **Level 0 — Isolated Interoperability:** systems have no direct electronic connection. Data exchange occurs via manual keyboard entry or extractable common media format (diskette).
- **Level 1 — Connected Interoperability:** systems are linked electronically. Allows to exchange files, little capability to fuse information together to support decision-making.
- **Level 2 — Functional Interoperability:** distributed systems residing on local networks allowing complex, heterogeneous data sets to pass from system to system.
- **Level 3 — Domain-Based Interoperability:** integrated systems connected via wide area networks (WAN) allowing multiple users to access data.
- **Level 4 — Enterprise-Based Interoperability:** systems can operate using a distributed global information space across multiple domains. Multiple users can access and interact with complex data simultaneously. Data and applications are fully independent and can be distributed.

#### 1.4.2 Domain classification

Aeronautical simulators are used in different domains:

- Electronic Component simulation
- Hybrid simulation
- Standalone simulation

- Coupled simulators
- Large distributed simulation
- Distributed numerical computation
- Web simulation
- Parallel Simulation

#### **1.4.2.1 Electronic Component simulation**

Used to design and check electronic equipments. “Hardware-in-the-loop” simulation.

#### **1.4.2.2 Hybrid Simulation**

Used to test and check unitarily some module in a realistic environment that include real or simulated equipments. “Hardware-in-the-loop” or “Man-in-the-loop” simulation.

#### **1.4.2.3 Standalone simulation**

Used to study or train specific equipment. “Man-in-the-loop” simulation. The goal is to improve the relationship between humans and systems:

- By studying new services, to facilitate the human job
- By training end users.

#### **1.4.2.4 Coupled simulators**

Used to study or train a small team to cooperate in a unique simulation environment. The Department of Defence (DoD) has promoted two military standards: DIS (Distributed Interactive Simulation) and HLA (High Level Architecture). No civil standard exists in this domain.

#### **1.4.2.5 Large distributed simulation**

Simulation where a large number of participants interact.

#### **1.4.2.6 Distributed numerical simulation**

To simulate complex system, usually not in real time, and without a man in the loop.

#### **1.4.2.7 Web simulation**

Uses Web protocol to distribute simulation components on Web.

#### **1.4.2.8 Parallel Simulation**

Simulations running on massive parallel system with a large number of processors.

### **1.4.3 Image target**

#### **Level of Interoperability:**

- Level 4 — Enterprise-Based Interoperability

#### **Domain:**

- Standalone simulation

- Coupled simulators
- Distributed numerical simulation
- Web simulation
- Parallel Simulation

## 1.5 Traceability

This document refines and trace requirements outputs D2.1 [1] and D2.2 [2] , see here below 1.5.1 and 1.5.2.

Requirements ID for Real Time domain, have the following syntax: RTR\_x\_y\_z.

Requirements ID for Numeric Simulation, have the following syntax: NSR\_x\_y\_z.

### 1.5.1 WP 2.1 Requirements

Id	Requirement description	Task
<b>Interoperability</b>		
RTR_1_1	The IMAGE system must allow heterogeneous simulations (in terms of hardware, software, data, network protocols, etc.) to cooperate.	WP23 [3]
RTR_1_2	The IMAGE system must allow distant simulations to cooperate with the illusion of being located in the same place.	WP23 [3]
RTR_1_3	The IMAGE system must allow incompatible simulations to cooperate.	WP23 [3] WP24 [4]
RTR_1_4	The IMAGE system must allow similar simulators to cooperate.	WP23 [3] WP24 [4]
RTR_1_5	The IMAGE system must manage time constraints for modules handling different times by supervision.	WP24 [4]
RTR_1_6	The IMAGE system must remain coherent: all participants should be aware of each other's action on objects concerning them.	WP23 [3] WP24 [4]
<b>Heterogeneity</b>		
RTR_2_1	The IMAGE system must handle simulators or simulation running on different Operating systems, platforms, network protocols, etc.	WP23 [3] WP24 [4]
RTR_2_2	The IMAGE system must handle simulations with pure simulation modules, real equipment modules, man in the loop (through HMI).	WP23 [3]
RTR_2_3	The IMAGE system must be able to handle simply simulation modules from various work domains.	WP23 [3]
RTR_2_4	The IMAGE system must be able to be simply used by "heterogeneous users".	WP23 [3] WP24 [4]
<b>Modularity</b>		
RTR_3_1	The IMAGE system has to be generic with regards to numerous types of applications.	WP24 [4] WP25 [5]
RTR_3_1_1	The IMAGE system must allow the use of existing simulators or simulation applications by encapsulating them.	WP24 [4] WP25 [5]

Id	Requirement description	Task
RTR_3_1_2	The IMAGE system must allow the use of simulation modules designed for IMAGE for other applications.	WP24 [4] WP25 [5]
RTR_3_1_3	The IMAGE system must foster the normalisation and standardisation the simulation components through development rules and tools.	WP25 [5]
<b>Flexibility</b>		
RTR_3_2_1	The IMAGE system must allow any module matching the IMAGE format (by encapsulation or using the IMAGE API) to join or leave a simulation during the run and being aware of all the other participants.	WP24 [4] WP25 [5]
RTR_3_2_2	The IMAGE system must allow any module matching the IMAGE format joining a simulation to be able to detect dynamically the available services and participants.	WP24 [4]
RTR_3_2_3	The IMAGE system must be able to run several simulations one after the other (for Monte-Carlo methods).	WP23 [3] WP24 [4]
RTR_3_2_4	The IMAGE system may be able to run a whole simulation on a single platform.	WP25 [5]
RTR_3_2_5	The IMAGE system must allow the modules to run without being part of a simulation.	WP23 [3] WP24 [4] WP25 [5]
RTR_3_2_6	The IMAGE system must allow modules to run sequentially and exploit each other's data.	WP23 [3] WP24 [4] WP25 [5]
RTR_3_2_7	The IMAGE system must allow a simulation to stop/restart.	WP24 [4]
<b>Compatibility</b>		
RTR_3_3_1	The IMAGE system must ensure the compatibility of many software used (e.g. Data Base Management Systems, etc.).	WP23 [3] WP25 [5]
RTR_3_3_2	The IMAGE system must manage the different user profiles of a distributed simulation.	WP24 [4]
<b>Optimisation of the data exchange</b>		
RTR_4_1	The IMAGE system must allow the exchange of data between simulators without common definitions of their objects.	WP23 [3] WP25 [5]
RTR_4_2	The IMAGE system must handle important data flows.	WP23 [3] WP24 [4]
RTR_4_3	The IMAGE system must provide a generic format that all the other users can understand/use but being able to get specific enough to satisfy the user.	WP23 [3] WP25 [5]
RTR_4_4	The IMAGE system must be able to exchange data both on LAN or WAN or through the internet.	WP23 [3] WP24 [4]
RTR_4_5	The IMAGE system supervision module should be able to cast data only to the module interested (multicast)	WP23 [3] WP24 [4]
RTR_4_7	The IMAGE system should ensure the availability of the treated data.	WP25 [5] WP23 [3]
<b>Security</b>		
RTR_5_1_1	The IMAGE system must allow a safe run of multi-sites simulations in terms of access to the simulation (unauthorized hosts can not join a simulation).	WP24 [4]

Id	Requirement description	Task
RTR_5_1_2	The IMAGE system must allow safely the running of multi-sites simulations in terms of access to the simulation information (unauthorized hosts can not access some information).	WP24 [4]
RTR_5_1_3	The IMAGE system must allow safely the running of multi-sites simulations in terms of data exchange (unauthorized hosts can not spy on the exchanged data).	WP24 [4]
RTR_5_1_4	The IMAGE system must handle the interaction between secured/unsecured networks.	WP24 [4]
RTR_5_1_5	The IMAGE system must make sure that the exchanged information are not changed, altered or modified without authorization.	WP24 [4]
RTR_5_1_6	The IMAGE system must provide multi-user distributed simulation data propriety protection capabilities.	WP24 [4]
<b>Safety</b>		
RTR_5_2_1	The IMAGE system increases the safety of the software and hardware modules by allowing them to be tested extensively from the beginning of the development.	WP25 [5]
RTR_5_2_2	The IMAGE system must provide testing and controlling tools to test and control modules and whole simulation behaviour.	WP24 [4]
RTR_5_2_3	The interaction between real world and simulation must be particularly safe.	WP24 [4]
<b>Robustness</b>		
RTR_5_3_1	The IMAGE system must detect and handle errors.	WP24 [4]
RTR_5_3_2	The IMAGE system must go on running even if a simulation module goes off. Being able to restart a module that went off.	WP24 [4]
RTR_5_3_3	The IMAGE system must be able to follow part or all the simulation, to access data without perturbing the simulation.	WP24 [4]
RTR_5_3_4	The IMAGE system must be able to detect if a connecting participant has a sufficient power and so will not degrade the performance of the networked simulation.	WP24 [4]
<b>Learning and experimentation support</b>		
RTR_6_1_1	The IMAGE system must integrate more efficiently in the simulation the learning supports associated to the different modules.	WP24 [4]
RTR_6_1_2	The IMAGE system must reinforce the guidelines for the experimentation supports of the simulation.	WP24 [4] WP25 [5]
RTR_6_1_3	The IMAGE system must improve the experimental framework (duration of the experimentation execution, number of replications needed, combination of input factors to be used for specific simulation run, etc.).	WP24 [4]
RTR_6_1_4	The IMAGE system must provide simulation analysis tools.	WP24 [4]
RTR_6_1_5	The IMAGE system must provide learning and training support.	WP24 [4]
<b>Administrative support</b>		
RTR_6_2_1	The IMAGE system must allow simulation services management.	WP23 [3] WP24 [4]

Id	Requirement description	Task
RTR_6_2_2	The IMAGE system must provide off-line simulation resource management.	WP24 [4]
RTR_6_2_3	The IMAGE system must be easily maintainable.	WP24 [4] WP25 [5]
RTR_6_2_4	The IMAGE system must store relevant data and provide tools to replay (the user can interact) or redisplay (the user is passive) a simulation.	WP24 [4]
RTR_6_2_5	The IMAGE system should provide debugging facilities.	WP24 [4]
RTR_6_2_6	The IMAGE system must enable data management in distributed simulation environment.	WP24 [4]
RTR_6_2_7	The IMAGE system must provide implementation constraints management capabilities.	WP24 [4] WP25 [5]
RTR_6_2_8	The IMAGE system must allow distributed and heterogeneous simulations interfaces management.	WP24 [4]
RTR_6_2_9	The IMAGE system must provide consolidated statistical facilities.	WP24 [4]
RTR_6_2_10	The IMAGE system must ensure distributed cross-referencing capabilities.	WP24 [4]
<b>Configurability</b>		
RTR_6_3_1	The IMAGE system must be able to configure the simulation with regard to the user needs as simply as possible.	WP24 [4]
RTR_6_3_2	The IMAGE system must be able to record/load a configuration.	WP24 [4]
RTR_6_3_3	The IMAGE system must build a hosting structure for a distributed simulation from a library of systemic and reusable components.	WP25 [5]
RTR_6_3_4	The IMAGE system must allow the user to build simulation clusters.	WP24 [4] WP25 [5]
<b>Norms and standards</b>		
RTR_7_1	The IMAGE system must be able to make the system respect a specific norm or standard or to demonstrate that it respects this standard or norm.	WP2
<b>Cost reduction</b>		
RTR_8_1	The IMAGE system will allow easy interfacing of components from different simulation systems.	WP25 [5]
RTR_8_2	The IMAGE system must allow the replacing of real equipment by computerized elements.	WP25 [5]
RTR_8_3	The IMAGE system will encourage the replacing expensive dedicated platforms by several cheaper generic platforms (typically PC).	WP23 [3]
RTR_8_4	The IMAGE system will allow the sharing of computers and network resources on a distributed system.	WP23 [3]
RTR_8_5	The IMAGE system must keep an acceptable level of quality (performance, speed, precision of the data, ...).	WP23 [3]

### 1.5.2 WP 2.2 Requirements

Id	Requirement	Task
<b>Interoperability</b>		

<b>Id</b>	<b>Requirement</b>	<b>Task</b>
NSR_01_1	Allowing heterogeneous simulations (in terms of hardware, software, data, network protocols, etc.) to co-operate.	WP23 [3]
NSR_01_2	Allowing distant simulations to cooperate with the illusion of being located in the same place.	WP23 [3]
NSR_01_3	Allowing incompatible simulations to co-operate.	WP23 [3]
NSR_01_4	Synchronisation	WP23 [3] WP24 [4]
NSR_01_5	Consistency	WP23 [3] WP24 [4]
NSR_01_6	Performing multi-physic simulations	WP23 [3] WP25 [5]
<b>Heterogeneity</b>		
NSR_02_1	Simulation running on different Operating systems, platforms, network protocols, etc.	WP23 [3]
NSR_02_2	Handling simulations with pure simulation modules, real equipment modules and man in the loop (through HCI).	WP23 [3]
<b>Cost Reduction</b>		
NSR_03_1	Interfacing easily components from different simulation systems (development time gain).	WP25 [5]
NSR_03_3	Choosing cheap image generating systems using graphics cards on PC technologies and monitors projection instead of complex immersing system.	WP23 [3]
NSR_03_4	In a more general way, replacing expensive dedicated platforms by several cheaper generic platforms (typically PC).	WP23 [3]
NSR_03_5	Sharing the computers and network resources on a distributed system.	WP23 [3]
NSR_03_7	Keeping an acceptable level of performance.	WP23 [3] WP25 [5]
<b>Modularity</b>		
NSR_04_1	Using existing simulators or simulation applications by encapsulating them.	WP25 [5]
NSR_04_2	Using simulation modules for other applications	WP25 [5]
NSR_04_4	Normalising and standardising the simulation components through development rules and tools.	WP2
<b>Flexibility</b>		
NSR_05_1	Allow encapsulation for the C and FORTRAN languages	WP25 [5]
NSR_05_2	Allowing any module matching the IMAGE format (by encapsulation or using the IMAGE API) to join or leave a simulation during the run (handled by the supervision part) and being aware of all the other participants.	WP24 [4]
NSR_05_3	Being able to detect dynamically the available services.	WP24 [4]
NSR_05_4	Being able to run several simulations one after the other (for Monte-Carlo methods).	WP23 [3] WP24 [4]
NSR_05_6	Being able to run a whole simulation on a single platform.	WP25 [5]
NSR_05_7	Allowing the modules to run without being part of a simulation.	WP23 [3] WP24 [4]
NSR_05_8	Allowing modules to run sequentially and exploit each	WP24 [4]

Id	Requirement	Task
	other's data.	WP25 [5]
<b>Optimisation of the data exchange</b>		
NSR_06_1	Exchanging data between simulators without common definitions of their objects.	WP23 [3] WP25 [5]
NSR_06_2	Handling important data flows.	WP23 [3]
NSR_06_3	Providing a generic format that all the other users can understand/use but being able to get specific enough to satisfy the user: preserving meaning and being mutually interpretable.	WP23 [3] WP25 [5]
NSR_06_4	Being able to exchange data both on LAN or WAN.	WP23 [3]
NSR_06_5	Exchanging data (mesh, geometry) among simulation packages with the minimum manual interaction.	WP25 [5]
NSR_06_6	Availability of treated data.	WP23 [3]
NSR_06_7	Sharing data via Internet	WP23 [3]
<b>Security</b>		
NSR_07_1	Running safely multi-sites simulations in terms of access to the simulation (unauthorised hosts can not join a simulation).	WP24 [4]
NSR_07_2	Running safely multi-sites simulations in terms of access to the simulation information (unauthorised hosts can not access some information).	WP24 [4]
NSR_07_3	Running safely multi-sites simulations in terms of data exchange (unauthorised hosts can not spy on the exchanged data).	WP24 [4]
NSR_07_4	Handling the interaction between secured/unsecured networks	WP24 [4]
NSR_07_5	Making sure that the exchanged information are not changed, altered or modified without authorisation.	WP24 [4]
NSR_07_6	Provide multi-user distributed simulation data propriety protection capabilities.	WP24 [4]
<b>Safety</b>		
NSR_08_1	Increasing the safety of the software and hardware modules by testing them extensively from the beginning of the development.	WP24 [4] WP25 [5]
NSR_08_2	Providing testing and controlling tools to test and control modules and whole simulation behaviour.	WP24 [4]
<b>Robustness</b>		
NSR_09_1	Going on running even if a simulation module goes off (handled by the supervision part). Being able to try to restart a module that went off.	WP24 [4]
NSR_09_2	Detecting and handling errors (handled by the supervision part).	WP24 [4]
NSR_09_3	Being able to follow part or all the simulation, to access data without perturbing the simulation.	WP24 [4]
NSR_09_4	Being able to detect if a connecting participant has a sufficient power and so will not degrade the performance of the networked simulation.	WP24 [4]
<b>Learning and experimentation support</b>		

Id	Requirement	Task
NSR_10_1	Increasing existing lack of difficulty of learning packages associated to simulations and simulator environment. There is a need to implement user-friendly interfaces and online help modules for the several modules.	WP24 [4] WP25 [5]
NSR_10_2	Enhancing guidance for simulation experimentation support. There is a need to provide online help documentation with examples of the modules and simulations and perhaps a small wizard to start the simulations.	WP24 [4] WP25 [5]
NSR_10_3	Allow for better experimental design support, e.g. duration of experimental run, number of replications needed, combination of input factors to be used for specific simulation run, etc.	WP24 [4] WP25 [5]
NSR_10_4	Improve the simulation environment, output design and analysis facilities.	WP24 [4] WP25 [5]
<b>Administrative support</b>		
NSR_11_1	Enable data management in distributed simulation environment.	WP23 [3] WP24 [4]
NSR_11_2	Provide off-line simulation resource management	WP24 [4]
NSR_11_3	Provide implementation constrains management capabilities.	WP24 [4] WP25 [5]
NSR_11_4	Allow for simulation services management.	WP23 [3] WP24 [4]
NSR_11_5	Allow for distributed and heterogeneous simulations interfaces management.	WP25 [5] WP24 [4]
NSR_11_6	Provide consolidated statistical facilities.	WP24 [4]
NSR_11_7	Ensure distributed cross-referencing capabilities.	WP24 [4]
<b>Configurability</b>		
NSR_12_1	Ensure that the simulation environment can be configured to correspond to the users needs.	WP24 [4]
NSR_12_2	Build simulation environment from library of distributed simulation systems to extract ready-built components of simulation.	WP24 [4] WP25 [5]
<b>Compatibility</b>		
NSR_13_1	Provide facilities for compatibility of various simulation packages used (e.g. DBMS', etc.). The data environment should ensure that any other tool would use data generated by any tool.	WP23 [3] WP24 [4] WP25 [5]
NSR_13_2	Enable compatibility of user logic for co-operating (distributed-heterogeneous) simulations	WP24 [4] WP25 [5]

## 2 Middleware & Interoperability

Middleware is the “glue” software. It must provide security, identification, authentication and synchronisation services and ensure the correct data interchange.

### 2.1 Interoperability

Interoperability is the ability of a set of models and simulation to provide services to and accept services from other models and simulations.

- **Technical interoperability.**
- **Data interoperability.**

### 2.2 Middleware

Middleware (systems software residing between the applications and the lower-level hardware and software infrastructure) coordinates how applications are connected and interoperate.

**Host infrastructure middleware** encapsulates and enhances native OS communication and concurrency mechanisms to create reusable network programming components, See as examples JVM (JAVA Virtual Machine) and ACE (ADAPTIVE Communication Environment).

**Distribution middleware** defines higher-level distributed programming models whose reusable APIs and components automate and extend the native OS network programming capabilities encapsulated by host infrastructure middleware. At the heart of distribution middleware are request brokers, such as CORBA, an open standard that allows objects to interoperate across networks.

**Common middleware services** augment distribution middleware by defining higher-level domain-independent services that allow application developers to concentrate on programming business logic.

**Domain-specific middleware** services are tailored to the requirements of particular domains. Domain-specific middleware services are targeted at vertical markets.

### 2.3 Applicable Requirements from WP2.1 to the task WP23

#### 2.3.1 Interoperability

Id	Requirement description
RTR_1_1	The IMAGE system must allow heterogeneous simulations (in terms of hardware, software, data, network protocols, etc.) to cooperate.
RTR_1_2	The IMAGE system must allow distant simulations to cooperate with the illusion of being located in the same place.
RTR_1_3	The IMAGE system must allow incompatible simulations to cooperate.
RTR_1_4	The IMAGE system must allow similar simulators to cooperate.
RTR_1_6	The IMAGE system must remain coherent: all participants should be aware of each other's action on objects concerning them.

### 2.3.2 Heterogeneity

Id	Requirement description
	<b>Heterogeneity</b>
RTR_2_1	The IMAGE system must handle simulators or simulation running on different Operating systems, platforms, network protocols, etc.
RTR_2_2	The IMAGE system must handle simulations with pure simulation modules, real equipment modules, man in the loop (through HMI).
RTR_2_3	The IMAGE system must be able to handle simply simulation modules From various work domains.
RTR_2_4	The IMAGE system must be able to be simply used by “heterogeneous users”.

### 2.3.3 Flexibility

Id	Requirement description
RTR_3_2_3	The IMAGE system must be able to run several
RTR_3_2_5	The IMAGE system must allow the modules to run without being part of a simulation.
RTR_3_2_6	The IMAGE system must allow modules to run sequentially and exploit each other's data.

### 2.3.4 Compatibility

Id	Requirement description
RTR_3_3_1	The IMAGE system must ensure the compatibility of the various software used (e.g. Data Base Management Systems, etc.).

### 2.3.5 Optimisation of the data exchange

Id	Requirement description
RTR_4_1	The IMAGE system must allow the exchange of data between simulators without common definitions of their objects.
RTR_4_2	The IMAGE system must handle important data flows.
RTR_4_3	The IMAGE system must provide a generic format that all the other users can understand/use but being able to get specific enough to satisfy the user.
RTR_4_4	The IMAGE system must be able to exchange data both on LAN or WAN or through the internet.
RTR_4_5	The IMAGE system supervision module should be able to cast data only to the module interested (multicast)

### 2.3.6 Administrative support

Id	Requirement description
RTR_6_2_1	The IMAGE system must allow simulation services management.

### 2.3.7 Norms and standards

Id	Requirement description
RTR_7_1	The IMAGE system must be able to make the system respect a specific norm or standard or to demonstrate that it respects this standard or norm.

### 2.3.8 Cost reduction

Id	Requirement description
RTR_8_3	The IMAGE system will encourage the replacing expensive dedicated platforms by several cheaper generic platforms (typically PC).
RTR_8_4	The IMAGE system will allow the sharing of computers and network resources on a distributed system.
RTR_8_5	The IMAGE system must keep an acceptable level of quality (performance, speed, precision of the data, ...).

### 3 Proposed Architecture

The IMAGE architecture is composed of the following elements:

- IMAGE middleware
- IMAGE Simulation Components
- IMAGE Tools
- Third party or commercial simulation components and tools
- IMAGE Bridge to external services
- Repository

#### 3.1 Simulation Component.

Simulation Component is developed in D2.5 [5] .

Heterogeneity	NSR_02_3
Interoperability	NSR_01_1, NSR_01_3, NSR_01_6
Modularity	NSR_04_1, NSR_04_2

#### 3.2 Simulation Commercial Component

General Commercial simulator that we wish to integrated into the system.

Interoperability	NSR_01_6
Heterogeneity	NSR_02_3
Cost Reduction	NSR_03_1, NSR_03_2
Modularity	NSR_04_3, NSR_04_4

##### 3.2.1 Wrappers

Wrappers mask the difference in implementation of legacy systems.

Interoperability	NSR_01_3
Modularity	NSR_04_1
Heterogeneity	NSR_02_2
Flexibility	NSR_05_1, NSR_05_2, NSR_05_8

#### 3.3 IMAGE Middleware

The IMAGE middleware follows a layering architecture:

- At the bottom layer, communication between heterogeneous operating systems is encapsulated.
- The second layer of the middleware handles data and time management.
- The third layer (Supervision, Administration and Security components) provides abstractions for communications and interoperability exchange.

Interoperability	NSR_01_1, NSR_01_2, NSR_01_3
Heterogeneity	NSR_02_1
Cost reduction.	NSR_03_1, NSR_03_5

### 3.3.1 Communications Service

Interoperability	NSR_01_5
Heterogeneity	NSR_02_1
Modularity	NSR_04_3, NSR_04_4
Optimisation of the data exchange	NSR_06_1, NSR_06_2, NSR_06_4, NSR_06_5

### 3.3.2 Time management Service

Interoperability	NSR_01_2, NSR_01_3, NSR_01_4
Cost Reduction	NSR_03_7

### 3.3.3 Data Management Service

Interoperability	NSR_01_5
Robustness	NSR_09_3
Modularity	NSR_04_4
Compatibility	NSR_13_2

### 3.3.4 Supervision Service

The supervision service is developed in D2.4 [4] .

Flexibility	NSR_05_3
Robustness	NSR_09_3, NSR_09_4
Safety	NSR_08_2
Administrative support	NSR_11_1-NSR_11_5

### 3.3.5 Analysis & Statistics Service

Administrative support	NSR_11_6, NSR_11_7
------------------------	--------------------

### 3.3.6 Administration Service

The administration service is developed in D2.4 [4] .

Administrative support	NSR_11_4, NSR_11_7
------------------------	--------------------

### 3.3.7 System Configuration Service

Flexibility	NSR_05_3
Administrative support	NSR_11_2, NSR_11_3
Configurability	NSR_12_1

### 3.3.8 User Management Service

Heterogeneity	NSR_02_2
Compatibility	NSR_13_2
Security	NSR_07_6
Administrative Support	NSR_11_2
Configurability	NSR_12_1

### 3.3.9 Security Service

The security service is developed in D2.4 [4] .

Security	NSR_07_1, NSR_07_6
----------	--------------------

### 3.3.10 Repository

This service allows storing and retrieving session and simulation data.

### 3.4 IMAGE Bridge Component

An IMAGE Bridge Component is defined to handle services from external systems.

Heterogeneity	NSR_02_1
Interoperability	NSR_01_3, NSR_01_5
Cost Reduction	NSR_03_1, NSR_03_4
Modularity	NSR_04_3
Flexibility	NSR_05_1
Compatibility	NSR_13_2
Optimisation of the data exchange	NSR_06_4, NSR_06_5

### 3.5 IMAGE Utilities Component.

Administrative support	NSR_11_4, NSR_11_7
------------------------	--------------------

#### 3.5.1 Supervision Service.

Flexibility	NSR_05_3
Robustness	NSR_09_3, NSR_09_4
Safety	NSR_08_2
Administrative support	NSR_11_1-NSR_11_5

#### 3.5.2 User Management Service.

Heterogeneity	NSR_02_2
Compatibility	NSR_13_2
Security	NSR_07_6
Administrative Support	NSR_11_2
Configurability	NSR_12_1

#### 3.5.3 System Configuration Service.

Flexibility	NSR_05_3
Administrative support	NSR_11_2, NSR_11_3
Comfigurability	NSR_12_1

#### 3.5.4 Analysis & Statistics Service.

Administrative support	NSR_11_6, NSR_11_7
------------------------	--------------------

### 3.6 IMAGE Tools Component.

#### 3.6.1 Testing and controlling tools

Safety	NSR_08_1, NSR_08_2
Robustness	NSR_09_3

#### 3.6.2 Learning and experimentation support

Learning and Experimentation Support	NSR_10_1, NSR_10_4
--------------------------------------	--------------------

#### 3.6.3 Post-processing results and visualisation.

Learning and experimentation support	NSR_10_4
--------------------------------------	----------

## 4 Conclusion

This document presents a potential architecture for the IMAGE system.

The set of elements used in the architecture are:

- Simulation Components.
- Commercial Simulation Component. Integrated using Wrappers.
- IMAGE Middleware. Offering five services.
- IMAGE Utilities.
- IMAGE Tools.
- IMAGE Bridge.

Each of them was described and referenced to the identifier of requirements met.

## 5 Documents

### 5.1 Applicable documents

- [1] WP 2.1
- [2] WP 2.2
- [3] WP 2.3 This document.
- [4] WP 2.4
- [5] WP 2.5
- [6] [DA1] contract: G4RD-CT-2002-00850
- [7] [DA2] proposal IMAGE: GRD1-2001-40317
- [8] [DA3] annexe 1: technical annexe IMAGE-ANNEXE-TECH-12-02.doc en date du 25/12/02
- [9] [DA4] annexe 2: general conditions
- [10] [DA5] IMAGE Consortium Agreement v1.0 (to be delivered)
- [11] [DA6] IMAGE Quality Plan D 2.6.2 v 1.0 (to be delivered)
- [12] [DA7] All the deliverable documents of the IMAGE Project (to be delivered)
- [13] [DA8] OKTAL Quality Manual v 8.0

### 5.2 Reference documents

- [14] **P. T. Barham, R. E. Barker, J. L. Metzger, J. R. Damiano**  
HLA as A Distributed Virtual Worlds Architecture  
International Conference on Virtual Worlds and Simulation, 1999
- [15] **P.T. Barham, R.E. Barker, R.E. Beckwith**  
Implementing a simple HLA Federate: Lessons Learned from a DIS perspective  
Simulation Interoperability Workshop, Spring 1998
- [16] **A. Buss, L. Jackson**  
Distributed Simulation Modeling: a Comparison of HLA, CORBA and RMI  
Winter Simulation Conference, 1998
- [17] **M. Rheinlander**  
The Model Driven Architecture and its application to M&S.  
Calytrix Technologies  
Available at [www.calytrix.com](http://www.calytrix.com)
- [18] **J. S. Dahmann, R. M. Fujimoto, R. M. Weatherly**  
The Dod High Level Architecture: An Update  
Winter Simulation Conference, 1998
- [19] **A. Radeski, S. Parr**  
Towards a Simulation Component Model for HLA  
Simulation Interoperability Workshop, Fall 2002
- [20] **M. R. Reid**

An Evaluation of the High Level Architecture (HLA) as a Framework for NASA Modeling and Simulation

25th NASA Software Engineering Workshop, 2000

**[21] S. Reilly, H. Williams**

IDL4HLA: Implementing CORBA IDL Middleware for HLA

Simulation Interoperability Workshop, Spring 2001

**[22] K. S. Sandeep K. S. Gupta, A. Jayasumana, P. K. Srimani**

CORBA Based HLA/RTI Design Approach

Summer Computer Simulation Conference (SCSC '97), July 1997

**[23] R. Smith**

High Level Architecture

[www.simulationfirst.com/ein5255](http://www.simulationfirst.com/ein5255)

**[24] S. Strassburger, T. Schulze, U. Klein, J.O. Henriksen**

Internet-based Simulation Using off-the-shelf simulation tools and HLA

Winter Simulation Conference, 1998

**[25] S. Strassburger**

High Level Architecture in Civilian Applications Domains

PhD Thesis, Otto-von-Guericke-Universitat Magdeburg, 2001

**[26] A. Tolk**

Avoiding another Green Elephant – A Proposal for the Next Generation HLA based on the Model Driven Architecture

Simulation Interoperability Workshop, Fall 2002

**[27] S. Vinoski**

CORBA: Integrating Diverse Applications Within Distributed Heterogeneous Environments

IEEE Communications Magazine, Vol. 35, No. 2, 1997

**[28] Object Management Group**

The Common Object Request Broker: Architecture and Specification.

Available at <http://www.omg.org>

**[29] Defense Modeling and Simulation Office**

High Level Architecture Object Model Template v1.3

**[30] Defense Modeling and Simulation Office**

High Level Architecture Rules v1.3

**[31] G. C. Fox, W. Furmanski, H. T. Ozdemir**

Object Web (Java/CORBA) based RTI to support Metacomputing M&S

International Test and Evaluation Association (ITEA) Workshop on High Performance Computing for Test and Evaluation, 1998

**[32] R. E. Schantz, D. C. Schmidt**

Middleware for Distributed Systems, Evolving the Common Structure for Network-centric Applications

Encyclopedia of Software Eng., Wiley & Sons, New York, 2001;

Available at <http://www.cs.wustl.edu/~schmidt/PDF/middleware-chapter.pdf>

**[33] A. Tolk**

Heterogeneous Synergy – A vision for the next generation of Warfighter IT systems

2000 Fall Simulation Interoperability Workshop

USA, September 2000

**[34] J. Dahmann**

HLA and Beyond: Interoperability Challenges  
1999 Fall Simulation Interoperability Workshop

**[35] J. Bettin**

Model-Driven Architecture Implementation & Metrics  
2002

Available at <http://www.softmetaware.com/whitepapers.html>

**[36] M. D'Hondt, T. D'Hondt**

The Tyranny of the Dominant Model Decomposition  
2002

Available at [http://www.softmetaware.com/oopsla2002/mda\\_workshop.html](http://www.softmetaware.com/oopsla2002/mda_workshop.html)

**[37] G. C. Fox, W. Furmanski, H. T. Ozdemir**

Object Web (Java/CORBA) based RTI to support Metacomputing M&S  
International Test and Evaluation Association (ITEA) Workshop on High Performance Computing for  
Test and Evaluation, 1998

**[38] R. Herzog, J. Mulder**

HLA on top of CORBA Common Object Services  
Euro Simulation Interoperability Workshop 2001

**[39] [NRC] National Research Council**

Realizing the potential of C4I. Fundamental challenges  
1999

Available at <http://www.nap.edu/html/C4I>

**[40] J. Russell**

IKE 2 – Implementing the Stateful Distributed Object Paradigm  
5th IEEE International Symposium on Object-Oriented Real Time Distributed Computing

**[41] R. Soley, OMG Staff Strategy Group**

Model Driven Architecture  
Object Management Group White Paper Draft 3.2  
2000

Available at <http://www.omg.org/mda>

**[42] Test and Training Enabling Architecture (TENA)**

Available at [http://www.tecnet.jcte.jcs.mil/TENA\\_Baseline/volume10/tasktoc.htm](http://www.tecnet.jcte.jcs.mil/TENA_Baseline/volume10/tasktoc.htm)

**[43] Sean Baker**

Web Services and CORBA

Available at <http://link.springer.de/link/service/series/0558/papers/2519/25190618.pdf>

**[44] Generic Toolbox for Interoperable Systems**

Available at [www.gti6.com](http://www.gti6.com)

**[45] C. Perez, T. Priol**

Grid Computing with Off-the-Shelves Middleware Technologies  
ERCIM News No.45 - April 2001

**[46] Joint Technical Architecture**

<http://www-jta.itsi.disa.mil>