SimWare-Kernel

Real Time Communication System for Simulation

(Paper by: Bruno Calvo, Ignacio Seisdedos)

http://www.nexteleng.com/

- Introduction
- Concepts
- Standards
- Implementation
- Interoperability scenarios
- Applications

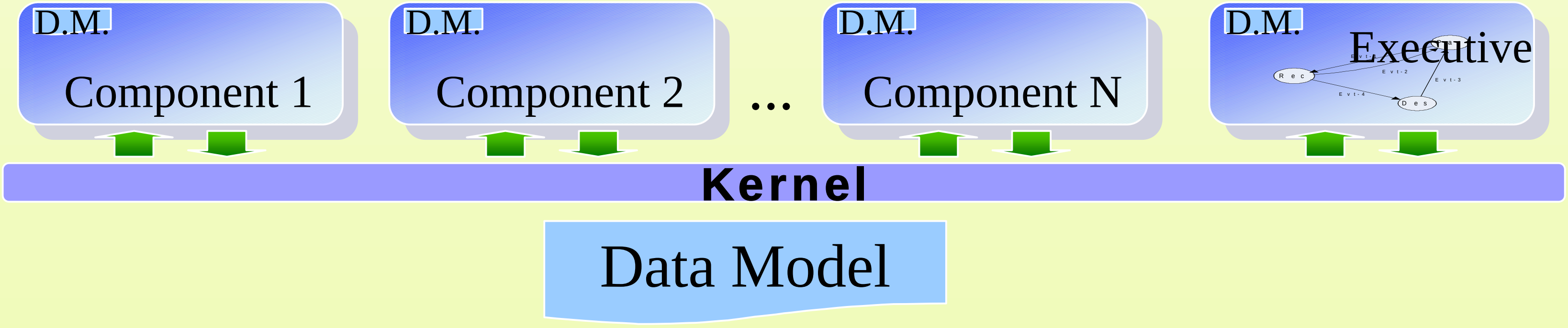
Introduction

SimWare is a simulation framework based on:

- Distributed simulation architecture (eSim)
- Mission support architecture (MSOE)

Concepts:

- **Publish / subscribe** paradigm
- Use of a **data model**



D.M. Component 1

D.M. Component 2

...

D.M. Component N

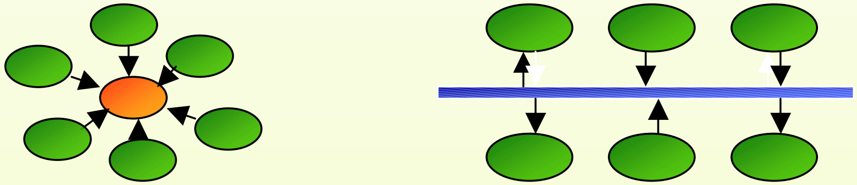
D.M. Executive

Kernel

Data Model

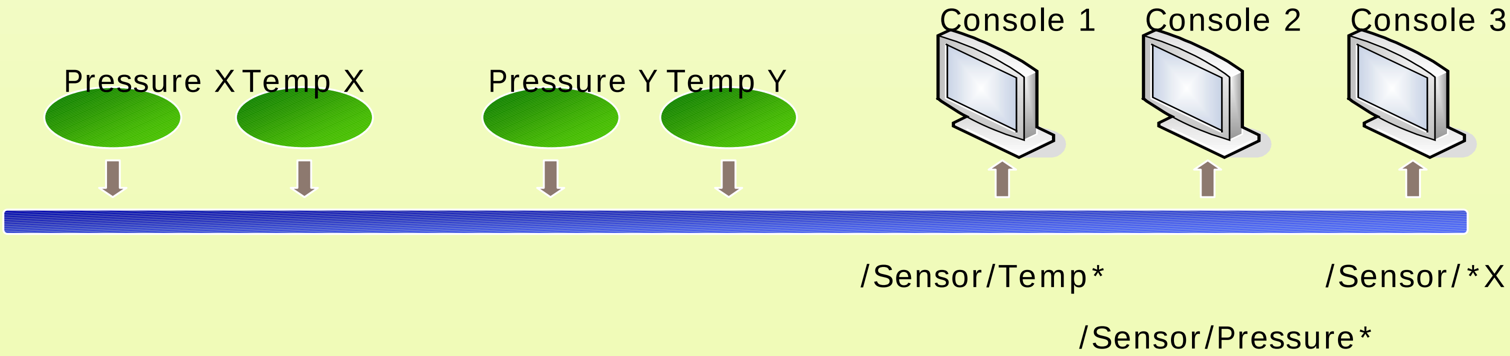
Concepts
Publish-Subscribe

Client-Server vs Publish-Subscribe



How is data identified?

- Type: data types defined in data model (temp, pressure,…)
- Topic: data sent by an entity (tempX, pressureX)

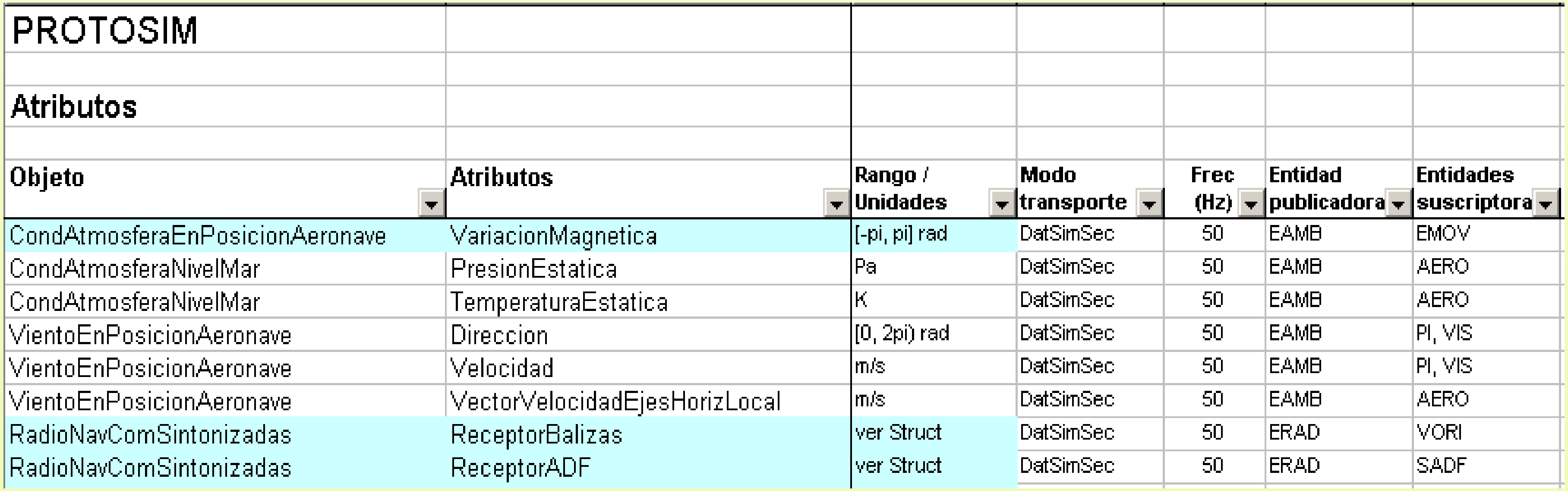


Concepts

Data Model

Objective: gather in a single document all data types used by the system

- Objects and interactions
- Atributes and parameters
- Transportation
- Frecuency
- Entities



PROTOSIM

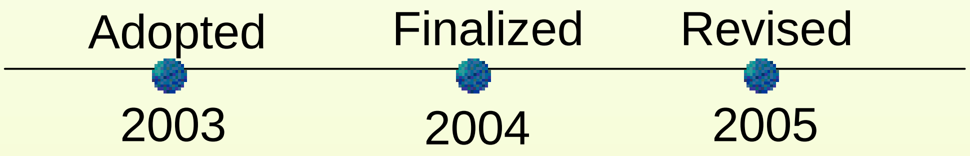
Atributos

Objeto	Atributos	Rango / Unidades	Modo transporte	Frec (Hz)	Entidad publicadora	Entidades suscriptora
CondAtmosferaEnPosicionAeronave	VariacionMagnetica	[-pi, pi] rad	DatSimSec	50	EAMB	EMOV
CondAtmosferaNivelMar	PresionEstatica	Pa	DatSimSec	50	EAMB	AERO
CondAtmosferaNivelMar	TemperaturaEstatica	K	DatSimSec	50	EAMB	AERO
VientoEnPosicionAeronave	Direccion	[0, 2pi) rad	DatSimSec	50	EAMB	PI, VIS
VientoEnPosicionAeronave	Velocidad	m/s	DatSimSec	50	EAMB	PI, VIS
VientoEnPosicionAeronave	VectorVelocidadEjesHorizLocal	m/s	DatSimSec	50	EAMB	AERO
RadioNavComSintonizadas	ReceptorBalizas	ver Struct	DatSimSec	50	ERAD	VORI
RadioNavComSintonizadas	ReceptorADF	ver Struct	DatSimSec	50	ERAD	SADF

Standards

DDS

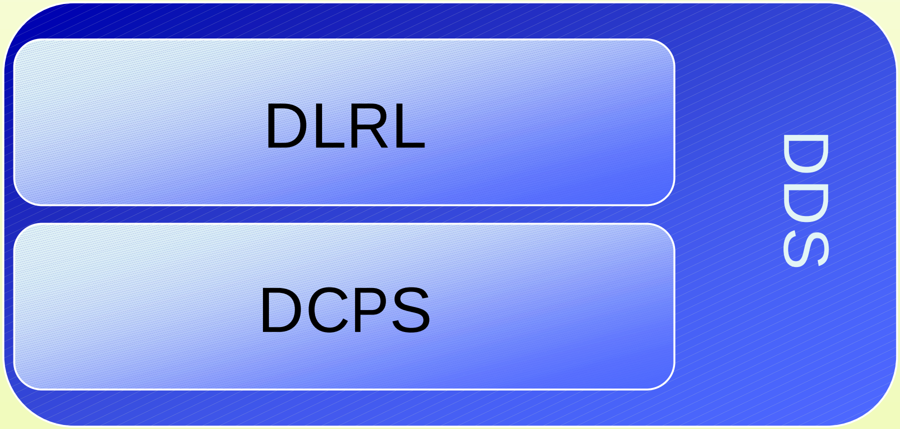
DDS: Data Distribution Service for Real-Time Systems



Adopted
2003

Finalized
2004

Revised
2005



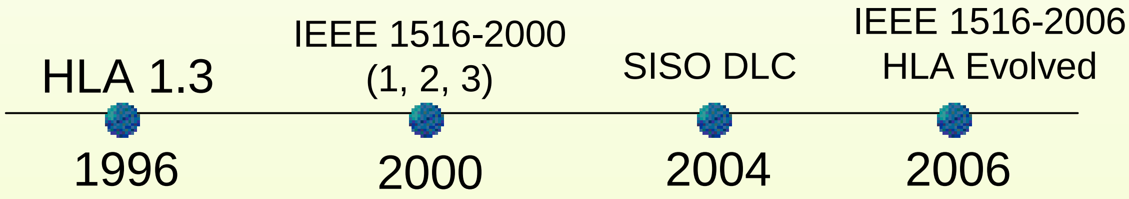
DLRL

DCPS

DDS

Standards

HLA

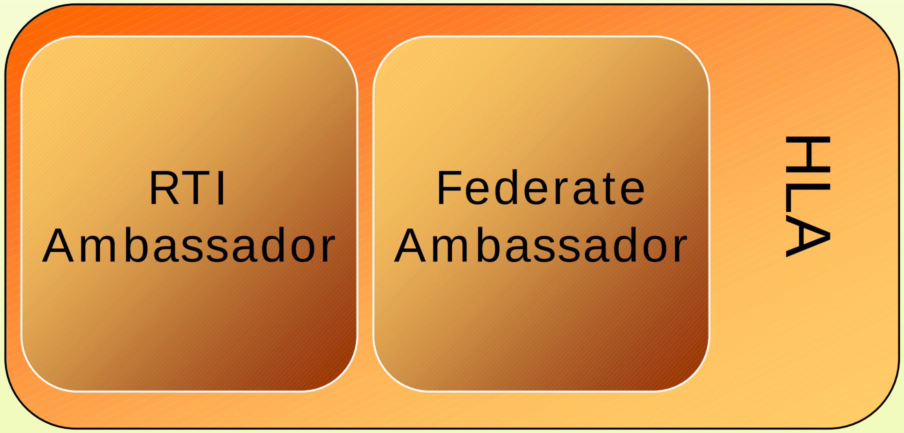


HLA 1.3
1996

IEEE 1516-2000
(1, 2, 3)
2000

SISO DLC
2004

IEEE 1516-2006
HLA Evolved
2006



RTI Ambassador

Federate Ambassador

HLA

Standards

Comparison DDS-HLA

<table>
<tr><th>HLA</th><th>DDS</th></tr>
<tr><td colspan="2"><em>Similarities</em></td></tr>
<tr><td colspan="2">Publish/subscribe data communication</td></tr>
<tr><td colspan="2">User specified data types described in external file (FOM - IDL)</td></tr>
<tr><td colspan="2">Multiple federations (DDS domains) can co-exist</td></tr>
<tr><td colspan="2">Standard defined API</td></tr>
<tr><td colspan="2">Content based filtering</td></tr>
<tr><td colspan="2">Suitable for simulation applications</td></tr>
<tr><td colspan="2">Tools for data specification, application development, debugging, monitoring</td></tr>
</table>

Standards

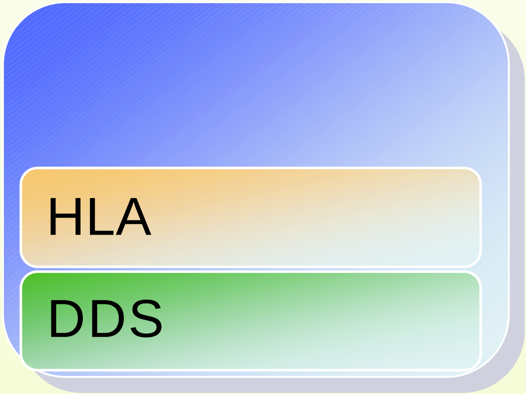
Comparison DDS-HLA

HLA	DDS
Differences	
Designed for simulation	Designed for real-time
Hierarchical object models	DDS supports hierarchical object graphs (DLRL)
Persistent objects	DDS has “durability” on topics
No implied QoS semantics	Rich QoS semantics
Logical time management	No implied time management semantics
No standard wire protocol	Open standard RTPS wire protocol

Implementation

Motivation for a DDS-based HLA

The similarities between DDS and HLA make DDS fit perfectly as the low-level layer of an enhanced version of HLA/RTI.



Benefits of this implementation

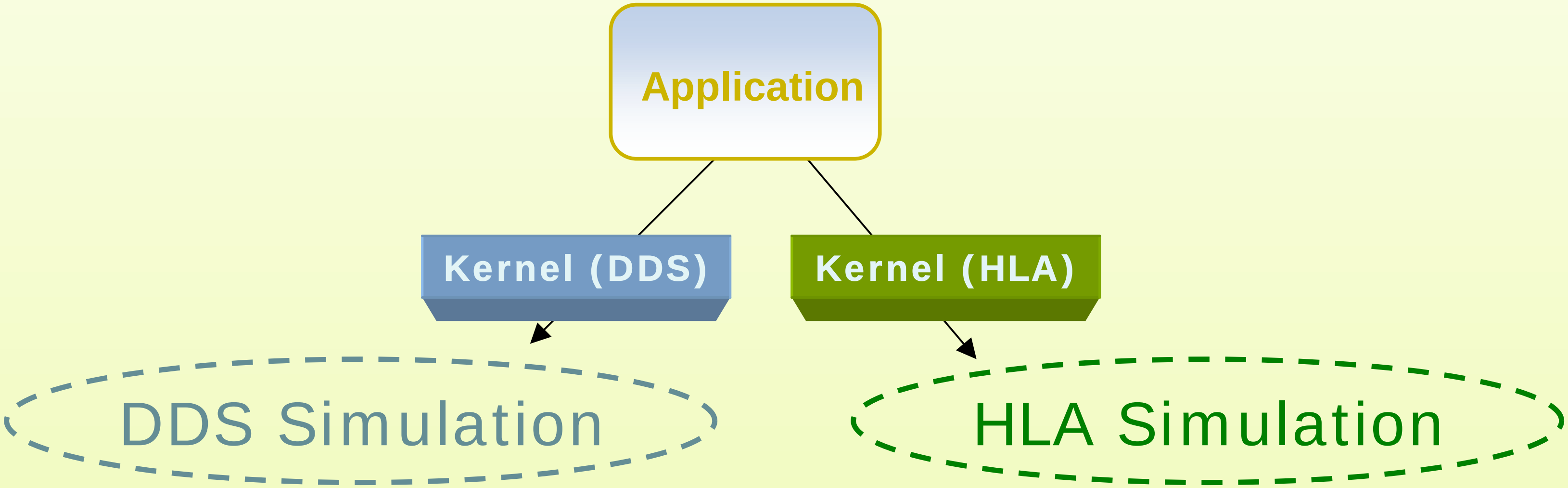
- Enhanced performance, which meets new interoperability requirements (optimized for RealTime, available on different RT OS, ...)
- Direct connection DDS/HLA (simple DDS data from sensors to HLA federations)
- Present to the user new DDS QoS parameters.
- Easy HLA migration, since Runtime Infrastructure shall meet the latest HLA Dynamic link compatible API.
- Standarize the wire-protocol (RTPS).

DDS is also available for real-time simulations, when there is no need of HLA semantics.

Implementation

SimWare-Kernel

Middleware based on publish-subscribe paradigm, offering in a single API, access to either DDS domains or HLA federations.

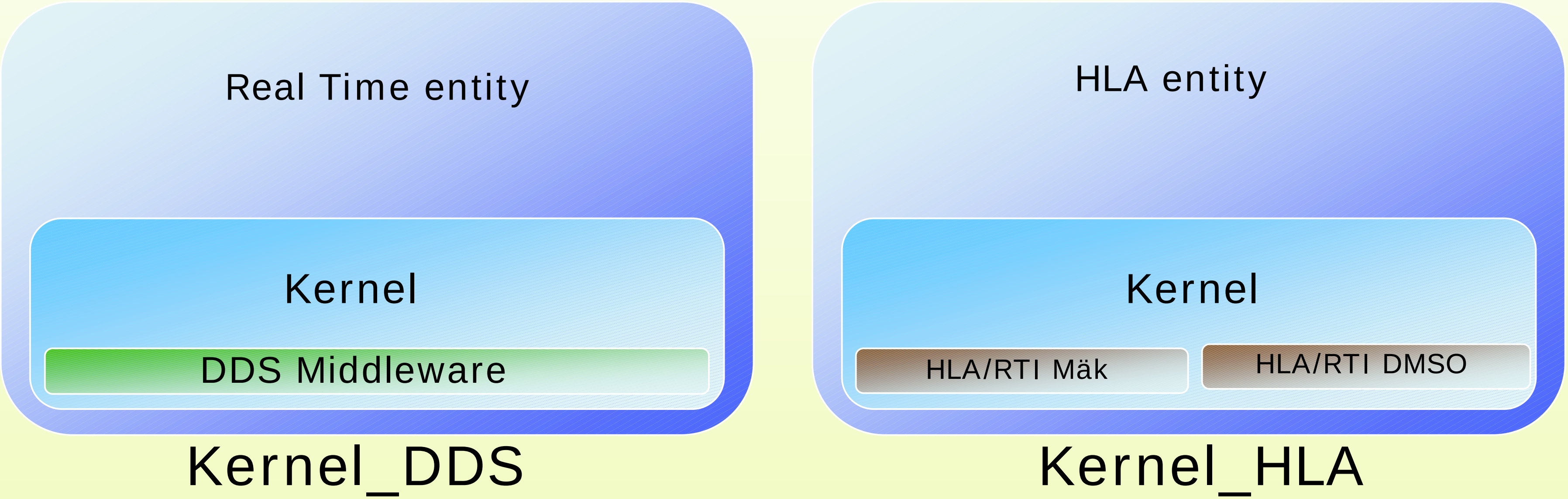


- Used inside simulator
- QoS parameters

- Used to join simulators
- Interoperability with current applications

Implementation

Kernel DDS vs HLA



Advantages

- Allows an easy distribution of components between HLA and DDS simulations
- Unifies HLA applications of many Runtime Infrastructure vendors through the use of the same interface.
- Allows an uniform use of data.
- Allows dynamic (from XML file) generation of data types.

Implementation
Kernel API

Kernel has a user-level API based on:

- Simplicity: focusing on just sending/receiving data.
- Having an advanced API to offer HLA services:
  - Declaration management (data discovery)
  - Instance based publications, instance discovery
  - Sinchronization points
  - Save / restore
  - Attribute based ownership
  - Durability

Implementation
Services

Basic services

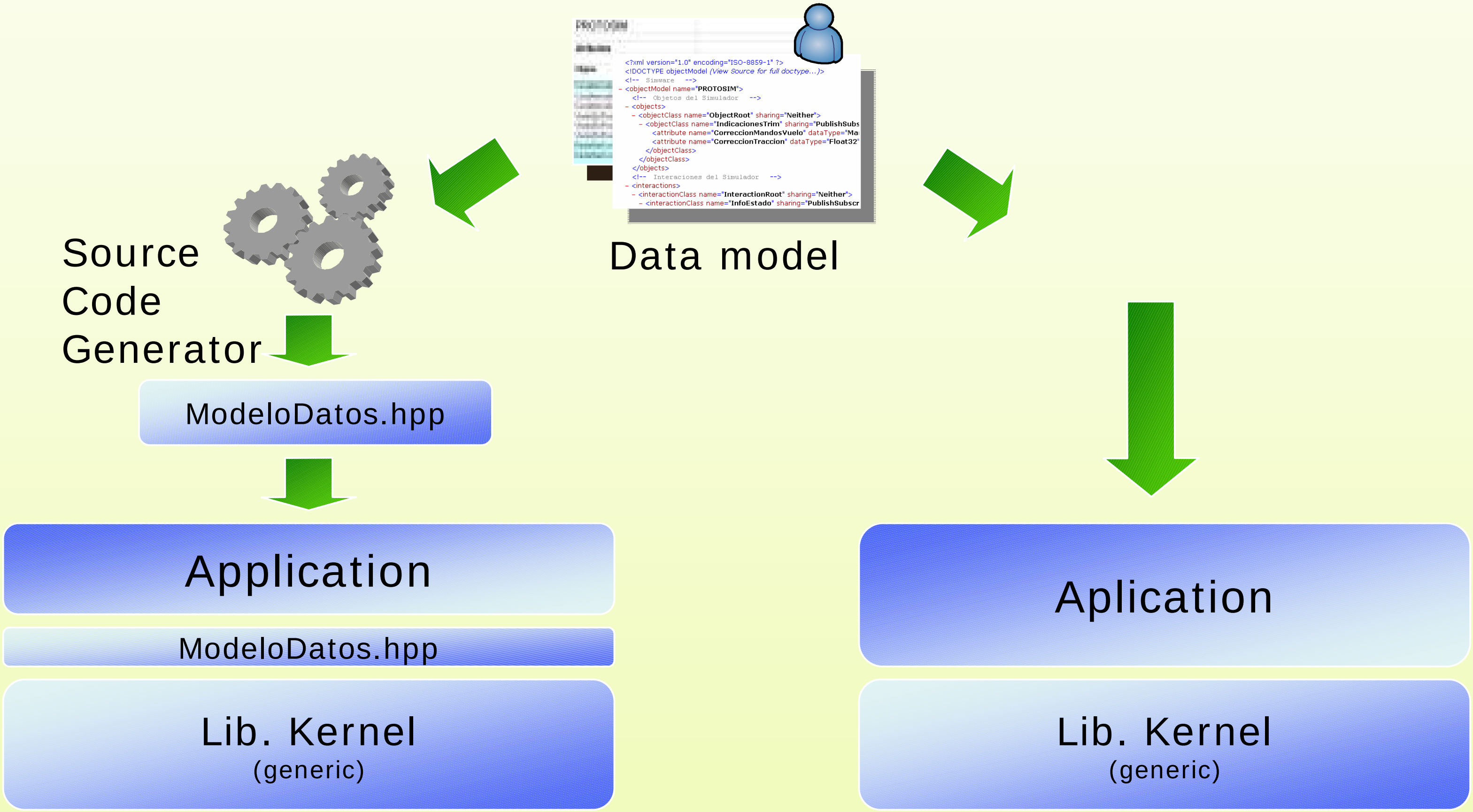
- QoS Transportation types: defined in the data model
- QoS Frecuency: defined in the data model
- Persistence: volatile (DDS) + transient (Kernel)
- Ownership: HLA property management
- Domains/Federations: many independent simulations at the same time
- Liveliness: declaration management, discovery, error management, etc.
- Dead Reckoning

Future

- Content filters
- Variable frequency
- Simulation discovery

Implementation

Data Model use options



```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!DOCTYPE objectModel (View Source for full doctype...)>
<!--  Simware   -->
- <objectModel name="PROTOSIM">
  <!--  Objetos del Simulador   -->
- <objects>
  - <objectClass name="ObjectRoot" sharing="Neither">
    - <objectClass name="IndicacionesTrim" sharing="PublishSubs
        <attribute name="CorreccionMandosVuelo" dataType="Ma
        <attribute name="CorreccionTraccion" dataType="Float32"
      </objectClass>
    </objectClass>
  </objects>
  <!--  Interaciones del Simulador   -->
- <interactions>
  - <interactionClass name="InteractionRoot" sharing="Neither">
    - <interactionClass name="InfoEstado" sharing="PublishSubscr
```

Data model

Source Code Generator

ModeloDatos.hpp

Application

ModeloDatos.hpp

Lib. Kernel
(generic)

Aplication

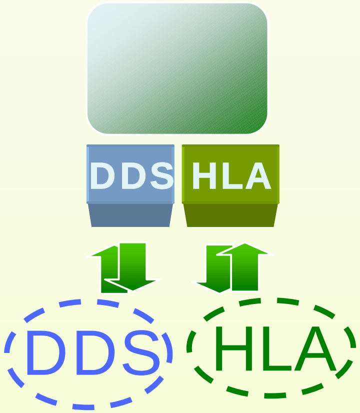
Lib. Kernel
(generic)

Specific application
(simulation modules, etc)

Generic application
(gateway, recorder, etc.)

Implementation

Gateway



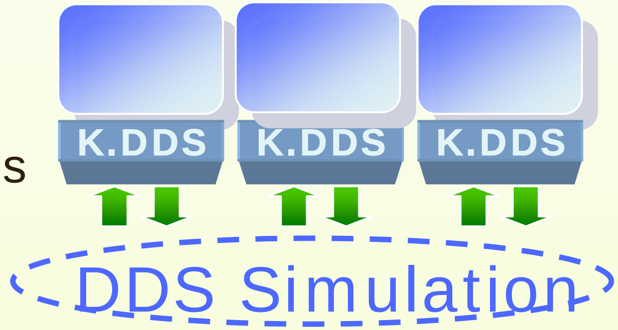
Application using Kernel middleware
designed mainly to join DDS and HLA applications.

- Joins DDS domains and HLA federations:
  - DDS-DDS
  - HLA-HLA
  - DDS-HLA
- Manages mapping of data between data models
- Filters, in order to allow only relevant data to be sent from one simulation to the other
- Joins not only user data, but also HLA services:
  - Ownership.
  - Durability
  - Entity discovery

Interoperability Scenarios

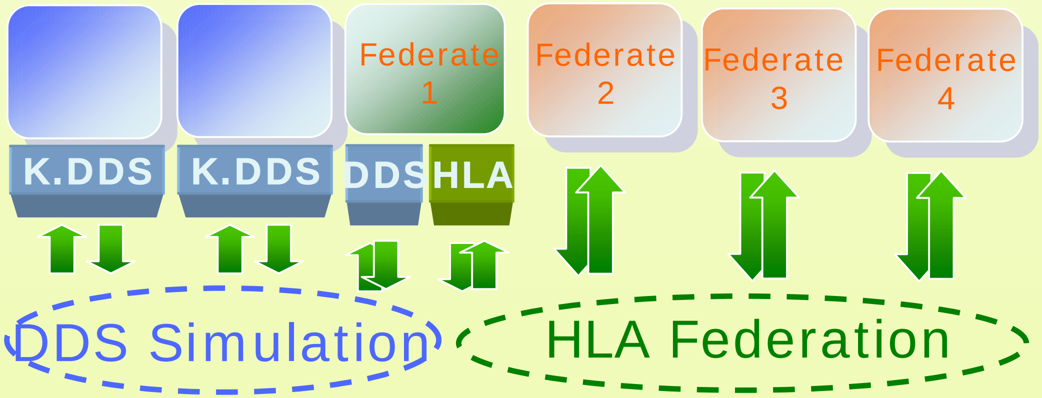
DDS simulation

- Performance improved simulations
- QoS parameters



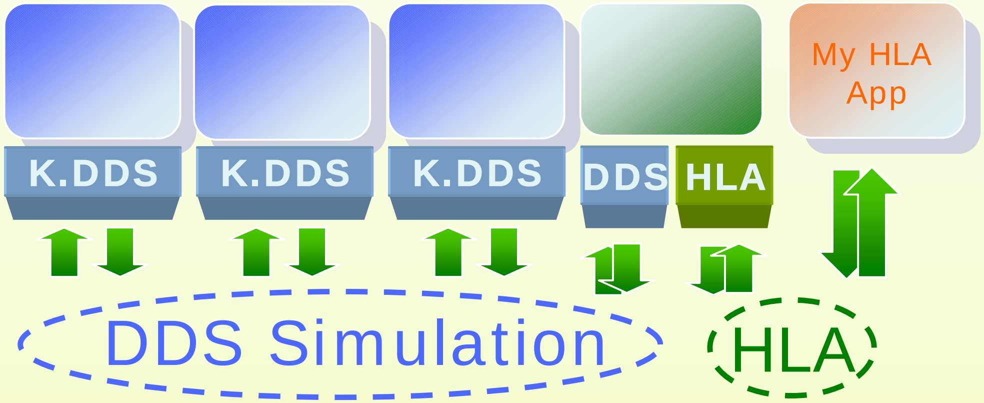
DDS simulation, offering filtered data to HLA

- High performance in DDS simulation, allows real-time
- Offer only relevant data to HLA federates, at lower frequency

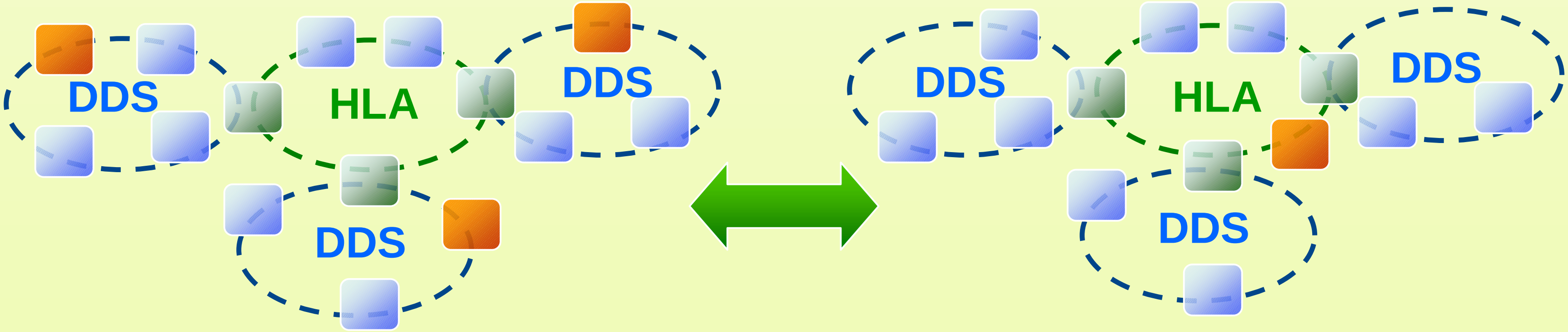


Interoperability
Scenarios

Reusing HLA applications into DDS simulations



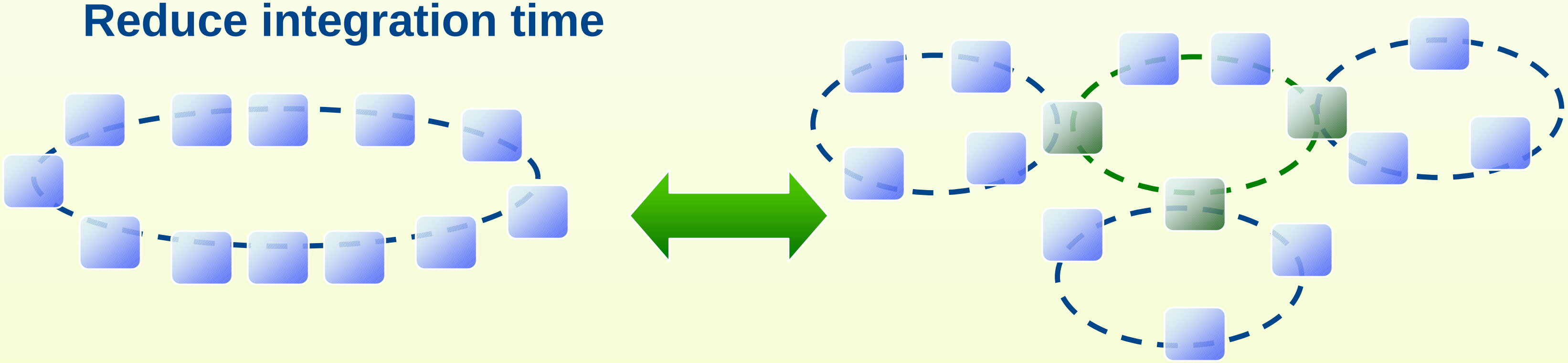
Distribution of applications. Reuse of components.



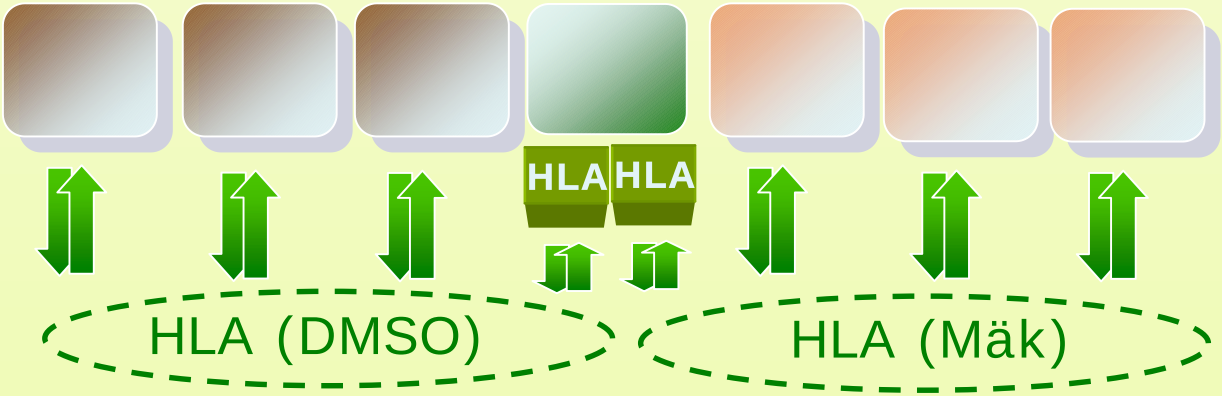
Interoperability

Scenarios

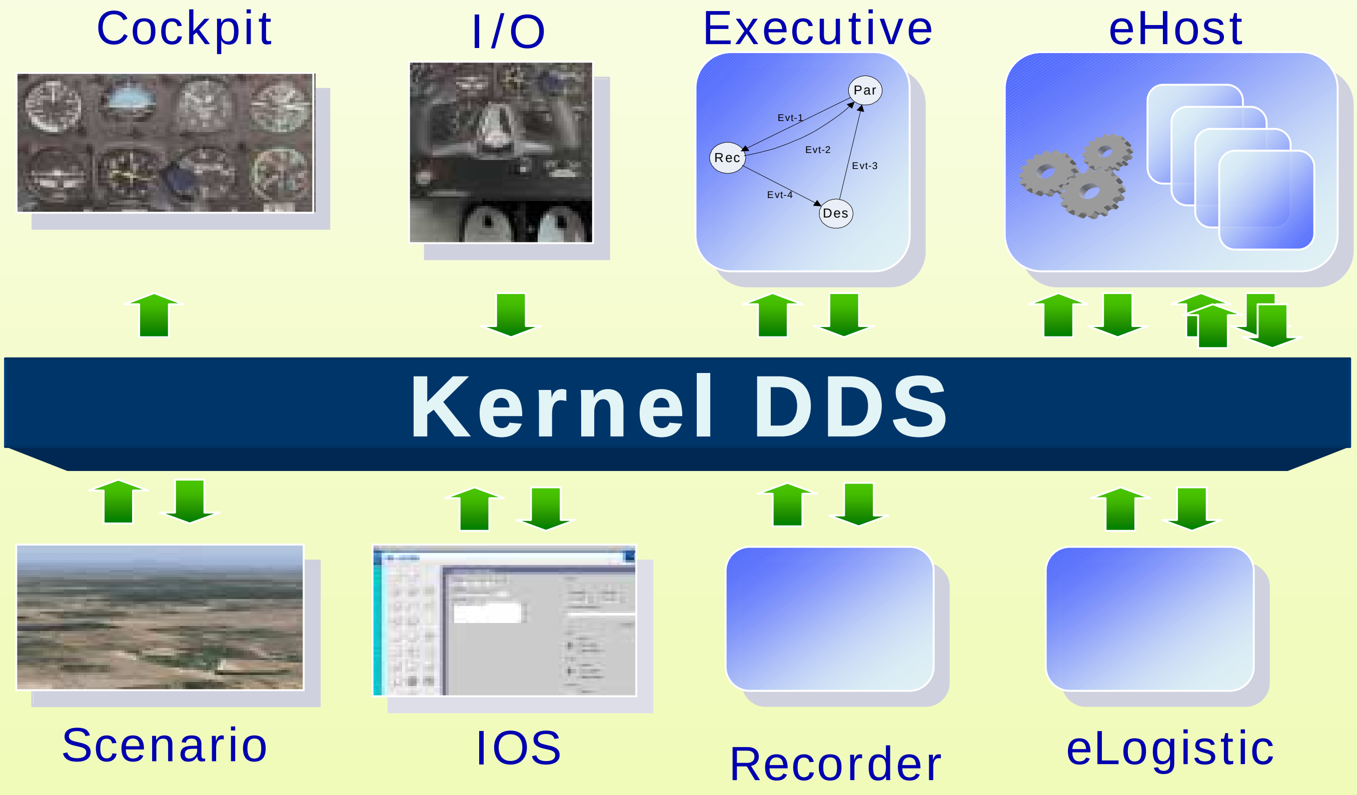
Reduce integration time



Join many HLA federations with different HLA/RTIs (DMSO, Mäk, ...)

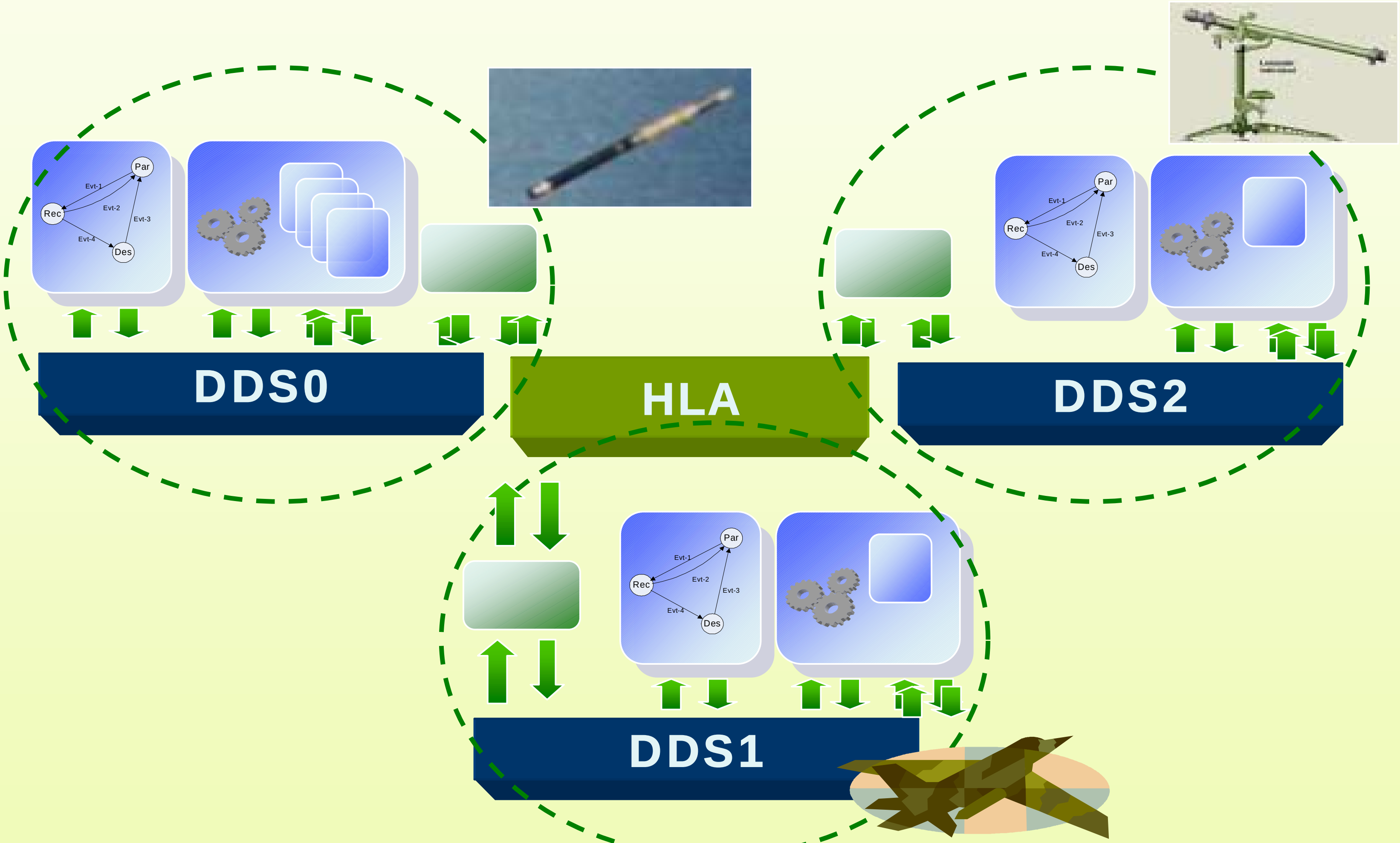


Examples
Flight Simulator



Examples

Missile Mistral Simulator



DDS0

HLA

DDS2

DDS1

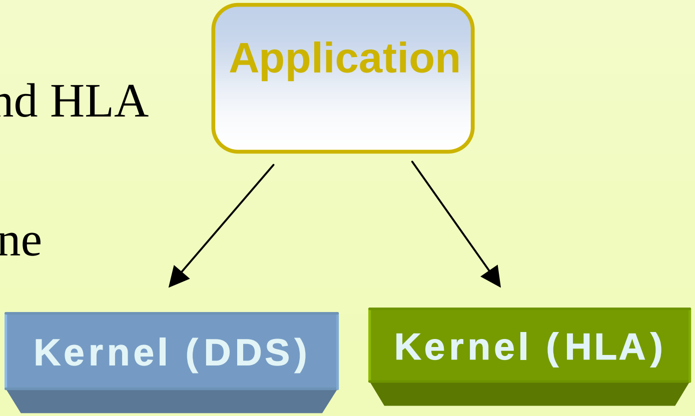
Summary

A DDS-based HLA could bring important benefits:

- Lower Latency
- Better communication determinism and RT behavior
- Enhanced QoS as defined by the DDS standard
- Wire interoperability
- Communication with non-simulation platforms

The SimWare Kernel is platform for distributed simulation that provides the best of both worlds:

- Offers a publish-subscribe that abstracts (joins) DDS and HLA
- Provides a powerful, unified single data model
- Enables many deployment scenarios which may combine existing HLA implementation platforms


Application
Kernel (DDS)
Kernel (HLA)