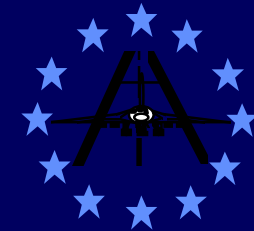


AVENUE



ATM Validation *EN*vironment for *U*se towards *EATMS*

TRANSPORT RESEARCH PROGRAMME
DG7 - TRANSPORT/AIR

TASK N° 4.1.3/24A

OMG transportation
16 November 1999

Contents

- Avenue Overview
 - Technical Overview
 - CCM
 - OASIS (Open Architecture for Simulation System) and PLUG (Presentation layer Universal Generator)
 - System Development Approach
 - Logical Model
 - Component Interface Definition (CIDs) and APIs
 - Component interconnection (first instance of the system).
-

Contents

- ➔ Avenue Overview
 - Technical Overview
 - CCM
 - OASIS and PLUG
 - System Development Approach
 - Logical Model
 - Component Interface Definition (CIDs) and APIs
 - Component interconnection (first instance of the system).
-

AVENUE

- European Union Project (4th Framework)
 - Construct a European ATM Validation Platform
 - European agreement on IDL interfaces
 - 1st Platform built using OASIS
 - Industry Driven
 - Validate operational concepts
 - Harmonize system interfaces
 - Provide practical system solutions
-

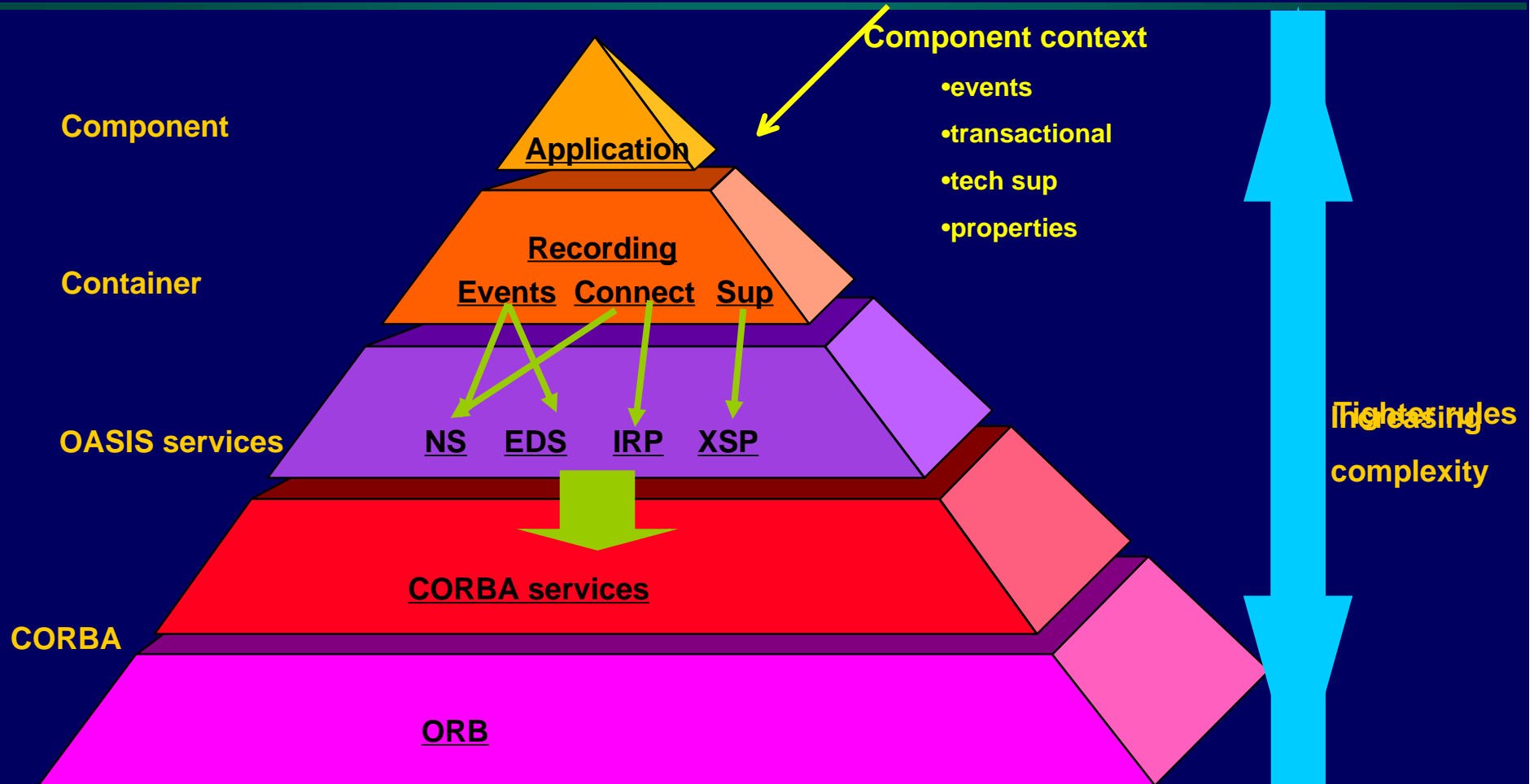
AVENUE Partners

- AENA Aeropuertos Espanoles y Navegacion Aerea (E)
 - Aérospatiale Société Nationale Industrielle (F)
 - Airsys ATM SA (F), ATM UK Ltd (GB)
 - Alcatel ISR Informatique de Systèmes et réseaux (F)
 - ALENIA Alenia Difesa, un'Azienda Finmeccanica SpA (I)
 - DERA Defense Evaluation and Research Agency (GB)
 - DFS Deutsche Flugsicherung GmbH (D)
 - EEC EUROCONTROL Experimental Centre
 - INDRA DTD, S.A (E)
 - ISDEFE Ingeniera de Sistemas para la Defensa de Espana (E)
 - NLR Nationaal Lucht- en Ruimtevaartlaboratorium (NL)
 - SOFREAVIA Société Française d'Etudes et de Réalisations d'Equipements
Aéronautiques (F)
-

Main design areas

- Infrastructure (middleware, recording, configuration etc.)
 - CORBA Component Model
 - Developed by EEC in line with standards : OASIS
 - Support by AIRSYS tools : PLUG
 - Application architecture
 - IDL specification
 - Independent of physical architecture
-

AVENUE component model



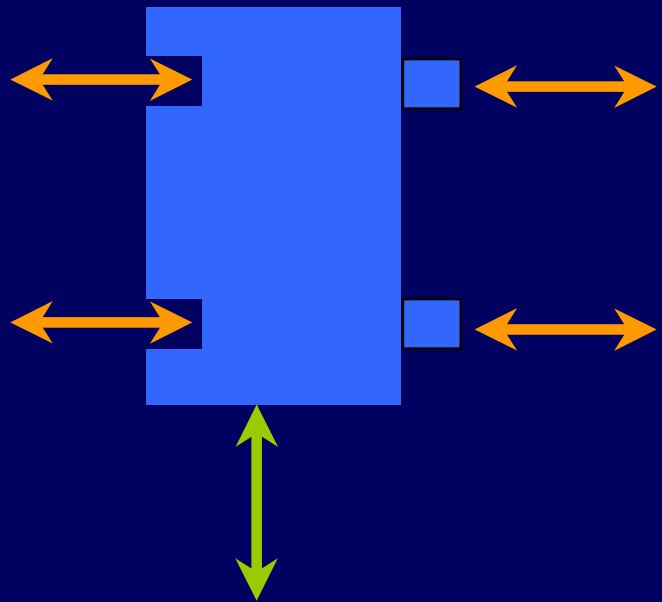
Contents

- ✓ Avenue Overview
 - ➔ Technical Overview
 - ➔ CCM
 - OASIS and PLUG
 - System Development Approach
 - Logical Model
 - Component Interface Definition (CIDs)
 - Component interconnection (first instance of the system).
-

CORBA Components

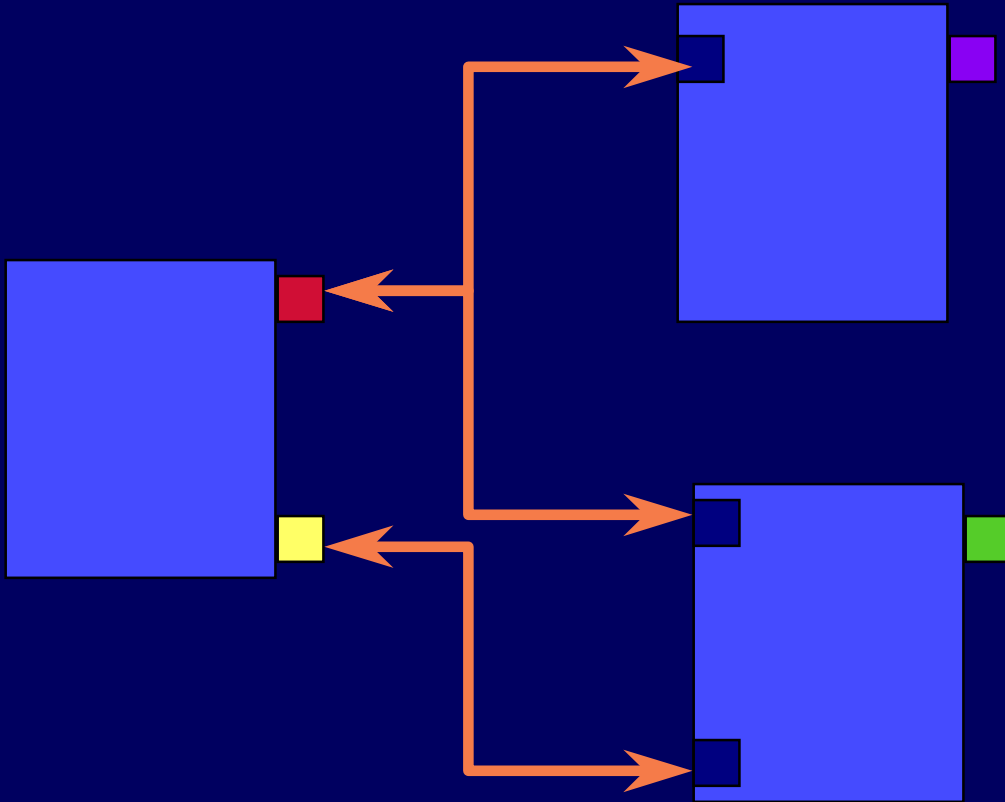
- Industry driven
 - based on experience of building complex distributed systems
 - follows other emerging models (EJB, JavaBeans)
 - Major content
 - Component extensions to IDL
 - Container / Component Model
-

Component IDL (IDL3)



- Extension to IDL
- «packaging» language
- Define a component in terms of...
 - Offered functionality
interfaces **provided**
events **emitted**
 - Dependencies
distant interfaces **required**
events **consumed**
 - Configuration attributes

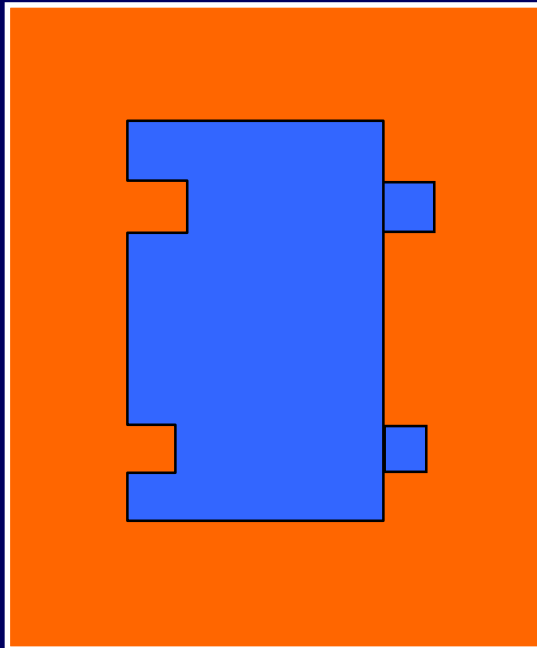
Component IDL



IDL3 provides

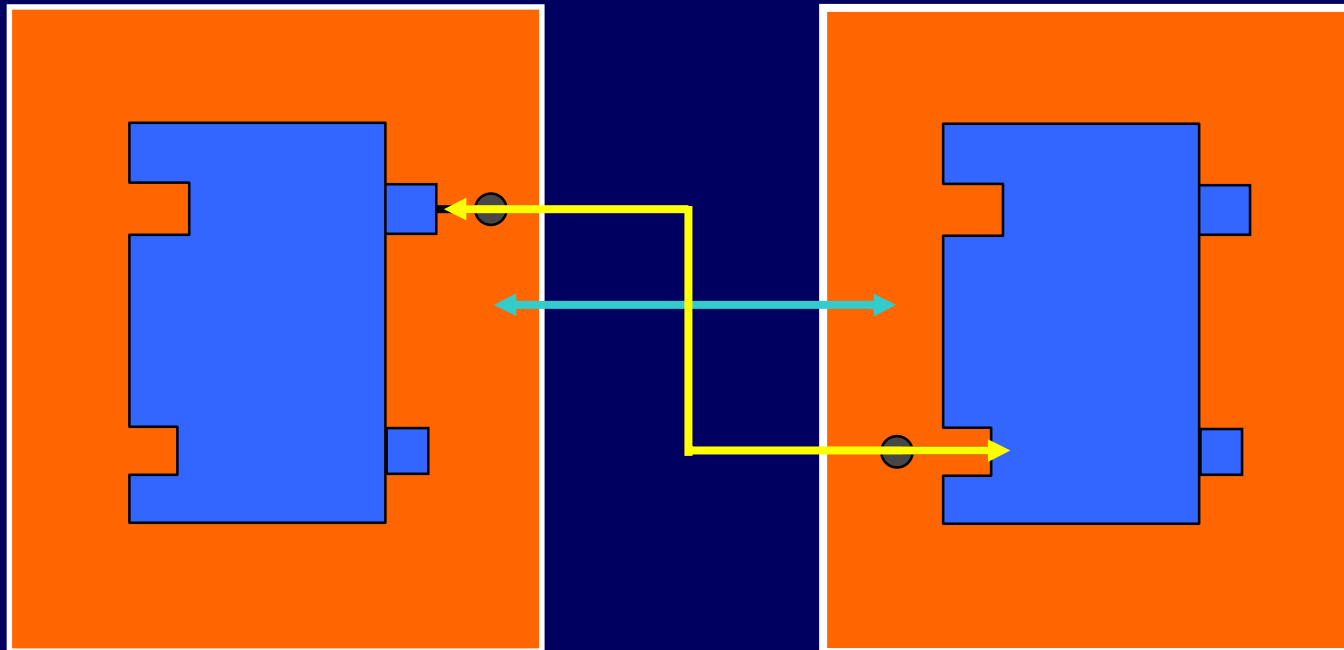
- architecture clarity
- natural connectivity
- dependency visibility

Components need containers



- container handles,
 - component interconnection
 - event issues
 - transactions
 - config / packaging issues
 - other functions
- Component is isolated from the underlying architecture.
- this is a standard & formal way of creating wrappers.

Component/Container interaction

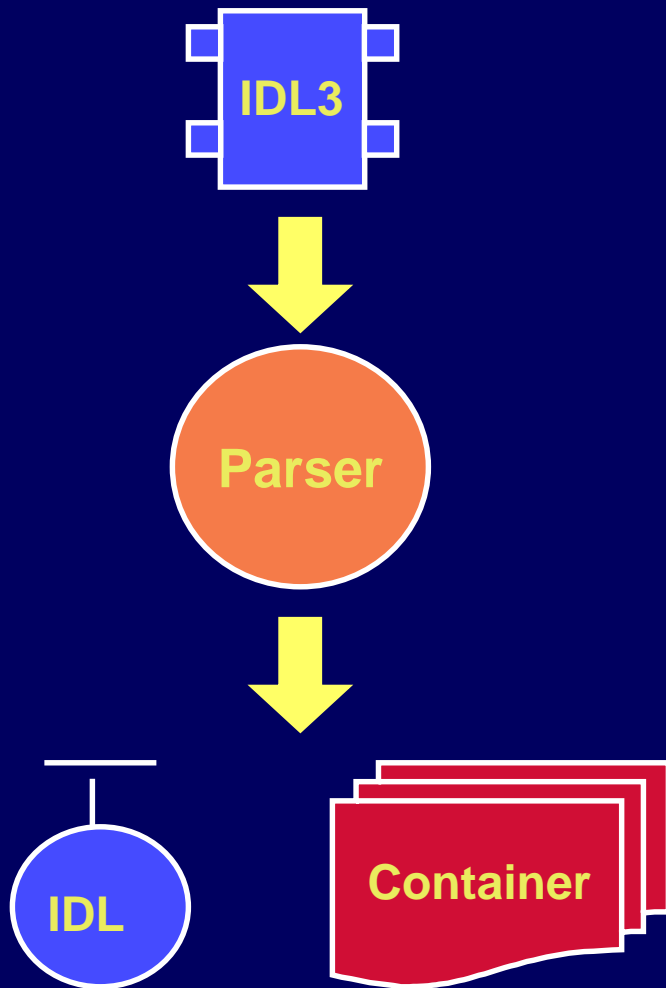


Components exchange information

Component/Containers

- Containers are like 'wrappers'
 - that isolate/protect the core function from the 'architecture' of the platform
 - provide a neutral and simple interface to the core function.
 - Tighten the rules for developers
 - but also...
 - can provide additional added value functions, such as data recording, data management
 - can be automatically generated.
 - E.g for a different architecture we need only use a different code generator.
-

Component IDL



- IDL3 is parsed, producing :
 - an IDL interface
 - defines the 'what' and not the 'how'
 - implemented by component developers
 - middleware *independent*
 - container code
 - provides the vendor specific 'how'

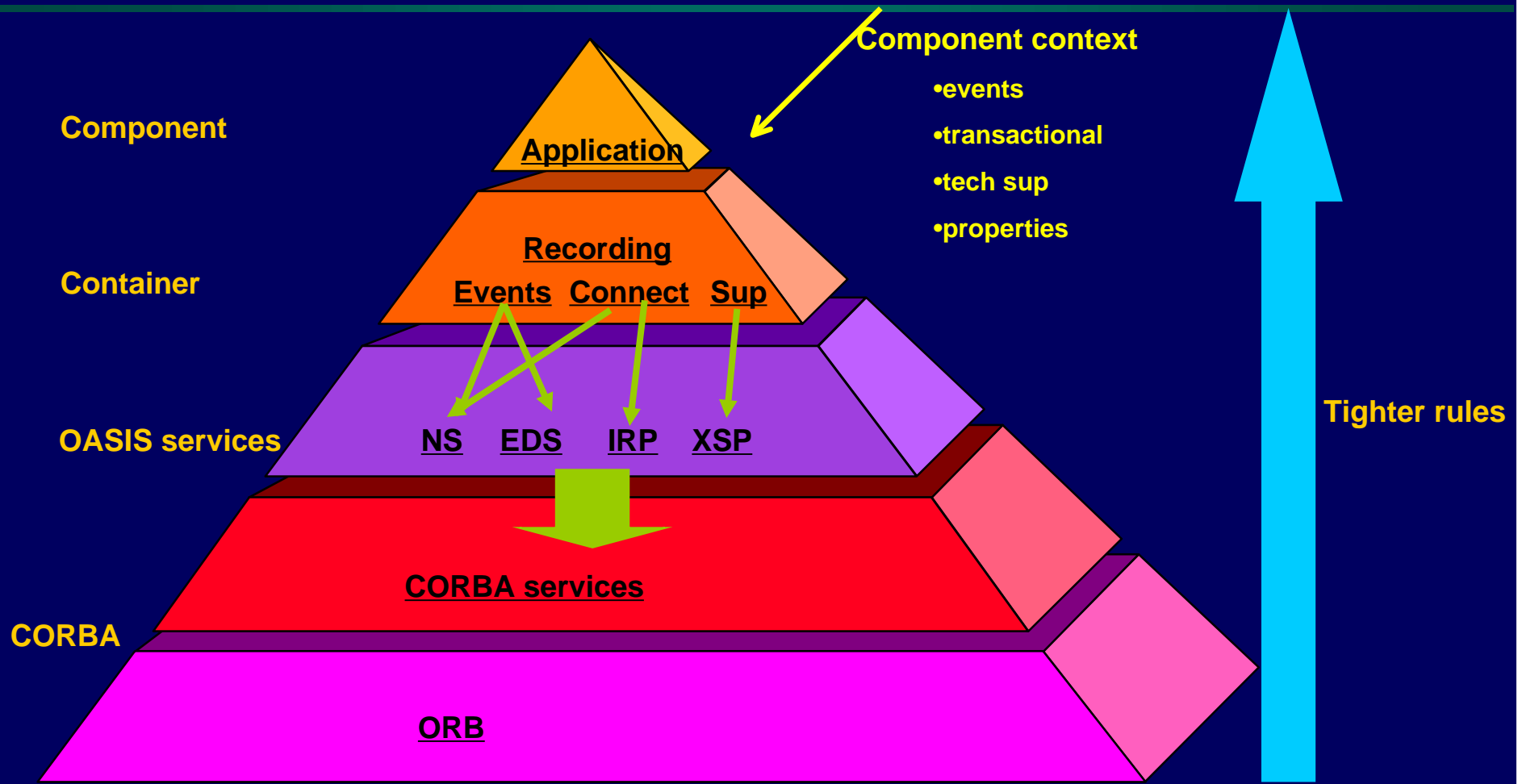
Contents

- ✓ Avenue Overview
 - ✓ Technical Overview
 - ✓ CCM
 - ➔ OASIS and PLUG
 - System Development Approach
 - Logical Model
 - Component Interface Definition (CIDs)
 - Component interconnection (first instance of the system).
-

OASIS (Open Architecture for Simulation System)

- CORBA middleware
 - large scale simulation and pre-operational validation
 - Built on ORBIX +..
 - lifecycle
 - publish subscribe
 - data management
 - supervision
 - supports ADA83, C++ and JAVA
-

Technical framework



OASIS in Avenue

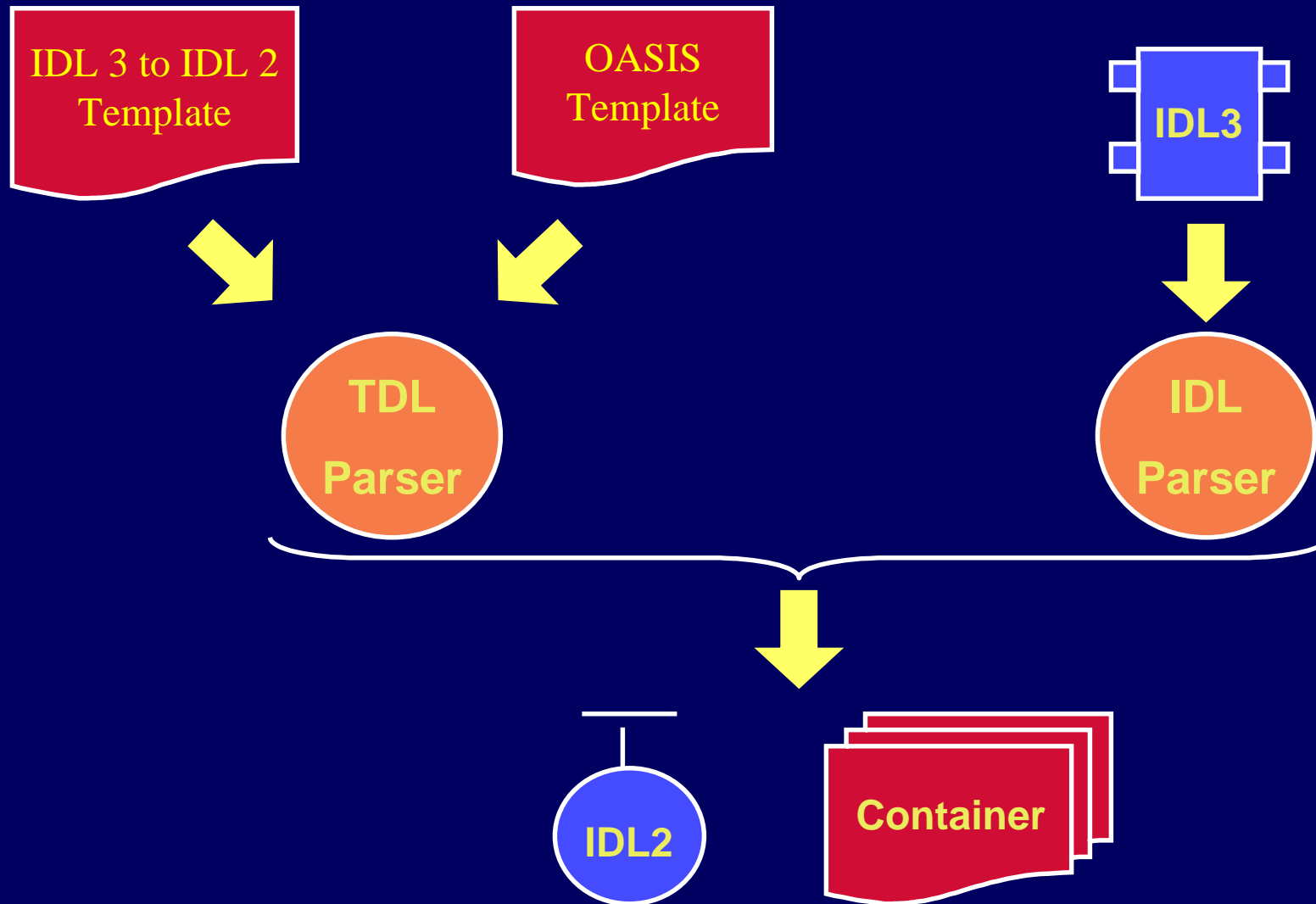
- Standard within EUROCONTROL
 - used by external partners
 - selected for AVENUE but upgraded
 - modified to CORBA component/container model.
 - Container uses OASIS services
 - Code generation (PLUG)
-

PLUG Code Generation Toolkit :

TDL parser

Plug : Presentation layer Universal Generator

TDL : Template Description Language



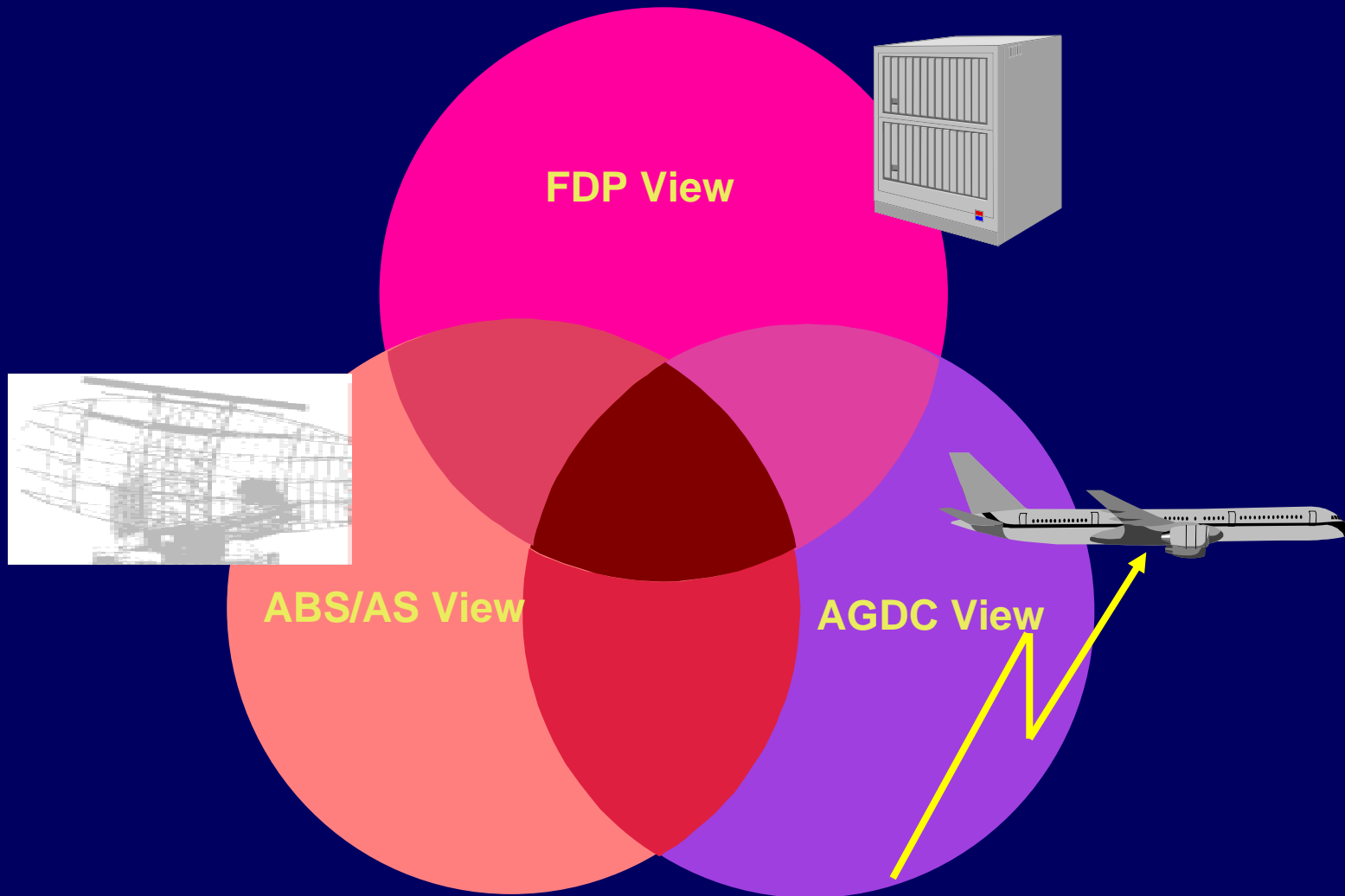
Contents

- ✓ Avenue Overview
 - ✓ Technical Overview
 - ✓ CCM
 - ✓ OASIS and PLUG
 - ➔ System Development Approach
 - Logical Model
 - Component Interface Definition (CIDs)
 - Component interconnection (first instance of the system).
-

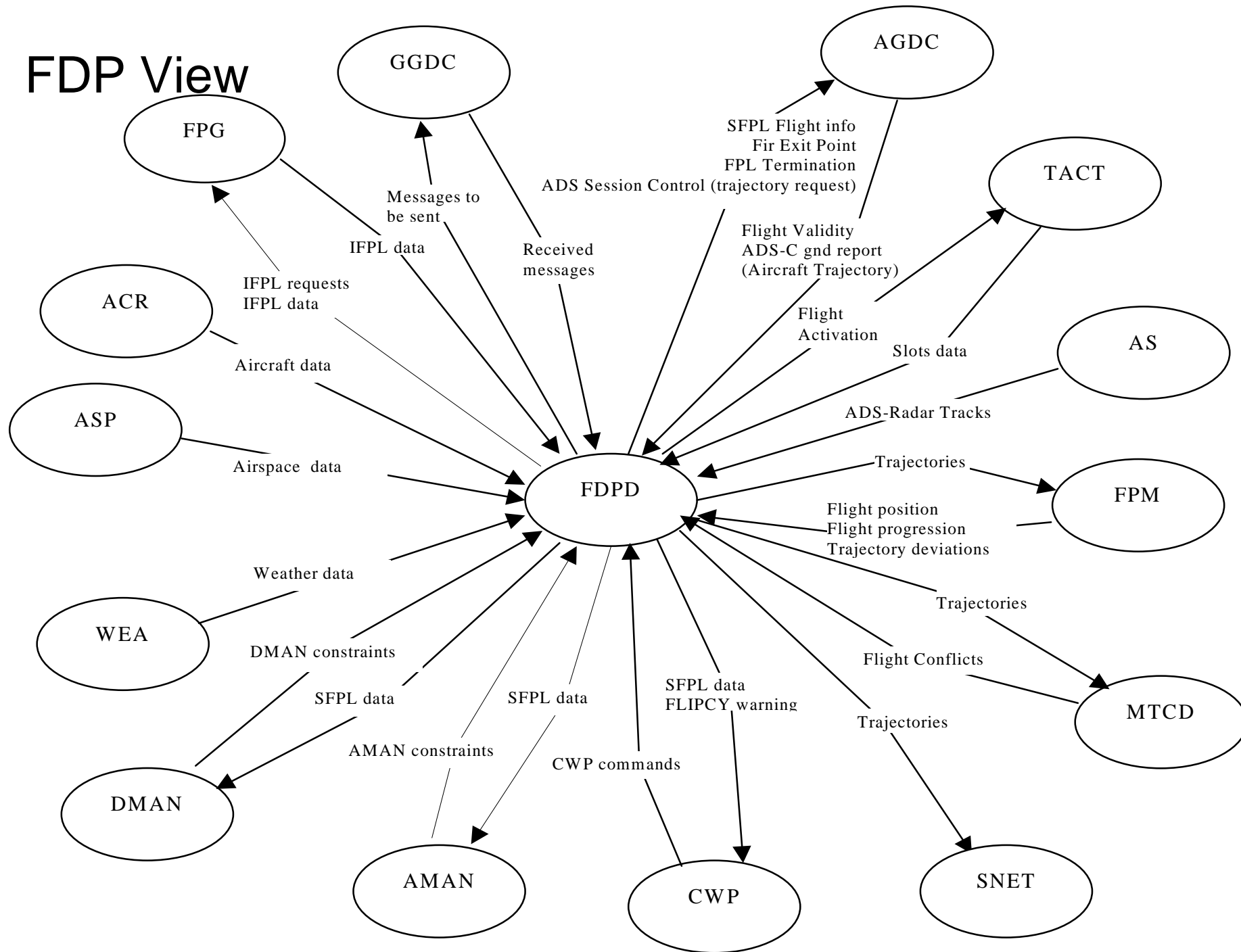
Logical architecture

- Logical modules derived from
 - CMS (common modular simulator)
 - Daarwin (Distributed ATM Architecture Integrating RNAV, Workstations, Tools and Networks)
 - ESCAPE (EUROCONTROL Simulation Capabilities And Platform for Experimentation)
 - PATIO (Platform for ATM Tools Integration up to Pre-operation)
 - Defined in SSDD (system/segment design document)
 - Interface definitions in terms of API (Provided interfaces in IDL).
-

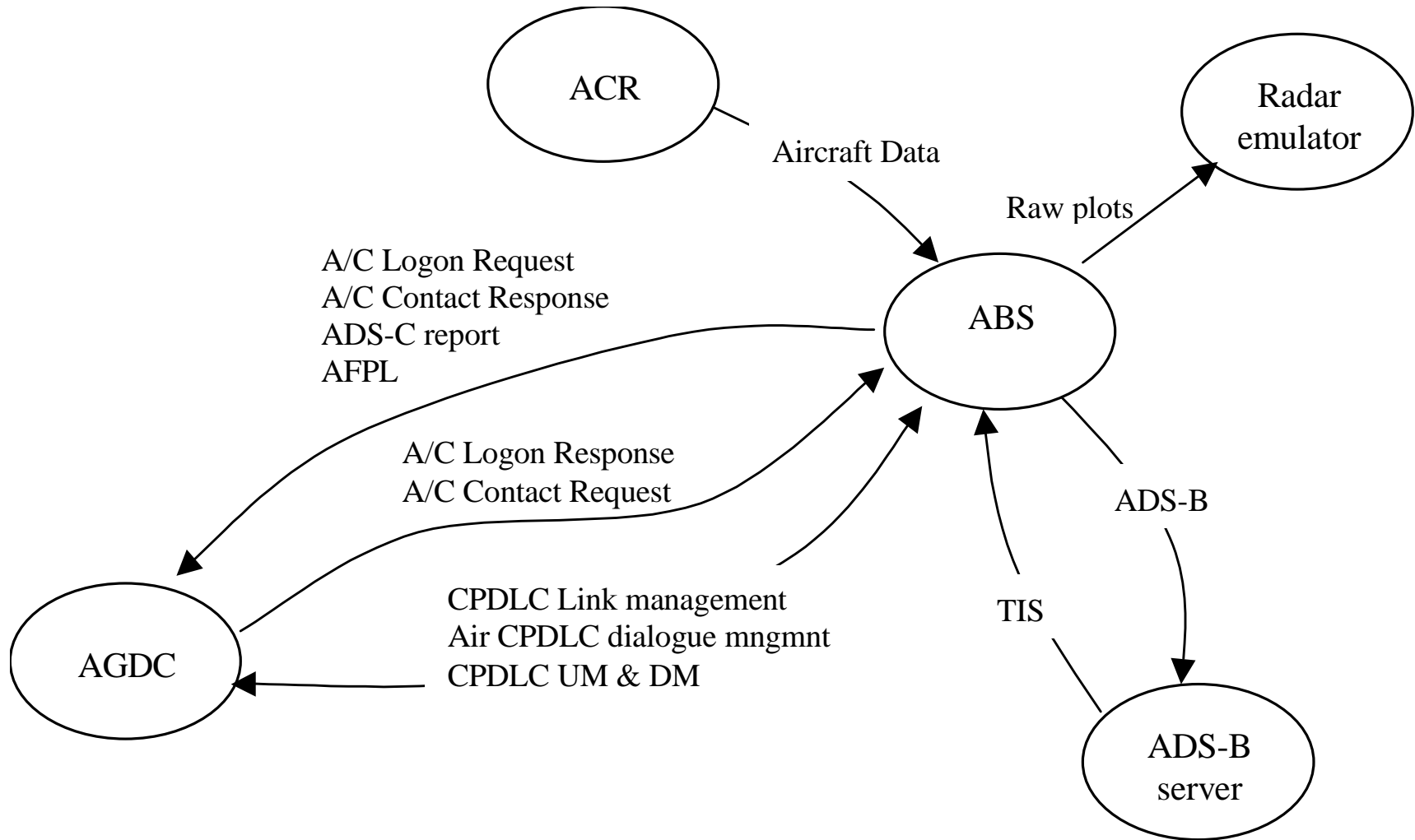
Decomposition is based on 3 views



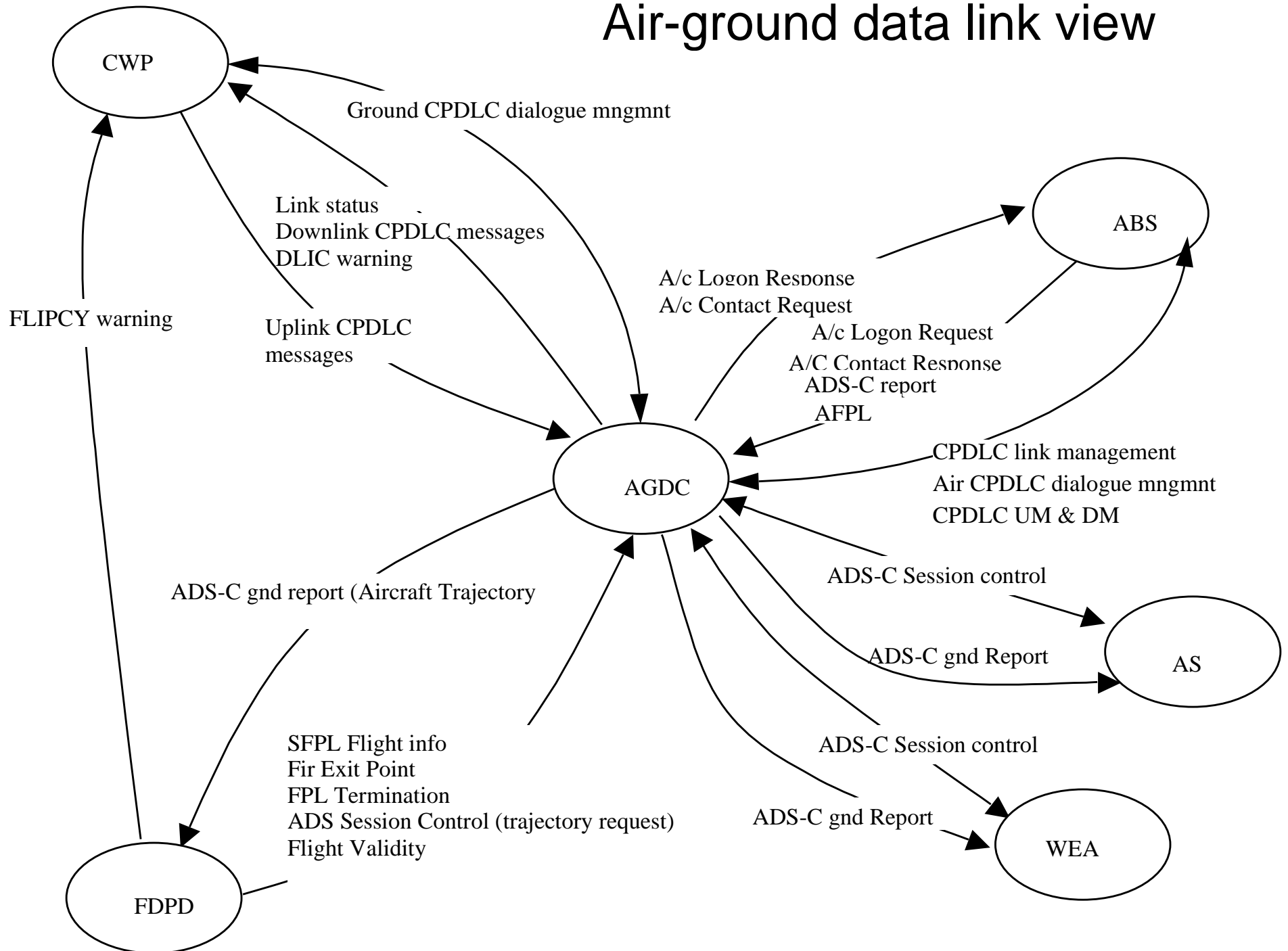
FDP View



Airborne/Surveillance View



Air-ground data link view

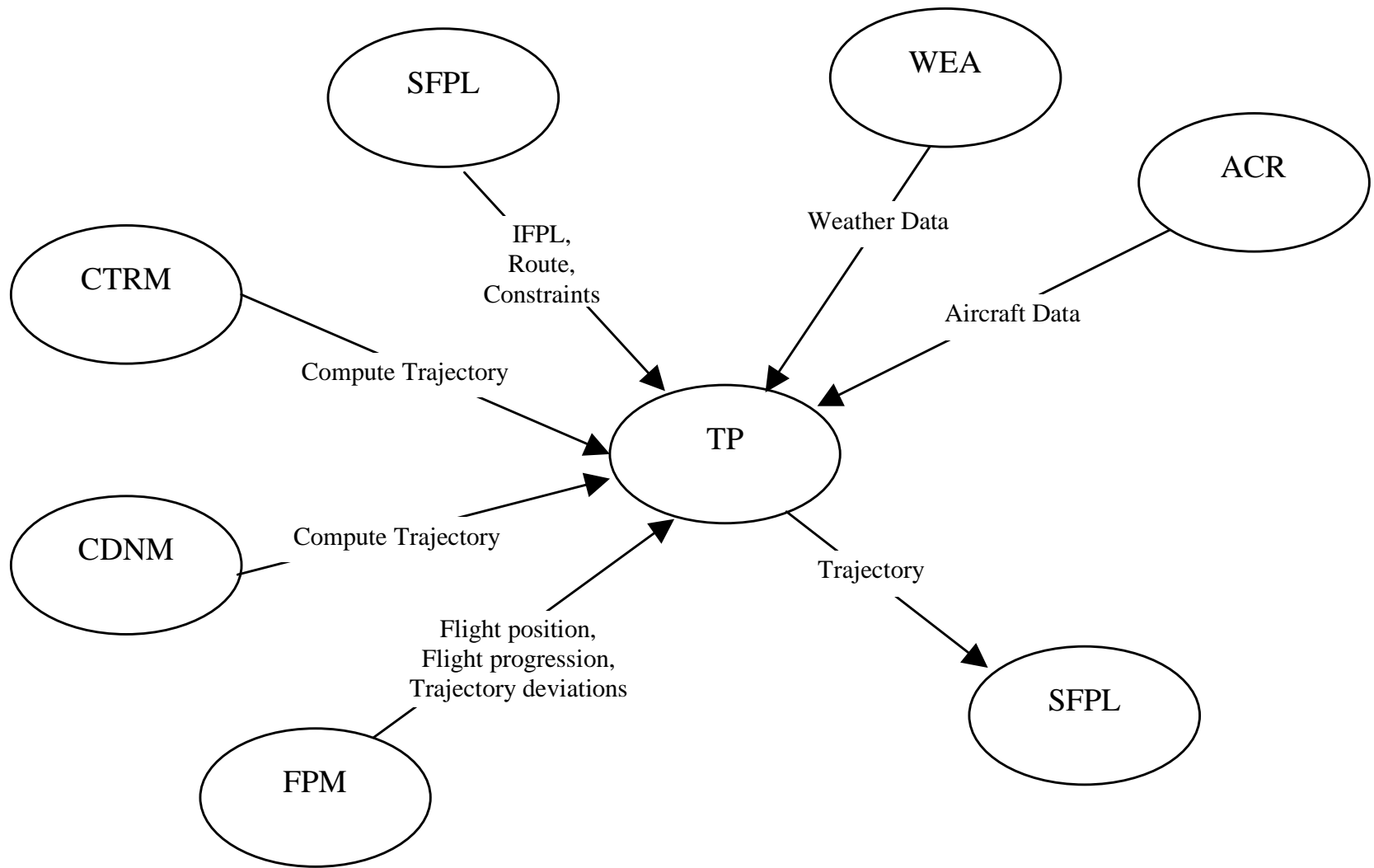


Modules are decomposed further into sub modules

- IFPM, supporting the Initial Flight Plan Information Management and the **IFPL** dataset,
- RTEM, supporting the Route Management and the **Route** dataset,
- CTRM, supporting the ATC Constraints Management and the **Constraints** dataset,
- TP, supporting the Trajectory Predictor and the **Trajectory** dataset,
- CDNM, supporting the Co-ordination Management and the **Traversed Sector List** and the **Co-ordination Sector List** datasets.
- SSRM, supporting the SSR Code Management and the **SSR Code** dataset,
- NTFM, supporting the Notification Management and the **Notified Sector List** dataset,
- CMCN, supporting the Civil/Military Crossings Management and the **Civil/Military Crossings** dataset,
- OCLM, supporting the Oceanic Clearance Management and the **Oceanic Clearance** dataset,
- CORL, supporting the Track/Callsign Correlation function and the **Correlation** dataset, and
- FLIPCY, supporting the Flight Plan Consistency function. No additional datasets are owned by this module.

Example: Trajectory Predictor

Context



APIs

- An API is defined for each module.
 - No mapping to *physical* components
 - No implementation details
 - Just a large number of services in CORBA IDL covering the ATM functions
 - First level (main) of interface harmonization
-

Example: Weather API

```
#include <AvWeaServices.idl>
#include <AvWeaEvents.idl>

component AvWea {

    //provided interfaces
    provides AvWeaServices::Services process; // processing services

    // Emitted events. publishes AvWeaEvents::Event sigmet_event;

};

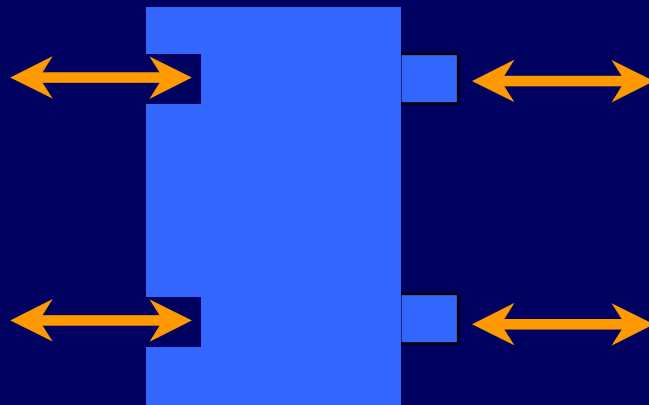
#endif // AV_WEA_MD_INCLUDED
```

Contents

- ✓ Avenue Overview
 - ✓ Technical Overview
 - ✓ CCM
 - ✓ OASIS and PLUG
 - ✓ System Development Approach
 - ✓ Logical Model
 - ➔ Component Interface Definition (CIDs)
 - Component interconnection (first instance of the system).
-

Components

- Modules are mapped to physical *'peer'* components
- Interfaces defined in CIDs (provided *and* required interfaces in IDL)



Required services

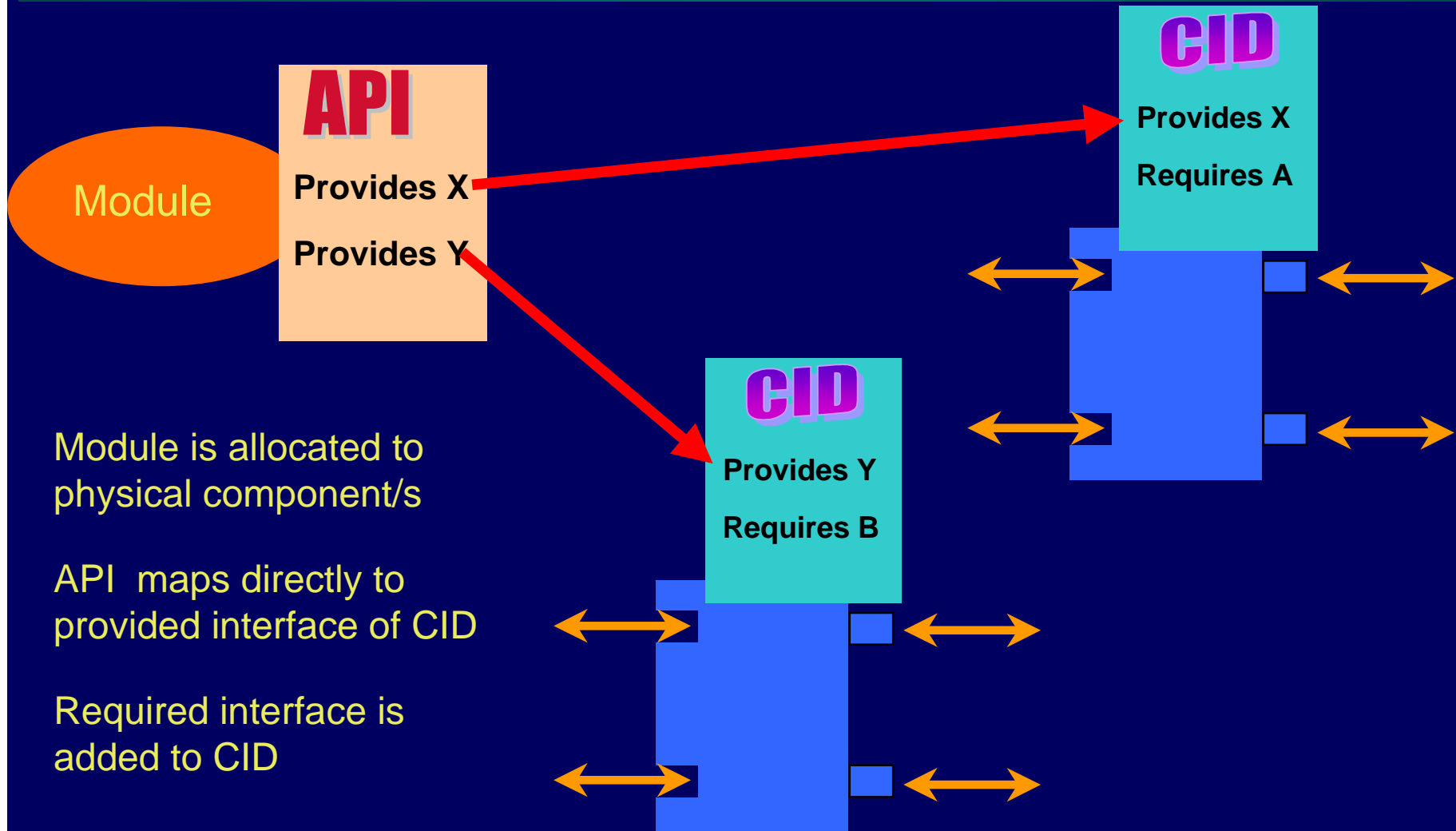


Provided services



CID

Relationship between API and CID



Module is allocated to physical component/s

API maps directly to provided interface of CID

Required interface is added to CID

Contents

- ✓ Avenue Overview
 - ✓ Technical Overview
 - ✓ CCM
 - ✓ OASIS and PLUG
 - ✓ System Development Approach
 - ✓ Logical Model
 - ✓ Component Interface Definition (CIDs)
 - ➔ Component interconnection (first instance of the system).
-

The first instance

- In parallel with module to component mapping is the physical architecture
 - connecting components together
 - static & dynamic modeling of the whole
-

