

---

# Aircraft flyover noise analysis

Acoustic data analysis

ASAM ODS / Web Service

Automatic test reporting

Example of application in AIRBUS Design Office

Aerospace Testing, Design & Manufacturing  
21-23 April 2009



# Overview

---

## ASAM ODS

- History
- Architecture
- Base model
- ATHOS server

## Acoustic data management

- How does it fit in with ASAM ODS?
- Software architecture
- OdsWebService
- General overview

## TrackReport

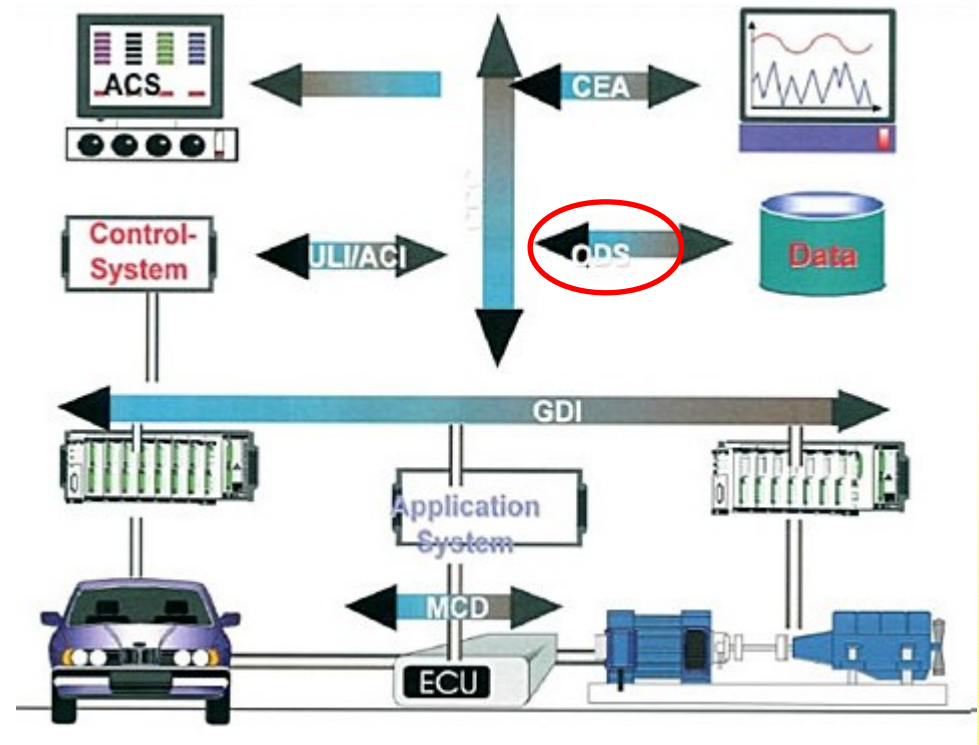
- Overview
- The new acoustic library
- Demo



# ASAM association

## Association for **S**tandardisation of **A**utomation and **M**easuring Systems

- Initiative of German Car manufacturers
- Standards for data models, interfaces and syntax specifications
- Applications: testing, evaluation, simulation
- More than 120 members



*ASAM interfaces for testing systems*



# ASAM ODS standard

## Open Data Services: version 5.1.1

### What for?

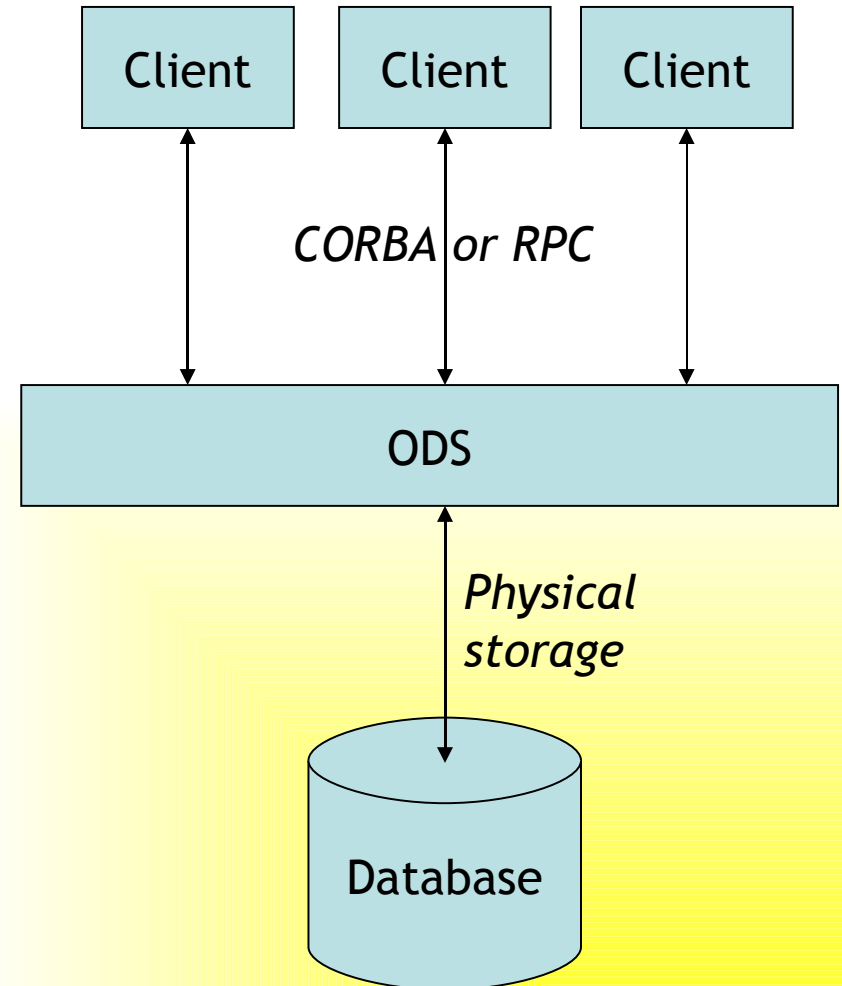
- Various clients, various data storage systems
- One common interface

### How?

- Defined by a standard
- Implemented by a server
- Over a DBMS
- Provides a CORBA or RPC interface

### Our clients

...usually use CORBA



## Object-based architecture

### Base model

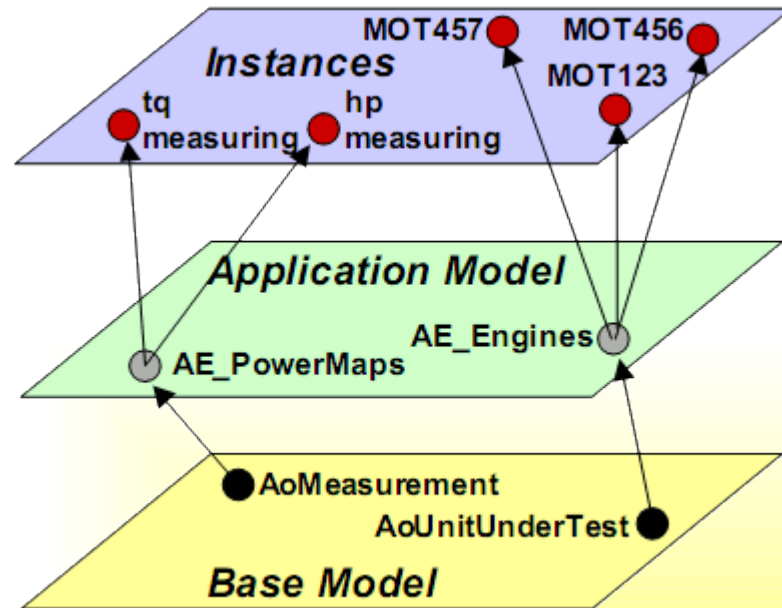
Minimal set of objects  
Rough data classification  
Covers the needs of various application areas

### Application model

Specialization of the elementary bricks  
Adapted to specific needs

### Instances

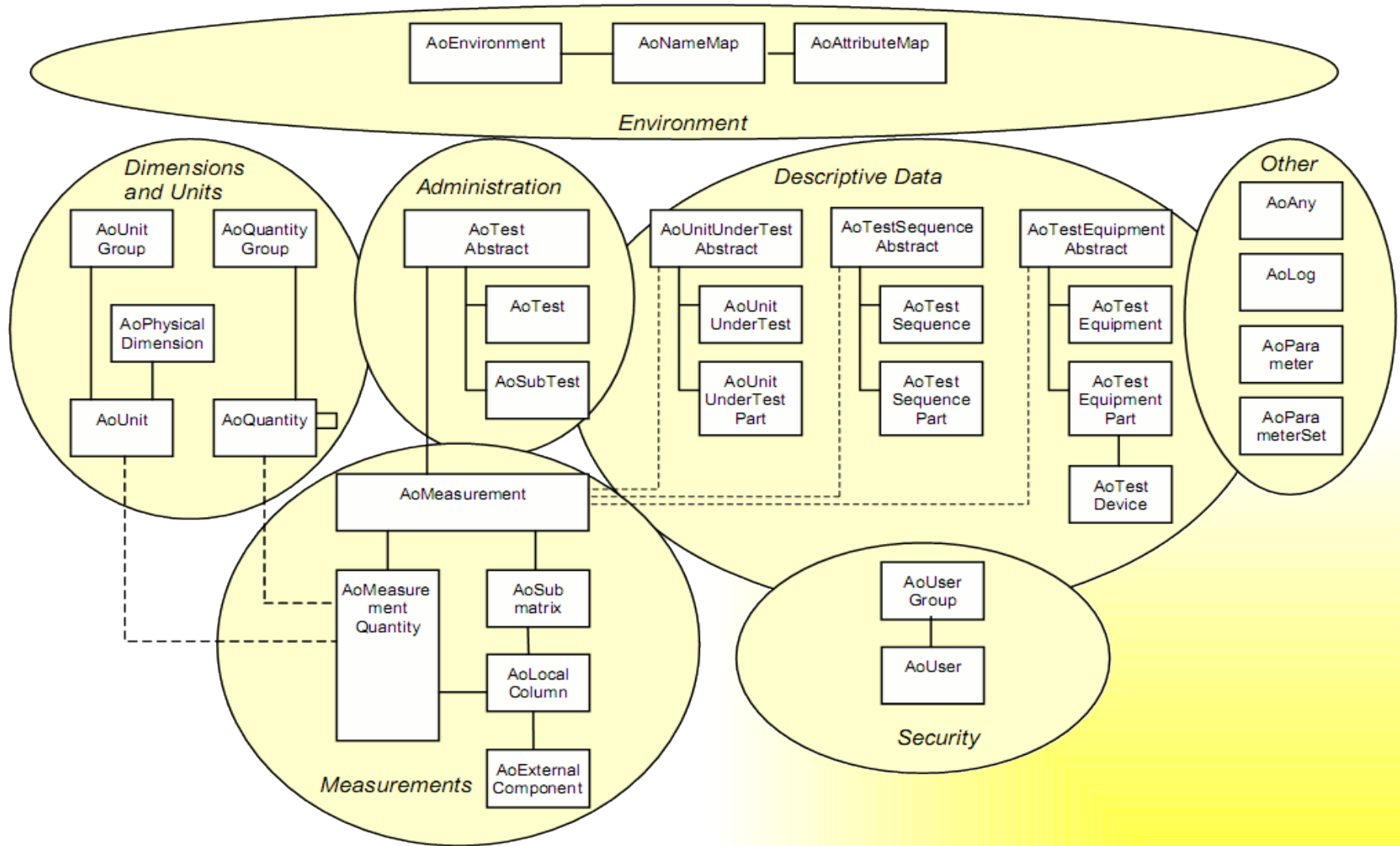
Test data is stored as instances of application elements



*Example relation between base model, application model and instances*



# ASAM ODS Base model



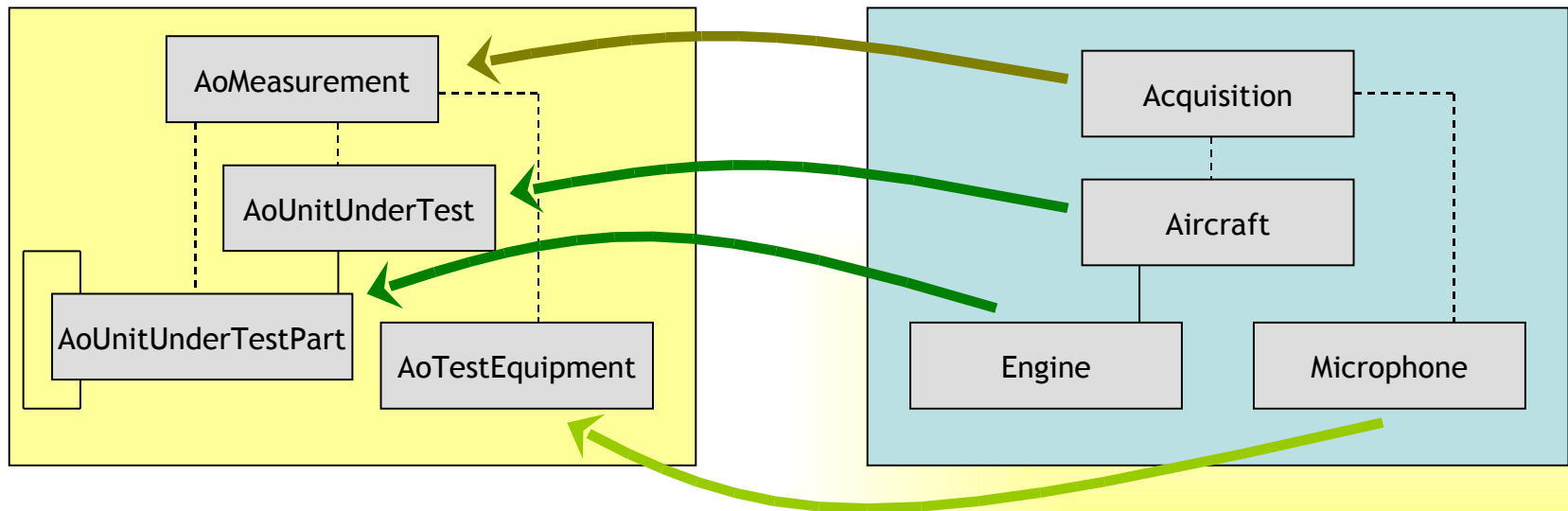
- Created by HighQSoft GmbH
- ASAM ODS v5 compliant
- CORBA and RPC implementation
- Full multithreading capabilities
- Mixed mode support
- Extended query support
- Various databases support



# Acoustic data management

Acoustic flight tests of aircrafts produces huge amounts of data

The hierarchy of concepts involved in the certification process fits in very well with the ASAM ODS base model



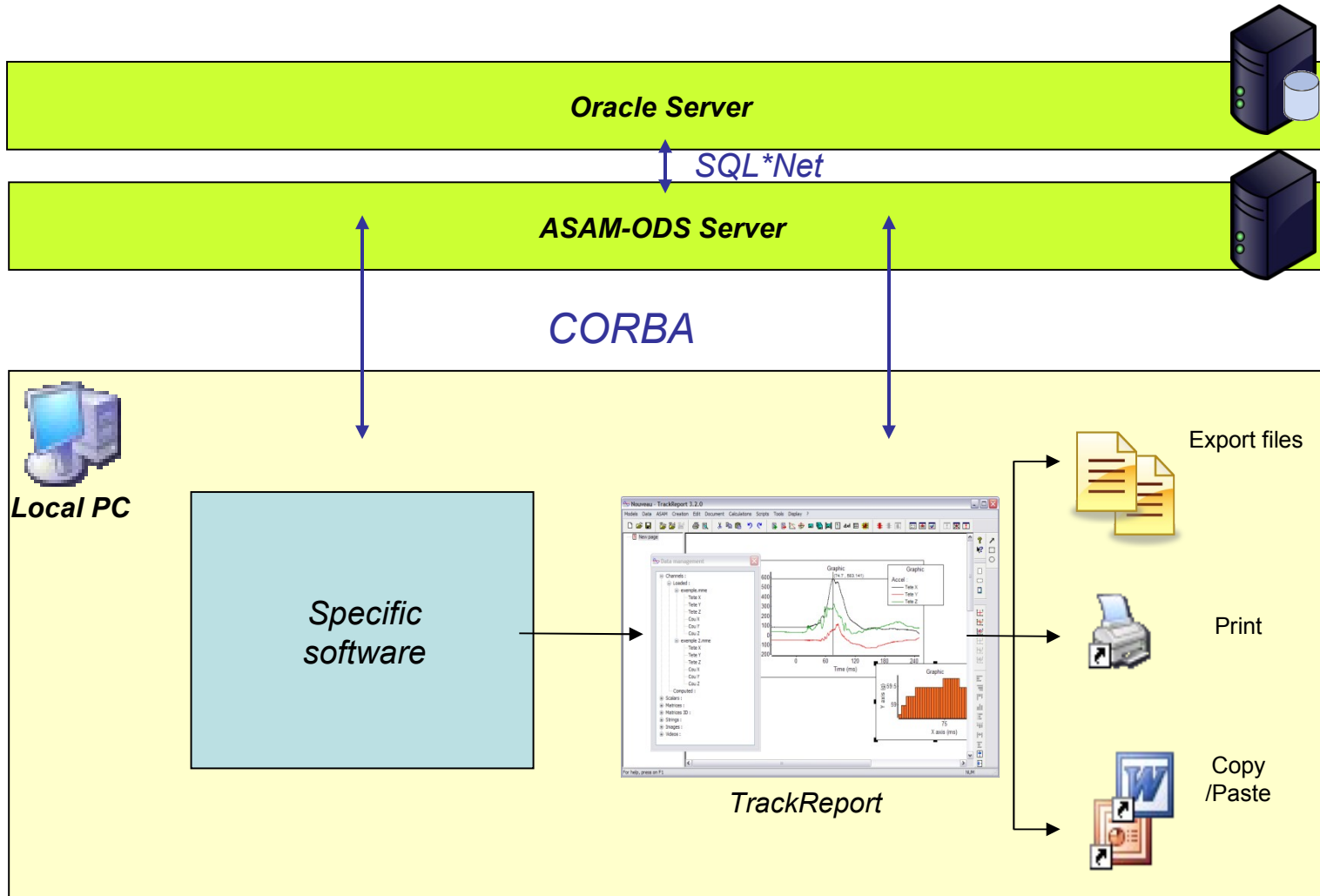
Security-related requirements would be fulfilled too

- Instance-wise authorizations
- Element-wise authorizations



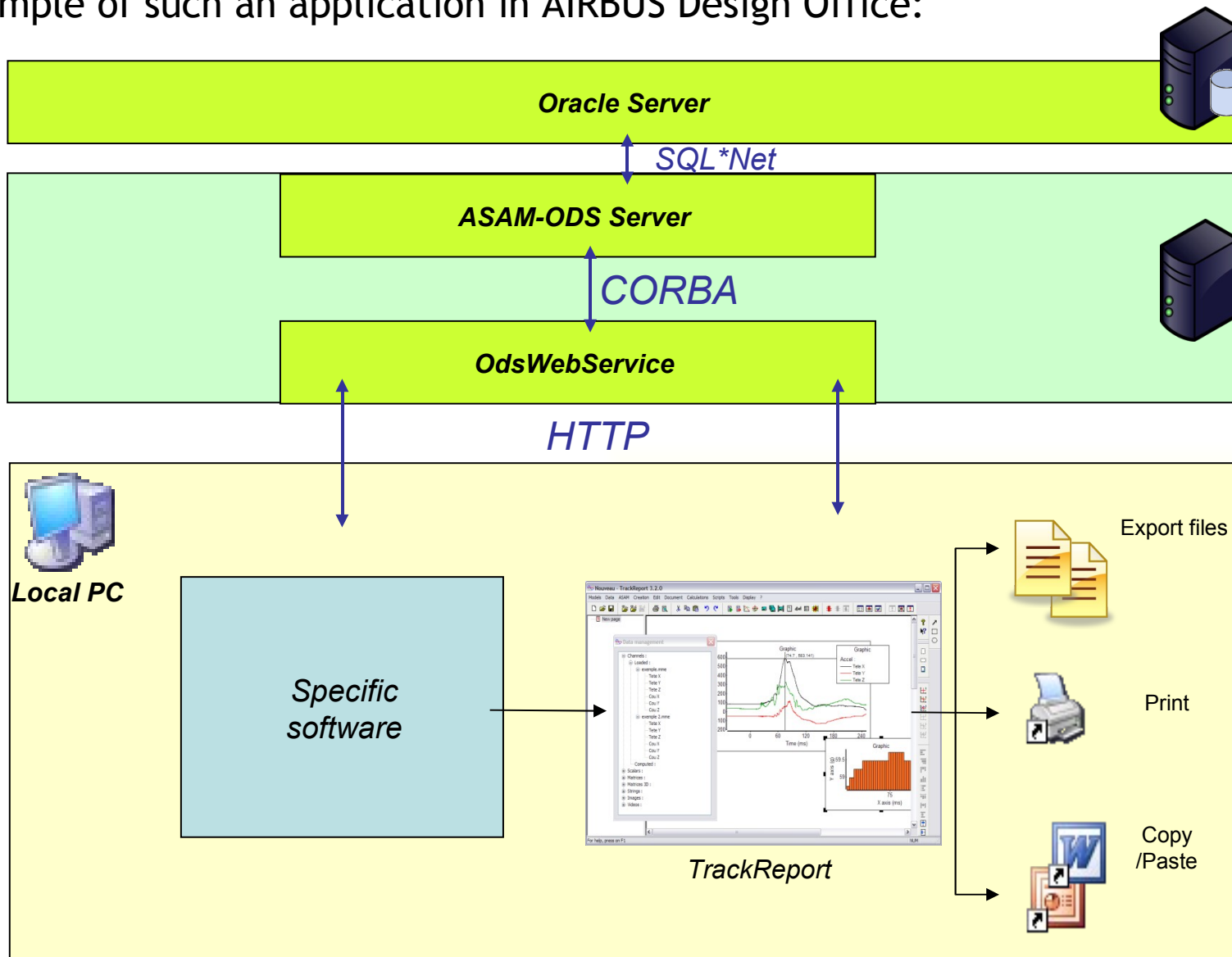


# Software architecture

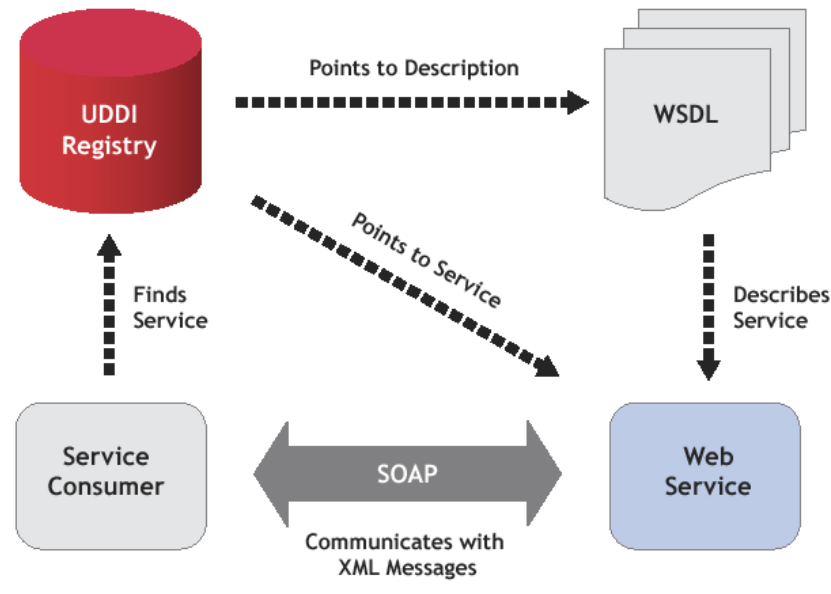


# Software architecture

Example of such an application in AIRBUS Design Office:



## What is a web service?



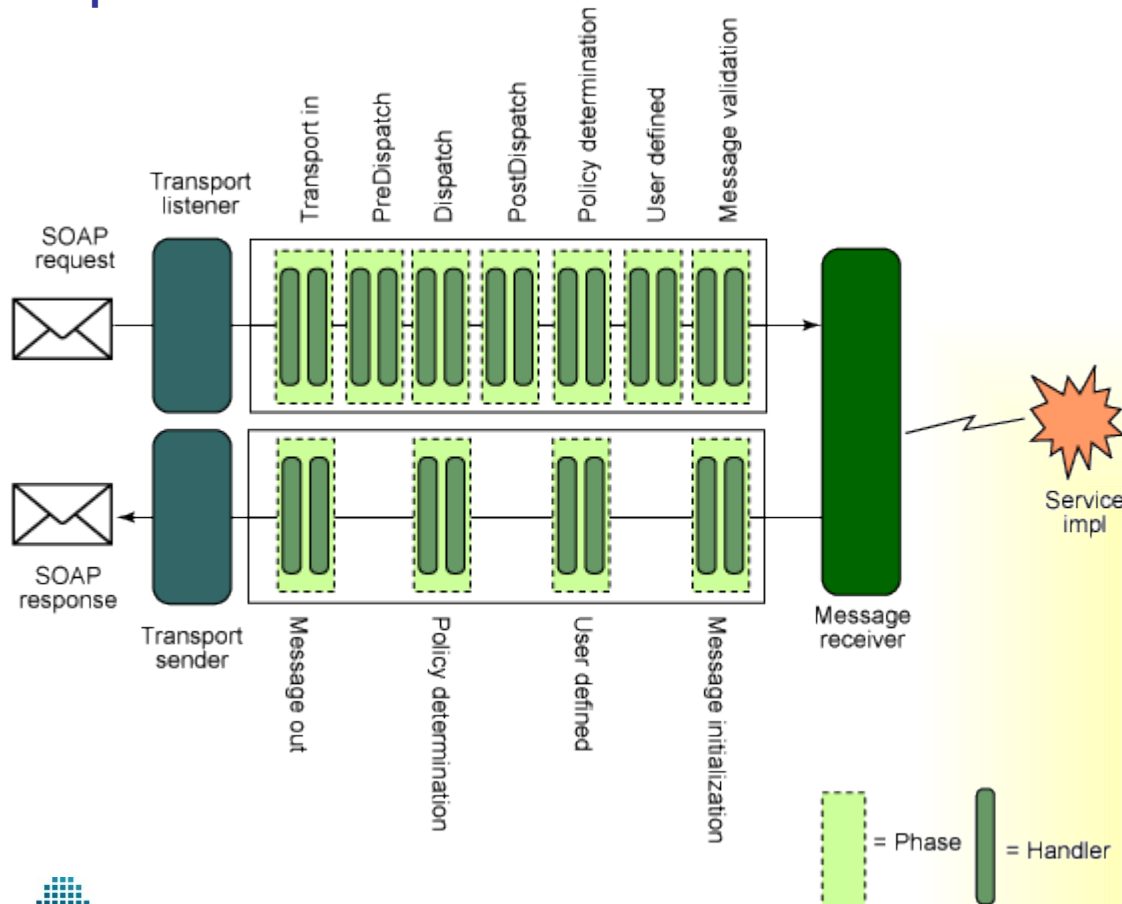
- Definition from W3C: "Web Service is a software application identified by a URI, whose interfaces and bindings are capable of being defined, described, and discovered by XML artifacts and which supports direct interactions with other software applications using XML-based messages via internet-based protocols".



# OdsWebService

OdsWebService is hosted by Apache Axis2 web services engine

## Apache Axis2



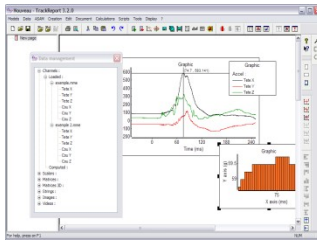
**Why do we need a web service engine?**

- Changes to the Web services landscape
- Performance
- Ease of use



# OdsWebService: Sample message exchange

Remote client



```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body>
    <tns:getAttributes
      xmlns:tns="http://www.orme-toulouse.com/OdsWebService/xsd">
      <tns:aid>4</tns:aid>
      <tns:iid>241</tns:iid>
      <tns:id_attr_name>id</tns:id_attr_name>
      <tns:attribute name="type" />
      <tns:attribute name="aircraft_version" />
      <tns:attribute name="status" />
      <tns:attribute name="creator_name" />
    </tns:getAttributes>
  </soapenv:Body>
</soapenv:Envelope>
```

Request

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body>
    <tns:getAttributesResponse
      xmlns:tns="http://localhost/OdsWebService/xsd">
      <tns:attribute tns:ods_type="DT_LONGLONG">
        <tns:name>aircraft_version</tns:name>
        <tns:value tns:flag="15">240</tns:value>
        <tns:unit_id>0</tns:unit_id>
      </tns:attribute>
      <tns:attribute tns:ods_type="DT_ENUM">
        <tns:name>status</tns:name>
        <tns:value tns:enumeration_type="result_status"
          tns:flag="15">non_validated</tns:value>
        <tns:unit_id>0</tns:unit_id>
      </tns:attribute>
      <tns:attribute tns:ods_type="DT_STRING">
        <tns:name>creator_name</tns:name>
        <tns:value tns:flag="15">user1</tns:value>
        <tns:unit_id>0</tns:unit_id>
      </tns:attribute>
    </tns:getAttributesResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

Response



OdsWebService



# TrackReport

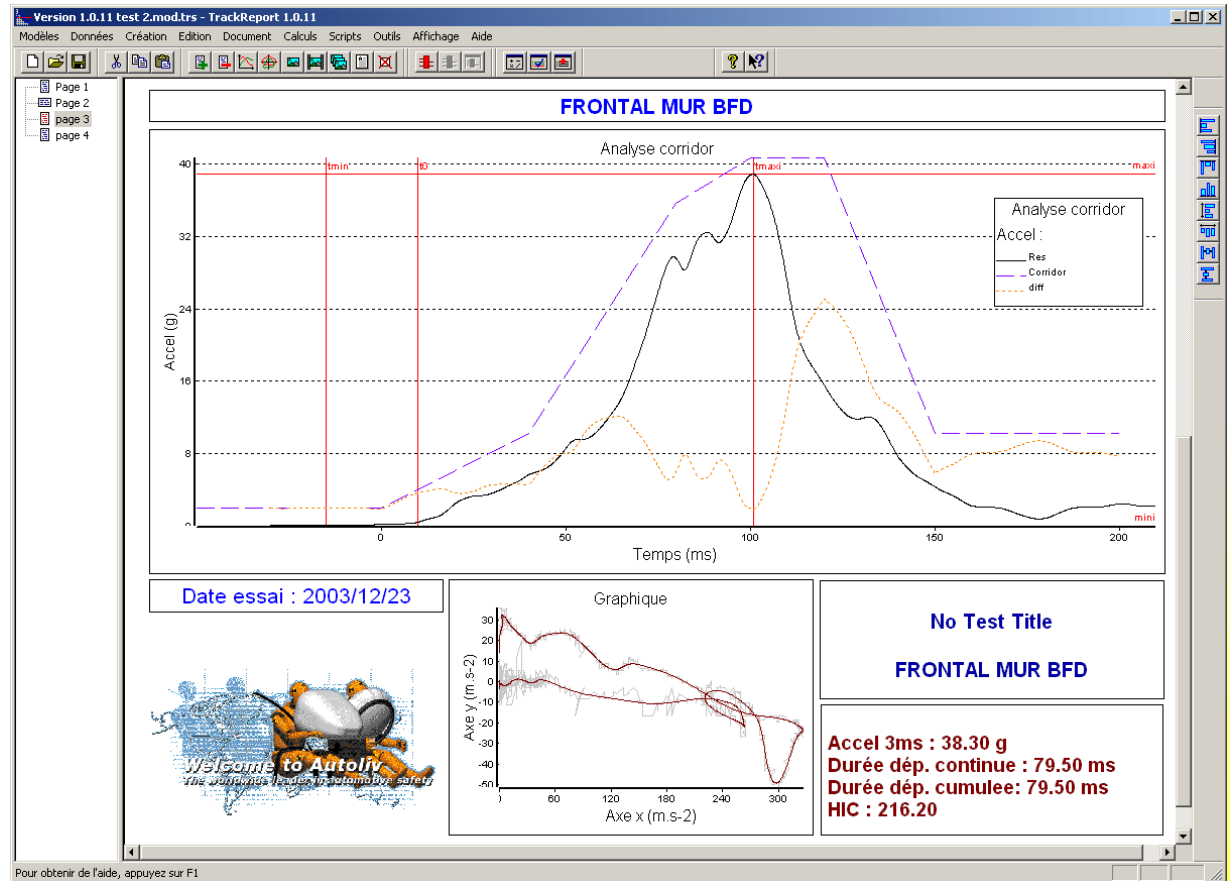
TrackReport is a graphical test report design tool

## Graphical objects:

- Curves  $y=f(x)$
- Parametric curves
- Images
- Image sequences
- AVI videos
- Dynamic text areas
- Tables
- ...

## Synchronization:

- Double cursors
- Curve/video synchro



# TrackReport

## Computation libraries:

- Arithmetic
- Numerical analysis
- Automotive
- Text strings
- Movies
- GUI
- Interpolation
- Signal
- Statistics
- Data synthesis
- Trigonometry
- Channels
- ...

## Unit manager :

- Automatic conversions
- Unit symbols management

The image shows two windows from the TrackReport software. The top window, titled 'HIC', is a configuration dialog for calculating Head Injury Criterion (HIC). It includes fields for 'Résultat : HIC max (scalaire)' (set to 'HIC'), 'Résultat : T1 (scalaire)' (set to 't1'), and 'Résultat : T2 (scalaire)' (set to 't2'). It also has dropdown menus for 'Composante X', 'Composante Y', and 'Composante Z' (all set to 'Tete X AVD Hrt', 'Tete Y AVD Hrt', and 'Tete Z AVD Hrt' respectively), and input fields for 'Instant t0', 'Durée (ms)' (set to 150), and 'Delta temps max (ms)' (set to 36). The bottom window, titled 'Pile de calculs', shows a list of operations: '\*\*\*\*\* VITESSES \*\*\*\*\*', 'vx = INTEGR(x (filtree), -15.052, 0)', 'vy = INTEGR(y (filtree), 4, 0)', 'vz = INTEGR(z (filtree), -10, 0)', 'module vitesse = RES3D(vx,vy,vz)', '\*\*\*\*\* CRITERES BIOMECHANIQUES \*\*\*\*\*', and '(HIC,,) = HIC (res,,,0.00 ms, 50.00 ms, 36.00 ms)'. Buttons for 'Annuler' and 'OK' are visible in both windows.

The image shows two windows from the TrackReport software. The top window, titled 'Gestionnaire d'unités', is a unit manager dialog. It features a tree view on the left with categories like 'Sans unité', 'Temps', 'Distance', 'Surface', 'Volume', 'Angle', 'Vitesse', 'Vitesse Rotation', 'Accélération', 'Accélération Rotation', and 'Masse'. On the right, there are buttons for 'Ajouter un type d'unité', 'Ajouter une unité', 'Ajouter un symbole', 'Supprimer', and 'Retour aux unités'. The bottom window, titled 'Aperçu rapide', is a graph showing a signal labeled 'z (signed)'. The y-axis is 'Axe y (m.s-2)' ranging from 0 to 200, and the x-axis is 'Axe x (s)' ranging from 0 to 0.2. The graph shows a signal that starts at 0, rises to a peak of approximately 200 at 0.12 seconds, and then decays back to 0. Buttons for 'OK' and 'Annuler' are visible at the bottom.



# TrackReport

## TrackL Scripts:

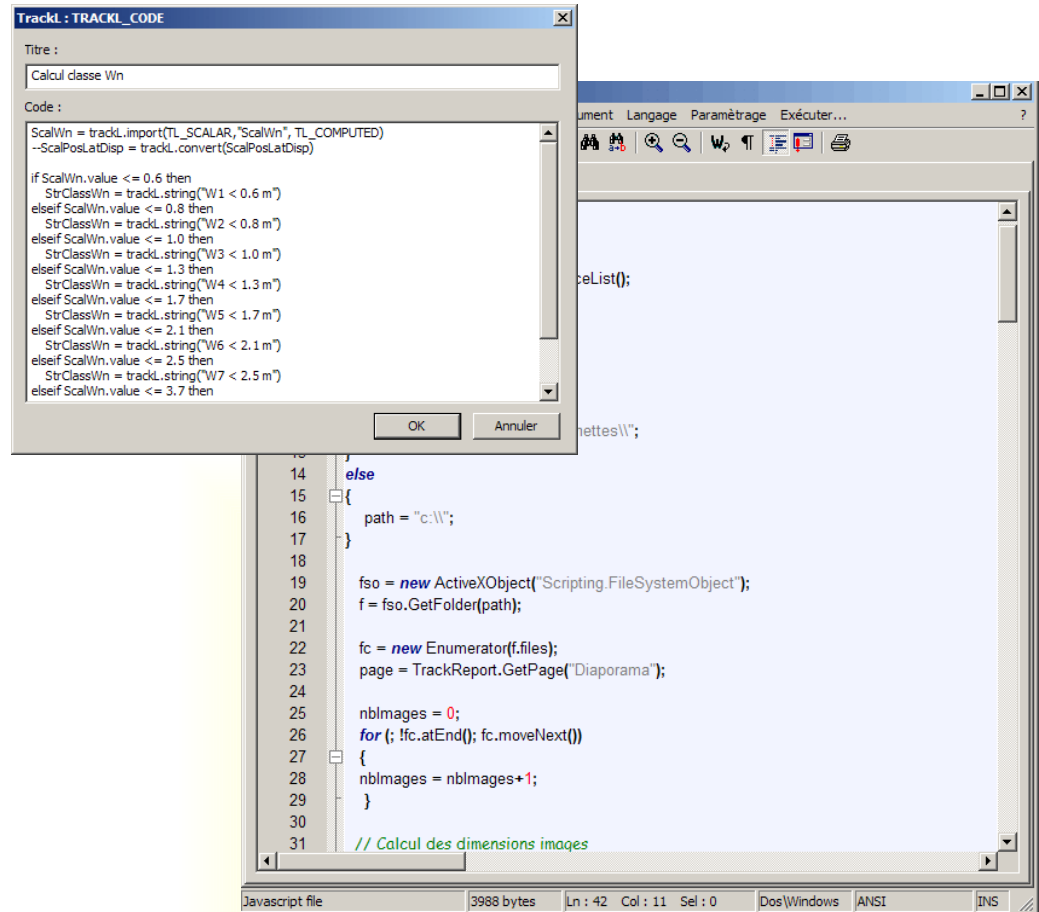
- Vector calculus
- New calculation functions

## JavaScript:

- New calculation interfaces
- Task automation
- Dynamic reports
- TrackReport driving

## Plug-in development Kit:

- New calculation functions
- New data readers



The image shows two overlapping windows from a software application. The top window, titled "TrackL : TRACKL\_CODE", contains a script for calculating window classes based on width. The bottom window, titled "Javascript file", contains JavaScript code for file system operations and report generation.

```
Titre :
Calcul classe Wn

Code :
ScalWin = trackL.import(TL_SCALAR, "ScalWin", TL_COMPUTED)
--ScalPostLatDisp = trackL.convert(ScalPostLatDisp)

if ScalWin.value <= 0.6 then
  StrClassWn = trackL.string("W1 < 0.6 m")
elseif ScalWin.value <= 0.8 then
  StrClassWn = trackL.string("W2 < 0.8 m")
elseif ScalWin.value <= 1.0 then
  StrClassWn = trackL.string("W3 < 1.0 m")
elseif ScalWin.value <= 1.3 then
  StrClassWn = trackL.string("W4 < 1.3 m")
elseif ScalWin.value <= 1.7 then
  StrClassWn = trackL.string("W5 < 1.7 m")
elseif ScalWin.value <= 2.1 then
  StrClassWn = trackL.string("W6 < 2.1 m")
elseif ScalWin.value <= 2.5 then
  StrClassWn = trackL.string("W7 < 2.5 m")
elseif ScalWin.value <= 3.7 then

14 }
15 else
16 {
17   path = "c:\\";
18 }
19 fso = new ActiveXObject("Scripting.FileSystemObject");
20 f = fso.GetFolder(path);
21
22 fc = new Enumerator(f.files);
23 page = TrackReport.GetPage("Diaporama");
24
25 nbImages = 0;
26 for (; !fc.atEnd(); fc.moveNext())
27 {
28   nbImages = nbImages+1;
29 }
30
31 // Calcul des dimensions images
```



- Logarithmic sum / difference / average
- A, B, C weighting
- Neat-band  $\leftrightarrow$  Third-octave-band  $\leftrightarrow$  Octave-band conversions
- Atmospheric absorption
- Directivity indexes
- Spherical divergence estimation
- Spectrum / matrix OASPL computation
- Noisiness, PNL, TPNL, EPNL
- Tone correction



---

Demo...

