

# The SAMANTA platform

**Jérôme LACAILLE**

*Emeritus Expert*

**SNECMA**

[jerome.lacaille@sneema.fr](mailto:jerome.lacaille@sneema.fr)

+33 1 60 59 70 24

**Aurélie GOUBY**

*Department Prognostic Health Monitoring Systems*

**SNECMA**

[aurelie.gouby@sneema.fr](mailto:aurelie.gouby@sneema.fr)

+33 1 60 59 42 53

**/01/**

# Snecma and Health Monitoring

# Snecma, Key figures

Revenue in Billion €\* 5,9

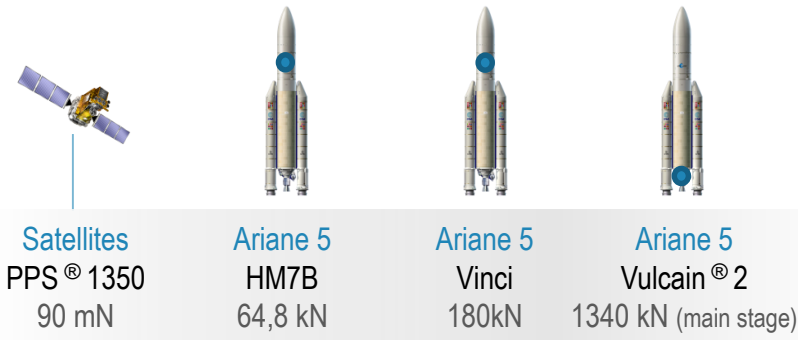
Employees worldwide \* 14 662

Sites worldwide 35

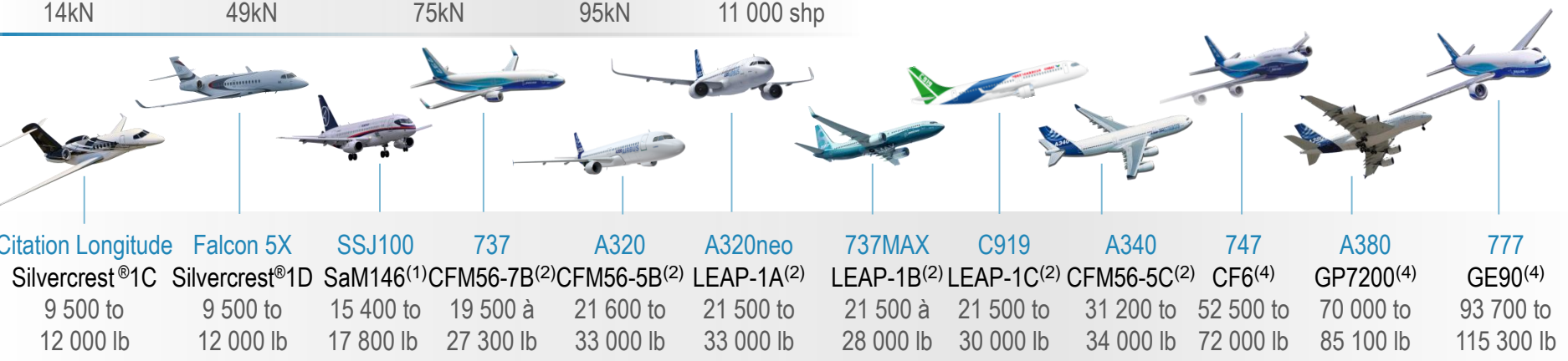
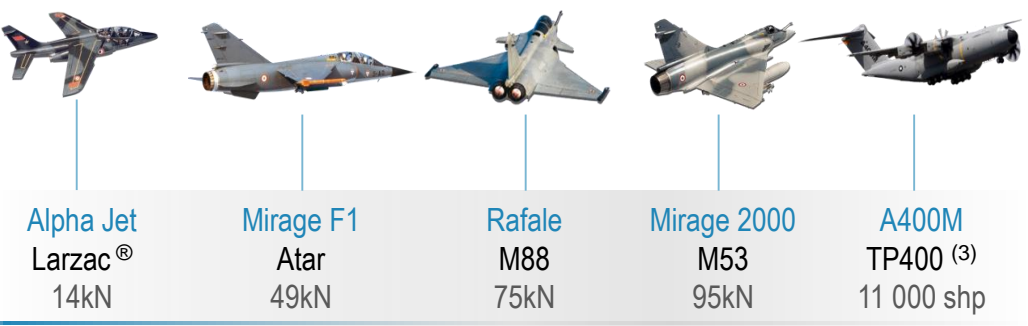


\* As of 2013 December 31st

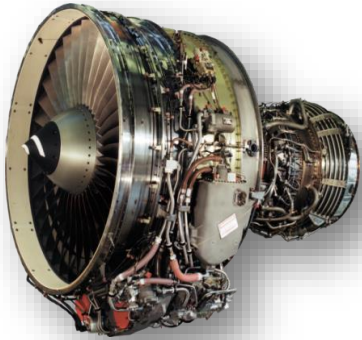
# Thrust: from 9 grams to 135 000 kilograms



(1) PowerJet (50/50 Snecma-NPO Saturn)  
 (2) CFM International (50/50 Snecma-GE)  
 (3) EPI (ITP, MTU, Rolls-Royce, Snecma)  
 (4) En coopération avec GE



# CFM56, THE worldwide civil engine aircraft best-seller



## CFM56-5B

Nearly 105 millions of cumulative flight hours

More than 5 895 engines in service at 198 operators

Thrust range: 21 600 to 32 000 lb

Applications : Airbus Family A318, A319, A320, A321



A320



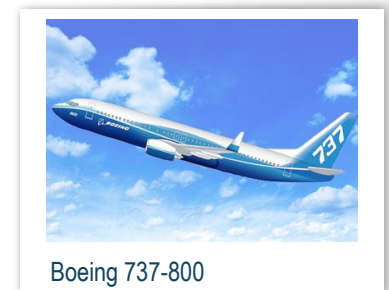
## CFM56-7B

More than 195 millions of cumulative flight hours

9 801 engines in service at 265 operators

Thrust range : 19 500 to 27 300 lb

Applications : Boeing Family 737-600, 737-700, 737-800, 737-900



Boeing 737-800

As of 2013 August 31st

CFM a 50/50 joint company of Snecma (Safran) and GE.

# Health Monitoring

## → Health Monitoring is a nearly real-time monitoring of system parameters in order to:

- Detect signs of failure
- Predict the remaining time before a required intervention (inspection or maintenance)
- Identify faulty components

## → Improve operational aircraft availability

- Aircraft takes off on time, lands on time at destination
- Reduction of operational events
- Planning and optimization of maintenance
- Reduction of stopping time engine

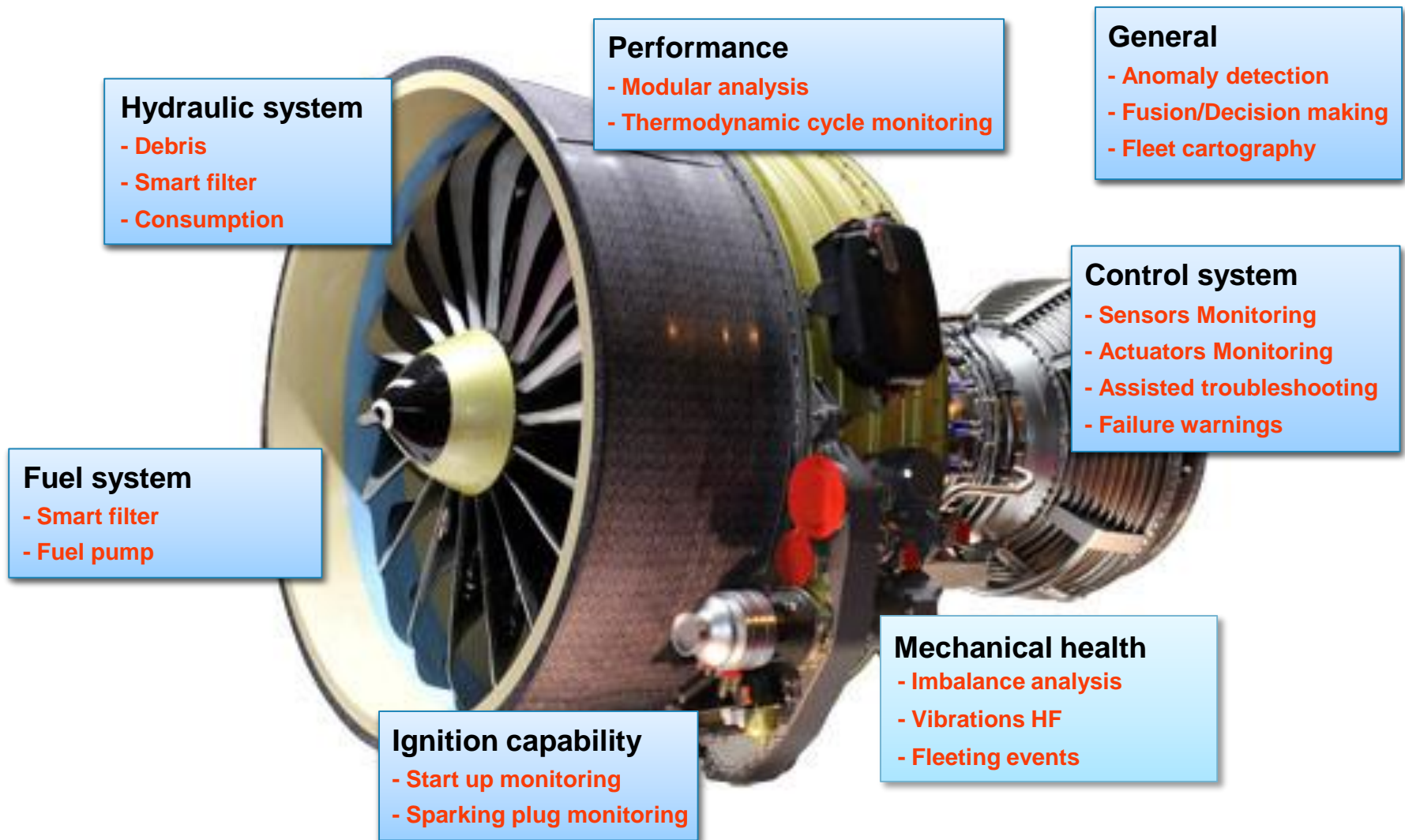
## → Reduce maintenance costs

- Assisted troubleshooting (isolating the faulty element)
- Limit secondary damage

## → Note: The health monitoring is a maintenance function, no impact on the current flight and aircraft safety



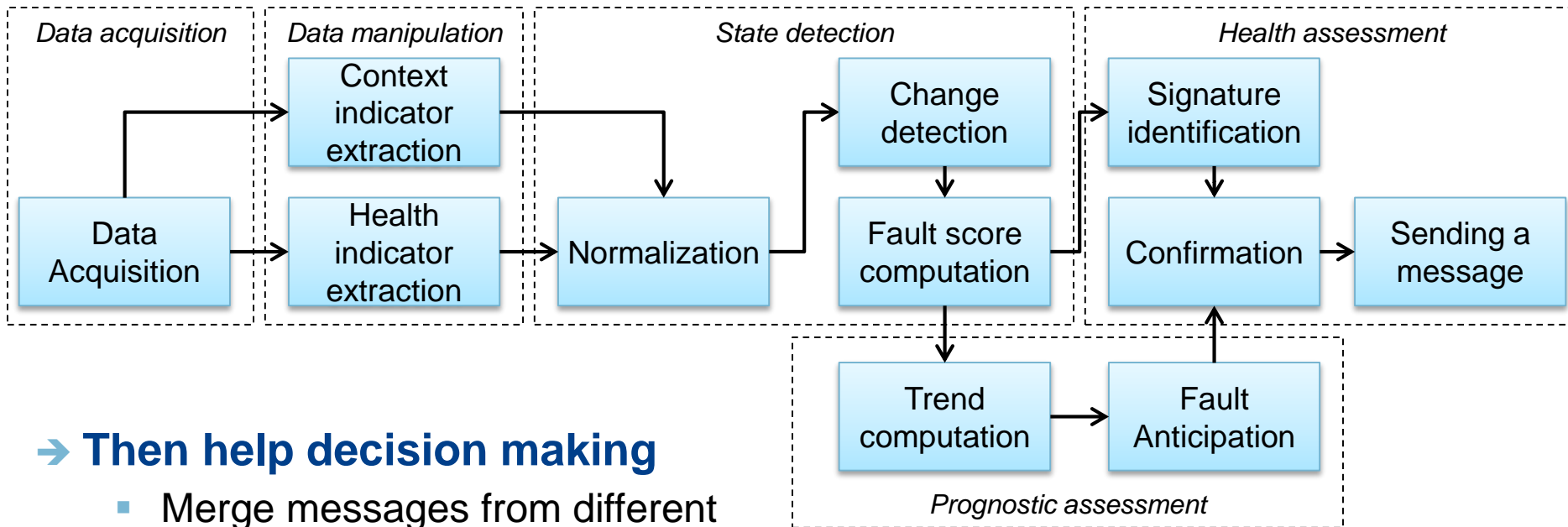
# Engine systems monitored



# Faults diagnose

## → Building alert messages from acquired measurements

- Using physical models
- Learning behavior using mathematical models
- Identification of fault signatures, drifts detection, component targeting



## → Then help decision making

- Merge messages from different diagnose sources



# /02/

## Introduction to SAMANTA

*Snecma Algorithm Maturation ANd Test Application*

# Introduction to SAMANTA

## → What is SAMANTA ?

- SAMANTA (Snecma Algorithm Maturation ANd Test Application) is an environment for **Design, Development** and **Maturation** of algorithms based on 3 items

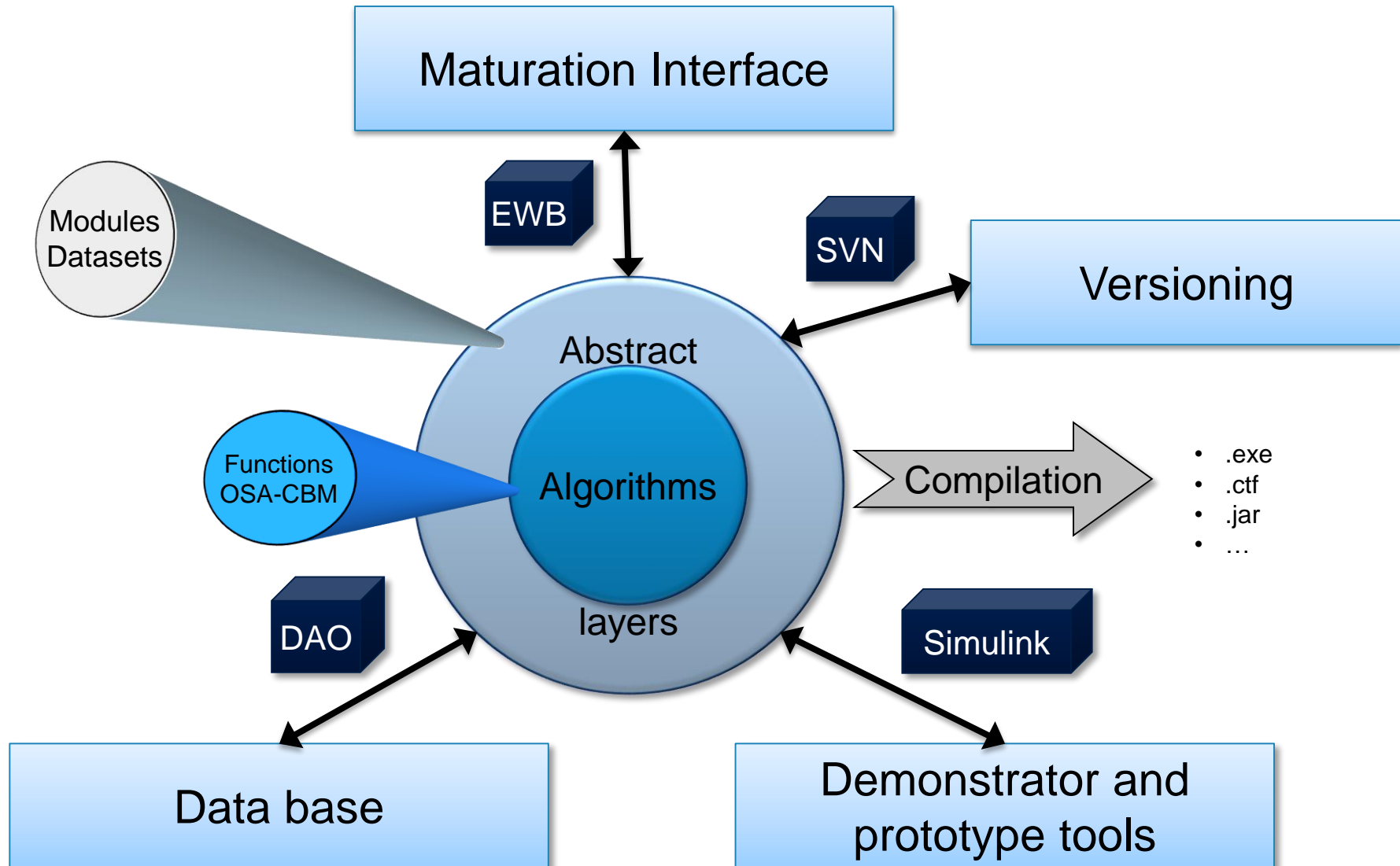
## → Why develop this platform ?

- To enable engineers to quickly and easily lay out algorithms without special knowledge in mathematics or computer science
- To facilitate exchanges between algorithm designers through scripts/displays/ common and consistent operations
- To capitalize on algorithms
- To create a complete interface between algorithms, data and associated documents

## → Why with MATLAB?

- To quickly and easily create a platform giving the freedom to design algorithms through written code or through connecting blocks in Simulink

# SAMANTA environment



# /03/

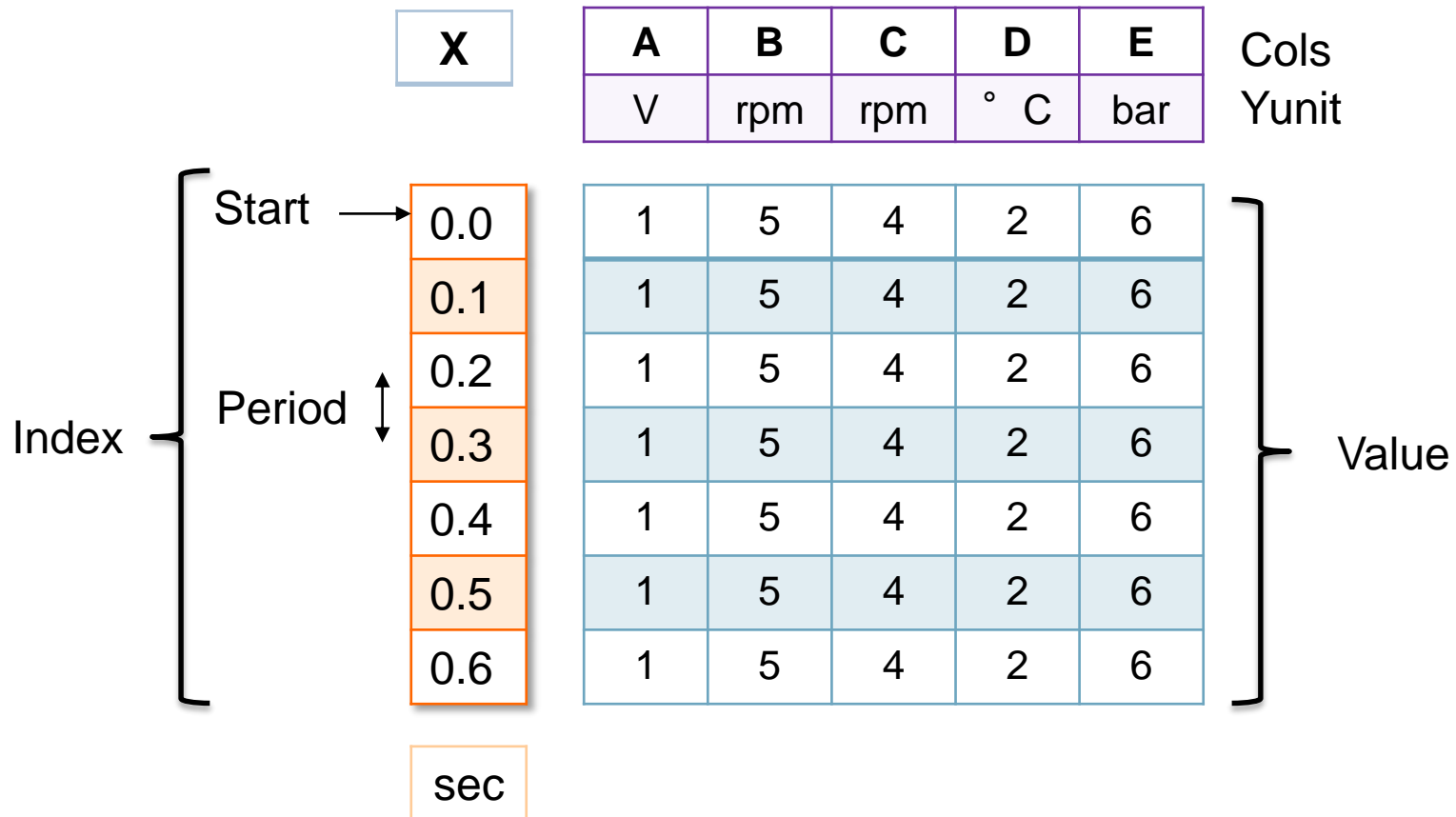
## Presentation of SAMANTA objects

*Signal, Opset, Mnode*

# SAMANTA objects – the signal

- **Algorithms process input data and produce results. These data come from:**
  - Flight measures or flight tests
  - Intermediate results produced by other algorithms
- **The SAMANTA *signal* is a structure containing and formalizing data through a standard:**
  - Table of values
  - Associated properties

# SAMANTA objects – the signal

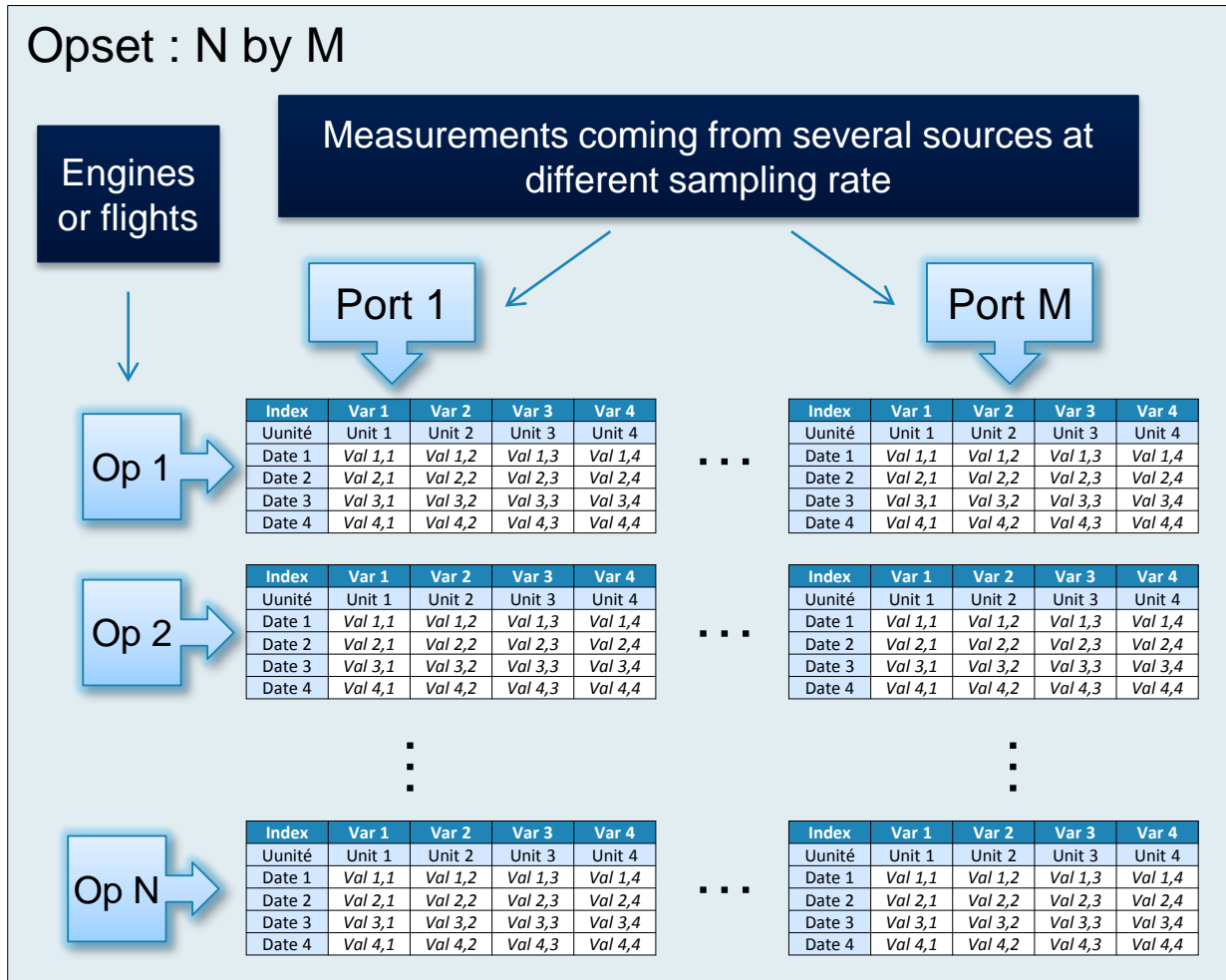




# SAMANTA objects – the opset

- **It is necessary to store the signals in MATLAB files: the SAMANTA object *opset* defines a list where to store the signals.**
  - Pointer to a backup file (.mat)
- **The *opset* is a sheet with n leaves, called operations**
  - Each operation comprises a series of signals
  - Signals must be similar from one operation to another
  - Easy access to all the information stored in the structure of the opset
- **The *opset* is serializable**
  - It is suitable for any method of execution of the algorithm

# SAMANTA objects – the opset

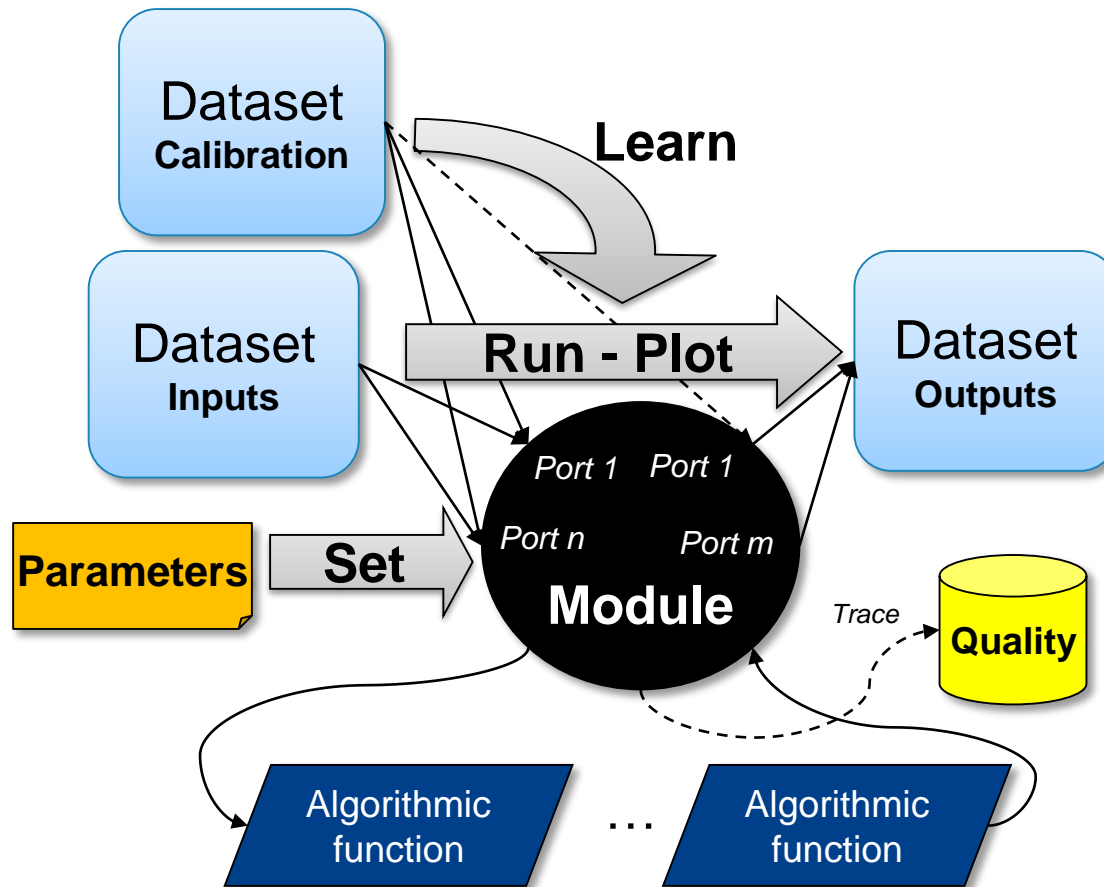


# SAMANTA objects – the *mnode*

- The SAMANTA *mnode* (short for "module node") is intended for data algorithmic processing.
- It allows the wrapping of any kind of algorithm
  - Uniform interface
  - Standard handling through a graphical interface
- The module operation is a sequence of several tasks:
  - Initialization, parameter validation, data accumulation, learning from accumulated data, processing performance on the inputs, display, ...
- The user also has different methods to use *mnode* including:
  - **Set** for parameterization
  - **Run** for execution
  - **Learn** for learning
  - **Plot** for display

# SAMANTA objects – the mnode

## → Summary of the *mnode* operation:



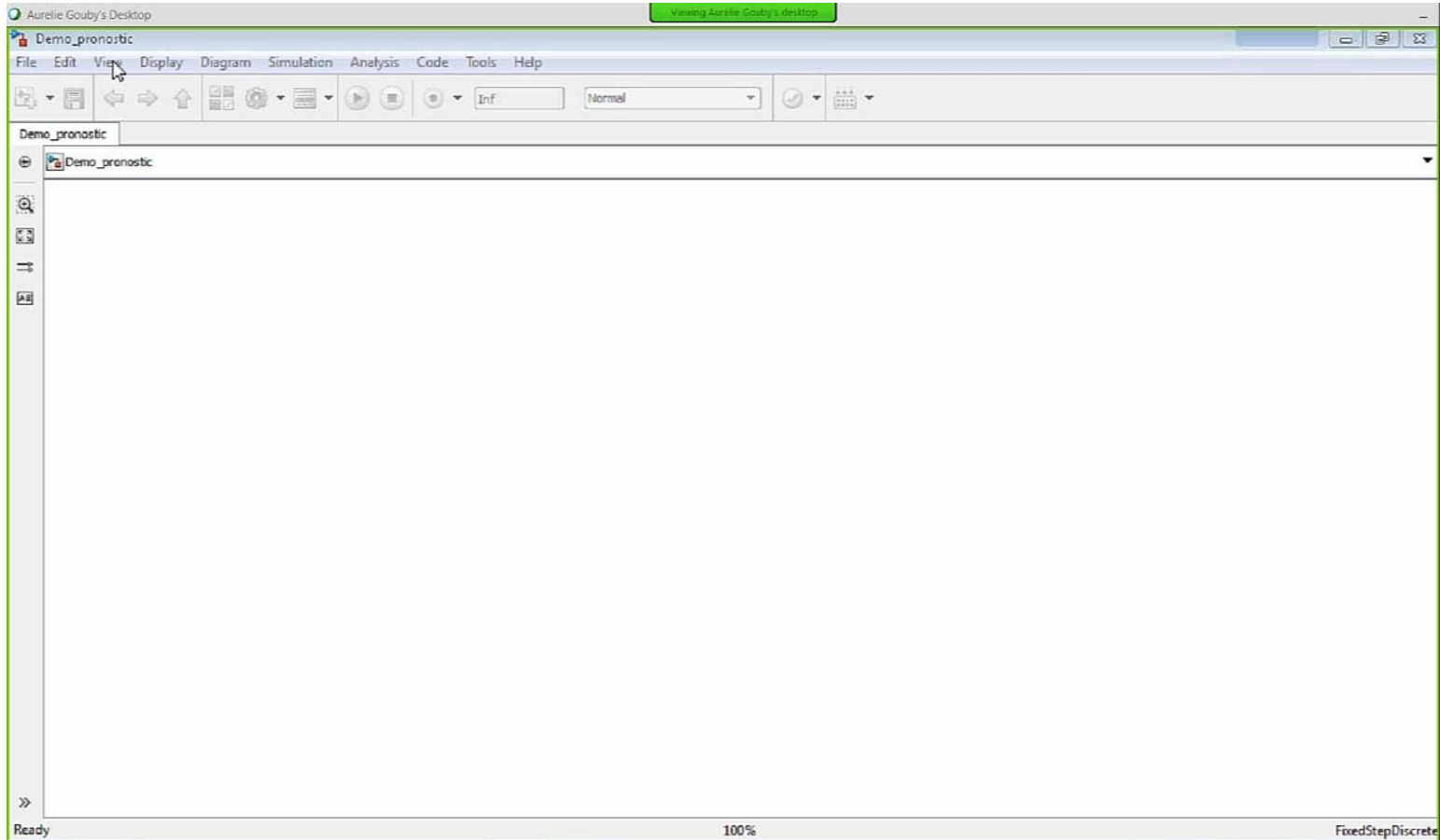
# /04/

## Utilizations

*With MATLAB or Simulink*

# SAMANTA utilization with Simulink

→ SAMANTA platform can be used via SIMULINK





# SAMANTA utilization with a MATLAB script

## → The SAMANTA platform can be used through MATLAB

- Create SAMANTA objects you want to use:
  - Treatment Module: *mnode*
  - Input: *opset*
- Set treatments:
  - Set modules
- Run treatments
  - Depending on the execution mode selected: run, learn
- Show results:
  - Via module: plot run mode
  - Via the generated data (*opset*)

## → In 'SCRIPT' mode, execution is a sequence of treatments, each one playing successively its entire *opset* input set and generates an output *opset*

# SAMANTA : methods used for the PHM

→ **The SAMANTA platform makes use of following MathWorks toolboxes:**

- Control System Toolbox
- Database Toolbox
- Fuzzy Logic Toolbox
- MATLAB Compiler
- Optimization Toolbox
- Robust Control Toolbox
- Signal Processing Toolbox
- Stateflow
- Statistics Toolbox
- System Identification Toolbox
- Wavelet Toolbox

**/05/**

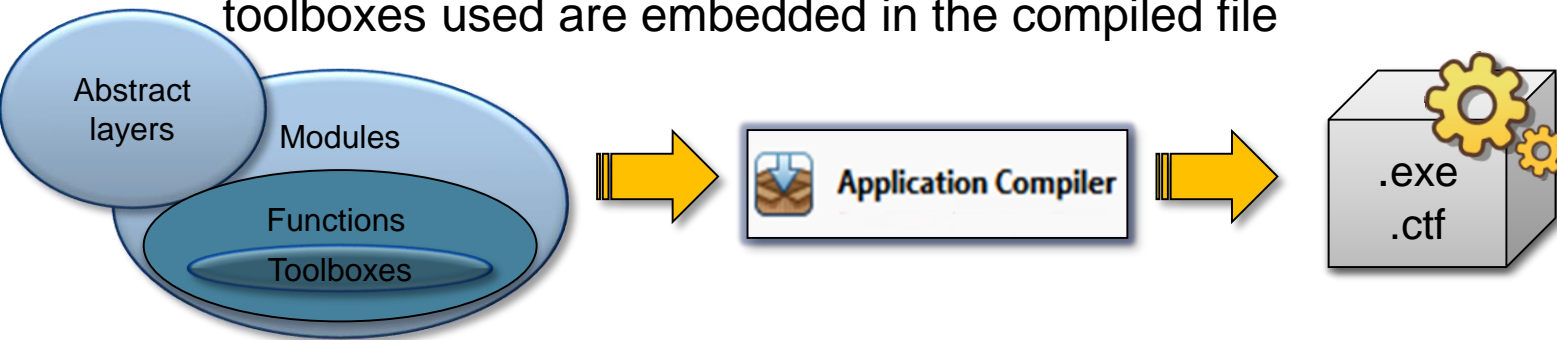
# Algorithms compilation and MPS usage

*Next steps*

# Algorithms compilation

## → SAMANTA algorithms can be compiled to be easily run in an environment without MATLAB.

- We use the MATLAB Compiler to create a compiled file from a SAMANTA application (.exe .ctf, .jar, ...)
- Abstract layers, codes of underlying modules and functions as well as toolboxes used are embedded in the compiled file

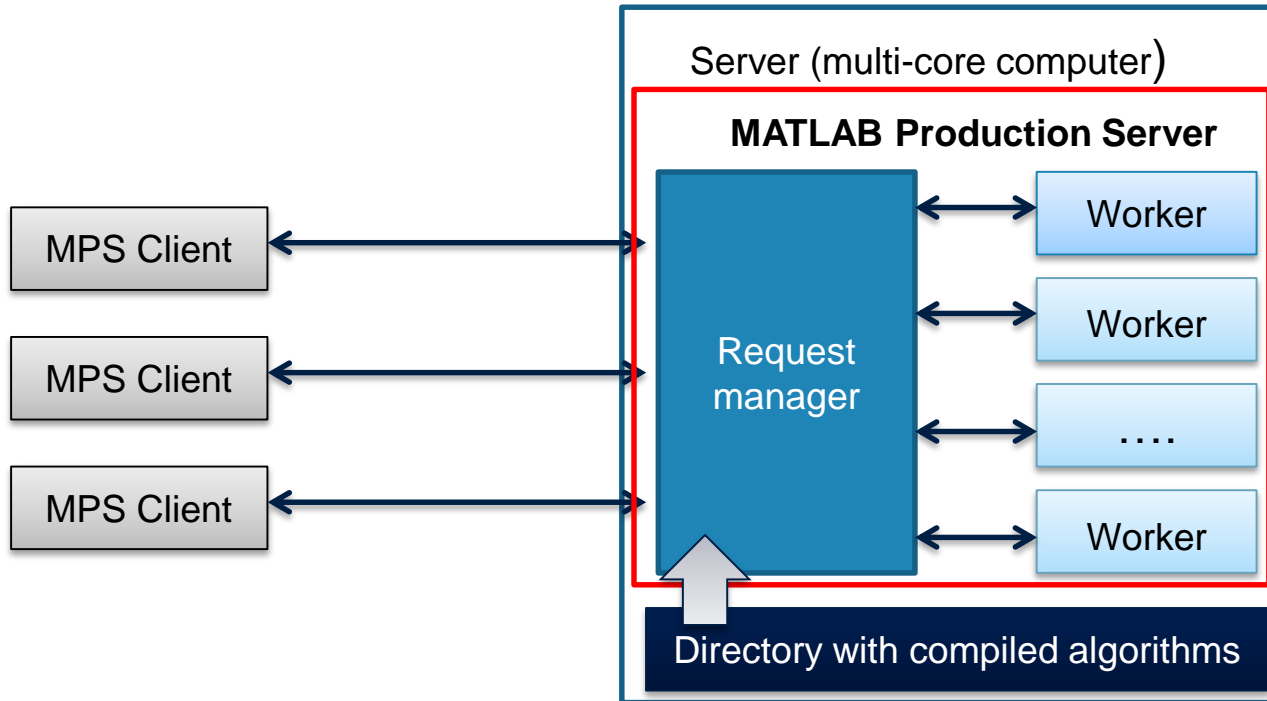


## → Once compiled, the SAMANTA algorithms can be used in the same way as in MATLAB thanks to the MCR (Matlab Component Runtime):

- Opset reading / processing/ writing
- Graphic displays specific to modules

# Use of compiled algorithms with the MPS

## → Example of using MPS to deploy SAMANTA algorithms



Manager

Worker

Directory

MPS Client

- Distributes client requests among the Workers depending on availability (load balancing)
- An instance of MCR that treats a client request at a time
- MPS readable directory where the compiled algorithms are deployed
- Client application to be developed with MPS client API available in Java or C #

**/05/**

# Conclusion



# Conclusion

- **The SAMANTA platform was created in 2007 and about 160 modules were designed since then**
  - Today about 15 engine monitoring algorithms have been developed, tested and matured through this platform and modules
  - The next step for Snecma is to compile these algorithms to be able to export them and use them in an operational environment thanks to the MPS
- **Thirty people are now using this platform in several companies of the SAFRAN Group**
  - Snecma, Turbomeca, Safran Engineering Services, Sagem,...
  - Among all regular users of the platform, only 1/3 have a computer science background

# Thanks for your attention