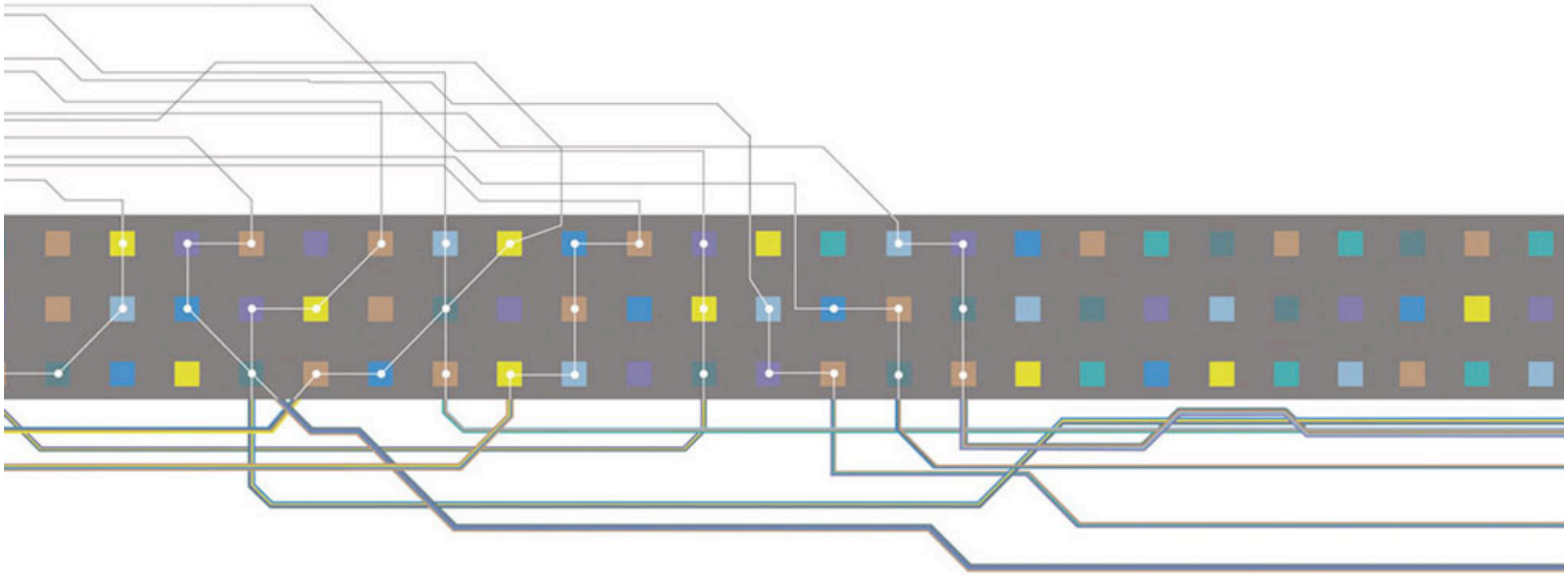
THALES



Really Hard Time Developing Hard Real Time

Etienne BORDE[1,2], Grégory HAÏK[1], Virginie WATINE[1], Laurent PAUTET[2]

(1) Thales Communications; (2) ENST

31/05/07, CAR'07, Paris

Software Engineering of RTE Systems

Development of ***Real Time Embedded (RTE) systems*** suffers from the same productivity problems than ***large-scale information systems***, such as:

- Platform heterogeneity
- Difficult testability
- Complex internal communication and interaction schemes
- Difficult configurability…

… plus many others !

- Timing issues
- Certification/assurance issues: safety-critical, mission-critical, security-critical
- Memory footprint
- Domain heterogeneity: telecommunications, vetronics, avionics, robotics

Software Engineering solutions for information systems must be ***adapted and extended*** to address RTE systems development

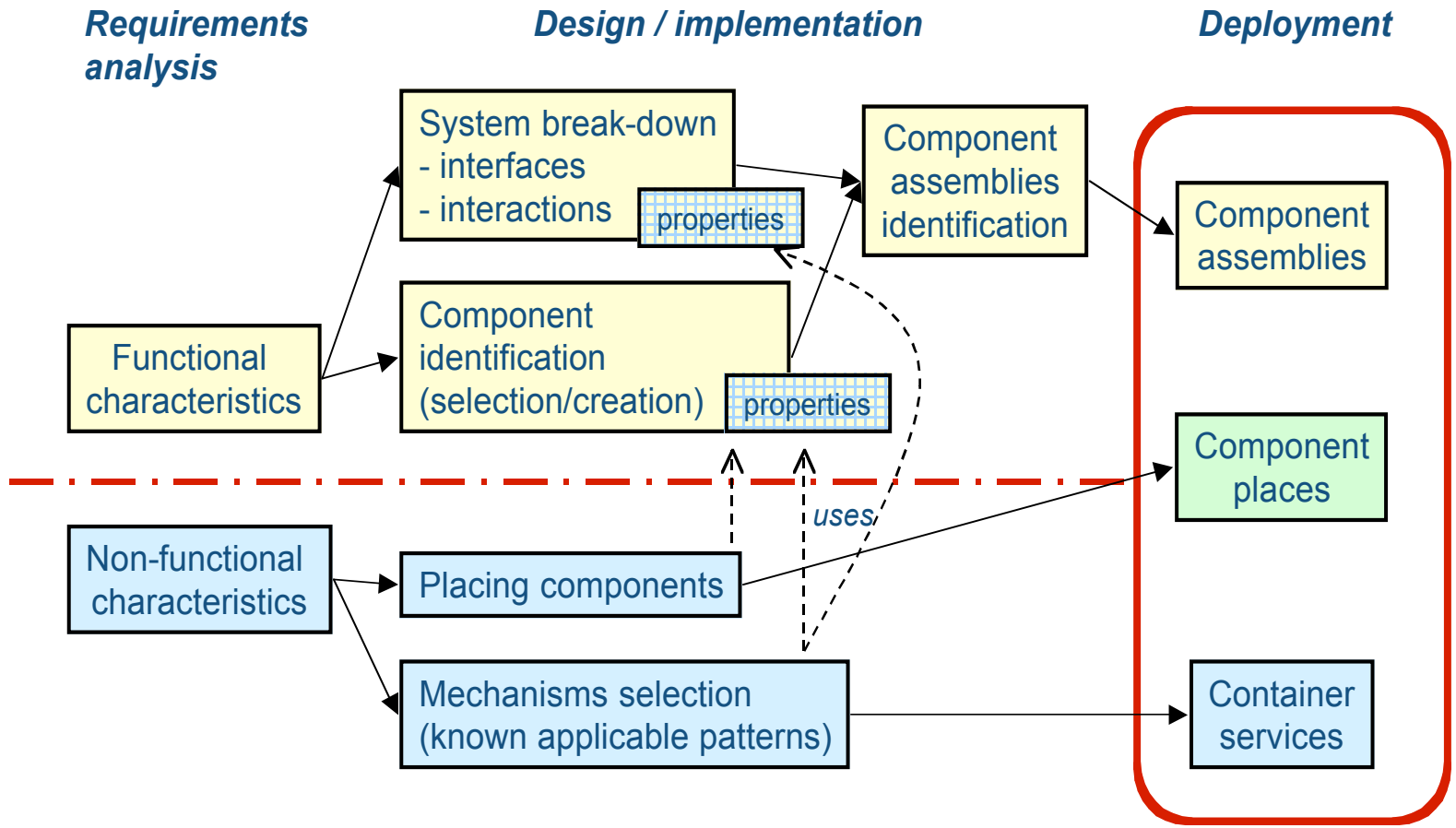
This document is the property of Thales Group and may not be copied or communicated without written consent of Thales

Adaptation of Component-based Development

- Traditional component-based methods (EJB, CCM) targets **Information Systems**:
  - Code reuse; Interoperability; Automatic deployment and configuration
- Non-functional needs of Information systems are
  - Communication support
  - Security
  - Persistency
  - Transactions
- Due to domain versatility of **RTE systems** and memory limitations, no such non-functional requirements list can be *a priori* devised
  - ***OMG Lightweight CCM specification*** defines empty component envelopes – no security, no persistency…
  - Up to the framework provider to ***tailor enveloppes to a particular domain***
  - Still, it requires a fine requirement analysis of domain – product line – application

This document is the property of Thales Group and may not be copied or communicated without written consent of Thales

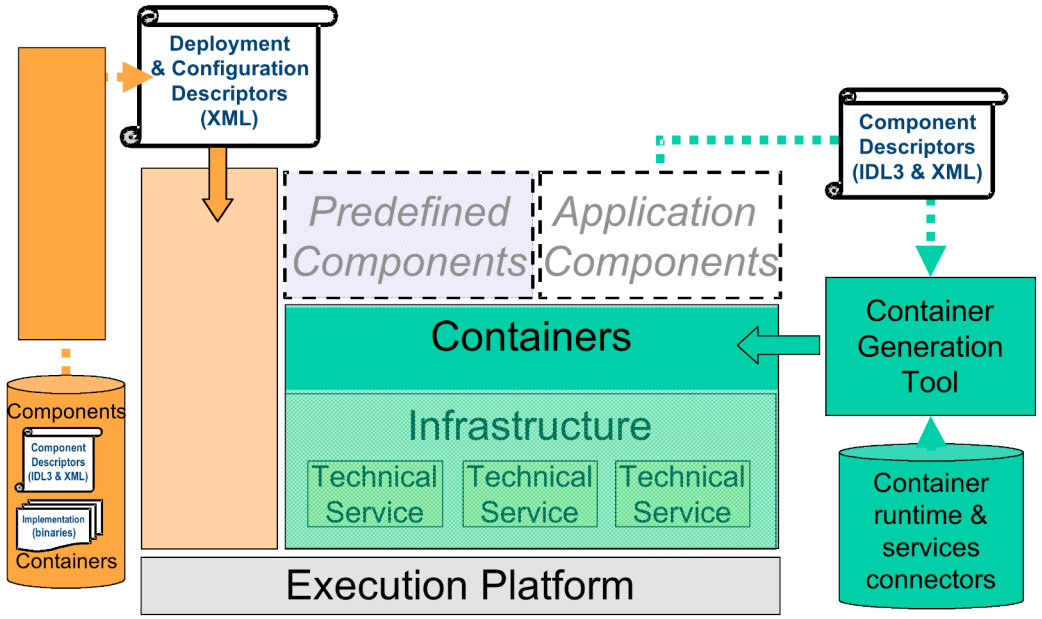
From Requirement Analysis to Runtime



This document is the property of Thales Group and may not be copied or communicated without written consent of Thales

Architecture of MyCCM Component Framework

Tailoring enveloppes while minimizing memory footprint calls for a ***modular architecture of the component framework*** itself.

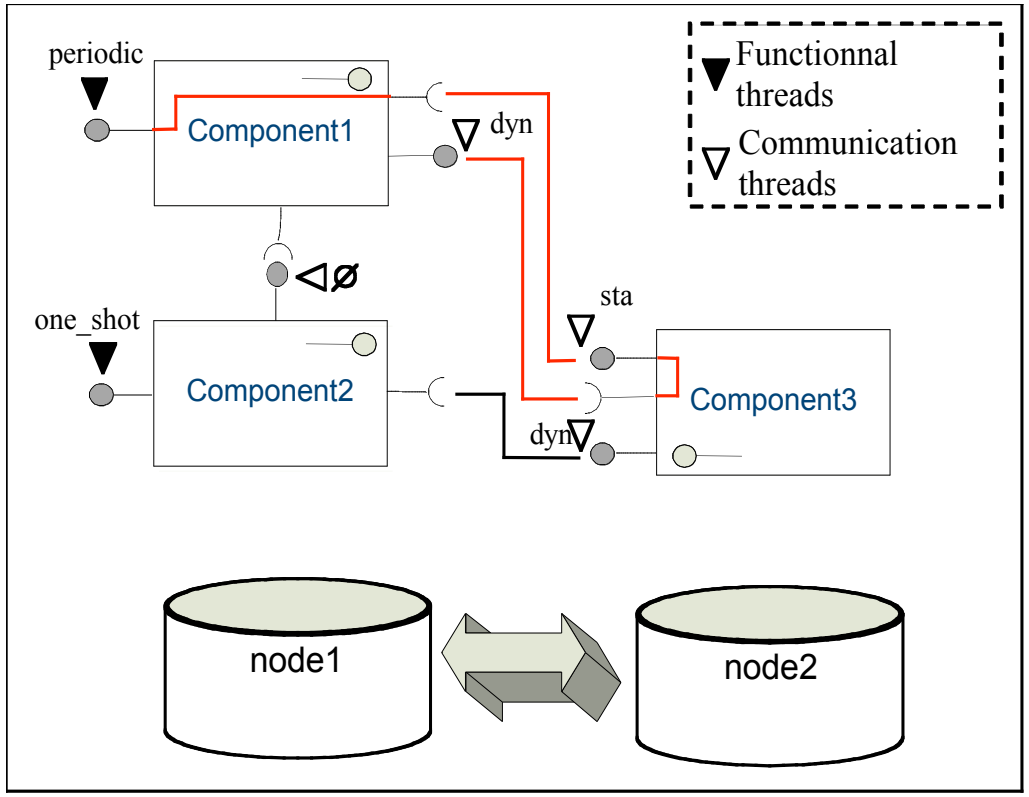


The extra benefits of this approach (beyond those of CBD) are :

- The ability to plug-in only what is strictly necessary
- The ability to adapt to various domains or product lines or even applications

This document is the property of Thales Group and may not be copied or communicated without written consent of Thales

MyCCM approach to Real Time



MyCCM enables the ***configuration of real-time scheduling parameters*** to:

- Define an activation model based on “periodic” and “one shot” “functional threads”
- Set the scheduling parameters of “communication threads” handling the component interaction mechanisms

This document is the property of Thales Group and may not be copied or communicated without written consent of Thales

MyCCM in the Scope of ARTEMIS Program

ARTEMIS is FREMM's IRST:

- Future European Multi-Role Frigates
- Infra-Red Sense and Track of the combat sub-system
- 3 Infra-red sensors, 1 multi-SBC processing unit
- Intensive computing
  - Track association algorithmic components
  - Visualisation algorithmic components

- ARTEMIS' MyCCM handles:
  - High throughput multicast communication (CCM event ports)
  - Application command and control (CCM facets/receptacles)
  - Model-driven (UML) configuration:
    - Deployment
    - Assembly
    - Real Time settings (threads, priorities, periods…)

This document is the property of Thales Group and may not be copied or communicated without written consent of Thales

Benefits for Software Architect

MyCCM improves software development ***productivity*** thanks to:

- Intensive code generation
  - Abstraction and generation of internal communication protocols
  - Generation of deployment code
  - Generation of threading artefacts
- Integration with modelling tools
  - Improving communication between team members
  - Facilitating verification
- Ease of testability
  - Potential functional validation on host platform
- Late binding to the target platform
  - Reduced integration risk

This document is the property of Thales Group and may not be copied or communicated without written consent of Thales

Limitations and Research Orientations

The MyCCM framework provides no support for:

- Proving the respect of end-to-end requirements
- Descriptive reconfiguration
- Multi-mission systems and late configuration

To tackle this issues, research efforts have been undertaken in the antagonistic fields of ***Verification and Validation*** and ***Flexibility***

This document is the property of Thales Group and may not be copied or communicated without written consent of Thales

Verification and Validation

To address ***hard real-time*** systems development, the framework must come with means to check that the deployed architecture will meet its timing requirements.

As a first step towards component-based architecture verification and validation, schedulability analysis should be performed:

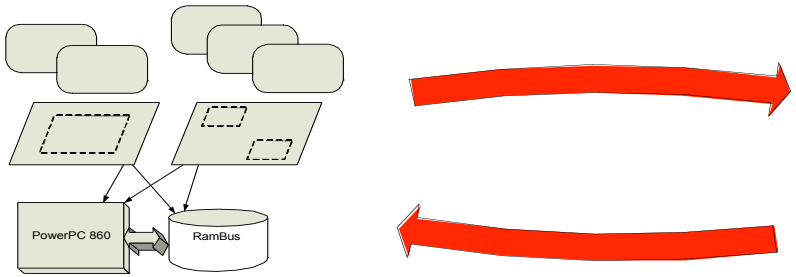
- This requires the framework user to provide a characterisation of the ***temporal properties of each component***.
- Combining this information with the activation model and corresponding communication threads, end-to-end ***execution times*** can finally be estimated.
- Transcribing this information in a tool like MAST, ***temporal analysis*** may be performed.

Many other requirements may be verified…

This document is the property of Thales Group and may not be copied or communicated without written consent of Thales

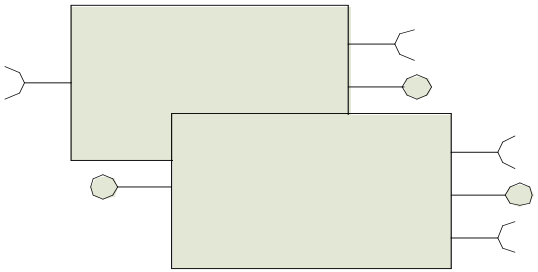
ITEA - SPICES Project Big Picture

AADL Modelling of Component-based Architecture



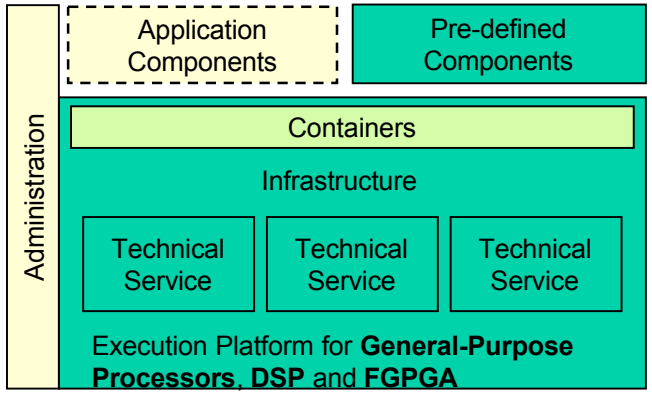
Lightweight CCM descriptors

Application Components Packages



Verification Techniques

- Schedulability, energy, memory analysis
  - MAST, Cheddar…
- Model Checking
  - BIP, TINA, CPNTool
- Simulation
  - SystemC, MyCCM, Ades
- Middleware code generation and verification
  - PolyORB-HI, Occarina



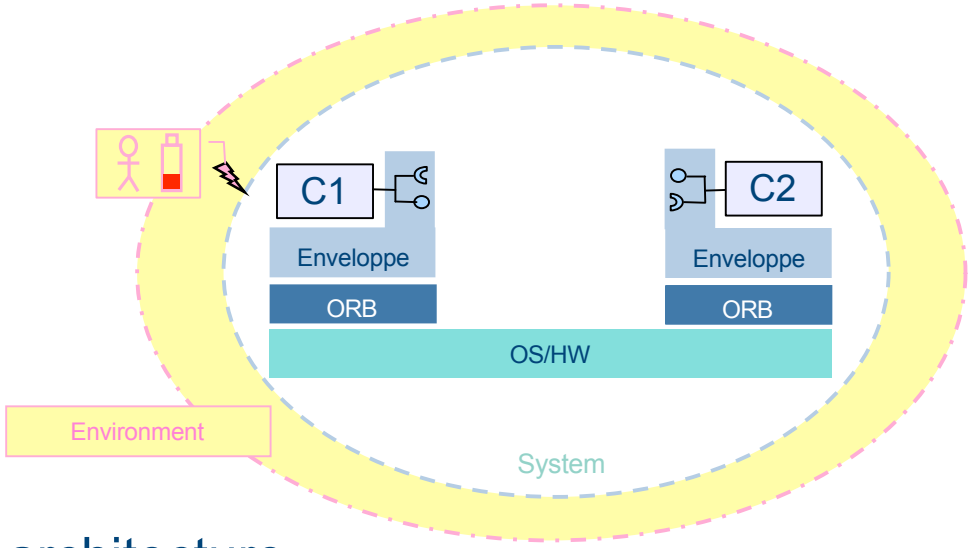
Running System

This document is the property of Thales Group and may not be copied or communicated without written consent of Thales

Flexibility: dynamic reconfiguration

On the other hand, as systems lifecycle are longer and longer, component based frameworks must adapt to various situations…

- Bug correction,
- Power limitations,
- User requests
- fault tolerance …



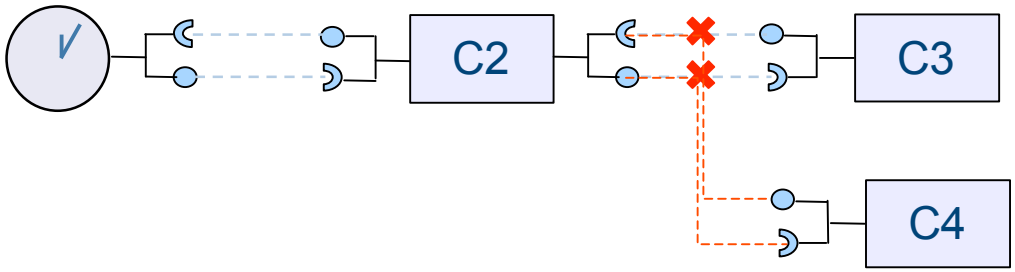
… that may modify the system architecture.

We call ***reconfiguration*** the process that drives system composition updates in response to internal or external events.

This document is the property of Thales Group and may not be copied or communicated without written consent of Thales

The adaptation process may thus involve modifications of the system composition:

- Functionality removal
- Component migrations, deletion, replacement…



To manage reconfiguration, ***a reconfiguration language*** may enable to generate the reconfiguration instructions.

Moreover, as requirements may be expressed onto the reconfiguration process, declarative reconfiguration is necessary to check the respect of these requirements.

These subjects are addressed in the scope of ***IST – FRESCOR*** (modification of scheduling parameters) and ***System@tic-INFLEXION*** (architecture modifications)

This document is the property of Thales Group and may not be copied or communicated without written consent of Thales

Flexibility: Multi-mission support

Multi-mission systems may be used in various application situations, with very little time available for reconfiguration.

For instance, an UAV usually used to monitor forest fires could be reconfigured in emergency for supervision of evacuation operations after an earthquake. Video camera may be changed during this reconfiguration.

In such cases, the goal is to ***help the system's end-user to configure it***. To do that, a tool based on semantics definition of components services should help to:

- ***Select adequate component*** services according to the semantic description of their goal and parameters.
- Assemble components by ***generating code that enable interaction*** of syntactically different but semantically compatible services.

These subjects are addressed in the scope of ***RNTL - Flex-eWare***

This document is the property of Thales Group and may not be copied or communicated without written consent of Thales

Concluding Remarks

To conclude, component based techniques are:

- Already applicable (FREMM)
- Easy to adapt
- Potentially applicable today to vetronics and robotics (OS/MW performances)

Special efforts still have to be done to address:

- hard-real time and critical systems,
- Multi-Mission and reconfigurable systems.

Of course, frameworks do not prevent the designer from working on the system predictability.

Still, it provides means to speed up the global development process (including V&V).

This document is the property of Thales Group and may not be copied or communicated without written consent of Thales