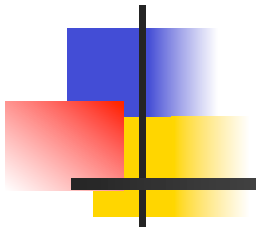
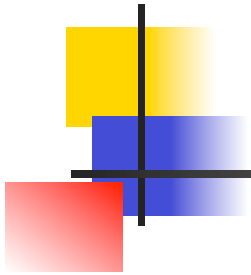


Integrated Modular Avionic

Laurent Pautet

Laurent.Pautet@enst.fr

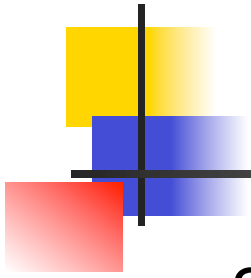
Version 1.1



Avionic systems

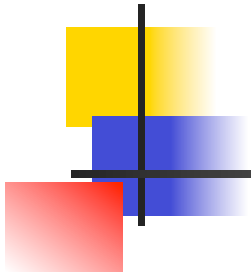
Functions enabling an aircraft (civilian or military) to perform its flight mission.

- Cabin
- Cockpit
- Navigation
- Energy
- Engines
- Flight control
- Communications



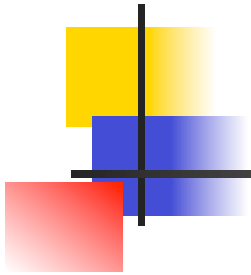
Cabine Functions

- Smoke Detection Function
- Fire Protection System
- Cabin Oxygen
- Crew Oxygen
- Cabin intercommunication data system
- Cabin Communication systems
- Cockpit Door Locking System
- Doors and Slide Control System
- In flight entertainment



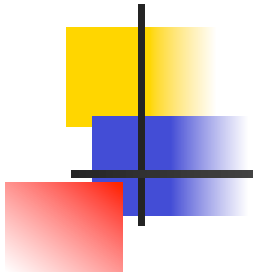
Cockpit Functions

- External And Taxiing Camera System
- Audio Control
- Flight Warnings System
- Control and Display System
- Electronic Centralized Aircraft Monitoring
- Head-Up Display
- Concentrator and Multiplexer for Video
- Digital Flight Data Recording System
- Tail Strike Indication System



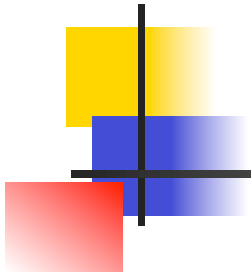
Energy Functions

- Electrical Load Management
- AC and DC Generation Control System
- Primary and Secondary Power Distribution Management
- Emergency Power Generation & Distribution
- Windows Heat Controller
- Exterior and Internal Lights (cockpit and cabin)
- Auxiliary Power Unit
- Circuit Breaker Monitoring
- Ice Detection
- Engine Control system



Flight Control Functions

- Flight Management
- Flight Envelope
- Automatic Flight Guidance
- Weight and Balance Back-Up Computation
- Flight Controls unit
- Flight Control Data Concentrator

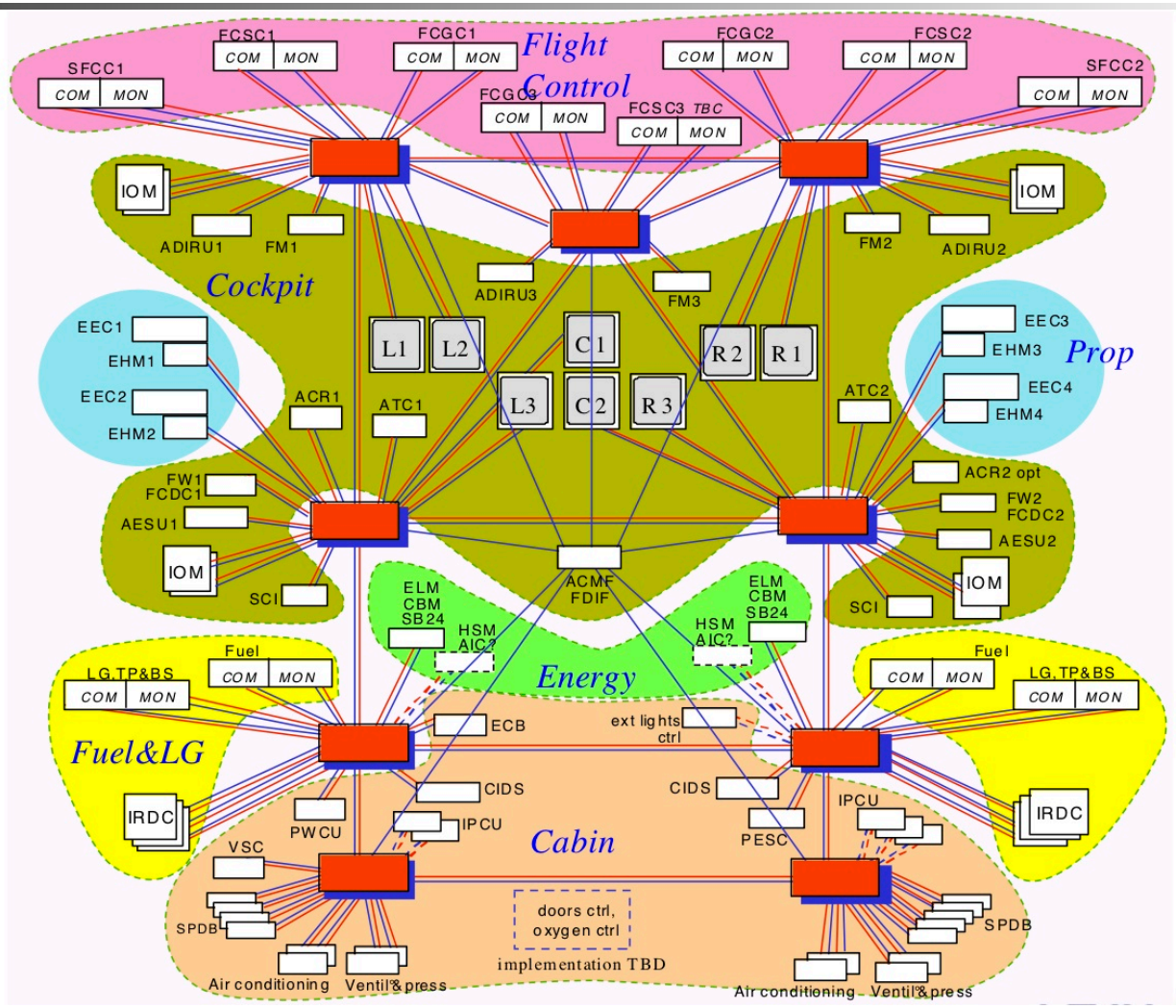


Navigation Functions

- Automatic Direction Finder
- VHF Omni directional Range (VOR)
- Distance Measuring Equipment
- Air Data Reference
- Multi Mode Receiver
- Onboard Airport Navigation System
- Radio Altimeter
- Weather Radar
- Traffic Collision Avoidance System
- Traffic Awareness and Warning System

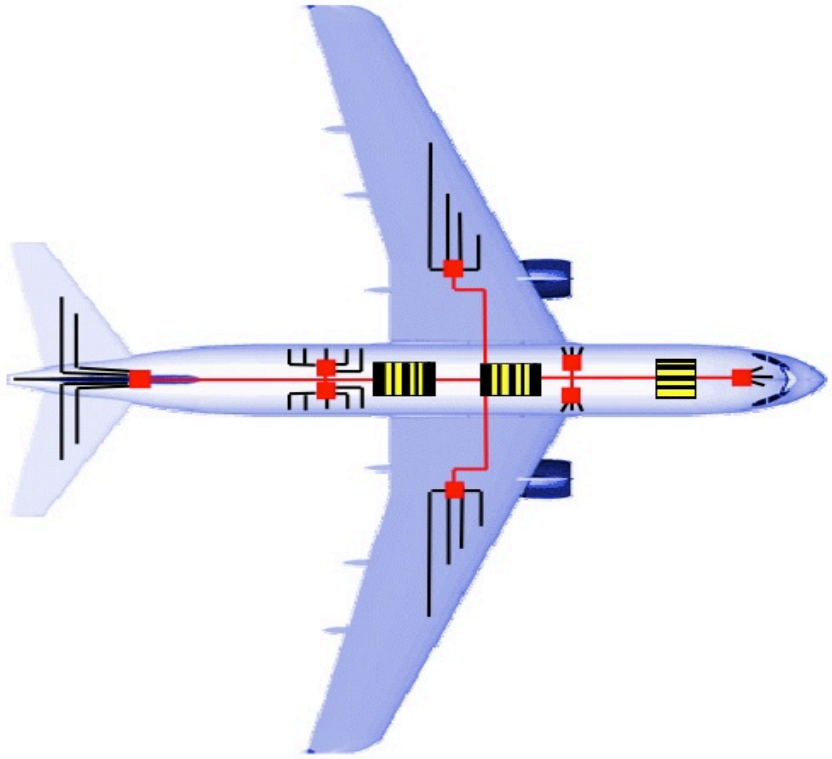
Avionic Architectures

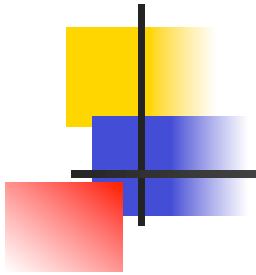
Avionics Data Communication Network



Federated Architecture

- Line Replaceable Unit (LRU)
  - a function,
  - software, hardware,
  - confinement,
  - a supplier
- Dedicated to a given aircraft
- Assembly of the different LRUs through a network of cables
- Actors and Sensors near the computer
- +100 km of cables
- 20-30 calculators



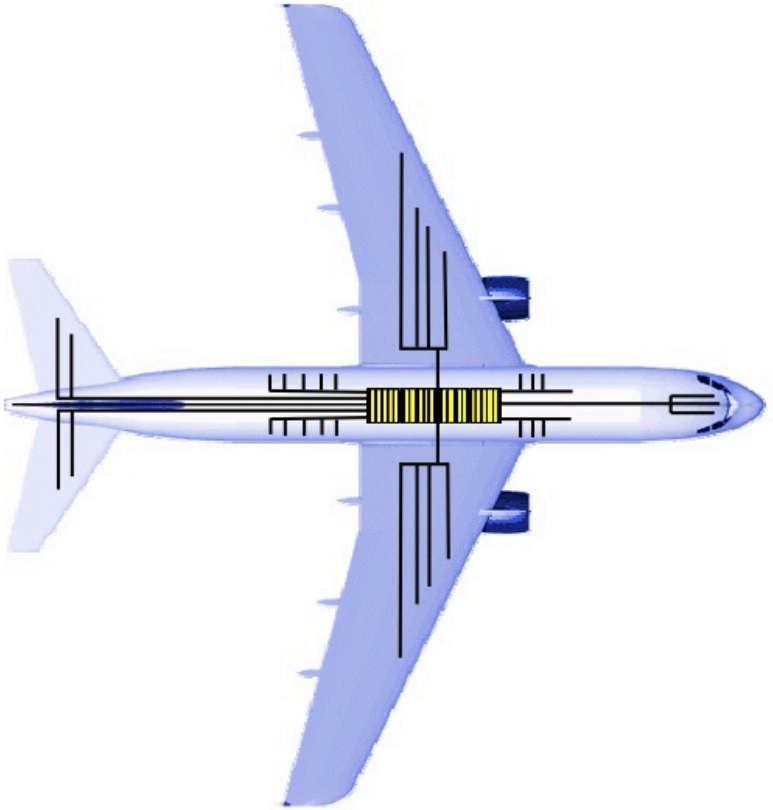


Objectives
Integrated Modular Architecture

- Reduce the impact of equipment
  - When designing the software and executing on the platform
- Standardize equipment, reduce costs
  - For consumer equipment use
- Reduce dependence on a supplier
- Improve portability and modularity
- Increase the number of functions
  - During the 10 years of aircraft design, needs change
- Reduce weight, volume and energy
- Reduce design and certification costs
- Reduce costs of maintenance

Integrated Modular Architecture

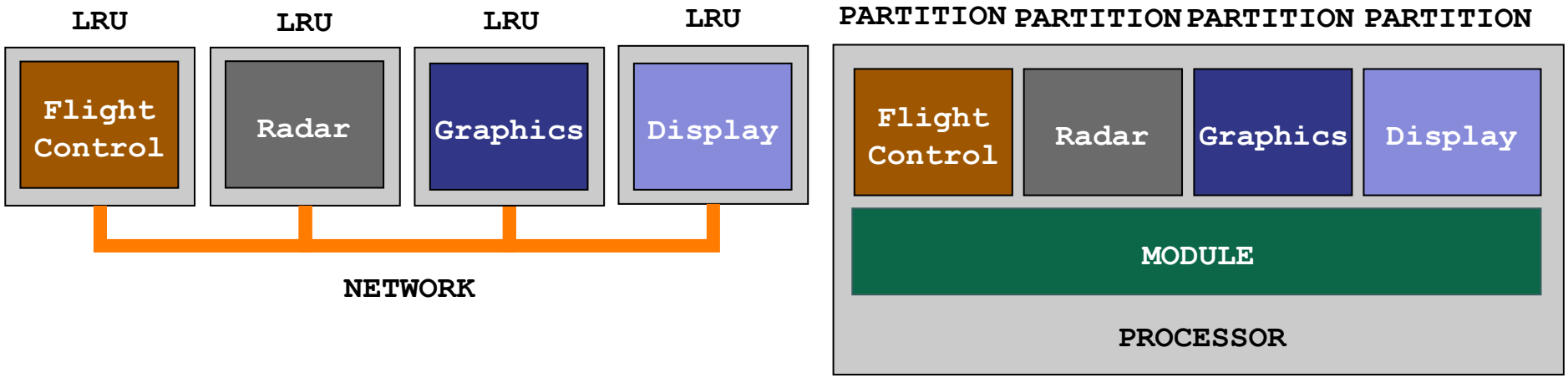
- Several functions, one calculator
- A provider designs the function
- An integrator allocates resources to the supplier for this function
- 6 to 8 non-dedicated calculators
- Reduce weight, volume and energy
- Ease new functions addition



Classical Architectures

- Federated Architecture
- Unit: LRU
- Integration: network

- Integrated Architecture
- Unit: partition
- Integration: module



Federated vs Integrated

Federated Architecture

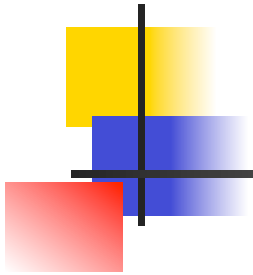
- One function, one material
- Well-established methodology
- Fairly easy design
- Fairly easy certification

- High volume / weight / energy
- Materials and cables
- Limited bandwidth
- 30-40 functions max per bus
- Low reuse / portability
- Tied to suppliers

Integrated Architecture

- Several functions, one material
- Lower volume / weight / energy
- High software and system reuse
- Strong portability
- Easy addition of functions

- Less established methodology
  - Functions communicating strongly on the same module
- More complex integration
- More complex certification

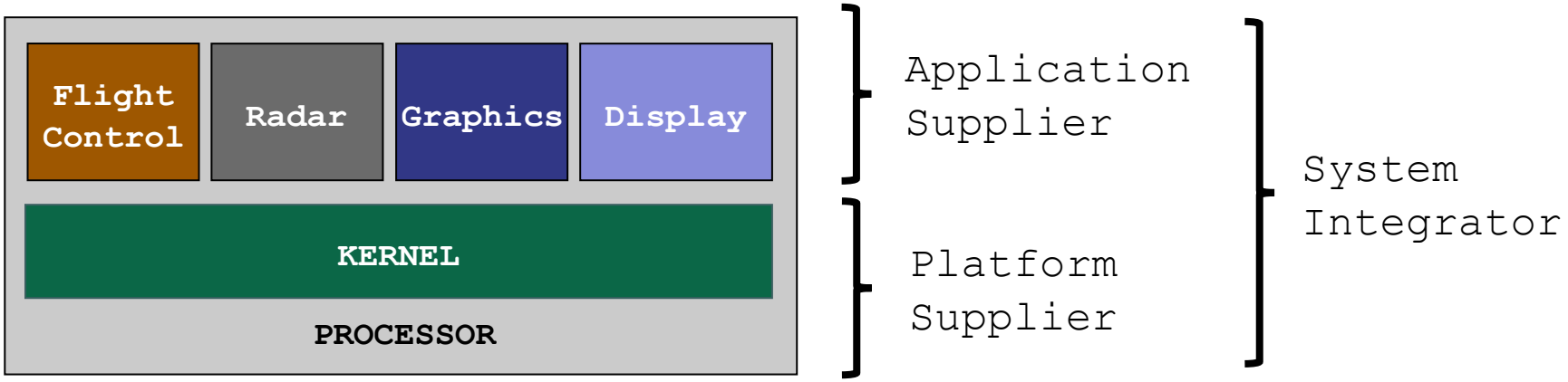


Design Process

- An avionics system must provide good reliability and therefore meet a set of requirements
- A certification agency ensures compliance with standards guaranteeing compliance with requirements such as FAA (USA) or EASA (EU)
- RTCA produces standards for certification
  - DO-297 for development cycle management
  - DO-178 for software
  - DO-254 for materials
  - DO-278 for air traffic management

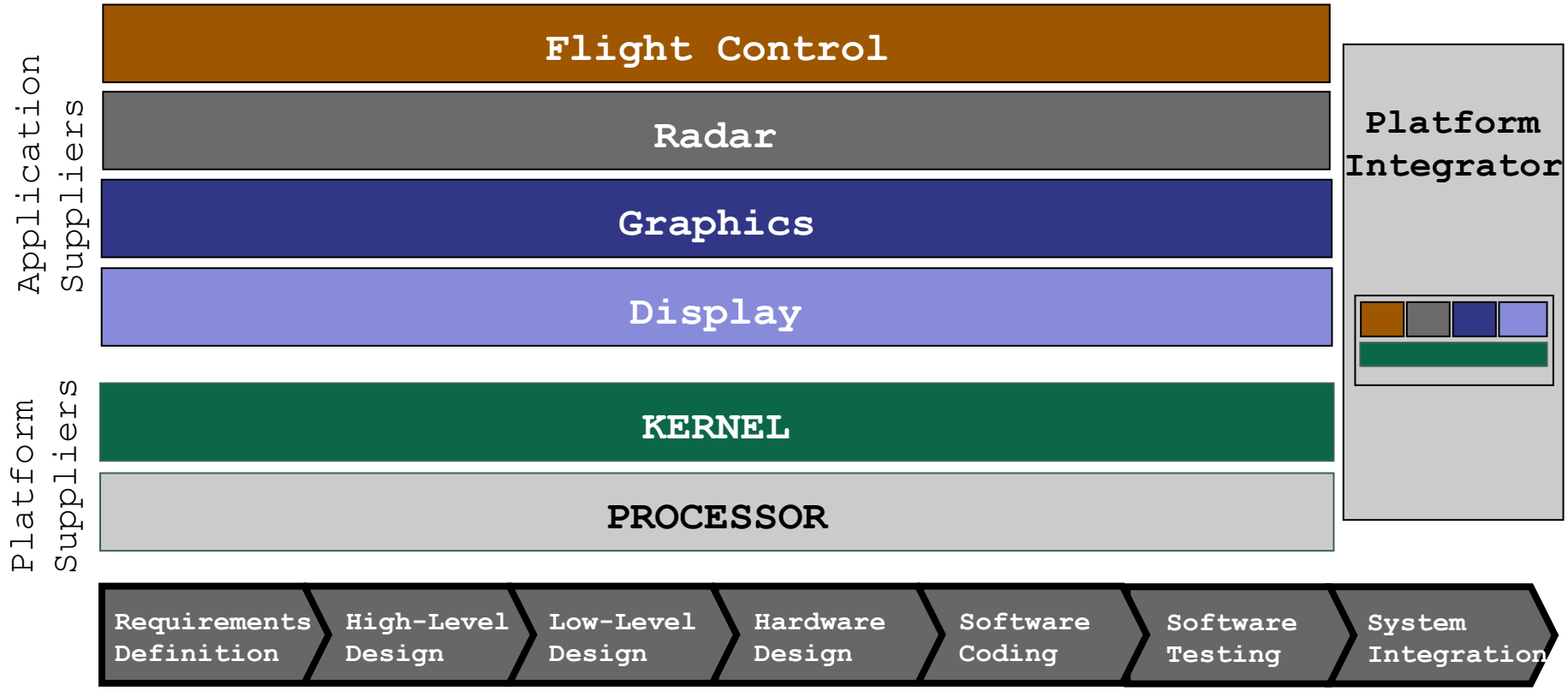
Standard DO-297 (1/2)

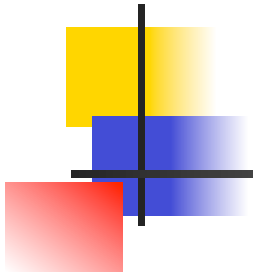
- Development organized around 3 roles
  - Platform Supplier (hardware + basic software such as kernel)
  - Application Supplier (software function)
  - System Integrator



Standard DO-297 (2/2)

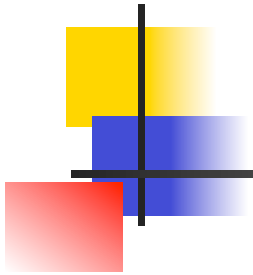
- Parallel and independent design and certification





DO-178

- DO-178 proposes rules to ensure the reliability of the software (functions, kernel, integration, etc.)
- A function is assigned a criticality level according to the severity of its failure
- The level of criticality determines the acceptable probability of occurrence of faults (in number per hour)
- It determines the development rules to be applied according to the level of criticality
- These rules apply to all development (planning, requirement, design, coding, testing, etc.)
- How to certify the code without formal verification?

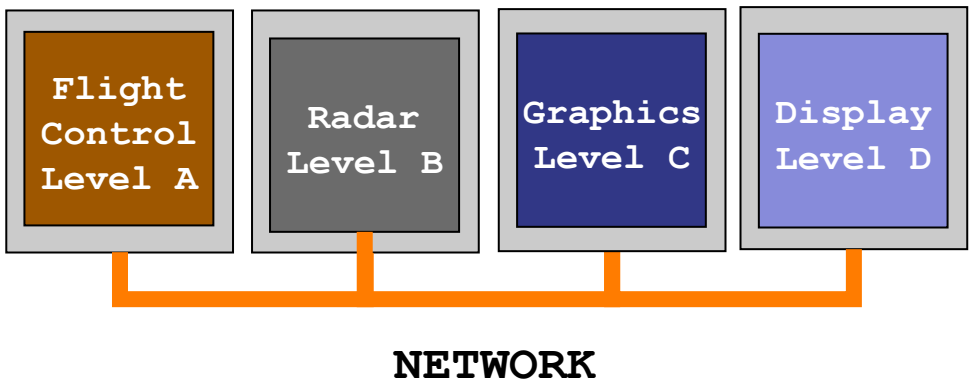


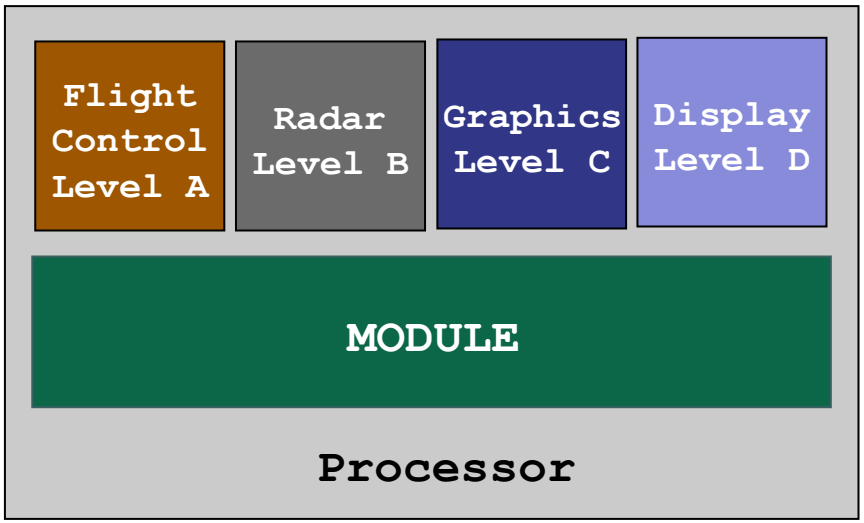
Criticality Level

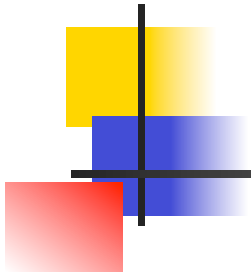
Criticality level	Specific rules	Volume of functions	Consequence	Max # of occurrences
E	0	5%	None	
D	28	10%	Minor	10^{-3}/h
C	57	20%	Major	10^{-5}/h
B	65	30%	Hazardous	10^{-7}/h
A	66	35%	Catastrophic	10^{-9}/h

Criticality and Architecture

- Federated Architecture
- Integrated Architecture
  - Different criticality levels on the same computer

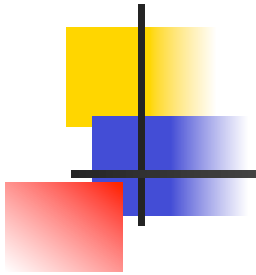






Objectives

- Ensure error containment whether the architecture is federated or integrated
- Ensure that a given criticality function does not disturb a higher criticality function
- Therefore, in the case of integrated architecture
  - Isolate functions spatially (memory) and temporally (CPU)
  - Prohibit a given criticality function from transmitting to a higher criticality function (same computer)

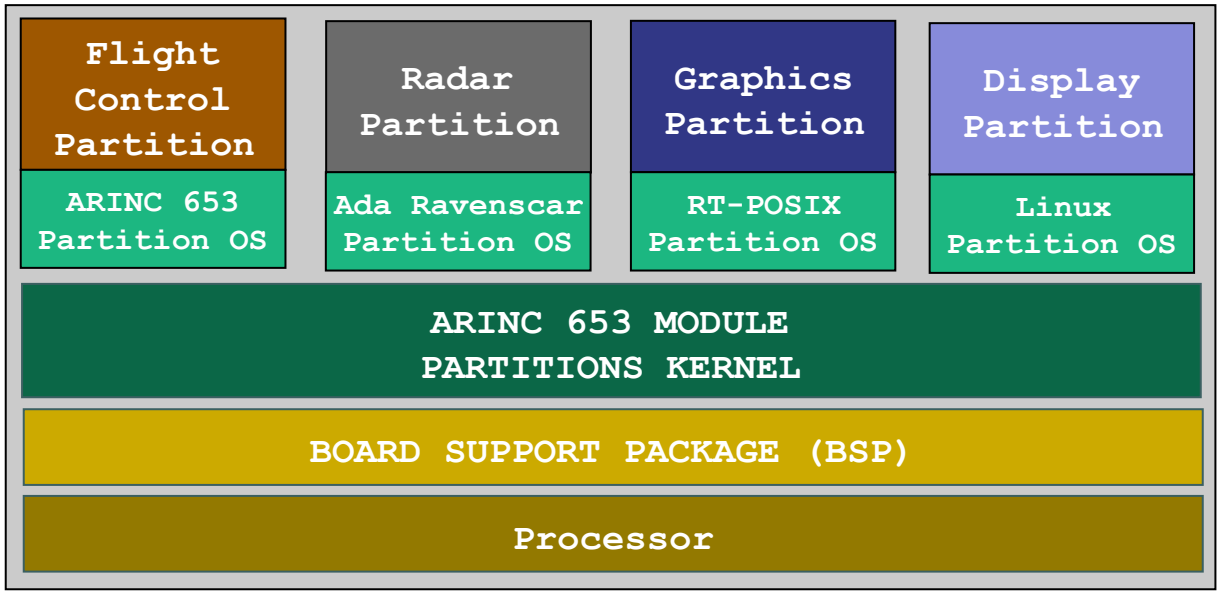


ARINC 653

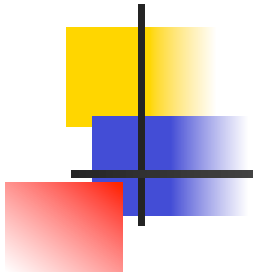
- ARINC 653 provides a specification to design an integrated architecture on top of a core on a processor
- The ARINC 653 kernel is certified so that if the functions are certified (independently), the whole becomes certified
- The ARINC 653 kernel must ensure spatial and temporal isolation and guarantee criticality constraints during communications
- APEX, API of ARINC 653, provides 7 services: Partition, Process, Time, Memory, Inter and Intra Partition Communication, Health Monitor
- The ARINC 653 kernel is hierarchical, a first level kernel executing partitions, each including a second level kernel executing processes
- ARINC 653 hides hardware specificities and dependencies

ARINC 653 – APEX
Partition

- Spatial and temporal isolation is ensured by preallocating :
  - Fixed-size time slots whose kernel prevents any overflow
  - Fixed-size memory areas protected by MMU mechanism
- A kernel within a partition can provide multitasking
- An XML file allows to configure these services at startup



ARINC 653 – APEX
Temporal Isolation



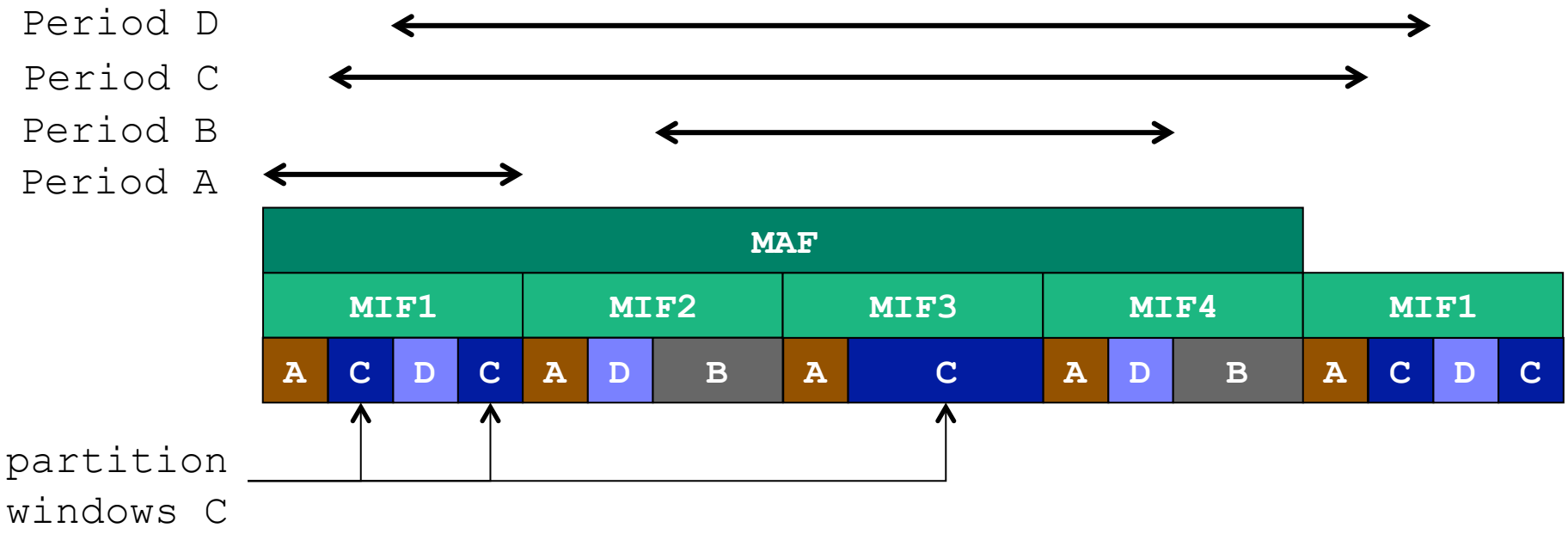
- Time is divided into periodic MAjor Frame (MAF)
  - Often the PPCM of periods of harmonic partitions
- A MAF is divided into several MInor Frames (MIF)
  - Often the GCD of periods of harmonic partitions
- Over its period each partition is broken down into several time slices called Windows Partition
- Each MIF consists of Partition Windows of multiple partitions
- The integrator assigns Partition Windows so that each partition fulfills its deadline
- The kernel checks that each partition does not temporally overflow the allocated Partition Window

ARINC 653 – APEX

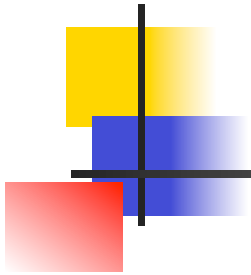
Temporal isolation

Partition	A	B	C	D
Period	10ms	20ms	40ms	40ms

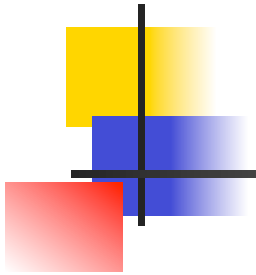
MAF	MIF
40ms	10ms



ARINC 653 – APEX
Spatial isolation



- Each partition has a memory area protected by the kernel when the partition is not active
- The kernel uses the mechanisms provided by the Memory Management Unit available in the processor
- An active partition therefore cannot write to the memory areas of other partitions.
- Memory areas for inter-partition communications (shared by two partitions) are also protected by the kernel

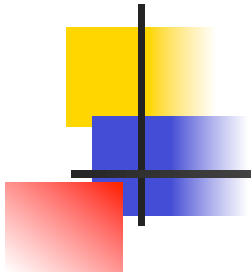


ARINC 653 – APEX

Process and Time

- Similar to a POSIX thread
- Runs in a partition (at least one process)
- Has attributes such as priority, period, capacity ...
- Respects preemptive, fixed priority scheduling

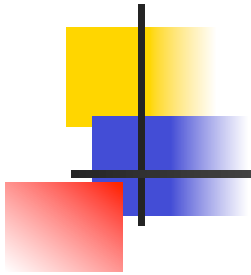
- An initialization process starts the partition
- A process can wait for a given time
- A process can wait until its next activation
- A process can get the current time



ARINC 653 – APEX Semaphore and Event

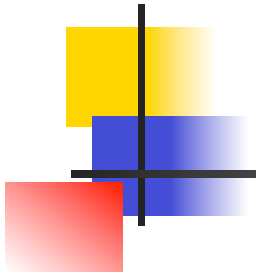
- Two mechanisms are available to synchronize processes of the same partition :
- Semaphores provide the classic semaphore mechanism. PIP and PCP policies are available.
- Events allow to wait for an event to be true and block otherwise.

ARINC 653 – APEX Communication intra-partition



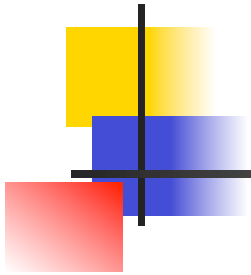
- Two mechanisms are available to communicate between processes of the same partition:
- Blackboard allows to overwrite the previous value of a data with a new value and read it as many times as necessary. It has an initial value.
- Buffer also allows to write several values of data but does not overwrite them and keeps them in a memory area in either FIFO or priority order. It also allows to read them by blocking if no value is available.

ARINC 653 – APEX
Communication inter-partition

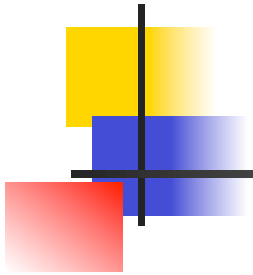


- Two mechanisms are available for communicating between partitions of the same computer. These mechanics are similar to Blackboard and Buffer.
- Sampling port allows to overwrite the previous value of data with a new value and read it as many times as necessary. It has an initial value.
- Queuing port also allows to write several values of a data but does not overwrite them and keeps them in a memory area in either FIFO or priority order. It also allows to read them by blocking if no value is available.

ARINC 653 – APEX Health Monitor

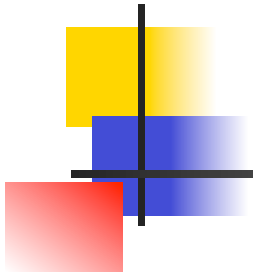


- Health Monitoring deals hierarchically with errors:
  - Identifies and reports errors
  - Associates processing with errors
  - At the process level for the Supplier Application
  - Partition level for System Integrator
  - At the module level for the Platform Supplier
- The system must guarantee that an error raised at one level is processed at this level or one
- Treatment may involve a restart
  - Cold restart (code and data are reallocated and reset)
  - Warm restart (reset from a previous or replicated context)



ARINC 664
AFDX – network for avionic systems

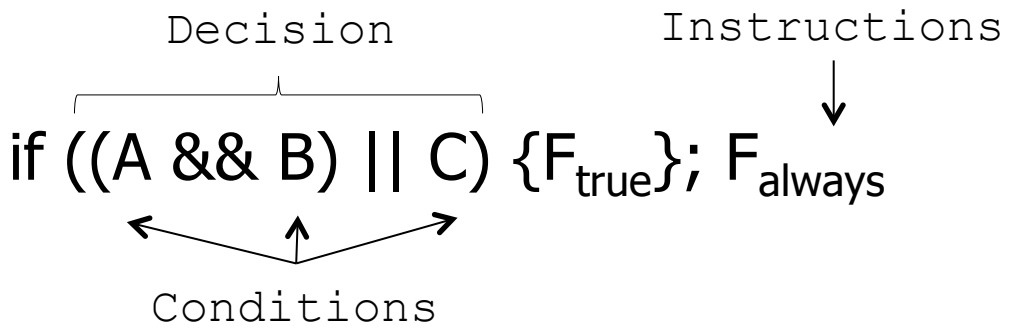
- ARINC 664 defines deterministic means of communication (somehow ARINC 653 for the network)
- AFDX (Avionic Full DupleX) is based on
  - Traditional (often redundant) equipment
  - An Ethernet-type switched network
  - Eenriched with bandwidth reservation
- Virtual links control transmissions and receptions in order to avoid collisions and re-transmissions
- AFDX is based on known standards (Ethernet) and takes advantage of the statically known system specifications to guarantee bounds on latencies



Code Coverage

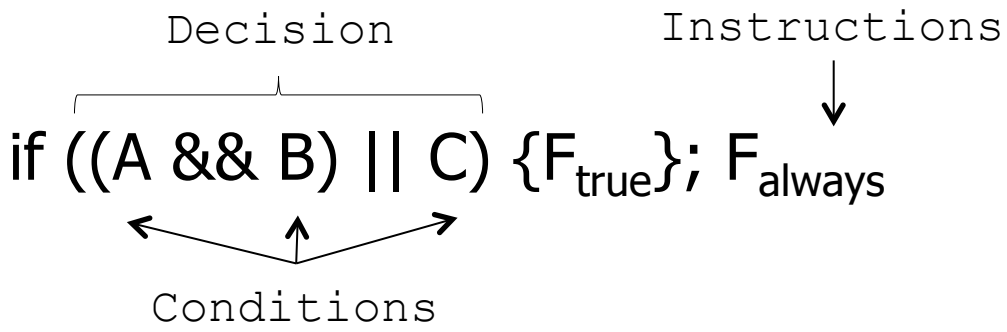
- DO-178 requires that enough tests be applied to check all chuncks of code: this is code coverage
- Code coverage can investigate code more or less deep
- There are 3 kinds of coverage requirement
  - Modified Condition / Decision Coverage for level A
  - Decision Coverage for level B
  - Statement Coverage for Level C
  - Levels D and E are not affected

Code Coverage
Statement Coverage – Decision Coverage



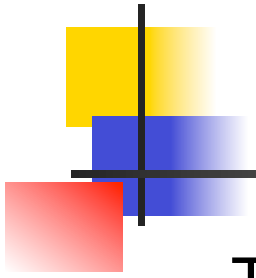
- Statement Coverage: execute all the instructions in the code at least once
  - Execute F_{true}+ F_{always} ; 1 test with the decision being true
- Decision Coverage: Execute at least once in each decision branch
  - Execute F_{true}+ F_{always} and F_{always} alone; 2 tests with the decision being true and false

Code Coverage
Modified Condition / Decision Coverage



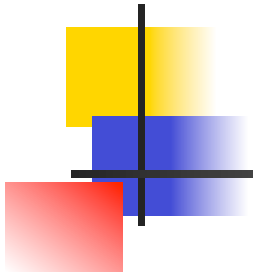
- MC/DC: Decision Coverage + each condition independently influences the outcome of the decision
- For N conditions, at least N + 1 tests are required. Here the results of test1 and test2 change, while A varies and B and C remain constant

	A	B	C	D
test1	T	T	F	T
test2	F	T	F	F
test3	F	T	T	T
test4	T	F	F	F



Code Coverage
Criticality Level

- The level of criticality implies the degree of coverage
  - Level A: Modified Condition / Decision Coverage
  - Level B: Decision Coverage
  - Level C: Statement Coverage
- Source code coverage is not equivalent to object code coverage
  - As we do not check the independence of the conditions, the object code coverage is similar to Decision Coverage



Conclusions

- There is nothing to prevent the application of IMA to railways, automobiles, software radio, etc.
- These industries claimed that the avionics approach is not transferable elsewhere
  - Small number of very expensive planes
  - Large number of functions of an aircraft
  - Critical requirements of an airplane (no stop button)
- The differences tend to fade so that other industries are interested in the AMI
- Railway (CENELEC 50128), automotive (ISO26262) or space (ECSS-E40A) standards tend to get close to AMI