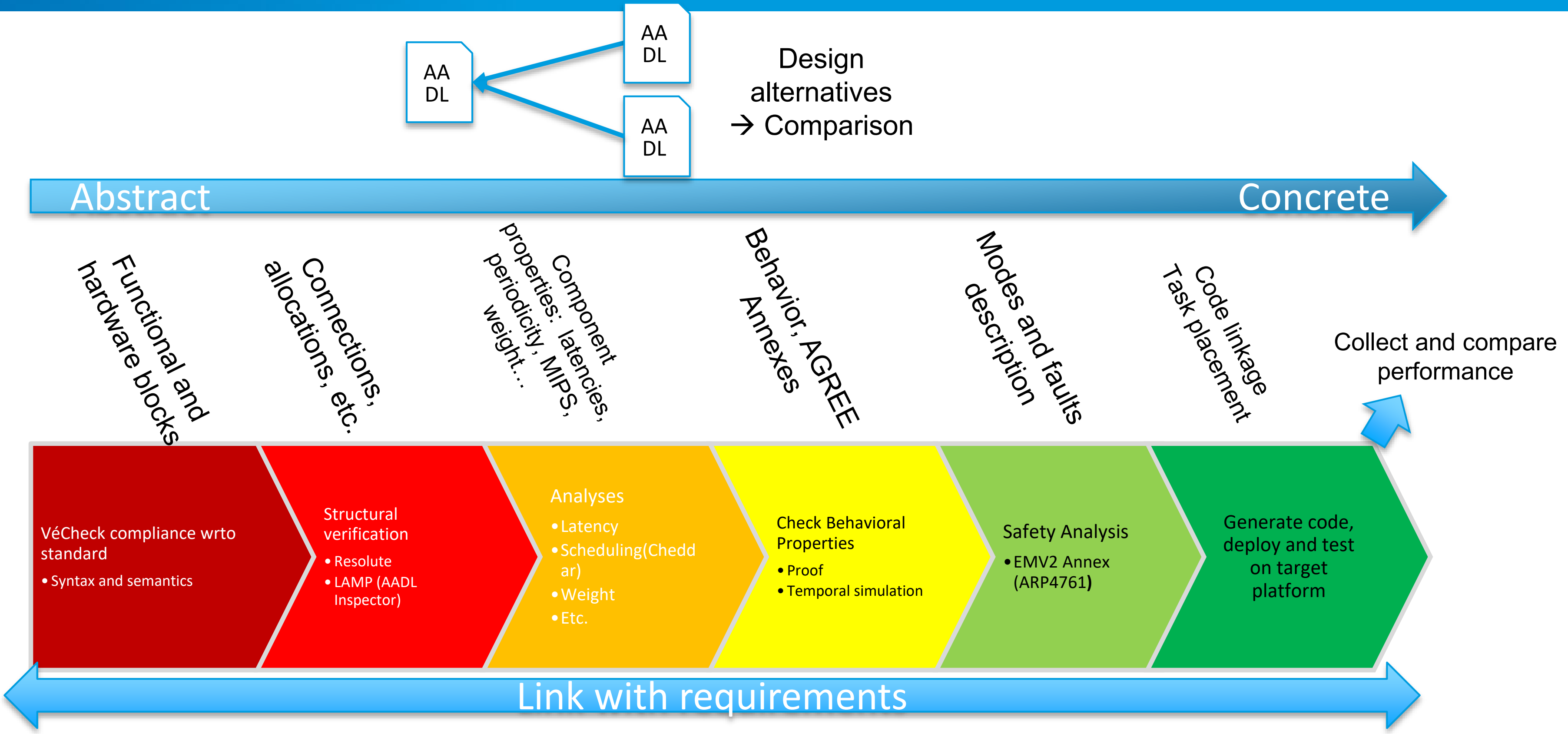
Accélérateur
de la transformation
numérique

PST Project
Continuous Virtual Integration with ALISA

27/06/2019

Design and Implementation Process with AADL


AADL
AADL
AADL
Design alternatives → Comparison
Abstract
Concrete
Functional and hardware blocks
Connections, allocations, etc.
Component properties: latencies, periodicity, MIPS, weight…
Behavior, AGREE Annexes
Modes and faults description
Code linkage Task placement
Collect and compare performance
Link with requirements

VéCheck compliance wrto standard

- Syntax and semantics

Structural verification

- Resolute
- LAMP (AADL Inspector)

Analyses

- Latency
- Scheduling(Cheddar)
- Weight
- Etc.

Check Behavioral Properties

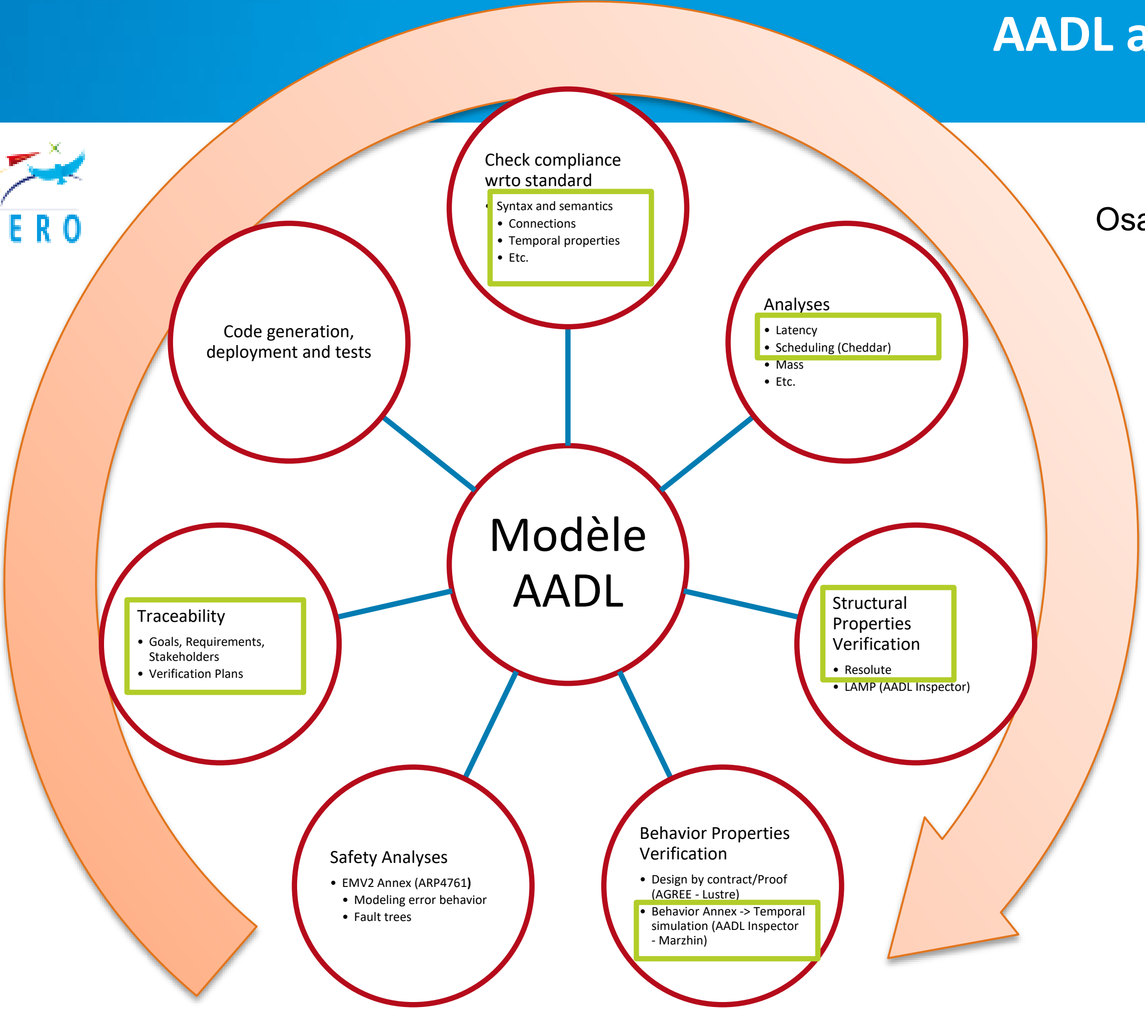
- Proof
- Temporal simulation

Safety Analysis

- EMV2 Annex (ARP4761)

Generate code, deploy and test on target platform

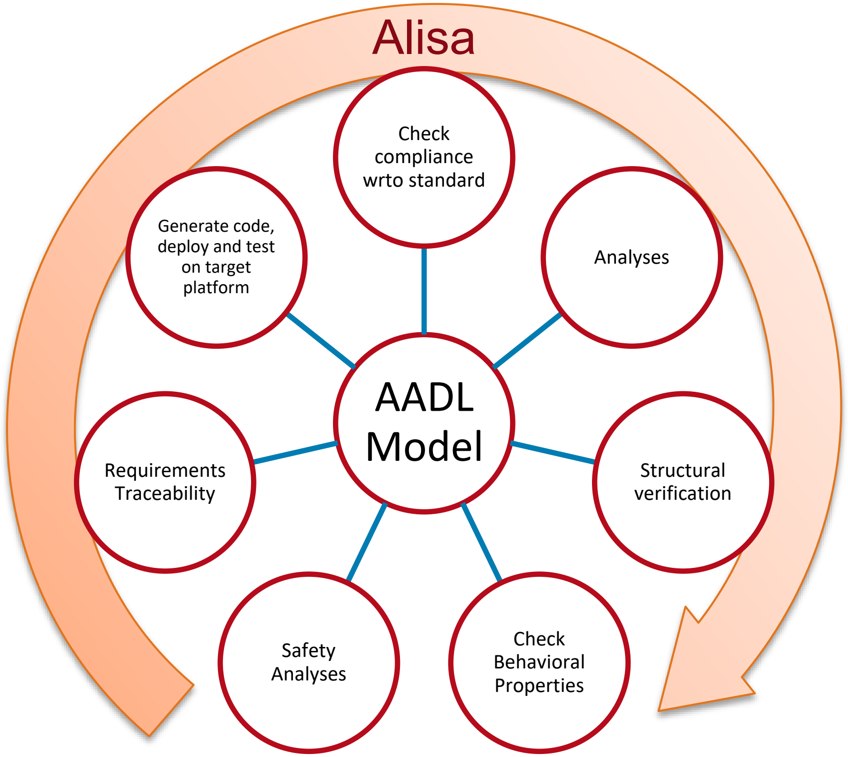
AADL and Verification


Modèle AADL
Check compliance wrto standard
Syntax and semantics
Connections
Temporal properties
Etc.
Analyses
Latency
Scheduling (Cheddar)
Mass
Etc.
Structural Properties Verification
Resolute
LAMP (AADL Inspector)
Behavior Properties Verification
Design by contract/Proof (AGREE - Lustre)
Behavior Annex -> Temporal simulation (AADL Inspector - Marzhin)
Safety Analyses
EMV2 Annex (ARP4761)
Modeling error behavior
Fault trees
Traceability
Goals, Requirements, Stakeholders
Verification Plans
Code generation, deployment and tests

Osate

Alisa

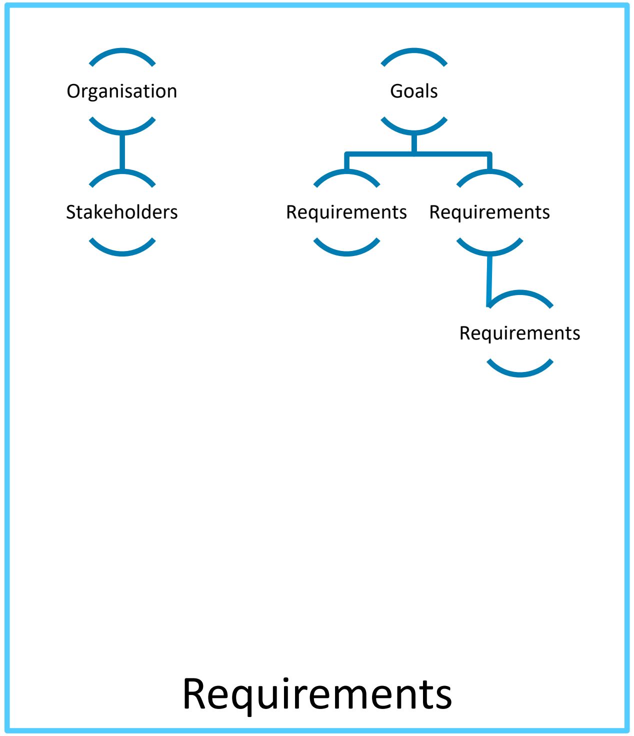
Why an automatic verification plan?

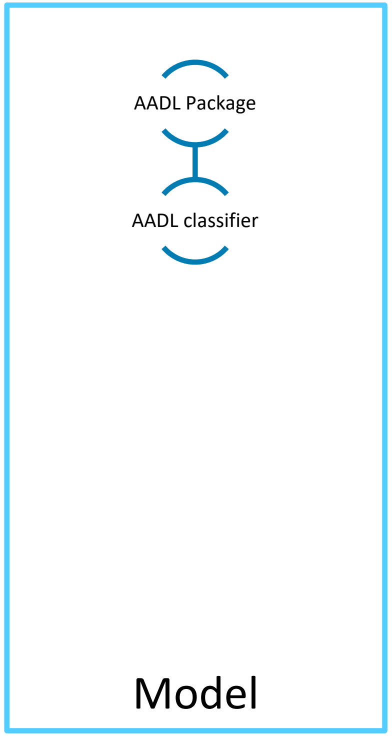

Alisa
Check compliance wrto standard
Generate code, deploy and test on target platform
Analyses
AADL Model
Requirements Traceability
Structural verification
Safety Analyses
Check Behavioral Properties

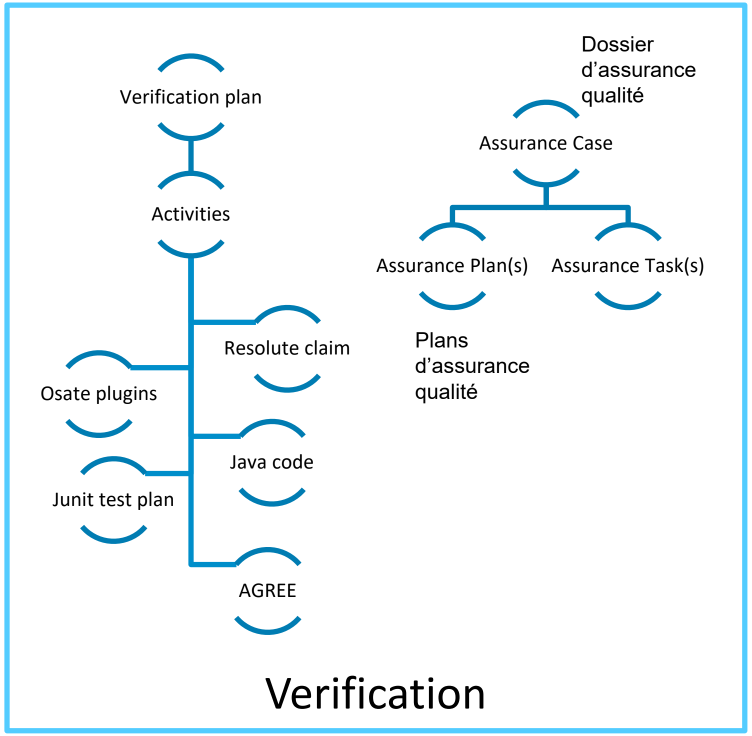
- **How do we scale?**
  - Adding verifications
    - Link to requirements
    - Integrate and sequence the steps of the design verification

→ The Alisa environment

ALISA - Concepts and organisation



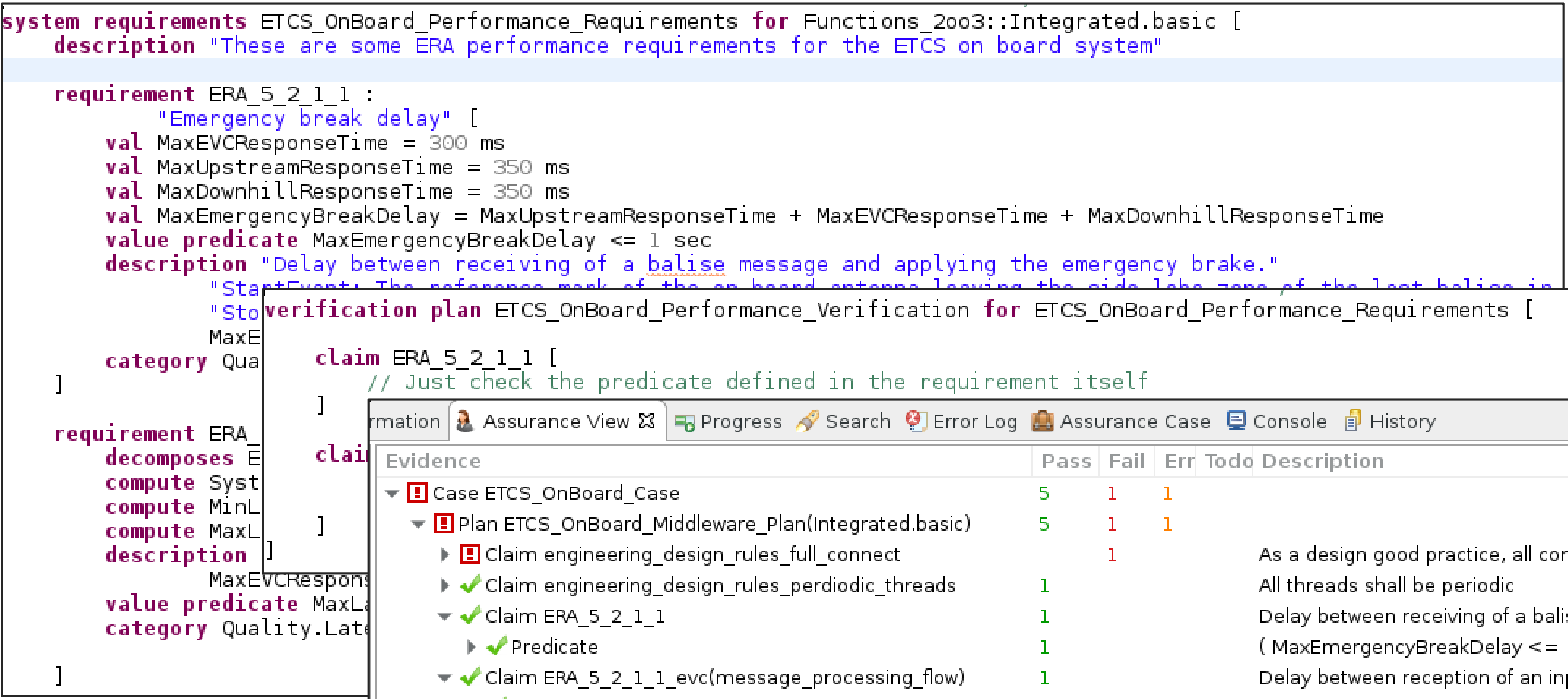




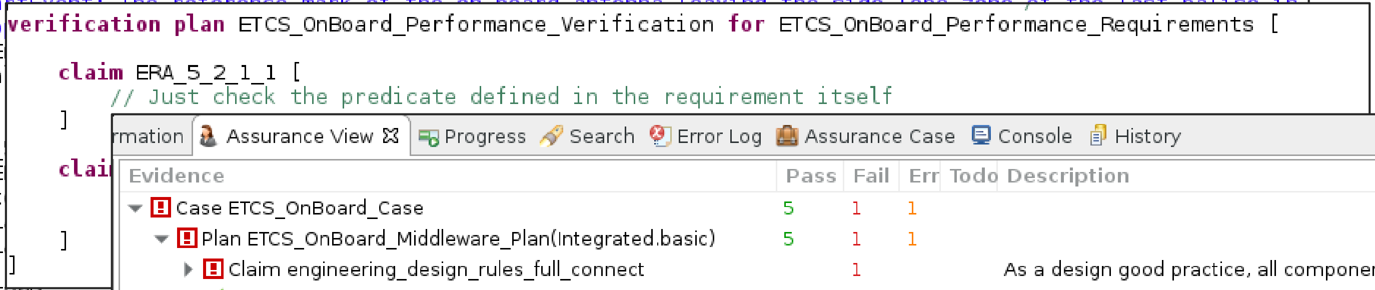
Requirement examples for the EVC

- **Latency < 300 ms**
- **Throughput >= 1000 msg/s**
- **Safety and availability**
  - 2oo3 (aka TMR) design
    - Verify some 2oo3 design constraints
      - Same threads shall run on each board
      - Boards shall be of the same model
- **Design rules (reusable good practices)**
  - All input and output ports, physical or logical, shall be connected.
  - All threads shall be periodic

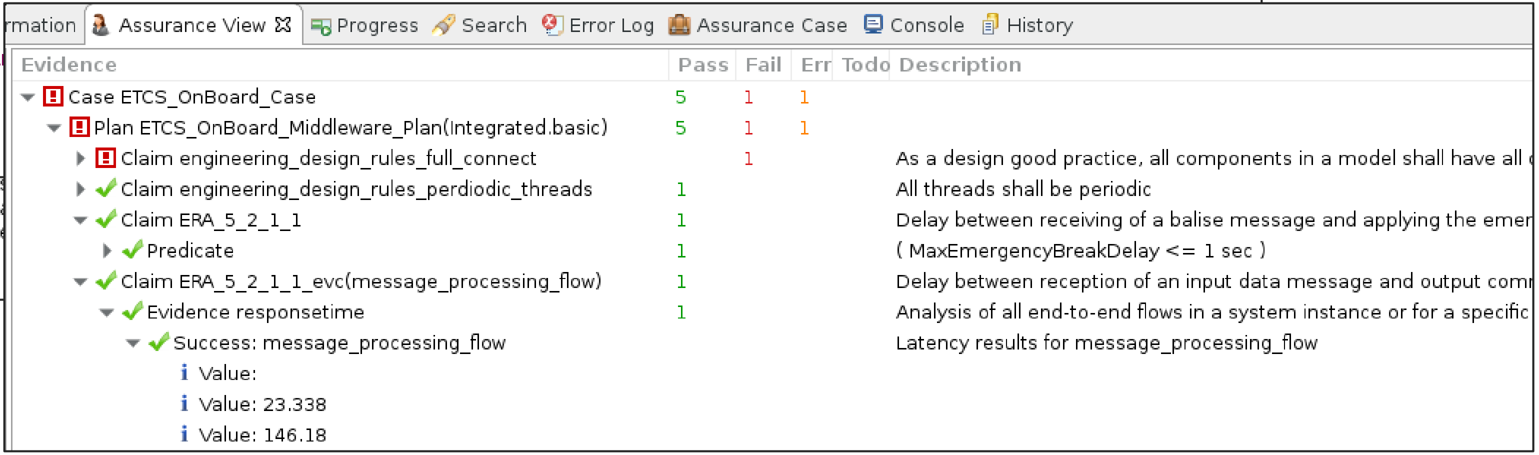
EVC Requirements - ALISA



```
system requirements ETCS_OnBoard_Performance_Requirements for Functions_2oo3::Integrated.basic [
   description "These are some ERA performance requirements for the ETCS on board system"

   requirement ERA_5_2_1_1 :
         "Emergency break delay" [
      val MaxEVCResponseTime = 300 ms
      val MaxUpstreamResponseTime = 350 ms
      val MaxDownhillResponseTime = 350 ms
      val MaxEmergencyBreakDelay = MaxUpstreamResponseTime + MaxEVCResponseTime + MaxDownhillResponseTime
      value predicate MaxEmergencyBreakDelay <= 1 sec
      description "Delay between receiving of a balise message and applying the emergency brake."
            "StartEvent: The reference mark of the on board antenna leaving the side lobe zone of the last balise in
            "Sto
            MaxE
      category Qua
   ]

   requirement ERA_
      decomposes E
      compute Syst
      compute MinL
      compute MaxL
      description
            MaxEVCRespons
      value predicate MaxL
      category Quality.Late

   ]
```



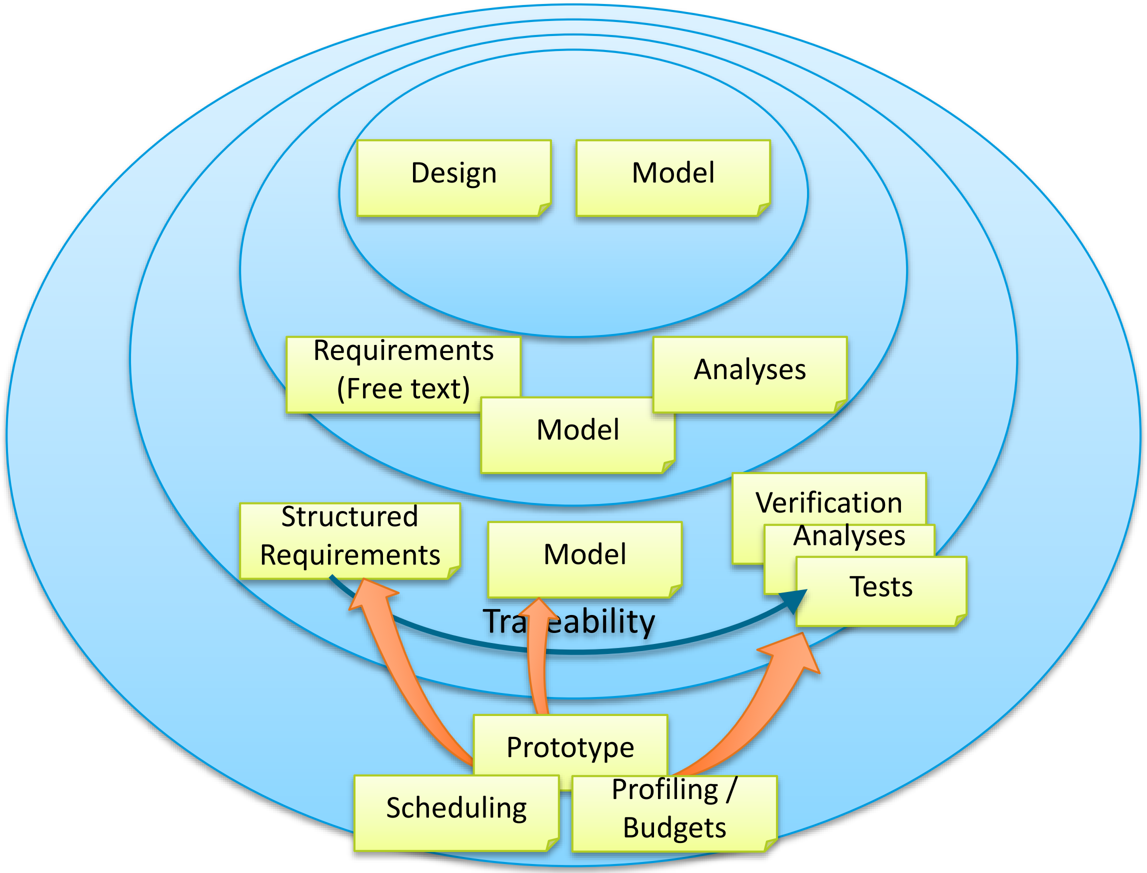
```
verification plan ETCS_OnBoard_Performance_Verification for ETCS_OnBoard_Performance_Requirements [

   claim ERA_5_2_1_1 [
      // Just check the predicate defined in the requirement itself
   ]

   clai


   ]
]
```



rmation | Assurance View | Progress | Search | Error Log | Assurance Case | Console | History

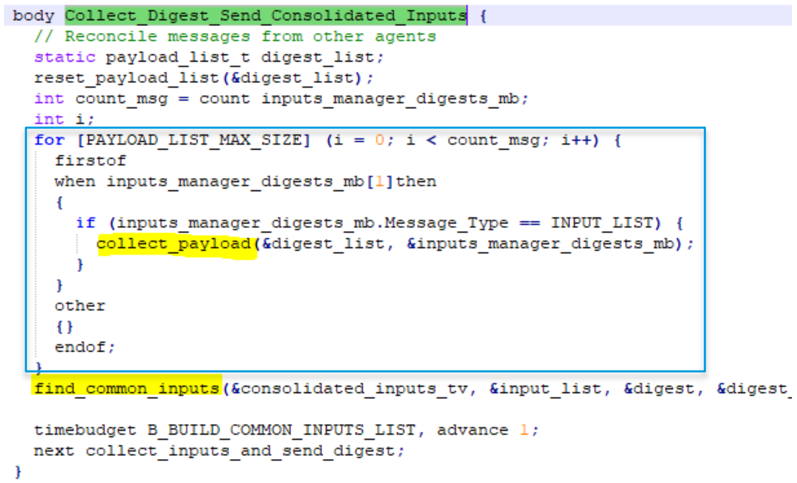
```
Evidence                                                  Pass  Fail  Err  Todo  Description
▾ Case ETCS_OnBoard_Case                                   5     1     1
  ▾ Plan ETCS_OnBoard_Middleware_Plan(Integrated.basic)    5     1     1
    ▸ Claim engineering_design_rules_full_connect                1                As a design good practice, all components in a model shall have all
    ▸ Claim engineering_design_rules_perdiodic_threads     1                      All threads shall be periodic
    ▾ Claim ERA_5_2_1_1                                    1                      Delay between receiving of a balise message and applying the emer
      ▸ Predicate                                          1                      ( MaxEmergencyBreakDelay <= 1 sec )
    ▾ Claim ERA_5_2_1_1_evc(message_processing_flow)       1                      Delay between reception of an input data message and output com
      ▾ Evidence responsetime                              1                      Analysis of all end-to-end flows in a system instance or for a specific
        ▾ Success: message_processing_flow                                        Latency results for message_processing_flow
            i Value:
            i Value: 23.338
            i Value: 146.18
```

Iterative incremental approach with AADL

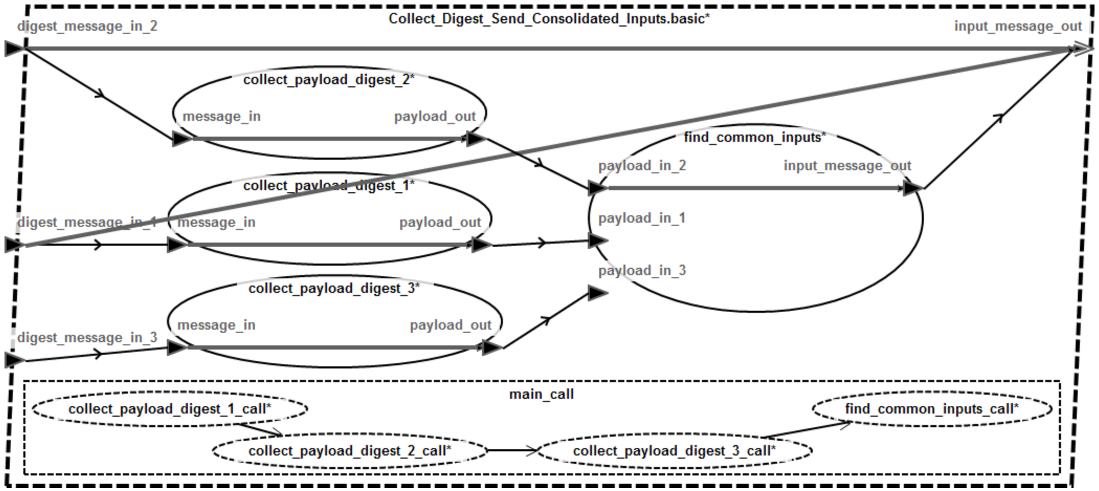


- Expressivity
- Performance Analyses
- Traceability and requirements verification
- Prototyping
- Model refinement

Model refinement



```
body Collect_Digest_Send_Consolidated_Inputs {
  // Reconcile messages from other agents
  static payload_list_t digest_list;
  reset_payload_list(&digest_list);
  int count_msg = count inputs_manager_digests_mb;
  int i;
  for [PAYLOAD_LIST_MAX_SIZE] (i = 0; i < count_msg; i++) {
    firstof
    when inputs_manager_digests_mb[1]then
    {
      if (inputs_manager_digests_mb.Message_Type == INPUT_LIST) {
        collect_payload(&digest_list, &inputs_manager_digests_mb);
      }
    }
    other
    {}
    endof;
  }
  find_common_inputs(&consolidated_inputs_tv, &input_list, &digest, &digest_

  timebudget B_BUILD_COMMON_INPUTS_LIST, advance 1;
  next collect_inputs_and_send_digest;
}
```



Model features refinement

Design space

- **Architecture choices**
  - 2oo3
  - Pipeline
- **Design parameters**
  - # Cores
  - Application and middleware budgets: $\boldsymbol{B}_A, \boldsymbol{B}_M$
  - Pipeline period: P_{pl}
  - FIFOs Size: L_Q

Constraints

- Response time: $\boldsymbol{T_r}$
- Incoming messages rate: $\boldsymbol{M}_{/s}$

KPIs

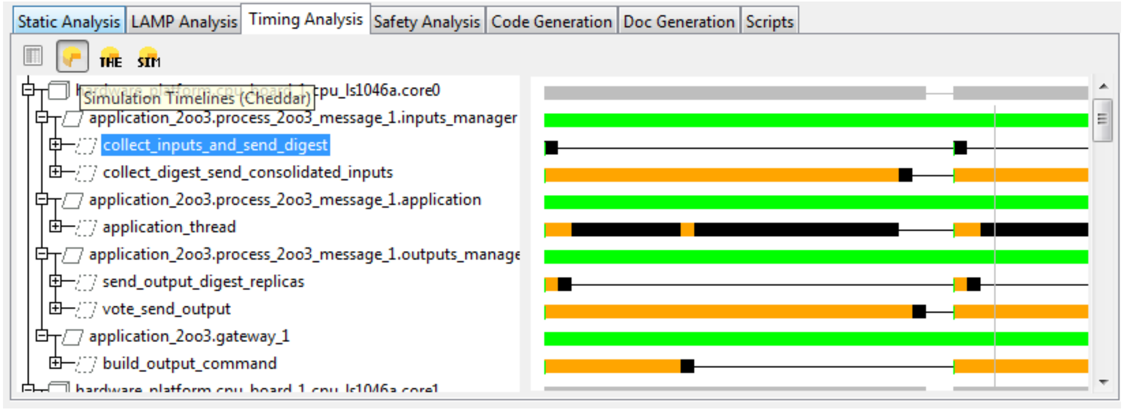
- **Sizing (hardware choice)**
  - Memory consumption: μ_c
  - # Cores
  - Price of the system: $C_\$$
- **Design quality**
  - Ratio Application budget/Pipeline Period: $\boldsymbol{R_A}$

Single core design

Formulas

- $R_A = \frac{B_A}{P_{pl}}$
- $L_Q = M_{/s} \times P_{pl}$
- $\mu_C = O(L_Q)$
- $B_M = \alpha L_Q = \alpha M_{/s} P_{pl}$

Scheduling - 1 core


Static Analysis | LAMP Analysis | Timing Analysis | Safety Analysis | Code Generation | Doc Generation | Scripts

Simulation Timelines (Cheddar)

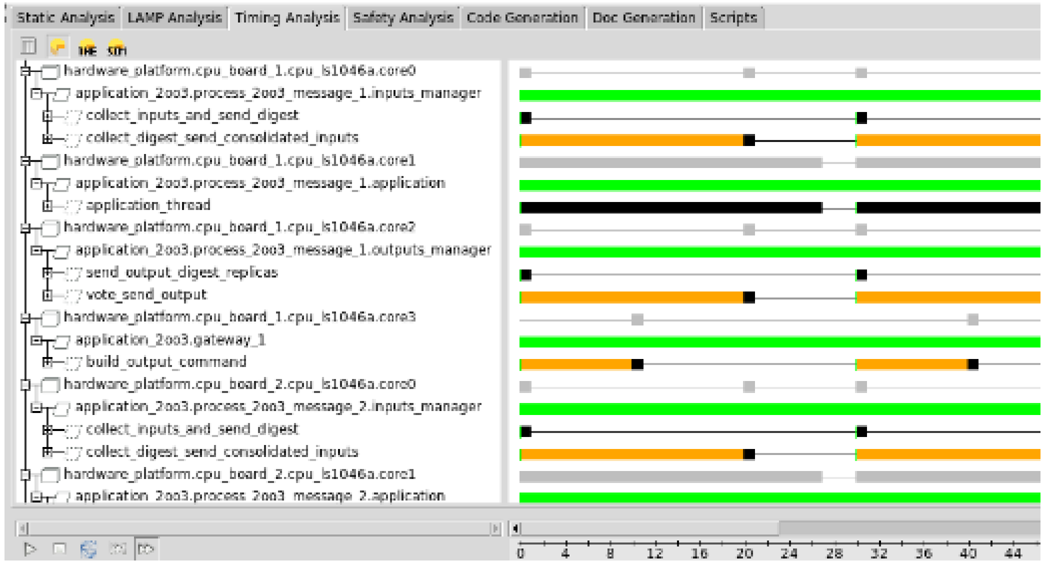
```
hardware_platform.cpu_board_1.cpu_ls1046a.core0
  application_2oo3.process_2oo3_message_1.inputs_manager
    collect_inputs_and_send_digest
    collect_digest_send_consolidated_inputs
  application_2oo3.process_2oo3_message_1.application
    application_thread
  application_2oo3.process_2oo3_message_1.outputs_manage
    send_output_digest_replicas
    vote_send_output
  application_2oo3.gateway_1
    build_output_command
hardware_platform.cpu_board_1.cpu_ls1046a.core1
```

Single core

- $B_A + B_M \approx P_{pl}$
- $R_A = 1 - \alpha M_{/s}$

Multi-core Design

Scheduling - 4 cores



hardware_platform.cpu_board_1.cpu_ls1046a.core0
application_2oo3.process_2oo3_message_1.inputs_manager
collect_inputs_and_send_digest
collect_digest_send_consolidated_inputs
hardware_platform.cpu_board_1.cpu_ls1046a.core1
application_2oo3.process_2oo3_message_1.application
application_thread
hardware_platform.cpu_board_1.cpu_ls1046a.core2
application_2oo3.process_2oo3_message_1.outputs_manager
send_output_digest_replicas
vote_send_output
hardware_platform.cpu_board_1.cpu_ls1046a.core3
application_2oo3.gateway_1
build_output_command
hardware_platform.cpu_board_2.cpu_ls1046a.core0
application_2oo3.process_2oo3_message_2.inputs_manager
collect_inputs_and_send_digest
collect_digest_send_consolidated_inputs
hardware_platform.cpu_board_2.cpu_ls1046a.core1
application_2oo3.process_2oo3_message_2.application

Multi-core

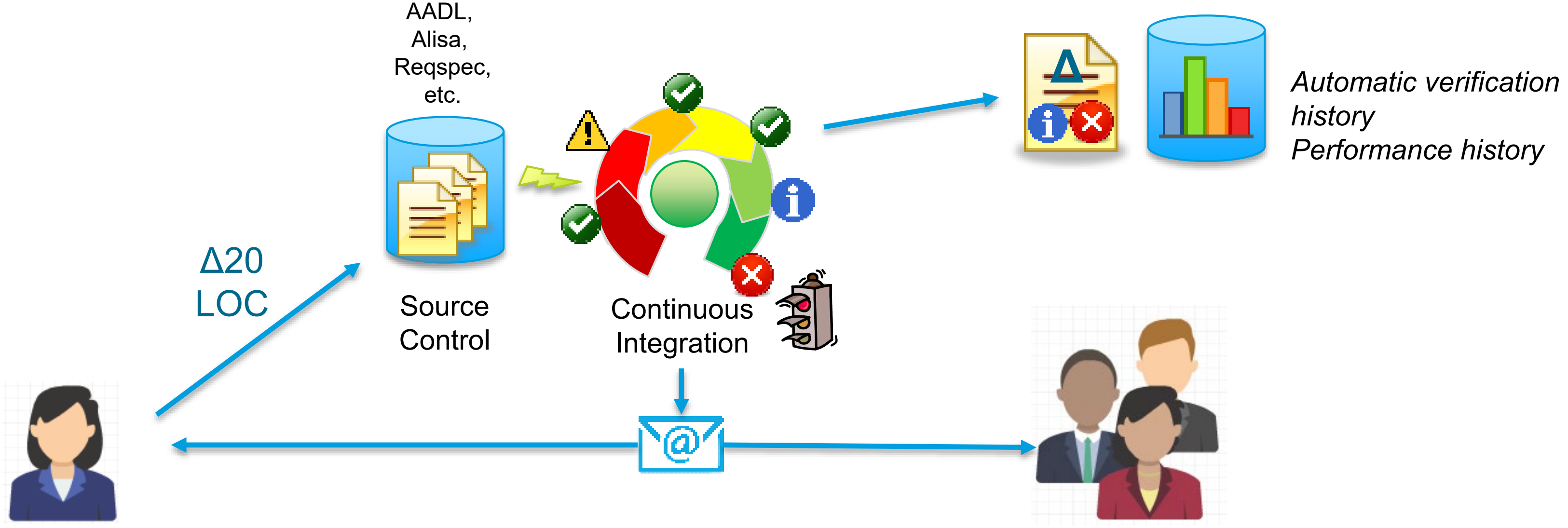
- $R_A \approx 1$

AADL/ALISA implementation

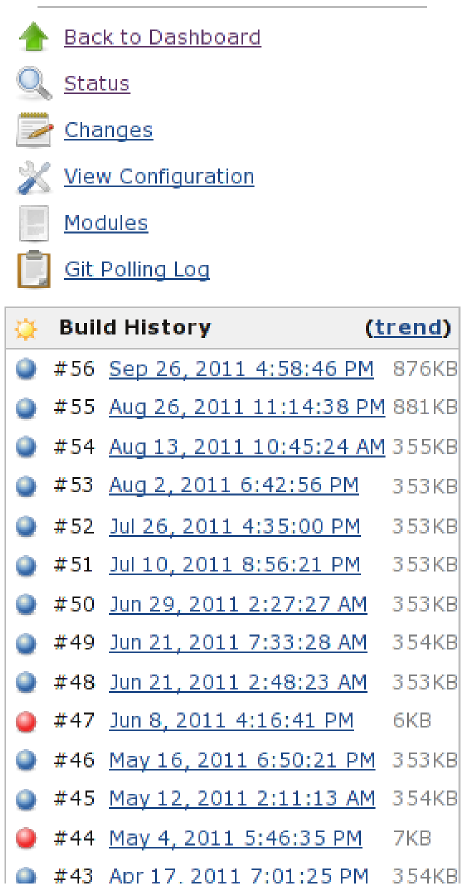
```
system requirements ETCS_OnBoard_Performance_2 for Middleware_2oo3::Integrated.basic [
    val ExpectedMessageRate : integer = 1000

    requirement message_rate : "EVC message processing rate" [
        description "The EVC shall process " ExpectedMessageRate " per second"
        compute ActualMessageRate : integer
        value predicate ActualMessageRate >= ExpectedMessageRate
    ]

    requirement FIFOs_size : "FIFOs size" [
        val MaxFIFOSize = 500 ms
        description "Keep FIFOs size under control: size shall be <= " MaxFIFOSize
        val FIFOsSize = ExpectedMessageRate * #Middleware_Properties::Default_Hyper_Period
        value predicate FIFOsSize <= MaxFIFOSize
    ]

    requirement application_budget : "Application budget" [
        val ApplicationBudget : Time = min(this.evc.application_2oo3.process_2oo3_message_2.application.application_thread#Compute_Execution_Time)
        val PipelinePeriod = #Middleware_Properties::Default_Hyper_Period
        val AppBudgetRatio = ApplicationBudget / PipelinePeriod
        value predicate AppBudgetRatio > 0.5
    ]
]
```

From automatic verification to continuous virtual integration: an agile engineering process based on AADL

Continuous virtual integration



Verification history



Back to Dashboard

Status

Changes

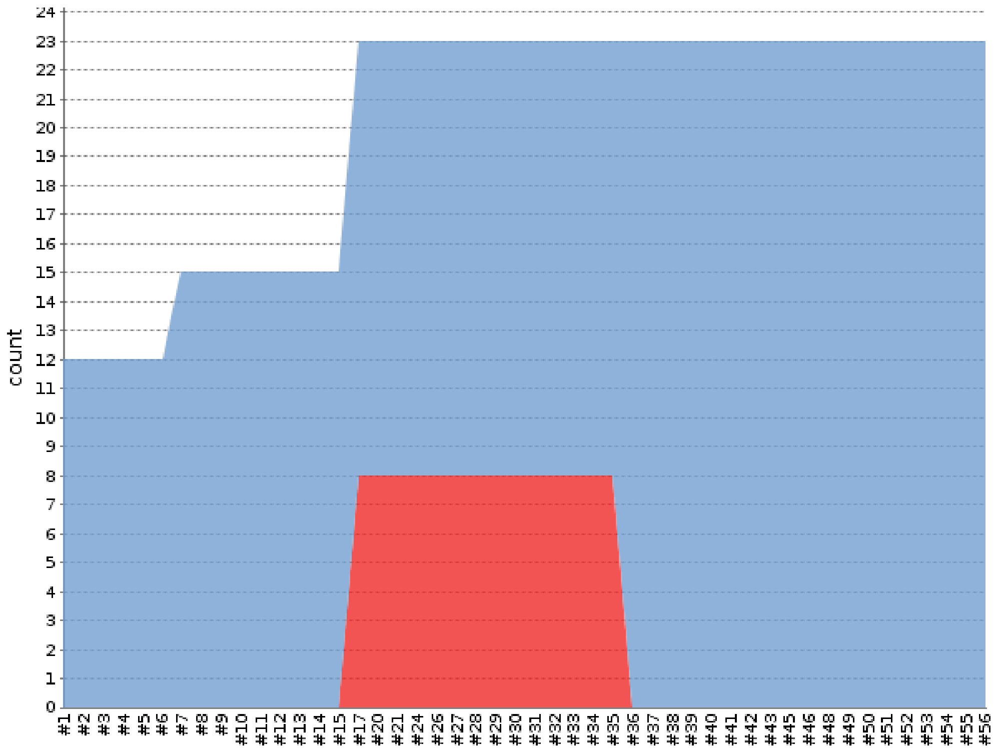
View Configuration

Modules

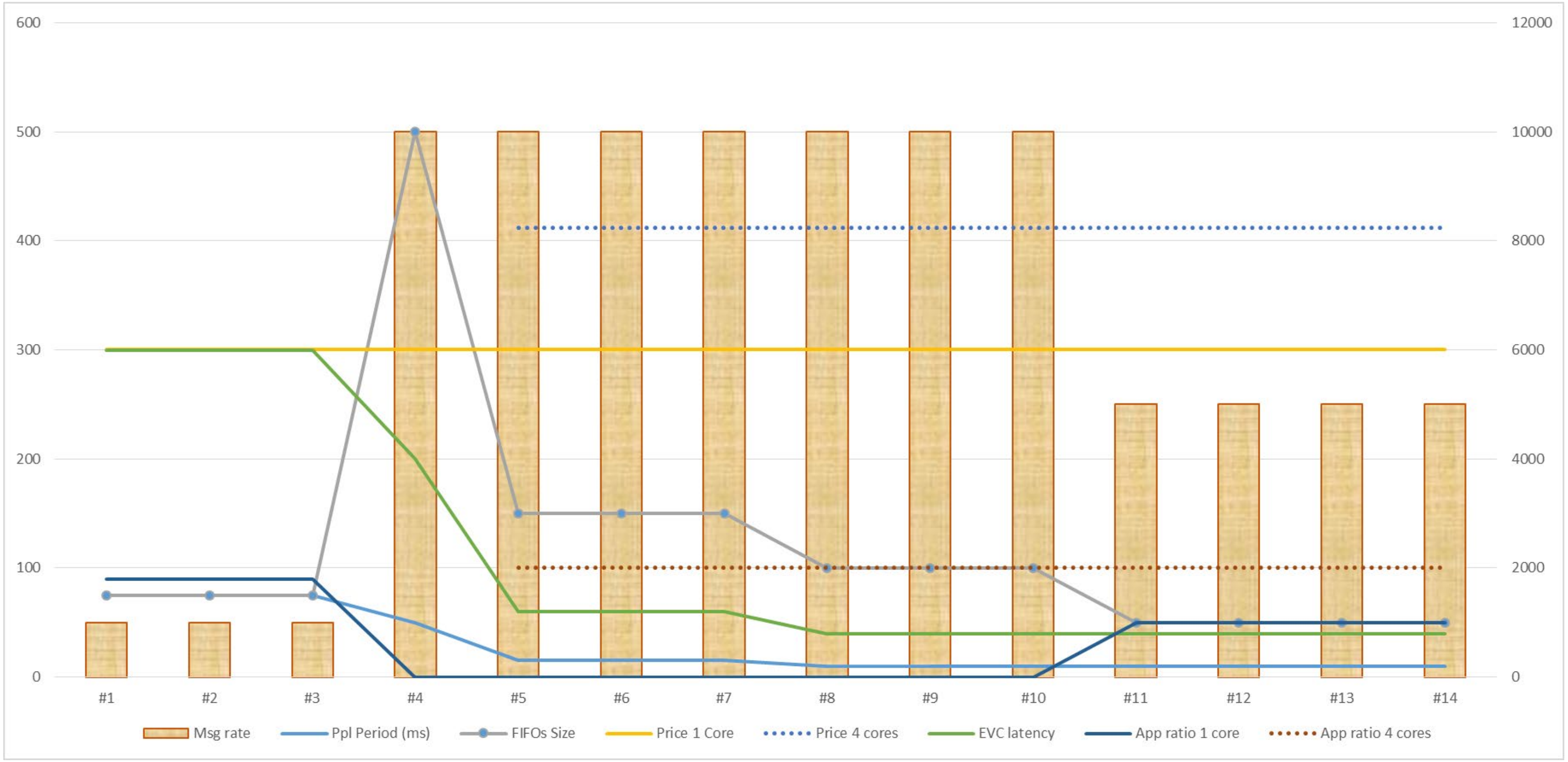
Git Polling Log

Build History (trend)

#56	Sep 26, 2011 4:58:46 PM	876KB
#55	Aug 26, 2011 11:14:38 PM	881KB
#54	Aug 13, 2011 10:45:24 AM	355KB
#53	Aug 2, 2011 6:42:56 PM	353KB
#52	Jul 26, 2011 4:35:00 PM	353KB
#51	Jul 10, 2011 8:56:21 PM	353KB
#50	Jun 29, 2011 2:27:27 AM	353KB
#49	Jun 21, 2011 7:33:28 AM	354KB
#48	Jun 21, 2011 2:48:23 AM	353KB
#47	Jun 8, 2011 4:16:41 PM	6KB
#46	May 16, 2011 6:50:21 PM	353KB
#45	May 12, 2011 2:11:13 AM	354KB
#44	May 4, 2011 5:46:35 PM	7KB
#43	Apr 17, 2011 7:01:25 PM	354KB



Performance history



SAVI

- SAVI (System Architecture Virtual Integration)
  - https://savi.avsi.aero/
- Several research projects:
  - US Army, DARPA, NASA, ESA, FDA
- Domains:
  - Avionics, aerospace, medical, nuclear, automotive, robotics

Building blocks

Osate/ALISA/AADL Inspector

AADL parsing, analysis and verification platform

Git/Repo

Versioning system for the comprehensive source of all artifacts:

- **Requirements**
- **Models and Code**
- **Verification activities**
- **Dockerfiles**

Jenkins

Continuous integration

Triggers verification check on any change to the artifacts

Docker

Container platform

Configuration management of the development, build and test environments

- **Have a way to store analyses result values in the `.xassure` file**
- **There are issues with Time and Time range values in reqspec (see https://github.com/paolo-crisafulli/alisa-value-predicate)**
- **Run ALISA headless**
- **Enable design goals**
- **Use Resolute functions for `compute` values computation**
- **Enhanced verification reporting**
  - Latency analysis