GNC System Verification and Certification Processes

Harald Ruess and Tewodros A. Beyene

fortiss GmbH
An-Institut Technische Universität München

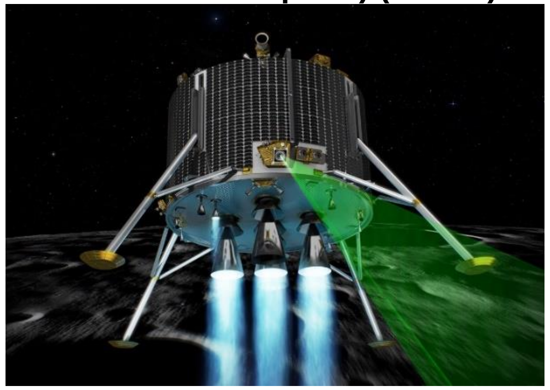
GNC Safety Certification

Safety Certification Challenges

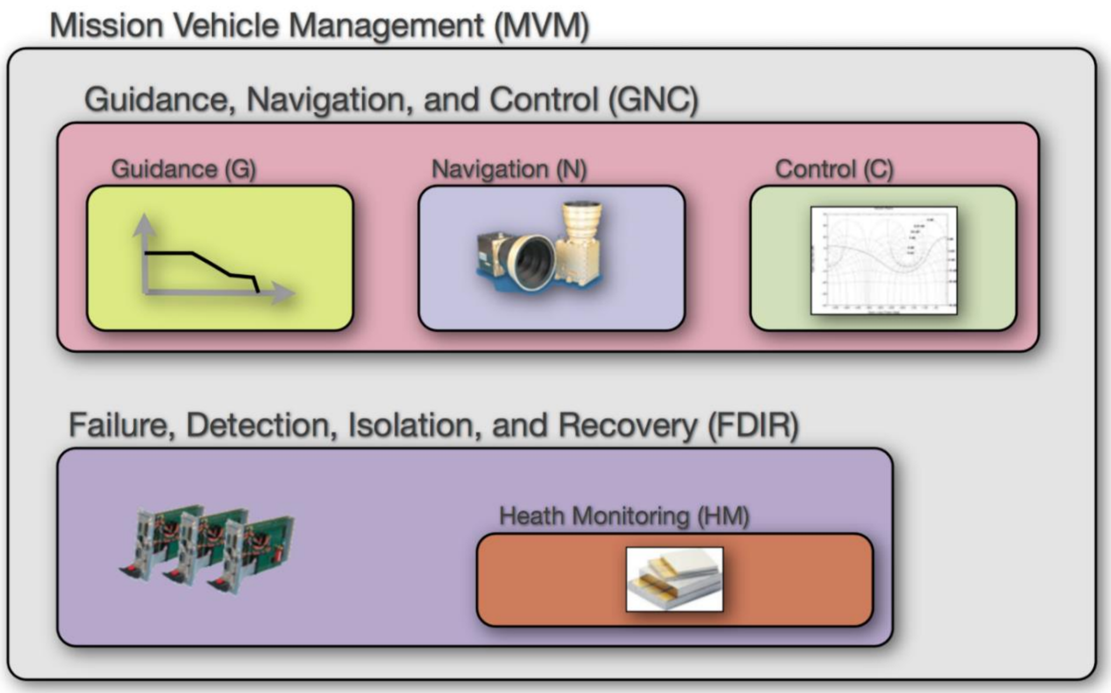
- **General challenges** for space systems
  - system level changes for small changes in components
  - testing and simulations cannot guarantee coverage
  - methods for **compositional certification** (cmp. DO 297)
  - methods for **early safety assessment**
  - certification of COTS hardware and software

- **GNC specific challenges**
  - efficient methods for **correct-by-construction** synthesis of FDIR components of the GNC system (cmp. ESA projects FASE/COMPASS)
  - **verification and validation methods for autonomous GNC**

ESA Lunar Lander (Artist Conception) (© ESA)



The PROBA-V Satellite (© ESA)



Unless embedded control software for highly automated and autonomous Systems can be developed, verified, and certified with less cost and effort – while still satisfying the highest dependability requirements – these new capabilities may never reach the market…

fortiss

GNC Safety Certification

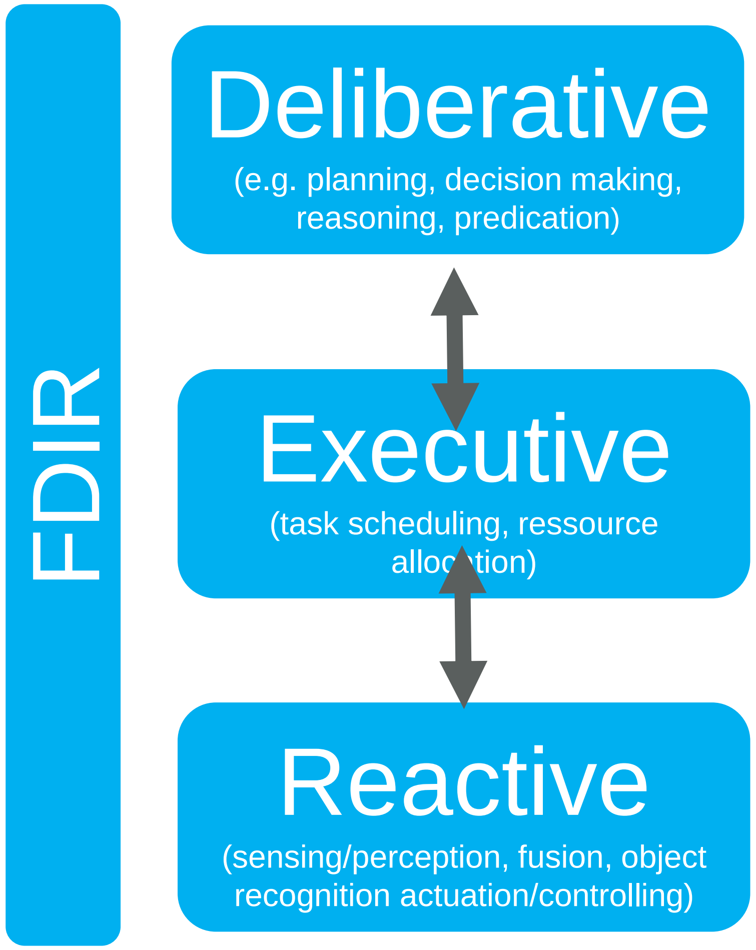
Safety Certification Challenges for Autonomous GNC

- In **piloted systems**, designers take advantage of the **human ability**
  - **to deal with uncertainty**,
  - to be able to make decisions with **incomplete or ambiguous information**, and
  - to provide the outer-loop control input that **manages any contingency while maintaining stability and control**.

- At least part of the **fallback safety mechanism** has always **relied on human intervention**.

- The **machine** itself remains completely **deterministic**

- Future space systems might make their **own judgements** and **decisions.**
  - New **V&V technologies needed** to enable timely and efficient certificate of the autonomous control systems
  - Need to cope with **environments** which
    - **cannot be comprehensively monitored or controlled**, and
    - in which **unpredictable events** may occur



- But **current certification processes** (e.g. civil aviation) are based on the idea that the correct behavior must be **completely specified** and **verified prior to operation**.

GNC Safety Certification

PART I:

Safety certification requirements

PART II

State-of-the-art analysis

PART III:

Gap assessment

PART IV:

Draft methodology

PART V:

Technology recommendations

fortiss

esa

ESA STUDY CONTRACT REPORT

ESA CONTRACT No **4000117995/16/NL/HK/as**	SUBJECT **Final Report**		CONTRACTOR **fortiss GmbH, Guerickestraße 25, 80805 Munich, Germany**
ESA CR( )No	STAR CODE:	No of volumes This is Volume No	CONTRACTOR'S REFERENCE

ABSTRACT:

Developing a safety certification methodology for GNC space systems needs understanding of safety certification requirements as well as state of the art in certifying safety critical systems in other domains. In this report, we present a work conducted to draft a safety certification methodology aimed at complementing existing certification practices in the space domain for GNC space systems. The work consists of five activities: studying safety requirements, analyzing state of art in safety certification, assessing the gap, proposing a draft certification methodology and providing technology recommendation for the realization of the draft methodology. Each of these five activities are presented in this report together with general introduction.

The work described in this report was done under ESA contract. Responsibly for the contents resides in the author or organisation that prepared it.

Names of the authors: **Tewodros A. Beyene**
Chih-Hong Chong
Harald Ruess

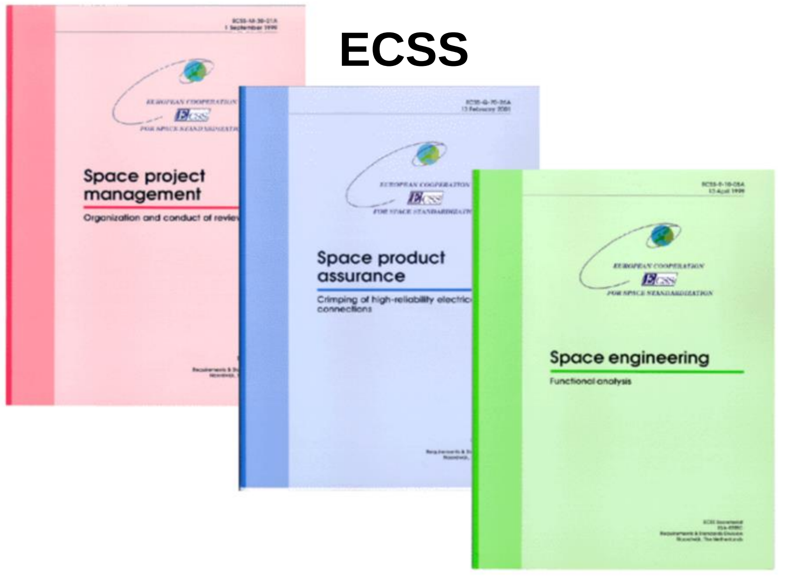
ESA STUDY MANAGER:	ESA BUDGET HEADING:
Dr. Guillermo Ortega (TEC-SAG) GNC, AOCS and Pointing Systems (TEC-SA) DIRECTORATE OF TECHNOLOGY, ENGINEERING AND QUALITY (D/TEC)	

GNC Safety Certification

Process-based Software Safety Standards for Autonomous Systems

- Current software safety standards (ISO 26262, DO 178C, ECSS) are largely prescriptive and **process-based**
  - Among the most effective safety standards
  - ... but: “Because we cannot demonstrate how well we’ve done, we’ll show how hard we’ve tried” (J. Rushby, HCSS Aviation Safety Workshop, 2006)

ECSS



- These standards recommend a set of techniques and methods for safe development of software, but they*
  - *“pay little attention to autonomy and to the particular advanced software technologies for system autonomy”*
  - *“In practice the recommended set of techniques and methods for safety-related software may not be easily applicable considering, e.g., the size and complexity of the software and of the input and state domains, the dependency of the software behaviour on knowledge bases, etc.”*

*Blanquart J P, Fleury S, Hernek M, Honvault C, Ingrand F, Poncet J C, Powell D, Strady-Lécubin N, and Thévenod P. Software Product Assurance for Autonomy On-Board Spacecraft. Proceedings of DASIA 2003 (ESA SP-532), pages 69A–69G. June 2003.

ISO 26262

fortiss

GNC Safety Certification

Product-base Software Safety Standards for Autonomous Systems

- Def Stan 00-56 Issue 3, **Safety Management Requirements for Defence Systems** presents a possible path towards a certification solution.*

- **…** system safety is justified using a **safety case** structured to present a risk-based argument that the **system is safe**.

- This is a **product-based** safety argument approach rather than a **process-based** one; it involves **the presentation of evidence that the actual developed system is safe**, as opposed to merely showing that it was developed using accepted good practice; 00-56:

  > "Within the safety case, the contractor shall provide compelling evidence that safety requirements have been met. Where possible, objective, analytical evidence shall be provided".

- This gives good scope for the certification of **novel classes** of systems, such as autonomous systems, as the system can be certified if a compelling **safety case** can be built for it.

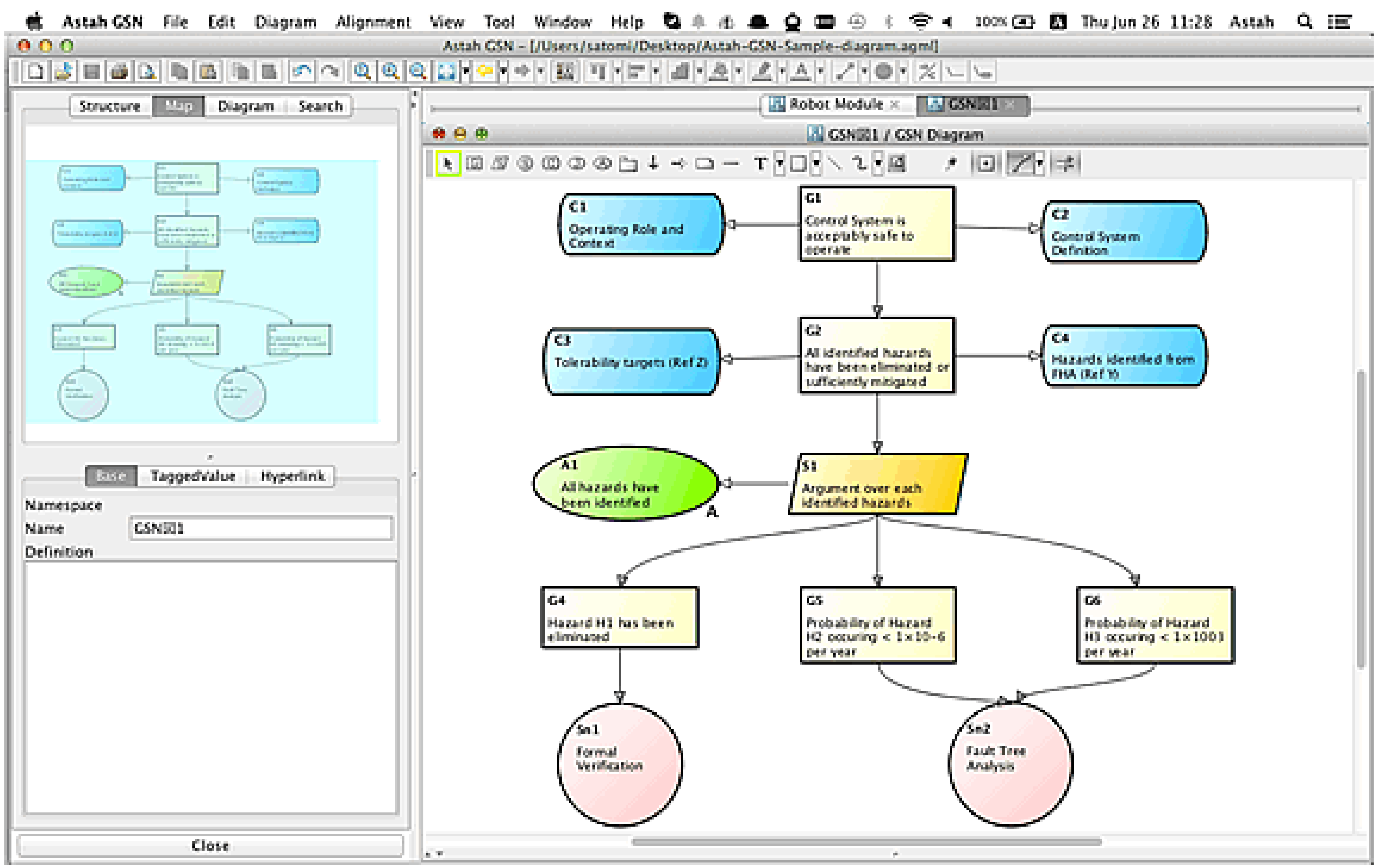
*R. D. Alexander, M. Hall-May, T. P. Kelly, Certification of Autonomous Systems under UK Military Safety Standards University of York; York, England

fortiss

GNC Safety Certification

Safety Case

A structured argument, supported by a **body of evidence** that provides a compelling, comprehensible and valid case that a system is safe for a given application in a given operating environment.



fortiss

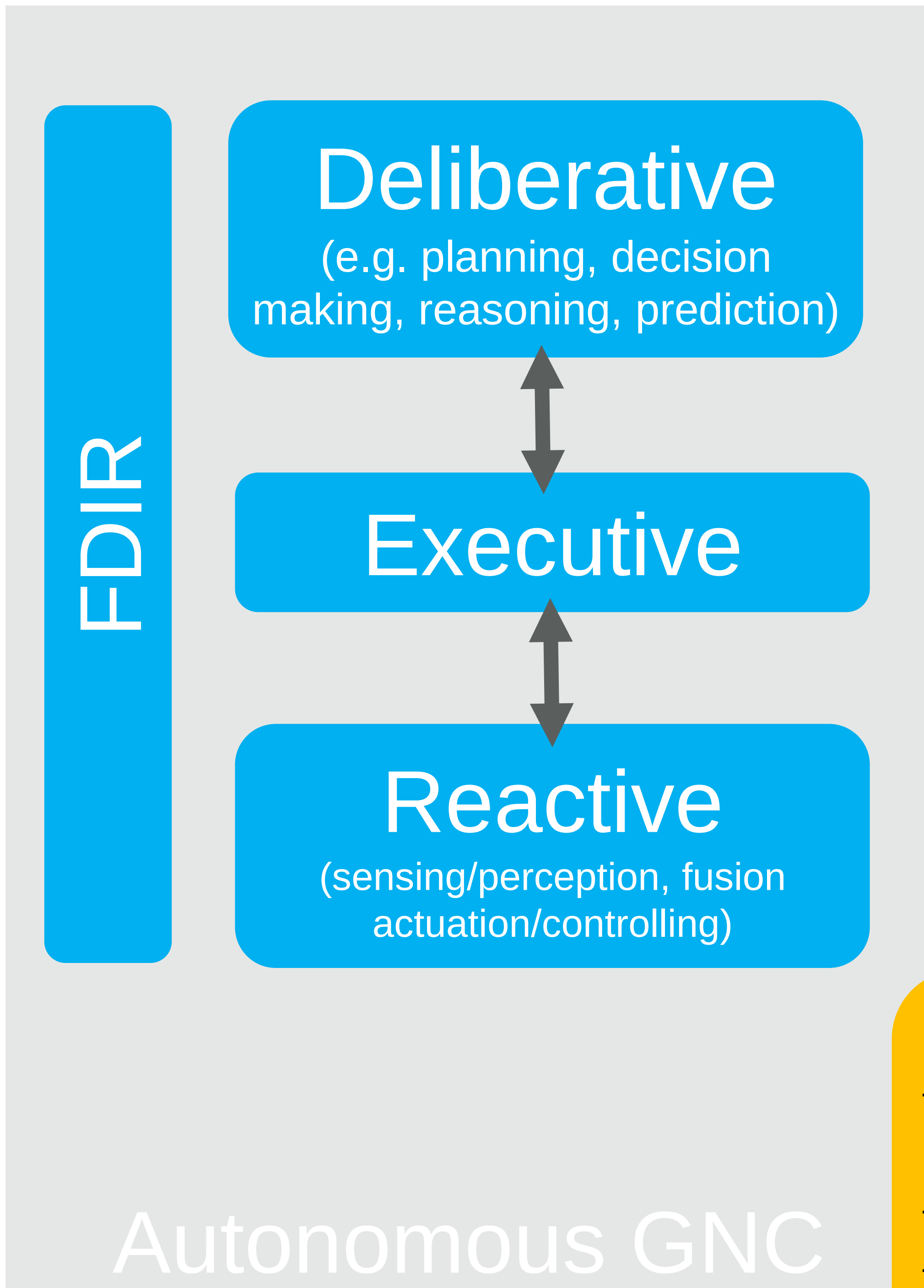
GNC Safety Certification

Product-based Software Safety Standards for Autonomous Systems (Cont.)

- …**but**: the main motive of the use of **autonomous systems** is for those situations where the full **details of the operating environment cannot be known ahead of time**
- Therefore it is **difficult to carry out risk estimation** using conventional techniques
- Standards such as 00-56 therefore provide a **framework** in which
  - the **safety** of any system **can potentially be argued**,
  - but there is **no extant guidance** on how to do this for **autonomous GNC**
- There is therefore a strong **need for** definition of
  - a general safety lifecycle for autonomous GNC,
  - expansion and development of existing safety analysis methods, and
  - for **substantial guidance on the development safety cases**.

GNC Certification Technology

Certification Methodology for 3-level Autonomous Architecture



Synthesis for autonomous FDIR including safety case (e.g. based on FAME/COMPASS project at ESA)

Verification of planning algorithms and/or cognitive structures

Resource efficient runtime monitoring / enforcement of safety goals

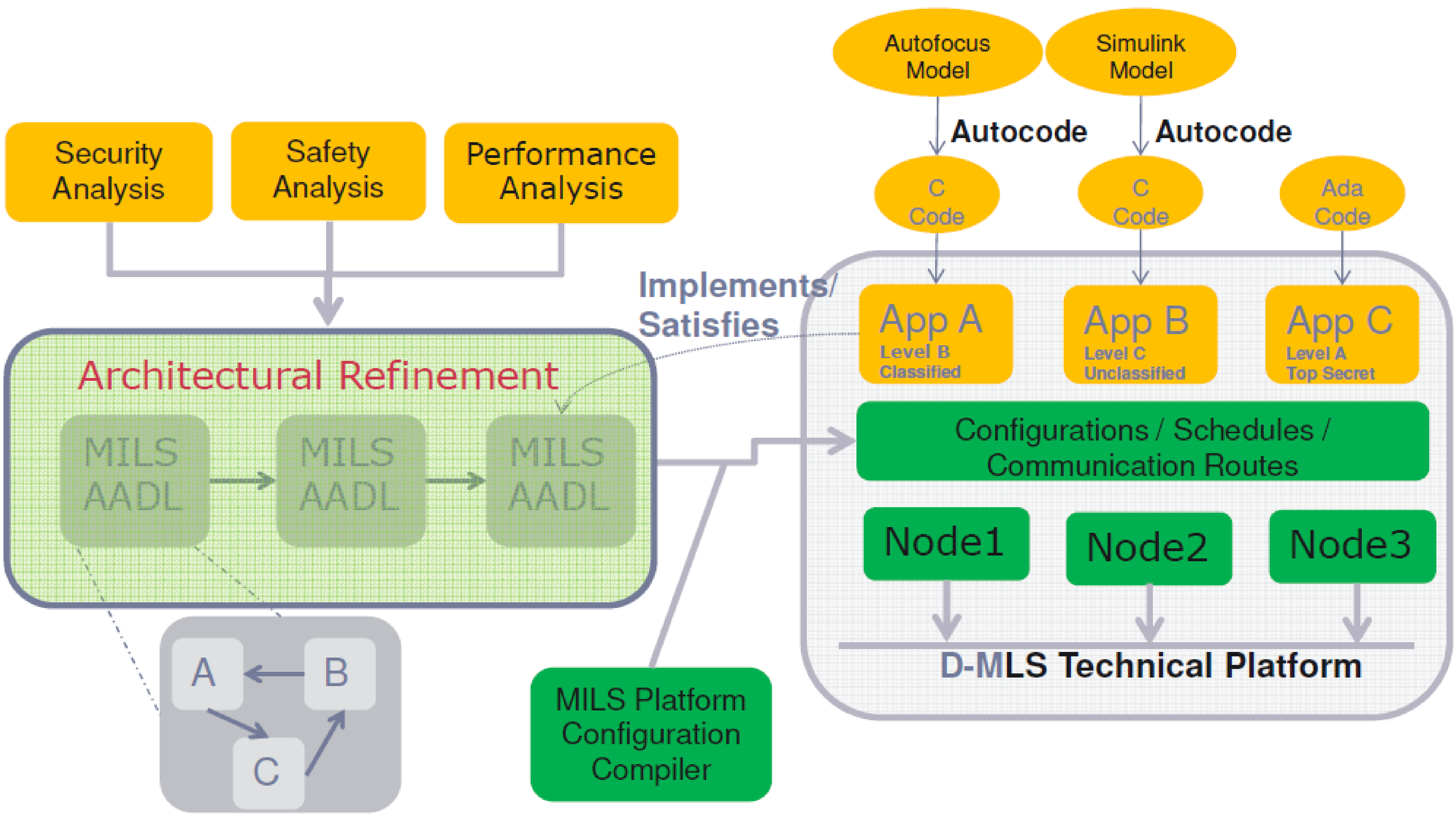
Establishing useful formal properties about neural networks and other machine learning techniques

Putting it all together in a **safety case**

- safety case pattern based on 3-level autonomous architecture
- combining component safety cases
- safety case might include parts which are only generated at runtime

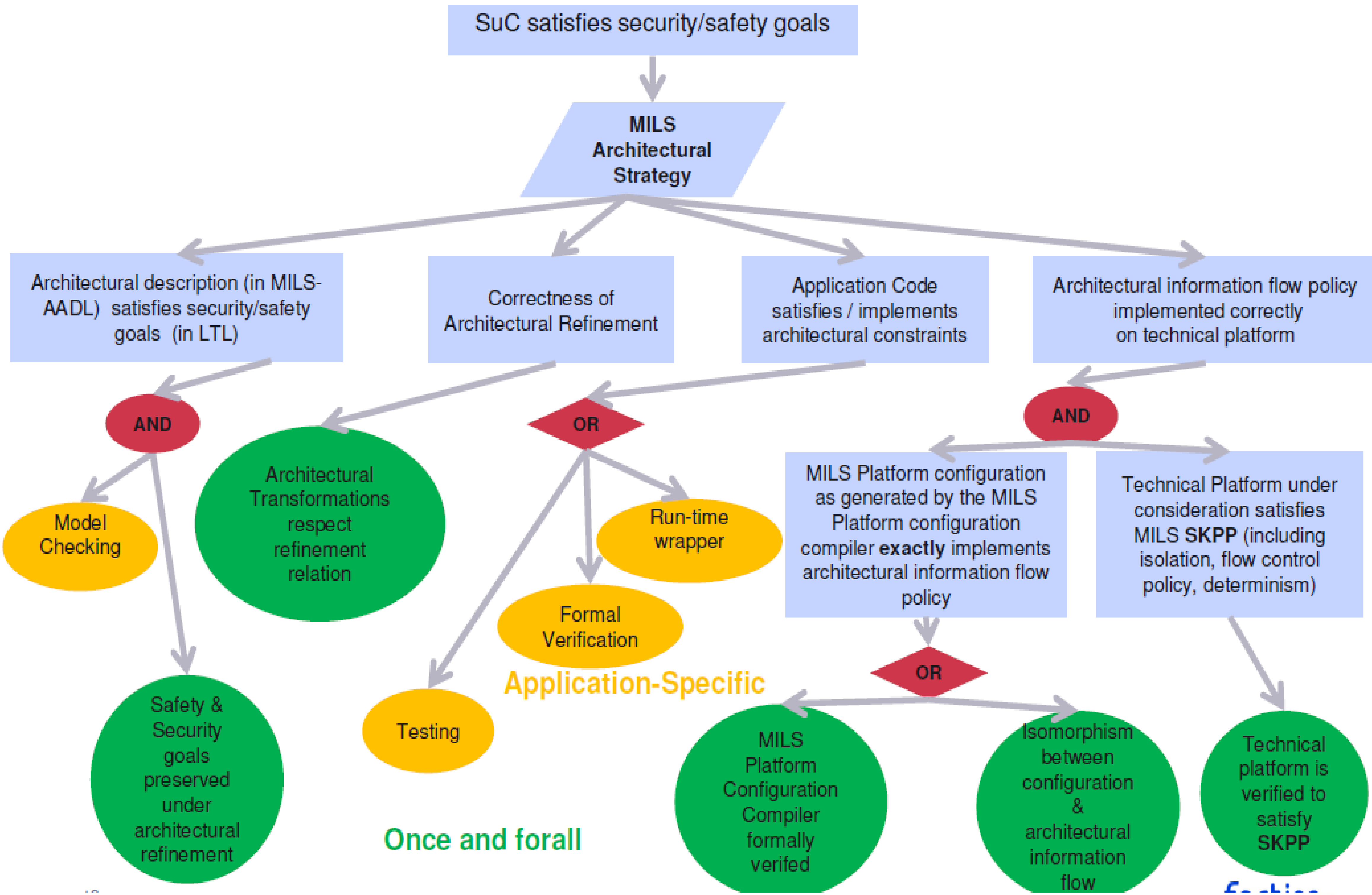
GNC Safety Certification

Technology nugget I: Architectural Safety Case Patterns (Mils)



GNC Safety Certification

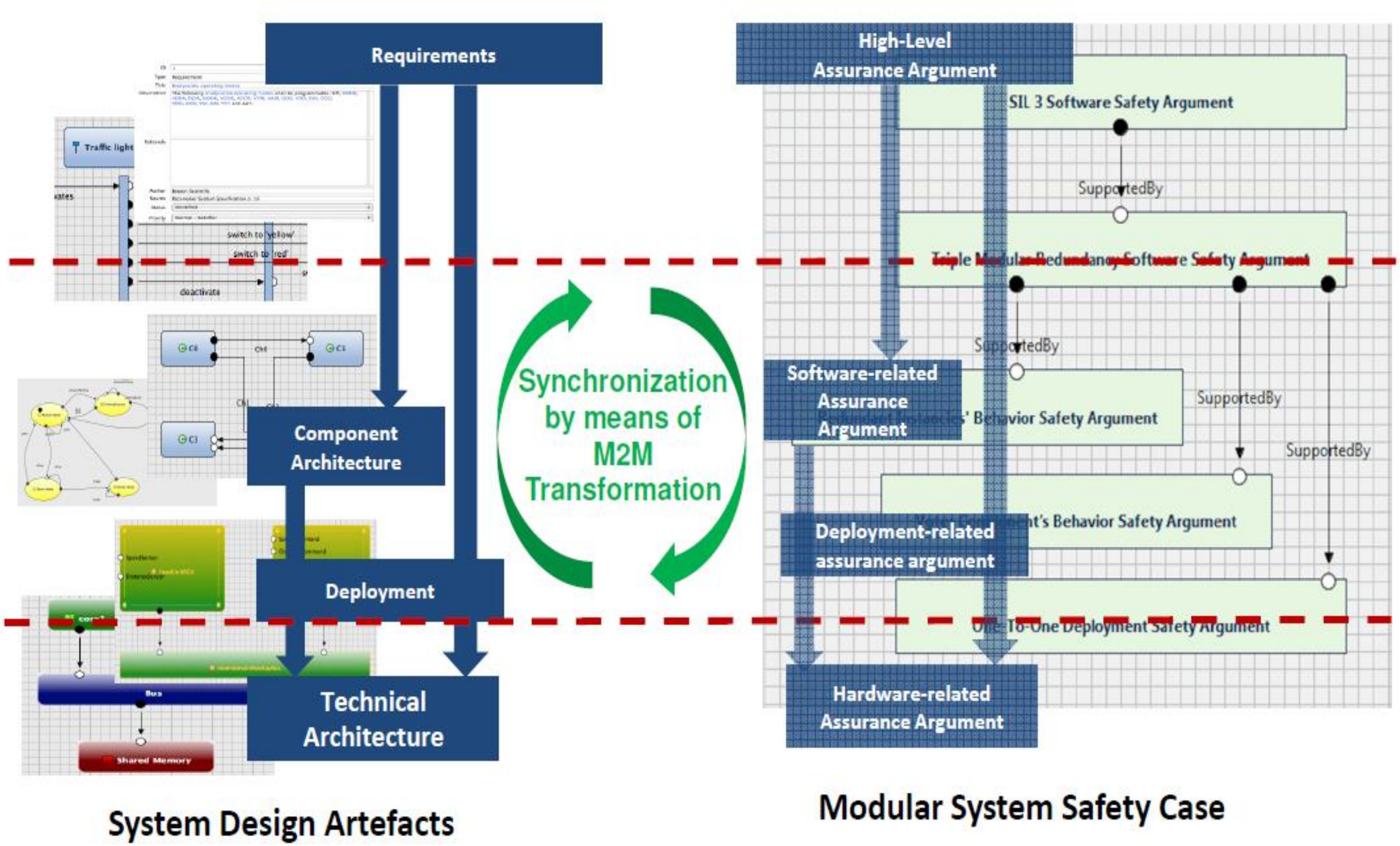
Technology nugget I: Architectural Safety Case Patterns (MILS)



fortiss

GNC Safety Certification

Technology nugget II: Integrated Development of System and Safety Case



Implemented in SystemFocus
(af3.fortiss.org)

fortiss

GNC Certification Technology

Technology Nugget III: Evidential SW Verification Tool Chain

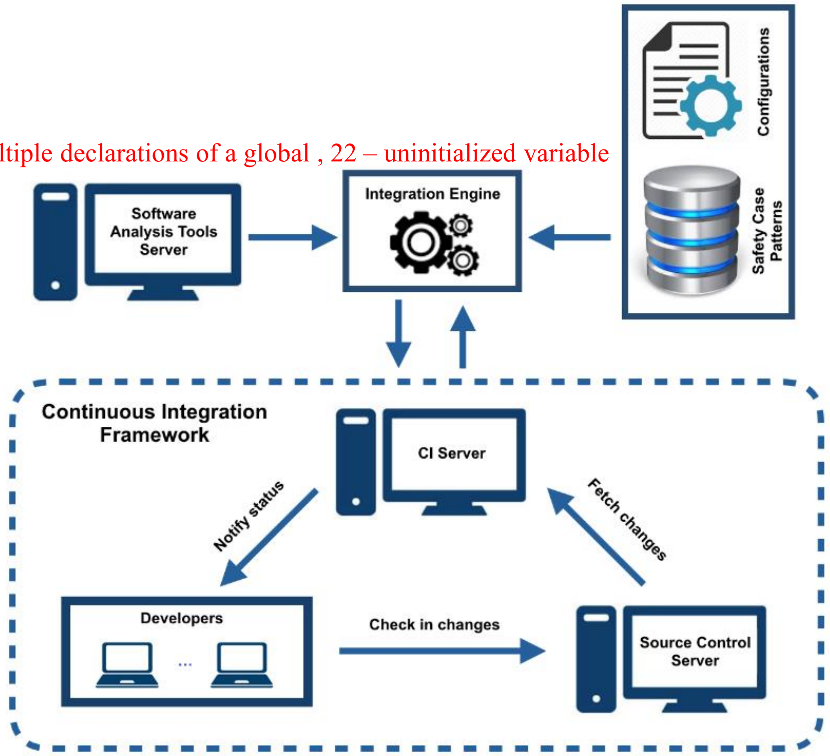
- Starting point: a set of agreed **coding guidelines**, which are **mechanically verifiable and measurably** effective (e.g. “The Power of 10: Rules for Developing Safety-Critical Code” G. Holtzmann, IEEE Computer Society "Computer" magazine))

- Use a portfolio of **static analyzers** for identifying potential defects
  - Portfolio because static analyzers are **incomplete**
  - For example, results from report about Toyota investigation by NASA:

    CodeSonar: 2272 – global variables declared with different types , 333 – case alters value , 99 – condition contains side-effect , 64 – multiple declarations of a global , 22 – uninitialized variable

    Coverity: 97 – declared but not referenced , 5 - include recursion

    Uno: 89 – possibly uninitialized variable, 2- Array of 16 Bytes initialized with 17 Bytes



- SCRUB (at JPL) - Integration of defect analysis into review process

- But static analysis is not only incomplete but also **unsound**
  - Use **model checking** and **test case generation** for eliminating **false positives**
  - In case defect is confirmed, a witness (e.g. input/trace) is generated

- Integration of **static analysis – refinement - review** cycle into **continuous integration** framework

- **Automatic generation of safety cases** from the generated evidence of analysis tools

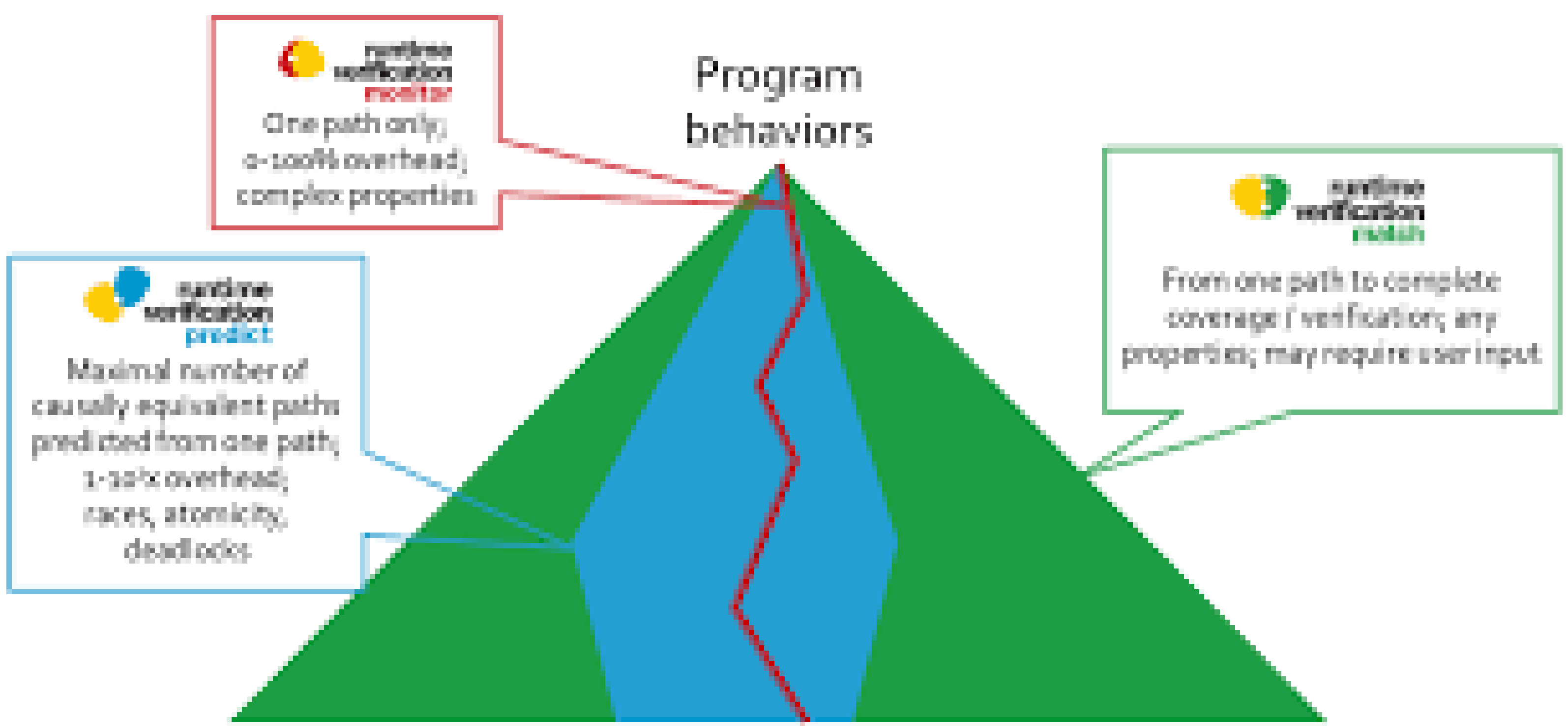
- Also: analysis of **modeling guidelines** (e.g. Savoir Autocode) but also **architecture** or **requirements** guidelines possible.

fortiss

GNC Certification Technology

Technology Nugget IV: Run-time analysis, Verification, and Certification

- Symbolic **reachability** analysis during **run-time** to guarantee (e.g. providing **certification** evidence) that at any instance where the planning algorithm makes a decision
- the system under consideration can always be maneuvered towards a safe state, within a bounded time horizon.



GNC Certification Technology

Technology Nugget V: Synthesis of Reactive Modules from Specification

Correct-by-construction generation of embedded **sense-compute-act** control software from high-level requirements

– Requirements expressed in stylized language such as EARS or linear temporal logic; e.g.

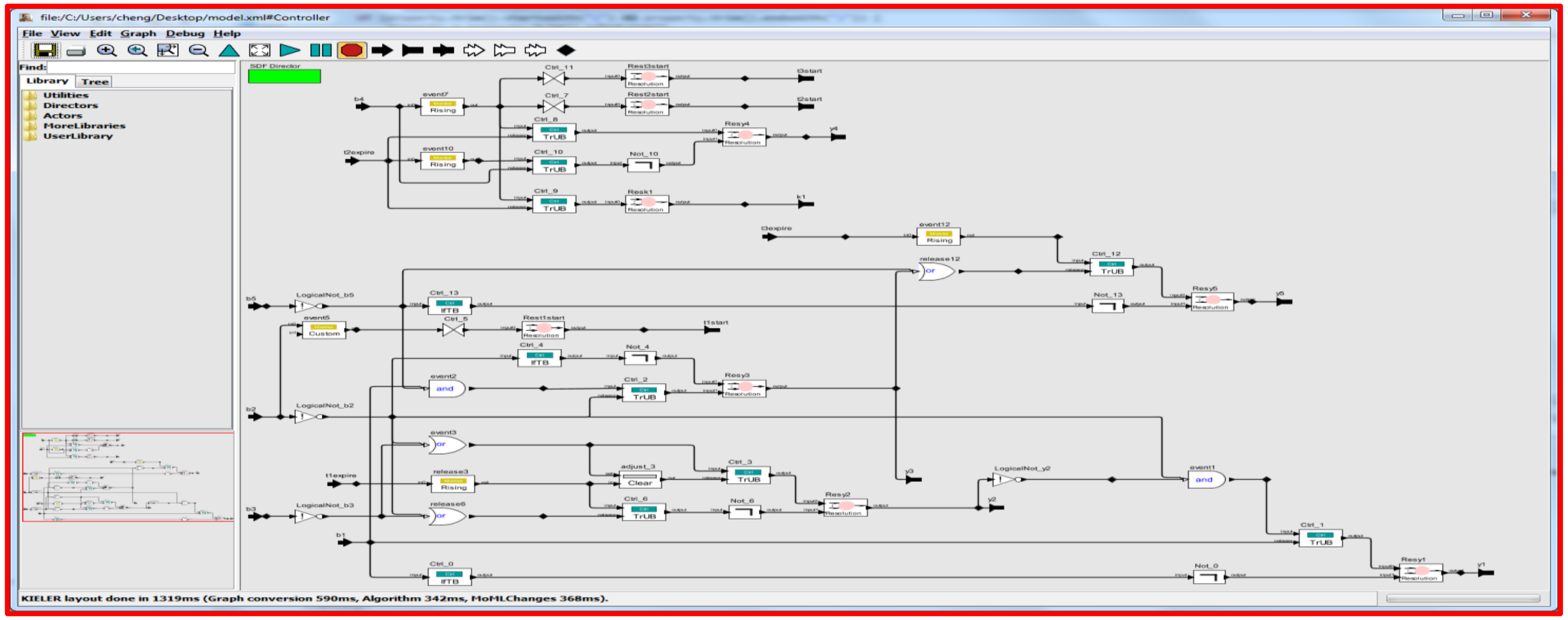
IF signal1 AND NEXT(signal2) THEN output1 UNTIL (output2 OR output3)

NEVER(output1 AND output2)

– Verified on 70+ industrial case studies (structure of synthesized code similar to hand-coded PLC programs)

– Complete traceability between requirements and synthesized code (➔ safety-critical applications)

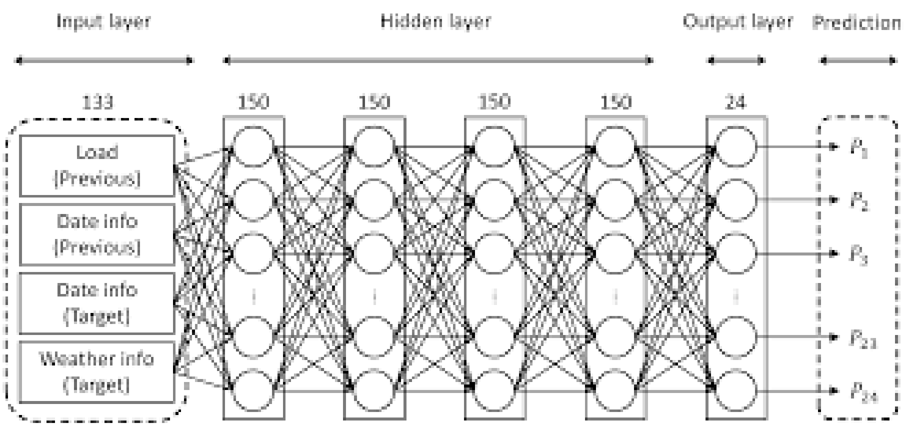
– Controllers produced in synchronous dataflow (SR/SDF in Ptolemy), statically scheduled and autocoded into programming languages such as C or structured text

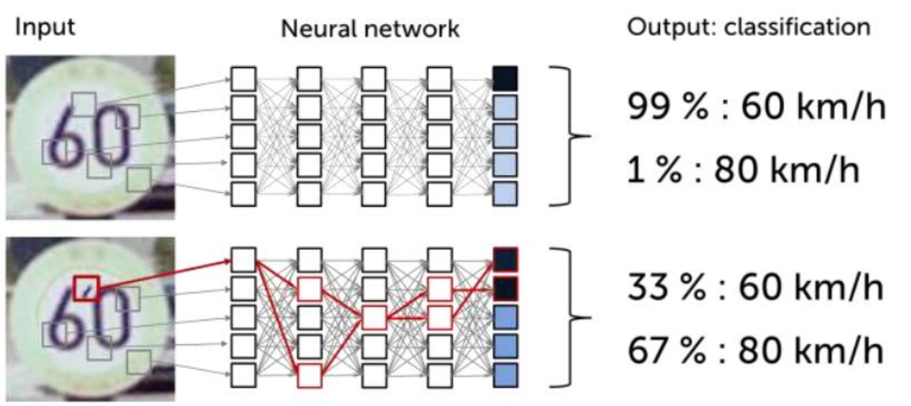


GNC Certification Technology

Technology Nugget VI: Verification of Neural Networks

- Experience-based programming based on **neural networks** increasingly propagated for perception, classification, and information fusion.
- …but: **dependable neural networks** crucial for safe and secure autonomous and decision systems
- Defining and computing **quantitative metrics** regarding how the system reacts over perturbation is a missing piece towards safety certification.
- Novel verification techniques needed for deriving characteristic measures (e.g. for resilience) and for establishing **safety properties for neural networks**





fortiss

GNC Safety Certification

Conclusions

We have outlined a methodology for **verifying and certifying autonomous GNC based on a classical 3-level architecture**

Certification methodology centered around the construction of **explicit safety cases**

- **Safety cases decompose** along vertical and horizontal structures of system design artefacts
- Integrated **safety case may guide safe and efficient system development**
- Architecture-centric approach provides opportunity for high-level **safety patterns** for substantially reducing the effort of building up safety cases
- System may **safely evolve/adapt** within the limits of the capability of adapting corresponding safety cases (also during operation)

Essential **verification and synthesis** methods are emerging; e.g.

- Integrated system and software analysis based on a portfolio of static analysis, formal analysis, automated test cases, machine learning
- Correct-by-construction synthesis of specialized functionalities in, say, FDIR (e.g. state estimation/Autofilter)
- Efficient runtime monitoring for enforcing safety objectives
- Modular Safety Cases and Safety Case Patterns

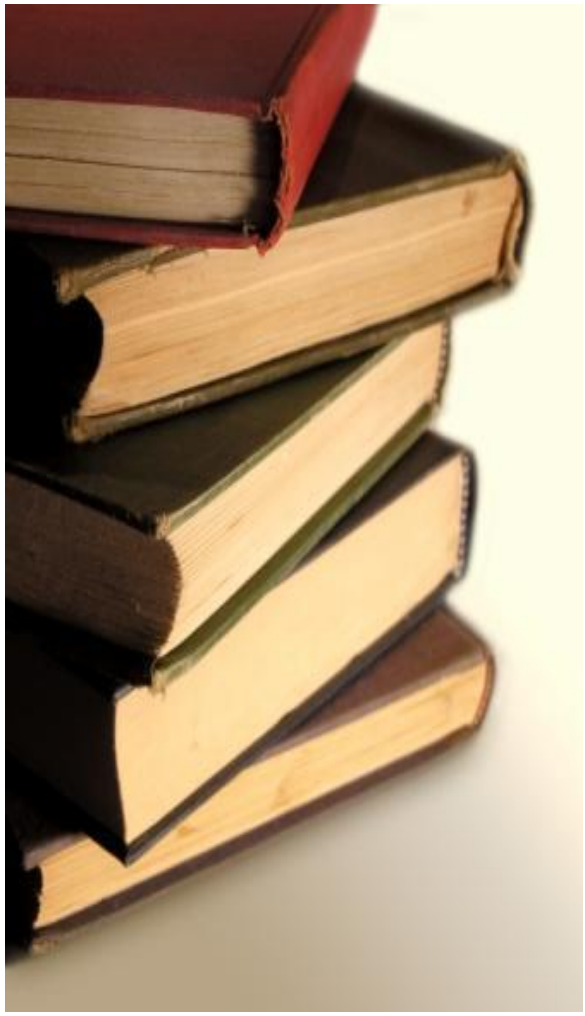
As a next step, suggested **certification methodology** needs to be fully developed and **validated** on a substantial autonomous **GNC case study.**

Substantial ongoing developments for developing large-scale safety cases, among others, in Japan (DEOS) and also in the US for producing safety cases e.g. for UCAVs.

Backup Slide

Process-based Software Safety Standards for Autonomous Systems

- Complexity and increasing requirements of avionics systems have outpaced the capabilities of current verification and certification methods
- Verification and certification based on manual reviews, process constraints, and testing are proving too expensive
  - Human inspection limited by the abilities of the reviewers.
  - Simulation and testing can only explore a minuscule fraction of the state space of any real system.
  - What does “100% code coverage” mean?
  - Typically more than half the costs for certification.



- Traditional methods cannot verify
  - adaptive control for upset recovery of aircraft,
  - intelligent control of spacecraft, <u>and</u>
  - control software for advanced military and UAVs operating in commercial airspace.

Unless safety-critical embedded software can be developed and verified with *less cost and effort* – while still satisfying the highest dependability requirements – these new capabilities may never reach the market.

fortiss

Backup Slide

Essential Features of Certification Methodology

- **Compositionality** refers to the ability to instantiate and compose **high-level certification artifacts**
  - such as safety claims and supporting evidence at the system level
  - from simpler and low-level certification artifacts at the sub-system, component or equipment level.

- **Continuity.** Automatically construct certification artifacts **at every stage of the development**
  - without a need for the development to be complete
  - safety assessment and certification at early stages of the development
  - **development process driven by safety considerations**

- **Reusability.** Analyze **impact of changes** to a system
  - during the development process
  - reusing the parts that are not affected.