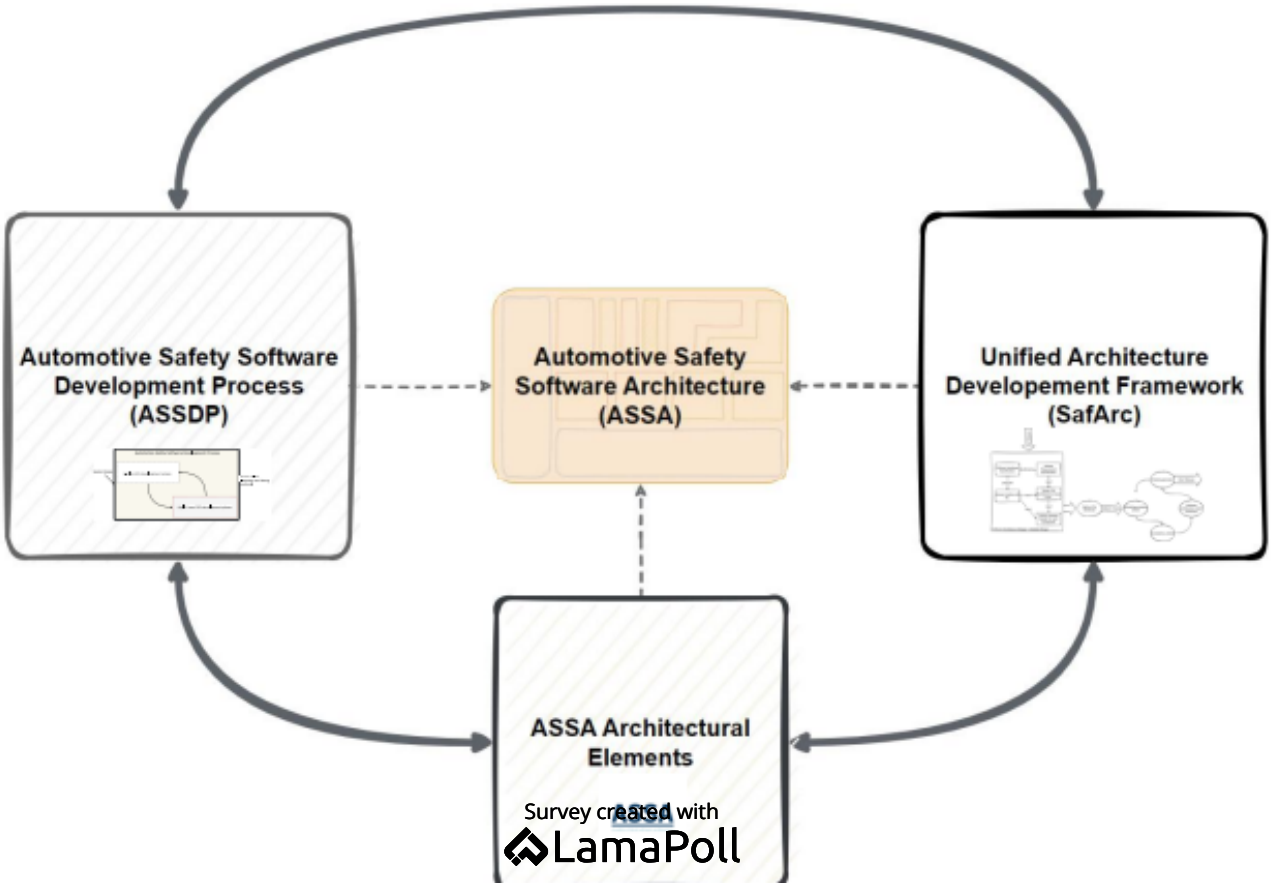
ASSA
A safety-critical software architecture

1. Welcome

Automotive Safety Software Architecture: A Fault-Tolerant Safety-Critical Software Architecture for Modern Vehicles

Abstract:

The demand for advanced software systems that focus on safety, especially in critical situations, has seen a significant rise due to the growing number of electrical and electronic systems found in road vehicles. A most important consideration for any safety-critical software system is that it is designed to be fault-tolerant and possesses the inherent capability to withstand faults through avoidance, detection, and containment measures. To tackle these challenges, several strategies are emerged, such as standardization and the incorporation of modularity, availability, and reusability in safety-critical software. The conventional architecture concepts utilized in software system design cannot be directly applied to the development of safety-critical embedded systems due to their inherent limitations in effectively addressing non-functional requirements and quality attributes such as safety and security. This paper presents a software reference architecture and development framework called Automotive Safety Software Architecture (ASSA) that effectively addresses these concerns and is specifically designed to meet the strict conditions associated with safety-critical software development.



Mohan Prabhu, V.. (2023, October 11). 32nd Aachen Colloquium Sustainable Mobility 2023. *Automotive Safety Software Architecture: A Fault-tolerant Safety-critical Software Architecture for Modern Vehicles*.

ASSA consist of three main pillars:

Automotive Safety Software Development Process (ASSDP):

It is a safety-critical software development process specifically outlined to increase the efficiency and reduce development cost and time of safety software development using DevOps methodology and reprocessing of iteration and increment-based software development approaches, not abandoning the ISO 26262 compliant requirements.

SafArc:
The existing software architecture design framework fails to address the requirements of safety quality attributes first-hand. Also, in case the software architecture framework is unable to detect the breaches of safety concerns at the architecture stage, then these errors can be seen in later development phases, which costs higher development costs and effort. Hence, ASSA proposes a unified software architecture development framework for safety-critical software development.
The goal of this framework is: to effectively address the design decisions for safety quality attributes using SW architecture views, following the safety analysis using the supplementary safety critical software architecture views, and In-time finding of architecture-related issues which may lead to errors of safety concerns using metric-based architecture evaluation. Checks involving consistency and modelling guideline in design helps in seeing the state and condition of software architecture.

ASSA Architecture Elements and their development framework:

Central Architectural Elements: These are the primary elements that form the basis of the ASSA. These elements define the general structure and organisation of software systems, ensuring the integration of functional and non-functional requirements and also supporting scalability and adaptability, not abandoning the safety requirements. This is possible by enforcing the two important central architectural elements: Safety Coordinator and Safety Libraries.

Reusable Elements: These components are specifically designed in such a way that they can be reused and are ready to connect and communicate in safety-critical SW. This implies that such software components can be utilized among various electronic control units (ECU) with less impact. The software components are pre-qualified for safety-critical development and come with a toolchain which helps configure and generate the source code and different development artefacts. E.g. reusable elements include SafIO (Safe Input Output Handler), SafCOM (Safe Communication Handler), SafFM (Safe Fault Manager) etc.

Application Specification Elements: ASSA offers comprehensive development best practices and design guidelines that are essential for the creation and execution of application-specific architectural components. These components can be created through either traditional manual coding or safety-oriented model-based software engineering approaches. These elements can be classified into two types:

1. Powertrain-specific SWC, which can be reused within a particular type of powertrain element,
2. Product-specific components are non-reusable and are designed for specific functionality.

★ **Please select a continent you are currently working in.**

- [ ] Europe
- [ ] Asia
- [ ] Africa
- [ ] North America
- [ ] South America
- [ ] Australia/Oceania
- [ ] Antarctica

★ **What domain of safety-critical system have worked in ?**

- ◯ Nuclear Sector (IEC 61513)
- ◯ Railway Application (EN 50128)
- ◯ Process industry (IEC 61511)
- ◯ Automotive (ISO 26262)
- ◯ Machinery (IEC 62061)
- ◯ Medical (IEC60601)
- ◯ Other

★ **Please indicate your role in the current organisation.**

- ◯ Functional Safety Manager
- ◯ Project Manager
- ◯ Software Architect
- ◯ Functional Safety Software Expert
- ◯ Software Developer/Tester
- ◯ Devops Engineer
- ◯ Other

How many years of experience do you have?

- [ ] <2 Years
- [ ] 2-5 Years
- [ ] 5-10 Years
- [ ] >10 Years

★ **Are you working or have you previously worked in safety critical software systems (Functional safety software systems)?**

◯ Yes ◯ No

★ **Are you certified Safety Engineer?**

◯ Yes ◯ No

2. ASSA Survey

What is the highest ASIL level safety-critical software system you have developed in your previous experience?

Note: ASIL: Automotive Safety Integrity Level similar to SIL

- [ ] ASIL D
- [ ] ASIL C
- [ ] ASIL B
- [ ] ASIL A

What is the highest level of ASIL Software components you are willing to reuse?

- [ ] ASIL D
- [ ] ASIL C
- [ ] ASIL B
- [ ] ASIL A

How many third-party libraries and SWC have you introduced in your project?

- [ ] <1
- [ ] 2-5
- [ ] >5

What are the main challenges that you have encountered during Safety critical software development?

Note: For the automotive domain, this is product development at the software level in accordance with ISO 26262:2018 - part 6

Text field

What are the general challenges that are faced for product development at the software level in safety-critical software development?

Note: For Automotive domain, this is product development at software level in accordance to ISO 26262:2018 - part 6

Text field

What are the challenges that are faced at requirement specification for product development at the software level in safety-critical software development?

Note: For the automotive domain, this is product development at the software level in accordance with ISO 26262:2018 - part 6

Text field

What are the challenges that are faced at software architectural design process for product development at the software level in safety-critical software development?

Note: For the automotive domain, this is product development at the software level in accordance with ISO 26262:2018 - part 6

Text field

What are the challenges that are faced in the software unit design and implementation process for product development at the software level in safety-critical software development?

Note: For the automotive domain, this is product development at the software level in accordance with ISO 26262:2018 - part 6

Text field

What are the challenges that are faced at the software unit verification process for product development at the software level in safety-critical software development?

Note: For the automotive domain, this is product development at the software level in accordance with ISO 26262:2018 - part 6

Text field

★ **What are the challenges that are faced in software integration & verification, and testing of the embedded software process for product development at the software level in safety-critical software development?**

Note: For the automotive domain, this is product development at the software level in accordance with ISO 26262:2018 - part 6

Text field

★ **What are current challenges that you face in safety-critical software development ?**

- [ ] Emergence of new features and safety requirements
- [ ] Newer safety standards and increasing safety requirements
- [ ] Interaction of Safety with other safety qualities like: Security
- [ ] Other

★ **What is the software development method used in your organisation for safety-critical software systems?**

- [ ] Agile
- [ ] Spiral
- [ ] Waterfall Model
- [ ] Iterative and incremental
- [ ] DevOps
- [ ] V-Model
- [ ] Other

★ **Have you experienced challenges in uplifting existing software components and their source code to higher ASIL levels within your organization?**

- ◯ Yes
- ◯ No

★ **Are you using any in-house developed reusable safety critical software elements (e.g. components) in your organization?**

- ◯ Yes
- ◯ No
- ◯ Maybe

If Yes, then at what level of the AUTOSAR SW layer?

- [ ] Application
- [ ] Service Layer
- [ ] ECU abstraction Layer
- [ ] Microcontroller abstraction Layer
- [ ] Complex device driver
- [ ] Other

How often do you encounter the need for reusable safety-critical software components in your organization?

Rate from **1 = not very often** to **5 = very often**

( ) ( ) ( ) ( ) ( )

Are you willing to introduce development and architecture framework into your project to enhance efficiency in terms of effort and cost?

( ) Yes ( ) No

Do you believe that adopting reference architecture and development frameworks like ASSA for safety-critical SW development may help increase efficiency in terms of cost and effort in your Organisation?

( ) Yes ( ) No

If Yes, then what are the challenges you think you will face while adopting such a reference architecture and framework into your organisation

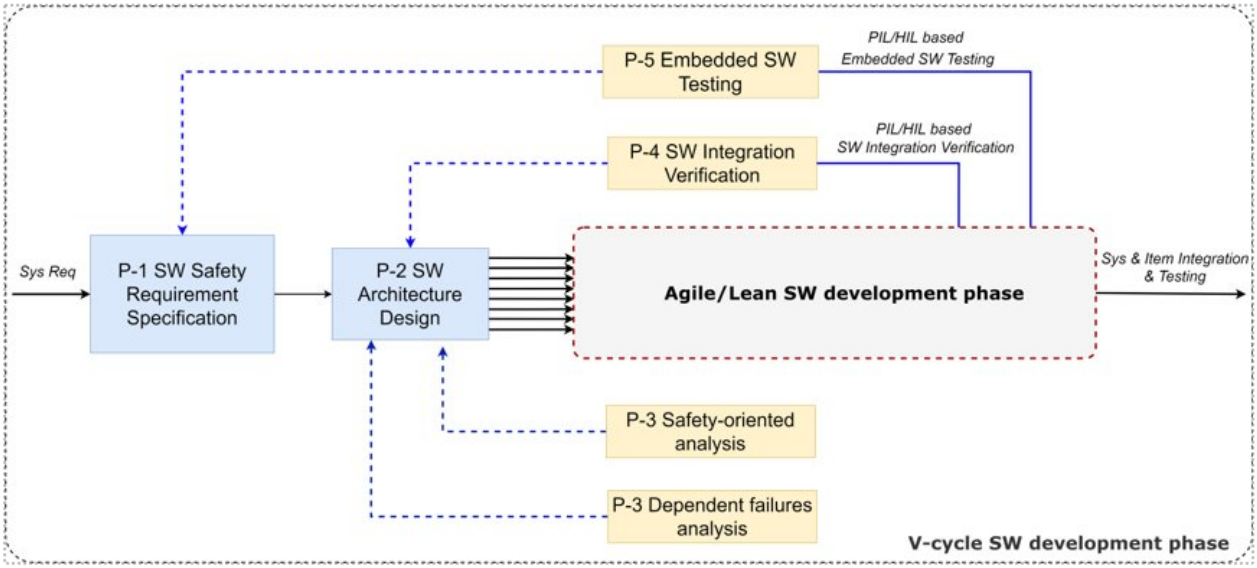
- [ ] Legacy system compatibility
- [ ] Regulatory complaince
- [ ] Cultural resistance
- [ ] Skill and training gaps
- [ ] Interoprability and ecosystem
- [ ] Performance and optimisation
- [ ] Other

How well do you believe ASSA aligns with your organization's long-term Technology strategy for safety-critical software development?

Rate from **1 = not very well** to **5 =very well**

○ ○ ○ ○ ○

How helpful is to have an DevOps driven safety-critical software development process which efficently strikes a good balance between initial upfront design (using Plan driven method) followed by lightweight evolutionary implementation & integration (Agile) yet compliant to safety standards.?



Mohan Prabhu, V.. (2023, October 11). 32nd Aachen Colloquium Sustainable Mobility 2023. *Automotive Safety Software Architecture: A Fault-tolerant Safety-critical Software Architecture for Modern Vehicles*.

Rate from **1 = not helpful** to **5 = very helpful**

○ ○ ○ ○ ○

Do you use DevOps principles like CICD in your organisation for Safety-critical software development to enhance the development process?

In software engineering, CI/CD or CICD is the combined practice of continuous integration and continuous delivery or, less often, continuous deployment. They are sometimes referred to collectively as continuous development or continuous software development.

○ Yes ○ No

How important is a CICD that is complaint to safety standard for efficient safety critical SW development?

Rate from **1 = not important** to **5 = very important**

○ ○ ○ ○ ○

Do you believe that CI can streamline safety-critical software development by providing key advantages as listed?

1. **Regular integration** (CI automatically creates new software builds, testing, and integration when source code updates are detected).
2. **Prevention of discrepancies** (CI helps prevent software code and design discrepancies, ensuring that the software remains consistent and functional).
3. **Decreased Manual testing workload** (CI automates the testing process, reducing the need for manual testing and saving time and effort .)
4. **Enhanced collaboration** (CI facilitates collaboration among geographically separated teams by providing a centralized platform for code integration and communication .).
5. **Early defect identification** (CI enables early identification and mitigation of defects by running automated tests and providing immediate feedback on the quality of the code .).
6. **Efficient merging of code modifications** (CI reduces the time and resources required for merging various code modifications, making the integration process more efficient).
7. **Geographically independent teams can efficiently collaborate**

○ Yes ○ No

Have you experienced benefits in terms of defect prevention and early defect identification through CI in safety-critical development?

○ Yes ○ No

Do you believe it is important to integrate CI in your safety-critical software development workflow?

○ Yes ○ No

How important is the compatibility of ASSA's development framework and reference architecture with existing software development tools and processes in your projects?

Rate from **1 = not important** to **5 = very important**

○ ○ ○ ○ ○

Is it helpful to have the standard requirements of reuable SW components which can be reused?

○ Yes ○ No

★ **Is it helpful to have a requirement development framework for safety critical SW development?**

Example: frameworks which are build based on EARS

◯ Yes ◯ No

★ **What architecture modelling tool are you using ?**

☐ Enterprise Architect

☐ Raphsody

☐ MagicDraw

☐ Others

★ **Does your organisation use any architectural frameworks for safety-critical software?**

Note: Architectural framework refers to a structured approach or set of guidelines that provides a foundation for designing and organizing the architecture of a software system. It helps in addressing various aspects of the system, such as functional and non-functional requirements, scalability, adaptability, and safety concerns.

◯ Yes ◯ No

★ **Are you familiar with metric-based software architecture evaluation and its benefits, as discussed below ?**

Note: Metric-based software architecture evaluation involves using quantitative measures to assess the quality and performance of a software architecture. It helps in identifying potential issues, evaluating adherence to design guidelines, and ensuring the achievement of desired quality attributes.
The benefits of metric-based software architecture evaluation include:

1. Early detection of architecture issues that could lead to safety concerns or violations.
2. Continuous evaluation of the architecture's metrics allows for proactive identification and resolution of potential problems.
3. Improved communication of design decisions related to safety quality attributes through software architecture views.
4. Accompanying safety analysis through additional safety-critical software architecture views.
5. Ensuring consistency in design through consistency and modelling guideline checks.
6. Generating important software artefacts and visualizing the state and health of the software architecture for better overview and understanding.

◯ Yes ◯ No

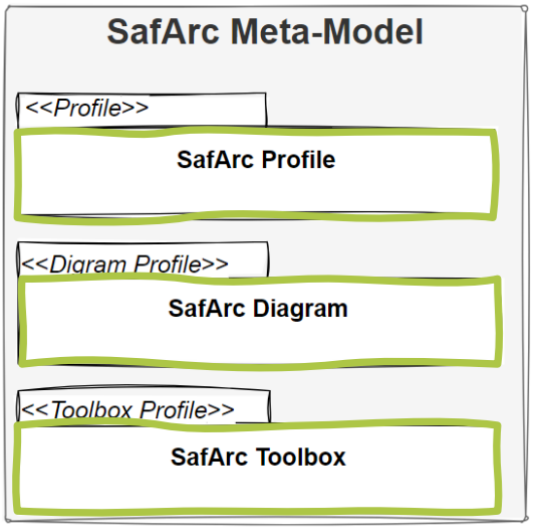
★ **In your organisation, from the CICD platform, are you able to map the generated artefacts to the individual work products from your safety-critical process?**

Note: For the automotive domain, this is product development at the software level in accordance with ISO 26262:2018 - part 6

◯ Yes ◯ No

Do you believe having a metamodel which includes the below points would help you better develop safety-critical software in organisations and projects?

1. One unified meta-model for software architecture and detailed design.
2. An elements profile consisting of the stereotypes used in modelling the architecture based on UML 2.
3. A predefined folder structure for different architecture views (UML diagrams) and defined architecture viewpoints that address the concerns of the different stakeholders like safety, security etc.
4. Toolbox, which provides the developer with a predefined set of diagrams and attributes that can be selected for the development of software architecture and detailed design

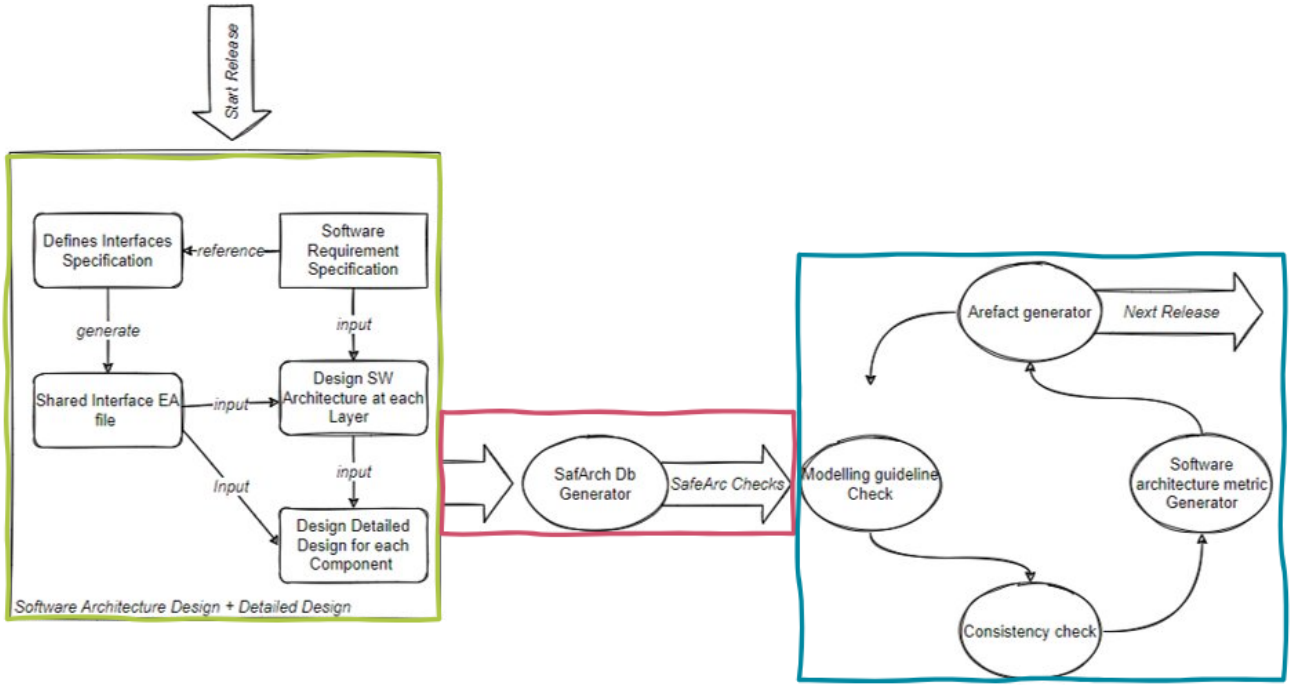


Mohan Prabhu, V.. (2023, October 11). 32nd Aachen Colloquium Sustainable Mobility 2023. *Automotive Safety Software Architecture: A Fault-tolerant Safety-critical Software Architecture for Modern Vehicles*.

Note: A metamodel is a model that defines the structure, constraints, and relationships of other models within a specific domain, providing a formal description of the elements, properties, and relationships that can exist in a model.

◯ Yes ◯ No

Does having an SW architecture tool like SafArc checker from ASSA help you develop SW architecture and detail design which is consistent, traceable and adheres to safety standards?



Mohan Prabhu, V.. (2023, October 11). 32nd Aachen Colloquium Sustainable Mobility 2023. *Automotive Safety Software Architecture: A Fault-tolerant Safety-critical Software Architecture for Modern Vehicles*.

In this context, the SafArc toolchain ensures software architecture design consistency, adherence to guidelines, and continuous evaluation. It includes the SafArc DataBase generator, which extracts relevant information from architecture models to generate databases, and the SafArc Checker, which uses the generated DataBase to establish compliance with ASSA modelling guidelines, ensure design consistency between software architecture and detail design, continuous evaluation of software architecture using metric-based software architecture evaluation and helping generate important artefacts like software architecture design specification, detailed design specification, AUTOSAR arxml file etc.

◯ Yes ◯ No

Do you believe reusable architectural elements and central architectural elements defined by ASSA will help you increase effeciency in your project?

◯ Yes ◯ No

Is it useful to have a development framework to design, manage and deploy reusable elements in your organisation and project?

Note: For the automotive domain, the software elements can be qualified for reusability in accordance with ISO 26262:2018 Part-8 clause 12 and developed according to ISO 26262:2018 - part 6

Yes No

★ **Is it helpful to have the possibility to run the SafArc tool in CI, which performs the following?**

1. Modelling guideline check
2. Consistency check
3. Artefact generation
4. Continuous architecture evaluation

◯ Yes ◯ No

★ **What is the highest level of ASIL you have developed in your experience?**

☐ ASIL A

☐ ASIL B

☐ ASIL C

☐ ASIL D

★ **In your project, are you developing mixed-critical systems?**

◯ Yes ◯ No

★ **If Yes, what are the challenges faced during the development of mixed-critical safety systems?**

☐ Spatial isolation

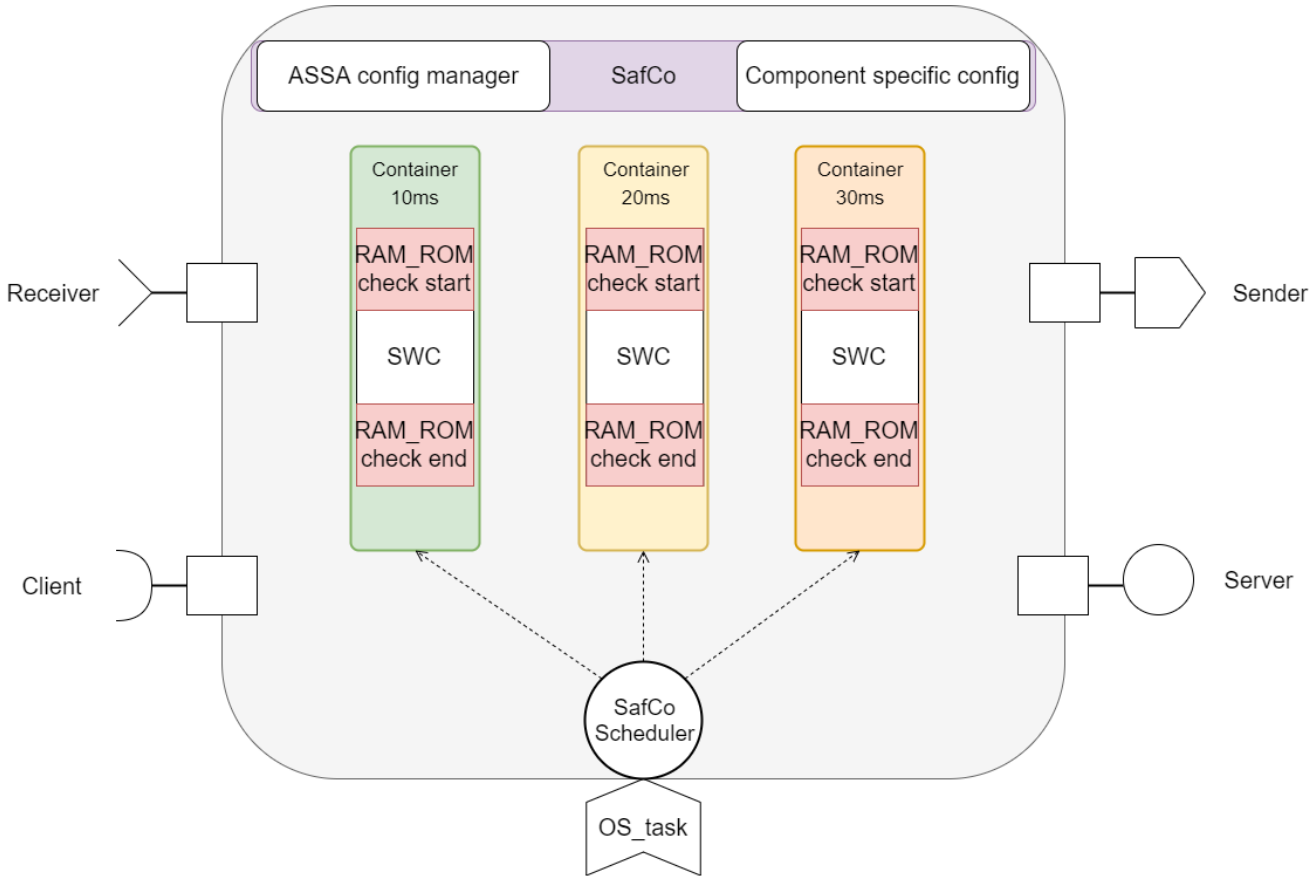
☐ Temporal isolation

☐ Common cause failure

☐ Freedom from interference

☐ Other

Do you think having a software element such as SafCo will be helpful for a mixed-critical SW system?



Note: The main functionality of SafCo is it is a simple orchestrator and as SWC to coordinate and encapsulate all software components inside ASSA. Another objective of SafCo is to handle all configurations and settings relevant to a particular ASSA instance which includes component specific configurations and overall configurations. SafCo takes care of optionally scheduling all software components inside ASSA through its internal. Using an internal scheduler ensures equal priority for multitask runnables in a specific ASIL level and prevents priority inversion caused by different rates or lower ASIL runnables. Multiple SafCos can be used in mixed-criticality systems to increase determinism and ensure independence.

◯ Yes ◯ No

Is it helpful to have a safety complaint C-library which is tested and verified in accordance to the safety standard in reducing the effort of Safety critical SW development

Note: For the automotive domain, the software elements can be qualified for reusable in accordance with ISO 26262:2018 Part-8 clause 12 and developed according to ISO 26262:2018 - part 6

◯ Yes ◯ No

Do you believe that constant use of reusable architectural elements that are developed according to ISO 26262:2018-part 8 clause 12 following ASIL D requirements can improve software quality, usability and maintainability within your organisations and projects?

◯ Yes ◯ No

Is it helpful to have a reusable communication handler with a dedicated toolchain that supports different communication protocols (e.g. CAN, LIN, and Ethernet) that helps you configure and generate code that adheres to safety standard complaints?

Note: For the automotive domain, the software elements can be qualified for reusability in accordance with ISO 26262:2018 Part-8 clause 12 and developed according to ISO 26262:2018 - part 6

◯ Yes ◯ No

Is it helpful to have a reusable input output handler with a dedicated toolchain that helps you configure and generate code that adheres to safety standard complaints?

Note: For the automotive domain, the software elements can be qualified for reusability in accordance with ISO 26262:2018 Part-8 clause 12 and developed according to ISO 26262:2018 - part 6

◯ Yes ◯ No

Is it helpful to have a reusable safety-specific fault manager that helps you configure and generate code that adheres to safety standard complaints?

Note: For the automotive domain, the software elements can be qualified for reusability in accordance with ISO 26262:2018 Part-8 clause 12 and developed according to ISO 26262:2018 - part 6

◯ Yes ◯ No

Do you think having a development framework which provides best practices and development guidelines for the design and implementation of application-specific architectural elements is useful?

Note: Application-specific element framework includes ASSA's software architecture development framework and toolchain (SafArc), a set of design/modelling guidelines and a toolchain for model-based SW development for the design and development of application-specific elements, which is based on Mathworks reference workflow following the ISO26262:2018 requirements etc

◯ Yes ◯ No

Is it helpful to have pre-defined test specifications, test environments and test case templates for reusable elements that can help increase efficiency and reduce effort when it comes to Unit testing and Component testing?

◯ Yes ◯ No

Is it beneficial to have the possibility to run unit testing and component testing automatically on CICD pipelines while being safety complaints?

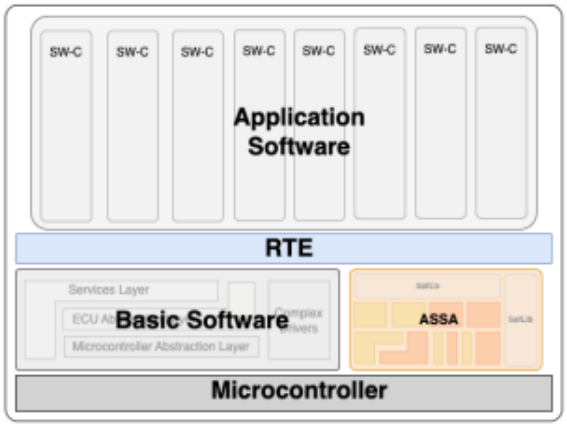
◯ Yes ◯ No

★ **What are the different standards that ASSA should comply with?**

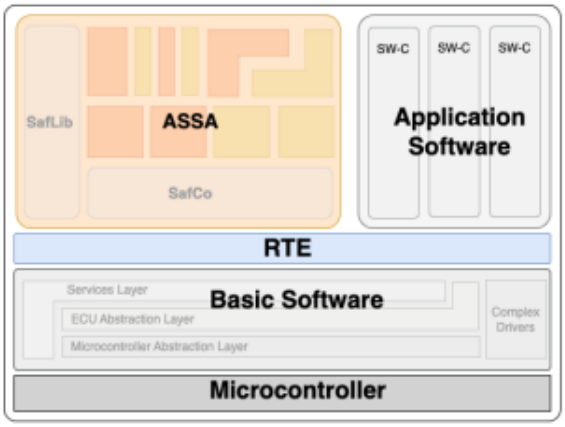
- [ ] AUTOSAR - AUTomotive Open System ARchitecture
- [ ] EGAS(electronic gas pedal) concept (Three layer monitoring system)
- [ ] ASPICE -Automotive Software Process Improvement Capability dEtermination
- [ ] Others

★ **How important is it to is for the reference architecture and its elements to be configurable to different AUTOSAR configurations without violating Safety standards**

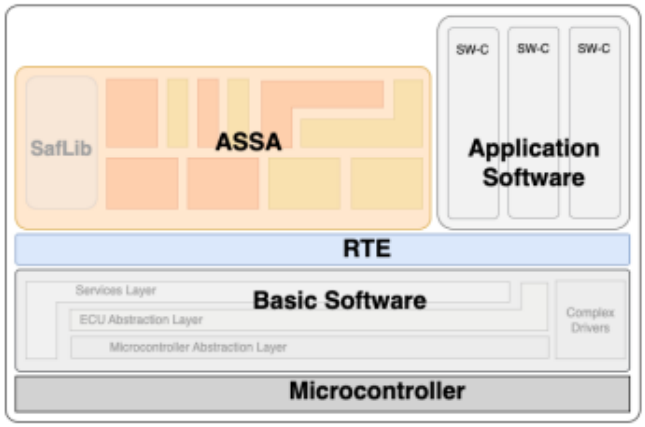
Case 1: ASSA as CDD



Case 2: ASSA as SW-C



Case 3: ASSA as SW-C without SafCo schedular



Mohan Prabhu, V.. (2023, October 11). 32nd Aachen Colloquium Sustainable Mobility 2023. *Automotive Safety Software Architecture: A Fault-tolerant Safety-critical Software Architecture for Modern Vehicles*.

Rate from **1 = not important** to **5 = very important**

○ ○ ○ ○ ○

How important is it for such a reference architecture to support AUTOSAR and Non-AUTOSAR?

Rate from **1 = very bad** to **5 = very good**

3. Thank you!

Thank you for your participation!

We want to thank you for your interest in our survey.

If you would like to see the results of this survey
don't hesitate to write a short email to *prabhu@fev.io or sbiradar@studenten.hs-bremerhaven.de*

Sincerely
Your Prabhu, Vinod Mohan/ Sharanabasaveshwar Biradar

Please give us feedback...