QORIX

Certifying the Uncertifiable?

How Eclipse S-CORE Makes Open Source Fit for Safe Mobility

Dr. Nico Hartmann
Qorix GmbH

3/6/26

CAN BOTH BE RIGHT AT THE SAME TIME?

The Core Conflict

- Code proves value.
- The community decides.
- Move fast. Iterate. Just build it.

- Code proves nothing without evidence.
  If it’s not compliant, it doesn’t ship.
- Who takes liability?
- Show me the safety case.

- **How do we combine both sides – without killing either?**

INTRODUCING ECLIPSE S-CORE

Where Open Source Meets Automotive Safety

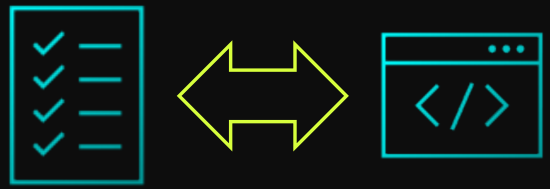
- Open-source core middleware for safety-critical SDV domains
- Developed under the Eclipse SDV Working Group
- Code-first. Modular. Hardware-agnostic.
- Industry-backed collaboration (OEMs, Tier-1s, Tech players)

Eclipse S-CORE

THE REALITY BEHIND OPEN SOURCE

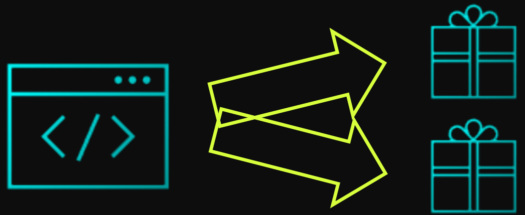
Why Open-Source?

- Code is standard
  - ⚡Where is the specification?
  - ⚡Where is the validation?



- Collaboration
  - ⚡Code compliance?
  - ⚡Staff qualification?

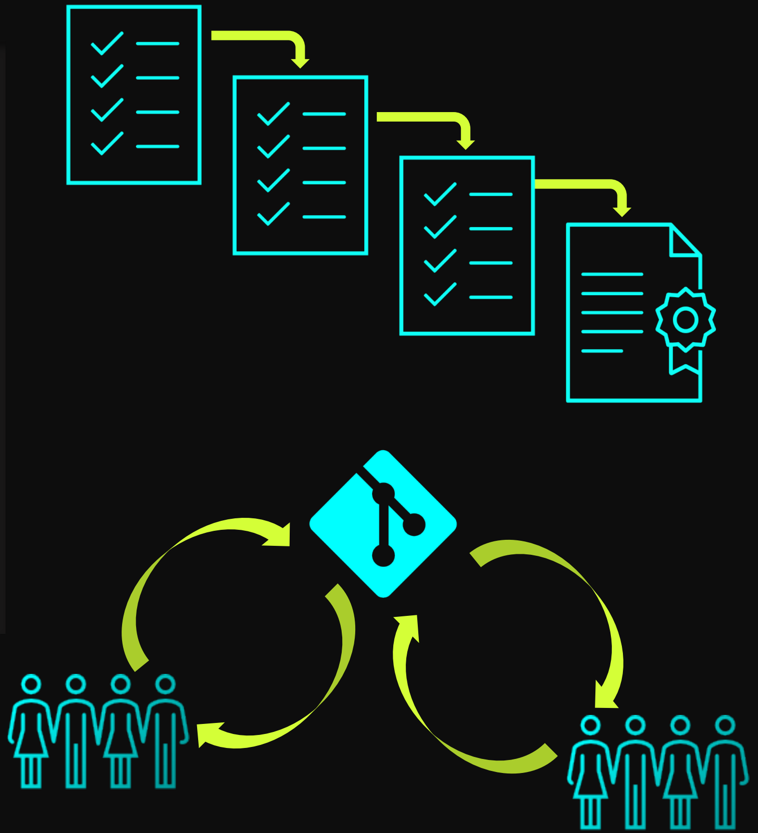
- Adoption
  - ⚡Liability?
  - ⚡Safety certification?



Safety in Open Source Dilemma

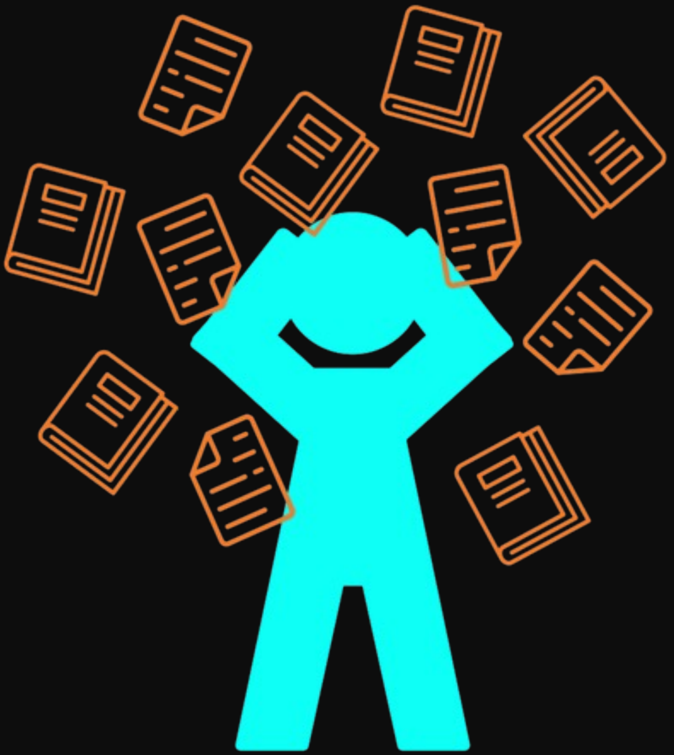
- Forcing a frozen certification model onto a living system creates friction, not safety.

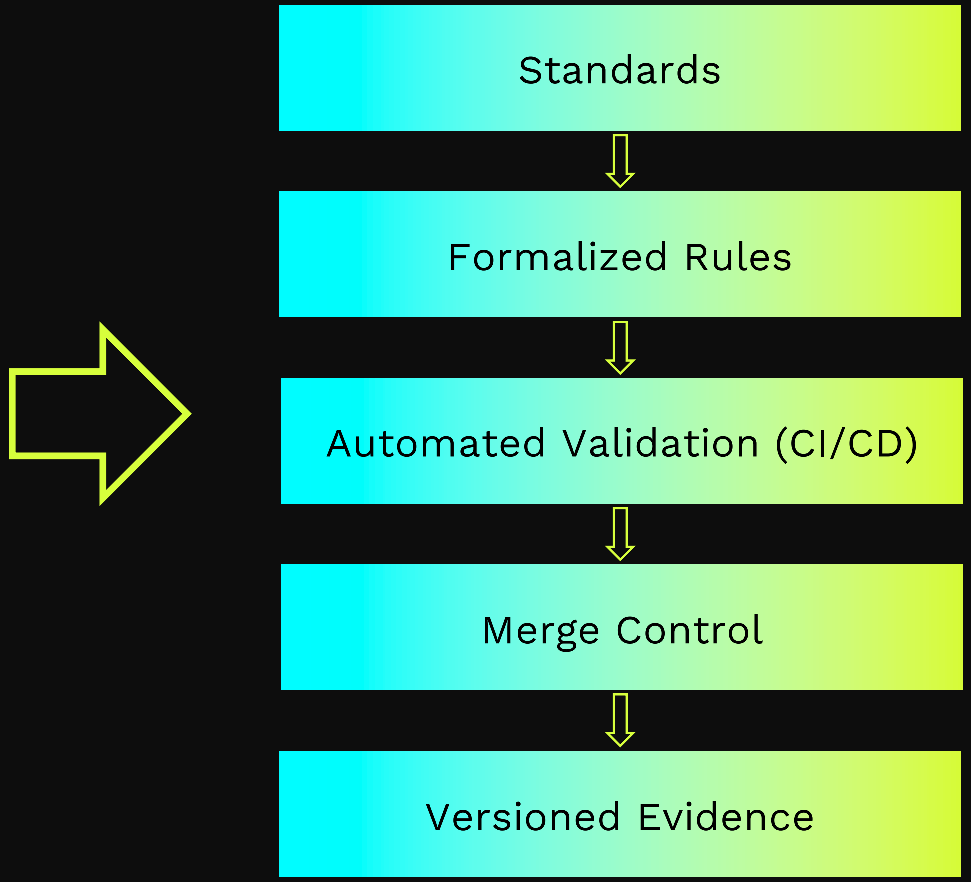
- Open-source ecosystems evolve continuously and collaboratively.



OPERATIONALIZING STANDARDS

Compliance by Interpretation vs. Compliance by Infrastructure





Standards

Formalized Rules

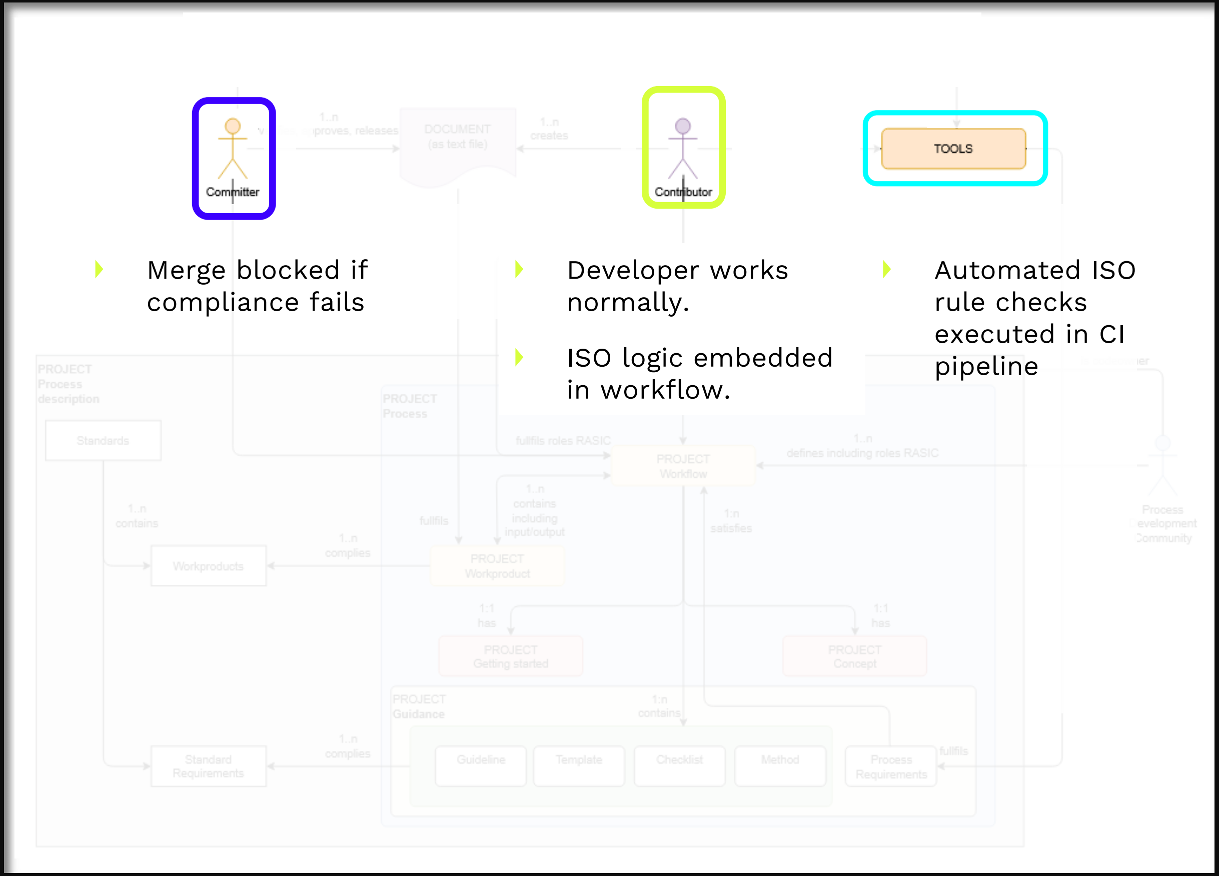
Automated Validation (CI/CD)

Merge Control

Versioned Evidence

EXECUTABLE COMPLIANCE IN PRACTICE

Compliance by Infrastructure – Not by Interpretation



- Merge blocked if compliance fails

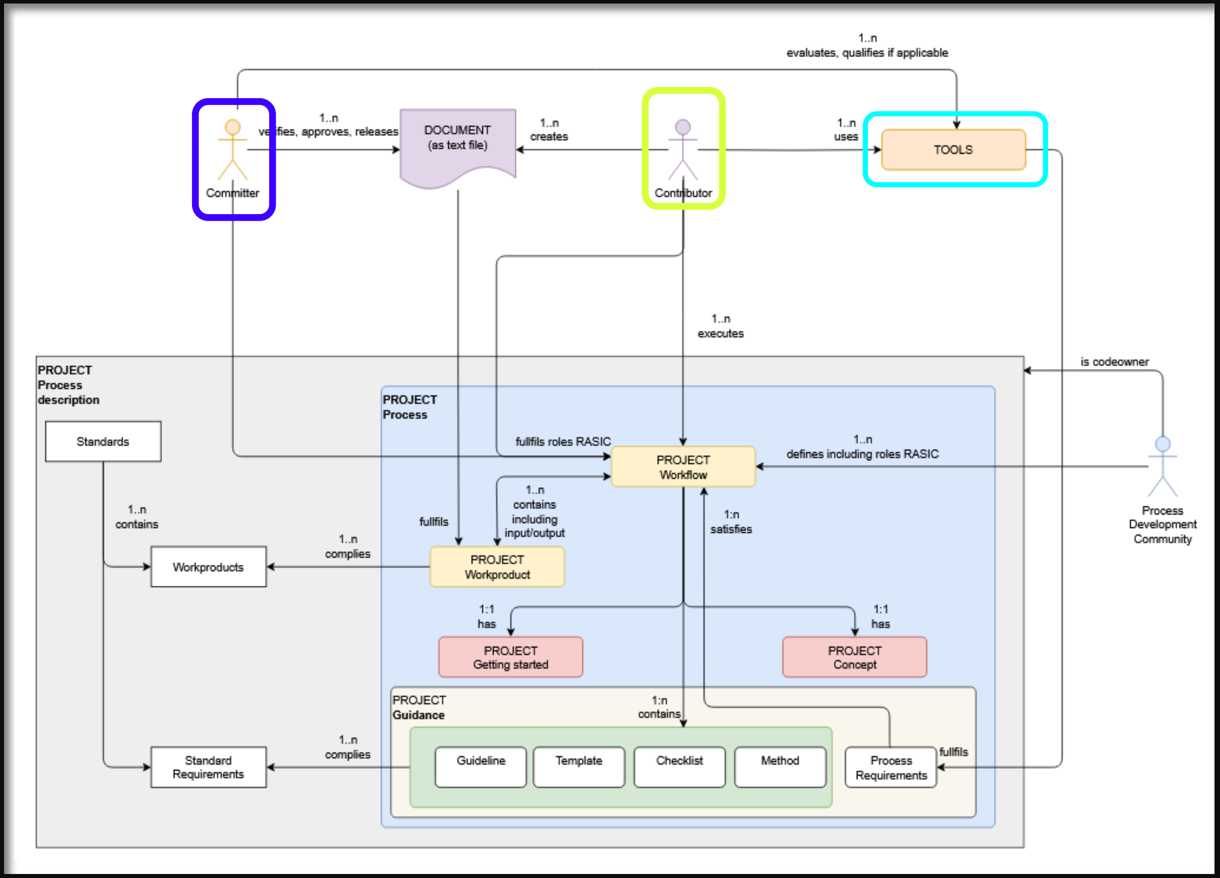
- Developer works normally.
- ISO logic embedded in workflow.

- Automated ISO rule checks executed in CI pipeline

https://eclipse-score.github.io/process_description/main/introduction/index.html

OPERATIONALIZING ISO 26262

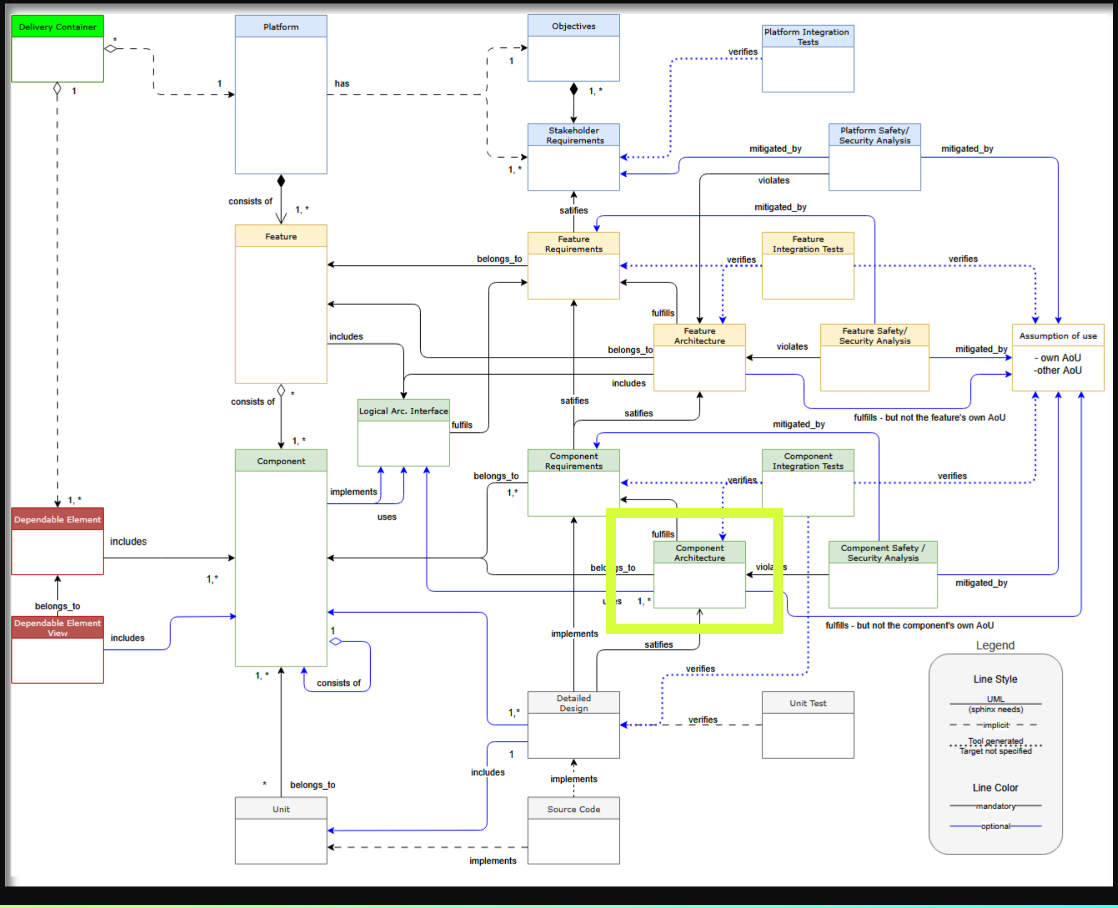
Compliance by Infrastructure – Not by Interpretation



https://eclipse-score.github.io/process_description/main/introduction/index.html

FROM REQUIREMENTS TO EVIDENCE – FULLY LINKED

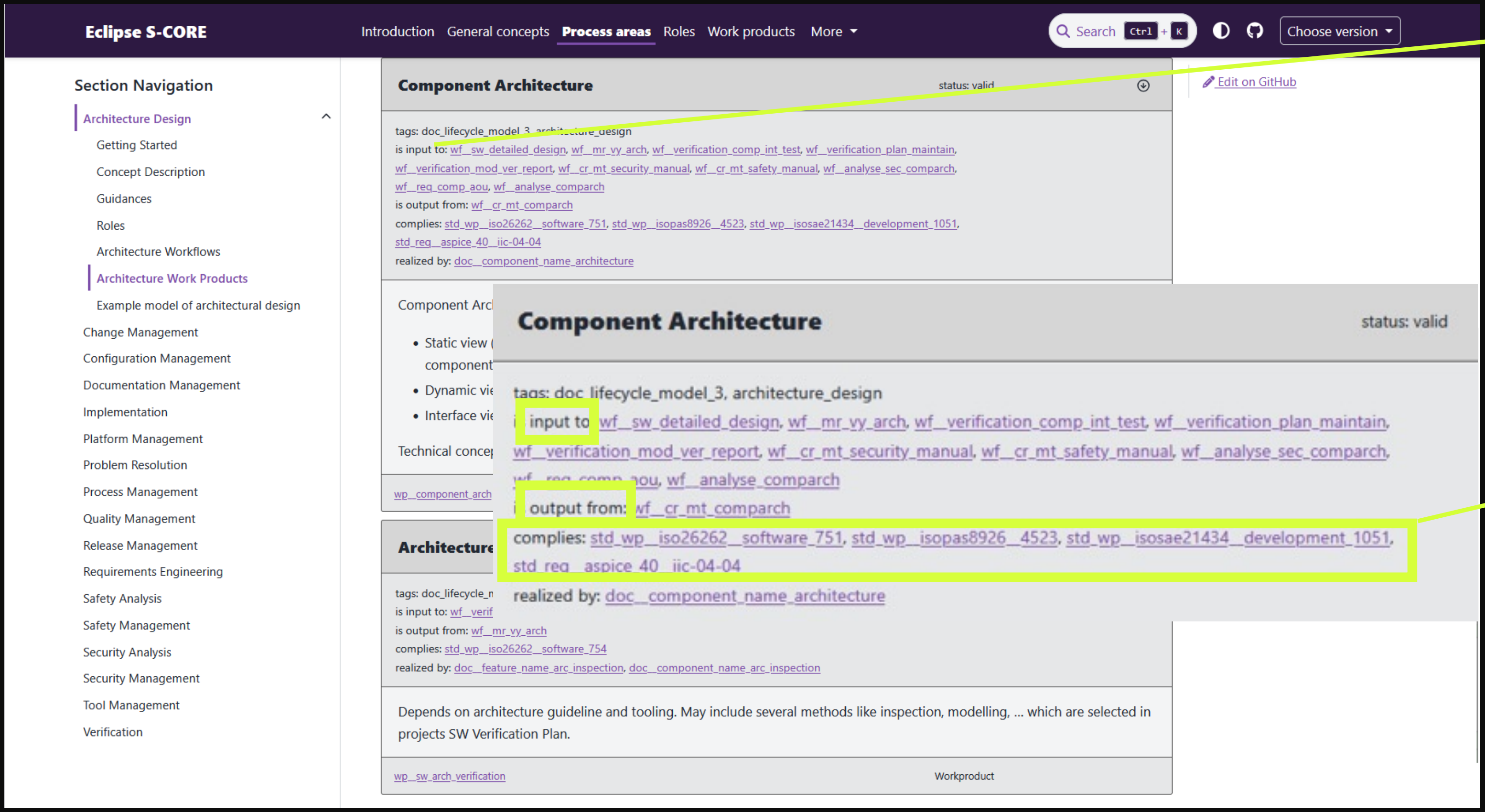
Compliance as a Traceable System Graph



https://eclipse-score.github.io/process_description/main/general_concepts/score_building_blocks_concept.html

AND IT WORKS: CODE IS COMPLIANCE

Formalized Rules



Section Navigation

- Architecture Design
  - Getting Started
  - Concept Description
  - Guidances
  - Roles
  - Architecture Workflows
  - Architecture Work Products
  - Example model of architectural design
- Change Management
- Configuration Management
- Documentation Management
- Implementation
- Platform Management
- Problem Resolution
- Process Management
- Quality Management
- Release Management
- Requirements Engineering
- Safety Analysis
- Safety Management
- Security Analysis
- Security Management
- Tool Management
- Verification

Component Architecture status: valid

tags: doc_lifecycle_model_3, architecture_design
is input to: wf__sw_detailed_design, wf__mr_vy_arch, wf__verification_comp_int_test, wf__verification_plan_maintain, wf__verification_mod_ver_report, wf__cr_mt_security_manual, wf__cr_mt_safety_manual, wf__analyse_sec_comparch, wf__req_comp_aou, wf__analyse_comparch
is output from: wf__cr_mt_comparch
complies: std_wp__iso26262__software_751, std_wp__isopas8926__4523, std_wp__isosae21434__development_1051, std_req__aspice_40__iic-04-04
realized by: doc__component_name_architecture

Component Arc

- Static view (
  component
- Dynamic vie
- Interface vie

Technical concep

wp__component_arch

Component Architecture status: valid

tags: doc_lifecycle_model_3, architecture_design
is input to wf__sw_detailed_design, wf__mr_vy_arch, wf__verification_comp_int_test, wf__verification_plan_maintain, wf__verification_mod_ver_report, wf__cr_mt_security_manual, wf__cr_mt_safety_manual, wf__analyse_sec_comparch, wf__req_comp_aou, wf__analyse_comparch
is output from: wf__cr_mt_comparch
complies: std_wp__iso26262__software_751, std_wp__isopas8926__4523, std_wp__isosae21434__development_1051, std_req__aspice_40__iic-04-04
realized by: doc__component_name_architecture

Architecture

tags: doc_lifecycle_m
is input to: wf__verif
is output from: wf__mr_vy_arch
complies: std_wp__iso26262__software_754
realized by: doc__feature_name_arc_inspection, doc__component_name_arc_inspection

Depends on architecture guideline and tooling. May include several methods like inspection, modelling, ... which are selected in projects SW Verification Plan.

wp__sw_arch_verification Workproduct

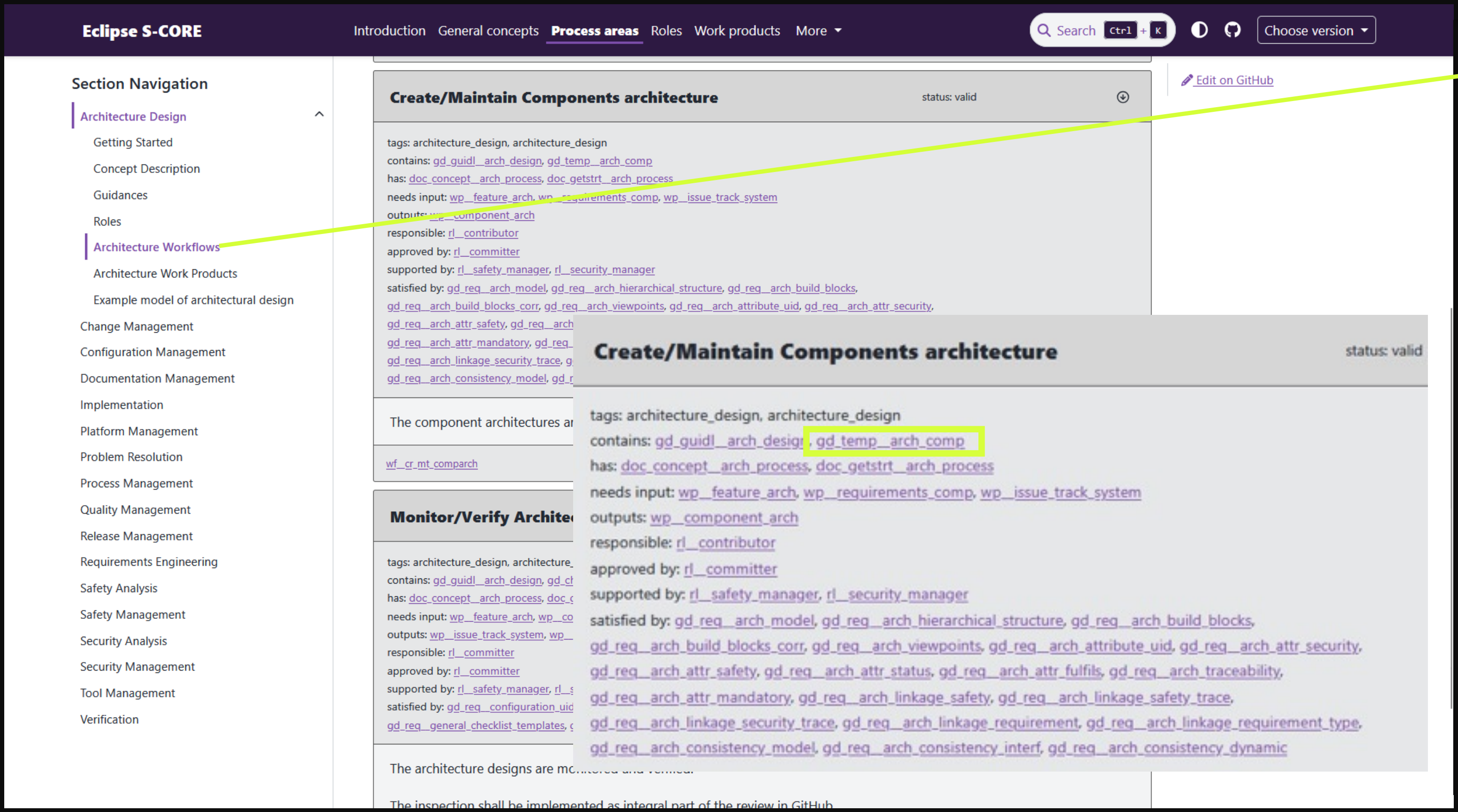
Templates In-/

Output is traceable

Work products show compliance acc. to standard requirements

AND IT WORKS: CODE IS COMPLIANCE

Formalized Rules



Section Navigation

- Architecture Design
  - Getting Started
  - Concept Description
  - Guidances
  - Roles
  - Architecture Workflows
  - Architecture Work Products
  - Example model of architectural design
- Change Management
- Configuration Management
- Documentation Management
- Implementation
- Platform Management
- Problem Resolution
- Process Management
- Quality Management
- Release Management
- Requirements Engineering
- Safety Analysis
- Safety Management
- Security Analysis
- Security Management
- Tool Management
- Verification

Create/Maintain Components architecture status: valid

tags: architecture_design, architecture_design
contains: gd_guidl__arch_design, gd_temp__arch_comp
has: doc_concept__arch_process, doc_getstrt__arch_process
needs input: wp__feature_arch, wp__requirements_comp, wp__issue_track_system
outputs: wp__component_arch
responsible: rl__contributor
approved by: rl__committer
supported by: rl__safety_manager, rl__security_manager
satisfied by: gd_req__arch_model, gd_req__arch_hierarchical_structure, gd_req__arch_build_blocks,
gd_req__arch_build_blocks_corr, gd_req__arch_viewpoints, gd_req__arch_attribute_uid, gd_req__arch_attr_security,
gd_req__arch_attr_safety, gd_req__arch
gd_req__arch_attr_mandatory, gd_req_
gd_req__arch_linkage_security_trace, g
gd_req__arch_consistency_model, gd_r

The component architectures ar

wf__cr_mt_comparch

Monitor/Verify Architec

tags: architecture_design, architecture_
contains: gd_guidl__arch_design, gd_ch
has: doc_concept__arch_process, doc_c
needs input: wp__feature_arch, wp__co
outputs: wp__issue_track_system, wp__
responsible: rl__committer
approved by: rl__committer
supported by: rl__safety_manager, rl__s
satisfied by: gd_req__configuration_uid
gd_req__general_checklist_templates, g

The architecture designs are monitored and verified.

The inspection shall be implemented as integral part of the review in GitHub

Create/Maintain Components architecture status: valid

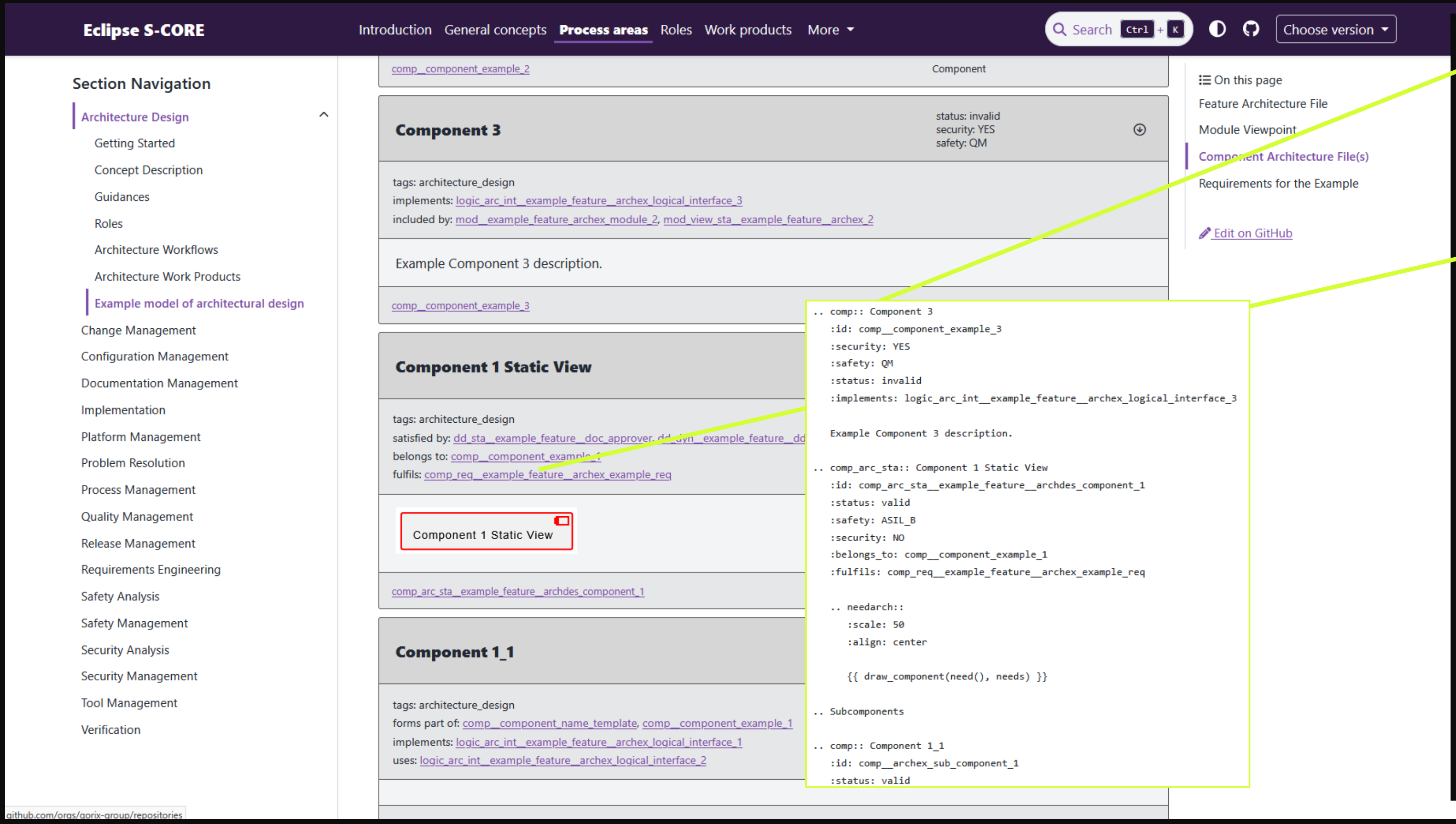
tags: architecture_design, architecture_design
contains: gd_guidl__arch_design, gd_temp__arch_comp
has: doc_concept__arch_process, doc_getstrt__arch_process
needs input: wp__feature_arch, wp__requirements_comp, wp__issue_track_system
outputs: wp__component_arch
responsible: rl__contributor
approved by: rl__committer
supported by: rl__safety_manager, rl__security_manager
satisfied by: gd_req__arch_model, gd_req__arch_hierarchical_structure, gd_req__arch_build_blocks,
gd_req__arch_build_blocks_corr, gd_req__arch_viewpoints, gd_req__arch_attribute_uid, gd_req__arch_attr_security,
gd_req__arch_attr_safety, gd_req__arch_attr_status, gd_req__arch_attr_fulfils, gd_req__arch_traceability,
gd_req__arch_attr_mandatory, gd_req__arch_linkage_safety, gd_req__arch_linkage_safety_trace,
gd_req__arch_linkage_security_trace, gd_req__arch_linkage_requirement, gd_req__arch_linkage_requirement_type,
gd_req__arch_consistency_model, gd_req__arch_consistency_interf, gd_req__arch_consistency_dynamic

- Workflow describes how to create/maintain templates
- A filled-out template becomes a managed document

https://eclipse-score.github.io/process_description/main/process_areas/architecture_design/architecture_workflow.html

AND IT WORKS: CODE IS COMPLIANCE

Formalized Rules



comp__component_example_2 Component

Component 3

status: invalid
security: YES
safety: QM

tags: architecture_design
implements: logic_arc_int__example_feature__archex_logical_interface_3
included by: mod__example_feature_archex_module_2, mod_view_sta__example_feature__archex_2

Example Component 3 description.

comp__component_example_3

Component 1 Static View

tags: architecture_design
satisfied by: dd_sta__example_feature__doc_approver, dd_dyn__example_feature__dd
belongs to: comp__component_example_1
fulfils: comp_req__example_feature__archex_example_req

comp_arc_sta__example_feature__archdes_component_1

Component 1_1

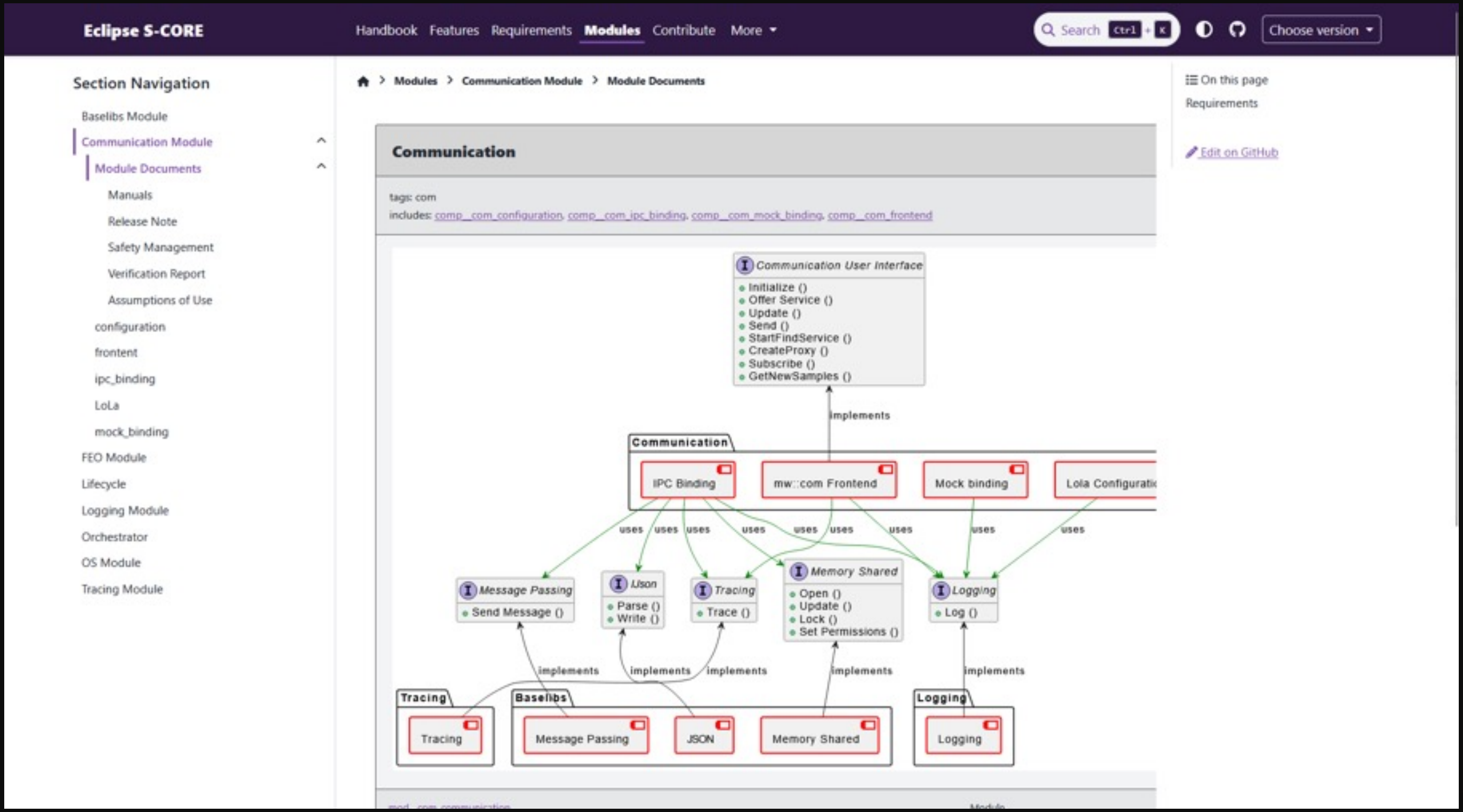
tags: architecture_design
forms part of: comp__component_name_template, comp__component_example_1
implements: logic_arc_int__example_feature__archex_logical_interface_1
uses: logic_arc_int__example_feature__archex_logical_interface_2

```
.. comp:: Component 3
   :id: comp__component_example_3
   :security: YES
   :safety: QM
   :status: invalid
   :implements: logic_arc_int__example_feature__archex_logical_interface_3

   Example Component 3 description.

.. comp_arc_sta:: Component 1 Static View
   :id: comp_arc_sta__example_feature__archdes_component_1
   :status: valid
   :safety: ASIL_B
   :security: NO
   :belongs_to: comp__component_example_1
   :fulfils: comp_req__example_feature__archex_example_req

   .. needarch::
      :scale: 50
      :align: center

      {{ draw_component(need(), needs) }}

.. Subcomponents

.. comp:: Component 1_1
   :id: comp__archex_sub_component_1
   :status: valid
```

- Documentation treated as code
- Ideas will become compliant contributions

AND IT WORKS: CODE IS COMPLIANCE

Automated Validation (CI/CD)



Communication

tags: com

includes: comp__com_configuration, comp__com_ipc_binding, comp__com_mock_binding, comp__com_frontend

- Processes / metamodel is applied in every build
- Merge is blocked with open issues
- Documentation is always up-to-date
- **Developer focus is on innovation**

https://eclipse-score.github.io/score/main/modules/communication/docs/index.html

Experiences from Eclipse S-CORE

Eclipse S-CORE

Clear reduction of typical issues like

- Not used templates
- Outdated documents
- Traceability failures
- Missing reviews

Still open points

- Logical failures still possible and can only be found (actually) by experienced experts
- Changes in the machine room (meta model) have a high impact / can break everything

On demand high provided certifiability is possible!

Thank you.

Contact us

Markus Schupfner
Chief Executive Officer Qorix
markus.schupfner@qorix.ai

Dr. Nico Hartmann
Chief Technology Officer Qorix
nico.hartmann@qorix.ai