CARNIQ

PROCESS FELLOWS

Welcome to our webinar

Automotive Architectural Design using carAISuite

Solving automotive engineering and compliance challenges using AI

Phalgun Upadhyaya
AI Architect -
CARNIQ Technologies GmbH

Alexander Feulner
Senior Consultant -
Process Fellows GmbH

CARNIQ

AGENDA

01 About CARNIQ

02 About ProcessFellows

03 Challenges in System Architecture Design

04 carAISuite tool live demo

05 Past guidelines and new trends in System Architecture

06 AI Use cases in carAISuite

07 Upcoming features

08 Q&A session

CARNIQ

Got questions? We'd love to hear them!

Use the Q&A chat box to submit your questions.

CARNIQ

Locations

- Munich, Germany
  Head quarters
- Mysuru, India

CARNIQ

Mission

To expedite the security of NextGen vehicles through disruptive innovations & technologies

CARNIQ

BUSINESS MODEL

01 Products

02 Services

CARNIQ

SERVICES

01

Cybersecurity Analysis

02

Cybersecurity Training

03

Cybersecurity Verification & Validation

04

Cybersecurity Development

05

Cybersecurity Management

CARNIQ

PRODUCTS

01 carAISuite

AI based Automotive Engineering & Compliance Tool

02 carSECURITY

Automotive Cybersecurity Process Landscape Tool

OUR PORTFOLIO

PROCESS FELLOWS

PROJECT MANAGEMENT

AGILE METHODS

ALM

ORGANIZATIONAL CHANGE

PROCESSES

SPICE

SQIL

SYSTEMS ENGINEERING

CYBERSECURITY

FUNCTIONAL SAFETY

OUR SERVICES

ASSESSMENT

From a **gap analysis** to an official **assessment**, we examine your projects and processes and **propose improvements.**

CONSULTING

Together with you, we develop the **best process solution** covering **complex** and **safety-critical** systems engineering, considering all the advisable general conditions.

COACHING

We **empower** your **employees** by accompanying them from the definition of measures to the completion of your product development.

TRAINING

We offer the **optimal solution** for you when it comes to further **training of your employees.** These can be standardized courses as well as individually tailored training courses.

SPICE 4 CARS

FREE

PROCESS FELLOWS

SPICE 4 CARS

YOUR SPICE KNOWLEDGE HUB

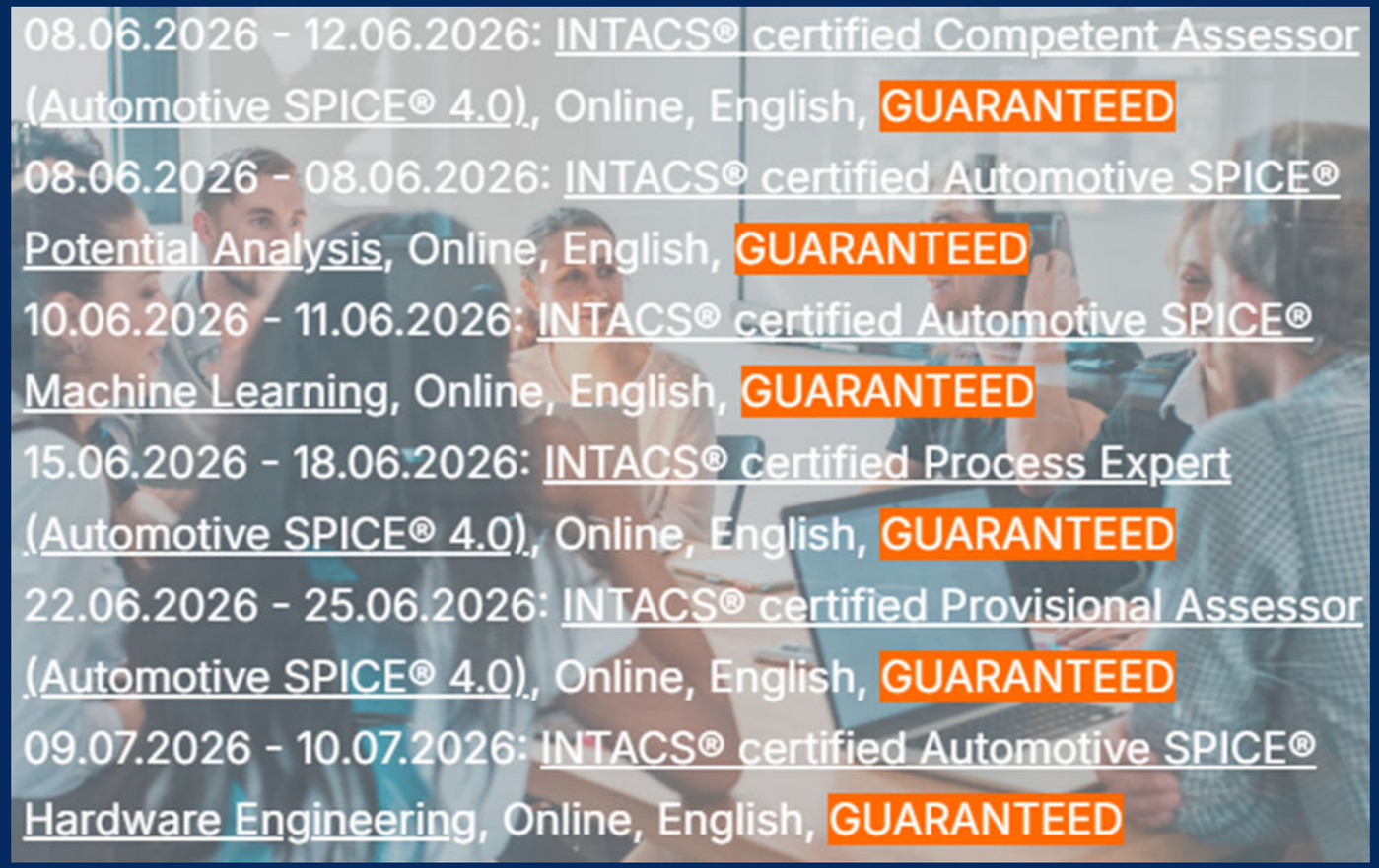
FOR AUTOMOTIVE

Celebrating next milestone:

500 **mapped SPICE model elements ...**

- Model Browser
- Comprehensive Knowledge Database
- Efficient Navigation
- Enabling Understanding of SPICE model elements

UPCOMING TRAININGS



Further training dates on our website:
https://www.processfellows.de/dates.html

Typical Challenges within System Architectural Engineering

CHALLENGES

Increasing System Complexity

- Systems are interdisciplinary (software, hardware, mechanics)
- Multiple abstraction levels (stakeholder → system → subsystem → software)

Understanding the **impact of changes** across the systems **becomes difficult.**

CHANGE VOLATILITY

- Agile iterations
- Customer feedback loops
- Regulatory updates

Requirements are frequently updated → **risk of inconsistency** and **outdated artifacts.**

CHALLENGES

TRACEABILITY OVERHEAD

- Bidirectional traceability required in regulated environments

Manual maintenance is time-consuming and error-prone.

INCONSISTENCY & AMBIGUITY

- Interface mismatches, integration bottlenecks

Errors propagate into lower design levels and test phases.

CHALLENGES

Siloed Toolchains

- Architecture tools separate from: Requirement tools;
- Test management systems; DevOps pipelines

Governance, alignment, and visibility become **difficult.**

CARNIQ

The carAISuite Way

Enable rapid and systematic development and compliance activities using Artificial Intelligence

Using carAISuite, the Automotive Architecture Generation activities will be performed by Artificial Intelligence leading systematic, reliable and quick results.

CARNIQ

SYSTEM ARCHITECTURE

DEMO

Development of System Architectural Engineering (1/2)

1968 – The „Software Crisis“

From Software Crisis to Formal Specification:
1968/1969 NATO Conferences on Software Engineering introduced the term Software Engineering as a response to large-scale project failures. One key realization was that **system complexity must be managed through structure and decomposition.**

1972 – Structured System Decomposition

Structured System Decomposition:
1972 – Parnas' Modular Design Principles - David Parnas introduced the concept of modular decomposition for controlling the complexity of the system architecture.

1987– Systems Engineering and Architecture Frameworks

1987 – MIL-STD-499 Series (Military Standard):
One of the **first major systems engineering standards.**
Introduced formal processes for: System definition, functional analysis, interface management, verification planning, lifecycle engineering **Basis for frameworks like Zachman, DoDAF**.

2005 – SysML Introduction

2005 – SysML Introduction
OMG introduces **Systems Modeling Language (SysML)** extending UML with: Requirements diagrams, Parametric modeling, Allocation relationships, etc. -> **emerging of MBSE**

2008 – ISO/IEC/IEEE 15288

Systems Engineering:
ISO/IEC/IEEE 15288:2008 defines a unified systems engineering lifecycle framework: System lifecycle processes, Technical processes Architecture definition.

Development of System Architectural Engineering (2/2)

2020 – 2023: Digital Systems Engineering

Continuous Digital Systems Engineering
Modern systems engineering integrates: Requirements, Architecture Simulation, Testing, Operations, Digital twins **Architecture becomes continuously synchronized system knowledge.**

2023–2025: AI-supported Architectural Development

Continuous and AI-Supported Architectural Development – integration with DevOPS, Continuous Integration, MBSE.
AI and LLMs: Research and early adoption in architecture design, automation, traceability support, check for completeness.

2026- 2030: (Forecast): Autonomous & Knowledge Centric System Architectural Engineering

Autonomous & Knowledge Centric System Architectural Engineering
Based on current research: AI Agents that supports architecture coordination, interface coordination, requirement allocation, variant management, verification preparation + simulation.

Researches and current studies indicates not just better architecture documentation, but **machine-reasonable engineering ecosystems!**

Characteristics of Modern Architectural Engineering

- **Lifecycle Integration**
  - Architecture is managed across: Engineering, Requirements, Testing, Domain Levels
  - Supports continuous refinement

Complexity
Tool silos

- **Continuous Traceability**
  - Impact analysis capabilities
  - Bidirectional traceability across several abstraction levels

Traceability
Overhead

- **Consistency & Verification**
  - Simulation & verification integration
  - Dependency analysis

Inconsistency
Complexity

- **Quality Governance & Review Mechanisms**
  - Standardized architecture templates
  - Formal review workflows

Ambiguity
Inconsistency

- **Model-Based & Structured Representation**
  - Integration with MBSE

Complexity

AI-Augmented Architectural Modeling (Emerging)

From Manual Control to Intelligent Assistance AI Use cases

- **Intelligent Architecture Generation**
  - Generates initial architecture models from: **Requirements, User Stories, SC**
  - Suggests: Components, Interfaces, Service Boundaries, Layered Structures
  - Creates: UML/SysML diagrams/viewpoint automatically

- **Consistency Verification**
  - Cross-checks architecture models against:
    - Requirements
    - **Safety** and **Security** constraints
    - Design principles
  - Detects:
    - **Missing interfaces**
    - Contradicting dependencies

- **Smart Impact Analysis**
  - Analyzes change requests semantically
  - Identifies potentially affected: Requirements, Architecture elements

- **Faster architecture generation**
- **Reduced manual modeling effort**

- **Reduced hidden inconsistencies**
- **Improved architecture quality assurance**
- **Faster review and audit preparation**

- **Faster change evaluation**
- **Reduced integration risks**
- **Improved decision support**

AI-Augmented Architectural Modeling (Emerging)

From Manual Control to Intelligent Assistance AI Use cases

- **Semantic Architecture Analysis**
  - Uses graph embeddings and semantic similarity analysis to:
    - Detect **duplicate components**
    - Reveal **overlapping functionalities**
    - Recommend architecture consolidation

- **Reduced architecture redundancy**
- **Better maintainability and reuse**

- **AI-Assisted Architecture Optimization**
  - Evaluates architecture alternatives using:
    - **Quality attributes**
    - **Performance constraints**
    - **Scalability**
  - Suggest improvement for: Coupling/cohesion, decomposition, design patterns

- **Reduced design iteration cycles**
- **Improved decision support**

CARNIQ

carAISuite

Compliance

Formal Reviews

- Cybersecurity, ASPICE, Functional Safety

Generation

Engineering Work Products

- System, Software, Hardware

Process Work Products

- Cybersecurity, ASPICE, Functional Safety

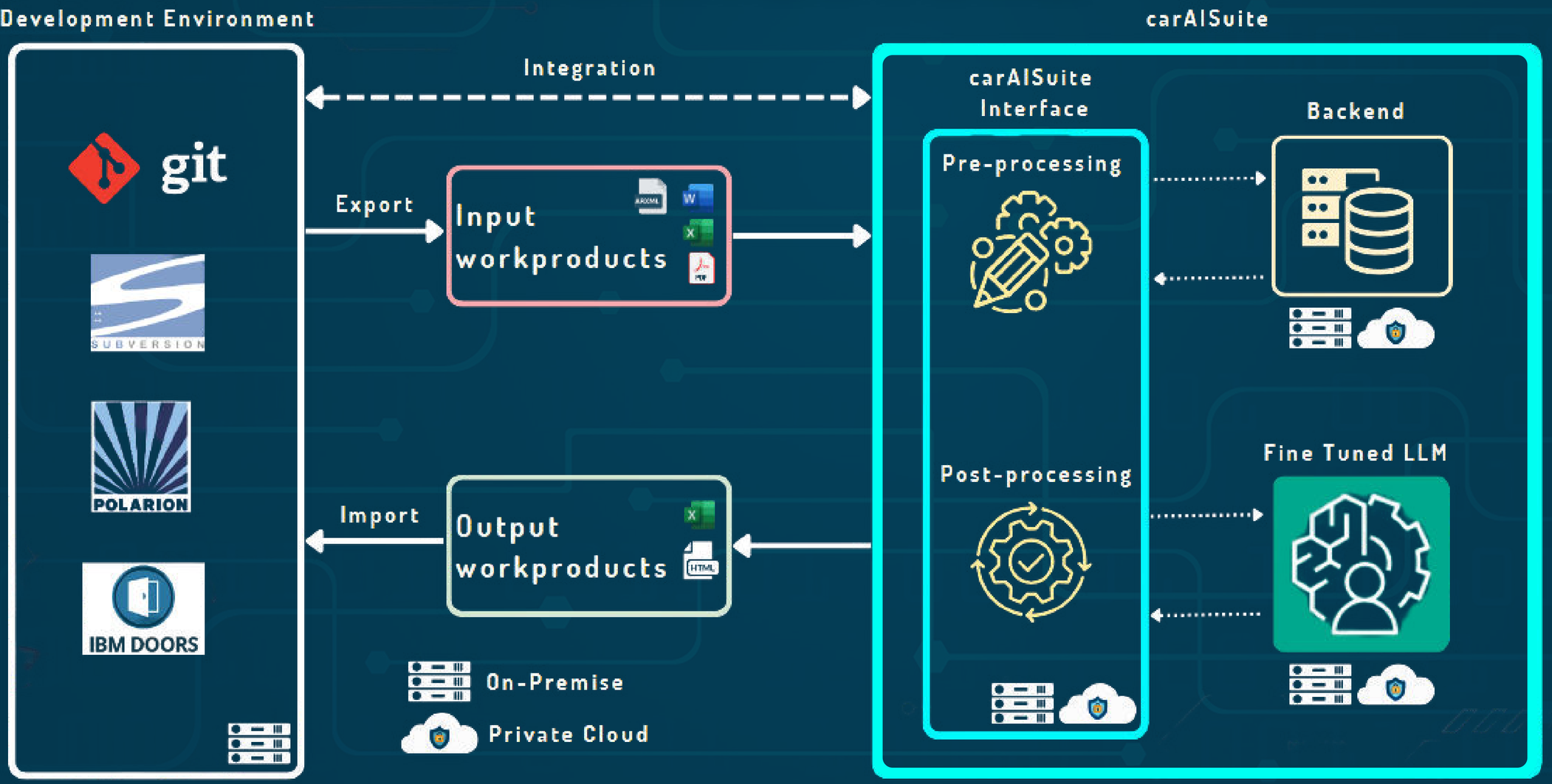
Elicitation

Requirement Elicitation

- Stakeholder Requirement Elicitation

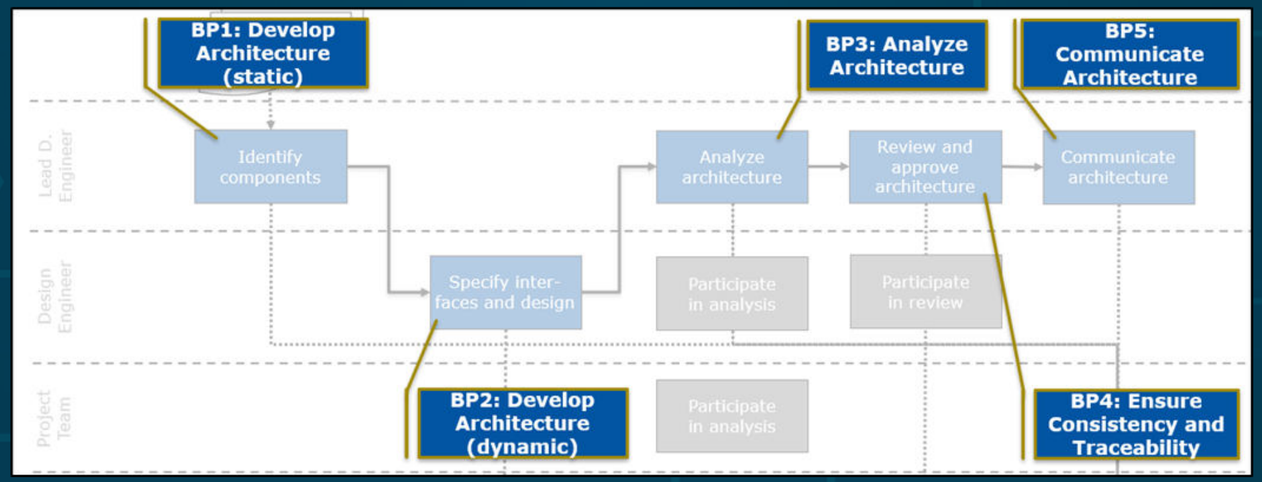
CARNIQ

ARCHITECTURE



Requirements by Automotive SPICE® v.4.0 for the System Architectural Design (SYS.3)

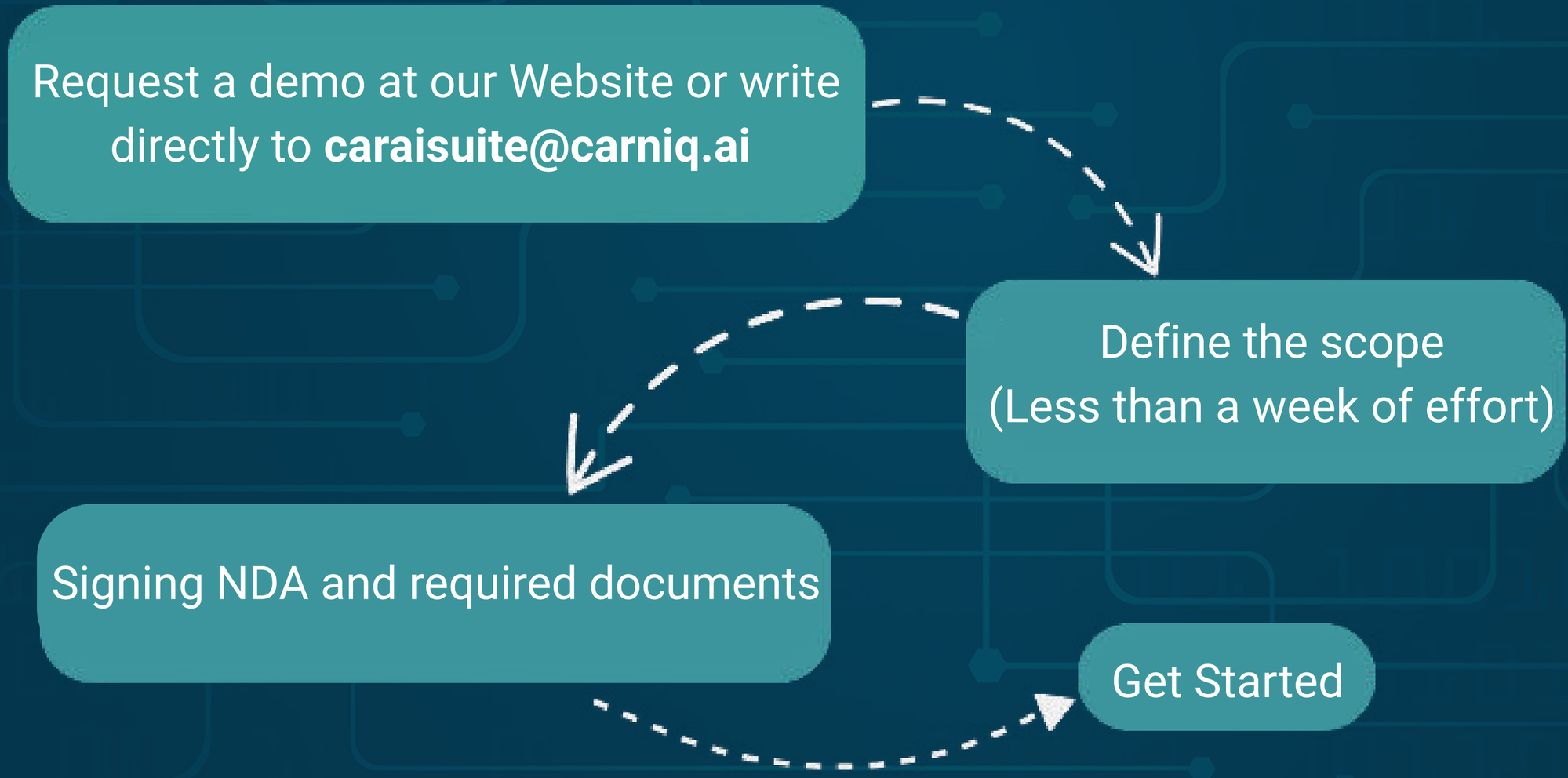
- **Purpose**
  - The purpose is to establish an analyzed system architecture, comprising static and dynamic aspects, consistent with the system requirements.



- **Required key characteristics for the System Architectural Design**
  - Architecture needs to provide different viewpoints -> **static and dynamic architecture**
  - Architecture needs to be **analyzed** (e.g. FMEA) + documented **design decision**
  - **Traceability and consistency** of the architecture needs to be ensured
  - **Communication** of the architecture to relevant stakeholders needs to be established

CARNIQ

EXECUTE FREE OF COSTS POCS WITH US



CARNIQ

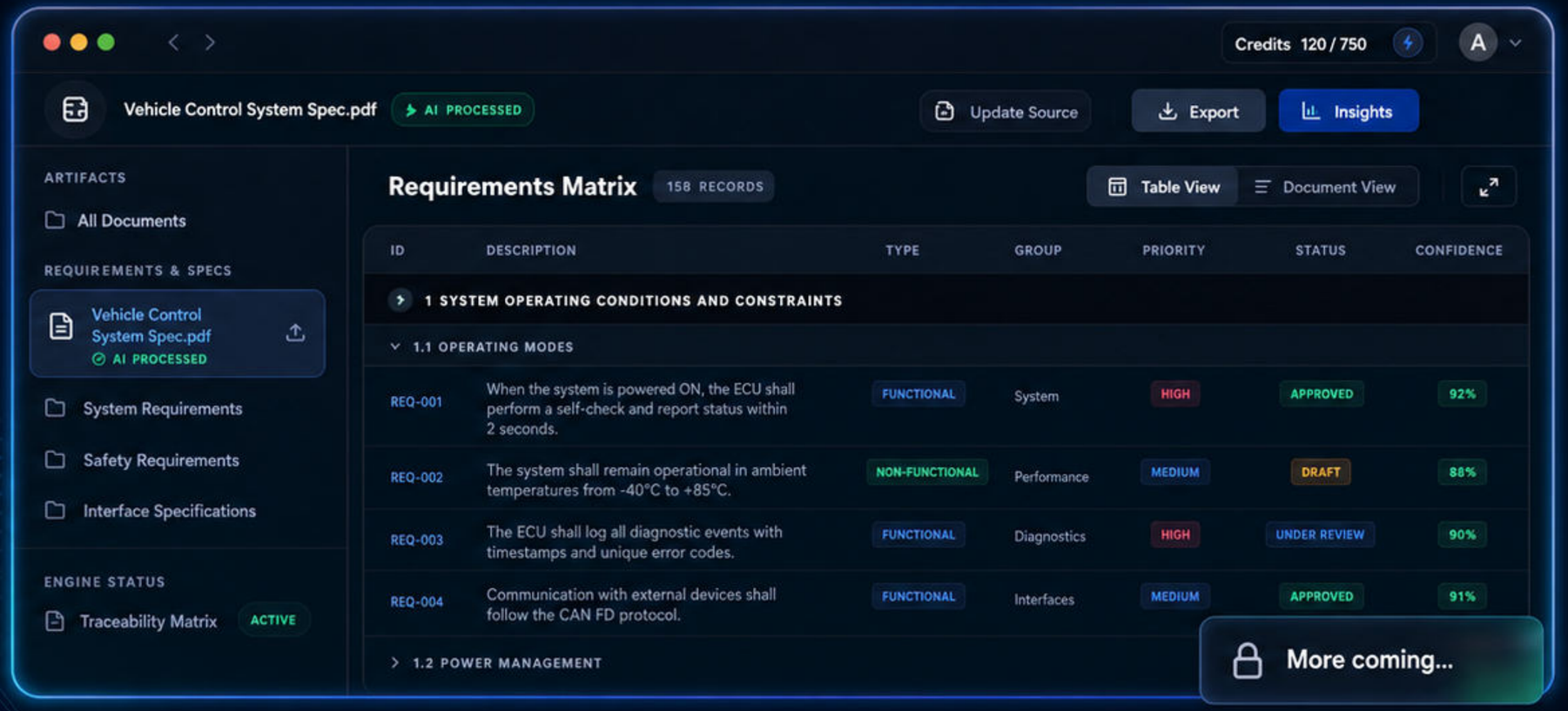
PROCESS FELLOWS

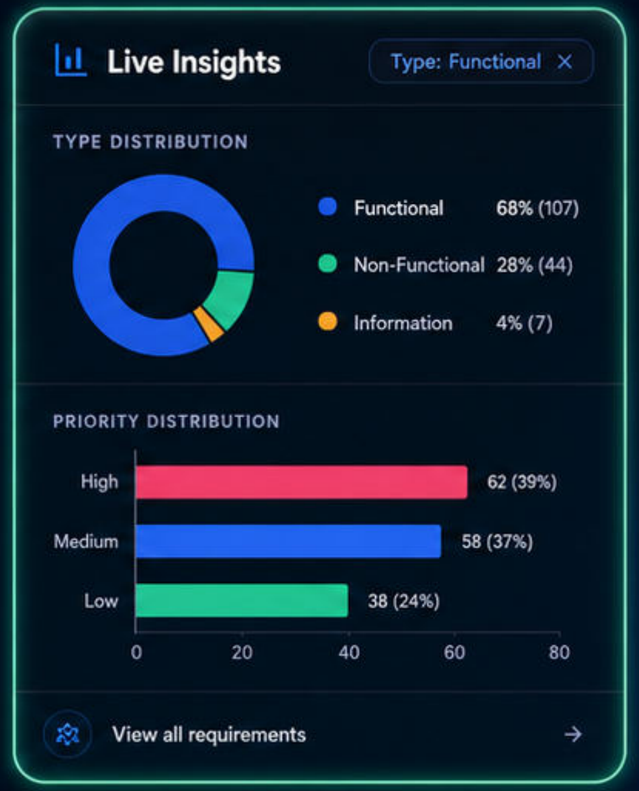
Time for your Questions...

CARNIQ

From raw specs to actionable requirements in minutes

AI assisted requirement structuring, live insights and traceability ready outputs.





Live document ingestion | AI-assisted requirement structuring | Traceability-ready exports

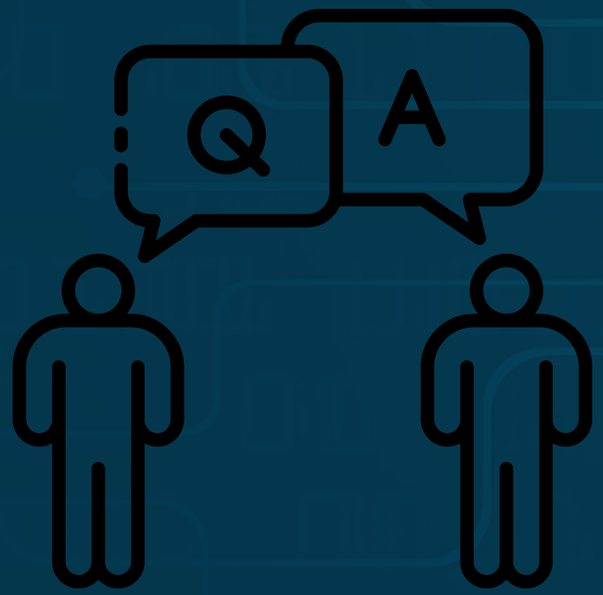
Launching soon • Early access

Click here to Sign Up or visit: www.carniq.ai/caraisuite

CARNIQ

WEBINAR Q&A HIGHLIGHTS

Answers to Questions from Participants



CARNIQ

Have you checked if exact same requirements, design constraints or goals generate exact same architecture every time or does it vary?

In general, LLM-based systems can produce variations between runs because they are probabilistic by nature.

However, in carAISuite we actively work to reduce that variability through:

- structured prompts,
- domain-specific workflow orchestration,
- engineering rules and constraints,
- controlled architecture patterns,
- and validation layers.

So for the same requirements and constraints, the core architecture intent and decomposition remain highly consistent.

Our goal is not to generate random creative outputs, but to support structured engineering workflows with reproducible and reviewable results.

We have a regeneration agent in place which exactly does this task.

CARNIQ

How do you handle “hallucinations”, what is your criteria for them?

Our approach to reducing hallucinations is based on multiple layers:

- structured input processing,
- domain-specific prompting,
- engineering rules and constraints,
- controlled workflow orchestration,
- traceability mechanisms,
- and validation-oriented review processes.

In addition to the generation agents, we also use dedicated quality and review agents that evaluate the generated outputs against predefined engineering criteria and consistency expectations.

The platform can generate structured review reports highlighting:

- ambiguities,
- inconsistencies,
- missing information,
- traceability gaps,
- interface concerns,
- and requirement quality observations.

We also use evaluation metrics and internal assessment mechanisms to measure aspects such as:

- requirement coverage,
- structural consistency,
- traceability completeness,
- architectural alignment with constraints,
- and output quality indicators.

The objective is not to treat AI generation as a black-box activity, but as a controlled engineering support workflow with explainability, reviewability, and validation support built into the process.

CARNIQ

How do you handle changes? Is it able to take a baseline of requirements and design baseline as input plus changes to requirements and then generate updated architectural design with minimal impact?

Our approach is based on identifying and analyzing deltas between the existing baseline and the updated inputs.

The platform includes AI-driven agents that detect changes in requirements, constraints, and engineering context, and then assess which parts of the architecture may be impacted.

If requirements baselines are maintained externally in tools such as DOORS or similar ALM platforms, the tool can work with those updated baselines as input. If the baselines are maintained within the platform, we also support internal baseline concepts and version-aware workflows.

The objective is not to regenerate the complete architecture from scratch every time, but rather to support controlled evolution of the design with minimal and traceable impact wherever possible.

Of course, the actual impact depends on the nature of the requirement change. Some changes may only affect interfaces or component responsibilities, while others can require broader architectural restructuring.

CARNIQ

Is the tool compliant with AI standards and regulations, such as the EU AI Act?

We are actively following the developments around the EU AI Act and related AI governance frameworks very closely.

Today, we position carAISuite as an AI-assisted engineering support tool designed to support engineering workflows rather than as a fully autonomous decision-making system.

The platform is built with several important principles in mind, including:

- human-in-the-loop validation,
- traceability of generated outputs,
- engineering review and oversight,
- controlled workflows,
- and reproducibility-focused generation approaches.

CARNIQ

Is this tool a plug-and-play solution, ready for immediate use, or does it require training on company specific data and architectural standards?

The platform is designed to be usable as a plug-and-play AI-assisted engineering solution, while also supporting deeper customization for organization-specific workflows and standards.

In the upcoming web-based version, which we plan to release in the coming weeks, we are also introducing a more user-friendly and scalable workflow for:

- project-based collaboration,
- configurable engineering contexts,
- reusable knowledge integration,
- and AI-assisted workflow orchestration directly through the web platform.

Our goal is to combine fast onboarding with the flexibility needed for real enterprise engineering environments.

CARNIQ

Does the tool perform evaluation of alternate architectures against the provided criteria and suggest the appropriate architecture?

We can already guide the generation process using:

- design constraints,
- functional decomposition,
- interface expectations,
- and domain-specific engineering patterns.

In terms of alternate architecture evaluation, we are progressively moving toward supporting comparative assessment of architectural options against defined criteria such as:

- complexity,
- modularity,
- scalability,
- interface density,
- maintainability,
- and process and compliance considerations.

The long-term vision is not only to generate architectures, but also to assist engineers in evaluating trade-offs between multiple architecture candidates.

That said, architecture selection in real projects is often influenced by organizational standards, legacy systems, safety goals, cost targets, and OEM-specific constraints. So we currently position the AI as a decision-support assistant rather than an autonomous architecture decision-maker.

The engineer still remains in control of the final architectural choice.

CARNIQ

Does it also capture rationale for design decisions and definition of functionality of each block and interface definitions?

The platform does not only generate architectural elements, but also provides supporting rationale based on:

- the analyzed requirements,
- identified constraints,
- inferred functional responsibilities,
- interface relationships,
- and the engineering context derived from the provided inputs.

For generated architectural blocks, the tool can provide descriptions of the intended functionality, responsibilities, and interactions within the system context.

For interfaces, the generated outputs can include interface definitions, communication relationships, and associated requirement traceability depending on the available input detail.

The goal is to make the generated architecture more explainable and reviewable rather than producing black-box outputs.

CARNIQ

Is it also able to generate dynamic views like sequence diagrams for key scenarios or state transition diagrams?

Yes — in addition to static architectural representations, the platform is also capable of supporting dynamic behavioral views such as sequence-oriented interaction flows and scenario-driven representations.

Based on the provided requirements, interfaces, functional flows, and system context, the tool can generate dynamic interaction views for selected use cases or operational scenarios.

This includes support for:

- sequence-oriented interaction representations,
- communication flows between architectural elements,
- scenario-based behavioral visualization,
- and state- or mode-related behavioral perspectives depending on the available input and context.

CARNIQ

Is there ability to define design semantics including rules for decomposition of logical and physical design assets across system architecture layers?

Yes — the platform is designed to support configurable design semantics and architecture rules across different system layers. Organizations can incorporate their own decomposition strategies, architectural patterns, layering concepts, interface rules, and physical/logical design conventions to align the generated outputs with their internal engineering standards and methodologies.

CARNIQ

Do you have plans for generation of test cases against architectural design?

carAISuite also support test specification generation at system and software layers.

CARNIQ

How many stakeholder requirements can the tool handle? 1k, 10k? more?

The platform is designed to handle large-scale engineering projects with thousands of requirements through agent-based orchestration, contextual segmentation, and scalable workflow processing.

CARNIQ

How does the tool handle conflicting requirements (e.g., maximum performance vs low cost)?

carAISuite includes quality and review agents that analyze the requirements to identify potential conflicts, ambiguities, and trade-off situations.

CARNIQ

PROCESS FELLOWS

THANK YOU!