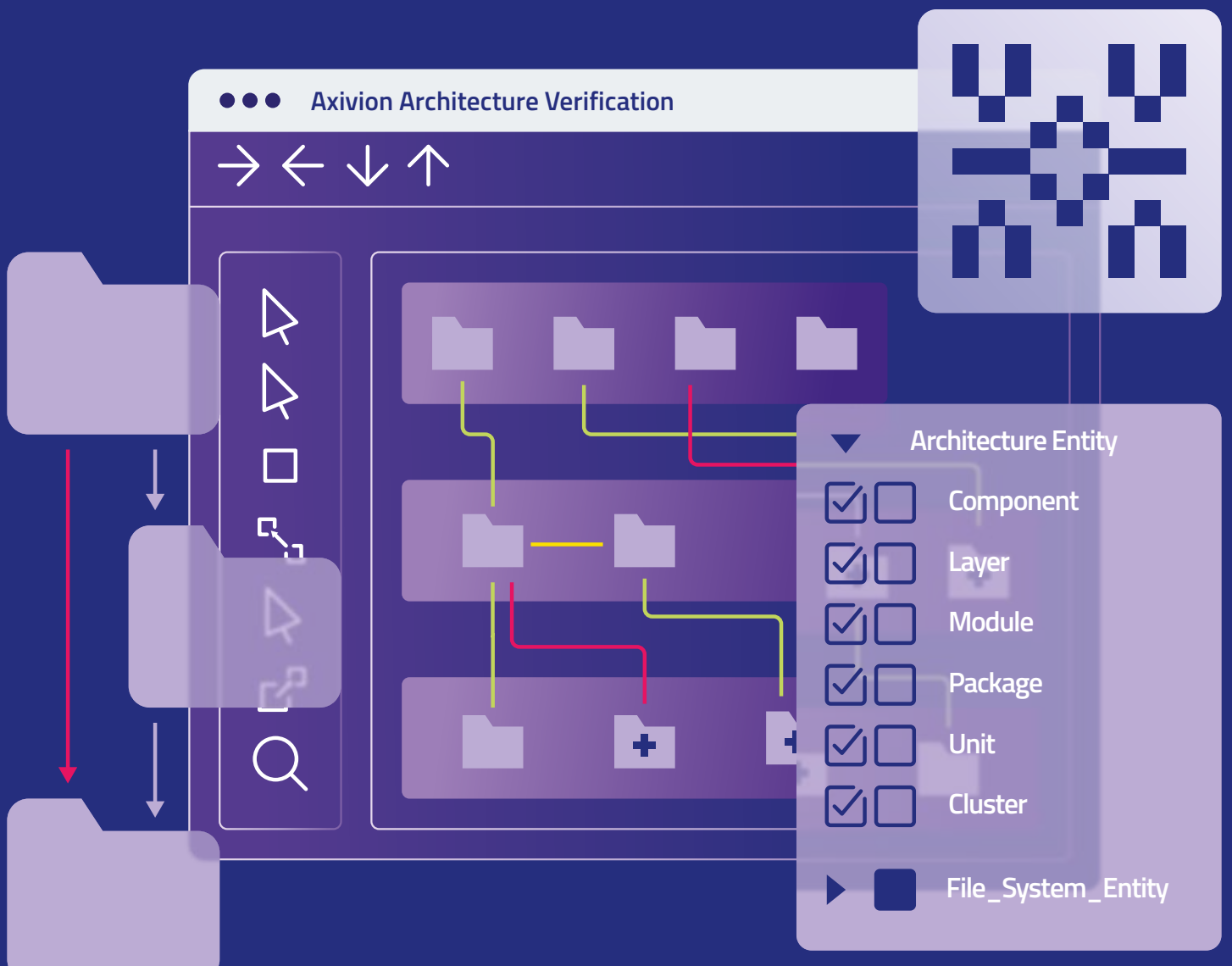


Axivion Architecture Verification

How to create the perfect foundation for future-proof software.



Axivion Architecture Verification

Ensure your code matches your software architecture.

Why Verify Your Architecture?

Software architecture and design need to match with the code for you to be sure you can use the software architecture as a guide and baseline for discussing the impact of new features. Only then is a long-term targeted and planned development of your products possible.

Axivion Architecture Verification ensures your code complies with your architecture. In addition to the functional architecture, the tool also reviews and checks safety and security architecture specifications for compliance, e.g. Freedom from Interference.

Use reliable documentation for decisive decision-making.

Your Benefits

- ✓ Replace error-prone manual reviews with exhaustive and reliable automated architecture verification.
- ✓ Capture the essential idea of your project regarding design and implementation to create a reliable foundation for your software.
- ✓ Eliminate the need to read the code to understand the architecture.
- ✓ Improve the onboarding of new team members so they can work efficiently right from the start.
- ✓ Use the architecture to understand the impact any changes to the code might have - in everyday development and for larger changes, such as refactoring or adding new features.
- ✓ Ensure that all changes to the code do not violate the software architecture to avoid architecture erosion (technical debt) and increase the lifecycle of your software.

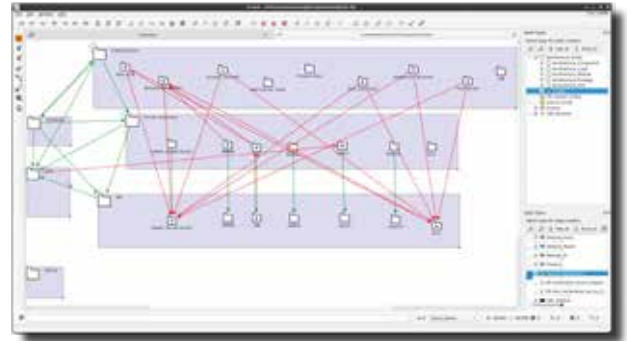
Easily integrate architecture verification into your established processes.

Key Features

- Integrated Modeler
- Interfaces to UML Tools
- AUTOSAR XML Import (ARXML)
- Freedom from Interference
- Architecture Reconstruction
- Architecture views for safety and security

Redocumentation of Software Architecture

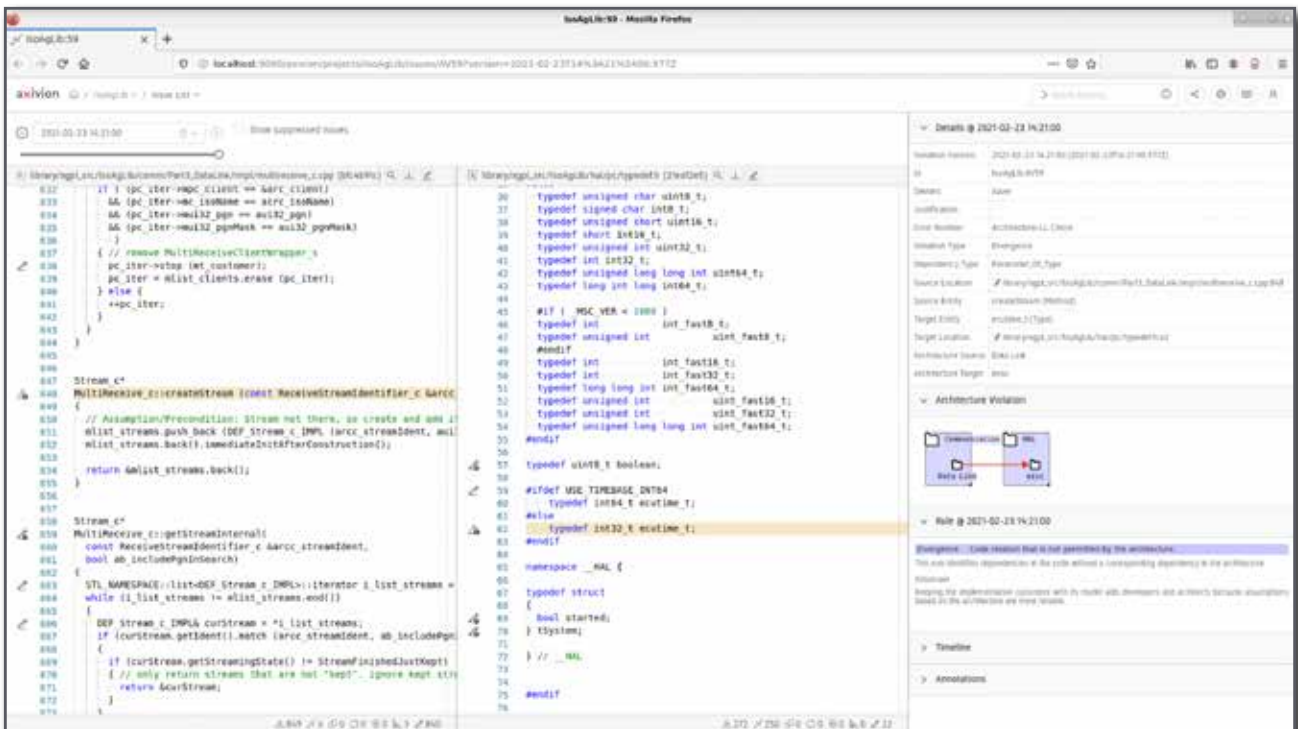
Axivion Architecture Verification can help you create a reliable basis for your project. No matter whether the architecture model exists (CASE-tool, XMI, XML etc.), you only have a documentation (Word, PowerPoint etc.) or you do not have a documentation at all (either for a live project or post project) - we will ensure that you can use the software architecture to easily explain what your software is about and how changes would affect it. This is particularly useful if you have to embed 3rd party code.



Seamless Integration Into Your Workflow

Ease of use is key when implementing new tools. Axivion Architecture Verification adapts to your needs and seamlessly integrates into your existing development environment. The extensive customization options make sure the architecture verification follows your rules and workflows. During the entire set-up process our dedicated experts will support you to ensure the results exceed your expectations.

Daily reports make it easy to identify any new issues and let you directly access the affected part of the code to make the necessary changes. Thanks to the delta analysis developers can directly see which issues in the code are new. When reviewing longer periods of time (e.g. multiple sprints or even the entire project history) smart filtering options help focus on specific types of issues.



The initial set-up is easier than you may think.

Simple Set-up in Just a Few Steps

With Axivion Architecture Verification creating and maintaining a reliable architecture can be done with ease. And once you have set it up, it automatically finds deviations early on and thus allows for corrections to be made before they affect the architecture.

Create your architecture model

You start by creating a model of your architecture. For this simply import any existing machine-readable model of your architecture via the interface or use the integrated modeler.

Create your code model

Extract your code model from your source code. The code model represents entities (e.g. source files) and classes or functions as well as the dependencies between them. Axivion can automatically construct your code model by analyzing your source code project.

Map the code to your architecture

Now establish how your code elements correspond to architecture components, by assigning the code elements to the architecture elements. Depending on your product structure and architecture model, this can be done:

- manually (e.g. with the Gravis modeler)
- automatically with Axivion by naming convention and hierarchical information or
- by using information given in the model (e.g. tagged values)

Interpret the dependencies

During this step you will interpret the dependencies within the architecture model and specify what they mean; in other words how to interpret architecture dependencies to match code dependencies. A simple example would be: Component A is allowed to call functions in component B. Of course, especially for UML models, far more advanced interpretations of dependencies (required/provided interfaces) are possible.

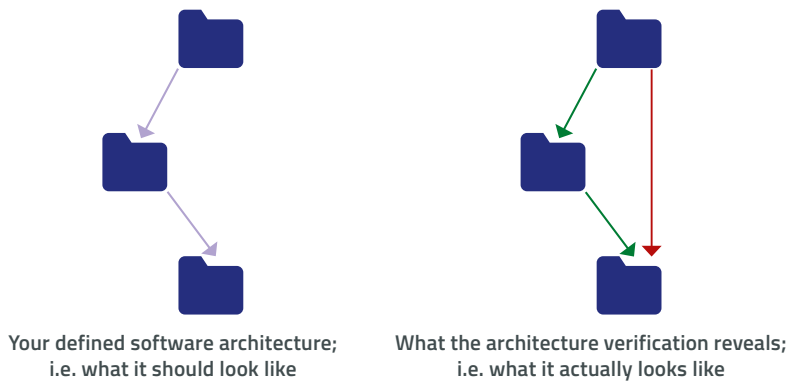
Verify Your Architecture - Every Day

Analyze your architecture

Now you are all set and can let Axivion verify your architecture. It will analyze your code and identify

- Convergences
- Divergences
- Absences

Benefit from sophisticated architecture verification.

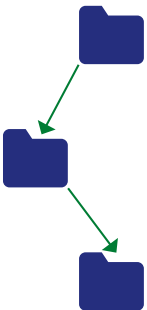


What to do with the results

When it comes to detecting deviations from the intended software architecture, nothing beats an automated process if you want exhaustive and reliable results. But it requires human intelligence to decide how to react to these findings. These are your options:

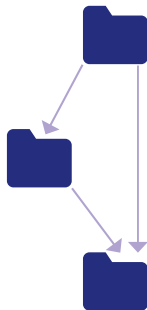
Make informed choices.

Correct the deviation
in the source code



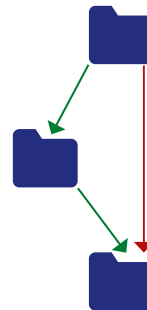
If this deviation is a mistake, the code should be corrected so it matches the architecture again.

Update the
architecture specifications



If this deviation actually makes sense, you should update the architecture accordingly.

Accept the
deviation temporarily



If this deviation is something you can accept temporarily, leave it and deal with it later.

Embed the architecture verification into the continuous analysis of your code.

Continuous Analysis of Your Code

Axivion Architecture Verification is not meant to give you just a single snapshot of your current architecture. It is designed to be part of your continuous software analysis and development process.

Once you have set up Axivion Architecture Verification, it easily integrates into the daily review process. Simply look at the deviations which are automatically detected and make the necessary changes as needed. This ensures your code always matches your software architecture.

Eliminating the error-prone manual verification and replacing it with reliable and precise diagnostic information, allows developers to tackle any findings at an early stage. This avoids costly adjustments at a later stage and helps you understand the impact of any changes you plan to make to your software.

The result: A reliable software architecture for longer lifecycles.

Freedom from Interference

Architecture According to ISO 26262

It is state of the art to allow several safety-relevant functions with different ASILs or QM classification to coexist on a common hardware. However, this requires a suitable software architecture to be in place for corresponding software projects according to ISO 26262. Compliance with the safety architecture is the basis for Freedom from Interference.

Axivion Architecture Verification ensures the consistent use of the defined interfaces and the selected communication mechanisms. Deviations from the architecture are immediately highlighted in the source text. This includes, among other things, unspecified function calls, overwriting of data or, more generally, reference to declarations not defined as interfaces.

Basic Technical Specifications

Note: This is just a rough overview and refers to Axivion 7.8. Please contact us for a complete list of specifications.

Supported languages and compilers

Languages | C, C++, C#

Supported operating systems

Host OS | Windows® 7/8/10/11, Windows® Server® 2008 R2/2012/2016/2019/2022
x86_64 GNU/Linux® (minimum requirement is glibc2.24 or later)
macOS® (minimum requirement is macOS 10.15)

Plugins

IDE | CLion, Eclipse™, Eclipse-based (e.g. Atollic TrueSTUDIO®, CodeWarrior®, DAVE™, STM32CubeIDE, TI Code Composer Studio™), Microsoft® Visual Studio®, Microsoft® Visual Studio Code®, Generic plugins

Supported UML® tools

UML® Tools | IBM Rational Rhapsody, Sparx Enterprise Architect (via XML or .qea-files), PlantUML

Other

Supported browsers | Microsoft® Edge, Mozilla Firefox®, Google Chrome™

Requirements | Python (3.8.1 - 3.12)
Java® Runtime (8, 11-14 and 17)

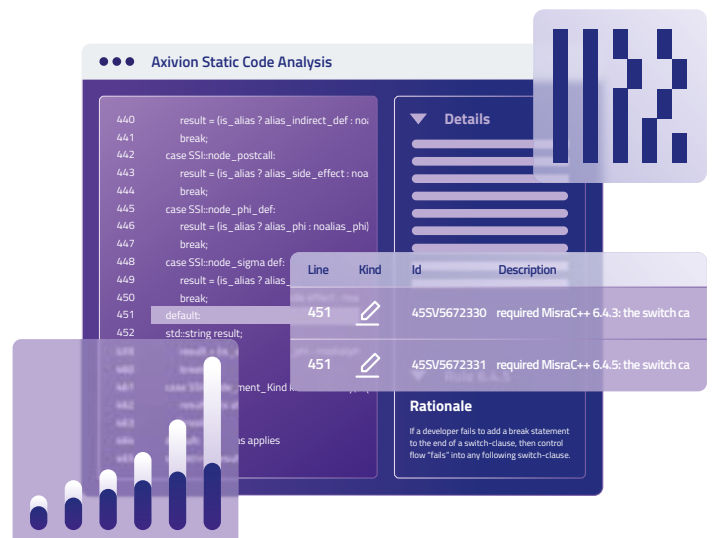
Technical data is subject to change without prior notice. All rights reserved. All company and/or product names are trademarks and/or registered trademarks of their respective manufacturers in their markets and/or countries. We are constantly making efforts to deliver the latest status of data to our partners. Specifications may change in the time between the product release and the release of this documentation.

Why stop here?

Axivion Architecture Verification will create a solid foundation for your software project. By adding Axivion Static Code Analysis you get an in-depth analysis of your code and can ensure it meets even the highest standards.



Visit our website to learn more or to schedule a meeting for a demo:
www.qt.io/axivion



Qt Quality Assurance

Qt Group (Nasdaq Helsinki: QTCOM) is a global software company, trusted by industry leaders and over 1.5 million developers worldwide to create applications and smart devices that users love. We help our customers to increase productivity through the entire product development lifecycle: from UI design and software development to quality management and deployment. Our customers are in more than 70 different industries in over 180 countries. Qt Group employs some 800 people, and its net sales in 2023 were 180.7 MEUR. To learn more, visit www.qt.io

www.qt.io/axivion