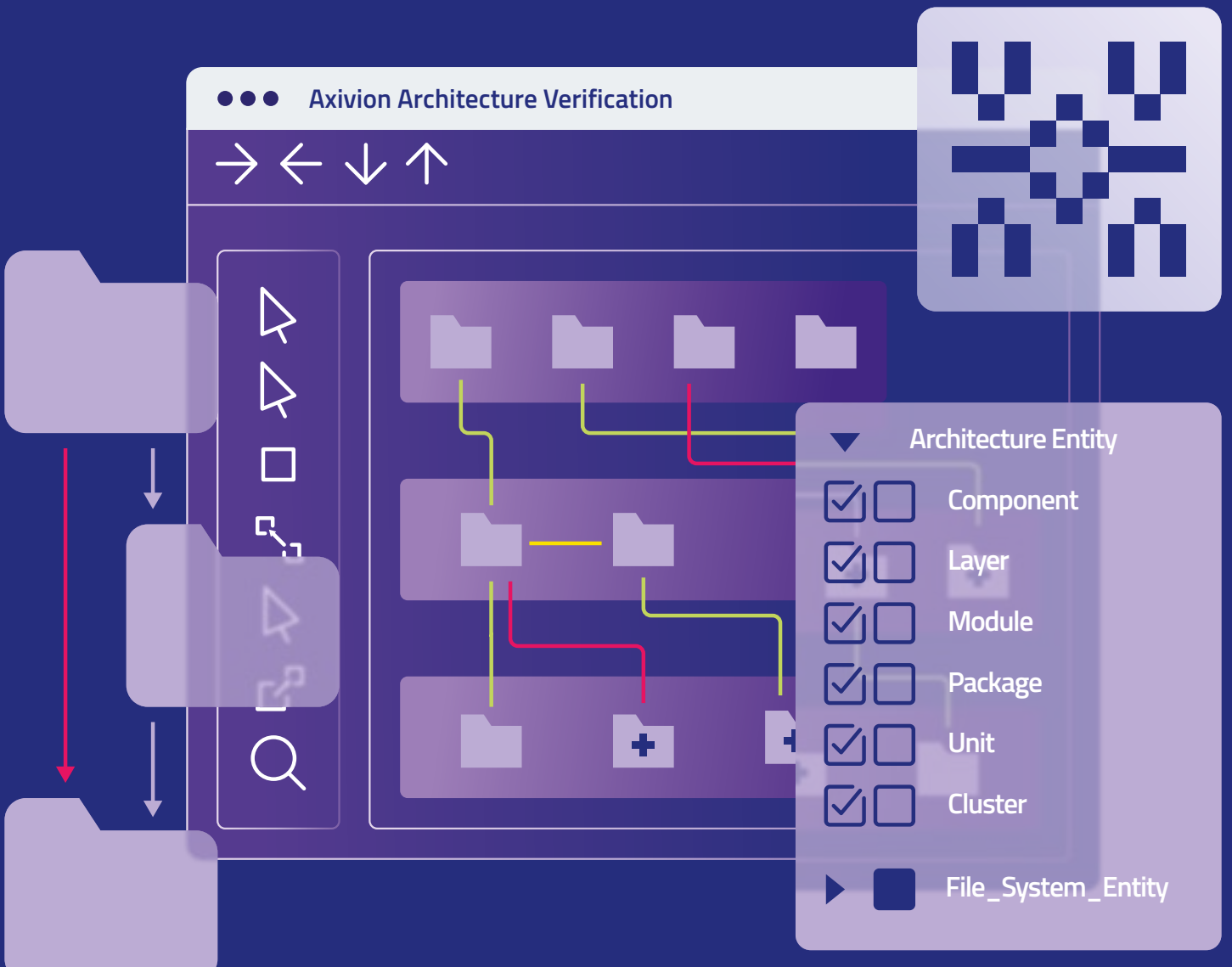


# Axivion Architecture Verification

미래를 대비한 소프트웨어의 완벽한 기반을 구축하는 방법



# Axivion Architecture Verification

Ensure your code matches your software architecture.

## Why Verify Your Architecture?

소프트웨어 아키텍처와 설계는 코드와 일치해야 하며, 이는 새로운 기능의 영향을 논의할 때 아키텍처를 가이드 및 기준으로 활용할 수 있게 합니다. 이를 통해 장기적인 목표와 계획에 따라 제품을 개발할 수 있습니다.

Axivion 아키텍처 검증은 코드가 아키텍처 규칙을 준수하는지 확인합니다. 이 도구는 기능적인 아키텍처 뿐만 아니라 안전과 보안을 고려한 아키텍처 사양(예: 간섭 방지)을 준수하는지도 검토하고 확인합니다.

Use reliable documentation for decisive decision-making.

## Your Benefits

- 오류가 발생하기 쉬운 수동 검증을 신뢰할 수 있으며 철저하게 자동화된 아키텍처 검증으로 대체하세요.
- 설계 및 구현과 관련된 프로젝트의 핵심 아이디어를 포착하여 소프트웨어의 안정적인 기반을 구축하세요.
- 아키텍처를 이해하기 위해 코드를 직접 확인할 필요가 없습니다.
- 새 팀원의 온보딩 프로세스를 개선하여 처음부터 효율적으로 업무에 참여할 수 있도록 하세요.
- 아키텍처를 활용하여 일상적인 개발 뿐 아니라 리팩토링, 혹은 새로운 기능 추가와 같은 대규모 변경이 코드에 미칠 영향을 파악하세요.
- 코드 변경이 소프트웨어 아키텍처를 위반하지 않도록 하여 아키텍처 침식(기술 부채)을 방지하고 소프트웨어의 수명 주기를 연장합니다.

Easily integrate architecture verification into your established processes.

## Key Features

- 통합 모델러
- UML 툴 인터페이스
- AUTOSAR XML 가져오기(ARXML)
- 간섭 방지 기능
- 아키텍처 재구성
- 안전 및 보안을 고려한 아키텍처 뷰

## Redocumentation of Software Architecture

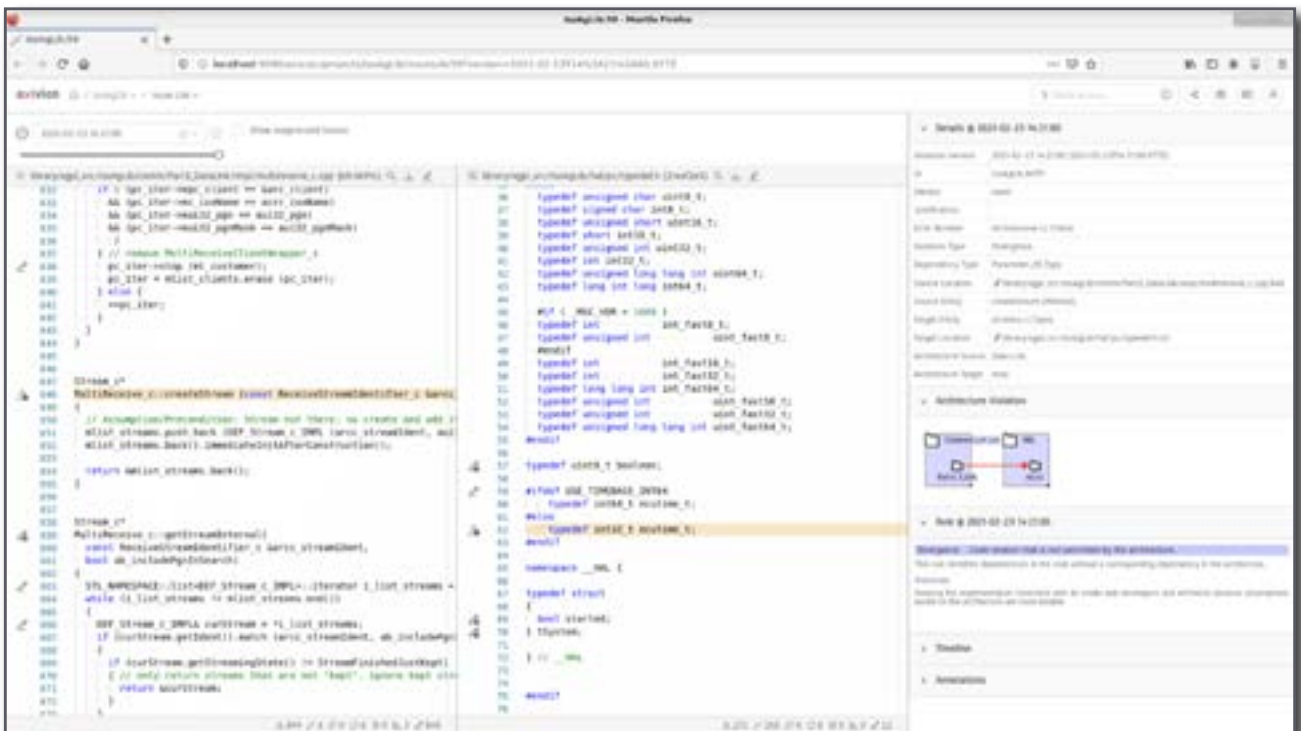
Axivion 아키텍처 검증은 프로젝트의 신뢰할 수 있는 기반을 구축하는데 도움이 됩니다. 아키텍처 모델이 있는 경우(CASE-tool, XML, XML 등), 문서만 있는 경우(Word, PowerPoint 등), 아니면 전혀 문서가 없는 경우(실시간 프로젝트 또는 완료된 프로젝트), Axivion은 소프트웨어 아키텍처를 통해 소프트웨어의 본질과 변경 사항이 어떤 영향을 미칠지 쉽게 설명할 수 있도록 도와드립니다. 특히 타사 코드를 포함해야 하는 경우에도 유용합니다.



## Seamless Integration Into Your Workflow

새로운 도구를 구현할 때 사용자 편의성이 핵심입니다. Axivion 아키텍처 검증은 사용자의 요구에 맞게 조정되며 기존 개발 환경과 원활하게 통합됩니다. 다양한 사용자 지정 옵션을 제공하여 아키텍처 검증이 사용자의 규칙과 워크플로우를 준수하도록 보장합니다. 또한 전문가가 셋업 단계 전반을 지원하여 최상의 결과를 달성할 수 있도록 돕습니다.

일일 보고서를 통해 새로운 이슈를 쉽게 파악하고 코드 영역에 직접 접근하여 필요한 변경을 수행할 수 있습니다. 델타 분석을 통해 개발자는 코드에서 새로 발생한 이슈를 직접 확인할 수 있습니다. 장기적인 검토(예: 여러 스프린트 또는 전체 프로젝트 기간) 시 스마트 필터링 옵션을 활용하여 특정 유형의 이슈에 집중할 수 있습니다.



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

## Simple Set-up in Just a Few Steps

Axivion 아키텍처 검증을 사용하면 신뢰할 수 있는 아키텍처를 쉽게 생성하고 유지할 수 있습니다. 또한 한 번 설정하면 자동으로 편차를 조기에 발견하고 아키텍처에 영향을 미치기 전에 수정할 수 있습니다.

### Create your architecture model

아키텍처 모델을 생성하는 것부터 시작합니다. 인터페이스를 통해 기존의 기계 판독 가능한 아키텍처 모델을 가져오거나 통합 모델러를 사용해 쉽게 진행할 수 있습니다.

### Create your code model

소스 코드에서 코드 모델을 추출합니다. 이 코드 모델은 엔티티(예: 소스 파일), 클래스, 함수와 그들 사이의 종속성을 포함합니다. Axivion은 소스 코드 프로젝트를 분석하여 이러한 코드 모델을 자동으로 구성할 수 있습니다.

### Map the code to your architecture

이제 코드 요소를 아키텍처 요소에 할당하여 코드 요소가 아키텍처 구성 요소와 어떻게 대응하는지 설정합니다. 이 과정은 제품의 구조와 아키텍처 모델에 따라 다음과 같이 수행할 수 있습니다.

- 수동 작업 : 예를 들어 Gravis 모델러와 같은 도구를 사용하여 직접 할당합니다.
- 자동화된 방법 : 이름 지정 규칙이나 계층 정보를 기반으로 하거나 모델에 포함된 정보를 활용하여 Axivion을 자동으로 사용합니다. 예를 들어, 태그된 값들을 활용할 수 있습니다.

### Interpret the dependencies

이 단계에서는 아키텍처 모델 내의 종속성을 해석하고 그 의미를 명확히 합니다. 즉, 아키텍처 종속성을 해석하여 코드의 종속성과 일치시키는 방법을 지정합니다. 예를 들어, UML 모델의 경우 종속성(필수/제공 인터페이스 등)을 보다 정교하게 해석할 수 있습니다.

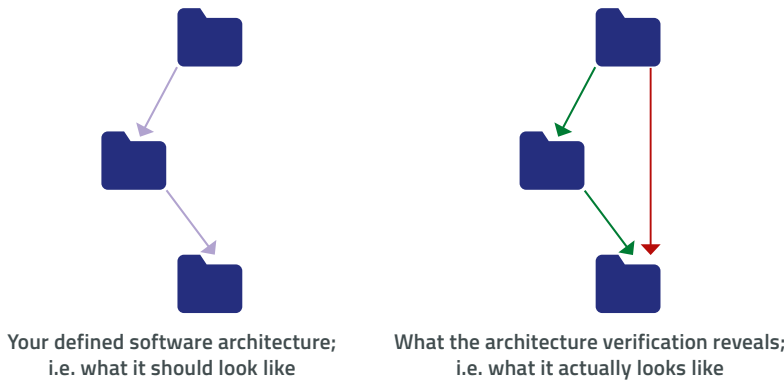
# Verify Your Architecture - Every Day

## Analyze your architecture

이제 모든 준비가 완료되어 Axivion을 사용하여 아키텍처를 검증할 수 있습니다. 코드를 분석하고 다음을 식별합니다.

- Convergences
- Divergences
- Absences

Benefit from sophisticated architecture verification.

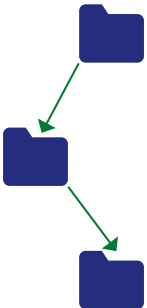


## What to do with the results

의도된 소프트웨어 아키텍처의 편차를 신뢰할 수 있을 정도로 철저하게 탐지하고 싶다면 자동화된 프로세스가 필수입니다. 그러나 이러한 발견에 대응하기 위해서는 사람의 능력이 필요합니다. 다음과 같은 선택지가 있습니다.

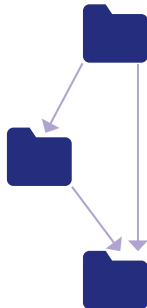
Make informed choices.

### 소스 코드 편차 수정



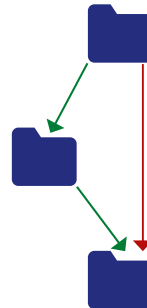
만약 이 편차가 실수라면, 코드를 수정하여 아키텍처와 다시 일치시켜야 합니다.

### 아키텍처 사양 업데이트



만약 이 편차가 실제로는 문제가 없는 것이라면, 그에 맞게 아키텍처를 업데이트해야 합니다.

### 일시적으로 편차 허용



만약 이 편차가 일시적으로 허용할 수 있는 수준이라면 그대로 두었다가 나중에 처리하세요.

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

## Continuous Analysis of Your Code

Axivion 아키텍처 검증은 단순히 현재 아키텍처에 대한 요약을 제공하는 것 이상으로 지속적인 소프트웨어 분석과 개발 프로세스의 핵심 요소가 될 수 있도록 설계되었습니다.

Axivion 아키텍처 검증을 한 번 세팅해두기만 하면, 그 후부터는 일상적인 검토 프로세스에 매끄럽게 통합이 가능합니다. 자동으로 감지되는 편차를 쉽게 확인하고 수정함으로써 코드가 항상 소프트웨어 아키텍처와 일치하도록 보장할 수 있습니다.

오류가 발생하기 쉬운 수동 검증을 신뢰할 수 있고 정확한 진단 정보로 대체한다면 개발자는 어떤 문제라도 조기에 발견하여 해결할 수 있습니다. 이는 비용이 많이 드는 사후 조정을 최소화하고 소프트웨어 변경 사항이 전체 시스템에 미치는 영향을 이해하는데 도움이 됩니다.

결과적으로, Axivion 아키텍처 검증은 더 긴 수명 주기를 위한 안정적인 소프트웨어 아키텍처를 제공합니다.

Freedom from Interference

## Architecture According to ISO 26262

ASIL 또는 QM 분류가 다른 여러 안전 관련 기능이 공통 하드웨어에서 공존할 수 있도록 하는 것은 최신 기술입니다. 그러나 이를 위해서는 ISO 26262에 따라 소프트웨어 프로젝트에 적합한 안전 아키텍처가 마련되어 있어야 합니다. 안전 아키텍처 준수는 간섭 방지를 위한 기초입니다.

Axivion 아키텍처 검증은 정의된 인터페이스와 선택한 통신 메커니즘의 일관된 사용을 보장합니다. 아키텍처를 벗어난 부분은 소스 코드에서 즉시 강조 표시됩니다. 이는 지정되지 않은 함수 호출, 데이터 덮어쓰기 또는 보다 일반적으로 인터페이스로 정의되지 않은 선언에 대한 참조를 포함합니다.

# 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.

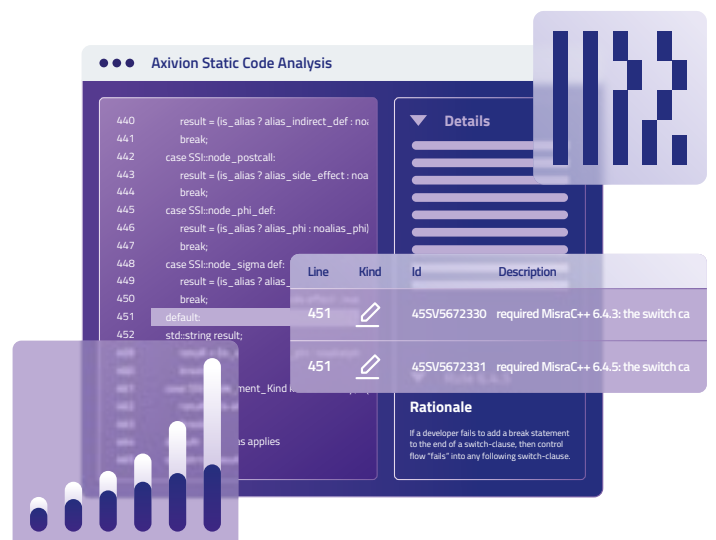
## Why stop here?

Axivion 아키텍처 검증은 소프트웨어 프로젝트의 견고한 기반을 구축합니다. Axivion 정적 코드 분석을 추가하면 코드를 심층적으로 분석하여 최고 수준의 표준도 충족시킬 수 있습니다.



웹사이트를 방문하여 더 자세한 내용을  
알아보고 데모를 체험해보세요.

[www.qt.io/axivion](http://www.qt.io/axivion)



# Qt Quality Assurance

큐티그룹(Qt Group)은 Qt 개발 프레임워크(Qt Framework) 를 상용 라이선스와 오픈소스 라이선스로 개발 및 배포하고 있습니다. Qt 개발 프레임워크를 이용하면 데스크탑과 임베디드 그리고 모바일용 앱에 이르기까지 다양한 운영체제와 플랫폼에 걸쳐 동작할 수 있는 애플리케이션을 제작할 수 있습니다. 전세계에서 백만 명이 넘는 Qt 개발자들이 차량용 인포테인먼트 시스템(IVI), 자동화 시스템(Automation), 의료(Medical), 가전 제품 및 군사용 소프트웨어에 이르기까지 70여 종류의 다양한 산업군에서 핵심적인 소프트웨어 개발에 Qt 개발 프레임워크를 사용하고 있습니다. Qt Group은 핀란드에 본사를 두고 있으며, 대한민국을 비롯해 미국, 독일, 일본, 중국, 러시아 등에서 지사와 연구소를 운영하고 있습니다. Qt 상용 라이선스 문의는 Qt 한국지사 전화번호 031-712-0045 혹은 이메일 [korea@qt.io](mailto:korea@qt.io) 로 문의 주시기 바랍니다. Qt와 관련한 보다 자세한 안내는 웹페이지 [www.qt.io](http://www.qt.io) 에서 확인하실 수 있습니다.