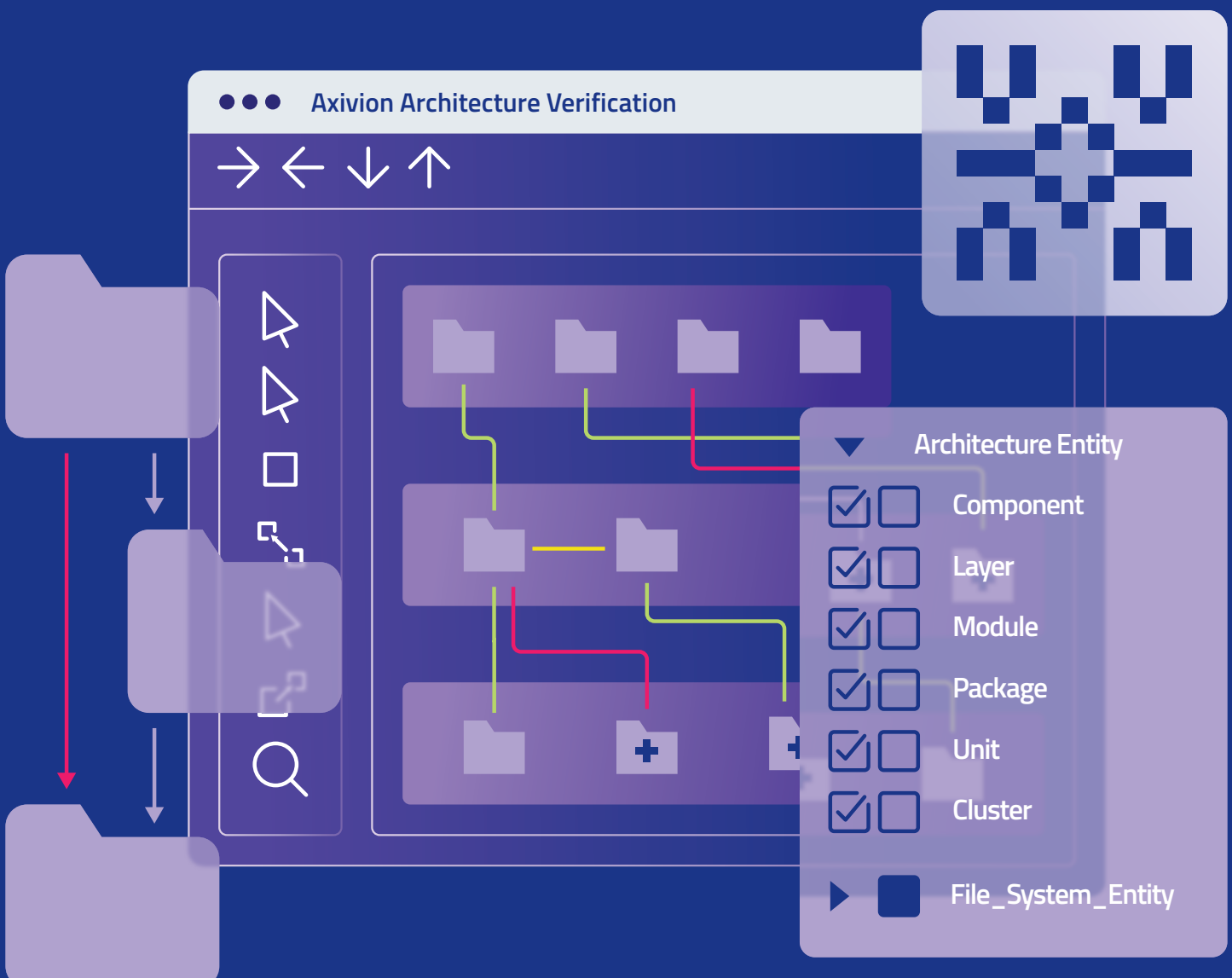
Qt Quality Assurance

Axivion アーキテクチャ検証

長期的に利用できるソフトウェアの基盤を作る



Axivion アーキテクチャ検証

アーキテクチャを検証する理由

コードがソフトウェアアーキテクチャに適合していることを確認

ソフトウェアアーキテクチャとデザインはコードと一致している必要があります。これにより、ソフトウェアアーキテクチャを新機能の影響を評価する際の基準として使用でき、長期的な目標に基づいた計画的な製品開発が可能になります。

Axivionアーキテクチャ検証は、コードがアーキテクチャに準拠していることを確認します。機能アーキテクチャに加え、このツールは安全性やセキュリティアーキテクチャの仕様（Freedom from Interferenceなど）についてもレビューし、準拠性をチェックします。

利点

信頼性のあるドキュメントを使用して意思決定を行う

- エラーの多い手動レビューを、徹底的で信頼性の高い自動アーキテクチャ検証に置き換え
- プロジェクトの設計と実装に関する重要なアイデアを捉え、ソフトウェアの信頼性の高い基盤を作成
- アーキテクチャを理解するためにコードを読む必要を排除
- 新しいチームメンバーのオンボーディングを改善し、最初から効率的に作業可能
- アーキテクチャを活用して、コードの変更がどのような影響を及ぼすかを理解し、日常の開発やリファクタリング、新機能の追加などの大規模な変更に対応
- コードへの変更がソフトウェアアーキテクチャに違反しないようにし、アーキテクチャの劣化を防ぎ、ソフトウェアのライフサイクルを延ばす

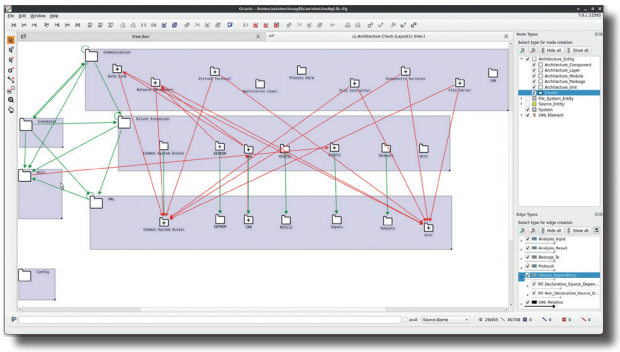
主要な機能

確立されたプロセスにアーキテクチャの検証を簡単に統合する

- 統合されたモデラー
  - UMLツールへのインターフェイス
  - AUTOSAR XMLインポート(ARXML)
  - FFI (Freedom From Interference)
- アーキテクチャの再構築
- 安全およびセキュリティのためのアーキテクチャビュー

ソフトウェア・アーキテクチャの再ドキュメント化

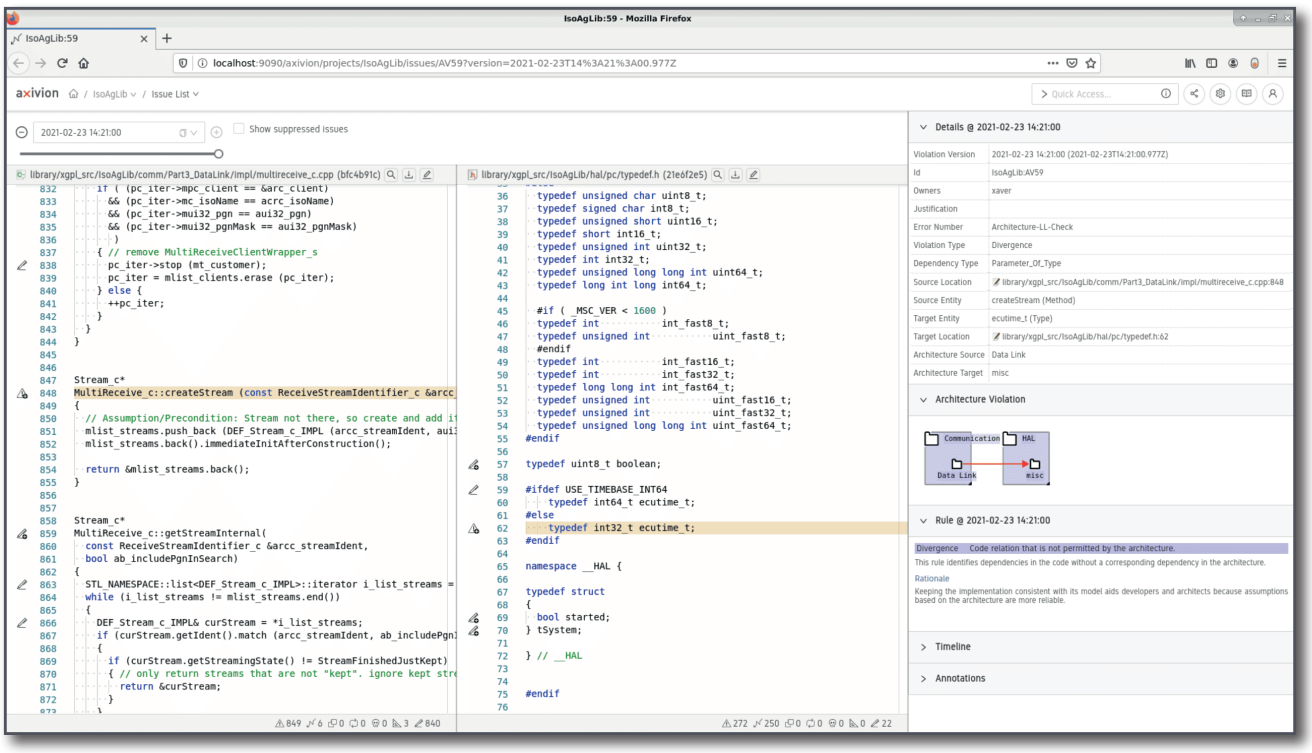
Axivionアーキテクチャ検証は、プロジェクトに信頼性の高い基盤を提供します。アーキテクチャモデルが存在する場合（CASEツール、XMI、XMLなど）、ドキュメントのみがある場合（Word、PowerPointなど）、またはドキュメントが全くない場合（稼働中のプロジェクトや完了したプロジェクトなど）でも、ソフトウェアアーキテクチャを活用して、ソフトウェアの概要や変更が及ぼす影響を簡単に説明できるようにします。特にサードパーティコードを統合する際に役立ちます。



既存開発環境へのシームレスな統合

新しいツールを導入する際には、使いやすさが重要です。Axivionアーキテクチャ検証は、ニーズに応じてカスタマイズされ、既存の開発環境にスムーズに統合されます。豊富なカスタマイズオプションにより、独自に策定したルールやワークフローに合わせたアーキテクチャ検証が可能です。セットアップ全体を通じて専門家がサポートし、期待以上の成果を実現します。

日々のレポートにより、新たな問題を迅速に特定し、影響を受けたコード部分に直接アクセスして必要な修正が行えます。差分解析により、コード内の新しい問題を即座に確認できます。複数のスプリントやプロジェクト全体などの長期間レビューする際には、スマートなフィルタリングオプションが特定の問題に焦点を当てるのに役立ちます。



初期設定は思ったより簡単

たったの数ステップで簡単に設定

Axivion アーキテクチャ検証を使えば、信頼性の高いアーキテクチャの構築と維持が簡単に行えます。設定後は、システムが自動的に早期に差異を検出し、アーキテクチャに影響が出る前に修正が可能です。

アーキテクチャモデル作成

まず、アーキテクチャ図を作成します。既存の機械可読モデルをインターフェースを通じてインポートするか、内蔵の作成ツールを使って作成できます。

コードモデル作成

ソースコードからコードモデルを抽出します。このコードモデルは、エンティティ（例：ソースファイル）、クラスや関数、そしてそれらの依存関係を表します。Axivionは、ソースコードプロジェクトを分析することで、コードモデルを自動的に構築できます。

コードをアーキテクチャにマッピング

次に、コード要素とアーキテクチャコンポーネントの対応を確立します。コード要素をアーキテクチャ要素に割り当てる方法は、製品構造やアーキテクチャモデルに応じて以下のいずれかで行えます：

- 手動（例：Gravisモデラーを使用）
- 自動（命名規則や階層情報を利用）
- モデル内の情報を使用（例：タグ付き値）

依存関係を解釈

このステップでは、アーキテクチャモデル内の依存関係を解釈し、それが何を意味するかを指定します。言い換えれば、アーキテクチャの依存関係をコードの依存関係に一致させる方法を解釈します。例えば、コンポーネントAはコンポーネントB内の関数を呼び出すことが許可されている、という単純な例があります。
もちろん、特にUMLモデルではより高度な依存関係の解釈（要求されるインターフェースや提供されるインターフェースなど）が可能です。

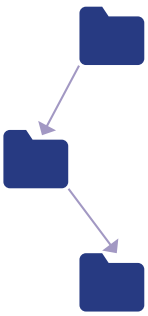
日々のアーキテクチャチェック

アーキテクチャ検証

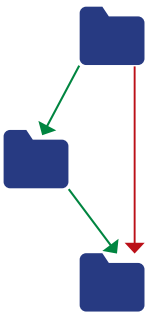
準備が整ったら、Axivionにアーキテクチャを検証させます。これにより、以下の点を検証して特定します：

- 一致 (Convergences)
- 不一致 (Divergences)
- 欠如 (Absense)

高度なアーキテクチャ検証の利点を活用する



仮説に基づくアーキテクチャモデル



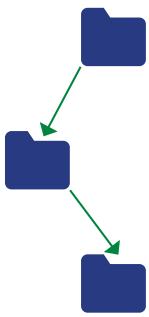
仮説と実装の一致を検証

結果の活用方法

意図したソフトウェアアーキテクチャからの差異を検出するには、自動化プロセスが最も効果的で信頼性があります。しかし、これらの結果にどのように対応するかを決定するには人間の判断が必要です。選択肢は以下の通りです：

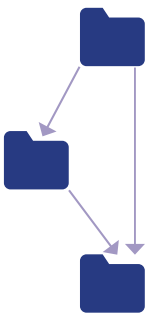
情報に基づいた意思決定を行う

ソースコードを修正する



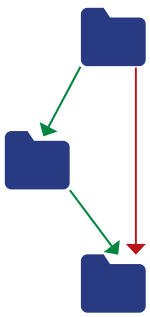
コードが誤りである場合は、コードを修正してアーキテクチャと一致させる必要があります。

アーキテクチャ仕様を修正する



コードが正当である場合は、アーキテクチャをそれに合わせて更新する必要があります。

差異を一時的に受け入れる



差異を一時的に受け入れ、将来の修正を計画します。

コードの継続的な分析

アーキテクチャ検証をコードの継続的な解析に組み込む

Axivion アーキテクチャ検証は、一時的なアーキテクチャのスナップショットを提供するものではありません。これは、継続的なソフトウェア分析と開発プロセスに組み込まれるように設計されています。

設定後は、Axivion アーキテクチャ検証 を日常のレビュープロセスに簡単に統合できます。自動的に検出された差異を確認し、必要に応じて修正を加えるだけで、コードが常にソフトウェアアーキテクチャと一致することが保証されます。

手動によるエラーの多い検証を排除し、信頼性の高い解析情報で置き換えることで、開発者は早期に問題に対処できます。これにより、後の段階での高額な修正を避け、ソフトウェア変更の影響を事前に把握できます。

結果として、長期間にわたって信頼性の高いソフトウェアアーキテクチャが実現します。

ISO 26262 準拠のアーキテクチャ

Freedom from Interference

異なるASILやQM分類の複数の安全関連機能を共通のハードウェア上で共存させることが可能です。しかし、これにはISO 26262に準拠した適切なソフトウェアアーキテクチャが必要です。安全アーキテクチャの遵守が、干渉のない状態を確保する基盤となります。

Axivion アーキテクチャ検証は、定義されたインターフェースと選択された通信メカニズムの一貫した使用を保証します。アーキテクチャからの差異はソースコード内で即座に強調表示され、未指定の関数呼び出し、データの上書き、またはインターフェースとして定義されていない宣言への参照などが検出されます。

基本仕様

Axivion Suite 7.8 の機能の一部を抜粋しています。技術仕様詳細についてはお問い合わせください。

対応言語とコンパイラ

言語	C, C++, C#

プラットフォームOS

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)

プラグイン

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

対応 UML® ツール

UML® ツール	IBM Rational Rhapsody, Sparx Enterprise Architect (via XMI or .qea-files), PlantUML

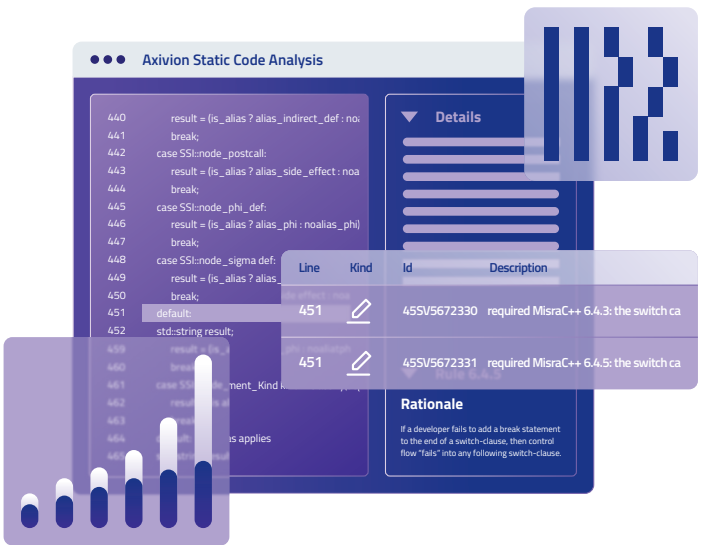
その他

対応ブラウザ	Microsoft® Edge, Mozilla Firefox®, Google Chrome™
要件	Python (3.8.1 - 3.12) Java® Runtime (8, 11-14 and 17)

Axivion Suite

Axivion アーキテクチャ検証は、ソフトウェアプロジェクトの堅実な基盤を提供します。これに加えて、Axivion 静的コード解析を利用することで、コードの詳細な分析が行え、最高水準の要件を満たすことが保証されます。

詳細情報やデモの予約は、当社のウェブサイトをご覧ください。
www.qt.io/ja-jp/axivion



Qt Quality Assurance

Qt Group（Nasdaq Helsinki: QTCOM）は、世界のリーダー企業や150万人以上の開発者から信頼されるグローバルなソフトウェア企業として、ユーザーに愛されるアプリケーションやスマートデバイスの開発を進めるお客様をサポートしています。また、UIデザインからソフトウェア開発、品質管理・展開に至るまで、製品開発ライフサイクル全体を通して、お客様の生産性向上を支援しています。

Qt Groupは、180カ国以上、70を超える業界のお客様へソリューションを提供しています。フィンランドのエスポーに本社を置き、世界中に約700人の従業員を擁しています。

www.qt.io/ja-jp/axivion