

# Kiuwan Architecture

- [Overview](#)
- [What are the benefits of using Kiuwan Architecture](#)
- [How does Kiuwan Architecture work](#)
- [Next Steps](#)

## Overview

Companies of all shapes, sizes, and colors need to develop software to support their business. No matter what the case is (in-house vs. third-party development, application portfolio size, development technologies, etc.), all have the same needs with different levels of complexity:

- Decrease application time-to-market
- Reduce application downtime
- Increase application comprehension for developers
- Decrease application maintenance and support costs



**Kiuwan Architecture** makes the Kiuwan platform a unique solution for the enterprise, providing different stakeholders with a comprehensive view, trends and insights of their application portfolio, allowing them to focus resources on areas of critical concern, starting from the initial phase of production all the way to delivery.

Please, visit our blog post on [How Application Inventory Management Unlocks Your App for Affordable Maintenance and Development](#)

## What are the benefits of using Kiuwan Architecture

Kiuwan Architecture provides a corporate view of an enterprise's software assets allowing this information to be accessed at different levels. From a single program or file up to the whole enterprise level.

- *Detects and surfaces the dependencies in your applications*, giving you the ability to truly understand your architecture and application's structure
- Produce and correlate with pinpoint accuracy a *detailed report on application components* to inform your business impact decisions
- Understand in a visual way the *complexity behind changes* in your applications
- Simplify and rationalize IT environments, *identifying those areas of value* within the application portfolio
- *Run impact analysis* for typical application maintenance tasks: planning, change requests and new functionality etc.
- *Evaluate the collateral effects and negative impacts* associated with the implementation of new specifications of the systems.
- *Support application migration and modernization* initiatives with the right information in your hands.
- *Verify that the application's software elements are compatible with the architecture normative* (such as illegal dependencies detection etc).
- Develop *obsolescence analysis to detect the components that are not used* by the system that consume resources and are a source of potential errors.



This is possible thanks to **Kiuwan Architecture's ability to generate the complete map of your applications' software components and relationships**.

## How does Kiuwan Architecture work

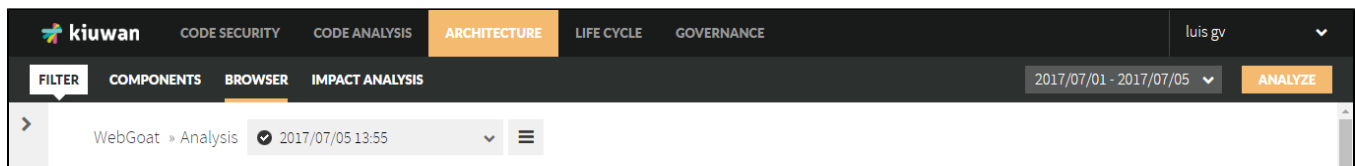
**Kiuwan Architecture automatically builds a map of a software system** (or "Application" in Kiuwan terms)

This is done by analyzing the source code and applying extraction rules in order to identify and insert into the model all **components** (*artifacts*) and **relationships** (*dependencies*).

That model is uploaded to Kiuwan cloud so you can use Kiuwan Architecture dashboards and pages to fully exploit all the above-mentioned functionalities.

Once the model is uploaded, you can go to Kiuwan Architecture module, select the application, and inspect the model by making use of the three available submodules:

- **Components**: a search page of application components
- **Browser**: a graphical view of components and dependencies
- **Impact Analysis**: a wizard to compose impact analysis queries



## Next Steps

For a full understanding of Kiuwan Architecture, we encourage you to carefully read and follow next sections:

- [Getting Started](#)
- [Components](#)
- [Browser](#)
- [Impact Analysis](#)

For didactic purposes, we will follow some examples based on WebGoat 7.1. source code. You can get it at <https://github.com/WebGoat/WebGoat/releases>