

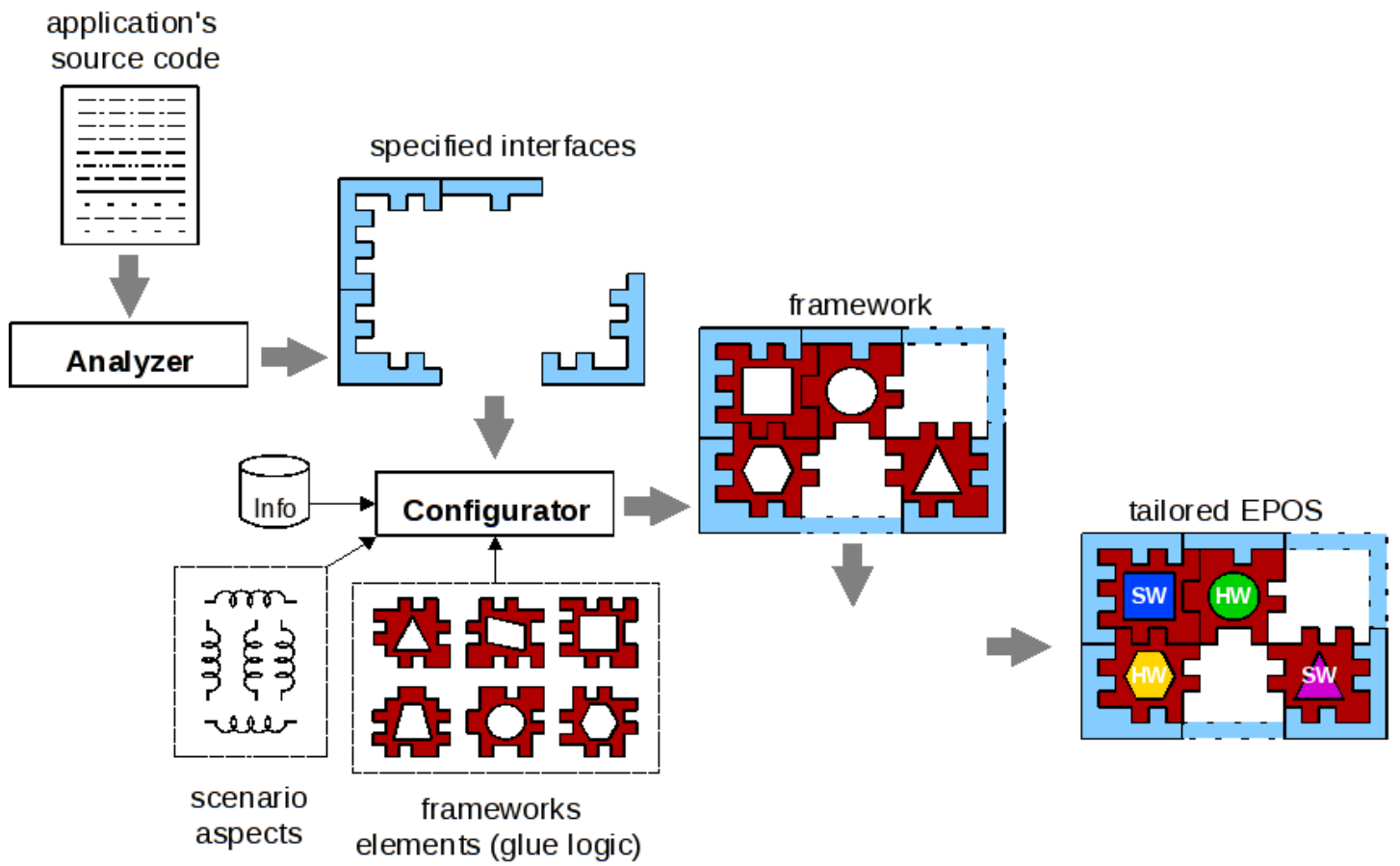
# EPOS Overview

The **EPOS Project** (*Embedded Parallel Operating System*) aims at automating the development of embedded systems so that developers can concentrate on what really matters: their **applications**. EPOS relies on the **Application-Driven Embedded System Design** (ADESD) method to guide the development of both software and hardware components that can be automatically adapted to fulfill the requirements of particular applications. Additionally, EPOS features a set of tools to select, adapt and plug components into an application-specific framework, thus enabling the automatic generation of an application-specific embedded system instances.

In order to deliver each embedded application a tailored computing system, EPOS takes on software components, which implement a set of scenario-independent system abstractions that can latter be adapted to a given execution scenario with the aid of scenario adapters. These abstractions are collected in a repository and are exported to the application programmers via inflated interfaces. This strategy, besides drastically reducing the number of exported abstractions, enables programmers to easily express their application's requirements regarding the computing system to be generated.

A particular kind of system abstractions, namely hardware mediators, are responsible for keeping higher level abstractions architecture-independent. One could imagine these components as parts of a configurable hardware abstraction layer. When combined with programmable logic intellectual properties, hardware mediators yield a customized software-hardware interface that may even give rise to a dedicated hardware platform. That is, a hardware platform that includes only the features needed to support the associated application.

An application designed and implemented according to the strategy proposed by EPOS can be submitted to a tool that will proceed syntactical and data flow analysis to extract a blueprint for the operating system to be generated and, when instantiated, the high-level description for the hardware platform to be synthesized. The blueprint is then refined by dependency analysis against information about the execution scenario acquired from the user via visual tools. The outcome of this process is a set of selective realize keys that will support both the compilation of the application-oriented operating system and synthesis of hardware components into a programmable logic device.



## EPOS Overview Presentation