

# The Fractal OW2 Project

**F4E** is an Eclipse based environment for developing Fractal applications.

**Main contributors:**  
 France Telecom R&D  
 INRIA  
 STMicroelectronics

**Fractal Distribution** provides a ready to use package containing a collection of Fractal tutorials, and the whole libraries needed to develop Fractal applications.

**Fractlet**  
Java annotations for the Fractal component model.

**FScript/ FPath**  
Domain-Specific Language (DSL) to program dynamic reconfigurations of Fractal architectures.

**Fractal Explorer**  
Generic graphical console for managing Fractal based applications. It allows to discover, introspect, manage, monitor and reconfigure Fractal applications at runtime.

**Fractal RMI**  
provides a set of components to create distributed bindings between Fractal components.

**Fractal JMX** allows the JMX management of Fractal applications through automatic exposition in JMX agents.

**Binding Factory** provides support for establishing remote bindings between Fractal components using diverse protocols.

**Behaviour Protocols Checker** allows to define, validate behavior protocols associated to Fractal interfaces.

**Cecilia** is a development environment for programming Fractal components with the C programming language

**Fractal ADL** is the base architecture description language of the Fractal component model. It provides a XML DTD for describing (for instance) component types, component implementations, component hierarchies and component bindings.

**Java Implementations**

- Julia** is the reference implementation of the Fractal component model. Using Julia it is possible to deploy a Fractal application relying on Julia runtime code generation of classes, interfaces, controllers. These tasks are accomplished internally using the ASM bytecode manipulation framework.
- AOKell** it is an implementation where controllers are implemented with a component-based approach. The glue between the application and the control levels are done using AOP (aspect-oriented programming). This glue can be created with Spoon or AspectJ.
- JuliaC** It allows generating and compiling the source code of the infrastructure (so called membranes) which is needed to run a Fractal/Java application. The application can then be run without having to use ASM to generate on-the-fly classes for interfaces, controllers and merge strategies.

**Fractal Java API**  
A language and message format that the implementations rely to provide interoperability between Java components and that Fractal programs use to communicate with the implementations.

**Fractal C API**