

The SODALITE Architecture

Elisabetta Di Nitto
Politecnico di Milano

The back-end for practically any new software application domain, from healthcare to aerospace, is in the data centre. Within such data centres, there are multiple and diverse computational resources ranging from public and private Cloud to HPC systems and exploiting a variety of components, such as accelerators, GPUs, configurable processors, and non-x86 CPUs such as ARMv8.

Managing applications on such composite and heterogeneous systems is hard. It requires different skill sets, depending on the targeted resources, the chosen configuration, and deployment technologies and is nowadays accomplished through the creation of software-defined infrastructures. SODALITE is addressing the gap emerging within such software-defined, application-specific infrastructures. It will provide developers and infrastructure operators the tools that simplify application and infrastructure description and suggest the best deployment scenarios. This will enable DevOps teams to deploy their applications in a simpler and faster way.

The goal of SODALITE is to provide design time support for optimal application deployment configuration and an execution environment for heterogeneous applications that is able to evolve based on the diverse circumstances occurring on heterogeneous, software-defined, high-performance, cloud, HPC and GPUs infrastructures.

In [1] we have announced the actors and use cases relevant to SODALITE. Here we shortly present the high level architecture that drives the development of the SODALITE platform. This is divided into three main layers, the Modeling layer, the Infrastructure as Code (IaC) layer, and the Runtime layer. Figure 1 shows these layers together with their relationships defined in terms of offered and used interfaces. The Modeling layer exploits the interfaces provided by the other two layers to offer to the end users (Application Ops Experts, Resource Experts and Quality Experts) the needed information concerning the application deployment and runtime configuration. In turn, it offers to the other layers the means to access the ontology and the application deployment models through the Semantic Reasoner API. The IaC layer offers to the modeling layer the APIs for preparing the deployment, for verifying the IaC and for predicting defects. Finally, the Runtime Layer offers the APIs for controlling the orchestration of an application deployment and for monitoring the status of the system. In turn, this layer relies on the APIs offered by the underlying technologies, in particular, Torque, OpenStack and Kubernetes as shown in Figure 1.

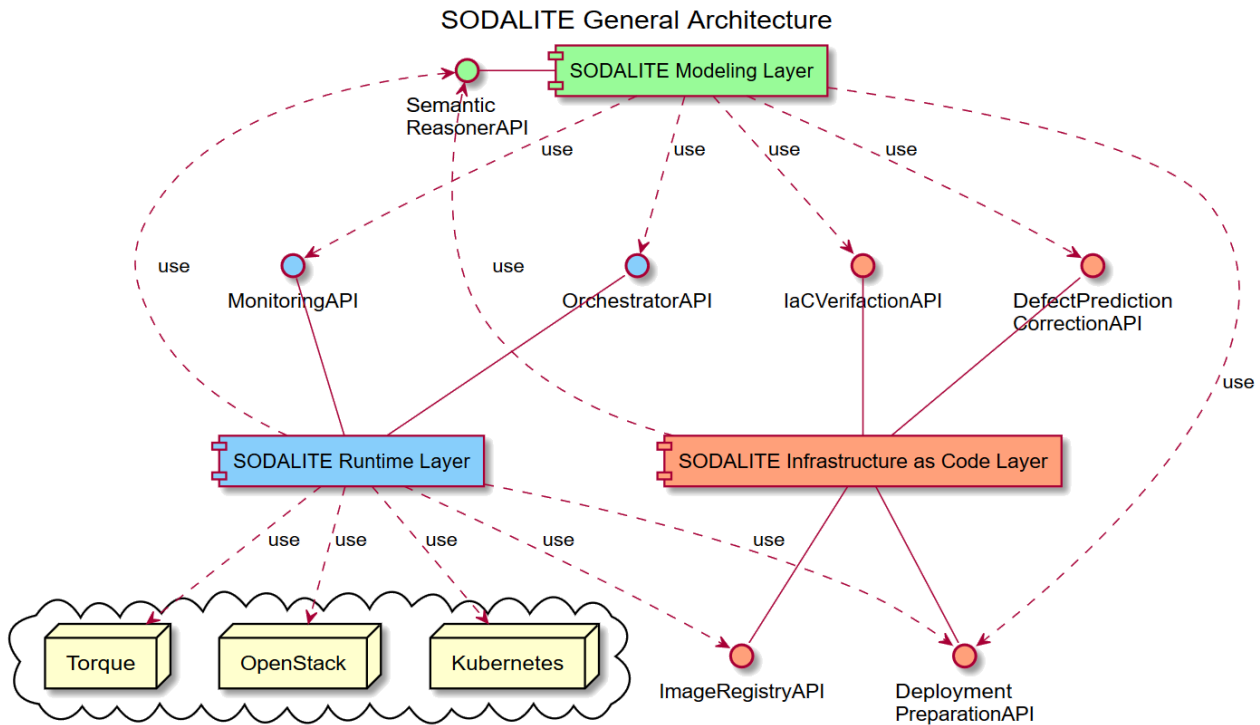


Figure 1: SODALITE Overall Architecture.

The SODALITE architecture is being currently implemented and the first components have been released and are available as open source projects (<https://github.com/SODALITE-EU>). They constitute a laboratory prototype to validate the specifics of the architecture and to test the underlying components within the context of three critical application domains, that is, simulation of the impact of spinal prostheses on the human body, measurement of hydric resources in specific geographical areas, and supporting optimized execution of vehicle assistants. Please visit us on github and get in touch with us if you want to contribute to improving the SODALITE results.

References

- [1] Luciano Baresi, Elisabetta Di Nitto, SODALITE actors and use cases, SODALITE blog post <https://www.sodalite.eu/content/sodalite-actors-and-use-cases>