

PIACERE Integrated Development Environment

Gorka Benguria Elguezabal (tecnalia)

gorka.benguria@tecnalia.com

PIACERE Project

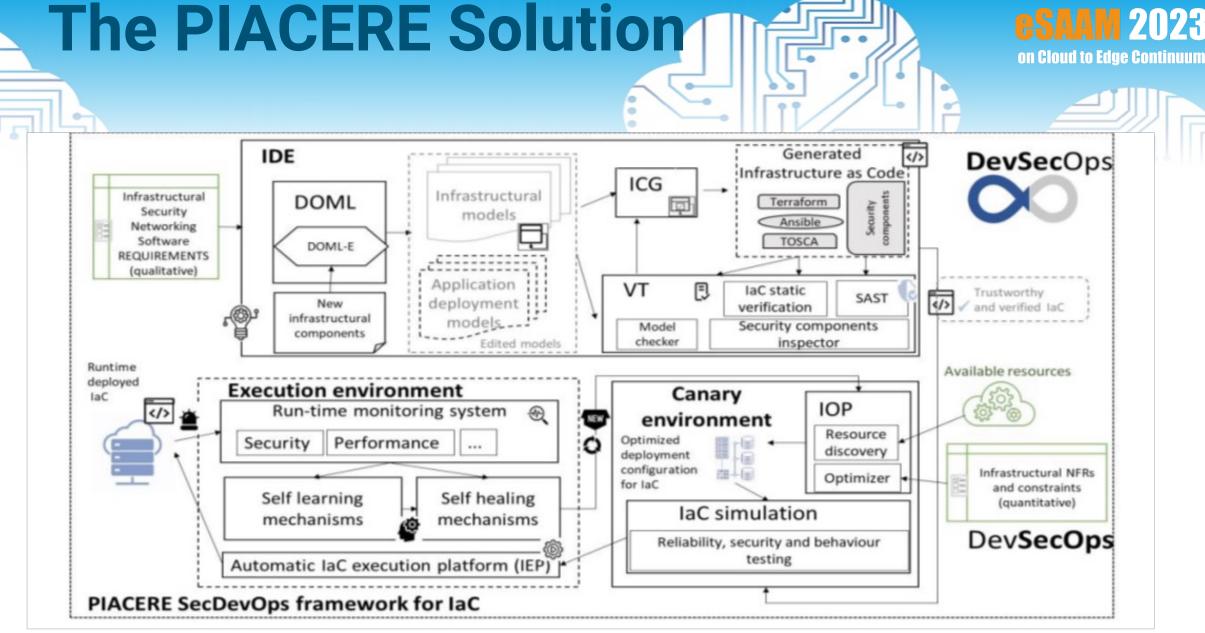


Vision: DevSecOps framework for the **development**, **deployment** and **operation** of **trustworthy** infrastructure-as-code.

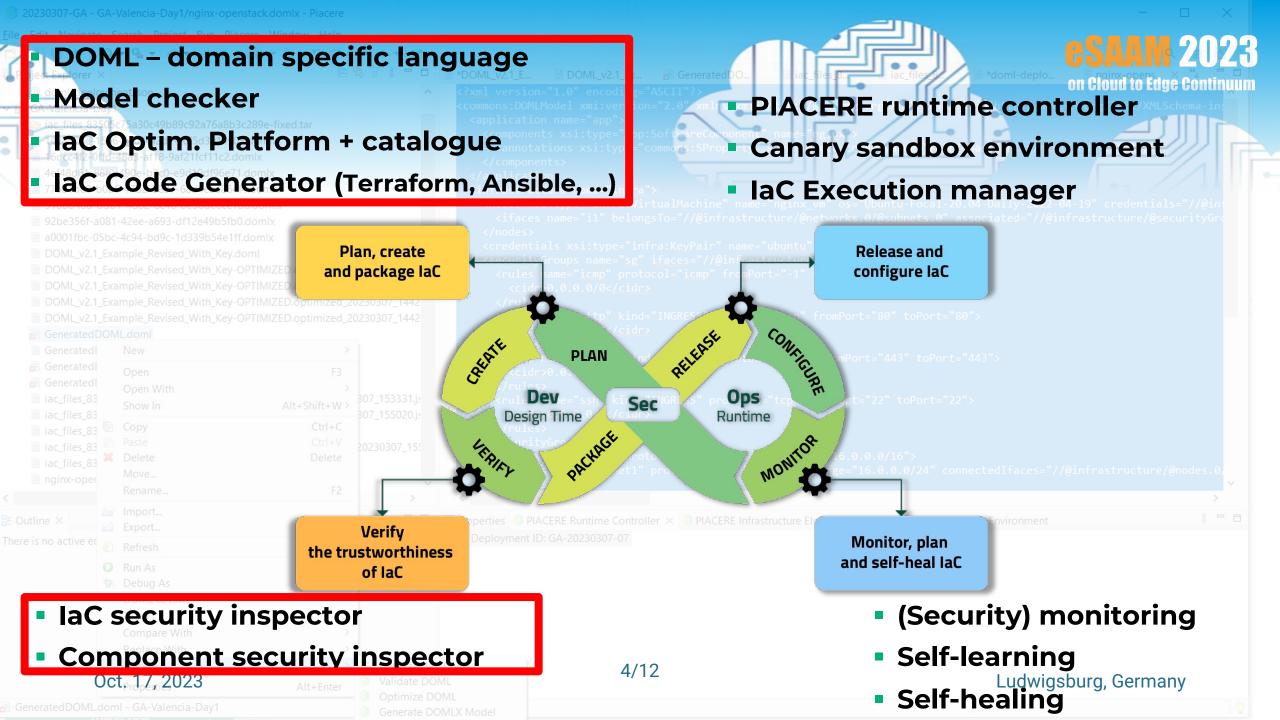
Goal: Framework with tools integrated in the IDE.

Status: PoC version already available!





3/12



DevSecOps Modelling Language (DOML



- End-user declarative language enabling the modelling of provisioning, deployment, and configuration of infrastructure
- multi-layer approach. An application can be described in four layers: application layer, abstract infrastructure layer, concrete infrastructure layer and optimization layer.
- Extensible
- EMF based and using Xtext Framework
 - Enables DOMLX to facilitate integration with other technologies

DevSecOps Modelling Language (D



eSAAM 2023 on Cloud to Edge Continuum

```
doml amurica
properties {
   entorno="pre"
   proyecto="amurica"
// Application Definition
application amurica {
   software_component Gestaut {
       // Interfaces should be separated by commas, but the IDE needs to be fixed
       provides { http, https }
       consumes { dbAccess, search }
   software_component ElasticSearch {
       provides { search }
       consumes { dbAccess }
   software_component Edi {
       provides { edi }
       consumes { dbAccess }
   dbms postgres {
       provides {
            dbAccess
infrastructure abstractInfra {
   // Networks
   net vpc {
       cidr "10.100.0.0/16"
        protocol "TCP/IP"
        subnet subnet1 {
```

IDE (Integrated Development Environment)



- Enables end users to access all piacere tools (design time and runtime)
- Two main integration approach depending on the needs of the tools: context menus and custom views
- Additional elements added to facilitate user experience
 - Preferences
 - Perspectives
 - Project type

VT (Verification Tools)





- Validate DOML models and verifying their consistency and correctness
- It is a service
- IDE communicates with it through a REST API, leveraging the DOMLX format
- Powered by the Z3 Theorem Prover and built with Python

IaC Security Inspector

- After the design phase
- Checks cover syntactic problems of the IaC languages
- Checks IaC component and their dependencies

IOP (PIACERE Optimizer Infrastructure)

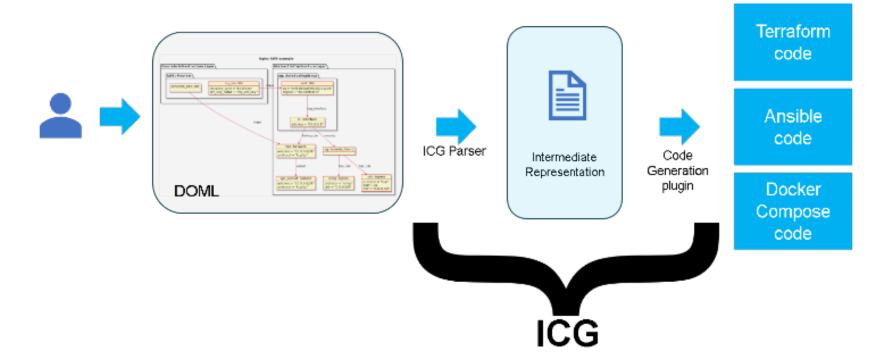


- find the optimum infrastructure to be deployed considering
 - The data provided as input
 - The infrastructure catalogue
- Objectives considered
 - Cost
 - Availability
 - Performance
- Different requirements possible
 - maximum cost for the overall configuration
 - a minimum performance
 - •
- Relays on multiobjective algorithms NSGA

ICG (Infrastructural Code Generator)



Translates DOML into IaC supporting currently terraform and ansible languages



Conclusion





- The motivation of the IDE was to
 - Facilitate the usage
 - Cover all phase in a single point
 - Increase the quality
- We have presented the Design time support of the IDE
 - There is a runtime support as well
- We are in the final tunning of the environment through the applications in three use cases
 - Transport
 - Government
 - Real time applications



Thank you!



www.piacere-project.org























Thank you!

Sponsored by:









Organized by:





