ELISA Enabling Linux in Safety Applications

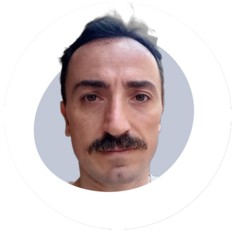
WORKSHOP

NASA Goddard

# Linking external test results to test cases in BASIL to support pre existing test infrastructures

Luigi Pellecchia - Red Hat

## Who I am



**Luigi Pellecchia**
*Principal*
*Software Quality Engineer*
Quality Engineering
In-vehicle OS
Red Hat

## Agenda

- What is BASIL
- BASIL Embedded Test Infrastructure
- How to use BASIL with pre existing test infrastructure
  - Gitlab CI
  - Github Actions
  - KernelCI (demo)
  - Testing Farm (demo)

ELISA Enabling Linux in Safety Applications

WORKSHOP

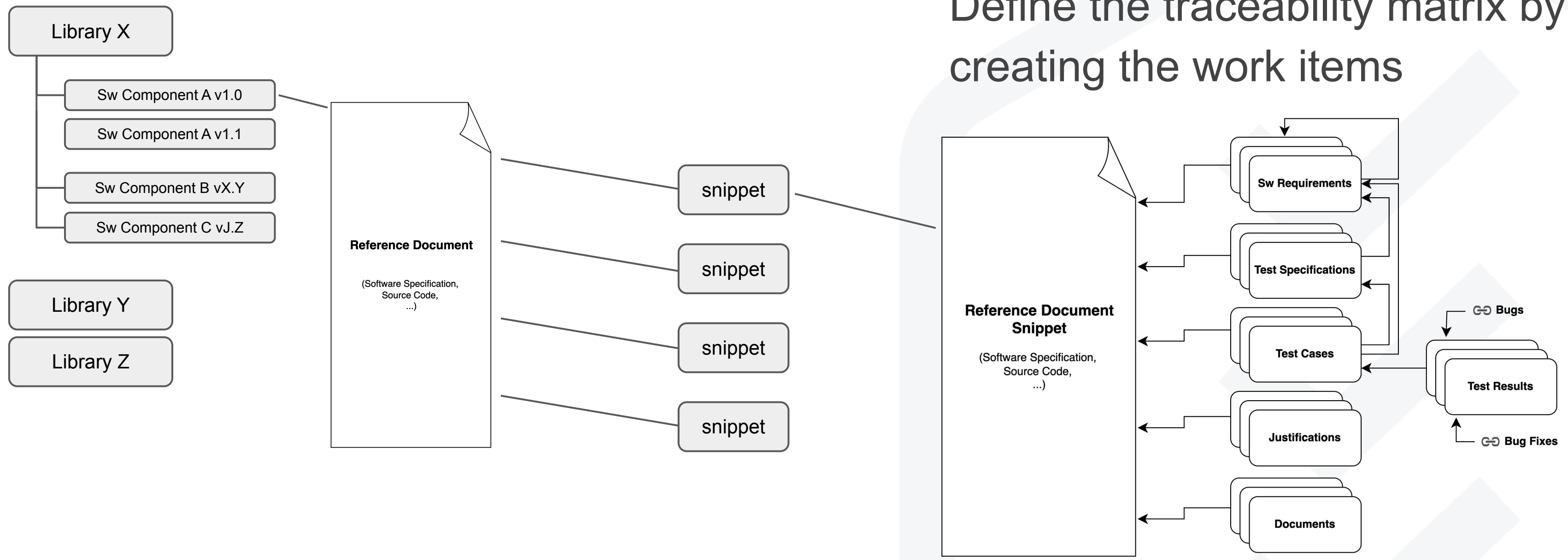
## BASIL The FuSa Spice

Tool developed to manage software related work items, design their traceability towards specifications and ensure completeness of analysis

- Born at Red Hat to support RHIVOS Functional Safety ISO 26262 Compliance Certification
- BASIL name comes from ASIL B
- Presented to ELISA Project on June 2023 during the Berlin Workshop
- Open Sourced and hosted at ELISA github

# BASIL The FuSa Spice

Define the traceability matrix by creating the work items



ELISA Enabling Linux in Safety Applications

WORKSHOP

## BASIL The FuSa Spice - key points

- Web App with user management
- Clarifies the gaps
- Support collaboration through comments, notifications and work item workflow
- Multiple mapping views to parallelize teams work
- Follow the project evolution
- Allow integration in CI and automated workflows via REST API
- Simplified deployment via containers

ELISA Enabling Linux in Safety Applications

WORKSHOP

## What we are trying to solve?

- Simplify the introduction of BASIL in companies and projects with pre existing test infrastructures
- Leverage open source and community-oriented test infrastructures initiatives

## KernelCI

KernelCI is a community-based open source distributed test automation system focused on upstream kernel development.

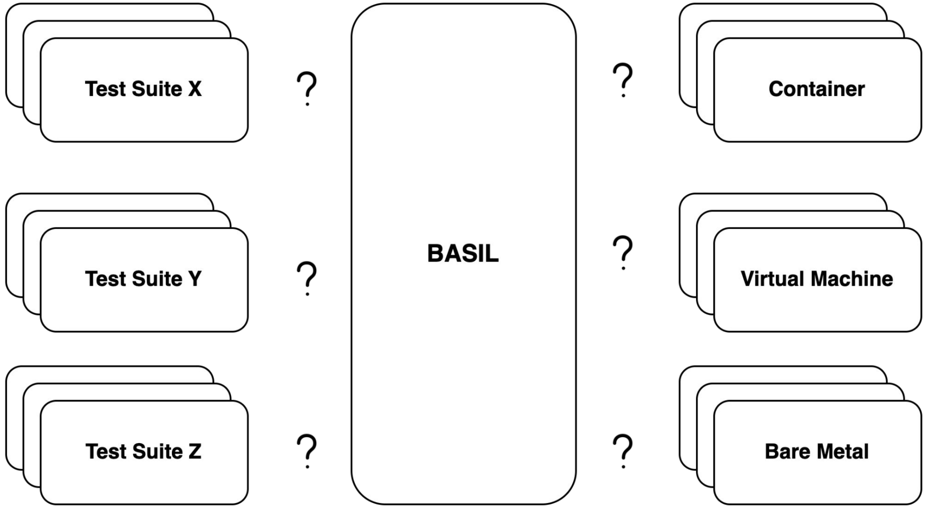
A Linux Foundation project

## Testing Farm

Testing Farm is a reliable and scalable Testing System as a Service.

A Red Hat project

# BASIL Embedded Test Infrastructure



tmt (Test Management Tool)

Python project that uses fmf metadata file (yaml) to abstract test case, test plans and user stories

Can provision different target environments

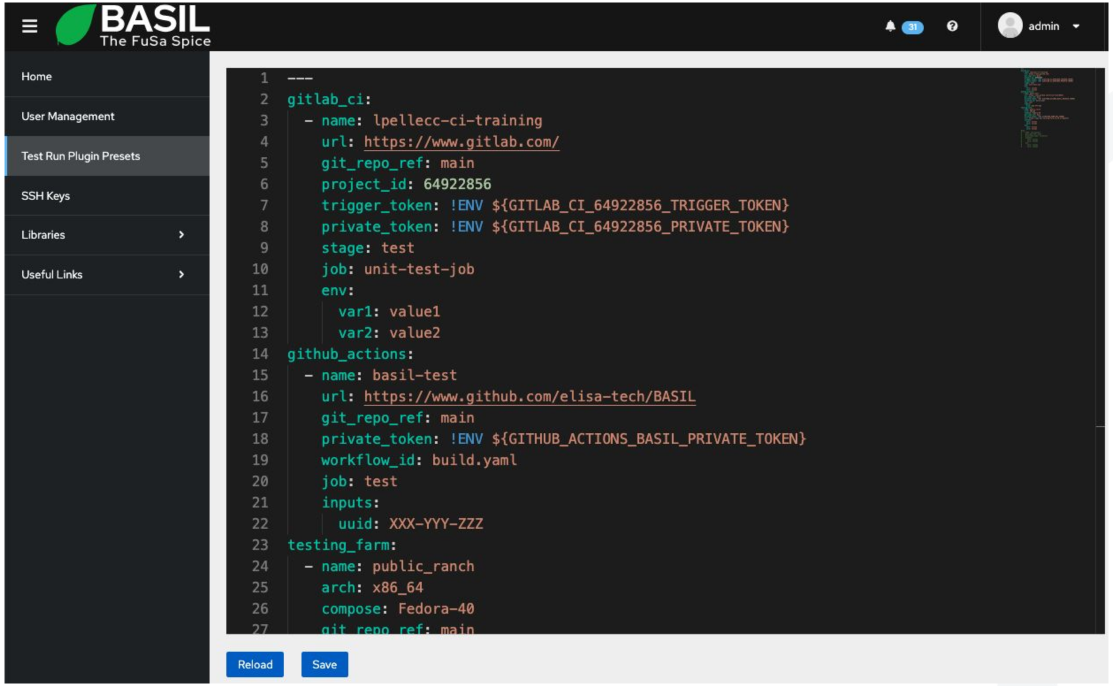
# BASIL Plugin based Test Infrastructure

| Test Infrastructure | Trigger and Trace | Trace pre existing runs | Test Infrastructure | BASIL Version |
|---|---|---|---|---|
| tmt | ✅ | ❌ | Embedded | 1.4 |
| Gitlab CI | ✅ | ✅ | External | 1.5 |
| Github Actions | ✅ | ✅ | External | 1.5 |
| KernelCI | ❌ | ✅ | External | 1.5 |
| Testing Farm | ✅ | ❌ | External | 1.5 |

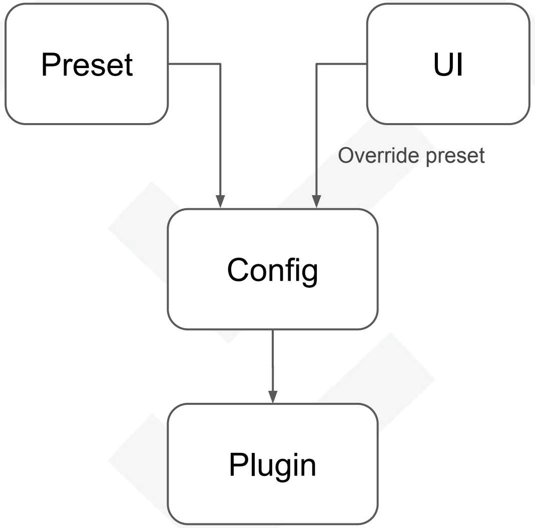
ELISA Enabling Linux in Safety Applications

WORKSHOP

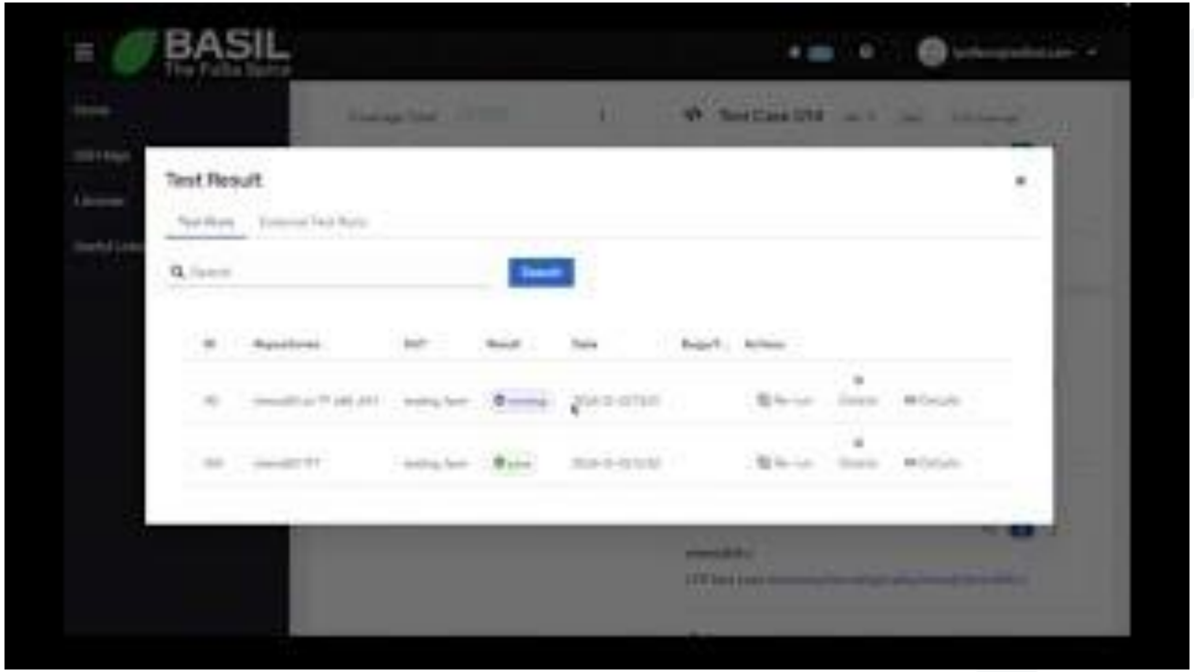
# Plugins Presets



```
---
gitlab_ci:
  - name: lpellecc-ci-training
    url: https://www.gitlab.com/
    git_repo_ref: main
    project_id: 64922856
    trigger_token: !ENV ${GITLAB_CI_64922856_TRIGGER_TOKEN}
    private_token: !ENV ${GITLAB_CI_64922856_PRIVATE_TOKEN}
    stage: test
    job: unit-test-job
    env:
      var1: value1
      var2: value2
github_actions:
  - name: basil-test
    url: https://www.github.com/elisa-tech/BASIL
    git_repo_ref: main
    private_token: !ENV ${GITHUB_ACTIONS_BASIL_PRIVATE_TOKEN}
    workflow_id: build.yaml
    job: test
    inputs:
      uuid: XXX-YYY-ZZZ
testing_farm:
  - name: public_ranch
    arch: x86_64
    compose: Fedora-40
    git_repo_ref: main
```



## Demo: Testing Farm



## Demo: KernelCI

ELISA Enabling Linux in Safety Applications

WORKSHOP

# Considerations and Next Steps

- KernelCI Plugin is configured to use Maestro API → We should move to a KCIDB oriented implementation

- Some test is configured with not fine granularity (e.g.: ltp syscalls), its up to the user to define the granularity required by its own purpose

## Thanks

ELISA
Enabling Linux in Safety Applications

WORKSHOP

# Licensing of Workshop Results

All work created during the workshop is licensed under Creative Commons Attribution 4.0 International (CC-BY-4.0) [https://creativecommons.org/licenses/by/4.0/] by default, or under another suitable open-source license, e.g., GPL-2.0 for kernel code contributions.

You are free to:

- Share — copy and redistribute the material in any medium or format
- Adapt — remix, transform, and build upon the material for any purpose, even commercially.

The licensor cannot revoke these freedoms as long as you follow the license terms.

Under the following terms:

Attribution — You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.

No additional restrictions — You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.