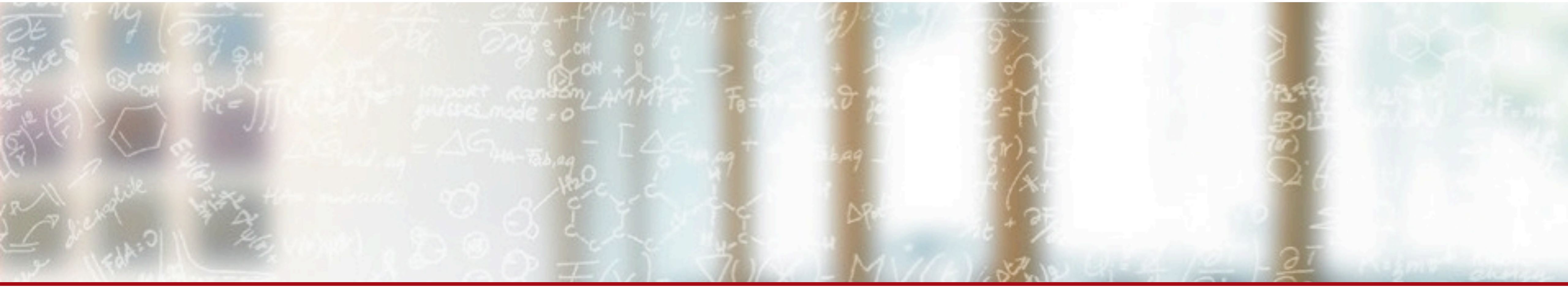
CSCS
Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

ETH zürich



ReFrame: A Regression Testing Tool for HPC Systems

Regression testing BoF, SC17

G. Peretti-Pezzi, V. Karakasis, CSCS

Nov. 14, 2017

Regression Testing of HPC Systems

Why is it so important?

- Ensures quality of service
- Reduces downtime
- Early detection of problems

Regression Testing of HPC Systems

But it’s a painful story

- In-house custom solutions per center
- Non portable monolithic regression tests
  - Tightly coupled to the system configuration and programming environments
- Large maintenance overhead
  - Replicated code of the system interaction details
  - Test’s logic is lost in unrelated lower level details

Reluctance to implement new regression tests!

What Is ReFrame?

A new regression framework that

- allows writing portable HPC regression tests in Python,
- abstracts away the system interaction details,
- lets users focus solely on the logic of their test.

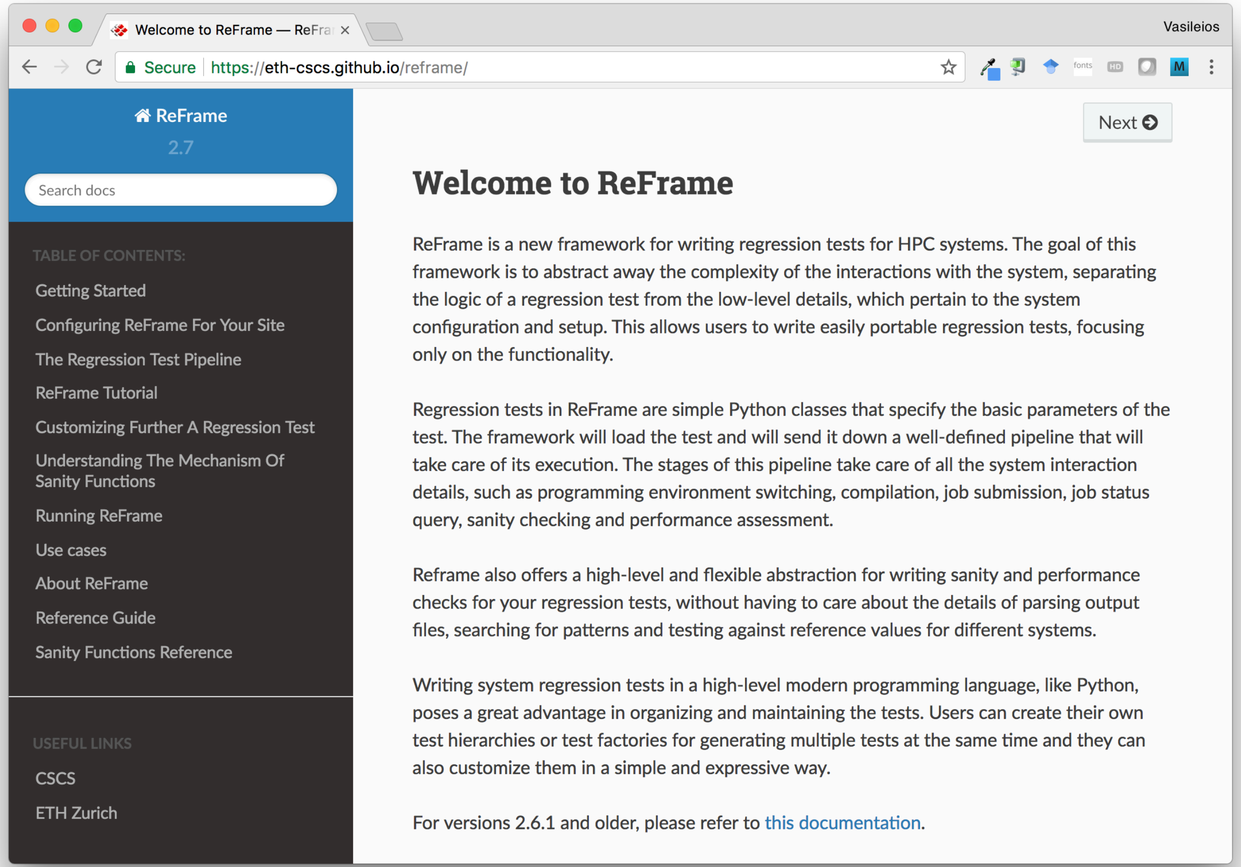

Welcome to ReFrame — ReFra
Vasileios
Secure https://eth-cscs.github.io/reframe/
ReFrame
2.7
Search docs
Next

TABLE OF CONTENTS:

- Getting Started
- Configuring ReFrame For Your Site
- The Regression Test Pipeline
- ReFrame Tutorial
- Customizing Further A Regression Test
- Understanding The Mechanism Of Sanity Functions
- Running ReFrame
- Use cases
- About ReFrame
- Reference Guide
- Sanity Functions Reference

USEFUL LINKS

- CSCS
- ETH Zurich

Welcome to ReFrame

ReFrame is a new framework for writing regression tests for HPC systems. The goal of this framework is to abstract away the complexity of the interactions with the system, separating the logic of a regression test from the low-level details, which pertain to the system configuration and setup. This allows users to write easily portable regression tests, focusing only on the functionality.

Regression tests in ReFrame are simple Python classes that specify the basic parameters of the test. The framework will load the test and will send it down a well-defined pipeline that will take care of its execution. The stages of this pipeline take care of all the system interaction details, such as programming environment switching, compilation, job submission, job status query, sanity checking and performance assessment.

Reframe also offers a high-level and flexible abstraction for writing sanity and performance checks for your regression tests, without having to care about the details of parsing output files, searching for patterns and testing against reference values for different systems.

Writing system regression tests in a high-level modern programming language, like Python, poses a great advantage in organizing and maintaining the tests. Users can create their own test hierarchies or test factories for generating multiple tests at the same time and they can also customize them in a simple and expressive way.

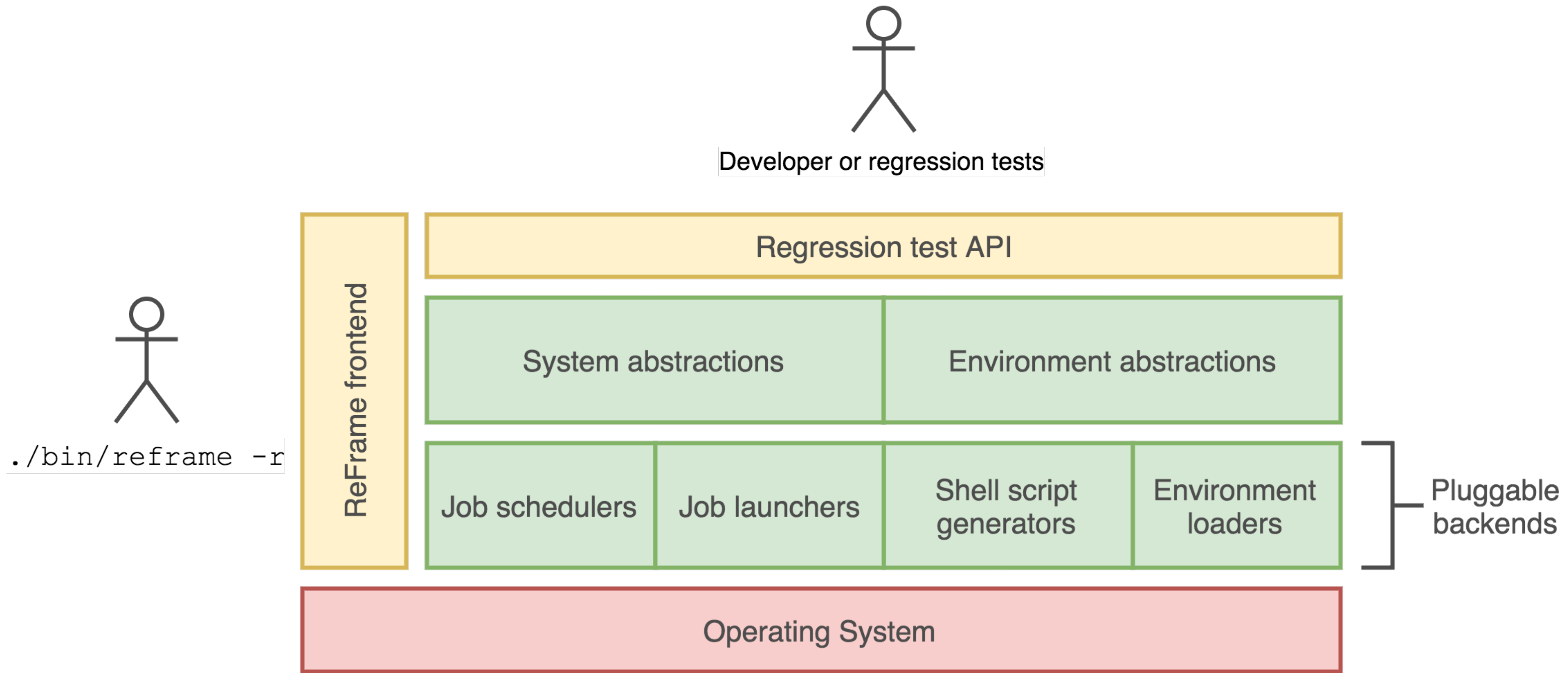
For versions 2.6.1 and older, please refer to this documentation.

https://github.com/eth-cscs/reframe

Design Goals

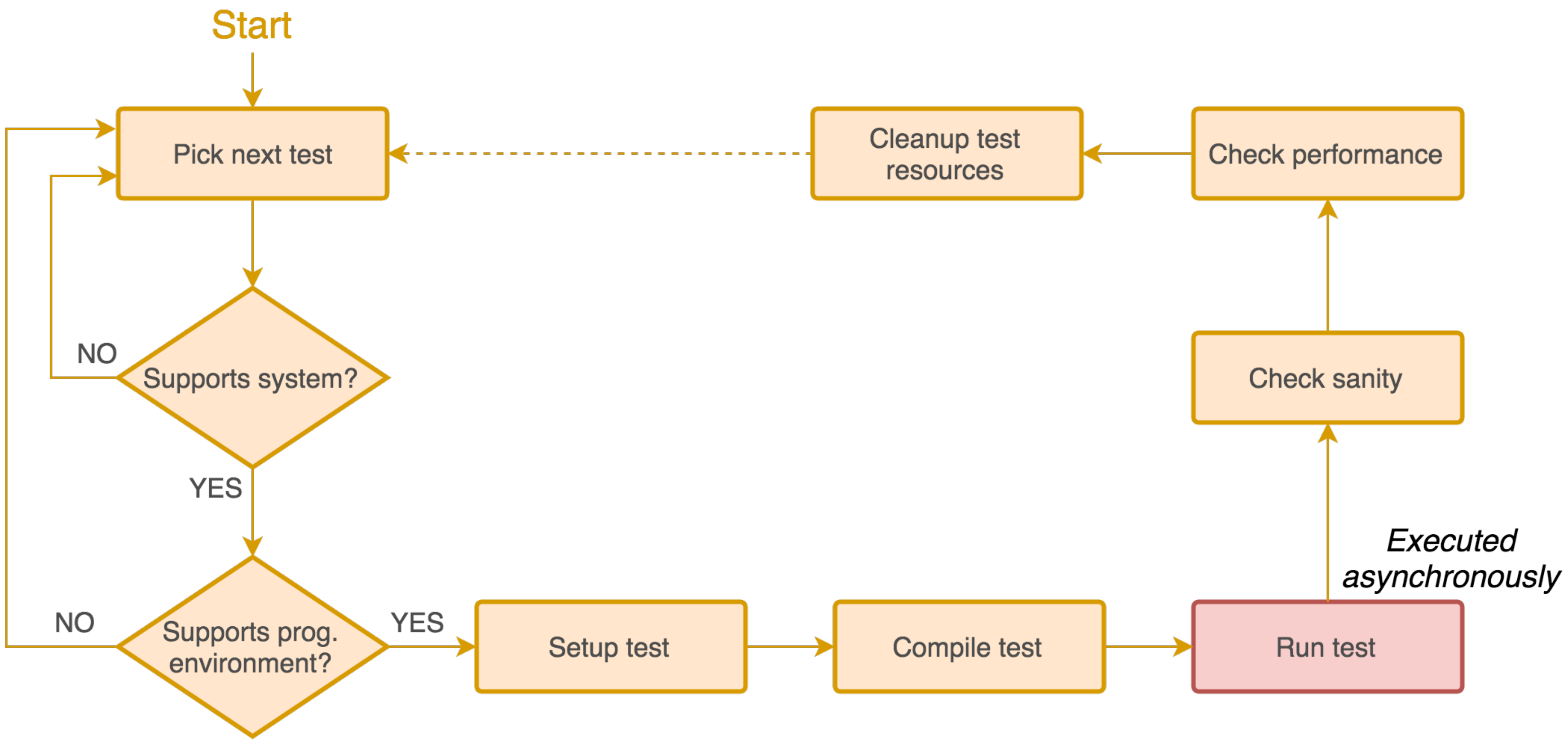
- Productivity
- Portability
- Speed and Ease of Use
- Robustness

ReFrame's architecture



The Regression Test Pipeline

A series of well defined phases that each regression test goes through



Key Features

- Support for Slurm
- Support for sanity and performance tests
- Flexible organization of the regression tests
- Rich command-line interface
- Progress and results reports
- Asynchronous execution of regression tests
- Complete documentation

Demo Time

1. Running ReFrame

2. How a regression test written in ReFrame looks like?

Go to demo

The CSCS Use Case

- ReFrame is used to test continuously all major systems in production
- Wide variety of performance and sanity tests implemented
  - Applications
  - Libraries
  - Programming environment tests
  - I/O benchmarks
  - Tools and debuggers
  - Job submission tests
- Two execution modes
  - **Production**: A wide aspect of the sanity and performance tests running daily overnight
  - **Maintenance**: Key functionality and performance tests run before and after maintenances

The CSCS Use Case

Comparison to our former shell script based solution

Maintenance Burden	Shell-script based suite	ReFrame
Total size of tests	**14635 loc**	**2985 loc**
Average test file size	179 loc	93 loc
Average effective test size	179 loc	25 loc

5x reduction in the amount of code of regression tests

Summary

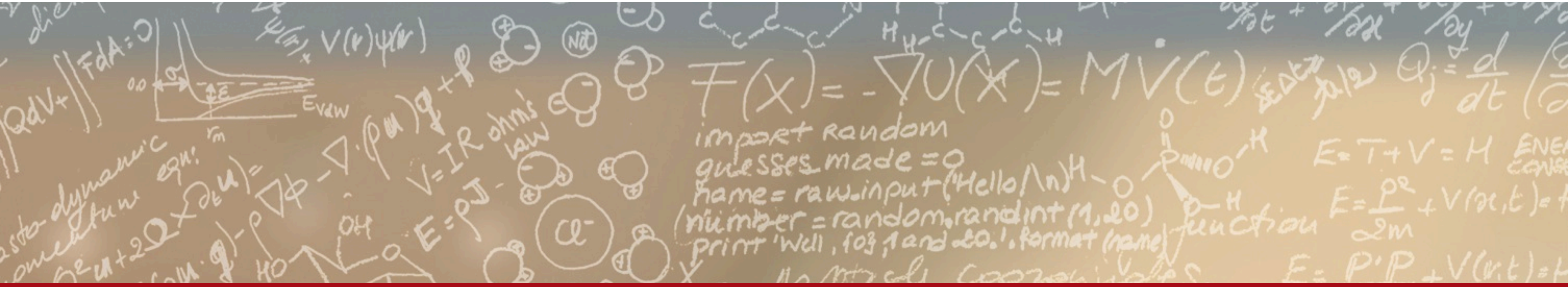
ReFrame makes writing regression tests for HPC systems an easy task!

- Actively developed, monthly release cycle
- The standard way of testing systems at CSCS
- Publicly available at https://github.com/eth-cscs/reframe

Try it out and give us some feedback!

CSCS
Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

ETH zürich



Thank you for your attention.