



---

# Challenges of building a test infrastructure around a cross-platform Desktop IDE

Attila Simon

Software QA Engineering Manager



# Contents

- **Silicon Labs**
- **Simplicity Studio**
  - Key features overview
- **The evolution of testing Simplicity Studio**
  - Manual testing
  - Test automation
  - Continuous Integration
  - Test report generation
  - Test framework creation
  - Testing on Silicon Labs HW
  - Parallel test execution
  - Latency-sensitive testing
  - Multiple OS support
  - Dynamic test case execution
- **Q&A**

# The Leader in Short Range IoT Wireless Connectivity



60%

Revenue Based on IoT



Breadth and Depth of Wireless IoT Protocols



#1

Share in Mesh



1st

to Market with Multi-protocol, BLE Mesh, BLE 5.1



Innovation

Performance, Power, CoEx, Xpress, Modules

**ember**

2012

Software ZigBee SoC

**ENERGY**  
micro

2013

Low-power 32-bit MCUs

**blue giga**

2015

BT Smart Modules

telegesis

2015

ZigBee/Thread Modules

**Micrium**

2016

Software RTOS

**ZENTRI**

2017

Cloud Connected Wi-Fi

**ZWAVE**

2018

Smart Home Protocol

**REDPINE SIGNALS**

2020

Ultra Low Power Wi-Fi

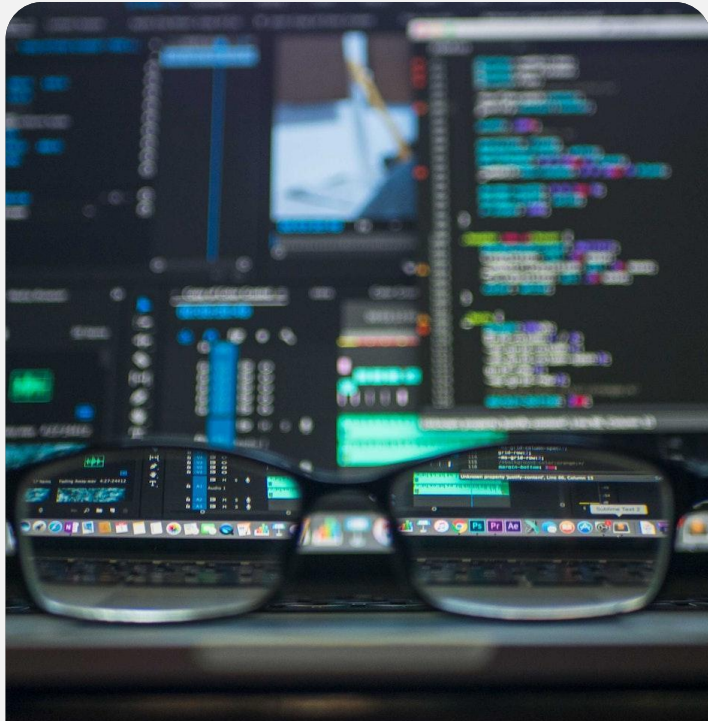


---

# Simplicity Studio

Simplicity Studio is the core development environment designed to support the Silicon Labs IoT portfolio of system-on-chips (SoCs) and modules.

# Key features overview



## POWERFUL IDE

- Fresh, clean user interface
- Built on latest Eclipse Framework
- SDK download and update manager



## SILICON LABS HW SUPPORT

- Automatic detection of connected development boards
- Context-aware developer resources
- Easy programming of pre-built demo apps



## ANALYSIS AND CONFIGURATION TOOLS

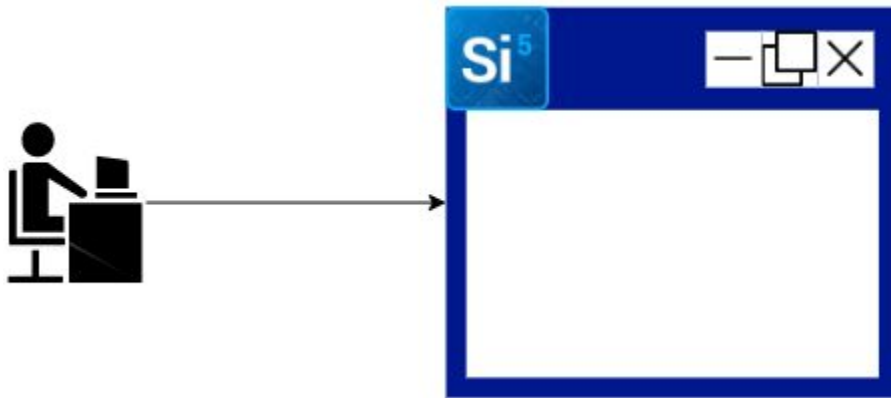
- Wireless network traffic capture and analysis
- Capture and display energy usage
- Pin tool to assign pin and peripheral hardware resources
- Proprietary radio configurator tool

---

# The evolution of testing Simplicity Studio



# Manual testing



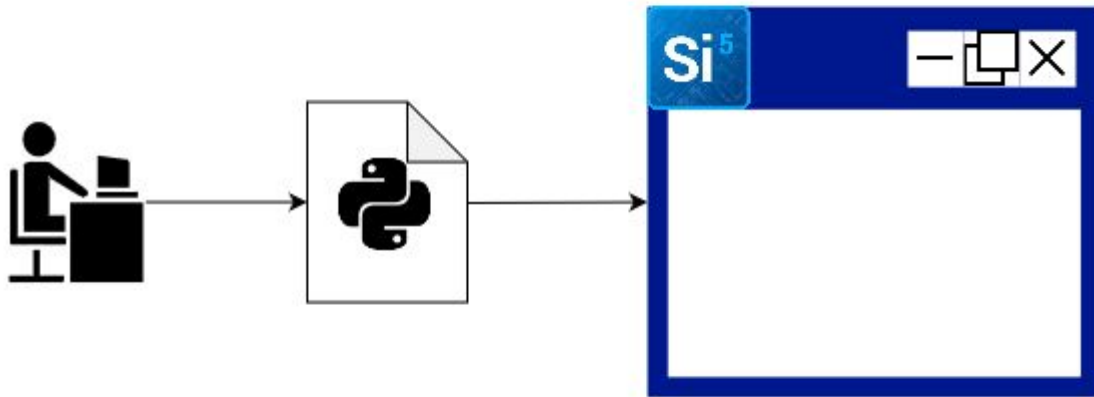
## Initial approach

- Manual testing of new features

## Challenge

- On average, 50+ new features are developed in each release
- Multiple versions are released each year

# Test automation



## Solution

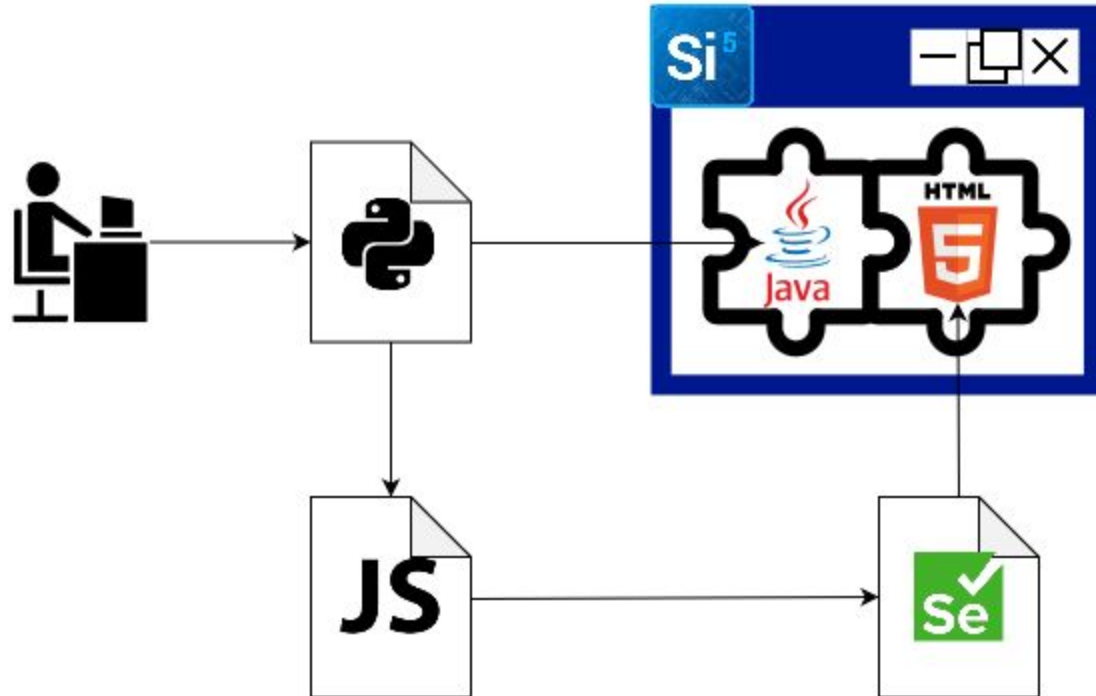
- Test automation with Python

## Challenge

- Simplicity Studio uses Eclipse's native SWT and JXBrowser components



## Test automation #2



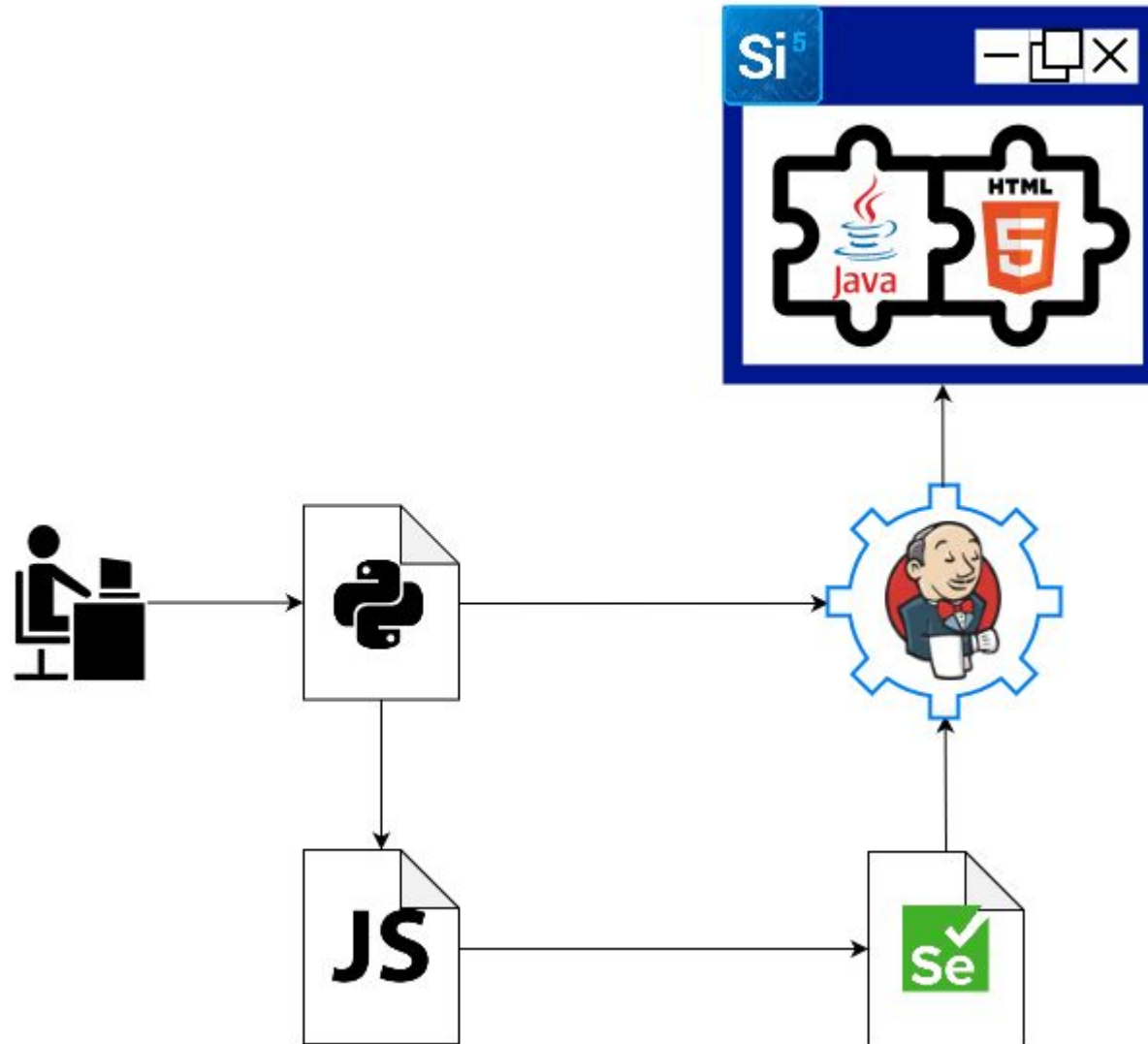
### Solution

- Test JXBrowser web GUI with JS + Selenium

### Challenge

- Development code changes daily

# Continuous integration



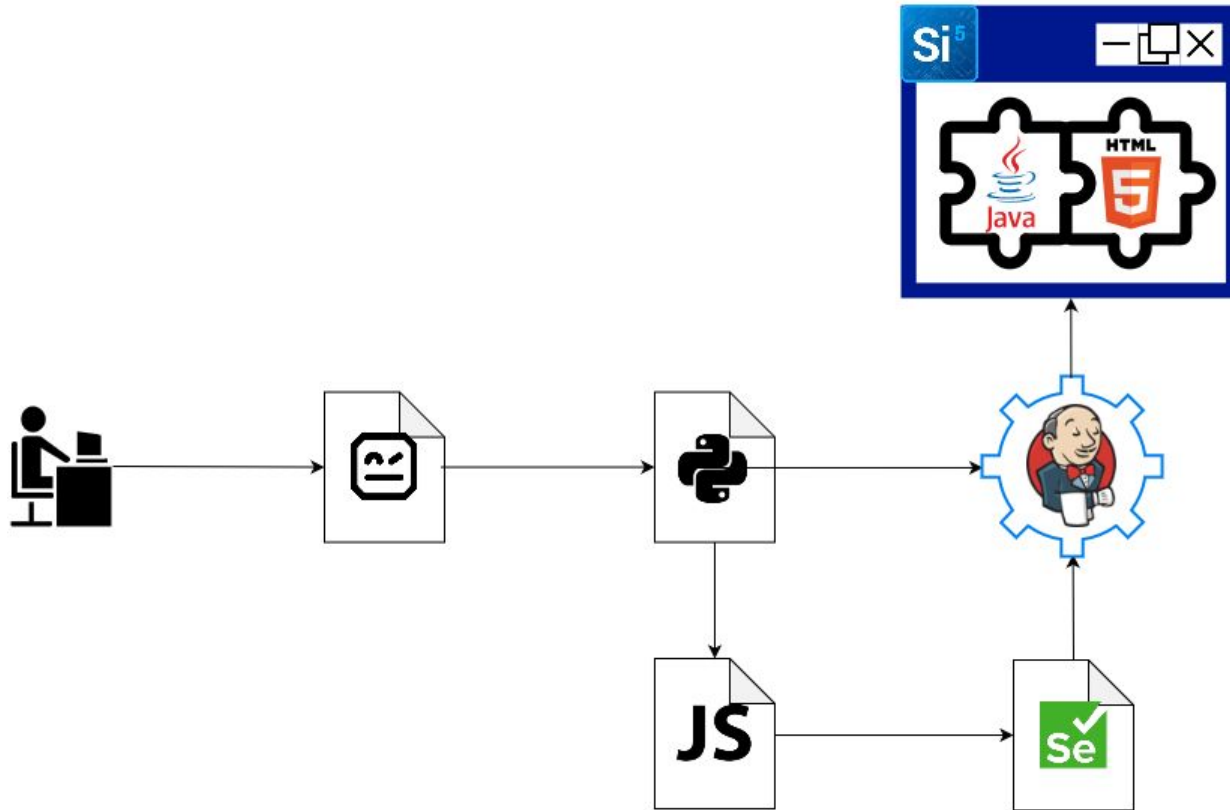
## Solution

- Use CI to automate regression testing

## Challenge

- Test result is hard to read, bugs need further investigation to be found

# Test report generation



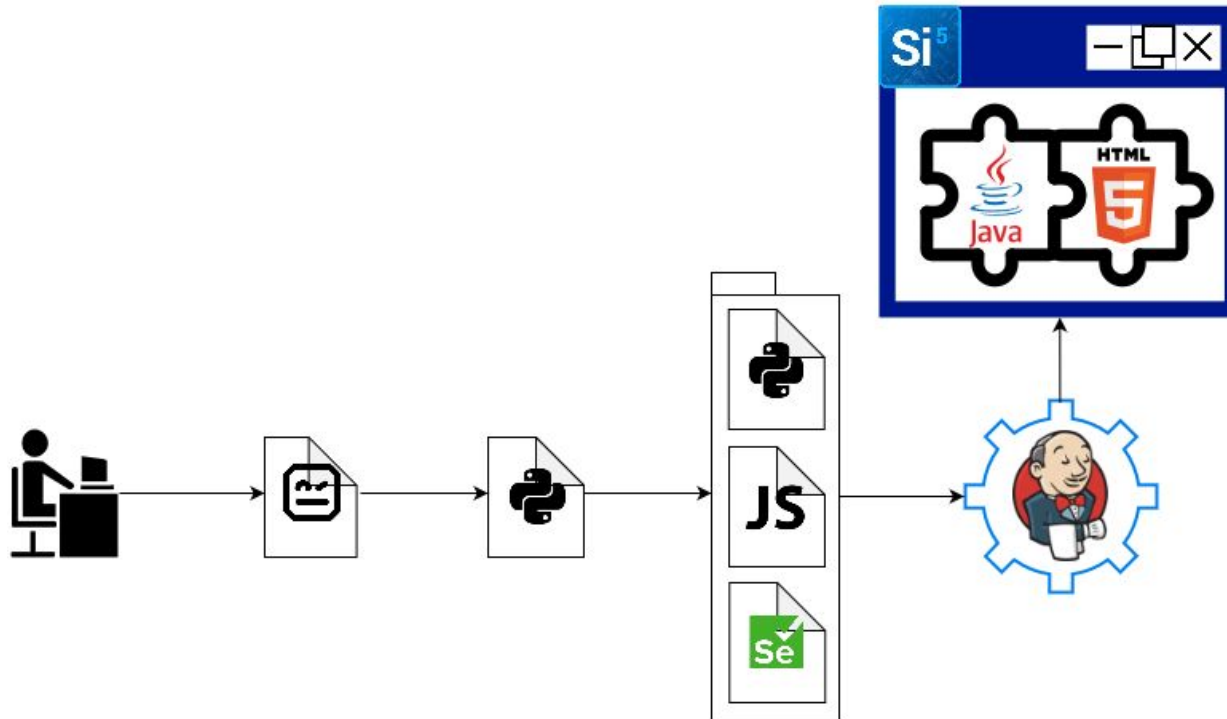
## Solution

- Use Robot framework to generate readable test report

## Challenge

- UI elements are similar, using the same components

# Test framework creation



## Solution

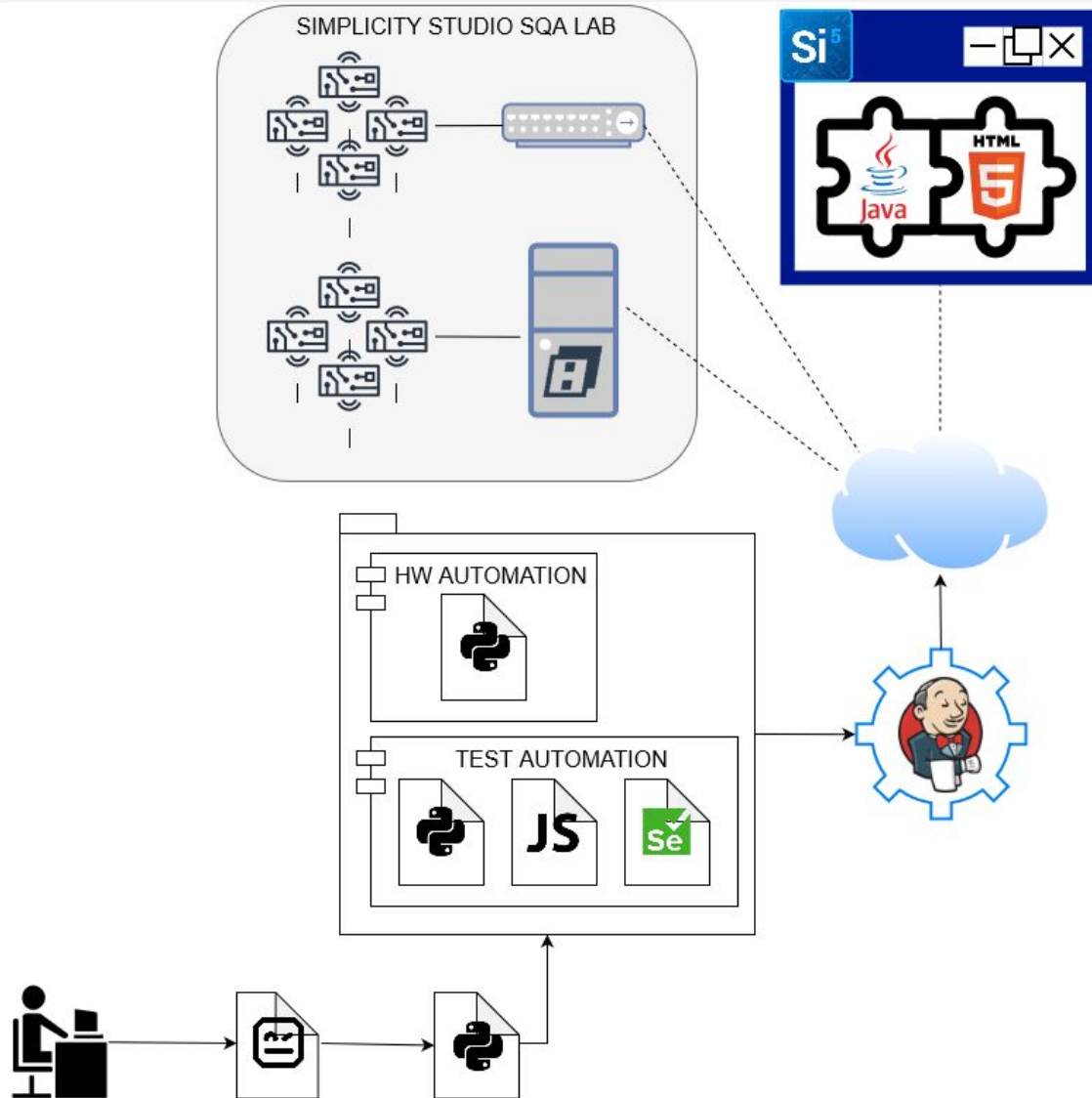
- Create a test framework tailored to Simplicity Studio GUI elements

## Challenge

- Simplicity Studio shows context-aware views when Silicon Labs HW is attached



# Testing on Silicon Labs HW



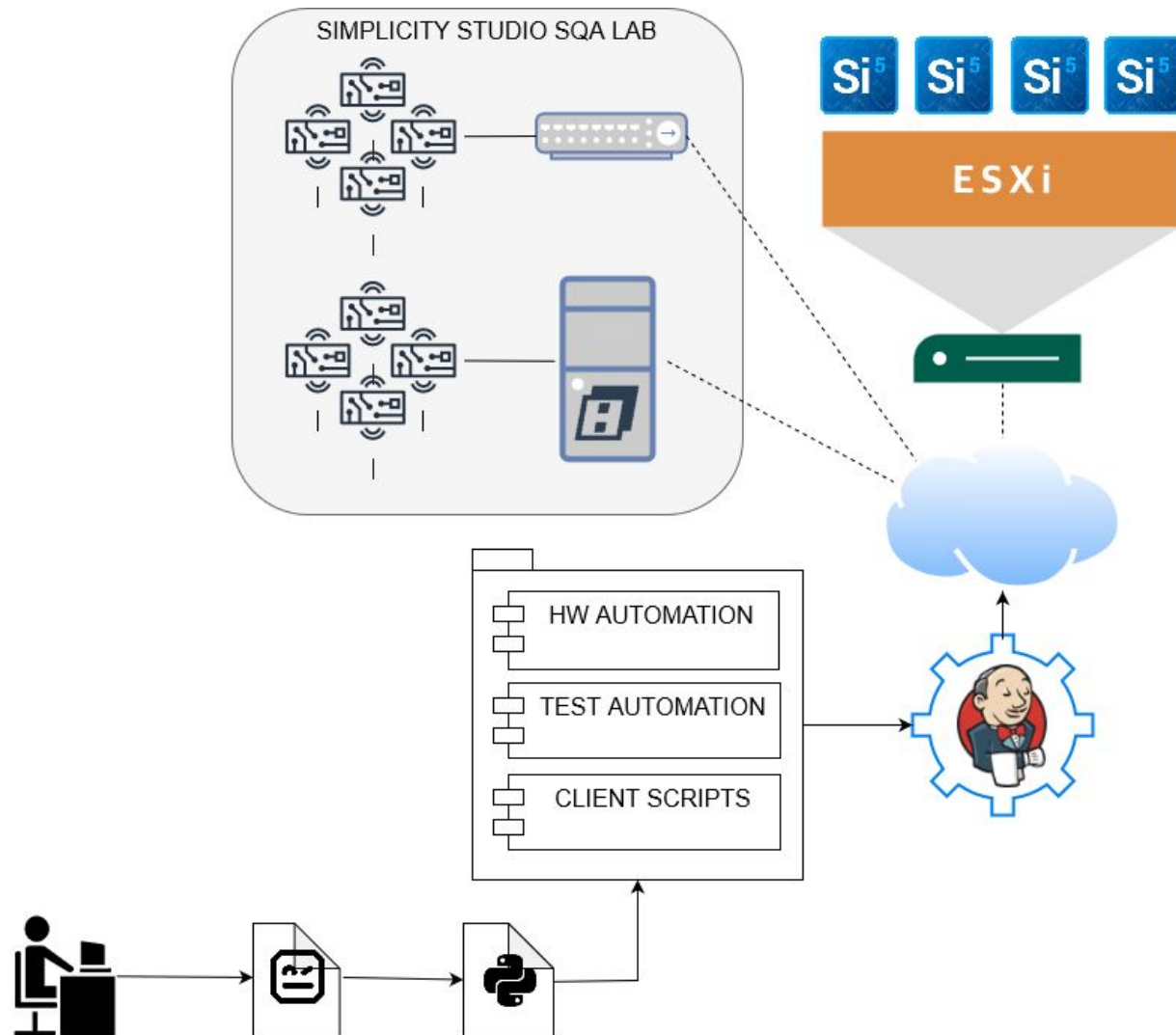
## Solution

- VirtualHere USB server to host Silicon Labs HW
- Connect HW directly to the network
- Add HW automation module to the test framework

## Challenge

- All possible configurations of the regression test set runs too long

# Parallel test execution



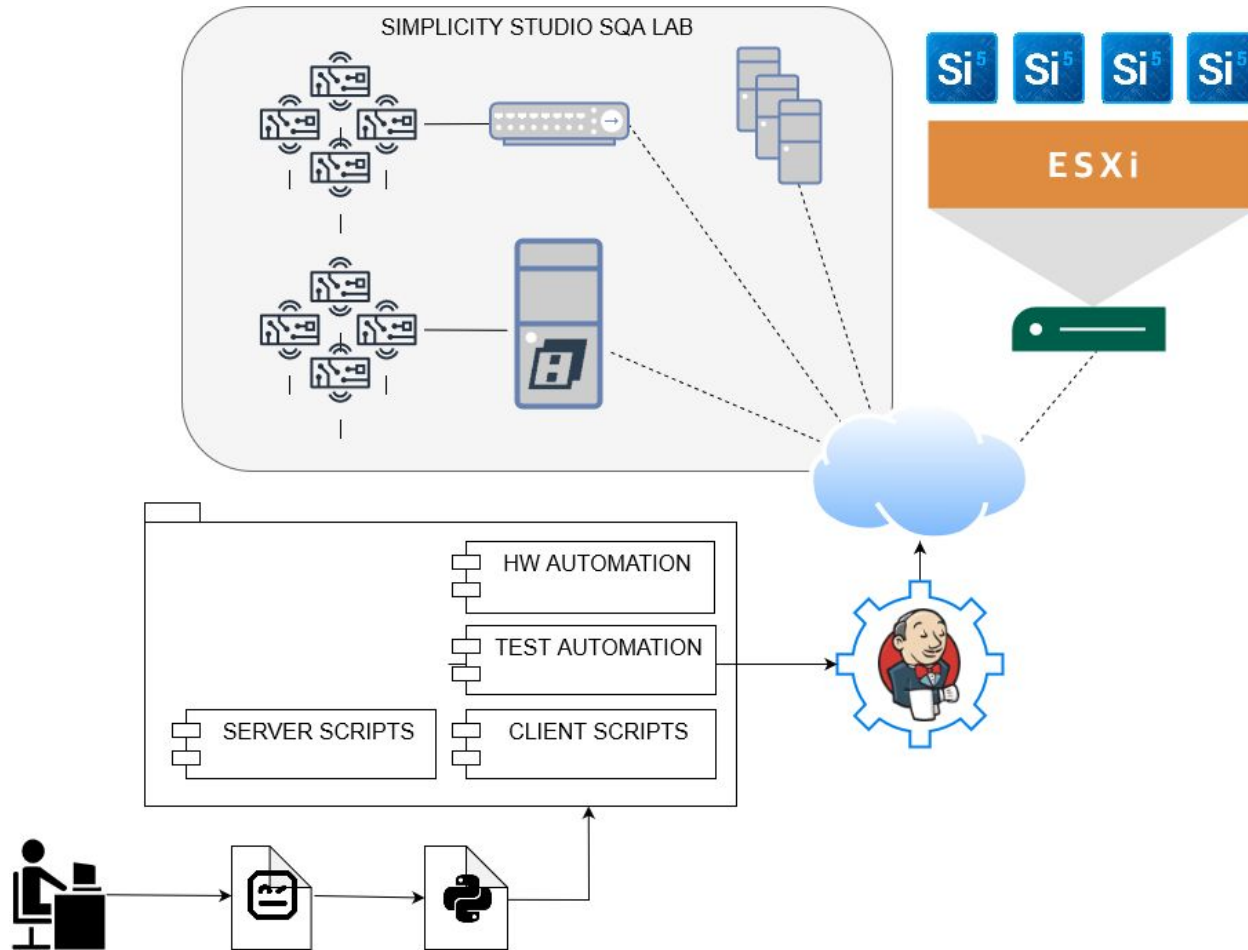
## Solution

- Run the tests on VMs to parallelize and therefore speed up test execution

## Challenge

- Latency becomes a problem for tests using Silicon Labs HW

# Latency-sensitive testing



## Solution

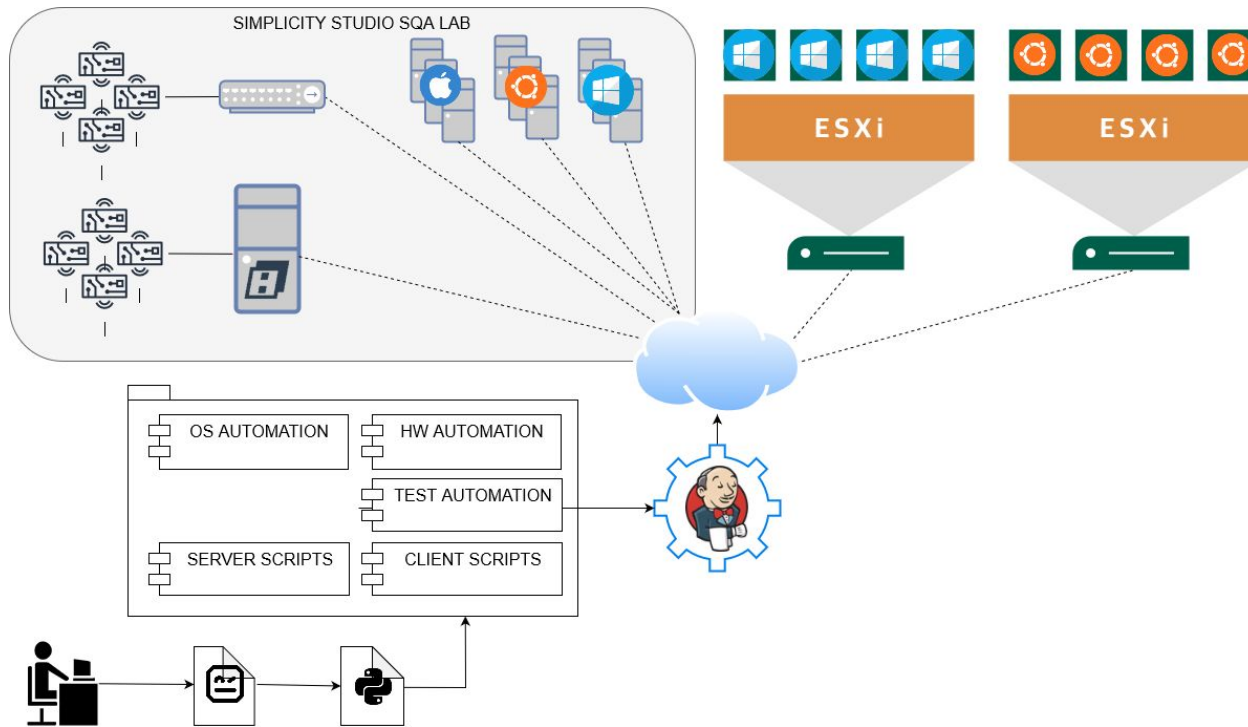
- Add low-latency servers to the SQA lab to run latency-sensitive tests on them

**This is our current solution**

## Challenge

- Simplicity Studio supports all major OS versions

# Multiple OS support



## Plan

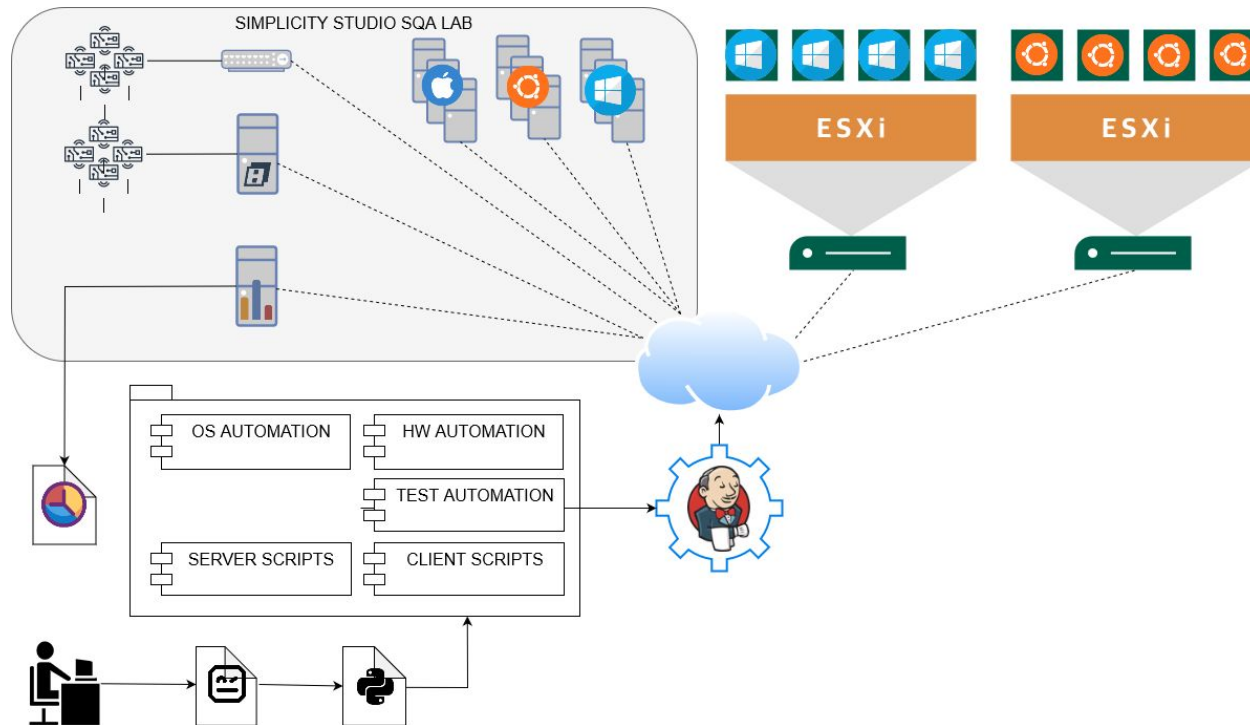
- Add low-latency servers with Linux and MacOS to the SQA lab to run latency-sensitive tests
- Add Linux VMs to run regression tests

## Next challenge

- All possible configurations of the regression test set runs too long (again!)



# Dynamic test case execution



## Plan

- Optimize test execution part to dynamically allocate test cases based on the number of available VMs
- Separate install and report generation parts of the infrastructure

## Next challenge

- ?



**Q&A**

**Thank you for your attention!**

**ATTILA SIMON**

SOFTWARE QA ENGINEERING MANAGER



[HTTPS://WWW.LINKEDIN.COM/IN/ATTILA-SIMON/](https://www.linkedin.com/in/attila-simon/)

