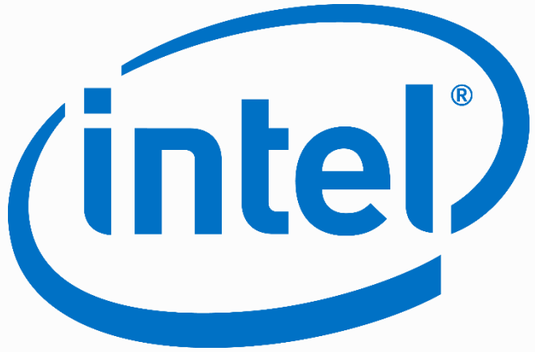
presented by



Implementing MicroPython as a UEFI Test Framework

Spring 2018 UEFI Seminar and Plugfest
March 26-30, 2018
Presented by Chris McFarland (Intel)

Agenda

- Introduction
- MicroPython
- Implementation in EDK II
- Test Framework
- Future Possibilities
- Summary & Call to Action

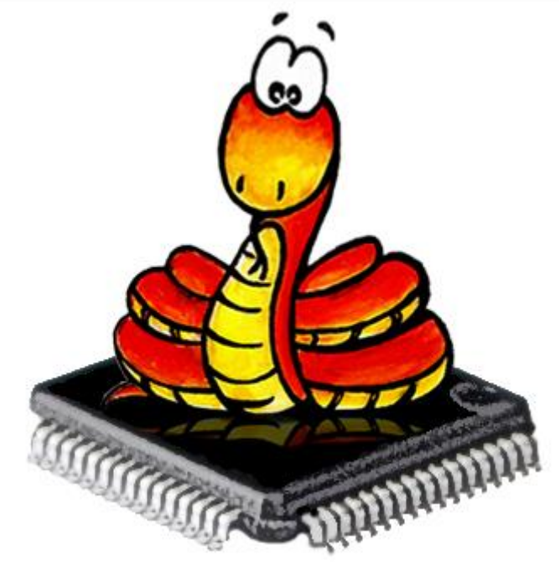
Implementing MicroPython in UEFI

Introduction

Introduction

MicroPython is a Python* 3 variant with memory & size optimizations for microcontrollers

This session describes implementation details, metrics, and future plans for a test framework based on the MicroPython engine for UEFI



Implementing MicroPython in UEFI

MicroPython

MicroPython

- Python has become a popular high-level language in academia and Industry
- Cpython* has been ported to UEFI for years but not widely used. Why?
  - Poor performance & large footprint
  - No direct access to EDK II resources
  - No direct access to hardware resources
  - Limited usage with UEFI Shell dependencies

MicroPython Features

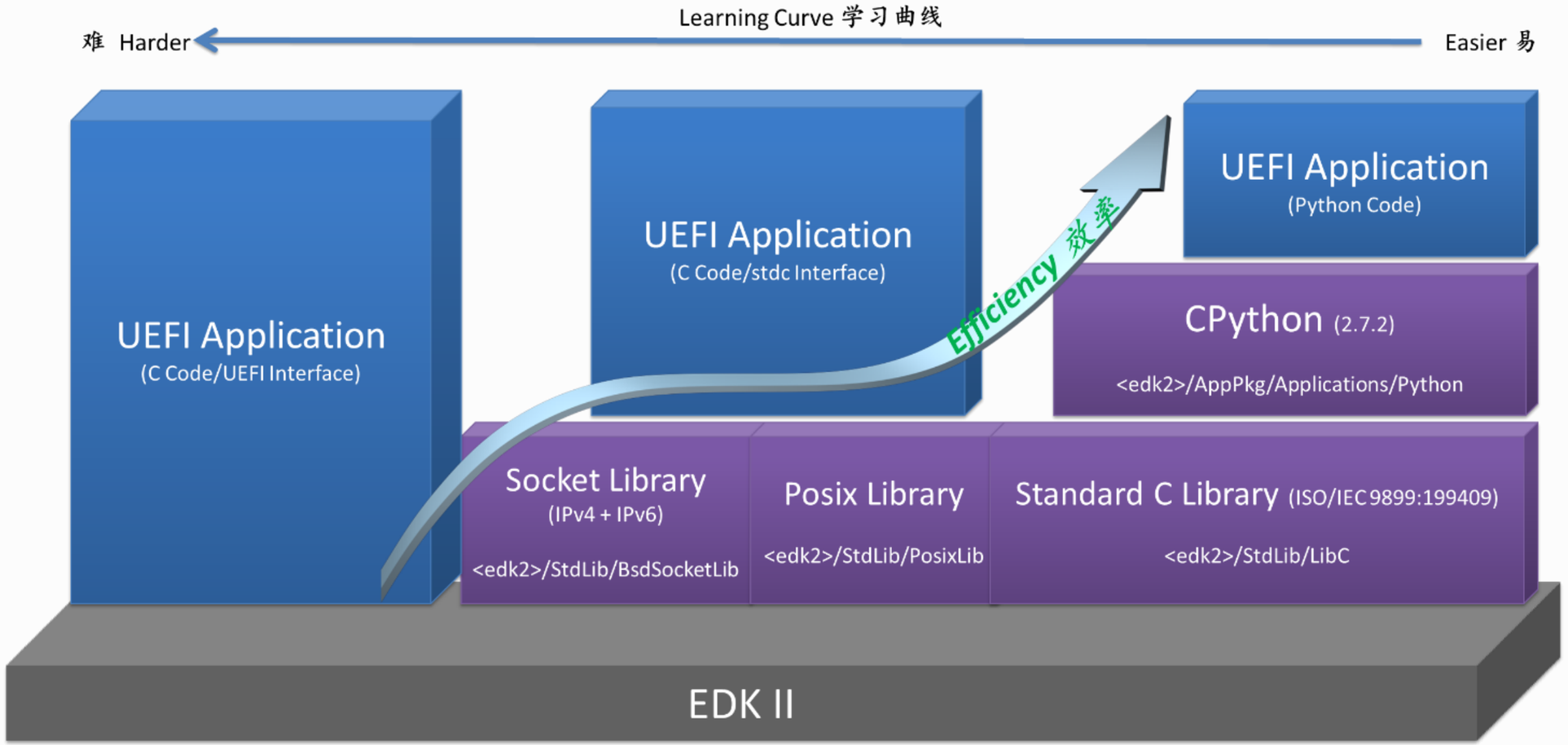
- MicroPython is a lean and efficient implementation of the Python 3 interpreter – micropython.org
- Project was created by Prof. Damien P. George (University of Cambridge)
- Aimed at “constrained environments” (microcontrollers, embedded devices, …)
- Open source project hosted on GitHub (over 4900 ★ and more than 1100 forks)
- Completely free with MIT license

Implementing MicroPython in UEFI

Implementation in EDK II

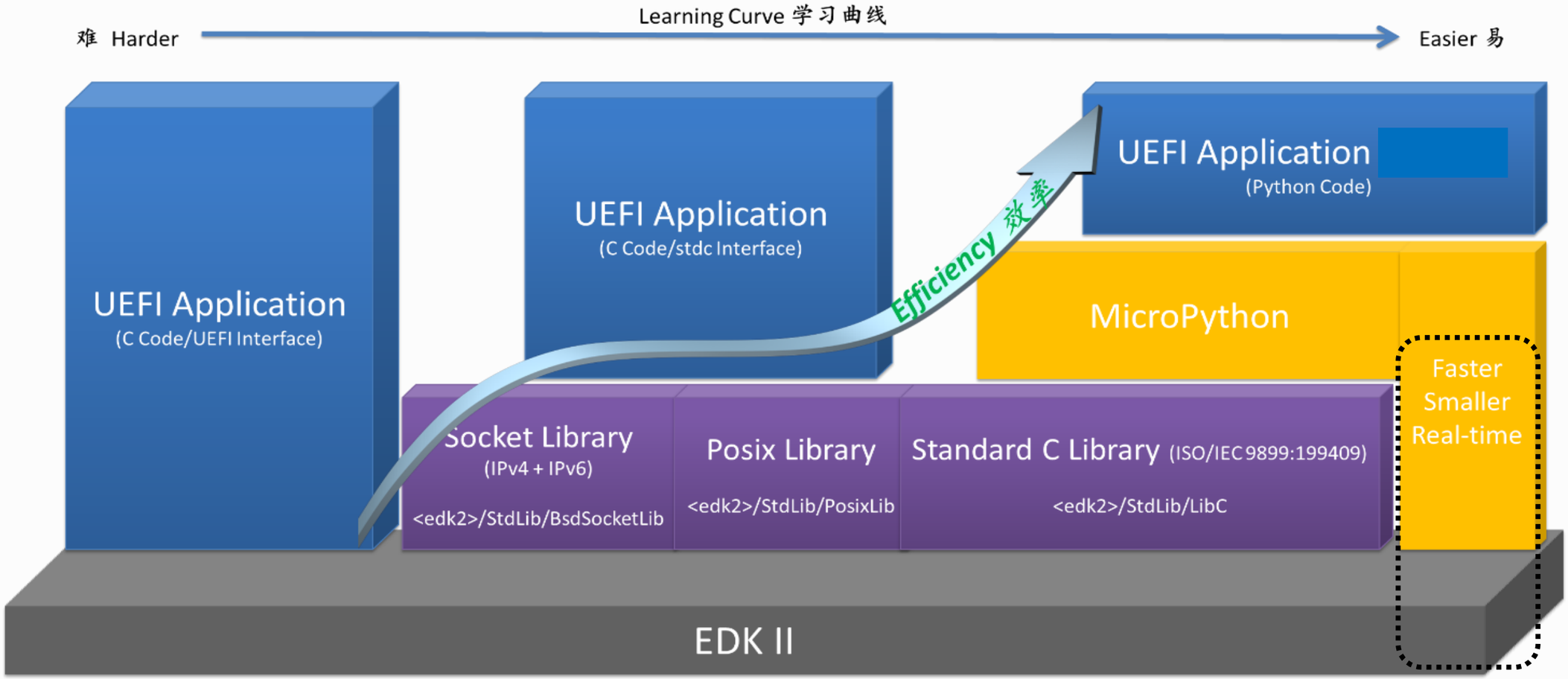
Implementation in EDK II

Current: CPython 2.7



Implementation in EDK II

Proposed: MicroPython (Python 3.x)



Features of MicroPython on UEFI

- Provides scripts with access to resources:
  - UEFI & EDK II interfaces
  - Hardware-level access
  - Interpreter and Native Interface
- Support for native/inline assembly code
- Note: There are small differences between MicroPython and Python 3 language behavior (documented on website)

MicroPython Usage for UEFI

- Replacing DOS/Legacy Applications
- Diagnostics & Manufacturing Testing
- Rapid Development of UEFI Applications
- Support for UEFI CPython scripts after conversion to 3.x
  (Python 2.7 expected to EOL in 2020)

Implementing MicroPython in UEFI

Test Framework

Test Framework

- Why create a MicroPython test framework?
  - Existing test tools & scripts are built for specific use cases, which are not covered well by other tools
  - Few unified EDK II testing frameworks exist for both development and QA engineers
  - Few standard available automation test tools or framework for EDK II community, esp. for new feature development and validation

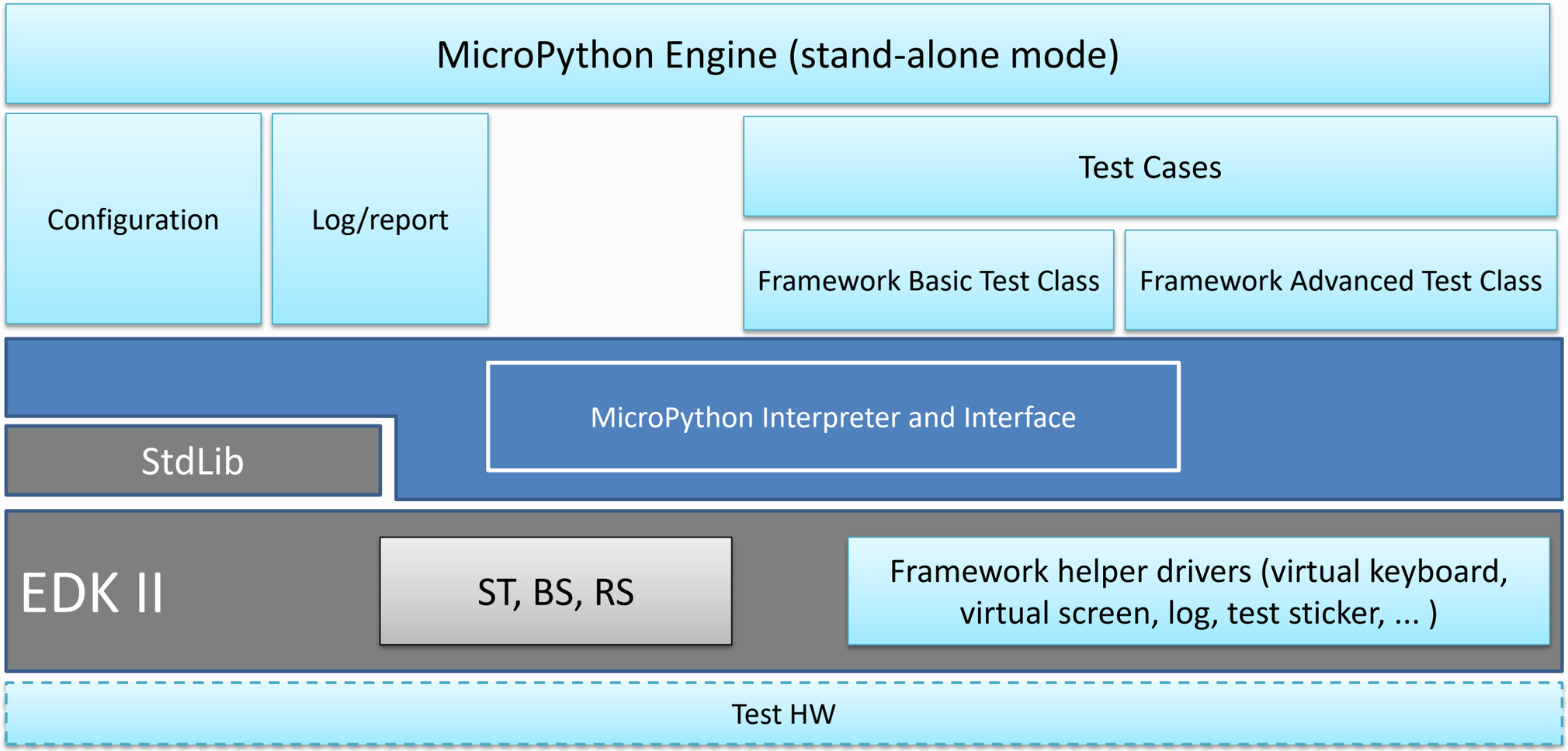
Test Framework Goals

- Improve Unit Test capabilities
  - EDK II developers heavily rely on code review and simple unit tests during new feature development
- Reduce knowledge requirements of EDK II & C as a pre-requisite for developing basic UEFI tests
  - Allow QA to more easily develop and maintain their own tools and test cases
  - Allow developers and QA engineers to focus primarily on authoring actual test logic

Test Framework Properties

- Scripts and libraries for MicroPython on UEFI
- Provide a set of convenient UEFI abstractions
- Designed to simplify firmware unit testing, development, automation, and test execution
- Lightweight, minimalistic, and extensible
- General enough to be useful in a variety of test scenarios (new feature development, unit test, manufacturing flow)
- Target Use Cases: black box tests, white box tests, functional tests, automated UI/human interaction

Test Framework Stack



Implementing MicroPython in UEFI

Future Tasks

Future Tasks

- Create Branches for MicroPython Engine and Framework
  - Engine will continue with MT License
  - Framework will use EDKII Default license
- Optimize and extend MicroPython Engine
  - Need to increase the number of default libraries included

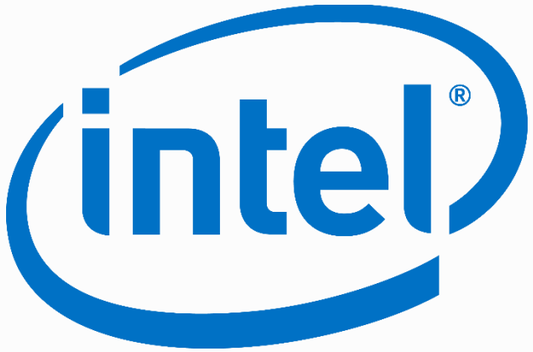
Future Tasks

- Continue Development Of Test Framework
  - Python script, Python basic lib, and asynchronous execution under DXE phase
  - Virtual Keyboard input simulator, and screen recognition
  - Python API wrapper for UEFI variable, PCD, and the other protocols etc
  - Full APIs for the common operations, like Verify, WaitUntil, Input, Capture etc.
  - Basic test case libraries, reference test cases and user manuals.
- Expect first release early Q3 2018

Thanks for attending the Spring 2018 UEFI Plugfest

For more information on the UEFI Forum and UEFI Specifications, visit http://www.uefi.org

presented by



Intel Legal Notice