ROBOTmk

E2E-Monitoring mit Robot Framework und Checkmk

Philipp Lemke
Simon Meggle

SLAC 2023
23.–25. Mai 2023 | Berlin

Agenda

- Introduction: who/what/why…
- Robot Framework: Basics
  - Installation eines Testclients
  - Architektur
  - Syntax
- Praxisbeispiel: wir implementieren einen E2E-Test für einen Webshop
- Checkmk:
  - Vorbereitung & Konfiguration
  - Discovery & Check-Konfiguration

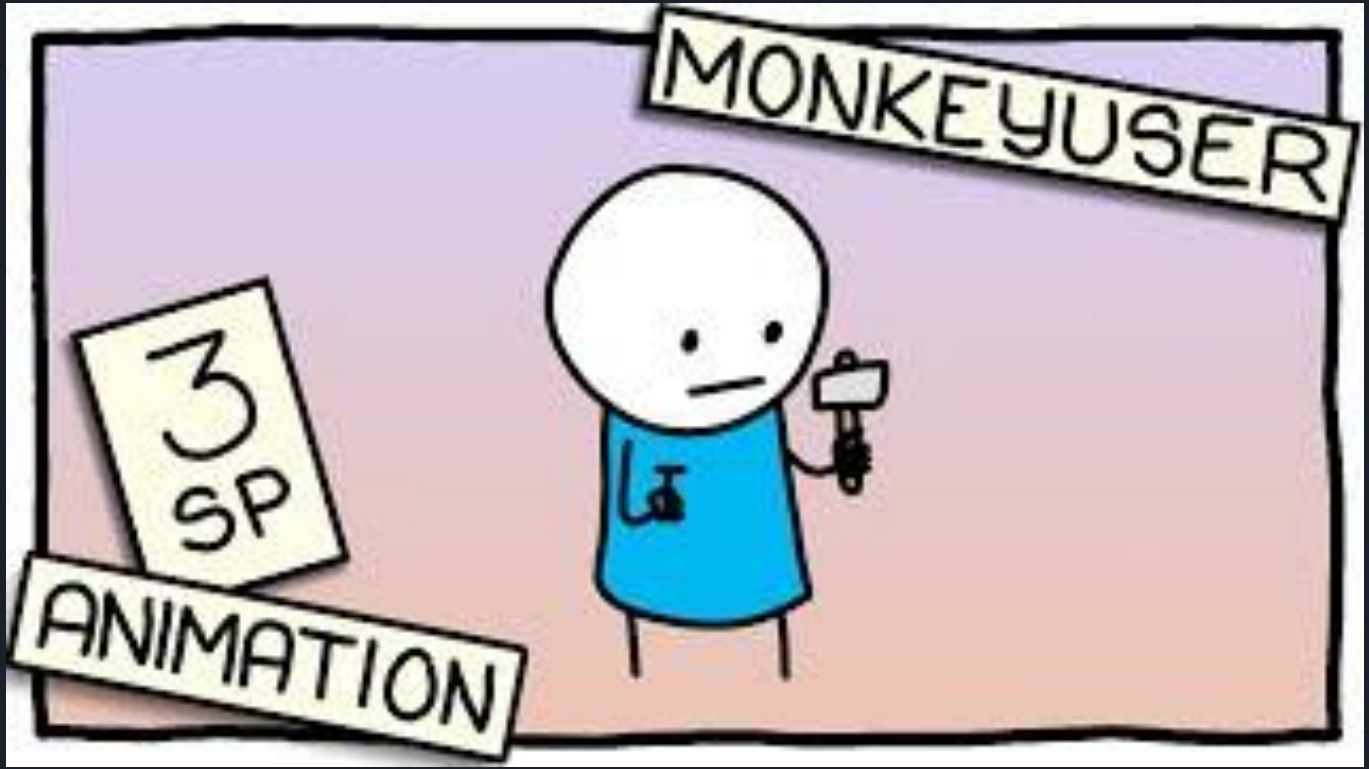
Die Referenten

Philipp Lemke

PL Automation Monitoring GmbH

Simon Meggle

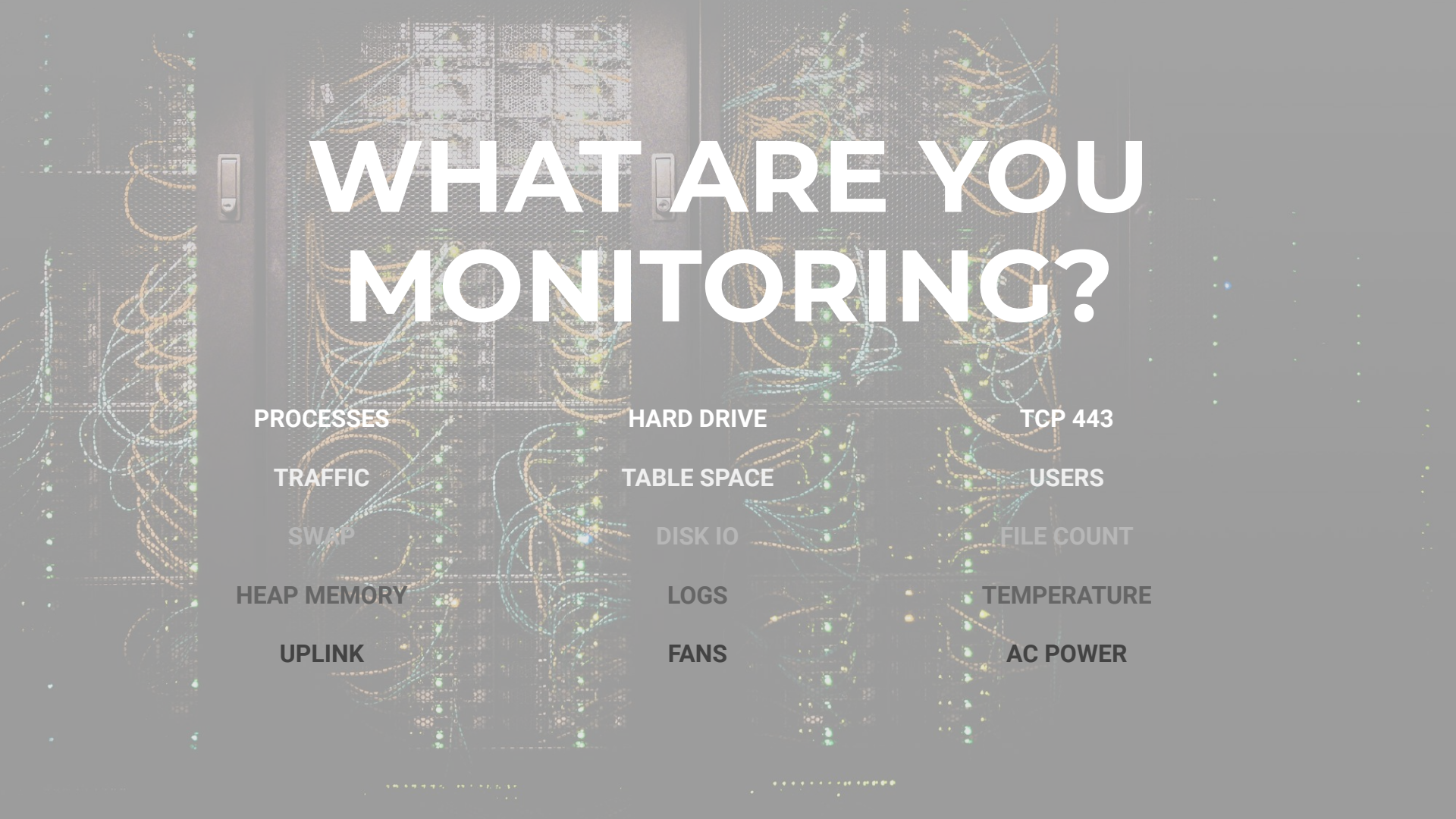
ELABIT GmbH
Checkmk GmbH


MONKEYUSER
3
SP
ANIMATION

Zum Workshop

- Classroom-Style
- Ein Wort zur “Flughöhe”...
- Fragen: jederzeit
- Folien & Source
- Pause ca. 10.15 Uhr

1. Intro



WHAT ARE YOU MONITORING?

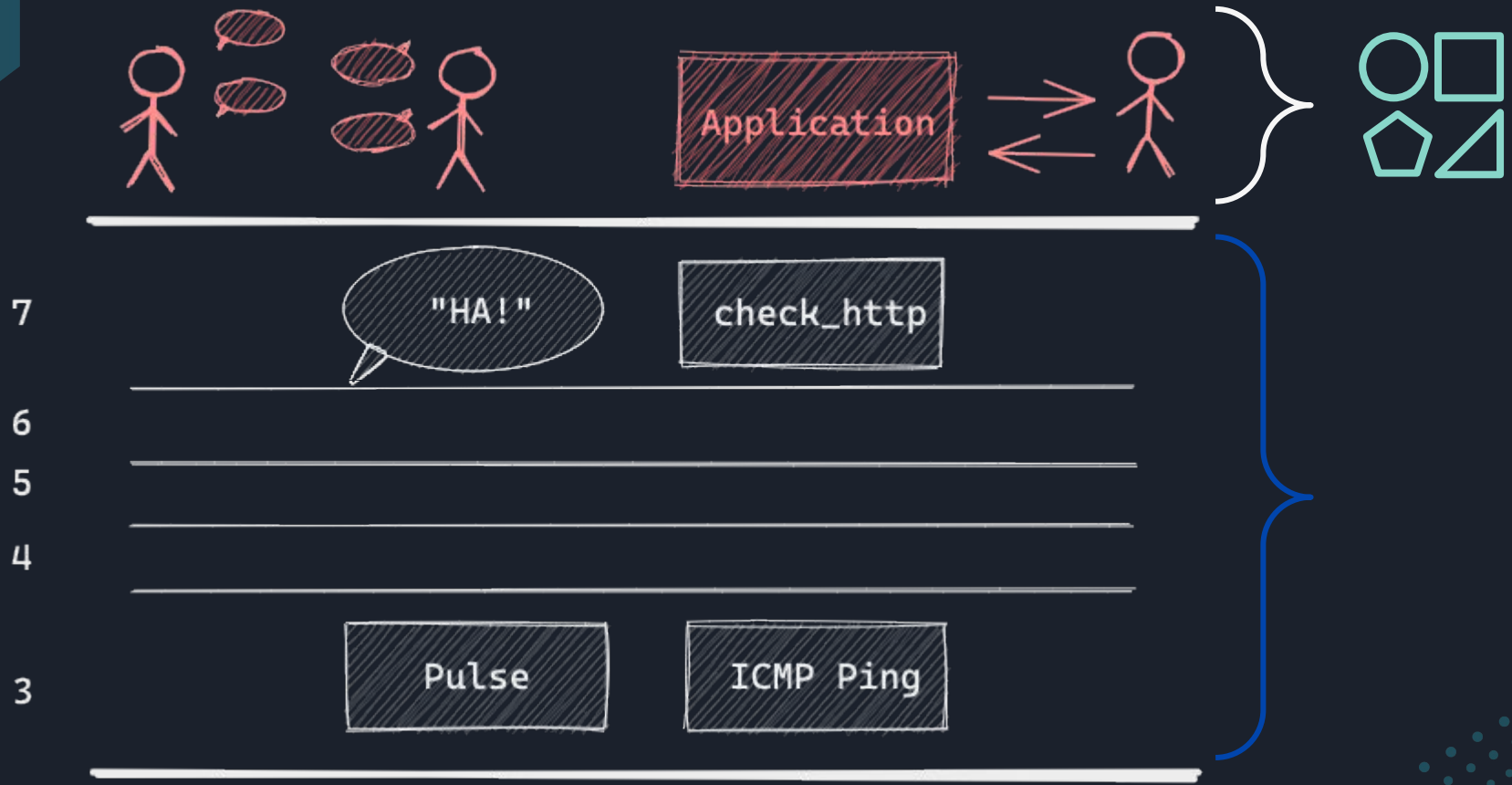
PROCESSES	HARD DRIVE	TCP 443
TRAFFIC	TABLE SPACE	USERS
SWAP	DISK IO	FILE COUNT
HEAP MEMORY	LOGS	TEMPERATURE
UPLINK	FANS	AC POWER

AVAILABLE.

PERFORMANT.

FUNCTIONING.


WHAT ARE YOU
NOT
MONITORING?


Application
7
"HA!"
check_http
6
5
4
3
Pulse
ICMP Ping

```python
from selenium import webdriver

driver = webdriver.Chrome(r'C:\Users\drivers\chromedriver.exe')
driver.maximize_window()
driver.get("http://www.seleniumeasy.com/test/basic-first-form-demo.html")
assert "Selenium Easy Demo - Simple Form to Automate using Selenium" in driver.title

eleUserMessage = driver.find_element_by_id("user-message")
eleUserMessage.clear()
eleUserMessage.send_keys("Test Python")

eleShowMsgBtn=driver.find_element_by_css_selector('#get-input > .btn')
eleShowMsgBtn.click()

eleYourMsg=driver.find_element_by_id("display")
assert "Test Python" in eleYourMsg.text
driver.close()
```

- Generic test automation framework
- written in Python
- huge community
- key features:
  - **library concept**
  - **keyword driven approach**

RBT
FRM
WRK

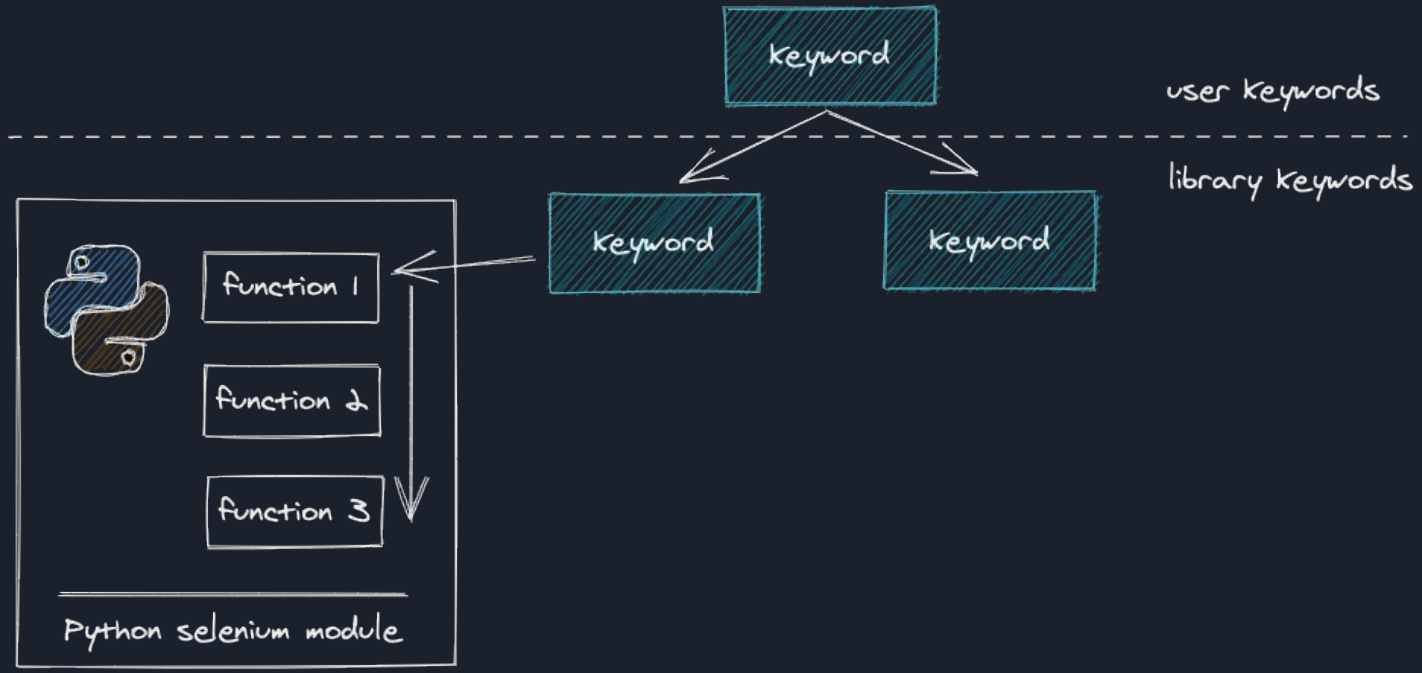
An open source acceptance test automation software that helps you to get things right.
robotframework.org

```
*** Settings ***
Library           SeleniumLibrary
Suite Teardown    Close All Browsers

*** Variables ***
${URL} =            http://www.seleniumeasy.com/test/basic-first-form-demo.html

*** Test Cases ***
Selenium Demo
    Open Browser                ${URL}   chrome  service_log_path=null
    Maximize Browser Window
    Wait Until Element Is Visible  at-cv-lightbox-close  timeout=10  error=None
    Click Element               at-cv-lightbox-close
    Page Should Contain         Selenium Easy Demo - Simple Form to Automate using Selenium
    Input Text                  user-message  Test Robot
    Click Button                css:#get-input > .btn
    Wait Until Element Contains  display  Test Robot  timeout=3  error="Testtext konnte nicht gefunden werden!"
```

ROBOT FRAMEWORK KEYWORDS

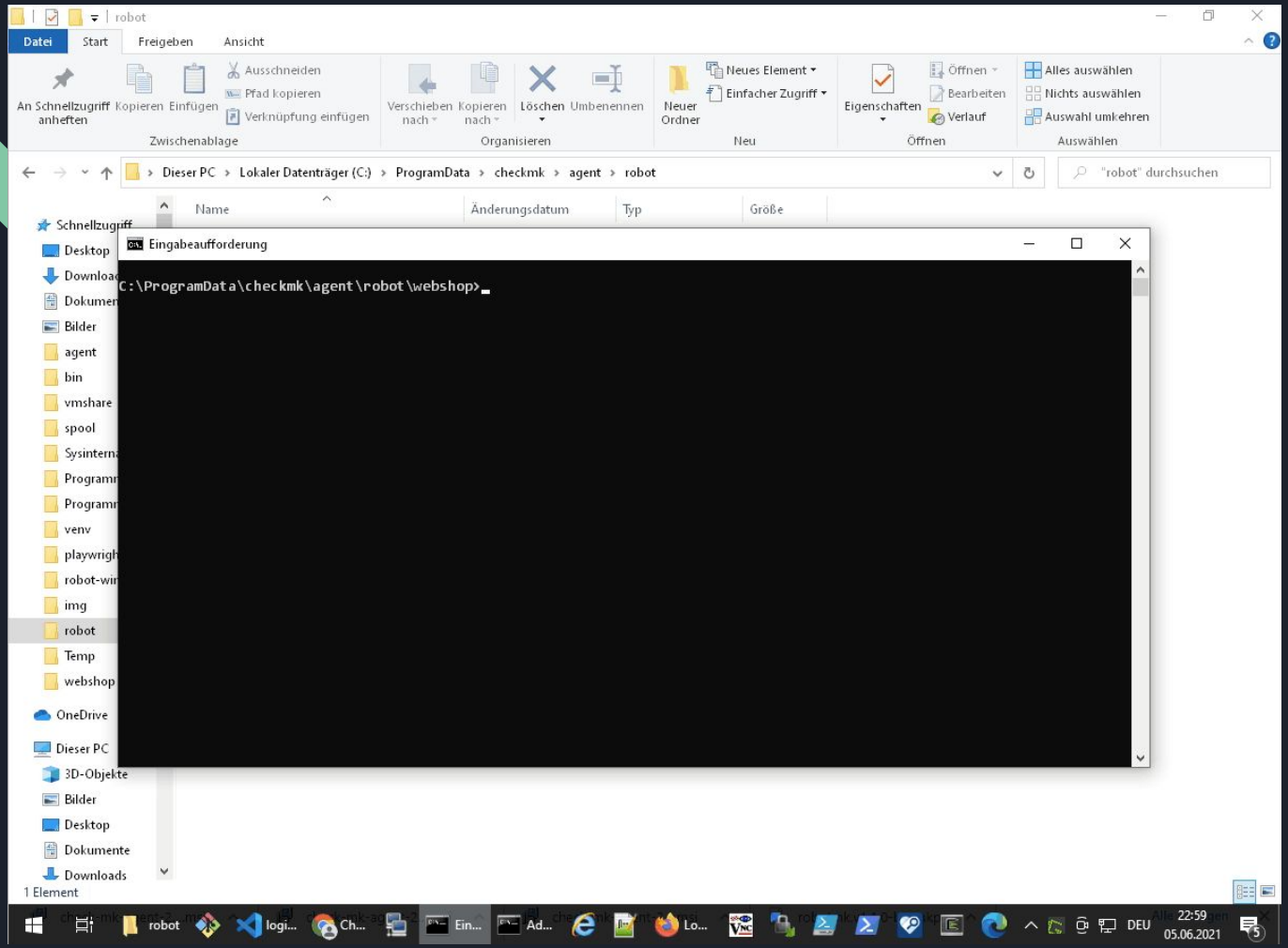


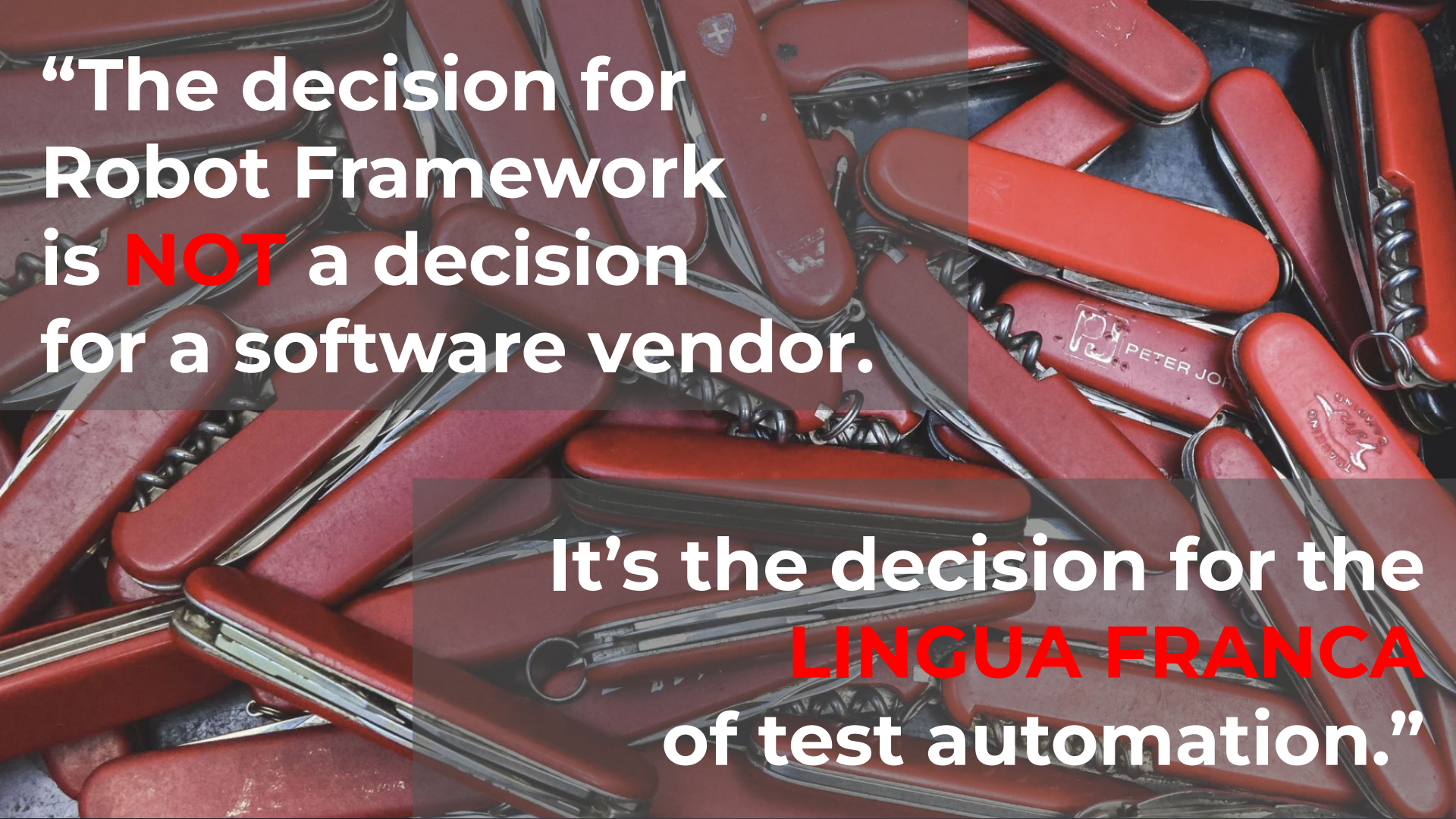
Libraries for every use case

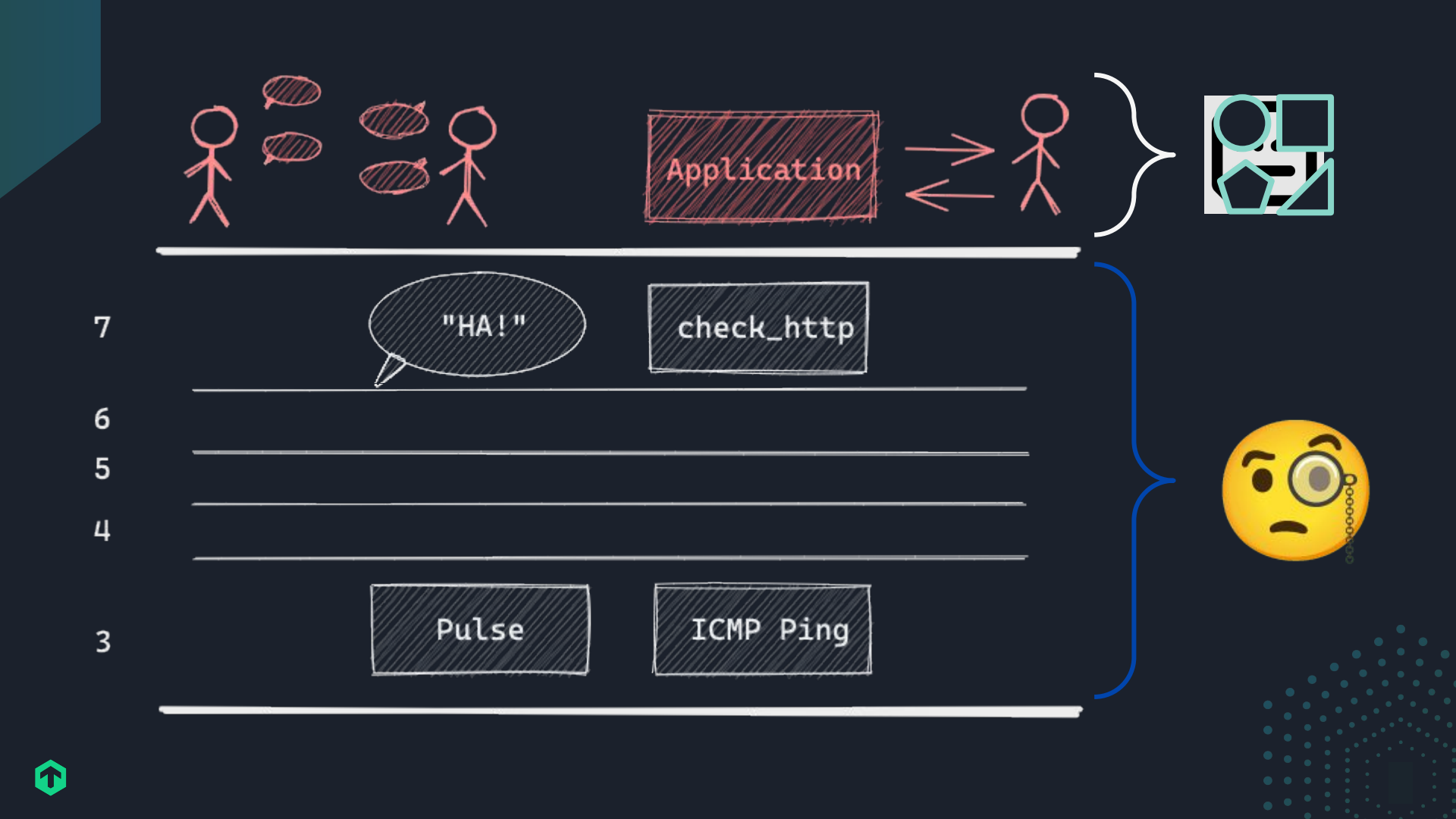
- Web automation (**SeleniumLibrary**, **Playwright**)
- Desktop/OS automation
  - by **Win32** API (FlaUI, Zoomba.Desktop, AutoIT)
  - by Image pattern recognition (**ImageHorizonLibrary**)
- Network (SSH, REST, SOAP, Telnet, ...)
- **Kubernetes** (KubernetesLibrary)
- **PDF/Image** comparison (DocTest library)
- **CryptoLibrary** (encrpyt sensitive data)
- many more...
- self-written libraries (Python functions become keywords)

Robot Framework in action

- demo web shop
- Windows UI


robot
Datei
Start
Freigeben
Ansicht
An Schnellzugriff anheften
Kopieren
Einfügen
Ausschneiden
Pfad kopieren
Verknüpfung einfügen
Zwischenablage
Verschieben nach
Kopieren nach
Löschen
Umbenennen
Organisieren
Neuer Ordner
Neues Element
Einfacher Zugriff
Neu
Eigenschaften
Öffnen
Bearbeiten
Verlauf
Öffnen
Alles auswählen
Nichts auswählen
Auswahl umkehren
Auswählen
Dieser PC
Lokaler Datenträger (C:)
ProgramData
checkmk
agent
robot
"robot" durchsuchen
Name
Änderungsdatum
Typ
Größe
Schnellzugriff
Desktop
Bilder
agent
bin
vmshare
spool
venv
img
robot
Temp
webshop
OneDrive
Dieser PC
3D-Objekte
Bilder
Desktop
Dokumente
Downloads
1 Element
Eingabeaufforderung
C:\ProgramData\checkmk\agent\robot\webshop>
DEU
22:59
05.06.2021


“The decision for
Robot Framework
is NOT a decision
for a software vendor.
It’s the decision for the
LINGUA FRANCA
of test automation.”

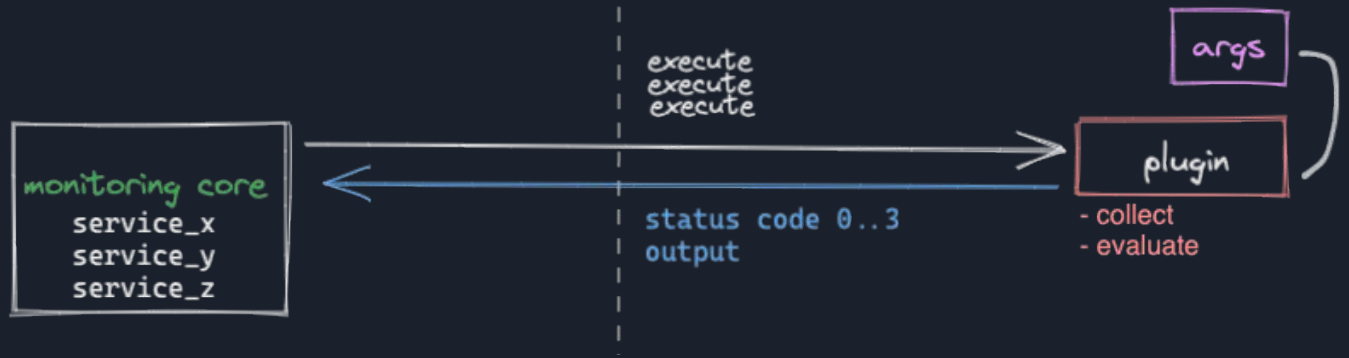

Application
7
"HA!"
check_http
6
5
4
3
Pulse
ICMP Ping

ROBOTMK: THE BRIDGE IN BETWEEN

- 100% ROBOT FRAMEWORK COMPATIBLE (NO “MAGIC”)
- 100% CONFIGURABLE BY CHECKMK’S WEB UI
- 100% RAW DATA POWERED

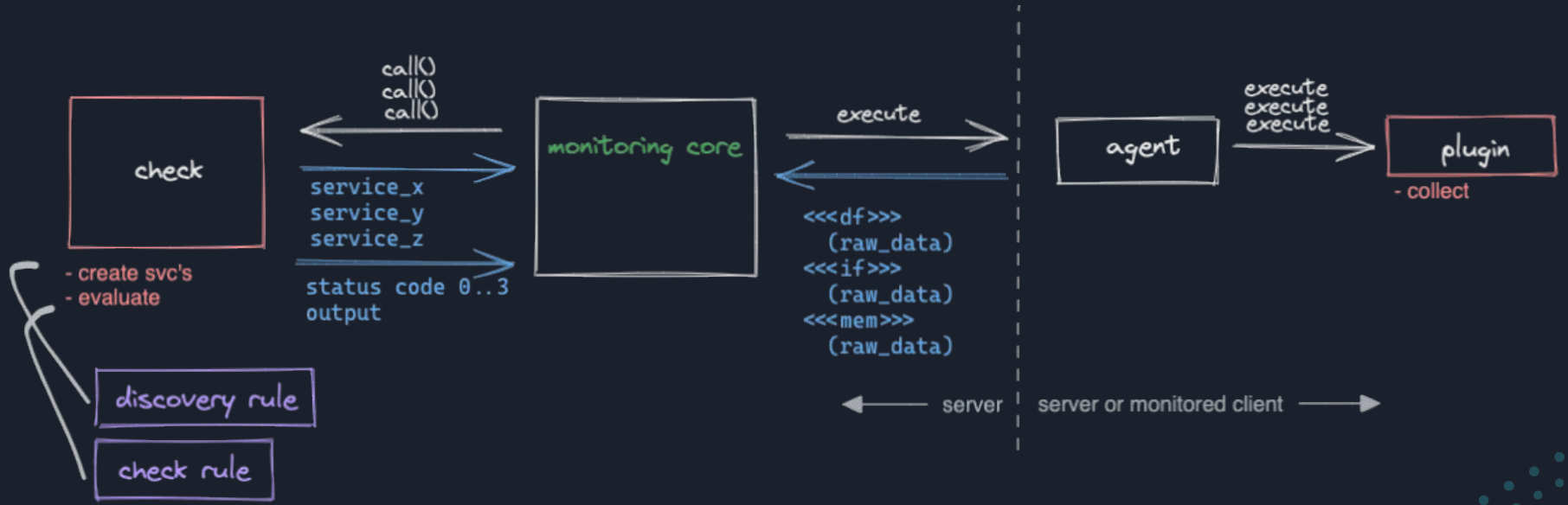
MONITORING ARCHITECTURES

Classical Plugin concept of Nagios, Naemon, Icinga, …

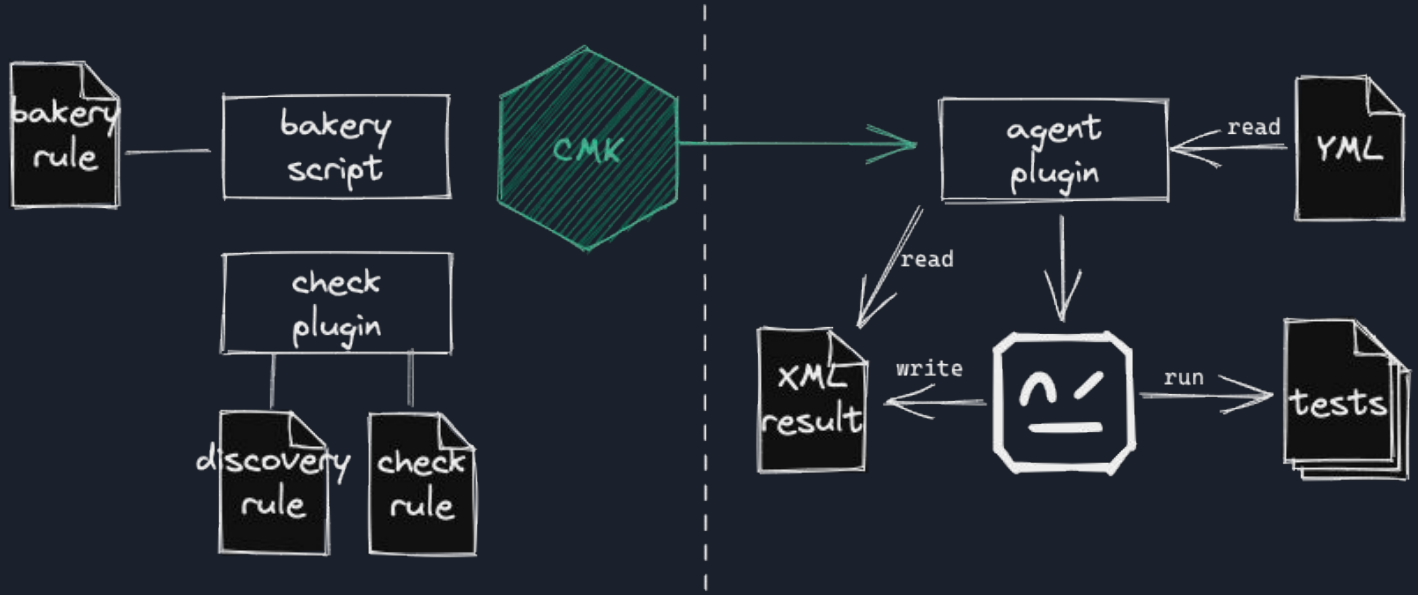


MONITORING ARCHITECTURES

Checkmk concept



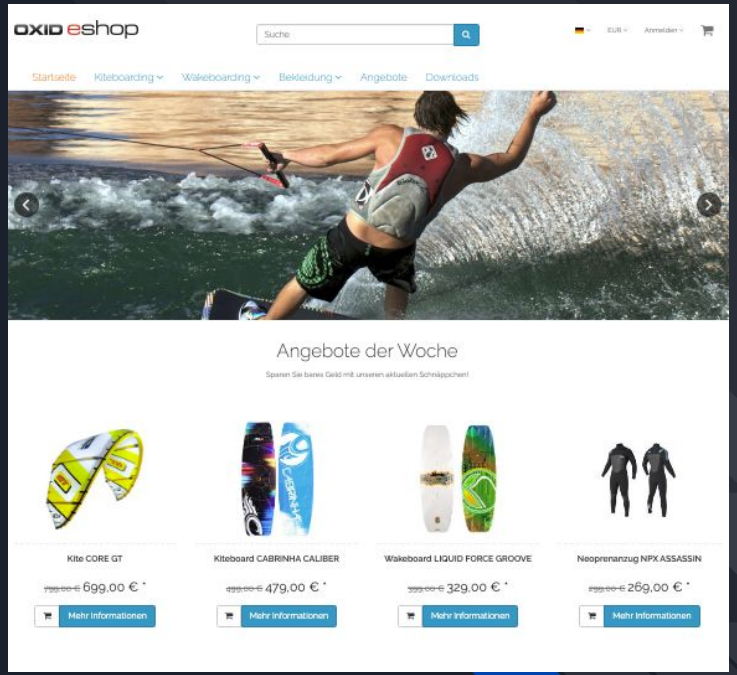
How Robotmk works



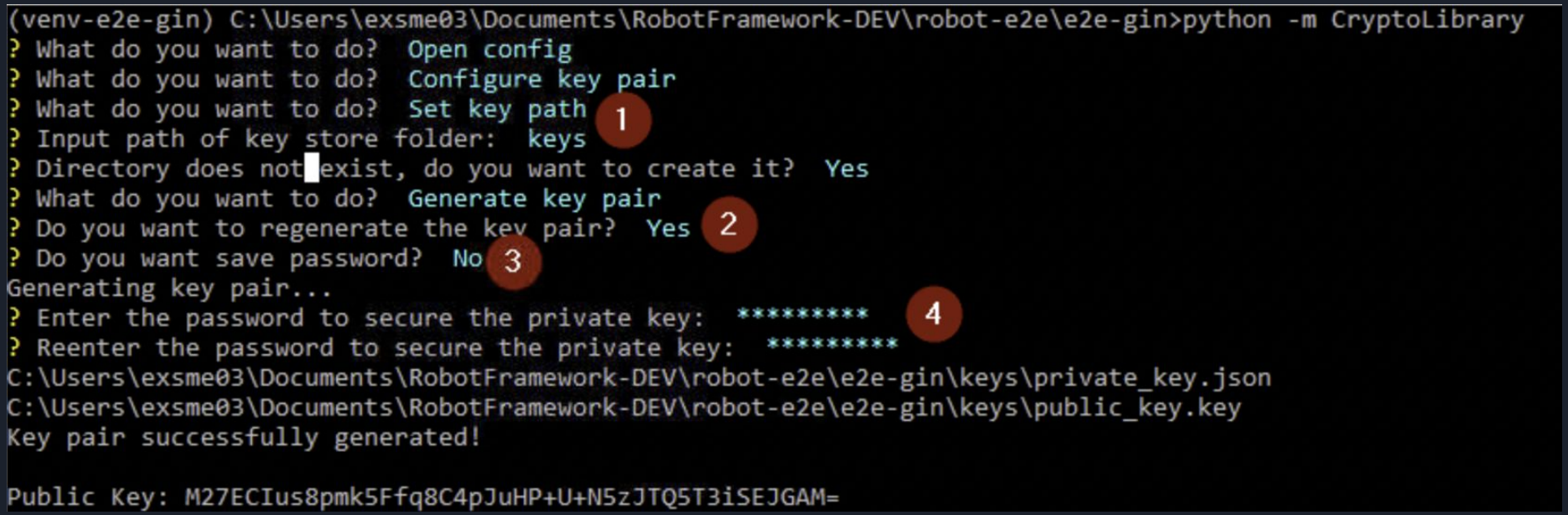
2. Robot Framework Basics

- Installation
- Syntax

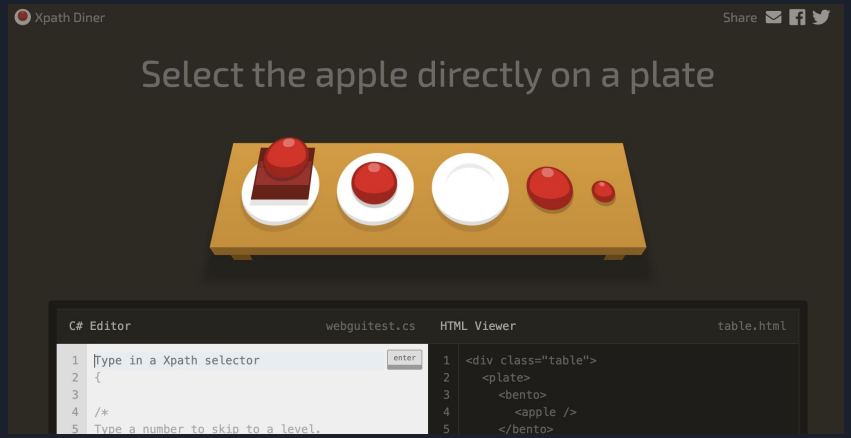
3. Web Testing mit Browser Library

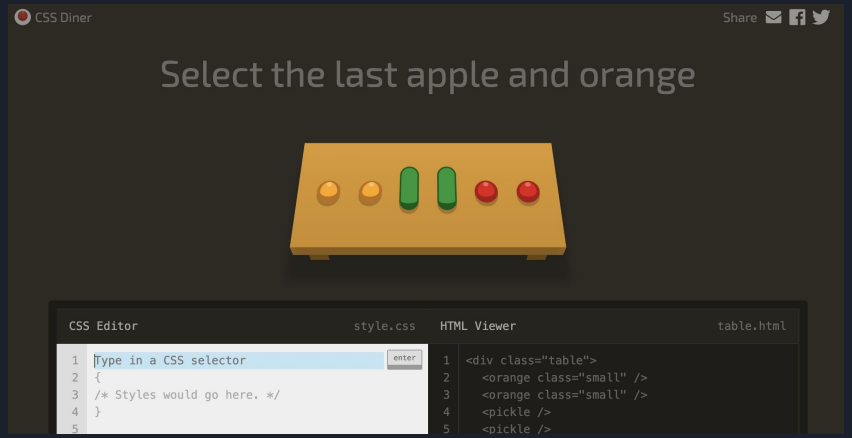


CryptoLibrary: Generate keypair



CryptoLibrary: Encrypt secret

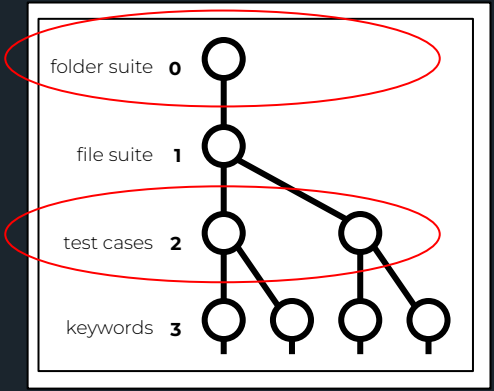




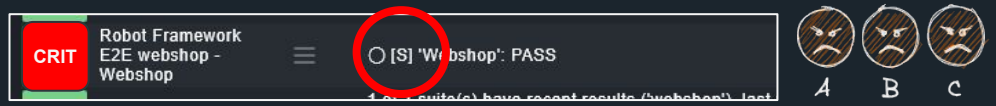
https://topswagcode.com/xpath/
https://flukeout.github.io

4. Checkmk: Einrichtung und Konfiguration

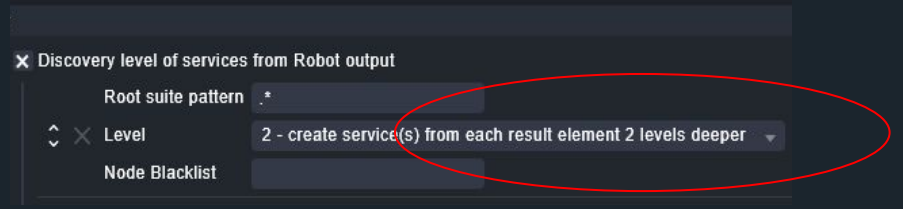
Step 5: Discover the Robot services

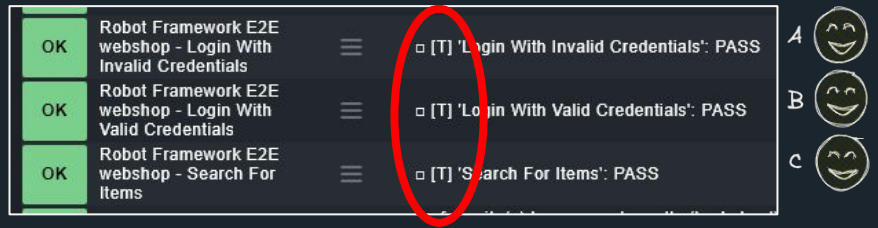


Default = 1 service per **RF suite**



Create one service per **RF test**





Re-execution of failed tests