



Containerized End-2-End Testing

Sven Hettwer
Software developer

10.10.2018

Agenda

- End-2-End Testing und seine Herausforderungen
- Eine Testumgebung im Container
- End-2-End Testing mit Sakuli
- Ausführung von Containern
- Demo
- Zusammenfassung



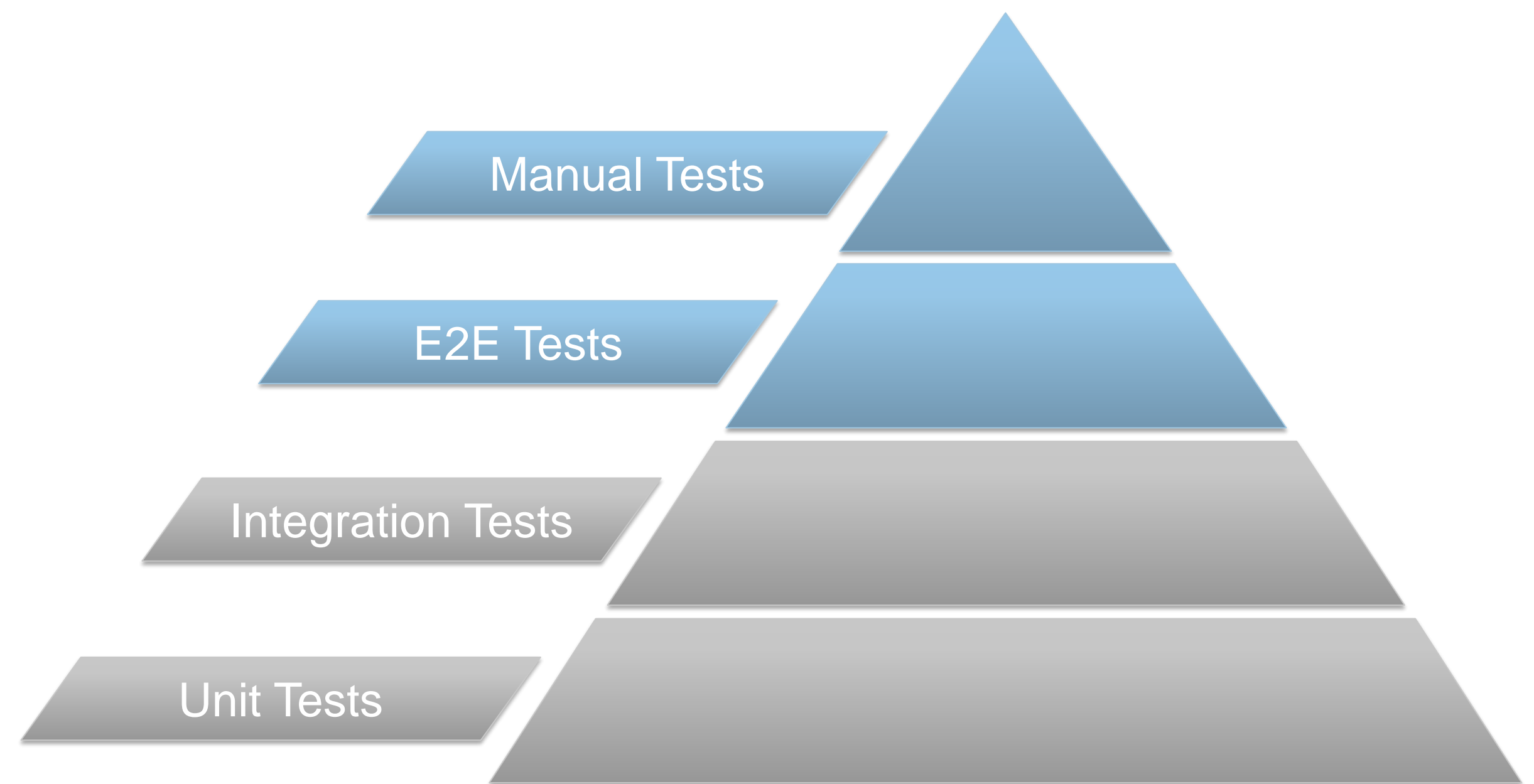
Agenda

- End-2-End Testing und seine Herausforderungen
- Eine Testumgebung im Container
- End-2-End Testing mit Sakuli
- Ausführung von Containern
- Demo
- Zusammenfassung



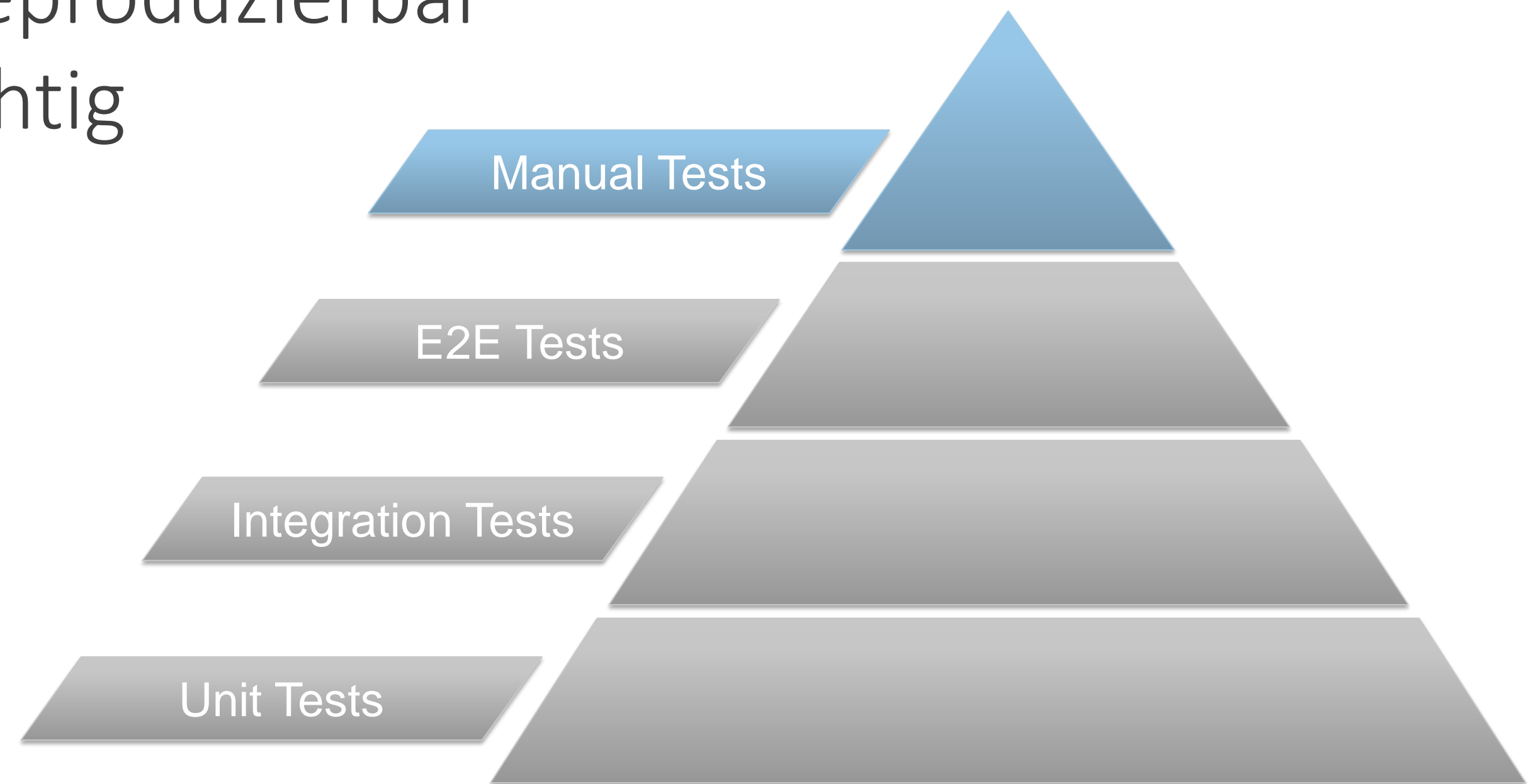
End-2-End Testing und seine Herausforderungen

- Testen kompletter Softwaresysteme
- Testen des Systems aus Nutzersicht
- Tests sind stark von der Umgebung abhängig
- Zwei Ansätze bei der Ausführung
 - Manuell
 - Automatisch



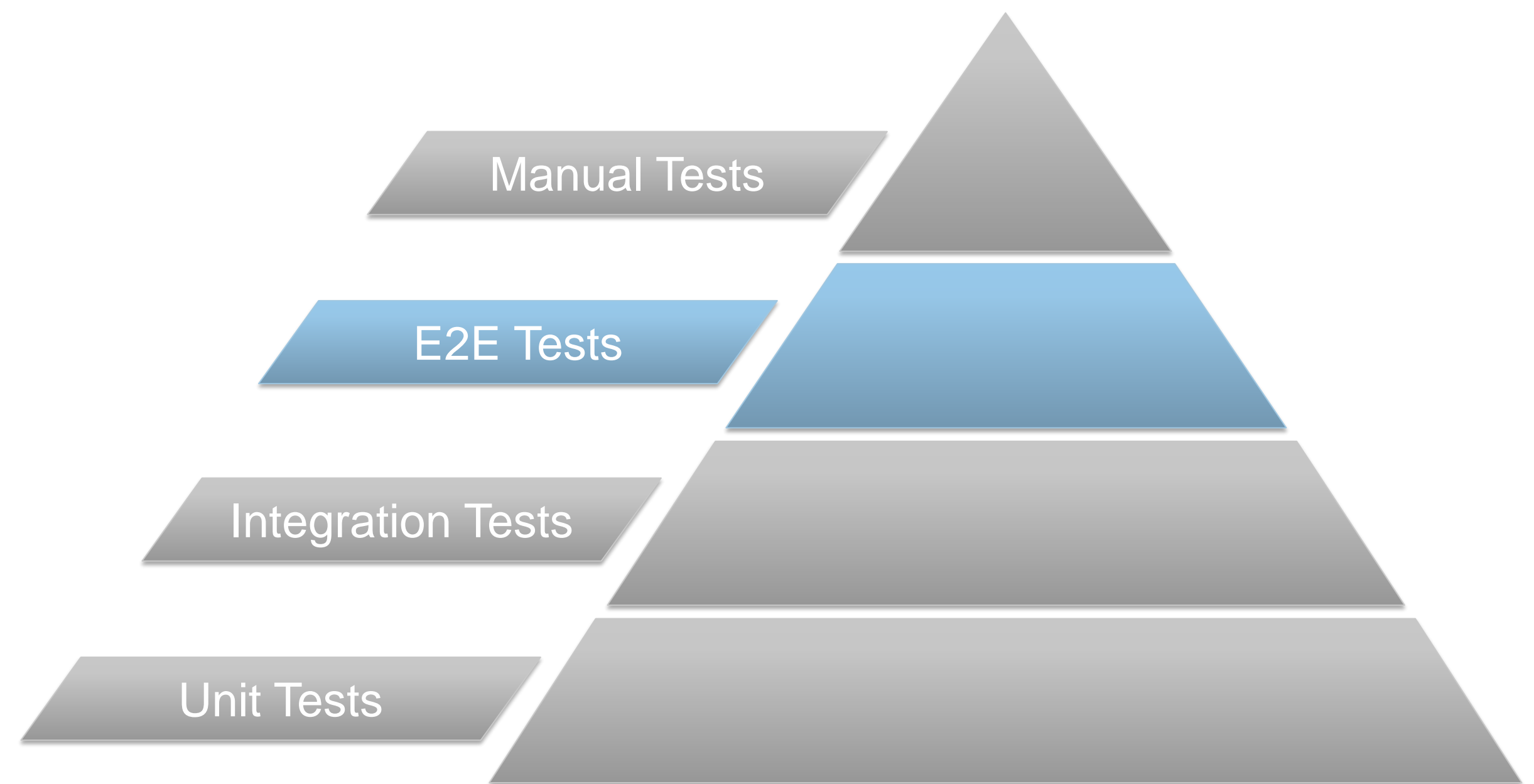
End-2-End Testing und seine Herausforderungen – Manuelle Tests

- Manuelle Tests sind
 - zeitaufwändig
 - fehleranfällig in der Ausführung
 - wiederkehrende Aufwände
- Gefundene Fehler sind schlecht reproduzierbar
- Trotzdem sind manuelle Tests wichtig



End-2-End Testing und seine Herausforderungen – Automatisierte Tests

- Testen die Funktionstüchtigkeit kritischer Pfade
- Testen im Browser und auf nativer UI
- Reduzieren manuelle Aufwände enorm
- Teuer in der Entwicklung
- Anfällig für UI-Änderungen
- Schwierig zu skalieren



Agenda

- End-2-End Testing und seine Herausforderungen
- Eine Testumgebung im Container
- End-2-End Testing mit Sakuli
- Ausführung von Containern
- Demo
- Zusammenfassung



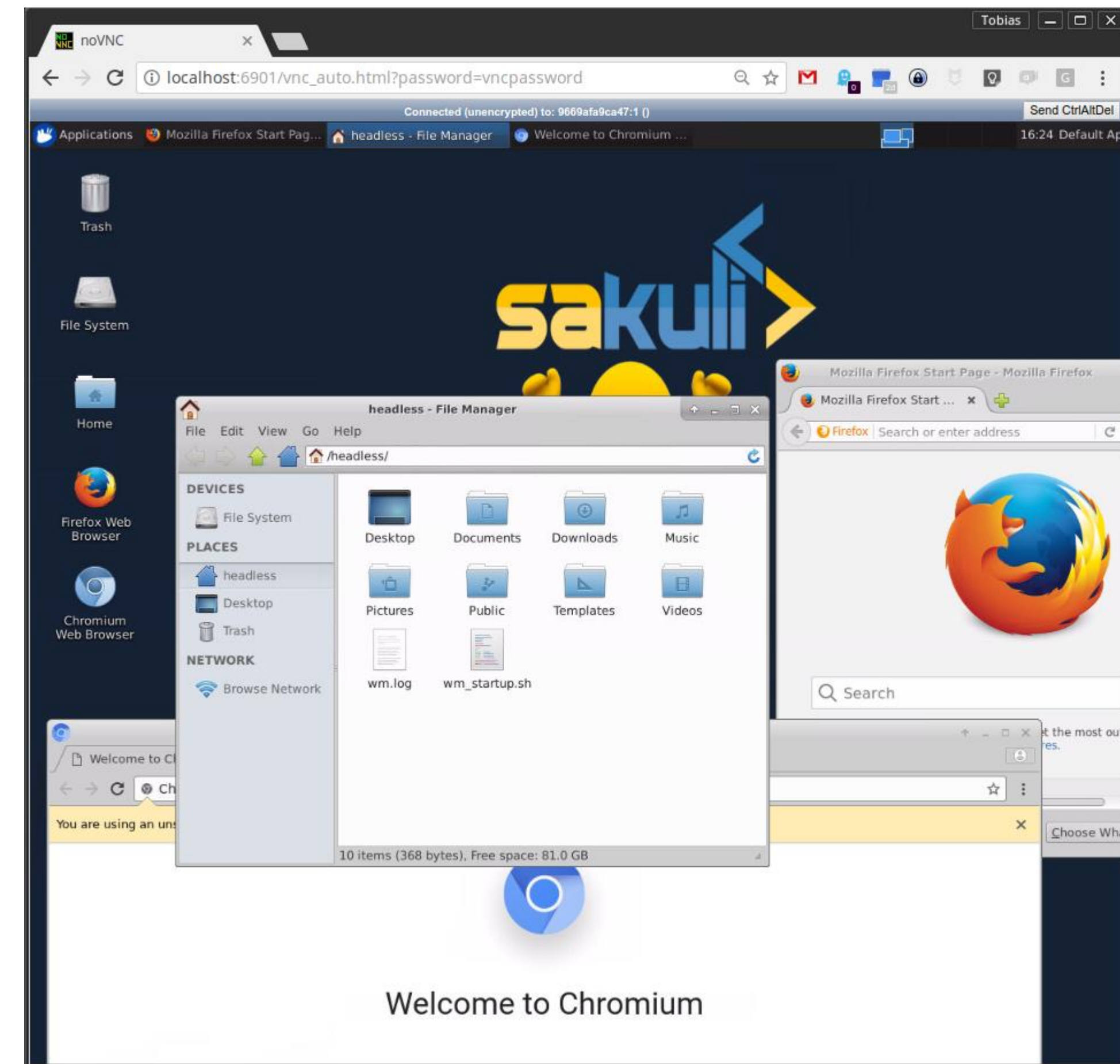
Eine Testumgebung im Container



Wie bekomme ich meine End-2-End-Tests in einen Container?!

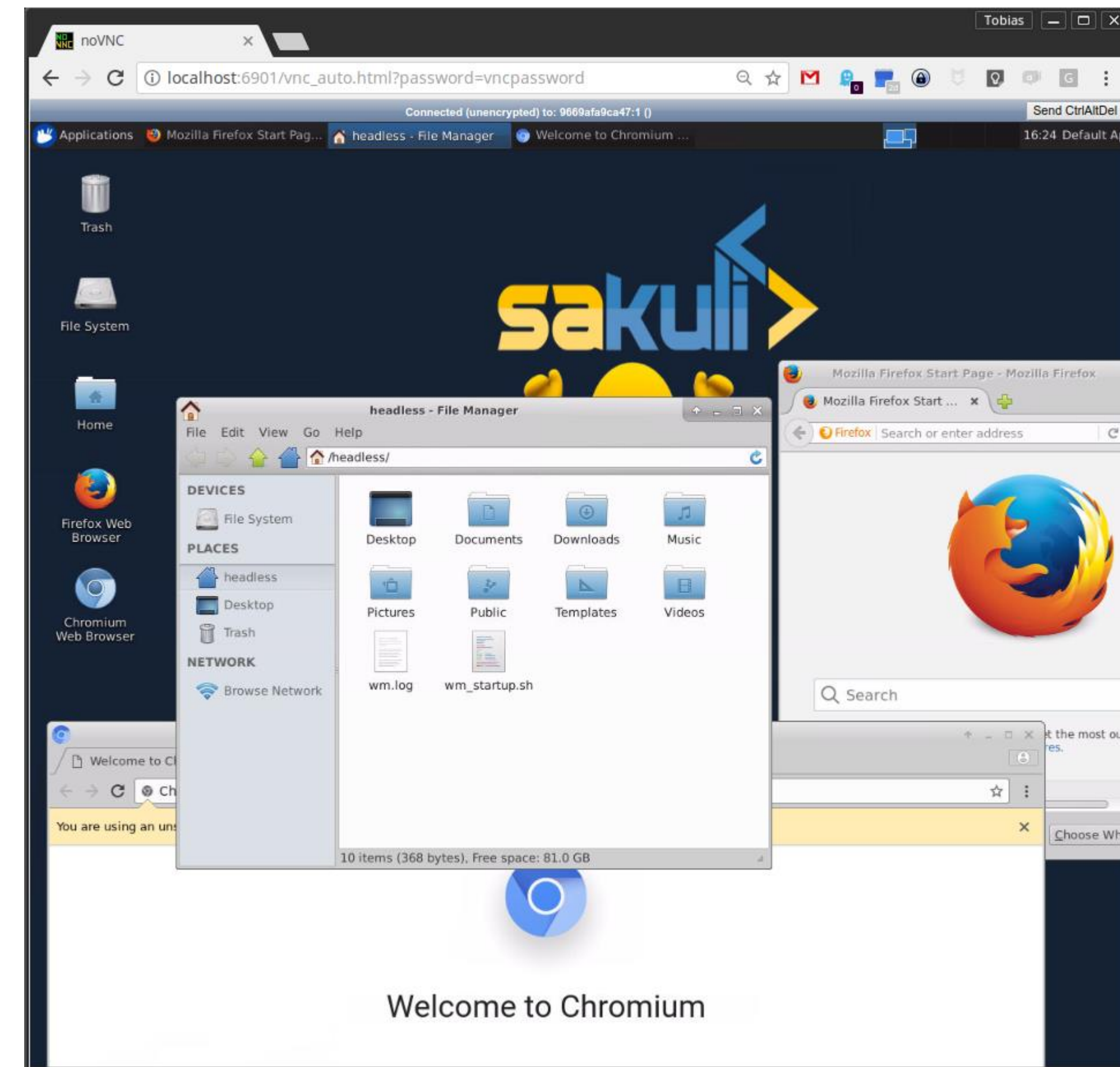
Eine Testumgebung im Container

- docker-headless-vnc-container
- Basis Docker Container mit Betriebssystem
 - Verschiedenste Linux-Distributionen
 - Inklusive verschiedener Desktops
 - Inklusive verschiedener Browser
- Steuerbar vom eigenen Rechner via Browser
- Erweiterbar mit Softwarepaketen, Tests, uvm.



Eine Testumgebung im Container

- Stabile, reproduzierbare Umgebung
- Bootet in Sekunden
- Skalierbar über die Cloud
- Keine Bereinigung der Umgebung notwendig
- Versionierbar über Docker Registries



Agenda

- End-2-End Testing und seine Herausforderungen
- Eine Testumgebung im Container
- End-2-End Testing mit Sakuli
- Ausführung von Containern
- Demo
- Zusammenfassung

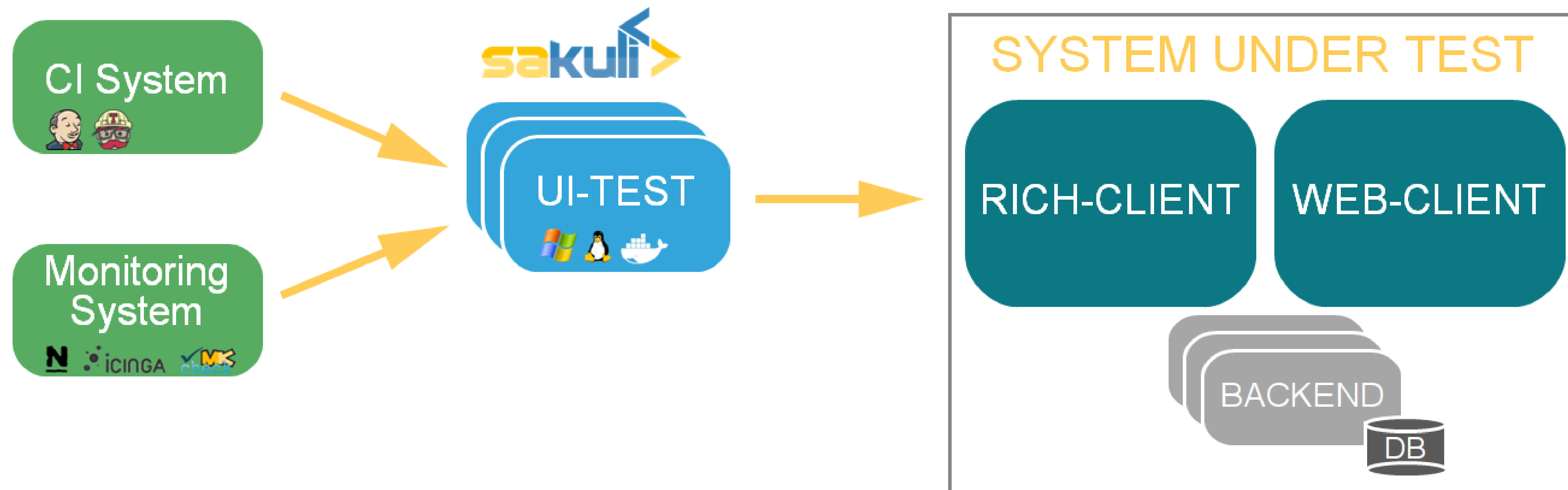


End-2-End testing with **sakuli**>

- Sakuli ist die Kombination zweier Frameworks
- Sakuli kombiniert die Stärken und eliminiert Schwächen
- Bereitstellung einer vereinheitlichten API
- Skalierbar über die Cloud
- Plattformunabhängig
- Native Integration gängiger Monitoring Tools



End-2-End testing with

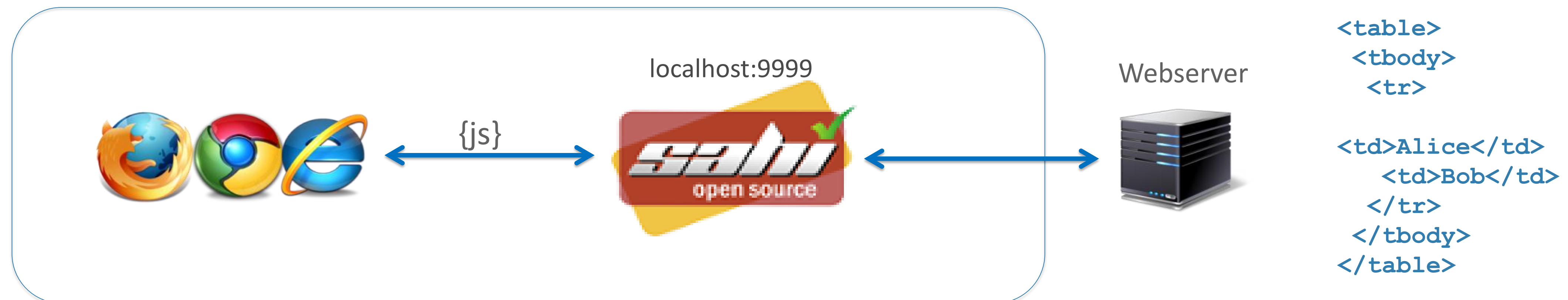


End-2-End testing with

Web testing tool (sahi.co.in, seleniumhq.org)

Methodenbasierter Zugriff auf das DOM:

```
_assertContainsText ("Logged in as: Sakuli", _div("user_field"));  
_click(_span("Loaded Run Tabela"));  
_assertExists(_table("cross_table_fixed"));  
_assertExists(_cell("testing allowed", _rightOf(_span("Name")), _under(_cell("Action"))));
```



End-2-End testing with

Visual automation tool (sikuli.org)

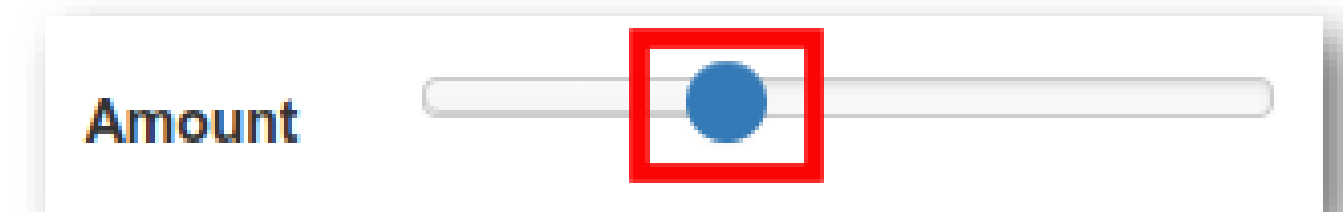
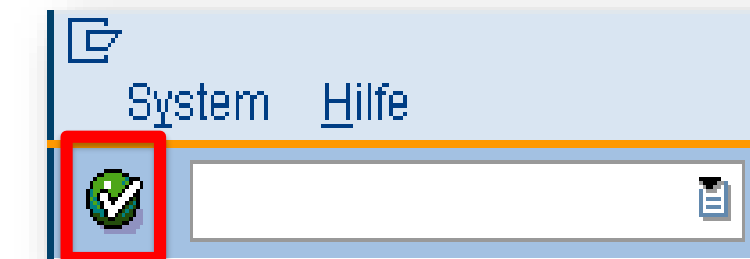
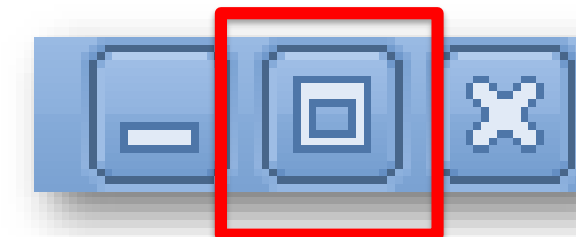
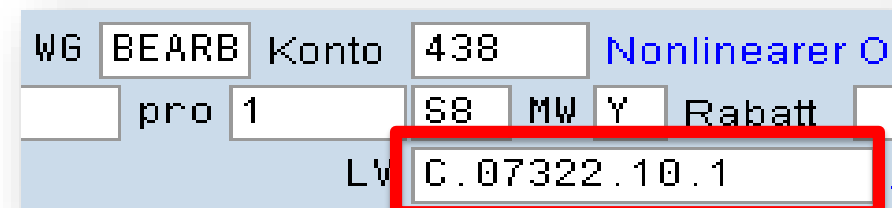
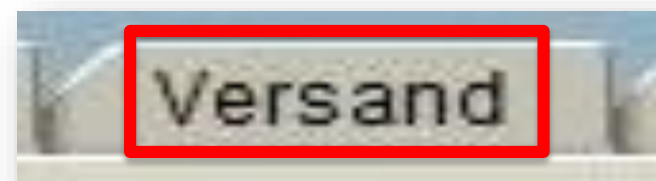
Bildererkennung mit Maus- und Tastaturinteraktion:

```
screen.find("sap_ok").click();
```

```
screen.find("sap_ok").right(40).click().type("2223");
```

```
var bubble = new Region().waitForImage("bubble.png", 20);
```

```
bubble.dragAndDropTo(bubble.left(35)).highlight();
```



End-2-End testing with **sakuli**

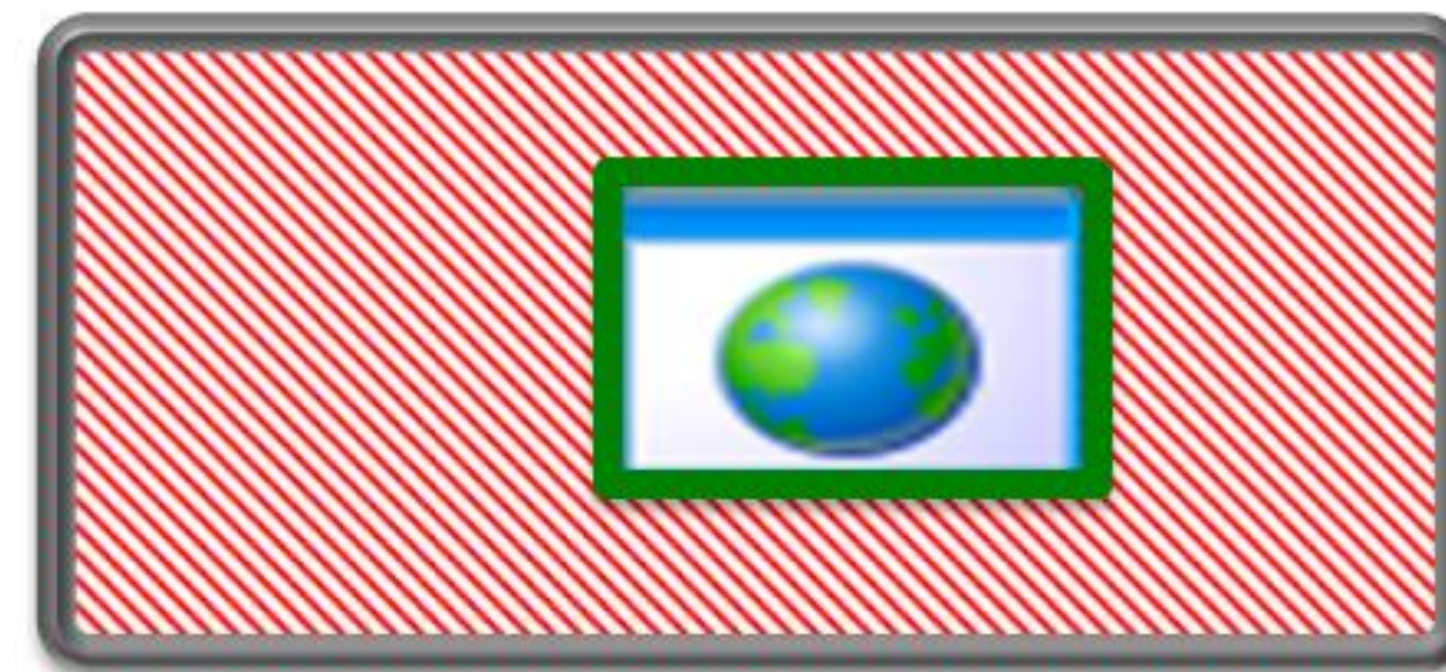
Sikuli

- ✓ universell
- ✗ ressourcenintensiv
- ✗ Bildschirm darf nicht gesperrt sein



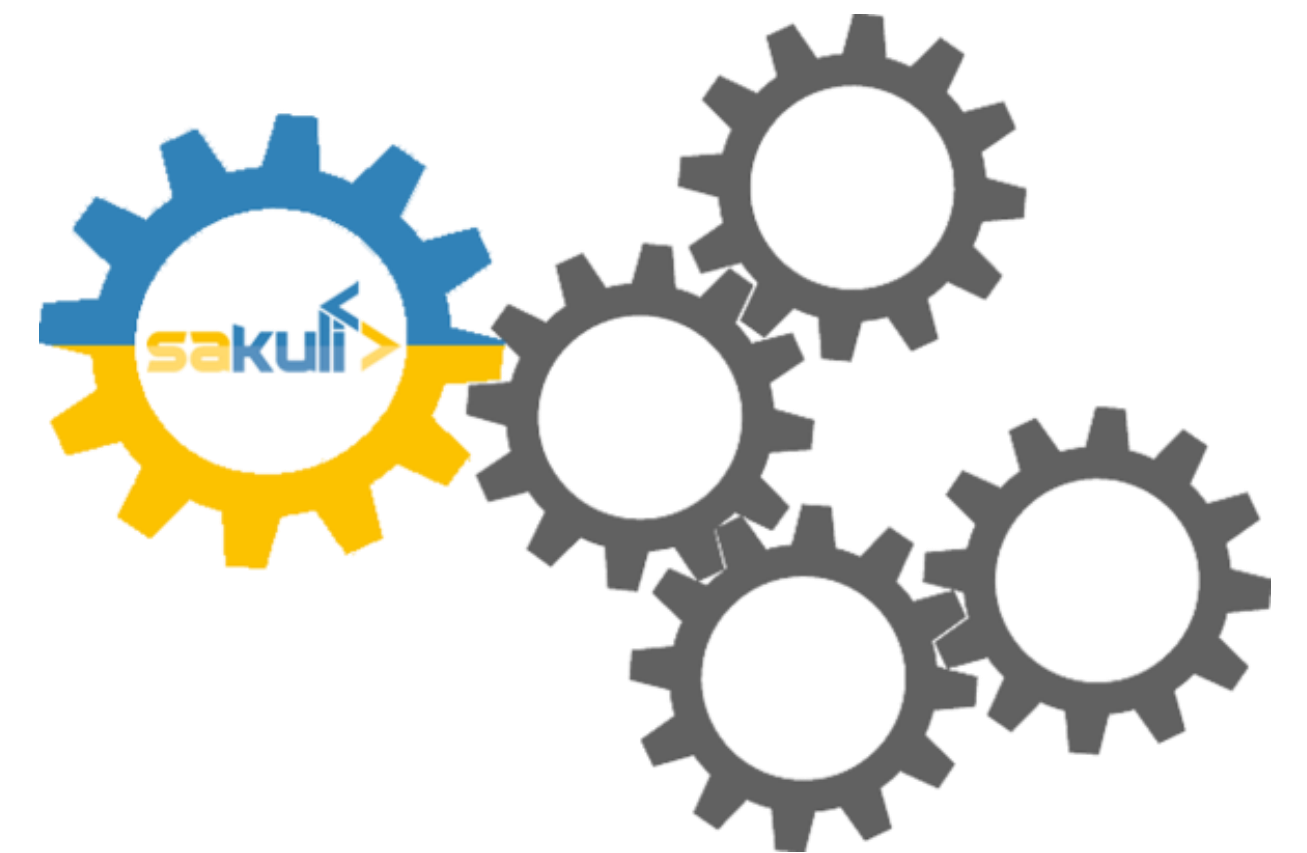
Sahi / Selenium

- ✗ Limitiert auf den Browser
- ✓ DOM-basierte Tests
- ✓ Tests sind einfach zu schreiben



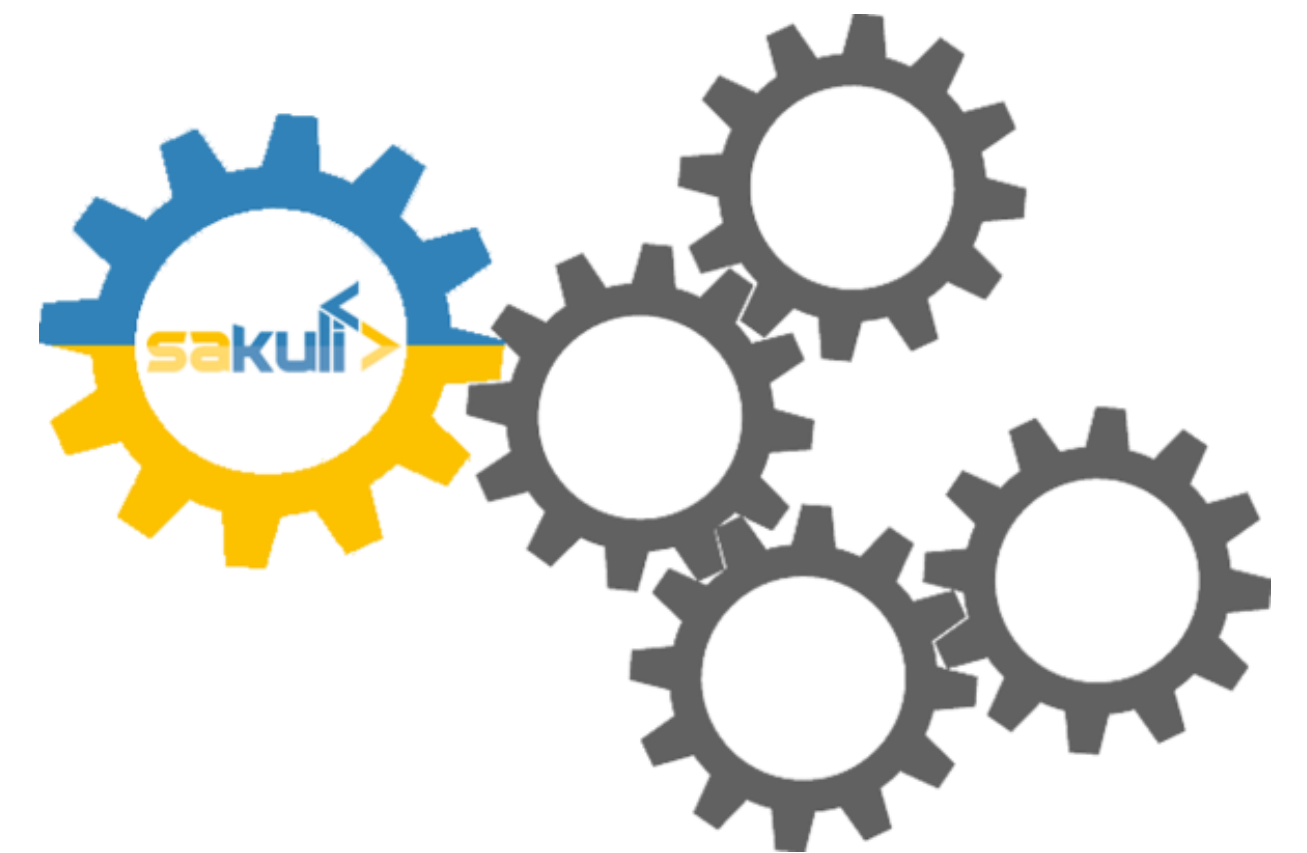
End-2-End testing with

- Sakuli bietet verschiedenste Ausführungsmöglichkeiten
 - Als native Installation
 - Als maven dependency
 - Innerhalb eines Docker containers



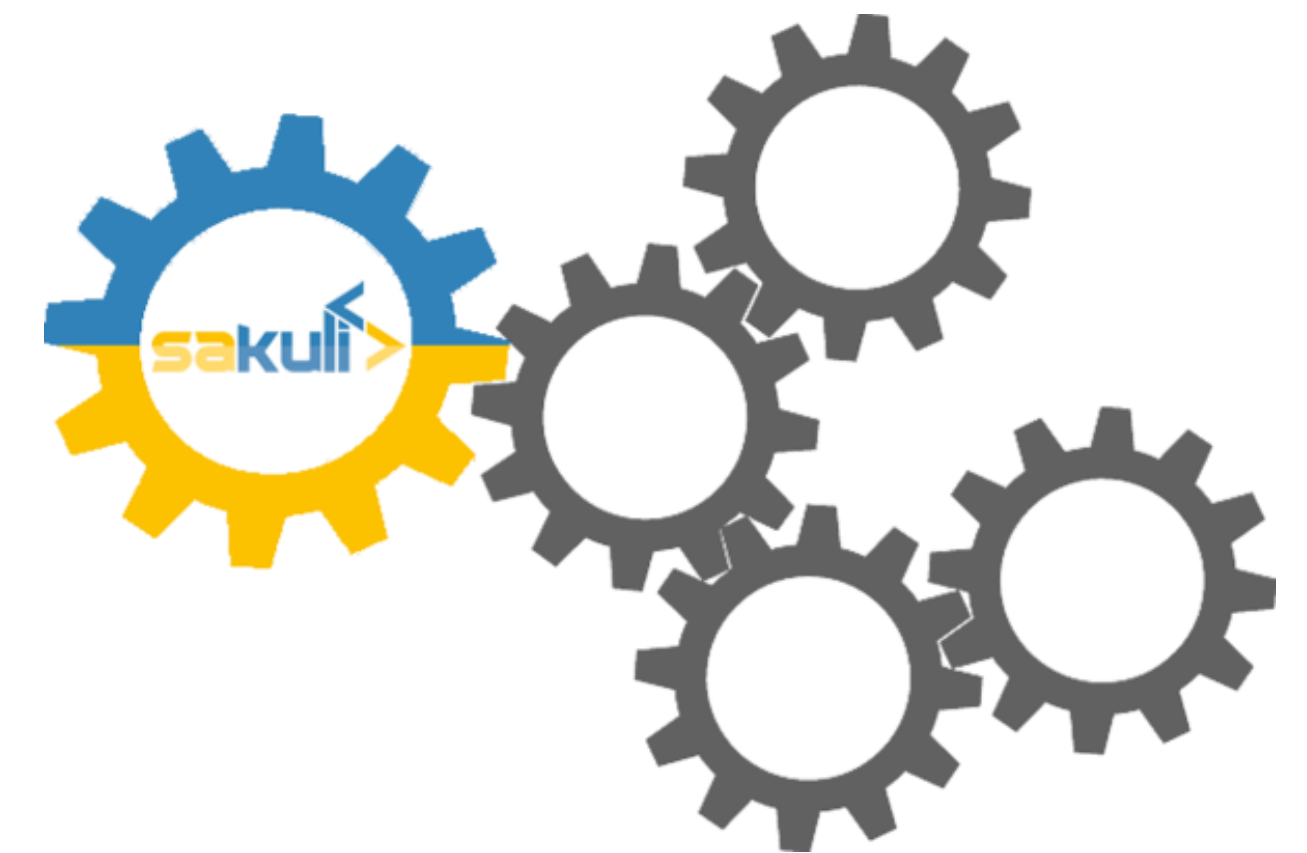
End-2-End testing with **sakuli** – Native Ausführung

- Unterstützt alle Plattformen: Windows, Linux, Mac
- Installierbar auf dem System des Benutzers
- Simple Javascript API
- Ausführung der Tests ohne vorheriges Kompilieren



End-2-End testing with **sakuli** – Ausführung mit Maven

- Einbindung als Maven dependency
- Java-DSL zum Verfassen von Tests
- Einfach integrierbar in den Maven life cycle
- Unterstützung bei der Entwicklung durch die IDE
- Optional: Selenium als Web-Testing-Engine



End-2-End testing with – Ausführung im Container

- Unterstütze Container-Technologien:
 - Docker Compose
 - Kubernetes
 - OpenShift
 - Source to image
- Sofortverfügbare Testumgebung
- Erlaubt sowohl Java als auch Javascript tests



End-2-End testing with – Ausführung im Container

- Macht sich den headless vnc container zunutze
- Ermöglicht Tests mit
 - Echten Desktop-Elementen
 - Einem echten Browser
 - Nativen Fat-Clients
- Einfach skalierbare über Cloud-Technologien



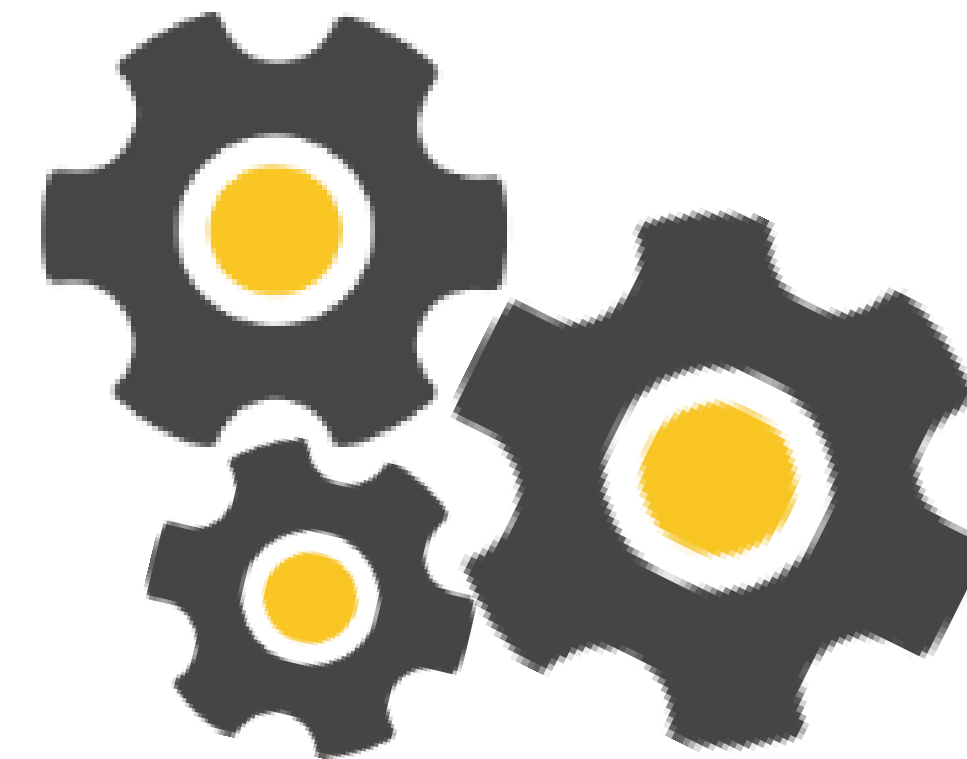
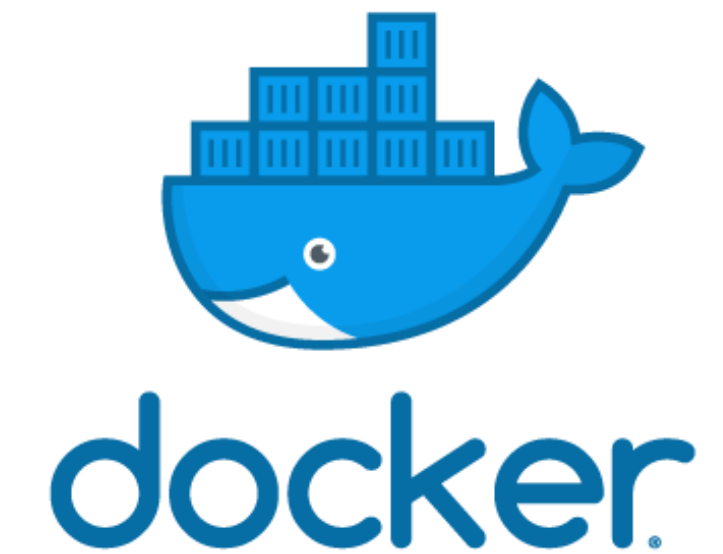
Agenda

- End-2-End Testing und seine Herausforderungen
- Eine Testumgebung im Container
- End-2-End Testing mit Sakuli
- Ausführung von Containern
- Demo
- Zusammenfassung



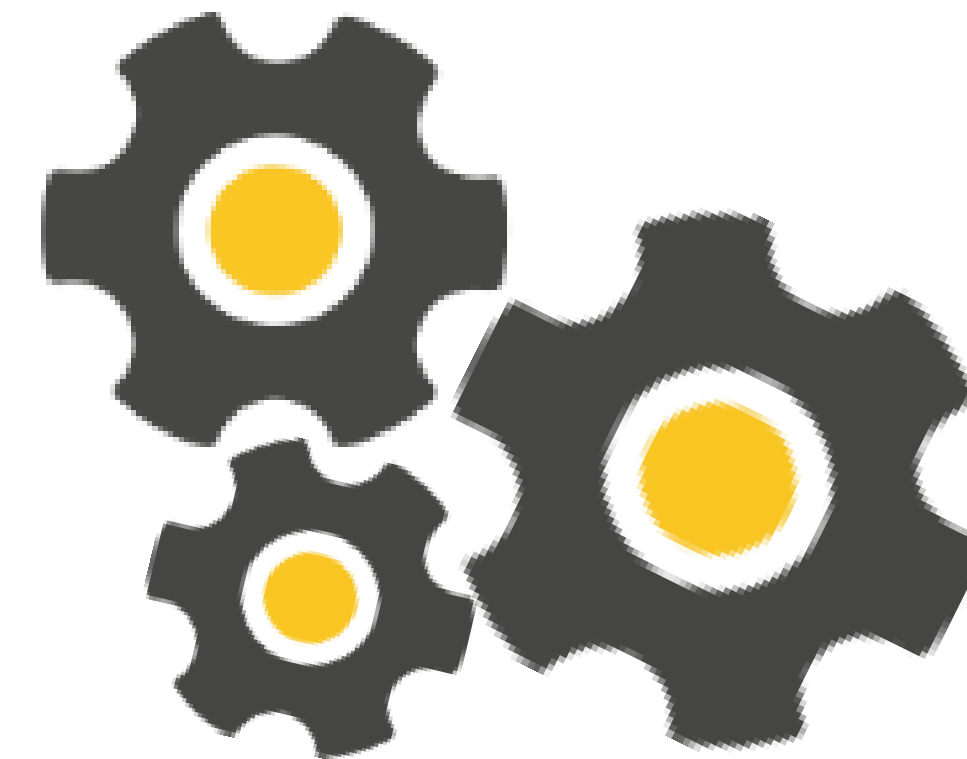
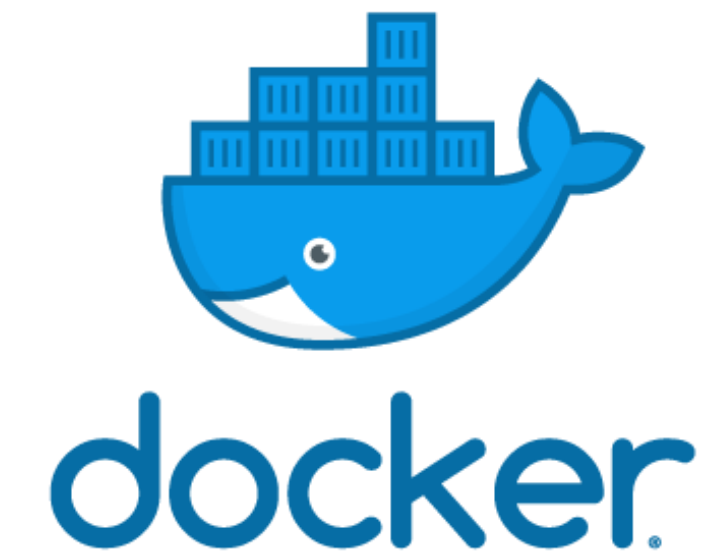
Ausführung von Containern

- Verschiedenste Möglichkeiten
 - Manuell
 - Build pipelines
 - Cronjobs



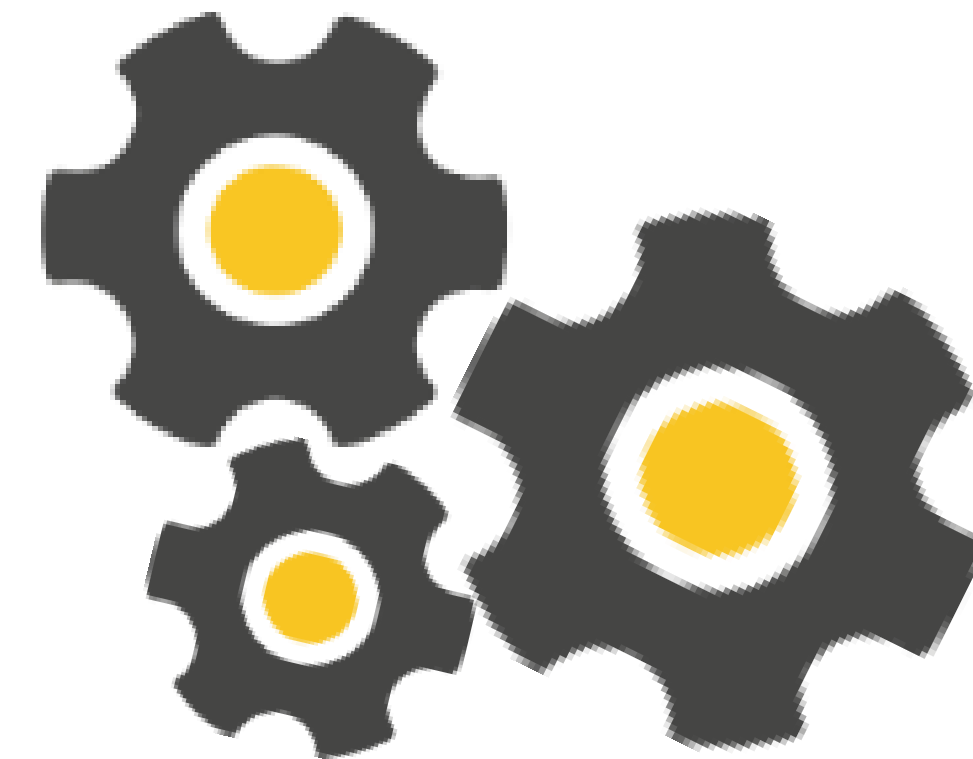
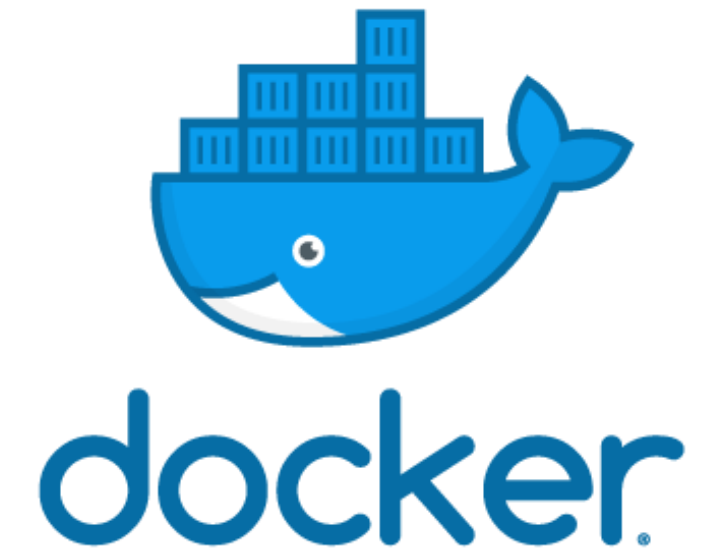
Ausführung von Containern – Manuell

- Befehl: *docker run*
- Nützlich während der Entwicklung
- Nützlich zum Rekonstruieren von findings
- Beliebt beim Prototyping



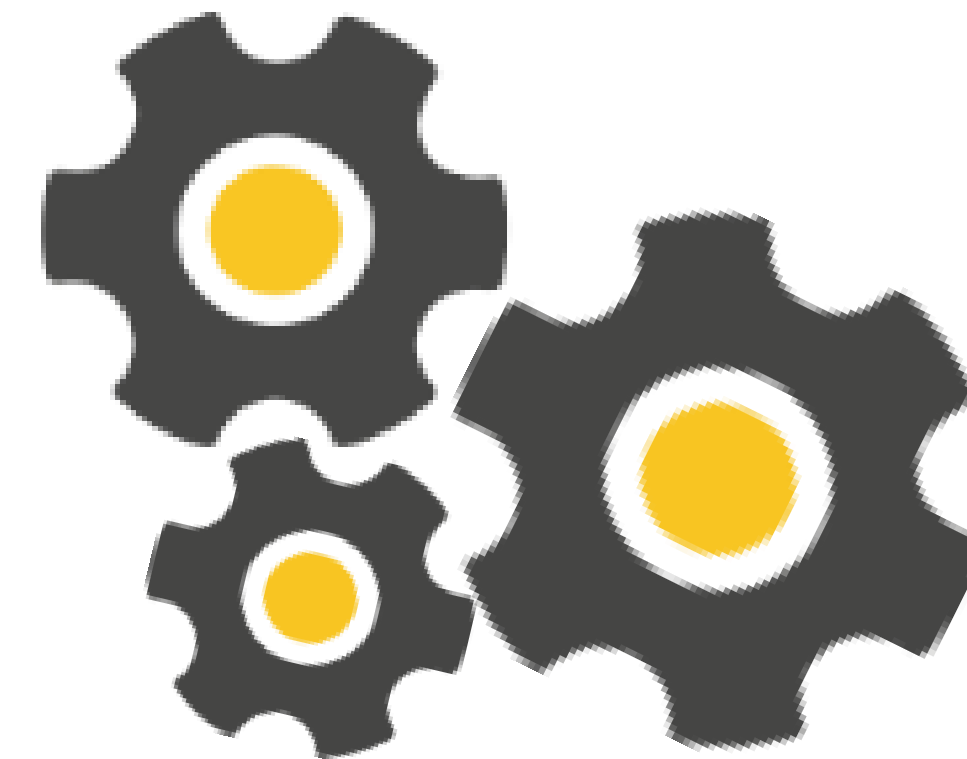
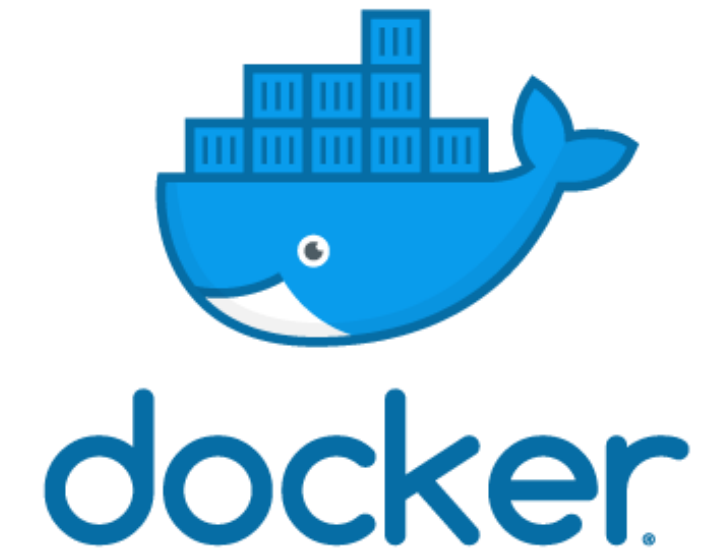
Ausführung von Containern – Build pipelines

- Build system braucht dockerd Zugriff
- Container müssen...
 - bereits gebaut sein
 - in der pipeline gebaut werden
- Verschiedenste Anbieter
- Ermöglicht automatisierte End-2-End Tests
- Tests sind Teil des development life cycle



Ausführung von Containern – Cronjobs

- Ausführung über container platform / container orchestration
 - OpenShift
 - Kubernetes
 - Rancher
- Zeitplan für Ausführung von Containern
- Besonders interessant für Monitoringzwecke



Agenda

- End-2-End Testing und seine Herausforderungen
- Eine Testumgebung im Container
- End-2-End Testing mit Sakuli
- Ausführung von Containern
- Demo
- Zusammenfassung



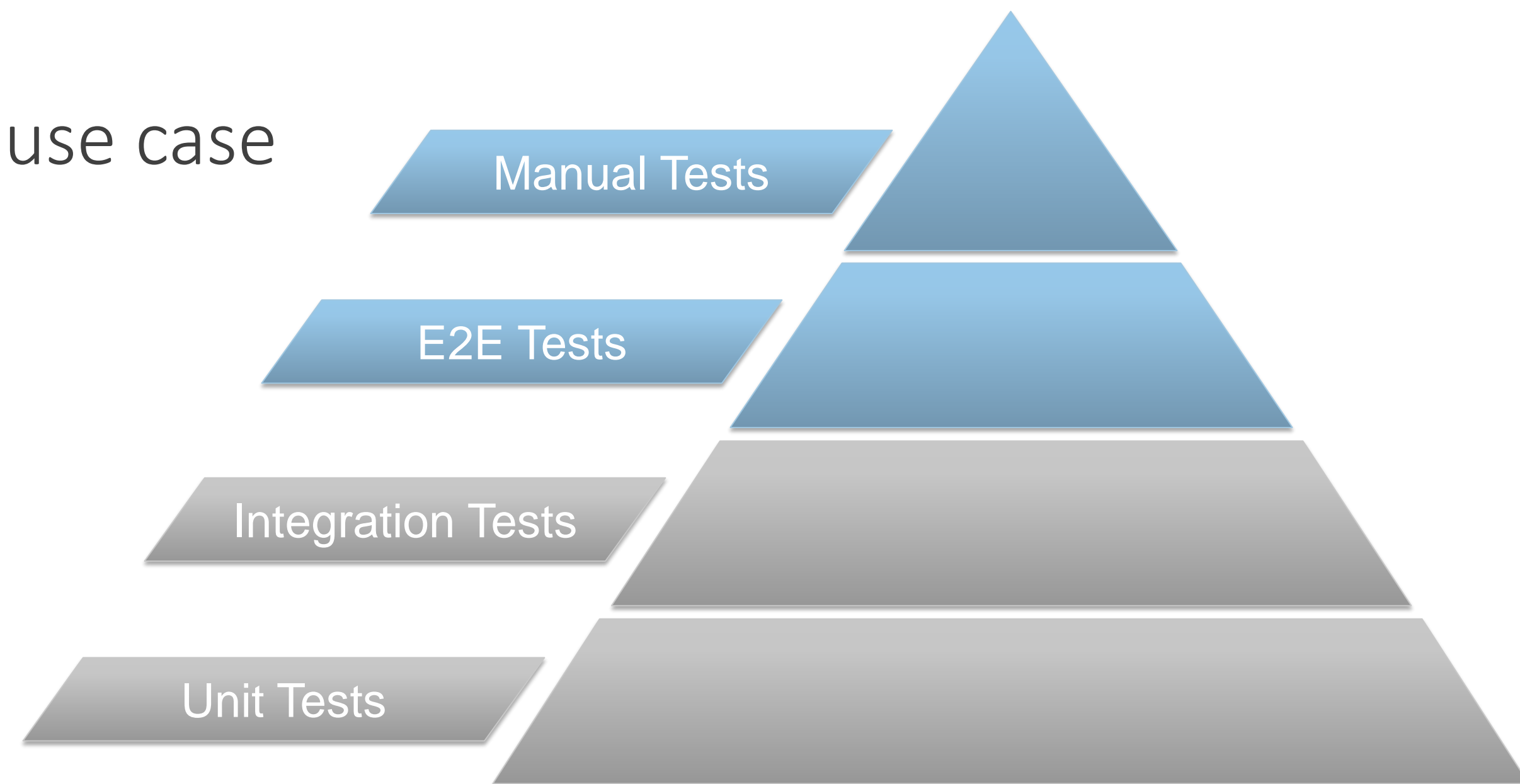
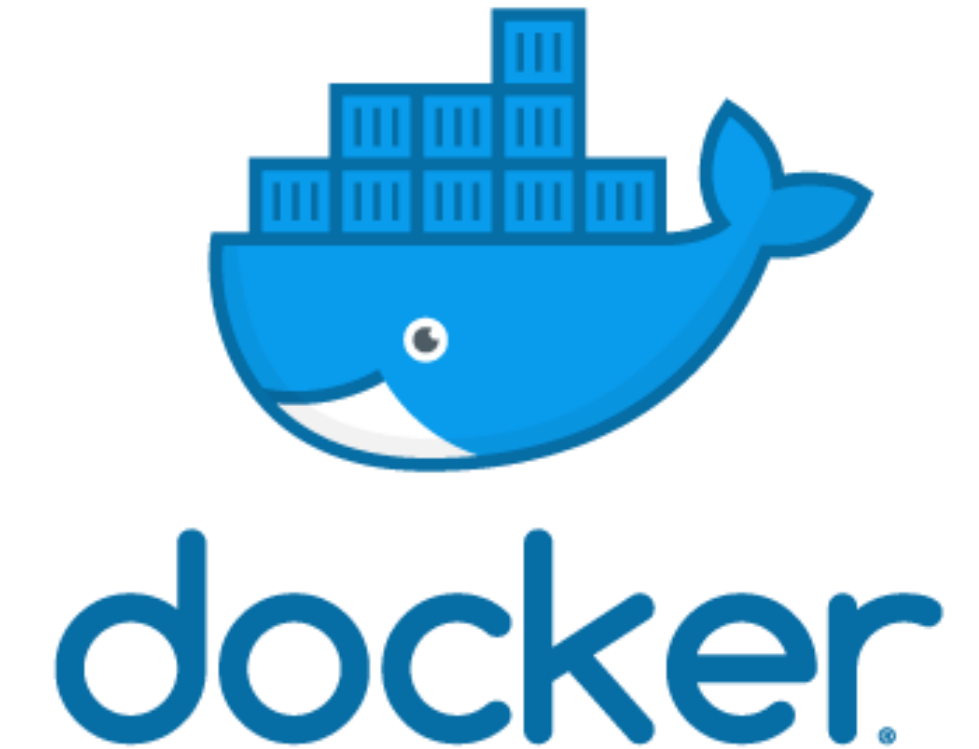
Agenda

- End-2-End Testing und seine Herausforderungen
- Eine Testumgebung im Container
- End-2-End Testing mit Sakuli
- Ausführung von Containern
- Demo
- Zusammenfassung



Zusammenfassung

- End-2-End tests sind nicht trivial
- Testumgebungen in Containern lösen viele Probleme
- Sakuli
 - Bietet vielseitige Einsatzmöglichkeiten
 - Plattformunabhängig
 - Bietet Skalierungsmöglichkeiten
- Diverse Ausführungsmöglichkeiten je nach use case









Fragen?



Sven Hettwer
Software developer

Kanzlerstraße 8
D-40472 Düsseldorf

-  @SvenHettwer
-  <https://github.com/svettwer>
-  Sven.Hettwer@consol.de
-  +49-211-339903-86



Vielen Dank!