

## End-to-End Application Monitoring

Sakuli: Open-Source Anwendungsüberwachung  
durch die Nutzerbrille



### Das Test-Framework im Überblick – Ihr Nutzen

- End-to-End Monitoring der Anwendungsschicht - **zuverlässige Überwachung** Ihrer Applikationen aus Nutzersicht
- **Automatisiertes Simulieren** von Benutzeraktionen auf grafischen Oberflächen
- **Universelles Testwerkzeug** für jeden Use case und jede Plattform
- Nahtlose **Übermittlung der Testergebnisse** an Drittsysteme wie Nagios, Icinga & Co.
- Automatisierte GUI-Tests in **Continuous-Integration-Umgebungen**
- **Kostenersparnis** dank 100% Open-Source

### Weitere Informationen

Details zu Sakuli und der Integration in Monitoring-Lösungen wie Nagios, Icinga u.a. finden Sie unter:

<https://www.consol.de/sakuli>

<https://github.com/consol/sakuli>

Ganz gleich, ob der Login-Vorgang im Webshop abbricht, der Remote-Zugriff in den Niederlassungen nur träge reagiert oder der Tagesreport nicht die aktuellen Zahlen enthält – **Störungen oder gar Ausfälle von Applikationen kosten Zeit und Geld**. Wer die Qualität seiner Anwendungen aus Nutzersicht kennt, kann eingreifen, noch bevor funktionelle oder inhaltliche Fehler den Anwender in seiner Arbeit beeinträchtigen.

Simulation von Useraktionen & Übermittlung an Nagios, Icinga etc.

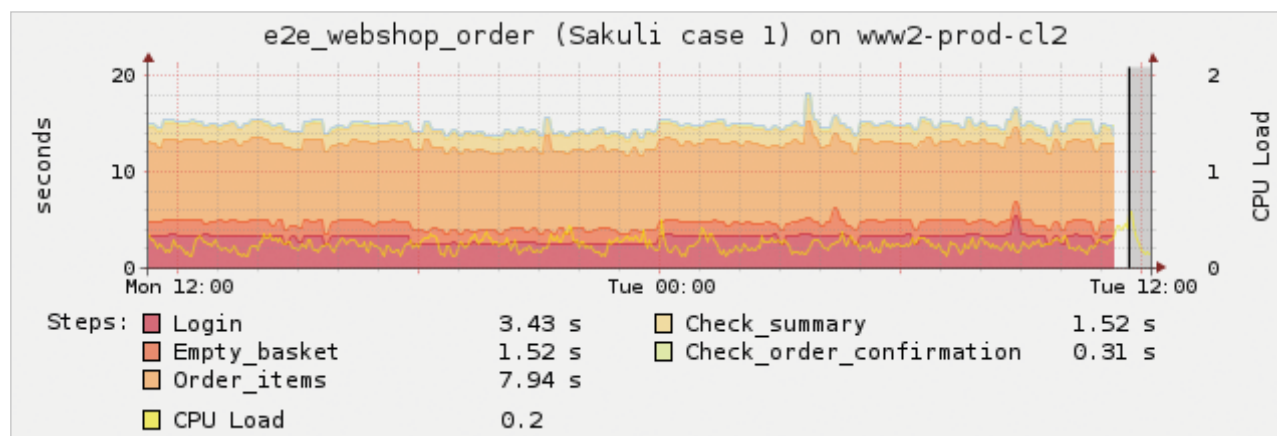
Das von ConSol\* entwickelte Framework **Sakuli** bringt alles mit, was Sie für das umfassende und automatisierte Testen von Benutzeroberflächen benötigen. Das Tool **simuliert Aktionen in beliebigen Programmen der Anwendungsschicht**, wertet deren Inhalte aus und misst dabei die Ausführungszeiten.

Sogenannte Forwarder-Module übermitteln zudem alle Testergebnisse sowie Screenshots gegebenenfalls aufgespürter Fehler an Drittsysteme wie Nagios oder Icinga, wo die Laufzeitwerte verarbeitet und in Graphen visualisiert werden. Alternativ dazu lassen sich Sakuli-Tests nahtlos in Continuous-Integration-Umgebungen (z.B. Jenkins) einbinden.



## Sakuli im Detail

**Zuverlässiges End-to-End Monitoring. Plattformunabhängig und universell.**



Beispiel für die Visualisierung der gemessenen Laufzeiten mit PNP4Nagios

## Sakuli ist mehr als die Summe seiner Teile

Mit **Sakuli** lassen sich die etablierten Open-Source-Tools Sahi und Sikuli über eine einfache, an JavaScript angelehnte Step-for-Step-Sprache so bedienen, als wären sie ein einziges Testing-Tool.

- **Sahi** simuliert die Bedienung von Webseiten, indem es Objekte wie Links, Buttons etc. im Browser nach dem Document-Object-Modell (DOM) identifiziert und bedient.
- **Sikuli** erkennt anhand von Bildmustern Elemente in beliebigen Desktop-Applikationen und simuliert darauf Nutzeraktionen durch automatisches Bedienen per Maus und Tastatur.

**Sakuli vereint die Funktionen der beiden Tools** für automatisierte Applikations-Tests. Stößt eines der Programme an seine Grenzen, kann das andere zum Einsatz kommen, so dass alle Bildschirminhalte unter die Lupe genommen werden.

**Das Ergebnis:** Universelle und zuverlässige Anwendungsüberwachung für jeden Use case.

## Sakuli in der Praxis: Einsatzbeispiele

### Unternehmensweite Performance von Enterprise-Applikationen

Um die Reaktionszeiten z.B. von Citrix-Applikationen an allen Standorten zu messen, loggt sich Sakuli kontinuierlich dort ein. Überschreiten Checks oder deren Teilschritte die vorgegebenen Laufzeiten, so schlägt das Monitoring-System Alarm.

### Funktionsgarantie im Online-Shop

Sakuli loggt sich als Test-User im Shop ein und prüft den Bestellvorgang von der Artikelsuche über den Warenkorb bis zur Bezahlung. Probleme können frühzeitig erkannt und teure Ausfälle vermieden werden.

### Einwandfreie SaaS-Dienstleistungen

Sakuli prüft die Verfügbarkeit und Funktion von Services externer Anbieter, bei denen es an Monitoring-Schnittstellen mangelt. Die Qualität der SaaS-Dienstleistungen wird damit messbar.