Selecting a Selenium Grid Infrastructure in an enterprise

Whitepaper

SELENIUM GRID

CONTINUOUS TESTING INFRASTRUCTURE

Homegrown, SaaS

or managed on-premises ?

Test automation is integral to achieving high quality and fast feedback cycles in an organization. It is also one of the main pillars for moving your organization to a CI/CD/DevOps and Shift Left approach.

Selenium Grid is the common approach for a Selenium cross browser testing infrastructure. Selenium Grid allows you to run tests on multiple browser / OS combinations in parallel and helps to reduce test suite execution times significantly.

There are three main options for setting up a Selenium Grid infrastructure:

- On-premises homegrown
- SaaS / Cloud
- On-premises vendor managed

This white paper is intended to give guidance for determining a suitable Selenium Grid Infrastructure solution for your enterprise organization. We will look at the currently available Selenium Grid solutions in the market and compare pros and cons.

Selecting a Selenium Grid Infrastructure in an enterprise

Whitepaper

Table of Contents

1 BACKGROUND 3

2 WHY DO YOU NEED A SELENIUM GRID IN YOUR ENTERPRISE? 3

2.1 What does Selenium Grid do? 4

2.2 Why is it important to decouple the test authoring side from the infrastructure side? 4

3 SELENIUM GRID SOLUTIONS 5

3.1 On-premises / private cloud vs. public cloud / Saas? 5

3.1.1 Security requirements 5

3.1.2 Performance 6

3.1.3 Integration into your existing infrastructure 6

3.2 Solution 1: On-premises, homegrown and self-maintained 6

3.3 Solution 2: Cloud / SaaS 7

3.3.1 Performance 8

3.3.2 Scaling 8

3.4 Solution 3: On-premises managed Selenium Grid 8

3.4.1 Security 8

3.4.2 Integration 8

3.4.3 Maintenance 9

3.4.4 Performance 9

3.4.5 Integration 9

3.4.6 Scalability 9

3.4.7 Support / SLA 9

3.4.8 Value add features 9

4 Cost 10

5 OTHER FACTORS 10

5.1 Productivity 10

5.2 Business continuity 10

6 CONCLUSION 11

Selecting a Selenium Grid Infrastructure in an enterprise

Whitepaper

1 BACKGROUND

In June 2018 the WebDriver protocol became a W3C standard. As a result, Selenium is now the standard for automating web (and mobile applications).

The WebDriver protocol defines a standard way how users can interact with a browser via an API.

A typical Selenium setup consist of two main components:

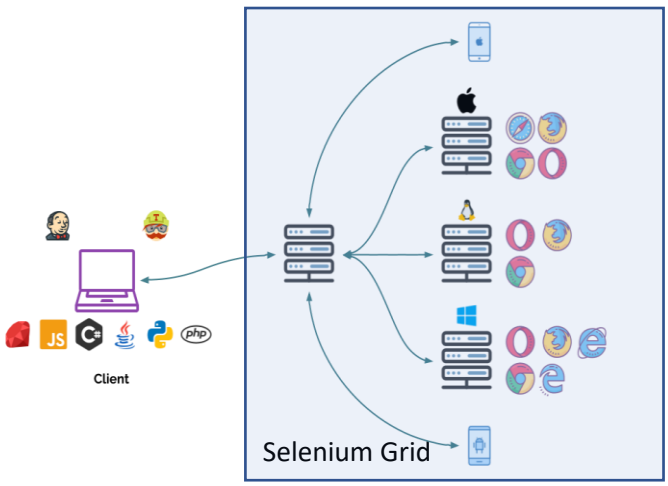
1. **Client side**

This is where the test framework resides, where tests are authored and where tests are run from (i.e. developer workstation, CI system).

2. **Browser / Mobile execution infrastructure**

The test execution infrastructure (Selenium Grid) receives the Selenium commands and translates them into browser and/or mobile specific actions.

This white paper focuses on the browser and mobile execution infrastructure resembled by a Selenium Grid.



Selenium is a W3C standard

"As of June 2018, Selenium / WebDriver is officially a W3C standard for browser automation"

Central infrastructure

"Selenium Grid is a central browser and mobile infrastructure for running automated tests"

2 WHY DO YOU NEED A SELENIUM GRID IN YOUR ENTERPRISE?

A stable, easy to maintain, secure and scalable test execution infrastructure is essential to an enterprise. In an organization, there are typically many sources of tests with different tools in play. In the Web / Mobile field most of those tools converge to use Selenium / Appium as the base protocol to control the browsers and mobile devices.

In a typical enterprise, Selenium based tests often originate from the following of sources:

- Developers using Selenium language bindings (i.e. Java, JS, C#)
- Protractor for Angular Applications
- Cucumber / BDD Frameworks
- Frameworks like TestCafe
- Commercial tools like Tricentis Tosca, Ranorex etc.

Choose between homegrown, cloud and managed on-premises Selenium Grid solutions.

2.1 What does Selenium Grid do?

Selenium Grid can be seen as a central browser and mobile infrastructure, which

- provides the correct browser / Selenium / driver combination when a test is executed on the infrastructure
- load balances and routes tests coming from the client side (i.e. CI system, developer workstation, commercial tools)
- manages browser resources
- provides reliability and browser crash recovery

2.2 Why is it important to decouple the test authoring side from the infrastructure side?

When teams start out, tests are typically executed on the same machine where they are authored. For example, a developer creates test scripts on his/her machine and then executes the scripts on the same machine. While this is sufficient for local debugging purposes, this approach has some fundamental drawbacks:

- The machine is occupied for the duration of the test run
- Only the browsers which are installed on that machine are available for running the tests
- Tests can only be executed in sequence or with low scalability which results in long test execution times
- If you want to run the tests from a CI system, the browsers need to be installed and maintained on the CI system as well.

Make sure to understand the security, scaling, performance and integration implications of each solution.

With Selenium Grid, the test authoring side gets decoupled from the test infrastructure side. Selenium Grid acts as the central test execution infrastructure where all the different test sources connect to and execute their tests.

3 SELENIUM GRID SOLUTIONS

There are different approaches for setting up and running a Selenium Grid infrastructure in your enterprise. In this paper we will focus on the following 3 main approaches:

- Homegrown and self-maintained (*homegrown**)
- Cloud / SaaS (*cloud**)
- On-premises vendor managed (*Selenium Box**)

* term used for this white paper

3.1 On-premises / private cloud vs. public cloud / Saas?

Enterprises first need to determine whether to use an on-premises solution or a cloud solution. There are numerous players in the public cloud / Saas market such as Saucelabs and Browserstack. For many organizations using a cloud service is a good approach, as these providers offer a large range of browser / OS combinations and there is no maintenance required on the infrastructure side.

The following three sections should help you to get a first indication as to whether you should stay on-premises or go to a SaaS provider.

3.1.1 Security requirements

Many enterprises have heavy restrictions and regulations when it comes to test data. For these organizations it is often not possible to use cloud services. Keep in mind the following aspects when using a cloud service:

- you send your entire test / test data to the provider
- the provider needs external access into your infrastructure
- the test is executed in the provider's browser infrastructure
- the provider typically retains logs, screenshots, videos of your tests

3.1.2 Performance

Let's look at how Selenium works: every command requires a roundtrip from the client to the Selenium Grid, from there on to your application under test and back.

Most enterprises already have their CI system, source code repository and test infrastructure close together. Adding a Selenium Grid which is outside of this environment typically adds latencies and can dramatically increase test execution times and stability.

Homegrown solutions are expensive to maintain and the required know-how often lies outside of the organization's core competencies

3.1.3 Integration into your existing infrastructure

Selenium Grid is at the heart of your DevOps pipeline. Thus, it is essential that the chosen Selenium Grid solution easily integrates with your current infrastructure and tools, i.e. LDAP integration. The ability to manage access rights and roles for your Selenium Grid as for your other tools makes the handling of your test infrastructure much simpler.

SaaS / cloud providers offer a wide range of browser / OS combinations, but may not be aligned with your organization's security requirements.

Based on the above considerations you should have come to a first conclusion as to whether you can go to a Cloud / SaaS provider or whether you should run your Selenium Grid infrastructure on premises.

Let's now look at the various options for setting up a Selenium Grid.

3.2 Solution 1: On-premises, homegrown and self-maintained

Building your own Selenium Grid is the starting point for many organizations. Selenium Grid is integrated into the open source distribution of Selenium and it is fairly simple to get a first demo up and running.

After the initial demo and evaluation, many organizations continue to use their engineering staff to build up the enterprise Selenium Grid environment. However, building and maintaining an enterprise grade Selenium Grid requires much more effort than a simple proof of concept.

Deep Selenium know-how, infrastructure, virtualization and system admin skills are required for developing a secure, scalable, consistently up-to-date and cost efficient Selenium Grid infrastructure. This know-how often lies outside a company's core

competencies and internal resources could be more efficiently used for core IT purposes.

One of the major obstacles with a *homegrown* solution is maintaining and scaling the infrastructure to growing needs. Such an effort is usually complex and resource intense.

Another major challenge is the maintenance of the various OS / browser combinations. With the frequent releases of new browser versions and the further development of Selenium and drivers, a team of engineers is necessary to keep the Selenium Grid up and running. Even a small mismatch between browser / driver / Selenium can lead to unpredictable functionality of the browser – leading to false positive or negative tests.

Cloud providers require external access to your infrastructure.

3.3 Solution 2: Cloud / SaaS

There are several vendors offering Selenium Grid as a SaaS solution. When using such a cloud service, the provider takes care of all the time consuming development and maintenance of the Selenium Grid infrastructure. Cloud providers boast a large range of browser / OS / combinations and in some cases support mobile testing.

Depending on the location of the cloud provider's data center, performance may be slow due to latency.

Depending on the industry that you're in, cloud providers can be a great fit. They offer various packages from entry level pricing to enterprise packages.

However, the larger the organization grows, the more difficult the use of a cloud provider becomes – first and foremost because of **security reasons.**

The use of a cloud service requires external access to your enterprise's infrastructure. You need to access the cloud provider's infrastructure. As well the browsers running in the provider's infrastructure need to access your application under test.

Large organizations as well as enterprises in the banking, insurance and medical field often have strong limitations in giving external access to their systems – making it impossible in many cases to use cloud providers.

With cloud based solutions, operations and maintenance lie outside of your organization, resulting in a lack of control over: infrastructure, security, performance, resource sharing, monitoring etc.

3.3.1 Performance

In many cases, cloud providers have a central data center. Depending on the location of the cloud provider's infrastructure and the distance to the customer's environment (CI, application under test), network latency can have an impact on the performance of tests. Each Selenium command requires a full roundtrip from the client to the Selenium Grid then to the application under test and back. If the customer infrastructure and the servers of the cloud provider are geographically far apart the network latency can drastically increase the test execution time.

3.3.2 Scaling

The price for cloud solutions often scale linear with the number of tests and the concurrency of tests. Enterprise subscriptions are also offered. However even with enterprise licenses, cloud based solutions often have limitations with regards to scalability and parallel / concurrent test runs due to the tunnels required.

These limitations should be taken into account and clarified with the vendors, as they can impact the overall goal for fast feedback cycles.

3.4 Solution 3: On-premises managed Selenium Grid

In 2016, Element34 Solutions introduced an on-premises enterprise Selenium Grid infrastructure solution (Selenium Box, https://seleniumbox.com). This solution is fully managed and overcomes all the shortcomings of a homegrown self-maintained solution and cloud solutions.

Let's look at the various aspects of an on-premises Selenium Grid solution to understand why the combination of an on-premises and fully managed solution is in many cases exactly what a large and security sensitive enterprise requires.

3.4.1 Security

Selenium Box runs completely on-premises or within your private cloud. No external access is required and no data or other information ever leaves your network.

Selenium Box guarantees security and eliminates privacy risks that are associated with cloud based solutions.

3.4.2 Integration

By running in the same network / infrastructure, the setup and integration of Selenium Box is straight forward. As a result,

A managed, on-premises solution takes the functionality and ease of use of the cloud providers to your corporate infrastructure.

No data leaves your network and no external network access is required.

Tests can run up to 10 times faster than on the cloud.

connecting with the CI, reporting and monitoring systems and the application under test is simple.

You can also easily integrate Selenium Box into your existing project workflow. A comprehensive API allows integration with the tools of your choice.

3.4.3 Maintenance

Selenium Box is fully managed and we as vendor ensure that the OS / browser / driver / Selenium combinations are always compatible with each other. Customers can be sure that the browsers which are released to the system are fully functioning and do not lead to unreliable tests. No further maintenance is required and the customer can focus their engineering staff on the core competencies of the business.

3.4.4 Performance

The proximity of the different systems in a CI pipeline to each other is key to fast and reliable tests. Selenium Box eliminates the latency issues of cloud solutions. In many cases the execution times of a test suite is up to 10 times faster with an on-premises solution compared to a Selenium Grid in the cloud.

3.4.5 Integration

Because Selenium Grid is at the heart of your DevOps and test infrastructure, it is vital that it easily integrates with your current infrastructure and tools. By running an on-premises solution like Selenium Box, you can i.e. hook into your corporate identity provider mechanism and manage the access rights and roles the same way as for your existing tools.

3.4.6 Scalability

As more and more tests are run within your organization, the need for scaling becomes increasingly important. Selenium Box allows customers to scale to their needs while being in full control of the cost. With Selenium Box, customers can simply add more computing power to the system which in turn allows for more tests to be run in parallel.

3.4.7 Support / SLA

Compared to a homegrown solution, a Selenium Box comes with enterprise level support and SLAs for issue resolution. Homegrown solutions in many cases are managed "on the side" by development teams and cannot ensure uptime, reliability and consistency.

3.4.8 Value add features

Selenium Box comes with many value add features like live view, video recording, monitoring & logging capabilities, access control and

sophisticated error recovery. *Homegrown* solutions lack value added features since they use the out-of-the-box Selenium Grid which is provided on http://seleniumhq.org. Therefore, teams are usually stuck with a barebones Selenium Grid that does not allow for easy troubleshooting, debugging and reporting. The use of a *homegrown* Selenium Grid creates the risk of a lower adoption of automated testing than with a feature-rich managed Selenium Grid solution.

4 Cost

Cost is an important factor in the selection process of a Selenium Grid solution. Let's be clear: even though Selenium is open source and does not incur any license fees, it still requires an investment to use Selenium.

Building a *homegrown* Selenium Grid solution typically takes several person years. Also the ongoing maintenance of such a solution usually requires a handful of expert engineers / operations staff.

Cloud solutions do not require in-house staff but incur license fees. For a larger enterprise, fees usually run in the lower to mid 6-digit (USD) space. As well, the integration of the cloud service into the enterprise infrastructure (if possible at all) can be a significant cost factor.

Selenium Box also incurs a license fee plus the hardware / infrastructure cost for running the system.

5 OTHER FACTORS

5.1 Productivity

Engineering resources are scarce these days and enterprises need to focus their staff activities on their core business. Building and maintaining a *homegrown* solution can be compared with rebuilding Microsoft Office. While it *could* be done, it adds no value to the enterprise, especially as there are commercial solutions in the market.

With a managed solution (cloud or on-premises), teams can fully focus on the task of writing meaningful tests rather than building and maintaining test infrastructure and / or working with hard to use and unstable solutions.

5.2 Business continuity

To implement and maintain a *homegrown* Selenium Grid a group of experts is required. In case of a change within the team (i.e.

reorganization, attrition), it can be difficult to fill these positions in a timely manner – potentially slowing down or stopping your test automation and CI/CD efforts all together.

6 CONCLUSION

A secure, reliable and scalable Selenium Grid cross browser infrastructure is crucial for a successful continuous testing and DevOps setup. Setting up and maintaining a mature enterprise grade Selenium Grid infrastructure requires a significant investment and is a complex undertaking.

While a *homegrown* solution can be a good starting point, for most enterprises they quickly become unmanageable. *Homegrown* solutions require a big up-front investment in terms of engineering resources as well as deep Selenium Grid know-how. Maintenance for a *homegrown* Selenium Grid is time consuming and prone to error due to frequent new release of browsers and the Selenium ecosystem.

Homegrown solutions also lack any comfort functions for easy debugging and troubleshooting.

When moving away from *homegrown* solutions, there are two choices:

- Cloud / SaaS (i.e. Saucelabs, Browserstack)
- On-premises and managed (i.e. Selenium Box)

While a *cloud* solution may be a good fit for some organizations, larger enterprises (especially in the financial field) require greater security measures and cannot always use a cloud service.

A managed on-premises solution may often be the best option. Solutions like ***Selenium Box*** provide maintenance and worry free operation of the Selenium Grid and unlimited scalability options.

A *vendor managed* on-premises solution may therefore be the best suited solution for all security sensitive organizations or for those who have high scaling needs.

Selecting a Selenium Grid Infrastructure in an enterprise

Whitepaper

Element34 Solutions GmbH
Teichweg 8
CH-8853 Lachen
Switzerland

Contact & information: contact@element34.com
https://seleniumbox.com