SELENIUM GRID - CONTINUOUS TESTING INFRASTRUCTURE

A guide to finding the Selenium Grid solution that best fits your needs.

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 cross browser and mobile testing infrastructure. Selenium Grid allows you to run tests on multiple browser / device / 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 in the market today:

- Inside your firewall: Homegrown / Open Source based
- SaaS / Cloud
- Inside your firewall: 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.

Table of Contents

1. BACKGROUND

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

Selenium provides a rich ecosystem in the open source as well the vendor supported space. Many commercial tools also support the WebDriver protocol, which makes them easy to integrate with any Selenium based ecosystem.

A typical test automation setup consist of the following components:

- Test authoring tools and platforms (left)
- Test execution infrastructure (middle)
- Applications under test (right)

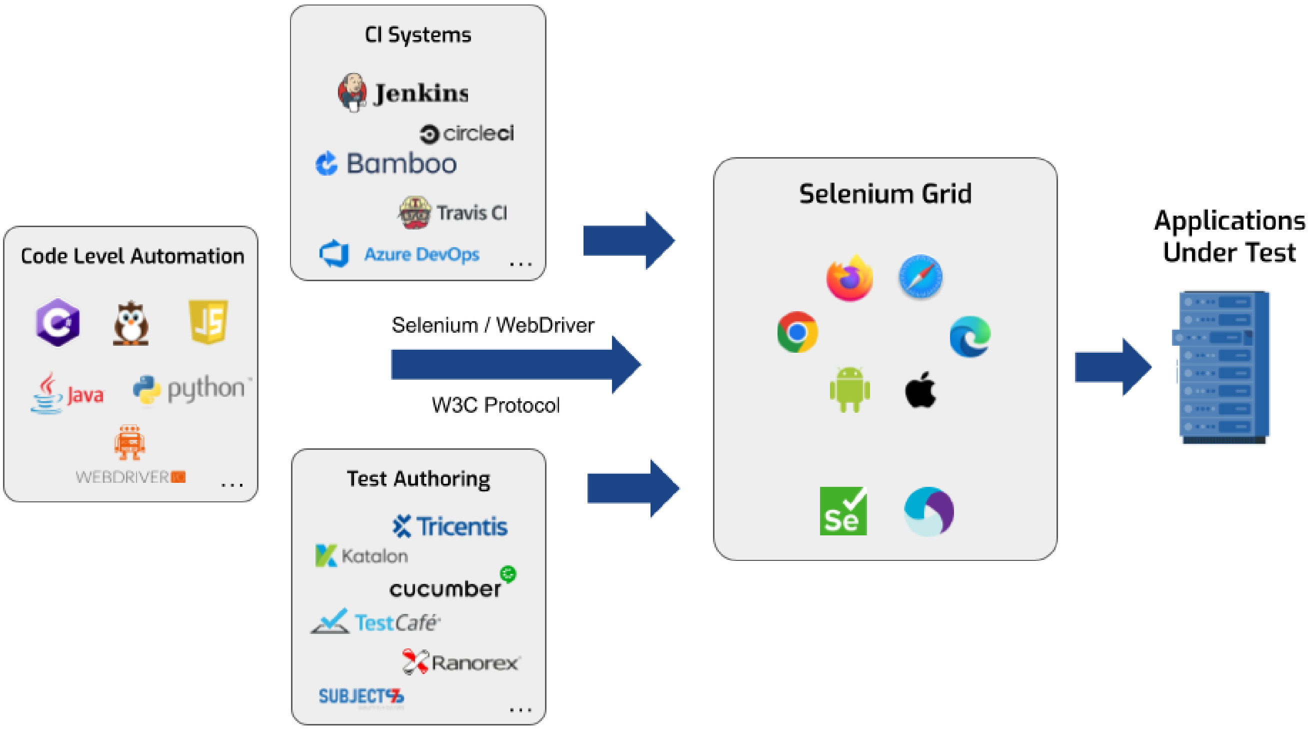

Fig 1. - A typical test automation setup

This whitepaper focuses on the test execution infrastructure resembled by a Selenium Grid.

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

A stable, easy to maintain, secure and scalable test execution infrastructure is essential for 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.

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;
- Provides monitoring and reporting capabilities.

2.2. The importance of decoupling the authoring from the test execution infrastructure

When teams start out, tests are typically executed on the same machine where they are authored. For example, developers create test scripts on their machine and then execute the scripts on the same machine. While this may be 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.

With Selenium Grid, test authoring gets decoupled from the test infrastructure. 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.

3.1. A word on homegrown solutions

In the scope of this whitepaper we will not focus on homegrown solutions as they are not comparable with SaaS/external and managed/internal solutions.

The major drawbacks around homegrown solutions are the following:

Enterprise Features
Homegrown solutions almost always lack the enterprise level features and integrations of SaaS and managed internal solutions (e.g. video recording, monitoring, reporting, integration in Active Directory, etc).

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 company, especially as there are commercial solutions readily available 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 working with hard to use and unstable solutions.

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 extremely difficult to fill these positions in a timely manner, and to perform adequate knowledge transfer – this will potentially slow down or stop your test automation and CI/CD efforts all together.

3.2. Inside or outside your firewall?

Enterprises first need to determine whether to use a solution that runs inside their corporate firewall or a SaaS solution which runs outside the firewall. There are numerous players in the SaaS market such as Saucelabs and Browserstack. For many organizations, using a SaaS 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.

There are 3 main differentiators between solutions running inside vs. outside the corporate firewall:

- Security and External Access;
- Data privacy;
- Performance;
- Integration.

In the next sections we'll look at these in more detail.

3.2.1. Security / external access

Many enterprises have heavy restrictions on providing external access to their systems. If you use a SaaS provider, external access needs to be granted to the SaaS provider (e.g. by IP whitelisting, tunneling).

On the contrary, internal solutions run on infrastructure that is inside the corporate firewall and hence do not require external access.

3.2.2. Data Privacy

Data privacy laws around the globe have been tightened in the past years and severe fines are imposed if they are breached (GDPR, CCPS, etc).

Running tests on browsers requires the use of test data that is input into the user interface. Also, data is visible in the browsers in clear text e.g. after logging into a bank account, the account balance is displayed.

When using a SaaS solution you need to keep in mind that:

- The test data is sent to the SaaS provider which is then input in the browser in the SaaS provider's infrastructure;
- All resulting data (e.g. account balance, health records) is visible to everyone at the SaaS provider with the respective access;
- all logs, screenshots, videos are kept at the SaaS provider for a period of time,

Furthermore you typically grant rights in the data to the SaaS provider to "use, copy, store, transmit, modify, create derivative works of and display the data".

All of the above may be less of an issue if you use 100% synthetically generated test data. In reality however, most enterprises use a mix of production, production-like, anonymized and pseudonymized data for test purposes.

To be compliant and avoid any type of data breaches, the use of such data requires the data to remain inside the corporate firewall. Hence the use of a SaaS provider with production type test data is prohibitive and will most certainly violate a wide range of data privacy regulations.

3.2.3. 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 in close proximity. Adding a Selenium Grid which is outside of such an environment adds latency and can dramatically increase test execution times and stability.

3.2.4. Integration into your existing infrastructure

With Selenium Grid at the heart of your DevOps pipeline, it is essential that the solution easily integrates with your current infrastructure and tools, e.g. LDAP for access control and user management.

Granting a SaaS provider access to your corporate LDAP system may prove difficult or be simply impossible.

3.2.5. How to decide whether to use a SaaS or not?

It's actually not that hard! You can use this simple decision graph to determine if using a SaaS solution is an option in your enterprise.

While there are additional decision factors such as performance and integration capabilities, those are typically less critical than the security and data privacy aspects.

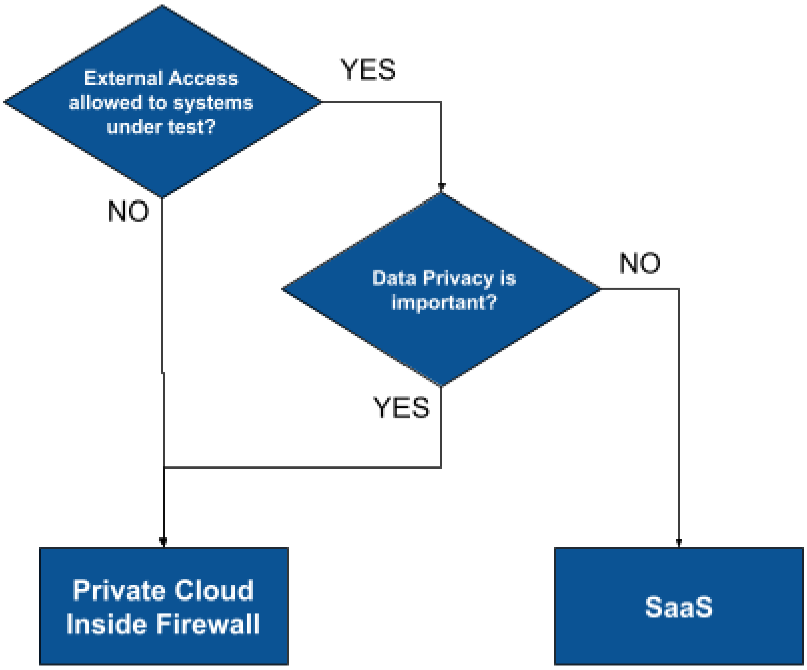
External Access allowed to systems under test?
YES
NO
Data Privacy is important?
NO
YES
Private Cloud Inside Firewall
SaaS

Fig.2 - A simple decision model for determining the right test execution approach

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

4. SaaS solutions

There are several vendors offering Selenium Grid as a SaaS solution. When using such a SaaS service, the provider takes care of all the time consuming development and maintenance of the Selenium Grid infrastructure.

SaaS providers boast a large range of browser/OS combinations and in some cases support mobile testing.

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, the more difficult the use of a SaaS provider becomes – first and foremost because of the abovementioned security and compliance reasons.

As mentioned before, using a SaaS service requires:

- The SaaS provider to access your enterprise's infrastructure;
- Data to be transmitted and shown to the SaaS provider.

Large organizations as well as enterprises in highly regulated industries like banking and financial services, insurance, biotech and healthcare typically have strong limitations in giving external access to their systems. Additionally, they are subject to data protection and data privacy laws that prohibit transmitting data to 3rd parties, including to SaaS solutions.

Generally it can be said that SaaS solutions provide a low entry barrier to get started, while operations and maintenance of the infrastructure lie outside of your organization. This results in a lack of control over infrastructure, security, usage of data, performance, resource sharing, monitoring etc.

4.1. Performance

In many cases, SaaS 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 a considerable impact on the performance of the 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 and in some cases even have an impact on its result.

4.2. Scaling

The price for cloud solutions often scale linearly with the number of tests and the concurrency of tests. Enterprise subscriptions are typically 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.

5. Managed Selenium Grid inside your corporate firewall

In 2016, Element34 introduced an Enterprise Selenium Grid solution called SBOX, that operates 100% within your corporate firewall. This solution is fully managed and overcomes all security risks and shortcomings of SaaS solutions.

Let's look at the various aspects of a managed, but yet internal, Selenium Grid solution to understand why the combination of an "inside your firewall" and fully managed solution is, in many cases, exactly what a large and security-sensitive enterprise needs.

5.1. Security

SBOX runs inside your corporate network. No external access is required and no data or other information ever leaves your network.

SBOX guarantees security and eliminates data privacy risks that are associated with SaaS based solutions.

5.2. Integration

By running in the same network/infrastructure, the setup and integration of SBOX is straightforward. As a result, connecting with the CI, reporting and monitoring systems and the applications under test is simple.

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

5.3. Maintenance

SBOX is fully managed and therefore all 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.

5.4. Performance

The proximity of the different systems to each other in a CI pipeline is key to fast and reliable tests. SBOX eliminates the latency issues found in SaaS solutions. In many cases the execution times of a test suite are up to 10 times faster with this type of solution when compared with SaaS alternatives.

5.5. Integration

Since 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-premise solution like SBOX you can, for example, hook into your corporate identity provider mechanism and manage the access rights and roles the same way as you would for your existing tools.

5.6. Scalability

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

5.7. Support / SLA

Compared to a homegrown solution, SBOX comes with enterprise level support and SLAs for issue resolution. Homegrown solutions in many cases are managed in “best effort” by development teams and cannot ensure uptime, reliability and consistency.

6. Conclusion

A secure, reliable and scalable Selenium Grid cross-browser infrastructure is crucial for a successful continuous testing and DevOps setup. Apart from setting up a homegrown infrastructure yourself, you have the choice between SaaS solutions operating outside your corporate firewall and vendor managed solutions operating inside your corporate firewall.

While a SaaS solution may be a good fit for some organizations, larger enterprises especially in highly regulated industries require greater security measures and may not be able to use a SaaS service at all.

A managed solution running inside your corporate firewall is in many cases the best option. Solutions like SBOX provide a secure and maintenance free operation of the Selenium Grid with unlimited scalability options.

For additional information visit www.element34.com or contact us at contact@element34.com