

# Considerations for selecting a Selenium Grid solution

---

A comparison between cloud based and in-house Selenium Grid solutions

Michael Palotas / Francois Reynaud  
Element34 Solutions GmbH

## TABLE OF CONTENTS

1. Background .....	3
2. Approaches to Selenium infrastructure .....	3
3. How <i>Selenium Grid</i> works .....	4
4. Deciding on an approach.....	4
4.1. General <i>Selenium Grid</i> architecture	4
4.2. Cloud vs. in-house .....	4
4.2.1. Build / self-operated vs. buy / managed Selenium Grid	5
4.3. Maintenance .....	5
4.4. Scalability.....	6
4.5. Cost / Pricing .....	6
4.6. Security and privacy .....	7
4.7. Performance .....	7
4.8. Reporting and monitoring .....	7
4.9. Corporate operating system images	8
4.10. Selenium API modifications.....	8
4.11. Web and mobile support.....	8
4.12. Auto-scaling abilities .....	8
5. Summary .....	10
6. About Element34 Solutions GmbH.....	11
7. Selenium Box at a glance.....	12
What is Selenium Box? .....	12
Who is Selenium Box for? .....	12
Selenium Box highlights .....	12
Why Selenium Box?.....	12

**This white paper is intended to give guidance for determining the right Selenium Grid solution and setup for your organization. It will compare cloud based Selenium Grid solutions with in-house solutions. For in-house solutions we will differentiate between self-built vs. commercial, managed solutions like the *Selenium Box* by Element34 Solutions GmbH.**

## 1. Background

According to the Gartner Magic Quadrant for software test automation (Gartner, December 2015), Selenium will become the de-facto standard by 2020 for test automation. From our experience, we would argue that today Selenium is already the standard for automating web and mobile applications. Selenium's JSON wire protocol is becoming a W3C standard and browser manufacturers like Google, Mozilla and Microsoft are already natively supporting the protocol within their browsers.

Many organizations have started to build up substantial test suites with Selenium. While Selenium allows for a relatively easy way to write tests, there is often a lack of a "running environment" where those tests can be run in a reliable and scalable fashion. With many organizations moving closer to continuous integration, it is essential that automated tests (unit, integration and GUI) can be run on demand or triggered by a CI system and provide fast feedback cycles (<10 minutes).

Organizations have approached the (sometimes missing) running environment in various ways. In many cases Selenium tests are still executed on the developer workstation with the locally installed

browser, locally on the CI agent or not integrated into the CI system at all.

In order to execute a larger number of Selenium tests within a short feedback cycle, running the tests in parallel is possible with Selenium's built in *Grid* component.

## 2. Approaches to Selenium infrastructure

Setting up a *Selenium Grid* for demo / proof of concept purposes is relatively simple. On the contrary, the setup and maintenance of an enterprise scale *Selenium Grid* is a highly complex and time consuming task where the following aspects should be taken into consideration.

- Cloud vs. in-house
- Build vs. buy
- Managed vs. self-maintained
- Pricing
- Scalability
- Performance
- Reporting and monitoring capabilities
- Mobile support
- Use of corporate operating system images
- Selenium API modifications

### 3. How Selenium Grid works

Before going into the details, let's review the functionality of the *Selenium Grid*.

The *Selenium Grid* has three main purposes:

1. Load balancing and routing of tests coming from the client side (i.e. CI system)
2. Resource management of the browsers
3. Reliability and browser crash recovery

A typical Selenium Grid setup in many organizations is as follows.

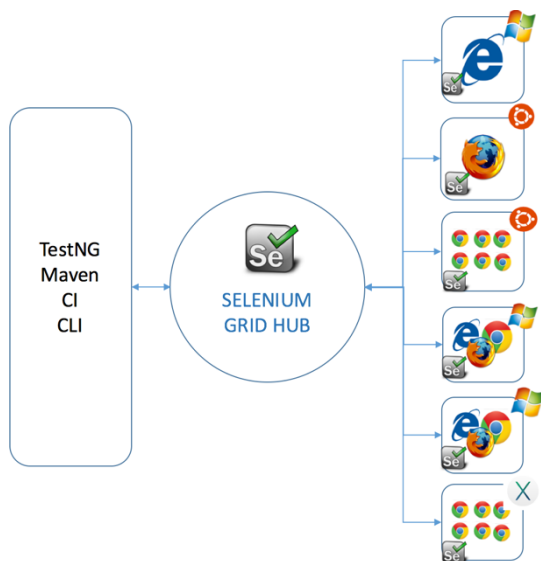


Figure 1 - Sample Selenium Grid

The browser nodes on the right side register with the Selenium Grid Hub and advertise their respective “capabilities”. When tests are sent to the Grid hub from the client (left) side they include “desired capabilities” which specify i.e. the browser and operating system version for the test.

The Grid hub matches the desired capabilities with the available capabilities

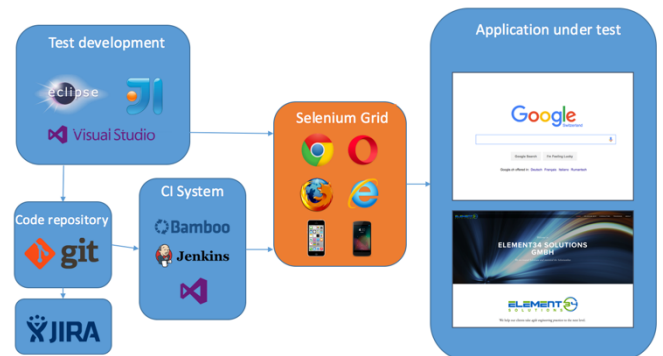
of the registered nodes. If there is a match, the test is forwarded to the respective browser on the node. (There are a few more things happening on the Grid hub of course, but for the purpose of this paper we won't go into more details.)

### 4. Deciding on an approach

#### 4.1. General Selenium Grid architecture

Most organizations host the test servers, CI system, source code repository, bug tracking system and ticketing system in-house. The *Selenium Grid* is deeply embedded and connected to those systems and should also remain in-house.

Figure 2 illustrates the interconnectivity between the various components and tools including the Selenium Grid within



the IT infrastructure.

Figure 2 - A view into the entire system

#### 4.2. Cloud vs. in-house

An in-house Selenium Grid gives full flexibility and control to the organization. As well, arising issues can be addressed directly rather than waiting for a resolution from a third party.

On the other hand, there are numerous cloud based providers offering a *Selenium Grid* as a service. They usually boast a large range of browser / OS / combinations and in some cases support mobile testing. When using a cloud based *Selenium Grid*, organizations do not need to worry about operating and maintaining the *Selenium Grid* infrastructure. This is all done by the cloud provider.

In many cases though, the security and access policies of organizations prevent them from using cloud based Selenium Grids. This is due to the required external access by the Selenium Grid to the application under test.

With cloud based solutions, operations and maintenance lie outside of the organization, resulting in a lack of control over how infrastructure is built, security, performance aspects, resource sharing, monitoring options, etc.

◀◀ **For organisations who have security/privacy concerns or require more control over their infrastructure, an in-house Selenium Grid is a better option over a cloud based solution.**

#### 4.2.1. Build / self-operated vs. buy / managed Selenium Grid

It may appear relatively simple to set up a *Selenium Grid* on your own. It is a matter of downloading the selenium-server-standalone.jar from <http://seleniumhq.org>, starting the server in *-hub* mode and registering the respective nodes with the hub. But that's not the whole story.

Many organizations try *Selenium Grid* as a proof of concept and then use their

engineering staff to build up the enterprise *Selenium Grid* environment. While under certain circumstances this can be a feasible approach, the involved parties soon realize that building up an enterprise grade *Selenium Grid* is very different from running a simple "hello world" like proof of concept.

Selenium specific know-how is required for building up a secure, scalable, consistently up-to-date and cost efficient *Selenium Grid* infrastructure. This know-how is often outside a company's core competencies and internal resources could be more efficiently used for core purposes.

◀◀ **Setting up an in-house Selenium Grid on your own is complex and usually lies outside of the core competencies of most companies.**

#### 4.3. Maintenance

New browser, driver versions and Selenium versions are released about every 2 weeks. As a result, the Grid needs to be updated with the relevant browsers and drivers every 2-3 days on average. In case of a new Selenium version, the new server has to be deployed to each node.

Managed in-house Selenium Grid solutions (i.e. Selenium Box) are maintenance free and have none of the other restrictions of cloud based solutions. Managed solutions provide the flexibility and security of an in-house solution while eliminating the tedious, time consuming and error prone update procedure.

◀◀ **Managed in-house Selenium Grid solutions are the best choice for eliminating the maintenance effort and ensuring error free configuration.**

For self-built / self-operated Grids the update effort is significant and can be complicated by human error, when for instance an invalid browser / Selenium version combination is deployed on the nodes which results in flaky and unreliable tests.

#### 4.4. Scalability

The ability to scale and running tests in parallel is an essential part of *Selenium Grid*. Scaling is especially relevant for organizations who either have a large number of Selenium tests or will provide a centralized *Selenium Grid* to the development teams. By running tests in parallel, the execution time of the test suites can be reduced significantly. Fast feedback in turn is one of the cornerstones for continuous integration.

In-house solutions generally scale very well and in many cases it is only a matter of supplying more CPU power to the grid in order to add more capacity for running tests in parallel.

❖ **To keep execution time to a minimum, scaling and running tests in parallel is necessary. In-house solutions offer better scaling capabilities, whereas most cloud based solutions have many limitations.**

Even with enterprise licenses / accounts, cloud based solutions often have severe limitations with regards to scalability and parallel test runs. In many cases they only allow running a single digit amount of tests in parallel time, which severely limits the ability for a high test throughput in the *Selenium Grid*.

#### 4.5. Cost / Pricing

As IT budgets continue to shrink, selecting a cost efficient *Selenium Grid* solution is vital.

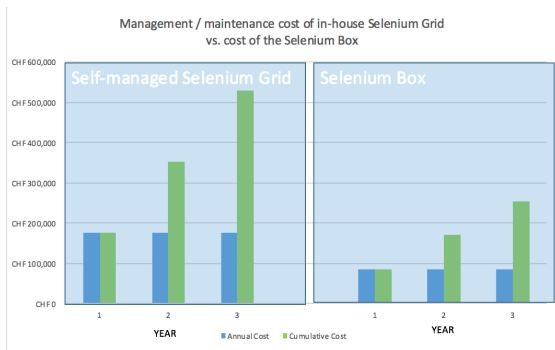
In-house solutions usually come at a higher entry price than an annual enterprise license of a cloud provider. However, they are significantly cheaper to operate and the amortization is in many cases reached within 1-2 years.

Entry packages to cloud solutions often start at an attractive price (< USD 50 / month). They are perfect for users who want to try out this type of service or teams who run a few tests infrequently.

It is however important to understand the pricing model of cloud solutions. The price usually scales linear to the number of tests / test minutes. The more automated tests you run the more expensive the monthly bill becomes.

❖ **In-house managed solutions are generally more cost-efficient and amortize within 1-2 years in comparison to cloud solutions.**

Even for enterprise accounts / licenses which usually run in the area of 25.000-35.000 USD / year, the available minutes and concurrent tests are mostly limited. Unlimited cloud solutions are rare and if they exist, they usually come with a price tag in excess of 100.000 USD / year. With those limitations, cloud solutions often do not sufficiently support the CI strategy of an organization, where running tests continuously is a requirement.



The above chart shows the maintenance cost of an in-house Selenium Grid. For this business case we assumed an hourly rate for an employee of USD 150 (fully loaded). It can be clearly seen as investing in-house resources for maintenance of the Selenium Grid is by far less cost efficient compared to the cost of the Selenium Box.

◀◀ **Managing and maintaining an in-house Selenium Grid is far more expensive than running the Selenium Box on premise.**

Note: the details of the above business case can be shared upon request (contact@element34.net).

#### 4.6. Security and privacy

As seen in section 4.1 General *Selenium Grid* architecture, the *Selenium Grid* is at the heart an organization’s wider test infrastructure. It requires access to the application under test, as well as the client side, which is usually resembled by the CI system or the developer workstations.

◀◀ **In-house solutions are inherently more secure as they do not require external access to the systems.**

An in-house Selenium Grid inherently eliminates security and privacy risks that can be associated with cloud based solutions, as it is run within the organization’s network and does not require any outside access.

Cloud solutions by nature *always* require access from the outside world into the heart of your IT infrastructure. Due to security and privacy policies for many organizations, especially in the financial sector, the use of cloud solutions becomes prohibitive.

#### 4.7. Performance

In-house solutions are operated within the same network as the rest of the test infrastructure. Therefore, time lag is not much of a factor and performance (and with that stability of the tests) is much higher compared to cloud solutions.

◀◀ **Long latencies can cause unreliable tests. This can be avoided by running using low latency / high performance in-house solutions.**

Due to external access from the cloud to the application under test, cloud based systems are generally slower than on premise systems. This can be due to long latencies and the internal architecture of the cloud Grid. In such cases, the user has no influence on the performance of the cloud based Grid. Time lag and unpredictable delays pose the risk of flaky and unreliable tests.

#### 4.8. Reporting and monitoring

Human readable test reports are essential for efficient analysis of test results. Managed in-house solutions like the Selenium Box give full flexibility for reporting and monitoring as they come with sophisticated test reports including

screenshots, video and raw (JSON). If needed, organizations can adapt the existing reports or can build custom reports based on the raw report data.

Cloud solutions usually come with pre-built test reports that satisfy the basic needs of organizations, but are generally not customizable.

◀◀ **Ensure that your solution provides a sophisticated and customizable reporting mechanism.**

For self-built in-house grids, reporting needs to be built from scratch as Selenium does not come with a sophisticated reporting mechanism by default. This usually results in an high in-house development effort.

#### 4.9. Corporate operating system images

For an organization that wants to test internal applications, it may be important to use their own images of the operating system (i.e. Windows) along with the browsers that are supported by the application.

In-house solutions allow to install the corporate operating system image along with the supported browsers.

◀◀ **In-house solutions allow organizations to run tests on their corporate operating system image along with your company browser.**

Cloud providers on the other hand, will not allow an organization to install their corporate image into the cloud environment.

#### 4.10. Selenium API modifications

Some cloud offerings use modified / non-standard Selenium APIs which inherently means that the tests are tied to the

vendor of the *Selenium Grid*. If the customer wants to change the vendor, the tests also need to be re-authored in most cases.

#### 4.11. Web and mobile support

Mobile web and native apps are an integral component for most businesses today. Running web and also mobile tests is becoming an essential part of automated testing, especially as users move their activities across devices and platforms.

Selenium natively supports mobile tests and there are several tools which implement the Webdriver protocol such as Appium, Selendroid and ios-driver.

◀◀ **Web and mobile support are an essential feature of your Selenium Grid**

Organizations should ensure mobile tests seamlessly integrate with web tests and can be run through the same *Selenium Grid* instead of building up separate environments for web and mobile. *Selenium Box* fully supports mobile testing through the same infrastructure as the web tests.

#### 4.12. Auto-scaling abilities

In practice the largest part of the *Selenium Grid* infrastructure is idle as only a small subset of the pre-configured OS/browser combinations in a Selenium Grid are usually executed at a time. This leads to a heavy underutilization of the in-house infrastructure where the *Selenium Grid* is operated.

◀◀ **Maximize your resource utilization by using auto-scaling solutions.**



Managed solutions help to better maximize the resource utilization of the Selenium Grid as they do not require a pre-configuration. Instead they automatically scale based on the number of incoming tests. Solutions like the Selenium Box provide such auto scaling capabilities.

## 5. Summary

The below comparison chart visualizes the advantages of an in-house, managed Selenium Grid over self-built and cloud based solutions. Especially from a pricing, maintenance, scaling and security perspective, managed in-house solutions like the Selenium Box are far superior over self-built or cloud based services.

	In-house, managed grid / Selenium Box	In-house grid self-built	Cloud based solution *
Fixed price	✓	!	✗
Unlimited scalability	✓	✗	✗
Unlimited concurrent tests	✓	!	✗
Unlimited testing minutes	✓	✓	✗
Auto-maintenance	✓	✗	✓
Browser crash recovery	✓	!	✓
Secure / no external access required	✓	✓	✗
Highly reliable	✓	!	✓
No vendor lock-in	✓	✓	✗
High performance	✓	!	✗
No Selenium Grid know-how required	✓	✗	✓
No personnel required for operating	✓	✗	✓
Built in test reports	✓	!	✓
Customizable test reports	✓	!	✗
Video reporting	✓	!	✓
System monitoring	✓	!	✓
Web support	✓	✓	✓
Mobile native app support	✓	!	✓
Mobile Web support	✓	!	✓
Full control over Selenium Grid	✓	✓	✗
Use of corporate OS image	✓	✓	✗
Use of unmodified Selenium API	✓	✓	✗
Standalone hardware solution	✓	✗	✗
In-house cloud solution	✓	✓	✗

Table 1 - comparison chart

✓ - Default / built in

! - Significant effort and Selenium know how required for implementation / not default

✗ - Not supported

\*some aspects may not be applicable to certain cloud providers

With in-house solutions, organizations can choose between self-built or buy and managed. Self-built solutions require a big up-front investment in terms of human resources as well as deep Selenium Grid know-how. Maintenance for a self-built Selenium Grid is time consuming and prone to error.

Managed solutions like the *Selenium Box* provide maintenance and worry free operation of the Selenium Grid and provide unlimited scalability options. Scalability should be built into the solution and should not add more cost. If security and privacy are a concern for your organization, in-house solutions are a much better fit than cloud solutions due to the external access requirement and lack of control over the internals of the cloud solution.

## 6. About Element34 Solutions GmbH

Element34 Solutions GmbH is a boutique consulting firm specialized in agile engineering practices and test automation solutions with Selenium. Selenium Grid was developed and open sourced by co-founder Francois Reynaud.

The *Selenium Box* was created and developed by Element34 in order to significantly ease the setup, maintenance and operation for an in-house Selenium Grid.

The company founders Michael Palotas and Francois Reynaud spent more than a decade working for eBay and were instrumental in bringing software development and test engineering at eBay into the agile age.

Both founders are key contributors and thought leaders to open source projects such as Selenium, Selendroid and ios-driver.

More information about the *Selenium Box* can be found at <http://www.element34.net/seleniumbox>

**Element34 Solutions GmbH**  
**Teichweg 8**  
**8853 Lachen**  
**Switzerland**  
**<http://www.element34.net>**  
**[contact@element34.net](mailto:contact@element34.net)**

## 7. Selenium Box at a glance

### What is Selenium Box?

Selenium Box is a secure, on-premise, auto-maintenance, auto-scaling Selenium Grid solution for running cross browser and mobile Selenium tests in-house.

### Who is Selenium Box for?

The Selenium Box is for organizations who require:

- Security and data privacy
- Scalability for running many tests in a short time
- A maintenance-free Selenium Grid (selenium versions, browsers, drivers)
- In-house engineering resources to focus on core competencies rather than building a selenium grid infrastructure.
- Support for all major browsers
- Full control over the Selenium Grid
- Seamless integration into common CI systems

### Selenium Box highlights

- Built in Selenium Grid
- Software or hardware solution
- Auto scaling Selenium Nodes
- Firefox, Chrome, Internet Explorer and Edge
- Android+ iOS
- Automated updates of Grid hub, nodes and browsers
- Self-healing nodes, crash recovery
- Out of the box, customizable reporting: static, video, live view, raw
- Full integration with common CI systems (Jenkins, Bamboo, TFS, TeamCity etc.)
- Advanced system monitoring options
- Use corporate operating system images
- Unmodified Selenium / Webdriver API

### Why Selenium Box?

- On-premise, secure
- Automated updates and maintenance
- Unlimited scalability
- No maintenance required
- Fixed price and simple pricing model
- Full control
- Cross browser and mobile support
- Focus on writing tests instead of building infrastructure
- No vendor lock-in