SOLUTION BRIEF

Enterprise Application Testing with Cloud-Based Device Farms

KEYSIGHT

eggplant
Test Automation Software

In today's world, customers interact with enterprises using mobile applications. Need to make a payment? Order food? Book a flight? There's an app for that. Competition in the app store is fierce. Customers can switch to a competitor by simply removing one application and installing a new one.

Enterprises improve user satisfaction, digital trust, and customer retention by making sure their applications are best in class. Their customers use various devices that run on different operating systems, so testing is vital for application performance.

Using Cloud-Based Mobile Device Farms with Eggplant Test Automation Solution

Testing apps on a wide range of devices is challenging. The device mix changes frequently and maintaining a centralized test lab can be costly. Not only do you have to procure the devices, but you also need to manage remote access and user permissions, keep the devices cool and the batteries charged, and much more.

Integrating Keysight's Eggplant test automation solution with software-as-a-service (SaaS) mobile device farms eliminates these problems. Mobile device farms in the cloud host thousands of devices, allowing enterprises to test their apps on users' actual devices. Using mobile device farms in the cloud, teams can focus on testing apps rather than managing various devices.

By connecting Eggplant software with a mobile device farm, you can design and run mobile app tests on emulators, private devices, and public devices hosted in the cloud. Instead of relying on bolt-ons, Eggplant software has native support for devices hosted in the cloud, so test automation engineers can design and run tests on these devices as if they were connected locally.

Table 1. Comparison of emulators, public devices, and private devices

	Emulator	Public device	Private device
Cost	Low	Medium	High
Flexibility	High	High	Low
Real-world testing	No	Yes	Yes
Shared	No	Yes	No

Unlock the full potential of test automation

Integrating Eggplant software with your SaaS mobile devices unlocks the full potential of test automation to accelerate releases.

Using Eggplant's visual approach to testing, you can test critical human actions to ensure that any visual clues meant for end-user interactions are intuitive. By testing at the user interface layer, Eggplant can spot visual errors that cannot be identified in the code, such as blue text on a blue background or display issues associated with different screen sizes and orientations.

Eggplant's model-based approach lets you create a digital twin of your system, including mobile apps and other digital assets, such as websites. Your users are rarely predictable, so Eggplant can test every possible end-to-end user journey using artificial intelligence (AI) and machine learning (ML) algorithms. It even auto-generates test cases and prioritizes the areas of your app that need testing the most, so you don't have to.

The Eggplant Advantage

Enterprises compete with thousands of apps in app stores. Users install these apps on a wide range of devices. To make sure your app stands out and doesn't have any compatibility issues, you need to test it on various devices.

Manually testing apps on multiple devices is labor-intensive and costly, and it dramatically slows release cycles. Eggplant software helps you automate and accelerate end-to-end testing of your users' digital journeys through your apps.

With support for SaaS mobile device farms, you can run Eggplant tests on any technology overcoming device fragmentation, so you eliminate usability issues before they reach your users. Cloud-based mobile device farms make it easier to use devices in every step of the testing life cycle: automation engineers developing and maintaining automation scripts, end-to-end testing in continuous integration / continuous delivery (CI / CD) pipelines, and large testing campaigns.

Using Eggplant software, testers can expand test coverage incrementally to parts of an application with known issues via intelligent automated exploratory testing. The AI and ML algorithms proactively explore areas close to these defects, as the probability of other bugs is high. Missed defects are costly to fix, especially when discovered in production. Eggplant's ability to increase test coverage helps reduce the chance of a buggy user experience, which can cause brand damage, disappoint end users, and hurt an organization's bottom line.

Ensure that your mobile apps work as expected by automating testing with Eggplant software. Run tests frequently on a wide range of devices and accelerate release cycles to exceed your end users' expectations.

To learn how to revolutionize your app testing, get to market faster, and improve end-user satisfaction, contact Keysight's Eggplant team today.

Competition in the app store is fierce, and enterprises want to ensure their applications work well for all customers.

Eggplant integrates with cloud-based mobile device farms, allowing testing of apps on a broad mix of devices. These devices are available in the cloud, removing the need to set up and manage on-premise device farms.

Questions to Consider

When you start using a SaaS mobile device farm, keep the following questions in mind.

What's your end-user device mix?

Ensure that testing focuses on the devices your customers use. Use telemetry from your application or metrics available from the app store. Build a list of devices that covers 80% of your user base. Find out if testing on the latest versions of Android and iOS operating systems gives you the right coverage or if you need to include previous versions.

Which devices do your test automation engineers use?

Your test automation engineers use mobile devices to write tests, maintain automation scripts, and perform manual testing. Which devices do you want them to access? If your application behaves or renders differently on different form factors (small phones, large phones, and tablets), consider giving your test automation engineers access to a set of devices covering these characteristics.

What are the end-to-end testing scenarios for your CI / CD pipeline?

Agile teams ensure their main branch is stable by including end-to-end functional tests in the CI / CD pipeline. When setting up end-to-end functional tests in a CI / CD pipeline, you need to balance the time it takes to execute tests with the need to keep feedback loops short. Which smoke tests will trigger early warnings for unintended regressions in your app's critical scenarios? On which devices do you need to run these tests?

How do you plan testing campaigns?

End-to-end tests in CI / CD pipelines are usually a subset of your end-to-end tests. How often do you want to run your full suite of end-to-end tests — daily, weekly, or monthly? On which set of devices do you want to run these tests?

Do you need private devices?

Cloud-based mobile device farms usually offer public devices and private devices. You can use both with Eggplant software. SaaS mobile device farms often provide thousands of public devices so you can install and test your app. You're not tied to a specific device and can swap devices as often as you like.

Customers share public devices, so you need to be comfortable deploying unreleased versions of your app to a device that others can access, even though most SaaS mobile device farm providers promise to clean devices after use. If your app connects to a back end, you need to make sure those public devices can access your back end using the public internet.

An alternative is private devices. Private devices dedicated to you offer extra features, such as virtual private network tunnels into your corporate network. You can't rotate them often, and they are usually more expensive to set up and maintain.

What about emulators?

You may want to include both emulators and physical devices in your test mix. Consider using emulators when you want to focus on the functionality of your app and when hardware scenarios, such as sensors, cameras, or geolocation, are unimportant.