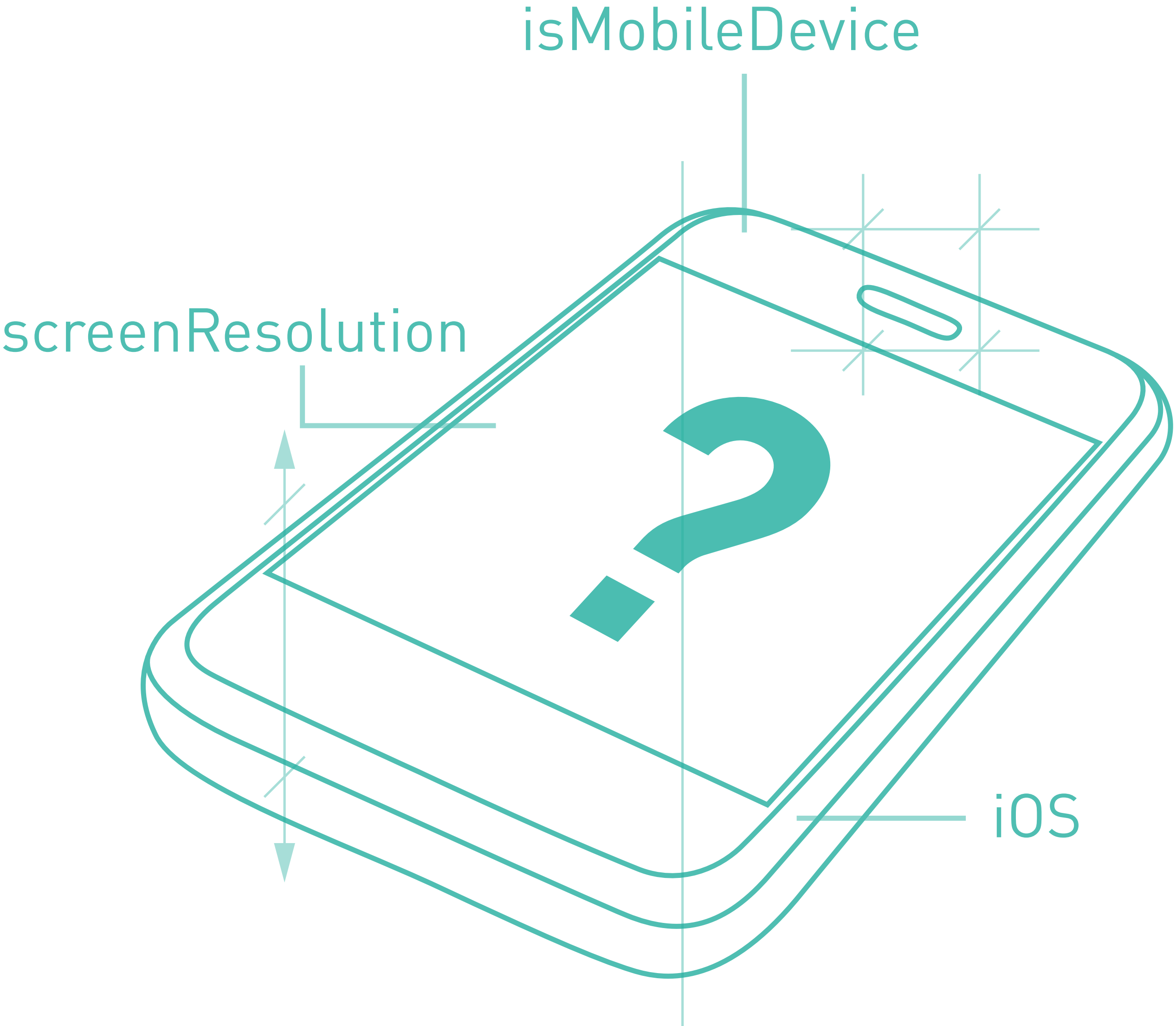
DeviceAtlas™

DEVICE DETECTION AND INTELLIGENCE BUYING GUIDE

WHAT YOU NEED TO KNOW TO MAKE AN INFORMED CHOICE



CONTENTS

INTRODUCTION

The mobile web is growing, fast. The proliferation of disparate device types across the market – smartphones, tablets, laptops, gaming consoles, smart TVs — has made for an ever more fragmented device landscape. These technological developments have significantly increased the need to identify how customers are experiencing and consuming content at the device level.

If you are a business with a strategic digital focus, you will have probably thought about these types of questions already:

- How do I make my website look great on all mobile and connected devices?
- How do I accurately deliver or target content downloads such as apps, games or ads to different mobile devices?
- How do I track exactly what devices visit my sites?
- How do I make sure my services and solutions are supported by accurate, comprehensive device data?

If you're not able to satisfactorily answer these questions, applying a third-party device detection solution might be the right choice.

Those who put in place a strategy to understand how their customers are accessing content and services on mobile and other web enabled devices will enjoy a significant competitive advantage in this increasingly connected environment. Those who gamble on "good enough" device detection solutions will risk losing customers, market share and ultimately profitability.

This paper will guide you through the process of evaluating and selecting the right device detection supplier by listing 8 most important factors to consider.

DETECTION METHOD

Device detection solutions analyze HTTP header fields (such as User-Agent string) that contain information about the requesting device and its browser. This process can be carried out in a number of ways.

? What HTTP headers do you use for device detection?

The device and browser information contained in a User Agent (UA) often isn't correct for the requesting device because many devices deliberately pretend to be something else. Therefore you need to look for a solution that is sophisticated enough to handle deliberate masquerades and other similar situations where a UA is not what is seems.

DEVICE DETECTION SHOULD ANALYZE ALL HTTP HEADERS

Device detection solutions should analyze all HTTP headers, and not just the User-Agent string, to ensure no inaccurate results are returned. This is essential to detect third-party browsers. Solutions that only look at the UA cannot reliably identify the device using Opera Mini, UC Browser, etc.

Given that Opera Mini and UC Browser enjoy some 5-15% market share, device detection solution that don't analyze all HTTP headers are unlikely to achieve accuracy higher than 85-95%.

MAKE SURE THE DETECTION METHOD IS ROBUST

Device detection solutions often rely on having seen every possible version of every possible User-Agent (UA) string to work correctly. This makes the detection method brittle in some cases. For example, a new version of Chrome, or Safari won't be recognized unless the detection system has been updated with the new UA.

It is best to look for robust solutions that will not fail in this situation, due to the fact that they don't require the entire User Agent string to work properly.

DETECTION ACCURACY

One of the core characteristics of a device detection solution is accuracy. However, this aspect is not always easy to verify.

In every web environment there is some traffic that reduces detection rate, including undetectable traffic due to e.g. erroneous HTTP headers. It is important to make sure the device detection solution is fully transparent on these kinds of traffic, as otherwise it is not possible to measure its accuracy at all.

? Does your solution flag unrecognized devices, so that the accuracy levels can be measured over time?

FALSE POSITIVES

Transparency on non-detected devices means lower rates of 'false positives' which are often used to conceal accuracy problems by returning a result, even a wrong result, for any request.

Device detection solutions reporting 100% accuracy are masking misdetections, and concealing inaccuracies that impact on overall results. For use cases where accuracy is paramount, such as web analytics and mobile advertising, this practice is not acceptable.

To test accuracy we recommend following the three steps:

1. Pick a few devices with known properties (it is best if these devices are not the current most popular ones)
2. Find HTTP headers for these devices by checking your own server logs, or looking for online sources (such as User-Agent, X-WAP-Profile, X-OperaMini-Phone-UA, Device-Stock-UA, or X-UCBrowser-UA)
3. Use the device detection solution and spot-check the returned properties against the known devices

If you know that traffic to your site consists of a few key devices you should test all of these.

DETECTION SPEED AND MEMORY FOOTPRINT

The speed of a device detection solution impacts on its real-time applications, such as the website's loading times, or ad targeting. Make sure that detection doesn't create a bottleneck, impeding overall speed of your online services.

Device detection vendors may provide you with impressive speed metrics but watch out for the small print. Speed depends on many different factors such as server resources, CPU, connectivity and more. Giving an 'average detection speed' for a solution is largely meaningless. It is best to take a trial and manually test the speed within your specific environment.

? How do I measure real world detection performance in my environment?

HOW TO TEST DETECTION SPEED?

To obtain meaningful performance metrics you need a large quantity of User-Agent strings ideally global in nature. It's best to use your own web logs as a source of HTTP headers to test real world performance. It is also possible to get HTTP headers for testing from free sources accessible online.

When testing performance it is important to use the full User Agent strings rather than snippets of these strings to find out if the entire string is required for detection. Some device detection solutions will perform differently depending on the number of properties requested. Ensure like-for-like comparisons by testing the same number of properties during detection tests.

SERVER AND MEMORY FOOTPRINT

While testing the speed of device detection solution, it is also worth checking if the memory footprint of the detection API is a good fit for your environment. The memory footprint should be small enough not to cause resource issues on production servers and should be stable over time.

PROPERTY COVERAGE

Device detection solutions typically offer a set of properties for each device in the database. It is important to ensure that your device detection provider supports the properties you'd like to target and analyze.

? Which device properties can I detect and target with your solution? Do you automatically populate properties for new devices?

TEST HOW PROPERTIES ARE DETECTED

Ensure that the properties you need are available. You can do this by checking if devices supported by the detection solution have values for the properties in question and whether these values are right. This requires acquiring User Agent strings from your logs (or other sources such as listed on page 5) and running as many of them as possible through the solution and tabulating out the results for the property in question.

WATCH OUT FOR PROPERTY ASSUMPTIONS AND SYNONYMS

Data completeness is not always a given. For example, a device detection solution may not have full data for a new device, but may do for the previous model. Automatically populating device properties according to assumptions is liable to cause accuracy problems.

The same can be said about using property synonyms. Diagonal screen size in inches and in centimeters is one property in practice but the device vendor might treat them as two separate entries seemingly increasing the coverage.

THE NUMBER OF DEVICES IN THE DATABASE

Device detection solutions all claim varying sizes for their databases, usually measured by "number of devices supported". It is important to consider the method of measuring the size of the device database.

? How do you measure the number of devices in your database?

The total number of distinct web-enabled mobile devices is changing constantly. The exact number is not as important as the global coverage of devices added to the base.

Device detection solutions tend to have different levels of coverage for devices in different territories. Traditionally, device detection solutions have good coverage of devices in their own territories, but have lesser coverage of others regions. Given the increasingly global nature of mobile web traffic, device detection solutions need to recognize global devices, not just local ones.

Device detection vendors who claim they have the highest number of devices may be counting non-meaningful combinations of devices and User Agent strings. If you see claims of 100,000s of 'device combinations', you may want to consider how meaningful that claim really is.

A further check that should be made is to distinguish between "included devices" in a database and "detectable devices"—if a device cannot be individually distinguished by the solution there is not much point in having an entry for it. For example, a phone can be available in many colors but it's not possible to tell them apart from the User Agent string data.

FREQUENCY OF UPDATES AND DATA SOURCES

Building and updating your own device database is extremely costly and time-consuming. A quality device detection solution helps you save plenty of time by constantly updating the database using a variety of sources.

? **How often do you update your device database? What data sources do you use?**

Daily updates are ideal, given that anything less frequent may miss out on traffic from devices that become popular overnight, such as iPhone 6. We recommend testing for data currency by trying to detect new devices immediately after their release.

THE DATA SHOULD BE MULTI-SOURCED

Multi-sourcing the device data is one of the most important aspects influencing detection accuracy.

Due to the enormous variety of mobile devices available today, it is virtually impossible to gather all the data from a single source. Furthermore, data sources invariably contain errors so it is vital that additional sources are available. A device detection solution should incorporate data from device manufacturers and mobile operator partnerships.

TRANSPARENCY

Many solution vendors do not make it clear where exactly their data comes from, leaving their customers open to potential legal issues. Look for device detection vendors that tell you exactly what the data sources are.

INTEGRATION AND ARCHITECTURE

Device detection solutions can differ substantially from an architecture and data distribution point of view. Check that the solution is a good fit in your environment and workflow.

? Is device awareness available at both, server level and application level? Which programming languages are supported?

Utilizing the web server level for device detection can ease the burden on application servers minimizing the processing and memory footprint. It is important to ensure that detection solution can support both deployment options.

A solid device detection solution will come with simple step-by-step implementation guides that come with code samples in all major programming languages.

FUTURE PROOFING

The device detection solution should be futureproof when it comes to the update path and frequency, API consistency, and the development cycle. The solution is also futureproof when it supports scenarios where device properties can change with operating system or browser updates. Phones are no longer static devices — their properties change over time with over-the-air (OTA) updates.

SUPPLIER REPUTATION AND MARKET EXPERIENCE

Building a robust and reliable device detection solution requires the investment of a great deal of resources over a long period. Choosing a detection supplier with little experience and short presence on the market is a risky move.

? Is your company experienced in delivering device detection for similar businesses? Are they a trusted supplier?

It is important to choose a reputable supplier that guarantees the quality of their services, are market-proven and have a range of clients and partners from different industries.

A device detection supplier should have a proven track record of working with industry leaders that confirm its ability to deliver high speed and high accuracy device detection at scale. It is important to consider case studies related to your industry to make sure that the chosen device detection supplier is a good fit for your particular type of business.

QUESTIONS TO ASK YOUR SUPPLIER

- ✓ Does your solution flag unrecognized devices, so that the accuracy levels can be measured over time?
- ✓ How do I measure real world detection performance in my environment?
- ✓ Which device properties can I detect and target with your solution?
- ✓ Do you automatically populate properties for new devices?
- ✓ How do you measure the number of devices in your database?
- ✓ How often do you update your device database?
- ✓ What data sources do you use?
- ✓ Is the device awareness available at both, server level and application level?
- ✓ Which programming languages are supported?
- ✓ Is your company experienced in delivering device detection for similar businesses?

TRY DEVICEATLAS FOR FREE

If you have a strategic need to ensure all devices are detected to retain market share and leadership, you can try DeviceAtlas device detection for free.

To start your trial contact us today at sales@deviceatlas.com or visit deviceatlas.com.