

ECMAScript/JavaScript on Wearables - Outline

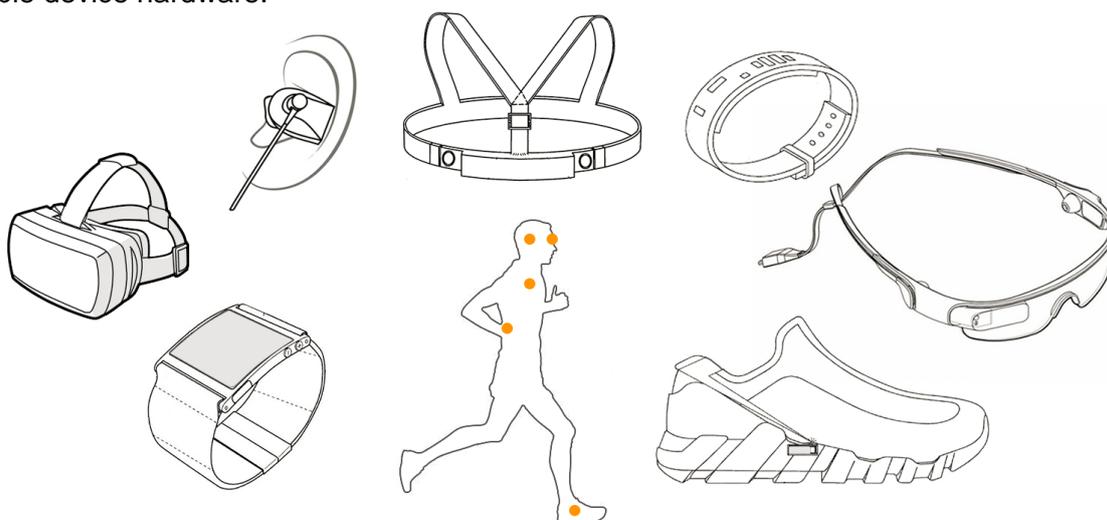
An opportunity to standardize on a popular established developer technology

Introduction

The wearable category of digital devices has moved from future concept to market reality. The category is still immature, with product creators and consumers still actively exploring the uses of these devices. One common indicator of a category which has not yet matured is the lack of standards, whether those are de facto standards like Windows on personal computers or true industry standards like HTTP and HTML on the web. There are many areas where standardization may be applied to wearables including the digital hardware, energy, radio, communication protocols, etc.

This paper focuses on software standardization. The first goal of software standardization is the creation of wearable products with built-in software that is reliable, cost-effective, secure, and built with modern technologies. Achieving this allows manufacturers to deliver wearable products to consumers with greater ease and predictability. The second goal is to allow users of wearables the freedom to install software from independent developers, just as they do on PC and mobile devices. Achieving this allows the wearables category to grow more rapidly by leveraging third parties to add interoperability with their hardware, software, and cloud services, thereby giving consumers the choice of adapting their wearable hardware to their individual needs.

We propose the standardization of software for wearables by applying the ECMAScript¹/JavaScript programming language. The success of JavaScript on the web, backend servers, and mobile is unquestioned. The language continues to evolve, carrying forward with it an ever-growing ecosystem of skilled developers. Standardizing at the language level is the right choice. Technical data is available that shows JavaScript is a realistic choice. A complete implementation of the current ECMAScript language standard (ECMAScript 2017, 8th edition) is available for low cost wearable device hardware.



¹ JavaScript and ECMAScript are the same thing. JavaScript was formally standardized by Ecma International as ECMAScript. In what follows for brevity we will use the term “JavaScript”.

Hardware diversity

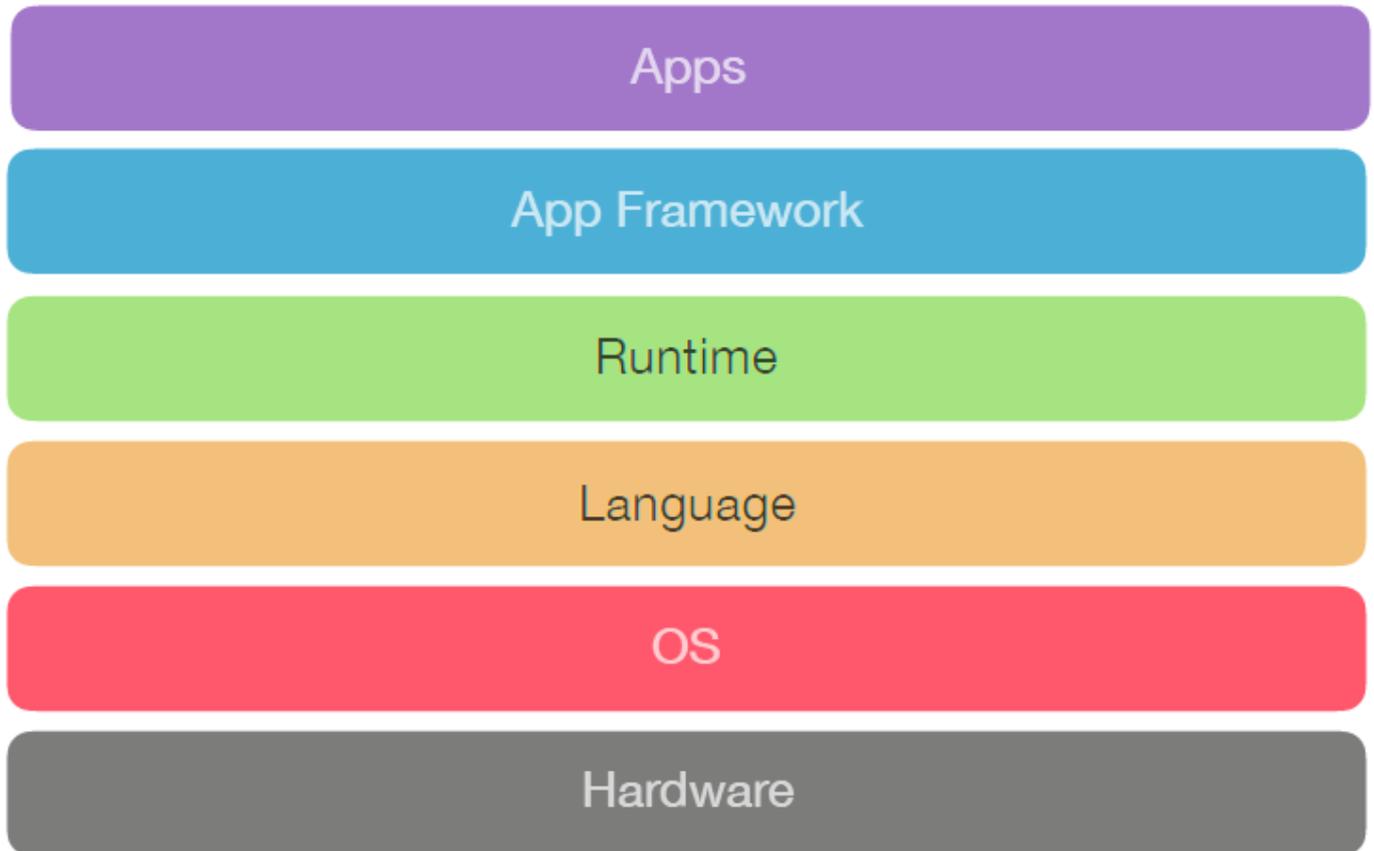
While wearables category is still taking shape, it is clear that the hardware used will remain diverse for the foreseeable future. There is no dominant manufacturer or product category on the horizon. This broad, and to some degree unpredictable, diversity of hardware capabilities presents a challenge for any standardization effort which, by necessity, must take a long-term view.

Rather than focus on the differences, it is helpful to consider the similarities shared by most wearable devices.

- Small physical size. The size tends to force a in reduce the hardware capabilities compared to mobile devices. One impact of this is that user interactions change due to reduction in screen size, or even elimination of a display entirely.
- Battery powered. Wearables are always operated on battery power, often a small battery due to the small physical size. This requires software to go beyond energy efficient to being energy aware.
- Connectivity. Nearly all wearable devices have a wireless connection used to share sensor readings and receive notifications. There are many different wireless technologies used with BLE, Wi-Fi, and LTE being common examples.
- Resource constraints. The small size and energy requirements tend to limit the available computing hardware including CPU power and speed, RAM, code space, and storage space.
- Responsive and reliable. Users expect the devices they wear to respond immediately and to have an even higher degree of reliability than their phones.
- Many sensor inputs.
- Cost remains important.

Architectural layers

The topic of software standardization is broad. It would be naive to suggest that the full software stack can or should be standardized. Let's consider each of the major software layers.



At the bottom of the software architecture stack is the operating system (OS). Wearable devices today are built with a wide range of operating systems. This is unlikely to change, as the choice of operating system is based on a complex set of factors which include the underlying hardware, corporate preference, and operating system capabilities. Some wearable devices use large operating systems, typically a Unix or Linux variant such as Android Wear and watchOS.

Devices built on these technologies tend to be more expensive and physically bigger, in part to accommodate larger batteries. Other systems use an RTOS, which is much lighter and energy efficient but also provides many fewer built-in software capabilities, putting more of the software burden on the device manufacturer.

Apps are at the top of the software architecture stack. One intended outcome of software standardization is the potential for apps to be written such that they can be used across wearable hardware from different vendors.

The runtime is the set of services provided by the operating system for applications to use. Depending on the wearable device, these services include memory management, networking,

sensor access, timing, and graphics. The runtime services are provided to the application through an API (application programmer interface).

Often apps are not built directly on the APIs provided by the operating system. Instead, app developers choose to build on an application framework (App Framework). It is reasonable to expect that wearables, in time, will also have its own collection of application frameworks for developers to choose among. Consequently, there is no need to standardize the application framework layer. To do so would limit developer choice, and may be wasted effort as application frameworks come in and out of style quickly.

It may be surprise to consider the programming language (Language) among the layers in a device's software stack. A common notion is that more-or-less any programming language can be used on any device, that the selection of language is a primarily a matter of developer preference. This is largely true on more capable devices, devices with effectively unlimited resources. The wearables category is now, and for the foreseeable future, defined by resource constraints. As a result, it seems preferable to select a single programming language to support well on wearables, rather than provide a variety of languages with uneven implementations.

If wearables is to standardize on a single programming language, the natural question arises as to which one. With the goal of reaching a large number of independent software developers, a scripting language is preferred. While there is some overhead, environments like the web show a scripting language benefits the greatest number of developers. The most widely used scripting language today is JavaScript, which is used for the web, backend servers, and mobile applications. The popularity of JavaScript brings with it an unmatched developer community. Further, over its twenty years, JavaScript has evolved into a thoroughly modern language with many language features to ease development.