[CP-01-015] Mobile Devices

Abstract

Mobile devices refer to a computing system intended to be used by hand, such as smartphones or tablet computers. Mobile devices more broadly refer to mobile sensors and other hardware that has been made for relatively easy transportability, including wearable fitness trackers. Mobile devices are particularly relevant to Geographic Information Systems and Technology (GIS&T) in that they house multiple locational sensors that were until recently very expensive and only accessible to highly trained professionals. Now, mobile devices serve an important role in computing platform infrastructure and are key tools for collecting information and disseminating information to, from, and among heterogeneous and spatially dispersed audiences and devices. Due to the miniaturization and the decrease in the cost of computing capabilities, there has been widespread social uptake of mobile devices, making them ubiquitous. Mobile devices are embedded in Geographic Information Science (GIScience) meaning GIScience is increasingly permeating lived experiences and influencing social norms through the use of mobile devices. In this entry, locational sensors are described, with computational considerations specifically for mobile computing. Mobile app development is described in terms of key considerations for native versus cross-platform development. Finally, mobile devices are contextualized within computational infrastructure, addressing backend and frontend considerations.

Keywords: cloud computing, data collection, location-based services, mobile development, mobile map, mobility, software development kit, ubiquitous computing, volunteered geographic information (VGI)

Author & citation

Ricker, B. (2019). Mobile Devices. The Geographic Information Science & Technology Body of Knowledge (3rd Quarter 2019 Edition), John P. Wilson (ed.). DOI: 10.22224/gistbok/2019.3.8.

Explanation

1. Definitions
2. Introducing Mobile Devices
3. Mobile Devices and App Development
4. Mobile Devices, Connecting to the Cloud and Other Devices

1. Definitions

mobile device: computing systems small enough to be used by hand, such as a smartphone or tablet, and including smartwatches and other wearable computers

locational sensors: sensors that are housed in mobile devices to detect the users' current location, including Global Positioning System (GPS) receivers (location is derived by connecting to GPS satellite constellation), WiFi (location is derived by connecting to wireless internet), Cell Id (location is derived by connecting to cell towers), and more

ubiquitous computing: an environment that is saturated with computing and communication abilities which are seamlessly integrated into our daily lives

interoperability: the ability for devices and/or equipment to communicate with each other and to exchange information

Integrated Development Environment (IDE): software that integrates all the necessary components to write, debug, run, compile, and execute code for application (app) development. IDEs typically bundle the reusable code found in Software Development Kits (SDKs)

software development kit (SDK): a software library, documentation, and code samples that are used to create programs to be run on a specific operating system. SDKs often include compilers, debuggers, documentation, and other files that help with the development process. SDKs typically, but not always, are utilized via an IDE

native app: an application (app) developed for deployment on a specific operating system

cross platform development: the design and development of a mobile application (app) to be deployed on a range of operating systems

progressive web apps: web apps that behave like mobile apps in that they work offline, can deliver push notifications and can access (locational) sensors in mobile devices

Random-Access Memory (RAM): computer data storage that retains the code and data being used by the machine in the moment

Central Processing Unit (CPU): a group of microchips located on the motherboard that control data flow to and from other parts of the computer

Graphics Processing Unit (GPU): the hardware module on an electronic circuit built into a computer with dedicated RAM dedicated for rendering graphics, image processing and required for interactive elements and multimedia such as video and maps

2. Introducing Mobile Devices

2.1 Mobile Devices and Society

The term **mobile device** refers to a computing system small enough to be used by hand, such as a smartphone or tablet computer. Mobile devices more broadly refers to mobile sensors, computers, and other hardware that have been made for relatively easy transportability. By this definition, mobile devices also include activity trackers, smartwatches, environmental sensors, flying cameras, and onboard computers attached to unmanned aerial vehicles and any mobile, portable computers.

The sensors within mobile devices that make these services possible include: Global Positioning System (GPS) receivers, WiFi, cell signal and other **locational sensors** now found in most mobile devices. Locational aware sensors such as GPS receivers and other survey equipment were once enormous (see Figure 1), expensive and difficult to operate, meaning only a few highly trained professionals had access to them. Spatial data collection

was only possible with the help of these highly trained professionals. Only those with sufficient sources of funding had access to the know-how, and the sensors, and the databases required for data collection. (See Figure 1: Early GPS receivers). Now most smartphones or mobile computers have more computing power than computers that took up entire rooms only 30 years ago.



Figure 1. GPS receivers used by the United States military circa 1978. Source: https://bit.ly/2TS41Xx

Sensors have gotten smaller, graphics cards have improved, and hardware prices have decreased, making mobile devices more accessible to the general public. People are becoming more comfortable and familiar with them. These tools that were once only available to trained professionals are now accessible and used for what have become daily tasks. Two thirds of the global population, 5 billion people, now have access to mobile devices (Aterfield, 2017). In theory, this means that two thirds of the global population have access to maps and locationally relevant information, due to the sensors housed within these devices. This confluence of events is increasing opportunities to collect and share geospatial information with a growing audience. The general public is gaining access to and regularly using technologies associated with Geographic Information Systems (GIS), even though the end users may not be familiar with the term GIS or aware that they are using them.

Examples of everyday use of mobile GIS include navigating an unfamiliar environment, looking up current traffic conditions, finding friends nearby with common interests around you, and providing location-based business recommendations. It is also possible to send push notifications using location-based beacons and other platform dependent sensors. Despite their prevalence and utility, with the uptake and accessibility of mobile devices have come concerns related to users' evolving relationship with them, including how space is navigated and understood, opportunities and concerns for more and alternative data collection practices.

The first real excitement around mobile devices and mapping came with the advent of the iPhone in 2007, when users could access web map functionality associated with, tiled maps seamlessly anywhere. This access is an affordance of WebGIS accessible via mobile devices (see **WebGIS**). One of the most common and useful routines of mobile device usage is for wayfinding. Alarming results from studies hint that people who depend on their mobile device for wayfinding have incomplete cognitive maps; when they get lost, they stay lost longer than those who use static paper maps (Ishikawa et al., 2008). This raises important concerns about attention and social interaction. Additionally, using a mobile device while in a new place may significantly influence experience and decision making in that place (Ricker, Hedley, & Daniel, 2014). Tourism research reveals that people traveling with their mobile devices feel more confident and are more likely to go somewhere new (Wang, Park, & Fesenmaier, 2012).

Mobile devices afford users tools and resources to both collect and disseminate location-based information both for professional and leisure purposes. With the help of mobile devices and their applications (apps), it is now possible to more easily and directly foster contact between citizens and their governments, where citizens can report potholes, a tree down, or other municipal challenges that need urgent attention.

These advancements are bringing to fruition ubiquitous mapping, which refers to the ability for users to access or create maps anywhere and at any time, to resolve immediate spatial problems (Morita, 2005, 2007). Mobile devices have disrupted society both personally and professionally, and have shifted social norms and expectations as a result. Now, geographic information is increasingly produced and consumed on mobile devices rather than on desktop computers (Muehlenhaus, 2013). Data can be collected in an active manner (when a user clicks submit) or in an ambient manner (when an app runs continuously in the background).

Traditional GIS was critiqued for only incorporating quantitative data. Critical geographers called for a shift to incorporate qualitative data into GIS as well (Schuurman, 2000, 2006; Sieber, 2004) (see **GIS and Critical Ethics** and **Feminist critiques**). Mobile devices are facilitating and encouraging more and diverse locational data to be collected and shared. People can now share qualitative information about their distinct experience or place in the world. However, this brings new ethical considerations into question in terms of whether users know their locational information is being collected or not. Some apps that do not explicitly offer location-based services still collect locational information about their users. This information can be used to tailor ads based on the user's location, or it is collected simply to sell to others (see **Location Privacy**). The granularity of the spatial data being collected has significant and direct ramifications for user privacy and also for mobile device battery life.

For a mobile app to reach the intended goals for the end user, a strategic plan for its design needs to be in place. This plan should include design in terms of a user interface, and what spatial data need to be collected and/or shared, and how and where they will be stored are among a myriad of other considerations (see **Strategic Planning for GIS Design**). Mobile devices are not a silver bullet for every problem. Careful consideration needs to be in place before app development takes place, to save time, money, and human resources.

2.2 Mobile Devices are Ubiquitous

Mobile devices are considered to be pervasive computing or **ubiquitous computing**, which refers to an environment that is saturated with computing and communication capacities that are seamlessly integrated with users' daily lives (Satyanarayanan, 2001; Weiser, 1991). Mobile devices and mobile computing are inherently geographic, because integrated computing can now happen anywhere (Goodchild et al., 2004). Mobile devices associated with ubiquitous computing do not facilitate anything necessarily new, but enable users to accomplish tasks faster and easier (Weiser, 1991). Ubiquitous computers are considered information appliances because they are so embedded in everyday life (Muhlauser, 2009).

Some argue that eventually, mobile devices will achieve invisibility, when devices disappear from user's consciousness, they have become transparent (Satyanarayanan, 2001). Weiser (1991) goes on to refer to this concept as "embodied virtuality" and suggests that computers in the future will be invisible not only by metaphor but due to their small size, in reality as well.

A mobile device interface should not capture the full attention of the users who will still need to execute foreground tasks (Pallotta et al., 2009), such as crossing the street, social interaction, or decision making while wayfinding. This is one of the many reasons why interface design is so important (see **Mobile Maps and Responsive Design**). Computers should only become noticeable when a message or alert has been passed to the user (example: when a doctor needs to be alerted that a patient requires immediate attention). Presently, mobile devices are immersive and distracting rather than transparent. Car crashes, pedestrian collisions, and other fatalities are regular occurrences caused by the distractions related to mobile devices. Unobtrusive interfaces are key to ubiquitous computing.

2.3 The Relevance of Mobile Devices for GIS&T

Mobile devices are deeply rooted and especially relevant for Geographic Information Science & Technology (GIS&T). Mobile devices and their role within spatial data infrastructure are important for planning and usability purposes, especially related to data collection. Traditionally, collecting data to be included in a GIS was extremely laborious. First, data were collected in the field on paper and then digitized once back in an office. This process afforded many opportunities for data entry errors. Now data can be documented directly in the field. Mobile apps developed for specialized data collection tasks can mitigate human error in data collection through the use of dropdown menus, and automated time and location detection. These locational data can be saved locally and/or directly can populate a remote spatial database when the mobile device has connectivity.

This makes data collection on the job much easier. Depending on how an app is configured, it is possible to geotag data that are entered with or without the enterer knowing it. The

precision of locational data may be otherwise questionable, as the error in the GPS is heavily influenced by the surroundings. However, even in the best case, the error for off the shelf mobile devices is around 3-10 meters accuracy, which depending on the application, may or may not matter.

Mobile devices facilitate data collection for work and for fun through: volunteered geographic information (VGI) (Goodchild, 2007), location-based services (LBS) (see Location-based Services), social media on the move, and many diverse citizen science apps. LBS are services that offer information about where a location aware device user is situated (Gartner, Cartwright, & Peterson, 2007). LBS are often populated by VGI (Ricker, Hedley, & Daniel, 2014). Social media and VGI evoke ethical concerns about data privacy, but in terms of data quality, mobile devices improve the likelihood of locational accuracy. (Note: sensor errors are always a challenge.)

As mentioned already, mobile devices have afforded the opportunity to collect data from a wide range of people and places that was once, not long ago, unfathomable. Disciplines outside geography and GIS are seeing the utility of these devices for data collection, making the skills of those with GIS&T backgrounds to collect and process these data streams even more relevant and valuable than ever.

3. Mobile Devices and App Development

3.1 Computational Considerations and Constraints

There are several computational considerations and constraints that are specific to mobile devices. These include the relatively limited computational capabilities due to their small size, network connectivity, the need for locational sensors to collect and deliver locationally relevant content, and battery life.

Mobile devices generally have less computing power than desktop computers, but more than desktop computers 20 years ago. While computing power is becoming smaller and cheaper, desktop computers are still faster and more powerful than mobile devices due to the availability of additional electric power, physical space for components, and better cooling. **Random-Access Memory (RAM)** is computer data storage that keeps the code and data being used by the machine in the moment. The **Central Processing Unit (CPU)** controls instructions and data flow to and from other parts of the computer, and relies on a group of microchips located on the motherboard. The amount of RAM a computer has influence on how fast the machine responds by how much data it can render at once. The **Graphics Processing Unit (GPU)** is the hardware module on an electronic circuit built into a computer that is composed of processors, registers, and dedicated RAM required for rendering graphics and image processing (see **GPU Programming for GIS Apps** and **Graphics Processing Units**). The GPU is vital to load images, including maps, and is vital for interactive elements and multimedia such as video. RAM is used to store source code files for the GPU. Since smaller devices have less powerful GPUs installed on them, and less RAM, they have less processing power.

Due to the unique computing capabilities associated with mobile devices, specialized technological considerations for mobile apps need to be deliberated. For this reason, new mapping frameworks are being developed to enhance speed and agility. One example is the increasing usage of vector formats for base maps, rather than the traditional approach

of rendering aerial photos or tiled rasters which take a lot of energy to load. This is important for rendering mobile maps efficiently. For example, Leaflet is a JavaScript library specifically optimized to load maps quickly on mobile devices. It only uses 38 KB of JavaScript compared to an alternative such as OpenLayers, which has much more functionality but is also much larger, in megabytes of JavaScript. For the same reason, Apple rewrote its map library MapKit; to make it smaller. It is now called MapKit JS since it is made with JavaScript. JavaScript libraries, when well written, should remove any bloat, clutter, and complexity to render quickly on mobile devices. End users will not know how many lines of code are being run, but they will know when the app is taking too long to render, and they will close it. Additionally, loading copious amounts of data (think satellite imagery base maps) while the users are using their cellular data plans instead of WiFi will have a significant monetary cost for them.

Network connectivity is another key factor for mobile device users and their user experience. Network connectivity is necessary to render a map, and to display location-based content. As the intensity of interactions between a users' personal computing spaces and their surroundings increases, and as more people use the same network, these factors will slow server bandwidth, causing frustration to a mobile device users (Satyanarayanan, 2001). This may lead to unpredictable variation in network quality in different places. Network connection is vital to connect people to relevant information that they are seeking on the go, including to communicate with each other. Some map apps for wayfinding are including functionality to be able to download basemaps of regions ahead of time so that these maps are stored locally, making them accessible even without connectivity.

Location in mobile computing is both relevant and important: from the location of the mobile device and its user, to the location of information transmission lines and towers. The location of information processing, the locations that are represented in the data, and the location of data storage (Goodchild et al., 2004). Most relevant for GIS&T, mobile apps are able to tap into locational sensors in the mobile devices to collect information from the user and/or deliver locational relevant information to the user. Most mobile devices such as smartphones have the following sensors that enable locational information to be detected from the user: a GPS receiver (connected to satellite constellation to triangulate the receiver's location), Cell ID (cell tower zone), WiFi (registered to a specific location), gyroscope (senses orientation of the device pitch and roll, which are useful for stitching together panoramic photo mosaics), accelerometer (how fast device is moving in a specific direction – useful for navigation), magnetometer (compass), proximity sensor (senses how close the device is to something else, like the user's ear), Bluetooth beacons (need to be set up in different locations and turned on) and more (Nield, 2017). Different devices may have more or fewer locational sensors (e.g. barometer to measure real-time hyper-local weather conditions). All of these sensors can be accessed when developing a mobile app. In terms of accessing a mobile device's location, the best sensor for the job depends on the location. For example, in a wide-open field, far from buildings, GPS would be the best sensor for locating the user. Whereas in a city dense with tall buildings that block GPS signals, or underground, then WiFi will provide the most precise locational information about the mobile device and its user.

As with any device that needs energy to function, battery life is a concern in pervasive computing (Satyanarayanan, 2001; Miluzzo et al., 2008). Mobile devices typically contain a of suite of powerful location aware sensors that require large amounts of memory and battery power. Screen size, size of the device, frequency of Bluetooth, GPS usage, and

other sensors are all influences that can affect energy use and drain the battery quickly (Miluzzo et al., 2008). App developers make choices about how often locational data should be collected, at what temporal and spatial intervals. Apps make it known to the user if they are accessing their locational information but not at what time and space interval. This has both energy and ethical ramifications. With these considerations in mind, in terms of mobile apps development, it is useful to compare the strengths and weaknesses of native versus cross platform development and web apps.

3.2 Native versus Cross Platform versus Web Apps Development

Part of the reason the saying, 'there is an app for that' is so prevalent is because apps are becoming increasingly easy to build. People are using narrowly targeted use case apps for personal reasons, and they are expecting to have equally efficient apps to use at work. Users have personal preferences in terms of the type of phone they like to use and are comfortable with, which is largely tied to the operating system. Bring Your Own Device (BYOD) to work is becoming increasingly prevalent to accommodate these personal preferences in terms of devices. The aim of the BYOD policy is to enhance workers productivity and give the perception that an employer is accommodating while simultaneously reducing costs (Gaff, 2015). Policies need to be in place for this to work smoothly for both employees and employers (Gaff, 2015) to address legal and privacy considerations. BYOD is an important consideration for GIS Analysts, as they are creating new or alternative opportunities for field data collection or access to GIS in the field (Kerski, 2013).

GIS analysts are increasingly expected and able to build mobile solutions for a GIS ecosystem. Building customized mobile GIS apps for smartphones is becoming more accessible to even those without formal computer science training or education, making them increasingly feasible to deploy as GIS software companies are building suites of tools to easily integrate mobile solutions. When developers are consulted immediately, especially those with little or no GIS experience or knowledge, they will likely recommend a mobile solution based on their skill set and knowledge base, not necessarily on what is best to meet the needs of the spatial task at hand. In reality, existing system structure and legacy will influence decision making, legacy of data servers and map libraries or stores (Goodchild et al., 2004), and the knowledge of the in-house developers. Developers are often drawn to learn the tools that will help them the most on the job or help them earn the most money (Puvvala et al., 2016). If no in-house developers are available, below are recommended considerations to take before deciding on a development strategy.

There are series of considerations to be made before the development process is chosen, the first of which is to consider the specific problem being addressed and the end user - the target audience for the app being built.

- What is the aim or goal of the app?
- What problem does it aim to solve?
- Who is the target audience?
- What devices (operating system) will the target audience be using/have access to?
- What locational sensors and level of accuracy are needed to be called upon to meet the aims of the task?
- How and where will users access the app?
- Will the users be offline?

- How are they expected to find and install the app?
- Does data and app rendering speed matter?
- What is the budget for the project?

Next, consider the app developer:

- Who will build the app?
- What is their skillset?
- Are they able to quickly learn a new tool or programming language?
- Are they more comfortable using a Graphical User Interface (GUI) to build an app?

Answers to the questions posed above will influence if a **native app**, **cross platform app**, or **progressive web app** is best suited to get the job done. This decision is dictated by what operating system (OS) is accessible to the end user, and is related to user experiences expectations associated with OS, which are largely associated with front end development. At the time of writing, there are three dominant operating systems for mobile devices. These are Android by Google, iOS for Apple devices and Windows for Microsoft related devices. Android runs on a wide range of mobile devices in all different sizes from different hardware manufacturers. iOS only runs on Apple hardware. Microsoft mobile Operating System (OS) is called Windows for Microsoft devices, and is similar to Android in that it runs on several different types of devices from a range of manufacturers (See Figure 2). With this in mind, there are largely three choices for mobile app development: Native App, Cross Platform App, or Progressive Web App. For each of these methods, there is software available so that mobile apps can be built and customized using a graphical user interface (GUI) for customizing, connecting code and layout of the user interface using specific **software development kits (SDK)**. To build highly customized tools within apps typically requires complex programming capabilities. Several common tasks and default settings associated with GIS are readily available in drag and drop GUI settings, within individual SDKs.

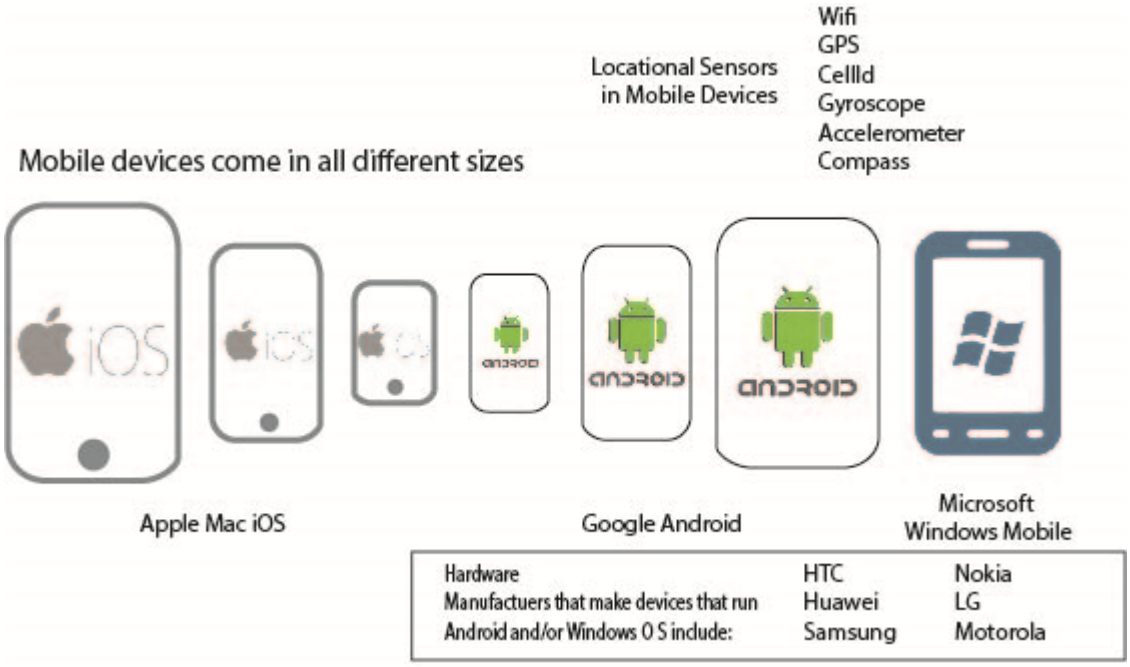


Figure 2. Representation of the wide range in sizes of different mobile devices, from different manufacturers, running different operating systems, all with several locational sensors included. Source: author.

3.2.1. Native App Development

Native apps usually afford the end user the best user experience. The main benefits of developing individual native apps in their associated SDKs is that a well written native app runs faster and there is less likelihood for the app to freeze or crash when running. There are specific frameworks in place for each platform to tap into the location sensors directly and efficiently. Specific design standards have been well defined for each mobile operating system platform, which means the user interface in a native app will likely be more familiar to the app user. This all leads to a better user experience and if distributed on an app store, it will more likely to be ranked higher in the platform's app store than an app developed in a cross-platform environment, since it is native to the device. For example: If a company needs specific data collected on a job site and the employees are issued Android devices, then developing a native Android App would make sense.

However, if three native apps need to be built, that is time consuming and expensive since that typically requires hiring three different programmers. To develop a native iOS (Apple) app requires of the use of Swift or Objective C SDKs and programming to be run in an **Integrated Development Environment (IDE)** called Xcode; whereas Android uses an IDE called Android Studio and the native programming language is Java. Finally, to develop for Windows, the development platform is Windows Mobile SDK for use in Visual Studio and code with C++. Native app development kits offer two separate frameworks, one to render the map and one to access the user's location. They each offer different default maps in their native environments. These native maps can be replaced with alternatives such as MapBox, Google Maps in Apple devices, Esri, and more. They also have different location detecting strategies. For example, for iOS development, whenever the CoreLocation Framework is called, users must be alerted that the app will be accessing their location.

Then the developer can decide when to capture this information; when users move at all or when they move 100 meters. Furthermore, it is also possible to collect the heading direction from the user. There are frameworks to most efficiently collect this information based on the sensors that are available in iOS devices. Each of the three platforms has different defaults and SDKs associated with mapping.

3.2.1 Cross Platform Development

Cross platform development is often seen as a welcome alternative to native development because in theory, the app only needs to be built once and then it can be ported to three different operating systems. This should save money and increase the availability of the app to a wider user audience, because these apps can be distributed directly on the three different native apps' stores. The downsides of a cross-platform development strategy are that the application code is usually run through an additional layer of abstraction that can introduce inefficiencies not present in natively developed apps. These inefficiencies may or may not manifest themselves in ways that are noticeable to the end user. The design of the app will likely not look exactly as the end user is used to seeing an app (particularly for iOS where user design regulations are most tightly controlled). The user interface may not look as familiar because the standard drag and drop icons that are offered in the Apple SDK and Android IDE are often not available for use in cross platform development environments.

To develop a cross platform app, a specialized app development suite can be used; for example, QT, PhoneGap, Ionic, React, and Sencha are all different cross platform development frameworks. After the app has been developed using one of these tools, then it still must be opened and customized within the native SDKs to clean up the code and to push the apps to their respective app stores for distribution. In the end, being able to run basic functionality in each of the native SDKs and/or IDE, even when building and distributing a cross platform app, is still required.

3.2.2 (Progressive) Web App Development

Since mobile devices connect to the web, web apps can be accessed through a web browser and the internet. They can be, but do not have to be, downloaded to be saved and reopened on mobile devices. Web apps are arguably the easiest to build and deploy. Typically written using HTML5, CSS, and JavaScript, Progressive Web apps are apps that behave like mobile apps in that they can be installed and saved on a mobile desktop, work offline, deliver push notifications, and access (locational) sensors in mobile devices. They are accessible to any operating system, but they are not distributed on app stores. They are downloaded directly from the web, which sometimes makes them less obvious to locate for audiences who use app stores to download apps. There is no standardization for web apps in terms of user interface design and there are limited SDKs.

4. Mobile Devices, Connecting to the Cloud and to Other Devices

Computing infrastructures, computing approaches, network, and services are all relevant to mobile devices and their effective utilization in any project. In Section 3: Mobile Devices and App Development, front end development considerations were discussed. Backend or database configurations to store and retrieve spatial data are also important to consider. These configuration decisions will influence **interoperability** with other data sources and

devices. This is another important development decision: where and how should data be stored? For mobile devices and their apps to be useful, appropriate data need to be housed somewhere in an organized way. This is often referred to as the back-end infrastructure. Databases are frequently hosted and are an important part of spatial data infrastructure (SDI) (see **Spatial Data Infrastructure**). These SDI need to be organized to efficiently collect and distribute relevant information to and from an end user. Data need to be collected and organized in a way that it is scalable, meaning they can accommodate the number using them, e.g. it doesn't become suddenly and unexpectedly unavailable. The database in place needs to be designed to accommodate the anticipated traffic. Additionally, users need to be able to achieve their goals, be it to make their own maps, send their data to friends, or see their data compared to others.

Mobile devices typically connect to the cloud, where computation can be distributed and conducted remotely. The cloud refers to distributed data centers or server farms, where data are stored, retrieved, and processed (Peterson, 2014). How a database sitting in the cloud is configured will influence what data types are collected and shared via mobile devices. For example, tweets with a geotag are considered vector data, a point with a string field. Tiled maps representing aerial imagery are rendered as rasters and require strong data connections to be saved locally and to load quickly.

As already mentioned, mobile devices offer opportunities to collect field data and connect to the cloud in real time. Data access in real time could help to alleviate data duplication and improve the accuracy of spatial data that are already available. To achieve these goals, an appropriate and effective web architecture needs to be in place. Additionally, appropriate team members must be given read/write privileges to a database. Authorization related to who is able to read and or write to the database and authentication related to verifying the identities of those authorized to access the database, are vital considerations for any SDI.

If a company is collecting big data, often they will be processed and cleansed to be useful. Before relevant information is parsed to a mobile device user, often times the noise in the data needs to be removed. There are several strategies to do deal with big spatial data, one popular algorithm being Map Reduce (see **Spatial MapReduce**).

In addition to connecting to the cloud, mobile devices are used to connect with other devices found in the home or on the job. Interoperability is imperative to the utility of mobile devices. Integrating information and devices that store, disseminate, exchange, manage, display, and analyze the sensing information is what makes mobile devices so valuable to society (Liang, Croitoru, & Tao, 2005). The real power of ubiquitous computing comes from interaction among the devices and networks (Weiser, 1991). Internet of Things (IoT) is of growing interest and mobile devices often act as the control platform for these other devices. This means that mobile devices act as an interface between other devices. For example, a mobile device may act as the interface between an unmanned aerial vehicle (UAV) (see **Unmanned Aerial Systems**) flying in the sky and the flyer (or hopefully helper) who is controlling the onboard camera angle and shutter during flight. A mobile device can also be used pre-flight to plan and set the flight path for a UAV. Mobile devices can be used to connect with IoT in your home, to check on and change the thermostat at your home while you are sitting in the office, and connect with smart refrigerators, toasters, vacuums, and more. Mobile devices afford endless research opportunities both societal and technological.

References

Aterfield, S. (2017). Number of Mobile Subscribers Worldwide hits 5 billion. GSMA.

Gaff, B. (2015). BYOD? OMG! Computer, 48 9(2), 10-11.

Gartner, G., Cartwright, W., & Peterson, M. P. (2007). Location Based Services and TeleCartography. Berlin: Springer.

Goodchild, M. F. (2007). Citizens as sensors: the world of volunteered geography. GeoJournal, 69(4), 211-221.

Goodchild, M. F., Johnston, D. M., Maguire, D. J., & Noronha, V. T. (2004). Distributed and mobile computing. In R. B. McMaster and E. L. Usery (Eds.), A Research Agenda for Geographic Information Science (257-289). Boca Raton, Florida: CRC Press.

Ishikawa, I., Fujiwara, H., Imai, O., & Okabe, A. (2008). Wayfinding with a GPS-based mobile navigation system: A comparison with maps and direct experience. Journal of Environmental Psychology, 28(1), 74-82.

Kerski, J. J. (2013). Core Tenets for Success: Teaching with GIS. Blog.

Liang, S., Croitoru, A., & Tao, C. (2005). A distributed geospatial infrastructure for Sensor Web. Computers and Geosciences, 31(2), 221-231.

Miluzzo, E., Lane, N. D., Fodor, K., Peterson, R., Lu, H., Musolesi, M., Eisenman, S. B., Zheng, X., Campbell, A. T. (2008). Sensing Meets Mobile Social Networks: The Design, Implementation and Evaluation of the CenceMe App. In SenSys '08: Proceedings of the 6th ACM Conference on Embedded Network Sensor Systems (337-350). ACM.

Morita, T. (2005). A working framework of ubiquitous mapping. In Proceedings of the 22nd International Cartographic Conference.

Morita, T. (2007). Theory and development of research in ubiquitous mapping. In: Gartner, G., Cartwright, W., Peterson, M.P. (eds) Location Based Services and TeleCartography. Lecture Notes in Geoinformation and Cartography. Springer, Berlin, Heidelberg.

Muehlenhaus, I. (2013). The design and composition of persuasive maps. Cartography and Geographic Information Science, 40(5), 401-414.

Muhlhauser, M., & Gurevych, I. (2009). Introduction to Ubiquitous Computing. In P. Zaphiris and C. S. Ang (Eds.), Human Computer Interaction (Vol. I, pp. 1-19). Hershey, PA: Information Science Reference.

Nield, D. (2017). All the Sensors in Your Smartphone, and How They Work. Gizmodo Blog.

Pallotta, V., Bruegger, P., & Hirsbrunner, B. (2009). Kinetic User Interfaces: Physical Embodied Interaction with Mobile Ubiquitous Computing Systems. In E. Szewczak

(Ed.), Selected Readings on the Human Side of Information Technology (p. 154-190). IGI Global.

Peterson, M. P. (2014). Mapping in the Cloud. New York and London: Guilford Press.

Puvvala, A., Dutta, A., Roy, R., & Seetharaman, P. (2016). Mobile Application Developers' Platform Choice Model. 2016 49th Hawaii International Conference on System Sciences (HICSS), Koloa, HI, USA, 2016, pp. 5721-5730.

Ricker, B., Hedley, N., & Daniel, S. (2014). Fuzzy boundaries: Hybridizing Location-based Services, Volunteered Geographic Information and Geovisualization Literature. Geography Compass, 8(7), 490–504.

Satyanarayanan, M. (2001). Pervasive Computing: Vision and Challenges. IEEE Personal Communications, 8(4), 10-17.

Schuurman, N. C. (2000). Trouble in the Heartland: GIS and its Critics in the 1990's. Progress in Human Geography, 24(4), 569-590.

Schuurman, N. C. (2006). Formalization Matters: Critical GIS and Ontology Research. Annals of the Association of American Geographers, 96(4), 726-739.

Sieber, R. E. (2004). Rewiring for a GIS/2. Cartographica, 39(1), 25-39.

Wang, D., Park, S., & Fesenmaier, D. R. (2012). The Role of Smartphones in Mediating the Touristic Experience. Journal of Travel Research, 51(4), 371-387.

Weiser, M. (1991). The Computer for the 21st Century. Scientific America, 265(3), 66-75.