

# Toy Computing Background

Laura Rafferty, Brad Kroese and Patrick C. K. Hung

**Abstract** The purpose of this chapter is to provide a background on the fundamental concepts of Toy Computing, including mobile services, physical computing, and augmented reality. It will also present some examples of toy computing products currently on the market. The chapter will also provide a background on security and privacy and relevant research works on these topics in order to provide the necessary foundations for the rest of this book.

**Keywords** Toy Computing · Mobile Services · Physical Computing · Location Privacy

## Section 1. Toy Computing Background

### *What is Toy Computing?*

Mobile devices have become prevalent in many aspects of our daily lives. The reason for this is the portability and flexibility of the devices which can easily support applications developed for a wide range of uses. More recently, another use for mobile devices has been introduced in the area of toys and gaming. Toy companies such as Hasbro, Mattel, and Tech4Kids have released toys that integrate with mobile platforms, providing new capabilities and add-ons to traditional functionality (D’Hooge und Goldstein 2001). These have been referred to as *Augmented* (Hinske & Langheinrich, Managing Augmented Toy Environments—A New Perspective for Smart Space Management 2007), *Interactive* (Luckin et al. 2003), or *Smart Toys* (Plowman und Luckin, February 2004), because they include sensory capabilities to allow them to detect and interact with their environment. Related fields include

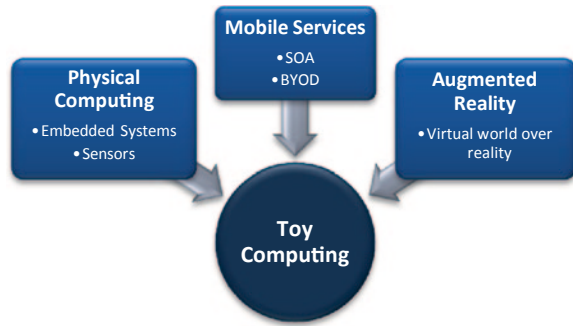
---

L. Rafferty (✉) · B. Kroese · P. C. K. Hung  
Faculty of Business and IT, University of Ontario Institute of Technology, Oshawa, Canada  
e-mail: Laura.Rafferty@uoit.ca

B. Kroese  
e-mail: brad\_kroese@hotmail.com

P. C. K. Hung  
e-mail: Patrick.Hung@uoit.ca

**Fig. 1** Toy computing components



physical computing, mobile services, context and location-based services, and augmented reality. At its most basic level, a toy computing system can be identified as a toy equipped with sensory technology, mobile computing power, and communication capabilities (Hinske & Langheinrich, *Managing Augmented Toy Environments—A New Perspective for Smart Space Management* 2007). This differs from a traditional electronic toy in how it incorporates a mobile component, whereas traditional electronic toys are isolated to their own proprietary platform. The two basic components that make up a toy computing system include a) the physical component, which is similar to a traditional toy, and b) the mobile component, a smartphone or tablet running an application to provide services to the user/toy.

The physical component of a toy computing system observes almost the same overall characteristics as a traditional toy, with the potential addition of embedded systems, networking capabilities or sensors designed to communicate in some way with the mobile component. This physical component can take the form of any traditional toy, such as a blaster (Tech4Kids 2013), block (ChineseCUBES 2014), or stuffed animal (Woollaston 2014). The physical component may or may not contain embedded systems or networking capabilities; however it must be able to interact in some way with the mobile component. An interaction can be physical, visual, auditory, or through networking such as Bluetooth, RFID or Wi-Fi.

In this configuration, the mobile device takes on the position as the primary computing device of the system. This includes the CPU, memory, sensory input, and output. The mobile component will run an application which operates in collaboration with the physical component to provide services to the user based on their interactions with the physical component. For the purpose of this work, we will be concentrating on *Toy Computing* from a mobile services perspective. There is a multitude of built-in sensory capabilities on mobile devices, which provide a new wave of opportunities for human computer interaction and personalized context-aware services. Depending on the toy, the sensory capabilities may either be located on the physical component, the mobile component, or both. Figure 1 illustrates the relationship between physical computing and mobile services to form a toy computing environment.

### Examples of Toy Computing Products

Toy computing is quickly gaining popularity in the toy industry. These toys have a wide variety of categories including toy blasters, language blocks for educational

**Fig. 2** Tek Recon “Havoc” blaster with mobile device mount. (Adapted from [www.tekrecon.com](http://www.tekrecon.com))



purposes, and methods of communication for children. Below are some examples of popular toy computing products currently on the market.

### **Tek Recon**

Tek Tecon (Tech4Kids 2013) is a line of toy blasters developed by Tech4Kids, marketed to children aged 8 years and up in 2013. While this product features a physical component identical in concept to a traditional toy blaster, the novelty is the ability to integrate with a mobile device. Referring to Fig. 2, the Tek Recon blaster features a mount on top where a smartphone is inserted. A mobile application has been developed by Tech4Kids which operates in collaboration with the physical blaster to augment traditional blaster-based games. The application provides several functionalities including a scope, which uses the smartphone camera to display what is in front of the user with additional features overlaid on top, such as ammunition, score, radio, and a GPS location map of other players. The application has networking functionality to create and join games with friends over a LAN or mobile network. The user is also required to create an account online, where the scores and account information are stored.

### **Sphero**

Another recent toy computing product in the industry is Sphero (Sphero 2014), first introduced in 2011 by Orbotix, which then released subsequent versions, Sphero 2.0 in 2013 and Sphero Ollie in 2014. Referring to Fig. 3, Sphero is a robotic ball which can be controlled and programmed through the user's smartphone or tablet. There are over 30 apps available for Sphero, most of which are games, while others

**Fig. 3** Sphero robotic ball. (Adapted from [www.think-geek.com](http://www.think-geek.com))





**Fig. 4** ChineseCUBES. (Adapted from [www.chinesecubes.com](http://www.chinesecubes.com))

are focused on education. This product is marketed not only to children and can be appropriate for any age group. While the physical ball component is a very simple and traditional concept, the capabilities of the toy increase substantially with the inclusion of robotics and a mobile device. The Sphero ball has wireless networking capabilities, an accelerometer and gyroscope, rolls in every direction, and glows different colors. Sphero can be programmed by the user through an app called Sphe-ro Macrolab, which includes a set of predefined macros, and more advanced users can use another app called orbBasic to program in a language based on BASIC.

### ChineseCUBES

ChineseCUBES (ChineseCUBES 2014) is a toy computing product first introduced in 2011 which combines augmented reality technology with physical blocks to help the user to learn Chinese characters. Referring to Fig. 4, the AR markers on the cubes are arranged in a certain order and detected by the webcam to create an interactive audio/visual experience with the software on the computer or mobile device. The software includes multiple features such as interactive stories, lessons and videos. The compute or mobile device does all of the sensing and processing in this scenario, and the physical cube components are entirely traditional.

### Toy Mail

Toy Mail (Toymail Co. LLC 2014) is a toy introduced in 2013 which can connect to the user's home WiFi network and interact with the free Toy Mail mobile app. Once the app is installed on a mobile device, the user can record a message which will be sent to the toy. When a message is received, the toy (as shown in Fig. 5) will make a snort, wheeze, or whine sound to let the user know that they have received a message, which can then be played and replied to.

### Other

Toy computing has been also been developed for a wide range of purposes such as language learning (Lee und Doh 2013), early childhood education, and for children

**Fig. 5** Toy mail character, “Snort.” (Adapted from [www.toymail.co](http://www.toymail.co))



with ADHD and autism. For example, Auti is a socially assistive robotic toy which encourages physical and verbal interactions in children with autism (Andreae et al. 2014). Educational toys such as roBlocks and SmartTile encourage children to learn about robotics and programming while they play (Gross und Eisenberg 2007). There has also been research on monitoring children’s developmental progress using augmented toys and activity recognition (Westeyn et al. 2012).

### Design Guidelines for Toy Computing

Hinske et al. (Hinske et al. Towards Guidelines for Designing Augmented Toy Environments 2008) provide a Summary of Design Guidelines for Integrating Pervasive Computing Technology into (Traditional) Toys:

1. The technological enhancement must have an added value.
2. Specify what actions/tasks are to be supported.
3. Let the focus remain on the toy and the interaction itself, not the technology.
4. Integrate the technology in such a way that it is unobtrusive, if not completely invisible.
5. Toys should be still usable (in the “traditional” way) even if the technology is switched off or not working.
6. Tightly intertwine design and implementation
7. The technology should be reliable, durable, and safe.
8. Offer immediate and continuous feedback.
9. The added technology should support the high dynamics of play environments.
10. Employ an iterative development process, including rapid prototyping and testing.

The above guidelines reinforce that the integration of pervasive computing technology (i.e. in the context of toy computing) should provide added value and seamless integration with the physical toy component. Further, the technology should be reliable, durable, and safe. From the perspective of privacy, this introduces a need for a privacy preserving framework which protects the child from privacy threats while not taking away from the play experience by introducing obtrusive policies.

## *Physical Computing*

Physical computing is a branch of computing which involves the integration of computing technology into a physical device which interacts with its environment. This is similar to the concept pervasive or ubiquitous computing, in which the computing device establishes itself into the users' daily physical activities. A pervasive computing environment is an information-enhanced physical space, not a virtual environment that exists to store and run software (Saha 2003); where the design of the system takes a human body as a given, and attempt to design within the limits of its expression (Sheth et al. 2013).

The distinction between physical and pervasive computing is that physical computing has more of a focus on the physical objects involved rather than completely seamless interaction. In toy computing the physical toy component is an active part of the user experience, whereas in pervasive computing there would be little to no physical component and the system works seamlessly with everyday activities. One of the main characteristics of both pervasive and physical computing devices is its ability to perceive context information on the surrounding environment in order to react accordingly (Saha 2003). This perception is done through sensors on the device such as a microphone, camera, or accelerometer. Perception of this context information is fundamental to the device's ability to make timely and context-sensitive decisions.

As mentioned previously, the physical component of the toy computing environment would be the traditional toy itself, which will be complemented with embedded systems or sensor technology which communicates with the mobile application. In this system, personalized services are provided to the user based on context data collected and inferred through sensors and other environment data. With the pervasiveness of modern mobile devices, vast amounts of information can be collected and inferred about the user and their environment. Physical computing often involves a networked environment, which introduces privacy and security issues, particularly related to the context information the devices are processing. While the toy is the physical component in this system, the mobile device is what provides computing functionality and sensory perception, as described in the next section.

Physical computing introduces physical objects as interface components. The examples in the previous section demonstrate this with a toy blaster, ball, and cubes, which are all used as an interface similar to a traditional toy, but with enhanced interactive capabilities. As seen in Fig. 6, the Sphero robotic ball acts as the physical interface component in a physical, toy computing environment.

**Fig. 6** Sphero robotic ball as a physical toy component. (Adapted from [www.gosphero.com](http://www.gosphero.com))



## Sensors

Modern mobile devices are created with a variety of sensory capabilities. In a toy computing environment, developers may embed sensors into the physical toy component, or take advantage of sensors already built into the mobile device. Through these sensors, motion and other data can be detected in a number of ways. Sensors can be categorized into three different types: motion sensors, position sensors, and environment sensors (Google [n.d.](#)). Below is an analysis on some of the different types of data that can be gathered from these sensors.

- **Motion Sensors:** Motion sensors capture the physical motions of a device. Mobile devices can include a number of sensors for measuring motion including an accelerometer, gyroscope, magnetometer, barometer, gravity, linear acceleration, and rotation vector. Motion is commonly represented through 6- or 9-axis sensor system (3-axis magnetometer, 3-axis gyroscope, 3-axis accelerometer). These types of sensors are commonly used for a variety of mobile applications such as games, as a way for the user to interact with the application (e.g. angling the device left or right to turn the character in a game). They have also been commonly used in fitness applications for tracking steps and calories lost during a walk, run, or jog. A popular example of this is *Zombies, Run!* (Six to Start [n.d.](#)), a mobile game application which takes motion sensor input while a user is running or jogging. The application provides missions for the user to complete by meeting certain fitness goals which correlate with the storyline.
- **Position Sensors:** Position sensors are also very popular in mobile systems. Some examples of these types of sensors include geomagnetic field sensor, proximity sensor, and GPS. Position sensors, particularly proximity and GPS, are very useful for mobile and toy computing due to the portability of mobile devices. Many applications use location-based services which use position sensors on the device to provide recommendations relevant to the location of the user. Some examples of this include Yelp (Yelp [2015](#)), UrbanSpoon (zomato [2015](#)), which allow users to read and post reviews of nearby restaurants and



other establishments. Other applications such as social media applications, Instagram (Instagram 2015) and Facebook (Facebook 2015), use position sensors to allow users to geotag their location along with their posts.

- **Environment Sensors:** Sometimes it is useful for an application to be able to detect data about the surrounding environment. While this is not used as widely as motion and position sensors in the mobile and toy computing environment, these sensors do have a lot of very useful applications in agriculture, health care, security systems, aeronautics. Types of environment sensors include sensors for relative ambient humidity, luminance, ambient pressure, and ambient temperature. The most popular environment sensors in the context of a mobile environment are probably luminance and sound sensors. An example of an application that uses environment sensors is PressureNet (Cumulonimbus 2015), an Android application that measures atmospheric pressure using the atmospheric sensors built into most Android phones. Most smartphones use luminance sensors to adjust screen brightness based on lighting conditions.

## Wireless Communication Technologies

While physical computing environment collects environment data through sensors, the data collected often needs to be communicated to a service provider or other devices over a wireless network. The service provider may be located on the user's mobile device, or another device on the local or wide-area network. Possible types of wireless communication technologies used in a mobile toy computing environment include: RFID, NFC, Bluetooth, WiFi, GSM, and UMTS/3GSM.

## Context Data

Data observed and collected through sensors gather context on the user and their environment. Context is defined succinctly by Dey and Abowd (Dey und Abowd 1999) as "any information that can be used to characterize the situation of an entity." Schilit et al. (Schilit et al. 1994) defined context as location, identities of nearby people and objects, and changes to those objects. Zimmermann et al. further categorized the elements for describing context information into five categories: individuality, activity, location, time, and relations. Individuality is personal information about a user, activity is data regarding physical activity, location is the GPS location, time is discrete time, and relations are inferences between two or more pieces of context data. In a context-aware system, services are provided to the user based on what is relevant to their context. Recent advances in mobile technology open up great opportunity for the collection and processing of context data in valuable ways. There are many types of private context data that can be collected via a mobile application. The collection of this data allows applications to adapt to the user's environment and personalize services accordingly.



## Types of Context Data

Mobile devices can capture a user's physical activity state (e.g. walking, standing, running, etc.) and store personalized information (e.g. location, activity patterns, etc.). This data is referred to as context data; data that is collected on the user and their environment. This data can be collected from sensors, provided explicitly by the user, or observed, such as the time of an event. Personal data can come in many forms including browsing history, friends list, and location information. Some other examples of relevant context information include (Schmidt 2005): Verbal context, roles of communication partners, goals of the communication/individuals, local environment, social environment (who is there), and physical and chemical environment. Information can be volunteered (e.g. profile data provided directly by the user) or observed (e.g. location data detected from GPS). Often, private information may seem trivial and not perceived as very sensitive to the user, while in practice it can actually reveal a large amount of personal information about them. The World Economic Forum (World Economic Forum 2011) defines three types of context data, as categorized by the way it is collected: volunteered, observed, and inferred:

- **Volunteered Data:** data that is explicitly provided by the user. This can include personal profile information or preference settings.
- **Observed Data:** data not directly given by the user, but is detected by the device/application often through a sensor. Some examples of observed data include GPS location and time.
- **Inferred Data:** data deduced based on analysis of a combination of volunteered and/or observed data (e.g. where a user is likely to be going based on typical behavior). A lot can be interpreted on a user and their environment through inferences based on collected data. There is great value on this inferred data that would not be explicitly provided by the user.

Volunteered and observed data can be analyzed to infer significant amounts of personal information about the user. For example, forecasting trip destinations based on data from driving habits (Dewri et al. 2013). Collected data is the basis for many valuable context-aware services, which provide custom content or services to the user based on what is most likely to be useful to them.

## Privacy Concerns

With all of this in mind, privacy is a growing concern among many users of mobile devices. While many users appreciate the value of targeted services, they still express concern over how their data is collected and managed without their knowledge. Cherubini et al. (Cherubini et al. 2011) identify privacy as a barrier to the adoption of mobile phone context services. 70% of consumers say it is important to know exactly what personal information is being collected and shared (MEF 2013), while 92% of users expressed concern about applications collecting personal information without their consent (Futuresight 2011). Mobile applications have adapted

countless services to better analyze context data and provide custom services that will bring the most value to a user based on what they are most likely to need.

While allowing context data to be collected for services can prove to be of great benefit to users, there is an ongoing tradeoff between utility and privacy (Chakraborty et al. 2013b). In this physical mobile and pervasive environment, the timely delivery of services is fundamental. The amount of information collected often results in a tradeoff required between disclosing sensitive data and receiving context-aware services. In order to provide the most relevant services to the user, more personal and context information must be collected, which raises concerns of privacy. For example, a service can send special promotions and coupons to a user depending on what is most relevant to them. In order to provide the most relevant promotions, the service will need to collect certain context data such as their location, and also potential profile information such as age and gender to help to determine what their interests may be based on demographic. To gain even more context of the user, the application may collect and retain historical data on the user such as previous movement patterns, to determine where they are likely to be at certain times, if they are travelling, or previous interactions with the application such as which promotions they had previously been interested in. In this example, it is clear that the more information is collected on the user, the more relevant services can be provided to them. However, the user may not be comfortable with the level of data that is collected and inferred on them. An application knowing where you are and what you are likely to be doing at any given time is likely to raise concern with users.

For this reason, context data is at the core of privacy concerns with many mobile applications. Privacy goals must be defined to ensure private data is managed responsibly. Further, detailed analysis is required to ensure that the user's sensitive behavior cannot be inferred based on collected data. There have been many solutions which aim to preserve the privacy of sensitive context data, as will be described further below. There are countless types of data that can be collected from a mobile device that must be considered when evaluating the scope of privacy. This is true of collected sensory data as described above, and also from within other applications, sensitive data can be collected such as a user's profile information, contact list, or calendar. All of this information can be collected and analyzed to determine context information about the user.

## **Location**

Location data can be defined as data representing where a user is physically located. Location is one of the most prominent types of data for context-based services, existing as a key parameter to define context (Schmidt et al. 1999). A user's location, combined with other context and historical data, can be used to infer an extensive amount of information including actions, speed, direction, and movement patterns. Location data can be collected from the device through GPS, WiFi, or mobile network satellite. It can also be inferred from other information such as IP address, although this can be inaccurate (e.g. in the case of a proxy).

Location is defined succinctly by Merriam-Webster (Merriam-Webster [n.d.](#)) as “a place or position.” This definition has been extended by the National Geographic Encyclopedia (National Geographic [n.d.](#)) to establish three different types of representing location: absolute location, relative location, and type of location as follows:

- **Absolute Location**—the location expressed in a range or exact GPS coordinates of latitude and longitude. The absolute location can be expressed as coarse or fine; for example, an entire country, city, block, or exact coordinates.
- **Relative Location**—the location relative to another entity as a reference point; for example, a relative location can be expressed as the distance between User A and User B, or distance between User A and Device C, or User A and location D.
- **Type of location**—the location expressed in an assigned category. Some examples of this could be home, office, street, mall, or restaurant.

Generally, location is represented as a 3-dimensional vector of GPS coordinates (latitude, longitude), and altitude (optional). A location event also includes a time-stamp. Android’s **GpsLocation** data structure represents the location of the device with the following data fields [1]:

- Size
- Flags
- Latitude
- Longitude
- Altitude
- Speed
- Bearing
- Accuracy
- Timestamp

Different ways of collecting location information can be more accurate than others. For example, there is GPS-based location (fine) or Network-based location (coarse) (Android [2015](#)):

- **ACCESS\_COARSE\_LOCATION** (Network-based)—allows an app to access approximate location derived only from network location sources (cell towers and Wi-Fi). This method varies in accuracy from 50 m in urban areas, and several kilometers in rural areas with less cell tower coverage.
- **ACCESS\_FINE\_LOCATION** (GPS-based)—allows an app to access precise location from location sources such as cell towers and Wi-Fi, and also the user’s GPS coordinates provided from their device. Accuracy for fine location using GPS is fairly accurate from 2 to 20 m.

While a huge number of mobile applications request access to user location data, this is one of the most sensitive types of context-data. The incredible amount of information that can be gathered from a user based on their location is immense. Whalen et al. (Whalen [2011](#)) discuss some of the current privacy issues in mobile devices mainly focusing on the storing and transmitting of sensitive location based information over extended periods. This research states that a large amount of users

do not even know that such information is being stored, and in some cases, still happens even if the user has explicitly restricted such data to be collected. This goes against the privacy principle of having the users consent before collecting this information. Another violation of privacy principles that is discussed in this paper is the amount of data that is being collected is much more plentiful, accurate, and goes on for a lot longer than it needs to. One of the causes of this disconnect is that most of the permissions for this information collection is buried in lengthy policies that users rarely read, and is enabled by default. It is very important to protect this information, having access to such information not only shows where we have been but it can be used to predict where we will be tomorrow, and that introduces a lot more security concerns. Patil et al. (Patil et al. 2012) go into further detail about the widespread usage of location data collection in mobile services and their interaction with social networking services. The paper details an online study on 362 participants to understand the preferences of users of location services. The majority of users expressed that their main incentive for using of these services was for social networking purposes. A number of users in this study (25 %) also indicated that they have regretted sharing their location on at least one occasion.

Location privacy is a huge concern in the mobile and wireless environment. While it can appear trivial, location-based data can infer a lot of sensitive information about a user, including their activities, habits, interests, and personal relationships. Inference attacks are possible, such as knowing when a user will be somewhere based on movement patterns and historical activity. This can potentially put a user at risk. Often, location-aware services do not require knowledge of the exact location, but rather, could provide just as valuable services with an approximation of the user's location (Pandit und Kumar 2012).

These research works identify a great need for location privacy management and enforcement in mobile services. While toy computing has become a recent development in the union of mobile service and toys, research on safety and privacy guidelines for toy computing seems to have been largely overlooked. To the best of our knowledge, there is a gap in the research area of location privacy in the context of toy computing. First, there is no formal model for enforcing privacy or location privacy in particular, for children using such toys.

## Section 2. Mobile Services

Mobile devices, such as smartphones, tablets, and e-readers, have become increasingly popular in recent years, successfully integrating themselves into the lives of many users. A recent poll of 5000 people by TIME magazine reveals that 54 % of respondents check their mobile device at least once an hour (TIME and Qualcomm 2012), while another study for GSMA shows that 68 % of participants identified themselves as users of mobile internet/apps, with 38 % of this subset considering themselves to be heavy users (Futuresight 2011). The immense popularity of these devices can be explained by their personal, portable, and pervasive nature. These

characteristics create a unique platform for services, particularly those based on context data. Often, mobile devices will have one single primary user. The portability of a mobile device makes it possible for a user to carry it with them wherever they go, making it a highly personal device as well. Mobile devices are also designed to be easy and fast to use, and easily connect to networks, allowing the user to always stay connected to data and services.

Mobile devices use mobile services, which are services accessible through a mobile network. Mobile services, like Web services, use Service Oriented Architecture (SOA) as described in Sect. 2.2.3.1. Mobile services can be context-aware, gathering context information from the mobile device, and providing relevant personalized services based on the context. To gather context information, a context-aware service can either listen for events sent by a context provider, or query the context provider. Gu et al. (Gu et al. 2004) propose a middleware for building context-aware mobile services, using a Service Locating Service to allow entities to locate different context providers. However, this model does not consider privacy preferences of the user.

### ***Mobile Games and Location-Based Services***

Location-based services, also known as location-aware mobile services, have become widely popular to provide information such as travel information, shopping, entertainment, and event information. Location-based services have been defined by Duri et al. (Duri et al. 2001) as “services in which the location of a person or an object is used to shape or focus the application or service.” Pura (Pura 2005) identifies location as one of the most promising applications of mobile commerce, due to the ability to allow service providers to offer customized services based on context and resulting in increased perceived value and loyalty of customers.

The mobile application industry has observed a widespread adoption of mobile game applications. This has been successful due to factors such as increased mobility and social network integration (Baber und Westmancott 2004). Location-based services have also been used in applications for games. The popular mobile game Angry Birds (Rovio 2015) has a location-based feature which allows users to compete with other based on a leader board associated with their location. MyTown (Booyah 2015) is another mobile game, reminiscent of Monopoly, where users can check in to a physical location, buy and sell properties, and collect rent from other players who check into the same location.

Kaasinen (Kaasinen 2003) conducted a study to investigate user needs for location-aware mobile services:

- Contents: topical up to date information, comprehensive relevant information, interaction (user is moving and can only provide limited interaction to device), push information based on both location and personalization, detailed search options, planning vs. spontaneity.
- Personalization: personal options and contents, user-generated content.

- Seamless service entities: consistency, seamless solutions to support the whole user activity.
- Privacy: the right to locate, use, store, and forward the location. Privacy requirements are based on legislation and social regulation. The paper also identifies P3P as a potential approach to manage user privacy preferences and compare them to the location-aware service's privacy practices.

### ***Bring Your Own Device (BYOD)***

Mobile services follow *Bring Your Own Device* (BYOD) architecture, meaning that the user has their own personal mobile device to run the service from. Mobile services thus need to be flexible and consider a variety of different devices. While the term BYOD is typically used to refer to employees bringing personal devices to a work environment, the same general idea is involved in any mobile services scenario. Mobile applications must operate in a controlled environment and must protect data and resources from other untrusted applications that may be running on the device. Further, the introduction of unregulated mobile devices onto a network can result in loss of control, data leaks, and potential network loss (Gartner 2014). BYOD can introduce complications when it comes to investigation in the case of a security breach. This can be made simpler through thorough planning of policies and contracts indicating employee and employer (or in a more general case, user and service provider) rights (Beckett 2014).

A toy computing environment considers several properties of BYOD, although outside of a corporate environment. For the purpose of this work, we will be considering the following BYOD characteristics:

1. The user's mobile device is untrusted.
2. The mobile application is operating on top of this untrusted device.

The objective of BYOD is to isolate business applications from the rest of the system. This means isolation from other applications running on the personal device (Disterer und Kleiner 2013).

### ***Mobile Service Architecture***

Figure 7 illustrates a multi-layered model which illustrates the relationship between the conceptual, logical, and language layers of mobile services. This framework has been adapted from the Web services logical model presented by Hung et al. (Hung et al. Towards Standardized Web Services Privacy Technologies 2004). This model is an extension of traditional Service-Oriented Architecture to include layers for privacy-related access control, and also an End-Point Device Profile for mobile devices. Each layer will be discussed in the subsequent sections.

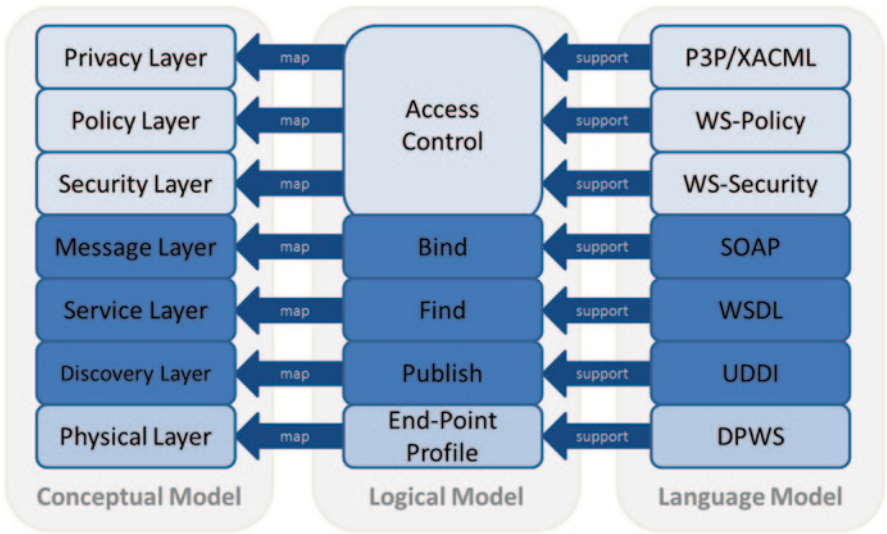


Fig. 7 Mapping between different models and layers. (Adapted from Hung et al. Towards Standardized Web Services Privacy Technologies 2004)

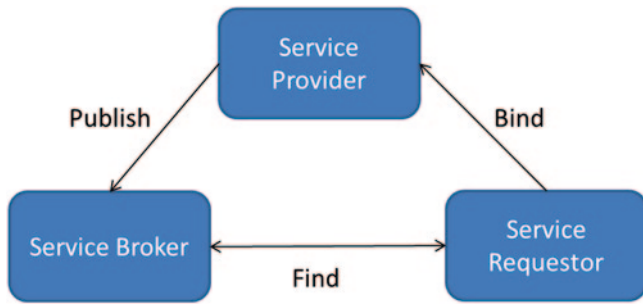
Service Oriented Architecture (SOA)

A theoretical model for Web services has been defined in Service Oriented Architecture (SOA). In our conceptual model, SOA consists of the message layer, service layer, and discovery layer. W3C defines SOA as a form of distributed systems architecture which typically maintains the following six properties (W3C 2004): (1) The architecture is defined in a *logical view*, in terms of what it does. (2) The *message orientation* property expresses how the service is defined in terms of the messages exchanged between provider and requester agents, rather than the internal architecture behind the provider’s services. (3) *Description orientation* enforces that a service is described by machine-processable metadata. (4) SOA messages are also *granular* and (5) *Platform neutral*, meaning services tend to use a small number of operations with large and complex messages, which are in a standardized platform-neutral format (ex. XML). Lastly, these services often tend to be oriented towards use over a *network*.

As seen in Fig. 8, SOA consists of three entities: service provider, service requester, and service broker, and 3 operations: publish, find, and bind.

1. **Discovery Layer (publish):** In the model, the service provider will first “publish” details of its service (description and location) to the service broker, who saves it to the Universal Description Discovery Integration (UDDI) registry. UDDI is an OASIS standard which provides a directory of services available from each service provider.
2. **Service Layer (find):** The service requester queries the service broker with the “find” operation to find the service it is looking for, who will then return





**Fig. 8** Service Oriented Architecture (SOA)

the details of the service. This layer uses Web Services Description Language (WSDL), an XML-based W3C standard for describing network services as a set of endpoints operating on messages (W3C 2001).

3. **Message Layer (bind):** Finally, the requester uses the connection details to “bind” to the provider and receive services. The message layer uses Simple Object Access Protocol (SOAP), an XML-based protocol for request and response messages in web services.

SOA is also been explored with mobile devices, with the mobile host acting as a service provider. The authors of (Fonseca et al. 2009) discuss the mobile host as a provider of services with SOA. It overviews the limitations with WS standards specifications on mobile cloud deployed services, as well as provide an architecture for supporting mobile clients in this environment. It has not been demonstrated in a real-life environment yet, although they are working on deploying it on Amazon EC2. Service-Oriented Architecture for Devices (SOA4D) (Fusion Forge n.d.) is an open-source initiative aimed at the development of service-oriented software components (SOAP, WS-\*, etc.) to fit the needs of embedded devices. SOA4D implements Device Profile for Web Service (DPWS), a specification designed for secure Web service communications on resource-constrained devices, as further described below.

### Device Profile for Web Services (DPWS)

When software is running on any device, the application will need to communicate with other services whether they are internal or external (over a network). The Device Profile for Web Services (DPWS) (OASIS 2009) follows the SOA framework for automatic device and service discovery for networked embedded devices. DPWS offers a standardized device representation of services on a network and this allows for access to a set of built-in services such as secure accessing of meta-data and exchange services by utilizing WS protocols. In other words, DPWS defines a minimal set of implementation constraints to enable secure Web service messaging, discovery, description, control, and eventing on resource-constrained

endpoints (OASIS 2009). The specification permits the definition of services for mobile devices considering the peer-to-peer direct communication between them that combine several devices as Service Oriented Architecture (SOA). DPWS allows sending secure messages to and from services, dynamically discovering a service, describing a service, subscribing to, and receiving events from a service.

In Web Services terms, a *profile* is a set of guidelines for how to use Web services technologies for a given purpose or application. Web services standards allow implementers to choose from a variety of message representations, text encodings, transport protocols, and other options, some of which are not interoperable. By constraining these decisions, profiles ensure that conforming implementations will work well together. DPWS is a profile developed by Microsoft and others for communication with and among networked devices and peripherals. The DPWS library for the.NET Micro Framework is not a full Web services implementation but a lightweight subset with only the functionality needed to support DPWS on a device (Microsoft 2007). DPWS was built on the foundation of existing web services (WS) and as such uses many common specifications such as XML, SOAP, WS-\*, WSDL and Message Transmission Optimization Mechanism (MTOM). DPWS defines two main types of services that are run by devices: hosting services, and hosted services (Microsoft 2007). Devices can be DPWS clients (invoking hosted services on devices), servers (providing hosting services), or both. DPWS for the.NET Micro Framework supports devices in either role or both simultaneously. Hosted services are the services that the device has, and depends on their hosting service for discovery. Hosting services allow other devices to use, subscribe and obtain metadata of the given services. DPWS defines the extensions required for using services in mobile devices, taking in account their specific constraints. A DPWS enabled device has access to provided functionality such as: the discovery of other, utilizing WSDL to describe a Web service, service subscription, and secure sending of messages, given that the other device also utilizes DPWS.

The Web Services for Devices (WS4D) (Want 2006) framework is an extension of DPWS to bring SOA and Web services technology to industrial automation, home entertainment, automotive systems and telecommunication systems. There have been ongoing initiatives to connect internet technologies and web services to resource-constrained devices in ad-hoc networks while conserving interoperability. WS4D provides technologies for easy setup and management of network-connected devices in distributed embedded systems (Golatoski et al. n.d.). Araujo and Siqueira (Araujo und Siqueira 2009) used WS4D to implement a DPWS Device Service Bus (DSB), establishing a Device Tunnel to deal with virtual devices and services.

Pohlsen et al. (Pohlsen et al. 2009) present a plug-and-play architecture for connecting medical devices through DPWS, using WS-Discovery protocol. Unlike traditional Web service architectures, the authors propose using a WS-Discovery proxy server rather than a UDDI server, to better meet the requirements of resource constrained devices. Further, the work uses SOAP-over-UDP (User Datagram Protocol) for multicast messaging, as included in DPWS. El Kaed et al. (El Kaed et al. 2011) present an implementation to interoperably connect Universal Plug and Play

(UPnP) and DPWS smart home devices such as a TV, printer, and light bulb. DPWS does not support fine-grained security requirements, direct authentication between devices without a third party, and does not propose a comprehensive authorization concept (Unger et al. 2010). All of these works present the foundation technologies for this research work. To the best of our knowledge, there is no unified framework for enforcement for location privacy in mobile services for toy computing.

## Section 3. Privacy and Access Control

### *Introduction to Privacy*

When it comes to any information technology, privacy and security are at the core of ensuring that goals are achieved effectively and without compromise of personal data. The three concerns of security are confidentiality, integrity, and availability. Confidentiality means that access to information is restricted only to intended parties. Integrity means that data is accurate and consistent and has not been tampered with, while availability means that resources and data remain available when needed by the legitimate parties. A foundation of security is required for privacy.

Information privacy is defined by Hung and Cheng (Hung and Cheng, Privacy, 2009) as “an individual’s right to determine how, when, and to what extent information about the self will be released to another person or to an organization.” In particular, personally identifiable information is any type of information that can be linked to an individual, including their activities, preferences, history, conversations, etc. In a mobile environment, personally identifiable information is also likely to be gathered from context data, as described in the previous section. Information privacy goals can be achieved through privacy preserving mechanisms such as access control, privacy policies, and privacy preferences.

### *Walled Garden*

In a toy computing environment, the concern is with the privacy of the user and that access to resources that can reveal context data are limited to the toy/game service application, and only used for purposes which comply with privacy regulations and are acceptable to the user. While the toy computing environment follows a BYOD model, it is required to identify a privacy preserving BYOD architecture. The Whitehouse has outlined three high-level means of implementing a BYOD program (Digital Services Advisory Group and Federal Chief Information Officers Council, United States of America 2012):

- **Virtualization:** Provide remote access to computing resources so that no data or corporate application processing is stored or conducted on the personal device;

- **Walled garden:** Contain data or corporate application processing within a secure application on the personal device so that it is segregated from personal data;
- **Limited separation:** Allow comingled corporate and personal data and/or application processing on the personal device with policies enacted to ensure minimum security controls are still satisfied.

In this context, a virtualized model would not be feasible or able achieve the privacy goals in a toy computing environment. However, privacy and security can be protected in a toy computing environment through the Walled Garden or Limited Separation approach. Walled Garden is a sandboxed and separated model which allows for processing to take place within a secure application which is separate from other applications and data. Limited separation allows the personal and corporate data and processing to comeingle together, but enacts policies to protect the data and resources. Limited separation approach raises the issue of having a trusted mechanism for policy enforcement. To our best knowledge, not many research works are discussing the concept of Walled Garden.

## *Access Control*

Access control is a security and privacy concept which aims to protect access to resources or data. The purpose of access control is to limit the actions or operations that a legitimate user can perform (Sandhu und Samarati 1994). There are two parts related to access control: the access decision, and the access enforcement. Access decisions can vary but the most basic are permit or deny. Access control decisions are made based on policies, for a variety of purposes. There are several different approaches to access control, including Mandatory Access Control (MAC) and Discretionary Access Control (DAC) (Open Web Application Security Project (OWASP), 2014), Role-based Access Control (RBAC) and Attribute-based Access Control (ABAC). In a DAC model, access decisions are based on the identity of users and/or membership in certain groups. Data owners are responsible for determining the type of access available to their resources. In MAC, sensitivity labels are assigned to users and resources. In this model, users are granted or denied access based on their security clearance and the label associated with the resource. Further, RBAC determines access to resources/data based on the role of the subject. Attribute-based access control makes access decisions based on attributes associated with subjects and objects. Access control

There are different types of policies which an access decision can be based on, e.g. privacy policies and security policies. Security policies are focused on maintaining confidentiality, integrity and availability of resources, while privacy policies are concerned with how and why data is used/shared/stored/etc. Privacy policies are the focus of access control decisions for the purpose of this work, and will be further described in the next section.

## ***Privacy Policies and Preferences***

Privacy policies describe an enterprise's data practices. This includes a description of what information is collected from users, what the information will be used for, how long it will be held, if/how the information will be shared to third parties, how long the information will be retained, etc. Consent is given by the user either implicitly or explicitly. Often, consent is implied just by using the services. Explicit consent can be given if the user is required to click "I agree" in regards to the privacy policy terms and conditions in order to receive services. Privacy policies are used for a company to outline their privacy practices relating to collection, use, retention, and sharing practices. Privacy preferences allow the user to create a set of rules to express how they wish their information to be managed.

### **Human Readable Policies**

Privacy policies are often provided to their users in natural language. Mobile applications often provide privacy policies to their users in this format. The purpose of these privacy policies is to provide the user with the details on why and how their information is collected while they are using the mobile application. As an illustration, the following is the *Furby Boom!* App Privacy Policy, available online or through the app:

Hasbro may collect non-personally identifiable information from devices that have installed a Hasbro app. This information is used to deliver services requested by users, such as content and updates within the app, as well as to support the internal operations of the app. For more information about the app, please contact us at <http://hasbro-new.custhelp.com/> (Hasbro 2013)

This policy is available before installing the application, and provides the user with an idea of what type of information is collected, and what the purpose is for its collection. This human-readable privacy policy is short and in simple terms, however it does not provide any detail on what information is actually collected, or how exactly it is used.

There are several concerns with how privacy policies are used in practice. In the case where a privacy policy is provided, the majority of users find them too complicated or long to read. Alternatively, as in the case of the *Furby Boom!* Privacy Policy, they can also be too vague. Human-readable privacy policies have a lot of limitations, some of which can be improved through the use of machine-readable policies.

### **XML and Machine Readable Policies**

Structured policy languages allow for automated enforcement of privacy policies and access decisions. A privacy policy language supports access constraints

(e.g. which subject can perform which action on which resource), as well as a description of access conditions. Policy languages must be platform independent, and able to integrate with the language used for access control policies (Anderson 2006). Privacy policies can be expressed in eXtensible Markup Language (XML) (W3C 2015) through policy assertion languages. XML is a flexible markup language used to describe data. XML is both human readable and machine readable, and many APIs have been developed for processing XML data. Various languages and tools have been developed for the specification of privacy policies and preferences based on XML, including P3P, EPAL, XACML, and WS-policy.

### **Platform for Privacy Preferences (P3P)**

Machine-readable privacy policy frameworks differ from human-readable ones. With machine-readable privacy policies, it allows the user to have more control over what information is collected and stored. Platform for Privacy Preferences (P3P) is a privacy policy framework created by the World Wide Web Consortium (W3C), based on XML designed to help end users manage their privacy while navigating websites that have differing privacy policies. User's privacy preferences are expressed using APPEL, A P3P Preference Exchange Language P3P also enables Websites to express their privacy practices in a standard format that can be retrieved automatically and interpreted easily by users of P3P browsers (Wenning 2007). P3P addresses user concerns about the type and number of data gathered by websites. At its most basic, any website that collects user information must clearly declare the reasons for the data collection, how it plans to use the information, and the amount of time it will retain the information. When using a P3P-compliant browser, cookies will be accepted, bypassed or denied depending on the previously mentioned user preferences. The user receives an alert when any privacy concerns arise and can override the previously set privacy level if they wish.

While P3P was primarily designed for Web sites, it has been the focus of many future directions including Web services and mobile services. In (Olurin et al. 2012), adaptation for the mobile environment is noted as a prominent future direction for P3P. Some major research questions are also addressed in this paper, including: how to create mobile-based privacy user agents that can communicate compact privacy policies of mobile web sites or applications to users, and how to delegate automatic access control privileges to mobile applications and websites based on user defined privacy preferences. Some concerns with moving P3P to the mobile environment, as outlined in (WAP-W3C 2000), include performance, security of the policies, extending P3P vocabulary for the mobile environment, and adapting the user interface for use on small mobile devices.

The traditional approach to P3P has several shortfalls in terms of enforcement. (Cranor, P3P is Dead, Long Live P3P! 2012b) reiterates how P3P has not been strongly embraced in practice. Popular websites such as Google and Facebook have published P3P "compact policies" (Cranor, Internet Explorer Privacy Protections also Being Circumvented by Facebook, and Many more 2012a). These policies state in human-readable code "this is not a P3P policy," while in practice, the system interprets it as a valid policy. In these situations, websites are able to technically comply with requirements but do not provide any actual privacy enforcement.

The authors of (Reay et al. 2009) performed an analysis on over 3000 P3P policies from 100,000 web sites to determine the relationship of privacy policies compared to legal requirements. The results of this study indicated that the surveyed website privacy policy statements had a widespread lack of adherence to legal mandates. Another report from the Canadian Internet Policy and Public Interest Clinic (Seligy und Lawson 2006) found similar results in a survey of 72 Canadian websites showing widespread noncompliance with PIPEDA. Many businesses are not taking necessary steps to preserve the privacy of their users. Another issue is faced by international web service companies (e.g. Google and Yahoo), who have difficulty with privacy regulation while they are required to address a multitude of different or conflicting international privacy laws and jurisdictions that must be negotiated (Reay et al. 2009).

### **Enterprise Privacy Authorization Language (EPAL)**

EPAL (Ashley et al. Enterprise Privacy Authorization Language (EPAL 1.2) 2003a) is another XML-based privacy policy language by W3C member IBM, designed to formalize internal privacy practices of an enterprise. EPAL is more suitable than P3P to express internal privacy policies that can be enforced by the enterprise's privacy management system. EPAL allows an enterprise to define its own list of data categories, data users, purposes, and actions, whereas P3P is limited to a predefined list (Ashley et al. The Enterprise Privacy Authorisation Language (EPAL)—How to Enforce Privacy Throughout an Enterprise 2003b).

### **eXtensible Access Control Markup Language (XACML)**

eXtensible Access Control Markup Language (XACML) (OASIS 2013) is an OASIS standard for access control language and architecture. The policy language uses the XML standard to define the policy and access control decision request and response. When there is an access request, an authorization decision/response can then be made based on the policy. XACML supports both centralized and decentralized policy management. XACML architecture uses the IETF Abstract Model for Policy Enforcement, which is further described in Sect. 2.3.5.

XACML specifies an abstract format for the authorization decision request as a description of the attempted resource access in terms of attributes (Anderson 2006). An XACML attribute is associated with one of four classes: Subject, Resource, Action, and Environment. Subject is the entity who is sending the access request, Resource is the resource that is to be accessed, and Action is the action to be performed on the resource (e.g. read or write). The environment attribute describes an additional characteristic of the request such as time of day.

The use of XACML has been widely adopted in Web services (Anderson 2006). A comparison between EPAL and XACML by Anderson (Anderson 2006), has recommended XACML for its functionality and flexibility. Lastly, Geospatial eXtensible Access Control Markup Language (GeoXACML) (Open Geospatial Consortium 2005) is an extension to XACML Version 2.0. by the Open Geospatial Consortium designed to control access to geospatial information. GeoXACML supports four types of functions: topological, geometric, bag & set, and conversion to manage geospatial information.



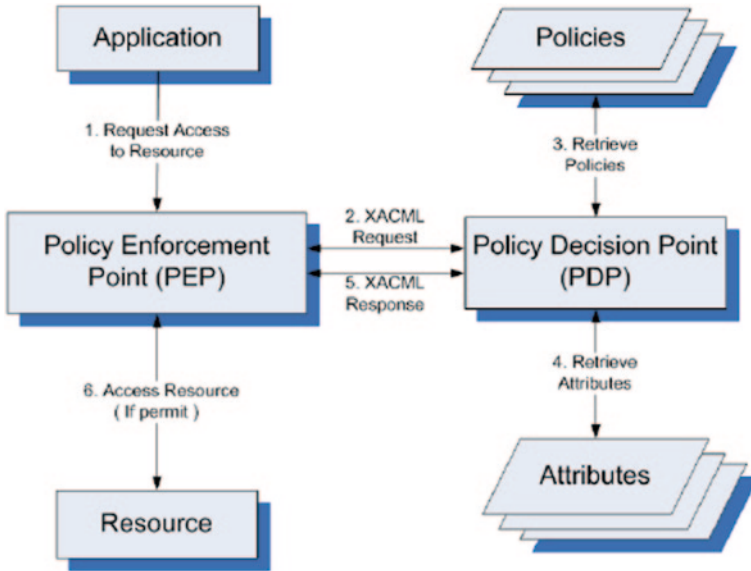


Fig. 9 IETF abstract model for policy enforcement. (Adapted from Waters et al. 1999)

### ***Abstract Model for Policy Enforcement***

A privacy policy alone does not guarantee that the policies will actually be enforced. This brings us onto the Abstract Model for Policy Enforcement proposed by IETF (terminology (Westerinen et al. 2001), model (Yavatkar et al. 2000)) and ISO (Open Systems Interconnection, 1966). This model has been used for policy enforcement for privacy policy languages such as EPAL and XACML.

Referring to Fig. 9, access control decisions are made by the Policy Decision Point (PDP), and enforced by the Policy Enforcement Point (PEP). When an application requests access to a resource, it sends the request to the PEP, which forwards the request to the PDP. The PDP then retrieves the policies and attributes to determine if the request complies. The PDP will make a decision and send a Permit or Deny response back to the PEP. The PEP will enforce the decision accordingly, providing access to the resource if permitted.

## **Section 4. Related Works in Privacy, Mobile and Location Services**

There exist a number of previous works in the fields of privacy, mobile services, and location-based services. To the best of our knowledge, no previous work exists which identifies a framework for privacy exclusively in the context of toy

computing, and especially with a focus on location. Further, although there has been work on the topic, there exists no widely accepted framework for privacy for any type of mobile services. This section will provide the reader with an overview of existing literature related to these topics.

### ***Mobile and Web Services Privacy Frameworks***

Hung et al. (Hung et al. Towards Standardized Web Services Privacy Technologies 2004) describe a vocabulary-independent privacy authorization language framework for Web services which addresses the privacy requirements (AC020) defined by the World Wide Web Consortium (W3C) in their Web Services Architecture (WSA) Requirements (World Wide Web Consortium (W3C) 2004). The framework recommends domain-specific vocabularies to be developed for different types of business applications (e.g. finance, healthcare, etc.). The authors introduce a protocol for enforcing privacy policies, in which privacy policies are described in P3P, and preferences exchange rules in APPEL. The paper also considers the use of privacy authorization language in other Web services-related languages such as WS-Policy, WS-Security, and WS-Privacy.

Access control is another area in which privacy is becoming more important. Traditional access control mechanisms such as discretionary access control (DAC), mandatory access control (MAC), and role based access control (RBAC) are not generally designed to accommodate privacy (Ferraiolo und Kuhn 1992), however some recent RBAC extensions have been introduced with a privacy-focused objective (Ni et al. 2007). Context- and location-based access control models have also been proposed (Seifert et al. 2009), where certain services and data can only be accessed in a certain context/location. This is especially useful in a BYOD scenario where users wish to separate work from personal activities depending on their context.

A lattice-based privacy aware access control (LPAAC) model is described in (Ghazinour und Barker 2013), in which data provider and collector privacy preferences are accommodated and enforced. This model allows the data collector to identify their privacy policies for purpose, visibility, granularity, and retention of data in terms of minimal acceptance limit (MinAL) and maximal acceptance limit (MaxAL). The data provider can then review the privacy policies and select their own preferences within the range, allowing them to receive services from the data collector while still being in control of their data. This paper also identifies the importance of enforcement, and provides an algorithm based on the above for determining the access decision to be enforced by the system it is being implemented on. The authors have also implemented their model using P3P (Ghazinour und Barker 2011).

ipShield, introduced by (Chakraborty et al. 2013b), is a privacy-aware framework designed to quantify an adversary's knowledge regarding the user's context and obscure it before sharing. This framework does not depend on the user being

anonymous, but instead focuses on choosing which data to share. It identifies several information disclosure systems, each corresponding to a specific privacy-utility tradeoff. Also introduces privacy mechanisms designed to realize those tradeoff points. Chakraborty et al. (Chakraborty et al. 2013a) propose a framework for protecting data against unwanted inferences. This technique involves a white list of inferences that are desirable and provide utility, as well as a black list for unwanted inferences that should be kept private. From there, the authors attempt to define how much the recipient can infer from shared data based on utility-privacy parameters. They identify bounds on the parameters and provide mechanisms for achieving the bounds.

### ***Location Privacy Techniques***

Various techniques have been used in attempt to preserve the privacy of a user's location. Different approaches could involve or not involve a trusted third party (Solanas et al. 2008). Some approaches include degrading the quality of location information (obfuscation) (Duckham und Kulik 2005) (Ardagna et al. 2007), creating fake location points (Taha und Shen 2013), uncertainty (Cheng und Prabhakar 2004) (Merrill et al. 2013), pseudonyms (Jorns et al. 2005), encryption (Fang et al. 2011) (Ashouri-Talouki und Baraani-Dastjerdi 2012), and k-anonymity (Gedik und Liu 2005). Policy-based access control is another technique which is used to decide whether a requesting subject can perform a given action on a data object. Various approaches for context-aware access control have been explored, which can also be used to preserve location privacy (Riboni et al. 2008).

IETF RFC6280 by Barnes et al. (Barnes et al. 2011) presents Geopriv, an architecture for location and location privacy in Internet applications. Geopriv is an Internet Best Current Practice, which enables users to express preferences for the disclosure of their location information. For example, the user can make a rule that their location is not to be disclosed beyond the intended recipient. This architecture binds the privacy rules to the data so that receiving entities are informed of when their data is shared to other parties.

## **Section 5. Chapter Summary**

In this chapter, we provided a background on the concept of toy computing, including the concepts of mobile services and physical computing. Next, we established a foundation on privacy in this context, including a description of XML-based privacy policy assertion languages including P3P and XACML. Finally, we provided an overview of some related works on mobile/web services privacy frameworks and location privacy.

## References

- Anderson, A. H. (2006). A Comparison of Two Privacy Policy Languages: EPAL and XACML. *Proceedings of the 3rd ACM Workshop on Secure Web Services*, (pp. 53–60). New York, NY.
- Andreae, H., Andreae, P., Low, J., & Brown, D. (2014). A Study of Auti: A Socially Assistive Robotic Toy. *IDC '14 Proceedings of the 2014 Conference on Interaction Design and Children* (pp. 245–248). New York, NY, USA: ACM.
- Android. (2015). *Location Strategies*. (Android Developer) Retrieved February 2015, from <http://developer.android.com/guide/topics/location/strategies.html>
- Araujo, G. M., & Siqueira, F. (2009). The Device Service Bus: A Solution for Embedded Device Integration through Web Services. *Proceedings of the 2009 ACM Symposium on Applied Computing* (pp. 185–189). New York, NY: ACM.
- Ardagna, C. A., Cremonini, M., Damiani, E., De Capitani di Vimercati, S., & Samarati, P. (2007). Location Privacy Protection Through Obfuscation-based Techniques. In *Lecture Notes in Computer Science: Data and Applications Security* (Vol. 4602, pp. 47–60). Redondo Beach, California, USA: Springer Berlin Heidelberg.
- Ashley, P., Hada, S., Karjoth, G., Powers, C., & Schunter, M. (2003a, November 10). *Enterprise Privacy Authorization Language (EPAL 1.2)*. Retrieved March 2015, from <http://www.w3.org/Submission/2003/SUBM-EPAL-20031110/>
- Ashley, P., Hada, S., Karjoth, G., Powers, C., & Schunter, M. (2003b). *The Enterprise Privacy Authorisation Language (EPAL)—How to Enforce Privacy Throughout an Enterprise*. Retrieved March 2015, from <http://www.w3.org/2003/p3p-ws/pp/ibm3.html>
- Ashouri-Talouki, M., & Baraani-Dastjerdi, A. (2012). Homomorphic Encryption to Preserve Location Privacy. *International Journal of Security and Its Applications*, 6(4), 183–190.
- Baber, C., & Westmancott, O. (2004). Social networks and mobile games: the use of bluetooth for a multiplayer card game. *6th International Conference on Human Computer Interaction with Mobile Devices and Services*, (pp. 98–107). Glasgow, Scotland.
- Barnes, R., Lepinski, M., Cooper, A., Morris, J., Tschofenig, H., & Schulzrinne, H. (2011). *An Architecture for Location and Location Privacy in Internet Applications*. IETF. Retrieved from <http://tools.ietf.org/html/rfc6280>
- Beckett, P. (2014). BYOD—Popular and Problematic. *Network Security*, 2014(9), 7–9.
- Booyah. (2015). *iTunes—MyTown2*. Retrieved February 2015, from <https://itunes.apple.com/app/mytown-2/id442345455>
- Chakraborty, S., Bitouze, N., Srivastava, M., & Dolocek, L. (2013a). Protecting Data Against Unwanted Inferences. *Proceedings of the 2013 IEEE Information Theory Workshop*. Seville, Spain.
- Chakraborty, S., Raghavan, K., Johnson, M., & Srivastava, M. (2013b). *A Framework for Context-Aware Privacy of Sensor Data on Mobile Systems. The Fourteenth Workshop on Mobile Computing Systems and Applications (ACM HotMobile2013)* (pp. 1–6, Article 11). New York, USA: ACM.
- Cheng, R., & Prabhakar, S. (2004). Using Uncertainty to Provide Privacy-Preserving and High-Quality Location-Based Services. *Workshop on Location Systems Privacy and Control (mobileHCI'04)*. Glasgow, Scotland.
- Cherubini, M., de Oliveira, R., Hiltunen, A., & Oliver, N. (2011). Barriers and bridges in the adoption of today's mobile phone contextual services. *MobileHCI '11* (pp. 167–176). Stockholm, Sweden: ACM.
- ChineseCUBES. (2014). *AR Cubes*. Retrieved from [https://www.chinesecubes.com/ar\\_cubes](https://www.chinesecubes.com/ar_cubes)
- Cranor, L. (2012a, February 18). *Internet Explorer Privacy Protections also Being Circumvented by Facebook, and Many more*. Retrieved October 2013, from TechPolicy.com: [http://www.techpolicy.com/Cranor\\_InternetExplorerPrivacyProtectionsBeingCircumvented-by-Google.aspx](http://www.techpolicy.com/Cranor_InternetExplorerPrivacyProtectionsBeingCircumvented-by-Google.aspx)
- Cranor, L. (2012b, December 3). *P3P is Dead, Long Live P3P!* Retrieved October 2013, from <http://lorrie.cranor.org/blog/2012/12/03/p3p-is-dead-long-live-p3p/>

- Cumulonimbus. (2015). *PressureNet*. (Google Play) Retrieved February 2015, from [https://play.google.com/store/apps/details?id=ca.cumulonimbus.barometernetwork&feature=nav\\_result#?t=W251bGwsMSwxLDMsImNhLmN1bXVsb25pbWJ1cy5iYXJvbWV0ZXJuZXR3b3JrIl0](https://play.google.com/store/apps/details?id=ca.cumulonimbus.barometernetwork&feature=nav_result#?t=W251bGwsMSwxLDMsImNhLmN1bXVsb25pbWJ1cy5iYXJvbWV0ZXJuZXR3b3JrIl0).
- Dewri, R., Annadata, P., Eltarjaman, W., & Thurimella, R. (2013). Inferring Trip Destinations from Driving Habits Data. *Workshop on Privacy in the Electronic Society*. Berlin, Germany.
- Dey, A. K., & Abowd, G. D. (1999). *Towards a Better Understanding of Context and Context-Awareness*. Georgia Institute of Technology, College of Computing.
- D’Hooge, H., & Goldstein, M. (2001). History of the Smart Toy Lab and Intel Play Toys. *Intel Technology Journal*, 2001(Q4).
- Digital Services Advisory Group and Federal Chief Information Officers Council, United States of America. (2012, August 23). *Bring Your Own Device*. Retrieved September 2014, from <http://www.whitehouse.gov/digitalgov/bring-your-own-device>
- Disterer, G., & Kleiner, C. (2013). BYOD Bring Your Own Device. *Procedia Technology*, 2013(9), 43–53.
- Duckham, M., & Kulik, L. (2005). A Formal Model of Obfuscation and Negotiation for Location Privacy. In H. G. al. (Ed.), *Pervasive Computing* (Vol. 3468, pp. 152–170). Munich, Germany: Springer-Verlag Berlin Heidelberg.
- Duri, S., Cole, A., Munson, J., & Christensen, J. (2001). An approach to providing a seamless end-user experience for location-aware applications. *1st International Workshop on Mobile Commerce*, 86(4), 20.
- El Kaed, C., Denneulin, Y., & Ottogalli, F.-G. (2011). Dynamic Service Adaptation for Plug and Play Device Interoperability. *Proceedings of the 7th International Conference on Network and Services Management*.
- Facebook. (2015). *Facebook Mobile*. (Facebook) Retrieved February 2015, from <https://www.facebook.com/mobile/>
- Fang, S.-H., Lai, W.-J., & Liang, Y.-C. (2011). An Encryption-Based Approach for Protecting Privacy in Network-Based Location Systems. *IEEE 2011 International Conference on Machine Learning and Cybernetics (ICMLC)* (pp. 377–380). Guilin: IEEE.
- Ferraiolo, D., & Kuhn, R. (1992). Role-based Access Control. *Proceedings of the 15th National Computer Security Conference*, (pp. 1–11).
- Fonseca, J., Abdelouahab, Z., Lopes, D., & Labidi, S. (2009). A Security Framework for SOA Applications on Mobile Environment. *International Journal of Network Security & ITS Applications*, 1(3), 90–107.
- Fusion Forge. (n.d.). *Welcome to the SOA4D Forge*. Retrieved September 2014, from <https://forge.soa4d.org/>
- Futuresight. (2011). *User Perspectives on Mobile Privacy—Summary of Research Findings*. GSMA.
- Gartner. (2014). *Key Challenges in BYOD*. Retrieved September 2014, from <http://www.gartner.com/technology/topics/byod.jsp>
- Gedik, B., & Liu, L. (2005). Location Privacy in Mobile Systems: A Personalized Anonymization Model. *25th IEEE International Conference on Distributed Computing Systems* (pp. 620–629). Columbus, OH: IEEE.
- Ghazinour, K., & Barker, K. (2011). Capturing P3P Semantics Using and Enforceable Lattice-based Structure. *Proceedings of the 4th International Workshop on Privacy and Anonymity in the Information Society*, (p. 4). New York, NY, USA.
- Ghazinour, K., & Barker, K. (2013). A Privacy Preserving Model Bridging Data Provider and Collector Preferences. *Proceedings of the Joint EDBT/ICDT 2013 Workshops* (pp. 174–178). New York, NY, USA: ACM.
- Golatowski, F., Bobek, A., & Zeeb, E. (n.d.). *Web Services for Devices—About*. (WS4D) Retrieved September 2014, from <http://ws4d.e-technik.uni-rostock.de/about/>
- Google. (n.d.). *Sensors Overview*. (developer.android.com) Retrieved September 2014, from [http://developer.android.com/guide/topics/sensors/sensors\\_overview.html](http://developer.android.com/guide/topics/sensors/sensors_overview.html)
- Gross, M. D., & Eisenberg, M. (2007). Why Toys Shouldn’t Work “Like Magic”: Children’s Technology and the Values of Construction and Control. *The First IEEE International Workshop on*

- Digital Game and Intelligent Toy Enhanced Learning (DIGITEL '07)*. Jhongli, Taiwan: IEEE Computer Society.
- Gu, T., Pung, H., & Zhang, D. (2004). A Middleware for Building Context-Aware Mobile Services. *IEEE Vehicular Technology Conference*. 5, pp. 2656–2660. IEEE.
- Hasbro. (2013). *Furby Boom*. Hasbro. Retrieved November 2013, from [http://www.hasbro.com/furby/en\\_CA/](http://www.hasbro.com/furby/en_CA/)
- Hinske, S., & Langheinrich, M. (2007). Managing Augmented Toy Environments—A New Perspective for Smart Space Management. *Proceedings of the 4th International Workshop on Managing Ubiquitous Communications and Services (MUCS)*. Munich, Germany.
- Hinske, S., Langheinrich, M., & Lampe, M. (2008). Towards Guidelines for Designing Augmented Toy Environments. *Designing Interactive Systems (DIS) 2008*. Cape Town, South Africa: ACM.
- Hung, P. C., & Cheng, V. S. (2009). Privacy. In *Encyclopedia of Database Systems* (pp. 2136–2137). Springer.
- Hung, P. C., Ferrari, E., & Carminati, B. (2004). Towards Standardized Web Services Privacy Technologies. *Proceedings of the IEEE International Conference on Web Services (ICWS'04)*. San Diego, CA.
- Instagram. (2015). *Instagram*. (Instagram) Retrieved February 2015, from <http://instagram.com/>
- Jorns, O., Jung, O., Gross, J., & Bessler, S. (2005). A Privacy Enhancement Mechanism for Location Based Service Architectures Using Transaction Pseudonyms. In *Lecture Notes in Computer Science: Trust, Privacy, and Security in Digital Business* (Vol. 3592, pp. 100–109). Copenhagen, Denmark: Springer-Verlag Berlin Heidelberg.
- Kaasinen, E. (2003, May). User Needs for Location-Aware Mobile Services. *Personal and Ubiquitous Computing*, 7(1), 70–79.
- Lee, S., & Doh, Y. Y. (2013). iSpy: RFID-Driven Language Learning Toy Integrating Living Environment. *CHI '13 Proceedings from the 2013 International Conference on Interaction Design and Children* (pp. 697–702). Paris, France: ACM.
- Luckin, R., Connolly, D., Plowman, L., & Airey, S. (2003). Children's Interactions with Interactive Toy Technology. *Journal of Computer Assisted Learning*, 19, 165–176.
- MEF. (2013). *MEF Global Privacy Report 2013*. MEF.
- Merriam-Webster. (n.d.). *Location*. Retrieved January 2015, from <http://www.merriam-webster.com/dictionary/location>
- Merrill, S., Basalp, N., Biskup, J., Buchmann, E., Clifton, C., Kuijpers, B., ... Savas, E. (2013). Privacy Through Uncertainty in Location-Based Services. *2013 IEEE 14th International Conference on Mobile Data Management* (pp. 67–72). Milan, Italy: IEEE.
- Microsoft. (2007). *Introducing Devices Profile for Web Services*.
- National Geographic. (n.d.). *Encyclopedic Entry: Location*. Retrieved January 2015, from [http://education.nationalgeographic.com/education/encyclopedia/location/?ar\\_a=1](http://education.nationalgeographic.com/education/encyclopedia/location/?ar_a=1)
- Ni, Q., Trombetta, A., Bertino, E., & Lobo, J. (2007). Privacy-Aware Role Based Access Control. *SACMAT '07: Proceedings of the 12th ACM Symposium on Access Control Models and Technologies* (pp. 41–50). France: ACM.
- OASIS. (2009, July). *OASIS Devices Profile for Web Services*. Retrieved December 2013, from <http://docs.oasis-open.org/ws-dd/ns/dpws/2009/01>
- OASIS. (2013). *eXtensible Access Control Markup Language (XACML) Version 3.0*. OASIS. Retrieved January 2015, from <http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-os-en.pdf>
- Olurin, M., Adams, C., & Logrippo, L. (2012). Platform for Privacy Preferences (P3P): Current Status and Future Directions. *2012 Tenth Annual International Conference on Privacy, Security and Trust (PST)*, (pp. 217–220). Paris, France.
- Open Geospatial Consortium. (2005, December 15). *GeoXACML*. Retrieved December 2014, from <https://geoxacml.secure-dimensions.com/>
- Open Systems Interconnection. (1966). Information Technology—Security Frameworks for Open Systems: Access Control Framework.



- Open Web Application Security Project (OWASP). (2014). *Access Control Cheat Sheet*. OWASP. Retrieved from [https://www.owasp.org/index.php/Access\\_Control\\_Cheat\\_Sheet](https://www.owasp.org/index.php/Access_Control_Cheat_Sheet)
- Pandit, A. A., & Kumar, A. (2012). Conceptual Framework and a Critical Review for Privacy Preservation in Context Aware Systems. *IEEE 2012 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery* (pp. 435–442). Sanya, China: IEEE.
- Patil, S., Norcie, G., Kapadia, A., & Lee, A. J. (2012). Reasons, Rewards, Regrets: Privacy Considerations in Location Sharing as an Interactive Practice. *Symposium on Usable Privacy and Security*. Washington, D.C.
- Plowman, L., & Luckin, R. (February 2004). Interactivity, Interfaces, and Smart Toys. *Computer*, 98–100.
- Pohlsen, S., Schlichting, S., Strahle, M., Franz, F., & Werner, C. (2009). A Concept for a Medical Device Plug-and-Play Architecture based on Web Services. *ACM SIGBED Review—Special Issue on the 2nd Joint Workshop on High Confidence Medical Devices, Software, and Systems (HCMDSS) and Medical Device Plug-and-Play (MD PnP) Interoperability*, 6(2), Article 6.
- Pura, M. (2005). Linking perceived value and loyalty in location-based mobile services. *Managing Services Quality*, 15(6), 509–538.
- Reay, I., Dick, S., & Miller, J. (2009). A Large-Scale Empirical Study of P3P Privacy Policies: Stated Actions vs. Legal Obligations. *ACM Transactions on The Web*, 3(2), 6:1–6:34.
- Riboni, D., Pareschi, L., & Bettini, C. (2008). Privacy in Georeferenced Context-aware Services: A Survey. *Privacy in Location-Based Applications (PiLBA'08)*, (pp. 24–43). Malaga, Spain.
- Rovio. (2015). *Angry Birds*. Retrieved February 2015, from <http://www.rovio.com/en/our-work/games/view/1/angry-birds>
- Saha, D. (2003). Pervasive Computing: A Paradigm for the 21st Century. *Computer*, 36(3), 25–31.
- Sandhu, R., & Samarati, P. (1994). Access Control: Principles and Practice. *IEEE Communications Magazine*, 1994(September), 40–48.
- Schilit, B., Adams, N., & Want, R. (1994). Context-Aware Computing Applications. *WMCA '94* (pp. 85–90). Washington, DC, USA: IEEE Computer Society.
- Schmidt, A. (2005). Interactive Context-Aware Systems Interacting with Ambient Intelligence. In G. Riva, F. Vatalaro, F. Davide, & M. Alcaniz (Eds.), *Ambient Intelligence* (pp. 159–178). IOS Press.
- Schmidt, A., Beigle, M., & Gellersen, H. W. (1999). There is more to context than location. *Computer & Graphics Journal*, 23(6), 893–902.
- Seifert, J., De Luca, A., & Conradi, B. (2009). A Context-Sensitive Security Model for Privacy Protection on Mobile Phones. *Proceedings of the 11th International Conference on Human-Computer Interaction with Mobile Devices and Services*. New York, NY.
- Seligy, J., & Lawson, P. (2006). *Compliance with Canadian Data Protection Laws: Are Retailers Measuring Up?* Ottawa: Canadian Internet Policy and Public Interest Clinic.
- Sheth, A., Anantharam, P., & Henson, C. (2013). Physical-Cyber-Social Computing: An Early 21st Century Approach. *Intelligent Systems, IEEE*, 28(1), 78–82.
- Six to Start. (n.d.). *Zombies, Run! 3*. (Six to Start) Retrieved September 2014, from <https://www.zombiesrungame.com/>
- Solanas, A., Domingo-Ferrer, J., & Martinez-Balleste, A. (2008). Location Privacy in Location-Based Services: Beyond TTP-based Schemes. *Privacy in Location-Based Applications (PiLBA'08)*, (pp. 12–23). Malaga, Spain.
- Sphero. (2014). *Sphero*. Retrieved August 2014, from <http://www.gosphero.com>
- Taha, S., & Shen, X. (2013). A Physical-Layer Location Privacy-Preserving Scheme for Mobile Public Hotspots in NEMO-Based VANETs. *IEEE Transactions on Intelligent Transportation Systems*, 14(4), 1665–1680.
- Tech4Kids. (2013). *Tek Recon*. (Tech4Kids) Retrieved August 2014, from <http://www.tekrecon.com/>
- TIME and Qualcomm. (2012, July). *Your Wireless Life: Results of TIME's Mobility Poll*. Retrieved November 2013, from <http://content.time.com/time/interactive/0,31813,2122187,00.html>
- Toymail Co. LLC. (2014). *Toy Mail*. (Toymail Co. LLC) Retrieved from <http://www.toymail.co/>



- Unger, S., Zeeb, E., Golasowski, F., Grandy, H., & Timmermann, D. (2010). Extending the Devices Profile for Web Services for Secure Mobile Device Communication. *4th International Workshop on Trustworthy Internet of People, Things & Services at the Internet of Things Conference*. Tokyo, Japan. Retrieved from [http://webcache.googleusercontent.com/search?q=cache:ArR-T87Qq\\_8J:www.imd.uni-rostock.de/veroeff/DWPS-DA-paper.pdf](http://webcache.googleusercontent.com/search?q=cache:ArR-T87Qq_8J:www.imd.uni-rostock.de/veroeff/DWPS-DA-paper.pdf)
- W3C. (2001). *Web Services Description Language (WSDL) 1.1*. W3C. Retrieved from [www.w3.org/TR/wsdl](http://www.w3.org/TR/wsdl)
- W3C. (2004). *Web Services Architecture*. Retrieved 2014, from <http://www.w3.org/TR/ws-arch>
- W3C. (2015). Extensible Markup Language (XML). W3C. Retrieved from <http://www.w3.org/XML>
- Want, R. (2006). An Introduction to RFID Technology. *IEEE Pervasive Computing*, 5(1).
- WAP-W3C. (2000). *Report from WAP-W3C Joint Workshop on Mobile Web Privacy*. Munich, Germany: W3C.
- Waters, G., Wheeler, J., Westerinen, A., Rafalow, L., & Moore, R. (1999). Policy Framework Architecture. IETF. Retrieved from <http://tools.ietf.org/html/draft-ietf-policy-arch-00>
- Wenning, R. (2007, November 20). *Platform for Privacy Preferences (P3P) Project: Enabling Smarter Privacy Tools for the Web*. Retrieved December 2013, from W3C: <http://www.w3.org/P3P/>
- Westerinen, A., Schnizlein, J., Strassner, J., Scherling, M., Quinn, B., Herzog, S., ... Waldbusser, S. (2001, November). Terminology for Policy-Based Management. IETF RFC 3198. Retrieved from <http://www.ietf.org/rfc/rfc3198.txt>
- Westeyn, T. L., Abowd, G. D., Starner, T. E., Johnson, J. M., Presti, P. W., & Weaver, K. A. (2012). Monitoring Children's Developmental Progress using Augmented Toys and Activity Recognition. *Personal Ubiquitous Computing*, 2012(16), 169–191.
- Whalen, T. (2011). Mobile Devices and Location Privacy: Where do we go from Here? *IEEE Security & Privacy*, 9(6), 61–62.
- Woollaston, V. (2014, February 20). *Step Aside Ted, There's A New Talking Teddy in Town: WikiBear Connects to the Web to Chat, Answer Questions and Tell Jokes*. Retrieved from DailyMail: <http://www.dailymail.co.uk/sciencetech/article-2564015/Step-aside-Ted-theres-new-talking-teddy-town-WikiBear-connects-web-chat-answer-questions-tells-jokes.html>
- World Economic Forum. (2011). *Personal Data: The Emergence of a New Asset Class*. World Economic Forum.
- World Wide Web Consortium (W3C). (2004, February 11). *Web Services Architecture Requirements*. Retrieved from <http://www.w3.org/TR/wsa-reqs/>
- Yavatkar, R., Pendarakis, D., & Guerin, R. (2000, January). A Framework for Policy-Based Admission Control. IETF RFC2753. Retrieved from <http://www.ietf.org/rfc/rfc2753.txt>
- Yelp. (2015). *Yelp Mobile*. (Yelp) Retrieved February 2015, from <http://www.yelp.ca/yelpmobile>
- zomato. (2015). *Urbanspoon*. (zomato) Retrieved February 2015, from <http://www.urbanspoon.com/>

<http://www.springer.com/978-3-319-21322-4>

Mobile Services for Toy Computing

Hung, P.C.K. (Ed.)

2015, VIII, 192 p. 110 illus., 105 illus. in color.,

Hardcover

ISBN: 978-3-319-21322-4