

DRO

Deakin University's Research Repository

Warren, I., Weerasinghe, T., Maddison, R. and Wang, Y. 2011, Odin^{Telehealth}: a mobile service platform for telehealth, in *MobiWIS 2011 : Proceedings of the 8th International Conference on Mobile Web Information Systems*, Elsevier, Amsterdam, The Netherlands, pp. 681-688.

DOI: [10.1016/j.procs.2011.07.089](https://doi.org/10.1016/j.procs.2011.07.089)

This is the published version.

©2011, Elsevier

Reproduced by Deakin University under the terms of the [Creative Commons Attribution Non-Commercial No-Derivatives Licence](https://creativecommons.org/licenses/by-nc-nd/4.0/)

Available from Deakin Research Online:

<http://hdl.handle.net/10536/DRO/DU:30081957>



The 8th International Conference on Mobile Web Information Systems (MobiWIS)

Odin^{Telehealth}: A Mobile Service Platform for Telehealth

I. Warren^a, T. Weerasinghe^a, R. Maddison^b, Y. Wang^c

^a*Department of Computer Science, University of Auckland, Auckland, New Zealand*

^b*Clinical Trials Research Unit, University of Auckland, Auckland, New Zealand*

^c*Department of Electrical and Computer Engineering, University of Auckland, New Zealand*

Abstract

In this paper we present our experience with developing telehealth applications using smartphones in conjunction with a mobile service provisioning middleware platform named Odin. Common requirements for mobile telehealth applications include the need to support multiple stakeholders, high levels of connectivity between users, real-time interaction, bidirectional communication channels for exchanging diverse data types, computationally intensive processing and security. Meeting these needs is a non-trivial task in mobile execution environments given the limitations of mobile devices and wireless and mobile networks. Odin enables a separation of concerns between application functionality and resource management governing mobile devices and wireless networking. Using Odin, application developers can rapidly develop telehealth applications without needing to address underlying complexity. We describe development of an Odin-based monitoring application that meets many of the aforementioned requirements associated with mobile telehealth. Based on evaluation, results for smartphone power consumption, network bandwidth usage, and communication latency suggest that Odin is an appropriate platform for general telehealth applications.

Keywords: telehealth, telemedicine, mobile services, middleware, service oriented architecture

1. Introduction

Telemedicine [1] involves the electronic exchange of medical information with the goal of improving patients' health. Closely associated with telemedicine is *telehealth* [2], which takes a broader view of remote healthcare, not necessarily involving clinical services. Telemedicine and telehealth include videoconferencing, transmission of still images, e-health patient portals, remote monitoring of vital signs, and continuing medical education.

Telemedicine holds much promise for addressing key challenges faced by healthcare across the world [3]. It is widely recognised that world populations are growing and that people are generally living longer. With longer life comes an increase in patient load, with more patients requiring regular and long-term treatment. In addition, the recent decline in the number of medical graduates entering primary care motivates use of assistive technologies. The skills shortage will have particular impact on patients living in remote and rural areas, with the diminished supply of medical specialists being located in distant city areas. Furthermore, the rising costs of healthcare provisioning [4] provide additional motivation for low cost aids like telemedicine. The wider span of telehealth covers sports medicine, coaching and monitoring of individuals' well-being in extreme operating environments.

Email addresses: ian-w@cs.auckland.ac.nz (I. Warren), twee003@aucklanduni.ac.nz (T. Weerasinghe), r.maddison@ctr.u.auckland.ac.nz (R. Maddison), ywan478@aucklanduni.ac.nz (Y. Wang)

In this paper, we present our recent work that builds on Odin [5], a mobile service middleware platform, and which targets telehealth. Fundamentally, an Odin-based telehealth application comprises commodity smartphones, augmented with body sensors, that host *mobile services*. A mobile service, like a conventional service, is one that offers a well-defined interface, and can be published, discovered and consumed. Contemporary smartphones are becoming increasingly ubiquitous, have sufficient computational resources for running services, and hence are a natural device for telehealth applications. Odin provides key infrastructure to promote mobile service accessibility, availability and scalability, and, in addition, allows mobile service applications to be rapidly developed.

The remainder of this paper is structured as follows. In Section 2 we present some diverse telehealth scenarios, and based on these extract a general statement of need for telehealth applications. We proceed in Section 3 with an overview of the Odin platform, and discuss its suitability for the telehealth domain. In Section 4, we describe a telehealth application that has been developed using Odin, and in Section 5 we report on its evaluation. In Section 6 we briefly discuss existing telehealth monitoring systems. Finally, we conclude in Section 7.

2. Telehealth Scenarios

The following scenarios serve to motivate the development of telehealth applications and illustrate their role in clinical and sports domains. Table 1 summarises key properties of each scenario application.

- *Post surgery monitoring of a heart-condition patient.* Once discharged following heart surgery, it is imperative that a patient's vital signs be closely monitored [6]. The patient could be equipped with a smart phone capable of monitoring heart rate, posture, and breathing rate. Particular combinations of these readings indicate cause for concern and should be brought to the immediate attention of remote medical staff. For example, in response to a patient's a heart rate being in excess of 140 BPM when not exercising, a remote healthcare professional might react initially by requesting the patient's smartphone to record and transmit a 6 second strip of ECG data, and depending on the result of this ask the patient to return to the hospital. Alternatively, the professional might have the application contact another individual, such as a relative, to check on the patient. In addition to generating events, the patient's smartphone could periodically transmit monitored data to be made available to medical personnel. The data could be used, for example, to identify rhythm disturbances, in preparation for scheduled patient consultations to get a better understanding of the patient's recovery and to guide treatment.
- *Diabetes management.* Diabetes covers a family of diseases that represent chronic conditions for sufferers [7]. Essentially, diabetics have a high blood sugar because either the body does not produce sufficient insulin or because cells do not respond to the insulin that is actually produced. A diabetic patient might carry a smartphone connected to a blood sugar level sensor implanted in their body. The sensor continuously relays the patient's blood sugar level to the smartphone. Through a combination of autonomous operation and interaction with remote healthcare workers, the patient's condition can be effectively managed. Using the sensor alone, the smartphone can react and advise the patient to take specified amounts of insulin and to make any immediate dietary adjustments. By periodically sending the monitored data to remote healthcare staff, longer-term trends in the patient's blood sugar level can be detected and used during scheduled consultations with the patient to proactively improve the quality of the individualised care plan.
- *Athlete training.* A competitive athlete benefits significantly from effective coaching. Using an unobtrusive body area network and smartphone, sensors could gather data including heart rate, breathing rate, skin temperature and hydration level, and make these available to the smartphone. In turn, the smart phone, in real-time, would send the captured data to a remote coach, who, in conjunction with software to help analyse the incoming data, would instruct the athlete appropriately, again in real-time. For example the athlete might be asked to take fluids to preempt dehydration, which is particularly damaging to an athlete's performance; once dehydration has started the effects cannot be reversed in a short time. In other cases, the athlete might be asked to run faster to increase their heart rate or to exert less as a means to pace herself. In addition to working with physiological data, the smartphone's GPS sensor could be used to track the athlete's location. Using software to display a map of the athlete's location, the coach might adapt the athlete's session by changing her route, terrain and elevation.

	Scenario		
	A	B	C
Multiple stakeholders	Patient Health professional Carer	Patient Health professional	Athlete Coach
Connectivity	High level of connectivity required in general; short periods of intermittent connectivity can be tolerated when gathering normal monitoring data		High level of connectivity required to support “conversation” between stakeholders
Real-time interaction	Applications involve a mix of communication, some of which is subject to real-time constraints, e.g. transmission of and responding to abnormal sign data		Coach requires monitoring data in real-time; athlete requires instruction in real-time
Bidirectional communication	Applications involve monitored subject initiating communication, e.g. a diabetic patient’s smartphone streams monitoring data to a remote monitoring agent (health professional); monitoring agents initiate communication, e.g. a doctor requests ECG data from a heart patient’s device or adjusts the sampling frequency		
Information exchange	Vital sign streaming, and potentially large datasets (e.g. ECG data); service requests		Vital sign and location streaming, plus voice instructional data
Processing	Potentially computationally intensive algorithms for analysing captured data, e.g. route/path selection for athlete based on physiological and location data		
Security	Medical applications are subject to confidentiality of transmitted data; access control is also important, e.g. only healthcare stakeholders should change the behaviour of vital signs monitoring		

Table 1: Scenario properties

3. Odin Middleware Platform

Odin [5] is a middleware platform for developing and deploying mobile service applications. As shown in Figure 1, an Odin application comprises 3 software components: i) a service, known as the *device-service*, running on a mobile device, ii) a *surrogate* that serves as a proxy for the device-service, and iii) an intermediary, hosted by a well-known Internet server, called a *surrogate-host*. A surrogate-host acts as a container for surrogates, and a surrogate facilitates communication between a device-service and its clients, be they mobile or otherwise. Odin adheres to service oriented architecture principles and is built using Java and Jini technologies.

As discussed in Section 2, connectivity among stakeholders is a basic requirement for mobile telehealth solutions. However, mobile and wireless networks are characterised by relatively low bandwidth, intermittent connectivity, and often expensive usage costs. Further difficulty is encountered when running mobile services since mobile network operators do not generally expose the addresses of devices connected on their networks [8]. Moreover, operators typically enforce security policies that prevent connections to mobile devices from outside their networks, and lack support for maintaining connectivity with roaming mobile devices [8]. These characteristics pose significant obstacles to realising telehealth applications with dependable bidirectional communication links for transmitting arbitrary data on demand.

Odin addresses these problems using i) its intermediary-based architecture and ii) an interconnect protocol for device-service to surrogate communication [8]. Using the intermediary architecture, clients communicate with a service, via its surrogate, as if it were a conventional service on a fixed network, and without regard for the hosting device’s fragile point of attachment to the network or its inherent mobility.

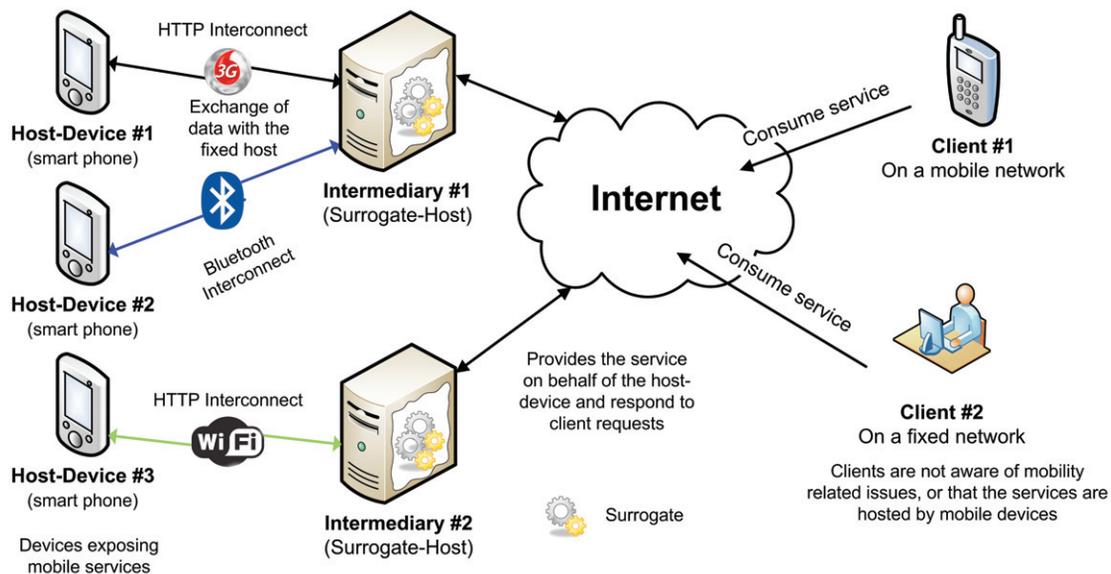


Figure 1: High level overview of Odin Architecture

The interconnect protocol solves the problems associated with service accessibility and mobility [8]. A role played by the device-service middleware is to establish a HTTP connection with its surrogate (such connection requests originating from *within* mobile networks are typically permitted by their operators). Once connected, a device-service sends periodic keep-alive messages to its surrogate. Any service requests received from clients are piggybacked onto keep-alive responses sent by the surrogate to the device-service. Service replies are then piggybacked onto subsequent keep-alive messages, the replies being relayed to requesting clients. Over a given connection, Odin offers asynchronous messaging, synchronous method invocation, publish/subscribe and streaming abstractions involving arbitrary data. This solution offers a rich bi-directional communication channel and addresses the needs concerned with information exchange sought by telehealth applications.

While mobile and wireless networks are brittle, today's smartphones typically offer multihoming capabilities that allow them to connect to multiple networks, such as 3G, WiFi and Bluetooth. Odin leverages this support with vertical handover [8], which enables a smartphone to transparently switch between networks as their availability changes. Odin's vertical handover mechanism [8] implements a lossless protocol that guarantees integrity of transmitted data regardless of network failure. In addition to reactive handover, which involves recovering from the unexpected loss of a network, Odin also provides proactive handover that automatically selects the preferred network when available. Network preference is configurable and is governed by factors such as speed and cost. Hence, Odin aims to provide the levels of network connectivity sought by telehealth applications involving, for example, mobile and cost-conscious patients.

Contemporary smartphones are much better resourced than their predecessors, but their local resources remain limited when compared to desktop hardware. Odin employs context-aware mechanisms to help conserve resource usage [9]. For example, maintaining a WiFi connection draws more power than having a Bluetooth link active; in cases where power supply is low, other factors being equal it is preferable to use the Bluetooth connection in order to prolong a mobile service's operation. Similarly, enabling GPS significantly reduces battery life, and the middleware is responsible for managing such sensors based on application need.

Odin's intermediary-based architecture also allows applications to leverage the additional computational power of surrogate-host servers. A mobile service application can offload computationally intensive tasks from the device-service to its surrogate. Returning to the athlete coaching scenario discussed in Section 2, for example, the analysis of athlete data and selection of an appropriate route would best be performed by the surrogate before relaying this to the coach's web browser. Hence, mobile service applications can be structured so that only smartphone-specific tasks need be processed by the smartphone, further contributing to resource management. Furthermore, use of the

surrogate provides some resilience to network failures and promotes service scalability, since surrogates may be able to service client requests drawing on cached data without needing to communicate with the device-service. With monitoring applications that involve patient tracking, for example, a patient's last known location can be returned from a surrogate cache.

Odin provides a rich API to assist application developers with rapidly developing mobile services. In short, this shields application developers from the complexity inherent in mobile applications that results from the networking and device characteristics discussed above. The API also allows developers to implement their own security mechanisms when transmitting data, which, as noted in Section 2, is a basic requirement for medical applications.

Others [10, 11, 12] have tackled mobile service provisioning using variations on intermediary-based architectures. MobileHost [11] and work done by Chaudron et al. [12] expose mobile services as Web services, while MSP [10] is another Jini-based middleware. Of these, only MSP, with protocols similar to Odin, operates without the need for contribution from mobile network operators. While other intermediary-based solutions demonstrate some value in using intermediaries, they lack features such as server-side resource leasing and caching that contribute to service availability and scalability. Yet other research [13, 14] has explored hosting mobile services directly on small devices; while they have a reduced footprint, they lack the capabilities associated with the intermediary-based alternatives.

4. Subject Monitoring Application

Using the Odin middleware, we have recently developed a number of telehealth applications, and focus on a generic monitoring application here. The application comprises a smartphone and Bluetooth body sensor capable of taking measurements that include heart rate, breathing rate, posture, and skin temperature. Figure 2 shows part of the user interface for the application.

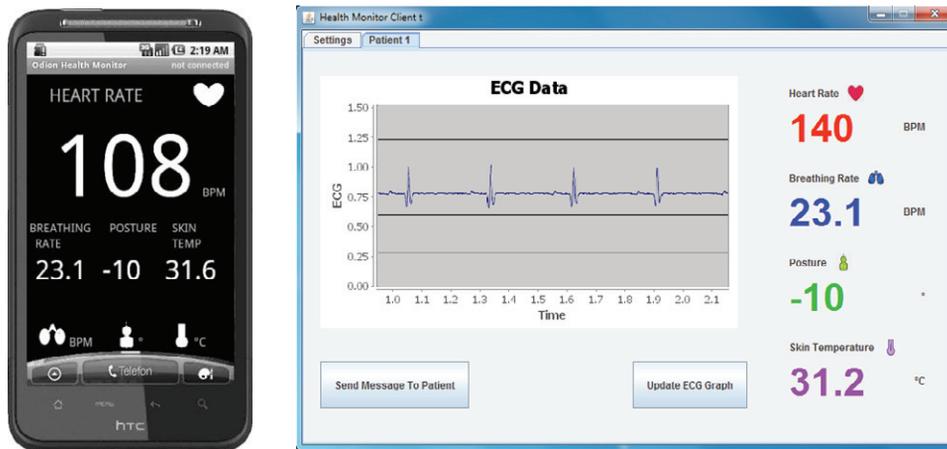


Figure 2: Patient Monitoring application user interfaces

The smartphone application (device-service) allows a user to view their vital sign data as they are gathered from the sensor. The application generates two kinds of event: i) a suspected fall event that is subscribed to by caregivers, and ii) a heart rate outside of bounds event subscribed to by medical staff. The smartphone application is responsible for processing sign data to the extent of detecting and firing the above events, and streaming the data to its surrogate for consumption by remote medical professionals. The surrogate component exposes the mobile service interface to clients (doctors and caregivers). The doctor's client application allows doctors to view monitored data, to request and display ECG data and to amend what data is actually captured and at what frequency it is transmitted. For caregivers, the service allows them, via an application that runs on their smartphone, to communicate their location to the patient. This is useful in cases where the patient is distressed and can call on a caregiver who is close by. As discussed above, suspected fall events are also presented to the caregivers.

5. Evaluation

The monitoring application introduced in the preceding section has been quantifiably evaluated with respect to battery use, bandwidth consumption and data transmission delay. We have focused on these aspects because the rate of power consumption determines a service's operational lifetime; operational cost to end users, particularly patients, is an important consideration and so bandwidth use has been measured; and transmission latency is of interest given the need for real-time interaction in many telehealth applications.

Experiments were conducted using a HTC Hero smartphone running Android 2.1 OS, with a keep-alive period of 10 seconds. The surrogate-host was deployed on a publically accessible server running Windows Server 2008 64-bit OS on an Intel Xeon x5570 2.96 GHz processor with 4GB of RAM. A 54Mbps 802.11g wireless network and 3G mobile broadband service was available for the smartphone to connect to the surrogate-host. The Internet connection used by WiFi and 3G connections had average download speeds of 3 Mbps and 2.4 Mbps respectively¹. A Bluetooth enabled Zephyr BioHarness was used as the body sensor. Throughout all experiments, GPS and Bluetooth were enabled, with either 3G or WiFi connectivity being active.

Smartphone operating conditions	Operational lifetime	Scenario	WiFi (sec)	3G (sec)
Service not deployed	13 hrs 55 mins	Client sending messages	5.670	5.543
Service idle WiFi connection to Surrogate Host	9 hrs 35 mins	Device sending data	1.340	4.995
Service idle 3G connection to Surrogate Host	9 hrs 5 mins	Table 3: Transmission latency		
Service running over WiFi	6 hrs 26 mins			
Service running over 3G	6 hrs 1 min			

Table 2: Operational lifetime for smartphone on a single charge

For power consumption, the smartphone was tested under five sets of operating conditions. First, the smartphone application was not deployed, and so this experiment yields a baseline operating period for the device when switched on but passive. The second and third cases involved deploying the application and having it maintain either a WiFi or 3G connection, but otherwise remaining idle. The remaining two cases involved running the application under full load, streaming monitored data at a periodic rate of 20 seconds through to the client and responding to client messages once per minute. Table 2 shows the operational lifetimes of the smartphone for each of these test scenarios.

Transmission latency was tested by measuring the time difference between sending and receiving two core pieces of information: i) messages sent by client applications and received by the device-service, and ii) monitoring data sent by the smartphone and received by the client. The information was tracked using unique identifiers generated within the client and BioHarness sensor. The smartphone and client were synchronized to have the same system time using a public Internet Time service. The experiment was conducted by sending 100 data points at 20 second intervals from the device, and 100 messages every 60 seconds from the client.

Results for the transmission latency experiments are presented in Table 3. On average, messages were delayed by 5.5 seconds regardless of the type of network connection being used. This is because, as described in Section 3, messages have to wait for keep-alive requests in order to be sent to the smartphone. The actual results, incurring a latency on average of approximately half the keep-alive period, agree with the expected behaviour. The results also reveal that use of WiFi to connect the device-service with the surrogate yields more responsive behaviour than when 3G is used. We hypothesise that the low network congestion on our local WiFi network and the high latencies associated with 3G are the main contributing factors to the observed results.

However, given the packet-switching nature and unpredictable usage patterns of underlying public IP-based networks, it is not possible to guarantee an upper bound on the time taken for a device/client communication. Moreover,

¹Network speeds were measured using `speedtest.net`.

timely communication necessarily requires connectivity, and while Odin attempts to maintain some form of connectivity, using vertical handover techniques, this cannot be guaranteed. Scenarios such as the device's power supply being exhausted or the device being located in an area that does not fall within coverage of any network cannot be remedied by software. While such eventualities are tolerable for many telehealth applications, our middleware/smartphone combination alone is inappropriate in cases involving hard real-time communication requirements and where loss of connectivity might endanger life.

Bandwidth usage was monitored when the service was running under the two full-load scenarios, as outlined above. A third party Android application was used to instrument bandwidth². The experiments revealed that the service consumed 0.90 MB/hr (both inbound and outbound traffic) regardless of using either the 3G or WiFi network interfaces. This was expected, since our middleware does not currently optimise data transmission when using WiFi connectivity; for both WiFi and 3G connections the middleware uses the HTTP interconnect outlined in Section 3.

Overall, the evaluation demonstrates that the Odin middleware is appropriate for a large subset of telehealth applications. Despite reducing a smartphone's operational lifetime by over 50%, the monitoring application has been demonstrated to run for over 6 hours without requiring recharging. The high rates of information exchange used in the full-load scenarios are unlikely to be reached in practice, hence a more realistic operating period would be in excess of 6 hours, but less than the 9 hours reported for an idle application. For many telehealth scenarios involving extended periods of operation, there are opportunities for recharging during use. The bandwidth required for the full-load scenarios amounts to approximately 21.6 MB per day. This equates to the bandwidth needed to stream a 3 minute YouTube video to the smartphone used in our experiments. The figures reported for transmission latency may be an issue for applications like athlete coaching. For such applications the keep-alive period can be reduced to lower latency and thereby improve responsiveness at the cost of higher bandwidth use.

6. Related Work

HomeCare [15], O'Brien et. al.'s wireless health sensors [16] and Wu et. al.'s RFID based solution [17] enable real-time data transmission over 3G/GPRS networks. The transmitted data are stored within the device, which forwards them to interested parties via a server. They lack support for bidirectional communication, thus preventing interested parties from providing real-time feedback to subjects or remotely configuring monitoring parameters.

Crilly et. al.'s work [18], the MultiTeled system [19] and Girao et. al.'s respiratory distress monitoring framework [20] all enable bidirectional communication to varying degrees. MultiTeled uses devices to capture and transmit bio signals to clients over GSM and satellite networks, the latter improving connectivity in areas that lack GSM coverage. However, the devices are bulky and hinder mobility [19]. Girao's solution notifies clients via SMS and allows PC-based administrative applications to configure mobile applications over limited-range Bluetooth. Crilly's solution supports storage and analysis of sensory data within a mobile device, with the responsibility for querying data being left to the remote clients [18]. The authors do not address fundamental issues, such as device discovery, maintaining connectivity with mobile users, and handling of concurrent client requests.

7. Conclusions

Given ongoing change in the healthcare landscape, telemedicine solutions are receiving increasing levels of interest. The needs of mobile telehealth applications are numerous, many of them conflicting with the basic characteristics of smartphone devices and wireless networking technologies. Meeting mobile telehealth requirements using smartphone technology requires much more than a standalone smartphone application. In this paper, we have described the application of Odin, a middleware platform for building mobile services, to the telehealth domain.

We have developed a mobile monitoring application comprising the Odin middleware and a smartphone augmented with a body sensor for vital sign capture. The application demonstrates that many of the needs of mobile telehealth solutions can be met. In particular, it has been shown to facilitate user interaction within acceptable timeframes; to offer bidirectional connectivity, enhanced with a rich set of communication primitives, between mobile users and remote clients; to allow for arbitrary data to be exchanged; and, of notable interest to telemedicine, for

²Network Monitor, available from Android Market at: <https://market.android.com/details?id=com.aob.android.mnm>

different stakeholders to be given different access rights and for communication to be secured. Furthermore, the use of Odin's surrogate-based architecture allows an application's function to be partitioned across the smartphone and surrogate entities, in this case offloading ECG-processing to a relatively well-resourced intermediary.

Based on evaluation, the combined middleware/application solution can operate, without the smartphone requiring to be recharged, for a period that is sufficient for many monitoring scenarios. Bandwidth use is dependent on the application being hosted by the middleware; in our experiments, high frequency streaming and messaging resulted in acceptable bandwidth consumption. The middleware is "best effort" with respect to connectivity and timely communication; the former being managed by automated and transparent vertical handover techniques. For timely communication, the middleware can be configured to be more responsive at the expense of using more bandwidth, but given the use of underlying packet switching network technology it is not possible to guarantee an upper bound for communication latency.

In the future, we intend to investigate the use of Odin in more collaborative telehealth applications that involve multiple subjects who may play the roles of both monitored subject and monitoring agent. We expect to leverage Odin's peer-to-peer communication facilities in realising such applications. Concerning the monitoring application presented in this paper, we will shortly be trialling this in a clinical setting involving cardiovascular patients. Further exploration of different telehealth application domains is planned, yielding new applications that will generate as yet unanticipated requirements for the underlying middleware platform.

References

- [1] J. Linkous, Telemedicine: An Overview, *Journal of Medical Practice Management* 18 (1) (2002) 24–7.
- [2] A. C. Norris, *Essentials of Telemedicine and Telecare*, John Wiley and Sons, Ltd., 2002.
- [3] P. H. J. Bellika, M. Johansen, Eight Challenges for Developing Telemedicine Applications, *The Journal on Information Technology in Healthcare* 6 (4) (2008) 295–302.
- [4] The Long-Term Outlook for Health Care Spending Washington, DC: Congressional Budget Office, <http://www.cbo.gov/doc.cfm?index=8758> (Aug. 2009).
- [5] T. Weerasinghe, I. Warren, Empowering Intermediary-based Infrastructure for Mobile Service Provisioning, in: *IEEE Asia-Pacific Services Computing Conference (APSCC)*, 2009, pp. 414–421.
- [6] R. M. Kleinpell, B. Avitall, Integrating Telehealth as a Strategy for Patient Management after Discharge for Cardiac Surgery: Results of a Pilot Study, *Journal of Cardiovascular Nursing* 22 (1) (2007) 38–42.
- [7] J. Polisen, Home telehealth for Diabetes Management: a Systematic Review and Meta-Analysis, *International Journal of Technology Assessment in Health Care*. 25 (3) (2009) 339–349.
- [8] A. Meads, A. Roughton, I. Warren, T. Weerasinghe, Mobile Service Provisioning Middleware for Multihomed Devices, in: *IEEE International Conference on Wireless and Mobile Computing, Networking and Communication*, IEEE Computer Society, Los Alamitos, CA, USA, 2009, pp. 67–72.
- [9] T. Weerasinghe, I. Warren, Odin: Context-Aware Middleware for Mobile Services, in: *6th World Congress on Services*, 2010, pp. 661–666.
- [10] P. Pawar, K. Wac, B. J. van Beijnum, P. Maret, A. V. van Halteren, H. Hermens, Context-Aware Middleware Architecture for Vertical Handover Support to Multi-homed Nomadic Mobile Services, in: *Proc. ACM Symposium on Applied Computing (SAC' 08)*, ACM, New York, NY, USA, 2008, pp. 481–488.
- [11] S. N. Srirama, M. Jarke, W. Prinz, MWSMF: A Mediation Framework Realizing Scalable Mobile Web Service Provisioning, in: *Proc. 1st International conference on Mobile Wireless Middleware, Operating Systems, and Applications (MOBILWARE' 08)*, ICST, Brussels, Belgium, Belgium, 2007, pp. 1–7.
- [12] I. Radovanovi, A. Ray, J. Lukkien, M. Chaudron, Facilitating Mobile Service Provisioning in IP Multimedia Subsystem (IMS) Using Service Oriented Architecture, in: *Proc. 5th International Conference on Service-Oriented Computing*, Springer-Verlag, 2007, pp. 383–390.
- [13] ZeroC, Inc, <http://www.zeroc.com> (2011).
- [14] H. Schmidt, A. Kohrer, F. J. Hauck, SoapME: A Lightweight Java ME Web Service Container, in: *Proc. 3rd Workshop on Middleware for Service Oriented Computing*, ACM, 2008, pp. 13–18.
- [15] M. Figueredo, J. Dias, Mobile Telemedicine System for Home Care and Patient Monitoring, in: *26th Annual International Conference on Engineering in Medicine and Biology Society*, 2004, pp. 3387–3390.
- [16] H. O'Brien, P. van de Ven, J. Nelson, A. Bourke, Smart Phone Interfaces to Wireless Health Sensors, in: *Proc. 12th IEEE International Conference on e-Health Networking Applications and Services*, 2010, pp. 180–186.
- [17] Y.-C. Wu, P.-F. Chen, Z.-H. Hu, C.-H. Chang, G.-C. Lee, W.-C. Yu, A Mobile Health Monitoring System Using RFID Ring-Type Pulse Sensor, in: *Proceedings of the Eighth IEEE International Conference on Dependable, Autonomic and Secure Computing*, IEEE Computer Society, Washington, DC, USA, 2009, pp. 317–322.
- [18] P. Crilly, V. Muthukkumarasamy, Using Smart Phones and Body Sensors to Deliver Pervasive Mobile Personal Healthcare, in: *Sixth International Conference on Intelligent Sensors, Sensor Networks and Information Processing*, 2010, pp. 291–296.
- [19] E. Kyriacou, S. Pavlopoulos, A. Berler, Multi-Purpose Healthcare Telemedicine Systems with Mobile Communication Link Support, *Journal of BioMedical Engineering OnLine* 2 (1) (2003) 7.
- [20] O. Postolache, P. Girao, M. Dias Pereira, G. Ferraria, N. Barroso, G. Postolache, in: *Instrumentation and Measurement Technology Conference, 2009. I2MTC '09*. IEEE.