

Uma Arquitetura Distribuída IoT para Gestão de Presença e Encontros de Pessoas em Ambientes Internos

Thiago Wallass N. Mendes¹, André A. Cardoso¹, Alysson C. Silva¹, Daniel S. Carvalho¹, Francisco J. da S. e Silva¹, Ariel S. Teles¹, Markus Endler², Marcos R. Júnior³

¹Laboratório de Sistemas Distribuídos Inteligentes
Universidade Federal do Maranhão (UFMA) – São Luís – MA – Brasil

²Laboratory for Advanced Collaboration (LAC)
Pontifícia Universidade Católica (PUC) – Rio de Janeiro – RJ – Brasil

³Faculdade de Ciência e Tecnologia (FCT)
Universidade Federal de Goiás (UFG) – Goiânia – GO – Brasil

{thiago.wallass, andre.cardoso, fssilva, ariel}@lstdi.ufma.br

Abstract. *Internet of Things applications for detecting people in indoor environments are becoming increasingly useful, especially in times of pandemic, when it is important to identify which people were close to each other in the same place and for how long. By considering this scenario, this paper presents a distributed software architecture that uses four components to provide services for storing and consulting data, meeting notifications, and identifying the devices involved. In a flexible way, beacons and Android devices are used to represent both people and physical spaces. In addition, the proposed architecture makes it possible to determine attendances in real time, calculate total stay time and check people meetings. The effectiveness of the proposed solution was demonstrated through an experimental evaluation.*

Resumo. *Aplicações de Internet das Coisas para a detecção de pessoas em ambiente internos tornam-se cada vez mais útil, principalmente em tempos de pandemia, em que é importante identificar quais pessoas estiveram próximas em um mesmo local e por quanto tempo. Considerando esse cenário, este artigo apresenta uma arquitetura de software distribuída que utiliza quatro componentes que proveem serviços para armazenamento e consulta de dados, notificações de encontros, e identificação dos dispositivos envolvidos. De maneira flexível, beacons e dispositivos Android podem ser usados para representar tanto pessoas quanto espaços físicos. Além disso, a arquitetura proposta possibilita determinar presenças em tempo real, calcular tempo total de permanência e verificar encontros de pessoas. A efetividade da solução proposta foi demonstrada através de uma avaliação experimental.*

1. Introdução

O paradigma de Internet das Coisas (IoT) objetiva conectar objetos do dia a dia à Internet de maneira a formar um sistema onde os mundos real e virtual se encontram e continuamente interagem [Borgia 2014]. A IoT pode ser utilizada no desenvolvimento de soluções para Cidades Inteligentes (CI). Uma CI utiliza tecnologias de IoT para transformar sua infraestrutura básica, otimizando seus recursos [Kon and Santana 2016]. Por

exemplo, prédios inteligentes podem possuir aplicações de controle de tráfego e acesso de ocupantes [Minoli et al. 2017]. Essa gestão de presença e encontros em prédios inteligentes provê informações sobre onde e por quanto tempo as pessoas passaram em cada local do prédio. Em um hospital, por exemplo, é possível identificar os locais onde estiveram os pacientes e profissionais, bem como calcular o tempo utilizado em cada etapa de atendimento.

Existem dificuldades na classe de aplicações de gestão de presença e encontros, principalmente devido à imprecisão com o uso de coordenadas geográficas em ambientes internos, impossibilitando o uso do GPS para identificar a localização exata dos ocupantes [Faragher and Harle 2015]. Outro desafio não trivial é manter uma base de dados com a localização de pessoas em cada instante de tempo, de maneira que também seja possível identificar suas permanências ou ausências em tempo real.

Este trabalho objetiva apresentar uma arquitetura distribuída capaz de servir de base para o desenvolvimento de aplicações IoT. Esta solução tem como foco a gestão de presença e encontros de pessoas em ambientes internos. Através de seus quatro componentes de software, com a utilização de *beacons Bluetooth Low Energy* (BLE) e dispositivos Android, é possível determinar a presença de pessoas em tempo real, manter um histórico dos dados e realizar buscas refinadas neles.

O restante desse artigo está organizado como segue. A Seção 2 aborda os conceitos fundamentais. A Seção 3 detalha os componentes de software integrantes da solução. A Seção 4 descreve uma avaliação da solução proposta e os resultados obtidos. Na Seção 5 são apresentados os trabalhos relacionados, bem como nossas contribuições. Por fim, a Seção 6 conclui o artigo.

2. Fundamentação Teórica

Esta seção aborda os conceitos fundamentais e tecnologias utilizadas no decorrer do artigo. Será apresentado o *middleware* M-Hub/CDDL, usado para a aquisição e distribuição de dados, e uma descrição detalhada sobre os *beacons* BLE.

2.1. O *Middleware* M-Hub/CDDL

Tendo como objetivo facilitar o desenvolvimento de aplicações de IoT, o *middleware* M-Hub/CDDL reúne conceitos e serviços de um *gateway* móvel (i.e., o M-Hub), que atua na aquisição de dados com base na descoberta e interação com objetos inteligentes [Talavera et al. 2015], e uma camada de distribuição de dados (i.e., o CDDL) [Gomes et al. 2017]. O CDDL é responsável por oferecer um meio de comunicação baseado em tópicos que media a comunicação entre aplicações consumidoras e serviços de objetos inteligentes detectados pelo M-Hub. Através do conjunto de interfaces oferecidas pelo M-Hub/CDDL é possível desenvolver aplicações voltadas para IoT que são capazes de detectar e se comunicar com objetos inteligentes embarcados com sensores e atuadores. Portanto, o M-Hub/CDDL é utilizado tanto na detecção e aquisição de dados de objetos inteligentes quanto na distribuição desses dados para outras instâncias do CDDL.

A Figura 1 ilustra os domínios de aplicação do M-Hub/CDDL. Conforme exibido no domínio *a*), o *middleware* é utilizado para: aquisição de dados oriundos de objetos inteligentes, repartir dados entre instâncias móveis do CDDL e para distribuir dados para

instâncias do CDDL que trabalham na nuvem. No domínio *b*), profissionais e aplicações específicas, através do M-Hub/CDDL, consomem os dados coletados.

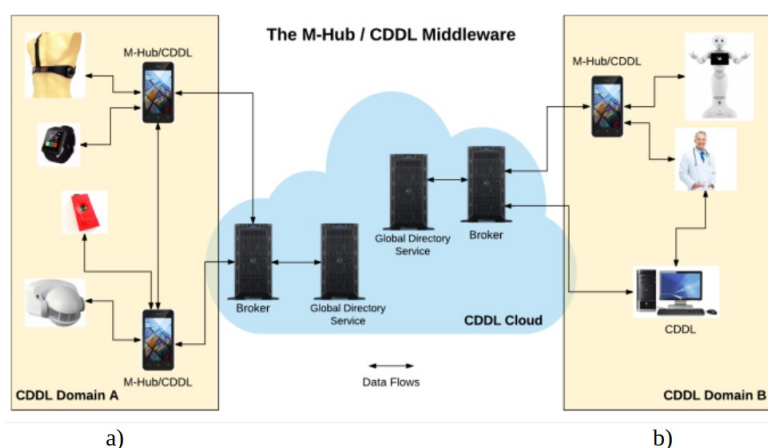


Figura 1. Visão geral do M-Hub/CDDL em um cenário de IoT [Gomes et al. 2017].

2.2. Beacons BLE

Os *beacons* BLE são dispositivos compostos por um *chipset Bluetooth*, uma bateria e uma antena. O funcionamento deste dispositivo se dá pela emissão regular de sinais de rádio, que são captados por aparelhos que estão com a interface *Bluetooth* ligada dentro de um raio de alcance do sinal. No entanto, os aparelhos não realizam tentativas de conexão com *beacons* BLE, como ocorre no *Bluetooth* clássico, ao contrário, eles detectam sua presença e recebem os pacotes de dados transmitidos. O raio de alcance de sinais do BLE é normalmente menor que o *Bluetooth* clássico, bem como menos dados são transmitidos. Devido a isso, o BLE traz benefícios como o baixo consumo de energia, demorando de meses a anos para terem suas baterias esgotadas [Kriz et al. 2016], além de serem dispositivos relativamente baratos.

3. Solução de Arquitetura de Software Distribuída

A solução proposta combina quatro componentes de software, com funções bem definidas, que podem ser utilizadas para o desenvolvimento de aplicações voltadas para a gestão de pessoas em prédios inteligentes. A arquitetura foi desenvolvida considerando os seguintes requisitos funcionais: (i) detectar a presença de pessoas em tempo real; (ii) manter uma base de dados que armazena os encontros; e (iii) fornecer consultas refinadas nesta base de dados. Também foram levados em conta os seguintes requisitos não funcionais: (i) consistência nos dados, registrando dados próximos aos reais; (ii) escalabilidade; e (iii) flexibilidade, de forma que possibilite o uso de seus componentes em diversos contextos.

3.1. Rendezvous

Para um melhor entendimento desta seção, os termos “encontro simples” ou “*rendezvous* simples” são definidos como um evento representado pela detecção de um objeto inteligente por um *gateway* M-Hub/CDDL. O modelo de um encontro apresentado na Figura 2 é composto por três identificadores únicos: o primeiro relacionado ao contexto geral da aplicação, e os outros dois relacionados ao objeto detectado, e ao M-Hub detector. O

modelo também recebe o *timestamp* do encontro e a intensidade do sinal. Uma sucessão de *rendezvous* simples pode ser resumida para gerar um *rendezvous* composto, representando o tempo total em que as duas entidades (i.e., o M-Hub e o objeto inteligente) permaneceram próximas.

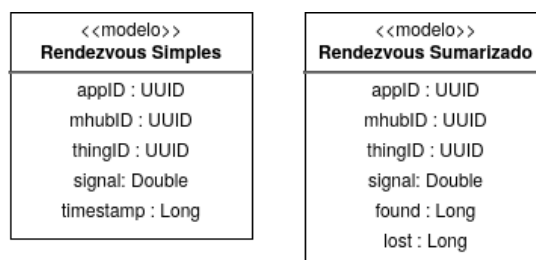


Figura 2. Modelo de *rendezvous* simples e composto.

3.2. HORYS

O HORYS (acrônimo para *Hub-Object Rendezvous Registry Service*) [Endler and e Silva 2018] é um serviço responsável por armazenar e fornecer consultas aos *rendezvous*. O CDDL é utilizado para receber as notificações de encontro e, ao ser notificado, o serviço registra o *rendezvous* simples e inicia o processo de sumarização dele. Neste processo, utiliza-se um valor limite chamado de frequência de funcionamento, sua função é ajudar no cálculo para determinar se os encontros estão acontecendo em tempo real.

Para encontrar *rendezvous* em tempo real, busca-se por um encontro, em que o *timestamp* seja maior que o *timestamp* atual subtraído do valor de frequência. Também é permitido realizar buscas considerando um *timestamp* arbitrário como atual, viabilizando determinar se objetos e M-Hubs estavam em contato naquela data e horários específicos.

A construção de um *rendezvous* composto ocorre da seguinte maneira: um primeiro *rendezvous* simples é enviado, cria-se um *rendezvous* composto com os campos *found* e *lost* possuindo o mesmo *timestamp* do *rendezvous* recebido. Os próximos *rendezvous* irão atualizar o campo *lost* do *rendezvous* composto com o novo *timestamp*. Caso um dos *rendezvous* seja recebido, com uma diferença de tempo maior que a frequência em relação ao encontro anterior, o *rendezvous* composto é encerrado. O evento que interrompe a sequência é considerado como primeiro de uma nova sucessão.

Nota-se que o HORYS não tem ciência do que representa os UUIDs de um encontro. Sua função é apenas registrar *rendezvous* simples e compostos. Dessa maneira, o serviço se torna flexível e pode ser utilizado em diversos tipos de soluções, deixando a critério de outro componente a identificação nominal dos encontros, o Serviço Semântico.

3.3. Serviço Semântico

Uma vez que os encontros estão armazenados no HORYS, existe a necessidade de se identificar as entidades envolvidas nos encontros. Este componente, denominado Serviço Semântico, tem como função mapear os relacionamentos entre pessoas, espaços físicos, M-Hubs e objetos. Através de uma base de dados relacional, este serviço consegue cadastrar e identificar os UUIDs presentes no componente anterior. Sua API é voltada apenas para identificação das entidades envolvidas nos encontros.

Para gerir presenças e encontros em espaços físicos, convencionou-se por padrão que pessoas portam M-Hubs e que espaços físicos possuem *beacons*. Atuando flexivelmente, o Serviço Semântico pode inverter a convenção: pessoas portam objetos, e espaços físicos possuem M-Hubs. Dessa forma, a configuração é deixada a critério da lógica de negócio utilizada.

3.4. Serviço Atendente

De maneira a não sobrecarregar os componentes já apresentados, foi concebido o Serviço Atendente. A função deste serviço é consultar as partes anteriores e fornecer buscas refinadas que sirvam às aplicações consumidoras do sistema. Este componente retorna os dados referentes à pesquisa substituindo os UUIDs pelos nomes. Dessa forma, é possível localizar pessoas e saber em quais locais elas estiveram.

Além disso, esse componente também introduz o conceito de encontros entre pessoas. Calculando a interseção de encontros entre pessoas e locais, é possível determinar se pessoas estiveram no mesmo local, ao mesmo tempo, e quanto tempo permaneceram juntas. A partir disso, pode-se tirar conclusões a respeito das interações entre elas. Por exemplo, a direção de um hospital poderá ter ciência de todos os profissionais que interagiram com um determinado paciente internado.

3.5. Serviço Android

Este componente é um serviço Android e tem a função de notificar encontros ao serviço HORYS. Esse serviço utiliza o *middleware* M-Hub/CDDL para detectar *beacons* BLE e notificar o serviço com os dados do encontro. Para evitar a notificação de objetos inteligentes não relacionados à arquitetura, na inicialização, o serviço requisita ao Serviço Semântico uma lista dos objetos cadastrados, e apenas estes são notificados. Para evitar que o serviço notifique o encontro de dois *beacons* próximos, é utilizada uma janela de seis detecções. Quando o sexto objeto cadastrado é detectado, o serviço apenas notifica o encontro com aquele de sinal mais forte.

3.6. Integração dos Componentes

Os componentes já apresentados possuem sólidas características que podem ser aproveitadas de maneira individual, porém o sistema foi desenvolvido para atuar em conjunto. A Figura 3 exibe uma visão geral dos componentes em funcionamento integrado. No lado esquerdo está o Serviço Android, que detecta os *beacons* BLE ou outros objetos inteligente e notifica o HORYS. No lado direito estão as aplicações consumidoras, que se comunicam com o Serviço Atendente para obter informações úteis para a gestão de presença e encontros. O Serviço Atendente se comunica com os Serviços HORYS e Semântico, a fim de montar a informação a ser entregue.

4. Avaliação

De forma a avaliar a solução proposta, elaborou-se um cenário de testes que compreende toda a arquitetura e possui ênfase no sistema de registro das presenças. Para isso, desenvolveu-se uma aplicação Android que faz uso da arquitetura distribuída IoT proposta. Os principais pontos avaliados foram: a consistência dos dados registrados e a disponibilidade do serviço Android. Para efetuar os testes, foram utilizados *smartphones*

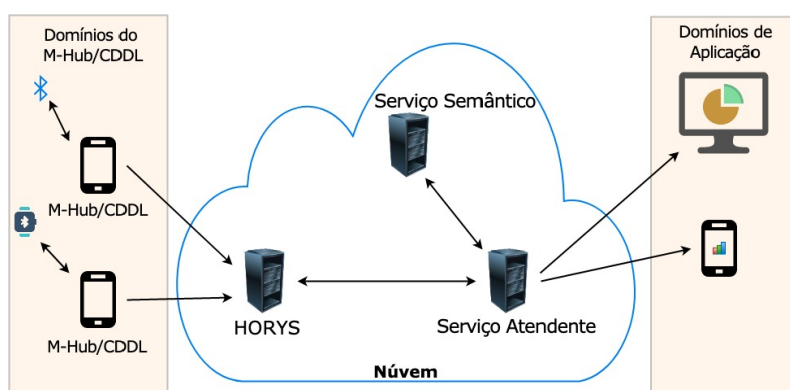


Figura 3. Visão geral dos componentes.

Android, como *beacon* foi utilizado uma pulseira *Mi Smart Band 4*. Deve-se ressaltar que ao longo da avaliação o dispositivo Android permaneceu com a tela apagada.

Durante os testes, o *beacon* foi posicionado no centro de uma sala (i.e., um ambiente interno) e uma pessoa utilizou este espaço físico normalmente. A pessoa utilizou o sistema para registrar o tempo passado (i.e., a presença) no ambiente durante três dias. Além disso, ao entrar ou sair da sala, o horário era registrado manualmente pela pessoa. O objetivo deste cenário de testes foi verificar se o tempo de permanência real no ambiente, registrado manualmente, foi compatível ao tempo contabilizado pela aplicação desenvolvida com a arquitetura proposta. Antes da execução do teste os relógios do HORYS e da pessoa foram sincronizados.

Conforme apresentado na Tabela 1, os resultados obtidos mostram que os valores registrados pelo sistema são compatíveis com os tempos reais de permanência. O tempo total computado pelo sistema apresentou valores altos, obtendo-se aproximadamente 98.36% do valor real.

Tabela 1. Resultados da Avaliação - Tempo Real x Tempo Computado.

	Valor Real (hh:mm:ss)	Valor Computado (hh:mm:ss)	Valor Perdido (hh:mm:ss)	Valor Computado (%)
14/06/20	03:13:00	03:10:05	00:02:55	98,48
15/06/20	03:20:00	03:16:51	00:03:09	98,42
16/06/20	07:14:00	07:06:35	00:07:25	98,29
Total	13:47:00	13:33:31	00:13:29	98,36

Os resultados obtidos constatarem que o sistema pode ser utilizado para detecção em tempo próximo ao real e determinação do tempo total de permanência em espaços físicos. Além disso, também verificou-se que o Serviço Android consegue permanecer muitas horas seguidas em funcionamento, sendo possível seu uso em situações reais.

5. Trabalhos Relacionados

Para gerenciar recursos de um prédio inteligente, Corna et al. [Corna et al. 2015] apresentaram uma solução que utiliza *iBeacons* para detecção de pessoas. Os *beacons*, que transmitem pacotes de dados continuamente, são distribuídos em salas específicas. Paralelamente, uma aplicação captura o dado e notifica o servidor do sistema de gerenciamento.

Executando algoritmos de classificação, o sistema é capaz de obter as localizações dos dispositivos com precisão e, a partir disso, é possível automatizar tomadas de decisões (e.g., desligar aparelhos de ventilação ao verificar que não há pessoas presentes).

A fim de determinar a localização em tempo real de funcionários de uma empresa, Syafrudin et al. [Syafrudin et al. 2018] apresentaram uma arquitetura composta por *beacons* BLE, uma aplicação que atua como *gateway* e um servidor para armazenamento de dados. Os *beacons* são posicionados em salas, a aplicação no *smartphone* do funcionário notifica o servidor. Através de uma aplicação web, é possível consumir os dados do servidor e visualizar o movimento dos funcionários nos postos de trabalho.

Chang et al. [Chang et al. 2016] propõem uma solução para detecção de presença através do uso de pulseiras *Bluetooth*. O usuário utiliza pulseiras de identificação, tornando desnecessário o uso de *smartphones*. O protocolo padrão do *beacon*, modificado pelos autores, permitiu a criação de uma rede, em formato de cadeia, em que foi possível a transmissão de dados entre *beacons*. No entanto, um computador com interface *Bluetooth* conectado à Internet foi requerido, atuando como *gateway* e enviando os dados a um servidor remoto. O sistema foi aplicado em um parque temático, onde os *beacons* foram distribuídos nas rotas de visitas.

O presente artigo, atuando no contexto de CI, especificamente Prédios Inteligentes, apresenta uma solução para a gestão de presença e encontros em ambientes internos, sendo possível determinar presenças em tempo real, calcular tempo total de permanência e verificar encontros entre pessoas. Como principal contribuição deste trabalho, podemos citar sua flexibilidade e capacidade de determinar intersecções de encontros. Corna et al. [Corna et al. 2015] e Syafrudin et al. [Syafrudin et al. 2018] apresentaram configurações que *beacons* são posicionados em espaços físicos, e *smartphones* são utilizados como *gateway*. Chang et al. [Chang et al. 2016], utiliza pulseiras com *Bluetooth* para representar pessoas. A abordagem proposta por este artigo possibilita o uso das duas abordagens: pessoas portando M-Hubs e espaços físicos possuindo objetos inteligente, e vice-versa. Esse recurso aumenta a aplicabilidade da solução proposta demonstrando sua flexibilidade, possibilitando o seu uso em diferentes contextos. Nossa solução também destaca-se pela capacidade de calcular encontros entre pessoas, podendo informar que grupos de pessoas estiveram interagindo entre si. Esse recurso é muito útil em períodos de quarentena, como a ocasionada pelo coronavírus, quando há a necessidade de se monitorar pessoas infectadas pelo vírus, ou com suspeita de infecção.

6. Conclusão

O presente artigo, apresentou uma arquitetura distribuída que pode ser utilizada para o desenvolvimento de soluções de IoT. Através de quatro componentes independentes, é possível construir aplicações para detecção de presenças em tempo real, determinação do tempo total de permanência em locais e verificar a existência de interações entre pessoas. A flexibilidade dos componentes se revela como a maior contribuição deste trabalho, em que é possível utilizar a arquitetura com as duas configurações possíveis, considerando *beacons* e *smartphones* como pessoas ou espaços físicos. Além disso, a partir da arquitetura proposta, é possível desenvolver outros tipos de soluções. Por exemplo, uma aplicação que monitora quanto tempo um aluno permanece nas dependências de seu laboratório, ou para descobrir se houve reuniões entre alunos.

Através do cenário de testes, demonstrou-se a efetividade do sistema. Os encontros foram notificados corretamente ao servidor, os dados armazenados foram consistentes e os métodos de busca demonstraram-se ser amplos. Como perspectivas futuras, pode-se citar um estudo direcionado para o uso de Processamento de Eventos Completos (CEP). O HORYS, funcionando como um processador de eventos (i.e., *rendezvous*) poderia se beneficiar do uso de regras CEP, tornando possível aumentar sua capacidade de recebimento de eventos, melhorando sua eficiência no processamento e até disparando notificações customizadas aos usuários do sistema. Levando em conta a privacidade dos usuários, também cita-se o desenvolvimento de uma camada de segurança para o *middleware* utilizado, garantindo integridade e confidencialidade dos dados.

Referências

- Borgia, E. (2014). The internet of things vision: Key features, applications and open issues. *Computer Communications*, 54:1–31.
- Chang, C.-m., Li, S.-C., and Huang, Y. (2016). Building bluetooth beacon-based network for spatial-temporal data collection. In *Proceedings of the International Conference on Communication and Information Systems*, pages 91–95. ACM.
- Corna, A., Fontana, L., Nacci, A., and Sciuto, D. (2015). Occupancy detection via ibeacon on android devices for smart building management. In *Design, Automation Test in Europe Conference Exhibition (DATE)*, pages 629–632.
- Endler, M. and e Silva, F. S. (2018). Past, present and future of the contextnet iomt middleware. *Open Journal of Internet Of Things (OJIOT)*, 4(1):7–23.
- Faragher, R. and Harle, R. (2015). Location fingerprinting with bluetooth low energy beacons. *IEEE Journal on Selected Areas in Communications*, 33(11):2418–2428.
- Gomes, B. D. T. P., Muniz, L. C. M., Da Silva e Silva, F. J., Dos Santos, D. V., Lopes, R. F., Coutinho, L. R., Carvalho, F. O., and Endler, M. (2017). A middleware with comprehensive quality of context support for the internet of things applications. *Sensors*, 17(12).
- Kon, F. and Santana, E. F. Z. (2016). Cidades inteligentes: Conceitos, plataformas e desafios. In *Congresso da Sociedade Brasileira de Computação*, pages 2–49.
- Kriz, P., Maly, F., and Kozel, T. (2016). Improving indoor localization using bluetooth low energy beacons. *Mobile Information Systems*, 2016.
- Minoli, D., Sohraby, K., and Occhiogrosso, B. (2017). Iot considerations, requirements, and architectures for smart buildings - energy optimization and next-generation building management systems. *IEEE Internet of Things Journal*, 4(1):269–283.
- Syafrudin, M., Lee, K., Alfian, G., Lee, J., and Rhee, J. (2018). Application of bluetooth low energy-based real-time location system for indoor environments. In *Proceedings of the 2nd International Conference on Big Data and Internet of Things*, page 167–171. ACM.
- Talavera, L. E., Endler, M., Vasconcelos, I., Vasconcelos, R., Cunha, M., and d. S. e. Silva, F. J. (2015). The mobile hub concept: Enabling applications for the internet of mobile things. In *2015 IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops)*, pages 123–128.