PRISER: Sistema para Gerenciamento de Notificações com Suporte a Privacidade de Dados

Luis A. Silva[1], **Valderi R. Q. Leithardt**[1,2], **Wemerson D. Parreira**[1], **Cláudio F. R. Geyer**[2]

[1] Universidade do Vale do Itajaí
Laboratório de Sistemas Embarcados e Distribuídos (LEDS)
Itajaí - SC, Brasil, 88302-901

[2]Universidade Federal do Rio Grande do Sul (UFRGS)
Instituto de Informática - GPPD
Porto Alegre, Brasil, 91501-970

luis.silva@edu.univali.br, {valderi, parreira}@univali.br

geyer@inf.ufrgs.br

Resumo. *Com o crescente número de dispositivos móveis que recebem notificações diariamente, é necessário gerenciar a variedade de informações produzidas. Novos dispositivos inteligentes são desenvolvidos todos os dias com a capacidade de gerar, enviar e exibir mensagens sobre seus status, dados obtidos e informações relativas a outros dispositivos. Consequentemente, o número de notificações recebidas por um usuário está aumentando e a tolerância a elas pode diminuir em pouco tempo. Com isso, se faz necessário o desenvolvimento de um sistema de gerenciamento e controles de notificações. Para tanto, este artigo propõe um sistema de gerenciamento de notificações e gerenciamento de alertas focado em ambientes e perfil de usuários, aplicando critérios de privacidade dos dados, fundamentado em uma arquitetura denominada PRISER, composta por três módulos que foram testados com uso dispositivos móveis.*

Abstract. *With the growing number of mobile devices receiving daily notifications, it is necessary to manage the variety of information produced. New smart devices are developed every day with the ability to generate, send, and display messages about their status, data, and information about other devices. Consequently, the number of notifications received by a user is increasing and their tolerance may decrease in a short time. With this, it is necessary to develop a management system and notification controls. In this context, this work proposes a notification and alert management system called PRISER. Its focus on user profiles and environments, applying data privacy criteria.*

1. Introdução

As evoluções tecnológicas no âmbito urbano têm permitido a integração de rede de sensores e dispositivos juntamente com os cidadãos presentes no contexto diário. Esses dispositivos são capazes de coletar e processar dados e de se comunicar por meio de uma rede [Goudos et al. 2017]. Embora hoje o termo seja utilizado de forma bem mais

abrangente, incluindo aplicações de saúde, logística, segurança, agricultura, entre outros, o objetivo principal é fazer com que os computadores capturem as informações do mundo real sem intervenção humana contínua [Rodrigues et al. 2017].

Os avanços das tecnologias de sistemas microeletrônicos, unindo as novas comunicações de redes sem fio, resultaram no desenvolvimento de dispositivos cada vez menores e com poder de processamento considerável, surgindo assim, a Internet das Coisas (IoT). A acentuada heterogeneidade de sistemas e dispositivos e as restrições de recursos impostas dificultam a aplicação das técnicas convencionais de segurança e privacidade relacionadas a IoT, levando à necessidade de sistemas de controle e gerenciamento específicos [Viel et al. 2018].

Com isso, a IoT é vista como uma premissa com grande potencial para integrações entre usuário e o ambiente, e um ambiente formado por tais dispositivos e sensores é chamado de ambiente inteligente. As informações presentes nesses ambientes são compartilhadas entre aplicações e plataformas, tendo por premissa atingir a interoperabilidade de dispositivos [Arasteh et al. 2016]. A capacidade funcional dos dispositivos inteligentes está diretamente relacionada à eficácia na comunicação, sendo que a principal tecnologia utilizada atualmente é a de comunicação sem fio. Com a evolução dessas tecnologias, suas atribuições devem ser devidamente organizadas e classificadas para contribuir com o cenário que estão inseridas.

Com o aumento do uso de dispositivos móveis, os computadores não são mais a única fonte de interrupções, o que leva à necessidade de um sistema de gerenciamento que aborde múltiplos dispositivos. Conforme reportado em [Mehrotra et al. 2016], no ano de 2016, um usuário já de dispositivo móvel recebia uma média 100 notificações diárias. Esse elevado influxo de notificações repercute negativamente junto ao usuário e sua a tolerância a esse tipo de mensagem tende a cair. Nesse sentido, [Okoshi et al. 2017] apresentam um sistema de notificações com base no contexto do usuário que age como uma camada entre recepção de notificação e a entrega ao usuário.

No âmbito de notificações, são utilizados recursos visuais, sonoros e hápticos (vibrações) com propósito de orientar a atenção do usuário sobre uma informação instantânea [Okoshi et al. 2017]. De acordo com essas funcionalidades e com o intuito de garantir a comunição entre os usuários e dispositivos presentes em ambientes inteligentes, é usual adaptar requisitos. Para tanto, segundo [Weber et al. 2018] são considerados os seguintes critérios para ambientes inteligentes e IoT: (*i)* a redução de custos, (*ii)* a melhoria da utilização, (*iii)* a comunicação relacionada ao usuário ou outro dispositivo inserido no ambiente (*iv*), o controle e gerenciamento da privacidade.

Para tanto, este trabalho contribui com a definição de um modelo para gerenciamento e controle de alertas e notificações para ambientes e dispositivos voltados para cenários IoT. Também integra aplicativos e dispositivos com base no módulo PRISER (*Privacy Services*). Este trabalho é uma melhoria e continuidade do modelo de controle e gerenciamento de privacidade para ambientes ubíquos proposto por [Leithardt et al. 2016]. Para um melhor entendimento e organização, este artigo esta dividido em sete seções. Na Seção II são apresentados os trabalhos relacionados. A Seção III é dedicada a descrição do modelo PRISER. Na Seção IV é descrito o Protótipo e Resultados Obtidos. Por fim, na Seção V, apresentamos as conclusões e trabalhos futuros.

2. Trabalhos Relacionados

Conforme a literatura pesquisada, sua maioria apresenta contribuições relacionadas ao controle de privacidade direcionado ao usuário e seus dispositivos heterogêneos. Em [da Costa et al. 2008], os autores apontam diversos desafios, dentre os quais o tratamento de privacidade na computação pervasiva por meio de mensagens e notificações.

Por outro lado, em [Qin et al. 2014], os autores focam em serviços para tornar os dispositivos mais silencioso. A solução DND (*Do-Not-Disturb* - Não perturbe) utiliza técnicas de aprendizado de máquinas para identificar o relacionamento entre o contexto atual de um usuário e o modo não perturbe do dispositivo. Com isso, a experiência anterior do usuário é utilizada como base, identificando se o usuário esta disponível ou não.

As notificações exclusivas para smartphones utilizando (*push notifications*) são abordadas em [Pan et al. 2015, Gudla and Bose 2016, Cho et al. 2016]. Essas propostas convergem dos dispositivos móveis e da IoT e tratam como o sistema pode lidar com o envio para múltiplos dispositivos e ainda tratar a comunicação permeando dispositivo para dispositivo (M2M - *machine-to-machine*) e a interação humana. Entretanto, a implementação do sistema de gerenciamento de notificações apresentado por [Fraser et al. 2016] define usuários e notificações, modelados a partir de uma abordagem chamada *info-bead*, sendo desenvolvido em conjunto à um algoritmo que utiliza lógica *Fuzzy*, aplicando regras e critérios definidos por sua modelagem. As abordagens de Okoshi, Nakazawa e Tokuda (2014) e Pielot et al. (2014) deixam claro que deve ser utilizado um método de decisão para avaliar o momento de interrupção do usuário. Por outro lado, a abordagem de Liu, Jiang e Striegel (2014) esclarece quanto a utilização da aproximação de dispositivos para o envio de notificações. Em Shirazi et al. (2014) e Pan et al. (2015) apontam o uso de um ambiente multi-dispositivos para melhorar os processos.

A proposta SmartNotifications [Corno et al. 2015], é preliminar e atende apenas para prever qual dispositivo deve ser selecionado para entregar a notificação. No entanto, o conjunto de dados de notificação utilizado para treinar os algoritmos e avaliar o resultado do sistema é parcialmente sintético e assume que os dados disponíveis para a notificação são explícitos. Em contraste, o sistema de gerenciamento proposto por Fraser, Yousuf e Conlan (2016) propõem prever o momento mais oportuno para o notificação a ser entregue ao usuário utilizando um conjunto de dados obtidos a partir de usuários reais. Apenas a derivação abstrata é utilizada pelo gerenciador para prever o tempo de entrega, esta abordagem é utilizada da mesma forma em Zhang, Liu e Wang (2016), com intuito de proteger a privacidade da localização do usuário, o qual teve seus dados coletados. Foram analisadas as informações sobre as implementações das soluções de gerenciamento de notificações e serviços de localização.

Dentre os trabalhos relacionados pesquisados, a maioria abordaram localização e o gerenciamento de notificações, boa parte dos trabalhos não abordam as características de privacidade do ambiente em que o usuário está inserido. Com isso, o presente trabalho propõe um modelo de gerenciamento para notificações e alertas, tendo em vista tornar o ambiente informativo e dinâmico de forma a controlar parâmetros de privacidade. Sendo assim, o perfil do usuário, sua localização, o tipo de ambiente, critérios de tempo, prioridade e as preferências do usuário serão considerados para definir o envio da mensagens e/ou alertas com base no controle e gerenciamento individual.

3. PRISER

A solução proposta fundamenta-se em um *middleware* de controle e gerenciamento de privacidade UbiPri [Leithardt et al. 2016], o qual difere das demais soluções existentes por prover um modelo genérico de controle e gerenciamento de privacidade para ambientes. O *middleware* UbiPri possui uma estrutura dividida em componentes, cada um deles com suas características específicas, sendo empregadas conforme a necessidade. Um de seus componentes refere-se à serviços, este é chamado **PRISER**, o qual é responsável pelo sistema de gerenciamento de notificações (SGN).

O controle e gerenciamento de notificações leva em consideração a privacidade do usuário em relação ao ambiente. O papel dos módulos está em definir e compartilhar suas regras e parâmetros com o Sistema de Gerenciamento de Notificações (SGN) conforme apresentado na Fig. 1. O SGN utiliza como base a localização que o usuário se encontra, suas preferências e também define os meios para envio de notificações. As características do meio para envio de mensagens são sensíveis à situação, conforme especificações definidas em [Fraser et al. 2016]. Para isso, o processo de envio de notificações foi fundamentado em critérios, tais como: tipo de ambiente, perfil do usuário, tempo para a entrega e prioridade. Os tipos de ambientes, foram classificados como: público, privado, restrito e personalizado, de acordo com definições de [Leithardt et al. 2016].

Com base nos critérios para o gerenciamento do envio de alertas é possível gerenciar o envio de mensagens de forma dinâmica, em várias configurações de envio. Com isso, o nível de severidade ou criticidade pode ser atribuído à notificação do usuário com base nos padrões do sistema (por exemplo, níveis de severidade ou criticalidade associados a certos eventos comuns que podem exigir atenção do usuário). Ou ainda personalizando critérios (por exemplo, os usuários que podem modificar os padrões do sistema, criar eventos personalizados associados a níveis de gravidade ou de importância definidos pelo usuário etc.). O SGN utiliza o modelo *publisher/subscriber*, na qual será consumido de um *broker*, neste caso agindo como um intermediador, responsável por armazenar e enfileirar os eventos a serem notificados, conforme mostrado na Fig. 1.

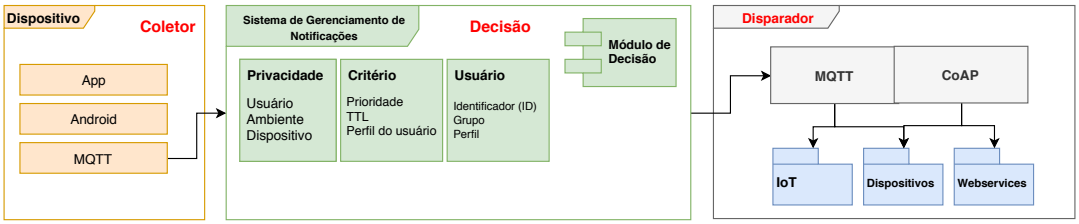

Figura 1. Arquitetura do sistema de gerenciamento de notificações (SGN)

Em um aspecto semelhante, ainda deve ser possível atribuir um valor de tempo de vida para a notificação (TTL). No caso de um dispositivo sem acesso à internet, em um determinado ambiente, deve ser possível enviar notificações de média e alta prioridade com outras configurações, como por exemplo um envio de SMS ou ainda em extremos casos, uma ligação ou alerta.

3.1. Modulo Coletor

O sistema proposto recebe as notificações provenientes de diferentes fontes externas e armazena no primeiro módulo (Coletor). Para isso, informações presentes nas

notificações referentes aos critérios e a privacidade são de domínio exclusivo do modulo de decisão. A coleta tem função de *buffer*, armazenando temporariamente os dados, enquanto estão sendo movidos e processados pelo módulo de decisão.

Para tanto, foi implementado um aplicativo para coleta e registro de notificações executado em segundo plano em dispositivos móveis. Os requisitos para o aplicativo são registros confiáveis e discretos em segundo plano e suporte para a maioria das versões Android. Este serviço, depois de conceder permissão ao usuário, é executado permanentemente em segundo plano no dispositivo e recebe retornos de chamada quando uma notificação é adicionada ou removida do sistema. Versões recentes do Android melhoraram significativamente as informações fornecidas por essa API, por exemplo, fornecendo informações se uma notificação foi removida pelo usuário ou pelo próprio aplicativo de notificação. A API está disponível desde o Android 4.3, que executa em 96,40% dos smartphones Android. Esse serviço fornece informações sobre quais aplicativos acionaram e removeram notificações. Também disponibiliza o conteúdo do texto, os níveis de prioridade, os padrões de vibração, entre outros atributos adicionais, a aplicação é representada na Fig. 2.

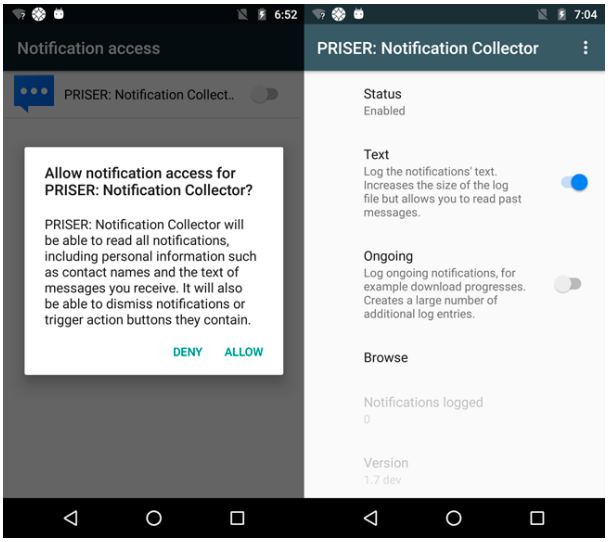

Figura 2. Aplicativo Notification Collector

3.2. Modulo de Decisão (Decisison)

Neste módulo as notificações são enviadas para o segundo modulo, tendo como principal função a tomada de decisões, recebendo informações relacionadas a privacidade do ambiente, dispositivo ou ainda do usuário. Um fluxograma apresentado pela Fig. 3 é utilizado para determinar a rota das notificações.

O módulo de decisão também é responsável por receber informações de critérios no contexto do usuário (ex. localização, status, atividade atual), bem como informações relevantes à notificação como o tempo de vida. Os critérios tem funções importantes no SGN, O fluxograma utiliza as informações com a finalidade de escolher os melhores dispositivos e as melhores tipos de alertas (ex. vibração, som ou sinal luminoso), para apresentar as notificações recebidas.

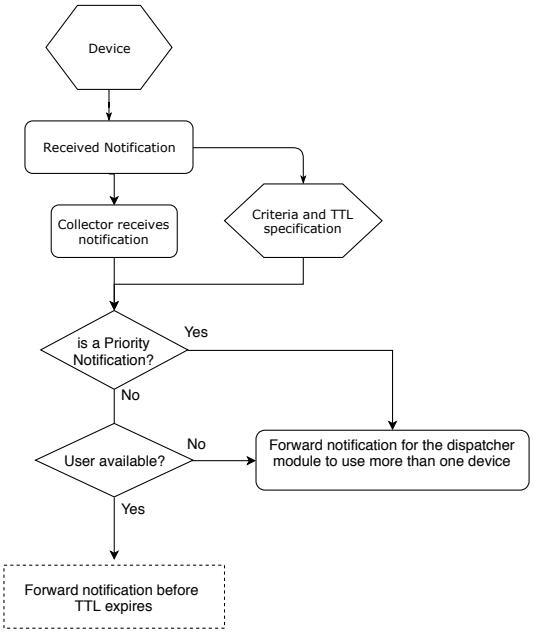

Figura 3. Fluxograma SGN

3.3. Modulo Disparador (Dispatcher)

O modulo disparador é responsável por gerenciar e adaptar as notificações aos dispositivos de destino escolhidos e enviar as mensagens. Ao receber notificações endereçadas em apenas um dispositivo, isso gera ajustes de alguns fatores. O primeiro fator, é que o usuário deve estar sempre carregando dados ou estar próximo do dispositivo. Segundo fator refere-se à conectividade, o dispositivo pode ficar desconectado ou mesmo sem uma bateria, é utilizado uma arquitetura com vários dispositivos, como mostrado na Fig. 4.

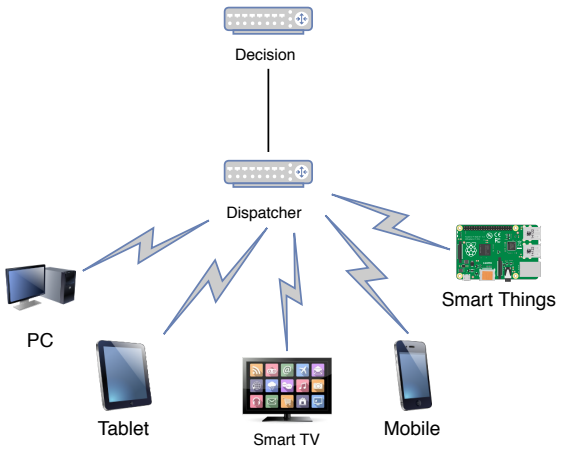

Figura 4. Disparados para múltiplos dispositivos

O módulo disparador também possui a funcionalidade de controlar o tipo de dispositivo que será enviado o alerta, de acordo com as características do hardware e suas definições de localização, tipo de alerta e mensagem enviada e/ou recebida.

4. Protótipo e Resultados Obtidos

Para avaliar a melhor abordagem da coleta de notificações, uma versão preliminar do módulo responsável foi desenvolvida e testada usando um conjunto de dados reduzidos. Para isso, o módulo de coleta de notificações instalado em dois dispositivos móveis com sistema operacional Android.

Além dos dispositivos utilizados para os testes, também foi efetuada uma comparação via computador pessoal, sendo utilizado o Android Debug Bridge (ADB) para conectar com o dispositivo. Nesta etapa foi focado no desempenho da aplicação e seu funcionamento. Os testes foram executados em intervalos de 24h, com o objetivo de avaliar a quantidade de notificações recebidas intervalos de duas horas, também foi analisado o uso de memória, CPU e quantidade de armazenamento utilizada.

Durante o período de teste os dispositivos receberam notificações referentes à aplicativos de mensagens de texto (SMS), notificações do sistema operacional e ainda lembretes de chamadas perdidas. Durante toda execução, a aplicação se manteve em segundo plano e coletou diversas notificações de sistema, mensagens e ligações, gerando eventos que apareceram no painel de notificação. O SGN utiliza as informações originadas por meio do próprio sistema operacional. Essas notificações são registradas com data e hora do evento, o aplicativo também registra os seguintes atributos:

- Nome do evento que originou a notificação (e-mail, maquina de lavar, WhatsApp e usuários ou grupos individuais);
- Estado do evento em termos de qual tipo de notificação está sendo enviada (tela ativada, tela desativada e desbloqueio de tela);
- Ação executada pelo usuário (notificação recebida e/ou removida ou ainda respondida);
- Conteúdos da mensagens;
- Tempo e dados do evento.

Conforme ilustrado pela Fig. 5 um objeto JSON é composto por todas as notificações obtidas, contendo os itens mencionados.

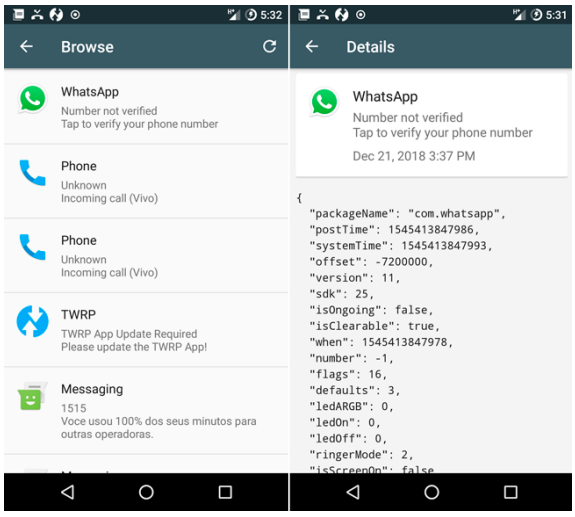

Figura 5. Lista de notificações armazenadas

As notificações são então armazenadas na memória persistente do dispositivo e podem ser acessadas apenas pelo usuário com privilégios de administrador, assim como em alguns sistemas operacionais.

4.1. Dispositivos

Os dispositivos utilizados nos testes foram selecionados por serem modelos comuns no uso diário, com processamento e memória entre a média dos dispositivos mais utilizados. Além disso, o computador usado para os testes não possuem um alto poder de processamento, simulando um usuário comum. Para testar o desempenho da aplicação sob alta carga de notificações foi preciso utilizar usuários virtuais. No entanto, a simulação de um grande número de usuários virtuais consome CPU, memória e outros recursos, podendo causar sobrecarga do sistema e gerar resultados inválidos.

Devido a certas limitações do sistema operacional, não é possível utilizar um computador pessoal para tal tarefa. Para isso, nesse experimento foram utilizados 30 (trinta) usuários virtuais, por meio de 10 (dez) contas de usuários em cada dispositivo. A partir versão 6 do sistema operacional Android, é possível ter mais de uma conta vinculada ao seu dispositivo. Os dispositivos utilizados foram do modelo XT1635-02 (Motorola). Para implementar os testes foi utilizado também um computador (Backend) com processador Intel® Core i5 2,7GHz com 8GB de memória RAM, este controlava os 2 dispositivos por meio da biblioteca ADB, conforme ilustrado na Fig. 6.

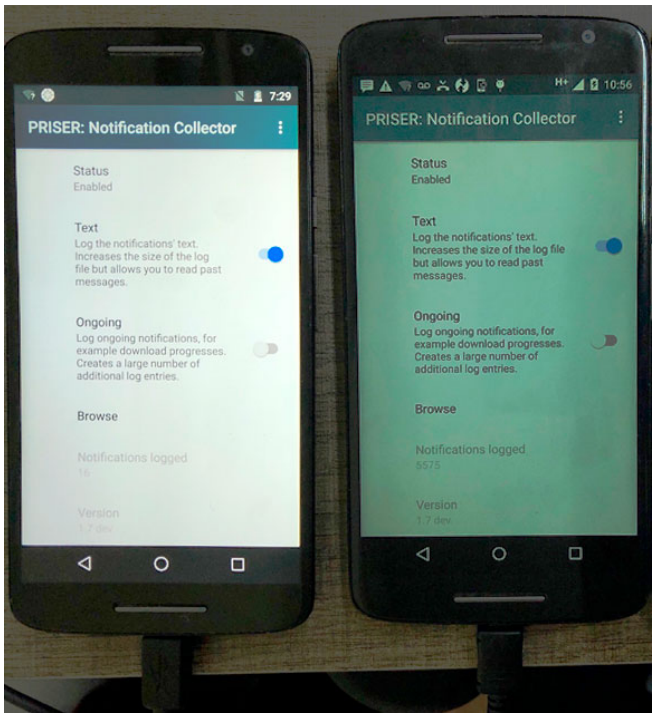

Figura 6. Dispositivos utilizados para testes

Os resultados obtidos puderam comprovar os conceitos apresentados no decorrer do trabalho, também foram fundamentados na taxonomia desenvolvida em [Silva et al. 2018]. Utilizamos regras e definições conforme a literatura pesquisada.

5. Conclusões e trabalhos futuros

No decorrer deste trabalho, foi possível identificar a importância em utilizar os critérios mencionados, levando em consideração a hierarquia do usuário, da privacidade de acordo com o ambiente e a hierarquia atribuída, também sendo possível definir o tipo de alerta. Como consequência, as definições para o sistema de gerenciamento de notificações foram devidamente relacionadas à privacidade do usuário. Desta forma, este trabalho contribuiu no desenvolvimento de uma aplicação com tratamento de diferentes tipos de notificações.

Além disso, foi possível garantir o envio e/ou recebimento de mensagens relevantes de acordo com regras previamente definidas. Os trabalhos relacionados apresentam limitações e não definiram uma arquitetura e/ou modelo com contribuição para trabalhar com múltiplos dispositivos.

Para tanto, apresentamos uma arquitetura dividida em três módulos principais para gerenciar as notificações recebidas. A principal contribuição deste trabalho foi o módulo de decisão implementado, gerenciando e decidindo quem deve receber a notificação, em qual(is) dispositivo(s), momento e com qual(is) modo(s) (vibração, som, luz). Os resultados permitiram validar o protótipo desenvolvido com base nas regras de privacidade.

Em trabalhos futuros estão sendo desenvolvidos outras formas de decisão por meio de aprendizado de máquina para incrementar o módulo de decisão. Outra linha de pesquisa é relacionada ao módulo disparador, também estao sendo testadas variações entre os protocolos MQTT, CoAP e OSGP com objetivo de comparar diversas mensagens em diferentes dispositivos e tipos de mensagens. Para execução de envio de alertas em larga escala estão sendo simulados, os primeiros resultados apontaram a necessidade de otimização do código. Outro item que está sendo implementado é a segurança das mensagens enviadas, testes utilizando criptografia para envio e recebimento de mensagens estão sendo realizados em conjunto com o protocolos mencionados anteriormente.

6. Agradecimentos

Este estudo foi financiado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Codigo 001 e teve suporte por meio do projeto de cooperação internacional Control and History Management Based on the Privacy of Ubiquitous Environments - Brasil / Portugal e também do projeto SMART-SENT: Desenvolvimento de Plataforma para Análise de Big Data em Aplicações de Cidades Inteligentes do Instituto de Informática da UFRGS.

Referências

Arasteh, H., Hosseinnezhad, V., Loia, V., Tommasetti, A., Troisi, O., Shafie-khah, M., and Siano, P. (2016). Iot-based smart cities: A survey. In *2016 IEEE 16th International Conference on Environment and Electrical Engineering (EEEIC)*, pages 1–6.

Cho, C., Kim, J., Joo, Y., and Shin, J. (2016). An approach for coap based notification service in iot environment. In *Information and Communication Technology Convergence (ICTC), 2016 International Conference on*, pages 440–445. IEEE.

Corno, F., Russis, L. D., and Montanaro, T. (2015). A context and user aware smart notification system. pages 645–651.

da Costa, C. A., Yamin, A. C., and Geyer, C. F. R. (2008). Toward a general software infrastructure for ubiquitous computing. *IEEE Pervasive Computing*, 7(1):64–73.

Fraser, K., Yousuf, B., and Conlan, O. (2016). A context-aware, info-bead and fuzzy inference approach to notification management. In *Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), IEEE Annual*, pages 1–7. IEEE.

Goudos, S. K., Dallas, P. I., Chatziefthymiou, S., and Kyriazakos, S. (2017). A survey of iot key enabling and future technologies: 5g, mobile iot, sematic web and applications. *Wireless Personal Communications. 2017.* .

Gudla, S. K. and Bose, J. (2016). Intelligent web push architecture with push flow control and push continuity. In *Web Services (ICWS), 2016 IEEE International Conference on*, pages 658–661. IEEE.

Leithardt, V., Rolim, C., Rossetto, A., Borges, G., Sá Silva, J., and Geyer, C. (2016). The classification of algorithms for privacy management in ubiquitous environments. *8º SBCUP-Simpósio Brasileiro de Computação Ubíqua e Pervasiva-XXXVI CSBC-Congresso da Sociedade Brasileira de Computação. 2016.*

Mehrotra, A., Pejovic, V., Vermeulen, J., Hendley, R., and Musolesi, M. (2016). My phone and me: understanding people's receptivity to mobile notifications. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pages 1021–1032. ACM.

Okoshi, T., Tsubouchi, K., Taji, M., Ichikawa, T., and Tokuda, H. (2017). Attention and engagement-awareness in the wild: A large-scale study with adaptive notifications. In *Pervasive Computing and Communications (PerCom), 2017 IEEE International Conference on*, pages 100–110. IEEE.

Pan, Z., Liang, X., Zhou, Y. C., Ge, Y., and Zhao, G. T. (2015). Intelligent push notification for converged mobile computing and internet of things. In *Web Services (ICWS), 2015 IEEE International Conference on*, pages 655–662. IEEE.

Qin, Y., Bhattacharya, T., Kulik, L., and Bailey, J. (2014). A context-aware do-not-disturb service for mobile devices. In *Proceedings of the 13th International Conference on Mobile and Ubiquitous Multimedia*, pages 236–239. ACM.

Rodrigues, V. F., Correa, E., da Costa, C. A., and da Rosa Righi, R. (2017). On exploring proactive cloud elasticity for internet of things demands. In *2017 XLIII Latin American Computer Conference (CLEI)*.

Silva, L. A., Leithardt, V. R., Dazzi, R. S., and Silva, J. M. S. S. (2018). Um modelo taxonômico de notificações e alertas aplicado a privacidade de dados. In *Anais...* Escola Regional de Alto Desempenho - ERAD 2018, Sociedade Brasileira de Computação.

Viel, F., Silva, L. A., Leithardt, V. R. Q., and Zeferino, C. (2018). Internet of Things: Concepts, Architectures and Technologies. In *2018 13th IEEE International Conference on Industry Applications (INDUSCON)*, pages 1–8. DOI:10.1109/INDUSCON.2018.

Weber, D., Voit, A., and Henze, N. (2018). Notification log: An open-source framework for notification research on mobile devices. UbiComp '18, pages 1271–1278, New York, NY, USA. ACM. DOI:10.1145/3267305.3274118.