Internet of Things: Survey About Management Solutions

Alexandre Fava, *DCC, UDESC,* Daniel Camargo, *DCC, UDESC,* and Mateus Boiani, *DCC, UDESC.*

***Resumo*—O paradigma da Internet das Coisas (*Internet of Things* (IoT)) vem ganhando espaço no cenário moderno da comunicação sem fio. Ainda há alguns desafios, como o gerenciamento de recursos, devido a heterogeneidade de tecnologias de hardware e software. Para suprir estes desafios, surgem soluções de gerenciamento de dispositivos IoT, que possibilitam facilitar aos usuários e desenvolvedores alguns requisitos considerados importantes que permeiam este paradigma. O presente trabalho tem por objetivo realizar um levantamento qualitativo destas soluções de monitoramento de IoT, permitindo sintetizar uma comparação simplificada destas soluções em relação aos requisitos de IoT.**

Keywords*—*Internet of Things, Internet das Coisas, Rede, Gerenciamento, Hardware, Software, ...

I. Introdução

A Internet das Coisas (tradução do termo em inglês *Internet of Things* (IoT)) é um novo paradigma que vem ganhando espaço no cenário moderno da comunicação sem fio (*wireless*). Kevin Ashton cunhou o termo em 1999, todavia a ideia de objetos do cotidiano embarcados com sensores, microcontroladores já era utilizada a mais de uma década, através dos termos "computação ubíqua" e "computação pervasiva" [1]. De acordo com relatório da Gartner [2], em 2015 haviam 4.9 bilhões de dispositivos conectados (um aumento de 30% em relação a 2014) e a *International Data Corporation* (IDC) estima que aproximadamente 32 bilhões de dispositivos de IoT estarão conectados à Internet até o ano de 2020 [3]. O conceito básico deste paradigma é a presença pervasiva de uma grande variedade de objetos físicos embarcados com sensores e atuadores que, através de esquemas de endereçamento único, são capazes de interagir uns com os outros para atingir objetivos comuns [4], [5].

Normalmente, a IoT pode ser composta por qualquer tipo de objeto comum, antes considerados "burros"devido sua incapacidade de comunicação, tornando-se inteligentes com as novas habilidades de processamento e interação entre si e seus usuários. Podem ser utilizados *gateways* para prover comunicação entre protocolos específicos de IoT ou diretamente com a Internet, enviando dados para serem processados em servidores remotos, normalmente com o auxílio da computação em nuvem. Esta abordagem permite que uma grande quantidade de dados (Big Data) possa ser processada em conjunto, possibilitando que sejam interpretadas em tempo real [6]. Estes objetos são capazes de capturar diversos tipos de informações e reagir a estímulos externos [4], permitindo o surgimento de uma variedade de aplicações que poderão se beneficiar dos novos tipos de dados, serviços e operações disponíveis. A IoT é uma das principais tecnologias emergentes que contribuem para concretizar novos domínios de aplicação das Tecnologias da Informação e Comunicação (TICs), a exemplo do domínio de cidades inteligentes, no qual o uso de tecnologias avançadas de comunicação e sensoriamento visa prover serviços de valor agregado para os órgãos administrativos de tais cidades e para seus respectivos cidadãos [7], [8].

Os recentes avanços tecnológicos, tais como a Rede de Sensores Sem Fio (RSSF), comunicação móvel e computação ubíqua, possibilitaram o aprimoramento do conceito da IoT. Entretanto, há alguns desafios ainda a serem superados na tentativa de facilitar o uso desse paradigma. Um destes desafios está relacionado a necessidade de gerenciamento dos seus recursos. Isto ocorre devido a heterogeneidade de tecnologias de hardware e software que compõe a abordagem da IoT. A recente expansão da IoT exige um gerenciamento abrangente de todos os seus componentes, como por exemplo, a rede de comunicação, os nós de processamento, uso de sensores/atuadores e dados coletados. Outros desafios estão relacionados à necessidade de escalabilidade desses ambientes, em termos de número de dispositivos conectados, à necessidade de gerenciar tais dispositivos e sua rede de comunicação, possibilitando disponibilidade contínua com um volume crescente de dados. Todavia, as soluções encontradas para alavancar a maturidade no que se refere ao gerenciamento da IoT são ainda escassas.

Especificamente em gerenciamento de IoT, são encontradas poucas abordagens na literatura, não apenas por ser um paradigma relativamente novo, mas devido ao fato de seus componentes terem recém conquistado sua maturidade. Podem ser citados como exemplos de componentes, conceitos de *System on a Chip* (SoC), Identificação por Rádio Frequência (RFID), Big Data. De forma subjetiva, estes componentes também devem ser gerenciados, como por exemplo:

É possível verificar que existem uma série de requisitos que podem ser levantados com os componentes que devem ser gerenciados em uma solução de IoT. O artigo está estruturado da forma como segue. Na Seção II é abordada a metodologia utilizada na pesquisa. Os requisitos para uma solução de gerenciamento de IoT são descritos na Seção III. Os principais protocolos encontrados em IoT na Seção IV. A análise das soluções encontradas é feita na Seção V e comparadas sobre quesitos comuns entre si na Seção VI. Por fim as considerações finais são dadas na Seção VII.

A. Fava, D. Camargo e M. Boiani são acadêmicos do Departamento de Ciência da Computação (DCC). Universidade do Estado de Santa Catarina – UDESC. Joinville-SC. E-mail: {afava, dcamargo, mboiani}@udesc.br

Manuscrito recebido em 29 de março de 2016; Revisado em 29 de março de 2016.

Tabela I. COMPONENTES GERENCIÁVEIS DE IOT.

Componentes gerenciáveis	Exemplos
Dispositivos, sensores e atuadores	Especificação de hardware, localização, estado
Fontes de energia	Oferta e demanda de energia
Dados coletados	Big Data, dados heterogêneos
Processamento local	SoC, Microcontrolador (MCU), Sistemas embarcados
Identificação única	IPv4/IPv6, endereço MAC, RFID
Qualidade da rede	Fluxo da rede, interferências, distância
Protocolos de comunicação	Tratamento de pacotes, topologias, segurança, escalabilidade

II. METODOLOGIA DE PESQUISA

O presente trabalho tem por objetivo reportar algumas das principais soluções de gerenciamento para IoT, através de pesquisa qualitativa, identificando tendências, descrevendo desafios para a difusão da IoT, apresentando questões abertas de pesquisa e suas futuras direções, através da compilação de uma lista de referência, permitindo assistir a pesquisadores da área. A metodologia aplicada a este trabalho visa reportar o estado atual das principais soluções de gerenciamento para IoT, examinando-os individualmente com o intuito de identificar as tendências e descrever os desafios encontrados para difusão deste paradigma. Para a pesquisa qualitativa, foi realizado um levantamento dos trabalhos relacionados através de fontes científicas, como jornais e conferências, bem como em alguns livros que apresentem alguma contribuição positiva para o gerenciamento de IoT. A lista completa destes locais inclui:

- ACM Digital Library;
- CAPES, Portal de Periódicos;
- IEEE Xplore;
- ScienceDirect;
- Google Scholar;
- SpringerLink; e
- Livros.

A pesquisa bibliográfica foi feita com uso das chaves de pesquisa em inglês: “Management of Internet of Things”, “Management IoT” e, em português, “Gerenciamento de Internet das Coisas ” e ‘Gerenciamento de IoT”. Foram encontradas quinze soluções que fazem o gerenciamento de IoT, citados em artigos e que tratam das categorias listadas na Tabela I de forma satisfatória.

III. REQUISITOS PARA O GERENCIAMENTO DE IOT

As soluções de gerenciamento de IoT devem satisfazer a um conjunto de requisitos visando atender às necessidades de aplicações, desenvolvedores e usuários, bem como alguns desafios que surgem nesse cenário [9], [10].

(i) Interoperabilidade;
(ii) Descoberta e configuração de dispositivos;
(iii) Ciência de contexto;
(iv) Escalabilidade;
(v) Gerenciamento de grandes volumes de dados;
(vi) Segurança;
(vii) Adaptação dinâmica; e
(viii) Interfaces de alto nível.

A interoperabilidade entre os diversos dispositivos e plataformas disponíveis no ambiente é um dos principais desafios do paradigma de IoT devido à necessidade de integração de um grande número de dispositivos e sua heterogeneidade tanto em termos de hardware quanto de software, protocolos, formatos de dados, etc. O requisito de descoberta e configuração deve permitir que aplicações sejam criadas de maneira mais rápida, gerando maior valor agregado para os usuários, deve ter a capacidade de fornecer informações de localização e estado do dispositivo, desconectar algum dispositivo roubado ou não reconhecido, atualizar software embarcado, modificar configurações de segurança, modificar remotamente configurações de hardware, localizar um dispositivo perdido, apagar dados sensíveis de dispositivos, e até mesmo configurar a interação entre dispositivos. A ciência de contexto diz respeito às informações do estado do objeto, seus vizinhos e sua localização, por exemplo, necessitam ser coletadas e processadas com o objetivo de efetuar ações ou reagir a estímulos com base nos dados extraídos. Soluções de gerenciamento de IoT devem então ser responsáveis pela coleta, gerenciamento e processamento de informações de contexto providas por múltiplas fontes, liberando as aplicações e usuários da tarefa de manipulá-las e tornando transparente tal manipulação. O requisito de escalabilidade refere-se à capacidade de assimilar um número crescente de dispositivos e requisições e funcionar corretamente, mesmo em situações de uso intenso. Devido à sua facilidade de provisão e uso de recursos computacionais, que podem ser alocados e liberados sob demanda, o paradigma de computação em nuvem tem surgido como uma solução promissora para endereçar a questão da escalabilidade em ambientes de IoT, proporcionando o surgimento da chamada “nuvem de coisas” (ou *cloud-of-things*, em Inglês).

O aumento do número de dispositivos gera um aumento no volume de dados transmitidos através da rede, nesse contexto, o gerenciamento de grandes volumes de dados é um requisito importante, permitindo o acompanhamento da demanda de coleta e análise de dados e, consequentemente, prover respostas, decisões e atuações de maneira eficiente. Nesse contexto, surgem desafios quanto ao processamento de transações, que podem ser realizadas na própria solução de gerenciamento, onde soluções baseadas em Big Data e computação em nuvem têm surgido como uma potencial solução para esses desafios, a fim de permitir lidar com um imenso volume de dados e não estruturado. Estratégias de segurança devem manter a integridade e privacidade dos dados disponibilizados, além de proteger os dispositivos e os recursos expostos à rede. O requisito de adaptação dinâmica deve garantir a disponibilidade e qualidade das aplicações durante a sua execução, sendo especialmente importante para aplicações em domínios críticos, a exemplo de aplicações de saúde, mantendo-se disponíveis e funcionando adequadamente, coletando, analisando e reagindo a mudanças no contexto em que elas e objetos a ela conectados estão inseridos. Por fim, uma interface de alto nível deve possibilitar ao usuário/desenvolvedor visualizar todos as informações de todos os dispositivos de forma simples e completa, sintetizando os requisitos anteriores através de gráficos, mapas físicos/topológicos e demais informações que facilitem a compreensão e o gerenciamento dos dispositivos

de IoT

Mesmo com a busca destes requisitos em soluções de gerenciamento de IoT, os protocolos que são comumente utilizados neste paradigma já tratam de alguns destes requisitos de forma simplificada. Com isso em mente serão discutidos alguns destes protocolos e com quais requisitos eles estão relacionados.

IV. Protocolos de Comunicação

Protocolo é uma convenção que controla e possibilita a comunicação entre dois sistemas computacionais, podendo ser implementados por hardware, software ou por uma combinação dos dois. De maneira simplificada, um protocolo pode ser definido como "as regras que governam" a sintaxe, semântica e sincronização da comunicação. Uma pilha de protocolos denota uma combinação específica de protocolos que trabalham de forma conjunta, já um modelo de referência é uma arquitetura de software que lista cada um dos níveis e os serviços que cada um deve oferecer.

O Modelo *Open System Interconection* (OSI) (formalizado em 1983) é um modelo de referência da ISO, que tem como principal objetivo ser um padrão para protocolos de comunicação de diversos sistemas, visando a garantia da comunicação fim-a-fim. O modelo clássico OSI é subdividido em sete níveis e é atualmente o mais utilizado para conceitualizar pilhas de protocolo. São eles: Camada Física, de Enlace de Dados, Rede, Transporte, Sessão, Apresentação e Aplicação, porém para o paradigma de IoT as Camadas Física e de Aplicação possuem destaque devido a sua especifidade em relação aos requisitos levantados na Seção III.

A. Camada física para redes sem fio

1) IEEE 802.11: O Instituto de Engenheiros Eletricistas e Eletrônicos (IEEE) é uma organização Americana responsável pela publicação de diversas normas internacionais (RFC), aprovou em 1997 o padrão para a *Wireless Local Area Network* (WLAN) ("Rede Local Sem Fios", em português), identificado como IEEE 802.11, mas também conhecido como Wi-Fi (nome do órgão que certifica a compatibilidade dos dispositivos). Desde a primeira aprovação, este padrão vem ganhando novos protocolos, de forma a se adequar às nova demandas, como por exemplo: 802.11a, 802.11b, 802.11g, 802.11n, 802.11ah, dentre outros. Estas protocolos possuem diferenças entre si, variando requisitos como a faixa de frequência de operação, largura de banda, velocidade de transferência, alcance e consumo de energia.

Dentro do conceito de IoT, destaca-se o protocolo 802.11ah, também conhecido como Wi-Fi HaLow, a ser aprovado em meados de 2016. Sua faixa de frequência opera abaixo de 1 GHz, permitindo um baixo consumo de energia em relação aos demais protocolos da família 802.11, e provê a criação de um grande número de dispositivos em rede, comunicando-se com largura de banda entre 150Kbps e 18Mbps em uma distância de até 1 quilômetro entre rádios. Com esta nova abordagem, estima-se que aplicações de *smart homes* tenham uma comunicação nativa com roteadores residenciais, descartando o uso de *gateways* e permitindo que os dispositivos comuniquem-se diretamente com as aplicações em nuvem. Estima-se que esta novidade aumentará a interoperabilidade entre dispositivos abrigados sob a mesma WLAN.

2) IEEE 802.15.4: O padrão IEEE 802.15.4 define um protocolo *Wireless Personal Area Network* (WPAN) para interconexão de dispositivos com reduzidas características de largura de banda, potência, complexidade, onde a frequência de operação define o alcance em uma rede sem fio. Além disso, as redes WPAN envolvem pouca infraestrutura, isso permite que sejam implementadas soluções de baixo custo e de eficiência energética otimizada. O padrão utiliza três faixas de frequência possíveis: 868Mhz, 915Mhz e 2.4Ghz, adequando-se às normas da maioria dos países. De forma geral, o alcance do rádio fica entre 10 e 100 metros, em algumas exceções podendo extrapolar este limite. A taxa de dados é de 250 Kbps, 40 Kbps, e 20 Kbps.

Normalmente a distribuição dos nós divide-se entres as topologias: estrela, árvore e malha. A identificação única de até 64 bits é formada por endereçamento similar ao MAC. Além disso, é possível realizar a detecção de energia do nó, além da indicação da qualidade do Link.

O IEEE 802.15.4 é um padrão que vem sendo amplamente adotado, utilizado até pouco tempo quase exclusivamente com o protocolo ZigBee na camada de aplicação. Contudo a alternativa de utilizá-lo com o IPv6, usando uma camada de adaptação chamada de 6loWPAN, surgiu há algum tempo, e vem ganhando adeptos desde então. Com estes novos protocolos na camada de aplicação do IEEE 802.15.4, o conceito de IoT está tornando-se cada vez mais viável no cotidiano.

B. Camadas superiores

1) ZigBee: O ZigBee é o protocolo executado em cima do IEEE 802.15.4, padronizado no final de 2004 quando a ZigBee Alliance anunciou sua disponibilidade para o mercado. Seu propósito inicial foi possibilitar aplicações de domótica (automação residencial), eliminando os cabos e utilizando-se de uma comunicação confiável, porém de baixo consumo de energia, proporcionando estender a vida útil das baterias dos dispositivos. A especificação do ZigBee define as camadas de rede e aplicação do modelo OSI e ainda o serviço de segurança entre ambas. Nestas camadas estão especificados mecanismos de conexão de um dispositivo à rede, identificação e armazenamento dos dispositivos vizinhos em uma tabela de roteamento, além de outras funcionalidades.

Um rádio ZigBee pode possuir uma pilha com quatro camadas básicas: Física, Rede, Transporte e Aplicação sendo a camada física e MAC de responsabilidade da norma IEEE 802.15.4. A camada Física (PHY) é responsável pela captura do sinal de onda de RF, fornecendo parâmetro que caracteriza a qualidade do sinal recebido. A camada PHY possui a função de ativar ou desativar o transceptor através da detecção de energia. A camada de Transporte foi desenvolvida para abrigar topologias múltiplas, como no caso do controle de energia, controle de acesso aos canais de rádio que evita as colisões. A camada também implementa o padrão AES (Advanced Encryption Standard) que tem a responsabilidade de criar as rotinas de segurança.

Camada de Rede é responsável pelo nível de rede ou comunicação, fazendo uso de algoritmos que permite balancear os custos oriundos da camada de aplicação e consumo energético, otimizando assim o desempenho. Camada de Aplicação responde pela gestão e suporte das aplicações e está dividida em três componentes: suporte à aplicação, ZDO Zigbee Device Object e a função de dispositivo de rede. Estão previstos três tipos de dispositivos lógicos: ZigBee coordenador, ZigBee roteador e ZigBee final.

Com o levantamento das características do protocolo ZigBee, mostra-se cobrir a maior parte dos requisitos levantados inicialmente (Seção III), considerando que estas características condizem apenas com as possibilidade de gerenciamento através do protocolo.

2) 6LoWPAN: O protocolo 6LoWPAN (*IPv6 over Low power Wireless Personal Area Networks*), consistem em uma adaptação do protocolo IPv6 com o protocolo IEEE 802.15.4, usado para suportar endereçamento IP. Sua funcionalidade é fazer a compressão dos cabeçalhos IPv6 para sua transmissão na rede Internet, ou seja, no *gateway*. O 6LoWPAN pode ser implementado em dispositivos fabricados para determinada função como por exemplo: *smartphones*, sensores pessoais, automação predial, logística, transporte, medidores de energia elétrica inteligentes, infraestrutura de redes, e muito mais. Isso acaba permitindo a interconexão de todos estes dispositivos diretamente com a Internet.

A integração de uma rede IEEE 802.15.4 usando 6LoWPAN com a Internet ou outras redes IP é feita de forma muito simples, já que se trata do mesmo protocolo. A implementação de um *gateway* envolve a camada rede, como mostrado na figura a seguir. A solução do 6LoWPAN começa na camada de aplicação e vai se fragmentando passando por todas as camadas abaixo até atingir a camada PHY, responsável por acomodar as necessidades de interfaces de baixo custo. Após isso o *gateway* faz a solução inversa, subindo as camadas até atingir a camada responsável pelo IP, logo após a camada física se encarrega da transmissão e assim que a mensagem chegar ao destinatário os protocolos finalizam na camada de aplicação.

3) SNMP: Foi no início de 1980 que o protocolo SNMP começou a ser desenvolvido, suas siglas são nada mais, nada menos que um acrônimo para Simple Network Management Protocol, podendo ser traduzido como Protocolo Simples de gerenciamento de redes. Este protocolo tem como premissa à flexibilidade e a facilidade de implementação. Sua especificação está contida na RFC 1157.

Em redes de computadores, normalmente as tarefas mais complexas de gerenciamento são realizadas pelo SNMP. Sua arquitetura é baseada no conceito de agente e gerente, explicando de maneira bem simples, o agente é o dispositivo da rede que esta sendo gerenciado, é dele que são coletadas qualquer tipo de informação ou dados. O gerente no caso é o dispositivo que realiza essas coletas, solicitando e analisando essas informações, que podem ir desde: uso de memória, temperatura, quantidade de pacotes IPs recebidos e muito mais que o agente esteja habilitado a fornecer, limitando as informações adquiridas somente ao hardware. Normalmente o gerente é um servidor centralizado, enquanto os agentes podem ser qualquer tipo de dispositivo que esteja conectado a rede.

O SNMP é responsável pela comunicação entre os agentes e o gerente, definindo os formatos e os tipos de pacotes que serão trocados nessa comunicação, além de interpretar informações recebidas dos agentes. Essas informações não são muito intuitivas para que o ser humano consiga interpreta-las de maneira rápida e prática, por isso é utilizado um software que receba essas informações e as converta em tabelas e gráficos para a analise dessas informações se torne o mais próxima do natural.

O SNMP atua ao lado de outras duas estruturas: SMI (Structure of Management Information) e MIB (Management Information Base). O primeiro define a estrutura básica das informações que serão coletadas, nessa etapa são especificados a criação de nomes, tipos e a forma como as informação serão codificadas para serem enviadas ao gerente, por exemplo. Com os dados devidamente estruturados pela SMI, a MIB pode começar a realizar sua função (dependendo pode haver mais de uma MIB). A MIB possui uma estrutura em árvore que contém os objetos gerenciáveis de um determinado dispositivo de rede. Essa estrutura não tem limites e, de acordo com a necessidade, pode ser expandida e atualizada. Um objeto gerenciável é uma visão abstrata de um recurso de um dispositivo da rede. Ele corresponde a uma estrutura de dados e operações obtida a partir do modelamento dos recursos desse dispositivo de rede. Figura 1: Árvore MIB parcial a partir da raiz

É importante salientar que o SNMP é divido por versões, no caso da SNMPv1, sua finalidade era de gerenciar dispositivos na arquitetura TCP/IP, já a segunda versão trouxe melhorias na leitura dos valores no agente e a terceira versão acabou trazendo melhorias na parte da segurança.

C. Considerações sobre protocolos de comunicação em IoT

Diversos protocolos de comunicação utilizados em IoT já fazem algum tipo de gerenciamento em relação à segurança das informações e qualidade da rede, porém indicam que ainda se faz necessário haver um gerenciamento mais específico devido a grande heterogeneidade de dispositivos e protocolos existentes. Após análise inicial dos principais protocolos possíveis de serem utilizado no paradigma de IoT, serão descritos as soluções encontradas que permitem o gerenciamento de dispositivos IoT.

V. Soluções de gerenciamento de IoT

Estão sendo utilizados como parâmetros de comparação os oito requisitos inicialmente levantados e destacadas algumas das principais características destas soluções. Durante a descrição das soluções de gerenciaento, estes requisitos podem ou não serem citados, devido a ausência de algumas destas informações.

A. EcoDif

O Ecossistema Web de Dispositivos Físicos ou simplesmente EcoDiF [1] é uma plataforma Web para conectar dispositivos e produtos, a fim de fornecer funcionalidades de controle,

[1] http://ubicomp.nce.ufrj.br/ecodif

visualização, processamento e armazenamento dos dados em tempo real. Esta plataforma pretende atuar como núcleo central de um ecossistema de IoT, oferecendo inúmeros serviços de software focados na conectividade entre dispositivos, além de oferecer também serviços de aplicação e serviços de apoio. Um exemplo da arquitetura da solução EcoDif é exemplificada na Figura 1.

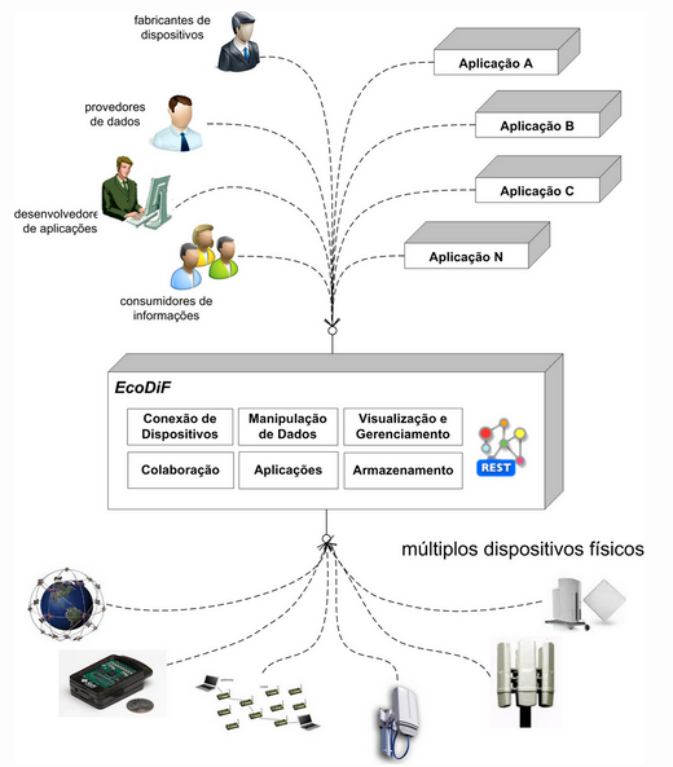


Figura 1. Modelo da Arquitetura do EcoDif.

A EcoDiF é composta de todos esse módulos: Conexão de Dispositivos, Manipulação de Dados, Visualização e Gerenciamento, Colaboração, Aplicações e Armazenamento. A EcoDiF emprega os princípios *REpresentational State Transfer* (REST) para disponibilizar as funcionalidades dos dispositivos físicos na Web utilizando duas abordagens. Na primeira, são implantados servidores Web embarcados em dispositivos e as funcionalidades desses dispositivos são disponibilizadas na forma de recursos RESTful. Na segunda abordagem, quando um dispositivo não possui recursos de hardware suficientes para executar um servidor embarcado, é possível utilizar outro dispositivo como ponte para disponibilizar as funcionalidades do dispositivo na Web através de uma interface RESTful. Estima-se que em breve a EcoDiF fornecerá uma *Application Programming Interface* (API) aberta que poderá ser utilizada por indivíduos, instituições de pesquisa e empresas para acessar os serviços providos pela plataforma.

Demonstrações práticas do potencial de integração de dispositivos heterogêneos em diferentes cenários reais já forma realizadas, como por exemplo: monitoramento de CPDs, monitoramento de dutos de água/gás, experiência participativas e monitoramento de metabolismo corporal. Estas são apenas as provas de conceito, porém os limites da plataforma vão muito além, como: aplicações de monitoramento ambiental, de monitoramento de infraestrutura pública, acompanhamento de trânsito e condições da estrada, compartilhamento de dispositivos de sensoriamento entre laboratórios acadêmicos e muito mais.

B. INRIA ARLES

A INRIA ARLES é uma Arquitetura de Software e Sistemas Distribuídos desenvolvida na França no Instituto Nacional de Pesquisa em Informática e Automação (INRIA). A ferramenta [2]provê a implementação de infraestruturas de gerenciamento para sistemas difusos interoperáveis, porém por ser uma plataforma muito recente, ainda está sendo estudada.

Essa plataforma de gerenciamento adota uma arquitetura orientada a serviços a fim de abstrair sensores e atuadores como serviços a fim de ocultar suas heterogeneidades, e depende fortemente de uma base de conhecimento que contenha informações sobre sensores, atuadores, fabricantes, conceitos físicos, unidades físicas, modelos de dados, modelos de erros, entre outros. Para tratar os desafios decorrentes da grande escala e da profunda heterogeneidade inerente à IoT, a proposta concentra sua contribuição em três núcleos: descoberta probabilística, composição aproximadamente ótima e estimação automatizada. Juntas, essas três características permitem à plataforma responder as requisições recebidas enquanto gerencia complexas relações entre precisão e tempo, memória, processamento e restrições energéticas dos dispositivos.

C. RestThing

A solução RestThing é uma infraestrutura de serviços Web baseada em REST, cujo objetivo é ocultar a heterogeneidade de dispositivos e prover um modo de integrar dispositivos em aplicações Web. A plataforma visa permitir que desenvolvedores criem aplicações usando princípios REST, combinando recursos físicos e Web, de modo que dispositivos e informações Web são ambos representados como recursos e manipulados por uma interface uniforme no estilo REST. Um exemplo da arquitetura da solução RestThing é dada na Figura 2.

Os elementos do RestThing são: aplicações; API RESTful; provedor de serviço; camada de adaptação; dispositivos embutidos, e; recursos Web. A API RESTful permite transmitir dados entre os sensores que usam IP, os *gateways*, o servidor Web e aplicações Web. Por utilizar o REST, a plataforma se torna bastante similar com a outra plataforma denominada EcoDiF.

D. S3OiA

A solução S[3]OiA é um acrônimo para Smart Spaces and Smart Objects Interoperability Architecture, é uma arquitetura orientada a serviço para integração de diferentes tipos de objetos, físicos ou virtuais usando tuplespace para expressar semanticamente informações dos dispositivos integrados pela plataforma e possibilitar a representação destes.

[2]https://www.rocq.inria.fr/arles

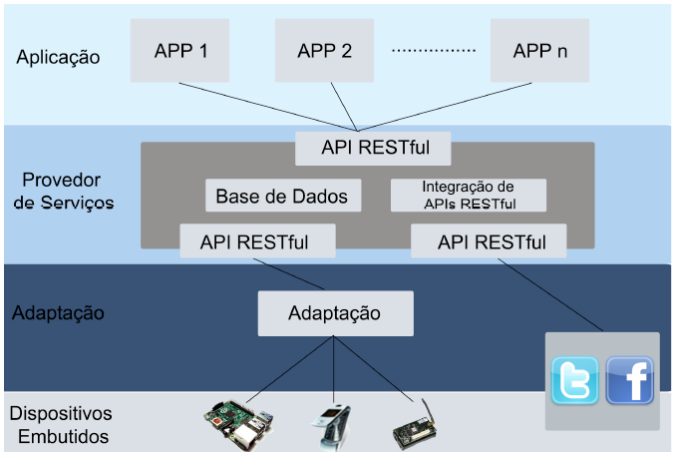


Figura 2. Modelo de Infraestrutura do RestThing.

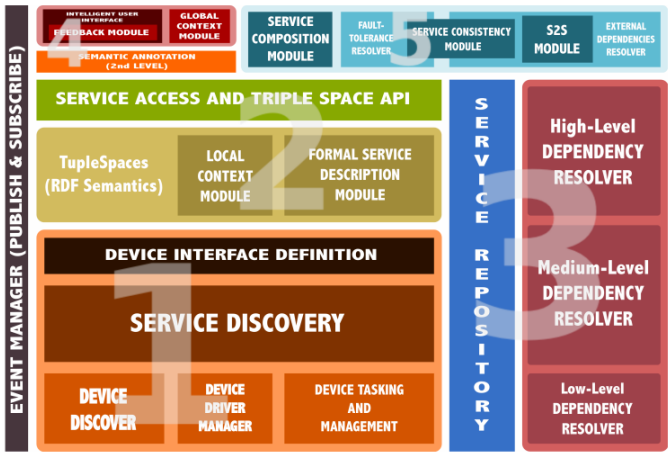


Figura 3. S^{3}OiA: Arquitetura e interoperabilidade.

Seus módulos são divididos em cinco grupos de módulos funcionais. Ao analisar a Figura 3 o módulo 1 seria o de "Descoberta de Serviços e Dispositivos", o módulo 2 "Exposição de Web Services e Triple Spaces", o módulo 3 "Repositório de Serviços e Resolução de Dependências", o módulo 4 "Interface de Interação" e o módulo 5 "Composição, Tolerância a Falhas e Dependências Distantes".

E. Ubiware

Ubiware é uma plataforma de gerenciamento baseada em três requisitos: automação, integração e interoperabilidade. A solução proposta incorpora princípios de sistemas multiagentes, entidades computacionais com comportamento autônomo que facilitam o desenvolvimento de sistemas complexos. No que tange à interoperabilidade, a proposta se baseia em suportar o máximo número de protocolos e dispositivos possível, por não acreditar no estabelecimento de novos padrões.

Ubiware visa permitir Global Enterprise Resource Integration (GERI). Ao contrário das soluções de EAI (Enterprise Application Integration), que estão preocupados com a integração

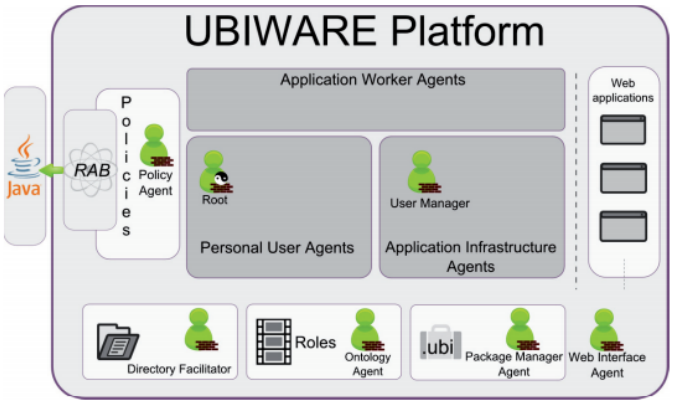


Figura 4. Plataforma Ubiware.

de apenas sistemas de computador, Ubiware visa integrar todos os tipos de recursos encontrados em uma empresa: digital, físicos e humanos. Com o Ubiware os dispositivos são capazes de se comunicar globalmente, ou seja, em toda a empresa. Para isso, como visto na Figura 4 e como mencionado no início, o fato de cada objeto ter um agente vinculado a ele implica que tal agente tem pleno conhecimento acerca de seu estado. Dessa forma, tal conhecimento pode ser trocado com outros agentes e utilizado para melhorar a execução de outros objetos vinculados à plataforma.

F. WoT Enabler

A WoT Enabler, é uma plataforma baseada em REST para a integração de sensores e compartilhamento. Implementada utilizando a linguagem de programação Ruby, seu servidor se comunica com uma base de dados conectada à arquitetura, responsável pelo armazenamento dos dados, os quais são enviados diretamente pelos dispositivos conectados (para dispositivos dotados de conexão direta à Internet) ou por *gateways* (para dispositivos que não possuem tal capacidade).

De modo similar às plataformas EcoDiF, Xively e RestThing, a arquitetura também usa EEML para a estruturação dos dados provenientes dos sensores, diferindo apenas na hierarquia de dados utilizada: um *system* representa uma coleção de dados referentes a um determinado ambiente; um sensor representa um sensor individual no contexto de um *system* e; um data é um par que se refere ao valor de uma medida aferida por um sensor em um dado instante de tempo.

G. Xively

Proposto pela LogMeIn a Xively[3] é uma plataforma que utiliza serviços de nuvem para gerenciar dados providos por dispositivos [11]. Esta plataforma fornece uma API que permite o envio de dados a partir do sensores que permite a visualização histórica dos dados e provê um mecanismo capaz de disparar eventos com base nos dados gerados pelos

[3]Xively - https://xively.com/

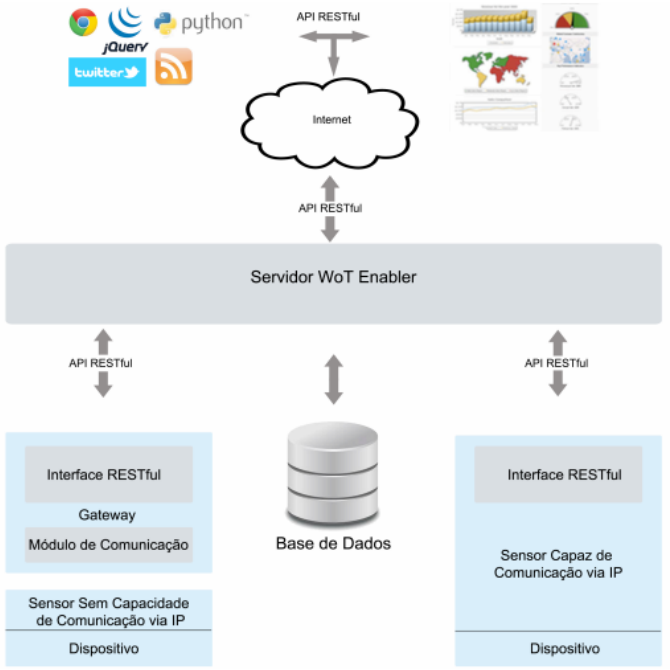


Figura 5. Arquitetura da plataforma WoT Enabler.

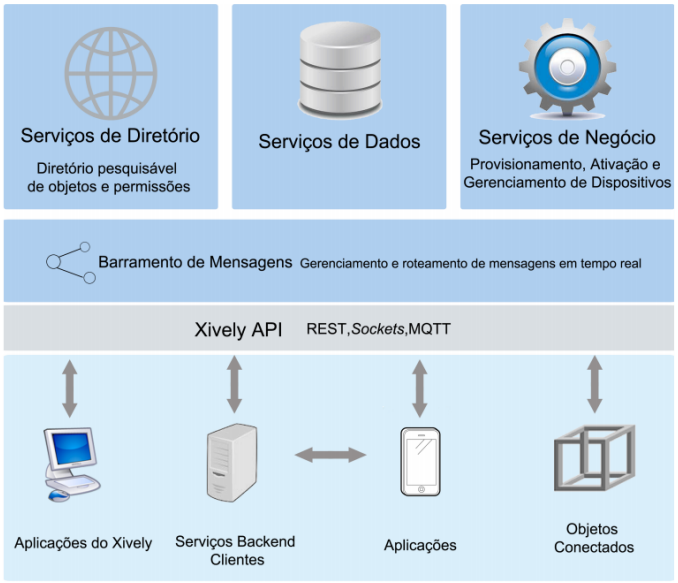
sensores. Tem como base princípios REST e padrões da Web. A plataforma fornece interfaces bem definidas e padronizadas.

A organização dos dados é baseada em *feeds, datapoints e datastreams*. Um *feed* representa os dados de um ambiente a ser monitorado em conjunto com seus *datastreams* que é responsável por representar os dados enviados por um determinado sensor neste ambiente. *Datapoints* representam um único valor de um *datastream* em um determinado instante de tempo.

Xively é uma solução comercial e de código fechado, desta forma não há detalhes quanto à arquitetura desta plataforma além do exposto na Figura V-G. Sabe-se que há pelo menos três formas de se recuperar dados disponibilizados pelos dispositivos conectados à Xively: (i) sensores enviam dados para a plataforma nos formatos JSON, XML (EEML) ou CSV usando API REST; (ii) através de sockets, e; (iii) através do protocolo MQTT [11].

H. Carriots

Carriots[4] assim como Xively também é uma plataforma para IoT que utiliza serviços de nuvem para gerenciar dados providos por dispositivos, porém, além de conectar dispositivos a dispositivos, também conecta sistemas a dispositivos, desta forma é possível conectar um sistema à plataforma e modela-lo como um dispositivo. Outra semelhança com a plataforma anterior é o uso da API RESTful. Carriots tem por objetivo coletar e armazenar qualquer dado originado dos mais diversos dispositivos e utilizá-lo em seu motor de aplicações e



disponibilizá-lo a seus usuários não importando o volume de dispositivos conectados [11].

Entidades da plataforma possuem uma hierarquia bem definida, isso significa que todos os dispositivos conectados à plataforma são associados a serviços e todos serviços pertencem a um projeto. Eventos como recebimento ou persistência de dados podem ser monitorados por *listeners*, entidades responsáveis por executarem análise e processamento em caso de alguma condição pré-configurada ter sido atendida.

A arquitetura lógica da Carriots é formada de módulos. São eles:

- API REST
- Big Data
- Gerenciamento de Projetos e Dispositivos
- Regras de Negócio e Processamento de Eventos
- Segurança
- *Logs e Debug*
- Painel de Controle
- Módulo de Comunicação Externa

Os dados trocados entre dispositivos e sistemas conectados podem ser representados de duas formas distintas: (i) sensores enviam dados nos formatos JSON ou XML usando a API REST, ou (ii) através do protocolo de mensagens MQTT. Um módulo de Big Data provê às aplicações a flexibilidade de gerenciar os dados dos dispositivos de modo que a massa de dados seja armazenada em uma arquitetura de Big Data em formato *schemaless*. O módulo de Gerenciamento de Projetos e Dispositivos contém os projetos criados pelos usuários e é responsável por atualizações do software embarcado nos dispositivos e por suas configurações. O armazenamento e a execução de eventos é feita através de *scripts* que utilizam o linguagem de programação Groovy[5], esta responsabilidade cabe ao módulo de Regras de Negócio e Processamento de

[4]Carriots - https://www.carriots.com/

[5]Groovy - https://groovy-lang.org

Eventos. A segurança é tradade de quatro formas: (i) uso de chaves pré-compartilhadas para a definição de privilégios de acesso; (ii) através da utilização do protocolo HTTPS para a encriptação do dados enviados pela rede; (iii) pela utilização de Hash HMAC e senha para autenticação e verificação de conteúdo; e (iv) por criptografia customizada a partir de *scripts* criados pelo usuário. O módulo *Logs Debug* fornece um console para depuração de erros e registros de mensagens. Painel de Controle permite o gerenciamento pelo usuário de todos os outros módulos e recursos da plataforma. O módulo de Comunicação Externa permite o envio de *e-mails*, SMS e a interação com outros sistemas – DropBox, Twitter, etc – [11].

I. LinkSmart (Hydra)

Anteriormente chamado de Hydra a plataforma LinkSmart[6] é um *middleware* de gerenciamento de dispositivos para IoT baseado na Arquitetura Orientada a Serviços oferencendo suporte ao desenvolvimento de aplicações formadas por dispositivos físicos heterogêneos que operam com recurso limitado em termos de poder computacional, energia e memória [11]. Oferece interface de serviços Web para controle de qualquer tipo de dispositivo físico e permite que desenvolvedores incorporem dispositivos físicos heterogêneos em suas aplicações. Sua arquitetura é composta de três camadas principais, são elas:

- Camada de Rede: Responsável pela comunicação com os dispositivos.
- Camada de Serviço: Responsável pelo gerenciamento de eventos, dispositivos, escalonamento de recursos, etc;
- Camada Semântica.

A descrição semântica dos dispositivos através do uso de ontologias de dispositivos é uma das características mais importante dessa plataforma. É responsável por representar as informações sobre os dispositivos baseada na ontologia de dispositivos FIPA (*Foundation for Intelligent Physical Agents*)[7] e permite a parametrização semântica para incluir informações dos dispositivos, como seus recursos de segurança [11].

LinkSmart contém um módulo chamado *Application Ontology Manager*, que permite o uso da ontologia de dispositivos para buscar propriedades, funções e atualizações de dispositivos. A camada semântica é composta de 5 módulos: (i) módulo de inferências (*reasoner*) responsável por inferir sobre o status dos dispositivos e indicar qual tipo de dispositivo entrou na rede; (ii) um módulo de consulta (*query*) utilizado para recuperar informações sobre os dispositivos e suas capacidades; (iii) um módulo de atualização, que permite inclusão, remoção e mudanças na ontologia; (iv) um módulo de versionamento, que como o nome já sugere, permite gerenciar diferentes versões da ontologia, incluindo diferentes versões dos dispositivos e serviços, e; (v) um módulo de interpretação e anotação, responsável por automaticamente atualizar a ontologia com novos tipos de dispositivos e por realizar análise e anotação dos dispositivos existentes e descrições que são incluídas na ontologia [11].

A plataforma também é capaz de distinguir entre dispositivos com recursos restritos – *non LinkSmart-enabled device* – que não são capazes de hospedá-la e dispositivos poderosos – *LinkSmart-enabled devices* – que são capazes de hospedá-la.

J. OpenIoT

A plataforma OpenIoT[8] tem por objetivo ser uma camada de suporte para aplicações em IoT usando um modelo baseado em infraestrutura de nuvem [11]. Os recursos IoT podem ser acessados por serviços sob demanda que seguem o modelo de computação em nuvem, sendo assim, torna-se possível que o serviço seja disponibilizado na nuvem de tal forma que o usuário pode configurar, implementar e usar o IoT. O projeto foca na convergência entre IoT e computação em nuvem, visando assim fornecer uma "nuvem das coisas-- *cloud of things* [11]. Desta forma a plataforma proposta permite a customização de ambientes de nuvem auto-organizáveis para aplicações IoT, tal que, provedores de serviços contendo infraestruturas de nuvem forneçam estes serviços IoT a usuários quaisquer.

A OpenIoT conecta os sensores com o ambiente de nuvem para que seja possível utilizar os recursos da nuvem para processar e gerenciar os dados, esse tipo de abordagem se torna útil para processamento massivo de sinais que não são suportados em infraestruturas de IoT devido aos recursos físicos limitado dos dispositivos. A fim de permitir a interoperabilidade entre os vários objetos, a solução proposto baseia-se na utilização da ontologia SSN (*Semantic Sensor Network*)[9] que serve como base para especificação de solicitações dos serviços combinando sensores, fluxo de dados e suas propriedades.

A arquitetura do OpenIoT é composto por três planos lógicos distintos: (i) Utilidade/Aplicação; (ii) virtualizado; e (iii) Físico. Tais planos, por sua vez, são compostos por sete módulos principais [11].

O plano Utilidades/Aplicações é composto por três módulos. São eles:

- Definição de Solicitação: Permite definir solicitações de serviço para a plataforma em tempo de execução, essas solicitações serão enviadas ao Escalonador.
- Apresentação de Solicitação: Via interface Web permite que o usuário seleciona *mashups* de um repositório.
- Configuração e Monitoramento: Permite o gerenciamento e a configuração de dispositivos conectos e dos serviços em execução na plataforma.

O plano Virtualizado incorpora módulos complementares ao plano utilidade/aplicações, são eles:

- Escalonador: Responsável por processar todas as requisições para serviços originadas no módulo Definição de Solicitação e garante acesso aos recursos necessários.
- Nuvem de Armazenamento de Dados: Armazena o fluxo de dados originado dos dispositivos conectados bem como metadados necessários para o funcionamento da plataforma. Para realizar esta tarefa, o módulo incorpora uma versão adaptada do *middleware Linked Stream*.

[6]LinkSmart - https://linksmart.eu

[7]FIPA - http://www.fipa.org/specs/fipa00091/index.html

[8]OpenIoT - http://www.openiot.eu/

[9]SSN - http://www.w3.org/2005/Incubator/ssn/ssnx/ssn

- Prestação de Serviços & Gerenciador de Recursos: Combina fluxo de dados para serem entregues aos serviços solicitantes e realiza a contabilidade e o faturamento dos recursos da plataforma.

O plano Físico é composto apenas pelo módulo Sensor *Middleware* que é responsável pela coleta, filtragem, combinação e anotação da semântica dos fluxo de dados providos por dispositivos físicos e virtuais, para realização desta tarefa, o módulo incorpora uma versão adaptada do *middleware* GSN (*Global Sensor Network*)[10], que fornece uma API RESTful para a interoperação com os dispositivos, bem como alguns recursos de segurança – autenticação e permissão de acesso através da definição de papeis – [11].

VI. Soluções *vs.* Requisitos

Considerando os oito requisitos levantados na Seção III, serão correlacionados com as características das onze soluções encontradas. Os requisitos são:

(i) Interoperabilidade;
(ii) Descoberta e configuração de dispositivos;
(iii) Ciência de contexto;
(iv) Escalabilidade;
(v) Gerenciamento de grandes volumes de dados;
(vi) Segurança;
(vii) Adaptação dinâmica; e
(viii) Interfaces de alto nível.

Na Tabela VI e na Tabela VI, o símbolo *check verde* denota que o requisito é completamente atendido, o símbolo *bola amarela* indica que o requisito é parcialmente atendido, e o símbolo *xis vermelho* indica que o requisito não é atendido.

Plataformas de *middleware*	Interoperabilidade	Descoberta e Gerenciamento de Dispositivos	Interfaces de Alto Nível	Ciência de Contexto
EcoDiF	✓	○	✓	○
Xively	✓	✓	✓	○
Carriots	✓	✓	✓	○
LinkSmart	✓	✓	×	✓
OpenIoT	✓	×	✓	×
RestThing	✓	×	✓	○
WoT Enabler	✓	×	✓	○
S^{3}OIA	✓	✓	×	✓
Ubiware	✓	×	×	✓
WSO2	✓	✓	✓	×
INRIA ARLES	✓	×	×	✓

Diante do exposto nas Tabelas VI e VI, constata-se que nenhuma solução de gerenciamento de IoT foi capaz de atender a todos os requisitos levantados. Tais soluções tratam apenas subconjuntos dos requisitos, abordando-os de formas diversas. A interoperabilidade é um exemplo. Apesar de ser atendida por todos, EcoDiF e OpenIoT consideram que o uso de protocolos e tecnologias Web amplamente utilizados são suficientes para mitigar os problemas da heterogeneidade entre os dispositivos, enquanto plataformas como Carriots, Xively e WSO2

[10]GSN - https://github.com/LSIR/gsn/wiki

Plataformas de *middleware*	Escalabilidade	Gerenciamento de Grandes Volumes de Dados	Segurança	Adaptação Dinâmica
EcoDiF	✓	✓	✓	×
Xively	✓	✓	✓	×
Carriots	✓	✓	✓	×
LinkSmart	×	✓	✓	×
OpenIoT	✓	✓	✓	×
RestThing	×	×	×	×
WoT Enabler	×	×	×	×
S^{3}OIA	×	×	×	✓
Ubiware	×	×	×	✓
WSO2	✓	✓	✓	✓
INRIA ARLES	×	✓	×	×

acreditam que o suporte a outros protocolos é importante. Há várias questões de pesquisa e desenvolvimento em aberto, as tecnologias e abordagens ainda divergem, além de não haver uma solução capaz de englobar todos os requisitos necessários para a concretização do paradigma.

VII. Considerações Finais

A ideia básica por trás do conceito de IoT é a presença generalizada de uma variedade de objetos (sensores, atuadores, telefones celulares, etc.) que, através de esquemas de endereçamento único e outros mecanismos de suporte baseados em padrões e protocolos ubíquos, são capazes de interagir uns com os outros e cooperar com seus vizinhos para alcançar objetivos comuns. Dessa forma, esse paradigma tem o potencial de contribuir em vários aspectos da vida contemporânea, propiciando uma vasta gama de aplicações que facilitem tarefas cotidianas. Domínios de aplicação como domótica, ambientes de vida assistida, monitoramento ambiental, automação industrial, transporte inteligente e gestão de negócios são apenas alguns dos possíveis cenários de aplicação em que esse novo paradigma irá desempenhar papeis primordiais em um futuro próximo.

O desafios para o paradigma de IoT vão desde soluções que permitam a interoperação e a integração dos diversos componentes que compõem os ambientes de IoT até a facilitação do desenvolvimento de aplicações para esses ambientes, passando por questões relativas à escalabilidade por conta do grande número de objetos envolvidos. Objetos que compõem a IoT possuem baixos recursos computacionais e de energia. Diversos protocolos de comunicação utilizados em IoT já fazem algum tipo de gerenciamento em relação à segurança das informações e qualidade da rede, porém indicam que ainda se faz necessário haver um gerenciamento mais específico devido a grande heterogeneidade de dispositivos e protocolos existentes. As soluções propostas precisam ter especial consideração por questões relativas à eficiência dos recursos. Nesse contexto, plataformas de gerenciamento de IoT devem satisfazer a um conjunto de requisitos a fim de satisfazer as demandas dos desafios apresentados. Neste artigo, foram encontradas onze soluções de gerenciamento de IoT existentes que lidam com esses desafios. Porém, estas soluções ainda possuem diversas deficiências, indicando que o estado da arte ainda não permite a plena concretização desse paradigma, indicando a emergência

de novas plataformas de gerenciamento que abordem os requisitos levantados por completo. Para os trabalhos futuros sobre este tema, estima-se que seja feito um levantamento de mais soluções de gerenciamento de IoT, permitindo compará-los sob o mesmo rigor científico, evitando possíveis imparcialidades.

REFERÊNCIAS

[1] C. Witchalls and J. Chambers, "The Internet of Things business index," The Economist, Report, Oct. 2013. [Online]. Available: analysis/1716

[2] C. Stamford, "Gartner Says 6.4 Billion Connected," Nov. 2015. [Online]. Available: http://www.gartner.com/newsroom/id/3165317

[3] V. Turner, D. Reinsel, J. F. Gantz, and S. Minton, "The Digital Universe of Opportunities: Rich Data and the Increasing Value of the Internet of Things," EMC and IDC, USA, White Paper 1672, Apr. 2014. [Online]. Available: http://www.emc.com/leadership/digital-universe/2014iview/executive-summary.htm

[4] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A Survey," *Comput. Netw.*, vol. 54, no. 15, pp. 2787–2805, Oct. 2010.

[5] A. Whitmore, A. Agarwal, and L. D. Xu, "The Internet of Things—A survey of topics and trends," *Inf Syst Front*, vol. 17, no. 2, pp. 261–274, Mar. 2014.

[6] I. Kotenko, I. Saenko, and S. Ageev, "Countermeasure Security Risks Management in the Internet of Things Based on Fuzzy Logic Inference," in *2015 IEEE Trustcom/BigDataSE/ISPA*, vol. 1, Aug. 2015, pp. 654–659.

[7] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, "Internet of Things for Smart Cities," *IEEE Internet of Things Journal*, vol. 1, no. 1, pp. 22–32, Feb. 2014.

[8] A. Guerrieri, V. Loscri, A. Rovella, and G. Fortino, *Management of Cyber Physical Objects in the Future Internet of Things: Methods, Architectures and Applications*. Springer, Jan. 2016.

[9] M. A. Chaqfeh and N. Mohamed, "Challenges in middleware solutions for the internet of things," in *2012 International Conference on Collaboration Technologies and Systems (CTS)*, May 2012, pp. 21–26.

[10] F. C. Delicato, P. F. Pires, and T. Batista, *Middleware Solutions for the Internet of Things*, ser. SpringerBriefs in Computer Science. London: Springer Science & Business Media, Sep. 2013. [Online]. Available: http://link.springer.com/10.1007/978-1-4471-5481-5

[11] P. F. Pires, F. C. Delicato, T. Batista, T. Barros, E. Cavalcante, and M. Pitanga, *Plataformas para a Internet das Coisas*, M. Martinello, M. R. N. Robeiro, and A. A. de Aragão Rocha, Eds. Porto Alegre, RS, Brasil: SBC, 2015. [Online]. Available: http://sbrc2015.ufes.br/?page_id=71