

Um estudo de caso para identificação de elementos e funcionalidades para uma arquitetura de referência para a Internet das Coisas

Kétly Gonçalves Machado
Universidade de São Paulo
Instituto de Matemática e Estatística
São Paulo, SP - Brasil
ketly.machado@gmail.com

Marco Aurélio Spohn
Universidade Federal da Fronteira Sul
Curso de Ciência da Computação
Chapecó, SC - Brasil
marco.spohn@uffs.edu.br

ABSTRACT

A principal proposta da Internet das Coisas é a criação de uma rede de dispositivos heterogêneos interconectados, de forma semelhante à Internet. Seu propósito é fazer com que dispositivos/coisas quaisquer como automóveis, eletrodomésticos, entre outros, tornem-se inteligentes, desempenhando tarefas e compartilhando informações. A Internet das Coisas tem ganho popularidade e demonstra grande importância para a inovação e a evolução de diversas áreas. Por conta disso, é evidente o crescimento do número de soluções desenvolvidas para a Internet das Coisas e, dessa forma, é fundamental que se estabeleça uma arquitetura de referência para melhorar o processo de desenvolvimento e a implementação dessas soluções. Existem diferentes iniciativas que buscam estabelecer uma arquitetura de referência para a Internet das Coisas; entretanto, até o momento não se chegou a um consenso. Visando contribuir com o estado da arte dessas iniciativas, o presente trabalho tem como objetivo identificar, através de um estudo de caso realizado sobre um conjunto específico de plataformas de Internet das Coisas, elementos e funcionalidades para uma arquitetura de referência, analisando as consequências da sua adoção.

CCS Concepts

•Software and its engineering → Requirements analysis; Software design engineering;

Keywords

Internet das Coisas; arquiteturas de internet das coisas; plataformas de internet das coisas

1. INTRODUÇÃO

O termo *Internet of Things* (IoT, Internet das Coisas) tem demonstrado crescente popularidade. Sua introdução é atribuída a Kevin Ashton, que o mencionou em uma apresentação realizada em 1999 para a empresa Procter & Gamble (P&G)¹. A principal pro-

¹Posteriormente, essa apresentação foi referenciada por Kevin em

posta da Internet das Coisas é a criação de uma rede de objetos interconectados, de forma semelhante à Internet. Os objetos, também chamados de dispositivos, podem ser de diversos tipos, são alguns exemplos: automóveis, eletrodomésticos, termostatos e dispositivos vestíveis. A Internet das Coisas, através de diversas tecnologias, dá condições para que os dispositivos desempenhem tarefas e sejam capazes de compartilhar informações, transformando-os em dispositivos inteligentes [1].

Diferentemente da Internet, na Internet das Coisas há a presença acentuada da heterogeneidade, de modo que os dispositivos podem ter diferentes capacidades de processamento, memória, armazenamento, entre outras características, o que exige a utilização de tecnologias alternativas para a implementação de soluções, como a computação em nuvem e a *fog computing*.

Existem diversas áreas e diferentes cenários nos quais a Internet das Coisas pode ser aplicada, um exemplo de aplicação é a área industrial, onde ela pode ser utilizada para o monitoramento do estoque de uma indústria. Nesse cenário, o sistema de controle de estoque verifica continuamente as entradas, as saídas e o quantitativo de material armazenado, sendo capaz de orientar ou até mesmo realizar a compra de produtos, além de gerenciar a sua distribuição. A Internet das Coisas também pode ser aplicada no sistema de saúde, na agricultura, nas cidades inteligentes, no transporte, entre outras áreas [1, 9].

A Internet das Coisas demonstra grande importância para a inovação e a evolução de diversas áreas, ela funciona como um meio para o alcance de fins significativos. Com a utilização da Internet das Coisas, será possível realizar um aproveitamento expressivo do volume gigantesco de dados produzidos pela humanidade, ou seja, além da coleta, serão realizados também o tratamento, a análise e a computação dos dados para extração de valor. Depois desse processo, os próprios dispositivos serão capazes de tomar decisões a partir dos dados obtidos. Além disso, a Internet das Coisas promete facilitar diversas atividades cotidianas, de tal forma que dispositivos simples, quando conectados, permitirão automatizar tarefas como o travamento de uma fechadura e o chaveamento de um interruptor de luz [1, 17].

Considerando que a ideia básica da Internet das Coisas é interconectar dispositivos através de uma rede, é possível dizer que o objetivo de uma plataforma de Internet das Coisas é preencher a lacuna existente entre os dispositivos e a rede. Ou seja, a função primária de uma plataforma é realizar a ligação entre os dados emitidos pelos dispositivos com a rede de dados propriamente dita. As plataformas diferem entre si nas suas características e nos seus métodos

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

um artigo publicado no *RFID Journal* (Disponível em: <http://www.rfidjournal.com/articles/view?4986>).

de implementação, possuindo diferentes arquiteturas e trabalhando com diferentes tecnologias no desenvolvimento de soluções para a Internet das Coisas.

Uma arquitetura de referência pode ser entendida como um conjunto de documentos que agregam as melhores práticas para o desenvolvimento de uma solução. Nesses documentos estão descritos os padrões e as integrações que devem ser praticados para se alcançar uma solução otimizada. Dessa forma, pode-se dizer que a arquitetura de referência sugere o método ideal para se implementar uma tecnologia [12]. Estabelecer uma arquitetura de referência para determinada tecnologia é fundamental para melhorar o processo de desenvolvimento e a implementação de soluções relacionadas. A partir da utilização de uma arquitetura de referência é possível melhorar a comunicação entre os envolvidos no desenvolvimento de um projeto, fazendo com que todos sigam o mesmo referencial para a estruturação dos elementos e para a integração das etapas.

Da mesma forma que ocorreu com a Internet, existem projetos em andamento para a definição de uma arquitetura de referência para a Internet das Coisas, levando em conta as suas particularidades. Apesar do número crescente de iniciativas que buscam estabelecer essa arquitetura, ainda não se alcançou consenso [1, 15, 21]. A identificação de uma arquitetura de referência é um processo longo e envolve diversos ajustes para a definição das tecnologias e dos requisitos necessários para a implementação de soluções para determinada área. Apesar disso, é possível elencar alguns aspectos com os quais a arquitetura de referência para a Internet das Coisas deve ser capaz de lidar: conectividade e comunicação; gerenciamento de dispositivos; coleta e análise de dados para extrair informações; escalabilidade; segurança e privacidade [21].

Considerando a importância e os benefícios proporcionados pela existência de uma arquitetura de referência, esse estudo de caso visa identificar elementos e funcionalidades para uma possível arquitetura de referência para a Internet das Coisas. Um conjunto de plataformas foi utilizado como objeto de estudo e, através do detalhamento das arquiteturas dessas plataformas, foram apontados elementos e funcionalidades para implementação em uma possível arquitetura de referência. O intuito desse estudo é contribuir com o estado da arte das iniciativas que buscam estabelecer uma arquitetura de referência para a Internet das Coisas.

O restante desse artigo está organizado da seguinte forma: a Seção 2 apresenta os principais fundamentos sobre Internet das Coisas; a Seção 3 trata sobre as principais iniciativas relativas a propostas de arquiteturas para Internet das Coisas; a Seção 4 apresenta uma revisão sucinta acerca das principais plataformas de desenvolvimento para Internet das Coisas, bem como discrimina as plataformas selecionadas para o estudo de caso; a Seção 5 apresenta uma análise detalhada sobre as principais características e aspectos arquiteturais das plataformas alvo do estudo de caso, bem como identifica os elementos e funcionalidades recomendáveis a uma possível arquitetura de referência para Internet das Coisas; finalizando, a Seção 6 apresenta as principais conclusões desse trabalho e recomendações de trabalhos futuros.

2. Internet das Coisas

O aumento no número de dispositivos conectados na Internet das Coisas é um dos fatores destacados na literatura [1, 13]. No gráfico da Figura 1, é apresentada uma previsão do número de dispositivos que estarão conectados até o ano de 2025. Observa-se um prognóstico de um aumento de aproximadamente 60 bilhões de dispositivos em relação ao ano de 2015; portanto, pode-se dizer que a Internet das Coisas terá um crescimento considerável em um futuro próximo, evidenciando a importância da definição de uma arquitetura

de referência.

Entender os elementos que integram a Internet das Coisas é fundamental para compreender o seu funcionamento. [1] apresentam os seis principais elementos necessários para a construção de uma solução para a Internet das Coisas: identificação, sensores/atuadores, comunicação, computação, serviços e semântica.

A identificação consiste no fornecimento de uma identidade clara e exclusiva para cada dispositivo dentro da rede, através de tecnologias específicas, como, por exemplo, o Código Eletrônico de Produto (*Electronic Product Code* – EPC) e o endereçamento IP. Identificar os dispositivos de forma individual é importante para entender cada um deles como um nó exclusivo dentro da rede e, dessa forma, conseguir atuar sobre e coletar dados especificamente de um determinado dispositivo.

Os sensores e atuadores são fundamentais para o desenvolvimento de soluções para a Internet das Coisas. Os sensores são responsáveis pela coleta de dados do contexto onde encontra-se determinado dispositivo e pelo envio desses dados para os servidores de armazenamento (*data warehouse*, base de dados ou nuvem), onde serão tratados. Os atuadores recebem informações ou comandos e, a partir deles, agem provocando alterações de estado no contexto onde encontra-se determinado dispositivo.

As tecnologias de comunicação são utilizadas para promover a conexão entre os diferentes dispositivos. A comunicação é um elemento crítico, pois os nós da Internet das Coisas precisam, geralmente, consumir uma quantidade mínima de energia e lidar com as perdas e os ruídos associados aos enlaces. Exemplos de tecnologias de comunicação utilizadas para a Internet das Coisas são WiFi, Bluetooth, LoRa, SigFox e NB-IoT.

O elemento computação está representado pelas unidades de processamento (microcontroladores, microprocessadores, entre outros) e pelas aplicações de *software* responsáveis pela habilidade computacional dos dispositivos da Internet das Coisas. Estão incluídas no elemento computação: plataformas de *hardware* (e.g., Arduino e Raspberry Pi), plataformas de *software* como os sistemas operacionais de tempo real (*Real Time Operating System* – RTOS) e também as plataformas de computação em nuvem.

Os serviços fornecidos pela Internet das Coisas podem ser categorizados da seguinte forma: Serviços de Identificação, responsáveis por identificar os dispositivos do mundo real trazidos para o mundo digital; Serviços de Agregação de Informação, que coletam e sintetizam os dados obtidos através dos sensores dos dispositivos; Serviços de Colaboração e Inteligência que agem sobre os Serviços de Agregação de Informação e utilizam os dados para tomar decisões e reagir de acordo com determinados cenários; e Serviços Ubíquos, que visam fornecer Serviços de Colaboração e Inteligência sempre que forem necessários, em qualquer lugar e para qualquer um que precise deles. O objetivo de toda aplicação de Internet das Coisas é atingir o nível de Serviços Ubíquos.

A semântica na Internet das Coisas refere-se à habilidade de extrair conhecimento dos dados de forma inteligente. Esse elemento abrange a descoberta de conhecimento, incluindo o reconhecimento e a análise dos dados e também a associação de demandas aos recursos apropriados. Para isso, utilizam-se técnicas da Web Semântica, como o *Resource Description Framework* (RDF) e a *Ontology Web Language* (OWL).

2.1 Desafios

Implementar soluções para a Internet das Coisas torna-se uma

²Disponível em: <https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/>. Acesso em: 01 abril 2019.

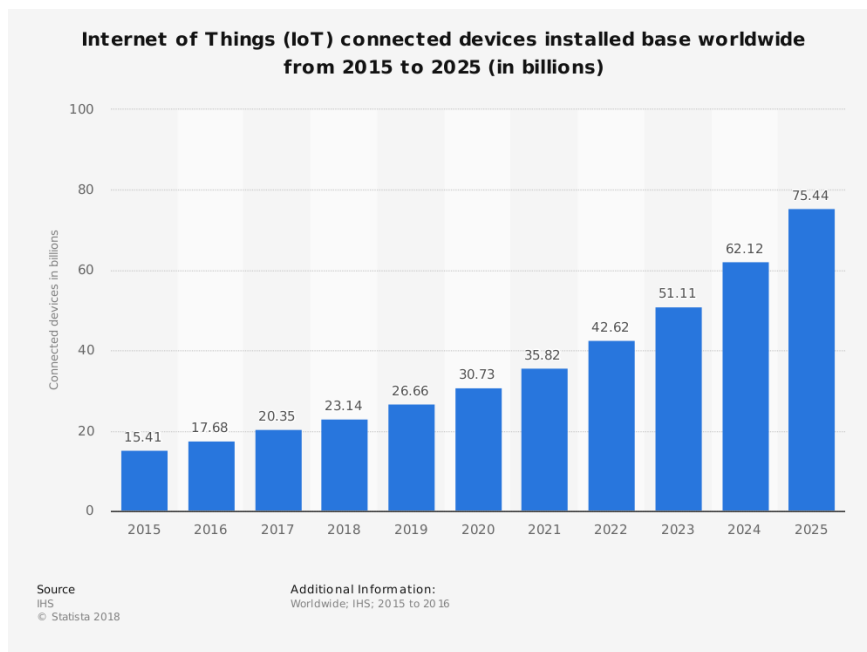


Figure 1: Dispositivos conectados à Internet das Coisas em todo o mundo de 2015 até 2025 (em bilhões).

Fonte: Statista, 2018².

atividade complexa por consequência dos diferentes desafios envolvidos no processo. Alguns dos principais desafios são a disponibilidade, a confiabilidade, a mobilidade, o desempenho, o gerenciamento, a escalabilidade, a interoperabilidade, a segurança e a privacidade. O tratamento desses desafios permite que os provedores de serviços e os programadores de aplicativos implementem suas soluções com eficiência.

Os serviços oferecidos pela Internet das Coisas devem estar disponíveis a qualquer tempo e em qualquer lugar. Para isso, é necessário garantir a disponibilidade em nível de *software*, assegurando que aplicações de Internet das Coisas forneçam serviços de forma simultânea a todos os usuários, independentemente da sua localização, e também em nível de *hardware*, assegurando a compatibilidade, em tempo integral, do *hardware* com as funcionalidades e protocolos da Internet das Coisas. Uma solução para garantir alta disponibilidade é implementar a redundância de itens críticos.

A confiabilidade refere-se ao correto funcionamento do sistema, com base nas suas especificações, mesmo em situações adversas e inesperadas. Com a confiabilidade é possível aumentar a taxa de sucesso da entrega dos serviços de Internet das Coisas e, por isso, ela deve ser implementada em *software* e *hardware*, por todas as camadas de uma arquitetura.

A mobilidade é outro desafio para a implementação de soluções para a Internet das Coisas, visto que espera-se que a maioria dos serviços seja entregue a usuários móveis. Além disso, é preciso considerar o gigantesco número de dispositivos presentes na Internet das Coisas, o que requer mecanismos eficientes para o gerenciamento da mobilidade. Preservar a conexão entre os serviços e os usuários que desejam acessá-los, enquanto estão em movimento, é uma importante premissa da Internet das Coisas. Uma alternativa para promover a continuidade dos serviços, é a utilização de esquemas de mobilidade que envolvam métodos capazes de permitir o acesso aos dados da Internet das Coisas mesmo em caso de indisponibilidade temporária de recursos.

A Internet das Coisas precisa desenvolver e melhorar os seus

serviços continuamente para atender aos requisitos dos usuários; entretanto, avaliar o desempenho dos serviços de Internet das Coisas é um grande desafio, já que depende do desempenho de muitos componentes e de suas tecnologias. Muitas métricas podem ser usadas para avaliar o desempenho da Internet das Coisas, incluindo velocidade de processamento, velocidade de comunicação e custo.

Interconectar bilhões ou trilhões de dispositivos inteligentes representa um grande desafio para os provedores de serviços de Internet das Coisas, principalmente quanto aos problemas relacionados ao gerenciamento de falhas, de configuração, de contabilidade, de desempenho e de segurança. O gerenciamento de dispositivos e aplicações de Internet das Coisas pode ser um fator crucial para o aumento de soluções nessa área, por isso é importante que seja realizado o aprimoramento dos protocolos e mecanismos de gerenciamento, além do desenvolvimento de novas abordagens.

A escalabilidade na Internet das Coisas refere-se à habilidade de se adicionar novos dispositivos, serviços e funções sem afetar negativamente a qualidade dos serviços existentes. Adicionar novas funcionalidades e suportar novos dispositivos não é uma tarefa fácil, especialmente porque é necessário adaptar-se às diferentes plataformas de *hardware* e aos diferentes protocolos de comunicação. Por esse motivo, as aplicações de Internet das Coisas devem ser projetadas desde o início para integrar serviços e operações extensíveis.

A interoperabilidade é outro desafio para a Internet das Coisas, considerando a necessidade de se lidar com um grande número de dispositivos heterogêneos que pertencem a diferentes plataformas. A interoperabilidade deve ser considerada pelos desenvolvedores de aplicações e também pelos fabricantes dos dispositivos, visando garantir a entrega de serviços para todos os clientes, independentemente das especificações da plataforma de *hardware* que utilizem. A interoperabilidade é um critério significativo a ser considerado no projeto e construção de aplicações para a Internet das Coisas, para que seja possível atender aos requisitos dos clientes.

A segurança e a privacidade representam um desafio significa-

tivo para a implementação de soluções de Internet das Coisas. Em redes heterogêneas, como é o caso da Internet das Coisas, não é simples garantir a segurança e a privacidade dos usuários e, o crescente número de dispositivos inteligentes conectados, que possuem dados sensíveis, exige que sejam estabelecidas propostas de gerenciamento que permitam um controle de acesso transparente e objetivo.

2.2 Plataformas

[24] buscam estabelecer o estado da arte do desenvolvimento de plataformas de Internet das Coisas e, para isso, apresentam um panorama geral da estrutura e das funcionalidades de um conjunto de plataformas. Uma plataforma de Internet das Coisas é definida como um *software* que permite a conectividade de dispositivos e a aquisição, o processamento, a transformação, a organização e o armazenamento dos dados produzidos por esses dispositivos. Pontos centrais da Internet das Coisas, como os desafios apresentados anteriormente, são altamente relevantes para as plataformas de Internet das Coisas. Inclusive, a arquitetura de uma plataforma deve levar em conta todos esses desafios e a sua abordagem de desenvolvimento deve ter como objetivo atenuar ou extinguir quaisquer problemas decorrentes desses fatores.

As plataformas de Internet das Coisas são, geralmente, baseadas na computação em nuvem, entregando um serviço do tipo plataforma como serviço (*Platform-as-a-Service* – PaaS). A conectividade ubíqua e confiável é uma característica chave para as plataformas, sendo realizada sobre qualquer canal de comunicação e priorizando a qualidade da conexão. Os recursos típicos das plataformas de Internet das Coisas são: conectividade como um serviço, monitoramento e manutenção de dispositivos, visualização de dados, análise de dados e lógica básica de aplicação, através de alertas e *triggers*.

As seguintes categorias de plataformas de Internet das Coisas foram identificadas por [24]:

- Plataformas de domínio específico, que facilitam cenários específicos de determinado domínio.
- Plataformas de tecnologia específica, que levam em conta apenas um conjunto específico de dispositivos. Geralmente, são plataformas fechadas, baseadas em dispositivos com tecnologia proprietária.
- Provedores de conectividade máquina a máquina (*Machine-to-Machine* – M2M), cujo recurso principal é a conectividade como um serviço e que têm como objetivo primário a aquisição e a análise de dados.
- *Middlewares* genéricos de larga escala, que fornecem uma gama completa de serviços de conectividade, mas também facilitam o desenvolvimento de aplicações, com base em dados coletados pelos dispositivos e transformados por ferramentas analíticas.
- Plataformas de serviços de apoio. Essas plataformas não oferecem conectividade M2M, então não são plataformas de Internet das Coisas; entretanto, oferecem funcionalidades que podem ser úteis para cenários de Internet das Coisas.

O desenvolvimento de plataformas de Internet das Coisas é impulsionado pela necessidade de se facilitar a conectividade M2M, que vêm crescendo a uma taxa sem precedentes. Com a ascensão das plataformas de Internet das Coisas, a interoperabilidade entre plataformas e a reutilização estão emergindo. Há situações em que plataformas de Internet das Coisas de domínio específico são feitas através da utilização de provedores de conectividade M2M. Da

mesma forma, os primeiros casos de colaboração entre plataformas aparecem, com soluções de interoperabilidade.

2.3 Análise de *Big Data*, Computação em Nuvem e *Fog Computing*

Conectar um grande número de dispositivos equipados com sensores à Internet gera o que se chama de *big data*, um grande volume de dados, estruturados e não estruturados [1]. Os dispositivos conectados precisam de mecanismos para armazenar, processar e recuperar os dados gerados. Porém, o *big data* é tão vasto que excede a capacidade de ambientes de *hardware* e de ferramentas de *software* convencionais de capturar, gerenciar e processar esses dados em um tempo assumido como aceitável.

A tecnologia de computação em nuvem é definida como um modelo de acesso sob demanda à uma rede de recursos computacionais configuráveis compartilhados, como redes, servidores, *data warehouses*, aplicações e serviços. Os serviços de computação em nuvem permitem que indivíduos e empresas usem remotamente componentes de *software* e de *hardware* de terceiros, de forma confiável e a baixo custo. A Internet das Coisas emprega um grande número de dispositivos embarcados, como sensores e atuadores, que geram *big data*, que requer a utilização de computações complexas para extração de conhecimento dos dados. Sendo assim, os recursos de computação e armazenamento em nuvem representam a melhor escolha para que a Internet das Coisas armazene e processe *big data*.

Uma importante vantagem do *big data* é a possibilidade de extração de conhecimento, através do qual uma empresa/grupo pode alcançar vantagem competitiva. Existem diferentes plataformas para análise de *big data*; entretanto, essas plataformas não são robustas o suficiente para lidar com as necessidades da Internet das Coisas, pois o amontoado de dados produzidos pela Internet das Coisas é grande demais para ser processado pelas ferramentas disponíveis.

A computação em nuvem oferece um novo mecanismo de gerenciamento para *big data* que permite o processamento de dados e a extração de conhecimento dos mesmos. Entretanto, empregar a computação em nuvem para a Internet das Coisas não é uma tarefa fácil, pois devem ser levados em consideração os seguintes desafios:

- A sincronização entre diferentes fornecedores de computação em nuvem, que pode afetar os serviços em tempo real, construídos sobre diversas plataformas de serviços em nuvem;
- A padronização da computação em nuvem, que necessita que as diversas plataformas sejam capazes de interoperar;
- O balanceamento entre os ambientes de serviço em nuvem comuns e as especificações da Internet das Coisas;
- A segurança dos serviços de Internet das Coisas baseados em nuvem, que deve lidar com as diferenças entre os mecanismos de segurança dos dispositivos e das plataformas;
- O gerenciamento da computação em nuvem e dos sistemas de Internet das Coisas, por possuírem diferentes recursos e componentes;
- A validação dos serviços de Internet das Coisas baseados em nuvem, necessária para garantir o fornecimento de serviços que atendam aos requisitos dos clientes.

A *fog computing* pode atuar como uma ponte entre os dispositivos inteligentes e os serviços de computação e armazenamento em nuvem. Através da *fog computing*, é possível estender os serviços

da computação em nuvem até os dispositivos na borda da rede. Em razão da sua proximidade com os usuários finais, em comparação com os *data centers* da nuvem, a *fog computing* tem potencial para oferecer serviços com melhor desempenho. Geralmente, há uma diferença significativa de escala entre a computação em nuvem e a *fog computing*, de forma que a computação em nuvem tem uma capacidade de computação, armazenamento e comunicação massiva em comparação com a *fog computing*. Na Figura 2, estão representados os papéis desempenhados pelos *data centers* da nuvem e pelos *fog nodes* na entrega de serviços de Internet das Coisas para os usuários finais.

A utilização da *fog computing* pode ser uma ótima escolha para os desenvolvedores de soluções para a Internet das Coisas, por conta das seguintes características:

- Localização, pois os recursos da *fog computing* estão localizados entre os dispositivos inteligentes e os *data centers* da nuvem, proporcionando um melhor desempenho;
- Distribuição, pois a *fog computing* baseia-se em “micro” *centers* com capacidades de armazenamento, processamento e comunicação limitadas em comparação com a nuvem e, por isso, é possível implantar muitos “micro” *centers* próximos aos usuários finais, por um custo reduzido em relação ao custo de implantação de *data centers* da nuvem;
- Escalabilidade, pois sua estrutura permite que sistemas de Internet das Coisas sejam mais escaláveis, de maneira que a medida em que o número de usuários finais aumenta, aumenta também o número de “micro” *centers* implantados, o que é inviável realizar com os *data centers* da nuvem por conta do custo elevado;
- Densidade de dispositivos, pois a *fog computing* auxilia no fornecimento de serviços resistentes e replicados;
- Suporte à mobilidade, pois seus recursos agem como uma “nuvem móvel”, dado que está próximo aos usuários finais;
- Tempo real, pois tem potencial para fornecer um melhor desempenho para os serviços interativos de tempo real;
- Padronização, pois a *fog computing* pode interoperar com diversos fornecedores de computação em nuvem;
- Análise agilizada, pois os recursos da *fog computing* podem realizar a agregação de dados para enviar dados parcialmente processados, ao invés de dados brutos, para *data centers* na nuvem, para que recebam processamento complementar.

Sendo assim, a *fog computing* tem potencial para aumentar o desempenho geral das aplicações de Internet das Coisas, pois tenta desempenhar parte dos serviços de alto nível oferecidos pela nuvem em recursos locais.

3. ARQUITETURAS DE Internet das Coisas

3.1 Industrial Internet Reference Architecture (IIRA)

A *Industrial Internet Reference Architecture* (IIRA) [6] é uma arquitetura aberta baseada em padrões para os sistemas de Internet das Coisas Industrial (*Industrial Internet of Things* – IIoT), desenvolvida pelo Consórcio de Internet Industrial (*Industrial Internet Consortium* – IIC).

³Disponível em: <https://erpinnews.com/wp-content/uploads/2018/01/edge-computing-diagram-1024x512-850x425.png>. Acesso em: 01 abril 2019.

3.1.1 Industrial Internet Architecture Framework (IIAF)

Em sistemas complexos, como os da Internet das Coisas Industrial, muitos são os *stakeholders* envolvidos. Esses *stakeholders* possuem diversas preocupações que abrangem o ciclo de vida completo do sistema e, por conta disso, é necessário um *framework* para identificar e classificar essas preocupações em categorias apropriadas, para que então sejam avaliadas e resolvidas sistematicamente. Para atender a essa necessidade, o IIC usou o padrão ISO/IEC/IEEE 42010:2011 para definir o *Industrial Internet Architecture Framework* (IIAF) [6]. O IIAF identifica convenções, princípios e práticas para uma descrição consistente de arquiteturas de Internet das Coisas Industrial. Esse *framework* arquitetural baseado em padrões facilita a avaliação e a sistemática e efetiva resolução das preocupações dos *stakeholders*. Além disso, é um recurso para orientar o desenvolvimento e a documentação da IIRA.

Um *framework* arquitetural contém as informações que identificam a estrutura fundamental da arquitetura e especifica as preocupações, *stakeholders*, pontos de vista, tipos de modelos, regras de correspondência e condições de aplicabilidade. O ponto principal do padrão ISO/IEC/IEEE para descrição de arquiteturas são os pontos de vista. Um ponto de vista compreende as convenções que delimitam a descrição e a análise de preocupações específicas do sistema. Um *stakeholder* pode ser um indivíduo, um time ou uma organização, que tem interesse em uma preocupação e, por consequência, interesse no ponto de vista e no sistema. Para auxiliar nas tarefas de descrever, analisar e resolver preocupações, podem ser definidos tipos de modelos para cada ponto de vista. A construção de pontos de vista e seus *stakeholders*, preocupações e tipos de modelos correspondentes pode ser considerada como o *frame* da arquitetura. As ideias para descrever, analisar e resolver o conjunto de preocupações específicas de cada um dos pontos de vista, podem ser expressas como a visão da arquitetura. Aplicar os tipos de modelos definidos em cada ponto de vista para descrever, analisar e resolver as preocupações, resulta na criação de modelos de arquitetura, que compõem a visão da arquitetura correspondente. As visões da arquitetura juntamente com seus modelos de arquitetura podem ser considerados como as representações da arquitetura.

O IIAF adota os conceitos e construções gerais do padrão ISO/IEC/IEEE para descrição de arquiteturas, especificamente, preocupação, *stakeholder* e ponto de vista como o seu *frame* da arquitetura e visões e modelos como a sua representação da arquitetura, na descrição e análise de importantes preocupações comuns da arquitetura para sistemas de Internet das Coisas Industrial. O IIAF representa a base da IIRA.

3.1.2 Industrial Internet Reference Architecture (IIRA)

A IIRA documenta o resultado da aplicação do IIAF aos sistemas de Internet das Coisas Industrial. Primeiramente, identifica e destaca as preocupações arquiteturais mais importantes, comumente encontradas nos sistemas de Internet das Coisas Industrial, e classifica-as em pontos de vista, juntamente com seus respectivos *stakeholders*. Na sequência, descreve, analisa e, onde apropriado, fornece orientações para resolver as preocupações nesses pontos de vista, resultando em algumas representações de arquitetura abstratas. O nível de abstração da IIRA exclui elementos arquiteturais cuja avaliação requer especificidades que estão disponíveis apenas em sistemas concretos.

Os pontos de vista da IIRA foram definidos através da análise de diversos casos de uso da Internet das Coisas Industrial desenvolvidos pelo IIC e por outras organizações, da identificação dos *stakeholders* relevantes dos sistemas de Internet das Coisas Industrial e da determinação do enquadramento adequado das preocupações. Os quatro pontos de vista definidos são:

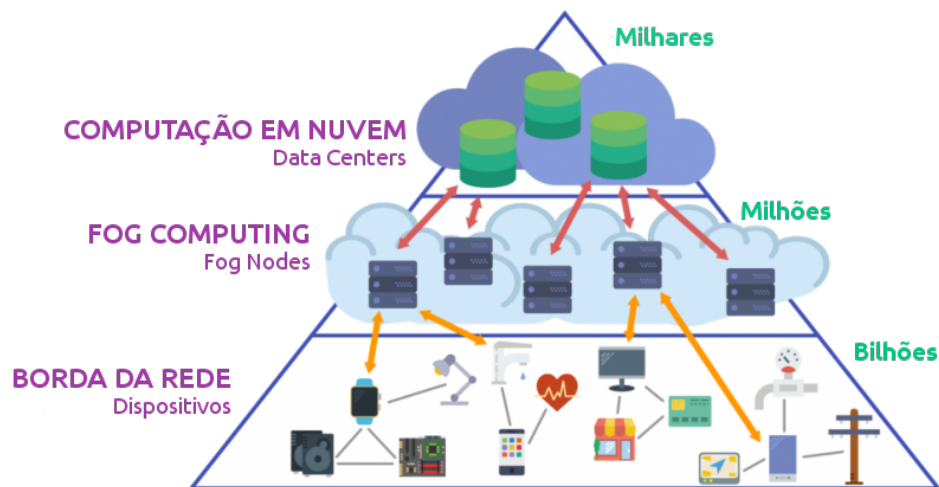


Figure 2: Papéis da computação em nuvem e da *fog computing* nos serviços de Internet das Coisas.

Fonte: Adaptado de ERPINNEWS, 2018³.

- Ponto de vista de negócios: Atende às preocupações de identificação dos *stakeholders* e de suas visões de negócio, seus valores e seus objetivos ao estabelecer um sistema de Internet das Coisas Industrial em seu contexto de negócios. Além disso, identifica como o sistema de Internet das Coisas Industrial alcança os objetivos estabelecidos, através do mapeamento das competências fundamentais do sistema;
- Ponto de vista de uso: Abrange as preocupações do uso esperado do sistema. É representado geralmente por sequências de atividades envolvendo usuários (humanos, sistemas, ou componentes de sistemas) que definem as funcionalidades pretendidas para alcançar as competências fundamentais do sistema;
- Ponto de vista funcional: Concentra-se nos componentes funcionais de um sistema de Internet das Coisas Industrial, suas estruturas e inter-relações, nas interfaces e suas inter-relações e na relação do sistema com elementos externos, para poder suportar o uso e as atividades do sistema como um todo;
- Ponto de vista de implementação: Preocupa-se com as tecnologias necessárias para a implementação dos componentes funcionais (ponto de vista funcional), seus esquemas de comunicação e seus procedimentos do ciclo de vida. Esses elementos são coordenados pelas atividades (ponto de vista de uso) e dão suporte às competências do sistema (ponto de vista de negócios).

Esses quatro pontos de vista formam a base para uma análise detalhada de conjuntos individuais de preocupações dos sistemas de Internet das Coisas Industrial.

3.2 Internet of Things – Architecture (IoT-A)

O projeto *Internet of Things – Architecture* (IoT-A) [16], desenvolvido pelo *Seventh Framework Programme* (FP7), propõe a criação de um modelo de referência arquitetural para a Internet das Coisas, bem como a definição de um conjunto de elementos fundamentais para estabelecer as bases de uma Internet das Coisas ubíqua.

A escolha central do projeto IoT-A foi basear seu trabalho no atual estado da arte, ao invés de iniciar do zero. Devido a essa escolha, características comuns foram utilizadas para formar a base

do modelo de referência arquitetural (*Architectural Reference Model – ARM*) para a Internet das Coisas. O ARM consiste em três partes:

- O modelo de referência para a Internet das Coisas, que fornece o nível de abstração mais alto para a definição do ARM e promove um entendimento comum do domínio da Internet das Coisas;
- A arquitetura de referência para a Internet das Coisas, que é a referência para a construção de arquiteturas de Internet das Coisas compatíveis e fornece pontos de vista e perspectivas sobre diferentes aspectos arquitetônicos;
- Diretrizes que discutem como os modelos, visões e perspectivas, definidos nas outras duas partes, podem ser de fato utilizados.

A Figura 3 apresenta uma visão geral do processo utilizado para a definição das diferentes partes que constituem o ARM.

3.3 P2413 - Standard for an Architectural Framework for the Internet of Things (IoT)

O projeto denominado *P2413 - Standard for an Architectural Framework for the Internet of Things (IoT)* [2, 3] foi iniciado por orientação da equipe de Internet das Coisas da *IEEE Standards Association* e visa suprir as necessidades do mercado, através do desenvolvimento do cenário tecnológico da Internet das Coisas. O objetivo principal do P2413 é o fornecimento de um *framework* arquitetural extensível e integrado para a Internet das Coisas. A intensão do grupo de trabalho envolvido no projeto é entregar um *framework* comum para todos os domínios da Internet das Coisas, a fim de aumentar a transparência arquitetural e o suporte a *benchmarking*, segurança e proteção. Além disso, o grupo de trabalho tem intenção de promover a interação entre domínios e, para tal, pretende estabelecer as descrições de vários domínios de Internet das Coisas, criar definições de abstrações de domínio e identificar semelhanças entre os diferentes domínios de Internet das Coisas. O

⁴Disponível em: <https://iotforum.org/wp-content/uploads/2014/09/D1.5-20130715-VERYFINAL.pdf>. Acesso em: 02 abril 2019.

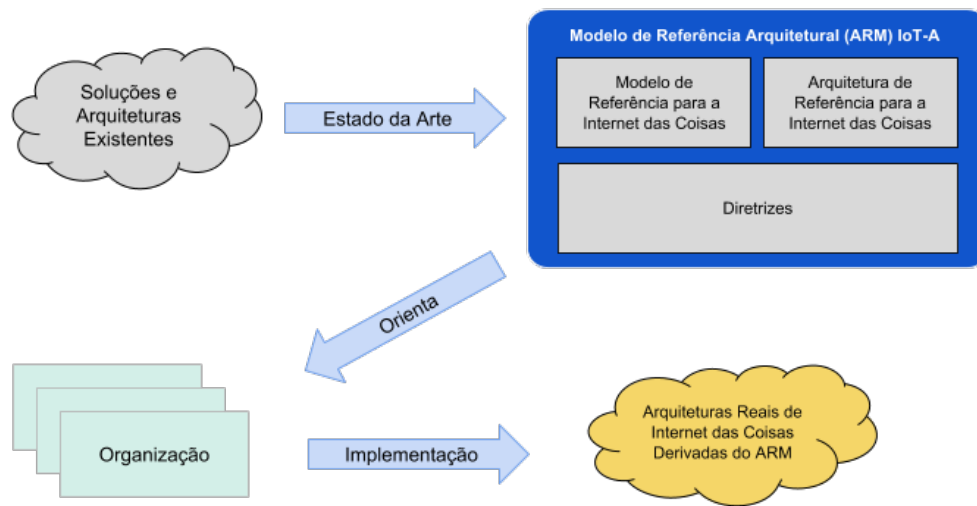


Figure 3: Blocos de construção do Modelo de Referência Arquitetural (ARM) para a Internet das Coisas.

Fonte: Adaptado de F.E. Project, 2013⁴.

grupo também visa contribuir para a interoperabilidade do sistema e impulsionar ainda mais o crescimento do mercado de Internet das Coisas. Atualmente, não há nenhuma documentação técnica sobre o projeto disponível.

3.4 Aspectos Arquiteturais a Nível de Plataformas

[13] definem um *framework* para a avaliação de um conjunto de plataformas, com o objetivo de verificar o quão amplamente elas dão suporte para as possíveis necessidades dos fornecedores de aplicações de Internet das Coisas, ao longo de todo o ciclo de vida do seu serviço ou aplicativo. Através do *framework* e da documentação disponível sobre características e serviços, é avaliada a maturidade das plataformas de Internet das Coisas. A conclusão dos autores, através dos resultados obtidos, é de que o suporte fornecido está fragmentado e, com isso, algumas das atividades, como a descoberta, compra, contabilidade e o faturamento de aplicativos e serviços, são apenas parcialmente suportadas pelas plataformas. Isso contribui para a baixa adesão e indica a perspectiva na qual as plataformas precisam ser melhoradas no futuro.

[8] abordam a heterogeneidade de plataformas em nuvem para a Internet das Coisas, evidenciando os problemas decorrentes dos diversos padrões e abordagens existentes. Para discorrer sobre esses problemas, os autores propõem uma arquitetura de referência abstrata para a Internet das Coisas, baseada em diversas plataformas, e realizam a comparação da arquitetura proposta com quatro plataformas de Internet das Coisas, três de código aberto e uma proprietária. Como conclusão, os autores apontam que a arquitetura de referência do artigo pode ser utilizada como base para a comparação e a avaliação de diferentes soluções para a Internet das Coisas, fornecendo um referencial comum para o desenvolvimento de novas plataformas.

[14] analisam um conjunto significativo de plataformas de Internet das Coisas, proprietárias e de código aberto, avaliando-as em relação a habilidade de atender às expectativas de diferentes usuários. A avaliação foi realizada como uma análise das lacunas do panorama atual da Internet das Coisas em relação ao suporte para tecnologias heterogêneas de sensores e atuadores, à propriedade dos dados e suas implicações para a segurança e para a pri-

vacidade, às capacidades de processamento e compartilhamento de dados, ao suporte oferecido aos desenvolvedores de aplicações, à plenitude do ecossistema de Internet das Coisas e à disponibilidade de *marketplaces* dedicados à Internet das Coisas. Essa análise visa destacar as deficiências das soluções atuais para melhorar a sua integração com os ecossistemas futuros. Com bases nos resultados da análise, os autores apresentam uma lista de recomendações para incremento nas plataformas de Internet das Coisas, de forma que as lacunas possam ser preenchidas.

[18] analisa plataformas em nuvem de Internet das Coisas, visando avaliar diferentes domínios de serviço, como desenvolvimento de aplicativos, gerenciamento de dispositivos, gerenciamento de sistemas, gerenciamento de heterogeneidade, gerenciamento de dados, ferramentas de análise, implantação, monitoramento, visualização e pesquisa. Uma comparação é apresentada para divulgar as plataformas de acordo com a sua aplicabilidade. Além disso, são descritos alguns problemas das plataformas em nuvem de Internet das Coisas, que deverão ser enfrentados pelos pesquisadores no futuro. O objetivo do autor é fornecer uma visão detalhada sobre um conjunto de plataformas em nuvem de Internet das Coisas e apontar os prós e contras de cada uma delas.

4. SELEÇÃO DE PLATAFORMAS PARA O ESTUDO DE CASO

Para realização do estudo de caso, um conjunto de plataformas foi selecionado para fundamentar a identificação de elementos e funcionalidades para uma possível arquitetura de referência para a Internet das Coisas. São elementos desse conjunto as seguintes plataformas: Arrayent, Axeda, Bugswarm, Carriots, DeviceHive, DSA, Evrythng, Exosite, GrooveStreams, IFTTT, Kaaproject, LinkSmart, Mbed, Nimbits, Particle.io, Autodesk SeeControl, SensorCloud, SiteWhere, PTC ThingWorx, Thinger.io, Thingsboard, ThingSpeak, WSo2 e Zetta. Essas plataformas foram selecionadas a partir de uma publicação do *site H2S Media* [20] que apresenta uma lista, segundo uma visão técnico mercadológica, das nove melhores plataformas de código aberto para desenvolvimento de projetos para a Internet das Coisas e a partir do artigo de [24] que apresenta o estado da arte do desenvolvimento de plataformas para a Internet das Coisas, explorando uma coleção de plataformas que possuem

“uma base de clientes significativa e parcerias com fabricantes de dispositivos e integradores de sistemas” [24]⁵.

Realizou-se uma pré-análise do conjunto de plataformas selecionado através da documentação de cada uma das plataformas. Buscou-se identificar quais plataformas apresentam uma estrutura e um acervo de funcionalidades significativos e que, dessa forma, apresentam informação suficiente para uma análise mais ampla.

A Tabela 1 apresenta, sucintamente, o resultado da pré-análise de cada uma das plataformas do conjunto definido originalmente. A nomenclatura das plataformas pode divergir em relação ao supramencionado, pois foram encontradas algumas diferenças entre as informações presentes nos artigos utilizados para a definição do conjunto de plataformas e o conteúdo das documentações oficiais das mesmas.

Após a pré-análise das plataformas (Tabela 1), constatou-se que as plataformas DeviceHive, LinkSmart e SiteWhere apresentam uma estrutura e um acervo de funcionalidades significativos, além de possuírem um detalhamento sobre sua arquitetura disponível em documentação específica, representando, dessa forma, o conjunto final de plataformas para o estudo de caso.

5. RESULTADOS

Nesta seção é apresentada, para as plataformas selecionadas (*i.e.*, DeviceHive [7], LinkSmart [10] e SiteWhere [11]), uma síntese da sua documentação, com destaque para o detalhamento da arquitetura empregada. Por fim, são indicados elementos e funcionalidades, inferidos através das plataformas analisadas, para aplicação em uma possível arquitetura de referência para a Internet das Coisas. Adicionalmente, discorre-se acerca das consequências quando da aplicação dos elementos e funcionalidades identificados.

5.1 DeviceHive

A plataforma de código aberto DeviceHive [7] visa possibilitar o gerenciamento de dispositivos inteligentes e a sua intercomunicação. A plataforma é baseada em microsserviços, uma arquitetura onde as aplicações são desmembradas em componentes mínimos e independentes, que funcionam em conjunto para realizar as mesmas tarefas [19]. A plataforma é escalável, independe do hardware e da nuvem utilizados e possui APIs para o gerenciamento de dispositivos, baseadas em diferentes protocolos, o que permite configurar e monitorar a conectividade dos dispositivos, controlá-los e analisar seu comportamento.

Os dispositivos podem se conectar à plataforma utilizando API REST, *WebSockets* ou o protocolo *Message Queuing Telemetry Transport* (MQTT) [23]. A estrutura da DeviceHive oferece suporte a bibliotecas escritas em várias linguagens de programação, incluindo Java, Node.js e Python, o que a torna independente de dispositivo, possibilitando inclusive a conexão de dispositivos de baixo nível como, por exemplo, o microcontrolador ESP8266, do fabricante chinês Espressif.

Todas as comunicações da plataforma são realizadas através de mensagens no formato *JavaScript Object Notation* (JSON). Além disso, para todos os serviços RESTful, a DeviceHive fornece o *Swagger*, um *framework* que possui ferramentas para o desenvolvimento de APIs, que permite testar a instalação e explorar as capacidades da plataforma. Os métodos fornecidos para implantação e operação da plataforma são: através das tecnologias *docker*, *docker-compose*, *kubernetes* ou através da instalação manual.

⁵p. 217, tradução nossa para “*a significant customer base and partnerships with device manufacturers and system integrators*”

⁶Diz-se que um serviço é RESTful se o mesmo segue os princípios da arquitetura de transferência de estado representacional (*Representational State Transfer* – REST).

A DeviceHive possui sua própria interface gráfica do usuário, o painel de administração. Ele permite criar os dispositivos, conectá-los a redes específicas, gerenciar usuários e gerar *JSON Web Tokens* (JWT), utilizados para o serviço de autenticação da plataforma. Também permite visualizar notificações recentes, enviar comandos para os dispositivos e modificar o tipo do dispositivo ou outras características técnicas. Além do painel de administração, a DeviceHive disponibiliza também um *datasource* para conexão à Grafana, plataforma utilizada para visualização, análise e monitoramento dos dados dos dispositivos.

Por ser um sistema de código aberto baseado em microsserviços, é permitida a realização de quaisquer mudanças na plataforma. Para contribuir com essas mudanças, existem os *plugins* DeviceHive, pequenas aplicações que recebem notificações da plataforma e implementam o caso de uso requisitado.

5.1.1 Arquitetura de Microsserviços DeviceHive

A construção da DeviceHive desenvolveu-se com ênfase nas questões de escalabilidade e disponibilidade. A plataforma permite a interconexão e é capaz de coletar dados de centenas de dispositivos simultaneamente. Além disso, pode ser escalonada para a quantidade necessária de instâncias, de forma a garantir a segurança e a disponibilidade dos dados. Na Figura 4 estão apresentados os componentes que constituem a arquitetura de microsserviços da plataforma DeviceHive e suas relações. Na sequência há uma descrição sucinta de cada componente.

Os papéis que podem se conectar com a plataforma são os usuários (*users*), os dispositivos (*devices*), a interface painel de administração (*admin panel*) e a plataforma Grafana. Como já mencionado, essa conexão pode ser feita através de API REST, de *WebSockets* ou do MQTT. Além disso, deve ser realizada uma requisição de autenticação para a plataforma, através do serviço de autenticação DeviceHive (*DH Auth Service*). Toda autenticação na plataforma é realizada através de *tokens* JWT. Os *tokens* contêm todas as informações sobre privilégios de usuários, dispositivos disponíveis, redes e tipos de dispositivos. O serviço de autenticação fornece uma API RESTful para geração, validação e atualização desses *tokens*.

A DeviceHive tem um *plugin* específico (*DH MQTT Connector*) para suporte à utilização do protocolo de comunicação MQTT, que utiliza o padrão publicação/assinatura. Nesse padrão, existem dois tipos de entidades: o cliente, que pode ser qualquer tipo de dispositivo capaz de interagir com o *broker*, assinando tópicos de mensagem de interesse e publicando mensagens em tópicos específicos e o *broker*, que é um servidor que recebe as mensagens dos clientes e as envia para os assinantes do tópico ao qual a mensagem pertence [23].

O serviço *front-end* (*DH Frontend Service*) fornece a API RESTful e a API *WebSocket* para conexão com a plataforma DeviceHive. Esse serviço é responsável pelas verificações preliminares, enviando requisições ao serviço *back-end* e entregando as respostas recebidas, de forma assíncrona. Além disso, também realiza algumas consultas simples ao banco de dados.

O serviço de gerenciamento de *plugins* (*DH Plugin Management Service*) é um microsserviço adicional da DeviceHive que permite gerenciar os comandos dos dispositivos e as assinaturas em um tópico específico do *broker*. O serviço é integrado com o *Swagger* para permitir que os *plugins* sejam registrados, removidos e atualizados.

O servidor *proxy WebSocket* é um componente do serviço de gerenciamento de *plugins*, desenvolvido em Node.js, que envolve algumas das funcionalidades essenciais do *broker* de mensagens,

⁷Disponível em: <https://files.readme.io/1a1bd86-Slide2.png>. Acesso em: 04 abril 2019.

Table 1: Pré-análise de plataformas candidatas ao estudo de caso.

Plataforma	Pré-análise
IoT Service X-Change (Arrayent)	A plataforma Arrayent foi adquirida em julho de 2017 pela empresa Prodea Systems. Atualmente, a empresa trabalha com a plataforma IoT Service X-Change, que utiliza os serviços da plataforma Arrayent. A documentação existente consiste apenas em uma visão de negócios sobre as plataformas e um breve detalhamento das mesmas.
Axeda	A plataforma Axeda foi adquirida em julho de 2014 pela empresa PTC Inc., que também é dona da plataforma ThingWorx. Constatou-se que apenas a plataforma ThingWorx é apresentada como solução para a Internet das Coisas, de forma que as funcionalidades da plataforma Axeda foram utilizadas como um complemento. A plataforma demonstra ter sido descontinuada.
BugSwarm	A plataforma BugSwarm utiliza a tecnologia <i>JavaScript</i> e o <i>Hypertext Transfer Protocol</i> (HTTP) para criar um <i>swarm</i> , que nada mais é do que um sistema de recursos que podem se comunicar através da utilização de uma política de acesso definida. Partindo dessa concepção, entende-se que a plataforma não oferece uma solução para a Internet das Coisas propriamente dita; porém, fornece condições para que aplicações de Internet das Coisas sejam desenvolvidas.
Altair SmartCore (Carriots)	Atualmente, a plataforma Carriots é mantida pela empresa Altair com o nome de Altair SmartCore. Essa plataforma é <i>cloud-native</i> e oferece um conjunto integrado de serviços e funcionalidades para conectar os dispositivos ao mundo digital. A documentação sobre a plataforma ainda está em construção, contemplando apenas algumas informações nas perspectivas de negócios e de utilização da plataforma.
DeviceHive	A plataforma DeviceHive oferece uma solução de código aberto com um grande número de opções de integração. A documentação sobre a plataforma é detalhada e está disponível publicamente.
DSA	A plataforma de código aberto DSA (<i>Distributed Services Architecture</i>) promete facilitar a intercomunicação, a lógica e as aplicações dos dispositivos em todas as camadas da infraestrutura da Internet das Coisas. Entretanto, a documentação disponível sobre a plataforma é incipiente.
Evrythng	A plataforma Evrythng oferece uma solução completa para a Internet das Coisas; porém, é uma plataforma proprietária e não apresenta informação suficiente sobre sua arquitetura.
Murano (Exosite)	A plataforma Murano, da empresa Exosite, oferece um conjunto de soluções para Internet das Coisas, porém é uma plataforma proprietária e a documentação é insuficiente para análise.
GroveStreams	A GroveStreams é uma interface de programação de aplicações (<i>Application Programming Interface – API</i>) RESTful ⁶ de código aberto mas com pouca documentação disponível.
IFTTT	A IFTTT (acrônimo de <i>If This, Then That</i>) define um protocolo conciso que é implementado por APIs de serviço e, por isso, pode ser vista como uma ferramenta de apoio para o desenvolvimento de soluções de Internet das Coisas, ao invés de uma plataforma de Internet das Coisas.
KaaProject	A plataforma de <i>middleware</i> KaaProject permite a implementação de soluções completas para a Internet das Coisas. Tem documentação disponível, com um vasto conjunto de informações detalhadas sobre sua arquitetura e implementação.
LinkSmart	A plataforma de código aberto LinkSmart oferece soluções para o desenvolvimento de aplicações em vários domínios da Internet das Coisas. Tem documentação detalhada disponível.

Plataforma	Pré-análise
Pelion (Mbed)	A plataforma Pelion, em conjunto com o sistema operacional Mbed, oferece uma solução para a ampliação do gerenciamento da conectividade, dos dispositivos e dos dados. A documentação disponível não é satisfatória para uma análise detalhada da plataforma.
Nimbits	A plataforma de código aberto Nimbits foi criada em 2001, originalmente como um sistema de coleta de dados, destinada ao controle e automatização de processos envolvidos em APIs de Internet das Coisas. Atualmente, o projeto Nimbits encontra-se arquivado.
Particle.io	A plataforma de código aberto Particle.io oferece uma solução completa para o desenvolvimento de aplicações de Internet das Coisas. Entretanto, a documentação disponível não apresenta informações satisfatórias para análise.
AutoDesk Fusion 360 (SeeControl)	O projeto da AutoDesk responsável pela plataforma SeeControl foi encerrado e, atualmente, as iniciativas da empresa na área de Internet das Coisas concentram-se na plataforma proprietária Fusion 360. Essa plataforma oferece a integração entre o Desenho Assistido por Computador (DAC), a Manufatura Auxiliada por Computador (MAC) e a Engenharia Assistida por Computador (EAC), permitindo a união dessas tecnologias em uma única plataforma. Em síntese, a plataforma oferece solução apenas para um domínio específico, além de não ter documentação detalhada publicamente disponível.
SensorCloud	A plataforma SensorCloud é <i>web-based</i> e pode ser gerenciada remotamente para permitir o armazenamento, a visualização e a análise de dados de sensores. Entretanto, a plataforma foca na coleta e gerenciamento de dados e disponibiliza pouca documentação para sua análise.
SiteWhere	A plataforma de código aberto SiteWhere oferece uma estrutura personalizável para praticamente qualquer caso de uso de Internet das Coisas, como dispositivos vestíveis, automação residencial, entre outros. Há uma extensa documentação acerca da plataforma publicamente disponível.
Thingier.io	A plataforma de código aberto Thingier.io fornece uma solução completa para o desenvolvimento de aplicações de Internet das Coisas. Tem documentação disponível, entretanto, não possui um detalhamento satisfatório sobre sua arquitetura.
ThingsBoard	A plataforma de código aberto ThingsBoard fornece solução para a coleta de dados, o processamento, a visualização e o gerenciamento de dispositivos. Possui documentação detalhada disponível.
ThingSpeak	A plataforma ThingSpeak permite que sejam realizados a coleta, o armazenamento, a visualização e a ação sobre os dados de sensores e atuadores. O diferencial da plataforma é que é desenvolvida pela MathWorks e, por isso, pode ser integrada com o MathLab para a análise dos dados coletados. Entretanto, a documentação disponível não permite uma análise detalhada.
PTC ThingWorx	A plataforma ThingWorx, da empresa PTC Inc., oferece uma suíte de funcionalidades para construção, implantação e gerenciamento de aplicações de Internet das Coisas. Possui documentação disponível e detalhada; porém, não foram encontradas informações específicas sobre a arquitetura da plataforma.
WSO2	A plataforma WSO2 oferece uma solução ágil de integração e constitui uma estrutura ampla para desenvolver, reutilizar e gerenciar integrações. É arquitetada em torno de uma base de código comum de tecnologias de integração de código aberto. Seus componentes podem ser usados individualmente ou como uma plataforma coesa de integração ágil. Numa perspectiva mais conservadora, não se trata de uma plataforma de Internet das Coisas de fato.
Zetta	A plataforma de código aberto Zetta é baseada na tecnologia <i>Node.js</i> e oferece um <i>toolkit</i> completo para geração de APIs para os dispositivos. O detalhamento da documentação disponível se mostra insatisfatório para análise.

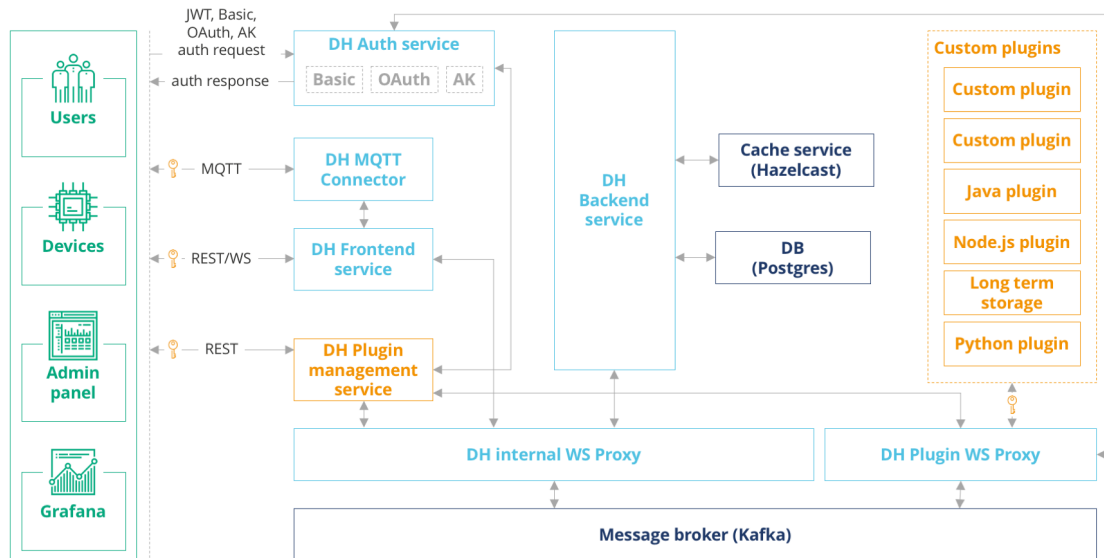


Figure 4: Componentes arquiteturais da plataforma DeviceHive.

Fonte: DeviceHive, 2018⁷.

permitindo a comunicação com o *broker* através de *WebSockets*. O *proxy WebSocket* interno (*DH Internal WS Proxy*) estabelece a comunicação entre os microsserviços da plataforma DeviceHive. O *proxy WebSocket* de *Plugin* (*DH Plugin WS Proxy*) permite que os usuários comuniquem-se com os *plugins* criados.

A DeviceHive utiliza a plataforma Apache Kafka como o *broker* de mensagens. O Kafka cuida da comunicação entre os serviços e realiza o balanceamento de carga entre eles.

O serviço *back-end* (*DH Backend Service*) é responsável por armazenar os dados no serviço de *cache* e no banco de dados, gerenciar as assinaturas no *broker* e recuperar dados, da *cache* ou do banco de dados, requisitados por outros serviços. Todas as comunicações com o serviço *back-end* são realizadas através do *broker*, pois não há uma API pública de acesso.

O armazenamento em sistema de gerenciamento de banco de dados relacional é necessário para persistir todos os metadados. Isso inclui dados sobre dispositivos, redes, usuários, tipos de dispositivos e definições de configuração importantes; ou seja, todos os dados que não são de série temporal. Na DeviceHive, esse armazenamento é realizado através do PostgreSQL.

Para armazenar os dados de série temporal, a plataforma DeviceHive emprega a tecnologia *In-Memory Data Grid* (IMDG), que utiliza a memória como área de armazenamento enquanto uma tarefa estiver sendo processada, ao invés de gravar os dados diretamente no banco. Isso é realizado porque nem sempre é necessário armazenar dados desse tipo por um longo período de tempo; porém, o acesso imediato aos dados é sempre necessário. Para esse propósito, a DeviceHive utiliza o Hazelcast, que é uma IMDG de código aberto baseada em Java. Dessa forma, todas as mensagens são armazenadas utilizando o serviço de *cache* distribuído, para que o acesso seja mais rápido, e removidas após dois minutos, por padrão.

5.2 LinkSmart

A LinkSmart é uma plataforma de código aberto para o desenvolvimento de aplicações de Internet das Coisas em diversos domínios,

como cidades inteligentes, Indústria 4.0 e outros. A plataforma fornece blocos de construção como serviços para a implementação eficiente de aplicações na Internet das Coisas. Esses serviços incluem a abstração de dispositivos, o armazenamento de dados, o gerenciamento de dados em tempo real e serviços avançados como mineração de dados e aprendizado de máquina. Seguindo o padrão dos microsserviços, os serviços da LinkSmart podem ser unidos dependendo do caso de uso no qual serão aplicados.

5.2.1 Componentes da Arquitetura de Microsserviços

Na Figura 5 estão ilustrados os componentes que constituem a plataforma LinkSmart e a infraestrutura de desenvolvimento e operação. Os componentes da plataforma são agrupados nas categorias visualização, segurança, provisionamento de serviços, gerenciamento de dados e aprendizado de máquina e abstração e integração de dispositivos. Na sequência, é apresentado um breve detalhamento para cada uma das categorias.

Para a pronta visualização (*Visualization*) dos dados produzidos pelos dispositivos, a LinkSmart faz uso da plataforma Grafana e *plugins* relacionados.

Na segurança (*Security*), a plataforma oferece serviços para a autorização, autenticação e contabilização nos protocolos comuns da Internet das Coisas. Nesse sentido, o *gateway* de borda da LinkSmart (*Border Gateway*) fornece um ponto único de entrada para um sistema autônomo qualquer de Internet das Coisas, que é constituído por dispositivos conectados e seus serviços auxiliares. As principais funcionalidades do *gateway* de borda são:

- O *Security Socket Layer* (SSL) *offloading*, que consiste na realização de criptografia e descriptografia SSL em um servidor dedicado, na borda dos sistemas autônomos protegidos;

⁸Disponível em: <https://docs.linksmart.eu/display/HOME/What+is+LinkSmart>. Acesso em: 04 abril 2019.

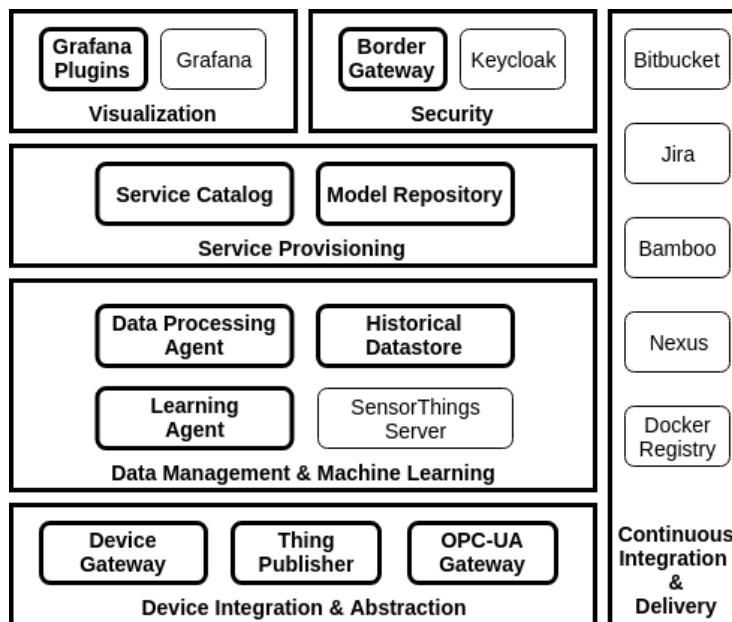


Figure 5: Componentes da plataforma LinkSmart.

Fonte: LinkSmart, 2018⁸.

- A autenticação e autorização para requisições HTTP e MQTT, onde usuários e suas permissões podem ser definidos usando um Provedor de Identidade em conformidade com o OpenID. Nesse sentido, o Keycloak é um exemplo de solução que pode ser utilizada como Provedor de Identidade;
 - Permissões para requisições HTTP podem ser definidas no nível de *endpoints* REST e métodos HTTP autorizados;
 - Permissões para requisições MQTT podem ser definidas no nível de tópicos e comandos MQTT (publicar, assinar, etc.).
- O envio de requisições HTTP para os serviços internos pode ser realizado através da definição de pseudônimos;
- Os endereços internos em respostas HTTP são traduzidos para endereços externos com os quais o requisitante é capaz de lidar.

Em uma infraestrutura de Internet das Coisas, o número de serviços e dispositivos pode crescer rapidamente e seus estados e conectividade podem mudar de forma dinâmica. O provisionamento de serviços (*Service Provisioning*) da LinkSmart auxilia no gerenciamento de tal infraestrutura, acompanhando os dispositivos, serviços, sistemas e subsistemas em execução. Os componentes utilizados para o provisionamento de serviços são o catálogo de serviços (*Service Catalog*) e o repositório de modelos (*Model Repository*).

O catálogo de serviços é a porta de entrada para os serviços e outros componentes. Sua funcionalidade cobre principalmente o descobrimento de serviços disponíveis, como *brokers*, *gateways* e outros. O catálogo de serviços é uma API RESTful que permite o registro e a atualização dos serviços, a navegação nas entradas do catálogo para que aplicações e outros clientes descubram os serviços registrados, a recuperação de informações sobre serviços específicos e a filtragem de serviços, para clientes procurando por serviços de acordo com suas competências.

O repositório de modelos da LinkSmart é um serviço que fornece as operações de armazenamento e pesquisa para a documentação de modelo de domínio. O modelo de domínio é representado no formato *XML Metadata Interchange* (XMI) e diferentes representações, como JSON, são geradas e fornecidas por esse serviço.

Muitas aplicações de Internet das Coisas lidam com um grande volume de dados, gerado por sensores e outros dispositivos. Dependendo do caso de uso, esses dados precisam ser armazenados de forma simples e processados de maneira mais complexa. Os componentes de gerenciamento de dados e aprendizado de máquina (*Data Management & Machine Learning*) levam em conta essa flexibilidade necessária e permitem coletar, armazenar e consultar de forma eficiente os dados de série temporal produzidos pelos sensores, além de realizar a mineração contínua dos dados, a análise sob demanda, o aprendizado contínuo e processamentos complexos. São componentes o armazenamento histórico de dados (*Historical Datastore*) e os agentes de processamento de dados (*Data Processing Agent*) e de aprendizado (*Learning Agent*).

O armazenamento histórico de dados da LinkSmart é um serviço modular para o armazenamento de dados de série temporal. A intenção do serviço é possuir múltiplas implementações usando diferentes *back-ends* e oferecendo diferentes níveis de funcionalidades implementadas. Diferentes implementações podem aceitar os dados originados dos sensores em diferentes formatos, e também suportar múltiplos formatos simultaneamente. Todas as implementações suportam o modelo de dados *Sensor Measurement Lists* (SenML) e podem oferecer formatos adicionais usando negociação via HTTP. Uma implementação completa do armazenamento histórico de dados oferece três APIs:

- API de registro: Realiza o registro dos metadados do sensor e dos detalhes acerca da coleta e armazenamento dos dados do sensor;
- API de dados: Realiza a submissão e a recuperação das medições brutas dos sensores;
- API de coleta: Realiza a recuperação das medições coletadas

dos sensores.

Os agentes de Internet das Coisas foram desenvolvidos para todos os tipos de processamento de dados com armazenamento reduzido, desde a anotação de dados até técnicas complexas de aprendizado de máquina. Os agentes atuam como um processador e parcialmente como um manipulador de mensagens, e são ideais para o gerenciamento inteligente e sob demanda dos dados ou para a análise dos ambientes de Internet das Coisas, desde a computação de borda até a computação em nuvem. Os agentes podem ser utilizados como um serviço autossuficiente de borda ou como um nó computacional da nuvem e oferecem o processamento de eventos complexos como serviço e o aprendizado de máquina em tempo real como serviço.

Os serviços de abstração e integração de dispositivos (*Device Integration & Abstraction*) da LinkSmart são responsáveis por dois requisitos básicos da Internet das Coisas: a conexão de dispositivos ou subsistemas ao mundo da rede IP e a abstração dos dispositivos ou subsistemas em um formato unificado de API e dados. Para isso, a LinkSmart fornece os conectores *gateway* de dispositivo (*Device Gateway*), *Thing Publisher* e *NiFi OPC-UA Gateway*.

O *gateway* de dispositivo (*Device Gateway*) fornece uma implementação do conector de dispositivo, que por sua vez fornece a integração de dispositivos heterogêneos, simplificando a integração de diversos dispositivos de Internet das Coisas. Implementando a funcionalidade do conector de dispositivo, o *gateway* de dispositivo age como um intermediário entre o protocolo de acesso ao *hardware* de baixo nível e a rede TCP/IP. O *Thing Publisher* destina-se à exposição contínua dos dados de sensores proprietários no formato *SensorThing* do *Open Geospatial Consortium* (OGC). O *NiFi OPC-UA Gateway* consiste em um *gateway* baseado na arquitetura unificada OPC (*Open Platform Communications Unified Architecture*) e que utiliza o projeto NiFi da Apache.

Na Figura 6 está representado o fluxo de dados entre os diferentes componentes que compõem a plataforma LinkSmart. Os componentes que possuem um círculo em negrito são aqueles desenvolvidos pela própria plataforma.

5.3 SiteWhere

A SiteWhere é uma plataforma de código aberto para criação da infraestrutura e dos aplicativos que constituem a Internet das Coisas. Em sua versão atual (2.0), a plataforma adota uma abordagem completamente distribuída, usando microsserviços para permitir o dimensionamento de componentes, de forma que o sistema possa se adaptar a um caso de uso específico. O sistema é construído através da abordagem de *framework*, utilizando APIs bem definidas, de forma que novas tecnologias podem ser facilmente integradas, à medida em que o ecossistema da Internet das Coisas evolui.

5.3.1 Visão Geral da Arquitetura de Microsserviços

Na arquitetura de microsserviços da plataforma SiteWhere, cada microsserviço controla um subconjunto específico de funcionalidades que é claramente definido e planejado a partir do trabalho realizado por outros microsserviços. Isso permite que partes do sistema sejam dimensionadas independentemente, enquanto outras partes podem ser retiradas completamente, caso não utilizadas. A abordagem de microsserviços também divide o código, fazendo com que seu entendimento e gestão sejam mais fáceis do ponto de vista de desenvolvimento. O diagrama da Figura 7 ilustra os microsserviços da plataforma e o fluxo geral de dados entre eles.

⁹Disponível em: <https://docs.linksmart.eu/display/HOME/LinkSmart+Dataflow>. Acesso em: 4 abril 2019.

A conectividade e a troca de dados entre microsserviços na plataforma SiteWhere é realizada através do gRPC, um sistema de chamada remota ao procedimento desenvolvido pelo Google. A SiteWhere também introduz o conceito de demultiplexador de API, mecanismo capaz de analisar a topologia da instância atual e adicionar ou remover conexões entre microsserviços dinamicamente. À medida em que o número de serviços é ampliado ou reduzido, a plataforma automaticamente conecta ou desconecta a ligação entre eles. Todas as comunicações entre microsserviços na plataforma são realizadas através desse mecanismo.

O SiteWhere utiliza o Hazelcast para fornecer uma *cache* distribuída para armazenamento de um subconjunto de dados principais, como dispositivos e atribuições, que raramente são atualizados. Essa *cache* é consultada antes de se executar uma requisição ao banco de dados, para tentar evitar o custo de uma leitura diretamente do banco.

Na sequência, são descritos os microsserviços principais incluídos na plataforma SiteWhere, conforme ilustrado na Figura 7. Cada microsserviço controla uma área específica de funcionalidades do sistema e é independente de outros microsserviços em termos de tempo de execução, armazenamento de dados e configuração. Entretanto, alguns microsserviços tem dependências nas APIs oferecidas por outros microsserviços e não podem ser executados de forma isolada.

O microsserviço de gerenciamento de instâncias (*Instance Microservice*) é utilizado para a inicialização de uma instância da SiteWhere, com a definição da estrutura inicial de configuração do serviço centralizado Apache Zookeeper, que é exigida pelos outros microsserviços.

O microsserviço de gerenciamento global de usuários (*User Management Microservice*) fornece as principais APIs e a persistência dos dados utilizados para o gerenciamento dos usuários do sistema. É usado inicialmente pelo microsserviço de gerenciamento de instâncias para inicializar o sistema com os usuários base. Depois disso, é acionado pelo microsserviço Web/REST para permitir o gerenciamento da lista de usuários.

O microsserviço de gerenciamento de inquilinos¹¹ (*Tenant Management Microservice*) fornece as principais APIs e a persistência dos dados utilizados para o gerenciamento dos inquilinos do sistema. É usado inicialmente pelo microsserviço de gerenciamento de instâncias para inicializar o sistema com os inquilinos base. Depois disso, é acionado pelo microsserviço Web/REST para permitir o gerenciamento da lista de inquilinos do sistema.

O microsserviço de gerenciamento global Web/REST (*Web/REST microservice*) inclui o Apache Tomcat, que fornece infraestrutura para os serviços REST principais, incluindo as interfaces de usuário Swagger. Esse microsserviço é, geralmente, conectado a todos os outros microsserviços no sistema, para que as chamadas da API possam ser delegadas para os microsserviços responsáveis pela implementação das funcionalidades requisitadas.

O microsserviço multi-inquilinos¹² de gerenciamento de dispositivos (*Multitenant Device Management Microservice*) fornece as principais APIs e a persistência dos dados utilizados para o gerenciamento dos modelos de dispositivo (*e.g.*, *sites*, especificações, dispositivos e grupos) para cada inquilino em uma instância da Si-

¹⁰Disponível em: <http://sitewhere.io/docs/2.0.0-rc2/images/platform/microservices-diagram.png>. Acesso em: 4 abril 2019.

¹¹Inquilinos não são usuários individuais, mas empresas clientes do *software* [5].

¹²Uma arquitetura multi-inquilinos permite que múltiplos inquilinos compartilhem os mesmos recursos físicos, mas permanecendo logicamente isolados [5].

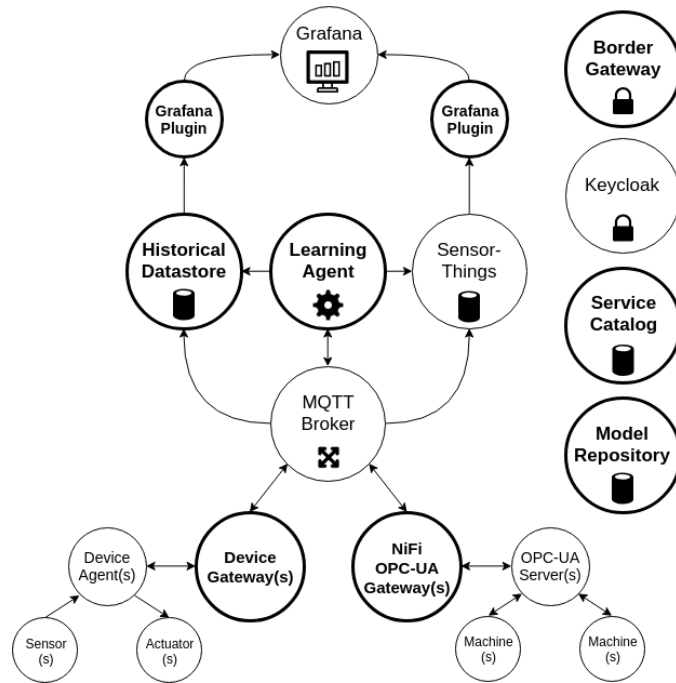


Figure 6: Fluxo de dados entre os componentes da plataforma LinkSmart.

Fonte: LinkSmart, 2018⁹.

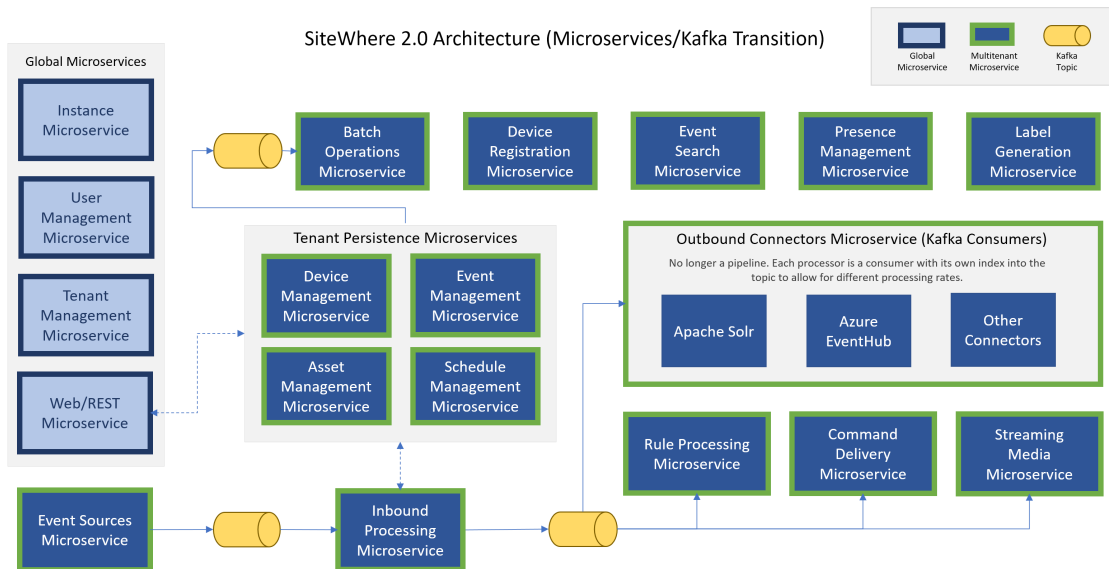


Figure 7: Diagrama de microserviços da plataforma SiteWhere.

Fonte: SiteWhere LLC, 2018.¹⁰

teWhere. O modelo de dispositivo é preenchido inicialmente com base nos *scripts* incluídos no *template* de inquilino, utilizado para a criação do inquilino.

O microserviço multi-inquilinos de gerenciamento de eventos (*Multitenant Event Management Microservice*) fornece as principais APIs e a persistência dos dados utilizados para o gerenciamento dos eventos de dispositivo (e.g., localizações, medições, alertas e invocações de comandos) para cada inquilino em uma instância da SiteWhere. O modelo de evento de dispositivo é preenchido inicialmente com base nos *scripts* incluídos no *template* de inquilino, utilizado para a criação do inquilino.

O microserviço multi-inquilinos de gerenciamento de ativos (*Multitenant Asset Management Microservice*) fornece as principais APIs e a persistência dos dados utilizados para o gerenciamento dos ativos para cada inquilino em uma instância da SiteWhere. O modelo de ativo é preenchido inicialmente com base nos *scripts* incluídos no *template* de inquilino, utilizado para a criação do inquilino.

O microserviço multi-inquilinos de gerenciamento de cronograma (*Multitenant Schedule Management Microservice*) fornece as principais APIs e a persistência dos dados utilizados para o gerenciamento de cronograma para cada inquilino em uma instância da SiteWhere. O modelo de cronograma é preenchido inicialmente com base nos *scripts* incluídos no *template* de inquilino, utilizado para a criação do inquilino.

O microserviço multi-inquilinos de operações em lote (*Multitenant Batch Operations Microservice*) fornece as principais APIs e a persistência dos dados utilizados para o gerenciamento das operações em lote para cada inquilino em uma instância da SiteWhere. O modelo de operações em lote fica vazio até a inicialização dos inquilinos, mas pode ser preenchido invocando APIs que produzem operações em lote. Esse microserviço ainda não está totalmente implementado.

O microserviço multi-inquilinos de fontes de eventos (*Multitenant Event Sources Microservice*) hospeda os mecanismos de inquilino que podem ser configurados para processar dados de diferentes fontes. Alguns exemplos incluem o consumo de dados de tópicos MQTT, *WebSockets*, chamadas REST via PUSH ou PULL, entre outras fontes. Depois de processados, os eventos são decodificados em um modelo de dados padronizado e enviados para um tópico do Kafka específico do inquilino, para processamento adicional.

O microserviço multi-inquilinos de processamento de entrada (*Multitenant Inbound Processing Microservice*) recebe os dados produzidos pelo microserviço de fontes de eventos. Esse microserviço valida os dados de entrada, interagindo com o microserviço de gerenciamento de dispositivos para verificar se o evento de entrada está relacionado com um dispositivo registrado. A carga útil dos dados de entrada é enriquecida com dados sobre dispositivos ou atribuições e, dessa forma, as informações podem ser utilizadas em etapas de processamento subsequentes, sem a necessidade de consultá-las novamente. Se o dispositivo não está registrado, então a carga útil é enviada ao microserviço de registro de dispositivos para processamento adicional. Depois que a carga dos eventos de entrada é enriquecida, os eventos são enviados para o microserviço de gerenciamento de eventos, para persistência.

O microserviço multi-inquilinos de registro de dispositivos (*Multitenant Device Registration Microservice*) recebe dados de um tópico do Kafka preenchidos pelo microserviço de processamento de entrada quando os eventos fazem referência à identificação de *hardware* para um dispositivo que não está registrado no sistema. Cada mecanismo de inquilino tem um gerenciador de registro de dispositivos, que deve ser configurado para indicar como os dispositivos não registrados devem ser tratados.

O microserviço multi-inquilinos de processamento de regras (*Multitenant Rule Processing Microservice*) recebe dados do tópico do Kafka que contém os eventos pré-processados e aplica lógicas condicionais para processar adicionalmente os eventos. Esse microserviço ainda não está totalmente implementado.

O microserviço multi-inquilinos de entrega de comandos (*Multitenant Command Delivery Microservice*) recebe dados do tópico do Kafka que contém os eventos pré-processados e, para invocações de comandos, controla o processamento dos comandos. Isso inclui o uso de restrições de roteamento e destinos de comandos que indicam como os comandos devem ser codificados, qual transporte deve ser usado e onde os comandos devem ser entregues.

O microserviço multi-inquilinos de conectores de saída (*Multitenant Outbound Connectors Microservice*) recebe dados do tópico do Kafka que contém os eventos pré-processados e permite que os dados dos eventos sejam encaminhados para outros pontos de integração de forma assíncrona.

O microserviço multi-inquilinos de gerenciamento de presença (*Multitenant Presence Management Microservice*) recebe dados do tópico do Kafka que contém os eventos pré-processados e utiliza os dados dos eventos para atualizar o estado de presença do dispositivo. Cada mecanismo de inquilino tem um gerenciador de presença de dispositivos, que é responsável por determinar quando os dispositivos não estão mais presentes e acionar eventos de mudança de estado, que podem ser usados para ativar ações com base na presença ou ausência de um dispositivo.

O microserviço multi-inquilinos de geração de etiquetas (*Multitenant Label Generation Microservice*) atende a requisições da API por recursos de etiquetas, como *QR codes*, códigos de barra e outras etiquetas de dispositivos customizadas. Cada mecanismo de inquilino tem um gerenciador de geração de símbolos, que pode ser customizado para gerar tipos específicos de saída, exclusivos para o inquilino.

O microserviço multi-inquilinos de pesquisa de eventos (*Multitenant Event Search Microservice*) fornece uma API para pesquisar *datasources* externos que contém informações sobre eventos da SiteWhere em um formato não padronizado. Esse microserviço ainda não está totalmente implementado.

O microserviço multi-inquilinos de *streaming* de mídia (*Multitenant Streaming Media Microservice*) visa permitir o armazenamento de dados binários, como áudio e vídeo. Esse microserviço ainda não está totalmente implementado.

5.4 Elementos e Funcionalidades Identificados

A partir da análise realizada sobre as três plataformas de Internet das Coisas, buscou-se identificar principalmente as características comuns a todas elas, que possam ser utilizadas em uma possível arquitetura de referência para a Internet das Coisas.

O principal fator a ser destacado é que todas as plataformas analisadas fundamentam sua estrutura na arquitetura de microserviços. Como já mencionado, essa arquitetura faz uso de componentes individuais e independentes, que desempenham funcionalidades específicas e que, quando agregados, constituem uma aplicação. [4] já haviam previsto a aproximação entre a arquitetura de microserviços e a Internet das Coisas, por ambas possuírem o mesmo objetivo arquitetural.

Através da análise da estruturação em componentes das plataformas, conclui-se que o emprego de uma arquitetura baseada em microserviços como referencial para o desenvolvimento de aplicações de Internet das Coisas contribui para a resolução de alguns dos principais desafios da Internet das Coisas, descritos na Seção 2.1.

A utilização de microserviços permite que o sistema seja escalonado de forma prática e ágil, pois a inclusão de novos dispositivos

e serviços, implica apenas na inclusão de uma nova instância de determinado microserviço ou na criação de um novo microserviço, evitando a complexidade da realização de mudanças em toda a infraestrutura do sistema. Outro desafio que pode ser solucionado com o uso da arquitetura de microserviços é a disponibilidade, pois os componentes podem ser replicados nos pontos críticos da aplicação, sem a necessidade de se replicar todo o sistema, evitando impactos no desempenho. O gerenciamento também é facilitado, pois a manutenção de componentes individuais de *software* torna-se relativamente mais fácil do que a manutenção de um sistema unificado, por questões de desenvolvimento, como o tamanho do código, por exemplo. Além disso, a confiabilidade também é um ponto positivo da arquitetura de microserviços, pois o impacto das falhas é menor se acontecem em um ambiente controlado, como é o caso dos componentes individuais, reduzindo-se as chances de comprometimento total do sistema.

Um indício significativo da ascensão da arquitetura de microserviços é o fato de que a plataforma SiteWhere, que em sua versão anterior empregava a abordagem monolítica, utilizando módulos que juntos entregavam um único processo, responsável por todas as funções da plataforma, passou a implementar, em sua versão mais recente que ainda está em pré-lançamento, a arquitetura de microserviços, justificando que a nova abordagem tem algumas vantagens em relação à anterior, como a escalabilidade e a criação de componentes que podem ser executados por processos individuais.

Outro fator observado através da análise das plataformas é a capacidade de adaptação a diferentes casos de uso; ou seja, espera-se que uma arquitetura de referência para a Internet das Coisas seja capaz de adaptar-se aos diferentes cenários de aplicações. Nesse sentido, a arquitetura de microserviços também demonstra certa vantagem, pois permite que sejam criados os mais diversos serviços, que podem possuir funcionalidades específicas. Desta forma, para a construção de uma aplicação para determinado cenário, são acoplados apenas os serviços que possuem funcionalidades apropriadas e necessárias para a aplicação. Além de facilitar o desenvolvimento dessas aplicações específicas, esse método não compromete a natureza de uso geral da arquitetura.

Foi possível analisar também que a abordagem publicação e assinatura, através do protocolo MQTT, é um dos principais métodos utilizados para troca de mensagens no desenvolvimento de aplicações de Internet das Coisas. [22] apresentam, inclusive, uma fundamentação que sustenta a afirmação de que o protocolo MQTT tem a seu favor um desempenho, para a Internet das Coisas, superior ao HTTP. As autoras apresentam dois cenários de comparação entre o MQTT e o HTTP: no primeiro cenário é avaliada a largura de banda necessária de acordo com o número de dispositivos e o número de tópicos e, no segundo cenário, avalia-se a largura de banda necessária de acordo com o volume de dados. A partir dos cenários avaliados, as autoras indicam que o número de bytes da transmissão depende do número de dispositivos conectados em um servidor e, além disso, os resultados obtidos implicam que o *overhead* do protocolo HTTP é superior ao do protocolo MQTT. Em particular, se o número de dispositivos conectados aumenta, como é comum em aplicações de Internet das Coisas, o *overhead* é crítico no protocolo HTTP. Ademais, o artigo fundamenta o fato de que o protocolo MQTT exige menos recursos de um servidor, em relação ao protocolo HTTP. Dessa forma, conclui-se também que uma arquitetura de referência deve priorizar a implementação da comunicação segundo a abordagem publicação e assinatura no estilo do protocolo MQTT.

6. CONCLUSÃO

A Internet das Coisas é uma área em constante expansão, que promete contribuir de forma significativa para a evolução e a modernização de diversos setores. Nesse sentido, a realização de pesquisas sobre os tópicos relacionados à Internet das Coisas é de suma importância para a consolidação da área. A definição de uma arquitetura de referência, um dos principais tópicos discutidos atualmente, é fundamental para a padronização de tecnologias e requisitos necessários para a implementação de soluções para a Internet das Coisas.

Existem diferentes iniciativas que tem como objetivo a definição de uma arquitetura de referência para a Internet das Coisas. Nesse trabalho, foram avaliadas três plataformas de Internet das Coisas, selecionadas a partir de um conjunto original de 24 plataformas, para identificar, nas respectivas propostas de arquitetura, elementos e funcionalidades para implementação em uma arquitetura de referência, visando contribuir com o estado da arte das iniciativas supramencionadas.

Como resultados, concluiu-se que a implementação de uma arquitetura baseada em microserviços pelas aplicações de Internet das Coisas traz diversas vantagens em relação aos demais métodos, incluindo a resolução ou abrandamento de desafios como a escalabilidade, a disponibilidade e o gerenciamento. Além disso, inferiu-se também que o protocolo de comunicação mais indicado para utilização em aplicações de Internet das Coisas é baseado na abordagem publicação e assinatura tendo-se como principal referência o protocolo MQTT.

Como trabalho futuro, pode-se ampliar o leque de plataformas analisadas detalhadamente. Adicionalmente, a implementação de aplicações reais permitirão uma avaliação mais realista das capacidades e limitações de cada uma das plataformas. Dentre os principais obstáculos a essas iniciativas, pode-se elencar a dificuldade na obtenção de documentação detalhada acerca de cada plataforma.

7. REFERENCES

- [1] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash. Internet of things: A survey on enabling technologies, protocols, and applications. *IEEE Communications Surveys Tutorials*, 17(4):2347–2376, Fourthquarter 2015. <https://doi.org/10.1109/COMST.2015.2444095>.
- [2] I. S. Association. P2413 - standard for an architectural framework for the internet of things (iot). Disponível em <https://standards.ieee.org/project/2413.html> (Acessado 12 09 2019).
- [3] I. S. Association. Standard for an architectural framework for the internet of things (iot). Disponível em <http://grouper.ieee.org/groups/2413/Intro-to-IEEE-P2413.pdf> (Acessado 12 09 2019).
- [4] B. Butzin, F. Golatowski, and D. Timmermann. Microservices approach for the internet of things. In *2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA)*, pages 1–6, Sept 2016. <https://doi.org/10.1109/ETFA.2016.7733707>.
- [5] C. T. G. E. Chede. Entendendo o modelo multi-tenancy. Disponível em https://www.ibm.com/developerworks/community/blogs/ctaurion/entry/entendendo_o_modelo_multi-tenancy?lang=en (Acessado 12 09 2019).
- [6] I. I. Consortium. The industrial internet of things. volume g1: Reference architecture [online]. Disponível em

- https://www.iiconsortium.org/IIC_PUB_G1_V1.80_2017-01-31.pdf (Acessado 12 09 2019).
- [7] DeviceHive. Guides devicehive. Disponível em <https://docs.devicehive.com/docs> (Acessado 12 09 2019).
- [8] J. Guth, U. Breitenbücher, M. Falkenthal, F. Leymann, and L. Reinfurt. Comparison of iot platform architectures: A field study based on a reference architecture. In *2016 Cloudification of the Internet of Things (CIoT)*, pages 1–6, Nov 2016. <https://doi.org/10.1109/CIOT.2016.7872918>.
- [9] R. Khan, S. U. Khan, R. Zaheer, and S. Khan. Future internet: The internet of things architecture, possible applications and key challenges. In *2012 10th International Conference on Frontiers of Information Technology*, pages 257–260, Dec 2012. <https://doi.org/10.1109/FIT.2012.53>.
- [10] LinkSmart. Linksmart® docs. Disponível em <https://docs.linksmart.eu/> (Acessado 12 09 2019).
- [11] S. LLC. Sitewhere ce 2.0.0-rc2 documentation. Disponível em <http://sitewhere.io/docs/2.0.0-rc2/> (Acessado 12 09 2019).
- [12] H. P. E. D. LP. O que é arquitetura de referência? Disponível em <https://www.hpe.com/br/pt/what-is/reference-architecture.html> (Acessado 12 09 2019).
- [13] O. Mazhelis and P. Tyrväinen. A framework for evaluating internet-of-things platforms: Application provider viewpoint. In *2014 IEEE World Forum on Internet of Things (WF-IoT)*, pages 147–152, March 2014. <https://doi.org/10.1109/WF-IoT.2014.6803137>.
- [14] J. Mineraud, O. Mazhelis, X. Su, and S. Tarkoma. A gap analysis of internet-of-things platforms. *Computer Communications*, 89-90:5 – 16, 2016. Internet of Things Research challenges and Solutions.
- [15] D. Miorandi, S. Sicari, F. D. Pellegrini, and I. Chlamtac. Internet of things: Vision, applications and research challenges. *Ad Hoc Networks*, 10(7):1497 – 1516, 2012. <https://doi.org/10.1016/j.adhoc.2012.02.016>.
- [16] F. E. Project. Internet of things architecture (iot-a) [online]. Disponível em <https://iotforum.org/wp-content/uploads/2014/09/D1.5-20130715-VERYFINAL.pdf> (Acessado 12 09 2019).
- [17] P. Raj and A. C. Raman. *The Internet of things : enabling technologies, platforms, and use*. Taylor & Francis, CRC Press, Boca Raton, FL, 2017.
- [18] P. P. Ray. A survey of iot cloud platforms. *Future Computing and Informatics Journal*, 1(1):35 – 46, 2016. <https://doi.org/10.1016/j.fcij.2017.02.001>.
- [19] I. Red Hat. Introdução aos microsserviços. Disponível em <https://www.redhat.com/pt-br/topics/microservices> (Acessado 12 09 2019).
- [20] H. M. Team. 9 best & top open source iot platforms to develop the iot projects. Disponível em <https://www.how2shout.com/tools/best-opensource-iot-platforms-develop-iot-projects.html> (Acessado 12 09 2019).
- [21] M. Weyrich and C. Ebert. Reference architectures for the internet of things. *IEEE Software*, 33(1):112–116, Jan 2016. <https://doi.org/10.1109/MS.2016.20>.
- [22] T. Yokotani and Y. Sasaki. Comparison with http and mqtt on required network resources for iot. In *2016 International Conference on Control, Electronics, Renewable Energy and Communications (ICCEREC)*, pages 1–6, Sept 2016. <https://doi.org/10.1109/ICCEREC.2016.7814989>.
- [23] M. Yuan. Conhecendo o mqtt. Disponível em <https://www.ibm.com/developerworks/br/library/iot-mqtt-why-good-for-iot/index.html> (Acessado 12 09 2019).
- [24] M. Zdravković, M. Trajanović, J. Sarraipa, R. Jardim-Gonçalves, M. Lezoche, A. Aubry, and H. Panetto. Survey of Internet-of-Things platforms. In *6th International Conference on Information Society and Techology, ICIST 2016*, volume 1, pages 216–220, Kopaonik, Serbia, Feb. 2016. ISBN: 978-86-85525-18-6.