

# Hellfire and COMPaaS Integration Layer

Benito Michelon  
Cristiano Bolla Fernandes

13 de abril de 2015

# Sumário

<b>1</b>	<b>Introdução</b>	<b>4</b>
<b>2</b>	<b>Internet of Things</b>	<b>5</b>
2.1	Aplicações . . . . .	5
2.1.1	Smart City . . . . .	5
2.1.2	Automação Residencial . . . . .	5
2.1.3	Sistemas Médicos e de Saúde Pessoal . . . . .	6
2.2	Plataformas e Ambientes para IoT . . . . .	6
2.2.1	mbed . . . . .	6
2.2.1.1	mbed OS . . . . .	6
2.2.1.2	mbed Device Server . . . . .	7
2.2.2	FlowCloud . . . . .	7
2.2.3	Arrayent . . . . .	8
2.2.3.1	Arrayent Connect Agent . . . . .	8
2.2.3.2	Arrayent Connect Cloud . . . . .	8
2.2.3.3	Arreyent Mobile Framework . . . . .	8
2.2.3.4	Arrayent Cloud Insight . . . . .	8
2.2.4	Samsung IoT Plataform . . . . .	8
2.2.5	IBM Internet of Things Foundation . . . . .	9
2.2.5.1	IBM Bluemix . . . . .	9
2.2.6	SmartSantander . . . . .	9
2.2.6.1	Authentication, Authorisation and Accounting (AAA) . . . . .	10
2.2.6.2	Testbed Management . . . . .	10
2.2.6.3	Experimental Support . . . . .	10
2.2.6.4	Application Support . . . . .	11
2.2.7	FIWARE . . . . .	11
2.2.7.1	FIWARE . . . . .	11
2.2.7.2	FIWARE Lab . . . . .	11
2.2.7.3	FIWARE Ops . . . . .	11
2.2.7.4	FIWARE Accelerate . . . . .	11
2.2.7.5	FIWARE Mundus . . . . .	12
<b>3</b>	<b>Cooperative Middleware Platform as a Service</b>	<b>13</b>
3.1	Application Level Events Interface . . . . .	13
3.2	Middleware Level Events Interface . . . . .	13
3.3	Device Level Events . . . . .	14
<b>4</b>	<b>HellfireOS</b>	<b>15</b>
<b>5</b>	<b>Hellfire and COMPaaS Integration Layer (HAC)</b>	<b>17</b>
5.1	Prova de Conceito . . . . .	18

<b>6</b>	<b>Proposta</b>	<b>19</b>
6.1	Proposta TCC 1 . . . . .	19
6.1.1	Objetivos . . . . .	19
6.1.2	Cronograma . . . . .	19
6.1.3	Lista de Atividades . . . . .	19
6.2	Proposta TCC 2 . . . . .	19
6.2.1	Objetivos . . . . .	19
6.2.2	Cronograma . . . . .	20
6.2.3	Lista de Atividades . . . . .	20

# 1 Introdução

A *Internet of Things* (IoT) é a uma rede de objetos físicos, denominados *Things*, que possuem suporte que lhes permite obter conectividade, fornecendo e trocando dados, a princípio desconexos, que quando utilizados em conjunto com regras de negócio, dão origem a informações de grande valia.

Com a expansão do acesso à internet e a evolução tanto no sentido da redução de custo quanto no sentido de complexidade tecnológica dos hardwares que permitem a conexão de dispositivos eletrônicos, o paradigma de IoT ganhou força. Com esta nova realidade, se torna cada vez mais imprescindível que as soluções surgindo no mercado estejam preparadas para fazer parte deste contexto complexo, criando um grande mercado para empresas visando a criação de plataformas que buscam permitir uma rápida adesão, fornecendo soluções com integração simples completa, desde os sensores responsáveis pela aquisição de dados até o gerenciamento e aplicação de regras sobre estes dados.

## 2 Internet of Things

O termo *Internet of Things* foi usado pela primeira vez pelos fundadores do *MIT Auto-ID Center*, mencionado especialmente pelo britânico Kevin Ashton no ano de 1999, referindo-se especificamente a área de Gerenciamento da Cadeia de Suprimentos [1]. Com o passar do tempo, as aplicações utilizando esse conceito se ampliaram, sendo aplicadas em áreas como transporte, cuidados com a saúde, automação residencial, entre outras. Devido a essa evolução no paradigma, a definição de *Things* foi se tornando cada vez mais abrangente, representando desde implantes de monitoramento cardíacos e transponders para identificação animal até automóveis com sensores integrados, sensores para análise de luminosidade e temperatura.

A *Internet of Things* consiste basicamente em uma rede de objetos físicos (*Things*) que fornecem informações específicas de seu contexto (*little data*), gerando assim, uma enorme quantidade de dados desconexos (*big data*) que precisam ser armazenados, processados e apresentados de uma forma eficiente e de fácil interpretação. O grande valor encontrado no conceito de *Internet of Things* é relação das informações produzida por essa rede de dispositivos.

### 2.1 Aplicações

Com a facilitação do acesso e integração de uma grande variedade de dispositivos heterogêneos, uma série de áreas distintas começou a obter vantagens do paradigma, aplicando interpretações dentro de seu contexto das informações recebidas para criar soluções para seus respectivos problemas.

#### 2.1.1 Smart City

Além das barreiras técnicas, a adoção do paradigma de *Internet of Things* também é prejudicado pela dificuldade na criação de um modelo de negócio claro e amplamente aceito para atrair investidores a financiar o desenvolvimento destas tecnologias [2]. Neste cenário complexo, a aplicação do conceito de *Internet of Things* em um ambiente urbano pode se tornar particularmente interessante por atender à forte intenção de muitos governos de adotar soluções TIC (Tecnologias da Informação e Comunicação) para administração pública.

Apesar de não haver uma definição formal amplamente aceita de *Smart City*, a intenção final é de alcançar uma melhor utilização dos recursos públicos, oferecendo uma melhor qualidade nos serviços oferecidos aos cidadãos ao mesmo tempo em que reduz os custos operacionais da administração pública [3]. Este objetivo pode ser atingido pela utilização de um *IoT* urbano, como por exemplo uma infraestrutura para controlar a utilização de iluminação pública baseada no clima ou sensores em latas de lixo para identificar a melhor rota para seu recolhimento com base em sua utilização. Grandes empresas como a IBM e a GE tem investido muito nessa área, criando pilotos bem sucedidos de *Smart City*.

#### 2.1.2 Automação Residencial

Quando o conceito de uma casa automatizada foi criada 80 anos atrás, muitas limitações técnicas, que perduraram até pouco tempo, transformaram esse conceito em um sonho inatingível. Com a evolução da internet e a popularização das tecnologias de conexões sem fio, os dispositivos conectados continuamente à rede em uma casa possibilitaram trazer essa antiga ideia à realidade. Este conceito, também conhecido como *Smart Home*, está gradualmente se definindo em relação às tecnologias e aplicações a serem utilizadas [4].

Atualmente, existem centenas de grandes empresas investindo neste segmento, criando dispositivos e softwares para otimizar o uso das moradias com soluções para economia de energia elétrica, segurança através de monitoramento e conforto. Grandes empresas como Samsung e Apple tem feito vastos investimentos para facilitar a integração de seus dispositivos com o ecossistema de automação residencial criado em cima do conceito de *Internet of Things*.

### 2.1.3 Sistemas Médicos e de Saúde Pessoal

Uma das áreas de aplicação mais proeminente é a relacionada à saúde. Atualmente existem muitas aplicações de IoT com o intuito de melhorar a saúde pessoal baseadas em dados como exercícios físicos realizados, análise da qualidade do sono e alimentação. Grandes empresas como a Apple estão investindo fortemente nessa área e tornando-a cada vez mais fértil com a criação de novos sensores e interligação de dados.

Ainda dentro da área da saúde, existem aplicações com foco maior na área médica, como, por exemplo, avaliação de pacientes a distância, possibilitando um atendimento mais rápido e específico baseados em uma série de informações provenientes de sensores com finalidades distintas como pressão sangüínea, frequência cardíaca, temperatura, entre outros. Essas aplicações representam um grande avanço, tendo em vista que pela primeira vez, é possível acessar informações precisas em tempo real, além de possibilitar a montagem de grupos de teste com uma facilidade sem precedentes. Essa área se tornou tão proeminente que possui uma terminologia própria, *IoMT* (*Internet of Medical Things*).

## 2.2 Plataformas e Ambientes para IoT

### 2.2.1 mbed

A empresa ARM provê uma solução completa de *Internet of Things* (IoT) chamada mbed[5]. Esta plataforma é dividida em três módulos, mbed OS, mbed Device Server e mbed Tools. A plataforma utiliza em seu sistema embarcado a tecnologia mbedTLS para criptografia de dados com baixo consumo de memória, desenvolvida inicialmente sob o nome de PolarSSL pela empresa holandesa Offspark, que foi comprada recentemente pela ARM.

Além de prover uma segurança padrão fim-a-fim entre apps e dispositivos IoT, uma das grandes preocupações da plataforma mbed é tratar o consumo de energia como uma das principais restrições.

#### 2.2.1.1 mbed OS

É um sistema operacional, disponibilizado de graça, para processadores da linha ARM Cortex-M, que são desenvolvidos visando a eficiência de energia e produtividade.

A arquitetura do mbed OS fornece componentes de software e um framework de aplicação que, combinados com a grande diversidade de empresas e desenvolvedores que disponibilizam bibliotecas e drivers, facilitam e agilizam o processo de desenvolvimento de aplicações para IoT.

O mbed OS provê suporte aos padrões como Bluetooth Smart®, Cellular, Thread, Wi-Fi®, e 802.15.4/6LoWPAN junto com TLS/DTLS, CoAP, HTTP, MQTT e Lightweight M2M

### 2.2.1.2 mbed Device Server

É um produto que possui licença paga e tem como sua funcionalidade principal permitir o gerenciamento e a conexão dos dispositivos de uma forma segura.

Provê a ligação entre os protocolos utilizados nos dispositivos IoT e a API que é utilizada por desenvolvedores web. Isto simplifica a integração de dispositivos IoT que geram ‘little data’ que vão para os servidores que analisam e agregam a informação gerando a ‘big data’.

O Device Server é escalável, podendo conectar e gerenciar milhões de dispositivos.

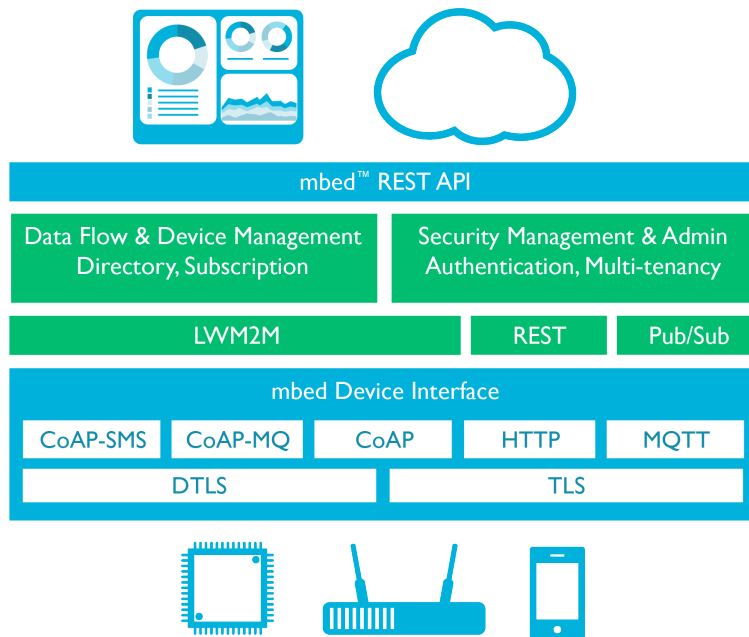


Figura 1: Arquitetura do mbed Device Server.

### 2.2.2 FlowCloud

A tecnologia *FlowCloud* provê uma plataforma segura e independente de aplicações para possibilitar a rápida construção, gerenciamento e implantação de serviços digitais. Foi desenvolvida pela empresa Imagination Technologies, que visa com esta solução, suprir as necessidades dos conceitos de *Internet of Things* e *Machine-to-Machine* (M2M) provendo meios de conectar dispositivos que estão separados geograficamente, estabelecendo uma conexão ponto a ponto (P2P) entre eles. A solução visa tratar desde pequenas aplicações de monitoramento em tempo real até grandes serviços com milhares de usuários [6].

Segundo as especificações descritas pela empresa criadora, as principais funcionalidades do *FlowCloud* incluem a capacidade de gerenciamento de dispositivos e usuários, serviço de troca de mensagem assíncrona, registro de eventos, troca de dados de alta segurança e pagamentos eletrônicos. Do ponto de vista analítico, é fornecida uma série de ferramentas para permitir ao administrador do serviço a criação de relatórios com utilização de dados em tempo real, capacitando o monitoramento de todas as interações dos usuários e dispositivos registrados no serviço.

### **2.2.3 Arrayent**

Arrayent [7] é uma plataforma IoT que oferece conexão e segurança entre dispositivos IoT e aplicações para smartphones, tablets e web com baixo custo e simplicidade. A plataforma é composta de 4 componentes: Arrayent Connect Agent, Arreyent Connect Cloud, Arrayent Mobile Framework e Arrayent Cloud Insight.

#### **2.2.3.1 Arrayent Connect Agent**

É utilizado por desenvolvedores de software embarcado. Funciona como um módulo do firmware que gerencia a seção do dispositivo com junto ao Arreyent Connect Cloud e abstrai essa responsabilidade para o desenvolvedor do firmware, deixando-o livre para se preocupar em desenvolver apenas as funcionalidades necessárias para o funcionamento do dispositivo.

Atualmente a plataforma suporta Wi-Fi, ZigBee, Z-Wave LAN e os sistemas operacionais Linux ou FreeRTOS.

#### **2.2.3.2 Arrayent Connect Cloud**

O Arrayent Connect Cloud é o coração da plataforma de IoT da Arrayent. Ele é essencialmente um sistema operacional baseado na cloud que agrega os dispositivos virtualizados, que são uma cópia dos dispositivos físicos, para que as aplicações web e mobile possam se conectar e recolher informações ou executar comandos sobre os dispositivos. Além disso, o Arreyent Connect Cloud pode emitir alertas, fazer update de firmware ‘over-the-air’, gerenciar contas de usuários e muito mais.

#### **2.2.3.3 Arreyent Mobile Framework**

O Arrayent Mobile Framework ajuda os desenvolvedores mobile a criar rapidamente aplicativos intuitivos e confiáveis para o mercado. Ele abstrai a complexidade envolvida na utilização da plataforma M2M da Arrayent, que é uma API web de baixo nível, deixando o desenvolvedor livre para se preocupar apenas com as funcionalidades e beleza dos aplicativos por ele desenvolvidos.

#### **2.2.3.4 Arrayent Cloud Insight**

O Arrayent Cloud Insight é responsável pela camada de negócio, fornecendo serviços de Business Intelligence, gerando relatórios de dados comuns a todos os seus produtos como localização do dispositivo, interações entre os dispositivos e as aplicações, tendência de pico de utilização e muito mais. O Arrayent Cloud Insight agrega, normaliza, filtra e entrega os dados dos dispositivos para qualquer solução de análise que você utilizar.

### **2.2.4 Samsung IoT Platform**

A plataforma IoT da Samsung provê uma conexão padronizada de diversos dispositivos a um dispositivo ‘smart’ ou hub.

A Samsung disponibiliza um SDK que simplifica o desenvolvimento de novos serviços, coletando e processando os dados de sensores e serviços que o usuário possa ter. Os desenvolvedores não precisam analisar ou fazer setup da infraestrutura complexa de conexão entre os dispositivos ou se preocupar com os dados do usuário na cloud.



A plataforma de IoT da Samsung foi desenhada para que os desenvolvedores se concentrem apenas nas funcionalidades dos serviços que estão sendo desenvolvidos por eles. Uma limitação do SDK da Samsung é que ele é disponibilizado apenas para a plataforma Android 4.1 ou superior.

### 2.2.5 IBM Internet of Things Foundation

A IBM desenvolveu a *IBM Internet of Things Foundation*, que consiste em um sistema na nuvem que torna simples extrair valor dos dispositivos *Internet of Things*. Combinada com a plataforma IBM Bluemix™, é possível acessar dispositivos e dados rapidamente, bem como compor aplicações de análise, aplicativos móveis e até um painel de visualizações.

A *IBM Internet of Things Foundation* provê amplo suporte aos seus usuários disponibilizando diversos manuais e tutoriais que explicam desde o que é a *Internet of Things* até a documentação detalhada das APIs.

#### 2.2.5.1 IBM Bluemix

O *IBM Bluemix* é uma solução de nuvem da IBM que permite aos desenvolvedores a criação e gerenciamento de aplicativos na nuvem de uma maneira ágil e fácil. O BlueMix é uma implementação da Arquitetura de Nuvem Aberta da IBM baseada em Cloud Foundry, uma plataforma como serviço (PaaS) de código aberto.

### 2.2.6 SmartSantander

O *SmartSantander* propõe uma instalação de pesquisa única no mundo, em uma escala de uma cidade, para fomentar o desenvolvimento de serviços e aplicações para uma *Smart City*. Os resultados de pesquisas efetuadas nesse ambiente devem influenciar as definições e especificações da arquitetura da Internet do Futuro do ponto de vista dos serviços de *Internet of Things* [8].

O SmartSantander irá possuir mais de 20.000 sensores para IoT em um cenário urbano. O projeto pretende promover os seguintes resultados [9].

1. Validação das abordagens de modelos arquiteturais da *Internet of Things*.
2. Identificação dos pontos chave para criação de IoT, particularmente os protocolos e mecanismos de gerenciamento e interação; tecnologia de dispositivos e serviços como localização, identificação, gerenciamento e segurança.
3. Identificação da aceitação das tecnologias e serviços de IoT.

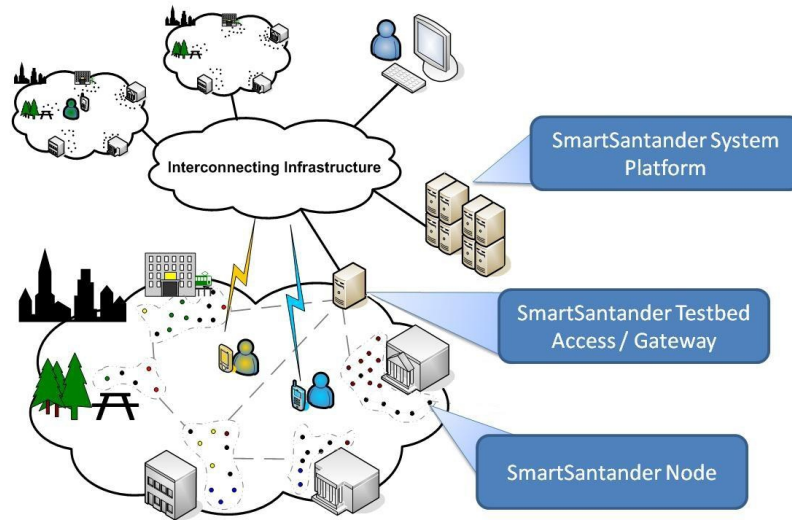


Figura 2: Funcionamento da infraestrutura do projeto SmartSantander em alto nível.

As instalações experimentais propostas pelo *SmartSantander* precisam atender uma série de demandas de diferentes *stakeholders*, desde desenvolvedores de soluções e provedores de serviços IoT até os usuários finais. Vários requisitos funcionais e não-funcionais foram levantados no princípio do projeto, levando em consideração os variados pontos de vista e necessidades dos *stakeholders* envolvidos. Os resultados foram divididos em quatro aspectos, criando o modelo arquitetura com os subsistemas descritos a seguir [8]:

#### 2.2.6.1 Authentication, Authorisation and Accounting (AAA)

Implementa funções que controlam e auditam o acesso à todas as funcionalidades do sistema. Possui processos comuns entre todos os tipos de usuários da instalação. Representa tipicamente o primeiro ponto de contato do usuário para que esse tenha acesso às demais funcionalidades do ambiente de testes, o *Testbed*. Além das funcionalidades de gerenciamento das contas de usuários no ambiente de teste, as funções do AAA garante que o usuário está propriamente autenticado e que possui a autorização necessária para acessar o ambiente de teste de acordo com suas permissões.

#### 2.2.6.2 Testbed Management

O *Testbed Management* engloba as funções que são relevantes para a operação do ambiente de teste. Este subsistema provê uma série de funcionalidades, como por exemplo configuração *plug and play* de novos componentes do *testbed*, monitoramento e avaliação de performance.

#### 2.2.6.3 Experimental Support

O *Experimental Support* provê todas as funcionalidades relativas ao ciclo de vida dos usuários do ambiente de teste. As funcionalidades são geralmente utilizadas por pesquisadores, mas também são utilizadas por provedores de sistemas, já que um sistema pode ser visto como um experimento de longa duração. Suas funcionalidades incluem funções para verificar a

ocupação dos recursos no ambiente de testes e suas características, mecanismos para configurar os experimentos, controle dos experimentos e análise de dados.

#### **2.2.6.4 Application Support**

O *Application Support* provê funcionalidades para serem utilizadas por desenvolvedores de aplicações e provedores de serviços para compor um *Smart Service* rodando nas instalações do *SmartSantander*. Estas são ferramentas para acessar facilmente informações de sensores e gerenciar dados.

#### **2.2.7 FIWARE**

O FIWARE é uma iniciativa para promover um ecossistema que capture as oportunidades que surgirão com a nova onda de digitalização causada pela integração das novas tecnologias relacionadas à Internet. Esse projeto é baseado nos cinco módulos descritos a seguir.

##### **2.2.7.1 FIWARE**

A plataforma FIWARE provê um simples porém poderoso conjunto de APIs que facilitam o desenvolvimento de *Smart Applications* em uma série de setores. As APIs disponibilizadas possuem uma especificação pública e sem necessidade de pagamento de royalties. Além das APIs, cada componente do FIWARE possui uma implementação como referência com código de fonte aberto e distribuída publicamente, possibilitando que mais provedores de solução entrem no mercado de maneira rápida e com baixo custo.

##### **2.2.7.2 FIWARE Lab**

Experimentos realizados em ambiente real aceleram o desenvolvimento das tecnologias da Internet do Futuro e seus serviços, qualificam o mercado de infraestruturas *Smart*, para possibilitar isso, o *FIWARE Lab* foi criado. O *FIWARE Lab* é um ambiente de teste não comercial criado para que a experimentação e inovação baseada nas tecnologias FIWARE sejam fomentadas. Investidores, desenvolvedores e demais interessados podem testar a tecnologia e suas aplicações neste ambiente, se utilizando de *Open Data* publicada por cidades e organizações. O *FIWARE Lab* é aplicado sobre uma rede geograficamente distribuída de nodos interligados criando uma grande infraestrutura experimental [10].

##### **2.2.7.3 FIWARE Ops**

O *FIWARE Ops* é uma coleção de ferramentas para facilitar a implantação, configuração e operação das instâncias de FIWARE. Este conjunto de ferramentas foi desenvolvido para ajudar na expansão da infraestrutura associada a uma instância de FIWARE adicionando novos nodos com o passar do tempo e permitindo a cooperação de múltiplos provedores de plataformas. Em resumo, o *FIWARE Ops* é a ferramenta utilizada para montar, operar e expandir o *FIWARE Lab*.

##### **2.2.7.4 FIWARE Accelerate**

O *FIWARE Acceleration Programme* busca promover a adoção das tecnologias FIWARE por parte dos integradores de soluções e desenvolvedores de aplicações com foco especial nos espe-

cialistas e start-ups. A União Europeia iniciou uma campanha ambiciosa em setembro de 2014, financiando com 80 milhões de euros especialistas e empreendedores que tenham intenção de desenvolver aplicações inovadoras baseadas na tecnologia FIWARE.

#### **2.2.7.5 FIWARE Mundus**

Mesmo tendo surgido na Europa, o FIWARE foi desenvolvido com uma ambição de adoção global. O *FIWARE Mundus* é um programa criado para para atingir esse objetivo, envolvendo investidores e interessados de outras regiões do mundo para participarem no desenvolvimento e utilização das tecnologias.

### 3 Cooperative Middleware Platform as a Service

Cooperative Middleware Platform as a Service (COMPaaS) é um sistema que provê aos usuários uma simples e bem definida plataforma de serviços para o desenvolvimento de aplicações de *Internet of Things*.

O COMPaaS amplia as especificações do EPCglobal em relação às interfaces de middleware RFID para prover serviços de alto nível. Além disso, provê também uma leve arquitetura baseada nas especificações ETSI para plataforma de serviços M2M, assim como uma aplicação web para integração de dispositivos físicos. O intuito principal desta plataforma se estende desde o gerenciamento de dados até a integração dos dispositivos, provendo uma serviços cooperativos e de alto nível para as aplicações [11].

Ele é composto por 3 subsistemas, a saber: Application Level Events Interface (ALE), Middleware Level Events Interface (MLE) e Device Level Protocol (DLE).

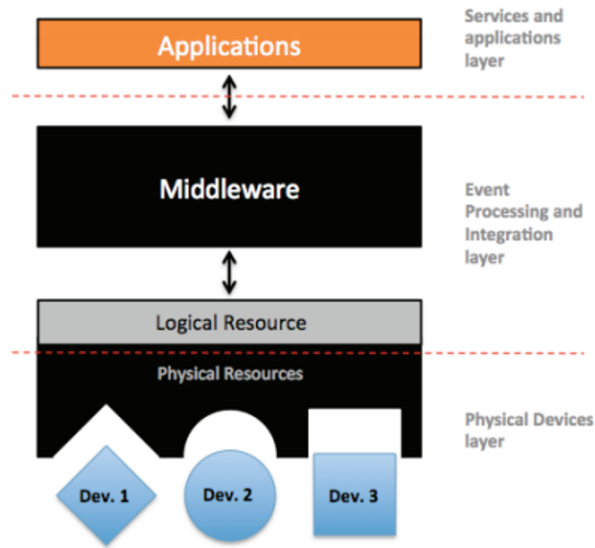


Figura 3: Arquitetura em alto nível do COMPaaS.

#### 3.1 Application Level Events Interface

O *Application Level Events* (ALE) expõe uma interface SOAP para a comunicação com as aplicações. Ele é responsável por receber as requisições das aplicações e encaminha-las ao MLE. As informações requisitadas são recebidas do MLE por meio de um WebSocket.

#### 3.2 Middleware Level Events Interface

O MLE expõe uma interface RESTful para a comunicação com o ALE. É responsável pelo recebimento das requisições oriundas do ALE para então encaminhar as mesma para o DLE. Os dados retornados do DLE, no formato de DataMessage object, são então enviados ao ALE por meio de WebSocket.

### **3.3 Device Level Events**

O DLE cria um dispositivo virtual em torno do dispositivo físico e é responsável por definir o formato do DataMessage object e expor as funcionalidades que são realizadas pelo dispositivo físico. O DLE é o responsável por saber das minúcias de cada dispositivo, então, para cada novo dispositivo que se deseja incluir na IoT, é necessário criar um DLE específico para aquele dispositivo. Isto é um pouco oneroso e será um dos temas abordados neste trabalho.

## 4 HellfireOS

O HellfireOS é um sistema operacional de tempo real preemptivo que é parte constituinte do Hellfire System (HFS). Projetado para sistemas MPSoC, possui gerenciamento dinâmico de tarefas que conta com a proteção contra inversão de prioridades. Seu escalonador possui as seguintes políticas de escalonamento: Rate Monotonic, Round Robin, Earliest Deadline First e Deadline Monotonic.

Possui mapeamento dinâmico de tarefas, através das primitivas para migração de tarefas. A migração pode ser implícita ou explícita. Na migração explícita, o designer insere pontos de migração no código da aplicação. Já a migração implícita é feita por um mecanismo automatizado que monitora a carga do sistema e envia ordem de migração para as tarefas. Geralmente, os benefícios da migração de tarefas são o balanceamento de carga nos processadores, decréscimo no consumo de energia e suporte a técnicas de tolerância a falhas.

HellfireOS é um micro-kernel portátil composto de vários módulos e pode ser configurado de diversas maneiras, ora visando o número máximo de tarefas em um processador, ora definindo o tamanho da pilha da tarefa. A ideia é permitir ao designer otimizar o tamanho final da imagem do kernel de acordo com as restrições da aplicação e de sistema. Permitindo assim que o designer permita ao sistema rodar em plataformas com baixa memória, como é o caso dos típicos sistemas embarcados críticos.

Dentre as bibliotecas mais relevantes ao desenvolvimento da integração abordada na seção 5, consta uma LibC customizada e a biblioteca matemática (math.h) com emulação de número de ponto flutuante. Outro ponto de interesse, nesse caso para a prova de conceito, é observarmos que o HellfireOS não possui todas as camadas de rede implementadas, forçando a criação ou das camadas faltantes, ou de outra solução, para possibilitar a integração.

A API do hellfire é dividida em 5 (cinco) grupos: Manipulação de Tarefas, Exclusão Mútua, Manipulação de Memória, Comunicação entre Processos e LibC.

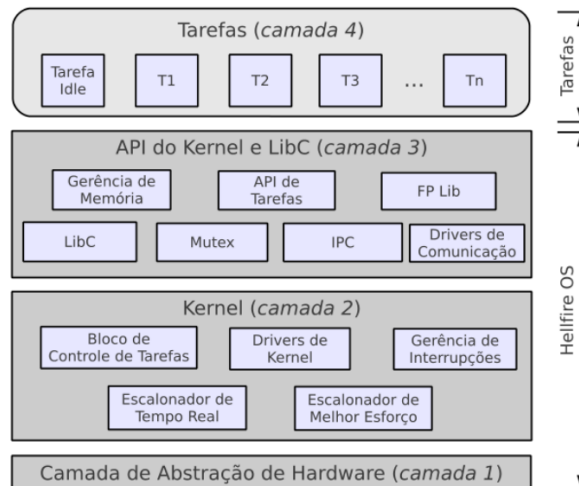


Figura 4: Arquitetura em alto nível do HellfireOS.

O Hellfire System conta também com uma plataforma para design que compreende diferentes níveis de abstração, partindo de aplicações escritas em C, até a prototipação em FPGAs. Isto é possível graças as ferramentas e módulos que fazem parte do Hellfire Framework (Hell-

fireFW).

O Hellfire Framwork apresenta um fluxo para o design de MPSoC baseado em sistemas embarcados do tipo crítico como equipamentos uso médicos, dispositivos de segurança e sistemas de aeronaves. Este tipo de sistemas apresenta uma necessidade específica de design, englobando preocupações especiais em relação à consumo de energia e alta performance mantendo o processamento de tempo real [12].



## 5 Hellfire and COMPaaS Integration Layer (HAC)

Atualmente, a plataforma COMPaaS possui a necessidade de criação de uma abstração para cada dispositivo que pretende se comunicar à ela, tornando o processo de adoção do COMPaaS oneroso neste aspecto. Para eliminar esta necessidade, propõe-se a adaptação do sistema HellfireOS, que já possui a infraestrutura necessária para suportar múltiplos tipos de sensores, para que suporte a comunicação com a plataforma COMPaaS de forma praticamente transparente do ponto de vista do suporte de hardware dos sensores ligados ao sistema, permitindo que os produtos que adotem o HellfireOS como seu sistema embarcado estejam aptos a se integrarem à solução IoT de forma mais rápida e simples.

Para possibilitar essa comunicação, será implementada uma camada de integração entre o sistema de tempo real, previamente descrito na seção 4, HellfireOS e a plataforma COMPaaS, também analisada anteriormente na seção 3. Essa camada consistirá em adicionar suporte a duas questões críticas no HellfireOS, uma biblioteca compatível com o protocolo de comunicação esperado pela camada de mais baixo nível do COMPaaS e, para prova de conceito, uma meio de comunicação entre o HellfireOS e a plataforma IoT, tendo em vista que o sistema operacional ainda não possui uma implementação de todas as camadas de rede necessárias, como por exemplo, o protocolo TCP.

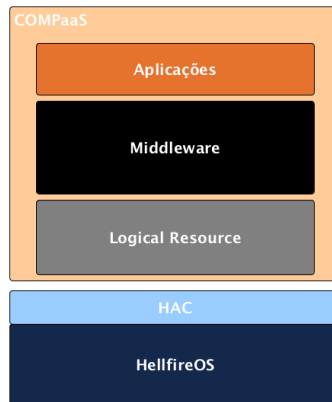


Figura 5: Arquitetura em alto nível da integração com a camada de integração.

Para fins de validação da camada de integração, será criado um protótipo utilizando uma placa CubieBoard rodando um sistema Linux que rodará o sistema HellfireOS virtualizado. O meio de comunicação entre o host Linux e o HellfireOS será feito através de um espaço de memória compartilhada, que será abstraída da camada de integração desenvolvida, permitindo que a mesma camada seja utilizada com poucas ou nenhuma alteração quando o HellfireOS possuir outros meios de comunicação via rede. Para possibilitar que essa comunicação entre host e HellfireOS funcione, será desenvolvida uma daemon no sistema host para interfacear sua camada de comunicação via rede com a comunicação via memória compartilhada com o sistema de tempo real.

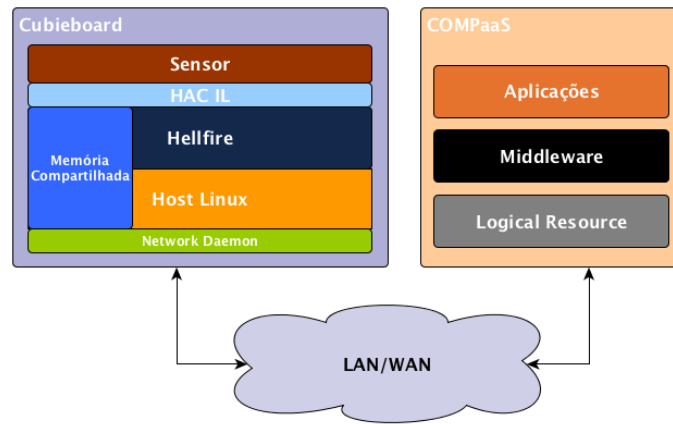


Figura 6: Arquitetura do protótipo proposto para prova de conceito.

## 5.1 Prova de Conceito

Para provar o funcionamento da camada de integração, será utilizada uma simulação de sensor médico baseado em funções matemáticas suportadas pelo HellfireOS ou mesmo um sensor real, caso haja o suporte Cubieboard, HellfireOS e sensor disponível.

## 6 Proposta

### 6.1 Proposta TCC 1

#### 6.1.1 Objetivos

Esta primeira etapa do trabalho de conclusão possui como objetivos, em primeiro lugar, elevar o conhecimento nas áreas de sistemas embarcados. Baseado no conhecimento de pesquisa feita no decorrer deste trabalho, será criado o ambiente necessário para efetuar a prova de conceito, obtendo o hardware necessário, e fazendo as instalações necessárias de sistemas operacionais, host Linux e o HellfireOS, este de forma virtualizada. Também serão iniciadas as medidas necessárias para suprir a falta de das camadas de protocolos de rede necessárias, implementando uma comunicação através de memória compartilhada entre os dois sistemas e, inclusive, uma deamon para interfacear o espaço de memória compartilhada no host Linux e a rede.

#### 6.1.2 Cronograma

Data	Evento
22.04.2015	Criação do ambiente da Cubieboard com o host Linux.
04.04.2015	Preparação do ambiente com o suporte e configurações de virtualização necessárias.
20.05.2015	Instalação do HellfireOS de forma virtualizada rodando sobre o host Linux.
21.05.2015	Início da implementação da comunicação entre host e HellfireOS através de memória compartilhada.
22.06.2015	Entrega do Volume Final de TC1.

#### 6.1.3 Lista de Atividades

- Conseguir o hardware necessário para as implementações.
- Escolha da distribuição adequada para rodar no hardware.
- Instalação do host no hardware.
- Pesquisa e configuração do suporte de virtualização no host.
- Pesquisa e instalação do HellfireOS como sistema virtualizado.
- Implementação da comunicação entre host e HellfireOS.

### 6.2 Proposta TCC 2

#### 6.2.1 Objetivos

Ao final do projeto, objetiva-se permitir que o HellfireOS se integre à plataforma COMPaaS, permitindo que os produtos que possuam o HellfireOS como seu sistema embarcado tenham uma integração mais rápida e simples de seus dispositivos sensores à plataforma IoT.

Nesta etapa final, serão finalizados os ajustes necessários do ambiente proposto para teste para então implementar a integração da plataforma COMPaaS com o sistema HellfireOS, descritos nas seções 3 e 4 respectivamente, com a criação do *Hellfire and COMPaaS Integration Layer* (HAC), descrito na seção 5, objetivando a criação de uma solução completa, desde o sistema operacional embarcado responsável pela gerência de múltiplos sensores, até a plataforma de IoT, possibilitando um uso de funcionalidade robusta e simples do HellfireOS na criação e aplicação de equipamentos integráveis no conceito de *Internet of Things*. Para prova e testes desta integração, será criado um simulador de sensor da área médica, baseada em estudos efetuados nesta segunda etapa do trabalho, possibilitando um teste da integração de forma completa, passando por todos os níveis descritos nas arquiteturas tanto da plataforma COMPaaS como no HellfireOS nas seções anteriores.

### 6.2.2 Cronograma

Data	Evento
07.08.2015	Implementação da comunicação entre host e HellfireOS através de memória compartilhada.
21.08.2015	Criação da daemon para interfacear a memória compartilhada do host e a rede.
09.10.2015	Término do desenvolvimento do HAC Integration Layer no HellfireOS.
06.11.2015	Criação do simulador de sensor para prova de conceito da integração.
13.11.2015	Teste completo da implementação.
22.11.2015	Entrega do Volume Final.
09.12.2015	Apresentação do Trabalho.

### 6.2.3 Lista de Atividades

- Testes para prova da implementação da comunicação entre host e HellfireOS.
- Definição do funcionamento da daemon proposta.
- Testes da implementação da daemon de comunicação entre memória compartilhada e rede.
- Testes do funcionamento da implementação do HAC Integration Layer.
- Definição das funcionalidades do sensor a ser simulado.
- Teste para prova da implementação da integração.

## Referências

- [1] K. Ashton, “That ‘internet of things’ thing,” *RFiD Journal*, 2009.
- [2] A. Laya, V.-l. Bratu, and J. Markendahl, “Who is investing in machine-to-machine communications?,” 24th European Regional ITS Conference, Florence 2013 88475, International Telecommunications Society (ITS), 2013.
- [3] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, “Internet of things for smart cities,” *Internet of Things Journal, IEEE*, vol. 1, Feb 2014.
- [4] Y. Song, B. Han, X. Zhang, and D. Yang, “Modeling and simulation of smart home scenarios based on internet of things,” in *Network Infrastructure and Digital Content (IC-NIDC), 2012 3rd IEEE International Conference on*, pp. 596–600, Sept 2012.
- [5] “mbed.” <http://mbed.com>. Acessado: 04/04/2015.
- [6] “Flowcloud.” <http://www.imgtec.com/flowcloud/>. Acessado: 05/04/2015.
- [7] “Arrayent.” [www.arrayent.com](http://www.arrayent.com). Acessado: 05/04/2015.
- [8] L. Sanchez, J. A. Galache, V. Gutierrez, J. M. Hernandez, J. Bernat, A. Gluhak, and T. Garcia, “SmartSantander: The meeting point between Future Internet research and experimentation and the smart cities,” in *Future Network & Mobile Summit (FutureNetw), 2011*, pp. 1–8, IEEE, June 2011.
- [9] “Smartsantander.” <http://http://www.smartsantander.eu>. Acessado: 10/04/2015.
- [10] T. Zahariadis, A. Papadakis, F. Alvarez, J. Gonzalez, F. Lopez, F. Facca, and Y. Al-Hazmi, “Fiware lab: Managing resources and services in a cloud federation supporting future internet applications,” in *Utility and Cloud Computing (UCC), 2014 IEEE/ACM 7th International Conference on*, pp. 792–799, Dec 2014.
- [11] L. A. Albernaz, R. T. Tiburski, E. de Matos, and F. Hessel, “Cooperative middleware platform as a service for internet of things applications,” *ACM*, 2015.
- [12] A. Aguiar, S. Filho, F. Magalhaes, T. Casagrande, and F. Hessel, “Hellfire: A design framework for critical embedded systems’ applications,” in *Quality Electronic Design (ISQED), 2010 11th International Symposium on*, pp. 730–737, March 2010.