

UNIVERSIDADE FEDERAL DE SÃO CARLOS

CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

**MIDDLEWARE PARA ARQUITETURAS DE IOT
DE CONTROLE DINÂMICO DE DISPOSITIVOS
CONECTADOS, BASEADO EM SUA
LOCALIZAÇÃO E CONTEXTO**

LEONARDO ALMEIDA SILVA FERREIRA

ORIENTADOR: PROF. DR. HELIO CRISTANA GUARDIA

São Carlos – SP

setembro/2017

UNIVERSIDADE FEDERAL DE SÃO CARLOS

CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

**MIDDLEWARE PARA ARQUITETURAS DE IOT
DE CONTROLE DINÂMICO DE DISPOSITIVOS
CONECTADOS, BASEADO EM SUA
LOCALIZAÇÃO E CONTEXTO**

LEONARDO ALMEIDA SILVA FERREIRA

Qualificação apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de São Carlos, como parte dos requisitos para a obtenção do título de Mestre em Ciência da Computação, área de concentração: Sistemas Distribuídos

Orientador: Prof. Dr. Helio Cristana Guardia

São Carlos – SP

setembro/2017

RESUMO

Nonono nonono nonono, nonono, nonono nonono nonono nononono nonno. Nonono nonono nonono, nonono, nonono nonono nonono nononono nonno. Nonono nonono nonono, nonono, nonono nonono nonono nononono nonno. Nonono nonono nonono, nonono, nonono nonono nonono nononono nonno. Nonono nonono nonono, nonono, nonono nonono nonono nononono nonno. Nonono nonono nonono, nonono, nonono nonono nonono nononono nonno. Nonono nonono nonono, nonono, nonono nonono nonono nononono nonno.

Palavras-chave: Computação Móvel, Comunicação sem Fio, Comunicação Oportunística

ABSTRACT

Nonono nonono nonono, nonono, nonono nonono nonono nononono nonno. Nonono nonono nonono, nonono, nonono nonono nonono nononono nonno. Nonono nonono nonono, nonono, nonono nonono nonono nonono nononono nonno. Nonono nonono nonono, nonono, nonono nonono nonono nononono nonno. Nonono nonono nonono, nonono, nonono nonono nonono nononono nonno. Nonono nonono nonono, nonono, nonono nonono nonono nononono nonno. Nonono nonono nonono, nonono, nonono nonono nonono nononono nonno.

Keywords: Computação Móvel, Comunicação sem Fio, Comunicação Oportunistica

LISTA DE FIGURAS

2.1	Visão física de uma rede de sensores	20
3.1	Visão orientada aos serviços dentro de uma rede de sensores	23
3.2	Arquitetura alto nível GSN	24
4.1	Arquitetura OpenIoT	26
4.2	Arquitetura OpenIoT	29
4.3	Arquitetura OpenIoT	29
4.4	Arquitetura OpenIoT	32
4.5	Esquema de um Descritor de serviço	33
4.6	Esquema de um Sensor (o esquema OSMO está incluso no OSDSpec)	34
4.7	Esquema de um tipo de sensor	35
4.8	Esquema de um identificador	35
4.9	Esquema de um identificador	36
5.1	Concepção da arquitetura dividida em 3 partes	38
5.2	Arquitetura CA4IOT em nível de componentes	41
5.3	Arquitetura CA4IOT	42
5.4	Arquitetura CA4IOT	42
5.5	Arquitetura CA4IOT	43
5.6	Arquitetura CA4IOT	43
5.7	Arquitetura CA4IOT	44
6.1	Arquitetura do OpenIoT Full extendida	48

6.2	Arquitetura do OpenIoT Light extendida	48
6.3	Diagrama de Stados do Scheduler extendido	49
6.4	Relacionamento de dados extendido	49
6.5	Diagrama de sequência 1	50
6.6	Diagrama de sequência 2	51
6.7	Exemplo de Contexto	52
6.8	Representação do ciclo do Conhecimento	52
6.9	Ciclo de Vida dos Objetos Virtuais	53
6.10	Arquitetura 2	53

LISTA DE TABELAS

SUMÁRIO

I Definições

CAPÍTULO 1 – OBJETO DA PESQUISA	12
1.1 Contextualização	12
Middleware	13
Plataforma	13
Arquitetura	14
Modelo	14
Objetos Virtuais	14
1.2 Objetivos	14
1.3 Metodologia	15
1.4 Organização do Trabalho	16

II Temas Relacionados 17

CAPÍTULO 2 – IOT E SEMÂNTICA	18
2.1 Internet das Coisas (IoT)	18
2.2 Redes de Sensores Sem Fio	19
2.3 IoT semântico	20
CAPÍTULO 3 – ARQUITETURA GSN	22
3.1 Objetos Virtuais	23

3.2	Processamentos dos streams de dados	23
3.3	Arquitetura e Implementação	23
3.4	Middleware de Extensão XGSN	23
	Abordagem geral	24
	Aspectos de gerenciamento de ontologia	24
	Arquitetura	24
	Exemplos	24
CAPÍTULO 4 – PLATAFORMA OPENIOT		26
4.1	Aplicação	27
	Definições	27
	Apresentação	27
	Configuração e Monitoramento	27
4.2	Virtualização	27
4.2.1	Scheduler	27
	Seleção de Sensores e Wrappers	27
	Ativação "indireta" Sensores Virtuais	28
	Armazenamento e execução de solicitações	28
	Controle do estado das solicitações	28
	Controle de acesso	28
4.2.2	Serviços de entrega e gestão de aplicativos	28
4.2.3	Plataforma de relacionamento dos dados - LSM-Light	30
4.3	Física	30
4.3.1	Middleware de controle de sensores (XGSN)	30
4.3.2	Controle dinâmico de Sensores	31
4.4	Principais objetos na plataforma OpenIoT	31
4.4.1	OSDSpec	31

4.4.2	OSMO	31
4.4.3	SensorTypes	31
4.4.4	DescriptiveIDs	32
4.4.5	SdumServiceResultSet	32
CAPÍTULO 5 – ARQUITETURA CA4IOT		37
5.0.1	Capacidades funcionais da plataforma	37
	Conectar sensores ao middleware IoT facilmente	38
	Entender e manter informações de contexto	38
	Extrair informações de contexto de alto nível	38
	Entender o problema do usuário	39
	Preencher a lacuna entre usuário e sensores	39
	Gerenciar usuários	39
5.0.2	Divisão em Camadas	39
	Camada de usuário	39
	Dados, semântica e definição de contextos	39
	Lógica e processamento de dados	40
	Descoberta de semântica e contextos	40
	Aquisição dos dados de sensores	40
	Camada de detecção	40
5.0.3	Componentes e Elementos da arquitetura	40
	Componentes	41
5.0.4	Middleware CASCoM	41
III Proposta de trabalho		46
CAPÍTULO 6 – PROPOSTA		47

6.1	Arquitetura Proposta	47
6.1.1	Descrição	47
6.2	Modelo Utilizado	47
6.2.1	Modelo de conhecimento	47
6.2.2	Definição dos Objetos	47
6.2.3	Estrutura em Grafo	47
6.2.4	Processamento dos dados	47
6.2.5	Processamento dos dados	47
	Modelo Básico	47
	Modelo Estático Complexo	47
	Modelo Dinâmico Complexo	48
6.3	Social Virtual Objects	50
6.4	Social Virtual Objects	50
	REFERÊNCIAS	54
	GLOSSÁRIO	55

Parte I

Definições

Capítulo 1

OBJETO DA PESQUISA

Este capítulo tem como propósito definir um contexto de partida para este trabalho, apresentando um enquadramento da investigação realizada, suas motivações, objetivos e hipóteses de pesquisa que vão fundamentar a resolução do problema em estudo. Por fim, será descrita a estrutura de organização deste trabalho.

1.1 Contextualização

Com os avanços recentes no mundo eletrônico, tem aumentado exponencialmente a disponibilidade de dispositivos e sensores, seu hardware tem se tornado cada vez menor, mais poderoso e com custo infinitamente mais baixo, enquanto que o mundo dos softwares tem sido direcionado para o desenvolvimento de arquiteturas cada vez mais distribuídas e integrados a partir de microserviços. Esses fatos fazem com que processos de análise, monitoramento e controle do mundo físico passem a se tornar cada vez mais acessíveis, e tornando as tecnologias envolvidas cada vez mais populares como temos visto recentemente acontecer com as plataformas de Internet das Coisas (IoT) e de aplicativos para sistemas distribuídos em tempo real.

Técnicamente falando, dispositivos eletrônicos possuem diversas complexidades relacionadas a sua natureza de funcionamento que tornam extremamente imprevisíveis e complexos o processo de integração das diferentes tecnologias existentes para implantação das redes de sensores heterogêneos em conjunto com ferramentas distribuídas de coleta, processamento e visualização das informações em tempo real.

Olhando de uma perspectiva diferente, os dados gerados e coletados (ou sentidos no caso de sensores) pela web já são uma realidade, criando uma necessidade real de coleta, análise

e enriquecimento dessa elevada quantidade de informações, estreitando cada vez mais o hiato existente entre a compreensão do mundo real e sua representação a partir de um mundo virtual. Para isso, tem crescido a necessidade de identificar virtualmente as entidades de nossa vida cotidiana, de forma a possibilitar que elas possam interagir entre si ou com pessoas do mundo real.

Esses desafios tem sido reunidos em conceitos como a Internet das Coisas (IoT), a Web das Coisas (WoT), Web Social das Coisas (SWoT) e Nuvem das Coisas (CoT). Conceitos que reuniram grande atenção da academia, da indústria e de muitas outras áreas.

No entanto, ao colocar em prática esses conceitos e expor ou coletar esses dados na Web, nos confrontamos com vários problemas no caminho, como a heterogeneidade das fontes de dados, a grande quantidade de dispositivos ou equipamentos que possuem capacidade, precisão, alcance ou frequências distintas. Além dos desafios físicos, também é necessário resolver diferentes questões técnicas como o suporte à utilização dos diferentes protocolos e tecnologias de IoT existentes para publicação de dados na Web (CoAP, XMPP, MQTT, etc.) e utilizados para resolução de diferentes casos de uso (??).

Uma das maneiras de abordar esses problemas de heterogeneidade é seguindo uma abordagem semântica. Usando modelos semanticamente ricos (ontologias que podem ser estendidas para um caso de uso específico), uma série de sistemas [21, 19] mostraram como fontes de dados muito desiguais podem ser compartilhadas e mutuamente compreensíveis, seguindo padrões e princípios emergentes, como Linked Dados [7]. No caso mais específico de dados de sensores, ontologias e vocabulários específicos como a Ontologia SSN (Sensed Network Semantic) [10] foram criados pela comunidade e já foram adotados em vários projetos [6, 14, 9, 16, 20]. Os padrões existentes para publicar e acessar dados anotados semanticamente (SPARQL1, Linked Data Platform2, etc.) estão ganhando adoção e estabelecendo melhores práticas para compartilhar dados

Middleware Em um middleware de IoT (Ex: X-GSN), aplicativos e usuários podem acessar dados de objetos e coisas interligados, escondendo a comunicação interna e aspectos de aquisição de baixo nível.

Plataforma Plataformas (Ex: OpenIoT) extrapolam o âmbito de um sistema ou middleware. Ele é composto por diversos módulos de sistemas e middlewares diferentes que em conjunto executam o trabalho proposto por uma plataforma.

Arquitetura Uma Arquitetura descreve todos os aspectos de uma plataforma, quais são seu módulos, como será realizada a interação entre eles, quais tipos de mensagens podem ser utilizadas em cada cenário e etc..

Modelo Um modelo é uma representação de objeto, contexto ou uma engrenagem específica utilizada para definição de uma arquitetura, em uma escala reduzida, de forma a dar maior foco á solução de um problema específico.

Objetos Virtuais Em uma plataforma de IoT (Ex: OpenIoT), objetos virtuais são .

1.2 Objetivos

Como em diversos cenários urbanos da Internet das Coisas, também tem se tornado bastante comum a utilização de diversos tipos de sensores integrados para compor informações mais complexas e relevantes aos variados tipos de negócio existente na indústria e ao dia-a-dia das pessoas.

Dentro dos diversos cenários que podem ser apoiados com a utilização de plataformas IoT, observa-se que aspectos referentes á identificação e ao endereçamento de recursos se tornam relevantes. Ao mesmo tempo, por possuírem especificidades em seus domínios, os mecanismos de nomeação e acesso a recursos normalmente diferem em cada cenário, tornando mais difícil a integração entre os sistemas presentes em cada um deles.

Com isso, considerando a massificação dos equipamentos eletrônicos e das redes de Sensores Sem Fio (RSSF), que reduziram de forma substancial os custos de implementação das plataformas de IoT e os desafios atuais para análise e controle dinâmicos de dispositivos móveis e conectados utilizando sua localização e contexto ao qual está inserido, os objetivos deste trabalho são:

- avaliar aplicabilidade do conceito de Internet das Coisas Semântica em cenários que incluem dispositivos conectados que se movem fisicamente dentro de um espaço controlado, pontuando modelos, arquiteturas e plataformas existentes;
- pesquisar outros modelos e tecnologias que possuem relevância ao trabalho proposto;
- analisar principais elementos e recursos aplicados, identificando os atributos mais relevantes para composição de suas representações virtuais e que sejam aplicáveis à formatação de estratégias de análise, movimentação desses recursos;

- analisar possíveis impactos na semântica dos recursos para sua correta identificação e definição dentro das redes de sensores avaliadas;
- definir conjunto de regras básicas para classificação e visualização dos dados recebidos de recursos conectados à rede.
- identificar as associações disponíveis entre os diversos tipos de recursos conectados;
- estruturar na forma de um grafo os diversos tipos de recursos identificados e suas possíveis associações.
- aplicar técnicas de nomeação e recomendação na otimização da busca dos recursos conectados à rede, de modo que sejam selecionados apenas os recursos mais adequados às regras definidas nos canais de publicação.
- Elaborar um middleware distribuído de apoio ao processo de mapeamento das regras de publicação, análise e visualização da informação processada em tempo real, considerando os diversos aspectos que influenciem esse processamento.

1.3 Metodologia

Para o desenvolvimento deste documento, o método de trabalho proposto consiste na realização das seguintes atividades:

- Revisão na literatura para identificar trabalhos existentes com propostas de arquiteturas, modelos e plataformas para IoT onde em conjunto com a análise de localização e contexto dos objetos virtuais, simplificaria o processo de aplicação dessas plataformas para em contextos mais dinâmicos do nosso dia-a-dia;
- Análise crítica das soluções propostas por outros autores, pontuando suas principais deficiências e como extendê-las de forma a viabilizar sua aplicação em cenários dinâmicos com dispositivos que se movimentam dentro do espaço físico avaliado;
- Desenvolver um novo middleware de IoT para análise de dispositivos que se movimentam dentro do espaço físico controlado por uma plataforma de IoT;
- Criar uma interface para visualização e análise dos resultados em tempo real;
- Analisar o novo modelo proposto e os resultados obtidos;

1.4 Organização do Trabalho

Este documento foi organizado em 3 partes e 8 capítulos fundamentando a proposta do trabalho realizado e que estão distribuídos da seguinte forma:

- Primeira parte** Apresentação dos objetivos da pesquisa, os métodos utilizados para seu desenvolvimento, contextualização do tema e definição de teorias e práticas existentes na literatura que tenham relevância ao desenvolvimento deste documento;
- Segunda parte** Desenvolvimento do trabalho realizado, debatendo os resultados esperados com a sua conclusão;
- Terceira parte** Por fim, as contribuições do trabalho proposto, apresentação de alguns aspectos sobre o tema com relevância na continuação das pesquisas em trabalhos futuros e conclusão da pesquisa realizada.

Parte II

Temas Relacionados

Capítulo 2

IoT E SEMÂNTICA

Este capítulo apresenta uma visão geral sobre os conceitos em torno da criação de uma Internet das Coisas semântica, seu funcionamento e seus principais modelos, arquiteturas e plataformas existentes e descritos a partir de diferentes trabalhos acadêmicos.

2.1 Internet das Coisas (IoT)

Em nosso dia-a-dia, podemos perceber os diversos indícios de como a Internet das Coisas (IoT) irá conectar os bilhões de sensores previstos para serem implantados ao redor do mundo até 2020 segundo previsões do IDC(??). Voltando ao final do século passado, Weiser já previa em suas publicações, como seriam as principais transformações do conceito de computador pessoal (WEISER; GOLD; BROWN, 1999) que viria a compor um novo conceito cunhado de "Computação Ubíqua" (WEISER, 1991) (WEISER, 1993) que com o desenvolvimento e ganho de popularidade deu origem à Internet das Coisas como conhecemos nos dias atuais.

Durante as últimas décadas houve uma enorme popularização da internet, evoluindo as tecnologias de comunicação sem fio, massificando a utilização de dispositivos móveis conectados e viabilizando o crescimento das redes sociais. Essa realidade tem permitido que as pessoas possam se comunicar ou realizar grande parte das tarefas do dia-a-dia a qualquer momento, de qualquer lugar, sem muitos problemas, apenas estando conectando na internet.

O conceito de Internet das Coisas propõe a criação de um ecossistema de "coisas"interconectadas entre si. Se entende por "coisa", objetos geralmente compostos por pequenos sistemas computacionais embarcados que contam com a habilidade de captar dados do ambiente analisado trocando informação com outras "coisas"de forma direta ou através de uma rede de interconexão (ex: internet).

Sua principal característica, que acaba inviabilizando uma implantação universal da Internet das Coisas nos dias atuais, é quanto a forma de busca e comunicação entre os diversos dispositivos ubíquos disponíveis na rede, que deve ser baseado em um identificador único dentro na rede (ex: Seu IP) ou seja, cada "coisa" conectada à rede terá que possuir seu identificador próprio. Porém, enquanto a migração do protocolo IPv4 para o IPv6 não se concretizar, será economicamente inviável implementar um ecossistema novo para a Internet das Coisas que seja amplamente universal. Fora isso, como essa característica não é a única, e novas tecnologias continuam sendo pesquisadas e lançadas no mercado, nada impede que sejam feitas adaptações nas redes existentes para alcançar implementações baseadas na Internet das Coisas. A mais promissora delas é a "Web das Coisas" cujo ecossistema é baseado em protocolos da Web existentes e que já estão consolidados, como detalhado no próximo item.

2.2 Redes de Sensores Sem Fio

A queda crescente dos custos de equipamentos eletrônicos e a redução no tamanho dos dispositivos simplificaram a sua mobilidade, fazendo com que as redes sem fio ganhassem rápida popularidade mundo afora, ganhando novos tipos de aplicações com objetivo de simplificar o dia-a-dia das pessoas, como previsto por Weiser em (WEISER, 1991).

Atualmente, tem crescido de forma exponencial a variedade de sensores(??) existentes e que podem ser obtidos sem muita dificuldade pela internet, onde junto à uma maior consciëntização das pessoas quanto aos benefícios que a interligação desses dispositivos em rede para análise em tempo real de informações ambientais do dia-a-dia, que geralmente são invisíveis a nossa atenção (WEISER; BROWN, 1997), faz da rede de sensores sem fio um dos pilares de sustentação da Internet das Coisas (IoT).

Segundo Akyildiz em (??), uma rede de sensores se diferencia de outras redes por ser composta de uma grande volume de nós sensores, que se encontram dentro do fenômeno natural analisado ou muito próximo a eles. Por isso, decisões sobre a arquitetura, modelo e equipamentos utilizados em uma Redes de Sensores Sem Fio (??) (RSSF ou WSN) se tornaram fator de sucesso em projetos de IoT.

Em uma pesquisa descrita em (??), com base em diferentes trabalhos já publicados anteriormente, Silva apresenta uma revisão das diversas possibilidades de aplicações das redes RSSF no contexto da agricultura de precisão onde esses trabalhos são delineados e caracterizados, possibilitando o uso das informações detalhadas em seu trabalho como dimensões do projeto proposto por este estudo.

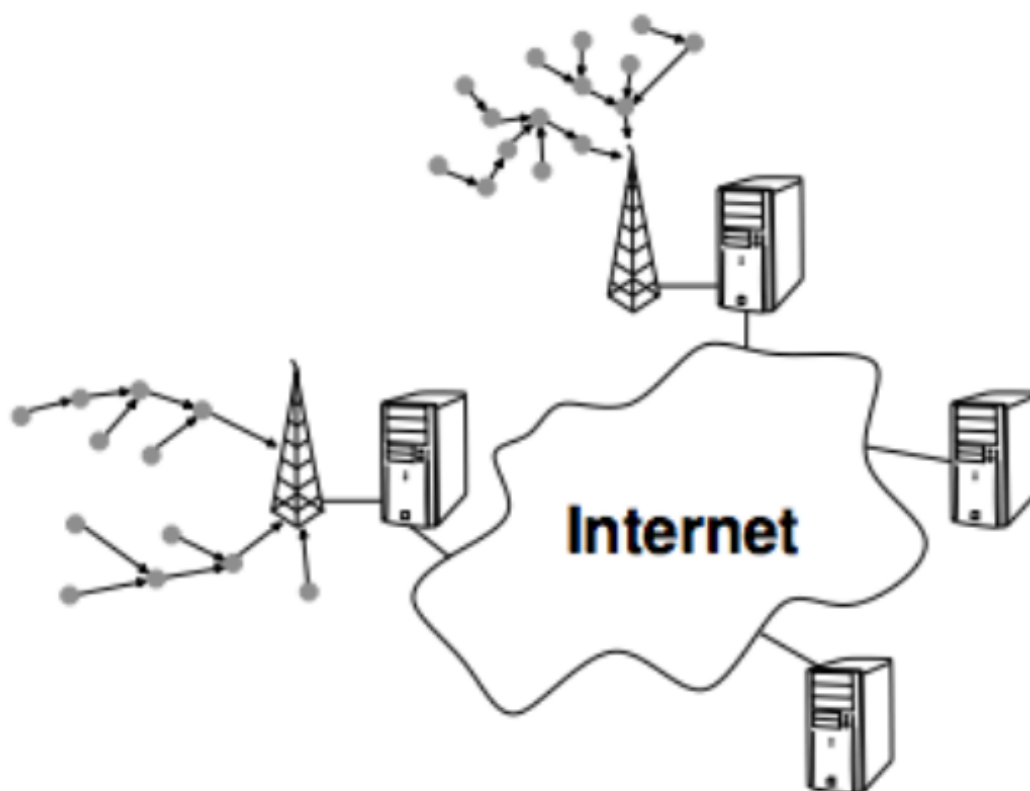


Figura 2.1: Visão física de uma rede de sensores

2.3 IoT semântico

(??)

Segundo a percepção do paradigma de microserviços, middlewares de IoT permitem que os usuários de dados ou informações que estejam na nuvem possam recuperar os dados desejados sem ter consciência de detalhes técnicos subjacente aos recursos que a criaram.

Por outro lado, o processo de configurar uma plataforma de IoT e posteriormente recuperar suas informações de forma correta e eficiente passa a ser um desafio muito elevado para alguém que apenas precisa da informação (os consumidores dos dados), pois requer conhecimentos muito técnicos ou de domínios bem específicos e que não necessariamente são familiares aos seus consumidores.

Isso nos leva aos estudos sobre a importância que uma abordagem baseada no conhecimento tem para a simplificação dos processos de configuração em plataformas ou middlewares de IoT de modo que seus usuários possam ficar apenas na informação que precisa, sem se preocupar com quais dispositivos ou tipos de dados são necessários para se chegar ao resultado esperado.

Em () e (), são demonstrados como recursos de uma plataforma de IoT podem ser descritos

com o uso de semântica, possibilitando que sejam utilizados na composição de novos serviços expostos a partir de fluxo de processamento. Essa abordagem automatizada para composição dos novos recursos de IoT baseados em "conhecimento"semântico tem alcançado um importante papel na evolução dos estudos sobre o tema.

Em (), Pereira demonstra como Ontologias podem ser utilizadas para representação, descoberta e correspondência de serviços expostos utilizando a rede mundial de computadores em sistemas embarcados de tempo real distribuídos

De modo a reduzir esses problemas de descoberta, estudos sobre os princípios de conhecimento e contexto tem sido amplamente apresentados de forma a serem incluídos como novos critérios para seleção de serviços.

No entanto, para usar a informação de contexto de forma eficaz, deve ser apresentado de forma inequívoca e a natureza dinâmica dos sistemas incorporados e em tempo real deve ser considerada. Para enfrentar esses desafios, os autores apresentam uma estrutura de descoberta de serviço para DERTS que usa ontologia de sistemas embutidos e em tempo real e um algoritmo de correspondência semântico para facilitar a descoberta de serviços de dispositivos em ambientes de sistema incorporados e em tempo real. A estrutura de descoberta de serviço proposto também considera as prioridades associadas com os requisitos colocados pelo solicitante durante o processo de descoberta do serviço.

Capítulo 3

ARQUITETURA GSN

(??)

A Global Sensor Networks (GSN) é um projeto iniciado pela EPFL em 2004 e cujo objetivo principal é o desenvolvimento de uma plataforma aberta e genérica para implementar redes de sensores e cujo processamento dos dados gerados por seus sensores é feita de maneira distribuída. Sua criação foi baseada no estudo dos requisitos encontrados nas plataformas desenvolvidas para sustentação de redes de sensores, que possuem grande semelhança entre si. Essa plataforma fornece infra-estrutura escalável para integrar diferentes tecnologias de sensores usando um pequeno conjunto de abstrações definidas, suporte a integração e descoberta de novas redes de sensores, consultas distribuídas, filtragem e combinação dos dados coletados de sensores. Durante sua operação, processos de configurações e adaptatividade dinâmica são feitas através de uma linguagem declarativa baseada em XML.

O middleware GSN possui um mecanismo agnóstico de protocolos extensível para adquirir dados de dispositivos sensores, usando wrappers configuráveis e implementando alguns desses protocolos.

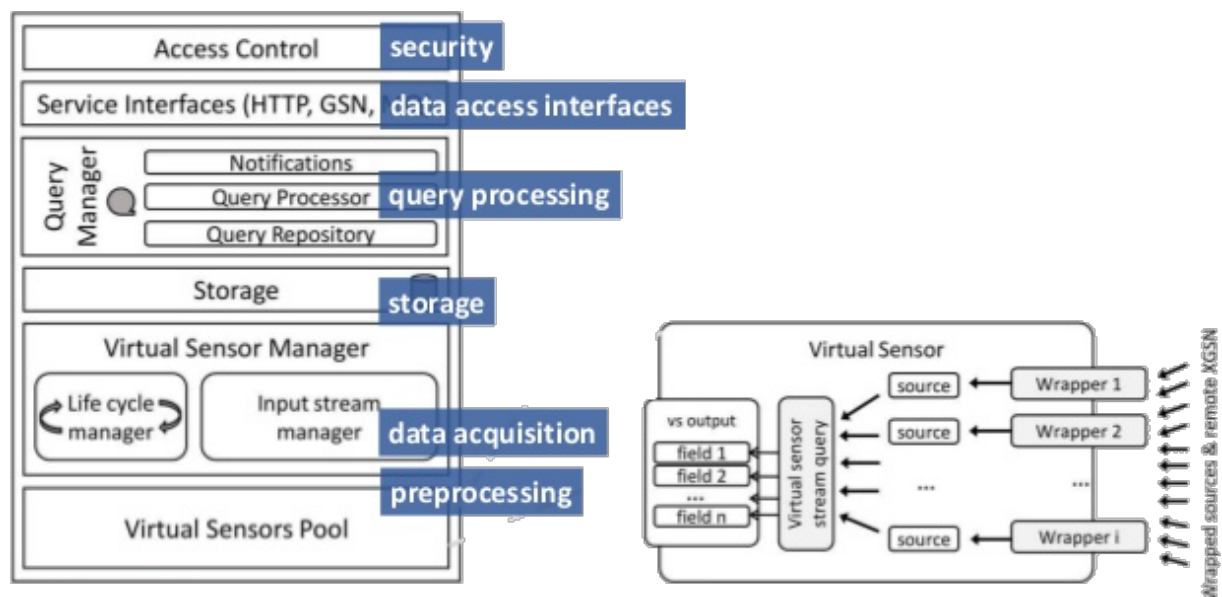


Figura 3.1: Visão orientada aos serviços dentro de uma rede de sensores

3.1 Objetos Virtuais

3.2 Processamentos dos streams de dados

3.3 Arquitetura e Implementação

3.4 Middleware de Extensão XGSN

Solução de middleware Open-Source descrito em (??) que lida com o ciclo de vida de sensores virtuais (dispositivos, objetos ou pessoas que observam propriedades em torno deles), fornecendo anotações semânticas para eles e os dados de observação que eles produzem. A idéia-chave é fornecer uma plataforma habilitada semanticamente de ponta a ponta para o gerenciamento de dados IoT, na qual o XGSN desempenha o papel de um middleware de aquisição de dados totalmente distribuído com recursos de anotação semântica. Descrevemos a arquitetura deste sistema e sua implementação, enfatizando o processamento de dados distribuídos que permite ao XGSN produzir diferentes camadas de observações agregadas.

Este middleware estende o GSN com a inclusão do conhecimento semântico que pode ser usado e estendido. Ele também pode ser utilizado como extensão para a plataforma OpenIoT abordada anteriormente, integrada ao Linked Sensor Middleware (LSM) (??) ou ser conectada a qualquer outra plataforma com armazenamento de dados habilitado para uso RDF.

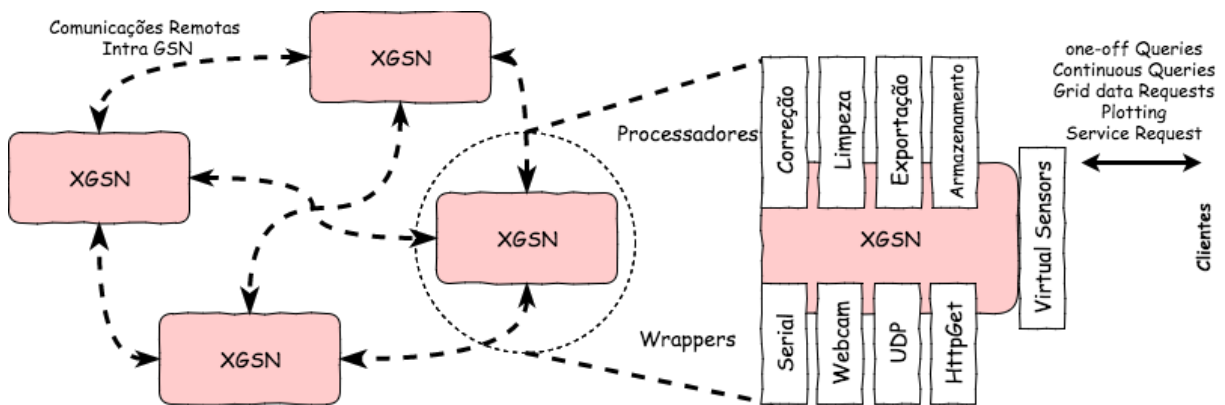


Figura 3.2: Arquitetura alto nível GSN

Abordagem geral Essa semântica explícita para representação virtual de um sensor facilita as tarefas de descoberta e pesquisa dentro de um ambiente de IoT.

Também para observações reais, o XGSN pode fornecer diferentes níveis de anotações semânticas, dependendo do tipo de sensor virtual exposto.

Por exemplo, em um cenário de qualidade do ar, o XGSN pode anotar cada medida feita por um sensor (incluindo valor, unidade, tipo de dados, etc.) como uma observação de uma propriedade (por exemplo, NO₂), conforme descrito na Figura 2.

No entanto, para alguns casos de uso de anotações de baixo nível não são úteis ou relevantes, portanto, o XGSN pode agregar, filtrar ou processar várias observações ao longo do tempo ou espaço. Isso produzirá indicadores em um sensor virtual de nível superior, cada um dos quais pode ser anotado com conceitos de nível superior de uma ontologia de domínio (por exemplo, uma observação de "baixa qualidade do ar"). Além disso, as correlações ainda mais complexas e o processamento, incluindo fontes de dados externas ou dados de outras instâncias do XGSN, podem levar a anotações que denotam conceitos passíveis de ação e humancompreensíveis, como alertas ou atividades.

Aspectos de gerenciamento de ontologia

Arquitetura

Exemplos Projetado como uma plataforma altamente extensível aproveitando as capacidades existentes no middleware GSN e baseado em representações semânticas de metadados e observação de sensores para orientação do processo de avaliação e publicação dos dados de sensores na Web, sendo este capaz de lidar com o processo de obtenção dos dados provenientes

de um grande número de dispositivos e protocolos diferentes.

Além dos sistemas tradicionais de gerenciamento de sensores, ele é capaz de enriquecer as descrições de sensores virtuais com conteúdo semanticamente anotado usando vocabulários padrão. Na abordagem proposta, os dados e observações dos sensores são anotados usando uma rede de ontologia baseada na ontologia do SSN, fornecendo uma representação padronizada e consultiva que facilita o compartilhamento, a descoberta, a integração e a interpretação dos dados. O XGSN gerencia o processo de anotação para as observações do sensor de entrada, produzindo fluxos RDF que são enviados para o Middleware do Sensor Vinculado habilitado para a nuvem, que pode armazenar os dados internamente ou executar o processamento contínuo de consultas. A natureza distribuída do XGSN permite a implantação de diferentes instâncias remotas que podem trocar dados de observação, de modo que os sensores virtuais possam ser agregados e consumir dados de outros sensores virtuais remotos. Neste artigo, mostramos como essa abordagem foi implementada no XGSN e foi incorporada à plataforma OpenIoT mais ampla, fornecendo um sistema altamente flexível e escalável para gerenciar o ciclo de vida dos dados do sensor, desde a aquisição até a publicação, no contexto da semântica. Web of Things.

Capítulo 4

PLATAFORMA OPENIoT

(??)

OpenIoT é uma implementação de código aberto da arquitetura de referência IERC que surgiu a partir do projeto OpenIoT(??) sendo uma das plataformas pioneiras em IoT a propiciar uma interoperabilidade semântica entre diferentes serviços de IoT na nuvem. Tendo papel fundamental no seu processo operacional, a ontologia SSN da W3C fornece um modelo genérico e baseado em padrões para criar representações virtuais de sensores físicos.

Sua arquitetura é composta por sete elementos principais, que pertencem a três camadas lógicas diferentes, conforme ilustrado na figura ??.

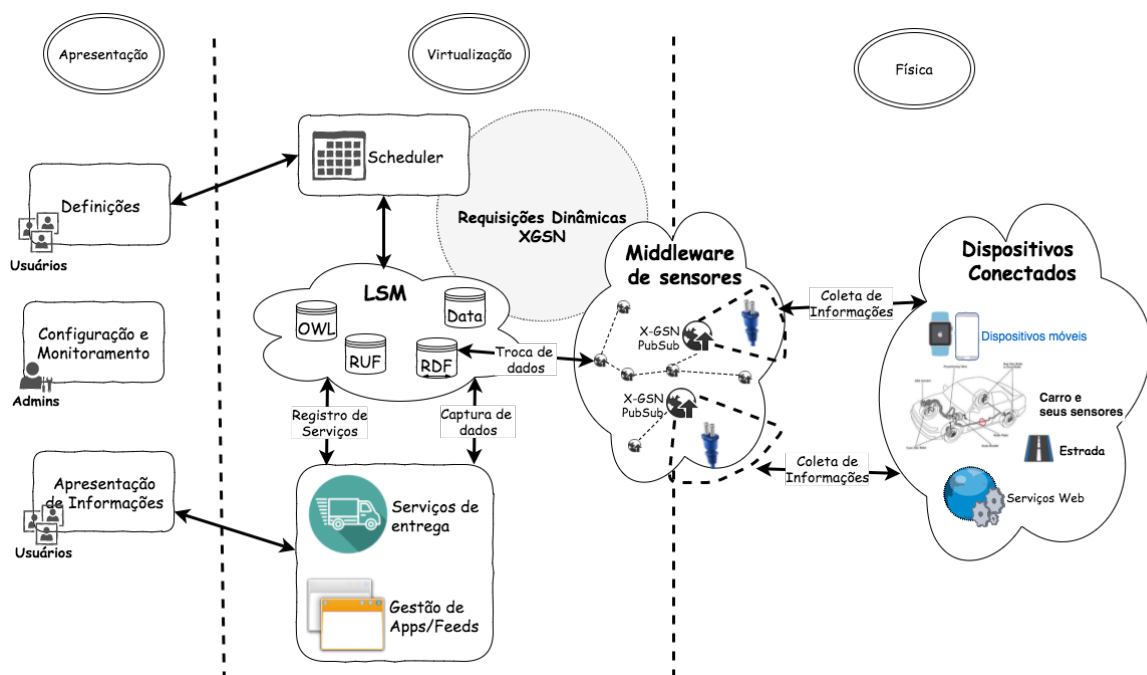


Figura 4.1: Arquitetura OpenIoT

Suas três camadas lógicas são:

4.1 Aplicação

Camada responsável pelas especificações on-the-fly de solicitações de serviço para a plataforma OpenIoT, fornecendo uma interface Web 2.0. Compreende um conjunto de serviços para especificar e formular tais solicitações, além de enviá-los ao Agendador Global. Dentre os componentes utilizados por esta camada estão:

Definições Permite a especificação on-the-fly de solicitações de serviço dentro da plataforma OpenIoT através de uma interface Web. Compreende um conjunto de serviços para especificar e formular tais solicitações, além de enviá-los ao módulo Scheduler.

Apresentação Seleciona os applets apropriados para facilitar a apresentação dos resultados das solicitações. Para visualizar essas solicitações, este componente se conecta diretamente ao módulo de Serviços de entrega para recuperar os dados relevantes.

Configuração e Monitoramento Permite a gestão, configuração e monitoramento das funcionalidades da plataforma OpenIoT.

4.2 Virtualização

4.2.1 Scheduler

Elemento da plataforma responsável por elaborar as solicitações baseadas nas informações enviadas pelo usuário (definição da solicitação), analisando cada requisição e atuando como ponto de interação com o resto da plataforma através de um banco de dados na nuvem. Possui as seguintes funcionalidades:

Seleção de Sensores e Wrappers Efetua consultas ao Service Discovery (baseado em propriedades semânticas dos sensores) através da entidade `?availableSensors?` para encontrar um conjunto de sensores (físicos ou virtuais) e ICO/Wrappers que cumprem os critérios especificados na solicitação. Como resultado da consulta recebe uma lista de sensores (com seus

identificadores únicos) que são utilizados na entrega do serviço especificado (ou seja, uma lista de sensores que cumprem critérios definidos como parte da solicitação de serviço).

Ativação "indireta" Sensores Virtuais Após a identificação dos Sensores/Wrappers que atendem aos critérios especificados, os nós GSN (sensores virtuais) são informados sobre qual deles é utilizado pelo serviço agendado. Procedimento aplicado no horário que o pedido será executado, mantendo essas informações na entidade "sensorServiceRelation" que é acessada pelos sensores virtuais para atualização.

Armazenamento e execução de solicitações Ele é responsável por identificar as informações sobre os serviços relacionados a uma solicitação específica a serem entregues ao Gerenciador de Entregas/Apps. Esse procedimento é realizado através das informações armazenadas na entidade "serviceDeliveryDescription".

Controle do estado das solicitações Com base nos ciclos de vida dos serviços, o Scheduler atualiza o estado das solicitações na entidade "serviceStatus", utilizada quando necessário para informar ao usuário o estado de sua requisição.

Controle de acesso Por fim, ele é responsável pelos diversos mecanismos de controle do acesso às entidades da plataforma com a ajuda de uma entidade chamada "user"

A Figura seguinte ilustra o ciclo de vida dos serviços IoT dentro da plataforma OpenIoT. As transições entre os diferentes estados ocorrem com base nas requisições feitas às APIs do Scheduler.

4.2.2 Serviços de entrega e gestão de aplicativos

Esse elemento possui uma dupla funcionalidade. Por um lado (gestor de serviços de entrega), permite a recuperação de dados dos sensores que compõem a solicitação e num segundo momento (gestão de aplicativos), mantém e recupera estruturas de informações sobre a utilização de serviços, dando suporte à processos de medição, cobrança e gerenciamento de recursos. Portanto, seu papel é fornecer meios para:

- Executar os serviços solicitados, devolvendo seus resultados;
- Acessar e processar fluxos de dados oriundos da LSM;

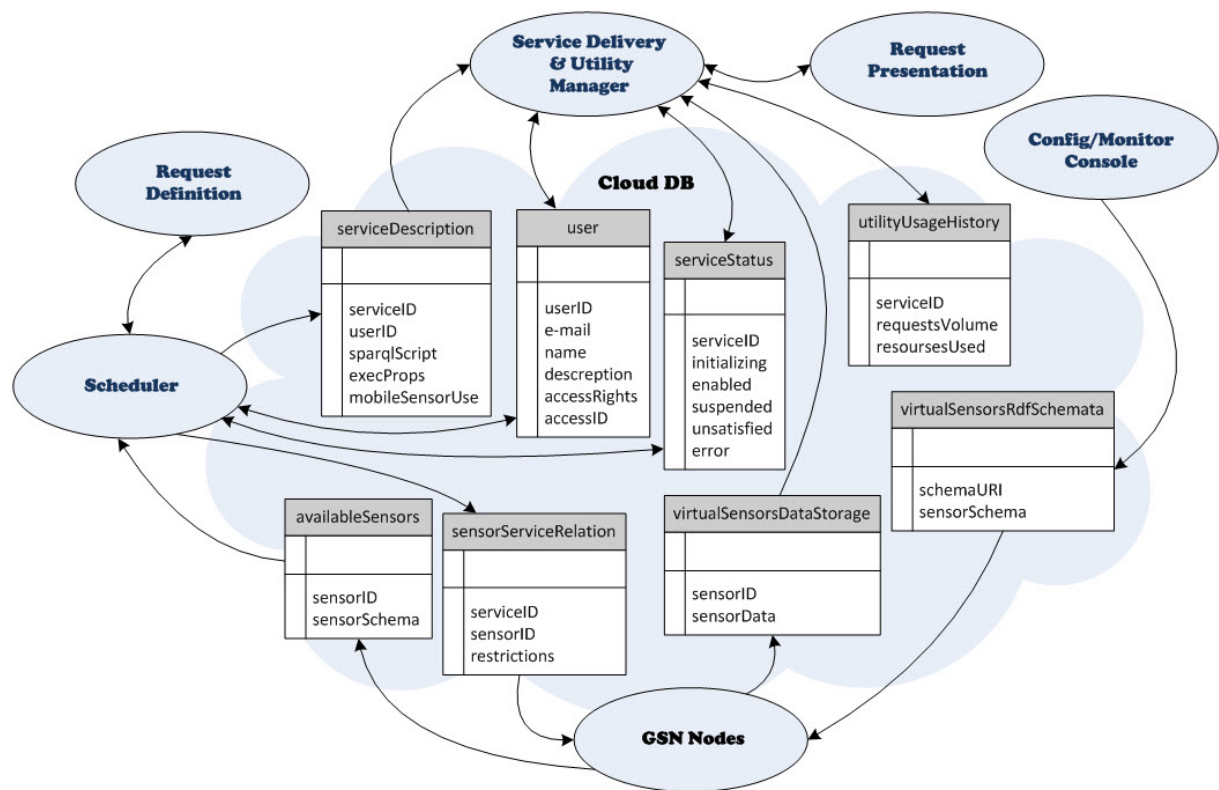


Figura 4.2: Arquitetura OpenIoT

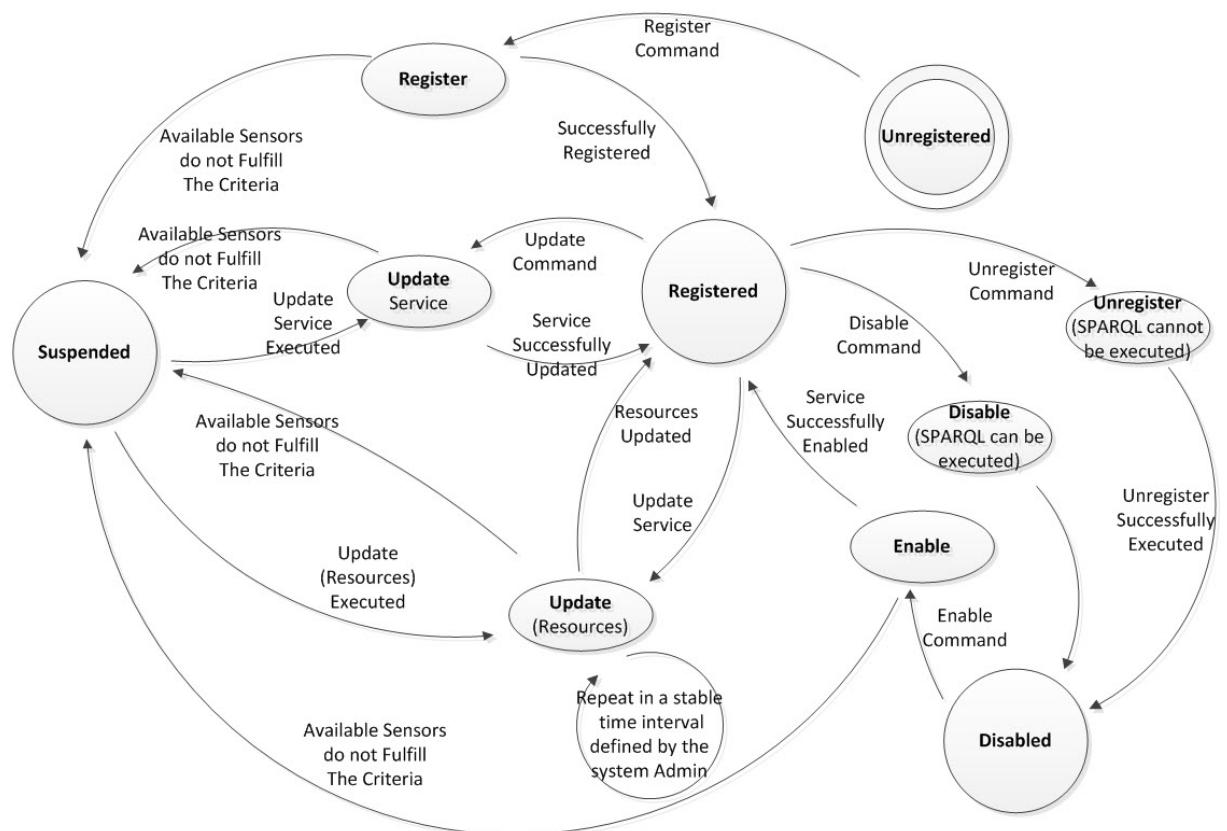


Figura 4.3: Arquitetura OpenIoT

- Considerar as instruções de processamento definidas durante o processo de formulação do pedido (Schedule);
- Acompanhar parâmetros de referência associados aos serviços em execução (ex:Tempo que um serviço é utilizado, Volume de dados transmitidos, Número e tipo dos sensores utilizados);
- Gerenciar e manter os registros desses parâmetros de referência;
- Executar rotinas e gerar relatórios de faturamento e cobrança com base nos recursos utilizados e parâmetros de referência.

4.2.3 Plataforma de relacionamento dos dados - LSM-Light

(??) No tratamento do processamento e fornecimento dos dados oriundos de sensores conectados na rede, o OpenIoT utiliza um middleware central chamado de Linked Sensor Middleware responsável pela identificação de todos os sensores definidos em sua base, facilitando o processo de coleta dos dados e garantindo ao mesmo tempo a sua própria anotação semântica. Ele possui as seguintes atribuições dentro da plataforma:

- Suas fontes de dados de entrada são sensores virtuais;
- Transforma os dados provenientes dos sensores virtuais em uma representação de dados vinculados (RDF), que são anotados de acordo com as ontologias suportadas;
- Importa ou transmite dados utilizando 2 tipos de abordagens: pull e push;

Na abordagem Pull, a pesquisa é feita periodicamente na fonte dos dados através de canais semelhantes a feeds de dados. Por outro lado, ao utilizar a abordagem Push, a fonte dos dados que passam a ser responsáveis pelo envio dos dados à plataforma. A plataforma OpenIoT da suporte à ambas abordagens, utilizadas dependendo do caso de uso real, cada abordagem possui diferentes vantagens e desvantagens descritas pelo seguinte trabalho (??).

4.3 Física

4.3.1 Middleware de controle de sensores (XGSN)

Um middleware de extensão da plataforma que lida com o ciclo de vida dos sensores virtuais ou dispositivos físicos, coletando, filtrando, combinando e aplicando anotações semânticas

no fluxo de dados produzidos por eles. Em outras palavras, ele funciona como um hub entre o mundo virtual e o físico. Seu funcionamento é baseado em instâncias distribuídas que pertencem a diferentes entidades administrativas.

4.3.2 Controle dinâmico de Sensores

Para criar um ambiente de IoT universalizado, precisa existir um forte compartilhamento e gestão de recursos dentro da rede de sensores. Na plataforma OpenIoT, o módulo X-GSN foi estendido de forma a poder identificar dinamicamente os sensores que estão atualmente ativos dentro da plataforma. Essa extensão cria uma nova funcionalidade que faz buscas periódicas ao LSM-Light analisando quais sensores continuam ativos, ativando/desativando dentro do módulo X-GSN os sensores virtuais relevantes.

O diagrama UML abaixo descreve as classes que compõem esta extensão de controle dinâmico dos sensores.

A interface Parser em conjunto com a classe SensorParse são responsáveis pela análise dos metadados RDF utilizando o LSM-Light e pela extração de identificadores dos sensores virtuais, enquanto que a classe DynamicControlTask encapsula as operações de ativação/desativação dos sensores virtuais. Utilizando o SparqlClient incluído neste módulo, as consultas SPARQL podem ser executadas, obtendo como resultado os identificadores dos sensores virtuais que serão ativados ou desativados.

4.4 Principais objetos na plataforma OpenIoT

4.4.1 OSDSpec

Esquema para especificação de descritores das solicitações

4.4.2 OSMO

Esquema com a definição dos objetos utilizados em sensores

4.4.3 SensorTypes

Esquema para definir o tipo de sensor utilizado

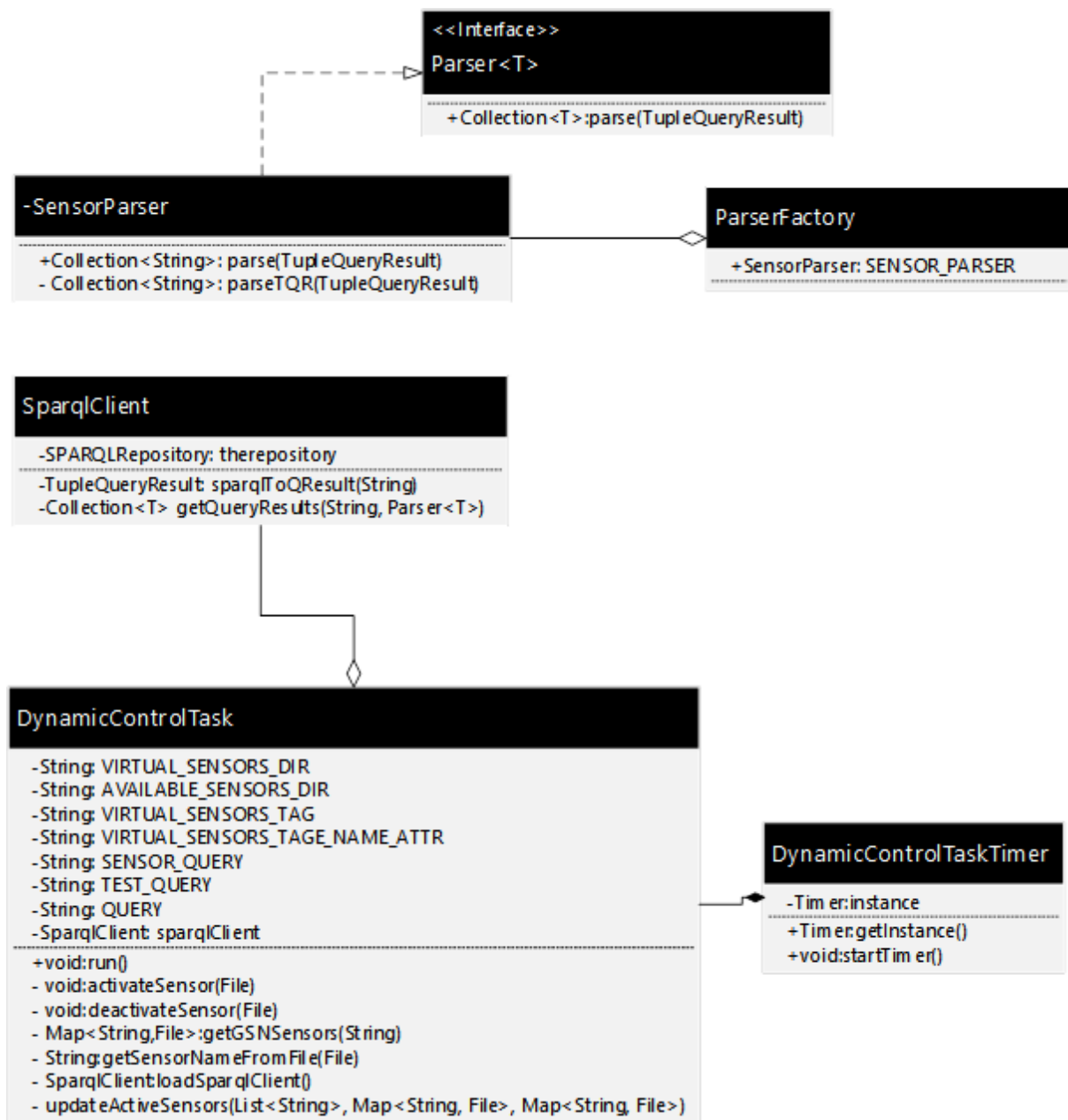


Figura 4.4: Arquitetura OpenIoT

4.4.4 DescriptiveIDs

Esquema de identificação de recursos dentro da plataforma

4.4.5 SdumServiceResultSet

Esquema de Result Set utilizado para retorno das solicitações

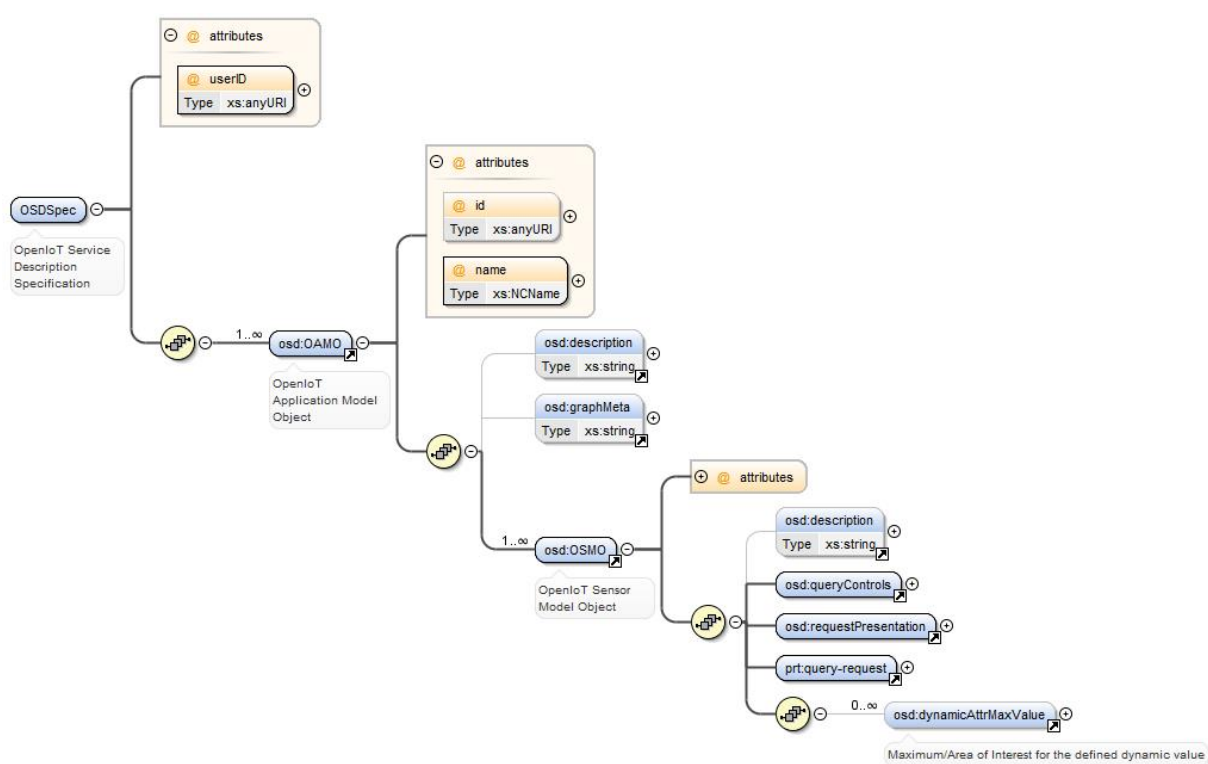


Figura 4.5: Esquema de um Descritor de serviço

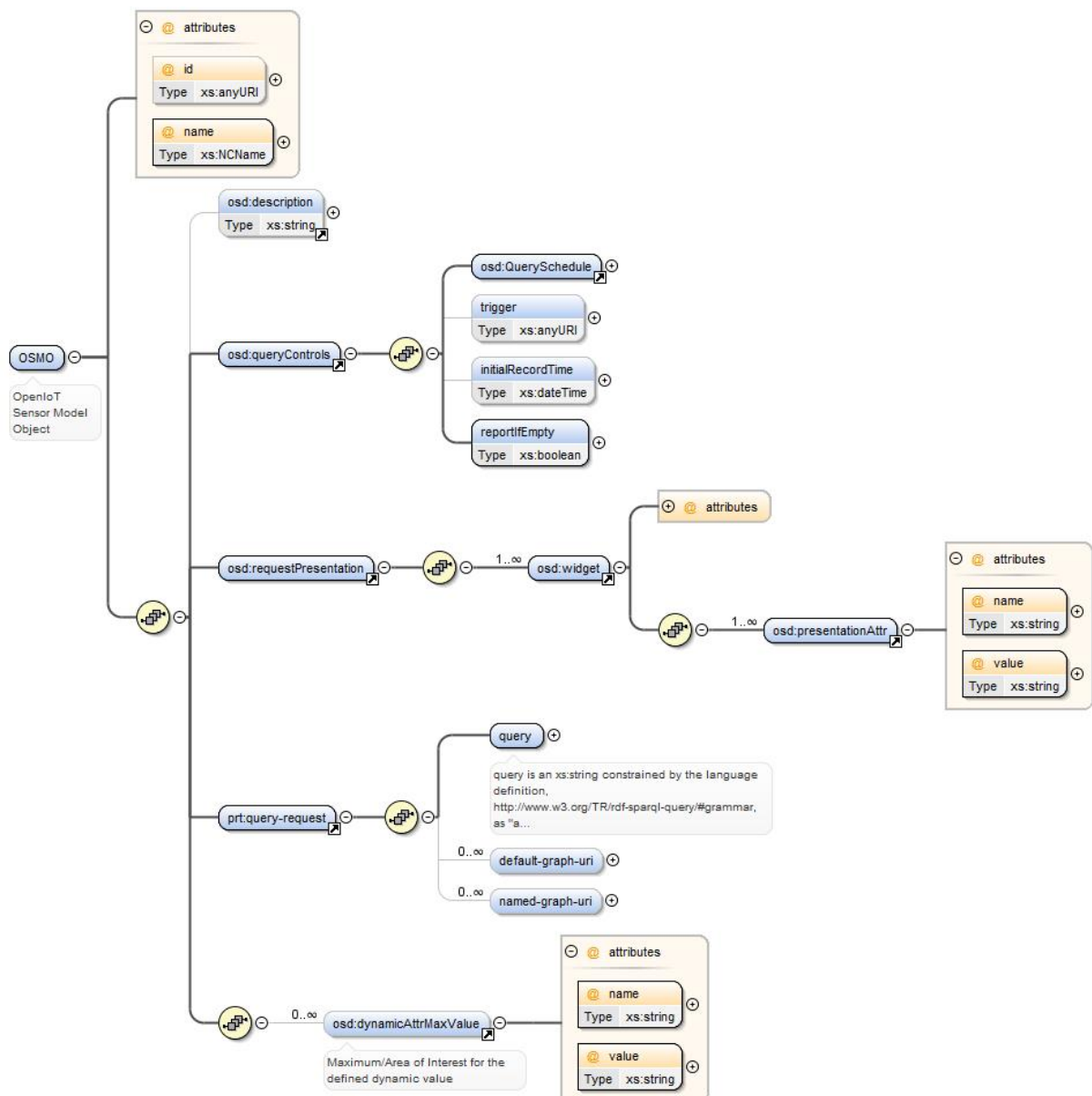


Figura 4.6: Esquema de um Sensor (o esquema OSMO está incluso no OSDSpec)

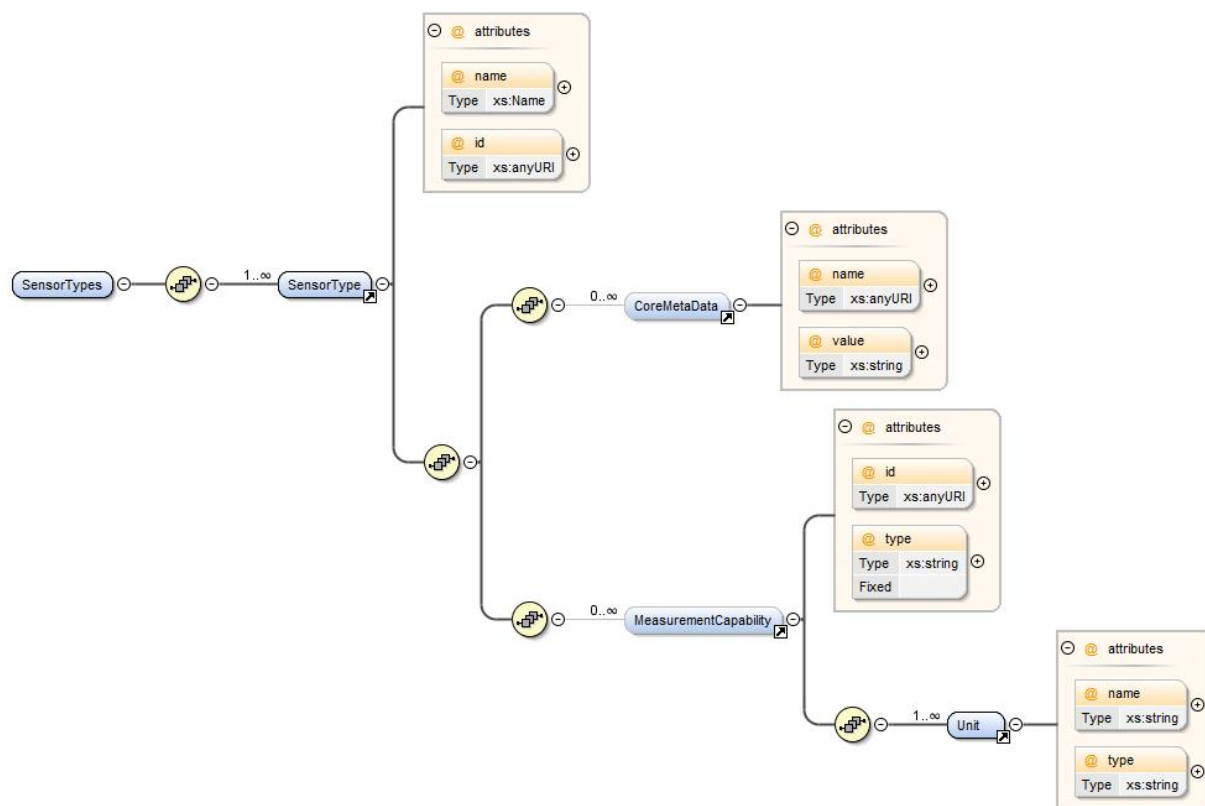


Figura 4.7: Esquema de um tipo de sensor

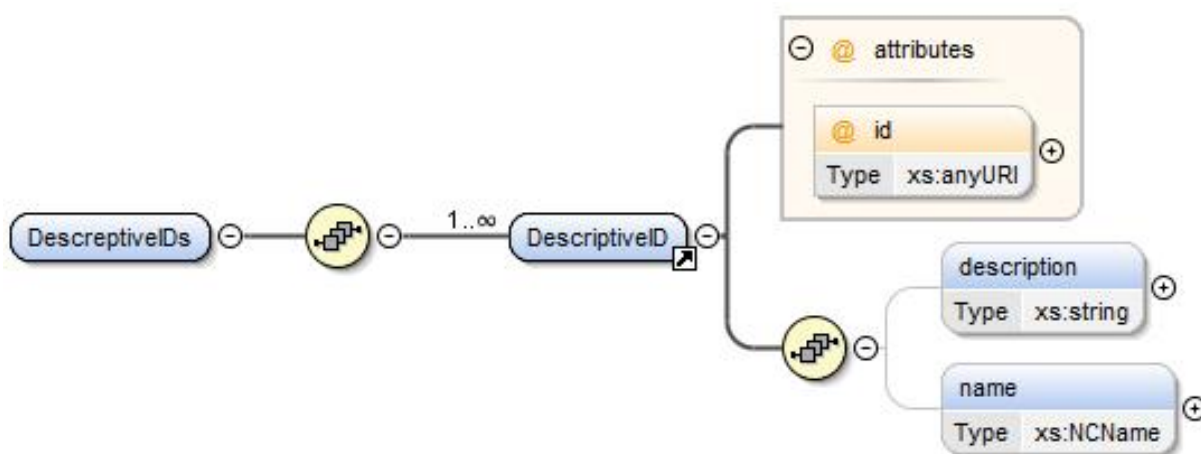
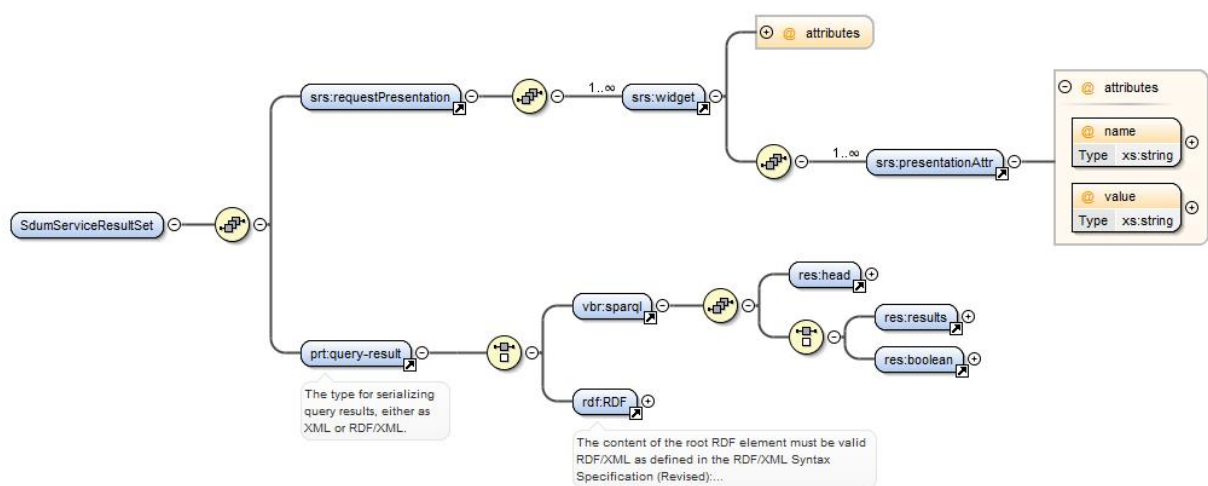


Figura 4.8: Esquema de um identificador

**Figura 4.9: Esquema de um identificador**

Capítulo 5

ARQUITETURA CA4IoT

Em (??), Perera propõe uma arquitetura orientada por contextos com objetivo principal de abstrair pro usuários as complexidades técnicas existentes nas plataformas de IoT, automatizando as tarefas de seleção dos sensores avaliados de acordo com os problemas e tarefas definidas pelos usuários. Essa plataforma se diferencia das outras existentes pelo fato de utilizar regras semânticas na identificação dos recursos que serão avaliados pela plataforma, enquanto as outras possuíam como foco principal catalogar sensores físicos, fazendo a seleção dos recursos por seus identificadores ou alguns de seus atributos.

Seu trabalho identifica os principais pontos necessários ao desenvolvimento de uma plataforma para IoT disponibilizando em forma de serviços os sensores virtuais controlados pela plataforma. Esses dispositivos virtuais são compostos pelo processamento dos dados coletados de um ou mais sensores diferentes que a partir de um fluxo de dados (streams) utilizam os diversos mecanismos de filtragem, associação e lógica disponibilizados pela plataforma para definir sua semântica.

Sua proposta de arquitetura foi concebida em 3 partes [figura ??] que define as capacidades funcionais necessárias para uma plataforma desse tipo, a divisão da arquitetura em camadas de alto nível e por fim a definição dos diversos módulos necessários para a implantação da arquitetura como um todo. A seguir será apresentado uma breve descrição da proposta realizada, suas diversas camadas e os elementos e componentes que fazem parte da solução.

5.0.1 Capacidades funcionais da plataforma

Com as capacidades funcionais descritas nesta seção, à plataforma podem obter o resultado solicitado a partir de três etapas, onde inicialmente o usuário descreve suas necessidades através de uma interface amigável, a plataforma avalia semanticamente o problema do usuário selecio-

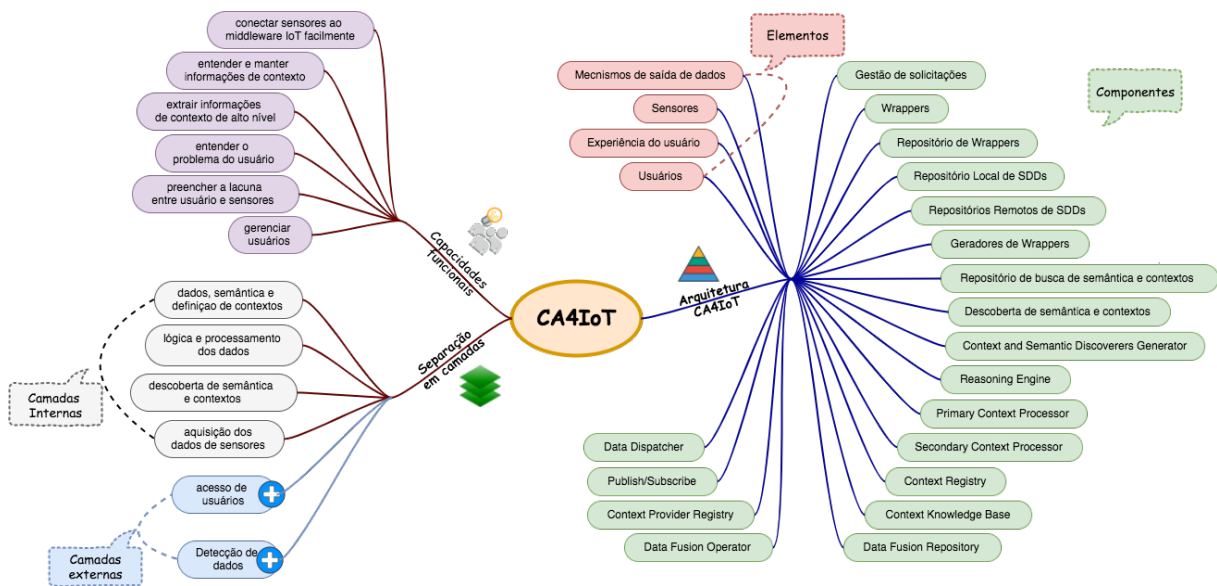


Figura 5.1: Concepção da arquitetura dividida em 3 partes

nando os sensores apropriados e por fim, os dados dos sensores seleccionados são recuperados, combinados e enviados ao usuáriu ou tilizando um stream de dados único. A fim de projetar a arquitetura proposta, Pereira inicialmente identificou os seguintes requisitos funcionais da solução:

Conectar sensores ao middleware IoT facilmente Esta é uma funcionalidade importante em que se espera que bilhões de sensores estejam conectados às soluções de middleware IoT. Devido à escala, não é possível conectar sensores manualmente por pessoas técnicas. Nós demonstramos como os sensores de conexão podem ser automatizados em [13]. Além disso, IEEE 1451 [14] e SensorML [15] permitem tornar o processo de automação mais sofisticado.

Entender e manter informações de contexto As informações de contexto sobre sensores precisam ser adquiridas e armazenadas com anotações apropriadas que facilitam a sua recuperação mais tarde. Informações atualizadas, como recursos do sensor, localização, taxa de amostragem, sensores próximos, duração da bateria, etc. precisam ser mantidas. Este conhecimento é necessário para seleccionar sensores apropriados com base no pedido dos usuários.

Extrair informações de contexto de alto nível CA4IOT precisa raciocinar e entender a solicitação do usuário. Por exemplo, como explicado na Figura 3, o CA4IOT deve ser capaz de entender a relação entre poluição ambiental e sensores de baixo nível, como sensores de temperatura e sensores de pH. Isso pode ser conseguido mantendo o conhecimento sobre domínios de aplicativos usando bases de conhecimento. Além disso, essas bases de conhecimento devem

ser capazes de se estender facilmente através do plugin de bases de conhecimento adicionais que contenham conhecimento sobre diferentes domínios de aplicação quando necessário.

Entender o problema do usuário O raciocínio (por exemplo, semântico ou estatístico) é essencial para entender a relação entre os requisitos dos usuários de alto nível e as capacidades dos sensores de baixo nível, como explicado na Figura 3. É necessário um raciocínio adicional para identificar informações de contexto relevantes com base em leitura de sensores e também para gerar novos conhecimentos (por exemplo, ler as coordenadas de localização GPS de dois sensores e decidir se estão nas proximidades).

Preencher a lacuna entre usuário e sensores Existem muitas operações que podem ser aplicadas aos dados do sensor. Uma operação pode ser tão simples quanto a média ou tão complexa quanto a combinação de várias leituras de sensores e calcular uma única leitura ou gerar valores em falta, avaliando os dados históricos dos sensores. Principalmente, as operações de fusão de dados são usadas para gerar novos conhecimentos de contexto.

Gerenciar usuários Trata-se de adquirir, raciocinar e armazenar informações do usuário. Quando os usuários fazem solicitações, o CA4IOT precisa acompanhá-los em termos dos requisitos, formato de saída necessário, restrições adicionais, como taxa de amostragem, volume de dados e assim por diante. Além disso, o CA4IOT precisa fornecer um mecanismo para interagir com os usuários que permitirão definir seus requisitos com facilidade.

5.0.2 Divisão em Camadas

Sua proposta de arquitetura consiste na divisão em quatro camadas que fazem parte do núcleo da plataforma e duas camadas adicionais definidas para controle da interação entre o núcleo da plataforma e o mundo exterior.

Camada de usuário Esta camada não faz parte da camada central. Os usuários podem ser usuários, aplicativos ou serviços humanos. User Oriented Front End (UOFE) é uma parte desta camada e, portanto, não é um componente central na arquitetura CA4IOT.

Dados, semântica e definição de contextos Responsável pela gerenciar usuários. Os componentes pertencem a esta camada são dispatcher de dados, gerente de solicitação e publique / inscreva-se.

Lógica e processamento de dados Essa é a camada mais importante no CA4IOT. É responsável pelo processamento de dados, raciocínio, fusão, geração de conhecimento e armazenamento. Os componentes pertencem a esta camada são registro de contexto, base de conhecimento de contexto, mecanismo de raciocínio, contexto e gerador de descoberta semântica, processo de contexto primário, processador de contexto secundário, registro de provedor de contexto, operador de fusão de dados e repositório de fusão de dados.

Descoberta de semântica e contextos Esta camada é responsável pelo gerenciamento de contextos e descobridores semânticos, que inclui gerar, configurar e armazenar. Os componentes pertencem a esta camada são descobridores de contexto e semântico, gerador de contexto e descobridor semântico e repositório de descoberta de contexto e semântico.

Aquisição dos dados de sensores Esta camada é responsável pela aquisição de dados. Esta camada aparece na maioria das soluções de middleware IoT, de rede de sensores e de gerenciamento de contexto com terminologias diferentes, como wrappers, gateways, manipuladores, proxies, mediadores, etc. Esta camada se comunica com sensores de hardware e software e recupera os dados do sensor no CA4IOT. Os componentes que pertencem a esta camada são wrappers de sensores, repositório de wrapper, gerador de wrapper, repositório local de definição de dispositivo de sensor (SDD) e repositório de nuvem SDD.

Camada de detecção Esta camada representa todos os sensores de software e hardware (físicos e virtuais). Além disso, esta camada não faz parte da arquitetura principal do CA4IOT.

5.0.3 Componentes e Elementos da arquitetura

Descendo mais um nível dentro da arquitetura proposta, Pereira separou as 6 camadas em uma série de componentes e elementos, definindo a responsabilidade de cada um deles dentro da estrutura.

In Section IV, we introduced all the components under each layer.

We have labelled each component (e.g. C1, C2). However, numbering does not reflect the execution or interaction order. Further, there are some other elements that do not belong to CA4IOT architecture but essential to be explained as they are evolve in execution process. These elements are also labelled (e.g. E1, E2).

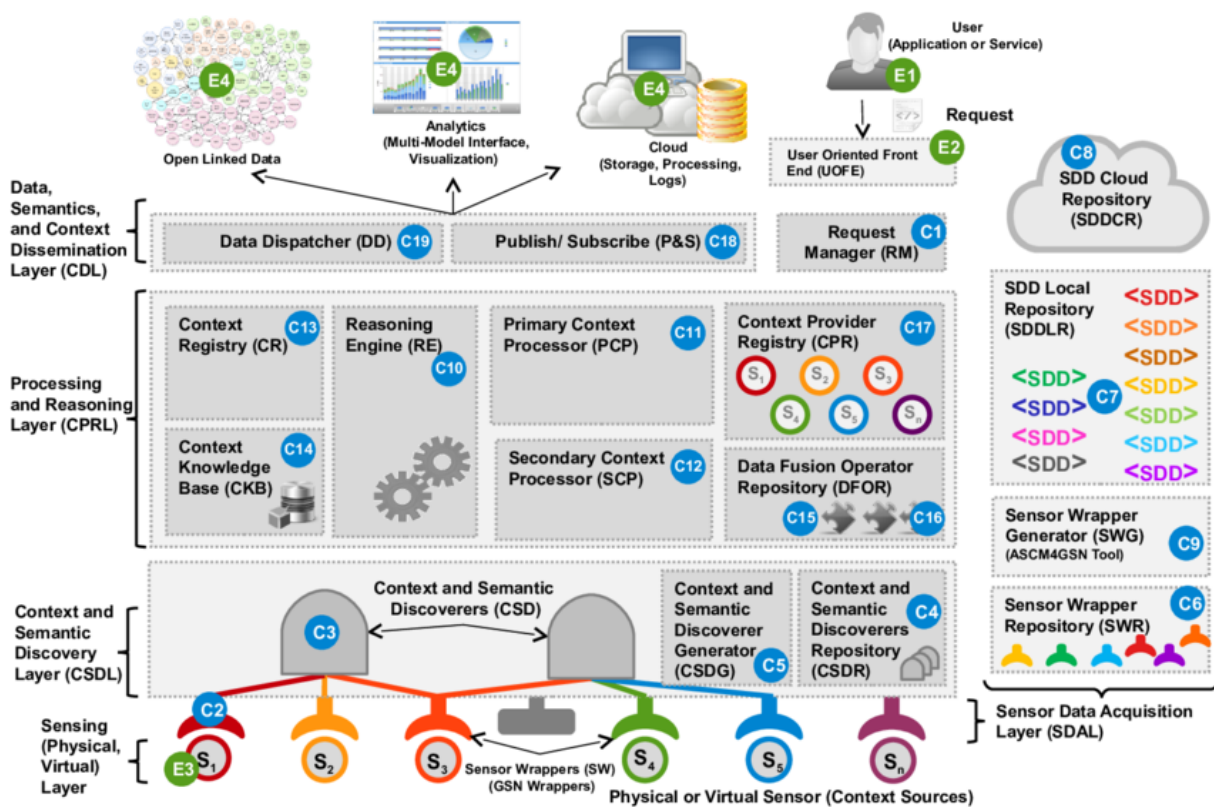


Figura 5.2: Arquitetura CA4IoT em nível de componentes

Componentes

5.0.4 Middleware CASCoM

Em /citeCASCoM, Perera propõe um modelo de composição de sensores baseado em contexto com a finalidade de simplificar o processo de configuração de sensores com a aplicação de mecanismos de filtragem, associações e lógica nas plataformas de IoT com a incorporação de tecnologias de semântica para solução dos desafios propostos. Para isso, ele separa as configurações necessárias ao paradigma de IoT em 2 grandes categorias: configurações em nível de sensores e em nível de sistema. Sendo a primeira categoria focada no controle do comportamento dos sensores e a última na configuração dos parâmetros de software relacionados à atividade dos sensores como: escalonamento de leituras, taxa de amostragem, frequência de comunicação de dados, padrões de comunicação utilizados e protocolos.

De forma mais específica, o modelo identifica, detalha suas composições e configura a atuação dos sensores e o processamento dos dados coletados de acordo com os objetivos dos usuários. Este modelo difere em relação ao proposto no modelo GSN

No middleware IoT existente (por exemplo, GSN), muitos arquivos de configuração e có-

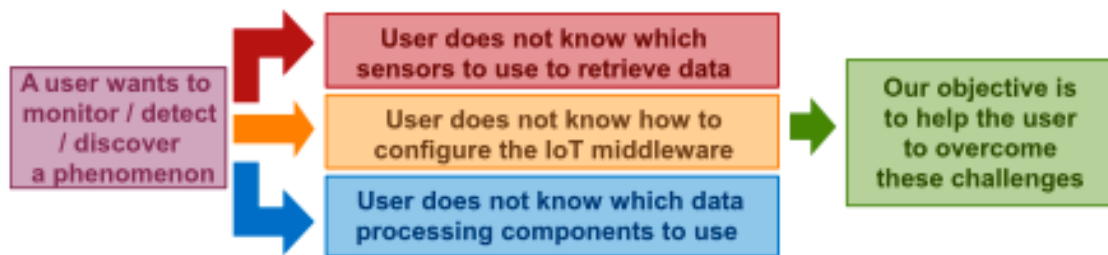


Fig. 3. The problem definition in general.

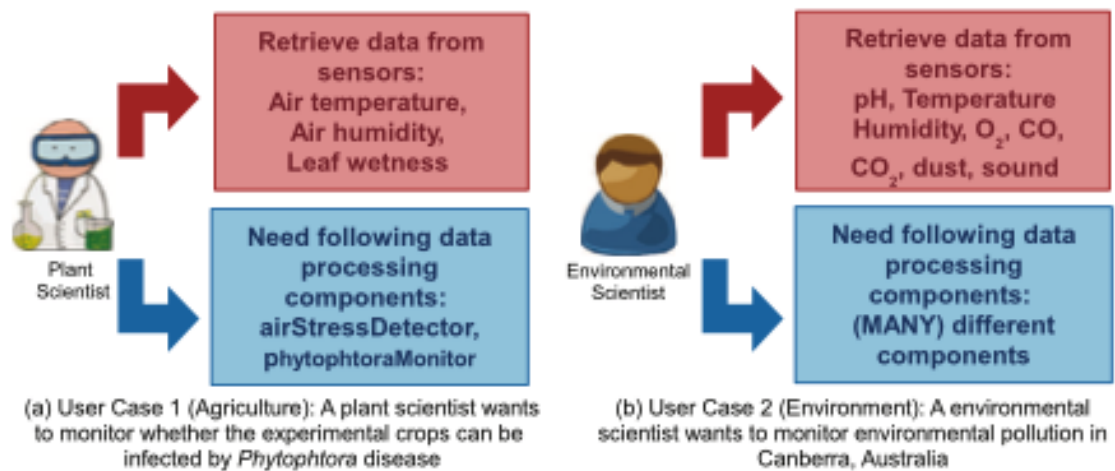


Figura 5.3: Arquitetura CA4IoT

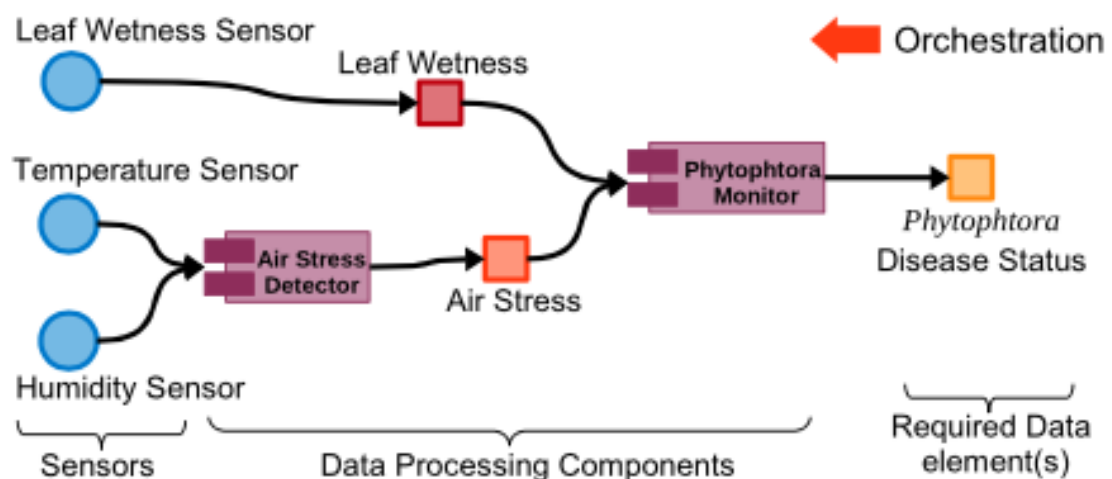


Figura 5.4: Arquitetura CA4IoT

digos de programação precisam ser definidos manualmente pelos usuários (sem qualquer ajuda da GSN). Um modelo ideal de configuração do middleware IoT deve abordar todos os desafios acima mencionados. O modelo de configuração que propomos neste artigo é aplicável a vários

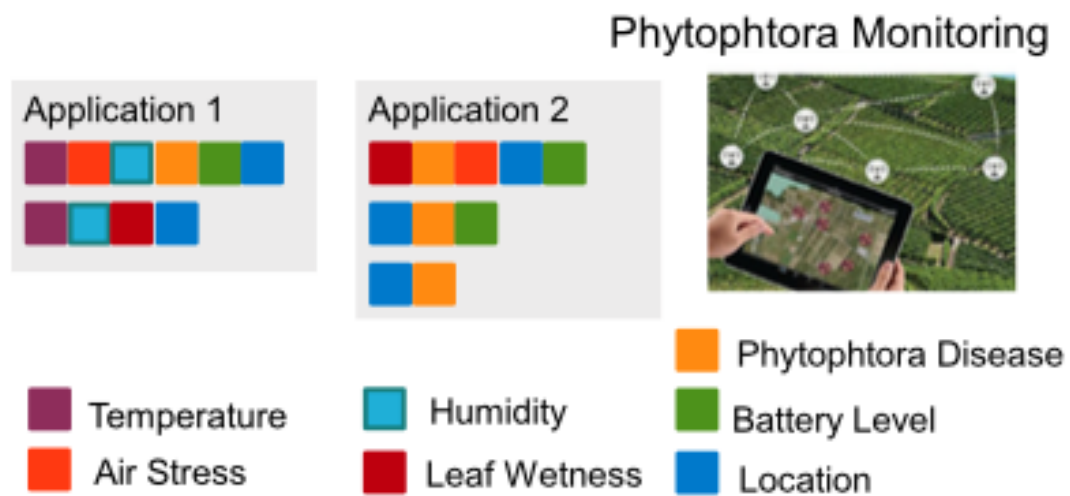


Figura 5.5: Arquitetura CA4IOT

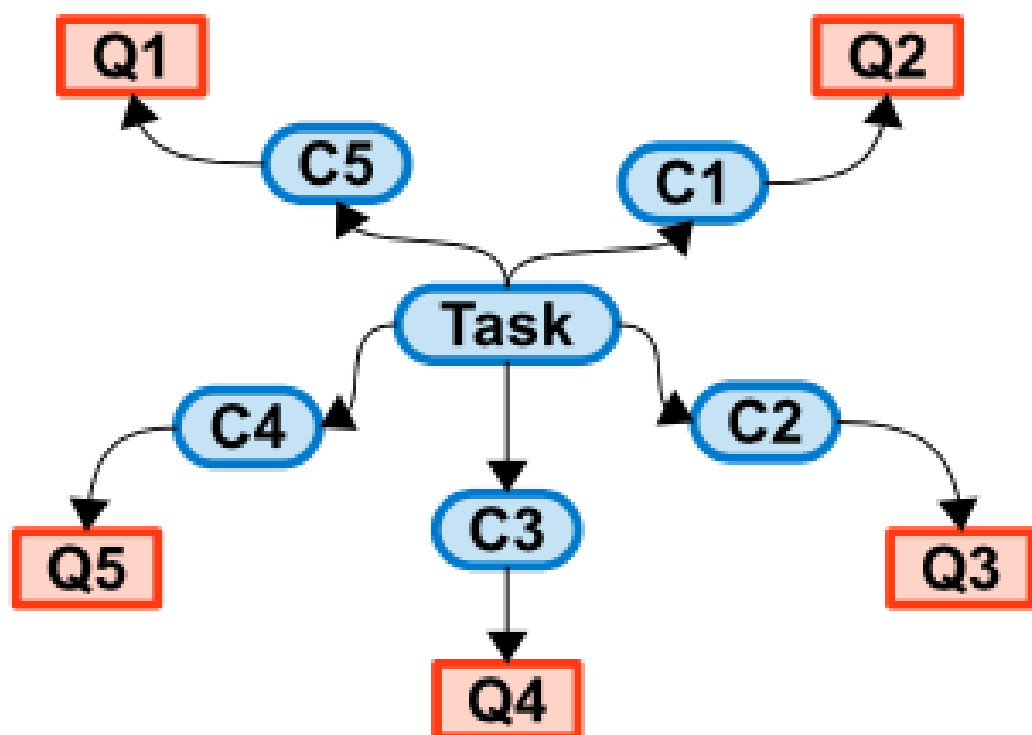


Figura 5.6: Arquitetura CA4IOT

outros paradigmas emergentes, como a detecção como serviço [4].

Seu modelo de solução, foi dividido nas 6 fases seguintes: Understand User Requirements; Select Data Processing Components; Select Sensors; Provide advice and Recommendations; Discover Additional Context; Context-based Cost Optimization. A saída do processo são as aplicações dos dados coletados atendendo às necessidades do usuário como demonstrado pela figura X abaixo.

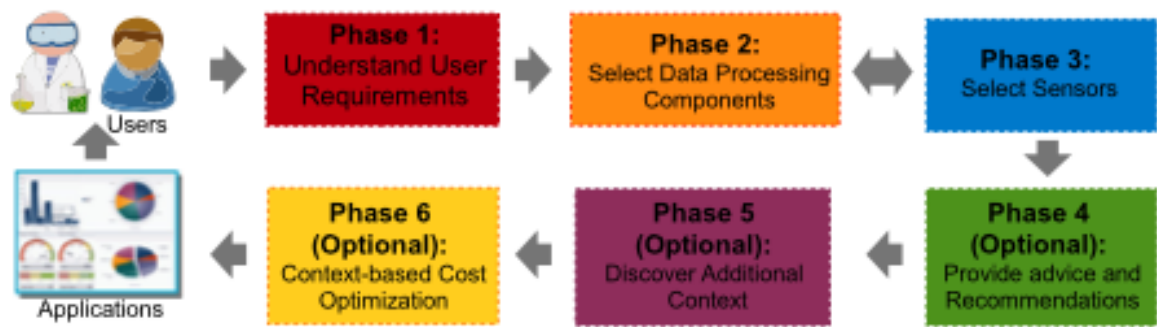


Figura 5.7: Arquitetura CA4IoT

Para criar a semântica de seu modelo, Perera utilizou a essência de 3 diferentes ontologias (QA-TDO, SCO e SSN) para descrição dos diferentes tipos de sensores, componentes de software e domínios de conhecimento.

Na fase 1, os usuários são facilitados com uma interface de usuário gráfica, baseada em uma abordagem de pergunta-resposta (QA), que permite expressar os requisitos do usuário. Os usuários podem responder a todas as questões possíveis. O CASCoM procura e filtra as tarefas que o usuário pode querer executar. Na lista filtrada, os usuários podem selecionar a tarefa desejada. Os detalhes da abordagem de QA são apresentados posteriormente nesta seção. Na fase 2, a CASCoM procura diferentes componentes de programação que permitem gerar o fluxo de dados necessário. Na fase 3, a CASCoM tenta encontrar os sensores que podem ser usados para produzir as entradas exigidas pelos componentes de processamento de dados selecionados. Se o CASCoM não produzir os fluxos de dados exigidos pelos usuários devido a recursos insuficientes (ou seja, a indisponibilidade dos sensores), ele fornecerá conselhos e recomendações sobre implementações futuras de sensores na fase 4. A Fase 5 permite aos usuários capturar informações de contexto adicionais. A informação de contexto adicional que pode ser derivada usando os recursos e conhecimentos disponíveis está listada para ser selecionada. Na fase 6, os usuários recebem uma ou mais soluções³. CASCoM calcula os custos de cada solução ao usar a técnica em 2 [2]. Por padrão, o CASCoM selecionará a solução com o menor custo. No entanto, os usuários podem selecionar os modelos de custo conforme necessário. Finalmente, o CASCoM gera todos os arquivos de configuração e códigos de programa listados na Figura 2 (a). Os dados começam a ser transmitidos logo após.

Resultados: A Figura 5 (a) mostra que o CASCoM permite reduzir consideravelmente o tempo necessário para a configuração do mecanismo de processamento de dados no middleware IoT. Especificamente, CASCoM permitiu que os três tipos de usuários completasse a tarefa dada 50, 80 e 250 vezes mais rápido (respectivamente) em comparação com a abordagem existente.

De acordo com a Figura 5 (b), a abordagem de reflexão Java leva um pouco mais de tempo especialmente ao inicializar. Embora a abordagem de reflexão Java possa adicionar mais flexibilidade ao nosso modelo, a sobrecarga adicional aumenta quando o número de componentes e operação envolvidos aumenta. Os custos gerais podem crescer até um nível inaceitável de forma muito rápida quando o GSN aumenta (por exemplo, mais pedidos de usuários).

De acordo com a Figura 5 (c), mesmo os especialistas em TI que conhecem o GSN podem economizar tempo usando CASCoM até 88 %. Especialmente, o tempo necessário para definir a classe VSD e VS foi significativamente reduzido. Ambos os arquivos podem ser gerados pelo CASCoM de forma autônoma em um segundo, mesmo para cenários complexos. No entanto, o tempo necessário para encontrar componentes e sensores de processamento de dados (e wrappers) depende do tamanho do modelo de dados semânticos. A Figura 5 (d) mostra como o tempo de processamento total variaria dependendo do tamanho do modelo de dados semânticos. Aproximadamente, um modelo semântico com 10.000 descrições de sensores e 10.000 componentes de processamento de dados pode ser processado para encontrar soluções para um determinado pedido de usuário em menos de um minuto. No entanto, a maior parte do tempo é tomada para ler o modelo de dados.

O processo de configuração real além da leitura do modelo de dados leva apenas 4 segundos e ele aumenta ligeiramente quando o tamanho do modelo aumenta.

Mostramos que é possível oferecer um modelo de configuração sofisticado para suportar especialistas que não sejam de TI. As tecnologias semânticas são amplamente utilizadas para suportar este modelo. Usando nossa implementação de prova de conceito, os especialistas em TI e não-TI conseguiram configurar o GSN em muito menos tempo. No futuro, planejamos ampliar nosso modelo de configuração em nível de sensor. Para conseguir isso, desenvolveremos um modelo que pode ser usado para configurar sensores de forma autônoma sem intervenção humana em ambientes inteligentes altamente dinâmicos no paradigma IoT.

Parte III

Proposta de trabalho

Capítulo 6

PROPOSTA

Este capítulo apresenta e analisa uma proposta de arquitetura para facilitar a implementação da internet das coisas na automação de modelos agrícolas existentes (com foco na agricultura familiar) reduzindo os elevados custos desses procedimentos, atualmente responsáveis por inibir sua disseminação. Nessa proposta, serão utilizadas diversas tecnologias já consolidadas pela literatura para resolução de diversos desafios e cenários dentro da Internet das Coisas.

6.1 Arquitetura Proposta

6.1.1 Descrição

6.2 Modelo Utilizado

6.2.1 Modelo de conhecimento

6.2.2 Definição dos Objetos

6.2.3 Estrutura em Grafo

6.2.4 Processamento dos dados

6.2.5 Processamento dos dados

Modelo Básico

Modelo Estático Complexo

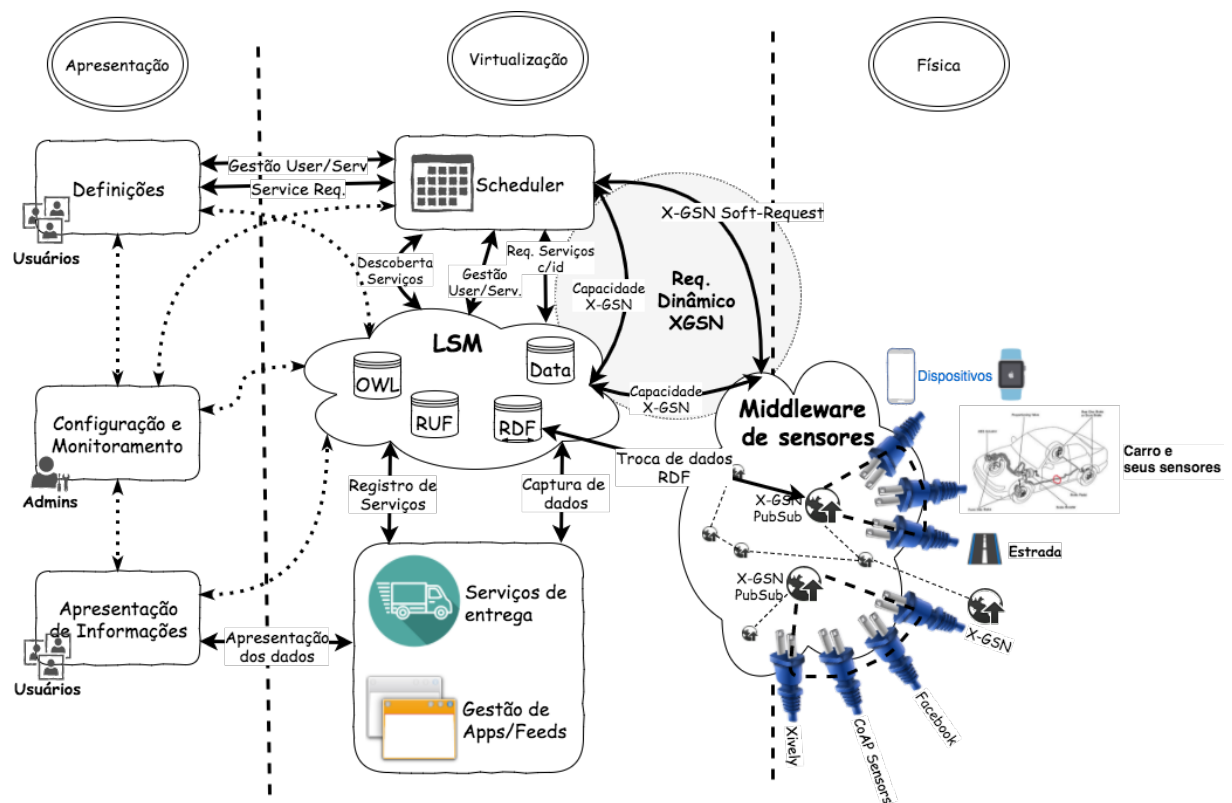


Figura 6.1: Arquitetura do OpenIoT Full extendida

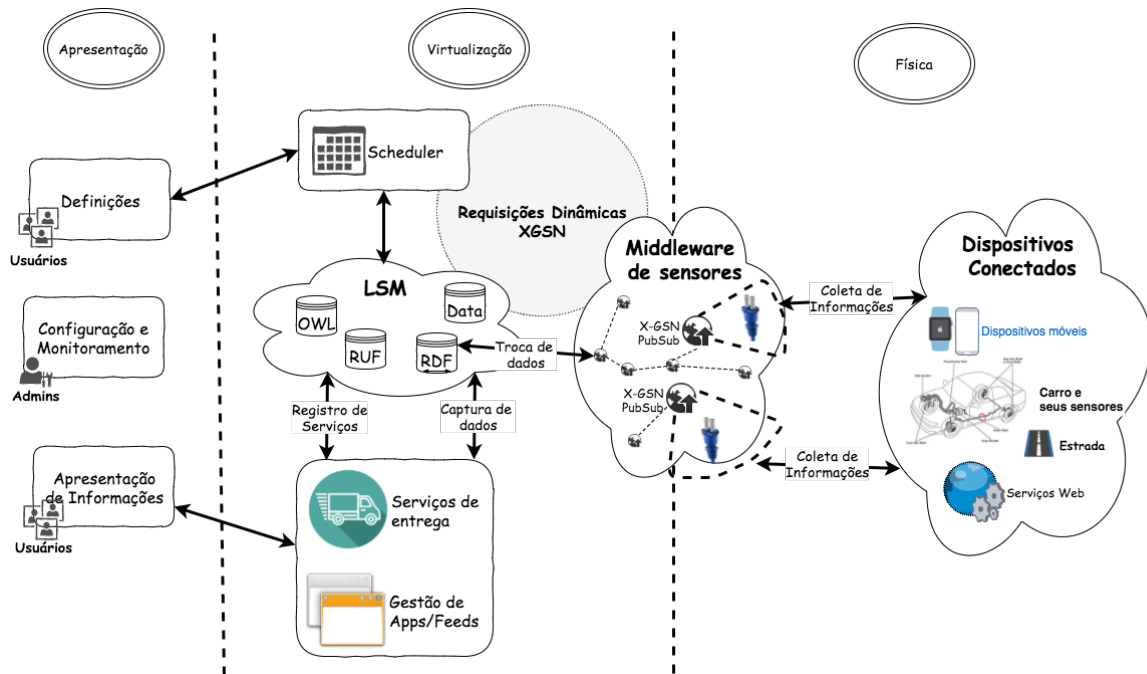


Figura 6.2: Arquitetura do OpenIoT Light extendida

Modelo Dinâmico Complexo

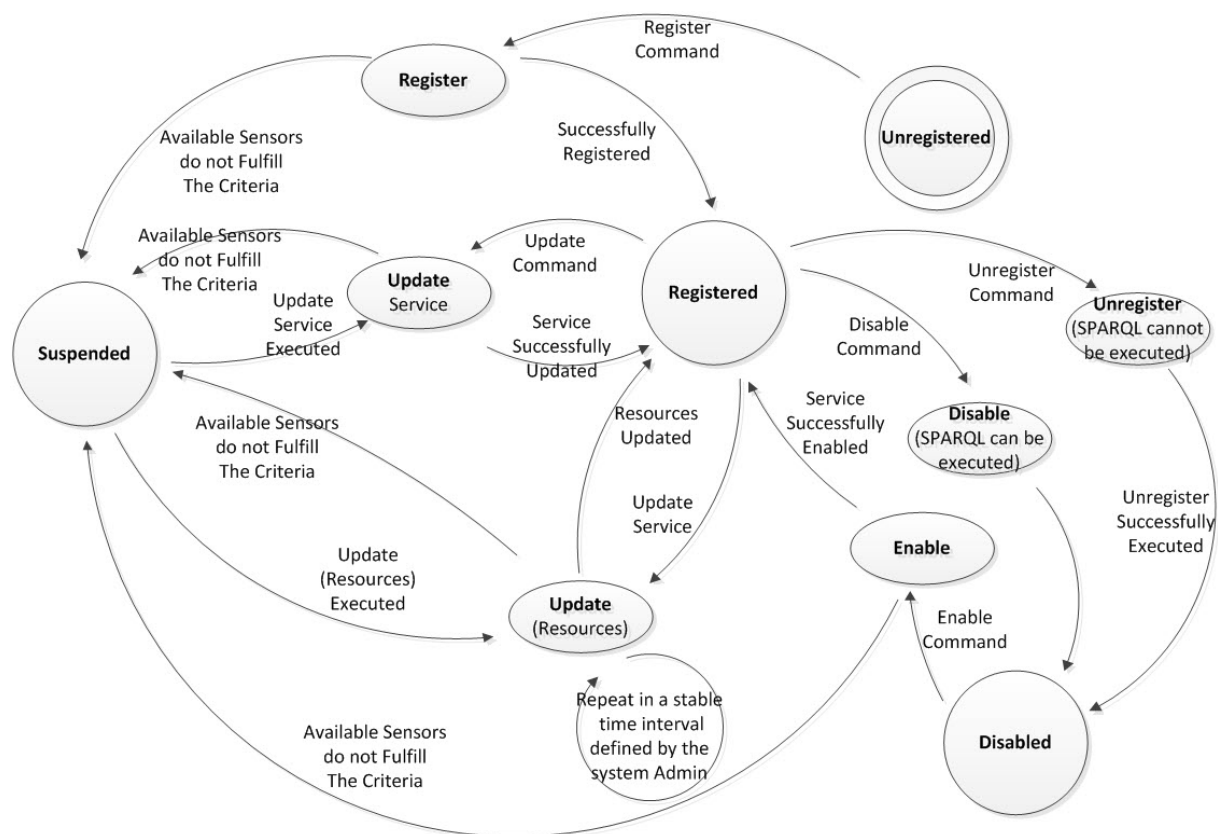


Figura 6.3: Diagrama de Stados do Scheduler estendido

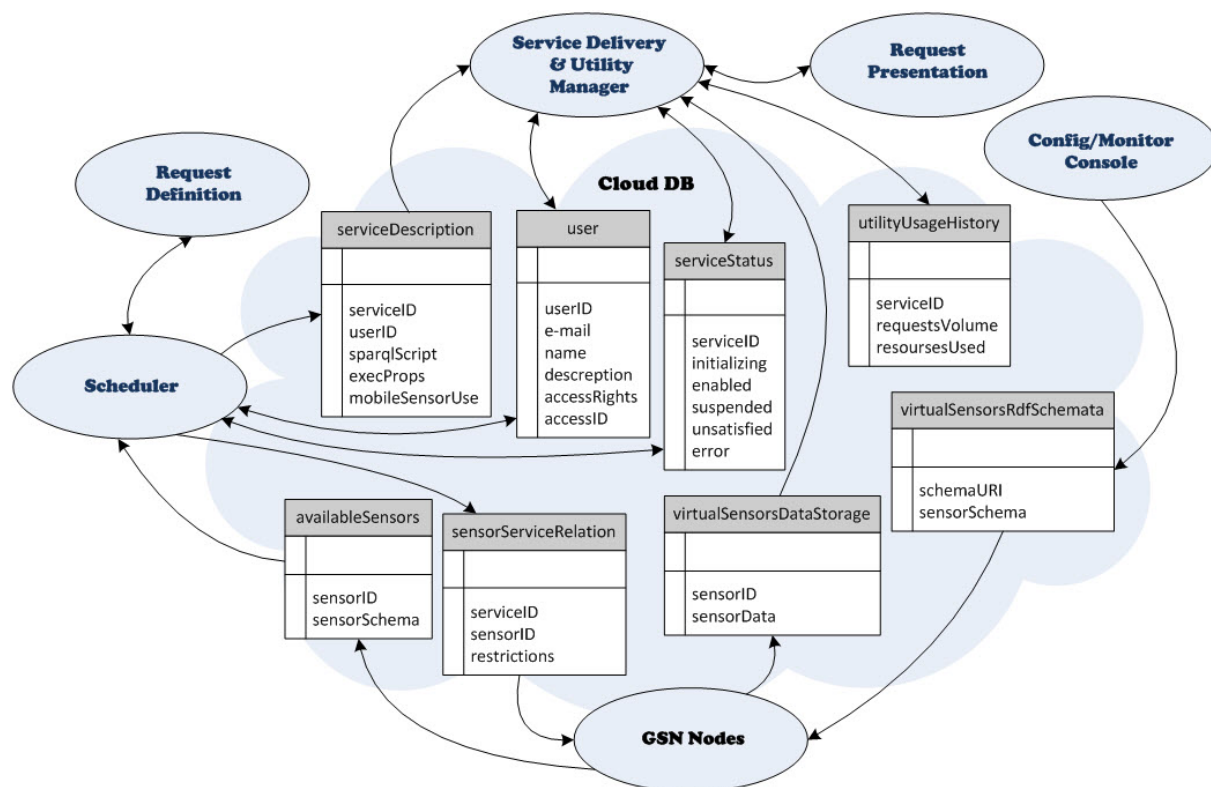


Figura 6.4: Relacionamento de dados estendido

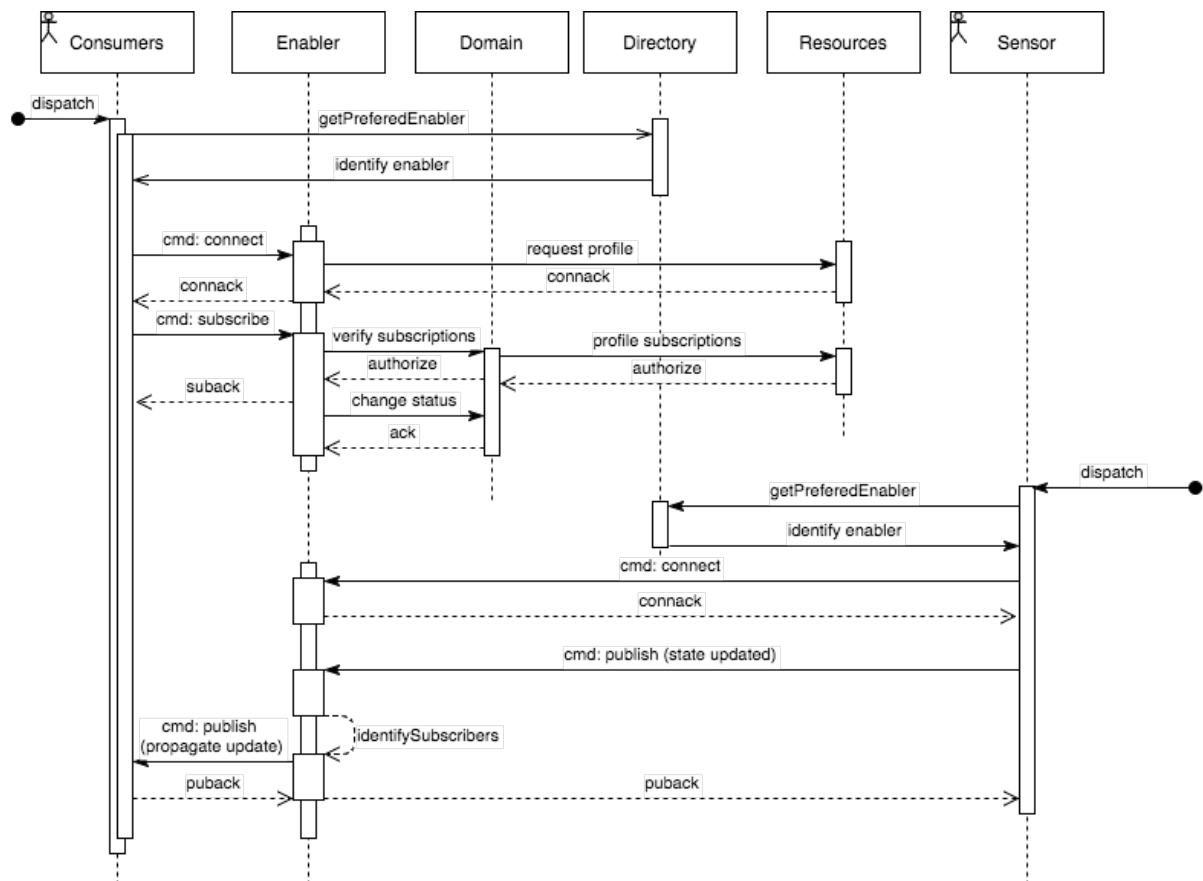


Figura 6.5: Diagrama de sequência 1

6.3 Social Virtual Objects

6.4 Social Virtual Objects

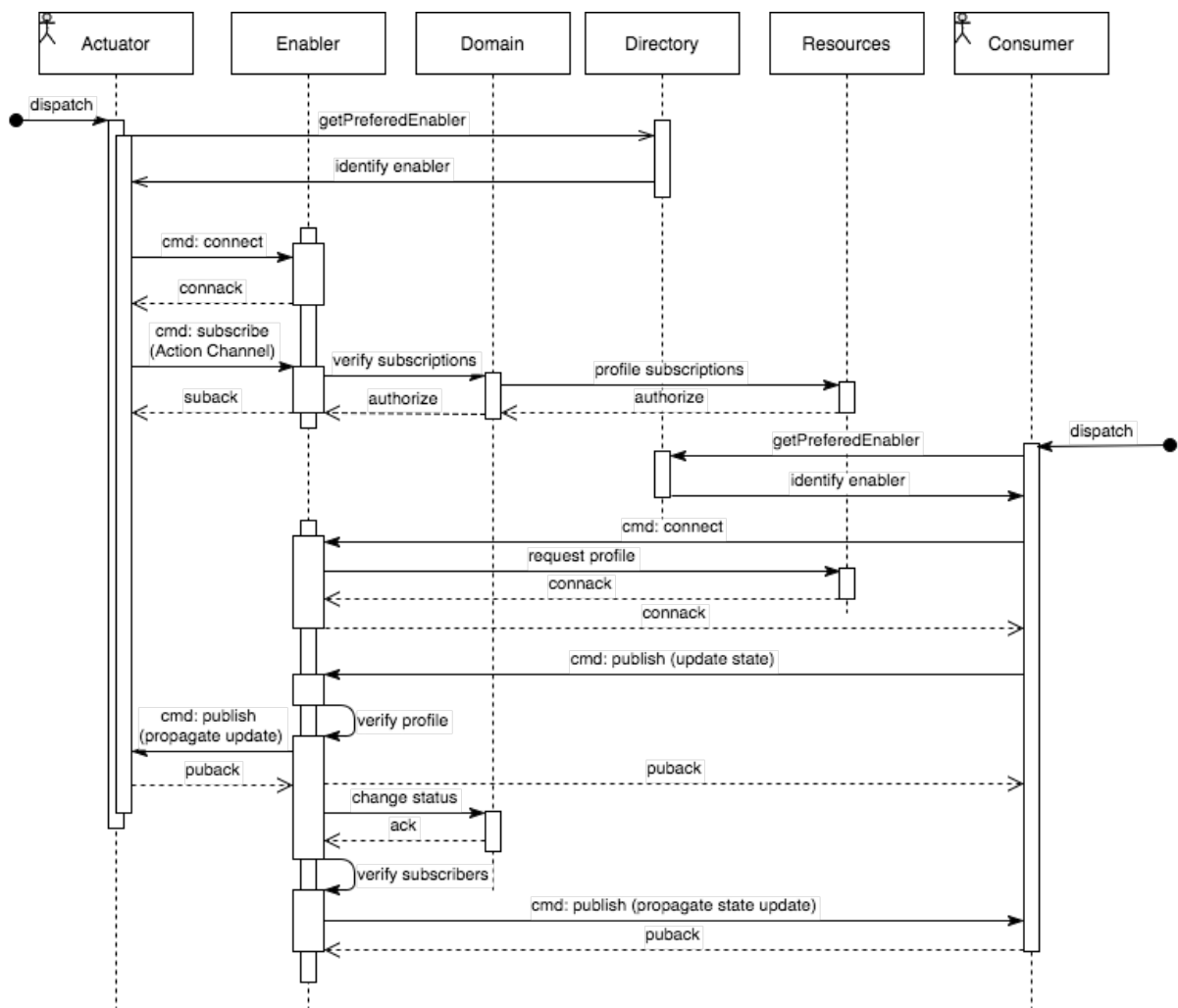


Figura 6.6: Diagrama de sequência 2

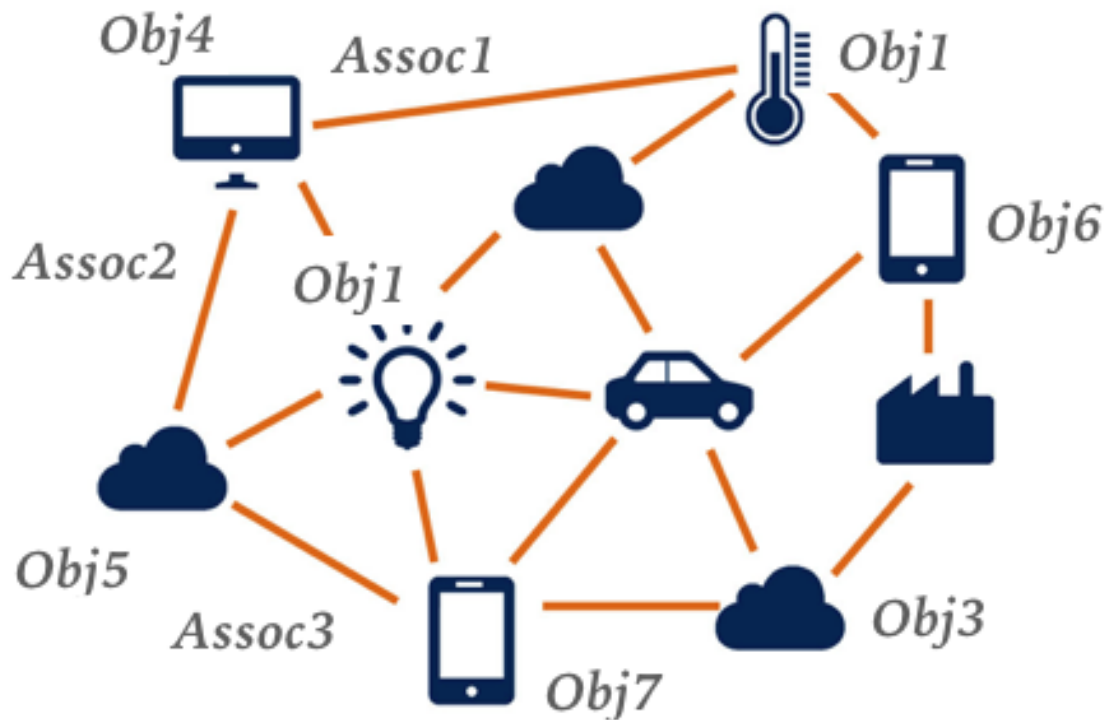


Figura 6.7: Exemplo de Contexto

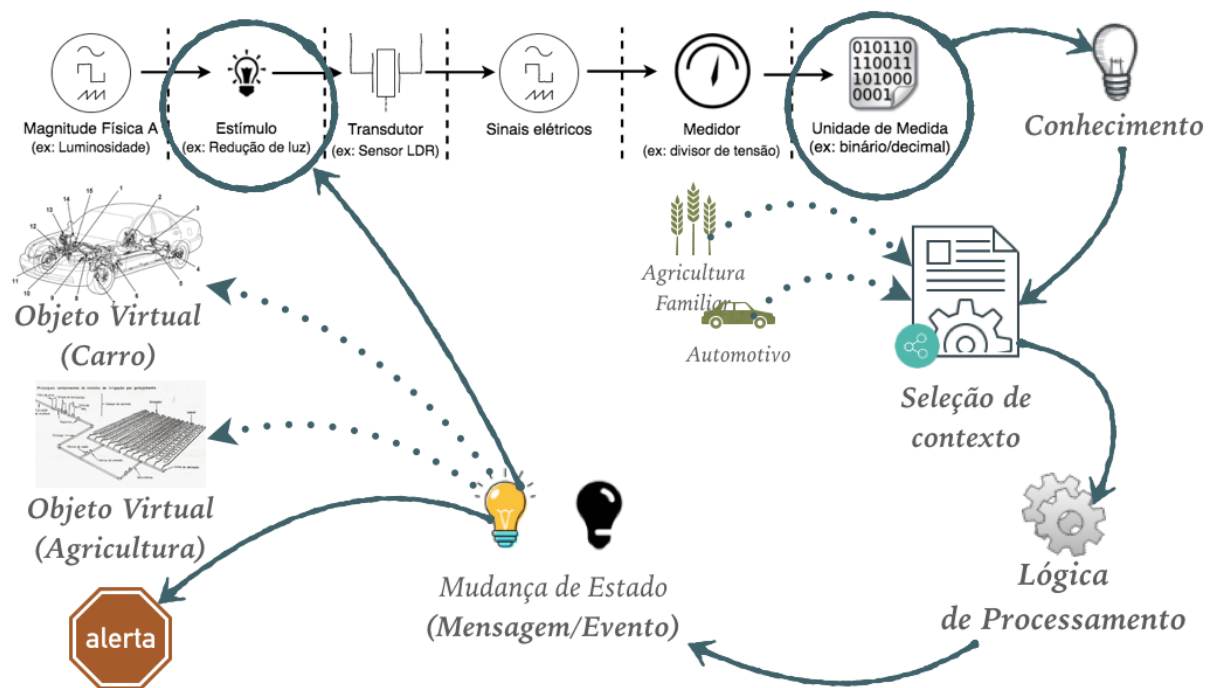


Figura 6.8: Representação do ciclo do Conhecimento

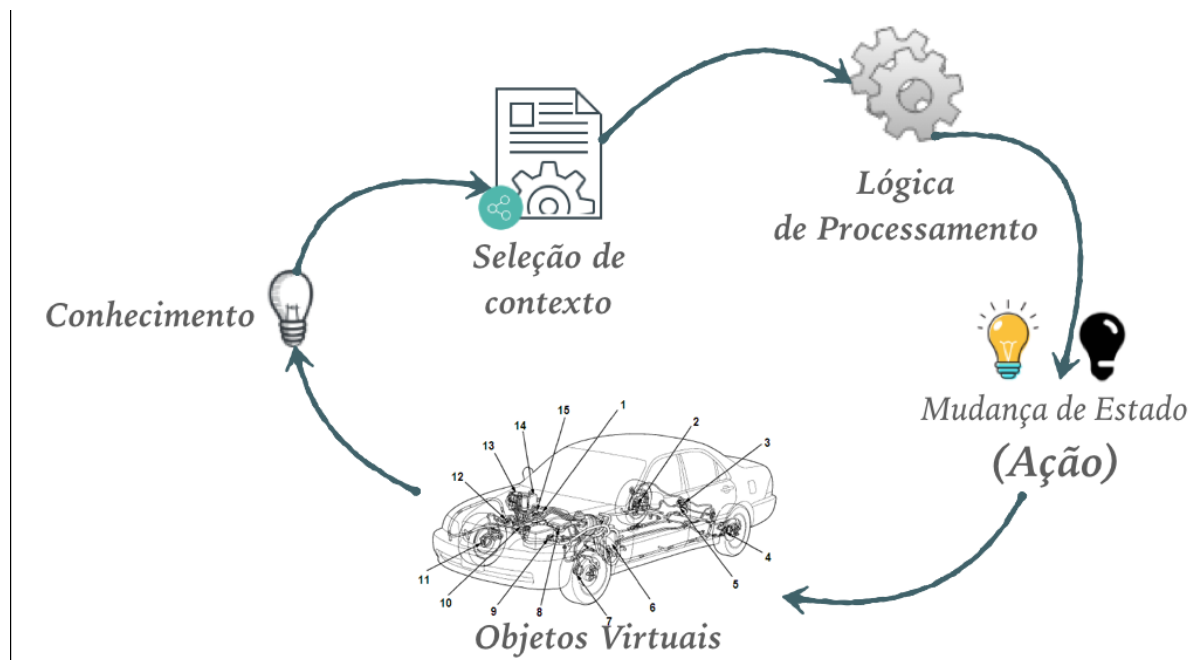


Figura 6.9: Ciclo de Vida dos Objetos Virtuais

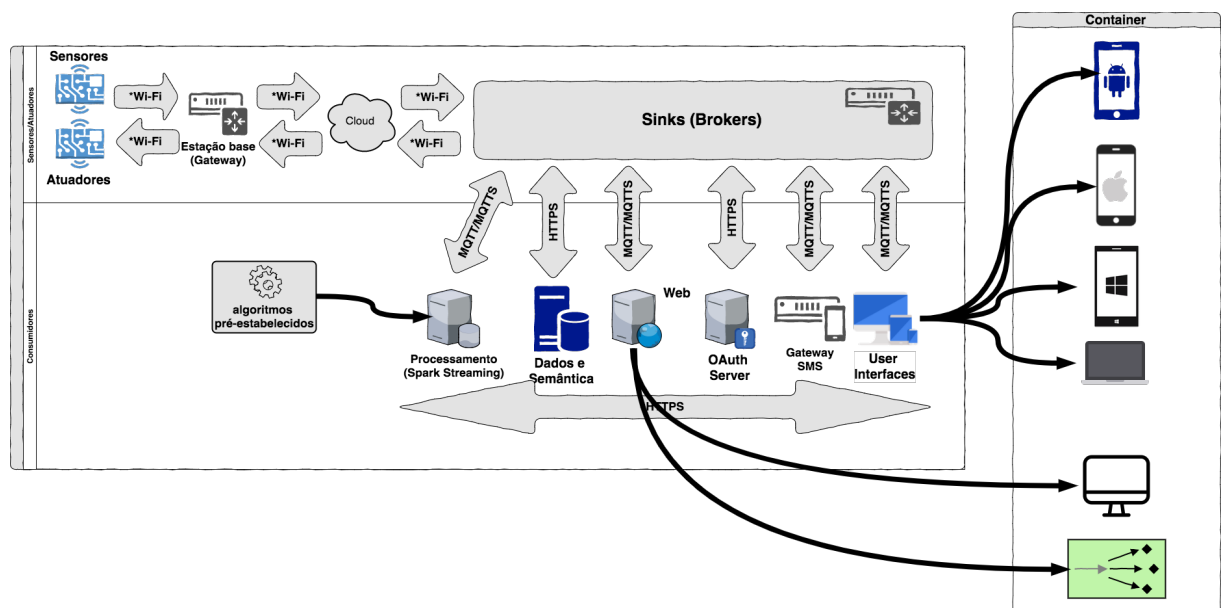


Figura 6.10: Arquitetura 2

REFERÊNCIAS

CHEN, Y. et al. Time-Reversal Wireless Paradigm for Green Internet of Things: An Overview. In: *IEEE Internet of Things Journal*. [s.n.], 2014. v. 1, n. 1, p. 1?1. ISSN 2327-4662. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6750095>>.

WEISER, M. The computer for the 21st century. *Scientific American*, v. 265, n. 3, p. 94–104, 1991. ISSN 1559-1662. Disponível em: <<http://www.nature.com/scientificamerican/journal/v265/n3/full/scientificamerican0991-94.html>>.

WEISER, M. Hot topics-ubiquitous computing. *Computer*, v. 26, n. 10, p. 71–72, Oct 1993. ISSN 0018-9162.

WEISER, M.; BROWN, J. S. The coming age of calm technology. In: DENNING, P. J.; METCALFE, R. M. (Ed.). *Beyond Calculation*. New York, NY, USA: Copernicus, 1997. cap. The Coming Age of Calm Technology, p. 75–85. ISBN 0-38794932-1. Disponível em: <<http://dl.acm.org/citation.cfm?id=504928.504934>>.

WEISER, M.; GOLD, R.; BROWN, J. S. The origins of ubiquitous computing research at parc in the late 1980s. *IBM Systems Journal*, v. 38, n. 4, p. 693–696, 1999. ISSN 0018-8670.

GLOSSÁRIO

EPFL – *École Polytechnique Fédérale de Lausanne*

IERC – *European Research Cluster on the Internet of Things*

LSM – *Linked Sensed Middleware*

RDF – *Resource Description Framework*

SSN – *Semantic Sensor Networks*