UNIVERSIDADE FEDERAL DO CEARÁ

CENTRO DE CIÊNCIAS

DEPARTAMENTO DE COMPUTAÇÃO

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

MESTRADO ACADÊMICO EM CIÊNCIA DA COMPUTAÇÃO

FRANCISCO LUCIANO CASTRO MARTINS JÚNIOR

UMA ARQUITETURA FLEXÍVEL DE SEGURANÇA PARA COMPARTILHAMENTO DE CONTEXTO EM INTERNET DAS COISAS

FORTALEZA

2022

FRANCISCO LUCIANO CASTRO MARTINS JÚNIOR

UMA ARQUITETURA FLEXÍVEL DE SEGURANÇA PARA COMPARTILHAMENTO DE CONTEXTO EM INTERNET DAS COISAS

Dissertação apresentada ao Curso de Mestrado Acadêmico em Ciência da Computação do Programa de Pós-Graduação em Ciência da Computação do Centro de Ciências da Universidade Federal do Ceará, como requisito parcial à obtenção do título de mestre em Ciência da Computação. Área de Concentração: Sistemas de Informação.

Orientador: Prof. Dr. Arthur de Castro Callado.

FORTALEZA

2022

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Sistema de Bibliotecas
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

M343a Martins Júnior, Francisco Luciano Castro.
Uma arquitetura flexível de segurança para compartilhamento de contexto em Internet das Coisas / Francisco Luciano Castro Martins Júnior. – 2022.
97 f. : il. color.

Dissertação (mestrado) – Universidade Federal do Ceará, Centro de Ciências, Programa de Pós-Graduação em Ciência da Computação, Fortaleza, 2022.
Orientação: Prof. Dr. Arthur de Castro Callado.

1. Internet das coisas. 2. Segurança. 3. Privacidade. 4. Aplicações sensíveis ao contexto. I. Título.
CDD 005

FRANCISCO LUCIANO CASTRO MARTINS JÚNIOR

UMA ARQUITETURA FLEXÍVEL DE SEGURANÇA PARA COMPARTILHAMENTO DE CONTEXTO EM INTERNET DAS COISAS

Dissertação apresentada ao Curso de Mestrado Acadêmico em Ciência da Computação do Programa de Pós-Graduação em Ciência da Computação do Centro de Ciências da Universidade Federal do Ceará, como requisito parcial à obtenção do título de mestre em Ciência da Computação. Área de Concentração: Sistemas de Informação.

Aprovada em: 25/11/2022.

BANCA EXAMINADORA

Prof. Dr. Arthur de Castro Callado (Orientador)
Universidade Federal do Ceará (UFC)

Prof. Dr. José Neuman de Souza
Universidade Federal do Ceará (UFC)

Prof. Dr. Eduardo James Pereira Souto
Universidade Federal do Amazonas (UFAM)

A meu avô, José Bezerra de Mesquita (*in memorian*), que sempre aplaudiu minhas conquistas e me acolheu nas várias batalhas até elas mesmo sem entender. Seu abraço e sua *"bença"* fizeram falta nesse processo e sempre farão. Obrigado por tudo e por tanto.

AGRADECIMENTOS

Em um diálogo de um seus livros, o autor norte-americano Ernest Hemingway expôs o pensamento: *quem está ao nosso lado nas trincheiras importa mais que a própria guerra*. Quero expressar aqui, então, meus agradecimentos.

A Deus, por ter me guiado nessa jornada, me concedendo o necessário para chegar até aqui e me protegendo nos tempos difíceis atravessados.

Minha gratidão à minha família. Aos meus pais, Fátima e Luciano, pelo amor, ensinamentos, cuidados, orações e por terem confiado em mim e feito sacrifícios que me permitiram voar. Espero dar orgulho a vocês e fazê-los felizes a cada dia. A minhas irmãs, pelo cuidado afetuoso, torcida confiante e palavras de apoio. Ao meu sobrinho, Levi, por me motivar a ser uma pessoa melhor e pelo afeto que tem por mim.

Ao meu companheiro, Ciro Secundino, por não ter soltado minha mão em meio à loucura que foi esse processo. Seu cuidado, acolhimento, paciência, compreensão e afeto demonstrados em gestos, silêncios e palavras foram determinantes.

Aos meus colegas de pós-graduação, Augusto, Joseane, Valdenir, João Batista, Andreza, Pedro e outros tantos, pelas partilhas e parcerias feitas nessa formação. Em especial, minha gratidão e carinho, aos amigos Belmondo, Rodrigo e Sayonara, sem vocês talvez até desse certo, mas foi um prazer dividir esse caminho com vocês, tornando-o mais leve (ou menos pesado) e transformando as aflições da pós-graduação em piadas e em uma amizade que permanecerá.

Aos amigos de outras frentes, Andreia, Kilbert, Karise, Lina, Mirelle, Eduardo, Rebeca e Tatiane, entre tantos outros, vocês fizeram mais por mim do que podem imaginar e contribuíram de muitos modos para que eu perseverasse. Em especial, ao amigo Adonias, parceiro de muitos projetos, mas também na vida, pelos momentos de incentivo, consolo e inspiração nessa jornada.

Aos que colaboraram de uma forma ou de outra com este trabalho. Para citar alguns: Caleb, Pedro e Sidney, que colaboraram em parte do desenvolvimento da prova de conceito; Antônia de Jesus, que auxiliou na revisão do *abstract* e com sua amizade agitada e cheia de energia, e Thiago Vidotto, pelos ensinamentos sobre ciência e escrita científica.

Ao IFCE *campus* Tauá, por ter proporcionado condições sempre que possível para que eu cumprisse as atividades do mestrado. Em especial, ao diretor de ensino Alan Sombra, pelo suporte e momentos de compreensão. Aos meus alunos, que torceram por mim e me

incentivaram muitas vezes e de muitas formas.

O ambiente da pós-graduação pode ser muito solitário às vezes. Então, minha gratidão ao Sr. Hudson, recepcionista do GReaT, pela cordialidade e auxílios, e a todos que fazem a Cantina da Química, pela educação de sempre e por terem fornecido café suficiente para os momentos de estudo e aflição, por vezes acompanhado de um sorriso que fazia a diferença em muitos dias.

A todos que compõem o MDCC. Aos professores pelos múltiplos conhecimentos trocados e construídos em sala e fora dela, em especial aos professores Victor Campos e Emanuel Bezerra. Carinhosamente, aos membros da secretaria do programa, Gláucia e Jonatas, pela atenção de sempre (foram muitos e-mails, eu reconheço).

Ao meu orientador, Prof. Arthur Callado, pela parceria, confiança e partilhas durante esse período de formação. Seu olhar e sensibilidade em enxergar o ser humano por trás do aluno me fizeram ter força, inspiração e determinação para concluir essa etapa. Lembrarei sempre do professor competente e esforçado, porém, mais ainda do ser humano gigante que demonstrou ser.

Aos membros da banca, professores Neuman Souza e Eduardo Souto, por terem aceitado o convite e pelas colaborações valorosas dadas nas etapas de qualificação.

"Senhor, fazei-me instrumento de vossa paz.
Onde houver ódio, que eu leve o amor.
Onde houver ofensa, que eu leve o perdão.
Onde houver discórdia, que eu leve a união.
Onde houver dúvidas, que eu leve a fé.
Onde houver erro, que eu leve a verdade.
Onde houver desespero, que eu leve a esperança.
Onde houver tristeza, que eu leve a alegria.
Onde houver trevas, que eu leve a luz."
(São Francisco de Assis)

RESUMO

Apesar da evolução do paradigma *Internet of Things* (IoT) nas últimas décadas, preocupações com segurança e privacidade ainda são uma necessidade e um desafio, motivado pela heterogeneidade, pela grande quantidade de dados e de dispositivos e pelas restrições destes, entre outros fatores. Adicionalmente, algumas aplicações possuem demandas específicas, como é o caso das aplicações sensíveis ao contexto, que podem usar informações sensíveis de diferentes entidades para fornecer contexto para várias aplicações. Assim, é necessário que novos recursos de segurança sejam pensados. Tecnologias como funções de redes virtualizadas, computação em nuvem, névoa e de borda têm sido exploradas para criação de estruturas de segurança. Em outra vertente, controle de acesso baseado em atributos (ABAC) e a adição de informações de contexto nas decisões de segurança vêm se demonstrando promissoras para IoT. Este trabalho apresenta uma arquitetura de implementação flexível denominada *Flexibility for Context-Aware Applications Security in IoT* (FCAAS-IoT) que visa prover confidencialidade e privacidade para compartilhamento de contexto em ambientes IoT. As funções de segurança definidas para os módulos permitem o controle de acesso às informações e a encriptação destas antes do envio, com base no modelo ABAC e no uso de contexto da solicitação para escolha de algoritmos e chaves criptográficas, mediante definição e análise de políticas, conferindo um potencial de adaptabilidade a diversos cenários de solicitação de contexto. Os resultados obtidos com a prova de conceito demonstraram o funcionamento correto das funções projetadas para diferentes variações de contextos e atributos e que a atuação da arquitetura aumenta o tempo de resposta no fornecimento de informações. Embora essa diferença cresça para contextos maiores, o acréscimo não é tão alto e pode ser reduzido com adequações nas políticas e na implantação dos módulos de acordo com o cenário de atuação, graças à flexibilidade da arquitetura. Extensões da pesquisa serão realizadas para verificar o funcionamento da FCAAS-IoT em diferentes implementações e aprimorar o uso de contexto de solicitação, através da consolidação de um modelo de políticas e inserção de aprendizagem de máquina no processo.

Palavras-chave: internet das coisas; segurança; privacidade; aplicações sensíveis ao contexto; ABAC; contexto de segurança.

ABSTRACT

Despite the evolution of the Internet of Things (IoT) paradigm in recent decades, concerns about security and privacy are still a necessity and a challenge, due to the heterogeneity, large amount of data and devices and their restrictions, among other aspects. Additionally, some applications have specific demands, such as context-aware applications, which can use sensitive information from different entities to provide context for various applications. Thus, it is necessary that new security features be developed. Technologies such as virtualized network functions, cloud, fog and edge computing have been explored in order to create security mechanisms. In another aspect, attribute-based access control (ABAC) and the use of context information in security decisions have shown to be promising for IoT. This work presents a flexible implementation architecture called Flexibility for Context-Aware Applications Security in IoT (FCAAS-IoT) that aims to provide confidentiality and privacy for context sharing in IoT environments. The security functions defined for the modules allow controlling access to information and encrypting it before sending it, based on the ABAC model and using the request context to choose algorithms and cryptographic keys, by defining and analyzing policies, conferring a potential for adaptability to various context request scenarios. The results obtained with the proof of concept demonstrated the correct functioning of the functions designed for different variations of contexts and attributes, and that the architecture performance increases the response time in providing information. Although this difference grows for larger contexts, the addition is not so high and can be reduced with adjustments in policies and in the implementation of modules according to the operating scenario, thanks to the flexibility of the architecture. Extensions of the research will be carried out to verify the functioning of FCAAS-IoT in different implementations and improve the use of request context, through the consolidation of a policy model and insertion of machine learning in the process.

Keywords: internet of things; security; privacy; context-aware applications; ABAC; security context.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

ABAC	*Attribute-Based Access Control*
ABE	*Attribute-Based Encryption*
AM	*Authorization Manager*
CAS	*Context-Aware Security*
CB	*Context Broker*
CM	*Context Manager*
CP-ABE	*Ciphertext-Policy Attribute-Based Encryption*
DAC	*Discretionary Access Control*
DoS	*Denial of Service*
EA	*Encryption Agent*
ETSI	*European Telecommunications Standards Institute*
IaaS	*Infrastructure as a Service*
IDSs	*Intrusion Detection System*
IoT	*Internet of Things*
IPSs	*Intrusion Prevention System*
LoRa	*Long Range*
MAC	*Mandatory Access Control*
MQTT	*Message Queuing Telemetry Transport*
NFV	*Network Functions Virtualization*
NFV-MANO	*NFV Management and Orchestration*
NFVI	*Network Function Virtualization Infrastructure*
PaaS	*Platform as a Service*
PAP	*Policy Administration Point*
PDA	*Policy Decision Agent*
PDP	*Policy Decision Point*
PEP	*Policy Enforcement Point*
PIP	*Policy Information Point*
PMA	*Policy Management Agent*
QoS	*Quality of Service*
RBAC	*Role-Based Access Control*

SaaS	*Software as a Service*
SDN	*Software Defined Network*
SIB	*Security Information Base*
UML	*Unified Modeling Language*
VMs	Virtual Machines
VNF	*Virtualized Network Function*
XACML	*eXtensible Access Control Markup Language*

SUMÁRIO

1 INTRODUÇÃO

O paradigma Internet das Coisas, do inglês *Internet of Things* (IoT), tem recebido bastante atenção nas últimas décadas, tanto por parte da indústria quanto da academia. Na IoT, objetos dotados de sensores e interfaces de comunicação são capazes de coletar dados do ambiente e compartilhá-los com outros dispositivos a fim de serem usados em diversas aplicações, podendo, em alguns casos, executar ações no mundo físico (MIORANDI *et al.*, 2012). Essa capacidade abre espaço para a implementação de aplicações em diversos domínios, como saúde, transporte, gerenciamento de energia, logística e automação residencial (ATZORI *et al.*, 2010; GUBBI *et al.*, 2013).

Apesar do crescimento e dos avanços obtidos pela IoT nos últimos anos, questões relacionadas à segurança continuam sendo ao mesmo tempo uma necessidade e um desafio. A capacidade dos dispositivos IoT de coletarem e compartilharem informações, possibilita o monitoramento de atividades de um usuário, refletindo seus comportamentos, o que sinaliza preocupações com privacidade (YANG *et al.*, 2017). Ou seja, por serem consideradas informações sensíveis sobre o usuário, deve haver formas de controlar quem pode ter acesso a elas, bem como as finalidades de utilização.

Por outro lado, como IoT prevê a conexão de um grande número de dispositivos, ocorre também o aumento das superfícies de ataques que visem desde o roubo de informações até a alteração do comportamento dos dispositivos (ALABA *et al.*, 2017). Portanto, é necessário garantir o cumprimento de requisitos de segurança, como confidencialidade, privacidade, disponibilidade e integridade para usuários, dados e aplicações.

Embora seja um requisito fundamental para a consolidação e aceitação da IoT, prover segurança dentro desse paradigma não é uma tarefa fácil. Conceitualmente, ambientes IoT são caracterizados por heterogeneidade, mobilidade e grande quantidade de dispositivos, muitos dos quais possuem restrições de recursos computacionais e alimentação energética. Tais características fazem com que soluções de segurança já consolidas em ambientes tradicionais, como algoritmos de criptografia, *firewalls* e mecanismos de autenticação não sejam completamente adequados aos ambientes IoT, o que reduz sua efetividade ou impossibilita sua utilização (FARRIS *et al.*, 2019).

Adicionalmente, alguns sistemas podem ter requisitos e particularidades de segurança específicos que devem ser considerados. Esse é o caso das aplicações sensíveis ao contexto, já bastante difundidas na computação ubíqua e pervasiva. Tais aplicações utilizam as informações

coletadas sobre uma entidade (usuário, ambiente, sistema etc.) para prover algum tipo de serviço, como exibição de dados ou execução de alguma operação (DEY, 2001). Um contexto utilizado por uma aplicação pode conter dados sensíveis vinculados a uma entidade, implicando uma maior preocupação no compartilhamento desse tipo de informação em comparação a dados comuns (MATOS *et al.*, 2018).

No âmbito da IoT, o compartilhamento de contexto de forma segura torna-se ainda mais desafiador, visto que uma fonte de contexto, como um sensor presente em um smartphone, pode fornecer informações que compõem contextos diferentes para diferentes consumidores. Apesar de existirem muitas plataformas destinadas à captação e gerenciamento de compartilhamento de contexto, poucas lidam com aspectos relacionados a segurança e privacidade (PERERA *et al.*, 2014). Assim, é necessário que a segurança em IoT incorpore também essas questões, dado o potencial de uso desse tipo de aplicação dentro desse paradigma.

Nessa perspectiva, soluções de segurança em IoT devem ser capazes de lidar com heterogeneidade, escalabilidade e dinamicidade dos ambientes, com a possibilidade de adaptar-se à maioria dos cenários previstos nesse paradigma. Frente a isso, muitas tecnologias têm sido investigadas para lidar com esses aspectos de diferentes formas, propondo mecanismos e ferramentas de autenticação, autorização, detecção e prevenção de intrusões e criptografia, por exemplo.

Em relação a autorização de acesso, a técnica *Attribute-Based Access Control* (ABAC), controle de acesso baseado em atributos, tem despontado como uma das mais promissoras, devido a sua capacidade de permitir o controle de acesso de forma flexível e com alta granularidade (SERVOS; OSBORN, 2017). Tais propriedades podem ser aplicáveis para a segregação de permissões de acesso a conjuntos de dados diferentes para diversos consumidores.

Do ponto de vista de infraestrutura para implantação das funcionalidades de segurança, muitos trabalhos propõem a utilização técnicas emergentes, como *Network Functions Virtualization* (NFV), virtualização de funções de rede, e outras mais consolidadas e em expansão, como *Cloud, Fog e Edge Computing*. Os benefícios existentes nessas abordagens podem ser aproveitados para lidar com aspectos como heterogeneidade, mobilidade dos nós, escalabilidade e restrições dos dispositivos (FARRIS *et al.*, 2019; DONNO *et al.*, 2019).

Além disso, informações de contexto têm sido investigadas para o projeto de mecanismos de segurança em ambientes dinâmicos, distribuídos e heterogêneos, como é o caso da IoT (MATOS *et al.*, 2018). Informações de contexto podem ser usadas para decisões de

segurança em ferramentas de autenticação, controle de acesso e detecção de intrusões, dentre outros (KOUICEM *et al.*, 2018). Assim, existe a possibilidade de projetar artefatos que protejam os dados de forma adequada para cada contexto apresentado, aumentando a robustez e eficiência.

Dessa forma, considerando os desafios apresentados de segurança em ambientes IoT somados às especificidades das aplicações sensíveis ao contexto, explorar o potencial dessas técnicas pode ser de grande utilidade para modelar soluções de segurança adequadas para IoT, adaptando-se aos vários cenários possíveis a esse paradigma, mantendo os princípios de proteção de dados satisfatórios, sobretudo do ponto de vista de privacidade e confidencialidade.

Neste trabalho, é proposta uma arquitetura de implantação flexível para segurança no compartilhamento de contexto em ambientes IoT, denominada *Flexibility for Context-Aware Application Security in IoT* (FCAAS-IoT). A estrutura é composta por módulos que desempenham as funções de segurança, implementando controle de acesso por meio de ABAC e utilizam informações sobre o contexto da solicitação para escolha de algoritmos de criptografia. Políticas são estruturadas e servem de base para determinar as decisões a serem seguidas durante a execução das funções de segurança, objetivando proteger de forma adequada as várias informações que podem compor um contexto usado por uma aplicação.

Experimentos realizados por meio de uma prova de conceito comprovaram o correto funcionamento das funções de segurança e apontaram um acréscimo no tempo de resposta diante da atuação da arquitetura que varia de acordo com a quantidade de informações requeridas.

1.1 Objetivos

Considerando o problema exposto e os possíveis caminhos a serem adotados para intervenção, são apresentados aqui os objetivos desta pesquisa, que guiaram a realização do trabalho.

1.1.1 Objetivo Geral

Propor uma arquitetura de segurança com estrutura flexível para compartilhamento de informações de contexto entre dispositivos fonte e aplicações consumidoras capaz de adaptar-se aos vários ambientes IoT, utilizando atributos para definição de permissões em mecanismo de controle de acesso e para decisões de uso de criptografia com base em contexto de solicitação.

1.1.2 Objetivos Específicos

Mediante o objetivo geral traçado, configuram-se como desdobramentos os seguintes objetivos específicos:

a) Relacionar as demandas de segurança de aplicações sensíveis ao contexto no âmbito da IoT, com as já existentes e identificadas para esse paradigma;
b) Investigar como controle de acesso baseado em atributos pode ser utilizado/adaptado para criação de políticas que atendam ao cenário de fornecimento e consumo de informações de contexto em ambientes IoT;
c) Identificar como as características das tecnologias emergentes para estruturação e fornecimento de serviços em ambientes de rede, como NFV, *cloud*, *fog* e *edge computing*, podem ser exploradas para a criação de uma arquitetura de segurança flexível;
d) Sistematizar a utilização de informações relacionadas às requisições de informações para a criação de um contexto de solicitação como base para estruturação de políticas em um mecanismo de decisão sobre o uso de recursos de criptografia;
e) Descrever as funções dos módulos que compõem a arquitetura e como ocorre a interação entre eles para a execução das funções de segurança.

1.2 Contribuições do trabalho

Levando em consideração o escopo desse trabalho e seus objetivos, o processo de pesquisa realizado até sua conclusão apresenta algumas contribuições, as quais são listadas a seguir:

a) Verificação da viabilidade do modelo ABAC para controle de acesso em IoT, extensivamente nas demandas de segurança das aplicações sensíveis ao contexto, com a aplicação desse modelo ao problema tratado.
b) Sistematização de políticas de gerenciamento de criptografia baseadas em informações que compõem contexto de solicitação, implantando o conceito de segurança que se adeque as especificidades de cada cenário.
c) Identificação de aspectos das tecnologias NFV, *cloud*, *fog* e *edge computing* que podem ser explorados para construção de arquiteturas de segurança flexíveis para ambientes IoT, sobretudo para as que necessitam de compartilhamento de

contexto.

1.3 Organização do trabalho

Esta dissertação é composta de seis capítulos, a contar desta introdução, que contextualiza o trabalho, expõe seus objetivos e contribuições.

O capítulo 2 compreende o referencial teórico. São apresentados os fundamentos a respeito de IoT e aplicações sensíveis ao contexto, abordando e correlacionando os aspectos de segurança de ambos. As técnicas de segurança ABAC e segurança ciente de contexto, usadas para embasar a proposta, também são apresentadas. Por fim, os elementos pertinentes ao escopo da pesquisa sobre as tecnologias de NFV, *edge, fog* e *cloud computing*, focando em como estas podem ser aproveitadas pela proposta.

O capítulo 3 traz uma seleção de trabalhos relacionados à temática, os quais são expostos e comparados com a proposta, a fim de identificar similaridades e diferenças, contribuindo para a elaboração da arquitetura e da avaliação do uso das técnicas que a embasam.

No capítulo 4 é feita a exposição da FCAAS-IoT, apresentando sua estrutura, o modo como as funções de segurança são providas e composição das políticas. Adicionalmente, é delineado um cenário que permite entender a aplicação da arquitetura, identificando possibilidades de implementação, possibilidades e restrições.

O capítulo 5 apresenta a prova de conceito desenvolvida para validar a arquitetura proposta. É feita a descrição dos experimentos, os objetivos da avaliação e são apresentados os resultados obtidos, bem como suas análises.

O capítulo 6 é a conclusão do trabalho, onde são abordados possíveis desdobramentos desta pesquisa como trabalhos futuros.

2 REFERENCIAL TEÓRICO

Este capítulo constitui o referencial teórico do trabalho, o qual sumariza os principais tópicos relacionados às tecnologias utilizadas na concepção da proposta e auxilia a entender o problema tratado.

São abordados os conceitos de IoT e aplicações sensíveis ao contexto, expondo seus principais tópicos e enfatizando a questão da segurança nesses nichos, a fim de mostrar as intersecções e desafios existentes. Em seguida, são apresentadas as técnicas que têm sido apontadas como promissoras para segurança em ambientes IoT dentro do escopo ora considerado e que serviram de base para a concepção da arquitetura proposta, a saber: controle de acesso baseado em atributos (ABAC), *Context-Aware Security* (CAS) e tecnologias relacionadas à infraestrutura de redes, como virtualização de funções de rede (NFV), *Cloud*, *Fog* e *Edge Computing*.

2.1 Internet das Coisas

Não existe um consenso a respeito do conceito de Internet das Coisas (IoT, do inglês *Internet of Things*) devido a sua abrangência e complexidade. Uma das definições mais conhecidas coloca esse paradigma como uma interseção entre três visões de orientação: coisas, comunicação e semântica (ATZORI *et al.*, 2010). Entretanto, de forma mais genérica, a IoT pode ser vista como um sistema dinâmico e com alto grau de distribuição, formado pela conexão de objetos (*smart objects*) que produzem e consomem informações, sendo capazes de interagir com o mundo físico por meio de sensores e atuadores (MIORANDI *et al.*, 2012).

Nessa perspectiva, objetos do cotidiano, como eletrodomésticos, roupas e automóveis, passam a ser capazes de coletar informações de seus ambientes e enviá-las por meio de uma conexão de rede para servirem de matéria-prima para várias aplicações. Desse modo, essa grande quantidade de objetos coletando e transmitindo volumes de dados consideráveis é uma característica marcante de ambientes IoT. Isso abre espaço para a criação de diversos serviços e melhoramento de outros existentes em muitas áreas, tais como saúde, transporte, logística e automação residencial (ATZORI *et al.*, 2010; GUBBI *et al.*, 2013).

A Figura 1 apresenta uma visão geral da IoT abrangendo a interação entre várias áreas. Os dados captados pelos objetos inteligentes sobre o domínio no qual se encontram podem ser compartilhados via conexões de rede com dispositivos presentes no mesmo domínio ou fora

dele. Por exemplo, os dados coletados a partir de um dispositivo podem ser enviados para uma infraestrutura de nuvem a fim de serem processados, armazenados ou utilizados por aplicações para diversas finalidades, podendo resultar em ações dentro do ambiente fonte dos dados ou em outro, de acordo com o projeto das aplicações. Assim, cria-se um alto grau de compartilhamento e integração.

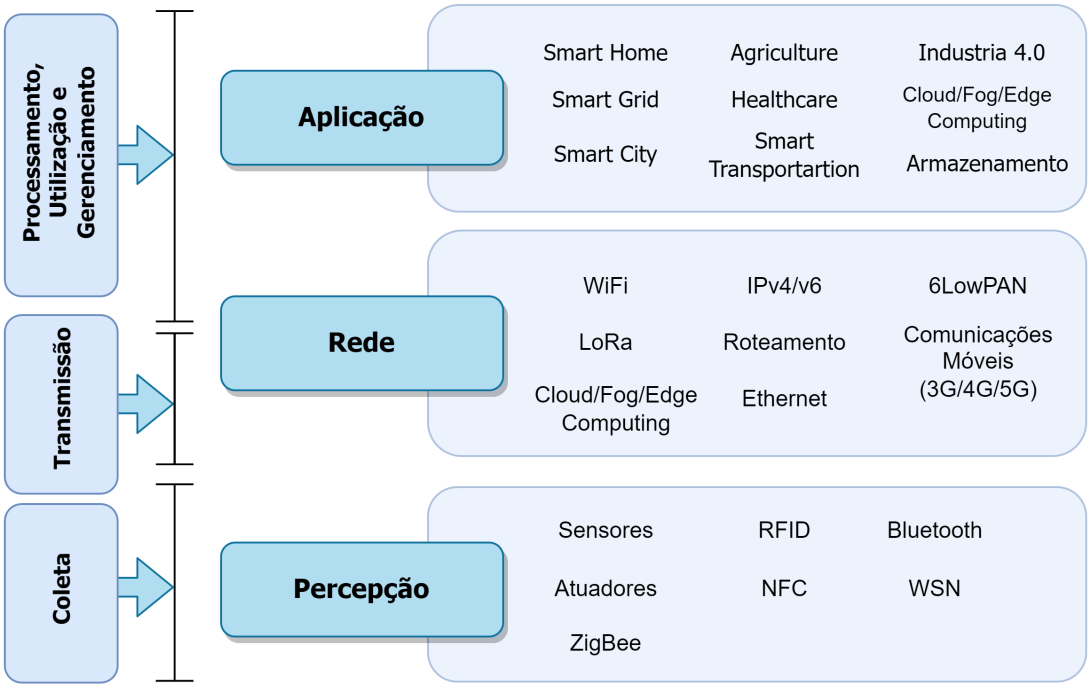
Figura 1 – Visão Geral da IoT.

Fonte: Miorandi *et al.* (2012)

A interação dos objetos inteligentes com o mundo físico se dá em três fases: (i) coleta, (ii) transmissão e (iii) processamento (BORGIA, 2014). Na mesma linha, Samie *et al.* (2016) descreve a operação geral de aplicações IoT em quatro estágios: aquisição, processamento, armazenamento e transmissão. As atividades inerentes a cada uma dessas fases, em geral, envolvem diversas tecnologias, as quais são comumente organizadas em arquiteturas em camadas, com o objetivo de facilitar sua compreensão em termos de funções a serem desempenhadas e o projeto de aplicações e tecnologias que viabilizem sua implantação.

Uma das arquiteturas mais difundidas é composta por três camadas: percepção, rede e aplicação. Outra proposta mais recente é composta por cinco camadas (objetos, abstração

Figura 2 – Arquitetura IoT de 3 camadas.



Fonte: elaborado pelo autor.

de objetos, serviço, aplicação e negócios), dando ênfase nos serviços de alto nível e no uso dos dados provenientes dos dispositivos inteligentes (AL-FUQAHA *et al.*, 2015). Entretanto, a arquitetura de três camadas é usada nesta seção para explicitar os elementos que compõem a IoT devido a sua simplicidade e por ser utilizada por muitos trabalhos relacionados a segurança nesse paradigma. A Figura 2 retrata essa organização, apresentando as técnicas e tecnologias que compõem cada camada e relacionando-as com as funções das fases de interação dos objetos inteligentes.

A camada de percepção agrupa os dispositivos que interagem com o mundo físico, desempenhando a função de coleta e, eventualmente, realizando ações no ambiente. A captação de informações é feita por meio de sensores presentes nos dispositivos inteligentes, variando em relação ao tipo de dado, precisão e capacidade de coleta. Já a possibilidade de intervir no ambiente se dá por meio de atuadores, que podem executar ações como acionar um determinado componente ou emitir alertas sonoros e luminosos, por exemplo. Grande parte dos dispositivos dessa camadas guardam em comum características físicas, como pouca capacidade de processamento e armazenamento de dados e restrições de alimentação elétrica (BORGIA, 2014).

Os dados coletados na camada de percepção precisam ser compartilhados entre dispositivos, o que fica a cargo da camada de rede. Como a mobilidade é um traço marcante

dos dispositivos IoT, a maioria das tecnologias utilizadas para conexão é sem fio, indo desde algumas mais difundidas, como Wi-Fi e redes de telefonia celular, até outras menos utilizadas pela população em geral, como *Long Range* (LoRa). Aqui ainda se enquadram questões como endereçamento e roteamento, ligadas diretamente à transmissão de dados. Nessa camada também se encaixam muitas tecnologias de computação em nuvem e névoa (*cloud* e *fog computing*), devido a sua capacidade de criarem grandes espaços de compartilhamento de dados (GUBBI *et al.*, 2013). É relevante frisar a importância da tecnologia 5G nas funções desempenhadas por essa camada, dado que vem sendo apontada como principal meio para a viabilização e difusão da IoT (LI *et al.*, 2018)

A camada de aplicação é a de mais alto nível e tem a função de prover suporte às aplicações dos diversos domínios IoT, utilizando dos dados coletados e compartilhados entre os dispositivos para fornecer serviços. Nela estão presentes funções relativas ao processamento, utilização e armazenamento de dados e gerenciamento de dispositivos, além de outras como provimento de segurança em diversos aspectos e partes do ambiente IoT.

Entretanto, cabe ressaltar que a evolução da IoT e das tecnologias, tanto de comunicação quanto de computação, envolvidas nesse paradigma torna essa segmentação funcional não rígida, fazendo com que determinadas funções sejam desempenhadas por componentes de várias camadas ou que uma determinada parte da infraestrutura agrupe várias funções.

Independente do domínio, aplicações IoT possuem requisitos em comum, tais como heterogeneidade, demandas de escalabilidade, algum grau de autonomia (comportamento independente da ação humana) e necessidade de um ambiente seguro para coleta e compartilhamento dos dados (BORGIA, 2014). Quando confrontados com algumas características da IoT, em especial, a mobilidade, a grande quantidade de dados e dispositivos e as restrições de processamento, armazenamento e energia destes, percebe-se que o cumprimento desses requisitos ainda é bastante desafiador.

Dentro dessas perspectiva, a segurança, entretanto, coloca-se como um dos principais desafios dentro da IoT (ATZORI *et al.*, 2010; ABOMHARA; KøIEN, 2014). Na subseção seguinte, discute-se de forma mais detalhada as questões relacionadas à segurança dentro desse paradigma.

2.2 Segurança em IoT

Desde o seu surgimento até o estágio atual, a segurança tem sido um ponto fundamental dentro da IoT. Nesse paradigma, o fato dos dispositivos coletarem dados não só do seu ambiente, mas também dos usuários, podendo, inclusive, monitorar hábitos de vida e padrões de comportamento, cria a necessidade de mecanismos de segurança e privacidade que evitem acessos e usos indevidos (YANG *et al.*, 2017). Tal preocupação permeia todo o cenário da IoT, abrangendo dados, dispositivos, redes de comunicação e usuários.

Por outro lado, atender a essa demanda não é uma questão simples graças a características dos objetos inteligentes e do próprio paradigma. Em geral, dispositivos IoT possuem restrições de alimentação energética e de recursos computacionais, como memória, armazenamento e processamento (KHAN; SALAH, 2018). Essas características não só inviabilizam o uso de mecanismos de segurança tradicionais, como dificultam o desenvolvimento de novas soluções (YANG *et al.*, 2017).

No cenário convencional de redes, a maioria dos recursos disponíveis para segurança demandam um grande volume de processamento, armazenamento de chaves e certificados de segurança ou envolvem intensos processos de comunicação, o que se contrapõe às restrições dos dispositivos. Adicionalmente, essas técnicas são construídas sob a perspectiva do usuário, enquanto que, na IoT, a comunicação também ocorre máquina a máquina, o que também deve ser considerado (ALABA *et al.*, 2017).

Do ponto de vista do ambiente IoT, existem muitos fatores que impactam no provimento de segurança. A quantidade e a diversidade de dispositivos demandam que os mecanismos de segurança deem suporte à heterogeneidade, escalabilidade e mobilidade (KOUICEM *et al.*, 2018). Desse modo, ferramentas e técnicas de segurança voltadas para a IoT devem ser capazes de lidar com esse grande volume de dados e dispositivos, comunicando-se por diversas tecnologias de comunicação e situados em redes sob a administração de diferentes organizações e que não possuem localização fixa em muitos casos.

Essa grande quantidade de objetos conectados na IoT também impacta no aumento das vulnerabilidades, uma vez que cada dispositivo é um potencial alvo de ataques de diversas naturezas (ALABA *et al.*, 2017). Não só os objetos inteligentes, mas também os demais componentes da IoT, como redes e aplicações, também são susceptíveis a ataques. Logo, projetar ferramentas e técnicas que abranjam as vulnerabilidades desses elementos e os vários tipos de ameças aos quais podem ser submetidos é uma tarefa potencialmente complexa.

Na literatura, existem muitos trabalhos que se dedicaram a entender e categorizar vulnerabilidades, ameaças e ataques no contexto da IoT. Em seu trabalho, Andrea *et al.* (2015) organizaram uma taxonomia com os tipos de ataques, dividindo-os em físicos, de rede, de software e de criptografia. De forma similar, Mahmoud *et al.* (2016) agruparam as ameaças de segurança em hardware, rede e aplicações. Em ambos é possível identificar a tendência em enxergar a segurança em IoT na perspectiva da arquitetura de três camadas (percepção, rede e aplicação), partindo das "coisas" até as aplicações.

Outra categorização é proposta por Khan e Salah (2018), na qual ataques e vulnerabilidade são divididos de forma ampla em três níveis (baixo, intermediário e alto) de acordo com as funções das camadas de IoT, considerando uma visão de necessidade de segurança multinível. O nível mais baixo agrupa problemas inerentes à camada física, como interferências e alterações de funcionamento. No nível intermediário estão as questões relacionadas aos protocolos de roteamento, comunicação e estabelecimento de sessões. Já os problemas referentes aos protocolos executados por aplicações IoT para fornecer e utilizar serviços compreendem os problemas de segurança de alto nível.

Assim, percebe-se que, desde os dispositivos mais restritos associados às funções de captação e sensoriamento, até os servidores de grande porte, que armazenam e consomem os dados captados, existem questões de segurança a serem observadas. Tais questões podem ser tanto ligadas às suas características físicas, quanto a sua operação e aos protocolos envolvidos no funcionamento. A Tabela 1 sumariza os principais tipos de ataques segundo esses trabalhos, dividindo-os por camadas da IoT e relacionando-os com as características de cada camada que criam vulnerabilidades que podem ser exploradas e potencializar a ocorrência destes, com base em Mahmoud *et al.* (2016) e Farris *et al.* (2019).

Os ataques da camada de percepção aproveitam-se das características físicas e de comportamento dos dispositivos inteligentes e podem ter como objetivo o roubo de informações, alteração no funcionamento ou inativação destes. Um objeto inteligente operado de forma remota, por exemplo, pode ser alvo de ataques físicos ou de instalação de equipamentos para fins de espionagem. Ataques que alterem o comportamento são potencialmente prejudiciais, como, por exemplo, alterar os intervalos em que o dispositivo permanece ativo ou a quantidade de mensagens a serem enviadas em um dado intervalo de tempo, pode gerar maior consumo de energia e recursos computacionais, levando a falhas e inativação do dispositivo.

Como a camada de rede é responsável pela comunicação entre dispositivos, os

Tabela 1 – Ataques de segurança em IoT.

Camada	Características/Vulnerabilidades	Ataques
Percepção	• Heterogeneidade dos nós; • Coleta de dados; • Restrições dos dispositivos; • Funcionamento autônomo; • Operação remota.	• Modificação de nó; • Interferência de sinal; • Injeção de código malicioso; • Dano físico; • Engenharia social; • Privação de sono; • Injeção de nós malicioso; • Espionagem; • Falsificação de nó; • Troca de chave criptográfica.
Rede	• Diversidade de protocolos; • Múltiplas tecnologias de comunicação; • Grande volume de dados; • Mobilidade dos nós.	• Análise de tráfego; • Clonagem de nó de roteamento; • Acesso não-autorizado; • Negação de serviço; • Man-in-the-Middle; • Ataques de informação de roteamento; • Sybil attack; • Sequestro de equipamento.
Aplicação	• Comportamento do usuário; • Diversidade de aplicações; • Grande quantidade de conexões.	• Malwares; • Scripts maliciosos; • DoS e DDoS; • Cyber Attacks; • Engenharia social.

Fonte: elaborado pelo autor.

ataques têm como alvo o fluxo de dados compartilhado e as informações relacionadas a esse processo. Alguns desses ataques podem ser considerados passivos, quando apenas capturam dados que trafegam na rede, enquanto outros influem no fluxo de dados, alterando o conteúdo de um pacote parcialmente ou substituindo conjuntos de dados completos. Devido a isso, ataques na camada de rede são potencialmente prejudiciais aos sistemas, pois podem ter uma grande abrangência (FARRIS *et al.*, 2019). Alterações nas informações de roteamento de um nó, por exemplo, podem afetar o encaminhamento de dados; ataques *Man-in-the-Middle*, podem ser usados tanto para espionagem quanto para envio de dados incorretos ou maliciosos; análise de tráfego de rede, pode fornecer informações a respeito do comportamento dos dispositivos e usuários, dentre outros.

No que diz respeito à camada de aplicação, muitas vulnerabilidades podem surgir desde o processo de desenvolvimento do software até a contaminação por *malwares* durante sua operação. Erros de código, por exemplo, podem abrir brechas a serem exploradas em ataques de negação de serviço, *Denial of Service* (DoS). Adicionalmente, a diversidade de aplicações e a forma como são desenvolvidas faz com que sejam adotados diferentes padrões e níveis de segurança. Cabe ressaltar, ainda, que o comportamento dos usuários pode potencializar vulnerabilidades existentes e favorecer a ocorrência de incidentes relacionados à segurança.

Considerando vulnerabilidades, ameaças, tipos de ataques e cenários da IoT, alguns

requisitos de segurança podem ser elencados e são comuns à maioria das aplicações nesse paradigma (KOUICEM *et al.*, 2018):

a. **Confidencialidade:** diz respeito a garantir que os dados sejam acessados apenas por quem de direito, protegendo seu conteúdo contra acessos indevidos. No contexto da IoT, não somente usuários, mas aplicações, sistemas, serviços e objetos podem requisitar acesso a recursos (MAHMOUD *et al.*, 2016).
b. **Integridade:** pode ser entendido como proteger as informações contra alterações intencionais ou não entre a origem e o destino, enquanto trafegam por diversas redes. Esse requisito é importante não só sob o aspecto de proteção da informação em si, mas também para a garantia da precisão e veracidade das informações, uma vez que o funcionamento de algumas aplicações dependem desses atributos para um correto funcionamento (LIN *et al.*, 2017).
c. **Disponibilidade:** consiste em garantir que dados estejam disponíveis a quem tem permissão de acesso sempre que necessário. Na IoT, esse requisito abrange não só informações, mas também acesso a objetos inteligentes e serviços que podem ser usados pelos diversos agentes envolvidos.
d. **Autenticação:** está muito ligada a confiança, pois visa confirmar a identidade das entidades (dispositivos, usuários, aplicações etc.) antes de permitir o acesso a recursos ou iniciar um processo de comunicação. Desse modo, mecanismos de autenticação buscam garantir a legitimidade das partes comunicantes e dos dados trocados entre elas no que diz respeito a origem destes (LIN *et al.*, 2017).
e. **Não-repúdio:** objetiva impedir que uma entidade negue que realizou determinada ação dentro do sistema, como envio de uma mensagem de solicitação, acionamento de um processo e alteração no comportamento de algum dispositivo, estando muito relacionada com auditoria e contabilização do uso de recursos.
f. **Privacidade:** diz respeito à proteção das informações críticas, sobretudo de usuários, de modo a evitar que seja possível caracterizar uma entidade ou deduzir algo sobre seu comportamento por meio de informações disponibilizadas, além daquilo que foi expressamente autorizado e sob o qual se tem ciência da disponibilização. Assim, mecanismos de privacidade procuram garantir algum grau de controle por parte do usuário sobre o uso e divulgação de suas informações.

Esses requisitos devem servir como guias no desenvolvimento e na implantação de

soluções de segurança no âmbito da IoT, bem como na aferição de sua efetividade. Além disso, cada tipo de aplicação e cenário pode ter suas próprias demandas de segurança, o que reforça ainda mais essa questão. Assim, é necessário o desenvolvimento de técnicas que lidem com a heterogeneidade, escalabilidade, mobilidade e demais demandas da IoT e que possam se adequar aos múltiplos cenários existentes sem perder a efetividade no fornecimento de segurança.

2.3 Aplicações sensíveis ao contexto

O conceito de contexto já é bastante difundido, sobretudo nas áreas ligadas à computação ubíqua e pervasiva. Uma das principais definições coloca contexto como sendo "qualquer informação que pode ser usada para caracterizar a situação de uma entidade, onde entidade pode ser uma pessoa, lugar ou objeto físico ou computacional"(ABOWD *et al.*, 1999). Refinando esse conceito, Dey (2001) acrescenta que entidade é "uma pessoa, lugar ou objeto que é considerado relevante para a interação entre um usuário e uma aplicação, incluindo o próprio usuário e suas aplicações".

Outra definição importante foi dada por Abowd e Mynatt (2000), a qual utiliza o que denominaram de 5Ws para definir o que é contexto: *Who* (quem), *What* (o quê), *Where* (onde), *When* (quando) e *Why* (por quê). Para esses autores, as respostas a essas perguntas seriam capazes de fornecer informações básicas sobre uma entidade, desde a sua localização física até características de estado ou identidade, por exemplo.

Quando uma aplicação utiliza informações de contexto para prover algum serviço ou executar alguma tarefa relevante para o usuário, ela possui a propriedade denominada de ciência de contexto (do inglês, *context-awareness*) (ABOWD *et al.*, 1999). Uma aplicação que usa informações de localização, horário, preferências de consumo e situação do trânsito para sugerir uma melhor rota ou programação para o usuário, por exemplo, pode ser caracterizada como uma aplicação sensível ao contexto ou ciente de contexto. Do mesmo modo, uma aplicação capaz de adaptar a temperatura do ar condicionado e a luminosidade, com base em informações como horário e temperatura externa, também se encaixaria nessa definição.

Com base nesses conceitos, percebe-se que existe um amplo espaço de possibilidades para aplicações cientes de contexto, as quais utilizem informações de diversas origens e sejam capazes de prover várias funcionalidades, em diferentes graus de complexidade, tanto no que diz respeito aos sistemas que utilizam as informações, quanto pela complexidade do contexto a ser usado.

Entretanto, mesmo diante desse grande número de possibilidades, três recursos básicos podem ser providos por aplicações sensíveis ao contexto de forma geral (PERERA *et al.*, 2014):

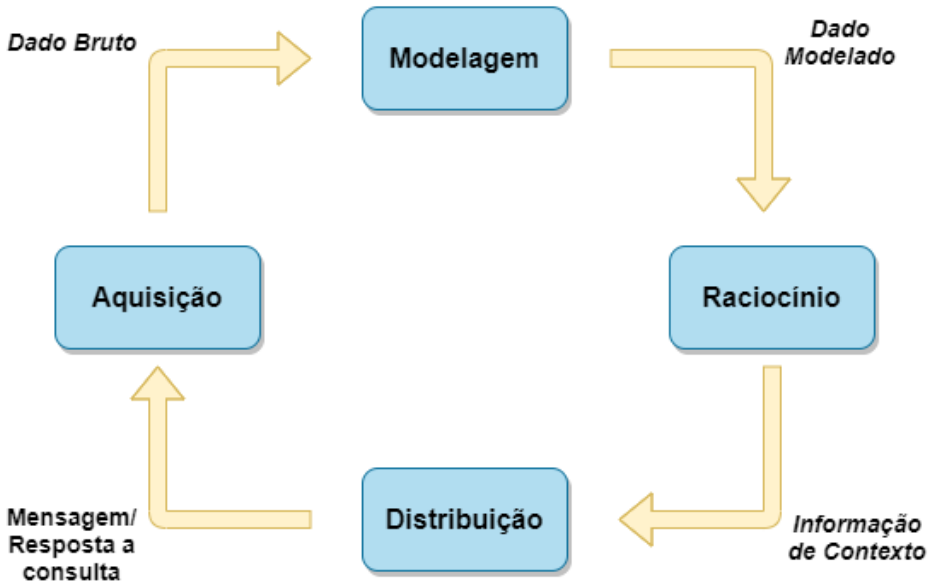
a. **Apresentação:** o contexto é utilizado pelas aplicações para decidir quais recursos ou serviços tornar visíveis para o usuário. Esse é o caso, por exemplo, de uma aplicação que usa a informação de localização atual e do ponto de destino para sugerir uma rota ou quais estabelecimentos estão próximos.
b. **Execução:** nesse caso, informações de contexto são base para a realização de ações, como acionamento de processos ou dispositivos, o que, em geral, acontece de forma autônoma. Por exemplo, uma aplicação que utilize imagens de câmeras de segurança, data e hora e estado de sensores de detecção de presença para emitir um alerta de invasão em uma residência se enquadraria nesse tipo.
c. **Marcação:** também denominada anotação, nesse caso o contexto é usado para associar determinadas informações a uma entidade. Ela permite, por exemplo, distinguir de qual sensor uma informação é proveniente nos casos em que várias informações são concatenadas para criação de um contexto mais complexo.

A combinação dessas funções, abre espaço para a implementação de diversos tipos de aplicações com diferentes comportamentos e propósitos. Entretanto, antes de serem usadas pelas aplicações, as informações de contexto passam por uma série de processos que podem ser organizados em um *Ciclo de Vida de Contexto*, composto por quatro fases, conforme mostrado na Figura 3: aquisição, modelagem, raciocínio e distribuição (PERERA *et al.*, 2014).

Na fase de aquisição, as informações de contexto são obtidas a partir de fontes, tais como sensores e base de dados, podendo ocorrer de forma periódica ou contínua. As informações adquiridas passam por um processo de modelagem que consiste em colocá-las em um formato pré-definido, com inserção de metadados. Nessa etapa, são utilizadas técnicas como linguagens de marcação, modelagem baseada em lógica, modelo chave-valor e ontologias (PRADEEP; KRISHNAMOORTHY, 2019).

Na fase de raciocínio, novos conhecimentos são obtidos a partir do contexto disponível, a fim de alcançar um maior grau de precisão, eliminando disparidades e dados corrompidos e conferindo potencial de utilização aos dados coletados. Essa fase envolve três etapas: pré-processamento, fusão de dados e inferência. Por fim, na fase de distribuição, as informações de contexto são compartilhadas com as aplicações, o que pode ser feito por meio de consul-

Figura 3 – Ciclo de vida do contexto.



Fonte: adaptado de Perera *et al.* (2014)

tas, esquemas de publicação-assinatura ou envio direto. Logo, nessa última fase, tecnologias de comunicação estão envolvidas para permitir envio dos fornecedores de contexto para as aplicações.

O ciclo de vida de contexto permite visualizar que a utilização desse tipo de informação é um processo complexo, por meio do qual dados em seu estado bruto são coletados de ambientes/sistemas reais, formatados, transformados em informações úteis e distribuídos às aplicações. Cada uma dessas etapas pode ser realizada de forma centralizada ou distribuída, a depender das tecnologias e arranjo desenhado para o sistema.

Independente do tipo de aplicação, as informações de contexto possuem algumas características comuns à maior parte delas, segundo Arfaoui *et al.* (2005), a saber: são coletadas através de sensores ou redes de sensores, os dispositivos de coleta têm pequeno porte e restrições de características físicas, podem ser compostas de diversas fontes distribuídas, mudam constantemente, são oriundas de objetos móveis, delimitam um tempo e um espaço e são imperfeitas e com um certo grau de incerteza.

Fazendo um paralelo entre essas características e os ambientes IoT, algumas semelhanças podem ser elencadas, como as restrições dos dispositivos, o caráter distribuído e a mobilidade destes. Entretanto, essas características passam por uma atualização, uma vez que ambientes IoT englobam também dispositivos com maior poder computacional e mais

autonomia energética, como *smartphones*, e dispositivos fixos, como sensores em semáforos ou eletrodomésticos podem ser fontes de contexto. Adicionalmente, cenários IoT podem utilizar informações de contexto para múltiplas finalidades e não apenas em um único domínio, como é comum a aplicações sensíveis ao contexto.

Além da complexidade envolvida nos processos do ciclo de vida de contexto, outras preocupações são presentes, como armazenamento e compartilhamento de informações, qualidade de serviço (*Quality of Service* (QoS)), qualidade do contexto, confiabilidade e segurança e privacidade das informações, usuários e sistemas (PRADEEP; KRISHNAMOORTHY, 2019). Essas questões são potencializadas na IoT, devido ao aumento da quantidade de dispositivos, dados e aplicações, heterogeneidade destes e maior grau de distribuição, o que torna o ambiente mais dinâmico e dificulta o controle sobre a disseminação e uso das informações.

Considerando o teor das informações de contexto, um dos principais desafios a serem superados é a segurança e privacidade na distribuição (MATOS *et al.*, 2020). Na seção seguinte é feito um recorte a respeito desse tema.

2.4 Questões de segurança em aplicações sensíveis ao contexto na IoT

Na IoT como um todo, deve haver meios que permitam controlar quais e para quem as informações são disponibilizadas, bem com o formas de garantir a não utilização para fins diferentes do escopo definido ou esperado (ATZORI *et al.*, 2010). No caso das aplicações sensíveis ao contexto, em especial dentro desse paradigma, algumas preocupações com segurança são acentuadas.

Em um primeiro momento, é necessário pontuar que a evolução dos sensores e a quantidade de dispositivos disponíveis com a IoT aumentou a capacidade de coleta e possibilitou a construção de contextos mais complexos e que carregam cada vez mais informações sobre os usuários, sistemas e ambientes, tornando a privacidade e a segurança ainda mais importantes (PERERA *et al.*, 2014). Caso as mensagens trocadas entre uma aplicação e o ponto fornecedor de contexto sejam interceptadas, por exemplo, as informações podem ser acessadas se não houver um esquema de proteção, colocando em risco a segurança das entidades sobre as quais as informações se referem e o funcionamento da aplicação.

Vários ataques de segurança podem ter efeitos ainda mais danosos nesse tipo de aplicação. Aqueles que têm como objetivo modificar informações ou comportamentos dos sensores podem ser ainda mais prejudiciais, visto que o funcionamento dos sistemas muitas vezes

está ligado à precisão dos dados. Esse é o caso de sistemas críticos, como transporte inteligente, automação industrial e assistência médica, nos quais violações desse tipo podem gerar impactos consideráveis, não só às aplicações, mas aos indivíduos (SEZER *et al.*, 2018). Logo, são necessários mecanismos que implementem autenticação, controle de acesso, confidencialidade e integridade das informações.

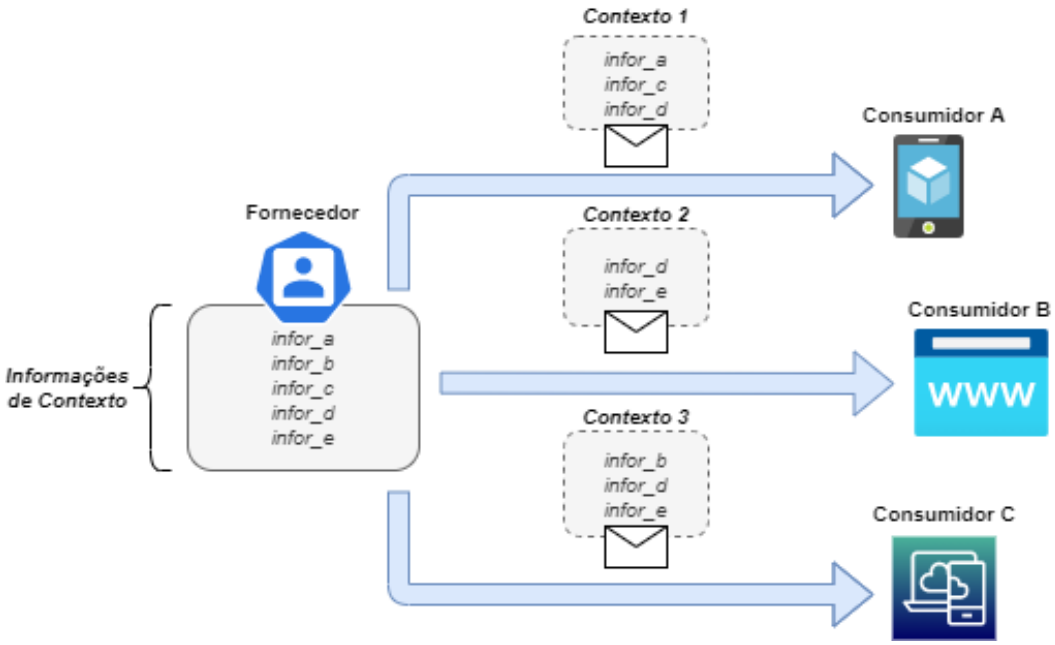
Ataques de interceptação, como *Man-in-the-Middle*, podem fazer com que informações de contexto sejam utilizadas por agentes não autorizados e/ou maliciosos. Além disso, é possível obter informações sobre entidades por meio do cruzamento de dados coletados a partir de pacotes interceptados e do mapeamento nos envios entre uma origem e um destino. Desta feita, são necessários também recursos que permitam o controle sobre quais informações serão disponibilizadas, para quem, para qual finalidade e até mesmo em que circunstâncias, além de auditoria, proteção contra rastreamento e aumento de confiança na troca de dados (MAHALLE; DHOTRE, 2020).

Além da proteção contra ataques, outro requisito de segurança que exige atenção nesse caso é a disponibilidade. Dependendo da quantidade e tipo dos sensores, é possível que um único nó forneça dados para compor contextos distintos a serem enviados e utilizados por várias aplicações. Ou seja, o ambiente de compartilhamento de contexto na IoT é mais dinâmico, dificultando o controle. Desse modo, é necessário garantir que as informações sejam disponibilizadas de forma segura, protegendo a privacidade dos usuários e aplicações, sem, no entanto, comprometer o provimento dos serviços com a qualidade esperada. Políticas de controle de acesso são um exemplo de técnica que pode ser empregada para segmentar acesso a conjuntos de dados.

A Figura 4 ajuda a pontuar melhor algumas questões. Uma fonte de contexto, que podem ser dispositivos sensores conectados a um usuário ou presentes no ambiente, fornece contexto para diversas aplicações mediante solicitações. Cada aplicação utiliza apenas um subconjunto das informações disponíveis, ou seja, cada aplicação tem sua própria concepção de contexto de acordo com seu escopo de funcionamento.

Nesse cenário, são necessários artefatos que possibilitem especificar permissões sobre as informações de contexto individualmente, uma vez que múltiplos contextos podem ser formados pela combinação destas e cada destinatário só deve ter acesso às informações devidas. Em outras palavras, é preciso implantar controle de acesso e autorização, mediante políticas e credenciais de segurança, por exemplo.

Figura 4 – Exemplo de fornecimento de contexto.



Fonte: elaborado pelo autor.

Além disso, é preciso garantir confidencialidade, fazendo com que apenas o destinatário consiga ter acesso às informações de contexto enviadas. Nesse caso, técnicas de criptografia poderiam ser empregadas, considerando, entretanto, que o fornecedor possivelmente tenha restrições de armazenamento e processamento e que serão necessárias várias chaves criptográficas dado que são muitos destinatários.

Adicionalmente, questões como heterogeneidade dos dispositivos e dados, distribuição e mobilidade dos dispositivos, múltiplas tecnologias de comunicação e requisitos de desempenho também devem ser considerados no projeto de esquemas de segurança nesse âmbito, tornando-se desafios importantes, dado que já estão presentes dentro da IoT de forma geral.

Portanto, percebe-se que a segurança no compartilhamento de contexto dentro da IoT é um desafio não só em termos de requisitos a serem cumpridos, mas também pela urgência e importância, a fim de evitar danos a usuários, criando um ramo de estudo relevante dentro desse paradigma, muito importante para sua aplicabilidade e expansão em múltiplos cenários.

2.5 Tendências de segurança em IoT

Segurança em IoT tem sido objeto de muitas pesquisas, em grande parte, em busca de soluções capazes de lidar com os desafios existentes e atender aos requisitos de privacidade e segurança, ambos tópicos já abordados em seções anteriores. Muitas tecnologias despontam como promissoras nesse âmbito.

Técnicas de *Machine Learning* (ML), por exemplo, vêm sendo muito empregadas para compor mecanismos de autenticação, autorização e para detecção de ataques em ambientes IoT, levando em consideração, principalmente a dinamicidade inerente a estes e a quantidade massiva de dispositivos e dados (XIAO *et al.*, 2018; HUSSAIN *et al.*, 2020). Tais abordagens permitem a implantação de soluções capazes de lidar com as constantes mudanças, conferindo adaptabilidade e respostas mais rápidas em casos de incidentes de segurança.

Outra tecnologia que vem recebendo bastante atenção é *blockchain*, a qual tem sido empregada para o desenvolvimento de soluções de autenticação e autorização, integridade dos dados e transações e mecanismos de privacidade (KHAN; SALAH, 2018). A arquitetura distribuída dessa tecnologia e a forma como os dados são armazenados e acessados abrem espaço para a criação de ambientes seguros de comunicação em muitos cenários da IoT.

Nesta seção, serão abordadas outras técnicas apontadas como promissoras para lidar com questões de segurança dentro da IoT e que foram utilizadas na concepção da arquitetura ora proposta, expondo seus fundamentos e pontuando como podem ser empregadas para provimento de segurança em ambientes inteligentes. Assim, são apresentadas duas técnicas, ABAC, focada em autorização, e segurança ciente de contexto (CAS), que pode ser empregada com múltiplas finalidades. Em um segundo momento, são abordadas tecnologias mais focadas em infraestrutura de organização dos dispositivos e processamento, que possuem potencialidades a serem aplicadas dentro do problema analisado e podem ser associadas à proposta aqui formulada, a saber: NFV, e os paradigmas de *Cloud*, *Fog* e *Edge Computing*.

2.5.1 Attribute-Based Access Control (ABAC)

Controle de acesso ou autorização é uma das principais ferramentas utilizadas para segurança, pois permite monitorar quem acessa cada recurso e prevenir de forma efetiva acessos e usos não autorizados, podendo ou não ser associado a mecanismos de autenticação e criptografia.

Existem diversas técnicas para controle de acesso que utilizam diferentes estratégias e características para implantar essa função. Dentre elas estão *Discretionary Access Control* (DAC), *Mandatory Access Control* (MAC) e *Role-Based Access Control* (RBAC). Esses modelos não apresentam um bom grau de adequação para ambientes dinâmicos, como a IoT, pois dependem de conhecimentos específicos sobre as entidades obtidos previamente ou da atribuição de funções e permissões para definir as políticas de controle de acesso aos recursos (QIU *et al.*, 2020). Tais características prejudicam a adaptabilidade na construção de políticas que consigam abranger a

diversidade e constantes mudanças que se colocam como características marcantes dos novos ambientes de computação e comunicação.

Diante dessa nova realidade, o modelo de Controle de Acesso Baseado em Atributos (ABAC, do inglês, *Attribute-Based Access Control*) tem se apresentado como promissor, sendo objeto de estudo de muitas iniciativas. Nesse modelo de controle de acesso são utilizados atributos associados às entidades (sujeitos, objetos e atividades) para definir a autorização de operações sobre um determinado objeto com base em políticas, regras e relacionamentos entre as entidades (HU *et al.*, 2013). Assim, características como localização, nome e identificadores, atreladas a requisitantes, que podem ser aplicações, usuários ou dispositivos, compõem um conjunto de atributos que é avaliado de acordo com políticas definidas para um determinado objeto (arquivo, sistema, banco de dados etc.) a fim de decidir se a operação requerida pode ser executada.

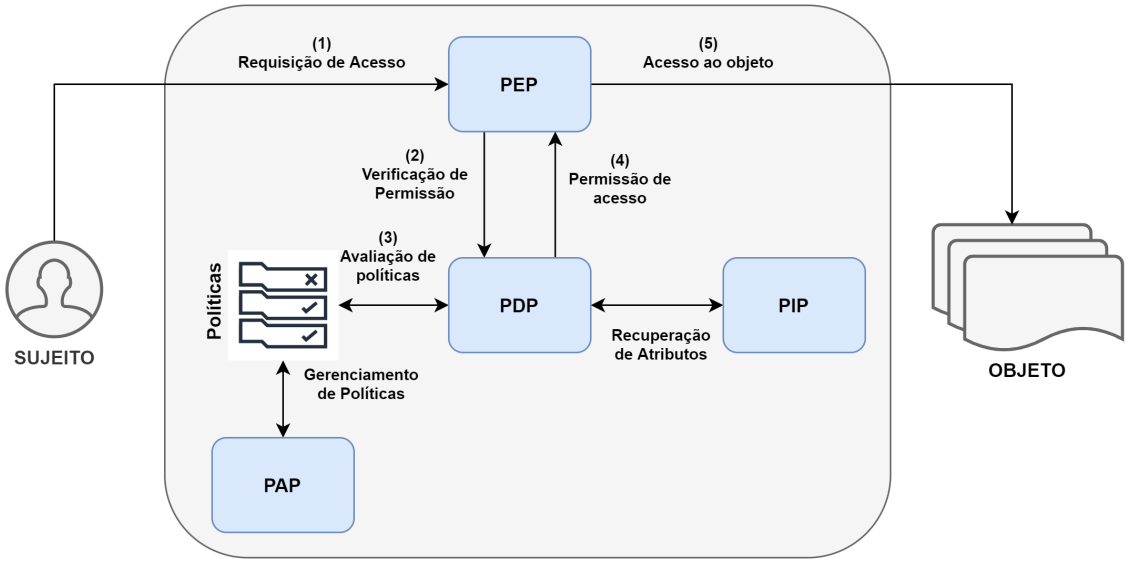
Toda a concepção do ABAC utiliza o conceito de atributos para regular os mecanismos de acesso, os quais são, preferencialmente, informações já presentes nas entidades envolvidas no processo de solicitação, embora também possam ser atribuídos manualmente. De forma geral, os atributos podem ser agrupados como segue (SERVOS; OSBORN, 2017):

a. **Usuário:** dizem respeito a informações dos sujeitos que estão requisitando acesso, como nome, idade, cargo, endereço ou algum identificador.
b. **Objeto:** referem-se aos atributos dos objetos do sistema, ou seja, daqueles que são protegidos pelo modelo de controle de acesso. Esse tipo pode incluir dados do objeto em si (autor, data de criação, data de modificação, tipo etc.) ou dados a respeito do conteúdo armazenado no objeto.
c. **Ambiente:** são informações provenientes do estado atual do ambiente, como data, hora, quantidade de usuários utilizando o recurso etc..
d. **Conexão:** compreende o conjunto de informações que só tem validade para a conexão atual em curso, como endereço IP, localização física e data/hora do início da conexão.
e. **Administrativo:** diz respeito a atributos de configuração que são inseridos pelo administrador do sistema de forma manual ou por meio de um sistema automatizado. Por exemplo, nível de segurança da rede no qual determinada política deve ser aplicada e nível de confiança mínimo aceitável para uma requisição.

Esses vários tipos de atributos são combinados para formalizar as permissões de

acesso por meio de políticas que são base desse modelo e são avaliadas para cada solicitação de acesso, verificando a conformidade entre os que são fornecidos e os que estão especificados para um dado objeto. Esse modelo é construído com base no padrão *eXtensible Access Control Markup Language* (XACML) (HU *et al.*, 2013). Embora haja muitas implementações diferentes do ABAC, todas possuem em comum um conjunto de componentes que sistematiza as funcionalidades necessárias para o funcionamento desse método. A figura 5 mostra esses elementos básicos da arquitetura de referência do ABAC e seu fluxo de informações durante o processo de autorização.

Figura 5 – Modelo ABAC.



Fonte: adaptado de Hu *et al.* (2013).

O *Policy Enforcement Point* (PEP) é responsável por receber as requisições de acesso e forçar a aplicação de políticas, dando início ao processo de autorização. Esse elemento também pode ser encarregado de coletar os atributos fornecidos para cada solicitação, formatando uma estrutura de dados que possa ser usada no decorrer do processo. O *Policy Decision Point* (PDP) realiza a verificação das políticas aplicadas à solicitação em análise e retorna se o acesso foi permitido ou negado, considerando os atributos fornecidos pelo PEP. A verificação do cumprimento das regras de acesso estabelecidas nas políticas pode ser feita por vários métodos, como lógica booleana e listas de atributos, por exemplo (HU *et al.*, 2013).

O *Policy Administration Point* (PAP) tem a função de possibilitar o gerenciamento das políticas, mediando o armazenamento, modificação e exclusão. As políticas podem ser armazenadas em um banco de dados, por exemplo, e serem recuperadas pelo PDP a cada

solicitação conforme necessário. Já o último elemento, *Policy Information Point* (PIP) tem a tarefa de fornecer atributos complementares necessários para a verificação de uma política, quando os atributos fornecidos na requisição são insuficientes.

Considerando esses componentes e suas interações, é possível visualizar como os atributos são usados no processo de autorização e como funciona a lógica de verificação das políticas, a qual compara os atributos fornecidos e retorna a permissão ou negação de acesso e, eventualmente, o tipo (leitura, leitura-modificação etc.). Assim, esses componentes agem de forma integrada para executar as tarefas de extração de atributos, análise de políticas e concessão de permissões, podendo ser implementados de diversas formas, desde que desempenhem as funções dentro do desenho do modelo.

O fato das políticas serem construídas com base em atributos já existentes das entidades, sem que haja necessidade de atribuição de informações, rótulos ou qualquer parâmetro que sirva como instrumento de verificação, confere ao ABAC uma implementação e administração mais fácil, automatizando as decisões de acesso (SERVOS; OSBORN, 2017). A forma como os atributos são obtidos para a construção do mecanismo de autorização afeta de forma significativa o desempenho do esquema implementado, o que potencializa ainda mais os benefícios do uso de atributos já existentes pelo ABAC em ambientes com grande quantidade de dispositivos e mudanças constantes na disposição e comunicação destes (QIU *et al.*, 2020).

Outro ponto forte dessa abordagem para criação de políticas é a possibilidade de especificar esquemas de autorização sem que sejam conhecidos previamente por completo todos os sujeitos, conferindo uma maior flexibilidade e gestão de autorização de acesso de inúmeras entidades, adequando-se a ambientes com alto grau de distribuição e que passem por mudanças frequentemente (HU *et al.*, 2015). Esse é o caso de muitos ambientes IoT, nos quais é dispendioso e até mesmo impossível praticamente mapear todas as entidades que solicitarão acesso a objetos protegidos por um esquema de autenticação, bem como as relações entre estes para fins de elaboração de políticas.

Devido a essas características, ABAC tem se apresentado como um dos modelos mais viáveis de controle de acesso para IoT, uma vez que objetos e suas informações podem ser acessados por várias entidades de forma distribuída, havendo um alto grau de mobilidade dos nós e dinamicidade das operações. Somando a isso a heterogeneidade e as restrições dos dispositivos, torna-se inviável a utilização de outros modelos de controle de acesso, como o MAC, DAC e RBAC, conforme citado anteriormente.

Portanto, ABAC pode prover a flexibilidade e granularidade necessárias, estabelecendo esquemas de autorização seguros e eficientes. Adicionalmente, a incorporação de informações de contexto tem sido investigada para aprimoramento do ABAC, de modo a torná-lo ainda mais adaptável e capaz de responder mais assertivamente considerando situações em tempo real (QIU *et al.*, 2020). Isso aumenta o seu potencial de benefícios e abre espaço para novas abordagens na elaboração e aplicação de políticas, sobretudo quando aliado a outras técnicas, como criptografia, por exemplo.

2.5.2 Context-aware Security (CAS)

Grande parte das ferramentas de segurança, como sistemas de detecção e prevenção de intrusões e *firewalls*, têm sua operação baseada em informações. Devido à capacidade de refletir o estado atual de sistemas e à riqueza de informações existentes, a sensibilidade ao contexto também tem sido investigada e aplicada à segurança. Na verdade, esse conceito foi esboçado anos atrás em Brezillon e Mostefaoui (2004), embora venha ganhando mais força e aplicações efetivas atualmente.

Context-Aware Security (CAS), também chamada de *Context-based Security*, é uma abordagem que consiste em utilizar informações de contexto na especificação de soluções de segurança, com base em situações de uso do sistema (BREZILLON; MOSTEFAOUI, 2004; MOSTEFAOUI; BREZILLON, 2004). Desse modo, informações coletadas a respeito de usuários, dispositivos e/ou ambiente podem ser combinadas e integradas a decisões de segurança durante o funcionamento do sistema.

Nesse sentido, surge o conceito de *Contexto de Segurança* que pode ser considerado como *"um conjunto de informações coletadas do usuário e do ambiente da aplicação que é relevante para a infraestrutura de segurança de ambos (usuário e aplicação)"*(BREZILLON; MOSTEFAOUI, 2004; MOSTEFAOUI; BREZILLON, 2004). Informações como data e hora, perfil do usuário, tipo de acesso, endereço IP, dispositivos envolvidos na operação, aplicação remetente, tipo de informação requerida, histórico de interações, entre outras, podem compor contextos de segurança.

CAS pode auxiliar na construção e aprimoramento de sistemas de segurança, reconfigurando os mecanismos e ajustando parâmetros, podendo ser incorporada em técnicas com múltiplas finalidades, tais como autenticação, encriptação e controle de acesso (HABIB; LEISTER, 2015). Por exemplo, um sistema de autenticação pode utilizar as informações de

rede, horário, localização e perfil de comportamento do usuário para validar um acesso; ou ainda, um sistema de detecção de intrusões pode fazer uso do tipo e quantidade de pacotes de um determinado protocolo que ingressam na rede para alertar sobre uma tentativa de invasão ou para adequar as regras de *firewall* para esse protocolo sem a intervenção do administrador do sistema.

A utilização de contexto para fins de segurança tem o potencial de inserir maior adaptabilidade aos sistemas, tornando-os capazes de responder a eventos e situações que exijam adequações com a finalidade de manter um nível de segurança condizente com o cenário para um dado instante. Assim, CAS demonstra-se bastante atrativo para sistemas dinâmicos e heterogêneos, como é o caso da IoT, para os quais técnicas existentes não são adequadas, pois, em geral, não levam em consideração informações contextuais, prejudicando sua efetividade frente ao relacionamento das entidades nesse tipo de sistema (MATOS *et al.*, 2018). Além disso, o alto grau de distribuição e os diversos usos da IoT tornam essa técnica ainda mais interessante dentro desse paradigma.

Com o auxílio de contexto para provimento de segurança em IoT, dispositivos inteligentes podem utilizar os dados coletados por seus diversos sensores e recebidos de outros dispositivos para tomar decisões de segurança que se adaptem ao contexto atual das transações em curso (RAMOS *et al.*, 2015). Um dispositivo pode, por exemplo, usar informações a respeito da reputação de um nó presente na mesma rede, ou seja, o grau de confiança, para decidir qual nível de privacidade adotar na comunicação, considerando o grau de privacidade dos dados a serem transmitidos.

Entretanto, o uso de informações de contexto para segurança possui as mesmas preocupações já conhecidas para sistemas sensíveis ao contexto. Todo o ciclo de vida de contexto deve ser delineado e monitorado, desde a coleta dos dados brutos até a transmissão, a fim de fornecer informações de contexto corretas, confiáveis e adequadas aos níveis de segurança requeridos pelos sistemas (RAMOS *et al.*, 2015). Problemas no contexto considerado nas tomadas de decisões podem causar grandes impactos, repercutindo na segurança de um nó ou sistema como um todo, resultando em prejuízos como vazamentos de informações e mau funcionamento dos mecanismos de segurança.

2.5.3 Network Function Virtualization (NFV)

Network Functions Virtualization (NFV) é uma abordagem emergente para projeto, implantação e administração de rede que vem ganhando cada vez mais atenção nos últimos anos.

A ideia básica é fazer a desvinculação entre a execução de funções de rede e os equipamentos que as executam (MIJUMBI *et al.*, 2016). Desta forma, NFV propõe que funções de rede, como *firewalls*, servidores de nomes, balanceadores de carga e roteadores, sejam implementadas via software e executadas em hardware genérico de servidores de propósito geral e estruturas de *data centers*, por meio de tecnologias de virtualização, criando funções de rede virtualizadas (*Virtualized Network Function*s (VNFs)).

O surgimento de NFV foi motivado pelas mudanças nas demandas no cenário de redes, colocando-se como alternativa para lidar com a diversidade de serviços, quantidade de dados, requisitos diversos de clientes e rápido surgimento de novas implementações de protocolos e serviços (YI *et al.*, 2018). Tal cenário passou a reforçar cada vez mais a necessidade de ferramentas que possibilitassem uma maior flexibilidade e agilidade na criação e modificações na infraestrutura de provimento de funções de rede.

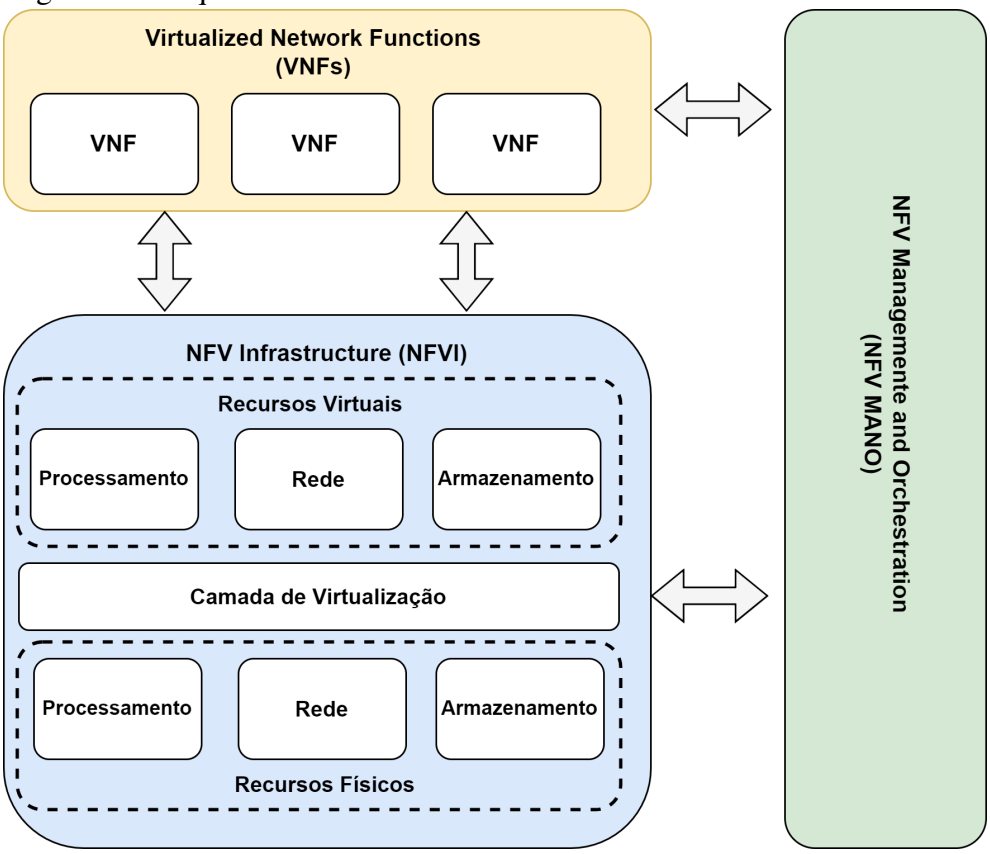
Assim, NFV tem como benefícios, entre outros: maior rapidez na implementação de novos serviços, redução de custos operacionais e de implantação, maior flexibilidade na atribuição de funções de rede, diminuição de consumo energético e maior abertura e padronização das funções, possibilitando o fornecimento de serviços por múltiplos fornecedores (ETSI, 2013).

Visando esses objetivos, muitas entidades têm concentrado esforços no sentido de delinear padrões que viabilizem a adoção de NFV (MIJUMBI *et al.*, 2016). Dentre essas iniciativas, destaca-se o *European Telecommunications Standards Institute* (ETSI), o qual lançou uma série de documentos com o intuito de construir e fortalecer a existência de funções de rede baseadas em softwares virtualizados. Uma das principais contribuições foi a especificação de um *framework* que sistematiza a estrutura necessária para a adoção de NFV. A figura 6 mostra a visão de alto nível do *framework*.

A arquitetura é composta por três camadas (YI *et al.*, 2018):

1. ***Network Function Virtualization Infrastructure* (NFVI)**: compreende toda a infraestrutura necessária à implementação de NFV, abrangendo os recursos de hardware (armazenamento, processamento e rede) e virtuais. Nessa camada, os recursos físicos dispostos em servidores genéricos são transformados em recursos virtuais por meio de softwares de virtualização (hipervisores), os quais são alocados posteriormente às várias funções de rede sob demanda. Nesse caso, os hipervisores criam uma espécie de subcamada de virtualização.
2. ***Virtualized Network Functions (VNFs)***: é formada por todas as funções de rede virtu-

Figura 6 – Arquitetura de referência ETSI NFV.



Fonte: adaptado de ETSI (2013)

alizadas e seus componentes, que podem ser alocadas em máquinas virtuais , do inglês *Virtual Machines (VMs)*, ou contêineres, de forma centralizada ou distribuída, providas pela NFVI.

3. ***NFV Management and Orchestration* (NFV-MANO)**: agrupa os componentes responsáveis por gerenciar todos os elementos da infraestrutura NFV, desempenhando tarefas como orquestração de recursos, gerenciamento do ciclo de vida de instâncias VNF, interface entre os módulos e mecanismos de virtualização.

Devido a suas vantagens, NFV tem se colocado como promissora para lidar com muitos desafios da IoT, podendo auxiliar na sua efetivação. A possibilidade de migração de funções em ambientes IoT para entidades virtualizadas possibilita o provimento de serviços adequados e especializados para esse domínio (ALAM *et al.*, 2020). Especificamente em relação à segurança, essa potencialidade pode ser explorada na construção de ambientes seguros para trocas de dados, que levem em consideração as restrições dos dispositivos e demandas existentes.

Nessa perspectiva, a seguir são destacadas algumas potencialidades da utilização de NFV para segurança dentro do paradigma da IoT (FARRIS *et al.*, 2019):

– **Restrições dos dispositivos:** como algumas funções de segurança demandam muito

processamento e armazenamento e esses recursos são escassos em dispositivos IoT, a migração destas para entidades virtualizadas pode viabilizar a utilização de técnicas de segurança mais robustas.

- **Escalabilidade:** as demandas em ambientes IoT podem variar em um curto período de tempo, no que diz respeito à quantidade de dispositivos, dados e requisições. Desse modo, NFV pode auxiliar a lidar com essa característica, provendo uma infraestrutura de funções de rede que se adeque à variação da demanda, aumentando e diminuindo o número de instâncias de uma função virtualizada conforme necessário e de forma simples, ou seja, pouco dispendiosa.
- **Heterogeneidade:** funções de segurança virtualizadas podem auxiliar administradores de segurança a implantar o mesmo nível de segurança em dispositivos heterogêneos por meio da adequação às especificidades de cada dispositivo.
- **Tolerância a falhas:** a orquestração de VNFs pode ser útil para criação de estruturas redundantes, aumentando a confiabilidade das funções de segurança, evitando indisponibilidade de serviço decorrente, por exemplo, de ataques de negação de serviço.
- **Mobilidade:** como essa é uma característica importante de muitas aplicações IoT, NFV pode auxiliar no suporte a esse requisito por meio da instanciação de funções virtuais próximo aos dispositivos IoT e migração destas de acordo com a locomoção e demandas dos dispositivos.
- **Flexibilidade:** o fato de funções virtualizadas serem exercidas por entidades de software abre espaço para uma maior flexibilidade no projeto e execução de mecanismos de segurança, adaptando-se às demandas de serviço e à realidade dos múltiplos cenários IoT.

Assim, é possível perceber o quanto funções de rede virtualizadas podem contribuir para o contexto de conexão atual, de forma especial auxiliando a tornar possível a efetivação da IoT. No tocante à segurança, essa tecnologia pode ser a viabilizadora de estruturas para IoT com um grau de segurança e privacidade adequados, protegendo aplicações, usuários e dispositivos. Além disso, NFV pode ser associada a outras tecnologias, como *Software Defined Network* (SDN) e *blockchain*, potencializando e evoluindo os mecanismos de proteção.

2.6 *Cloud, Fog e Edge Computing*

Os avanços nas áreas de computação e comunicação resultaram no aumento no volume de dados, das exigências de processamento e armazenamento e da complexidade das

aplicações. Nessa perspectiva, emergiram paradigmas de computação que propunham novas formas de organizar os recursos, dispositivos e a execução das funções necessárias aos diversos tipos de cenários. Os três principais são *Cloud Computing*, *Fog Computing* e *Edge Computing* e estão diretamente relacionados à IoT. Nesta seção, serão apresentados seus elementos principais e como estes podem ser empregados em estratégias de segurança em IoT, dentro do escopo ora considerado.

A mais consolidada entre essas abordagens é a *cloud computing* ou computação em nuvem, que surgiu da necessidade de prover uma maior capacidade de armazenamento e processamento além da existente nos dispositivos, bem como de possibilitar o compartilhamento de aplicações. Uma das definições mais recentes apresenta nuvem como sendo uma estrutura de computação que tem por objetivo fornecer qualquer coisa como um serviço, permitindo sua virtualização, agrupamento, compartilhamento, provisionamento e disponibilização de forma rápida, sem que haja um grande esforço para isso, do ponto de vista de gerenciamento (ELAZHARY, 2019).

Esses serviços são providos por servidores, que são usualmente organizados em grandes estruturas centralizadas de *data centers*, possibilitando uma enorme capacidade de processamento e armazenamento de dados. Para que isso ocorra, os dispositivos devem remeter os dados produzidos à nuvem via rede, a fim de serem processados, armazenados e disponibilizados a outros dispositivos (consumidores) via solicitações (SHI *et al.*, 2016). Assim, percebe-se um papel central da nuvem na disponibilização de dados e serviços, feita por meio de solicitações/respostas a dispositivos via tecnologias de comunicação.

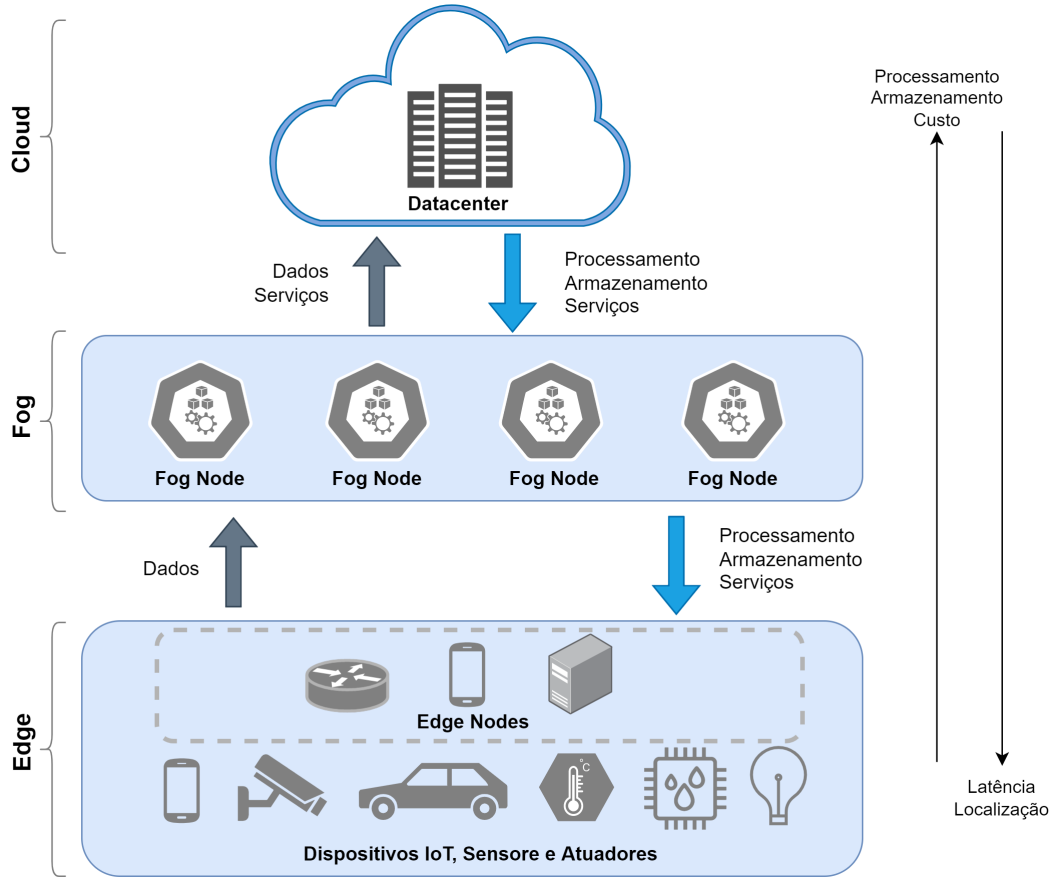
Usualmente, os três principais modelos de serviços disponibilizados em *cloud computing* são: Infraestrutura como Serviço (*Infrastructure as a Service* (IaaS)), plataforma como serviço (*Platform as a Service* (PaaS)) e software como serviço (*Software as a Service* (SaaS)) (KHAN *et al.*, 2019). Independente do tipo de serviço, *cloud computing* tem como características principais: atendimento sob demanda sem intervenção do usuário, amplo acesso de recursos via rede, agrupamento de recursos de modo a atender às solicitações, elasticidade na liberação de recursos e controle no uso dos serviços (MELL *et al.*, 2011).

Apesar dos benefícios obtidos pela computação em nuvem, esse modelo foi se tornando ineficiente com o passar do tempo, especialmente com o crescimento gradual da IoT. Isso deve-se basicamente ao que segue: (i) o aumento expressivo na quantidade de dispositivos aumentou o consumo de largura de banda em interações com a nuvem, além do que grande parte

desses dados fosse usada localmente; (ii) o fato da nuvem ser uma estrutura centralizada aumenta a latência nas solicitações e impossibilita o uso de contexto local, dois atributos necessários a muitos tipos de aplicações; (iii) as restrições de recursos dos dispositivos IoT dificultam a implantação dos protocolos envolvidos nos processos de solicitação e resposta, e (iv) a grande quantidade de dispositivos IoT necessita de uma ampla distribuição geográfica das estruturas de armazenamento e processamento, o que é incompatível com o modelo de computação em nuvem usualmente adotado (SHI *et al.*, 2016; DONNO *et al.*, 2019; KHAN *et al.*, 2019).

Assim, os paradigmas de *Fog* e *Edge Computing* foram propostos para lidar com essas questões. A Figura 7 apresenta a relação entre os três paradigmas num contexto de IoT. A diferença entre essas abordagens está basicamente no local onde as tarefas de processamento e armazenamento são desempenhadas, sendo que, nos níveis *fog* e *edge* busca-se trazê-las para mais próximo dos dispositivos, como forma de reduzir a latência, não substituindo a nuvem, mas podendo trabalhar associadamente (MIJUSKOVIC *et al.*, 2021). O trabalho de Kalyani e Collier (2021), por exemplo, aborda como as características dessas três abordagens podem ser exploradas no domínio de *smart agriculture*.

Figura 7 – Organização *Cloud, Fog* e *Edge Computing*.



Fonte: adaptado de Donno *et al.* (2019), Laroui *et al.* (2021)

Uma definição que consegue abarcar suficientemente o conceito de *Fog Computing* é dada por Yi *et al.* (2015), que define como sendo uma arquitetura com distribuição geográfica formada por dispositivos heterogêneos conectados de forma ubíqua situados na borda da rede, que podem ou não ser independentes da nuvem para fornecer elasticamente serviços de computação, armazenamento e comunicação aos clientes na proximidade. Nessa arquitetura, os dispositivos (*fog nodes*) possuem os componentes físicos e lógicos necessários às atividades de processamento, transmissão e armazenamento, localizando-se de forma distribuída entre a nuvem e os dispositivos na borda da rede (DONNO *et al.*, 2019).

Assim, pode-se depreender que uma *Fog* atua como intermediária entre a nuvem e os dispositivo. Por um lado, provê um conjunto de serviços aos dispositivos antes disponibilizados pela nuvem centralizada, mas com menor latência devido a maior proximidade com estes. Por outro, fornece e obtém serviços da própria nuvem, como disponibilizar dados a respeito dos dispositivos e armazenamento e processamento mais densos, respectivamente.

Seguindo a linha de aproximar-se cada vez mais dos dispositivos a fim de diminuir a latência e aumentar a percepção de contexto, a *Edge Computing* compreende o conjunto de tecnologias que permite que as atividades computacionais sejam desempenhadas na borda da rede, ou seja, onde os dispositivos estão (SHI *et al.*, 2016). Nessa proposta, dispositivos com capacidade computacional maior dentro de uma rede, como gateways ou smartphones, são capazes de desempenhar algumas tarefas para outros dispositivos a eles conectados. Assim, são criados vários pontos de fornecimento de serviços distribuídos amplamente e próximos aos dispositivos que fornecem e consomem os dados, criando um modelo mais condizente com as demandas de ambientes IoT.

Apesar de possuírem semelhanças entre si, como distribuição geográfica, diminuição da latência e descentralização, *fog* e *edge* configuram-se como paradigmas distintos. Enquanto que *fog computing* assemelha-se mais ao modelo de serviço de computação em nuvem, sendo baseada em microcontroladores e microcomputadores, provendo, portanto, recursos intermediários, a *edge computing* possui menor capacidade computacional, uma vez que depende dos recursos disponibilizados pelos dispositivos, com maior grau de proximidade (DONNO *et al.*, 2019; KALYANI; COLLIER, 2021).

As propriedades e potencialidades desses três paradigmas podem ser aproveitadas para criar soluções de seguranças ou mesmo lidar com aspectos inerentes a essa questão no âmbito da IoT. Entre as principais contribuições que podem ser destacadas estão (HASSIJA *et*

al., 2019; SHA *et al.*, 2020):

a) A estrutura computacional mais robusta disponibilizada pela nuvem e nós das arquiteturas de borda pode colaborar com a implantação de esquemas de segurança mais completos e antes dificultados pelas restrições dos dispositivos, fazendo com que funções de segurança que demandem maior poder de processamento sejam desempenhadas por outras entidades próximas ao dispositivo IoT.

b) O alto grau de distribuição provido, especialmente na computação em névoa e de borda, podem auxiliar na implementação de sistemas tolerantes a falhas e com alta escalabilidade, fazendo com que as funções de segurança tenham maior disponibilidade, além de evitar que os dados tenham que trafegar por muitos saltos na rede entre origem e destino.

c) O fato de haver uma camada intermediária antes dos dispositivos IoT colabora para a proteção dos dados, uma vez que o gerenciamento e armazenamento executados nessas estruturas são mais robustos em comparação aos dispositivos IoT.

d) A proximidade dos nós de névoa e borda em relação aos dispositivos torna possível o uso de informações do contexto do usuário, do dispositivo ou da rede onde estes se encontram para criar mecanismos de segurança adaptativos à situação do cenário ou das necessidades dos usuários, com o uso, por exemplo, de políticas de controle de acesso e autenticação específicos.

Entretanto, a junção dessas abordagens com a IoT introduz também desafios relacionados à segurança, como a proteção da privacidade dos dados durante o armazenamento temporário e a detecção de atividades maliciosas em aplicações com altas restrições de tempo de execução (HASSIJA *et al.*, 2019). Independente disso, são notórias as possibilidades existentes, que apontam diversos caminhos para a criação de um ambiente de compartilhamento de dados seguro diante desse cenário em constante evolução.

2.7 Considerações finais

Nesse capítulo foram apresentados conceitos fundamentais para traçar um panorama geral da problemática abordada no trabalho. Apesar de muito estudada, a segurança em IoT ainda coloca-se como um desafio com múltiplas questões que merecem atenção. Quando inseridas as especificidades trazidas pelas aplicações sensíveis ao contexto, algumas dessas questões são

adensadas e outras são introduzidas.

As metodologias e técnicas ora expostas apontam possíveis soluções. O modelo ABAC pode prover um método de autorização que se adapte diante das múltiplas fontes, origens e usos das informações contextuais. CAS tem a potencialidade de aumentar o grau de segurança de acordo com a necessidade, tomando por base informações contextuais. E os paradigmas de computação em nuvem, névoa e borda, bem como a virtualização de funções de rede, auxiliam a lidar com questões relacionadas às dificuldades de implantação de segurança em ambientes IoT.

No capítulo seguinte, são apresentados trabalhos que se relacionam com o escopo aqui abordado, alguns dos quais utilizando técnicas e estratégias expostas nesse capítulo as quais ajudarão a entender como tais recursos vêm sendo aproveitados.

3 TRABALHOS RELACIONADOS

O tópico de segurança em IoT vem sendo explorado há alguns anos, dado o crescimento e desafios existentes nesse paradigma. Do mesmo modo, o compartilhamento de contexto continua sendo alvo de esforços de pesquisa na área da computação, sobretudo nos nichos de computação ubíqua e pervasiva, muitos dos quais convergem para ambientes IoT. Assim, existem muitos trabalhos propondo soluções de segurança para compartilhamento de informações no âmbito da IoT, sob várias perspectivas e fazendo uso de diferentes técnicas. O trabalho de Perera *et al.* (2014) sumariza algumas dessas propostas de compartilhamento de contexto em IoT, analisando-as e destacando alguns desafios, dentre os quais, segurança e privacidade.

Neste capítulo, são apresentados alguns trabalhos relacionados à problemática tratada nessa pesquisa, destacando sobretudo como segurança e privacidade são implementadas nas soluções proposta. Ao final, esses trabalhos são analisados e comparados com a arquitetura proposta nesta dissertação.

3.1 Seleção dos trabalhos

Dada a grande quantidade de propostas existentes que abordam segurança em IoT e/ou compartilhamento de contexto, optou-se por realizar uma investigação de literatura de caráter exploratório, visando identificar trabalhos que contribuíssem para a discussão da proposta ora formulada, bem como possibilitassem visualizar como técnicas de segurança que embasam a arquitetura vêm sendo usadas.

Assim, foram selecionados trabalhos que atendessem a pelo menos dois dos critérios a seguir:

i Similaridade do escopo com a proposta deste trabalho;

ii Utilização do modelo ABAC como alternativa para controle de acesso e autorização;

iii Uso de informações de contexto para fins de segurança de algum modo, implementando segurança ciente de contexto.

iv Proteção de informações de contexto durante a fase de compartilhamento.

Considerando tais critérios, as buscas foram feitas principalmente nas bases de dados *IEEE Xplore* e *Scopus*, visto que estas agregram boa parte da literatura significativa sobre os temas tratados. Nas buscas, foram utilizadas composições de termos como *((“Internet of Things”*

OR "IoT") AND (security OR privacy OR "access-control")) AND (("context-aware" OR "context-awareness") AND (security OR privacy OR "access-control") AND ("attribute-based access control" OR ABAC)), as quais foram adaptadas conforme a necessidade das plataformas usadas. Como recorte temporal, foi definido que seriam considerados trabalhos de 2015 até 2021. Adicionalmente, foram excluídos artigos de revisão de literatura.

Após a leitura dos resumos dos trabalhos retornados e da análise daqueles que mais se adequavam aos objetivos ora propostos, foram selecionados dez trabalhos que serão apresentados na seção seguinte.

3.2 Trabalhos selecionados

Liu (2015) propõem uma abordagem de compartilhamento seguro de contexto em ambientes pervasivos com conexões oportunísticas denominado Magpie, que utiliza avaliação dinâmica de confiança e preservação de privacidade. Ao receber uma requisição, o dispositivo avalia o grau de confiança com base no histórico e na frequência de interações, similaridade de contexto e ambiente de compartilhamento. O resultado dessa avaliação e o grau de sensibilidade da informação requerida são usados para decidir sobre a estratégia de preservação de privacidade (compartilhamento completo, ofuscamento de dados ou negação). Apesar de sua operação dinâmica, o Magpie pode demandar muito processamento e armazenamento de informações, uma vez que todo o processo de avaliação é feito pelo dispositivo IoT, o que inviabiliza sua adoção em muitos cenários. Além disso, não tem uma implementação flexível, por ser projetado para um tipo específico de comunicação.

Também fazendo uso de avaliação de confiança, Wang *et al.* (2017) projetaram um modelo de controle de acesso baseado em ABAC inserindo esse recurso para fins de autenticação. Um módulo faz a avaliação de confiança com base em certificados digitais fornecidos pela aplicação solicitante, resultando em um nível de 1 a 5. Essa classificação é usada junto com os atributos do requisitante na avaliação de políticas de controle de acesso. Embora a classificação de confiança adicione um nível a mais de segurança, ela se baseia em certificados, o que demanda armazenamento e pode reduzir a dinamicidade da solução. Adicionalmente, as informações não são criptografadas antes do compartilhamento, o que compromete a confidencialidade.

Em seu trabalho, Ramos *et al.* (2015) propõem um *framework* para segurança adaptativa ciente de contexto focado em permitir adaptações de segurança por parte dos dispositivos IoT. Um componente denominado *security manager* é responsável por implementar os mecanismos

de gerenciamento de identidade, autorização via *token* de acesso, gerenciamento de confiança e reputação e criptografia, com base nas informações de contexto. Por um lado, a proposta prevê um bom grau de flexibilidade de implementação e lida bem com a dinamicidade da IoT. Entretanto, as funções de segurança envolvem uma grande participação dos dispositivos, tanto na captação de contexto quanto nas decisões de segurança, além de adotar apenas um esquema de criptografia.

Utilizando *Fog* e *Edge Computing*, Matos *et al.* (2018) definem uma arquitetura de compartilhamento de contexto com ciência de contexto para decisões de segurança. A arquitetura é composta de dois módulos: um implementado em nível de *fog* e responsável pelo compartilhamento de contexto e outro que gera as informações e toma das decisões de segurança. São utilizadas regras pré-definidas para o domínio e histórico de contexto de eventos para definir quais regras devem ser aplicadas. Apesar de utilizar políticas, estas são centradas no domínio de implementação, o que pode limitar a flexibilidade de implementação da arquitetura.

Gabillon *et al.* (2020) definem um modelo de autorização para IoT baseado em ABAC para sistemas que usam *Message Queuing Telemetry Transport* (MQTT) como protocolo de compartilhamento, no qual políticas regulam a inscrição em tópicos e o envio das informações. A linguagem das políticas é baseada em lógica de primeira ordem e permite a especificação de forma flexível e considerando o contexto das ações, especialmente a frequência de ocorrência de eventos. O fato dessa proposta focar apenas em um protocolo, restringe sua aderência a cenários MQTT. Adicionalmente, a confidencialidade fica a cargo somente da autorização de inscrição e envio de mensagens, não havendo mecanismos de criptografia ou outros que fortaleçam esse requisito.

Arfaoui *et al.* (2019) propõem uma abordagem de controle de acesso baseada em atributo contextual que usa informações de contexto e criptografia baseada em atributo de política de texto cifrado (*Ciphertext-Policy Attribute-Based Encryption* (CP-ABE)) para prover segurança e privacidade aos dados compartilhados por dispositivos inteligentes. Chaves simétricas são atribuídas a entidades e *tokens* de autenticação obtidos a partir do contexto do usuário são usados para controlar o acesso a recursos. A criptografia é realizada por um gateway IoT e a decriptação dos dados depende do cumprimento de atributos especificados nas políticas e de *tokens* de autenticação válidos. Como pontos negativos, destaca-se a utilização de autoridades de fornecimento de atributos e geração de chaves e o uso de apenas um tipo de criptografia, o que pode limitar a aplicação da proposta.

Com foco em sistemas sensíveis ao contexto, Mahalle e Dhotre (2020) propõem um modelo conceitual de framework para atuar entre fontes de contexto e aplicações. Uma entidade de software, denominada agente de segurança, é encarregada de garantir privacidade, confidencialidade e integridade das informações de contexto. O algoritmo proposto utiliza listas de controle de acesso, perfis de usuário, políticas e dados das solicitações para classificar níveis de segurança e de confiança e assim decidir sobre informações e o conhecimento das entidades que requisitarão contexto, o que dificulta a implementação em ambientes IoT. Além disso, não são fornecidos detalhes a respeito de como são determinados os níveis de segurança e confiança, bem como sobre a implementação.

Em seu trabalho, Sylla *et al.* (2019) expõem uma arquitetura de segurança e privacidade com ciência de contexto para *smart cities* centrada no usuário. Os módulos da arquitetura são executados como funções virtuais de rede em nós próximos aos usuários organizados em dois planos: (i) segurança e privacidade e (ii) de conhecimento. O plano de segurança e privacidade provê segurança contextual por meio da implementação de políticas que são adotadas com base no contexto coletado pelo módulo de conhecimento e nas informações solicitadas. Reputação dos dispositivos e *blockchain* são usados para confiança e autorização respectivamente. Embora seja conceitualmente flexível e dinâmica, a grande quantidade de módulos pode tornar sua implementação difícil em ambientes menos complexos. Além disso, as funções de segurança dependem de uma intensa captação de contexto, o que pode sobrecarregar os dispositivos IoT.

Focado na proteção de registros eletrônicos de saúde em ambientes de nuvem, Psarra *et al.* (2019) propõem um modelo de segurança sensível ao contexto no qual as decisões de controle são baseadas em ABAC, considerando informações sobre o contexto do usuário e da requisição. Relativo a autorização, *Attribute-Based Encryption* (ABE) é usado para criptografar as chaves utilizadas pelos proprietários para criptografar os dados. Embora utilize ABAC e considere informações de contexto para segurança, o modelo é focado apenas na proteção de dados em nuvem, não levando em consideração muitos aspectos da IoT.

Al-Muhtadi *et al.* (2021) propõem uma estrutura de segurança para ambientes pervasivos de compartilhamento de contexto. A autenticação de sensores e aplicações para acesso ao sistema é feita por meio de certificados digitais baseada no protocolo *Cerberus*. Após autenticada, uma entidade é associada a *brokers*, que a atribui chaves usadas para encriptar os dados. Além disso, algoritmos de *hash* e assinaturas digitais são utilizados para prover integridade. Embora os experimentos tenham demonstrado a eficiência em termos de desempenho e garantia de

Tabela 2 – Trabalhos relacionados.

Trabalho	Domínio	Controle de Acesso	Criptografia	Privacidade	Flexibilidade	Contexto para Segurança	Proteção de Contexto
Liu and Julian (2015)	Ambientes pervasivos	- Não especificado	Ausente	- Avaliação de confiança - Ofuscamento de contexto	Baixa	Sim	Sim
Wang et al. (2015)	IoT	- ABAC	Ausente	- Avaliação de confiança	Baixa	Não	Sim
Ramos et al. (2015)	IoT	- Tokens de acesso	CP-ABE	- Gerenciamento de identidade - Avaliação de confiança e reputação	Média	Sim	Sim
De Matos et al. (2018)	IoT	- Políticas de acesso	Ausente	- Avaliação de solicitação - Histórico de interações - Regras de segurança	Alta	Sim	Sim
Arfaoui et al. (2019)	IoT	- Tokens de acesso	CP-ABE	- Políticas de criptografia baseadas em contexto	Baixa	Sim	Sim
Gabilon et al. (2020)	Ambientes MQTT	- ABAC	Ausente	- Políticas de controle de acesso	Baixa	Sim	Indiretamente
Mahalle and Dhotre (2020)	Sistemas sensíveis ao contexto	- Perfis de usuário - Listas de controle de controle de acesso	Ausente	- Avaliação de confiança - Níveis de segurança	Baixa	Parcialmente	Sim
Sylla et al. (2020)	Smart city	- Blockchain	Ausente	- Avaliação de contexto para decisões de segurança	Alta	Sim	Indiretamente
Psarra et al. (2020)	Nuvem	- ABAC	ABE	- Encriptação pelo usuário	Baixa	Sim	Não
Al-Muhtadi et al. (2021)	Ambientes pervasivos	- Certificados digitais	Chave simétrica	- Controle de acesso	Média	Não	Sim
Este trabalho (FCAAS-IoT)	IoT	- ABAC	Adaptável	- Controle de acesso centralizado - Políticas de encriptação	Alta	Sim	Sim

requisitos de segurança, o uso de certificados digitais e as chaves envolvidas no processo de criptografia podem não ser adequados a muitos ambientes IoT. Além disso, a autorização não considera políticas de privacidade do usuário.

Na seção seguinte, os trabalhos aqui apresentados são comparados em relação à arquitetura proposta como produto desta pesquisa, ressaltando aspectos relevantes que são mantidos ou se diferenciam.

3.3 Análise comparativa dos trabalhos

A Tabela 2 traz o comparativo entre os trabalhos apresentados e a arquitetura proposta, resumindo aspectos como o domínio para o qual foram projetados, a flexibilidade para adaptação a vários cenários, técnicas de controle de acesso e criptografia adotadas, mecanismos de garantia de privacidade, uso de informações contextuais para segurança e aplicação ou foco na segurança no compartilhamento de contexto.

Em relação a flexibilidade, a maioria dos trabalhos apresenta um baixo potencial de adaptabilidade, tanto levando em conta a implementação física quanto os mecanismos de

segurança adotados. Quando considera-se a heterogeneidade inerente à IoT e a diversidade de cenários que podem ser criados dentro desse paradigma, a adoção de artefatos e recursos menos flexíveis pode ser um fator limitante para a consolidação da segurança. Adicionalmente, é necessário levar em conta que esse paradigma encontra-se ainda em processo de definição objetivando sua ampla disseminação, o que pode demandar alterações nos esquemas de segurança, fazendo com que esse aspecto da flexibilidade seja um ponto importante a ser considerado.

Sobre as técnicas para controle de acesso, a maior parte dos trabalhos considerados utiliza políticas para definir os procedimentos e/ou requisitos de autorização. Nesse sentido, o modelo ABAC apresentou uma boa aderência. Isso denota que, em ambientes IoT, a utilização de políticas pode ser viável para implementar mecanismos que satisfaçam as condições de segurança necessárias, de forma similar aos sistemas convencionais. Entretanto, é necessário salientar que, nos trabalhos analisados, os mecanismos de controle de acesso não sugerem formas de inserção do usuário no estabelecimento de políticas o que pode ser um elemento importante, sobretudo quando consideramos informações críticas, como as que são usadas por aplicações sensíveis ao contexto.

A utilização de criptografia foi outro ponto analisado. Percebeu-se que esse recurso é parcialmente utilizado nos trabalhos considerados, sobretudo naqueles focados em controle de acesso. Ou seja, em muitos deles, os dados são compartilhados sem que haja um processo de encriptação anterior, o que pode comprometer o grau de confidencialidade e privacidade. Além disso, observou-se que, naqueles que preveem a encriptação de dados, apenas um algoritmo é utilizado, não havendo abertura para a adoção de múltiplos esquemas que possam adequar-se a cada cenário.

Focando na garantia de privacidade, destaca-se o uso de avaliação de confiança e utilização de informações para classificação de níveis de segurança e de confidencialidade para compartilhamento dos dados, os quais são comumente associados a esquemas de controle de acesso. Ambos mecanismos reafirmam o potencial da utilização de informações correntes para o aprimoramento de decisões de segurança, criando recursos mais dinâmicos, capazes de executar uma proteção mais adequada a cada situação.

No entanto, é necessário ponderar que essas informações devem ser coletadas de modo a garantir a confiabilidade de origem e mensurar o impacto do processo de coleta e análise no funcionamento geral das aplicações, principalmente em relação a requisitos computacionais e à complexidade e volume de informações envolvidas no processo. Tais preocupações são

relevantes a fim de não inviabilizar a adoção dos mecanismos em ambientes restritos nem impactar de forma negativa no desempenho das aplicações e provimento dos serviços.

Em relação à proteção de informações de contexto durante o compartilhamento, a maioria das propostas consegue dar suporte de alguma forma na etapa de compartilhamento, uma vez que são voltadas à proteção de dados nos processos de comunicação. Entretanto, uma solução que permita considerar os vários cenários de compartilhamento e uso de contexto, do ponto de vista de requisitantes, fornecedores e agrupamento de informações, pode fornecer uma segurança aprimorada, principalmente considerando questões de privacidade.

Assim, a arquitetura ora proposta pode ser comparada em relação aos trabalhos expostos pelo que segue:

a) Possui um alto potencial de flexibilidade de implementação, podendo ser adaptada aos vários cenários da IoT, no que diz respeito a infraestrutura, domínios e especificidades de cada caso;
b) Utiliza o contexto da solicitação para decisões de criptografia obtido por um componente da própria arquitetura, o que possibilita a adoção de vários esquemas criptográficos e sua utilização conforme a necessidade para um determinado arranjo de solicitação;
c) Uso de políticas adaptáveis para modelar o controle de acesso e gerenciar os mecanismos de criptografia, as quais podem ser adaptadas às demandas de privacidade dos usuários e das organizações, por se basearem em atributos e contextos das solicitações em seu funcionamento e construção;
d) Fornecer acesso controlado ao contexto através da centralização dessas informações e possibilidade de aplicação de diferentes políticas para cada informação. Além disso, as informações contextuais são obtidas e fornecidas por módulos da arquitetura, o que impede o acesso direto às fontes, favorecendo a privacidade e proteção.

3.4 Considerações finais

Esse capítulo expôs trabalhos relacionados à pesquisa e foi feita a comparação destes com a proposta delineada. Identificou-se que os princípios que norteiam a proposta formulada estão alinhados ao que vem sem sendo pensado e discutido sobre o assunto, com foco em compartilhamento de contexto. Além disso, destaca-se a carência de artefatos com um bom grau

de flexibilidade para adaptar-se aos vários cenários IoT de fornecimento de dados.

4 PROPOSTA DE ARQUITETURA

Neste capítulo, é apresentada a arquitetura proposta, denominada *Flexibility for Context-Aware Applications Security in IoT* (FCAAS-IoT), que tem o objetivo de fornecer segurança no compartilhamento de contexto para aplicações, por meio de acesso controlado ao contexto e encriptação de mensagens baseando-se em informações de contexto de solicitação, com capacidade de ser adaptada para implantação em vários cenários IoT.

Primeiramente é dada uma visão geral da arquitetura, explicando as estratégias adotadas e apresentando os módulos que a compõem. Em seguida, são abordadas as políticas usadas nas funções de segurança, expondo sua estrutura e elementos fundamentais, bem como questões inerentes aos seus papéis nos mecanismos de segurança propostos. Após isso, é explicado o funcionamento da arquitetura, abordando cada fase de atuação e os papéis dos módulos. Por fim, é apresentado um cenário de aplicação da arquitetura, a fim de facilitar a visualização e compreensão de sua atuação, bem como discutir possibilidades e aspectos de implementação.

4.1 Princípios Norteadores

O principal objetivo da FCAAS-IoT é possibilitar que informações de contexto sejam compartilhadas entre fontes geradoras e as aplicações solicitantes em ambientes IoT garantindo privacidade e confidencialidade. Para tanto, a arquitetura baseia-se no modelo ABAC de controle de acesso e usa informações sobre o contexto das solicitações para definir os esquemas de criptografia a serem aplicados, utilizando, em ambos casos, políticas para gerir as funções de segurança.

Nesse sentido, FCAAS-IoT utiliza as seguintes estratégias:

a. **Acesso controlado ao contexto:** as informações de contexto são centralizadas em um componente e são acessadas por meio de um processo de autorização baseado no modelo ABAC, coordenado por políticas que decidem sobre a permissão ou negação de acesso. Assim, é possível controlar quais informações são disponibilizadas, sem que haja uma conexão direta entre fornecedores e consumidores, o que contribui para uma maior privacidade.
b. **Criptografia ciente de contexto:** a arquitetura prevê a utilização de políticas construídas levando em consideração o contexto das solicitações para determinar

qual algoritmo será utilizado na encriptação das informações de contexto em um dado evento de compartilhamento. Desse modo, é possível, por exemplo, implementar vários algoritmos no sistema, o que pode contribuir para a adaptabilidade da arquitetura às demandas do ambiente IoT, como tipo de dispositivos e padrão de segurança adotado, aumentando o nível de confidencialidade do sistema;

c. **Flexibilidade de implementação:** a divisão dos componentes da FCAAS-IoT em módulos agrupando e definindo suas funções e relações, possibilita uma implementação flexível da arquitetura, podendo se adequar às necessidades da organização e às especificidades do cenário considerado. É possível, por exemplo, optar por uma estruturação centralizada ou distribuída, ou pela replicação de algum dos módulos mediante o arranjo e demandas existentes no ambiente a ser implementado;

d. **Adequação de privacidade:** a utilização de atributos como elementos de controle de acesso e de informações sobre o contexto da solicitação para decisões de criptografia permite a elaboração de políticas que respeitem as opções de privacidade dos usuários proprietários das informações de contexto e as diretrizes de segurança das organizações, o que resulta em um bom potencial adaptabilidade a diversos cenários e em um maior controle sobre o uso das informações de contexto. Além disso, as políticas podem ser modificadas, adicionadas e excluídas de modo a refletir os arranjos de segurança desejados.

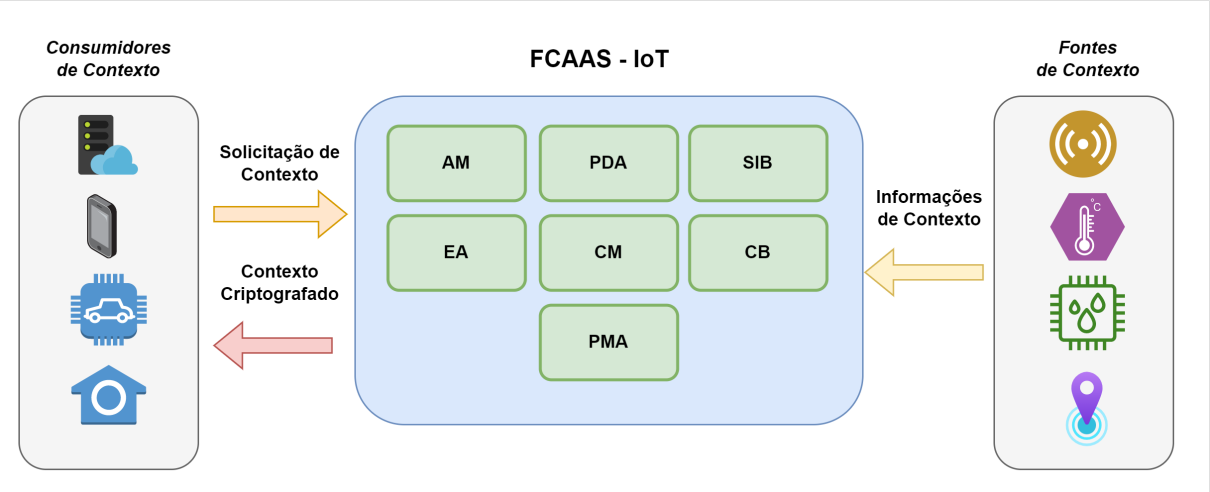
A seguir, são apresentados os módulos que compõem a FCAAS-IoT, trazendo uma visão geral do seu funcionamento e da atuação da arquitetura. Mais adiante, detalham-se as etapas de funcionamento e o papel de cada um dos módulos e sua comunicação de forma mais detalhada.

4.2 Visão Geral

A FCAAS-IoT é composta por módulos que desempenham funções específicas e agem conjuntamente para analisar requisições de informações de contexto provenientes de entidades e implementar regras de segurança garantindo privacidade e confidencialidade. Cada módulo é executado de modo independente funcionalmente, o que garante uma maior flexibilidade na implementação e a possibilidade de lidar com demandas inerentes à IoT e necessárias a mecanismos de segurança como escalabilidade, heterogeneidade e tolerância a falhas. A figura 8

apresenta uma visão de alto nível da arquitetura.

Figura 8 – FCAAS-IoT: visão geral.



Fonte: elaborado pelo autor.

A proposta é que os módulos funcionais atuem como intermediários entre consumidores e fontes de contexto, podendo ser implementados e geridos por uma ou várias organizações. No âmbito da FCAAS-IoT, denomina-se consumidor de contexto (*context consumer*) qualquer aplicação, dispositivo ou serviço que requisite informações de contexto e as utilize em sua operação para provimento de algum serviço. Fonte de contexto (*context source*) é qualquer entidade (sensores, objetos inteligentes, *smartphones* etc.) que gere informações de contexto que possam ser disponibilizadas a terceiros. Por fim, alinhado às definições de Abowd *et al.* (1999) e Dey (2001), considera-se contexto como sendo qualquer informação ou conjunto de informações gerado por uma fonte e que traga características sobre sua entidade originadora ou sobre o ambiente no qual esta encontra-se inserida e que possa ser usada para o provimento de algum serviço ou execução de alguma atividade.

Cabe ressaltar, entretanto, que, devido à natureza da IoT, uma mesma entidade pode atuar simultaneamente como consumidor e fonte de contexto. Por exemplo, um smartphone pode executar aplicações que dependam de informações de contexto de outras entidades ao mesmo tempo que fornece informações contextuais captadas por seus sensores a outros consumidores de contexto. Tal situação é contemplada no FCAAS-IoT, visto que é possível definir políticas para cada informação de contexto fazendo distinção das fontes.

No que diz respeito aos componentes da arquitetura, as funções dos módulos da FCAAS-IoT são resumidas a seguir:

a. ***Authorization Manager* (AM):** faz a interface entre a arquitetura e os consu-

midores de contexto, recebendo as requisições e retornando as informações contextuais após a execução dos procedimentos de segurança ou a notificação de negação de acesso. Além disso, esse componente é responsável pela extração de atributos, tanto os fornecidos pela entidade solicitante quanto os relacionados à própria requisição, colocando-os em um formato adequado à execução das demais etapas, comunicando-se com os demais módulos.

b. ***Policy Decision Agent*** **(PDA):** módulo responsável pela avaliação das políticas, tanto de controle de acesso quanto de gerenciamento de criptografia. Para tanto, o PDA recupera as políticas armazenadas na SIB de acordo com as informações de contexto solicitadas e usa os atributos enviados pelo AM para decidir sobre a permissão ou negação de acesso e a respeito dos procedimentos de criptografia a serem adotados.

c. ***Security Information Base*** **(SIB):** consiste em uma base de dados que armazena as políticas de controle de acesso e de gerenciamento de criptografia e é acessada durante o processo de autorização pelo PDA. Além disso, o PMA acessa a SIB para operações de gerenciamento das políticas, como modificações, exclusões e inserções.

d. ***Context Broker*** **(CB):** módulo estruturado em forma de tópicos, atuando como um broker, que armazena as informações de contexto enviada pelas fontes e as fornece conforme solicitado pelo CM para composição do contexto requisitado pelo consumidor.

e. ***Context Manager*** **(CM):** elemento que obtém as informações de contexto solicitadas segundo as especificações feitas pelo AM, solicitando-as ao CM e organizando-as em uma estrutura de contexto em formato adequado ao compartilhamento.

f. ***Encryption Agent*** **(EA):** a função básica desse módulo é realizar os processos inerentes à encriptação das informações de contexto fornecidas pelo CM antes de serem enviadas ao AM para serem entregues aos consumidores. Para tanto, o EA deve ser capaz de executar os algoritmos de criptografia e outros procedimentos relacionados a essa função de acordo com o definido nas políticas de gerenciamento de criptografia e recuperadas pelo PDA durante a autorização.

g. ***Policy Management Agent*** **(PMA):** esse elemento representa o agente responsá-

vel pelo gerenciamento das políticas na SIB, realizando operações de inserção, exclusão e modificação de acordo com as necessidades da organização.

A execução coordenada das funcionalidades de cada módulo interagindo com os demais consegue receber, analisar e atender às solicitações de consumidores de contexto, provendo informações de forma segura tal qual projetado para cada caso. Além disso, enxergar os módulos orientado às atividades que devem desempenhar para o provimento das funções de segurança é útil para planejar sua implantação. Por exemplo, em ambientes com menor demanda de solicitações e/ou quantidade de consumidores e fontes de contexto, pode-se optar por uma implantação centralizada, ao passo que, em ambientes que impliquem necessidade de escalabilidade, pode-se pensar em uma implantação distribuída com replicação dos módulos. Em ambos casos, basta garantir que cada instância consiga executar suas atividades no âmbito das funções de segurança, o que denota um elemento a favor da flexibilidade proposta pela arquitetura.

4.3 Políticas

Políticas são uma ferramenta comum em mecanismos de segurança, utilizadas para definir regras e procedimentos para atingir um determinado objetivo, como o acesso a um recurso, por exemplo. FCAAS-IoT também faz uso de políticas para determinar as diretrizes a serem seguidas durante a execução das funções de segurança com o objetivo de permitir o compartilhamento de informações de contexto garantindo a confidencialidade e a privacidade.

FCAAS-IoT utiliza dois tipos de políticas: (i) de controle de acesso e (ii) de gerenciamento de criptografia. As políticas foram projetadas com o objetivo de serem flexíveis no que diz respeito à sua implementação, possibilitando uma adequação a vários cenários e tipos de informações, visto que a heterogeneidade é algo inerente à IoT e que contextos podem ser compostos por vários tipos de informações. Assim, do ponto de vista conceitual, as políticas são construídas e estruturadas com base em XACML (do inglês, *eXtensible Access Control Markup Language*)[1] e no modelo ABAC.

As políticas são armazenadas na SIB e utilizadas durante a fase de autorização, sendo acessadas pelo PDA conforme as demandas de requisições de contexto geradas e os atributos fornecidos. O PMA é responsável pelo gerenciamento das políticas na SIB, realizando operações de inserção, consulta, modificação e exclusão. Cabe destacar que neste trabalho não é

[1] https://www.oasis-open.org/committees/tc$_h$*ome.php*?*wg*$_a$*bbrev* = *xacml*

fornecido um modelo rígido de definição de políticas e nem responsabilidades no que diz respeito à administração e definição destas. Por outro lado, busca-se expressar como elementos mínimos podem ser combinados para implantar o esquema de segurança proposto. Nas subseções a seguir, são abordados cada tipo de política separadamente, trazendo explicações e questões inerentes a cada um.

4.3.1 Políticas de controle de acesso

As políticas de controle de acesso visam especificar requisitos relacionados a atributos a serem cumpridos para permitir ou negar o acesso a uma informação. Como a metodologia utilizada para autorização em FCAAS-IoT é o modelo ABAC, os campos que estruturam as políticas de controle de acesso foram organizados conforme a documentação que define esse modelo, que pode ser encontrada em Hu *et al.* (2013) e baseia-se em XACML. O Código-fonte 1 apresenta um exemplo de estrutura básica de organização dos campos desse tipo de política, modelada em formato JSON.

Código-fonte 1 – Exemplo de estrutura de política de controle de acesso.

```
{
    "id": <string>,
    "targets": {<string>, <string>, <string>},
    "rules": {
        "subject": {<attributes_and_conditions>},
        "resource": {<attributes_and_conditions>},
        "context": {<attributes_and_conditions>},
    },
    "permission": <allow or deny>,
    "priority": <string>
}
```

A seguir cada campo da política é explicado:

a) **Identificador (*id*)**: campo utilizado para identificar a política de forma única para fins de gerenciamento e organização, facilitando, por exemplo, a exclusão ou modificação de uma política;

b) **Alvos (*targets*)**: especificam sob quais elementos a política atua, ou seja, em que situações a política deve ser analisada a fim de verificar sua aplicabilidade. Nesse campo podem ser discriminadas as informações de contexto e os dispositivos fornecedores destas para que, quando esta informação for solicitada, a política seja analisada;

c) **Regras (*rules*)**: esse campo elenca os atributos e condições a serem satisfeitas para verificar a aplicabilidade ou não da política em questão. Esses atributos são características dos solicitantes, dos recursos solicitados e do contexto da solicitação. As regras estabelecem relações entre os atributos que serão avaliadas durante a análise das políticas;

d) **Permissão (*permission*)**: especifica a permissão ou negação de acesso que será resultado da avaliação da política, caso os atributos e condições estabelecidos no campo regras sejam satisfeitos;

e) **Prioridade (*priority*)**: esse campo é utilizado na resolução de conflitos entre políticas, criando uma hierarquia entre elas. Em caso de conflito, o valor de prioridade de cada política é analisado e será aplicada aquela que tiver maior valor ou conforme definido pela organização.

Em relação ao campo regras presente nas políticas, este representa um bloco de relações entre atributos que devem ser verificadas considerando os atributos recebidos nas requisições. Assim, neste campo estão contidos atributos relativos aos sujeitos solicitantes das informações de contexto, às informações solicitadas e ao contexto da solicitação, abrindo espaço para a construção de vários arranjos que reflitam os cenários da IoT e as políticas de segurança.

As políticas de controle de acesso podem ser especificadas pelos proprietários das informações ou pelos administradores da organização, permitindo uma maior adequação às opções de privacidade. Por exemplo, poderia ser especificada uma política de controle de acesso na qual as informações sobre a temperatura e pressão arterial de um paciente só seriam fornecidas a usuários com os atributos de cargo médico e prevenientes de dispositivos na rede da organização.

Levando em conta os objetivos da FCAAS-IoT, recomenda-se que as políticas sejam planejadas do ponto de vista das informações de contexto, de modo a estabelecer um controle mais refinado do acesso a cada informação, impedindo a composição de contexto de forma indevida. Uma determina fonte de contexto, ilustrativamente, pode fornecer informações

de localização, temperatura, estado e tempo de atividade, as quais podem ser agrupadas em vários tipos de contextos. Assim, poderia ser delineada uma política para cada uma dessas informações garantindo que cada uma só fosse disponibilizada preservando o nível de privacidade pretendido, fazendo com que um determinado contexto só seja fornecido caso as políticas para cada informação que o compõem sejam respeitadas.

4.3.2 Políticas de gerenciamento de criptografia

As políticas de gerenciamento de criptografia atuam no escopo da FCAAS-IoT para definir qual algoritmo de criptografia deve ser usado com base nas informações de contexto de solicitação coletadas. Essas informações podem ser endereço IP, protocolo, identificação do dispositivo, localização destes, horário da solicitação, dentre outras. Adicionalmente, essas políticas visam fornecer as chaves criptográficas ou o local de obtenção destas.

Código-fonte 2 – Exemplo de estrutura de política de gerenciamento de criptografia.

```
{
    "id": <string>,
    "targets": {<string>, <string>, <string>},
    "context": {
        "info_type": {<group_of_attributes>},
        "info_type": {<group_of_attributes>},
        "info_type": {<group_of_attributes>},
    },
    "c_infor": {
        "alg": <string>,
        "key": <string>
    },
    "priority": <string>
}
```

A utilização do contexto da solicitação para decisões de criptografia permite que múltiplos algoritmos sejam implementados no sistema e utilizados conforme o caso. Essa estratégia possui três motivadores: (i) os diferentes cenários de conexão da IoT ofertam diferentes

níveis e padrões de segurança; (ii) as informações de contexto podem ser solicitadas por diversos dispositivos, o que inviabiliza o armazenamento de uma chave para cada requisitante e até mesmo ter conhecimento de cada um, dada a amplitude da IoT, e (iii) a heterogeneidade dos dispositivos que compõem a IoT, os quais variam em termos de capacidade computacional e de armazenamento e são capazes de implementar diferentes tipos de mecanismos de encriptação.

O Código-Fonte 2 ilustra a estrutura conceitual de uma política de gerenciamento de criptografia estruturada com base em JSON. Os campos que compõem a política são:

a) **Identificador (*id*):** campo utilizado para identificar a política de forma única para fins de organização e administração;
b) **Alvos (*target*):** definem quando aquela política deve ser analisada, sendo usado na recuperação das políticas. Por exemplo, esse valor pode estabelecer os protocolos usados para as requisições como critério para definição da análise. Em nível conceitual, qualquer informação que componha o contexto da solicitação pode ser usado nesse campo;
c) **Contexto (*context*):** representa o contexto da solicitação, definindo grupos de informações de cada tipo que servirão como base para a análise da política e determinação da sua aplicabilidade ou não. Por exemplo, no campo IP podem ser elencados vários endereços de rede para os quais aquela política se aplica;
d) **Informações de criptografia (*c_infor*):** informa o algoritmo de criptografia e a chave ou local de obtenção desta que serão usados na encriptação dos dados. Caso o contexto de solicitação fornecido seja compatível com o estabelecido na política, os valores desses campos são retornados;
e) **Prioridade (*priority*):** valor numérico utilizado para definir a ordem de análise das políticas, com o intuito de garantir que o contexto de solicitação mais restrito seja aplicado em detrimento dos mais genéricos. Por exemplo, poderiam ser analisadas primeiro as políticas que se aplicam a um contexto específico, como um determinado endereço de rede, e, caso nenhuma se aplique, passar para contextos que possuem características mais gerais, chegando, em última instância, até uma política padrão.

Ilustrativamente, poderia ser especificada uma política que determinasse que, para requisições provenientes de dispositivos externos à rede da organização e de um protocolo específico como MQTT, fosse utilizada criptografia AES com uma chave especificada ou um

esquema de criptografia leve. Outro exemplo de política: para um determinado conjunto de dispositivos internos à rede da organização, utilizar criptografia RSA com chave obtida em um servidor especificado.

4.4 Execução das funções de segurança

A operação do FCAAS-IoT, desde o recebimento da solicitação de contexto até a entrega das informações ao consumidor é dividida em três etapas: (i) Autorização, (ii) Obtenção de contexto e (iii) Fornecimento de contexto. Com o intuito de focar no provimento das funções de segurança realizadas no processamento das requisições e disponibilização de contexto, assumimos que:

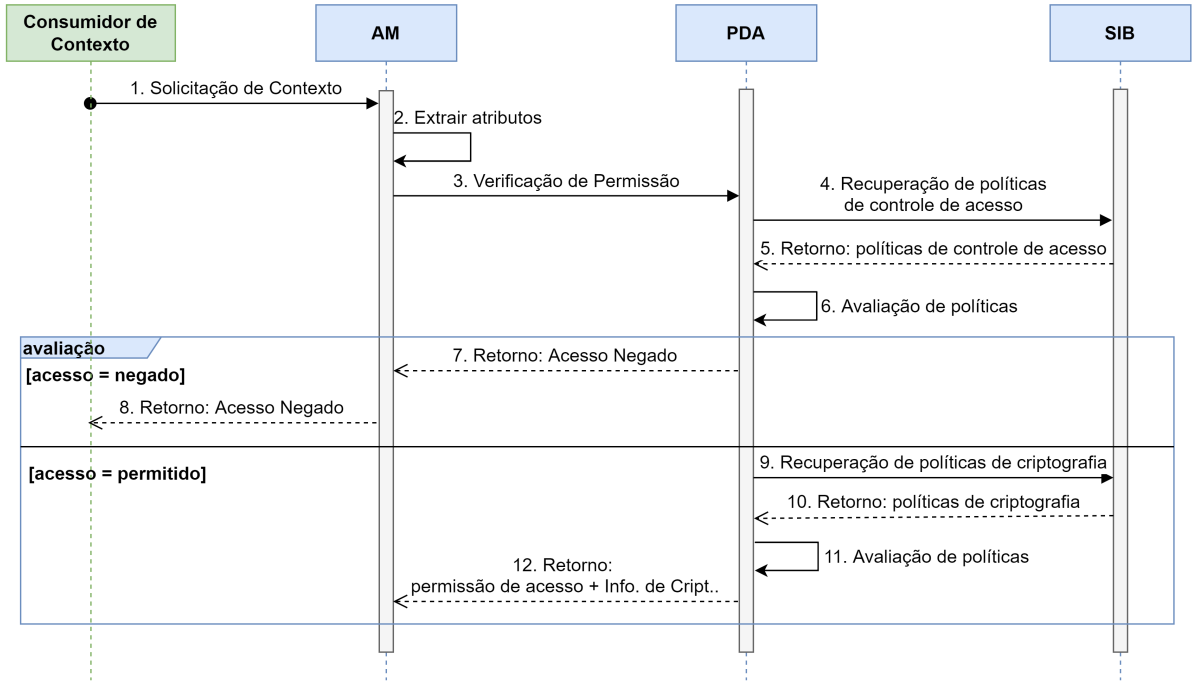
a) Consumidores e fontes de contexto têm confiança nos módulos da FCAAS-IoT, ou seja, comunicam-se de forma segura;
b) A comunicação entre os módulos é confiável e segura, garantindo a entrega das mensagens;
c) A infraestrutura escolhida para implantação dos módulos provê as condições necessárias para sua execução de forma isolada, sem interferência mútua;
d) As políticas de segurança armazenadas na SIB são confiáveis;
e) As informações de contexto já encontram-se aptas a serem compartilhadas, ou seja, não tratamos questões de obtenção, modelagem e raciocínio próprias do ciclo de vida de contexto.

Nas subseções a seguir, é feito o detalhamento de cada uma das etapas de funcionamento da FCAAS-IoT, explicando o papel dos módulos, como estes se comunicam e atuam para o provimento das funções de segurança. Também são discutidas questões relacionadas à implementação.

4.4.1 Fase I - Autorização

A fase de autorização tem o objetivo de verificar se um consumidor de contexto tem permissão de acesso às informações requeridas de uma determinada entidade, levando em consideração os atributos fornecidos pelo requisitante e os extraídos da solicitação em si. A Figura 9 esquematiza os procedimentos envolvidos nessa fase, por meio de um diagrama de sequência construído com base em *Unified Modeling Language* (UML).

Figura 9 – Fase de autorização.



Fonte: elaborado pelo autor (2021).

A obtenção de informações contextuais por parte de um consumidor inicia com o envio de uma requisição ao *Authentication Manager* (AM) (mensagem 1), o que pode ser feito utilizando protocolos comuns à IoT, como CoAP e MQTT, e a ambientes tradicionais de rede, como HTTP, por exemplo. Logo, o AM deve ser provido de interfaces de comunicação que permitam o recebimento e processamento de mensagens de acordo com os protocolos utilizados no cenário. Essa requisição precisa conter as especificações do contexto desejado, bem como de sua fonte geradora, além de alguns atributos que puderem ser fornecidos pelo solicitante, como identificação do usuário e do dispositivo, por exemplo, os quais serão usadas no processo de autorização.

Ao receber a requisição, o AM extrai os atributos enviados pelo requisitante e as especificações das informações requeridas (mensagem 2). Além disso, o AM obtém atributos a respeito da própria solicitação e que façam sentido para o esquema de segurança delineado pelas políticas, como endereço IP, horário e protocolo, formando o *contexto de solicitação*. Esses conjuntos de informações são organizados em uma mensagem de verificação de permissão de acesso enviada ao *Policy Decision Agent* (PDA) (mensagem 3). Os dados que comporão o contexto de solicitação, assim como os atributos que serão usados na verificação de controle de acesso e nas decisões de criptografia, devem ser definidos de acordo com as especificações do ambiente e de modo a permitir o grau de segurança desejado, refletindo na definição das

políticas.

Com base nos atributos recebidos do AM, o PDA realiza primeiro a verificação de permissão de acesso. Como um contexto pode ser composto por várias informações e as políticas de controle de acesso são especificadas para cada informação de um dada fonte, as consultas às políticas também são feitas individualmente. Assim, para cada informação solicitada é gerada uma busca na *Security Information Base* (SIB) (mensagem 4) e são retornadas todas as políticas referentes aquela informação (mensagem 5). Após a recuperação de todas as políticas, segue-se a avaliação destas (mensagem 6), que consiste basicamente em verificar os atributos fornecidos em comparação às especificações nas políticas, tendo como retorno uma permissão ou negação de acesso.

Nessa lógica, caso o acesso a uma informação que compõe o contexto requisitado seja negado, toda a requisição será indeferida, mesmo que as demais tenham sido aprovadas. Por exemplo, se o contexto solicitado é formado por dados de temperatura, localização e estado de funcionamento de uma entidade, mas os atributos fornecidos na solicitação não atendam ao especificado na política para a informação de localização, as demais não serão fornecidas. Essa estratégia visa impedir acessos parciais e indevidos ao contexto, que podem resultar na dedução de outras informações a partir das que forem fornecidas indevidamente. Busca-se, assim, dar maior controle no compartilhamento das informações e garantir o cumprimento das opções de privacidade definidas.

Caso o resultado da avaliação de controle de acesso seja negativo (acesso negado), o PDA notifica ao AM (mensagem 7), o qual enviará mensagem ao consumidor de contexto informando da negação de acesso (mensagem 8). Entretanto, caso o resultado da avaliação seja a permissão de acesso, o PDA procederá com a etapa seguinte da autorização que é obter as informações que serão usadas no processo de encriptação. Para isso, esse módulo recupera as políticas de gerenciamento de criptografia armazenadas na SIB (mensagem 9), usando como parâmetro o conteúdo do *contexto de solicitação* fornecido pelo AM no início do processo. Após recuperadas (mensagem 10), as políticas são avaliadas (etapa 11) e o resultado retornado deve ser, no mínimo, a especificação do algoritmo de criptografia e da chave criptográfica ou o local de obtenção desta, que serão enviadas ao AM (mensagem 12) para proceder com a etapa seguinte de obtenção de contexto.

Essa estratégia procura inserir ciência de contexto no processo de escolha de algoritmos e outros recursos que serão usados para encriptar o contexto antes de seu compartilhamento.

Como o contexto de solicitação utilizado nesse processo é obtido pelo AM, que é um componente da arquitetura, pode-se conferir uma confiabilidade a respeito dos dados que o compõem. Assim, é possível definir estratégias de confidencialidade dos dados com base nessas informações, adaptando-se a questões, como redes e dispositivos dos quais se originaram as solicitações, capacidade computacional dos dispositivos e nível de criticidade das informações solicitadas.

Diante do exposto, a fase de autorização da FCAAS-IoT tem como atividades a solicitação de contexto, extração de atributos e composição de contexto de solicitação, verificação de permissão de acesso e obtenção das informações de criptografia. Com o acesso autorizado e recursos de criptografia obtidos, o fluxo das funções de segurança segue com a fase de obtenção de contexto, abordada na seção seguinte.

4.4.2 Fase II - Obtenção de contexto

De forma resumida, essa fase tem o intuito de obter as informação de contexto solicitadas e entregá-las ao módulo responsável pela encriptação. Nessa etapa, dois módulos desempenham papéis fundamentais: o *Context Manager* (CM) e o *Context Broker* (CB).

O CB tem a função de armazenar as informações de contexto recebidas das fontes e organizá-las em tópicos. Esse esquema utiliza o conceito de comunicação indireta via publicação/assinatura (*publish/subscribe*), já bastante difundido na IoT, sobretudo com o protocolo MQTT, por permitir que várias entidades troquem informações sem a necessidade de comunicação direta e que vários destinatários tenham acesso a uma mesma informação ou conjunto de informações de forma simples.

Nesse esquema, as informações de contexto são enviadas para os tópicos estruturados como *"nome_da_fonte/informação"*. Assim, cada informação é vista como um objeto, podendo ter suas próprias políticas de controle de acesso para fins de privacidade e serem recuperadas individualmente. Essa organização objetiva facilitar o acesso parcial a conjuntos de informações conforme os vários arranjos de contexto solicitados pelas entidades. Por exemplo, caso o contexto desejado seja composto apenas pela localização e nível de bateria, o requisitante não teria acesso a todos os dados armazenados sobre determinada fonte, sendo fornecidas apenas as informações especificadas e conforme permissão de controle de acesso obtida na etapa anterior. Além de implementar acesso controlado, essa abordagem pode facilitar a obtenção de contextos distintos para cada solicitação.

É importante frisar também que essa estratégia de centralizar as informações de

contexto no *Context Broker* pode viabilizar a adoção de medidas adicionais de segurança e controle sobre esse componente, o que incrementa a proteção das próprias informações, como protegê-lo com artefatos de segurança a nível operacional de rede já bastante difundidos, como *firewalls* e *Intrusion Detection Systems* (IDSs). Além disso, é possível uma flexibilidade em relação à distribuição, como, por exemplo, existir uma instância desse componente em cada rede, agregando as informações de contexto das fontes ali conectadas.

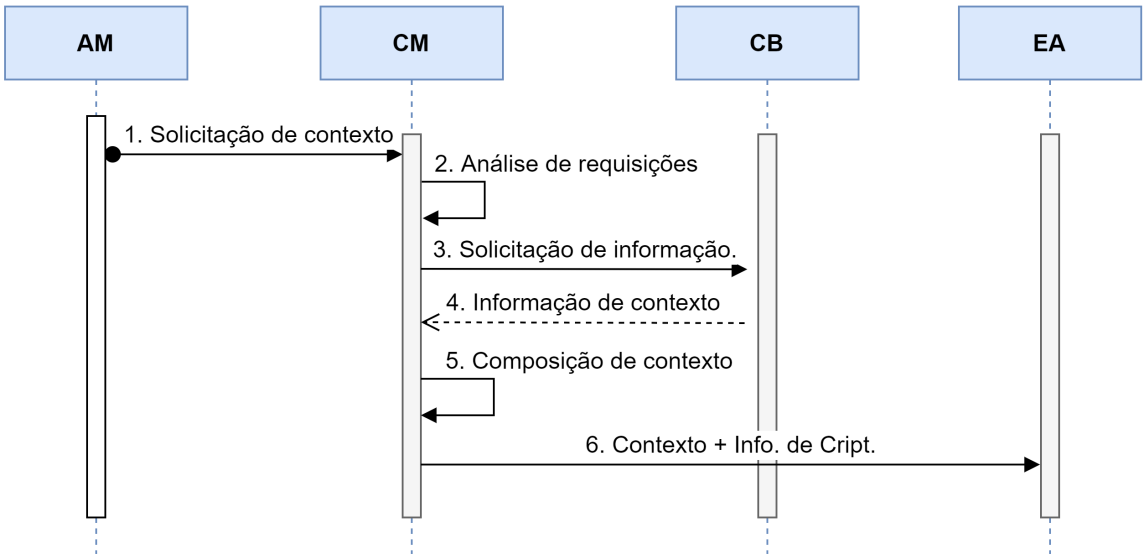
Atuando em conjunto nessa etapa, o CM gerencia a obtenção dessas informações armazenadas no CB, analisando a requisição emitida pelo AM e agrupando-as de modo a formar o contexto que será compartilhado na etapa seguinte. Assim, o comportamento do CM é análogo a um *subscriber*, acessando os tópicos do CB de acordo com as necessidades de composição de contexto. Somente ao final das requisições é que o contexto solicitado estará completo e apenas esse módulo e o *Encryption Agent* terão acesso a ele antes da encriptação, o que permite traçar estratégias para uma maior proteção desses dois componentes durante a implantação da arquitetura.

A Figura 10 apresenta um diagrama de sequência UML que representa as interações entre os módulos na fase de obtenção de contexto. Para fins de simplificação e de dar maior foco na execução das funções da arquitetura, foi considerado que as informações de contexto já encontram-se armazenadas no CB e prontas para serem disponibilizadas. Esse processo de envio das informações das fontes de contexto para o CB pode ser feito utilizando protocolos IoT, como MQTT e CoAP, por exemplo. Entretanto, considerando que o CB armazena as informações via tópicos e que a maioria dos dispositivos fonte possuem recursos restritos, demonstra-se bem viável a utilização do protocolo MQTT, no qual as fontes atuariam como publicadores (*publishers*).

O processo inicia com o AM enviando para o CM uma requisição de contexto (mensagem 1), contendo a especificação das informações requeridas e suas respectivas fontes e os dados sobre os procedimentos de encriptação obtidos na etapa anterior. Ao receber a mensagem, o CM faz a análise e separa os campos referentes às fontes e informações solicitadas (mensagem 2), a fim de gerar requisições de cada informação individualmente ao CB (mensagens 3 e 4), acessando os tópicos conforme especificado. Assim, o CM não precisa conhecer todos os tópicos do *Context Broker*, fazendo com que a consulta seja feita de forma dinâmica e sob demanda.

Após obter todas as informações, o CM faz a composição de contexto (etapa 5),

Figura 10 – Fase de obtenção de contexto.



Fonte: elaborado pelo autor.

associando os valores obtidos a cada informação especificada. Do ponto de vista de implementação, o contexto poderia ser gerado em formato JSON ou YAML, os quais possuem uma boa flexibilidade na representação de dados e facilitam a recuperação destes por parte das aplicações, por se basearem em um modelo chave-valor. Essa fase termina com o envio de uma mensagem do CM para o EA contendo o contexto formatado e as informações de criptografia obtidas na fase autorização.

4.4.3 Fase III - Fornecimento de contexto

A etapa de fornecimento de contexto tem como objetivo criptografar as informações de contexto obtidas pelo CB antes de serem compartilhadas com os consumidores pelo AM.

Na IoT, conforme exposto anteriormente, a criptografia é um desafio para implementação de segurança, dada a quantidade e heterogeneidade dos dispositivos e as restrições de computação e energia de muitos destes. Adicionalmente, cenários IoT podem ser compostos tanto por esses dispositivos restritos, como sensores, quanto por dispositivos mais robustos, como *smartphones* ou até mesmo servidores na nuvem. Logo, entidades que utilizam serviços de aplicações sensíveis ao contexto podem ter recursos que viabilizem um ou outro esquema de criptografia, o que implica em decisões de segurança e interfere em aspectos de implementação.

Levando em consideração tais elementos, é proposta a utilização de princípios de segurança ciente de contexto na FCAAS-IoT, a fim de possibilitar uma abordagem mais flexível e

dinâmica para as decisões de criptografia. Outro potencial ganho dessa abordagem é conferir um nível de segurança adicional, visto que o esquema de criptografia não seria baseado em apenas um algoritmo e chave, fazendo com que a quebra do padrão adotado em um dado compartilhamento não comprometesse a segurança do sistema como um todo, visto que nem todos os dados seriam criptografados segundo um mesmo parâmetro.

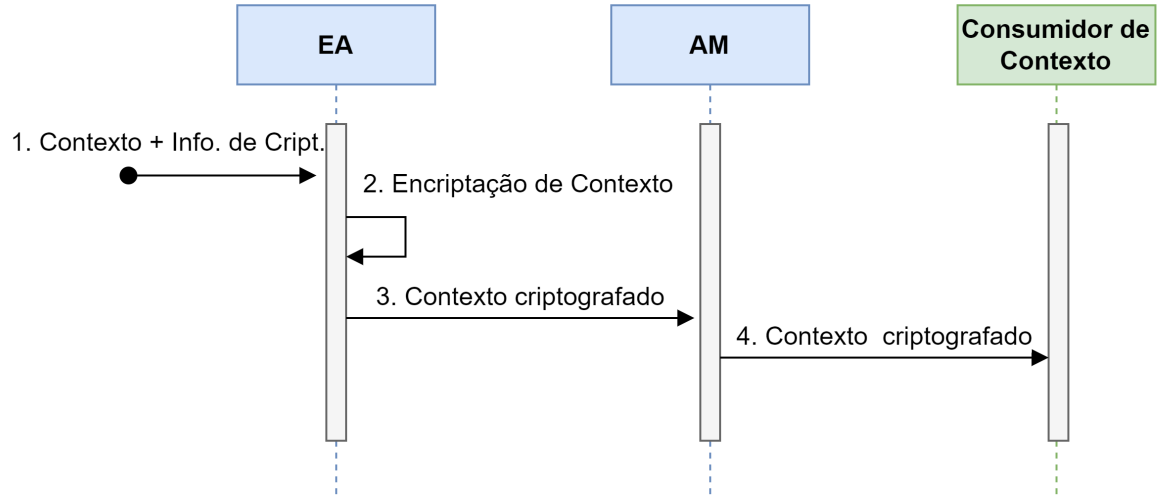
As políticas de gerenciamento de criptografia armazenadas na SIB determinam quais diretrizes serão seguidas para criptografar as informações mediante um contexto de solicitação. Esse contexto é formado por informações que podem variar de uma requisição para outra e é extraído pelo AM na fase de autorização, podendo ser composto por informações como horário, protocolo da solicitação, IP do solicitante e identificação do dispositivo, dependendo das especificidades do cenário de aplicação e da segurança desejada. Desse modo, é possível aplicar diferentes tipos de criptografia para cada caso conforme as políticas especificadas pela organização.

Como reflexo disso, é possível citar a diminuição de chaves a serem armazenadas e a possibilidade de adaptação do algoritmo de criptografia usado sem alterar os demais componentes do sistema. Adicionalmente, ressalta-se a potencial facilidade em implementar novos esquemas de criptografia, em caso de mudanças de chaves ou até mesmo violações, bastando modificar, excluir ou inserir uma política.

Nesse arranjo, seria possível, por exemplo, definir um esquema de criptografia de chave pública clássico e robusto quando a solicitação fosse proveniente de uma entidade que possui maior capacidade computacional e da qual se conhece a chave, como um servidor da organização que acesse as informações habitualmente, e criptografia ABE (*Attribute-Based Encryption*) para outros dispositivos que acessem os dados de contexto protegidos com menor frequência. Ou ainda, utilizar um determinado tipo de criptografia para requisições feitas em um contexto específico, como horário, localização ou rede de origem.

A Figura 11 apresenta os processos envolvidos na etapa de fornecimento de contexto, modelados por meio de um diagrama UML. Essa fase tem início com o envio de uma mensagem do CM para o EA (mensagem 1), contendo o contexto obtido e as informações de criptografia (algoritmo e chave). Ao receber essa mensagem, o EA verifica qual algoritmo deverá ser adotado nesse compartilhamento. Caso a chave seja disponibilizada na própria mensagem, o EA executa a encriptação dos dados, do contrário, pode buscar a chave no local designado no campo *"key"* da mensagem, como um servidor de banco de dados em nuvem ou um arquivo armazenado

Figura 11 – Fase de fornecimento de contexto.



Fonte: elaborado pelo autor.

no próprio EA, por exemplo. Cabe ressaltar que é possível, ainda, que o EA utilize técnicas que implementem integridade, como o uso de *hash* para geração de códigos de autenticação de mensagens, por exemplo. Finalizado esse processo, o EA envia o contexto criptografado para o AM (mensagem 3), que, por sua vez, envia para o consumidor de contexto (mensagem 4).

Em termos de implementação, o EA deverá ser capaz de executar vários algoritmos conforme delimitado pela organização nas políticas de gerenciamento de criptografia. O armazenamento de chaves deve ser feito no próprio EA, dentro da política de gerenciamento de criptografia ou em uma base de dados ou servidor específico de acordo com as especificidades e necessidades de cada arranjo.

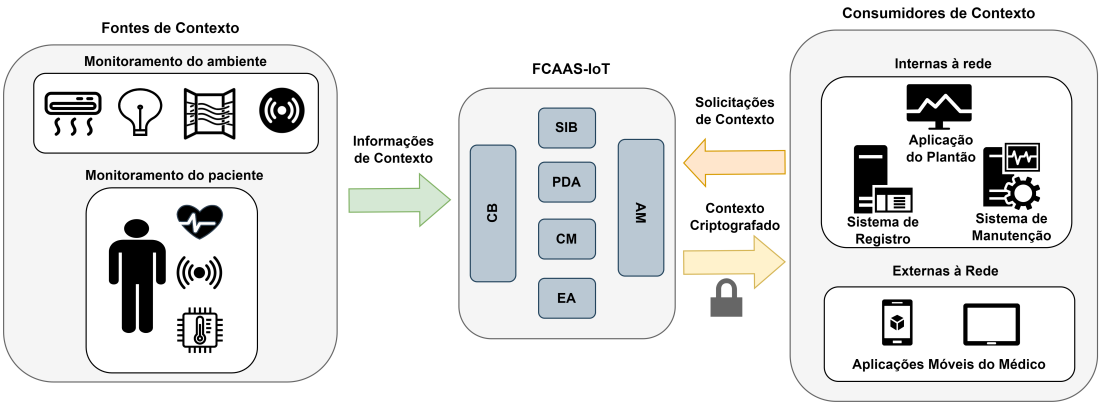
Portanto, as tarefas executadas nessa fase são complexas e podem exigir uma grande quantidade de recursos computacionais. Assim, as técnicas de NFV, *cloud, edge* e *fog computing* podem ser utilizadas para permitir a alocação de recursos computacionais suficientes e prover a infraestrutura necessária para que as tarefas de encriptação sejam realizadas de acordo com o necessário, por meio de balanceamento de carga, múltiplas instâncias do agente de encriptação, dentre outras questões.

4.5 Cenário de Aplicação

A fim de auxiliar no entendimento da atuação da FCAAS-IoT e discutir aspectos de implementação, foi delineado um cenário de aplicação baseado em *healthcare*, o qual é um dos principais domínios da IoT e tem um grande potencial para desenvolvimento de aplicações cientes

de contexto, muitas das quais utilizando dados críticos, o que demanda maiores preocupações com segurança. A Figura 12 sintetiza o cenário modelo.

Figura 12 – Cenário de implementação da FCAAS-IoT.



Fonte: elaborado pelo autor.

No cenário, um paciente internado em um hospital é monitorado por diversos equipamentos e sensores que captam informações a respeito de sua saúde, como pressão arterial, temperatura, oxigenação, batimentos cardíacos e glicemia. Além disso, são vinculadas ao paciente suas informações pessoais e dados relativos a sua rotina de internação, como medicações recebidas, refeições, dados sobre o sono e movimentação. Nessas últimas, dados referentes a alimentação, medicação e horários de banho, por exemplo, podem ser alimentados pela equipe de atendimento, já outros, como sono e movimentação, podem também ser coletados por dispositivos sensores.

Além dos dados relativos ao paciente, o quarto possui equipamentos inteligentes, como lâmpadas, cama articulada, ar condicionado, portas, janelas e sensores que captam condições do ambiente, como temperatura, umidade e luminosidade. Esses dispositivos podem fornecer tanto informações sobre o ambiente do quarto quanto a respeito do seu estado de funcionamento, as quais são utilizadas para controlá-los de forma automatizada via aplicações ou pela equipe do hospital.

No âmbito da FCAAS-IoT, os artefatos atrelados ao paciente e ao ambiente são *fontes de contexto*, fornecendo dados que podem ser agrupados de várias formas para compor contextos a serem utilizados por diversas aplicações com finalidades específicas. Desse modo, o dado coletado por uma entidade pode integrar contextos para diferentes aplicações. Assim, é desejável controlar o uso dessas informações, a fim de evitar acessos e usos indevidos.

Por outro lado, aplicações que usam os dados fornecidos pelas fontes de contexto

em seu funcionamento para prover algum tipo de serviço, realizar alguma ação ou para fins de registro são consideradas *consumidores de contexto*. Logo, cada aplicação utiliza um conjunto de informações que compõem o seu contexto, as quais podem ser solicitadas e fornecidas periodicamente ou sob demanda, dependendo da necessidade e da lógica de operação.

Nesse cenário, são consideradas as seguintes aplicações:

a) **Sistema de registro dos pacientes**: aplicação executando em um servidor na rede interna do hospital, que armazena as informações a respeito de todos os pacientes, tanto de dados de saúde, quanto de histórico e dados pessoais;
b) **Aplicação dos médicos**: aplicação executando em um dispositivo móvel do médico (tablet ou smartphone, por exemplo), que tem acesso a todas as informações dos pacientes associados àquele profissional. Essa aplicação pode solicitar as informações periodicamente e, com base em condições pré-estabelecidas, executa algumas ações, como gerar alertas ou exibições personalizadas conforme necessário;
c) **Aplicação da equipe de plantão**: tem a função de exibir as informações de todos os pacientes internados sob responsabilidade dos profissionais da equipe de plantão em exercício e que estejam vinculados às necessidades dos pacientes;
d) **Equipe de manutenção**: aplicação disponível à equipe de manutenção do hospital, que monitora o funcionamento dos equipamentos, a fim de detectar defeitos e mau uso.

Considerando os objetivos e funcionamento dessas aplicações, cada um necessita de um conjunto específico de informações de contexto dentre as disponíveis em todo o ecossistema do hospital. Assim, uma mesma informação pode integrar o contexto de várias aplicações consumidoras, bem como o contexto de uma aplicação pode ser composto por informações provenientes de várias fontes. Nessa perspectiva, é necessário um recurso que permita que apenas as informações necessárias à operação de cada aplicação sejam disponibilizadas, havendo, portanto, um controle de acesso atrelado ao consumidor de contexto.

Expandindo a análise do cenário, é necessário considerar especificidades das solicitações. Do ponto de vista de usuários, cada aplicação será utilizada por vários usuários que compõem as diversas equipes médicas e de manutenção do hospital e que só deverão ter acesso às informações de contexto nos horários e dispositivos pré-determinados de acordo com suas escalas de trabalho. Desta feita, o controle de acesso deve ser capaz de dar suporte a essas

demandas e permitir que alterações nos mecanismos de controle sejam feitas de forma a não prejudicar a segurança do sistema como um todo, como, por exemplo, em casos de trocas de escalas, adição de funcionários ou ingresso de novos pacientes.

Especificamente em relação à aplicação dos médicos, existe a particularidade de acesso por meio de vários dispositivos móveis, tanto na rede interna do hospital como em redes externas cujos mecanismos de proteção nem sempre serão conhecidos ou confiáveis. Portanto, por meio do contexto de solicitação, poderiam ser atribuídos algoritmos e chaves de criptografia diferentes a depender do dispositivo e/ou da rede na qual o médico estivesse emitindo as requisições de contexto. Tal ação potencialmente auxiliaria a manutenção da privacidade e da confidencialidade.

Do mesmo modo, o sistema de registro dos pacientes deve possuir amplo acesso às informações, tendo seu contexto composto por quase que a totalidade destas. Logo, o nível de segurança demandado é maior. Assim, poderiam ser usados esquemas de segurança mais robustos, uma vez que essa aplicação executaria sempre no servidor interno à rede, que possui capacidade computacional elevada, abrindo espaço para algoritmos e técnicas mais dispendiosas computacionalmente.

Assim, é possível perceber que as funções de segurança propostas pela FCAAS-IoT conseguem adequar-se às necessidades de proteção das informações, possibilitando modelar políticas que abranjam os vários tipos de consumidores de contexto e que levem em conta aspectos das solicitações para otimizar a proteção dos dados, independentemente da composição dos contextos solicitados.

Do ponto de vista de implantação dos módulos, nesse cenário, vários arranjos são possíveis, podendo considerar questões como localização, balanceamento de carga e organização das demandas de coletas. Diante de uma quantidade de consumidores, fontes e solicitações relativamente baixa, é possível adotar uma implantação centralizada, com todos os módulos executando em um servidor na rede interna, por exemplo. Uma estrutura centralizada traria potenciais benefícios, como a simplicidade, facilidade de administração e centralização de políticas e informações de contexto. Entretanto, tal arranjo poderia trazer questões como a existência de um único ponto de falhas e possibilidade de sobrecarga do dispositivo de rede, o que geraria atrasos no processamentos das solicitações e entrega dos dados.

Caso o hospital tenha uma área grande, composta por vários prédios, setores e equipes, poderiam ser utilizadas as técnicas de *cloud*, *fog* e *edge computing*, a fim de distribuir melhor

as instâncias dos módulos de acordo com as demandas e com as funções a serem desempenhadas. Cabe frisar, entretanto, que independente das opções de implementação adotadas, é necessário garantir que as funções de segurança consigam ser providas conforme delineado pela execução das tarefas de cada módulo e a comunicação entre eles.

Por exemplo, poderia ser implantada uma instância do CB por setor, a fim de agregar as informações das fontes ali presentes, função essa que poderia ser alocada em um dispositivo na borda da rede. Nessa lógica, o CB atenderia a requisições de informações de contextos feitas pelo CM que executaria, por exemplo, em um servidor em nível de *fog*, responsável por acessar os CBs de todos os setores de um prédio. No mesmo nível, poderiam ser implementados os demais módulos, com múltiplas instâncias, objetivando atender às demandas de setorização, segurança e QoS. Além disso, o nível de *cloud* seria útil para instanciar réplicas de módulos que lidem com armazenamento de políticas ou chaves de criptografia, como a SIB, para fins de backup, por exemplo. Adicionalmente, seria possível também, agrupar a implementação de módulos que trabalham de forma associada, como CB e CM, PDA e SIB, com o objetivo de melhorar o desempenho das funções de segurança levando em conta a estrutura disponível.

Em uma outra perspectiva, o AM poderia ser replicado com o objetivo de aumentar a disponibilidade, evitar sobrecarga desse componente ou até mesmo separar em cada instância as solicitações de acordo com sua origem (rede interna ou externa) ou tipo de protocolo de comunicação, o que facilitaria a programação desse elemento, uma vez que não seria necessário que todas as instâncias possuíssem interfaces de comunicação para todos os protocolos. Outra opção seria prover múltiplas instâncias do EA, de modo que cada uma fosse responsável por um tipo de criptografia, o que daria margem para estratégias mais aprimoradas para efetivação das políticas de encriptação.

Para construir tais implementações com múltiplas instâncias, tecnologias de NFV seriam úteis, as quais auxiliariam na alocação desses elementos através da estrutura da rede e de acordo com as demandas elencadas, bem como, devido à flexibilidade provida por essa tecnologia tanto na facilidade para criação de instâncias quanto na possibilidade de modificações mais rápidas no código dos módulos.

Entretanto, é importante destacar que, apesar da potencial flexibilidade, a escolha das tecnologias para implantação dos elementos da FCAAS-IoT pode impactar em aspectos como complexidade da estrutura e tempo de resposta, devendo haver a mediação entre as necessidades de fornecimento de contexto cada cenários, os recursos disponíveis, o nível de segurança desejado

e o conjunto de tarefas que devem ser realizadas por cada um dos elementos da arquitetura.

4.6 Considerações finais

Neste capítulo, foram fornecidos detalhes da FCAAS-IoT, expondo as estratégias usadas, elementos da arquitetura, funcionamento das funções de segurança e composição das políticas. Além disso, um cenário de implantação da arquitetura foi traçado com o intuito de apresentar possibilidades de implantação e seu caráter adaptativo para várias situações.

Diante do exposto, ficam perceptíveis os benefícios da abordagem modularizada e baseada em funções da arquitetura proposta, com o objetivo de adaptar-se aos vários contextos da IoT, provendo flexibilidade desde o projeto das políticas de segurança até as várias possibilidades de implantação dos módulos, visando garantir principalmente privacidade e confidencialidade no compartilhamento e uso das informações de contexto pelas aplicações.

5 PROVA DE CONCEITO

Este capítulo apresenta a prova de conceito elaborada bem como os experimentos realizados com o propósito de analisar aspectos relevantes da FCAAS-IoT, considerando seus objetivos e o modo como as funções de segurança são desempenhadas.

5.1 Escopo geral

Conforme exposto no capítulo anterior que define a arquitetura, o principal objetivo da FCAAS-IoT é prover segurança e privacidade para aplicações sensíveis ao contexto em ambientes IoT, utilizando, para tanto, funções de segurança baseadas no modelo ABAC de controle de acesso e no uso de informações a respeito das requisições para decisões sobre criptografia. Levando isso em consideração e os princípios de funcionamento de aplicações desse tipo, foram formuladas duas hipóteses para guiar os experimentos:

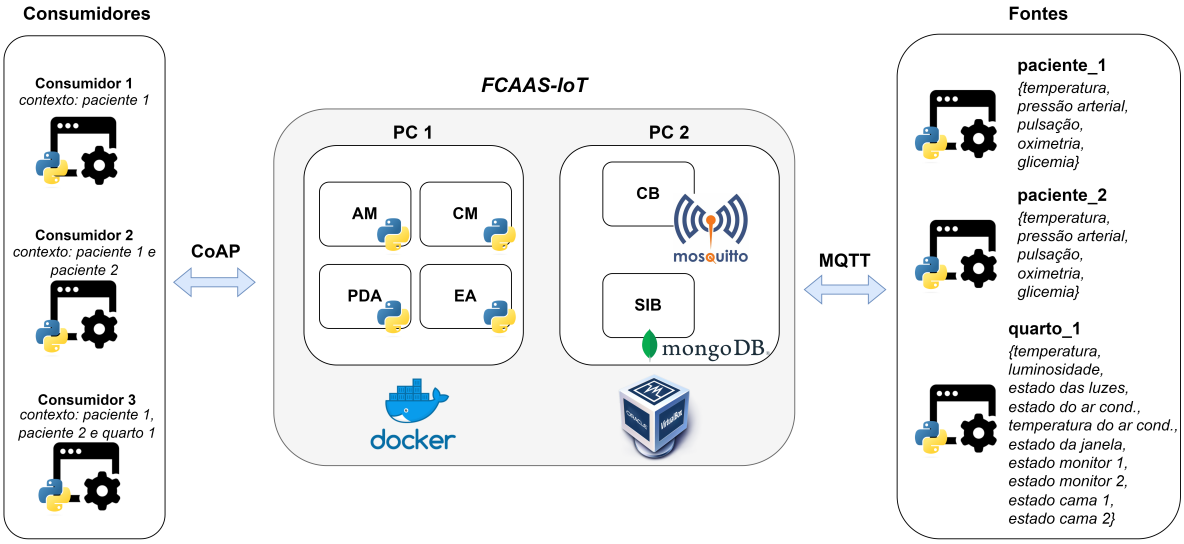
a) FCAAS-IoT consegue prover segurança no fornecimento de contextos com diferentes composições a vários consumidores em um ambiente IoT?
b) As funções de segurança da FCAAS-IoT impactam o tempo de resposta no fornecimento de contexto para as aplicações? Caso sim, existe variação de acordo com a quantidade de informações requeridas?

Desse modo, foi elaborado um cenário que consiste em uma adaptação do cenário exposto na seção 4.5, possibilitando emular o funcionamento da arquitetura para diferentes combinações de requisições, variando fontes, consumidores e composição de contexto. O cenário é apresentado na Figura 13.

Foram projetadas três fontes de contexto: *paciente_1, paciente_2* e *quarto_1*. Cada paciente pode fornecer cinco informações de contexto, a saber, temperatura, pressão arterial, pulsação, oximetria e glicemia. Sobre o quarto, dez informações podem ser elencadas: temperatura, luminosidade, estado das luzes, estado do ar-condicionado, temperatura do ar-condicionado, estado das janelas, estado de dois monitores e estado do sistema de duas camas. Portanto, o cenário modelado possui vinte informações que podem ser combinadas em contextos distintos. Como mecanismo de envio dos dados das fontes para a FCAAS-IoT foi definido o protocolo MQTT, devido ao fato de ser bastante comum para situações desse tipo em ambientes IoT.

O cenário possui ainda três consumidores que requerem diferentes informações das fontes de contexto. O Consumidor 1 possui como contexto as informações referentes ao paciente

Figura 13 – Cenário da Prova de Conceito.



Fonte: elaborado pelo autor.

1, o Consumidor 2, as informações de ambos os pacientes, e o contexto do Consumidor 3 é formado pelas informações dos pacientes e dos dispositivos do quarto. O protocolo CoAP foi escolhido como método de comunicação dos consumidores com a FCAAS-IoT por ser bastante utilizado em ambientes IoT por parte das aplicações.

Para cada informação de contexto foi definida uma política de controle de acesso que sistematiza os atributos necessários à avaliação de permissão de acesso. As políticas são compostas por atributos de sujeito (nome, cargo e equipe), recurso (nome, fonte e estado), ação (método) e contexto (ip, porta, dispositivo, protocolo, localização, horário, data e estado de login), totalizando quinze atributos.

No processo de recuperação das políticas da SIB, foi definido que o atributo utilizado seria a “fonte” dos dados. Assim, todas as políticas referentes àquela fonte são recuperadas, mas apenas aquelas que dizem respeito às informações discriminadas na solicitação são avaliadas, utilizando como parâmetro de separação o atributo “nome”.

Em relação às políticas de encriptação, quatro foram definidas, uma para cada consumidor e uma política padrão para ser aplicada quando nenhuma das demais for adequada. São usados como atributos pelas políticas o endereço IP, identificador do dispositivo, protocolo e porta. Como mecanismos de encriptação, foi definida a utilização de chave de sessão, gerada pelo EA para criptografar as informações de contexto e uso de um esquema de chave pública para criptografar a chave de sessão, a qual é fornecida como resultado da fase de autorização. Assim, as políticas de criptografia permitem a escolha da chave pública específica a ser utilizada para criptografar a chave de sessão, mediante o contexto de solicitação apresentado.

Na seção seguinte, são apresentados os recursos computacionais e de infraestrutura usados para implementar o cenário ora descrito e realizar os experimentos que serão posteriormente detalhados.

5.2 Materiais e recursos

Além do cenário modelado, a Figura 13 também apresenta alguns recursos que foram utilizados para construção do cenário formulado para esta prova de conceito.

Para a construção dos módulos, foram adotadas diferentes estratégias de acordo com as funções desempenhadas por cada um dentro da arquitetura. Os módulos AM, CM, PDA e EA foram programados em *Python* e executados em containers, utilizando para tanto a ferramenta Docker de containerização. Esses módulos foram executados em um computador (PC1) com sistema operacional *Ubuntu 20.04*, processador *Intel Core i5* e 12GB de memória RAM.

Os módulos CB e SIB foram executados em um computador (PC2), utilizando uma máquina virtual *Ubuntu Server 18.04*, com 2GB de RAM e dois núcleos de processamento, virtualizada por meio do hipervisor *Oracle VirtualBox*, versão 7.0[1]. Para implantar o módulo CB, foi utilizado o broker Mosquitto[2], por ser uma ferramenta simples e que cumpria os requisitos funcionais desse componente. Já o módulo SIB foi implantado por meio do banco de dados *noSQL MongoDB*[3], o qual possibilitou armazenar as políticas de controle de acesso e de encriptação em formato JSON. Optou-se por executar esses módulos em outro computador a fim de evitar sobrecargas de processamento que gerassem atrasos na execução das funções e pudessem, eventualmente, interferir no desempenho das funções.

Em relação às políticas, tanto de controle de acesso quanto de criptografia, estas foram definidas em arquivos no formato JSON e armazenadas na SIB em coleções separadas, através do módulo PMA, também programado em Python, conforme o que foi apresentado no capítulo anterior que define as políticas, e realizando as adaptações necessárias tendo em vista as bibliotecas e especifidades da linguagem de programação usada.

Para a implantação das funções relativas ao ABAC, foi utilizada a biblioteca *PyABAC*[4], tendo sido feitas adaptações necessárias ao cenário, principalmente na definição das políticas de controle de acesso, considerando que estas utilizam o padrão XACML. A Figura 14

[1] https://www.virtualbox.org/
[2] https://mosquitto.org/
[3] https://www.mongodb.com/
[4] https://py-abac.readthedocs.io/en/latest/

mostra uma das políticas definidas na prova de conceito, no caso para a informação *temperature* da fonte paciente_1, criada a partir de adaptações do modelo fornecido na documentação da biblioteca e seguindo a sintaxe do padrão XACML.

Figura 14 – Exemplo de política de controle de acesso.

```
{
    "uid": "100",
    "description": "Policy description.",
    "effect": "allow",
    "rules": {
        "subject": {"name": {"condition": "IsIn", "values": ["Eva", "Ciro", "Lina", "Arthur"]},
                    "team": {"condition": "IsIn", "values": ["001", "002", "003"]},
                    "role": {"condition": "IsIn", "values": ["doctor", "nurse", "director"]}},
        "resource": {"name": {"condition": "Equals", "value": "temperature"},
                     "status": {"condition": "Equals", "value": "current"},
                     "source": {"condition": "Equals", "value":  "patient_1"}},
        "action": {"method": {"condition": "Equals", "value": "get"}},
        "context": {"protocol": {"condition":  "IsIn", "values": ["coap", "http", "mqtt", "tcp"]},
                    "ip":  {"condition":  "AnyOf",
                            "values": [{"condition": "CIDR", "value":  "192.168.1.0/24"},
                                       {"condition": "CIDR", "value": "132.20.0.0/16"},
                                       {"condition": "CIDR", "value":  "127.0.0.1/32"}]},
                    "time":  {"condition":  "Any"},
                    "port":  {"condition":  "IsIn", "values": ["5683", "1883", "80"] },
                    "date":  {"condition":  "Any"},
                    "device": {"condition": "Any"},
                    "logged": {"condition": "Equals", "value": "True"},
                    "location":  {"condition":  "Any"}}
    },
    "targets": {
        "subject_id": "*",
        "resource_id": ["patient_1"],
        "action_id": "*"
    },
    "priority": 0
}
```

Fonte: elaborado pelo autor.

A implementação das funções de criptografia foi construída utilizando a biblioteca PyCryptodome[5], tanto para a encriptação e decriptação dos dados quanto para geração das chaves. Conforme o cenário descrito, os módulos RSA e AES foram escolhidos como algoritmos de criptografia de chave pública e simétrica, respectivamente. Como chaves de sessão, foram usados conjuntos de 16 bytes gerados aleatoriamente, e, como base para geração de chaves públicas e privadas para o RSA adotou-se conjuntos de 2048 bits também gerados aleatoriamente e o esquema de criptografia RSAES-OAEP. A Figura 15 apresenta o exemplo de uma dessas políticas, no caso, a que deve ser aplicada quando as requisições forem provenientes do dispositivo "server", correspondente ao Consumidor 3 no cenário desta prova de conceito.

Consumidores e fontes de contexto foram programadas em Python. Os atributos fornecidos nas solicitações foram armazenados em arquivos JSON e recuperados pelos consumidores de acordo com o perfil de cada solicitação. As chaves utilizadas pelos consumidores para descriptografar os dados foram armazenadas na ferramenta MongoDB executada no PC2 em um

[5] https://pycryptodome.readthedocs.io/en/latest/

Figura 15 – Exemplo de política de criptografia.

```
{
    "id": "103",
    "tags": ["coap"],
    "priority": 3,
    "context": {
        "ip": ["192.168.1.0/24"],
        "port": ["5683"],
        "protocol": ["coap"],
        "device": ["server"],
    },
    "c_infors": {
        "alg": "rsa",
        "key": "LS0tLSTTN5 [...]T0ZG1najS0tLS0t",
        }
}
```

Fonte: elaborado pelo autor.

banco de dados separado das políticas e são recuperadas quando um contexto criptografado é recebido considerando o contexto de solicitação adotado. Para a comunicação via CoAP entre os consumidores e FCAAS-IoT foi usada a biblioteca aiocoap[6].

Por fim, as fontes são representadas por programas Python. Dado que a atuação da FCAAS-IoT se dá no momento do compartilhamento da informação de contexto, após passar por todo o ciclo de vida, os valores enviados ao CB foram gerados aleatoriamente dentro de um intervalo definido e em um formato que fizesse sentido para aquele tipo de dado. Assim, os dados são convertidos em *strings* antes de serem enviados pelas fontes. Para comunicar as fontes e a arquitetura, foi utilizada a biblioteca Eclipse Paho MQTT[7], de modo que as fontes atuassem como *publishers*, visto que as funções do CB seriam exercidas pelo *broker* Mosquitto. As informações foram enviadas previamente das fontes para o CB, ou seja, quando uma requisição é feita por um consumidor, o valor fornecido já estava armazenado na arquitetura não sendo solicitado à fonte para atender à requisição.

Nas seções seguintes, são relatados os experimentos realizados e fornecidos mais detalhes que forem pertinentes a cada caso.

5.3 Experimento 1

Nesse experimento, o foco foi investigar a primeira hipótese: se a arquitetura consegue prover segurança no fornecimento de contextos com diferentes composições a vários consumidores. Em outras palavras, buscou-se verificar a efetividade da lógica de implantação

[6] https://aiocoap.readthedocs.io/en/latest/index.html
[7] https://pypi.org/project/paho-mqtt/

das funções de segurança. Levando em conta o funcionamento da arquitetura, efetividade foi definida como a combinação dos elementos:

I. Aplicação correta das políticas de controle de acesso, permitindo ou negando acesso às informações de contexto mediante os atributos fornecidos;
II. Uso correto dos algoritmos e chaves definidos nas políticas de criptografia, levando em conta o contexto de solicitação.

No cenário descrito para esta prova de conceito, todas as políticas de controle de acesso elencam atributos que devem ser cumpridos para que a autorização se efetive. Ou seja, caso os atributos fornecidos pelo consumidor estejam presentes nos que forem listados respectivamente nas políticas que tratam da informação requerida, o acesso é concedido, do contrário, é negado. Assim, para verificar a aplicação correta das políticas, basta submeter requisições que atendam às especificações das políticas e outras que estejam fora deles, observando os resultados obtidos de negação ou permissão. Adicionalmente, é necessário observar se o mecanismo de autorização consegue evitar acessos parciais ao contexto, que consiste em garantir que o acesso ao contexto seja negado caso a política para uma ou mais informações que compõem o contexto solicitado não seja atendida.

Do mesmo modo, as políticas de criptografia elencam informações de contexto de segurança para as quais o algoritmo e chave especificados em cada política devem ser usados para encriptar os dados. Conforme explicitado anteriormente, no cenário foram definidas 4 políticas: uma para cada consumidor de contexto e uma padrão que seria aplicada quando nenhuma das demais for adequada. A informação que difere entre as políticas é a especificação do dispositivo, determinando qual chave pública será usada para criptografar a chave de sessão gerada pelo EA para a requisição. Logo, para averiguar a corretude na aplicação das políticas, era preciso variar o valor desse atributo e verificar se o consumidor consegue ter acesso ao dado descriptografado, considerando que no processo de decriptação está definida a chave adequada a isso. Ou seja, se os dados forem criptografados pela arquitetura usando a chave definida na política correta esperada, o consumidor conseguirá ter acesso a estes.

Diante disso, foi definido que seriam geradas requisições fornecendo diferentes valores de atributos para cada um dos consumidores de contexto. Dentre os atributos de controle de acesso, foi selecionado “name” do grupo “subject” para variar testando as políticas. Já em relação às políticas de encriptação, para cada consumidor, foi feita a utilização do valor correto para o atributo dispositivo, a fim de testar a aplicação da política adequada para aquele

consumidor, e de um valor que não satisfizesse nenhuma das políticas de criptografia, a fim de testar a aplicação da política padrão, sendo feita a configuração adequada nos consumidores para cada caso.

Para o Consumidor 1, que tem como contexto as informações relativas ao paciente 1, foi gerada uma requisição de contexto fornecendo um valor que não atendia à política para a informação oximetria. Como resultado, todo o acesso ao contexto foi negado e não houve a continuidade da verificação, mesmo que as informações anteriores já tivessem sido autorizadas. Em seguida, foi feita uma requisição com atributos adequados à permissão de acesso e com a especificação adequada do dispositivo. Verificou-se que o acesso foi autorizado e que a política de criptografia correta foi aplicada, considerando que o consumidor conseguiu descriptografar os dados corretamente.

Finalizando essa etapa, alterou-se o valor do atributo *"device"* fornecido para um que não atendesse às políticas de encriptação contidas na SIB, objetivando forçar o módulo PDA a aplicar a política padrão. Foram mantidas as informações de criptografia configuradas no consumidor de contexto que eram, portanto, incompatíveis com a chave que deveria ser retornada pela política e usada para encriptar os dados caso o funcionamento ocorresse conforme esperado.

O comportamento obtido foi que o consumidor recebeu informações de contexto criptografadas, mas, o processo de decriptação não obteve sucesso, sinalizando que a chave privada utilizada não era adequada. Para comprovar a aplicação da política padrão, essa etapa foi repetida, fazendo a troca da chave privada no emissor pela adequada à política de encriptação padrão. Isso possibilitou a decriptação dos dados e consequente leitura e exibição pelo consumidor, mostrando a correta aplicação da política padrão.

O Consumidor 2 tem como contexto as informações relativas aos dois pacientes, ou seja, duas fontes diferentes. Assim, primeiro foi elaborada uma requisição com atributos que atendiam corretamente às políticas para as informações do paciente_1, mas que continha um valor de atributo de nome do sujeito inadequado para a informação de oximetria do paciente_2. O resultado obtido foi a negação do acesso. Prosseguindo, a requisição foi configurada para atender às políticas de todas as informações e a designação correta do dispositivo foi feita, resultando na permissão de acesso e no acesso aos dados pelo consumidor após a decriptação. O procedimento para verificação da política de encriptação padrão foi repetido, o que gerou resultado igual ao observado para o consumidor 1.

O último consumidor, Consumidor 3, possui contexto formado pelas informações de

ambos os pacientes e do quarto. Nesse caso, a última informação do contexto foi selecionada como elemento de teste (estado da cama 2). Assim, de modo similar aos consumidores anteriores, foi gerada uma requisição com valor inadequado de nome de sujeito para a política da informação e, em seguida, uma requisição que satisfizesse as políticas de todos os atributos. Como resultado, no primeiro caso, o acesso foi negado, e, no segundo, foi permitido com a correta leitura dos dados após decriptação. O procedimento para verificação da política de criptografia padrão também foi repetido para esse consumidor, tendo igual resultado.

5.3.1 Discussão dos resultados

Tendo em vista o relatado na execução do Experimento 1, os resultados podem ser resumidos em:

a) FCAAS-IoT conseguiu aplicar corretamente as políticas de controle de acesso mediante os atributos fornecidos, negando ou permitindo;
b) Acessos parciais ao contexto são evitados, já que é necessário apenas que o acesso a uma informação que compõe o contexto seja negada para que toda a requisição seja indeferida, mesmo que as demais tenham sido autorizadas;
c) As políticas de criptografia conseguem estipular tratamentos diferenciados para os dados na encriptação considerando o contexto de solicitação;
d) O mecanismos de utilização de uma política de encriptação padrão para casos em que as demais não se apliquem funciona.

Diante disso, pode-se concluir que as funções de segurança definidas na FCAAS-IoT conseguem proteger as informações de contexto para diferentes arranjos de composição e de requisitantes. Parte disso se deve à alta granularidade fornecida pelo modelo ABAC usado como base para especificação das políticas de controle de acesso, que permite diferentes listagens e combinações de atributos, resultando em um alto potencial de adequação à heterogeneidade e diversidade de cenários da IoT.

Sobre os recursos de criptografia, os resultados apontam para a viabilidade da utilização de informações de contexto para distinção de mecanismos de encriptação adequados a diferentes contextos, o que pode ser usado para aumentar a confiabilidade e privacidade das informações de contexto fornecidas, enquanto, paralelamente auxilia em questões como redução da quantidade de chaves e adequação de criptografia para dispositivos restritos. Assim, esse recurso da FCAAS-IoT consegue aplicar segurança ciente de contexto para uma finalidade

específica, demonstrando que esse recurso pode ser explorado em cenários IoT.

5.4 Experimento 2

Para verificar o impacto dos procedimentos envolvidos nas funções de segurança propostas pela FCAAS-IoT sobre o tempo de resposta, foi realizado um experimento comparando o resultado de diferentes composições de requisições em um cenário com e sem a arquitetura.

Os módulos da FCAAS-IoT foram projetados para atuar entre consumidores e fontes de contexto. Assim, para emular um cenário sem a arquitetura foi utilizado um arranjo muito comum na IoT, que consiste em um *gateway* responsável por receber requisições, obter as informações e retorná-las aos solicitantes. Assim, o cenário normal construído foi composto por dois elementos: o *gateway* e um broker MQTT. O *gateway* foi programado em *Python* e executado em um container *Docker* no PC1, e sua função era receber requisições via CoAP dos consumidores e obter as informações especificadas nas mensagens via solicitações MQTT ao broker. Para o broker, foi utilizado o mesmo componente que emulou o CB na arquitetura, um *broker* Mosquitto, executando no PC2, dado que o comportamento era similar.

A Tabela 3 traz informações a respeito das configurações deste experimento. Dentro do objetivo traçado, foi selecionado como métrica o tempo de resposta, que aqui é entendido como o tempo decorrido entre o instante em que o consumidor emite a solicitação e o momento em que ele consegue ler os dados, ou seja, que estes estariam disponíveis para uso, o que ocorre após a decriptação dos dados no cenário com a arquitetura. Logo, para cada requisição foi coletado o valor do tempo de resposta em milissegundos (ms).

Levando em conta que os procedimentos de controle de acesso são executados para cada informação que compõe o contexto solicitado, esperava-se que a quantidade de informações requeridas impactasse no tempo de resposta. Nesse sentido, conforme especificado na Figura 13, foi feita a variação na composição do contexto solicitado, gerando requisições com 5, 10 e 20 informações em ambos cenários.

Na execução dos experimentos, as requisições foram geradas em blocos de 100 requisições, salvando-se o tempo de resposta de cada requisição. Para introduzir aleatoriedade, cada requisição era gerada em um intervalo entre 2 e 4 segundos, funcionando também como uma forma de evitar que o processamento da requisição anterior afetasse de algum modo. Ao final do bloco de requisições, a média e o desvio padrão dos tempos de resposta foram calculados e armazenados. Esse procedimento foi repetido 50 vezes para cada variação na quantidade de

Tabela 3 – Configurações do Experimento 2.

Informação	
Cenários avaliados	Sem e com FCAAS-IoT
Quantidade de cenários	6
Métrica	Tempo de resposta (ms)
Fator	Campos de informação solicitados por requisição
Níveis	5, 10 e 20 informações
Quantidade de requisições por bloco	100
Intervalo entre requisições	Aleatório entre 2 e 4 segundos
Intervalo entre blocos	5 segundos
Quantidade de repetições	50

atributos com e sem a arquitetura, gerando um arquivo com 50 registros de média e desvio padrão para cada experimento. Assim, foram coletadas 5000 amostras para cada experimento. Essa quantidade de requisições visa reduzir a interferência de valores discrepantes sobre o resultado final, uma vez que podem ocorrer problemas relacionados ao funcionamento dos protocolos e componentes do cenário que venham a influir sobre o tempo de resposta.

A execução dos experimentos foi coordenada por meio de um *script* Python, cujo funcionamento pode ser resumido como segue:

I. Iniciar os *containers* que emulam os módulos da FCAAS-IoT ou o *gateway*, conforme o caso;

II. Após 3 segundos, iniciar o bloco de 100 requisições, salvando a média e o desvio padrão desse conjunto de coletas;

III. Destruir os *containers* e, após 5 segundos, reiniciar o processo.

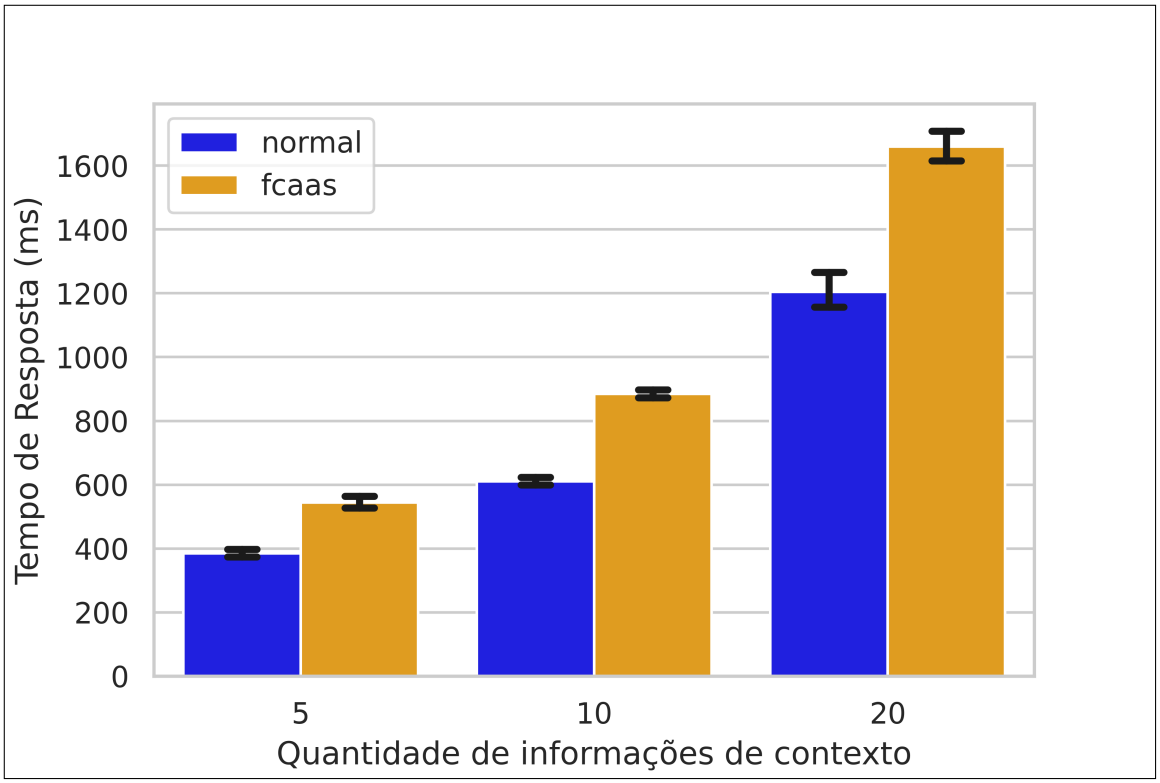
É importante esclarecer que essa ação de destruir os *containers* após cada conjunto de requisições teve o objetivo de evitar que possíveis instabilidades no Docker afetassem os experimentos, uma vez que não foi usado nenhum gerenciador de containers, como Docker Swarm e Kubernetes, que pudesse reconstruir os containers ou replicá-los em caso de falhas. Assim, ao reconstruir os containers os módulos da arquitetura recebiam cada bloco de requisições sem influência do anterior. O mesmo justifica o intervalo de tempo antes do início das requisições. Cabe ressaltar que tais decisões foram tomadas com base em experimentos prévios durante a definição e construção do ambiente desta prova de conceito e com o intuito de mitigar eventuais interferências dos elementos do cenário sobre os resultados.

Ao final dos seis experimentos, foi calculado o tempo de resposta médio usando a média dos tempos de respostas obtidos em cada experimento e o desvio padrão. Os resultados são apresentados e discutidos na subseção seguinte.

5.4.1 Resultados e discussão

A Figura 16 apresenta um gráfico com os resultados obtidos no Experimento 2, relacionando o tempo de resposta médio em milissegundos (ms) por quantidade de informações de contexto requisitadas em um cenário com e sem a atuação da arquitetura, com intervalo de confiança de 95%.

Figura 16 – Resultados Experimento 2.



Fonte: elaborado pelo autor.

A Tabela 4 sumariza alguns resultados obtidos, agrupados pela quantidade de informações requeridas, a saber: tempo de resposta médio, obtido pela média dos 50 blocos de requisições, desvio padrão dessas médias e maior e menor tempo médio obtido considerando todos os blocos de requisições para cada cenário.

Em uma visão geral, pode-se observar que existe uma diferença no tempo de resposta nos cenários em que existiu a atuação da arquitetura em comparação ao cenário normal. Tal comportamento era esperado, uma vez que as funções de segurança da FCAAS-IoT envolvem análise das políticas de controle de acesso e de criptografia, obtenção das informações de contexto e encriptação dos dados, tarefas estas que implicam processamento e recuperação de dados, além da comunicação entre os módulos.

Entretanto, quando analisa-se a diferença entre os resultados obtidos com e sem a

Tabela 4 – Resultados do experimento 2.

Quantidade de Informações	Cenário	Tempo de Resposta Médio (ms)	Desvio Padrão das Médias (ms)	Menor Tempo de Resposta (ms)	Maior Tempo de Resposta (ms)
5	Normal	385	44	331	609
	FCAAS-IoT	544	67	478	801
10	Normal	610	45	554	755
	FCAAS-IoT	885	44	807	1003
20	Normal	1203	197	1085	2261
	FCAAS-IoT	1659	166	1487	2424

arquitetura, percebe-se que o tempo acrescido foi inferior a meio segundo (500 ms), a saber 159, 275 e 456 ms, para 5, 10 e 20 informações requeridas, respectivamente. Assim, o tempo de resposta adicionado pela atuação da FCAAS-IoT pode não prejudicar o funcionamento da maioria das aplicações consumidoras de contexto, mesmo que estas tenham restrições altas de tempo.

Ainda observando a comparação do tempo médio com e sem a FCAAS-IoT, é possível destacar o aumento progressivo das diferenças entre os tempos de resposta com a quantidade de informações que compõem o contexto solicitado. A variação na quantidade de fontes e de informações requeridas impacta na quantidade de políticas que serão recuperadas e consequentemente analisadas. Logo, tal tendência era esperada, principalmente quando se leva em consideração que cada informação de contexto tem suas próprias políticas, as quais devem ser avaliadas em sua totalidade para cada item do contexto solicitado, para decidir sobre a permissão ou negação de acesso.

Esse aspecto deve ser levado em consideração ao projetar o esquema de políticas de controle de acesso. Por exemplo, quanto mais políticas existirem por informação, maior será o tempo de análise de autorização, já que todas serão consideradas no processo decisório. Em escopos que trabalhem com contextos com muitos tipos de informação, mas dependam de baixos tempos de resposta, é importante haver um planejamento na construção de políticas, fazendo com que, por exemplo, todas as regras de acesso a respeito de uma informação constem apenas em uma política ou que uma mesma política verse sobre mais de uma informação de contexto.

Entretanto, é relevante frisar que a busca por melhor QoS não deve resultar em uma redução na segurança, principalmente da privacidade, o que pode ocorrer, caso sejam adotadas políticas mais genéricas, que englobem muitas informações objetivando reduzir a quantidade de políticas analisadas para cada requisição.

Cabe ressaltar, ainda, que o fato de FCAAS-IoT possuir flexibilidade em sua imple-

mentação, pode auxiliar em reduzir o tempo de resposta dependendo das demandas e especificidades de cada cenário. Implementações centralizadas, por exemplo, conseguem reduzir do tempo de resposta total pela minimização do atraso de comunicação entre os módulos. Outro exemplo, tecnologias de *Edge* e *Fog Computing* podem ser usadas para multiplicar as instâncias dos módulos, levando-as para mais próximo de consumidores e fontes e otimizando o processamento de requisições, além da possibilidade de implementar balanceamento de carga e tolerância a falhas, dentre outras questões.

5.5 Considerações finais

Nesse capítulo foi relatada a prova de conceito elaborada para analisar alguns aspectos da FCAAS-IoT, mais especificamente a efetividade das funções de segurança propostas e o impacto sobre o tempo de resposta no fornecimento de contexto.

Os experimentos realizados via emulação demonstraram que os mecanismos de segurança, as atividades exercidas pelos módulos, sua comunicação e as políticas de controle de acesso e criptografia conseguem atuar de forma correta, protegendo as informações de contexto em vários arranjos de composição e origem e destino de dados.

Além disso, as diferenças no tempo de resposta em relação a um cenário sem a atuação da FCAAS-IoT não foram tão significativas mesmo diante de contextos compostos por uma maior quantidade de informações, o que pode ser contornado com ajustes no funcionamento das aplicações ou em combinações na implantação dos módulos.

6 CONCLUSÕES E TRABALHOS FUTUROS

Diante dos desafios de segurança em IoT e das especificidades das aplicações sensíveis ao contexto nesse ambiente, este trabalho propôs uma arquitetura denominada FCAAS-IoT para atuar entre as entidades que fornecem informações de contexto e as aplicações que as consomem a fim de prover compartilhamento de contexto seguro. As funções dos módulos e a forma como estes interagem, bem como os modelos de políticas adotados, buscam tornar a arquitetura adaptável a diversos cenários de fornecimento de contexto e de infraestrutura do ambiente.

Os experimentos realizados demonstraram que os procedimentos realizados pelos componentes da arquitetura e os modelos de segurança definidos funcionam para diferentes composições de contexto e combinações de fontes e consumidores. Em relação ao tempo de resposta, o acréscimo obtido em relação a um cenário sem a arquitetura não inviabiliza a adoção da FCAAS-IoT e pode ser reduzido com adequações na implantação dos módulos e na definição de políticas graças à flexibilidade incorporada ao projeto.

Assim, FCAAS-IoT consegue utilizar o modelo ABAC e informações contextuais de segurança para incorporar confidencialidade e privacidade no compartilhamento de contexto, com possibilidade de adaptação a diferentes arranjos devido sua estrutura modularizada, podendo integrar-se a outras tecnologias como NFV, computação em nuvem, névoa e de borda.

Entretanto, são necessários experimentos a fim de avaliar o desempenho da FCAAS-IoT em outros cenários, variando tecnologias de implantação dos módulos, quantidade de atributos usados nas solicitações e políticas e diferenças de funcionamento em arranjos centralizados e distribuídos, por exemplo.

Diante do apresentado, pretende-se como extensões desta pesquisa em trabalhos futuros:

a) Aprimorar o uso de informações para contexto de solicitação, consolidando um modelo de políticas com linguagem padronizada, que facilite a definição de regras e a utilização de um conjunto abrangente de informações;
b) Inserir técnicas de *machine learning* no processo de análise das políticas, principalmente nas políticas de criptografia, auxiliando na seleção de atributos que deverão ser considerados para recuperação de políticas a serem analisadas e recuperadas, mediante as condições da rede e nível de segurança pretendido;
c) Verificar a possibilidade de integração dos elementos da arquitetura com outras

ferramentas de segurança utilizadas, sobretudo, em ambientes de rede tradicionais, como *firewalls, IDSs e IPSs*, de modo a fornecer e consumir funcionalidades que agreguem no provimento de segurança, criando um ecossistema de compartilhamento de contexto seguro, que pode ser aproveitado, inclusive para outros tipos de aplicações na IoT;

d) Avaliar diferentes implementações da arquitetura, variando contextos de aplicação, tecnologias para implantação dos módulos e arranjos de composição destes, explorando tecnologias como NFV, *cloud, fog* e *edge computing*, a fim de identificar potencialidades, problemas e outros aspectos que devam interfiram na execução das funções de segurança;

e) Sistematizar a implantação da arquitetura, disponibilizando um guia que oriente a respeito de definição de políticas, tecnologias a serem usadas para criação dos módulos, bem como suas implicações no desempenho e funcionamento da arquitetura.

REFERÊNCIAS

ABOMHARA, M.; KøIEN, G. M. Security and privacy in the internet of things: Current status and open issues. *In*: 2014 INTERNATIONAL CONFERENCE ON PRIVACY AND SECURITY IN MOBILE SYSTEMS (PRISMS), 2014, Aalborg. **Anais...** [*S. l.*]: IEEE, 2014. p. 1–8.

ABOWD, G. D.; DEY, A. K.; BROWN, P. J.; DAVIES, N.; SMITH, M.; STEGGLES, P. Towards a better understanding of context and context-awareness. *In*: GELLERSEN, H.-W. (ed.). **Handheld and Ubiquitous Computing. HUC 1999. Lecture Notes in Computer Science**. Heidelberg: Springer, 1999. v. 1707, p. 304–307. ISBN 978-3-540-48157-7.

ABOWD, G. D.; MYNATT, E. D. Charting past, present, and future research in ubiquitous computing. **ACM Trans. Comput.-Hum. Interact.**, Association for Computing Machinery, New York, v. 7, n. 1, p. 29–58, mar. 2000.

AL-FUQAHA, A.; GUIZANI, M.; MOHAMMADI, M.; ALEDHARI, M.; AYYASH, M. Internet of things: A survey on enabling technologies, protocols, and applications. **IEEE Communications Surveys & Tutorials**, IEEE, [*s.l.*], v. 17, n. 4, p. 2347–2376, 2015.

AL-MUHTADI, J.; SALEEM, K.; AL-RABIAAH, S.; IMRAN, M.; GAWANMEH, A.; RODRIGUES, J. J. A lightweight cyber security framework with context-awareness for pervasive computing environments. **Sustainable Cities and Society**, Elsevier, [*s.l.*], v. 66, p. 102610, 2021.

ALABA, F. A.; OTHMAN, M.; HASHEM, I. A. T.; ALOTAIBI, F. Internet of things security: A survey. **Journal of Network and Computer Applications**, Elsevier, [*s.l.*], v. 88, p. 10–28, 2017. ISSN 1084-8045.

ALAM, I.; SHARIF, K.; LI, F.; LATIF, Z.; KARIM, M. M.; BISWAS, S.; NOUR, B.; WANG, Y. A survey of network virtualization techniques for internet of things using sdn and nfv. **ACM Comput. Surv.**, Association for Computing Machinery, New York, v. 53, n. 2, p. 1–40, abr. 2020. ISSN 0360-0300.

ANDREA, I.; CHRYSOSTOMOU, C.; HADJICHRISTOFI, G. Security and privacy in the internet of things: Current status and open issues. *In*: IEEE SYMPOSIUM ON COMPUTERS AND COMMUNICATION (ISCC), 2014, Larnaca. **Anais...** [*S. l.*]: IEEE, 2015. p. 180–187.

ARFAOUI, A.; CHERKAOUI, S.; KRIBECHE, A.; SENOUCI, S. M.; HAMDI, M. Context for ubiquitous data management. *In*: INTERNATIONAL WORKSHOP ON UBIQUITOUS DATA MANAGEMENT, 2005, Tokyo. **Anais...** [*s.l.*]: IEEE, 2005. p. 17–24.

ARFAOUI, A.; CHERKAOUI, S.; KRIBECHE, A.; SENOUCI, S. M.; HAMDI, M. Context-aware adaptive authentication and authorization in internet of things. *In*: IEEE INTERNATIONAL CONFERENCE ON COMMUNICATIONS (ICC), 2019, Shanghai. **Anais...** [*S. l.*]: IEEE, 2019. p. 1–6.

ATZORI, L.; IERA, A.; MORABITO, G. The internet of things: A survey. **Computer Networks**, Elsevier, [*s.l.*], v. 54, n. 15, p. 2787–2805, out. 2010. ISSN 1389-1286.

BORGIA, E. The internet of things vision: Key features, applications and open issues. **Computer Communications**, Elsevier, [*s.l.*], v. 54, p. 1–31, dez. 2014. ISSN 0140-3664.

BREZILLON, P.; MOSTEFAOUI, G. K. Context-based security policies: a new modeling approach. *In*: IEEE ANNUAL CONFERENCE ON PERVASIVE COMPUTING AND COMMUNICATIONS WORKSHOPS, 2004, Orlando. **Anais...** [*s.l.*]: IEEE, 2004. p. 154–158.

DEY, A. K. Understanding and using context. **Personal and Ubiquitous Computing**, Springer, [*s.l.*], v. 5, n. 1, p. 4 – 7, fev. 2001.

DONNO, M. D.; TANGE, K.; DRAGONI, N. Foundations and evolution of modern computing paradigms: Cloud, iot, edge, and fog. **IEEE Access**, IEEE, [*s.l.*], v. 7, p. 150936–150948, 2019. ISSN 2169-3536.

ELAZHARY, H. Internet of things (iot), mobile cloud, cloudlet, mobile iot, iot cloud, fog, mobile edge, and edge emerging computing paradigms: Disambiguation and research directions. **ACM Comput. Surv.**, Elsevier, [*s.l.*], v. 128, p. 105–140, fev. 2019.

ETSI. **Network functions virtualisation (nfv): Architectural framework**. Sophia Antipolis, 2013. v. 1, n. 2.

FARRIS, I.; TALEB, T.; KHETTAB, Y.; SONG, J. A survey of network virtualization techniques for internet of things using sdn and nfv. **IEEE Communications Surveys & Tutorials**, IEEE, [*s.l.*], v. 21, n. 1, p. 812–837, 2019. ISSN 1553-877X.

GABILLON, A.; GALLIER, R.; BRUNO, E. Access controls for iot networks. **SN Computer Science**, Springer, [*s.l.*], v. 1, n. 1, p. 1–13, set. 2020.

GUBBI, J.; BUYYA, R.; MARUSIC, S.; PALANISWAMI, M. Internet of things (iot): A vision, architectural elements, and future directions. **Future Generation Computer Systems**, Elsevier, [*s.l.*], v. 29, n. 7, p. 1645–1660, set. 2013. ISSN 0167-739X.

HABIB, K.; LEISTER, W. Context-aware authentication for the internet of things. *In*: INTERNATIONAL CONFERENCE ON AUTONOMIC AND AUTONOMOUS SYSTEMS, 11., 2015, Rome. **Anais...** [*s.l.*]: [*s.n.*], 2015. p. 1–6.

HASSIJA, V.; CHAMOLA, V.; SAXENA, V.; JAIN, D.; GOYAL, P.; SIKDAR, B. A survey on iot security: application areas, security threats, and solution architectures. **IEEE Access**, IEEE, [*s.l.*], v. 7, p. 82721–82743, jun. 2019. ISSN 2169-3536.

HU, V. C.; FERRAIOLO, D.; KUHN, R.; FRIEDMAN, A. R.; LANG, A. J.; COGDELL, M. M.; SCHNITZER, A.; SANDLIN, K.; MILLER, R.; SCARFONE, K. *et al.* **Guide to attribute based access control (ABAC): definition and considerations**. Gaithersburg, 2013.

HU, V. C.; KUHN, D. R.; FERRAIOLO, D. F.; VOAS, J. Attribute-based access control. **Computer**, IEEE, [*s.l.*], v. 48, n. 2, p. 85–88, fev. 2015. ISSN 1558-0814.

HUSSAIN, F.; HUSSAIN, R.; HASSAN, S. A.; HOSSAIN, E. Machine learning in iot security: Current solutions and future challenges. **IEEE Communications Surveys & Tutorials**, IEEE, [*s.l.*], v. 22, n. 3, p. 1686–1721, abr. 2020. ISSN 1553-877X.

KALYANI, Y.; COLLIER, R. A systematic survey on the role of cloud, fog, and edge computing combination in smart agriculture. **Sensors**, MDPI, [*s.l.*], v. 21, n. 17, p. 5922, set. 2021.

KHAN, M. A.; SALAH, K. Iot security: Review, blockchain solutions, and open challenges. **Future Generation Computer Systems**, Elsevier, [*s.l.*], v. 82, p. 395–411, maio 2018.

KHAN, W. Z.; AHMED, E.; HAKAK, S.; YAQOOB, I.; AHMED, A. Edge computing: A survey. **Future Generation Computer Systems**, Elsevier, [*s.l.*], v. 97, p. 219–235, ago. 2019.

KOUICEM, D. E.; BOUABDALLAH, A.; LAKHLEF, H. Internet of things security: A top-down survey. **Computer Networks**, Elsevier, [*s.l.*], v. 141, p. 199–221, ago. 2018.

LAROUI, M.; NOUR, B.; MOUNGLA, H.; CHERIF, M. A.; AFIFI, H.; GUIZANI, M. Edge and fog computing for iot: A survey on current research activities & future directions. **Computer Communications**, Elsevier, [*s.l.*], v. 180, p. 210–231, dez. 2021.

LI, S.; XU, L. D.; ZHAO, S. 5g internet of things: A survey. **Journal of Industrial Information Integration**, Elsevier, [*s.l.*], v. 10, p. 1–9, jun. 2018.

LIN, J.; YU, W.; ZHANG, N.; YANG, X.; ZHANG, H.; ZHAO, W. A survey on internet of things: Architecture, enabling technologies, security and privacy, and applications. **IEEE Internet of Things Journal**, IEEE, [*s.l.*], v. 4, n. 5, p. 1125 – 1142, mar. 2017. ISSN 2327-4662.

LIU, C. Pervasive context sharing in magpie: Adaptive trust-based privacy protection. *In*: INTERNATIONAL CONFERENCE ON MOBILE COMPUTING, APPLICATIONS, AND SERVICES, 2015, Berlin. **Anais...** [*s.l.*]: Springer, 2015. p. 122–139.

MAHALLE, P. N.; DHOTRE, P. S. Security issues in context-aware systems. *In*: ____. **Context-Aware Pervasive Systems and Applications**. Singapore: Springer Singapore, 2020. cap. 7, p. 137–149.

MAHMOUD, R.; YOUSUF, T.; ALOUL, F.; ZUALKERNAN, I. Internet of things (iot) security: Current status, challenges and prospective measures. *In*: INTERNATIONAL CONFERENCE FOR INTERNET TECHNOLOGY AND SECURED TRANSACTIONS (ICITST), 2015, London. **Anais...** [*s.l.*]: IEEE, 2016.

MATOS, E. de; TIBURSKI, R. T.; AMARAL, L. A.; HESSEL, F. Providing context-aware security for iot environments through context sharing feature. *In*: IEEE INTERNATIONAL CONFERENCE ON TRUST, SECURITY AND PRIVACY IN COMPUTING AND COMMUNICATIONS/ 12TH IEEE INTERNATIONAL CONFERENCE ON BIG DATA SCIENCE AND ENGINEERING (TRUSTCOM/BIGDATASE), 17., 2018, New York. **Anais...** [*s.l.*]: IEEE, 2018.

MATOS, E. de; TIBURSKI, R. T.; MORATELLI, C. R.; FILHO, S. J.; AMARAL, L. A.; RAMACHANDRAN, G.; KRISHNAMACHARI, B.; HESSEL, F. Context information sharing for the internet of things: A survey. **Computer Networks**, Springer, [*s.l.*], v. 166, jan. 2020.

MELL, P.; GRANCE, T. *et al.* **The NIST Definition of Cloud Computing**. Gaithersburg, 2011.

MIJUMBI, R.; SERRAT, J.; GORRICHO, J.; BOUTEN, N.; TURCK, F. D.; BOUTABA, R. Network function virtualization: State-of-the-art and research challenges. **IEEE Communications Surveys & Tutorials**, IEEE, [*s.l.*], v. 18, n. 1, p. 236–262, 2016. ISSN 1553-877X.

MIJUSKOVIC, A.; CHIUMENTO, A.; BEMTHUIS, R.; ALDEA, A.; HAVINGA, P. Resource management techniques for cloud/fog and edge computing: An evaluation framework and classification. **Sensors**, MDPI, [*s.l.*], v. 21, n. 5, p. 1832, mar. 2021.

MIORANDI, D.; SICARI, S.; DE PELLEGRINI, F.; CHLAMTAC, I. Internet of things: Vision, applications and research challenges. **Ad Hoc Networks**, Elsevier, [*s.l.*], v. 10, n. 7, p. 1497–1516, set. 2012.

MOSTEFAOUI, G. K.; BREZILLON, P. Context-aware adaptive authentication and authorization in internet of things. *In*: IEEE ANNUAL CONFERENCE ON PERVASIVE COMPUTING AND COMMUNICATIONS WORKSHOPS, 2004, Orlando. **Anais...** [*s.l.*]: IEEE, 2004. p. 28–32.

PERERA, C.; ZASLAVSKY, A.; CHRISTEN, P.; GEORGAKOPOULOS, D. Context aware computing for the internet of things: A survey. **IEEE Communications Surveys & Tutorials**, IEEE, [*s.l.*], v. 16, n. 1, p. 414–454, 2014. ISSN 1553-877X.

PRADEEP, P.; KRISHNAMOORTHY, S. The mom of context-aware systems: A survey. **Computer Communications**, Elsevier, [*s.l.*], v. 137, p. 44–69, mar. 2019.

PSARRA, E.; VERGINADIS, Y.; PATINIOTAKIS, I.; APOSTOLOU, D.; MENTZAS, G. A context-aware security model for a combination of attribute-based access control and attribute-based encryption in the healthcare domain. *In*: WEB, ARTIFICIAL INTELLIGENCE AND NETWORK APPLICATIONS, 2020, Caserta. **Anais...** [*s.l.*]: Springer, 2019. p. 1133–1142.

QIU, J.; TIAN, Z.; DU, C.; ZUO, Q.; SU, S.; FANG, B. A survey on access control in the age of internet of things. **Internet of Things Journal**, IEEE, [*s.l.*], v. 7, n. 6, p. 4682–4696, jun. 2020. ISSN 2327-4662.

RAMOS, J. L. H.; BERNABE, J. B.; SKARMETA, A. F. Managing context information for adaptive security in iot environments. *In*: INTERNATIONAL CONFERENCE ON ADVANCED INFORMATION NETWORKING AND APPLICATIONS WORKSHOPS, 29., 2015, Gwangju. **Anais...** [*s.l.*]: IEEE, 2015.

SAMIE, F.; BAUER, L.; HENKEL, J. Iot technologies for embedded computing: A survey. *In*: INTERNATIONAL CONFERENCE ON HARDWARE/SOFTWARE CODESIGN AND SYSTEM SYNTHESIS (CODES+ISSS), 2016, Pittsburgh. **Anais...** [*s.l.*]: IEEE, 2016. p. 1–10.

SERVOS, D.; OSBORN, S. L. Current research and open problems in attribute-based access control. **SN Computer Science**, Association for Computing Machinery, New York, v. 49, n. 4, p. 45, jan. 2017. ISSN 0360-0300.

SEZER, O. B.; DOGDU, E.; OZBAYOGLU, A. M. Context-aware computing, learning, and big data in internet of things: A survey. **Internet of Things Journal**, IEEE, [*s.l.*], v. 5, n. 1, p. 1–27, nov. 2018. ISSN 2327-4662.

SHA, K.; YANG, T. A.; WEI, W.; DAVARI, S. A survey of edge computing-based designs for iot security. **Digital Communications and Networks**, Elsevier, [*s.l.*], v. 6, n. 1, p. 195–202, maio 2020.

SHI, W.; CAO, J.; ZHANG, Q.; LI, Y.; XU, L. Edge computing: Vision and challenges. **Internet of Things Journal**, IEEE, [*s.l.*], v. 3, n. 5, p. 637–646, out. 2016. ISSN 2327-4662.

SYLLA, T.; CHALOUF, M. A.; KRIEF, F.; SAMAKÉ, K. Towards a context-aware security and privacy as a service in the internet of things. *In*: WISTP 2019: INFORMATION SECURITY THEORY AND PRACTICE, 2019, Paris. **Anais...** [*s.l.*]: Springer, 2019. p. 240 – 252.

WANG, J.; WANG, H.; ZHANG, H.; CAO, N. Trust and attribute-based dynamic access control model for internet of things. *In*: INTERNATIONAL CONFERENCE ON CYBER-ENABLED DISTRIBUTED COMPUTING AND KNOWLEDGE DISCOVERY (CYBERC), 2017, Nanjing. **Anais...** [*s.l.*]: IEEE, 2017. p. 342–345.

XIAO, L.; WAN, X.; LU, X.; ZHANG, Y.; WU, D. Iot security techniques based on machine learning: How do iot devices use ai to enhance security? **IEEE Signal Processing Magazine**, IEEE, [*s.l.*], v. 35, n. 5, p. 41–49, set. 2018. ISSN 1053-5888.

YANG, Y.; WU, L.; YIN, G.; LI, L.; ZHAO, H. A survey on security and privacy issues in internet-of-things. **IEEE Internet of Things Journal**, IEEE, [*s.l.*], v. 4, n. 5, p. 1250–1258, abr. 2017. ISSN 2327-4662.

YI, B.; WANG, X.; LI, K.; DAS, S. k.; HUANGL, M. A comprehensive survey of network function virtualization. **Computer Networks**, Elsevier, [*s.l.*], v. 133, p. 212–262, mar. 2018.

YI, S.; HAO, Z.; QIN, Z.; LI, Q. Fog computing: Platform and applications. *In*: THIRD IEEE WORKSHOP ON HOT TOPICS IN WEB SYSTEMS AND TECHNOLOGIES (HOTWEB), 2015, Washington. **Anais...** [*s.l.*]: IEEE, 2015. p. 73– 78.