Uma Proposta de Arquitetura para Autorização Federada com Internet das Coisas

DA SILVA, Gabriela and DA SILVA, Carlos <http://orcid.org/0000-0001-9608-439X>

Uma Proposta de Arquitetura para Autorização Federada com Internet das Coisas

Carlos Eduardo da Silva[1], **Gabriela Cavalcante da Silva**[1]

[1]Instituto Metrópole Digital - Universidade Federal do Rio Grande do Norte (UFRN)
Avenida Senador Salgado Filho, 3000, Lagoa Nova – CEP: 59.078-970 – Natal/RN

gabicavalcantesilva@gmail.com, kaduardo@imd.ufrn.br

Abstract. *Internet of Things (IoT) has been applied to several application domains, including as mechanisms for physical access control. However, existing solutions do not take into account appropriate protocols for the process of authentication and authorization. In this context, this paper presents a proposal of architecture for physical access control based on IoT and integrated with mechanism for federated authentication and authorization based on the RBAC model.*

Resumo. *A Internet das coisas (Internet of Things - IoT) vem sendo utilizada em diversos domínios de aplicação, incluindo como mecanismos para controle de acesso físico. Entretanto, as soluções existentes não consideram o uso de protocolos apropriados no processo de autenticação e autorização. Neste contexto, este artigo apresenta uma proposta de arquitetura para controle de acesso físico baseado em IoT e integrada a mecanismos de autenticação federada e autorização baseada no modelo RBAC.*

1. Introdução

Existem diversas definições para o que seria Internet das Coisas (*Internet of Things* - IoT) [Minerva et al. 2015], que apresentam como características em comum a existência de dispositivos e sistemas que interligam sensores e atuadores do mundo real à Internet. Neste contexto, IoT pode ser enxergada sob a perspectiva de sistema ciber-físicos (*Cyber-Physical Systems* - CPS). Os CPSs são compostos por um conjunto de agentes em rede, incluindo sensores, atuadores, unidades de processamento e controle e dispositivos de comunicação [Cardenas et al. 2008], que executam embarcados em dispositivos ou em ambiente centralizado, como uma nuvem computacional.

Dentre as diversas áreas de aplicação para IoT (por exemplo, carros conectados [Evans-Pughe 2005], dispositivos portáteis, dispositivos de monitoramento de saúde, bem estar e *fitness* [Islam et al. 2015], sensores de rede sem fio [Lewis 2004], entre outros) é possível encontrar seu uso para controle de acesso físico. Neste contexto, é possível identificar soluções[1] que visam o uso de IoT para controle de acesso físico, como a solução formada por um Arduino e um leitor RFID para controlar portas[2]. Entretanto, baseado em informações públicas, essas soluções não consideram o uso de protocolos seguros nos processo de autenticação e autorização, se baseando no identificador do cartão RFID e na

[1]https://kintronics.com/solutions/ip-door-access-control/
[2]http://www.instructables.com/id/Arduino-RFID-Door-Lock/

utilização de ACLs (*Access Control Lists* - listas de controle de acesso) para a definição de políticas de acesso.

Neste contexto, nosso objetivo é abordar a utilização da IoT para modelarmos um *Cyber-Physical System* que forneça controle de acesso físico através de uma autenticação federada. Mais especificamente, este artigo apresenta uma proposta de arquitetura para permitir integração entre federação de identidades, controle de acesso e Internet das Coisas (IoT - Internet of Things). Com isso, um usuário poderá transpor suas credenciais de uma federação de identidades (como a federação CAFe da RNP) para o mundo físico. Como estudo de caso, iremos desenvolver um protótipo de sistema de controle de acesso físico baseado em IoT.

Este artigo está organizado da seguinte forma: a seção 2 apresenta uma breve introdução sobre autorização, seguida de algumas soluções existentes que exploram IoT para controle de acesso. Na seção 3 temos a nossa proposta de arquitetura, e por fim, concluímos na seção 5.

2. Contextualização e Trabalhos Relacionados

O modelo de identidades federadas [Chadwick 2009] permite resolver o problema de múltiplas credenciais de acesso. Neste modelo, cada usuário só precisa gerenciar uma única credencial de acesso e essa o permite acessar diferentes provedores de serviço. Porém, esse acesso só é liberado após a autorização ser dada pelas políticas de segurança de cada provedor de serviço. Estes procedimentos constituem mecanismos que permitem a realização de controle de acesso.

Existem diversos modelos para a especificação de políticas de controle de acesso. Um dos modelos mais difundidos é o RBAC/ABAC, em que um é baseado em papéis (RBAC - *Role based access control* [Sandhu et al. 1996]) e outro em atributos (ABAC - *Attribute Based Access Control* [Hu et al. 2014b]). Nestes modelos, permissões são associadas a papés/atributos que são atribuídos aos usuários, de forma que um acesso será concedido caso o usuário requisitante tenha o papel/atributo especificado na política. O modelo ABAC se apresenta como uma evolução do RBAC e visa herdar as melhores práticas de modelos anteriores e tratar atributos variáveis e específicos de um ambiente, ao invés de considerar o papel como único atributo.

Um mecanismo de controle de acesso, como os baseados nos modelos RBAC/ABAC, seguem uma arquitetura funcional similar, na qual pode-se identificar um conjunto de componentes. A Figura 1 apresenta esses componentes e a interação entre eles. Quando um indivíduo solicita uma operação em um determinado objeto, essa solicitação é interceptada pelo *Policy Enforcement Point* - **PEP**. O PEP tem o papel de proteger o objeto e de fazer cumprir as decisões de controle de acesso, permitindo ou não o acesso do sujeito (*subject*) aos recursos (*object*). Para isso, o PEP interage com o **PDP** (*Policy Decision Point*). O PDP é o componente responsável por tomar uma decisão permitindo ou não o acesso requisitado. Essas decisão são baseadas nas políticas de controle de acesso, armazenadas em um repositório (*Policy Repository*). O PDP utiliza também o *Policy Information Point* - **PIP** para obter informações que o auxiliarão na tomada de decisão, tais como atributos do sujeito e/ou objeto (a partir do *Attribute Repository*) e condições do ambiente (*Environment Conditions*). Uma vez de posse das informações necessárias, o PDP toma uma decisão e informa a mesma para o PEP, que por sua vez, permite ou não

o acesso ao objeto. Políticas de controle de acesso são administradas através do *Policy Administration Point* - **PAP**.

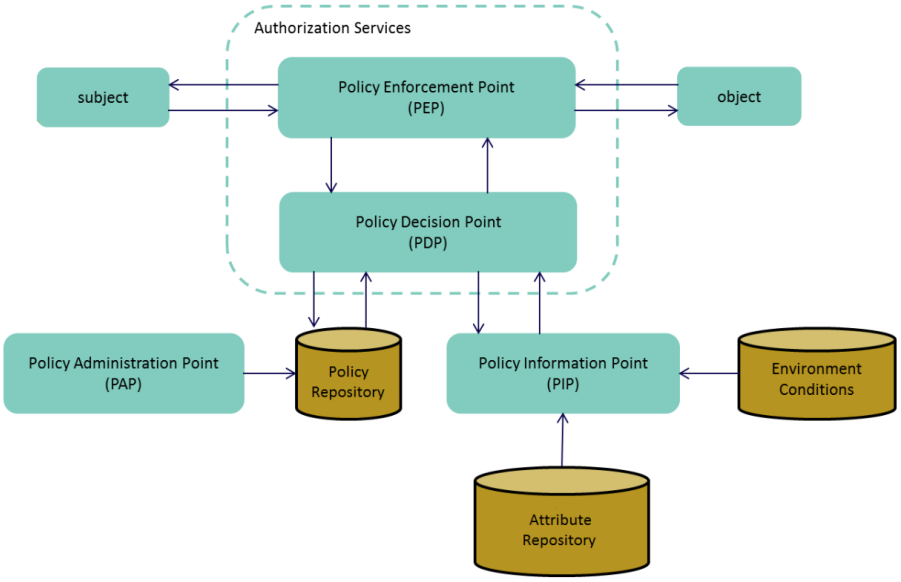

Figura 1. Exemplo de Pontos Funcionais de Autorização [Hu et al. 2014a]

Nos voltando para o contexto de IoT, encontramos algumas soluções que propõem desde arquitetura até topologias com IoT para autorização e controle de acesso. Uma dessas soluções pode ser encontrada em [Fremantle et al. 2014], que traz como principal contribuição verificar se seria viável e eficaz usar OAuth2.0 [Hardt 2012] como parte do fluxo do protocolo *Message Queue Telemetry Transport* (MQTT)[3] e também dentro de um MQTT *broker* para tomadas decisões de controle de acesso federadas e direcionadas pelo usuário. Outra solução considerada foi [Liu et al. 2012]. Neste trabalho, os autores propõem uma arquitetura abstrata para autenticação de usuários de Internet das Coisas, identificando uma série de componentes como um gateway para coisas, que também atua como um provedor de serviço usando OpenID Connect para autenticar usuários. Eles também apresentam um método de autenticação e controle de acesso baseado nos componentes definidos.

Em [Ndibanje et al. 2014] é feita uma análise sobre autenticação e controle de acesso do método usando IoT apresentado em [Liu et al. 2012]. Segundo os autores, o protocolo apresentado é custoso na troca de mensagens e a avaliação de segurança não é suficientemente forte. A primeira crítica levantada por [Ndibanje et al. 2014], foi sobre a falta de detalhes claros sobre todo o processo de autenticação em relação às mensagens trocadas. Em seguida, ele destaca a não separação entre as principais etapas conhecidas de um processo de autenticação normal, como a fase de registro e a fase de login. E questiona a contribuição para o aspecto de controle de acesso, [Liu et al. 2012] não possui uma proposta de esquema e não descreve efetivamente como o RBAC seria usado no projeto.

Este levantamento preliminar mostrou que a utilização de IoT em cenário de controle de acesso físico está começando a receber atenção da comunidade, com diversas

[3]http://mqtt.org/documentation

soluções em desenvolvimento. Entretanto, essas soluções não consideram questões relacionadas a múltiplos fatores de autenticação, ou a utilização de protocolos de autenticação e autorização adequados. Algumas abordagens têm atacado este problema com foco em OpenID Connect, ou com SAML [Domenech et al. 2016, Domenech 2015], onde foi explorado a utilização de um cliente "ativo" que permita com que coisas possam realizar a autenticação em uma federação de identidades. Entretanto, notamos uma ausência de trabalhos que explorem o perfil ECP definido pela especificação SAML para permitir a autenticação federada em ambientes não-Web.

3. Proposta de Arquitetura

Apesar da falta de clareza ou fragilidade encontrada em parte das soluções analisadas [Ndibanje et al. 2014], o estudo delas foi importante para podermos modelar nossa proposta de solução IoT levando em consideração o que poderíamos ou não incorporar, assim como o que seria possível trazer de novo. O resultado pode ser visto abaixo, na Figura 2.

Nossa arquitetura é composta por um *Controlador Físico*, atuando como um *gateway* para os dispositivos físicos (em nosso exemplo um leitor e um mecanismo de trava de uma porta). Esse controlador físico por sua vez é conectado a um *Controlador Lógico*, que representa um componente de software capaz de armazenar e processar os dados produzidos pelas "coisas" (representado pelo módulo *CORE*).

No tocante a uma federação de identidades, todo o conjunto dispositivos físicos, controladores físico e lógico constituem um provedor de serviço (*Service Provider*). Como o *Controlador Lógico* apresenta maior poder computacional, ele é responsável por implementar os protocolos (representado pelo móduglo *SAML SP*) exigidos pela federação. Desse modo, o *Controlador Lógico* é responsável por interagir com um provedor de identidades (*Identity Provider* - IdP).

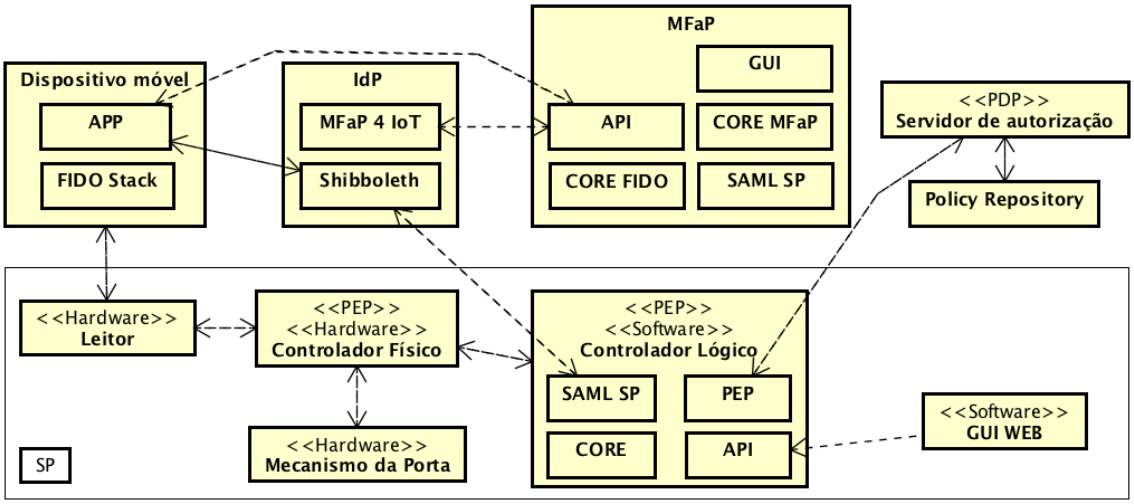

Figura 2. ***Diagrama de Componente do Protótipo IoT com AuthN/AuthZ em federação***

O IdP é o componente responsável pela autenticação do usuário, ele vai interagir com outro componente do nosso protótipo, o *Multi-Factor Provider* (MFaP), para fazer a

autenticação do usuário com o segundo fator. O MFaP é responsável por implementar um serviço FIDO (*FIDO UAF Server* representado pelo módulo *CORE FIDO*) que é acessado através de uma API REST. Desse modo, o IdP delega ao MFaP o processo de autenticação via FIDO através do módulo *MFaP 4 IoT* (um cliente para a API oferecida pelo MFaP). Adicionalmente, o MFaP também oferece um serviço de registro dos dispositivos móveis (módulo *CORE MFaP*) atuando como um SP dentro da federação que é acessado através de uma interface Web (módulgo *GUI*). O acesso à interface Web do MFaP se dá através do fluxo de autenticação normal via SAML (módulo *SAML SP*).

O dispositivo móvel do usuário deve possuir suporte ao FIDO (representado pelo módulo *FIDO Stack*) e contem um aplicativo desenvolvido por nós. O *FIDO Stack* é um componente de software embarcado no telefone móvel durante seu processo de fabricação, isto é, de acordo com a especificação [Machani et al. 2014], todo dispositivo FIDO UAF deverá passar por um processo de certificação promovido pela *FIDO Alliance*. Para o desenvolvimento desse protótipo, como será feito uso de telefone não certificado pelo FIDO Alliance, optou-se por fazer uso do aplicativo *Dummy FIDO UAF Client*[4] não certificado pelo FIDO, mas que implementa o protocolo FIDO UAF.

O aplicativo Android permite ao usuário cadastrar o dispositivo como autenticador FIDO junto ao MFaP, além de atuar como cliente para a solução de controle de acesso físico, interagindo com o *Leitor* e permitindo ao usuário se autenticar através do FIDO.

No que diz respeito a controle de acesso, as políticas de autorização, assim como os mecanismos de suporte são responsabilidade do PDP (*Policy Decision Point*). O *Controlador Físico* e *Controlador Lógico* fazem o papel de PEP (*Policy Enforcement Point*), sendo responsáveis por executar as decisões do PDP. Em nosso cenário, consideramos o uso de políticas XACML [OASIS 2010] baseadas no modelo RBAC/ABAC [Hu et al. 2014b]. Além disso, está sendo considerado o padrão FIDO UAF [Machani et al. 2014] como único fator de autenticação para o cenário com IoT.

Por uma questão de espaço processo, não detalharemos o processo de registro com diagramas, nos limitaremos a dar uma breve descrição. O processo de registro de um dispositivo FIDO UAF (no nosso exemplo é o telefone Android) se inicia com o usuário acessando o MFaP para habilitar o FIDO UAF como fator de autenticação. Como o MFaP é um provedor de serviço, temos então um fluxo de autenticação tradicional SAML. Nessa etapa, será possível realizar autenticação do usuário através do envio das credenciais via IdP. O MFaP então gera um QRCode que deverá ser lido pelo App no dispositivo móvel. Como o MFaP implementa FIDO UAF Server, ele possui um protocolo próprio de desafio e resposta para registrar uma chave pública que será gerada no dispositivo do usuário. De acordo com o protocolo FIDO UAF, o usuário precisa se autenticar localmente em seu dispositivo como parte do fluxo para geração do par de chaves. Se a autenticação do usuário em seu dispositivo ocorreu com sucesso, então o aplicativo mobile envia a resposta do desafio para o MFaP, que valida a resposta, salva em banco a chave pública enviada pelo aplicativo e envia para ele uma mensagem de sucesso.

Uma vez registrado o dispositivo, consideramos um cenário onde não seria desejado ou possível o usuário se autenticar com seu nome de usuário e senha como primeiro fator, como por exemplo, em sistemas de controle de acesso físico a ambientes. Sendo

[4] `https://github.com/emersonmello/dummyuafclient`

assim, o usuário, em posse de um dispositivo com FIDO UAF registrado previamente em seu IdP, faria uso desse como o único fator de autenticação. O fato do usuário ter que passar pela autenticação biométrica local em um dispositivo FIDO UAF certificado, garantia aqui a confiança de que o usuário é realmente quem diz ser. A Figura 3 apresenta um exemplo de autenticação de acordo com este cenário.

Em nosso cenário de exemplo, representado no diagrama de atividade na Figura 3, um usuário requisita abrir uma porta (atividade 1) aproximando seu dispositivo móvel a um leitor (por exemplo, via NFC ou *bluetooth*). O leitor então notifica o Controlador Físico da proximidade do dispositivo, e ele por sua vez requisita autenticação ao Controlador Lógico, que requisita o APP Cliente (atividades 2-4). O APP solicita o FIDO AuthN Request ao Servidor FIDO (atividade 5). Como resultado desta solicitação, o Servidor FIDO gera o desafio e o envia para o App Cliente (atividade 6). O usuário responde o desafio e envia de volta ao Servidor FIDO (atividade 7). Caso a resposta seja válida (atividade 8), o APP Cliente solicita um SAML Request ao Controlador Lógico (atividade 10), que gera e envia de volta o SAML Request para o App Cliente. O aplicativo manda este SAML Request juntamente com o ID do usuário e o FIDO AuthN Responde para o IdP ECP (atividade 12).

Quando o IdP recebe a mensagem vinda do dispositivo móvel, ele valida o AuthN Response junto ao Servidor FIDO (atividades 13-14), e então gera e envia um SAML Response para o App Cliente (atividade 15), que encaminha para o Controlador Lógico (atividade 16). O Controlador Lógico valida a mensagem (atividade 17) e disponibiliza e realiza o processo de autorização com base nos atributos do usuário (atividade 18). Por fim, a porta pode ser destravada (atividade 19).

4. Proposta de Implementação

Na arquitetura do SP para o cenário de controle de acesso físico de ambientes, projetamos dois protótipos pensadas para a solução com Bluetooth e com NFC. Na Figura 4 vemos a solução NFC, envolvendo Raspberry Pi, módulo Leitor RFID-RC522, e a trava magnética para porta como componentes de Hardware. Para a solução Bluetooth, o protótipo é similar, temos somente a substituição do módulo Leitor RFID-RC522 pelo USB Bluetooth Adapter.

A interface GUI provida pelo *Controlador Lógico* é implementada usando *framework* Flask[5]. No *Controlador Lógico* teremos o Apache httpd com módulo mod_shib, para que possa assim atuar como um provedor de serviço (SP) Shibboleth e permitir a interação com IdPs da federação.

Para o SP solicitar a autenticação do usuário com mais de um fator ao IDP, é necessário utilizar o MFA Profile [Refeds 2017]. O SP utilizará o elemento <*RequestedAuthnContext*> na mensagem SAML para informa ao IdP como esse deverá autenticar seus usuários.

Por parte do IdP é necessário criar um novo fluxo de autenticação dentro de um contexto de configuração MFA. A comunicação entre IdP e FIDO exige que a API REST do MFaP seja acessada pelo IdP, mais precisamente pelo authnFlow com os parâmetros esperados. Os principais métodos são de requisição e resposta da operação de registro,

[5]http://flask.pocoo.org

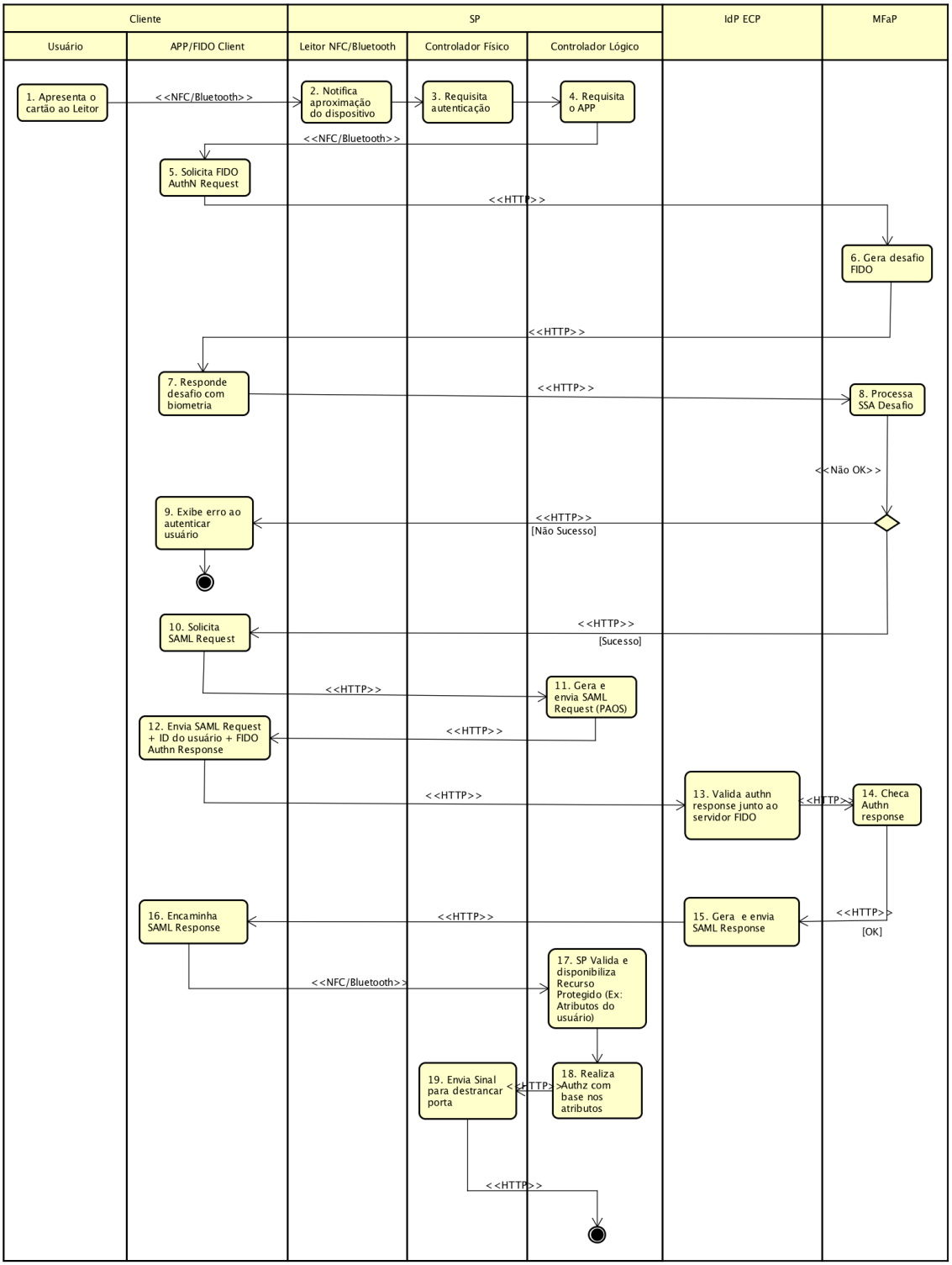

Figura 3. ***Diagrama de Atividade do Protótipo IoT com AuthN/AuthZ em federação***

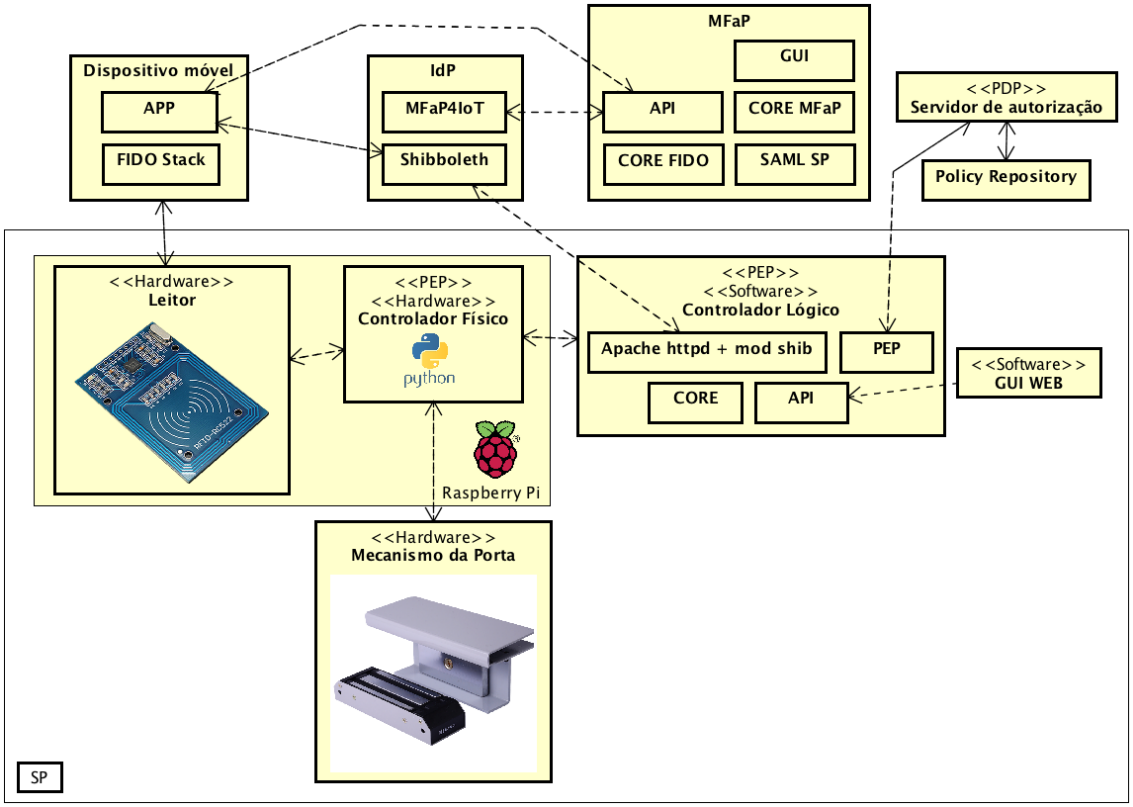

Figura 4. Diagrama de Componentes Concretos da solução SP.

requisição e resposta da operação de autenticação, e verificação de responsta de autenticação FIDO. Desse modo, o fluxo de autenticação consistirá em um módulo cliente para a API REST do MFaP, que será invocado durante o fluxo de autenticação de usuário.

Para a autenticação no IdP será desenvolvido um novo fluxo de autenticação em *SWF* chamado *mfa-fido-flow* que será tratado dentro do contexto de configuração MFA. Este fluxo será acionado através da lógica configurada no arquivo *mfa-auth-config.xml*. Esse arquivo está localizado no diretório *authn* do *IdP*, e deve ser editado para que o para que se delegue o controle do fluxo de login para o *MFA*. As transições entre fluxos de autenticação existentes e o momento em que cada fluxo chega ao fim são controlados por mapas de regras definidos em um *bean* chamado `shibboleth.authn.MFA.TransitionMap`.

5. Conclusão

Neste trabalho apresentamos nossa proposta de arquitetura para integrar IoT, controle de acesso e autenticação federada. Este trabalho é parte do GT-AMPTo (Autenticação Multi-fator para Todos) da RNP, onde a nossa responsabilidade é em implementar os mecanismos de autorização. A arquitetura apresentada está em fase de detalhamento, com a definição de um primeiro protótipo em bancada envolvendo Raspberry Pi 2, módulo Leitor RFID-RC522, trava Lockitron Mechanical Assembly, Arduino Nano, e Driver para Motor de passo DRV8834.

Como próximo passo, projetamos acrescentar suporte a autenticação federada ao protótipo, integrando-o com a federação CAFe. Em seguida, acrescentaremos suporte a

autorização com políticas RBAC, e autenticação em dois fatores com FIDO.

Agradecimentos

Este trabalho foi desenvolvido no contexto do Grupo de Trabalho Autenticação Multifator para Todos (GT-AMPTo) financiado pela RNP.

Referências

Cardenas, A. A., Amin, S., and Sastry, S. (2008). Secure control: Towards survivable cyber-physical systems. In *2008 The 28th International Conference on Distributed Computing Systems Workshops*, pages 495–500.

Chadwick, D. W. (2009). Federated identity management. In *Foundations of Security Analysis and Design V*, pages 96–120.

Domenech, M. C. (2015). Uma infraestrutura de autenticação e de autorização para a web das coisas. Master's thesis, Universidade do Vale do Itajaí.

Domenech, M. C., Boukerche, A., and Wangham, M. S. (2016). An authentication and authorization infrastructure for the web of things. In *Proceedings of the 12th ACM Symposium on QoS and Security for Wireless and Mobile Networks*, Q2SWinet '16, pages 39–46, New York, NY, USA. ACM.

Evans-Pughe, C. (2005). The connected car. *IEE Review*, 51(1):42–46.

Fremantle, P., Aziz, B., Kopecký, J., and Scott, P. (2014). Federated identity and access management for the internet of things. In *Proceedings of the 2014 International Workshop on Secure Internet of Things*, SIOT '14, pages 10–17, Washington, DC, USA. IEEE Computer Society.

Hardt, D. (2012). The oauth 2.0 authorization framework. RFC 6749, RFC Editor. `http://www.rfc-editor.org/rfc/rfc6749.txt`.

Hu, V. C., Ferraiolo, D., Kuhn, R., Friedman, A. R., Lang, A. J., Cogdell, M. M., Schnitzer, A., Sandlin, K., Miller, R., Scarfone, K., Hu, V. C., Ferraiolo, D., Kuhn, R., Friedman, A. R., Lang, A. J., Cogdell, M. M., Schnitzer, A., Sandlin, K., Miller, R., Scarfone, K., and Cybersecurity, S. (2014a). Guide to attribute based access control (abac) definition and considerations (draft). *NIST Special Publication*.

Hu, V. C., Ferraiolo, D., Kuhn, R., Schnitzer, A., Sandlin, K., and Scarfone, K. (2014b). SP 800-162. Guide to Attribute Based Access Control (ABAC) Definitions and Considerations. Technical report, National Institute of Standards and Technology, McLean and Clifton, VA, United States.

Islam, S. M. R., Kwak, D., Kabir, M. H., Hossain, M., and Kwak, K. S. (2015). The internet of things for health care: A comprehensive survey. *IEEE Access*, 3:678–708.

Lewis, F. L. (2004). Wireless sensor networks. *IEE Review*, pages 11–46.

Liu, J., Xiao, Y., and Chen, C. L. P. (2012). Authentication and access control in the internet of things. In *ICDCS Workshops*, pages 588–592. IEEE Computer Society.

Machani, S., Philpott, R., Srinivas, S., Kemp, J., and Hodges, J. (2014). Fido uaf architectural overview. *FIDO Alliance, December*.

Minerva, R., Biru, A., and Rotondi, D. (2015). Towards a definition of the internet of things (iot). Technical report, IEEE.

Ndibanje, B., Lee, H.-J., and Lee, S.-G. (2014). Security analysis and improvements of authentication and access control in the internet of things. *Sensors*, 14(8):14786–14805.

OASIS (2010). extensible access control markup language (xacml) version 3.0. http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-cs-01-en.pdf.

Refeds (2017). Refeds mfa profile. https://refeds.org/profile/mfa.

Sandhu, R. S., Coyne, E. J., Feinstein, H. L., and Youman, C. E. (1996). Role-based access control models. *Computer*, 29(2):38–47.