Identity of Things Primer

/ ForgeRock Identity Platform 6.5

Latest update: 6.5.2

ForgeRock AS.
201 Mission St., Suite 2900
San Francisco, CA 94105, USA
+1 415-599-1100 (US)
www.forgerock.com

Abstract

Identity of Things (IoT) use cases incorporating ForgeRock Identity Platform™.

Table of Contents

Preface

This guide describes in general terms how you can use ForgeRock components today in your Identity of Things (IoT) ecosystem.

The following table lists typical IoT use cases described in this guide, and indicates which ForgeRock Identity Platform components you can use in each use case. This guide also provides links to additional documentation related to each use case.

ForgeRock Identity Platform Components You Can Use in IoT Today

IoT Use Case	Identity Gateway	Identity Management	Access Management	Directory Services	Identity Message Broker
Creating and storing a smart device identity		✓		✓	
Authenticating a smart device to a cloud service	✓		✓	✓	
Authorizing a smart device to access cloud APIs			✓	✓	
Self-Registering an IoT user identity		✓		✓	
Customizing the IoT user experience			✓	✓	
Revoking an association between device and user identities		✓	✓	✓	
Protecting IoT resources	✓		✓	✓	
Securing data at the edge			✓	✓	✓

This guide includes general statements of functionality for the following software versions:

- ForgeRock Access Management 6.5, with web policy agents 5.5, and Java EE policy agents 5.5
- ForgeRock Identity Management 6.5
- ForgeRock Directory Services 6.5
- ForgeRock Identity Gateway 6.5
- ForgeRock Identity Message Broker 5.5

This document is not meant to serve as a statement of functional specifications. Software functionality may evolve in incompatible ways in major and minor releases, and occasionally in

maintenance (patch) releases. Release notes for a ForgeRock product mentioned in ths guide may cover many incompatible changes. If you see an incompatible change for a stable interface that is not mentioned in the ForgeRock product release notes, please report an issue with the product documentation for that release. See "*Getting Support*".

Chapter 1

ForgeRock's Role in the Identity of Things

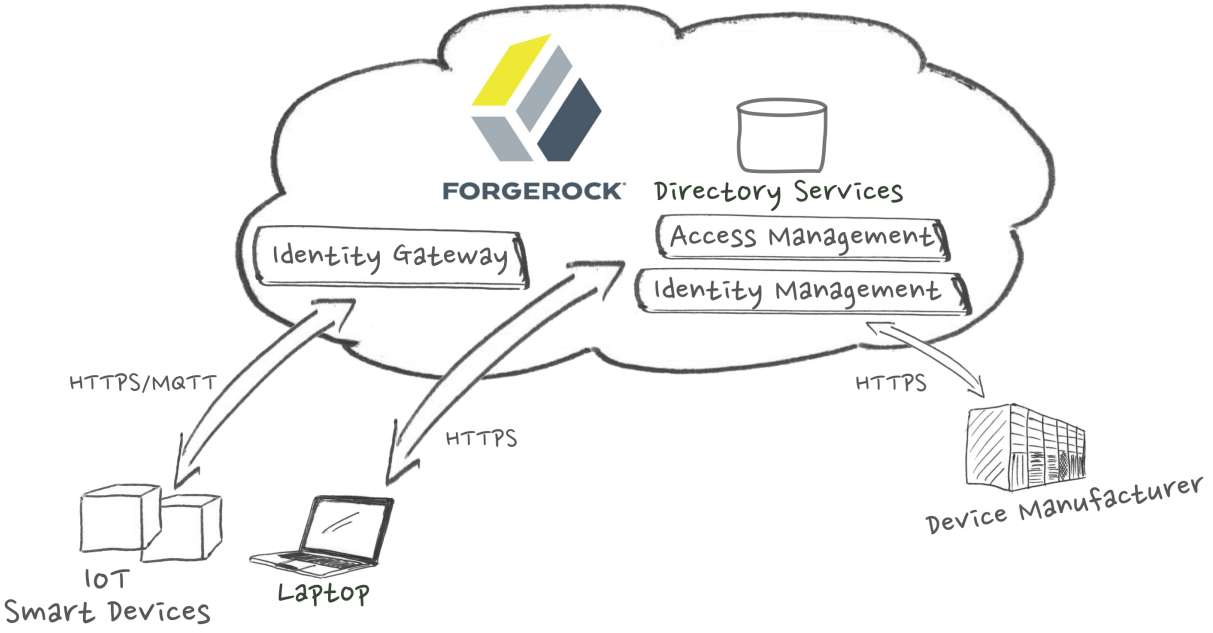
This chapter explains basic IoT concepts and terminology, and describes how ForgeRock Identity Platform fits into IoT.

If you've thought about when or how to move your company into IoT, then you already know about the amazing products innovation has made possible in recent years. Wristwatches that can make phone calls, self-regulating home appliances, self-driving cars, life-saving medical monitors—these become reality when devices can connect to the internet.

The complexity of your IoT ecosystem is determined first by the types of entities—people, devices, and resources—in your network topology. Then you'll have to consider the relationships between entities and how data is exchanged among them.

The ForgeRock Identity Platform ensures that only authenticated, authorized people and devices can access and exchange data in your enterprise. The use cases in this guide showcase ForgeRock functionality you can use today to manage identities in your IoT ecosystem.

ForgeRock's Role in IoT



1.1. Devices Have Identities Too!

The ForgeRock Identity Platform can securely manage millions of digital identities for your comany. We often think of an identity as a representation of an individual—a person. In a traditional enterprise, an identity is usually assigned to an application or web service user. In an IoT ecosystem, identity management is just as important for devices as it is for users.

Here is common example. A consumer establishes an account with a company that provides internet services. The consumer purchases audio speakers for each room in the household. Then the consumer connects the audio speakers to the internet. Now each household member can stream audio services for music, television, and other home entertainment. Does this seem familiar? The consumer has just set up a household IoT ecosystem.

In this example, each member of the household has an identity, and each audio speaker has an identity. The parents in the household can stream audio unconditionally in every room in the home. But the parents configure the internet service to allow their children to use the audio speakers only under specified conditions. For example, the parents can allow or deny a child access to streaming audio content based upon the time of day, or based upon the room in which the audio speakers are

located. Each audio speaker must have a unique identifier, and each audio speaker must be must be associated with each member of the household.

In the business world, the relationships among service users and internet devices are more complex. But the concepts of customer identity and access management (CIAM) are the same. In the ForgeRock Identity Platform, a user is represented by a unique username. The user's e-mail and phone number are stored in the Directory Service as user attributes. Similarly, an internet device can be represented by a unique serial number. Device details such as media access control (MAC) address and physical location address can be stored as device attributes. In this way, you can think of users and devices as peer objects in the data repository.

1.2. ForgeRock Is a Natural Fit for IoT

ForgeRock's identity-centric approach secures the relationships between service providers and users, between users and devices, and between devices and other devices. This lets you build new, secure, personalized user experiences for your customers. ForgeRock Identity Platform provides the building blocks you need to get started with IoT:

- **Flexible Identity Stores**

  ForgeRock Directory Services provide full and fractional data replication for users, devices, configuration, and schema. You can use REST and LDAPv3 APIs to implement your identity storage solutions.

- **Appropriate Access Protocols for CIAM**

  ForgeRock Access Management uses a complex authorization engine that can be accessed over REST natively from applications, brokers or devices. Or from agents for platforms such as Apache, NGINX or Tomcat.

- **Identity Management**

  ForgeRock Identity Management provides the means to store device related data, the flexible schema of IDM, and a vast and simple connector framework to integrate different stores.

- **Identity Gateway**

  ForgeRock Identity Gateway enforces security and access control in conjunction with ForgeRock Access Management modules. IG helps you seamlessly integrate web applications, APIs, and microservices with the platform.

- **Unified RESTful APIs**

  ForgeRock Identity Platform delivers one common REST API framework across the entire platform to provide a single, common method to invoke any of our identity services. Decoupled from the UI framework, the unified REST API makes it simple to connect the Platform to any digital thing—from mobile devices and cars to set-top boxes and machines.

1.3. IoT FAQs

1.3.1. What Is An IoT Ecosystem?

An IoT ecosystem is any of group of network-attached devices that interact with other devices, services, applications, and people. An IoT ecosystem can include smart devices, constrained devices, users, and any other resources that send and receive data to each other. An IoT ecosystem can be as small and simple as a household filled with sound and light systems, entertainment devices, and other appliances that are remotely controlled over the internet by household members. Or an IoT ecosystem can be as large and complex as a health care organization with numerous facilities, hundreds of care providers, and a myriad of medical devices and other healthcare resources.

1.3.2. What Are Smart Devices?

Smart devices are electronic tools or gadgets that connect to digital networks. People use these gadgets to access content and communication services directly over HTTP. Smart devices are often seen in the consumer identity space where examples include sports wearable technology, health monitoring devices, connected vehicles and media systems.

The IEEE Computer Society defines a smart device as an electronic tool that has some of the following capabilities: communications; identity; memory and status tracking; sensing and learning. See *Smart Devices and Controllers* .

1.3.3. What Are Constrained Devices?

Like smart devices, *constrained devices* are also electronic tools. In the operational technology landscape, constrained devices are often used as sensors, logic controllers, and production line monitors.

1.3.4. How Do Smart Devices and Constrained Devices Compare?

Both smart devices and constrained devices can operate fully in both the consumer and operational technology fields. Both types of devices are capable of collecting, aggregating, and sending data. Both types of devices can interact with with APIs and web services. But constrained devices are typically much smaller than smart devices, and have lower processing capabilities.

The following table compares smart device and constrained devices.

Smart Devices vs. Constrained Devices

	Smart Device	Constrained Device
CPU Power	Mobile phone equivalent	Thermostat equivalent

	Smart Device	Constrained Device
Electric Power	Household electricity or rechargeable long life battery; always on	None-replaceable, non-recharable battery only; limited on time
Memory	Hundreds of MB	Less than 256 kB
Storage	Hundreds of MB to multiple GB	None to hundreds of kB
Networking	HTTP or HTTPS	CoAP, MQTT, and other non-HTTP protocols
User Interface	None to basic	None

1.3.5. What Is MQTT?

The OASIS Committee specifies Message Queue Telemetry Transport (MQTT) as a simple and lightweight messaging protocol designed for constrained devices and low-bandwidth, high-latency or unreliable networks. MQTT minimizes network bandwidth and device resource requirements, while also attempting to ensure reliability and some degree of assurance of delivery. MQTT is ideal for the emerging machine-to-machine (M2M) or internet of things (IoT) world of connected devices. MQTT is also useful for mobile applications where bandwidth and battery power are at a premium. For more information about MQTT, see the *MQTT Version 3.1.1 - OASIS Standard*.

1.3.6. What Is CoAP?

The coap.technology community defines Constrained Application Protocol (CoAP) as a specialized web transfer protocol used with constrained nodes and constrained networks in IoT. CoAp is designed for machine-to-machine (M2M) applications such as smart energy and building automation. CoAp is based on on RFC 7252. For more information about CoAP, see the *coap.technology website*.

1.3.7. What Is an IoT Broker?

A device or piece of software that acts as a transport intermediary. Many lower-level IoT devices do not communicate over HTTP(S). They perform local interactions using protocols such as MQTT or CoAP. In these situations, a broker can be used to collect and aggregate the data, and to manage these lower-level IoT devices. An IoT broker has communication interfaces, and makes APIs available to clients and cloud services that can communicate over HTTPS.

1.3.8. What Is a Trusted Execution Environment?

A Trusted Execution Environment (TEE) is a highly secured area of a main processor. The TEE is an isolated execution environment, secured through the use of trusted applications and asset confidentiality measures. Access tokens used by IoT devices are often stored in a TEE.

Chapter 2

Using the Platform with Smart Devices

This chapter illustrates how you can use ForgeRock Identity Platform products in every phase of the smart device life cycle.

The following use cases are explained in this chapter:

- "Creating and Storing a Device Identity Using IDM and DS"
- "Creating an IoT User Identity Through Self-Registration"
- "Authenticating a Device to a Cloud Service Using a Digital Certificate"
- "Authorizing a Device to Access the Cloud API Using OAuth 2.0"
- "Customizing the IoT User Experience Through User-Managed Access"
- "Revoking An Association Between IoT User and Device Identities"

2.1. About Smart Device Use Cases in This Guide

Together, the use cases in this guide tell the story of a typical smart device life cycle. In each of the following use cases, the smart device is an audio speaker that can access streaming audio services for music, television, and other home entertainment. Embedded software and its ability to connect to the internet over HTTPS make this a smart speaker. The smart speaker has no graphical user interface of its own. But an end user can use a secondary computing device, such as a mobile phone or laptop computer, during setup and configuration.

ForgeRock Identity Platform links the end user's identity to the device identity, *personalizing* the device. Once the end user personalizes the smart speaker, the smart speaker can connect to its manufacturer cloud service of behalf of the end user. For example, the end user can push the smart speaker ON button to begin streaming music —without having to access a website or other user interface.

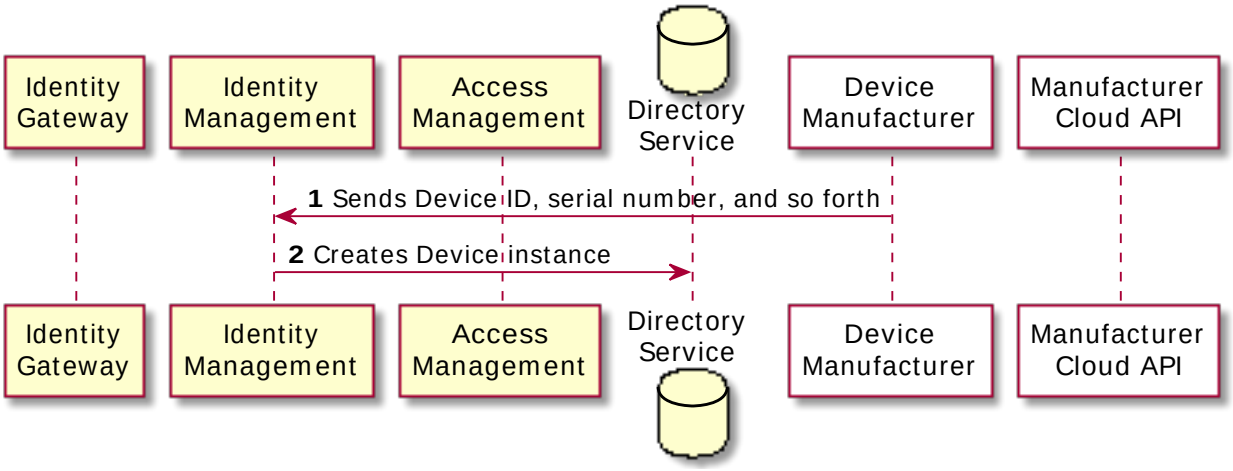
The end-user interactions in these use cases could all take place within one user session. It's likely that the end user would set up the smart speaker and be listening to streaming music within 15 minutes. But ForgeRock's role begins well before the end user purchases the device. Identity management starts when the smart speaker is being manufactured.

2.2. Creating and Storing a Device Identity Using IDM and DS

In this use case, the smart speaker manufacturer creates a unique identity for each smart speaker on the assembly line. Factory workers stamp each smart speaker with a unique serial number. System administrators use ForgeRock Identity Manager to create a device profile for each smart speaker, and to assign each smart speaker a unique media access control (MAC) address. The device profile data is stored in ForgeRock Directory Server.

Additionally, each smart speaker is equipped with "call home" software before it leaves the factory. This software lets the smart speaker contact the manufacturer cloud service directly—without a user interface—over HTTPS. The following figure illustrates the communication flow among ForgeRock Identity Platform components and the manufacturer.

Creating and Storing a Device Identity



You can use IDM to create device profiles and to provision an identity data store. For more information about provisioning, see the following topics in the Identity Management product documentation:

- Overview of Identity Management Modules
- Managing Users, Groups, Roles and Relationships
- Sample Deployments
- Identity Management Documentation Landing Page

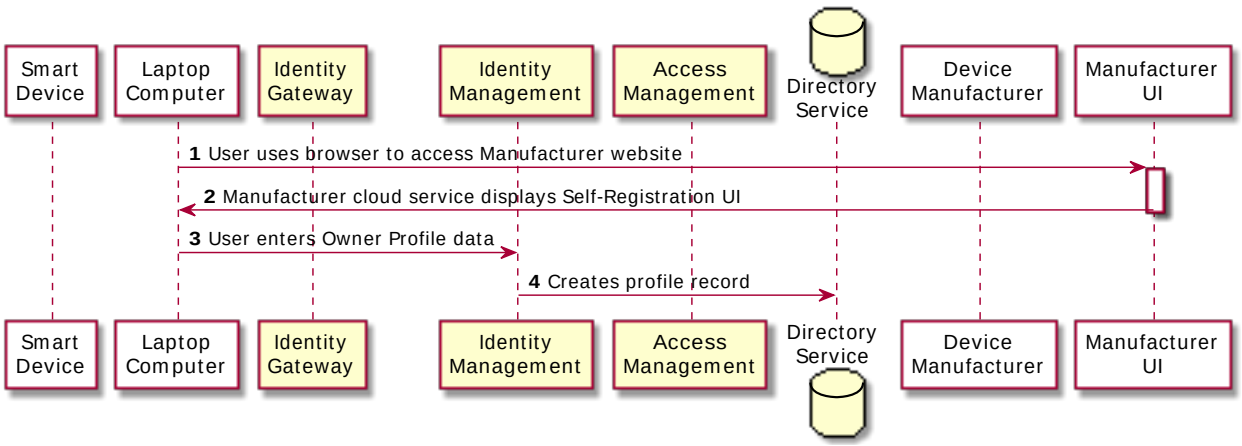
You can use DS to store and index device profiles. DS scales to millions of users and devices. For more information about directory data and storage, see the following topics in the Directory Services product documentation:

- Overview of Directory Services Modules
- Understanding Directory Services
- Managing the Repository
- Managing Data Replication
- Directory Services Documentation Landing Page

2.3. Creating an IoT User Identity Through Self-Registration

In this use case, after purchasing the smart speaker, the end user uses a laptop computer to access the smart speaker manufacturer website. The end user enters information such as username and email address, establishing a user identity in the manufacturer DS. IDM provides a default Self-Regisration UI enabling your customers to register themselves through your website.

Creating a User Identity Through Self-Registration



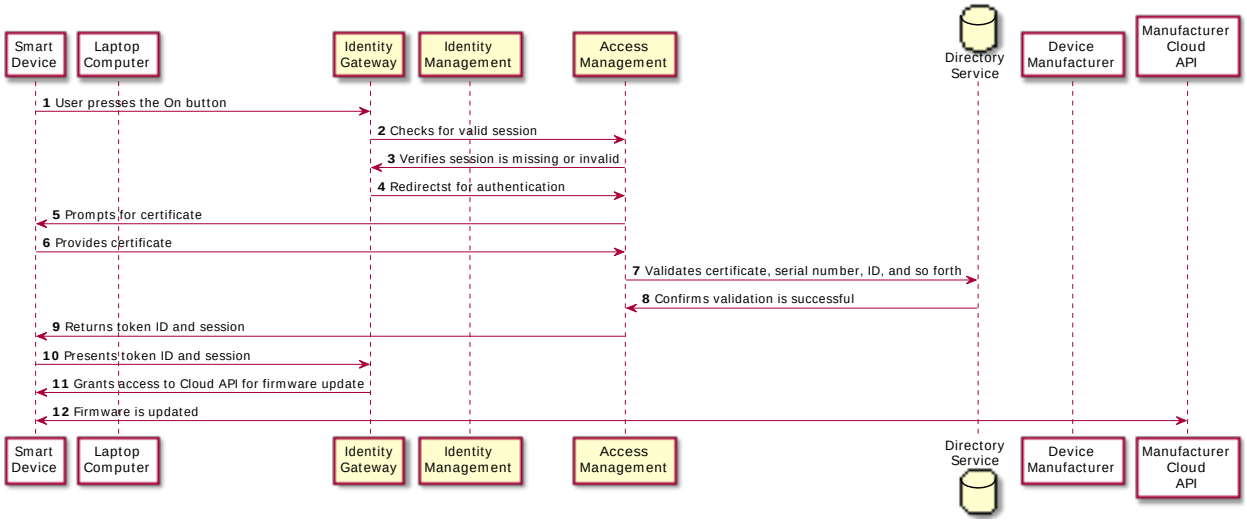
For more information about self-registration see the following topics in the Identity Management product documentation:

- *ForgeRock User Self-Service Guide*
- Configuring User Self-Service
- Authentication Module Properties

2.4. Authenticating a Device to a Cloud Service Using a Digital Certificate

In this use case, the end user pushes the smart speaker ON button. The smart speaker "calls home," attempting to communicate with its manufacturer. The communication is intercepted by AM, which requests the smart speaker credentials. The smart speaker presents the certificate provisioned at the factory, and AM issues a session ID and a token. Now authenticated, the smart speaker can begin its initial firmware download.

Smart Device Calls Home



For information about certificate-based authentication, see the following topics in the Access Management and Identity Gateway product documentation:

- Introducing Authentication
- Certificate Authentication Module
- Getting Login Credentials From Data Sources

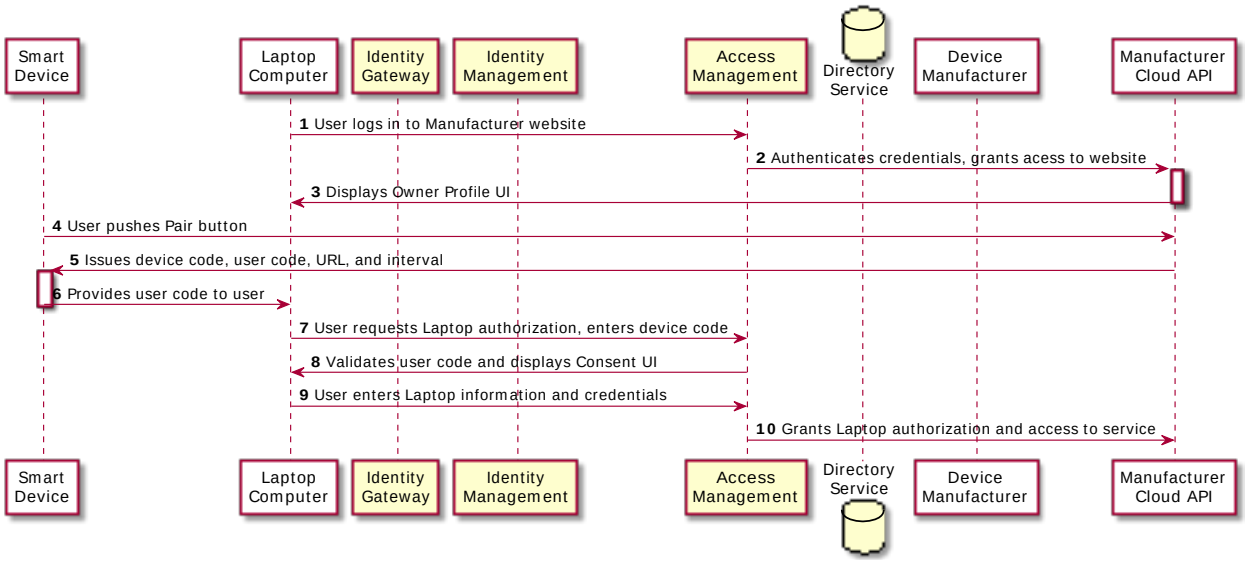
2.5. Authorizing a Device to Access the Cloud API Using OAuth 2.0

In this use case, the end user personalizes the smart speaker. OAuth 2.0 authorization links the end user identity to the smart speaker identity. Beyond merely establishing device ownership, once the

identities are linked, the smart speaker is authorized to communicate with the manufacturer cloud API on the end user's behalf. In this use case, the end user authorizes the smart speaker to connect to the manufacturer cloud API each time the ON button is pushed.

OAuth 2.0 authorization is designed for smart devices that have limited user interfaces. In this use case, the smart speaker has only a power ON button and a small, one-line display screen. The end user starts the OAuth 2.0 process by pushing the smart speaker ON button for the first time. The following figure illustrates the OAuth 2.0 device authorization process.

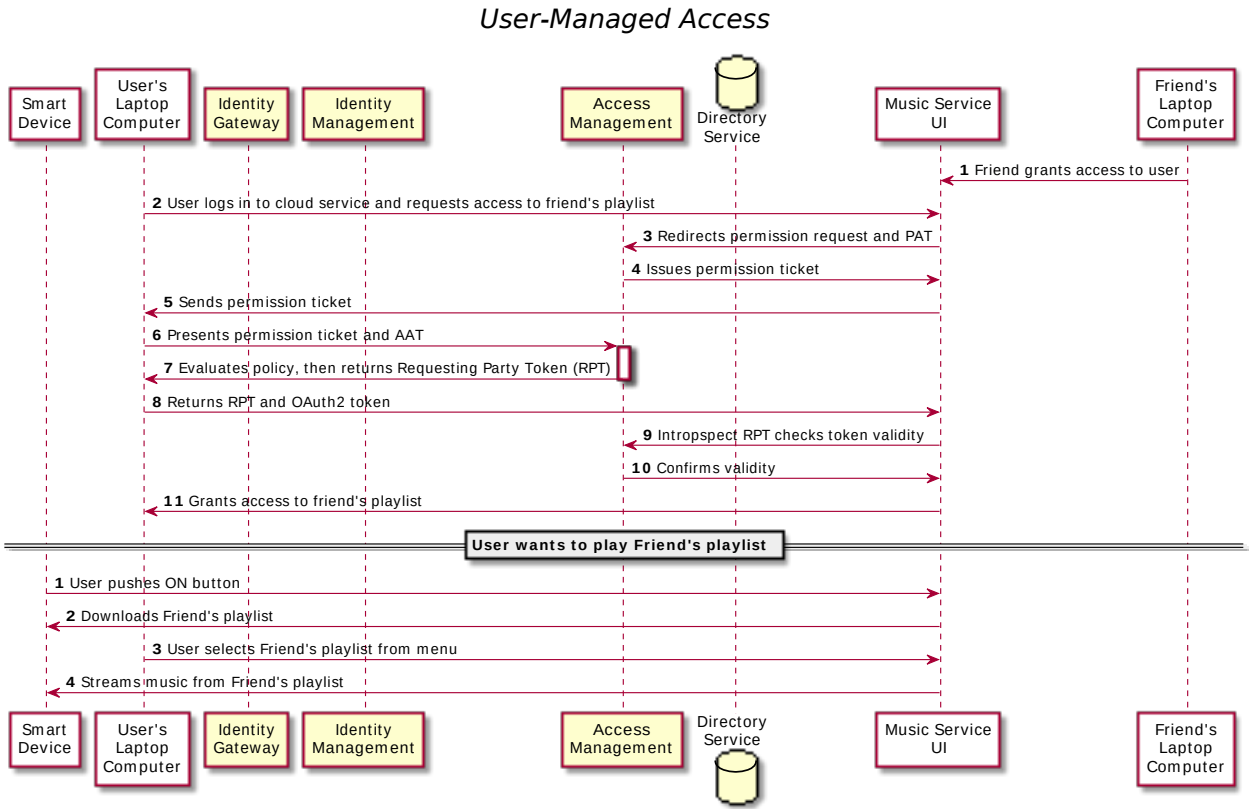
Linking User and Device Identities Using OAuth 2.0



- *ForgeRock OAuth 2.0 Guide*
- OAuth 2.0 Device Flow
- *OAuth 2.0 Device Flow for Browserless and Input Constrained Devices*

2.6. Customizing the IoT User Experience Through User-Managed Access

In this use case, the end user's friend has invited the end user to share a music playlist. The friend has created the playlist and stored it through a third-party music service. The smart speaker end user accesses the music service website using a laptop computer. AM provides a user-managed access (UMA) interface through which the end user can provide consent to third-party services. The following figure illustrates a typical UMA communication flow.



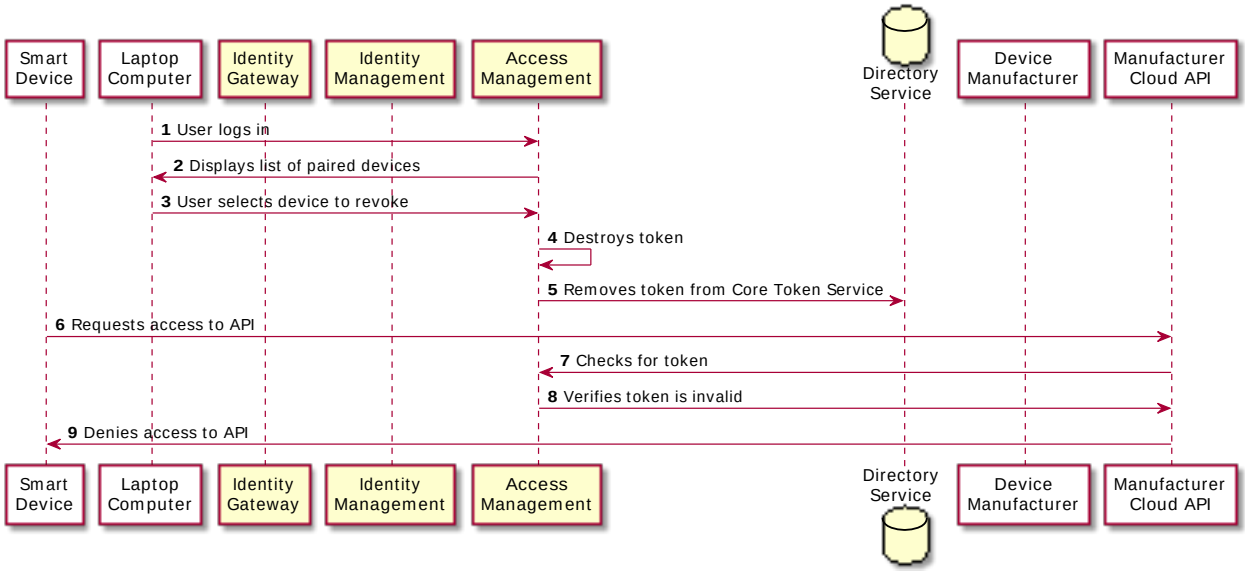
For more information about UMA, see the *ForgeRock User-Managed Access (UMA) 2.0 Guide*.

2.7. Revoking An Association Between IoT User and Device Identities

There will be times when you want to end the association between an end-user identity and the device identity. For example, the end user in our use case might lose the device, or sell the device to another person. In these situations, both the end user and device manufacturer will want to ensure that original device owner no longer has responsibility for any service charges associated with the device. The device manufacturer must also ensure that the original user identity—and any information contained in the original user profile—is not accessible to anyone who can access the device.

The following figure illustrates a typical communication flow for revoking the association between IoT user and smart device.

Smart Device Revocation



For more information about revoking authorization, see the following documentation:

- User Consent Management
- *OAuth 2.0 Token Revocation*

Chapter 3

Using the Platform to Protect IoT Resources

This chapter describes ways you can use ForgeRock Identity Platform to protect servers and APIs in an IoT ecosystem.

3.1. Protecting An IoT Resource Server

You can protect your IoT cloud service by configuring IG to act as an OAuth 2.0 resource server. When resources are protected with OAuth 2.0, IoT users can use their credentials with an OAuth 2.0-compliant identity provider such as ForgeRock Access Manager, Facebook, Google, and others to access the resources. This eliminates the need for the IoT user to set up an account with yet another third-party application.

For more information about protecting IoT resources, see the Identity Gateway product documentation:

- About IG As an OAuth 2.0 Resource Server
- Supporting UMA Resource Servers
- *ForgeRock Identity Gateway Guide*

3.2. Authenticating IoT-Issued Access Tokens Using OAuth 2.0 PoP

The ForgeRock Identity Platform is an early adopter of the OAuth 2.0 Proof of Possession (PoP) standard. PoP ensures that a *bearer token*, a token presented by a client, is presented by its rightful owner. A client could be a web browser accessing an application, or an IoT device connecting to a cloud service on behalf of an end user.

In the smart speaker use case, the manufacturer configured PoP as part of the OAuth 2.0 Device Flow that links the end user identity with the smart speaker identity. This protects against bearer token theft, and reduces the risk of man-in-the-middle attacks.

AM provides a transparent challenge/response-style interaction to prove the client is the intended owner of the access token. The client makes a normal request for an access token from the authorization service (AS), but the request includes an additional parameter. The parameter contains

some public key infrastructure (PKI) data the client has access to. The PKI data is typically the public key of an asymmetric key pair.

This key is then baked into the access token. If the access token is a JWT, the JWT contains this public key. The JWT is then signed by the authorization service. If using a stateful access token, the AS token introspection endpoint can relay the public key back to the resource server at lookup time.

For more information about Proof of Possession, see the following topics in the AM product documentation:

- Using OAuth 2.0 JSON Web Token Proof-of-Possession
- *ForgeRock User-Managed Access (UMA) 2.0 Guide*

3.3. Limiting Request Rates Using API Throttling Filters

IG provides a filter for regulating, or *throttling*, the number of requests an IoT user, device, or service can submit against a particular endpoint or set of endpoints. This helps prevent misuse of your cloud service or custom API. In the smart speaker use case, if the end user sends more requests than the number specified by the manufacturer, access to the server is denied.

For more information about using ThrottlingFilter, see Throttling Filters and Policies in the Identity Gateway product documentation.

Chapter 4

Using The Identity Message Broker At the Edge

The Identity Message Broker (IMB) secures data and communication that pass through the *edge*, or perimeter, of your physical IoT ecosystem. The edge might exist around a premises such an office building, a college campus, or a private residence. Or the edge could be defined by the confines of a car or a plane. IoT devices collect data and can trigger simple operations within the edge. But to leverage analytics and other complex computing tools, IoT devices communicate with cloud services that exist beyond the edge of the physical IoT ecosystem.

This chapter describes how you can use the IMB can to secure message transmission—in both directions—between IoT devices and cloud services.

4.1. About Identity Message Broker

The IMB is a publish-subscribe broker service that secures and hardens message transmission between IoT devices and cloud services. When an IoT device sends a request to a cloud service, following Access Management authentication, IMB does the following:

- Opens a connection with the IoT device.
- Validates an ID token from Access Management.
- Verifies that the authenticated IoT device is authorized to publish messages to a given topic.
- Verifies that the authenticated IoT device is authorized to receive messages on a given topic from cloud applications.

The IMB is ideal in IoT ecosystems where a lightweight publish-subscribe messaging service such as MQTT is already in place. For example, MQTT is often used with constrained IoT devices such as sensors that require low-level computing or communication capabilities. Sensors can collect and transmit data to cloud applications. The IMB secures data from such devices at the edge of the IoT ecosystem.

4.2. An Identity Message Broker Use Case

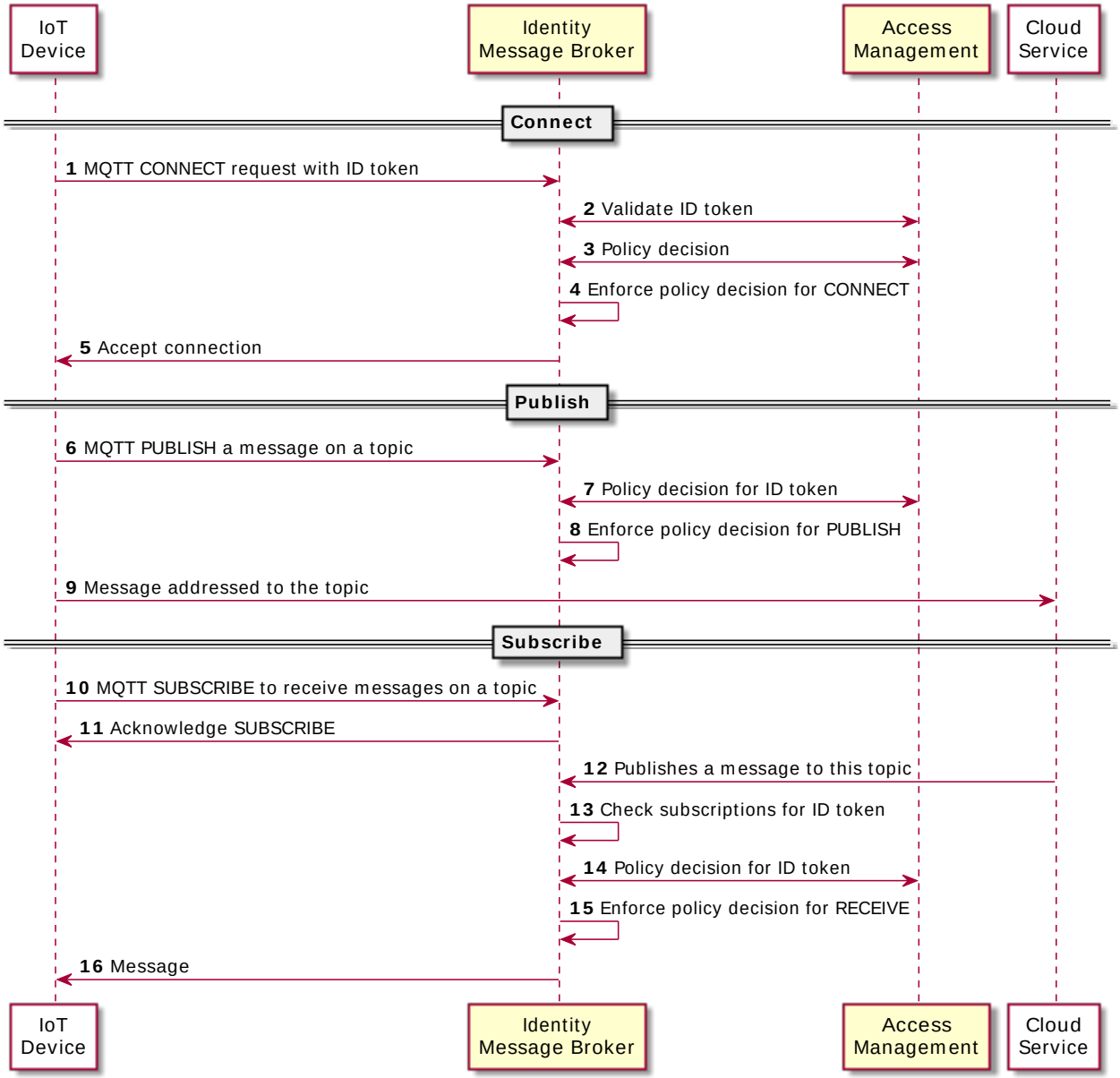
A conventional home air temperature thermostat is a good example of a simple sensor. The thermostat triggers home heating or cooling devices based on the temperature range the end user

has configured. When the it's too cold in the house, the thermostat sends a message to the air heating system to change its state to ON. All of this happens within the physical boundaries of the home.

A wifi-enabled thermostat, an IoT device, can do much more when it connects to a cloud service. For example, the wifi-enabled thermostat might store information about the times of day the home heating and cooling devices turn on and off. When the thermostat connects to the internet, it publishes an MQTT message containing time-of-day, temperature, and status information. Its manufacturer receives the data by subscribing to messages from the thermostat. The cloud service can analyze the data over time. The cloud service then publishes MQTT messages for the thermostat. The thermostat subscribes to messages from the cloud service to receive refined settings for improved efficiency and cost savings.

The following figure illustrates the data flow among a wifi-enabled home thermostat, the IMB, and a cloud service.

Message Broker Secures Data At the Edge



In this use case, the IMB secures all publish and subscribe messages sent between a single home thermostat and its manufacturer's cloud service. A factory production line, a fleet of rental cars, or other commercial settings can have hundreds of thousands of registered IoT devices—each with its own identity—in constant communication with a myriad of cloud services. The IMB will scale to meet those needs.

For detailed information about deploying and configuring Identity Message Broker, see the *Guide to the ForgeRock Identity Message Broker*.

Appendix A. Getting Support

For more information or resources about the ForgeRock Identity Platform and ForgeRock Support, see the following sections:

A.1. Accessing Documentation Online

ForgeRock publishes comprehensive documentation online:

- The ForgeRock Knowledge Base offers a large and increasing number of up-to-date, practical articles that help you deploy and manage ForgeRock software.

  While many articles are visible to community members, ForgeRock customers have access to much more, including advanced information for customers using ForgeRock software in a mission-critical capacity.

- ForgeRock product documentation, such as this document, aims to be technically accurate and complete with respect to the software documented. It is visible to everyone and covers all product features and examples of how to use them.

A.2. Using the ForgeRock.org Site

The ForgeRock.org site has links to source code for ForgeRock open source software, as well as links to the ForgeRock forums and technical blogs.

If you are a *ForgeRock customer*, raise a support ticket instead of using the forums. ForgeRock support professionals will get in touch to help you.

A.3. How to Report Problems or Provide Feedback

If you have questions regarding ForgeRock Identity Platform that are not answered by the documentation, you can ask questions on the Internet of Things forum at https://forum.forgerock.com/group/internet-of-things/forum/.

If you have a valid subscription with ForgeRock, report issues or reproducible bugs at https://backstage.forgerock.com.

When requesting help with a problem, include the following information:

- Description of the problem, including when the problem occurs and its impact on your operation
- Description of the environment, including the following information:
  - Software versions of supporting components
  - Software release version
  - Any patches or other software that might be affecting the problem
- Steps to reproduce the problem
- Any relevant access and error logs, stack traces, or core dumps

A.4. Getting Support and Contacting ForgeRock

ForgeRock provides support services, professional services, training through ForgeRock University, and partner services to assist you in setting up and maintaining your deployments. For a general overview of these services, see https://www.forgerock.com.

ForgeRock has staff members around the globe who support our international customers and partners. For details on ForgeRock's support offering, including support plans and service level agreements (SLAs), visit https://www.forgerock.com/support.