Internet of Things
2017/2018

LESHAN
(pictures from standards docs & software descriptions in presentations)

Johan Lukkien
Leila Rahman

John Carpenter, 1982
Guiding questions

- How does LESHAN support the development of systems using LWM2M?
Two interfaces

- Two interfaces:
  - an API to develop applications, typically as a library
  - the protocol between library instances
    - to be implemented by the library
- Libraries on both sides may be different

- The library can be limited to just a means for generating protocol messages

- Example:
  - CoAP protocol
  - Californium library

- A *framework* structures this development, even beyond a single device
Goal of the Leshan framework

- Let developers develop LWM2M servers, clients and IoT applications
  - provide Java libraries (CoAP, LWM2M) and APIs
  - support the programming of the client objects
  - implement the server functionality and make it available to applications
- Monitor an installed system
  - through a Web UI

LWM2M stack

- LWM2M Objects
- LWM2M Protocol
  - CoAP
  - DTLS
- UDP
- SMS
Run-time organization

- Services, registries and data stores supporting the LWM2M operations
- Web interface and web API for applications that use the devices
Example of querying list of endpoints and querying an endpoint’s resource by an application through Leshan server demo
Leshan programming model

- A programmer develops client and server as separate executables,
  - using Java
  - using an API delivered by Leshan

- The API represents basic operations, e.g.,
  - creating resources, objects
  - sending messages in the correct format

- The real behavior of both client and server are up to the programmer

Example Server (in Java)

```java
// Build LWM2M server
lwServer = new LeshanServerBuilder().build();

// Listen to registrations/deregistration
lwServer.getClientRegistry().addListener(new ClientRegistryListener() {
    @Override
    public void registered(Client client) {
        System.out.println("New registered client with endpoint: " + client.getEndpoint());
    }
    @Override
    public void updated(Client clientUpdated) {
        System.out.println("Registration updated");
    }
    @Override
    public void unregistered(Client client) {
        System.out.println("Registration deleted");
    }
});
// Start
lwServer.start();
System.out.println("Demo server started");
```
Life cycle of client and server

- Similar in development
- Deployment client: put code on device
  - through direct interaction
  - through a server
- Client execution and termination under control of server
- Application logic via web interface
(Limited) tooling

• The Eclipse development environment

• Test Servers, where clients can connect to
  – The lwm2m server at http://leshan.eclipse.org/
  – The bootstrap server at http://leshan.eclipse.org/bs/

• Demo projects and integration tests project

  Leshan-client-demo: a simple demo client.
  Leshan-server-demo: a lwm2m demo server with a web UI.
  Leshan-bsserver-demo: a bootstrap demo server with a web UI.
  Leshan-integration-tests: integration automatic tests.
At run-time

- The Java JRE is the runtime environment for client and server

- The Leshan LWM2M API provides the following functionality
  - Client initiated bootstrap
  - Registration/Deregistration
  - Create objects
  - Read, Write, Execute Resources
  - Resource Observation
  - TLV encoding/decoding

- The process model addresses
  - ‘how it runs’ (not worked out here)
  - concurrency in client
    - responsibility of programmer to use Java concurrency control to avoid blocking waits in client
  - concurrent operation of runtime services in server
At run-time: client

- Client side is mainly API on top of Californium
  - Californium: CoAP messaging
  - run-time support for common functions

- API supports the creation of objects and their logic and to initialize them
  - the objects are approached using the LWM2M messaging
Concluding

• Leshan is mainly a library with application support services inside a single executable
  – to ease the construction and transmission of LWM2M messages
  – to implement standard functionality of LWM2M
  – to increase the abstraction level of development
    • a layer say, that brings a predefined structure

• Provisioning for distributed application is through browser access and the client-server relations

• The respective components are started as executables on the respective devices
Guiding questions

• How does LESHAN support the development of systems using LWM2M?