www.secitc.eu

SCHEDULE

of

The 11th International Conference on Security for Information Technology and Communications SECITC 2018: www.secitc.eu

Informatics Security | CyberSecurity Master

Bucharest University of Economic Studies

www.secitc.eu

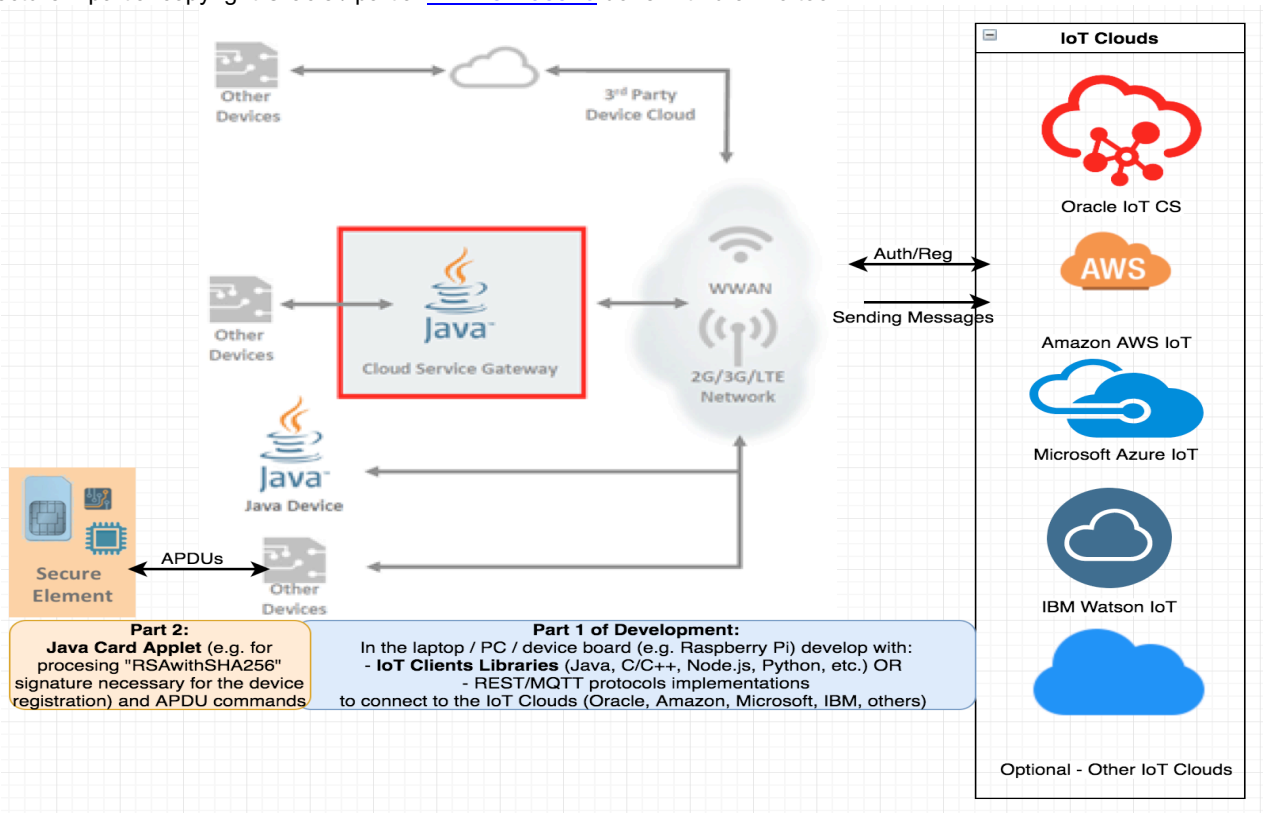
Hackathon - On Friday, 09 November 2018, Room 2001D, 18:00-19:30 / Zoom.us

MSc. and PhD. Students, who want to participate and start the Dev Hackthon on IoT & Security are invited. Single student or teams of three candidates are accepted. **More details on the conference website: www.secitc.eu - this activity is not connected with Springer LNCS.**

Deadline for the hack-days projects by sending the source code for the solution/challenge: on Monday, 12 Nov. 2018, 23:59 GMT to secitc@gmail.com | secitc@ase.ro (the submission must contain the source code, configuration files and compile/running info; also, the submission is flexible in terms of receiving the source code via public repositories GitHub, SVN, etc., although GitHub is preferred). The challenge for this Software Development Hackathon is to provide a solution into two parts for connecting a device to various IoT Clouds:

- Part 1 – connect a laptop or PC or Dev board (e.g. Raspberry Pi) to all this Internet of Things (IoT) Clouds by using directly the communications protocols (e.g. REST API – HTTP, MQTT, etc.) or the device client libraries (e.g. Java, C/C++, node.js – ECMAScript/JavaScript, Python, etc.):
  - Oracle IoT CS: https://cloud.oracle.com/iot (Get 30 days free: https://myservices.us.oraclecloud.com/mycloud/signup?language=en&sourceType=_ref_coc-asset-opcPAASIoT)
  - Amazon AWS IoT: https://aws.amazon.com/iot/
  - Microsoft Azure IoT: https://azure.microsoft.com/en-gb/overview/iot/ (Get free account: https://azure.microsoft.com/en-gb/free/)
  - IBM Watson IoT: https://www.ibm.com/internet-of-things / https://www.ibm.com/us-en/marketplace/internet-of-things-cloud
- Part 2 – Try to separate the cryptographic security execution from the host/device client library into Java Card simulator or real Java Card – card / token / element for creating an Java Card applet and host client side (for APDUs exchange) in order to externalize parts of the cryptographic secure algorithms used for signing the registration/authentication messages to the IoT Clouds.

Architecture – partial copyright Oracle / partial www.ism.ase.ro done with draw.io tool:



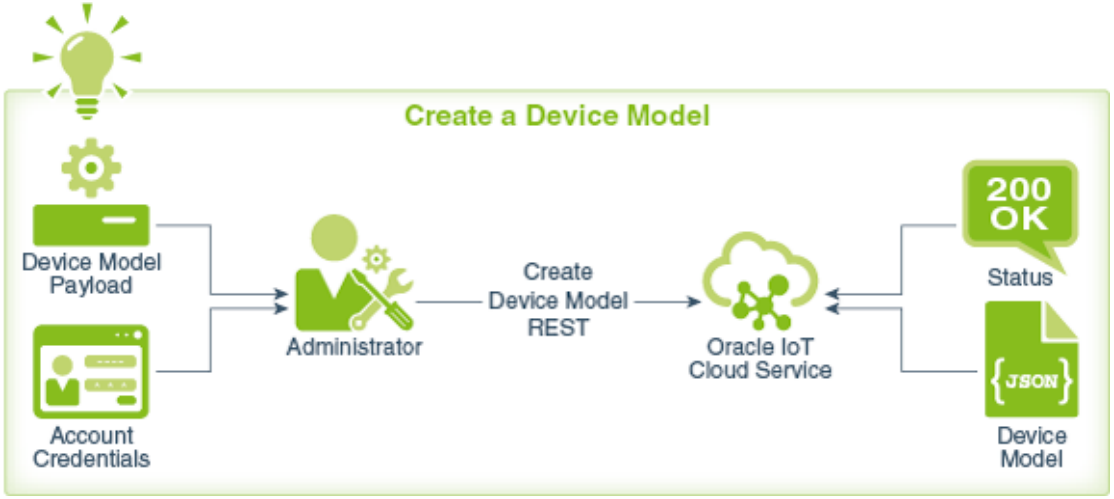
For the part 1 the following credentials are available for 20 days for Oracle IoT Cloud Services (for the other clouds the team should create trial instances and to document this action):
https://secitc5iotjls-secitc.gbcom-south-1.oraclecloud.com/ui/login/login.html
User: stud / Pass: TestTestP_01

Connect from the Java (node.js/Python/C/C++, etc) device client library using HTTP-REST or MQTT to the IoT Cloud

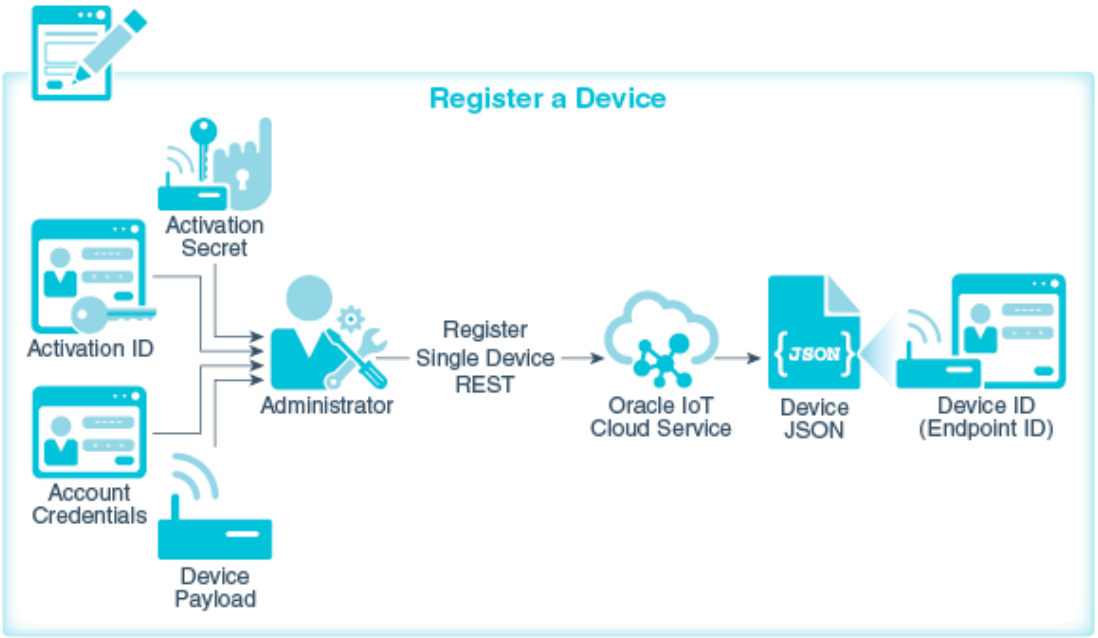
- e.g. starting point for Oracle IoT CS – copyright Oracle:
  https://www.oracle.com/webfolder/technetwork/tutorials/obe/cloud/iot/IoT%20Quick%20Start%20Java/iot_quickstart.html

  https://www.oracle.com/technetwork/indexes/downloads/iot-client-libraries-2705514.html
  https://docs.oracle.com/en/cloud/paas/iot-cloud/index.html
  https://docs.oracle.com/en/cloud/paas/iot-cloud/iotrq/QuickStart.html
  https://www.oracle.com/webfolder/technetwork/tutorials/obe/cloud/iot/quickstart/quickstart_java/qs_iot_java.html

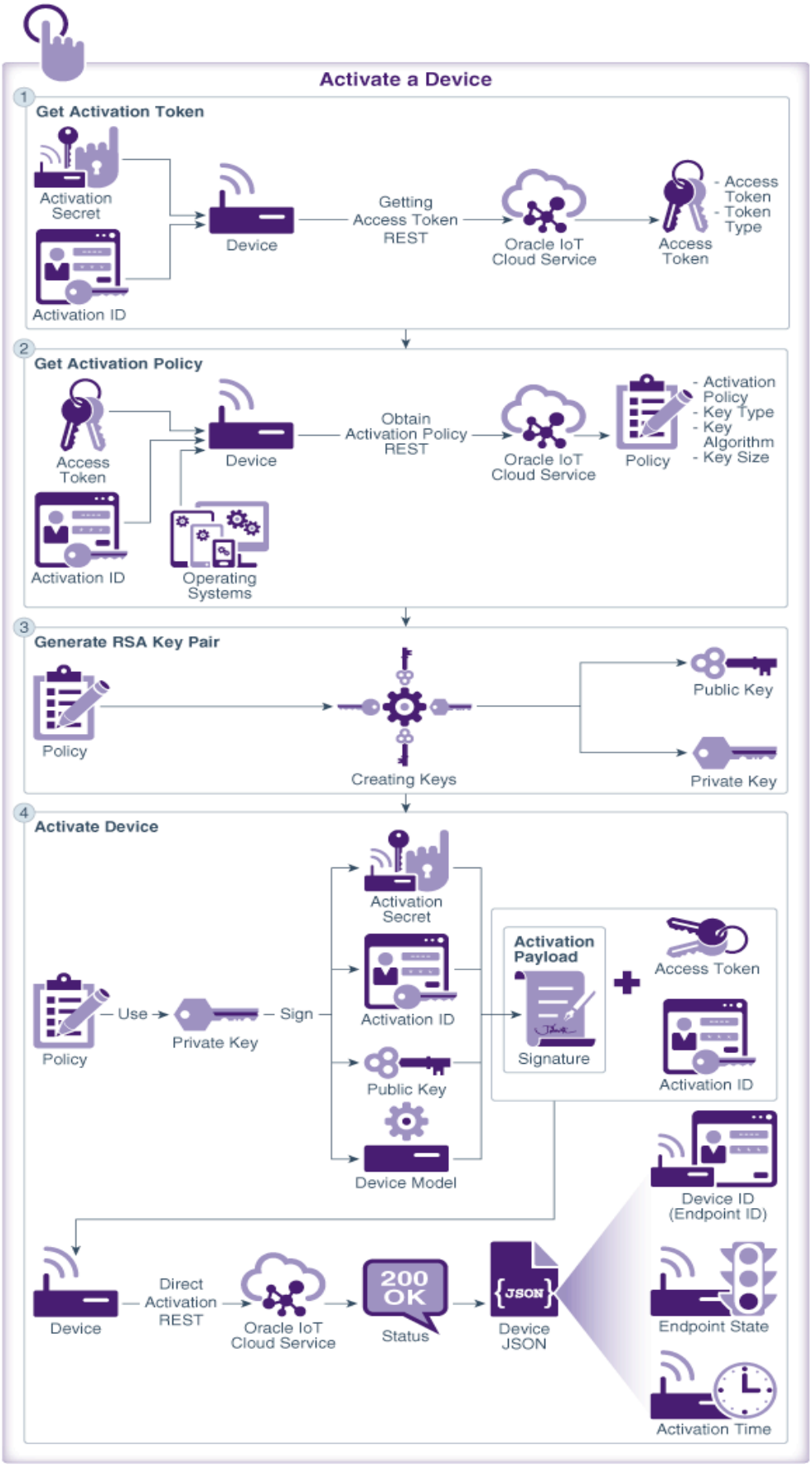
Step 1: Create a device model



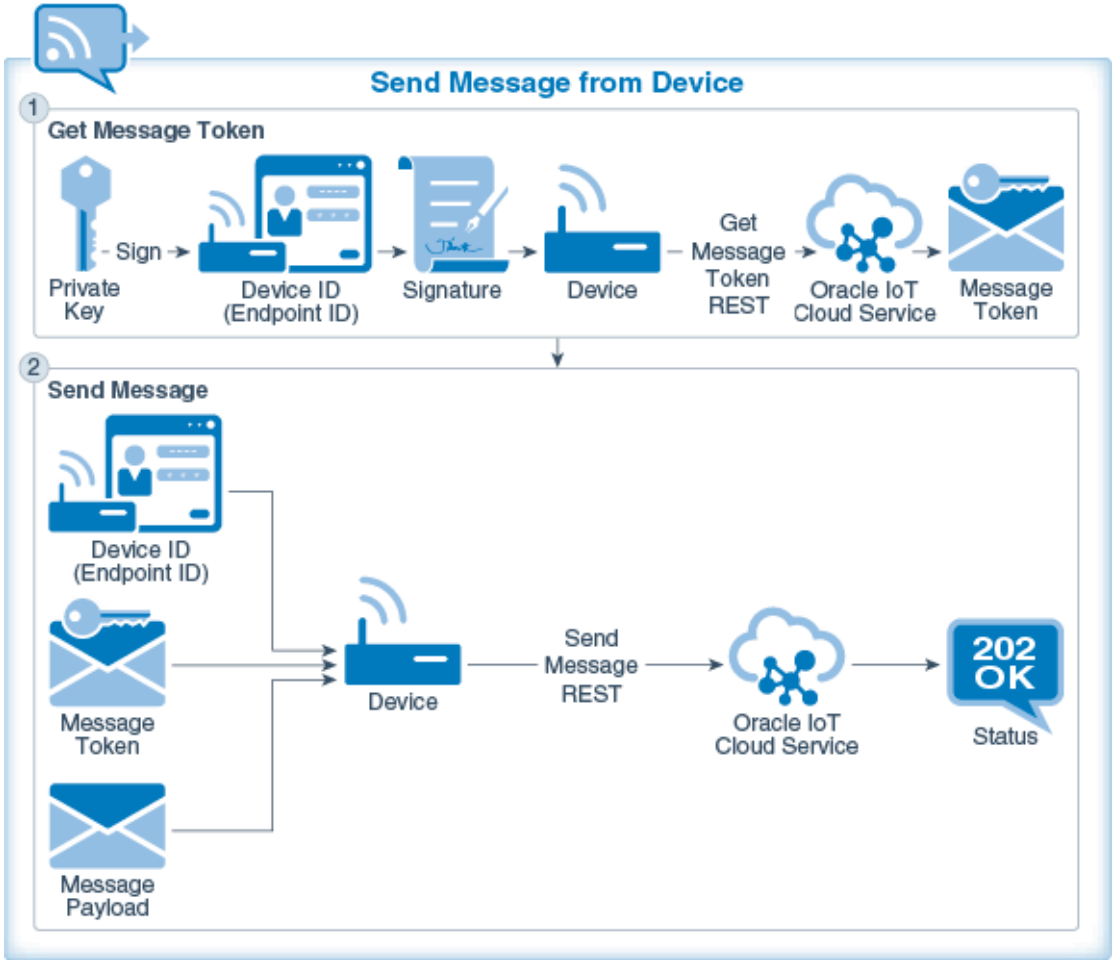
Step 2: Register a device



Step 3: Activate a device



Step 4: Send message from device



For the working start project please download from this link the client for the Oracle IoT CS (for other clouds including this one, part incomplete, the hack team should develop and validate):
https://drive.google.com/open?id=1V8WWh2Vvpq6jGZDZMFyUDH1PsgdCRao4

Part 2: Try to use a real card / secure element or the Oracle Java Card SDK/RI/Simulator: https://www.oracle.com/technetwork/java/embedded/javacard/downloads/javacard-sdk-2043229.html

in order to externalize and implement in the JavaCard applet various cryptographic functions which are used in the process of sending messages into the Cloud (e.g. oracle is using RSAwithSHA256 with asymmetric key on 2048 bits / other IoT Clods need for the device authentication, another cryptographic algorithm).