Motivation

- IoT devices provide constrained execution environments with limited processing, storage, and memory resources
  - offload parts of application business logic onto these devices
  - IoT gateways
- Dynamically adapt to inevitable changes such as new requirements or adjustments in regulations
  - purchase and sell these application components in an IoT application market
Challenge and Solution

- Significant differences in device capabilities
- Large number of devices in typical IoT systems

LEONORE

- service oriented infrastructure and toolset for provisioning application components on edge devices in large-scale IoT deployments
- installable application packages are fully prepared
- pull-based and push-based provisioning
Scenario

- Gateways participating in an IoT infrastructure are resource-constrained
- Large-scale deployments comprising thousands of gateways with a wide variety of different supported execution environments
- Requirements of these gateways change over time, which makes updates necessary
- In order to sustain operations all updates need to be efficient and fast
Approach
Application Packages

- Usually an application in the IoT domain consists of different application components and supporting files (artifact)
  - id
  - binary folder
  - control folder
  - path file
- Gateways only have to unpack the package and execute the provided installation instructions
IoT Gateway

> a container, hosting application packages
> a profiler, monitoring the current status of the gateway
> an agent, communicating with the provisioning framework
> a connectivity layer, supporting different communication protocols and provisioning strategies
  - a pull-based approach, queries the framework for provisioning tasks
  - a push-based approach, the framework pushes new updates to the gateway and the agent triggers the local provisioning
LEONORE – Provisioning Framework

- Repositories
  - artifact repository
  - IoT gateway repository
  - package repository
- Package Management
- Dependency Management
- Package Builder
- IoT Gateway Management and IoT Gateway Handler
**LEONORE – Provisioning Framework**

- **Provisioning Handler**
  - Chooses the suitable provisioning strategy according to the information provided by the IoT gateway management
  - Then the handler triggers the building of gateway-specific application packages by invoking the package builder
  - Once the builder creates the packages, the provisioning handler executes the provisioning strategy
LEONORE – Provisioning Framework

- **Balancer**
  - scale components by replicating them and therefore distributing the workload across multiple computing resources
  - components that should be scaleable are grouped together in so-called LEONORE nodes
Provisioning of Application Packages
Evaluation

- **Pull-base approach**
  - gateway’s agent pulls the provisioning framework for new tasks in a configurable interval
  - generates increased load on the framework

- **Push-base approach**
  - gateway’s agent only registers the gateway once at the framework and then remains idle until the framework pushes an update

- **2 applications**
  - SVM, 120KB
  - Java8, 12MB
Setup

- OpenStack
  - Docker image is used to virtualize and mimic the physical gateway
Scenario 1: 100 - 1000 IoT Gateways (pull)

(a) Evaluation Results for SVM

(b) Evaluation Results for SVM - Scatter

(c) Evaluation Results for JVM

(d) Evaluation Results with for JVM - Scatter
Scenario 1: 100 - 1000 IoT Gateways (push)

(a) Evaluation Results for SVM

(b) Evaluation Results for SVM - Scatter

(c) Evaluation Results for JVM

(d) Evaluation Results with for JVM - Scatter
Scenario 2: 500 - 4000 IoT Gateways

(a) Evaluation Results for SVM with **Pushing**

(b) Evaluation Results for SVM with Pushing - Scatter
Conclusion

- **LEONORE**
  - service oriented infrastructure and toolset for provisioning application components on edge devices in large-scale IoT deployments
  - installable application packages are fully prepared
  - pull-based and push-based provisioning