Le Web des Objets, de la théorie à la pratique

Implémentation de plusieurs cas d'utilisation, utilisation et création d'outils pour développer des applications RESTful du Web des Objets

UNIVERSITÉ DE FRIBOURG SUISSE
UNIVERSITÄT FREIBURG SCHWEIZ

Université de Fribourg, Département d'Informatique, Groupe Génie Logiciel

Software Engineering Group

Plan

1. Bases théoriques
2. Objectifs
3. Réalisation
4. Cas d'utilisation
5. Implémentation du rideau de fer
6. Conclusions

1. Bases théoriques 1/2

- Le Web des Objets et ses technologies
  - Protocole HTTP
  - Applications REST
  - Différents formats d'échange de données
    - XML
    - JSON
    - HTML
    - etc

1. Bases théoriques 2/2

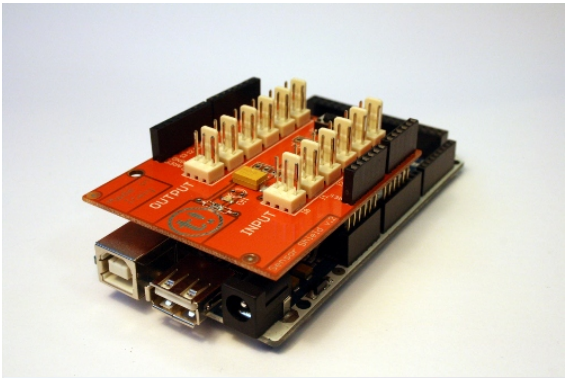
- Le xWoT méta-modèle
  - Développé par A. Ruppen
  - Un méta-modèle, définition
    - Trois niveau d'abstraction pour représenter une réalité
  - Le xWoT est composé de deux parties
    - La partie physique
    - La partie virtuelle

2. Objectifs

- Utiliser le xWot méta-modèle pour implémenter plusieurs objets connectés au Web
- Faciliter la création de futures applications du WoT

3. Réalisation de la partie physique

- Arduino
- Raspberry PI
- Meccano

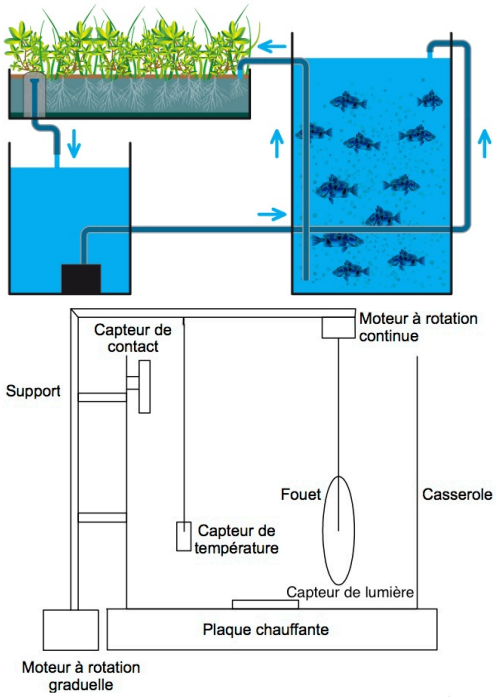


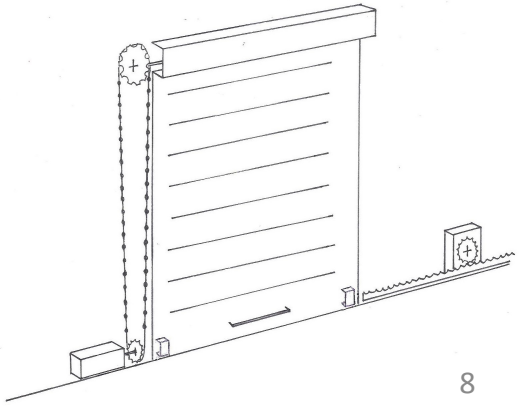
3. Réalisation de la partie virtuelle

- Plugins Eclipse et scripts Python pour la modélisation
- Framework Jersey pour le service Web
- Maven pour la gestion du projet et la génération automatique de code
- Ruby, Rubygems et Ruby on Rails pour le code côté client

4. Cas d'utilisation

- Système d'aquaponie
- Machine à caramels
- Rideau de fer

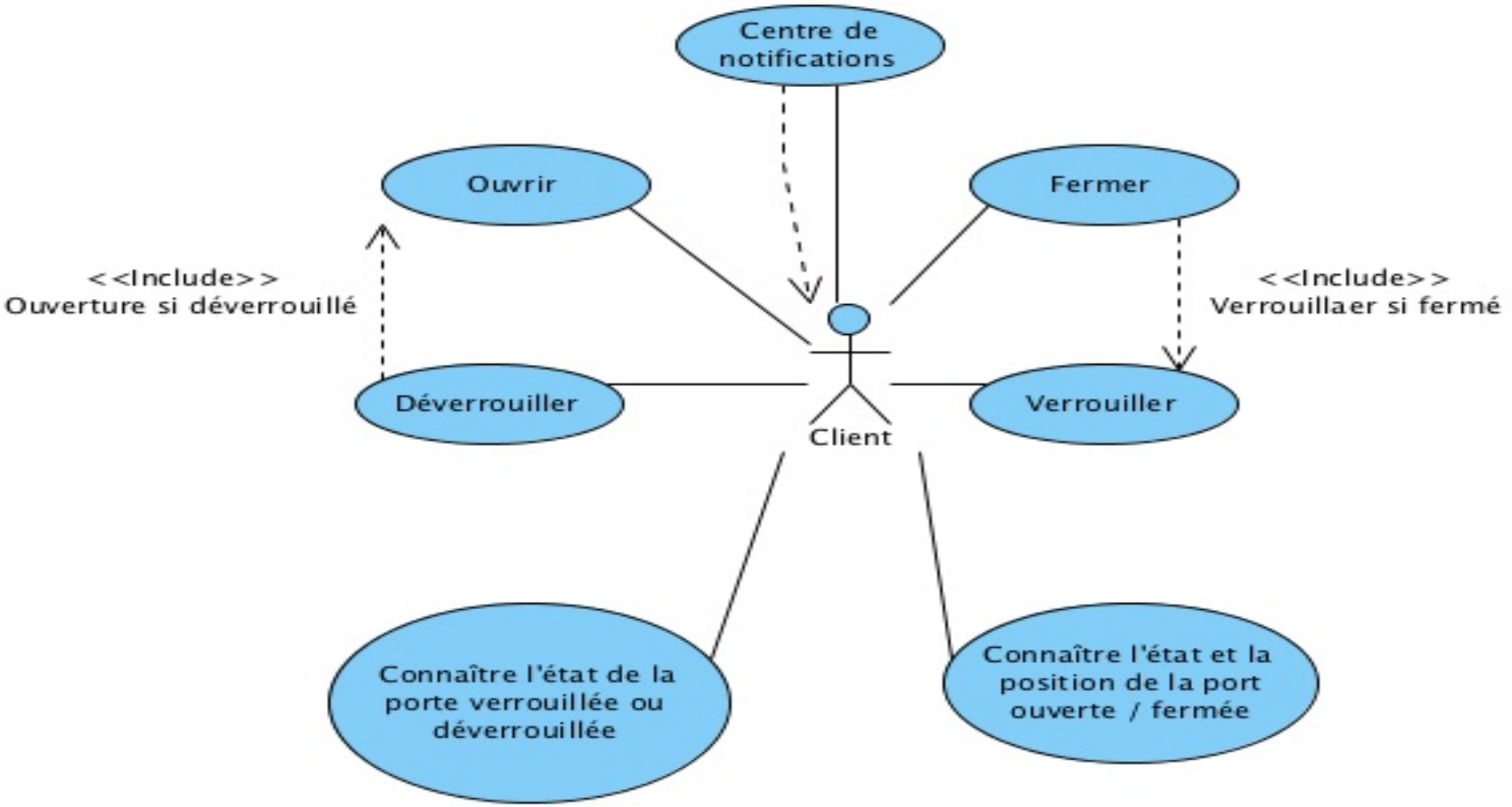




5. Rideau de fer – Modélisation

- Modélisation
  - Composé de deux sous-devices (ouverture/ fermeture et verrouillage/déverrouillage)
  - Forme deux ressources de contexte avec publishers
  - Données échangées modélisées grâce au langage XSD

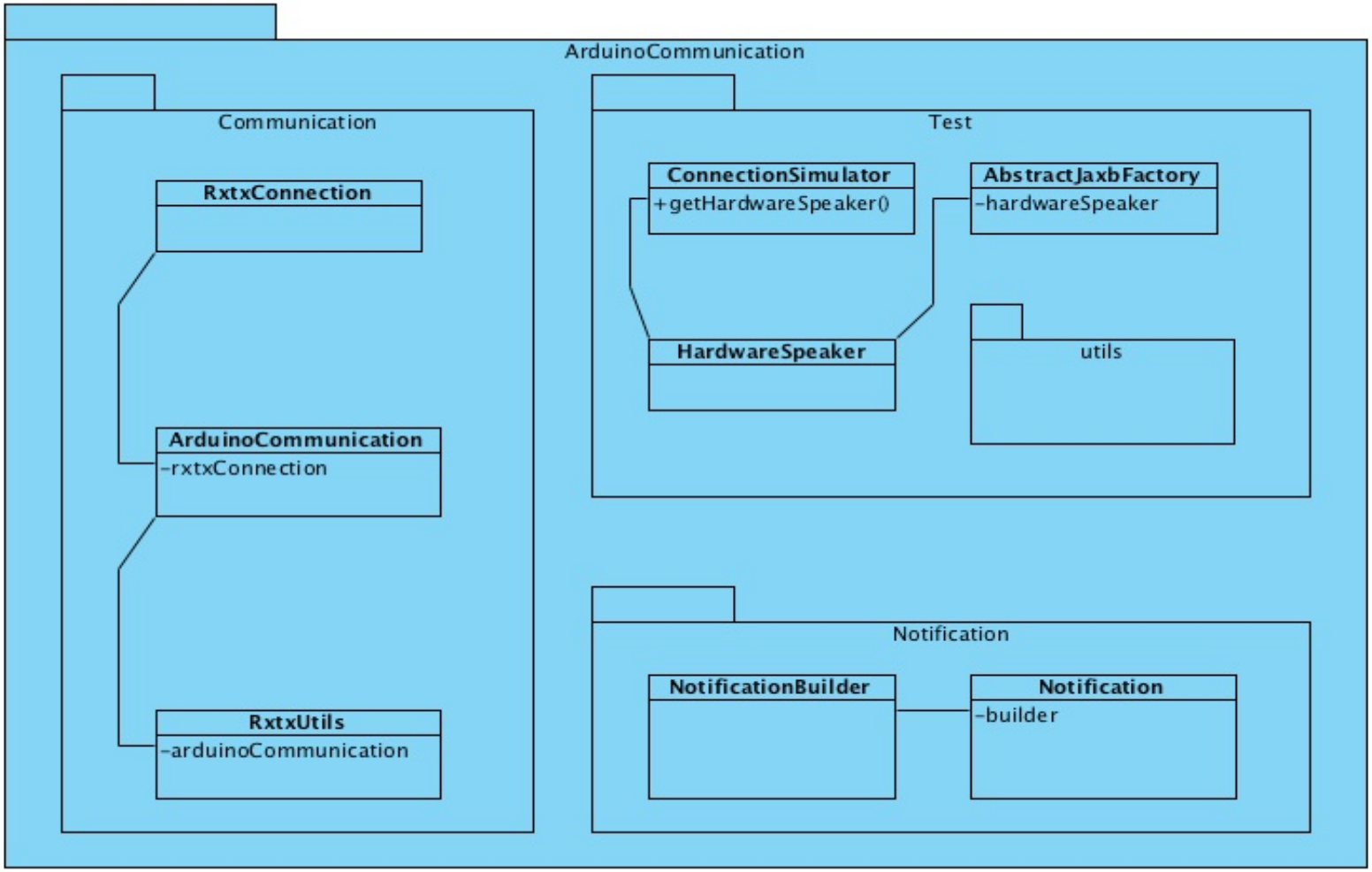
5. Diagramme d'utilisation



5. Rideau de fer – Arduino et service Web

- Programmer un Arduino
  - Contrôle de la porte
  - Communication avec le service Web
- Implémentation du service Web en trois parties:
  - Interagir avec l'Arduino en fonction de la requête du client
  - Ecriture de tests et donc simulation de l'Arduino
  - Gestion des notifications

5. Bibliothèque créée et utilisée pour l'implémentation du rideau de fer



5. Rideau de fer - Communication

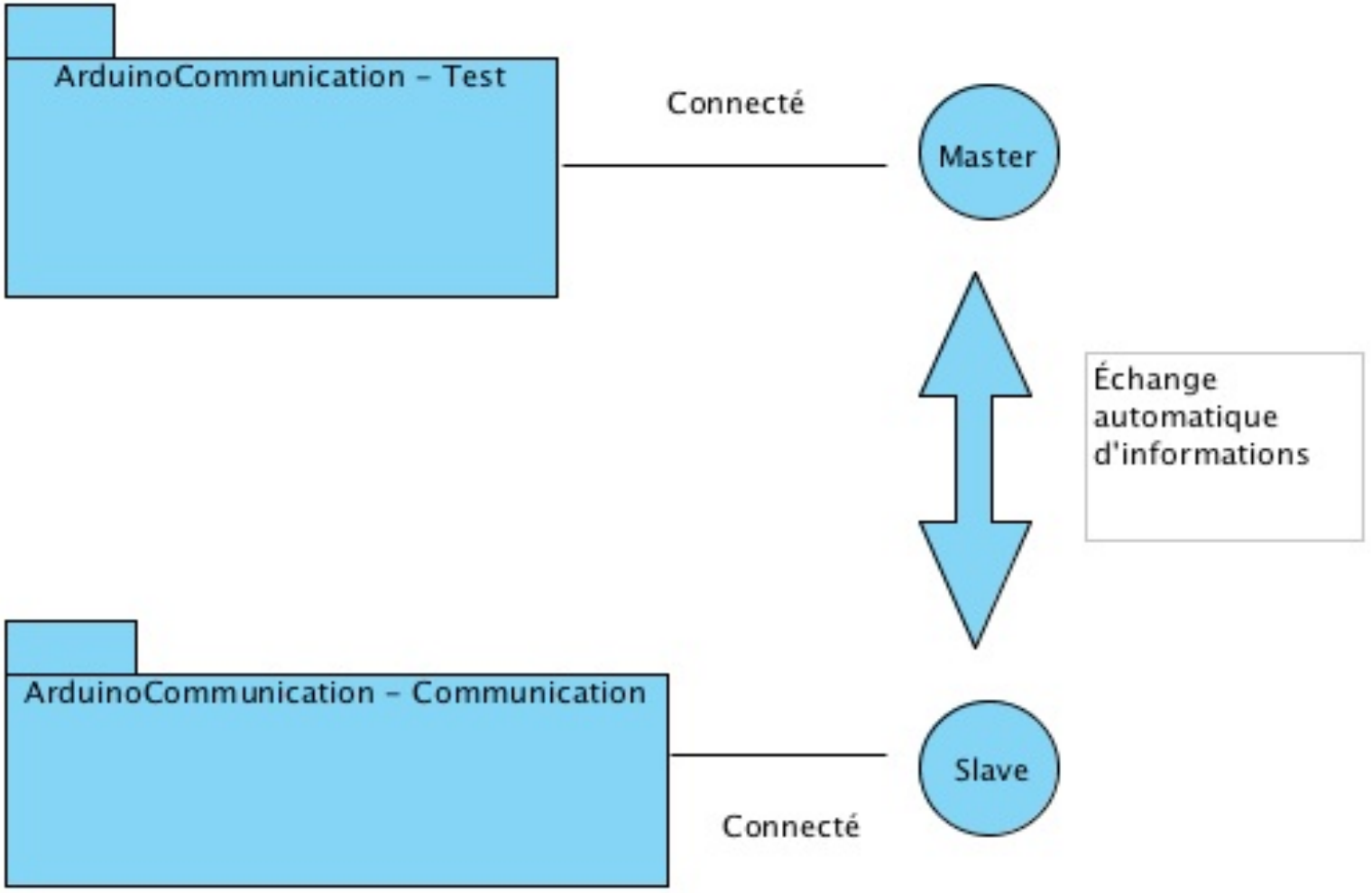
- Communication entre l'Arduino et le service Web
  - Utilisation de JSON pour structurer les données
  - Un composant Arduino est représenté par un élément JSON
  - Un composant est identifié par le numéro du pin auquel il est branché sur l'Arduino
  - JSON transformé en objet Java et inversement

    getComponent(Tinkershield.i_0, LinearPotentiomenter.class);
- Semblable à l'utilisation d'une bdd traditionelle

5. Rideau de fer – Tests unitaires et d'intégration

- Développement plus aisé et plus rapide
- Nécessité de simuler l'Arduino
  - Utilisation de socat
  - Deux niveaux d'abstraction :
    - Simuler la lecture et l'écriture de chaînes de caractères
    - Simuler l'envoi d'informations représentant des composants de l'Arduino

5. Diagramme de simulation de l'Arduino



5. Rideau de fer - Clients

- Une interface graphique pour contrôler le rideau de fer
- Un outil en ligne de commande pour recevoir des notifications, appelé icwot (Inversion of Control for Web Of Things). Principe de l'inversion de contrôle

5. Rideau de fer - Conclusions

- Deux bibliothèques créées et utilisables comme dépendances Maven
- Définition d'un nouvel archétype Maven
- Un service Web avec Jersey
- Un client Web
- Un serveur pour recevoir les notifications (icwot)

6. Conclusions du travail

- Le méta-modèle offre de belles possibilités de développement
- Basé sur des technologies open-source, facilite l'implémentation de futurs applications du Web des Objets
- Plusieurs possibilités d'améliorations
  - Meilleure communication Arduino <-> Serveur
  - Langage de gestion des événements
  - Modélisations de composants matériels
  - Meilleure bibliothèque de tests