

Conception d'une application modulaire pour une passerelle de visualisation intégrant des données issues de divers dispositifs

Travail de Bachelor

Non confidentiel

**Département :** TIC

**Filière :** Informatique et systèmes de communication

**Orientation :** Informatique logicielle

Nicolas Crausaz

11 avril 2023

Travail proposé par :

Loris Gavillet

YALK

Rue Basse 43, 1422 Grandson

Supervisé par :

Patrick Lachaize

# Préambule

Ce travail de Bachelor (ci-après TB) est réalisé en fin de cursus d’études, en vue de l’obtention du titre de Bachelor of Science HES-SO en Ingénierie.

En tant que travail académique, son contenu, sans préjuger de sa valeur, n'engage ni la responsabilité de l'auteur, ni celles du jury du travail de Bachelor et de l'Ecole.

Toute utilisation, même partielle, de ce TB doit être faite dans le respect du droit d’auteur.

HEIG-VD

Le Chef du Département

Yverdon-les-Bains, le Sélectionner une date

# Authentification

Le soussigné, Nicolas Crausaz, atteste par la présente avoir réalisé seul ce travail et n’avoir utilisé aucune autre source que celles expressément mentionnées.

Nicolas Crausaz

Yverdon-les-Bains, le Sélectionner une date

# Résumé

Cliquez ou appuyez ici pour entrer du texte.

**Table des matières**

[Préambule I](#_Toc132120067)

[Authentification III](#_Toc132120068)

[Résumé V](#_Toc132120069)

[Chapitre 1 Introduction 1](#_Toc132120070)

[1.1 Contexte 1](#_Toc132120071)

[1.2 Adapter votre modèle 1](#_Toc132120072)

[1.3 Exemple de figure 1](#_Toc132120073)

[1.4 Tableaux 1](#_Toc132120074)

[1.5 Notes de bas de page 2](#_Toc132120075)

[1.6 Citations et bibliographie 2](#_Toc132120076)

[Chapitre 2 Conclusion 3](#_Toc132120077)

[Bibliographie 4](#_Toc132120078)

[Annexes 5](#_Toc132120079)

**Table des figures**

[**Figure 1** - Exemple de graphique 1](#_Toc129262255)

**Liste des tableaux**

[**Tableau 1** - Liste des cantons 2](#_Toc129262231)

**Liste des codes sources**

[**Listing 1** - Exemple de code Java 1](#_Toc129262239)

## Introduction

**Ce travail de Bachelor consiste en la mise en place d’une application Web, destinée à s’exécuter sur une passerelle physique dans le réseau local d’un utilisateur. L’intérêt de cette passerelle est d’offrir un outil central pour mettre en place des interactions avec des services externes et dispositifs réseau, sous forme de modules. Cette solution permet à des utilisateurs d’automatiser des actions en utilisant / développant des modules interagissant avec leurs propres services existants. La passerelle reliée à un ou plusieurs moniteurs permettra l’affichage d’informations issues des interactions grâce à un tableau de bord modulable.**

**Ce travail revient à effectuer un « Proof of Concept » permettant de mettre en place une architecture de base et son implémentation, ainsi qu’un module, pour démontrer les possibilités et les limites d’une telle solution.**

**TODO…**

### Contexte

**TODO**

### Cahier des charges

**Les besoins fonctionnels suivants sont considérés comme des éléments de priorité, composant les objectifs fondamentaux du travail.**

1. ****Modules****

**Un module est un composant logiciel destiné à offrir des fonctionnalités à la passerelle, principalement de gérer les interactions avec un dispositif précis : une API externe, un dispositif USB, Wifi etc. Les besoins liés aux modules sont :**

1. **Gestionnaire : Permettant l’activation / désactivation et la configuration du comportement d’un module.**
2. **Conception module « Proof of Concept » : Développement d’au moins un module spécifique permettant de gérer l’interaction entre un dispositif local et un service externe, permettant de démontrer l’intérêt de la solution.**
3. ****Application Web****
4. **Structure : L’application offre deux interfaces : administration et visualisation. L’interface d’administration nécessite une authentification de l’utilisateur, la visualisation est accessible sans authentification, car elle ne propose pas de fonctionnalité d’édition. Les deux interfaces sont accessibles uniquement dans le réseau local.**
5. **Interface d’administration : permet l’activation / désactivation et la configuration des modules par l’utilisateur. Permet aussi la configuration du tableau de bord modulable, en laissant le choix à l’utilisateur de disposer les affichages des modules selon ses préférences.**
6. **Interface de visualisation : L’interface de visualisation affiche le tableau de bord selon la configuration effectuée dans l’administration. Lors d’une modification de la disposition des modules depuis l’administration, le tableau de bord devra se mettre à jour sans nécessiter d’interaction supplémentaire de l’utilisateur.**

****Besoins non fonctionnels****

**Support physique et système d’exploitation**

**La passerelle sera un Raspberry Pi relié au minimum à une source de courant, une connectivité réseau (avec ou sans fil) et pouvant être reliée à un ou deux moniteurs externes. Le système d’exploitation contiendra le nécessaire pour faire fonctionner l’application Web.**

**Architecture locale**

**Tout le nécessaire au bon fonctionnement de l’application s’exécutera sur la passerelle. La passerelle et son application web ne seront pas exposée en dehors du réseau local, mais pourront interagir avec des services externes (http).**

**Performance**

**Du fait de l’architecture (ressources) de la passerelle relativement limitée, l’application s’exécutera sans consommation trop excessive de ressources et répondra dans un temps acceptable de manière à offrir une utilisation fluide.**

**Interface intuitive**

**L’interface de l’application web sera intuitive à utiliser pour un utilisateur ayant un bagage technique lui permettant de comprendre et configurer les modules et interactions.**

**Application fiable**

**L’application doit être capable de gérer des erreurs pouvant survenir et de pouvoir continuer à fonctionner. En cas de perte de connectivité puis de reconnexion réseau, l’application devra pouvoir refonctionner correctement sans interaction utilisateur.**

**En cas d’erreur non récupérable, il sera acceptable dans ce projet de simplement redémarrer l’application ou la passerelle.**

**Sécurité**

**L’application n’est pas conçue pour gérer des interactions critiques pouvant avoir des aspects de sécurité importants (transactions financières, etc.). Dans le cadre de ce travail, il n’est pas non plus nécessaire d’implémenter un système de sauvegarde de la configuration de la passerelle ou de sa réplication.**

****Extensions****

**Les extensions ci-dessous sont des besoins fonctionnels complémentaires faisant partie de la solution complète souhaitée par l’entreprise. Ces dernières ne seront pas réalisées dans le cadre de ce projet, sauf si le temps le permet, mais qu’il faut garder en tête lors de la conception des autres besoins pour prévoir leur ajout ultérieur.**

**Modules**

**Dans la version souhaitée par les besoins fonctionnels, tous les modules sont stockés localement sur la passerelle. Il serait intéressant de mettre en place un système d’import de nouveaux modules par l’utilisateur, afin de mettre en avant l’aspect modulaire de l’application.**

**Autonomie**

**Mise en place et configuration automatique / assistée de la passerelle dans le réseau local de l’utilisateur. Ceci permet de s’assurer une configuration correcte et connue pour l’exécution de l’application.**

**Gestion des écrans**

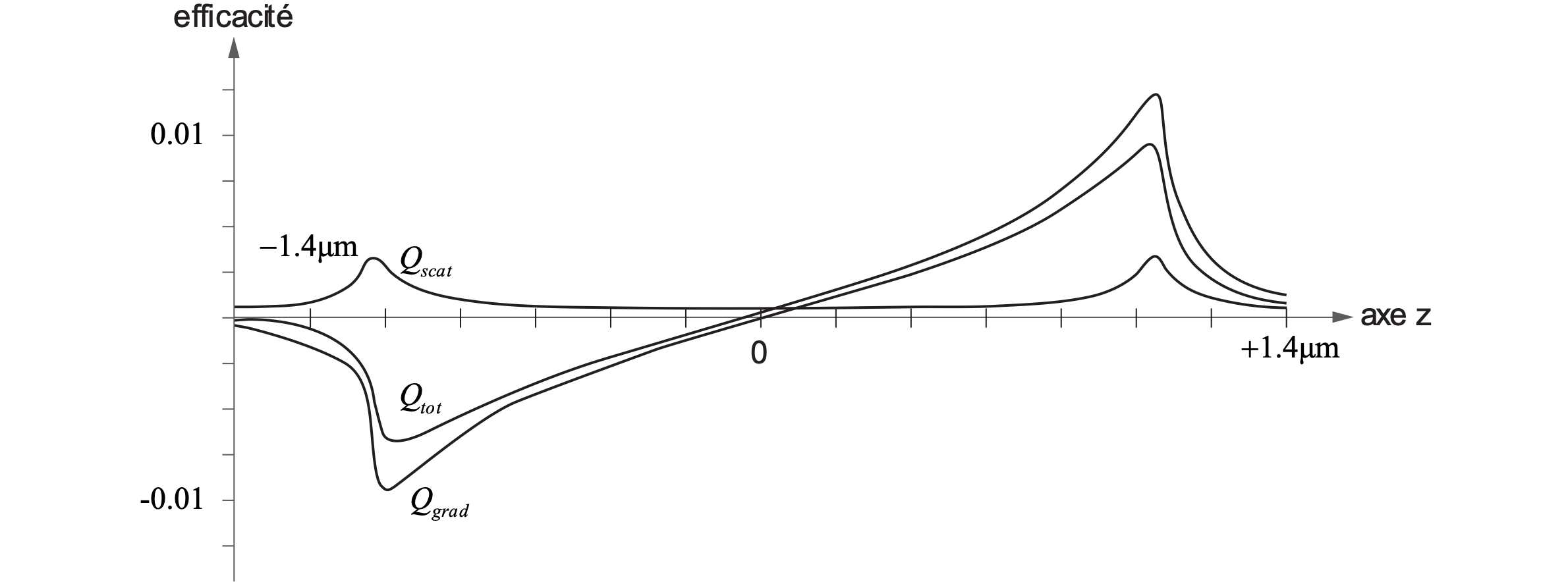
**Détection des écrans reliés à la passerelle pour pouvoir adapter le contenu du tableau de bord, selon l’orientation, la taille, la disposition relatives des écrans. Ceci permettrait d’ajouter un indicateur visuel et une aide à l’utilisateur configurant le tableau de bord.**

**Système d’exploitation**

**Il serait imaginable de créer une image d’un système d’exploitation contenant le strict nécessaire au fonctionnement de l’application. Ceci permettrait de limiter les interactions possibles de l’utilisateur final avec le logiciel, améliorant la sécurité et la stabilité de l’application.**

### Contraintes client

Pour présenter vos résultats d’expérience, vous pouvez soit dessiner des graphiques manuellement en utilisant des outils de dessin vectoriel comme Inkscape ou Adobe Illustrator, comme illustré à la Figure 1.



**Figure 1** - Exemple de graphique

Notez que la coloration syntaxe est conservé lorsque vous copier-coller du code depuis Intellij IDEA. Le Listing 1 en est un exemple.

|  |
| --- |
| public class Main {  public static void main(String[] args) {  System.*out*.println("Hello world!");  }  } |

**Listing 1** - Exemple de code Java

### Tableaux

Concernant les tableaux un seul conseil : restez simple et minimaliste, n’ajoutez des séparateurs que là ou c’est nécessaire pour améliorer la lisibilité. Une liste de quelques cantons suisses est donnée à titre d’exemple dans le Tableau 1.

|  |  |  |
| --- | --- | --- |
| Abréviation | Nom du canton | Depuis |
| ZH | Zürich | 1er mai 1351 |
| BE | Berne | 6 mars 1353 |
| FR | Fribourg | 22 décembre 1481 |
| VD | Vaud | 19 février 1815 |
| VS | Valais | 4 août 1815 |
| NE | Neuchâtel | 19 mai 1815 |
| GE | Genève | 19 mai 1815 |

**Tableau 1** - Liste des cantons

### Notes de bas de page

Parfois, il est plus élégant d’annoter une définition en utilisant une note de bas de page [[1]](#footnote-1). Alternativement il est possible d’annoter un paragraphe avec une note marginale.

### Citations et bibliographie

Citer vos sources est essentiel. Avec **Microsoft Word** vous pouvez facilement citer des articles, des livres ou des sites internet. Toutes les citations dans le texte seront automatiquement regroupées en fin de document dans la section « Bibliographie ». Par exemple, citons un article d’Einstein (EINSTEIN 1905) ou le livre de Dirac (DIRAC 1981).

Parfois il peut être utile d’utiliser un gestionnaire de bibliographie. La communauté académique recommande l’outil [Zotero](https://www.zotero.org/) qui permet de gérer une bibliothèque numérique d’ouvrages et de références numériques. Il permet également de générer une bibliographie compatible avec **Microsoft Word**. Il suffira d’installer au préalable le plugin, pour pouvoir ajouter l’onglet *Zotero*.

## Analyse

### Division des objectifs

La première partie de l’analyse a consisté en la définition des fonctionnalités….

Cette division à permis de décomposer les bases de l’architecture de l’application, en séparant les fonctionnalités dans plusieurs composants ayant chaque leur rôle respectif.

**Module**

Configuration

Affichage

Structure

**Système de gestion de module (Module Manager)**

Chargement / suppression d’un module

Observation d’un dossier

Activation et désactivation de modules

Détection d’erreurs

**Interactions entre les composants**

EventEmitter (Pattern Observer)

http (frontend – backend)

Server Sent Events

**Serveur**

API accessible au réseau local

Pas vraiment rest…

Interface

Application web client

Listes des fonctionnalités par pages

### Module « Proof of Concept »

TODO : Choix de plusieurs idées de modules PoC

### Essais effectués

Lister les tests d’architecture effectués et les technologies testées

Parler de l’archi API – App client, Electron

Parler des tests effectués pour le rendu

Approches

Sous-division des objectifs

Système de module

Base de données

Restrictions, limitations

Tests effectués

Choix des technologies

Gestion des erreurs

## Architecture et modélisation

UMLs, graphiques

CI/CD

Monorepo

Mockups

## Implémentation

## Conclusion

Cliquez ou appuyez ici pour entrer du texte.

# Bibliographie

DIRAC, Paul Adrien Maurice. «The Principles of Quantum Mechanics.» *International series of monographs on physics.* Clarendon Press, 1981.

EINSTEIN, Albert. «Zur Elektrodynamik bewegter Körper. (German) [On the electrodynamics of moving bodies].» *Annalen der Physik* 322, n° 10 (1905): 891-921.

# Annexes

**Glossaire**

**Index**

Adobe Illustrator, 1

Inkscape, 1

1. La note en bas de page (ou note de bas de page) est une forme littéraire, consistant en une ou plusieurs lignes ne figurant pas dans le texte. [↑](#footnote-ref-1)