

# Cahier des charges – v1

---

PROJET	NAO BOX
DOMAINE D'ACTION	Pilotage du robot NAO lors de sa présentation sur les salons
DESCRIPTION	Mettre à disposition des personnes qui présentent NAO sur les salons un boîtier de contrôle de ses actions. Le robot sera commandé de manière intuitive via un seul périphérique afin d'alléger le package nécessaire à son fonctionnement
MAITRISE D'OUVRAGE	Bernard BUFFIERE
CHEF DE PROJET	Elie METAHRI
ETUDIANTS	Elie METAHRI, Bastien VAUTIER, Jérémie LIECHTI



## Introduction

---

Ce projet est proposé par l'école 3iL de Rodez dans le cadre de notre projet tutoré de fin d'étude. Dans de nombreux salons, le robot NAO est employé par l'école pour réaliser des démonstrations et de facto attirer de nombreux curieux sur le stand de l'école.

Etant donné que plusieurs éléments parfois lourds et encombrants (ordinateur, tablette, connectiques...) doivent être amenés sur le salon pour effectuer les démonstrations dans de bonnes conditions, il n'est pas aisé de transporter tout ce matériel. C'est pourquoi il est nécessaire de développer un système tout en un, simple, léger et facile à utiliser pour pouvoir interagir avec le robot et en faisant appel à un nombre restreint d'équipements.

Actuellement, pour son bon fonctionnement, NAO doit être équipé d'un PC portable jouant le rôle de serveur, d'une tablette tactile connectée en wifi au serveur pour transmettre les ordres au robot ainsi que d'un routeur wifi permettant de réaliser la liaison entre les différents appareils : ce n'est pas simple !

# Expression fonctionnelle du besoin

---

## Description du Projet

Ce projet a pour but la réalisation d'un point d'accès pour NAO. Ce point d'accès permettra aux personnes utilisant le robot de le contrôler à l'aide d'un PC portable ou d'une tablette afin de lui faire faire différentes actions qui ont été programmées et intégrées au préalable dans la mémoire interne du robot. Ce point d'accès permettra également de relayer à la future application et donc à la personne qui le commande des informations de monitoring tel que le niveau de batterie restant et pourquoi pas l'état de chauffe de ses moteurs.

## Objectifs du projet

Les objectifs sont simples : avoir un système Plug & Play le plus simple et le plus ergonomique possible ce qui signifie que le système devra être simple et accessible, joliment présenté et facile à utiliser. Le système devra opérer en sécurité grâce à des accès à droits restreints (système de sessions avec un compte utilisateur et compte administrateur dissocié). Il doit également être autonome. A tout moment l'utilisateur doit être en mesure d'avoir accès aux données de gestion et de contrôle des paramètres de NAO.

## Quelles sont les contraintes ?

La NAO-BOX doit être autosuffisante avec son alimentation, ce qui interdit le développement d'un système nécessitant une ou des sources externes. Du point de vue visuel, il doit avoir éventuellement un bouton de reset ainsi qu'un voyant signalant que le système est fonctionnel. Par ce que la sécurité est importante ici, le système doit être en mesure de refuser toute communication avec un périphérique ou appareil non autorisé, ce qui implique qu'il va falloir mettre en place un système d'exclusions pour permettre aux périphériques et appareils autorisés de se connecter à la NAO-BOX pour dialoguer avec le robot.

## Solution proposée

---

De manière globale, la solution que nous souhaitons apporter pour répondre au besoin est une solution peu onéreuse, et facile à mettre en place. En effet, la NAO-BOX peut être réalisée grâce à l'utilisation d'un micro-ordinateur appelé Raspberry Pi dont la taille n'excède pas celle d'une carte de crédit. De plus, il est possible d'ajouter des modules à ce Raspberry afin de pouvoir communiquer facilement avec le robot.

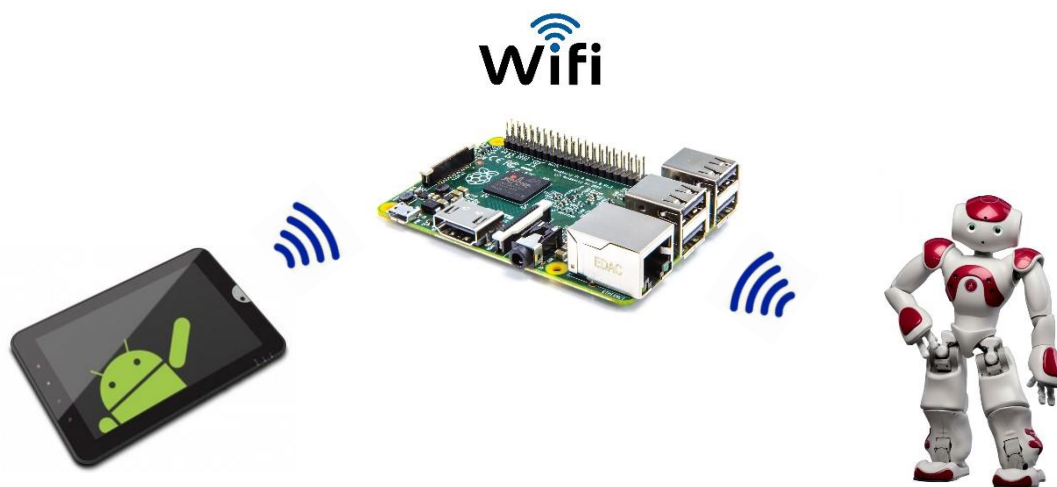


Schéma d'ensemble de la solution

### Matériel nécessaire pour la réalisation du projet

Comme dit précédemment, cette solution est peu onéreuse dans la mesure où ce système exige peu de matériel et que 3iL en possède déjà une bonne partie. De plus, et nous le verrons plus bas dans ce document, ce système nécessite plus de développement logiciel (aucun surplus de matériel n'est demandé), que de besoins pratiques en terme de composants et de périphériques.

MATERIEL	Tablette tactile
UTILITE	Elle permettra de se connecter à l'interface web pour envoyer des commandes et des ordres à NAO
CONTRAINTES	<ul style="list-style-type: none"><li>▪ Elle possédera une carte wifi pour permettre les échanges avec le Raspberry</li><li>▪ Ainsi qu'un navigateur web pour exécuter l'application qui sera développée</li></ul>

MATERIEL	Robot NAO
UTILITE	C'est l'objet de démonstration. Il est le cœur du système. Etant déjà possédé par l'école, il ne sera pas nécessaire de l'acheter à nouveau

MATERIEL	Raspberry Pi
UTILITE	<ul style="list-style-type: none"><li>▪ Il permettra le dialogue avec le robot et le périphérique via wifi</li><li>▪ En outre, il servira de support à l'installation d'un serveur web (Apache et SQL) pour gérer les informations</li></ul>
CONTRAINTES	<ul style="list-style-type: none"><li>▪ L'ordinateur étant vendu nu et sans carter, il sera nécessaire d'y ajouter un carter et une alimentation (micro USB 5v)</li><li>▪ La communication avec les périphériques se faisant uniquement par wifi, il y'a obligation d'acheter une carte wifi (clé USB wifi)</li><li>▪ Et parce que sans support de stockage on ne peut rien faire, il faudra également veiller à l'achat d'une carte mémoire avec idéalement 16 Go de stockage</li><li>▪ La souris, le clavier et l'écran nécessaires à sa configuration pourront être fournis par l'école</li></ul>
REMARQUE	Plusieurs modèles de ce micro-ordinateur existent actuellement. En ce qui concerne la fluidité du matériel, il serait nécessaire de s'orienter vers le dernier modèle de Raspberry Pi 2 modèle B

## Communication entre la tablette et le robot NAO

La communication entre la tablette et le robot NAO doit se faire via le wifi du Raspberry. Ce dernier doit créer un réseau afin que la tablette ainsi que NAO puissent s'y connecter. La communication entre le robot et le Raspberry se fera grâce à l'échange de trames fabriquées via le langage JavaScript et utilisant principalement la librairie web fournie avec NAO et en s'appuyant sur la librairie socket.IO du Framework Node.JS. L'étude de la documentation du robot sera alors nécessaire afin de connaître les protocoles de communication et d'utiliser un langage plus approprié si le JavaScript ne convient pas.

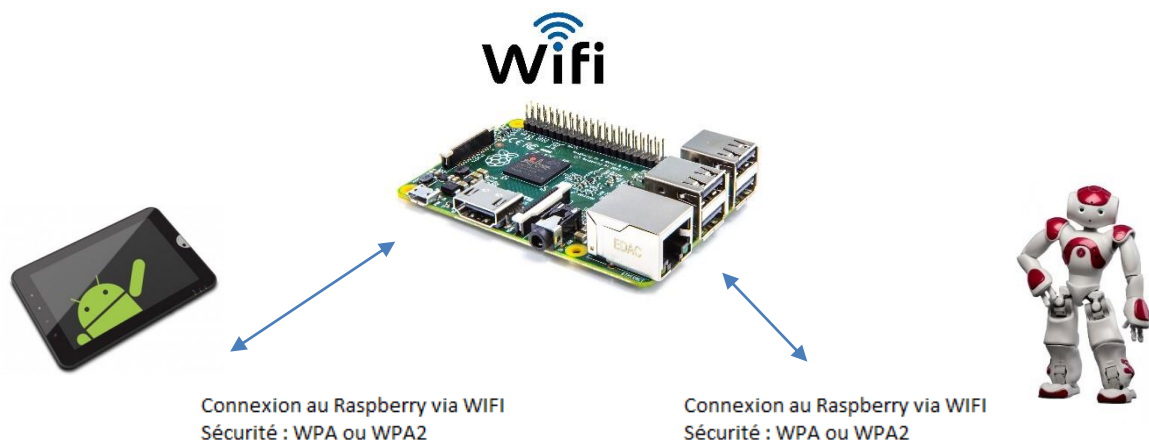


Schéma d'ensemble de la solution

## Détails de l'application

L'application doit être une application simple et fonctionnelle. Cette dernière sera réalisée sous la forme d'un site Web. L'utilisateur disposera alors d'un espace d'administration ainsi que d'un espace utilisateur. Ce dernier pourra alors gérer via l'interface d'administration les fonctionnalités que les utilisateurs lambda posséderont lorsqu'ils seront connectés sur l'espace User.

Quelques précisions concernant les différentes interfaces ou modules :

MODULE D'INTERACTION	Administrateur
PRECISIONS	<ul style="list-style-type: none"><li>▪ Elle sera accessible via un couple de login et mot de passe</li><li>▪ Cette interface permettra d'ajouter / masquer des fonctionnalités pour les utilisateurs de l'application</li><li>▪ Elle permettra d'autoriser les périphériques à se connecter à NAO</li><li>▪ Elle permet de créer et de gérer des fonctionnalités de NAO</li></ul>

MODULE D'INTERACTION	Utilisateur
PRECISIONS	<ul style="list-style-type: none"><li>▪ Elle sera accessible via un couple de login et mot de passe</li><li>▪ Cette interface permettra de transmettre les ordres à NAO</li></ul>

<b>MODULE D'INTERACTION</b>	Point d'entrée de l'application
<b>PRECISIONS</b>	<ul style="list-style-type: none"><li>▪ Il sera capable d'interroger une base de données afin de savoir si l'appareil qui tente de se connecter à lui est autorisé. Si ce dernier est autorisé, il vérifie s'il possède les droits administrateur ou bien les droits utilisateur</li><li>▪ Dans le cas contraire il devra rejeter la connexion</li></ul>

## Technologies utilisées

Voici ci-dessous la liste des différentes technologies que nous allons utiliser pour la réalisation de ce projet :

<b>EDITEURS DE TEXTE</b>	<ul style="list-style-type: none"><li>▪ NotePad++</li><li>▪ Sublime Text</li><li>▪ Dreamweaver</li></ul>
<b>LANGAGE PROGRAMMATION</b>	<ul style="list-style-type: none"><li>▪ HTML, PHP, SQL</li><li>▪ C++ (pour la vérification des autorisations d'accès)</li></ul>
<b>SERVEUR</b>	<ul style="list-style-type: none"><li>▪ Wampserver lors du développement</li><li>▪ Lamp lors de la mise en production</li></ul>
<b>DOCUMENTATION UML</b>	Visual Paradigm
<b>DOCUMENTATION PROJET</b>	<ul style="list-style-type: none"><li>▪ Gantt Project (réalisation du diagramme de Gantt)</li><li>▪ Suite Office</li></ul>
<b>PARTAGE DE FICHIERS</b>	Espace partagé sur Google Drive
<b>PARTAGE DE SOURCES</b>	Github
<b>MAQUETTAGE IHM</b>	<ul style="list-style-type: none"><li>▪ Photoshop</li><li>▪ Paint</li></ul>