

*Application de **guidage***



Halle aux Farines

Scénarios d'utilisation

Étudiants



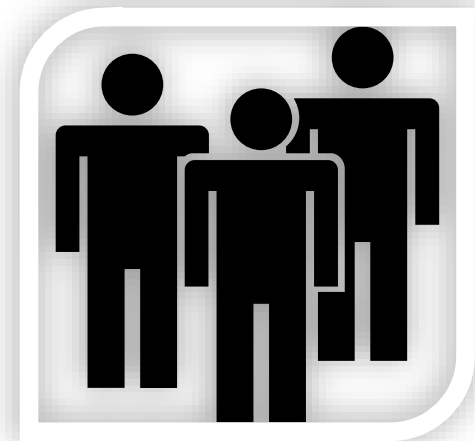
*Trouver un
itinéraire optimal*

Enseignants



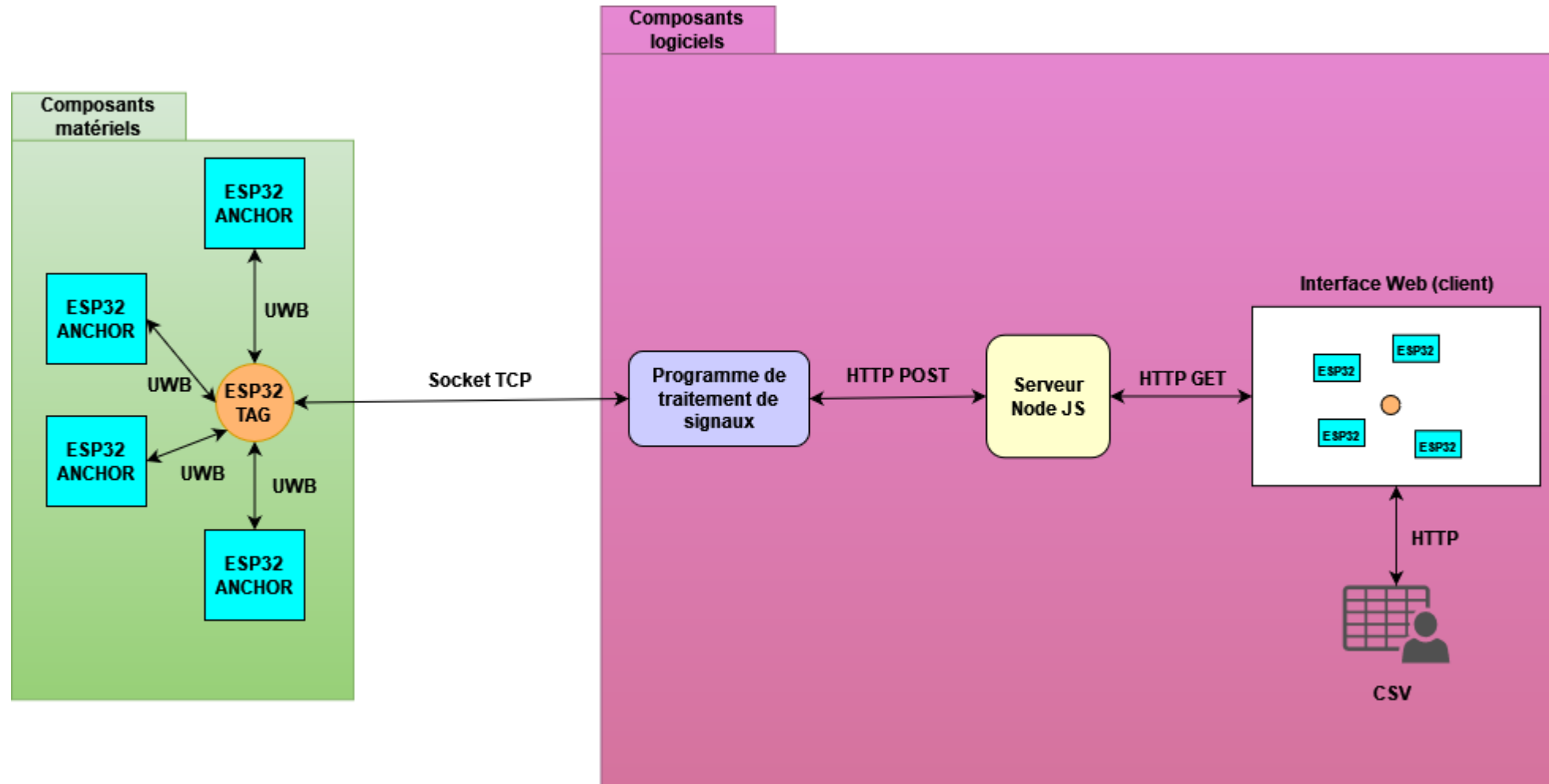
Gain de temps

Visiteurs

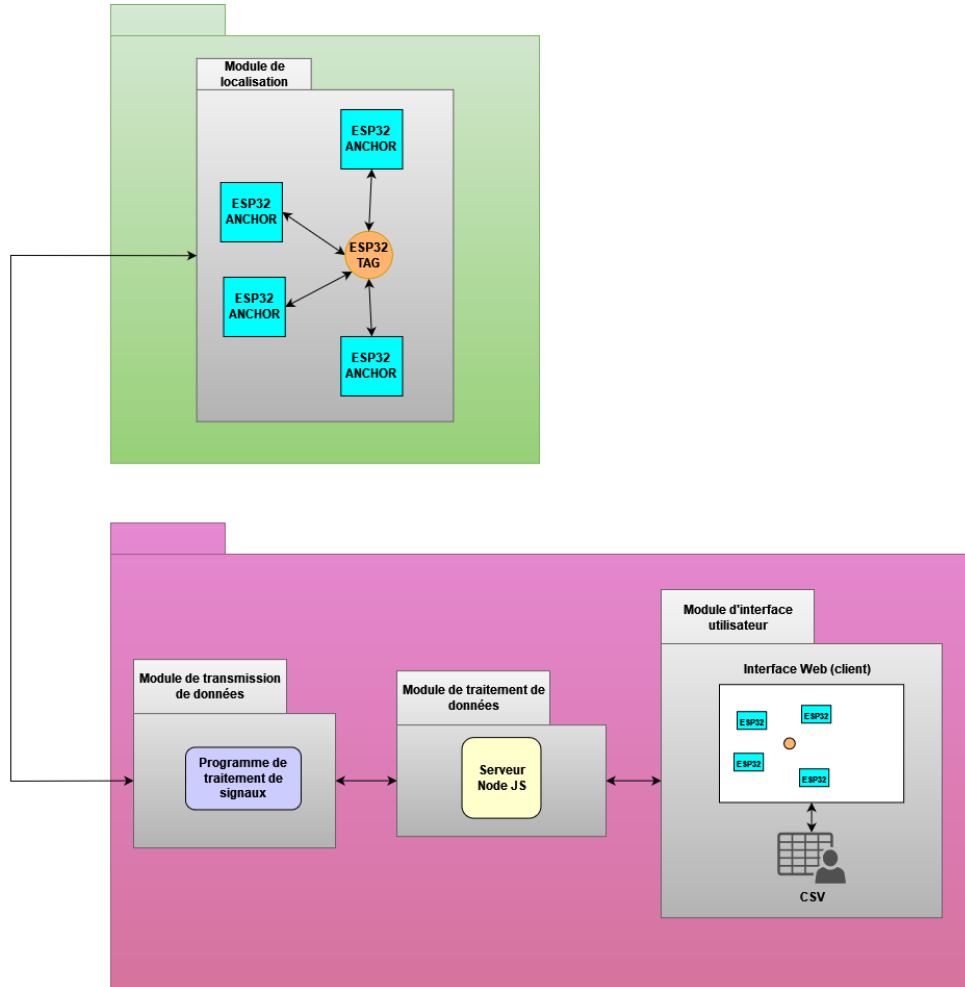


*Repérer les points
d'intérêts*

Architecture, conception et gestion de projet (1/4)



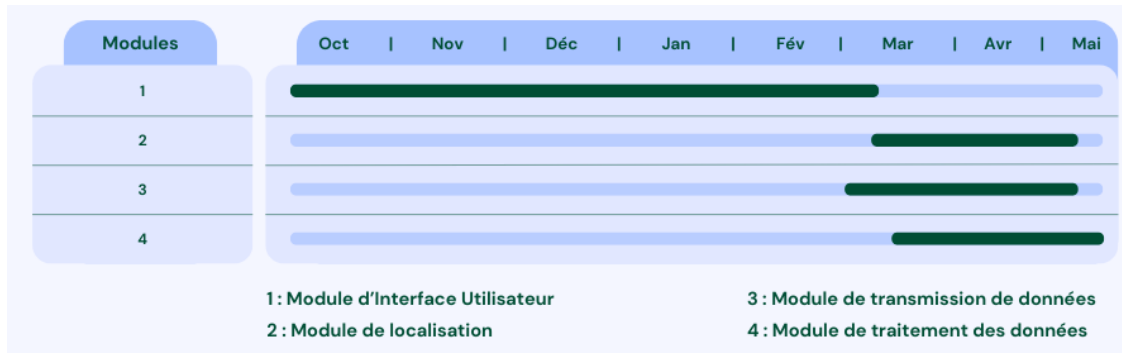
Architecture, **conception** et gestion de projet (2/4)



Module	Tâches
Module d'interface utilisateur	<ul style="list-style-type: none"> - Modélisation du plan des étages en utilisant les plans vectoriels SVG - Modélisation des plans sous forme de graphe stocké dans un fichier CSV - Mise en place de l'interface Web - Implémentation de l'algorithme de Dijkstra pour trouver le plus court chemin - Implémentation d'un espace administrateur
Module de localisation	<ul style="list-style-type: none"> - Implémentation du code dans le matériel ESP32 (émetteurs et récepteurs)
Module de transmission de données	<ul style="list-style-type: none"> - Implémentation du programme de traitement de signaux en Python
Module de traitement de données	<ul style="list-style-type: none"> - Implémentation du serveur NodeJS - Implémentation de l'algorithme de trilatération

Architecture, conception et **gestion de projet** (3/4)

Calendrier de Jalon



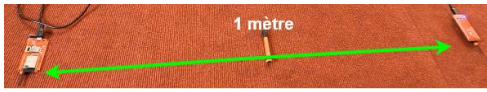
Compétences techniques



Architecture, conception et gestion de projet (4/4)

Tests Unitaires

Module de localisation :



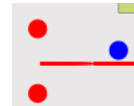
Anchor 1	Tag
7D, 0.99	84, 0.99
7D, 1.01	84, 1.01
7D, 1.00	84, 1.00
7D, 0.97	84, 0.97

Module de traitement de données et d'Interface Web :

1) Coordonnées estimées de l'utilisateur : { x: 0.6, y: 0.9 }



2) Coordonnées estimées de l'utilisateur : { x: 1.1, y: 1.8 }



Test d'Intégration

Module de transmission de données :

1) Data successfully sent to the server.
{ 'links': [{ 'A': '85', 'R': '1.1' }, { 'A': '84', 'R': '1.1' }] }
{ 'A': '85', 'R': '1.1' }
{ 'A': '84', 'R': '1.1' }

2) Data successfully sent to the server.
{ 'links': [{ 'A': '85', 'R': '1.5' }, { 'A': '84', 'R': '1.9' }] }
{ 'A': '85', 'R': '1.5' }
{ 'A': '84', 'R': '1.9' }

Les difficultés rencontrées et leurs solutions

- Localisation sur plusieurs étages

- **Problème** : Impossible d'implémenter la localisation sur deux étages avec la triangulation en 3D en raison des variations de hauteur des étages et des obstacles.
- **Solution** : Utilisation de la trilatération en 2D pour la localisation sur un seul étage.

- Portée limitée des ESP32

- **Problème** : Les ESP32 avaient une portée limitée à 30 mètres, insuffisante pour une couverture complète.
- **Solution** : Ajout de deux bornes supplémentaires à 30 mètres des premières pour assurer une couverture continue :



Programmation

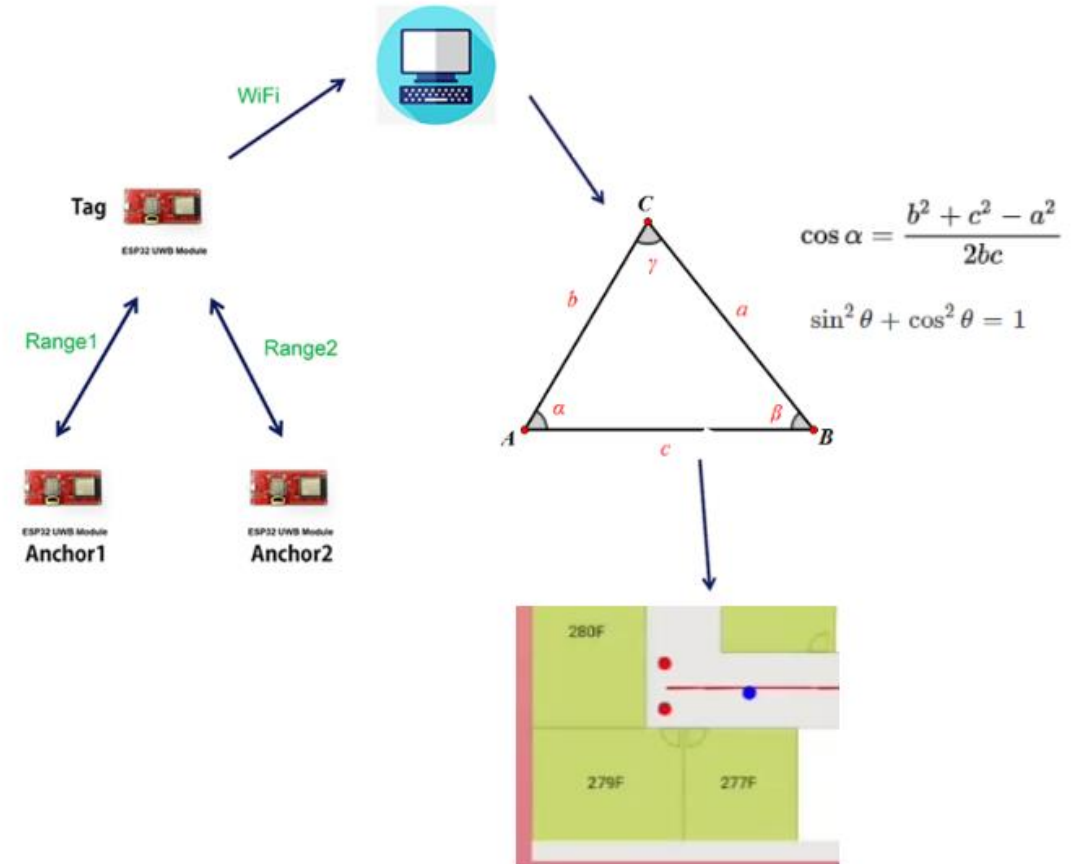
Algorithme de trilatération

```

function tagPos(a, b, c) {
  // Calculer le cosinus de l'angle A en utilisant la loi des cosinus
  const cosA = (b * b + c * c - a * a) / (2 * b * c);

  // Calculer les coordonnées x et y du point C
  const x = b * cosA;
  const y = b * Math.sqrt(1 - cosA * cosA);

  // Retourner les coordonnées arrondies à une décimale
  return [parseFloat(x.toFixed(1)), parseFloat(y.toFixed(1))];
}
  
```



Conclusion

- **À quoi ressemblerait la version 2 de votre logiciel ?**
 - Extension de la localisation à plusieurs étages.
 - Amélioration de l'interface utilisateur avec des notifications en temps réel.
 - Optimisation des performances de l'algorithme de recherche du plus court chemin en utilisant par exemple l'algorithme A*.
- **Que changeriez-vous si vous deviez refaire votre projet ?**
 - Choix de matériel avec une meilleure portée et précision.
 - Planification anticipée de la gestion des obstacles physiques.