**Plan du mémoire**

**Introduction générale**

* Présenter le contexte : existence de beaucoup d’applications basées sur le modèle publication/abonnement et nécessité de normaliser les APIs à la manière des APIs Web, grâce à la mise en place d’une spécification OpenPS équivalente à OpenAPI.

**Chapitre 1** : La spécification OpenAPI

* Étendre le document que tu as déjà écrit
* Présenter OpenAPI a partir de l’article de sa spécification
* Citer les outils liés à OpenAPI : le générateur, l'éditeur en ligne, etc.

**Chapitre 2 :** Le modèle publish/subscribe

* Compléter le document que tu as déjà commencé à écrire
* Présenter l'utilité de pub-sub pour les nouvelles applications telles les applications IoT
* Présenter quelques projets utilisant les protocoles publish/subscribe : citer quelques exemples de protocoles, MQTT, syncps (NDN), et le service Cloud Pub/sub de google
* Comparer pub-sub avec client/serveur (dans un tableau)
* Présenter les détails du protocole AMQP

**Chapitre 3** : La spécification OpenPS

* Expliquer l’intérêt de cette spécification pour l’uniformisation des APIs.
* Présenter la spécification OpenPS (a partir du document de Abane)

**Chapitre 4** : Un générateur de code pour OpenPS

* Présenter le générateur automatique de code (OpenAPI codegen)
* Présenter le langage de templating Mustache et son fonctionnement
* Décrire l’architecture du générateur OpenPS (je vais t'aider un peu ici)
* Cas d’utilisation : décrire l'exemple telemetry controller-agent, et montrer le code généré

**Conclusion générale**

* Expliquer l’intérêt du travail et quelques perspectives
* Mentionner quelques limites des spécifications OpenAPI et OpenPS