**Une spécification pour le model Publication/Abonnement**

**Introduction générale** :

Dans cette dernière décennie, La rivalité entre les entreprises relève plusieurs défis technologiques parmi lesquels la possibilité des systèmes interactifs autonome, la conception des logiciels agissant comme passerelle entre d'autres applications pour offrir des services, la collecte d'information et leur diffusion. Cela nécessite l'automatisation des systèmes car l'internet autorise toute sorte d'objet pouvant communiquer en réseau. De ce fait, tout programme de recherche d'information et les technologies de la communication (TIC) sont motivés par la nécessité de combler l'écart croissant entre la demande et l'offre de services [1]. Depuis la reconnaissance de l'approche pub/sub (publication/abonnement) pour la diffusion d'information, sa contribution dans la recherche et dans la commercialisation des produits, ont permis son utilisation dans plusieurs contextes d'applications. Les systèmes de communication dotés de ce paradigme (pub/sub) ont connu un gain de popularité significatif et ont montré son efficacité dans un grand nombre de domaine. Alors que la conception de base de pub/sub dérive de la manière dont les notifications circulent des expéditeurs aux destinataires et prend son efficacité de la manière dont l'information est communiquée. C'est-à-dire que, les destinateurs expriment leur intérêt en émettant des abonnements pour des notifications spécifiques, indépendamment des éditeurs (*publiser*) qui les produisent, puis ils sont notifiés de manière asynchrone pour toutes les notifications [2]. Ce système est un modèle service-à-service permettant de découpler les applications afin d'augmenter les performances, la fiabilité et l'évolutivité. Il montre son importance dans la communication des applications instantanées distribuées.

L'ensemble des systèmes actuels utilisent des ressources limitées communique avec les Cloud et sont intéressés par d'autre information d'utilisateur d'où la nécessité d'amener un mécanisme d'intégration d'applications qui prédominent sur Internet qui sont les APIs REST. Ces APIs REST indiquent les formats de représentation des données. Elles décrivent les moyens de communication entre un client et un server. Mais cette technique de représentation de transfert d'état nécessite des configurations manuelles de la part des clients et un de ces grands obstacles est le manque d'exemple d'utilisation dans la documentation [3]. Il existe d'autre moyen automatisant ces difficultés et capable d'expliqué à la machine ainsi que les humains comment fonctionne une API REST appelé la spécification standard de programmation d'interface qui ouvre d'énormes opportunités pour un grand nombre de nouvelle application promettant d'améliorer la qualité de nos vies.

Une spécification similaire à celle de OpenAPI tenant compte de la criticité des ressources avec une communication asynchrone et permettant aux différentes objets dans Iot des prendre des décisions tout en offrant la possibilité pub/sub serait la bienvenue. Cette solution est également efficace en tant que middleware de messagerie pour l'intégration de services ou en file d'attente pour exécuter des tâches en parallèle car elle augmente la flexibilité et la stabilité du système dans son ensemble.

En programmation informatique, une interface de programmation d'application (API) est un ensemble de définitions de sous-programmes, de protocoles et d'outils pour la construction de logiciels d'application. En termes généraux, il s'agit d'un ensemble de méthodes de communication clairement définies entre différents composants logiciels. Une bonne API facilite le développement d'un programme informatique en fournissant tous les blocs de construction, qui sont ensuite assemblés par le programmeur.

Comme alternative, ils peuvent simplement ajouter des fonctionnalités supplémentaires. Cela fait de l'API REST la méthode d'intégration la plus utilisée.

Les API REST sont utilisées comme mécanisme d'intégration d'applications prédominant sur Internet.

Dans un environnement aux ressources limitées, il est important de comprendre la valeur

[1] Roffia, Luca, et al. "A semantic publish-subscribe architecture for the Internet of Things." *IEEE Internet of Things Journal* 3.6 (2016): 1274-1296.

[2] Baldoni, Roberto, et al. "On the modelling of publish/subscribe communication systems." *Concurrency and Computation: Practice and Experience* 17.12 (2005): 1471-1495.

[3] Sohan, S. M., et al. "A study of the effectiveness of usage examples in REST API documentation." *2017 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*. IEEE, 2017 page 53.