



**IMT Atlantique**

Bretagne-Pays de la Loire

École Mines-Télécom

# Introduction à ZeroMQ

Hugo SELLE

# SOMMAIRE

## 1. CONTEXTE

- 1.1 Contexte historique
- 1.2 Contexte technique
- 1.3 Suspendisse in dapibus

## 2. APPLICATION ET DEVELOPPEMENT

- 2.1 Communauté
- 2.2 Utilisation dans d'autres systèmes
- 2.3 Avantages et inconvénients



**IMT Atlantique**  
Bretagne-Pays de la Loire  
École Mines-Télécom

# CHAPITRE 1

## CONTEXTE



**IMT Atlantique**  
Bretagne-Pays de la Loire  
École Mines-Télécom

## 1.1 Contexte historique



Pieter Hintjens - (1962 - 2016)  
PDG belge chez *iMatix*

- ▶ 2004 : **AMQP** est développé pour la banque **JPMorganChase** par entre autre **iMatix**
- ▶ 2007 : le domaine zeromq.org est enregistré et le projet est lancé
- ▶ 2008 : ØMQ/0.x
- ▶ 07 / 2009 : ØMQ/1.x : pour les pionniers (finance)
- ▶ 01 / 2010 : ØMQ/2.x : ouverture au public
- ▶ 03 / 2010 : iMatix quitte le groupe de travail AMQP
- ▶ 2011 : CERN publie une étude comparant des solutions de **middleware** dont ØMQ
- ▶ 2011 : ØMQ/3.x : marché de masse (cloud)

- ▶ Boom des objets connectés dans les années 2000
  - ▶ La **connexion** entre noeuds était **coûteuse** :
    - custom TCP ou UDP
    - messages gérés par des **agents**
  - ▶ Il faut quelque chose de plus **rapide** et plus **simple** que AMQP
- ⇒ ZeroMQ : une bibliothèque de “messaging” rapide et haute-performance pour application distribuée et concurrente

	RabbitMQ	ZeroMQ
Gestion des messages	<b>basé sur un MOM</b> avec Agent de message	<b>sans Agent de message</b> (Message Broker)
Implémentation	<b>Framework</b> de messagerie	<b>Bibliothèque</b> de sockets
Langage	Erlang	C++
Débit	80 k msg / sec	8 M msg/sec
Latence	0.4 ms	0.03 ms

# CHAPITRE 2

# APPLICATION ET

# DÉVELOPPEMENT



**IMT Atlantique**  
Bretagne-Pays de la Loire  
École Mines-Télécom

## 2.1 Communauté

La communauté est très active autour de ØMQ :

- ▶ **1000 développeurs**
- ▶ **120 personnes sur IRC** (Internet Relay Chat)
- ▶ utilise le Collective Code Construction Contract **C4** mode de développement inspiré par :
  - ⇒ Permet aux nouveaux utilisateurs de participer au projet plus facilement
- ▶ Le guide de ZeroMQ a un exemple détaillé pas à pas qui montre comment contribuer au projet



WIKIPEDIA  
L'encyclopédie libre



GitHub



ØMQ - The Guide

<http://zeromq.org/docs/contributing>

<https://en.wikipedia.org/wiki/ZeroMQ>

<https://www.slideshare.net/pieterh/overview-of-zeromq>



## 2.2 Utilisation dans d'autres systèmes

Plusieurs systèmes utilisent ZeroMQ :

- ▶ **IPython** : un shell interactif pour Python et pour ØMQ la programmation parallèle
- ▶ **Dropbear** : un client et serveur SSH libre. Un Dropbox pour les systèmes embarqués sous Linux
- ▶ **Loggly** : Une solution pour la gestion de fichier de log pour le Cloud utilise ØMQ depuis 2010.
- ▶ **Lubyk** : Outil de processing (patcher) open source de signaux dynamiques en temps réel

IP[y]: IPython  
Interactive Computing



## 2.3 Avantages et inconvénients

Avantages	Inconvénients
<ul style="list-style-type: none"><li>▶ ØMQ plus <b>rapide</b> et plus <b>légère</b> que d'autres solutions</li><li>▶ De nombreuses possibilités d'architecture de communication de messages avec les patterns</li><li>▶ <b>Compatible</b> avec de nombreux langages et <b>intégrés</b> dans de nombreuses plateformes</li><li>▶ Facilité de <b>passage à l'échelle</b> de communication intra logiciel, inter logiciel à grande échelle</li><li>▶ Une couche de sécurité a été rajoutée en 2013</li></ul>	<ul style="list-style-type: none"><li>▶ ØMQ <b>n'est pas totalement robuste</b> : une donnée erronée et il est facilement possible de faire tomber l'application.</li><li>▶ ØMQ n'est <b>pas adapté à tous les types d'applications</b></li><li>▶ Plus <b>complexe à mettre en place dans un environnement ubiquitaire</b> car la détection d'un noeud manquant ne se fait pas rapidement</li><li>▶ <b>API bas niveau</b> et ne présente pas autant de fonctionnalités que d'autres bibliothèques ou framework</li></ul>