

# De quelle façon l'Internet des objets va-t-il affecter les testeurs ?

Dans une série d'articles, Paul Gerrard, consultant et spécialiste des tests, aborde diverses questions sur ces derniers. Dans cet article, il explique que l'Internet des objets (IoT) change complètement la donne pour les testeurs et pour les tests. Paul Gerrard présente une architecture composée de sept couches et traite de la portée de l'IoT et des problèmes qui en résultent en matière de tests.

## Introduction

Dans cette série d'articles, je souhaite explorer la façon dont l'Internet des objets (IoT, ou de manière plus large « Internet of Everything », IoE) va affecter les tests et les testeurs.

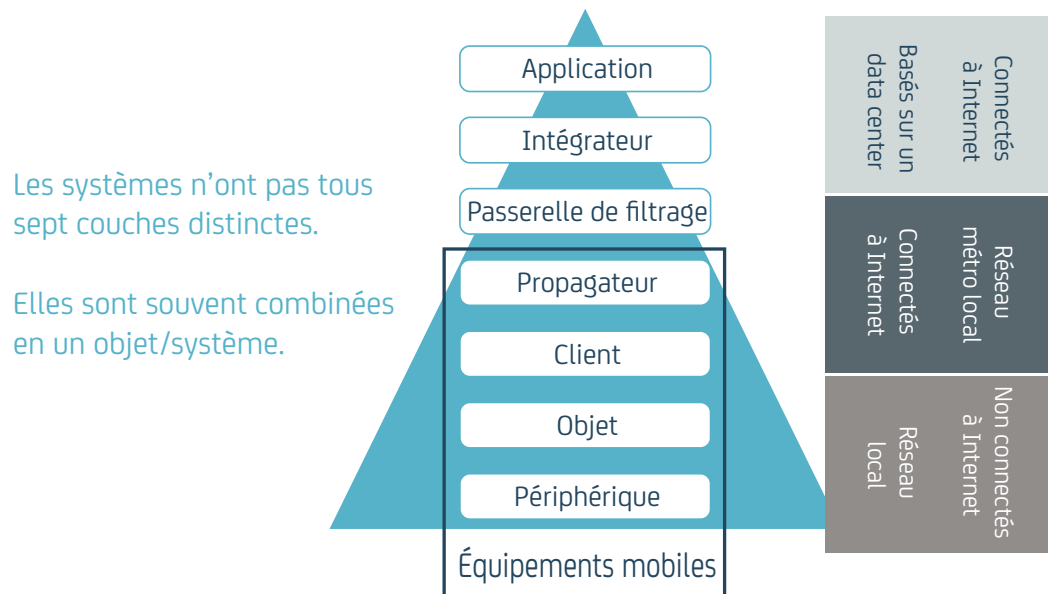
Il semble que tout doive un jour devenir « intelligent ». En parallèle des applications domestiques, des secteurs tels que la vente au détail, la fabrication, le transport, l'agriculture et les télécommunications adoptent l'IoT avec enthousiasme. Cela va sans dire, le gouvernement et l'armée poussent la recherche et, bien sûr, les phénomènes émergents (villes intelligentes) vont surveiller ou prendre le contrôle de notre quotidien.

Cet article fait référence à une série d'articles [1, 2, 3] dont je suis l'auteur et tire certaines conclusions quant à l'impact sur les tests et les testeurs.

### Un modèle d'IoE en couches

Au même titre que les normes évoluent pour les équipements réseau situés en périphérie, il n'existe pas de norme ou de convention en matière d'architecture pour l'IoE. Toutefois, certains modèles se dégagent et, comme on pouvait s'y attendre, leur architecture peut être décomposée en couches logiques.

## Architecture en sept couches



Ce modèle constitué de sept couches est un amalgame de plusieurs styles de schémas publiés. Il n'est associé à aucun équipement ou produit en particulier et ne s'aligne sur aucune technologie spécifique. En termes d'échelle et de perspective, ces couches sont mieux représentées sous la forme d'une hiérarchie ayant comme couche de base les appareils. Les sept couches sont décrites dans le document en référence [2]. Indépendamment de son implémentation, le modèle en couches peut vous aider à comprendre l'architecture de votre système.

## Les risques d'échec

L'architecture en sept couches peut vous aider à comprendre la fonction de chaque composant dans une implémentation IoE. J'ai utilisé le modèle en couches pour créer une liste de ce que j'appellerais les modèles de risque [2].

Toutefois, il existe aussi des risques pour la société ou les personnes relayés par les médias et auxquels nous devons être attentifs. Ils ne concernent peut-être pas votre application, mais si c'est le cas, ils seront certainement propres à votre projet. Voici les principaux participants :

- **Risques pour la société/les personnes** : la sécurité et le respect de la confidentialité priment.
- **Complexité** : les interactions entre les équipements peuvent être imprévues, inattendues et inconnues.
- **Confidentialité** : la collecte de données est répandue, mais invisible et largement hors de notre contrôle.
- **Utilisation frauduleuse** : l'IoT offre des avantages, mais l'omniprésence des réseaux et des données incitent aux activités criminelles.
- **Sécurité des entreprises** : les systèmes d'entreprise auparavant sécurisés sont désormais connectés à des réseaux beaucoup plus étendus et moins sécurisés.

Dans des contextes tels que les systèmes de transport, les lignes de production, les chaînes de télévision, la génération/distribution d'énergie et les villes intelligentes, le potentiel pour le piratage informatique, les perturbations et le terrorisme est sans limite. Le secteur de la sécurité a beaucoup à apprendre et encore davantage à faire, car l'IoT change complètement la donne.

L'IoE instaure des niveaux inédits de complexité et d'évolutivité. Les risques non fonctionnels sont relativement connus et nous savons comment les traiter. Par contre, la nouveauté réside dans la nécessité de réaliser des tests fonctionnels et des simulations à l'échelle.

## La portée des tests dans l'IoE

L'éventail de préoccupations qui accompagnent l'IoE est plus important que jamais. Les systèmes IoT ne sont pas tous immenses, complexes et coûteux, mais ils présentent tous un profil de technicité et de risque différent de ce que nous connaissons. Les principaux domaines d'action pour le testeur IoT :

- **Tests au niveau du matériel** : les appareils du plus bas niveau sont certes sophistiqués, mais ils exécutent des fonctions simples pour la majorité, prises en charge par les fabricants.
- **Évolutivité** : une application sur un appareil connecté peut avoir une architecture simple, mais son utilisation pourrait être étendue à plusieurs millions d'utilisateurs.
- **Fonctionnalité au niveau des objets et des serveurs** : la plupart des tests fonctionnels s'effectuent au niveau des concentrateurs locaux, des agrégateurs et des serveurs basés sur des data centers. Les architectures varient de simples applications Web à des systèmes hébergeant des dizaines de sous-systèmes.
- **Objets mobiles** : ils se déplacent au travers des périmètres des réseaux et errent entre les réseaux. Les conditions environnementales, les sources de données et la localisation des équipements affectent leur comportement. Ils sont sensibles à l'alimentation, aux interférences, à la puissance du réseau, aux problèmes d'itinérance et d'encombrement.
- **Réseaux itinérants** : certains systèmes (comme les voitures connectées) transportent leur propre réseau local. Un réseau itinérant rencontre d'autres réseaux susceptibles d'interférer ou risque d'introduire un réseau non autorisé ou non sécurisé, ce qui pose des problèmes de sécurité.
- **Les risques liés à la sécurité réseau se situent à plusieurs niveaux** : les appareils non autorisés utilisent votre réseau et espionnent ou injectent des données factices. Les points d'accès non autorisés détournent les connexions et les données de vos utilisateurs. Des vulnérabilités existent à tous les niveaux de votre architecture et sont propices aux attaques.

- **Enregistrement, provisioning, échec et sécurité des appareils** : l'enregistrement et le provisioning initiaux des appareils sont sujets aux échecs. Les appareils sont sensibles aux pannes d'alimentation, à la neige, à la chaleur, au froid, au vandalisme, aux animaux, aux voleurs, etc. Il peut être nécessaire de tester les mises hors tension et sous tension, ainsi que les processus automatisés d'authentification, de configuration et d'enregistrement.
- **Confusion au niveau de la collaboration** : les appareils itinérants (repreons l'exemple des voitures) collaborent de manière complexe et à très grande échelle. Cela dit, les accidents arrivent, les conducteurs changent d'avis, les places de stationnement se libèrent et se remplissent de manière aléatoire. L'algorithme d'optimisation doit donc s'adapter à des situations qui évoluent rapidement. Ces services ne doivent pas non plus dérouter les utilisateurs.
- **Intégration à tous les niveaux** : flux de données et de contrôle simples et complexes, de bout en bout.
- **Big Data - logistique** : les services de stockage du Big Data feront partie du système à tester.
- **Big Data - analyse et visualisation** : la visualisation et la science des données s'inscrivent vraisemblablement dans le périmètre de l'IoT. Cela inclut des données opportunes, précises et cohérentes, ainsi que leur filtrage, fusion, intégration et rapprochement.
- **Confidentialité des données personnelles et d'entreprise** : les pirates informatiques et les escrocs constituent une menace, mais votre propre gouvernement peut également être considéré comme un ennemi potentiel.
- **Appareils connectés et embarqués** : les appareils embarqués et connectés portés par les humains engendrent de nouveaux défis uniques.
- **Tout connecté** : le temps viendra où tous les appareils utilisés dans un hôpital, un hôtel ou une usine, par exemple, seront connectés.

Le nombre de problèmes que nous devons prendre en compte pour tester l'IoE s'est multiplié et l'envergure des tests requis s'est également accrue.

### Tests fonctionnels évolutifs

Lorsque nous testons la fonctionnalité de composants plus haut dans l'architecture, notamment au niveau de l'intégrateur et de l'application, nous pouvons être amenés à simuler des milliers ou des millions d'appareils sur le terrain. Le nombre de combinaisons et de permutations est difficilement calculable ou prévisible. Nos simulations génèrent en boucle les scénarios à tester, enregistrent les résultats et répètent potentiellement les simulations pour permettre une analyse ultérieure.

Les composants de plus haut niveau doivent pouvoir être testés. Nous aurons besoin d'équipements tels que des gestionnaires d'exception, des utilitaires qui injectent des données, capturent et reproduisent ou répètent des scénarios. Cem Kaner a beaucoup écrit sur ce qu'il appelle « les tests automatisés à grande échelle » [4]. C'est un bon point de départ.

Ces techniques peuvent également être appelées tests Big Data. Nous allons devoir trouver des données qui répondent à nos objectifs. Nous allons devoir générer, marquer, modifier et amorcer des données afin de pouvoir suivre leur utilisation. Nous aurons besoin d'outils pour superviser l'utilisation des données marquées, ainsi que de la capacité à rapprocher les données de toutes origines : collecte, stockage, utilisation et suppression. Nous aurons besoin de nouveaux outils de visualisation des tests pour supporter les diagnostics et le débogage.

Le volume joue aussi un rôle. Que les tests spécifiques soient importants ou non, nous passerons beaucoup de temps à traiter des résultats, des visualisations et des prises de décision à grande échelle.

## Environnements de test, tests sur le terrain

Dans un cas simple, un laboratoire de test pour un produit de gestion d'un environnement domestique peut être configuré dans un bureau quelconque, car la portée du réseau local se réduit à un foyer unique. Dans le cas d'un système de gestion d'un environnement urbain qui supervise les niveaux de pollution de l'air à l'échelle d'une ville, par exemple, le relevé des données des capteurs peut être simulé en laboratoire. Il serait toutefois logique de piloter le service dans un environnement urbain réel afin de calibrer les capteurs ainsi que les processus d'agrégation et d'intégration des données pour obtenir des visualisations de données pertinentes.

## Outils de support

Même si les tests manuels seront toujours nécessaires et réalisables, une proportion bien plus importante de tests que celle dont nous avons l'habitude devra être exécutée par des outils, qui devront traiter d'énormes quantités de tests. Le défi n'est pas tant d'avoir besoin d'outils pour exécuter les tests, mais plutôt de savoir « comment concevoir les centaines, milliers ou millions de tests dont nous avons besoin pour alimenter les outils ? »

Les appareils qui font leur apparition aujourd'hui ont tous leurs nuances et leurs complications et seront confrontés à des événements imprévus. Même un système simple peut être confronté à des milliers de scénarios. Les systèmes deviennent de plus en plus complexes et doivent être testés. La situation est à sens unique.

## Analyse de tests, visualisation et prise de décisions

Je développerai la question de l'analyse de tests dans un autre article de cette série. Si vos systèmes de production sont des plates-formes d'expérimentation, alors le code qui sert aux équipes marketing peut aussi servir aux développeurs et aux testeurs. Envisagez votre code analytique comme votre réseau de capteurs. Les processus DevOps sont eux aussi des objets.

## Tests de performance et données de test

Les tests de performance dans un univers IoT ne sont pas très différents de ceux que vous connaissez déjà. Le plus compliqué, c'est que les données relevées par les capteurs doivent être cohérentes. Par exemple, pour être pertinents, les messages reçus par les voitures dans une ville doivent être mis en correspondance avec une localisation physique. Un jeu de coordonnées de localisation généré au hasard ne pourra pas faire l'affaire. Nous aurons besoin d'ensembles de données sûrs, issus d'opérations ou d'utilitaires réels, capables de générer des données pertinentes et fiables pour les tests.

## L'avenir des tests et des testeurs

J'ai la conviction que pour faire de l'IoE une réalité, des niveaux élevés d'automatisation des tests seront nécessaires. L'automatisation ne facilitera pas les tests, mais elle les rendra possibles.

L'automatisation des tests dans un environnement DevOps représente une source de données d'analyse au même titre que les systèmes de production, de sorte que les techniques et outils d'analyse constituent un autre domaine en développement.

Les testeurs doivent-ils apprendre à rédiger du code ? La réponse est simple. Oui. Il est possible que votre fonction ne le nécessite pas, mais la tendance aux États-Unis et en Europe est aux offres d'emploi combinant compétences en codage et autres aptitudes techniques. De plus en plus, vous devrez mettre au point vos propres utilitaires, télécharger, configurer et adapter des outils Open Source ou créer des tests automatisés.

L'IoT et les tendances numériques déplacent les tests plus en amont. Il semble que toutes les entreprises cherchent à adopter cette approche. Les activités liées aux tests, ou plutôt la partie réflexion des tests est avancée dans le processus de développement. Je vous conseille de suivre cette approche.

## Résumé

L'IoT accroît l'échelle, la diversité et la complexité des tests.

Les tests concernant les composants de bas niveau ou les sous-systèmes restent globalement inchangés. Toutefois, au niveau du système, nous aurons besoin de méthodes de simulation et d'automatisation des tests à grande échelle. Les outils nécessaires n'existent pas forcément à l'heure actuelle, donc vous devrez peut-être créer les vôtres en attendant que les fournisseurs rattrapent leur retard.

L'automatisation des tests à grande échelle requiert des modèles de test, des générateurs de données de test et des oracles automatiques. La modélisation, la simulation, l'analyse, la visualisation et la prise de décision assistée par des outils deviendront des compétences importantes pour les architectes et les équipes de test. Les testeurs devront apprendre comment créer de meilleurs modèles de test et comment les utiliser avec des outils de modélisation et de simulation plus techniques.

La création d'environnements de test fiables et de données de test pertinentes sera (comme toujours) un véritable casse-tête.

Des environnements de test à grande échelle en laboratoire et sur le terrain seront nécessaires et les frontières entre l'expérimentation en production et les tests en laboratoire deviendront floues. L'analyse de tests issue des processus DevOps deviendra une discipline critique pour les tests dans l'IoT.

---

## À propos de l'auteur

Paul Gerrard est consultant, enseignant, auteur, webmaster, développeur, testeur, conférencier, entraîneur d'aviron et éditeur. Il a assuré de nombreuses missions de conseils concernant tous les aspects des phases de test et d'assurance qualité logiciels, et s'est spécialisé dans l'assurance qualité des tests. Il a animé des présentations et des didacticiels lors de nombreuses conférences sur le thème des tests, en Europe, aux États-Unis, en Australie et en Afrique du Sud, et a été récompensé à plusieurs reprises.

Diplômé des universités d'Oxford et de l'Imperial College London, Paul a été lauréat en 2010 du prix d'excellence Eurostar European Testing, et en 2013 du prix The European Software Testing Awards (TESTA) Lifetime Achievement.

En 2002, Paul a coécrit l'ouvrage « Risk-Based E-Business Testing » avec Neil Thompson. En 2009, il a rédigé « The Tester's Pocketbook ». En 2011, il a cosigné « The Business Story Pocketbook » avec Susan Windsor et a rédigé « Lean Python » en 2014.

Cette année-là, Paul a également été président de programme à la conférence EuroSTAR de Dublin.

Il est directeur de Gerrard Consulting Limited, directeur de TestOpera Limited et hôte du Test Management Forum.

Courriel : [paul@gerrardconsulting.com](mailto:paul@gerrardconsulting.com)

Twitter : [@paul\\_gerrard](https://twitter.com/paul_gerrard)

Site Internet : [gerrardconsulting.com](http://gerrardconsulting.com)

Pour plus d'informations, rendez-vous sur la page **Développement et test** de CA Technologies.



Restez connecté à CA Technologies sur [ca.com/fr](http://ca.com/fr)



CA Technologies (NASDAQ : CA) fournit les logiciels qui aident les entreprises à opérer leur transformation numérique. Dans tous les secteurs, les modèles économiques des entreprises sont redéfinis par les applications. Partout, une application sert d'interface entre une entreprise et un utilisateur. CA Technologies aide ces entreprises à saisir les opportunités créées par cette révolution numérique et à naviguer dans « l'Économie des applications ». Grâce à ses logiciels pour planifier, développer, gérer les performances et la sécurité des applications, CA Technologies aide ainsi ces entreprises à devenir plus productives, à offrir une meilleure qualité d'expérience à leurs utilisateurs et leur ouvre de nouveaux relais de croissance et de compétitivité sur tous les environnements : mobile, Cloud, distribué ou mainframe. Pour plus d'informations, rendez-vous sur [ca.com/fr](http://ca.com/fr).

#### Références

- 1 « The Internet of Everything—What is it and how will it affect you? », Paul Gerrard, <http://gerrardconsulting.com/sites/default/files/IOEWhatIsIt2.pdf>
- 2 « Internet of Everything—Architecture and Risks », Paul Gerrard, <http://gerrardconsulting.com/sites/default/files/IOEArchitectureRisks.pdf>
- 3 « Internet of Everything—Test Strategy », Paul Gerrard, <http://gerrardconsulting.com/sites/default/files/IOETestStrategy.pdf>
- 4 « An Overview of High Volume Automated Testing », Cem Kaner, <http://kaner.com/?p=278>