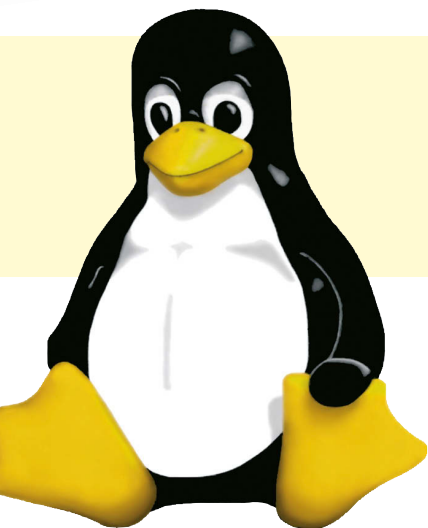


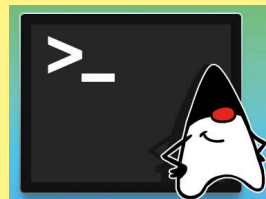


## Créer ses **balises HTML** avec les **composants Web**



Java 9

**Enfin un shell Java !**



**+**  **Windows 10**  
**= 2 systèmes en 1**

Comprendre et utiliser Windows Subsystem for Linux



**Azure** à la maison ?

C'est possible avec **Azure Stack**

**Ecrire un serveur  
REST Web API en C++**

Construire sa **radio FM**

**MSX** Le standard  
qui échoua

ikoula  
HÉBERGEUR CLOUD

PRÉSENTE

# CLOUD **IKOULA** ONE



Le succès est votre prochaine destination

MIAMI SINGAPOUR PARIS  
AMSTERDAM FRANCFORT — — —

CLOUD **IKOULA** ONE est une solution de Cloud public, privé et hybride qui vous permet de déployer **en 1 clic et en moins de 30 secondes** des machines virtuelles à travers le monde sur des infrastructures SSD haute performance.



[www.ikoula.com](http://www.ikoula.com)



[sales@ikoula.com](mailto:sales@ikoula.com)



01 84 01 02 50

ikoula  
HÉBERGEUR CLOUD



NOM DE DOMAINE | HÉBERGEMENT WEB | SERVEUR VPS | SERVEUR DÉDIÉ | CLOUD PUBLIC | MESSAGERIE | STOCKAGE | CERTIFICATS SSL

## Que reste-t-il à la Science-Fiction ?

La technologie aurait-elle tout volé à la Science-Fiction ? Quand on évoque les technologies dans Star Trek, effectivement, il reste peu de choses à créer mais pas des moindres : le synthétiseur de nourriture, la vitesse à distorsion et la téléportation. Nous n'avons pas encore les Cylons modèle "toaster" de Battlestar Galactica, mais la robotique avance très vite. Un T800, Terminator est encore loin d'exister, mais il ne relève plus de la seule science-fiction.

Un ordinateur capable de penser et d'agir comme dans 2001 ? La réalité n'est pas pour demain mais l'Intelligence Artificielle fait d'énormes progrès et les recherches autour de l'informatique quantique sont des éléments cruciaux pour aller dans cette direction. Bien sûr, nous ne sommes pas dans la matrice dans laquelle les machines règnent alors que l'homme n'est finalement que son carburant.






Nous rêvions dans Cosmos 1999 d'une station lunaire. Il existait des espoirs dans les années 70 et 80 mais ils furent rapidement stoppés. Fin septembre, Etats-Unis et Russie ont annoncé une volonté de travailler ensemble sur l'installation et l'exploitation d'une base lunaire permanente. La Lune comme tremplin vers d'autres planètes n'est pas une idée nouvelle mais peut être une étape essentielle pour l'avenir de l'Homme qui n'est peut-être pas sur Terre, surtout à la vitesse où nous consommons et détruisons les ressources. Nous ne parlons pas encore de Terra Formation mais dans 100 ou 150 ans, ce sera une autre histoire. Regardez l'incroyable révolution informatique depuis les missions Apollo (voir notre dossier dans Programmez! 210) et l'informatique actuelle.

Mais nous pouvons aussi voir l'autre côté du miroir. La science-fiction de type Dark City, Matrix, Blade Runner, pour ne citer qu'elle, nous montre aussi un monde technologique néfaste dans lequel une poignée de sociétés contrôlent le monde. Est-on aussi éloigné de cette réalité ? Le pouvoir aux peuples aurait-il encore un sens dans 50 ou 100 ans ? Mr Robot serait-il une autre vision de ce monde ?

Mais restons dans notre monde actuel. Ce mois-ci, beaucoup de bons codes pour vous, notamment sur Android et Java. Et nous explorerons, dans le détail Azure Stack, un Microsoft Azure mais chez vous !

**François Tonic**  
ftonic@programmez.com



<b>Tableau de bord</b> 4	<b>Agenda</b> 6
<b>ABONNEZ-VOUS !</b> 11	<b>Matériel</b> 8
 <b>Retour terrain</b> 14	 <b>Joel Polsky, fondateur de Stackoverflow</b> 15
<b>Recrutement</b> 17	<b>Dossier eCommerce</b> Partie 2 18
 <b>Azure Stack</b> Partie 1 29	
<b>Co-innovation, co-clustering</b> 50	 <b>Web Components</b> 39
 <b>Windows 10</b>	 <b>Images Docker</b> 56
 <b>Linux + Windows</b> 47	
 <b>ReactiveX avec RxJava</b> 65	<b>Algos génétiques Partie 2</b> 60
<b>Maker</b> 68	<b>MSX</b> 75
<b>JShell</b> 69	<b>Commitstrip</b> 82
<b>C++</b> 72	

**Dans le prochain numéro !  
Programmez! #213, dès le 2 décembre 2017**

**Programmation**

Les 10 erreurs les plus courantes en Java

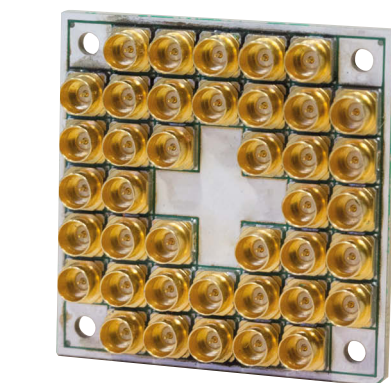
**Défi Jedi**

Créer son propre IDE en C++ avec la librairie Scintilla Partie 1

# L'informatique quantique : avenir es-tu là ?

Depuis quelques mois, une certaine frénésie agite le milieu technologique avec l'informatique quantique. Plusieurs annonces ont été faites notamment par Microsoft et Intel. Mais d'autres acteurs sont actifs, comme Google. IBM a annoncé un projet ambitieux : une puce de 17 qubits. Cette puce expérimentale est un pas supplémentaire vers une future informatique quantique. Intel travaille avec QuTech pour tester et améliorer les algorithmes et les modèles quantiques. Le qubit est un bit quantique qui peut avoir plusieurs états simultanés ce qui n'est pas possible dans notre informatique actuelle basée sur le binaire pur et dur : 0 et 1. Un q-bit

ayant plusieurs états multiplie les possibilités de calculs et de traitements. De l'aveu même d'Intel, le qubit est une entité fragile et qu'il doit avoir un environnement parfaitement stable pour fonctionner. Mais Intel n'est pas pour autant le chantre de l'informatique quantique pour tout le monde. Elle sera limitée à des domaines très précis : IA, big data, etc. Mais c'est en tout cas une étape très intéressante vers des processeurs quantiques toujours plus puissants. Fin septembre, Microsoft parlait d'informatique quantique durant sa conférence Ignite. Pour l'éditeur, le quantique est un domaine de recherche comme l'IA ou la réalité mixte. Un



des objectifs de Microsoft est de créer un langage de programmation pour développer les programmes quantiques qui sera intégré directement à Visual Studio. Un simulateur de qubit a même été annoncé : jusqu'à 40 qubits sur Azure. Mais attention à la mémoire vive nécessaire. L'éditeur a l'ambition de construire un écosystème complet. Le sujet est pris très au sérieux par l'éditeur qui propose une vision large du quantique sur

son site web : <https://www.microsoft.com/en-us/quantum/default.aspx>  
Cela prouve que les constructeurs et les fournisseurs technologiques avancent sur le quantique et n'hésitent pas à investir en R&D et à en parler publiquement. La Chine n'hésite pas non plus à l'évoquer.

## Ataribox

devrait voir le jour au printemps 2018.

"the future of coding is no coding at all" / le grand patron de GitHub.

Vous voulez savoir comment fonctionne

## Hey Siri,

un post très complet par les équipes d'Apple : <https://machinelearning.apple.com/2017/10/01/hey-siri.html>

## Paris – New York en 30 minutes :

Elon Musk voudrait le proposer avec une fusée... Même en rajoutant le temps pour embarquer et trouver le spaceport, c'est toujours plus intéressant que l'avion...

## Windows Mobile

est mort : pas de nouvelles fonctions, pas de matériels, juste des patchs pour les bugs et la sécurité.

## iOS 11 devance iOS 10.

## AutoML :

Google veut que le système code lui-même d'autres IA et machine learning sans intervention humaine.

**Alibaba**, le prochain géant du cloud ? Beaucoup d'ambitions sur l'IA, la blockchain, l'IoT. Le Chinois cible la Chine et l'Asie.

## Rémunérations 2018 selon PageGroup

Le groupe PageGroup a sorti son étude 2018 sur les rémunérations. Selon le document, en moyenne un candidat reçoit 3 offres d'embauches. La demande sur les profils et postes en sécurité est forte cette année et la tendance devrait se poursuivre.

Plus de la moitié des profils connaissent des hausses : développeurs, profils orientés data, consultants et urbanistes. Ces hausses concernent surtout les profils à partir de 2 ans d'expérience. Les profils débutants restent très tendus en bac+2, à partir de 23-28 k€. Pour le reste, peu de surprises.

## INDEX TIOBE DU MOIS

10/2017	10/2016	Change	Langage	%	Evolution en %
1	1	=	Java	12.431%	-6.37%
2	2	=	C	8.374%	-1.46%
3	3	=	C++	5.007%	-0.79%
4	4	=	C#	3.858%	-0.51%
5	5	=	Python	3.803%	+0.03%
6	6	=	JavaScript	3.010%	+0.26%
7	7	=	PHP	2.790%	+0.05%
8	8	=	VB.NET	2.735%	+0.08%
9	11	▲	Assembleur	2.374%	+0.14%
10	13	▲	Ruby	2.324%	+0.32%

Très peu de changements. On constate juste le retour de Ruby et de l'assembleur dans le top 10. Swift chute assez lourdement et pointe à la 16e place.

## Countries With the Best Developers By Domain

(Ranked By Average Score)

Algorithms		Java		Data Structures	
Rank	Country	Rank	Country	Rank	Country
1	Russia	1	Poland	1	China
2	Poland	2	Bulgaria	2	Taiwan
3	China	3	Hungary	3	Philippines
4	Switzerland	4	Switzerland	4	Russia
5	Taiwan	5	Russia	5	Japan
C++		Tutorials		Mathematics	
Rank	Country	Rank	Country	Rank	Country
1	France	1	Hungary	1	China
2	Russia	2	Italy	2	Czech Republic
3	Hungary	3	Switzerland	3	Finland
4	Italy	4	Poland	4	Ukraine
5	Switzerland	5	Sweden	5	Japan
Python		SQL		Shell	
Rank	Country	Rank	Country	Rank	Country
1	Hong Kong	1	Denmark	1	Czech Republic
2	Poland	2	Finland	2	Hong Kong
3	Bulgaria	3	Switzerland	3	Hungary
4	China	4	Russia	4	Poland
5	Russia	5	Hong Kong	5	China
Artificial Intelligence		Functional Programming		Databases	
Rank	Country	Rank	Country	Rank	Country
1	Japan	1	China	1	Switzerland
2	Belgium	2	Switzerland	2	Italy
3	Vietnam	3	Russia	3	Denmark
4	Russia	4	Japan	4	Taiwan
5	Ireland	5	Taiwan	5	Hungary
Ruby		Distributed Systems		Security	
Rank	Country	Rank	Country	Rank	Country
1	Finland	1	Sri Lanka	1	Ukraine
2	Nigeria	2	Malaysia	2	China
3	Switzerland	3	Pakistan	3	Switzerland
4	France	4	Ukraine	4	Czech Republic
5	Poland	5	Finland	5	Colombia

SÉMINAIRE DÉVELOPPEMENT  
WINDOWS WEB LINUX ANDROID IOS  
DÉVELOPPEZ 10 FOIS PLUS VITE

# WINDEV 23

## Tour

### C'EST PARTI !

#### VOUS ÊTES INVITÉ

35

SUJETS

pour découvrir les  
923 nouveautés de la  
version 23.

DSI

Développeurs

WebDesigners

Architectes logiciel

Ingénieurs

Start-ups...

#### INSCRIVEZ-VOUS VITE !

MONTPELLIER	mardi 7 Novembre
TOULOUSE	mardi 14 Novembre
BORDEAUX	mercredi 15 Nov.
NANTES	jeudi 16 Novembre
PARIS	mardi 21 Nov.
LILLE	mercredi 22 Nov.
BRUXELLES	jeudi 23 Novembre
STRASBOURG	mardi 28 Novembre
GENÈVE	mercredi 29 Nov.
LYON	jeudi 30 Novembre
MARSEILLE	mardi 5 Décembre
MONTREAL	jeudi 7 Décembre

GRATUIT

13h45 à 17h45

10.000 places seulement.  
Réservation nécessaire

923  
nouveautés

DÉCOUVREZ LES  
923 NOUVEAUTÉS  
de WINDEV 23, WEBDEV 23  
et WINDEV Mobile 23

WINDEV  
VU À LA TÉLÉ  
EN 2017  
SUR TF1, M6,  
BFM TV, FRANCE 2,  
CANAL+



PCSOFT

WWW.PCSOFT.FR

## novembre

**NI DAYS PARIS** 7 novembre / Paris

National Instruments est un constructeur / éditeur phare du monde de l'embarqué et de l'industrie. Chaque année, la journée NI Days permet de découvrir les dernières versions des outils d'instrumentation, l'écosystème. De nombreuses sessions sont jouées durant la journée.

Pour en savoir plus : <https://www.ni.com/nidays/>

**OPENSTACK DAY FRANCE**

21 novembre / Paris



OpenStack Day France sera l'opportunité de rencontrer des représentants de la Fondation OpenStack, les acteurs de l'écosystème du cloud en Open Source, les utilisateurs finaux et les contributeurs de la communauté de développeurs en France. OpenStack Day France est organisé par l'Association OpenStack-FR en collaboration avec Osones et Red Hat. L'évènement compte déjà Osones et

Objectif Libre comme sponsors. Pour en savoir plus : [openstackdayfrance.fr](http://openstackdayfrance.fr)

**DEV DAY**

28 novembre / Belgique

La 5ème édition du Dev Day, qui aura lieu à Mons (Belgique) le 28 novembre 2017, proposera aux participants des conférences sur les technologies modernes, tous horizons confondus. De nos jours, peu importe que l'on soit développeur C#, PHP ou JavaScript ! Les concepts de programmation sont transversaux et les technologies de plus en plus ouvertes. Le Dev Day s'adresse à tous les développeurs. Que vous souhaitiez développer une application mobile, intégrer de l'intelligence artificielle, déployer et scaler des micro-services en quelques secondes, ou exposer un service via une API GraphQL, cette journée est faite pour vous !

**DOTJS**

30 novembre &amp; 1er décembre / Paris

La grande conférence dotJS revient à Paris fin novembre. A l'heure où nous imprimons le détail de l'agenda n'est pas connu mais on aura des sessions de très bons niveaux techniques comme chaque année. ParisJS propose le 29 novembre un meetup dotJS

## décembre

**PARIS OPEN SOURCE SUMMIT**

6 &amp; 7 décembre / Paris

Paris Open Source Summit, 1er événement européen libre et open source, est le fruit de la fusion de Solutions Linux et de l'Open World Forum, deux événements emblématiques du Libre et de l'Open Source. L'objectif est d'exposer les innovations technologiques, la réalité et le dynamisme économique de ses solutions ainsi que les impacts sociétaux de cette filière

re numérique. Le 6 décembre aura lieu le Student DemoCamp. Articulée autour de l'idée d'innovation ouverte emmenée par l'Open Source et l'Open Data, la SDC 2017 récompensera les meilleurs projets dans 3 catégories : Ecosystème du numérique ouvert (Open Source, Open Data, Open Gov, Open Robot, Open IOT, IA ...);

- Technologies et Devops ;
- Business et Start Up.

Paris Open Source Summit garde la particularité de solutions linux en proposant de nouveau un village associatif qui sera composé des associations majeures du secteur. Cette année des villages de regroupement d'entreprises seront proposés à tous nos partenaires.

## Communautés

**PARISJUG ANNONCE :**

14/11/17 - Soirée Qualité logiciel

12/12/17 - Soirée Vertx

09/01/18 - YoungBlood V

**GDG Toulouse propose le 6 décembre prochain du langage Go :**

Réaliser ses WebServices en Go par Patrice Trognon. Comment réaliser ses services web REST/JSON en Go, un exemple complet sera présenté avec gestion d'une base de données... Lien : <https://www.meetup.com/fr-FR/GDG-Toulouse/>

**.Net Toulouse propose le 20 novembre : serverless Azure**

Session proposée et animée par Samir AREZKI. Dans cette session, nous discuterons l'intérêt des architectures serverless.

Lien : <https://www.meetup.com/fr-FR/Meetup-NET-Toulouse/>

**Meetup Delphi & C++ Builder Paris**

Un meetup autour de Delphi et de C++ Builder est proposé sur Paris le 9 novembre avec la présentation des version Tokyo. Il sera aussi question de l'application FMX avec le système de bases de données SQLite, les techniques de sérialisation et de stockage d'objets métiers. Elle est centrée sur le développement d'applications 2D/3D comme cette calculatrice opérationnelle ou cette main animée.

Lien : <https://www.meetup.com/fr-FR/Paris-Delphi-et-C-Builder-Meetup/events/244172606/>

## 2018

**MakemeFest Angers revient en avril !**

Le nouveau salon maker reviendra en avril à Angers. L'évènement 2017 avait réuni plus de 70 startups et makers et plus de 3000 visiteurs. Pourquoi pas vous ?

Site officiel : <http://makeme.fr>



# SERVEURS DÉDIÉS XEON®

AVEC

**ikoula**  
HÉBERGEUR CLOUD

Optez pour un serveur dédié dernière génération et bénéficiez d'un support technique expérimenté.

 **debian**  **ubuntu**  **CentOS**  **Windows Server 2012**



POUR LES LECTEURS DE  
**PROGRAMMEZ\***

**OFFRE SPÉCIALE -60 %**  
À PARTIR DE

**11,99€**  
HT/MOIS

~~29,99€~~

CODE PROMO  
**XEPRO17**

✓ **Assistance technique  
en 24/7**

✓ **Interface Extranet**  
pour gérer vos prestations

✓ **KVM sur IP**  
pour garder l'accès

✓ **Analyse et surveillance  
de vos serveurs**

✓ **RAID Matériel**  
en option

✓ **Large choix d'OS**  
Linux et Windows

\*Offre spéciale -60 % valable sur la première période de souscription avec un engagement de 1 ou 3 mois. Offre valable jusqu'au 31 décembre 2017 23h59 pour une seule personne physique ou morale, et non cumulable avec d'autres remises. Prix TTC 14,39 €. Par défaut les prix TTC affichés incluent la TVA française en vigueur.

**CHOISISSEZ VOTRE XEON®**

<https://express.ikoula.com/promoxeon-pro>



**ikoula**  
HÉBERGEUR CLOUD

 /ikoula

 @ikoula

 sales@ikoula.com

 01 84 01 02 50

NOM DE DOMAINE | HÉBERGEMENT WEB | SERVEUR VPS | SERVEUR DÉDIÉ | CLOUD PUBLIC | MESSAGERIE | STOCKAGE | CERTIFICATS SSL

# Adafruit Circuit Playground

*Cette carte de prototypage rapide ressemble aux plateformes Lilypad et Flora. La destination première n'est pas forcément le wearable comme les autres cartes, mais elle peut parfaitement convenir par les dimensions et les capteurs embarqués.*

Contrairement aux autres cartes du même genre, la Circuit Playground embarque différents capteurs directement intégrés. Il s'agit d'une carte idéale pour découvrir en douceur l'électronique et les premiers montages. Le prix reste relativement raisonnable : -20 \$. Malheureusement en France, l'objet est un peu cher : env. 24 €.

## Une carte complète

La carte utilise le microcontrôleur Atmega32u4 fonctionnant à 8 Mhz. Il dispose d'un connecteur microUSB. Il supporte USB HID signifiant qu'il peut se comporter comme une souris ou clavier quand il est connecté à un ordinateur. On dispose de :

- 1 mini NeoPixel de 10 leds RGB ;
- 1 accéléromètre (dit capteur de mouvement) 3 axes (XYZ) de type LIS3DH ;
- 1 capteur de température ;
- 1 capteur de luminosité (photorésistance) ;
- 1 capteur son (microphone) ;
- 1 buzzer ;
- 2 boutons ;
- 1 interrupteur ;
- 8 broches compatibles pinces crocodiles (I2C, UART, 4 broches analogiques / PWM en sortie) ;
- 1 connecteur pour batterie externe ;
- LEDS rouge et verte (led 13 et fonctionnement de la carte) ;
- 1 bouton RESET (toujours utile...) ;
- 1 port série matériel : TX + RX ;
- 2 broches GND et 2 broches 3,3V ;
- 1 broche VBATT.

A noter que 8 petites zones sont des zones tactiles. Chaque zone possède une résistance de 1 Mohm. Elles sont marquées : #0, #1, #2, #3, #6, #9, #10, #12. Ce qui peut permettre des interactions sympathiques.

La NeoPixel est la seconde couronne, vers l'intérieur du circuit. Elle utilise une puce SK6812-3535. Elle est branchée sur la broche 17 et fonctionne en 3,3V. L'accéléromètre n'est pas connecté en I2C mais en SPI. La carte est donc très compacte pour contenir l'ensemble des pins, capteurs et leds mais la fabrication est de qualité, et chaque élément est bien identifié par son numéro (#x) et son usage. Bien entendu, la carte est compatible Arduino et Arduino IDE.

Sur Windows, il faudra installer les pilotes

Adafruit, avec connexion de la carte :

[https://github.com/adafruit/Adafruit\\_Windows\\_Drivers/releases/download/2.0.0.0/adafruit\\_drivers\\_2.0.0.0.exe](https://github.com/adafruit/Adafruit_Windows_Drivers/releases/download/2.0.0.0/adafruit_drivers_2.0.0.0.exe)

Sur Linux et macOS, en principe, vous n'aurez rien à faire. Côté Arduino IDE, mieux vaut avoir une version récente, minimum 1.6.4 selon le constructeur. Dans les préférences, indiquez le chemin : [https://adafruit.github.io/arduino-board-index/package\\_adafruit\\_index.json](https://adafruit.github.io/arduino-board-index/package_adafruit_index.json)

dans le champ Additional boards manager URL. Cliquez sur OK. Puis dans le menu Tools -> Boards -> Boards Manager, installez les Adafruit AVR Boards (sélectionnez l'option Contributed dans la liste déroulante dans l'option type). Vous pouvez aussi installer un IDE contenant par défaut les cartes Adafruit mais le constructeur ne le recommande pas.

Voilà ! Maintenant, si tout est ok, la Circuit Playground est disponible dans la liste des boards. On peut maintenant connecter la carte. Pour charger du code dans la Circuit Playground, vérifier dans le menu Boards :

- Boards : Adafruit Circuit Playground (attention : ne sélectionnez pas la version Express) ;
- Ports : vérifier que le port est bien sélectionné, sur mon poste, COM4 ;
- Programmez : USBtinyISP.

Pour exploiter les capteurs de la carte, il faut installer la librairie Adafruit\_CircuitPlayground. En principe, vous pouvez le faire directement depuis la fenêtre Manage librairies en tapant Circuit Playground et en faisant l'installation automatique. Sinon, il faut télécharger la librairie et l'installer manuellement :

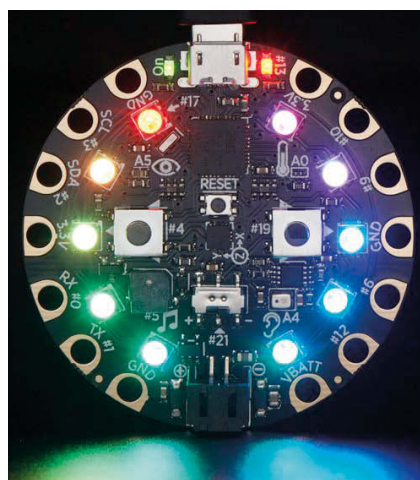
[https://github.com/adafruit/Adafruit\\_CircuitPlayground](https://github.com/adafruit/Adafruit_CircuitPlayground)

Tout est prêt. De nombreux exemples de codes sont disponibles avec la librairie, cela permet de comprendre comment fonctionne chaque capteur et composant.

## Quelques tests

Rien ne vaut le test réel pour jauger une carte. Nous commençons par les zones tactiles. Le code est très simple :

1. On n'oublie pas de faire un joli `#include <Adafruit_CircuitPlayground.h>`
2. On initialise la librairie dans le void setup : `CircuitPlayground.begin();`
3. Dans le void loop, on peut par exemple faire : `Serialln(CircuitPlayground.readCap(1));`



Le 1 représente la zone tactile #1 de la carte. Vous pouvez choisir la zone que vous souhaitez. L'outil Serial Plotter de l'IDE permet de voir le comportement et quand la zone est touchée ou non. Si on va plus loin, on peut créer son propre Makey Makey, une carte qui permet d'utiliser tout et n'importe quoi (ou presque) pour interagir. Les kits sont vendus à partir de 49 €. Là, avec la Circuit Playground on peut faire la même chose. L'exemple d'Adafruit est très intéressant : <https://learn.adafruit.com/circuit-playground-fruit-drums/overview>

L'accéléromètre n'est pas toujours un capteur facile à maîtriser au niveau de l'usage et du code. Le capteur détecte jusqu'à 8g. Il est connecté en SPI. Qui dit accéléromètre 3 axes dit mouvements en X, Y et Z. La lecture des axes se fait avec `CircuitPlayground.motion`. Il rajoute à la fois l'axe. La valeur retournée est une valeur en virgule flottante. Il est possible de limiter les g détectés en utilisant le `setAccelRange`. Il est suffisamment sensible pour détecter les chocs ; utile pour créer des interactions / actions quand la carte est embarquée dans un objet, un jouet. Le capteur détecte aussi l'action de secouer la carte. Côté microphone, il est présent mais pour nous, il manque de sensibilité. A voir ce que cela donne en conditions réelles. Le capteur de température est quant à lui assez sensible. Sa lecture est ultra facile avec `temperature()`. Par défaut, la valeur est donnée en C°. Il suffit de mettre `temperatureF0` pour avoir le Fahrenheit. Pour le prix, la Circuit PlayGround est vraiment excellente ! Le tout-en-un est une bonne idée pour démarrer avec quelques capteurs. Il y a suffisamment de broches pour étendre les capacités même si on passe principalement par des pinces crocodiles et non de la soudure. La librairie est plutôt bien faite.

Après il faut aimer le tout-en-un et les limitations que cela donne pour les objets connectés que l'on monte avec cette carte. Côté documentation, Adafruit propose des exemples et une documentation de base mais comme souvent avec ce genre de cartes, la documentation disponible est limitée.

ENABLING  
DIGITAL  
EVERYWHERE

[opensourcesummit.paris](http://opensourcesummit.paris)

#OSSPARIS17

1<sup>ER</sup> ÉVÉNEMENT EUROPÉEN  
LIBRE & OPEN SOURCE



# PARIS OPEN SOURCE SUMMIT

6 & 7  
DÉCEMBRE  
2017

DOCK PULLMAN

Email : [cjardon@weyou-group.com](mailto:cjardon@weyou-group.com) – Tel : 01 41 18 60 52

SPONSOR DIAMOND



SPONSORS PLATINUM

alter way  Microsoft

**Smile**  
OPEN SOURCE SOLUTIONS

SPONSORS GOLD

 BlueMind  henix  IBM

 redhat.  Sensio Labs  
Créateur de  Symfony

SPONSORS SILVER

 axelar  centreon  cozy.io

 AXELIT  ZABBIX  devoteam  XIVO

ORGANISÉ PAR



EN PARTENARIAT AVEC



# Orange Pi Zero : une alternative à la Pi Zero (vraiment) disponible !

**V**ous ragez de ne pas trouver facilement la Pi Zero ou la Pi Zero W et surtout vous ne pouvez en acheter qu'une à la fois ! L'avantage de l'Orange Pi Zero est sa disponibilité et surtout de pouvoir l'acheter par conteneurs entiers.

Avertissement : le tarif officiel de la carte est de 6,99 \$ en version 256 Mo de RAM et 8,99 \$ pour la version 512 Mo de RAM. Attention tout de même aux frais de port depuis la Chine. En France, on trouve la carte -/+ 20 €. Mais en cherchant un peu, on peut la trouver moins chère.

## Orange Pi Zero vs Raspberry Pi Zero / Zero W

Orange Pi n'est pas un inconnu dans le monde des cartes et la Pi Zero n'est qu'une évolution des grandes cartes du constructeur.

	Orange Pi Zero	Pi Zero	Pi Zero W
Processeur	Allwinner H2+ 4 cœurs Cortex A7 1,2 Ghz	BCM2835 Arm 1 Ghz	BCM2835 1 Ghz
RAM (Mo)	256 ou 512	512	512
USB	1 micro-usb 1 USB classique	2 micro-usb	2 micro-usb
HDMI / VGA	-	mini HDMI	mini HDMI
Stockage	SD	SD	SD
GPIO	2 rangées dont 1 soudées	40 (non soudés)	40 (non soudés)
Ethernet	10/100	-	-
WiFi	Oui	Non	Oui + Bluetooth
Poids	26g	9g	9g
Dimension	48 x 46 mm	65 x 30 mm	65 x 30 mm
Prix	6,99 \$ (256 Mo RAM) 8,99 \$ (512 Mo RAM)	5 €	11 €

## Premier boot

Tout d'abord, il faut installer un système. Le plus classique : c'est une distribution Armbian. On décompresse l'archive et on flashe la carte SD. Du connu et reconnu. Pour se connecter à la carte, vous devez tout d'abord la relier en Ethernet et récupérer son adresse IP. Le plus simple est de la chercher via son routeur. On ouvre une session SSH classique. Le mot de passe par défaut est 1234 (en root). Le système vous demande de changer le mot de passe root et de créer un nouvel utilisateur. En quelques minutes, votre board est connectée.

Vous pouvez passer par une carte USB – Série mais on se complique inutilement la vie. Comme toujours, on fait de suite :

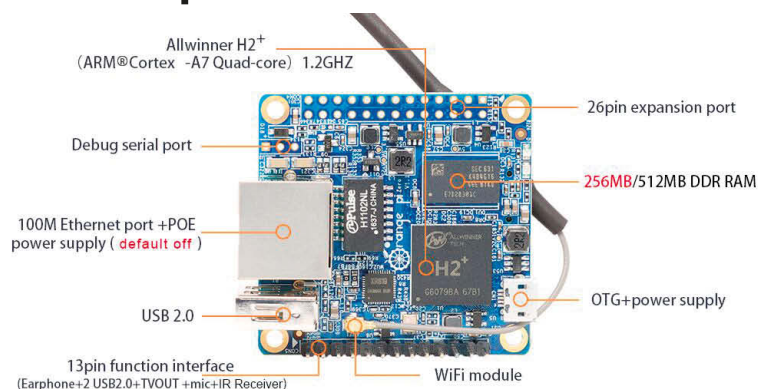
```
Sudo apt-get update
Sudo apt-get upgrade
```

La mise à jour peut prendre du temps si votre réseau est lent... Puis configurez la time zone avec `sudo dpkg-reconfigure tzdata`. La dernière étape est la configuration du WiFi :

```
nano /etc/network/interfaces
```

Vous devez remplacer les dernières lignes du fichier de configuration par :

```
allow-hotplug wlan0
iface wlan0 inet dhcp
wpa-ssid "orangezero"
wpa-psk "MOT_DE_PASSE_WIFI"
```



### Les +

- Prix
- Dimension
- Fabrication

### Les -

- Documentation pas toujours disponible
- Dégagement de chaleur
- OS parfois instable
- Pas de port vidéo

Remplacez le ssid et le mot de passe selon vos désirs. Enregistrez et quittez le fichier (control+X). Si vous faites un `ifup wlan0`, le WiFi s'active et il devient actif. Si tout va bien... Et vous le retrouvez logiquement si vous faites un `iwconfig wlan0`. Il est possible d'attribuer une ip fixe à votre Orange Pi Zero, surtout si vous vous en servez en serveur. Dans ce cas il faut configurer le fichier en utilisant `inet static` et non `inet dhcp` puis indiquer l'adresse, le netmask, la gateway... Le boot de la carte peut être long surtout si la SD est lente. Pour configurer les fonctions de base, vous pouvez utiliser l'outil graphique `sudo armbian-config`. L'outil est pratique pour installer quelques outils de bases comme Samba.

Pour connaître l'état de la carte (température, charge) : `armbianmonitor -m`

La documentation n'est pas toujours simple à trouver mais par rapport à d'autres cartes, on arrive à trouver. En serveur web, DLNA, ou domotique, la carte peut être une solution intéressante. Attention cependant à ne pas trop en demander, surtout en version 256 Mo de mémoire vive. Bien entendu, il existe d'autres distributions serveurs ou desktop : Ubuntu, Raspbian Server ou Android.

Quelques idées de projets autour des Orange Pi :

<https://diyprojects.io/mini-pc-orange-pi-raspberry-pi-board-computer/orange-pi-raspberry/#.WebqrxQ10cx>

## Et l'écran ?

Par défaut, la carte ne possède aucun port HDMI ou VGA, ni connecteur vidéo dédié. Le display n'est pas indispensable dans toutes les conditions d'utilisation mais si vous installez une distribution complète de type Ubuntu ou Android sur votre Orange Pi Zero, l'écran devient indispensable. On dispose d'une broche video-out sur les headers soudés. Il faut connecter un câble RCA sur les broches GND et video-out. Le montage est simple à réaliser.

Vidéo : <https://www.youtube.com/watch?v=KKn1G2TzCtM>

Vous pouvez tenter l'aventure en passant par les GPIO : <http://www.instructables.com/id/Orange-Pi-Zero-Connect-TFT-SPI-ST7735/>

Attention : ce montage est plus compliqué et il faudra une configuration spécifique.

# NE RATEZ AUCUN NUMÉRO

## Abonnez-vous !

# PROGRAMMEZ!

le magazine des développeurs

### Nos classiques

1 an ..... 49€\*

11 numéros

2 ans ..... 79€\*

22 numéros

Etudiant ..... 39€\*

1 an - 11 numéros \* Tarifs France métropolitaine

### Abonnement numérique

PDF ..... 35€

1 an - 11 numéros

Souscription uniquement sur  
[www.programmez.com](http://www.programmez.com)

Option : accès aux archives 10€

### Nos offres d'abonnements 2017

1 an ..... 59€

11 numéros + 1 vidéo ENI au choix :

• Arduino\*

Apprenez à programmer votre microcontrôleur

• jQuery\*

Maîtrisez les concepts de base



2 ans ..... 89€

22 numéros + 1 vidéo ENI au choix :

• Arduino\*

Apprenez à programmer votre microcontrôleur

• jQuery\*

Maîtrisez les concepts de base

Offre limitée à la France métropolitaine

\* Valeur de la vidéo : 34,99 €

Toutes nos offres sur [www.programmez.com](http://www.programmez.com)

# Oui, je m'abonne

ABONNEMENT à retourner avec votre règlement à :

Service Abonnements PROGRAMMEZ, 4 Rue de Mouchy, 60438 Noailles Cedex.

☐ Abonnement 1 an : 49 €

☐ Abonnement 2 ans : 79 €

☐ Abonnement 1 an Etudiant : 39 €  
Photocopie de la carte d'étudiant à joindre

☐ Abonnement 1 an : 59 €

11 numéros + 1 vidéo ENI au choix :

☐ Abonnement 2 ans : 89 €

22 numéros + 1 vidéo ENI au choix :

☐ Vidéo : Arduino

☐ Vidéo : jQuery

☐ Mme ☐ M. Entreprise : \_\_\_\_\_ Fonction : \_\_\_\_\_

Prénom : \_\_\_\_\_ Nom : \_\_\_\_\_

Adresse : \_\_\_\_\_

Code postal : \_\_\_\_\_ Ville : \_\_\_\_\_

email indispensable pour l'envoi d'informations relatives à votre abonnement

E-mail : \_\_\_\_\_ @ \_\_\_\_\_

☐ Je joins mon règlement par chèque à l'ordre de Programmez !

☐ Je souhaite régler à réception de facture

\* Tarifs France métropolitaine

# Optimiser la partie front

• **Matthieu OULOUHODJIAN**,  
intégrateur HTML  
**Osaxis**

*Qu'est-ce que la performance web ? Pour le dire simplement, la performance web correspond à l'optimisation du temps de chargement d'un site ou d'une application web.*

Certains critères sont importants à prendre en compte, parmi lesquels :

- La localisation de l'utilisateur ;
- La connexion de l'utilisateur ;
- Le temps de réponse du serveur ;
- La construction de la partie front.

C'est ce dernier point qui est abordé dans la suite de cet article, à savoir le front-end et son optimisation. [1]

Le chargement **T** d'un site est composé de différents temps : **T = T1 + T2 + T3 + T4** avec :  
**T1** : La requête envoyée au serveur (T1 ~ 0).  
**T2** : le traitement de la requête par le serveur.  
**T3** : l'envoi du résultat de la requête par le serveur.  
**T4** : le traitement de la réponse du serveur pour afficher le résultat.

## Quel sont les principaux impacts d'une mauvaise performance web ?

Attendre le chargement d'une page est quelque chose de frustrant. Passées les 2-3 secondes de chargement, l'attention de l'utilisateur est pratiquement perdue entraînant un risque fort de départ du site.

Le temps de chargement est pris en compte dans le référencement (SEO). En 2010, Google avait pénalisé des sites à cause d'une lenteur excessive <sup>(1)</sup>. D'un point de vue général, des problèmes de performances web ont un impact économique. Dare Boost résume dans son article <sup>(2)</sup> plusieurs études démontrant, par exemple, que parmi les acheteurs en ligne, 67% abandonnent leurs achats à cause d'une lenteur trop importante. Certains utilisateurs viendraient même à penser que cela impacte la sécurité du site utilisé, un temps de réponse trop important laissant à penser qu'un traitement non désiré s'exécute en parallèle.

## Optimiser le front-end en compressant le HTML, le CSS et le Javascript

La compression revient à enlever dans le code les commentaires, espaces, séparateurs et autres caractères inutiles. Avec un code plus léger, la bande passante est moins surchargée par la taille des fichiers et le site s'affiche plus rapidement.

Afin de le faire convenablement, des outils sont

déjà disponibles en ligne ou à installer sur le projet. On peut, par exemple, coder en SASS puis utiliser Compass avec Grunt.

Dans ce domaine, un des compresseurs les plus complets est Compress My Code car il compresse le HTML, le CSS et le Javascript, et possède plusieurs options comme la non suppression des commentaires. Il en existe d'autres, par exemple spécialisés dans le Javascript, comme JS Compress ou Javascript Minifier.

## Optimiser les images

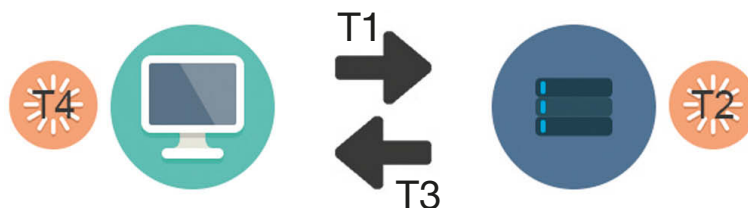
En 2014, les images constituaient, en moyenne, environ 60% du poids d'une page web <sup>(3)</sup>. Il faut donc les optimiser et réduire leur taille tout en gardant une qualité adéquate. Il est important de réussir à trouver un équilibre entre les deux; différents supports et logiciels existent afin de réduire le poids des images.

Le plus connu reste Adobe Photoshop qui peut exporter directement les images pour le web. Autrement, Kraken.io est un outil en ligne très efficace. Il est possible de souscrire à un abonnement gratuit mais pour en profiter au maximum et surtout souvent, car la limitation se fait en poids d'images, il faudra souscrire à un abonnement payant.

De plus, il faut rester vigilant sur le format de l'image, un JPG étant par exemple moins volumineux et donc à privilégier par rapport à un PNG, bien que ce dernier soit de meilleure qualité.

## Utiliser des outils d'analyse

Une fois le développement du site achevé, il est possible, grâce à certains outils, de simuler les performances de celui-ci. La rapidité de chargement, la qualité du code et même les optimisations d'images ou de compressions peuvent



être vérifiées pour améliorer l'expérience utilisateur. Il est recommandé de ne pas sauter cette étape pour analyser les développements effectués.

## PageSpeed Insight

PageSpeed Insight est un outil mis à disposition par Google. En premier lieu, il faut entrer l'URL du site à tester. Suite à cela, l'outil va effectuer son analyse et délivrer les améliorations possibles. Pour ce faire, il attribue une note sur 100, distincte entre l'expérience mobile et sur ordinateur. Ensuite une liste s'affiche dans un design très simple avec des solutions proposées pour chaque optimisation détectée, le tout accompagné d'un visuel de la page du site simulée.

Le site est évalué sur de nombreux points, notamment la taille des images, la compression des ressources CSS et Javascript ainsi que l'exploitation du cache. L'outil propose même de télécharger les images, scripts Javascript et ressources CSS optimisés pour la page testée.

PageSpeed possède également un module qui optimise automatiquement le site sur le serveur sur lequel il est installé. [2]

## Varvy

Varvy est un outil particulièrement bien conçu d'un point de vue ergonomique. Au départ, une fois le site chargé, il offre un visuel sur le nombre total de fichiers Javascript, CSS et d'images chargés ainsi qu'un aperçu de la page demandée. Sur le côté, on trouve une liste résumant les points pouvant être optimisés sur le site, répartis par catégorie serveur, page et ressources. En bas de page, la solution présente les optimisations détaillées. Différentes couleurs instinctives facilitent la compréhension de l'analyse, chaque fichier pouvant impacter les per-

(1) <https://webmasters.googleblog.com/2010/04/using-site-speed-in-web-search-ranking.html>

(2) <https://www.dareboost.com/fr/webperf-impacts>

(3) <https://blog.dareboost.com/fr/2014/07/performance-web-tendances>

Mobile

Ordinateur

Needs Work

75 / 100

Certaines des optimisations des performances les plus communes manquent à cette page, ce qui pourrait ralentir l'expérience utilisateur. Veuillez examiner les recommandations suivantes.

**Optimisations possibles**

- Exploiter la mise en cache du navigateur
- Éliminer les codes JavaScript et CSS qui bloquent l'affichage du contenu au-dessus de la ligne de flottaison
- Réduire la taille des ressources CSS
- Optimiser les images
- Réduire la taille des ressources JavaScript
- Réduire la taille des ressources HTML

**Optimisations trouvées**

Télécharger des images, des scripts JavaScript et des ressources CSS optimisés pour cette page.

2

Summary

Resources (20 js files / 9 images / 12 css files)

**Server:**

- Slow server response
- Compression enabled
- Browser caching issues
- Keep-alive enabled
- Minimal redirects

**Page:**

- No bad requests
- HTML not minified
- Request size is fine
- Visible content prioritized
- Render blocking CSS / JS found

**Resources:**

- Images not optimized
- JavaScripts seem async
- CSS not minified
- No @import CSS
- JavaScript not minified

Note: These values may change often if you display third party content (ads, widgets, etc.).

CSS delivery

JavaScript usage

CSS size: 124 k  
External: 123615 / Internal: 369 / In attribute: 211

Total Javascript size: 350 k  
External: 346406 / Internal: 3784

3

formances étant facilement identifiables. Par ailleurs, un lien redirigeant vers le site web de Varvy est proposé à la suite de chaque section détaillée, permettant ainsi d'être redirigé vers des articles correspondant à l'optimisation à apporter. [3]

## DareBoost

DareBoost est un outil assez complet. En premier lieu, il propose un tableau de bord synthétique avec notamment un pourcentage de qualité d'optimisation du site testé. Ensuite, il est appréciable de voir le chargement en image du site testé avec un aperçu de la page rafraîchie toutes les 100 millisecondes. Le détail des optimisations se base autant sur la compilation des fichiers que sur le SEO. En effet, DareBoost teste si des balises <h1> et <h2> sont présentes avec des mots clés ou même les surcharges CSS. Très complet sur l'analyse des sites, l'outil propose même une solution de résolution pour la globalité des optimisations détectées. Enfin, particularité appréciable, DareBoost offre également la possibilité de comparer deux sites entre eux. [4]

## Pingdom

Pingdom est un outil qui se révèle utile sur le long terme. Un essai gratuit de 14 jours est disponible mais par la suite, il faut souscrire à un abonnement payant. Pingdom offre un visuel sur toutes les performances des pages d'un site

en termes de chargement. Il permet surtout de comprendre comment est ressentie l'expérience utilisateur par les consommateurs du site. Il possède également son propre outil pour tester les performances directement. La simulation s'effectue depuis les serveurs situés en Suède, en Australie ou encore en Californie. Cette fonctionnalité peut se révéler utile dans le cadre de sites internationaux.

Comme d'autres, Pingdom affiche un aperçu de la page simulée ainsi qu'une note moyenne allant de A à F, A représentant la meilleure note. Cette moyenne est calculée en attribuant des notes à plusieurs critères tels que la taille des requêtes, le minimum de redirections ou encore l'utilisation du cache. A la suite de ces premiers critères, la solution présente le détail du pourcentage du poids de chaque type de fichier chargé par rapport au poids total de ceux-ci. Enfin, Pingdom édite un graphique

représentatif des requêtes des fichiers du site avec différents choix de filtre : ordre de chargement, temps de chargement, taille des fichiers, type de fichier et URL. [5]

## Et aussi...

Il existe de nombreux autres outils pour tester les performances web côté front-end. Parmi ceux-ci, on peut citer PageLocity qui analyse aussi bien le SEO que l'intégrité du code HTML ou encore les performances de chargement. Il existe également des extensions sur le navigateur Google Chrome. Par exemple, Page Load Time ajoute une icône dans la barre d'outils qui affiche le temps de chargement de la page sur laquelle le navigateur se trouve. De son côté, l'extension Performance-Analyser donne en direct les temps de chargement du site, les requêtes effectuées ainsi que les ressources utilisées.

74%

Correct, mais... peut mieux faire!

Voir les priorités

10 Problèmes

13 Améliorations

67 Succès

VISITEUR SIMULÉ: Chrome Paris 8.0/1.5Mbps (Latence: 50 ms)

Modifier

Requêtes 50

Poids 584 ko

Premier octet 0.34 sec

Début affichage 0.93 sec

Fin chargement 1.75 sec

Erreurs navigateur 0 OK

Support d'HTTP/2: 18%

Speed Index: 1118

Technologies:

4

Summary

Performance grade B 85

Load time 1.76 s

Faster than 75 % of tested sites

Page size 570.1 kB

Requests 49

Tested from San Jose on Sep 27 at 18:02

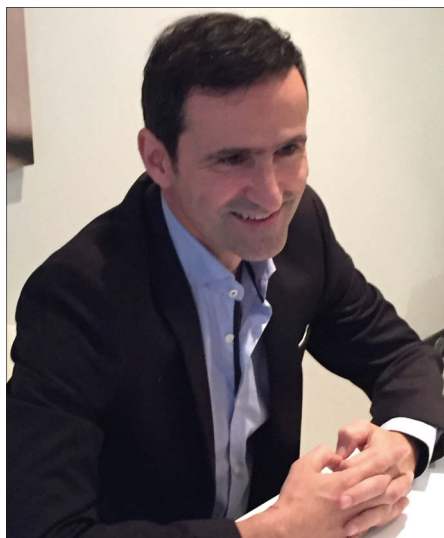
Performance insights

GRADE	SUGGESTION
F 10	Leverage browser caching
B 80	Remove query strings from static resources
A 93	Serve static content from a cookieless domain
A 97	Specify a cache validator
A 100	Avoid bad requests
A 100	Minimize redirects
A 100	Minimize request size
A 100	Specify a Vary: Accept-Encoding header

Response codes

RESPONSE CODE	RESPONSES
200 OK	49

5



# Pourquoi **France Télévisions** utilise-t-elle des outils de performance pour les applications déployées ?

• François Tonic

*Dans un précédent numéro de Programmez!, nous avons évoqué les outils de gestion de performances des applications, ou APM. Ces outils sont trop souvent méconnus des équipes techniques et des développeurs. Et pourtant, ils peuvent aider à améliorer l'expérience utilisateur de ces apps et résoudre de nombreux problèmes qui peuvent léster les performances. Dominique Face (directeur infrastructure technologies & services, France Télévisions) nous explique le pourquoi du comment.*

**L**e groupe France Télévisions rassemble plus de 9 800 collaborateurs et le département informatique gère plus de 250 applications très diverses : qu'elles soient intranet, desktop ou mobiles, elles couvrent l'ensemble des activités du groupe (administratif, technique, contenu, RH, etc.). L'utilisation d'un outil de performances remonte à environ 3 ans lorsque des utilisateurs métiers se sont plaints des performances d'une application. En une demi-journée, avec l'outil Dynatrace, il a été possible d'identifier les goulets d'étranglements et l'origine des problèmes de performances.

## Un outil très utile

*"Quand nous n'avons pas d'outils de performances, nous devons analyser les performances brique par brique, observer les comportements sur les postes de travail, vérifier la puissance des PC, tracer les trafics et flux réseaux, tracer les bases de données. Les problèmes peuvent se situer au niveau du code ou de l'infrastructure ou les deux. Et chaque application repose sur de multiples briques techniques. Pour trouver la brique fautive et identifier précisément le problème, cela devient difficile. Si le code est à l'origine des problèmes, il ne sert pas à grand chose d'analyser les infrastructures techniques, c'est une perte de temps, il faut analyser dans le code."* précise Dominique Face.

Un outil de type APM est tout sauf un artifice marketing. Il permet de tuner les applications et l'ensemble des infrastructures, selon l'outil que l'on déploie. Car ces outils peuvent être côté serveur, réseau ou application. Si on veut aller très loin dans l'analyse, il faut opter pour une plateforme APM capable d'aller dans le code et d'identifier les lignes de code fautives.

*"J'ai une longue expérience du monde de l'assu-*

*rance, plus de 15 ans. Nous disposons d'un service de qualité de services et de performances et à mon arrivée chez France Télévisions, j'ai proposé d'installer un outil APM sans mettre en place une équipe. L'équipe intégration assure les travaux d'intégration des livraisons applicatives des maîtrises d'oeuvre. C'est à elle que j'ai confié l'intégration des outils de performances et leurs utilisations".*

L'usage de ces outils se fait ponctuellement et uniquement sur les applications orientées Web/Intranet. Généraliser l'APM se révélerait trop coûteux pour le groupe. *"Nous n'avons pas forcément besoin de l'outil partout. Si une application n'a pas de problèmes de performances, nous n'avons pas besoin de l'outil. Dynatrace est très efficace pour les applications web. Dans une application legacy utilisant des technologies propriétaires de type Forms, Powerbuilder, Citrix, ..., c'est plus difficile."* Poursuit Dominique Face.

L'outil a été déployé dans le système d'information des ressources humaines, qui est utilisé par l'ensemble des salariés pour saisir les congés, les plannings, etc., ainsi que dans le SI finance. *"Avec cette solution, nous pouvons tester les performances en amont ainsi que le code."* Complète Dominique Face.

## Une démarche simple

Comme l'outil n'est pas systématiquement utilisé et qu'aucune équipe dédiée ne s'en occupe, il n'y a pas réellement de procédures à suivre. *"On travaille par opportunité"* nous indique Dominique Face. Deux cas de figure se présentent :

- 1 Quand un projet va passer en production et que l'on demande une analyse des performances ;
- 2 Quand une direction métier se plaint : aide aux diagnostics.

Quand la décision est prise d'utiliser l'outil, la méthodologie est simple :

- 1 On déploie l'outil sur les infrastructures hébergeant l'application dont il faut mesurer et analyser les performances : il faut généralement ½ journée pour analyser l'application et les couches techniques ;
- 2 Génération du rapport de mesure de performances ;
- 3 Analyse du rapport pour identifier les problèmes et proposer des solutions : 1 à 2 jours pour déployer les solutions curatives en production.

*"Nous ne faisons pas de nouveaux tests quand l'application est redéployée dans une nouvelle version. L'absence d'équipe dédiée limite nos possibilités car la charge de travail induite est non négligeable. Cependant, régulièrement, les utilisateurs peuvent remonter les difficultés lors des comités d'exploitation mensuels"* poursuit Dominique Face.

## Un bilan après 3 ans ?

*"Difficile de faire un retour après 3 ans quand on parle de performances. La mesure des performances et de la satisfaction client sont mesurables de multiples façons. Avec l'outil APM, nous sommes en mesure d'objectiver les problèmes de temps de réponse et en parallèle, nous réalisons des enquêtes de satisfaction interne auprès de nos utilisateurs pour leur donner la parole et mener des plans de progrès permanents."* Confie M. Face. Pour un utilisateur, bien souvent, il dira que le réseau ne fonctionne pas mais le problème est souvent tout autre. *"Je pense que tout DSI devrait avoir un tel outil pour garantir la qualité globale des applications et intervenir rapidement quand il y a un problème ou une demande d'une direction métier"* conclut Dominique Face. •



# Joel Polsky, développeur et fondateur de Stack Overflow

*Nous avons rencontré Joel Spolsky, expert mondialement reconnu en matière de développement de logiciels; il a créé en particulier la plateforme de gestion de projets Trello et le site d'échanges pour les programmeurs Stack Overflow.*

• Yves Grandmontagne

## Stack Overflow, la plateforme de Q&A des développeurs

Plateforme communautaire de Questions/Réponses (Q&A) pour les développeurs, Stack Overflow est devenue exceptionnellement populaire au fil des années. « A l'origine, notre objectif était de capter l'audience de *Joe on Software*, qui était d'environ 30 000 lecteurs, de faire mieux, et de proposer d'échanger entre développeurs », nous a confié Joel Polsky, CEO de la plateforme qu'il a co-fondée avec Jeff Atwood. La plateforme est actuellement fréquentée chaque mois par plus de 50 millions de développeurs ! Et Joel Polsky est très clair, Stack Overflow n'est pas un réseau social ! De même que par rapport à un Google, dont les réponses sont dispersées et souvent peu pertinentes par rapport aux attentes des développeurs, ces derniers ne consacraient que 1% de ce même temps sur Stack Overflow pour obtenir une réponse.

Joel Polsky a également joué la carte de l'innovation, en introduisant tout d'abord le vote des développeurs qui influence les réponses, puis la gamification avec des points qui s'accumulent en fonction du nombre d'uploads ou de réponses. « Tout cela participe à rendre Stack Overflow très efficace pour le développeur, en envoyant des signaux de haut niveau juste sur les réponses. Voilà pourquoi tous les développeurs de la planète nous ont rejoints, et qu'ils nous consultent en moyenne 7 fois par mois ». La plateforme serait fréquentée régulièrement par 40 à 50 millions de développeurs. Elle aurait cependant mesuré à 75 millions le nombre d'utilisateurs sur l'ensemble de ses services. « Nous pensons qu'il y a entre 50 et 60 millions de développeurs dans le monde. Le reste, ce sont des programmeurs accidentels, dont ce n'est pas le job, et à qui il arrive d'écrire du code ». Que pense-t-il des programmeurs citoyens, cette vision que nous sommes tous potentiellement des pro-

grammeurs ? L'idée fait sourire Joel Polsky... « Si on admet qu'il y a 50 millions de développeurs dans le monde, il reste 7 milliards de personnes qui ne développent pas ! ».

## Les modèles économiques et le code propriétaire

« Lorsque j'ai développé mon blog, j'ai dû développer mon propre programme, rappelle Joel Polsky. Aujourd'hui, programmer devient plus simple. Il y a Wordpress ou Blogger, il y a même Facebook et Twitter qui sont proches du blogging et de la publication. Et puis il y a les communautés. Et le computing devient plus abordable et plus puissant ». Il est par contre plus prudent sur les langages qui traduisent le langage naturel en langage machine. Il y aurait encore trop de 'détails' du langage naturel qui échappent à la machine.

« Si on admet qu'il y a 50 millions de développeurs dans le monde, il reste 7 milliards de personnes qui ne développent pas ! ».

Quel est le 'business model' de Stack Overflow ? La plateforme dispose de 4 sources de revenus : Stack Overflow Talent est un immense catalogue de talents dans lequel les entreprises peuvent piocher ; la publicité ; Stack Overflow Enterprise, un espace où les entreprises affichent leurs codes propriétaires afin d'en faciliter le partage avec la communauté ; et Developer Insight, destiné à exploiter les énormes volumes de données sur les développeurs conservées par la plateforme afin d'en permettre l'analyse, et par exemple de localiser les plus fortes communautés de développement sur des applications spécifiques.

L'évocation du modèle 'Enterprise' soulève

## LE PARCOURS EXCEPTIONNEL DE JOEL POLSKY

Né en 1965, diplômé en sciences de l'informatique à l'université de Yale, Joel Polsky rejoint en 1991 l'équipe de développement Microsoft au poste de Program Manager, où de 1991 à 1994 il conçoit VBA (Visual Basic for Applications), une implémentation du langage de programmation Visual Basic qui permet d'étendre les fonctionnalités d'Excel.

En 2000, il fonde sa première entreprise, Fog Creek Software, qui développe des outils de gestion de projet pour les développeurs de logiciels, dont Copilot, un outil d'assistance à distance. Et en 2011, il crée Trello, une solution collaborative et hébergée de gestion de projets, inspirée de la méthode Kaban, et proposée en mode freemium. Trello devient indépendante en 2014, avant de la céder à Atlassian contre 425 millions de dollars. Jovial, il commente simplement cette belle opération : « Je n'ai plus besoin de travailler... »

Auparavant, il aura créé en 2000 le blog *Joel on Software*, qui a participé à sa reconnaissance par la communauté des développeurs. « A l'époque, il n'y avait pas d'outil, j'ai dû programmer intégralement le blog », nous confie-t-il. Le succès est au rendez-vous, mais ne suffit pas au bouillonnant Joel Polsky. Alors, capitalisant sur ses expériences, il va créer Stack Overflow en 2008, en collaboration avec Jeff Atwood. Il en est encore aujourd'hui le CEO, alors qu'il a cédé sa place à la tête de Fog Creek.

une question, celle du code propriétaire et de la relation entre le développeur et la société propriétaire de la licence. « C'est en effet un problème intéressant. Quand un code est sur Stack Overflow, personne ne sait vraiment d'où il vient et s'il fait l'objet d'une licence. C'est pour cela que nous avons créé Stack Overflow Enterprise, qui facilite le partage de secrets avec l'entreprise. Alors que les entreprises peuvent s'affronter dans des procédures qui peuvent durer des années, si le développeur et l'entreprise entrent en compétition, ils peuvent coopérer sur la plateforme et les problèmes peuvent être réglés en une heure... »

### L'évolution de la programmation

« Personne ne comprend combien est important ce que font les programmeurs ! Les ordinateurs sont partout et les logiciels dominent la route. Cela se traduit par le logiciel qui remplace ce que nous avions l'habitude de faire directement et vient se placer entre nous et tout ce que nous utilisons. Tout ce que nous faisons peut aujourd'hui trouver son origine dans le logiciel. C'est pourquoi les développeurs ont une énorme opportunité d'influencer le monde ».

A l'évocation des App Store, et des millions de développeurs dont les applications végètent dans les limbes des catalogues, Joel Polsky préfère prendre exemple sur Trello. Certes, la solution a profité de sa présence sur les App Stores, mais c'est la création d'une communauté d'usage qui a assuré son succès. « Il est très difficile de comprendre ce qu'il y a derrière le succès d'une application, et chaque application a son histoire ». Et de pointer plus spécifiquement l'Apple Store qui donne trop de pouvoir à Apple. « Tout le monde devrait pouvoir mettre son application sur l'App Store ».

## L'interview vérité de Joel Polsky

*Joel Polsky a accepté de se livrer au jeu de l'interview vérité, un mot amène une réponse rapide...*

### Un succès ?

Stack Overflow

### Un échec ?

Fog Creek Software a lancé beaucoup, beaucoup d'applications, mais la plupart ont été un échec ! Mais je pense que mon plus grand échec fut quand j'ai essayé de créer une plateforme d'emplois en Inde. Il y a pourtant beaucoup de programmeurs en Inde, prêts à travailler pour quelques roupies, mais ce fut un échec.

### Une pratique ?

Pour moi la meilleure pratique est nommée « Chose que vous n'avez jamais faite ». Même lorsqu'il a à sa disposition des tonnes de



lignes de code, le programmeur est tenté de partir d'une feuille blanche, en imaginant qu'il va apporter un code propre alors que la masse peut être une source d'erreurs. Je pense au contraire que la clé est dans la masse.

### Un conseil ?

Mon principal conseil aux programmeurs est de communiquer afin d'être écoutés et que les gens les comprennent, c'est très important mais il est trop facile de l'oublier.

## LA COMMUNAUTÉ FRANÇAISE DU DÉVELOPPEMENT

Les programmeurs français forment l'un des groupes les plus importants dans le monde. « On ne mesure pas vraiment la dimension des programmeurs français car ils parlent plutôt bien l'anglais », nous confie Joel Polsky. L'anglais est la première langue dans les échanges de la communauté des programmeurs, le japonais en deux et le portugais en trois, en particulier avec l'apport du Brésil. « Le français est en neuvième position, mais c'est vraiment parce que les programmeurs français échangent en anglais. »

**Dans le prochain numéro !  
Programmez! #213, dès le 2 décembre 2017**

### Programmation

Les 10 erreurs les plus courantes en Java

### Défi Jedi

Créer son propre IDE en C++ avec la librairie Scintilla Partie 1

# POSTES À POURVOIR EN CDI : INGÉNIEURS DÉVELOPPEURS

*Que vous soyez débutant ou confirmé, votre candidature nous intéresse !*

Notre seul critère de sélection : vous aimez le développement et vous souhaitez en faire votre métier, au moins pour les premières années de votre carrière chez Osaxis. Vous pourrez évoluer sur des postes de lead developer, scrum master, chef de projet, ou consultant (technique ou métier).

Nous ne sommes pas attachés à une technologie en particulier, nous essayons de toujours pousser celles qui nous semblent les plus efficaces pour un contexte donné. Par exemple, en ce moment nous réalisons des projets sur Symfony, Drupal, AngularJS, .NET entity, JEE, Docker...

Vous intégrerez dans un premier temps notre équipe forfait pour apprendre à connaître nos méthodes et nos collaborateurs. Nous étudierons ensuite ensemble les missions les plus à même de développer votre potentiel.

Notre siège est basé à Paris Convention et nos clients se répartissent entre Paris et plus généralement en Ile de France.



## À propos de OSAXIS

[www.osaxis.fr](http://www.osaxis.fr)

*Osaxis est une ESN à taille humaine fondée par trois anciens développeurs, aussi nous sommes vigilants à ne pas reproduire tous les travers du métier (être traité comme du bétail, rester coincé sur une mission sans intérêt, se sentir plus salarié du client pour qui on intervient que de la société qui nous emploie, ...)*

*Rejoignez-nous et intégrez une équipe à taille humaine où votre avis compte*

*Envoyez-nous votre CV à l'adresse [rh@osaxis.fr](mailto:rh@osaxis.fr) et nous vous recontacterons très rapidement.*

*Rémunération proposée pour un CDI : à partir de 36 Keuros annuels*

*Lieu : Ile de France*



## STAGES DE FIN D'ÉTUDES

*Participation à un projet de développement mobile, JEE, LAMP et/ou .Net*

Vous êtes étudiant en école d'ingénieur ou en cycle universitaire, vous vous destinez aux métiers de développeur, chef de projet ou consultant technique, les nouvelles technologies vous passionnent.

.NET, iOS, Android, LAMP, AngularJS, Node JS, Docker ou JEE sont pour vous bien plus que des acronymes.

Vous finalisez vos études et vous avez besoin d'un stage pour valider votre dernière année.

### Ensemble, relevons vos premiers défis professionnels !

OSAXIS (<http://www.osaxis.com>) intervient dans les métiers du conseil technologique et de l'ingénierie, principalement chez des Grands Comptes et auprès des administrations, à Paris et en région parisienne.

Nous recherchons tout au long de l'année des stagiaires (ingénieur en fin d'étude) dans

le cadre de missions de développement mobile, JEE, LAMP et/ou .Net. Il s'agira d'implémenter, au sein d'une équipe aguerrie à ce type d'environnements, des modules fonctionnels dans les technologies retenues.

*Encadré par des consultants experts, le stage se déroulera de la façon suivante :*

- Initiation et perfectionnement aux méthodes agiles
- Développement de modules fonctionnels dans le respect des méthodologies et des normes de développement mises en place (interface, objet, mapping objet-relationnel, SQL dans SGBDR)
- Appropriation des technologies : un serveur d'applications avec son framework (Spring, JSF, Entity Framework, Symfony...) ou un environnement mobile (iOS, Android...) ; un IDE (Eclipse, Visual Studio, PHPStorm...) ; etc.

■ Appropriation du framework technique mis en place, de l'environnement objet multi-couches implémenté

■ Tests techniques et fonctionnels associés

■ Déploiement en recette et en production

Tout au long de votre stage, vous serez accompagné par une équipe aguerrie dans les technologies sus-citées, le tout dans une ambiance agréable et dans un environnement convivial. Cette expérience constituera un tremplin idéal pour évoluer vers un métier d'ingénieur expert, de chef de projet et/ou de consultant.

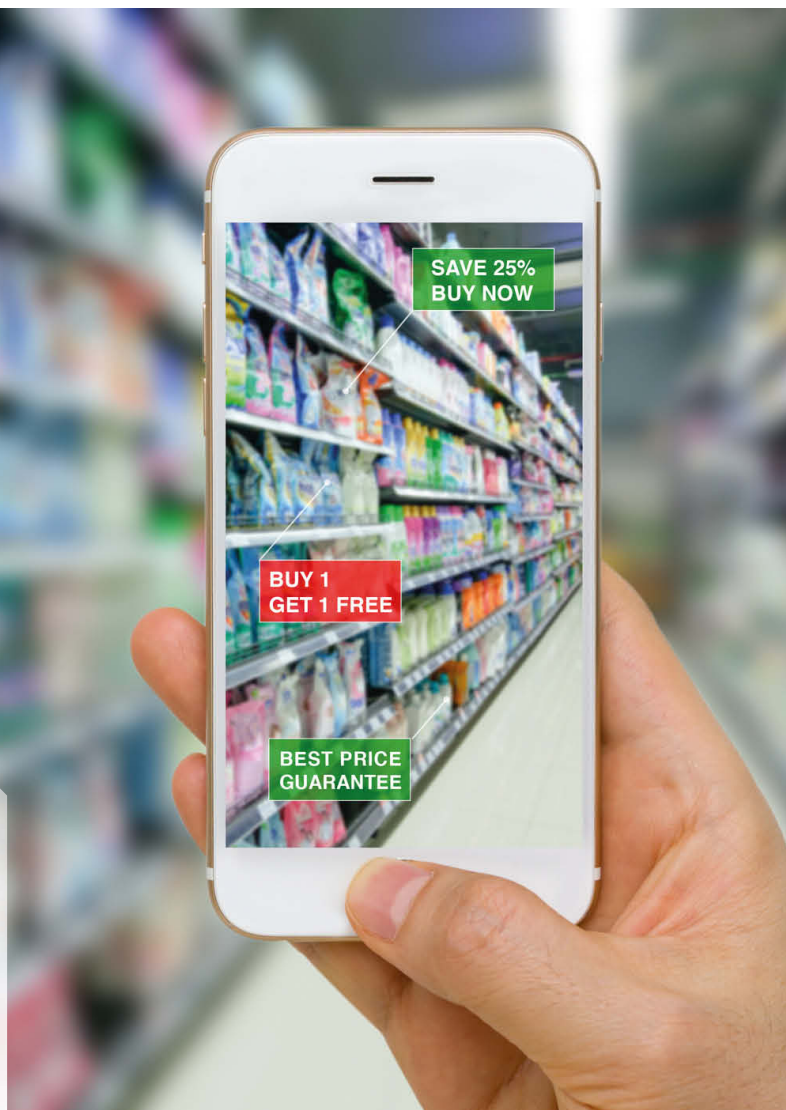
**Envoyez-nous votre CV à l'adresse [rh@osaxis.fr](mailto:rh@osaxis.fr) et nous vous recontacterons très rapidement.**

**Rémunération proposée pour le stage :**

1900 € bruts mensuels

**Lieu :** Paris 15e

# Le eCommerce ne s'improvise pas



© supparsom

**D**ans ce numéro, nous poursuivons notre dossier eCommerce. Comme vous avez pu le lire dans Programmez! 211 du mois dernier, un projet de boutique en ligne n'est pas quelque chose qui surgit comme ça... Il s'agit d'un projet complexe car il touche à la logistique, aux paiements, à l'expérience utilisateur, aux produits / services proposés. Même une simple boutique avec quelques références ne peut pas se faire à la va-vite.

Il faut aussi se dire qu'une mauvaise expérience utilisateur pèsera forcément sur les ventes. Si vous avez trop d'étapes de validation et de paiement, cela signifiera souvent que ce processus a été mal pensé dès le départ.

Vous n'acceptez que la carte bleue ? Que faites-vous de celles et ceux qui veulent payer autrement ? Faut-il intégrer les paiements de type Apple Pay ? Le paiement est un des points critiques de toute boutique en ligne et il

n'est pas le plus simple à implémenter car chaque banque a son API. Ne sous-estimez pas cette fonctionnalité.

Le design n'est pas non plus à négliger, surtout quand vous êtes en Responsive Design. La partie mobile reste difficile à caler car l'agencement est à repenser.

La gestion des stocks, la logistique, les expéditions sont d'autres sources critiques et de tensions. Il faut être extrêmement rigoureux, tenir à jour l'état des stocks, savoir où sont stockées les marchandises, etc.

La rédaction

# Interview de Patrick Schaudel, responsable du département Gestion des produits e-business chez 1&1



## Quels sont les mauvais réflexes ou mauvaises pratiques qu'il faut éviter quand on développe une boutique de e-commerce ?

La décision de créer une boutique en ligne est prise rapidement, il peut être cependant très frustrant d'avoir une boutique en ligne qui fonctionne mais qui ne génère pas suffisamment de trafic. Voici 5 conseils à suivre pour éviter les erreurs les plus courantes en débutant dans l'e-commerce :

- Ne pas analyser l'environnement : avant de lancer une boutique en ligne, il est nécessaire de bien connaître le marché, de définir un objectif et de cibler sa clientèle. Cette boutique en ligne représente-t-elle un complément de revenus ou est-ce l'activité principale du vendeur ? Quels produits souhaite-t-il vendre et à quel cœur de cible ? Qui sont ses concurrents, quels sont leurs produits et leurs prix ? Comment faire la différence ? Et enfin : pourquoi les internautes devraient-ils choisir d'acheter des produits chez lui et comment faire connaître sa boutique en ligne ? L'investissement financier à prévoir doit également être pris en compte : en plus des frais prévisibles, il est recommandé de planifier un budget pour le marketing ainsi que des réserves financières pour les premiers mois car il est rare de faire des bénéfices dès le début.

Astuce : commencer par proposer un choix restreint de produits afin de savoir lesquels sont les plus demandés et quel accueil leur sera réservé.

- Ne pas choisir une solution technique adaptée : le choix du système pour l'installation d'une boutique est crucial. Pour savoir quelle solution est la mieux adaptée, il faut comprendre quel type de boutique le commerçant souhaite créer, quel est son niveau de connaissance technique et à quelles ressources il a accès. Le nom de domaine de la boutique doit être simple, facilement reconnaissable, unique et correspondre aux produits proposés, à l'en-

treprise ou à son secteur d'activité.

- Ne pas accorder d'importance à l'ergonomie : temps de chargement rapide, sélection de moyens de paiement, navigation intuitive, belles images et description détaillée des produits permettent de réduire le taux de rebond et d'augmenter le taux de conversion. Comme les achats via mobile ont enregistré une hausse importante dans l'e-commerce, il est essentiel que votre boutique soit responsive. Pour vérifier l'ergonomie de leur boutique, les commerçants devraient se mettre à la place de leurs clients afin de répondre au mieux à leurs attentes.
- Ne pas rassurer, ne pas donner confiance : le sérieux et l'intégrité d'une boutique ont un réel impact sur la décision d'achat. Les modes de paiement proposés sont un bon exemple : les internautes veulent pouvoir choisir entre plusieurs méthodes de paiement, qui doivent être sécurisées et simples à utiliser. Les données clients sensibles doivent être chiffrées lors des transferts afin d'éviter le vol d'informations personnelles et le phishing. Le protocole de sécurisation des échanges SSL (Secure Socket Layer) permet le transfert des données en toute sécurité. Les conditions générales, les mentions légales, une déclaration de protection des données et des coordonnées de contact doivent également être affichées sur la boutique en ligne. La présence d'avis clients permet aussi de renforcer la confiance des internautes qui peuvent s'informer en lisant les commentaires laissés par d'autres clients.
- Être difficile à trouver : si personne ne connaît la boutique, celle-ci n'aura pas de visiteurs et donc aucune vente ne sera effectuée. La plupart des recherches pour un produit ou un service est effectuée par le biais des moteurs de recherche. Il est ainsi recommandé aux commerçants d'être présents dans les principaux annuaires en ligne et d'utiliser des modèles optimisés pour le référencement naturel (SEO), ainsi que d'avoir recours à la publicité ciblée (SEA), particulièrement au lancement. Pour cela, certains fournisseurs proposent des outils pour faciliter la mise en place de référencement SEO et SEA. De plus, lier la boutique à des marketplaces et portails externes comme Amazon et eBay permet d'optimiser sa présence en ligne et peut ainsi générer plus de ventes. Il est également judicieux d'être présent sur des portails de comparaison de prix comme Google

Shopping, Shopping.com, LeGuide, Idealo et autres afin de maximiser les chances d'avoir plus de trafic à ses débuts. Dans l'idéal, le système sur lequel repose la boutique en ligne propose une connexion à plusieurs portails de comparaison de prix afin que le marchand puisse choisir lequel est le mieux adapté à son secteur d'activité.

## Quand on utilise un CMS, vaut-il mieux utiliser un module boutique ou tout coder ?

Il y a aujourd'hui tellement de choix que les jours du « tout coder » sont bien dépassés. Pourquoi ne passe-t-on plus par du code pur ? D'abord, les exigences ont tellement évolué que coder un module boutique de A à Z sur la base du standard actuel coûterait très cher. Ensuite, tous les CMS les plus populaires ont un choix de modules boutiques allant du plus basique au plus puissant. Et ce, que l'on utilise un CMS Open Source ou propriétaire. Ce grand mouvement vers les CMS + Modules est une réalité aussi dans les agences Web. Toutes ou presque basent leurs devis sur des CMS et des modules. Certaines ont pu développer leurs propres modules boutiques, mais cela reste plus l'exception que la règle. Les tabous CMS et Modules sont tombés depuis plusieurs années.

Par contre, il faut faire attention à certains points quand on choisit son module :

- Le module est-il bien adapté au pays dans lequel je souhaite vendre ? Qu'il s'agisse des formats (dates, monnaie, unités), des modes de paiement, des modes de livraison, de la conformité avec les règles en vigueur, ou de la langue des interfaces, pensez à vérifier que le module e-commerce que vous souhaitez utiliser satisfasse votre cahier des charges.
- Pensez à l'année prochaine et aux suivantes : le module est-il activement développé et mis à jour ? De ce côté, -à encore, tous les modules ne se valent pas. Préférez la sécurité des modules les plus populaires, comme WooCommerce pour le CMS WordPress.
- Choisissez un module adapté à vos connaissances techniques : si vous n'êtes pas encore très familier avec votre CMS, et que vous n'avez qu'un ou deux produits à vendre, installer un gros module e-commerce n'est pas forcément le choix le plus judicieux. Orientez-vous peut-être vers un module boutique simplifié de type « Panier simplifié ». Moins d'options, mais beaucoup plus simple et rapide à mettre en place.

## Comment assurer de bonnes performances à sa boutique (latence, disponibilité, etc.)

C'est simple : la technologie utilisée en arrière-plan doit permettre de répondre aux exigences de la boutique.

- Les utilisateurs qui optent pour un logiciel de création de boutique en ligne ou une solution SaaS disposent de l'environnement d'hébergement dont ils ont besoin.
- Dans le cas d'une solution de boutique en ligne basée sur un contrat d'hébergement managé, le fournisseur permet à ses utilisateurs de bénéficier d'un temps de chargement optimal, d'une protection contre les défaillances, d'un support technique et prend également en charge les principales mises à jour.
- Une infrastructure Cloud fournit un environnement de haute performance qui peut si besoin évoluer au rythme de la boutique et qui est compatible avec les boutiques basées sur des solutions open source. Les offres Cloud de ce type comprenant le « load balancing » permettent de répartir le trafic d'un site Web sur différents serveurs et ainsi de faire face aux pics de charge saisonniers ou liés à une campagne. De manière générale, il est recommandé pour les marchands possédant des boutiques de grande envergure d'opter pour un hébergeur proche de sa clientèle ou proposant le CDN (Content Delivery Network).

Les packs hébergement ajustables sont également appropriés pour les boutiques professionnelles. En effet, les différents niveaux de performance disponibles permettent à l'utilisateur d'adapter la performance en fonction de ses besoins, en toute flexibilité.

### **Question sécurité : quels sont les bons réflexes pour le développeur ?**

La sécurité de l'infrastructure et des données est primordiale pour la fiabilité de la boutique et constitue un élément important à prendre en compte lors du choix de son hébergeur. En tant que partenaire des petites et moyennes entreprises, 1&1 ne se contente pas de fournir les services dont les e-commerçants ont besoin pour réussir sur Internet. Nous mettons également tout en œuvre pour assurer la sécurité de ces services et des données clients des utilisateurs de nos solutions. Pour cela, nous hébergeons les boutiques en ligne 1&1 sur des serveurs géo-redondants dans des datacenters certifiés. De plus, nos packs hébergement incluent un certificat SSL facile à installer et l'échange de données chiffrées sans frais supplémentaires. Nos datacenters sont aussi sécurisés contre la cybercriminalité grâce à la protection anti-DDoS et de puissants pare-feu, qui protègent les données des clients et les ressources en ligne.



## Interview de Romain Huet, *responsable des relations avec les Développeurs, Stripe*

### **Le paiement est un des éléments les plus sensibles dans une boutique en ligne. Pourquoi ? Et quelles sont les principales contraintes techniques ?**

Le paiement est probablement l'étape la plus sensible dans le processus d'achat en ligne puisque le consommateur a déjà choisi son produit et accepté les modalités de l'achat. Or, bien que déterminante, cette étape a trop longtemps été négligée par les marchands car trop complexe. Il n'y a pas si longtemps, un e-commerçant devait d'abord établir un contrat de VAD (Vente à Distance) avec sa banque, nouer des liens avec chaque réseau de carte (Visa, Mastercard, American Express), puis implémenter une infrastructure de paiement en lisant des centaines de pages de documentation et en assurant ensuite la maintenance. S'il se lançait dans un nouveau pays, il devait recommencer la démarche. Concrètement, une entreprise devait donc s'appuyer sur une équipe de développeurs, de juristes, nouer des relations avec les institutions financières, conduire des audits de sécurité, et constamment rester en veille pour suivre les innovations dans le secteur. Beaucoup d'entreprises se sont refroidies devant tant de complexité. Et celles qui s'y sont aventurées ont investi des moyens considérables pour une expérience rapidement obsolète, notamment sur mobile. Les chiffres le montrent. Seulement 5% du commerce se fait aujourd'hui en ligne... Et ce, 25 ans après la création d'Internet !

### **En général, quand on parle paiement sur une boutique, on parle API bancaire, Apple Pay ou encore Paypal, quelle est votre position ?**

C'est bien plus complexe que cela. Quand on parle de paiement, on parle de normes ISO, de PCI DSS, de licences... Ces complexités freinent considérablement les entreprises qui souhaitent innover. Or la capacité pour une entreprise à exécuter rapidement tout en proposant une expérience utilisateur optimale est le nerf de la guerre. C'est exactement ce que propose Stripe aux marchands de toutes tailles : en quelques lignes de code ils peuvent - via une API - disposer d'une infrastructure de paiement digne des plus belles entreprises technologiques du monde, et ce, quel que soit leur modèle économique (e-commerce, plateformes, places de marché, SaaS...). Une plateforme, par exemple, a des

contraintes bien spécifiques en matière de transactions. Au-delà d'accepter un paiement en provenance d'un consommateur — généralement mobile — une marketplace doit être capable d'immédiatement verser son dû à un vendeur tiers, tout en prélevant sa commission. Cela paraît simple sur le papier mais c'est un véritable cauchemar technique et juridique, notamment en matière de vérifications d'identités qui diffèrent pays par pays ! Stripe Connect, lancé il y a quelques mois en France, répond précisément aux problématiques des plateformes et des places de marché, et leur fait gagner un temps considérable. C'est notre ADN : une infrastructure pensée par des développeurs, pour permettre à des développeurs de se concentrer sur ce qui compte le plus aujourd'hui : l'expérience utilisateur, le design sur-mesure, et les fonctionnalités différenciantes.

### **Quels sont les modèles de développements habituels ? Et les mauvaises pratiques à éviter ?**

Il y a quelques années, l'adage voulait que le paiement soit trop stratégique pour être confié à un tiers. C'est aujourd'hui précisément l'inverse. Au-delà du coût, des contraintes de sécurité, de stabilité, et des fonctionnalités de base que proposent les différents outils du marché, il est important de regarder la facilité d'intégration et la flexibilité de l'API pour offrir une expérience de paiement optimale. Il y a quelques semaines, nous avons lancé Stripe Elements, un moyen simple de créer une expérience d'achat rapide et personnalisée, avec un effort d'intégration minimal. Elements permet d'éviter beaucoup d'erreurs que commettent encore la majorité des sites e-commerce et des applications mobiles. Au-delà de la conformité aux normes PCI, Elements offre par exemple la validation des données en temps réel, la localisation des devises et des modes de paiement, sans oublier le support d'Apple Pay et de la Payment Request API - le nouveau standard du W3C - dans le navigateur. Elements s'adapte à toutes les tailles d'écran et permet d'éviter la redirection en dehors d'un site web ou d'une application au moment du paiement, ce qui est tout simplement une mauvaise expérience pour l'utilisateur.

### **Quelles différences entre un paiement sur desktop et sur mobile ?**

Le mobile a durablement transformé les façons de consommer et pose de nouveaux challenges pour les marchands : s'adapter à un écran plus petit, minimiser les informations demandées à un utilisateur en mobilité, et s'intégrer parfaitement aux différents formats de navigation (web mobile ou application). Dans le même temps, les exigences des consommateurs ont considérablement évolué, notamment sur mobile. Un consommateur veut une expérience de paiement simple, fluide, rapide et sécurisée. La norme est aujourd'hui au paiement en un clic ou un tap sur mobile. C'est ce qui explique le succès croissant d'Apple Pay ou d'Android Pay, notamment dans les achats en ligne, que Stripe permet d'intégrer facilement dans des applications mobiles avec des SDKs natifs. Que vous soyez une entreprise technologique, une startup ou une entreprise centenaire, il n'y a plus aucune raison de ne pas offrir à vos consommateurs une expérience de paiement digne de ce nom.

# Du produit à sa **commercialisation** en ligne... et au-delà : réussir sa boutique en ligne

Partie 2

*La conception d'un site e-commerce ne peut pas se résumer à une simple opération technique. Un ensemble de sujets tant métiers que techniques sont à considérer, avant sa mise en ligne. Une fois le site ouvert aux utilisateurs, il convient de l'exploiter, de le faire vivre et le maintenir. Ce dossier développe les aspects théoriques liés à la mise en œuvre d'un site e-commerce, ainsi qu'un exemple de site basé sur le développement d'un instrument de musique MIDI réalisé il y a quelque temps : la tirasse MIDI. [1]*

• Jean-Marie HEITZ  
• Xavier FLAMANT  
**Netapsys**

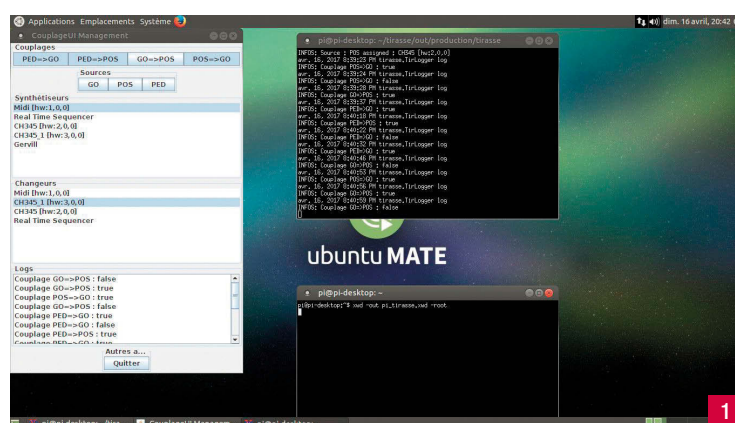
**NET/PSYS**  
GROUPE SODIFRANCE

## LA VIE QUOTIDIENNE DU SITE Exploitation du site

*Une utilisation du site très différente selon ses parties prenantes*

L'exploitation du site à divers aspects impliquant d'une manière ou d'une autre des acteurs physiques :

- Le commerçant doit faire vivre le site, de sorte que l'intérêt des clients ne se délite pas dans le temps et tout en respectant ses objectifs commerciaux. Cette animation du site peut revêtir différentes formes, notamment :
  - Le renouvellement du contenu : en tant que site web à part entière, la mise à jour du contenu est source de capture et de fidélisation des clients. C'est l'objet des éléments de CMS présents dans Magento, à savoir les blocs CMS, qui sont des morceaux de pages, et les pages CMS qui sont comme leur nom l'indique des pages à part entière. Dans ces deux entités, du contenu est écrit et mis en forme soit manuellement, soit à travers d'une interface WYSIWYG. Cette interface est spécialisée, car elle permet notamment l'intégration de contenu dynamique tel que des variables, ou des URL dynamiques (c'est-à-dire des URL qui sont générées en tenant compte des URL de base du store sur lequel on se trouve, ce qui permet de garder des blocs CMS identiques entre les différents environnements). Avec l'existence des blocs CMS, une possibilité souvent utilisée est celle de générer des pages qui montrent le même bloc CMS (par exemple le header ou le footer), d'où une centralisation et une non redondance de code facilitant la mise à jour du site. Des éléments plus complexes peuvent être ajoutés de manière assistée dans les pages grâce aux widgets, ceux-ci étant des blocs pourvus d'une interface de configuration générée majoritairement grâce à des directives XML
  - les promotions : plusieurs mécanismes de promotions diffèrent par leur étendue, leur mode de fonctionnement et d'application :
    - Tout d'abord les promotions configurées au niveau du produit et agissant sur celui-ci : définition d'un prix spécial de date à date, ou un prix qui est appliqué en fonction de la quantité mise au panier et du groupe de clients dont le client fait partie,
    - Les promotions appliquées par règle de prix catalogue continuent d'agir sur le prix de mise du produit au panier, mais de manière plus globale : par exemple, les produits faisant partie d'une catégorie particulière, ou les produits ayant un certain jeu d'attributs (règle de



Le logiciel de tirasse en action sur un Raspberry Pi

réduction pourcentage ou par montant fixe),

- Les promotions appliquées par règle de panier : ce sont les règles les plus souples mais également les plus compliquées, dont l'application est évaluée via des collecteurs spécifiques lors de la collecte des totaux de quote. Elles autorisent également des promotions par application de codes de réduction,
- La communication vers les clients : activement via des newsletters, ou plus passivement via des flux RSS, avec les problématiques qui sont inhérentes à chaque type de communication (non passage en spam pour les newsletters et fidélisation nécessaire pour la justification des flux RSS, pour ne citer que ces deux aspects). Il ne faut pas oublier non plus que l'espace client, qui peut contenir en particulier un résumé des commandes et la possibilité pour le client de changer ses informations personnelles, peut éventuellement également devenir un moyen de communication,
- Au niveau des clients, il s'agit tout d'abord de naviguer sur le site, de constituer un panier, puis de convertir le panier en commande pour réaliser la vente. Ainsi, à travers ses opérations, le client déclenche la création d'une session de front office, déclenche l'enregistrement de logs de visites, modifie la ou les quotes qui lui sont associées, jusqu'au moment où il passe le tunnel et réalise la commande. Il y a éventuellement des flux de paiement qui entrent en jeu pour que la commande soit validée,
- Au niveau du système, selon les mécanismes utilisés réellement par le commerçant, il peut y avoir génération d'enregistrement en BD de facture, génération d'avis, ainsi qu'une gestion correspondante de l'état de la commande, avec d'éventuels aspects de génération de documents PDF et de mails,

- Le système générant des données, il est également nécessaire au commerçant d'analyser les données enregistrées, de sorte à pouvoir vérifier si les résultats escomptés sont atteints par le site, ou s'il y a des ajustements à prévoir. À ces fins, plusieurs niveaux de données sont possibles :
- Les données immédiatement disponibles : ce sont les informations qui sont fournies par les grids de Magento, telles que les listes de clients ou de commandes : ces données relativement "atomiques" peuvent donner une vision de l'activité instantanée, de même que les informations sur les sessions qui sont disponibles depuis le BO,
- Les données agrégées calculées au sein de Magento : des rapports sont prévus pour être générés à intervalles réguliers. Ils peuvent donner des informations de tendances pouvant aider un commerçant à dégager un "profil de fonctionnement" de son site et des tendances,
- Les données issues d'informatique décisionnelle : à ce niveau, on sort du logiciel e-commerce pour réunir des données qui peuvent venir de plusieurs sources, comme les données de serveur (logs des accès de serveur web par exemple), les données du logiciel e-commerce, les données recueillies par analytics, via la navigation des clients, ou même des sources de données complémentaires qui peuvent d'une manière ou d'une autre influencer sur l'activité (ex : la météo). On peut chercher à analyser manuellement les données et en tirer soi-même des informations et des décisions métiers
- Les éléments trouvés via de l'intelligence artificielle : il s'agit de se poser des problématiques telles qu'elles pourraient être analysées via des approches d'intelligence artificielle ; on trouve dans les types de problèmes rencontrés des problèmes de groupement (clusterisation), de régression (trouver le modèle mathématique correspondant aux données), ou de la prédiction. Il s'agit à ce niveau d'applications et de recherches sur la science des données.

En ce qui concerne la tirasse, la vie du site impliquera bien évidemment des actions clients avec la génération des commandes, mais étant donné qu'il s'agit d'un marché de niche et d'un mode de mise à disposition assez particulier, l'animation du site se fera surtout via du contenu et les modules CMS.

## Intégration dans le Système d'Information

**La maison E-Commerce étant réalisée, il est temps de la raccorder aux réseaux urbains : les échanges de flux.**

L'intégration dans le SI consiste généralement en plusieurs flux :

- Export ou synchronisation de la liste des clients,
- Import de la liste des produits et éventuellement des stocks,
- Export de la liste des commandes.

Pour ce faire, plusieurs solutions sont envisageables :

- Travail sur la base de données : méthode généralement pas recommandable pour les imports, en revanche, elle peut être satisfaisante la plupart du temps au niveau des exports : comme on utilise une base MySQL, il est tout à fait possible de faire du « SELECT INTO OUTFILE » de sorte à sortir les données dans un fichier par exemple. De la même manière MySQL dispose d'une sortie en ligne de commande en XML.
- Travail manuel depuis Magento : Magento dispose dans les affichages tabulaires (grids) de possibilités d'export vers des fichiers CSV et dispose d'un assistant qui permet d'importer et d'exporter des customers et des produits. Il est également possible de configurer des imports/exports via des profils écrits en XML.
- Shell Magento : Magento dispose d'une classe de Shell qui permet d'écrire des scripts qui peuvent être lancés depuis la ligne de com-

mande. Ceci est donc également un moyen pour faire des traitements, notamment pour des exécutions uniques.

- Utilisation de web services : Magento propose une batterie de web services qu'il est donc envisageable d'utiliser dans le cadre d'outils d'ETL (comme Talend ou les Integration Services d'un MS SQL Server).
- Utilisation de files de message : l'utilisation de ce genre d'élément est favorisée par Magento 2 qui supporte dans sa version Enterprise les files. Néanmoins, il est envisageable même dans un Magento 1 de les utiliser. En effet, la nature asynchrone des files de messages permettrait a priori de traiter les messages entrants via une tâche CRON, tandis que des messages sortants pourraient être générés via l'instrumentation du code - au travers d'observateurs par exemple. Le découplage est assez intéressant car, il peut aider à résoudre des situations où des systèmes passent en mode maintenance et doivent se synchroniser par la suite lorsqu'ils repassent en ligne.

Ce sont là les types de solutions les plus courantes. Néanmoins, d'autres types de problématiques peuvent se poser et nécessiter des développements.

Pour les curieux : un exemple traité dans un autre article est l'intégration de la VOIP avec un Backoffice Magento :

<https://blog.netapsys.fr/couplage-telephonie-informatique-et-application-a-magento-premiere-partie/>  
<https://blog.netapsys.fr/couplage-telephonie-informatique-et-application-a-magento-deuxieme-partie/>  
<https://blog.netapsys.fr/couplage-telephonie-informatique-et-application-a-magento-troisieme-partie/>  
<https://blog.netapsys.fr/couplage-telephonie-informatique-et-application-a-magento-quatrieme-partie/>  
<https://blog.netapsys.fr/couplage-telephonie-informatique-et-application-a-magento-cinquieme-partie/>

## MAINTENANCE DU SITE E-COMMERCE

Le site e-commerce doit être l'objet d'une certaine maintenance s'ajoutant au monitoring des infrastructures. On peut distinguer deux types d'opérations : les opérations d'exploitation et les opérations d'évolution.

### Les opérations d'exploitation

**De l'entretien quotidien automatisé aux réparations ponctuelles plus conséquentes ... en passant par la serrurerie**

D'une part, il y a des opérations de maintenance automatisées, via des tâches périodiques programmées : il s'agit en particulier d'opérations de nettoyage des journaux, de nettoyage des devis (quotes) qui ne sont plus valides, d'agrégation de données pour les rapports. Les exécutions des tâches CRON sont journalisées dans la table « cron\_schedule ». Il convient de l'analyser relativement régulièrement, pour vérifier que toutes les tâches se passent correctement et que les tâches attendues y soient présentes (si une tâche prend trop de temps, il peut arriver que des tâches ne puissent plus s'exécuter car, elles sont trop loin de leur point d'exécution ; il peut même arriver que les tâches n'aient pas du tout été planifiées).

D'autre part, les opérations manuelles d'exploitation consistent la plupart du temps à résoudre des problématiques que le client remonte dans le cadre de sa Tierce Maintenance Applicative (TMA). Il y a une constante dans la plupart de ces opérations : il s'agit d'abord de récupérer les informations nécessaires à la reproduction et à l'analyse du problème, puis de les analyser et ensuite voir s'il y a des possibilités de correction ou de contournement.

L'étape de récupération des informations est importante : les informations qui sont le plus souvent utilisées sont un accès à la base de données

(voire un dump, pour pouvoir effectuer des requêtes "lourdes") et un accès aux divers journaux (exceptions et logs Magento, logs Apache HTTPD etc.). Il n'est parfois d'ailleurs pas inutile d'avoir accès à des sauvegardes un peu plus anciennes pour chercher à reproduire le problème remonté. La connaissance de la structure de la base de données et du SQL est souvent fortement utile pour analyser les problèmes – de même qu'une bonne compréhension des particularités des développements spécifiques utilisés par le client.

Un exemple usuel de problèmes est une dégradation progressive des performances : le plus souvent, il s'agit alors d'analyser le code et le comportement de MySQL pour à la fois optimiser les requêtes effectuées et de créer les index nécessaires pour ne pas avoir à parcourir directement les tables de données.

Un autre exemple de tâche est l'application des patches de sécurité. On pourrait se dire que l'application des patches de sécurité est une opération simple, car il suffit en théorie de lancer le fichier du patch. En pratique, c'est plus compliqué que cela, car il faut tenir compte des développements spécifiques qui peuvent interférer. Un exemple classique est l'ajout des "formkeys", des tokens qui veillent à ce qu'un formulaire ait d'abord bien été généré par magento avant d'avoir obtenu les informations de sa soumission : la mise en place des formkeys est simple au niveau du PHP, mais encore faut-il que le formkey soit bien présent dans les formulaires, d'autant plus que les surcharges de template, qui contiennent l'inclusion des formkeys, est fréquente : dans ce cas, il faut faire l'opération d'ajouter la formkey dans la surcharge de template. [2]

## Les opérations d'évolution

### Pleins feux sur un brico ... développement : le moyen de paiement Promesse de don

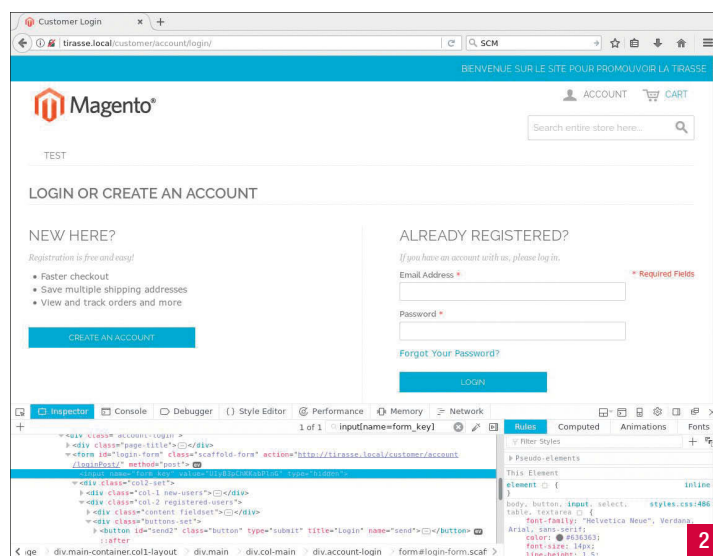
Les opérations d'évolutions sont rendues possibles et "propres" grâce à un design modulaire qui structure le projet en profondeur. Ainsi, dans le cadre de Magento, l'ensemble du cœur du code est séparé dans des modules, eux-mêmes regroupés au sein de namespaces et enfin d'aires de code. En pratique, 3 aires de code existent :

- Core, pour le code natif ;
- Community, pour les modules communautaires ;
- Local, pour les modules spécifiques au projet et qui ne seront pas réinjectés dans la communauté.

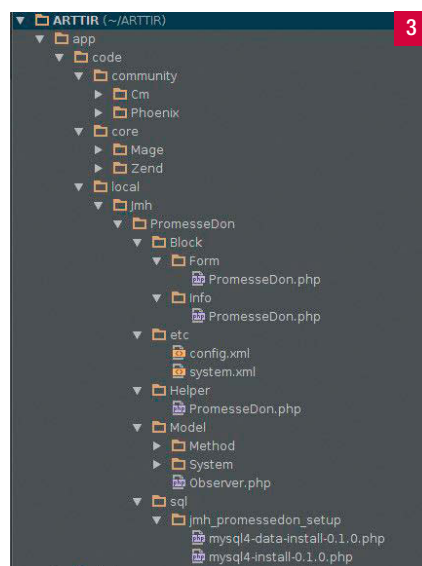
Afin d'illustrer cette partie, nous avons préparé un petit module de paiement qui serait à développer dans le cadre du site de la tirasse : il s'agit du module de promesse de don. Le but est de pouvoir indiquer quels sont les articles qui sont éligibles à la promesse de don et que les articles éligibles ne soient payables qu'avec ce moyen de paiement. Ceci a pour conséquence qu'un devis mélangeant des articles éligibles et non éligibles ne pourra jamais être transformé en commande, faute de moyen de paiement disponible pour tous les éléments.

En ce qui concerne la répartition du code dans les modules, plusieurs types de séparation sont envisageables :

- Réunir tout le code concernant la fonctionnalité que l'on développe au sein d'un même module : c'est ce qui se fait généralement pour les modules communautaires, mais cela a l'inconvénient que si un développeur ne connaît pas tellement les tenants et aboutissants de la fonctionnalité, il peut avoir du mal à trouver le code recherché ;
- Réunir le code en se basant sur la structure native de Magento : dans le namespace lié au projet, reproduire en fonction des besoins la struc-



Exemple de page avec une formkey



Développement de l'arborescence de Magento et du module de Promesse de Don

ture de Magento : ainsi on pourra retrouver un module Catalog pour ce qui touche les produits et les catégories, un module « Payment » pour les moyens de paiement, etc. L'inconvénient est que le code d'une fonctionnalité est réparti sur plus de fichiers et il est donc facile de ne pas voir une partie du code liée à une fonctionnalité ;

- Il y a aussi toutes les démarches "hybrides" réunissant des éléments des deux démarches évoquées au-dessus.

Dans le cadre de l'article, afin d'avoir des fichiers synthétiques, nous avons opté pour la création d'un namespace appelé Jmh et un seul module, appelé PromesseDon. [3]

Tous les modules ont une structure commune, qui se décline en fonction des besoins. Les fichiers php qui sont dans les modules sont, hors les installeurs, des classes, dont le nom reflète l'emplacement. Dans un module, on pourra avoir des répertoires qui seront notamment :

- etc : configuration du module, avec notamment config.xml, qui est le seul fichier obligatoire du module, en dehors du fichier d'activation du module, qui sera dans app/etc/modules ;
- controllers : Magento ayant une structuration MVC, le répertoire

controllers contient les contrôleurs qui font le lien entre les données et la présentation des données ;

- **Model** : les éléments liés aux données (et à la base de données) ;
- **Helper** : des fichiers utilisés en singleton qui sont à disposition pour être utilisés souvent d'un peu partout et également pour les traductions ;

**Block** : les blocs affichables : ce sont donc des aspects de présentation qui sont traités ici ;

- **sql** : il arrive fréquemment que les modules nécessitent des modifications de la base de données lors de leur installation : c'est surtout à cela que sert ce répertoire ; malgré son nom, la plupart des installeurs sont écrits en php.

Le module PromesseDon respecte cette structure.

Lorsqu'on veut analyser ce que fait un module, la première des choses à faire est d'analyser le fichier config.xml. Celui du module PromesseDon est reproduit ci-dessous.

```
<?xml version="1.0"?>
<config>
  <modules>
    <Jmh_PromesseDon>
      <version>0.1.0</version>
    </Jmh_PromesseDon>
  </modules>
  <global>
    <events>
      <payment_method_is_active>
        <observers>
          <jmh_payment_prevent_other_means_when_don_involved>
            <class>payment/observer</class>
          </jmh_payment_prevent_other_means_when_don_involved>
        </observers>
      </payment_method_is_active>
    </events>
    <blocks>
      <payment>
        <rewrite>
          <form_promesseDon>Jmh_PromesseDon_Block_Form_PromesseDon</form_promesseDon>
          <info_promesseDon>Jmh_PromesseDon_Block_Info_PromesseDon</info_promesseDon>
        </rewrite>
      </payment>
    </blocks>
    <models>
      <adminhtml>
        <rewrite>
          <system_config_source_cms_block>Jmh_PromesseDon_Model_System_Config_Source_Cms_Block</system_config_source_cms_block>
        </rewrite>
      </adminhtml>
      <payment>
        <rewrite>
          <method_promesseDon>Jmh_PromesseDon_Model_Method_PromesseDon</method_promesseDon>
        </rewrite>
      </payment>
    </models>
  </global>
</config>
```

```
<_pro_messeDon>
  <observer>Jmh_PromesseDon_Model_Observer</observer>
  <rewrite>
    <payment>
      </models>
    </payment>
  </models>
  <helpers>
    <payment>
      <rewrite>
        <promesseDon>Jmh_PromesseDon_Helper_PromesseDon</promesseDon>
      </rewrite>
    </payment>
  </helpers>
  <resources>
    <jmh_promessedon_setup>
      <setup>
        <module>Jmh_PromesseDon</module>
        <class>Mage_Eav_Model_Entity_Setup</class>
      </setup>
    </jmh_promessedon_setup>
  </resources>
  <product>
    <collection>
      <attributes>
        <promesse_don_eligible/>
      </attributes>
    </collection>
  </product>
  <sales>
    <quote>
      <item>
        <product_attributes>
          <promesse_don_eligible/>
        </product_attributes>
      </item>
    </quote>
  </sales>
  </global>
  <default>
    <payment>
      <promesseDon>
        <active>1</active>
        <model>payment/method_promesseDon</model>
        <order_status>complete</order_status>
        <title>Promesse Don</title>
        <allowspecific>0</allowspecific>
        <group>offline</group>
        <form>promessedon_form</form>
        <info>promessedon_info</info>
      </promesseDon>
    </payment>
  </default>
</config>
```

Ici, deux mécanismes fondamentaux de Magento sont utilisés : les rewrites et la programmation événementielle.

Les rewrites : la plupart des objets ne sont pas générés par du "new" dans Magento, mais par des appels à des méthodes statiques de la classe

Magento.php, qui est le point central pour récupérer des dépendances dans le code. Ainsi, si on veut utiliser un Model, on charge le modèle par un appel à `Magento::getModel()`, en indiquant le nom du module, suivi du nom de la classe. C'est ainsi par exemple que la classe de la surcharge est appelée « `Magento_Payment_Model_Observer` », qui est « `Jmh_PromesseDon_Model_Observer` », via `payment/observer`. Les rewrites sont utilisés aussi bien pour les classes surchargées que pour les nouvelles classes que l'on veut appeler depuis le nom d'un module tiers.

L'observateur est le deuxième élément qui entre en jeu : Magento dispose d'un système événementiel qui permet de s'enregistrer pour recevoir et traiter des événements. Les déclarations des écouteurs sont indiquées, comme on a pu le deviner, dans `config.xml`, tandis que l'émission d'un événement est déclenché par l'appel de `Magento::dispatchEvent`. Dans le cadre de `PromesseDon`, j'utilise un observateur pour désactiver les autres méthodes de paiement que la promesse de don quand le panier contient un article éligible :

```
/**
 * Forbids other payment means when promesse don eligible products are involved
 * @param Varien_Object $observer
 * @event payment_method_is_active
 */
function preventOtherThanPromesseDonWhenPromesseDonInvolved($observer)
{
    $evt = $observer->getData('event');
    $result = $evt->getData('result');
    $methodInstance = $evt->getData('method_instance');
    $quote = $evt->getData('quote');
    $pdonHelper = Mage::helper('payment/promesseDon');
    if ($methodInstance->getCode() != Jmh_PromesseDon_Model_Method_PromesseDon::PROMESSE_DON_CODE)
    {
        $result->isAvailable = false;
    }
}
```

`PdonHelper` est un `Helper`, c'est-à-dire un singleton ne présentant généralement pas d'état et contenant des fonctions appelées d'un peu partout. Son code va récupérer les produits correspondant à chaque item de panier et vérifier s'ils sont éligibles. La vérification d'éligibilité d'un produit se trouve également dans le `helper` :

```
public function checkEligibleProduct($product)
{
    $res = false;
    if ($product) {
        $eligible = $product->getPromesseDonEligible();
        if ($eligible) {
            $res = true;
        }
    }
    return $res;
}
```

J'ai ajouté l'attribut de produit `promesse_don_eligible` sur les produits en passant par l'installateur déclaré dans le fichier `config.xml`. L'installateur ajoute l'attribut de la manière suivante :

```
$installer = $this;
$entityType = $installer->getEntityType('catalog_product');
$defaultSetId = $installer->getDefaultAttributeSetId($entityType);
$defaultGroupId = $installer->getDefaultAttributeGroupId($entityType, $defaultSetId);
$installer->addAttribute(
    $entityType,
    'promesse_don_eligible',
    array(
        'label' => 'promesse_don_eligible', 'type' => 'int',
        'required' => 0,
        'default' => 0,
        'user_defined' => 1, 'input' => 'select',
        'source' => 'eav/entity_attribute_source_boolean',
    )
);
$attrId = $installer->getAttributeId('catalog_product', 'promesse_don_eligible');
$installer->addAttributeToGroup($entityType, $defaultSetId, $defaultGroupId, $attrId);
```

On retrouve les notions vues précédemment : attribut, type d'entité, attribut set et attribut group.

Concernant la fonctionnalité de paiement proprement dite, elle est relativement simple : il suffit d'étendre la classe `Magento_Payment_Model_Abstract` :

```
class Jmh_PromesseDon_Model_Method_PromesseDon extends Magento_Payment_Model_Method_Abstract
{
    /**
     * payment method code constant
     */
    const PROMESSE_DON_CODE = 'promesseDon';
    * @var bool cant be used internally

    protected $_canUseInternal = false;
    * @var bool cannot be used for recurring profiles

    protected $_canManageRecurringProfiles = false;
    * @var string payment code

    protected $_code = Jmh_PromesseDon_Model_Method_PromesseDon::PROMESSE_DON_CODE;

    * @var string form block
    protected $_formBlockType = 'payment/form_promesseDon';

    * @var string info block
    protected $_infoBlockType = 'payment/info_promesseDon';

    * @var bool do not review payment
    protected $_canReviewPayment = false;

    /** we cannot edit
     * @return bool
     */
    public function canEdit()
    {
        return false;
    }
}
```

```
/**
 *Check whether the payment method is applicable to the quote
 * @param Mage_Sales_Model_Quote $quote
 * @param int|null $checksBitMask
 * @return bool
 public function isApplicableToQuote($quote, $checksBitMask)
 {
     $applicable = parent::isApplicableToQuote($quote, $checksBitMask);
     if ($applicable) {

         $applicable = $pdonHelper->checkOnlyEligibleItems($quote);
         return $applicable;
     }
 }
```

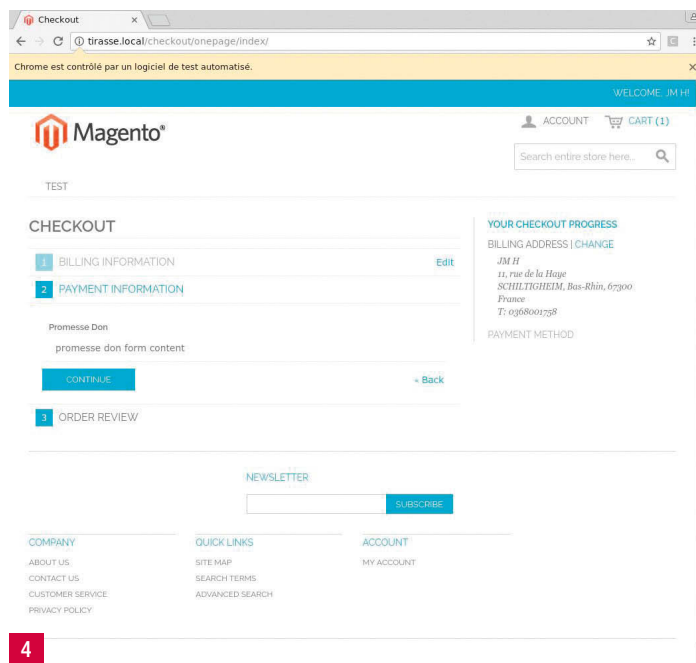
Au début de la classe, des attributs permettent la configuration de l'instance. Ainsi la seule méthode que nous avons dû implémenter est la fonction pour vérifier si la méthode est applicable au panier, qui fait appel au helper pour effectuer son travail. Dans la méthode de paiement, un composant joue un rôle plus important que d'habitude : il s'agit des valeurs par défaut indiquées à la fin de config.xml, car l'emplacement de ces variables dans la configuration globale de Magento est utilisé pour récupérer les méthodes de paiement installées sur le système, avec notamment le paramètre "model" qui permet de récupérer la classe responsable du paiement. Comme on l'a compris, il n'y a pas de flux financier en jeu dans ce pseudo-moyen de paiement. Toutefois, il faut quand même que la commande arrive à un état où elle est considérée comme payée. Magento s'en charge grâce à son code natif : `Mage_Sales_Model_Order_Payment::place` contient le nécessaire pour positionner le statut qui est défini comme statut après paiement. En remontant les appels, on constate que la méthode sera appelée (transitivement) depuis les fonctions `submit` de `Mage_Sales_Model_Quote` et que celles-ci sont lancées depuis `Mage_Checkout_Model_OnePage::saveOrder`, appelé depuis `Mage_Checkout_Controller_OnePageController::saveOrderAction`. Et c'est justement le tunnel `one-page` (par défaut) qui sera utilisé dans la tirasse. [4]

Le paiement pourra donc être considéré comme effectué au niveau de la commande. Toutefois, aucune facture ne sera produite par le système. De ce fait, un petit réglage sera nécessaire pour que les produits téléchargeables payés par une promesse de don puissent être téléchargés : il s'agit de mettre l'état des items compatibles avec le téléchargement à "pending", dans le back-office. Étant donné que ce réglage n'est pas directement lié au moyen de paiement, nous nous sommes abstenus de le mettre dans le setup du module et dans les valeurs par défaut.

## Les tests unitaires

**Vérifier rapidement et ponctuellement si ce qu'on a fait marche et n'a pas impacté ce qu'on a fait avant.**

On a vu que le `Model Jmh_PromesseDon_Model_Method_PromesseDon` ne contenait pas réellement de code métier, mais qu'à la fois l'observateur et le `Model` faisaient référence à du code "métier" dans le helper du module. Ce code est critique pour la fonctionnalité, car c'est lui qui permet de dire ou non si l'on autorise ce type de paiement pour le panier. Comme c'est du code critique, il convient de le protéger de sorte



La Promesse de Don dans le tunnel de vente

à détecter des lacunes ou d'éventuelles régressions qui pourraient être introduites par des évolutions ultérieures. C'est là que les tests unitaires entrent en jeu. A titre d'exemple, je vais produire ici le test de la fonction du helper montré plus haut.

```
/**
 *checkEligibleProduct Data Provider
 * @return array
 */
public function checkEligibleProductDataProvider()
{
    return array(
        array(new Varien_Object(array('promesse_don_eligible'=>1)), true),
        array(new Varien_Object(array('promesse_don_eligible'=>0)), false), array(new Varien_Object(), false)
    );
}

/**
 * @param boolean $expected
 * @param Mage_Catalog_Model_Product $product
 * @dataProvider checkEligibleProductDataProvider
 */
public function testCheckEligibleProduct($product, $expected)
{
    //setup
    $h=Mage::helper('payment/promesseDon');
    //exercice
    $computed=$h->checkEligibleProduct($product);
    //verify
    $this->assertEquals($expected, $computed);
    //implicit teardown
}
```

Ce code est divisé en deux fonctions : la fonction de dataProvider, qui n'a qu'un rôle de fournisseur de cas de tests par rapport à la fonction de test proprement dite (une entrée de tableau renseigne l'ensemble des paramètres requis par la fonction de tests) et la fonction de test.

Les tests unitaires utilisés ici sont des tests PHPUnit, mais ils ont la même structure que la plupart des tests unitaires :

- Une phase de setup, qui permet de mettre en place l'environnement du test ;
- Une phase d'exécution du code à tester ;
- Une phase de vérification du résultat ;
- Une phase (ici implicite) de nettoyage, de sorte à remettre l'environnement tel qu'on l'avait trouvé.

Les tests unitaires sont normalement conçus pour ne pas générer de modification de leur environnement (dont la base de données) et sont normalement rapides à exécuter. De ce fait, on peut les exécuter régulièrement et même automatiser leur exécution (par exemple, à chaque dépôt de code) dans le cadre d'une intégration continue. [5]

Il faut bien comprendre que les tests unitaires sont utiles même par moment lors de l'écriture du code, lorsqu'on veut commencer à déboguer. En effet, il arrive que les techniques des tests unitaires soient tout aussi rapides à mettre en œuvre que des tests plus manuels, tout en obtenant des bénéfices à plus long terme. Et on les apprécie vraiment quand un changement de code provoque une modification de comportement qui est détectée par un test, car on peut alors vérifier s'il s'agit d'une régression ou d'une évolution nécessaire de comportement. Dans tous les cas, cela rend le développeur plus confiant en son code.

Pour les curieux : une base de données est souvent requise en tant que fixture lorsqu'on travaille avec Magento ... Et on en arrive à une notion de golden database, qu'il faut savoir retrouver, d'une manière (<https://blog.netapsys.fr/test-unitaire-approche-db/>) ou d'une autre (<https://blog.netapsys.fr/tests-unitaires-utilisant-une-base-de-donnees-de-nouvelles-perspectives/>).

## Les tests d'acceptation

### Ou comment gonfler ses ventes ... en non-production !

Un autre type de tests est également intéressant : il s'agit des tests d'acceptation ... automatisés. Il s'agit dans notre cas de vérifier que des cas d'utilisations se réalisent de la manière attendue. Ces tests sont des tests de "haut niveau", peu rattachés au code : on travaille au travers d'un navigateur et on lui demande d'effectuer des actions, semblables aux actions manuelles, dont on cherche ensuite à analyser le retour pour voir s'il est conforme ou non.

Dans le cadre de cet article, nous nous intéressons à l'outil Codeception (<http://codeception.com/>). C'est un outil qui permet de facilement réaliser des tests d'acceptation, en particulier au niveau du poste du développeur, car il permet d'écrire en PHP les tests et de les exécuter grâce au protocole WebDriver, (via une librairie venant a priori de facebook). C'est grâce à lui qu'on peut ensuite s'interfacer avec un navigateur, éventuellement via un programme tiers de passerelle comme c'est le cas par exemple avec Chromium, et ce afin de réaliser un scénario de passage de commande avec un compte d'utilisateur existant.

L'outil est facile à installer et à utiliser, car il s'agit surtout d'avoir une installation PHP fonctionnelle et de récupérer le fichier phar correspondant à la version de PHP utilisée. Ensuite, sur Ubuntu, on installe (via apt) le package chromium-chromedriver, pour le navigateur Chromium. Le site de Codeception est bien documenté et fournit le nécessaire pour configurer l'installation minimale pour un poste de développeur.

Le script fonctionnel est le suivant :

```
root@barttir:/var/www/magento/dev/tests/unit# phpunit --configuration phpunit.xml
PHPUnit 5.4.6 by Sebastian Bergmann and contributors.

Runtime:       PHP 7.0.18-0ubuntu0.17.04.1 with Xdebug 2.5.0
Configuration: /var/www/magento/dev/tests/unit/phpunit.xml

.....                                     13 / 13 (100%)

Time: 275 ms, Memory: 8.00Mb

13 / 13 test cases passed!
root@barttir:/var/www/magento/dev/tests/unit#
```

Exemple d'exécution de tests phpunit

```
papa@papa:~/CODECEPTION$ php -f codecept.phar -- run --steps
Codeception PHP Testing Framework v2.3.4
Powered by PHPUnit 6.2.3 by Sebastian Bergmann and contributors.

Acceptance Tests (1) -----
FirstTest: Try to test
Signature: FirstTest:tryToTest
Test: tests/acceptance/FirstTest.php:tryToTest
Scenario --
I am on page "/"
I see "bienvenue sur le site pour promouvoir la tirasse"
I click "test"
I see "Test2"
I click "tirasse", "[class=product-name]"
I see "Add to Cart"
I click "input[type=checkbox]"
I click "Add to Cart"
I see "tirasse was added to your shopping cart."
I click "Proceed to Checkout"
I see "Returning Customers"
I fill field "login[username]", "myusername@maydomain.com"
I fill field "login[password]", "mypassword"
I click "Login"
I see "Select a billing address"
I click "Continue"
I wait 5
I see "Promesse Don"
I click "Continue", "#payment-buttons-container"
I wait 5
I click "Place Order"
I wait 10
I see "Thank you for your purchase"
PASSED

Time: 28,3 seconds, Memory: 8,00MB

papa@papa:~/CODECEPTION$
```

La même chose, en exécution ...

```
public function tryToTest(AcceptanceTester $I)
{
    $I->amOnPage("/");
    $I->see("bienvenue sur le site pour promouvoir la tirasse");
    $I->click('test');
    $I->see('Test2');
    $I->click('tirasse','[class=product-name]');
    $I->see('Add to Cart');
    $I->click('input[type=checkbox]');
    $I->click('Add to Cart');
    $I->see('tirasse was added to your shopping cart.');
```

```

$I->wait(5);
$I->see('Promesse Don');
$I->click('Continue',"#payment-buttons-container");
$I->wait(5);
$I->click('Place Order');
$I->wait(10);
$I->see('Thank you for your purchase');
}

```

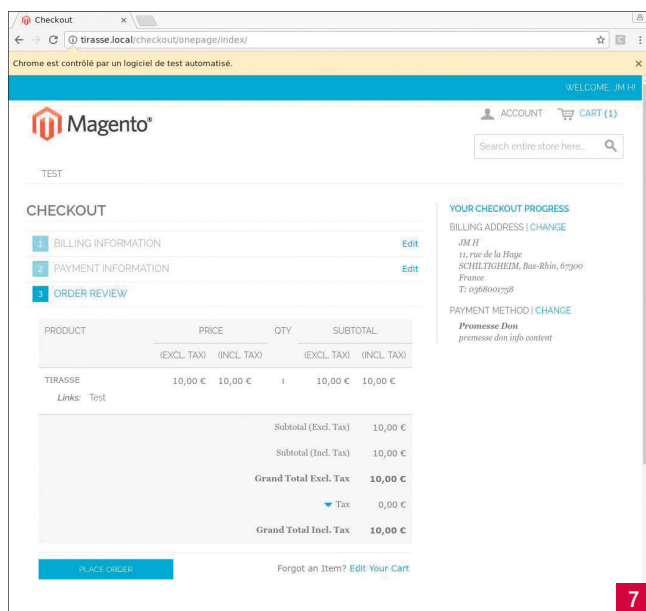
[6]

La réalisation de ce script a pris peu de temps, mais a permis de se rendre compte que l'écriture de bons tests d'acceptation est une chose qui peut s'avérer non triviale :

- Il peut y avoir des différences entre ce que l'utilisateur voit et ce que le navigateur voit : nous avons par exemple été confrontés à des casses différentes entre ce qui est indiqué dans le test (et qui est dans le code) et ce qui est vu dans le navigateur (qui peut être formaté via du CSS par exemple).
- Il peut être difficile de bien cibler des éléments : en effet, tous les éléments n'ont pas des ids, et certains noms se retrouvent plusieurs fois dans les pages.
- Des faux positifs sont possibles : nous cherchions à accéder à la home-page alors que la machine virtuelle n'était pas lancée. L'étape de `$I->see('tirasse')` a cependant été validée, car oui, le nom tirasse était présent dans l'URL et était présent dans le contenu de la page d'erreur : donc le comportement du test était pertinent, mais la recherche n'était pas assez spécifique pour détecter le problème.
- Les techniques peuvent avoir du mal à savoir quand les chargements sont terminés : c'est pour cela, que des waits sont rajoutés dans les tests, pour laisser le temps au code JS de s'exécuter et de modifier l'affichage avant que les assertions soient évaluées

Nous en déduisons qu'il reste encore de l'expérience à acquérir pour pouvoir faire de "bons" tests automatisés, mais il est évident que Codeception fournit un environnement efficace pour faciliter leur écriture, surtout pour un débutant. [7]

Une attention particulière doit être apportée au dernier type de tests : la



Exécution du test : le navigateur fait son marché tout seul ...



Exemple d'une installation pouvant bénéficier de la tirasse MIDI (tirasse MIDI non déployée ici)

recette client. La recette client peut intervenir à différentes phases lors des développements, que ce soit sur des développements de fonctionnalités spécifiques ou des ensembles de modifications constituant une release. Dans tous les cas, cette recette est extrêmement importante, car le client est celui qui connaît le mieux son fonctionnement métier et son besoin. De plus, il doit le plus souvent se familiariser aux nouvelles fonctionnalités pour pouvoir les mettre correctement en œuvre en production. Il faut donc que le client comprenne et prenne conscience de l'importance de ces tests pour qu'ils puissent fournir toute la valeur ajoutée requise. Ceci nécessite donc également au passage l'existence d'environnements de tests bien isolés de la production et des environnements de développements, pour que chacun puisse travailler sur sa "partie" en toute quiétude : d'où à minima un environnement de préproduction complet (c'est-à-dire, avec les environnements de tests des dépendances) pour les tests clients finaux.

## CONCLUSION

Le processus pour vendre un produit en ligne est complexe et nécessite un investissement en travail et en ressources conséquents ; aussi bien pour sa conception que pour son exploitation et sa maintenance. C'est pour cela que le site de la tirasse ne restera probablement qu'un exemple illustratif.

Néanmoins, il y a une dimension essentielle qui ne peut pas être transcrite dans un article : la création d'un site e-commerce est également une aventure humaine, que ce soit du côté des développeurs ou du côté des clients. Si vous avez un projet, je vous invite à partager cette aventure humaine avec nos équipes à l'occasion de séminaires ou de salons professionnels.

Pour les curieux : après avoir lu cet article, une petite pause s'impose ... Un son et lumière, ça vous dit ?

Si oui : <https://blog.netapsys.fr/magento-son-et-lumiere-webmidi-et-dmx/>

[8]

# Azure Stack : le Cloud Hybride de Microsoft

Partie 1



Daniel Tizon  
Principal Consultant chez Trivadis SA (Suisse)  
Consultant Azure et Azure Stack  
**Microsoft PSELLER**  
daniel.tizon@trivadis.com - @daniel\_tiz  
**trivadis**  
makes IT easier.

*Imaginez qu'il soit possible de posséder son propre cloud Azure, non pas hébergé sur l'un des datacenters de Microsoft en Irlande ou aux Pays-Bas, mais dans sa propre salle serveurs, ses propres locaux, ou dans le datacenter de son hébergeur local. C'est ce que permet Azure Stack !*

Sachant que Microsoft Azure est disponible aujourd'hui dans plus de 38 régions, Azure Stack est un moyen de bénéficier d'Azure dans de nombreux autres lieux, et pourquoi pas dans votre propre datacenter si vos besoins l'exigent ! Azure Stack est donc une extension d'Azure.

Sous la pression de la transformation numérique, beaucoup d'entreprises vont dans le cloud public aujourd'hui pour bénéficier d'architectures modernes et rafraîchir leurs applications historiques. Cependant les entreprises continuent de garder des applications sur site pour des raisons ou obstacles technologiques ou réglementaires.

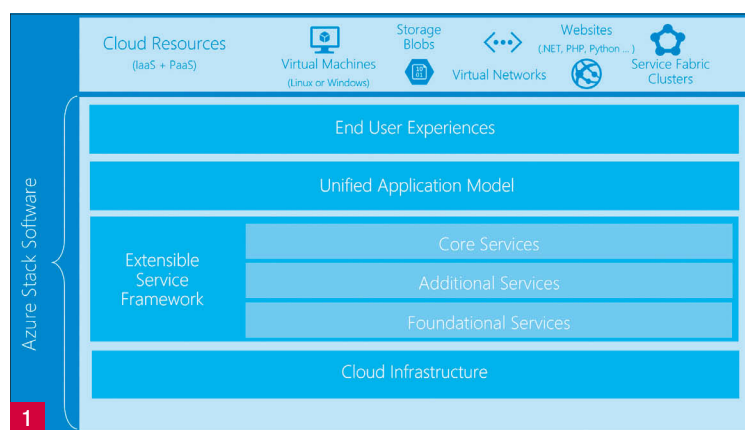
Azure Stack répond à plusieurs objectifs :

- Étendre Azure avec Azure Stack pour lever les freins et accompagner l'innovation. La possibilité de déployer des applications innovantes en utilisant des outils, des process, des Framework identiques aussi bien dans le cloud que dans vos environnements sur site.
- Permettre à vos développeurs de gagner en productivité en leur permettant de construire et déployer des applications avec la même expérience, qu'elles soient déployées dans Azure ou Azure Stack.
- Exécuter des services Azure localement. Déployer des solutions de cloud computing privées ou hybrides selon vos besoins. Avec Azure Stack, vous pouvez héberger localement des services IaaS et PaaS avec la même expérience d'administration et les mêmes outils que ceux utilisés par vos équipes pour Azure.

Les compétiteurs principaux de Microsoft que sont Amazon et Google dans le cloud n'offrent pas de solutions aux entreprises devant conserver une partie de leur infrastructure sur site. D'un autre côté VMware qui faisait référence en matière de virtualisation d'entreprise a abandonné son initiative cloud après avoir vendu sa solution vCloud Air Business à l'hébergeur OVH en avril dernier. Pour finir, Open Stack serait une solution pour son cloud privé, mais demande des compétences très pointues pour le mettre en place et l'opérer, sans parler du fait qu'il faut apprendre un cloud encore différent.

Ce qui caractérise le Cloud selon Microsoft est qu'il ne doit pas être associé par son lieu d'hébergement, mais plutôt par ses capacités à fournir un moyen moderne et simple pour déployer et monitorer ses applications sur des infrastructures fiables, sécurisées, redondantes et élastiques.

Le grand intérêt d'Azure Stack est qu'il permet maintenant aux clients qui, pour des raisons réglementaires ne pouvaient pas bénéficier de solutions cloud, de pouvoir en bénéficier en toute sécurité sur site tout en capitalisant sur des compétences Azure très répandues. Il n'y a donc pas de nouveaux portails, ou de nouvelles API à apprendre. Le portail Azure Stack, les outils de Scripting (PowerShell, Azure CLI, AzCopy, Blob Storage Explorer), les templates ARM au format JSON sont les mêmes sur Azure Stack et sur Azure.



Vous l'avez compris, Azure Stack est l'arme absolue de Microsoft pour compléter sa stratégie cloud. Azure Stack était annoncé et très attendu depuis quelques années. Fort de son expérience avec Azure, Azure Stack partage en très grosse partie le même code source qu'Azure. C'est de l'Azure en miniature, non pas à l'échelle d'un datacenter de plusieurs milliers de nœuds, mais pour l'entreprise ou pour l'hébergeur local ! [1]

## Architecture d'Azure Stack

Pour réaliser un POC Azure Stack, Microsoft propose une version pouvant tourner sur une seule machine, sous le nom de version Azure Stack Development Kit (ASDK), mais c'est uniquement pour tester la solution. Azure Stack pour un usage en production se présente sous la forme d'un système intégré qui est une combinaison de software, de services et de hardware. Afin de garantir la meilleure expérience possible dans le déploiement et le management d'Azure Stack par vos équipes IT, seule une liste certifiée de matériels peut faire tourner Azure Stack pour le moment, sachant qu'Azure Stack sera livré préinstallé avec le matériel.

Les configurations hardware proposées sont composées de 4 à 12 machines (nœuds), qui correspondent à ce que Microsoft appelle un Scale Unit que vous pouvez commander auprès de Dell, HP, Lenovo, et plus tard chez Cisco et Huawei (Voir tableau Configurations matérielles Azure Stack [2]).

Quelques remarques sur les configurations matérielles : dans la version ASDK, Microsoft recommande 128 GB de RAM pour permettre de tester tous les services dont les AppServices et les SQL Databases. Mon retour d'expérience est que 128 Go restent insuffisants. Prévoyez de pouvoir étendre la mémoire à 256 Go pour pouvoir vraiment tester Azure Stack dans des conditions satisfaisantes. Prévoyez également un gros disque pour la partie OS. Une fois que vous aurez booté sur votre

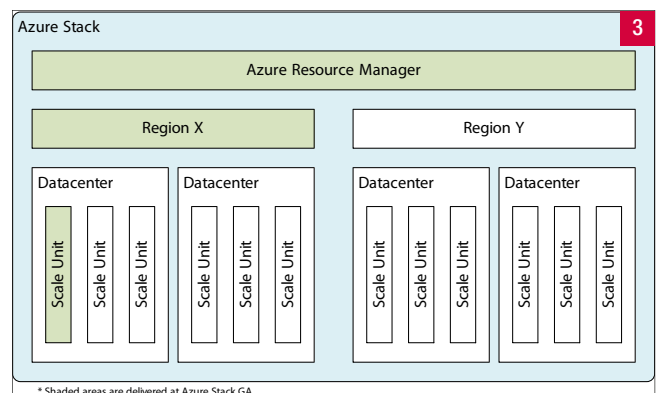
2	AZURE STACK OFFER			
	Proof Of Concept ASDK	DELL	HPE	Lenovo
Model	NO NAME	PowerEdge R730xd Rack Server	HPE DL380 Gen9	x3560 M5
Number of Server/Size	1	Small: 4 Medium: 8 Large: 12	4 8 12	Entry : 4-8 node (25U Rack) Enterprise Starter: 4-12 node (42U Rack) Enterprise Full: 12 node (42U Rack)
Compute CPU x Cores/Threads	Minimum: Intel Xeon E5-2650v4 GHz 2.2 (2 x 12/24) Intel Xeon E5-2660v4 GHz 2.0 (2 x 14/28) Recommended: Intel Xeon E5-2620v4 GHz 2.1 x (2 x 16/32)  Recommended: Intel Xeon E5-2683v4 GHz 2.1 x (2 x 16/32)	Low: Intel Xeon E5-2640v4 GHz 2.4 (2 x 10/20) Medium: Intel Xeon E5-2650v4 GHz 2.2 (2 x 12/24) High: Intel Xeon E5-2660v4 GHz 2.0 (2 x 14/28)	Intel Xeon E5-2650v4 GHz 2.2 (2 x 12/24) Intel Xeon E5-2660v4 GHz 2.0 (2 x 14/28) Intel Xeon E5-2683v4 GHz 2.1 x (2 x 16/32) Intel Xeon E5-2695v4 GHz 2.1 x (2 x 18/36) Intel Xeon E5-2698v4 GHz 2.2 (2 x 20/40) Intel Xeon E5-2699v4 GHz 2.2 (2 x 22/44)	Intel Xeon E5-2680v4 GHz 2.4 (2 x 14/28) Intel Xeon E5-2683v4 GHz 2.1 (2 x 16/32) Intel Xeon E5-2690v4 GHz 2.6 x (2 x 14/28)
Memory - RAM (GB)	Minimum: 96 Recommended: 128 Best experience: 256 or more	Low: 256 Medium: 384 High: 512	256 386 512 640 768	256 512
Storage S2D Cache SSD (TB)		Low: 5.7 Medium: 11.5 High: 11.5	4.8 (4x1.2TB) 6.4 (4x1.6TB)	Low: 3.2 (4x800GB) High: 6.4 (4x1.6TB)
Storage S2D Capacity (TB)	Minimum: 4x140GB (SSD or HDD) -> 400 GB for S2S Recommended: 4x250GB (SSD or HDD) -> 800GB for S2S Best experience: 4x3TB ->10TB for S2S	Low: 40 Medium: 60 High: 80	60 (10x6TB)	Low: 40 (10x4TB) High: 60 (10x6TB)
Management Server Hardware Software	Azure Stack Host (contains Hyper-V Services) is provided as Boot on VHD. You need one OS Disk  Minimum : 1x300GB (to get 200GB left for the VHD) Best experience: 1x3TB	HLH Dell R430 Windows Server 2016  OpenManage Essentials OpenManage Network Manager	HLH DL360 G9 Windows Server 2016 2 x E5-2620v4, 4 x 16 GB RAM, 4 x 600 GB SAS Drive HPE OneView with iLO Advanced	HLH x3550 M6 Windows Server 2016 2 x E5-2620v4, 64GB RAM, 4 x 600GB SAS drive XClarity Administrator (the Pro edition is available at additional cost)
Networking	Windows Server 2012 R2 Certification required for NIC; no specialized features required	2 x Dell EMC Networking S4048-ON for ToR switching  1 x Dell EMC Networking S3048-ON for BMC switch  Mellanox ConnectX-3 Pro Dual Port (10GbE SFP+)	2 x HPE FlexFabric 5900AF 48XG 4QSFP+ Switch for ToR  1 x HPE FlexFabric 5900AF 48G 4XG 2QSFP+ Switch for BMC switch  HPE Ethernet 10Gb 2-port 546FLR-SFP+ Adapter	2 x Lenovo RackSwitch G8272 for ToR  1 x Lenovo RackSwitch G8052 for BMC switch  Mellanox ConnectX-4 Dual Port 10/25GbE
Services		Consultative Services ProDeploy ProConsult	Consulting and Implementation Services Integration Services Financial Services	Planning Fulfillment Deployment Support Optimization

partition Azure Stack (Boot sur VHD), l'espace libre sera bien utile pour construire vos solutions à base d'ISOS et VHD depuis cette même machine.

Concernant les offres matérielles pour un usage en production, DELL propose 3 configurations : Low, Medium, High, sans possibilité de mixer. HPE propose un très large choix de CPU, tout est à la carte sur les autres types de ressources. Lenovo propose 3 configurations avec deux choix d'options sur la RAM et le stockage tout en apportant une offre de services très complète.

Pour vous donner un ordre de grandeur, avec un Scale Unit de 12 serveurs, vous pouvez provisionner plus de 400 VM de taille D2 (2 virtual CPU, 7Go RAM), et pouvez bénéficier de plus de 40 To de stockage.

Aujourd'hui avec la sortie en General Availability d'Azure Stack, vous ne pouvez avoir qu'une seule région (local) et un seul Scale Unit. Par la suite, il est prévu dans la roadmap la possibilité de plusieurs Scale Unit par DataCenter, plusieurs DataCenter par région et plusieurs régions. [3] Sans grandes surprises, Azure Stack s'appuie sur des services fournis au-dessus de Windows Server 2016 DataCenter.



**Storage Space Direct (S2S) :** permet d'accéder à un pool de disques répartis sur plusieurs serveurs faisant partie d'un cluster multi-nœuds. Cette technologie permet un haut niveau de disponibilité et de performances du stockage pour un prix très réduit comparé au coût traditionnel des SAN et NAS.

**Network Controller** : services réseaux qui permettent de gérer, d'assurer la sécurité et la communication de toute la couche réseau d'Azure Stack. Les services fournis incluent la gestion Firewalls, Répartiteurs de charge logiciels, Réseaux Virtuels, et Passerelles RAS.

**Hyper-V** : Hyper-V est la couche de virtualisation utilisée par Azure Stack. Elle fournit les services de Compute, Networking, Storage et Isolation. Voyez Azure-Stack comme une couche logicielle au-dessus de Hyper-V qui en termes de management et d'automatisation va bien au-delà de ce qu'on peut faire avec Hyper-V seul ou System Center Virtual Machine Manager (SCVMM), ou enfin VMWare ESXi

**Windows Server Failover Cluster** : Fonctionnalité de Windows Server qui permet à un groupe de serveurs de fonctionner ensemble dans un cluster tolérant aux pannes. Les serveurs mis en cluster sont connectés par des câbles physiques et logiciels. Si l'un des nœuds du cluster tombe, les autres nœuds commencent à fournir le service (un processus appelé failover). Le cluster failover s'appuie en outre sur un stockage partagé Cluster Shared Volume (CSV).

**Server Core** : Server Core est une version allégée de Windows Serveur 2016. Lorsqu'on se connecte sur un des nœuds d'Azure Stack, sur la console Hyper-V Manager, on remarque qu'une quinzaine de VM « d'infrastructure Azure Stack » sont provisionnées. Pour la plupart, elles sont basées sur Server Core (plus économes en ressources qu'une version de Windows avec une GUI).

**System Center 2016** : Azure Stack lui-même est complètement tolérant aux pannes et dispose d'outils de gestion et de supervision. Sachez cependant que Microsoft propose l'utilisation de System Center 2016 en option, pour superviser Azure Stack, via les composants Operation Manager, Service Manager ou Orchestrator. [4]

## Architecture logique d'Azure Stack [5] [6]

La couche logicielle Azure Stack est provisionnée lors du setup, sous forme de 13 VM qui offrent l'infrastructure de base. Il y a une VM par rôle dans la version Azure Stack Development Kit (ASDK) qui est mono-nœud. Ces rôles sont redondés dans la version de production d'Azure Stack et répartis sur 4, 8 ou 12 nœuds, et leur bonne exécution est pilotée par le service Windows Server Failover Cluster.

**AzS-ACS01** : Azure Stack storage services.

**AzS-ADFS01** : Active Directory Federation Services (ADFS). Pour vous connecter à vos portails Azure Stack, vous devez utiliser une identité

ADFS (celle-ci) ou une identité cloud (AAD provenant d'une souscription Azure ou O365 que vous aurez « connectée » comme fournisseur d'identité de votre Azure Stack).

**AzS-BGPNAT01** : Routeur périphérique. Fournit les fonctionnalités NAT et VPN pour Azure Stack.

**AzS-CA01** : Services de Certificate authority pour les rôles Azure. Assure la sécurité des échanges et de l'authentification entre les différents services d'Azure Stack.

**AzS-DC01** : Services Active Directory, DNS, et DHCP pour Azure Stack. Il contient les comptes de services utilisateurs et groupes nécessaires à son fonctionnement. Le domaine par défaut d'Azure Stack est azurestack.local.

**AzS-ERCS01** : Emergency Recovery Console VM. Cette VM est à utiliser en cas d'urgence pour arrêter la plateforme Azure Stack par exemple. Il est important d'arrêter le système en suivant un protocole bien précis.

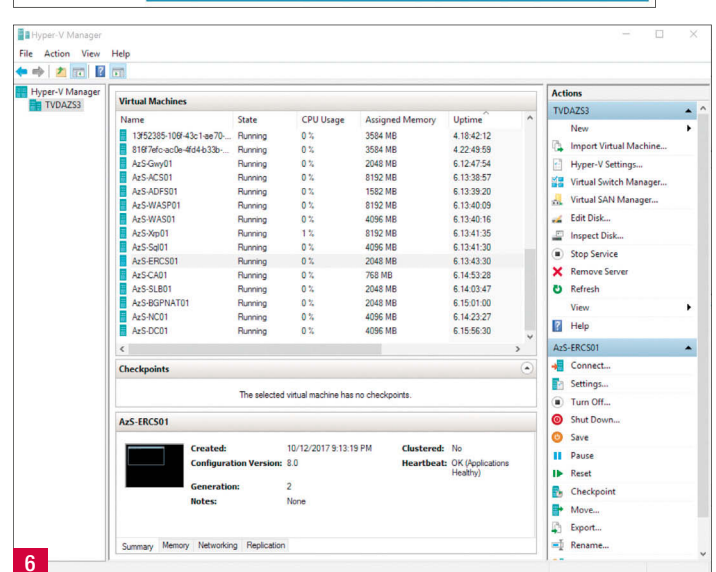
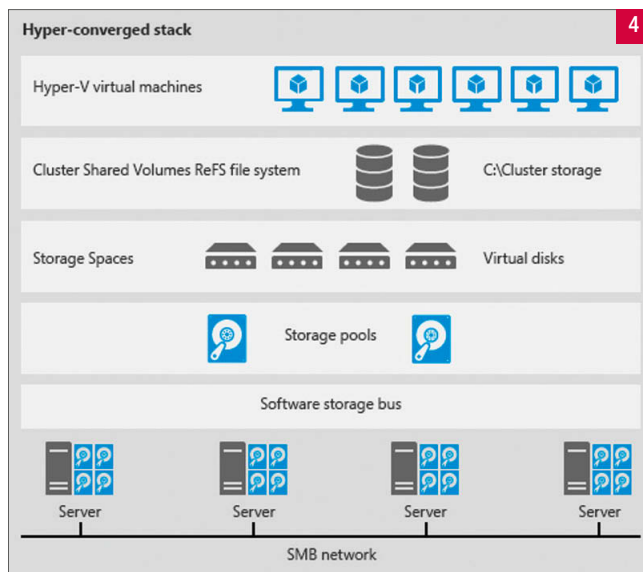
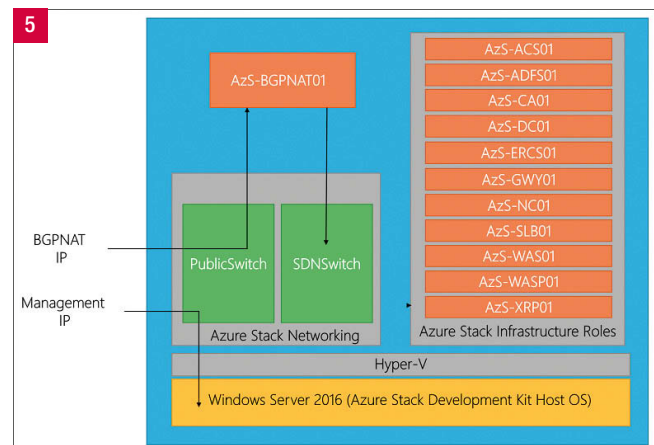
**AzS-GWY01** : Service de passerelles périphériques telles que connexions VPN site-to-site pour les réseaux utilisateurs.

**AzS-NC01** : Contrôleur Réseau, gère les services réseaux d'Azure Stack.

**AzS-SLB01** : Service de Répartition de Charge/multiplexeur d'Azure Stack pour à la fois l'infrastructure utilisateur et celle d'Azure Stack.

**AzS-SQL01** : Espace de stockage interne pour les rôles d'infrastructure Azure Stack.

**AzS-WAS01** : Portail Azure Stack administrateur et services Azure Resource Manager (ARM). Héberge le portail administrateur d'Azure Stack. Il permet de construire les offres et d'effectuer les tâches de « cloud operator ».



**AzS-WASP01** : Portail Azure Stack utilisateur (tenant) et services Azure Resource Manager (ARM). Ce portail est le portail utilisateur d'Azure Stack. Il est l'équivalent du portail Azure que nous connaissons.

**AzS-XRP01** : Contrôleur de gestion de l'infrastructure pour Microsoft Azure Stack, inclut les fournisseurs de ressources Compute, Network, et Storage.

Ces VM d'infrastructure sont visibles depuis Hyper-V, aux côtés des VM utilisateur (tenant) qui sont provisionnées selon les ressources (IaaS ou PaaS) créées depuis le portail Azure Stack

## Quels services sont disponibles aujourd'hui sur Azure Stack ?

### Fonctionnalités IaaS

Les services proposés dans Azure Stack sont fournis sous forme de « Resource providers ».

Par défaut sont présents les services IaaS suivants :

- **Storage** : permet de créer des ressources de type **Storage account**, dont les performances dépendent des disques physiques sous-jacents installés sur l'infrastructure Azure Stack.
- **Network** : permet de créer des ressources de type **Virtual Network**, **Virtual Network gateway**, **Network connection**, **Public IP**, **NIC**, **Network security group**, **Load balancer**, **Route table**, **Local network gateway**.
- **Health** : permet de monitorer la bonne santé de l'infrastructure Azure Stack ; version du produit, statuts et alertes de chaque resource provider, ou rôles d'infrastructure.
- **Compute** : permet de créer des ressources de type **Virtual machine**, **Availability Set**, **Virtual machine scale set**. Une VM scale set est un ensemble de VM provisionnées derrière un LoadBalancer. Des règles et des métriques (compteurs de performance) permettront d'en ajouter ou d'en retirer dynamiquement pour adapter l'infrastructure selon les besoins.
- **KeyVault** : permet de créer des ressources de type **Key vault**. Permet d'encrypter et de stocker des clés ou secrets comme des mots de passes dans un coffre-fort numérique. [7] [8]

### Fonctionnalités PaaS

Les **fonctionnalités PaaS** sont disponibles après l'installation de ressources providers additionnelles, et le provisionning de VM sous-jacentes managées qui offrent ces services.

**SQLAdapter** : permet de créer des ressources de type **SQL Hosting Server** ou **SQL Database**.

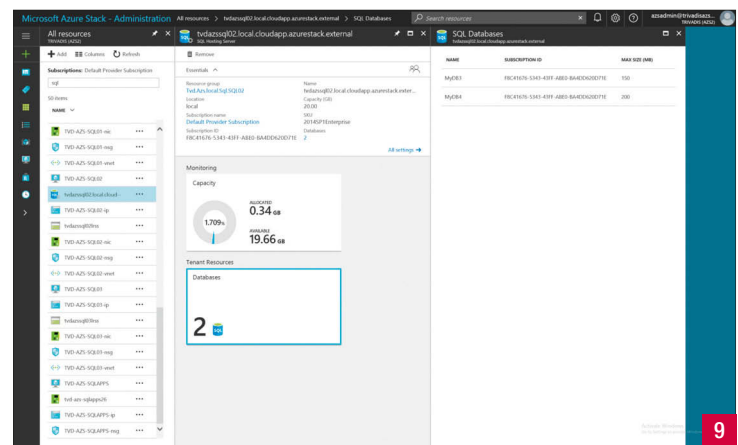
Après avoir installé le SQLAdapter, il faut instancier des VM SQL Server, puis les déclarer en tant que SQL Hosting Server. A partir de là, vos utilisateurs deviennent autonomes pour pouvoir créer des SQL Databases as a service au gré de leurs besoins et des quotas que vous aurez définis sur votre host. L'administration, l'application de patches de sécurités, ou de Services Packs sur le Host restent à la charge de votre administrateur SQL. [9]

**App Service** : permet de créer des ressources de type **App Service Plan**, **Web App**, **API App**, **Function App**.

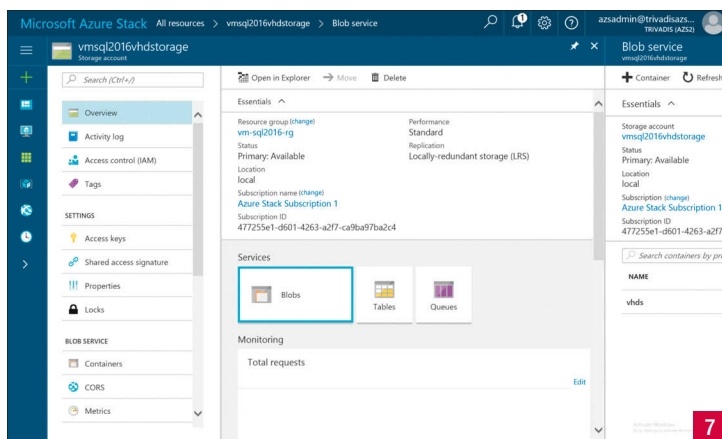
C'est probablement le service le plus évolué que nous ayons aujourd'hui dans Azure Stack. Avec ses fonctionnalités de montée en charge, ses capacités d'auto-réparation, de tolérance aux pannes, la gestion des mises à jour des VM sous-jacentes en mode PaaS, il s'appuie sur le même code et même mécanique que sa version Azure.

Pour pouvoir fournir ce service à vos utilisateurs, il faut indiquer au système quel doit être le VHD de base à utiliser pour créer les VM Web Worker (VM se trouvant derrière un App Service Plan). Vous indiquez au système combien d'instances vous souhaitez provisionner par taille de Web Worker, et le système s'occupe de tout.

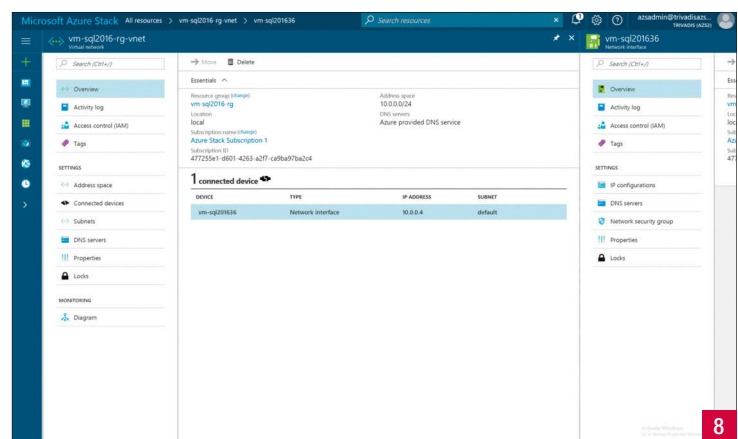
5 nouvelle VM d'infrastructure sont installées (dans la version ASDK) pour fournir tous les services annexes. (Editeur HTML, portail KUDU, service FTP, loadbalancers, stockage déporté, APIs,...). Par la suite, des Virtual machine scale sets sont créés par taille d'instances. Tous les composants sont alors téléchargés et installés automatiquement sur les VMs dont les derniers updates Windows, les Framework Java, serveur Tomcat, versions PHP, versions de .NET,...).



SQL Hosting server et SQL Databases



Storage Account



Virtual Network

Une fois qu'un rôle Web Worker est prêt, il est disponible dans le pool. Il devient utilisable par vos utilisateurs pour créer un AppService Plan et héberger des WebApps. L'instanciation d'un Web Worker rôle pouvant prendre 8h de temps, il faut prévoir de les créer à l'avance. Il faut bien comprendre qu'ils ne sont pas créés à la volée lorsqu'un utilisateur crée un nouvel App Service Plan. [10] [11]

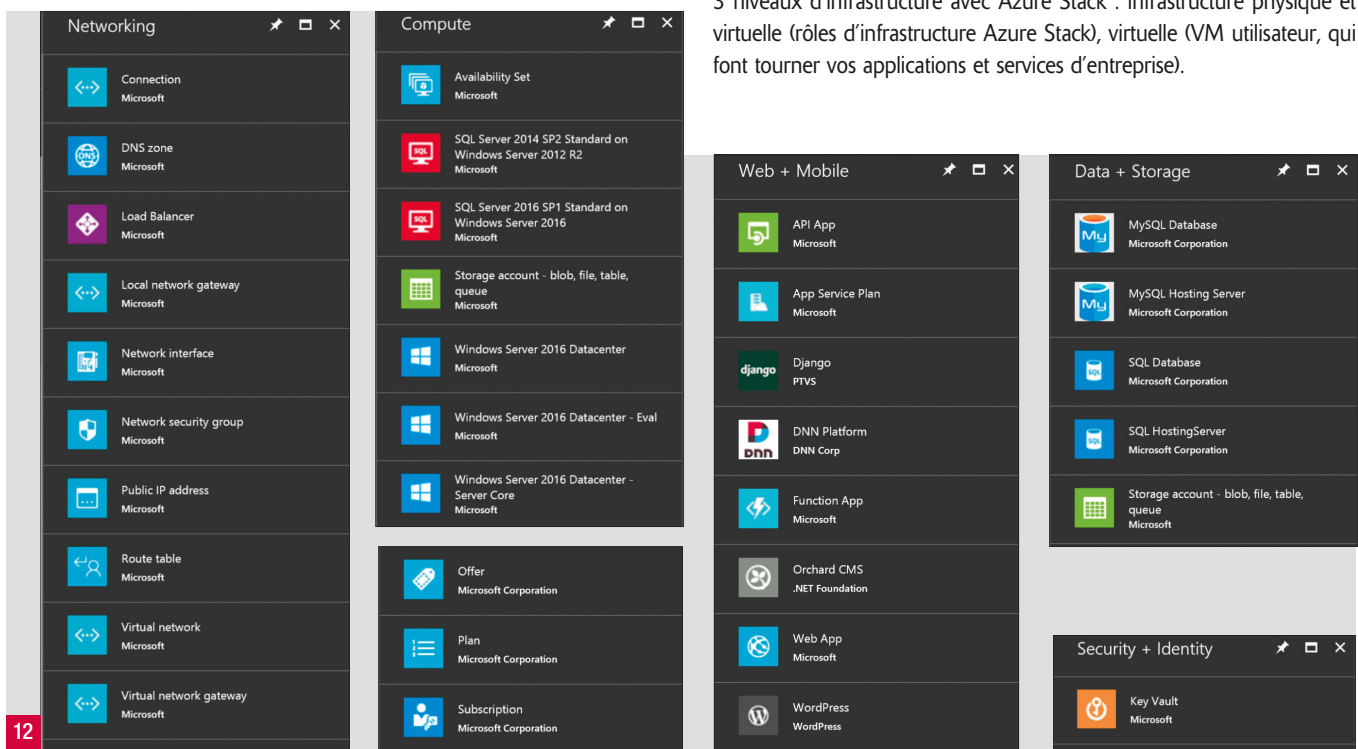
**MySQLAdapter** : Permet de créer des ressources de type **MySQL Hosting Server**, ou **MySQL Database**.

Le MySQLAdapter fonctionne sur le même principe que les SQLDataAdapter. A noter que pour la première fois, cette fonctionnalité était disponible dans la CTP3 d'Azure Stack, avant même d'être proposée en preview dans Azure. [12]

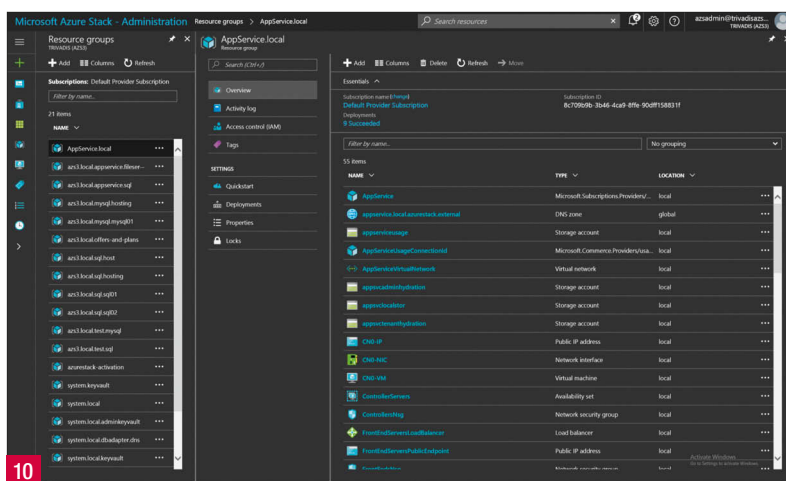
## La mise en oeuvre Operateur Azure Stack, un nouveau métier

Avec l'introduction d'Azure Stack naît un nouveau métier : l'Opérateur Azure Stack. C'est ce rôle qui va gérer l'infrastructure physique des environnements Azure Stack. Contrairement à Azure, où les opérateurs sont des employés de Microsoft, avec Azure Stack il faut prévoir une certaine organisation pour faire tourner et opérer l'environnement Azure Stack de bout en bout au quotidien. En cas de problème matériel l'opérateur Azure Stack travaillera en coordination avec le fournisseur de Hardware, et en cas de problème sur la partie software, avec Microsoft ou l'un de ses partenaires.

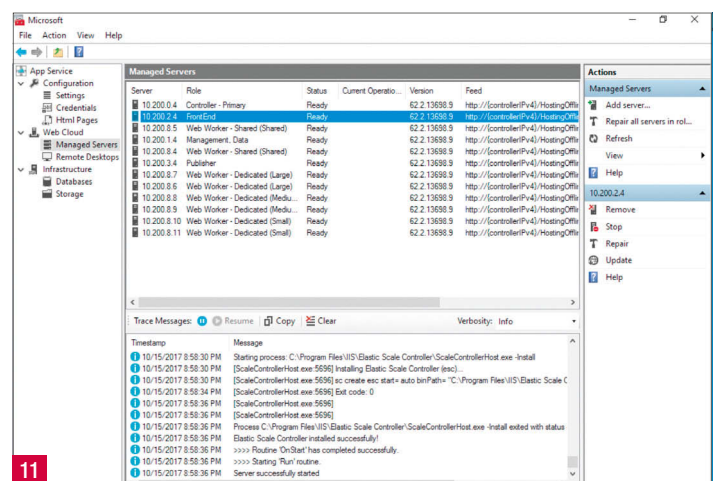
L'opérateur Azure Stack se fera accompagner par un architecte cloud Azure dans les phases importantes du projet, mais travaillera aussi avec les « Administrateurs Azure » qui assureront l'administration classique de l'infrastructure virtualisée IaaS créée au-dessus d'Azure Stack. Il y a donc 3 niveaux d'infrastructure avec Azure Stack : infrastructure physique et virtuelle (rôles d'infrastructure Azure Stack), virtuelle (VM utilisateur, qui font tourner vos applications et services d'entreprise).



Azure Stack services



Les dessous de la fabrique des App Service Plans



VM Contrôleur CNO-VM

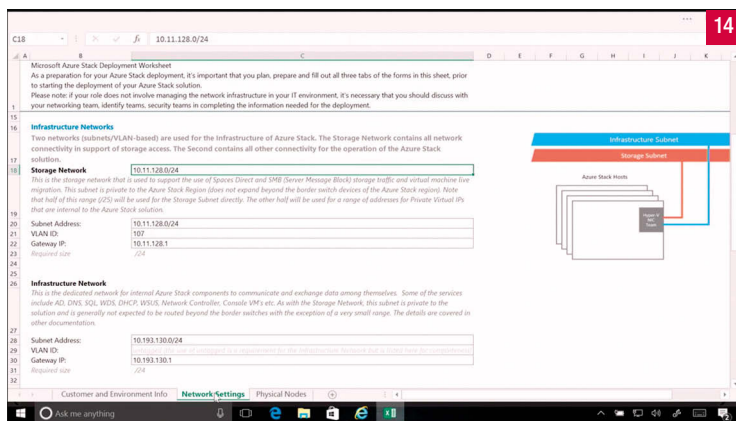
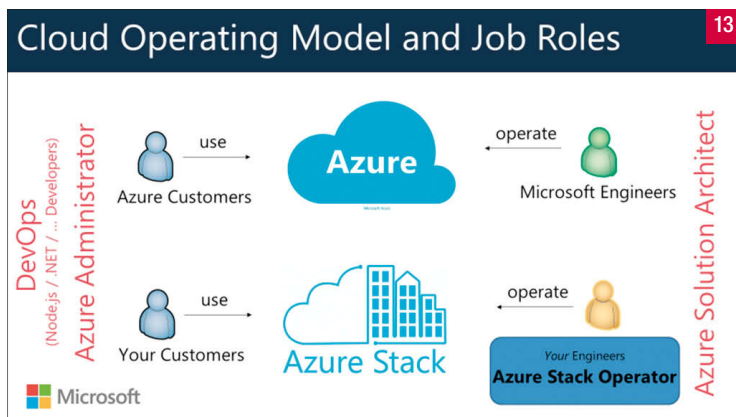
Selon la taille des organisations, il pourra cumuler plusieurs de ces rôles. Bref, le métier d'ingénieur système et réseau se diversifie et se modernise. [13]

## La planification

La première étape est de travailler avec votre fournisseur le Hardware pour planifier la taille du Système basé sur les besoins de vos applications et services, et vos prévisions de croissance. En fonction, il vous sera proposé une configuration de 4 à 12 nœuds, une configuration pouvant aller de 80 à 528 coeurs physiques, de 40 à 80 To de stockage et de 1 à 9 To de RAM. (selon les offres des fournisseurs, [2]). Ensuite, il vous faudra compléter un fichier Excel « Deployment Worksheet » en trois onglets, qui servira de document de référence pour la personnalisation de votre Azure Stack afin de l'intégrer avec votre infrastructure existante : Customer and Environment Info, Networking Settings, Physical Nodes. [14]

## La livraison

Le fournisseur de hardware livre le matériel dans votre datacenter avec des consultants qui viennent intégrer le système dans votre environnement. Il s'assure que le système fonctionne avant de vous le confier. Par défaut, cette intégration est très simple, car Azure Stack est un système complètement autonome et indépendant de vos systèmes de monitoring. Attendez-vous lors du premier boot de votre système à une attente de 8h, ou plus avant que vous ne puissiez commencer à le tester. Au-delà du paramétrage de base, vous pouvez décider avec votre intégrateur de le connecter au reste de votre infrastructure, que ce soit avec votre Active Directory ou votre Azure AD, ou enfin avec votre système de supervision global, System Center par exemple.



## Les deux portails d'Azure Stack

Avec Azure Stack, on a un produit très similaire à Azure, avec son portail en tout point identique à celui d'Azure, si ce n'est que son URL est par défaut : <https://portal.local.azurestack.external> au lieu de <https://portal.azure.com>. Par défaut aussi, le portail est vide, il y a aucune souscription, aucun service, tant que vous ne les avez pas définis avec le portail d'administration d'Azure Stack : <https://adminportal.local.azurestack.external>.

Notez que les URLs de ces deux portails peuvent être personnalisés sur une instance de production Azure Stack. [15] [16]

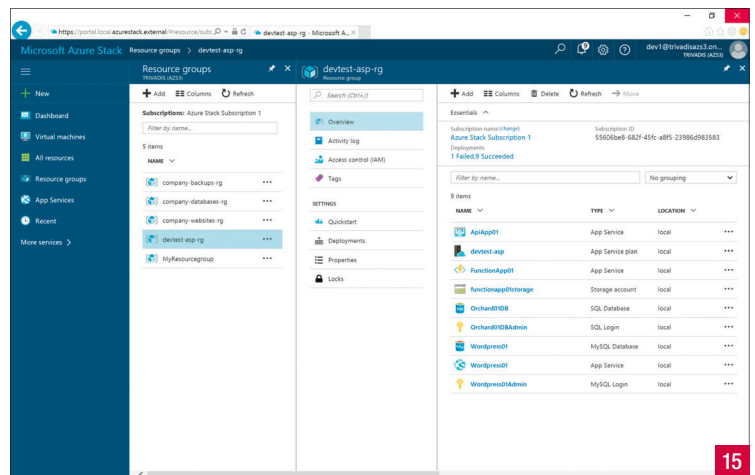
## La création d'Offres, Plans et Quotas

Pour créer des ressources dans Azure, vous devez posséder une souscription Azure s'appuyant sur un SKU, par exemple : Trial, PayAsYouGo, MSDN, Enterprise Agreement, ou CSP... Vous pouvez avec une même identité posséder plusieurs souscriptions Azure, de même type ou non. Avec Azure Stack, c'est exactement le même paradigme. L'opérateur cloud Azure Stack définit une liste d'offres, plans et quotas qui déterminent les types de ressources et les quantités qui pourront être créées par vos utilisateurs, selon le type de souscriptions qu'ils posséderont.

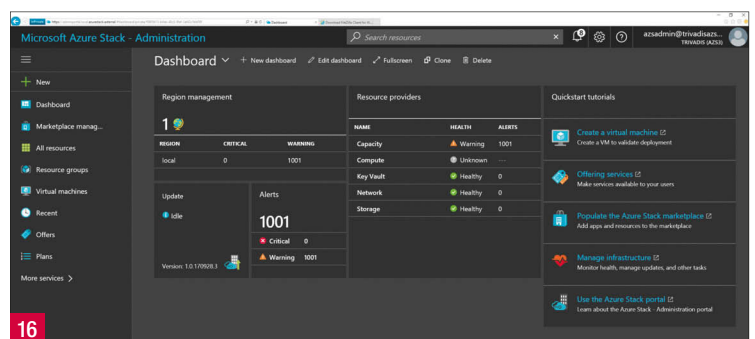
**Offre** : ce sont des modèles de souscriptions, que vous allez offrir à vos utilisateurs. Une offre détermine au final les types de ressources qui pourront être provisionnées, la quantité de ces ressources. Une offre peut être modifiée dans le temps par l'ajout d'add-on plans qui profitent aussi aux instances (souscriptions) déjà créées préalablement.

Exemples d'offres : Azure Stack Trial, Azure Stack for Developer, Azure Stack Standard, Azure Stack Gold, ...

**Plan** : les ressources que vous souhaitez attacher à une offre le sont à travers des Plans. Les plans sont une combinaison de types de ressources et de quotas. Comme mentionné, il y a aussi des Add-on plans. Exemples de Plan : Trial Compute Plan, Developer Compute Plan, Trial Network Plan, Trial Storage Plan, ...



Portail Azure Stack



16

Exemples de quotas : Trial Compute Quota, Developer Compute Quota, Trial Network Quota, Trial Storage Quota [17]

## La délégation d'offres.

Par défaut, lorsqu'on crée des offres dans Azure Stack, seul l'Azure Stack Service Administrator peut créer et attribuer des souscriptions (basées sur les offres) à des utilisateurs, à moins de rendre l'offre publique. Dans ce cas, n'importe quel utilisateur de l'AD ou AAD lié à Azure Stack peut créer des souscriptions.

Afin de permettre une certaine maîtrise tout en déléguant la création de souscriptions Azure Stack à d'autres, Microsoft a prévu la possibilité pour l'Azure Stack Service Administrator de créer des Offres dont la création de souscriptions sera déléguée à d'autres collègues (des Chefs de

Services par exemple). La délégation peut également être utilisée pour faire du « multi-tenant », c'est-à-dire partager son infrastructure Azure Stack entre plusieurs sociétés, avec des administrateurs délégués pouvant consommer des ressources de votre Azure Stack dans la limite des quotas liés aux offres que vous leur avez mises à disposition. Avec ce mécanisme, des hébergeurs traditionnels vont pouvoir proposer de l'Azure Stack « mutualisé » à leurs clients, ...comme solution de Cloud Privé. Chaque client pourra se connecter à sa souscription Azure Stack avec un compte de leurs AAD respectifs fédérés avec celui paramétré avec votre Azure Stack.

## Le Marketplace

Le Marketplace est le lieu où vos utilisateurs vont faire leur marché pour créer des ressources Azure Stack. Ce Marketplace est vide au moment où vous vous faites livrer votre Azure Stack.

Après la création des offres de souscriptions, la prochaine étape est de remplir le Marketplace avec vos propres solutions, applications et services, ou bien d'en importer depuis Azure, sachant que Microsoft, et ses partenaires ISV (Editeurs de logiciel) proposent une quantité très importante d'éléments modèles prêts à être importés dans votre Azure Stack.

[18]

Pour créer vos propres solutions, le plus simple est d'utiliser le «Marketplace publishing tool » car il vous permet de publier votre propre template ARM au format JSON dans la galerie Azure Stack, en y définissant toutes les propriétés : logos, description, nom de l'éditeur, etc.

[19]

Un template ARM s'appuie en outre sur un disque virtuel VHD existant ou VM (provenant d'Hyper-V, d'Azure, ou converti depuis VMWare ou fourni par Microsoft) qui aura été généralisé avec la commande Sysprep. Mieux, votre image VHD peut également être construite à partir d'un ISO et de son dernier cumulative update.

```
# Specify Azure Active Directory tenant name
```

```
$TenantName = "trivadisazs3.onmicrosoft.com"
```

```
Import-Module .\Connect\AzureStack.Connect.psm1
```

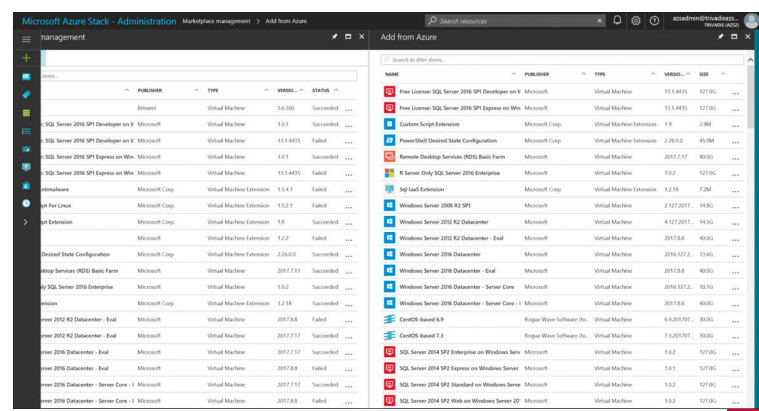
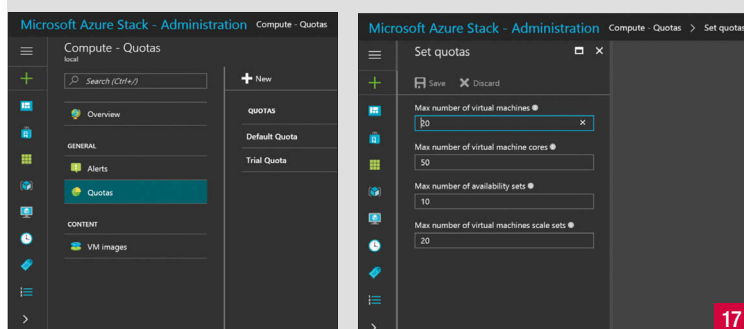
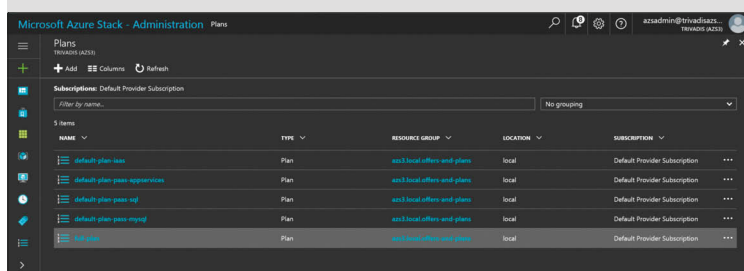
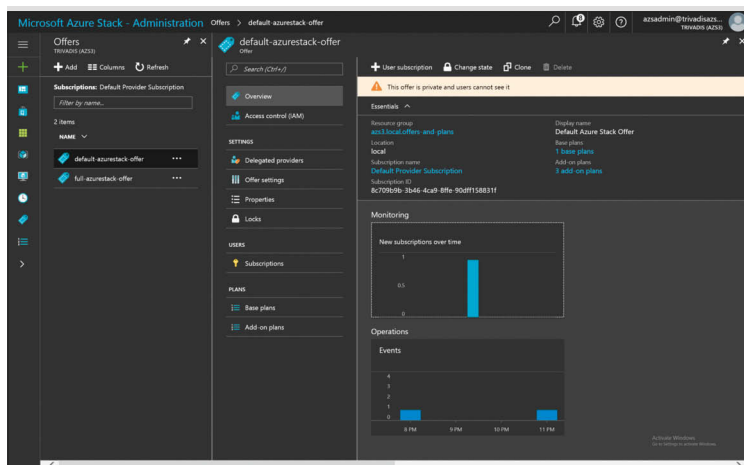
```
Import-Module .\ComputeAdmin\AzureStack.ComputeAdmin.psm1
```

## Add-AzureRMEnvironment

```
-Name "AzureStackAdmin" `
```

```
-ArmEndpoint "https://adminmanagement.local.azurestack.external"
```

```
$TenantID = Get-AzsDirectoryTenantId
```



```

-AADTenantName $TenantName `
-EnvironmentName AzureStackAdmin

# Authentification
Login-AzureRmAccount `
-EnvironmentName "AzureStackAdmin" `
-TenantId $TenantId

# Ajoute un VHD dans la liste des VM images + une entrée dans la galerie Azure Stack
Add-AzsVMImage `
-publisher "MicrosoftWindowsServer" `
-offer "WindowsServer" `
-version "1.1707.0" `
-osType "Windows" `
-sku "2016-StandardCore" `
-osDiskLocalPath "E:\INSTALL\VHD\1.1707.0\2016-StandardCore.vhd" `
-CreatGalleryItem $true

# Génère un VHD dans la liste des VM images à partir d'un ISO et de son dernier cumulative update
$ISOPath = "F:\INSTALL\ISO\14393.0.161119-1705.RS1_REFRESH_SERVER_EVAL_X64_FRE_EN-US.ISO"
$CUUpdatePath = "F:\INSTALL\ISO\Windows Server 2016\CU-20170104\windows10.0-kb3213522-x64_fc88893ff1f1be75cac5f5aee7ff1becee55c89dd.msu"
New-AzsServer2016VMImage `
-ISOPath $ISOPath `
-CUPPath $CUUpdatePath

```

Exemple de script PowerShell pour créer des VM Images

Pour créer des templates ARM au format JSON prêts à fonctionner dans Azure Stack, vous pouvez vous inspirer des exemples fournis par

Microsoft depuis le GitHub [Azure/AzureStack-QuickStart-Templates](https://github.com/Azure/AzureStack-QuickStart-Templates). Vous y trouverez une cinquantaine d'exemples. La seule différence avec les templates destinés à Azure sont les versions des types de ressources (api-versions) qui sont parfois différentes. [20]

Une fois le Marketplace peuplé, avec vos solutions personnalisées ou importées d'Azure, les utilisateurs peuvent créer des VM, ou utiliser des extensions. [21]

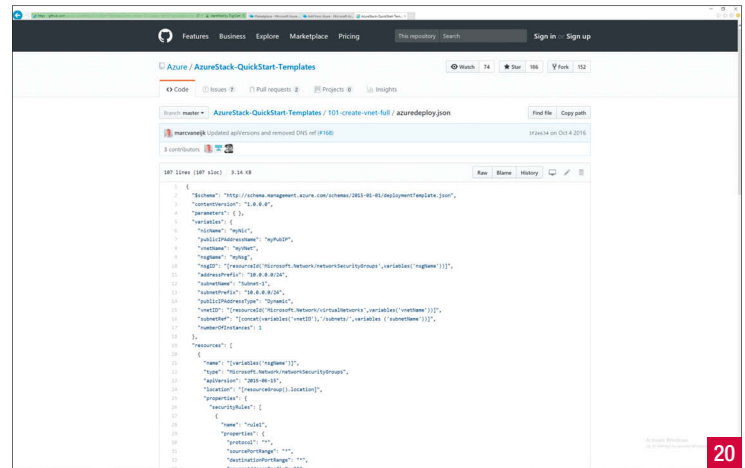
## Azure Stack, la plateforme cloud pour vous, développeur !

Azure Stack, permet aux développeurs de bénéficier d'une infrastructure à base de VM Windows ou Linux pour héberger leurs machines de développement, d'intégration ou de test, mais propose bien plus que cela !

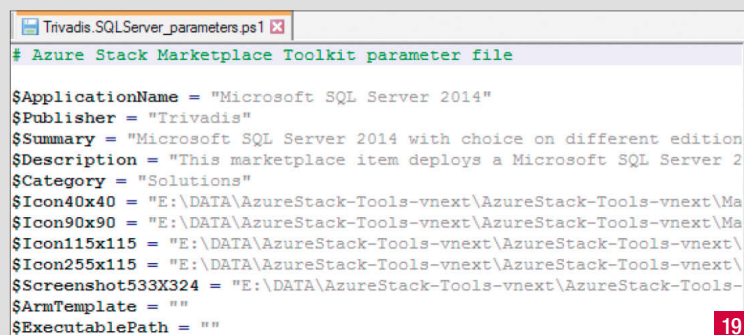
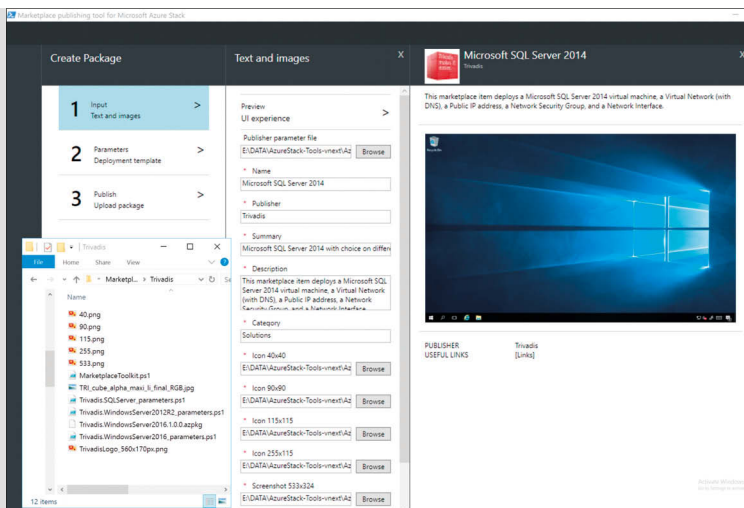
Azure Stack, expose aussi aux développeurs un certain nombre d'APIs offrant des services de stockage de fichiers, blob, tables, queues que vous pouvez exploiter depuis vos applications .NET, PHP, Python, Node.js ou Java. Il s'agit des mêmes API qui existent sur Azure. [22]

D'autre part, Azure Stack offre un modèle de déploiement et d'exécution d'application en plateforme as a service (PaaS), il s'agit des App Services Plans. Basés sur des VM managées, ils contiennent les dernières versions (mais aussi les précédentes) des frameworks Java, serveur Tomcat, versions PHP, versions de .NET. Il n'est pas nécessaire de les patcher ou de les administrer. La fabrique d'AppServices Plan s'occupe de tout.

Les App Service Plans permettent d'héberger des applications Web (Web App), des services web (API App) ou encore des fonctions serverless (Function App). Les App Service plans donnent la flexibilité d'allouer une

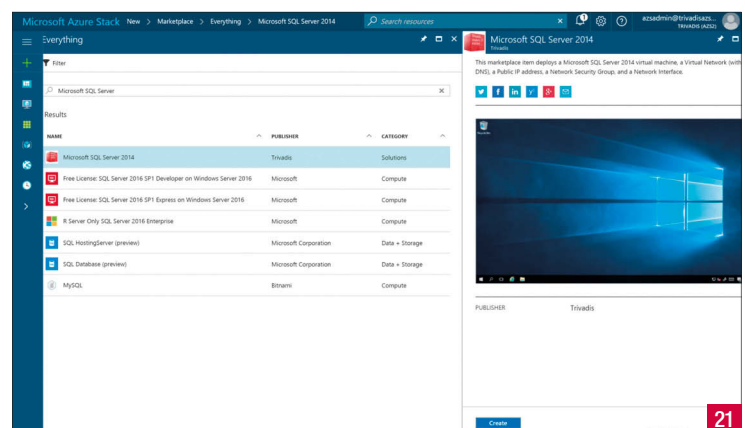


Exemple de template ARM en JSON pour Azure Stack



Création d'une solution avec le Marketplace publishing tool

programmez! - novembre 2017



ou plusieurs Apps sur un même plan. Ainsi, vous pouvez faire des économies dans vos environnements de dev et de test en partageant un plan pour l'hébergement de plusieurs apps.

Vous pouvez déployer des mises à jour de vos applications en bénéficiant de fonctionnalités de staging, roll-back, testing-in-production. Vous pouvez monitorer tous les aspects de vos applications en temps réel ainsi que consulter l'historique avec des logs détaillés.

## Le déploiement depuis votre environnement de développement

Microsoft a intégré depuis quelques années déjà la possibilité de déployer une application .NET, Java ou PHP dans Azure depuis Windows à partir de Visual Studio, IntelliJ, Eclipse. Pour un déploiement depuis une machine Linux on utilise Azure Command-Line Interface (Azure CLI). Le déploiement dans Azure Stack se fait avec ces mêmes outils et les mêmes plug-ins d'intégration créés initialement pour Azure.

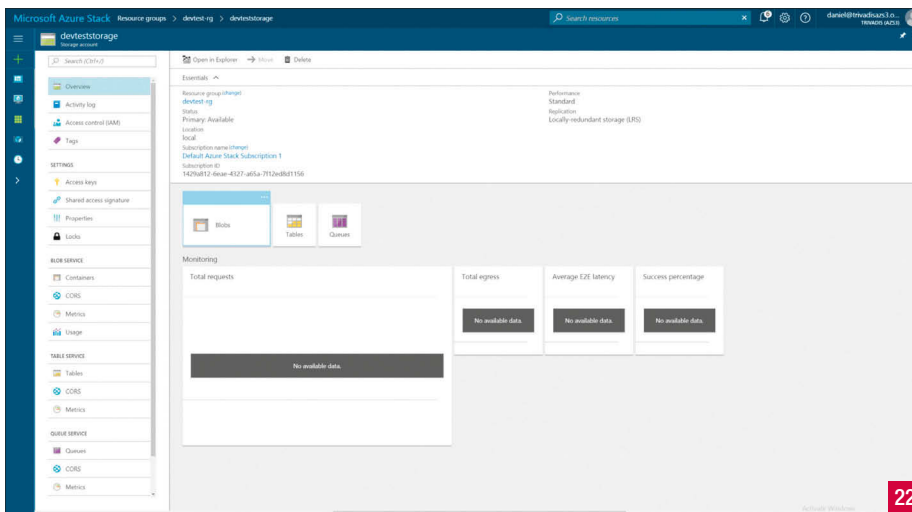
Depuis Visual Studio le déploiement vers un site Web hébergé dans Azure Stack se fait en quelques clics. Il suffit d'utiliser la fonctionnalité Publish Web App, et un assistant vous guide pour effectuer le déploiement sur un site existant ou vous propose d'en créer un nouveau. [23]

Choisissez la cible de type : Microsoft Azure App Service [24]

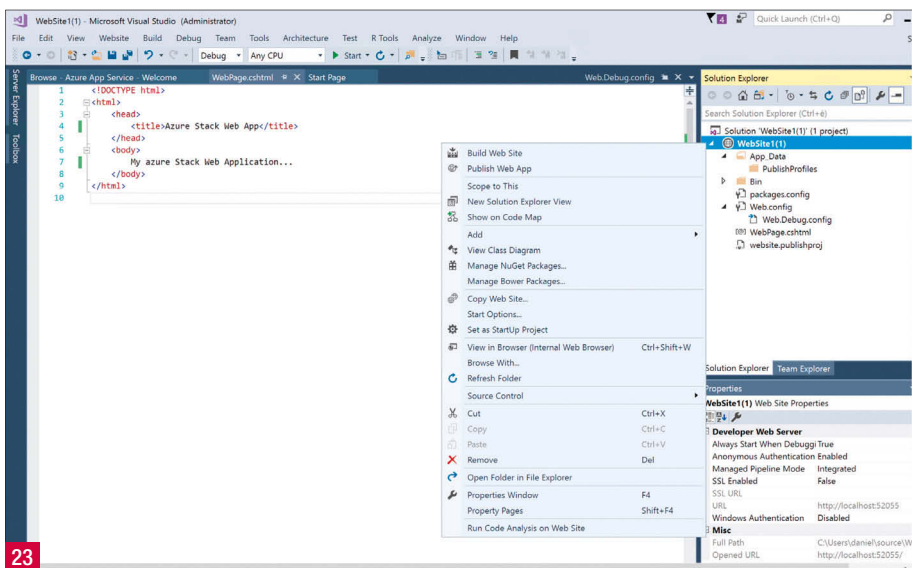
Il faut ensuite s'authentifier avec un compte qui possède au moins une souscription Azure Stack. Vous voyez apparaître vos différentes souscriptions Azure et Azure Stack. Dans l'exemple ci-dessous il s'agit bien d'une souscription Azure Stack. L'assistant propose de choisir une Web App déjà existante dans la souscription qui sera écrasée avec le contenu de notre application, ou de choisir d'en créer une nouvelle. [25]

Dans le cas d'une création, l'assistant propose d'utiliser un App Service Plan existant auquel on souhaite ajouter une nouvelle Web App, on peut choisir le type d'App (Web App dans cet exemple), ou bien décider de provisionner un nouveau serveur (App Service Plan) [26]

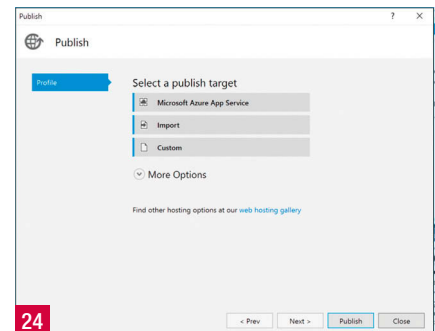
Dans le cas d'une création d'un nouveau App Service Plan, on choisit la taille du serveur qui hébergera cette première application (Web App). [27] Le serveur est créé en moins d'une minute, car il était déjà provisionné



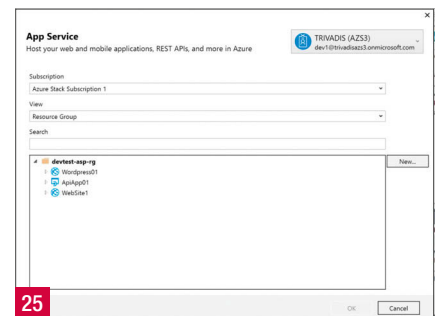
Storage Account : blobs, tables, queues



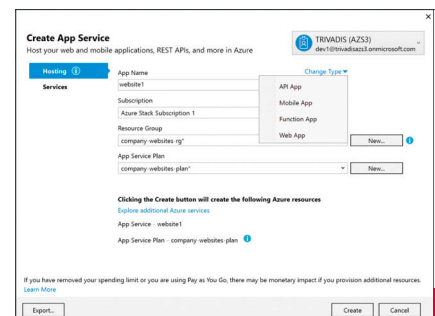
23



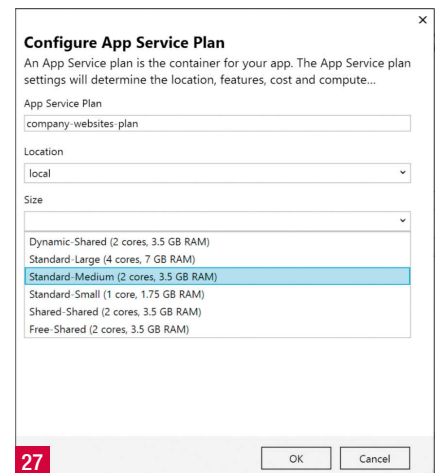
24



25



26



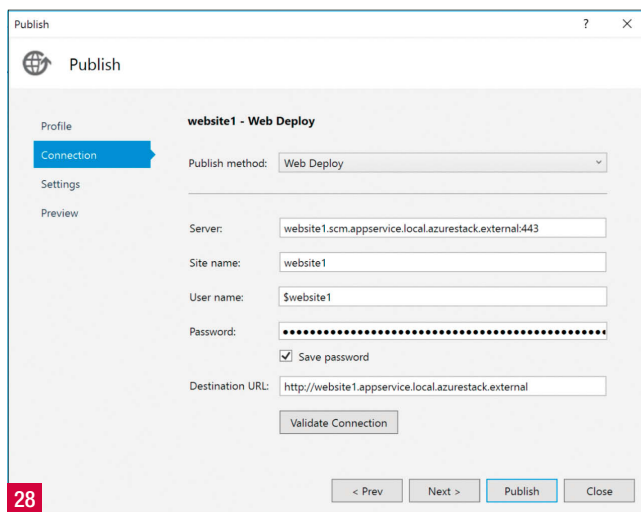
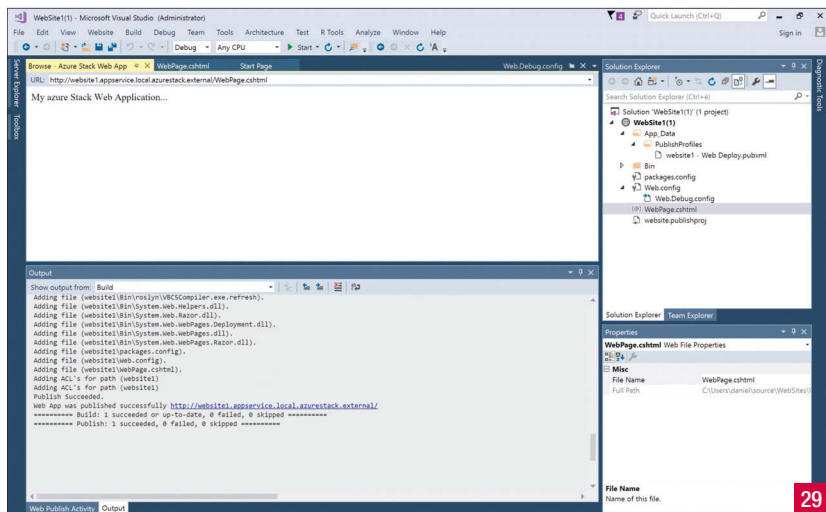
27

en amont par votre Azure Stack cloud operator. Cette opération consiste à réserver le serveur (dans la mesure où il y en avait de disponible dans la taille demandée), et à lui donner un nom dans votre souscription Azure Stack. Apparaît ensuite l'assistant de publication à proprement parler, qui effectue cette opération en un seul clic sur le bouton Publish. Lors des

déploiements suivants, vous arriverez directement sur cet écran, déjà pré-rempli. [28]

Au bout de quelques secondes à quelques minutes selon la taille de votre application, celle-ci est déployée. [29]

La suite dans **Programmez! 213**



# Abonnez-vous ! PROGRAMMEZ!

le magazine des développeurs

## Nos classiques

1 an ..... 49€\*

11 numéros

2 ans ..... 79€\*

22 numéros

ÉTUDIANT ..... 39€\*

1 an - 11 numéros

## Nos offres d'abonnements 2017

1 an ..... 59€

11 numéros + 1 vidéo ENI au choix :

• Arduino\*

Apprenez à programmer votre microcontrôleur

\* Tarifs France métropolitaine

• jQuery\*

Maîtrisez les concepts de base

2 ans ..... 89€

22 numéros + 1 vidéo ENI au choix :

• Arduino\*

Apprenez à programmer votre microcontrôleur

Offre limitée à la France métropolitaine

• jQuery\*

Maîtrisez les concepts de base

\* Valeur de la vidéo : 34,99 €

Toutes nos offres sur [www.programmez.com](http://www.programmez.com)

**Oui, je m'abonne**

ABONNEMENT à retourner avec votre règlement à :  
Service Abonnements PROGRAMMEZ, 4 Rue de Mouchy, 60438 Noailles Cedex.

- ☐ Abonnement 1 an : 49 €  
☐ Abonnement 2 ans : 79 €  
☐ Abonnement 1 an Etudiant : 39 €  
 Photocopie de la carte d'étudiant à joindre

- ☐ Abonnement 1 an : 59 €  
 11 numéros + 1 vidéo ENI au choix :  
☐ Abonnement 2 ans : 89 €  
 22 numéros + 1 vidéo ENI au choix :

- ☐ Vidéo : Arduino  
☐ Vidéo : jQuery

☐ Mme ☐ M. Entreprise : \_\_\_\_\_ Fonction : \_\_\_\_\_

Prénom : \_\_\_\_\_ Nom : \_\_\_\_\_

Adresse : \_\_\_\_\_

Code postal : \_\_\_\_\_ Ville : \_\_\_\_\_

**email indispensable pour l'envoi d'informations relatives à votre abonnement**

E-mail : \_\_\_\_\_ @ \_\_\_\_\_

☐ Je joins mon règlement par chèque à l'ordre de Programmez !

☐ Je souhaite régler à réception de facture

\* Tarifs France métropolitaine

# Web Components : Que du Bonheur ! N'est-ce pas?



Sara HARKOUSSE  
@Sara\_harkousse  
[www.saraharkousse.com](http://www.saraharkousse.com)  
Lead développeuse web chez  
**Dassault Systèmes**  
[www.3ds.com](http://www.3ds.com)  
Une des leaders Duchess France  
[www.duchess-france.org](http://www.duchess-france.org)  
Co-fondatrice et CTO chez  
**Maydée**  
[www.maydee.fr](http://www.maydee.fr)

*Il y a 6 ans, l'annonce des Web Components a fait le buzz parmi les développeur.euse.s Web. Certain.e.s estiment qu'ils vont révolutionner la manière dont nous créons des applications Web et j'en fais partie. L'idée est de permettre aux développeur.euse.s d'enrichir le HTML avec des éléments personnalisés, programmables et réutilisables, en s'appuyant sur des technologies standardisées.*

*En d'autres termes, les Web Components sont l'encapsulation d'une structure HTML et de comportements dans un composant qui peut être importé depuis une source externe dans le but d'une éventuelle réutilisation. Les Web Components ne sont pas l'histoire d'une seule spécification W3C (World Wide Web Consortium), mais quatre. Elles sont indépendantes et peuvent être utilisées séparément. Néanmoins, elles prennent tout leur sens utilisées conjointement. Elles étaient approuvées par tous les principaux navigateurs et elles peuvent être utilisées aujourd'hui dans vos applications (avec un polyfill si besoin).*



## Un peu d'histoire

L'annonce des Web Components a été faite par un Ingénieur Logiciel chez Google appelé Alex Russell, à l'occasion de la conférence Fronteers à Amsterdam, le 7 octobre 2011. L'idée est de créer nos propres balises et de tirer profit, au maximum, des standards de la plateforme Web dans le but de minimiser le risque d'adoption ; un risque bien moins élevé que celui de l'enfermement dans un framework ou une librairie qui peuvent être dépréciés plus rapidement que l'on imagine. Donc, au lieu d'attendre des années pour pouvoir utiliser des nouvelles spécifications du Web comme l'apport des balises `<img>`, `<form>`, `<video>`, `<audio>`, ..., pourquoi ne pas créer les nôtres ?

Une première version v0 des spécifications des Web Components sort quelques années plus tard. Elle est essentiellement poussée par les équipes de Google. Donc Chrome est le premier navigateur à les implémenter. Les autres navigateurs ont du mal à suivre étant pas très convaincus par la totalité des spécifications.

En 2016, il semble que les principaux navigateurs arrivent à un consensus, d'où la naissance de la version v1. Néanmoins, jusqu'à ce jour, ce n'est pas encore implémenté en totalité, d'où le besoin des polyfills fournis par les équipes de Polymer chez Google. Vous pouvez les trouver sur <https://github.com/webcomponents/webcomponentsjs> (Dans cet article, je ne m'étendrai pas plus sur leurs utilisations, puisque c'est bien détaillé dans le readme du répertoire en question).

Vous l'avez sûrement remarqué, les articles et les discussions sur les Web Components envahissent la toile ces dernières années. Ce nouveau concept divise encore. Certain.e.s sont partisan.e.s du concept, convaincu.e.s par son aspect révolutionnaire, tout comme d'autres hésitant.e.s par rapport à cette promesse pas encore totalement tenue à ce jour.

Dans cet article, le but est d'analyser, séparément, chaque spécification d'une manière pragmatique, dans l'espoir d'arriver à une conclusion

convaincante sur l'éventualité d'une utilisation en production ou pas, voire partiellement.

## Mise en place de l'environnement de développement

Tous les exemples de code figurant dans cet article sont disponibles sur Github [https://github.com/SaraHarkousse/Web\\_Components](https://github.com/SaraHarkousse/Web_Components). Je vous invite à tester par vous-même le code, voire à le modifier pour une meilleure compréhension. Dans ce cas, les outils nécessaires sont :

- Un IDE ; personnellement j'utilise Atom <https://atom.io/>. Visual Studio Code <https://code.visualstudio.com/> est très bien aussi, donc il peut correspondre à un choix très plausible de votre part ou tout simplement un Notepad++ ;
- Un serveur en local. Le protocole `file://` est suffisant pour certains exemples, mais pour d'autres non, notamment ceux où des fichiers HTML sont importés. On y reviendra plus tard afin d'en expliquer la raison. Personnellement, j'utilise « Web Server for Chrome » pour la simplicité de sa mise en place. Vous pouvez télécharger l'extension de Chrome sur <https://chrome.google.com/webstore/detail/web-server-for-chrome/ofhbbkphhbklhfoeikjpcbhmlcogjgh>. Une fois le serveur lancé, on peut tester le code sur n'importe quel navigateur.
- Git. Au cas où vous ne l'auriez jamais utilisé, je vous invite à jeter un coup d'œil sur ce guide <http://rogerdudler.github.io/git-guide/>. Afin de récupérer une version locale du code sur votre ordinateur :
  - Créez un nouveau dossier « Programmez » à n'importe quel endroit de votre disque dur.
  - Ouvrez le terminal et naviguez vers le nouveau dossier.
  - Lancez la commande suivante sur votre terminal :  

```
git clone https://github.com/SaraHarkousse/Web_Components
```
- Et finalement, de la bonne humeur.

Après l'installation de la petite extension serveur à partir du Google Web Store, vous verrez une icône du serveur Web. [1]

Cliquez sur l'icône pour ouvrir l'interface du serveur Web et choisissez un répertoire pour servir le contenu statique. Dans votre cas, c'est le dossier principal de votre projet.

En l'occurrence, c'est : `d:\Programme\Web_Components\VanillaJS\`. [2]

Accédez à <http://127.0.0.1:8887>. Vous y verrez tout le contenu de votre projet.

## Etat des lieux des 4 Standards derrière les Web Components

Dans la suite de l'article, je ne traiterai que la version v1 des spécifications. Le W3C propose 4 nouvelles spécifications pour définir les Web Components dans leur ensemble, dont vous trouverez une description détaillée ci-dessous :

### HTML Template

Afin de définir le squelette d'un nouvel élément HTML instanciable, la balise template vient à la rescousse. Elle peut contenir la structure DOM de ce nouveau composant.

Sa particularité réside dans le fait qu'elle permet de définir un fragment inerte d'HTML au moment du parsing (analyse syntaxique du DOM) de la page par le navigateur, qu'on peut utiliser ensuite comme base qu'on injecte dans le DOM à la demande. En termes de performance, c'est très puissant puisqu'on évite d'exécuter des scripts ou de faire des requêtes au serveur. On ne le fait qu'au moment opportun.

```
<template id="template">
  <div id="container">
    
  </div>
</template>
```

Ce squelette est alors contenu dans la balise template. Pour l'instancier, il faut cloner le contenu du template remonté par la fonction `querySelector`. C'est une étape nécessaire, puisque l'élément `div` dont l'id est « container » n'est pas un fils du nœud template, mais du `DocumentFragment` retournée par `template.content`.

En d'autres termes, puisque cette structure est inerte, elle n'est pas consi-

dérée comme étant dans le DOM. Donc la fonction `document.querySelector`, avec comme argument le sélecteur `#template`, ne remontera rien.

Notez que la duplication de la structure est faite en utilisant `cloneNode(true)`. Il est nécessaire de définir la valeur de la profondeur à `true`. Ainsi, tous les nœuds enfants et attributs seront clonés.

```
var template = document.querySelector('#template');
var clone = template.content.cloneNode(true);
```

Une fois le clone attaché à un autre nœud, son contenu prend vie. En d'autres termes, l'image 'logo.svg' sera récupérée et le style CSS sera appliqué au contenu du template.

```
var host = document.querySelector('#host');
host.appendChild(clone);
```

La balise template est supportée par tous les principaux navigateurs. Plus de détails sur <http://caniuse.com/#search=template> [3]

Vous pouvez retrouver l'intégralité du code source sur mon répertoire Github: [https://github.com/SaraHarkousse/Web\\_Components/tree/master/VanillaJS/Template](https://github.com/SaraHarkousse/Web_Components/tree/master/VanillaJS/Template) ou dans le répertoire

`d:\Programme\Web_Components\VanillaJS\Template`, si vous avez déjà cloné le projet Web\_Components sur votre disque dur (plus de détails dans la partie « Mise en place de l'environnement de développement »).

Afin de tester cet exemple, accédez à <http://127.0.0.1:8887/Template/index.html>

Vous êtes sensé.e.s voir l'icône des Web Components en rotation au milieu de votre page web.

### HTML Template : une spécification pas si révolutionnaire

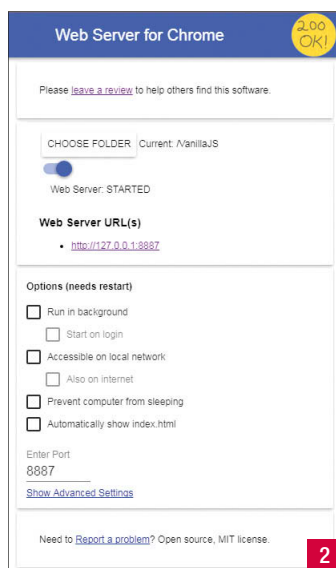
Si vous êtes habitué.e.s aux moteurs de templates comme Handlebars et Mustache, etc., vous pouvez vous attendre aux fonctionnalités suivantes :

- Les expressions avec les accolades comme `<h1>{{title}}</h1>` utilisables dans les blocs pour retourner les données courantes de votre modèle de vue.
- Les éléments de logique comme `#if`, `#unless`, `#with`, `#list`, `#each`, etc. utilisables pour afficher des blocs ou pas suivant des conditions, répéter certains blocs, etc.

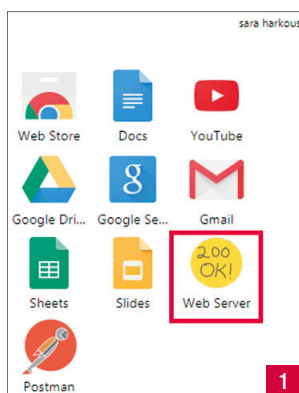
Vous serez malheureusement déçu.e.s. Ces concepts ne sont pas supportés par le standard de la balise template. Si vous êtes intéressé.e.s par l'utilisation de ces fonctionnalités, je vous recommande de jeter un coup d'œil à la librairie Polymer 2.0 et bientôt Polymer 3.0 (version bêta déjà disponible, pas encore prête pour utilisation en mode production).

De mon point de vue, je trouve que la spécification de la balise template est bien faite. Elle tient sa promesse, mais en aucun cas elle ne révolutionne la manière avec laquelle je crée les différentes structures de mes composants.

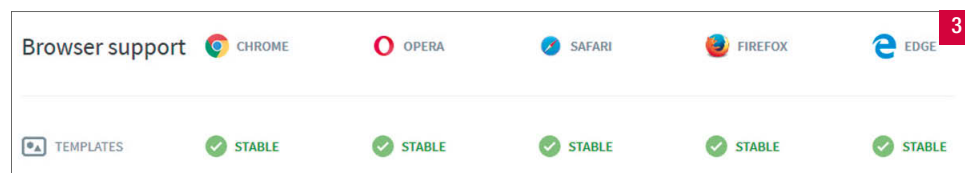
On peut imaginer la création de plusieurs templates pour plusieurs composants, mais comment garantir le cloisonnement du DOM et des styles CSS ?



Interface de Chrome Web Server installé sur Chrome



Chrome Web Server installé sur Chrome



Template : Compatibilité des navigateurs. Source : <https://www.webcomponents.org/>

## Shadow DOM

Brièvement, le Shadow DOM permet une portée locale pour une structure HTML et son style CSS associé, comme c'est le cas pour la balise `<iframe>` par exemple. Ce concept n'est pas nouveau puisque c'est déjà ce qui a été fait avec les balises `<video>` et `<audio>`, par exemple, dans les navigateurs modernes.

Dans une grande entreprise avec différentes équipes, responsables de création de composants d'une application Web, ça peut être vital. On peut éviter le besoin de conventions de nommages de classes CSS par composant, dans le but d'éviter d'impacter le reste des éléments d'une page HTML.

Le shadow DOM permet de masquer le squelette d'un élément dans le DOM par rapport aux autres éléments dans la page. Il n'est en aucun cas utilisé pour stocker des données sensibles dans le DOM, puisqu'il reste visible à l'utilisateur.ice via les outils de développement fournis par tous les navigateurs.

Pour le créer, rien de plus simple que le code ci-dessous. Dans cet exemple, on utilise `attachShadow` pour insérer un Shadow DOM, dans un nœud `host`. Ensuite, on définit tout le balisage et le contenu en utilisant la propriété `innerHTML`. Notez qu'on utilise les quotes inverses (backtick) pour revenir à la ligne pour plus de lisibilité (remplacer les backticks par des quotes simples sur IE11). On peut aussi imaginer l'utilisation de la balise `template` afin de contenir le balisage et le CSS. On y reviendra plus tard.

```
var host = document.querySelector('#host');
var root = host.attachShadow({mode: 'open'});
root.innerHTML = `
  <link rel="stylesheet" href="style.css">
  <div id="container">
    
  </div>
`;
```

Vous pouvez retrouver l'intégralité du code source sur mon répertoire Github:

[https://github.com/SaraHarkousse/Web\\_Components/tree/master/VanillaJS/ShadowDOM](https://github.com/SaraHarkousse/Web_Components/tree/master/VanillaJS/ShadowDOM) ou dans le répertoire

`d:\Programmez\Web_Components\VanillaJS\ShadowDOM`, si vous avez déjà cloné le projet `Web_Components` sur votre disque dur (plus de détails dans la partie « Mise en place de l'environnement de développement »).

Afin de tester cet exemple, accédez à <http://127.0.0.1:8887/ShadowDOM/index.html>. Vous êtes sensé.e.s voir l'icône des Web Components en rotation tout au milieu de votre page web.

Si on inspecte le résultat du code ci-dessus, on trouve ceci :

```
<div id="host">
  #shadow-root (open)
    <link rel="stylesheet" href="style.css">
    <div id="container">
      
    </div>
</div>
```

Le Shadow Root; (`#shadow-root` dans l'image ci-dessus): c'est ce qui est attaché à l'élément dont l'id est `host`. Il correspond au Shadow DOM. Techniquement, ce n'est pas un nœud comme les autres. C'est un type particulier de `DocumentFragment` (aussi utilisé pour contenir le segment inerte de la balise `template`). `DocumentFragment` représente une version légère de `Document` pour stocker une structure de nœuds, tout comme

un document standard. La principale différence est que les modifications qui y sont apportées n'affectent pas le Document conteneur, vu que le `DocumentFragment` ne fait pas partie de la structure du DOM.

Dans ce cas, l'élément dont l'id est `host` est appelé le Shadow Host. Il peut accéder à son Shadow DOM via l'appel de sa propriété `.shadowRoot`.

On appelle Shadow Tree tous les éléments attachés au Shadow Root. Ils sont inaccessibles du monde extérieur. Si on essaie la ligne ci-dessous, la variable `container` aura la valeur `null`.

```
var container = document.querySelector('#container');
```

## Les deux modes

Dans l'exemple, on a choisi le mode `open` pour la création du Shadow DOM « `{mode: "open"}` » comme argument de la fonction `attachShadow0`. Ce mode nous donne le moyen d'accéder au Shadow DOM via l'appel de la propriété `.shadowRoot` sur le Shadow Host dans notre code JavaScript. Il nous permet également d'accéder au Shadow Host depuis le Shadow DOM via l'appel de la propriété `.host`.

```
var host = document.querySelector('#host');
var root = host.attachShadow({mode: 'open'});
host.shadowRoot; // retourne le Shadow DOM
host.shadowRoot.host; // retourne le Shadow Host
```

Si on choisit le mode `closed` « `{mode: "closed"}` » comme argument de la fonction `attachShadow` pour créer le Shadow DOM, l'appel de la propriété `.shadowRoot` sur le Shadow Host dans notre code JavaScript retournera la valeur `null`. Le mode `closed` n'est pas un mécanisme de sécurité. Cela donne juste un faux sentiment de sécurité. Personne ne peut empêcher quelqu'un de modifier le fonctionnement de `Element.prototype.attachShadow0`.

Généralement, le mode `open` est plus utilisé puisqu'il donne la possibilité d'étendre le composant contenant le Shadow DOM par son auteur ainsi que les utilisateur.ice.s.

## Le style CSS dans Shadow DOM

Comme expliqué plus haut, le style appliqué à la structure HTML dans un Shadow DOM n'impacte pas le reste de l'application et vice-versa. Cela nous permet de choisir librement les noms des classes CSS. Donc la notion du Shadow DOM a introduit un nouveau sélecteur CSS appelé `:host`. Ce dernier cible la structure cachée dans le Shadow Root. On peut utiliser `:host(<selector>)` comme `:host(:hover)` afin de cibler différents états du même hôte. En accédant à l'URL <http://127.0.0.1:8887/ShadowDOM/index.html>, normalement, vous avez pu constater l'apparition d'une bordure de la couleur 'tomato' (Eh oui ! Ça existe ! ;) au hover sur la page.

Si on souhaite que notre composant soit personnalisable par les utilisateur.ice.s, on peut utiliser les variables CSS. Malheureusement, ce n'est pas encore supporté par tous les navigateurs.

```
// Style du Shadow DOM
:host {
  border-width: var(--border-width, 1em);
  border-style: solid;
  border-color: tomato;
}

// Moyen de personnalisation de la couleur de la bordure par l'utilisateur
```

```
#host {
  border-width: 2em;
}
```

Le Shadow DOM est un concept très puissant. Malheureusement, il n'est pas encore supporté par tous les navigateurs. [4]

Même si des polyfills existent, il est important de noter que l'utilisation d'un polyfill développé en JavaScript, qui a pour but de pallier à des comportements liés au CSS n'est pas optimale. Plusieurs développeurs de la communauté open-source ont remarqué des comportements différents d'un navigateur à un autre dans le cas d'utilisation de polyfills.

Par conséquent, il est plutôt recommandé de n'utiliser le Shadow DOM que s'il est nativement supporté. On peut ajouter une ligne de code (comme dans le code ci-dessous) pour détecter si le Shadow DOM est nativement supporté par le navigateur. Côté CSS, soit on utilise le sélecteur `:host` soit le sélecteur du conteneur du composant ; `#host` : dans notre exemple.

```
var shadowflag = HTMLElement.prototype.attachShadow;

var host = document.querySelector('#host');
var root = host;

if (shadowflag) {
  root = host.attachShadow({mode: 'open'});
}

root.innerHTML = `
<link rel="stylesheet" href="style.css">
<div id="container">
  
</div>
`;
```

Jusqu'à maintenant, vous pouvez créer des templates pour les structures HTML de vos composants, et garantir un cloisonnement du DOM et de styles CSS, mais vous allez vous retrouver avec des arborescences de DOM illisibles comme sur l'exemple ci-dessous : [5]

Pourquoi ne pas avoir une belle structure DOM plus lisible, où la fonctionnalité de chaque balise est plus identifiable et compréhensible comme ceci : [6]

## Custom Elements

Le standard des Custom Elements est, à mon sens, le plus important des 4 standards derrière les Web Components puisqu'il permet de créer de nouveaux éléments HTML.

Rien de plus simple que d'utiliser la fonction `define` de l'API native `customElements` comme sur l'exemple ci-dessous:

```
class MyComponent extends HTMLElement {
  // logique du composant
}

customElements.define('my-component', MyComponent);
```

La fonction `define` prend, en entrée, deux arguments :

- L'identifiant du composant qui correspond au nom de la balise du nouvel élément HTML qu'on souhaite créer ;
- Une classe ES6 qui hérite de l'interface `HTMLElement`. Elle va contenir toute la logique du composant.

On n'a plus qu'à appeler notre balise :

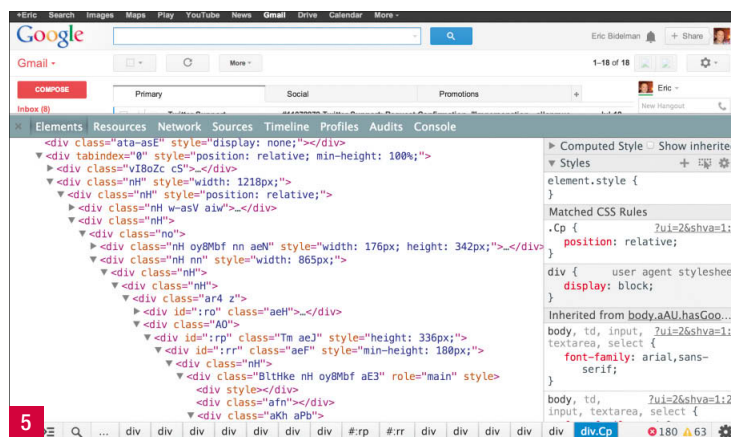
```
<my-component></my-component>
```

dans nos pages HTML. Il faut en revanche respecter quelques règles simples au moment du choix du nom du nouvel élément :

- Il doit comporter un tiret '-' (celui du 6). Cette règle est obligatoire. Elle permet de différencier les balises natives de la plateforme Web comme `<img>`, `<video>`, etc., des nouvelles créées par les développeurs. En plus, elle garantit une compatibilité ascendante, dans le cas où de nouveaux éléments sont ajoutés par les navigateurs.
- Il ne doit pas contenir de lettre capitale.
- Il ne doit pas correspondre aux noms réservés comme `color-profile`, `font-face`, etc.

```
<hangout-module>
  <hangout-chat from="Paul, Addy">
    <hangout-discussion>
      <hangout-message from="Paul" profile="profile.png"
        profile="118875919496626375791" datetime="2013-07-17T12:02">
        <p>Feel in' this Web Components thing.</p>
        <p>Heard of it?</p>
      </hangout-message>
    </hangout-discussion>
  </hangout-chat>
  <hangout-chat...</hangout-chat>
</hangout-module>
```

Une arborescence DOM plus jolie



L'arborescence de DOM de Gmail

Browser support					
CHROME	OPERA	SAFARI	FIREFOX	EDGE	
SHADOW DOM	STABLE	STABLE	STABLE	POLYFILL	POLYFILL
			DEVELOPING	CONSIDERING	

Shadow DOM : Compatibilité des navigateurs. Source : <https://www.webcomponents.org/>

Une des différences phares, de la spécification v1 des Custom Elements par rapport à la v0, est qu'elle nous offre la possibilité de tirer profit de toutes les nouveautés de ES6 (appelé aussi ES2015). C'est un réel avantage.

La classe `MyComponent` comprend la définition de plusieurs fonctions. Dans la partie qui suit, on va les détailler. On y trouve le constructeur ci-dessous :

```
constructor () {
  super();
  this.root = this;
  this.root.innerHTML = `
    <style>
      ...
    </style>
    <div id="container">
      
    </div>
  `;
  this._speed = 0;
}
```

Il est primordial d'appeler la fonction `super` dans le constructeur. Elle garantit l'héritage de toutes les propriétés et les attributs de l'interface `HTMLElement`. Dans cette fonction, on peut décrire l'état initial du composant (sa structure HTML et les propriétés internes).

## Les attributs

Le nouvel élément `<my-component>` correspond à l'icône des Web Components en rotation dont la vitesse est modifiable via un attribut appelé « speed ».

```
<my-component speed="2s"></my-component>
```

Vous pouvez retrouver l'intégralité du code source sur mon répertoire Github :

[https://github.com/SaraHarkousse/Web\\_Components/tree/master/VanillaJS/CustomElements](https://github.com/SaraHarkousse/Web_Components/tree/master/VanillaJS/CustomElements) ou dans le répertoire

`d:\Programmes\Web_Components\VanillaJS\CustomElements`, si vous avez déjà cloné le projet `Web_Components` sur votre disque dur (plus de détails dans la partie « Mise en place de l'environnement de développement »).

Afin de tester cet exemple, accédez à <http://127.0.0.1:8887/CustomElements/index.html>. Vous êtes sensé.e.s voir l'icône des Web Components en rotation au milieu de votre page web.

On peut définir plusieurs attributs à notre nouvelle balise. Ajoutons l'attribut « speed » :

```
get speed () {
  return this._speed;
}
set speed (val) {
  if (this._speed === val) return;
  this.setAttribute('speed', val);
}
```

Dans le code ci-dessus, on a ajouté :

- Un getter qui a pour rôle de retourner la valeur de l'attribut `speed` (qui est appelé, par exemple, à l'utilisation de la fonction `getAttribute` de

l'élément `<my-component>`). Dans notre cas, on a aussi défini la propriété `_speed`. Généralement, on choisit le même nom pour plus de lisibilité ;

- Un setter dans lequel on met à jour la valeur de la propriété `_speed` si différente de la courante, et on fait le lien entre la valeur de la propriété et la valeur de l'attribut en appelant la fonction `setAttribute`.

Vous l'avez sûrement compris, l'objet `this` correspond à l'élément `<my-component>`.

## Les réactions du composant

Maintenant, comment peut-on détecter le changement de la valeur de l'attribut `speed`? Le standard des Custom Elements offre la possibilité de définition de callbacks appelés aux moments intéressants du cycle de vie du composant. Ils sont appelés "les réactions".

Pour notre besoin, on va utiliser la fonction `attributeChangedCallback`. Elle est appelée à chaque fois qu'un attribut du composant change. Seuls les attributs listés dans la fonction `observedAttributes` sont concernés. Ce qui est plus performant.

On se retrouve, donc, avec le code suivant :

```
static get observedAttributes() {
  return ["speed"];
}

attributeChangedCallback (name, oldValue, newValue) {
  if (oldValue === newValue) return;
  switch (name) {
    case 'speed':
      this._speed = newValue;

      var img = this.querySelector("img");
      img.style.animationDuration = newValue;
      // ...
      // autre code
      break;
  }
}
```

La fonction `attributeChangedCallback` a trois arguments ; le nom de l'attribut concerné par un changement, l'ancienne valeur et la nouvelle. Dans notre cas, on a besoin de modifier la valeur de la durée de la rotation de l'image. Notez l'utilisation du `switch` dans le code ci-dessous. Certes, il n'est pas nécessaire puisqu'on n'a listé qu'un seul attribut dans le tableau retourné par la fonction `observedAttributes`, mais c'est un pattern très répandu parce qu'on a généralement besoin de déclarer plusieurs attributs.

Il existe encore d'autres réactions d'un Custom Element comme :

- `connectedCallback`: appelé chaque fois que le composant est inséré dans le DOM.
- `disconnectedCallback`: appelé chaque fois que le composant est supprimé du DOM.
- `adoptedCallback`: appelé chaque fois que le composant est déplacé dans un nouveau document.

## Hériter d'un élément existant

On peut aussi étendre un élément existant pour lui rajouter des comportements comme un bouton par exemple :

```
class CustomButton extends HTMLElement {
  // logique du composant
}
customElements.define('custom-button', CustomButton, {extends: 'button'});
```

Dans ce cas, la classe CustomButton hérite de l'interface HTMLElement. De cette manière, on hérite de toutes les propriétés et attributs de l'élément <button>. Notez que la fonction define prend, en entrée, un troisième argument {extends: 'button'}. On est obligé de spécifier l'élément qu'on veut étendre même si on a déjà déclaré l'héritage de l'interface HTMLElement parce qu'il s'avère que plusieurs éléments peuvent hériter de la même interface et donc ce n'est pas un point différenciateur. Une autre particularité, c'est qu'on ne peut pas appeler notre balise comme ceci :

```
<custom-button></custom-button>
```

mais plutôt :

```
<button is="custom-button"></button>
```

avec l'addition d'un nouvel attribut « is » qui correspond au nom choisi du composant. Ce comportement est un peu décevant. Par exemple, Safari a décidé de ne pas l'implémenter. Mon conseil est donc de ne créer que des éléments "from scratch" afin de bénéficier des noms de balises personnalisés.

La spécification Custom Elements n'est pas supportée par tous les navigateurs. Plus de détails sur <https://caniuse.com/#search=custom%20elements%20v1> [7]

## Communication

Comme défini plus haut, on peut modifier la vitesse de rotation de l'icône des Web Components via l'attribut « speed » de l'élément <my-component>. La question qui se pose maintenant est : comment puis-je déclencher cette modification par un autre élément ? Y a-t-il un moyen de communication entre les éléments HTML ?

On peut imaginer le cas de figure suivant : on utilise un input de type range pour définir la vitesse qu'on souhaite appliquer à la rotation de l'icône. [8]

On peut facilement dire qu'on peut toujours utiliser la fonction querySelector pour trouver l'élément <my-component>, puis utiliser la fonction setAttribute pour changer la valeur de l'attribut « speed ». Est-ce la meilleure solution ? La réponse est non.



Exemple de communication entre éléments HTML

L'utilisation des Web Components nous encourage à avoir une architecture modulaire pour nos applications Web avec des composants indépendants qui communiquent entre eux par le biais des événements. Un des avantages à cela est l'impact des erreurs sur l'application. Si un composant tombe en panne, l'application peut continuer à fonctionner, et par conséquent, l'utilisateur peut toujours, au moins à en utiliser une partie. Alors que si on utilise directement les instances des éléments afin de modifier certaines de leurs propriétés, on risque de faire crasher toute l'application à la moindre erreur.

La communication entre éléments se fait d'une manière très simple et intuitive.

```
customElements.whenDefined("my-component").then(() => {
  var comp = document.querySelector("my-component");
  comp.addEventListener("speed-change", (data) => {
    comp.speed = data.detail.val;
  });
});

var inputWithRange = document.querySelector("input[type='range'];
inputWithRange.addEventListener("change", (event) => {
  var newevent = new CustomEvent("speed-change", {
    "detail": {"val": event.target.value + 's'}
  });
  inputWithRange.dispatchEvent(newevent);
});
```

On utilise la fonction addEventListener pour lier le code de changement de valeur de l'attribut « speed » à l'évènement appelé « speed-change », qui est, d'un autre côté déclenché au changement de valeur de l'input de type range. Afin de personnaliser les noms des événements, on utilise CustomEvent pour le faire.

Maintenant, voyons comment réutiliser un composant dans un fichier HTML externe qui sera appelé dans le fichier principal, en l'occurrence notre index.html.

## HTML Imports

La spécification HTML Imports permet d'importer un fichier HTML comme on le fait pour les CSS et les fichiers JavaScript. On peut déclarer l'import comme ci-dessous dans notre fichier index.html.

```
<link rel="import" href="component.html">
```

On peut facilement imaginer le contenu du fichier component.html : il peut contenir le squelette du composant en utilisant la balise template ou pas, le style CSS du composant et le code JavaScript qui récupère l'élément et l'attache au nœud souhaité dans le DOM.

Browser support					
	CHROME		OPERA		SAFARI
	FIREFOX		EDGE		
	CUSTOM ELEMENTS		STABLE		STABLE
	POLYFILL		DEVELOPING		POLYFILL
			CONSIDERING		

Custom Elements : Compatibilité des navigateurs. Source : <https://www.webcomponents.org/>

```
<link rel="stylesheet" href="style.css">
<template id="template">
  <div id="container">
    
  </div>
</template>
<script src="script.js"></script>
```

Afin de récupérer le contenu du fichier component.html, on utilise `document.querySelector` pour remonter le contenu du HTML Import et puis avec la même méthode, on récupère le nœud root du composant.

```
var link = document.querySelector("link[rel='import']");
var content = link.import;
var el = content.querySelector("#container");
var clone = el.cloneNode(true);
var host = document.querySelector("#host");
host.appendChild(clone);
```

Vous pouvez retrouver l'intégralité du code source sur mon répertoire Github :

[https://github.com/SaraHarkousse/Web\\_Components/tree/master/VanillaJS/HTMLImports](https://github.com/SaraHarkousse/Web_Components/tree/master/VanillaJS/HTMLImports) ou dans le répertoire `d:\Programme\Web_Components\VanillaJS\HTMLImports`. Afin de tester cet exemple, accédez à <http://127.0.0.1:8887/HTMLImports/index.html>. Vous êtes sensé.e.s voir l'icône des Web Components en rotation au milieu de votre page web.

Si vous testez ce code sur Chrome, vous aurez l'alerte suivante :  
*"[Deprecation] Styling master document from stylesheets defined in HTML Imports is deprecated, and is planned to be removed in M65, around March 2018. Please refer to https://goo.gl/EGXzpw for possible migration paths."*

En d'autres termes, jusqu'à mars 2018, le fait d'incorporer du style CSS au sein du HTML importé, quelle que soit la manière (via la balise `<style>` ou style externe `<link rel="stylesheet">`) est déprécié. Passé cette date, cette pratique ne sera plus supportée.

Dans une architecture modulaire, c'est assez logique de créer un fichier HTML par composant dans le but de contenir la structure DOM, le style CSS et le code JavaScript qui ajoute une certaine logique au composant comme on l'a fait dans l'exemple component.html. On s'attend à ce que le style dans le fichier style.css n'ait d'effet que sur les éléments DOM dans le fichier component.html alors que ce n'est pas le cas. En fait, on peut même incorporer du CSS qui s'applique sur les éléments DOM du fichier index.html (qui est appelé master document dans l'alerte soulevé par Chrome).

Il est assez facile de contourner la question pour ne plus avoir l'alerte en

déplaçant manuellement ou par script l'incorporation du fichier style.css vers index.html mais ce n'est pas la solution souhaitée.

Plusieurs raisons motivent la décision des équipes Chrome. Entre autres, l'introduction de ce comportement à rendu Blink; le moteur de rendu utilisé par Chromium très complexe. En outre, c'est assez déroutant pour les développeur.se.s Web de voir d'autres éléments dans l'index.html impactés par le style des composants définis dans différents fichiers HTML. [9]

La spécification HTML Imports n'est pas supportée par tous les navigateurs. Plus de détails sur <http://caniuse.com/#search=html%20import>

Comme on peut le remarquer, la spécification HTML Imports divise. Safari et Firefox, ne sont pas du tout convaincus par ce standard qui a plusieurs défauts. Par exemple, dans le cas d'une application Web qui importe plusieurs fichiers HTML par composant, il est indispensable d'ajouter une étape de build qui va concaténer tous les fichiers HTML pour n'en avoir qu'un seul, dans le but d'éviter de marteler le serveur avec plusieurs requêtes.

Réellement, chaque HTML Import correspond à un `HttpRequest` (donc le protocole `file://` n'est pas suffisant pour faire fonctionner le code utilisant HTML Imports). L'équipe Polymer chez Google, avait résolu ce problème en fournissant un outil de build appelé `Vulcanize`, qui a comme but de concaténer tous les fichiers HTML voulant être importés en un seul, dans le but de faire qu'une seule requête au serveur. A l'avenir, des technologies telles que `HTTP/2` et `Server Push` déprécieront l'utilité d'un tel outil en mode production.

La décision de déprécier l'incorporation du style CSS dans un fichier importé via HTML Import dans un premier temps n'est que la première étape vers la dépréciation complète des HTML Imports ; une fois un successeur concret on obtient un consensus. Cette première dépréciation est un tremplin pour cela.

Il s'avère que ES module correspond à ce nouveau successeur de HTML Imports tant attendu. Il enrichit le Javascript avec deux nouveaux mots clés : `import` et `export`.

```
// export
export const Component = // ...

// import
import {Component} from './components/custom-component.js';
```

Le mot `export` ajoute un tag à la fonctionnalité qu'on souhaite importer via le mot `import` dans un autre fichier. La différence ici c'est qu'on ne peut utiliser que du code JavaScript, donc plus de moyen natif d'importer un fichier HTML dans un autre comme on le faisait avec HTML Imports. On sera encore dans l'obligation d'utiliser des librairies tierces si on souhaite importer des fichiers HTML type `requirejs` ou `webpack`.

Browser support					
	CHROME	OPERA	SAFARI	FIREFOX	EDGE
HTML IMPORTS	STABLE	STABLE	POLYFILL ON HOLD	POLYFILL ON HOLD	POLYFILL CONSIDERING

HTML Imports : Compatibilité des navigateurs

## Mélangeons le tout

Jusqu'à maintenant, on a analysé les spécifications W3C séparément dans le but de montrer qu'on peut n'utiliser que certaines d'entre elles si on le souhaite, mais on peut aussi les mélanger afin de montrer tout leur potentiel.

Ci-dessous un exemple de code d'un fichier my-component.html dans lequel on trouve la balise template qui contient la structure du composant, le style CSS associé dans le fichier style.css ainsi que la logique en JavaScript dans le fichier script.js (il contient la classe de définition du Custom Element <my-component>). Je suis partisane d'avoir des fichiers séparés par langage.

```
<template id="template">
  <link rel="stylesheet" href="style.css">
  <div id="container">
    
  </div>
</template>

<script src="script.js"></script>
```

D'un autre côté, dans le fichier index.html, je ne fais qu'importer mon composant (my-component.html) et puis utiliser la balise définie. N'est-ce pas très propre, très lisible ?

Et surtout facilement évolutif. Plus intéressant encore, c'est qu'on peut utiliser ces nouvelles balises, nativement construites, avec n'importe quel framework. Par exemple, pour les fans d'Angular, et surtout par la structure et organisation qu'il peut apporter à un projet, ils peuvent l'utiliser mais avec des Web Components. Le coût de la prochaine migration à la nouvelle version d'Angular sera forcément plus faible.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Web Component</title>
  <style>
    body {
      display: flex;
      // ...
    }
  </style>
  <link rel="import" href="my-component.html">
</head>
<body>
  <my-component speed="2s"></my-component>
  <input type="range" name="range" min="1" max="10" step="1" />
  <input type="text" name="text" value="" />
</body>
</html>
```

Vous pouvez retrouver l'intégralité du code source sur mon répertoire Github :

[https://github.com/SaraHarkousse/Web\\_Components/tree/master/VanillaJS/WebComponent](https://github.com/SaraHarkousse/Web_Components/tree/master/VanillaJS/WebComponent) ou dans le répertoire `d:\Programmes\Web_Components\VanillaJS\WebComponent`. Afin de tester cet exemple, accédez à <http://127.0.0.1:8887/WebComponent/index.html>.

Vous êtes sensé.e.s voir l'icône des Web Components en rotation au milieu de votre page web avec un input de type range qui permet de changer la vitesse de rotation de l'icône et un input de type text qui affiche cette valeur.

## CONCLUSION

Les spécifications W3C prennent tout leur sens lorsqu'elles sont utilisées conjointement. D'ailleurs, l'une des idées répandues, est que pour faire des Web Components, on doit utiliser la totalité des standards ce qui, par conséquent, rebute certains utilisateur.trice.s.

Surtout que, comme on l'a vu, le support n'est pas encore complet par tous les navigateurs, et il subsiste encore pas mal de défauts et bugs à résoudre.

Donc au lieu de ne pas les utiliser, par conséquent, retarder leur développement et acceptation par tous les navigateurs, mieux vaut en tirer profit séparément.

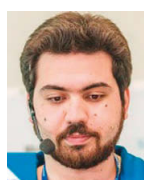
- Utiliser la balise template dans le cas des composants simples, sinon privilégier d'autres bibliothèques de templating comme Handlebars.
- N'utiliser le Shadow DOM que s'il est nativement supporté par le navigateur.
- Ne pas se priver d'utiliser les Custom Elements, même avec les polyfills. Ces derniers sont bien faits.
- Ne pas utiliser les HTML Imports. Privilégier les ES modules en attendant une solution meilleure pour l'import des fichiers HTML.

Ci-dessous une sketchnote, d'un de mes derniers talks à ce sujet, qui a été faite par une jeune femme très talentueuse appelée Malwine lors de la conférence RuhrJS en Allemagne. Elle résume parfaitement les conclusions principales de cet article.



# Windows Subsystem for Linux (WSL), l'outil astucieux pour intégrer Linux à Windows

## Installation pas-à-pas et utilisation avancée



Igor Leontiev @VISEO  
**MVP Azure**  
Architecte Azure &  
TechLead Blockchain



Vasyl Kutsyk @VISEO  
Développeur Java full  
stack

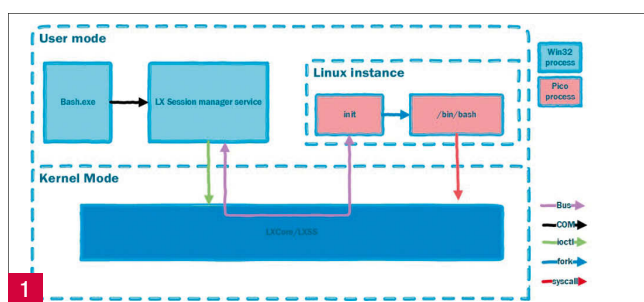


Vincent Thavonekham  
@VISEO  
MS Regional Director  
**MVP Azure**

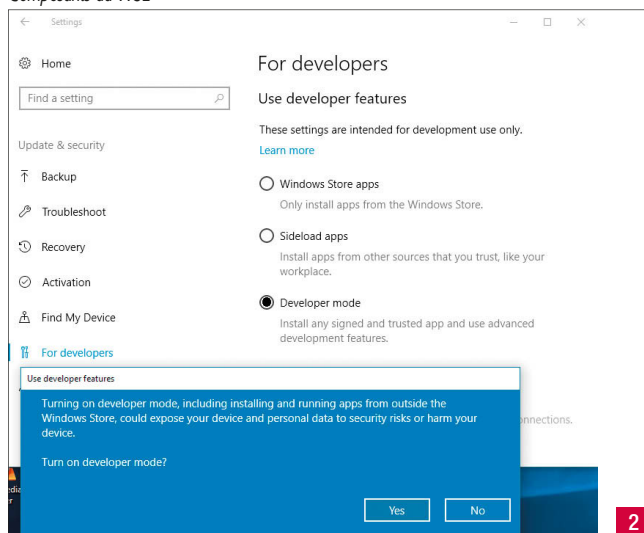
Dans cet article, nous allons voir par quelle magie Microsoft est parvenu en 2016 à faire fonctionner du Linux sur des OS Windows, via **Windows Subsystem for Linux (WSL)**. Comment activer cela pas-à-pas et présenter un « Hello World »? Enfin, voir des exemples d'utilisation plus avancés qui parleront autant aux développeurs qu'aux SysAdmin.

Windows Subsystem for Linux (WSL) – représente un ensemble de composants qui permettent d'exécuter du code natif Linux ELF64 sur du Windows ! Il comporte à la fois les composants « user mode » et ceux du « kernel mode ». Focus sur quelques fonctionnalités principales :

- Le service "User mode session manager" permet de gérer le cycle de vie de l'instance Linux ;
- Les drivers "Pico provider drivers" (lxss.sys, lxcore.sys) émulent un kernel Linux en traduisant les Syscalls internes à Linux ;
- Les processus "Pico" qui hébergent une version non modifiée du "user mode" de Linux.



Composants du WSL



Windows 10 Developer Mode

### Comment cela peut-il bien fonctionner dans le détail ?

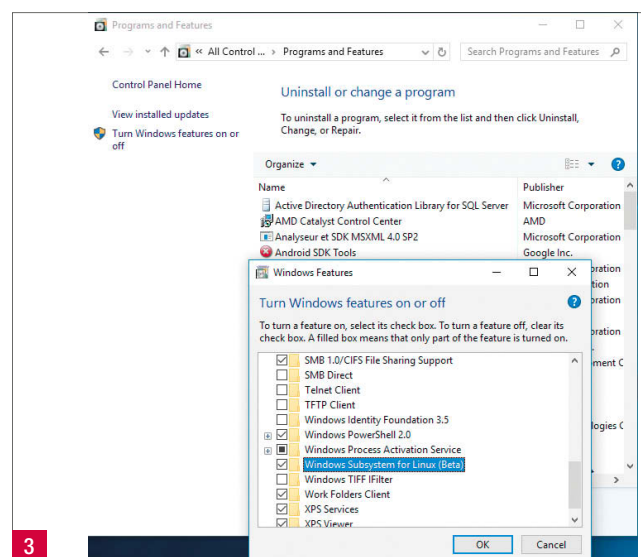
Risquons-nous à traduire en français un fragment de la documentation de Microsoft (Lien1) :

- « C'est une zone qui s'immisce entre les binaires du « user mode » Linux et les composants du Kernel Windows » (figure 1).
- « C'est là où la magie s'opère. En plaçant des Binaires Linux **non modifiés** dans les processus Pico, nous permettons aux appels vers le système Linux d'être redirigés directement vers le Kernel Windows. Les Drivers lxss.sys et lxcore.sys traduisent les appels au système Linux en APIs NT et émulent le Kernel Linux »

### Comment l'activer pas-à-pas ?

Prérequis : Windows 10 Anniversary Update

- Ouvrir les Settings app > Update & Security > For Developers ;
- Activer l'option "Developer Mode" ;
- Valider l'activation, puis attendre les phases de recherche / d'installation des packages « Developer Mode », et l'activation du « Remote Tooling for Desktop » . [2]
- Ouvrir le "Control Panel" > "Programs and Features" > "Programs" ;
- Cliquer à gauche sur "Turn Windows features on or off" ;
- Activer la case à cocher "Windows Subsystem for Linux (Beta)", puis "OK" ; [3]
- Attendre la fin de la recherche des bons composants et l'application des changements, puis confirmer le Reboot de l'ordinateur ;
- Attendre la fin des mises à jour, et suite au Reboot, appuyer sur Start (ou utiliser la touche « Windows »), et saisissez "bash", puis "Enter." Notez qu'il n'y a pas d'icône, nous verrons pourquoi ultérieurement ; [4]



Activation WSL (Beta)

- Au premier chargement, taper 'y' pour mettre à jour Ubuntu et attendre environ 5 minutes... [5]

**⚠ Note :** lors du premier lancement de `bash.exe`, il vous sera demandé de configurer la Locale (par exemple US), d'accepter les « Terms & Conditions » du service, et de créer un compte et mot de passe de l'environnement Bash.

**⚠ Attention danger :** pour les utilisateurs avancés, saisissez dans CMD la commande ci-dessous pour configurer par défaut à « root » sans saisie de mot de passe.

```
lxrun /install /y
```

Le prochain login se fera donc à ce niveau, avec en prime une nouvelle icône et un nom pour la commande Bash, qui s'appellera à présent « Bash on Ubuntu on Windows ».

```
root@localhost:/mnt/c/WINDOWS/system32#
```

[6]

Pour désinstaller, simplement lancer la commande ci-dessous (pour effacer complètement, voir Lien4).

```
lxrun /uninstall
```

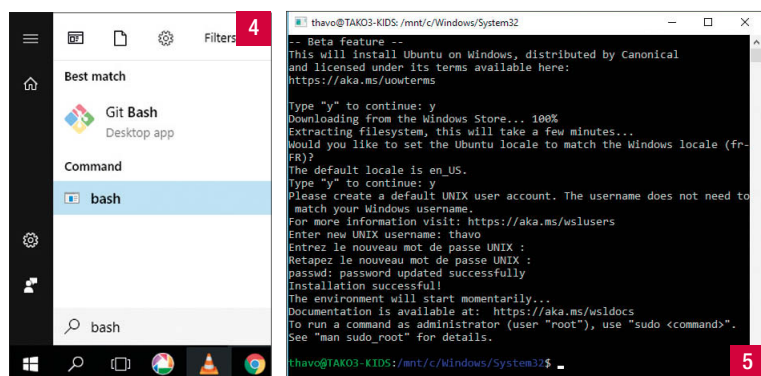
[7]

## En complément d'Ubuntu, WSL supporte également les distributions Fedora, OpenSuse !

Toutes ces distributions sont disponibles sur Windows Store. Il est facile de les installer depuis les pages suivantes et cliquez sur GET.

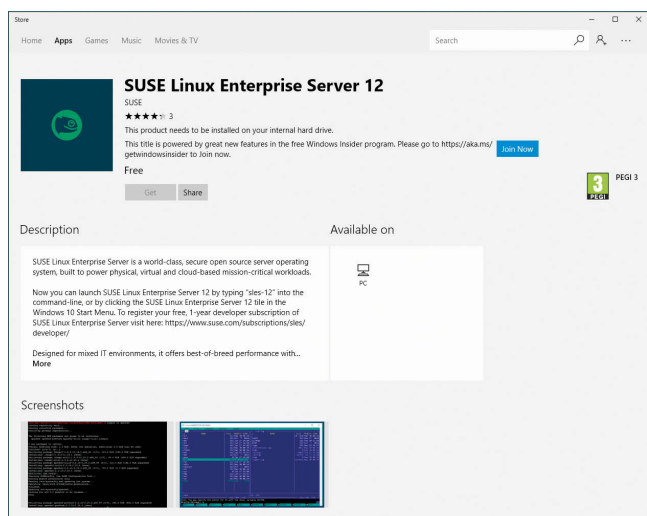
**⚠ Note :** Fedora distro est bientôt disponible (attendu pour "Windows Fall Creators update").

**⚠ Attention danger :** le sous-système Windows pour Linux ne fonctionne que sur le disque principal de votre système (généralement c'est votre disque C:). Assurez-vous que les nouvelles applications soient stockées sur votre lecteur système. [8]



Bash dans le menu Démarrer

Installation de Ubuntu sous Windows



Distros Suse disponibles

## Comment créer son premier « Hello World » ?

C'est très simple et rapide de développer son Bash. Par exemple, pour installer Python, il suffit de taper la commande ci-dessous, accepter la mise à jour et attendre quelques secondes :

```
sudo apt install python
```

Il est également rapide d'exécuter du Python ou des modules pour créer, par exemple, un **SimpleHTTPServer**. Créons un simple fichier HTML et chargeons-le dans le navigateur. Pour cela, débutons par la création d'un répertoire.

```
mkdir webtest
```

Puis créons le fichier dans ce répertoire.

```
echo "" > webtest/index.html
```

Ajoutons-y du contenu HTML dans la splendide interface noire que nous aimons tant de vim.

```
vim webtest/index.html
```

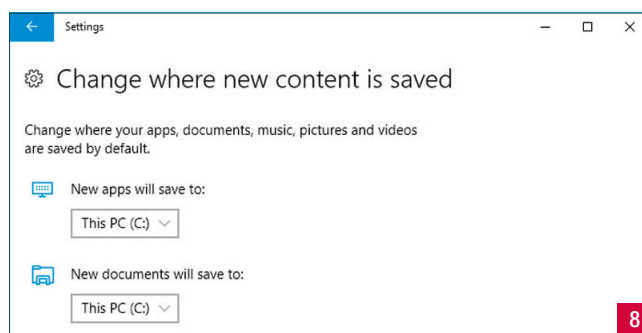
```
<!DOCTYPE html>
<html>
<body>
  <p><h1>Hello from bash webtest!</h1></p>
</body>
</html>
```

Sauvegardons le tout (raccourcis clavier "ESC", puis ":x") et lançons le serveur Python **SimpleHTTPServer** pour afficher le fichier dans le navigateur.

```
cd webtest
python -m SimpleHTTPServer
```

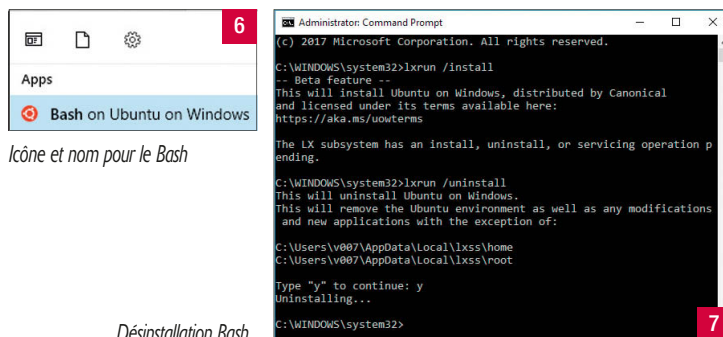
[9]

A présent, ouvrez un navigateur à l'URL <http://localhost:8000/>, et voilà ! [10]



8

Paramétrage système



icône et nom pour le Bash

Désinstallation Bash

7

Nous avons donc vu comment installer WSL, exécuter du Bash et monter un serveur web en quelques lignes de code afin d'afficher un « Hello World ».

## Fonctionnement de Bash sous Windows ; est-ce une VM ? Comment aller plus loin ?

A présent nous disposons d'un Kernel Linux avec une interface intégrée à Windows. Lorsqu'une application souhaite utiliser le Kernel Linux, les appels sont traduits directement sur le système Windows en natif. Puis, toutes les réponses du Kernel sont elles-mêmes traduites en réponse Linux natif. Ainsi l'application pense être exécutée dans un OS Linux. Par conséquent, il n'est pas nécessaire de modifier l'application. Dans Channel 9 (cf lien2) et dans MSDN (cf lien3), force est de constater qu'il y a des « Drawbridge Pico Process » et « Pico Drivers », et cela constitue la base de Windows Subsystem for Linux. De là, vous comprendrez que l'exécution de toute commande Linux va à son tour exécuter des commandes Windows. A présent, revenons à notre « Hello World », et allons plus loin. Si nous souhaitons le diffuser de manière globale, ce n'est pas juste une question d'utiliser **iptables** (ou équivalent) pour ouvrir les ports. En effet, il faut nécessairement configurer les firewalls de Windows. Par conséquent, il est très important de comprendre qu'il n'est pas possible d'utiliser toutes les fonctionnalités d'un Bash natif Linux. Ce n'est pas un environnement virtuel et il n'est pas possible d'avoir le control intégral du système d'exploitation.

Cependant, il est possible d'aller plus loin avec Linux Bash en le combinant à Azure CLI. C'est d'ailleurs probablement pourquoi Microsoft propose cela. Par exemple, mettons à jour notre liste de package source :

```
echo "deb [arch=amd64] https://apt-mo.trafficmanager.net/repos/azure-cli/ wheezy
main" | sudo tee /etc/apt/sources.list.d/azure-cli.list
```

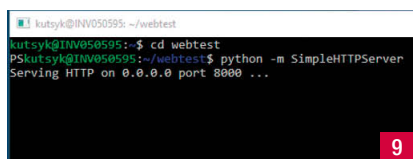
Puis exécutons les commandes pour récupérer des clefs et réaliser des mises à jour :

```
sudo apt-key adv --keyserver packages.microsoft.com --recv-keys 417A0893
sudo apt-get install apt-transport-https
sudo apt-get update && sudo apt-get install azure-cli
```

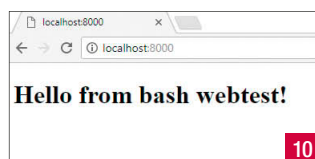
Il devient possible de lancer des commandes **Azure-CLI** en utilisant la commande **az**. A noter que **Azure-CLI 2** a été créé pour que les administrateurs Linux puissent appeler toutes les fonctionnalités qu'ils ont l'habitude d'utiliser. Alors, créons une machine virtuelle Azure en utilisant CLI, et connectons-nous-y ! Un prérequis étant de créer un « Azure Ressource Group » et d'identifier sa localisation pour choisir le Data Center le plus approprié (utiliser la commande **az account list-locations**).

```
az group create -n AzureCliGroup -l northeurope
```

```
{
  "id": "/subscriptions/XXXXXXXX-XXXX-XXXX-XXXXXXXXXXXX/resource
Groups/AzureCliGroup",
  "location": "westus2",
  "managedBy": null,
  "name": "AzureCliGroup",
}
```



Lancement du server Python



Affichage de la page dans un navigateur

```
"properties": {
  "provisioningState": "Succeeded"
},
"tags": null
}
```

A présent, nous créons la machine virtuelle Linux dans ce groupe de ressource.

```
az vm create -n AcliLinuxVM -g AzureCliGroup --image UbuntuLTS
```

Lors de l'exécution de cette commande, l'Azure CLI 2.0 recherche une "key pair SSH" stocké dans votre répertoire `~/.ssh`. S'il n'y en n'a pas, vous pouvez demander via Azure CLI d'en créer une automatiquement en ajoutant le paramètre `--generate-ssh-keys parameter`

```
az vm create -n AcliLinuxVM -g AzureCliGroup --image UbuntuLTS --generate-ssh-keys
```

La commande **az vm** retourne des paramètres de sortie, une fois que la VM a été complètement créée et est opérationnelle. Nous allons y retrouver différentes informations sur la nouvelle VM, comme son IP public (très pratique !).

```
{
  "fqdns": "",
  "id": "/subscriptions/XXXXXXXX-XXXX-XXXX-XXXXXXXXXXXX/resource
Groups/AzureCliGroup/providers/Microsoft.Compute/virtualMachines/AcliLinuxVM",
  "location": "northeurope",
  "macAddress": "00-0D-3A-B1-DA-6F",
  "powerState": "VM running",
  "privateIpAddress": "xx.xx.xx.xx",
  "publicIpAddress": "xx.xx.xx.xx",
  "resourceGroup": "AzureCliGroup",
  "zones": ""
}
```

Avec cette IP publique, utilisons Putty pour nous y connecter.

```
ssh xx.xx.xx.xx
```

C'est tout, nous voici connectés ! Il est à présent possible de configurer les serveurs, les bases de données, etc.

## Conclusion et usages

Il est aisé d'utiliser WSL sans installer d'outils additionnels ; nous retrouvons nos outils standards et productifs qui nous sont chers, comme VIM, Putty, etc. Ceci avec une ouverture vers des fonctionnalités complémentaires du monde du Cloud ! WSL nous a été très utile pour travailler rapidement, notamment sur les technologies de Blockchain pour limiter le nombre d'outils, de nous permettre de monter des processus DevOps-VSTS pour des projets exigeants comme pour les voitures connectées Renault (lien5).

## Références :

- Lien1 : <https://blogs.msdn.microsoft.com/wsl/2016/04/22/windows-subsystem-for-linux-overview/>
- Lien2 : <https://channel9.msdn.com/Blogs/Seth-Juarez/Windows-Subsystem-for-Linux-Architectural-Overview>
- Lien3 : <https://blogs.msdn.microsoft.com/wsl/2016/04/22/windows-subsystem-for-linux-overview/>
- Lien4 : <https://www.howtogeek.com/261188/how-to-uninstall-or-reinstall-windows-10s-ubuntu-bash-shell/>
- Lien5 : Plusieurs présentations du projet ePassport / Carnet d'entretiens connectés lors du Microsoft Experiences <https://experiences17.microsoft.fr/content/livety>

# Co-innovation, co-clustering and Co.



Pascale Bailly  
**Orange Labs Services**



*Les scientifiques utilisent les algorithmes pour accomplir les opérations ad hoc, par exemple l'algo du plus court chemin pour trouver les plus courts chemins, l'algo de clustering pour classifier une collection d'individus, etc. Personnellement, dans mon métier j'utilise certains algorithmes d'une manière inhabituelle, voici pourquoi et comment.*

L'importance de l'innovation dans les entreprises va croissant dans un contexte accéléré d'hypercompétition et de commoditisation. L'innovation, qu'est-ce en partie ? Par exemple trouver des nouveaux produits et des nouveaux services, éventuellement trouver de nouveaux procédés de fabrication, de meilleurs frameworks, de nouveaux déploiements, des bons canaux de distribution, en résumé améliorer, revoir ou pivoter le(s) BM (business modèle) de l'entreprise. Dans ce cadre j'ai travaillé avec, et souvent animé, des ateliers d'innovation collective, internes à l'entreprise ou multi-entreprises ou avec des Ecoles. Si quelques méthodes bien installées existent, même mises en pratique par des spécialistes, il en ressort trop souvent des pistes rebatues et galvaudées (lesdits océans rouges), et une vision non partagée qui mène de facto à des difficultés majeures lors de la mise en œuvre. Par ailleurs, les techniques de ranking participent aussi à la difficulté d'innover. Si elles ont pour ambition de rassembler autour de la meilleure idée (ou des trois premières), elles éliminent les autres idées, perdent de vue les nuances, les croisements, éliminant ainsi bien souvent l'innovation. Les mises en place opérationnelles des résultats en pâtissent en conséquence. En résumé, comment serait-il possible d'adhérer à une stratégie qui ne retient que lesdites meilleures idées tout en éliminant toutes les autres idées émises ?

Prenant le taureau par les cornes, j'ai progressivement développé une méthode de travail et surtout une méthode de dépouillement des idées d'un atelier ou workshop ou barcamp. Dans cet article je décrirai essentiellement le dépouillement.

## CONTRAINTES, GROUPE, CAHIER DES CHARGES

### Les contraintes.

**Contraintes de temps** : le résultat final doit être rendu très vite car il faut présenter les orientations d'innovation au prochain comité directeur. Et, le workshop ne peut souvent durer qu'un jour car les participants ont tous un agenda bien rempli.

**Contraintes d'outils** : il faut utiliser des PC portables de bureau. Il faut utiliser du logiciel connu par nos ingénieurs, nos *business units*, nos thé-sards, nos stagiaires bac+5. J'ai donc opté pour Python, langage n°5 mondial, recommandé par l'Education Nationale, les bibliothèques R, numpy, scilab et autres pourvu qu'elles soient open source et gratuites. Python est plus lent que C mais ce n'est pas critique pour la small data.

### La dynamique de groupe

Afin d'obtenir de la cross-fertilisation, faire sortir chacun de ses préoccupations bloquantes, des idées préformatées par un travail déjà élaboré, je tiens à mélanger des technos, des marketeurs, des financiers, des sociologues, des managers et des non managers. Tous les animateurs de groupes le savent, il faut aussi susciter des attitudes bienveillantes, tolé-

rantes, faire tomber les défis d'égos comme les timidités, et cela dès le premier quart d'heure sinon cet atelier de 15 à 20 personnes sera pagailleux et improductif.

J'y ajoute trois items fondamentaux :

- pas de ranking des participants, pas de ranking des idées émises.
- toutes les idées sont conservées : pas d'élimination des idées fustelles fantaisistes. On ne juge pas les idées en bonne/mauvaise, mais on leur attribue des features (critères) tels que : innovant, coûteux, responsabilité sociale ou environnementale, utilisateurs familiaux ou non, levier de création de valeur, etc.
- anonymisation des idées et des votes. On dépouille les idées suivant les critères, aucune data d'input ne comporte d'identificateur de participant.

### Le cahier des charges

Résumons le cahier des charges d'un workshop aux questions : « peux-tu organiser un workshop pour innover sur tel sujet ? », ou « quel est le business model qui conviendrait pour tel sujet ou programme d'innovation ? ».

Sans prétention à devenir spécialiste du sujet, je me mets au courant des derniers développements business et des nouvelles technologies pour voir quelles adjacences vont être facilitées ; profitant pour cela du large spectre de connaissances pointues des labos d'Orange en marketing stratégique, finance, informatique, algorithmes, traitement du signal, optique, sociologie, futurologie, etc. Je m'enquiers de l'état de la concurrence, je regarde les investissements mondiaux réalisés, je cherche à voir les compétences requises sur ce sujet et les domaines voisins, j'explore l'écosystème industriel et entrepreneurial, trouve quelquefois des sociétés et des startups qui possèdent ces compétences et avec lesquelles on pourrait peut-être demain envisager un partenariat, etc.

Avec ce bagage je prépare des courtes présentations de 5 minutes qui permettront d'ouvrir le workshop puis de faire les transitions entre les exercices de la journée. Je ne décide rien, ce sont les participants à l'atelier qui trouveront des idées et des pistes.

## UN PEU DE THÉORIE

Parlons des algorithmes. A l'issue d'un atelier, nous avons des idées émises, par exemple 170 idées, des critères ou features, par exemple 6 critères. Et nous avons un tableau de notes de votes, les participants ayant attribué un critère majeur à chacune des idées émises ; c'est de la small data.

Nous souhaitons en extraire des informations descriptives suffisamment parlantes représentant les résultats pour les exposer à des commanditaires qui ne sont ni informaticiens, ni statisticiens ni datascientists. Nous recherchons aussi des pistes suffisamment différenciatrices à leur proposer.

Nous procédons en deux temps : clusterisation puis explicitation.

*Avertissement : pour parler d'un concept comme classification, j'emploie souvent le terme anglais qui tend de plus à s'imposer dans toutes les documentations, même francophones ou hispanophones.*

## Quelques familles d'algos de clusterisation

On dit aussi classification, taxonomie. Nous voulons regrouper les 170 idées en une dizaine de clusters, ou grappes ou classes. Ce regroupe-ment doit se faire sur la base de ressemblances entre idées, ressemblances qui se voient sur les notes selon les 6 critères (features) dans notre exemple. Il existe des centaines de méthodes et d'algos de clusterisation. Au cours des années, mes stagiaires et moi en avons testé quelques-unes.

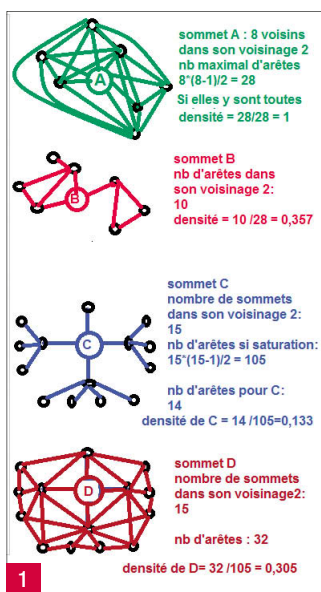
## La famille basée sur les centroïdes

L'exemple le plus connu est l'algorithme K-means ou K-moyennes.

Le logiciel choisit au hasard K centroïdes qui sont des points de l'espace à 6 dimensions de nos 6 features. Puis selon un critère de distance il rattaché chaque idée au centroïde le plus proche, on obtient ainsi K clusters ainsi qu'une matrice d'affinité des idées. Il répète ce processus 1000 fois ce qui n'est pas énorme pour des small datas, avec 1000 tirages aléatoires des centroïdes, à chaque fois on enrichit la matrice d'affinité qui finit par représenter les meilleures affinités au cours des 1000 tirages. Pour finir, l'algo partitionne l'ensemble des idées en K clusters à partir de cette matrice des affinités. On imagine le nombre de variantes possibles : variantes de distances, de loi aléatoire de positionnement des centroïdes, d'exploitation finale de la matrice d'affinités. Sans compter un éventuel souci d'équilibrage de la taille des clusters. En effet une partition des 170 objets en 10 clusters de (2, 2, 2, 2, 2, 2, 2, 2, 2 et 152 objets) serait particulièrement inefficace ; on inventera dès lors diverses variantes d'équilibrage qui ne trahissent pas trop les distances.

## La famille basée sur les zones de densité d'un graphe

Certains problèmes peuvent être modélisés par un graphe qui relie, par une arête de valuation 1, certaines paires d'objets à partitionner. Certaines zones du graphe ont beaucoup d'arêtes entrecroisées, d'autres zones sont au contraire bien moins touffues. L'algo détecte, de manière itérative, la zone la plus dense qui forme un embryon de cluster, puis voit ceux des points voisins des points de l'embryon qui lui sont densément reliés auquel cas ces nouveaux points sont ajoutés au cluster, sinon l'algo baisse un peu l'exigence de densité et cherche un autre embryon de cluster. Et ainsi de suite. Voici une illustration d'une variante de « densité » que l'on peut adopter, il s'agit de la densité aux distances 2 et 1 d'un sommet. Sur ces quatre exemples, souvent à l'encontre d'une impression intuitive, le point rouge est plus dense que le marron qui est plus dense que le bleu. [1]



Densité de plusieurs sommets

## La famille d'analyse spectrale

Nous avons utilisé la fonction de Spectral-Coclustering de la bibliothèque Python du projet scikit-learn destinée au Machine Learning (BSD License, © 2010 - 2016). La clusterisation n'est qu'une infime partie de ce vaste projet.

Essayons d'expliquer cela en termes pas trop ésotériques.

- **pourquoi co-clustering ?** Parce que l'on clusterise simultanément les idées et les features, c'est à dire que la population à partitionner est formée des 170 idées+6 features= 176 individus dans le même espace.
- **pourquoi spectral ?** Mue par une légitime curiosité j'ai interrogé certain(s) matheux(ses), je vous livre ce que j'ai compris ; l'algo travaille sur une matrice carrée, nommée « laplacienne » de notre population de 176 individus, puis calcule la « laplacienne réduite » de cette laplacienne. Il s'agit d'algèbre linéaire, le spectre est l'ensemble des valeurs propres de la matrice.

L'algo traite même un point délicat de mes exigences, l'équilibrage de la taille des clusters. Comment ? En donnant un « poids » 1 à chaque idée et critère, et en minimisant une fonction qui combine les poids et la laplacienne réduite : cette équation a pour résultat le clustering cherché.

## La famille de la diagonale de blocs

Dans la pratique les noms des lignes sont les idées suivies des critères, et idem pour les noms de colonnes. A chaque étape, l'algo permute 2 lignes et 2 colonnes qui se croisent sur la diagonale principale. Cette permutation a pour but de rapprocher quelques grandes valeurs de la diagonale principale tout en évitant d'en éloigner d'autres grandes.

Je tente d'illustrer cette méthode ici visuellement, elle consiste à prendre la matrice carrée symétrique qui réunit les 170 idées et les 6 critères, comme pour le coclustering spectral.

Je délimite le long de la diagonale principale des blocs carrés (ici 3 blocs sur une matrice 14x14 pour simplifier). Pour simplifier la visualisation les cellules de la diagonale contiennent un antislash.

**Le but du jeu est :** mettre le plus possible de grosses valeurs dans les blocs carrés et donc le plus possible de petites valeurs hors des blocs.

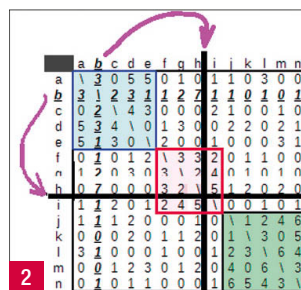
Sur la matrice ci-jointe les sommes des valeurs sont :

in-blocs = 158, hors-blocs = 106, total = 264, rapport in/hors = 1,4906 [2]

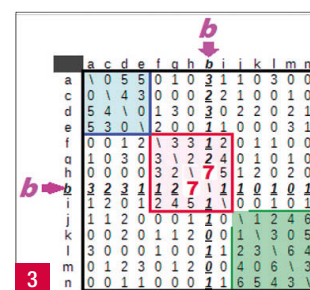
Admettons que je remarque une grosse valeur 7 en (b,h) et (h,b), je souhaite tenter une manœuvre, extraire la colonne b et l'insérer entre les colonnes h et i, voir l'épais trait noir ; idem pour les lignes b, h-i afin de conserver la symétrie matricielle. Les noms des lignes et colonnes correspondront alors à la permutation a c d e f g h b i j k l m n.

Le bloc bleu deviendra de taille 4x4, le bloc rouge de taille 5x5, les autres blocs seront invariants, voyons les valeurs obtenues : [3]

in-blocs = 162, hors-blocs = 102, total = 264 (inchangé), rapport in/hors = 1,588



Avant permutation



Après permutation

Incontestablement il y a un progrès, donc on abandonne abcdefghijklmn au profit de acdefgh**h**ijklmn.

Si on réitère des essais de déplacement (ligne et colonne) en conservant les permutations les plus avantageuses, on finit par obtenir la meilleure diagonale de blocs, donc les meilleurs clusters. Bien entendu l'essai systématique des factorielle(n) permutations serait trop long, par exemple plus de 10 puissance 300; l'algo doit utiliser certaines méthodes de sélection des meilleurs candidats à la permutation ce qui donnera lieu à des variantes dépendant de l'imagination des mathématiciens.

## Algos d'explicitation

Maintenant il faut exploiter les clusters. Pour mémoire, nous avons choisi le nombre de 10 clusters.

## Modules de dendrogramme

Ce mot bizarre désigne un arbre ou un cladogramme tel celui que les zoologues dessinent depuis Buffon pour classer les espèces animales. La population étudiée est celle des 10 clusters précédents assortis des 6 features. Un algo les fusionne par récurrence jusqu'à l'obtention d'une seule famille, celle de tous les clusters.

On peut faire plusieurs sortes de dessins d'un dendrogramme. Nous avons opté pour la métaphore du bassin fluvial (cf. Fig 4)!

Par exemple, après plusieurs propositions de doctorants et stagiaires, j'ai retenu cette formule de meilleure fusion :

```
entrop_classique = calc_entropie_list_avec_repets(list_modalites_repets)
logn = math.log2(N) # le pire desordre pour N items
#####
# ici formule pragmatique de compromis
# entre (taille du nuage) et (richesse d'information de Shannon)
gain_coherence = -10000*(logn-entrop_classique)/(N*math.log2(N))
# on choisira plus tard le regroupement de clusters qui a le plus grand
# "gain_coherence" #
#####
```

## Modules de percentiles

Mettons-nous à la place d'un destinataire du rapport final. Si nous lui présentons des tableaux ou graphiques de notes entre 0 et 20, il aura du mal à imaginer les grandes tendances en lisant :

« le cluster 3 recueille 17½ en écologie et 3¾ en succès-client tandis que le cluster 8 vaut 5¼ en écologie mais 12 en innovation, mais le cluster 4 etc. »

C'est la raison pour laquelle, à partir d'un certain moment on adopte une échelle textuelle des percentiles. Ce mot anglais correspond à nos déciles ou quartiles ou quintiles, bref nos n-iles de statistique. J'ai adopté par exemple les quartiles baptisés « faible », « moyen », « fort » et « très fort » pour chaque feature numérique ; c'est lisible par un humain.

## Module de matrices 3D ou 2D et demi

Voici une question éternelle, les idées et clusters étant exprimés par des échelles de 4 quartiles pour 6 features, comment représenter les propriétés sur un papier en 2D ?

Un petit algo me propose une feature pour l'axe x, une pour l'axe y et une autre pour l'axe z qui sera représentée en tailles et couleurs des ronds de numéro des clusters (cf. Fig. 5)

# PLACE À LA PRATIQUE

## Mon métier

La partie d'agilité et de développement par des méthodes de business modèles du métier de stratège n'existait pas, je l'ai inventée depuis 2013. D'ailleurs la plupart des lecteurs exerceront eux aussi dans cinq ans un métier qui n'existe pas aujourd'hui.

Je suis développeuse. Pas de logiciels mais de méthodes de BM (business models).

Pour faire quoi ? Pour offrir à diverses divisions de l'entreprise des pistes de stratégie, anticipation et prospective visant à créer de la valeur. Le tout en cycles de devops, c'est à dire que je produis rapidement des pistes de business à explorer, je les fournis aux intéressés et je vois immédiatement leur réaction en feed-back, si c'est mauvais je dois améliorer mes méthodes, si c'est bon je dois peaufiner des perfectionnements. La démarche est complètement pragmatique et heuristique. Dans ce cadre les ateliers et les algos ne sont qu'une partie de mon travail.

Concernant le développement logiciel, ma compétence se limite à des spécifications fonctionnelles, par exemple :

- « svp, peux-tu rajouter l'équilibrage des tailles de clusters à ton algo ? »
- « svp, peux-tu trouver un module python qui clusterise de manière un peu moins banale que K-means ? »
- « svp, peux-tu rajouter le calcul de la corrélation entre 2 features ? »

## Cas d'usage concret

Prenons la mise en place d'un workshop participatif dédié à ouvrir des pistes possibles. Rappelons qu'il ne s'agit que d'un exercice et que d'autres s'y ajoutent, les différentes *business units* et responsables de programmes recherchent de la valeur ajoutée par le biais de ces workshops et tranchent ensuite.

## Sujet traité

### Exemple d'une demande.

Imaginer un « Meccano » de logiciels et de matériels pour les petites classes d'une école.

Interdiction de réinventer le bus scolaire connecté, la TV scolaire, la Box communautaire, les langages Scratch ou Logo, etc.

**AVANT** : pour obtenir un vaste panorama des innovations, j'ai recherché notamment les acquisitions et investissements mondiaux, ainsi que les nouvelles technologies pour écoles primaires et maternelles ; elles foisonnent sur Kickstarter, MIT, Stanford, Wikipedia mais ne sont pas spécialement pensées pour Orange. Puis j'ai précisé ce panorama avec divers chercheurs, ingénieurs, *marketeurs*, financiers... Et ensuite j'ai monté quelques séries de slides de 5 à 10 minutes destinées à amorcer les 3 ou 4 exercices de la journée.

**PENDANT** : l'atelier tous participants s'est bien déroulé, avec des spécialistes et des généralistes de nombreux domaines, marketing et ingénierie, mais aussi des spécialistes de sciences économiques et sociales, des stagiaires, des néophytes... L'ambiance a été bonne, nous avons récolté 170 idées soumises aux votes selon 6 critères ou features.

**APRES** (c'est surtout ceci qui montre mes utilisations d'algorithmes) : comme expliqué précédemment, nous avons fait tourner un algo de clustering, j'ai imposé 10 clusters pour une moyenne de 16 idées. Puis un algo de dendrogrammation qui a donné les regroupements de clusters

selon cet arbre binaire que j'ai dessiné selon la métaphore de bassin fluvial. Au passage signalons que l'arbre binaire est ordonné, car j'ai demandé que soient positionnées à l'ouest les confluences les meilleures selon notre critère de théorie de l'information du « meilleur gain en cohérence », (la fusion 0+9 est donc meilleure que la fusion 4+3 elle-même meilleure que 1+8+2+3+4) [4]

## DETECTION DE PISTES.

C'est un travail effectué à plusieurs, non automatisé. Nous lisons et re-lisons les contenus des idées des 10 clusters, pour chaque cluster ses votes moyens selon les 6 critères, votes compactés en quartiles (faible, moyen, fort, très fort).

J'ai essayé de voir, du point de vue ouest/est, en plus du choix de meilleur gain en cohérence, s'il y avait des orientations préférentielles (+ou- innovant à l'ouest ?, +ou- écologique, +ou- urgent? Etc. Le tout selon les votes des participants et non mon sentiment personnel). Après plusieurs relectures, nous avons trouvé effectivement de vraies orientations qui seraient fastidieuses à énumérer. Ces orientations m'ont poussée à envisager quelques « tendances fortes » détaillées ici et restituées aux demandeurs. Deux exemples (A et B) :

## A) Regroupement des clusters 3,4,2

Ils sont composés de 56 idées dont beaucoup représentent (selon les participants) les bases urgentes à mettre en place pour que l'entreprise soit efficace dans le secteur. Les quartiles (25 % percentiles) des critères de ce regroupement sont :

Services tiers =très forts, efficaces pour ce marché =fort, Succès client =fort, Utiles à la fois pour les marchés écoles et d'autres =fort , Urgent fort, RSE =fort.

Voici comment nous avons pu synthétiser les qualités dominantes dans ces 56 idées :

« Les 'douleurs' d'utilisation, pour les enseignants, assistants pédagogiques ou directeurs, liées à certains des équipements (lors de l'installation ou de la configuration, ou dues à des batteries qui se déchargent trop vite, ou à des pannes d'un composant) seront 'RENVERSEES' ou 'ELIMINEES'\*1, et il est souhaitable qu'une panne soit identifiée avant que le client ne soit obligé de faire quoi que ce soit. »

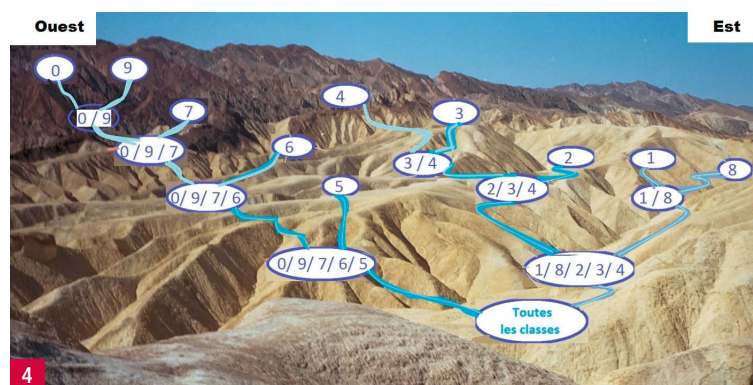
On y trouve aussi une orientation RSE : elle est reliée aux usages de qualité atmosphérique en relation avec le Ministère de la Santé et Météo France, une idée étant d'ajuster la planification des sorties scolaires selon les prévisions de pollution atmosphérique.

Quelques idées qui incluaient des partenariats potentiels innovants y sont ajoutées : par exemple une antenne bluetooth pour récupérer des travaux d'élèves sur ardoises magiques, combinée avec une lampe de plafond de salle de classe pourrait donner lieu à un double partenariat de notre entreprise avec une marque de luminaires et le Rectorat.

Remarques : il est clair que je n'aurais jamais inventé ces 56 idées toute seule, ni aucun des participants isolé dans son bureau. En revanche c'est moi qui ai proposé un exercice sur le thème 'RENVERSER', 'ELIMINER', 'DOUBLES PARTENARIATS'\*2. On comprend dès lors sur ces exemples, que ceci ne peut fonctionner que si l'ambiance de l'atelier n'est pas cris-

(1) Inspiré d'Osborn qui en 1979 a proposé une liste de questions destinées à faire jaillir des idées. Ces questions ont été ensuite regroupées sous l'acronyme S.C.A.M.P.E.R. : S pour substituer, C pour changer, A pour adapter, M pour modifier, E pour éliminer, R pour renverser.

(2) Suivant le levier de deux partenaires représentés en 'losange' : cf. page 46 [http://willstratemy.com/download/MinesTelecom/2014-Will\\_IMT-BusinessModelsNumeriques.pdf](http://willstratemy.com/download/MinesTelecom/2014-Will_IMT-BusinessModelsNumeriques.pdf)



Dendrogramme des fusions des clusters sous la forme d'un bassin fluvial.

pée, et que les participants vont au-delà des limites bloquantes telles que la dette technologique (ancienne version logicielle, remise à jour, anciens routeurs, etc.). Il est clair aussi qu'il ne suffit pas d'émettre de sympathiques idées de *fonctionnement sans douleur*, ce ne sera sans doute pas facile d'y parvenir avec un futur « meccano » software-hardware.

## B) Regroupement des clusters 1 et 8

Ils regroupent 44 idées jugées en moyenne beaucoup plus innovantes que le regroupement 3+2+4.

Les quartiles (25 % percentiles) des critères de ce regroupement sont : Innovant =très forts, services tiers efficaces pour ce marché =fort, succès client =fort, utiles à la fois pour les marchés école et d'autres =fort , Urgent fort, RSE =moyen.

Ces idées proposent souvent d'effectuer une combinaison de plusieurs fonctions dans un équipement, ou une hybridation avec un appareil pré-existant qui n'est pas du domaine d'Orange, par exemple une ardoise magique qui projette des vidéos. Avec parfois une sous-traitance de fonctionnalité à une entreprise tierce.

Un autre trait dominant de 1+8 : l'absence de connotation ou visibilité technique du hardware, à savoir que les composants du meccano soient noyés dans le décor ou les utilitaires des locaux, fonctionnent sans qu'on les voie, ne présentent aucune abondance de rallonges USB, Ethernet, antennes, fils de chargeurs.

Enfin une feature originale apparaît, voici l'idée : l'école était un lieu clos mais les gens ressentent un besoin d'ouverture ; Orange qui possède l'expertise de la gestion des réseaux et de la relation avec divers clients, aurait en charge de déclencher des propositions de contact entre l'école et la cité (city) et réciproquement. Les idées ont foisonné dans ces 2 clusters, par exemple : liaison avec les pharmacies, hôpitaux, pédiatres, alertes allergies, tempêtes, pics de pollution ; ou théâtres, zoos, jardins partagés, musées du voisinage ; ou supérettes pour les fournitures de rentrée, et bien sûr l'Hôtel de ville.

Le métier d'opérateur de réseau s'enrichirait d'un rôle d'« ouvrier sur la ville » dans un univers de plus en plus complexe.

Rien de vraiment nouveau, ce concept a déjà été signalé il y a plusieurs années par certains sociologues, économistes et prospectivistes, l'exemple classique étant la « sweet railway station ». Autrefois les gares étaient des excroissances dans la ville, maintenant elles cumulent des galeries commerciales, lieux culturels, passages transparents, carrefours multimodaux piétons+vélus+voitures+bus+tramways, etc. L'opérateur de

transports est devenu aussi un « ouvrier vers ... », comme à St Lazare ou dans les futures gares du Super Métro parisien.

Mais je dois reconnaître qu'ici j'ai été bluffée par 4 réussites :

- a) qu'un lot d'idées liées à ce virage ait émergé spontanément à partir d'exercices de maïeutique d'idées dans un workshop.
- b) qu'à partir de votes sur 6 critères, un algo de clusterisation ait rangé ces idées dans seulement 2 classes – les nuances d'un quartile entre 1 et 8 portant juste sur la RSE et l'Urgence.
- c) que, dans une fusion supplémentaire, muni des noms des 10 clusters, de leurs quartiles et de leurs taille N, l'algo de dendrogrammation ait groupé les clusters 1 et 8, dont la lecture humaine de leurs 44 idées fait apparaître le phénomène de « ouvrier sur la ville ».
- d) vérification inverse : quelques idées opposées (l'école coffre-fort électronique de ses dessins et travaux, l'école cocon des écoliers et maîtres) sont clusterisées hors 1 et 8 (idées 29 et 157 attribuées aux clusters 2 ou 6).

### Au bas de la montagne,

En regardant la fusion des clusters représentée par l'image du bassin fluvial, vous voyez qu'il ne reste plus que 2 gros fleuves, 1+8+2+3+4, confluence des deux fusions commentées ci-dessus, et son complémentaire. Notre examen humain de leurs différences nous a montré ceci en résumé :

0+9+7+6+5 à l'ouest possède une forte Responsabilité Sociale et Environnementale, celle de 1+8+2+3+4 est moyenne. Les différences sur les 6 critères sont donc très légères ce qui se conçoit bien dans la mesure où l'algo de dendrogrammation fusionne deux fleuves les plus proches possible, ce à tous les étages.

### Un résultat en matrice 2D et demi

L'examen du dendrogramme n'est pas le seul outil, il est global en ce sens où celui-ci prend en compte les 6 critères. La visualisation en 2D1/2 se contente de 3 critères, en voici un exemple : [5]

Les 3 dimensions ici sélectionnées sont le succès, l'urgence et l'utilité à la fois pour les marchés écoles et d'autres (notée Pro & GP), on constate qu'il y a une case de 3 critères forts, celle qui contient les clusters 6 et 1. Premier enseignement : 6 et 1 sont complètement étrangers sur le bassin fluvial, j'en déduis que les 3 dimensions (succès, urgence, Pro & GP) suffisent à expliquer leurs points communs, les 3 autres dimensions étant marginales pour ces deux clusters.

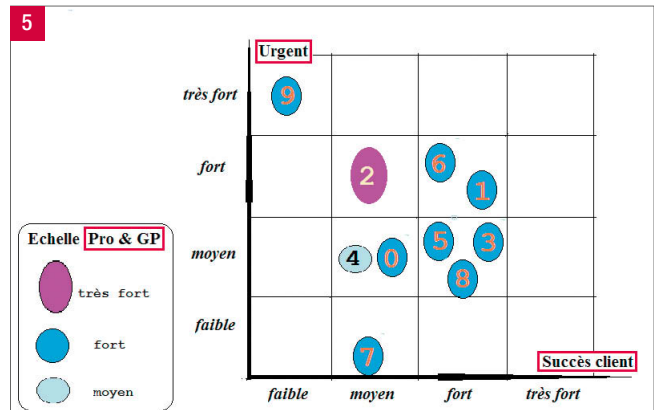
Autres enseignements :

- Le cluster d'idées (le 7) à mettre en place avec une faible urgence avec un effet moyen quant au probable succès client, n'emporte pas l'adhésion des participants et est en tout cas considéré le moins urgent à planifier.
- Par contre, le cluster 9 est considéré d'une urgence extrême à mettre en place, mais sans détenir la clé du succès client. Il s'agit ici d'investissements nécessaires mais estimés non monétisables.

Et une autre remarque : la visualisation de résultats d'un algo me sert pour deux aspects.

- visualisation de recherche, comme vous l'avez lu, la vision d'un dessin suscite souvent des questions. Non seulement de ma part mais aussi de la part des commanditaires.
- visualisation de communication. Communication nécessaire lors des réunions de restitution des résultats, lors de la rédaction du rapport final, lors des notes de synthèse pour les demandeurs.

Nous avons encore examiné d'autres matrices 2D1/2 à pistes intéressantes dont le récit dépasserait le cadre de cet article.



Exemple de matrice en 2D et demi

### La pratique en résumé

A l'occasion de divers ateliers sur des sujets chaque fois différents, voici ce que nous avons constaté, mais il s'agit juste d'un petit nombre d'utilisations.

**Centroïdes : K-means** ne nous a pas fourni de pistes très géniales, mais il faut reconnaître que beaucoup de choses dépendent du nombre K que l'humain impose a priori. Une variante des centroïdes fondée sur une sorte de modèle d'attraction newtonienne entre idées dans l'espace à 6 dimensions, plutôt que sur des distances ; cette méthode a donné des bonnes idées, mais son gros danger est qu'au cours des itérations, les grosses planètes avalent systématiquement les petits astéroïdes, d'où une difficile programmation des fonctions d'équilibrage des tailles.

**Zones de densité d'un graphe.** La nature de nos données ne nous a pas permis de trouver une bonne utilisation jusqu'à présent. Essentiellement parce que la donnée issue d'un atelier est un tableau de notes 170 x 6, ce qui donne un graphe beaucoup trop régulier ou répétitif. La porte n'est pas fermée, si par bonheur nous disposons de 2 jours pour un atelier, le jour supplémentaire permettant aux participants de raccrocher des « signaux faibles » à certaines idées, ou bien des « méta-features » ou encore d'inventer de nouvelles features sur lesquelles voter à nouveau.

**Analyse spectrale.** Pour l'instant cette méthode est celle qui nous a le mieux réussi. Notre tactique avait été d'imposer le nombre K de clusters et d'exiger un « bon » équilibrage des tailles de clusters, ce qui est très souple puisque nous avons obtenu 10 clusters dont les tailles allaient de 4 à 26 idées. Lorsque je qualifie ces résultats de satisfaisants, ce n'est pas d'après une hypothétique métrique de satisfaction mais parce que les 10 clusters ont donné lieu à des explications particulièrement claires issues du deuxième algorithme.

A ce stade l'output que je délivre aux intéressés est la collection des 10 clusters avec le texte de leurs idées. Les intéressés peuvent déjà dire : « le cluster 1 est plutôt orienté sur l'économie d'énergie », « les idées du cluster 6 semblent un peu éthiques », « le cluster 7 ouvre une nouvelle création de valeur ».

Mais je ressens tout de même la nécessité de leur fournir aussi plus de matière à réflexion basée sur les business models.

**Dendrogrammes.** Je suis agréablement surprise par l'utilité cognitive de

cette représentation synthétique : les gens comprennent bien la logique de confluences progressives, surtout lorsque les features convergent bien elles aussi. Pour simplifier je me contente d'arbres binaires. Le seul paramètre de réglage sur lequel on doit travailler à chaque workshop est la formule de combinaison entre l'entropie de Shannon et l'équilibre des tailles. Les dataviz sont là, sinon les gens ne comprendraient rien.

D'où la métaphore du bassin fluvial (© Orange\_P. Bailly, 2017), et la représentation par matrices 3D notamment.

## CONCLUSION

Une chose essentielle. Dans des contextes d'hypercompétition et de commoditisation accélérés, de progrès accélérés vers l'intelligence artificielle générale contrastant avec l'intelligence artificielle qui traitait jusqu'ici un seul domaine à la fois, les métiers et activités, l'innovation collective que vous exercerez dans cinq ans n'existent pas aujourd'hui.

Je tiens à remercier mes ex-stagiaires ainsi que Buffon, Laplace et Shannon sans qui cet article n'aurait pas été écrit.

## Références pour approfondir.

Projet scikit= [scikit-learn.org/stable/index.html](https://scikit-learn.org/stable/index.html)

Centroides= [en.wikipedia.org/wiki/Centroidal\\_Voronoi\\_tessellation](https://en.wikipedia.org/wiki/Centroidal_Voronoi_tessellation)

K-means= [en.wikipedia.org/wiki/K-means\\_clustering](https://en.wikipedia.org/wiki/K-means_clustering)

Small data= [smallgroup.com](https://smallgroup.com)

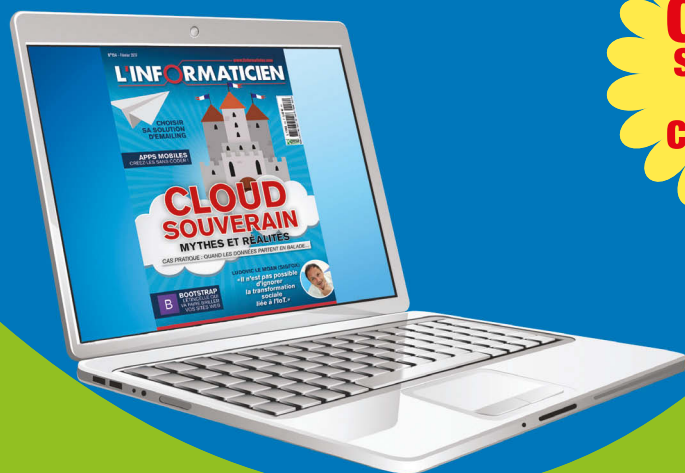
Pascale Bailly et José Parra Diaz, mars 2016, Cahier de prospective sur l'efficacité collective, think tank Fondation Télécom et Institut Mines-Télécom (pages 91 à 103)

<https://www.fondation-mines-telecom.org/wp-content/uploads/2016/01/Cahier-Efficacite-collective-web-2016-03-17.pdf>

Dynamique de groupe= [fr.wikipedia.org/wiki/Dynamique\\_de\\_groupe](https://fr.wikipedia.org/wiki/Dynamique_de_groupe)

Education nationale Python= Informatique, classes préparatoires scientifiques, Eric Le Nagard, éd. PEARSON, 2013, ISBN 978-2-7440-7675-6

# L'INFORMATICIEN + PROGRAMMEZ ! versions numériques



2 magazines mensuels  
22 parutions / an + accès aux archives PDF

PRIX NORMAL POUR UN AN : 69 €  
POUR VOUS : 49 € SEULEMENT\*

Souscription sur [www.programmez.com](http://www.programmez.com)

\* Prix TTC incluant 1,01€ de TVA (à 2,10%).

# Images **Docker** multi-plateformes



Adrien Blind,  
Technical Thought Leader  
@Société Générale  
Docker Captain



Ludovic Piot,  
Architecte DevOps et Cloud  
@theGarageBandOffIT

**M**ars 2013, **Docker** s'ouvre à la communauté *open source* et suscite l'intérêt de nombreux acteurs, et pour cause ! **Docker** porte en lui les gènes d'un formidable outil de *packaging* et de déploiement. *PORtable* est la première de ses propriétés observables, résumées à travers l'acronyme PO-DI-LI-SO(1).

La faute au **Dockerfile**, décrivant dans une syntaxe proche du *shellscript* la construction d'une image de *container*. La faute aussi à ses commandes ultra simples de construction, de publication et de déploiement(2) d'une image à travers les différents environnements **Docker** depuis le poste du développeur, jusqu'aux clusters de production.

Toutefois, la limitation de cette portabilité aux seules plateformes Linux tournant sur processeurs **x86** génère de la frustration chez tous les *DevOps* qui rêvaient depuis si longtemps au "*build once, run everywhere*".

Pendant des mois, la pédagogie autour des conteneurs consiste à les comparer (avec circonspection) à des VMs. Ils en reprennent certains principes sous forme allégée (et moins aboutie(3)) : isolation des processus en cours, portabilité de l'artefact d'une plateforme à une autre, comme un élément atomique.

Il n'en demeure pas moins une désillusion certaine : un conteneur ne sait faire tourner que du processus Linux **x64/amd64**, sur un serveur Linux-**x86** portant un runtime Docker. Là où un serveur **KVM** va savoir faire tourner une **VM Ms Windows**, et un serveur **x86 QEmu** saura simuler un processeur **ARM**.

## 2014 : la communauté Raspberry PI envahit le Hub Docker

Dès novembre 2013 et la version 0.6.6, des *hacks* ouvrent la possibilité d'utiliser **Docker** sur processeurs **ARM**, en particulier sur **Raspberry PI**(4). La société **Resin.io**, spécialisée dans l'IoT, devient l'acteur de référence pour la fourniture d'*images de base* pour plateforme Linux-**ARM** et plateformes embarquées.

D'autres suivront progressivement, dans le sillage de Resin.io, à l'instar de **Microsoft**, qui montrera dès 2015(5) les conteneurs **Windows**, bientôt

*Commençons par le début : Docker est né chez DotCloud, pour réaliser les déploiements applicatifs sur leur offre de cloud PaaS. Celle-ci reposait alors sur le système d'exploitation Linux, et supportait tout logiquement les architectures matérielles x86/amd64 (de simples machines virtuelles hébergées sur le cloud IaaS d'Amazon Web Services).*

suivis des conteneurs **HyperV**.

Le **Hub Docker** commence donc à porter des images dédiées à des plateformes hétérogènes, sans qu'il soit vraiment conçu pour cela :

- Les fichiers *manifest* qui décrivent les images publiées dans le Hub ne portent aucune notion de plateforme, d'architecture de processeur ou d'OS ;
- Seuls les noms des images indiquent explicitement le type de plateforme à laquelle l'image est destinée ;
- Des organisations spécifiques comme [armhf](#) regroupent les images destinées aux plateformes autres que **x86** ;
- Aucun mécanisme ne contrôle, lors du **docker run**, qu'une image déployée sur un runtime **Docker** est bien destinée à la plateforme sur laquelle tourne le runtime en question. Il en résulte des erreurs parfois difficiles à interpréter.

## Janvier 2016 : le Hub Docker gère le multi-arch

En janvier 2016, une nouvelle version du **Registry Docker** apparaît, estampillée v2.3(6). Outre les *manifests* décrivant traditionnellement les images, celle-ci supporte désormais en plus un nouveau type d'objet, optionnel, les *manifest lists*(7).

Celles-ci permettent d'exposer virtuellement une image dans le **Registry**, en y attachant concrètement une ou plusieurs images réelles sous-jacentes. La grande nouveauté réside dans le fait qu'à chacun de ces pointeurs sont associées des précisions relatives à la plateforme pour laquelle cette image réelle est destinée : architecture processeur, système d'exploitation (et éventuellement sa version), fonctionnalités, etc.

```
{
  "schemaVersion": 2,
  "mediaType": "application/vnd.docker.distribution.manifest.list.v2+json",
  "manifests": [
    {
      "mediaType": "application/vnd.docker.image.manifest.v2+json",
```

(1) *PORtable* - Disposable - Live - Social : cf. <https://youtu.be/oDmc9GrVE2k>

(2) Respectivement **docker build**, **docker push**, **docker pull** et **docker run**.

(3) cf. les talks de Quentin Adam sur les différences entre conteneurs et VMs [https://youtu.be/uS\\_aMgqcMJw](https://youtu.be/uS_aMgqcMJw)

(4) cf. [ce post de resin.io](#)

(5) <https://blog.docker.com/2015/04/microsoft-demonstrates-docker-in-build-2015-keynote-address/>

(6) <https://integratedcode.us/2016/04/22/a-step-towards-multi-platform-docker-images/>

(7) <https://docs.docker.com/registry/spec/manifest-v2-2/>

```

"size": 7143,
"digest": "sha256:e692418e4cbaf90ca69d05a66403747baa33ee08806650b51fab815ad7fc331f",
"platform": {
  "architecture": "ppc64le",
  "os": "linux",
}
},
{
  "mediaType": "application/vnd.docker.image.manifest.v2+json",
  "size": 7682,
  "digest": "sha256:5b0bcabd1ed22e9fb1310cf6c2dec7cdef19f0ad69efa1f392e94a433501270",
  "platform": {
    "architecture": "amd64",
    "os": "linux",
    "features": [
      "sse4"
    ]
  }
}
}
}
}

```

Ainsi, le **Registry** expose désormais à la fois de véritables images, ces pseudo-images référençant quant à elles plusieurs images concrètes. Pour nous en rendre compte, vous pouvez par exemple vous rendre sur l'entrée de l'image Python(8) dans le **Hub Docker**. Vous pouvez par exemple y voir que cette image est disponible pour plusieurs architectures :

• Supported architectures: ([more info](#))  
amd64, arm32v5, arm32v7, arm64v8, i386, ppc64le, s390x, windows-amd64

Pour aller plus loin, nous pouvons utiliser un petit outil appelé **manifest-tool**(9). Celui-ci a été écrit par le *Docker Captain* Phil Estes, car il n'existe pas aujourd'hui de fonctionnalités natives dans la **CLI** de **Docker** pour ce faire. Mais nous allons y revenir dans un instant.

```
./manifest-tool --username myuser --password mypassword inspect python:latest
```

(myuser et mypassword servant à vous identifier sur le **Hub Docker**).

On voit effectivement apparaître que cette référence "pointe" vers 8 images différentes, construites pour autant de plateformes :

```

Name: python:latest (Type: application/vnd.docker.distribution.manifest.list.v2+json)
Digest: sha256:eb20fd0c13d2c57fb602572f2705f7f1e87f606045175c108a7da1af967313e
* Contains 8 manifest references:
1 Mfst Type: application/vnd.docker.distribution.manifest.v2+json
1 Digest: sha256:e36969aaef672c06ed8919c77cb875257700808d453badfd30558aa0c5450b98
1 Mfst Length: 2007
1 Platform:
1   - OS: linux
1   - OS Vers:
1   - OS Feat: []
1   - Arch: amd64

```

```

1   - Variant:
1   - Feature:
1 # Layers: 8
   layer 1: digest = sha256:85b1f47fba49da65256f07c8790542a3880e9216f9c491965040f35ce2c6ca7a
   layer 2: digest = sha256:5409e9a7fa9eab9287c6992a75dc9b70811a01e96a24872a02ad07bad557b20a
   layer 3: digest = sha256:661393707836d5196874d18a5d89808079f01cb4376800584bfc1f7b33dd0852
   layer 4: digest = sha256:1bb98c08d57eb9aa8b8a810bcee569cbeab2e6828840f36deb1ef485b3ae16b2
   layer 5: digest = sha256:c842a08369e2b03637ee98013843a8c973e8736d9fcc8999e73acd8d3dee1ff1
   layer 6: digest = sha256:85dae9ff820c01e11b3ba11a457142205080c8f4fb8713f18dfba53971f11437
   layer 7: digest = sha256:86149ff515a26477752748914449cf0afaab6a0beec7f40f113ad70fbd3c97b5
   layer 8: digest = sha256:4aaec9c0a093ee78238fce05924a5ace95687368cc3bdd82557fd2afc387637d

2 Mfst Type: application/vnd.docker.distribution.manifest.v2+json
2 Digest: sha256:280cda4992300bd55a4e581e4db1835c4252ef457730338347820039829ccf3b
2 Mfst Length: 2007
2 Platform:
2   - OS: linux
2   - OS Vers:
2   - OS Feat: []
2   - Arch: arm
2   - Variant: v5
2   - Feature:
2 # Layers: 8
   layer 1: digest = sha256:29dee24d6376416a80f3fdb145082e8dd352694bfdfc639e49e26ddbfb8d8cb52
   layer 2: digest = sha256:dffbd68139d082d5462baf2dcc79e9164c03c8aa02e40c979170c408d744c80b
   layer 3: digest = sha256:d25a8d00f5623c0cf6540a91059805ce464adbb2a5a50059f743796346302529
   layer 4: digest = sha256:41f70cf8e24fc506bff66944b0995d7b0a8d41bd4c0fb4563121c98a32944b1
   layer 5: digest = sha256:73a41710f4de861d18aa12d636e81c1dc6d64a31cfc4ce6fdbf1ffb93479f3fa
   layer 6: digest = sha256:0375ec976e550dde26b06569822d06a20c6c0c1543051f566fd4c5d0de8ecfe5
   layer 7: digest = sha256:5b59f22f6e6a831f71ca47e8caba8eaf3cd0d1220ed314922a74c88351a15da9
   layer 8: digest = sha256:86d944d535076aea69eaf6614b33654c43d2d88284489af92cc89730f5b93c4a

```

Il est à noter que nous sommes dans cet exemple dans un cas particulier : celui des images officielles. En effet, celles-ci disposent désormais toutes(10)(11) de *manifests lists*, et ce depuis septembre 2017.

(8) [https://hub.docker.com/\\_/python/](https://hub.docker.com/_/python/)

(9) <https://github.com/estesp/manifest-tool/>

(10) <https://github.com/docker-library/official-images#architectures-other-than-amd64>

(11) <https://blog.docker.com/2017/09/docker-official-images-now-multi-platform/>

## Comment créer vos propres manifest lists ?

Nous allons pour ce faire utiliser ce même outil, le **manifest-tool**. Dans cet exemple, nous allons maintenant créer une **manifest list**, pour exposer virtuellement une image multi-architecture.

Pour cela, nous allons tout d'abord décrire un document précisant le nom de notre **manifest list** et les images sous-jacentes qui lui sont associées. Le contenu d'une **manifest list** parle d'elle-même :

- La **manifest list** est un fichier **YAML** ;
- Sous l'unique **référence** *myrepo/debian:jessie* taggée **jessie** ;
- On trouve 3 **manifests** pour 3 images **Docker** différentes, toutes 3 référencées dans des organisations différentes, et chacune destinée à une architecture processeur spécifique.

```
debian-jessie.yaml
image: myrepo/debian:jessie
manifests:
-
  image: debian/jessie
  platform:
    architecture: amd64
    os: linux
-
  image: arm/debian/jessie
  platform:
    architecture: arm
    os: linux
-
  image: ppc64le/jessie
  platform:
    architecture: ppc64le
    os: linux
```

L'enjeu suivant consiste tout simplement à alimenter le **Hub**, ou votre **registry** avec cette nouvelle **manifest list**. Toujours à l'aide du **manifest-tool**, il suffit de faire ce qui suit :

```
/manifest-tool --username myuser --password mypassword push from-spec debian-jessie.yaml
Digest: sha256:0c7d930875707b58a763a12566f4d938be731376efc59c543bcbe739d
a80082a 743
```

Notez que cette **manifest-list** n'est pas récursive : celle-ci ne peut pointer que vers de véritables images ; référencer la manifest-list générera une erreur lors de l'émission de cette **manifest-list** sur le hub.

Il serait bien sûr possible d'utiliser ensuite le **manifest-tool** pour aller inspecter cette nouvelle **manifest list** fraîchement générée comme nous l'avons fait tout à l'heure pour l'image python. De même, vous la verrez par exemple apparaître sur l'interface graphique du **Hub**.

Cependant, vous n'y verrez cette fois-ci pas apparaître les différentes architectures supportées ou d'autres informations précisant qu'il s'agit d'une **manifest-list**, contrairement à ce que l'on a pu voir tout à l'heure pour une image officielle.

En effet, **Docker** n'a pas encore exposé nativement dans la CLI les fonctionnalités permettant de générer et manipuler soi-même ces **manifest lists** (d'où l'utilisation du **manifest-tool** qui permet de profiter du fait que le hub est déjà prêt).

C'est d'ailleurs tout l'enjeu de la *pull request* 138(12) portée par Christy L

Norman Perez, actuellement en cours de revue : celle-ci vise justement à exposer les nouvelles commandes **docker manifest create|annotate |push**.

## Le runtime Docker devient multiplateforme

A ce stade, nous disposons donc d'un **Hub**, ou de **registries**, capables de référencer virtuellement sous un même nom des images disponibles pour des architectures différentes. Mais revenons un peu en arrière dans le temps, et refocalisons-nous cette fois-ci sur le démon **Docker**.

Il faut attendre avril 2016 et la version 1.11 du **runtime** pour que **Docker** supporte officiellement les processeurs **ARM64**.

En septembre 2016, il devient possible de lancer un conteneur **IIS** sur un **runtime Docker** porté par **Windows 2016 Server**.

Et le mois suivant, sur la scène du Palais des Congrès, lors de la conférence *MS Experiences 16*, vos serveurs démontrent(13) la possibilité de monter un cluster **Swarm** multi-architectures avec des nœuds **Raspberry PI**, des nœuds **Windows Server 2016** dans **Azure** et des nœuds **Linux-x86**. *Docker runs (almost) everywhere !*

L'enthousiasme est cependant vite ravalé : OK, en labo (et en démo), le concept multi-architecture est bel et bien fonctionnel. Et l'on peut enfin *packager* et déployer un conteneur **Apache** pour **Windows** de la même façon qu'un **Apache** pour **Linux** : via un **Dockerfile**. Mais cela se fait au prix d'une orchestration un peu rigide dans laquelle il faut orienter les images à déployer sur tel ou tel nœud selon qu'il s'agit d'une architecture **x86**, **ARM**, **Ms Windows**, etc. Et surtout, cette orchestration par discrimination d'architecture processeur et OS n'est pas nativement portée par les outils **Docker** : il faut gérer cela via des règles d'anti-affinité ou dans un outil de plus haut niveau (**Ansible**, les *pipelines Jenkins*, ou autres).

Techniquement, il reste donc possible de demander à **Docker** d'exécuter sur une machine **x86**, une image conçue pour une autre architecture (ex. **ARM**) : il en découlera une erreur souvent difficile à interpréter (apparaissant parfois sous la forme d'une *exec format error*).

## Et la lumière fut : docker engine supporte désormais les manifest lists

Pour résoudre le problème que nous venons d'évoquer, il nous suffit désormais de référencer à notre démon **Docker**, une **manifest list**, et non directement une image.

En effet, le démon **Docker** est mis à jour pour s'adapter à ce nouveau fonctionnement.

Depuis la référence d'image invoquée (ex. : *myrepo/debian:jessie*) et selon l'architecture processeur (**PowerPC 64-bits**) et le système d'exploitation (**Linux**) qui portent le **runtime Docker**, le client va implicitement sélectionner l'image **Docker** appropriée décrite dans la *manifest list*. Ici, l'image **ppc64le/debian:jessie**.

Cela fonctionne avec une commande **docker run** traditionnelle, mais aussi avec une commande **docker service create**, si l'on s'adresse à un *cluster swarm*.

Et cela nous ouvre de nouveaux horizons : quelques paragraphes plus haut, nous vous parlions d'hybridation, en cumulant dans un même cluster des ressources aux architectures variées. Imaginons que l'on crée un service reposant sur une image multi-architectures, et que l'on sollicite

(12) <https://github.com/docker/cli/pull/138>

(13) <https://fr.slideshare.net/adrienblind/docker-cloud-hybridation-orchestration>

la création de plusieurs réplicas, ou conteneurs associés à ce service. L'orchestrateur déploiera alors des conteneurs sur tous les nœuds dont l'architecture correspondra à l'un des pointeurs référencés. En revanche, il ne déploiera pas de conteneurs sur les nœuds dont l'architecture ne correspond à aucun pointeur.

Dans ce dernier exemple, nous créons un service cette fois-ci attaché à notre **manifest list** plus une image traditionnelle :

```
docker service create --name myapp --detach=true --replicas 3 myrepo/debian:
jessie
m41acjc71o7sv3z4a6dc69slc
```

```
docker service ps myapp
```

ID	NAME	IMAGE	NODE	DESIRED STATE
CURRENT STATE	ERROR	PORTS		
g4ztbaz4vxf	myapp.1	myrepo/debian:jessie	vm1	Running
Running 6 seconds ago				
xu9e4ytdvc6u	myapp.2	myrepo/debian:jessie	pi1	Running
Running 4 seconds ago				
r3rpuhvf1w	myapp.3	myrepo/debian:jessie	pi2	Running
Running 2 seconds ago				

On voit ici que nous avons instancié des conteneurs associés à une même **manifest list**, sur des serveurs dont les architectures diffèrent (la **vm** étant en architecture **amd64**, et les **RPi** en **ARM**). Donc en apparence, on a bien la possibilité de faire fonctionner le même service sur des architectures différentes ; techniquement cependant, ce sont bien des images différentes, créées spécifiquement pour chaque architecture, qui sont exécutées.

Cela nous amène à un point d'attention particulier : pour que cela fonctionne, il faut que les images bâties pour chacune des architectures ait un comportement similaire (utilisation des mêmes variables d'environnement de configuration, même programme/paramètres à démarrer, etc.). Un contre exemple parfait serait par exemple de faire une **manifest-list** référençant l'image officielle de Consul et celle de Hypriot : les paramètres et points d'entrées étant différents, il en résulte

un comportement différent à l'exécution des conteneurs selon l'architecture, et donc l'image, qui est utilisée.

## CONCLUSION

A travers cet article, on se rend compte de l'enjeu que représente cette nouvelle fonctionnalité des images **Docker multi-arch**.

Répondant à une attente originelle de la communauté, **Docker** adresse enfin des plateformes aux architectures variées. Il aura fallu la patience de 2 ans d'efforts de développement communautaire sur les différentes briques de l'écosystème (la **registry**, le **runtime**, la **CLI**) pour que cette gestion se fasse de manière (presque) transparente.

Et ça arrive à point nommé ! Avec le *machine learning* et l'intelligence artificielle, l'industrie se montre de plus en plus friande d'architectures à processeurs spécialisés (**FPGA**, **TensorFlow Process Unit** et autres **GPUs**), tandis que les datacenters commencent à se remplir de serveurs **Linux-ARM** à consommation réduite pour servir les traitements Web plus classiques. Enfin, "l'edge computing et l'IoT sont sur toutes les lèvres", amenant à intégrer là encore les architectures **ARM**. Docker se présente donc plus que jamais comme un outil facilitant le *packaging* et le déploiement d'un composant logiciel quel qu'il soit, en différenciant très facilement chaque architecture technique sous-jacente. Ainsi gérer des *pipelines* de *software factory* portant des tests multi-architectures, ou déployer des applications complexes reposant sur des composants **MS Windows** et **Linux** s'avère plus simple que jamais.

Reste malgré tout du travail pour finaliser ces fonctionnalités. Tout d'abord intégrer le **manifest-tool** à la **CLI Docker**, comme on l'a vu. Mais aussi étendre cette fonctionnalité aux autres "activités" de **Docker**. Je pense en particulier à la commande **docker build**, qui gagnerait là encore en simplicité et en autonomie si elle savait adresser en une seule commande la construction des différentes images selon les plateformes disponibles sur la chaîne de build continu. De même, une extension des capacités du multi-stage build prenant en compte des instructions conditionnelles selon la plateforme à laquelle le build est destinée permettrait, là encore, de gagner en simplicité : le **Dockerfile** devenant alors un véritable outil d'orchestration de la chaîne de build continu.

# 1 an de Programmez! ABONNEMENT PDF : 35 €



Abonnez-vous directement sur : [www.programmez.com](http://www.programmez.com)

Partout dans le monde.

# Algorithmes Génétiques : application au problème du Voyageur de Commerce

• Sylvain Saurel  
sylvain.saurel@gmail.com  
Développeur Android  
<https://www.ssaurel.com>

*La première partie de cet article consacré aux Algorithmes Génétiques nous aura permis de découvrir leur fonctionnement mais également leur mise en application en Java au travers de la résolution d'un problème simple. Dans cette seconde partie, nous allons appliquer les Algorithmes Génétiques pour résoudre le célèbre problème du Voyageur de Commerce.*

Partie 2

Bien connu des informaticiens, le problème du Voyageur de Commerce fait figure de classique parmi les problèmes d'optimisation. Étant donné une ville et des distances entre chacune de ces villes, il s'agit de déterminer un plus court chemin visitant chaque ville une et une seule fois, tout en terminant dans la ville de départ (figure 1). En dépit de l'apparente simplicité de ce problème, il n'existe pas à l'heure actuelle d'algorithme permettant de trouver rapidement une solution exacte dans tous les cas. [1]. Ce problème d'algorithmique célèbre va donc se révéler être un terrain de jeu idéal pour une mise en application plus poussée des Algorithmes Génétiques en Java.

## Modélisation d'une Ville

La première étape dans la création d'un programme de résolution du problème du Voyageur de Commerce consiste à modéliser les différents éléments du problème. Nous commençons donc par modéliser une ville, au sein d'une classe `City`, maintenant des coordonnées cartésiennes, et permettant de connaître la distance entre la ville courante et une autre ville via une méthode `distanceFrom` appliquant le classique théorème de Pythagore. Ceci nous donne le code suivant :

```
public class City {
    private int x;
    private int y;

    public City(int x, int y) {
        this.x = x;
        this.y = y;
    }

    // Getters et Setters ...

    // Calcul de la distance entre 2 villes
    public double distanceFrom(City city) {
        double deltaXSq = Math.pow((city.getX() - getX()), 2);
        double deltaYSq = Math.pow((city.getY() - getY()), 2);
        double distance = Math.sqrt(Math.abs(deltaXSq + deltaYSq));

        return distance;
    }
}
```

## Modélisation d'une Route

L'étape suivante nous conduit à modéliser une route au sein d'une classe éponyme `Route` prenant en entrée un individu, représenté par un objet

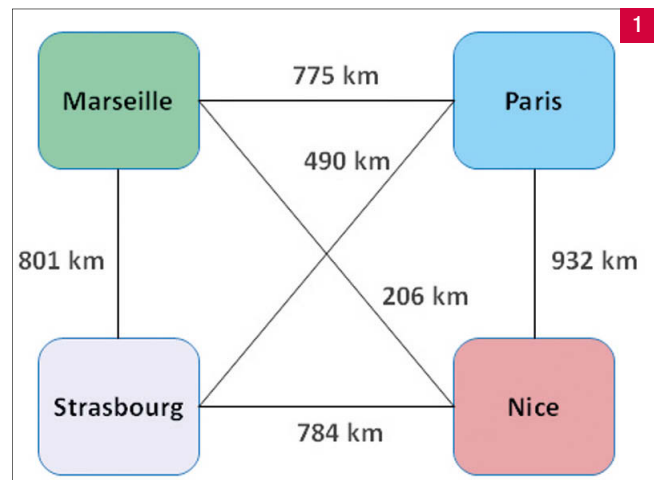


Figure 1 : Problème du Voyageur de Commerce

Individual comme vu dans la première partie de cet article, et une liste de villes. Le constructeur va ensuite créer la route à partir des informations contenues dans les gènes du chromosome de l'instance d'Individual. L'objet `Route` propose une méthode `getDistance` calculant et stockant la distance totale à parcourir pour voyager au travers de toutes les villes de la route, en tenant compte de l'ordre spécifié. Ceci donne le code suivant :

```
public class Route {
    private City route[];
    private double distance = 0;

    // Création de la route à partir de l'individu
    public Route(Individual individual, City cities[]) {
        int chromosome[] = individual.getChromosome();
        route = new City[cities.length];

        for (int geneIndex = 0; geneIndex < chromosome.length; geneIndex++) {
            route[geneIndex] = cities[chromosome[geneIndex]];
        }
    }

    // Calcul de la distance totale
    public double getDistance() {
        if (distance > 0) {
            return distance;
        }
    }
}
```

```
double totalDistance = 0;

for (int cityIndex = 0; cityIndex + 1 < route.length; cityIndex++) {
    totalDistance += route[cityIndex].distanceFrom(route[cityIndex + 1]);
}

totalDistance += route[route.length - 1].distanceFrom(route[0]);
distance = totalDistance;

return totalDistance;
}
}
```

## Modélisation d'un Individu

Dans tout Algorithme Génétique, il y a la notion d'individus au sein d'une population comme vu dans la première partie de cet article. La classe `Individual` modélisant un individu doit être légèrement adaptée pour son application au problème du Voyageur de Commerce. Ainsi, la construction est différente puisque nous allons utiliser chaque numéro de ville disponible pour les gènes du chromosome. De fait, si le chromosome a une taille de  $n$ , les gènes seront valorisés de 0 à  $n-1$  avec des opérations de crossover et de mutation qui se chargeront de modifier ces valeurs par la suite au fur et à mesure des générations de populations. Enfin, on ajoute une méthode `containsGene` à cette classe vérifiant la présence d'une ville, via son index, au sein des gènes d'un chromosome d'un individu, ce qui donne donc le code suivant :

```
public class Individual {

    private int[] chromosome;
    private double fitness = -1;

    public Individual(int chromosomeLength) {
        int[] individual = new int[chromosomeLength];

        for (int gene = 0; gene < chromosomeLength; gene++) {
            individual[gene] = gene;
        }

        chromosome = individual;
    }

    // Présence d'une ville donnée au sein des gènes du chromosome
    public boolean containsGene(int gene) {
        for (int i = 0; i < chromosome.length; i++) {
            if (chromosome[i] == gene) {
                return true;
            }
        }

        return false;
    }

    // Reste de la classe identique à la première partie de l'article ...
}
```

## Modélisation de la Population

La population d'individus utilisée dans le cadre de notre Algorithme Génétique est modélisée via une classe `Population`, en tous points identique à celle présentée dans la première partie de l'article. Pour résumer, la population est composée d'une liste d'objets `Individual` et possède une propriété `populationFitness`, permettant de connaître le niveau d'adaptabilité moyen des membres de la population. Les méthodes principales étant `getFittest` retournant le  $n$ -ième élément le plus adapté de la population à une génération donnée. Enfin, la méthode `shuffle` permettant de brasser les individus de la population dans le but de renforcer le côté aléatoire des opérations d'évolution que sont le crossover et la mutation.

## Abstraction de l'Algorithme Génétique

L'abstraction de l'Algorithme Génétique définie via la classe `GeneticAlgorithm` dans la première partie de l'article demeure identique d'un point de vue des méthodes exposées. Néanmoins, puisque le problème est différent, il est nécessaire d'adapter l'implémentation de ses méthodes. En premier lieu, on rajoute un paramètre de configuration supplémentaire dans la classe nommée `tournamentSize` qui permettra d'augmenter la part de hasard lors de la sélection d'un parent candidat au crossover.

Comme expliqué lors de la définition du problème du Voyageur de Commerce, il n'existe pas de solution exacte dans tous les cas, ce qui nous amène à définir un nombre maximum de générations de population à créer pour trouver une solution optimale avant d'arrêter la boucle de création de générations. Ceci est réalisé au sein de la méthode `isTerminationConditionMet` :

```
public boolean isTerminationConditionMet(int generationsCount, int maxGenerations) {
    return (generationsCount > maxGenerations);
}
```

Pour le problème du Voyageur de Commerce, la valeur du niveau d'adaptabilité est inversement proportionnelle au total de la distance d'une route qui est calculée au sein de la classe `Route`. Pour refléter ces changements, la méthode `calcFitness` prend la forme suivante :

```
public double calcFitness(Individual individual, City cities[]){
    Route route = new Route(individual, cities);
    double fitness = 1 / route.getDistance();

    // Stockage de la valeur d'adaptabilité pour une route donnée
    individual.setFitness(fitness);

    return fitness;
}
```

L'évaluation de la population se fait dans la méthode `evalPopulation` en appelant la méthode `calcFitness` des individus de la population passée en entrée. Le niveau d'adaptabilité moyen des individus, c'est-à-dire la qualité des solutions potentielles, est calculé et stocké dans l'instance de classe `Population` considérée :

```
public void evalPopulation(Population population, City cities[]){
    double populationFitness = 0;
```

```
for(Individual individual : population.getIndividuals()) {
    populationFitness += calcFitness(individual, cities);
}

double avgFitness = populationFitness / population.size();
population.setPopulationFitness(avgFitness);
}
```

La sélection d'un individu parent pour le crossover est modifiée avec l'introduction d'un tournoi de sélection. On crée ainsi une population d'individus avant de brasser la population actuelle passée en paramètre. On place ensuite dans la nouvelle population le nombre d'individus défini par tournamentSize pour rajouter une part de hasard dans la sélection du parent. Il ne reste au final plus qu'à retourner comme parent, l'individu le plus prometteur via un appel à la méthode getFittest(0) :

```
public Individual selectParent(Population population) {
    // Tournoi de sélection
    Population tournament = new Population(tournamentSize);
    // Brassage de la population actuelle
    population.shuffle();

    for (int i = 0; i < tournamentSize; i++) {
        Individual tournamentIndividual = population.getIndividual(i);
        tournament.setIndividual(i, tournamentIndividual);
    }

    // On renvoie le plus prometteur
    return tournament.getFittest(0);
}
```

Le croisement demeure identique dans ses grands principes, puisque l'on applique le crossover suivant le pourcentage défini en entrée, et en préservant les éléments les plus prometteurs. En revanche, le mécanisme en lui-même est adapté au problème du Voyageur de Commerce, et l'on va donc sélectionner tout d'abord un parent via le tournoi de sélection réalisé au sein de selectParent. On crée ensuite un chromosome initialisé à vide.

Nous générons alors 2 positions aléatoires au sein du chromosome, en positionnant la plus basse en début et la plus haute en fin. Dans un premier temps, on copie le contenu du premier parent situé entre les positions basses et hautes au sein du chromosome du nouvel individu. Dans un second temps, on parcourt les gènes du second parent situés après la position haute, et avant la position basse. Si l'individu en cours de fabrication n'a pas la ville parcourue en son sein, c'est-à-dire que sa méthode containsGene retourne faux pour l'index de ville en cours d'examen, alors on l'ajoute en cherchant une position de rechange. Finalement, on ajoute le nouvel individu créé, à la population de la génération suivante pour arriver au code suivant :

```
public Population crossoverPopulation(Population population){
    Population newPopulation = new Population(population.size());

    for (int populationIndex = 0; populationIndex < population.size(); populationIndex++) {
        // Parent 1
        Individual parent1 = population.getFittest(populationIndex);
```

```
// Critère d'application du crossover
if (crossoverRate > Math.random() && populationIndex >= elitismCount) {
    // Sélection du parent 2 via tournoi de sélection
    Individual parent2 = selectParent(population);

    // Création du nouvel individu
    int offspringChromosome[] = new int[parent1.getChromosomeLength()];
    Arrays.fill(offspringChromosome, -1);
    Individual offspring = new Individual(offspringChromosome);

    // Préparation des index
    int substrPos1 = (int) (Math.random() * parent1.getChromosomeLength());
    int substrPos2 = (int) (Math.random() * parent1.getChromosomeLength());

    final int startSubstr = Math.min(substrPos1, substrPos2);
    final int endSubstr = Math.max(substrPos1, substrPos2);

    // Copie de la partie centrale du parent 1 sur le nouvel individu
    for (int i = startSubstr; i < endSubstr; i++) {
        offspring.setGene(i, parent1.getGene(i));
    }

    // Parcours des gènes du parent 2
    for (int i = 0; i < parent2.getChromosomeLength(); i++) {
        int parent2Gene = i + endSubstr;

        if (parent2Gene >= parent2.getChromosomeLength()) {
            parent2Gene -= parent2.getChromosomeLength();
        }

        // Si l'individu n'a pas la ville, on l'ajoute
        if (!offspring.containsGene(parent2.getGene(parent2Gene))) {
            // On cherche la position de rechange
            for (int ii = 0; ii < offspring.getChromosomeLength(); ii++) {
                if (offspring.getGene(ii) == -1) {
                    offspring.setGene(ii, parent2.getGene(parent2Gene));
                    break;
                }
            }
        }
    }

    newPopulation.setIndividual(populationIndex, offspring);
} else {
    newPopulation.setIndividual(populationIndex, parent1);
}

return newPopulation;
}
```

Enfin, la modification de la classe GeneticAlgorithm se termine par la méthode mutatePopulation dont le contenu est en grande partie identique à ce que nous avons vu dans le problème trivial résolu au cours de la première partie de cet article. Identique à cela près, que la mutation ne peut se faire en inversant simplement les valeurs 0 et 1 sur les gènes du chromoso-

me des individus. Dans le cas du problème du Voyageur de Commerce, on va inverser les valeurs de 2 gènes aléatoirement comme suit :

```
public Population mutatePopulation(Population population){
    Population newPopulation = new Population(populationSize);

    for (int populationIndex = 0; populationIndex < population.size(); populationIndex++) {
        Individual individual = population.getFittest(populationIndex);

        // On préserve les individus élites
        if (populationIndex >= elitismCount) {
            for (int geneIndex = 0; geneIndex < individual.getChromosomeLength(); geneIndex++) {
                // Critère d'application de la mutation
                if (mutationRate > Math.random()) {
                    // Position d'échange pour la mutation
                    int newGenePos = (int) (Math.random() * individual.getChromosomeLength());
                    // Echange
                    int gene1 = individual.getGene(newGenePos);
                    int gene2 = individual.getGene(geneIndex);
                    individual.setGene(geneIndex, gene1);
                    individual.setGene(newGenePos, gene2);
                }
            }
        }

        newPopulation.setIndividual(populationIndex, individual);
    }

    return newPopulation;
}
```

## Résolution du problème

La dernière étape consiste à mettre en musique les différentes parties de notre Algorithme Génétique définies jusqu'alors, afin de résoudre le problème du Voyageur de Commerce. Nous commençons donc par définir le nombre de générations que nous allons générer afin d'essayer de trouver une solution optimale au problème via une constante MAX\_GENERATIONS. Nous créons ensuite les villes en générant leurs coordonnées cartésiennes de manière aléatoire. Il est temps alors d'instancier un objet GeneticAlgorithm en passant en entrée différents paramètres : nombre de villes, pourcentage de mutations, pourcentage de croisements, taille du pool d'élites, et enfin, la taille de la population pour le tournoi de sélection du crossover.

Un appel à la méthode initPopulation avec le nombre de villes en entrée nous renvoie ainsi la population initialisée. Une évaluation de la population permet de calculer la qualité des routes faisant partie de la population d'individus de la génération 0. On affiche alors la meilleure route actuellement connue en créant un objet Route avec, en entrée, le meilleur individu de la population et la liste des villes considérées.

La boucle de création de générations peut démarrer avec comme condition de terminaison le retour de la méthode isTerminationConditionMet. Un appel à la méthode crossoverPopulation permet une application du croisement sur la population avant de passer à l'étape de mutation. On évalue ensuite la population avant d'incrémenter l'index de génération de la population d'individus courante. Après avoir généré MAX\_GENERATIONS générations d'individus, on arrête l'exécution de l'algorithme. Le meilleur individu de la population, qui représente donc notre solution

optimale après MAX\_GENERATIONS générations, est récupéré via un appel à getFittest(0) sur l'objet Population. A partir de cet individu, on crée l'objet Route optimal, et on affiche le résultat. Cela nous donne le code final suivant :

```
public class PbVoyageurDeCommerce {

    public static int MAX_GENERATIONS = 10000;

    public static void main(String[] args) {
        // Création des villes avec leurs coordonnées cartésiennes sur une grille de 100 par 100
        int numCities = 100;
        City cities[] = new City[numCities];

        for (int cityIndex = 0; cityIndex < numCities; cityIndex++) {
            int xPos = (int) (100 * Math.random());
            int yPos = (int) (100 * Math.random());
            cities[cityIndex] = new City(xPos, yPos);
        }

        // Création instance de GeneticAlgorithm
        GeneticAlgorithm ga = new GeneticAlgorithm(100, 0.001, 0.9, 2, 5);
        // Init Population
        Population population = ga.initPopulation(cities.length);
        // Première Evaluation
        ga.evalPopulation(population, cities);

        Route startRoute = new Route(population.getFittest(0), cities);
        System.out.println("Distance meilleure route initiale : " + startRoute.getDistance());

        int generation = 1;

        // Boucle de création de générations
        while (!ga.isTerminationConditionMet(generation, MAX_GENERATIONS)) {
            // Meilleure route après n générations
            Route route = new Route(population.getFittest(0), cities);
            System.out.println("Distance meilleure route Génération " + generation + " : " + route.getDistance());

            population = ga.crossoverPopulation(population);
            population = ga.mutatePopulation(population);
            ga.evalPopulation(population, cities);

            generation++;
        }

        System.out.println("Arrêt de la recherche après " + MAX_GENERATIONS + " générations.");
        Route route = new Route(population.getFittest(0), cities);
        System.out.println("Distance meilleure route : " + route.getDistance());
    }
}
```

L'exécution de notre Algorithme Génétique dans le but de rechercher une solution optimale après 10000 générations sur une instance de problème comprenant 100 villes nous permet de constater l'efficacité de notre implémentation, puisque l'on passe d'une meilleure route dont la

distance valait 5134 (figure 2) à une route optimale dont la distance est tombée à 1159 ! (figure 3). Le tout pour un algorithme ayant tourné moins de 60 secondes, ce qui est plutôt performant. [2] [3]

## CONCLUSION

Ces deux articles consacrés aux Algorithmes Génétiques auront permis de mettre en évidence les grands principes de fonctionnement de cet

outil indispensable du domaine de l'Intelligence Artificielle. La mise en pratique sur un problème classique d'optimisation tel que celui du Voyageur de Commerce montrant à quel point les Algorithmes Génétiques peuvent se révéler performants pour des problèmes dont on ne peut trouver de solution exacte dans tous les cas. En bref, ils constituent un outil qu'il est toujours intéressant de maîtriser, et d'avoir dans sa panoplie de développeur.

```
<terminated> PbVoyageurDeCommerce [Java Application] C:\Program Files\Java\jre1.6.0_18\bin\java
Distance meilleure route initiale : 5134.179613537692
Distance meilleure route Génération 1 : 5134.179613537692
Distance meilleure route Génération 2 : 4991.96088816337
Distance meilleure route Génération 3 : 4938.112465656586
Distance meilleure route Génération 4 : 4817.343030101004
Distance meilleure route Génération 5 : 4740.41442202085
Distance meilleure route Génération 6 : 4732.79287856122
Distance meilleure route Génération 7 : 4609.101438345537
Distance meilleure route Génération 8 : 4602.604811861173
Distance meilleure route Génération 9 : 4475.946531566686
Distance meilleure route Génération 10 : 4321.332389803635
Distance meilleure route Génération 11 : 4321.332389803635
Distance meilleure route Génération 12 : 4233.664652250753
Distance meilleure route Génération 13 : 4166.407304530086
Distance meilleure route Génération 14 : 4107.2272065042225
Distance meilleure route Génération 15 : 3979.869366330993
Distance meilleure route Génération 16 : 3968.879955217595
```

Meilleure Route au démarrage

```
<terminated> PbVoyageurDeCommerce [Java Application] C:\Program Files\Java\jre1.6.0_18\bin\javaw.exe
Distance meilleure route Génération 9982 : 1167.28686005348
Distance meilleure route Génération 9983 : 1167.28686005348
Distance meilleure route Génération 9984 : 1167.28686005348
Distance meilleure route Génération 9985 : 1167.28686005348
Distance meilleure route Génération 9986 : 1167.28686005348
Distance meilleure route Génération 9987 : 1167.28686005348
Distance meilleure route Génération 9988 : 1167.28686005348
Distance meilleure route Génération 9989 : 1159.2396425315362
Distance meilleure route Génération 9990 : 1159.2396425315362
Distance meilleure route Génération 9991 : 1159.2396425315362
Distance meilleure route Génération 9992 : 1159.2396425315355
Distance meilleure route Génération 9993 : 1159.2396425315355
Distance meilleure route Génération 9994 : 1159.2396425315355
Distance meilleure route Génération 9995 : 1159.2396425315355
Distance meilleure route Génération 9996 : 1159.2396425315355
Distance meilleure route Génération 9997 : 1159.2396425315355
Distance meilleure route Génération 9998 : 1159.2396425315355
Distance meilleure route Génération 9999 : 1159.2396425315355
Distance meilleure route Génération 10000 : 1159.2396425315355
Arrêt de la recherche après 10000 générations.
Distance meilleure route : 1159.2396425315355
```

Route optimale trouvée après 10000 générations

# Vous recrutez des développeurs ?

**PROGRAMMEZ!**  
le magazine des développeurs

**peut vous aider**



Contactez-nous : [ftonic@programmez.com](mailto:ftonic@programmez.com)

# ReactiveX avec RxJava : un cas concret



Olivier HUBER  
Responsable Pédagogique  
Formation à **Zenika**

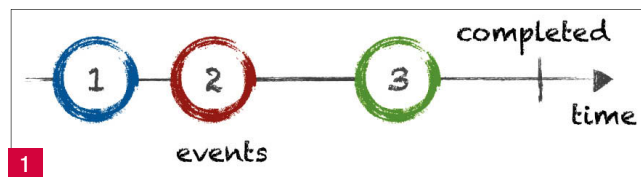
*La programmation réactive est un paradigme de programmation qui gagne en popularité depuis plusieurs années. Hors, lorsqu'on commence à faire des recherches sur le sujet, on se heurte à beaucoup de confusions et c'est bien dommage ; ce n'est pas si compliqué que cela, et, surtout, on peut en tirer des bénéfices immédiats qui vont faciliter notre vie de développeur.*

**E**n premier lieu, il s'agit de comprendre la programmation réactive (avec sa connotation fonctionnelle) et ses bénéfices en regard de la programmation impérative plus classique. Mais il s'agit surtout de changement de paradigme et en particulier de notre réflexe de revenir toujours et encore à notre chère programmation procédurale. Il se trouve que les paradigmes "classiques" comme la programmation impérative deviennent vite difficile à lire et ne peuvent exprimer certains aspects facilement comme la modélisation de flux d'événements, la programmation d'opérations asynchrones ou le temps-réel.

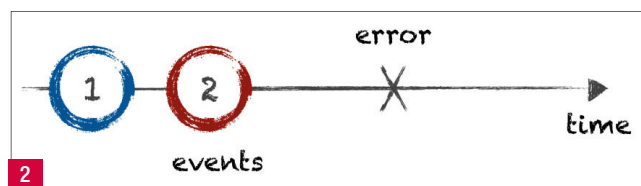
*J'étais parti comme à l'habitude en raisonnant de manière impérative et en introduisant petit à petit des API avec des callback, etc. Résultat des courses : du code qui fonctionne, mais très difficile à relire puis à modifier sans changer les API clientes. Sans parler de la gestion des erreurs lors de plusieurs appels asynchrones. Après avoir compris que "tout est flux" j'ai pu appréhender ce nouveau paradigme et en retirer les bénéfices.* – Olivier HUBER

Le principe de programmer en termes de gestion de flux est aussi vieux que le pipe Unix. Le concept n'est pas nouveau, loin de là.

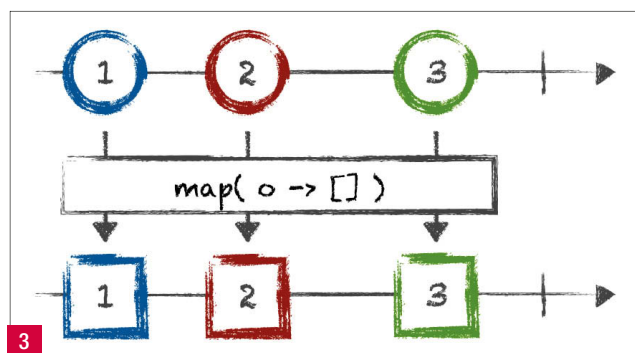
Dans l'esprit Craftsmanship, la programmation réactive améliore la lisibilité de votre base de code ainsi que sa maintenance et facilite avec élégance la résolution de scénarios complexes et ceci avec peu d'opérations. Elle permet une gestion des erreurs asynchrones adaptée et une facilité de programmer de façon concurrente avec un niveau d'abstraction plus élevé. Mais le plus important est que la programmation réactive permet de construire une interface stable et transparente sans se soucier de l'aspect de la concurrence ou du cache, etc. Les détails d'implémentations peuvent être modifiés après coup sans affecter les API utilisées. Il est important de comprendre qu'en programmation réactive, tout est flux. Voici un flux de 3 éléments dans le temps qui finit par retourner une indication de fin de flux. Si cette indication de fin de flux n'arrivait jamais on parlerait de flux infini : [1]



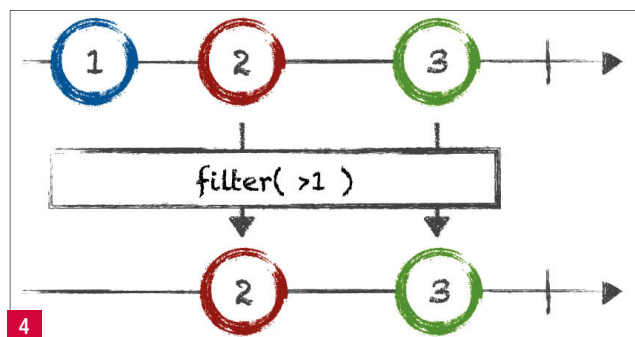
Si une erreur survient on reçoit une indication d'erreur et le flux s'arrête. [2]



Pour manipuler ces flux de données on utilise des opérateurs tels que l'opérateur "map" permettant de transformer chaque élément un à un [3]

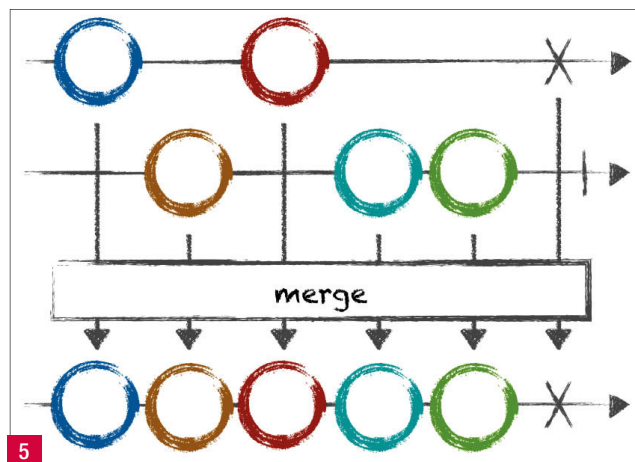


La sortie de l'opérateur est toujours un nouveau flux  
Ici l'opérateur "filter" produit un nouveau flux en ignorant certains éléments [4]



On peut également combiner plusieurs flux à l'aide d'opérateurs tels que "merge" [5]

Que la production des éléments soit synchrone ou asynchrone l'API reste la même pour le développeur qui souscrit au flux.



Je vous propose de rentrer dans le vif du sujet et donc dans l'API ReactiveX. Et plus particulièrement, pour nos exemples, l'implémentation RxJava [6]. En effet vous remarquerez que vous pouvez implémenter ReactiveX dans plus de 18 langages [6]. Voici un simple Hello world en Java standard que l'on affiche dans la console :

```
String greeting = "Hello world!";
System.out.println(greeting);
// Console:
// Hello world!
```

L'équivalent en RxJava sera le suivant :

```
Observable<String> greetingObservable = Observable.just("Hello");
greetingObservable.subscribe(System.out::println);
// Console:
// Hello
```

Vous notez que vos API retourneront dorénavant des Observable et que l'on doit s'abonner à ce flux pour recevoir les "éléments". Vous avez des méthodes doOn... qui vous permettent comme dans notre exemple d'intervenir dans la chaîne du flux pour ici simplement « logger »

```
Observable<String> greetingObservable = Observable.from(Arrays.asList("Hello", "World!"));

greetingObservable
    .doOnSubscribe() -> System.out.println("OnSubscribe")
    .doOnError(error -> System.err.println("OnError:" + error.getMessage()))
    .doOnNext(value -> System.out.println("OnNext:" + value))
    .doOnCompleted() -> System.out.println("OnCompleted")
    .subscribe();

// Console:
// OnSubscribe
// OnNext:Hello
// OnNext:World!
// OnCompleted
```

Si on veut créer un Observable pour avoir la main sur la production d'un flux d'éléments Il est alors important de respecter le contrat Rx [http://reactivex.io/documentation/contract.html] qui précise, entre autres, d'appeler la méthode onCompleted ou onError pour clore le flux :

```
Observable<String> greetingObservable = Observable.create(subscriber -> {
    subscriber.onNext("Hello");
    subscriber.onNext("World!");
    subscriber.onCompleted();
});
```

Alors prenons maintenant un exemple qui peut vous servir, et qui, pour le coup, s'adapte très bien au concept de flux.

Ce fichier Excel [7] n'est pas très "propre". Certaines lignes ou cellules sont vides, bref un fichier de données comme on en traite souvent.

Un code procédural traitant ce fichier ressemblera souvent à ça :

```
public void process(String excelFile, String sheetName) {
    try {
        Workbook workbook = WorkbookFactory.create(new File(excelFile));
        for (int i = 0; i < workbook.getNumberOfSheets(); i++) {
            if (workbook.getSheetName(i).equals(sheetName)) {
                Sheet sheet = workbook.getSheetAt(i);
```

ReactiveX

Introduction

Docs

Languages

Resources

Community

## Languages

- Java: RxJava
- JavaScript: RxJS
- C#: Rx.NET
- C#(Unity): UniRx
- Scala: RxScala
- Clojure: RxClojure
- C++: RxCpp
- Lua: RxLua
- Ruby: Rx.rb
- Python: RxPY
- Go: RxGo
- Groovy: RxGroovy
- JRuby: RxJRuby
- Kotlin: RxKotlin
- Swift: RxSwift
- PHP: RxPHP
- Elixir: reaxive
- Dart: RxDart

### ReactiveX for platforms and frameworks

- RxNetty
- RxAndroid
- RxCocoa

	A	B	C	D	E	F	G
Agence	Nom	Prénom	Embauche	email			
Rennes	Valle	Yolande	30 décembre 2012	yolande.valle@zenika.com			
Paris	Clement	Etienne	3 février 2016				
Nantes	Munoz Gomi	Saturnino	10 octobre 2014	saturnino.munoz-gomez@zenika.com			
Bordeaux	d'Orville	Hans	18 mai 2010	hans.orville@zenika.com			
Lyon	Le Saux	Jean-Yves	15 novembre 2008	jean-yves.lesaux@zenika.com			
Lille	Preis	Anne Belinda	22 juin 2007	anne-belinda.preis@zenika.com			
Rennes	Zhang	Xinsheng	14 juin 2012	xinsheng.zhang@zenika.com			
Rennes	Du	Yue	22 mars 2013	yue.du@zenika.com			
Lille	Gong	Wan	31 juillet 2008	wan.gong@zenika.com			
Rennes	Cheng	Xiaolin	3 juillet 2009		xiaolin.cheng@zenika.com		
Rennes	Branet	Marie-Noelle	28 août 2015		marie-noelle.branet@zenika.com		
Nantes	Crowley	John	2 mai 2007		john.crowley@zenika.com		
Lille	Olsson	Asa	7 juin 2009		asa.olsson@zenika.com		
Rennes	Breines	Ingeborg	28 janvier 2014		ingeborg.breines@zenika.com		

```
for (int rowIndex = 1; rowIndex <= sheet.getLastRowNum(); rowIndex++) {
    Row row = sheet.getRow(rowIndex);
    if (row != null) {
        String email = email(row);
        String location = location(row);
        String lastname = lastname(row);
        String firstname = firstname(row);
        Date employmentDate = employmentDate(row);

        if (employmentDate == null || employmentDate.toInstant().isAfter(Instant.now().minus(365, ChronoUnit.DAYS))) {
            System.out.println(firstname + " " + lastname + " " + "<" + email + ">" + " from " + location
                + " was hired on " + employmentDate);
            if (!StringUtils.equalsIgnoreCase(StringUtils.stripAccents(firstname + " " + lastname +
                "@zenika.com"), email)) {
                System.err.println("The email " + email + " doesn't match the name " + firstname +
                    " " + lastname);
            }
        }
    }
}

workbook.close();
} catch (Exception e) {
    e.printStackTrace();
}
```

Difficile de se replonger rapidement dans la logique de ce code pour en décortiquer les règles métiers. Alors je vous avais déjà dit qu'en programmation réactive "tout est flux", et que l'on devait s'efforcer de changer de

paradigme et pousser le raisonnement jusqu'au bout. Tentons d'appliquer ce principe et nous aurions alors ceci :

```
workbookObservable(excelFile)
// Flux de 0 ou 1 fichier excel <Workbook>
.flatMap(workbook -> Observable.from(workbook::sheetIterator))
// Flux d'onglets <Sheet>
.flatMap(sheet -> Observable.from(sheet::rowIterator))
// Flux de lignes <Row>
.flatMap(row -> Observable.from(row::cellIterator))
// Flux de cellules <Cell>
.subscribe(
    System.out::println,
    System.err::println
);
```

Partant d'un simple nom de fichier on aurait donc dans l'ordre :

- 1 • La création d'un flux de 1 « workbook » Excel ;
  - 2 • Il est transformé dans un flux d'onglet (on va donc "itérer" sur les onglet) ;
  - 3 • Puis on continue la transformation pour arriver à un flux de lignes de chaque onglet ;
  - 4 • Enfin, on finit par itérer sur les cellules de chacune de ces lignes pour avoir un flux de cellules ;
  - 5 • On s'abonne alors à ce flux en fournissant un call-back pour afficher le contenu des cellules et un autre pour afficher une potentielle erreur.
- Donc ces simples lignes vont nous permettre d'afficher simplement toutes les cellules de toutes les lignes de tous les onglets de tout - l'unique (mais flux tout de même) fichier Excel. Alors c'est maintenant que l'on peut s'amuser. Nous avons à notre disposition toutes les briques nécessaires pour interagir avec ces flux de données. On ne sélectionne que l'onglet qui nous intéresse à l'aide de l'opérateur "filter" par exemple Et vous remarquerez qu'à la suite on peut aussi ajouter un opérateur switchIfEmpty détectant que le flux est potentiellement vide pour afficher dans ce cas une erreur explicite

```
workbookObservable(excelFile)
.flatMap(workbook -> Observable.from(workbook::sheetIterator))
.filter(sheet -> sheet.getSheetName().equalsIgnoreCase(sheetName))
.switchIfEmpty(Observable.error(new UnknownSheetException("No sheet
found with name " + sheetName)))
.flatMap(sheet -> Observable.from(sheet::rowIterator))
.flatMap(row -> Observable.from(row::cellIterator))
.subscribe(
    System.out::println,
    System.err::println
);
```

On retire l'en tête de ligne et on fait alors usage de l'opérateur "skip". Enfin au lieu de simplement itérer sur le contenu des cellules on peut transformer cette ligne Excel en un objet Java ou Json représentant notre Contact :

```
workbookObservable(excelFile)
.flatMap(workbook -> Observable.from(workbook::sheetIterator))
.filter(sheet -> sheet.getSheetName().equalsIgnoreCase(sheetName))
.switchIfEmpty(Observable.error(new UnknownSheetException("No sheet
found with name " + sheetName)))
.flatMap(sheet -> Observable.from(sheet::rowIterator))
.skip(1)
```

```
.flatMap(this::buildContactObservable)
.subscribe(
    contact -> System.out.println(contact),
    error -> System.err.println(error.getMessage())
);
```

Le nouveaux flux ainsi transformé délivre des objets Contact, ce qui permet à présent d'appliquer de nouveaux opérateurs pour appliquer diverses règles métiers :

- Une règle sur la validité du mail du contact ;
- Une règle sur la date d'embauche ;
- Une règle pour contrôler le nom par rapport au mail.

Au final au lieu d'afficher le contact dans la console, on peut par exemple envoyer un mail aux contacts en question.

```
workbookObservable(excelFile)
.flatMap(workbook -> Observable.from(workbook::sheetIterator))
.filter(sheet -> sheet.getSheetName().equalsIgnoreCase(sheetName))
.switchIfEmpty(Observable.error(new UnknownSheetException("No sheet found with
name " + sheetName)))
.flatMap(sheet -> Observable.from(sheet::rowIterator))
.skip(1)
.flatMap(this::buildContactObservable)
.compose(new FilterEmail())
.compose(new FilterEmploymentDateForTheLast(600, ChronoUnit.DAYS))
.compose(new CheckEmailAgainstFullName())
.subscribe(
    contact -> sendEmailInvitation(contact),
    error -> System.err.println(error.getMessage())
);
```

Bref cela devient un jeu d'enfant de manipuler les données et nous avons un code lisible, modulable. Un bon craftsman aurait pu refactorer le code procédural que je vous ai montré au début, l'API ReactiveX offre déjà des dizaines d'opérateurs permettant de gérer les cas les plus complexes. Je vous laisse découvrir le code complet et son explication dans notre projet Github <https://github.com/olivierhuber/reactivex-rxjava> J'espère que vous trouverez du plaisir à mettre en pratique la programmation réactive et qu'elle vous facilitera pleinement la tâche dans vos projets au quotidien.

Vous pouvez retrouver mes articles sur le blog Zenika :

<https://blog.zenika.com/2016/06/16/programmation-reactive-une-entree-en-matiere/>

<https://blog.zenika.com/2017/02/15/reactivex-avec-rxjava-un-cas-concret/>

Et mes vidéos sur la chaîne <https://www.youtube.com/user/ZenikaITv>

## OLIVIER HUBER, un passionné de création logicielle.

Pour lui, notre métier est l'un des rares à nous donner la capacité de matérialiser une idée rapidement et quasiment gratuitement. Initié il y a plus de dix ans à l'agilité par la pratique de l'eXtrême Programming, il devient Agiliste convaincu et Scrum Master par la suite. Il intervient pendant 16 ans en développement / conseil / formation chez des grands comptes sur des besoins Java, JEE et Web de façon générale, à la fois côté serveur (Spring, Hibernate, JPA et leurs amis) ou client (AngularJS, GWT, Wicket, jQuery, Twitter Bootstrap).

Il rejoint Zenika en septembre 2008 pour évoluer parmi des collègues qui ont la même ferveur afin de partager au maximum ses connaissances (formations, conférences, ...). En 2014, il passe meta-formateur comme responsable pédagogique et accompagne les formateurs tout en continuant d'intervenir sur divers évènements techniques.

# Software Defined **Radio FM** avec décodage RDS

On trouve des puces permettant de recevoir la radio FM et décoder les messages RDS à moins d'un dollar (RDA5807M, SI4703, SI4705). Elles sont si petites qu'elles sont généralement vendues préinstallées sur un circuit imprimé (PCB).

• Killian Siou

Elles ont en commun de se piloter via un bus I<sup>2</sup>C. Il a été développé au début des années 1980 par Philips. Il existe maintenant une multitude de composants utilisant ce standard de communication, dont nos microcontrôleurs favoris. Nous aurions pu choisir d'utiliser un contrôleur de la famille des Arduino pour ce projet. Mais, le Wemos D1 Mini propose des composants empilables intéressants, un contrôleur plus rapide, plus d'espace de stockage et le wifi, offrant beaucoup plus d'évolution possible pour un prix entre 3 (version lite) et 5 dollars (version pro). Je reste toujours stupéfait de trouver des machines aussi puissantes dans 2,5 grammes.

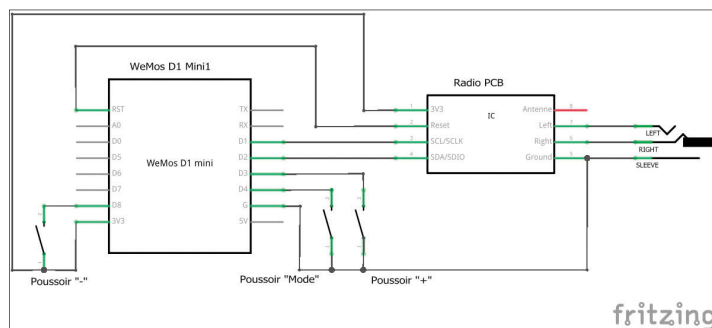
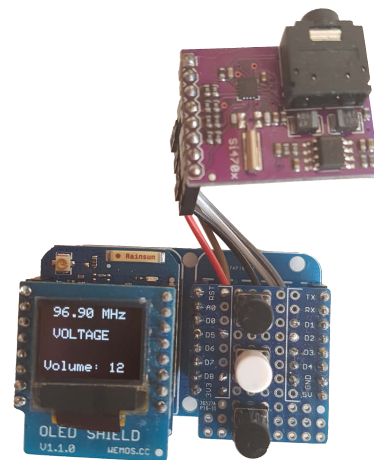
	D1 mini Lite	D1 Mini V2.2.0	D1 mini Pro
Contrôleur	ESP-8285 (dérivé du ESP-8266)	ESP-8266	ESP-8266
Flash	1 Mo	4 Mo	16 Mo
Wifi	Oui	Oui	Oui, Céramique et connecteur IPEX pour antenne externe
Interface USB	CH340G	CH340G	CP2104
Prix	3\$	4\$	5\$

Nous utiliserons l'écran OLED empilable (4,5 dollars). Mais, tout autre écran OLED SSD1306 conviendra. Il faudra toutefois ajuster l'affichage en fonction de la taille de l'écran.

## Le montage

Les connections (identiques pour toute les versions du D1 Mini) :

Pin	Function	ESP-8266 Pin	Nous utiliserons
TX	TXD	TXD	
RX	RXD	RXD	
A0	Analog input, max 3.3V	A0	
D0	IO	GPIO16	
D1	IO, SCL	GPIO5	Bus I <sup>2</sup> C – SCL/SCLK
D2	IO, SDA	GPIO4	Bus I <sup>2</sup> C – SDA/SDIO
D3	IO, 10k Pull-up	GPIO0	Bouton poussoir + vers la masse
D4	IO, 10k Pull-up, BUILTIN_LED	GPIO2	Bouton poussoir "menu" vers la masse
D5	IO, SCK	GPIO14	(uniquement pour Si478x) Reset
D6	IO, MISO	GPIO12	
D7	IO, MOSI	GPIO13	
D8	IO, 10k Pull-down, SS	GPIO15	Bouton poussoir - vers le 3,3V
G	Ground	GND	
5V	5V	-	
3V3	3.3V	3.3V	3,3V
RST	Reset	RST	



Connecter les broches de la partie radio en fonction du modèle choisi. Les RDA5807M ne nécessitent pas de la broche « Reset ». Autour de 0,5\$ c'est le composant le moins cher du marché. Par contre, il faut de bons yeux ou une bonne loupe pour réaliser les soudures. Car, le circuit imprimé est généralement très petit sur les modèles que j'ai pu trouver (1cmx1cm). Il faut aussi prévoir d'ajouter un jack femelle.

Sur les Si470x ne cherchez pas l'antenne, elle est intégrée dans la sortie audio. Il faut prévoir un long câble pour une bonne réception. Ils sont un peu plus cher (3-4 \$). Mais, les circuits imprimés de ce modèle sont généralement plus faciles à souder et la sortie jack audio est intégrée. Nous voilà donc avec un montage pour un peu plus de 11 dollars auquel il faudra associer à une enceinte portable. En général ces dernières possèdent une sortie USB qui nous servira d'alimentation.

## La programmation

Matthias Hertel a réalisé une bibliothèque pour les composants radio proposés ( <http://mathertel.github.io/Radio/> ).

Deux petites adaptations pour correspondre au D1 mini, il faut changer dans le fichier SI4703.cpp :

- La ligne 36

```
#define resetPin 2
```

Par :

```
#define resetPin D5
```

- Et la ligne 38:

```
#define SDIO A4
```

Par :

```
#define SDIO SDA
```

Afin de supporter la résolution 64x48 de l'écran proposé par Wemos, il faut ajouter la librairie, Adafruit\_SSD1306 alternative depuis [https://github.com/mcauser/Adafruit\\_SSD1306.git](https://github.com/mcauser/Adafruit_SSD1306.git) (Ou changer les valeurs par défaut dans le fichier Adafruit\_SSD1306.h)

Afin de se rassurer sur la qualité du montage, il est utile de tester si l'écran et le tuner sont bien visibles sur le bus I<sup>2</sup>C. Si tout est en place, le programme « I<sup>2</sup>C Scanner » fourni en exemple avec la librairie Wemos vous retournera les adresses de l'écran et du tuner. Code sur [programmez.com](https://github.com/mcauser/Adafruit_SSD1306.git). Bonne écoute !

# JShell : Java se dote enfin d'un Shell

• Sylvain Saurel  
sylvain.saurel@gmail.com  
Développeur Android  
<https://www.ssaurel.com>

*Java 9 marque l'arrivée de la modularisation au sein du JDK avec notamment le fameux projet Jigsaw qui est enfin arrivé à son terme. Dans cet article, nous vous proposons de partir à la découverte de JShell qui met (enfin) un Shell à disposition des développeurs Java.*

Que l'attente fut longue ! C'est, en substance, la première réaction des développeurs Java, suite à la sortie officielle de Java 9 le 21 Septembre 2017. Après divers reports, la nouvelle mouture de Java est enfin disponible. Si cette réaction peut paraître exagérée concernant la sortie de Java 9 qui aura connu un peu plus d'un an de retard, elle s'avère être un doux euphémisme concernant la modularisation du JDK. En effet, les travaux concernant Jigsaw ont commencé dès 2008. Il aura ainsi fallu attendre près de 10 ans pour que la modularisation du JDK soit implémentée de manière effective au sein de la plateforme Java ! Une éternité dans un monde informatique en perpétuelle évolution. Jigsaw et la modularisation du JDK constituent clairement la partie centrale de Java 9. De fait, on aurait tendance à oublier que Java 9 apporte également un certain nombre d'autres nouveautés qui viendront également impacter la vie des développeurs. Parmi ces nouveautés, on pourra citer JShell dont nous allons partir à la découverte au sein de cet article.

## Qu'est-ce que JShell ?

Un Shell est un outil bien connu des développeurs et présent dans bon nombre de langages de programmation. Répondant également à l'acronyme REPL pour Read-Eval-Print-Loop, un Shell va mettre à disposition des développeurs un environnement interactif au sein duquel ils pourront entrer des expressions plus ou moins complexes, les évaluer et obtenir directement en retour le résultat de leur évaluation à l'écran.

Les outils REPL sont très répandus au sein de langages de script populaires comme Python ou Node par exemple. Ils ont été adoptés depuis plusieurs années maintenant par des langages bâtis sur la JVM comme Clojure ou Groovy. Les équipes d'Oracle en charge de Java ne pouvaient donc pas continuer à ignorer cette tendance et ont donc décidé, avec Java 9, de mettre à disposition JShell, le Shell officiel de la plateforme Java.

JShell devrait devenir rapidement très populaire en facilitant l'apprentissage du langage Java pour les débutants grâce aux possibilités de programmation interactive qu'il offre. En sus, JShell permettra aux développeurs plus aguerris de tester du code rapidement sans avoir à lancer un environnement de développement complet. Tester une nouvelle bibliothèque de code deviendra chose aisée et l'on peut également imaginer la création de scripts Java remplaçant les scripts Bash habituels pour tester la disponibilité d'un service Web, ou effectuer des opérations de transfert de fichiers, puisque JShell permet l'exécution de scripts Java enregistrés précédemment.

## Lancement de JShell

Avant toute chose, il vous faudra télécharger Java 9 sur le site d'Oracle prévu à cet effet : <http://jdk.java.net/9/>. Comme dit précédemment, JShell est un composant standard du JDK sous Java 9. Par conséquent, une fois Java 9 installé, vous pourrez le lancer simplement au sein d'un terminal via la commande `jshell` et évaluer une première expression pour vérifier que tout fonctionne correctement :

```
$ jshell
| Welcome to JShell -- Version 9
| For an introduction type: /help intro
```

```
==> 2+3
$1 ==> 5
==>
```

Parmi les fonctionnalités intéressantes offertes par JShell, on notera la possibilité de charger des bibliothèques de code externes au format JAR mais également via le nouveau format des modules Java introduit avec Jigsaw. Ici, nous pouvons choisir de charger la bibliothèque JUnit en utilisant la variable d'environnement `CLASSPATH` comme de coutume au lancement de JShell :

```
$ jshell --class-path junit-4.12.jar
| Welcome to JShell -- Version 9
| For an introduction type: /help intro
==> import org.junit.Test
==>
```

Le chargement d'un module au format Jigsaw peut se faire au lancement de JShell via l'option `--add-modules` :

```
$ jshell --add-modules jdk.incubator.httpclient
| Welcome to JShell -- Version 9
| For an introduction type: /help intro
==> import jdk.incubator.http.*;
==>
```

Il est également possible de lister l'ensemble des imports de bibliothèques externes ou de modules réalisés au cours d'une session courante de JShell en exécutant la commande `/imports` :

```
==> /imports
| import java.io.*
| import java.math.*
| import java.net.*
| import java.nio.file.*
| import java.util.*
| import java.util.concurrent.*
| import java.util.function.*
| import java.util.prefs.*
| import java.util.regex.*
| import java.util.stream.*
```

Enfin, il est possible à tout moment d'obtenir de l'aide en exécutant la commande `/help` :

```
==> /help
| Type a Java language expression, statement, or declaration.
| Or type one of the following commands:
| /list [<name or id> | -all | -start]
| list the source you have typed
| /edit <name or id>
| edit a source entry referenced by name or id
| /drop <name or id>
| delete a source entry referenced by name or id
| /save [-all | -history | -start] <file>
```

```
| Save snippet source to a file.
| /open <file>
| open a file as source input
| /vars [<name or id> | -all | -start]
| list the declared variables and their values
| /methods [<name or id> | -all | -start]
| list the declared methods and their signatures
| /types [<name or id> | -all | -start]
| list the declared types
| /imports
| list the imported items
```

## Expressions simples

Les expressions évaluables de JShell sont constituées par tout ce qui peut être exprimé en Java. Nous pouvons prendre une chaîne de caractères et effectuer quelques opérations de base proposées par l'objet String :

```
==> "JShell sur Programmez !".substring(11).toUpperCase()
$1 ==> "PROGRAMMEZ !"
==>
```

Le résultat de l'évaluation de chacune des expressions entrées dans JShell est assigné à une variable générée. Ainsi, le résultat de l'opération concernant la chaîne "JShell sur Programmez !" a été assigné à la variable \$1 que l'on peut utiliser par la suite :

```
==> printf("%s", $1)
PROGRAMMEZ !
```

Vous aurez également remarqué, avec joie, qu'avec JShell, l'utilisation du point-virgule en fin de ligne devient optionnelle.

## Typage et définition de variables

Lancé en mode verbeux avec l'option `--feedback verbose`, JShell s'avèrera être un allié précieux pour les débutants souhaitant connaître le type des expressions évaluées. En effet, un débutant pourra se demander le type du résultat d'une division entre un nombre entier ou un nombre flottant :

```
==> 13 / 22
$1 ==> 0
| created scratch variable $1 : int
==> 2.0 / 4.0
$2 ==> 0.5
| created scratch variable $2 : double
==> 3.0 / 6
$3 ==> 0.5
| created scratch variable $3 : double
==> 10 / 20.0
$4 ==> 0.5
| created scratch variable $4 : double
==>
```

Néanmoins, JShell conserve la possibilité de définir une variable avec son type comme cela est fait habituellement en Java. Les mots-clés permettant de définir la visibilité d'une variable sont également disponibles bien qu'ils n'aient aucun effet au cours d'une session JShell où tout le code exécuté se trouve au sein du même scope :

```
==> int i = 1;
i ==> 1
==> i = 2;
i ==> 2
```

```
==> private int j;
j ==> 0
==> public int k;
k ==> 0
```

En outre, les mots clés `final` et `static` sont utilisables mais renverront un avertissement à l'évaluation puisqu'ils n'auront aucun effet. Une variable déclarée `final` pourra donc être assignée à une nouvelle valeur.

```
==> final int l = 3;
| Warning:
| Modifier 'final' not permitted in top-level declarations, ignored
| final int l = 3;
| ^--^
l ==> 3
==> static int m = 3;
| Warning:
| Modifier 'static' not permitted in top-level declarations, ignored
| static int m = 3;
| ^---^
m ==> 3
==> l = 4;
l ==> 4
```

Enfin, une variable déjà définie au cours d'une session JShell peut être à tout moment redéfinie avec un nouveau type. La nouvelle définition de la variable remplaçant immédiatement l'ancienne avec remise à la valeur par défaut du nouveau type :

```
==> int i = 1;
i ==> 1
==> long i;
i ==> 0
==>
```

## Déclaration de méthodes

La déclaration de méthodes de haut niveau constitue une des fonctionnalités les plus puissantes de JShell. Elles peuvent être déclarées directement dans l'invite de commandes et ensuite être utilisées au sein de la session JShell courante. Elles se comportent comme des méthodes statiques utilisables à tout moment sans avoir à instancier un objet. Définissons une méthode permettant d'additionner 2 entiers et appelons ensuite cette méthode :

```
==> int add(int a, int b){
...> return a + b;
...> }
| created method add(int,int)
==> add(10, 20);
$4 ==> 30
==>
```

Comme pour la déclaration de variables, JShell permet de déclarer à nouveau une méthode au sein de la même session. La nouvelle déclaration venant remplacer la précédente. De fait, il devient aisé de tester pas à pas une méthode et sa définition :

```
==> int add(int a, int b){
...> return a - b;
...> }
| modified method add(int,int)
==> add(10, 20);
```

```
$5 ==> -30
==>
```

## Déclaration de classes

Continuant notre découverte de JShell, nous pouvons passer à la déclaration de classes. Une fois une classe déclarée, elle devient utilisable au sein de la session JShell courante. Dans cet exemple, nous allons créer un objet `Person` avec une propriété ainsi que le `getter` et le `setter` associés de la même manière que nous aurions pu le faire dans fichier Java habituellement :

```
==> class Person{
...> private String name;
...> public String getName(){
...> return this.name;
...> }
...> public void setName(String name){
...> this.name = name;
...> }
...> }
| created class Person
```

L'instanciation de l'objet `Person` nouvellement déclaré et l'utilisation de ses méthodes se faisant comme de coutume :

```
==> Person p = new Person()
p ==> Person@ff5b71f
==> p.setName("Sylvain Saurel")
==> p.getName()
$8 ==> "Sylvain Saurel"
==>
```

Là encore, il est possible à tout moment de redéfinir la classe `Person` et JShell remplacera l'ancienne définition par la nouvelle en l'écrasant. Tout naturellement, JShell permet l'héritage entre classes ou l'implémentation d'interfaces définies (ou non) lors de la session courante :

```
==> class Man extends Person { }
| created class Man
==> class Woman extends Person { }
| created class Woman
```

## Import de classes

Comme vous avez pu le constater jusqu'à présent, JShell met à disposition des développeurs tout ce que le langage Java autorise. Pour accéder aux classes des modules non chargés par défaut par le JDK modulaire de Java 9, on pourra utiliser le nom complet d'une classe :

```
==> java.time.Instant.now();
$10 ==> 2017-10-06T15:20:14.500250Z
```

Il est également possible d'importer directement une classe et tout son contenu ou mieux encore de ne charger que les méthodes statiques que l'on souhaite utiliser :

```
==> import java.time.Instant;
==> Instant.now();
$11 ==> 2017-10-06T15:30:09.600589Z
==> import static java.time.Instant.now;
==> now();
$12 ==> 2017-10-06T15:35:40.196620Z
==>
```

Cependant, en important uniquement une méthode statique d'un objet,

on obtiendra une erreur si l'on essaie d'appeler une autre méthode statique du dit objet :

```
==> java.time.Instant.then();
| Error:
| cannot find symbol
| symbol: method then()
| java.time.Instant.then();
| ^-----^
==>
```

## Gestion des exceptions

Décidément puissant, JShell se charge également de la gestion des exceptions. Chaque fois qu'une commande exécutée dans JShell va se lancer avec une exception, cette dernière sera automatiquement gérée et la stack trace Java complète sera alors affichée avec le numéro des lignes de la même manière qu'avec un programme Java classique. Ceci se révélant essentiel pour diagnostiquer et résoudre des problèmes lorsque l'on travaille au sein d'une session JShell :

```
==> String upperThenSub(String input){
...> String upper = input.toUpperCase();
...> return upper.substring(50);
...> }
| created method upperThenSub(String)
==> upperThenSub("JShell est dans Programmez !");
| java.lang.StringIndexOutOfBoundsException thrown: String index out of range: -22
| at String.substring (String.java:1852)
| at test (#1:3)
| at (#2:1)
==>
```

Ici, JShell nous précise que nous avons une exception *StringIndexOutOfBoundsException* à la ligne 3 au sein de la méthode `upperThenSub`, première commande de notre session. Nous pouvons revoir la déclaration de cette méthode en exécutant la commande `/list` avec le nom de la méthode en paramètre :

```
==> /list upperThenSub
1 : String upperThenSub (String input){
String upper = input.toUpperCase();
return upper.substring(50);
}
==>
```

Petit bémol ici avec la commande `/list` qui ne numérote pas les lignes de la méthode affichée ce qui peut s'avérer pénible pour des méthodes assez longues.

## CONCLUSION

La plupart des langages modernes mettent à disposition des développeurs un Shell autorisant la programmation interactive. Avec JShell, la plateforme Java fait désormais partie de cette catégorie de langage. Cette fonctionnalité faisait clairement défaut et Java 9 vient corriger un manque criant. Outil puissant, JShell viendra faciliter le travail d'apprentissage du langage Java tout en permettant aux plus aguerris de s'en servir pour tester rapidement des bibliothèques de code ou réaliser directement des scripts Java remplaçant les habituels scripts Bash. En bref, si JShell est resté relativement méconnu jusqu'alors, il est fort à parier qu'il connaisse un grand succès et devienne une des nouveautés de Java 9 préférée des développeurs. •

# Un serveur REST Web API en C++



Christophe PICHAUD  
Consultant sur les technologies  
Microsoft  
christophepichaud@hotmail.com  
www.windowsscnp.net



*Imaginez que vous puissiez développer le back-end de votre solution logicielle sur un serveur avec le roi des langages qu'est le C++ ? Sur Linux ? Oui. Sur Windows ? Oui. Un seul code portable ? Oui. Le tout sans complexité délirante, juste avec des classes et des méthodes de traitement... Et on retourne du JSON pour un front-end en JS ? Chiche ! Suivez-moi, je vous explique...*

## Casablanca alias C++ REST SDK

Pour développer une Web API, beaucoup d'entre nous ne connaissent que ASP.NET et C#. Depuis MFC ISAPI et ATL Server, Visual C++ ne fournit plus de templates de projets pour des développements serveur. Un SDK reprend le flambeau : le REST SDK, nom de code « Casablanca ». Ce SDK écrit en C++ moderne (C++ 11) propose de réaliser du code serveur et client en utilisant les derniers patterns, à savoir, le support du http et du JSON.

Disponible sur Github à l'adresse suivante : <https://github.com/Microsoft/cpprestsdk>. Ce SDK propose aussi des exemples de code pour commencer en douceur.

## Un serveur http REST simple

Examinons le code minimum pour créer un serveur Web API qui retourne du JSON. Premièrement, il faut définir une classe qui contient un `http_listener`.

```
using namespace web;
using namespace http;
using namespace utility;
using namespace http::experimental::listener;

class MyServer
{
public:
    MyServer() {}
    MyServer/utility::string_t url;

    pplx::task<void> open() { return m_listener.open(); }
    pplx::task<void> close() { return m_listener.close(); }

private:
    static void handle_get(http_request message);
    static void handle_put(http_request message);
    static void handle_post(http_request message);
    static void handle_delete(http_request message);

    http_listener m_listener;
};
```

Ensuite, il faut configurer ce listener en lui fournissant un URI sur lequel on se met en écoute ainsi que les handlers pour les opérations GET, PUT, POST et DELETE. Mais avant tout, il faut déclarer le serveur comme étant une variable globale comme suit :

```
std::unique_ptr<MyServer> g_http;
```

Ensuite, il faut préparer les éléments indispensables au serveur, à savoir son URI d'écoute :

```
utility::string_t port = U("34568");
utility::string_t address = U("http://localhost:");
```

```
address.append(port);

uri_builder uri(port);
uri.append_path(U("MyServer/Action/"));

auto addr = uri.to_uri().to_string();
g_http = std::unique_ptr<MyServer>(new MyServer(addr));
g_http->open().wait();

ucout << utility::string_t(U("Listening for requests at: ")) << addr << std::endl;
```

L'appel à `open()` active l'écoute. Reste à découvrir le code du constructeur de la classe `MyServer`, car il faut enregistrer les handlers sur les méthodes HTTP :

```
MyServer::MyServer/utility::string_t url) : m_listener(url)
{
    std::function<void(http_request)> fnGet = &MyServer::handle_get;
    m_listener.support(methods::GET, fnGet);

    std::function<void(http_request)> fnPut = &MyServer::handle_put;
    m_listener.support(methods::PUT, fnPut);

    std::function<void(http_request)> fnPost = &MyServer::handle_post;
    m_listener.support(methods::POST, fnPost);

    std::function<void(http_request)> fnDel = &MyServer::handle_delete;
    m_listener.support(methods::DEL, fnDel);
}
```

OK, le corps du serveur http est là... Regardons le corps des méthodes HTTP. Nous allons en coder une seule mais le système est simple.

```
void MyServer::handle_post(http_request message)
{
    ucout << message.to_string() << endl;
    message.reply(status_codes::OK);
};

void MyServer::handle_delete(http_request message)
{
    ucout << message.to_string() << endl;
    message.reply(status_codes::OK);
};

void MyServer::handle_put(http_request message)
{
    ucout << message.to_string() << endl;
    message.reply(status_codes::OK);
};
```

Ces méthodes ne sont pas implémentées mais patience. Voici ce que

nous allons coder. Sur la méthode GET si on nous envoie « get\_developers », alors on dump une structure de données en JSON.

## Le corps du handler GET

La structure de données JSON retournée est définie comme suit :

```
{ "job": "Developers", "people": [ { "age": 25, "name": "Franck" }, { "age": 20, "name": "Joe" } ] }
```

Voyons le corps de cette méthode GET :

```
void MyServer::handle_get(http_request message)
{
    ucout << U("Message") << U(" ")
        << message.to_string() << endl;

    ucout << U("Relative URI") << U(" ")
        << message.relative_uri().to_string() << endl;

    auto paths = uri::split_path(uri::decode(message.relative_uri().path()));
    for (auto it1 = paths.begin(); it1 != paths.end(); it1++)
    {
        ucout << U("Path") << U(" ")
            << *it1 << endl;
    }

    auto query = uri::split_query(uri::decode(message.relative_uri().query()));
    for (auto it2 = query.begin(); it2 != query.end(); it2++)
    {
        ucout << U("Query") << U(" ")
            << it2->first << U(" ") << it2->second << endl;
    }

    auto queryltr = query.find(U("request"));
    utility::string_t request = queryltr->second;
    ucout << U("Request") << U(" ") << request << endl;

    if (request == U("get_developers"))
    {
        Data data;
        data.job = U("Developers");
        People p1;
        p1.age = 25;
        p1.name = U("Franck");
        data.peoples.push_back(p1);
        People p2;
        p2.age = 20;
        p2.name = U("Joe");
        data.peoples.push_back(p2);

        utility::string_t response = data.AsJSON().serialize();
        ucout << response << endl;

        message.reply(status_codes::OK, data.AsJSON());
        return;
    }

    message.reply(status_codes::OK);
};
```

On commence par dumper les éléments reçus en entrée ? On affiche la requête, l'URL demandée et les paramètres. Vous remarquerez que `uri::split_query` retourne une map de string pour récupérer les paramètres de la requête. Si la requête est « get\_developers », alors on déclare une structure de données et on génère du JSON. [1]

## La génération du JSON

Voyons le code qui permet de générer du JSON à partir d'une structure de données. Il n'y a rien de magique, regardez :

```
struct People
{
    utility::string_t name;
    double age;

    static People FromJSON(const web::json::object & object)
    {
        People result;
        result.name = object.at(U("name")).as_string();
        result.age = object.at(U("age")).as_integer();
        return result;
    }

    web::json::value AsJSON() const
    {
        web::json::value result = web::json::value::object();
        result[U("name")] = web::json::value::string(name);
        result[U("age")] = web::json::value::number(age);
        return result;
    }
};
```

Cette structure représente un développeur. Il y a 2 propriétés : name et age. Le mécanisme de transformation d'une chaîne se fait avec les méthodes `object.at().as_string()` et `web::json::value::string()`. Pour un entier, c'est avec `object.at().as_integer()` et `web::json::value::number()`. C'est assez simple. Pour la structure suivante, qui met en œuvre un tableau (vector), il faut utiliser d'autres méthodes :

```
struct Data
{
    std::vector<People> peoples;
    utility::string_t job;

    Data() {}

    void Clear() { peoples.clear(); }
```

```
static Data FromJSON(const web::json::object &object)
{
    Data res;

    web::json::value job = object.at(U("job"));
    res.job = job.as_string();

    web::json::value p = object.at(U("people"));
    for (auto iter = p.as_array().begin(); iter != p.as_array().end(); ++iter)
    {
        if (!iter->is_null())
        {
            People people;
            people = People::FromJSON(iter->as_object());
            res.peoples.push_back(people);
        }
    }

    return res;
}

web::json::value AsJSON() const
{
    web::json::value res = web::json::value::object();
    res[U("job")] = web::json::value::string(job);

    web::json::value jPeoples = web::json::value::array(peoples.size());

    int idx = 0;
    for (auto iter = peoples.begin(); iter != peoples.end(); iter++)
    {
        jPeoples[idx++] = iter->AsJSON();
    }

    res[U("people")] = jPeoples;
    return res;
}
};
```

Pour générer un tableau en JSON, il faut utiliser la fonction `web::json::value::array`. Ensuite on itère sur le vector pour associer les éléments du tableau un par un.

## La partie client

Le côté client peut être développé avec n'importe quelle technologie à partir du moment où il est permis de déclencher des requêtes http et de lire du JSON. Ici, le client est écrit aussi en C++ pour vous montrer comme c'est facile d'utiliser le REST SDK :

```
#ifdef _WIN32
# define iequals(x, y) (_stricmp((x), (y))==0)
#else
# define iequals(x, y) boost::iequals((x), (y))
#endif

int wmain(int argc, wchar_t *argv[])
```

```
{
    utility::string_t port = U("34568");
    if(argc == 2)
    {
        port = argv[1];
    }

    utility::string_t address = U("http://localhost:");
    address.append(port);
    http::uri uri = http::uri(address);

    http_client client(http::uri_builder(uri)
        .append_path(U("/MyServer/Action/")).to_uri());

    while (true)
    {
        std::string method;
        ucout << "Enter method.";
        cin >> method;

        http_response response;

        if (iequals(method.c_str(), "get_developers"))
        {
            utility::ostringstream_t buf;
            buf << U("?request=")
                << utility::conversions::to_string_t(method)
                << U("&city=Paris");

            response = client.request(methods::GET, buf.str()).get();

            ucout << U("Response ") << response.to_string() << endl;

            json::value jdata = json::value::array();
            jdata = response.extract_json().get();
            Data data = Data::FromJSON(jdata.as_object());
        }
        else
        {
            ucout << utility::conversions::to_string_t(method)
                << ": not understood\n";
        }
    }

    return 0;
}
```

## CONCLUSION

Le C++ REST SDK est facile à mettre en œuvre et permet d'avoir des fonctionnalités serveur sans difficultés particulières. Le support du JSON n'est pas trop pénible. Les opérations sont simples et la performance est au rendez-vous, et ceci en standard. L'avantage de la solution proposée c'est qu'elle ne dépend pas de IIS. Le module serveur est *stand-alone*. Pour finaliser l'application, il faut lui ajouter une couche de service Windows et là, c'est la fête !

# La plateforme MSX

• Daniel Tizon

Principal Consultant chez

Trivadis SA (Suisse)

Au plus près des technologies Microsoft, du MSX jusqu'à Azure et Azure Stack

Complètement incompatibles entre elles, ces machines à base de processeur Z80 (8 bits) répondaient au doux nom de ZX Spectrum, ZX81, Amstrad CPC464, Commodore 64, Alice. C'est à ce moment qu'arrive le fameux MSX.

Pourquoi le MSX est-il important dans l'histoire de la micro-informatique ? Simplement parce qu'il est le fruit du rapprochement de deux sociétés très en vogue à l'époque, le Japonais ASCII et l'américain MICROSOFT. Leurs fondateurs respectifs, Kazuhiko Nishi et Bill Gates, tous deux âgés de 28 ans se sont associés pour créer le MSX. Leur objectif était de créer un standard, avec la promesse d'une compatibilité totale des programmes ou cartouches d'une machine MSX à l'autre, quel que soit sa marque. La répartition des rôles était simple, ASCII définissait les spécifications matérielles et Microsoft s'occupait d'écrire le système d'exploitation et le langage de programmation, le MSX-BASIC.

A sa sortie, le standard MSX était supporté par plus de 15 marques la plupart japonaises. Il s'agissait de Canon, Casio, Fujitsu, Hitachi, Kyocera, Mitsubishi, NEC, Pioneer, Sanyo, Sharp, Sony, Toshiba, Victor/JVC ou encore Yamaha. Se sont joint également les Coréens Goldstar, Daewoo et Samsung, l'Américain Spectravideo ainsi que l'Européen Philips.

Les années 1983 et 1984 ont marqué le début de la micro-informatique familiale. C'était le début de la micro-informatique, le moment où les premiers ordinateurs sont rentrés dans les foyers. De nombreux constructeurs se sont lancés, aucun standard n'existait. Il était très difficile de s'y retrouver. Cadeau incontournable de Noël 1984, le magazine SVM (Science & Vie Micro) sortait son numéro spécial, « Noël : 50 ordinateurs familiaux, une méthode pour choisir ». Pour des parents, commander son micro-ordinateur à Noël était l'assurance que son enfant s'éveille aux technologies modernes !



## La norme MSX [1]

Tout ordinateur MSX se devait d'avoir un certain nombre de caractéristiques communes afin d'assurer cette compatibilité. Comme toute norme, elle a évolué au cours des années. Après le MSX, sont apparus le MSX 2, puis le MSX 2+, et enfin le MSX Turbo-R. [15]

Le MSX était architecturé autour d'un processeur Z80 (8 bits). Si la norme prévoyait 8 Ko de RAM au minimum, en règle générale les MSX en avaient 32 ou 64 Ko. Côté graphique, il y avait un processeur graphique dédié, et une mémoire vidéo dédiée également. Les capacités graphiques maximales étaient 256x212 pixels en 16 couleurs. Le langage de développement était le MSX-BASIC, un basic considéré comme l'un des plus puissants de l'époque (Gestion des sprites avec gestion des collisions, instruction Circle, et plutôt rapide). Le fait d'avoir une séparation de la mémoire centrale

et de la mémoire vidéo n'était pas toujours perçu comme un avantage pour le développement sur cette plateforme (nécessité de transférer des données de l'un à l'autre, avec un coût en performances).

Le processeur sonore était sur 3 voix (polyphonique) et sur 7 octaves. Les jeux pouvaient être sur un support K7 audio, ou sous forme de cartouche d'une capacité de 16 ou 32 Ko. Le

lecteur de disquette est arrivé un peu plus tard, sous forme optionnelle sur le port cartouche ou port d'extension, avec un système d'exploitation MSX-DOS, une variante du CP/M.

Le MSX 2 avait un certain nombre d'améliorations avec plus de mémoire, un meilleur processeur graphique qui supportait 256 couleurs et non plus 16, dont le support de l'incrustation vidéo, la sauvegarde de la date et heure par une pile. Le lecteur de disquettes était présent en standard, avec support des simples et double densités (360 Ko et 720 Ko) au format 3,5 pouces ce qui était moderne pour l'époque ; un standard qui a supplanté le format 5,25 pouces. Les jeux étaient désormais proposés sur disquettes, ou sur cartouche MegaRom, c'est-à-dire des cartouches contenant une ROM de 128, 256 ou 512 Ko.

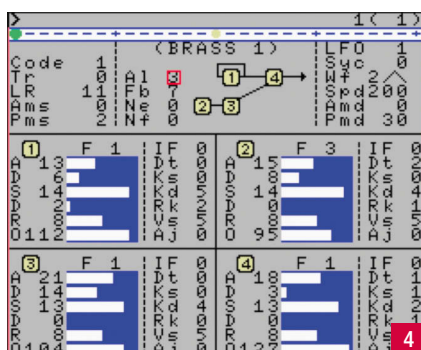
Le MSX 2+ apportait peu de nouveautés par



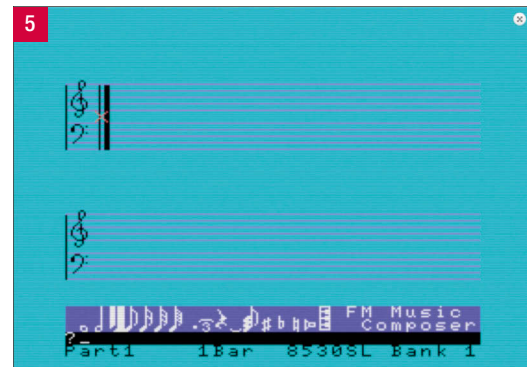
MSX Yamaha CXSMII (1985)



SFG-05 Voice menu (F3)



YRM-502 Voice Editor



FM Music Composer (1984)

rapport au MSX 2, si ce n'est le passage à 19268 couleurs, mais concrètement les jeux ne pouvaient pas exploiter ce mode d'affichage qui faisait sens par contre pour l'affichage d'images digitalisées. L'introduction du MSX-MUSIC compatible Yamaha YM2413 pour une synthèse FM sur 9 canaux qui s'ajoutait au processeur sonore 3 canaux du MSX 1. Sans être dans la norme, tous les MSX 2+ l'intégraient, et un certain nombre de jeux en ont tiré profit.

Le MSX Turbo-R fut la dernière évolution du MSX. Sa principale particularité était son architecture autour d'un processeur 16 bits ASCII R800 cadencé à 7 MHz, une belle promesse en termes de performances. En plus il possédait également un processeur Z80 à 3.5 MHz comme ses prédécesseurs, afin de conserver une compatibilité ascendante. Panasonic sortira un premier modèle, le FS-A1-ST équipé de 256 ko de mémoire vive début 1991. Un second modèle voit le jour en fin d'année, le FS-A1-GT, qui, lui, est équipé d'origine de 512 ko et dispose en plus d'une interface MIDI ainsi que d'une ROM spécifique de 512 Ko comportant le logiciel MSX VIEW (un tout en un traitement de texte, agenda, etc.).

## Un MSX pour quoi faire ?

Essentiellement pour jouer, mais aussi pour faire de la musique, de la photo et du montage vidéo. Formidable outil pédagogique, son langage de programmation qu'il contenait, le MSX BASIC permettait de ne plus rester passif devant sa télé mais de devenir acteur (jeu) ou créatif pour ceux qui se lançaient dans la programmation. A propos de programmation, le magazine Programmez n'existait pas encore à l'époque mais son équivalent s'appelait « Hebdogiciel », il contenait des listings de code

sources à retaper, en BASIC ou en assembleur, avec des sources et auteurs différents selon les plateformes ZX Spectrum, ZX81, Amstrad CPC464, Commodore 64, Alice, MSX,...

## Quelques MSX marquants

Outre le respect d'un standard, certains ordinateurs MSX avaient des particularités qui en faisaient des machines d'exception pour des usages bien précis.

**Le Yamaha CX5MII** était vendu principalement comme un instrument de musique dans des boutiques spécialisées aux côtés des synthétiseurs DX. Bien qu'un véritable MSX1, c'était une machine améliorée conçue pour la production de sons et de musique. Cet ordinateur disposait d'un générateur de son FM (SFG-05) intégré qui le transformait en un vrai synthétiseur avec générateur de son FM, jusqu'à 8 canaux de musique, 4 opérateurs, 8 algorithmes, avec 48 sons prédéfinis. En fait, il était l'équivalent du synthétiseur DX-9.

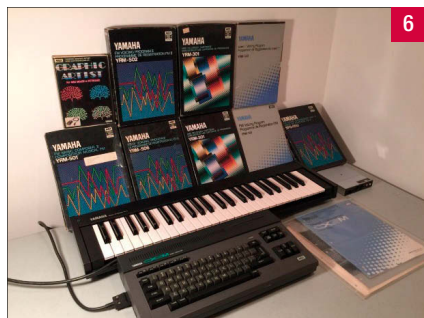
Son logiciel principal pouvait être lancé depuis le BASIC avec la commande CALL MUSIC ou \_MUSIC. Les autres, comme le Voice Editor ou le Music Composer étaient fournis sous forme de cartouches additionnelles. [2][3][4][5]

Mais surtout, ce MSX disposait d'un connecteur MIDI lui permettant de connecter un clavier musical YK-01, YK-10 ou YK-20 (4 octaves) vendu séparément, pour faire de la composition ou rejouer une partition avec le FM Music Composer. D'autre part, Yamaha a produit un paquet de cartouches : [6]

## Cartouches Yamaha dédiées à la musique

UDC-01 Data Memory Cartridge  
YRM-101/YRM-01 FM Music Composer  
YRM-102/YRM-12 FM Voicing Program  
YRM-103 DX7 Voicing Program  
YRM-104/YRM-11 Music Macro  
YRM-105 DX9 Voicing Program  
YRM-301/YRM-31 MIDI Recorder  
YRM-302/YRM-32 RX Editor  
YRM-303 MIDI Macro & Monitor  
YRM-304/YRM-34 DX7 Voicing Program II  
YRM-305 DX21 Voicing Program  
YRM-501 FM Music Composer II  
YRM-502 FM Voicing Program II  
YRM-504 Music Macro II  
YRM-506 FB-01 Voicing Program

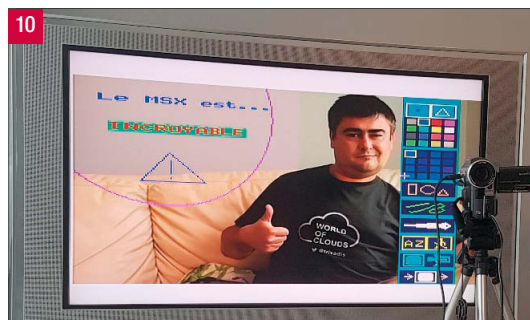
**Le Philips NMS-8280 [7]** est un MSX 2 sorti en 1987 qui était très prisé par les amateurs de Photo/Vidéo. En effet, cette machine était capable de digitaliser (capturer des images) à partir d'une source vidéo comme votre télé en RVB ou une source vidéo composite provenant de votre caméscope VHS. Il permettait aussi de réaliser des effets spéciaux sur des vidéos grâce au mixage par surimposition d'images numériques générées par votre MSX au-dessus d'une source vidéo. Cet ordinateur avait plusieurs potentiomètres sur sa façade avant pour contrôler le mixage vidéo, audio, les niveaux de digitalisation vidéo et un bouton pour sélectionner la source d'entrée. Ce MSX 2 disposait de 128Ko de RAM et 128 Ko de Video RAM, d'une souris et d'un clavier séparé, 2 ports cartouches sur son côté gauche, mais surtout de 2 lecteurs de disquettes 720K. [8][9][10]



Philips NMS-8280 façade arrière



Photos digitalisées puis éditées depuis le NMS-8280.



Capture et incrustation vidéo avec le logiciel Video Graphics

Le **Sony HB-700** est un des MSX 2 le plus répandus en France. Pourvu de 256 Ko de RAM, 128 Ko de VRAM, d'un lecteur de disquettes 720 Ko, il faisait très professionnel. Sa capacité mémoire exceptionnelle le rendait également très attractif pour faire tourner ses copies de sauvegardes de cartouches MegaRoms (cf. chapitre « copies de sauvegardes »).



Disposant d'un clavier séparé de l'unité centrale et d'une souris, comme le Philips, il était livré avec 2 disquettes, l'une contenant un logiciel appelé « Hybrid » pouvant être considéré comme un système d'exploitation graphique multilingue (Anglais, Italien, Espagnol, Français, Néerlandais, Allemand). La seconde était une disquette de démo pour montrer les capacités des MSX 2 et de cette machine en particulier. [11][12][13][14]

Le **MSX Turbo-R** [15] de Panasonic est le dernier et ultime MSX de tous les temps. Il est le successeur du MSX 2+, et en partage donc toutes ses caractéristiques. Doté d'une mémoire vidéo dédiée de 128 ko, il peut afficher jusqu'à 19268 couleurs en 256\*212 (ou 256\*424 en entrelacé). Les autres modes graphiques MSX 1 et MSX 2 sont bien sûr toujours supportés. Il est également équipé d'une puce sonore Yamaha YM-2413 OPLL (MSX-Music introduit en standard avec le MSX 2+) produisant jusqu'à 9 voix FM. Les nouveautés sont qu'il bénéficie de 256 Ko ou 512 Ko de RAM, et du dernier né des micro-processeurs, créé par ASCII, le R800 à 14 Mhz. C'est un processeur 16 bits, rétro-compatible

avec le Z80 qui équipait jusqu'alors tous les modèles de MSX des générations précédentes. Sa vitesse est équivalente à un Z80 à 28,6 Mhz, soit 8 fois plus rapide qu'un MSX de base. D'autre part, il possède une nouvelle puce de son PCM qui peut produire des sons digitalisés à 44 KHz. Le Turbo-R est donc très doué pour afficher de belles images digitalisées et faire de la musique FM. Un lecteur de disquettes 2DD, 2 ports cartouches, deux ports joysticks, une touche de « Pause » hardware et un Rensha-Turbo (Autofire réglable) complètent son équipement.

Au final, le MSX Turbo-R était un MSX 2+ amélioré. Il ne sortira qu'une poignée de jeux pour cette machine d'exception. Le premier Seed Of Dragon était assez décevant, mais sont sortis ensuite Fray, un très beau RPG, puis Illusion City très réussi également.

Panasonic était donc la seule société à produire le MSX Turbo-R, le FS A1ST et le FS A1GT. Le GT avait 512 Ko de RAM, une interface MIDI et une ROM de 512 Ko contenant l'environnement graphique MSX View. Panasonic a stoppé la production du Turbo-R lorsqu'il a lancé sa console de jeux 3DO en 1992.



15

## Les prémisses de la Publication Assistée par Ordinateur (PAO) [16]

Le premier, l'unique, le fameux logiciel de PAO (Publication Assistée par Ordinateur) n'est arrivé qu'en 1987 sur MSX, et sur MSX2 pour être précis. Il s'appelait « Dynamic Publisher ». C'est avec que j'ai fait mes premiers exposés et comptes-rendus de travaux pratiques au lycée alors que je n'étais qu'en classe de seconde. Bien mieux que la machine à écrire, j'étais pratiquement le seul de ma classe à utiliser un logiciel de PAO en 1987. Bien que la mise en page se fasse tout en Bitmap et l'imprimante était une imprimante à aiguilles en noir & blanc, et ça produisait son effet ! [16][17]

## Pionnier dans la production de photos numériques

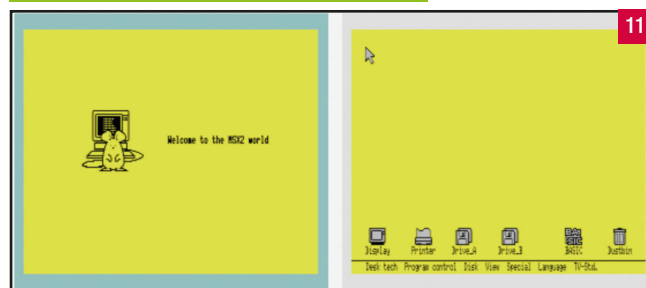
A part avec des appareils photo Polaroid, il était impossible en 1987 pour un particulier de produire des photos numériques et de les imprimer immédiatement en format A4 sur une imprimante. Les appareils photo-numériques n'existaient pas.

Grâce aux capacités de digitalisation du proces-



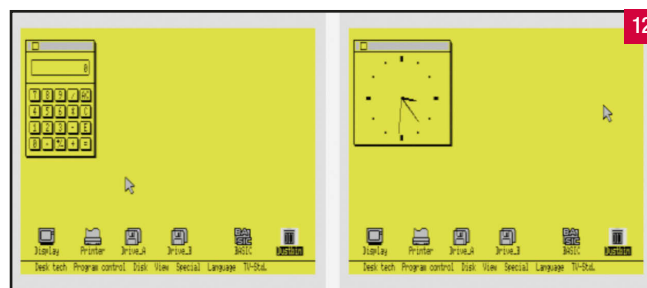
16

### Logiciel Hybrid :



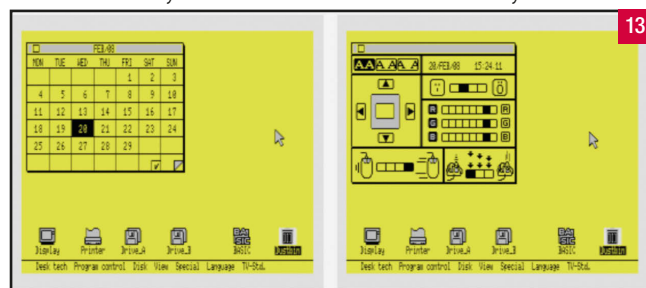
Intro Hybrid

Menu Hybrid



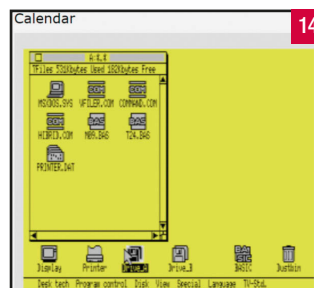
Calculatrice

Horloge



Calendrier

Panneau de configuration



Fichiers sur Disquette A :

seur graphique V9938 du MSX 2, fonctionnalité exploitée par le Philips NMS 8280, ou bien en utilisant une cartouche d'extension Sony HBI-V1, il était possible en quelques minutes de prendre une photo numérique en 256 x 212 pixels et en 256 couleurs à partir du MSX, d'un caméscope VHS et l'imprimer (en noir & blanc) sur une imprimante. [18][19]

## Des temps de chargement performants, de 1 seconde à 1min 30

Les jeux MSX étaient disponibles soit sous forme de K7, un support sous forme de bande magnétique, appelé aussi cassette audio que l'on utilisait dans les Walkman Sony, les Ipod de l'époque, soit sous forme de cartouche qui contenaient une ROM de capacité 16 ou 32 Ko. Les jeux en cartouche se chargeaient instantanément, comme sur votre XBOX aujourd'hui, alors que les jeux en K7... comment dire il fallait attendre 1min30 ou 3min pour charger votre jeu qui pesait 16 ou 32 Ko. En effet, La vitesse de chargement était de 1200 bauds (bit par seconde) soit 150 octets par seconde. Ecouter le son enregistré sur ces K7 dans son walkman était assez désagréable et strident. Il s'agissait d'une technologie similaire utilisée dans les FAX et les Modems apparus dans les années 95. Selon son lecteur de K7, il fallait trouver le bon niveau de volume pour que le jeu se charge, ou que sa sauvegarde soit lisible. Eh oui ! Les sauvegardes des parties des jeux de l'époque se faisaient sur K7 audio également !

C'était le cas du fameux jeu METAL GEAR... mais nous y reviendrons. Avec le MSX 2 les jeux étaient soit distribués sous forme de cartouches MegaRom d'une capacité de 128 à 512 Ko, soit sur disquette(s) 3,5 pouces d'une capacité de 360 ou 720 Ko. Certaines des cartouches MegaRom avaient parfois un logo SCC, ce qui signifiait qu'elles intégraient une puce sonore en plus de la ROM, ce qui apportait 5 voix supplémentaires d'une qualité supérieure à celles du processeur sonore du MSX. Des jeux comme Nemesis, King's Valley 2, F1 Spirit ou enfin SD Snatcher en faisaient partie. Les jeux sur disquettes étaient plus lents à charger que les ROMs ou MegaRom (quelques secondes), plus fragiles aussi, mais étaient bien pratiques comme support pour créer ou utiliser ses « copies de sauvegarde ». [20][21][22]

## Des jeux mythiques

Le principal contributeur de jeux sur MSX était la firme Konami, bien connue pour ses jeux d'arcades. Elle avait porté ses jeux sur MSX. Les plus connus sur MSX étaient Frogger, Contra, Gradius 2 (Nemesis), Nemesis 1,2 3, Parodius, Vampire Killer (Castlevania). Konami a surtout sorti des titres exclusifs et mythiques sur MSX tels que Penguin Adventure, Knightmare, King's Valley et SD Snatcher.

De nombreux autres éditeurs de jeux ont édité des jeux sur MSX, tels que Namco, Taito, HAL Laboratory, T&E Soft, Infogrames, et Compile.

**Penguin Adventure** est un jeu vidéo développé et édité par Konami sur l'ordinateur MSX en 1986. Il s'agit de la suite d'Antarctic Adventure. Il s'agit d'un jeu d'adresse en contre-la-montre.

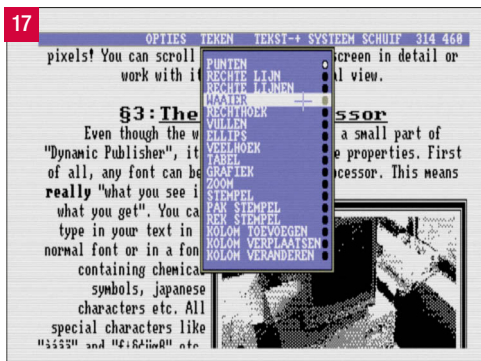
Le joueur dirige un manchot qui doit sauter au-dessus des crevasses et des ennemis (en l'occurrence des phoques qui sortent de crevasses dans le sol) pour atteindre les stations scientifiques internationales d'Antarctique qui marquent la fin de chaque niveau. Le joueur peut attraper des drapeaux qui augmenteront son score. C'est le premier jeu sur lequel ait travaillé le créateur Hideo Kojima, en qualité de directeur assistant. Ce dernier est avant tout connu pour sa série de jeux vidéo Metal Gear, vendue à plus de 49 millions d'exemplaires.

**Knightmare**, est un jeu vidéo conçu par Konami et publié en 1986 sous forme de cartouche pour le standard MSX. Dans ce jeu de tir à défilement vertical, avec un gameplay excellent, le héros Popolon se bat contre divers démons, dont Hudnos qui détient sa bien-aimée, Aphrodite.

**Vampire Killer** est un jeu vidéo d'action/aventure développé et édité par Konami, sorti en 1986 sur MSX 2. Sur le thème du Bien contre le Mal grâce au mythe du vampire : la famille Belmont affronte tous les cent ans le seigneur du mal Dracula, grâce à un fouet magique et un armement rudimentaire composé principalement d'un crucifix-boomerang, d'une hache, d'une dague de jet, d'une montre et d'une fiole d'eau bénite.

**King's Valley 2** est un jeu vidéo de type puzzle développé par Konami sorti en 2 versions, MSX et MSX 2 en 1988. Vous êtes dans la peau d'un aventurier qui doit résoudre différents tableaux tout en échappant à d'effrayantes momies.

**Metal Gear** est un jeu vidéo d'action-infiltration conçu par Hideo Kojima, développé et édité par Konami sur MSX 2 en 1987. Le jeu est



PAO avec Dynamic Publisher



Photo digitalisée et imprimée en 1987, j'étais bien jeune...



Carte de digitalisation Sony HBI-V1



Jeu en Cassette (K7) 16 à 32 Ko



Jeu en cartouche 16 à 512 Ko



Disquette 360 ou 720 Ko

adapté sur NES en 1987 et sur Commodore 64 et DOS en 1990. Considéré comme un précurseur du genre infiltration, le jeu marque les débuts de la série Metal Gear, l'une des plus appréciées et des plus anciennes de l'histoire du jeu vidéo. Une suite directe, Metal Gear 2: Solid Snake, est sortie en 1990 sur MSX 2 tandis qu'un épisode non-canonique, Snake's Revenge, sortait la même année sur NES. Ce n'est qu'en 1998 que Hideo Kojima fut connu du grand public, lorsqu'il sort le 3ième volet Metal Gear Solid sur PlayStation, grâce à la qualité du scénario et à une réalisation graphique et technique exemplaire qui a fait franchir un cap supplémentaire aux jeux vidéo sur console.

**Rollerball** est un jeu vidéo de flipper développé et édité par HAL Laboratory, sorti en 1984 sur MSX et porté en 1988 sur NES.

**Ping Pong** est un jeu vidéo de sport développé et édité par Konami. Vous jouez au Ping Pong à deux ou contre le MSX. De nombreux autres jeux de sports ont été édités par Konami dont Tennis, Golf, Baseball, Hyper Sport 1,2,3, et Hyper Olympic 1,2,3.

**Hyper Rally** est l'un des 3 jeux de courses de voitures développé et édité par Konami sur MSX, celui-ci est sorti en 1985. Les deux autres étant : Road Fighter (1985) et le dernier F1 Spirit (1987) [23]

On ne pourra pas tous les citer, mais voici une liste supplémentaire de quelques autres jeux MSX qui ont marqué cette console : Aleste, Aleste 2, Arkanoïd, Beamrider, Bubble Bobble, Circus Charlie, Comic Bakery, Eggerland Mystery, Galaga, The Goonies, Hole In One Professional, Hydride, Laydack, Mandragore, Pac-Man, Quarth, Red Lights Of Amsterdam, R.Type, Salamander, Sorcery, StarShip Rendez-vous special stage Michelle, River Raid, TNT, USAS.

## Les copies de sauvegardes

Même si Internet n'existait pas, il y avait des communautés MSX un peu partout, et des club utilisateurs qui échangeaient sur des utilisations avancées comme la modification de jeux pour avoir des vies illimitées, ou encore effectuer des copies de sauvegarde de leurs jeux préférés.

Les jeux en K7 pouvaient se copier comme on copiait des K7 de musique avec un double lecteur enregistreur de K7. Les jeux en cartouche qui étaient en ROM pouvaient être copiés avec quelques lignes d'assembleur et un loader en Basic était capable de charger le jeu en RAM. Des protections anti-piratage sont apparues dont le principe était de faire des modifications mémoire, sans effet sur une ROM originale, mais destructrice sur une copie chargée en RAM. Dans ce cas c'était écran noir....le jeu s'autodé-

truisait et cela demandait des compétences d'assembleur pour faire un fix. Sur le jeu de voitures « Hyper-Rally », la protection était sournoise, le jeu copié se lançait en apparence, mais la barre espace qui faisait office de pédale d'accélération était inopérante... Les jeux en disquette pouvaient se copier, mais pas depuis le MSX directement, mais plutôt depuis un Atari, qui avait des logiciels de copie avancés capables de reproduire des anomalies comme un trou laser. Le traitement de texte « Dynamic Publisher » qui était en disquette, pouvait se copier simplement par un copy \*.\* sur une disquette vierge qui avait été formatée de façon un peu originale. Il suffisait d'acheter des disquettes non pré-formatées (c'était le cas par défaut) de compter le nombre de piste pendant le formatage en tendant l'oreille contre son lecteur de disquettes, et d'éjecter la disquette au bout de 76 pistes au lieu d'attendre la fin des 80.

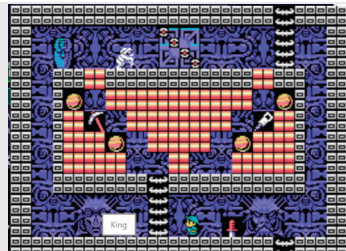
## Les langages de développement

Le langage de développement le plus commun sur MSX était le MSX BASIC qui était fourni en standard, mais il y avait aussi le langage Cobol, tous deux avaient été développés par Microsoft. Les développeurs avancés développaient directement en assembleur Z80. Le langage Logo

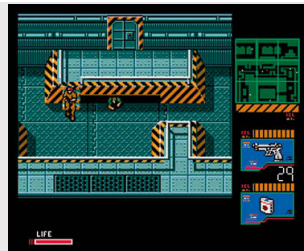
23



Nemesis (MSX, 1986)



King's Valley 2 (MSX, 1988)



Metal Gear 2 / Solid Snake (MSX2, 1990)



RollerBall (MSX, 1984)



Yie Ar Kung-Fu 2 (MSX, 1985)



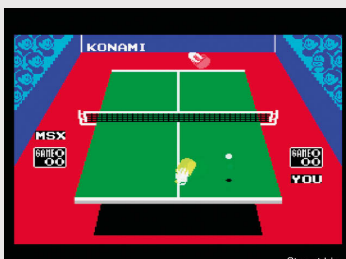
Knightmare (MSX, 1986)



Pinguin Adventure (MSX 2, 1986)



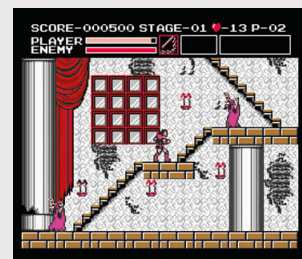
Hyper-Rally (MSX, 1985)



Konami Ping Pong (MSX, 1985)



SD Snatcher (MSX2, MSX2+ 1990)



Vampire Killer / Castlevania (MSX2, 1986)

était également présent, mais plutôt destiné à un usage pédagogique. Le MSX-DOS, le système d'exploitation du MSX était une variante du système d'exploitation CP/M, système d'exploitation créé en 1974 par Gary Kildall, fondateur de Digital Research et porté sur Z80. Plus tard Microsoft sortira le MS-DOS, lui aussi fortement inspiré du CP/M. Grâce à cet ADN un peu particulier, le MSX pouvait faire tourner DBASE 2, un SGBD destiné à faire partager des fichiers de données par plusieurs utilisateurs. [24]

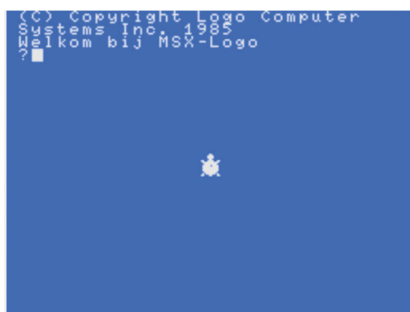
## Développement d'un jeu, à base de sprites

Voici un exemple de code développé en MSX-BASIC, tiré du livre MSX BASIC aux éditions PHILIPS par Albert Sicker, avec de légères adaptations.

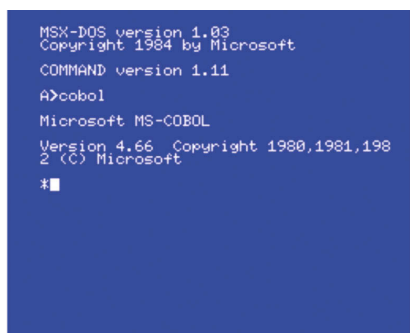
Un sprite est une figure que nous pouvons définir au moyen d'une matrice de points. On lui affecte un numéro et ensuite on peut gérer toutes sortes d'interactions, comme le déplacer sur l'écran, ou encore gérer la collision entre plusieurs sprites. Sans les sprites, on utiliserait des instructions PRINT ou PSET en MSX pour



DBase II



Language Logo



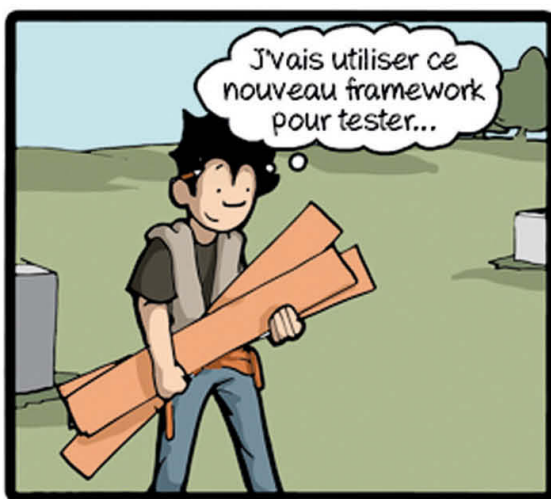
Language Cobol

## Caractéristiques techniques

Standard MSX	MSX	MSX 2	MSX 2+	MSX Turbo-R
Date de sortie	1983	1986	1988	1991
Processeur	Zilog Z-80A 3.58 Mhz			R800 à 7,16 MHz + Zilog Z-80A 3.58 Mhz
Son	PSG (AY-8910 1.815 Mhz)		PSG (AY-8910 1.815 Mhz) + MSX-MUSIC (Yamaha YM2413)	PCM 15.75 MHz + MSX-MUSIC (Yamaha YM2413)
Mémoire RAM	8 Ko à 64 Ko	64 Ko à 256 Ko RAM Disk (Memory Mapper)		256 Ko ou 512 Ko RAM Disk
Mémoire ROM	32 Ko contenant MSX BASIC 1.0	48 Ko contenant MSX BASIC 2.0	48 Ko contenant MSX BASIC 3.0	
Mémoire Vidéo	16 Ko	128 Ko ou 192 Ko	128 Ko ou 192 Ko	
Processeur Vidéo	VDP (TI 9918 ou 9928)	Yamaha V9938 (numérisation et incrustation possibles en 256 couleurs)	Yamaha V9958 (numérisation et incrustation possibles en 19268 couleurs)	
Vidéo	Screen 0 (mode texte 40 x 24 caractères, 2 couleurs sur 16) Screen 1 (mode texte 32 x 24 caractères, 16 couleurs) Screen 2 (mode graphique haute résolution 256 x 192 pixels, 16 couleurs) Screen 3 (mode graphique basse résolution 64 × 48 - 4×4 pixel blocks) 32 sprites avec détection de collision			
		Screen 9 (mode texte 80 x 24 caractères, 4 à 16 couleurs sur 512) Screen 5 (mode graphique 256 x 212pixels, 16 couleurs sur 512) Screen 6 (mode graphique 512 x 212 pixels, 4 couleurs sur 512) Screen 7 (mode graphique 512 x 212 pixels, 16 couleurs sur 512) Screen 8 (mode graphique 256 x 212 pixels, 256 couleurs) + modes entrelacés qui doublent le nombre de lignes à 424		
			Screen 10 (mode graph. 256 x 212 pixels, 16 coul sur 12499) Screen 11 (mode graph. 256 x 212 pixels, 16 coul sur 12499) Screen 12 (mode graph. 256 x 212 pixels, 19268 couleurs)	
Son	3 voies, 8 octaves + PCM 4 bits		3 voies, 8 octaves + PCM 4 bits + 9 canaux (FM)	3 voies, 8 octaves + PCM 4 bits + 9 canaux (FM)
Clavier	73 touches, 5 touches de fonction, flèches et 3 touches spécialisées			
Connecteur cartouche	50 contacts, 1 ou 2 connecteurs			
Connecteur d'extension	50 contacts, non obligatoire			
Port imprimante	Port Parallèle, centronics			
Port manette	1 ou 2			
Interface vidéo	RVP (Péritel) ou video-composite			
Port cassette	Un port cassette			
Langage	Msx Basic 1.0	Msx Basic 2.0	Msx Basic 3.0	Msx Basic 4.0 / 4.1
OS	Msx Basic 1.0	Msx-Dos 1.0		
Stockage		lecteur 3 1/2 360k/720ko simple/double face	lecteur 3 1/2 720ko double face	



# Les sides projects



CommitStrip.com



Une publication Nefer-IT, 57 rue de Gisors, 95300 Pontoise - [redaction@programmez.com](mailto:redaction@programmez.com)

Tél. : 09 86 73 61 08 - Directeur de la publication & Rédacteur en chef : François Tonic

Secrétaire de rédaction : Olivier Pavie

Ont collaboré à ce numéro : Y. Grandmontagne, S. Saurel

Nos experts techniques : M. Oulouhadjian, J-M Heitz, X. Flamant, D. Tizon, S. Harkousse, I. Leontiev, V. Kutsyk,

V. Thavonenkham, A. Blind, L. Piot, P. Bailly, O. Huber, K. Siou, C. Pichaud,

Couverture : Microsoft, Linux, © Talaj - Maquette : Pierre Sandré.

Publicité : PC Presse, Tél. : 01 74 70 16 30, Fax : 01 40 90 70 81 - [pub@programmez.com](mailto:pub@programmez.com).

Imprimeur : S.A. Corelio Nevada Printing, 30 allée de la recherche, 1070 Bruxelles, Belgique.

Marketing et promotion des ventes : Agence BOCONSEIL - Analyse Media Etude - Directeur : Otto BORSCHA [oborscha@boconseilame.fr](mailto:oborscha@boconseilame.fr)

Responsable titre : Terry MATTARD Téléphone : 09 67 32 09 34

Contacts : Rédacteur en chef : [ftonic@programmez.com](mailto:ftonic@programmez.com) - Rédaction : [redaction@programmez.com](mailto:redaction@programmez.com) - Webmaster : [webmaster@programmez.com](mailto:webmaster@programmez.com) -

Publicité : [benoit.gagnaire@programmez.com](mailto:benoit.gagnaire@programmez.com) - Evenements / agenda : [redaction@programmez.com](mailto:redaction@programmez.com)

Dépôt légal : à parution - Commission paritaire : 1220K78366 - ISSN : 1627-0908 - © NEFER-IT / Programmez, octobre 2017

Toute reproduction intégrale ou partielle est interdite sans accord des auteurs et du directeur de la publication.

**Abonnement** : Service Abonnements PROGRAMMEZ, 4 Rue de Mouchy, 60438 Noailles Cedex. - Tél. : 01 55 56 70 55 - [abonnements.programmez@groupe-gli.com](mailto:abonnements.programmez@groupe-gli.com) - Fax : 01 55 56 70 91 - du lundi au jeudi de 9h30 à 12h30 et de 13h30 à 17h00, le vendredi de 9h00 à 12h00 et de 14h00 à 16h30. **Tarifs abonnement** (magazine seul) : 1 an - 11 numéros France métropolitaine : 49 € - Etudiant : 39 € CEE et Suisse : 55,82 € - Algérie, Maroc, Tunisie : 59,89 € Canada : 68,36 € - Tom : 83,65 € - Dom : 66,82 € - Autres pays : nous consulter.  
**PDF** : 35 € (monde entier) souscription sur [www.programmez.com](http://www.programmez.com)



Sur abonnement ou en kiosque

# Le magazine des pros de l'IT



Mais aussi sur le web

L'INFORMATICIEN

JUSQU'AU 15 décembre

## OPÉRATION POUR 1 EURO DE PLUS

Pour bénéficier de cette offre exceptionnelle, il suffit de commander WINDEV Mobile 23 (ou WINDEV 23, ou WEBDEV 23) chez PC SOFT au tarif catalogue avant le 15 Décembre 2017. Pour 1 Euro de plus, vous recevrez alors le ou les magnifiques matériels que vous aurez choisis. Offre réservée aux sociétés, administrations, mairies, GIE et professions libérales, en France métropolitaine. L'offre s'applique sur le tarif catalogue uniquement. Voir tous les détails sur : **WWW.PCSOFT.FR** ou appelez-nous au **04.67.032.032**

Le Logiciel et le matériel peuvent être acquis séparément. Tarif du Logiciel au prix catalogue de 1.650 Euros HT (1.980,00 TTC). Merci de vous connecter au site [www.pcsoft.fr](http://www.pcsoft.fr) pour consulter la liste des prix des matériels. Tarifs modifiables sans préavis.

**Aucun  
abonnement  
à souscrire.**  
Compatible  
tous opérateurs

### CHOISISSEZ :

- iPhone X 64GB  
OU
- iPhone 8 256GB  
OU
- iPhone 8 Plus 64GB  
OU
- MacBook Air  
13,3" 128GB  
OU
- 2x iPad  
9,7" 128GB

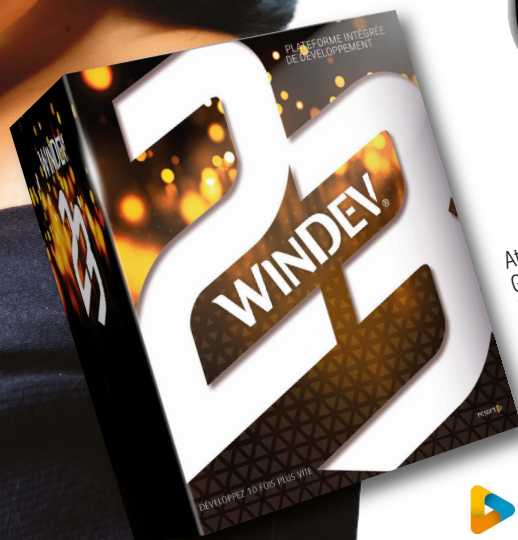
(Détails et autres  
matériels sur  
[www.pcsoft.fr](http://www.pcsoft.fr))

# COMMANDEZ WINDEV 23 OU WEBDEV 23 OU WINDEV MOBILE 23 ET RECEVEZ LE NOUVEL iPhone X

Choix de la couleur  
sur le site



## iPhone X



Atelier de  
Génie Logiciel  
Professionnel  
Cross-Plateformes

 [WWW.PCSOFT.FR](http://WWW.PCSOFT.FR)