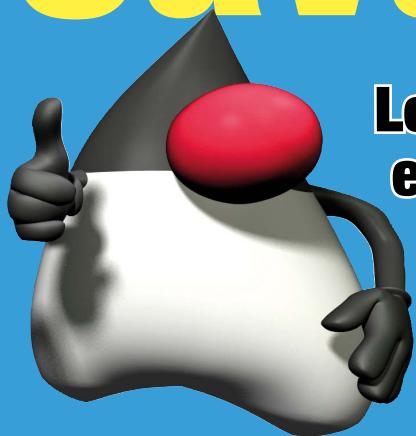


PROGRAMMEZ !

#208 - juin 2017

le magazine des développeurs

Java 9



Les bonnes et mauvaises surprises du nouveau Java

.NET
Core

.Net

devient universel avec .Net Core

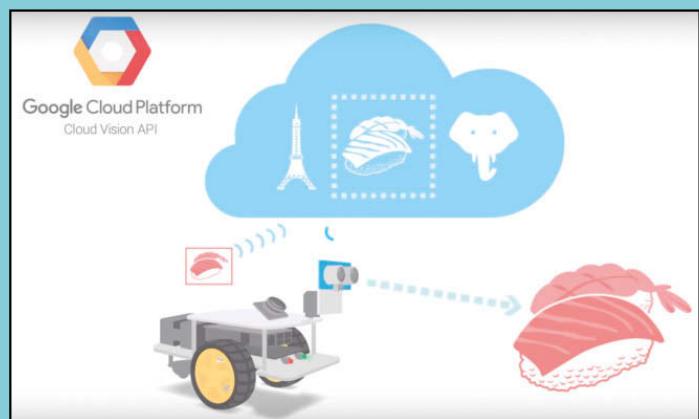


Résumés des conférences
Devoxx France
&
//BUILD 2017

Gulp

Votre boîte à outils
JavaScript à utiliser d'urgence

Tout savoir sur
Google Cloud Vision



Low code
- de codes pour aller + vite

M 04319 - 208 - F: 6,50 € - RD



SERVEURS DÉDIÉS **XEON®**

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

debian ubuntu CentOS Windows Server 2012

AVEC

ikoula
HÉBERGEUR CLOUD



POUR LES LECTEURS DE
PROGRAMMEZ*

OFFRE SPÉCIALE -60 %

À PARTIR DE

11,99€

~~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.

CHOISSISEZ VOTRE XEON®

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



ikoula
HÉBERGEUR CLOUD



/ikoula



@ikoula



sales@ikoula.com



01 84 01 02 50

Toi aussi, invente ta liste d'excuses pour les bugs, les crashes et quand l'utilisateur n'est pas content

- 1 Ça marche sur ma machine...
- 2 Vous êtiez où quand le programme a crashé ?
- 3 Pourquoi voulez-vous l'utiliser de cette manière ?
- 4 Vous ne pouvez pas utiliser cette version sur votre système...
- 5 Pouvez-vous vérifier s'il y a un virus sur votre système ?
- 6 Quelqu'un a (forcément) changé mon code...
- 7 Il fonctionne mais il n'a pas été testé...
- 8 Ce n'est pas le bon code...
- 9 Je ne peux pas tout tester!
- 10 C'est juste une malheureuse coïncidence, désolé.
- 11 Vous devez avoir une mauvaise version...
- 12 Je n'ai pas touché à mon code depuis plusieurs semaines...
- 13 Il doit y avoir des choses bizarres dans les données utilisées!
- 14 Cela doit être un problème matériel!
- 15 Comment est-ce possible?
- 16 Il fonctionnait très bien hier!
- 17 Je n'ai jamais eu ce problème avant!
- 18 C'est étrange...

Bonus : Vous avez appelé le support technique, veuillez patienter quelques minutes...

Liste non exhaustive.

Vous pouvez mixer les excuses.

Pensez à alterner les réponses de temps en temps.

François Tonic
ftonic@programmez.com

Agenda	Tableau de bord	Matériel
6	4	8

Reportage hackathon
11



Hololens
13

Chronique
16



ABONNEZ-VOUS !
9

VoIP & Magento partie 3
64

.Net Core
18



Java 9
25



Low Code
41



Résumés de la conférence
Devoxx 2017
32

Conférence //BUILD 2017
45



Automatisez-vous la vie
avec Gulp
54



Azure & architecture
serverless
57

Bloquer POP/IMAP
60



Blockchain
69

Spécial Google Cloud Vision
73



Complétez votre
collection !
47

Commodore 64
81

CommitStrip
82



Dans le prochain numéro !
Programmez ! #209, dès le 1er juillet 2017

Spécial été
Monter et programmer son robot !

Android
Découvrir le langage Kotlin

Performance
Comment gérer et surveiller les performances de son application ?

Apple a dépassé les 800 milliards \$ de capitalisation !

Microsoft évoque le futur des téléphones maison : oui mais pas sous la forme actuelle. Windows Mobile mort ou pas mort ?

Microsoft bis annonce son concurrent d'Amazon Echo pour l'autonome, avec un design que l'on aime ou pas. Pas encore de prix ni de SDK.

Amazon Echo arrive dans une version avec un écran. L'avenir des assistants est-il à l'écran ?

Java 9 est toujours prévu pour le 27 juillet prochain.

Affaire San Bernardino. Le débloque de l'iPhone aurait coûté + 900 000 \$

Tesla veut une voiture réellement autonome pour 2019.

Un robot policier K5 a été agressé et renversé par un homme ivre à Mountain View. Il existe un vide juridique sur ce genre d'agression.

ARM VA ARRIVER SUR PC

Les processeurs ARM sont déjà omniprésents sur les terminaux mobiles. Pas encore sur nos ordinateurs. Des solutions existent sur la partie serveur mais cela reste assez confidentiel. Mais les premiers PC arriveront vers la fin 2017, Qualcomm, fondeur bien connu, l'a annoncé il y a quelques semaines. Une version ARM de Windows 10 devra être installée. Mais il faudra voir comment les constructeurs vont réagir et s'ils sortiraient des modèles ARM. Car, il faut le système et les applications, et là, le chantier reste colossal. Microsoft avait subi un échec cinglant avec Windows RT en 2013, une version ARM pour tablettes. Bref, le succès n'est absolument pas garanti.

Par contre, les barebones ARM ne sont pas une chimère et aujourd'hui, on peut monter très facilement un serveur ou un cluster ARM, par exemple avec des Raspberry Pi ou des ODROID, sans aucune difficulté. Docker supporte officiellement depuis bientôt un an les machines ARM...

GOOGLE VA IMPOSER UN NOUVEAU LANGAGE DANS ANDROID

Ce fut sans aucune doute la grande annonce de la conférence Google I/O 2017 : un nouveau langage va arriver dans Android et ce, dès la version O. Il s'agit du langage Kotlin. Il n'est pas nouveau mais il va enfin pouvoir sortir de l'ombre. Il est développé par les équipes de JetBrains, éditeur de l'excellent IntelliJ. Il fonctionne avec une JVM, il est compatible avec Java et peut être compilé en JavaScript. Kotlin est orienté objet, avec un soupçon de fonctionnel et n'imposera pas de réécrire les applications car il s'interface avec Java et surtout, Google a prévu des outils de conversion.

Cependant, il ne faut pas

avoir d'illusions, mieux vaut un projet pur Kotlin pour en profiter totalement. Kotlin se rajoute donc à Java et à C++, sans les remplacer... pour le moment. Le site officiel du langage évoque bien entendu la comparaison avec Java :

<https://kotlinlang.org/docs/reference/comparison-to-java.html>

Le langage est open source sous licence Apache 2. Et les sources sont accessibles sur Github. Les créateurs du langage se sont inspirés de Java, C#, Scala et Groovy. Mais malgré plusieurs années de développement, Kotlin doit encore s'enrichir et prouver sa robustesse. Souvenez-vous de Swift d'Apple. Il a fallu



plus de 2 ans pour stabiliser le projet et proposer une version la plus complète possible et capable d'aller en production.

Pour le moment, Java n'est pas abandonné de la plateforme. Des fonctions Java 8 seront supportées mais Google prévient qu'à terme, Kotlin sera LE langage d'Android. Les relations avec Oracle n'ont pas aidé.

LES SERVEURS MAISON VONT-ILS TUER LES CONSTRUCTEURS DE SERVEURS TRADITIONNELS ?

Les conférences développeurs sont parfois l'occasion de découvrir des designs serveurs / d'infrastructures très intéressants. Google a dévoilé la nouvelle version de son monstrueux TPU, alias tensor processing unit. Il est taillé pour le machine learning et l'intelligence artificielle, très gros consommateurs de calculs. Les nouvelles unités affichent une puissance théorique de 180 teraflops ! Et chaque baie, TPU Pods, peut recevoir jusqu'à 64 TPU...

Il y a deux ans, Facebook publiait son propre design matériel pour les serveurs dédiés à l'IA. Le tout était mis sur le projet Open Compute. Microsoft fait la même chose. Apple travaille aussi sur son propre design serveur. Le mouvement Open Compute a incité les gros acteurs du marché à y participer. Les gros acteurs ne cherchent pas des machines "standards" mais des designs sur mesure adaptés à leurs besoins. Mais de là à dire que les constructeurs vont disparaître, il y a encore une énorme marge.

INDEX TIOBE

Pas de changement dans la popularité des recherches sur les langages en mai. Le trio de tête ne change pas. Par contre il y a du changement après : Python progresse, C# baisse. Un retour en force de l'assembleur dans les recherches.

Mai 2017	Mai 2016	Evolution	Langage	%	Evolution
1	1	=	Java	14.639%	-6.32%
2	2	=	C	7.002%	-6.22%
3	3	=	C++	4.751%	-1.95%
4	5	▲	Python	3.548%	-0.24%
5	4	▼	C#	3.457%	-1.02%
6	10	▲	VB.Net	3.391%	+1.07%
7	7	=	JavaScript	3.071%	+0.73%
8	12	▲	Assembleur	2.859%	+0.98%
9	6	▼	PHP	2.693%	-0.30%
10	9	▼	Perl	2.602%	+0.28%

OPÉRATION POUR 1 EURO DE PLUS

Pour bénéficier de cette offre exceptionnelle, il suffit de commander WINDEV Mobile 22 (ou WINDEV 22, ou WEBDEV 22) chez PC SOFT au tarif catalogue avant le 30 juin 2017. Pour 1 Euro de plus, vous recevez 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 ouappelez-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 pour consulter la liste des prix des matériels. Tarifs modifiables sans préavis.



iPhone 7 Plus

CHOISISSEZ :

- iPhone 7 Plus 128GB
ou
- iPhone 7 256GB
ou
- **2x iPad New 9,7" 128GB**
ou
- **MacBook Air 13,3" 128GB**

(Détails et autres matériels sur www.pcsoft.fr)

JUSQU'AU 30 JUIN
COMMANDÉZ
WINDEV MOBILE 22
OU WEBDEV 22 OU WINDEV 22
ET RECEVEZ
LE NOUVEL
iPhone 7

Choix de la couleur
sur le site



Apple®, iPhone®, iPad®, iPad Air®, iPad Mini™ sont des marques déposées de la société Apple. Apple n'est pas un organisateur ou un sponsor de cette opération.

Tél Paris: 01 48 01 48 88
Tél Montpellier: 04 67 032 032

 WWW.PCSOFT.FR

Venez rencontrer Programmez !

à la DevFest Lille et à Maker Faire Paris

9 JUIN LES 10 ET 11 JUIN

Juin

L'APRÈS-MIDI DU DOMAIN-DRIVEN DESIGN

7 juin / Issy-les-Moulineaux

Deux ans après le succès de l'après-midi du Software Craftsmanship, nous reviendrons le 7 juin prochain chez Microsoft France avec la même formule boostée en live-coding mais pour vous proposer cette fois une immersion dans le DDD que vous n'êtes pas prêt d'oublier. Fini le code métier noyauté par de la plomberie technique ou du code « générique » sans aucune saveur métier. Que vous ayez des connaissances ou non à propos du Domain Driven Design, cette conférence vous permettra de repartir avec les idées claires, mais surtout pleines d'astuces concrètes pour pouvoir les appliquer dès le lendemain sur vos projets (notamment « legacy »). Etudiants, développeurs et architectes sont les bienvenus. Inscription sur le Meetup DDD Paris : <http://tinyurl.com/l92xtz>



BEST OF WEB 2017

les 8 & 9 juin / Paris

Vous avez raté les meetups phares de Paris ? Une exceptionnelle séance de rattrapage est possible durant les deux journées du Best of Web. On va y parler Angular, JS, node, Web Composants, CSS, Meteor, React, etc. La journée du 8 est consacrée aux formations et le 9 sera uniquement consacré à des conférences et sessions techniques. Que du super lourd ! Inscription : <http://bestofweb.paris>

MEETUP SOFTWARE CRAFT-MANSHIP

29 juin

Organisé par Cellenza, ce meetup sera consacré au craftsmanship et aura pour inviter spécial, Sandro Mancuso.

Inscription : <https://www.eventbrite.fr/e/billets-craft-aftertech-by-cellenza-with-sandro-mancuso-34721647382>

DEVFEST LILLE 2017

9 juin / Lille

Le Google Developer Group de Lille (@GDGLille) organise la première édition du DevFest Lille. Au programme, des conférences et des codelabs autour des technologies que nous aimons tous : Web, Mobilité, Cloud, IoT, ... Les sessions donnent envie : IoT, JS, HTML5, mobilité, DevOps, serverless, micro-services, langage Go, etc. Commitstrip sera présent. Rendez-vous sur <https://devfest.gdglille.org/>

MAKER FAIRE PARIS 2017

du 9 au 11 juin / Paris

Cette nouvelle édition du Maker Faire se déplace à la Villette et se tiendra en dehors de la foire de Paris comme ce fut le cas les années passées. Comme chaque année, ce sera l'occasion de découvrir des centaines de makers et de projets divers et variés. Un événement à ne pas rater !

PARIS CONTAINER DAY

13 juin / Paris

2e édition de la conférence dédiée aux conteneurs et aux bonnes pratiques. Le principal thème de cette année sera « les conteneurs en production ». De nombreuses sessions comme chaque année ! Informations : paris-container-day.fr

SESSIONS ORGANISÉES PAR ZENIKA

Matinale Bien choisir son framework JS en 2017, le 20 juin chez Zenika Paris.

Inscription à l'ensemble des événements sur <https://zenika.com/agenda/>

CONFÉRENCE PRESTASHOP DAY

15 juin à Paris

De 9h à 19h, plus de 40 Conférences, Ateliers, Masterclass, remise des PrestaShop Awards... s'enchaîneront et se croiseront pour offrir à chaque visiteur la possibilité de créer un programme sur-mesure tout en réservant du temps à l'échange. Quelques temps forts :

- **09h30** Discours inaugural de Bruno Lévéque, fondateur de PrestaShop
- **09h45** Keynote de Corinne Lejbowicz, CEO de PrestaShop, sur les dernières annonces et actualités de l'entreprise.
- **10h15** Innovation et Scalabilité, conférence animée par le CEO de Whyers, jus de cerveau des meilleurs start-upers, avec 4 entreprises comme Stripe et Stuart, dont la technologie innovante peut accélérer le déploiement et la réussite d'un business en ligne.
- **11h00** Success Stories, les secrets des start-ups à la croissance fulgurante, avec notamment les co-fondateurs de Bonne Gueule, « mode homme et conseils en style masculin » et de M. Moustache, « collection de souliers pour Monsieur et Madame ».

Cette année les visiteurs auront l'occasion d'échanger avec les intervenants après les conférences dans l'espace prévu à cet effet, l'espace « Share ». Inscription : prestashopday.com



PRÉSENTE

CLOUDIKOULAONE



 Le succès est votre prochaine destination

MIAMI SINGAPOUR PARIS
AMSTERDAM FRANCFORT --

CLOUDIKOULAONE 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

 sales@ikoula.com

 01 84 01 02 50

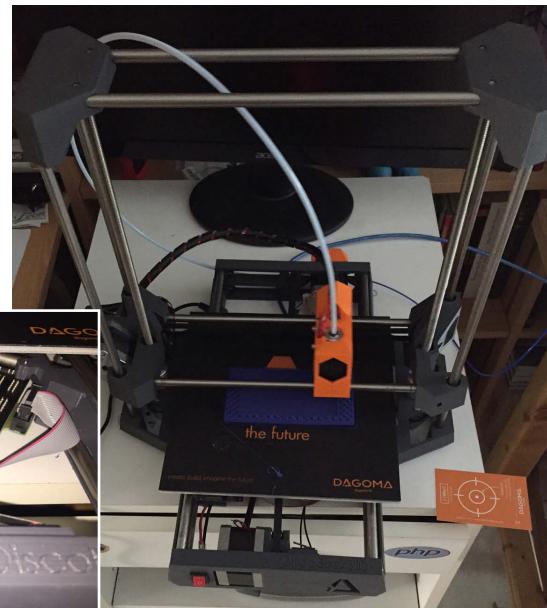

ikoula
HÉBERGEUR CLOUD

Impression 3D avec la DiscoEasy200

Dagoma propose plusieurs modèles d'imprimantes 3D dont la DiscoEasy200. Le modèle n'est pas nouveau mais après une déconvenue, un modèle Reprap de type Delta, à monter (courroies livrées trop courtes, impossibilité de calibrer un des moteurs), nous avons opté pour une autre reprap mais cette fois-ci, clé en main : donc toute montée. Vous payerez 100€ plus cher que la version en kit (299€).

L'imprimante est très simple : un socle, le plateau, et les supports verticaux pour la tête d'impression. Simple mais efficace. La technologie d'impression est dite par couche successive. Là encore, du classique. Quand on choisit une imprimante 3D, il faut se méfier du volume d'impression disponible. Ici, nous avons 200 x 200 x 200 mm. Espace correct pour imprimer des pièces de tailles relativement importantes. L'autre intérêt que nous voyons à ce volume, et aux concurrentes directes : c'est l'usage de pièces standards du marché. Il est donc facile de changer la buse d'impression ou les moteurs. Si un élément du chassis casse, vous pouvez l'imprimer vous-même, les fichiers 3D sont disponibles.

Nous aimons beaucoup l'encombrement de l'imprimante et son poids (env. 5 kg). La partie logicielle plaira aux débutants et aux utilisateurs qui ne veulent pas s'embêter. La configuration de base prend 5 minutes avec l'application dédiée. Et l'outil d'impression est lui aussi plutôt



intuitif et on ne s'occupe de rien ou presque. On apprécie aussi l'usage de bobines de filaments non propriétaires. Certains constructeurs obligent à acheter uniquement chez eux à cause du support non standard. Là, aucun souci : on choisit la marque que l'on veut, et la couleur.

Après quelques semaines d'utilisation, nous n'avons pas grand chose à redire : la qualité de l'impression est correcte. On peut choisir entre plusieurs niveaux de remplissage de l'objet et la qualité des couches (différentes dans l'épaisseur). Chaque option impacte, parfois énormément, le temps d'impression. Sur la vitesse, pas de miracle, l'impression 3D n'est pas quelque chose de rapide. Et comme souvent, vous aurez régulièrement des impressions ratées pour x raisons : blocage, l'objet qui se détache du plateau, buse trop pleine, etc. Dagoma devrait tout de même améliorer plusieurs points ergonomiques :

- Une connexion du câble USB très peu pratique ;

- Un accès au lecteur SD peu pratique ;
- L'entrée du filament située à l'arrière, ce qui n'est pas toujours le plus simple ;
- Absence de toute fixation de la bobine sur le châssis.

D'autre part, pour le prix, l'absence d'un écran de contrôle est regrettable quand on fait une impression directement depuis une carte SD. Pour le même prix, d'autres modèles proposent l'écran par défaut. L'écran peut être rajouté (49€).

La spatule est pratique mais elle n'est pas assez rigide quand l'objet est bien collé au plateau. Dommage aussi que le Buildtak, la couche que l'on met sur le plateau, ne soit pas livré en standard. Pour le prix c'est un peu mesquin. Heureusement, qu'un scotch 3M Bleu est posé sur le plateau mais il faudra rapidement le changer. Site : <https://dagoma.fr>

Notre note globale : 14/20

pcDUINO : UNE PLATEFORME ARDUINO HORS NORME

Vous aimez Arduino pour vos montages mais vous rêvez d'une carte beaucoup plus puissante. Les cartes pcDUINO fusionnent l'Arduino et une carte PC ! Faut-il essayer ? Nous avons opté pour la version pcDUINO 3B, une version classique de la plateforme. Il existe une dizaine de déclinaisons, la plupart ayant les mêmes dimensions. La carte apparaît un peu massive et reprend grossièrement la taille de l'excelente Pine64.

La fiche technique est plutôt alléchante : Compatible Arduino, Linux et Android, Support Java, C, C++, GPIO Arduino, Ethernet 10/100, Wifi, Port SATA, USB,

HDMI 1.4, SoC AllWinner A20 ARM Cortex A7, GPU Mali 400 Dual Core, 1 Go de mémoire vive, 4 Go de stockage interne + lecteur microSD. Une distribution Linux est par défaut installée sur la carte, facilitant le démarrage sur la plateforme. Bien entendu, vous pouvez utiliser une autre version en l'installant sur une microSD. L'unique port USB devient vite un problème. Pour le prix, deux ports n'auraient pas été un luxe. La seule solution est d'utiliser un hub USB. La pcDUINO est une alternative aux cartes mini / nano-PC du marché et pourrait se comparer aux Raspberry Pi,

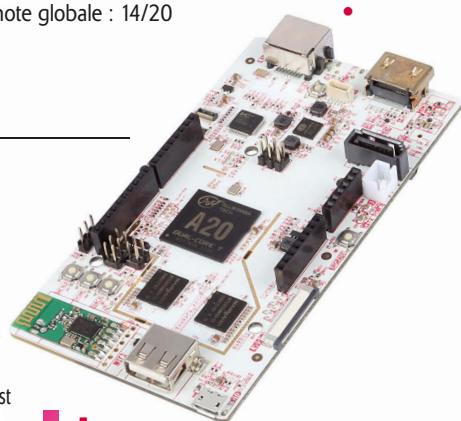
C.H.I.P. et autres ODROID et Pine64. Mais attention à ne pas trop en demander ! Les ressources matérielles sont intéressantes (CPU, GPU, stockage...) mais l'unique port USB est franchement mesquin surtout pour une carte que l'on trouve entre 60 et 70€ en France. C'est cher et pour ce prix nous avons l'excellente ODROID-C2 et pour bien moins cher un C.H.I.P. ou une Pine64. La documentation est comme souvent pour ces cartes le point faible et ce n'est pas la plateforme la plus active en France. La compatibilité Arduino est un plus si vous en faites.

Les +

- Compatibilité Arduino
- Support Android
- La diversité des usages
- Ressources matérielles

Les -

- Tarif
- Dimensions
- 1 seul port USB
- Documentation



Abonnez-vous à **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

Option :
accès aux archives **10€**

Nos offres spéciales été 2017

1 an **59€**

11 numéros + 1 vidéo ENI au choix :

- Big Data avec Hadoop
- Framework Spring

(Valeur de la vidéo de 29,99 à 59,99 €)



2 ans **89€**

22 numéros + 1 vidéo ENI au choix :

- Big Data avec Hadoop
- Framework Spring

(Valeur de la vidéo de 29,99 à 59,99 €)

Offre limitée à la France métropolitaine

Toutes nos offres sur www.programmez.com



Oui, je m'abonne

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 : Big Data avec Hadoop

Vidéo : Framework Spring

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

Le prix **Bernard Novelli**, un concours pour les développeurs en herbe !

• Par Martine Brilleaud et Thierry Garcia

Baptisé en hommage à un pionnier dans la création de jeux, le prix Bernard Novelli récompense un jeune programmeur pour un projet en rapport avec le jeu et les mathématiques. Il est attribué chaque année durant la cérémonie des Trophées Tangente au palais du Luxembourg.

L'objectif du prix Bernard Novelli est de récompenser le travail d'un jeune développeur en herbe. Le jeu présenté doit être en rapport avec les mathématiques, que ce soit dans ses règles ou sa conception. Le candidat s'inscrit sur le site www.tropheestangente.com avant le 15 juillet. Il a alors jusqu'au 30 septembre pour finaliser son projet avant d'envoyer un dossier complet. Beaucoup de jeunes candidats choisissent d'approfondir un projet commencé durant l'année scolaire au lycée pendant l'enseignement de terminale d'ISN (Informatique et science du numérique ; cet enseignement de spécialité de Terminale S est proposé dans de nombreux lycées depuis la rentrée 2012). Mais certains sont juste des passionnés qui apprennent la programmation en

autodidactes, comme ce collégien de 13 ans qui a épâté le jury en présentant un jeu réalisé en utilisant la plateforme de développement de jeu « Unity » et un de ses langages de programmation intégrés qu'il avait appris tout seul.

2017 : le concours est ouvert aux étudiants

Une nouveauté pour la session 2017 est l'ouverture du concours aux étudiants jusqu'au niveau L3. Néanmoins la candidature reste individuelle et si le projet présenté est issu d'un travail de groupe, il doit très clairement identifier la contribution du candidat.

Les candidats envoient la description de leur jeu avec toutes les règles et le développement dans le langage de leur choix.

Le jury, constitué de membres représentant les partenaires* valide les dossiers et se réunit en octobre pour désigner le vainqueur.

Les principaux critères sont les suivants :

- Originalité et créativité du concept ;
- « jouabilité » et plaisir des utilisateurs ;

- Qualités techniques (algorithmique, concepts mathématiques et code) ;
- Contenu susceptible d'être amélioré, complété et adapté vers une version exploitable.

Le vainqueur reçoit la dernière calculatrice formelle CASIO et une œuvre d'art mathématiques. Mais surtout la possibilité de voir son projet développé par un professionnel et mis à disposition sur des sites de téléchargement.

Le palmarès des années antérieures, le règlement complet et la procédure d'inscription sont disponibles sur le site www.tropheestangente.com.

* Les partenaires du Prix Novelli

Tangente, magazine de mathématiques, organisateur du concours,

Magma Mobile, société éditrice de jeux sur téléphone mobile,

CASIO, fabricant de calculatrices,

SIF (Société Informatique de France), société savante d'informatique,

Programmez !, votre magazine

COLOR TAQUIN, UN JEU ADDICTIF

En 2015, le jury a choisi de récompenser le lycéen grenoblois Romain Guetta pour son jeu Elmer's Game élaboré dans le cadre de l'option ISN de terminale S. Actuellement en cours de développement, le jeu, rebaptisé « color taquin », semble très prometteur sur téléphone. Le principe ? Reconstituer une grille colorée dans le temps le plus bref possible, en utilisant une règle proche du fameux « taquin » pour déplacer les cases. Le jury a remarqué la sophistication de l'algorithme utilisé, le nombre de lignes de code l'extensibilité du jeu et le caractère « addictif » du jeu.



LA PLANÈTE DES SIGNES VAINQUEUR EN 2014

Des + et des -, présenté par Mathieu Roget, est un jeu de grille pivotante dans lequel des billes décorées d'un signe arithmétique (plus ou moins) tombent les unes sur les autres. La grille du jeu est un carré de 4 x 4 cases pouvant accueillir des symboles + ou -. Le joueur peut faire basculer les symboles en pivotant la grille vers la gauche ou la droite, les blocs tombant par gravité. Lorsque deux blocs - s'entrechoquent, ils se transforment en blocs + (respectant ainsi la fameuse règle des signes de nos

cours de maths) et lorsqu'un bloc + tombe au sol, il disparaît. Le but est de vider la grille, sachant que des blocs + et - apparaissent à chaque pivotement de grille.

Le jury a apprécié l'originalité et a remarqué les difficultés algorithmiques qui ont été implémentées (rotation, pesanteur, collision), la qualité et l'organisation du code, les multiples possibilités d'extensibilité et la possible



adaptation en jeu sur smartphone. Le jeu, adapté par la société Argine, est disponible en téléchargement gratuit sous le nom « La planète des signes » sur Google Play.

L'équipe d'Alcatel-Lucent Enterprise (ALE) raconte sa participation au Hacking Health Camp 2017



Du 17 au 19 mars, l'innovation en santé était au centre de toutes les attentions à Strasbourg avec la tenue du Hacking Health Camp, le plus grand Hackathon Européen dédié à la Santé. Pendant trois jours, l'évènement international vise à innover et créer des outils numériques à destination du domaine médical. Pour ce faire, médecins, développeurs, patients, hackers et entrepreneurs conçoivent des prototypes tout en échangeant leurs idées sur la santé de demain.

Alcatel-Lucent Enterprise (ALE) est un partenaire historique des établissements de santé en France et dans le monde. L'objectif premier de la participation d'ALE à cette nouvelle édition est de renforcer la capacité d'innovation de l'entreprise et sa connaissance sectorielle. En ouvrant au maximum la participation au Hackathon, aux ingénieurs et développeurs de l'entreprise, ALE leur offre l'occasion d'acquérir des pratiques d'open innovation par le « faire ». La contrainte du temps limité inhérente au Hackathon est un moteur fabuleux pour adopter une posture d'entrepreneur innovant.

14 collaborateurs d'ALE ont participé, sur la base du volontariat, à cette fabuleuse aventure humaine. Parmi eux, 11 développeurs et designers graphiques spécialisés en langages Java, Javascript, C, C++, AngularJS, NodeJS, plateforme Android et environnement Cloud.

Les participants ont choisi de s'investir sur trois projets :

- **PôleBook**, une application en code natif, de type réseau social, disponible sur smartphone ou sur le Web. Un outil de communication et de collaboration pour les soignants, sans données patients et en intégrant les règles de l'art du privacy by design, projet porté par le pôle gynécologie obstétrique des Hôpitaux Universitaires de Strasbourg (ce projet a remporté le prix décerné par l'incubateur La Java-Ness) ;

- **Precovery**, une application permettant une meilleure préparation des patients à une opération en ambulatoire, de façon à réduire le taux de complications ultérieures, projet porté par un chirurgien en gastro-entérologie des Hôpitaux Universitaires de Strasbourg ;
- **E.P.J. 3D**, défi autour des technologies 3D, scanners, modélisation et impression d'esthétiques de prothèses.

2 jours & 2 nuits pour réfléchir, coder, et produire ensemble les outils santé de demain

Le premier jour se divise en deux parties, avec une partie faite de conférences inspirantes et d'ateliers de formation sur les thématiques clefs de la e-Santé et une autre correspondant au lancement du Hackathon. La journée qui suit est consacrée au développement des applications ou prototypes, en équipe. Le troisième et dernier jour est dédié à la présentation des prototypes devant le jury.

Equipe PôleBook

Trois volontaires d'ALE – dont 1 développeur, une responsable marketing, une responsable de business développement se sont associés au projet PôleBook, porté par des médecins et une cadre de santé. Un chef de projet venant du Conseil de l'Europe et deux étudiantes en fin d'étude de design du Lycée Le Corbusier de Strasbourg se sont également associés à l'équipe. Une équipe équilibrée et diversifiée en

types de profils, expériences, et parité hommes-femmes (certainement une recette du succès !).

Plusieurs méthodologies sont proposées et retenues pour permettre à ce groupe nouvellement constitué de travailler de manière optimisée :

- Méthode de travail de groupe intégrant les fondamentaux de l'intelligence collective, permettant à chacun d'exprimer son savoir faire et de partager son expérience ;
- Conseils pour une présentation d'impact en public ;
- Et surtout l'approche Lean Startup permettant de gagner en efficacité pendant les 50 heures du Hackathon : compréhension des problématiques vécues par le personnel soignant, définition de 7 histoires utilisateur (user stories), conception de l'arborescence de l'application, et codage d'un livrable viable (Minimum Viable Product ou MVP) pour être présenté au jury à la fin du weekend.

Benoit Trinité, chef de produit Rainbow(1) CPAas au sein d'ALE témoigne : « La constitution des équipes est allée très vite : nous étions deux développeurs sur ce projet, Massimo Pibiri et moi-même. Nous avons décidé de faire équipe au sein du projet PôleBook. Ne pas se connaître et travailler ensemble : un vrai challenge ! Nous avons mis en place nos objectifs lors du dîner. C'est à 3h du matin que les décisions finales ont été tranchées ainsi que les

choix stratégiques pour le prototype. A ce stade, nous avons une bonne maîtrise de la répartition des tâches, les designers ont bien avancé sur les éléments graphiques de leur côté ». Pour les deux développeurs, il s'agit de choisir une orientation technique à la croisée des compétences et du projet à réaliser, ce qui s'avère complexe. Après de nombreuses recherches et tentatives, le couperet tombe vers 3h du matin, il semble impossible de combiner les techniques qui constituent le bagage de chacun. « Nous décidons alors de miser sur les compétences de Massimo car plus stratégiques pour le prototype. Il se concentre sur l'application pour smartphone, intégrant les éléments graphiques, alors que je développe le côté serveur d'un système de messagerie basique ».

« Et c'est alors que la seconde difficulté technique se présente à nous et j'étais loin d'imaginer que cela pourrait arriver, surtout lors d'un Hackathon ! Une contrainte technique en tant qu'utilisateurs invités du réseau de la Faculté de Médecine, rend impossible une communication directe entre nos deux utilisateurs... Au pire, on ne démontrera que l'aspect « expérience utilisateur » de l'application et pas le fonctionnel. ... ».

19h : stress : toujours pas de plan B.

« Un de mes collègues me propose d'héberger mon développement dans le Cloud, qui lui est accessible depuis nos ordinateurs. A 21h, j'ai fini de déménager tout mon développement. » L'objectif est maintenant pour l'équipe de se concentrer sur le pitch final de 3 minutes. Fin du projet à 1h du matin. Dimanche, 8h : « on ne pourra pas intégrer tous les écrans de la démo dans l'application. On va donc subtilement alterner écran intégré dans l'application et écrans non encore intégrés lors de la démo. Vers midi, nous nous entraînons au pitch en conditions réelles dans le grand amphithéâtre ; occasion pour nous de prendre nos marques, vérifier le matériel et d'avoir le retour d'un des membres du jury».



La présentation finale du projet PôleBook est alors couronnée de succès et remporte le prix Innovation numérique décerné par La Java-Ness, incubateur de start-ups ! Il a obtenu du jury des notes de 19/20 et 16/20 attribuées respectivement par le Dr Soler, Directeur Scientifique de l'IRCAD(2) et Franck Ohrel, fondateur de start-up.

Equipe Precovery

10 développeurs et designers d'ALE se sont associés au défi proposé par Cristians Gonzalez, médecin-chercheur à l'IHU de Strasbourg.

Emmanuel Helbert, en charge de l'Innovation chez ALE, raconte :

« Au vu de la préparation amont du défi, nous avions décidé d'implémenter un simulateur de BOT avec notre plateforme Rainbow(1) comme moteur de notification. Cela permettait de s'affranchir de la problématique d'un serveur local. En plus du BOT, nous voulions montrer un dashboard dynamique qui permet au chirurgien de suivre les progrès du patient en fonction de ses réponses au BOT. Une page Web était la solution immédiate la plus simple à la vue des compétences des personnes présentes. Enfin, côté patient, il fallait une application client permettant d'afficher les questions du BOT et de saisir les réponses de celui-ci. Là encore, une page Web faisait très bien l'affaire parce que pouvant être affichée lors de la démo à la fois sur l'écran d'un de nos postes de bureau et sur l'écran d'un smartphone ou d'un PC. Un GitHub a très rapidement été créé pour le partage du code et les choses ont avancé très rapidement. Il fallait un évènement déclencheur du scénario

et on s'est demandé comment intégrer l'étape d'authentification sur l'application pour un patient. Stéphane s'y est mis toute la nuit du vendredi au samedi pour développer une authentification par lecture de code 2D (QR Code) en utilisant un plugin Cordova dans un projet Ionic. Finalement, samedi fin d'après-midi, la majeure partie du scénario était fonctionnelle sans problèmes majeurs. Il s'agissait alors de projeter le scénario du pitch sur les différents éléments graphiques que montrait l'application. Cela nous a permis d'adapter certains écrans au discours. Fin de matinée on se retrouve dans l'amphi pour tester le pitch en conditions réelles. On se rend vite compte à l'issue de cette session à blanc qu'il y a un problème de visibilité de la démo : la lecture du QR code prend trop de temps sur les 3 minutes imparties, il y a trop d'écrans à diffuser pour seulement deux sources vidéos. Pour le pitch final, la démonstration a été restreinte au partage d'un écran de smartphone au travers d'une application de mirroring ».

Une fabuleuse aventure humaine qui se prolongera au sein du département R&D d'Alcatel-Lucent Enterprise

Au-delà du succès du projet PôleBook au Hackathon et de la formidable aventure humaine, ALE souhaite poursuivre le projet et en faire une réalité en accord avec les Hôpitaux Universitaires de Strasbourg. Pour les deux équipes d'ALE, le retour des coachs est très précieux pour co-construire un cas d'usage pertinent qui recevra l'adhésion des patients et des soignants. Toutes les équipes ALE sont prêtes à s'investir à nouveau en 2018 pour continuer à apprendre des professionnels de santé et acteurs de l'écosystème du Hackathon et contribuer à optimiser les parcours de soin !

(1) : Alcatel-Lucent Rainbow est une plate-forme de gestion des relations en mode Cloud qui relie les employés, leurs contacts professionnels et les systèmes. Rainbow permet aux employés de mieux collaborer et d'être plus productifs en s'affranchissant des frontières de l'entreprise.

(2) : IRCAD : Institut de Recherche contre les Cancers de l'Appareil Digestif

HoloLens :

La Réalité Mixte au service de l'Entreprise



Jean Milleron,
Manager Software Engineering

- Guillaume Pugnet,
Senior Consultant Software Engineering
- Cédric Gouge,
Group Manager Software Engineering

 **avanade**

Les individus de cette nouvelle génération sont nos collaborateurs d'aujourd'hui et de demain. Bien au fait de ces technologies, ils attendent de l'entreprise qu'elle les adopte également, dans des scénarios qu'ils identifient souvent eux-mêmes, devançant l'entreprise, confirmant ainsi la fin d'une époque lointaine où toutes les technologies de pointe étaient accessibles par l'entreprise avant de l'être des particuliers. Les sociétés d'aujourd'hui doivent également voir l'intérêt de cette technologie, et son usage doit évidemment satisfaire des objectifs de rentabilité au sens large. Il pourra s'agir d'un gain direct de par l'amélioration de processus ciblés, ou de gains indirects en satisfaisant la nécessité croissante d'une image de modernité exigée par le grand public et par les collaborateurs existants ou futurs. La proposition Microsoft satisfait cette dualité de modernité et de rentabilité en proposant son appareil, l'HoloLens, sorti en février 2016 et disponible en France depuis Octobre 2016. Onéreux, l'appareil en est tout de même bien connu du grand public : l'effet WOW garanti à sa première utilisation, et un marketing ciblé, ont permis à Microsoft d'obtenir une visibilité croissante et méritée. D'un autre côté, l'appareil est ciblé pour l'entreprise ; HoloLens donne naissance à des cas d'usage jusqu'alors inédits ou inaccessibles, en redéfinissant l'expérience utilisateur par l'usage de la Réalité Mixte.

La Réalité Mixte : Sens et perception

« Rien ne vient à nous que falsifié et altéré par nos sens. »

Montaigne, *Essais*.

Avec plus de 6 millions d'appareils vendus, l'année 2016 – marquée par la sortie de la version finale de l'Oculus Rift de Facebook en Mars, puis du HTC Vive en Avril, et, enfin, par celle du Playstation VR en Octobre – a été l'année où la Réalité Virtuelle s'est rendue disponible au grand public. Conforme aux prédictions, la Réalité Virtuelle continue sa montée dans sa phase « Hype », et devient de plus en plus connue de la population. Elle deviendra, dans un avenir inévitable, une technologie « banale », « évidente » pour la nouvelle génération.



La **Réalité Virtuelle** (VR) propose d'immerger ses utilisateurs dans un environnement virtuel. L'immersion doit être la plus complète, l'utilisateur doit se sentir ailleurs. L'outil de réalité virtuelle permet de tromper ces deux organes de perception humains que sont la vue et l'ouïe, à l'aide de machines puissantes, auxquelles l'utilisateur est relié par un fil.

La **Réalité Augmentée** (AR), quant à elle, fournit de l'information. Consciente de son environnement, elle propose à son utilisateur des marches à suivre, des informations de trajet, adaptées à son orientation, de la superposition d'objets, de Pokémons, le tout en général à l'aide de l'outil mobile dont tout le monde dispose, le smartphone. L'humain continue à percevoir la réalité physique, et peut faire confiance à ses sens, la Réalité Augmentée fait office de guide, d'aide.

La **Réalité Mixte** (MR) combine l'immersion de la Réalité Virtuelle avec la superposition d'information de la Réalité Augmentée. L'utilisateur n'est plus immergé en totalité, ses sens ne sont pas remplacés, il voit et il entend le monde réel. Aussi, l'information n'est plus seulement représentée par du texte ou des images sur un écran, mais par des objets fixes et vivants autour de l'utilisateur. Des hologrammes de lumière et de son.

Microsoft est le premier à proposer avec HoloLens un appareil autonome pensé pour cette réalité Mixte.

HoloLens Autonome et pratique

Au-delà d'avoir le bénéfice évident pour un utilisateur de jouir de la pleine liberté de ses deux mains, l'autonomie se caractérise également pour HoloLens par l'absence complète de câbles. Contrairement à la plupart des appareils de Réalité Virtuelle, le casque HoloLens est une machine **complète**, faisant tourner son propre système d'exploitation, Windows 10.

Son design unique lui permet de répartir son poids plume de 579 grammes sur un anneau de support dissocié des « lunettes » proprement dites. De ce fait, le poids de l'appareil ne repose pas sur le nez de l'utilisateur. De plus, il n'y a pas la nécessité de masquer complètement le monde extérieur, réduisant à zéro la nécessité du casque de toucher le visage. Les lunettes peuvent rester ainsi en suspension dans l'air, maintenues par l'anneau de support conçu pour cet usage, et autorise l'usage de lunettes de vision sans aucun problème.

L'appareil est compatible Wi-Fi et Bluetooth, mais ne dispose pas d'emplacement pour carte

SIM. HoloLens possède une autonomie de 2h30 pour un usage intensif, à plus de 5h pour un usage modéré.

Conscient de l'environnement

Microsoft a su réutiliser son savoir-faire, en utilisant certainement son expérience Kinect, pour peaufiner cet élément clé de la Réalité Mixte, qui est celui de la contextualisation de l'expérience par la détection de l'environnement.

L'arsenal de capteurs de l'HoloLens permet en effet une détection bluffante de précision de la pièce dans laquelle elle se trouve. Difficile à tromper, l'appareil sait où il est, s'il a déjà été ici, ce qui a changé depuis la dernière fois qu'il y était. Un hologramme posé quelque-part, se retrouve au même endroit jusqu'à ce qu'on l'enlève. Une illusion parfaite, qui n'est cependant pas possible dans les lieux ouverts, pour lesquels l'appareil n'a pas de points d'ancrages auxquels se rattacher.

L'équipement permettant la détection de la pièce, sont les suivants :

- Un capteur de mouvement inertiel ;
- 4 caméras de capture de l'environnement (2 latérales, 2 frontales) ;
- 1 caméra de détection de profondeur.

En outre, l'appareil se dote des capteurs suivants :

- 1 caméra vidéo de 12 Mégapixels ;
- 1 capteur de luminosité ambiante ;
- 4 microphones orientés, permettant une spatialisation de l'origine des sons.

Microsoft, au travers du SDK d'HoloLens répond au besoin de compréhension de l'environnement, permettant d'analyser la pièce dans laquelle on se trouve, et d'en déduire les surfaces planes que constituent le sol, le plafond, les murs et éventuellement les tables et les chaises. Ces capacités limitées ne suffisent pas à adresser des problématiques de compréhension de l'environnement plus complexe, comme l'identification d'un objet particulier ou d'une personne. De nombreux projets à l'initiative de startups telle que Diota ou d'industriels tel que Vuforia, anciennement propriété de Qualcomm et aujourd'hui aux mains de PTC, tentent de répondre à cette problématique.

Interactif

Microsoft propose à l'utilisateur quelques mouvements simples de la main permettant d'interagir avec l'appareil. Le fait de positionner la main devant soi (pas forcément dans le champ de vision), index levé, indique à l'appareil que l'utilisateur est prêt. Le fait ensuite de baisser ou de lever l'index revient à effectuer la

même opération que sur une souris, permettant le « click » et les opérations de « drag & drop ». Un dernier geste consistant à joindre ses cinq doigts et à les ouvrir vers le haut s'apparente à la touche « Windows » d'un PC.

La cible de ces actions est simplement un curseur qui est positionné au centre de la vision de l'utilisateur. Microsoft juge que ces quelques gestes suffisent pour interagir avec n'importe quelle application typique HoloLens, et nous n'avons pas encore eu affaire à des cas nécessitant l'implémentation de la prise en charge d'un nouveau type de geste.

L'intégration de l'assistant personnel Cortana permet également de converser avec l'appareil. Ce dernier est pour l'instant uniquement disponible en anglais, même si une connexion Internet pourrait permettre la prise en charge du français. Une reconnaissance vocale personnalisée reste possible sur l'appareil, mais l'implémentation devra être complètement prise en charge par l'application.

Un affichage limité mais précis

Une fois sur le nez, l'appareil va projeter une image sur une couche semi-transparente de l'appareil, pour chaque œil. Cette image va se superposer à la vision de l'utilisateur, qui percevra les parties illuminées des lunettes comme faisant partie de sa réalité. Le calcul en temps réel de l'image à projeter en fonction du déplacement de la tête, spécifiquement pour chaque œil, est assuré par un processeur dédié, construit par Microsoft baptisé HPU pour « Holographic Processing Unit ».

Cette image projetée dans chaque œil ne couvre pas le champ de vision de l'utilisateur, mais plutôt une zone rectangulaire, qui équivaut à regarder un écran de 14" (35cm) à une distance de 60 cm.

Cette limitation à une zone réduite du champ de vision a pu donner lieu à des déception de la part des premiers utilisateurs qui ont été exposés aux premières vidéos faites par le marketing Microsoft, faisant miroiter une couverture complète de la vue. Microsoft a depuis agrémenté ses vidéos d'un rectangle apparaissant lorsque la vidéo présente une vue à la première personne.

L'affichage réduit de l'HoloLens est compensé par une précision accrue. En effet, la précision se calcule en nombre de pixels par degré angulaire, et, même s'ils ciblent des types de réalité alternatives différentes, là où l'Oculus et le Vive tournent autour de 12 pixels par degré, HoloLens surprend car l'appareil est capable d'afficher plus de 40 pixels par degrés !

L'illusion du son

L'appareil ne masque pas l'oreille. En effet, suivant les principes de Réalité Mixte, l'HoloLens permet d'entendre les sons ordinaires de notre entourage inaltérés, et propose des haut-parleurs au-dessus de l'oreille. Lorsque les objets sont en dehors de notre champ de vision, un des meilleurs moyens de percevoir ces objets est à travers le son. Microsoft exploite complètement ce principe à travers la notion de son spatialisé proposée par l'appareil. Ce son spatialisé permet de contrôler l'attention de l'utilisateur à l'aide de bruits directionnels et de l'immerger davantage à l'aide de sons d'ambiance.

Adapté aux SI des entreprises

Affichant un prix public de 3 299 TTC pour la version de base, la version Entreprise à 5 489 TTC (prix de départ) inclut les fonctionnalités typiques permettant la gestion d'un parc HoloLens en Entreprise : support du Windows Store pour entreprises, l'intégration aux outils de MDM (Mobile Device Management), le mode kiosque, les mises à jour, gestion des identités, support, etc. Aussi, HoloLens s'appuie sur des standards déjà établis dans le monde de la 3D, par l'utilisation des outils Microsoft classiques comme .NET et DirectX par Visual Studio, mais aussi par l'utilisation de Unity 3D qui est un outil de référence notamment dans la création de scènes dans le monde du jeu-vidéo. Les entreprises peuvent donc puiser dans le vivier déjà existant de profils pouvant directement programmer des applications pour HoloLens.

Microsoft investi, investit

HoloLens est un produit en production. Il ne s'agit plus d'une bêta, d'une expérimentation, mais vraiment d'un produit fini, supporté par Microsoft. Il indique par ailleurs une nouvelle itération de HoloLens prévue pour 2019. Ce changement lent contraste par rapport aux itérations rapides offertes par les acteurs de la Réalité Virtuelle comme HTC Vive ou Oculus. Il garantit cependant une stabilité et un support cohérent du produit.

Avec l'arrivée de Windows Holographic, la surcouche de Microsoft Windows 10 supportant les casques 3D qui définit un nouveau standard permettant l'arrivée de nouveaux appareils 3D à bas coût, Microsoft marque son intérêt en étant moteur de cette révolution 3D.

La Réalité Mixte : un outil d'entreprise

Les discussions avec des professionnels confirment l'intérêt de ces nouvelles technologies



pour l'entreprise, mais également aident à discerner la vraie valeur ajoutée de chacune de ces solutions. Si la Réalité Virtuelle s'impose dans les scénarios de visualisation d'environnements en trois dimensions, où la technologie est déjà mature et les grands acteurs du marché de la conception assistée par ordinateur proposent déjà des solutions, elle s'avère inadaptée au travail collaboratif car elle isole l'utilisateur et le limite dans sa liberté de mouvement.

Néanmoins, pour la Réalité Mixte, l'étude de scénarios métiers montre qu'il existe plusieurs thèmes récurrents, quelle que soit l'industrie, pour lesquels la Réalité Mixte s'impose :

- **L'assistance didactique** à l'exécution de tâches, nécessitant une approche didactique visant à montrer à l'utilisateur les actions à réaliser ou à l'aider à procéder à ces actions. Par exemple, on pourra aider à l'assemblage de pièces mécaniques en affichant les pièces à assembler, une à une, et en guidant l'utilisateur dans les différentes étapes. On pourra apporter de l'information contextuelle aidant l'utilisateur à réaliser sa tâche, en particulier dans des situations où **deux mains libres** sont nécessaires.

- **L'assistance à distance**, permettant par exemple à un technicien de faire appel à un interlocuteur expérimenté. Le « sachant » pourra par exemple être devant un poste PC classique, verra ce que le technicien voit, et peut alors orienter le technicien, et pointer ou annoter des éléments dans l'environnement du technicien.

- **La collaboration** entre plusieurs personnes, qu'elles soient dans la même pièce ou dans des lieux différents, visualisant un même produit, ou des données. Il s'agit de voir une scène à plusieurs, la modifier, tourner autour, si possible sans être gêné par des câbles.

- La **visualisation en recherche et développement** de l'impact de la modification de paramètres sur un modèle 3D. Il s'agit de pouvoir visualiser ces changements au-delà de la représentation biaisée d'un moniteur 2D classique. Il peut s'agir par exemple de scénarios d'architecture ou de conception produit.

- **L'éducation et la formation** peuvent utiliser la réalité mixte car elle permet de représenter fidèlement des concepts 3D souvent difficiles à décrire autrement. On peut penser en particulier aux cours d'anatomie, ou de mécanique.

- La **prévisualisation au réel** de mobilier dans une pièce est un thème également adapté à la réalité mixte. Elle permet de pré-visualiser en temps réel l'encombrement d'un objet dans une pièce, par exemple, une machine-outil en entreprise, une cuisine équipée chez un particulier, ou un scanner dans un hôpital. Ces thèmes se déclinent en scénarios métiers qui imposent souvent de répondre à plusieurs problématiques techniques complexes : comprendre son environnement physique, savoir se positionner et s'orienter dans cet environnement et s'intégrer avec des systèmes d'information. Les applications de Réalité Mixte apportent de la valeur et une vraie réponse aux besoins des entreprises dès lors qu'elles sont capables de consommer ou d'alimenter ces systèmes d'informations.

Une approche nouvelle pour les sociétés de service

La conception et le développement d'applications de Réalité Mixte est au croisement de mondes et de techniques très différentes :

- La présentation ou l'affichage relèvent directement du domaine du **jeu vidéo**, qui a défini les méthodes et outils pour concevoir,

animer et afficher des éléments en deux et trois dimensions.

- La gestion des données fait appel aux concepts du **big data et de l'IoT** au travers de scénarios impliquant de collecter des informations sur des capteurs IoT, pouvant communiquer avec l'appareil de réalité augmentée ou mixte, de les traiter à l'aide du Machine Learning, puis d'exploiter les résultats, là aussi en réalité augmentée ou mixte.
- La **mobilité** des appareils tels qu'HoloLens impose un modèle applicatif similaire à celui des applications pour smartphone, incluant la gestion de ressources limitées, mais également la gestion de l'absence de connectivité dans le cas de zone blanche, ou d'absence de wifi sur certains sites industriels.

Ce constat a une conséquence directe sur les compétences nécessaires à la réalisation de ces applications. Un développeur d'applications d'entreprise devra se former aux concepts de la 3D, apprendre les concepts théoriques et mathématiques qui en découlent, s'approprier les outils adaptés.

Une application de Réalité Mixte est réussie en premier lieu si elle fait un usage intelligent des capacités de détection de l'environnement, et des possibilités d'affichage de l'appareil, tout en gardant en vue la valeur ajoutée pour l'utilisateur et l'entreprise. Ainsi, la phase de conception Expérience Utilisateur (UX) prend une place prépondérante dans le développement de l'application, et les aspects techniques interviennent tard dans le développement, après plusieurs cycles itératifs UX / Client. Ce phénomène est le prolongement de ce que l'on observe déjà dans le développement d'application mobile, où il n'est plus envisageable de proposer en entreprise une application qui n'ait pas le niveau UX des applications grand-public d'aujourd'hui.

Les compétences UX doivent donc être repensées. En effet, au-delà de la nécessité de connaître les limites de l'appareil utilisé, c'est à eux qu'incombe également la conception d'objets 3D, ou bien l'intégration d'objets 3D existants déjà dans l'entreprise. Ces profils UX n'interviennent plus ponctuellement avant une période longue de développement, ils sont présents pendant l'intégralité de la construction de l'application.

Par conséquent, une société de services pourra proposer des services touchant à la Réalité Mixte si elle est capable de concentrer de nouvelles compétences sur 3 plans : technique 3D, UX 3D, Méthodologie UX First.

La vraie valeur des Tests Logiciels



Bernard Homès
Président
TESSCO sas
Fondateur et ex-président
du Comité Français des
Tests Logiciels
(bhomes@tesscogroup.com)

Une expérience de 33 ans dans la qualité et les tests logiciels.

Souvent le coût des tests est vu uniquement à court terme, lié à la livraison d'une version spécifique de logiciel, alors qu'il devrait être vu dans l'optique d'une amélioration continue de la qualité et des services que les équipes de tests fournissent à l'entreprise et aux actionnaires. Trop souvent, le calcul de la valeur des tests se limite à une différence entre les coûts de détection et de correction des défaillances et le coût de la non-qualité (c'est à dire la somme des coûts subis par l'entreprise et ses clients en cas de défaillance en production). Combien de responsables de tests, ou de DSIs, ont eu à répondre à des commentaires tels que :

- Pourquoi les tests coûtent-ils si cher et prennent-ils tellement de temps ?
- Comment se fait-il que cette défaillance n'aît pas été trouvée lors des tests ?
- Nous pourrions faire des économies substantielles en externalisant les tests, et nous obtiendrions aussi un meilleur service.

Si telle est votre situation, vous n'êtes pas seuls : partout dans l'industrie vos collègues souhaitent que vous fournissiez de la valeur et doutent que vous la fournissiez. Beaucoup d'entre eux considèrent les tests comme un mal nécessaire, un coût à limiter et à contrôler, car les tests leur semblent un trou noir qui absorbe des ressources, sans fournir quoi que ce soit en retour.

Mauvaises et bonnes nouvelles

La mauvaise nouvelle est que c'est un problème que le responsable des tests doit résoudre. C'est la responsabilité du responsable de tests de s'assurer de livrer de la valeur et de communiquer sur cette valeur. Si cette valeur n'est pas démontrée et livrée, ce n'est qu'une question de temps avant que l'équation ne se résolve par un calcul simple : remplacer l'équipe de test par une équipe moins chère. Démontrer la valeur livrée est un aspect clé pour fournir encore plus de

Souvent l'on cherche à voir la valeur des tests en la comparant à l'absence de tests. De nos jours cette comparaison n'a plus de raison d'être : tout le monde est conscient de l'utilité des tests, mais certains d'entre nous ne se rendent pas compte de ce que les tests peuvent apporter en plus d'une réduction des défauts trouvés par les clients.

valeur. La bonne nouvelle est que c'est votre problème et que vous avez tous les paramètres pour le résoudre. Vous en êtes capable, cela a été fait de nombreuses fois, dans de nombreuses industries et quelle que soit la taille de la société.

Commençons par penser différemment

L'évaluation de la rentabilité des tests passe par une évolution du paradigme pour les tests : ce n'est pas une question de tests ou de qualité. C'est une question de résultat pour l'entreprise et de performances de l'entreprise. Le test doit être reconnu comme un contributeur clé du succès de l'entreprise, avec des mécanismes de gestion transparents, des mesures améliorées et un focus continu sur l'amélioration des processus et des produits. Les quatre étapes de l'amélioration sont illustrées ici : [1]

Modifier son mode de pensée et éviter les pièges de valeur

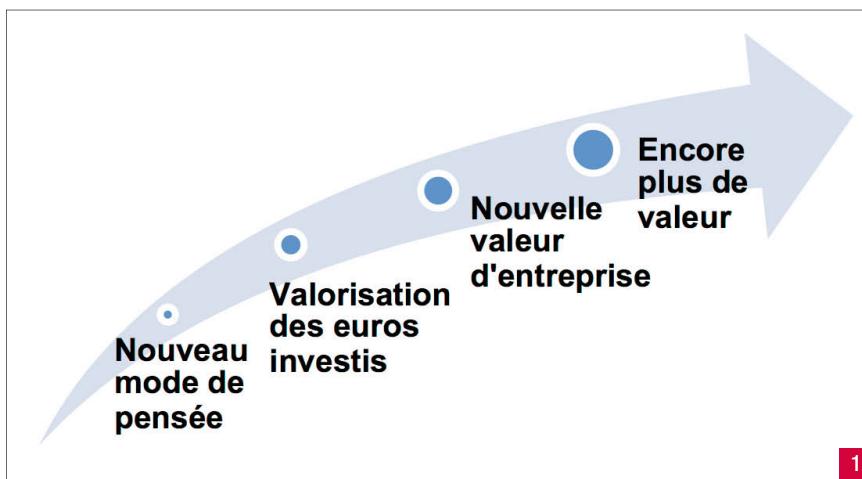
Les pièges de valeur sont les pratiques et comportements qui sur le moment peuvent paraître correctes, mais qui en fait amèneront des soucis. Ils empêchent les activités de tests de fournir de la valeur, et empêchent le reste de l'entreprise d'apercevoir la valeur livrée. Les pièges de valeur ne sont pas des défaillances claires telle que la livraison d'un logiciel de mauvaise qualité, mais sont plus insidieux. Dire « le développement est le client des tests, et le client a toujours raison » peut sembler une bonne idée, mais sur le long terme cela mènera les tests à la faute car les clients ont souvent tort ; et appeler des collègues « clients » sépare les membres de l'équipe de tests du reste de l'entreprise. Une équipe de tests prise dans un piège de valeur, se focalise sur elle-même. C'est une erreur, car ce sont les performances de (et pour) l'entreprise qui doivent diriger les tests.

Surmonter les pièges de valeur requiert que les équipes de tests reconnaissent qu'elles sont dans un environnement changeant, qui demande de nouveaux modes de pensée.

Montrer que les tests fournissent de la valeur pour les euros investis

Reconnaitre la valeur des tests pour l'entreprise nécessite plus qu'un simple changement de point de vue. Dans beaucoup d'entreprises, les tests ont une longue et pénible histoire. Même quand des cadres comprennent que les tests créent de la valeur, il est nécessaire de leur montrer que c'est une valeur spécifique pour l'entreprise, là où cela compte le plus. Les tests managers qui réussissent commencent par démontrer et améliorer la valeur des tests en démontrant la valeur pour les euros investis. Ils démontrent chaque jour que les équipes de tests méritent leur rôle de partenaire privilégié de l'entreprise, en fournissant des informations de coûts, de performances et de qualité que les « clients » des équipes de tests peuvent utiliser pour contrôler leurs propres coûts. Certains responsables de tests comparent l'efficacité de leurs équipes à celles d'autres sociétés semblables, de façon à pouvoir fournir des comparatifs de coûts à leurs « clients ». Communiquer sur la valeur et les coûts est un point de départ essentiel. Il permet de démontrer que le moyen de réduire le coût des tests n'est pas simplement de couper dans les budgets, mais plutôt d'ajuster le niveau de

En cas de défaillance (défaut passant en production), ce sont les processus qu'il faut étudier, les personnes ne font pas des erreurs volontaires. Une analyse des causes racines vous dira pourquoi la défaillance est arrivée. Les métriques vous permettront de déterminer si vous vous améliorez.



1

qualité là où une qualité ou une consommation excessive des tests n'apporte rien en termes de performances de l'entreprise.

Montrer comment les tests améliorent les performances de l'entreprise

Les responsables de tests efficaces ne se contentent pas de gérer leurs équipes de test, ils aident aussi leurs collègues à prendre les bonnes décisions en termes d'investissement en tests et qualité des logiciels. En aidant leurs collègues à identifier leurs besoins, en arbitrant sur les investissements, en exécutant les projets et en s'assurant que les bénéfices escomptés soient atteints, les responsables de tests créent un cercle vertueux qui maximise visiblement les investissements en test. Ceci permet de montrer l'apport constant que les tests fournissent à chacun des domaines de l'entreprise et les métriques mis en place permettent de mesurer les améliorations de façon continue, d'un mois sur l'autre.

Montrer comment les tests apportent encore plus de valeurs

Quand il se focalise sur les performances de l'entreprise et fournit de la valeur mesurable,

tôt ou tard le responsable des tests n'est plus considéré comme un simple technicien exécutant, mais comme un partenaire utile, capable de contribuer au-delà de sa seule spécialité. Certes, cela n'arrive pas du jour au lendemain, mais en traitant chaque initiative comme une opportunité d'entreprise, en démontrant la valeur des investissements consentis, et en mesurant les améliorations résultantes, le responsable des tests peut obtenir des responsabilités plus étendues.

Construire la valeur de bas en haut

Beaucoup de responsables de tests se plaignent de ne pas avoir l'opportunité de démontrer le potentiel que les tests possèdent pour améliorer l'entreprise, ou que, quand ils en ont la possibilité, personne ne les écoute. Cela provient souvent du manque de crédibilité associé aux équipes de tests. Le succès peut être atteint en créant cette perception de valeur pas à pas, chaque étape étant la fondation de la suivante.

Cela commence par une compétence à fournir le service attendu de façon fiable, économique et avec un très haut niveau de qualité. Si vous n'êtes pas en mesure de fournir de façon fiable et efficace, il restera toujours un soupçon dans l'esprit de vos partenaires, et ils ne vous feront pas confiance.

Finalement, le responsable des tests qui suit cette voie vers la valeur devient un égal des autres cadres supérieurs de l'entreprise, et augmente fortement leur impact et leur influence. Cette influence va plus loin que la simple gestion des tests, s'étendant aux équipes de développement et au marketing, voire à des niveaux hiérarchiques supérieurs.

Quelques points de réflexion

Les recommandations qui précèdent, et les quatre étapes qui les composent permettent d'éviter des problèmes prévisibles comme ceux-ci :

- Quand les équipes de tests fournissent peu de valeur, le test sera considéré comme un mal nécessaire, un coût à réduire à sa plus simple expression, de façon à garder un niveau de qualité minimal sans plus ;
- Quand le responsable des tests et ses équipes ne sont pas en mesure de créer des mécanismes de supervision transparents, les tests seront regardés avec soupçons car rien ne permettra de déterminer si les montants investis sont utilisés à bon escient ;
- Quand le responsable des tests et ses équipes ne sont pas en mesure de discuter du business de l'entreprise avec les autres parties prenantes (cadres, dirigeants, ...), ils seront toujours considérés comme des exécutants, pas des membres à part entière de l'entreprise ;
- Quand le responsable des tests et ses équipes, ne peuvent pas lier les investissements consentis à des améliorations mesurables des performances de l'entreprise, l'encadrement cherchera ailleurs un avantage compétitif et ignorera les manières dont le test et la qualité peuvent être appliqués pour améliorer l'entreprise.

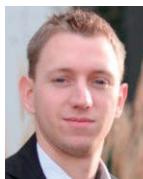
Un certain nombre de responsables des tests se reconnaîtront (ou leur entreprise) dans ces descriptions. Si c'est votre cas, vous n'êtes pas seuls. Cependant ce n'est pas un destin inévitable.

•



1 an de Programmez!
ABONNEMENT PDF : 35 €
 Abonnez-vous directement sur :
www.programmez.com
 Partout dans le monde.

Tour d'horizon de .NET Core



- Christophe GIGAX
Co-fondateur du Microsoft User Group de Strasbourg. Auteur du livre sur ASP.NET Core MVC : Maîtrisez ce Framework Web puissant, ouvert et multiplateforme

Au travers de .NET Core, les entreprises sont capables de construire des applications de tous types : Micro-service, Web via ASP.NET Core, Cloud avec Azure ou Web API pour la consommation de données. Ainsi, cette nouvelle plateforme est là afin de résoudre les problèmes de compatibilité avec les autres OS tels que Linux et macOS, mais aussi de proposer une solution de déploiement plus simple et plus légère pour ces applications.

Généralités

On peut considérer .NET Core comme étant une version plus légère mais surtout multiplateforme du .NET Framework classique que nous connaissons depuis des années. La plateforme est utilisable via les langages C# et F#, et bien-tôt avec Visual Basic via différents IDE dont Visual Studio, Visual Studio Code, Sublime Text ou Vim. La volonté des équipes de .NET Core est claire ici : permettre au plus grand nombre d'utiliser la plateforme selon les outils que chacun préfère.

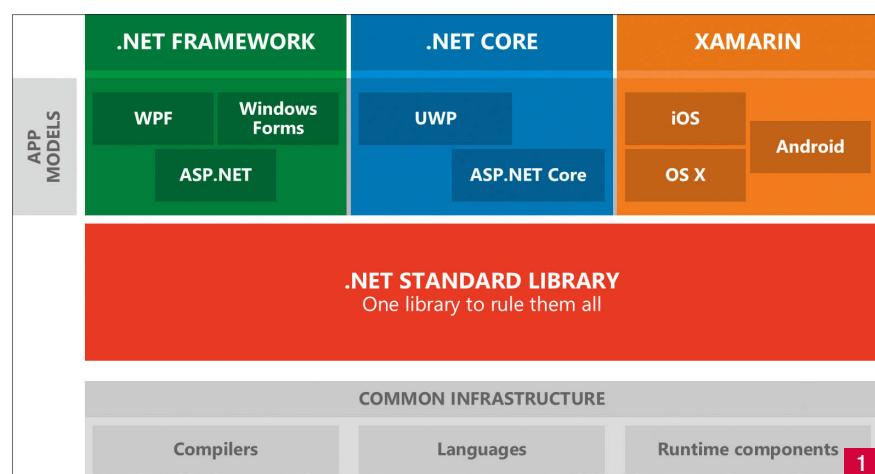
.NET Core implémente la spécification .NET Standard, ce qui permet avec la version 2.0 une meilleure interopérabilité entre les plateformes .NET Core, .NET Framework et Xamarin. L'objectif ici est le partage de code sans concession. [1]

.NET Core est bien plus vaste encore, car la plateforme embarque un outil en lignes de commandes très puissant permettant d'effectuer toutes les opérations nécessaires à la bonne gestion d'une application .NET Core sur Windows, Linux ou MacOS. Les IDE modernes tels que Visual Studio 2017 ne font qu'interpréter et lancer ces mêmes commandes via l'interface graphique.

L'aspect Open Source est également très soigné auprès de la plateforme .NET Core. L'intégralité du code faisant fonctionner la plateforme se trouve sur GitHub, dont le repository principal qui est dotnet/core ici : <https://github.com/dotnet/core>. A partir de là, il existe une multitude de repository faisant partie de l'environnement de .NET Core :

- dotnet/cli – outil en lignes de commandes ;
- dotnet/corefx – bibliothèques de classes fondamentales de .NET Core ;
- dotnet/coreclr – runtime de .NET Core ;
- dotnet/docs – documentation ;
- dotnet/sdk – SDK de .NET Core ;
- dotnet/standard - .NET Standard ;
- nuget/home – gestionnaire de paquet NuGet ;

.NET Core est une plateforme de développement très complète intégrée à l'éco-système Microsoft permettant la création d'applications multiplateformes et extrêmement souples à déployer. Publié via la .NET Foundation, la plateforme .NET Core est Open Source et utilise des licences MIT et Apache 2. Ce projet de grande envergure est la preuve concrète de la stratégie de Microsoft tournée vers l'Open Source.



Modèle de programmation avec Net Standard

- aspnet/home – point de départ de l'environnement ASP.NET Core ;
- aspnet/Docs – documentation d'ASP.NET Core ;
- dotnet/netcorecli-fsc – repository sur F# pour .NET Core.

Installation

L'installation de .NET Core est simple sur les différentes plateformes (Windows, Linux et Mac), et permet même d'être installé sur Docker. Ce fut réellement le souhait des équipes de Microsoft : proposer une plateforme de développement simple et déployable quasiment partout dans le paysage informatique d'aujourd'hui.

Sur Windows

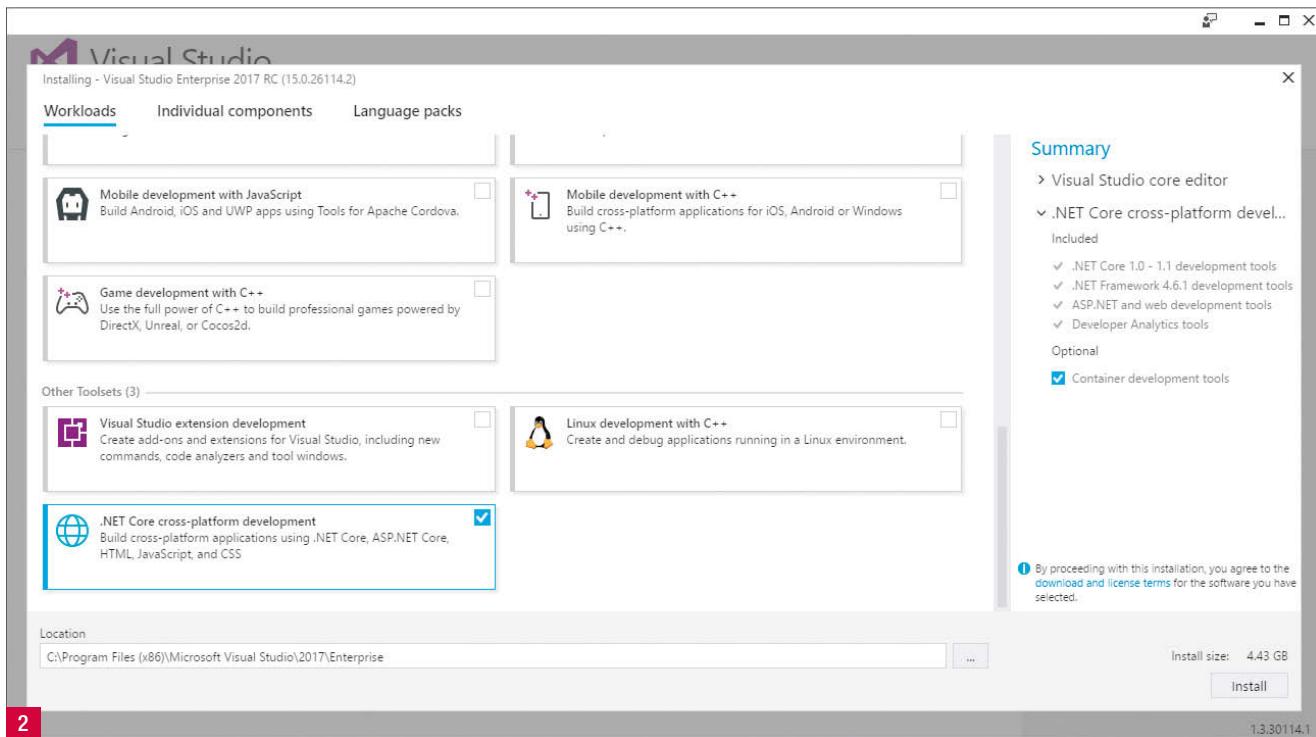
Concernant la plateforme de Microsoft, il existe 2 techniques différentes afin d'installer .NET Core. La première est tout simplement d'installer l'IDE Visual Studio 2017 qui propose un module .NET Core durant l'installation. Avec ceci, le développeur est prêt à développer avec la plateforme. [2]

La seconde technique utilise le SDK .NET Core pour Windows et particulièrement un outil en ligne de commande intitulé dotnet. Il faut tout d'abord installer le SDK (<https://www.microsoft.com/net/core#windowscmd>), et ainsi le développeur bénéficiera immédiatement de la plateforme .NET Core via l'outil en lignes de commandes dotnet.

Sur Linux

Les distributions supportées par .NET Core sont :

- Ubuntu & Mint ;



Installation du module .NET Core avec Visual Studio 2017

- Red Hat ;
- Debian ;
- Fedora ;
- CentOS & Oracle ;
- openSUSE.

Linux est la plateforme qui contient le plus de distributions différentes ; chaque distribution possède donc sa spécificité. La suite de l'article présentera l'installation de .NET Core sur les plateformes Linux les plus utilisées dernièrement, c'est-à-dire Mint, Ubuntu, Debian et openSUSE.

Sur Linux Mint, .NET Core est installable à partir de la version 17. Il faut tout d'abord utiliser une commande permettant de s'assurer que l'outil **apt-get** puisse bien télécharger le package nécessaire.

```
sudo sh -c 'echo "deb [arch=amd64] https://apt-mo.trafficmanager.net/repos/dotnet-release/ trusty main" > /etc/apt/sources.list.d/dotnetdev.list'
sudo apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv-keys 417A0893
sudo apt-get update
```

Les mêmes commandes sont utilisables sur Ubuntu 14.04. Concernant Ubuntu 16.04, il faut utiliser les commandes suivantes :

```
sudo sh -c 'echo "deb [arch=amd64] https://apt-mo.trafficmanager.net/repos/dotnet-release/ xenial main" > /etc/apt/sources.list.d/dotnetdev.list'
sudo apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv-keys 417A0893
sudo apt-get update
```

Et enfin pour Ubuntu 16.10 c'est à peine différent :

```
sudo sh -c 'echo "deb [arch=amd64] https://apt-mo.trafficmanager.net/repos/dotnet-release/ yakkety main" > /etc/apt/sources.list.d/dotnetdev.list'
sudo apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv-keys 417A0893
sudo apt-get update
```

Il suffit ensuite de lancer l'installation du SDK .NET Core via la commande suivante (il est important qu'aucune autre version de .NET Core soit installée avant) :

```
sudo apt-get install dotnet-dev-1.0.1
```

La version de .NET Core installée ici est 1.1. Il est possible de vérifier les versions de .NET Core par distribution à cette adresse : <https://www.microsoft.com/net/download/linux>.

Pour Debian la démarche est un peu différente. Il faut tout d'abord s'assurer tout de même qu'aucune autre version de .NET Core n'est installée. Ensuite, il faut télécharger les binaires du SDK .NET Core, les extraire puis ajouter l'outil **dotnet** dans le PATH. Les commandes ci-dessous effectuent toutes ces actions automatiquement :

```
sudo apt-get install curl libunwind8 gettext
curl -sSL -o dotnet.tar.gz https://go.microsoft.com/fwlink/?linkid=843453
sudo mkdir -p /opt/dotnet && sudo tar zxf dotnet.tar.gz -C /opt/dotnet
sudo ln -s /opt/dotnet/dotnet /usr/local/bin
```

Enfin, sur la distribution openSUSE, le principe est le même que sur Debian, mais les commandes diffèrent selon les versions. Concernant openSUSE avec comme version minimum 13.2, les commandes sont les suivantes :

```
sudo zypper install libunwind libicu
curl -sSL -o dotnet.tar.gz https://go.microsoft.com/fwlink/?linkid=843447
```

Pour openSUSE 42.1 minimum, les commandes sont :

```
sudo zypper install libunwind libicu
curl -sSL -o dotnet.tar.gz https://go.microsoft.com/fwlink/?linkid=843451
```

Enfin, l'installation du SDK et l'ajout dans le PATH se fait de la manière suivante :

```
sudo mkdir -p /opt/dotnet && sudo tar zxf dotnet.tar.gz -C /opt/dotnet
sudo ln -s /opt/dotnet/dotnet /usr/local/bin
```

Il est clair que sur Linux, la version et la manière d'installer la plateforme peuvent être différentes d'une distribution à une autre, mais l'objectif de .NET Core est clair : proposer une plateforme légère et puissante de développement sur la plupart des systèmes d'exploitation, et Linux ne déroge pas à la règle. Les équipes de Microsoft travaillent d'arrache-pied sur ce sujet.

Sur Mac

Afin de faire fonctionner .NET Core sur Mac, il est d'abord nécessaire d'installer OpenSSL. Pour faciliter cette installation, nous allons utiliser Homebrew, un gestionnaire de package sur Mac populaire et très utile. Les commandes suivantes permettent d'installer OpenSSL :

```
brew update
brew install openssl
mkdir -p /usr/local/lib
ln -s /usr/local/opt/openssl/lib/libcrypto.1.0.0.dylib /usr/local/lib/
ln -s /usr/local/opt/openssl/lib/libssl.1.0.0.dylib /usr/local/lib/
```

La commande **ln** sous Mac permet de créer un lien symbolique. Dans notre cas, cela permet à .NET Core de trouver les librairies correspondantes à OpenSSL et à la cryptographie dans les librairies du système. Par la suite, l'installation de .NET Core se fait via l'installateur prévu à cet effet : <https://go.microsoft.com/fwlink/?linkid=843444>. La dernière version installée sera la 1.1.

L'ensemble des téléchargements disponibles pour .NET Core est disponible ici : <https://www.microsoft.com/net/download/core#/sdk>.

Sur Docker

Docker est une plateforme mettant en œuvre le principe de « container ». Ce principe permet d'encapsuler des processus logiciels complexes afin d'embarquer en même temps les dépendances, et ainsi de pouvoir exporter des exécutables sur n'importe quel système d'exploitation, peu

importe la configuration nécessaire ou encore le nombre de dépendances.

Dans le but de faire fonctionner .NET Core sur Docker, il faut dans un premier temps installer Docker. Le site officiel de la plateforme propose plusieurs installateurs suivant votre système d'exploitation : <https://www.docker.com/community-edition#/download>.

Ensuite, il suffit de lancer la commande suivante afin de disposer d'un environnement intégrant .NET Core déjà prêt :

```
docker run -it microsoft/dotnet:latest
```

Cet environnement est disponible depuis le Microsoft Docker Hub : <https://hub.docker.com/r/microsoft/dotnet/>. [3]

Et voilà ! Il n'y a rien de plus simple. Le développeur peut ainsi développer dans cet environnement et exporter sa plateforme vers n'importe quel autre système sans contrainte.

De quoi est composé .NET Core ?

Le framework .NET est composé d'une multitude de briques lui permettant de faire fonctionner le code source de manière managée et sécurisée. .NET Core ne déroge pas à la règle, et possède lui aussi ses propres composants logiciels.

Un runtime

Le runtime de .NET Core, intitulé CoreCLR, est le cœur du nouveau framework. C'est lui qui s'occupe d'allouer les objets, de gérer le 'garbage collector', de charger les classes définies par le framework et par le développeur, de gérer le système des types et ainsi de suite.

L'essentiel du runtime se trouve dans le package **Microsoft.NET Core.Runtime. CoreCLR**.

Volontairement ce package ne fait pas grand-chose tout seul. En effet, le CoreCLR n'implémente que le minimum de classes vitales dont il a besoin pour subsister, c'est-à-dire *System.Object*, *System.String* et ainsi de suite.

Cela permet d'avoir notamment un runtime complètement découpé des implémentations bas niveau que l'on peut trouver dans les applications classiques comme *System.Collection*, *System.IO* ou d'autres.

Avec le CoreCLR on trouve également un nouveau compilateur JIT (Just-

PUBLIC | AUTOMATED BUILD

microsoft/dotnet ☆

Last pushed: 6 days ago

Repo Info Tags Dockerfile Build Details

Short Description

Official images for .NET Core for Linux and Windows Server 2016 Nano Server

Full Description

Supported tags and respective Dockerfile links

- 1.0.4-runtime, 1.0-runtime ([1.0/debian/runtime/Dockerfile](https://hub.docker.com/r/microsoft/dotnet/))

Docker Pull Command

```
docker pull microsoft/dotnet
```

Owner



microsoft

Page officielle du Hub Docker de Microsoft

in-Time), appelé RyuJIT dont les travaux pour concevoir ce nouveau compilateur ont tourné autour de :

- La compatibilité avec les anciens JITs x86 et x64 ;
- Améliorer les performances tant au niveau de l'allocation mémoire que de la génération de code ;
- Assurer un haut niveau de performance lors des optimisations et des transformations ;
- S'assurer que l'architecture de RyuJIT couvre la plupart des scénarios possibles.

Écrit en C++, ce nouveau runtime est constitué autour de 2.2 millions de lignes de code, dont le JIT constituant 320k de ligne et 55k constituant le GC.

Plusieurs bibliothèques

CoreCLR ne serait rien sans CoreFX. Ce dernier est un ensemble de bibliothèques fournissant l'essentiel des types primitifs, collections et autres utilitaires essentiels à la création d'applications d'envergure. Le repository GitHub se trouve ici : <https://github.com/dotnet/corefx>.

On y retrouve des bibliothèques bien connues telles que :

- System.Collections ;
- System.IO ;
- System.XML ;

Microsoft a également mis à disposition du public une banque de données relative aux API utilisables sur ses plateformes : <https://docs.microsoft.com/en-us/dotnet/api/>. On y retrouve notamment tous les namespaces développés au sein de CoreFX.

Un ensemble d'outils du SDK

Enfin, .NET Core peut aujourd'hui jouir d'un outil en lignes de commandes extrêmement performant intitulé **dotnet**. Ce dernier permet une multitude de choses telles que la compilation de projet .NET, le lancement de l'application, le testing, la gestion des paquets NuGet et bien d'autres encore.

Le but de cet outil est la simplicité d'utilisation couplée à l'extensibilité de ses fonctionnalités propres à un outil de type CLI. Les IDE de type Visual

Studio 2015 / 2017 ne font que lancer des commandes de l'outil **dotnet** lorsque le développeur travaille avec .NET Core.

L'outil est composé de 2 parties importantes :

- Un driver ;
- Des commandes bien précises intégrées au package.

Typiquement, le développeur va utiliser l'outil via une console en lançant des commandes du type **dotnet {command}**. Le driver ici va simplement mapper la commande avec un exécutable (contenu dans le dossier du PATH de **dotnet**) selon la convention de nommage **dotnet-{command}**. Tous les arguments sont ensuite transmis à l'exécutable sous-jacent. Il est donc extrêmement facile de rajouter ses propres commandes si le développeur en a besoin. Les équipes en charge de l'outil ont même pensé à écrire des guidelines à ce sujet : <https://github.com/dotnet/cli/blob/master/Documentation/general/cli-ux-guidelines.md>.

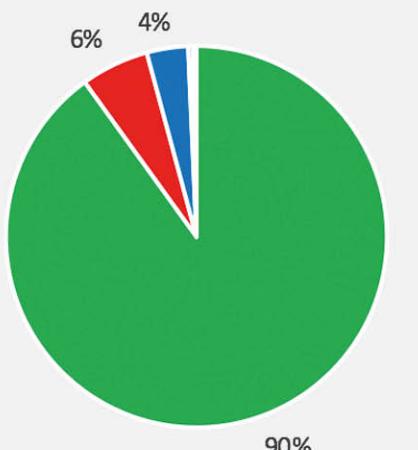
L'architecture de .NET Core

.NET Core a été conçu comme un produit multiplateforme et avec une grande capacité d'adaptation sur les différents OS (Windows, Mac et Linux). Par exemple, des projets de portage sont actuellement en cours afin de porter .NET Core sur d'autres plateformes. Le projet LLILC (LLVM base MSIL Compiler) en est l'illustration parfaite. Son but est de produire du code .NET Core multi-plateforme selon le compilateur LLVM.

L'environnement est divisé en plusieurs composants, rendant ainsi plus facile l'adaptation des composants spécifiques à chaque plateforme en remplaçant simplement la brique nécessaire, et non tout l'environnement, et ceci peut être fait selon un calendrier différent. Le fil conducteur du produit est de développer des bibliothèques indépendantes aux plateformes, et ainsi réduire le code dupliqué, augmenter l'efficacité des développeurs et limiter le code spécifique à chaque plateforme. Dans le cas où il n'y a pas le choix, les équipes de Microsoft utilisent soit des implémentations spécifiques à la plateforme, soit de la programmation conditionnelle, avec une forte préférence pour la dernière option.

Les diagrammes ci-dessous représentent la part de code partagé que l'on peut trouver dans CoreFX, mais également la part de code spécifique pour chaque plateforme. [4]

~ Lines of CoreFX C# Code



■ Shared ■ Windows ■ Unix ■ Linux ■ OSX

~ Lines of Platform-specific CoreFX C# Code

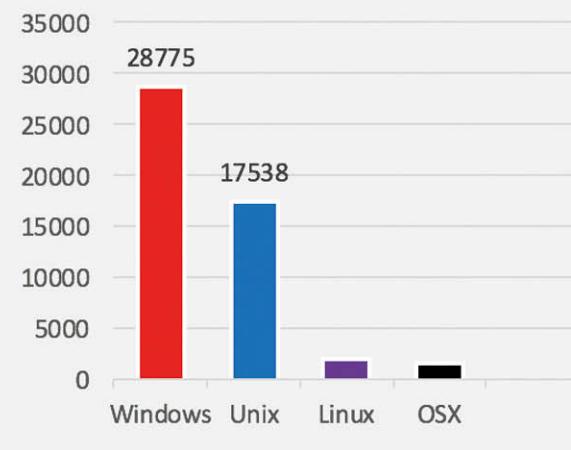


Diagramme montrant les proportions de code partagés et de code spécifiques aux plateformes

La majorité du code est donc partagée. Concernant le code spécifique, on peut voir que majoritairement Windows possède le plus de code, ceci étant notamment dû à certaines fonctionnalités présentes sur Windows, telle le registre, qui ne sont pas présentes sur les autres plateformes.

.NET Core est une version plus légère que .NET Framework, et ainsi on peut noter des différences significatives entre les 2 versions du framework. Tout d'abord, on peut noter que .NET Core n'implémente pas les modèles d'applications types WPF ou WinForms car ces derniers reposent sur DirectX. Cependant, .NET Core implémente bien les modèles d'application Console et ASP.NET. Ensuite, d'un point de vue des APIs implémentées par chacun, nous pouvons dire que .NET Core implémente la majorité des APIs de .NET Framework, mais selon des règles de nommage différentes ou selon des namespaces différents. Cela induit un portage nécessaire du .NET Framework vers .NET Core.

En revanche, .NET Core implémente bien la spécification .NET Standard permettant ainsi à .NET Core de croître au fil du temps selon l'évolution de la spécification. .NET Core implémente quelques sous-systèmes de .NET Framework, mais pas tous car la nouvelle plateforme veut se doter d'une implémentation et d'un modèle de programmation plus simples. Par exemple, .NET Core n'implémente pas le Code Access Security (CAS), alors qu'il implémente Reflection.

En termes de système d'exploitation, .NET Framework est limité par les systèmes Windows alors que .NET Core supporte macOS et Linux.

Toutes ces différences font que .NET Core est réellement un renouveau du .NET Framework apportant de nouvelles perspectives, de meilleures performances et plus de simplicité.

Quels usages pour .NET Core ?

Fondé sur une nouvelle architecture, nous avons bien compris que .NET Core est une version plus légère du .NET Framework classique. Cependant, nous pouvons nous demander pour quel usage il a été conçu. Selon Microsoft, il existe 6 scénarios pour lesquels il faut choisir .NET Core :

- Les besoins cross-platform. En effet, .NET Core a été conçu afin de fonctionner sur Windows, Linux et Mac. Ainsi, il est tout à fait possible de construire une application ASP.NET Core ou des Web Services sur Linux ou Mac selon les pré-requis du projet ;
- Les architectures micro-services. De par sa nature totalement indépendante et facilement scalable, .NET Core est un candidat idéal à l'élaboration d'une application avec l'architecture micro-services. Le développement Cloud prend de plus en plus de place aujourd'hui dans l'écosystème, et Microsoft se devait de répondre à ces problématiques avec une plateforme solide ;
- La performance et la scalabilité. Du fait que .NET Core est une plateforme plus légère, ses performances sont bien plus élevées permettant à des applications Web conçues avec .NET Core de répondre à une plus grande demande de manière simultanée ;
- L'outil en ligne de commande dotnet. Sa simplicité d'utilisation fait de cet outil un pilier de la plateforme. Si votre projet nécessite de gérer le projet via des lignes de commandes, .NET Core reste un très bon choix car il a été réellement conçu autour du CLI ;
- Multiplicité des versions de .NET côté à côté. Les applications avec .NET Core peuvent cohabiter dans le même système côté à côté car le runtime est livré et embarqué avec l'application. Il n'est plus nécessaire d'installer le runtime comme avec le .NET Framework classique. Il est donc maintenant plus facile de faire fonctionner plusieurs applications avec des versions différentes ;

- Les applications UWP. Les Universal Windows Platform sont un type d'application permettant de créer des applications Windows 10 et Windows 10 Mobile. .NET Core est utilisé de manière sous-jacente afin de faire fonctionner ces applications.

Vis-à-vis du .NET Framework classique, il existe toujours des cas où il faut continuer avec l'ancien framework. Tout d'abord, si l'application existe déjà avec le .NET Framework classique, il n'est pas forcément nécessaire de tout migrer vers .NET Core.

Cependant, les nouveaux projets peuvent débuter avec .NET Core. Ensuite, beaucoup de bibliothèques ne sont pas compatibles .NET Core. Si le projet utilise l'une de ces bibliothèques, il n'est pas conseillé d'utiliser .NET Core. Enfin, il existe toujours des technologies .NET qui ne se basent pas sur .NET Core comme WPF, WinForms et ainsi de suite. Si le projet est de cette nature, il est impossible d'utiliser la nouvelle plateforme, il faut donc utiliser .NET Framework.

'dotnet', l'outil à tout faire

Présenté dans les sections précédentes, .NET Core est livré avec un outil en ligne de commande très puissant intitulé **dotnet**. Cet outil permet de gérer une application .NET Core (ASP.NET ou console) avec des commandes très simples via n'importe quelle console. Afin de découvrir cet outil, nous allons créer une application et décrire les commandes nécessaires à la bonne gestion de l'application.

Ma première app

Une fois le SDK .NET Core installé, la première commande à connaître est **dotnet new**. Cette commande lance un assistant permettant de configurer les premiers paramètres de l'application comme son nom, le type de projet, le framework utilisé ou encore le langage à utiliser. L'utilitaire propose plusieurs templates accessibles au développeur.

Templates	Short Name	Language	Tags
Console Application	console	[C#], F#	Common/Console
Class Library	classlib	[C#], F#	Common/Library
Unit Test Project	mstest	[C#], F#	Test/MSTest
xUnit Test Project	xunit	[C#], F#	Test/xUnit
ASP.NET Core Empty	web	[C#]	Web/Empty
ASP.NET Core Web App	mvc	[C#], F#	Web/MVC
ASP.NET Core Web API	webapi	[C#]	Web/WebAPI
NuGet Config	nugetconfig		Config
Web Config	webconfig		Config
Solution File	sln		Solution

[5] - Template disponible

Dans notre exemple, nous allons créer une application ASP.NET Core Web API. La commande utilisée est alors la suivante :

```
dotnet new webapi
```

La commande crée une arborescence de fichier typique pour une application ASP.NET Core Web API avec des contrôleurs Web API. La particularité des applications ASP.NET Core c'est que ce sont de simples applications console sur lesquelles nous allons lancer un hôte Web, en particulier Kestrel pour ASP.NET. Comme chaque application console, elle comporte un point d'entrée qui est le **Main** avec sa classe **Program**.

```
public class Program
{
    public static void Main(string[] args)
    {
        var host = new WebHostBuilder()
            .UseKestrel()
```

```

    .UseContentRoot(Directory.GetCurrentDirectory())
    .UseIISIntegration()
    .UseStartup<Startup>()
    .Build();

    host.Run();
}
}

```

Le **WebHostBuilder** est ici responsable de la création de l'hôte de l'application Web : c'est lui qui va lancer l'app selon les critères indiqués par les méthodes d'extensions. Les méthodes utilisées ici sont :

- **UseKestrel** : indique qu'on va utiliser le serveur Kestrel pour héberger l'application ;
- **UseContentRoot** : indique où chercher les fichiers de contenu de l'application Web, tels les fichiers MVC, les vues Razor, les fichiers de paramétrage JSON, et ainsi de suite Ici on utilise la racine du projet comme dossier de recherche ;
- **UseIISIntegration** : indique que l'application peut être délivrée à travers un serveur IIS et permet ainsi à IIS d'être utilisé comme *reverse proxy*. Cette méthode ne paramètre pas un serveur comme le fait **UseKestrel**, cependant il ne faut jamais exposer Kestrel directement au grand public, c'est pour cela que IIS est important ici ;
- **UseStartup** : indique quelle classe *Startup* il faut utiliser. Cette classe est la classe de paramétrage de l'application Web. Elle va permettre de configurer les différents middlewares à utiliser, les pages d'erreurs, les logs, les routes MVC et ainsi de suite.

Dans notre exemple, la classe *Startup* est très simple. Dans un premier temps, on configure l'**appsettings** du projet.

```

public Startup(IHostingEnvironment env)
{
    var builder = new ConfigurationBuilder()
        .SetBasePath(env.ContentRootPath)
        .AddJsonFile("appsettings.json", optional: false, reloadOnChange: true)
        .AddJsonFile($"appsettings.{env.EnvironmentName}.json", optional: true)
        .AddEnvironmentVariables();
    Configuration = builder.Build();
}

```

Dans le template de base, le fichier **appsettings.json** permet de configurer les niveaux de log (d'où 2 fichiers différents selon l'environnement, production ou développement), mais il permet également de stocker du paramétrage supplémentaire qui peut être ensuite réutilisé dans l'application de manière ultérieure (URL pour les APIs, chaîne de connexion ...).

```

{
    "Logging": {
        "IncludeScopes": false,
        "LogLevel": {
            "Default": "Warning"
        }
    }
}

```

La classes *Startup* offre ensuite 2 méthodes importantes :

- **ConfigureServices** : configuration des services à inclure dans l'application. Dans notre exemple, nous configurons simplement MVC afin d'accéder aux APIs selon des routes URL ;

```

public void ConfigureServices(IServiceCollection services)
{
    // Add framework services.
    services.AddMvc();
}

```

- **Configure** : configuration des middlewares de l'application. Dans notre cas, on configure simplement des logger et MVC selon son paramétrage par défaut.

```

public void Configure(IApplicationBuilder app, IHostingEnvironment env, ILoggerFactory
loggerFactory)
{
    loggerFactory.AddConsole(Configuration.GetSection("Logging"));
    loggerFactory.AddDebug();

    app.UseMvc();
}

```

A partir d'ici, notre application est créée mais ne peut pas être lancée : il faut d'abord récupérer les paquets NuGet nécessaires à notre application pour fonctionner. Pour ce faire, rien de plus simple, il suffit de lancer la commande suivante dans le répertoire de l'application :

dotnet restore

L'outil en ligne de commande va ainsi s'appuyer sur le **csproj** afin de déterminer quel paquet il faut télécharger. Ce procédé est relativement nouveau depuis l'arrivée de Visual Studio 2017, puisqu'auparavant l'outil s'appuyait sur un fichier **project.json** pour le paramétrage du projet (et notamment les paquets NuGet). Il faut bien garder à l'esprit ici que l'outil évolue constamment et s'adapte rapidement aux nouveautés de la plateforme.

```

$ dotnet restore
Restoring packages for C:\Users\Christophe\Documents\_\Temp\_\Temp.csproj...
Generating MSBuild file C:\Users\Christophe\Documents\_\Temp\obj\_\Temp.csproj.nuget.g.props.
Generating MSBuild file C:\Users\Christophe\Documents\_\Temp\obj\_\Temp.csproj.nuget.g.targets.
Writing lock file to disk. Path: C:\Users\Christophe\Documents\_\Temp\obj\project.assets.json
Restore completed in 2.06 sec for C:\Users\Christophe\Documents\_\Temp\_\Temp.csproj.

NuGet Config files used:
  C:\Users\Christophe\AppData\Roaming\NuGet\NuGet.Config
  C:\Program Files (x86)\NuGet\Config\Microsoft.VisualStudio.Offline.config

Feeds used:
  https://api.nuget.org/v3/index.json
  C:\Program Files (x86)\Microsoft SDKs\NuGetPackages\
```

[6] Restauration des paquets NuGet

La commande **restore** comporte plusieurs options :

- **-s** : spécifie la source de paquet NuGet à utiliser pour la restauration ;
- **-r** : spécifie le runtime .NET Core à utiliser afin de restaurer les paquets ;
- **--packages <PACKAGES_DIRECTORY>** : spécifie le dossier vers lequel il faut installer les paquets ;
- **--disable-parallel** : permet de désactiver la restauration en parallèle des paquets NuGet ;
- **--configfile <FILE>** : permet de spécifier le fichier de configuration NuGet à utiliser ;

- **--no-cache** : désactivation du cache ;
- **--ignore-failed-sources** : traitement des erreurs de restauration comme étant des warnings ;
- **--no-dependencies** : spécifie qu'il faut ignorer les références de projet. L'outil va ainsi restaurer uniquement le projet racine ;
- **-v** : activation du mode verbose.

Enfin, afin de faire fonctionner notre application, il nous suffit de la faire fonctionner avec la commande suivante :

```
dotnet run
```

L'application est disponible selon le port indiqué :

```
$ dotnet run
Hosting environment: Production
Content root path: C:\Users\Christophe\Documents\_Temp
Now listening on: http://localhost:5000
Application started. Press Ctrl+C to shut down.
```

[7] L'application ASP.NET Core fonctionne via le serveur Kestrel

La commande **run** possède également ses propres options :

- **-c** : configuration à utiliser pour construire le projet. Par défaut, l'outil utilise **Debug** ;
- **-f** : spécifie le framework à utiliser ;
- **-p** : chemin vers le fichier de projet à utiliser.

Pour finir, une application ASP.NET Core non déployée n'a finalement que peu de sens. Très souvent, l'équipe va déployer l'application sur un IIS afin qu'elle soit disponible en production. L'utilitaire prévoit ce scénario via la commande :

```
dotnet publish
```

On peut remarquer que l'outil a généré une DLL. Afin de lancer cette DLL, il faut alors lancer la commande suivante :

```
dotnet MaDll.dll
```

Ce procédé n'est pas forcément intuitif : on s'attend plutôt à avoir un exécutable. Pour ce faire, il faut se rendre dans le fichier **csproj** du projet et rajouter les lignes suivantes :

```
<PropertyGroup>
  <OutputType>Exe</OutputType>
  <RuntimeIdentifiers>win10-x64 </RuntimeIdentifiers>
</PropertyGroup>
```

Ces lignes peuvent être ajoutées à n'importe quel **PropertyGroup**. Il suffit ensuite de lancer la même commande **publish**, en spécifiant la plateforme que l'on cible :

```
dotnet publish -r win10-x64
```

L'option **-r** indique quel runtime on cible. L'exécutable est ainsi disponible selon le chemin suivant : `.\bin\Debug\netcoreapp1.1\win10-x64`. Les options de la commande sont les suivantes :

- **-f** : indique le framework utilisé pour le déploiement ;
- **-r** : indique le runtime à utiliser ;
- **-o** : indique le chemin vers lequel le déploiement doit être généré ;

- **-c** : indique la configuration à utiliser (Release ou Debug) ;
- **--version-suffix** : permet de spécifier la valeur de la version du déploiement ;
- **-v** : active le mode verbose de la génération.

Les outils de développement avec .NET Core

Nous venons de parcourir l'utilitaire .NET Core qui permet de créer et gérer une application .NET Core (console ou ASP.NET). L'outil **dotnet** est l'outil central pour la plateforme, c'est lui qui va effectuer les opérations de *build* ou de restauration des paquets.

Cependant, le développeur peut utiliser plusieurs outils à sa disposition afin de l'aider dans son développement mais également pour garantir un confort de production respectable.

Le développeur peut utiliser un simple éditeur de texte pour modifier son code, puis peut basculer sur une console pour lancer ou relancer son application. Il existe l'éditeur Sublime Text, très populaire dans le monde de l'Open Source, qui permet de développer des applications .NET Core avec IntelliSense, si le plugin OmniSharp est installé.

Ce plugin permet de rajouter une expérience de développement .NET supplémentaire en y ajoutant des fonctionnalités bien connues des gros éditeurs : IntelliSense, Auto-complétion, GoTo Definition, Find Usages etc. OmniSharp est utilisable avec d'autres éditeurs tels que :

- Atom ;
- SublimeText ;
- Brackets – très utile pour le développement Front-End surtout ;
- Emacs ;
- Vim ;
- Visual Studio Code.

Tous ces éditeurs sont disponibles sur Windows Linux et Mac. Le développeur peut également utiliser s'il le souhaite, sur Windows, Visual Studio 2017. L'IDE de Microsoft propose un environnement de développement plus complet qu'un simple éditeur de texte et une interface plus riche. Cependant, ce qu'il faut bien comprendre avec .NET Core, c'est que Visual Studio 2017 ne fait qu'utiliser, de manière sous-jacente, les commandes de l'outil **dotnet**. Ainsi il n'y a pas de magie, et le développeur sait exactement ce que fait l'IDE.

CONCLUSION ET FUTUR DE .NET CORE

.NET Core est bien plus qu'un nouveau framework plus light que Microsoft a voulu créer : c'est une véritable plateforme avec ses nouvelles APIs, ses nouveaux SDKs et son outil de gestion **dotnet** performant et très simple d'utilisation.

N'oublions pas le point essentiel de .NET Core : l'Open Source. Tout le code est accessible sur GitHub, et libre à chacun de suivre les Pull Requests ou de faire des suggestions concernant la plateforme.

Concernant les nouveautés qui vont arriver, la version 2 preview de .NET Core va sortir au 2^{ème} trimestre 2017, apportant notamment le support de .NET Standard 2.0 et annonçant enfin la sortie officielle du très attendu SignalR Core (bibliothèque de communication en temps réel entre un serveur et un client).

Ces nouveautés vont sans nul doute être accompagnées de plusieurs nouvelles APIs et fonctionnalités sur la plateforme .NET Core. La version finale de .NET Core devrait sortir pour le 3^{ème} trimestre 2017.

Java 9 : une mouture nommée désir !

• Sylvain SAUREL
Ingénieur d'Etudes
Java / Android
sylvain.saurel@gmail.com
[https://www.ssaurel.com](http://www.ssaurel.com)

Attendue pour début 2016, la version 9 de Java devrait finalement arriver au cœur de l'été 2017 après un certain nombre de retards dûs principalement à Jigsaw, le fameux projet devant doter Java d'un système de modules performant. Ce projet devenu une utopie pour bon nombre de développeurs sera accompagné par un certain nombre d'autres nouveautés moins médiatiques mais tout aussi intéressantes que nous vous proposons de découvrir dans cet article.

Le besoin de doter la plateforme Java d'un système modulaire digne de ce nom ne date pas d'hier. En effet, le projet Jigsaw est né il y a près de dix ans, et durant ce laps de temps les échanges au sein de la communauté concernant le périmètre et son fonctionnement n'auront cessé de ralentir la trajectoire d'un projet au parcours plus que sinueux. De fait, l'intégration de Jigsaw au sein du JDK n'aura cessé d'être repoussée passant de Java 7 à Java 8 pour enfin arriver avec Java 9.

Pour cette dernière version, il aura en outre fallu repousser la date de sortie à deux reprises puisque cette mouture était initialement attendue pour début 2016 ! Si chercher un fautif pour ces retards n'apparaît pas comme constructif, on est en droit de se demander si l'engagement d'Oracle dans la plateforme Java n'est pas un peu trop léger. Surtout à l'heure où l'éditeur a par ailleurs remercié un bon nombre des évangélistes chargés de propager la bonne parole de par le monde en présentant les nouveautés de Java tout en recueillant les besoins exprimés par la communauté afin de construire le futur de la plateforme. Ainsi, s'il aura fallu attendre plus de trois ans pour que Java 9 arrive, qu'en sera-t-il pour Java 10 et Java 11 ?

En attendant d'en savoir plus sur l'accueil que la communauté réservera à Java 9 et sur l'avenir de la plateforme, concentrons-nous sur le présent et les nouveautés apportées par cette nouvelle version.

Les longues années d'attente n'auront donc pas été vaines puisqu'avec Java 9, la plateforme va enfin disposer d'un système de modules lui permettant de franchir un cap pour rester une plateforme moderne.

Qu'est-ce qu'un Module ?

Au cœur de Jigsaw, comme de tout système modulaire, se trouve la notion de module. Ainsi, un module Jigsaw est une partie d'un logiciel définissant, au sein d'un fichier module-info.java, les trois informations suivantes :

- Quel est le nom du Module ?
- Que doit-il exporter ?
- De quoi a-t-il besoin ?



La réponse à ces trois questions est donnée via le nom du module, ce qu'il exporte et enfin ce dont il a besoin (figure 1).

Dans le but évident de limiter les risques de conflits, il est préférable de suffixer le nom du module avec le package du programme de rattachement du module. Pour répondre

à la seconde question, le module doit fournir une liste de tous ses packages considérés comme public API pour les rendre utilisables par d'autres modules.

De fait, il sera impossible d'utiliser à l'extérieur du module une classe d'un package non exporté et ce même si elle est déclarée comme pu-

blique. Enfin, la réponse à la dernière question consiste à déclarer la liste des modules nécessaires au module en cours de définition.

Toutes les classes des packages exposées publiquement par ces modules deviennent alors accessibles par le module en cours de définition. En termes de visibilité des classes, Jigsaw apporte donc une évolution majeure puisque depuis la création de Java et jusqu'à Java 8, insérer un type public au sein du classpath le rendait de facto accessible par n'importe quel autre type. Jigsaw fait ainsi passer l'accessibilité des types Java de :

- Public ;
- Private ;
- Default ;
- Protected.

à :

- Public pour tout le monde qui vient lire le module (exports) ;
- Public à des modules particuliers venant lire le module (exports to) ;
- Public à toutes les classes au sein du module lui-même ;
- Private ;
- Default ;
- Protected.

La nouvelle logique associée à la visibilité introduite par Jigsaw demandera un effort et un temps d'adaptation certains aux développeurs Java.

Un JDK enfin modularisé

Comme dans tout système modulaire qui se respecte, il est impératif d'éviter les dépendances circulaires. De fait, les dépendances entre modules doivent former un graphe acyclique.

La mise en œuvre de ce principe sur le JDK aura été une lourde tâche pour les équipes d'Oracle en charge de Java. En effet, le JDK comportait jusqu'alors un grand nombre de dépendances circulaires rendant difficile sa compréhension. Le module java.base se trouve désormais à la racine du graphe de dépendances du runtime Java et est le seul module possédant uniquement des dépendances entrantes.

Chaque module créé va lire java.base de manière implicite de la même manière que chaque objet dérive de java.lang.Object actuellement. Enfin, le module java.base exporte lui aussi un certain nombre de packages tels que java.lang, java.util ou encore java.math .

Java 9 va enfin mettre à disposition des développeurs un JDK modulaire ce qui signifie qu'il sera possible de définir précisément les modules du runtime Java nécessaires à la bonne exécution d'une application. En effet, il n'y a aucun intérêt à avoir un environnement supportant Swing ou Corba pour une application réalisant uniquement des traitements batchs locaux en arrière-plan. Dans la suite de cet article, nous verrons comment créer un environnement Java sur mesure et optimisé pour la bonne exécution d'une application.

Création d'un Module

Afin de mettre en pratique les concepts théoriques présentés jusqu'ici, nous allons définir un module ayant vocation à proposer une opération de vérification des codes postaux. Ce module est défini au sein d'un fichier module-info.java :

```
module com.ssaurel.zipvalidator {
    exports com.ssaurel.zipvalidator.api;
}
```

Notre premier module se nomme com.ssaurel.zipvalidator, exporte le contenu du package com.ssaurel.zipvalidator.api et ne lit aucun autre module excepté java.base de manière implicite. Ce module va être lu par le module de vérification d'adresse com.ssaurel.addresschecker défini comme suit :

```
module com.ssaurel.addresschecker {
    exports com.ssaurel.addresschecker.api;
    requires com.ssaurel.zipvalidator;
}
```

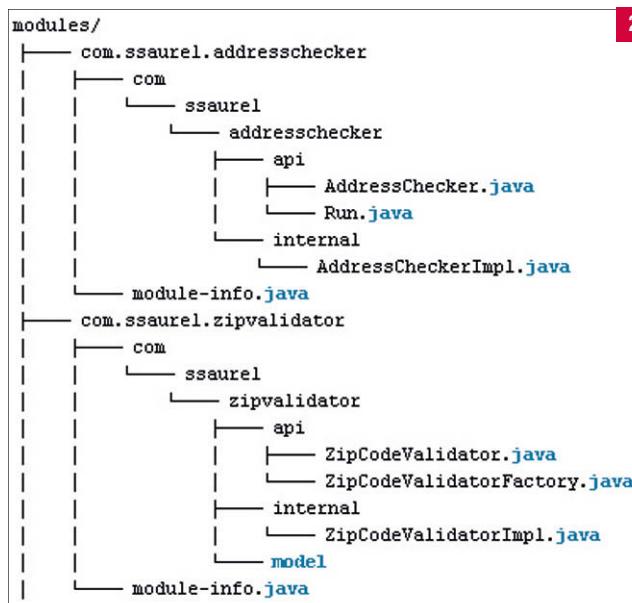
En termes d'arborescence de fichiers, l'organisation de nos modules est présentée à la [figure 2](#).

Par convention, les modules sont placés au sein de répertoires ayant le même nom que le module qu'ils contiennent. Le code source des classes de nos modules n'ayant pas grand intérêt pour le sujet nous concernant, à savoir la découverte et la prise en main de Jigsaw, il n'est pas détaillé dans cet article.

Utilisation d'un Module

Une fois les modules définis, il est temps de les compiler pour pouvoir les utiliser. Nous commençons donc par compiler le module com.ssaurel.zipvalidator puisqu'il n'a besoin d'aucun autre module pour fonctionner. La compilation est réalisée de manière classique avec l'exécutable javac et la ligne de commandes suivante :

```
javac -d com.ssaurel.zipvalidator \
$(find com.ssaurel.zipvalidator -name "*.java")
```



On notera ici l'emploi de la commande find pour lister tous les fichiers sources Java au sein du répertoire du module. Ensuite, il faut passer à la compilation du second module, qui, lui, a besoin de lire le module com.ssaurel.zipvalidator. Il faut donc donner des précisions au compilateur javac quant à notre structure de modules. Pour ce faire, Jigsaw propose l'option –modulepath (utilisable également sous la forme rac-courcie -mp). La compilation du module com.ssaurel.addresschecker se fait donc comme suit :

```
javac -modulepath . -d com.ssaurel.addresschecker \
$(find com.ssaurel.addresschecker -name "*.java")
```

On remarque clairement que l'approche retenue est similaire à celle utilisée avec le classpath. Ici, on a bien précisé à javac que les modules compilés étaient à chercher au sein du répertoire courant. Cependant, compiler les modules de manière séparée n'est clairement pas pratique et c'est pourquoi l'option -modulesourcepath a été introduite au sein de javac. La compilation de nos modules peut donc être simplifiée de la manière suivante :

```
javac -d . -modulesourcepath . $(find . -name "*.java")
```

Une fois la compilation réalisée avec succès, il est possible d'exécuter notre module pour vérifier si le code postal 13013 est correct :

```
java -mp . -m com.ssaurel.addresschecker/com.ssaurel.addresschecker.api.Run 13013
```

Pour exécuter notre programme, il faut préciser à l'exécutable Java où il doit trouver les modules compilés ainsi que le module à appeler, et, enfin, le point d'entrée du programme en y passant en entrée le ou les paramètres éventuellement attendus.

Création de Jars Modulaires

En entreprise, les programmes Java sont rarement utilisés en fournissant uniquement les classes compilées. Ils sont packagés au sein de fichiers de type Jar. Pour répondre à ce besoin essentiel, la notion de Jar modulaire a été introduite avec Jigsaw. Relativement semblable à un Jar classique, le Jar modulaire se distingue par la présence en son sein d'un fichier compilé nommé module-info.class. La création d'un Jar modulaire nommé zipvalidator.jar peut se faire avec l'exécutable jar de la manière suivante :

```
jar --create --file bin/zipvalidator.jar \
--module-version=1.0 -C com.ssaurel.zipvalidator .
```

Ici, après avoir défini le nom et l'emplacement du Jar modulaire zipvalidator.jar, on définit sa version et le nom du module à packager. Enfin, il est nécessaire de packager le module com.ssaurel.addresschecker :

```
jar --create --file=bin/addresschecker.jar --module-version=1.0 \
--main-class=com.ssaurel.addresschecker.api.Run \
-C com.ssaurel.addresschecker .
```

La classe contenant la fameuse méthode main, qui est le point d'entrée d'un Jar modulaire, n'est pas définie au sein du fichier module-info.jar comme cela avait été initialement envisagé par l'équipe en charge de Jigsaw, mais au sein du fichier Manifest présent dans le Jar comme cela est fait actuellement. Les Jars modulaires générés ayant été placés au sein du répertoire bin, l'exécution du module se fait comme suit :

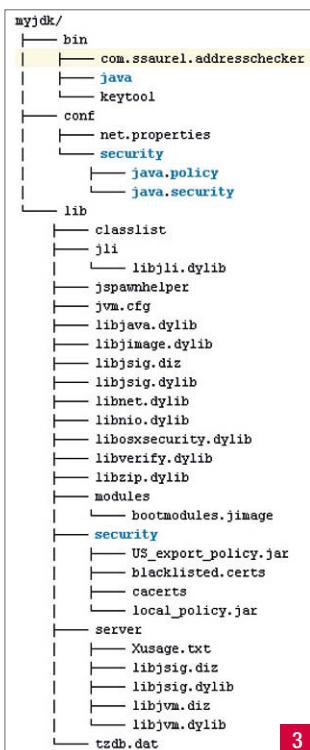
```
java -mp bin -m com.ssaurel.addresschecker 13013
```

Ici, il n'est pas nécessaire de préciser à l'exécutable Java le point d'entrée du module puisqu'il a été défini au moment de sa création dans le fichier manifest.

JVM sur mesure

Parmi les nouveaux outils relatifs à Jigsaw introduits par Java 9, il en est un essentiel qui va donner la possibilité aux développeurs de créer leurs propres distributions de JVMs sur mesure. Répondant au doux nom de `jlink`, ce nouvel outil met à profit la nature nouvellement modulaire du JDK sous Java 9. Il devient ainsi possible de définir avec précision quels modules sont nécessaires à l'exécution d'un programme et d'inclure seulement ces derniers au sein d'une JVM taillée sur mesure. La création d'une JVM sur mesure incluant nos modules se fera comme suit à l'aide de l'outil `jlink` :

```
jlink --modulepath $JAVA9_BIN/..../images/jmods/:modules/bin
--addmods com.ssaurel.addresschecker --output myjdk
```



Dans cette ligne de commande, `jlink` prend en entrée le chemin vers les modules du JDK, nos modules, les modules à inclure et enfin le dossier de sortie de la JVM personnalisée qui sera créée. Le contenu de cette dernière est présenté à la [figure 3](#).

Avec une JVM taillée sur mesure pour l'exécution de nos modules, l'environnement d'exécution a été ramené à 47 MB. En appliquant certaines options de compression offertes par `jlink`, il est même possible in fine d'obtenir un environnement ne dépassant pas les 20 MB ! Ceci va permettre de produire des applications optimisées pour des environnements embarqués où la mémoire est contrainte par exemple.

L'exécutable Java nous permet de lister les modules qui sont présents au sein d'une distribution de JVM en exécutant la commande suivante :

```
java -listmods
```

Figure 3 : Notre JVM sur mesure

Le résultat obtenu à la suite de cette commande montre bien la présence de nos deux modules ainsi que du module `java.base` en version 9.0 :

```
com.ssaurel.addresschecker
com.ssaurel.zipvalidator
java.base@9.0
```

L'exécution de notre module permettant de tester la validité d'un code postal se fera de la sorte avec notre JVM custom :

```
./myjdk/bin/com.ssaurel.addresschecker 13013
```

Des inquiétudes autour de Jigsaw ?

Ce tour d'horizon de Jigsaw nous aura permis de prendre conscience du changement que la modularisation du JDK va engendrer dans le quotidien des développeurs d'applications Java. La chaîne de build entière s'en trouve affectée et il y aura un certain nombre d'incompatibilités avec des outils comme Maven ou Gradle ainsi que les IDE qui devront s'adapter. C'est l'une des raisons qui a finalement conduit IBM et Red Hat à voter contre l'implémentation des modules Java via Jigsaw au début du

mois de Mai 2017. Pour ces deux poids lourds de l'univers Java, il existe actuellement trois soucis majeurs concernant le nouveau système de modules de la plateforme Java et son implémentation Jigsaw :

- Jigsaw n'est pas une standardisation mais plutôt une réinvention, et de nombreux cas de déploiement d'applications ne pourront être réalisés avec ;
- L'écosystème va être perturbé par Jigsaw car ce dernier va induire de nombreux changements pour des millions d'utilisateurs et de développeurs Java ;
- Jigsaw va provoquer une fragmentation de la communauté Java du fait des capacités d'interopérabilités insuffisantes avec les outils déjà existants. Le risque de se retrouver avec deux mondes distincts dans l'univers Java est bien réel : d'un côté le monde Jigsaw et de l'autre le reste du Monde avec OSGI, les modules JBoss ou encore Java EE.

Ce rejet de Jigsaw par IBM et Red Hat est un coup dur porté au système de modules tant attendu par la plateforme Java ; à l'heure actuelle, on ne sait pas encore quelles en seront les conséquences sur Java 9 tant en termes de contenu de la version que de la date de mise à disposition générale. Il faudra suivre attentivement les retours qu'apporteront les équipes d'Oracle en charge de Java dans les semaines à venir. Une seule chose est sûre, le temps en presse si l'on ne veut pas assister à un nouveau report pour Java 9. En tout état de cause, il faudra trouver un consensus pour que la plateforme soit bel et bien modularisée afin d'alléger le runtime Java, élément essentiel pour assurer sa pérennité sur le long terme.

JShell REPL

La plupart des langages compilés du marché proposent aux développeurs un outil de type REPL (Read-Eval-Print-Loop) permettant d'exécuter de manière interactive des blocs de code sans avoir à les intégrer au sein de classes ou de méthodes. Sur la plateforme Java, ce n'était jusqu'alors pas le cas et cela commençait à devenir un frein pour que Java soit utilisé comme langage d'apprentissage de la programmation informatique à des débutants notamment. Intégré tardivement à la liste des fonctionnalités potentielles de Java 9, le projet Kulla fera bien partie de la version finale sensée être mise à disposition en Juillet 2017. Désormais connu sous le nom de JShell, ce projet facilitera la programmation exploratoire en devenant l'outil idéal pour tester de nouvelles APIs rapidement et offrira aux débutants un outil simple d'accès pour découvrir les bases du langage Java. JShell accepte les blocs de codes, les variables, les méthodes, les définitions de classes mais également les imports et les expressions simples. Un des gros avantages étant que tous ces éléments sont évalués directement par JShell. Une fois ce dernier lancé au sein d'un terminal, la commande `/help` est à disposition pour lister l'ensemble des options proposées. Il est bon de souligner que l'outil propose une complétion du code ce qui s'avère très pratique à l'usage. Pour commencer avec JShell, nous allons définir une simple expression arithmétique puis imprimer son résultat à l'écran :

```
-> 3 * (4 + 5)
| Expression value is: 27
| assigned to temporary variable $1 of type int

-> System.out.println($1);
27
```

La commande `/list` permet ensuite d'accéder à l'historique des lignes de codes saisies au cours de la session JShell courante :

```
-> /list
9 : 3 * (4 + 5)
10 : System.out.println($1);
```

La définition d'une variable de type String se fait également de manière classique et il est possible de lister l'ensemble des variables définies dans la session courant via la commande /vars :

```
-> String s = "Sylvain Saurel"
| Added variable s of type String with initial value "Sylvain Saurel"

-> /vars
| int $1 = 27
| String s = "Sylvain Saurel"
```

Définir une classe avec JShell n'est pas plus compliqué :

```
-> class Pet {}
| Added class Pet

-> class Cat extends Pet {}
| Added class Cat
```

Mieux encore, JShell permet d'utiliser des APIs Java telles que Swing pour tester la construction d'une interface graphique rapidement ou réaliser un appel réseau via la classe URL de la sorte :

```
-> URL obj = new URL("https://www.ssaurel.com/blog")
| Added variable obj of type URL with initial value https://www.ssaurel.com/blog

-> URLConnection conn = obj.openConnection()
| Added variable conn of type URLConnection with initial value sun.net.www.protocol
.http.HttpURLConnection:https://www.ssaurel.com/blog

-> conn.getHeaderFields()
| Expression value is : ...
```

En mettant à disposition des développeurs Java toutes les fonctionnalités que l'on est en droit d'attendre d'un outil REPL, JShell deviendra rapidement un outil d'apprentissage et d'exploration indispensable pour tout un chacun du débutant au développeur plus aguerri. En outre, la possibilité de charger des Jars externes via la commande /classpath est une excellente idée qui permettra de découvrir rapidement et directement des APIs nouvelles au sein de JShell.

HTTP/2

La plateforme Java supporte le protocole HTTP depuis ses débuts mais la dernière version supportée, la 1.1, date de 1999 et souffre d'un certain nombre de limitations. En sus, le support proposé par la plateforme n'est pas orienté API et pas vraiment spécifique au protocole HTTP avec la fameuse classe URL. Conséquence de ce support plus que moyen, la plupart des développeurs ont recours à une bibliothèque tierce, telle que Apache HttpComponents, pour réaliser des appels HTTP au sein de leurs programmes.

Le support de HTTP/2 décidé avec la mise en place d'une API dédiée au sein de Java 9 est donc un pas important pour la plateforme et son ave-

nir à moyen terme. La nouvelle API fait table rase du passé, en abandonnant la volonté d'indépendance avec le protocole qui avait dirigé la conception de la précédente API. A contrario, la nouvelle API cible uniquement HTTP en gardant à l'esprit que HTTP/2 ne change pas fondamentalement la sémantique du protocole. Tout ceci aura permis d'obtenir une API indépendante de la version du protocole. Avec Java 9 et la nouvelle API, une requête HTTP sera réalisée et gérée de la sorte :

```
HttpResponse response = HttpRequest
    .create(new URI("https://www.ssaurel.com/blog"))
    .body(noBody())
    .GET().send();

int responseCode = response.responseCode();
String responseBody = response.body(asString());

System.out.println(responseBody);
```

En proposant une construction basée sur le design pattern Builder, la nouvelle API est beaucoup plus simple et surtout agréable à utiliser au quotidien pour les développeurs qui pourront se passer de l'utilisation de bibliothèques tierces. Une autre nouveauté sympathique offerte par l'API HTTP/2 est la possibilité de réaliser des appels asynchrones gérés dans un thread séparé en arrière-plan par la JVM :

```
HttpRequest req = HttpRequest
    .create(new URI("https://www.ssaurel.com/blog"))
    .body(noBody())
    .GET();

CompletableFuture<HttpResponse> aResp = req.sendAsync();
Thread.sleep(10);

if (!aResp.isDone()) {
    aResp.cancel(true);
    System.out.println("Echec pour obtenir une réponse rapidement ... ");
    return;
}

HttpResponse response = aResp.get();
```

Process API

Depuis sa création, la plateforme Java n'a jamais été réputée pour fournir un support digne de ce nom pour contrôler et gérer les processus systèmes. C'est un problème majeur qui a conduit les développeurs Java à mettre au point un certain nombre de hacks dépendants de la plateforme cible pour ne serait-ce que récupérer le PID du processus courant ! Il faut bien se rendre compte que ce manque criant de la plateforme mettait à mal le fameux slogan "Write once, run anywhere" créé par Sun pour son langage Java. Ainsi, en Java 8, récupérer le PID du processus courant pouvait être fait de la sorte :

```
public static void main(String[] args) throws Exception {
    Process proc = Runtime.getRuntime().exec(new String[]{"./bin/sh", "-c", "echo $PPID"});
    if (proc.waitFor() == 0) {
        InputStream in = proc.getInputStream();
```

```

int available = in.available();
byte[] outputBytes = new byte[available];
in.read(outputBytes);

String pid = new String(outputBytes);
System.out.println("PID = " + pid);
}
}

```

Nul besoin de longs discours pour convaincre tout un chacun que ce code était inutilement compliqué et bien sûr source d'erreurs. En Java 9, il devient possible de réaliser cette opération en une seule ligne avec la mise à jour de l'API Process :

```
System.out.println("PID = " + Process.getCurrentPid());
```

Outre la concision apportée par cette API, le point le plus important est qu'elle fonctionne sur toutes les plateformes. Les développeurs pourront dire adieu aux hacks utilisés rendant leurs programmes dépendants d'une plateforme. Autre exemple intéressant, l'API Process va permettre de récupérer le nom et l'état des processus tout en obtenant un certain nombre d'informations les concernant. On peut ainsi imaginer récupérer tous les processus en cours d'exécution sur le système hôte avec le code suivant :

```

import java.time.Instant;
import java.time.Duration;
import java.time.temporal.ChronoUnit;

public class ProcessAPIDemoUtil{
    public static void printProcessDetails(ProcessHandle currentProcess){
        ProcessHandle.Info currentProcessInfo = currentProcess.info();

        if (currentProcessInfo.command().orElse("").equals("")){
            return;
        }
        System.out.println("Processus PID: " + currentProcess.getId());
        System.out.println("Commande Pathname : " + currentProcessInfo.command().orElse(""));

        String[] arguments = currentProcessInfo.arguments().orElse(new String[]{});

        if (arguments.length != 0){
            System.out.print("Arguments: ");

            for(String arg : arguments){
                System.out.print(arg + " ");
            }

            System.out.println();
        }

        System.out.println("Démarré à : " + currentProcessInfo.startInstant().orElse(Instant.now()).toString());
        System.out.println("Démarré depuis : " + currentProcessInfo.totalCpuDuration().orElse(Duration.ofMillis(0)).toMillis() + "ms");
        System.out.println("Propriétaire : " + currentProcessInfo.user().orElse(""));
    }
}

```

```

public class Demo{
    public static void main(String[] args){
        // filtrage sur processus ayant une commande / on prend 5 résultats
        ProcessHandle.allProcesses()
            .filter(processHandle -> processHandle.info().command().isPresent())
            .limit(5)
            .forEach((process) ->{
                ProcessAPIDemoUtil.printProcessDetails(process);
            });
    }
}

```

Stack-Walking API

Actuellement, Java offrait un accès limité et peu performant au contenu de la stack trace pour les développeurs. Avec Java 9, les choses seront différentes du fait de l'introduction de l'API Stack-Walking. Cette dernière fournit une API standard efficace pour accéder au contenu de la stack trace, le parcourir mais également le filtrer. Le but étant de pouvoir obtenir un accès performant aux informations de la stack trace. Une question légitime que peuvent se poser certains est : pourquoi vouloir accéder aux éléments de la stack trace ? La réponse tient en trois points principaux :

- Comprendre le comportement d'une application à l'exécution ;
- Logguer un élément particulier de la stack trace pour faciliter le travail de débogage ;
- Connaître l'appelant d'une méthode particulière dans le but d'identifier la source d'une fuite mémoire.

Avant Java 9, l'API permettant d'obtenir des informations sur la stack trace était basée sur la classe StackTraceElement. Obtenir une stack trace à un instant donné se faisait en créant un objet Throwable puis en appelant la méthode getStackTrace :

```
StackTraceElement[] stackTrace = new Throwable().getStackTrace();
```

Malheureusement, cette approche était catastrophique en termes de performances puisque la JVM capturait un snapshot complet de la stack, et ce même si on ne souhaitait accéder qu'aux premiers éléments de celle-ci. En outre, il n'était pas possible d'accéder à une instance d'objet java.lang.Class de la classe ayant déclaré la méthode représentée par un élément de la stack trace. Un certain nombre de problèmes de sécurité se posaient également pour pouvoir travailler avec les éléments de la stack trace. Pour ces raisons notamment, il était temps que la plateforme Java se dote d'une API plus performante et c'est chose faite avec l'API Stack-Walking.

Au cœur de cette API, on retrouve la classe StackWalker. Dans ce qui suit, nous allons découvrir comment obtenir une instance de StackWalker et comment utiliser cette instance pour parcourir seulement quelques éléments de la stack trace. Commençons donc par un premier exemple où nous récupérons une instance de StackWalker via sa méthode statique getInstance et où nous récupérons tous les éléments de la stack trace au sein d'une liste d'objets StackFrame. Vous l'aurez compris, un élément de la stack trace est modélisé par cette dernière classe. Cela nous donne le code suivant :

```

import java.util.List;
import java.util.stream.Collectors;

```

```

public class StackWalkerDemo {
    public static void main(String[] args) {
        methodA();
    }

    public static void methodA() {
        methodB();
    }

    public static void methodB() {
        methodC();
    }

    public static void methodC() {
        StackWalker sw = StackWalker.getInstance();
        List<StackWalker.StackFrame> frames;

        frames = sw.walk(frames_ -> frames_.collect(Collectors.toList()));
        frames.forEach(System.out::println);
        System.out.println();

        long numFrames = sw.walk(frames_ -> frames_.count());
        System.out.printf("Nombre total d'éléments : %d%n%n", numFrames);

        frames = sw.walk(frames_ -> frames_.limit(2).collect(Collectors.toList()));
        frames.forEach(System.out::println);
    }
}

L'exécution de ce code produit la sortie suivante :
StackWalkerDemo.c(StackWalkerDemo.java:27)
StackWalkerDemo.b(StackWalkerDemo.java:19)
StackWalkerDemo.a(StackWalkerDemo.java:14)
StackWalkerDemo.main(StackWalkerDemo.java:9)

Nombre total d'éléments : 4

StackWalkerDemo.c(StackWalkerDemo.java:34)
StackWalkerDemo.b(StackWalkerDemo.java:19)

```

Sur le premier affichage, on constate que tous les éléments de la stack trace sont affichés alors que sur le second, on remarque que l'appel de la méthode `limit` permet de limiter le nombre d'éléments de la stack trace remontés et donc affichés.

L'objet `StackFrame` permet d'obtenir un certain nombre d'informations telles que : l'index bytecode, le nom de la classe associée à la méthode courante, le fichier source de déclaration, si la méthode est native ou non, le numéro de ligne dans le fichier source, et, enfin, le nom de la méthode. Pour obtenir ces informations, il faudra passer en paramètre de la méthode `getInstance` de l'objet `StackWalker` la constante `StackWalker.Option.RETAIN_CLASS_REFERENCE`, ce qui nous donne le code suivant :

```

public class StackWalkerDemo {
    public static void main(String[] args) {
        methodA();
    }
}

```

```

public static void methodA() {
    methodB();
}

public static void methodB() {
    methodC();
}

public static void methodC() {
    StackWalker sw = StackWalker.getInstance(StackWalker.Option.RETAIN_CLASS_REFERENCE);
    sw.forEach(f->
    {
        System.out.printf("Bytecode index: %d%n",
                           f.getByteCodeIndex());
        System.out.printf("Nom de la classe: %s%n",
                           f.getClassName());
        System.out.printf("Classe déclarante: %s%n",
                           f.getDeclaringClass());
        System.out.printf("Nom du fichier: %s%n",
                           f.getFileName());
        System.out.printf("Native ?: %b%n",
                           f.isNativeMethod());
        System.out.printf("Numéro de ligne : %d%n",
                           f.getLineNumber());
        System.out.printf("Nom de la méthode : %s%n%n",
                           f.getFileName());
    });
}
}

```

Bien entendu, l'API Stack-Walking propose des fonctionnalités bien plus poussées mais ces deux exemples auront permis de proposer un aperçu de sa mise en œuvre avec Java 9. Les développeurs trouveront rapidement un avantage à y recourir tant en termes de performance qu'en termes de simplicité d'emploi durant le développement.

Divers

Outre les nouveautés majeures présentées jusqu'ici, cette nouvelle version de Java vient avec son lot de petites évolutions visant soit à améliorer la vie des développeurs au quotidien soit à préparer les évolutions futures de la plateforme. Parmi ces nouveautés, on pourra citer la poursuite des travaux entrepris avec Java 7 dans le cadre du projet Coin. Bien entendu, les modifications ici sont moins importantes mais on pourra par exemple souligner la possibilité d'utiliser une ressource référencée par une variable marquée `final` ou juste effectivement `final` directement au sein d'un bloc `try-with-resources` sans avoir à déclarer explicitement une nouvelle variable à chaque fois. Notons également l'introduction du support des méthodes `private` au sein des interfaces. Java 8 avait rendue possible la déclaration de méthodes `default` ou `static` avec des implémentations au sein d'interfaces. Cependant, il était impossible de restreindre l'appel à ces méthodes au-delà de l'interface de déclaration. Java 9 corrige donc ce problème et les méthodes déclarées `privées` au sein d'interfaces peuvent être vues comme des méthodes de type `helper`. Toujours au rayon amélioration du confort des développeurs au quotidien, on citera l'ajout à l'API Collections de méthodes `factory` permettant de faciliter la création d'instances de collections et de maps immuables.

Au niveau de la JVM, G1 est enfin positionné comme garbage collector par défaut afin de maximiser les performances à l'exécution. La segmentation du code mis en cache en trois zones distinctes va également dans le sens d'une volonté constante d'améliorer les performances de la JVM qui sert de plateforme de base à bien d'autres langages que Java. Ces zones distinctes sont :

- JVM Internal pour le code devant toujours rester en cache ;
- Profiled code pour le code mis en cache pour une courte durée ;
- Non-Profiled code pour le code mis en cache pour une longue durée.

Au niveau du compilateur javac, les efforts d'amélioration sont également constants à chaque version de Java. Pour Java 9, le compilateur évolue pour pouvoir utiliser tous les coeurs disponibles sur la machine hôte afin d'améliorer les temps de compilation des projets Java. Enfin, il se voit également doter de nouvelles options pour donner la possibilité aux développeurs de mieux contrôler le code généré avec plus de possibilités de gestion au runtime.

Au rayon des absents majeurs, on pourra regretter le retrait de Java 9 de la JSR 354 visant à intégrer l'API Money and Currency pour doter la plateforme d'une solution standard de gestion monétaire. De même, il avait été proposé d'intégrer une API légère facilitant la manipulation de données au format JSON au sein de la plateforme Java mais cela a été reporté apparemment pour Java 10 car jugé non prioritaire par Oracle. Ces absences sont d'autant plus regrettables que les implémentations des APIs sont déjà prêtes depuis début 2016 et que Java 9 aura connu près de deux ans de retard, ce qui aurait sûrement permis de faire un effort supplémentaire pour intégrer ces APIs au périmètre de cette nouvelle mouture.

Java 10...

Alors qu'après la sortie de Java 7, Mark Reinhold, l'architecte en chef de la plateforme Java chez Oracle, annonçait des nouvelles versions de Java tous les deux ans afin d'être plus réactif aux évolutions des langages et des technologies, force est de constater qu'on est plutôt sur une sortie de Java tous les trois ans. Pire encore, chaque sortie s'accompagne de multiples reports et de changements de périmètres (toujours à la baisse malheureusement) laissant la communauté se poser des questions quant à la pérennité du langage sur le long terme. Ne voit-on d'ailleurs pas des voix s'élevaient pour annoncer que Java est le nouveau COBOL ? En attendant de savoir si les prochaines sorties de Java seront plus rapprochées avec moins de tumultes, nous pouvons d'ores et déjà nous intéresser au possible contenu de Java 10.

Les nouveautés que l'on est en droit d'attendre pour Java 10 ou même Java 11 sont désormais souvent intégrées au sein de projets aux noms souvent exotiques ou étonnantes. On pense ainsi au projet Amber annoncé en Janvier qui propose en particulier trois évolutions pour Java. La première concerne l'inférence de type avec la création de variables locales et l'introduction des mots clés var et val. Ainsi, la construction suivante :

```
List<String> list = new ArrayList<String>();
final Stream<String> stream = getStream();
```

deviendrait :

```
var list = new ArrayList<String>();
val stream = getStream();
```

La seconde concerne les enums. Le projet Amber souhaite les améliorer

en leur ajoutant de nouvelles possibilités. Une des ces évolutions consisterait à proposer un support des génériques au sein des enums. La seconde serait d'autoriser l'ajout de méthodes à des valeurs d'enums. Utilisées conjointement, ces fonctionnalités nous permettront de définir des enums avec des types génériques comme suit :

```
public enum MathematicalConstants <T> {
    E < Double > (Math.PI),
    PI < Double > (Math.E),
    ZERO < Integer > (0),
    ONE < Integer > (1),
    i < Complex > (new Complex(0, 1))
    private MathematicalConstants(T underlyingObject) {
        this.underlyingObject = underlyingObject;
    }
}
```

Enfin, la dernière piste de travail du projet Amber vise à améliorer et simplifier l'utilisation des Lambdas introduits en 2014 avec Java 8 en rendant possible l'utilisation du symbole underscore pour indiquer un paramètre inutilisé au sein d'une Lambda expression :

```
BiFunction<Integer, String, String> biss = (i, _) -> String.valueOf(i);
```

A ce projet Amber, on ajoutera le projet Vallhalla dont le but est d'offrir la possibilité aux développeurs de travailler avec des objets immuables et libres de toute référence. Ou encore, le projet XRender Graphics Pipeline ayant pour ambition de créer un nouveau pipeline graphique pour Java 2D. Enfin, le bien nommé projet Panama cherche à améliorer l'interconnexion entre le code géré par la JVM et les APIs pour des bibliothèques non managées par la JVM. Pour les instigateurs de ce projet, il est dommage que des développeurs Java se privent de bibliothèques en code natif excellentes sous prétexte que leur interconnexion avec la plateforme soit quelque peu chaotique. Ce projet visera donc à remplacer l'API JNI qui a vécu et qui présente bien des défauts pénalisant les développeurs depuis un certain nombre d'années désormais.

CONCLUSION

Lancé en 2011, Java 7 avait été une évolution plutôt qu'une révolution pour les développeurs Java avec l'introduction d'un certain nombre de nouveautés visant à améliorer le quotidien des développeurs avec notamment le projet Coin. A contrario, avec l'introduction des expressions Lambdas et des Streams, Java 8 fut une véritable révolution pour les développeurs qui ont largement adoubé cette nouvelle version en migrant massivement leurs applications. Avec la mise à disposition du tant attendu système de module Jigsaw, Java 9 s'inscrit dans la lignée de Java 8 et impactera considérablement la vie des développeurs d'applications qui devront acquérir au plus vite la logique de modules associée à ce nouveau système. Outre l'introduction d'un système modulaire, Java 9 offrira aux développeurs un certain nombre d'APIs nouvelles qui faciliteront leur vie avec par exemple l'API HTTP/2 et l'API Process.

Le but ultime étant de prouver que la plateforme Java peut rester compétitive en offrant des performances accrues au runtime et donc rassurer in fine sur la pérennité de la plateforme. Il reste désormais simplement à espérer que Java 9 sortira bien en Juillet 2017 malgré les remous autour de Jigsaw, et que Java 10 ne mettra pas aussi longtemps à sortir des cartons d'Oracle.

Retours sur Devoxx FR 2017



Eric Manuguerra
Directeur Technique,
SQLI Suisse



Pendant 3 jours, les 2890 participants au DEVOXX 2017 ont pu choisir parmi une multitude d'excellentes présentations, 220 cette année. [1]

Venus de SQLI Suisse, Ulrich Marie, Etienne Courcier et moi-même, avons pu bénéficier d'un très beau programme, avec des conférences qui portaient sur :

- Les architectures micro-services ;
- Les pratiques, outils et plateformes du DevOps ;
- L'intelligence artificielle et le Big Data ;
- Le développement front-end ;
- La sécurité, la performance ;
- L'éthique et la responsabilité des ingénieurs.

L'enthousiasme, la qualité des présentations, l'innovation et l'échange étaient une fois de plus au rendez-vous.

Cet article présente un résumé des sessions auxquelles nous avons pu assister.

Big Binouze : Le Big Data, de la récolte jusqu'à la production à grande échelle !

En partant de la question "Qui n'a jamais rêvé de trouver la meilleure bière près de chez lui ?", Mathias Kluba et Alexis Slawny ont présenté les concepts au cœur du Big Data de la collecte de données à la visualisation en passant par la transformation à l'aide des solutions de traitement distribuées Hadoop et Spark.

CQRS et Event Sourcing from scratch

Emilien Pecoul et Florent Pellet ont présenté l'architecture logique d'une application construite selon le pattern d'Event Sourcing avec CQRS. L'approche a été démontrée à partir de l'exemple d'une application "Mixter", "Twitter like". Les fonctionnalités de base (tweet et retweet) ont été codées "live" en C#. [2]

Live coding de CQRS et Event sourcing

Cette présentation était très intéressante : à la fois claire et pragmatique. Elle a permis de démystifier le pattern d'Event Sourcing et d'apprécier la puissance de l'approche, notamment pour décomposer la complexité selon l'axe lecture / écriture et accroître la performance. Le

La 6ème édition du Devoxx Paris s'est déroulée comme les années précédentes au Palais des Congrès, Porte Maillot.



1



2

pattern d'event source est particulièrement intéressant pour implémenter des applications qui gèrent des données "temporelles" basées sur des modifications historisées, comme par exemple la ligne de vie de messages Twitter, ou les écritures d'un compte en banque. La présentation est disponible ici : <https://www.slideshare.net/ouarzy/universit-eventourcing-cqrs-devvoxx-fr-2017>.

Execution de code - Mais que se passe-t-il vraiment ?

Alex Snaps, spécialiste de la manipulation de byte code, a expliqué ce qui se produit dans la machine lorsque le code s'exécute : au niveau du code machine (byte code) et du processeur.

Les caches d'instructions (L1i) et de données (L1d)

A travers des exemples en C, Java, Rust et un simulateur d'assembleur - <https://schweigi.github.io/assembler-simulator/>, A.Snaps a présenté les problèmes que les machines virtuelles adressent pour simplifier le développement et optimiser les performances.

Integration in a #Microservices Style

Abdellatif BOUCHAMA, ingénieur en intégration chez Red Hat, a présenté une chaîne DevOps (pipeline) pour la construction d'un micro-service d'intégration avec Apache Camel (support de 49 patterns d'intégration, les EIP). La chaîne DevOps est entièrement gérée par Fabric 8 de Pivotal qui intègre les outils DevOps Docker, Kubernetes, Jenkins, Gogs, Nexus et Forge en une seule solution. À travers sa présentation, Abdellatif BOUCHAMA a suggéré de ne pas réinventer la roue, mais d'utiliser plutôt une plateforme complète qui permet de construire, de tester et de déployer une architecture micro-services, telle que Fabric 8. [3]

Fabric8 : "an opinionated open source Microservices Platform based on Docker, Kubernetes and Jenkins"

La démonstration a notamment illustré les fonctions d'orchestration (scaling, répartition de conteneurs, zéro downtime) et de pipeline avec la suite d'outils Gogs, Jenkins, Nexus, Docker.

Getting started with graph database

Lauren Schaefer a montré comment réaliser une application exploitant une base de données de type "Graphe" via le service IBM Graph. IBM Graph s'appuie sur Apache Tinker Pop - <https://tinkerpop.apache.org/>.

L'application est un site de commande de photos en tirages papier. La présentation est disponible ici : <https://www.slideshare.net/LaurenHaywardSchaefer/getting-started-with-graph-databases-73890331>.

Accroissement de popularité : les bases de données orientées graphes sont de plus en plus populaires (courbe du haut)

La séance a permis de voir comment modéliser des données orientées graphe et comment effectuer des requêtes sur ces données via la console IBM.

Spring Cloud et Zipkin à la rescousse du tracing distribué

Dans une architecture distribuée, à base de micro-services, comment surveiller les échanges ? Comment évaluer les latences qui peuvent être induites par les appels réseau et comment mettre en évidence les micro-services fautifs le cas échéant ? [4]

Interface graphique de Zipkin permettant de visualiser une trace distribuée

Florian Garcia d'Ippon a effectué une démonstration de Zipkin, le système de tracing distribué développé par Twitter et les composants qui permettent de dialoguer avec la solution, notamment Spring Cloud Sleuth.

Orchestration de conteneurs : le choix des armes !

L'année passée a vu converger les fonctionnalités des moteurs de conteneurs. Dans cette université de 3h, Jean-Louis Rigau, Thomas Auffredou et Yannick Lorenzati du cabinet de consulting Xebia ont présenté un état des lieux des principaux orchestrateurs utilisés dans le cycle de vie DevOps d'une application :

- Intégration continue ;
- Déploiement ;
- “Infrastructure-as-Code” ;
- Exploitation.

Les trois orchestrateurs majeurs du moment ont été comparés :

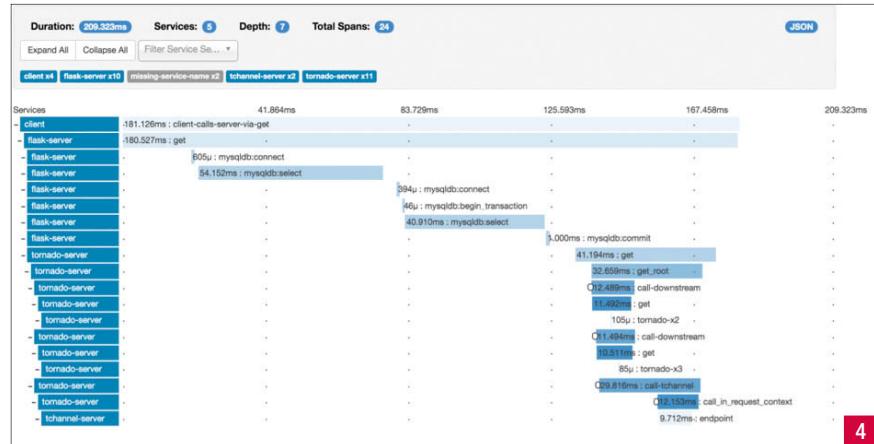
- Mesos / Marathon
- Docker Swarm
- Kubernetes

L'enjeu actuel n'est pas au niveau de la maîtrise de l'environnement système, mais plutôt au niveau de l'« orchestration », l'objectif étant de déléguer à l'orchestrateur :

- Le déploiement ;
- La gestion de résilience ;
- La surveillance pour un service en haute disponibilité.

Et plus globalement le cycle de vie des conteneurs. Les 3 solutions comparées sont très similaires dans leur approche et dans les services qu'elles proposent.

The screenshot shows the fabric8 dashboard with a sidebar containing 'Dashboard', 'Pipelines', 'Builds', 'Metrics', 'Settings', and 'Source'. The main area shows 'Environments' with 'Testing' and 'Staging' sections. Each section contains a box for 'camely-1: 1.0.1' and 'Build #1'. A red arrow points to the top right corner of the dashboard.



Elles sont robustes, relativement faciles à mettre en œuvre et à maintenir. Docker Swarm mode a l'avantage d'être simple à installer et à configurer, Mesos est polyvalent (peut orchestrer autre chose que des conteneurs Docker). Kubernetes enfin est une solution plus modulaire, un peu plus complexe à mettre en œuvre, mais qui offre davantage de fonctionnalités.

Etre productif avec JHipster

JHipster est un générateur d'applications Spring Boot + AngularJS, entièrement Open Source, qui permet de démarrer, développer et mettre en production votre projet rapidement. Largement utilisé, il possède une importante communauté de développeurs (plus de 261 contributeurs !), et de nombreuses entreprises l'ont choisi comme framework de référence.

Ces très nombreuses contributions font que JHipster possède un grand nombre d'outils, astuces et idées, qui correctement utilisés permettent de gagner du temps au début du projet. Au cours de la séance, Julien Dubois - créateur et principal développeur de JHipster - a démontré la création d'une application basée sur des micro services en partant de zéro pour aller jusqu'au déploiement en production.

JHipster est toujours aussi « magique », même si 3 ans après sa première présentation au Devoxx, le public a désormais l'habitude du projet et n'accueille plus avec autant de ravisement les fonctionnalités présentées. JHipster

n'en reste pas moins un outil à essayer absolument. Les 100 slides de la présentation sont disponibles ici <https://t.co/ZAOAn5iLB>.

RECETTES, OUTILS, API ET SERVICES POUR CONSTRUIRE SON ÉQUIPE OU SA STARTUP

« Découvrez-en moins de 30 minutes des outils, des APIs et des services pour votre projet, votre équipe, votre produit ou votre startup. » Nicolas Martignole nous livre un « Tool in Action » intéressant ; il présente les outils utilisés pour l'organisation du Devoxx et par Captain Dash.

On retiendra que les communications dans une organisation sont soit synchrones (réunion, visio conférence, Skype, ...), soit asynchrones (emails, documents, etc..) et que remplacer les emails par des plateformes de communications collaboratives comme Slack ou Hipchat apporte un gain significatif en termes de productivité, de communication et de clarté. Une autre tendance intéressante de cette année est le développement de ChatBots pour automatiser la gestion du support ou la gestion des alertes et du monitoring par exemple. On repart de la présentation avec une liste d'outils à découvrir ou à adopter. Les outils du Devoxx France :

- Google Suite (Docs, Agenda, Form)
- Dropbox-Docusign : signature électronique
- Mailgun et Mailjet
- Mailchimp
- Twilio
- Insightly
- Skype
- Appear.in
- Zoom.us
- Clever Cloud

Les outils utilisés chez Captain Dash :

- Asana (gestion des tâches)
- Google Suite
- Google Drive
- Mailjet
- Mailchimp
- OpsGenie
- Runscope
- Lastpass
- AirCall (IVR)

Keynotes : de la responsabilité des ingénieurs

Les ingénieurs, les concepteurs d'algorithmes et les programmeurs ont une responsabilité très importante dans notre société car leurs réalisations ont un impact très fort sur l'évolution de la vie sociale. L'avènement des objets connectés et l'enregistrement des données personnelles aboutissent à « une marchandisation intégrale de la vie ». Eric Sadin, écrivain et philosophe, au cours d'une keynote affectée et émouvante, a dénoncé le phénomène et affirmé son opposition. L'auditoire formé d'ingénieurs en informatique, ouvriers de ce processus, a répondu à son discours par de vibrants applaudissements.

Comprendre la loi de Conway pour réussir sa transformation DevOps

Clément Rochas (Xebia) a présenté La loi de Conway : les organisations produisent des systèmes d'information dont le design découle de leurs structures de communication.

Quand Melvin Conway a établi cette théorie en 1967, il ne se doutait sans doute pas de la puissance de son constat : il est aujourd'hui recommandé de maîtriser ce concept avant d'organiser les équipes et les aider à devenir agiles ou DevOps. A l'aide de cas concrets, il nous démontre qu'il est facile de prédire le code qui sera produit dans une organisation et donc les problèmes qui seront rencontrés, ou qu'inversement, on peut comprendre comment une entreprise est organisée à la simple lecture du code qu'elle produit. On comprend donc que

les répercussions des choix d'organisations sur les équipes IT et sur leurs livrables sont énormes. Mettre en place une culture DevOps au sein d'une organisation doit donc faire partie d'une stratégie élaborée sur l'intégralité de la chaîne de décision. Les slides sont disponibles ici : <https://fr.slideshare.net/crochas/devoxx17-reussir-sa-transformation-devops>.

Kit d'orchestration avec Docker Swarm Mode

Vincent Demeester, mainteneur du projet Docker a présenté Docker Swarm Mode, l'orchestrateur intégré à la distribution de Docker 1.12. La présentation des concepts qui permettent de gérer le parc de conteneurs était accompagnée d'une démonstration. [5]

Docker Swarm visualizer

Vincent Demeester a créé un cluster Docker en quelques minutes puis démontré les fonctions disponibles :

- Le Swarm Mode : intégré depuis la version 1.12, il remplace le Swarm Kit des versions précédentes, peu intuitif à configurer ;
- Les évolutions majeures de l'API docker ;
- Le service discovery Swarm Mode qui permet l'enregistrement automatique des conteneurs ;
- Le chiffrement par défaut des communications entre conteneurs (automatic TLS keying and signing) ;
- La possibilité de faire tourner d'autres types de containers que docker (LXD ou rocket) ;
- Le modèle déclaratif de gestion des services ;
- La gestion de la scalabilité ;
- La gestion réseau multi-host ;
- Le load balancing automatique entre conteneurs fournissant le même service ;
- Les mises à jour roulantes (rolling update) pour les déploiements « zero downtime » ;
- L'utilisation du consensus Raft pour l'élection des nœuds maîtres ;
- Les nœuds manager / worker.



Docker reprend dans Swarm Mode, ce qui a fait le succès de Kubernetes, son principal concurrent en termes d'orchestration, et va même jusqu'à en copier la sémantique (à l'exception de la notion de pod).

On note également :

- La possibilité d'exploiter la « Services Stack » pour le déploiement de grappes de conteneurs liés (équivalent de docker-compose dans un contexte de cluster) ;
- La notion de secret pour les informations sensibles. Toutes les informations sont disponibles sur le site de Docker et accompagnées d'un tutoriel détaillé : <https://docs.docker.com/engine/swarm/>.

Containers' Jungle. Docker, Rocket, RunC, LXD ... WTF ?

L'intégralité de la présentation de Nicolas De Loof (CloudBees) & Yoann Dubreuil (CloudBees) est disponible sur Youtube : <https://www.youtube.com/watch?v=zfMtF2Mo2oE>

Spring Framework 5

Stéphane Nicoll, développeur du framework Spring et de Spring Boot, a présenté les nouveautés proposées dans la version 5 :

- La nouvelle baseline (pré-requis) : Java 8, Servlet 3.1+, JMS 2+, JUnit 5, JPA 2.1+ ;
- Les améliorations de performance (en particulier pour le démarrage des applications Spring) ;
- L'approche « réactive » grâce à Spring reactive ;
- Le support du langage Kotlin.

La présentation est disponible ici : <https://speakerdeck.com/snicoll/spring-framework-5-dot-0-themes-and-trends>.

DevSecOps – Security at Devops speed with EDF

La conférence, menée par Brian Fox, Chief Technical Officer de Sonatype, a mis l'accent sur les failles de sécurité que peuvent introduire les librairies tierces (dépendances Java, .Net ou Javascript) provenant de référentiels publics, tel que Maven central. Comment surveiller et empêcher qu'une librairie, dont dépend une application, introduite de manière quasi transparente par la déclaration d'une dépendance, n'indue une faille de sécurité critique ?

La solution présentée repose sur Nexus Lifecycle : produit de Sonatype.

Au secours ma prod est sous Docker !

Le passage en production d'une architecture Microservices sous Docker entraîne des difficultés, car les activités liées à l'exploitation sont menées de façons différentes et nouvelles :

- Le monitoring des micro-services (bon fonctionnement, alertes) ;

- La surveillance des traces applicatives (logs) et techniques ;
 - La gestion des données persistantes ;
 - Reprises sur panne ;
 - Gestion des dépendances entre micro-services ;
 - Gestion des mises à jour avec Zéro downtime ;
 - Gestion du réseau entre les conteneurs ;
- Parmi les difficultés concrètes rencontrées par François Teychené a rapporté les cas suivants :
- Mon conteneur n'arrête pas de redémarrer sans que je puisse voir ses logs ;
 - Une application a démarré sur le serveur de bases de données et elle consomme trop de CPU ;
 - La machine qui héberge le conteneur Postgres est tombée, pourquoi le conteneur n'a-t-il pas redémarré sur une autre machine ?
 - Docker-compose n'est pas directement utilisable pour un orchestrateur – comment démarrer la grappe de conteneur dans le bon ordre et avec la bonne configuration réseau (cf. Docker swarm mode) ?

Première production sous Docker

Finalement, le rapport entre l'exploitant et les applications passe d'une relation de propriétaire à un animal de compagnie à une relation d'éleveur de bétail. Les services sont anonymes. En cas de panne on les supprime. De plus, l'emplacement des services sur la grille de machine n'est plus géré par un homme, mais par un orchestrateur.

La présentation est disponible : https://docs.google.com/presentation/d/12b3wR1ZRL1x5LV93hifm55FCTgp2Jvz0wqYNIdbODKw/edit#slide=id.g35f391192_00.

10 méthodes pour rendre heureux les développeurs, la 7e va vous étonner !

Une entreprise sans bons développeurs, c'est comme du code sans test unitaire : à un moment ou à un autre, ça ne fonctionne plus. Aussi devient-il impératif, sinon vital, d'attirer les meilleurs d'entre eux, et de savoir les retenir au sein de son entreprise. Si cela est simple à dire, réussir à maintenir un contexte attractif dans la durée n'est pas évident.

Comment créer un environnement de développement qui incite les développeurs à donner le meilleur d'eux-mêmes et à progresser chaque jour ? Comment assurer la cohérence des évolutions techniques tout en permettant aux équipes de tester les dernières nouveautés ? Basés sur leurs expériences, Cyril Lakech (Axa) et Romain Linsolas (Société Générale) ont partagé leurs découvertes et leurs convictions. Les 10 points qu'ils ont exposés sont les suivants : [6]

Le 7e, les petits plus, c'est ce qui peut rendre la vie des développeurs plus sympathique : le café gratuit, les salles de pause, le baby-foot ... La présentation est disponible ici : <https://linsolas.github.io/devoxx-france-2017/index.html##/>

BUILDING CLOUD NATIVE PROGRESSIVE WEB APPS - PART 1 ET 2

Matt Raible et Josh Long, deux stars de la communauté Open Source, et qui ont le mérite d'être de très bons orateurs, ont proposé un double atelier de 1H30 permettant de créer « from scratch » une « Progressive Web App » permettant d'afficher un catalogue de bières. .

- Dans la première partie de l'atelier, Josh assisté de Mat, a réalisé la couche microservice basée sur Spring Boot.
- Dans la seconde partie, Matt a développé le module front-end, basé sur le tout nouveau Angular 4 (déjà !).

Les deux compères font les clowns, amusent l'auditoire... Mais n'en restent pas moins de véritables artistes lorsqu'ils développent... c'est impressionnant de maîtrise et de fluidité.

La séance a permis de découvrir ce qu'est une Progressive en continuité du travail amorcé par Cordova et Ionic, et on associe le meilleur des deux mondes :

Changement instantané : grâce au Service Worker, l'accès au réseau est optionnel ;

Notification Push : la Notification API et la Push API proposent des notifications contextuelles et adéquates et mettent à jour l'application sans action de l'utilisateur (l'utilisateur reçoit une notification pour l'informer qu'un nouveau message est disponible dans son réseau social), même si le navigateur est fermé (grâce au Service Worker) ;

Responsive : l'utilisateur accède à l'application

à travers différents périphériques, l'application doit faire de même et suivre l'utilisateur sur son téléphone, sa tablette ou son ordinateur (voire sa montre) ;

Sécurisé : les données de l'utilisateur sont précieuses et il faut s'en occuper comme il se doit. L'utilisation de HTTPS est indispensable pour le service worker, et aussi pour les applications Web progressives (et oui cela signifie également gérer un certificat, mais https sera bientôt la norme) ;

Disponible sur l'écran d'accueil : l'application doit être proche de l'utilisateur, il doit pouvoir y accéder depuis l'écran d'accueil de son téléphone, comme une application native. C'est l'objet du manifeste des applications Web. C'est une leçon d'architecture logicielle et au final, Matt et Josh ont entre les mains une application Web progressive et résiliente, développée en ... 1h30. Ce que le Back End a gagné en simplicité, notamment grâce à Spring boot, à Java 8 et à la programmation fonctionnelle est compensé par une complexité et une sophistication croissante de l'application cliente à laquelle le framework Angular n'est pas étranger.

EN CONCLUSION

Nous avons également assisté à plusieurs autres conférences très intéressantes, telles que :

- Akka avec Java 8 : finalement c'est cool !
- Intégration continue d'images Docker ;
- Les bornes des limites : comment maintenir de la cohérence dans votre architecture microservices ;
- Les problèmes que l'on rencontre en microservices : configuration, authentification et autres joyeusetés ;
- Le changement c'est maintenant !
- Organisez la logique de votre application Web avec Redux.

L'ensemble des sessions est disponible sur la chaîne Youtube du Devoxx FR : les conférences, les Keynotes et les Tools in action. •

6

EN BREF...

- 1 INVESTIR DÈS LE RECRUTEMENT
- 2 PROPOSER UNE CARRIÈRE AUX DEVS
- 3 L'ENVIRONNEMENT DU DÉVELOPPEUR
- 4 ORGANISER SON TRAVAIL
- 5 SOFTWARE CRAFTSMANSHIP FTW
- 6 L'OUVERTURE TECHNOLOGIQUE
- 7 LES PETITS PLUS
- 8 S'ORGANISER EN COMMUNAUTÉS
- 9 CONTRIBUER À L'OPEN-SOURCE
- 10 PARTICIPER AUX EVENTS DEV(OXX)

Loan Tricot et le Machine learning



Valérie Sevel
Senior Developer
Aim Services
@valerie_sevel

Cette année, j'ai eu de nouveau la chance d'aller à Devoxx. Mais pour la première fois, j'ai pris mon billet en avance afin d'avoir un "Golden Ticket". Ces places limitées donnent la possibilité de participer à la sélection des speakers pour la conférence. On peut voter de 1 à 10 les sujets proposés. Si on n'a pas d'avis sur un sujet, on peut voter 0. C'est une expérience très prenante vu le nombre de sujets proposés ~900 ! J'ai tout noté (enfin, je crois) avec quelques 0, tout de même.

A un moment donné, je suis tombée sur un sujet proposé par un lycéen de 17 ans sur le Machine Learning. Je me suis demandée si c'était un "fake" et même si ce n'était pas cas, le niveau de la conférence Devoxx étant assez haut, est-ce qu'il serait un assez bon speaker ? Cela m'a interpellée, je ne pouvais pas juste m'abstenir en mettant 0 alors j'ai envoyé un email au jury en leur disant que même si son sujet n'était pas pris, il méritait son 1/4 d'heure de gloire si son parcours était validé.

Ce lycéen, c'est Loan Tricot. J'ai été ravie de voir que son sujet avait été sélectionné à Devoxx. J'ai donc assisté, avec curiosité, à son Tools In Action sur "Démocratiser le machine learning avec TensorFlow" https://cfp.devoxx.fr/2017/talk/XPA-6595/Democratiser_le_machine_learning_avec_TensorFlow. Le moins qu'on puisse dire, c'est que je n'ai pas été déçue ! Il a présenté son sujet avec une prestance incroyable. Et lorsque l'effet démo est survenu, il a parfaitement rebondi en prenant des questions dans le public tout en réussissant à fixer le problème et à finalement faire sa démo. Je dis : "Chapeau bas". Le but de sa présentation était de montrer que le machine learning n'est plus réservé aux chercheurs malgré le fait qu'il soit très concentré sur les mathématiques, car grâce aux nouveaux outils comme TensorFlow, ce n'est plus nécessairement le cas. Après cette performance bluffante, il a été assailli par plusieurs personnes. J'ai préféré le contacter plus tard pour lui proposer une interview.



Val : Est-ce que tu es surdoué ou autrement dit « enfant intellectuellement précoce » ?

Loan : Non, je ne suis pas surdoué. Mais c'est flatteur de demander :)

Val : Tu as fait une belle prestation de speaker à Devoxx. Comment t'es-tu préparé ?

Loan : Les vacances scolaires d'avant Devoxx m'ont servi à préparer une présentation que je croyais géniale. Ensuite le jour avant mon départ pour Paris, j'ai répété devant ma famille. Ils n'ont rien compris. L'horreur !

J'ai donc entièrement réorganisé ma présentation (en conservant quelques slides, quand même) le mercredi soir et le jour même de la présentation. J'ai répété brièvement certaines parties isolées de la présentation (introduction, explication d'une opération mathématique) mais je n'ai pas eu le temps de répéter la présentation entière et de me timer avant de passer à Devoxx. Je n'étais pas sûr de tenir les 30 minutes ! Et la démo... N'en parlons pas.

Val : Comment t'es-tu senti avant-pendant-après ?

Loan : Avant, progressivement stressé. Surtout quand je suis monté sur scène. On m'a équipé du micro et je me sentais alors très glamour, cela me faisait aussi penser à TED. La salle était encore vide à ce moment et j'avais peur qu'elle le reste, finalement elle s'est remplie pendant mon attente.

Pendant, complètement différent ! C'est très surréel, je parlais presque plus fluidement que d'habitude (ce qui a écrasé ma peur d'articuler trop

vite). Devant les 200 participants j'expliquais un sujet que je connais très bien, je ne pensais pas à mon temps (que je ne pouvais pas prédire!). Après, je n'ai pas vraiment eu le temps de redescendre : Quicksign m'a approché, puis SII pour faire des présentations, et d'autres pour poser des questions... J'ai eu une conversation très longue et intéressante avec quelqu'un qui veut faire une thèse dans une branche du machine learning. J'ai ensuite participé au meet & greet.

Val : Comment t'es-tu intéressé au Machine Learning ?

Loan : C'est une histoire que j'ai souvent racontée, surtout en conférence parce qu'elle est un formidable outil ice breaker qui lance la conversation. J'étais en première. Les lycées proposent dans la filière générale ce qu'ils appellent des TPE – ou Travaux Personnels Encadrés. Les lycéens choisissent un sujet sur lequel ils vont travailler l'année durant ; le projet se termine par une présentation notée pour le bac.

Mon premier sujet était l'aquaponie : c'est une méthode d'agriculture dont le concept est de nourrir les plantes avec des excréments de poissons. Autant dire que ça n'intéressait pas trop un adolescent. Pendant les premières vacances de l'année, j'ai réfléchi et ai repensé à un article qui décrivait une forme d'intelligence artificielle appelée réseaux de neurones ; elle était soi-disant incomprise, même par les chercheurs qui travaillaient dessus. Bien sûr, j'avais mal compris, ou ma mémoire était mauvaise. Mais mon arrogance m'a porté à en faire mon sujet de TPE. Pendant l'année de première, j'ai donc dû apprendre le python et beaucoup de mathématiques. J'ai suivi un cours du

MIT en ligne, lu le livre neural networks and deep learning de Michael Nielsen. Tout ça pour comprendre et coder un réseau de neurones. Depuis, j'ai participé dans une compétition sur la plateforme Kaggle pour améliorer ma compréhension et ai suivi un cours en ligne de Stanford. J'ai aussi exploré un certain nombre de publications scientifiques (qui sont les réels lanceurs dans le sujet, une grande partie sont relativement approchables).

C'est ensuite que j'ai eu la chance de faire un stage chez AODocs, durant l'été de ma première. C'était certainement une accélération (et des vacances d'été spéciales !) dans mon intérêt et mon apprentissage du code et du machine learning. J'ai appris à travailler dans le Cloud, et surtout à travailler avec des données qui ne sont pas parfaitement propres comme j'en avais l'habitude (les jeux de données pour la recherche scientifique sont souvent trop parfaits). Super expérience !

Val : Quelle est ta première expérience avec le code ?

Loan : Mon père est DSI d'une société de pharma. Il a essayé de m'intéresser à l'informatique sans succès quand j'étais plus jeune avec Scratch. Il ne m'a pas lancé dans le machine learning, mais c'est grâce à ses connaissances que j'ai pu faire mon stage à AODocs.

Val : Est-ce que tu as conscience que tu peux jouer un rôle modèle pour ta génération ? J'ai 2 grands ados. Bien que je les ai emmenés à Devoxx4Kids quand ils étaient plus jeunes, ils ne se sont pas lancés seuls. Ils restent de gros "gamer".

Loan : Je n'y ai pas trop pensé, mais ce serait génial ! Avoir une carte graphique de "gamer" m'a permis de me lancer dans le machine learning. La carte graphique est nécessaire pour paralléliser les calculs, c'est un heureux hasard que je l'ai achetée avant de m'y intéresser ! Ce qui prend me prend 1h sur un processeur, prend quelques minutes sur une carte graphique.

Val : J'ai vu sur ton site que tu avais monté un "Hack Club" ? Ce genre de club est très courant aux US, as-tu vécu là-bas ?

Loan : Je n'ai jamais vécu aux US. J'ai passé 2 ans en Angleterre, c'est là que j'ai appris l'anglais. Hack Club est une non-profit menée par Zach Latta aux US, lancée il y a environ 2-3 ans. Pendant mon stage à AODocs, j'ai découvert un groupe d'adolescents de San Francisco qui se lançaient dans des projets de startups, et ça me faisait rêver. Zach Latta fait partie de ce groupe. Il a gagné un Thiel



Fellowship qui lui garantit \$100k sur 2 ans et un superbe réseau d'entrepreneurs, et l'a utilisé pour agrandir sa non-profit Hack Club. C'est en fait un réseau de clubs d'informatique, répandu surtout dans les US, mais un peu dans d'autres pays. Il n'y en a qu'un seul en France (le mien). Quand je l'ai découvert, j'ai postulé à travers son site Web. Il m'a interviewé, et j'ai pu lancer mon club avec ses ressources et son accord. On est 7 ou 8 dans le club ; on se réunit tous les mardis dans une salle qu'on emprunte au lycée. Aujourd'hui on travaille sur le site Hack Club Magendie (notre lycée), et un membre va bientôt mettre en ligne une version rénovée du site Web de notre lycée. D'autres projets sont par exemple de coder le jeu de pong.

Val : Que dirais-tu pour motiver les jeunes à s'inscrire dans la même démarche ?

Loan : Il n'y a vraiment qu'à se lancer ! Si vous ne parlez pas anglais, contactez-moi et je serai très heureux d'aider et d'agir en intermédiaire avec Hack Club. N'importe quel lycéen qui code est le bienvenu. Vous pouvez m'envoyer un message par Twitter (@loantricot), me poser des questions par mail (ltricot@gmail.com), proposer un projet commun...

Val : Qu'est-ce qui t'a poussé à postuler à Devoxx ?

Loan : En décembre 2016, je suis allé à la conférence NIPS de machine learning avec un ami, Simon Dupouy. On était facilement les plus jeunes, des lycéens entourés de 6000 chercheurs ! Un des speakers invités, Drew Purves, nous a mentionné durant son keynote, faisant référence à notre âge. Cela a mené à plusieurs rencontres, notamment

avec James Weaver, qui m'a introduit aux organisateurs de Devoxx et m'a encouragé à y postuler. Merci James ! NIPS était mémorable.

Val : Tu passes le bac en juin. Quels sont tes plans pour l'an prochain ?

Loan : J'ai été admis à Polytechnique (France) pour faire partie de la première promotion de leur nouveau Bachelor de l'X.

Val : Peux-tu expliquer les grandes lignes de ce que tu as présenté à Devoxx ?

Loan : Ma présentation avait pour but de démystifier le machine learning pour les développeurs. J'ai donc montré à travers un exemple comment n'importe qui peut utiliser des méthodes de machine learning pour résoudre des problèmes, qui il y a 10 ans, nécessitaient un doctorat.

Le machine learning sert souvent à classifier des images ; dire qu'une image est une photo d'un chien, d'une voiture, d'un téléphone... Il fait cela en construisant des représentations de l'image progressivement plus abstraites. Au début il y a les pixels, puis ceux-ci s'assemblent en traits et en angles, puis en pattes et en roues, et enfin en chiens et en voitures. Ces représentations sont construites avec un assemblage d'opérations mathématiques paramétrées. Le machine learning sert à trouver ces paramètres.

Pour conclure, je trouve que c'est une belle rencontre et cela véhicule un message positif. Il faut essayer de réaliser ses rêves et le clap de fin pourrait être "JUST DO IT".

J'aurai le plaisir d'entamer de nouvelles discussions avec Loan car il devrait venir à Genève en juin pour rejouer sa présentation au GenevaJUG.

Un dernier mot : "Un Golden Ticket, c'est un peu comme un calendrier de l'avent". Au lieu d'ouvrir des fenêtres, on découvre les nouveaux thèmes presque tous les jours et ça permet d'attendre avec bonheur le jour J. J'ai un Mea Culpa, j'avoue que j'ai moins bien noté les sujets arrivés au dernier moment et j'ai aussi donné de très bonnes notes aux speakers qui m'avaient beaucoup plu les années précédentes.

Encore une fois, les organisateurs de Devoxx ont fait un travail remarquable, le niveau est toujours aussi haut. C'est passionnant, tout en étant "fun". En parlant de "fun", un autre coup de cœur a été la conférence de Matt Raible et Josh Long qui ont fait de 2 heures de Live Coding un véritable show sur le sujet "Building Cloud Native Progressive Web Apps" (https://cfp.devoxx.fr/2017/talk/XBZ-1299/Building_Cloud_Native_Progressive_Web_Apps_-_Part_1). Je vous laisse découvrir tout cela sur YouTube !

Mon expérience sur Devoxx 2017

- Mikael Kassighian, Architecte Technique chez Pictet.
- Arnaud de Giuli, Entreprise Solution Architect chez Cross.

D'où vient cette joie d'aller à Devoxx ? De se trouver enfermés dans ces salles remplies pour l'essentiel de mâles aux codes vestimentaires identiques (basket, jeans, t-shirt, sac à dos) ? De n'avoir le soleil que par quelques fenêtres perdues au hasard des allées ? De manger chaque midi un sandwich assis sur une marche d'escalier et boire un café filtre tiré d'un thermos géant ? C'est que dans chacun de ces développeurs, Lead tech, architecte et autres gradés, existe une quête. Celle-ci bien entendu varie. On rencontre ceux qui cherchent à Devoxx, l'Imprimatur, cette idée que nous sommes à jour techniquement, que nous sommes toujours dans le mouvement, que le train n'est pas passé, que nous marchons sur le bon chemin. Se trouvent aussi ceux qui, tels des pêcheurs, cherchent la nouvelle technologie qui améliorera leur système ; ils jettent le filet sans savoir précisément ce qu'ils ramèneront. D'autres enfin y trouveront le champ lexical à la mode pour cramer durant la pause déjeuner.

Mais si nous allions à Devoxx pour une autre raison ? Et si nous allions à Devoxx, tel un investisseur pour y gagner de l'argent ? Nous vous proposons dans cet exercice ludique de changer de perspective. On raconte que Warren Buffett n'investit que sur des secteurs économiques qu'il comprend. Où en était l'action Google quand nous avions compris que c'était "le moteur de recherche" ou l'action Amazon lorsque vous avez saisi la force des services S3. Nous partirons ici de l'hypothèse (et elle peut être contestée) que ce qui est mis en avant par Devoxx reflète les tendances à venir, et en bons investisseurs avertis que nous sommes, nous savons qu'anticiper les technologies à venir est le moyen le plus propice pour avoir un rendement à gros levier.

Microservices

Conférences

Les bornes des limites : comment maintenir de la cohérence dans votre architecture microservices / Clément Delafargue

Les problèmes que l'on rencontre en micro-service : configuration, authentification et autre joyeuseté / Quentin Adam

Les microservices incitent à réduire le couplage tout en poussant chacun à penser ses services en termes de responsabilités. Mais les problèmes n'ont pas disparu pour autant ! Comme pour les applications Web où nous avons repoussé la problématique de session utilisateur vers les navigateurs ; dans les microservices, nous reportons beaucoup de problèmes vers les systèmes qui nous appellent.

Et les talks nous ont montré que cette prise de conscience est réelle : nous avons passé l'étape de la découverte et avons maintenant du recul

dessus. Ce retour d'expérience est marqué par trois points : Nous avons tout d'abord des conditions à respecter avant de faire des microservices. S'il faut un formulaire et deux jours d'attente pour une VM, oubliez les microservices ! Si vous n'avez pas du monitoring et du logging avancés, vous lutterez pour comprendre vos bugs, oubliez les microservices ! Si vos déploiements se font à la main et passent par un processus en 4 étapes de validation, oubliez les microservices ou changez d'entreprise ! Deuxièmement, de nouveaux problèmes vont apparaître. Une mauvaise définition de la frontière des microservices, reléguant les appels intra-jmv à des appels RPC. Ce peut être un surdosage d'échanges HTTP (l'usage d'un système orienté message permet de pallier ce genre d'excès). Il peut y avoir des problèmes de sérialisations et d'expression des contrats interservices. Enfin il faut être conscient des problèmes qui ont été simplement repoussés. Nous parlons ici du transactionnel, de la fraîcheur des données, de la concurrence. Ajoutez à cela la complexité algorithmique qui n'est pas résolue mais déplacée vers l'architecture logicielle. Mais au-delà du buzzword, ne serait-ce pas du SOA (Service Oriented Architecture) maquillé ? Dans nos systèmes de plus en plus complexes, nous avons besoin d'approches architecturales fiables et SOA semblait répondre à ce besoin. Mais suite aux difficultés de mise en place, on a assisté à l'enterrement un peu rapide du branding SOA sans solution de remplacement. Les micro-services ressemblent à l'évolution que nous attendions (et non à une révolution). Les micro-services sont la marche actuelle de nos architectures, ce qui justifie d'investir dedans (notamment à travers des solutions conteneurs par exemple).

Java 9..10 et Scala en muse

Conférences

Java 9 modulo les modules/Jean-Michel Doudoux, Pattern Matching en Java (10 ?)/ Remi Forax

Hands-on

Scala class, bien démarrer avec Scala

Jigsaw s'annonce comme étant le gros morceau de Java 9, avec la mise en oeuvre d'un système de module permettant d'isoler notre code autrement que par les étiquettes de visibilité sur classe. Il ne serait pas infondé de rétorquer que OSGI offrait déjà ce type de possibilités, cependant cette approche n'a pas créé l'adhésion de l'ensemble de la communauté.

Mais peut-être que ce qu'il y a de plus révolutionnaire, bien que moins mis en avant, soit la non-rétrocompatibilité assurée entre versions Java. En effet, du code pourra être retiré du sdk sous réserve d'avoir été annoté "deprecated" dans la version précédente, annonçant à l'avenir quelques surprises au runtime. En outre, la JVM ne pourra supporter à la compilation que la version courante et les trois versions précédentes. Ce choix, somme toute courageux d'Oracle, va permettre à Java de ne pas porter éternellement "les fautes" du passé.

La vitalité de Java n'est plus à remettre en question, pour autant, lorsque l'on sait que le caractère '_' va devenir réservé en 9 et que, la 10 nous annonce la mise en place du pattern matching, alors oui nous pouvons le dire : Scala est devenu la muse de Java.

Spring 5 Tout va bien

Conférence

Spring Framework 5.0/Stéphane Nicoll

La nouvelle génération de Spring framework

s'annonce fructueuse. Désormais pour en profiter il faudra s'acquitter au minimum d'une version 8 ou supérieure du JDK ; y sera intégré le support de JEE7 (JEE8 est prévu pour la 5.1) à savoir Servlet 3.1 et plus, JMS 2.0, JPA2.1 et JUnit5. L'accent a été mis sur l'amélioration des performances et de l'usage mémoire, et non sur l'amélioration du scan du classpath, lui-même étant déjà très rapide (moins de 100ms pour 5000 classes dont 200 bean). Java 9 n'offrant pas, contrairement à ce qui avait été annoncé, de webclient HTTP /2, Spring 5 propose le sien, avec une API réactive et une composition d'appels simplifiée.

Le plus gros morceau se situe dans la partie réactive. La dimension réactive n'augmentera pas les performances de nos services Web, mais offrira une meilleure gestion de la scalabilité. Avec la Reactive Streams API et leur implémentation "Mono" et "Flux", la logique impérative disparaît au profit d'une démarche type pipeline et opérateurs. Avec de telles nouvelles interfaces, Spring veut nous cacher la complexité d'une programmation réactive de bout en bout de la chaîne de nos appels. Aujourd'hui il est impossible avec du JTA de faire du réactif car il n'existe pas de driver associé.

Enfin cerise sur le gâteau, Spring 5 va supporter le langage Kotlin, profitant ainsi au maximum de l'inférence de type et sa syntaxe allégée. De nombreuses extensions sont prévus dont Spring Data et Spring Boot.

JEE8, quel avenir pour les standard ? Une simple voiture balai ?

Conférence

Java EE 8 ? Ah ben non peut-être/ David Delabasse

JEE, c'est un peu comme ces séries dans lesquelles les premières saisons nous ont passionnés mais qu'on a fini par délaissé, tout en espérant secrètement un reboot capable de reproduire en nous l'effervescence des premiers jours. Laissons ici nos préjugés, car nous devons savoir si oui ou non nous devons investir dans les serveurs applicatifs lourds ?

Bien entendu, nous avons de bonnes choses ; JSON-B pour le binding JSON qui se veut le pendant de JAXB pour le XML sans la génération de code à partir d'un schéma. L'API élégante en elle-même s'inscrit avant tout comme un pont vers les solutions existantes. Une des briques importantes est JAXRS-2.1 une API responsive, consistante, de belle écriture, qui mérite de toute façon de s'y attarder. Est-ce que ces nouveautés pousseront les dévelop-

peurs Java qui ont quitté les serveurs d'applications à y revenir ? Malheureusement comme l'a très bien dit Magnus Carlsen "Too weak to slow". JEE8 arrive trop tard, sans avoir un marqueur innovant pour nous pousser à abandonner des solutions plus légères. Nous pouvons cependant parier qu'au regard de cette modernisation, l'upgrade vers la version 8 est pleinement justifié pour ceux qui sont sous JEE7.

Ethereum

Hands-on

Développer une application de pari en ligne avec la blockchain Ethereum/F. Kha/B. Fontaine. Si pour la plupart d'entre nous la blockchain reste un sujet éloigné, dont le seul intérêt est d'étaler notre science sur les clés de hash et le cryptage asymétrique, l'Ether via les smart contrat va nous faire changer d'avis, en permettant au développeur d'être enfin un acteur au sein de la blockchain.

Plus que d'assurer le transfert sécurisé d'un portefeuille à un autre comme le bitcoin, le smart contrat écrit en viper (proche de python) ou solidity (proche de javascript) permet d'inventer tout mécanisme de transfert d'actif. Sous condition d'un oracle, c'est-à-dire d'une source de données extérieure, des transferts de monnaie sécurisés par la chaîne peuvent se mettre en place, automatiquement. Par exemple, il est possible de programmer un contrat d'assurance liant un agriculteur à un oracle qui pourrait être Météo France, qui en cas de sécheresse sur plus de 7 jours, lui garantit réparation sous forme de compensation financière, le tout de façon autonome. Ou encore et c'était l'exercice du hands-on, reproduire dans un smart contrat la souscription et le versement des gains sur des paris hippiques. Les possibilités sont multiples et pourraient même affecter les bourses. Sous Ethereum, il est possible de tester en local son smart contrat, ou dans des environnements de pré-production ou des mineurs bénévoles travaillent à l'exécution du code. Cependant, coder un smart contract reste un exercice complexe car il faut sécuriser le code tout en le rendant public, optimiser son coût (se mesure en gaz), et anticiper le versionning. Il y a fort à parier que l'Ether eclipse à terme le Bitcoin.

Vertx/rxjava ou reactive programming?

Conférence

Application réactives avec Eclipse Vert.x / Julien Ponge, Julien Viet

RxJava, RxJava 2, Reactor: Etat de l'art des Reactive stream en Java / David Wursteisen

Microservices réactifs avec Eclipse Vert.x et Kubernetes / Clément Escoffier

Nous en entendions parler depuis plusieurs mois mais nous avions juste retenu que l'on pouvait facilement faire un petit serveur Web avec... C'était une erreur !

Même si ses contributeurs le présentent comme un toolkit, nous y voyons les débuts d'une nouvelle stack applicative.

En effet, aujourd'hui nous avons besoin de reactive programming, de solution light à des fins de simplification pour la scalabilité, de programmation fonctionnelle, etc. Si les frameworks traditionnels l'ont bien compris et s'appliquent à aller vers ces nouveaux paradigmes, Vertx fait table rase : c'est un framework qui a été pensé pour répondre à ces besoins dès sa conception.

Jusqu'à présent, dans le monde Java, nous avions :

- JEE et Spring: tous les deux pensés et adaptés pour faire des architectures lasagne ou en couche ;
- OSGi : pensé pour faire des architectures cannelloni ou component ;
- Des frameworks plus smart comme play framework, adaptés pour le DIY.

Mais avec les besoins actuels, nous avons souvent l'impression que nous tordons ces frameworks pour y répondre.

Comme Spring à son époque a répondu au problème de couplage fort des EJB avec l'injection de dépendances, Vertx apporte des solutions aux besoins modernes d'architecture que nous nommerons ravioli, service ou stream. Ainsi l'utilisation du reactive programming, le changement de paradigme vers la programmation fonctionnelle, le fait d'être light pour développer des micro-services font partie des concepts qui ont guidé la création de ce toolkit. Ainsi, même si, étant dans notre zone de confort, nous sommes productifs avec nos stacks préférées, elles collent de moins en moins aux exigences d'aujourd'hui. Bien évidemment, elles l'ont bien compris et s'adaptent (micro-profiles/swarm, spring boot, etc.) mais plutôt que de ré-animer une énième fois le patient n'est-il pas bon parfois de repartir d'une feuille blanche ?

Vertx nous semble être un candidat parfait pour ce nouveau départ !

Gradle dans l'innovation, Maven dans le contentement.

Conférence

Des builds incroyablement rapides avec Gradle 4 / Cédric Champeau

Maven était présent, quoi que nous puissions en dire, tel une force tranquille, un navire amiral qui trace son sillon. Non ! vous ne verrez pas Maven directement au sein des universités, ou autres rencontres nocturnes, mais vous retrouverez Maven caché dans les "Hands-on", ces lieux où en quelques heures, votre ordinateur portable en main, vous programmez sous la supervision d'experts dans un domaine. Et là, sans surprise, les formateurs vous préparent un projet, porté quelquefois par une image docker, mais pratiquement à chaque fois par un pom.xml.

En face, Gradle open source, porté par la société Gradle Inc. Pour Gradle, notre build doit avoir la même attention que notre code, c'est-à-dire nous devons pouvoir le monitorer, le profiler et surtout l'optimiser.

La première chose à faire est de lancer le Gradle deamon. Le temps de build peut être divisé par 3, profitant allègrement du JIT de la JVM (eh oui, Gradle est écrit en Java !).

Pour la partie monitoring, grâce au Gradle scan, nous avons un reporting nous donnant la liste des tâches exécutées et leurs durées, les statistiques GC et surtout des propositions d'optimisation du build.

Nous ajoutons à cela les incremental build. Maven nous a donné de mauvaises habitudes avec le "mvn clean install" ; en effet un petit changement induit de rejouer le cycle complet du build. Avec l'incremental build, Gradle sait, en fonction du changement, se concentrer uniquement sur les tâches impactées.

Depuis la 3.4, Gradle permet de distinguer le compile classpath du runtime classpath, à ne pas confondre avec les scopes compile et runtime de Maven. Le compile classpath représente les dépendances nécessaires à la compilation et le runtime classpath les dépendances propres aux détails d'implémentation de ma classe. Le C++ nous aide à mieux comprendre cette distinction. Ainsi le fichier .h, par ses dépendances et ses signatures de méthode, définit ce qui est nécessaire au compile classpath, et les dépendances présentes seulement dans le .cpp : le runtime classpath.

Pour exprimer cette différence dans la config, Gradle met à disposition les mots clé "api" pour une dépendance compile et "implementation" pour une dépendance runtime. Enfin le build cache (à partir de la version 3.5) permet au sein d'une équipe, essentiellement pour un projet de grande ou moyenne taille de réduire de manière phénoménale le temps de build (passé de 2mn à 15s). Ainsi, plutôt que de compiler les sources sur notre poste, Gradle va chercher s'il

existe une version déjà buildée de notre source. Une idée serait d'activer cette option sur le CI pour qu'il en devienne la référence. Face à une concurrence qui ne propose plus rien, Gradle innove à tout-va, avec pour objectif de réduire toujours plus la boucle de feedback. Oui, nous irons vers le "buy".

Keycloak

Conférence

Mettre en place sa sécurité et sa gestion d'identité en 2017 / Sébastien Blanc

Dans les applications traditionnelles et concernant la sécurité, nous rencontrons deux cas de figure chez nos clients : ceux qui la gèrent et ceux qui nous demandent de la gérer...

Pour ceux qui la gèrent la solution est facile : il faut s'intégrer et cela fait partie de notre métier (merci middleware ;-).

Par contre pour ceux qui nous demandent de la gérer... Même si nous nous devons d'avoir des connaissances en la matière, la sécurité n'est pas notre cœur de métier, et les bonnes pratiques nous poussent à ne pas implémenter la sécurité directement dans nos applications. Aussi on propose à nos clients une solution tierce que l'on intègre dans notre application.

Parmi les choix que nous avons eu à faire, nous connaissons :

- CAS (open source) : l'argent économisé dans le coût de licence était perdu dans le temps que nous passions à le configurer (la sécurité n'est pas notre métier...);
- Crowd (atlassian): parfait pour nous car simple mais souvent des clients réfractaires car propriétaires et pas celui recommandé par les experts en sécurité ;

- Forgerocks (ancien sun microsystem) : la Rolls Royce mais très très très compliqué : si vous pouvez répondre aux questions lors de l'installation vous aurez droit à notre plus profond respect en termes de sécurité.

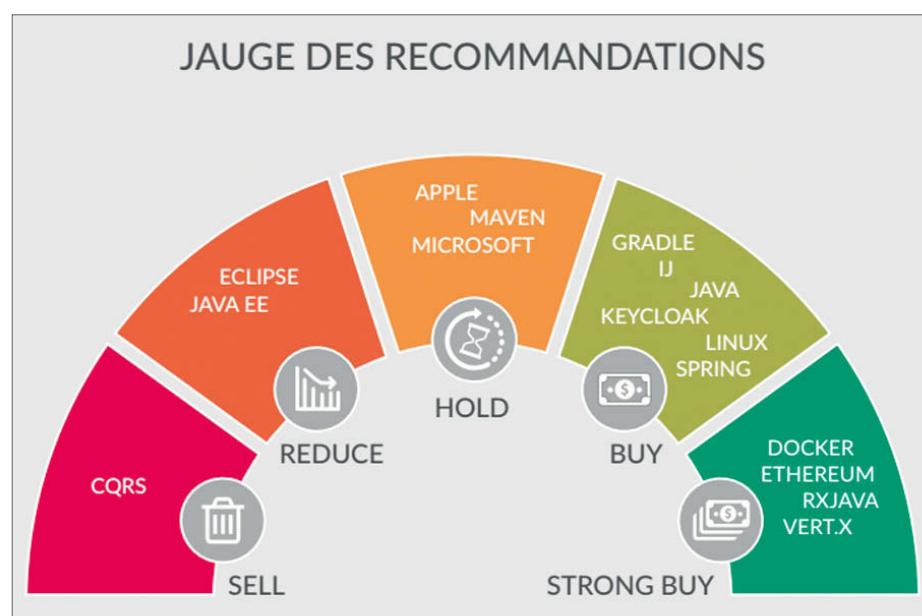
- Simple LDAP : simple mais ne répond pas aux problématiques de fédération qui sont de plus en plus présentes.

Avec Keycloak nous avons une solution simple à configurer (et qui ne me coûte pas des jours de configuration), gratuite, moderne qui permet de poser les bases chez les clients qui n'ont rien sans avoir besoin d'une expertise.

Apple peut être un peu surcoté, Windows discret, Linux en force

Qu'il est difficile de parler d'Apple sereinement. A Devoxx, l'énorme majorité des conférenciers utilisent un Mac. Si vous êtes loin de cet univers, vous aurez peut-être l'impression d'être mis sur la touche. Mais comme souvent, il est facile de se laisser tromper par les apparences. C'est dans les hands-on que s'exprime la réalité du terrain. Là, les machines des participants s'ouvrent, et oh, surprise ! bien peu de Mac au regard des Windows et autre Linux. Cependant, Linux peut être considéré comme le grand gagnant. D'une part parce que Linux est la plateforme idéale pour tout ce qui est container, et c'est le mood en ce moment, d'autre part Linux est la seule plateforme qui a eu le droit à une conférence de plus de deux heures (Linux est ton meilleur ami (Pierre Antoine Grégoire - Olivier Robert - Nicolas Helleringer)

•



Low Code : nouvelle tendance de la programmation ou simple mode ?



François Tonic

Depuis l'apparition de l'informatique, la programmation a toujours été une question sensible : aller vite et pour le moins cher possible. Et l'utilisateur veut son application pour hier, et qu'elle fonctionne parfaitement, qu'elle soit belle comme les autres, et qu'elle fasse exactement ce qu'il en attend. Bref, le développeur doit faire des miracles. Et bien entendu, la moindre faiblesse, et ce sera la faute du développeur.

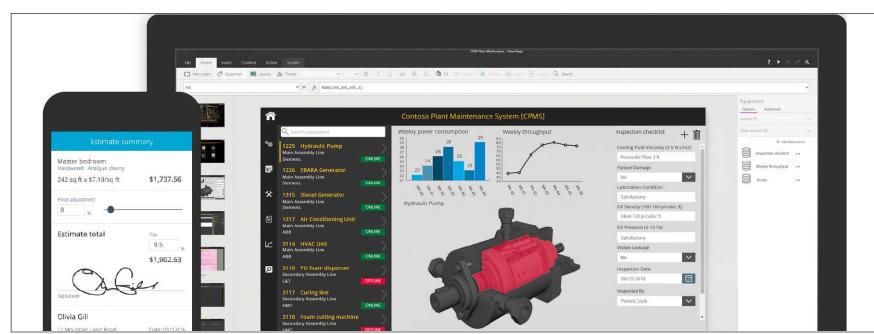
Vouloir simplifier le développement et proposer des environnements pour concevoir très rapidement un logiciel n'est donc pas une nouveauté. Cela me rappelle les débats à l'époque de VB 3 et les composants d'interfaces que l'on pouvait utiliser pour nous simplifier la vie. « La programmation Lego » disaient les puristes. Nous pouvons voir l'idée du Low Code comme une version modernisée de cette programmation simplifiée : on assemble, on paramètre mais il y a toujours du code. En préparant cet article, je m'aperçois que le low code n'est pas forcément l'ami des départements informatique. Car par définition le low code permet à tout le monde, ou presque, de « coder » une fonction, une application. Et donc, on pourrait vouloir « développer » dans son coin sans que les équipes techniques n'interviennent. On retrouve les débats autour du shadow IT, cette informatique gérée par les utilisateurs et les équipes, en dehors du département informatique car certaines demandes prennent trop de temps ou ne sont jamais traitées. Toujours cette idée d'aller vite. Mais peut-on raisonnablement comparer des solutions low code avec les IDE ultra complets tels que Visual Studio ou Eclipse ? La logique de ces outils n'est pas du tout la même et surtout, il ne sera pas pertinent pour les développements de type low code. En low code, les outils se veulent plus simples et dépouillés de nombreuses fonctions. Mais le low code peut répondre à des besoins utilisateurs, à des applications orientées données / métiers. Ces solutions peuvent générer des apps pour le Web, le Cloud, le desktop et le mobile.

« Le low code n'est pas le no-code »

Le low code peut répondre à des besoins utilisateurs, à des applications orientées données / métiers. Ces solutions peuvent générer des apps pour le Web, le Cloud, le desktop et le mobile.

Ne pas confondre low code et no-code

Parfois on confond low code et no-code. Le low code signifie parfaitement que le code est toujours présent mais en proportion réduite. Il faut tout de même avoir un minimum de connaissances / compétences pour utiliser ces solutions et générer des applications. Le no-code veut supprimer totalement la notion de codes.



sances / compétences pour utiliser ces solutions et générer des applications. Le no-code veut supprimer totalement la notion de codes.

L'application sans codes : les 3 mythes

C'est un des principes du low code mais on ne le prend qu'au sens strict du terme : créer une app sans code. Une étude du cabinet Forrester, en 2016, mettait en avant 3 mythes :

- 1 : Les solutions low code sont faites pour des développeurs non professionnels, pour les utilisateurs ;
- 2 : Ces solutions suppriment toute programmation ;
- 3 : Le low code signifie petite montée en charge, donc pour des utilisations limitées avec peu d'utilisateurs et de fonctionnalités.

Ces mythes sont faux. Oui, les solutions low code sont pour des utilisateurs avancés non développeurs mais ils peuvent être utilisés par des développeurs. Ensuite, à un moment donné, il faut rentrer dans le code, notamment pour la glue entre les composants et les couches, tout ne peut pas être fait via des diagrammes ou des briques préfabriquées. L'intégration est un exercice délicat. Ensuite, le low code peut parfaitement être utilisé à grande échelle avec des milliers d'utilisateurs.

Quelle définition ?

Il n'existe pas de définition unique car chaque fournisseur de low code aura sa vision. Si nous prenons la définition d'OutSystems, une plate-forme low code doit rassembler :

- Un développement visuel pour dessiner / définir les workflows (métiers), l'interface, les modèles de données ;
- Une intégration automatique des données (SQL, NoSQL, etc.) ;
- Une gestion du cycle de vie : déploiement facilité, gestion de projet...

Autre définition possible : les plateformes low code fournissent des environnements de développement pour que les équipes métiers développent leurs logiciels rapidement avec un minimum de code et une connaissance limitée de la programmation. Ces plateformes permettent de prototyper, de construire des applications pouvant monter en charge à une large échelle, sans besoin de construire une infrastructure complexe. Les développeurs et non développeurs peuvent utiliser ces applications. Certains projets nécessitent des développements plus lourds et complexes, d'autres non. Les solutions low code se diffèrentent des solutions no code par une plus grande richesse fonctionnelle et la personnalisation poussée. L'offre low code est importante. Plus d'une quinzaine de solutions existent sur le marché avec plusieurs poids lourds comme Appian, Outsystems, K2, Zoho Creator. Mais les gros éditeurs ne veulent pas être en retard. Ainsi, Google, Microsoft, Salesforce possèdent leur propre solution de low code. Pas toujours facile d'avoir une idée précise du chiffre d'affaires de ce marché. En 2015, le cabinet Forrester estimait le marché à presque 4 milliards \$ pour 2017 et dépassera 10 milliards en 2019.

Le point de vue de Jean Cadeau (K2)

La surface d'utilisation est dépendante des outils utilisés pour la conception des applications. On peut en effet distinguer les applications à créer en 3 catégories :

1. Applications de production personnelle ;
2. Applications d'équipes ;
3. Applications d'entreprises.

Chaque catégorie disposera d'outils qui lui sont propres :

1. Par exemple Microsoft Powerapps ou IFTTT ;
2. Par exemple Nintex ;
3. Par exemple K2.

Les différentes cibles

Dans le cas 1, les outils vont donner la possibilité à l'utilisateur un peu geek de se créer des routines en fonction d'évènements prévus (par exemple : si un nouveau document est déposé à tel endroit dans ma GED, m'envoyer une notification mobile et envoyer un mail à tel(s) utilisateur(s)).

Ces applications là sont assez souvent liées à des objets connectés et devraient donc avoir un essor important dans les années à venir. Elles permettent de couvrir des actions que les gens faisaient manuellement auparavant et qui étaient donc un peu rébarbatives, mais ces actions sont très dépendantes de la façon dont l'utilisateur organise son travail. C'est en ça que l'on va appeler cela des applications de production personnelle. Ce qui est alors mis en place par une personne n'a pas forcément d'intérêt pour une autre, mais chacun améliore son activité.

L'éditeur fournit alors des outils qui permettent des interactions simples avec l'utilisateur et qui va permettre de dessiner un « mini-processus » pour séquencer les actions à réaliser.

Dans le cas 2, les outils ont pour vocation de permettre la création à l'échelle d'un service ou d'une équipe. Comme par exemple la gestion des comptes-rendus de réunion, la gestion des demandes de déplacement... Comme pour le cas 3, l'éditeur fournit des outils qui vont permettre de créer des interfaces utilisateurs qui vont permettre de créer des processus pour séquencer ces interfaces. En termes de personnalisation des formulaires, on pourra aller plus loin que pour le cas 1, mais si l'on souhaite faire des choses un peu dynamiques, il sera alors vite nécessaire de mettre la main dans le cambouis et d'y aller à coup de Javascript...

Ici on accélère donc la production d'applications, mais on ne s'affranchit pas totalement du code. Plus l'application est avancée plus on sera confronté aux limites du produit.

Dans le cas 3, les outils vont permettre de créer rapidement et simplement des applications critiques et centrales au sein de son organisation ; la possibilité d'écrire du code est offerte pour des raisons d'ouverture mais cela est rarement nécessaire.

Pour de telles applications, il y a un besoin fort d'interconnexion avec le SI existant ; une des force de ces outils sera donc d'être en mesure d'aller mettre à jour et lire des informations dans le SI existant en proposant des connecteurs qui permettront de ne pas avoir à écrire

de code comme par exemple avec les ERP (SAP, JDE...), les outils de CRM (Dynamics, Salesforce...), les bases de données (Oracle, Sybase, SQL Server...), les outils de GED (SharePoint, Documentum...).

Au-delà de la connexion au SI, l'outil de low code/no code va permettre au concepteur de créer les différentes interfaces utilisateurs de son application (donc les formulaires) en gérant les interactions avec l'utilisateur final (listes déroulantes en cascades, messages contextuels, règles de validation etc.). Il pourra ensuite gérer le séquencement de formulaires entre les différents intervenants. Les AGL fournis par l'éditeur permettent alors réellement de « dessiner du code » et d'aller très loin dans la personnalisation et la complexité des applications.

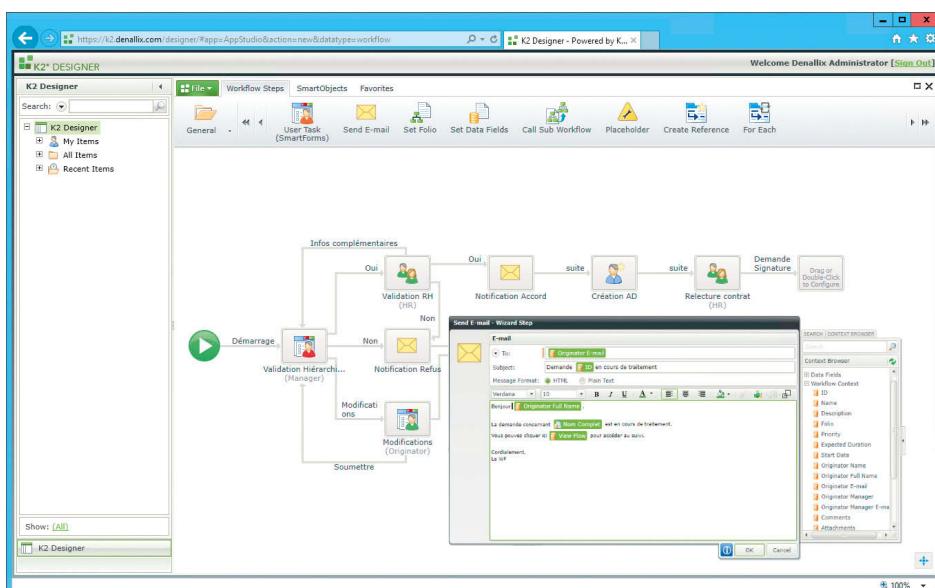
Les exemples d'applications : gestion du piquage qualité sur une chaîne de production, création de nouveau produit dans l'industrie ; dans la banque personnelle, gestion des demandes d'ouvertures de comptes, des demandes de crédit ; la gestion des incidents ou des réclamations clients, etc.

Qui peut le plus peut le moins, les outils de cette catégorie vont également permettre de couvrir ce qui est porté par les catégories inférieures.

Même si cela est également valable (mais dans une moindre mesure) pour les catégories 1 et 2, je vais me concentrer sur la catégorie 3 que je maîtrise beaucoup plus et pour laquelle le ROI est plus quantifiable :

En comparaison à du code normal, l'objectif des outils « low code/no code » est d'accélérer la mise en place des applications, en abstrayant au maximum la complexité technique inhérente, pour avoir un Time-To-Market beaucoup plus rapide (jusqu'à 20x plus rapide en fonction des besoins). Les compétences nécessaires des concepteurs sont donc moins pointues (et donc forcément moins chères). Il est même possible, lorsque les applications ne nécessitent pas trop de connexions différentes au SI existant, de faire construire les applications par des non informatiens (mais rompus aux technologies)... Ce que les analystes américains appellent les Citizens Developer.

De plus, de tels outils viennent avec un certain nombre de fonctionnalités toutes packagées qu'il n'y a donc pas à implémenter au sein de ses applications : comme par exemple, la gestion des absences des utilisateurs, les interfaces

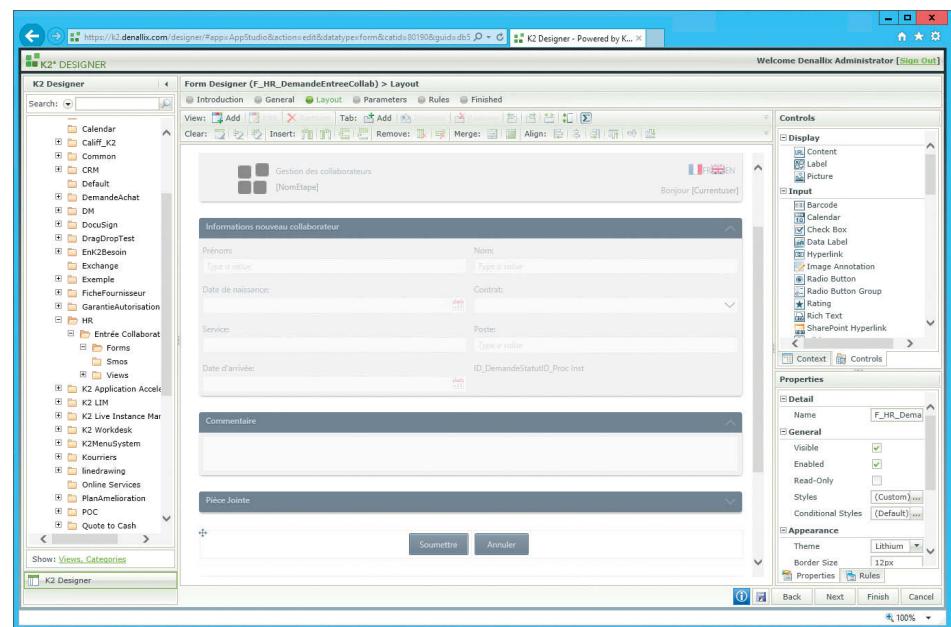


d'administration techniques, les dashboards de suivi applicatif, la persistance des données, la traçabilité des modifications, la mise à disposition d'applications mobiles natives permettant d'accéder et d'actionner ses tâches, etc.

Les avantages pour l'entreprise sont donc les suivants :

- Elle peut délayer sur plus de profils la création d'applications (donc donner un peu plus de pouvoir sur la gestion de ses outils à des utilisateurs métier qui auparavant étaient plus dépendants des services informatiques) ;
- Avec sa même équipe de développeurs existants, elle va pouvoir produire plus et plus rapidement (et améliorer ainsi son image auprès de ses salariés) ;
- Traçabilité native ;
- Mise à disposition de fonctionnalités natives qui n'étaient pas forcément imaginées dans un 1er temps ;

Les inconvénients : il y en a quelques-uns et ils sont surtout portés par les développeurs qui vont devoir changer leur façon de concevoir les applications et qui seront plus contraints que lorsqu'ils faisaient du code pur. Il peut y avoir une réticence à l'utilisation de tels produits car il y a une impression de moins contrôler ce qui se passe, même s'il est possible d'écrire du code



donc de faire exactement ce que l'on veut... si on fait cela en permanence, on perd les avantages de l'outil.

Pour moi, les outils Low Code/No Code, c'est le futur de l'informatique. Il y a de nombreux outils qui proposent cela de manière plus ou moins complète, mais cette philosophie com-

mence à imprégner les entreprises afin de réduire le Time-To-Market. D'ailleurs cette année est sortie la 1ère matrice « LowCode/NoCode » du Gartner... Donc même si les analystes s'y mettent, c'est que le marché est en pleine expansion

Point de vue de Nicolas CHABRIER

Pourquoi voit-on émerger le « low-coding » ?

On pourrait commencer par une mauvaise rédaction de collège avec « De tout temps les hommes, ont cherché à faire du low-Code » car ce thème est aussi vieux que l'histoire du développement informatique.

En effet, créer des applications sans coder semble être le saint Graal que chaque éditeur semble promettre à tout le monde depuis près de 40 ans.

Mais aujourd'hui cette utopie devient une nécessité tant les besoins d'informatisation sont grands et tellement tous les moyens pour répondre à la digitalisation de notre environnement sont acceptés avec bienveillance.

Chaque entreprise ou administration se voit, en effet, confrontée à des challenges dont elle ignoret même l'existence il y a quelques années (dématerrialisation des processus, transparence, traçabilité, accessibilité, mobilité ...) Bref, la tâche est immense et on ne peut physiquement pas embaucher des armées d'informaticiens pour répondre à tous ces chantiers.

Concrètement, l'informatisation étant une nécessité moderne pour la pérennité des acteurs

économiques, chacun doit prendre sa part de responsabilité et contribuer à cette mutation.

Mais qu'est-ce que le « low-code » ?

A l'instar d'une analyse littéraire, il convient de s'intéresser aux termes employés.

Tout d'abord, démarrons par le mot « Code », on comprend qu'il s'agit d'une solution IT qui va permettre la création, la génération de code informatique. On est donc en présence d'une technologie qui a pour vocation de fournir une solution productive au-delà du prototypage. L'ambition affichée est de se substituer à de la génération manuelle (et intellectuelle) de programmes informatiques.

Le terme « Low » a aussi son importance et a d'ailleurs été le fruit d'une évolution marketing. Certains parlaient (ou parlent encore) de « No

Code ». Or la réalité est tout autre et l'on sait maintenant d'expérience que le « No-Code » est au meilleur des cas une solution très restrictive, et au pire un mensonge éhonté.

Alors concrètement, c'est quoi ?

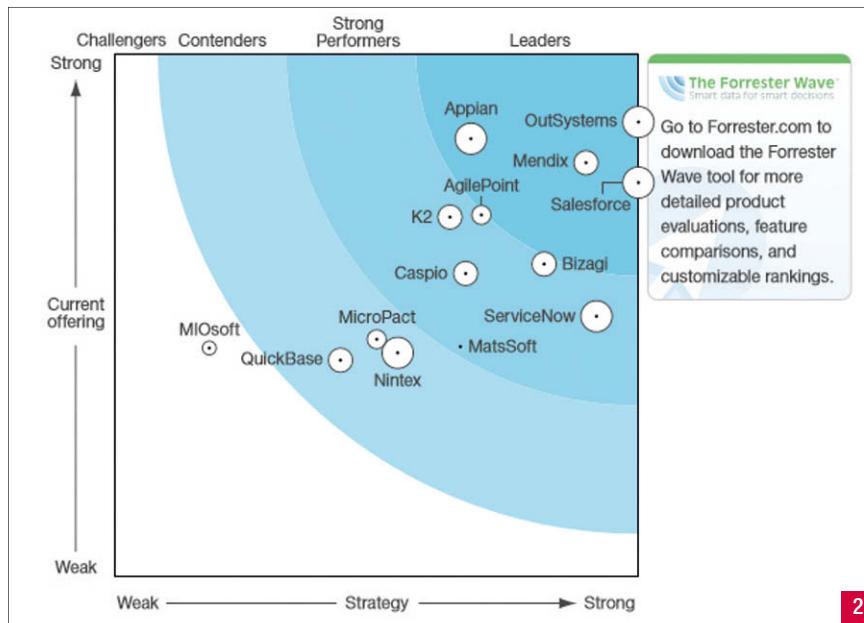
Le « low-coding » est donc une réponse technologique à des problèmes récurrents présentant des similitudes de conception et pouvant ainsi être factorisés et traités de manière unifiée.

Regardons un logiciel quelconque et essayons de voir de quoi il se compose. Comme un iceberg, l'utilisateur final voit et perçoit le logiciel via son interface. C'est la partie visible qui est fondamentale pour l'expérience utilisateur.

Une solution « low-Code » doit gérer les interfaces utilisateurs.



Besoin de customisation



On trouve des acteurs assez méconnus du grand public (même chez les professionnels de la profession pour les adeptes du pléonasmes) qui pourtant présentent des profils intéressants. Une analyse de Forrester a par exemple montré qu'une implémentation d'une solution Appian ou K2 présentait un ROI de plus de 450% à 5 ans.

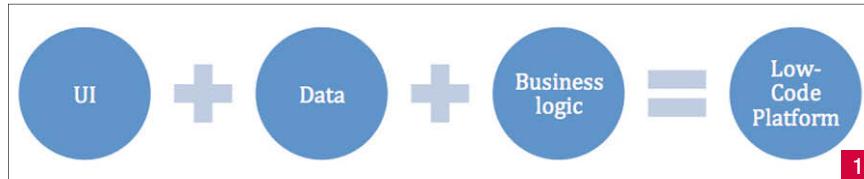
Mais qui va faire du « low code » ? Est-ce un risque pour les informaticiens ?

Il y a autant d'informaticiens que de problèmes informatique. En effet, il y a peu de points communs entre un DBA Oracle et un développeur Angular, mais c'est de l'informatique et tous deux mobilisent une pensée cartésienne pour trouver une solution efficace. Le « Low-Code » ne déroge pas à la règle, il s'adresse à une population de personnes cartésiennes sachant structurer leurs problèmes et proposer des solutions simples ou un assemblage de solutions simples pour solutionner un besoin.

Quid des développeurs, des informaticiens, vont-ils se faire manger la laine sur le dos ? A mon avis, il n'en est rien. Le « Low-Code » est une nouvelle voie qui va justement permettre de répondre aux besoins d'informatisation des entreprises et permettre aux professionnels de se focaliser sur les réalisations complexes à forte valeur ajoutée. [3]

Nous voyons émerger une troisième voie entre le développement spécifique reposant sur l'utilisation de langages et de patterns et l'implémentation de solutions verticales.

Derrière tout risque se cache une opportunité. Ici, clairement l'IT renforce son rôle central dans les réflexions et dans le fonctionnement de l'entreprise. Qui pourrait s'en plaindre ?



Un logiciel est aussi un outil d'interconnexion de données. Rares sont les Softwares de nos jours qui ne consomment ou ne fournissent pas de données (qu'elles soient externes, c'est-à-dire des solutions extérieures au logiciel, ou qu'elles soient intrinsèques, c'est-à-dire produites par le logiciel lui-même).

Une solution « low-code » doit gérer l'accès aux données. Entre ces deux extrêmes (interface et donnée), il y a le cœur du logiciel avec des multiples concepts :

- Moteur d'orchestration pour assurer le séquencement des actions ;
- Moteur de règles pour assurer l'exécution d'une logique métier spécifique ;

- Gestion de la sécurité et de l'authentification ;
- ...

Ce que nous pourrions qualifier de composants de logiques Métier (pas de l'interface, pas de la donnée pure mais autre chose...).

Bref, on parle de plateforme de « low-code » lorsque l'on a au moins 3 des éléments suivants. [1]

Et après ?

Maintenant il convient de voir, si réellement le résultat est au rendez-vous. Alors de nombreux organismes se sont penchés sur l'analyse d'une implémentation « Low-Code » et citons par exemple le cas de Forrester. [2]

//Build 2017 : les annonces à retenir de la conférence développeurs

Les développeurs passionnés



INFINITE SQUARE

L'édition 2017 de //BUILD, l'évènement Microsoft dédié aux développeurs, vient d'avoir lieu à Seattle. Voici un résumé des principales keynotes et des sessions qui nous ont le plus marqué durant ces trois jours.

KEYNOTE JOUR 1 IoT edge, Azure Cosmos DB, IA... [1]

Satya Nadella a débuté la Keynote, comme en est l'usage, pour communiquer sur la stratégie globale Microsoft.

Sur la partie applicative, Satya a insisté sur la fin du « one-device » et des interfaces « mobile first ». Toute application doit maintenant être multi-device et multi-hardware. Concrètement, chaque application doit être capable de m'apporter un service qui s'adapte automatiquement pour me proposer la meilleure interface homme machine en fonction de ma situation : sur mon pc, en mobilité, dans ma voiture, chez moi... Le terme "Windows as a Service" a de nouveau été évoqué pour parler de l'avenir de Windows, preuve que Windows mute également vers ce mode de consommation multi-usage (Desktop, IoT, VR...). Les annonces liées à Windows seront toutefois réservées pour la 2ème keynote, dédiée à Windows et au matériel.

Annonce de IoT Edge

Vient ensuite le tour des annonces et nouveautés liées à Azure, chaque présentation étant illustrée par un exemple concret d'implémentation et une référence client pour prouver la réalité des scénarios démontrés.

Azure IoT Edge permet d'exporter des fonctionnalités d'Azure IoT (IoT Hub / Stream Analytics / Azure Function / Machine learning sur une machine hors d'Azure), en utilisant des containers dockers exécutés en local. De la logique présente dans Azure peut ainsi être repackagée et exécutée sur un environnement local en quelques clics. Chaque device peut ainsi devenir une extension d'exécution du Cloud.

<https://azure.microsoft.com/en-us/campaigns/iot-edge/>

Azure AD Graph et O365

L'API Azure AD Graph continue à évoluer en profondeur en proposant de plus en plus d'interfaces pour permettre aux développeurs de consommer, modifier de la donnée et créer de nouveaux usages. Un utilisateur dans le Graph



n'est pas qu'un User, mais c'est également maintenant des Devices, pour renforcer le message « multi-devices ». Par exemple, une réunion dans un contexte d'entreprise, est un objet qui doit pouvoir être manipulé depuis différents usages, à des moments différents (avant, prise de rendez-vous depuis Cortana, préparation en mobilité, pendant, interaction depuis MS Teams, après, suivi, résumé partage depuis Office 365).

Les nouveautés Azure

Scott Guthrie a ensuite pris le relais de Satya Nadella pour entrer plus en profondeur sur **quelques fonctionnalités d'Azure** :

- Intégration d'un shell bash directement dans le portail Azure.
 - Annonce d'une application IOS et Android disponible pour gérer le portail Azure (et faire des mises en productions depuis les transports en commun :))
 - Possibilité de debugger en production de manière sécurisée depuis Visual Studio en utilisant des « debug snapshots » générés automatiquement par Application Insights (<https://docs.microsoft.com/en-us/azure/application-insights/app-insights-snapshot-debugger>).
 - Support natif des containers dockers dans Visual Studio 2017 et dans Azure Service Fabric.
- Viennent ensuite les **annonces de nouveaux services** :
- MySQL as a Service et PostgreSQL as a Service sont maintenant officiellement disponibles, avec le même positionnement que Azure SQL Database.

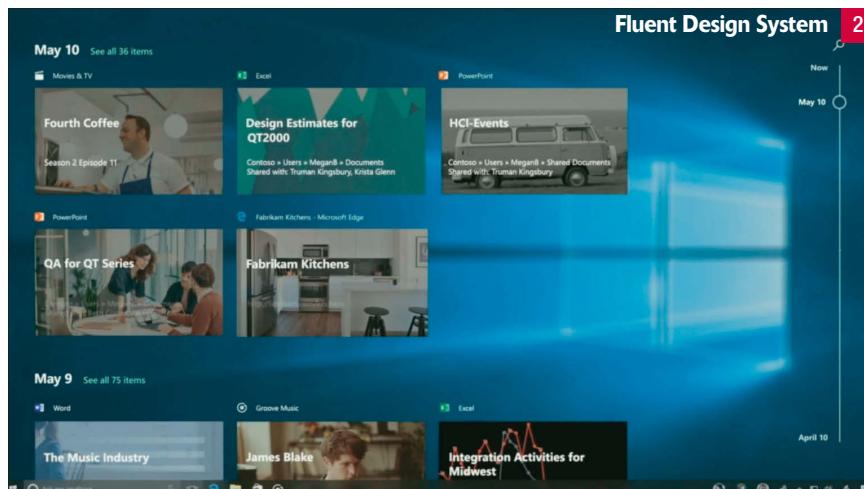
- Apparition d'Azure Cosmos DB, une base de données NoSQL géo-distribuée sur plusieurs data centers, avec plusieurs modes de synchronisation de données. C'est pour moi l'annonce la plus majeure et intéressante de cette keynote.
- Visual Studio pour Mac est en RTM et devient le successeur officiel de Xamarin Studio sur cet OS.

La dernière partie de la keynote a enfin été dédiée à l'IA, illustrée au travers des interfaces homme machine de type BoT et des analyses et traitements de données intelligentes et en temps réel s'appuyant sur les cognitive services (Image, vidéo, son). Le tout illustré par un exemple bluffant d'une nouvelle fonctionnalité intégrée dans PowerPoint pour traduire et sous titrer en temps réel une présentation.

KEYNOTE JOUR 2 Fall Creators Update, Fluent Design System, Windows Mixed Reality...

La seconde journée de l'évènement Build 2017 est cette fois-ci focalisée sur Windows et son écosystème.

Et c'est Terry Myerson, vice-président exécutif du groupe Windows et Devices qui démarre cette journée en nous rappelant que Windows 10 ne décompte pas moins de 500 millions de Devices actifs par mois, et est bien évidemment la version de Windows la plus utilisée dans le monde entier. Il annonce ensuite l'arrivée de la nouvelle mise à jour de Windows 10



3 Windows Mixed Reality



baptisée **Fall Creators Update** qui, comme son nom l'indique, a pour vocation de permettre aux utilisateurs et aux développeurs encore plus de créativité dans leur quotidien.

Fluent Design System et Microsoft Graph [2]

Microsoft présente ensuite la définition de leur nouveau **Fluent Design System**, qui a été réfléchi pour aider à créer des applications disponibles à travers un large panel de plateformes diverses et variées. Ce design est basé sur 5 principes : La lumière (Light), la profondeur (Depth), les animations (Motion), la matière (Material) et l'échelle (Scale). (<http://fluent.microsoft.com/>). En plus de cette nouvelle ergonomie, la Fall Creator Update va permettre aux utilisateurs de bénéficier de l'intégration de Windows 10 avec les API Microsoft Graph. Concrètement cela se matérialise par de nombreuses petites nouveautés :

- **OneDrive Files On-Demand** : tous les fichiers synchronisés sur OneDrive seront visualisables à travers tous les Devices connectés, allant de la tablette Windows au Mobile iOS. Ceux-ci n'occuperont aucun espace mémoire, mais seront téléchargés à la volée lorsque manipulés par un utilisateur.
- **Windows Timeline** : une Timeline récapitu-

lant l'utilisation de ses applications (compatibles Windows Timeline) à travers tous ses Devices sera disponible, permettant ainsi de rouvrir facilement un fichier précédemment modifié. Mieux encore, lorsque je quitte mon ordinateur avec une application ouverte par exemple, la fonctionnalité de « **Pick Up Where I Left Off** » de Cortana va me proposer sur mes autres Devices (PC, mobile iOS, Android, etc.) de rouvrir cette même application dans ce même contexte, permettant ainsi aux utilisateurs une transition fluide entre tous leurs Devices.

- **Clipboard(Cloud Powered)** : il sera également possible de faire un « copier » d'une valeur depuis son ordinateur, puis de naviguer sur son Device utilisant une App compatible « Clipboard » telle que le clavier Swift sur iOS, et de pouvoir faire un simple « coller » sur son autre Device. Simple, mais réellement pratique.

Développement depuis Windows

Côté développement pur, la volonté de standardisation de la plateforme .NET se poursuit, et l'annonce de l'arrivée du **.NET Standard 2.0 pour les App UWP** (Universal Windows Plat-

form) va bien dans ce sens. Autre nouvelle qui sera fortement appréciée des développeurs XAML : l'annonce d'un **XAML Standard 1.0** qui permettra d'unifier la création d'interfaces sous toutes les plateformes mobiles !

On note également la disponibilité du **Project Rome SDK sur iOS** en Preview, permettant d'interagir avec les API de Microsoft Graph pour faciliter les interactions cross-devices. (<https://github.com/Microsoft/project-rome>)

C'est avec plaisir que l'on apprend également que le récent **Visual Studio Mobile Center**, véritable point central pour gérer son développement mobile de A à Z, supportera enfin les **applications UWP** !

C'est avec une petite touche d'humour que l'on apprend que de nombreux systèmes seront également présents sur le Windows Store tels que les OS Linux Ubuntu, Suse ou encore Fedora, mais aussi le logiciel iTunes. Autre annonce à ne pas manquer : l'apparition du **Xamarin Live Player**, qui permet aux développeurs Xamarin pour iOS de pouvoir déboguer à distance, ou même de modifier l'UI en Xaml d'une application iOS, sans devoir passer par un Mac ! (Disponible également pour Android !)

Windows Mixed Reality et Motion Controllers

Cette seconde journée se clôture par l'un des sujets les plus innovants démarré par Microsoft ces dernières années : Hololens ainsi que tous les devices de réalité augmentée, réalité virtuelle, etc. D'ailleurs, pour uniformiser le tout, Microsoft se réfère au terme de **Windows Mixed Reality** pour englober tout cet univers, et laisse entendre que le développement devrait se révéler en grande partie uniforme sur cette plateforme. Pour anecdote, Hololens, qui a commencé sa distribution publique l'année dernière, représente à l'heure actuelle 22 000 développeurs, 70 000 concepts et plus de 5,5 millions d'heures passées sur ces devices. Le produit est actuellement disponible dans 9 pays, et devrait bientôt être disponible en Chine, autant dire que l'avenir du sujet semble toujours très prometteur.

Pour finir, la dernière annonce de cette Keynote de la deuxième journée de l'événement Build 2017 n'est autre que l'arrivée de **Motion Controllers** pour les devices de *Windows Mixed Reality*, basés sur du tracking optique intégré, possédant un joystick, un touchpad ainsi qu'une gâchette. De plus, un pack de ces Motion Controllers + Casque VR Acer à 399\$ devrait apparaître sur le marché. [3]

Tous les numéros de

programmez!

le magazine des développeurs

sur une clé USB (depuis le n° 100)



Clé USB 4 Go.
Photo non
contractuelle.
Testé sur Linux,
OS X,
Windows. Les
magazines sont
au format PDF.

* tarif pour l'Europe uniquement.
Pour les autres pays, voir la boutique en ligne

Commandez la directement sur notre site internet : www.programmez.com

Complétez votre collection

Prix unitaire : **6,50€**



<input type="checkbox"/> 180	: __ ex	<input type="checkbox"/> 200	: __ ex
<input type="checkbox"/> 191	: __ ex	<input type="checkbox"/> 203	: __ ex
<input type="checkbox"/> 196	: __ ex	<input type="checkbox"/> 206	: __ ex
<input type="checkbox"/> 197	: __ ex		



Commande à envoyer à :

Programmez!
7, avenue Roger Chambonnet
91220 Brétigny sur Orge

Prix unitaire : 6,50 €
(Frais postaux inclus)

soit exemplaires x 6,50 € = € soit au **TOTAL** = €

M. Mme Mlle Entreprise : _____ Fonction : _____

Prénom : _____ Nom : _____

Nom : _____

Règlement par chèque à l'ordre de Programmez !

VISUAL STUDIO FOR MAC EN VERSION FINALE

On l'attendait avec impatience, depuis 6 mois. La [BUILD 2017](#) nous apporte Visual Studio for Mac en version finale. Voici en détail comment installer et utiliser cette version de Visual Studio tant attendue !

Installer et configurer Visual Studio for Mac

Lorsqu'on installe celui-ci, il nous propose d'installer plusieurs composants additionnels :

- Le kit de développement Android, ainsi que Xamarin Android ;
- Le kit de développement iOS, ainsi que Xamarin iOS. (XCode doit être aussi installé) ;
- Le kit de développement Mac, ainsi que Xamarin Mac OS. (XCode doit être aussi installé) ;
- Le kit de développement .Net Core ainsi que tous les outils pour développer en .Net Core sous Mac OS ;
- Workbooks : l'outil permettant d'avoir des exemples de code interactif [4]

Premier lancement

Lors du premier lancement on s'aperçoit que Visual Studio est configuré comme la version community. [5]. Après m'être identifié avec mon compte Microsoft Pro celui-ci passe automatiquement en version Entreprise.

Ouvrir plusieurs solutions

Une des choses les plus gênantes lorsque l'on commence à développer sur Mac c'est lorsque l'on ouvre une solution; si une solution est déjà ouverte dans Visual Studio for Mac celui-ci ferme la solution courante pour ouvrir la nouvelle. Les développeurs de Visual Studio for Mac ont suivi à la lettre le paradigme de Mac OS, les applications sont exécutées dans une seule ins-

tance. Pour régler ce souci, rien de plus simple, lorsque vous ouvrez une solution il faudra décocher "Close current Workspace". [6]

Et Xamarin Studio dans tout ça ?

Visual Studio for Mac est l'évolution de Xamarin Studio. L'idée de Microsoft étant d'avoir un nommage cohérent entre tous les IDE qu'ils développent. Désormais Visual Studio à 3 moutures :

- Visual Studio 20XX (la version dédiée à Windows) ;
- Visual Studio Code (la version gratuite d'un éditeur de texte comparable à un Notepad++) ;
- Visual Studio for Mac (la version dédiée à MacOS).

Et les inconvénients dans tout ça ?

Visual Studio for Mac ne prend pas en charge tout ce que peut faire un Visual Studio 20XX :

- Pas de .Net complet, on est toujours limité à .Net Core ;
- Pas de possibilité de créer une app UWP.

Autre inconvénient, celui-ci nécessite toujours

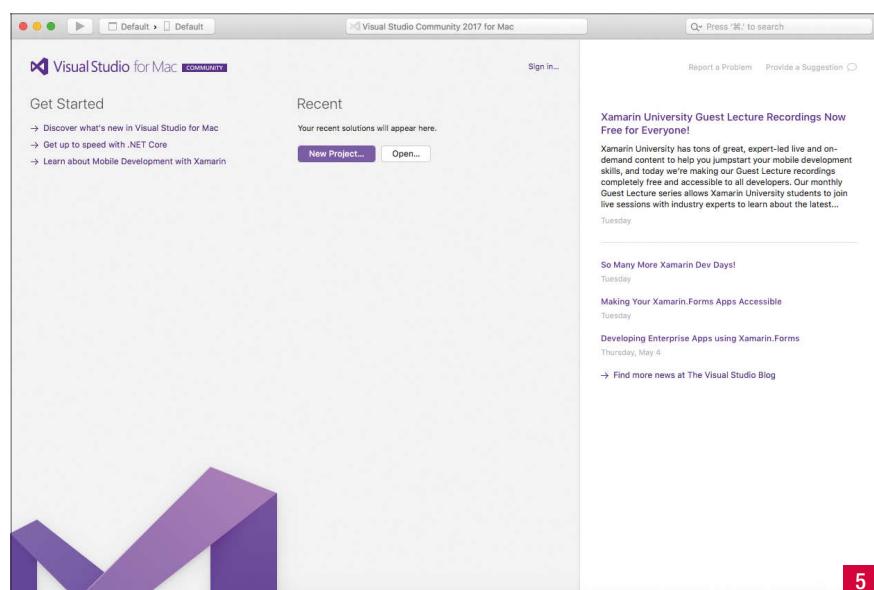
l'installation de XCode pour pouvoir développer des applications iOS et Mac OS.

Conclusion sur Visual Studio for Mac

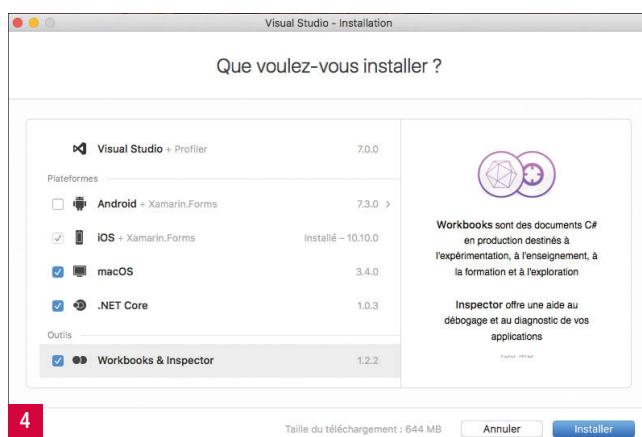
Visual Studio for Mac est l'outil idéal pour faire vos développements .Net Core sur Mac et vos développements Xamarin. L'IDE se rapproche de plus en plus de Visual Studio sur Windows; si vous êtes développeur Windows et que vous devez faire une app sur Mac, vous ne perdrez pas vos habitudes. Néanmoins l'outil est encore très loin de ce que propose Visual Studio sur Windows. Cet IDE, est issu de la stratégie globale de Microsoft de ne plus se limiter plus aux développeurs Windows.

.NET STANDARD 2.0 ET VISUAL STUDIO 2017

Lors de la première journée de cette //build 2017, Scott Hanselman, Scott Hunter et Kasey Uhlenhuth nous ont présenté les nouveautés autour du .NET et notamment autour de Visual Studio 2017.

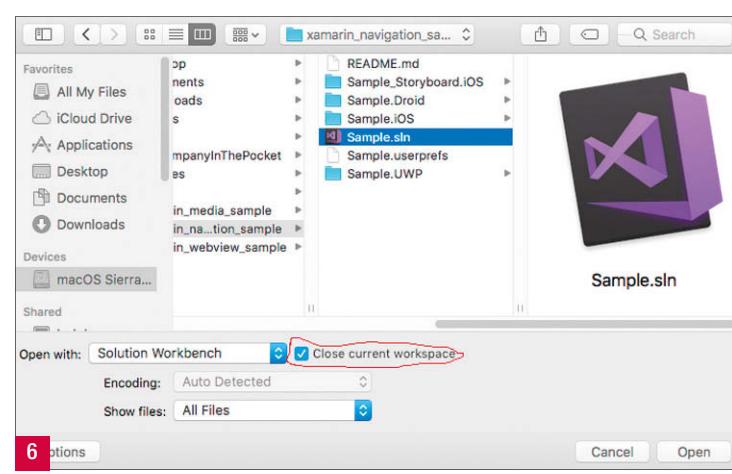


5



4

VS for Mac Installation



6

.NET est ouvert et actif

Le premier point mentionné est l'investissement de Microsoft dans l'open-source ces dernières années. Notamment avec le nombre de Pull Requests autour du .NET qui a explosé depuis 2015, .NET évolue constamment et est très actif, et cela se voit autant sur Github que sur StackOverflow via des sondages. Ils rappellent qu'ASP.NET Core est très performant et que cela se traduit par des équipes qui passent de NodeJS à ASP.NET Core. Avec une telle transition sur un serveur de même envergure, cela leur permet d'augmenter considérablement les nombres de requêtes par seconde de leurs applications.

.NET Architecture

Un nouveau site Web est mis à disposition : <https://www.microsoft.com/net/learn/architecture>.

L'objectif de ce site est de fournir une documentation, autour du .NET, la plus complète possible pour que les entreprises puissent mettre en place les meilleures architectures/patterns possibles dans leurs applications.

Cela inclut de la documentation, mais aussi des exemples, et cela sur divers sujets : applications mobiles (Xamarin), Azure, applications Web...

.NET Standard 2.0

La librairie .NET Standard 2.0 est disponible. Pour rappel, l'idée du .NET Standard est d'unifier toutes les plateformes .NET via un contrat. Et ce contrat s'étoffe avec .NET Standard 2.0. [7]

L'objectif reste le même que .NET Standard 1.6, c'est à dire une connaissance unique à apprendre pour toutes les plateformes .NET, avec toute la chaîne d'outils qui fonctionne pour toutes les plateformes : Visual Studio, Visual Studio pour Mac, Visual Studio Code...

.NET Standard 2.0 apporte beaucoup plus de contrats; cela permet d'utiliser de l'ancien code .NET dans des nouvelles technologies. L'exemple présenté lors de la session est une application Windows Forms qui manipule des DataSets. Le code métier qui manipule des DataSets est alors déplacé dans une librairie .NET Standard. L'application Windows Forms continue de fonctionner parfaitement en référençant cette librairie. Ensuite, une nouvelle

application ASP.NET Core peut aussi référencer et utiliser cette librairie .NET Standard, comme on peut le voir dans leur démonstration. [8]

.NET Core 2.0

Premier point : .NET Core 2.0 supporte .NET Standard 2.0, cela veut dire qu'une application ASP.NET Core 2 peut utiliser environ 70% des APIs du framework .NET. ASP.NET Core et Entity Framework Core sont maintenant inclus directement dans .NET Core 2.0 pour simplifier le packaging. La taille des packages est alors augmentée, mais il va être possible de réduire la taille de ces packages pour ceux qui en ont besoin. Autre point important, les références étant simplifiées, la configuration devient réellement plus simple, plus concise.

Les commandes ont aussi été améliorées, ainsi il est possible, en spécifiant un paramètre à la commande dotnet new, de créer le projet selon plusieurs templates de projets mis à disposition.

Visual Studio 2017

Un des problèmes remontés assez souvent autour de Visual Studio est la nécessité d'installer plusieurs extensions pour avoir une meilleure expérience.

Ainsi, les nouveautés de l'IDE version 2017 RTM du 7 mars sont :

- De meilleures performances et une installation plus efficace ;
- Une navigation plus rapide ;
- Live Unit Testing (.NET Core non supporté) ;
- Amélioration de l'IntelliSense ;
- Une aide aux exceptions plus poussée ;
- Plus de refactoring et de fixes automatisés ;
- C# 7.0 !

Avec Visual Studio 2017 Update 3 qui va sortir, de nouveaux points sont présents :

- Encore plus d'IntelliSense ;
- Live Unit Testing pour .NET Core ;
- Du diagnostic de code sur plusieurs points, comme la sécurité ;
- Encore plus de refactoring !
- C# 7.1.

L'audience aura retenu deux moments intenses :

- Lors d'une NullRef-

renceException, la valeur qui est null est maintenant indiquée au développeur.

- Il n'est plus obligatoire d'unloader le projet pour pouvoir éditer son .csproj !

Sinon, une feature très intéressante en termes de productivité et que les utilisateurs de Visual Studio Code doivent bien connaître : le "Go To All" avec la possibilité de choisir ce que l'on recherche. Il est alors possible de chercher des types, des membres, des symboles ou même des fichiers en spécifiant le type d'objet que l'on recherche. En réalité la feature Go To All existait déjà sous Visual Studio 2015 ("Ctrl + virgule" par défaut), mais il n'était pas possible de cibler un type d'entité particulier.

.NET Core et Visual Studio arrivent sur Mac

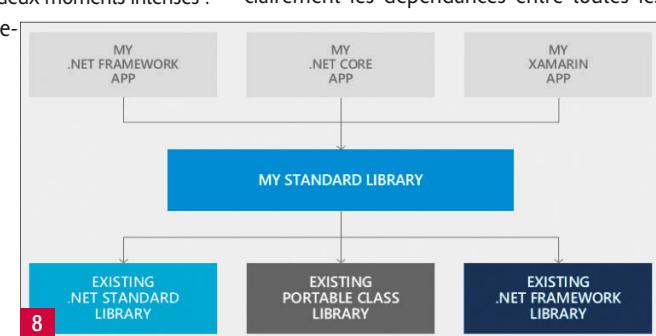
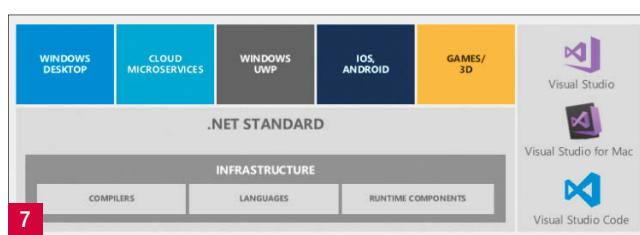
Comme l'annonce le titre de cette partie, .NET Core et Visual Studio seront disponibles sur Mac, et cela avec la même expérience de développement que sur Windows. C'est à dire qu'il sera possible de débugger, publier sur un site Web Azure, etc.

Les diagnostics Azure

Côté Azure, un point d'attention a été porté sur l'aspect diagnostic. Il est important de pouvoir diagnostiquer ses applications, et cela sans devoir modifier celles-ci en conséquence, ni même de devoir les republier. L'objectif est de pouvoir découpler au maximum la partie diagnostic de la partie métier de l'application.

Il est alors possible de diagnostiquer son application depuis le portail tout naturellement, mais aussi via Visual Studio par le biais de "snap points" pour débugger une application en exécution. L'idée est vraiment de dissocier le diagnostic de l'application en soi. Dans le passé, il fallait aller sur NuGet, ajouter les références, mettre en place les branchements, etc... Maintenant, l'idée est de n'avoir qu'à publier son application, et Azure s'occupe du reste.

La killer feature tourne autour des dépendances, grandement améliorées; on peut voir clairement les dépendances entre toutes les



DOTNET Standard 2.0 - Exemple

briques de l'application ce qui permet à l'utilisateur d'avoir une vision adaptée en fonction de son besoin. Les problèmes sont alors regroupés, et on retrouve un concept de "bucket" où les exceptions sont mises en avant, de manière groupée et selon leur importance. L'interface semble très intuitive et, même si Application Insights est déjà une pépite en soi, ces nouveaux diagnostics semblent réellement être un très grand atout pour l'avenir. En plus de retrouver la Call Stack d'une exception, on peut alors télécharger un dump qui représente le snapshot de cette exception, afin d'investiguer encore plus en profondeur l'exception. Ainsi, avec cette notion de "snap point", opposé au breakpoint, rien n'est arrêté; il est possible de débugger sur Visual Studio sans impacter l'application qui tourne sur Azure. Par contre, la démonstration n'a pas fonctionné, mais mettons ça sur le fameux "effet démonstration" !

Azure Functions

Enfin, la session se finit avec les Azure Functions, qui supportent maintenant le .NET Framework et .NET Core. On a aussi de l'IntelliSense, des références entre projets, des tests unitaires et de l'analyse de code. Que demander de plus ? Le fait de pouvoir les publier directement sur Azure depuis Visual Studio par exemple, mais aussi de les débugger autant localement qu'en remote.

Conclusion sur .NET Standard 2.0 et Visual Studio 2017

Cette session, assez riche en nouveautés, semble très prometteuse pour l'avenir de .NET en général, de Visual Studio et d'Azure. Dans l'ensemble, on retient que l'expérience de développement est fortement enrichie et que la productivité, ainsi que l'encouragement à produire un code de qualité maintenable, sont au centre de cet écosystème.

Il ne reste plus qu'à tester tous ces nouveaux outils, puis opposer la réalité à toutes ces belles promesses qui nous ont été faites !

Pour voir la session, c'est par ici : [Three Run-times, one standard... .NET Standard: All in Visual Studio 2017](#)

RÉCAPITULATIF DES ANNONCES OFFICE & SHAREPOINT

Nous avons pu découvrir lors de la Build 2017 un certain nombre d'annonces au sein de la galaxie Office (Office / Office 365 / SharePoint) que je vais vous résumer dans cet article.

SharePoint Framework (SPFx)

La session "[Create the modern workplace with the SharePoint Framework](#)", présentée par Vesa Juvonen et Daniel Kogan, a été l'occasion de découvrir plusieurs nouvelles fonctionnalités intéressantes concernant le SharePoint Framework. Tout d'abord, la disponibilité (en Dev preview) des **connexions entre Web Parts** : un classique régulièrement demandé pour interagir entre plusieurs Web Parts (comme un cas typique de Master/Details). Le mécanisme se base sur un principe d'évènement émis par la Web Part fournissant les données qui sera redi- patché par le SPFx vers les différentes Web Parts s'étant abonné à celui-ci. [9]

Ensuite, l'annonce des **extensions** du SharePoint Framework : les Application, Command Set et Field Customizers. Ces extensions viennent combler les vides laissés par le passage aux Modern Sites qui ont bridé les personnalisations auparavant possibles, principalement pour des raisons de robustesse et de support, et offrent donc un cadre bien délimité pour étendre vos sites. On retrouve ainsi sous un autre nom des concepts déjà connus pour les vieux routards de SharePoint :

- **Application Customizer** : on reprend la logique des custom actions puisqu'on peut ajouter des scripts au niveau d'une collection, d'un site, d'une liste. A noter qu'une zone de l'interface est réservée pour l'affichage d'informations spécifiques.

- **Command Set Customizer** : ici aussi, on

retrouve nos custom actions, mais liées à l'enrichissement de la toolbar des listes et du menu contextuel des éléments/documents.

- **Field Customizer** : vous l'avez compris, ici, c'est l'équivalent du JSLink, donc la personnalisation de l'affichage des champs d'une liste.

La 3eme annonce concerne le support d'**OpenId Connect pour les solutions tierces**, qui vise notamment à simplifier l'utilisation du Microsoft Graph au sein des développements réalisés avec le SPFx. En effet, jusqu'à présent il fallait pas mal de code (merci [Adal.js](#) tout de même qui aidait beaucoup) pour pouvoir s'authentifier auprès de l'application et consommer les API. Les extensions et OpenId Connect ne sont pas encore disponibles, mais le seront dans les prochaines semaines en dev preview.

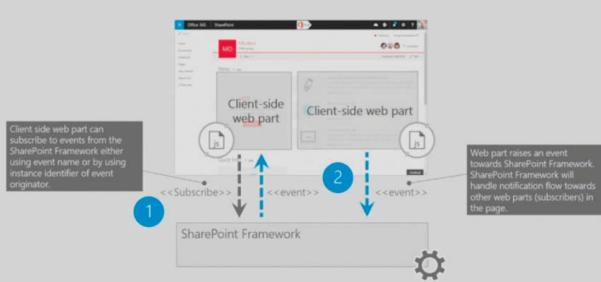
On notera l'utilisation du terme "**Canvas**" pour désigner les différents points d'entrées pour les extensions Office & SharePoint, que ce soit un document, une page, une conversation, autant d'endroits dans lesquels on va pouvoir bénéficier d'un socle commun.

Pour finir, une roadmap sur les prochains mois a été présentée, avec comme focus la consolidation des outils actuels (notamment le Workbench) et l'implémentation des nouveautés du SPFx, mais aussi de futures améliorations des add-ins SP qui ne sont pas oubliés (ni forcément remplacés par le SPFx) : [10]

Add-ins Office

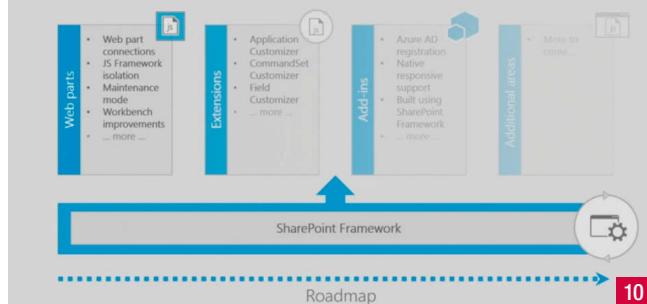
Une des nouveautés qui me semble la plus attendue concerne la fonctionnalité de SSO (Single Sign On) : il va être enfin possible d'authentifier facilement la personne connectée et de consommer les API Office 365 (MS Graph) sans avoir à passer par des popups d'authentification et toute une plomberie complexe. Il est désormais possible de demander un Access Token par simple appel à Office.js (nouvelle méthode `Office.context.auth.getAccessTokenAsync`), puis de l'échanger côté serveur depuis un de vos services pour obtenir le jeton final qui vous permettra de faire les requêtes aux Web

SharePoint Framework Web Part Events



9

SharePoint Framework Roadmap



10

API. C'est en preview, à découvrir dans la petite vidéo de Tim sur "[Add-in Office SSO](#)" et dans [sa documentation déjà disponible](#). Il faudra bien entendu configurer son application dans Azure AD, et mentionner son identifiant/URL et les scopes de permissions dans le manifest de l'add-in. Vous trouverez ci-dessous le schéma général du fonctionnement du SSO : [11]

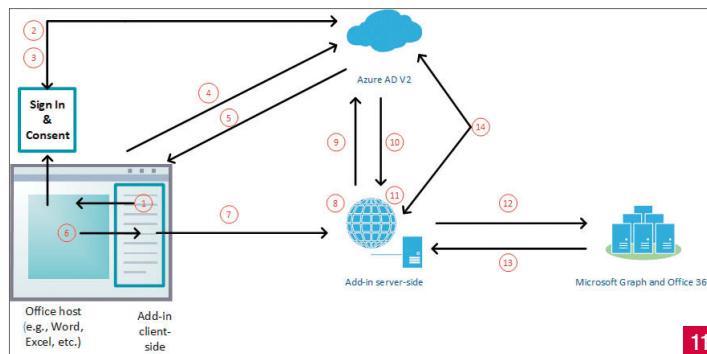
Autre petite nouveauté découverte dans la session "[What's new in Office add-in UX](#)" animée par Humberto Lemaza, la possibilité de ré-ouvrir automatiquement un Taskpane de notre add-in. Ainsi, en cas d'utilisation de votre add-in, il est possible pour ce dernier de "marquer" le document pour lui indiquer d'afficher ce panneau la prochaine fois que le document est ouvert. Il faudra pour cela modifier le manifest de l'add-in pour indiquer que le panneau est à ouvrir automatiquement, et côté code (Office.js) modifier le document pour ajouter la propriété "Office.AutoShowTaskpaneWithDocument". Si vous voulez tester, c'est par ici : <http://aka.ms/autoopen>.

Microsoft Graph

Du côté du Graph, ce sont principalement des annonces de passage de beta en GA (pour Planner, SharePoint Site, OneNote), mais nous avons aussi l'arrivée de l'API pour Teams (qui vient enrichir l'API Groups) afin de consulter les channels, en créer ou envoyer un message sur l'un d'entre eux, et l'extension de l'Insight API qui ajoute un peu d'intelligence et d'analyse sur les interactions du graphe avec l'ajout des API "Used" (les derniers documents vus ou modifiés) et "Shared" (les éléments que l'on m'a partagé). [12]

Vous retrouverez les différentes vidéos de la Build concernant Office sur [Your Guide to Office at Build 2017](#).

Les nouveautés que j'attendais le plus sont clairement le SSO pour les add-ins Office et la simplification des appels au Graph depuis le SPFx, n'hésitez pas à me faire part des vôtres dans les commentaires.



11

PROJECT ROME

Lors de la Build 2017, Microsoft a annoncé le projet Rome ayant pour but de permettre de nouvelles interactions entre les applications et les appareils. Voici un résumé de la session consacrée à ce sujet.

Présentation de Rome

Le projet Rome est un toolkit permettant à une application sur un périphérique client (local) d'interagir avec des applications et des services sur un périphérique hôte (distant). Cela permet aux développeurs de concevoir des applications cross-device et cross-platform centrées autour de tâches utilisateur. Voici plusieurs scénarios présentés lors de la conférence :

Application de visualisation de photos :

Imaginons que vous utilisez une application de visualisation de photos sur votre appareil mobile (iOS, Android ou Windows) et que vous souhaitez afficher vos photos sur un écran plus grand afin de profiter votre entourage, l'application va pouvoir lancer la même application sur votre PC Windows 10 en affichant la même photo qui est sur votre téléphone. Vous pourrez également passer d'une photo à une autre sur votre mobile, et votre PC fera de même automatiquement.

Lancement d'un site Web sur un appareil distant :

De la même manière, imaginons que vous utilisez une application sur votre smartphone qui vous invite à visiter un site Web. Vous pouvez demander à ce que le lien Web s'ouvre sur votre PC pour une navigation plus agréable sans avoir à recopier le lien manuellement. L'application va utiliser les API de Microsoft pour ouvrir le lien Web avec le navigateur par défaut de votre PC. Beaucoup d'autres scénarios sont possibles, comme le jeu en local, le lancement d'une application avec un paramètre sur un appareil distant, démarrage d'un appel Skype, et bien d'autres.

API Graph

Le projet Rome fonctionne aussi avec les API Graph de Microsoft. Ces API permettent d'utiliser les services de Microsoft avec de simples appels REST. Grâce à cela, vous pouvez lancer une application sur votre PC Windows par une requête REST.

Un peu de code

Les API du projet Rome sont très simples d'utilisation, voici plusieurs exemples :
Découverte d'un appareil par son IP :

```
private async Task<RemoteSystem> getDeviceByAddressAsync(string IPAddress)
{
    // construct a HostName object
    Windows.Networking.HostName deviceHost = new Windows.Networking.HostName(IPAddress);
    // create a RemoteSystem object with the HostName
    RemoteSystem remotesys = await RemoteSystem.FindByHostNameAsync(deviceHost);
    return remotesys;
}
```

Lancement d'une application sur un appareil distant :

```
await RemoteLauncher.LaunchUriAsync(
    new RemoteSystemConnectionRequest(remoteDevice),
    new Uri("bingmaps:?cp=47.6204~-122.3491&sty=3d&rad=200&pit=75&hdg=165"));
```

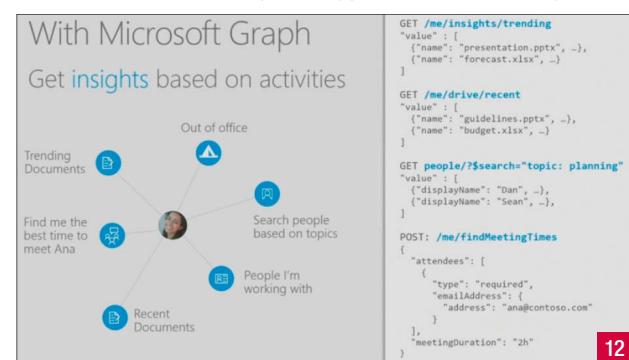
Liens utiles

Documentation du projet Rome : <https://docs.microsoft.com/en-us/windows/uwp/launch-resume/connected-apps-and-devices>

Page GitHub : <https://github.com/Microsoft/project-rome>
Session Build 2017 : <https://channel9.msdn.com/Events/Build/2017/88025>

Conclusion sur le projet Rome

Le projet Rome paraît très prometteur pour créer des interactions entre différents appareils, cependant, les appareils distants ne peuvent être que des appareils Windows 10, ce qui réduit



12

le champ des possibles pour les scénario cross-device.

XAMARIN LIVE PLAYER

Xamarin Live Player est un outil annoncé il y a quelques jours à Build 2017 qui semble très prometteur car il permet de voir le rendu de son application en temps réel pendant le développement.

Comment l'utiliser

Types de projets supportés

Xamarin Live Player est disponible pour les application iOS, Android et Xamarin Forms.

Utilisation

Téléchargez l'application Xamarin Live Player sur [l'App Store](#) ou le [Play Store](#) ;

Dans Visual Studio for Mac, cliquez sur **Tools > Manage devices... : [13]**

Scannez le QR Code avec votre appareil mobile pour le coupler avec Visual Studio ;

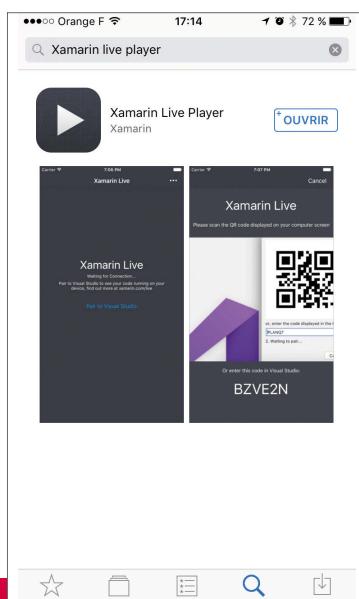
Sélectionnez l'appareil cible « Xamarin Live Player » : [14]

Et vous pouvez exécuter l'application sur votre appareil !

Configuration du debugger

Vous avez trois configurations possibles pour exécuter l'application :

- **Start Without Debugging** : vous pouvez modifier l'application et voir les changements apparaître après la sauvegarde du fichier ;
- **Start Debugging** : vous pouvez utiliser les points d'arrêts mais vous ne pouvez pas modifier le code ;
- **Live Run Current View** : vous pouvez modifier l'application et voir les modifications



directement dans le player. C'est la vue courante qui est affichée.

Les prérequis

Xamarin Live Player est encore en preview et disponible sur Visual Studio 2017 (sur Windows) et sur Visual Studio for Mac.

Vous devez faire la mise à jour sur le canal Alpha pour disposer de la fonctionnalité Xamarin Live Player.

Pour de plus amples informations sur la marche à suivre pour mettre à jour Visual Studio sur le canal Alpha, consultez le lien suivant : https://developer.xamarin.com/recipes/cross-platform/ide/change_updates_channel/

Limitations

Cet outil étant encore en preview, il a certaines limitations comme :

- Sur Android, les UI faites avec AXML ne sont pas encore utilisables ;
- Certaines fonctionnalités des Storyboard iOS ne sont pas utilisables ;
- Les XIB iOS ne sont pas utilisables ;
- Support limité de System.Reflection (cela affect certains NuGet comme SQLite ou Json.NET) ;
- Certaines classes système ne peuvent être passées outre ;
- Certaines fonctionnalités spécifiques aux plateformes ne pourront pas fonctionner dans l'application Xamarin Live Player (mais elle a été configurée pour des opérations courantes, comme l'accès à la caméra).

Toutefois, Xamarin Live Player est en preview et les équipes Xamarin travaillent dessus pour supprimer ces limitations.

Retour d'expérience

J'ai testé Xamarin Live Player avec le projet exemple proposé dans la documentation.

J'ai tout d'abord été impressionné par la simplicité de connecter son device à Visual Studio : c'est instantané et fonctionnel.

En revanche, pour la partie debug, je n'ai réussi qu'à utiliser le mode Without Debugger. Dans

le mode Live Run Current View, j'avais beau modifier mon fichier, l'application ne se mettait pas à jour. N'oublions pas que ce produit est encore en preview, et que des améliorations sont à venir. Néanmoins, Xamarin Live Player reste très pratique pour tester son application sur un vrai device (en opposition au simulateur) de façon très simple et très rapide.

Liens utiles

Documentation de Xamarin Live Player : <https://developer.xamarin.com/guides/cross-platform/live/>

Liens vers la session :

<https://channel9.msdn.com/Events/Build/2017/C9L24>

La sortie de cette fonctionnalité est une très bonne nouvelle d'autant plus qu'il existe depuis longtemps des outils similaires pour les technologies concurrentes (comme React native).

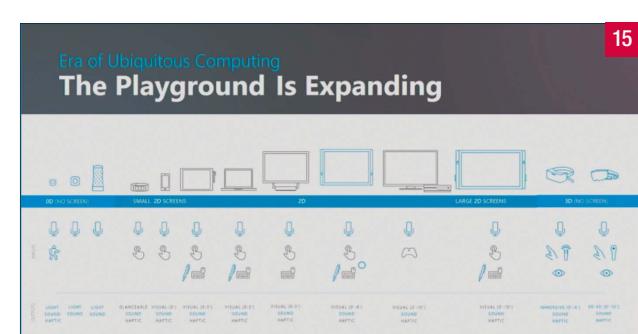
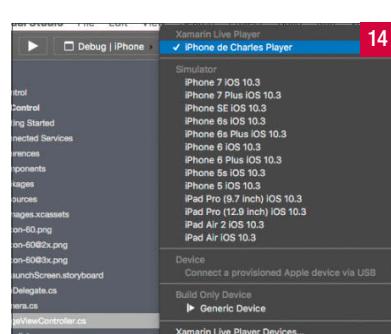
Le fait qu'il soit possible de tester son application sur son appareil mobile sans fils, sans compte développeur Apple (pour iOS), et sans temps de compilation, rend le développement beaucoup plus agréable.

FLUENT DESIGN

Microsoft vient d'annoncer de nouvelles guidelines en termes de design et d'ergonomie. **Il s'agit d'une grande étape pour Microsoft qui quitte le monde du flat pour entrer dans un nouveau paradigme immersif et multi-dimensionnel qu'il introduit sous le nom de "Fluent Design" aussi connu sous le nom de code "Neon"...**

Pourquoi est-ce important ?

Tout d'abord car le monde de l'interaction homme-machine au sens large du terme est en train d'évoluer : les outils/périphériques/unités de calculs sont de plus en plus divers, répandus et interconnectés (**Microsoft parle d'Ubiquitous Computing**). Nous sommes habitués aux écrans plats (montres, téléphones, ordinateurs, télévisions, etc.) mais la réalité virtuelle ou mixte commence à prendre de l'ampleur. Il ne faut pas non plus



oublier les interfaces à ... 0 dimensions que sont les bots ou les périphériques commandés par la voix (Alexa, Cortana, etc.).

Ces différentes interfaces sont aussi à considérer en termes d'interactions avec l'utilisateur : via la voix, via le toucher, via un clavier, via un stylo, via des accessoires dédiés, gestes visuels, etc. [15]. Cela devient encore plus complexe lorsque l'on envisage que les utilisateurs passent d'un device à un autre et souhaitent garder le même contexte !

Et deuxièmement parce que les attentes des utilisateurs en termes de créativité deviennent de plus en plus grandes. Il n'y a qu'à regarder le succès des filtres Snapchat ou Facebook pour en avoir un exemple. Les gens ne communiquent maintenant pas uniquement via la voix mais surtout en racontant des histoires. Les utilisateurs s'attendent à retrouver cette créativité au sein des interfaces applicatives qu'ils utilisent : immersion, expériences sensorielles, quel que soit l'usage (pro ou particulier).

Quels sont les axes de travail ?

Microsoft propose de travailler autour de 5 grands axes. [16]

- Lumière / Light : permet de montrer facilement à l'utilisateur où il se trouve, ce qu'il regarde. Cela permet aussi de créer une certaine dimension émotionnelle plus immersive.
- Profondeur / Depth : cela est massivement utilisé dans les jeux vidéo et il est naturel de le retrouver dans les applications.
- Mouvement / Motion : permet de donner de la vie dans tout ce que l'on fabrique. Permet aussi de donner du contexte.
- Etoffe / Material : donne de la chaleur à une interface et donne envie de la toucher. Permet aussi d'instiller une hiérarchie visuelle.
- Echelle / Scale : il est bien sûr important de faire une interface capable de s'adapter à n'importe quelle taille d'écran ou au monde physique.

Comment va se manifester ce langage ?

Microsoft va commencer à intégrer ce nouveau langage visuel à partir de la mise

à jour "Fall Creator Update" mais va le déployer par vagues. Ce que nous voyons maintenant est donc la première et les suivantes en seront des évolutions basées notamment sur le retour des utilisateurs, mais aussi les nouveaux usages émergeants.

La première vague est constituée notamment de plusieurs éléments :

- **Une nouvelle animation de survol des éléments actifs** (notamment les boutons) : on passe d'un mode binaire(actif/inactif) à une utilisation de la lumière pour montrer l'élément actif. Dans la capture d'écran de la calculatrice on peut par exemple voir comment la bordure des boutons actifs autour du 5 s'illuminent pour montrer à l'utilisateur qu'ils sont cliquables. Cela sera intégré de base dans les contrôles du Framework.
- **Acrylic material** : le fameux fond "semi-flou" qui permet de voir le fond d'écran par derrière l'application. Microsoft nous promet que beaucoup de choses se passent sous le capot pour produire cet effet visuel.
- **Connected animations** : déjà présent dans les précédentes versions, les animations connectées deviennent la norme pour passer d'un écran à un autre.
- **Conscious control** ; faire que les interfaces graphiques aillent plus loin que d'être simplement responsive et qu'elles aient conscience de ce que veut faire l'utilisateur. Ce sont notamment des petites touches à droite et à gauche comme l'élargissement d'une scrollbar lorsque l'utilisateur approche sa souris ou encore des zones de textes qui s'agrandissent lorsque l'utilisateur approche un stylet au lieu d'utiliser son clavier.
- **Perspective Parallax** : donner de la profondeur et un sentiment de hiérarchie à vos interfaces. On retrouve ici des choses déjà vues sous Windows 8... et qu'on appréciait :) [17]

Et après ? [18]

Microsoft a déjà en tête les prochaines évolutions de ses guidelines avec notamment :

- La lecture de vidéos ou images à 360° : comment rendre cela intuitif pour les utilisateurs

quel que soit son périphérique (écran plat versus HoloLens par exemple) ?

- Conscious headers : rendre les entêtes de listes dynamiques un peu à la manière de ce que l'on retrouve sur iOS.
- Speech, son spatial, Z-depth layering, etc.

Par où commencer en tant que développeur / designer ?

Pour éviter de se retrouver un peu perdu devant ce nouveau système de design, Microsoft propose de nous guider via la plateforme elle-même (applications natives, les contrôles du SDK et les samples) mais aussi de la documentation et des exemples.

Tout se trouve sur <http://fluent.microsoft.com/> !

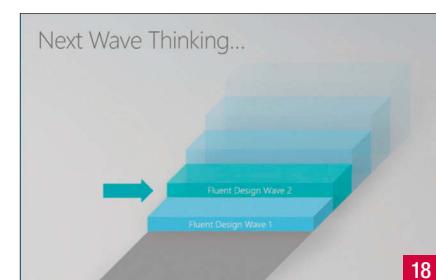
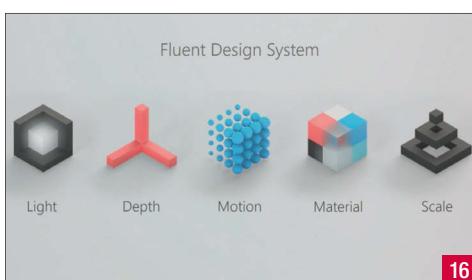
Plus important encore peut-être, Microsoft a travaillé avec Adobe pour intégrer directement le langage Fluent au sein de leurs outils et notamment au sein d'Adobe XD.

Quel impact pour nous en tant que développeur

L'impact de ce nouveau design va rester somme toute minime puisque l'intégration au niveau des différents composants du Framework sera faite automatiquement par Microsoft dans le SDK. Pas de soucis de ce côté donc et cela sera autant d'exemples à suivre et autant de composants à réutiliser pour construire nos interfaces. Le design responsive prend cependant de plus en plus d'importance et il devient difficile de passer outre : il est donc temps de réviser nos connaissances sur le sujet et peut être de dialoguer encore plus avec nos designers pour intégrer ces concepts dès le départ d'un projet.

Que retenir ?

Au premier abord ce sont des modifications minimes au sein de Windows (et Microsoft au sens large) mais on peut se rendre compte qu'il s'agit ici d'un changement plus profond pour quitter (enfin diront certains) le monde de l'aplat si cher à Microsoft jusqu'ici. La stratégie ici tenue de ne pas tout changer d'un bloc mais d'y aller par étapes ne peut que s'avérer payante. J'ai hâte de retrouver ces designs dans Windows et ... nos applications !



Automatissez-vous la vie avec **Gulp** !

• Fabien Pigere

À l'heure actuelle, la concurrence sur Internet est énorme, être visible devient d'une importance primordiale. Le temps de chargement initial d'un site à la première consultation peut faire la différence. En effet de plus en plus d'utilisateurs consultent les sites à partir de tablettes/téléphones, avec une connexion pas forcément au top (WiFi ou 3G), comme nous le voyons figure 1, la 3G est normale, mais 100ms à faire une requête, soit 10 requêtes par seconde, ce qui est lent, vu de la part de l'utilisateur. De plus, de nos jours, le client est impatient et n'hésite pas à zapper entre les sites si le site ne s'affiche pas très vite. Nous allons voir comment, grâce à Gulp, accélérer le temps de chargement entre deux et quatre fois, sans trop de complications !

Le truc est très simple en fait : au chargement de la page HTML, le navigateur lance des requêtes pour chaque ressource externe à charger JS, CSS, image, SVG ...) ce qui prend du temps, surtout que les navigateurs limitent le nombre de requêtes en parallèle, ralentissant encore plus l'affichage final. Moins il y a de requêtes, mieux c'est ! C'est là que Gulp agit : il permet de minifier, concaténer les sources, et beaucoup plus, étant extensible par système de plug-in, que nous verrons à la fin.

Qu'est-ce que Gulp ?

Gulp est une boîte à outils JavaScript open source, bâtie sur Node et NPM pour automatiser les tâches répétitives du développement Web, comme la minification, le linting (vérification statique de la qualité des sources), l'optimisation, les tests unitaires, etc. Nous nous intéresserons surtout à l'optimisation.

Pour pouvoir utiliser Gulp, il faut avoir Node et NPM installé bien sûr, qui s'installe à partir de cette URL [1].

Lancez une ligne de commandes dans votre projet, puis tapez : node -v puis npm -v pour vérifier que l'installation s'est bien passée. Puis il faut installer Gulp : npm install Gulp

Gulp est maintenant utilisable !

Gulp est par lui-même assez nu, il a cependant un écosystème de plus de 300 plug-ins, dont nous allons voir les plus utiles. Commençons par le plug-in de concaténation, le plus simple, mais le plus utile au final, en effet grâce à lui, notre site de test passe de 12 fichiers JavaScript à un seul fichier, et donc 11 requêtes de gagnées !

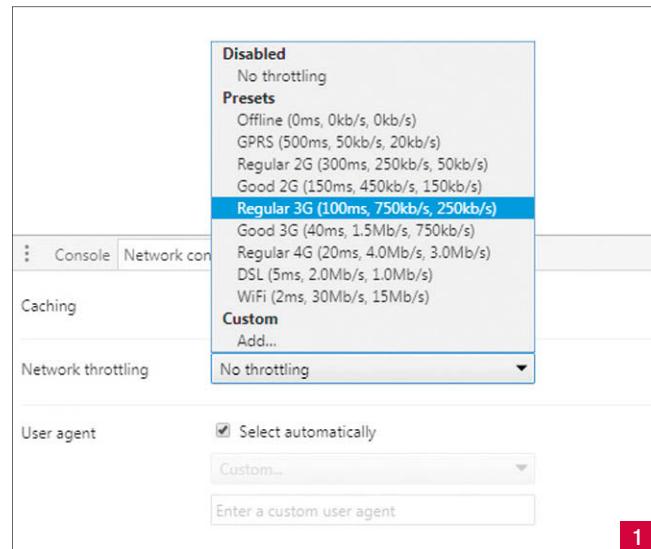
Pour l'installer : npm install -save-dev gulp-concat

Maintenant, il faut créer un fichier gulpfile.js, qui contiendra nos actions :

```
var gulp = require('gulp');
var concat = require('gulp-concat');

gulp.task('minjs', function () {
  return gulp.src("./js/*.js")
    .pipe(concat('all.js'))
    .pipe(gulp.dest('dist'))
    .pipe(gulp.dest('dist'));
});

gulp.task('default', function () {
  gulp.watch("./js/*.js", ["minjs"]);
});
```



1

Détaillons cette base de travail :

- On commence par dire à Node de charger Gulp&Gulp-concat. Puis on déclare une tâche appelée minjs, qui prend tous les JS du répertoire "js" et les transforme en dist/all.js. On passe de 23 requêtes à une seule !
- À la fin on définit la tâche par défaut, et on demande à Gulp de lancer minjs si un fichier js est modifié grâce à watch(). C'est tout simple et puissant !

Pour tester, tapez simplement gulp dans le répertoire. Attention la tâche se lance au changement des sources, tapez donc un espace dans ceci, puis sauvegardez, pour que la magie opère !

Minifier le JS

Rajoutons un autre plug-in pour minifier les sources et gagner encore un peu de bande passante :

Il faut installer le module Uglify : npm install -save-dev gulp-uglify

Puis modifier légèrement notre tâche :

```
gulp.task('minjs', function () {
  return gulp.src(files)
    .pipe(concat('all.js'))
    .pipe(gulp.dest('dist'))
    .pipe(uglify())
```

```

  .pipe(gulp.dest('dist'));
});

```

Maintenant notre JavaScript est optimal ! On passe de 3850 ko à ...434 !
Soit un gain de plusieurs secondes en 3G, yeah !

Passons au CSS

Nous allons faire de même avec le CSS, avec un petit bonus ; nous allons auto préfixer le CSS, vous savez, rajouter les -moz, -webkit et compagnie, et ceci automatiquement, et optimiser le CSS . Nous avons besoin de deux plug-ins pour cela :

```
npm install --save-dev gulp-clean-css
npm install --save-dev gulp-autoprefixer
```

Et d'une nouvelle tâche :

```

var autoprefixer = require('gulp-autoprefixer');
var cleanCSS = require('gulp-clean-css');

```

```

gulp.task('mincss', function () {
  return gulp.src("css/*.css")
    .pipe(autoprefixer())
    .pipe(concat('all.css'))
    .pipe(cleanCSS())
    .pipe(gulp.dest('dist'));
});

gulp.task('default', function () {
  gulp.watch(["js/*.js", "css/*.css"], ["minjs", "mincss"]);
});

```

De même que tout à l'heure nous sommes passé de 10 requêtes à une, soit pour l'instant au total, 31 requêtes en moins, c'est énorme, 3 secondes de 3G !

Et maintenant, le HTML

Le HTML peut lui aussi être minifié, pourquoi s'en priver ? Là-aussi nous avons besoin d'un plug-in :

```
npm install --save-dev gulp-html-minifier
```

Et d'une nouvelle tâche :

```

var htmlmin = require('gulp-html-minifier');

gulp.task('minhtml', function () {
  return gulp.src('*HTML')
    .pipe(htmlmin({collapseWhitespace: true}))
    .pipe(gulp.dest('./dist'));
});

gulp.task('default', function () {
  gulp.watch(["*HTML", "js/*.js", "css/*.css"], ["minjs", "mincss", "minhtml"]);
});

```

Et après ?

Nous pourrions nous arrêter là, mais, en regardant de près, nous voyons plusieurs problèmes en fait :

- 1) L'ordre de concaténation n'est pas connu, ce qui pourrait être très dangereux pour des CSS ou JS en cas de surcharge.
- 2) Le HTML final contient bien sûr les balises script/css originales, qui ne nous servent plus à rien.
- 3) De même le HTML final n'inclut pas all.js ni all.css ; nous devons les modifier à la main, ou prévoir un HTML différent en développement et en production.
- 4) Les tâches Gulp sont exécutées en parallèle, sans ordre précis, ce qui pourrait être dangereux.
- 5) Nous avons actuellement deux fichiers all.css et all.js, soit deux requêtes. Nous pourrions les faire disparaître...en les incluant directement dans le HTML !

Ceci est un des problèmes classiques de Gulp. Nous allons voir comment corriger tout cela, grâce à notre propre plug-in et à un module Node appelé Cheerio.

Cheerio vient à notre secours !

Qu'est-ce que Cheerio ? C'est l'équivalent de Jquery dans Node, à l'origine surtout destiné au scraping (extraire les informations d'une page Web distante), il va nous être d'un grand secours pour tous les problèmes listés. Commençons par la classique installation :

```
npm install --save-dev cheerio
```

Puis le script permettant d'avoir exactement les bons fichiers à traiter, ceci dans le bon ordre cette fois

(pour simplifier, nous nous baserons sur le fichier index.html seulement, dans le cas d'une SPA) :

```

var cheerio = require('cheerio');

var files = [];
var filesCSS = [];
var txt = fs.readFileSync("index.html");
var $ = cheerio.load(txt);

$("script").each(function () {
  var file = $(this).attr("src");
  console.log(file);
  if (file && file.indexOf("http") === -1)
    files.push(file);
});

$("link").each(function () {
  var file = $(this).attr("href");
  console.log(file);
  if ($(this).attr("rel") === "stylesheet" && file && file.indexOf("http") === -1)
    filesCSS.push(file);
});

var all = ["gulpfile.js", "index.html"].concat(files, filesCSS);

gulp.task('minjs', function () {
  return gulp.src(files)
    .pipe(concat('all.js'))
    .pipe(gulp.dest('dist'))
    .pipe(uglify())
    .pipe(gulp.dest('dist'));
});

```

```
});

gulp.task('mincss', function () {
  return gulp.src(filesCSS)
    .pipe(autoprefixer())
    .pipe(concat('all.css'))
    .pipe(cleanCSS())
    .pipe(gulp.dest('dist'));
});

gulp.task('minhtml', ['mincss', 'minjs'], function () {
  return gulp.src('*HTML')
    .pipe(injectSvg())
    .pipe(htmlmin({collapseWhitespace: true}))
    .pipe(p("dist/all.js", "dist/all.css"))
    .pipe(gulp.dest('./dist'));
});

gulp.task('default', function () {
  gulp.watch(all, ["minjs", "mincss", "minhtml"]);
});

```

Maintenant nous ne regardons QUE les fichiers qui nous intéressent, et nous les traitons dans le bon ordre, et la tâche HTML dépend des deux tâches mincss et minjs, il s'exécutera seulement lorsque les deux seront finis grâce à "gulp.task('minhtml', ['mincss', 'minjs'], function ()...)" qui indique à Gulp les dépendances comme second paramètre.

Pour les problèmes suivants, nous allons écrire notre propre plug-in, qui devra :

- Retirer les balises d'inclusion de CSS/JS (problème 2) ;
- Ajouter le all.js et all.css directement dans le HTML final.

Écrivons donc plugin.js :

```
var through = require('through2'); // npm install --save through2
var cheerio = require('cheerio');
var fs = require('fs');

// consts
var PLUGIN_NAME = 'gulp-example';

module.exports = function (jsn, cssn) {
  return through.obj(function (file, encoding, callback) {
    {
      var mybuf = file.contents.toString('utf-8');
      var $ = cheerio.load(mybuf);
    }
  });
}
```

```
$("script").each(function () {
  var $this = $(this);
  if ($this.attr("src") && $this.attr("src").indexOf("http") === -1)
    $this.remove();
});

$("link").each(function () {
  var $this = $(this);
  if ($this.attr("href") && $this.attr("href").indexOf("http") === -1)
    $this.remove();
});

var js = fs.readFileSync(jsn);
var css = fs.readFileSync(cssn);
$("head").append("<style>" + css + "</style>");
$("body").append("<script>" + js + "</script>");
file.contents = new Buffer($HTML);
return callback(null, file);
}, jsn, cssn);
};


```

Détail : Through2 est une librairie simplifiant le travail d'écriture de plug-ins. Nous voyons donc que nous parcourons notre flux, nous retirons toutes les balises de script (en faisant bien attention à ce que cela ne soient pas des balises inline), toutes les balises de style, avec la même précaution, puis nous injectons le CSS dans le header, et les scripts en fin du BODY ce qui est une bonne pratique. Utilisons maintenant notre plug-in :

```
var p = require('./plugin');

gulp.task('minhtml', ['mincss', 'minjs'], function () {
  return gulp.src('*HTML')
    .pipe(injectSvg())
    .pipe(htmlmin({collapseWhitespace: true}))
    .pipe(p("dist/all.js", "dist/all.css"))
    .pipe(gulp.dest('./dist'));
});

```

EN CONCLUSION

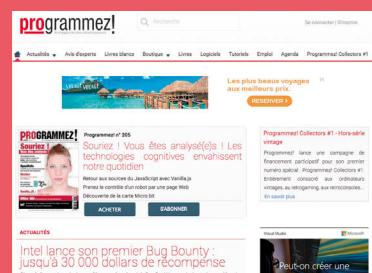
L'optimisation du chargement est très importante de nos jours, Gulp aide à cela d'une façon simple et extensible, et comme nous l'avons vu, allié à Cheerio, il devient d'une puissance redoutable. Pour la petite histoire, sur mon dernier site, j'ai pu optimiser de 350% le temps de chargement de la page principale. Je vous laisse écrire de meilleurs plug-ins (mais vérifiez quand même qu'il n'existe pas sur le site NPMJS.COM avant !).

[1] <https://nodejs.org/en/download/>

Restez connecté(e) à l'actualité !

- **L'actu** de Programmez.com : le fil d'info quotidien
- **La newsletter hebdo** : la synthèse des informations indispensables.
- **Agenda** : Tous les salons, barcamp et conférences.

Abonnez-vous, c'est gratuit ! www.programmez.com



Architecture Serverless avec les Azure Functions



Laurent Yin
Consultant
Cellenza
Cloud, DevOps,
Intégration



Manon Pernin
Consultante **Cellenza**
Cloud, DevOps,
Intégration

cellenza
DOES IT BETTER | Consultant Expertise Microsoft & méthodes agiles

Pour autant, la valeur ajoutée d'une entreprise par rapport à une autre se porte sur les étapes de **conception fonctionnelle et technique** puis de **développement**. C'est sur ces étapes qu'un investissement de temps est nécessaire pour pouvoir proposer les meilleures solutions. Le reste doit pouvoir soit être automatisé, pour les phases de tests et de déploiement par exemple, soit être abstrait pour ce qui est de l'allocation des ressources. [1]

Abstraction

Avec la virtualisation, le passage sur des VMs a été le début de l'abstraction des ressources physiques sous-jacentes. En passant sur le Cloud, une nouvelle étape a été franchie puisque le provisionnement des VMs en mode IaaS est facilité. Nous avons ensuite vu apparaître la notion de PaaS qui a ajouté une nouvelle couche d'abstraction : plus besoin de gérer le middleware.

Néanmoins, une partie de la gestion de la capacité des briques PaaS reste à la charge des clients : surveiller l'utilisation des ressources physiques (CPU, mémoire, ...), ajuster l'allocation des ressources selon l'utilisation, mettre en place du scaling automatique, horizontal ou vertical. De plus, **la facturation de ces briques PaaS est basée sur les ressources allouées et non sur les ressources consommées**. Maintenant, nous entendons parler de **Serverless** qui propose un nouveau niveau d'abstraction associé à un nouveau mode de facturation.

Les briques Serverless sont en effet conçues pour gérer nativement l'allocation de ressources physiques et donc le scaling automatique. En utilisant ces briques, les équipes IT peuvent se concentrer sur les fonctionnalités à implémenter.

Un des enjeux des entreprises aujourd'hui est de réduire le *Time-To-Market*, c'est-à-dire le délai de mise sur le marché d'une idée. Dans le cas d'une application, on veut par exemple pouvoir proposer au plus vite une fonctionnalité aux utilisateurs si le besoin a été identifié. Le *Time-To-Market* est donc dans ce cas-là le temps dédié à la conception fonctionnelle, la conception technique, le développement, les tests puis le déploiement de l'application.

Evolution du Monitoring et de la facturation

Certes le monitoring va pouvoir être simplifié mais il ne doit pas pour autant être oublié ! Afin de suivre l'usage des fonctionnalités mises en place, on va se concentrer sur les événements et les durées d'exécution. Ces deux métriques sont également prises en compte pour établir la facturation.

Que propose Microsoft Azure ?

De nombreux services Azure vont vous permettre de mettre en place une architecture Serverless, mais les Azure Functions constituent le service essentiel de ce type d'architecture pour sortir des scénarios classiques et pouvoir mettre en place des fonctionnalités spécifiques à son entreprise.

Azure Functions

Une Azure Function est un service Azure permettant d'héberger des bouts de code, des fonctions, sans embarquer une application dans sa globalité.

Pour créer des « Azure Functions » dans Azure, il faut commencer par créer une « Function App ». Il s'agit d'un service basé sur une Web

App Azure (App Service) au sein duquel on pourra créer plusieurs fonctions. On

peut voir ci-contre une Function App avec ses deux fonctions.

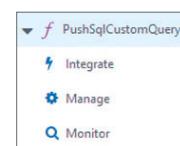
Scénarios

Un des atouts est d'offrir un **large choix de langages** : C#, F#, Node.js, Python, PHP, Batch, Bash ou PowerShell. Selon votre besoin, vous pourrez choisir d'exécuter votre fonction en mode planifié, d'exposer votre fonction en tant qu'API REST ou en tant que webhook, ou encore de déclencher une exécution sur la base d'un événement : nouveau message dans un Service Bus par exemple. Ce type de déclenchement est configurable simplement à l'aide des **connecteurs natifs** proposés : Service Bus, Event Hub mais aussi OneDrive, Twilio... De la même façon, il est possible de choisir son point de sortie.

Toutes ces possibilités vous permettent d'utiliser les Azure Functions dans de nombreux cas !

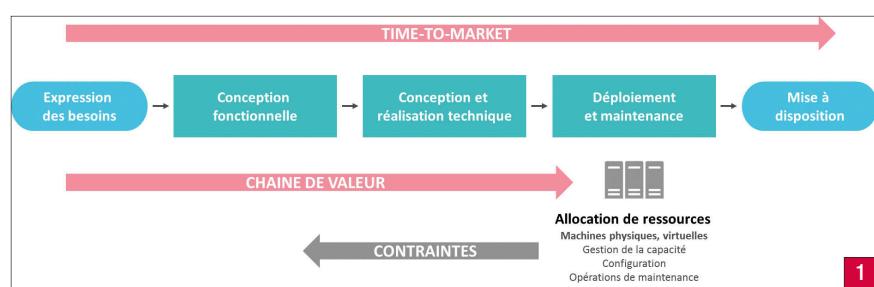
Outillage Portail Azure

Dans le portail Azure, après création d'une fonction, quatre fenêtres sont proposées :



La fonction

C'est ici que vous pourrez coder directement si vous souhaitez mettre en place rapidement une fonction. Vous avez également accès aux



différents fichiers de configuration vous permettant d'ajouter les packages nécessaires ou d'uploader vos propres librairies. C'est ici aussi que vous pourrez tester votre fonction et visualiser les logs d'exécutions. [2]

Integrate

La partie Integrate vous permet de configurer les bindings : triggers, entrées et sorties de la fonction.

Manage

Dans cette partie, vous pouvez choisir de désactiver votre fonction, de la supprimer ou encore de gérer les clés d'accès pour appeler la fonction.

Monitor

Cette dernière partie vous donne la liste des dernières exécutions avec leur statut et leur durée. Les logs de chacune de ses exécutions sont également disponibles. [3]

Remarque : Comme évoqué précédemment, une Function App est basée sur une Web App. Vous retrouverez donc un grand nombre de configurations semblables aux Web Apps. Il est, par exemple, possible de définir des paramètres d'applications qui seront communs à toutes les fonctions de la Function App.

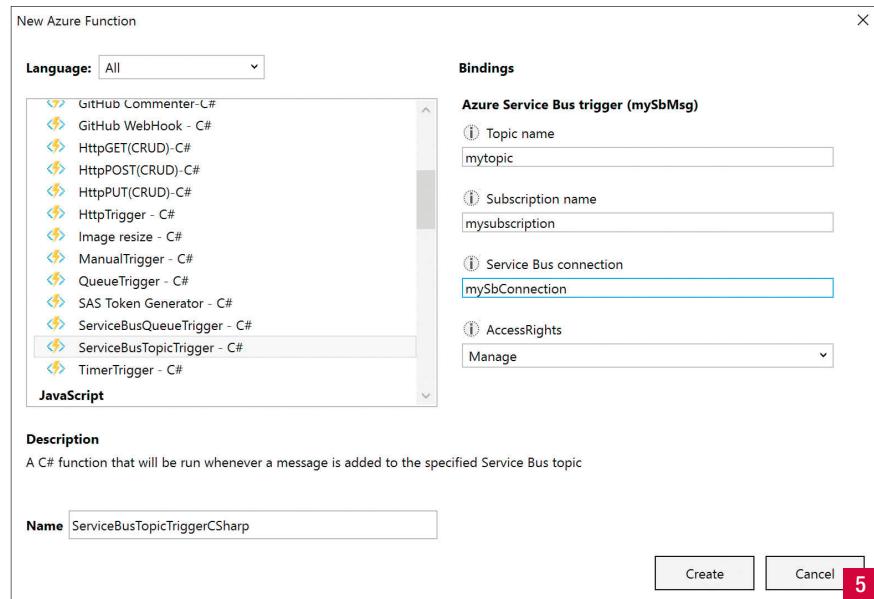
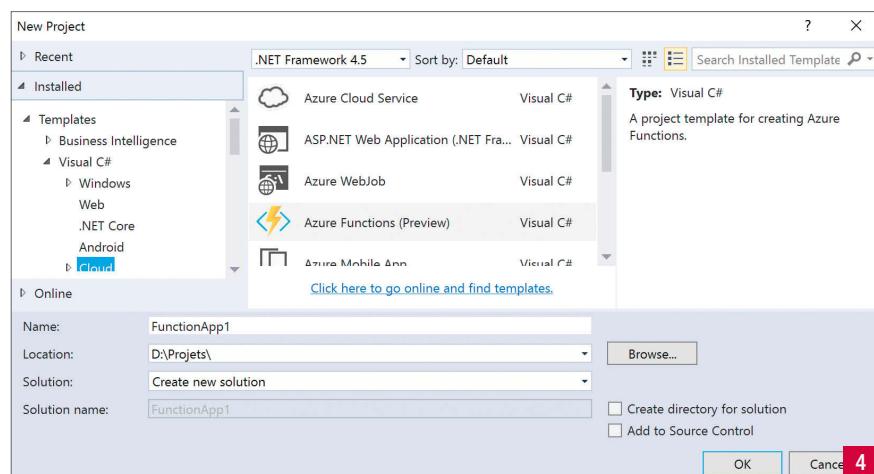
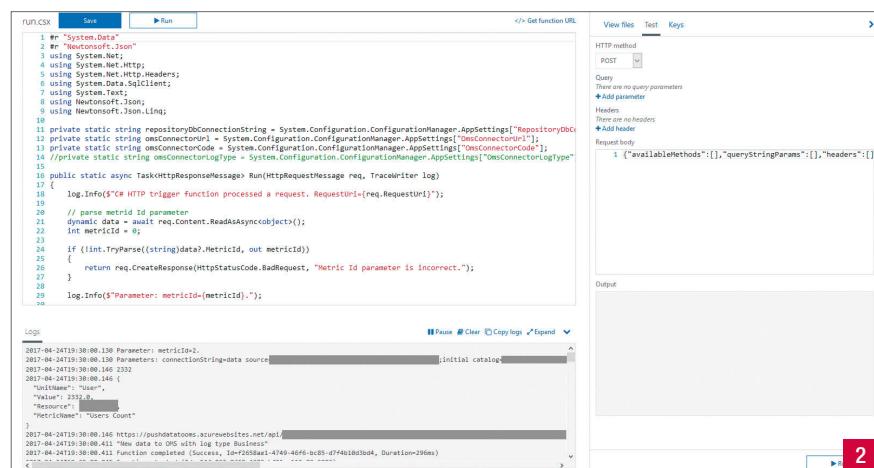
Développement avec Visual Studio

Pour aller plus loin dans le développement avec le service Azure Functions, vous avez la possibilité d'exploiter le module **Visual Studio**.

Tools for Azure Functions. Les avantages sont multiples : vous pouvez tout d'abord profiter de l'interface de développement sur laquelle vous avez l'habitude de travailler. Cela vous apportera alors l'autocomplétion avec IntelliSense, les tests sur un environnement local, ou encore la possibilité d'archiver plus facilement votre code en reprenant vos méthodes standards. En effet, une fois l'outil installé, les projets de type Azure Functions seront directement accessibles parmi les choix disponibles. [4]

Au niveau du projet, le menu contextuel propose alors l'ajout des fonctions. Les templates sont directement listés pour vous permettre de créer rapidement des scénarios d'usage commun. L'interfaçage avec des services comme Azure Storage, Azure Service Bus sont facilités par le paramétrage des connectivités externes (ou bindings). [5]

D'autre part, si vous souhaitez déployer le code sur Azure, il est possible d'utiliser la fonctionnalité de déploiement native en cliquant sur « Publier » dans le projet Azure Functions en question. L'interface se charge alors de récupérer automatiquement les Functions Apps dispon-



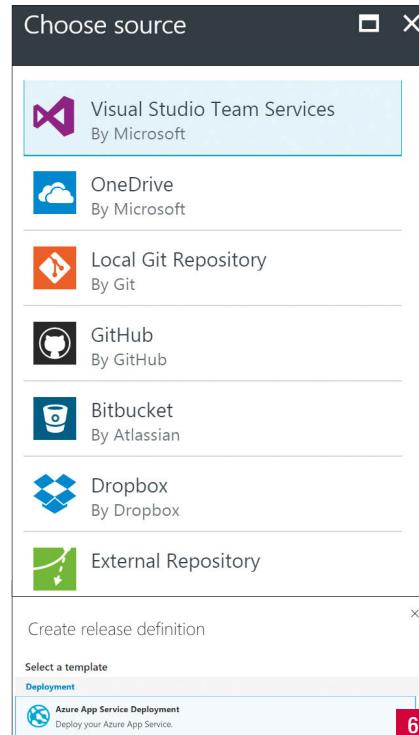
nibles sur votre souscription Azure. Vous pouvez également en créer directement si nécessaire.

Vous remarquerez que le déploiement des Azure Functions sous Visual Studio s'apparente fortement au déploiement des Web Apps Azure. En effet, le développement des Azure Functions se base sur le WebJobs SDK, c'est-à-dire celui utilisé pour la création de tâches déployées à l'origine sur Azure Web Apps.

Déploiement continu sur Azure Functions

Bien entendu, l'un des challenges auxquels se confrontent les Azure Functions est le déploiement continu. En ce sens, ce service Serverless d'Azure offre la possibilité de gérer son développement réalisé avec les gestionnaires de sources comme Visual Studio (Team Services), Git ou GitHub. Les fonctions sont des composants de code simples à déployer, compte tenu de leur taille légère. Cela peut vite entraîner un nombre de déploiements grandissant, qui s'accroît avec le nombre de fonctions, d'où l'importance de l'outil pour favoriser les bonnes pratiques de déploiement continu. [6]

En complément, si vous souhaitez gérer de manière plus approfondie le déploiement grâce à Visual Studio Team Services, vous pouvez profiter des opérations existantes pour le déploiement d'Azure App Service, les Functions Apps se basant sur les App Services.

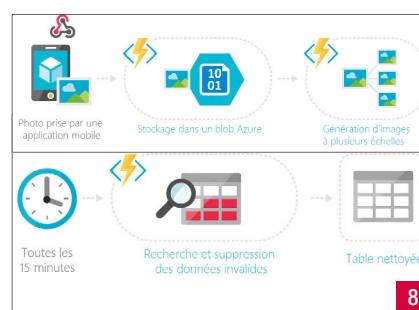


cutent de façon éphémère, ou lorsque la charge n'est pas prévisible pour dimensionner l'infrastructure. En utilisant les fonctions, vous n'avez plus besoin de gérer l'allocation des ressources techniques sous-jacentes, et vous réduisez les coûts liés à la sous-utilisation de la plate-forme. [8]

Problématiques d'intégration

Les scénarios d'intégration présentent également de l'intérêt sur Azure Functions. Avec la croissance du nombre d'APIs disponibles sur le marché, qui offrent des possibilités d'interaction toujours plus vastes, l'ajout de quelques lignes de code permet d'assurer la connectivité avec ces services et de mieux les combiner au sein d'un système d'information. On mutualise alors ce code pour interagir avec les réseaux sociaux, se connecter à la CRM, ou encore requérir les services de recherche, d'analyse et de traitement de données. [9]

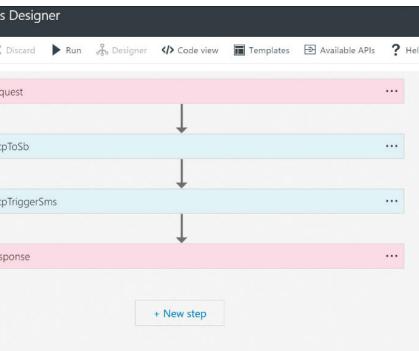
Loin d'aller à l'encontre des concepts de la SOA, les Azure Functions demeurent des services qui peuvent être appelés ou déclenchés au cours d'un processus. En ce sens, Azure Logic Apps intègre nativement les Azure Functions et facilitent la composition des services. La complémentarité entre « bouts de code spécifiques » et connecteurs natifs les fait aller de pair. Les Logic Apps apportent alors une couche logique avec une gestion d'erreurs plus poussée pour mieux suivre la consommation des services qui en est faite.



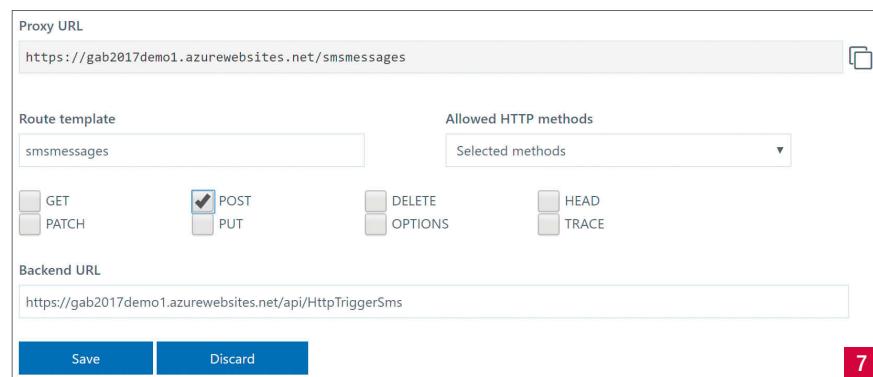
Les Proxies pour enrichir les fonctions Azure

Il est désormais possible d'activer la nouvelle fonctionnalité Azure Function Proxies afin d'exposer un point d'entrée unique pour les différentes Functions Apps ou APIs que vous avez à disposition. Les requêtes clientes n'interrogent alors qu'une unique API en surface, le Proxy, lorsqu'une multitude de vos fonctions vont les traiter en coulisse. De cette façon, vous pourrez mieux organiser les fonctions que vous avez développées, en masquant les différentes URL de vos fonctions grâce au Proxy. [7]

Les proxies se révèlent particulièrement utiles si vous comptez utiliser de plus en plus de fonctions avec une orientation axée microservices. Vous pouvez aussi tirer profit de cette fonctionnalité pour simplifier vos tests, pour basculer d'un environnement sur l'autre par exemple.



Pour conclure, les architectures Serverless se basent sur des nanoservices, tels que les Azure Functions, qui offrent des perspectives intéressantes dans de nombreux cas d'usage : moins de maintenance, concentration sur la fonctionnalité donc réduction du Time-To-Market. **L'enjeu majeur de ces nouvelles architectures sera donc porté sur les problématiques de sécurité et de gestion du cycle de vie des services.**



Cas d'usages

Les Azure Functions dans une architecture serverless sont adaptées aux traitements qui s'exé-

Une méthode « originale » pour bloquer l'accès POP/IMAP à Office 365



Jean-Benoit Paux
Manager Infrastructure
Avanade

La solution Software as a Service (SaaS) de Microsoft, Office 365, devrait en 2017 dépasser les 100 millions d'utilisateurs actifs. Nous avons, ces dernières années, accompagné de nombreux clients lors de cette transition et principalement pour la migration de leur messagerie vers Exchange Online.

Le Cloud offre bien des avantages (moins de serveurs à gérer, moins d'expertises à avoir, un service géré de bout en bout, des innovations apportées plus rapidement par exemple) mais cela est contrebalancé par une standardisation de la configuration de la solution. Toute personnalisation de la solution demande en effet d'être revue régulièrement suite aux nouveautés apportées par l'éditeur. Ces personnalisations doivent aussi potentiellement être gérées utilisateur par utilisateur et donc automatisées en local si l'on veut généraliser cela à l'ensemble des utilisateurs.

Un exemple de cette personnalisation est la configuration des protocoles POP/IMAP dans Exchange Online. Par défaut, tous les utilisateurs ont POP/IMAP activé sur leur boîte aux lettres et peuvent donc s'y connecter de chez eux, de leur smartphone. Le besoin que l'on retrouve couramment en clientèle consiste à désactiver POP/IMAP pour l'ensemble des utilisateurs à l'exception de quelques boîtes de services ou dites « applicatives » car consommées par des applications tierces (logiciels type CRM ou d'ITSM par exemple).

Quand on réfléchit à la manière de ne donner accès à ces fonctionnalités qu'à une sous population, on pense immédiatement à aller trifouiller la console d'administration d'Exchange à la recherche de ces paramètres. Mais si on pouvait faire autrement ?

La méthode « traditionnelle »

La méthode « traditionnelle » consiste à se connecter sur le portail d'administration d'Exchange et à désactiver/activer POP/IMAP sur les boîtes aux lettres : [1]

Cette solution ne fonctionne que lorsqu'on n'a que quelques boîtes à gérer, que l'on crée ses boîtes manuellement et qu'on pensera systématiquement à aller chercher l'option, autant dire que ce sera souvent oublié.

La 2ème solution souvent implantée est l'utilisation conjointe de PowerShell, du Task Scheduler (ou plus moderne d'Azure Automation) et d'un groupe Active Directory ou Azure AD contenant les boîtes aux lettres « autorisées » à faire du POP/IMAP.

En effet, il suffit par exemple de lancer tous les jours un script qui va chercher l'ensemble des boîtes aux lettres, vérifier si l'utilisateur est

general
mailbox usage
contact information
organization
email address
mailbox features
member of
MailTip
mailbox delegation

Email Connectivity
Outlook on the web: Enabled
[Disable](#) | [View details](#)

IMAP: Enabled
[Disable](#)

POP3: Enabled
[Disable](#)

MAPI: Enabled
[Disable](#)

Save Cancel

membre du groupe « O365 Allowed POP/IMAP Users » et lui activer les protocoles POP/IMAP tout en pensant à bien le désactiver pour les autres. Ceci est rendu possible par la commande PowerShell :

```
Set-CasMailbox -Identity MaBoiteActiver -PopEnabled $true -ImapEnabled $true
```

Toute cette mécanique basée sur des groupes est expliquée par Microsoft dans un article de blog disponible ici : <https://blogs.technet.microsoft.com/zarkatech/2011/08/09/automatically-disable-pop3-imap-in-office-365>

Une alternative à cette méthode est d'utiliser un attribut Active Directory pour taguer les boîtes aux lettres nécessitant POP/IMAP (vous pouvez dédier un des attributs extensionAttribute1 à 15 par exemple si vous avez étendu votre schéma avec l'extension Exchange).

Cette méthode est lourde, demande de maintenir des scripts, n'est pas immédiate (l'ajout au groupe requiert d'attendre le prochain lancement de la tâche à moins de la lancer manuellement).

Maintenant que vous connaissez la méthode traditionnelle, n'y-a-t-il pas un autre moyen d'autoriser ou bloquer l'accès à ces protocoles ? C'est ce que nous allons voir dans cette méthode « originale ».

La méthode « originale »

La théorie

La plupart de nos clients utilisent la fédération d'identité avec ADFS (Active Directory Federation Services) et synchronisent leurs comptes utilisateurs de leur Active Directory local (« on-prem ») avec Azure AD grâce à Azure AD Connect.

Lorsqu'on a configuré la fédération d'identité avec « Microsoft Office 365 Identity Platform », un certain nombre de claims a été ajouté. Microsoft nous les présente sur notre infrastructure ADFS lorsqu'on s'identifie sur la plateforme Office 365.

Un claim en particulier va nous intéresser : Client Application du type <http://schemas.microsoft.com/2012/01/requestcontext/claims/x-ms-client-application> tel que décrit sur cette page

[https://technet.microsoft.com/en-us/library/dn592182\(v=ws.11\).aspx#Claim%20Types](https://technet.microsoft.com/en-us/library/dn592182(v=ws.11).aspx#Claim%20Types)

Deux valeurs vont nous intéresser particulièrement :

- Microsoft.Exchange.Pop
- Microsoft.Exchange.Imap

Ce claim est utilisé pour toutes les authentifications basées sur le profil actif de la fédération où c'est Microsoft lui-même qui vient s'authentifier sur notre ADFS et non l'application lui-même, profil dit passif (ça concerne tous les services web-based comme SharePoint ou toutes les applications qui font de « l'authentification moderne » basée sur la bibliothèque ADAL comme Outlook 2016, Office mobile apps).

Attention : cette solution ne fonctionne donc qu'avec les comptes dit « fédérés » et non les comptes « in cloud » ou « managés » (synchronisation de mot de passe ou passthrough authentication). Pour ceux-là, s'ils exis-

tant, il faudra se rabattre sur la solution classique. A partir de là, il est très facile d'imaginer la suite :

- On vérifie si l'utilisateur n'est pas membre du groupe « O365-IMAP Allowed Users » (prenons le cas où on veut contrôler indépendamment l'accès au protocole POP et IMAP).
- On vérifie si l'application type utilisée est Microsoft.Exchange.Imap (ou Microsoft.Exchange.Pop)

Dans ce cas, on bloque l'accès sinon on autorise (utilisateur membre du groupe ou tout autre protocole d'accès).

La configuration

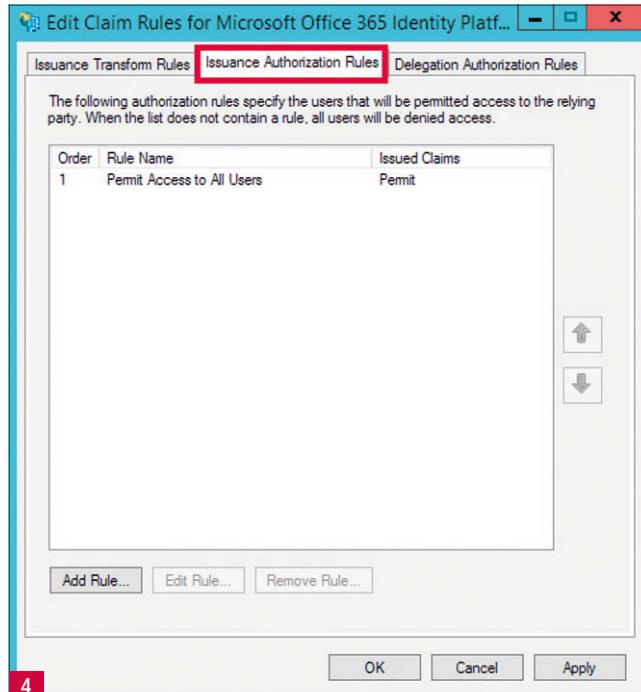
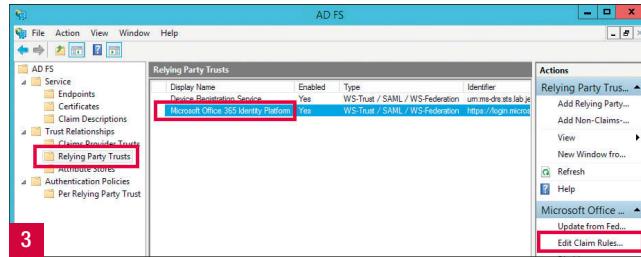
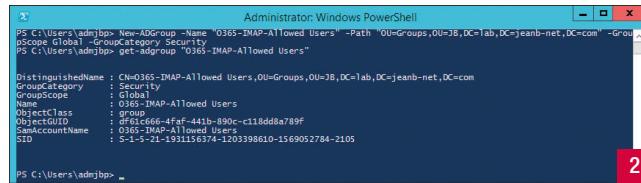
La configuration sera légèrement différente selon la version d'ADFS utilisée mais les concepts restent les mêmes. Les étapes décrites ci-dessous sont basées sur ADFS 3.0 présent dans Windows Server 2012 R2 :

- Créer un groupe Active Directory « O365-IMAP Allowed Users » :

```
New-ADGroup -Name "O365-IMAP-Allowed Users" -Path "OU=Groups,OU=JB,DC=jb,DC=lab" -GroupScope Global -GroupCategory Security
```

- Récupérer le SID du groupé précédemment créé : [2]

```
(Get-ADGroup -Name "O365-IMAP-Allowed Users").SID.Value
```



- Se connecter à la console d'administration d'ADFS et aller dans les Relying Party Trust [3]
- Sélectionner le Relying Party « Microsoft Office 365 Identity Platform » puis cliquer sur « Edit Claim Rules ».
- Rendez-vous dans l'onglet « Issuance Authorization Rules » et cliquer sur « Add » [4]
- Sélectionner « Send Claims Using a Custom Rule » et cliquer sur Next [5]
- On va d'abord vérifier que l'utilisateur ne fait pas partie du groupe sus mentionné :
 - Entrer comme nom de règle : Check O365-IMAP-Allowed Users group membership
 - Entrer comme règle : [6]

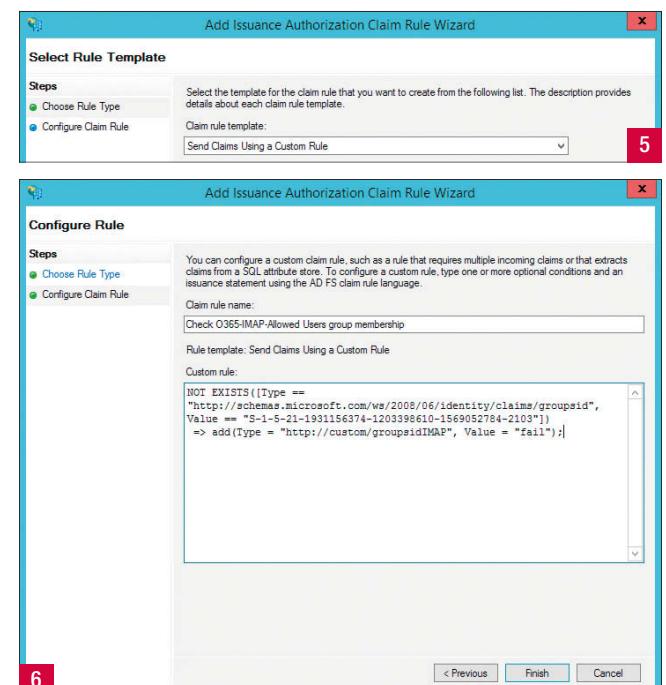
```
NOT EXISTS([Type == "http://schemas.microsoft.com/ws/2008/06/identity/claims/groupsid", Value == "VOTRE VALEUR DE SID DU GROUPE CREE"]) => add(Type = "http://custom/groupsidIMAP", Value = "fail");
```

- Cliquer sur Finish

- Refaire une nouvelle règle sur le même modèle (« Send Claims Using a Custom Rule ») avec cette fois ces informations :
 - Entrer comme nom de règle : Deny IMAP except for O365-IMAP-Allowed Users
 - Entrer comme règle : [7]

```
c1:[Type == "http://schemas.microsoft.com/2012/01/requestcontext/claims/x-ms-client-application", Value == "Microsoft.Exchange.Imap"]
&& c2:[Type == "http://custom/groupsidIMAP", Value == "fail"]
=> issue(Type = "http://schemas.microsoft.com/authorization/claims/deny", Value = "DenyUsersWithClaim");
```

- Descendre la règle par défaut « Permit Access to All Users » en dernier [8] Il est possible de faire la même chose pour le POP en utilisant un autre groupe :
- Recopier la règle « Check O365-IMAP-Allowed Users group membership » en un équivalent « Check O365-POP-Allowed Users group membership » en adaptant :
 - Le nom du groupe AD ;



- Le nom du claim ajouté en <http://custom/groupsidPOP> par exemple.
- Recopier la règle « Deny IMAP except for O365-IMAP-Allowed Users » en une règle « Deny IMAP except for O365-POP-Allowed Users » en adaptant dans les conditions :
 - Le nom du claim généré à l'étape précédente ;
 - Le nom de l'application en Microsoft.Exchange.Pop.

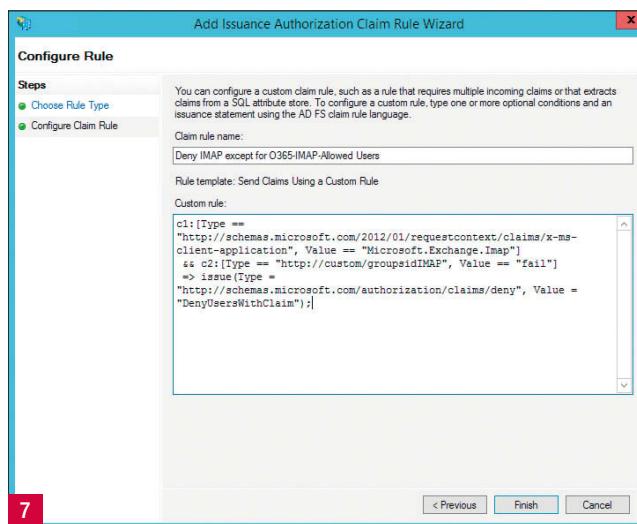
Il est aussi possible de mettre une condition unique pour autoriser les 2 protocoles en même temps en utilisant comme claim rule :

```
c1:[Type == "http://schemas.microsoft.com/2012/01/requestcontext/claims/x-ms-client-application", Value =~ "Microsoft\Exchange\.(Imap | Pop)"]
&& c2:[Type == "http://custom/groupsidIMAP", Value == "fail"]
=> issue(Type = "http://schemas.microsoft.com/authorization/claims/deny", Value = "DenyUsersWithClaim");
```

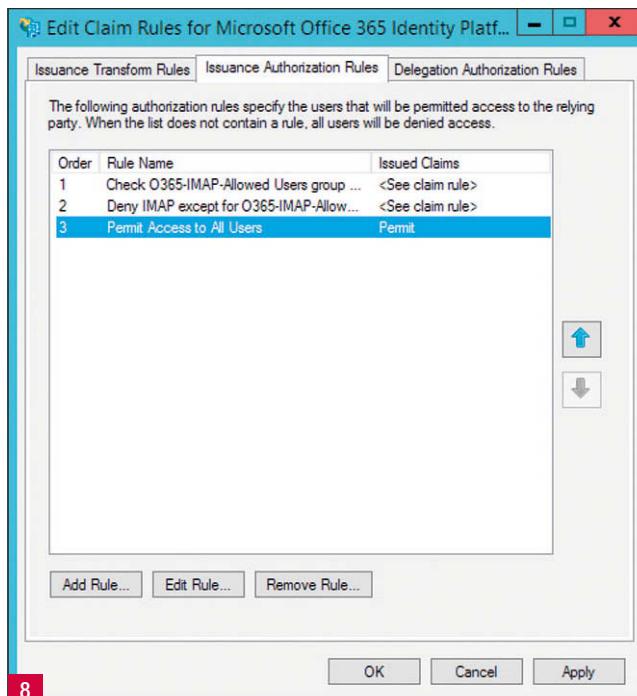
Vérification du fonctionnement

La première étape pour vérifier le fonctionnement va être d'ajouter l'utilisateur dans le groupe créé pour l'occasion.

Ensuite, il suffit de télécharger un client POP/IMAP (Thunderbird,



7



8

Mailbird, Windows Mail, un client smartphone qui fonctionne sur ces protocoles) et de configurer un accès :

- Serveurs POP/IMAP/SMTP : outlook.office365.com ;
- Nom d'utilisateur : l'UPN du compte Office 365 ;
- Mot de passe : le mot de passe AD ;
- Port IMAP : 993 (Chiffrement SSL/TLS) ;
- Port POP : 995 (Chiffrement SSL/TLS) ;
- Port SMTP : 587 (Chiffrement STARTTLS) .

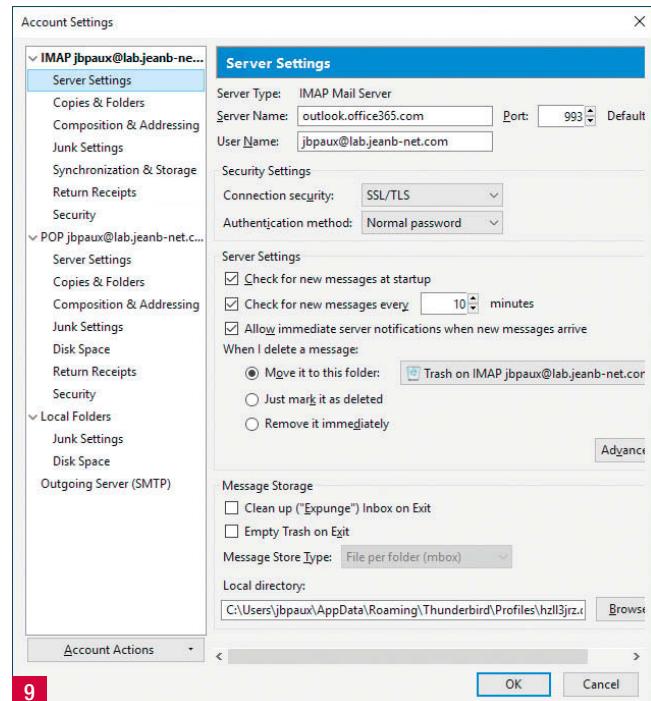
Par exemple avec Thunderbird : [9]

Lorsque l'utilisateur est membre du groupe, tout va bien, on peut se connecter. Le supprimer du groupe entraîne directement une erreur (avec Thunderbird) : [10]

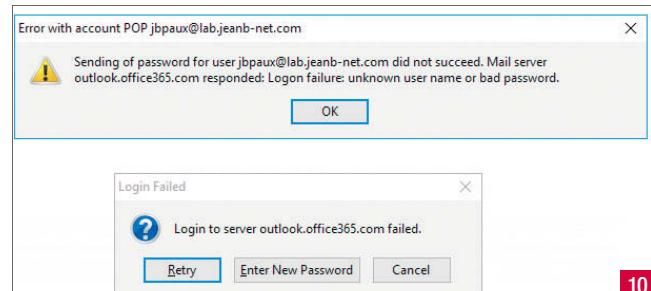
Et voilà ! Vous avez configuré l'accès à Exchange Online en POP/IMAP sans même avoir touché Exchange Online et vos boîtes aux lettres. Vous gérez tout ça directement depuis votre AD local. A noter que vous pouvez de la même manière le faire pour le protocole ActiveSync ou complexifier encore le setup pour gérer les adresses IP autorisées.

Troubleshooting

ADFS va nous permettre de voir les requêtes arriver sur le serveur. En cas de ferme ADFS avec plusieurs serveurs, il va falloir cependant configurer sur chaque serveur de la ferme l'audit et regarder chaque event log. Une solution de centralisation de log est sûrement la première étape pour faciliter le troubleshooting. Une solution d'analyse de log type Splunk ou « Log Analytics » dans Microsoft Operations Management



9



10

Un support client optimisé en connectant la VoIP à votre Back Office Magento

Partie 3

• Jean-Marie Heitz,
Ingénieur études et développement
e-commerce chez
Netapsys Grand Est
Jean-marie.heitz@netapsys.fr

POSSIBILITÉS TRADITIONNELLES DE CTI AU NIVEAU DES UA

FONCTIONNALITÉS INTEGRIÉES A SIP [1]

Nous allons commencer tout d'abord par des fonctionnalités d'information qui sont présentes directement dans le protocole SIP.

Pour commencer, il s'agit de voir comment établir un appel avec SIP. L'UA qui souhaite établir un appel va envoyer au proxy SIP, ou éventuellement même directement à l'autre UA, s'il la connaît, une requête INVITE, qui permet de commencer à négocier l'appel.

Cette requête fait partie d'un ensemble de requêtes et de réponses (réponses éventuellement provisoires en codes 1XX, puis une réponse définitive) que l'on appelle transaction, tandis que cette première transaction démarre également un dialogue SIP, identifié par un call ID qui va être transféré de bout en bout, et qui va englober l'ensemble des transactions liées à l'appel.

La syntaxe de SIP ressemble assez fortement à ce qui se fait en HTTP ou SMTP. On retrouve une première ligne qui indique la requête en elle-même, à la manière de HTTP, puis on retrouve des headers, dont deux headers qui vont être essentiels pour du CTI : le header From, qui indique l'appelant, et le header To, qui indique l'appelé. Ces deux headers permettent de véhiculer à la fois un nom, ainsi qu'une adresse SIP, structurés de la même manière que les From et To des corps des messages SMTP.

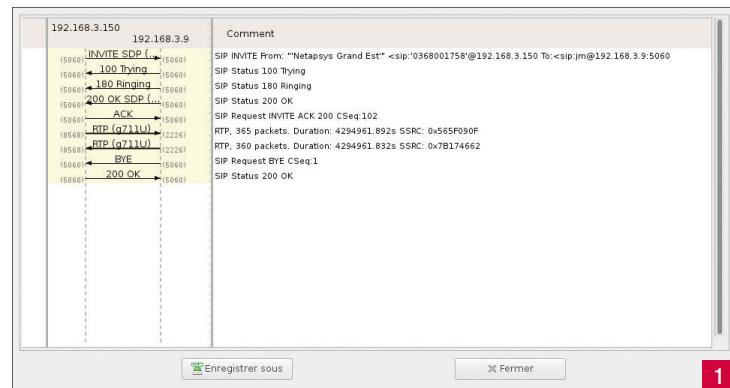
Au niveau d'une passerelle de téléphonie, pour un appel téléphonique entrant on peut donc forger des messages From utilisant la présentation du nom, et éventuellement aussi la présentation du numéro de l'appelant dans une adresse SIP.

La réponse au message INVITE, qui contient une description de session SDP indiquant les paramètres de communication média souhaités, pourra, par exemple, être une réponse OK, qui contiendra les paramètres de communication média de l'autre entité, ce qui constitue la négociation de média entre les deux entités. S'ensuit alors un message ACK qui confirme la bonne réception de la réponse.

En fin d'appel, une des deux entités pourra envoyer une requête BYE, avec sa réponse OK.

```
INVITE sip:jm@192.168.3.9:5060 SIP/2.0
Via: SIP/2.0/UDP
192.168.3.150:5060;branch=z9hG4bK5345de41
Max-Forwards: 70
From: "Netapsys Grand Est" <sip:0368001758@192.168.3.150>;tag=as6632cd81
To: <sip:jm@192.168.3.9:5060>
Contact: <sip:0368001758@192.168.3.150:5060>
Call-ID:2a78be2f5ec2f52835f6ab5045e6872e@192.168.3.150:5060
```

Nous terminons notre série d'articles sur Magento et la VoIP. Bonne découverte.



CSeq: 102 INVITE
User-Agent: Asterisk PBX 13.10.0
Date: Tue, 02 Aug 2016 19:14:13 GMT
Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, REFER,
SUBSCRIBE, NOTIFY, INFO, PUBLISH, MESSAGE
Supported: replaces, timer
P-Asserted-Identity: "Netapsys Grand Est"
<sip:0368001758@192.168.3.150>
Content-Type: application/sdp
Content-Length: 286

v=0
o=root 974966669 974966669 IN IP4 192.168.3.150
s=Asterisk PBX 13.10.0
c=IN IP4 192.168.3.150
t=0 0
m=audio 8568 RTP/AVP 0 8 3 101
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:3 GSM/8000
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-16
a=maxptime:150
a=sendrecv

Exemple de requête SIP INVITE, avec les champs From et To

De la sorte, on constate qu'un UA peut disposer dans les appels des informations relatives à l'identité de l'appelant et de l'appelé, ce qui constitue une des bases du CTI. Cependant, il faut également pouvoir observer un UA pour pouvoir réagir en déclenchant les fonctionnalités applicatives. Les fonctionnalités d'observation sont obtenues en conjuguant les fonctionnalités de plusieurs RFC simultanément :

- La RFC 3265, qui est maintenant remplacée par la RFC 6665, définit un jeu de deux requêtes : la requête SUBSCRIBE, qui permet de souscrire à des événements issus d'un autre UA, et la requête NOTIFY, qui

permet justement d'envoyer les notifications au souscripteur. Cependant, cette RFC décrit uniquement un mécanisme générique de souscription et de notification, les charges utiles étant laissées à la charge de la spécification de packages d'événement.

- Pour le CTI, il y a justement un package très utile, qui est le package "dialog" décrit dans la RFC 4235. Ce package inclut notamment les informations sur l'appelant et l'appelé, ce qui fait qu'on peut montrer (si on dispose des autorisations nécessaires) l'activité d'un autre UA en matière d'appels émis et reçus ; le package permettant de suivre grâce à un ensemble d'états le déroulement d'un dialogue SIP.
- Il y a encore une autre RFC complémentaire qui peut être utile pour compléter le mécanisme SUBSCRIBE/NOTIFY : c'est la RFC 3903, qui définit la requête PUBLISH. Cette requête permet de publier un état vers un autre UA.

```
<?xml version="1.0"?>
<dialog-info xmlns="urn:ietf:params:xml:ns:dialog-info" version="1" state="partial"
entity="sip:jm7@10.0.0.2">
<dialog id="idcdeb6026" call-id="ce532d45-1c339164-77585c77@10.0.0.6"
local-tag="624FF151-C7F6C400" remote-tag="as6cff82d1" direction="initiator">
<state>confirmed</state>
<local>
<identity display="jm7">sip:jm7@10.0.0.2</identity>
<target uri="sip:jm7@10.0.0.6:5060">
<param pname="+sip.rendering" pval="yes" />
</target>
</local>
<remote>
<identity>sip:3@10.0.0.2</identity>
<target uri="sip:3@10.0.0.2">
</target>
</remote>
</dialog>
</dialog-info>
```

Exemple d'événement dialog

On constate donc qu'on a tout le nécessaire pour obtenir des informations sur un appel en cours. Cependant, on pourrait également souhaiter déclencher un appel sur un autre poste. Ceci est possible grâce à la méthode REFER, définie dans la RFC 3515 : dans un contexte hors dialogue, on envoie la requête au futur appelant, et le champ Refer-To contient la cible à appeler.

UA-CSTA

On a donc remarqué que le protocole SIP avait déjà un gros nécessaire pour à la fois observer les appels d'un UA et pour émettre des appels. Cependant, ce n'est pas la seule solution protocolaire qui est possible. Par exemple, l'ECMA a défini une utilisation de SIP qui va permettre d'établir des sessions SIP à des fins de monitoring ou de commande : défini dans le Technical Report 87, il s'agit de UA-CSTA, User Agent Computer Supported Telecommunication Applications. En pratique, on crée des sessions SIP qui n'auront pas un contenu SDP, mais application/csta+xml. Le contenu va pouvoir lancer par exemple une commande initiale, puis on utilisera des requêtes SIP INFO pour les prochaines requêtes. L'appel sera terminé comme d'habitude par une requête BYE. De manière symétrique, on pourra déclencher également du monitoring depuis l'INVITE initiale, et ce sera alors l'UA monitoré qui va déclencher des messages

INFO à destination de l'application de management, jusqu'à ce qu'un BYE vienne arrêter la conversation.

Les messages CSTA en format XML sont définis dans le standard ECMA 323, tandis que la charge utile proprement dite est définie dans le standard ECMA 269. Il faut ensuite voir au niveau des UA quelles sont les commandes et les événements supportés. On peut par exemple regarder la commande MakeCall pour émettre des appels, ou MonitoringStart et MonitoringStop pour le monitoring.

D'autres API ont bien évidemment été développées, comme par exemple TAPI, JTAPI, ou TSAPI.

LES API DU TELEPHONE : Notifications URI, Push URI, et Browser

La plupart des smartphones disposent d'une stack HTTP qui permet de configurer manuellement le poste. Cette stack HTTP comporte fréquemment une URI qui permet de déclencher une numérotation d'appel. De manière réciproque, la plupart des smartphones permettent également d'envoyer des notifications sur une ou plusieurs URL lorsque des événements surviennent (appel entrant, appel sortant, combiné décroché, combiné raccroché...). Cependant ces fonctionnalités sont des fonctionnalités propriétaires et donc non standardisées. Si l'on souhaite les utiliser, il est donc nécessaire de se plonger attentivement dans la documentation fournie par les constructeurs. Il convient toutefois de faire attention avec ce genre de fonctionnalités. En effet, la réponse d'un appel HTTP peut parfois prendre du temps à venir, et il faut considérer comment se comporte le téléphone : va-t-il attendre la réponse avant de continuer dans son exécution ? Il faudra donc vérifier ce genre de comportement, définir si nécessaire des stratégies asynchrones pour fournir une réponse appropriée au téléphone, et définir des temporisations pour éviter de bloquer l'utilisation du téléphone. On notera que la possibilité de pousser du contenu vers les téléphones ne se limite parfois pas à du texte, mais permet également de manipuler un minimum le téléphone en lui faisant exécuter une charge utile spécifique.

On en arrive à ce moment à l'intégration non pas du téléphone dans l'application, mais de l'application dans le téléphone. En effet, certains téléphones fournissant des écrans "graphiques" (même couleur pour certains modèles), avec des résolutions non négligeables, prévoient carrément des langages permettant d'utiliser ces capacités de présentation. On peut par exemple trouver des charges XML ou des charges HTML qui seront interprétées par les téléphones, charges qui intègrent les spécificités des téléphones. On trouve dans ces API des possibilités pour augmenter l'ergonomie de ces applications : la possibilité de définir des actions particulières pour les softkeys, c'est à dire, les touches contextuelles qui s'affichent directement sur l'écran.

Dans le cadre de Magento, on peut mettre à profit les possibilités de theming qui existent dans Magento pour créer un thème spécifique pour les téléphones, ainsi que des contrôleurs qui chargeront ce thème, de sorte à créer un microsite spécialisé pour ces périphériques.

DEVELOPPEMENT DE PLUGINS POUR LES SOFTPHONES

Exemple d'interface entre Magento et Jitsi (réalisé via des Web services) [2]

D'autres types de téléphones existent également, comme par exemple les softphones. Certains softphones se prêtent particulièrement bien au développement de plugins, comme par exemple Jitsi, qui utilise en interne une infrastructure OSGI qui est divisée en bundles. En fonction des

solutions utilisées, on peut carrément développer des plugins pour les téléphones, de manière à lancer des URL forgées pour interagir avec l'application. A l'inverse, pour les appels sortants, on peut éventuellement chercher à intégrer au niveau système la gestion de certains protocoles tels que le protocole SIP en utilisant des URI forgées au niveau applicatif. Cependant, la question de l'authentification se pose dans ce cas, car comme on est dans le cadre d'appels, il faut pouvoir s'identifier rapidement pour accéder aux informations. Dans ce contexte, une ouverture de navigateur prend déjà du temps, et une authentification par mot de passe prendrait trop de temps. Au niveau de Magento, comme il s'agira d'accéder au BO, il serait bon de mettre une méthode d'authentification rapide en place comme une authentification par clef (si on dispose d'une méthode rapide pour déverrouiller le stockage des clefs, ou si l'on peut utiliser les certificats personnels stockés dans l'ordinateur) ou l'authentification intégrée Windows, si on utilise cette plateforme et que l'on proxifie avec un IIS le système.

Toutefois, au lieu de chercher à instancier le navigateur pour afficher les informations, une autre solution serait de passer par les Web services fournis par Magento. Un tel accès suppose de bien identifier les personnes qui vont en avoir besoin, et de pouvoir si nécessaire gérer de manière suffisamment fine les autorisations. Néanmoins, un tel fonctionnement permettrait de rester dans le softphone, ce qui veut dire qu'il n'y aurait pas d'autre application à lancer, et que l'on peut déverrouiller par exemple au démarrage un keystore en cas d'authentification par clef.

En ce qui concerne l'utilisation d'une application, on peut également chercher à centraliser l'utilisation de cette application par les autres logiciels pour essayer de profiter au mieux du couplage téléphonique réalisé dans le softphone :

- On peut tout d'abord, au niveau du navigateur, utiliser l'application pour gérer plus de protocoles (comme par exemple les telURI).
- On peut également faire de même au niveau du système, si le système le permet. C'est par exemple le cas dans Windows, où l'on peut définir dans des packages d'installation des clefs dans le registre de manière correspondante.

SPÉCIFICITÉS DU CONTEXTE WEB ROUTAGE D'INFORMATIONS : MIDDLEWARE DE MESSAGING

Jusqu'à présent, on a parcouru des fonctionnalités qui étaient surtout fournies nativement par des hardphones. On a vu que l'on peut récupérer des informations des téléphones et lancer des appels. Néanmoins, il faut tenir compte d'une pratique très répandue qui est de mettre l'ensemble des téléphones physiques dans un réseau à part (VLAN au minimum, si ce n'est carrément un réseau physique séparé, en raison de la nécessité d'une alimentation POE des téléphones pour assurer la continuité de service même si l'on a une coupure de courant). Il ne faut pas oublier qu'il faut absolument réussir à passer des appels, puisque l'on pourrait aussi devoir émettre des appels d'urgences). De ce fait, il faudra quand même mettre en place une architecture qui permettra non seulement de récupérer les informations des UA, mais également de les acheminer jusqu'aux postes informatiques des utilisateurs. Il y a donc une problématique de routage des messages sous-jacente. Cette problématique peut être résolue par exemple en utilisant un framework de files de messages (message queuing), c'est à dire en utilisant des services comme RabbitMQ, dont l'utilisation est justement prévue dans la version EE de Magento2. Pour l'observation des appels émis et reçus, il s'agira donc de créer un système avec :

- Un producteur côté téléphonie, qui va récupérer les messages ;
- Un système de routage des informations ;
- Des consommateurs applicatifs, qui s'interfaceront par exemple avec Magento.

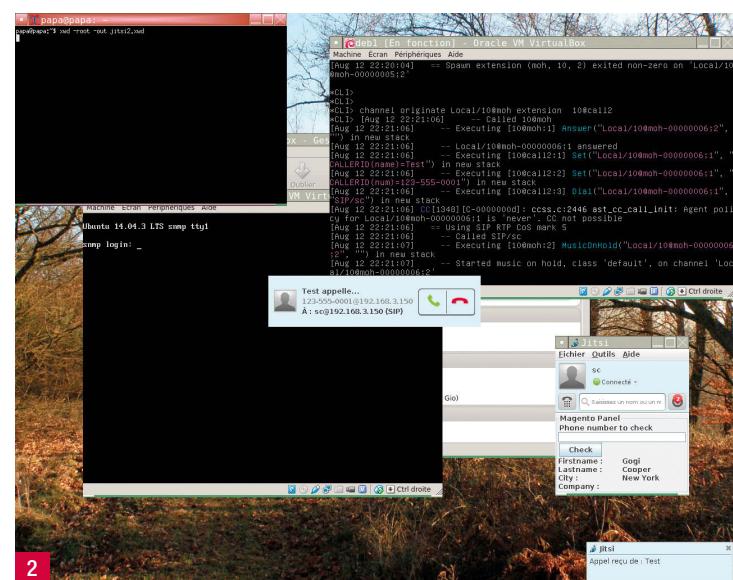
En pratique, il faudra plusieurs ouvertures de files sur le serveur de files de messages pour réussir à faire fonctionner le système. Par exemple :

- Une file de commandes, qui permettra d'envoyer des commandes aux systèmes d'observation (démarrer une observation d'une ressource par exemple) ;
- Des files créées uniquement par les consommateurs, qui auront des topics qui seront les URI des ressources (ou au minimum les identifiants qui se trouvent avant le domaine). De la sorte, un échange de type topic fera l'affaire pour dispatcher les événements sur tous les clients désireux d'obtenir des infos sur une ressource donnée. Cette file pour chaque client servira aussi pour les réponses directes aux commandes. Dans le système, la clef de routage la plus naturelle, hors les clefs spéciales pour la liaison directe vers chaque observateur et la file de commande, sera la Request URI de la ressource, puisque cette dernière est déjà utilisée dans les requêtes SUBSCRIBE. C'est donc de ce fait un excellent candidat à cet effet.

COUCHE CLIENT WEB : LES WEB-WORKERS

L'avantage d'utiliser un système de files de messages comme RabbitMQ est qu'il existe un grand nombre de clients pour s'interfacer avec ce système, dont des clients Javascript pour le Web. A cela s'ajoute la possibilité d'avoir des connexions plus "pérennes" que les connexions HTTP standard : les websockets, par exemple. Par contre, il faut alors que cette pérennité s'applique aussi au niveau de l'application Web, ce qui n'est pas évident puisque la navigation résulte du fait d'aller de page en page. De ce fait, deux types d'approches sont envisageables :

- Abuser de l'AJAX : les appels AJAX permettent de récupérer le contenu d'autres pages tout en restant sur une même page. Mais ce mode de fonctionnement n'est guère possible si toute l'application n'a pas été prévue pour fonctionner de la sorte initialement.
- Utiliser des web workers : les webworkers sont une nouveauté de HTML5. Ils permettent dans une certaine mesure de faire des travaux en parallèle, et peuvent communiquer avec les pages à l'aide de messages. Deux types de workers existent : les workers dédiés et les workers partagés. Ce sont ces derniers qui nous intéresseront le plus, car si on intègre les mécanismes de communication dans un worker



partagé, ce worker subsistera jusqu'à ce que plus aucune page du site ne soit chargée. A noter que dans le cadre de la navigation avec un seul onglet, les tests que j'ai réalisés montrent que le fait d'aller sur une autre page du même site ne remet pas le compteur de référence à 0. En revanche, si on force le rechargement de la page dans ces conditions, là le compteur de référence sera réinitialisé, et un nouveau Web worker sera créé.

On comprend donc que la deuxième solution, utilisant les Web workers partagés, semble plus plausible, puisque Magento a intégré dans son fonctionnement standard de la navigation à travers plusieurs pages (par exemple, une page d'édition, associée à une page de post qui va renvoyer sur la page d'édition par exemple, agrémentée d'un message de session) ; cela nécessitera néanmoins une division de l'application Web en au moins deux couches :

- La couche du worker
- La couche autour du worker (IHM d'interaction avec le worker).

De plus, le développement de l'application risque d'être plus difficile, car il est a priori moins aisés de débugger avec des workers. En pratique, des navigateurs comme Google Chrome fournissent des outils permettant d'inspecter les workers partagés.

En revanche, il subsiste des limitations à l'intérieur des workers. En effet, il n'est pour l'instant, par exemple, pas encore possible d'utiliser WebRTC dans un worker partagé. Ceci notamment car il faut associer un élément de DOM à un flux de média, par exemple, pour restituer les flux distants à l'utilisateur local.

TEL URI et ENUM

tel:+33368001758

Exemple de Tel URI

A l'inverse, on peut aussi créer des petites applications pour réorienter et transformer les URL vers le navigateur Web, en ayant pour intention de déployer le minimum d'intelligence sur les postes, de sorte à faciliter la maintenabilité du système en réduisant les mises à jour requises.

Dans cette optique, on peut mettre à profit plusieurs notions pour utiliser un algorithme "stable" se basant sur des données centralisées :

- Tout d'abord, il faut se rappeler que les numéros de téléphones publics sont standardisés dans la norme E.164 ;
- Ensuite, on peut remarquer que la RFC 3966 définit le schéma de protocole "tel", dans lequel on peut en particulier retranscrire les numéros publics E.164. Autrement dit, il est possible d'inclure dans une forme standardisée des numéros de téléphone dans des documents, via des liens ;
- Fort de cette encapsulation du numéro de téléphone E.164 dans une tel URI, on comprend qu'on peut chercher à supporter le schéma de protocole tel au niveau des machines ;
- On peut également à nouveau décapsuler le numéro de téléphone E.164 des tel URI publics, pour retrouver le numéro E.164 d'origine, et ensuite utiliser les services ENUM, qui s'appuient sur un algorithme strictement défini et des enregistrements DNS pour dériver une URI depuis un numéro E.164. Cette URI peut par exemple être une URI SIP, mais pourrait très bien aussi être une URI HTTP, ce qui correspond assez bien aux deux exploitations que l'on pourrait souhaiter en faire ;
- Le problème avec ENUM, c'est qu'en théorie, on devrait disposer d'un registre public en France nommé 3.3.e164.arpa. Or, il y a eu des essais

en 2001-2002, mais je n'ai pas vu d'informations sur la commercialisation de services ENUM en France. Du fait de sa non-popularité, il est nécessaire d'utiliser un registre ENUM privé pour résoudre les adresses et fournir ainsi un service à ses utilisateurs. Néanmoins ce problème est plus un problème commercial que technique. D'autre part, il faut bien voir que le schéma d'URI tel permet également d'encapsuler des numéros privés, dont on peut éventuellement aussi souhaiter assurer une translation privée. Dans ce cas, un usage privé de ENUM pourrait être intéressant (avec des enregistrements DNS non publics). Si l'on souhaite personnaliser les numéros avec des URI qui n'ont qu'une signification intra-entreprise, ce n'est que cette approche qui est de toute façon utilisable.

Il est intéressant donc de reconsidérer d'éventuelles applications pratiques de Tel URI et de ENUM :

- A travers son numéro de téléphone, fournir aux utilisateurs extérieurs un ou plusieurs autres services pour joindre son interlocuteur dans une entreprise ; ceci en incorporant des liens avec le protocole tel dans des documents avec des hyperliens et en fournissant des enregistrements publics ;
- A travers un service ENUM privé, on arrive à faire du CTI qui peut fournir une manière souple pour récupérer des informations sur un numéro, ou bien même de l'appeler (par exemple si on utilise une URL SIP pour un softphone, ou une URL HTTP pour un webphone hébergé dans une page web) ;

On constate qu'au niveau de Magento, on pourra vraisemblablement non seulement inclure les tel URI dans les pages Web, et les mails qui sont produits par le système, mais également dans des PDF, puisque le Zend Framework utilisé dans Magento contient la possibilité d'inclure des liens dans les PDF (au minimum dans la version livrée avec Magento 1.9.2.4).

LES WEBPHONES

Exemple d'intégration de webphone : la stack SIP en JS en elle-même est dans l'onglet à droite de l'onglet actif, tandis que les contrôles sont déportés dans la page de l'onglet actif. Un appel est en cours (l'utilisation du micro est matérialisée par le point rouge). [3]

Les webphones sont des softphones intégrés dans les pages Web. Concrètement, ils rassemblent plusieurs technologies : dont une technologie de transport de la signalisation, qui est bien souvent les websockets. Cette couche est le plus souvent pilotée par une stack de signalisation, écrite en JS, comme par exemple une stack SIP. Si cette stack ne traite que de la signalisation, pour être utile, elle interagit cependant avec le WebRTC (Web Real Time Communication), qui est un ensemble technique permettant de gérer notamment :

- La capture et la restitution des flux audio ou vidéo ;
- La gestion des endpoints media et la problématique NAT associée ;
- La préparation de description et de réponse d'offres en JSEP (Javascript Session Establishment Protocol, protocole encore en version draft, qui encapsule en gros du SDP).

L'usage d'un webphone ainsi constitué nécessite quelques prérequis techniques :

- Disposer d'un IPBX capable de s'interfacer avec du transport en Web socket (et en plus, websocket au-dessus de SSL) - ou au minimum d'un proxy SIP qui pourra se mettre dans la route pour assurer la communication ;
- Disposer des mécanismes nécessaires pour assurer la résolution des

problématiques NAT dans le cadre de l'utilisation de ICE (Interactive Connectivity Establishment) : serveur STUN (Simple Traversal of UDP through NAT - souvent utilisé avec deux serveurs), serveur TURN (Traversal Using Relay NAT).

A l'usage, on constate que c'est le plus souvent l'API de signalisation qui va demander la configuration à utiliser, et que l'API va s'interfacer d'elle-même avec le WebRTC, de sorte que le développeur a majoritairement à manipuler l'API de signalisation.

Avec la spécificité de Magento qui est que l'on navigue sur différentes pages, il faut faire en sorte de pouvoir conserver la continuité des appels en cours. De là, on en déduit qu'il pourrait être plus intéressant d'intégrer la véritable partie moteur webphone dans une page "maître" spécifique qui restera tout le temps active, tandis que la partie d'interfaçage graphique devrait se situer dans des blocs et du code JS intégré aux pages Magento. La liaison entre les deux étant assurée par un shared Web worker pour assurer la transmission des informations entre la page maître et les autres pages. Ceci signifie également qu'il est à ce moment nécessaire de créer un protocole de communication entre le Web worker et les autres pages. C'est donc un travail non négligeable qui peut se présenter. L'utilisation du webphone a surtout l'avantage de faciliter à long terme la maintenabilité (puisque le code est centralisé et nécessite un seul déploiement central, et pas sur tous les postes) et l'utilisation "tout terrain" de la solution ; les parties liées au matériel et les problématiques NAT étant laissées au navigateur. De plus, en étant déjà dans l'interface Web de l'utilisateur, on est aussi au bon endroit pour présenter des fonctionnalités CTI sur les appels. A l'inverse, on est à "l'endroit le plus éloigné" pour interroger les services : c'est à dire qu'il faudra éventuellement mettre en place un nécessaire pour récupérer au minimum un stock d'informations de base sur l'appel, qui permettront à l'utilisateur de cerner rapidement le

The screenshot shows the Magento Admin Panel. In the top navigation bar, 'Dashboard' is selected. Below it, the 'Webphone' section is active, showing a form to register a phone number. The 'Dashboard' section displays lifetime sales of 39 763,08 € and average orders of 1 988,15 €. The 'Orders' tab is selected in the dashboard summary. A red '3' is visible in the bottom right corner of the dashboard area.

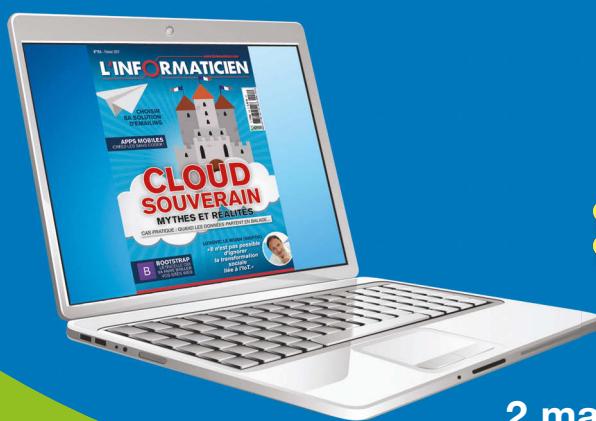
contexte métier de l'appel (par exemple accès à la balance des paiements du client, aux anciennes commandes passées,...), quitte à fournir des liens vers d'autres ressources dans l'interface pour obtenir des informations détaillées. La fourniture de ces informations pourra a priori se faire par exemple via des appels XHR ou un système de files de messages.

CONCLUSION

En conclusion, on a vu au travers de ce dossier que le CTI est possible à la base, car les protocoles de téléphonie intègrent les informations nécessaires au couplage. On a constaté que ces dernières peuvent être exploitées de manière centralisée (Asterisk) ou non (les UAC SIP), voire même intégrées dans les interfaces Web de diverses manières (dont webphone). Le CTI ne se limite cependant pas à Magento : les téléphones Android sont un autre exemple connu. •

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

Les bénéfices de la **Preuve d'Autorité** Proof-of-Authority (PoA) dans les blockchains



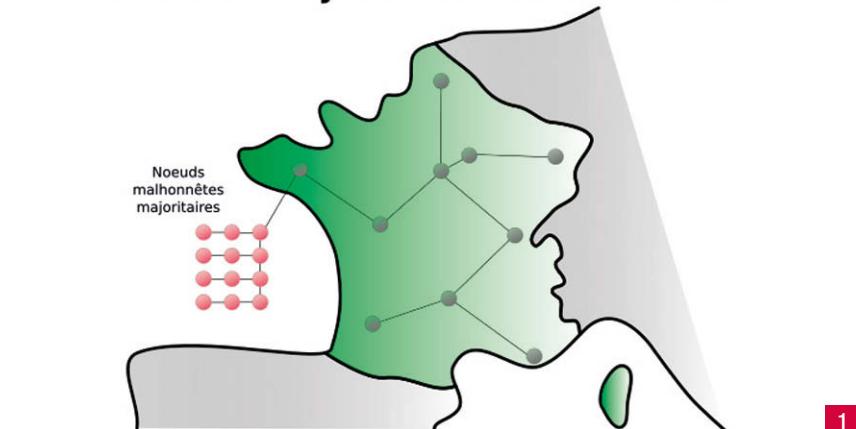
Romain
Gaborieau
Concepteur
développeur
SQLI Nantes
 SQLI
Connected Business Stories

La blockchain commence à s'immiscer dans nos systèmes informatiques en offrant une nouvelle manière de penser nos échanges avec les autres. On la connaît aujourd'hui principalement grâce au Bitcoin qui est composé de milliers d'acteurs. Lorsque l'on souhaite créer sa propre blockchain sur le réseau de son entreprise dans des proportions moindres - quelques noeuds - il peut être intéressant de regarder plus en détails les mécanismes de consensus. Parmi ces derniers, nous pouvons citer la Preuve de travail (ou Proof-of-work, la plus commune), la Preuve d'enjeux (ou Proof-of-stake), ou encore la Preuve d'Autorité (ou Proof-of-authority), aux propriétés bien distinctes. La Preuve d'autorité nous semble la plus adaptée à un contexte de blockchain privée et pourtant la moins documentée. Nous allons y remédier.

Retour sur la **Preuve de Travail**

La preuve de travail est le mécanisme utilisé pour faire fonctionner la majorité des blockchains publiques, dont notamment Bitcoin. Le principe est de permettre à l'ensemble des noeuds, appelés mineurs, d'être actifs sur la blockchain et de prouver à l'ensemble du réseau sa légitimité à écrire sur la blockchain (création et ajout d'un bloc). Techniquement, pour pouvoir créer et ajouter un nouveau bloc, un mineur doit trouver la solution d'un problème mathématique très coûteux en ressources. Une fois cette solution trouvée, il peut diffuser à l'ensemble du réseau le bloc contenant (entre autres) la solution. Chaque noeud qui réceptionne ce bloc peut vérifier instantanément que la solution est correcte, puis l'ajouter ou non à sa copie locale. Ce processus assure une confiance envers les noeuds qui publient le nouveau bloc. En effet, pour qu'un bloc soit valide, il est nécessaire que la majorité des mineurs l'ajoutent à leur copie locale. Pour qu'un acteur malhonnête puisse agir sur la blockchain, il doit alors posséder cette majorité, ce qui est à ce jour impossible au vu des besoins de performance demandés, car il faudrait posséder un grand nombre de machines, ce qui implique un coût très important. [1] Mais cela n'est valable que sur des blockchains d'au moins plusieurs milliers de noeuds. Sur des chaînes privées de quelques dizaines d'acteurs, il est plus facile d'obtenir la puissance équivalente à la majorité des noeuds (un bon serveur a autant de puissance de calcul qu'une ou plusieurs dizaines d'ordinateurs de bureau). La preuve de travail ne permet donc pas d'apporter la confiance nécessaire dans les blockchains constituées de peu de noeuds.

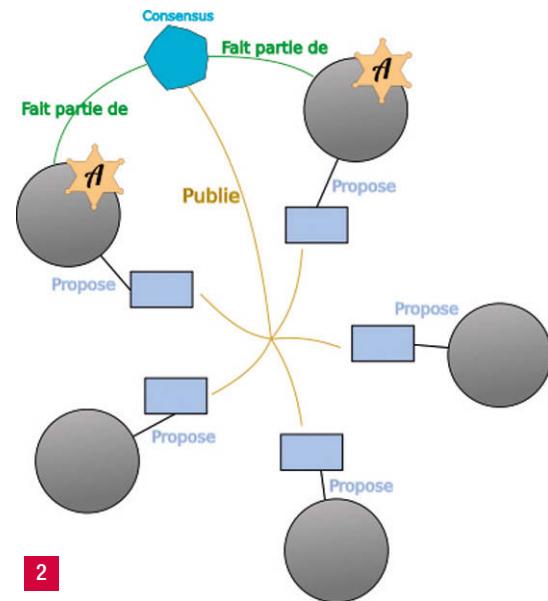
Blockchain attaquée par monopole de la majorité des noeuds



Attaque d'une blockchain par monopole majoritaire

La **Preuve d'Autorité** entre en jeu

La *Preuve de Travail* a montré ses limites et c'est ici que la *Preuve d'Autorité* (**POA**) répond aux besoins. Dans une chaîne limitée en nombre d'acteurs, on sait que quoi qu'il arrive, il serait possible de devenir le possesseur majoritaire de la horde de mineurs, car n'importe qui peut devenir mineur. Avec la **POA**, on définit à l'avance quels sont les noeuds qui feront *autorité*, c'est-à-dire qui joueront le rôle de mineurs. On perd alors le principe de confiance, ce qui nous oblige à affecter les autorités à des parties avec un intérêt commun minimal. [2] Prenons l'exemple d'une enseigne d'une vingtaine de magasins et dont chaque employé utilise un ordinateur possédant un noeud de la blockchain. La stratégie globale de l'enseigne est promulguée et partagée en utilisant la



Fonctionnement de la blockchain

blockchain. Imaginons qu'il y ait, en tout, une centaine de noeuds, alors la **POA** est bien indiquée dans ce cas. En effet, avec la preuve de travail, il serait possible pour un employé de configurer et déployer une centaine de noeuds supplémentaires afin de "hacker" la blockchain. Avec la **POA**, si nous décidons que dans chaque magasin, l'ordinateur de chaque directeur est le noeud qui a autorité, on peut avoir confiance. En effet, aucun de ces ordinateurs ne sont accessibles par d'autres personnes que leur propriétaire. De plus, chaque directeur étant en compétition avec les autres pour le poste de PDG, il est peu probable que la moitié d'entre eux s'allient pour modifier la blockchain.

L'autorité sous toutes ses formes

On peut appliquer ce principe dans de nombreux cas : les autorités séparées entre la direction, les syndicats, le CE, les délégués des employés ou entre des professeurs, des administratifs et des bureaux d'élèves... Mais le second point intéressant de la *Preuve d'Autorité* est la gestion en écriture de sa chaîne. En effet, selon les méthodes de **POA** utilisées, il est possible d'empêcher les noeuds non-autorités de faire des demandes d'inscription de données sur la blockchain (d'émettre des transactions). Cela permet donc de garder une chaîne transparente, car ses données sont toujours lisibles, mais un tiers extérieur au contexte ne pourra pas agir sur la blockchain.

Imaginons une application qui gère les budgets des différents départements d'une société. Le pôle administratif peut réaliser des versements vers les établissements secondaires et inversement. Mais les employés, qui ont accès à la blockchain, afin de pouvoir garder un œil sur les transactions et vérifier par exemple qu'il n'y a pas de fraudes, ne peuvent quant à eux pas interagir avec cette blockchain.

CONCLUSION

La **POA** est un concept qui est très intéressant pour les blockchains privées de petite envergure. Elle permet d'éviter le monopole d'un acteur malhonnête et de gérer les accès en lecture. Malgré une baisse de confiance, on obtient une blockchain plus rapide, demandant moins de ressources de calcul et ainsi une baisse de la consommation énergétique. Elle vient juste d'apparaître dans des clients Ethereum tels que Parity et donc balbutie encore. Mais elle offre des perspectives intéressantes que nous suivrons de près.

Les smart-contracts Ethereum sur Android avec Ethereum-android



Guillaume
Nicolas
Ingénieur
développeur
SQLI Nantes



Dans un article précédent, nous avons appris à développer une application Android décentralisée contenant un noeud Geth complet. Ce savoir nous a permis de lire et d'écrire sur la blockchain. Cependant, la particularité d'Ethereum est que la blockchain peut stocker bien plus que des transactions financières : du code exécutable. Ce code est appelé smart-contract.

Mais quelle est cette sorcellerie ?

Un *smart-contract* est un ensemble structuré d'instructions exécutées selon certaines conditions. Cette notion prend tout son sens dans l'environnement blockchain. En effet, grâce à l'immuabilité de la blockchain et au protocole Ethereum, il devient possible d'enregistrer de façon sécurisée et sûre un contrat entre plusieurs parties. Ainsi, il est par exemple possible de *programmer* une location de véhicule. Le contrat permet de démarrer la course automatiquement à la réception du paiement du client, et de verser l'argent bloqué durant toute la durée du transport, sur le compte du propriétaire. Nous vous raconterons cette histoire très prochainement sur ce blog et il ne s'agira que d'une application des *smart-contracts* parmi les nombreux possibles.

Vous pouvez également interagir avec un *smart-contract* déployé sur une blockchain avec [Ethereum-android](#). La librairie permet d'appeler des fonctions du *smart-contract* via une transaction (persistance de la modification de l'état du *smart-contract*) ou via un appel local (sans persistance).

C'est l'histoire d'un compteur

Prenons l'exemple d'un *smart-contract* *compteur*, implémenté en [langage Solidity](#) et présenté ci-dessous.

```
contract Counter {
    int public counter;
    event fivesMore(int);

    function Counter(){
        counter = 0;
    }

    function increase() {
        counter++;
        if(counter % 5 == 0) {
            fivesMore(counter);
        }
    }
}
```

Ce contrat présente :

- 1 constructeur ; le constructeur *Counter* permet d'initialiser la valeur du compteur à 0.
- 1 attribut ; l'attribut *counter* ayant la visibilité *public*, un getter est automatiquement généré.
- 1 événement ; l'événement *fivesMore* va permettre d'être notifié à chaque fois que le compteur passe par un multiple de 5.
- 1 fonction ; la fonction *increase* permet d'incrémenter de 1 la valeur du compteur, et générer l'événement dans les conditions présentées précédemment.

Mais d'abord...

Il est indispensable de déployer en amont le contrat sur la blockchain car [Ethereum-android](#)

ne le permet pas pour le moment. La communauté Ethereum a développé un [compilateur Solidity en ligne](#) capable de s'exécuter dans un navigateur. Cet outil permet d'écrire des smart-contracts, de les compiler, d'obtenir le code compilé et même d'exécuter le contrat en local (dans le navigateur), ou sur une blockchain particulière via une connexion RPC à un noeud existant. Un mode debug extrêmement détaillé est également disponible pour exécuter pas à pas les fonctions du contrat. Une fois le code compilé, l'outil propose un script de déploiement du smart-contract destiné à l'interpréteur Javascript de Geth. Vous n'avez donc plus qu'à copier ce script et à le coller dans l'interpréteur.

L'interface du contrat (var abi...) ainsi que son adresse (contract.address) devront être renseignées dans l'application Android. Vous obtiendrez l'adresse dans les logs de l'interpréteur comme spécifié dans le script ci-dessus (Code : `console.log('Contract mined! address: ' + contract.address[...]);`), ou bien en appelant `counter.address` dans l'interpréteur. L'adresse n'est disponible qu'une fois que la transaction du déploiement du contrat a été prise en compte par un mineur.

Étape 1

La première étape pour pouvoir faire appel à un contrat depuis l'application Android est de convertir l'interface du contrat du format Json (variable...) en Java. Cette interface équivalente est :

```
interface Counter extends ContractType {
```

```
@SolidityEvent.Anonymous(false)
@SolidityEvent.Parameters({
    @SolidityEvent.Parameter(indexed = false, name = "fivesMore", type = SInt256.class)
})
SolidityEvent<SInt256> fivesMore();

@SolidityFunction.ReturnType(SInt256.class)
SolidityFunction<SInt256> counter();

@SolidityFunction.ReturnType(SVoid.class)
SolidityFunction<SVoid> increase();
```

Nous pouvons noter qu'il s'agit d'une interface Java qui étend le type `ContractType` et que chaque méthode de l'interface correspond aux fonctions exposées par notre *smart-contract*. Les méthodes retournent des `SolidityFunction` et des `SolidityEvent`.

Dans l'idée du compteur sur la blockchain, nous pourrions appeler la fonction `counter()` sans persistance et `increase()` avec persistance afin de respectivement récupérer et modifier la valeur du compteur.

Chaque SolidityFunction est annotée par `ReturnType`. Ce qui permet de spécifier le type de retour de façon statique car le type générique d'un objet Java est perdu à la compilation et non disponible au runtime.

Étape 2

La seconde étape consiste à récupérer l'instance du contrat dans l'application. C'est à ce moment que nous renseignons son interface Java spécifiée dans l'étape précédente, ainsi que son adresse sur la blockchain.

```
String contractAddress = "0xffffffffffffffffffff" /* put here your contract address*/
ContractSample contract = ethereumjava.contract.withAbi(Counter.class).at(contract
Address);
```

Appeler des fonctions d'un smart-contract

L'application est désormais capable d'appeler localement ou via la blockchain les fonctions du *smart-contract* :

Pour appeler localement on utilise la fonction `SolidityFunction#call()`. Il s'agit en fait d'un *getter* de la variable `counter` qui lit sa valeur de type `int`.

```
Slnt256 counter = contract.counter().call(); //--> is 0 at start
```

L'appel d'une fonction de *smart-contract* de manière persistante est asynchrone. En effet, il est nécessaire d'attendre que la transaction contenant l'appel à la fonction soit minée.

Pour pouvoir être notifié de l'inscription de notre transaction dans la blockchain, nous utilisons la méthode `SolidityFunction#sendTransactionAndGetMined()`. Il s'agit d'une méthode spécifique à [Ethereum-android](#) qui encapsule beaucoup de code et dont nous avons souvent besoin.

Cette méthode prend à minima 2 paramètres :

- L'identifiant du compte qui appelle cette fonction,
 - Un montant de *gas*.

- Un montant de *gas*.

Le *gas* est une notion spécifique aux smart-contracts d'Ethereum. Ce mécanisme permet d'éviter les boucles infinies lors de l'exécution des smart-contracts. Le code s'exécute sur la machine virtuelle Ethereum (EVM) de chaque noeud du réseau à la réception de la transaction contenant l'appel à une fonction de smart-contract (une fois ajoutée à un bloc

par un mineur). S'agissant de code exécutable contenant une suite d'instructions, chaque instruction peut prendre plus ou moins de temps à s'exécuter en fonction de son type. Chaque type possède un *coût fixe* en unités de gas. Une unité de gas ayant un prix, le compte qui appelle une fonction de smart-contract devra veiller à intégrer dans la transaction suffisamment d'unités de gas et posséder le montant associé sur son compte. Le nombre d'unités nécessaires est indiqué lors des tests avec le compilateur en ligne. Il est commun de fournir un nombre entier supérieur à l'estimation. Soyez rassurés, la somme associée au montant de gas non utilisé n'est pas prélevée sur le compte.

```
String accountId = "0x000000000000..."; /* put here the account which call the
function */
contract.increase().sendTransactionAndGetMined(accountId, new BigInteger("90000")) // gas
.observeOn(AndroidSchedulers.mainThread())
.subscribe(t -> {
    System.out.println(t);
    /*
    Call has been made on the contract
    This can checked with the transaction reference
    */
});
```

Observez la redirection du flux de données sur le thread principal d'Android, à la sauce RxJava, grâce à l'opération observeOn. Cet appel déclenche un changement de thread dans l'exécution de la chaîne d'opérations du flux de données. Toutes les opérations qui s'exécutent après le observeOn se feront sur le *main thread* d'Android. Ainsi, il sera possible de modifier directement la vue depuis la méthode call() du subscribe. Il faut également préciser que, comme le spécifie RxJava, la méthode subscribe() peut prendre une Action en unique paramètre. Cela correspond au comportement à avoir lors de la réception d'une nouvelle donnée (onNext).

À présent, si nous récupérons la valeur du compteur via

```
contract.counter().call();
```

La valeur renournée est égale à 1.

Souscrire à des événements de smart-contract

Le smart-contract compteur propose d'être notifié toutes les 5 incrémenta-

tions. Voici comment souscrire à cette notification.

```
contract.fivesMore().watch()
.observeOn(AndroidSchedulers.mainThread())
.subscribe(c -> {
    System.out.println(c);
    /*
    You've been notified that counter was incremented 5 more times

    Parameter 'c' has the current counter's value.
    You can directly update your app view
    */
});
```

De la même manière que lors de l'appel à sendTransactionAndGetMined, nous redirigeons le flux du traitement sur le thread principal d'Android. L'intérieur de la fonction call() sera exécuté à chaque fois que l'événement fivesMore sera émis sur la blockchain.

That's all folks

Vous êtes à présent prêt à développer votre première application Android décentralisée, contenant un nœud blockchain complet. À ce stade, vous pourrez :

- Vous connecter à une blockchain depuis votre smartphone ;
- Créer des portefeuilles virtuels ;
- Emettre des transactions sur le réseau ;
- Analyser en profondeur la blockchain ;
- Récupérer un smart-contract déployé sur une blockchain ;
- Communiquer avec un smart-contract déployé ;
- Etre notifié lorsqu'une transaction est inscrite dans la blockchain ;
- Etre notifié lors de l'émission d'un événement Solidity d'un smart-contract.

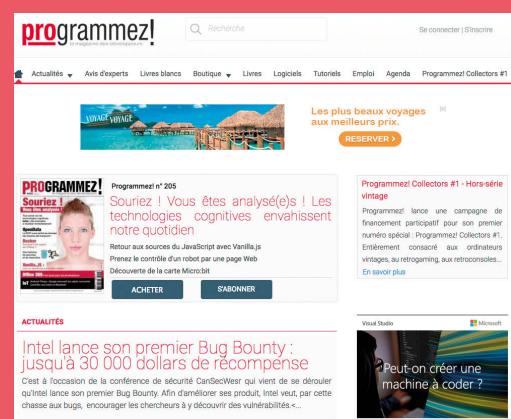
Important : ce projet est actuellement en cours de développement et vous est proposé en version alpha, non stable. Il est destiné à un usage expérimental et ne doit être en aucun cas être utilisé en production.

La librairie va continuer à évoluer. Par exemple, nous souhaitons limiter les contraintes d'architecture de l'application Android en déplaçant le lancement du service Geth en dehors de la classe Application. De plus, pour le moment, toutes les méthodes de l'API exposée par Geth ne sont pas implémentées.

Restez connecté(e) à l'actualité !

- **L'actu** de Programmez.com : le fil d'info quotidien
- **La newsletter hebdo** : la synthèse des informations indispensables.
- **Agenda** : Tous les salons, barcamp et conférences.

Abonnez-vous, c'est gratuit ! www.programmez.com



La reconnaissance d'images à portée de main avec l'**API Cloud Vision** de Google



Aurélie Vache
Lead Developer chez **Atchik**
Duchess France
& DevFest Toulouse Leader
@aurelievache

L'API Cloud Vision de Google, qui est sortie récemment en GA (General Availability) est un nouveau service de reconnaissance d'images, faisant partie de l'offre de Machine Learning (apprentissage automatique) de Google. Grâce à cette API, Google continue de démocratiser les technologies de Machine Learning utilisées en interne et fournit aux développeur.se.s une simple API REST extrêmement puissante. [1]

Que permet la Cloud Vision API ?

L'API Cloud Vision fournit un ensemble de fonctionnalités pour analyser des images. Google Cloud Vision permet aux développeurs d'utiliser le pouvoir du Machine Learning afin de collecter des informations sur une image. Les caractéristiques des images détectées peuvent être par exemple les principales couleurs d'une image, les principaux objets qui la composent, la détection de visages, de logos, de contenus non appropriés à caractère sexuel ou de violence, de localisation, de texte dans les images via l'OCR (Optical Character Recognition).

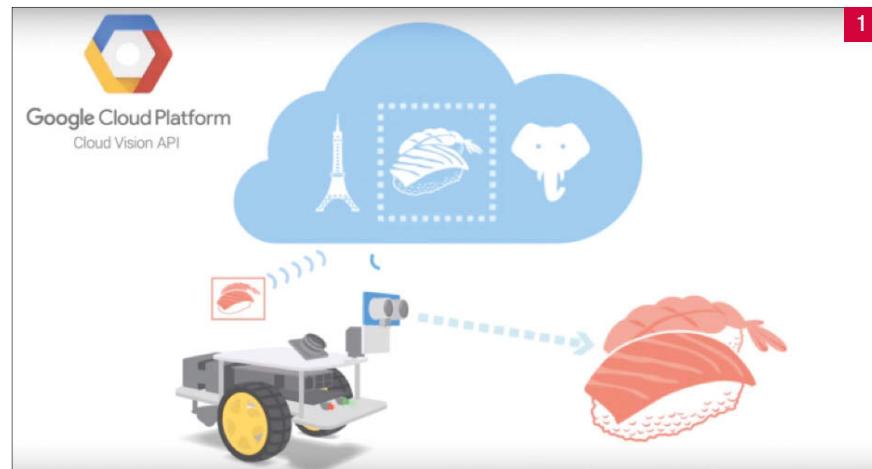
Si vous appelez l'API REST avec une image d'une pizza par exemple, vous pourrez obtenir comme résultat qu'il s'agit de nourriture (food) et d'une pizza ;-. Par exemple, plus vous demanderiez de LABEL et plus vous obtiendriez de résultats vous permettant de classifier une image. L'API de Vision est utilisée en interne par Google; sans le savoir nous utilisons donc des outils de Google qui utilisent cette API, tels que Street View, Photos, Search Image et Translate.

[2]

REST API

Afin d'appeler l'API REST il suffit de faire une requête POST en utilisant le endpoint `images:annotate` avec comme requête la feature que l'on veut utiliser :

```
POST https://vision.googleapis.com/v1/images:annotate?key=YOUR_API_KEY
{
  "requests": [
    {
      "images": {
        "content": "/9j/7QBEUGhvdG9zaG9...base64-encoded-image-content...fxNW
zvDEeYxxzj/Coa6Bax//Z"
      }
    }
  ]
}
```



```
"features": [
  {
    "type": "LABEL_DETECTION", //voir la liste des types de features possibles
    plus bas
  }
]
```

Les exemples de réponses seront précisés lors du listing des features possibles.

Librairies

Comme d'habitude, vous pouvez utiliser les librairies fournies par Google dans vos projets en Java, Python, C#, Go, NodeJS, PHP et Ruby. Allons maintenant voir ce que l'API a sous le capot comme fonctionnalités.

Quelles sont les fonctionnalités de Cloud Vision ?

- LABEL_DETECTION : une analyse performante de l'image afin d'en extraire une large série de catégories sous forme de mots-clés descriptifs.

```
{
  "responses": [
    {
      "labelAnnotations": [
        {
          "mid": "/m/0bt9lr",
          "description": "dog",
          "score": 0.96969336
        },
        {
          "mid": "/m/04rk",
          "description": "mammal",
          "score": 0.92070323
        },
        {
          "mid": "/m/09686",
          "description": "vertebrate",
          "score": 0.89664793
        }
      ]
    }
  ]
}
```

- TEXT_DETECTION : grâce à la technologie OCR, l'API va être capable d'extraire les informations textuelles d'une image. On fournit à l'API par exemple une capture écran et l'API y extrait le texte contenu dans celle-ci en y détectant la langue du texte (locale). Cette fonctionnalité peut être très utile pour reconnaître le texte sur une plaque d'immatriculation, sur un panneau de signalisation, sur la plaque du nom des rues.

```
{
  "responses": [
    {
      "textAnnotations": [
        {
          "locale": "fr",
          "description": "Salut, ça va ?\nOui et toi ?",
          "boundingPoly": {
            "vertices": [
              {
                "x": 53,
                "y": 15
              }
            ]
          }
        }
      ]
    }
  ]
}
```

```
{
  "x": 272,
  "y": 15
},
]
}
},
{
  "description": "Vive l'extraction de texte dans une capture d'écran !",
  "boundingPoly": {
  ...
}
```

- FACE_DETECTION : détection des visages dans les images incluant une reconnaissance des sentiments (joie, tristesse, etc.).

```
{
  "responses": [
    {
      "faceAnnotations": [
        {
          "boundingPoly": {
            "vertices": [
              {
                "x": 669,
                "y": 324
              },
              {
                "x": 760,
                "y": 324
              },
              ...
            ]
          },
          "fdBoundingPoly": {
          ...
        },
        {
          "type": "CHIN_RIGHT_GONION",
          "position": {
            "x": 861.7816,
            "y": 568.26685,
            "z": 50.14255
          }
        },
        ],
        "rollAngle": -5.0866733,
        "panAngle": 1.5809642,
        "tiltAngle": -18.30349,
        "detectionConfidence": 0.9379803,
        "landmarkingConfidence": 0.24303582,
        "joyLikelihood": "VERY_LIKELY",
        "sorrowLikelihood": "VERY_UNLIKELY",
        "angerLikelihood": "VERY_UNLIKELY",
        "surpriseLikelihood": "VERY_UNLIKELY",
        "underExposedLikelihood": "VERY_UNLIKELY",
        "blurredLikelihood": "VERY_UNLIKELY",
        "headwearLikelihood": "VERY_UNLIKELY"
      }
    }
  ]
}
```

```
...
],
"landmarkAnnotations": [
  {
    "mid": "/m/0c7zy",
    "description": "Petra",
    "score": 0.5403372,
    "boundingPoly": {
      "vertices": [
        {
          "x": 153,
          "y": 64
        },
        ...
      ]
    },
    "locations": [
      {
        "latLng": {
          "latitude": 30.323975,
          "longitude": 35.449361
        }
      }
    ]
  }
]
```

- LANDMARK_DETECTION : détection d'éléments/de localisation géographiques et architecturaux (monument, montagne...).

```
{
  "responses": [
    {
      "landmarkAnnotations": [
        {
          "mid": "/m/0260wgf",
          "description": "Capitole de Toulouse",
          "score": 0.8286499,
          "boundingPoly": {
            "vertices": [
              {
                "x": 883,
                "y": 899
              },
              {
                "x": 4220,
                "y": 899
              },
              ...
            ]
          },
          "locations": [
            {
              "latLng": {

```

```
    "latitude": 43.604416,
    "longitude": 1.443361
  }
}
]
}
]
}
```

- LOGO_DETECTION : détection des logos d'entreprises et de marques dans des images.

```
{
  "responses": [
    {
      "logoAnnotations": [
        {
          "description": "Programmez",
          "score": 0.58307266,
          "boundingPoly": {
            "vertices": [
              {
                "x": 14,
                "y": 22
              },
              {
                "x": 449,
                "y": 22
              },
              {
                "x": 449,
                "y": 92
              },
              {
                "x": 14,
                "y": 92
              }
            ]
          }
        }
      ]
    }
  ]
}
```

- SAFE_SEARCH_DETECTION : utilisation de la technologie SafeSearch de Google qui permet de déterminer si une image est inappropriée ou explicite (contenu pour adulte, violence...). SafeSearch est utilisé dans les résultats de recherche de Google.

```
{
  "responses": [
    {
      "safeSearchAnnotation": {
        "adult": "POSSIBLE",
        "spoof": "POSSIBLE",

```

```

  "medical": "LIKELY",
  "violence": "POSSIBLE"
}
}
]
}

```

- IMAGE_PROPERTIES : calculez un ensemble de propriétés sur l'image (telles que les couleurs dominantes de l'image).

```

{
  "responses": [
    {
      "imagePropertiesAnnotation": {
        "dominantColors": {
          "colors": [
            {
              "color": {
                "red": 69,
                "green": 42,
                "blue": 27
              },
              "score": 0.15197733,
              "pixelFraction": 0.14140345
            },
            ...
          ]
        }
      }
    }
  ]
}

```

Nouveautés de la version 1.1 (sortie lors du Cloud Next 17) :

- WEB_DETECTION : la fonctionnalité va s'appuyer sur la puissance de Google Image Search pour rechercher des informations sur l'image demandée, cela permet également d'obtenir l'url des images qui sont similaires à celle recherchée.

```

{
  "responses": [
    {
      "webDetection": {
        "webEntities": [
          {
            "entityId": "/m/0cbhh",
            "score": 8.57296,
            "description": "Toulouse"
          },
        ]
      }
    }
  ]
}

```

Web annotations



2b

```

{
  "entityId": "/m/0404s6",
  "score": 6.7826004,
  "description": "Arno"
},
{
  "entityId": "/m/0dy6rb",
  "score": 2.98696,
  "description": "Sée"
],
{
  "fullMatchingImages": [
    {
      "url": "https://pbs.twimg.com/media/CwWuUI4XAAEaX2.jpg"
    },
    {
      "url": "https://pbs.twimg.com/media/Cx72uZLxcAlxpRg.jpg"
    },
    ...
  ],
  "partialMatchingImages": [
    {
      "url": "https://pbs.twimg.com/media/CwWuUI4XAAEaX2.jpg"
    },
    ...
  ]
}

```

Les objets *fullMatchingImages*, *partialMatchingImages* et *pagesWithMatchingImages* vont permettre de savoir où est-ce que cette image existe/est hébergée sur Internet. [2b] [3]

- CROP_HINTS : pour savoir comment "cropper"/couper de manière optimale son image

```

{
  "responses": [
    {
      "cropHintsAnnotation": {
        "cropHints": [
          {
            "boundingPoly": {
              "vertices": [
                {},
                {}
              ]
            }
          }
        ]
      }
    }
  ]
}

```



Web Entities

Toulouse	8.57296
Arno	6.7826
Sée	2.98696
Twitter	0.74142
Software Developer	0.48484
MySQL	0.38213
Image	0.38201
Hashtag	0.37943
Media	0.37881
Craft	0.37137
technics	0.34595
PHP	0.31021
France	0.00013

Pages with Matched Images

```
        "x": 240
    },
    {
        "x": 240,
        "y": 65
    },
    {
        "y": 65
    }
],
},
"confidence": 0.79999995,
"importanceFraction": 1
}
]
}
}
]
}
```

Limites

Qui dit API dit généralement limitation, les voici :

- 4Mo par image ;
 - 8Mo par requête ;
 - 10 requêtes par seconde ;
 - 700 000 requêtes par feature par jour ;
 - 20 000 000 requêtes par feature par mois ;
 - 8 images par seconde ;
 - 16 images par requête.

Comme d'habitude avec les services de Google, les limitations évoluent et elles ont tendance à augmenter dans le temps. Je vous conseille d'aller les regarder de temps en temps : <https://cloud.google.com/vision/docs/limits>

Coût

Une des philosophie des service de Google est de ne payer que pour l'utilisation que vous en faites, tout comme avec BigQuery. Voici le tableau des coûts : [4]

Pour réaliser un POC (Proof Of Concept), pas de soucis il n'y aura pas de coût mais une fois dépassées les 1000 unités, la facture risque de grimper vite. Quand je pense que le stockage et l'analyse de nos données mensuelles chez Atchik coûtent une vingtaine de centimes de dollars par mois, je me dis que la note risque de devenir salée si on utilise plusieurs features pour un grand volume d'images.

Plus d'information sur les coûts : <https://cloud.google.com/vision/docs/pricing>

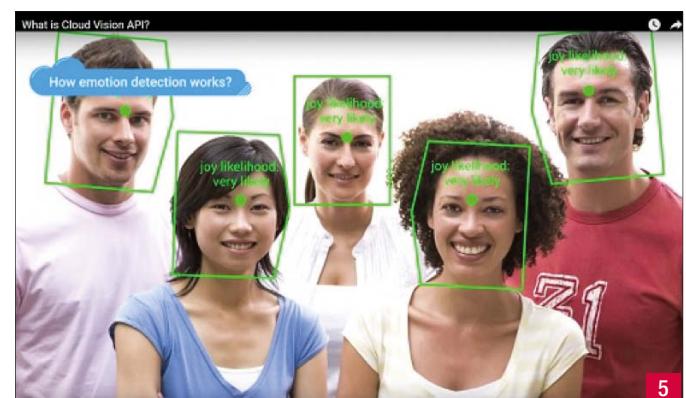
Quelle utilisation peut-on faire de cette API ?

L'utilisation d'une telle API permet d'ouvrir le champ des possibles :

- La reconnaissance de **labels** (de mots-clés) dans une image permet plusieurs possibilités comme l'amélioration de l'indexation et la catégorisation de ces dernières, ainsi que la génération automatique de nuage de tags.

```
{  
  "responses": [  
    {  
      "textAnnotations": [  
        {  
          "locale": "en",  
          "description": "\n",  
          "boundingPoly": {  
            "vertices": [  
              {  
                "x": 577,  
                "y": 104  
              },  
              {  
                "x": 578,  
                "y": 104  
              },  
              {  
                "x": 578,  
                "y": 108  
              }  
            ]  
          }  
        ]  
      }  
    ]  
  ]  
}
```

Feature	1 - 1000 units/month	1001- 1,000,000 units/month	1,000,001 to 5,000,000 units/month	5,000,001 - 20,000,000 units/month
Label Detection	Free	\$1.50	\$1.50	\$1.00
OCR	Free	\$1.50	\$1.50	\$0.60
Explicit Content Detection	Free		Now free with Label Detection*	
Facial Detection	Free	\$1.50	\$1.50	\$0.60
Landmark Detection	Free	\$1.50	\$1.50	\$0.60
Logo Detection	Free	\$1.50	\$1.50	\$0.60
Image Properties	Free	\$1.50	\$1.50	\$0.60
Web Detection	Free	\$3.50	\$3.50	Contact Google for more information
Document Text Detection	Free	\$3.50	\$3.50	Contact Google for more information



- La reconnaissance **faciale** permet de lister les images contenant des personnes, de connaître l'état émotionnel d'une personne et de faire de l'analyse de sentiment sur des photos. [5]
 - La reconnaissance de **texte** dans des images pourrait être intégrée dans une voiture connectée par exemple. On pourrait envisager d'analyser les photos prises par un capteur relié à la voiture, la reconnaissance de texte sur les panneaux environnant la voiture pourrait permettre au conducteur d'être mieux guidé qu'un GPS standard par exemple. [6]
 - La reconnaissance de **logo** est très utile pour les marques qui vont pouvoir analyser un énorme échantillon de photo et analyser l'utilisation de leurs produits, l'affichage de leur marque.
- Analyser un texte est généralement plus facile qu'analyser du texte dans une image, grâce à cette API les deux sont désormais de complexité équivalente. Les services de traduction vont notamment pouvoir être améliorés grâce à cette API.
- La reconnaissance **SafeSearch** est la raison pour laquelle je me suis intéressée à la Cloud Vision API. Savoir si une image est sensible ou non est une donnée qui peut être très utile pour améliorer nos filtres de modération automatisée et intelligente. Les filtres sont créés pour aider les humains afin de traiter efficacement et rapidement un message.
- On peut également utiliser cette feature dans une application qui permettrait à ses utilisateurs d'uploader un avatar pour refuser automatiquement les avatars sensibles, de type violent ou à caractère sexuel.
- La reconnaissance **Web** peut permettre de lister tous les sites qui hébergent votre image par exemple, et qui font du plagiat; donc pour effectuer de la détection de copyright c'est parfait.

Cas pratique

Création d'un compte Google

Si vous n'avez pas encore de compte Google, vous devez en créer un. Connectez-vous sur la console de Google Cloud (console.cloud.google.com) et créez un projet.

Notez l'ID du projet, vous en aurez besoin plus loin.

Ensuite il faut activer le paiement (billing) dans la console pour développeurs afin de pouvoir utiliser les ressources de Google Cloud telles que le Datastore et le Storage.

Les nouveaux utilisateurs de la plateforme Cloud de Google peuvent avoir accès à un free trial de 300€ (<https://cloud.google.com/free-trial>).

Authentification (credentials)

Afin d'utiliser l'API dans une application, il faut que cette dernière soit authentifiée. Le mécanisme d'authentification le plus simple implique le passage d'une clé API directement au service. Lors de l'utilisation de l'API Vision, il est recommandé d'activer une clé API à des fins de test et un compte de service pour l'utilisation de l'API en production.

- Allez sur la console de google pour développeurs (<https://console.cloud.google.com>) ;
- Selectionnez le projet que vous avez créé ;
- A gauche, cliquez sur le menu hamburger puis sur Gestionnaire d'API > Identifiants ;
- Cliquez sur Créer des identifiants > Clé API ;
- Selectionnez Clé de navigateur pour cette clé puis cliquez sur Créer ;
- Copiez cette clé, vous allez en avoir besoin.

Activer la Cloud Vision API

La clé que vous avez créée permet à votre application de s'authentifier auprès de l'API Google Cloud Vision. Pour activer l'API :



6

Cliquez sur le menu hamburger puis sur Gestionnaire d'API >Activer une API ;

- Cherchez "Cloud Vision API" ;
- Cliquez sur la Cloud Vision API ;
- Puis sur Activer l'API.

Trève de bla bla, place au code !

Afin de tester ce qu'avait dans le ventre cette API, j'ai réalisé une petite classe Java permettant de tester les différentes features.

On va commencer par exporter sa clé API dans son environnement (on pourrait l'ajouter dans son fichier .bashrc pour ne pas avoir à le répéter à chaque fois par exemple) :

```
$ export GOOGLE_APPLICATION_CREDENTIALS=<path_to_service_account_file>
```

Ensuite, si vous utilisez Maven, il faudra ajouter la dépendance :

```
<!-- https://mvnrepository.com/artifact/com.google.apis/google-api-services-vision -->
<dependency>
  <groupId>com.google.apis</groupId>
  <artifactId>google-api-services-vision</artifactId>
  <version>v1-rev347-1.22.0</version>
</dependency>
```

On va maintenant écrire une petite classe Java permettant d'utiliser toutes les features que l'API permet au moment où je rédige cet article : **code complet sur notre site ou directement sur :**

<https://github.com/scrally/java-cloud-vision-api/blob/master/GoogleTest.java>

Si vous n'êtes pas trop Java, vous pouvez tester l'API via le Google Cloud Shell (<https://cloud.google.com/shell/docs/> & https://console.cloud.google.com/cloud-shell?project=api-project-<votre_id_projet>&hl=fr) :

On commence par exporter sa clé API pour y accéder dans le shell

```
scrally@api-project-666:~$ export API_KEY=xxx
```

On va ensuite se créer un fichier JSON en initialisant une requête en appelant la fonctionnalité détection de Label sur une image que l'on met dans notre bucket sur le Cloud Storage :

```
scrally@api-project-666:~$ vi request.json
```

Décoder le passé

PHARAON

PHARAON

LE MAGAZINE DE L'ÉGYPTE ÉTERNELLE

MAI-JUIN-JUILLET-2017 - N°29

Le trésor de la tombe de Toutankhamon

► La redécouverte
de l'ancienne
Héliopolis

► L'identité
de Ramsès VIII
enfin découverte ?

► Le règne
du roi Abou Hor



Imprimé en UE - Printed in EU - BEL/LUX : 7,90€ - PORT.CONV : 1,50€ - P-F/U/S : 8,50€ - CH : 12,30 FS - CAN : 12,50\$CAD

Kiosque | Abonnement | PDF

www.pharaon-magazine.fr

```
{
  "requests": [
    {
      "image": {
        "source": {
          "gcsImageUri": "gs://your_bucket/demo-image.jpg"
        }
      },
      "features": [
        {
          "type": "LABEL_DETECTION",
          "maxResults": 10
        }
      ]
    }
  ]
}
```

On donne les droits nécessaires pour accéder à l'image et y récupérer des informations.

```
$ gsutil acl ch -g AllUsers:R gs://your_bucket/demo-image.jpg
```

Si on n'exécute pas cette commande, on obtiendra l'erreur suivante :

```
image-annotator::User lacks permission.: Can not open file: gs://your_buket/demo-image.jpg
```

On appelle l'API de vision avec la requête précédemment paramétrée

```
$ curl -s -X POST -H "Content-Type: application/json" --data-binary @request.json https://vision.googleapis.com/v1/images:annotate?key=${API_KEY}
```

On obtient la réponse au format JSON contenant des labelAnnotations :

```
{
  "responses": [
    {
      "labelAnnotations": [
        {
          "mid": "/m/02wbm",
          "description": "food",
          "score": 0.96290076
        },
        {
          "mid": "/m/02q08p0",
          "description": "dish",
          "score": 0.9487846
        }
      ]
    }
  ]
}
```

```
    },
    {
      "mid": "/m/068_x",
      "description": "pizza cheese",
      "score": 0.9431161
    }
  ]
}
]
```

```
$ gsutil acl ch -g AllUsers:R gs://your_bucket/selfie.jpeg
Updated ACL on gs://your_bucket/selfie.jpeg
```

```
$ vi selfierequest.json
```

```
$ curl -s -X POST -H "Content-Type: application/json" --data-binary @selfierequest.json https://vision.googleapis.com/v1/images:annotate?key=${API_KEY}
```

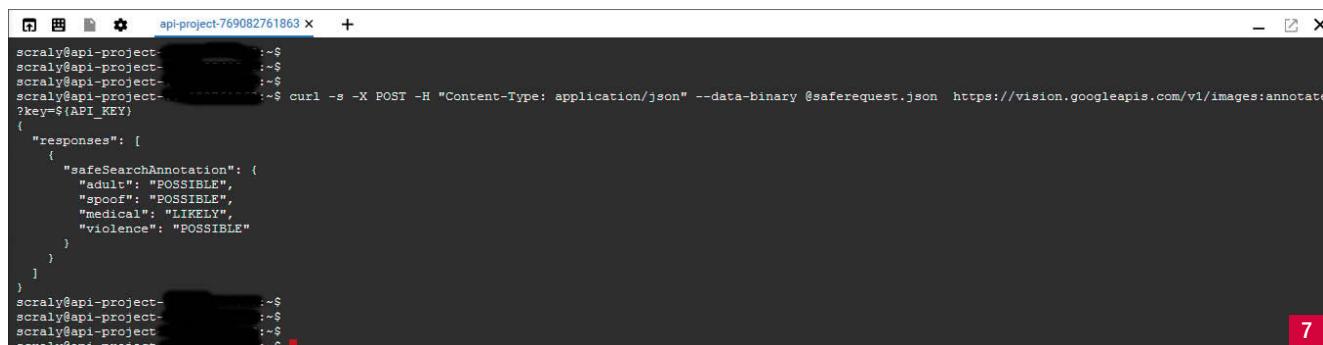
Avec SAFE_SEARCH, une fonctionnalité que j'aime bien, on peut très facilement et rapidement savoir si une photo est du contenu pour les adultes, montrant de la violence, dans le domaine médical ou catégorisé comme spoof. [7]

On peut également interagir avec l'API avec le CLI de Google :

```
gcloud beta ml vision detect-documents
gcloud beta ml vision detect-faces
gcloud beta ml vision detect-image-properties
gcloud beta ml vision detect-labels
gcloud beta ml vision detect-landmarks
gcloud beta ml vision detect-logos
gcloud beta ml vision detect-safe-search
gcloud beta ml vision detect-text
gcloud beta ml vision detect-web
```

CONCLUSION

En démocratisant l'API Cloud Vision, Google permet encore une fois aux développeurs d'accéder à des algorithmes extrêmement puissants et performants avec une simple API REST. Y accéder et la coupler avec la Natural Language API, stocker les données dans BigQuery et les visualiser dans Data Studio est à la portée de tous. La prochaine étape pour ma part va être d'estimer le coût que cela peut avoir pour utiliser Vision en production et quels avantages productifs cela peut nous donner. Vous pouvez retrouver l'intégralité du code source sur mon repository Github : <https://github.com/scrally/java-cloud-vision-api>



```
scrally@api-project-769082761863 ~
scrally@api-project- ~$ curl -s -X POST -H "Content-Type: application/json" --data-binary @selfierequest.json https://vision.googleapis.com/v1/images:annotate?key=${API_KEY}
{
  "responses": [
    {
      "labelAnnotations": [
        {
          "mid": "/m/068_x",
          "description": "pizza cheese",
          "score": 0.9431161
        }
      ]
    }
  ]
}
]
```

Commodore 64 : objet mythique

- Timeline : 1983
- Objet : ordinateur
- Constructeur : Commodore

Commodore fut un des constructeurs les plus emblématiques des années 1970 et 80. Sa rivalité avec Atari a été fracasante comme nous l'avons vu le mois dernier. Parmi les plus belles machines du constructeur, nous avons le Commodore 64 ou C64. Il fut officiellement montré au public en janvier 1982 au fameux salon électronique : le CES. Résultat : +10 millions d'unités vendues ! Le C64 est un ordinateur personnel 8-bit, à l'instar des Atari 400 et 800. Commodore produisait des ordinateurs depuis plusieurs années. Le premier véritable modèle fut le PET, une machine monobloc datant de 1977. Trois ans plus tard, le constructeur sort un modèle très important pour la suite : le VIC-20. Il s'agit du premier ordinateur personnel de la marque en 8-bit. Il inaugure un design plus compact qui sera repris par le C64 et ses successeurs. Malgré les limitations matérielles, le VIC-20 fut un gros succès auprès du grand public. Il possède un processeur MOS 6502 1 Mhz, 5 ko de mémoire vive (4 réellement utilisable), un support 16 couleurs, plusieurs interfaces pour une manette, les cartouches, la vidéo, etc. Le tout fonctionnant avec un logiciel intégré : ROM BASIC. L'ordinateur possédait 16 Ko de ROM.

La révolution C64

Reprenant la même stratégie que le VIC-20, Commodore présente rapidement le C64. Les deux modèles partagent beaucoup de choses comme le design, un processeur 8-bit. Mais pour répondre aux critiques, le C64 booste les ressources matérielles : 3 canaux audio, un processeur plus récent, plus de ports et surtout une



mémoire vive de 64 Ko et 20 Ko de ROM. Le C64 bénéficiait de performances graphiques et audio meilleures que les concurrents directs (Apple II et Atari 400/800 ou encore le TI99). Un des clés du succès fut la disponibilité de la machine dans les magasins « normaux » et plus uniquement chez les revendeurs et les magasins d'électronique. Cette visibilité était risquée mais fut une complète réussite. Pour proposer une machine à un tarif accessible, le constructeur traquait les coûts et notamment en produisant lui-même plusieurs composants clés comme les puces. Le C64 démocratisa l'accès à l'informatique. Le surplus de ressources, par rapport au VIC-20, a fait exploser l'écosystème et plusieurs milliers de logiciels étaient disponibles sur différents supports. Les jeux étaient bien entendu importants mais les logiciels personnels étaient aussi très répandus. Même si le lecteur de K7 était toujours aussi lent. Un lecteur de disquette externe fut proposé mais il souffrait de lenteurs et d'une certaine fragilité.

La disponibilité des outils de développement et

la possibilité de bidouiller la machine ont fait le reste.

Pour répondre à la concurrence et attirer de nouveaux utilisateurs, le constructeur sort en 1984 le Commodore 16 qui reprend le même design que le C64, à quelques détails près sur les touches et l'organisation du clavier. Pour réduire les coûts et donc le prix de vente, le C16 embarque des ressources réduites : mémoire vive de 16 Ko. Mais ce modèle propose aussi des nouveautés intéressantes comme une nouvelle version de BASIC, la v3.5 mais les capacités graphiques brident le potentiel de la machine.

Une machine robuste mais susceptible

On dispose de deux ports joysticks (standard Atari), de la possibilité d'utiliser des cartouches, d'une sortie A/V et via le câble antenne. Vous trouverez facilement un câble périphérique compatible C64. Un port permet de connecter une imprimante ou un lecteur de disquettes. Le lecteur K7 possède son propre port (format propriétaire) et enfin, le C64 possède un port dit utilisateur, accessible aux développeurs et pratique pour y connecter, par exemple, une interface RS232. Ce port est l'héritage des modèles précédents et est lui aussi propriétaire. Ce port était piloté par le circuit intégré 6522 de MOS Technology, compatible avec les processeurs 6502. Il possède 20 lignes I/O. Ce port est accessible depuis le BASIC intégré à la machine.

Un exemple est proposé sur ce site : <https://coronax.wordpress.com/2012/08/11/the-commodore-64-user-port/>

Sur le design général, il existe quelques variantes. Par exemple, le C64C reprend le design plus plat du C128. Le modèle C64G reprend le design du C64 d'origine. Il possédait un OS graphique, le GEOS.



Cotation et rareté

Rareté : ■■■

Côte : ■■■

Accessoires : ■■■

Logiciels : ■■■

Le C64 n'est pas un modèle rare. Il est facile à trouver. Entre 50 et 70 €, il est possible de trouver un ensemble complet : C64 + lecteur de K7 + logiciels + câbles. Eventuellement, jusqu'à 80-90 € pour un modèle en parfait état (caisse non jaunie) avec lecteur de disquettes.

Au-delà de 100 €, c'est trop cher.

Sur les périphériques et accessoires, le marché est très actif et vous trouverez facilement. Attention aux tarifs.

Quand on laisse un codeur en autonomie



CommitStrip.com



Une publication Nefer-IT, 7 avenue Roger Chambonnet, 91220 Brétigny sur Orge - redaction@programmez.com
Tél. : 01 60 85 39 96 - Directeur de la publication & Rédacteur en chef : François Tonic

Secrétaire de rédaction : Olivier Pavie
Ont collaboré à ce numéro : S. Saurel
Nos experts techniques : V. Sevel, J-M Heitz, J. Milleron, M. Kassighian, E. Manuguerra, J. Cadeau, N. Chabrier,

L. Yin, M. Pernin, J-B Paux, C. Gigax, F. Pigere, A. vache, B. Homès,
Couverture : © D.R. - Maquette : Pierre Sandré.

Publicité : PC Presse, Tél. : 01 74 70 16 30, Fax : 01 40 90 70 81 - pub@programmez.com.

Imprimerie : 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

Responsable titre : Terry MATTARD Téléphone : 09 67 32 09 34

Contacts : Rédacteur en chef : ftonic@programmez.com - Rédaction : redaction@programmez.com - Webmaster : webmaster@programmez.com -
Publicité : pub@programmez.com - Evenements / agenda : redaction@programmez.com

Dépot légal : à parution - Commission paritaire : 1220K78366 - ISSN : 1627-0908 - © NEFER-IT / Programmez, juin 2017

Toute reproduction intégrale ou partielle est interdite sans accord des auteurs et du directeur de la publication.

Encart, catalogue Pcssoft jeté sur une partie du tirage abonné du magazine Programmez!

Abonnement : Service Abonnements PROGRAMMEZ, 4 Rue de Mouchy, 60438 Noailles Cedex. - Tél. : 01 55 56 70 55 - 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



Sur abonnement ou en kiosque

Le magazine des pros de l'IT

Mais aussi sur le web



Ou encore sur votre tablette

OPÉRATION POUR 1 EURO DE PLUS

Pour bénéficier de cette offre exceptionnelle, il suffit de commander WINDEV Mobile 22 (ou WINDEV 22, ou WEBDEV 22) chez PC SOFT au tarif catalogue avant le 30 juin 2017. Pour 1 Euro de plus, vous recevez 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 pour consulter la liste des prix des matériels. Tarifs modifiables sans préavis.



CHOISISSEZ:

- Galaxy S8
ou
- Galaxy S8+
ou
- Smart TV 4K 140 cm
ou
- Tablette Galaxy Tab S3+clavier
ou
- 2xTablettes Galaxy Tab S2 9,7"

(Détails et autres matériels sur www.pcsoft.fr)

Atelier de
Génie Logiciel
Professionnel

JUSQU'AU 30 JUIN
COMMANDÉZ
WINDEV MOBILE 22

OU WEBDEV 22 OU WINDEV 22

ET RECEVEZ
LE NOUVEAU

Galaxy S8 | S8+

Choix de
la couleur
sur le site



DAS Galaxy S8 = 0.315 W/kg – DAS Galaxy S8+ = 0.260 W/kg. Le DAS (débit d'absorption spécifique des appareils mobiles) quantifie le niveau d'exposition maximal de l'utilisateur aux ondes électromagnétiques. La réglementation française impose que le DAS ne dépasse pas 2W/kg. L'utilisation d'un kit mains libres est recommandée.

Tél Paris: 01 48 01 48 88
Tél Montpellier: 04 67 032 032



WWW.PCSOFT.FR