

4. Gestion des données urbaines dans les nuages informatiques

- Brève histoire des nuages informatiques
- Modèles de service et de déploiement
- Technologie clé : la virtualisation
- IaaS : les points de vue utilisateur et fournisseur
- **PaaS : programmation et déploiement des applications**
- Stockage de données
- Traitement de données
- Traitement de flux de données

Christine Morin VILLES INTELLIGENTES : DÉFIS TECHNOLOGIQUES ET SOCIÉTAUX

Dans cette séquence, nous étudions la couche intermédiaire de la pyramide des services du cloud.

Un service de plateforme, ce qu'on appelle Plateformes as a Service, permet de programmer les applications pour le cloud, et permet de les déployer au-dessus d'un cloud IaaS.

C'est donc une couche de services essentielle pour les développeurs d'applications dans le contexte des villes intelligentes.

Les centres de données, comme ceux d'Amazon, hébergent différentes classes d'applications.

Applications typiques dans le cloud

- Hébergement d'applications web
- Systèmes de stockage et de backup
- Commerce électronique
- Calcul haute performance
- Moteur de recherche
- Hébergement de serveurs applicatifs
 - Applications de bureau, d'entreprise, pour smartphone et IoT

Grandes familles d'applications

2

Parmi les applications typiques du cloud, on peut citer :

- les applications web,
- les applications de sauvegarde de données et de back-up,
- les applications de commerce électronique,
- les applications à haute performance,
- les moteurs de recherche
- différents serveurs applicatifs pour différents types d'applications, comme les applications de bureautique, les systèmes d'information en entreprise, les back-end, des applications mobiles, etc. Donc pour les objets connectés.

On voit donc qu'il se distingue plusieurs familles d'applications. Et il se trouve que les applications de chaque grande famille ont des besoins similaires.

IaaS : Fonctionnalités de base



Flexibilité mais complexité

3

Comme on l'a vu dans la séquence précédente, les clouds d'infrastructures offrent des fonctionnalités de base qui permettent de déployer les applications dans le cloud.

L'intérêt de se placer au niveau IaaS est la **flexibilité** offerte.

Mais a contrario, il en résulte une très grande **complexité** pour le programmeur et l'administrateur d'application.

Donc l'IaaS, comme j'ai déjà dit est en quelque sorte le langage assembleur du cloud.

PaaS : services de haut niveau

- Chaque plate-forme cible une famille d'applications spécifique
 - Environnement de déploiement facile à utiliser
 - Garantie de haut niveau pour les applications

Faciliter la programmation et le déploiement
d'applications dans le cloud

4

Une plateforme offre des services de haut niveau, afin de **faciliter la programmation et le déploiement des applications dans le cloud**.

Chaque plateforme cible une famille d'applications spécifique, et offre pour cette famille un **environnement de déploiement facile à utiliser** et des **garanties de haut niveau pour les applications**, comme par exemple le temps moyen de service d'une requête pour un serveur web.

PaaS : des solutions commerciales



5

Il y a pléthore de services de type plateforme sur le marché, solutions commerciales et solutions en open source.

De nombreux travaux de recherche dans le domaine des clouds sont centrés sur ce modèle de services.

Ce transparent donne un aperçu de quelques solutions commerciales. Bien entendu, les grands opérateurs de clouds comme Amazon, Google et Windows offrent des solutions de type Platform as a Service. De nombreux autres acteurs existent, qui offrent des plateformes qui peuvent déployer des applications au-dessus de clouds IaaS, qu'ils soient publics et/ou privés et recourent par exemple des poids des serveurs web dans un cloud Amazon.

PaaS : des solutions open source



6

Les solutions développées en open source ne manquent pas non plus. On en voit quelques-unes ici, dont la plateforme ConPaaS, qui a été développée dans le cadre de projets européens, et que nous allons prendre comme exemple pour illustrer notre propos ultérieurement dans la séquence.

Différentiation selon les familles d'applications ciblées

- Applications web



- HPC (sac de tâches, batch), big data (MapReduce), web, mobile



7

Donc il est intéressant de comprendre comment se différencient les différentes plateformes. Un premier critère est la famille d'application considérée :

- De très nombreuses plateformes ciblent le déploiement de **services web** dans le cloud.
- D'autres plateformes visent d'autres types d'applications, comme par exemple les **applications scientifiques**, qui sont exécutées en mode batch, telles que les applications de type sac de tâches ou des applications Big Data, comme les applications MapReduce, que nous étudierons dans une autre séquence.
- D'autres encore facilitent le déploiement des back end, d'applications mobiles et qui trouveront donc leur utilisation dans le contexte des villes intelligentes.

Différentiation selon les langages supportés

- Principaux langages
 - **Java, PHP, Python, Ruby**
- Autres langages répandus
 - .NET, Go, Node.js, Perl
- Autres langages supportés par certaines plates-formes
 - R, C++, Clojure, Scala, Javascript



8

Le langage de programmation est un autre aspect qui caractérise les plateformes.

Les principaux langages supportés sont Java, PHP, Python et Ruby.

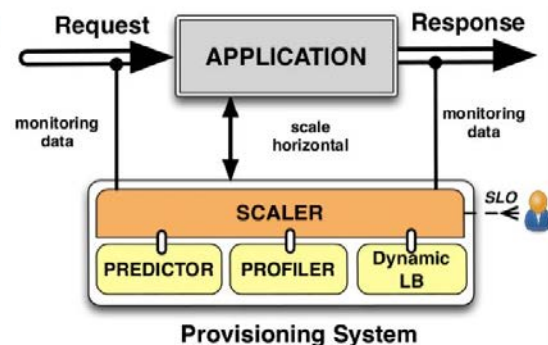
D'autres langages, qu'ils soient classiques comme JavaScript ou Perl ou .NET, sont aussi présents dans certaines plateformes.

Cela peut être également des langages plus exotiques, comme Clojure, par exemple.

Différentiation selon les services offerts

- Construction et hébergement d'application
- Stockage de données
- Redimensionnement
- **Auto-dimensionnement**
- Equilibrage de charge
- Tolérance aux fautes

[FERNANDEZ14]



9

Les plateformes se différencient également par les services offerts. Ils peuvent permettre / offrir :

- la construction et l'hébergement d'applications,
- le stockage de données,
- des fonctionnalités de redimensionnement de l'application en cours d'exécution. Cela peut être fait par exemple de manière automatique.
- des services **d'équilibrage de charges** ou de **tolérance aux fautes**.

Différentiation selon les ressources utilisées

- Ressources propriétaires



- Ressources d'un cloud public



- Ressources d'un ou plusieurs clouds



10

Certaines plateformes sont liées à un cloud propriétaire, comme c'est le cas par exemple pour la plateforme Windows Azure, qui utilise les ressources du cloud de Microsoft et pour Amazon Web Services, qui utilise les ressources du cloud Amazon.

En revanche, d'autres plateformes comme Heroku utilisent les ressources d'un cloud public. Ici, c'est que le cloud public Amazon EC2.

Enfin, d'autres plateformes peuvent utiliser les ressources d'un ou plusieurs clouds. C'est le cas par exemple des solutions open source, Cloud Foundry ou ConPaaS.

Autres critères de différenciation

- Type d'architecture
 - **Machines virtuelles dédiées à une application**
 - Containers dédiés à une application
 - Cluster Hadoop
- Modèle de facturation
 - **Facturation selon la consommation**
 - Abonnement
 - **Combinaison des deux modèles**
- Termes du contrat de service
 - **Disponibilité**
 - Temps de réponse
 - Conditions restrictives pour les pénalités en cas de non respect du contrat



11

On peut citer plusieurs autres critères qui permettent de différencier les offres de type Paas.

Donc certaines plateformes vont gérer des **machines virtuelles dédiées à une application**. C'est actuellement la solution la plus répandue, tandis que d'autres vont gérer des **containers dédiés à une application**. Donc l'usage des contenaires tend à se développer. D'autres plateformes gèrent des ressources qui ne sont pas dédiées à une application. Par exemple, ça peut être un cluster Hadoop qui va du coup être partagé entre plusieurs applications, qui vont être par exemple des applications Map Reduce déployées au-dessus d'un **cluster Hadoop**.

Le modèle de facturation est également un facteur différenciant. En général, donc il y a une **facturation selon la consommation**. Dans certains cas, il peut y avoir en outre un **abonnement**. Ou ça peut être donc une **combinaison des 2 modèles**.

Les termes du contrat de service sont également un facteur différenciant pour les différentes plateformes.

Quelques plateformes proposent des termes liés à la **disponibilité**, comme c'est le cas pour les plateformes dont le logo apparaît dans l'encadré, sur le transparent. Bien souvent, ce sont des termes qui concernent le **temps de réponse** pour les requêtes, notamment pour les serveurs web.

Et ce qu'il faut savoir, c'est qu'il y a **souvent des conditions restrictives pour les pénalités en cas de non-respect du contrat**. Le client doit apporter la preuve de non-respect de la qualité de service et faire sa déclaration dans des délais courts, et reçoit en échange son dédommagement sous forme de crédit d'utilisation de la plateforme.

Un exemple de PaaS open source : ConPaaS

- Applications cibles
 - Applications web
 - Calcul à haute performance (MapReduce, sacs de tâches)
 - Services de stockage (bases de données, système de fichiers)
- Service indépendant du cloud IaaS sous-jacent



Simplicité - Flexibilité – Elasticité - Extensibilité

12

Pour terminer, je vais présenter brièvement la solution open source ConPaaS.

ConPaaS est une plateforme qui cible principalement 3 types de services :

- les applications web,
- les applications de calcul à haute performance, comme par exemple Map Reduce et les sacs de tâches,
- les services de stockage. Différents types de base de données et un système de fichiers extensible sont offerts par la plateforme ConPaaS.

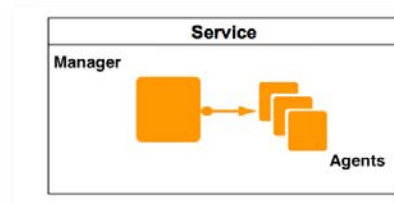
Les services de ConPaaS sont indépendants de la plateforme sous-jacente. Donc ConPaaS tourne sur des clouds IaaS variés, comme Open stack, open nebula, Amazon EC2.

ConPaaS a été conçu pour être **simple d'utilisation** et propose une interface graphique extrêmement simple, pour déployer tous les services proposés. Une interface en ligne de commande est également disponible pour bénéficier de fonctionnalités avancées.

ConPaaS offre beaucoup de **flexibilité**, en permettant de construire des applications comme une composition de services. Donc il gère l'**élasticité** des services déployés grâce à une interface très simple, qu'elle soit en ligne de commande, ou une interface graphique.

Enfin, ConPaaS est une plateforme **extensible**. Elle peut être enrichie au fil du temps par de nouveaux services.

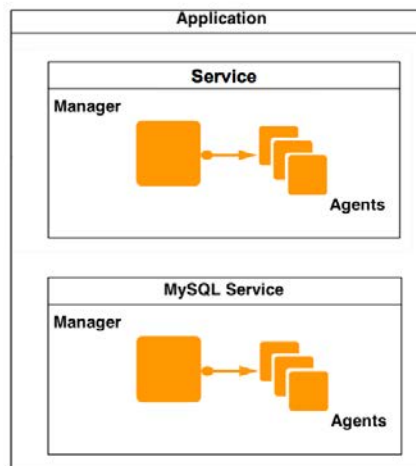
Un service ConPaaS



[PIERRE15]

13

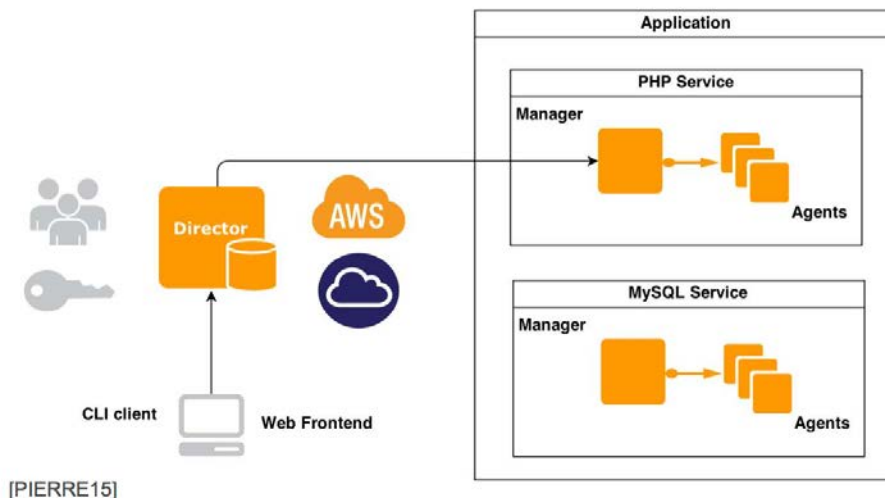
Une application ConPaaS



[PIERRE15]

14

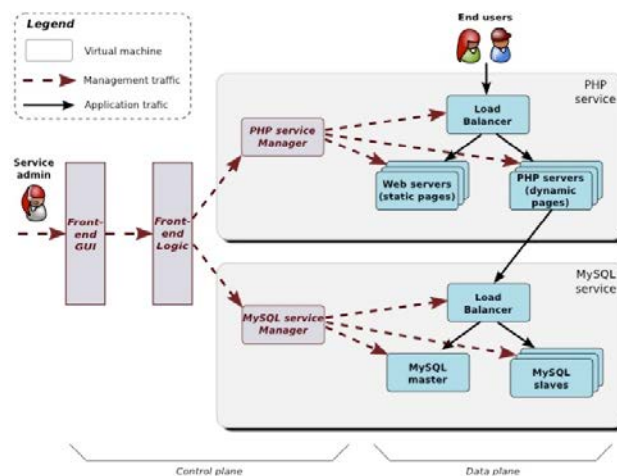
Architecture ConPaaS



[PIERRE15]

15

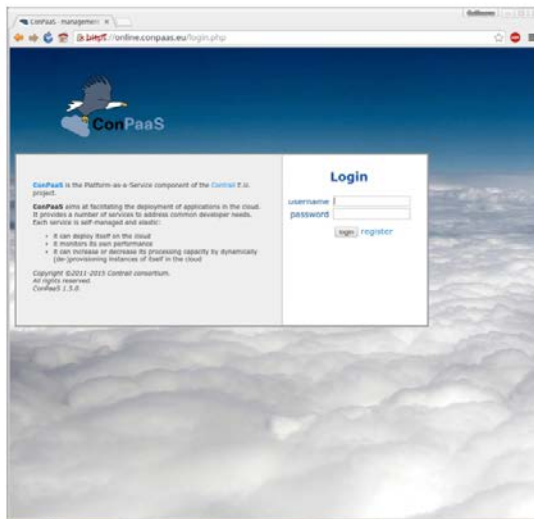
Un exemple d'application web composée de 2 services ConPaaS



[PIERRE12]

16

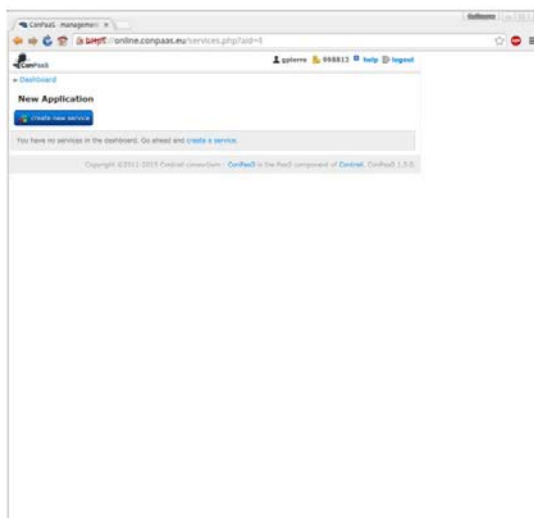
Interface web ConPaaS pour la gestion d'applications



- **Login**
- Création d'un service
- Gestion d'un service
- Arrêt d'un service
- Suppression d'un service

17

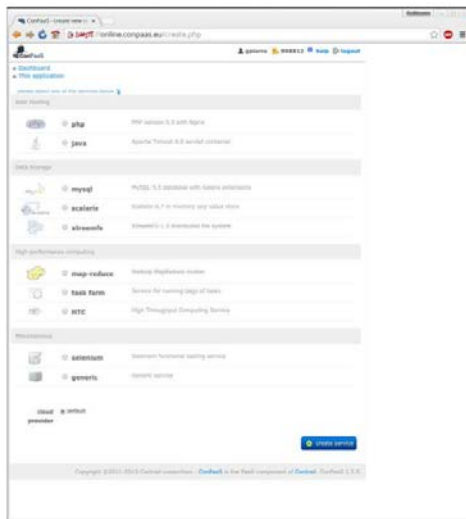
Interface web ConPaaS pour la gestion d'applications



- Login
- **Création d'un service**
- Gestion d'un service
- Arrêt d'un service
- Suppression d'un service

18

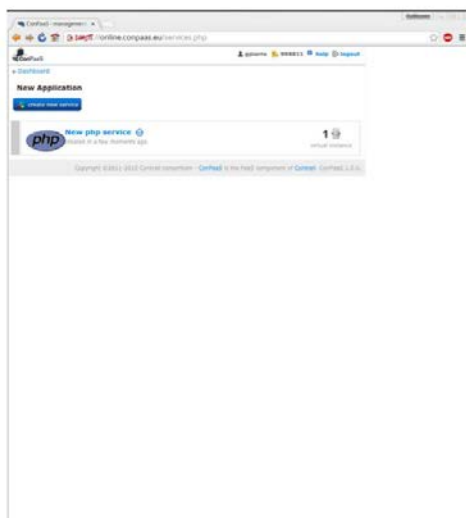
Interface web ConPaaS pour la gestion d'applications



19

- Login
- **Création d'un service**
- Gestion d'un service
- Arrêt d'un service
- Suppression d'un service

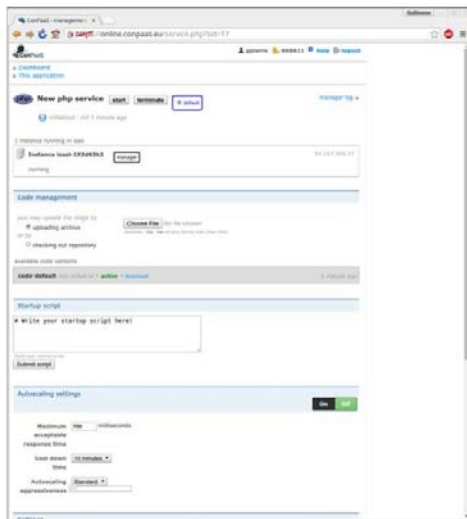
Interface web ConPaaS pour la gestion d'applications



20

- Login
- **Création d'un service**
- Gestion d'un service
- Arrêt d'un service
- Suppression d'un service

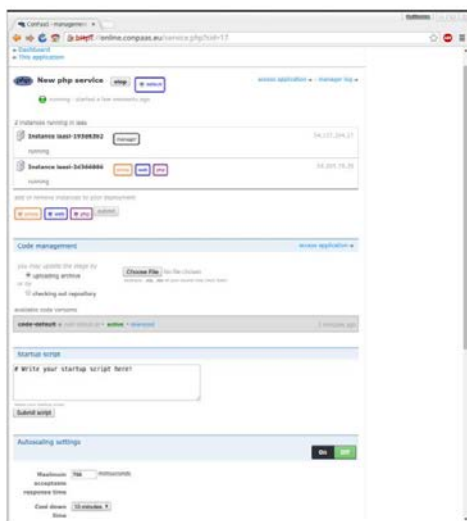
Interface web ConPaaS pour la gestion d'applications



21

- Login
- **Création d'un service**
- Gestion d'un service
- Arrêt d'un service
- Suppression d'un service

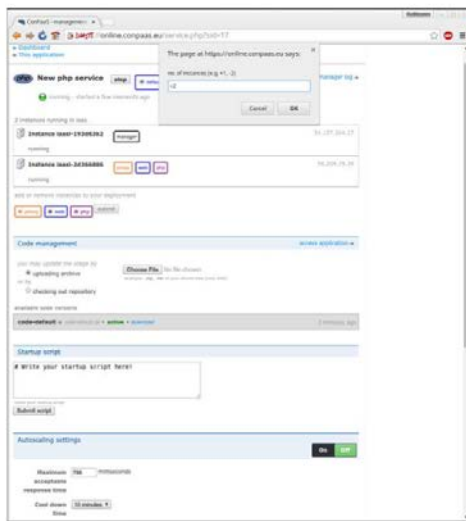
Interface web ConPaaS pour la gestion d'applications



22

- Login
- **Création d'un service**
- Gestion d'un service
- Arrêt d'un service
- Suppression d'un service

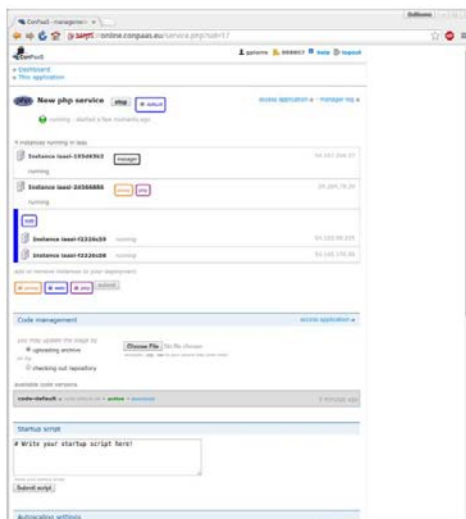
Interface web ConPaaS pour la gestion d'applications



23

- Login
- Création d'un service
- **Gestion d'un service**
- Arrêt d'un service
- Suppression d'un service

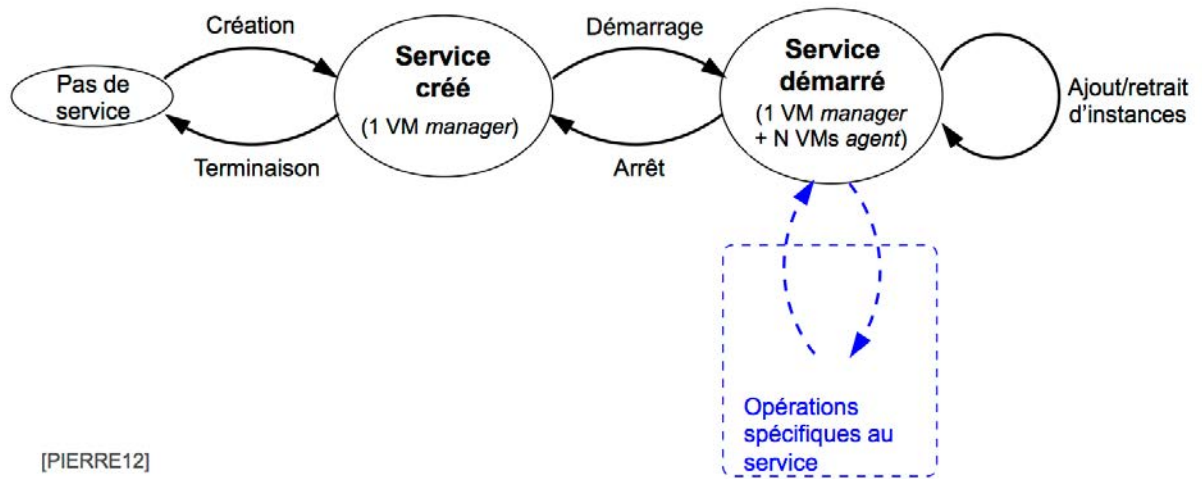
Interface web ConPaaS pour la gestion d'applications



24

- Login
- Création d'un service
- **Gestion d'un service**
- Arrêt d'un service
- Suppression d'un service

Cycle de vie d'un service



[PIERRE12]

Donc les programmeurs et administrateurs d'applications pour la ville intelligente devront faire un choix éclairé du type de plateforme qu'ils utiliseront en fonction des besoins de leurs applications.

Références bibliographiques

- [FERNANDEZ14] H. Fernandez, G. Pierre, T. Kielmann, *Autoscaling Web applications in heterogeneous cloud infrastructures*, IEEE International Conference on Cloud Engineering (IC2E), 2014, pages 195-204.
- [PIERRE15] G. Pierre, Heterogenous Cloud Computing for Research- Managing Cloud Services with ConPaas, <https://harnesscloud.github.io/2015-07-15-feltham/conpaas/slides.pdf>
- [PIERRE12] G. Pierre, *ConPaaS: an integrated runtime environment for elastic cloud applications*, présentation du 24/07/12 dans le cadre du projet européen FP7 Contrail - <http://www.conpaas.eu/wp-content/uploads/2012/05/conpaas-slides.pdf>

Illustrations & photos : crédits

p. 3 : © Guillaume Pierre, Techniques de développement logiciel pour le cloud. Cours de Master 2, Université de Rennes 1, novembre 2014.

p. 9 : © Thilo Kielmann, projet européen Contrail

p. 13-15 : © Guillaume Pierre, projet européen Contrail

p. 16 : © Guillaume Pierre

p. 25 : © Guillaume Pierre