

# Des routes et des ponts (6) – créer une infrastructure numérique

*Dans ce nouveau chapitre de l'ouvrage de [Nadia Eghbal](#) [Des routes et des ponts](#) que le groupe Framalang vous traduit semaine après semaine (si vous avez raté [les épisodes précédents](#)), l'autrice<sup>(1)</sup> établit cette fois-ci une comparaison éclairante entre l'infrastructure physique dont nous dépendons sans toujours en avoir conscience et l'infrastructure numérique dont la conception et le processus sont bien différents.*

## Qu'est-ce qu'une infrastructure numérique, et comment est-elle construite ?

*Traduction Framalang : Diane, Penguin, Asta, teromene, Julien / Sphinx, Luc, salade, AFS, xi, goofy*

Dans un chapitre précédent de ce rapport, nous avons comparé la création d'un logiciel à la construction d'un bâtiment. Ces logiciels publiquement disponibles contribuent à former notre infrastructure numérique. Pour comprendre ce concept, regardons comment les infrastructures physiques fonctionnent.

Tout le monde dépend d'un certain nombre d'infrastructures physiques qui facilitent notre vie quotidienne. Allumer les lumières, aller au travail, faire la vaisselle : nous ne pensons pas souvent à l'endroit d'où viennent notre eau ou notre électricité, mais heureusement que nous pouvons compter sur les infrastructures physiques. Les partenaires publics et privés travaillent de concert pour construire et maintenir nos

infrastructures de transport, d'adduction des eaux propres et usées, nos réseaux d'électricité et de communication.

De même, même si nous ne pensons pas souvent aux applications et aux logiciels que nous utilisons quotidiennement, tous utilisent du code libre et public pour fonctionner. Ensemble, dans une société où le numérique occupe une place croissante, ces projets *open source* forment notre infrastructure numérique. Toutefois, il existe plusieurs différences majeures entre les infrastructures physiques et numériques, qui affectent la manière dont ces dernières sont construites et maintenues. Il existe en particulier des différences de coût, de maintenance, et de gouvernance.

## **Les infrastructures numériques sont plus rapides et moins chères à construire.**

C'est connu, construire des infrastructures matérielles coûte très cher. Les projets sont physiquement de grande envergure et peuvent prendre des mois ou des années à réaliser.

Le gouvernement fédéral des États-Unis a dépensé 96 milliards de dollars en projets d'infrastructure en 2014 et les gouvernements des différents états ont dépensé au total 320 milliards de dollars cette même année. Un peu moins de la moitié de ces dépenses (43 pour cent) a été affectée à de nouvelles constructions ; le reste a été dépensé dans des opérations de [maintenance de l'infrastructure existante](#).

Proposer puis financer des projets de nouvelles infrastructures physiques peut être un processus politique très long. Le financement des infrastructures de transport a été un sujet délicat aux États-Unis d'Amérique au cours de la dernière décennie lorsque le gouvernement fédéral a été confronté à un [manque de 16 milliards](#) de dollars pour les financer.

Le Congrès a récemment voté la première loi pluriannuelle de

financement des transports depuis 10 ans, affectant 305 milliards de dollars aux autoroutes et autres voies rapides après des années d'oppositions politiques empêchant la budgétisation des infrastructures [au-delà de deux années](#).

Même après qu'un nouveau projet d'infrastructure a été validé et a reçu les fonds nécessaires, il faut souvent des années pour le terminer, à cause des incertitudes et des obstacles imprévus auxquels il faut faire face.

Dans le cas du projet Artère Centrale/Tunnel à Boston, dans le Massachusetts, connu aussi sous le nom de [Big Dig](#), neuf ans se sont écoulés entre la planification et le début des travaux. Son coût prévu était de 2,8 milliards de dollars, et il devait être achevé en 1998. Finalement, le projet a coûté 14,6 milliards de dollars et n'a pas été terminé avant 2007, ce qui en fait le projet d'autoroute le plus cher des États-Unis.

En revanche, les infrastructures numériques ne souffrent pas des coûts associés à la construction des infrastructures physiques comme le [zonage](#) ou l'achat de matériels. Il est donc plus facile pour tout le monde de proposer une nouvelle idée et de l'appliquer en un temps très court. MySQL, le [second système](#) de gestion de base de données le plus utilisé dans le monde et partie intégrante d'une collection d'outils indispensables qui aidèrent à lancer le premier boum technologique, fut lancé par ses créateurs, Michael Widenius & David Axmark, en mai 1995. [Ils mirent moins de deux années à le développer](#).

Il a fallu à [Ruby](#), un langage de programmation, moins de trois ans entre sa conception initiale en février 1993 et sa publication en décembre 1995. Son auteur, l'informaticien Yukihiro Matsumoto, a décidé de créer le langage après une discussion avec ses collègues.

# Les infrastructures numériques se renouvellent fréquemment

Comme l'infrastructure numérique est peu coûteuse à mettre en place, les barrières à l'entrée sont plus basses et les outils de développement changent plus fréquemment.

L'infrastructure physique est construite pour durer, c'est pourquoi ces projets mettent si longtemps à être planifiés, financés et construits. [Le métro de Londres](#), le système de transport en commun rapide de la ville, fut construit en 1863 ; les tunnels creusés à l'époque sont encore utilisés aujourd'hui.

Le pont de Brooklyn, qui relie les arrondissements de Brooklyn et de Manhattan à New York City, fut achevé en 1883 et n'a pas subi de rénovations majeures avant 2010, plus de cent ans plus tard. L'infrastructure numérique nécessite non seulement une maintenance et un entretien fréquents pour être compatible avec d'autres logiciels, mais son utilisation et son adoption changent également fréquemment. Un pont construit au milieu de New York City aura un usage garanti et logique, en proportion de la hausse ou la diminution de la population. Mais un langage de programmation ou un framework peut être extrêmement populaire durant plusieurs années, puis tomber en désuétude lorsque apparaît quelque chose de plus rapide, plus efficace, ou simplement plus à la mode.

Par exemple, le graphique ci-dessous montre l'activité des développeurs de code source selon plusieurs langages différents. Le langage C, l'un des langages les plus fondamentaux et les plus utilisés, a vu sa part de marché diminuer alors que de nouveaux langages apparaissaient. Python et JavaScript, deux langages très populaires en ce moment, ont vu leur utilisation augmenter régulièrement avec le temps. Go, développé en 2007, a connu plus d'activité dans les dernières années.

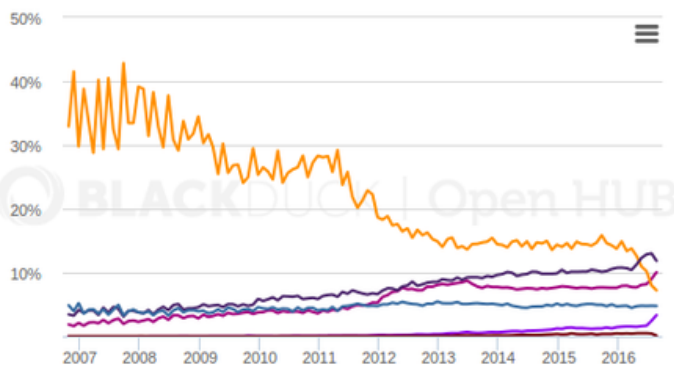
## Compare Languages

Monthly Commits Monthly Contributors Monthly Lines of Code Changed Monthly Projects

### Monthly Commits (Percent of Total)

The lines show the count of monthly commits made by source code developers. Commits including multiple languages are counted once for each language.

[More](#)



<input type="checkbox"/>	C
<input type="checkbox"/>	Go
<input type="checkbox"/>	JavaScript
<input type="checkbox"/>	PHP
<input type="checkbox"/>	Python
<input type="checkbox"/>	Rust
<input type="checkbox"/>	[None]

Copie d'écran du site <https://www.openhub.net/languages/compare> prise le 23/10/2016.

Tim Hwang, dirigeant du Bay Area Infrastructure Observatory, qui organise des visites de groupe sur des sites d'infrastructures physiques, faisait remarquer la différence dans [une interview de 2015](#) donnée au *California Sunday Magazine* :

*« Beaucoup de membres de notre groupe travaillent dans la technologie, que ce soit sur le web ou sur des logiciels. En conséquence, ils travaillent sur des choses qui ne durent pas longtemps. Leur approche c'est 'On a juste bidouillé ça, et on l'a mis en ligne' ou : 'On l'a simplement publié, on peut travailler sur les bogues plus tard'. Beaucoup d'infrastructures sont construites pour durer 100 ans. On ne peut pas se permettre d'avoir des bogues. Si on en a, le bâtiment s'écroule. On ne peut pas l'itérer. C'est une façon de concevoir qui échappe à l'expérience quotidienne de nos membres. »*

Cependant, comme l'infrastructure numérique change très fréquemment, les projets plus anciens ont plus de mal à trouver des contributeurs, parce que beaucoup de développeurs préfèrent travailler sur des projets plus récents et plus

excitants. Ce phénomène est parfois référencé comme le « [syndrome de la pie](#) » chez les développeurs, ces derniers étant attirés par les choses « nouvelles et brillantes », et non par les technologies qui fonctionnent le mieux pour eux et pour leurs utilisateurs.

## **Les infrastructures numériques n'ont pas besoin d'autorité organisatrice pour déterminer ce qui doit être construit ou utilisé**

En définitive, la différence la plus flagrante entre une infrastructure physique et une infrastructure numérique, et c'est aussi un des défis majeurs pour sa durabilité, c'est qu'il n'existe aucune instance décisionnelle pour déterminer ce qui doit être créé et utilisé dans l'infrastructure numérique. Les transports, les réseaux d'adduction des eaux propres et usées sont généralement gérés et possédés par des collectivités, qu'elles soient fédérales, régionales ou locales. Les réseaux électriques et de communication sont plutôt gérés par des entreprises privées. Dans les deux cas, les infrastructures sont créées avec une participation croisée des acteurs publics et privés, que ce soit par le budget fédéral, par les entreprises privées ou les contributions payées par les usagers.

Dans un État stable et développé, nous nous demandons rarement comment une route est construite ou un bâtiment électrifié. Même pour des projets financés ou propriétés du privé, le gouvernement fédéral a un intérêt direct à ce que les infrastructures physiques soient construites et maintenues.

De leur côté, les projets d'infrastructures numériques sont conçus et construits en partant du bas. Cela ressemble à un groupe de citoyens qui se rassemblent et décident de construire un pont ou de créer leur propre système de recyclage des eaux usées. Il n'y a pas d'organe officiel de

contrôle auquel il faut demander l'autorisation pour créer une nouvelle infrastructure numérique.

Internet lui-même possède deux organes de contrôle qui aident à définir des standards : l'IETF (Internet Engineering Task Force) et le W3C (World Wide Web Consortium). L'IETF aide à développer et définit des standards recommandés sur la façon dont les informations sont transmises sur Internet. Par exemple, ils sont la raison pour laquelle les URL commencent par "HTTP". Ils sont aussi la raison pour laquelle nous avons des adresses IP – des identifiants uniques assignés à votre ordinateur lorsqu'il se connecte à un réseau. À l'origine, en 1986, il s'agissait d'un groupe de travail au sein du gouvernement des USA mais l'IETF est devenue une organisation internationale indépendante en 1993.

L'IETF elle-même fonctionne grâce à des bénévoles et il n'y a pas d'exigences pour adhérer : n'importe qui peut rejoindre l'organisation en se désignant comme membre. Le W3C (World Wide Web Consortium) aide à créer des standards pour le World Wide Web. Ce consortium a été fondé en 1994 par Tim Berners-Lee. Le W3C a tendance à se concentrer exclusivement sur les pages web et les documents (il est, par exemple, à l'origine de l'utilisation du HTML pour le formatage basique des pages web). Il maintient les standards autour du langage de balisage HTML et du langage de formatage de feuilles de style CSS, deux des composants de base de n'importe quelle page web. L'adhésion au W3C, légèrement plus formalisée, nécessite une inscription payante. Ses membres vont des entreprises aux étudiants en passant par des particuliers.

L'IETF et le W3C aident à gérer les standards utilisés par les pièces les plus fondamentales d'Internet, mais la couche du dessus – les choix concernant le langage utilisé pour créer le logiciel, quels frameworks utiliser pour les créer, ainsi que les bibliothèques à utiliser – sont entièrement auto-gérés dans le domaine public (bien entendu, de nombreux projets de logiciels propriétaires, particulièrement ceux qui sont régis

par de très nombreuses normes, tels que l'aéronautique ou la santé peuvent avoir des exigences concernant les outils utilisés. Ils peuvent même développer des outils propriétaires pour leur propre utilisation).

Avec les infrastructures physiques, si le gouvernement construit un nouveau pont entre San Francisco et Oakland, ce pont sera certainement utilisé. De la même façon, lorsque le W3C décide d'un nouveau standard, tel qu'une nouvelle version de HTML, il est formellement publié et annoncé. Par exemple, en 2014, le W3C a annoncé HTML 5, la première révision majeure de HTML depuis 1997, qui a été développé pendant sept ans.

En revanche, lorsqu'un informaticien souhaite créer un nouveau langage de programmation, il ou elle est libre de le publier et ce langage peut ou peut ne pas être adopté. La barre d'adoption est encore plus basse pour les frameworks et bibliothèques : parce qu'ils sont plus faciles à créer, et plus facile pour un utilisateur à apprendre et implémenter, ces outils sont itérés plus fréquemment.

Mais le plus important c'est que personne ne force ni même n'encourage fortement quiconque à utiliser ces projets. Certains projets restent plus théoriques que pratiques, d'autres sont totalement ignorés. Il est difficile de prédire ce qui sera véritablement utilisé avant que les gens ne commencent à l'utiliser.

Les développeurs aiment se servir de l'utilité comme indicateur de l'adoption ou non d'un projet. Les nouveaux projets doivent améliorer un projet existant, ou résoudre un problème chronique pour être considérés comme utiles et dignes d'être adoptés. Si vous demandez aux développeurs pourquoi leur projet est devenu si populaire, beaucoup hausseront les épaules et répondront : « C'était la meilleure chose disponible ». Contrairement aux *startups* technologiques, les nouveaux projets d'infrastructure numérique reposent sur les effets de réseau pour être adoptés par le plus grand nombre.



L'existence d'un groupe noyau de développeurs motivés par le projet, ou d'une entreprise de logiciels qui l'utilise, contribue à la diffusion du projet. Un nom facilement mémorisable, une bonne promotion, ou un beau site Internet peuvent ajouter au facteur « nouveauté » du projet. La réputation d'un développeur dans sa communauté est aussi un facteur déterminant dans la diffusion d'un projet.

Mais en fin de compte, une nouvelle infrastructure numérique peut venir d'à peu près n'importe où, ce qui veut dire que chaque projet est géré et maintenu d'une façon qui lui est propre.

(1) pourquoi « autrice » ? Il semble que ce mot soit [légitime au regard de l'histoire de la langue](#). Voyez aussi [une étude plus complète](#).