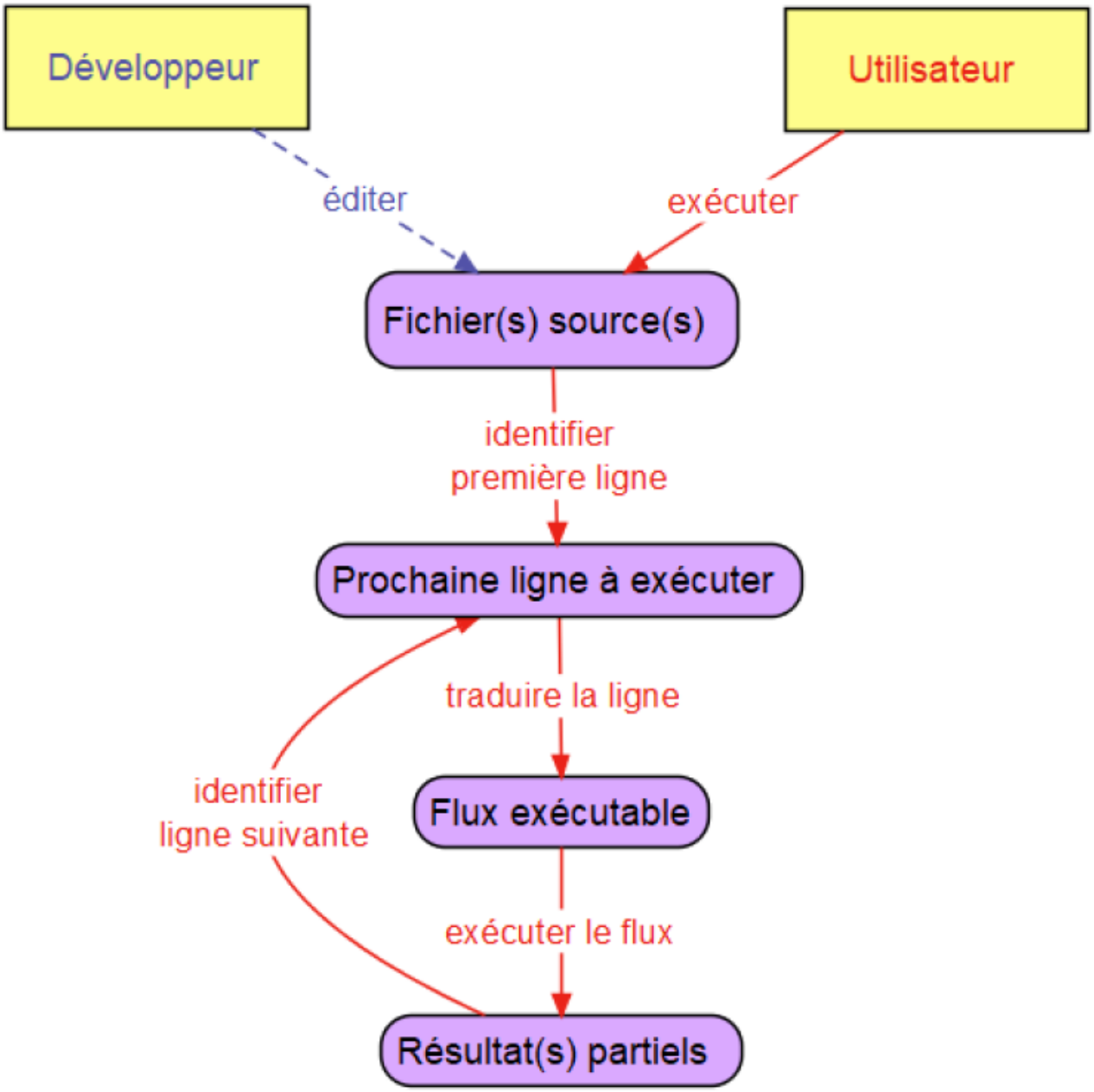
Modèles de traduction-exécution de programmes

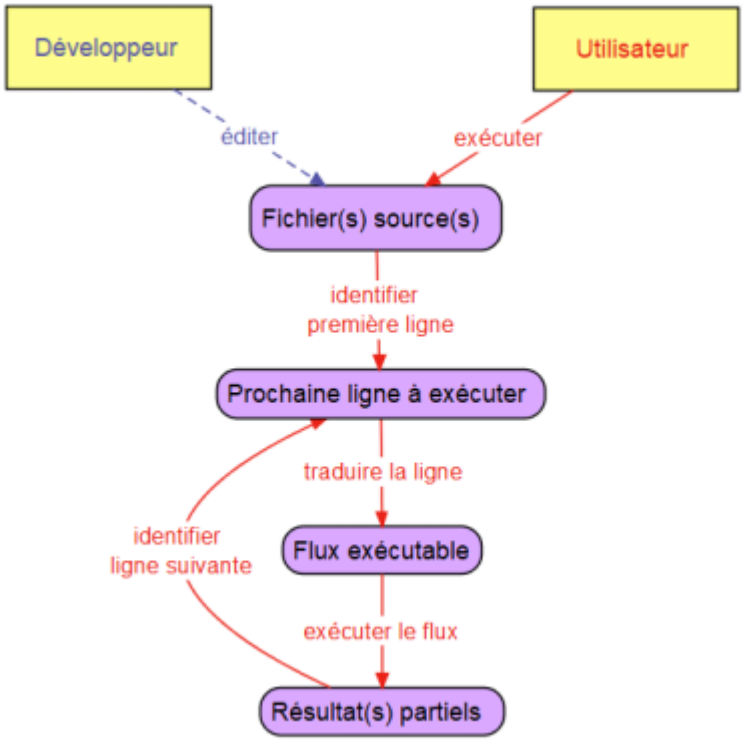
Langage interprété (PHP, JavaScript)



- Une traduction pour chaque exécution ;
- Un programme n'est pas nécessairement traduit (donc testé) en totalité lors d'une exécution ;
- Deux exécutions distinctes du même programme peuvent ne pas traduire les mêmes lignes et, par exemple, témoigner d'anomalies différentes ;
- Il n'y a pas d'analyse globale du code (cohérence + couverture) ;

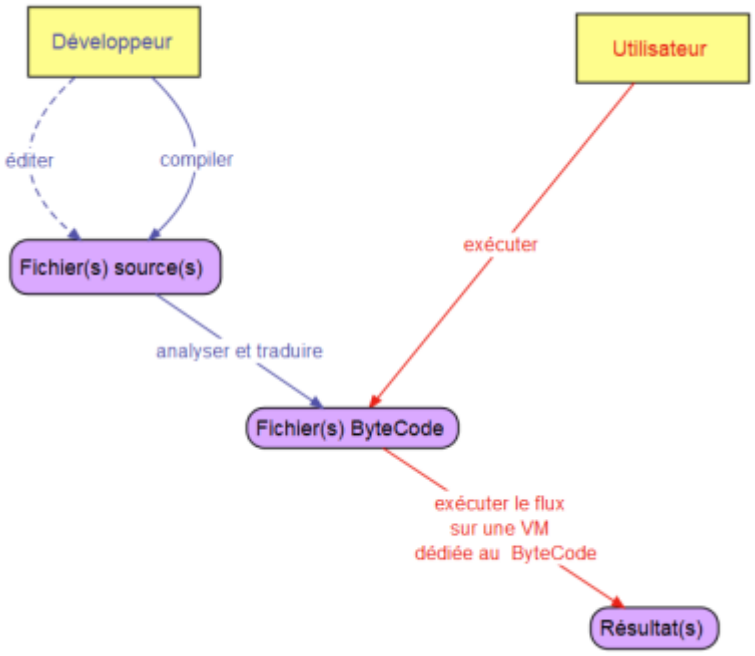
- Il faut un traducteur sur la machine de l'utilisateur ;
- Le processus de mise au point est délicat car on n'est jamais certain d'avoir tout testé ;
- Le fichier source est accessible à l'utilisateur (cybersécurité) ;

Langage compilé (C, C++)



- Analyse globale du code (cohérence + couverture) systématique ;
- Une seule traduction pour n exécutions ;
- Un programme est nécessairement traduit en totalité avant toute exécution ;
- Le processus de mise au point est surprenant mais efficace : une erreur de traduction peut en cacher plusieurs dizaines d'autres ;
- Le fichier exécutable est spécifique à une plateforme (Linux, Windows, etc.), il en existe donc plusieurs ;
- Il faut une compilation distincte pour chaque plateforme cible ;
- Le fichier source n'est pas accessible à l'utilisateur (cybersécurité) ;

Langage hybride (Java, C#)



- Analyse globale du code (cohérence + couverture) systématique ;
- Une seule traduction en ByteCode pour n exécutions ;
- Un programme est nécessairement traduit en totalité avant toute exécution ;
- Le processus de mise au point est surprenant mais efficace : une erreur de traduction peut en cacher plusieurs dizaines d'autres ;
- Le fichier exécutable est spécifique à une VMet son ByteCode ;
- Il faut une seule compilation pour toutes les plateforme cibles car la VM est unique ;
- Le fichier source n'est pas accessible à l'utilisateur (cybersécurité) ;
- Il existe une dépendance à la VM(présence et tenue à jour) qui peut être contraignante ;
- La performance de la VM conditionne l'intérêt d'ensemble ;

**Java **

- Les fichiers ByteCode sont des fichiers à extension “.class” ;
- Le déploiement s'effectue en embarquant tous les fichiers nécessaires dans une archive compressée dont la structure est connue. Le fichier compressé a une extension “.jar”, pour Java ARchive ;
- La VM Java est communément nommée JVM, pour Java Virtual Machine ;

**C# **

- Les fichiers ByteCode sont des fichiers à extension “.” ;
- Le déploiement s'effectue en ;
- La VM C# est commune à plusieurs langages et porte le nom du framework sous-jacent nommé .Net ou DotNet.

From:
https://wiki.siochaptalqper.fr/ - **Wiki SIO Chaptal**

Permanent link:
https://wiki.siochaptalqper.fr/doku.php?id=bloc2:prog:gen:modelestraduction&rev=1663667403

Last update: **2022/09/20 11:50**