



Coder Proprement
Intégration Continue

Intégration continue

« L'intégration continue est un ensemble de pratiques utilisées en génie logiciel consistant à vérifier à chaque modification de code source que le résultat des modifications ne produit pas de régression dans l'application développée. »

Wikipedia



Au menu

- **Théorie**
 - Méthodologie à la « R.A.C.H.E »
 - Intégration continue
 - Applications théoriques
- **Pratique**
 - Outils du marché
 - Intégration continue chez Amiltone
- **Démonstrations**
- **Fastlane**
- **Projet**



Théorie



Projet

- Site d'actualités
 - Consultation des actualités
 - Compte utilisateur
 - Ajout de commentaires

ADMINISTRY

Logged in as Admin | Settings | Logout
You have 1 new message!

English | Search | German

Dashboard

Welcome back, Admin

Last login: Monday July 12th, 2010 at 11:32am from 127.0.0.1
No error login attempts.

Invoices: 10

Total Invoices	10	1,322.10 €
Total Paid	9	900.00 €
Total Due	1	422.10 €
Total Overdue	0	0.00 €

Clients This Month

Clients This Month	1
Sales This Month	9
Clients Total	5
Sales Total	9

Sales: 1

Total Paid	9	900.00 €
Sales This Month	9	900.00 €

Client Stats

Active	1/5	20%
Pending	2/5	40%
Suspended	2/5	40%

Reports

Sales this year

Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec

Your Account

Administrator (admin)
Free Account.

Support

Documentation and FAQ

Latest in the Community

Maximize every interaction with a client
Read more
Diversification for Small Business Owners
Read more

Le Monde.fr

Meteo à 10h30 - Paris

Le Monde.fr | le web avec Recherche sur Le Monde.fr Abonnez-vous au journal Recevez les newsletters gratuites PIAGET POLO FORTYFIVE

ACTUALITES PERSPECTIVES PRATIQUE ANNONCES LE DESK LE MOSQUE NEWSLETTERS MULTIMEDIA REFERENCES

International Planète Europe Politique Société Camer Economie Médias Sports Technologies Culture Européennes Cannes Roland-Garros

PIAGET POLO FORTYFIVE

Des débris d'avion retrouvés dans l'océan Atlantique

Rien n'indique pour l'instant que ces débris appartiennent à l'A330 disparu. Si tel était le cas, cela pourrait indiquer que l'avion a essayé de revenir vers son point de départ.

- Vol Rio-Paris : les questions qui se posent
- Comment les pilotes font face aux orages et à la grêle
- "L'orage fait partie du quotidien des pilotes"
- Un "effroyable mystère"

ET AUSSI

Richard Descoings : "Il faut arrêter de rêver au Grand Soir de l'éducation"

Le directeur de Sciences Po Paris a remis au président les conclusions de ces quatre mois de consultation sur le lycée.

VOS RÉACTIONS

Le lien tient aux centaines de milliers de PC sous windows transformés en zombies qui sont les fantassins de ces

Nous y voyons aussi un défi industriel.

11:45

Le Monde.fr

EN DIRECT à 11h33

En France, les chauffeurs routiers étrangers seront payés au smic

Jeux vidéo

« Swatting » : prison ferme pour un canular dangereux

Bac

Bac : l'angoisse du prof à la saisie des notes

Euro 2016

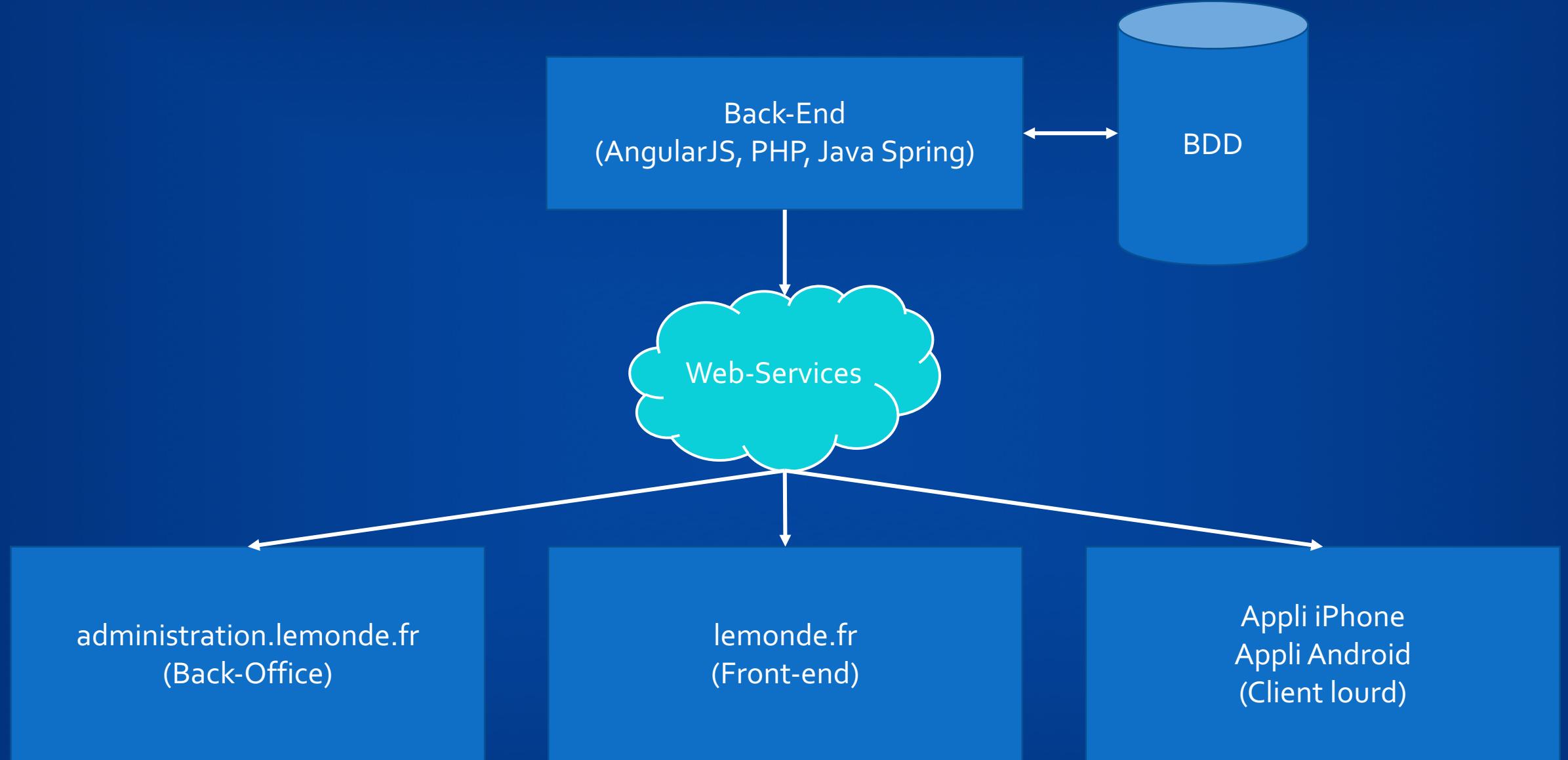
Euro 2016 : pour en finir avec ces interminables prolongations

5

Équipe projet

- 12 développeurs
- 1 chef de projet
- Produisent :
 - 1 back-end avec web-services (NodeJS, PHP, Java Spring)
 - 1 back-office web (Bootstrap, AngularJS)
 - 1 front-end web (Bootstrap, AngularJS)
 - 1 client lourd (Application Android)





Méthodologie projet

- Gestion des sources
 - Git (GitHub, GitLab, Bitbucket)
- Environnements
 - Développement
 - Pré-production
 - Production



Organisation d'une livraison complète

- Par environnement
 - (Si nécessaire) remise à zéro de la BDD ou mise en place d'un jeu de données
 - Compilation & packaging du back-end
 - Déploiement du back-end
 - Compilation & packaging du back-office
 - Déploiement du back-office
 - Compilation & packaging du front-end
 - Déploiement du front-end
 - Compilation des clients lourds
 - Mise à disposition des clients lourds



LA « R.A.C.H.E »



Fonctionnement

- On récupère les sources depuis le dernier commit
- On fait les opérations à la main ou par des scripts
- On exécute les tests localement sur sa machine
- On fait les mises en production à tour de rôle



« Oups le commit »

- Le dernier commit d'un développeur intègre par erreur du code de test qui n'aurait pas du être envoyé sur Git
- Les développeurs qui récupèrent ce commit auront des problèmes
- Il faut diagnostiquer le problème
- Les développements sont immobilisés
- Temps perdu : 2 (développeurs) x 30mn = 1h



« Ça marche chez moi »

- Le développeur a oublié de commit un fichier
- Les tests unitaires fonctionnent sur sa machine en local
- Les développeurs qui récupèrent ce commit ne pourront pas dérouler les tests unitaires
- Il faut trouver la source du problème
- Temps perdu : 3 (développeurs) x 20mn = 1h



« Le script marche pas »

- C'est au tour de Martin de faire la livraison
- Le script de packaging renvoie une erreur
- La documentation n'est en fait pas à jour
- La mise en place de l'environnement local n'a pas été faite correctement
- La livraison est retardée
- Temps perdu : 2 (développeurs) x 20mn = 40mn
- Retard de la livraison : 20mn (un vendredi après-midi...)



« Il faut livrer en production »

- Arrêt des développements
- Il faut bloquer une demi-journée pour une personne
- Le développeur va suivre la procédure pour mettre manuellement en production
- Temps passé: 1 (développeur) x 3h = 3h



« Les accès ont changé »

- Paul doit maintenant déployer la livraison
- Il n'arrive pas à se connecter au FTP
- Les accès à la machine ont été changés
- Personne n'a été prévenu
- Temps perdu : 1 (développeur) x 5mn = 5mn
- Retard de la livraison : 3h (les responsables étaient en réunion)



Conclusion

- ½ à 2 jours de développement (équipe de 12 personnes) par semaine
- Problèmes détectés aux pires moments (quand on doit effectuer l'action)
- De 10mn à 4h de retard pour les livraisons chaque semaine
- Prévenus des dysfonctionnements au dernier moment



INTÉGRATION CONTINUE



Intégration Continue

- Automatisation des tâches répétitives
- Harmonisation des processus de développement internes
- Pourquoi ?
 - Contrôler systématiquement tout changement du code source
 - Aucune valeur ajoutée à ce qu'un humain le fasse
 - Empêcher les inévitables erreurs humaines
 - Centralisation des procédures



Intégration Continue

- Pour faire quoi ?
 - Récupérer le code source
 - L'analyser
 - Le compiler
 - Le tester
 - Le packager
 - Le déployer
 - Jongler entre différents environnements
 - Être alerté en cas de problème

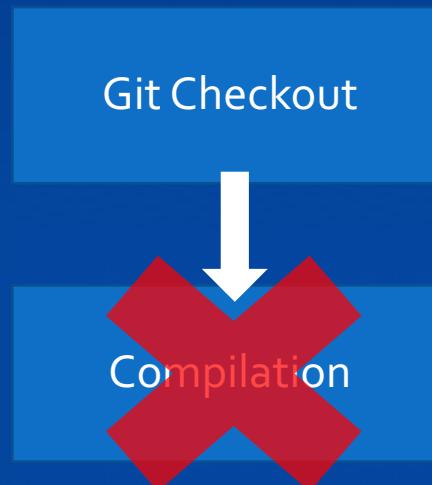


Intégration Continue

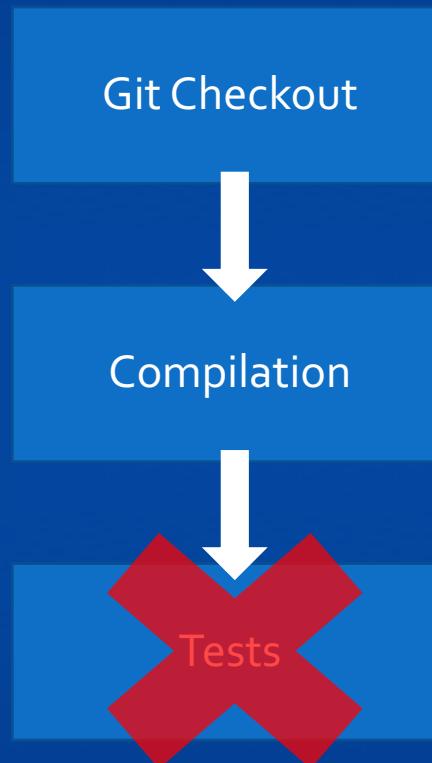
- Quand ?
 - Démarré automatiquement après un Git Push
 - Démarré par un événement dans le flux de travail
 - Démarré manuellement par un développeur
 - Démarré périodiquement



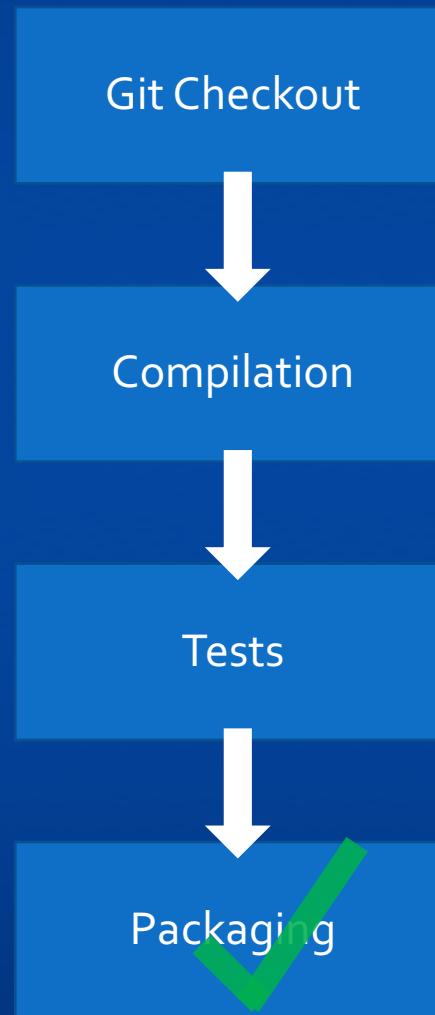
« Oups le commit »



« Ca marche chez moi »



« Le script marche pas »



« Il faut livrer en production »



« Les accès ont changé »



Avantages

- Gagner du **temps**
- Déetecter tout de suite les problèmes
- Mieux diagnostiquer d'où vient le problème
- Être alerté automatiquement d'un dysfonctionnement
- Mise en production sans intervention humaine



APPLICATIONS THÉORIQUES



Harmonisation des processus

- Il faut repenser son cycle de développement
- Il faut découper chaque action comme une sous-tâche à automatiser
- Force la remise en question et la simplicité
- Trouver les chaines d'intégration continue nécessaires pour son projet



Exemples de chaines d'intégration

- **Validation du code soumis**
 - Récupérer le code sur Git
 - Compiler le code
 - Tester le code
- Cas d'exécution
 - Push sur Git
- Alerte
 - En cas d'échec



Exemples de chaines d'intégration

- **Mise à disposition d'une librairie interne**
 - Récupérer le code sur Git
 - Compiler le code
 - Tester le code
 - Envoyer le binaire sur le gestionnaire d'artefacts
- Cas d'exécution
 - Manuellement
- Alerte
 - En cas d'échec



Généralités

- Pour chaque besoin ou problématique, une chaîne d'intégration continue existe
- Attention à bien mutualiser et harmoniser ses actions pour les réutiliser
- Plus de stress pour les mises en production
- Tout se fait en parallèle des développements
 - Gain de temps
 - Gain d'argent



Pratique



OUTILS DU MARCHÉ



Catégories

- Auto-hébergé
 - Gestion de l'infrastructure
 - Maitrise
- Solution Cloud
 - Simples à mettre en place
 - Facilement accessibles de l'extérieur
 - Plus « sexy »
- 3 Exemples : Travis CI, GitLab, Jenkins





Travis CI

- Solution Cloud
- Fonctionne naturellement avec GitHub
- Configuration dans les sources
 - Fichier de configuration yaml
 - Syntaxe simple & claire
- Gratuit sur les projets open-source





Travis CI

- **Avantages**

- Simple à mettre en place et à exploiter
- Beaucoup de mécanismes basiques clé en main
- Très visuel
- Parfait pour débuter

- **Inconvénients**

- Rapidement onéreux sur des repository privés
- Très dépendant de serveurs tiers
- Peut devenir difficile à utiliser pour du spécifique
- Écosystème peu modulable
- Pas le plus rapide





GitLab CI

- Solution Cloud ou auto-hébergée
- Fonctionne de pair avec GitLab
- Configuration dans les sources
 - Fichier de configuration yaml
- Gratuit dans la « Community Edition »





- **Avantages**

- Parfaitement intégré à GitLab
- Fonctionnement assez similaire à Travis CI
- Très visuel

- **Inconvénients**

- Peut devenir difficile à utiliser pour du spécifique
- Écosystème peu modulable
- Nécessite de faire le pari « tout GitLab »





Jenkins

- Solution historique (fork de Hudson)
- Généralement auto-hébergé mais existe en Cloud
- Configuration « à la papa »
- Gratuit et open-source





Jenkins

- **Avantages**

- Très documenté et suivi, nombreux plugins
- Permet d'absolument tout faire
- Gratuit

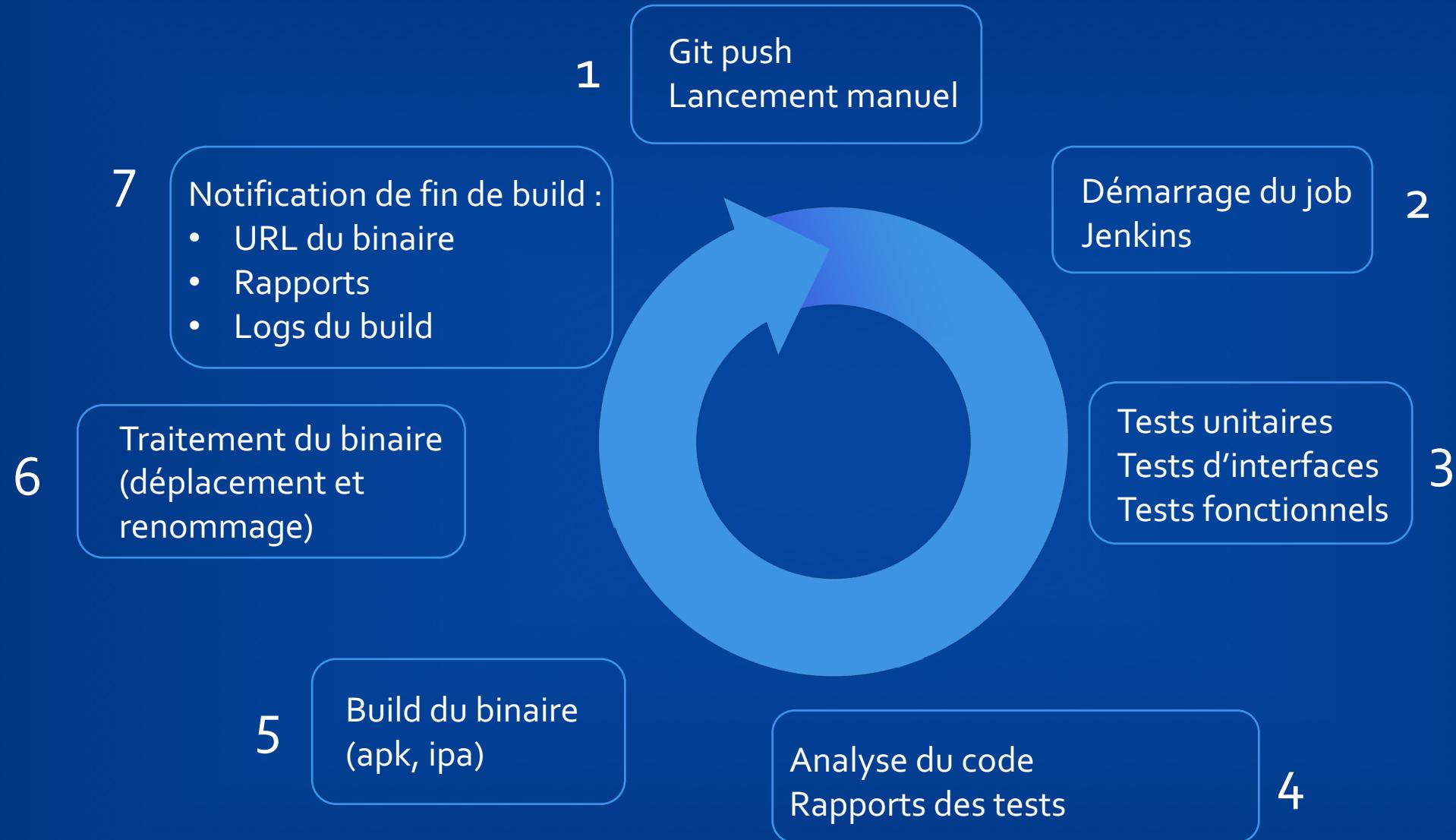
- **Inconvénients**

- Difficile à appréhender
- Assez peu visuel et intuitif



INTÉGRATION CONTINUE CHEZ AMILTONE





Démonstrations



Fastlane



Avant

- Outils d'automatisation
 - Android : gradle task • **Groovy**
 - Tous : scripts • **Shell**
- Difficultés
 - Mise en place
 - Montée en compétence
 - Maintenance





Outil d'automatisation en ligne de
commande pour projets mobiles



Sous le capot

- Ruby 🎥
- Environnement
 - Terminal
 - Linux ou macOS

```
+-----+-----+
|                               fastlane summary
+-----+-----+
| Step | Action                                | Time (in s) |
+-----+-----+
| 1   | Verifying required fastlane version | 0
| 2   | default_platform                      | 0
| 3   | produce                                | 13
| 4   | cert                                    | 5
| 5   | sigh                                    | 7
+-----+-----+
[23:25:36]: fastlane.tools finished successfully 🎉
```



Cas d'utilisation Android

- assemble + signature + zipalign + renommage d'apk
 - + envoi Play Store en beta
 - + envoi Appetize
 - + envoi Nexus
- tests instrumentation + screenshots + mise à jour fiche Play Store
- création émulateur + exécution tests d'instrumentation + suppression émulateur



Cas d'utilisation iOS

- upgrade build number + compilation + signature certif/provisioning
 - + envoi TestFlight
 - + envoi Appstore
 - + envoi Nexus

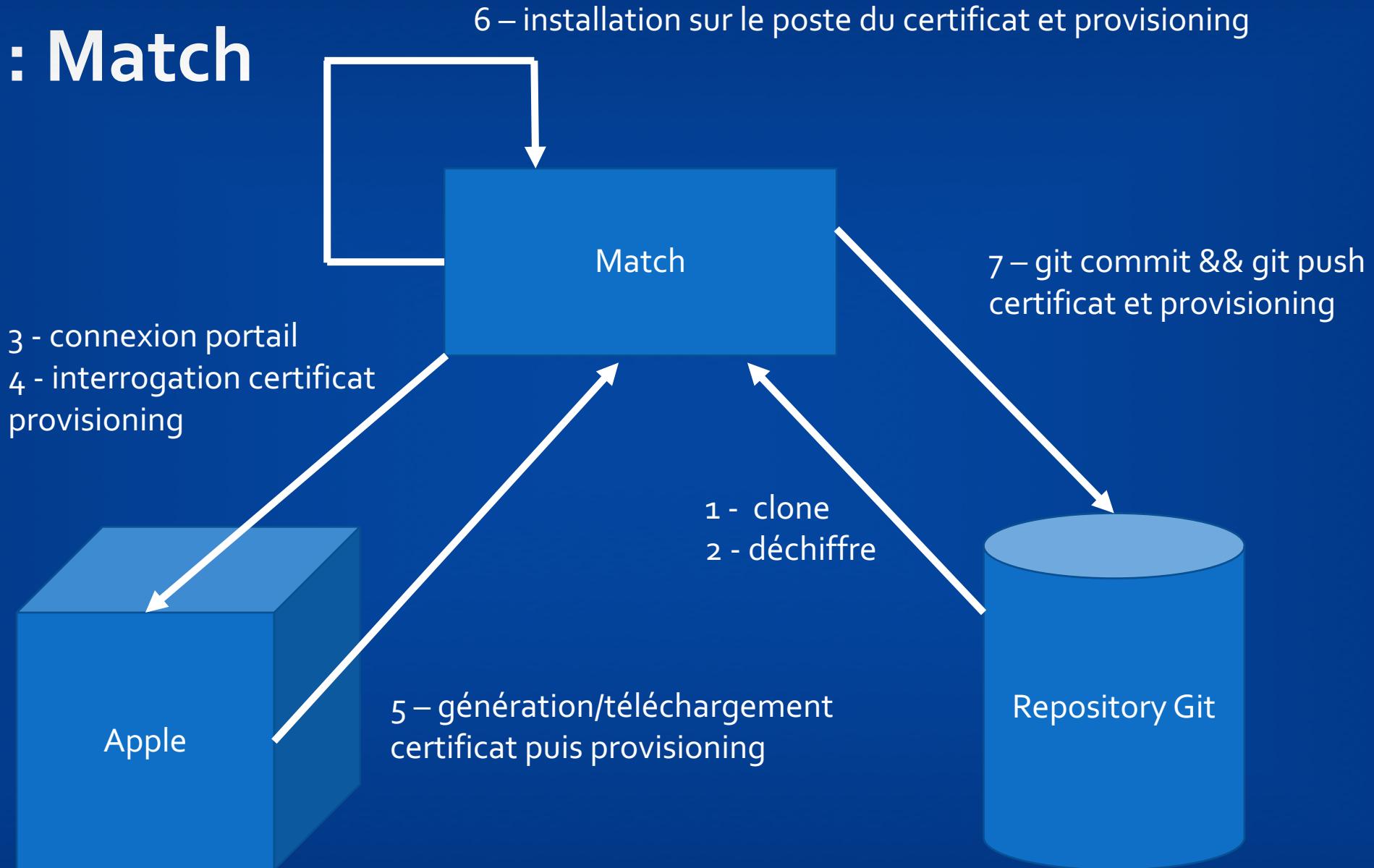


iOS : Match

- Générer les certificats et provisioning profiles
- Tout synchroniser sur un repository Git
- Sécurisé et automatisé



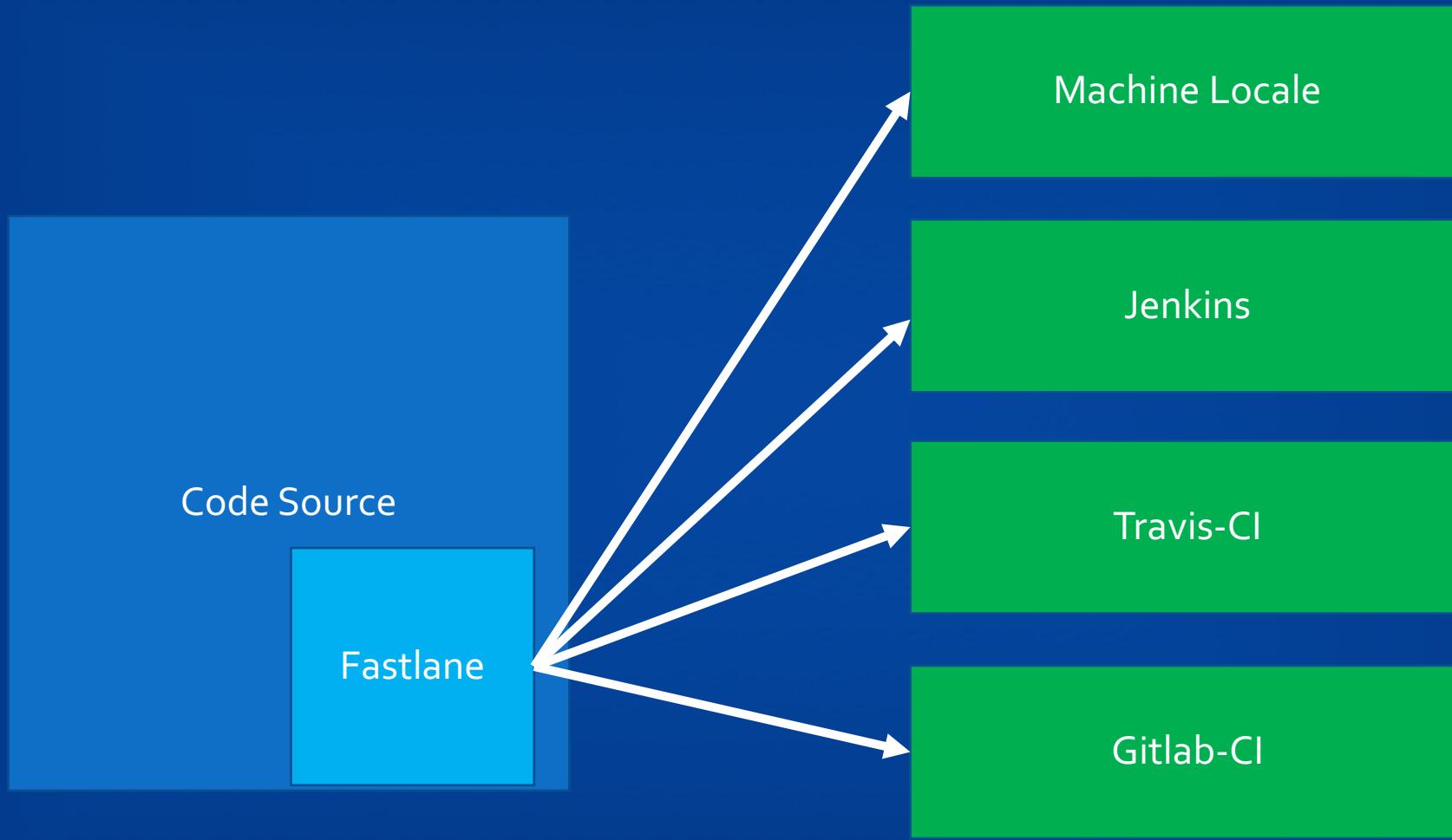
iOS : Match



iOS : Gym

- **xcodebuild** pour les gens normaux
- gère (presque) tous les types de compilation
 - Scheme
 - Target
- gère l'export
 - adhoc
 - in-house





Avantages

- Simple à prendre en main
- Utilisable par l'équipe de développement
- Centralisation des chaines de production ou d'intégration
- Facilement transposable sur n'importe quel CI
- Possibilité d'écrire ses propres actions
- Tout est combinable et réutilisable



Questions



Projet



Consignes

- Projet Android
 - Hébergé sur GitHub
 - Intégration continue Travis-CI
-
- Groupes formés au premier TD
 - Rendu au plus tard le 14 décembre à 23h59
 - Triche détectée = zéro



Evaluation

- Tout ce qui a été enseigné : note sur 20
- Points bonus à glaner
- Note finale du module : Contrôle continu (coeff 0,15) + Note du projet (coeff 1)



Projet Android

- Application de consultation d'information sur des Comics
- Un fichier .json en entrée pour les données
- Ecran d'affichage de la liste des comics
- Ecran d'affichage du détail d'un comics
- Partage d'un comics



adescamps@amiltone.com
mobilefactory.amiltone.com

