D HAT
ORUM
pe, Middle East & Africa

re

PaaS + Commodity Hardware : Mise en place d'un socle d'infrastructure agile

Philippe Poirier
Cabinet d'Architecture / La Banque Postale

LA BANQUE POSTALE

roupe La Poste

nes d'activité

urquoi construire un nouveau socle technique

évelopper / déployer le nouveau SI Distribution Omni Canal « Cap Client 3.0 » de
nque Postale

Replatformer » le patrimoine J2EE legacy cœur du SI Bancaire

aiter l'obsolescence du STMC* historique de la Banque Postale

Et disposer d'un écosystème technique plus agile

*Socle Techniqu

rs IT pour gagner en agilité

iser / Simplifier les chaines de liaison

er les standards du Web (IaaS/X86/Open Source)

r les principes de résilience au niveau logiciel pour utiliser du Commodity Hardware.

un meilleur découplage Application / Middleware / Infrastructure

er la productivité sur l'ensemble du cycle projet (délivrer plus rapidement de nouvelles nalités métier, innover plus vite)

re les changements techniques / applicatifs sans interruption de service

dynamiquement les infrastructures à l'activité transactionnelle réelle

les usages métier SI des standards techniques limitants (nombre de MEP aux limites de la é, …)

*Socle Techni
*Mise En Pro

ndrier

but du RFP

Nouveau socle fabriqué avec l'aide de Red Hat Consulting

Choix du PaaS OpenShift + JBoss

IaaS v1 livré
***STOM v1 livré**

***MEP CC 3.0 R3**

Premiers repla
Application J2

Construction du socle technique

- Construction des environnements
- Tests de charge du STOM
- Tests d'exploitabilité d'une application J2EE Legacy en production
- Hébergement du cycle de vie logiciel de la 1ère release CC3

Jan 2015 | **Juin 2015** | **Mai 2016** | **Oct 2016** | **N**

r signe d'agilité :

lous n'avions jamais fabriqué un nouveau socle technique en moins de 1 a

: Tous les écosystèmes techniques ne sont pas égaux

cle technique Multi Canal
eries / Unix AIX / WAS

*Chaine de liaison

cations = N actes de déploiement

dédiés à des applications

jour avec interruption de service

rtante de coordination /
ur EPA & MCO

humaine sur incident

créer, maj & tester

Patterns d'architecture

Simplification

+

Rationalisation

+

Standardisation

+

Automatisation

Aujourd'hui : Socle Techniqu
x86 / OpenShift / JB

1. L'Unité de déploiement es
   de liaison + l'application
2. Le serveur n'est plus déd
   application
3. Mise à jour possible sans
   de service
4. Réduction des charges d
   planification & coordinatio
5. Reprise automatique sur
6. PCO / PRA natif

Au-delà de l'infrastructure technique x86,
économies les plus importantes sont générées par l'agilité du modèle opérationnel

oom sur les déploiements techniques et applicatifs

Hier Sur STMC : Websphere	Aujourd'hui Sur STOM : OpenShift/JBoss
Environnements Amonts : Assemblage, Test fonctionnel, Recette utilisateur, Test de charge, .	
TMC : 1 à 2 installations du socle (0,5 à 1 jrs par tallation)	**STOM** : Le PaaS est installée une fois pour toute
pplication : à n déploiements (30 mn par déploiement) ombre de déploiement limité par jour	**Application** : •8 à n déploiements de 3 mn →*CDL : container + HAProxy + Jboss + Application •Plus de limite : on déploie aussi souvent que nécessaire
Environnements de Pré-production et de Production	
cle STMC : 2 installations (HPIA+HPIB) **plication** : 2 déploiements actif/passif	**Socle STOM** : Le PaaS est installée une fois pour toute **Application** : 2 déploiements actif/actif
Rupture technique, rupture de charge	Le même pattern d'architecture, le même outil déploiement

00 déploiements / an, > 400 personnes en capacité de déployer (dont 80 exploi

D HAT
ORUM
pe, Middle East & Africa

re

urquoi évoluer vers le commod hardware ?

- IaaS v2 / STOM v2 -

nements

possible de développer des applications nativement résilientes

plications peuvent s'affranchir des mécanismes de résilience portés jusqu'alors par tructure technique

uxueux et couteux d'héberger des applications résilientes sur un IaaS Résilient

andards IT nativement résilients sont incompatibles avec les infrastructures résiliente nBase, …)

ions IT engagées

Création du IaaS v2 : Enrichissement du IaaS v1 avec une offre commodity hardware non résiliente :

- ➔ « Servers bare metal » avec disques embarqués
- ➔ Abandon de l'hyperviseur et de ses mécanismes de résilience

Création du STOM v2 basé sur OpenShift v3.3 :

- ➔ Traiter l'obsolescence de OpenShift v2
- ➔ Disposer de la boite à outil « Docker »
- ➔ Améliorer la haute disponibilité (vulnérabilité sur gear primaire supprimée)
- ➔ Gérer plus finement la scalabilité horizontale (cpu & ram)

pes directeurs retenus pour le « commodity hardware »

astructure commodity hardware n'embarque pas de mécanisme de résilience, le SLA*

pplications hébergées sur une infrastructure SLA 0 sont nativement résilientes

nne d'un serveur est une situation normale, elle est sans impact sur la QoS des applic
gées dans le SI

sque de panne est compensé par :

- Un nombre suffisamment important de serveurs physiques différents
- Les applications sont présentes sur plusieurs serveurs physiques différents
- Un serveur physique n'héberge pas un trop grand nombre d'applications

Comment adresser les besoins en résilience du STOM v2 ?

en cours avec RH Storage pour adresser les besoins en résilience du STOM v2 par le s

*Service Level Agreement

aines échéances

er 2017 : Livraison du STOM v2 sur « Commodity hardware »

017 : Déploiement de la release 5 de Cap Client 3.0

017 : Extension du STOM v2 au PHP

remiers déploiements des applications issues de la filière de développement tactique

016 à T4 2019 : Replatforming des applications legacy J2EE Intranet / Inter
ies / Websphere) sur STOM v2

egacy J2EE actuellement hébergé sur Websphere Pseries/AIX

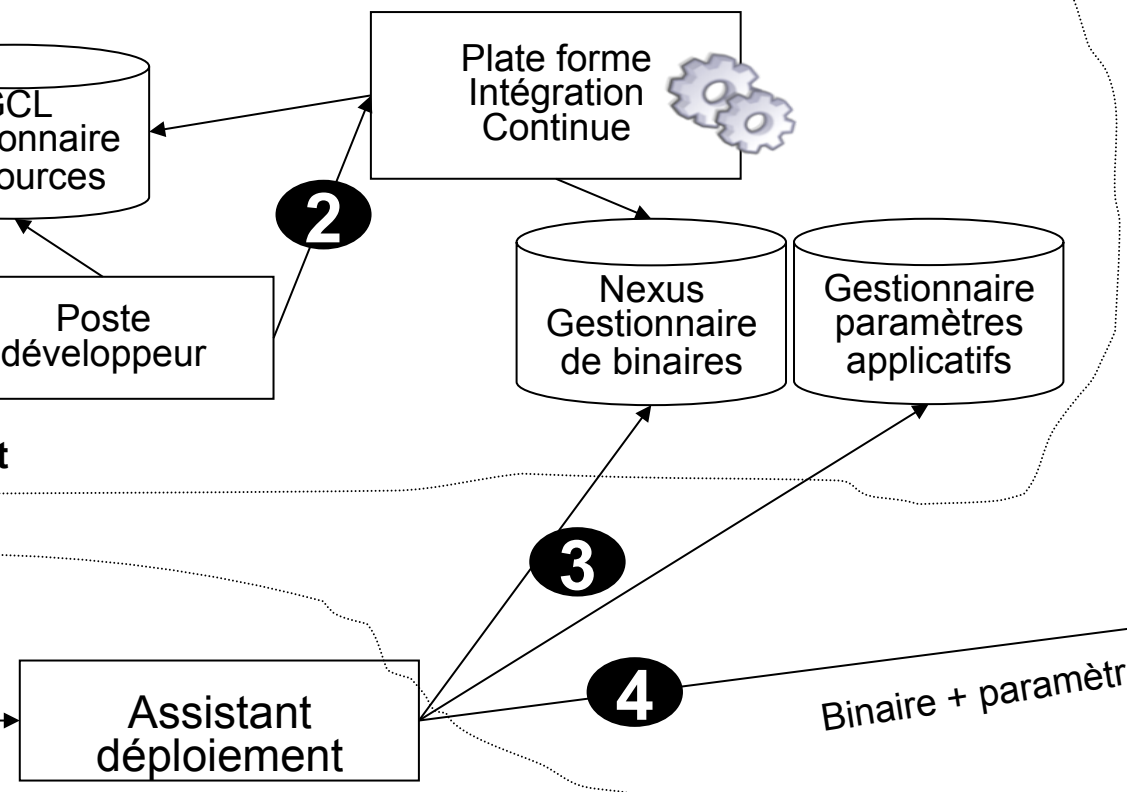
Avec OpenShift

osons d'une véritable plateforme de consolidation pour simplifier l'exploitation d'un é
evenu trop distribué et complexe à opérer en dehors des pratiques du cloud computi

ANNEXES

ement des activités du cycle de vie logiciel

mpilation, tests unitaires
continue (contrôle Qualité, fabrication du binaire,
utomatisée de tests bouchonnés, …)



3 Récupérer les binaires des applications +
associé à l'environnement cible.

4 Déploiement (via API) de l'application su

STOM (OpenShift / Jboss

A
P
I

5 OpenShift exécute les tâ
déploiements sur le(s) envi
concernés

Déploiement continue (x)
Assemblage (3)
Test fonctionnel (3)
Recette utilisateur (3)
Test bout en bout (3)

RIT (2)
Qual. Ma
Rece
Pr

lication déploiement :

ploiement des applications sur l'ensemble des phases
ntrôler le workflow applicatif (phases précédentes réalisées,
haines phases, critères qualité respectés,…)

redhat®