

# Introduction au PaaS à travers l'exemple de Google appEngine

Khaled BEN DRISS

Directeur BU Ingénierie Logicielle

Khaled.bendriss@oxia-group.com

6 avril 2011



## Les Mercredis de l'innovation - OXIA

# Khaled BEN DRISS

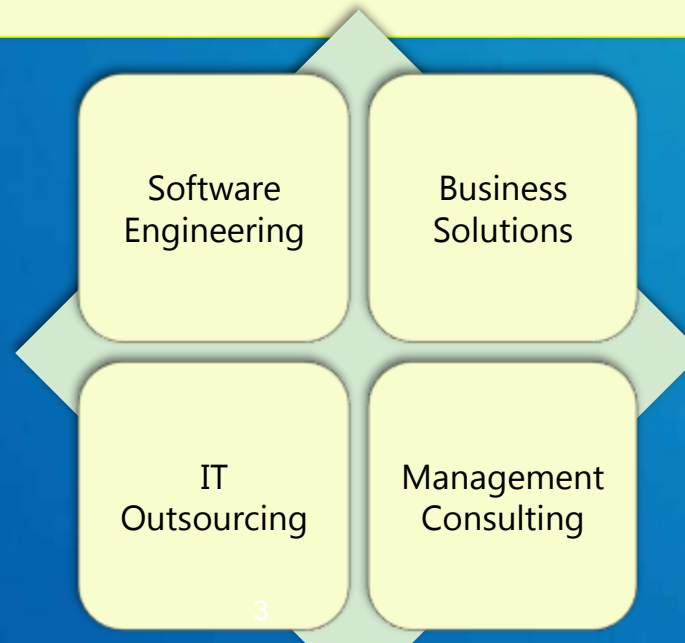


- Ingénieur de l'Ecole Centrale de Lyon (ECL'1991)
- Docteur de l'Université Paris 6 ( P&MC ' 1994)
- Ancien Maître Assistant à l'Ecole Polytechnique de Tunisie
- Directeur BU ingénierie Logicielle
  - & Consultant Senior : Architecte de Système d'Information
- Membre IEEE (Executive committee 2011-2012 du Tunisia section )
- [www.oxia-group.com](http://www.oxia-group.com)
- Blog : <http://net-progress.blogspot.com/>

# Le groupe OXIA en bref

## ● Un leader des Systèmes d'Information et du Nearshore au Maghreb :

- Date de création : 1998
- Forme juridique : Société Anonyme
- Effectif : +250 personnes (juin 2012)
- 80% de l'activité réalisée en dehors de la Tunisie
- Filiale à Paris
- Filiale à Alger
- Filiale à Dubai
- Certification ISO 9001 version 2000 (AFAQ QUAL/2005/25140 ) depuis +8 ans



# Agenda

- Rappels Cloud Computing
- XaaS : la trilogie du cloud
- Le cloud et ses usages
  - Scénarios d'application en entreprise
- Zoom sur le PaaS de google
- Questions / Réponses

# Le révolutionnaire : IT = commodité

Une transformation de l'industrie est en cours : La suprématie du web Browser

Electricité



Eau



Construire votre solution

PlugIn, souscrire & payer à l'usage

# Le révolutionnaire : IT = commodité

Une transformation de l'industrie est en cours : La suprématie du web Browser

Electricité



Eau



Application  
métiers



Construire votre solution

PlugIn, souscrire & payer à l'usage

# La trilogie du cloud computing : XaaS

- Système d'exploitation complet, ...
- Exemples
  - Flexiscale
  - AWS: EC2



IaaS  
Infrastructure as a  
Service

## Modèle de fourniture d'infrastructure

- Accès à une pile d'infrastructure :
  - Système d'exploitation complet
  - Firewalls
  - Routeurs
  - Load balancing

# La trilogie du cloud computing : XaaS

SaaS  
Software as a  
Service

## **Modèle de fourniture de logiciel**

- Un « business model » pour des composants logiciels hébergés, paiement à l'usage ou à la demande
- Pas de logiciel, ni de serveur à gérer
- Le Service est délivré à travers le navigateur web

- CRM, Planification financière, RH, office, email
- Exemples:
  - Salesforce.com
  - SurgarCRM
  - Email : Live, gmail, Hotmail
  - Docs.com, googleDocs

# La trilogie du cloud computing : XaaS

PaaS  
Platform as a  
Service

## **Modèle de fourniture de Plateforme**

- Fourniture d'une plateforme logicielle hébergé pour déployer des applications (ciblant la plateforme)

- Stockage, base de données,

- Exemples :

- Google App Engine

- Azure de Microsoft

- Amazon WS: S3

- cloudfoundary

# IT as a Service



***Software***  
*as a Service*

***Donner le pouvoir aux  
utilisateurs***



***Platform***  
*as a Service*

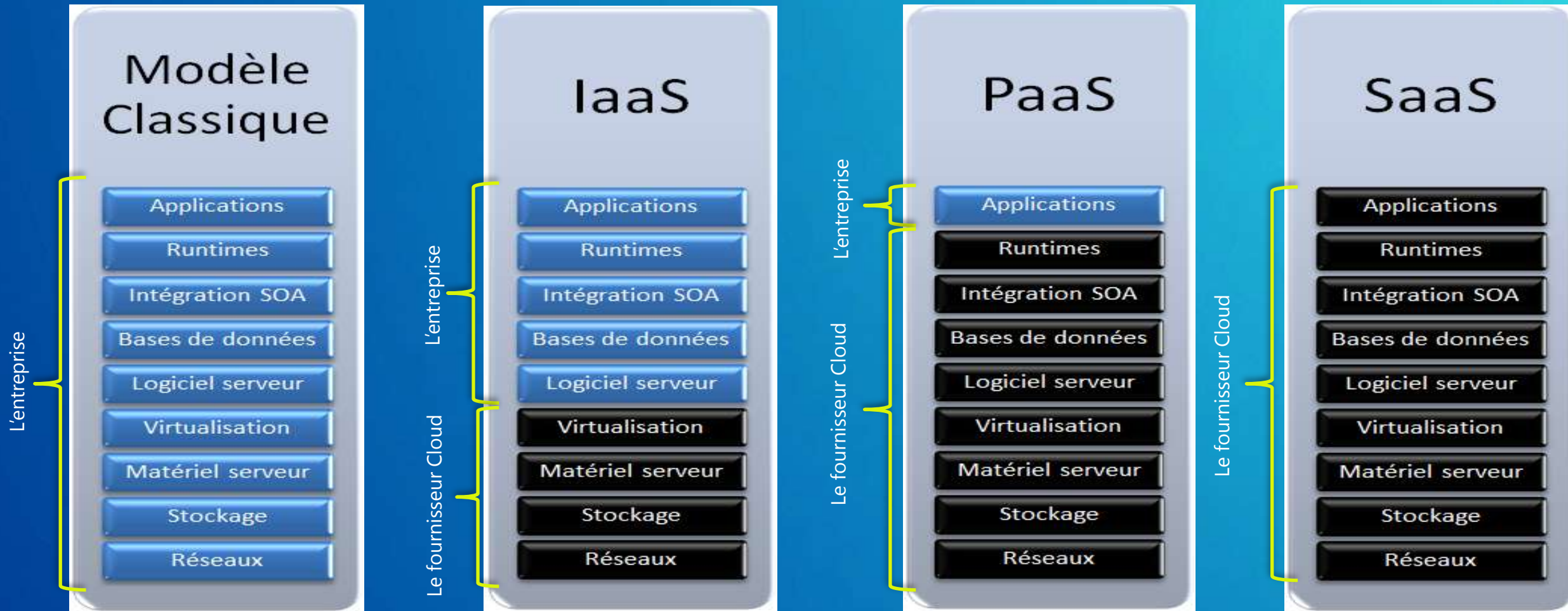
***Développer/ déployer  
rapidement des applications***



***Infrastructure***  
*as a Service*

***Minimiser la gestion de  
l'infrastructure***

# IaaS, PaaS, SaaS : qui maintient quoi ?



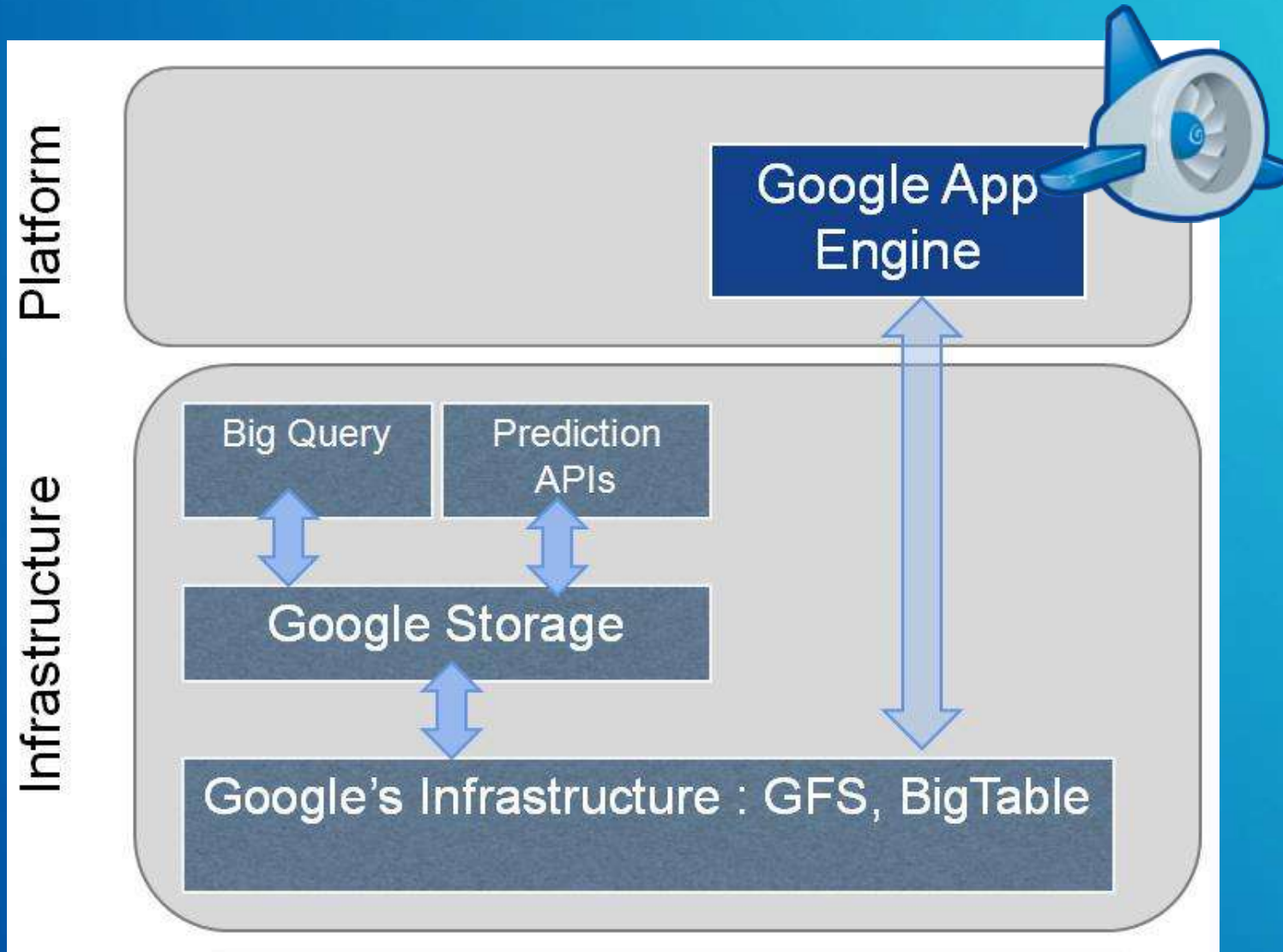
# 5 caractéristiques essentielles

- Self-service, à la demande
- Accès réseau ubiquitaire
- Mise en commun des ressources
  - Indépendance de l'emplacement
  - Homogénéité
- « Elasticité » rapide
- Service mesuré en permanence

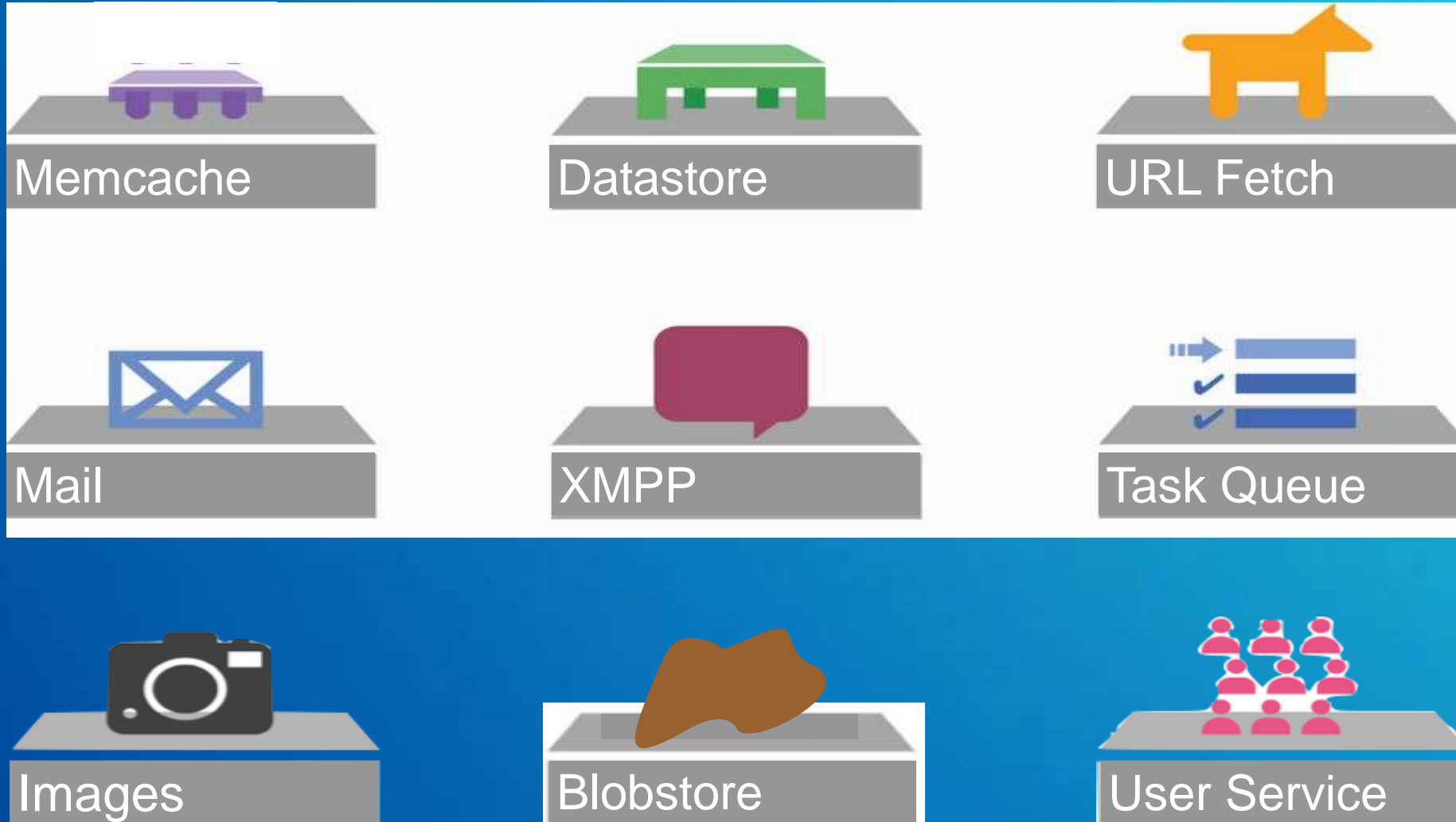
# Un PaaS sert à simplifier la vie du développeur

- Objectifs
  - Simple à utiliser
  - Simple à administrer
  - Montée en charge simple

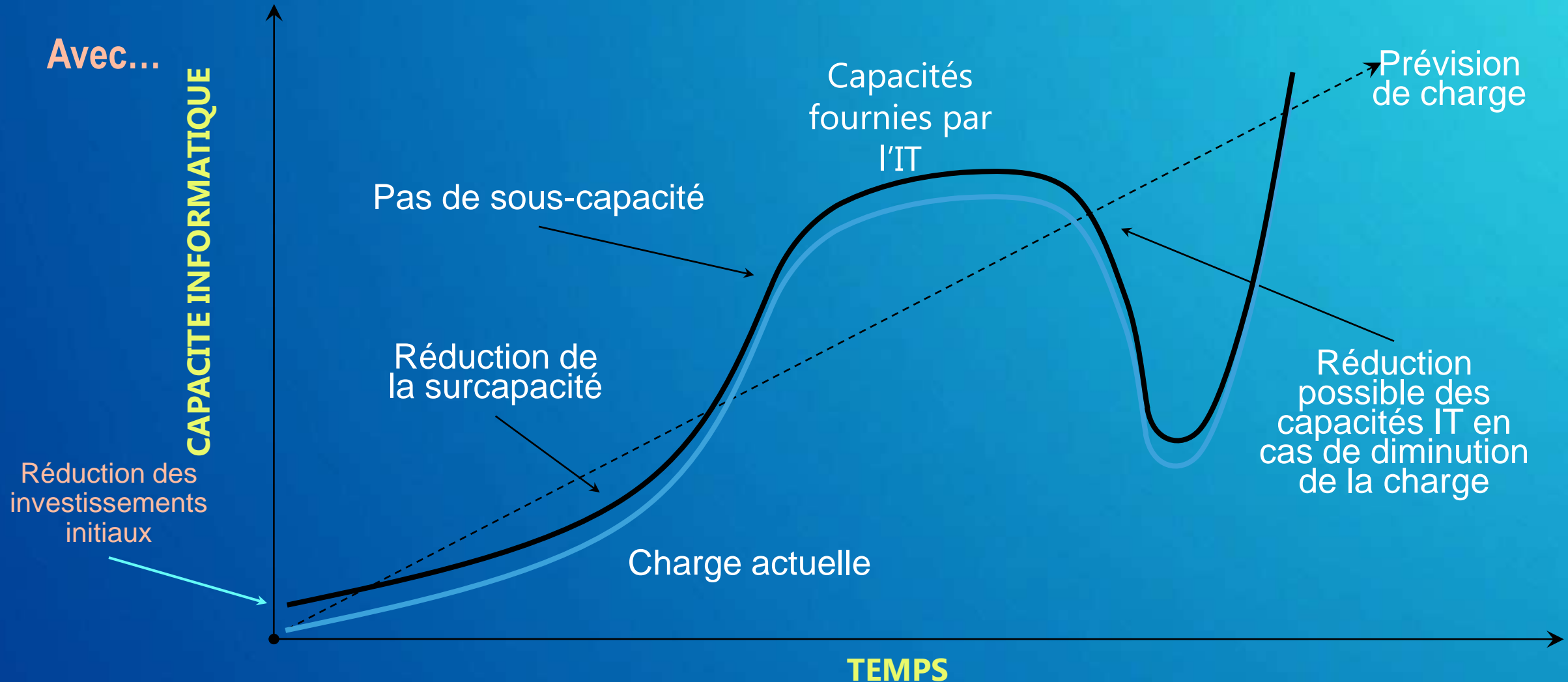
# Vue globale du PaaS (exemple de google)



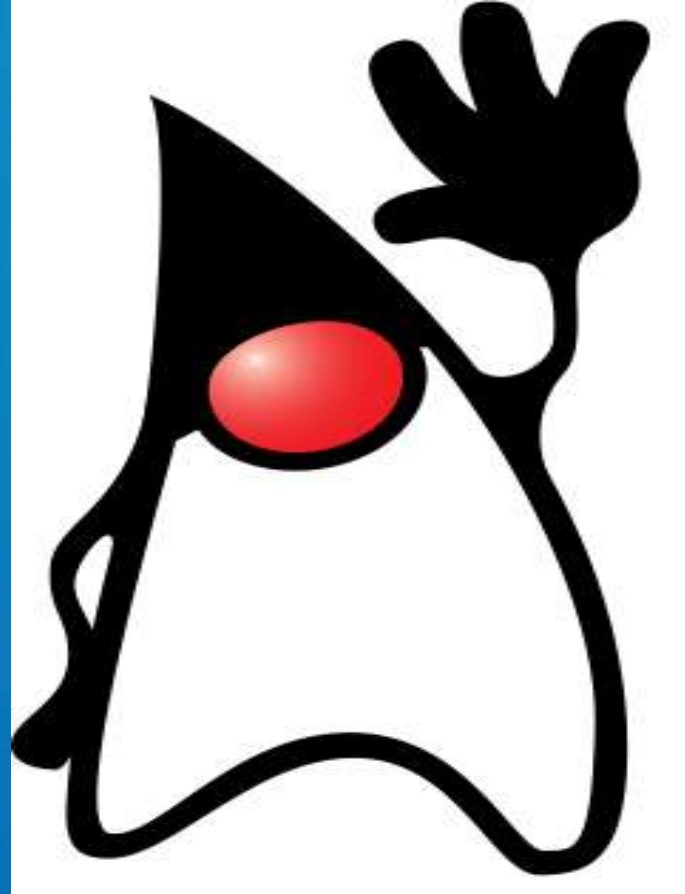
# Services accessibles au PaaS de google



# Bases économiques du Cloud Computing

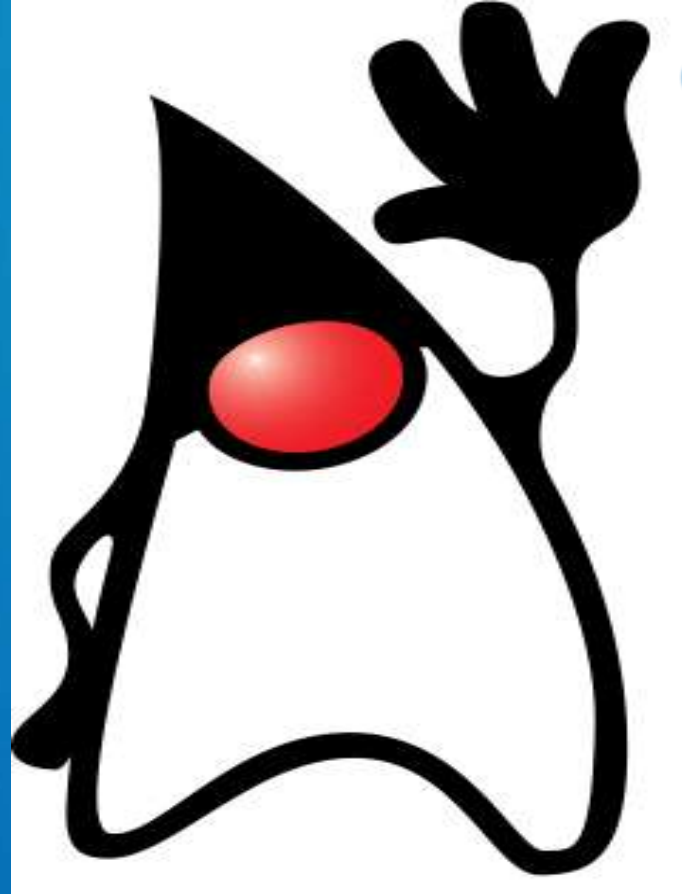


# Language runtimes



Java

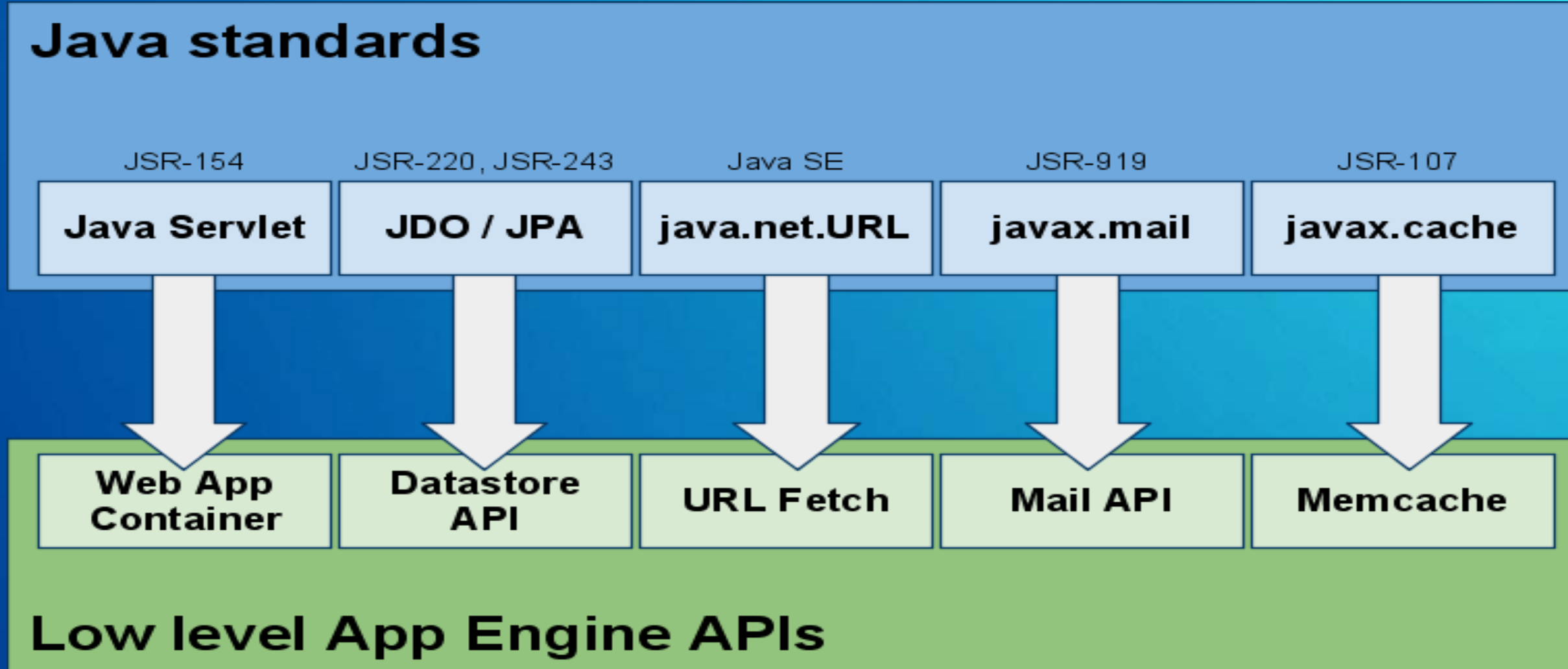
# Language runtimes



Java

- JVM languages
  - Scala
  - JRuby (Ruby)
  - Groovy/Grails
  - Quercus (PHP)
  - Rhino (JavaScript)
  - Jython (Python)

# Assurer la portabilité



# Vision sur les étapes

- Terre à terre : du code source

# Vision sur les étapes

- Demander la création d'un compte le compte
- Attendre l'arrivée d'un code secret par SMS pour compléter la création du compte GAE
- Créer une application
- Le lien : <http://appengine.google.com/>



# Créer votre application

Google App Engine khaled.bendriss@gmail.com | [My Account](#) | [Help](#) | [Sign out](#)

## Create an Application

**Application Identifier:**  
 .appspot.com   
You can map this application to your own domain later. [Learn more](#)

**Application Title:**  
  
Displayed when users access your application.

**Authentication Options (Advanced):** [Learn more](#)  
Google App Engine provides an API for authenticating your users. If you choose not to use this, anyone in the world will be able to access your application. However, if you choose to use this, you'll need to specify how who can sign in to your application.

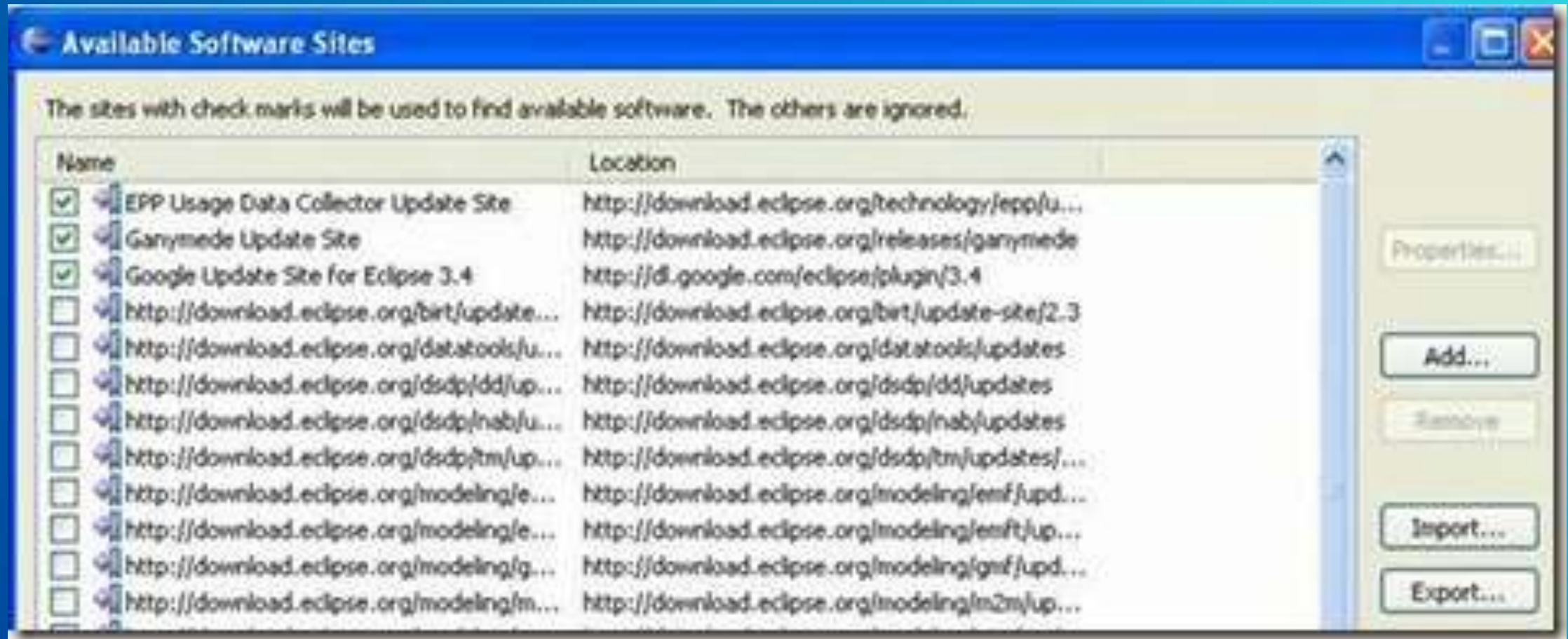
**Open to all Google Accounts users (default)**  
If your application uses authentication, anyone with a valid Google Account may sign in. (This includes all Gmail Accounts, but does not include accounts on any Google Apps domains.)  
[Edit](#)

**Terms of Service:**  

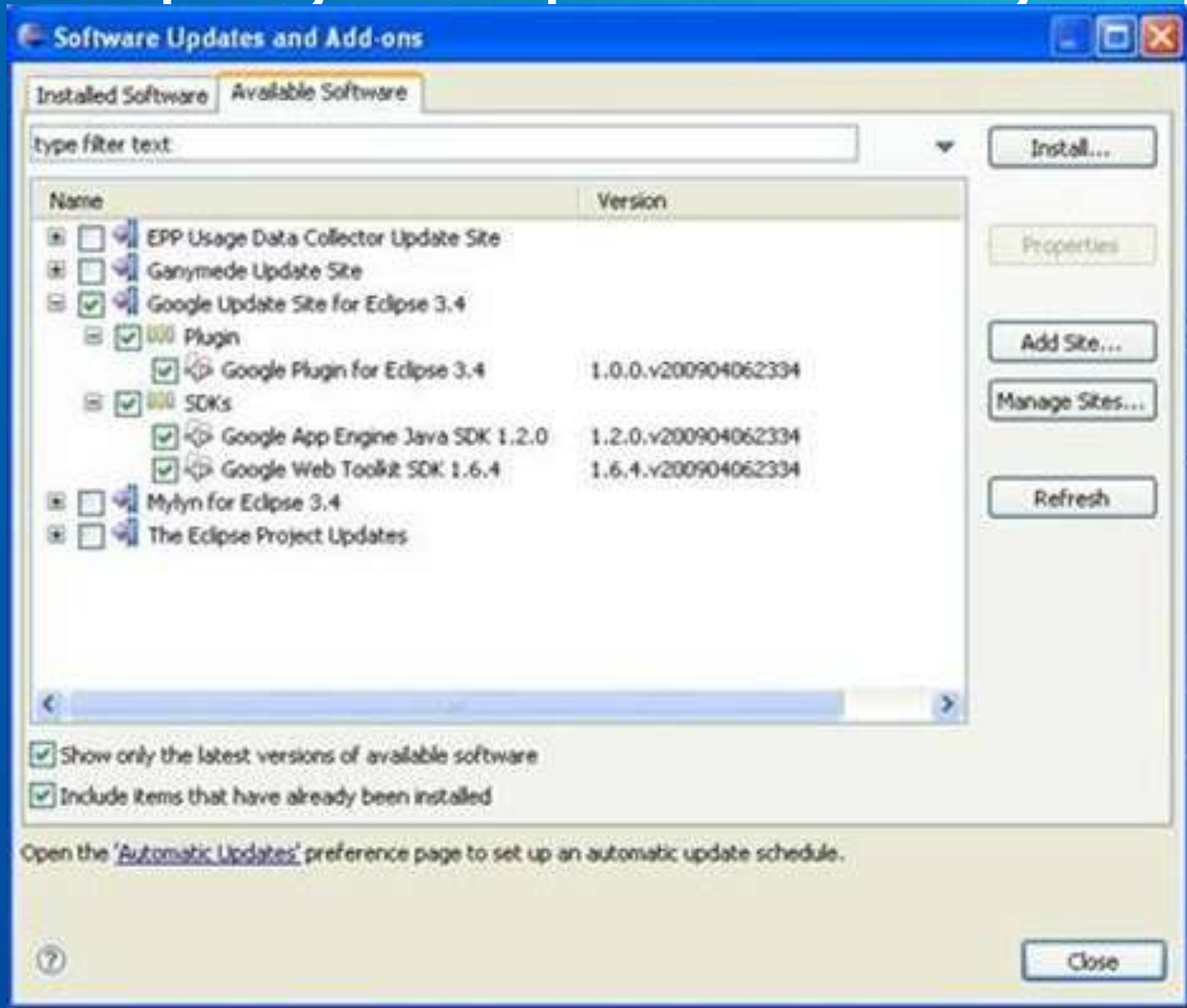
application. You must provide legally adequate privacy notice and protection for those users. To do so, at a minimum, you must incorporate the privacy terms available at <http://code.google.com/appengine/privacy.html> into the privacy policy for your application. If the users provide you with user names, passwords, or other login information or personal information, you must make the users aware that the information will be available to your application and to Google.

3.4 If the user provides you with Google Account information, you may only use that

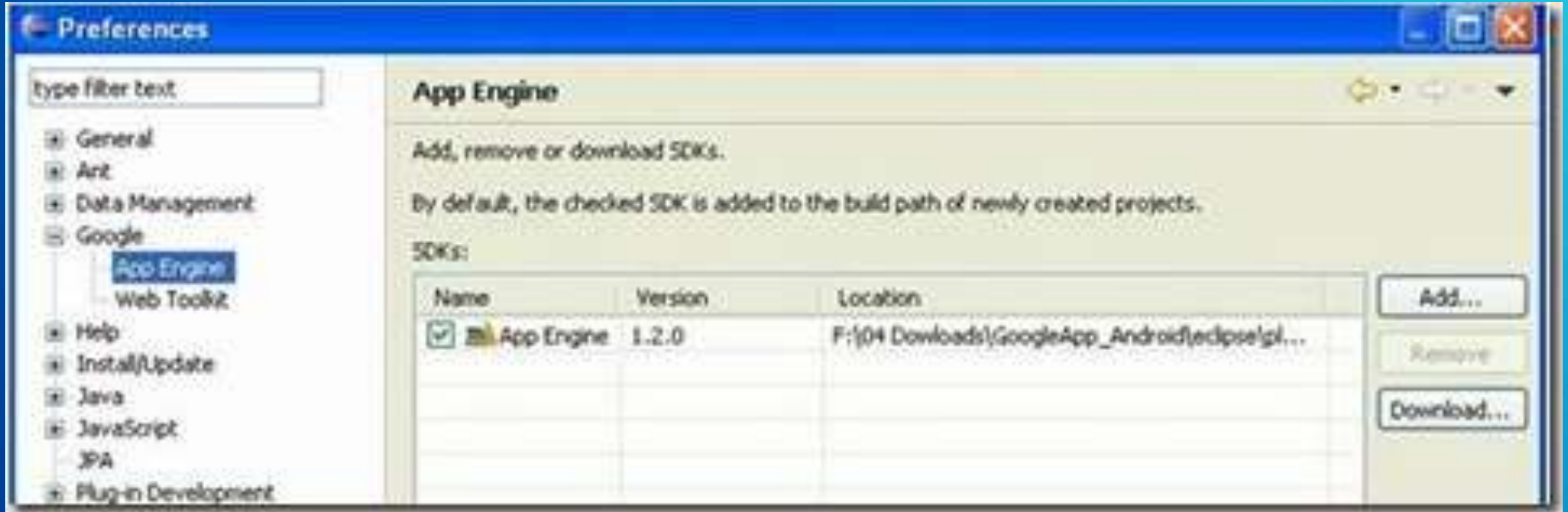
# Installer le plugin Eclipse de Google App Engine



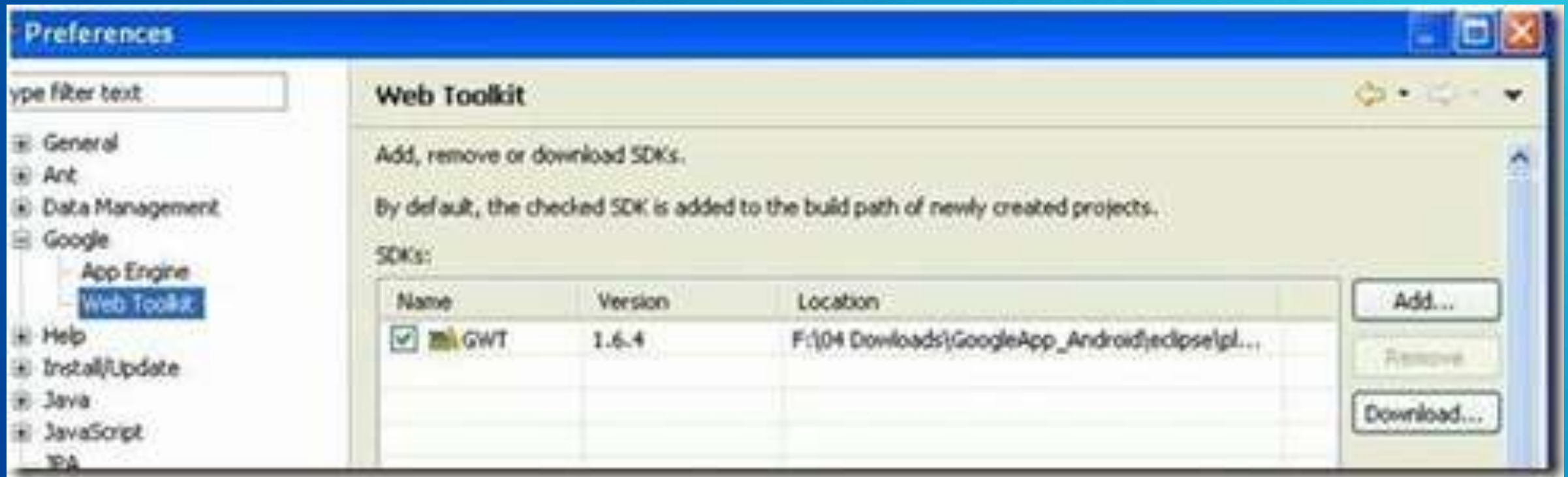
# Installer le plugin Eclipse de Google App Engine



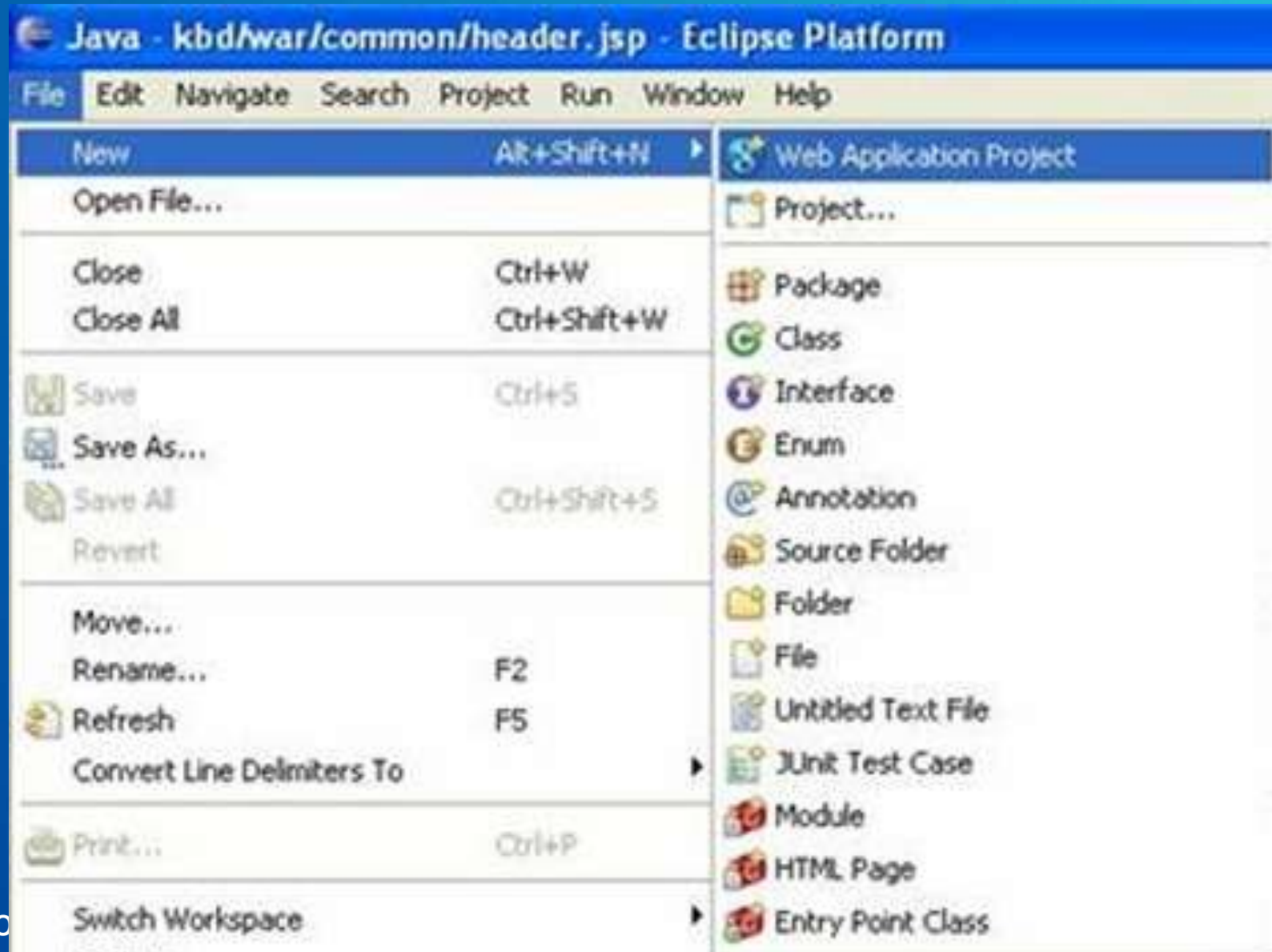
# Utiliser Eclipse GAE



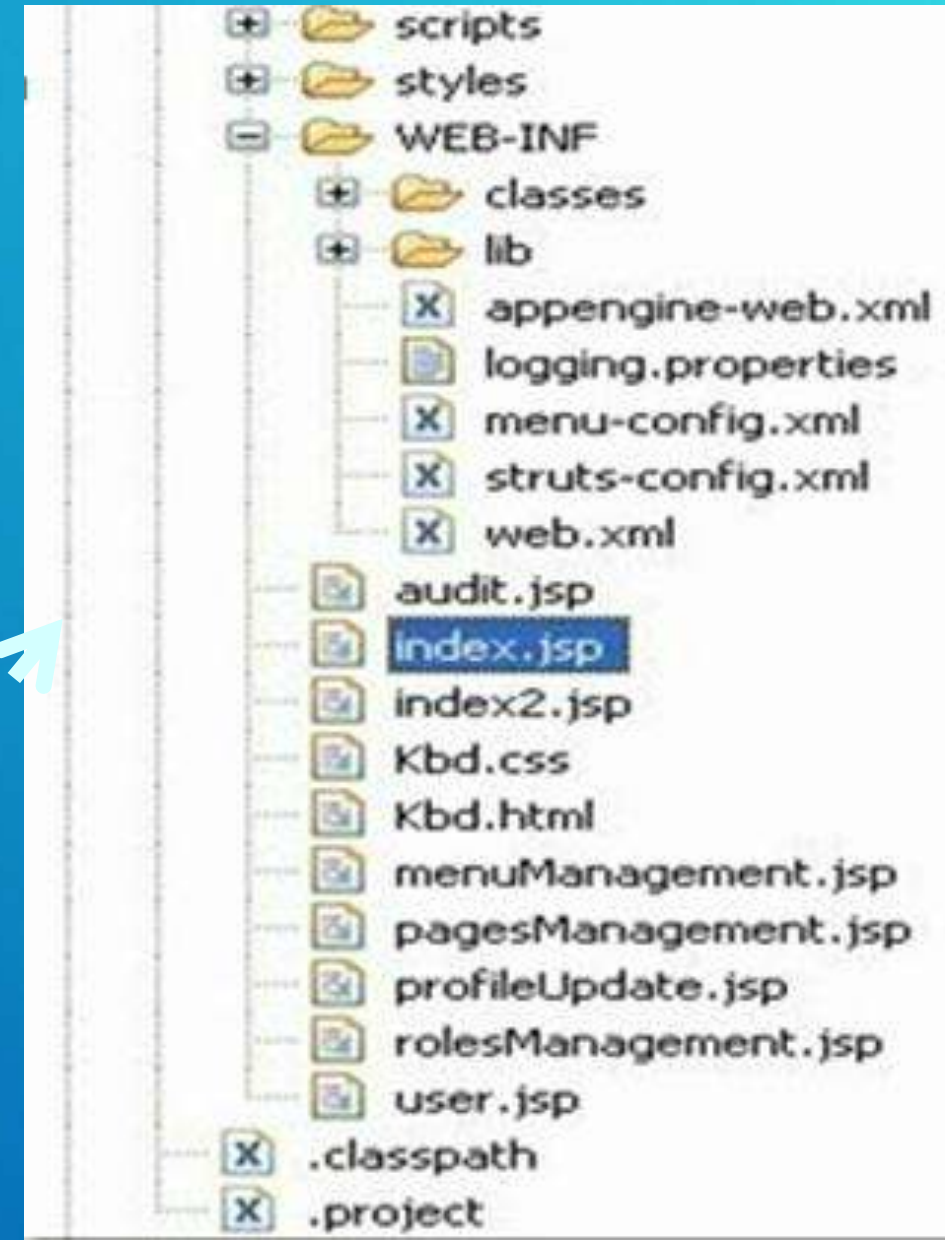
# Utiliser Eclipse GAE



# Continuer normalement avec Eclipse

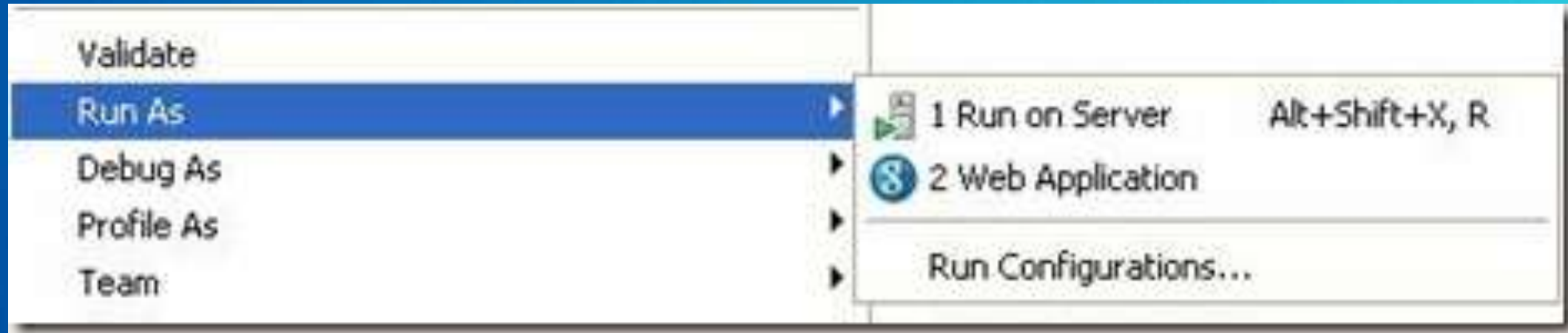


# Continuer normalement avec Eclipse



Sfax 27/12/2012

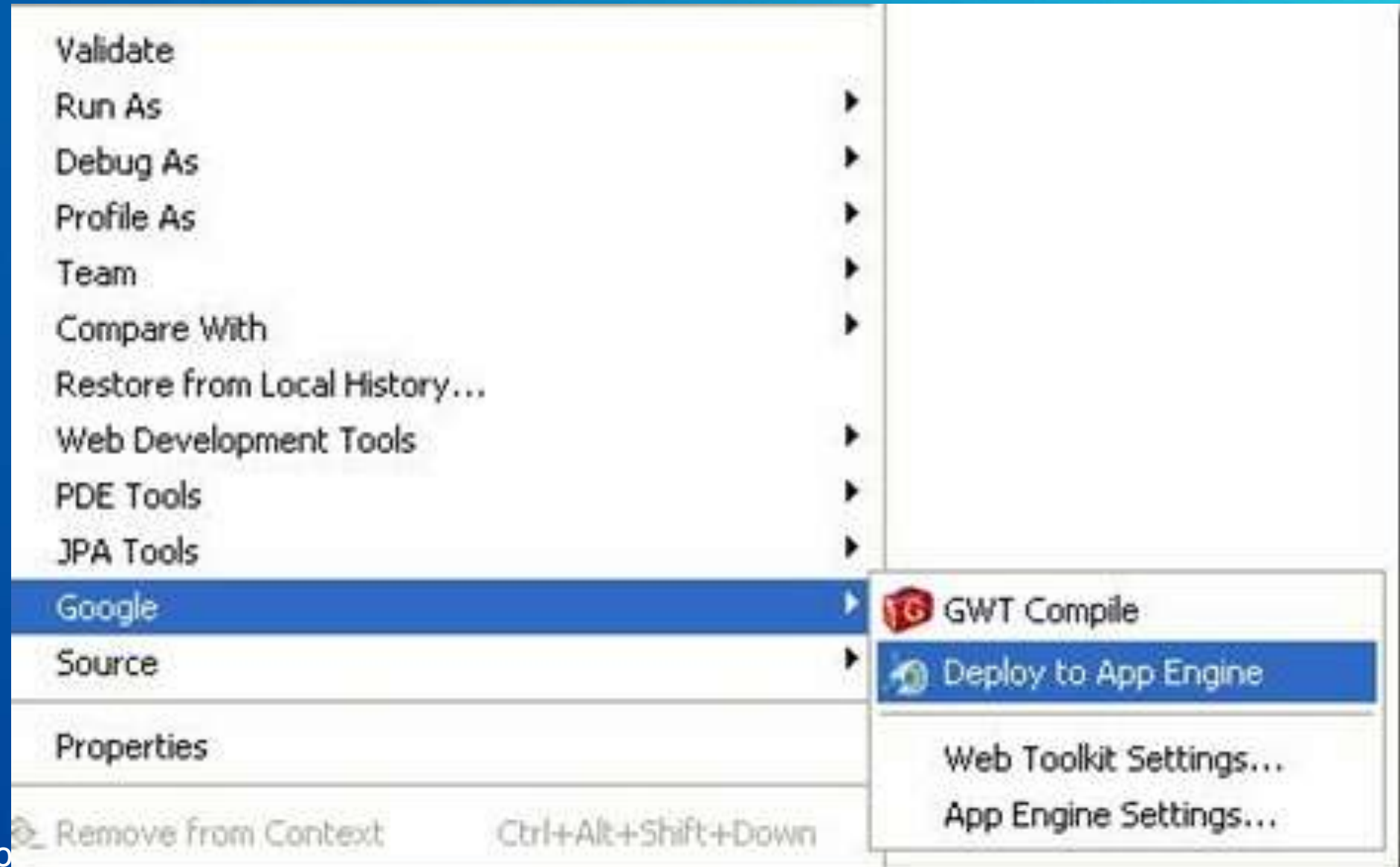
# Exécuter en local



# Exéc



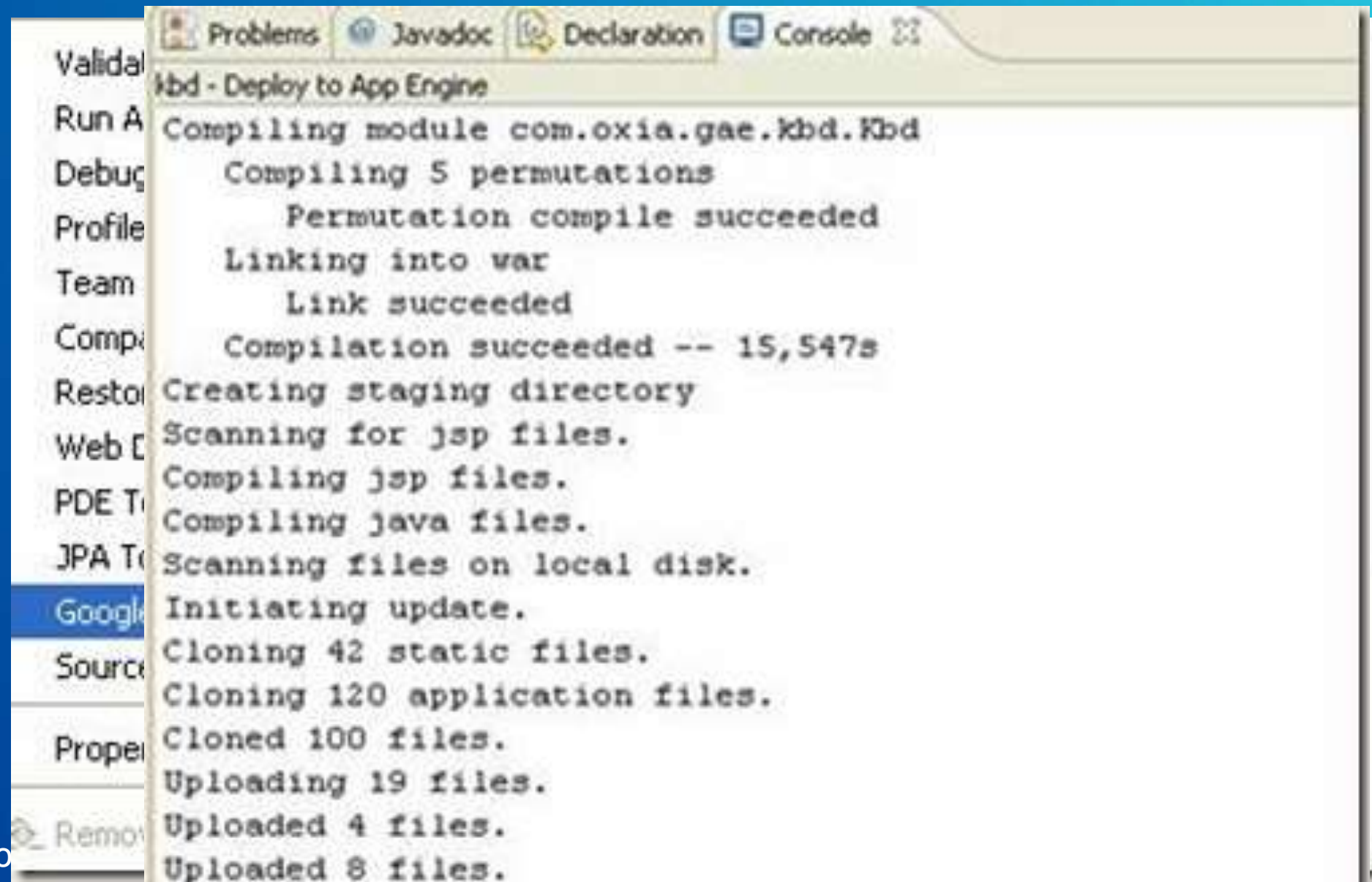
# Déploiement dans les nuages (GAE)



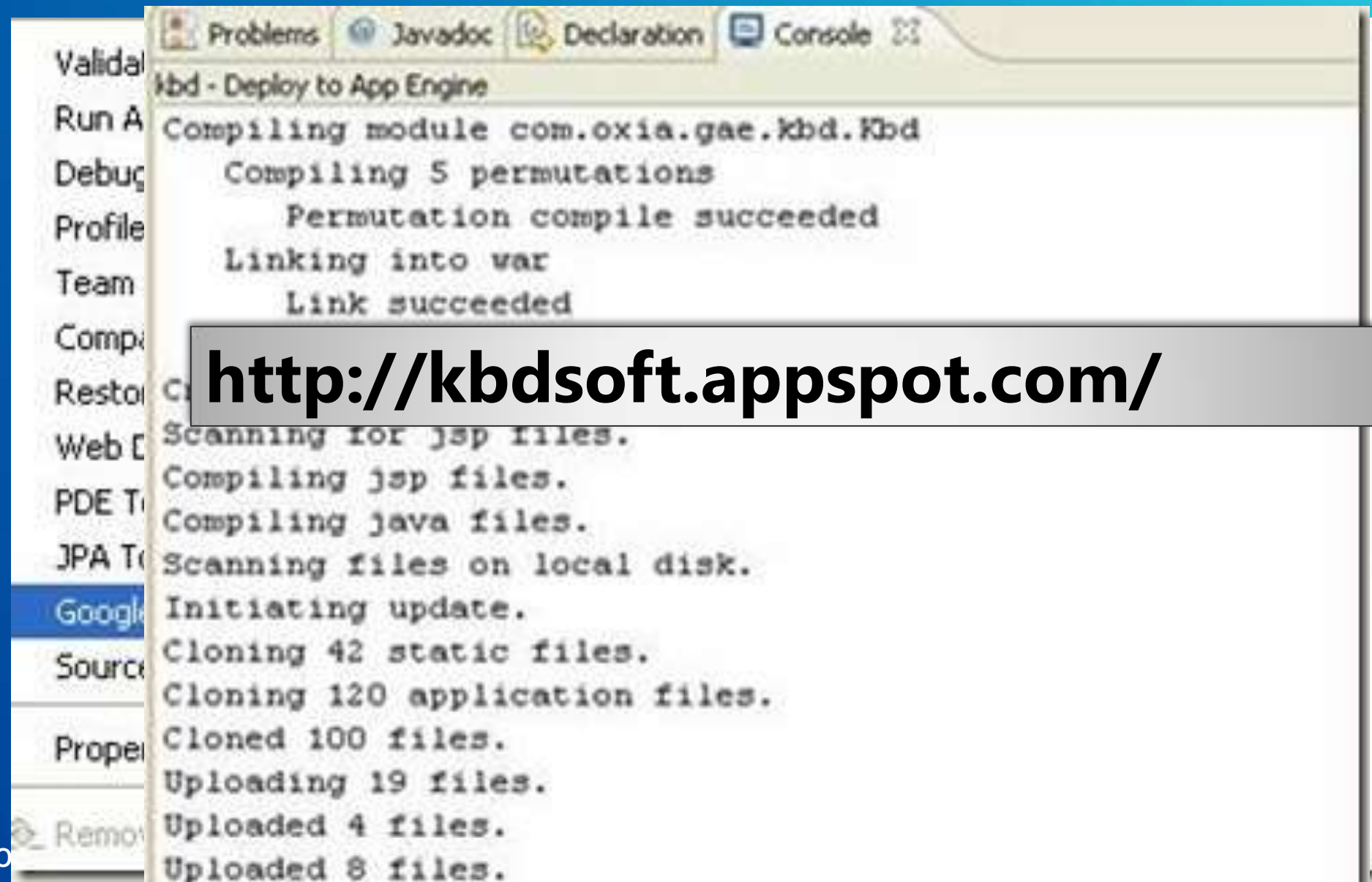
# Déploiement dans les nuages (GAE)



# Déploiement dans les nuages (GAE)



# Déploiement dans les nuages (GAE)



# Idem avec Grails, mais en 10 étapes

1. `grails create-app my_First_3G_app`
2. `cd my_First_Grails_GAE_app`
3. `grails uninstall-plugin hibernate`
4. `grails install-plugin gorm-jpa`
5. `grails install-plugin app-engine`
6. `grails create-domain-class com.oxia.att.test.Message`
7. `grails generate-all com.oxia.att.test.Message`
8. add to `grails-app\conf\Config.groovy` -  
`google.appengine.application=" app3g"`
9. `grails set-version 1`
10. `grails app-engine deploy`

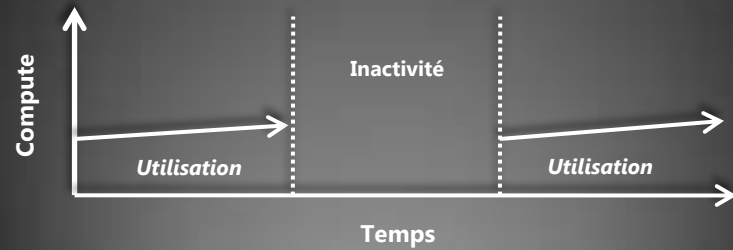
- Pour visualiser le résultat, allez au <http://app3g.appspot.com>  
Exemple de cette approche : <http://np-grails.appspot.com/twitting/list>

# Attention : le modèle de développement change avec un PaaS

- Un PaaS/cloud offre des dizaines de machines en parallèle  
Vs un super serveur

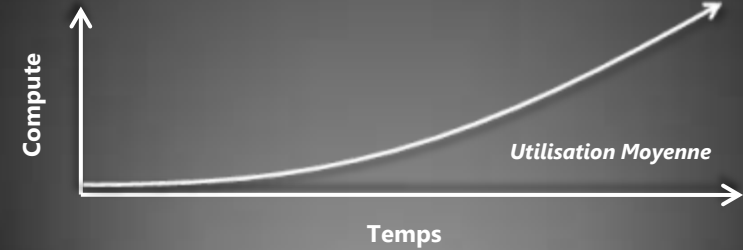
# Scénarios pour le PaaS/Cloud

## "Puissance Ponctuelle"



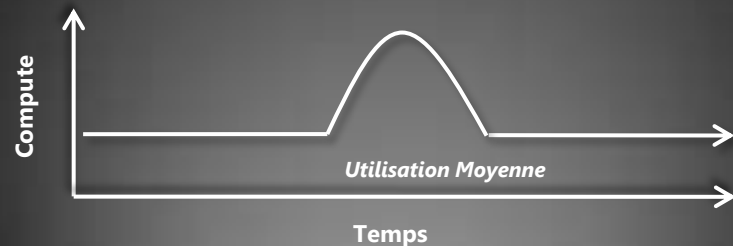
- Travaux ponctuels, par exemple batch
- La capacité excédentaire est inutilisée
- Time to market peut être ralenti

## "Croissance Rapide"



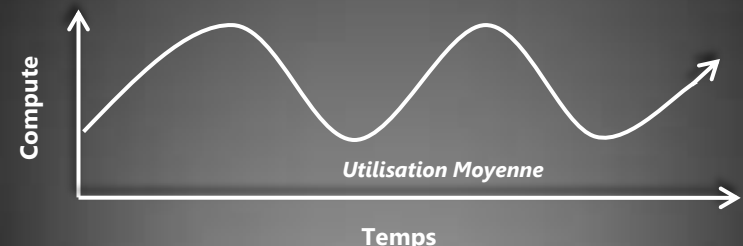
- Service populaire = montée en charge
- Tenir le rythme de croissance challenge l'IT
- Complexité du déploiement initial

## "Pics Imprévisibles"



- Pics de demande difficiles à prévoir
- Les pics impactent les performances
- Impossible de prévoir les cas extrêmes

## "Pics Prévisibles"



- Services à forte saisonnalité
- Pics dus à une augmentation de la demande
- Complexité IT et capacité gâchée

# Attention : le modèle de développement change avec un PaaS

- Un PaaS/cloud offre des dizaines de machines en parallèle  
Vs un super serveur
- Prix est fonction du temps de calcul & du stockage
- La notion de multi-tenant
- La délégation de traitement (email, images, search, ...)
- NoSQL & BigData
- ...



# Conclusion

- ✓ Choisissez votre PaaS et allez y

# Questions - Réponses