

# DevOps illustré : La jungle de la configuration d'une application

par Dimitri Baeli et Gilles Duguglielmo  
@dbaeli & @gdigugli



# Speakers

**@dbaeli**

- **Chien de berger Agile pour**



- **VP Quality eXo Platform**
- **R&D Prima-Solutions**

**@gdigugli**

- **Développeur java depuis 1999**
- **Architecte pour**
- **ILOG - IBM**
  - ✓ **librairie graphique 2D**
  - ✓ **moteur de règles**
- **Prima-Solutions**
  - ✓ **plate-forme de services pour J2EE**
  - ✓ **code génération de modèle métier**



# La jungle DevOps

- ❑ La configuration à chaud d'une application est complexe et difficile à maîtriser
- ❑ La configuration statique est très commune dans les systèmes d'exploitation
- ❑ La gouvernance de la configuration des applications est assez mal maîtrisée
- ❑ XML est un langage très pénible pour configurer une application depuis une console



# Des imbroglios DevOps:

## le changement d'un paramètre



**L'url du webservice de paiement change la nuit prochaine. Comment je change ça sur l'application en production ?**

**Est-ce qu'on peut changer ça dans le script de lancement ?**



# Des imbroglios DevOps:

## le changement d'un paramètre



**1** Tu mets à jour le paramétrage dans le fichier `web.xml` à l'intérieur du **WAR** et tu redémarre l'application

**3** On doit faire un build de l'application avec la nouvelle URL



**2** La valeur est dans la table **CONFIG\_SETTINGS**, édite la valeur de la clé `ws_url`



# Des imbroglios DevOps:

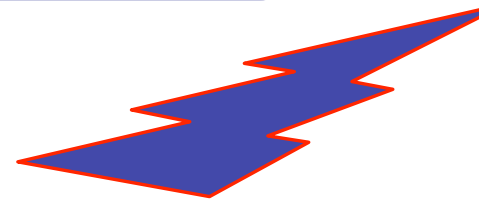
## le changement d'un paramètre

**Est-ce qu'il y a un outil pour faire ça ?**

**Je n'ai pas accès à la base de donnée, il faut appeler le DBA**

**Est-ce qu'on peut changer ça avec un script sh**

**Comment j'édite le fichier web.xml ?**



# Des imbroglios DevOps:

## la gouvernance de la configuration



**Quels sont les paramètres de l'application et leurs valeurs de production ?**

**Quelles variables de JVM sont utilisées par notre application ?**



# Des imbroglios DevOps:

## la gouvernance de la configuration



**Je vais demander à l'équipe de développement et vous transmettre une feuille Excel**

**Je vais faire un grep dans le code source et chercher les occurrences de `System.getProperty()`**





# Qu'est-ce qu'on peut faire ?

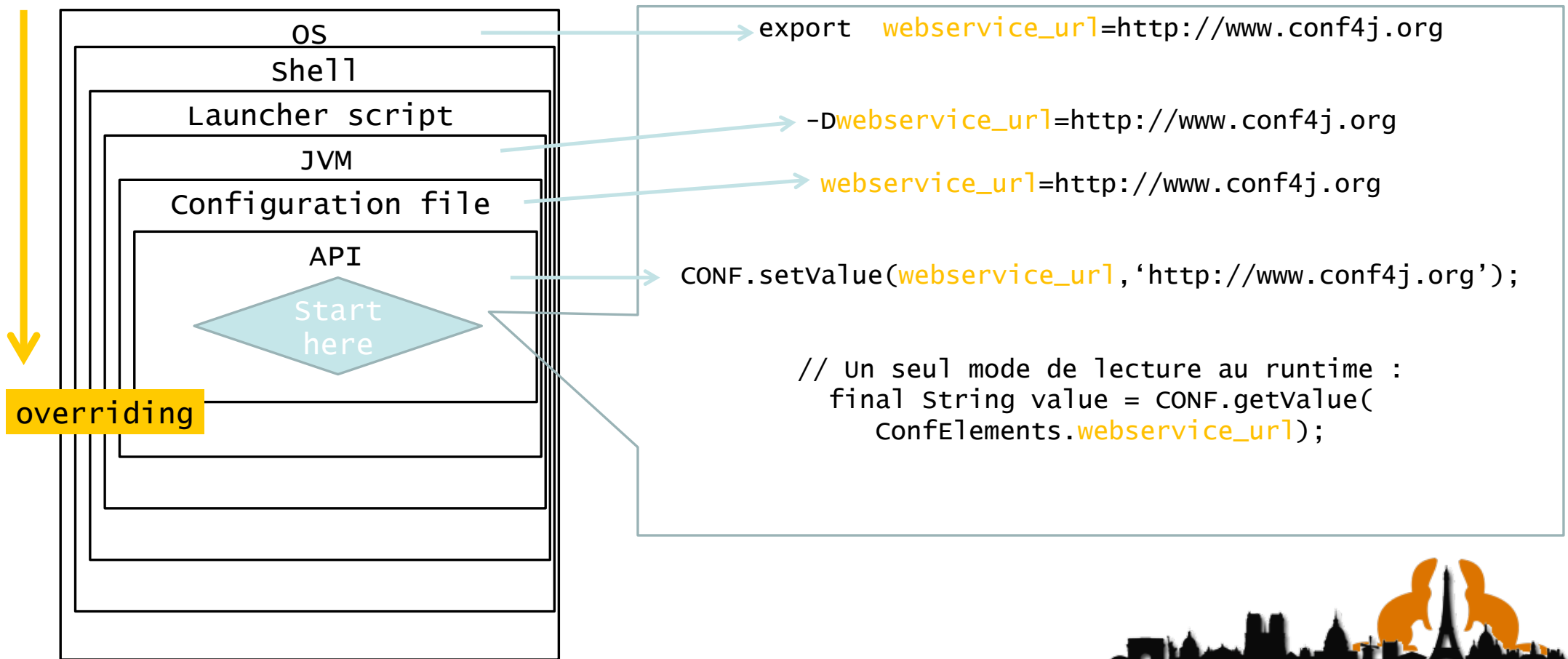
- ☐ Expliciter la configuration dans le code
- ☐ Rendre les variables de configuration aussi simple à manipuler que des variables de classes
- ☐ Détecter les changements dans le code lors de la mise en production (MeP) d'une nouvelle version
- ☐ Fournir de la documentation à jour pour chaque MeP
- ☐ Utiliser les mécanismes standards de l'OS pour faire de la configuration



# Action !



# Les couches de configuration



# La gouvernance des paramètres

```
public class ConfElements {  
  
    @Conf4j(value = "http://localhost",//  
    description = "the base url for devoxx website", //  
    scope = { webapp, unit_test },//  
    devPurposeOnly = true)  
    public static String devoxx_base_url;  
  
    @Conf4j(value = "${devoxx_base_url}/display/FR12/Accueil",//  
    description = "the Accueil url for devoxx website",//  
    scope = { webapp, unit_test })  
    public static String devoxx_home_url;  
  
    @Conf4j(value = "${devoxx_base_url}/display/FR12/Agenda",//  
    description = "the Agenda url for devoxx website", //  
    scope = { webapp, unit_test })  
    public static String devoxx_agenda_url;  
  
    @Conf4j(value = "${devoxx_base_url}/display/FR12/UnitTest",//  
    scope = unit_test,//  
    description = "the UnitTest url for devoxx website")  
    public static String devoxx_unittest_url;  
}
```

- Utiliser des champs dans une classe unique
- Annoter les champs
- Les clefs sont injectées



# Scope d'un paramètre

```
public class ConfElements {  
    @Conf4j(scope = { webapp, unit_test },//  
    devPurposeOnly = true)  
    public static String devoxx_base_url;  
    @Conf4j(scope = { webapp, unit_test })  
    public static String devoxx_home_url;  
    @Conf4j(scope = { webapp, unit_test })  
    public static String devoxx_agenda_url;  
    @Conf4j(scope = unit_test)  
    public static String devoxx_unittest_url;  
}
```

	webapp	unit_test
devoxx_base_url	X	X
devoxx_home_url	X	X
devoxx_agenda_url	X	X
devoxx_unittest_url		X

**scope** déclare l'utilisation de chaque variable lors de son exécution.

➤ Vérification au runtime

**devPurposeOnly** indique que ce paramètre doit être changé lors de la MeP.

➤ Export d'un template de configuration pour la MeP



# Accès à l'environnement

```
public interface ConfService {  
    String[] getMultiValues(String key);  
  
    List<String> getKeys();  
  
    String getValue(String key);  
  
    boolean getBooleanValue(String key);  
  
    int getIntegerValue(String key, int valueIfError);  
  
    String setValue(String key, String value);  
  
    void dumpConf(PrintStream os, boolean filter);  
  
    void checkScope(PrintStream os) throws IOException;  
  
    void checkUnused(PrintStream os) throws IOException;  
  
    void initFolders();  
}
```

➤ Semblable à  
System.getProperty()



# Démo



# Ceci n'est pas une librairie Just Fork it !



<https://github.com/dbaeli/conf4j>

