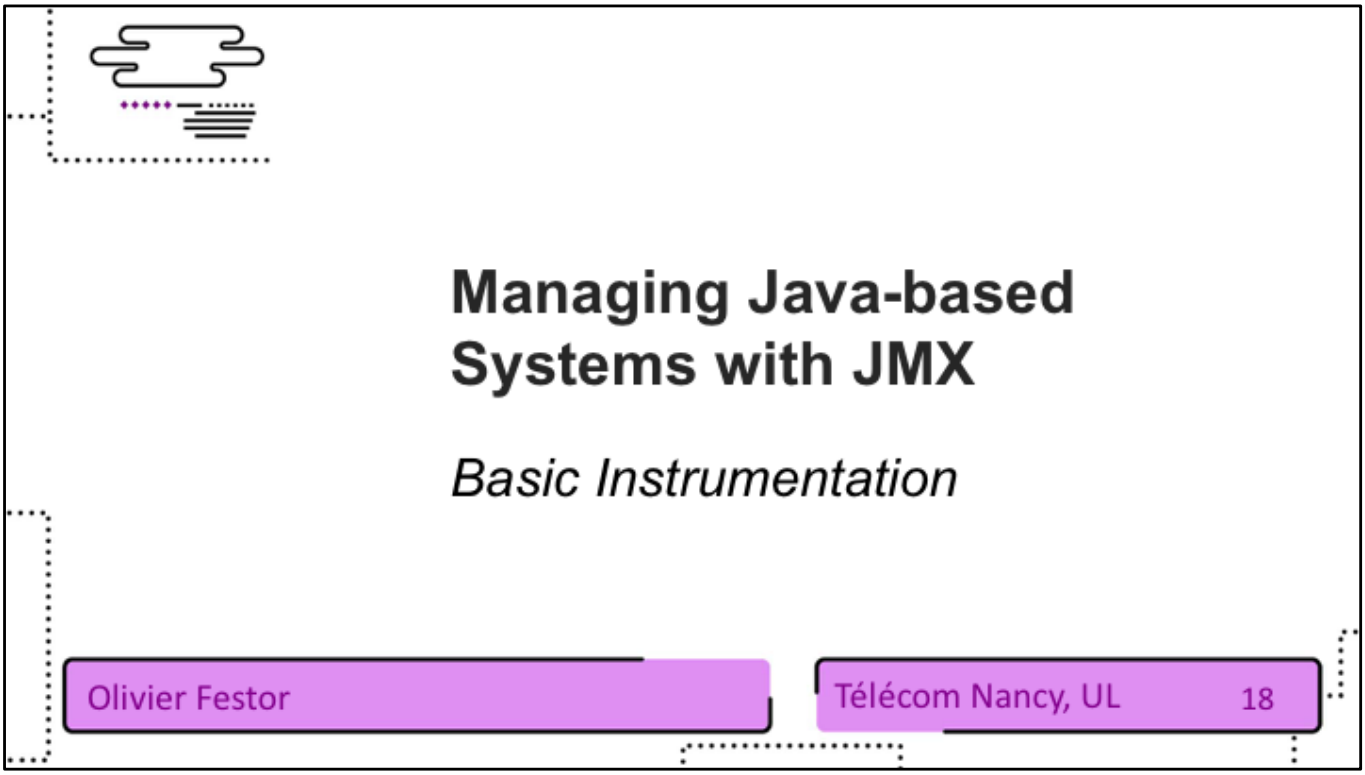
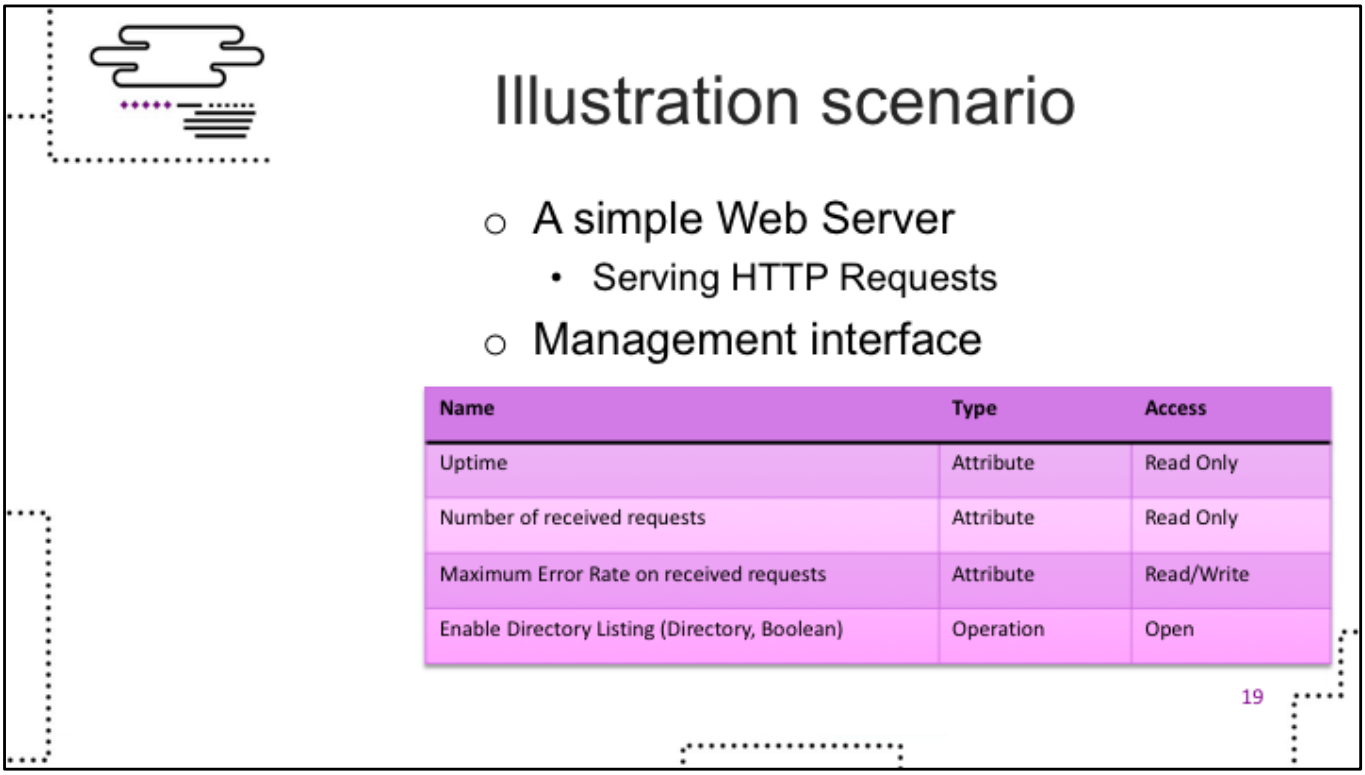


Cette leçon est consacrée à la couche d'instrumentation d'une application à l'aide de JMX.
A partir d'un exemple simple d'application, nous détaillerons les étapes de l'instrumentation partant de la construction d'un objet de supervision à son exploitation à distance en passant par son enregistrement dans un agent.

Il existe plusieurs types d'objets de supervision. Ils seront brièvement présentés dans cette leçon. Seul le modèle standard sera développé dans un premier temps.

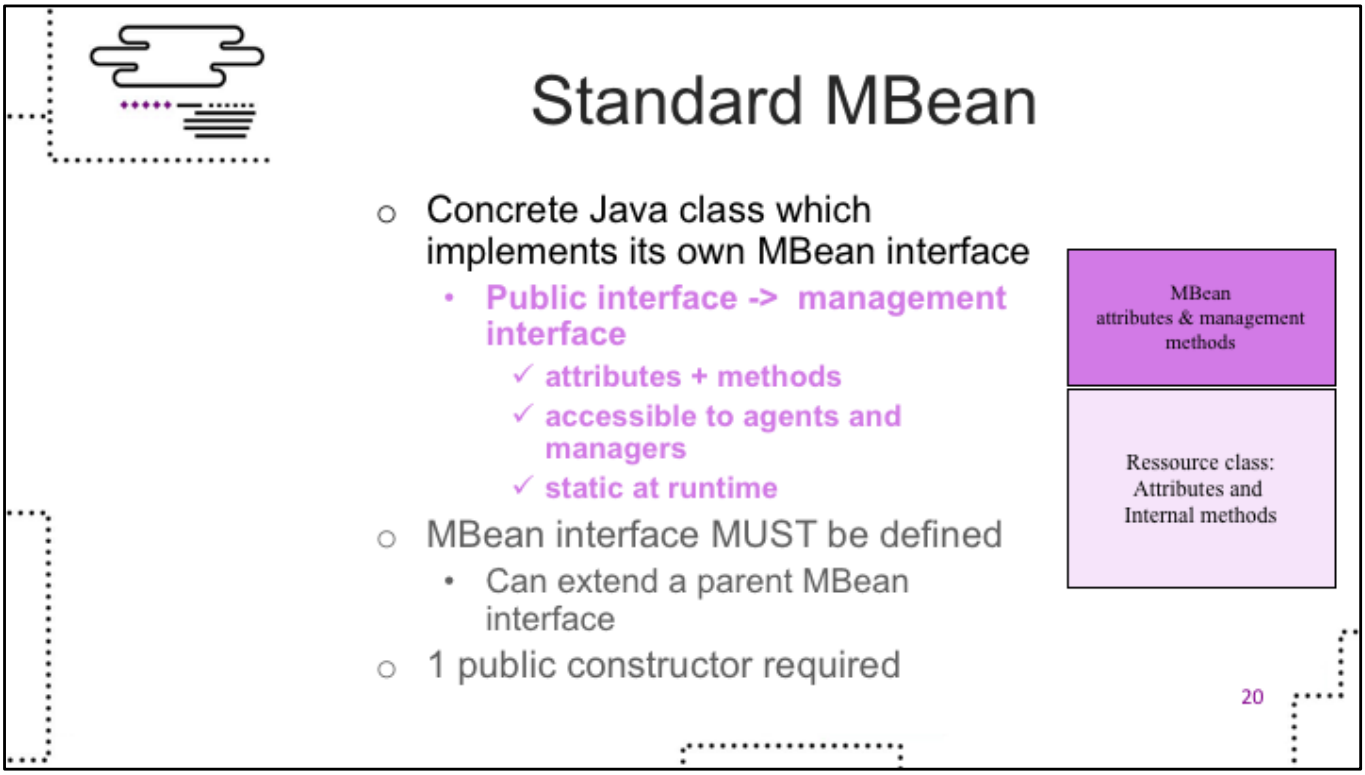


L'application qui va nous intéresser est un serveur web basique dont la fonction principale est de répondre à des demandes de renvoi de pages web à partir d'une requête HTTP Get. Notre serveur est implémenté en Java. Vous pouvez réaliser votre propre instrumentation en utilisant par exemple la distribution « nanohttp » comme base.

Notre cible d'instrumentation est la suivante : nous souhaitons pouvoir demander via la couche de supervision :

- L'*uptime* (le temps en secondes écoulé depuis le démarrage du serveur),
- Le nombre de requêtes HTTP reçues depuis le démarrage du serveur.

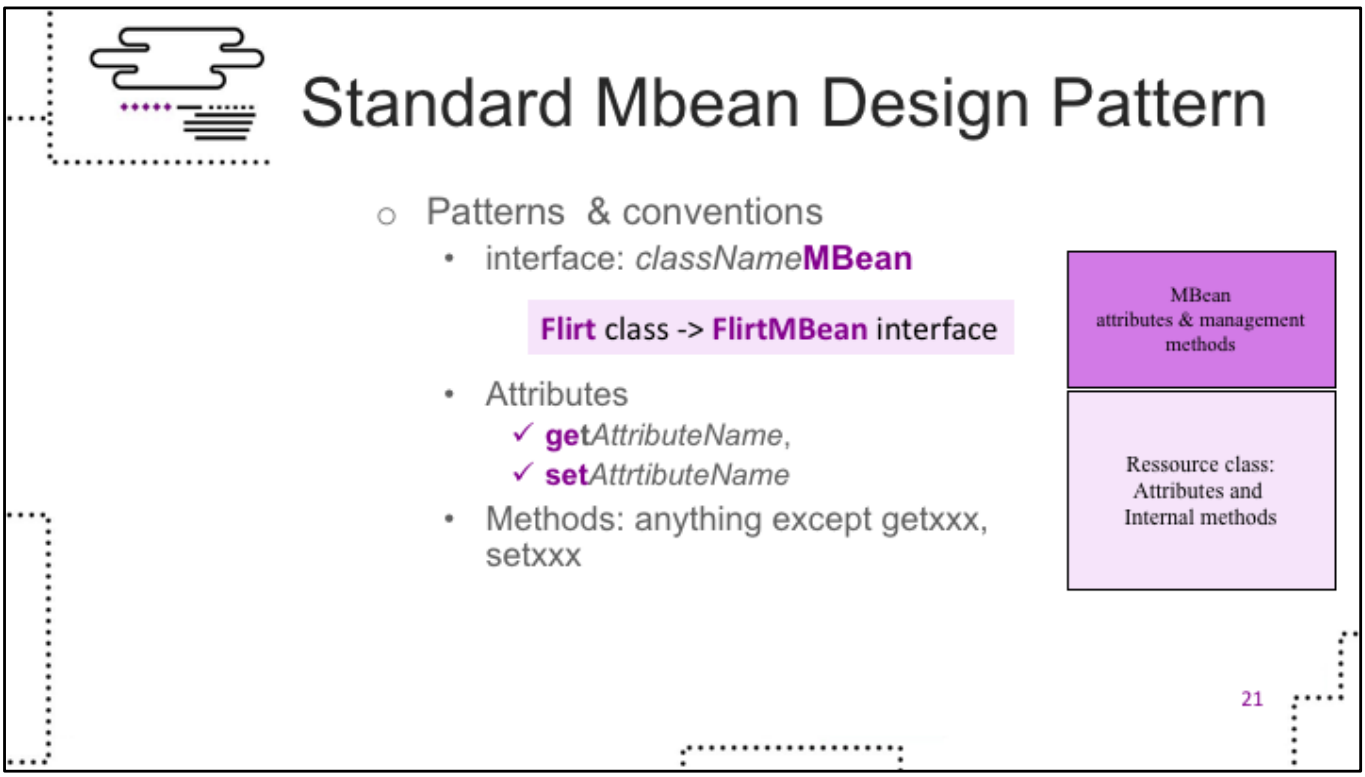
De plus nous souhaitons pouvoir fixer le taux maximal d'erreurs sur des requêtes (le dépassement donnera lieu à l'émission d'une notification qui sera par exemple interprétée comme une alarme)
Nous souhaitons également pouvoir autoriser, respectivement interdire la lecture de répertoires (lorsqu 'une URL pointe sur un répertoire qui ne contient pas de fichier index.html)



Pour réaliser notre instrumentation, nous allons utiliser, dans un premier temps, le modèle le plus simple : le Mbean standard.
Un Mbean standard est composé de deux éléments :

- Une interface obligatoirement publique qui définit les éléments de l'interface de supervision plus
- Une classe java concrète qui implémente cette interface et qui lie les opérations de supervision aux opérations de l'application.

L'interface de supervision décrit les attributs de supervision ainsi que les opérations autorisées au travers de l'interface de supervision. Tous les attributs et opérations définis dans cette interface deviennent visibles sur le système de supervision. Cette interface est définie avant la compilation de la classe et avant chargement dans la machine virtuelle. Elle est statique à l'exécution.

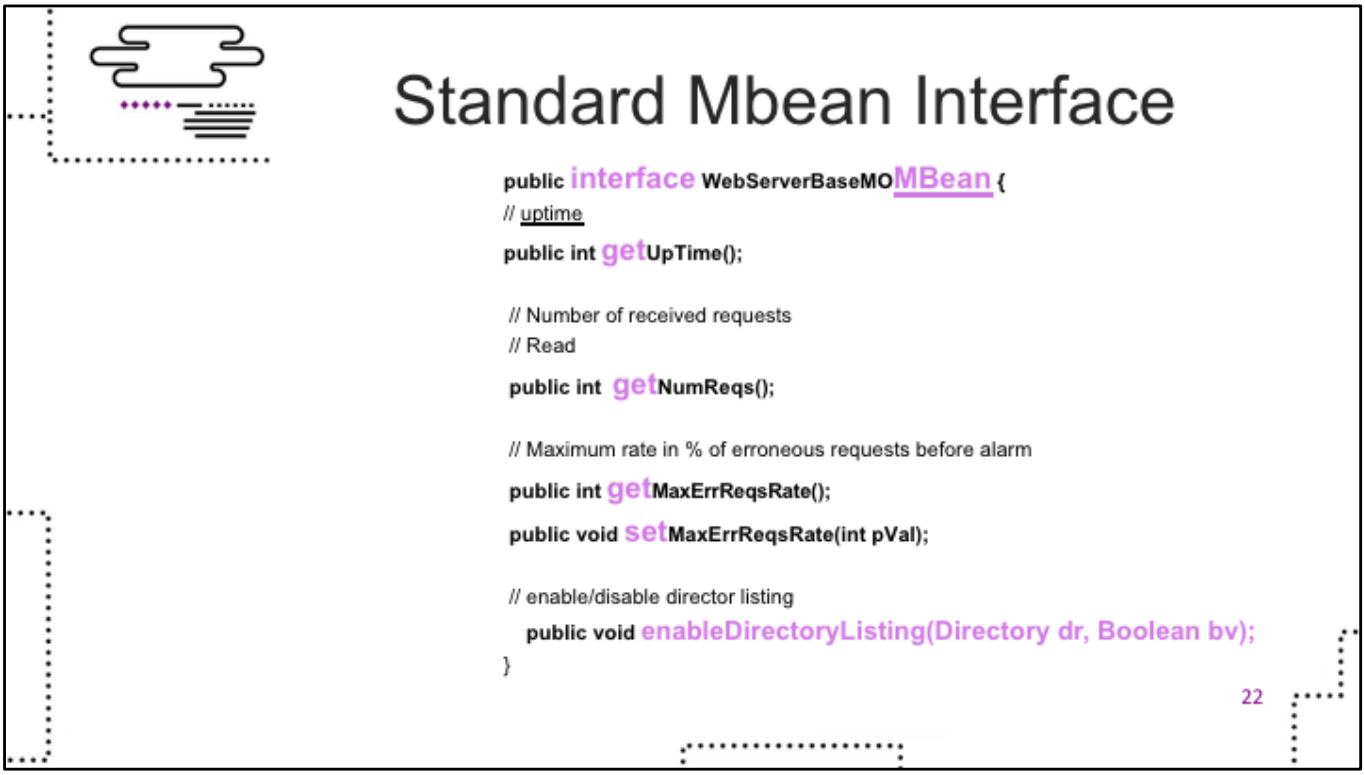


Définir un Mbean standard nécessite le respect d'un ensemble de patrons de conception.
Dans un premier temps, tout objet de supervision doit, s'il souhaite être visible par l'agent, définir une interface. Il doit également offrir un constructeur public pour être instanciable.

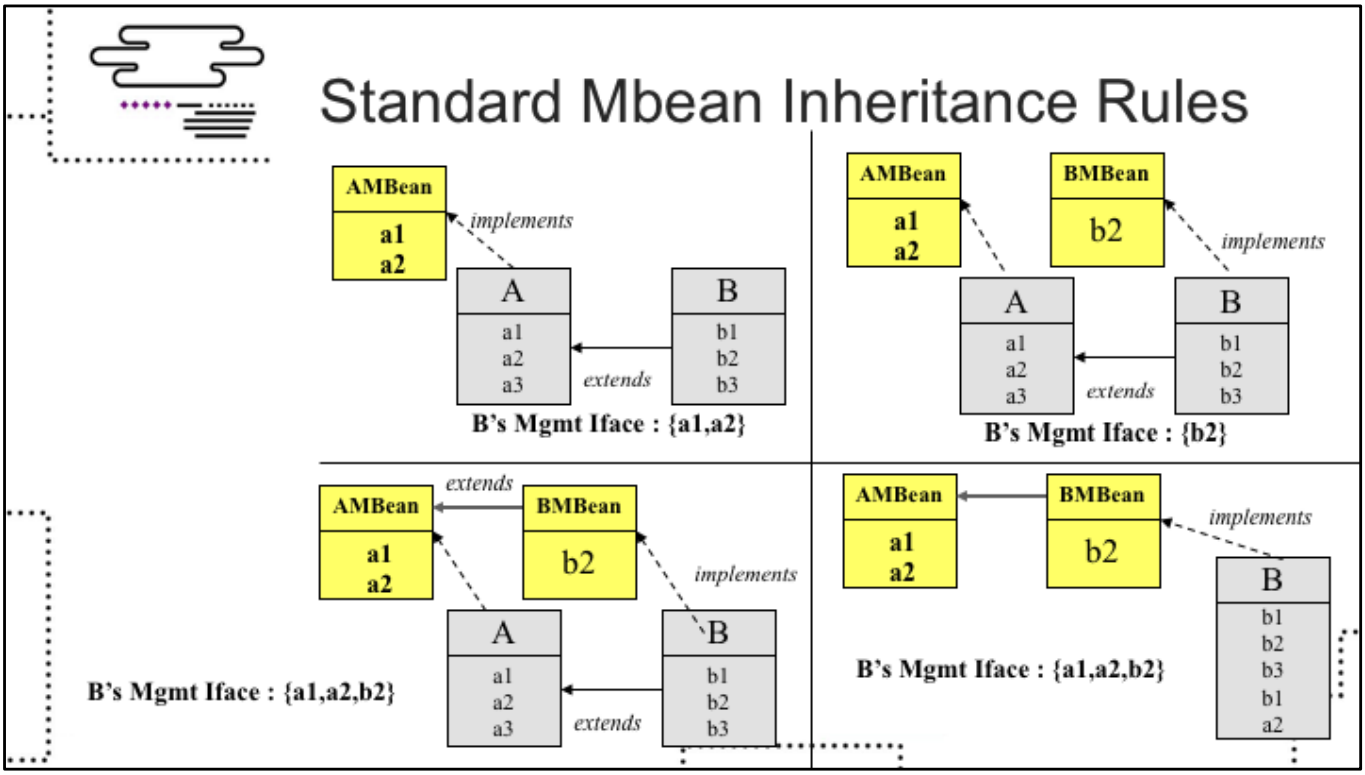
L'interface de supervision est extraite par l'agent au travers du Mbean Server par introspection de la classe de l'objet de supervision. Pour identifier quelle interface est celle de supervision, le serveur de Mbeans va rechercher dans la définition de la classe une interface qui porte le même nom que celle-ci, suffixé de Mbean.

Si par exemple, la classe de mon Objet de Supervision s'appelle 'Flirt', le Mbean Serveur va, au moment de l'enregistrement d'une instance de cet objet, rechercher une interface dénommée 'FlirtMBean', comme illustré dans la planche.

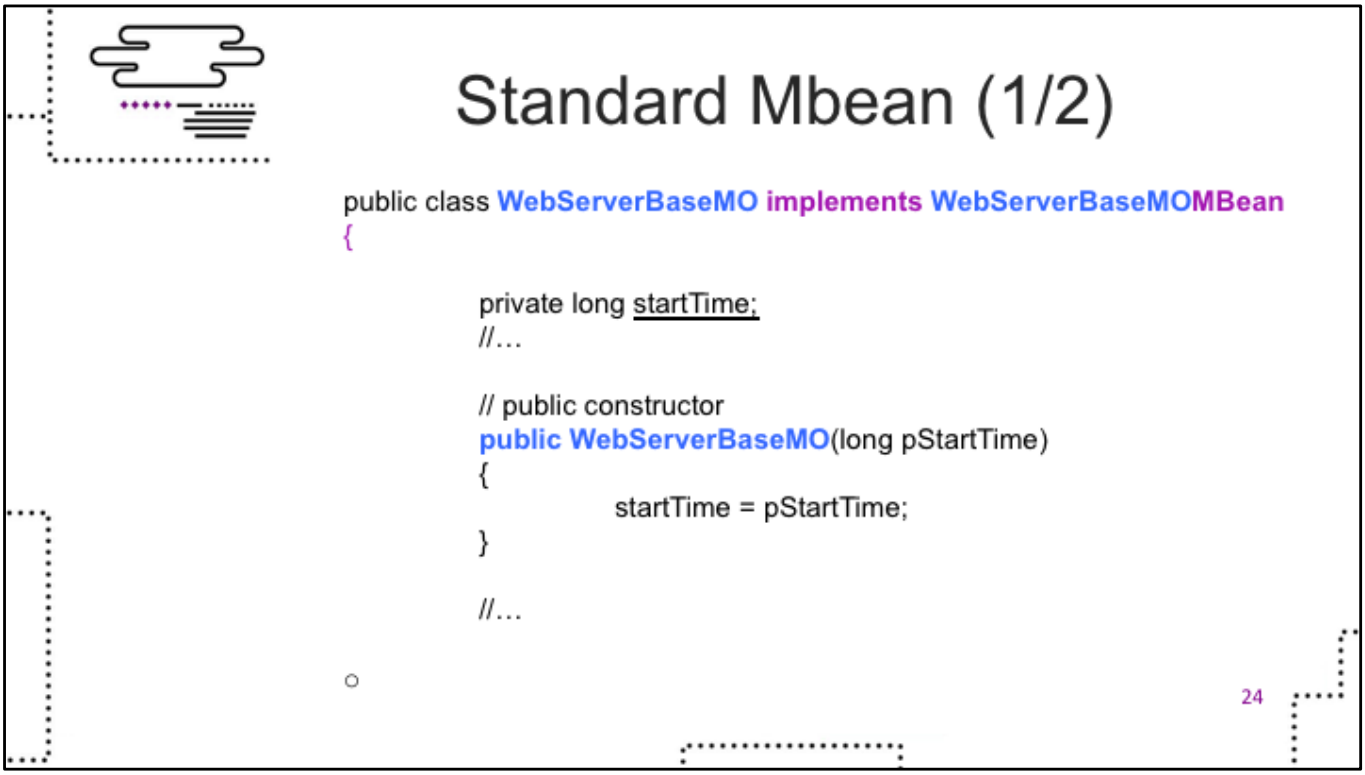
Dans cette interface, le serveur de Mbean va chercher à identifier les attributs et les méthodes. Dans l'interface, tous les éléments sont définis par des méthodes. Pour différencier une fonction d'un attribut, on utilise les préfixes des noms qui donnent en plus les droits sur les attributs. Par exemple, tout ce qui ne commence pas par *get* ou *set* est considéré comme une fonction.



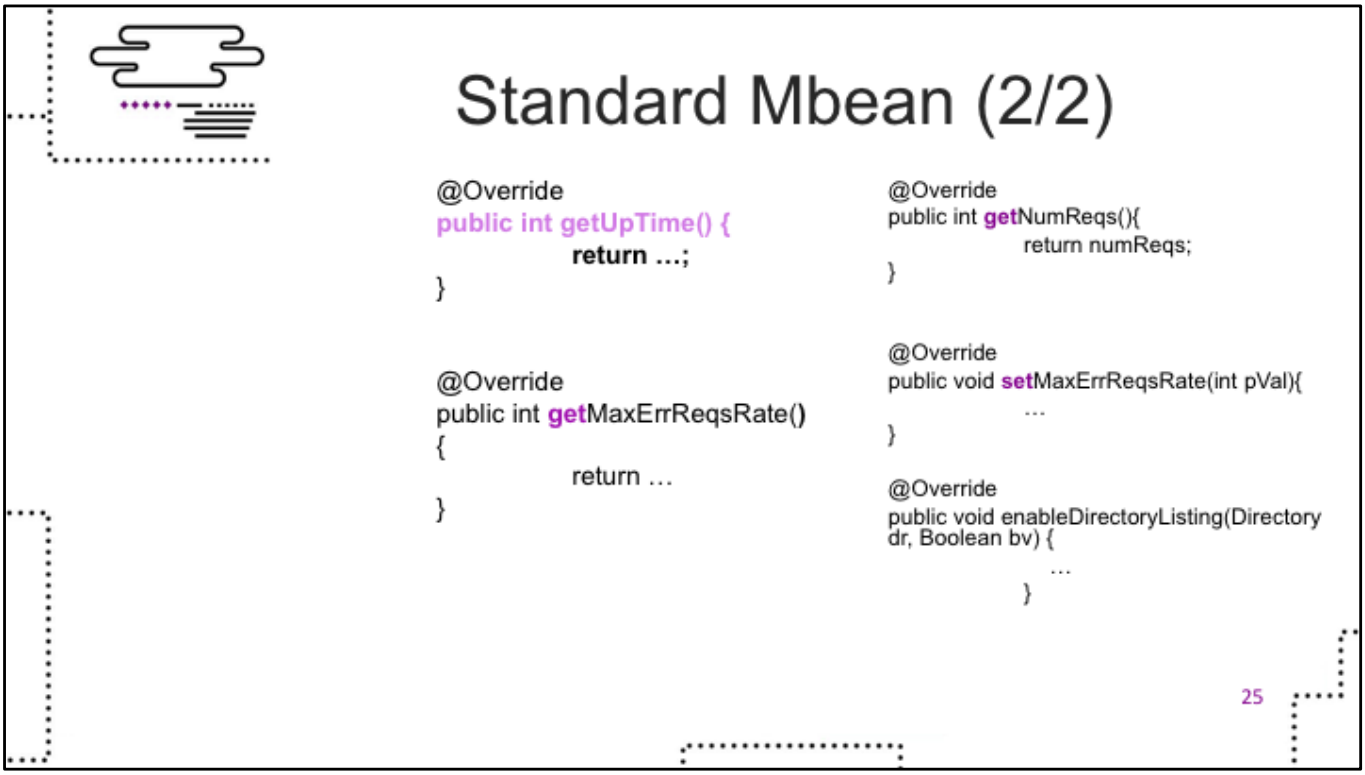
Sur notre objet de supervision, ceci donne l'interface telle que définie dans la planche Elle a bien un nom de classe (WebServerBaseMO). On y retrouve également les attributs au travers des fonctions de lecture et d'écriture : UpTime en lecture seule et NumReqs en lecture seule,MaxErrReqsRate en lecture et une méthode de gestion de l'autorisation d'énumération des contenus des répertoires ne comprenant pas le fichier « index.html ». Cette méthode a deux paramètres : le nom absolu du répertoire et un booléen qui modélise l'autorisation, respectivement 'l'interdiction de l'énumération.



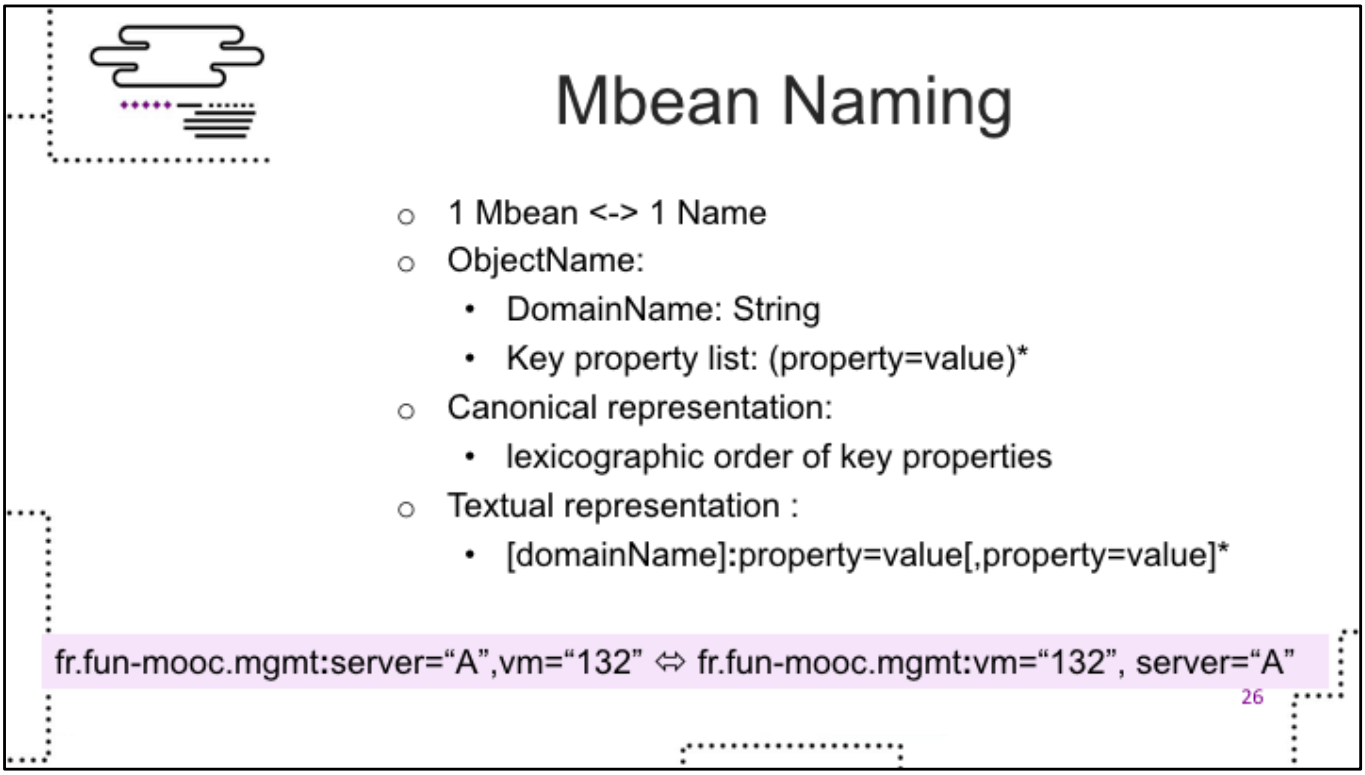
Les interfaces de supervision en JMX se composent comme suit: une classe d'objets de supervision hérite de l'interface de supervision de sa superclasse si et seulement si elle n'a aucune interface de supervision propre ou qu'il a une interface de supervision propre qui étend explicitement celle de sa super-classe (cas 1 en haut à gauche & 3 en bas à gauche).
Dans le cas contraire, son interface se limite à celle qu'elle implémente (cas 2 en haut à droite) . Celle-ci peut étendre une autre interface de supervision sans devoir hériter d'un objet qui l'implémente (cas 4 sur la planche).



Regardons maintenant comment ces interfaces sont à implémenter dans la classe de l'objet de supervision Le Mbean (ici WebServerBaseMO) va instancier l'interface de supervision. Son code a typiquement pour rôle de relier les objets et opérations déclarées dans l'interface de supervision aux objets et opérations de la partie fonctionnelle de l'application ou de la machine virtuelle. On retrouve bien dans l'objet géré le constructeur public exigé par le patron de conception des Mbeans.



On retrouve ensuite la définition des différentes méthodes qui représentent l'interface de supervision : les méthodes de lecture/écriture des attributs et les méthodes qui représentent des fonctions de supervision.

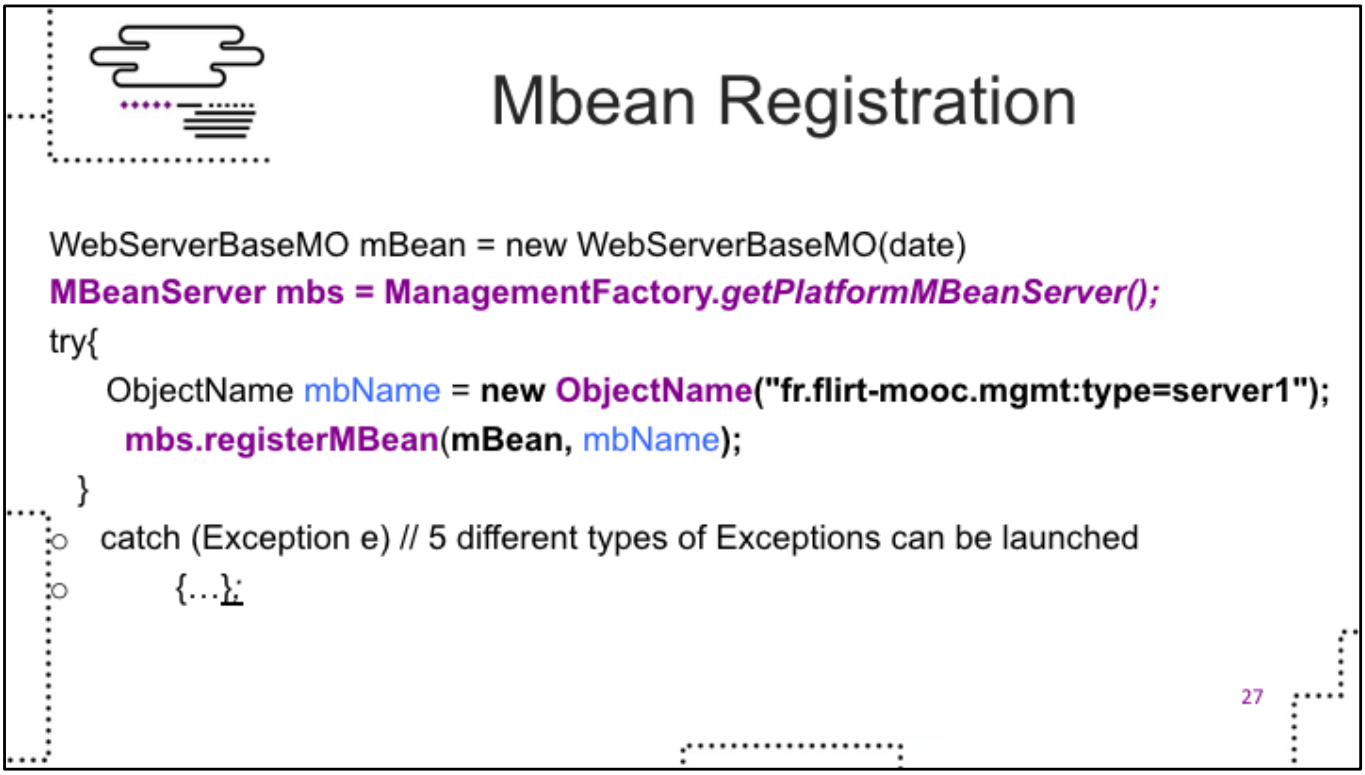


Le nommage est essentiel en JMX car c'est au travers de noms uniques que les objets de supervision seront visibles et accessibles via l'interface de supervision. Pour les nommer, JMX définit une structure appelée
ObjectName. Une instance d'*ObjectName* est associée à chaque instance de Mbean et l'identifie de façon unique.

Un nom est composé de deux parties :

- un préfixe de domaine (comme on les trouve dans des URL),
- une suite de couples <attribut, valeur>. La forme canonique suit l'ordre lexicographique des noms d'attributs.

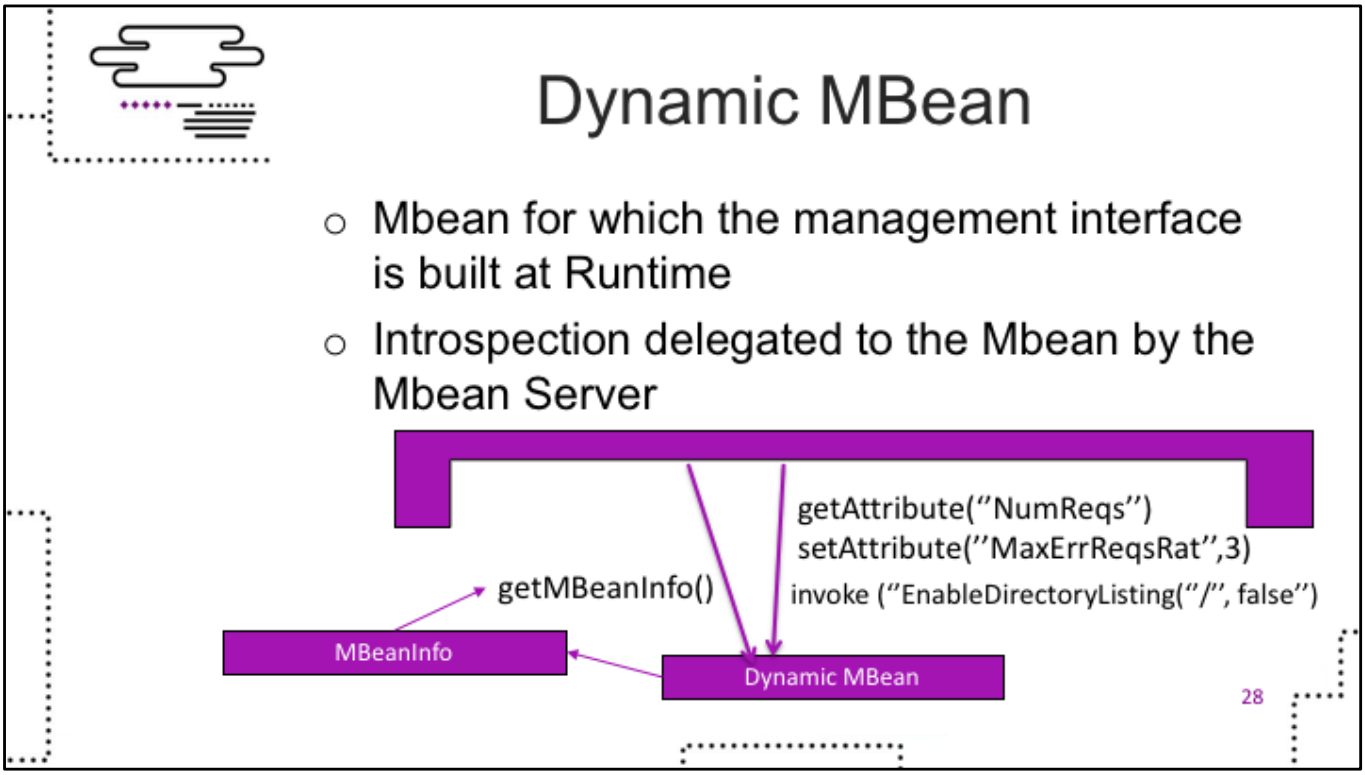
Chaque nom a une représentation textuelle, comme présenté dans l'exemple. En raison de l'ordre lexicographique, les deux noms présentés ci-dessous sont les mêmes.



En suivant le cheminement présenté, nous disposons désormais :

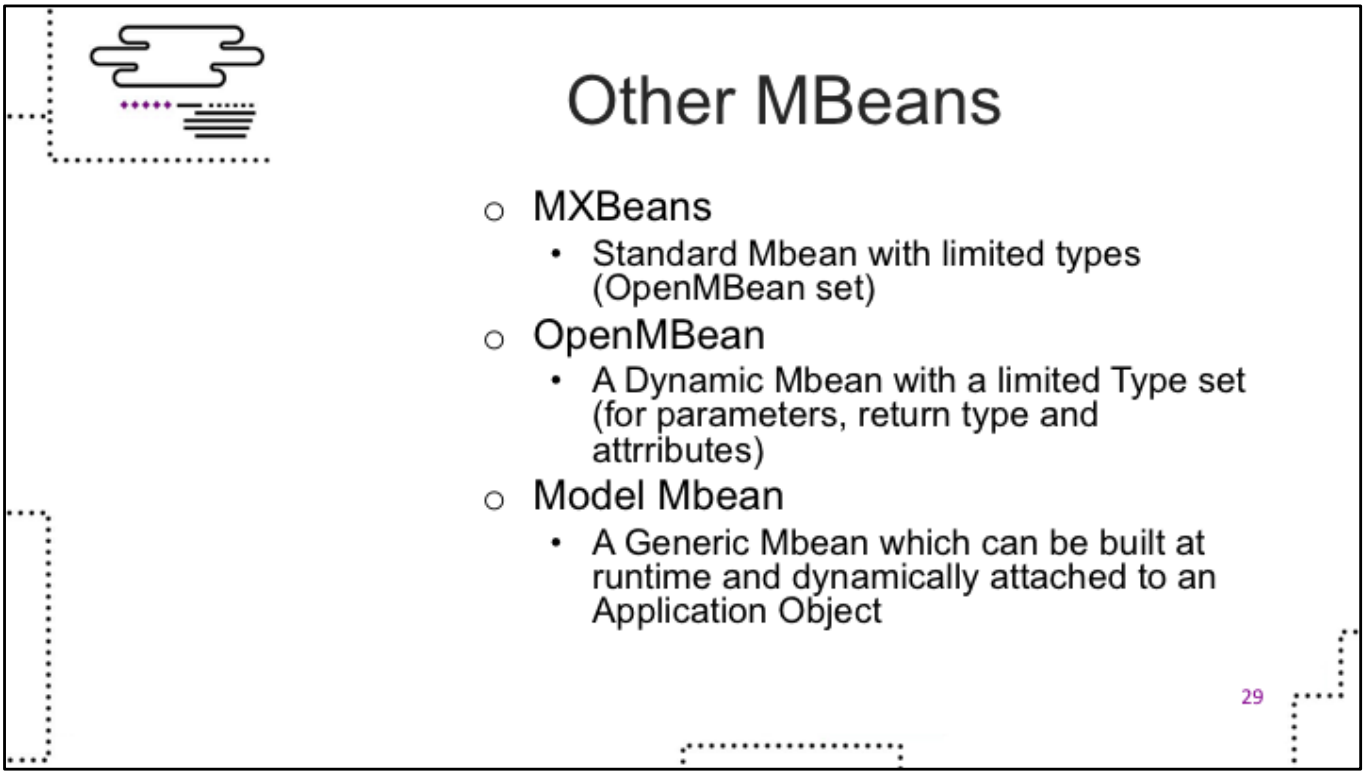
- d'une interface de supervision,
- d'un objet de supervision qui implémente cette interface (objet créé en ligne 1),
- d'Un serveur de supervision qu'il suffit de récupérer de la machine virtuelle actuelle (on peut en créer un soi-même également).

L'objet de supervision peut être associé à son nom et enregistré dans le serveur.
Une fois l'enregistrement fait, l'objet devient visible sur l'interface de supervision.



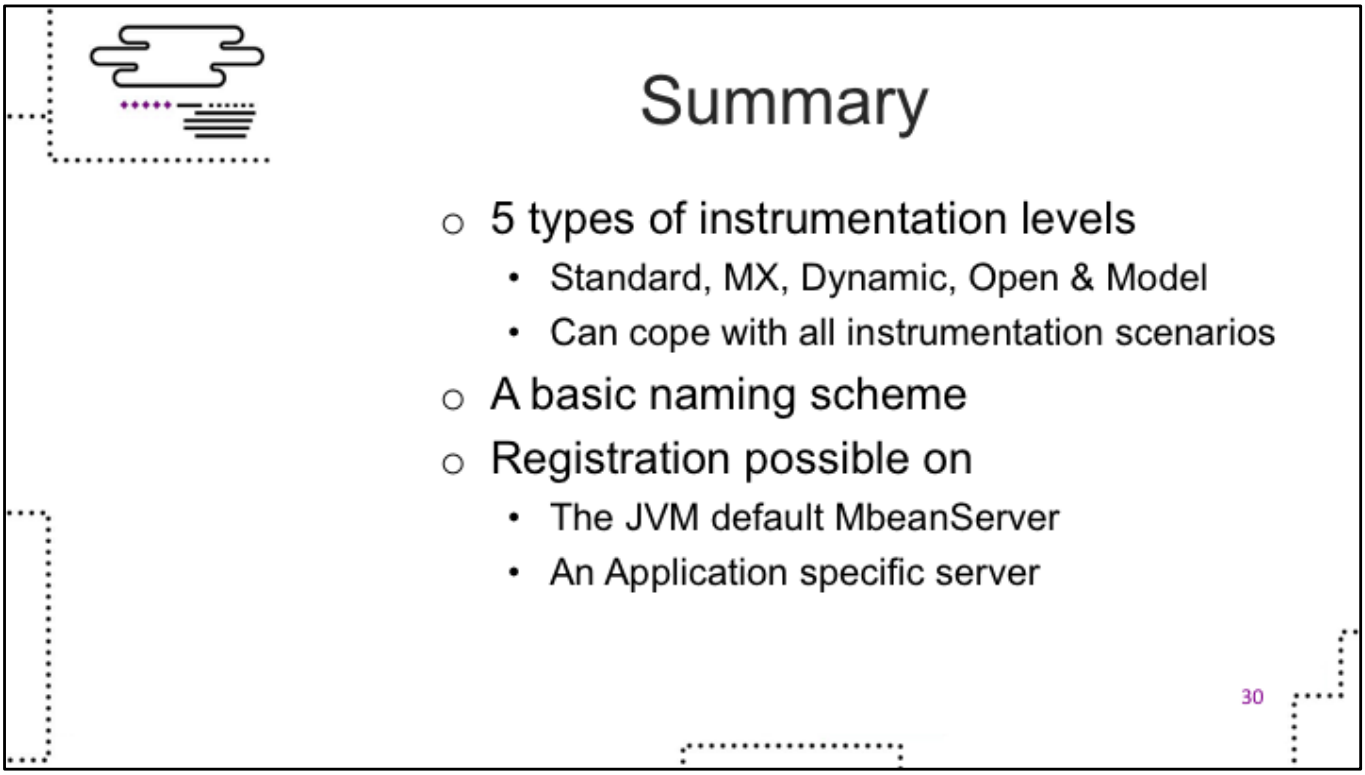
Dans un Mbean standard, l'interface de supervision est figée au moment de la compilation. Une fois l'interface compilée, l'objet va devoir servir les attributs qui y sont définis. Ceci peut poser un problème car certaines informations/méthodes de supervision ne peuvent être connues qu'à l'exécution (la présence ou pas d'un service particulier sur un système d'une fonction de mesure de la température par exemple ou l'existence de certains services de journalisation) .
Afin de permettre la composition dynamique à la volée de l'interface de supervision, JMX dispose du Mbean dynamique. Ce n'est plus l'agent qui construit l'interface de supervision et y accède par introspection mais l'objet de supervision lui-même. Pour cela, l'objet de supervision construit les méta-données sous la forme d'un Objet MBeanInfo. Ces méta-données (liste des attributs, méthodes) sont générées à l'exécution par l'objet de supervision.
La collecte des données de supervision est elle également déléguées de l'agent à l'objet de supervision par des opérations d'invocations indirectes comme la lecture ou l'affectation de variables, mais également l'invocation de méthodes (*enableDirectoryListing* dans l'exemple ici).

Pour celles et ceux qui ont vu les principes de la gestion OSI, le Mbean dynamique permet naturellement et simplement d'implémenter les packages conditionnels qui existent dans le formalisme GDMO et dans l'approche OSI de supervision, aujourd'hui pratiquement disparue.



Le MBean dynamique et le Mbean standard forment les principaux modèles de Mbean. Il en existe cependant 3 autres dont 2 sont des sous-classes des deux principaux. Le MBean est un objet de supervision qui suit le modèle de Mbean standard mais dont les types des attributs et des paramètres sont restreints à un sous-ensemble (chaînes de caractères, entiers, booléens, Composite, …) De même, le « Mbean Open » est un Mbean dynamique dont les attributs et paramètres sont limités au même sous ensemble de types que le MXBean. Ceci permet de réaliser des interfaces de superivsion (côté application) générique vu qu'elles ne traitent que d'un sous-ensemble de données.

Le dernier type de Mbean est le Mbean modèle. Ce dernier ajoute un niveau de généricité au-delà du Mbean dynamique en permettant de créer des Mbean indépendants des objets de supervision et de les attacher directement à des objets de supervision à l'exécution. On peut ainsi créer une bibliothèque d'objets de supervision génériques et les lier à la demande à des applications.



Dans JMX, tout est mis en œuvre pour faciliter grandement l'instrumentation des applications. Le concepteur dispose de 5 modalités pour réaliser les objets d'instrumentation de son système. Ces objets peuvent ensuite être mis en œuvre suivant les patrons de déploiement vus dans la leçon dédiée à l'architecture JMX.

Le nommage des objets est basique et composé de deux parties : un préfixe et un ensemble de couples <attribut,valeur>.

Un objet de supervision créé est finalement enregistré dans le système de supervision et devient "supervisable" au travers du serveur de Mbeans.