

Spring Boot Compléments

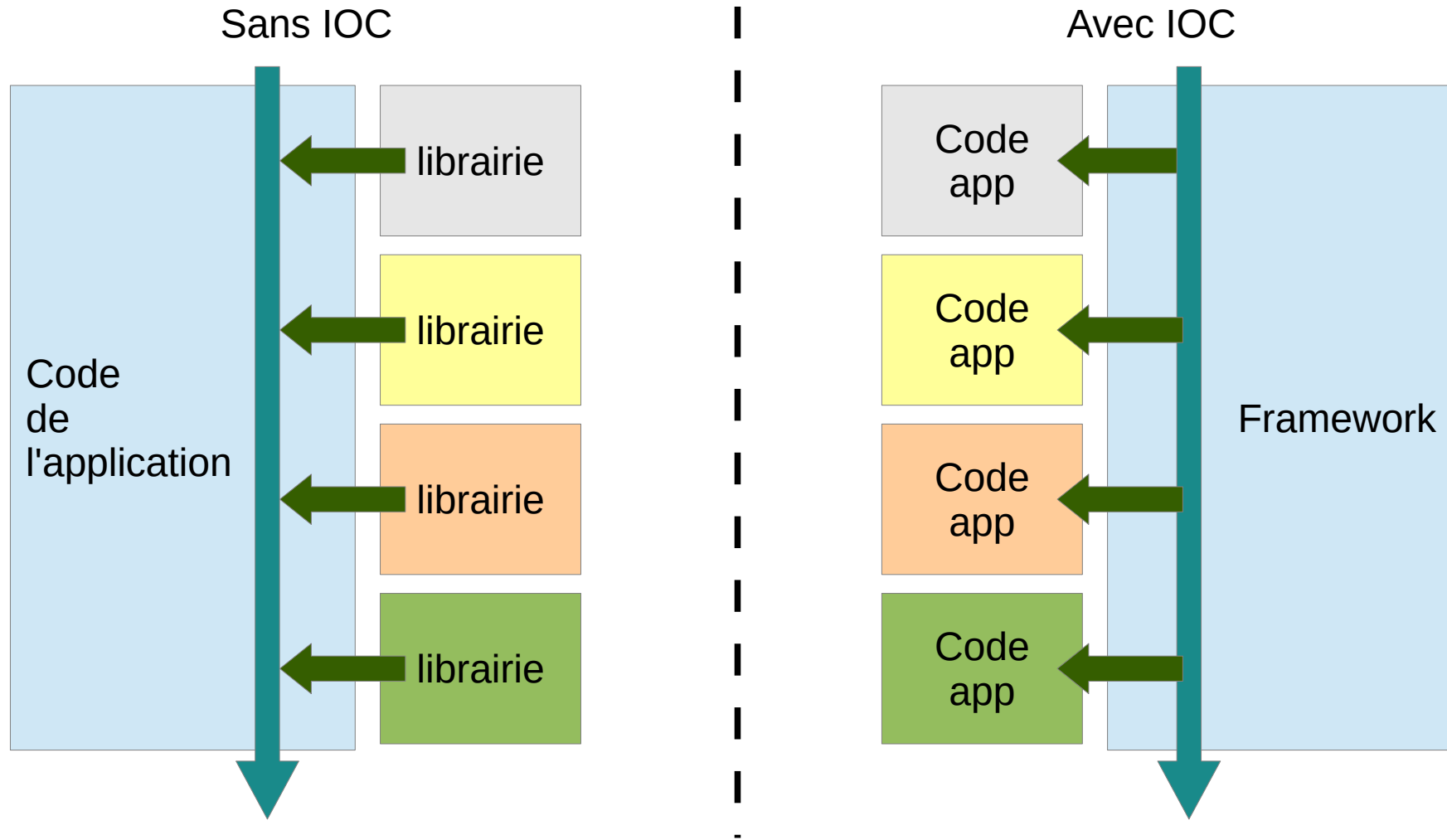
Scheduling

Spring MVC et templating

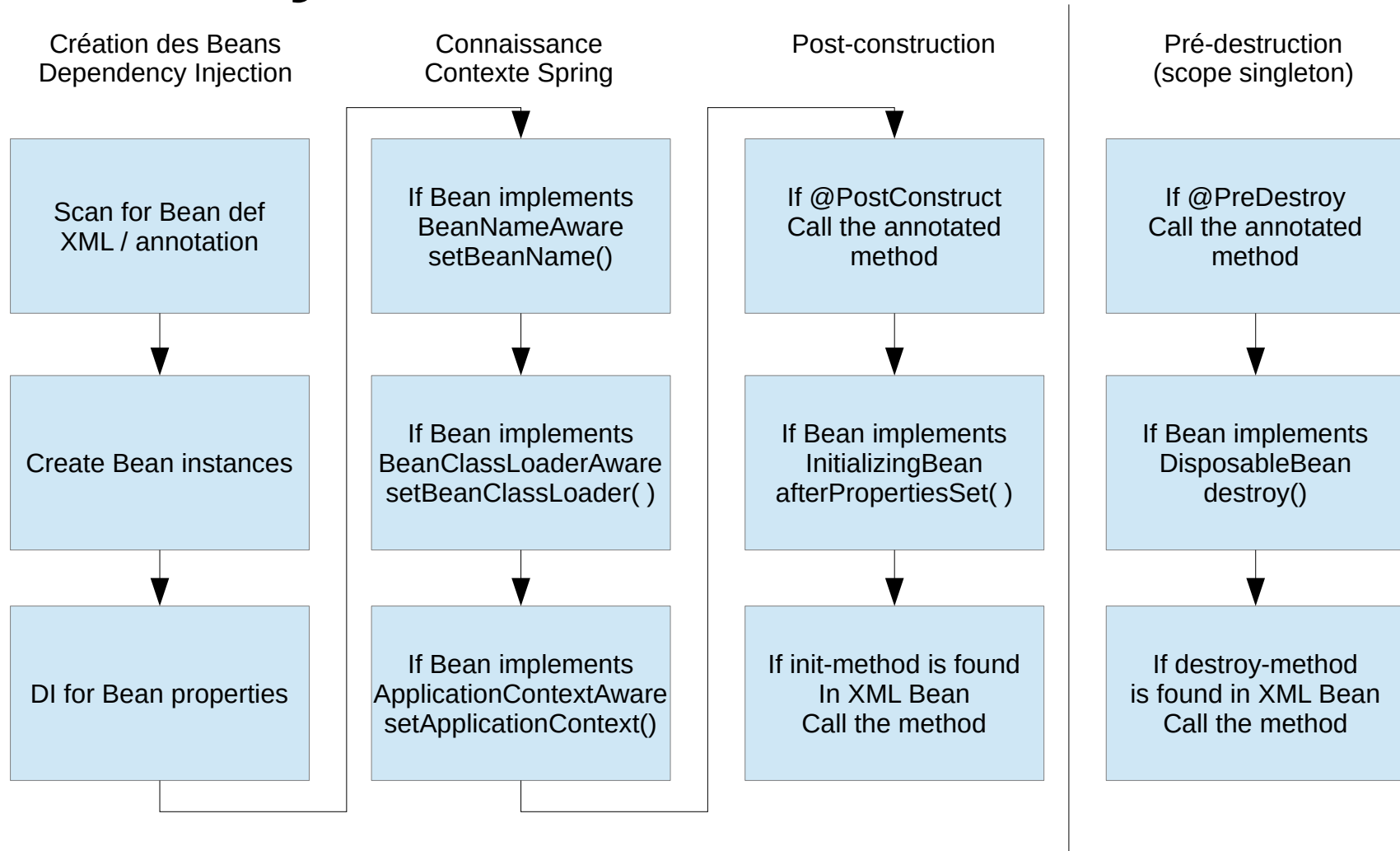
Sommaire

- Compléments
- Scheduling : des traitements à intervalles régulier
- Ajouter des pages web avec du templating

Compléments



Compléments : Cycle de vie des beans



Compléments :

Spring.io – projets à suivre

- Spring for GraphQL (le successeur de REST?)
- Spring AI : très intéressant
 - Intégration de l'api de la plupart des chatbots du moment...
 - Pas encore en version 1 : va évoluer rapidement
 - Utilisable dès à présent dans vos projets
 - Exemple : assistant rédaction fiche produit dans ERP

Scheduling

- Sur un projet SpringBoot vierge :
 - Activer le scheduling avec `@EnableScheduling`
 - Positionner des « cron » sur vos méthodes de services avec `@Scheduled(cron = "* * * * *")`
 - Un fois lancé le projet est en attente : pas de serveur web mais l'application reste bloquée dans le thread principal
- TP : Afficher un tick toutes les secondes dans la console.

Spring MVC

- Sur un projet SpringBoot vierge :
 - Ajouter la dépendance « Spring Web »
 - Ajouter la dépendance « Thymeleaf »
 - On dirige la page d'accueil en créant un @Controller et en redirigeant vers /index par exemple.
 - Un controller peut retourner un « ModelAndView » ou une chaîne de caractère (page html du template)
 - Un model est une simple Map<String,Object>
 - Dans le template le model est accessible directement
- TP : Créer votre page d'accueil et une page listant les villes (« /towns »)
 - Pour les plus avancés : testez d'autres moteurs de templating (cf doc <https://docs.spring.io/spring-framework/reference/web/webmvc-view.html>)

TP Final : Mini projet

- En prenant une idée de votre choix
- Créez un projet Spring Boot from scratch
- Ajoutez les dépendances nécessaires
- Ajouter les couches applicatives (mvc, rest, dto, service, model, repository)
- Faire une « stack » complète pour une entité
- Livrez votre projet sur github