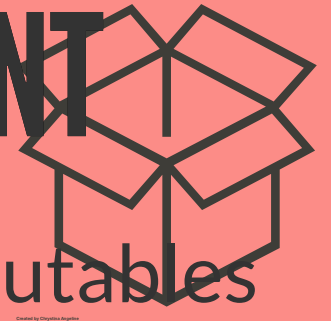


# LOG210 SÉANCE #12

## ANALYSE ET CONCEPTION DE LOGICIELS

1. Diagrammes de déploiement ← S20203
2. Diagrammes de composants
3. Révision diagrammes UML
4. Sondage sur technologies
5. Révision Observateur

# DIAGRAMMES DE DÉPLOIEMENT

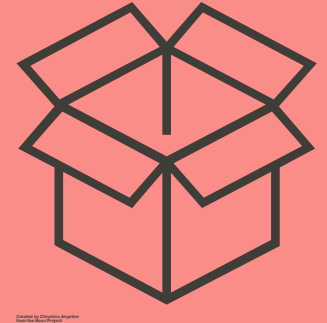


Documenter (1) comment les fichiers exécutables seront affectés sur les nœuds de traitement et (2) la communication entre composants physiques



- <http://wahlnetwork.com/2012/06/03/hp-discover-2012-targeted-sessions/>
- <http://gadgets.in.com/speck-pixelskin-hd-wrap-ipad-2-case.htm>
- [http://wiki.open.hr/wiki/Samsung\\_I7500\\_\(Samsung\\_Galaxy\)](http://wiki.open.hr/wiki/Samsung_I7500_(Samsung_Galaxy))

# TYPES DE NŒUDS



## Nœud physique (équipement)

Ressource de traitement physique (p.ex. de l'électronique numérique), dotée de services de traitement et de mémoire destinés à exécuter un logiciel. Ordinateur classique, cellulaire, etc.

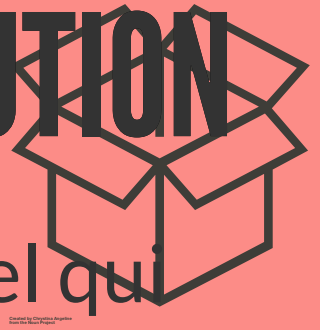
# TYPES DE NŒUDS



## Nœud d'environnement d'exécution (EEN *execution environment node*)

Ressource de traitement logiciel qui s'exécute au sein d'un nœud externe (comme un ordinateur) et offrant lui-même un service pour héberger et exécuter d'autres logiciels.

# NŒUD D'ENVIRONNEMENT EXÉCUTION



- Système d'exploitation (OS) est un logiciel qui héberge et qui exécute des programmes
- Machine virtuelle (JVM ou .NET)
- Moteur de base de données (p.ex. PostgreSQL) exécute les requêtes SQL
- Navigateur Web héberge et exécute JavaScript, applets Flash/Java
- Moteur de workflow
- Conteneur de servlets ou conteneur d'EJB

# DIAGRAMME DE DÉPLOIEMENT

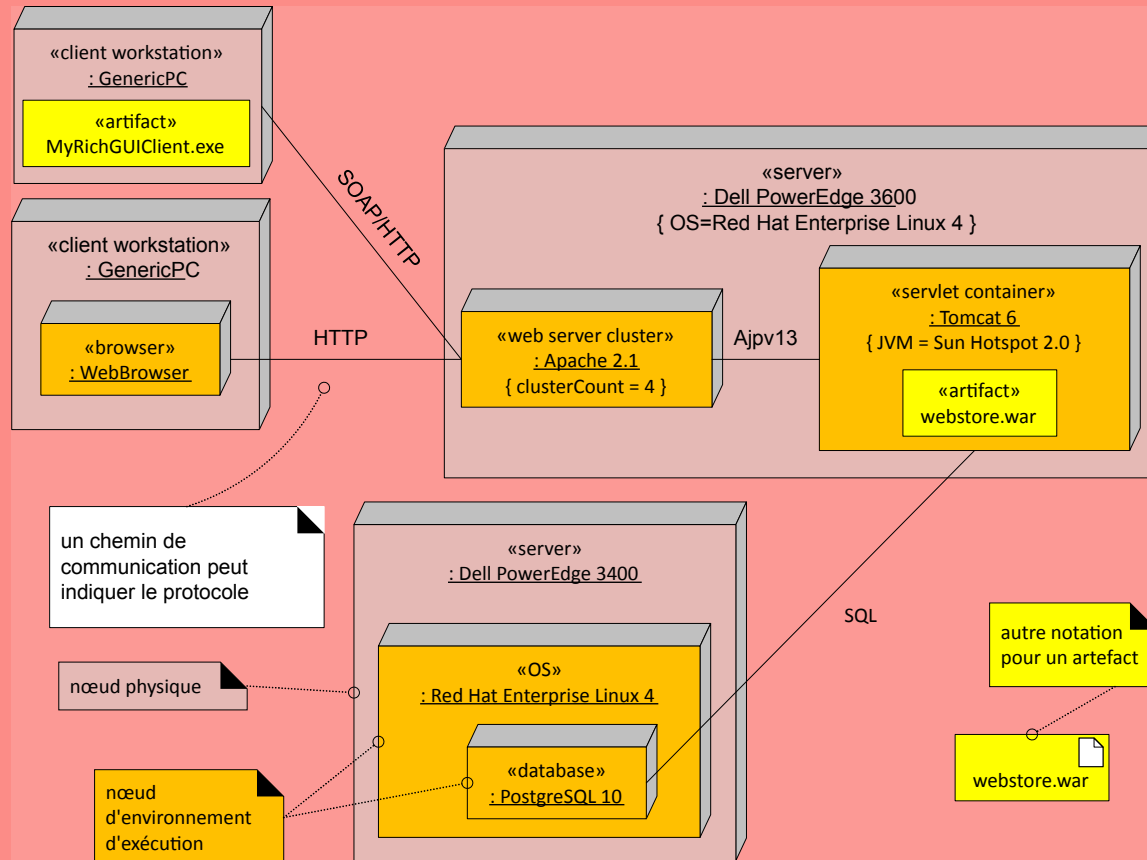
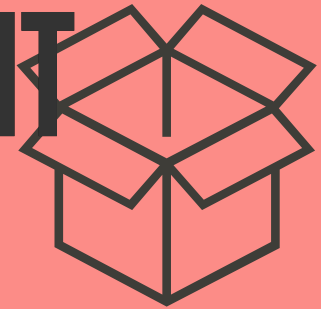


fig. F31.1, A37.1

# DIAGRAMME DE DÉPLOIEMENT

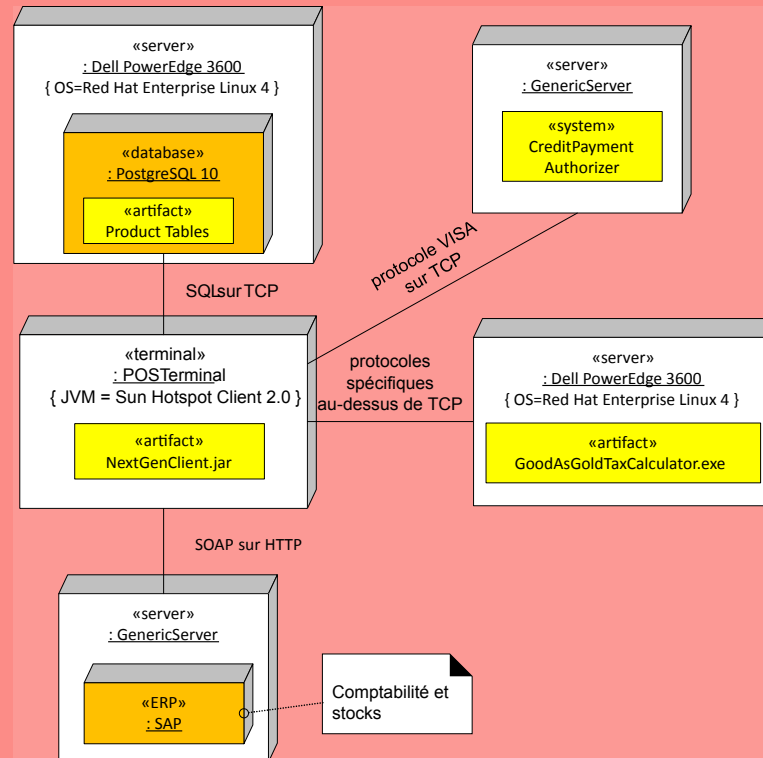
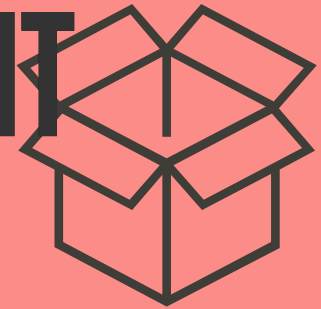
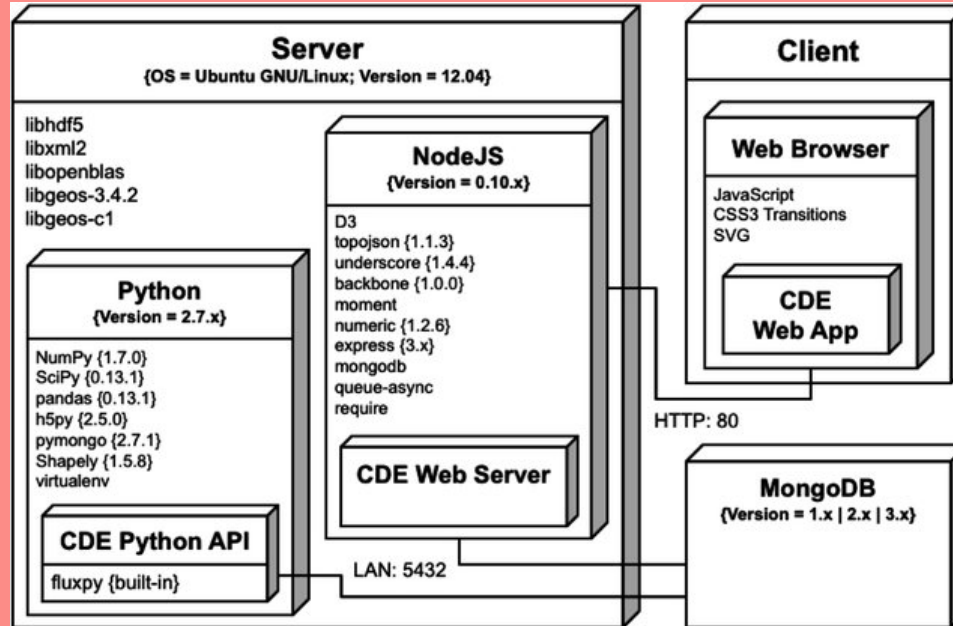


fig. F32.2, A38.2

# EXEMPLE 1/

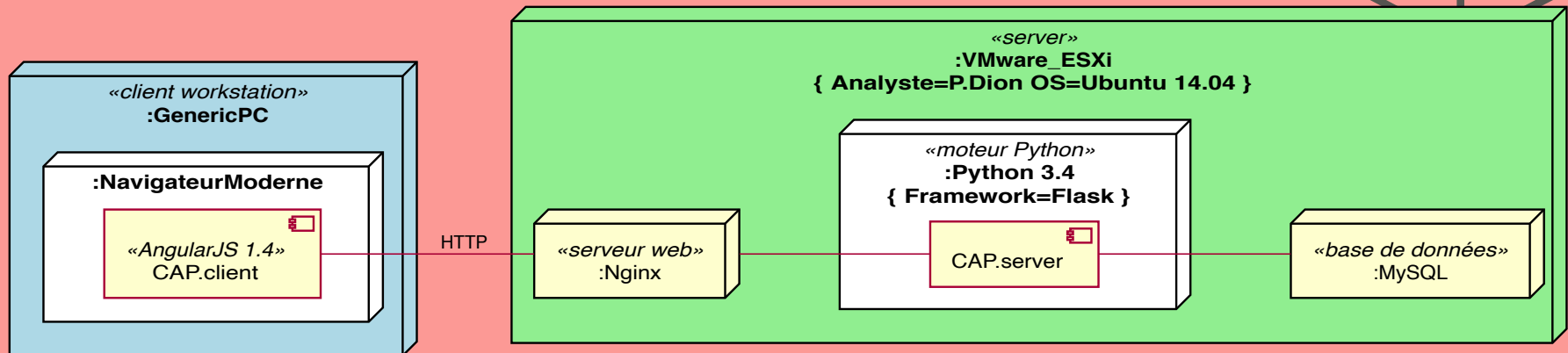
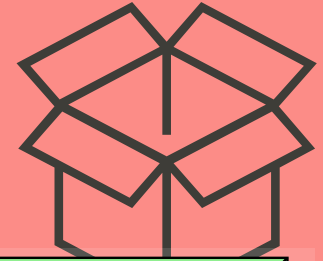


La figure 2 de Distributed visualization of gridded geophysical data: a web API for carbon flux:



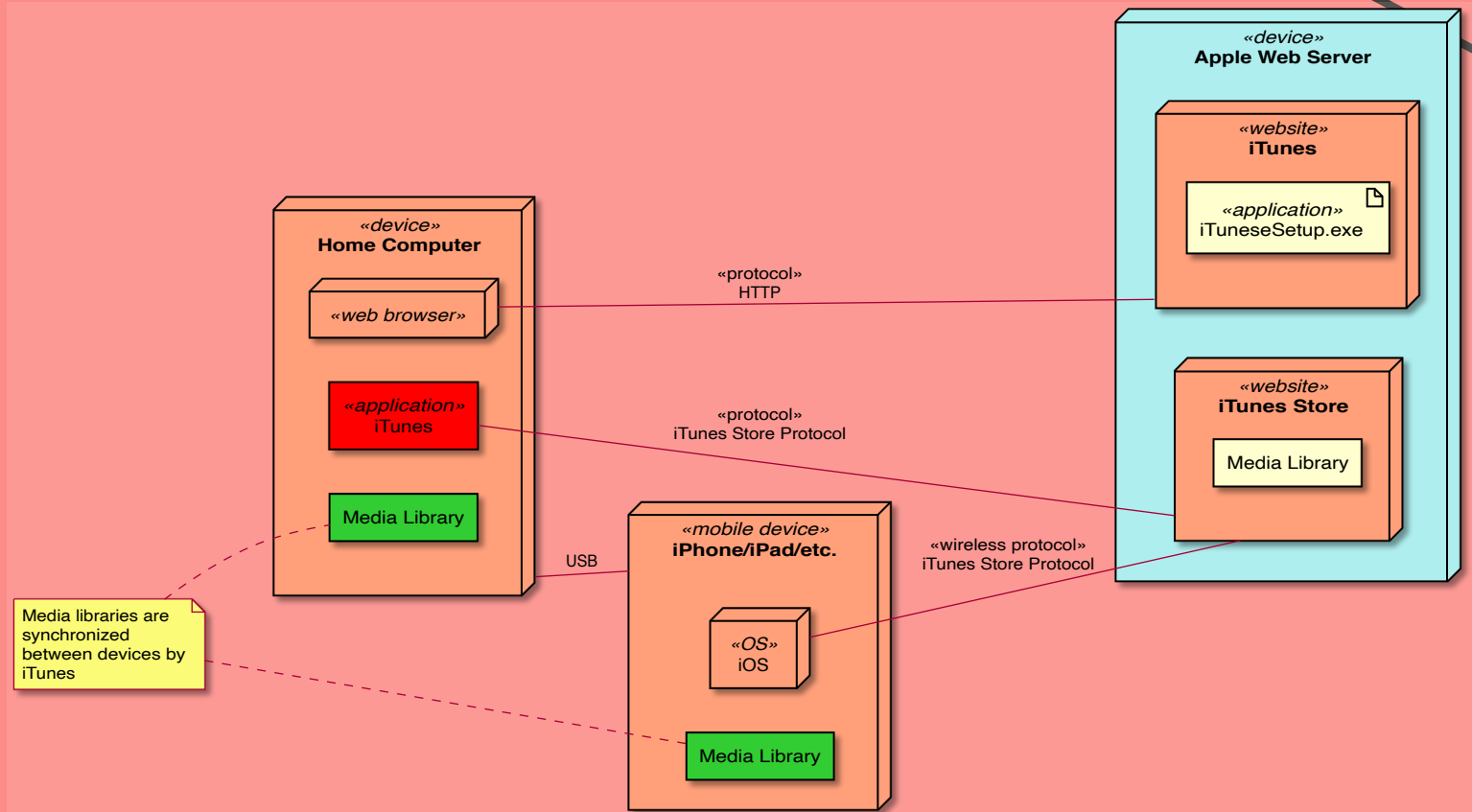
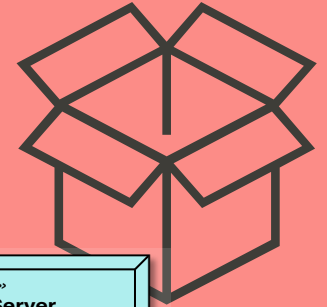


# EXEMPLE 2/



(PlantUML)

# EXEMPLE 3/



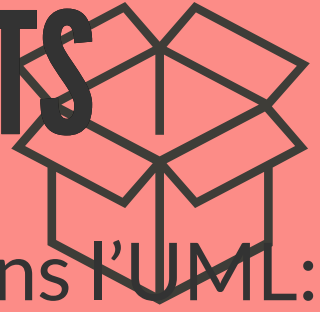
(PlantUML)

# LOG210 SÉANCE #12

## ANALYSE ET CONCEPTION DE LOGICIELS

1. Diagrammes de déploiement
2. Diagrammes de composants ← S20203
3. Révision diagrammes UML
4. Sondage sur technologies
5. Révision Observateur

# DIAGRAMMES DE COMPOSANTS



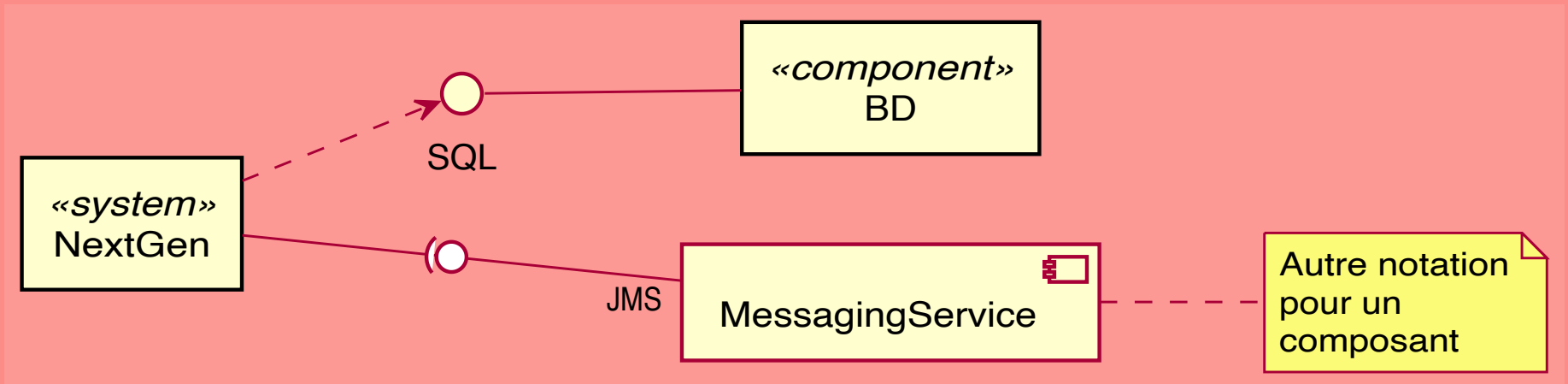
Un composant n'est pas clairement défini dans l'UML:

« Un composant est la **partie modulaire** d'un système qui encapsule son contenu et dont la manifestation est **remplaçable** dans son environnement. Il définit son **comportement** en termes d'**interfaces** fournies et requises. Ainsi, un composant peut être utilisé comme un type dont la conformité est définie par ces interfaces fournies et requises. »

# EXEMPLE EN PLANTUML



Figure F31.2 Composants UML



(PlantUML)

# EXISTE-T-IL RÉELLEMENT DES COMPOSANTS LOGICIELS?



**They have to exist.** Sales and marketing people are talking about them. Components are not a technology. Components are about how customers want to relate to software.

- They want to be able to buy their software a piece at a time, and to be able to upgrade it just like they can upgrade their stereo.
- They want new pieces to work seamlessly with their old pieces, and to be able to upgrade on their own schedule, not the manufacturer's schedule.
- They want to be able to mix and match pieces from various manufacturers. This is a very reasonable requirement. It is just hard to satisfy.

Ralph Johnson do component exist

# MÉTAPHORE CINÉMA MAISON



- IFTTT
1. Heroku
  2. IOT



viet / photostream

# DIAGRAMME DE COMPOSANTS

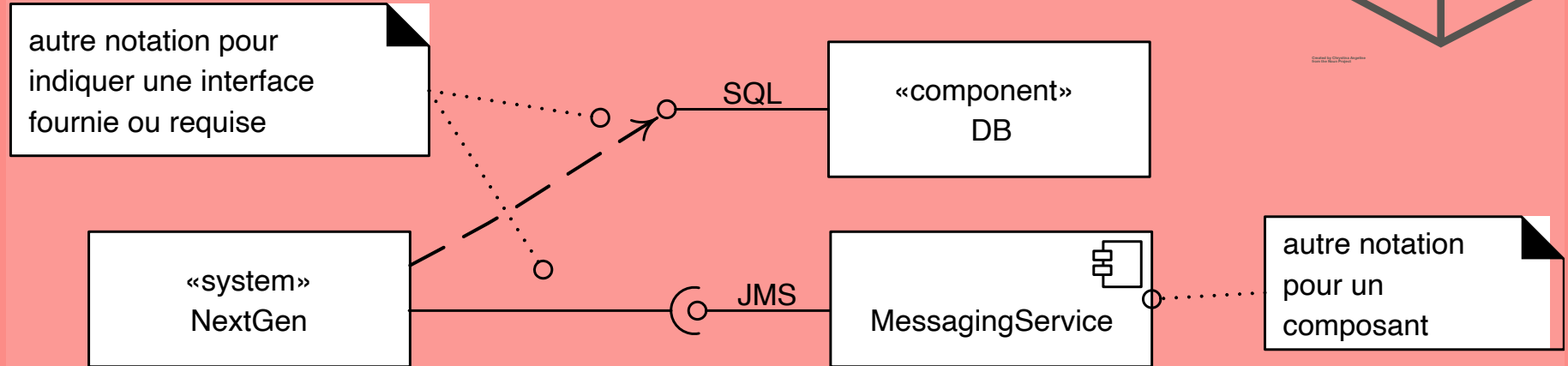


fig: 31.2, A37.2



# D'AUTRES EXEMPLES

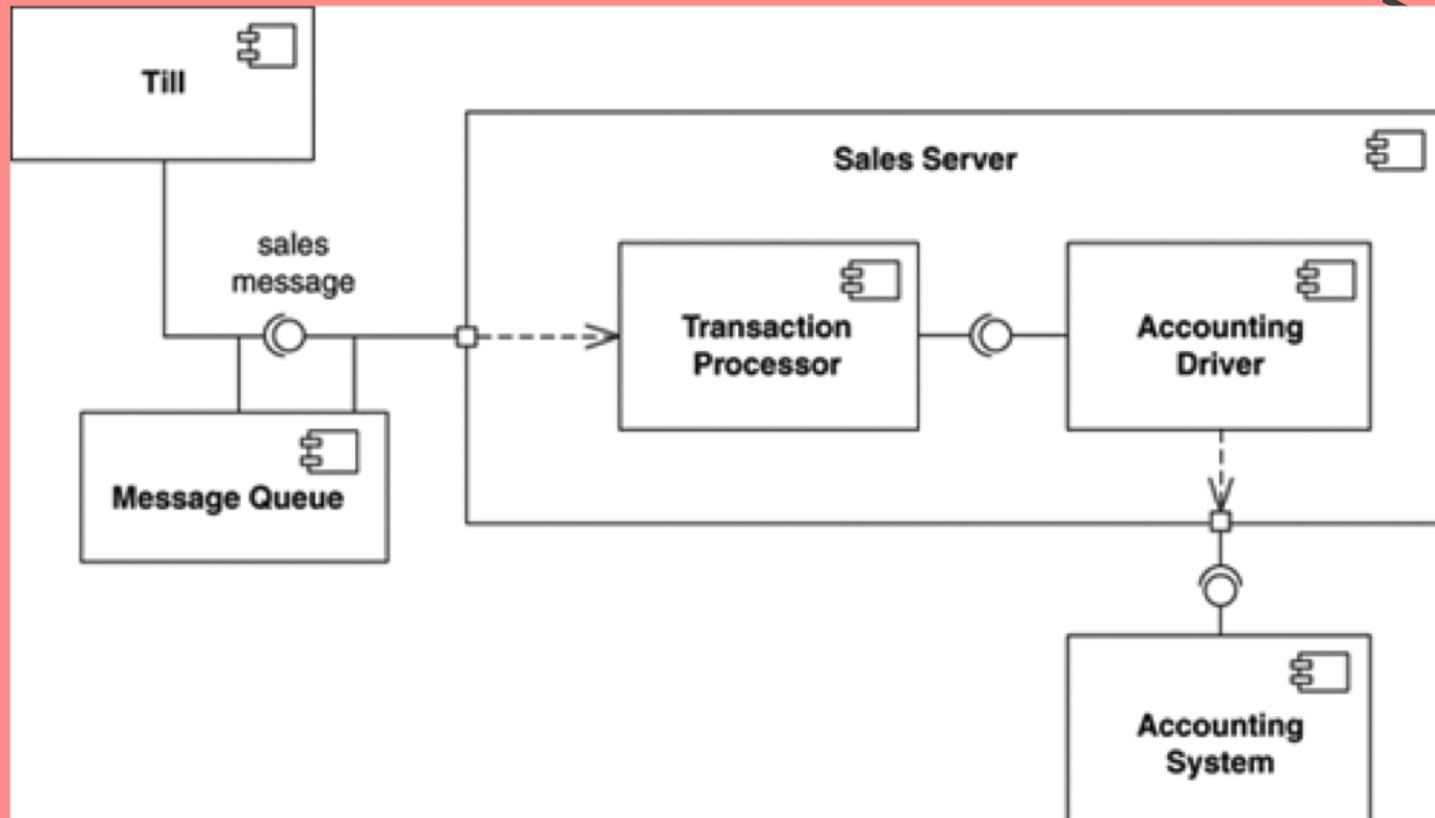
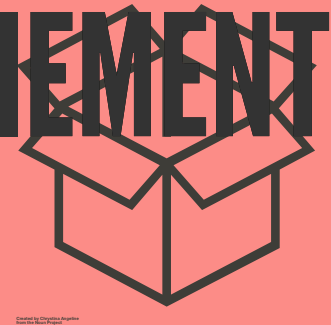


fig.: F14.2 UML Distilled: A Brief Guide to the Standard Object Modeling Language, Third Edition

# EXERCICES DIAGRAMME DE DÉPLOIEMENT




1. Application web
2. Tutorial online
3. UML deployment diagram - Apple iTunes
4. UML Deployment Diagrams Examples

Prenez le temps regarder la documentation de plantUml pour les diagrammes de déploiement et de composants. Nous ferons un exercice la semaine prochaine.

- <https://plantuml.com/deployment-diagram>
- <https://plantuml.com/component-diagram>

# LOG210 SÉANCE #12

## ANALYSE ET CONCEPTION DE LOGICIELS

1. Diagrammes de déploiement
2. Diagrammes de composants
3. Révision diagrammes UML  S20203
4. Sondage sur technologies
5. Révision Observateur

# MÉTHODOLOGIE DE DÉVELOPPEMENT UML

- UML est un langage de modélisation général qui inclut un large éventail de concepts, notations et diagrammes.
- Avant d'intégrer UML dans un processus de développement, il est fondamental de définir correctement une méthodologie pour prendre en charge son utilisation.
  - La définition d'une méthodologie de développement facilitera l'intégration et l'adoption d'UML, augmentera la qualité des modèles et augmentera la productivité des développeurs.

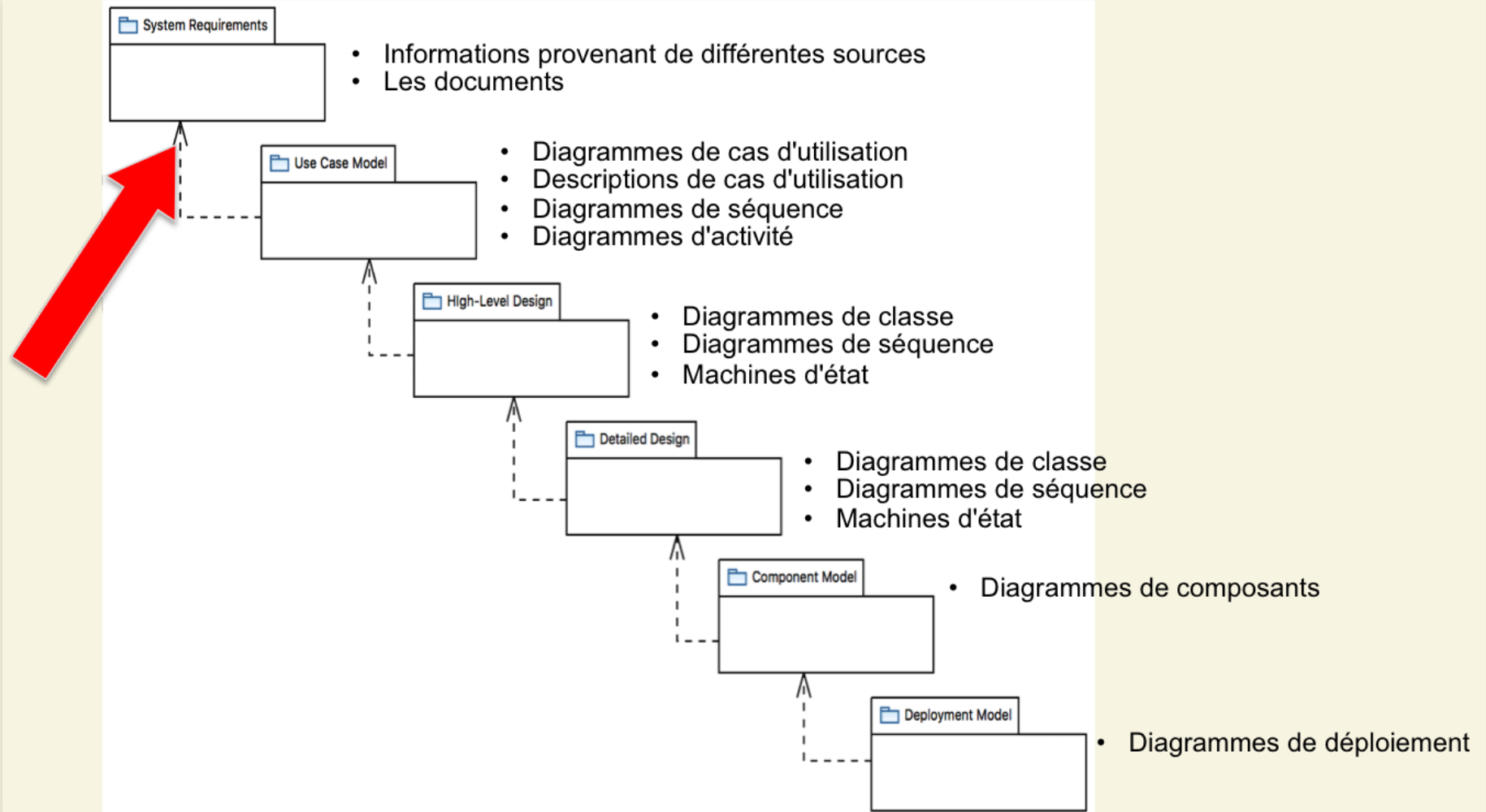
# MÉTHODOLOGIE DE DÉVELOPPEMENT UML

- Pour chaque phase de développement , vous devez définir
  - Ensemble de diagrammes UML utilisés dans la phase
  - Rôle de chaque diagramme
  - Sous-ensemble de la notation UML utilisée
  - Remarque: certains diagrammes peuvent jouer différents rôles dans différentes phases du processus de développement .
    - Par exemple . Les diagrammes de séquence peuvent être utilisés pour décrire les interactions entre le système et les acteurs lors de la phase de modélisation de cas d'utilisation , fournir des détails sur l'exécution de scénarios lors de la phase de conception et capturer des traces d'exécution lors de la phase de test.
- Définir un ensemble de règles et de directives pour développer des éléments de modèle aux différentes phases de développement

# PERSONNALISER / CUSTOMISER UML

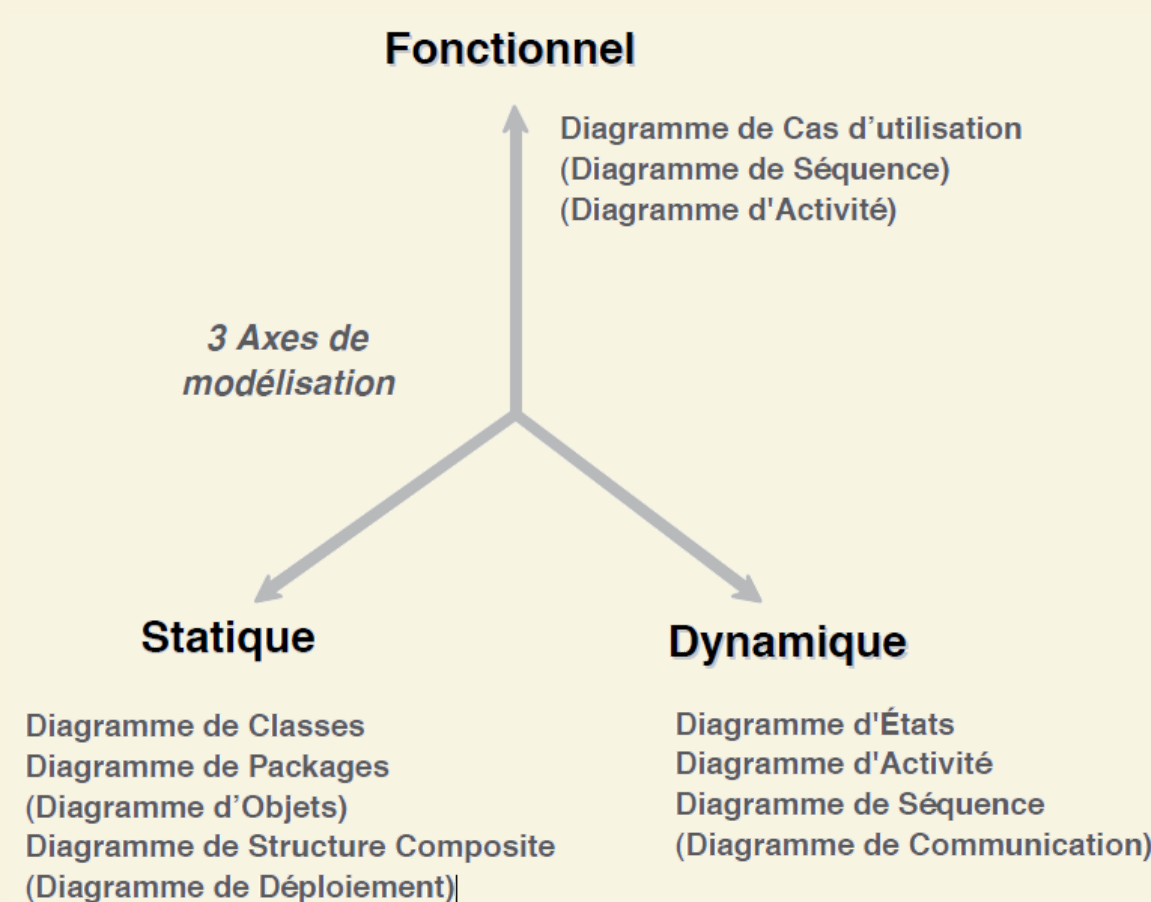
- UML fournit un ensemble de concepts, notations et diagrammes pouvant être utilisés dans un large éventail de contextes de développement.
- Cependant, seul un sous-ensemble de ces éléments est requis dans tout contexte de développement
- Le mécanisme de profil UML peut être utilisé pour personnaliser le langage UML pour des domaines de développement spécifiques.
- Mais la plupart des outils UML n'ont pas réussi à fournir un support de premier ordre pour la personnalisation et la modélisation spécifique à un domaine.

# ÉTAPES DE MODÉLISATION



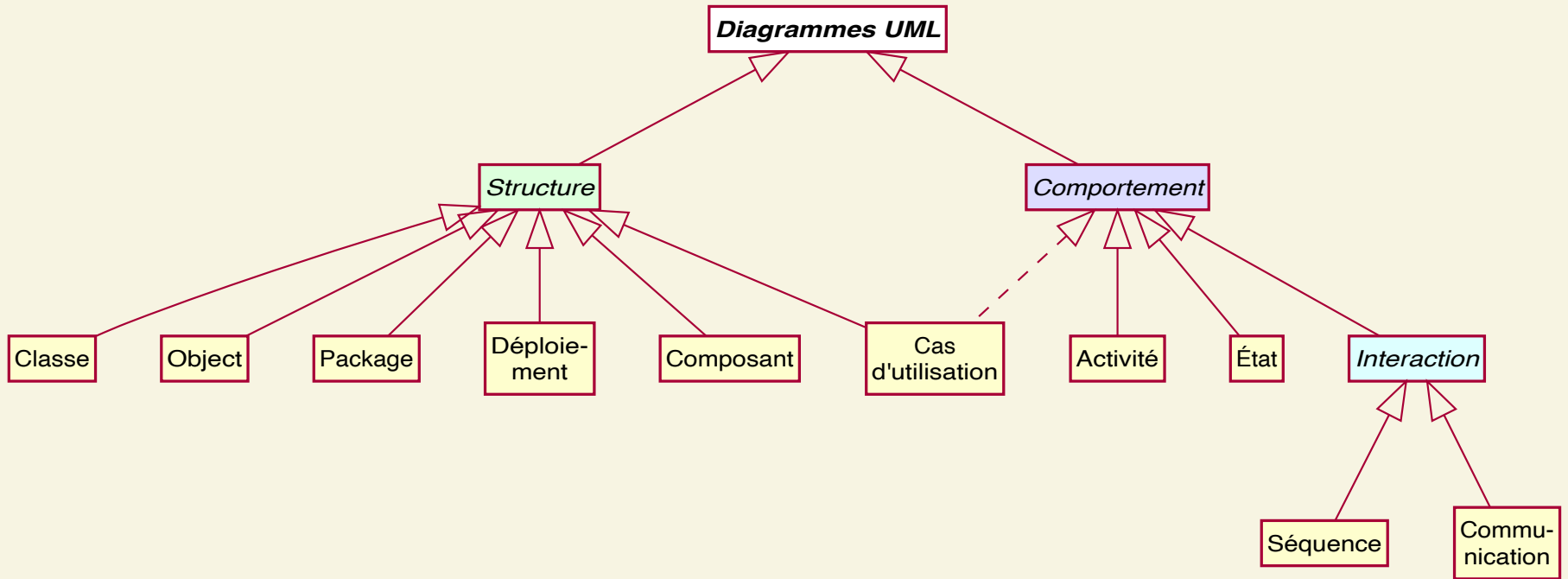
# DIMENSIONS DE LA MODÉLISATION UML

Perspective selon UML par la pratique





# STRUCTURE VS COMPORTEMENT



- Développer Itérativement
- Utilisez la décomposition et le packaging pour structurer votre modèle de cas d'utilisation
- Définir l'ensemble des acteurs
- Définir / adopter un ensemble de guidelines et une convention de nommage
- Utiliser un modèle de description de cas d'utilisation «standard»
- Éloignez-vous de trop de détails , vous en parlerez plus tard
- **Remarques:**
  - Rappelez-vous que l'un des objectifs principaux des cas d'utilisation est de communiquer avec les parties prenantes , qui incluent des personnes de divers horizons.

**Pas de solution magique , écrire des cas d'utilisation est un art!**

- Respecter le décalage de représentation
- Commencez par définir les classes directement dérivées des exigences et du modèle du domaine. Ajoutez de manière itérative de nouvelles classes et raffinez -les selon vos besoins au fur et à mesure de votre progression dans le processus de conception
- Définissez les principaux attributs de la classe . Ajoutez de manière itérative de nouveaux attributs au fur et à mesure que vous avancez dans le processus de conception de manière itérative
- Utiliser une convention de nommage cohérente pour les classes et les attributs

# LES MEILLEURES PRATIQUES – DIAGRAMME D'INTERACTION

- Se concentrer sur le comportement global du système et la communication entre le système et l'ensemble des acteurs
  - Ne commencez pas à prendre des décisions de conception à ce stade
- Concentrez-vous sur les principaux scénarios , n'essayez pas d'être exhaustif
- Décomposer les diagrammes de séquences longs en fragments plus petits, le cas échéant
  - Définir des interactions séparées et utiliser « Interaction Use » pour faire référence à des interactions définies séparément

# LES MEILLEURES PRATIQUES – DIAGRAMME ACTIVITÉ


- Choisissez le «bon» niveau d'abstraction / détails
  - Évitez le piège de spécifier les détails de conception,... vous le ferez plus tard!
- Maintenir la cohérence avec les descriptions de cas d'utilisation
  - Ne signifie pas une traçabilité individuelle
- Concentrez-vous sur la clarification et la compréhension du modèle

# LES MEILLEURES PRATIQUES – “DIAGRAMME D'ÉTAT”

- Choisissez le «bon» niveau d'abstraction / détails
  - Évitez le piège de spécifier les détails de conception,... vous le ferez plus tard dans le processus de conception!
- Maintenir la cohérence avec le modèle de cas d'utilisation
  - Ne signifie pas une traçabilité formelle individuelle
- Exploiter les relations de cas d'utilisation (et de scénarios) pour structurer les machines d'états
- Ne définissez pas une machine d'états pour chaque classe , mais uniquement pour les classes actives


# LOG210 SÉANCE #12

## ANALYSE ET CONCEPTION DE LOGICIELS

1. Diagrammes de déploiement
2. Diagrammes de composants
3. Révision diagrammes UML
4. Sondage sur technologies... 
5. Révision Observateur

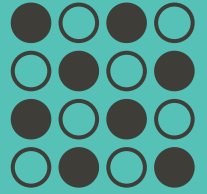
# LOG210 SÉANCE #12

## ANALYSE ET CONCEPTION DE LOGICIELS

1. Diagrammes de déploiement
2. Diagrammes de composants
3. Révision diagrammes UML
4. Sondage sur technologies
5. Révision Observateur  S20203



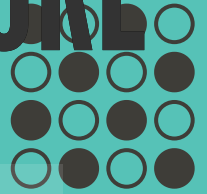
# EXERCICE OBSERVATEUR



Created by Jonathan Li  
from the Noun Project

- Rappeler le patron Observateur
- Réviser le diagramme de structure
- Réaliser les diagrammes de comportement
  - Communication
  - Activité
  - Séquence

# SOLUTION : OBSERVATEUR STRUCTURE



Created by Jonathan Li  
from the Noun Project

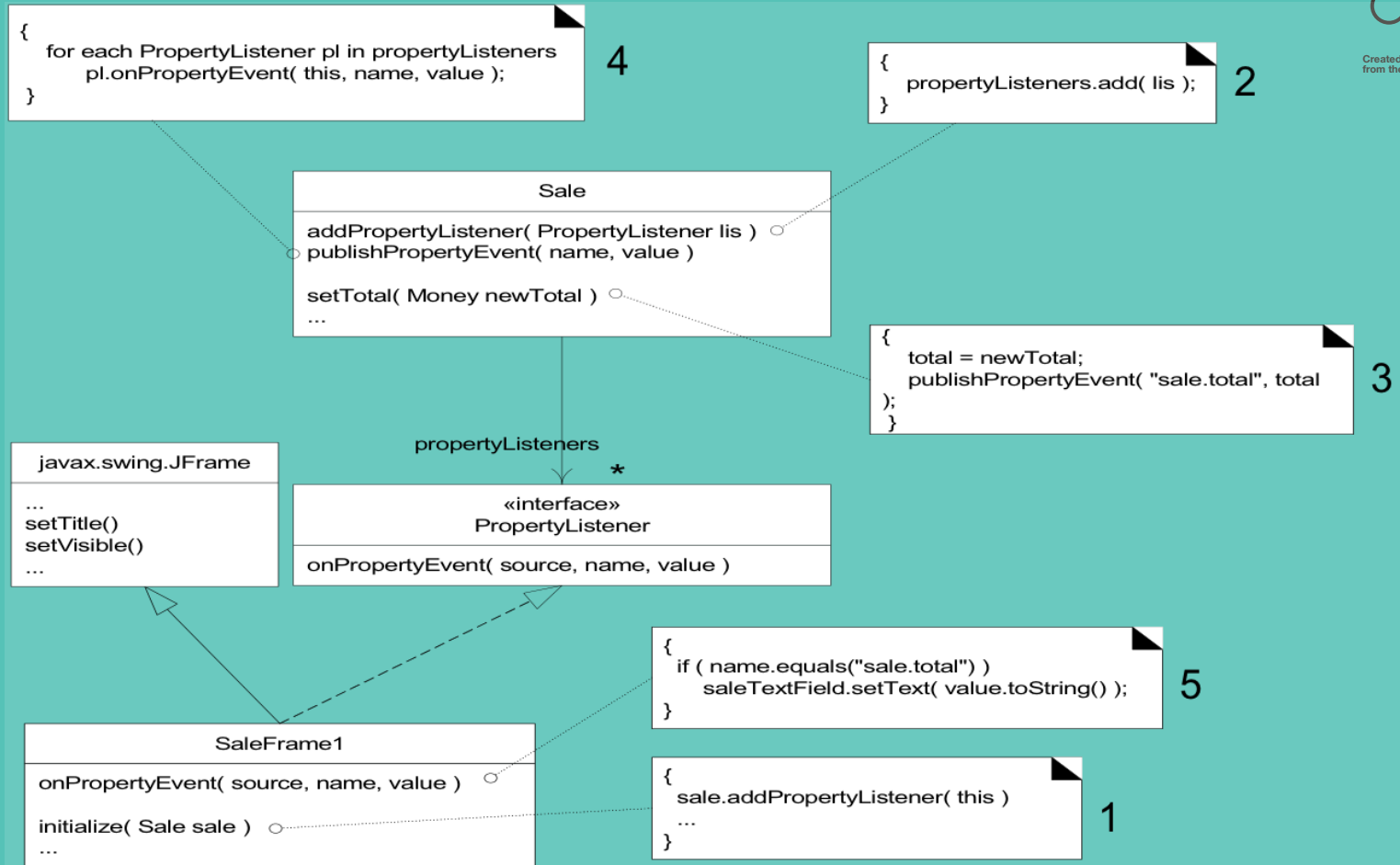


fig. F26.22

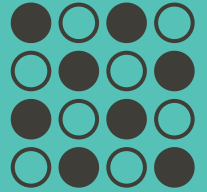


# OBSERVATEUR COMMUNICATION

Created by Jonathan Li  
from the Noun Project

Voir la note pour la solution

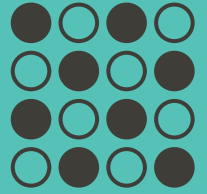
# OBSERVATEUR ACTIVITÉ



Created by Jonathan Li  
from the Noun Project

## Voir la note pour la solution

# OBSERVATEUR SÉQUENCE

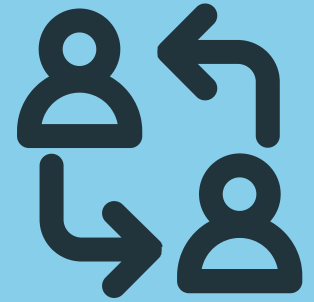


Created by Jonathan Li  
from the Noun Project

Voir la note pour la solution

# SÉANCE #12

## RÉTROACTION: PAGE D'UNE MINUTE



Created by Prithvi  
from the Noun Project

1. Quels sont les deux [trois, quatre, cinq] plus importants [utiles, significatives, surprenantes, dérangementantes] choses que vous avez apprises au cours de cette session?
2. Quelle (s) question (s) reste (s) en tête dans votre esprit?
3. Y a-t-il quelque chose que tu n'as pas compris?

<https://1drv.ms/u/s!An6-F73ulxAOhVyiCB46jTeINVLs>

