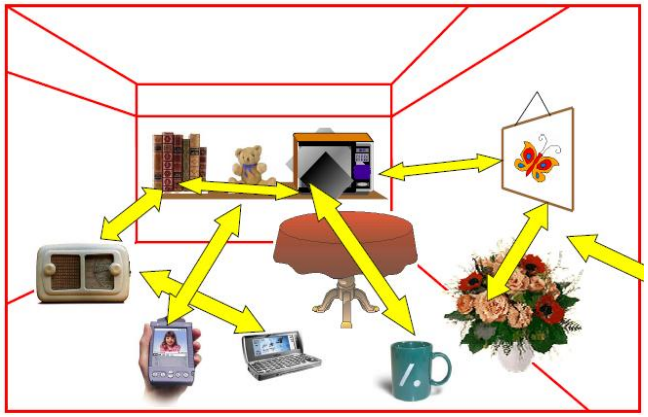
Adaptation de Services en Informatique Ambiante et Projet CONTINUUM

Jean-Yves Tigli, tigli@polytech.unice.fr,
Equipe RAINBOW
Laboratoire I3S – Université de Nice Sophia Antipolis - CNRS

Cadre : Informatique Ambiante

- Ubiquituous Computing ou Informatique Ambiante …

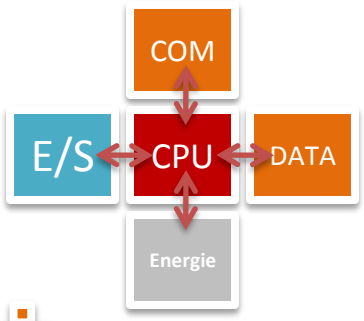


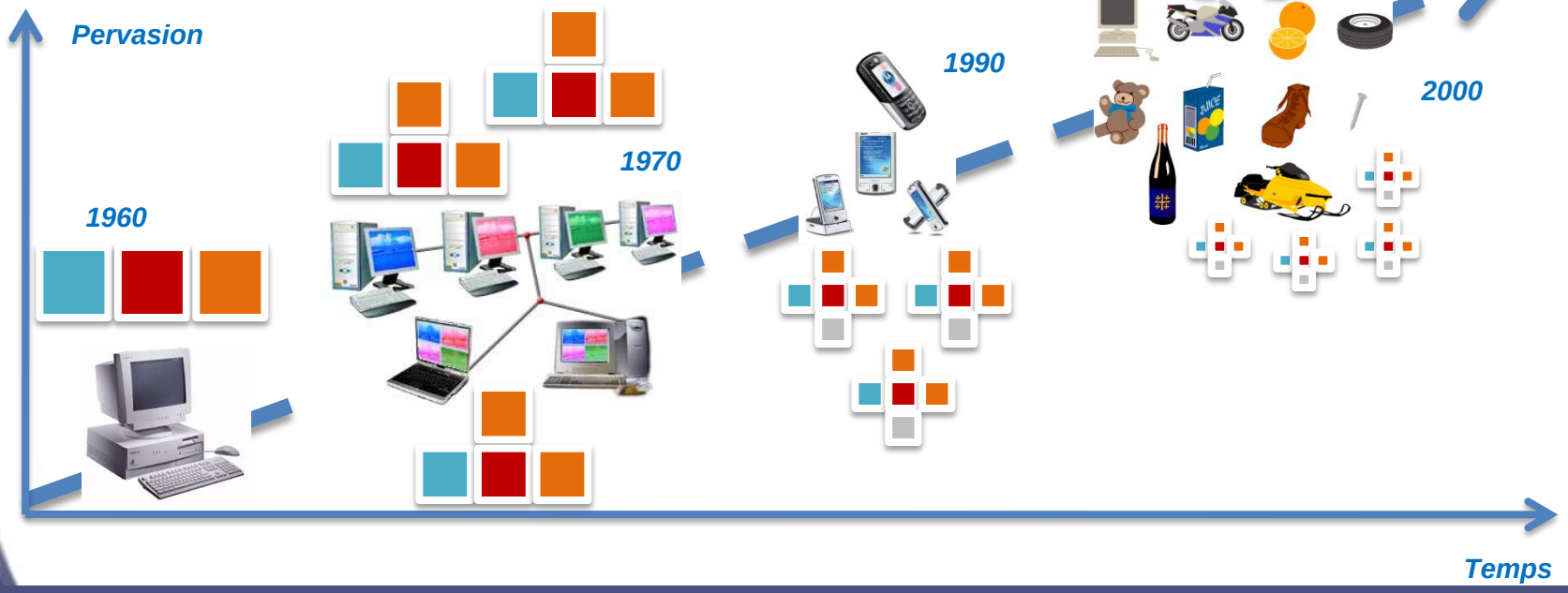
- [Weiser 1991]

« Silicon-based information technology, is far from having become part of the environment »

Cadre : Informatique Ambiante

- ✓ **Depuis Von Neumann … les évolutions …**

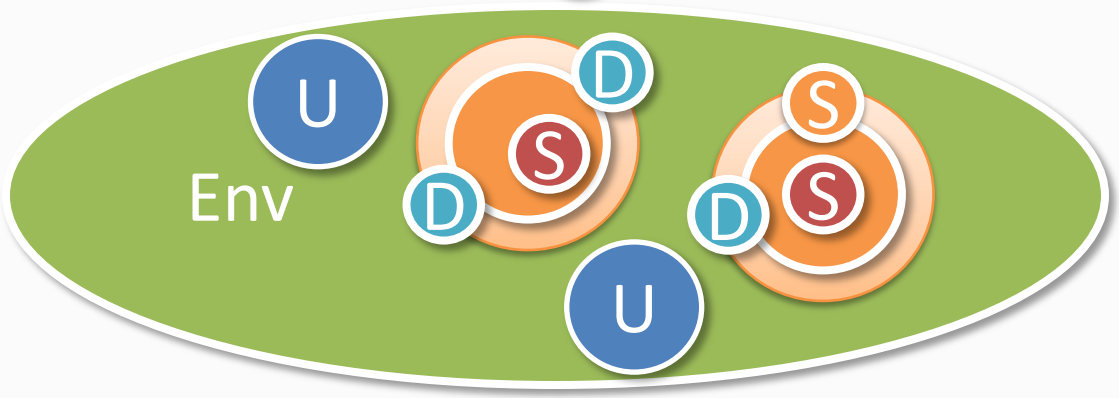




Adapté de « Nano-informatique et intelligence ambiante », JB Waldner, Hermes Science Publishing, 2007

Vision et organisation générales

- ✓ Un Système composé de Dispositifs D et de Services S
  - **Dispositifs** D : Entité physique, artefact, équipement, objet du monde réel assurant un ensemble de fonctions et/ou offrant un ensemble de services.
  - **Services** Logiciels S : Service est une entité logicielle capable de fournir une fonction utilisable par un tiers
  - **Infrastructure** : Ensemble des services et dispositifs disponibles et imposés
- ✓ Evoluant et interagissant avec un Environnement physique et avec des Acteurs (ex. Utilisateurs U)



- ✓ Adaptant ses Services Logicielles S

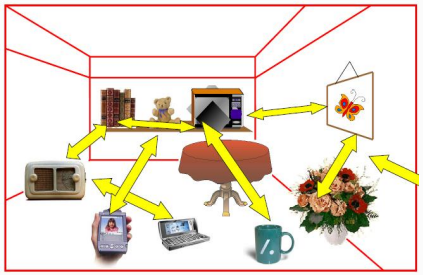
Les contraintes de l'informatique Ambiante

Hétérogénéïité
Dynamicité

- ✓ Des dispositifs hétérogènes voire insolites (entrées, sorties)
  - Sur les utilisateurs
  - Dans l'environnement
- ✓ Des services et dispositifs imposés (infrastructure)
  - Des services "boites noires"
  - Des dispositifs avec leurs caractéristiques
- ✓ Des services et dispositifs hautement dynamiques
  - Apparition, disparition de dispositifs comme conséquence directe de la mobilité de l'utilisateur

Les enjeux de l'informatique Ambiante

- ✓ Des applications adaptées au contexte
  - Des applications fédérant spontanément divers dispositifs et services partiellement connus a priori
  - Des applications évoluant avec l'infrastructure, l'environnement et les utilisateurs pour maintenir leurs principales fonctionnalités.



- ✓ Des usages variés
  - Des utilisateurs variés avec des besoins différents
  - Des situations d'usages différentes pour un même utilisateur (contexte)

Projet Continuum

- ✓ **CONTinuité de service en INformatique UbiqUitaire et Mobile**
- ✓ **Objectif :**

  « garantir la continuité de service auprès d'utilisateurs en situation de mobilité dans des environnements aux ressources variables, dynamiques et hétérogènes, et ceci dans le respect d'un équilibre maîtrisé entre autonomie logicielle et contrôle humain »

- ✓ **Partenaires :**

Projet Continuum

- ✓ **Enjeux scientifiques :**
  - Maitriser l'adaptation des services à la situation (I3S-RAINBOW))
  - Maitriser l'hétérogénéité sémantique des différents services (LIG –HADAS)
  - Maitriser l'équilibre entre autonomie logicielle et contrôle humain (LIG –IIHM)

- ✓ **Expérimentations :**
  - Les premières expérimentations menées par des sociétés de l'industrie de l'eau.
  - Objectif : amélioration de la disponibilité des services d'information pour les agents de terrain et mobiles quelque soit leur activité.

Intergiciel orientée Services pour Dispositifs

Service Lightweight Component Architecture (SLCA)

Architecture et Informatique Ambiante

- Inspiré de l'approche SOA qui offre un « *bus de service pour l'entreprise* »
- Vers une approche pour offrir un « *bus de service pour un espace ambiant [Coutaz, Crowley 08]*»

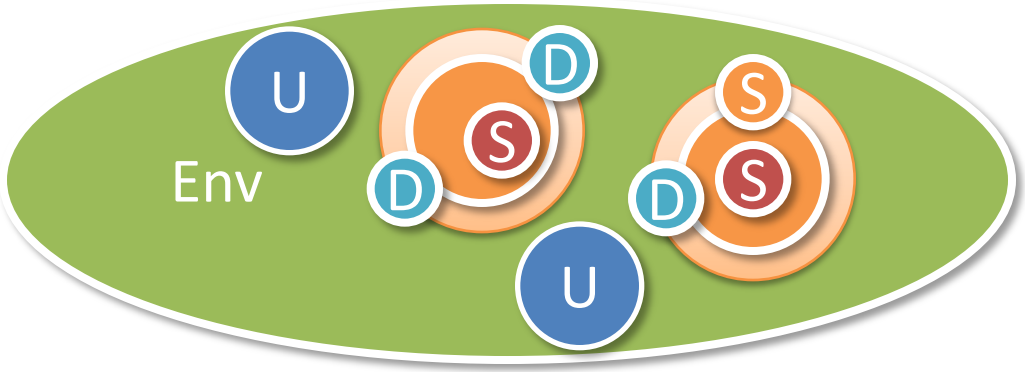
  - Des nuances services et services pour dispositifs
    - Localité
    - Mobilité
    - Haute dynamicité
    - Interactions directes environnement physique / utilisateurs
    - Réactivité

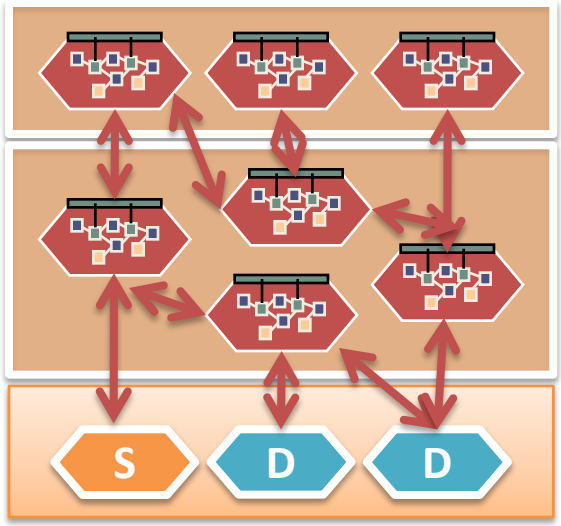
  - Des évolutions
    - SOA et EDA (event based communications)
    - Recherche et Découverte distribuée
    - Localité dans la composition (LCA)

Modèle d'architecture SLCA : Service Lightweight Component Architecture

Notre Modèle de composition SLCA...

- ✓ Trois principes (Pi)

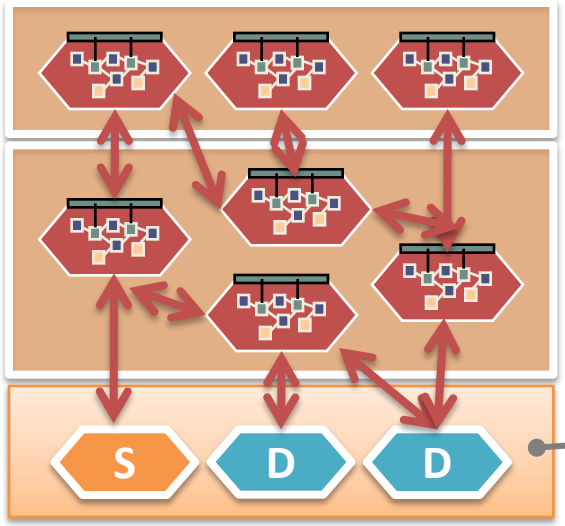




Infrastructure de
Services pour Dispositifs

Notre Modèle de composition SLCA…

- ✓ P1 : Une infrastructure de Services et Dispositifs



Infrastructure de Services pour Dispositifs

1

Une Infrastructure de Services et de Dispositifs (ex. UPnP, DPWS …)

✓Gestion contextuelle Apparition / Disparition de Services

✓Interaction entre Services : requête /réponse et événementielle

Notre Modèle de composition SLCA...

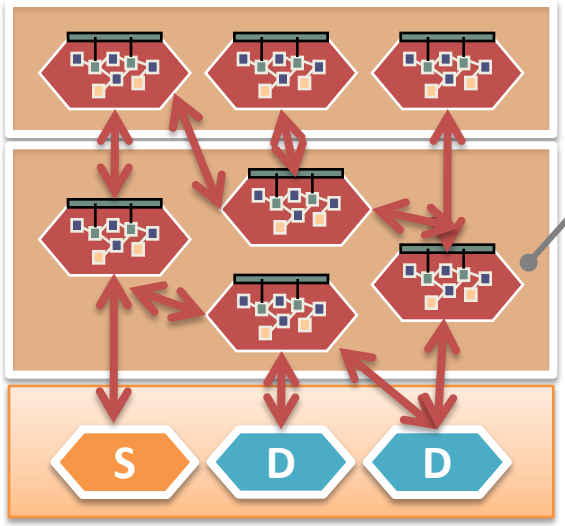
✓ P2 : Composition de services (LCA)

2

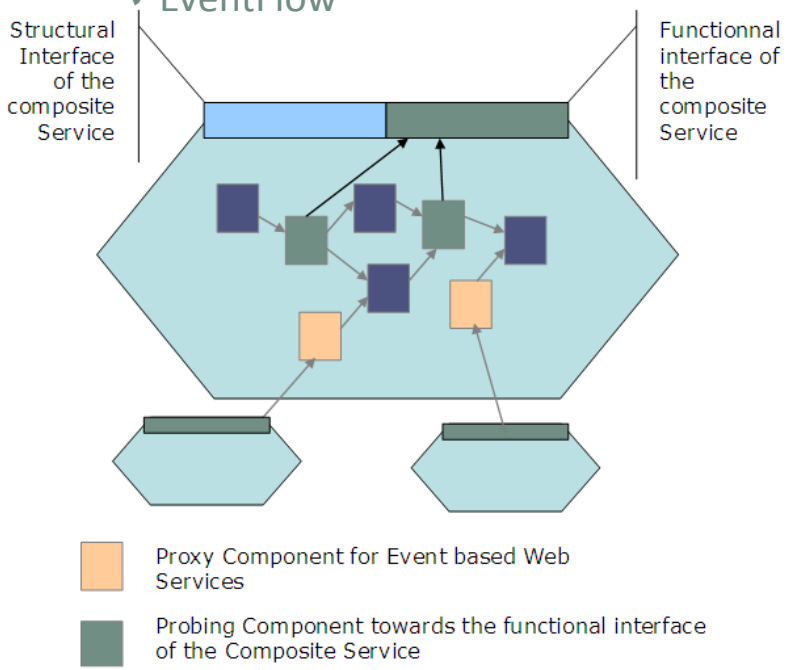
Des services dynamiques composites

✓Composition non basée sur un langage et moteur d'exécution mais sur un assemblage de composants légers et son conteneur

✓EventFlow

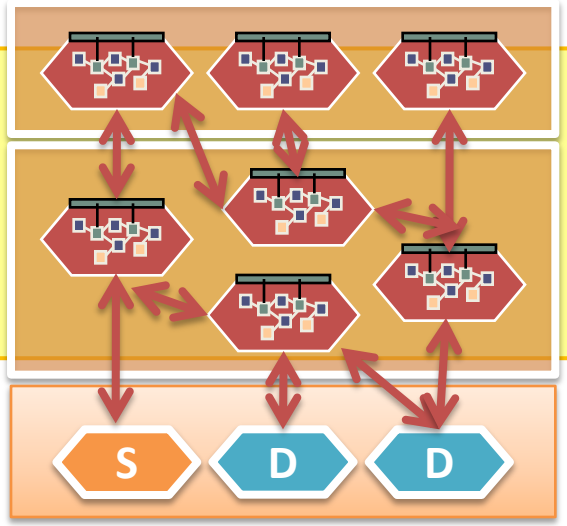


Infrastructure de Services pour Dispositifs



Et l'adaptation ?

- ✓ P3 : Adaptation



Infrastructure de
Services pour Dispositifs

Quid de
l'adaptation ?

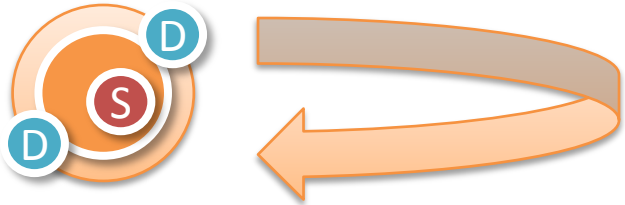
3

L'adaptation

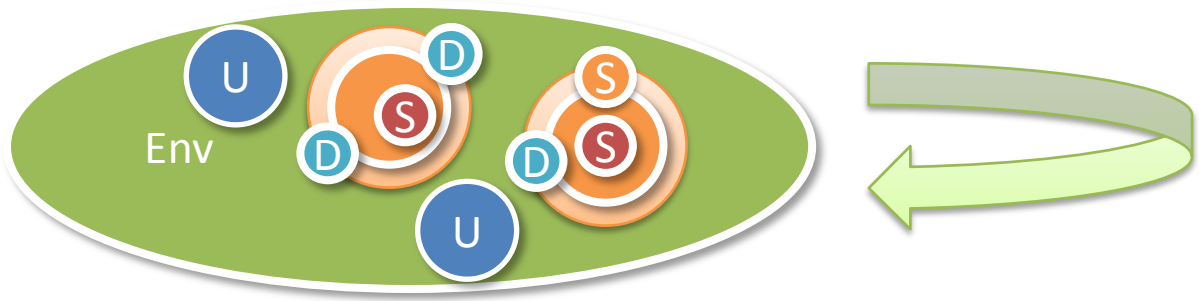
Adaptation du Système au Contexte

Adaptation du Système au Contexte - Problématique

- ✓ Une Application initiale en Informatique Ambiante n'est composé a priori que de l'ensemble de services et dispositifs de l'infrastructure
- ✓ Comment faire adapter l'application quand l'environnement et l'infrastructure évolue continuellement ?
  - Adapter le système à l'évolution de son infrastructure

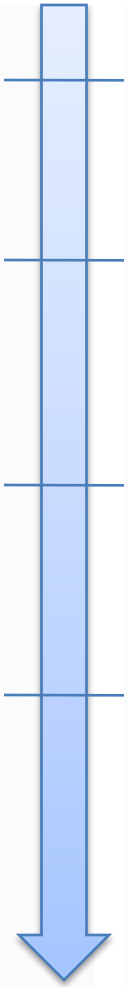


  - Adapter le système à l'évolution de son environnement physique



Une approche inspirée des aspects

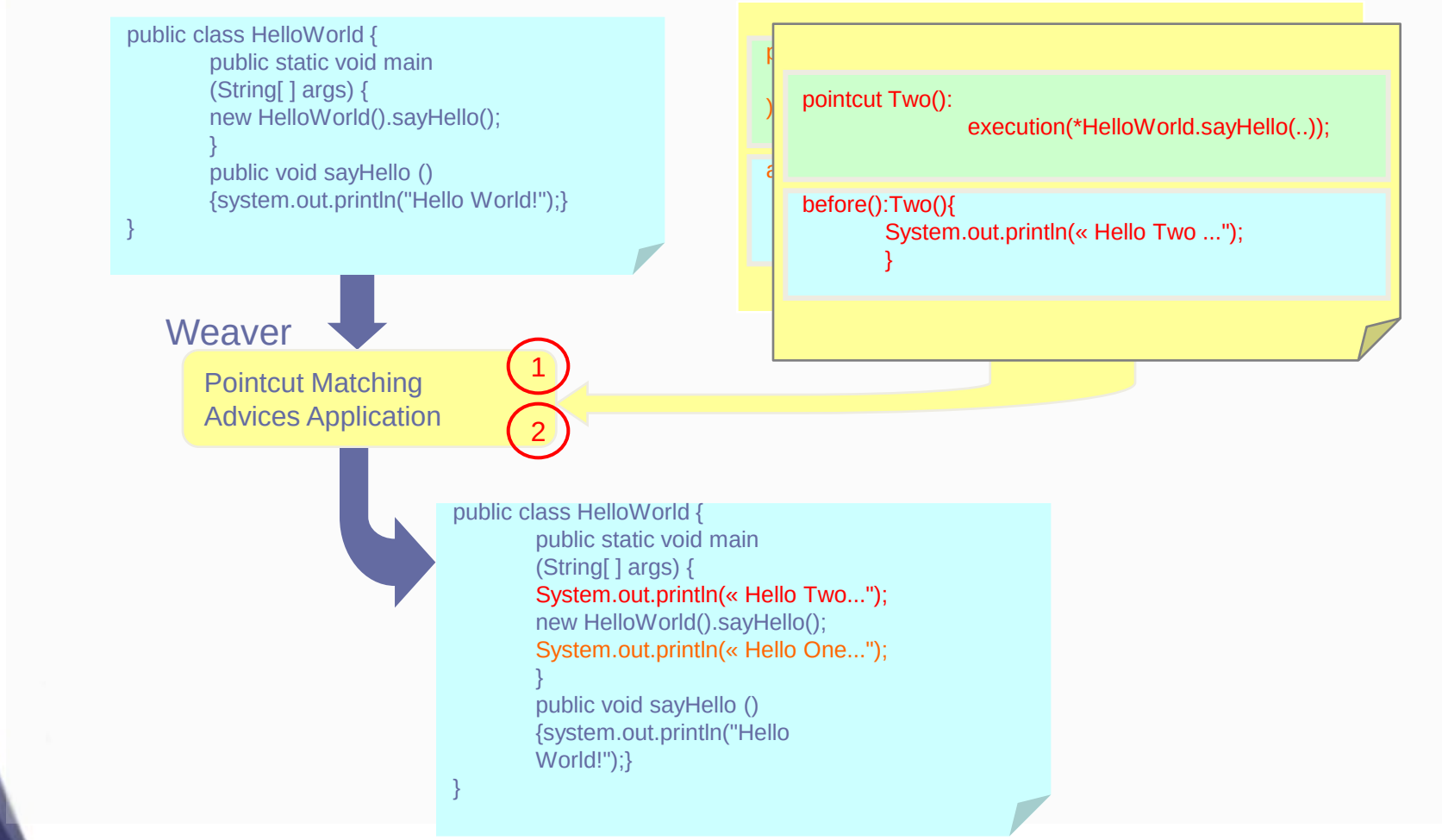
- ✓ AOP [Kiczales 97]
- ✓ AspectJ [Kiczales 01]
- ✓ AO4BPEL [Charfi 04]
- ✓ EAOP [Douence 06]
- ✓ AA [Tigli 06, Cheung 07,]



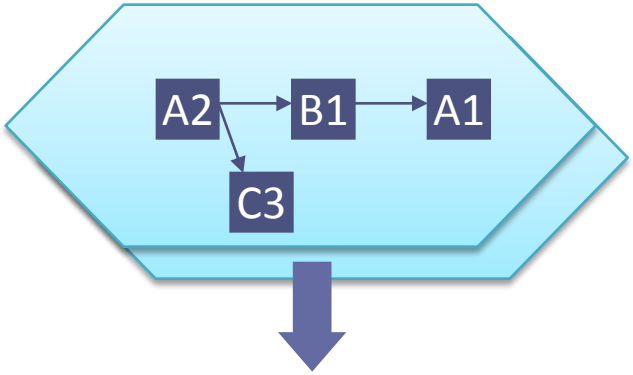
- ✓ Minimiser la dispersion de code
- ✓ Préoccupations transverses, non fonctionnelles
- ✓ Modularité Transverse
- ✓ Déclenchement sur événements
- ✓ Auto adaptation transverse (tissage « opportuniste » d'aspects sur SLCA)

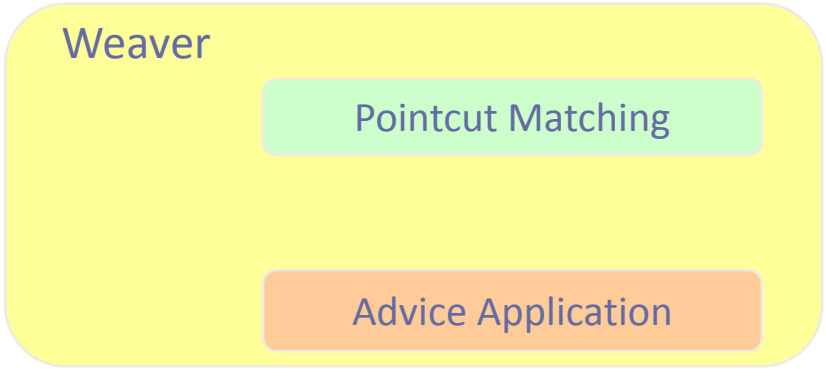
Aspects d'Assemblage

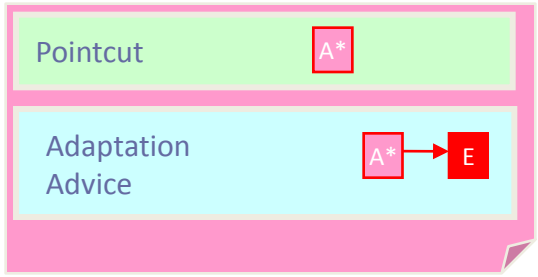
Rappel : Principe AOP

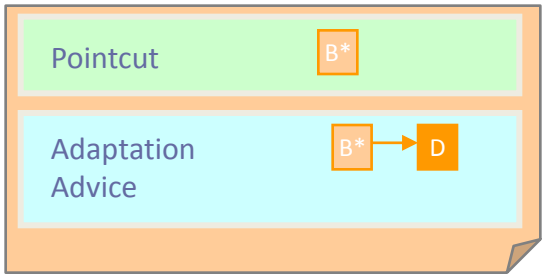


Le principe des Aspects d'Assemblage (AA)

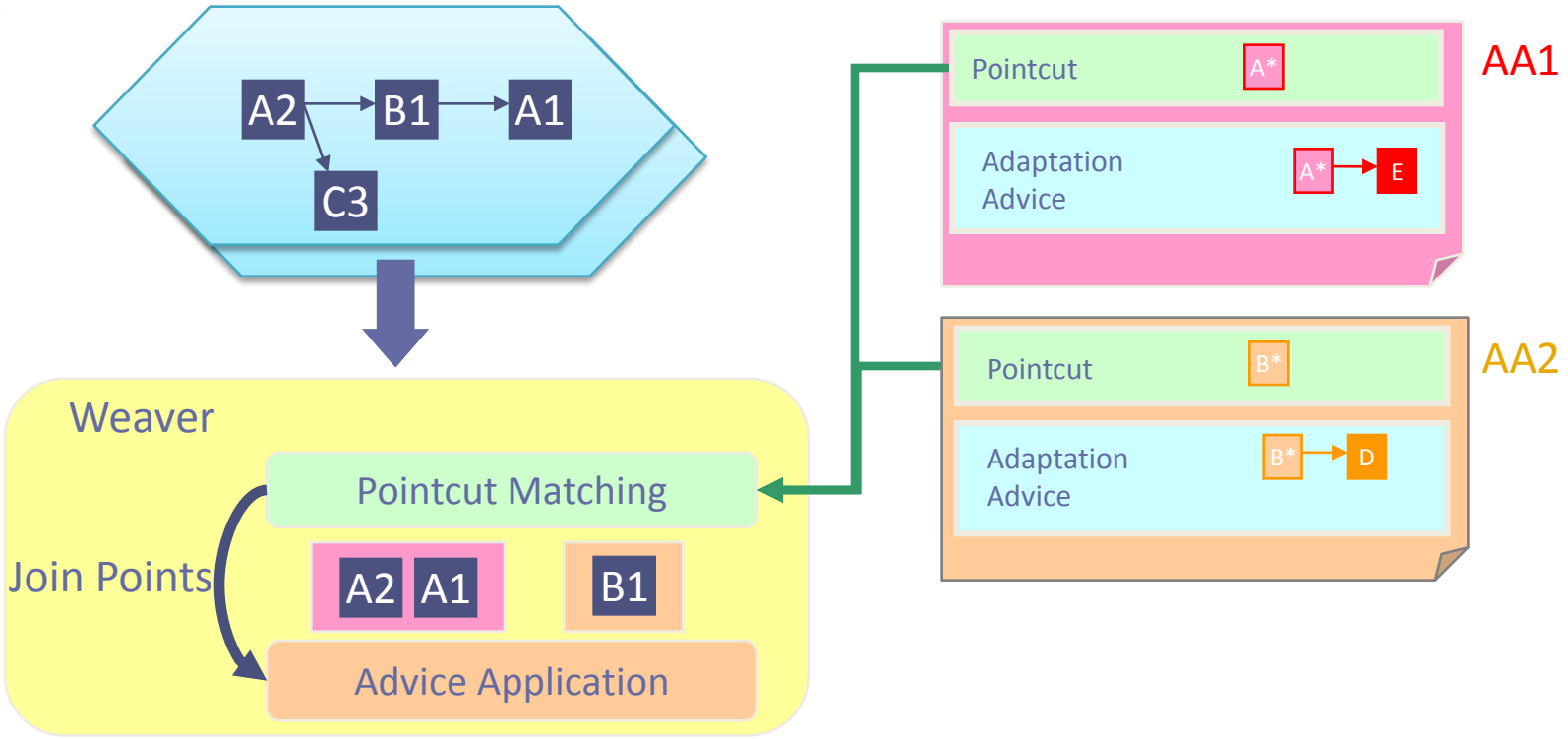




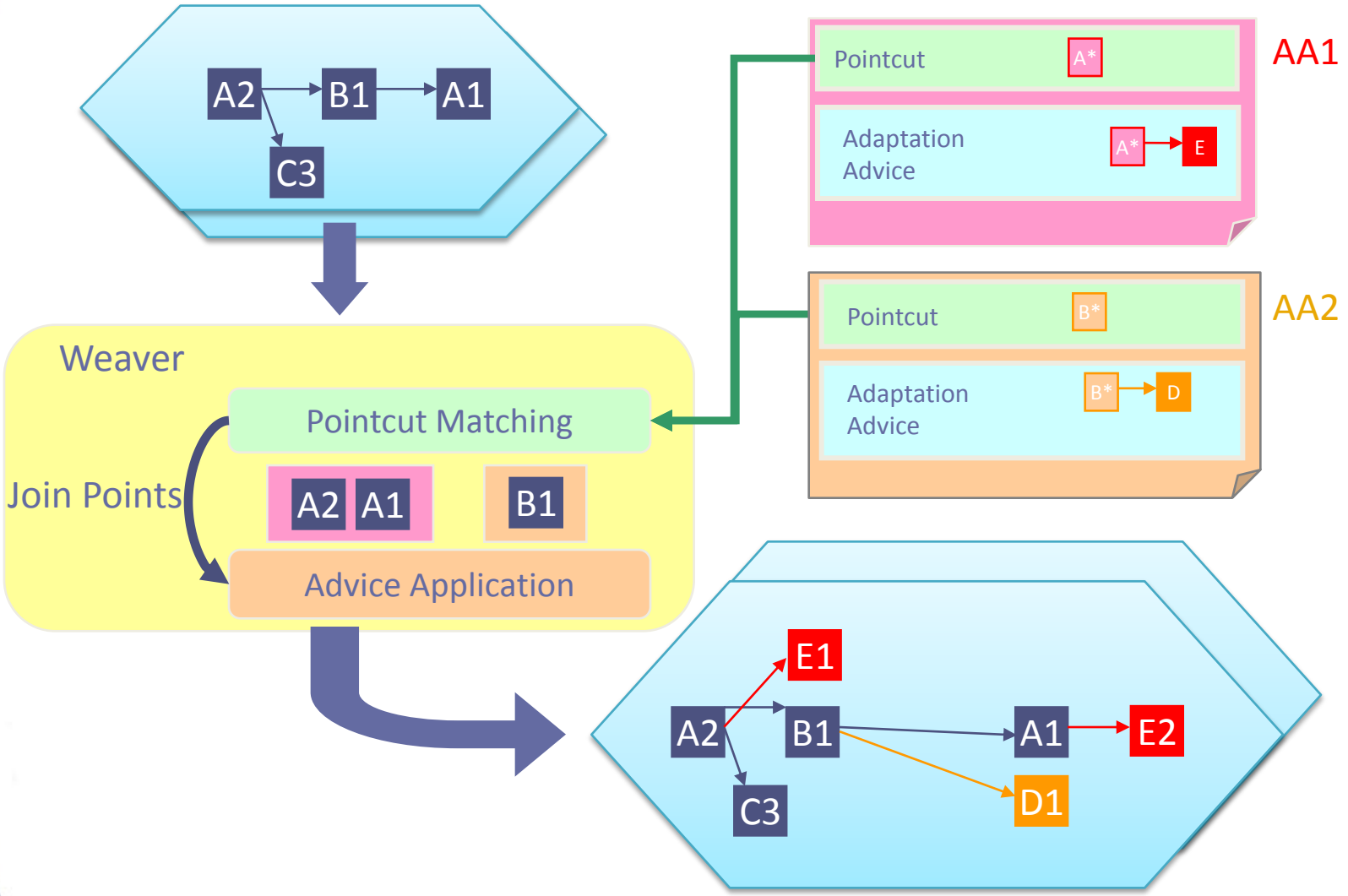




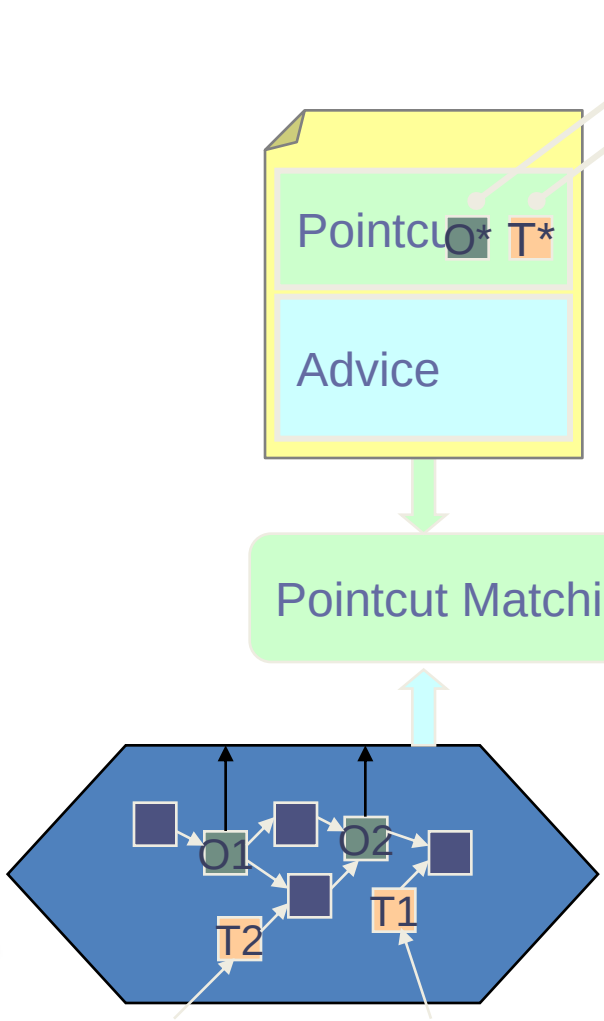
Le principe des Aspects d'Assemblage (AA)



Le principe des Aspects d'Assemblage (AA)



AA: Pointcut Matching




```
observed := /t*/ ;
timeout :=
    /ct*/  { a[substr($1,3)]=$1 }
    END    { for(i=1;i<=NR;i++){print a[i]} } ;
```

Les composants des Joinpoints

O1 T1

O2 T1

O1 T2

O2 T2

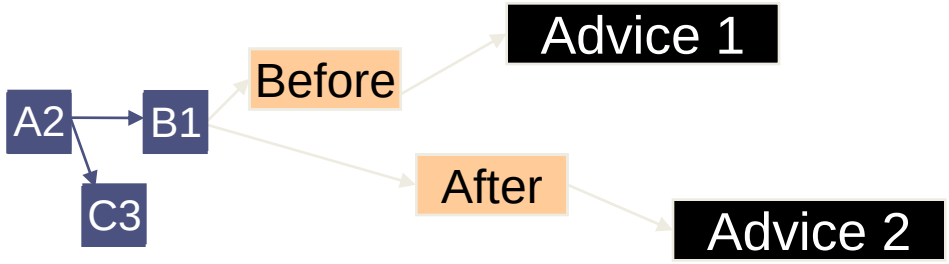
Enfin, les Advices se basent sur des méthodes (entrée) ou des événemnents (en sortie) comme join points

```
observed.^Out1 …
Timeout.check …
```

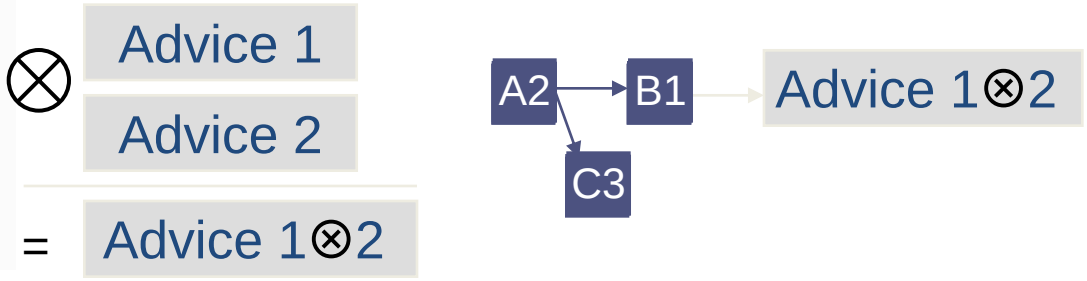
AA: Deux types d'Advices

- ✓ **Blackbox Advices :**
  - Tissage externe entre les advices
- ✓ **Greybox Advices :**
  - Nous connaissons en partie la sémantique des advices
  - Possibilité de Fusion en cas de conflit

```
SCHEMA Ex (observed):
observed.^Out1 -> Bcomp.do;  CALL /* before
observed.^Out2 -> CALL ; Bcomp.do; /* after
```

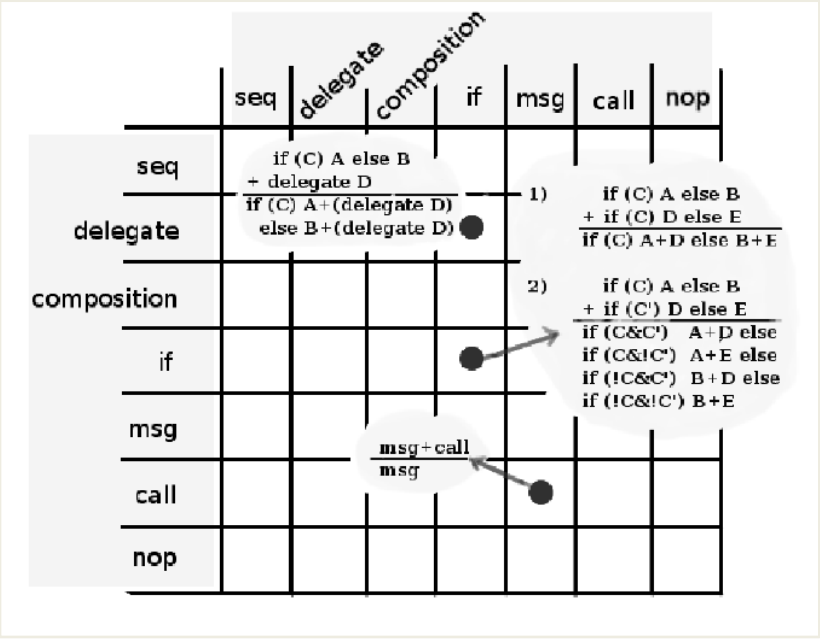



```
SCHEMA Ex (observed, timeout):
    observed.^Out ->
            (  IF ( timeout.Check ) CALL )
    timeout.Check ->
            (  timeout.Start ; CALL      )
```



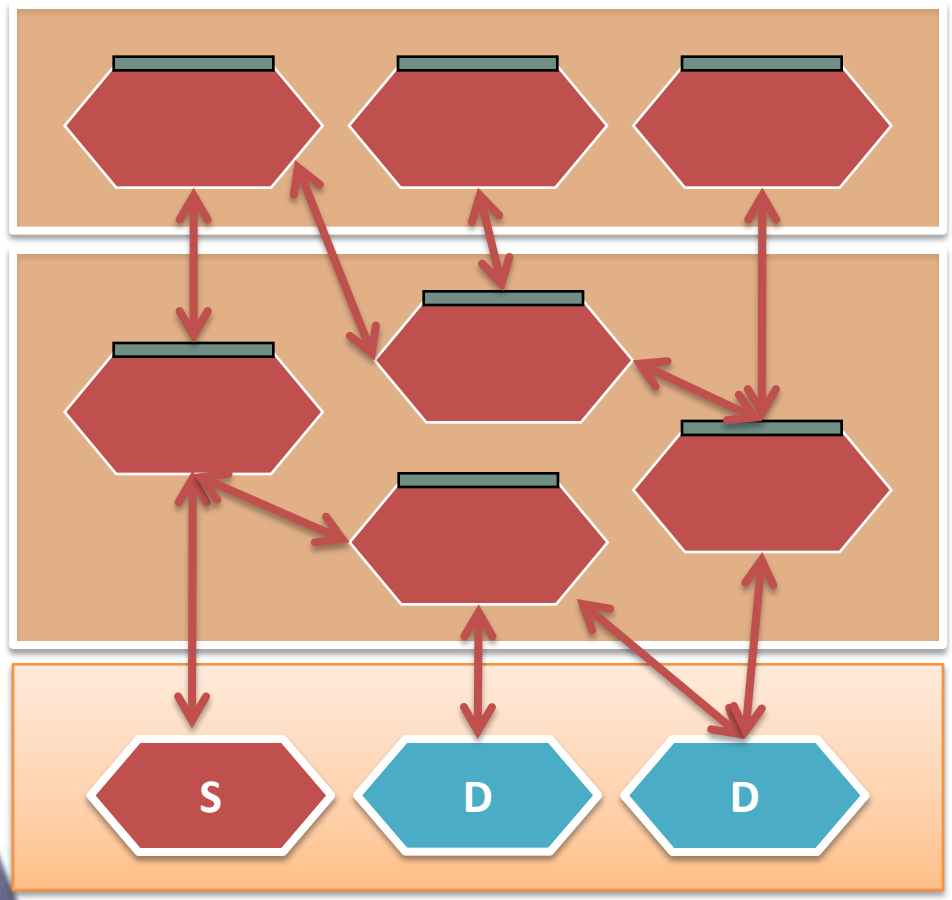
AA : Logique de fusion pour les Greybox Advice

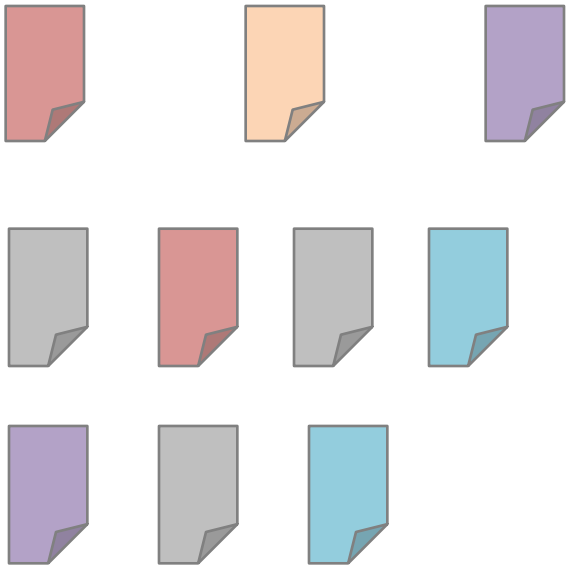
- ✓ Logique de Fusion
- ✓ Avec parfois des propriétés préétablies



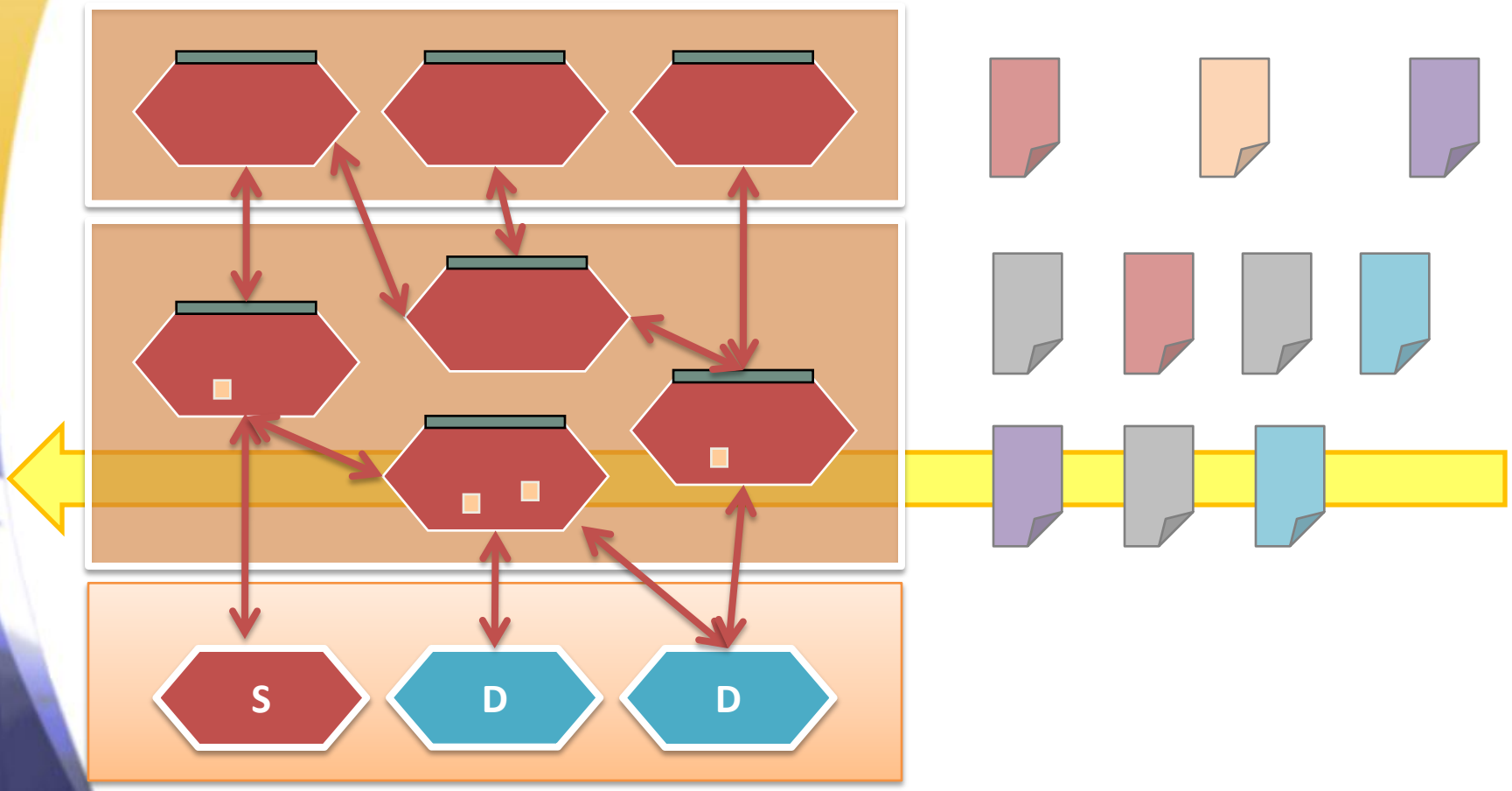
Commutativity : AA0 ⊗ AA1 = AA0 ⊗ AA1
Associativity : (AA0 ⊗ AA1) ⊗AA2 = AA0 ⊗ (AA1 ⊗ AA2)
Idempotence : AA0 ⊗ AA0 = AA0

Cycle d'adaptation SLCA / AA

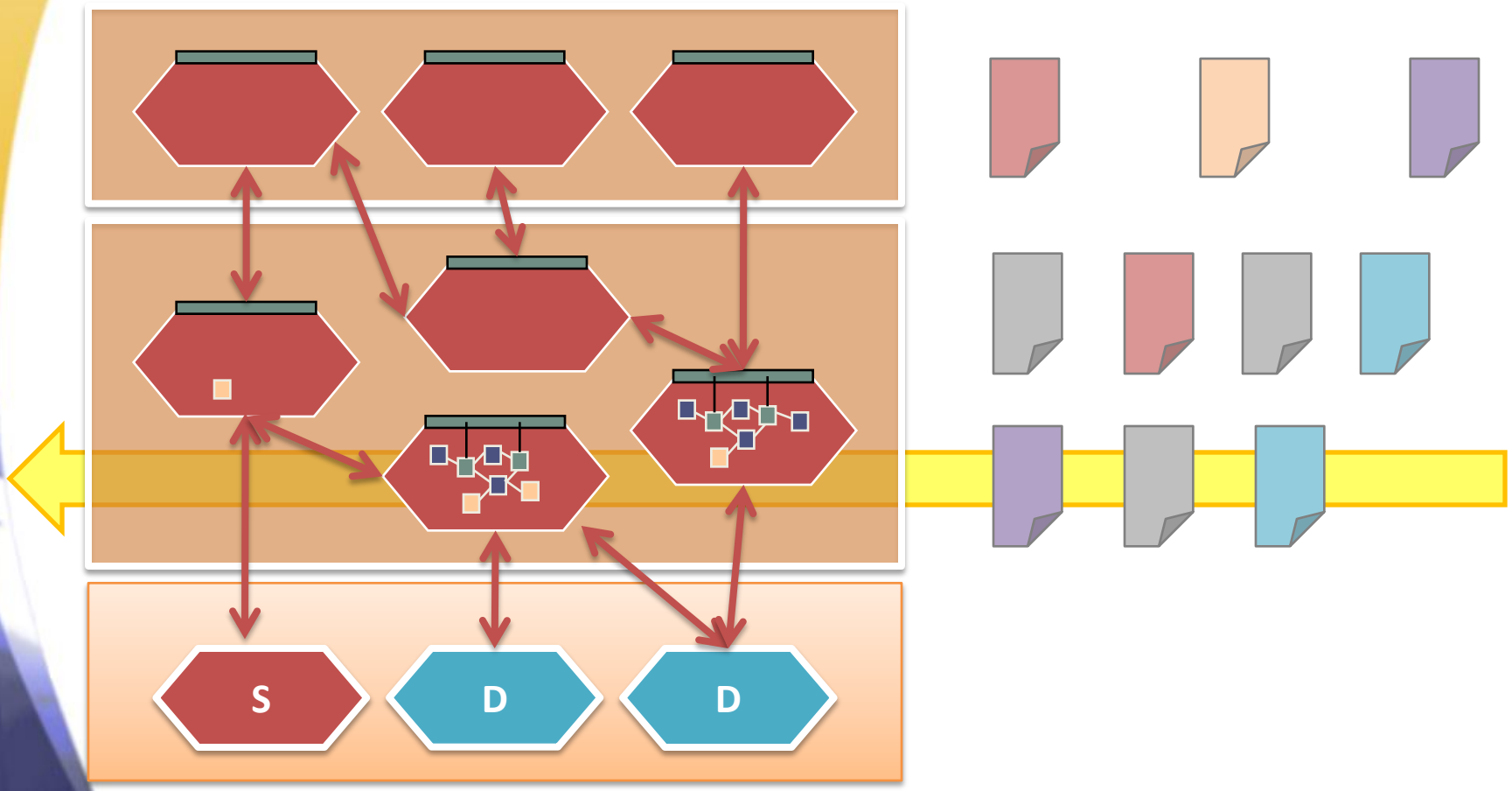




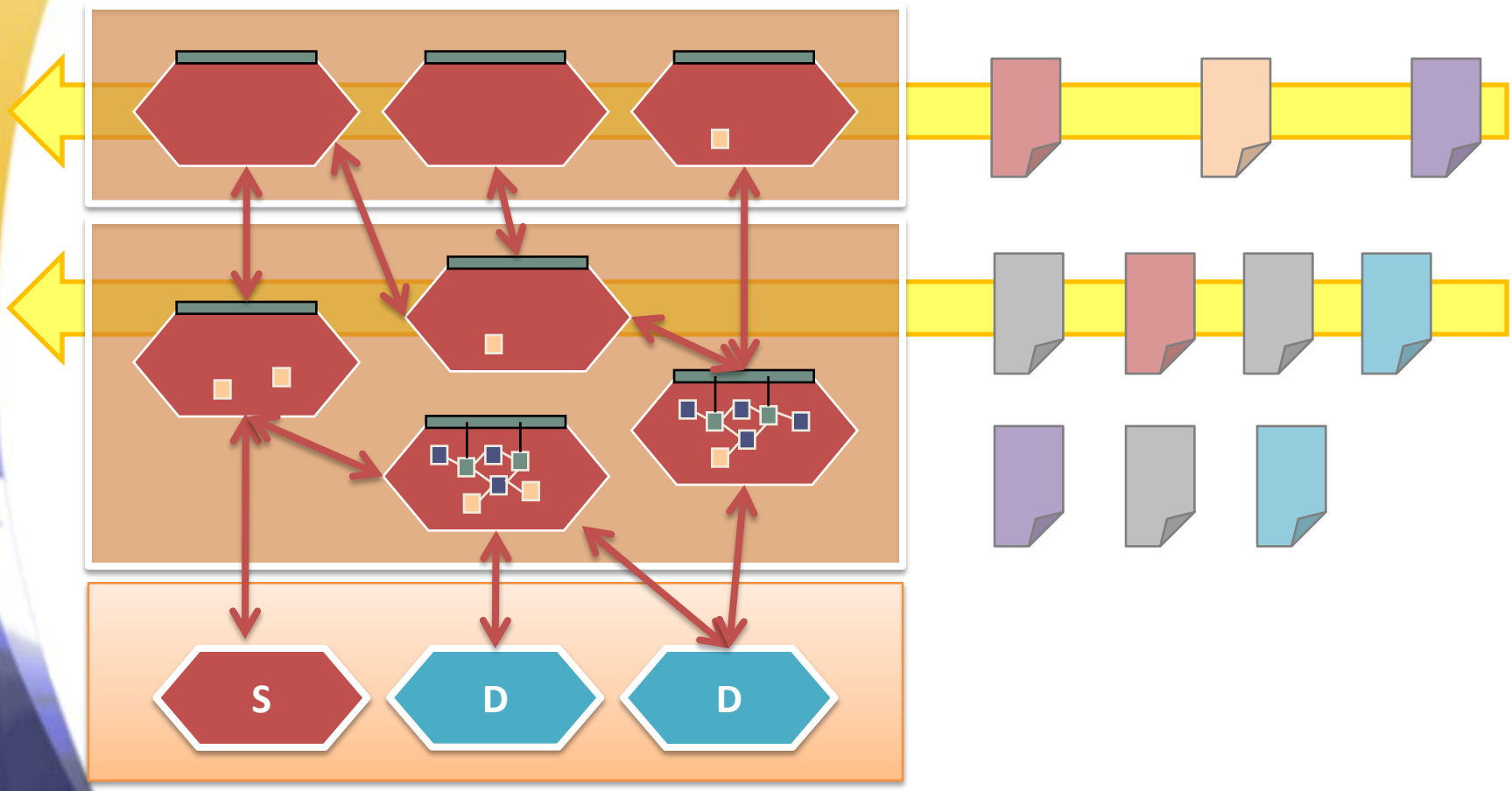
Cycle d'adaptation SLCA / AA



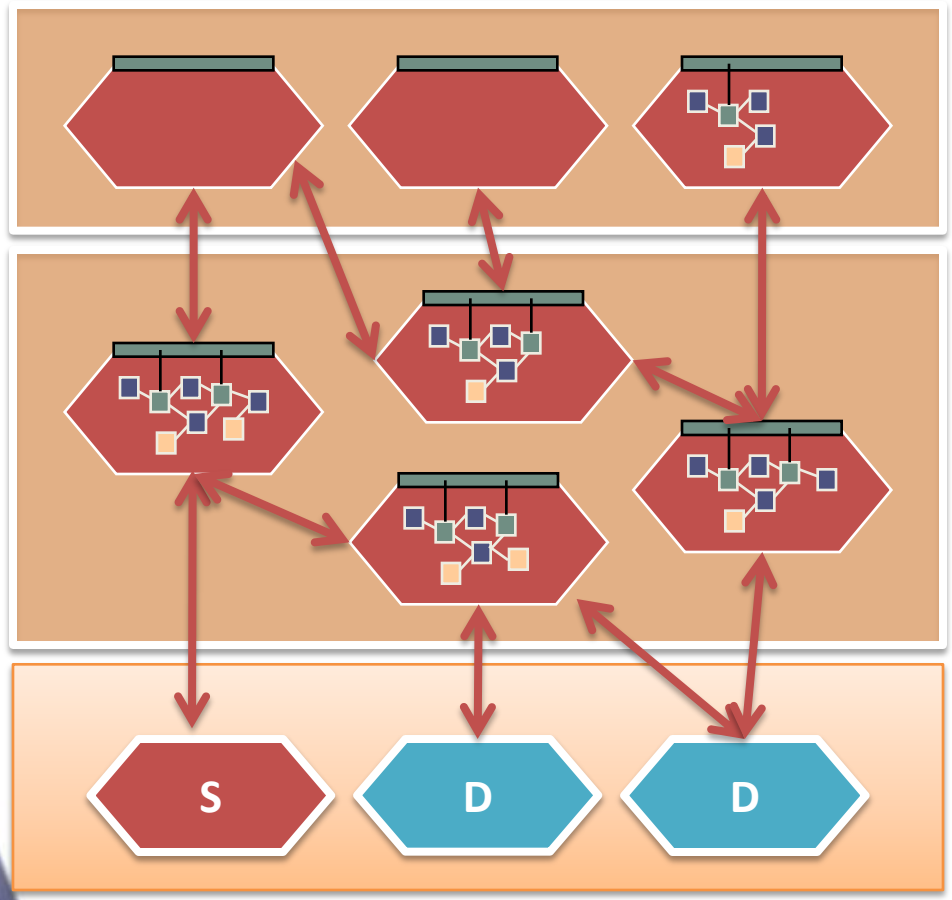
Cycle d'adaptation SLCA / AA

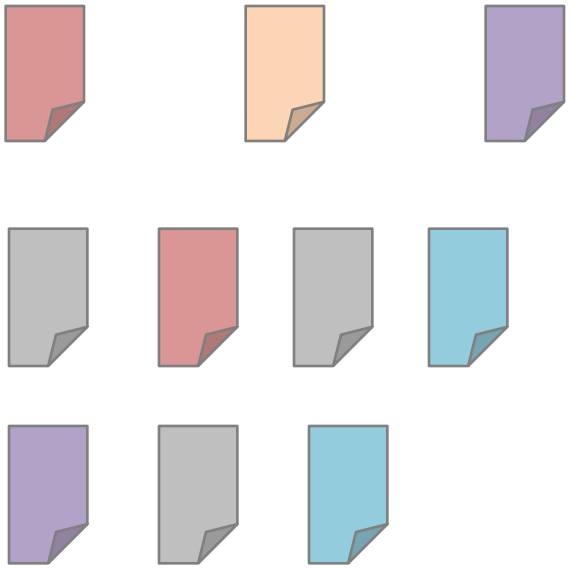


Cycle d'adaptation SLCA / AA

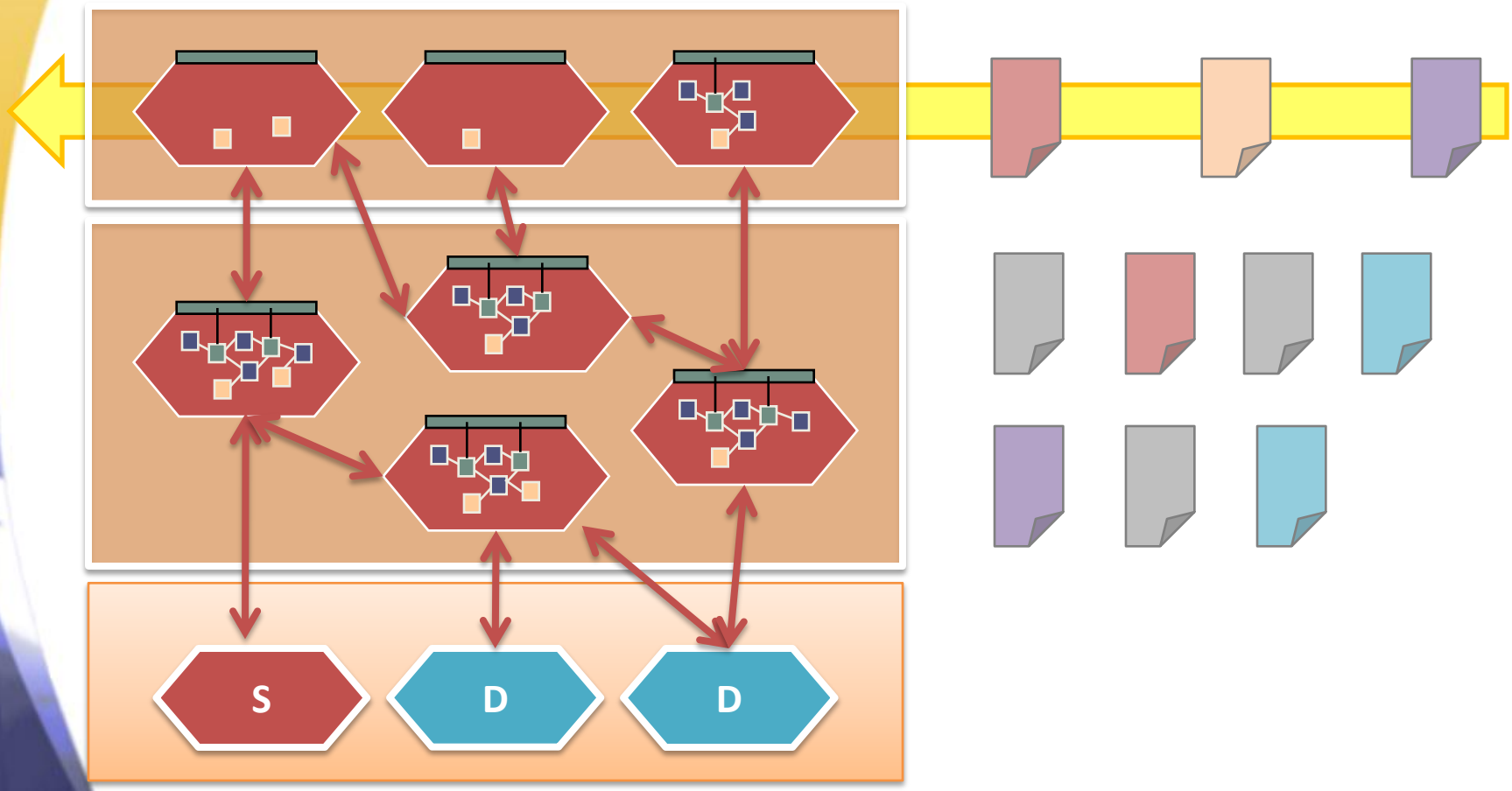


Cycle d'adaptation SLCA / AA



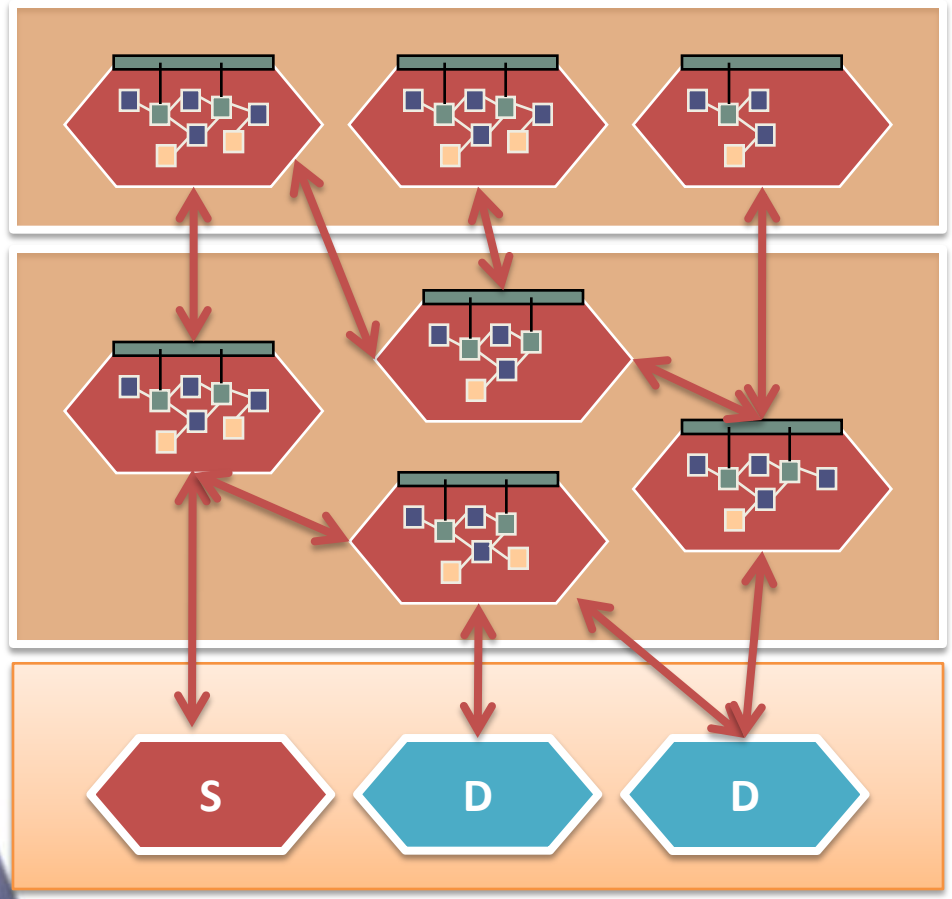


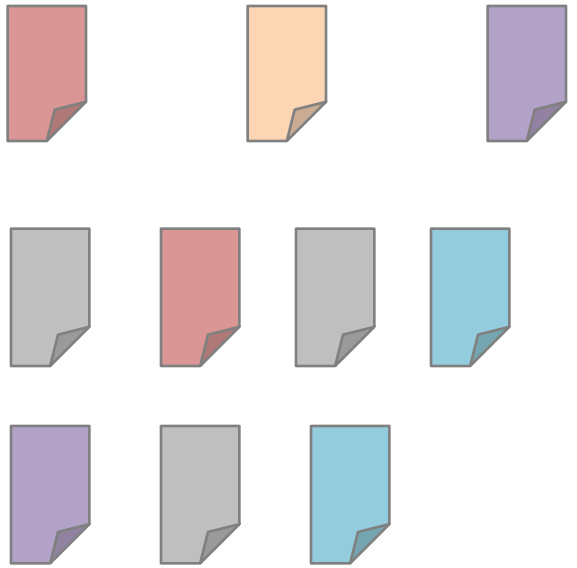
Cycle d'adaptation SLCA / AA



Cycle d'évolution SLCA / AA

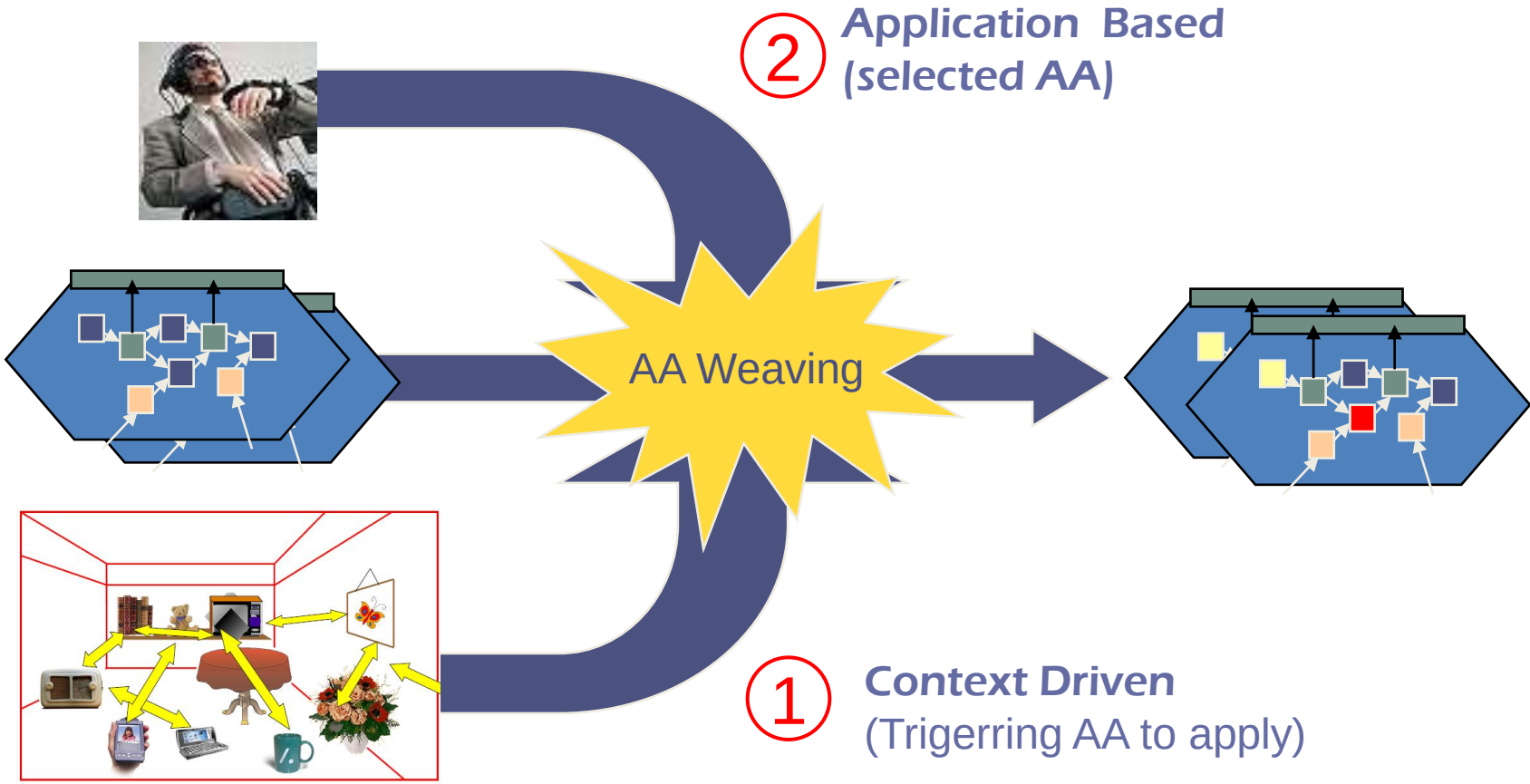
(*) De même pour les adaptations
par application/retrait d'AA



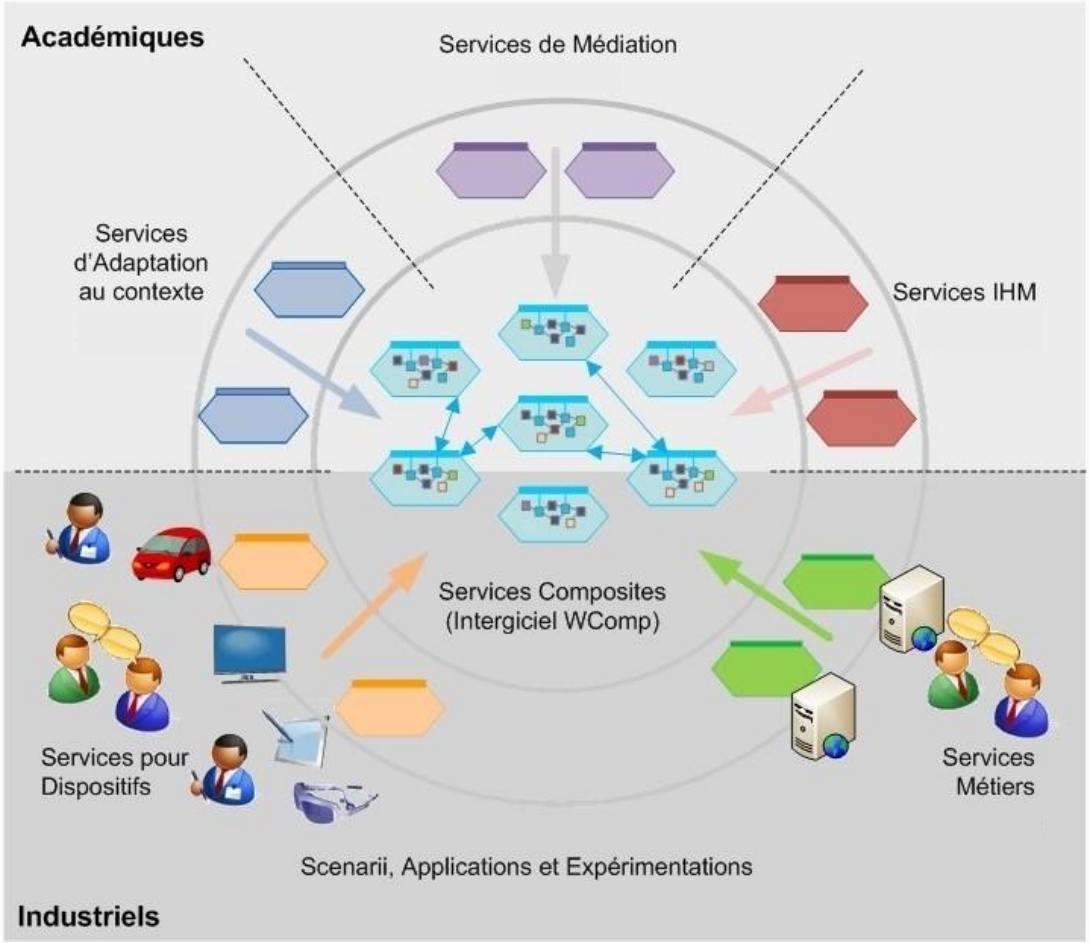


- Revue Int. : AOT'09
- Conf. Int. : RSP'06, MPAC'07 …
- Expérimentation : WComp (5 dépôt APP), Ubiquarium

DEMO


2
Application Based
(selected AA)
AA Weaving
1
Context Driven
(Trigerring AA to apply)

Conclusion et Perspectives dans le cadre de CONTINUUM



Questions ?