Propagation d'événements entre passerelles OSGi™

Didier Donsez*, Gaël Thomas (* puis +)

** Université Joseph Fourier (Grenoble 1)*

IMA IMAG/LSR/ADELE (UMR CNRS 5526)

+ Université Pierre et Marie Curie (Paris 6)

LIP6 SRC (UMR CNRS 7606)

Didier.Donsez@imag.fr, Gael.Thomas@lip6.fr

Diapositive de résumé

- Introduction
  - Motivations (i, ii)
  - Principes des ponts (i, ii, iii)
- Rappels
  - Bus à messages
  - Event Admin Service (i, ii)
- Conception des ponts
  - Architecture Event Admin – Pont (i, ii)
  - Correspondances entre Event Admin et ponts
- Expérimentation et validation
  - Event Admin – Pont Ivy
  - Event Admin – Pont Flash
  - Event Admin – Pont JMS
- Conclusion and perspectives
  - Q & A

Diapositive de résumé

- Introduction
  - Motivations (i, ii)
  - Principes des ponts (i, ii, iii)
- Rappels
  - Bus à messages
  - Event Admin Service (i, ii)
- Conception des ponts
  - Architecture Event Admin – Pont (i, ii)
  - Correspondances entre Event Admin et ponts
- Expérimentation et validation
  - Event Admin – Pont Ivy
  - Event Admin – Pont Flash
  - Event Admin – Pont JMS
- Conclusion and perspectives
  - *Q & A*

Motivations (i)

- Nombreuses applications distribuées orientées événements
  - Applications de collecte de données (RFID, Capteurs)
  - Médiation entre services (Enterprise Service Bus (ESB), …)
  - Applications collaboratives (CSCW)
  - Administration de passerelles (déploiement, gestion d'erreurs)
  - …
- Contraintes de ces applications
  - Communication distribuée par événements

  ⇒ Bus de communication distribué par événement
  - Chargement et déchargement dynamique de blocs de code

  ⇒ Plateforme de déploiement dynamique de code

Motivations (ii)

- OSGi devient le noyau universel de Java pour charger, décharger et lier du code dynamiquement
  - JSR 277 & JSR 291
  - J2ME, J2SE, et maintenant J2EE
- Spécifie un protocole de communication par événement

  EventAdmin depuis la R4
- Les événements sont échangés entre services locaux
  - Pas d'échange de messages inter-passerelles
  - Pas d'échange de messages avec d'autres types de plateformes

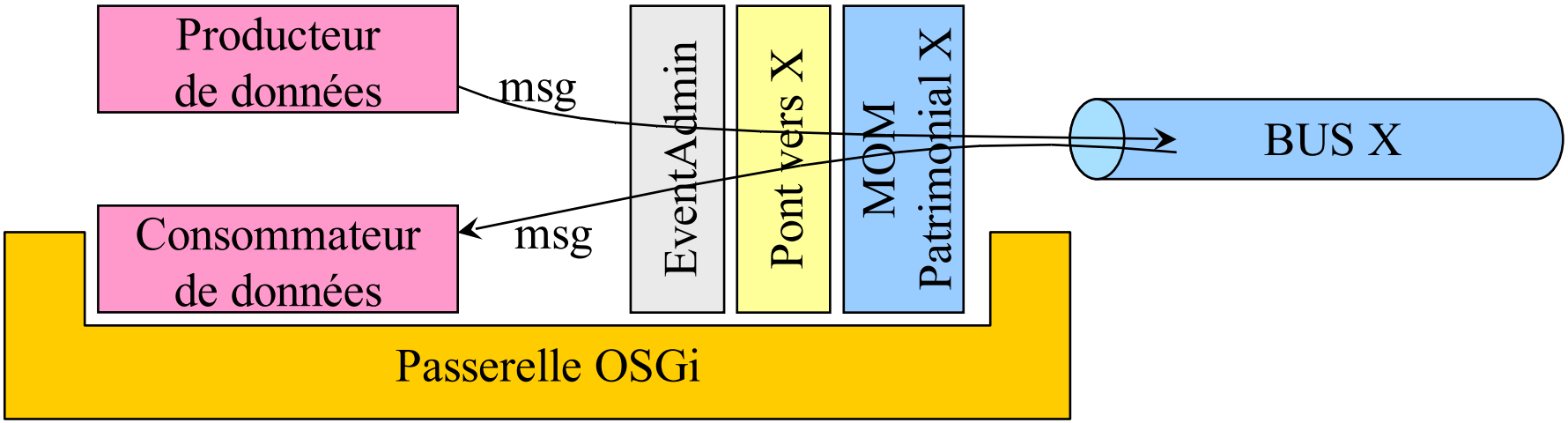
    (Serveurs J2EE, messges flash, e-mails…)

Principes et intérêt des ponts (i)

- Objectifs de nos travaux
  - Propager les événements de l'EventAdmin entre passerelles
  - Propager les événements de l'EventAdmin à l'extérieur des passerelles

  ⇒ Gérer différents types de bus de communication par message
- Comment

  Via des ponts (bridges) réutilisant les bus à messages (MOM) patrimoniaux



Principes et intérêt des ponts (ii)

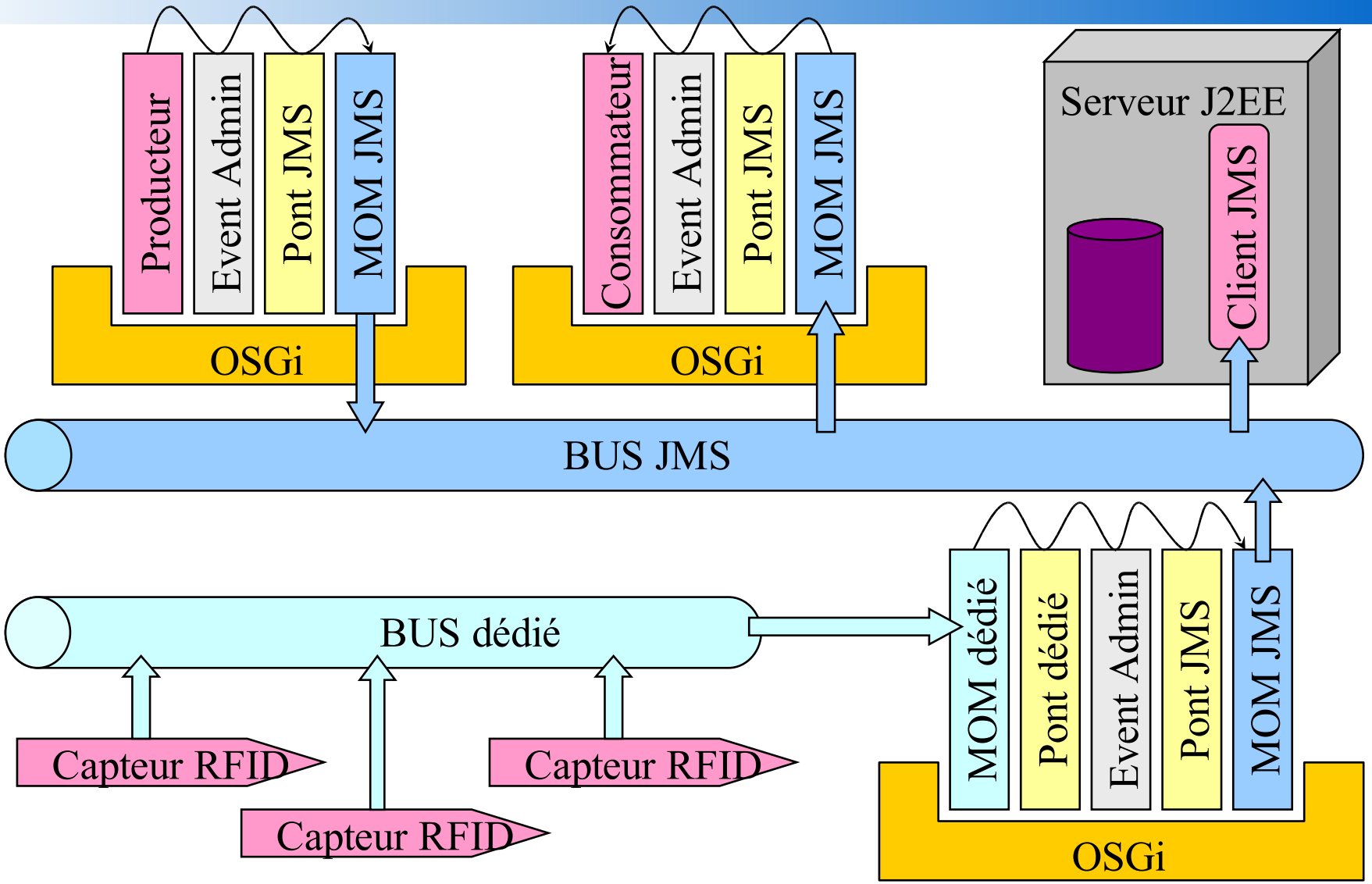
- Compatibilité avec la spécification EventAdmin
  - Consommateurs et producteurs de messages indépendants du MOM

  ⇒ Augmente la réutilisabilité des services

- Utilisation de plusieurs ponts simultanément
  - Passerelle entre différents MOM

  ⇒ Diminue l'hétérogénéité logicielle

Principes et intérêt des ponts (iii)



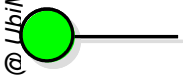
Diapositive de résumé

- Introduction
  - Motivations (i, ii)
  - Principes des ponts (i, ii, iii)
- Rappels
  - Bus à messages
  - Event Admin Service (i, ii)
- Conception des ponts
  - Architecture Event Admin – Pont (i, ii)
  - Correspondances entre Event Admin et ponts
- Expérimentation et validation
  - Event Admin – Pont Ivy
  - Event Admin – Pont Flash
  - Event Admin – Pont JMS
- Conclusion and perspectives
  - *Q & A*

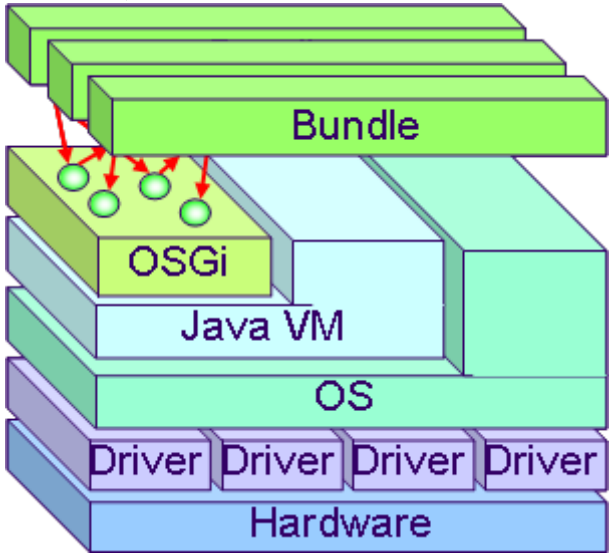
Principaux concepts d'OSGi

- Framework:
  - Environnement d'exécution pour bundles
    - Oscar (Objectweb) / Felix (Apache), Knopperfish, Equinox, SMF, ProSyst, Siemens VDO, …
  - Notification d'événements

- Bundles:
  - Unité d'exécution et de déploiement



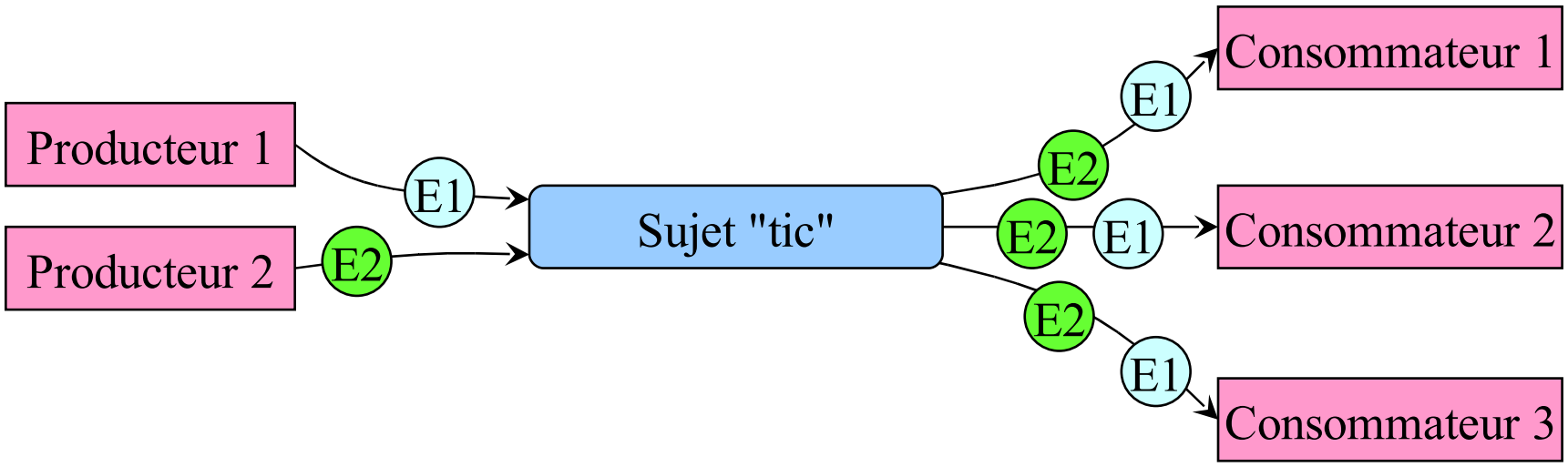
- Services:
  - Objets Java implantant un contrat



Bus à messages

Définitions :

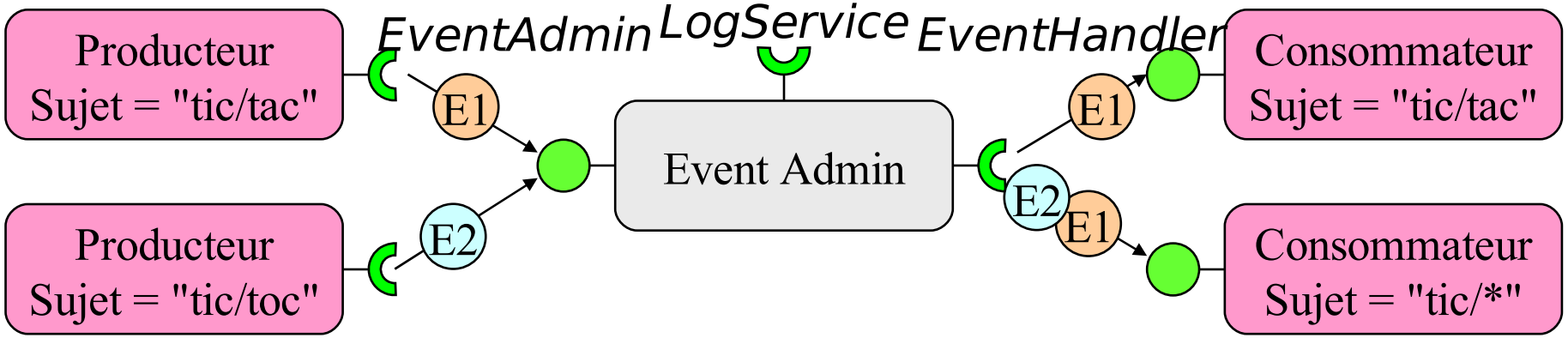
- Sujet (topic) : canal de communication partagé possédant un nom
- Producteur : écrit des messages
- Consommateur : lit les messages
- Action de poster : écrire un message
  - Synchrone : le producteur est synchronisé sur la réception du message
  - Asynchrone : le producteur n'attend pas la réception



Event Admin Service (i)

- Bus à message d'OSGi
  - Centralisé
  - Publication synchrone et asynchrone
- Terminologie
  - Event = sujet + contenu
  - EventAdmin = serveur du bus à message
  - Publisher : producteur
  - EventHandler : consommateur
- Remarque
  - Événements spéciaux d'OSGi liés aux cycles de vie des Services, bundles et framework
  - Gestion d'une liste noire des consommateurs défectueux
  - Gestion d'une arborescence de sujet (tic/tac/*)

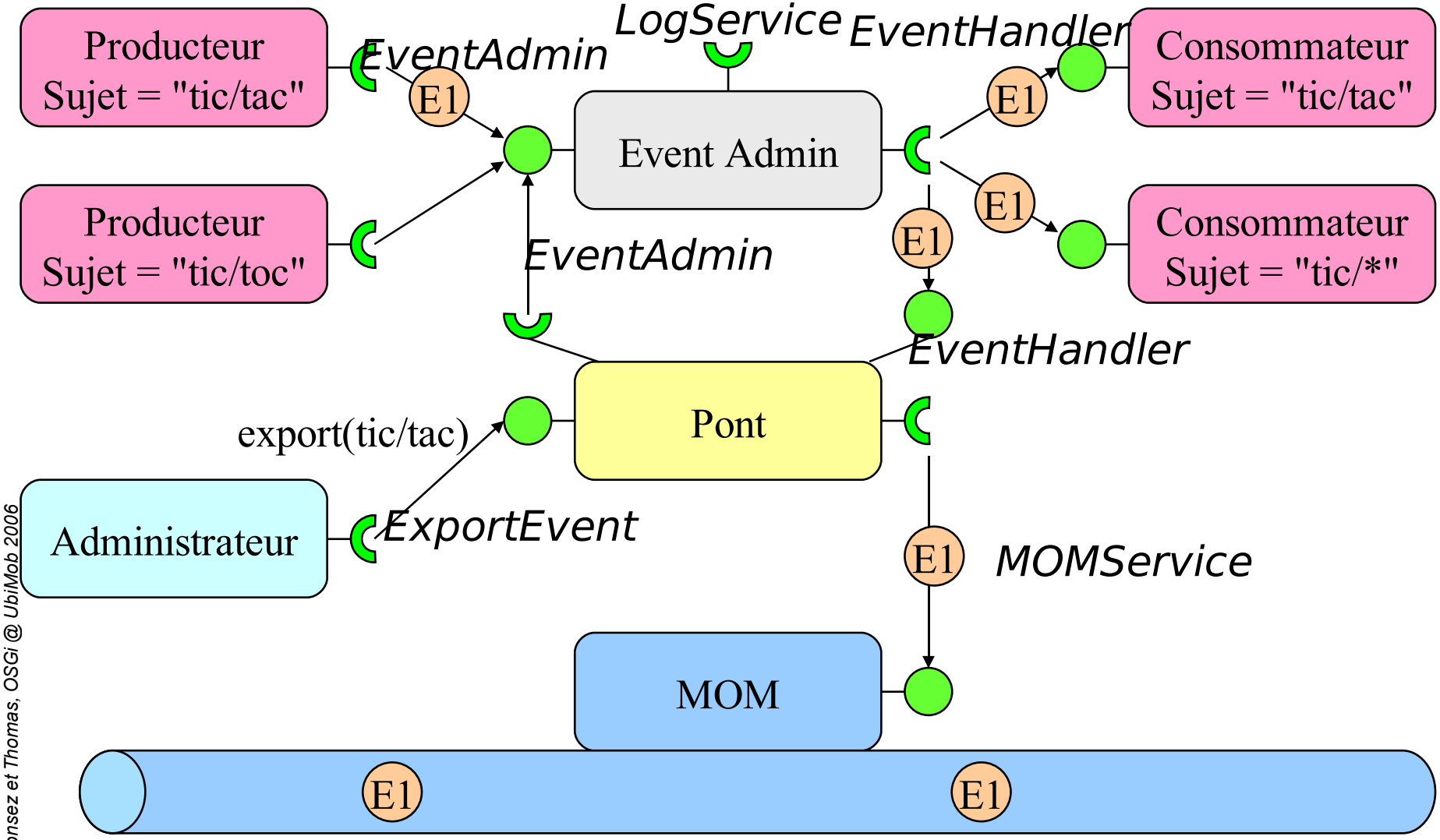
Event Admin Service (ii)



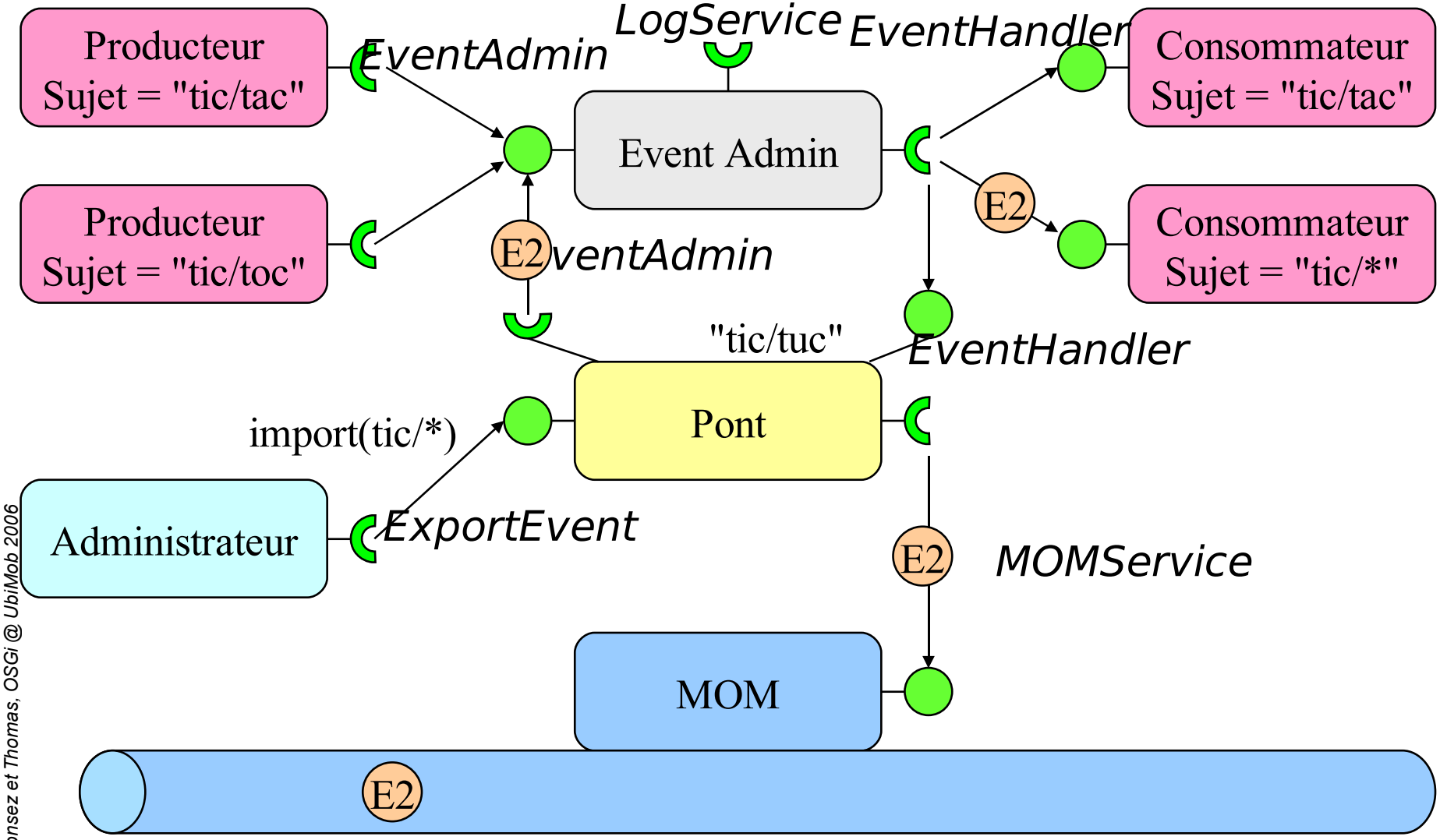
Diapositive de résumé

- Introduction
  - Motivations (i, ii)
  - Principes des ponts (i, ii, iii)
- Rappels
  - Bus à messages
  - Event Admin Service (i, ii)
- Conception des ponts
  - Architecture Event Admin – Pont (i, ii)
  - Correspondances entre Event Admin et ponts
- Expérimentation et validation
  - Event Admin – Pont Ivy
  - Event Admin – Pont Flash
  - Event Admin – Pont JMS
- Conclusion and perspectives
  - *Q & A*

Architecture Event Admin – Pont (i)



Architecture Event Admin – Pont (ii)



Correspondances entre Event Admin et ponts

- Propagation des données
  - Tout ne peut pas être propagé!!
    Messages non sérialisables ou dépendants de la plateforme (BundleID…)
  - ⇒ Déclaration explicite des sujets à propager

- Correspondance entre sujets EventAdmin/MOM
  - Tous les sujets EventAdmin passent par un unique sujet du MOM
  - ⇒ Saturation du réseau
  - A un sujet EventAdmin correspond un sujet du MOM
  - ⇒ Optimisation de l'utilisation de la bande passante
  - 👎 Ce n'est pas toujours possible

- Post synchrone/asynchrone
  - 👎 Ce n'est pas toujours possible

Diapositive de résumé

- Introduction
  - Motivations (i, ii)
  - Principes des ponts (i, ii, iii)
- Rappels
  - Bus à messages
  - Event Admin Service (i, ii)
- Conception des ponts
  - Architecture Event Admin – Pont (i, ii)
  - Correspondances entre Event Admin et ponts
- Expérimentation et validation
  - Event Admin – Pont Ivy
  - Event Admin – Pont Flash
  - Event Admin – Pont JMS
- Conclusion and perspectives
  - *Q & A*

Expérimentation et validation

- Trois expérimentations
  - Pont Ivy : protocole de diffusion sur un réseau local
  - Pont Flash : utilise des sockets pour communiquer
  - Pont JMS : spécification de bus à messages de Sun
- Couvrent différents types de communication par message
  - Ivy : diffusion, pas d'administration
  - Flash : unicast, pas d'administration
  - JMS : diffusion, administration

Event Admin – Pont Ivy

- Ivy : protocole de diffusion de messages sur une adresse multicast ou broadcast
  - Ne nécessite pas d'infrastructure particulière
  - Message codés en XML et reçus suivant des filtres
- Intérêt : diffuser les événements EA dans un réseau ad-hoc

  Communication inter-passerelles
- Principes :
  - Traduction des événements EA en XML
  - Communication par diffusion globale

    ⇒Tous les sujets EA exportés sont diffusés (saturation du réseau)
  - Uniquement post asynchrone
  - Pas d'administration

Event Admin – Pont Flash

- Flash : présentation graphique dynamique et interactive
  - Très utilisé dans le monde de l'infographie
  - Exécuté dans une machine virtuelle Flash
  - Canal de communication (local) avec l'extérieur (socket)
- Intérêt : en remplacement des canevas Java (swing…)
  - Canevas Java peu utilisés dans le monde de l'infographie
  - Présentation plus rapide à construire en Flash
- Principes :
  - Traduction des événements EA en XML
  - Association d'un client Flash à un unique pont
  - Communication point à point
    ⇒Tous les sujets EA exportés transitent dans une unique socket
  - Uniquement post asynchrone
  - Pas d'administration

Event Admin – Pont JMS

- JMS : spécification d'un bus à message en Java
  - Notions de sujet, de producteur, de consommateur
  - Intégré à la spécification J2EE (MessageDrivenBean)
- Intérêts : diffuser des messages dans un réseau structuré
  - Communication inter-passerelles
  - Communication avec des serveurs d'application J2EE
- Principes :
  - Traduction des événements EA en événements JMS
  - Association d'un sujet EA à un sujet JMS
    - ⇒ Seules les passerelles intéressées par un sujet reçoivent les messages
  - Uniquement post asynchrone
  - Basé sur l'implantation Joram (ObjectWeb)
  - Administration complexe (construction d'un réseau de serveurs JMS)
  - Meilleur passage à l'échelle en nombre de passerelles

Diapositive de résumé

- Introduction
  - Motivations (i, ii)
  - Principes des ponts (i, ii, iii)
- Rappels
  - Bus à messages
  - Event Admin Service (i, ii)
- Conception des ponts
  - Architecture Event Admin – Pont (i, ii)
  - Correspondances entre Event Admin et ponts
- Expérimentation et validation
  - Event Admin – Pont Ivy
  - Event Admin – Pont Flash
  - Event Admin – Pont JMS
- Conclusion and perspectives
  - Q & A

Conclusion and perspectives

- Propagation d'événements hors des passerelles OSGi
  - Transparent pour les producteurs/consommateurs EventAdmin
  - Communication inter-passerelles et vers d'autres applications
  - Passerelle entre différents MOM
- Travaux actuellement utilisés dans l'équipe pour
  - Service de médiation (transformation de données) entre Web Services
  - Suite logicielle RFID
- Perspectives
  - Intégration des travaux de Bob Brady sur ECF
    Pont entre EventAdmin et Eclipse Communication Framework
  - Étude et réalisation d'autres ponts
    Siena*, *GENA (UPnP), JINI Eventing, DPWS (WS-Eventing), JXTA, BIP, Jabber, SIP, CORBA CosEvent* …

*: en cours

Q & A