



**HAL**  
open science

# Déploiement de composants logiciels sur des équipements mobiles communicants : une approche coopérative

Hervé Roussain, Frédéric Guidec

► **To cite this version:**

Hervé Roussain, Frédéric Guidec. Déploiement de composants logiciels sur des équipements mobiles communicants : une approche coopérative. JC'05, Apr 2005, Le Croisic, France. pp.83-88. hal-00342647

**HAL Id: hal-00342647**

**<https://hal.science/hal-00342647>**

Submitted on 27 Nov 2008

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Déploiement de composants logiciels sur des équipements mobiles communicants : une approche coopérative

Hervé ROUSSAIN – Frédéric GUIDEC

Laboratoire Valoria – Université de Bretagne Sud  
{Herve.Roussain,Frederic.Guidec}@univ-ubs.fr

---

## Résumé

Le déploiement d'applications à base de composants logiciels implique l'obtention préalable des composants nécessaires à ce déploiement. Dans un réseau d'infrastructure, les composants visés peuvent aisément être téléchargés en mode client-serveur depuis un espace de stockage distant, souvent qualifié d'étagère à composants ou de dépôt de composants (*component repository*). Dans cet article nous nous intéressons plus spécifiquement au déploiement d'applications logicielles sur des équipements mobiles (eg ordinateurs portables et assistants numériques personnels) assemblés en réseau ad hoc autonome. Nous montrons que le schéma client-serveur traditionnel est inapproprié dans un tel contexte. Nous présentons ensuite un modèle que nous avons défini afin de supporter le déploiement de composants sur une base coopérative, chaque équipement mobile gérant sa propre étagère de composants, et interagissant en mode pair-à-pair avec les équipements voisins pour obtenir les composants qui lui font défaut, et faire bénéficier ses voisins des composants dont il dispose localement.

**Mots-clés :** Déploiement de composants logiciels, équipements mobiles communicants, réseaux ad hoc, systèmes pair-à-pair

---

## 1. Introduction

Les applications logicielles devenant de plus en plus complexes, une approche encouragée aujourd'hui pour maîtriser cette complexité consiste à concevoir ces applications en termes d'assemblages de composants logiciels. Les composants logiciels peuvent être vus comme des unités indépendantes, réutilisables et remplaçables destinées à remplir une fonction clairement définie au sein d'une application logicielle [1, 15]. Des travaux récents portant sur la problématique générale du déploiement d'applications à base de composants ont contribué à l'identification de phases élémentaires sur lesquelles repose ce déploiement. Il a ainsi été proposé de distinguer des phases relatives à la mise à disposition, l'acheminement, l'installation, la configuration, l'exécution, l'adaptation et la suppression des composants constitutifs d'une application [3, 10]. Dans cet article nous nous intéressons plus particulièrement aux deux premières de ces phases, à savoir la mise à disposition des composants dans un espace de stockage, désigné ici par le terme générique « d'étagère à composants » (*component repository*), et la gestion de leur acheminement, depuis ces étagères, jusqu'aux équipements constituant les cibles du déploiement.

Les travaux relatifs au déploiement de composants logiciels supposent dans leur grande majorité que ce déploiement doit être effectué sur des équipements connectés à un réseau d'infrastructure, tel qu'illustré dans la figure 1. Dans un tel réseau, certains équipements choisis parmi les plus stables et les plus accessibles peuvent avoir pour rôle d'héberger des étagères à composants, et mettre en œuvre des programmes serveur capables de délivrer à la demande les composants disponibles sur ces étagères. Les autres équipements du réseau peuvent alors se comporter en tant que simples clients vis-à-vis de ces serveurs, téléchargeant depuis les serveurs les composants requis lorsque le propriétaire (ou l'administrateur) d'un équipement client décide d'y déployer une nouvelle application, ou de mettre à jour une application existante.

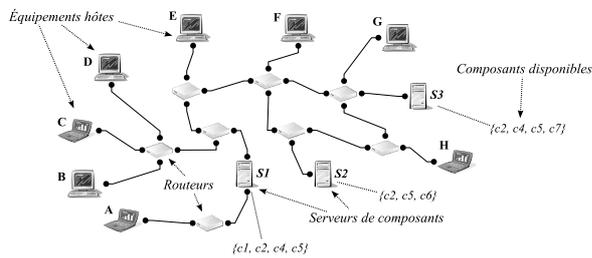


FIG. 1 – Illustration du déploiement de composants dans un réseau d’infrastructure

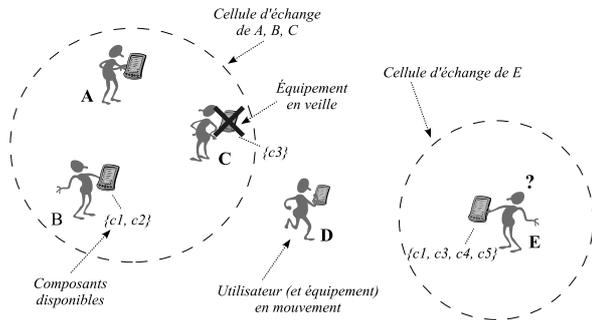


FIG. 2 – Illustration du déploiement de composants dans un réseau ad hoc dynamique

Depuis quelques années sont commercialisés des équipements mobiles et communicants, tels que des ordinateurs portables et des assistants numériques personnels (PDAs) dotés d’interfaces de communication sans fils de type Wi-Fi (IEEE 802.11). Ces interfaces permettent l’accès à un réseau d’infrastructure, mais elles peuvent aussi fonctionner en mode ad hoc, ce qui laisse entrevoir de nouvelles perspectives – mais aussi de nouveaux défis – pour le déploiement et la mise à jour des applications logicielles sur de tels équipements mobiles et communicants. En effet, dans un réseau fonctionnant en mode ad hoc chaque équipement peut échanger directement de l’information avec les équipements situés dans son voisinage immédiat [14]. La figure 2 illustre un réseau de ce type, constitué d’un ensemble d’ordinateurs portables et de PDAs. La caractéristique majeure d’un tel réseau est son aptitude à se constituer, et à évoluer ensuite de manière spontanée au gré de la volatilité des équipements considérés (ces équipements étant fréquemment mis en veille et réactivés par leurs propriétaires), et au gré de la mobilité des équipements. Une autre caractéristique est la connectivité fluctuante du réseau, qui peut se trouver fractionné – de manière plus ou moins épisodique – en « îlots » entre lesquels aucune communication n’est possible.

Pour les propriétaires d’ordinateurs portables et de PDAs dotés d’interfaces Wi-Fi, la perspective de pouvoir y déployer des applications à base de composants logiciels paraît tout aussi intéressante lorsque ces équipements participent à un réseau ad hoc que lorsqu’ils sont connectés à un réseau d’infrastructure. Cependant, les spécificités même des réseaux ad hoc nous amènent à reconsidérer la problématique générale du déploiement de composants logiciels dans ce contexte précis. En effet le schéma de déploiement de composants reposant sur le modèle client-serveur traditionnel n’est guère envisageable dans un réseau ad hoc, dans la mesure où aucun équipement n’est a priori suffisamment stable et accessible dans ce type de réseau pour qu’on puisse lui confier le rôle de serveur de composants. En revanche, chaque équipement participant au réseau est censé pouvoir fonctionner – au moins de manière transitoire – en toute autonomie, offrant à son propriétaire un certain nombre d’applications produites par assemblage de composants. Ce fonctionnement autonome nécessite que chaque équipement maintienne une étagère locale sur laquelle vont être entreposés des composants permettant de bâtir les applications intéressant l’utilisateur. Dès lors, un schéma coopératif peut être mis en œuvre, dans lequel chaque équipement participant au réseau ad hoc peut fournir aux équipements voisins des copies des composants disponibles sur son étagère locale, tout en ayant lui-même la possibilité de se procurer auprès de ses voisins des composants qui lui font défaut.

## 2. Vue d’ensemble de la plate-forme JASON

Dans cette section, nous présentons la plate-forme JASON (*Java Ad hoc Services on ad hOc Networks*), destinée à supporter le déploiement d’applications à base de composants logiciels sur des équipements mobiles capables de communiquer en mode ad hoc.

Cette plate-forme est construite suivant un modèle en couches (voir figure 3). La couche supérieure supporte l’exécution des applications logicielles considérées. Dans la version actuelle de la plate-forme JASON, ce support est assuré par l’environnement d’exécution JAMUS, qui est dédié à l’hébergement

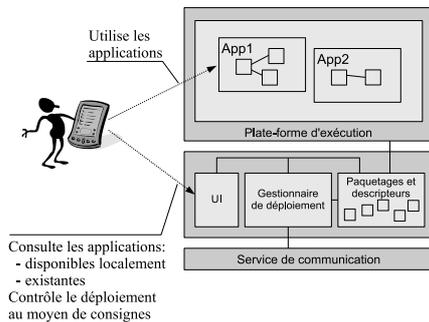


FIG. 3 – Vue d'ensemble de la plate-forme de déploiement JASON

```

<package-descriptor>
  <general-information
    name="JMessengerImpl"
    version="1.3"
    provider="Laboratoire Valoria"
    category="communication/messaging"
    summary="JMessenger is a P2P messenger"/>
  <dependencies>
    <required package="JMessengerUI" version="1.2"/>
    <required package="P2PAsyncDissemination"/>
    <required package="AddressBook" version="2.0"/>
  </dependencies>
</package-descriptor>

```

FIG. 4 – Exemple de descripteur de paquetage

sécurisé de code mobiles non sûrs [9]. L'interfaçage de la plate-forme JASON avec des *frameworks* d'exécution tels que Julia (implémentant le modèle Fractal [2]) ou OSCAR (implémentant le modèle orienté services OSGi [6]) est également à l'étude.

La couche inférieure de la plate-forme offre quant à elle des services de communication. Ces services sont présentés en détails dans [5]. Ils sont spécifiquement dédiés à la dissémination de documents au sein d'un réseau sans fils fonctionnant en mode ad hoc.

La couche intermédiaire de notre modèle fait l'objet du reste de cet article. Cette couche a pour fonction principale d'assurer, de manière coopérative, la dissémination de composants logiciels au sein d'une communauté d'équipements mobiles interagissant en mode ad hoc. Elle met en œuvre une étagère à composants, sur laquelle des composants vont pouvoir être entreposés et mis à disposition de la plate-forme d'exécution qui constitue la couche supérieure du modèle. Elle comprend en outre un gestionnaire de déploiement, dont le rôle est d'interagir avec l'utilisateur afin de lui présenter les applications disponibles localement, et de lui proposer le cas échéant le déploiement de nouvelles applications. Le gestionnaire de déploiement interagit en outre avec les gestionnaires pairs situés sur des équipements voisins afin de se procurer auprès d'eux les composants qui vont lui permettre de satisfaire les consignes de déploiement exprimées par l'utilisateur local.

### 2.1. Composants, applications et paquetages

Le déploiement de composants implique le transport de ces composants à travers le réseau, et leur stockage sur des étagères. Pour ce faire les composants sont encapsulés dans des paquetages, qui constituent donc des unités de stockage et de transport pour les composants. Par ailleurs les paquetages peuvent également encapsuler des données requises par une application, ou bien encore la description de l'architecture d'une application (telle que la description d'un assemblage de composants CCM [11], ou celle d'une architecture Fractal [2]).

#### Descripteurs de paquetages

À chaque paquetage est associé un descripteur, qui apporte des informations sur l'identité du paquetage, sur son contenu, son numéro de version, etc. Le descripteur de paquetage est en général embarqué dans le paquetage lui-même. Il peut également être extrait de ce paquetage et manipulé indépendamment de celui-ci.

Dans la version actuelle de la plate-forme JASON, les descripteurs de paquetages sont exprimés en XML. La figure 4 présente ainsi le descripteur (simplifié) d'un paquetage encapsulant le composant principal d'une application de messagerie. Ce descripteur fait état de dépendances vis-à-vis d'autres paquetages. De telles dépendances entre paquetages peuvent par exemple traduire des dépendances entre les éléments de code (source ou exécutable) embarqués dans ces paquetages. Elles peuvent également exprimer des dépendances entre un paquetage encapsulant la description de l'architecture d'une application, et les paquetages encapsulant les composants constitutifs de cette architecture. Lorsque le propriétaire d'un équipement donne au gestionnaire de déploiement la consigne de déployer localement une certaine application, le gestionnaire de déploiement peut ainsi s'appuyer sur les dépendances exprimées

dans le descripteur de cette application afin de déterminer quels sont les paquetages dont il dispose déjà localement, et quels sont ceux qu'il va lui falloir essayer de se procurer auprès des équipements voisins.

### **Étagères de paquetages**

Les composants logiciels étant encapsulés dans des paquetages, la plate-forme JASON met en œuvre une étagère de paquetages plutôt qu'une étagère de composants proprement dite. Cette étagère peut également héberger des descripteurs de paquetages. Le fait d'entreposer sur l'étagère certains descripteurs – et notamment les descripteurs d'applications – permet au gestionnaire de déploiement de présenter à l'utilisateur la liste des applications susceptibles d'être déployées localement s'il en fait la demande, les paquetages référencés par ces descripteurs étant disponibles sur des équipements du voisinage.

La gestion de l'étagère de paquetages est assurée par le gestionnaire de déploiement. Compte tenu des ressources de stockage limitées offertes par les équipements mobiles, le gestionnaire de déploiement doit notamment supprimer de l'étagère les descripteurs et paquetages de composants jugés obsolètes ou inutiles, ou bien encore les descripteurs de paquetages considérés comme inaccessibles dans le voisinage. Cette activité réalisée en tâche de fond par le gestionnaire de déploiement implique la définition et la mise en œuvre d'heuristiques dont la description détaillée dépasse le cadre de cet article.

### **2.2. Interactions entre gestionnaires de déploiement voisins**

Outre la gestion de l'étagère locale, le gestionnaire de déploiement a pour mission d'assurer la collecte des paquetages nécessaires au déploiement d'une application, conformément aux consignes exprimées par l'utilisateur. Pour ce faire il doit interagir avec les gestionnaires pairs installés sur les équipements du voisinage, tout en aidant dans le même temps ces gestionnaires pairs à satisfaire les consignes qu'ils ont eux-même reçu de leurs utilisateurs. Plusieurs modes d'interaction entre gestionnaires de déploiement voisins sont définis et peuvent être mis en œuvre, de manière alternative ou complémentaire, dans la plate-forme JASON. Ces modes d'interaction sont implémentés en s'appuyant sur les services élémentaires fournis par la couche de communication de la plate-forme. Chaque gestionnaire de déploiement peut ainsi :

- annoncer à son voisinage les paquetages disponibles sur l'étagère locale ;
- découvrir les paquetages disponibles sur les équipements voisins ;
- rechercher dans le voisinage les paquetages nécessaires au déploiement d'une application dont l'utilisateur local a demandé qu'elle soit déployée localement ;
- solliciter de la part d'un équipement voisin la transmission d'un paquetage ;
- réceptionner un paquetage en provenance d'un équipement voisin, et déposer ce paquetage sur l'étagère locale.

### **Annnonce de disponibilité de paquetages**

Le mécanisme d'annonce permet au gestionnaire de déploiement d'un équipement de faire savoir aux équipements voisins quels sont les paquetages dont il dispose localement. Une annonce est un document embarquant les descripteurs des paquetages dont le gestionnaire de déploiement local souhaite faire connaître la disponibilité. Ce document peut être diffusé à l'intention de tous les équipements voisins, ou adressé à certains équipements spécifiquement. Son émission peut en outre être réalisée ponctuellement, périodiquement, ou suite à l'apparition d'un nouvel équipement dans le voisinage (la couche de communication de la plate-forme JASON offre des services capables de gérer la réémission périodique, ou encore l'émission sur événement, des documents qui lui sont confiés).

Lors de la réception d'une annonce provenant d'un équipement voisin, le gestionnaire de déploiement de l'équipement récepteur peut examiner les descripteurs contenus dans cette annonce, et les confronter avec le contenu de l'étagère locale. Pour chaque descripteur ainsi considéré, cette opération peut se solder par :

- le dépôt du descripteur sur l'étagère locale : cette action se justifie notamment lorsque le descripteur est un descripteur d'application, dans la mesure où le gestionnaire d'application ayant ainsi « appris » l'existence d'une nouvelle application pourra ensuite proposer à l'utilisateur de déployer cette application localement s'il le souhaite.
- l'émission d'une requête adressée à la source de l'annonce, sollicitant de sa part le transfert du paquetage référencé par le descripteur : cette action se justifie lorsque le paquetage dont un voisin vient

- d'annoncer la disponibilité fait partie des paquetages recherchés par le gestionnaire de déploiement local pour satisfaire une consigne de déploiement exprimée par l'utilisateur ;
- aucune action : le descripteur est tout simplement ignoré. Ceci se justifie par exemple lorsqu'un exemplaire du paquetage considéré est déjà présent sur l'étagère locale.

### Recherche et découverte de paquetages

Les mécanismes de découverte et de recherche de paquetages constituent une approche alternative – ou complémentaire – aux mécanismes évoqués précédemment. La découverte et la recherche de paquetages impliquent tous deux la diffusion de requêtes par le biais desquelles le gestionnaire local va inviter ses voisins à annoncer les paquetages dont ils disposent. Cette approche permet notamment à un équipement qui vient d'être réactivé suite à une longue phase de veille d'identifier rapidement quels sont les paquetages disponibles dans son nouveau voisinage, sans attendre qu'une annonce soit faite spontanément par l'un des équipements voisins. Elle permet également à un gestionnaire de déploiement ayant reçu pour consigne de déployer une certaine application de rechercher activement les paquetages qui lui font encore défaut pour mener à bien cette opération.

### 3. Travaux apparentés

Des travaux proposant une approche coopérative (ou pair-à-pair) du déploiement d'applications et de composants logiciels ont été proposés au dessus de SoftwareDock [7] et de OSGi [4]. SoftwareDock est un *framework* pour le déploiement distribué, utilisant des agents mobiles comme support au transport des composants. [4] propose d'organiser les équipements participant aux déploiement en réseau pair-à-pair à clé de hash. Ces deux approches se distinguent de la nôtre par le fait qu'elles considèrent principalement le déploiement de composants dans des réseaux d'infrastructure, l'objectif étant d'équilibrer la charge du réseau et des serveurs, ou encore de tolérer des pannes temporaires de ces derniers.

De nombreux travaux portant sur la tolérance aux déconnexions [8, 12] peuvent contribuer à aider au déploiement de composants dans des environnements temporairement déconnectés. Cependant ces travaux visent essentiellement à proposer des façons d'émuler des opérations connectées dans des environnements temporairement déconnectés. De notre point de vue elles se prêtent donc mieux au déploiement dans des réseaux sans fils à couverture discontinue (de type *hot spot*) qu'au déploiement dans des réseaux ad hoc proprement dits.

Le modèle adopté dans la plate-forme JASON présente des similitudes marquées avec le modèle SATIN [16]. Ce dernier associe un modèle de composant et un modèle de déploiement, et vise spécifiquement au déploiement de composants logiciels dans les réseaux ad hoc. Il se base sur l'utilisation de primitives d'auto-organisation de la mobilité du code, à partir desquelles il est possible de construire des systèmes mobiles auto-organisés. Le support à l'acheminement des composants est présenté comme permettant la reconfiguration d'architectures de composants par le support de la migration de code. Ce support repose sur des mécanismes offerts par EMMA [13], un intergiciel permettant la dissémination « épidémique » de messages, très semblable dans son principe à celui que nous proposons dans [5], et qui constitue la couche de communication de la plate-forme JASON.

### 4. Conclusion et perspectives

Dans cet article nous avons montré que le déploiement d'applications à base de composants logiciels sur des ordinateurs portables et des PDAs assemblés en réseaux ad hoc ne peut se satisfaire du schéma de déploiement habituel consistant à télécharger les composants requis depuis des serveurs hébergeant des étagères à composants (ou dépôts de composants). Nous avons ensuite présenté dans les grandes lignes un modèle coopératif que nous proposons afin d'assurer le déploiement de composants logiciels dans ce type d'environnement.

Ce modèle est mis en œuvre dans le cadre de la plate-forme JASON. Dans l'état actuel du développement de cette plate-forme, l'installation et l'exécution d'applications s'effectuent au sein de l'environnement JAMUS, dédié à l'hébergement sécurisé de codes applicatifs considérés comme non sûrs. L'interfaçage de la plate-forme JASON avec des environnements d'exécution tels que Julia [2] et OSCAR [6] est à l'étude, ce qui devrait à terme permettre à JASON de supporter le déploiement d'applications Fractal et de services de type OSGi.

Nous travaillons également à la définition et à la mise en œuvre d'heuristiques destinées à guider l'activité du gestionnaire de déploiement qui, dans la plate-forme JASON, a pour rôles multiples de gérer une étagère de composants locales, d'interagir avec l'utilisateur (en acceptant notamment les consignes de déploiement exprimées par celui-ci), et d'échanger des composants avec les gestionnaires pairs situés sur les équipements du voisinage.

### Remerciements

Ce travail est financé par le Conseil Régional de Bretagne dans le cadre du programme « Renouvellement des compétences dans les laboratoires ». Contrat référencé B/1042/2002/012/MASC.

### Bibliographie

1. Felix Bachmann, Len Bass, Charles Buhman, Santiago Comella-Dorda, Fred Long, John Robert, Robert Seacord, and Kurt Wallnau. Technical concepts of component-based software engineering. Technical report, Carnegie Mellon University – Software Engineering Institute, 2000.
2. Eric Bruneton, Thierry Coupaye, Matthieu Leclercq, Vivien Quéma, and Jean-Bernard Stefani. An Open Component Model and Its Support in Java. In *Component-Based Software Engineering, 7th International Symposium*, pages 7 – 22. Springer-Verlag Heidelberg, 2004.
3. Antonio Carzaniga, Alfonso Fuggetta, Richard S. Hall, Dennis Heimbigner, André van der Hoek, and Alexander L. Wolf. A characterization framework for software deployment technologies. Technical Report CU-CS-857-98, Dept. of Computer Science, University of Colorado, Avril 1998.
4. Stéphane Frénot. Gestion du déploiement de composants sur réseau p2p. In *Decor'04*, pages 113–124, 2004.
5. Frédéric Guidec and Hervé Roussain. Asynchronous Document Dissemination in Dynamic Ad Hoc Networks. In *Second International Symposium on Parallel and Distributed Processing and Applications (ISPA'04)*, Hong-Kong, China, Décembre 2004.
6. Richard S. Hall and Humberto Cervantes. An OSGi Implementation and Experience Report. In *IEEE Consumer Communications and Networking Conference*, Janvier 2004.
7. Richard S. Hall, Dennis Heimbigner, and Alexander L. Wolf. A cooperative approach to support software deployment using the software dock. In *International Conference on Software Engineering*, pages 174–183, 1999.
8. Nabil Kouici, Denis Conan, and Guy Bernard. Caching Components for Disconnection Management in Mobile Environments. In *International Symposium on Distributed Objects and Applications*, Octobre 2004.
9. Nicolas Le Sommer and Frédéric Guidec. Jamus : une plate-forme d'accueil sécurisée pour le code mobile. In *Langages et Modèles à Objets (LMO 2003)*, L'objet, pages 203–215. Hermes Science, Janvier 2002.
10. Vincent Lestideau. *Modèles et environnement pour configurer et déployer des systèmes logiciels*. PhD thesis, Université de Savoie, Décembre 2003.
11. Raphaël Marvie, Philippe Merle, Jean-Marc Geib, and Mathieu Vadet. OpenCCM : une plate-forme ouverte pour composants CORBA. In *Actes de la seconde Conférence Française sur les Systèmes d'Exploitation (CFSE'2)*, Paris, France, Avril 2001.
12. Marija Mikic-Rakic and Nenad Medvidovic. Toward a Framework for Classifying Disconnected Operation Techniques. In *ICSE 2003 Workshop on Software Architectures for Dependable Systems*, 2003.
13. Mirco Musolesi, Cecilia Mascolo, and Stephen Hailes. Adapting Asynchronous Messaging Middleware to Ad Hoc Networking. In *2nd International Workshop on Middleware for Pervasive and Ad-Hoc Computing (MPAC2004)*, 2004.
14. Charles Perkins. *Ad Hoc Networking*, pages 2–3. Addison-Wesley, 2001.
15. Clemens Szyperski. *Component Software : Beyond Object-Oriented Programming*. ACM Press, Addison-Wesley, 1998.
16. Stefanos Zachariadis, Cecilia Mascolo, and Wolfgang Emmerich. SATIN : A component model for mobile self organisation. In *CoopIS/DOA/ODBASE (2)*, pages 1303–1321, 2004.