

Contributions aux systèmes répartis en environnements ubiquitaires : adaptation, sensibilité au contexte et tolérance aux fautes

Soutenance HDR de **UPMC**
SORBONNE UNIVERSITÉS

Denis Conan

Institut Mines-Télécom, Télécom SudParis
CNRS SAMOVAR UMR 5157, ACMES/MARGE

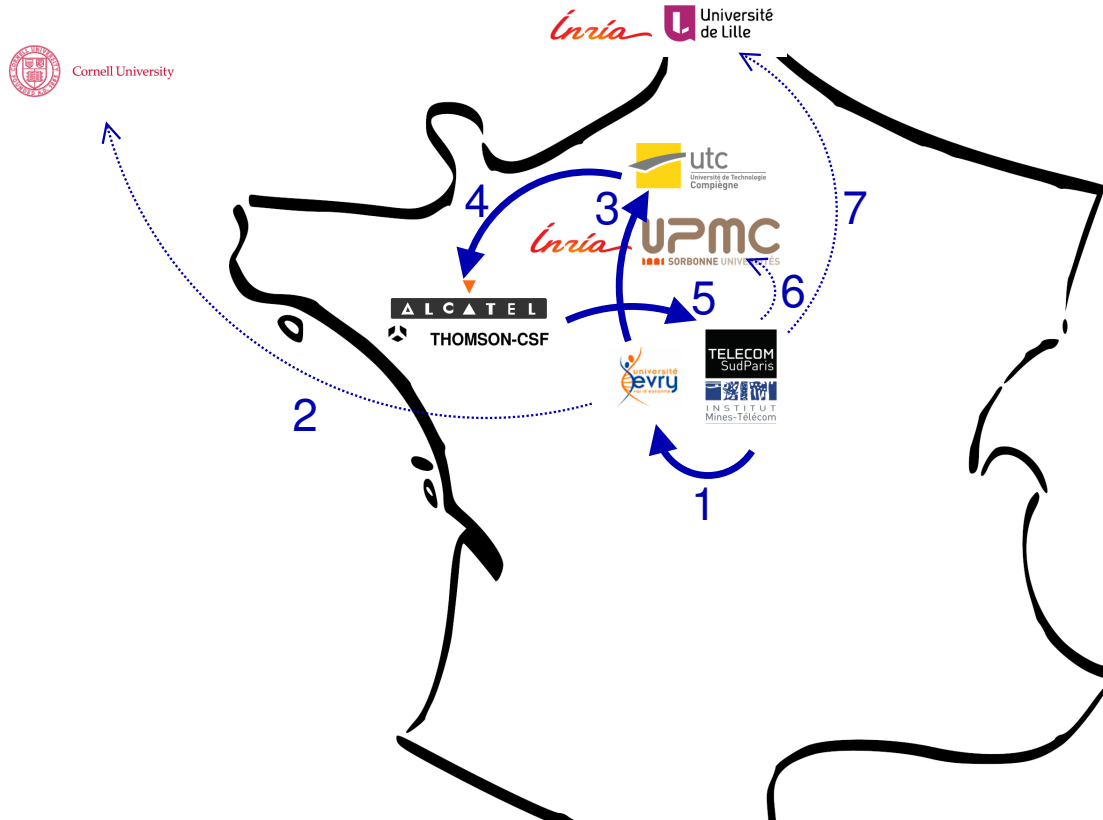
Jeudi 9 juillet 2015

Plan de la présentation

- 1 Introduction
 - Parcours
 - Contexte général
 - Objectif général
 - Contributions
- 2 Adaptation des applications aux déconnexions
- 3 Sensibilité au contexte des applications
- 4 Tolérance aux fautes des applications
- 5 Conclusion et perspectives



Parcours



Évolution des dispositifs « + petits et – chers »

Classification proposée dans [Bell, 2008]

Autres classes : prix constants, super-calculateurs

Années 1990

Réseaux locaux, ordinateurs portables, assistants personnels numériques.

Mobilité



Années 2000

Bluetooth et WiFi, téléphones mobiles.

Ubiquité



Années 2010

NFC, RFID, objets communicants, nuage.

Immersion



Évolution des usages et des services

Mobilité

Synchronisation de données,
édition collaborative.

Informatique mobile

[Satyanarayanan, 1996]



Ubiquité

E-commerce personnalisé,
tourisme pro-actif,
gestion de crise.

Informatique ubiquitaire

[Weiser, 1991, Satyanarayanan, 2001]



Immersion

E-santé,
ville intelligente,
contrôle énergétique,
jeux pervasifs.

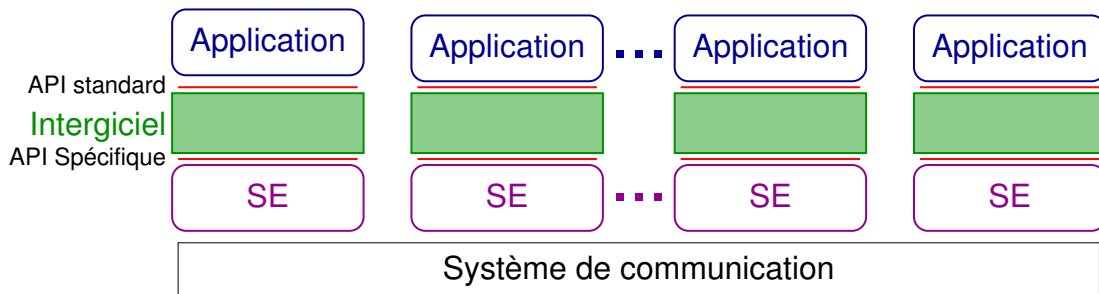
Internet des Objets

[Atzori et al., 2010]



Objectif général

Objectif de mes travaux = Maîtriser la complexité de la répartition



Systemes répartis avec mobilité, ubiquité et IoT

- **Mobiles et objets connectés** ⇒ **Hétérogénéité exacerbée**
- **Forte dynamique** ⇒ **Topologie variable et imprévisible**

Contribution sur l'adaptation aux déconnexions

- Motivations : informatique mobile
 - Plus particulièrement, ressources matérielles limitées, variabilité des communications
- Défis : gestion des déconnexions, continuité de service, adaptation
- Questions : qui (de l'application ou de l'intergiciel) fait quoi, quand, et comment ?

Contribution sur la sensibilité au contexte

- Motivations : informatique ubiquitaire puis IoT
 - Plus particulièrement,
 - Dispositifs enfouis, questionnant leur utilisabilité et la distraction de l'utilisateur
 - Grands systèmes
- Défis : autonomie, sensibilité au contexte, passage à l'échelle
- Questions : qu'est-ce que le contexte ? modélisé, géré comment ? quid du passage à l'échelle ?

Contribution sur la tolérance aux fautes

- Motivations : informatique ubiquitaire
 - Plus particulièrement, déconnexions + défaillances, partitionnements fréquents
- Défis : déconnexion Vs. défaillance, groupe partitionnable
- Questions : quelle différence entre déconnexion et défaillance ? quel modèle de système réparti dynamique ? quelles propriétés ?

Plan « adaptation aux déconnexions »

- 1 Introduction
- 2 **Adaptation des applications aux déconnexions**
 - Déconnexion et continuité de service
 - Bilan sur l'adaptation aux déconnexions
- 3 Sensibilité au contexte des applications
- 4 Tolérance aux fautes des applications
- 5 Conclusion et perspectives



Déconnexion et continuité de service

Limites des solutions système en 2000 :

- Coda [Kistler et al., 1991], Rover [Joseph et al., 1995], Bayou [Petersen et al., 1997]
- Manque séparation des préoccupations et généricité

Limites des solutions intergiciel objet en 2000 :

- Minimum CORBA et *Wireless* CORBA
- Uniquement déconnexions de courtes durées

Propositions :

- Déconnexions « longues » pour applications objet patrimoniales
- Concept de composants déconnectés

Bilan sur l'adaptation aux déconnexions

Leçons « du passage de l'objet au composant »

- Meilleure séparation des préoccupations (conteneurs ouverts), gestion des dépendances (offert + requis)
- Clarification des rôles dans la (stratégie) collaboration : architecte (politique), développeur (mécanisme), utilisateur (profil)

Dissémination

- ANTe [Kouici et al., 2006], DOA [Kouici et al., 2004], DOMINT

Encadrements

- DEA + Thèse Nabil Kouici [Kouici, 2005]

Projets : ITEA VIVIAN (2000-2002), ITEA OSMOSE (2003-2005), RNTL franco-finlandais AMPROS (2003–2005) (domaine gestion de crise)

Plan « sensibilité au contexte »

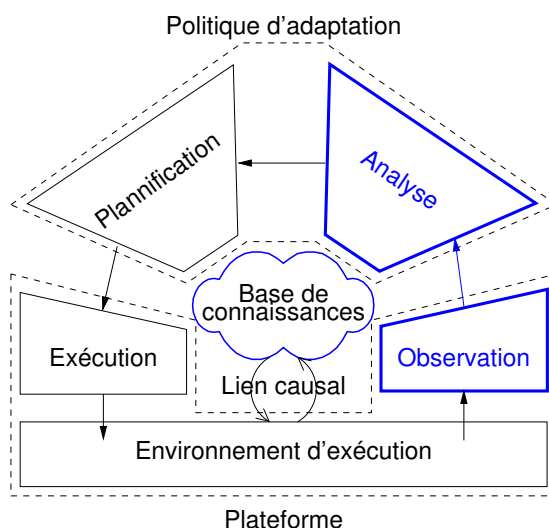
- 1 Introduction
- 2 Adaptation des applications aux déconnexions
- 3 **Sensibilité au contexte des applications**
 - Adaptation avec boucle d'autonomie
 - Vers la gestion de contexte autonome
 - Structuration de la gestion de contexte
 - Distribution des informations de contexte
 - Bilan sur la sensibilité au contexte
- 4 Tolérance aux fautes des applications
- 5 Conclusion et perspectives



Adaptation avec boucle d'autonomie

De l'adaptation à l'autonomie [Huebscher et al., 2008]

- Structuration : boucle d'autonomie K-MAPE [Kephart et al., 2003]



Sensibilité au contexte comme une fonctionnalité intergiciel

- Service de gestion de contexte [Coutaz et al., 2005]

Vers la gestion de contexte autonome

Limites des solutions de supervision en 2006

- Grappes/grilles de machines : Phoenix [Boutros Saab et al., 2000], LeWYS [Cecchet et al., 2005]
- Concept de contexte non formulé
 - Faible niveau d'abstraction + infos système

Limites des solutions contexte en 2006

- État de l'art : [Mostefaoui et al., 2004, Henricksen et al., 2005]
- Gestion *ad hoc* dans la partie applicative
 - Exception : Context Toolkit [Dey et al., 2001], orientation objet

Propositions :

- Structuration de la gestion de contexte pour l'autonomie
- Distribution des informations de contexte

Structuration de la gestion de contexte

Orientation donnée

- « Tableau noir » et « Espace de n-uplets »
- CASS [Fahy et al., 2004], Gaia [Román et al., 2002], CoBrA [Chen, 2003], CMF [Korpipää et al., 2003], SOCAM [Gu et al., 2004], et PACE [Henricksen et al., 2005]
- Gestion centralisée et abstraction faible des traitements

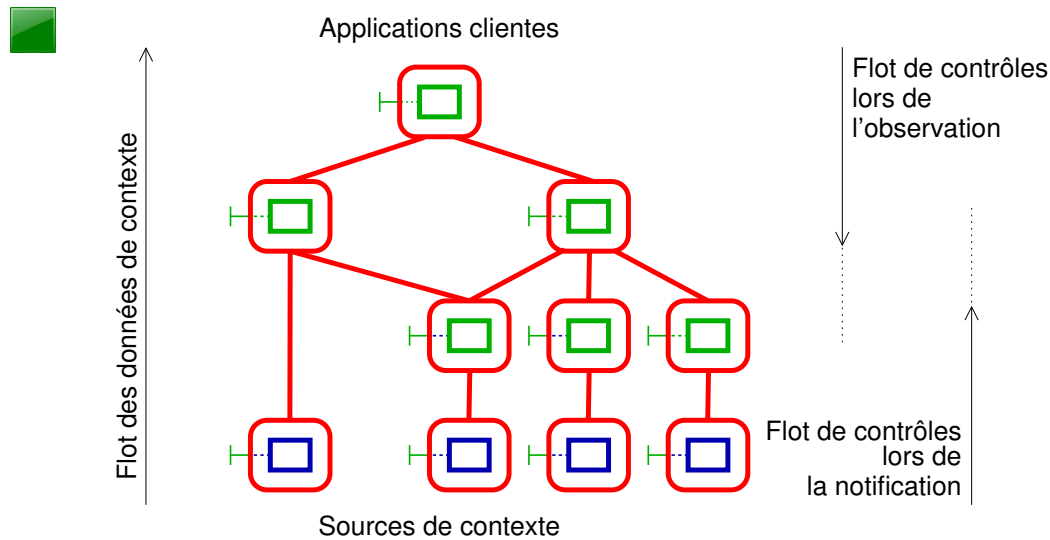
Orientation processus avec des objets

- Encapsulation des traitements pour leur réutilisation
- Context Toolkit [Dey et al., 2001], RCSM [Yau et al., 2002], SAJE [Courtrai et al., 2003], Hydrogen [Hofer et al., 2003], MoCA [da Rocha et al., 2006], et WildCAT [David et al., 2005]
- Complexité graphe de dépendances + pas de composite

Concept de nœud de contexte

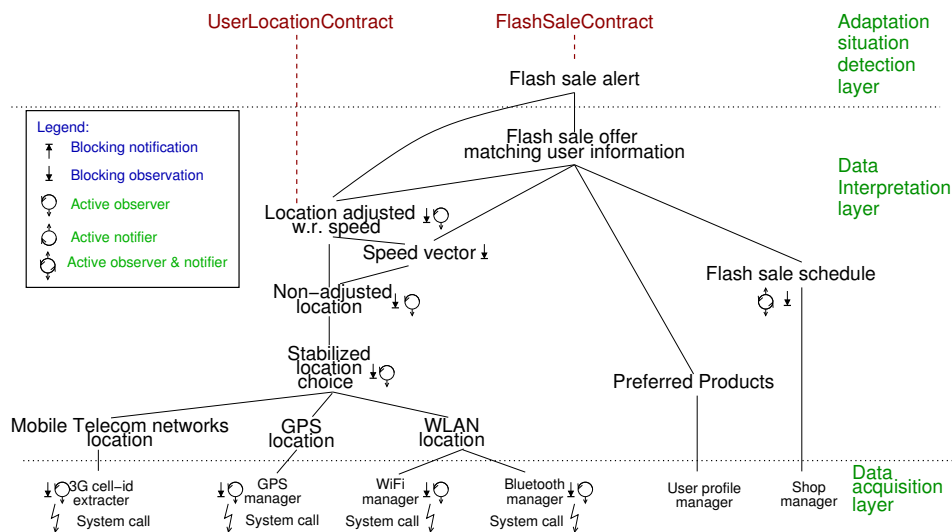
Orientation processus avec des composants

- Construction à partir d'éléments composés plutôt que programmés
 - Modèle de composant FRACTAL avec hiérarchie et partage
- Architecture de composants pour la gestion de contexte



Canevas logiciel et langage dédié

Canevas logiciel COSMOS et langage COSMOS DSL



Expérience : Samsung Galaxy S2 phone

- Création 3ms, traversée 0.4ms, nb max 5500, et hauteur max 45

Distribution des informations de contexte

Limites des solutions en 2011

- Uniquement espace ambiant entourant un/des utilisateur(s)
IoT \approx des espaces ambiants [Bellavista et al., 2012, Perera et al., 2014]
 - De nouveaux défis : passage à l'échelle, vie privée, etc.

Problème du passage à l'échelle

- Évnt discret \implies syst. rép. à base d'évts (DEBS) [Eugster et al., 2003]
- Modèle de données ouvert + besoin expressivité du filtrage
 \implies filtrage basé contenu
- Données semi-structurées \implies réseau de courtiers [Mühl et al., 2006]
 - Plutôt que réseau pair-à-pair [Kermarrec et al., 2013]

Systèmes multi-échelles

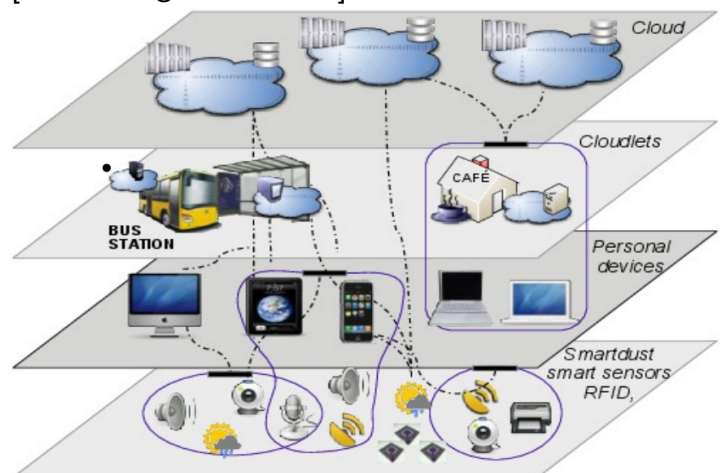
Multi-échelle plutôt que passage à l'échelle en « force brute »

- De « espace ambiant entourant un/des utilisateur/s »
à « IoT \approx des env. ambiants » [Bellavista et al., 2012, Perera et al., 2014]
 - De nouveaux défis : **passage à l'échelle**, **hétérogénéité**, etc.

Caractérisation multi-échelle

- Point de vue / Dimension
 - Équipements / Capacité stockage
 - Équipements / Puissance calcul
 - Données / Volume
 - Réseaux / Latence
 - Réseaux / Bande passante
 - Réseaux / Portée de transmission
 - Géographie / Distance
 - Géographie / Localisation admin.

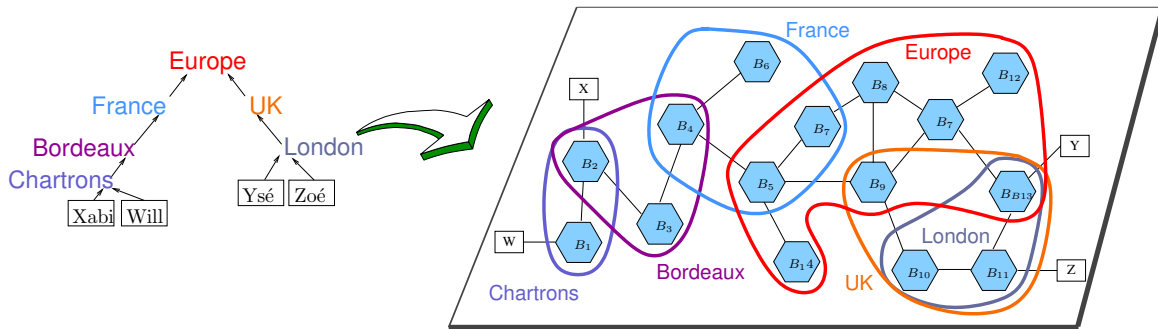
[Rottenberg et al., 2014]



Concept de portée de distribution

Portée de distribution (en anglais, *scope*)

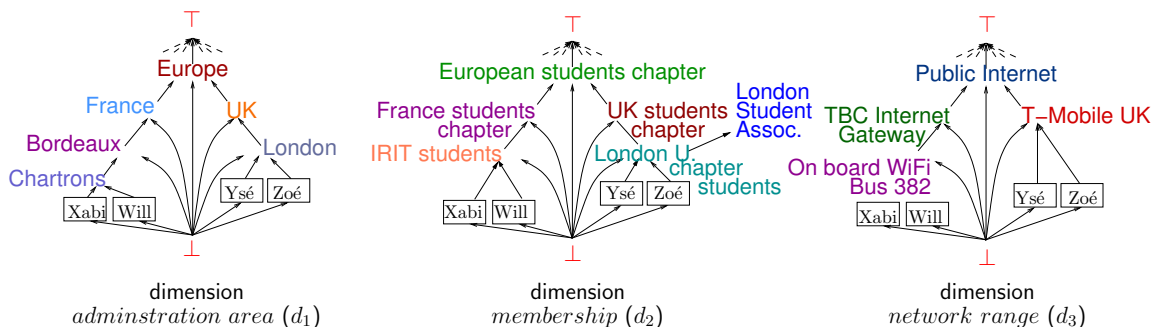
- Confine la dissémination dans un ensemble de clients (producteurs et consommateurs)
- Est récursivement membre d'autres portées



Distribution multi-échelle

Multiples portées de distribution

- Distribution d'une notification contrainte selon plusieurs critères
 - *i.e.* dans plusieurs zones logiques correspondant chacune à une **échelle** dans une **dimension** d'un **point de vue** de l'architecture
 - Pseudo-portées \perp et \top





Projection des graphes de portées

+

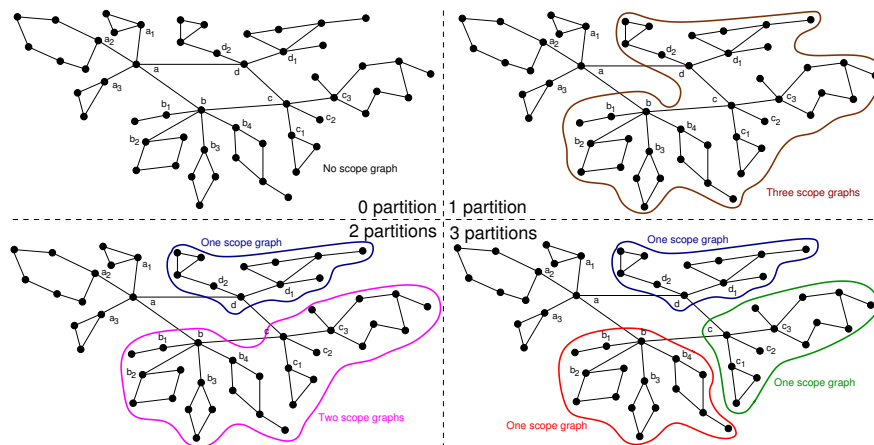
 MUDEBS

Expérience : % du nombre de messages subscribe

■ 0 : 100%, 1 : 55%, 2 : 33%, 3 : 19%

Expérience : % du nombre de messages notification

■ 1 partition, filtres *false* : aucun 100%, L_3 : 84%, L_2 : 58%, L_1 : 41%



Bilan sur la sensibilité au contexte

📖 Dissémination

- Structuration : SCICO (2013) [Chabridon et al., 2013], DAIS'2011 [Chabridon et al., 2011] ICDIM'2010 [Bouzeghoub et al., 2010], DOA'2009 [Taconet et al., 2009], DAIS'2007 [Conan et al., 2007], COSMOS
- Distribution : M4IoT'2014 [Marie et al., 2014], MW4NG'2014 [Lim et al., 2014a], MUDEBS, MUCONTEXT

🎓 Encadrements

- Cong Kinh Nguyen (Ingénieur 2008–10), Léon Lim (PostDoc 2013–15)

Projets :

- ARC INRIA BROCCOLI (2008–2009), IT CASAC (2008), PICF TOTEM (2009–2012) FUI CAPPUCINO (2007–2010), ANR INCOME (2012–2015)

Bilan sur la sensibilité au contexte (suite)

Leçons

- Structuration de la gestion de contexte
 - Ingénierie dirigée par les modèles
 - Architecture à base de composants & Méta-modèle & Langage dédié & Génération
- Distribution des informations de contexte
 - Asynchronisme : Système réparti à base d'événements de base & *Mode « global » pour le mode observation* & *Qualité de contexte* & *Respect de la vie privée*
 - Contrôle du passage à l'échelle : Distribution multi-échelle
 - Découpage en zone selon dimensions de points de vue

2003-

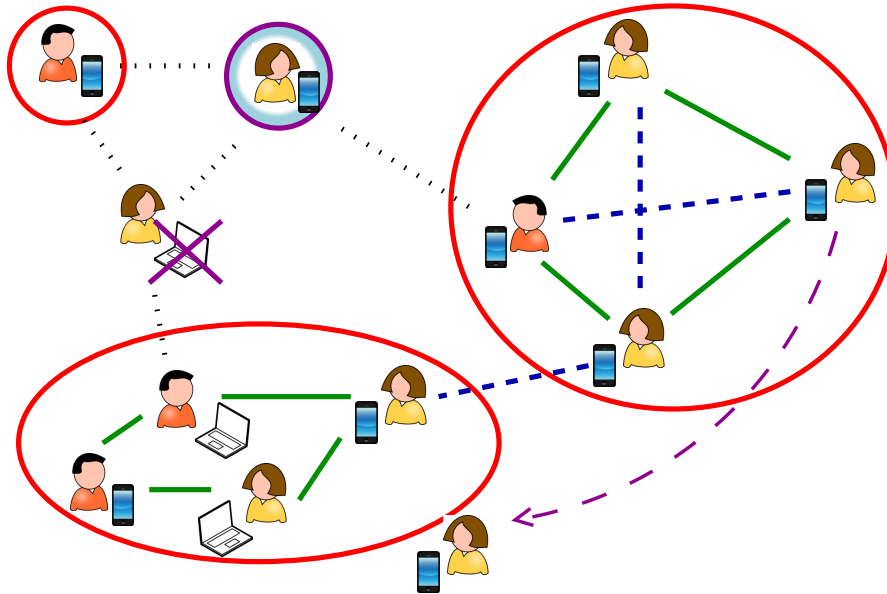
Plan « Tolérance aux fautes »

- 1 Introduction
- 2 Adaptation des applications aux déconnexions
- 3 Sensibilité au contexte des applications
- 4 **Tolérance aux fautes des applications**
 - Mobilité et tolérance aux fautes
 - Partitionnements avec réconciliation
 - Gestion de groupe partitionnable
 - Bilan sur la tolérance aux fautes
- 5 Conclusion et perspectives



Mobilité et tolérance aux fautes

Mobilité imprévisible et fréquente des utilisateurs/appareils
 Mobilité + déconnexions + défaillances \Rightarrow Partitionnements



Partitionnements avec réconciliation

Partitionnements transitoires

- Déconnexions / partitionnements transitoires
- Mais, durent suffisamment longtemps pour gêner les utilisateurs
- Réplication optimiste avec réconciliation [Saito et al., 2005]

Limites des solutions de la littérature en 2003

- Pas de distinction entre arrêts francs et déconnexions
- [Chandra et al., 1996], [Aguilera et al., 1999], [Chockler et al., 2001]

Propositions :

- Détecteur de connectivité pour anticiper les déconnexions
- Détecteur de déconnexion pour informer les utilisateurs de la cause

Nouveaux détecteurs

»» Approche

- Stabiliser la connectivité et annoncer les déconnexions
- Distinguer les causes des partitionnements

■ Détecteur de défaillance

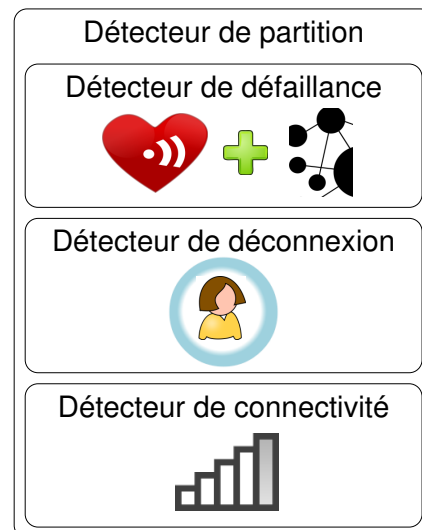
- Avec infos sur la topologie

■ Détecteur de déconnexion

- Complétude et précision, à terme parfaites car \nexists fausse détection

■ Détecteur de connectivité

- Hystérésis pour prévention de l'effet ping-pong



Gestion de groupe partitionnable

Partitionnements fréquents et « permanents »

- Réseaux mobiles spontanés
- Arrivée et départ continus de nœuds sans connaissance de Π

Limites des solutions de la littérature [Pleish et al., 2008]

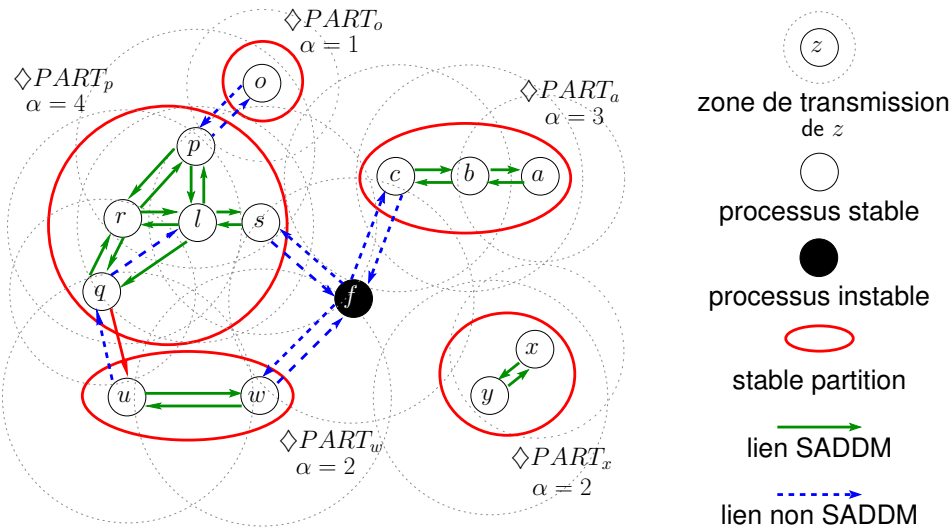
- [Chockler et al., 2001] : Vivacité non assurée si connexions par intermittence pour toujours
- [Babaoglu et al., 2001] : Autorise mise en œuvre triviale mais inutile
- Dans les deux cas, Π connu et fixe

✓ Proposition

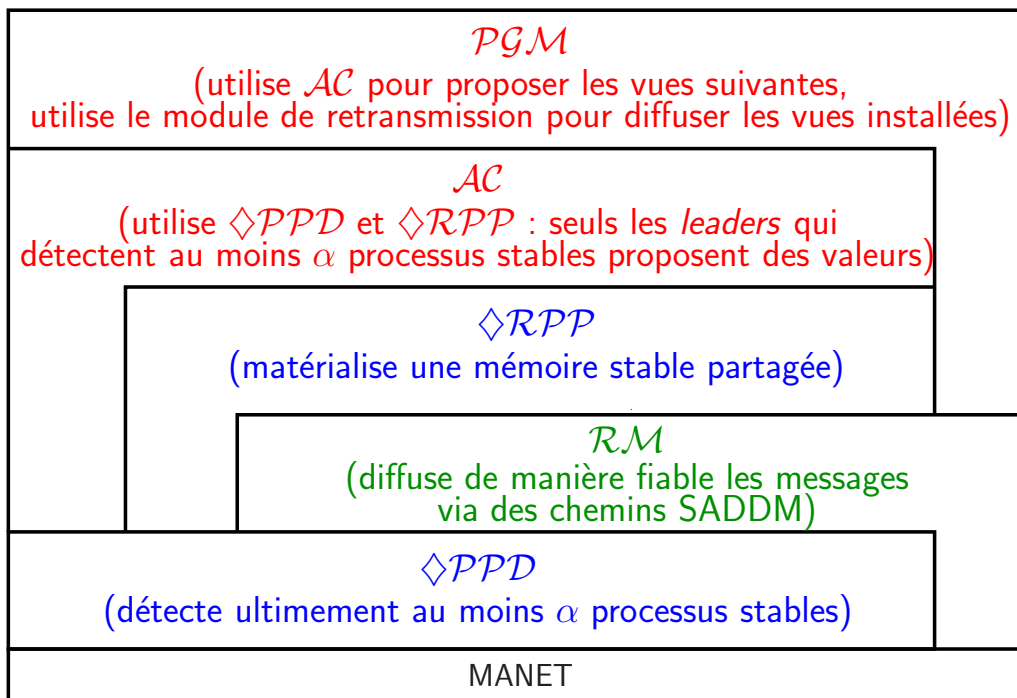
- Modèle pour MANETs et solution qui est
 - Assez faible pour être résoluble
 - Assez forte pour éviter une mise en œuvre triviale et inutile

Modèle de système réparti

- Arrivée infinie avec accès simultané borné [Merritt et al., 2000]
- Liens SADDM + condition et critère de stabilité



Architecture de la solution



Bilan sur la tolérance aux fautes



Dissémination

- NCA'2010 [Arantes et al., 2010], NCA'2008 [Conan et al., 2008]
- JPDC (2014) [Lim et al., 2014b], EDCC'2012 [Lim et al., 2012]



Encadrements

- Plusieurs stages M2 + Thèse de Léon Lim [Lim, 2012]

Première solution à la gestion de groupe partitionnable

- Assez forte pour faciliter la conception des applications réparties
- Assez faible pour être résoluble

Leçons

- Approche par collaboration entre l'application et l'intergiciel
- Questions ouvertes : utilisabilité et cas d'usage

Plan « conclusion et perspectives »

- 1 Introduction
- 2 Adaptation des applications aux déconnexions
- 3 Sensibilité au contexte des applications
- 4 Tolérance aux fautes des applications
- 5 Conclusion et perspectives
 - Bilan général
 - Leçons apprises
 - Démarche suivie
 - Résumé des activités de recherche
 - Perspectives



Bilan général

Adaptation aux déconnexions

- Contraintes ressources + variabilité communication
- Existant : sans intergiciel avec transparence, fichier ou BD ou objet
- Contributions :
 - Structuration des applications sur les appareils mobiles
 - Objets déconnectés pour applis patrimoniales CORBA
 - Conteneur et gestion de cache de composants
 - Modélisation des points de variation
 - Politiques de déploiement et de remplacement

Bilan général (suite)

Sensibilité au contexte

- Environnements ouverts, utilisateurs mobiles, objets volatiles
- Existant : contexte système ou pas de structuration intergicelle
- Contributions :
 - Structuration de la gestion de contexte
 - Architecture composants nœuds de contexte & méta-modèle, langage dédié et génération
 - Distribution des informations de contexte
 - Distribution multi-échelle pour le passage à l'échelle

Bilan général (suite)

Tolérance aux fautes

- Systèmes fortement dynamiques et dispositifs moins fiables
- Existant :
 - Partitionnements de longue durées : pas de distinction entre défaillance et déconnexion
 - Formation spontanée de groupes : pas de gestion de groupe partitionnable résoluble et sans mise en œuvre triviale
- Contributions :
 - Détecteurs déconnexion avec détecteurs défaillance et partition
 - Détecteur participant et 1^è gestion de groupe partitionnable
 - Modèle : liens SADDM & condition et critère de stabilité
 - Solution « à la » Paxos avec « synode abandonnable »

Leçons apprises

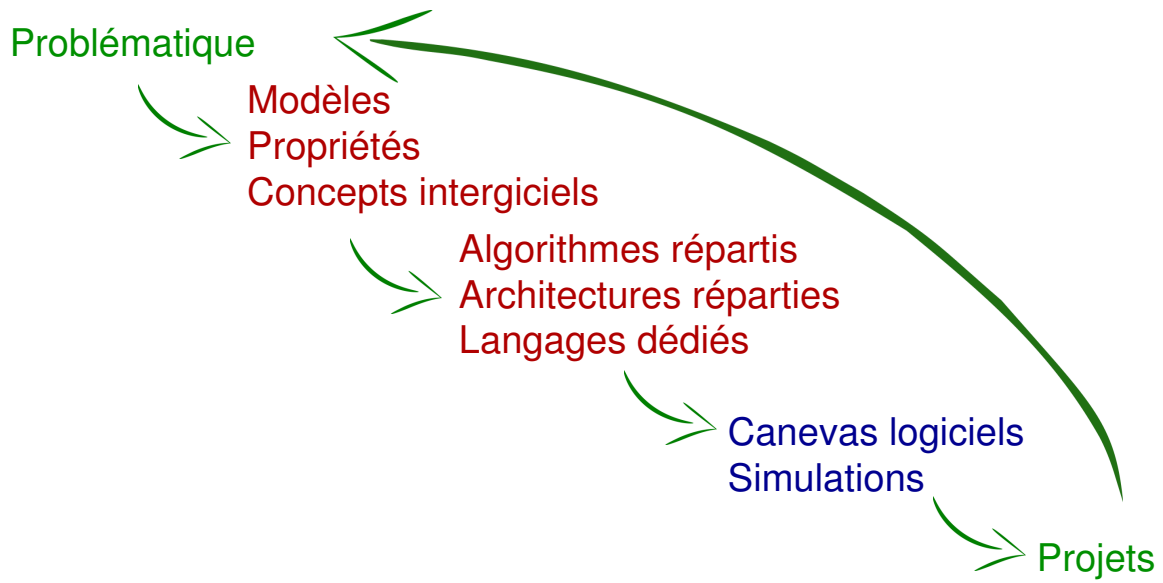
En termes de modèles

- Ubiquité \implies Forte dynamicité \implies Risques d'instabilité
 - Connectivité (effet ping-pong, locale)
 - Condition (qui) et critères (à combien) de stabilité
- Possible de former des « petits » groupes (partitionnables)

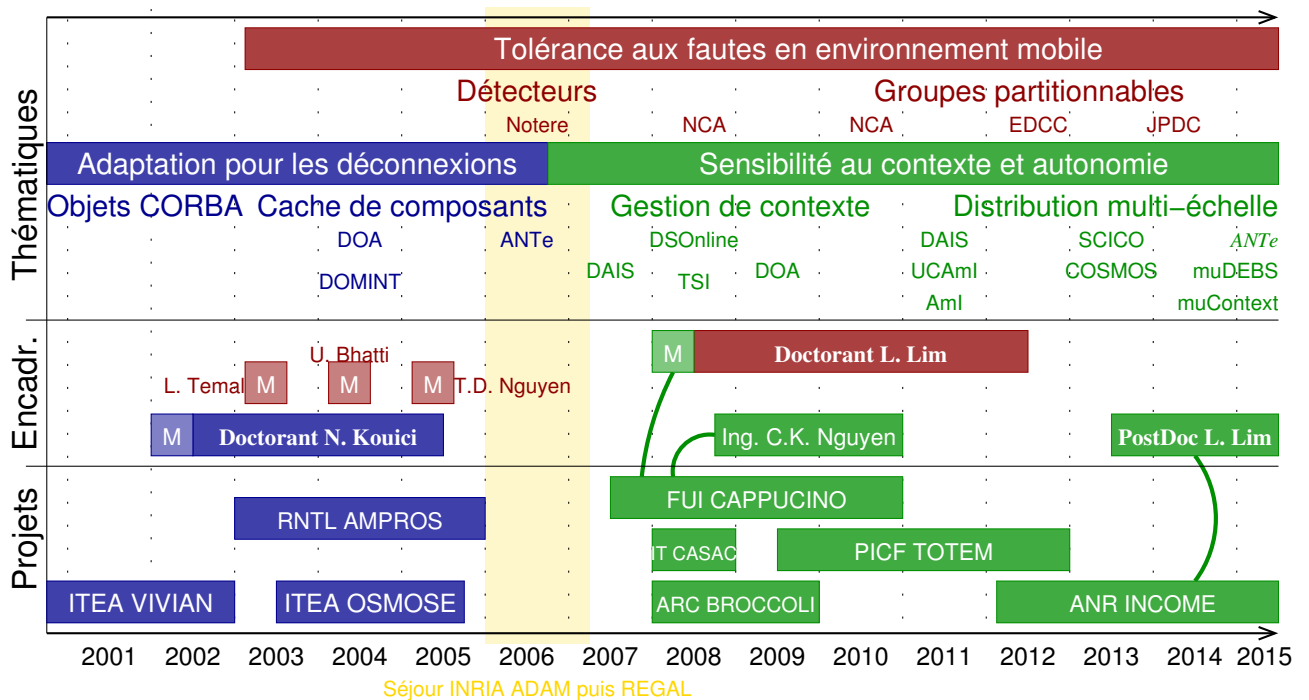
En termes d'architectures

- Intergiciel pour la mise en œuvre de la stratégie collaboration
- Composant : IDM avec traçabilité de la conception à l'exécution
- Passage à l'échelle et hétérogénéité \leftrightarrow multi-échelle
 - DEBS pour IoT : Portée de distribution (point de vue / dimension)

Démarche suivie



Résumé des activités de recherche



Dissémination des informations en provenance de l'IoT

- Objectif : espace ambient délimité et accès contrôlé aux autres
- Oscillation cyclique entre centralisation et répartition
[Want et al., 2015]
- Points de départ : caractérisation multi-échelle du système réparti
 - Au delà du *mobile cloud computing*
[Satyanarayanan, 2010, Fernando et al., 2013]
 - Au delà de ce que [Satyanarayanan, 2001] nomme « passage à l'échelle localisé » (*localised scalability*)
- Défis : nous ajoutons la mobilité / dynamicité
 - Changement d'espace ambient des clients
 - Mobilité des courtiers (embarqués dans des bus, etc.)

Tolérance aux fautes dans l'IoT

- Objectif : système partitionnable et ensemble de participants non connu
- CAP [Fox et al., 1999, Gilbert et al., 2002] pour le large échelle :
 - Impossibilité d'avoir conjointement cohérence (forte), (haute) disponibilité, tolérance partitionnements
 - IoT : comm. sans fil, applis différentes des applis large échelle
⇒ Ré-examen [Gilbert et al., 2012]
- Défis : pertinence, utilisabilité du concept « groupe partitionnable »
 - Pas de copie de référence
 - Collaboration et indication aux utilisateurs de l'état d'avancement des activités [Brewer, 2012]
- Points de départ : communications de groupe, machine à états, services de coordination
 - Critère (durée, distance, énergie, fonctionnel, etc.)
& Condition (dynamique) de stabilité
 - Mobilité individuelle et de groupe

Ingénierie logicielle des algorithmes répartis

- Complexe = fautes, taille, composition, activités [Collet et al., 2015]
- Coût du développement logiciel d'une mise en œuvre algorithmique
- Défis : de l'abstraction à la vérification ou l'exécution
 - Pseudo-code \implies langage dédié [Mernik et al., 2005]
 - Langage interne à un langage généraliste (concision)
 - Permettant la structuration du système complet
 - Langage de définition d'architecture
 - Langage de prog. objet, composant
 - *Mixins* ou prog. par aspect pour instrumentation
 - Et outillé pour appliquer les principes des méthodes agiles
 - Émulation (par « scriptage »), simulation, vérification de modèle, etc.
- Points de départ :
 - Spin [Holzmann, 1997], IOA [Garland et al., 1997], PlusCal [Lamport, 2009], Splay [Leonini et al., 2009], Overlog [Alvaro et al., 2010], Distal [Biely et al., 2013]

Références

- Aguilera, M., Chen, W., et al. (1999).
Using the Heartbeat Failure Detector for Quiescent Reliable Communication and Consensus in Partitionable Networks.
Theoretical Computer Science, 220(1) :3–30.
- Alvaro, P., Condie, T., Conway, N., Hellerstein, J., et al. (2010).
I Do Declare : Consensus in a Logic Language.
ACM Operating Systems Review, 43(4) :25–30.
- Arantes, L., Sens, P., Thomas, G., Conan, D., et al. (2010).
Partition Participant Detector with Dynamic Paths in Mobile Networks.
In *Proc. 10th IEEE International Symposium on Network Computing and Applications*, Cambridge, MA, USA.
- Atzori, L., Iera, A., et al. (2010).
The Internet of Things : A survey.
Computer Networks, 54(15) :2787–2805.
- Babaoğlu, O., Davoli, R., et al. (2001).
Group Communication in Partitionable Systems : Specification and Algorithms.
IEEE Transactions on Software Engineering, 27(4) :308–336.
- Bell, G. (2008).
Bell's law for the birth and death of computer classes.
Communications of the ACM, 51(01) :86–94.
- Bellavista, P., Corradi, A., Fanelli, M., et al. (2012).
A Survey of Context Data Distribution for Mobile Ubiquitous Systems.
ACM Computing Surveys, 44(4) :24:1–24:45.

Références (suite)

- Biely, M., Delgado, P., Milosevic, Z., et al. (2013).
Distal : A Framework for Implementing Fault-tolerant Distributed Algorithms.
In *Proc. 43rd IEEE International Conference on Dependable Systems and Networks*, Budapest, Hungary.
- Boutros Saab, C., Bonnaire, X., et al. (2000).
A flexible monitoring platform to build cluster management services.
In *Proc. IEEE International Conference on Cluster Computing*, pages 258–265, Chemnitz, Germany.
- Bouzeghoub, A., Taconet, C., Jarraya, A., Do, N., et al. (2010).
Complementarity of Process-oriented and Ontology-based Context Managers to Identify Situations.
In *Proc. 5th International Conference on Digital Information Management*, pages 222–229, Thunder Bay, Canada.
- Brewer, E. (2012).
Pushing the CAP : Strategies for Consistency and Availability.
IEEE Computer, 45(2).
- Caavalcante, E., Alves, M., Batista, T., Delicato, F., et al. (2015).
An Analysis of Reference Architectures for the Internet of Things.
In *Proc. of the 1st ACM International Workshop on Exploring Component-based Techniques for Constructing Reference Architectures*, pages 13–16, Montréal, Canada.
- Cecchet, E., Elmeleegy, H., Layaïda, O., et al. (2005).
Implementing Probes for J2EE Cluster Monitoring.
Studia Informatica.
- Chabridon, S., Conan, D., Abid, Z., et al. (2013).
Building Ubiquitous QoC-Aware Applications through Model-Driven Software Engineering.
Elsevier Journal of Science of Computer Programming, 78(10).

Références (suite)

- Chabridon, S., Ngo, C.-C., Abid, Z., Conan, D., Taconet, C., et al. (2011).
Towards QoC-Aware Location-based Services.
In *Proc. 11th IFIP WG 6.1 International Conference on Distributed Applications and Interoperable Systems*, Lecture Notes in Computer Science, Reykjavik, Iceland. Springer-Verlag.
- Chandra, T. et al. (1996).
Unreliable Failure Detectors for Reliable Distributed Systems.
Journal of the ACM, 43(2) :225–267.
- Chen, H. (2003).
An Intelligent Broker Architecture for Pervasive Context-Aware Systems.
PhD thesis, University of Maryland, Baltimore, USA.
- Chockler, G., Keidar, I., et al. (2001).
Group Communication Specifications : A Comprehensive Study.
ACM Computing Surveys, 33(4) :427–469.
- Collet, P., Du Bousquet, L., Duchien, L., et al. (2015).
Gdr génie de la programmation et du logiciel — défis 2025.
<http://gdr-gpl.cnrs.fr/node/160>.
- Conan, D., Rouvoy, R., et al. (2007).
Scalable Processing of Context Information with COSMOS.
In Indulska, J. et al., editors, *Proc. 6th IFIP WG 6.1 International Conference on Distributed Applications and Interoperable Systems*, volume 4531 of *Lecture Notes in Computer Science*, pages 210–224, Paphos, Cyprus. Springer-Verlag.

Références (suite)

- Conan, D., Sens, P., Arantes, L., et al. (2008).
Failure, Disconnection and Partition Detection in Mobile Environment.
In *Proc. 7th IEEE International Symposium on Network Computing and Applications*, Cambridge, MA, USA.
- Courtrai, L., Guidec, F., Le Sommer, N., et al. (2003).
Resource Management for Parallel Adaptive Components.
In *Proc. IEEE IPDPS Workshop on Java for Parallel and Distributed Computing*, pages 134–141, Nice, France.
- Coutaz, J., Crowley, J., Dobson, S., et al. (2005).
Context is Key.
Communications of the ACM, 48(3) :49–53.
- da Rocha, R. et al. (2006).
Context Management in Heterogeneous, Evolving Ubiquitous Environments.
IEEE Distributed Systems Online, 7(4).
- David, P.-C. et al. (2005).
WildCAT : a generic framework for context-aware applications.
In *Proc. 3rd International Workshop on Middleware for Pervasive and Ad-hoc Computing*, pages 1–7, Grenoble (France).
- Dey, A., Salber, D., et al. (2001).
A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications.
Special issue on context-aware computing in the Human-Computer Interaction Journal, 16(2–4) :97–166.
- Eugster, P., Felber, P., Guerraoui, R., et al. (2003).
The Many Faces of Publish/Subscribe.
ACM Computing Surveys, 35(2).

Références (suite)

- Fahy, P. et al. (2004).
CASS : Middleware for Mobile Context-Aware Applications.
In *Proc. ACM MobiSys Workshop on Context Awareness*, Boston, USA.
- Fernando, N., S.W., L., et al. (2013).
Mobile cloud computing : A survey.
Future Generation Computer Systems.
- Fox, A. et al. (1999).
Harvest, Yield and Scalable Tolerant Systems.
In *Proc. 7th Workshop Hot Topics in Operating Systems*, pages 174–178, Rio Rico, Arizona, USA.
- Garland, S., Lynch, N., et al. (1997).
IOA : A Language for Specifying, Programming, and Validating Distributed Systems.
Technical report, MIT Laboratory for Computer Science.
- Gilbert, S. et al. (2012).
Perspectives on the CAP Theorem.
IEEE Computer, 45(2).
- Gilbert, S. et al. (2002).
Brewer's conjecture and the feasibility of consistent, available, partition-tolerant web services.
ACM SIGACT News (Technical Column), 33(2) :51–59.
- Gu, T., Pung, H., et al. (2004).
A middleware for building context-aware mobile services.
In *Proc. 59th IEEE Vehicular Technology Conference*, volume 5, pages 2656–2660.

Références (suite)

Henricksen, K., Indulska, J., McFadden, T., et al. (2005).
Middleware for Distributed Context-Aware Systems.
In *Proc. 7th International Symposium on Distributed Objects and Applications*, Lecture Notes in Computer Science, Agia Napa (Cyprus). Springer-Verlag.

Hofer, T., Schwinger, W., Pichler, M., Leonhartsberger, G., Altmann, J., et al. (2003).
Context-Awareness on Mobile Devices — the Hydrogen Approach.
In *Proc. 36th Annual Hawaii International Conference on System Sciences*, Hawaii (USA).

Holzmann, G. (1997).
The Model Checker Spin.
IEEE Transactions on Software Engineering, 23(5) :1–17.

Huebscher, M. et al. (2008).
A survey of Autonomic Computing—Degrees, Models and Applications.
ACM Computing Surveys, 40(3) :7:1–7:28.

Joseph, A. D., deLespinasse, A. F., Tauber, J. A., Gifford, D., et al. (1995).
Rover : A Toolkit for Mobile Information Access.
In *Proc. 15th ACM Symposium on Operating Systems Principles*, pages 156–171.

Kephart, J. et al. (2003).
The Vision of Autonomic Computing.
IEEE Computer, 36(1) :41–50.

Kermarrec, A.-M. et al. (2013).
XL Peer-to-Peer Pub/Sub Systems.
ACM Computing Surveys, 46(2) :16:1–16:45.

Références (suite)

Kistler, J. et al. (1991).
Disconnected Operation in the Coda File System.
In *Proc. 13th ACM Symposium on Operating Systems Principles*, pages 213–225, Pacific Grove, USA.

Korpipää, P., Mäntyjärvi, J., Kela, J., Keränen, H., et al. (2003).
Managing Context Information in Mobile Devices.
IEEE Pervasive Computing, 2(3) :42–51.

Kouici, N. (2005).
Gestion des déconnexions pour applications réparties à base de composants en environnements mobiles.
PhD thesis, Institut National des Télécommunications, en co-accréditation avec l'Université d'Evry Val d'Essonne, Évry (France).

Kouici, N., Conan, D., et al. (2004).
Caching Components for Disconnection Management in Mobile Environments.
In Z. Tari et al, editor, *Proc. 6th International Symposium on Distributed Objects and Applications*, volume 3291 of *Lecture Notes in Computer Science*, pages 1322–1339, Agia Napa, Cyprus. Springer-Verlag.

Kouici, N., Conan, D., et al. (2006).
État de l'art de la gestion de cache logiciel pour tolérer les déconnexions en environnements mobiles.
Annales des Télécommunications, 61(11–12) :1458–1483.

Lamport, L. (2009).
The PlusCal Algorithm Language.
In *Proc. 6th International Colloquium on Theoretical Aspects of Computing*, volume 5684 of *Lecture Notes in Computer Science*, pages 36–60, Kuala Lumpur, Malaysia.

Références (suite)

Leonini, L., Rivière, E., et al. (2009).

SPLAY : Distributed systems evaluation made simple (or how to turn ideas into live systems in a breeze).
In *Proc. of the 6th USENIX Symposium on Networked Systems Design and Implementation*, pages 185–198, Boston, MA, USA.

Lim, L. (2012).

Gestion de groupe partitionnable dans les réseaux mobiles spontanés.

PhD thesis, Institut Mines Télécom, Télécom SudParis, en co-accréditation avec l'Université d'Évry Val d'Essonne, Évry (France).

Lim, L. et al. (2012).

An Eventual α Partition-Participant Detector for MANETs.

In *Proc. 9th European Dependable Computing Conference*, Sibiu, Romania.

Lim, L. et al. (2014a).

Distributed Event-Based System with Multiscoping for Multiscalability.

In *Proc. 9th Middleware Workshop on Middleware for Next Generation Internet Computing*, Bordeaux, France.

Lim, L. et al. (2014b).

Partitionable group membership for Mobile Ad hoc Networks.

Journal of Paralled and Distributed Computing, pages 2708–2721.

Marie, P., Lim, L., Manzoor, A., Chabridon, S., Conan, D., et al. (2014).

QoC-Aware Context Data Distribution in the Internet of Things.

In *Proc. Middleware Workshop on Middleware for Context-Aware Applications in the IoT*, Bordeaux, France.

Références (suite)

Mernik, M., Heering, J., et al. (2005).

When and How to Develop Domain-Specific Languages.

ACM Computing Surveys, 37(4) :316–344.

Merritt, M. et al. (2000).

Computing with Infinitely Many Processes Under Assumptions on Concurrency and Participation.

In *Proceedings of the 14th International Symposium on Distributed Computing*, volume 1914 of *Lecture Notes in Computer Science*, pages 164–178, Toledo, Spain. Springer-Verlag.

Mostefaoui, G., Pasquier-Rocha, J., et al. (2004).

Context-aware computing : A guide for the pervasive computing community.

In *IEEE/ACS International Conference on Pervasive Services*, pages 39–48, Novi Sad (Serbia/Montenegro).

Mühl, G., Fiege, L., et al. (2006).

Distributed Event-Based Systems.

Springer.

Perera, C., Zaslavsky, A., Christen, P., et al. (2014).

Context Aware Computing for The Internet of Things : A Survey.

IEEE Communications Surveys and Tutorials, 16(1) :414–454.

Petersen, K., Spreitzer, M., Terry, D., Theimer, M., et al. (1997).

Flexible Update Propagation for Weakly Consistent Replication.

In *Proc. 16th ACM Symposium on Operating Systems Principles*, pages 288–301, Saint Malo, France.

Pleish, S., Rützi, O., et al. (2008).

On the Specification of Partitionable Group Membership.

In *Proc. 7th European Dependable Computing Conference*, Kaunas, Lithuania.

Références (suite)

Román, M., Hess, C., Cerqueira, R., Ranganathan, A., Campbell, R., et al. (2002).

Gaia : A Middleware Infrastructure to Enable Active Spaces.

IEEE Pervasive Computing, 1(4) :74–83.

Rottenberg, R., Leriche, S., Taconet, C., Lecocq, C., et al. (2014).

MuSCa : A Multiscale Characterization Framework for Complex Distributed Systems.

In *Proc. 3rd Workshop on Model Driven Approaches in System Development*, Warsaw, Poland.

Saito, Y. et al. (2005).

Optimistic Replication.

ACM Computing Surveys, 37(1) :42–81.

Satyanarayanan, M. (1996).

Fundamental Challenges in Mobile Computing.

In *Proc. 15th ACM Symposium on Principles of Distributing Computing*, pages 1–7, Philadelphia, USA.

Satyanarayanan, M. (2001).

Pervasive Computing : Vision and Challenges.

IEEE Personal Communications, 8(4) :10–17.

Satyanarayanan, M. (2010).

Mobile Computing : the Next Decade.

In *Proc. 1st ACM Workshop on Mobile Cloud Computing & Services : Social Networks and Beyond*, pages 1–6, San Francisco.

Références (suite)

Taconet, C., Kazi-Aoul, Z., Zaier, M., et al. (2009).

CA3M : A runtime model and a middleware for dynamic context management.

In *Proc. 11th International Symposium on Distributed Objects and Applications*, volume 5870 of *Lecture Notes in Computer Science*, pages 513–530, Algarve, Portugal. Springer-Verlag.

Want, R., Schilit, B., et al. (2015).

Enabling the Internet of Things.

IEEE Computer, 48(1).

Weiser, M. (1991).

The Computer for the 21st Century.

Scientific American, pages 94–100.

Yau, S., Karim, F., Wang, Y., Wang, B., et al. (2002).

Reconfigurable Context-Sensitive Middleware for Pervasive Computing.

IEEE Pervasive Computing, 1(3) :33–40.