

# Adaptation et Négociation de Contents

Tayeb LEMLOUMA

10 décembre 2004

Équipe WAM, INRIA Rhône-Alpes



# Plan

- 1 Introduction
- 2 Environnements hétérogènes
- 3 Adaptation
- 4 Négociation
- 5 Standardisation : Device Independence
- 6 Conclusions



# Introduction

- Les réseaux et les systèmes d'accès à l'information deviennent de plus en plus hétérogènes.
- De nombreux appareils hétérogènes existent aujourd'hui.
- De nombreuses applications complexes existent du côté serveur.
- Émergence de l'utilisation des appareils mobiles pour l'accès au contenu.
- Chiffres : 1.5 milliard de mobiles; Février 2004 : le milliardième abonné mobile.
- **Problème** : comment assurer l'accès au contenu à travers des appareils hétérogènes ?



# Environnements hétérogènes

- Grande diversité d'appareils qui accèdent au Web : assistants personnels, ordinateurs portables, téléphones mobiles, téléviseurs, etc.
- Les appareils sont limités : formats supportés, réseaux et protocoles d'accès, taille d'affichage, énergie..
- Le contenu du Web est de plus en plus complexe : plusieurs formats (HTML, XHTML, SMIL, SVG,...), différentes fonctionnalités et ressources médias (animations, vidéo, audio ..), scénarios de présentation complexes, ...





# Adaptation

- 1- Les utilisateurs veulent accéder aux contenus complexes
- 2- Les utilisateurs veulent accéder au même contenu en utilisant différents appareils limités



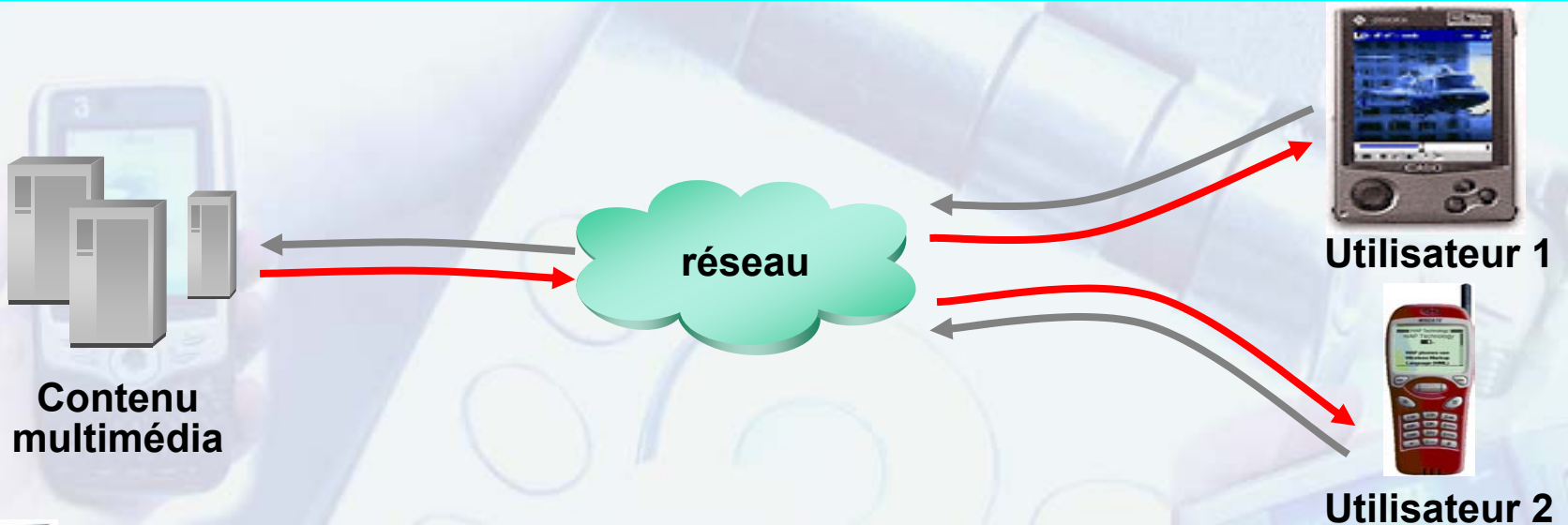
- Capacités d'affichage
- Protocoles d'accès
- Support des médias
- Langages et formats de documents
- Réseaux

# Adaptation

Le contenu doit être adapté selon les préférences des utilisateurs et les capacités des terminaux

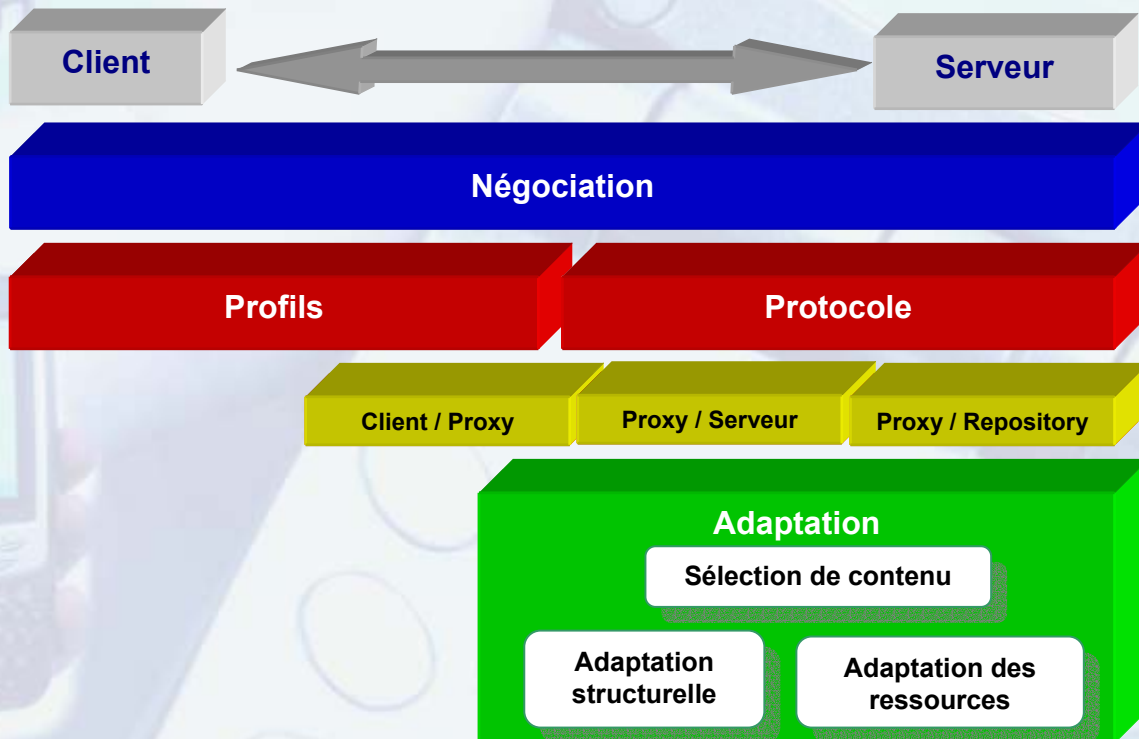
## L'adaptation et la négociation de contenu doivent assurer :

- La prise en compte du contexte
- La négociation du contenu sur la base des profils des clients
- L'adaptation du contenu des serveurs
- Assurer la transmission du même contenu sous différents formats

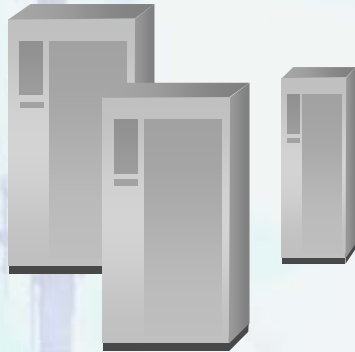


# Composants de l'architecture

Une architecture d'adaptation et de négociation de contenu peut être vue comme un ensemble de couches



# Négociation



Négociation



- Consiste à trouver un consensus entre les exigences du client, ses caractéristiques et les capacités d'adaptation du système

## Serveur :

- Transmission du même contenu vers différents clients : adaptation
- Gestion des variantes

## Client :

- Capacités matérielles
- Préférences utilisateur
- Formats de contenu supporté
- Protocoles utilisés & réseaux

- **Objectif d'une stratégie de négociation** : prendre la meilleure décision afin d'envoyer un contenu adapté au contexte du client





# Problématique de la Négociation

- Quel est le contexte de l'environnement (en termes de contraintes)?
- Comment assurer la gestion et l'analyse du contexte ?
- Le client est-il capable d'utiliser le contenu d'origine ?
- S'il y a plusieurs variantes du contenu d'origine, laquelle doit-on envoyer ? Comment assurer une gestion des variantes ?
- Le système est-il capable de transformer le contenu d'origine afin de satisfaire le contexte du client ?
- Quelles méthodes d'adaptation doit-on appliquer et comment ?
- Comment combiner les méthodes d'adaptation afin d'assurer de nouvelles capacités d'adaptation ?



# Négociation

- Description du contexte : cadre de travail CC/PP (Composite Capabilities/Preferences Profile), se limite à la description des clients
- Négociation : protocole HTTP/1.0 et TCN (Transparent Content Negotiation), négociation basée sur les variantes, description de contexte limitée
- Transformation et adaptation : - techniques limitées à des besoins spécifiques - les langages de transformations (ex : XSLT) manquent de mécanismes d'adaptation dynamiques

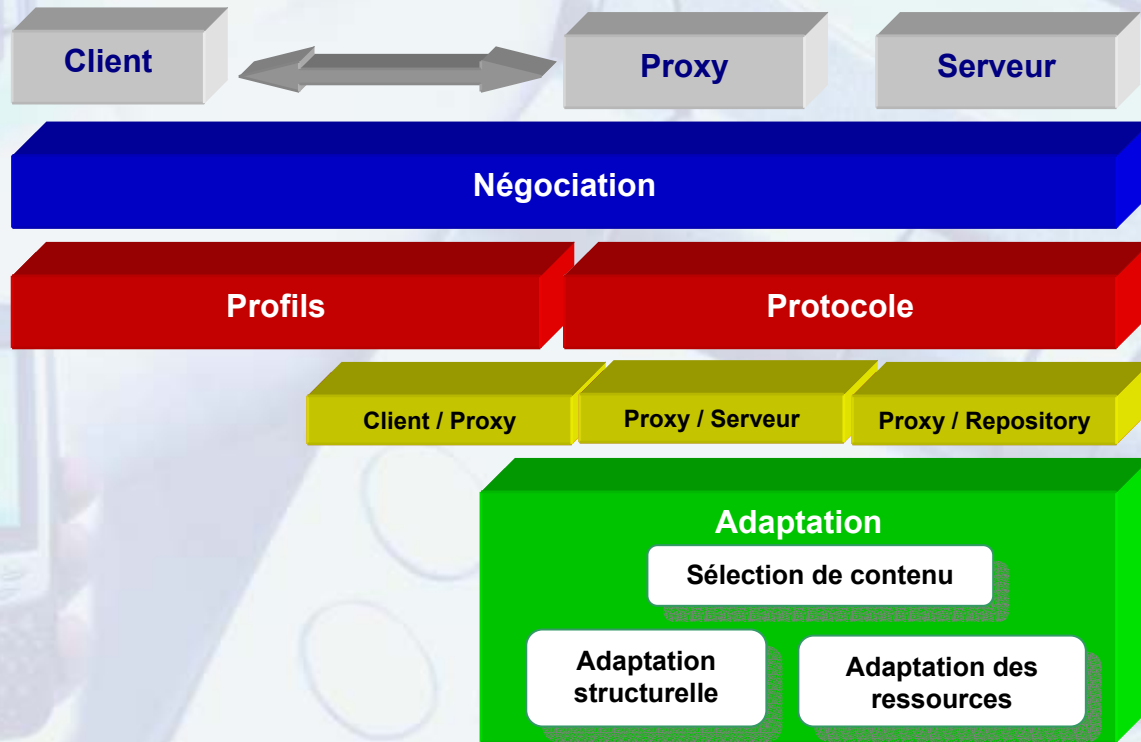


**Les solutions existantes sont limitées et ne résolvent le problème que partiellement**

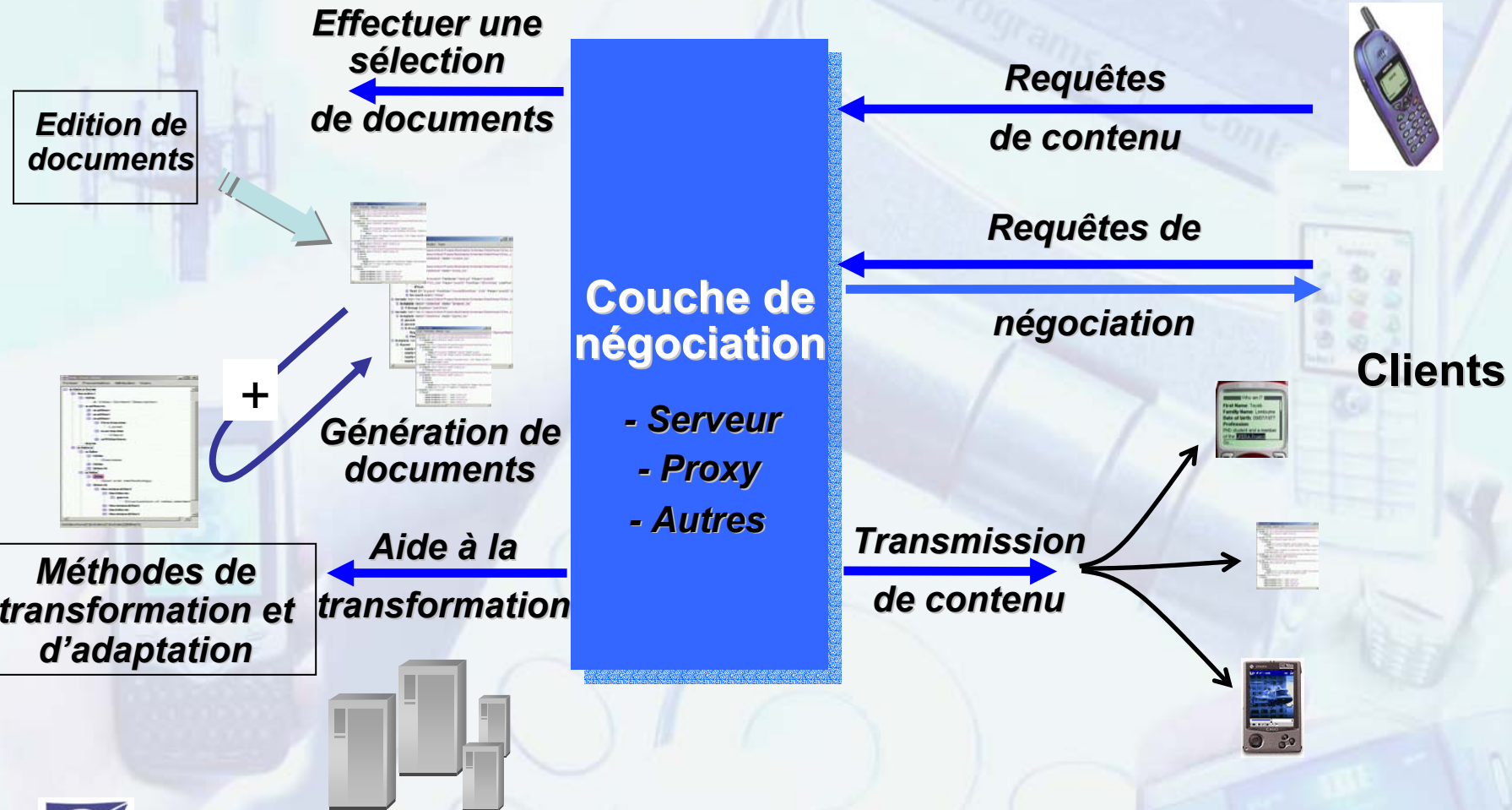


# Architecture NAC

L'architecture NAC (Negotiation and Adaptation Core) assure la négociation et l'adaptation de contenu dans un environnement hétérogène



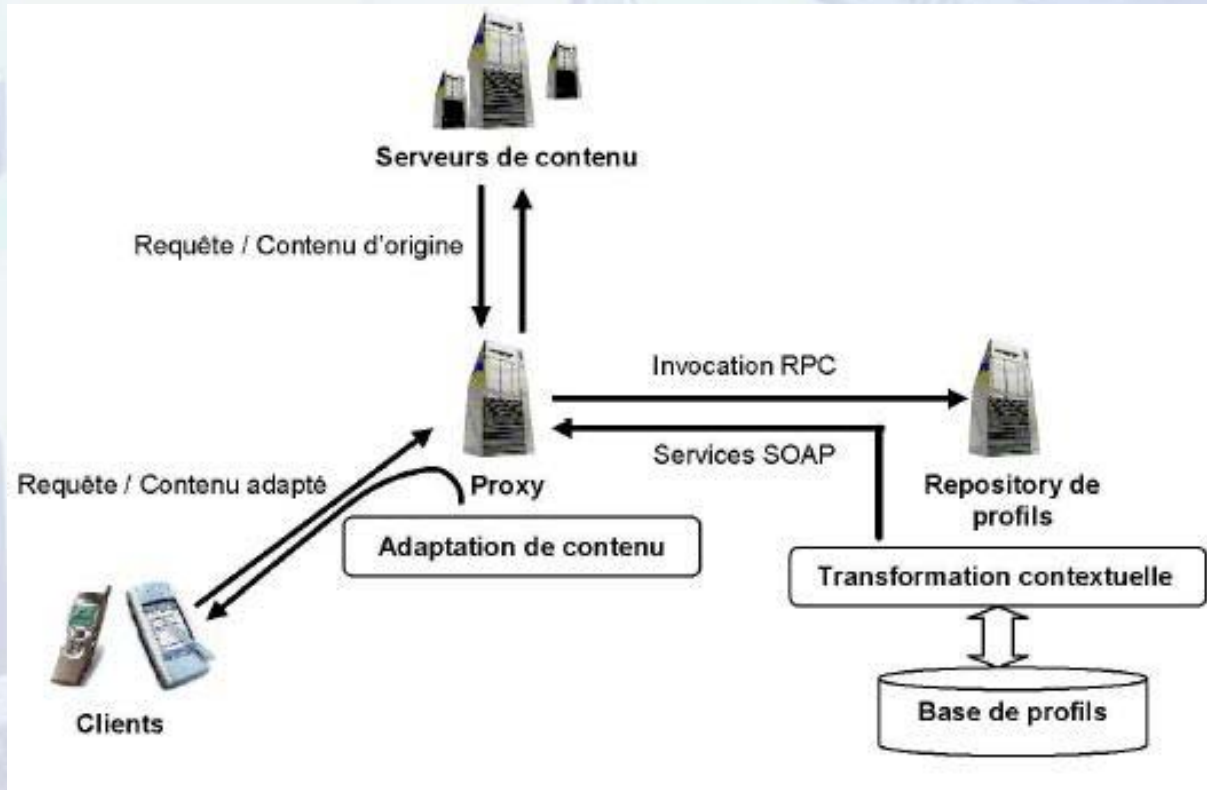
# Fonctionnement de la négociation





# Le rôle du proxy

- **Proxy** : entité intermédiaire entre l'ensemble des clients et l'ensemble des serveurs



- Garantir une adaptation universelle dans les architectures Client/Serveur

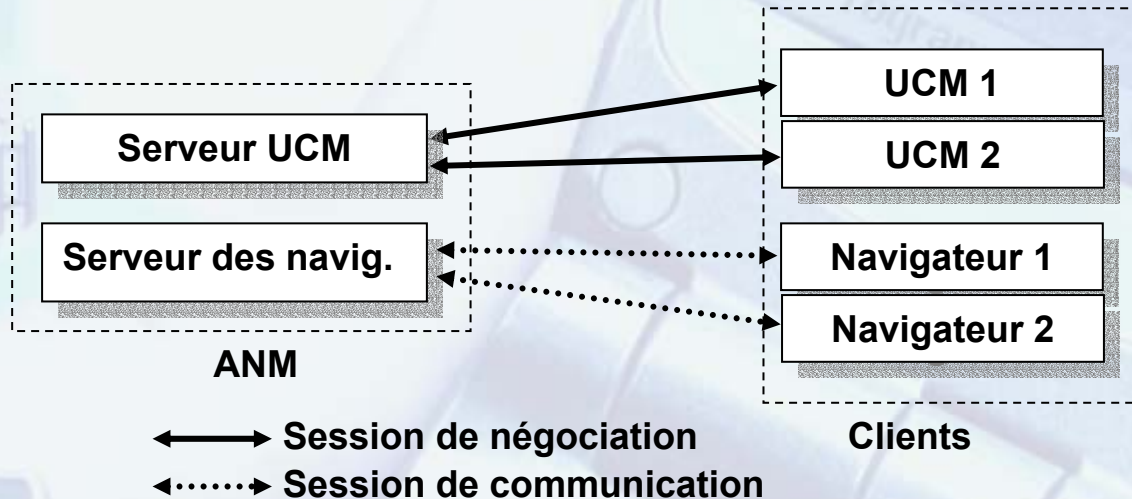
# Réalisation

## Architecture NAC

- ≈ 26 000 lignes de code java & C++
- Repository (intégrant des services Web) de 117 profils d'appareil, écrits en UPS
- Développement d'un **protocole de négociation**
- **Module ANM** : module d'adaptation et de négociation de contenu (intégrant les fonctionnalités d'un proxy)
- **Module UCM** : module pour les appareils mobiles (PDA, téléphones, etc.)
- **Négociation dans *PocketSMIL*** : négociation du navigateur SMIL
- **Méthodes d'adaptation de contenu**
  - Transformations structurelles
  - Adaptation des ressources médias
  - Adaptation des documents composites
- **Autres packages et APIs** : package CC/PP, adaptation temps réel de vidéo..



# Vue globale sur l'architecture NAC



- **Communication** : requêtes de contenu
- **Négociation** : information concernant le contexte et le changement du contexte
- **Multithreading** : utilisé pour supporter les accès concurrents des clients : navigateurs et modules UCM



# Entités principales de NAC

- 1 **Module d'adaptation et de négociation (ANM)**
- 2 **Transmission du contexte : UCM et repository de profils**
- 3 **Présentation de contenu multimédia pour les systèmes embarqués (PocketSMIL)**
- 4 **Méthodes et techniques d'adaptation**





# Repository de profils

**Objectif** : garantir une gestion et une analyse efficace des profils des clients

- **Fournir des services relatifs au traitement des profils :**

- **Interrogation de profils**
- **Extraction de contextes**
- **Mises à jour des contextes**

- **Minimiser le trafic et l'échange de messages de négociation**

- **Fournir des services qui peuvent être utilisés par les entités de l'architecture (proxy, serveur, client)**

Méthodes exposées	Paramètres	Signification
getProfile	profileID	interroge un profil dans le repository
getContextAtomicValue	profileID CCPPComponentID contextEntity	interroge une valeur dans un profile
getSubContext	profileID XPathExpression	interroge un sous contexte dans un profile
updateContextAtomicValue	profileID CCPPComponentID contextEntity	met à jour une valeur dans un profile

Méthodes exposées du repository SOAP

Repository (publication HTML) :

<http://opera.inrialpes.fr/people/Tayeb.Lemlouma/MULTIMEDIA/UPS-Client-Repository/ups-client-repository.html> (117 appareils mobiles)



# Protocole d'acquisition

- Indépendant du protocole de communication → peut être utilisé avec les protocoles existants tels que HTTP.
- Le protocole définit un ensemble minimal de requêtes :
  - GET\_GLOBAL\_PROFILE
  - OK\_SENDING\_PROFILE
  - OK\_SENDING\_CHANGE
  - NO\_PROFILES\_CHANGE
  - NO\_PROFILE\_ACQUISITION
- Les requêtes sont échangées entre le serveur *UCM* (proxy ou serveur) et le module *UCM* du client

# UPS : Schémas de profils universels (Universal Profiling Schema)

Un nouveau cadre de travail qui étend CC/PP et HTTP pour l'adaptation et la négociation du contenu

## UPS est basée sur



**CC/PP : Composite Capabilities/Preference Profiles**

<http://www.w3.org/2000/07/04-ccpp#>



**RDF : Resource Description Framework**

<http://www.w3.org/1999/02/22-rdf-syntax-ns#>

## UPS ajoute



**Extension : Six nouveaux schémas spécifiques à la négociation de contenu**

[http://www.inrialpes.fr/opera/people/Tayeb.Lemlouma/  
NegotiationSchema/\\*03012002#](http://www.inrialpes.fr/opera/people/Tayeb.Lemlouma/NegotiationSchema/*03012002#)



# Adaptation : SMIL vers Vidéo

<par>

```
<video id="vid" region="region_video" src="Videos/orange4_f.mpg"/>
```

```

```

```

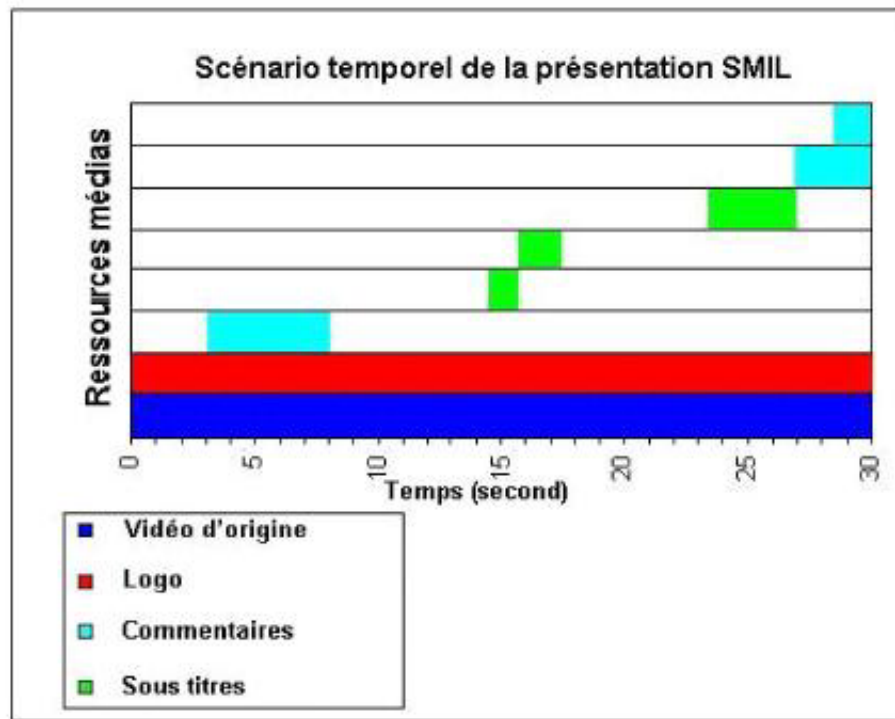
```

</par>





# Évaluation



Scénario temporel des présentations SMIL

## Vidéo 2

écomprimé  
(24 bits)  
x288

25.0

227940

750

0.04

39577

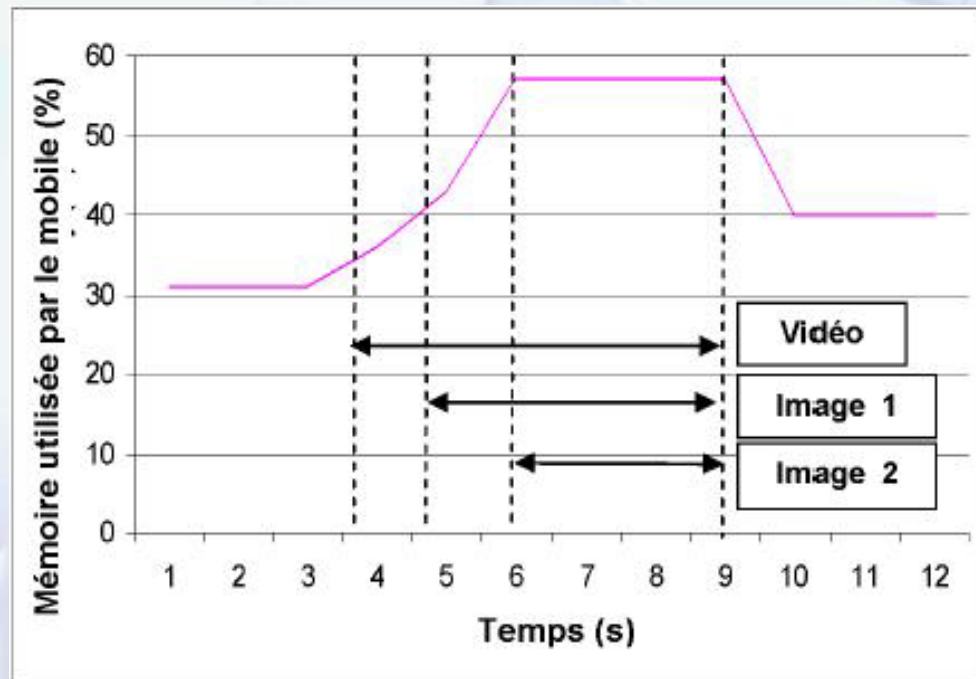
Temps d'



# Évaluation

## Adaptation de la vidéo pour la dimension du contexte : mémoire utilisée

Adapter le flux vidéo sur la base des capacités du client. La dimension considérée est l'utilisation en cours de la mémoire afin d'éviter la dégradation des performances du client



*Filtrage de trames*

$\alpha$

*Transmission de flux*

Filtrage de trame lorsque l'utilisation de la mémoire dépasse une limite pré-estimée  $\alpha$



# Groupe de travail : Device Independence

- **La mission du groupe de travail d'indépendance aux terminaux est d'éviter la fragmentation du Web en un ensemble d'espaces accessible seulement par un sous ensemble de clients.**
- **En particulier le groupe**
  - Définit les conditions d'accès au Web par de multiple terminaux.
  - Révision des spécifications internes ou externes au W3C
  - Fournir des cas d'utilisation pour les différentes activités du consortium
  - Définition des techniques qui permettent aux auteurs d'améliorer l'indépendance aux terminaux de leurs contenus
  - Proposer, dans des domaines spécifiques, non couverts par les autres groupes, des recommandations qui améliore l'indépendance aux terminaux



# Groupe de travail : Device Independence

- CC/PP
- Terminologies DI
- Techniques d'édition pour l'indépendance aux terminaux
- Principes d'indépendance aux terminaux
- Sélection de contenu
- Caractéristiques de présentation de contenu





# Conclusions

- Les réseaux et les systèmes d'accès à l'information deviennent de plus en plus hétérogènes : appareil d'accès, contenu, applications, protocoles, etc.
- Définir des stratégies d'adaptation devient nécessaire afin de garantir un accès au contenu qui satisfait l'environnement hétérogène.
- La négociation permet d'effectuer les meilleures techniques d'adaptation afin de transmettre, en final, un contenu qui considère les limitations matérielles des clients et les préférences des utilisateurs.
- Efforts : architecture NAC, travail continu du groupe de travail d'indépendance aux terminaux du W3C.





# Merci

**E-mail** : [Tayeb.Lemlouma@inrialpes.fr](mailto:Tayeb.Lemlouma@inrialpes.fr)

**Web** : <http://wam.inrialpes.fr/people/lemlouma/>

