Client-Side Context Storage

Stephanie Riche
HP Labs Grenoble

Client-Side Context Storage

HP Labs Grenoble

- Created in April 1999
- Collaboration with European labs & universities
- Research on local environment of appliances with personalization

Client-Side Context Storage

Overview

Goal:

Enabling personalization of the digital experience

Requirements:

1. *Get knowledge* : Rich user context information

   User preferences & habits , user location, time, activity, environment, history of interaction with services…

2. *Store knowledge* : Protection of user Privacy

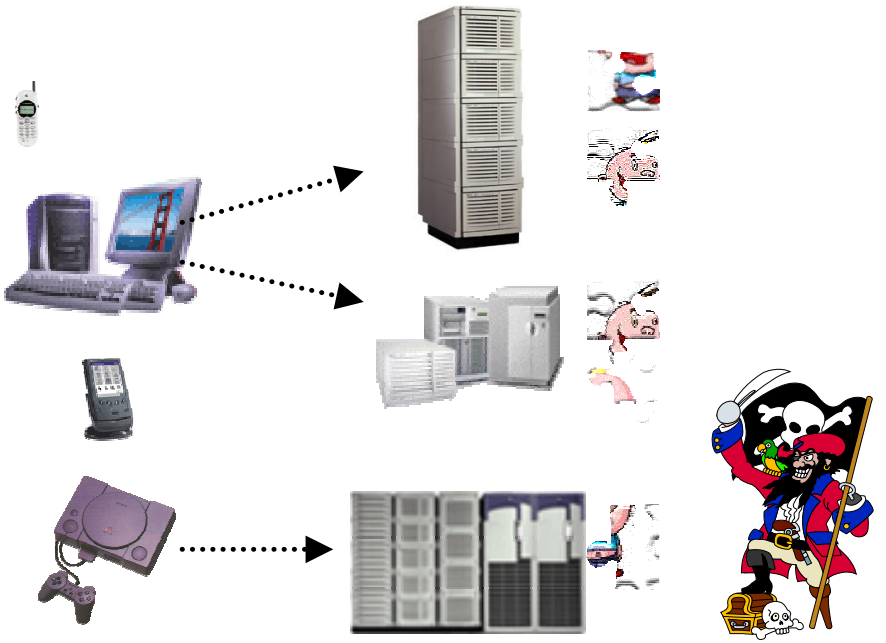
   Access context information from anywhere, any user device in a trusted way

3. *Use knowledge* : Personalize the interaction

   User interfaces, web services, smart spaces…

Client-Side Context Storage

Existing personalization & context-aware architecture

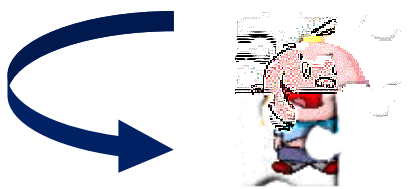


Drawbacks:

- Low privacy protection
- Inconsistence of data

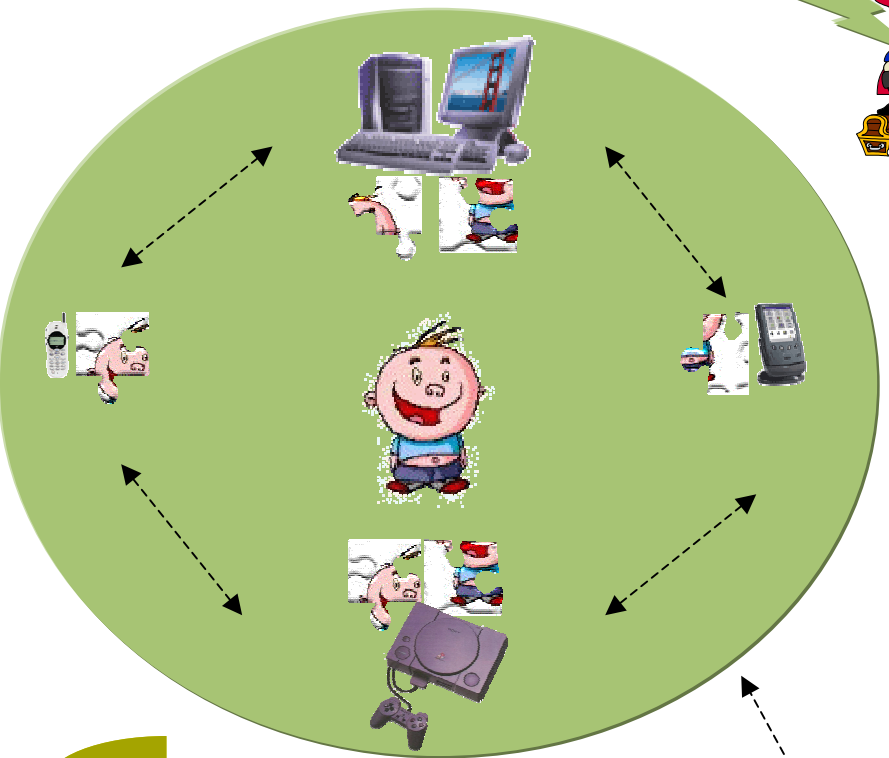
=> Frustration of the user.

Server-side & service-centric approach



Client-Side Context Storage

User-centered architecture



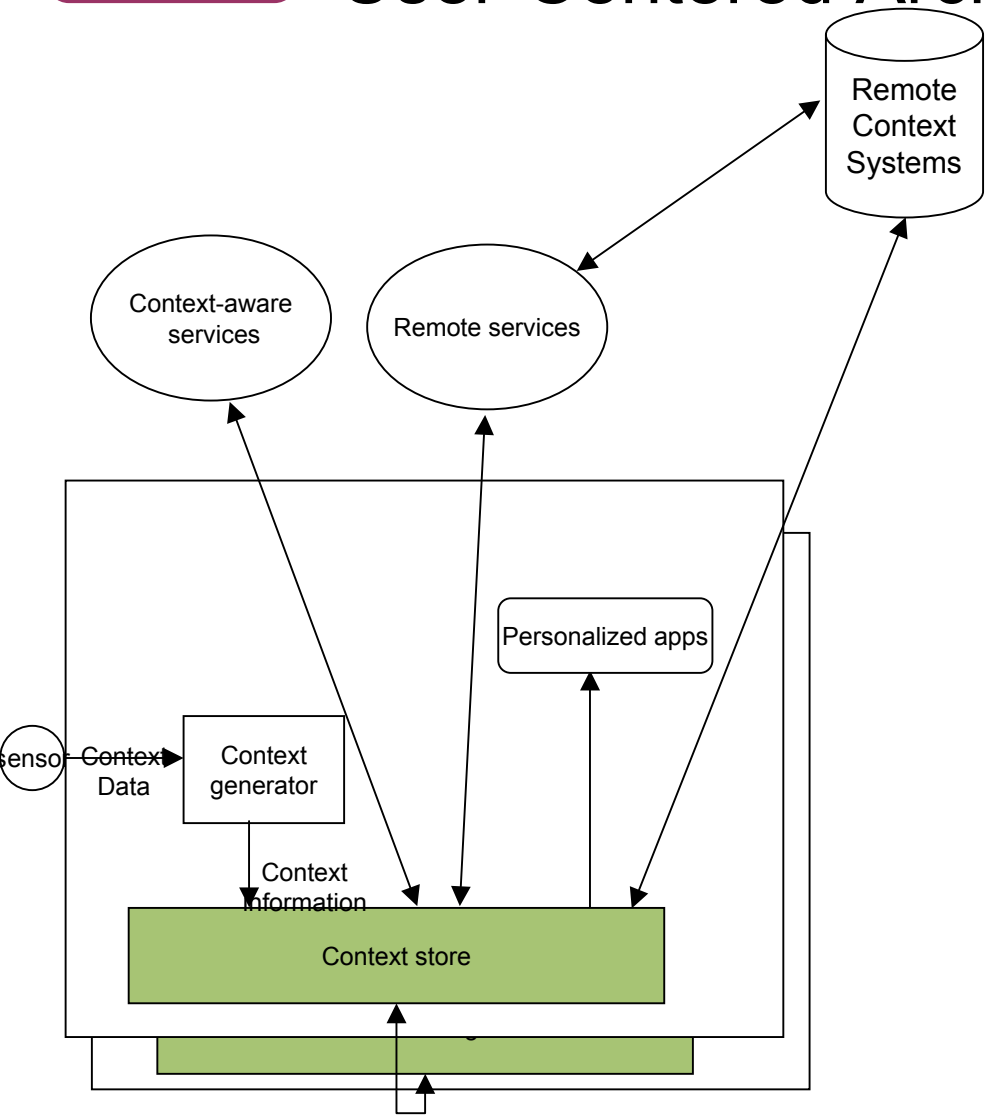
Advantages:

- Privacy protection.
- Context data consistency
- Always available context

=>Unified & enhanced customer experience.

Client-Side Context Storage

User-Centered Architecture



The context storage system on user devices can be used by remote context systems as:

- A “cache”
- An additional source of context information

Client-Side Context Storage

Context Storage System Requirements

- Simple memory model, no “hot sync” effect:
  - Minimize user involvement to reconcile replicas
  - Awareness about information consistency: **dependable system**

  => limit system divergence
- Adapt to device characteristics

  => Selective replication, asynchronous communication
- Personal storage
  - Privacy enforcement : **privacy awareness & control**

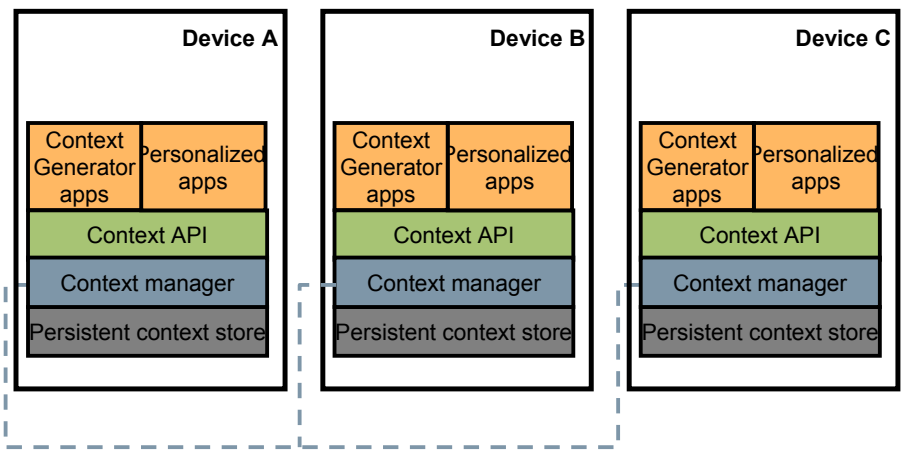
    => Control & log of accesses
  - Personal usage

  => Adapt to user behavior

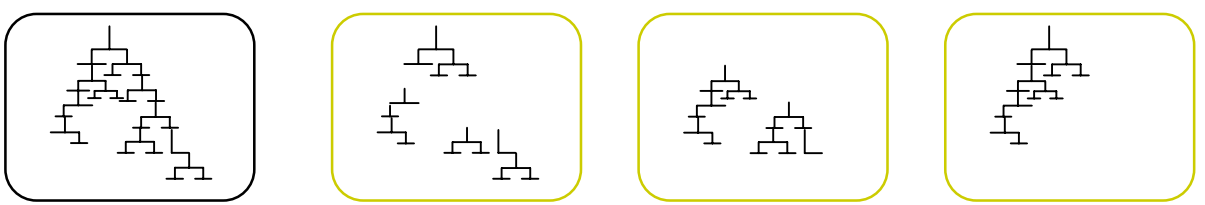
Client-Side Context Storage

Prototype design

- Single master optimistic protocol
- Migration of master replicas along the user
- Pro-active replication based on observed access patterns

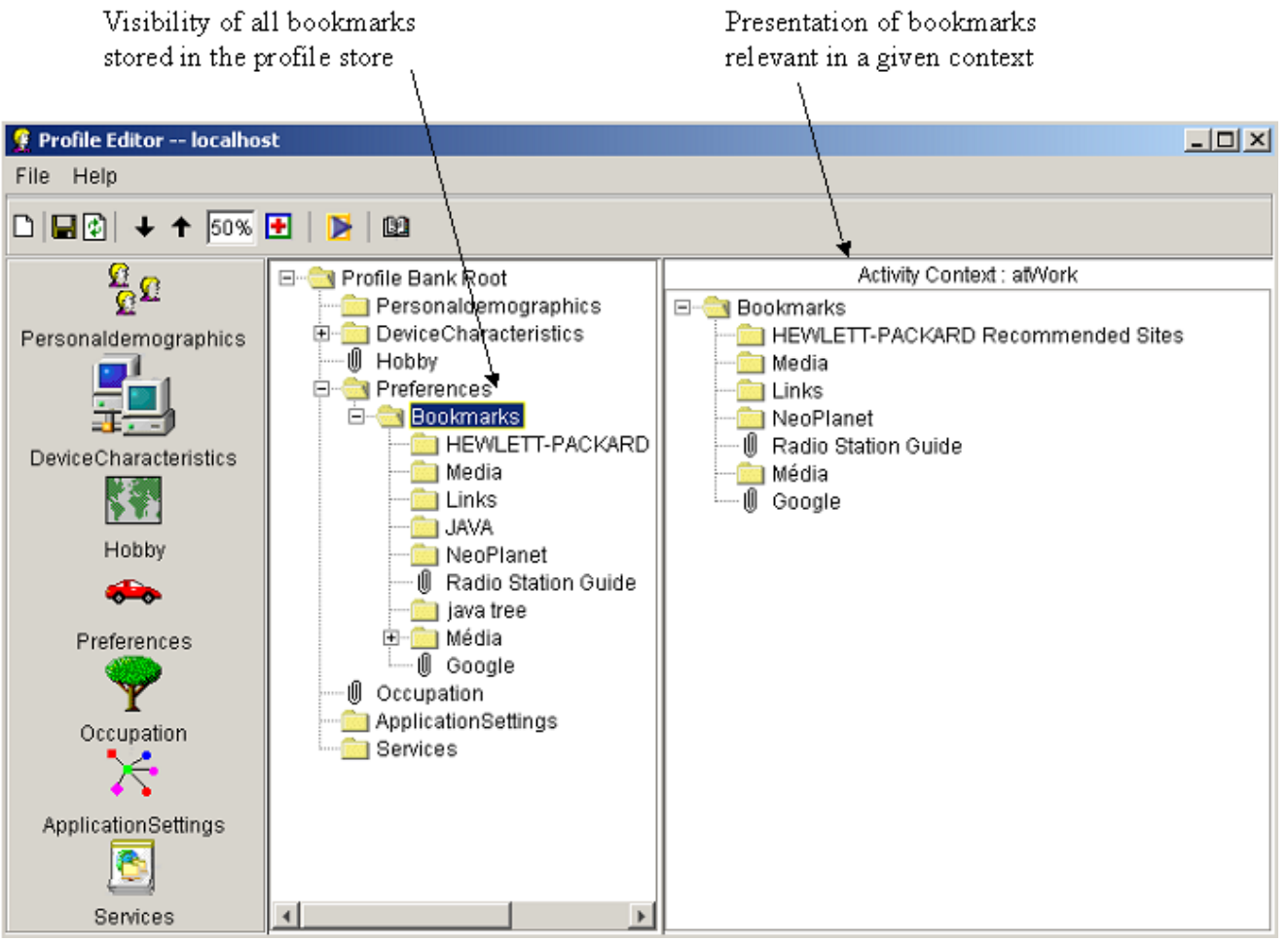


Virtual representation of the full context




hp
invent

Demos


Visibility of all bookmarks stored in the profile store
Presentation of bookmarks relevant in a given context
Profile Editor -- localhost
File Help
50%
Personaldemographics
DeviceCharacteristics
Hobby
Preferences
Occupation
ApplicationSettings
Services
Profile Bank Root
Personaldemographics
DeviceCharacteristics
Hobby
Preferences
Bookmarks
HEWLETT-PACKARD
Media
Links
JAVA
NeoPlanet
Radio Station Guide
java tree
Média
Google
Occupation
ApplicationSettings
Services
Activity Context : atWork
Bookmarks
HEWLETT-PACKARD Recommended Sites
Media
Links
NeoPlanet
Radio Station Guide
Média
Google

Client-Side Context Storage

Context Storage System versus Distributed file systems

	Context Storage System	Distributed file systems
Access patterns	Conflicts are frequents & user involvement should be minimized	Conflict are rare & user involvement is ok
Number of users	1	Usually hundreds
Data requirements	Usually unknown by the user but could be pro-actively fetched.	Usually unknown by the user but could be pro-actively fetched
Data Types	Heterogeneous, privacy requirements at a fine grain level	Files

Client-Side Context Storage

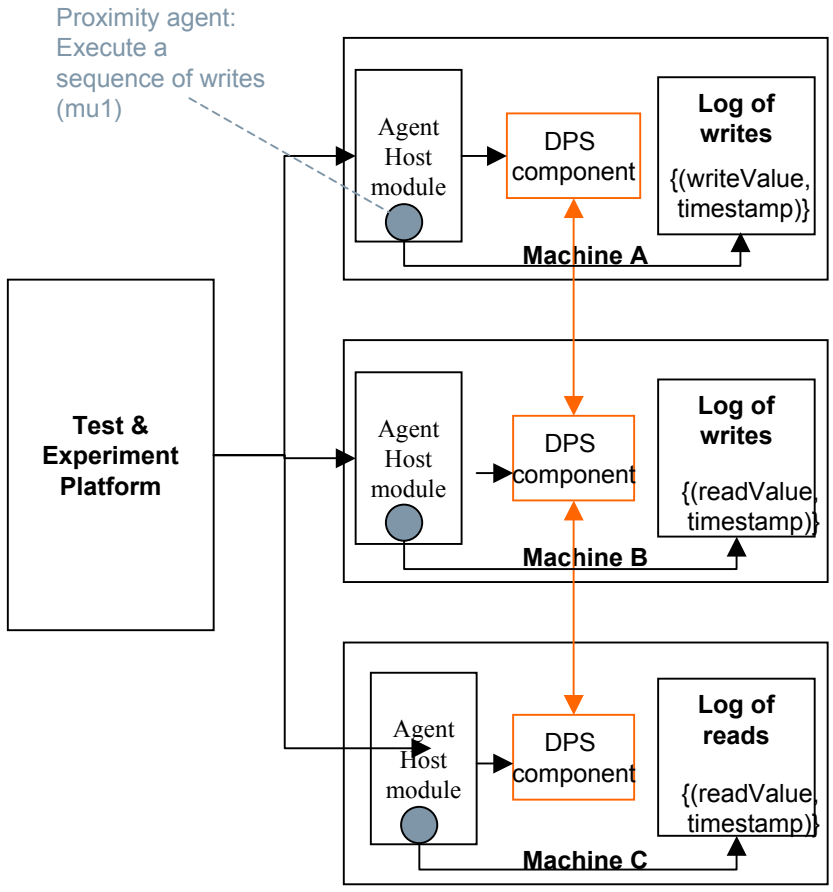
Tests & experimentations

• **Goals:** Rapid experimentation of emerging concepts. Measure metrics to compare several approaches

• **Experimentation framework**

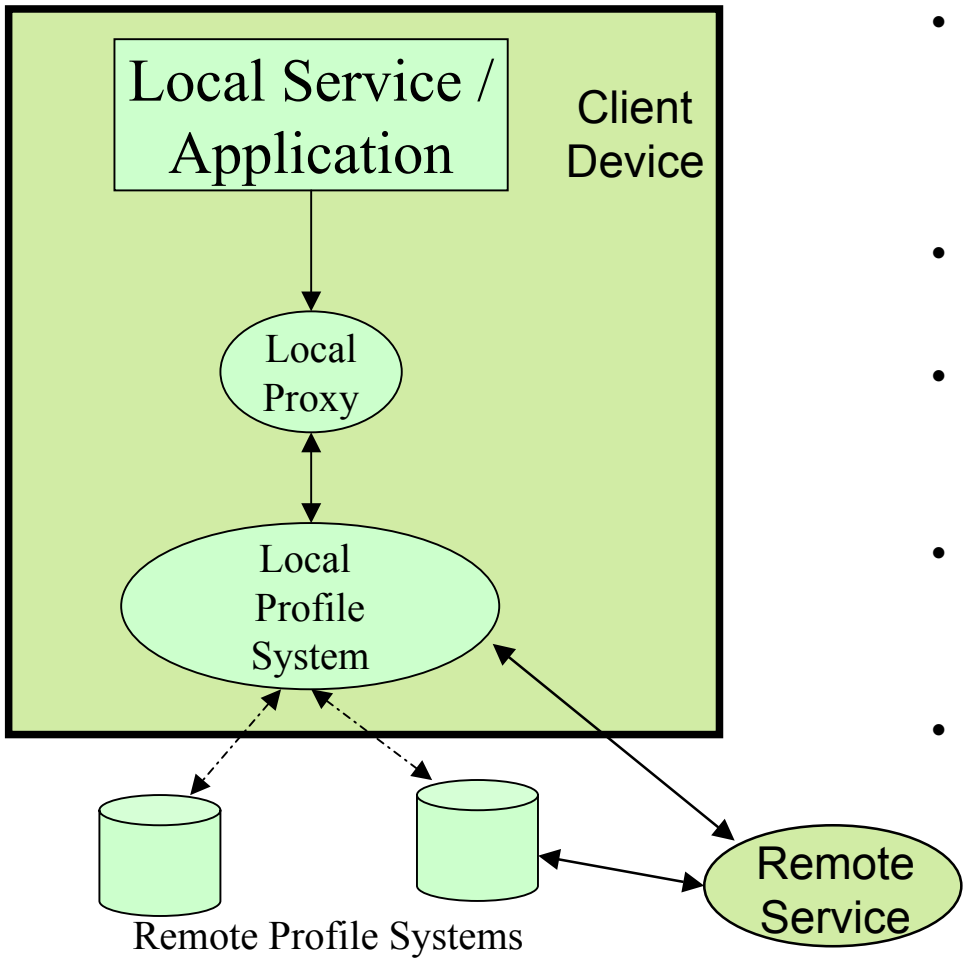
Framework allowing rapid experimentation on a distributed system. Address the issue of rapid deployment and execution of a distributed scenarii.

- On each machine resides an **agent host** in charge of receiving proximity agents.
- A **proximity agent** can be deployed on a remote machine, can migrate with state information upon machines. Proximity agents can be remotely accessed for information collection or configuration purpose.



Client-Side Context Storage

An aside - Client-side Federation



- A local profile system on a client device can be used as a “cache” for remote profile systems.
- Synchronization when device is connected to internet.
- Intercept calls to remote profile systems, and redirect locally.
- Remote services may call client device profile or remote one.
- Local device may have local only data as well as cached data from remote systems.