A Content-Centric Platform for Home Networks

Pai-Hui Wang
Institute of Computer Science and Engineering, College of Computer Science
National Chiao Tung University
Hsinchu, Taiwan, Republic of China
artistwang@nctu.edu.tw

Tse-Han Wang
Network Management Laboratory
Chunghwa Telecom Laboratories
Taoyuan, Taiwan, Republic of China
wangth@cht.com.tw

Shih-Ting Lin
Institute of Computer Science and Engineering, College of Computer Science
National Chiao Tung University
Hsinchu, Taiwan, Republic of China
stlin@cs.nctu.edu.tw

Pei-Wen Chen
Institute of Network Engineering, College of Computer Science
National Chiao Tung University
Hsinchu, Taiwan, Republic of China
pwchen@cs.nctu.edu.tw

Chien-Chao Tseng
Department of Computer Science
National Chiao Tung University
Hsinchu, Taiwan, Republic of China
cctseng@cs.nctu.edu.tw

Jiun-Long Huang
Department of Computer Science
National Chiao Tung University
Hsinchu, Taiwan, Republic of China
jlhuang@cs.nctu.edu.tw

***Abstract* — Due to the advance in wireless communication technologies, smartphones and tablets are becoming part of our life. People can use mobile devices to take pictures to share their life with others. In consideration of privacy and usability, this paper focuses on developing a private platform for content sharing in home. To facilitate content management and fast content retrieval, we design a Content-Centric Platform (CCP) that allows users to manage and search contents according to the tags (or attributes) of contents. With CCP, family members can access shared contents by the tags of contents without knowing the exact physical locations of the contents. The CCP platform has an auto-tagging subsystem that can add annotated tags automatically for the shared contents. It then applies a data mining technique to analyze the tags of the contents and creates a Tag Tree to represent the association of contents based on the tags. Furthermore, CCP also adopts a Tag Cloud mechanism to guide users to retrieve contents of interest. The functionalities provided by CCP include device joining and leaving, content information collection and distribution, auto-tagging and tag analysis. Finally, we have implemented a prototype system to verify the effectiveness of the CCP for content sharing in home networks.**

Keywords—Home network, Content-centric, Contents sharing, Auto-tagging, Tag association.

I. Introduction

With the increasing popularity of mobile devices, people are able to capture photos to record their life easily. Therefore, the volume of the user-created contents is increasing at a fast pace. People used to share their contents with other family members by showing photos directly or via USB drives, but both ways are time-consuming and inconvenient. If deciding to upload their photos to online content sharing services, they have to worry about the security and privacy issues.

Several studies have proposed some services to address the aforementioned problem. Park and Kim proposed in [1] an integrated directory service for contents distributed across devices. Díaz-Sánchez et al. proposed in [2] an open middleware able to share content from inside network to cloud. However, the proposed middleware does not support automatic annotation and tag-guided browsing. Gambette and V´eronis proposed in [3] a words (tags) display method which are able to visualize a text as a tree of word clouds. The proposed method is helpful in classifying the main topics of documents. However, the proposed method does not support narrow down search. Song et al. proposed in [4] a structure-based tag recommendation approach. It supports narrow down search for content search, but does not provide tag trees in global view.

Traditional content sharing systems such as NAS/FTP servers require administration effort. UPnP/CIFS (Common Internet File System) [5] do not collect content information, because they are directory hierarchy browsing on the basis of filename and directory-based searching/browsing. In addition, they have arbitrary naming convention and decentralized view of shared folders since the contents are distributed over devices. Therefore, it is time-consuming and hard to find target contents via the above systems.

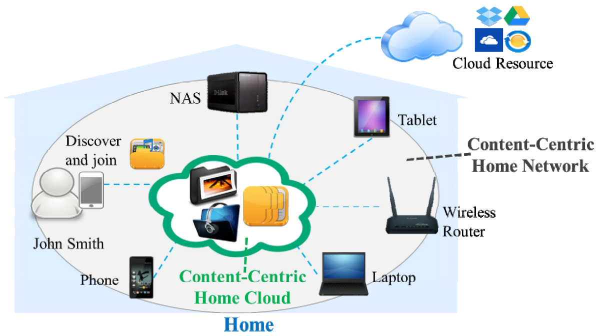

Fig. 1. CCHN Usage Scenario

In this paper, we design the Content-Centric Platform (CCP) for Home Networks (CCHN) to facilitate efficient and convenient photo sharing among family members. The architecture of CCHN is shown in Fig. 1. All home devices with storage and network connectivity can participate in the CCHN. Although the scope of CCHN is in-home, a "proxy node" can be created at the home router (e.g., home NAS) or at the cloud to be part of a CCHN. Moreover, each object could be stored in more than one devices. The directory service should take the content availability, capability to be distributed or not, current loading and "grade of host" (a smartphone, a laptop, an AC-powered PC or NAS...) into consideration. For example, if a photo is available at a NAS and a smartphone, the directory service should direct the requestor to the NAS, instead of the smartphone (since it is power-limited and mobile). The mobile contents could be in and out of premises without warning. The directory service should be aware of the content availability as the device moving in and out of premises. The proxy host can request all the new contents as they become available and serve the contents to family members. Each content should be either given a unique name or associated with one or more aggregated names within the namespace. Contents are searched or exchanged by full or partial names. Contents can also be associated with one or more aggregated names, e.g.

"John Smith's 16th birthday". Contents should be encrypted and only family members can decrypt them.

In addition, CCHN has two important functions. One is Auto-Tagging, which automatically utilizes personal information and cloud resources to annotate local contents stored in each device. The other is Tag Analysis which employs a data mining algorithm to generate tag association from global view tags. Therefore, CCHN organizes the resultant tag association as Tag Cloud or Tag Tree to facilitate convenient content searching.

II. PROPOSED ARCHITECTURE

A. *Design Concept of CCHN*

■ CCP Use Case

As shown in Fig. 2, we used UML (Uniform Modeling Language) to describe the features, functions and use cases of CCP.

- Device Discovery: discover each other's presence on the network.
- Proxy Host Selection: User assigns a device to act as the CCP proxy.
- Device join: CCP proxy and CCP devices form the CCHN, and maintain content information of CCHN.
- Content Sharing: CCP devices can browse and download all contents in CCHN.
- Content Collecting: Proxy Host can collect content or tag information (Metadata) from CCP Devices.

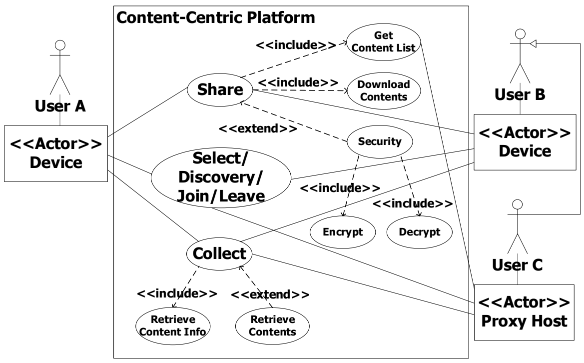

Fig. 2. CCP Use Case Diagram

B. *Overview of CCHN System Architecture*

Fig. 3 shows the system architecture of the CCHN, which integrates home devices with storage (such as smartphone, NAS, tablet, wireless router and laptop) to manage and share contents via tags, and guides users to browse contents with the aid of Tag Cloud/Tag Tree.

CCHN has three layers. Application (User Agent) layer displays contents based on Tag Cloud and Tag Tree. Content-Centric Platform layer formed by home devices with storage aggregates content information of distributed contents. Content-Centric Platform layer includes two important plug-ins modules: Auto-Tagging and Tag Analysis modules. Moreover, Home Networks layer manages the underlying network protocols such as UPnP, and DLNA.

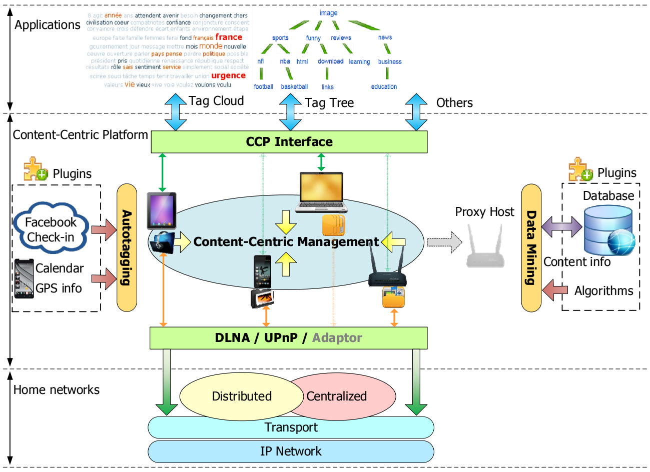

Fig. 3. CCHN System Architecture

Moreover, two main entities are as follows.

- ◆ CCP Devices (Content provider and consumer): To maintain local contents, CCP devices generate metadata (including content info, tag info and content or tag relation info) via Auto-Tagging module so that users can browse remote contents by Tag Cloud/Tag Tree.
- ◆ CCP Proxy: To collect content or tag information, CCP proxy computes aggregated data from metadata and pushes aggregated data to CCP devices.

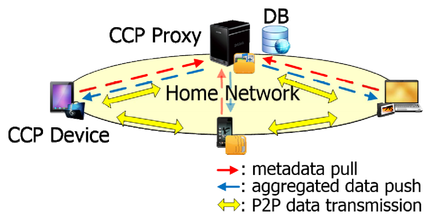

Fig. 4. CCP Device and Proxy Entities

C. *Software Architecture and Components*

■ Software Components

Fig. 5 shows the design of CCP device and CCP proxy.

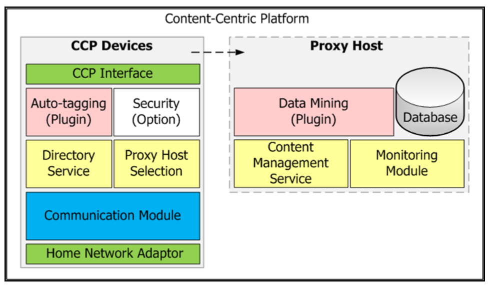

Fig. 5. CCP Software Components

- ◆ CCP Devices
  - CCP interface: Provides an unified way to access aggregated data for APP UI to form the Tag Cloud/Tag Tree.
  - Directory Service: Provides

- Local list (content) and Content info. CID, title, format, URL, file path, etc. and monitors share folder to keep content list.
- Remote list (content) provides aggregated data of remote contents.

● CCP Proxy Selection: Devices negotiate to elect a powerful device as the CCP proxy.

● Communication Module: Handles control and data connections between devices via an HTTP client and server.

● Home Network Adaptors: May support any device discovery technology. We currently use UPnP adaptor to provide UPnP-related functionalities.

◆ CCP Proxy

● Content Management Service

- Pull content or tag information (metadata) from CCP devices periodically.
- Push aggregated data to CCP devices after collecting metadata.

● Monitoring Module: Maintain availability of devices for listening announcements when CCP devices send join requests, and send search requests for Device discovery.

■ Software Architecture

As shown in Fig. 6, the architecture is formed by User Agent, CCP devices and CCP proxy. There are some differences between CCP device and CCP proxy. There are some gray out modules (Content Management Service, Monitoring, and Data Mining) and interfaces in devices, because these modules and interfaces are only enabled when the device was selected to be a CCP proxy.

User Agent can send request or get content info via CCP Interface for displaying Tag Cloud/Tag Tree.

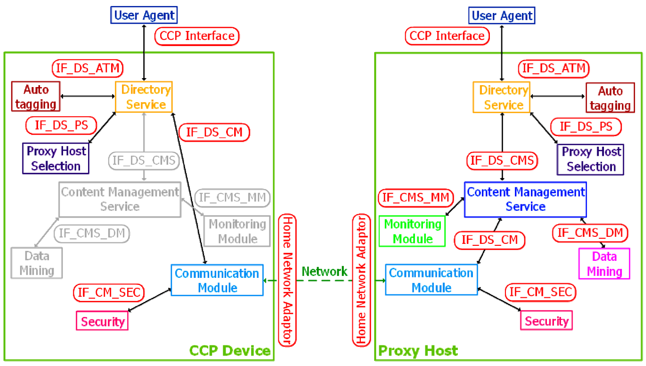

Fig. 6. CCP Software Architecture

■ Interaction of CCP device and CCP proxy

Fig. 7 shows the five-step simple interaction process between CCP devices and CCP Proxy, and these five steps are summarized in Table I.

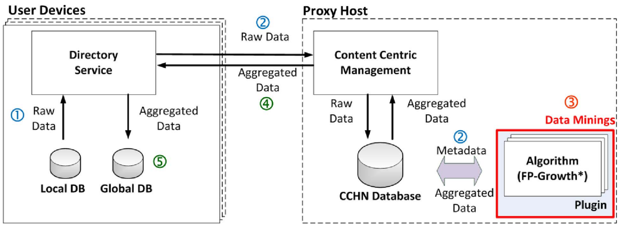

Fig. 7. Interaction Steps

TABLE I. INTERATION PROCESS

Process	CCP Device	CCP Proxy
Metadata collection	1. Create a local list	2. Pull Raw Data and merge metadata
Metadata analysis		3. Analyze association of tags
Aggregated data push	5. Render UI	4. Push aggregated data

■ Plug-ins Module – Auto-Tagging

The module extracts EXIF (Exchangeable image file format) data from images and match with date from calendar and GPS from Facebook places [7] and Google location [8]. Meanwhile, it provides APIs to User Agent for accessing tag information of contents.

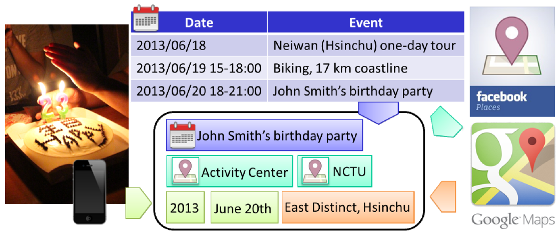

Fig. 8. Auto-Tagging Machanism

■ Plug-ins Module – Tag Analysis

● Tag Weighting Mechanism (for Tag Cloud)

Fig. 9. Tag Weighting Mechanism

● Tag Analysis Mechanism (for Tag Suggestion)

The Tag Analysis Mechanism is to analyze metadata to generate aggregated data. CCP adopts FP-Growth algorithm to find Frequent Patterns (FPs) [14] then based on FPs to construct Frequency and Association Tree (FA Tree) [15]. CCP then uses aggregated data to assist users in finding contents. Furthermore, CCP provides a common API to reconstruct Tag Cloud/Tag Tree from aggregated data.

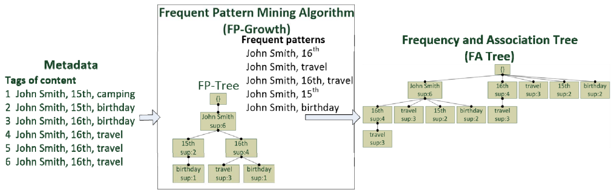

Fig. 10. Tag Association Mechanism

III. Implementation Result

A. Prototype of CCHN

We implemented a prototype of CCHN, and Fig. 11 shows the screenshots of CCP running in tablets.

Fig. 11. CCP Prototype Screenshots

B. Tag-Guided Search

We use the photos taken in a real travel to Tokyo as an example to show the prototype of CCP. Fig. 12 shows the global view of these photos via Tag Cloud and Tag Tree, while Fig. 13 shows the steps of narrow down search via Tag Tree.

Fig. 12. Tag Cloud and Tag Tree

Fig. 13. Narrow Down Search

IV. Conclusion and Future Work

In this paper, we proposed an architecture of Content-Centric Platform for Home Networks which provides an user-friendly search/browsing for content-sharing. So, users have no need to know the locations of contents. Besides, analyzing association of annotated tags can guide users finding desire contents, because we also provided a simply narrow down browsing way.

In the future, we will enhance the CCP proxy auto-selection mechanism, incremental update mechanism, and allow users to select friends dynamically from social networks and publish contents to the group of the selected friends.

References

[1] J. Park and S. Kim, “A Transparent Contents Sharing Service with Virtual Media Server,” in Proceedings of International Conference on Convergence Information Technology, 2007

[2] D. Díaz-Sánche, F. Almenarez, A. Marín, D. Proserpio and Patricia A. Cabarcos, “Media Cloud: An Open Cloud Computing Middleware for Content Management,” IEEE Transactions on Consumer Electronics, 2011

[3] P. Gambette and J. Ve´ronis, “Visualising a Text with a Tree Cloud,” in Proceedings of the International Federation of Classification Societies 2009 Conference, 2009

[4] Y. Song, U. Farooq and B. Qiu, “Hierarchical Tag Visualization and Application for Tag Recommendations,” in In Proceedings of ACM International Conference on Information and Knowledge Management, 2011

[5] UPnP Forum, “UPnP Device Architecture version 1.1,” 2008.

[6] Cling, UPnP/DLNA library for Java and Android, http://4thline.org/projects/cling/.

[7] Facebook APIs, https://developers.facebook.com/docs/reference/apis/ .

[8] Android Developers Location APIs, http://developer.android.com/google/play-services/location.html .

[9] J.-T. Kim, Y.-J. Oh, H.-K. Lee, E.-H. Paik and K.-R. Park, “Implementation of the DLNA Proxy System for Sharing Home Media Contents,” IEEE Transactions on Consumer Electronics, vol.53, no.1, 2007.

[10] M. Seltzer and N. Murphy, “Hierarchical File Systems Are Dead,” in Proceedings of the 12th Conference on Hot Topics in Operating Systems, 2009.

[11] A. Ghosh and I. Arce, “In Cloud Computing We Trust - But Should We?,” IEEE Security and Privacy, vol. 8, no. 6, 2010.

[12] M. Ames and M. Naaman, “Why We Tag: Motivations for Annotation in Mobile and Online Media,” Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, 2007.

[13] R. Rymon, “Search Through Systematic Set Enumeration,” Technical Reports (CIS), 1992.

[14] J. Han, J. Pei and Y. Yin, “Mining Frequent Patterns without Candidate Generation,” in Proceedings of the ACM SIGMOD International Conference on Management of Data, 2000.

[15] J.-L. Koh and S.-F. Shieh, “An Efficient Approach for Maintaining Association Rules based on Adjusting FP-tree Structures,” in Proceedings of the International Conference on Database Systems for Advanced Applications, 2004.