



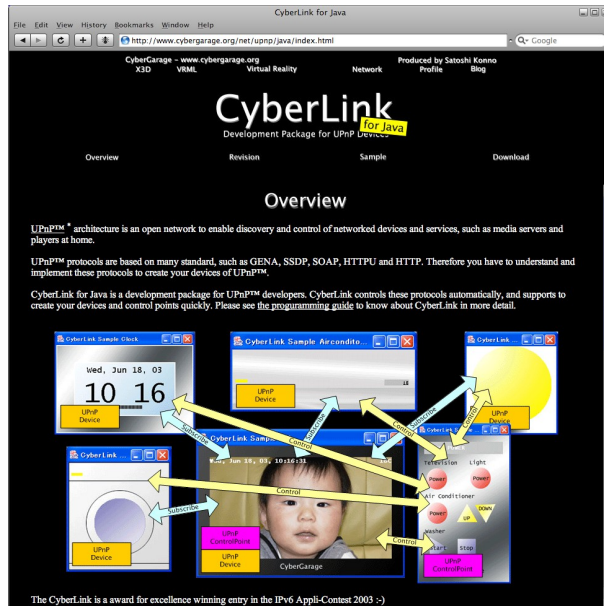
# Realization of Personalized Central Device for Internet Services in Home Networking using CyberLink on Androd.

Apr 14, 2008

Satoshi Konno  
skonno@cybergarage.org  
<http://www.cybergarage.org>

# What is CyberLink ?

CyberGarage



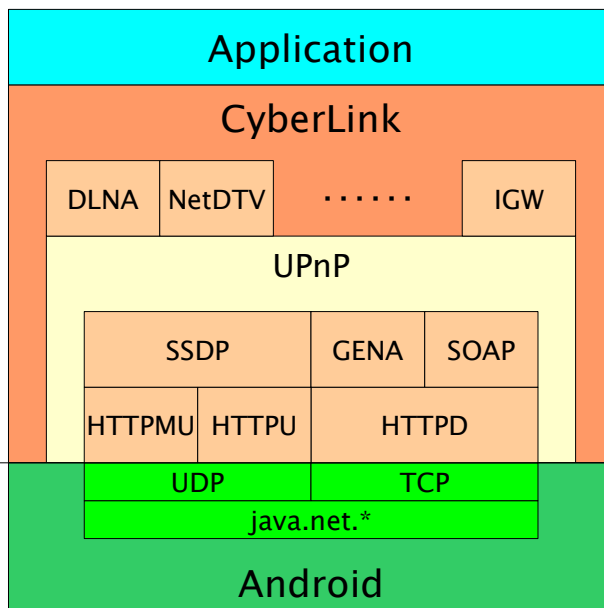
I am developing my applications for Android based on the following protocol stack which is my open source project.

CyberLink for Java

<http://www.cybergarage.org/net/upnp/java/>

Basically, the CyberLink is a protocol stack of UPnP\*1) with the upper protocol such as DLNA\*2).

I have ported the source codes into Android already, I am developing the applications now. I could port the source codes easily from Android SDK M5 because the SDK version supported multicast and fixed some major problems. Thanks :-)



I have released the protocol stack using other programming languages such as C++, C, Perl. **Epecially, Nokia joins the project as the developer and the project is used in their consumer product such as their Internet tablet.**

CyberLink for C

<http://www.cybergarage.org/net/upnp/c/>

I ported the project into other embedded platforms such as uTron and T-Engine which are Japanese OS.



Nokia Internet Tablet 770

\*1) UPnP : Universal Plug and Play – <http://www.upnp.org/>

\*2) DLNA : Digital Living Network Alliance – <http://www.dlna.org/>

# Development Concept for Android

Recently, there are many networked devices in home but the home devices are isolated from Internet because of some reason now.

I would connect the home devices with Internet services using my applications on Android devices. Using my applications, I hope that Android devices will be a personalized central device of the home networking.



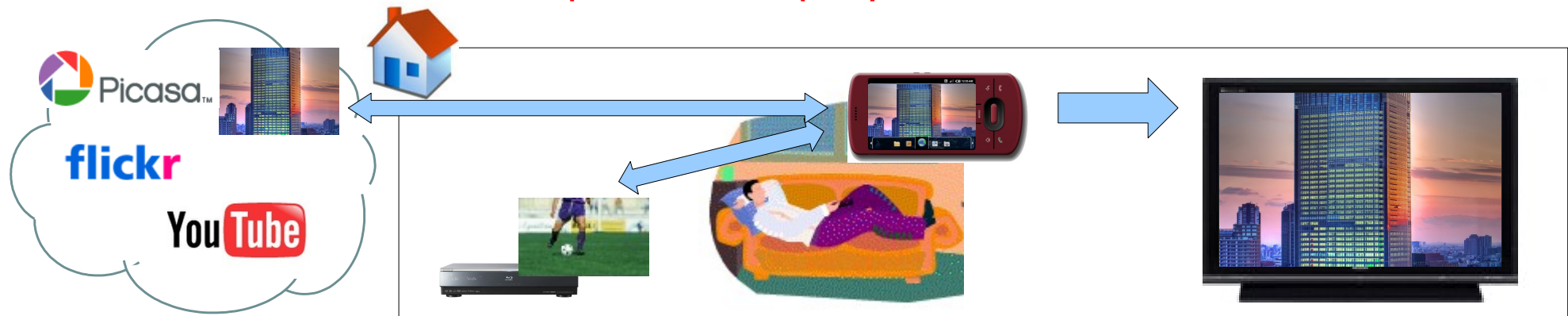
Using my application, the home devices can see or synchronize Internet contents with no hardware change because the my application emulates the home protocols as a proxy device.

Using my application on Android, the following use cases are available in the future. For the contest, I have implemented only some use cases yet.

## 1. Watch Internet contents on Television

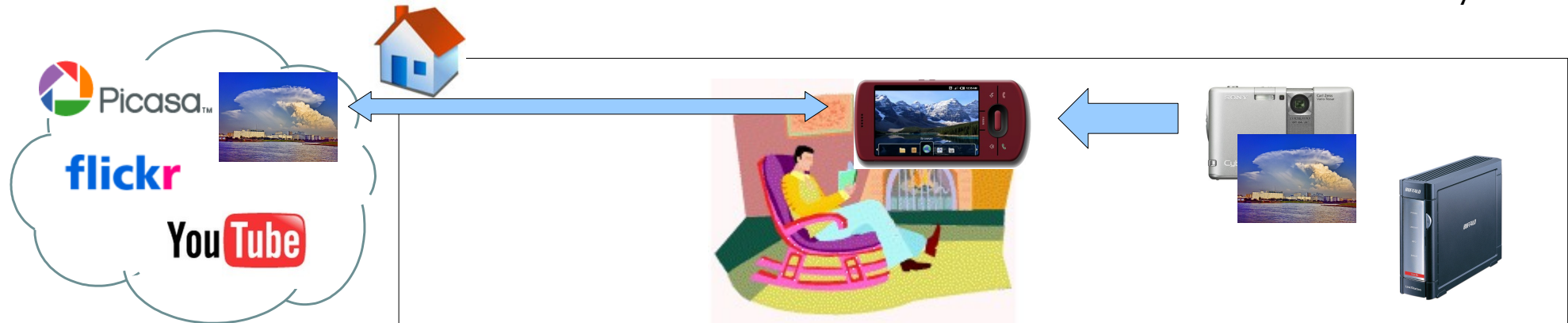
Using my applications, the user uses the Android device as a networked remote control, selects Internet contents and watches the contents on a other big television. The use case is based on DLNA v1.0 or v1.5.

The television of DLNA doesn't need to change the firmware to watch the Internet contents because the Android device emulates DLNA protocol as the proxy to transfer the Internet contents.



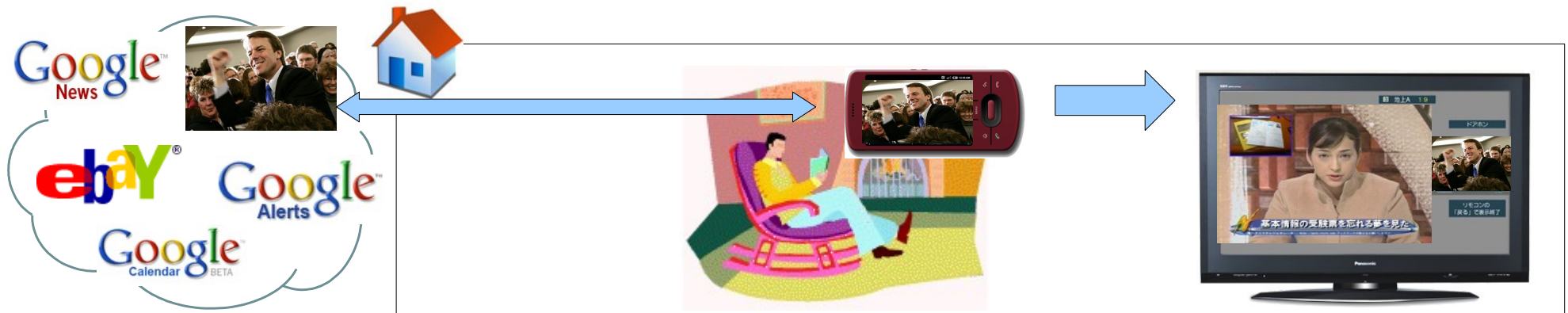
## 2. Synchronize Home contents with Internet Services Automatically

The user come back home with the networked home devices such as the digital camera, the Android device finds the new devices and uploads the contents into Internet services such as Picasa automatically.



## 3. Notify Internet contents into Television

The Android devices can interrupt and send any notifies of Internet to television whether the user see the broadcast contents in home. The use case is based on DLNA v1.5 or NDT\*3) v2.1 in Japan.



## 4. Access to Home contents and Devices

While you are out, you can access to your networked devices in your home using the Android device. The use case is based on UPnP IGD\*4) and the original firewall software which is implemented on Android or other devices such as PC or NAS.



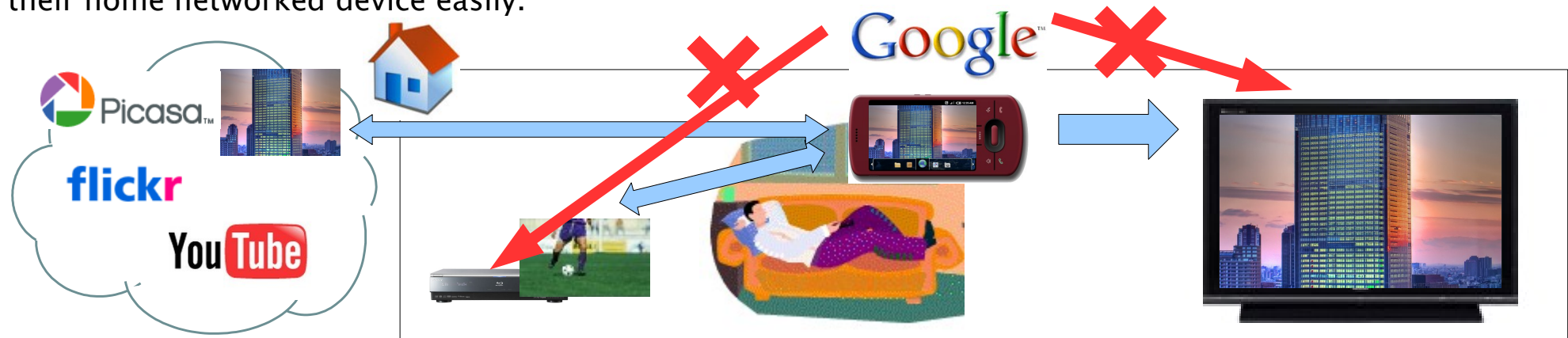
\*3) NDT : Networked Digital Television – <http://nw-dtv.jp/>

\*4) IGD : Internet Gateway Device – <http://www.upnp.org/standardizeddcps/igd.asp>



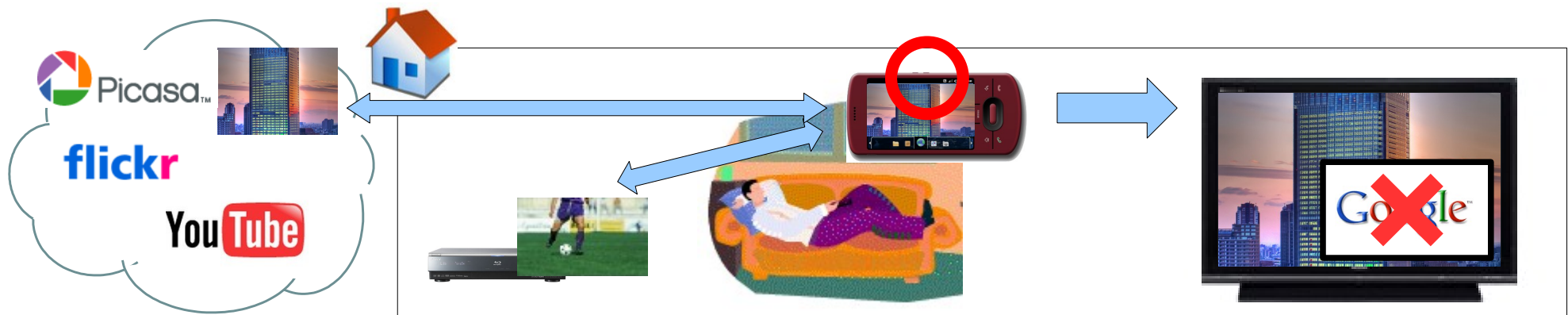
## 1. Compatibility with Home Networked Devices

User of Android devices can use their home networked devices with the Android devices seamlessly, because the application of Android emulates the standard protocol in home networking such as DLNA and NDT. To divert the standard networked devices, **Google doesn't need to develop other home electronics such as television directly.** The User can see the Internet services across the Android devices by practiced hand of their home networked device easily.



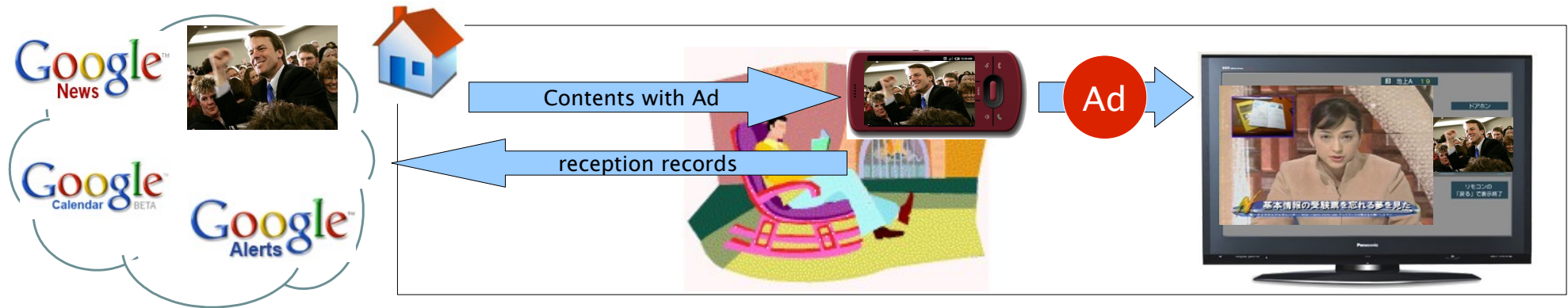
## 2. Hardware Independent

It is hard to implement the service of Internet on networked devices in home such as television directly because the embedded devices has any limitations and it is difficult to upgrade for the new services of Internet. **To use the Android devices as the proxy, it is easy to implement and upgrade for the new services.**



## 3. Push Contents and Optimized Advertisement

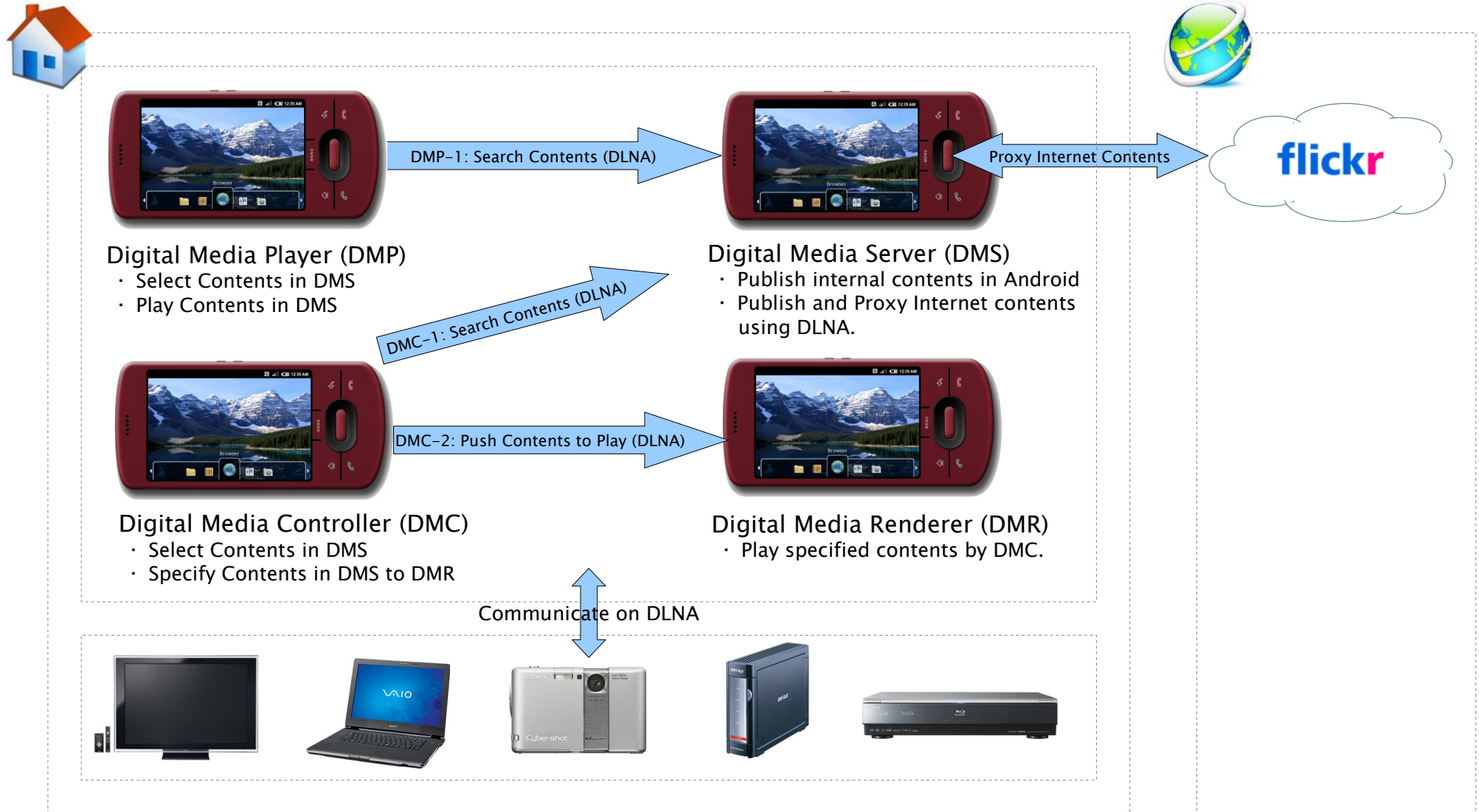
Using alert functions of the standard protocols in home networking such as DLNA v1.5 and NDT v2.1, Google can send any notifications of Internet services and advertisement to the users directly. To collect the user's behaviors such as watched contents in home, Google can send the optimized advertisement more.





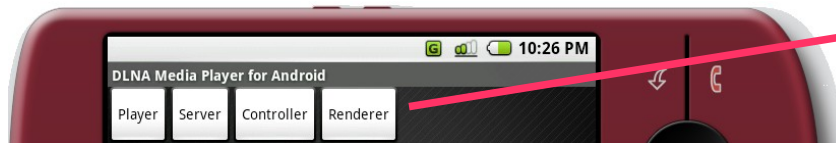
# Current Implementation for ADC

I have implemented only my first use case, Watch Internet contents on Television, for the contest now. To explain easily, I create the following four applications as a applications, but the applications connect with the other applications using networking such as DLNA.



Currently, Android emulator has no local interface yet. However, you can see the Internet contents using other DLNA hardwares across the Android's application with no the hardware changes when Android supports the local interface. Because the applications are based on DLNA and emulate the protocol to play Internet contents.

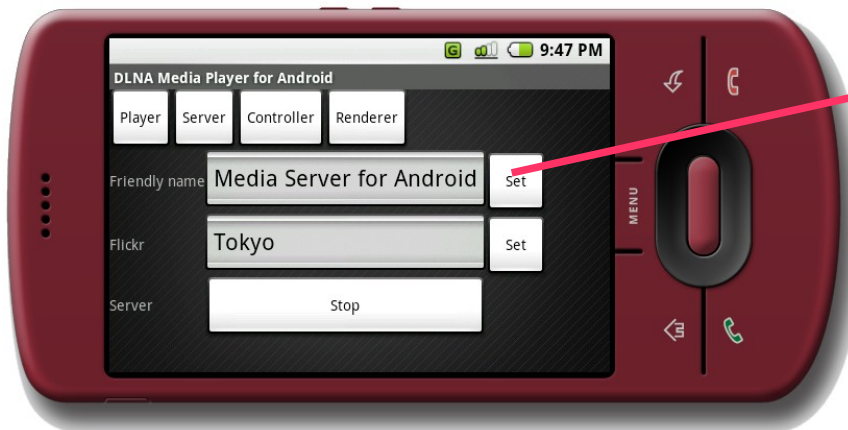
## Tab Buttons



Use the four buttons to select the applications DMP, DMS, DMC and DMR.

## Server – Digital Media Server (DMS)

DMS is implemented as a standard media server of DLNA to public the internal contents such as photo images in the SD card, but it is extended to public Internet contents, Flickr, seamlessly. The DMS runs as the HTTP proxy to transfer from the Internet contents to home devices.



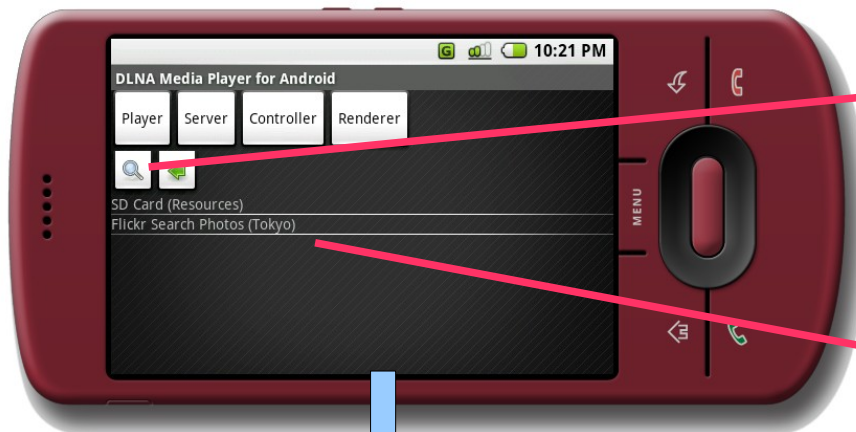
Set a friendly name for DMS, and restart.

Set a keyword to search photo images of Flickr, and restart.

Start or stop DMS.

## Player – Digital Media Player (DMP)

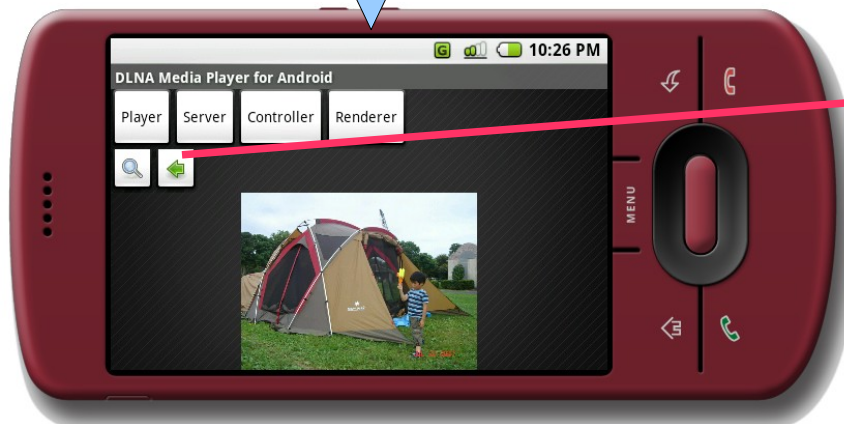
DMP is implemented as a standard media player of DLNA. I have implemented using VideoView to play the movie contents, but the function is not run normally yet.



Search media servers in home network.

Content list view. Display the child nodes when the selected item is the container node or the media server. Display the contents when the selected item is the item node.

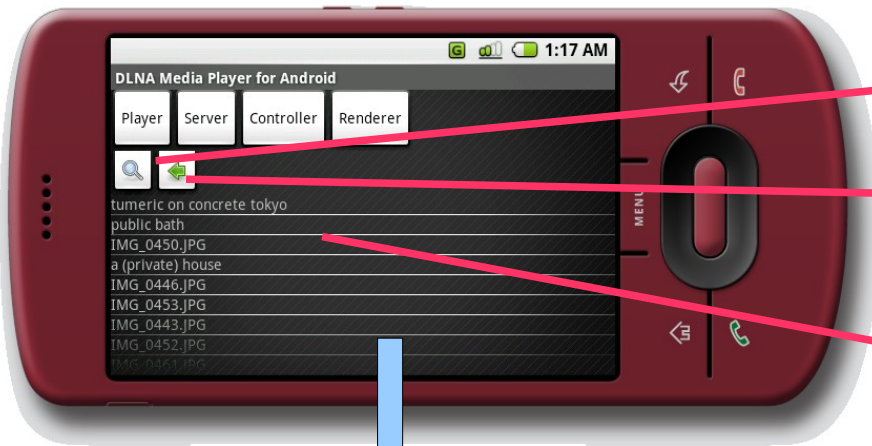
Select Item Node



Return the parent content list view.

## Controller – Digital Media Controller (DMC)

DMC is implemented as a standard media controller of DLNA. Basically, the functions of the DMC is same as DMP, but DMC sends the selected contents to DMR without displaying. It is a core function to implement my push type applications.



Search media servers in home network.

Return the parent content list view.

Content list view. Display the child nodes when the selected item is the container node or the media server. Display the contents when the selected item is the item node.

## Renderer – Digital Media Renderer (DMR)



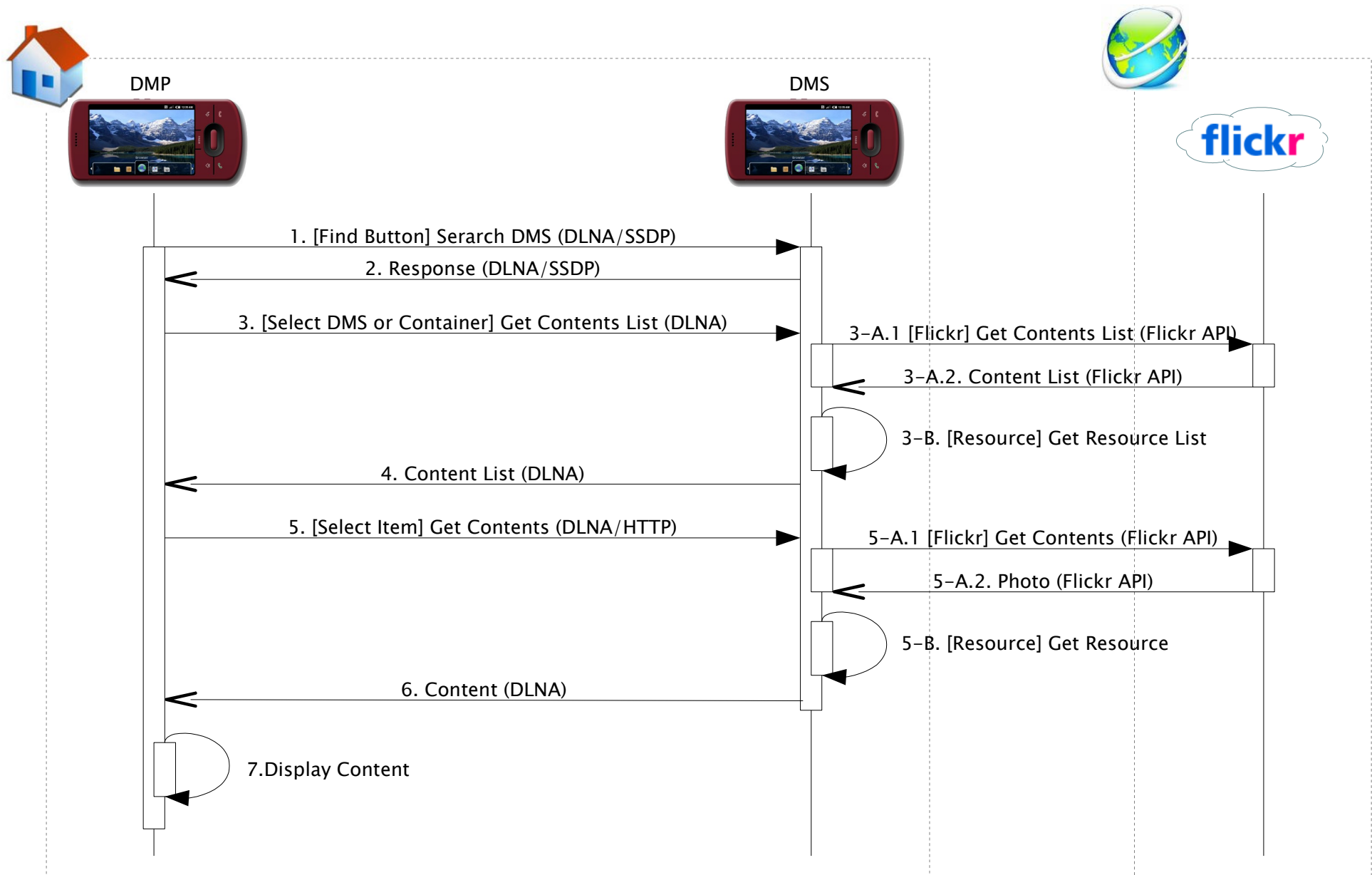
DMR is implemented as a standard media renderer of DLNA. Finally, I will not add the application to my integrated application because I will use the other standard DLNA device such as television for DMR.

Replay the current content (Not Implemented yet)

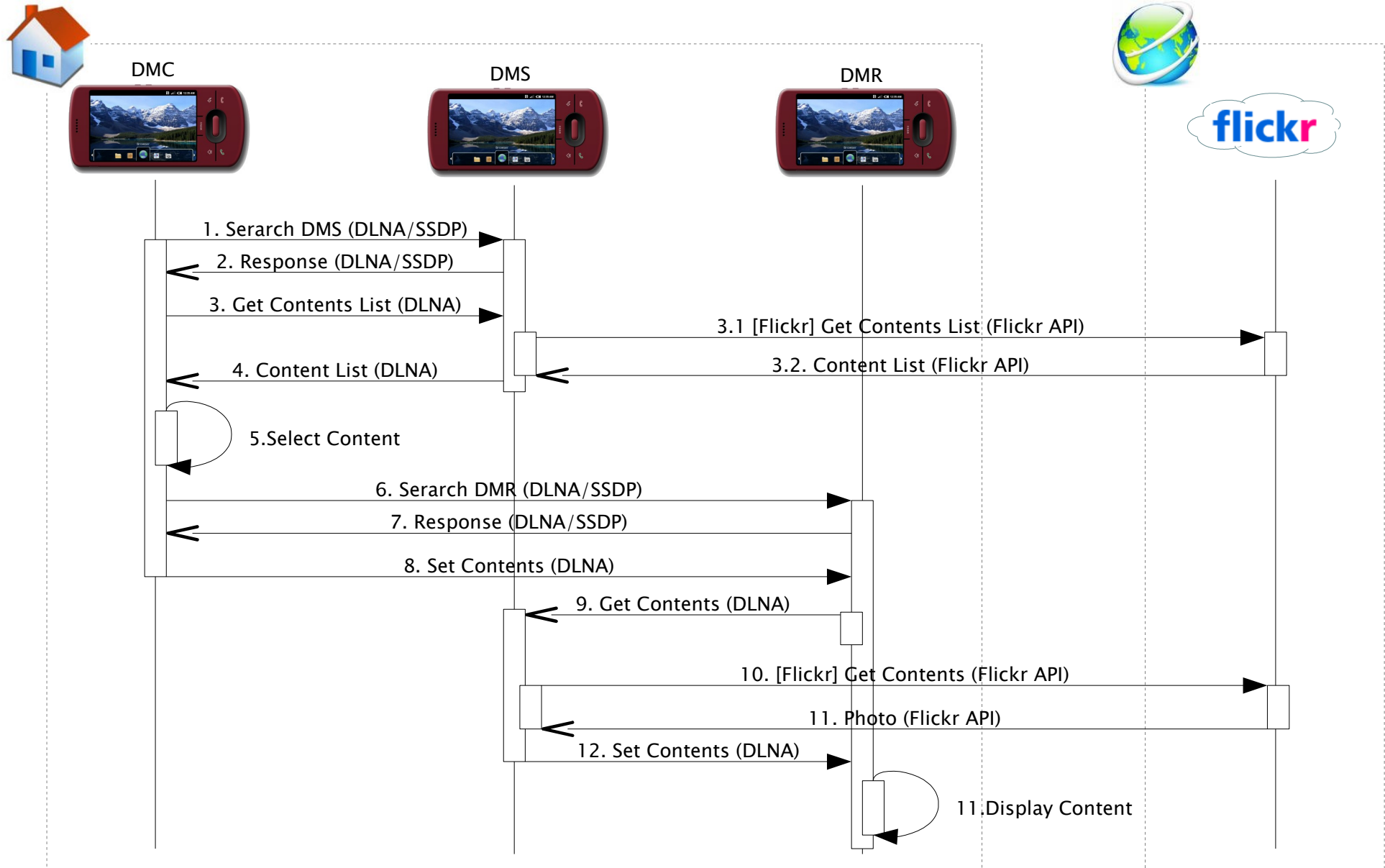
Stop the current content (Not Implemented yet)

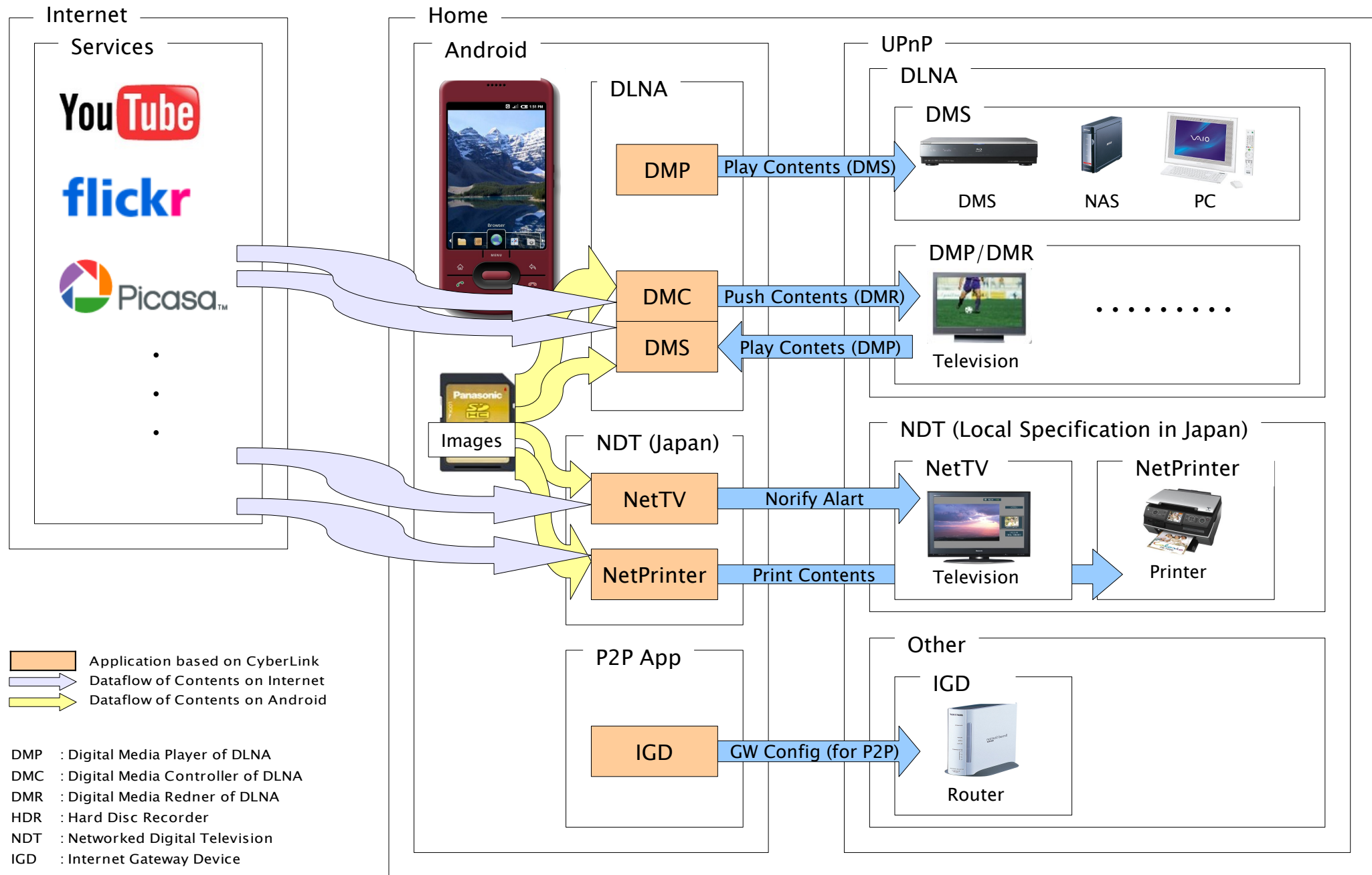


I have implemented a standard DMP and a extended DMS for Android. The DMS is extended to proxy an Internet Service, Flickr. You can see the Internet contents of the DMS using other standard DMPs such as Sony's BRAVIA and Toshiba's REGZA.



I have implemented a standard DMC and DMR to implement the push type service in the next release. I implemented the DMR too, but you can send any contents to other standard DMRs such as Sony's BRAVIA and Toshiba's REGZA using the DMC.





# Development Plan for Next Release

To implement my use case application, I have the following proposal for Android.

## 1. Android SDK for M3

To implement my applications on Android, the following functions was lacked.

No	Section	Function
1	Network	Not support multicast such as <code>java.net.MulticastSocket</code>
2		<code>Socket::getLocalAddress()</code> returns loopback address, <code>127.0.0.1</code> , instead of the bound address
3	XML	Using JAXP, <code>org.w3c.dom.Node::getAttributes()</code> returns null always.

## 2. Android SDK for M5 (Emulator)

It seems that the current SDK has sufficient functions, but the emulator has the following limitations to implement my applications on Android.

No	Section	Function
1	Network	Support multiple interface to connect local area network of home network.
2		Firewall setting to connect servers in Android from other devices in home network.

## 3. Android Product

To implement my applications on Android product, the following functions are needed.

No	Section	Function
1	Networking	Support WiFi or LAN port hardware.



According to my use case plan, I will develop the following functions to release as the final applications after the contest.

I will release the following source codes using BSD license as open source product.

No	Section	Class	Task	1	2	3	4	5	6	7	8	9	10	Current *3)	Release *3)
1	UPnP	Android	Porting											16929	17000
2	DLNA	General	UPnP AV											11284	16000
3			Detail *1)												
4		Android	Application (DMP/DMS/DMC)											3561	12000
5			Extension of Internet Service												
6			Production Test												
7	DTV	General	Porting *2)											-	4000
8		Android	Application (DTV v2.1)											-	8000
9			Extension of Internet Service												
10			Production Test												

\*1) Compliance specification of DLNA, \*2) The protocol stack is implemented using CyberLink for C, \*3) Number of source code lines.

I will purchase the following specifications and products to release the my final product if I win the contest. It is hard to develop the embedded devices by personal because of the cost to get the specifications and verification products.

No	Section	Class	Maker	Product	URL	Cost (US\$)
1	DLNA	Specification	-	-	<a href="http://www.dlna.org">http://www.dlna.org</a>	500
2		TV	Sony	BRAVIA X5000	<a href="http://www.sony.jp/products/Consumer/bravia/">http://www.sony.jp/products/Consumer/bravia/</a>	2500
3		HDR	Toshiba	VARDIA RD-S601	<a href="http://www3.toshiba.co.jp/hdd-dvd/">http://www3.toshiba.co.jp/hdd-dvd/</a>	800
4		NAS	Buffalo	LS-H250GL	<a href="http://buffalo.jp/products/catalog/storage/hd_lan.html">http://buffalo.jp/products/catalog/storage/hd_lan.html</a>	300
5		Camera	Sony	DSC-G1	<a href="http://www.sony.jp/products/Consumer/DSC/DSC-G1/">http://www.sony.jp/products/Consumer/DSC/DSC-G1/</a>	600
6		Printer	EPSON	PM-T960	<a href="http://www.epson.jp/products/colorio/printer_multi/">http://www.epson.jp/products/colorio/printer_multi/</a>	400
7	DTV	Specification	-	-	<a href="http://nw-dtv.jp">http://nw-dtv.jp</a>	500
8		TV	Panasonic	VIERA PZ750SK	<a href="http://panasonic.jp/viera/">http://panasonic.jp/viera/</a>	2500
9		Entry Phone	Panasonic	VL-SP880	<a href="http://panasonic.jp/door/">http://panasonic.jp/door/</a>	300
10		Printer	EPSON	PMT-960	<a href="http://www.epson.jp/products/colorio/printer_multi/">http://www.epson.jp/products/colorio/printer_multi/</a>	400