



Come funzionano e come si sfruttano gli standard che permettono ai dispositivi di configurarsi in modo invisibile all'utente finale.

Computer e TV si parlano, con tecnologie aperte

Di questi tempi, almeno secondo fiere di settore e spot pubblicitari, ogni angolo delle nostre case dovrebbe essere un Media Center. Qualsiasi film, foto, brano musicale o gioco digitale accessibile da un *qualunque* dispositivo elettronico in casa dovrebbe essere ugualmente disponibile, o controllabile, da qualsiasi altro prodotto dello stesso genere, nella stessa casa. A prima vista tutto ciò potrebbe sembrare solo l'ennesimo modello di condivisione di file P2P, ovvero da pari a pari senza un server centrale. In realtà, le differenze fra quanto potrebbe avvenire oggi in qualsiasi "salotto 2.0" e la condivisione di file fra *computer*, tecnicamente possibile a costi molto contenuti da almeno vent'anni non sono proprio trascurabili.

Se l'Internet delle Cose è ancora di là da venire, oggi il mercato, e soprattutto le abitudini delle persone, sono già cambiati a tal punto che non è affatto improbabile trovarsi in casa più dispositivi assai diversi ma capaci di collaborare *direttamente*, senza che nemmeno uno di loro sia un computer tradizionale. Scenari del genere sono l'obiettivo e la ragione di vita della Digital Living Network Alliance

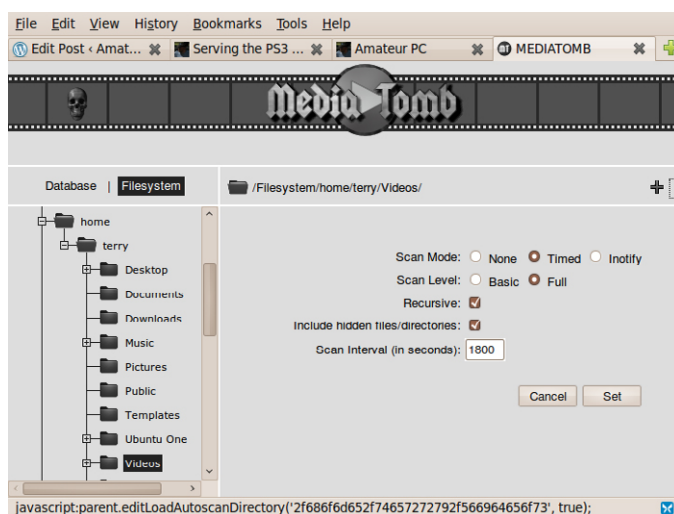
(Dlna, www.dlna.org): un consorzio di fornitori e distributori di contenuti e periferiche per home entertainment, creato da Sony e Intel una dozzina d'anni fa.

L'Alleanza promuove sviluppo e uso di reti domestiche di computer, console per giochi, terminali mobili e altri dispositivi consumer capaci di scambiarsi audio, video e fotografie con la massima facilità possibile. Questo da un lato crea nuove possibilità di intrattenimento (ma anche di studio o lavoro da casa), dall'altro nuovi rischi. La conoscenza di certi standard aperti e l'uso di software Open Source

possono aumentare le prime e ridurre i secondi, ed è proprio di questo che parleremo questo mese. Protocolli e problemi generali sono descritti nei paragrafi che seguono, mentre l'altro articolo della rubrica presenta alcuni programmi Open Source capaci di lavorare come abbiamo accennato.

UPNP, O IL SALOTTO CHE DIVENTA COME IL WEB

Fin dai suoi inizi, Dlna ha pubblicato linee guida e gruppi di standard su come garantire l'effettiva interoperabilità di computer, Smart TV, console e simili. Quello più rilevante, o almeno il più rilevante in questa occasione, si chiama Universal Plug and Play (UPnP, <http://upnp.org>). Nel mondo UPnP la prima distinzione fra dispositivi hardware, l'unica che conta



Mediatomb, un server UPnP controllabile da qualsiasi browser, ha un'interfaccia spartana e piuttosto datata, ma fornisce anche servizi opzionali, come la scansione automatica a intervalli regolari dei vari archivi audio/video.

davvero in quel contesto, è in base alla loro funzione. Gli oggetti (o i portali su Internet) che conservano fisicamente i vari contenuti multimediali, per trasmetterli a richiesta a chiunque lo chieda, sono *media server*. I *media renderer* sono altoparlanti, proiettori, Smart Tv, console e loro periferiche: in altre parole, qualsiasi prodotto che effettivamente riproduce per i suoi utenti i vari contenuti, o parte di essi, ricevuti dai *media server*. I controllori o *control point* UPnP, infine, sono i telecomandi di tutto l'ambiente. L'esempio più comune di questa categoria potrebbero essere gli smartphone e tablet con cui i vari membri della famiglia visualizzano su un loro *televisore* foto o video conservati nel *computer* che sta in un'altra stanza. Nulla impedisce a uno stesso prodotto hardware di svolgere contemporaneamente più di una di queste funzioni, anzi. Qualsiasi Smart Tv, ad esempio, le offre sempre tutte e tre: dal punto di vista UPnP il suo schermo è un *media renderer*, il sintonizzatore o decoder interno un *media server* e il telecomando un *control point*. Come fa UPnP a far accadere tutto questo? Una risposta potrebbe essere che un oggetto UPnP può comportarsi come un normale sito Web, con in più la capacità di spiegare tutte le operazioni di cui è capace e come richiederle, a qualunque altro oggetto dello stesso tipo che sia disposto ad ascoltare. Senza richiedere alcuna competenza o procedura di configurazione specifica all'utente finale. Non c'è nemmeno bisogno che almeno uno dei dispositivi coinvolti sia un personal computer vero e proprio. È grazie

**UPnP
è ovunque**
Media server, smart TV, telefoni, desktop e tablet utilizzano questo standard per la condivisione video

a queste caratteristiche che UPnP semplifica moltissimo la creazione di reti domestiche, o aziendali, il cui scopo primario è la condivisione *locale* di contenuti multimediali.

Il bello di UPnP è che tutto questo avviene combinando vari standard aperti, in larga parte già esistenti, e tutti basati su Internet. Librerie e altre parti software del sistema possono essere scritte in qualsiasi linguaggio di programmazione, per qualsiasi sistema operativo e senza alcun componente obbligatorio. Qualsiasi combinazione di normali pagine Web, script da riga di comando o interfacce a finestre native può funzionare come interfaccia, se utilizza i protocolli giusti.

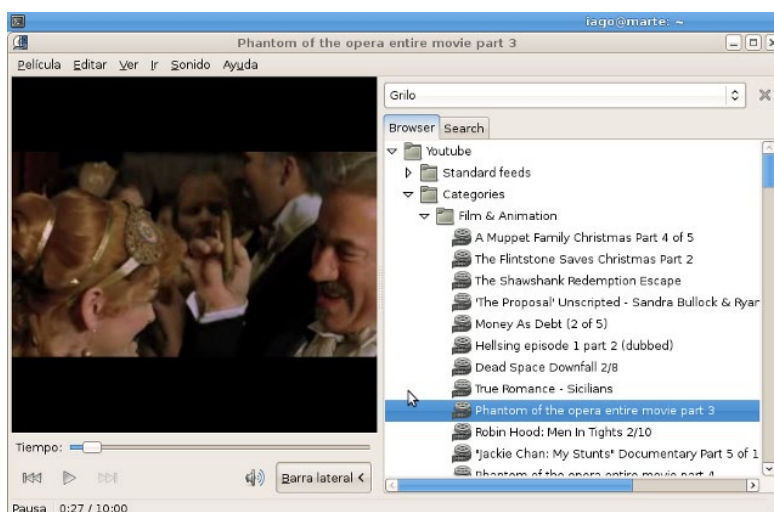
La stessa flessibilità c'è a livello fisico: condivisione e controllo via UPnP possono avvenire su qualsiasi rete cablata o wireless, quindi non servono driver hardware. Le comunicazioni avvengono comunque scambiando gli stessi pacchetti Ip (Internet Protocol) di cui è già costituito il traffico di Internet.

Oltre a questo ogni dispositivo compatibile UPnP, qualunque origine o funzione abbia, può entrare e uscire da una rete locale automaticamente, senza causare alcun problema a sé stesso o agli altri, e senza mai lasciare in sospeso le comunicazioni di controllo. Tutto questo avviene grazie a procedure ben definite di autoconfigurazione, scoperta e annuncio. Per cominciare, ogni periferica UPnP deve contenere software che si comporti come fanno i normali computer, tablet e smartphone appena si connettono a una rete locale: richiedere ai router, tramite il protocollo Dhcp, un

È TORNATO SOAP

Il Simple Object Access Protocol (Soap, www.w3schools.com/webservices/ws_soap_intro.asp) è uno standard per lo scambio di messaggi tra componenti software, formattato secondo la sintassi XML, che sta vivendo una nuova stagione di popolarità proprio grazie al suo uso nei "salotti intelligenti". Normalmente la parte che invia la richiesta iniziale è chiamata client, e quella che la esegue Web Service. La parte "object" della definizione indica che richieste e risposte seguono le regole classiche della programmazione orientata agli oggetti. Soap venne sviluppato nei primi anni duemila per portare su Web le chiamate di procedure software remote, cioè le richieste a un programma, da parte di un altro che gira su un altro computer, di svolgere qualche operazione per conto di quest'ultimo. Questo era possibile anche prima, ovviamente, ma non sempre praticabile attraverso Internet. I messaggi Soap, che sono scritti in Xml, vengono trasmessi con i protocolli Http o Smtip, cioè viaggiano sugli stessi canali normalmente utilizzati da pagine Web e, rispettivamente, posta elettronica.

I plugin del progetto Grilo possono aggiungere supporto UPnP a player Open Source di qualsiasi tipo, sia per video locali sia, come nell'esempio, per YouTube e altri archivi online.



indirizzo numerico Ip, e se disponibile anche un nome di dominio (per esempio "proiettore.sala-riunioni.ufficio"), per usarli come identificatori su quella stessa rete. La differenza fra gli oggetti UPnP e i computer tradizionali è che i primi, se non ricevono istruzioni del genere, si assegneranno sempre un indirizzo Ip da soli, ovviamente diverso da quelli che rilevano analizzando il traffico in rete, e inizieranno ad annunciare la propria presenza e capacità.

Grazie a queste procedure di "auto-promozione" server e renderer UPnP comunicano la propria disponibilità a tutti i *control point* che potrebbero essere sulla stessa rete. Simmetricamente i controllori si mettono a disposizione, in ascolto, per scoprire quali client potrebbero controllare, e per far cosa. In entrambi i casi si usano messaggi semplicissimi, che contengono solo alcune caratteristiche essenziali del dispositivo trasmittente, come un codice di identificazione e un puntatore a informazioni più dettagliate, in un formato chiamato SsdP (*Simple Service Discovery Protocol*).

Oltre a trovarsi l'un l'altro, senza aiuti da terze parti, i componenti UPnP devono conoscere esattamente tutte le proprie capacità. I *control point*, in particolare, devono saper imparare da soli tutte le caratteristiche dei server o renderer che vogliono controllare, altrimenti come potrebbero farlo senza coinvolgere pesantemente l'utente? Questo apprendimento avviene con lo stesso meccanismo con cui un utente di Internet, umano o motore di ricerca, scoprirebbe di cosa parla un



Un media server UPnP perfettamente funzionante, in pochi centimetri di spazio, con software e scheda madre Open Source (vedi Box Risorse).

sito Web: un *control point* deve infatti visitare la "home page" dell'oggetto da controllare.

Il puntatore menzionato nei paragrafi precedenti, che deve essere presente in ogni messaggio di annuncio UPnP, non è altro infatti che un normale Url, leggibile anche con qualsiasi browser connesso alla stessa rete locale. Caricando quell'Url si otterrà, direttamente dal dispositivo che interessa, un file di testo nell'immane formato Xml. Il suo contenuto è un manuale di istruzioni completo del dispositivo stesso, ma formattato in modo che qualsiasi software con le librerie giuste possa capirlo. Al suo interno si trovano innanzitutto dati banali, ma utilissimi, come nome e numero di serie della periferica corrispondente, e link al servizio cliente o a manuali online.

La parte più importante di quel documento è però la sua lista di comandi e

servizi. Tutte le azioni di cui il dispositivo è capace sono elencate, divise appunto per servizio. A ogni azione è allegato un elenco completo di tutte le opzioni possibili. Se necessario, la "home page" contiene anche una descrizione di tutte le variabili di stato che sarebbe possibile leggere o impostare nel dispositivo. Da quel momento in poi, ogni richiesta di esecuzione comandi da un *control point* a un server o *renderer* riceverà in risposta il risultato di quei comandi e il nuovo stato del dispositivo, se necessario.

Riassumendo, ogni prodotto degno dell'etichetta UPnP è in grado di insegnare a qualsiasi altro oggetto che parli quel linguaggio, chiunque ne sia il fabbricante, come controllarlo direttamente.

Per fare in modo che una rete UPnP funzioni davvero manca però ancora una tessera del puzzle, ovvero la notifica degli eventi ("eventing" nella



RISORSE

Le specifiche ufficiali e complete di tutte le parti dell'ambiente UPnP si trovano, ovviamente, su www.UPnP.org, ma non è quello il posto migliore per iniziare a conoscere questa famiglia di standard. Consigliamo invece, almeno per chi volesse anche imparare a programmare per applicazioni UPnP, le guide corrispondenti nel sito per sviluppatori di Microsoft (<https://msdn.microsoft.com/en-us/library/ms899563.aspx>). La presentazione online all'indirizzo <http://si.deis.unical.it/~talia/aa0506/dis/es6-2p.pdf> è una introduzione a Soap in Italiano e relativamente semplice.

Chi invece volesse semplicemente trasformare un generico computer Linux in un media server UPnP potrebbe prendere come riferimenti iniziali le procedure descritte nei due articoli di Linux Magazine (www.linux-magazine.com/Online/Blogs/Productivity-Sauce/Set-up-a-DLNA-Server-in-a-Minute) e SuperUser.com (<http://superuser.com/questions/628813/UPnP-dlna-render-on-linux>). Una configurazione dettagliata di un Raspberry Pi come micro-server UPnP si trova invece su <http://blog.scphillips.com/2013/07/playing-music-on-a-raspberry-pi-using-UPnP-and-dlna-revisited/>. Un altro progetto dello stesso tipo, il cui risultato è visibile in Figura 3, è descritto nel forum <http://yabb.jrriver.com/interact/index.php?topic=89044.0>.

documentazione UPnP). I *control point* su una stessa rete possono essere anche parecchi (per esempio, tutti gli smartphone di una famiglia o gruppo di amici in grado di “vedere” la stessa TV in salotto). In condizioni del genere, se tutti i controllori verificassero continuamente lo stato di ogni dispositivo genererebbero parecchio traffico inutile che potrebbe addirittura causare problemi nella trasmissione audio o video.

Per evitare queste eventualità il protocollo UPnP prevede che ogni *server* o *renderer* invii una notifica, a tutti e soli i controllori che ne hanno fatto richiesta, ogni volta che ha terminato l'esecuzione di un comando, o che una delle sue variabili di stato cambia per qualunque motivo. In questo modo i singoli *control point* dovranno soltanto preoccuparsi di “abbonarsi” ai dispositivi che gli interessano, o meglio a quelli che i loro utenti vorranno gestire in ogni momento.

I tipi di oggetti gestibili, i servizi disponibili con ognuno di loro e il modo di gestirli sono specificati in appositi protocolli di controllo chiamati Dcp (“Device Control Protocol”). A titolo di esempio, in ambito audio/video esistono le due categorie di oggetti già citati (*media renderer* e *media server*), che sono capaci di offrire in totale quattro servizi: gestione delle connessioni, controllo del rendering, elenco dei contenuti disponibili e quello chiamato “AVTransport”. Quest'ultimo e opzionale servizio permette di controllare in maniera molto più fine quali contenuti vengono inviati a chi, e come: un esempio potrebbe essere l'ordine di

inviare a uno e uno solo degli schermi disponibili una specifica scena di un video registrato in precedenza e salvato su un disco di rete.

Altri componenti e servizi che aumentano la flessibilità di UPnP sono, citandone solo due fra tanti per ragioni di spazio, quelli chiamati Web4CE e QoS. Il primo è un metodo, basato sulla variante del linguaggio Html chiamata Ce-HTML (<http://en.wikipedia.org/wiki/CE-HTML>), per controllare periferiche UPnP direttamente da un browser, anziché da software o telecomandi hardware dedicati.

La Qualità di Servizio (QoS) è invece un concetto molto più generale, fondamentale in molti campi delle telecomunicazioni. Sulle reti digitali, incluse quelle UPnP, viaggiano flussi di traffico con caratteristiche ed esigenze diverse fra loro. Un comando di cambiare canale, per esempio, potrebbe anche arrivare con ritardi “enormi” come mezzo secondo, senza causare alcun problema agli utenti, purché arrivi perfettamente integro. Nella visione di un film, invece, la perdita di un fotogramma causerebbe molto meno fastidio di un ritardo fra fotogramma e fotogramma continuamente variabile di più di *pochi millisecondi*. I comandi QoS servono proprio a prevenire questi problemi, specificando quali flussi dati devono avere priorità sugli altri, e qual è il massimo ritardo con cui devono attraversare la rete.

UPNP SU MISURA? SÌ, CON SOFTWARE E HARDWARE OPEN SOURCE

I Media Center Open Source descritti in queste pagine consentono già da anni di usufruire con poca fatica dei servizi UPnP da Linux. Come sempre però, il bello e il valore maggiore del software Open Source rimane il fatto che, avendo le competenze giuste, ci si può costruire una soluzione perfettamente su misura.

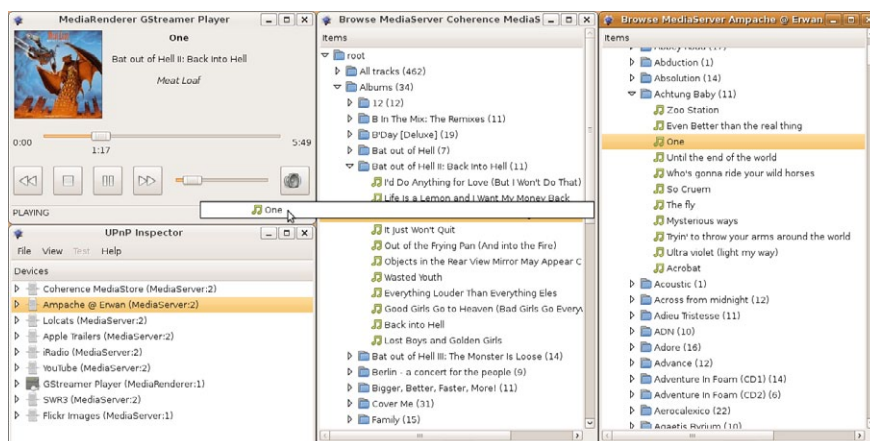
Questi tipi di progetti, oltre ad avere un interesse innegabile per gli hobbisti “digitali”, possono avere anche un valore notevole come strumenti didattici, per avvicinare alla programmazione anche studenti altrimenti difficili da raggiungere.

Il primo strumento per attività del genere su Linux è probabilmente MiniUPnP (<http://miniUPnP.free.fr>), che è composto da due parti principali: la prima è una libreria per *control point*, che una volta compilata occupa meno di 50 KByte di spazio in memoria. Il codice sorgente è Ansi C, quindi portabile.

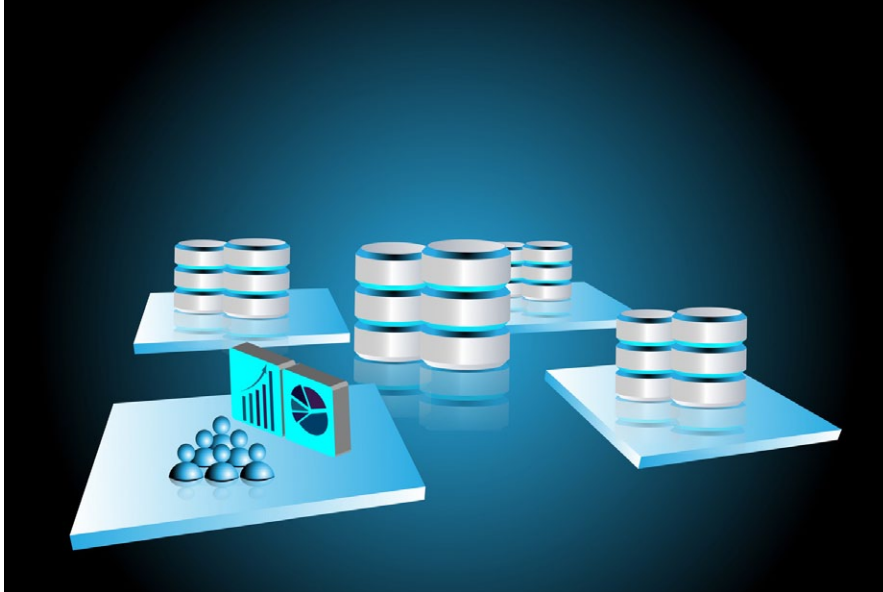
L'altro componente base di MiniUPnP è il server, chiamato miniUPnPd. Come la sua controparte ha un ingombro di memoria ridotto, ed è quindi utile anche per applicazioni *embedded*. È utilizzando codice come questo che alcuni hacker riescono, già da anni, a costruirsi Media Center miniaturizzati come quelli basati su Raspberry Pi, il popolare microcomputer (vedi il box Risorse). A un livello diverso da MiniUPnP e meno impegnativo, c'è la libreria chiamata Grilo (<https://wiki.gnome.org/Projects/Grilo>): grazie ad essa si possono scrivere facilmente plugin per collegare a server UPnP anche player per desktop Linux come Totem o RhythmBox. Dopo aver installato e configurato Grilo, qualsiasi applicazione con i relativi plugin, che giri sullo stesso computer, potrà accedere a i server UPnP della rete locale, qualsiasi sia la loro natura. Particolarmente interessante è il fatto che i plugin di Grilo possono connettersi anche a “servizi” che non sono affatto lettori Dvd e altre periferiche, ma tradizionali portali Web come YouTube, Jamendo, Last.FM o Flickr.

Notifiche invisibili

I dispositivi in rete devono poter comunicare in modo efficiente, senza sovraccaricare il network locale



Sulla vostra rete locale potrebbero già esserci programmi che supportano UPnP, magari a vostra insaputa e con una configurazione non corretta. Potete scoprirlo con UPnP Inspector.



Con le distribuzioni Linux più popolari è facile gestire i media center ricorrendo, se necessario, a collaudati plug-in scaricabili dalla Rete.

Linux e standard Upnp, non solo media center

Il supporto Linux per i sistemi Upnp, grazie alla natura aperta dei vari standard coinvolti, è abbastanza buono da essere, in un certo senso, difficile da scoprire. I plugin come Grilo permettono a praticamente tutti i software Open Source per media center tradizionali, come Xbmc (<http://xbmc.org>) e a parecchi media player come RythmBox (<https://wiki.gnome.org/Apps/Rhythmbox>) di comunicare senza particolari problemi con applicazioni di questo tipo. Nella maggior parte dei casi, soprattutto se si utilizzano distribuzioni popolari come Ubuntu, Debian o Fedora, per servirsi di Upnp basterà installare qualche pacchetto extra, magari da archivi online

diversi da quelli normalmente usati. Diverso è il caso per chi volesse metter su un media center software *soprattutto* per applicazioni Upnp, oppure controllare e gestire in maniera più efficiente i vari dispositivi Upnp della propria rete domestica. Ecco alcuni programmi per Linux pensati proprio per questi casi.

GmediaRender, Mediatomb e Rygel

Questi programmi sono tutti (almeno) *media server*. Il primo, la cui home page è <http://gmrender.nongnu.org>

org è anche un *renderer* molto spartano, comunque dotato di tutte le funzioni base. Lo citiamo soprattutto perché è utilizzato come "motore" Upnp di vari altri progetti Open Source in ambito media center / multimedia. Mediatomb (<http://mediatomb.cc>), invece, è solo un server, capace di lavorare con GmediaRender o qualsiasi altro *renderer*.

L'interfaccia controllabile da browser potrebbe intimidire chi è abituato a quella dei media center commerciali da salotto, ma è molto semplice da usare, dopo l'installazione e configurazione iniziali, e anche abbastanza completa. È possibile sia visualizzare velocemente le anteprime contenute nelle foto digitali, sia leggere i metadati dei file musicali come genere musicale, oppure nome dell'autore o dell'album.

Software gratuiti

Sono numerosi i programmi disponibili per la gestione di foto, video e musica.

UN PROCESSO IMPORTANTE PER IL FUTURO DELL'OPEN SOURCE

Le violazioni, anche involontarie, delle licenze software Open Source non sono certo una novità, ma negli ultimi mesi c'è stata una novità importante nel modo in cui vengono combattute. Questa volta, invece della Free Software Foundation o di altre organizzazioni non-profit dello stesso genere, è un'azienda privata a essere scesa in campo per bloccare usi impropri del suo software libero. Ximpleware è una software house che produce, fra le altre cose un parser (analizzatore) di codice Xml, disponibile sia con licenza Gpl 2 sia con una commerciale. Nei mesi scorsi Ximpleware ha citato per danni non solo altre due aziende per aver usato il suo software con licenza Gpl, ma senza rispettarne tutti i termini, ma anche altri clienti della prima. In breve Versata, un'altra software house, ha utilizzato il parser di Ximpleware in un suo prodotto, che ha poi licenziato (fra gli altri), all'azienda di servizi finanziari Ameriprise. Quando quest'ultima ha modificato il software "senza permesso", sostenendo di poterlo fare proprio perché contenendo parti Gpl era tutto software Open Source, Versata le ha fatto causa. Solo a quel punto Ximpleware si è accorta dell'intera faccenda, a cui ha reagito come abbiamo già detto. Come finirà non è possibile dirlo adesso, ma quel che conta è proprio il fatto che stavolta sia un'azienda ad adire vie legali per mantenere e far rispettare la natura Open Source di un suo prodotto.



NEWS

Mediatomb può usare automaticamente, seguendo le indicazioni dell'utente, gli stessi metadati per riordinare e presentare i file audio in vari livelli di cartelle. In generale, è possibile visualizzare gerarchicamente sia i contenuti, sia i relativi metadati. Se un archivio contiene video o brani musicali in formati non supportati dalla configurazione standard di Mediatomb, quasi sempre basta installare uno dei suoi plugin per risolvere il problema. Il supporto per Url esterni alla rete locale consente di trattare portali come Last.FM o YouTube come se fossero cartelle sul proprio disco rigido. Essendo compilabile senza problemi anche per processori Arm, Mediatomb è infine disponibile anche per microcomputer basati su Raspberry Pi e altre schede del genere.

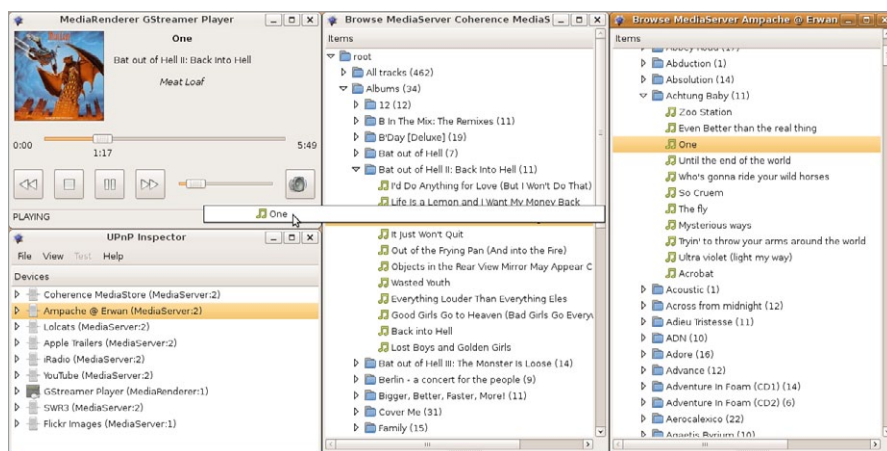
Rygel (<https://wiki.gnome.org/action/show/Projects/Rygel>), invece, è un server e renderer presente come pacchetto binario negli archivi ufficiali delle principali distribuzioni Linux. Come Mediatomb, grazie a plugin di terze parti e a una semplice interfaccia grafica permette senza troppi sforzi di riprodurre su smartphone, PlayStation 3 o normali TV, audio e video in formati altrimenti non leggibili da quelle piattaforme. Sempre da Rygel è possibile reindirizzare l'audio di un filmato a qualsiasi altoparlante esterno, ma con supporto Dlna/Upnp.

ASSISTENTI UPNP PER GLI UTENTI PIÙ ESPERTI

Come abbiamo accennato nell'articolo principale Upnp è talmente facile da usare, e talmente privo di procedure di sicurezza, che se si ha intenzione di farne uso pesante conviene dotarsi anche di strumenti per gestirlo senza perdere troppo tempo. In ambito Linux, per facilitare queste attività c'è innanzitutto l'utility djmound (<http://djmound.sourceforge.net>), che scopre tutti i media server Upnp della sua rete locale e li "monta" come se fossero cartelle di un normale filesystem Linux. Questo consente di utilizzare quei contenuti anche da player senza supporto Upnp,

ma soprattutto di farne backup periodici automaticamente, con programmi come rsync.

Di tutt'altro tipo, ma ugualmente utile, è il programma upnp-inspector (<http://coherence.beebits.net/wiki/UPnP-Inspector>), visibile nella figura sotto. Installandolo, si potranno visualizzare tutti i dispositivi Upnp presenti in rete, analizzarne lo stato e anche impartirgli dei comandi. Chiudiamo con un software più adatto a utenti che si spostano con il laptop da una rete all'altra: Upnp Port-Mapper (<http://upnp-portmapper.sourceforge.net>) aiuta a riconfigurare qualsiasi router compatibile Upnp sempre dalla stessa interfaccia grafica, anziché quella specifica di ogni modello.



**UN NUOVO
BROWSER
SI PREPARA
A ENTRARE
IN SCENA**



Un browser Open Source, derivato direttamente da Chrome, ma sviluppato direttamente da uno dei fondatori di un altro browser, per di più proprietario come Opera? Sembra strano, ma è proprio così. Al momento in cui scriviamo Vivaldi (<https://vivaldi.net>) sembra a malapena entrato nella fase beta della sua esistenza, ma si presenta già come un prodotto da tenere d'occhio nel 2015. L'interfaccia, almeno al momento, è piuttosto scarna, ma sotto il cofano sono già presenti funzioni assenti in altri browser a meno, quando esistono, di installare dei plugin. Fra queste citiamo l'"avanti veloce" (Fast Forward) che cerca di indovinare quale sarà la prossima pagina che vorremo leggere per caricarla prima ancora di un clic sul link corrispondente, e la flessibilità nella gestione delle schede: i loro titoli possono, ad esempio, apparire anche in fondo o ai lati della finestra del browser, non solo in cima.