



UNIVERSITÀ DEGLI STUDI DI FIRENZE
Facoltà di Ingegneria

Corso di Laurea Triennale in
INGEGNERIA INFORMATICA

Analisi e sviluppo di un'applicazione Social TV per Smart TV

Analysis and development of a Social TV app for Smart TV

Tesi di Laurea di

Manuele Guaita

Relatore:

Prof. Paolo Nesi

Correlatore:

PhD Ing. Pierfrancesco Bellini

Anno Accademico 2012-2013

Indice

1	Introduzione	1
1.1	Cos'è una Smart TV	1
1.2	Proprietà di un'applicazione per Smart TV	2
1.3	Confronto tra le varie Smart TV	3
1.4	ECLAP per Smart TV e l'ambiente di sviluppo Samsung	5
1.5	Altre Applicazioni su Samsung Smart TV	10
2	ECLAP overview	12
2.1	ECLAP overview	12
3	ECLAP per Smart TV: analisi dei requisiti e casi d'uso	21
3.1	Analisi dei requisiti	21
3.2	Casi d'uso	22
3.3	Layout delle schermate in fase di pre-sviluppo	28
4	Eclap per Smart TV: progettazione	32
4.1	Architettura	32
4.2	Applicazione scene-based e componenti usati	33
4.3	Struttura directory e diagrammi di sequenza	36
4.4	Comunicazione client-server: modulo Drupal	51
4.5	Database gestito da MySQL	63
5	Eclap per Smart TV: validazione e uso	66
5.1	Screenshot e loro descrizione	66
5.2	Problemi incontrati	80
6	Conclusioni	82
6.1	Conclusioni e possibili sviluppi	82
	Appendice	84

INDICE

Bibliografia

97

Elenco delle figure

1.1	<i>Componenti di un'applicazione Smart TV</i>	5
1.2	<i>Architettura della convergenza</i>	8
1.3	<i>AllShare API</i>	9
2.1	<i>Portale ECLAP</i>	12
2.2	<i>Tipi di contenuti riprodotti da ECLAP</i>	14
2.3	<i>MyStoryPlayer di un laboratorio teatrale</i>	15
2.4	<i>Grafico sociale per "you PARA DISO"</i>	16
2.5	<i>Nodi e arco del Social Graph</i>	17
2.6	<i>Funzionalità del Social Graph</i>	17
2.7	<i>Architettura a livelli di ECLAP</i>	18
2.8	<i>Utenti ECLAP raggruppati per attività principale</i>	19
3.1	<i>Diagramma dei casi d'uso ECLAP per Smart TV</i>	23
3.2	<i>Prototipo della schermata Home</i>	28
3.3	<i>Prototipo della schermata SceneContent</i>	29
3.4	<i>Prototipo della schermata di login</i>	29
3.5	<i>Prototipo della schermata delle impostazioni</i>	30
4.1	<i>Diagramma Deployment ECLAP per Smart TV</i>	33
4.2	<i>Diagramma delle classi ECLAP per Smart TV</i>	34
4.3	<i>Directory ECLAP per Smart TV</i>	37
4.4	<i>Diagramma di sequenza - Riproduzione immagine</i>	40
4.5	<i>Diagramma di sequenza - Riproduzione video</i>	41
4.6	<i>Diagramma di sequenza - Visualizzazione dei contenuti selezionando canali o categorie</i>	42
4.7	<i>Diagramma di sequenza - Visualizzazione dei metadati di un contenuto</i>	43
4.8	<i>Diagramma di sequenza - Ricerca dei contenuti tramite keyword</i>	44
4.9	<i>Diagramma di sequenza - Visualizzazione dei simili di un contenuto</i>	45
4.10	<i>Diagramma di sequenza - Accesso autenticato</i>	46

ELENCO DELLE FIGURE

4.11	<i>Diagramma di sequenza - Visualizzazione dei commenti di un contenuto</i>	47
4.12	<i>Diagramma di sequenza - Commentare un contenuto</i>	48
4.13	<i>Diagramma di sequenza - Gestione delle impostazioni</i>	49
4.14	<i>Diagramma di sequenza - Aggiungere ai preferiti un contenuto</i>	50
5.1	<i>Screenshot - Home</i>	67
5.2	<i>Screenshot - Passaggio ai canali (Prima)</i>	67
5.3	<i>Screenshot - Passaggio ai canali (Dopo)</i>	68
5.4	<i>Screenshot - Passaggio alle categorie (Prima)</i>	68
5.5	<i>Screenshot - Passaggio alle categorie (Dopo)</i>	69
5.6	<i>Screenshot - Schermata della scena Content</i>	69
5.7	<i>Screenshot - Helpbar</i>	70
5.8	<i>Screenshot - Ricerca tramite tastierino numerico</i>	71
5.10	<i>Screenshot - Ricerca con la keyword "dario fo"</i>	71
5.9	<i>Screenshot - Ricerca tramite tastiera virtuale</i>	72
5.11	<i>Screenshot - Video a schermo intero</i>	72
5.12	<i>Screenshot - Immagine a schermo intero</i>	73
5.13	<i>Screenshot - Visualizzazione dei metadati</i>	74
5.14	<i>Screenshot - Visualizzazione dei metadati (scroll bar attiva)</i>	74
5.15	<i>Screenshot - Visualizzazione dei simili di un contenuto</i>	75
5.16	<i>Screenshot - Login</i>	76
5.17	<i>Screenshot - Navigazione con accesso autenticato</i>	76
5.18	<i>Screenshot - Logout</i>	77
5.19	<i>Screenshot - Popup dei commenti</i>	78
5.20	<i>Screenshot - Commento abilitato</i>	78
5.21	<i>Screenshot - Impostazioni</i>	79
5.22	<i>Screenshot - Popup di conferma impostazioni</i>	79

Capitolo 1

Introduzione

1.1 Cos'è una Smart TV

Con il termine *Smart TV*^[1] si definisce la categoria di apparecchiature elettroniche che hanno come principale caratteristica l'integrazione di funzioni e di servizi legati a Internet e, in particolare, al cosiddetto *Web 2.0* all'interno di apparecchi televisivi.

Lo stesso termine è usato in modo più generico per definire la convergenza tecnologica tra il mondo dei personal computer e quello della televisione. Questi dispositivi, tramite il collegamento alla rete, offrono la possibilità di usufruire di servizi online tipicamente destinati ai normali personal computer, quali navigazione Internet, video on demand, Internet TV, multimedialità, servizi in streaming e social network, mantenendo o potenziando alcune caratteristiche di interattività dei televisori e dei dispositivi collegati, messi in commercio dopo la nascita della televisione digitale terrestre e satellitare. Ciò avviene grazie alla presenza di applicazioni (*App*) *ad hoc*.

La tecnologia alla base del concetto di *Smart TV*^[1] può essere integrata all'interno dell'apparecchio televisivo o può essere resa disponibile tramite *set-top box*, cioè dispositivi esterni destinati ad aumentare le funzionalità dei televisori tradizionali. L'aumento di funzionalità si ottiene grazie alla possibilità di installare applicazioni all'interno della *Smart TV*, che possono essere sviluppate dal produttore della TV, dallo stesso utilizzatore o da terze parti e rilasciate sul mercato in forma gratuita o a pagamento.

L'appellativo *smart*^[1] (traducibile dalla lingua inglese come intelligente) è stato utilizzato per questa categoria di dispositivi in analogia a quanto avvenuto nel mondo della telefonia mobile con l'introduzione degli smartphone, cioè telefoni cellulari dotati di avanzate caratteristiche multimediali e interattive. All'apparenza molto simili, forniscono agli utenti differenti esperienze, sulla base

delle caratteristiche di ogni dispositivo e dell'interfaccia utente. La *Smart TV*, a differenza dello smartphone, presenta le seguenti caratteristiche:

- l'utente può interagire con il dispositivo sia con il telecomando e sia con un pad direzionale *D-pad* a cui viene associato un cursore
- la dimensione dello schermo, che permette all'utente di interagire anche a distanza maggiore rispetto a uno smartphone, è un aspetto da non sottovalutare quando si sviluppa un'applicazione in quanto si riflette nella scelta dei colori, nella dimensione dei caratteri e nel focus.

I tre fattori che hanno influenzato l'evoluzione della televisione e dell'entertainment negli ultimi anni sono, in primis, l'alta definizione, seguita dal 3D ed infine dalle TV interattive e connesse alla rete, dette *Smart TV*.

Ormai tutti i maggiori produttori di TV hanno progettato e prodotto il proprio concetto di TV connessa alla rete, tra questi *Samsung* è l'azienda che maggiormente punta su questo settore, infatti è stata una delle prime ad aver prodotto una TV in grado di riconoscere l'utente (grazie ad un sistema di riconoscimento biometrico attivato tramite la fotocamera integrata nei suoi top di gamma) e di poter essere controllata con la voce e con dei semplici gesti.

L'utente medio è convinto che possedere una TV connessa gli consentirà esclusivamente di navigare sul web, intrattenersi con contenuti on demand oppure utilizzare le applicazioni studiate appositamente per la televisione. In realtà utilizzare una *Smart TV* in un ecosistema domestico apre le porte ad un concetto molto più vasto: sarà possibile vedere cosa succede in casa grazie ad uno smartphone o un tablet, sfruttando la webcam integrata nella TV. Lo stesso dispositivo mobile, può ad esempio trasformarsi in un ulteriore schermo televisivo grazie alla modalità "Second View" che consente, appunto, di vedere lo stesso contenuto riprodotto dalla TV, oppure visualizzare contemporaneamente un canale differente.

Insomma, immaginando il futuro, tanto gli utenti evoluti quanto quelli moderati si spingono sempre più verso un concetto di interconnessione tra l'interno e l'esterno della casa, attraverso un ecosistema capace di immagazzinare contenuti anche sul cloud: uno strumento unico, che funzioni da centro dell'intrattenimento domestico semplificando la fruizione di contenuti, on demand o provenienti da diversi dispositivi, rendendola interattiva, rilassante e piacevole.

1.2 Proprietà di un'applicazione per Smart TV

Per lo sviluppo di un'applicazione per *Smart TV* bisogna prendere in considerazione una serie di accorgimenti al fine di ottenere un buon prodotto e un alto gradimento da parte dell'utente.

CAPITOLO 1. Introduzione

In particolare, lo schermo della TV non dovrebbe essere coperto da troppe funzioni o troppe informazioni, il *layout* dovrebbe essere *user-friendly*.

La navigazione all'interno dell'applicazione dovrebbe essere abbastanza semplice e non richiedere l'uso di un manuale di riferimento ed essere chiara e precisa. Per far ciò l'uso dell'applicazione deve essere intuitivo, bisogna tener conto del dispositivo di input utilizzato per controllare l'applicazione, inoltre il riutilizzo di funzioni, dove possibile, rende più facile il suo apprendimento e il suo utilizzo.

Quando il tempo necessario, per aprire un'applicazione o per elaborare delle richieste, supera un certo periodo, è necessario che venga visualizzata un'animazione di caricamento che indica all'utente il cambiamento, a breve, della schermata secondo la sua richiesta.

Visualizzare un'applicazione sulla schermo TV è simile alla visualizzazione di una pagina web usando un browser su un computer; la differenza sta nella risoluzione dello schermo, nelle specifiche hardware e nel controllo remoto che funziona tramite interfaccia utente.

Le *Smart TV* hanno cambiato la modalità con cui l'utente si rapporta con la TV, da passiva ad attiva, per il fatto che esiste una vera e propria interazione tra utente e TV, anche se questa interazione resta limitata e meno coinvolgente rispetto all'uso degli smartphone, in quanto viene utilizzata in una relazione di relax.

1.3 Confronto tra le varie Smart TV

I principali produttori di *Smart TV* sono: *Samsung*, *Sony*, *Philips*, *LG*. Il processo di sviluppo di un'applicazione comune per tutte le *Smart TV* si può esplicitare nei seguenti passaggi:

- registrarsi su un sito dedicato, che cambia da marca a marca, per poter usufruire del materiale necessario allo sviluppo
- scaricare e installare un *SDK (Software Development Kit)*, che si integra con l'*IDE (Integrated Development Environment) Eclipse*
- in una prima fase di sviluppo, testare l'applicazione creata su un emulatore fornito dal produttore e, in seguito, su una TV
- pubblicare l'applicazione creata dopo un processo di verifica da parte del produttore.

Purtroppo non esiste uno standard unico per lo sviluppo di applicazioni per *Smart TV*, infatti, ad esempio, i due attori principali (*Samsung ed LG*) hanno due *SDK*

CAPITOLO 1. Introduzione

molto differenti, come anche *Sony*, *Philips* e altre aziende propongono un proprio *SDK* e un proprio stile di sviluppo. Inoltre, anche gli accordi per la distribuzione delle applicazioni variano da azienda ad azienda. Ad esempio *Samsung* permette a uno sviluppatore di pubblicare esclusivamente negli Stati Uniti, mentre per pubblicare nel resto del mondo è necessario diventare partner, condizione che comunque può essere rifiutata da *Samsung* nel caso in cui non ritenga valido il richiedente.

Tutte le apps per *Smart TV* di qualunque marca sono basate sullo standard aperto *CE-HTML*, che è un sottoinsieme di standard Internet aperti già esistenti come:

- *XHTML1.0*
- *CSS TV Profile 1.0*
- *DOM level 2 (Core, Style, Events, e un sottoinsieme di moduli HTML)*
- *ECMAScript 262 (3rd edition)*
- *XMLHttpRequest (AJAX)*.

CE-HTML è uno standard basato su *XHTML* per sviluppare pagine web con interfacce utente remote per dispositivi elettronici di consumo su reti *UPnP*.

Invece le varie *Smart TV* si differenziano per le *API (Application Programming Interface)* usate e il sistema su cui vengono installate e gestite le varie applicazioni, il che rende complicata un'automatica versatilità del codice.

Con la maggiore richiesta delle *Smart TV*, alcuni produttori si sono evoluti stringendo accordi fra di loro e con altre organizzazioni che lavorano in questo settore, un esempio che testimonia ciò è la *Smart TV Alliance*.

Molto interessante è anche il prodotto di *Google*, chiamato *Google TV*, nata da una collaborazione tra *Intel*, *Sony* e *Logitech*, che sta conquistando una fetta di mercato sempre più sostanziosa, grazie anche all'inserimento in questi ultimi anni di altre aziende molto importanti nel settore come *LG*. Per gli sviluppatori risulta molto interessante la *Google TV* in quanto presenta al suo interno il sistema operativo *Android*, ciò permette lo sviluppo di un'applicazione che, con i giusti accorgimenti, può essere utilizzata su vari dispositivi, dagli smartphones alle *Smart TV*. *Google* fornisce, inoltre, una documentazione molto più chiara ed esplicativa rispetto a quelle degli altri produttori.

1.4 ECLAP per Smart TV e l'ambiente di sviluppo Samsung

L'applicazione *ECLAP per Smart TV* è stata sviluppata per *Samsung Smart TV*. È un'applicazione multimediale della categoria *Social TV*. Per *Social TV*^[2] si intende la convergenza di Social Media e Televisione, più precisamente è l'attività di interazione attraverso i Social Network, ad esempio pubblicando commenti, opinioni o voti relativamente a trasmissioni d'intrattenimento, talkshow, film o telefilm fruibili attraverso la televisione. L'interazione è bidirezionale (dai Social alla TV e viceversa) e può avvenire direttamente attraverso il televisore (definizione forte di Social TV) oppure attraverso altri device (definizione debole di Social TV), in contemporanea con la visione televisiva oppure no.

ECLAP per Smart TV si può definire una *Social TV* in quanto, in questo primo sviluppo, permette di commentare i contenuti visualizzati, interagendo in questo modo con gli altri utenti.

Samsung ha messo a disposizione degli sviluppatori un *SDK*, continuamente aggiornato, che include:

- *editor* basato su *Eclipse*
- *emulator TV*
- *debugger*
- *automatic test tool*.

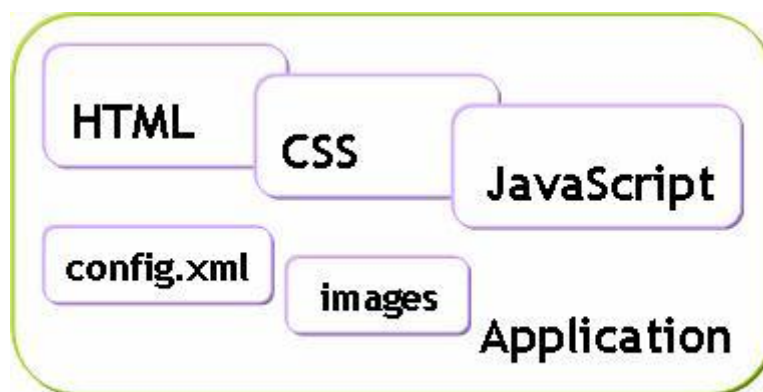


Figura 1.1: Componenti di un'applicazione Smart TV

Un'applicazione è una pagina web composta da:

CAPITOLO 1. Introduzione

- pagina *HTML*
- file *CSS*, definisce lo stile dell'applicazione
- files *Javascript*, controllano il funzionamento dell'applicazione
- file *config.xml*, contiene informazioni sulla versione dell'applicazione.

Una *Smart TV* include un *Application Manager* che permette di adottare una serie di misure per aumentare la velocità di apertura di un'applicazione caricando i files *Javascript* e i files *CSS Background Image* quando necessario, gestendo il consumo della memoria e il *Watch Dog* per non permettere che un'applicazione si chiuda durante l'elaborazione di dati di grandi dimensioni. Inoltre riconosce e identifica gli eventi dovuti ai tasti del telecomando quando vengono premuti dall'utente.

Le features attualmente supportate per le applicazioni *Javascript* sono le seguenti:

- supporto per *HTML5* e *CSS3*
- *App Framework APIs*^[4], che include *Framework API*^[4] (*core API*, *Scene Manager*, *Utility*), *UI Component API*^[5] (*Button*, *List*, *Scroll*,...) e *Service API*^[6] (*VideoPlayer*, *Image Viewer*, *Smart Remote*, *USB*), che semplificano lo sviluppo di applicazioni per *Samsung Smart TV*, garantendo allo sviluppatore l'uso di un fornito insieme di funzionalità
- *Common Modules*^[7], moduli di oggetti comuni che tutte le applicazioni possono usare
- *AllShare API*^[8], fornisce le interfacce per accedere ai providers multimediali
- *Convergence App API*^[9], consente ai dispositivi che supportano il protocollo *HTTP* di comunicare con la *Smart TV*
- *File API*^[10], permette di gestire l'apertura, scrittura e lettura di un file memorizzato all'interno della TV
- *Interactive Remote API*^[11], permette di interagire con la *Smart TV* attraverso altri dispositivi, come smartphones
- *Web Device API*^[12], consente di sfruttare nelle proprie funzioni *Javascript* funzioni specifiche della TV, ad esempio funzioni per il riconoscimento vocale e gestuale.

CAPITOLO 1. Introduzione

- l'*editor*, che ha diversi *UI Components*^[5] pronti all'uso come *Button*, *Checkbox*, *Date picker*, *Help bar*, *Input*, *Label*, *List box*, *Pop-up*, *Scrollbar*, *Video*.

I *Common Modules* permettono alla propria applicazione di visualizzarla correttamente sullo schermo, di riconoscere e identificare gli eventi del telecomando, di usare i plugins e di comunicare con l'*Application Manager*^[13].

Per usare i *Common Modules* è necessario includerli nel file *index.html*, da inserire nella parte superiore del documento per evitare errori. Il *\$MANAGER_WIDGET* permette all'applicazione di accedere ai *Common Modules* previsti.

ESEMPIO:

Inserire in *index.html*:

```
<script type="text/javascript"
src="$MANAGER_WIDGET/Common/API/TVKeyValue.js"></script>
```

Creare un oggetto:

```
var tvKey = new Common.API.TVKeyValue().
```

Infine, una volta istanziato l'oggetto, si può usare una variabile o un metodo del

modulo, come ad esempio: *tvKey.KEY_LEFT*.

I *Common Modules*^[7] disponibili sono i seguenti:

- *TVKeyValue Object*^[14], è un oggetto che definisce il codice dei tasti della TV
- *Widget Object*^[15], ha la funzione di avviare una funzione in modo efficiente
- *Plugins Objects*^[16], permette di utilizzare alcune funzioni dei *plugins*
- *CImageViewer Module*^[17], è un modulo che permette di visualizzare immagini *JPEG*, che possono anche essere visualizzate sullo schermo con una risoluzione 1920x1080 pixel
- *IME Module*^[18], abilita l'inserimento del testo nelle applicazioni
- *SSO Module*^[19], abilita *SSO(single-sign on)*^[20] nelle applicazioni.

Samsung Smart TV oltre a supportare applicazioni basate su *Javascript*, supporta tre tipi di applicazioni *Flash*: l'applicazione *Flash* indipendente che usa direttamente il flash player fornito da *Samsung*, l'applicazione *Flash App* che usa il *Flash Plug-in* collegato al *Samsung TV Browser* (gli oggetti *SWF* sono incorporati nel file *HTML*) e l'applicazione *AIR*.

CAPITOLO 1. Introduzione

Per gestire i comandi della *Smart TV* esistono, oltre al telecomando, dei modi alternativi.

Uno è l'uso di un dispositivo smart esterno, come smartphone, tablet o qualsiasi altro dispositivo che si possa collegare a una *Smart TV* tramite il protocollo *UPnP*^[21] (*Universal Plug and Play*) e comunicare con essa tramite protocollo *HTTP* (*SmartRemote Service API*^[22]). *UPnP*^[21] è un protocollo di rete che permette a diversi terminali di connettersi l'uno all'altro. Il termine “*plug and play*”, che significa “inserisci e utilizza”, indica la possibilità di utilizzare un componente non appena viene connesso al computer o alla rete.

Altri due sistemi che permettono di controllare i comandi che agiscono sulla TV sono il riconoscimento vocale^[23] e gestuale^[24].

Il riconoscimento vocale consente agli utenti di controllare la TV con i comandi vocali, supportando due modalità di riconoscimento: *embedded* e *server*. Con la prima modalità la *Smart TV* cerca di abbinare campioni vocali a comandi predefiniti, ciò viene eseguito dal televisore. Invece con la seconda modalità, i campioni vocali vengono inviati ad un server esterno che li converte in testo e invia i risultati al televisore.

La modalità di riconoscimento dei gesti è *embedded* e la *Smart TV*, attraverso una telecamera, abbina i gesti campione ai gesti predefiniti. Quindi l'insieme dei gesti disponibili sono forniti da *Samsung* e non è possibile aggiungerne altri.

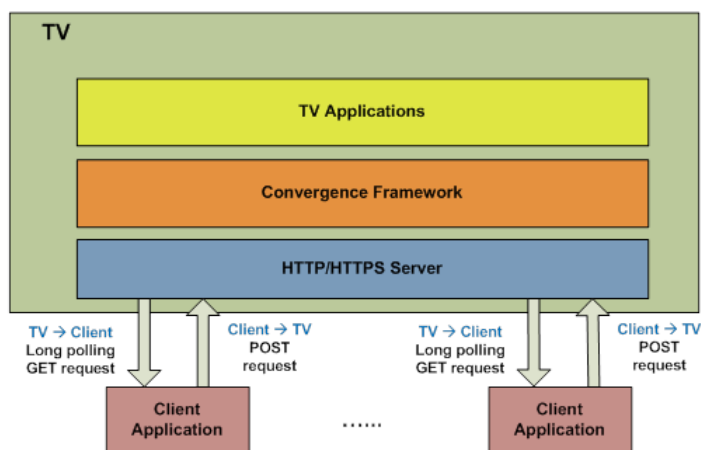


Figura 1.2: Architettura della convergenza

Per consentire la comunicazione bidirezionale tra la TV e un dispositivo, come uno smartphone, viene usato il *framework Convergence* (figura 1.2), che supporta:

- *HTTPs* per comunicazioni sicure

CAPITOLO 1. Introduzione

- i seguenti tipi di scambio di messaggi: dal device alla TV, dalla TV al device, dalla TV a un gruppo di device, da un device a un gruppo di device
- comunicazioni full-duplex in 2 direzioni:
 - i client mandano messaggi all'applicazione della TV usando richieste *POST*
 - una volta che il device client è connesso, deve mantenere un polling *GET* per permettere alle applicazioni TV di mandare messaggi ai clients in qualsiasi momento.

Il framework fornisce un'interfaccia *REST-based* che permette a un'applicazione del device client di connettersi a un'applicazione TV, mandare e ricevere messaggi, iscriversi o lasciare un gruppo^[25].

Per consentire a un'applicazione TV di ricevere messaggi da un'applicazione mobile, si utilizza il modulo *NService*^[26], che fornisce meccanismi per il collegamento di devices client, inviare e ricevere messaggi, e entrare a far parte o lasciare device groups.

Per usare uno standard comune per la comunicazione su rete locale di molteplici dispositivi audio e video, viene usato il sistema *DLNA*. È presente in modo nativo in tutti gli smartphone, tablet e *Smart TV* di ogni marca, tuttavia ogni produttore crea il proprio software di gestione di questo sistema, ad esempio per *Samsung* si chiama "*AllShare*". Gli sviluppatori possono usare le *AllShare APIs*^[8] tramite `webapis.allshare` e ogni servizio può essere fornito attraverso l'attributo `serviceconnector`. La relazione tra le *APIs* è mostrato nella *figura 1.3*.

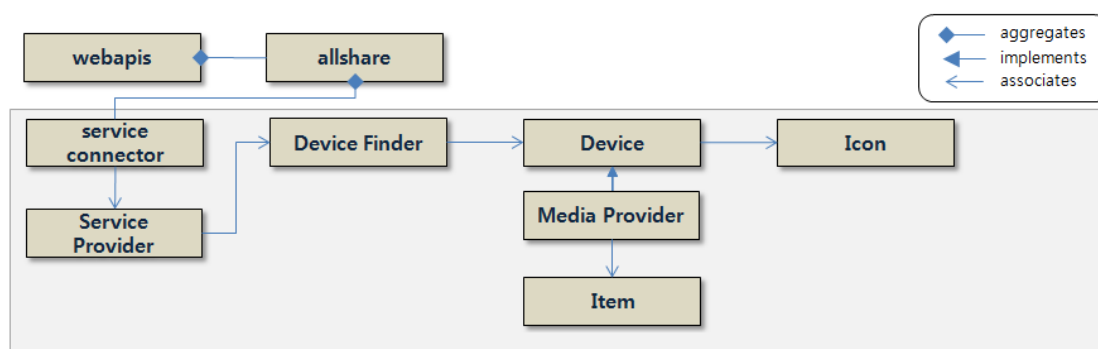


Figura 1.3: AllShare API

Il *Multi-Screen SDK*^[27] fornisce agli sviluppatori un insieme di *API* e strumenti e consente esperienze multi-screen per applicazioni come giochi, social, multimediali etc.

Un'applicazione multi-screen comprende due applicazioni, una da eseguire su *Samsung Smart TV* e l'altra su dispositivo mobile. Le *Multi-Screen APIs* offrono la possibilità di far scoprire a un dispositivo mobile una *Samsung Smart TV* compatibile e servono per connettere le due applicazioni e quindi far comunicare i due dispositivi

1.5 Altre Applicazioni su Samsung Smart TV

La strategia di *Samsung* ha l'obiettivo di offrire un'ampia gamma di contenuti e servizi di qualità agli utilizzatori della *Smart TV* con le sue 144 App. Di queste *Youtube* è l'applicazione che più si avvicina a *ECLAP per Smart TV* in termini di funzionalità offerte.

L'applicazione *Youtube per Smart TV* è strutturata in questo modo: la schermata principale è data da un'unica lista di canali e categorie selezionabili, con la possibilità di vedere le anteprime dei contenuti appartenenti a un canale/categoria selezionato/a. Le anteprime sono formate da immagine, titolo e data di pubblicazione, e vengono mostrate tre alla volta e con una grandezza tale da occupare la maggior parte della schermata. Durante la visualizzazione di un contenuto, premendo la freccia *DOWN*, è possibile vedere i contenuti simili al contenuto trasmesso e contenuti dello stesso autore. Inoltre, attraverso le frecce *LEFT*/*RIGHT*, si può vedere il precedente/successivo contenuto tra quelli simili. Rispetto a *ECLAP per Smart TV* presenta schermate con meno funzionalità, tanto che non viene usata l'helpbar per guidare l'utente durante l'uso dell'applicazione. Con l'applicazione *Youtube* è possibile commentare un contenuto accoppiando la *Smart TV* con un dispositivo mobile connesso alla stessa rete. In questo modo la *Smart TV* fa da secondo schermo ed esclusivamente da riproduttore di contenuti, quindi attraverso il dispositivo mobile si scelgono i contenuti da vedere, si crea una playlist di contenuti, sfruttando tutte le funzionalità che *Youtube* offre, tra cui l'interazione con gli altri utenti, rendendola comunque un'applicazione social.

Un'altra applicazione interessante è *Rai.tv* che permette, nella sezione replay, di rivedere tutta la programmazione Rai dell'ultima settimana relativa ai canali televisivi RAI 1, RAI 2, RAI 3, RAI Premium e RAI 5, di avere altre interessanti funzioni, come la guida TV e di offrire gratuitamente contenuti e video on demand. L'area VOD (video on demand), strutturata nelle sezioni Top Video, Fiction, Cartoni e Programmi, propone invece sia le puntate integrali che "highlights". La

CAPITOLO 1. Introduzione

“guida programmi”, uno dei servizi più apprezzati dagli utenti, facilita infine la scelta del proprio contenuto preferito.

Chili TV consente di noleggiare o acquistare film in streaming e download accedendo a un archivio di 2000 film (anche in Full HD) tramite la piattaforma di video-on-demand senza canone né costi di attivazione, ma solo pagando il prezzo del singolo titolo. Un limite di questo servizio è rappresentato dalla necessità di disporre di connessioni adeguate, infatti per lo streaming base basta 1 Mbps, mentre per quelli di maggiore qualità sono necessari almeno 2,5 Mbps e 4,5 Mbps.

Anche *Deezer*, il noto servizio di musica in streaming, è approdato sulle *Smart TV* di *Samsung* potendo offrire un ampio catalogo di brani musicali direttamente dalla TV. Chi è già in possesso di un abbonamento al servizio musicale, avrà accesso alle proprie playlist, canzoni o suggerimenti direttamente tramite la *Smart TV*, oltre a poter accedere all'intero database composto da più di 20 milioni di brani. Chi invece non ha ancora provveduto a sottoscrivere un abbonamento, può usufruire del servizio Discovery, che offre un accesso illimitato ai canali radio, nonché dell'ascolto delle clip delle canzoni, della durata di trenta secondi.

Capitolo 2

ECLAP overview

2.1 ECLAP overview

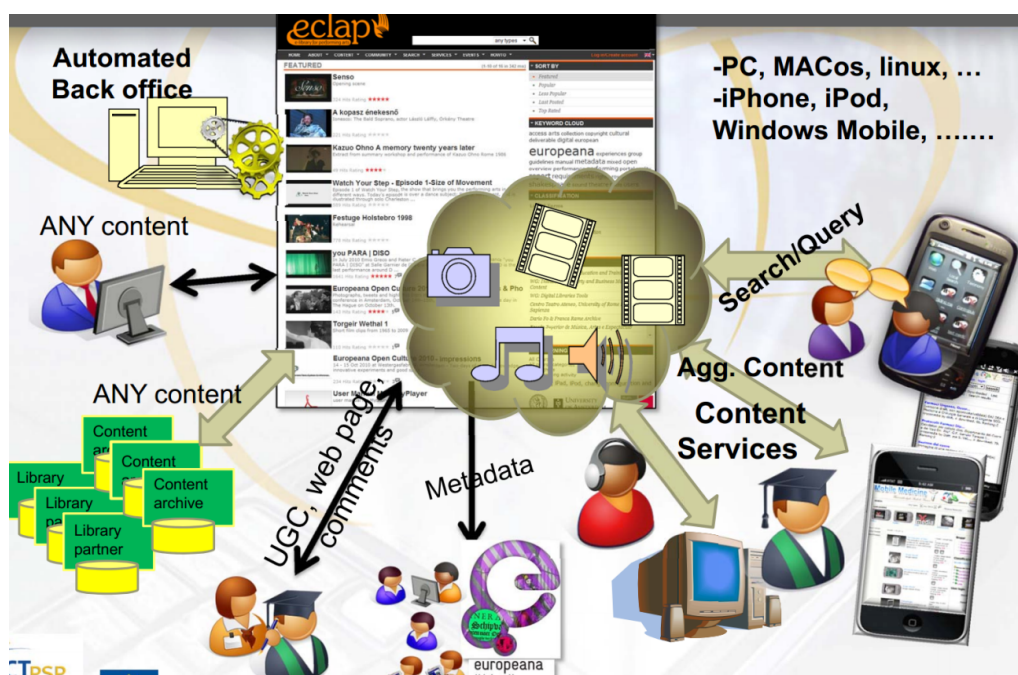


Figura 2.1: Portale ECLAP

La ricchezza e il valore del patrimonio delle arti dello spettacolo è indiscutibile, ma anche se queste collezioni vengono ora digitalizzate e pubblicate online, restano comunque sparse perché il coordinamento tra le biblioteche digitali e il campo delle arti dello spettacolo è carente. Vista comunque la forte domanda di accesso a questi contenuti da parte del pubblico, *ECLAP* (*European Collected Library*

CAPITOLO 2. ECLAP overview

of Artistic Performance) è nata con l'idea di colmare questa lacuna, creando un ricco archivio online europeo di tutte le arti dello spettacolo e crea una rete di istituzioni nel campo dello spettacolo con l'intento di introdurli nell'Europa digitale, fornendo loro gli strumenti necessari per digitalizzare i propri contenuti^[28].

Gli obiettivi di *ECLAP*^[28] sono:

- raccogliere e rendere disponibili attraverso i portali *ECLAP* ed *Europeana* il materiale più rilevante delle arti dello spettacolo europeo; i materiali provenienti dal teatro, danza, musica, cinema e film rappresentano spettacoli, lezioni, corsi, materiale didattico e altro nel formato video, audio, documenti, immagini, animazioni, playlist, annotazioni, contenuti interattivi, ecc...
- creare una rete stabile e aperta tra le istituzioni europee delle arti dello spettacolo con lo scopo di aiutarli a sfruttare i contenuti digitali e fornire loro tecnologie e strumenti innovativi
- fornire soluzioni e servizi alle principali istituzioni delle arti dello spettacolo, come immissione di contenuti, arricchimento di metadati, distribuzione e aggregazione di contenuti in *Europeana*, gestione dei diritti di proprietà intellettuale, visibilità dei canali di contenuti, ricerca semantica multilingue/query di ricerca fuzzy
- fornire soluzioni e servizi a una varietà di utenti: insegnanti, studenti, artisti, ricercatori e appassionati delle arti dello spettacolo a scopo educativo, informativo e d'intrattenimento.

ECLAP è un consorzio che riunisce istituzioni europee delle arti dello spettacolo, università e istituti di ricerca. I partners affiliati sono 19 e provengono da tredici Paesi diversi, che unendo le loro competenze e conoscenze scientifiche si prefiggono di raggiungere gli obiettivi fissati da *ECLAP*. Nasce come un progetto di ricerca cofinanziato dal Programma Ict-Psp Cip della Commissione Europea e consiste in una prima fase di sviluppo strutturale, una seconda di raccolta dei primi contenuti e una terza di promozione^[28].

ECLAP fa uso di un database di indicizzazione avanzata e strumenti di distribuzione per la produzione e la diffusione del ricco patrimonio artistico europeo e questo si traduce in un arricchimento e promozione della cultura europea.

Attualmente *ECLAP* distribuisce più di un milione di contenuti, di cui circa 125000 provengono da 32 prestigiose istituzioni europee, ed è in grado di riprodurre immagini, video, documenti, audio, ebooks, ePub, animazioni, slides, playlists, raccolte, 3D, musica braille, annotazioni, ecc...^[28] (*figura 2.2*). È possibile accedere ai contenuti offerti da *ECLAP* anche attraverso dispositivi mobili (iPhone,

CAPITOLO 2. ECLAP overview

iPad, iPod, Android and Windows Phone 7) installando l'applicazione *Content Organizer*^[29], con cui è possibile accedere, scaricare e organizzare i contenuti del portale *ECLAP*.



Figura 2.2: Tipi di contenuti riprodotti da ECLAP

È possibile registrarsi al portale *ECLAP* accedendo alla pagina dedicata attraverso il link *Login/Create account*, una volta entrati nella pagina di registrazione si possono inserire i dati di login e le informazioni personali: username, e-mail, nome, cognome, stato e, facoltativamente, a quale gruppo/i del portale aderire. Dopo aver completato il modulo di registrazione, viene inviata, all'indirizzo di posta elettronica specificato, una password generata in automatico che l'utente può cambiare accedendo al portale e aprendo la pagina per la modifica del proprio profilo.

Agli utenti autenticati è data la possibilità di sfruttare alcuni strumenti innovativi e di eseguire ulteriori azioni oltre alla riproduzione di contenuti. Alcune azioni aggiuntive che permettono di interagire con i contenuti sono: commento, aggiunta ai preferiti, creazione e condivisione di playlist, creazione di collezioni private/pubbliche e raccomandazione/promozione. Alcuni strumenti interessanti e innovativi offerti da *ECLAP* sono: stabilire sincronizzazioni/annotazioni tra contenuti audiovisivi (audio, video e immagini) sfruttando lo strumento

CAPITOLO 2. ECLAP overview

MyStoryPlayer^[30] e navigare tra le relazioni semantiche degli oggetti di *ECLAP* (contenuti, utenti, collezioni annotazioni, commenti, gruppi, ecc...) attraverso lo strumento *Social Graph*^[31].

MyStoryPlayer^[30] (figura 2.3). è uno strumento che sfrutta le annotazioni audiovisive integrate in *ECLAP* e permette di interagire con oggetti multimediali attraverso relazioni logiche e temporali. Quindi un utente può collegarsi a diversi contenuti multimediali che mostrano lo stesso argomento o evento, creando così dei rapporti espliciti fra di loro che permettono di passare da un contenuto all'altro all'interno della stessa interfaccia. *MyStoryPlayer* è utile per riprodurre contemporaneamente uno stesso evento ripreso da diverse angolazioni e, inoltre, permette di sincronizzare risorse di formato diverso (video, immagini, diapositive, audio) che trattano lo stesso argomento, ad esempio a scopo didattico.



Figura 2.3: *MyStoryPlayer* di un laboratorio teatrale

Un approccio più innovativo per la navigazione nella piattaforma *ECLAP* è rappresentato dall'applicazione *Social Graph*^[31] (figura 2.4), che permette di visualizzare e interagire con gli elementi di *ECLAP* e ha l'obiettivo di migliorare

CAPITOLO 2. ECLAP overview

l'interazione sociale fra gli utenti di *ECLAP*. Attraverso l'interfaccia grafica è possibile vedere per ogni elemento gli oggetti associati.



Figura 2.4: Grafico sociale per “you PARA / DISO”

Si può accedere al *Social Graph* dalla home page di ciascun utente o durante la visualizzazione di un contenuto, e ogni utente grazie a questo sistema può navigare fra le numerosi relazioni esistenti, ad esempio tra contenuti, utenti, annotazioni, ecc...

CAPITOLO 2. ECLAP overview

Social Graph è costituito da nodi e archi (*figura 2.5*). I nodi indicano elementi di *ECLAP* (video, immagini, utenti, gruppi e annotazioni), rappresentati da rettangoli, o specifiche relazioni esistenti tra due o più oggetti diversi, rappresentati da cerchi. Gli archi rappresentano i collegamenti tra gli oggetti e hanno una direzione utile a leggere in modo inequivocabile il rapporto tra gli oggetti.

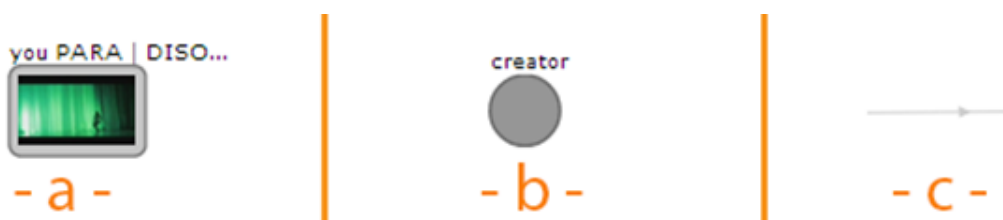


Figura 2.5: Nodi e arco del Social Graph

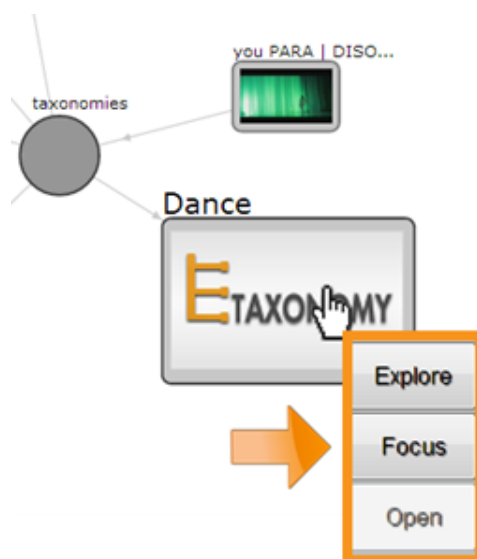


Figura 2.6: Funzionalità del Social Graph

L'uso del *Social Graph* è intuitivo ed è possibile aumentare/ridurre la sua dimensione, spostare il grafico, trascinare e rilasciare a piacere i vari nodi del grafico. Cliccando su un nodo è possibile esplorarlo visualizzando le sue relazioni, centralizzare il *Social Graph* sull'elemento selezionato e mostrare l'elemento selezionato attraverso il riproduttore multimediale (*figura 2.6*).

Per caricare contenuti sul portale *ECLAP* esistono vari metodi: nel caso di utente privato è possibile sfruttare il servizio *Fast Upload* di *ECLAP*, altrimenti

CAPITOLO 2. ECLAP overview

nel caso di partner affiliato, che si suppone fornisca grandi quantità di contenuti di dimensione sostenute, si può sfruttare il servizio *Upload* di *ECLAP* che dà la possibilità di caricare contemporaneamente più file (funziona bene con circa 150) di dimensioni non superiori a 1 GByte l'uno. Nel caso si voglia caricare files di dimensione superiori a 1 GByte è possibile sfruttare il sito FTP di *ECLAP*, che può trasferire più di 200 GByte di contenuti al giorno, oppure attraverso l'invio di hard disk o DVD / CD o nastri^[32].

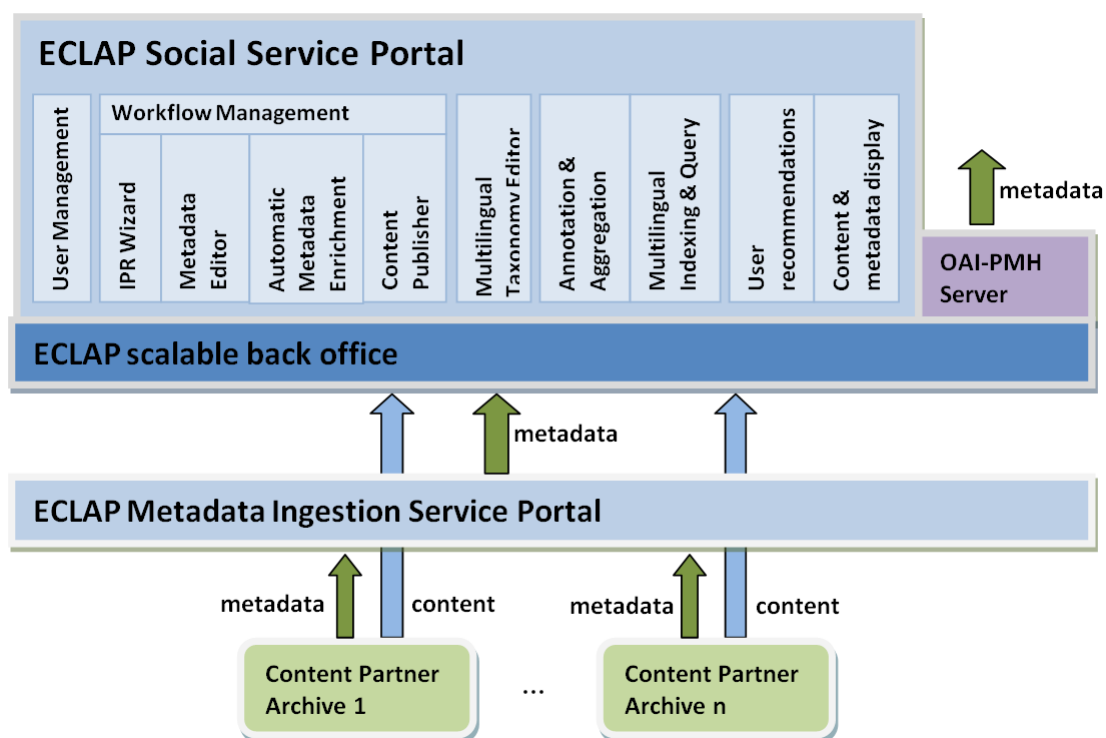


Figura 2.7: Architettura a livelli di ECLAP

L'architettura a livelli di *ECLAP* (figura 2.7) per la gestione dei contenuti e dei metadati è costituita da:

- *ECLAP Social Service Portal* è uno strumento front-end e fornisce strumenti di front-office per lavorare su contenuti e metadati, gestire contenuti, ricercare contenuti e utenti simili, gestire i diritti di proprietà intellettuale con la procedura guidata *IPR* con cui si possono impostare i permessi o le restrizioni sull'utilizzo dei contenuti pubblicati sul portale *ECLAP*, ecc...
- *ECLAP scalable BackOffice* lavora in back-side e ha il compito di gestire in modo automatico tutte le attività di elaborazione, codifica e archiviazione. Questa parte fa uso di *AXMEDIS AXCP*, uno strumento gratuito

CAPITOLO 2. ECLAP overview

per qualsiasi istituzione no profit, usato per automatizzare la produzione, la protezione e la distribuzione di contenuti multimediali. *AXMEDIS* è un insieme di standards per contenuti digitali dell'Unione Europea, nato inizialmente come un progetto di ricerca parzialmente finanziato dalla Commissione Europea, e ora distribuito gratuitamente sotto forma di framework open source. *ECLAP BackOffice* è una soluzione scalabile nell'inserimento dei contenuti, nella gestione del database, nell'elaborazione dei metadati, ecc...

- *ECLAP Metadata Ingestion Service Portal* serve ai fornitori di contenuti o alle persone autorizzate dagli stessi a fornire metadati da associare al contenuto caricato. Ciò può avvenire attraverso il *Metadata Editor di ECLAP* o fornendo file *XML* o *CSV* insieme a un foglio di stile *XSLT* utile per la trasformazione del file *XML*. *ECLAP* esegue automaticamente la trasformazione dei metadati nel modello *EDM* del portale *Europeana* e li pubblica su tale portale, in modo da offrire maggiore visibilità al relativo contenuto. *Europeana* è una biblioteca digitale europea che include libri, film, dipinti, giornali, archivi sonori, mappe, manoscritti ed archivi. Attualmente più del 90% dei contenuti di *ECLAP* viene indicizzato in *Europeana*, mentre il reale file del contenuto è accessibile solo tramite il portale *ECLAP*^[32].

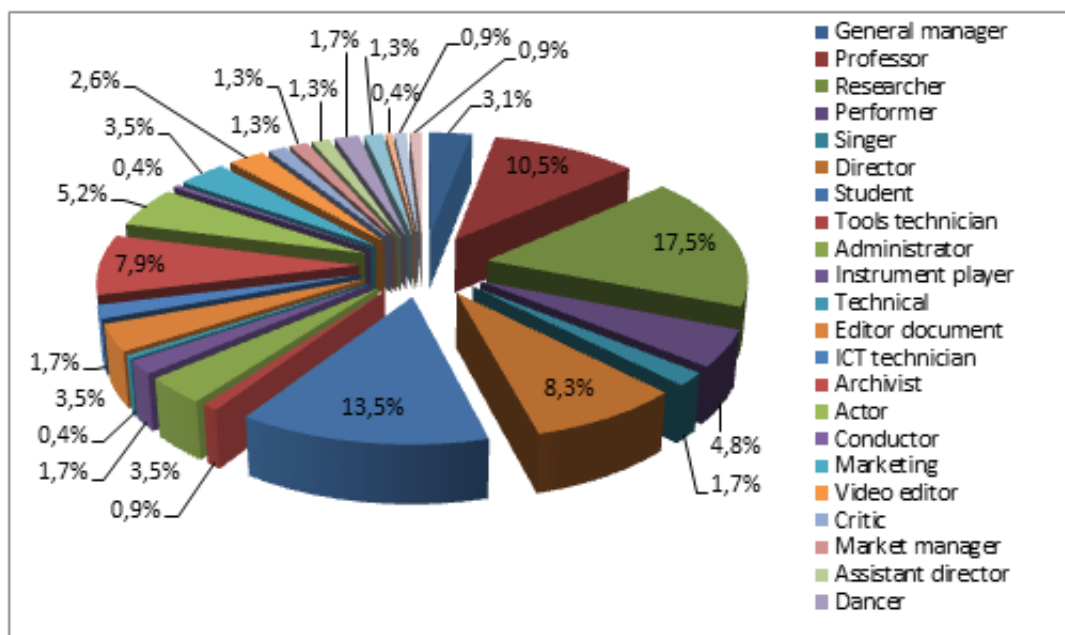


Figura 2.8: Utenti ECLAP raggruppati per attività principale

CAPITOLO 2. ECLAP overview

ECLAP fornisce una serie di statistiche del suo utilizzo e delle analisi fatte a diversi livelli tra cui statistiche generali sugli accessi, download e upload, statistiche generali sulle query statistiche generali, sulle azioni effettuate dall'utente, analisi della *Social Network di ECLAP* e statistiche dei gruppi sull'arricchimento dei contenuti e della promozione sociale.

Gli utenti di *ECLAP* provengono da più di 90 Paesi diversi di tutto il mondo, di cui la maggior parte sono Paesi europei, ma sono ben rappresentati anche Stati Uniti, Canada, Cina e Giappone.

La *figura 2.8* mostra la distribuzione degli utenti di *ECLAP* secondo la loro attività principale, la maggior parte è rappresentata da ricercatori (17,5%), studenti (13,5%) e professori (10,5%)^[33].

Capitolo 3

ECLAP per Smart TV: analisi dei requisiti e casi d'uso

3.1 Analisi dei requisiti

- [R1] Visualizzazione schermata principale all'avvio dell'applicazione
 - [R1.1] La schermata principale è organizzata nel seguente modo: In alto a sinistra è visualizzato un video *In primo piano* in esecuzione. Nella parte centrale, lateralmente al *video player*, sono rappresentati, tramite un'anteprima e un titolo, tre contenuti *Simili* al video *In primo piano* in esecuzione. Nella parte inferiore sono rappresentati, sempre tramite una preview e un titolo, altri sei contenuti *In primo piano*. Sempre nell'area centrale, al di sopra dei tre contenuti *Simili*, è evidente un campo ricerca, che consente di selezionare una lista di contenuti in base a una *keyword*. Nella parte destra è visibile una classificazione tassonomica dei contenuti e un elenco di canali (*Miei preferiti*, *Ultimi visti*, *In primo piano*, *Più visti*, *Ultimi caricati*, *Più votati* e *Visti dagli amici*). La selezione di una voce tassonomica o di un canale permette di accedere ai relativi contenuti. Per poter visualizzare i tre canali *Miei preferiti*, *Ultimi visti* e *Visti dagli amici* è necessario effettuare il login.
- [R2] Schermata matrice contenuti
 - [R2.1] Si accede selezionando una voce dall'elenco posto a destra della schermata principale, precedentemente descritto. Questa schermata mostra, su più pagine, tutti i contenuti trovati. Ogni pagina è composta da una matrice 3x5 di contenuti, di cui è visibile la relativa ante-

prima con il relativo titolo. In alto a destra è visualizzato il conteggio delle pagine.

- [R3] Schermata di login
 - [R3.1] Si accede dalle schermate descritte in [R1] e [R2]. È composta da due campi di input in cui inserire, dopo l'effettuazione della registrazione a www.eclap.eu, nome utente e password.
- [R4] Schermata di modifica impostazioni
 - [R4.1] Si accede dalle schermate descritte in [R1] e [R2]. È divisa orizzontalmente in tre diverse sezioni: una relativa alla lingua dell'interfaccia, una al tipo di contenuto da visualizzare e la terza relativa alla lingua dei metadati. Le lingue dell'interfaccia implementate sono italiano e inglese. I contenuti possono essere visualizzati per mezzo di solo video, solo immagini o entrambi.
- [R5] I metadati relativi a un contenuto possono essere visualizzati sia dalle schermate descritte in [R1] e [R2] e sia dalla schermata di visualizzazione a schermo intero del contenuto, premendo i tasti *INFO* o *C(giallo)* rispettivamente.
- [R6] Commentare/Aggiungere ai preferiti un contenuto.
 - [R6.1] Queste due azioni si possono effettuare dalla schermata di visualizzazione a schermo intero del contenuto, dopo l'esecuzione del login. Inoltre saranno visibili, uno per volta, i commenti precedenti.
- [R7] Nella *helpbar*, presente in ogni schermata, è indicata ogni possibile azione con associato il relativo tasto del telecomando.
- [R8] La navigazione tra le varie schermate avviene premendo il tasto *RETURN*.
- [R9] Il sistema è intuitivo nell'uso e accessibile a tutti, salvo alcune azioni che sono riservate a utenti autenticati come descritto in precedenza.

3.2 Casi d'uso

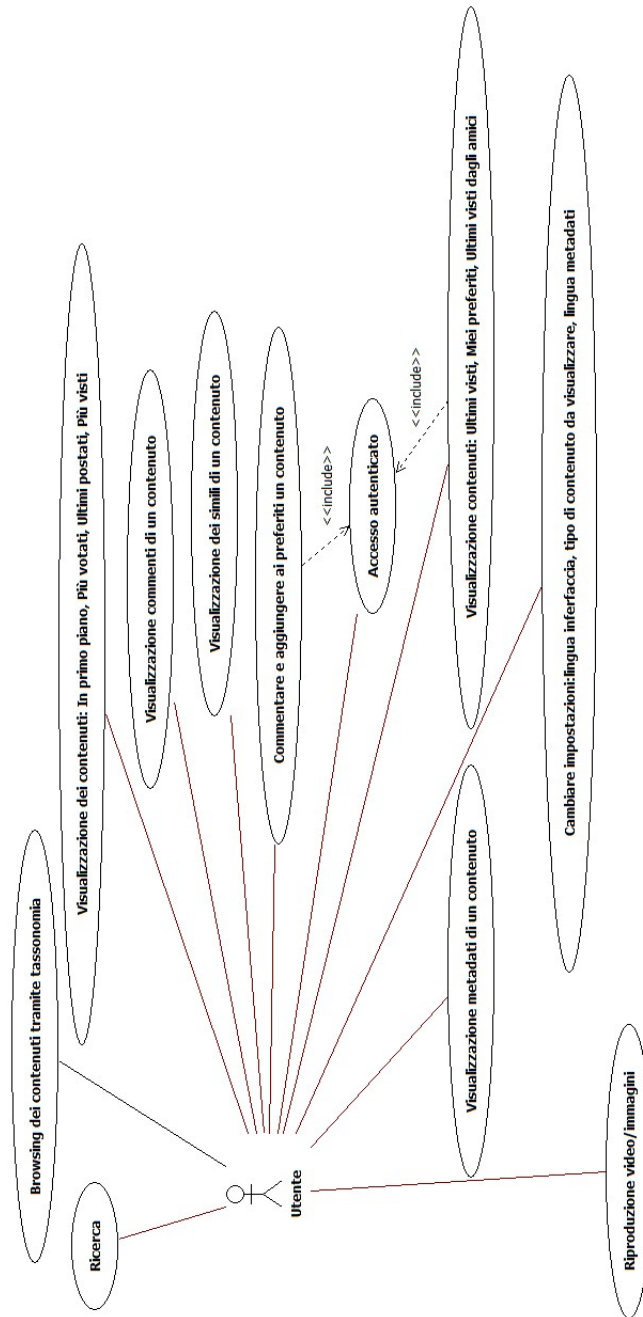


Figura 3.1: Diagramma dei casi d'uso ECLAP per Smart TV

Visualizzazione dei contenuti (In primo piano, Più visti, Ultimi caricati e Più votati)

Precondizione: Schermata principale/Schermata matrice contenuti

Sequenza di eventi:

- i.* Selezionare il canale di proprio interesse.
- ii.* Dopo aver premuto il tasto *ENTER* saranno visualizzati i contenuti relativi al canale selezionato.

Invariante: Il caso d'uso termina incondizionatamente se il focus viene spostato in una porzione di schermo non relativa ai canali o se si passa a un'altra schermata.

Visualizzazione dei contenuti (Miei preferiti, Ultimi visti e Visti dagli amici)

Precondizione: Schermata principale/Schermata matrice contenuti e autenticazione utente.

Sequenza di eventi:

- i.* Selezionare il canale di proprio interesse.
- ii.* Dopo aver premuto il tasto *ENTER* saranno visualizzati i contenuti relativi al canale selezionato.

Invariante: Il caso d'uso termina incondizionatamente se il focus viene spostato in una porzione di schermo non relativa ai canali o se si passa a un'altra schermata.

Browsing dei contenuti tramite tassonomia

Precondizione: Schermata principale/Schermata matrice contenuti

Sequenza di eventi:

- i.* Spostare il focus nella porzione di schermo relativa alla tassonomia.
- ii.* Navigare all'interno delle sottocategorie per scegliere quella di proprio interesse. Le categorie che presentano delle sottocategorie sono contrassegnate da un triangolo. I tasti di navigazione sono: frecce *UP/DOWN* per navigare, *ENTER* per selezionare, *LEFT* per tornare alla categoria madre.

CAPITOLO 3. ECLAP per Smart TV: analisi dei requisiti e casi d'uso

Invariante: Il caso d'uso termina incondizionatamente se il focus viene spostato in una porzione di schermo non relativa alla tassonomia o se si passa a un'altra schermata.

Ricerca

Precondizione: Schermata principale

Sequenza di eventi:

- i.* Il caso d'uso inizia premendo il tasto *B (verde)* del telecomando, in tal modo il focus è spostato al campo di input.
- ii.* Dopo aver digitato una keyword tramite tastiera/tastierino numerico, premendo il tasto *ENTER* sarà visualizzato un elenco di possibili contenuti attinenti ai parametri di ricerca impostati.

Invariante: A partire dal punto *i*, il caso d'uso termina quando è premuto il tasto *RETURN*.

Riproduzione immagini/video

Precondizione: Schermata principale/Schermata matrice contenuti

Sequenza di eventi:

- i.* Selezionare l'anteprima di un contenuto e premere il tasto *ENTER*.
- ii.* Il contenuto relativo all'anteprima selezionata viene visualizzato a schermo intero.

Invariante: Il caso d'uso termina quando viene premuto il tasto *RETURN*, tornando così alla schermata precedente.

Accesso autenticato

Precondizione: Schermata principale/Schermata matrice contenuti

Sequenza di eventi:

- i.* Il caso d'uso inizia quando viene premuto il tasto *C (giallo)*.
- ii.* Si visualizza la schermata di login.

CAPITOLO 3. ECLAP per Smart TV: analisi dei requisiti e casi d'uso

iii. Digitare nome utente e password dopo essersi registrati su `www.eclap.eu`, nel caso non sia già stato fatto.

iv. Premere il tasto *ENTER* per accedere ai servizi riservati agli utenti autenticati

Invariante: Il caso d'uso termina in due modi: premendo il tasto *RETURN*, con cui si torna alla schermata visualizzata precedentemente, o premendo il tasto *ENTER*. In quest'ultimo caso se la coppia *nome utente/password* inserita è errata allora comparirà un popup in cui si avvisa l'utente che c'è stato un errore e si torna al punto *ii*, altrimenti si torna alla schermata visualizzata precedentemente come utente autenticato.

Visualizzazione metadati di un contenuto

Precondizione: L'anteprima di un contenuto deve essere selezionata o il contenuto deve essere mostrato a tutto schermo.

Sequenza di eventi:

i. Se è stata selezionata l'anteprima di un contenuto, premere il tasto *INFO*; se è visualizzato un contenuto a schermo intero, premere il tasto *B (verde)*.

ii. Viene mostrato un popup con i metadati relativi al contenuto: titolo, durata (solo per i contenuti video), data di pubblicazione, numero di visualizzazioni, data dell'evento, soggetto, descrizione e chi l'ha pubblicato.

Invariante: Il caso d'uso termina quando viene selezionato e cliccato il *button CHIUDI* che verrà mostrato nel relativo popup o se viene premuto il tasto *RETURN*.

Visualizzazione dei simili di un contenuto

Precondizione: Un contenuto deve essere mostrato a tutto schermo.

Sequenza di eventi:

i. Dopo aver premuto la freccia *DOWN* verrà mostrato un popup con all'interno le anteprime dei contenuti simili a quello visualizzato.

Invariante: Il caso d'uso termina quando viene cliccato il tasto *RETURN*, che chiude il popup, oppure premendo il tasto *ENTER*, con cui si può visualizzare il contenuto simile selezionato.

Visualizzazione dei commenti di un contenuto

Precondizione: Un contenuto deve essere visualizzato a schermo intero

Sequenza Eventi:

- i.* Dopo aver premuto il tasto *INFO* verranno mostrati, se ci sono e uno alla volta, i commenti relativi al contenuto visualizzato.

Invariante: il caso d'uso termina in due modi: premendo il tasto *INFO*, si chiuderà il popup di visualizzazione dei commenti, o premendo il tasto *RETURN*, con cui si torna alla schermata visualizzata precedentemente.

Commentare un contenuto

Precondizione: Un contenuto deve essere visualizzato a schermo intero, il popup relativo agli eventuali commenti deve essere mostrato e l'utente deve essere autenticato

Sequenza di eventi:

- i.* Dopo aver premuto il tasto *ENTER* o *D (blu)* si attiva il campo di input per poter commentare il contenuto.
- ii.* Comporre il commento e premere il tasto *ENTER* per inserirlo.

Invariante: il caso d'uso termina incondizionatamente quando viene premuto di nuovo il tasto *INFO*, in questo caso verrà semplicemente chiuso il popup di visualizzazione dei commenti. Invece, se viene premuto il tasto *RETURN* si torna alla schermata visualizzata in precedenza. Oppure premendo il tasto *D (blu)*, con cui viene eliminato il campo di input relativo all'immissione di un commento.

Aggiungere ai preferiti un contenuto

Precondizione: Un contenuto deve essere visualizzato a schermo intero e l'utente deve essere autenticato

Sequenza di eventi:

- i.* Dopo aver premuto il tasto *INFO* è possibile aggiungere il contenuto visualizzato nei preferiti con il tasto *A (rosso)* o commentare tale contenuto.

Invariante: il caso d'uso termina in due modi: premendo il tasto *INFO*, si chiuderà il popup di visualizzazione dei commenti, o premendo il tasto *RETURN*, con cui si torna alla schermata visualizzata precedentemente.

Cambiare le impostazioni (lingua interfaccia, tipo di contenuto da visualizzare, lingua metadati)

Precondizione: Schermata principale/ Schermata con matrice contenuti

Sequenza di eventi:

- i.* Il caso d'uso inizia quando viene premuto il tasto *D* (*blu*).
- ii.* Si accede alla schermata di *settings*, in cui è possibile selezionare, tramite le frecce *UP/DOWN*, i tre tipi di impostazioni e, tramite le frecce *LEFT/RIGHT*, le scelte relative alle diverse impostazioni.

Invariante: il caso d'uso termina premendo il tasto *RETURN/ENTER* e confermando le eventuali modifiche effettuate. Si torna così alla schermata precedente.

3.3 Layout delle schermate in fase di pre-sviluppo

Prima dello sviluppo dell'applicazione è stato realizzato un prototipo delle schermate, alcune delle quali sono state leggermente modificate al fine di migliorare l'esperienza dell'utente.

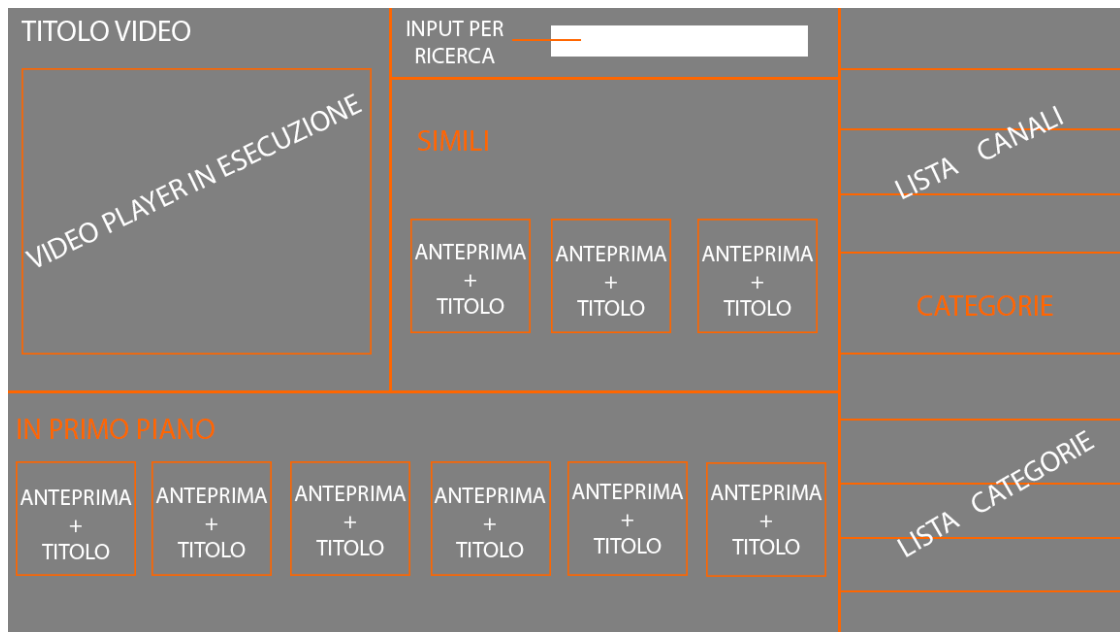


Figura 3.2: Prototipo della schermata Home

CAPITOLO 3. ECLAP per Smart TV: analisi dei requisiti e casi d'uso

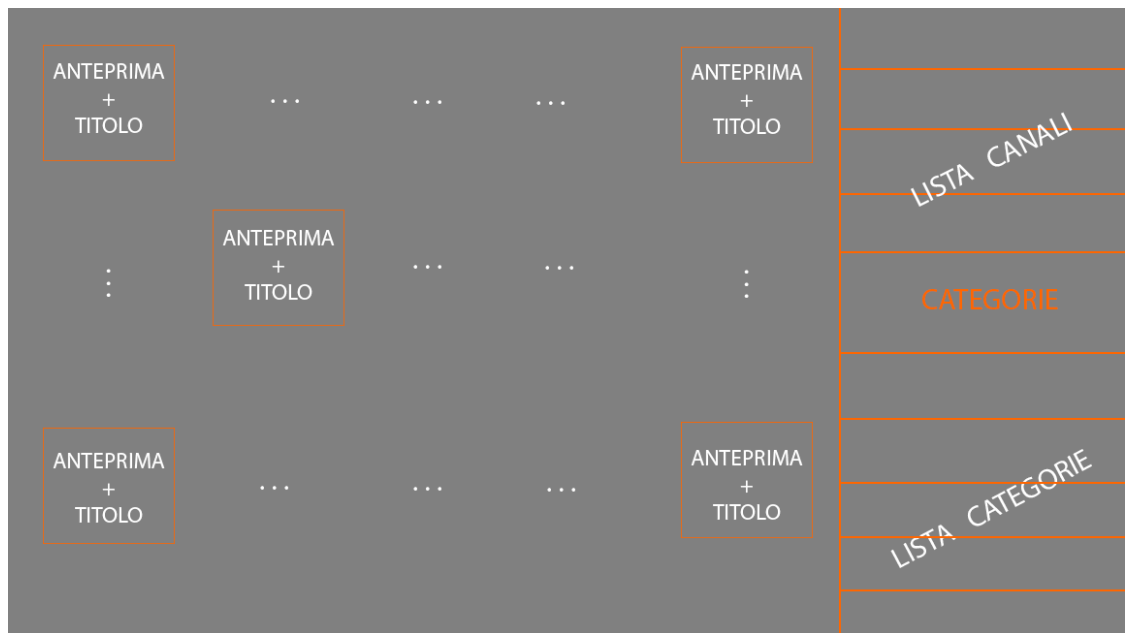


Figura 3.3: Prototipo della schermata SceneContent

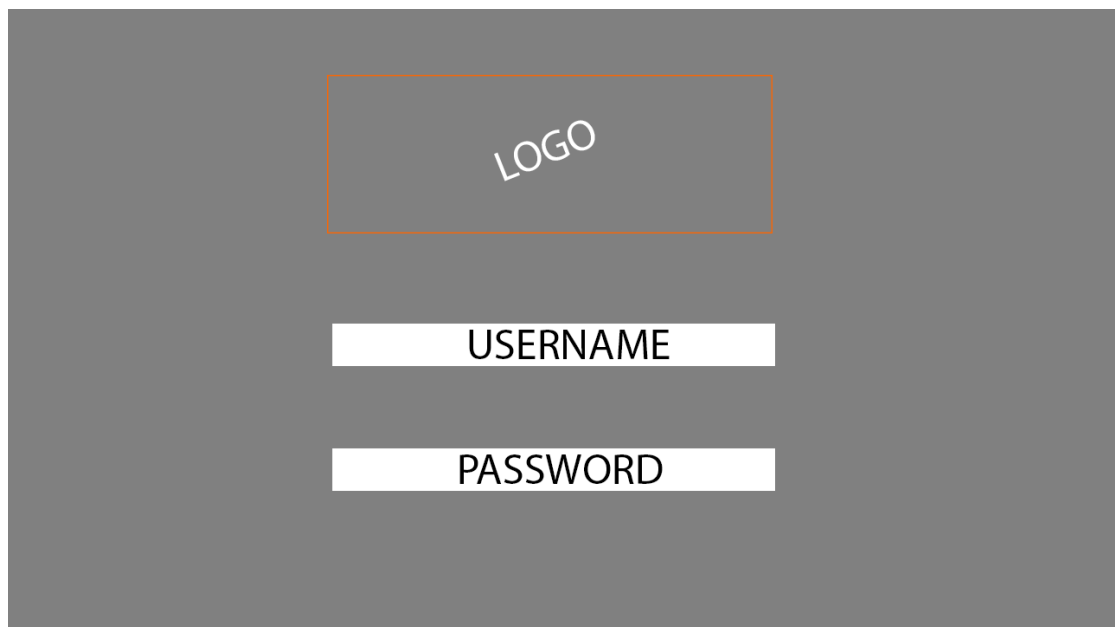


Figura 3.4: Prototipo della schermata di login

CAPITOLO 3. ECLAP per Smart TV: analisi dei requisiti e casi d'uso

Nella *figura 3.2* viene rappresentata la schermata principale, nella quale viene mandato in esecuzione un video estratto casualmente tra quelli *In primo piano*, in modo che l'utente possa interagire sin dall'inizio con l'applicazione. Vengono offerti, inoltre, una serie di contenuti *In primo piano* e *Simili* al video in esecuzione, con lo scopo di interessare e incuriosire l'utente. Nella stessa schermata è presente un campo di input per la ricerca dei contenuti tramite *keyword* e, sul lato destro, una barra verticale che permette di visualizzare un insieme di contenuti selezionando una categoria specifica o uno dei vari canali proposti.

Scegliendo una *categoria/canale* si passa alla schermata di *figura 3.3* occupata per la maggior parte da una serie di anteprime di contenuti inerenti alla scelta, ordinati secondo una matrice 3x5. Nella stessa schermata è visibile la barra verticale sulla destra descritta precedentemente, in modo che l'utente possa continuare la navigazione fra i vari canali e le varie categorie senza dover obbligatoriamente tornare nella schermata principale.

La schermata di *figura 3.4*, accessibile dalle schermate appena descritte, riguarda quella di login, costituita semplicemente da un'immagine del logo dell'applicazione e da due campi di input in cui inserire nome utente e password. Questa schermata ha subito dei leggeri cambiamenti in fase di verifica, come si può vedere nel capitolo *Validazione e uso*, aggiungendo una tastiera nella metà destra della schermata per aiutare l'utente a digitare in modo corretto ed efficiente i due parametri che permettono l'accesso al portale.



Figura 3.5: Prototipo della schermata delle impostazioni

CAPITOLO 3. ECLAP per Smart TV: analisi dei requisiti e casi d'uso

La *figura 3.5* rappresenta la schermata relativa alle impostazioni dell'applicazione. Sono state previste tre tipi di impostazioni: lingua dell'interfaccia, tipo di contenuto da visualizzare e lingua metadati. Accanto a ogni impostazione sono visibili le relative opzioni che sarà possibile selezionare.

Capitolo 4

Eclap per Smart TV: progettazione

4.1 Architettura

ECLAP per Smart TV è un'applicazione *web-based* accessibile via web per mezzo della rete Internet in una architettura di tipo *three-tier*^[34] “a tre strati”, che offre determinati servizi all'utente.

L'espressione architettura *three-tier*^[34] indica una particolare architettura software di tipo *multi-tier*^[35] che prevede la suddivisione dell'applicazione in tre diversi strati (o livelli) che interagiscono fra loro, secondo le linee generali del paradigma *client-server*, utilizzando interfacce ben definite. In questo modo, ciascuno dei tre strati può essere modificato o sostituito indipendentemente dagli altri, conferendo così scalabilità e manutenibilità all'applicazione.

I tre livelli in cui si sviluppa *ECLAP per Smart TV* sono:

- *livello di presentazione*, che costituisce l'interfaccia utente dell'applicazione web e corrisponde a quello che nelle applicazioni client-server standard è il client. Questo livello è rappresentato dall'applicazione *ECLAP per Smart TV* sviluppata
- *livello applicazione*, che contiene la logica elaborativa dell'applicazione ed è in grado di soddisfare le richieste di dati e di elaborazioni provenienti dal client. Le modalità di realizzazione di tale livello dipendono spesso dalle caratteristiche e dalle tecnologie supportate dal server web e/o da componenti installati sul server applicativo. In *ECLAP per Smart TV* per generare i contenuti dinamici richiesti dal client è stato creato e installato un modulo scritto in *PHP* nel server gestito da *Drupal*
- *livello dati*, che fornisce servizi non direttamente disponibili tramite il server web. In *ECLAP per Smart TV*, tale livello è rappresentato da un

CAPITOLO 4. Eclap per Smart TV: progettazione

database *MySQL* utilizzato per la gestione della persistenza dei dati e la loro interrogazione.

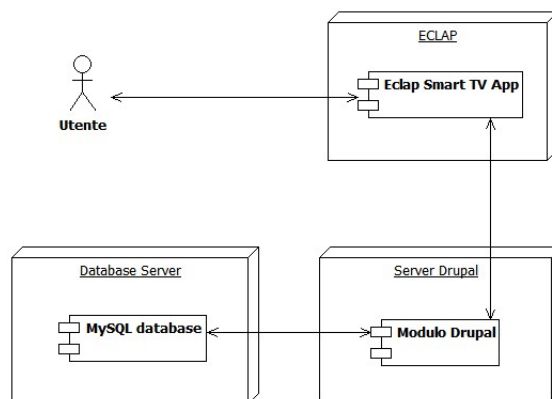


Figura 4.1: Diagramma Deployment ECLAP per Smart TV

4.2 Applicazione scene-based e componenti usati

Dal punto di vista visivo, le applicazioni per *Smart TV* sono formate da una o più scene, che posso essere considerate come strati sovrapposti ed è ciò che l'utente vede e con cui interagisce. Ogni scena ha uno stato *show/hidden* e *focused/blurred*. È possibile sviluppare applicazioni scene-based gestendo le scene e i relativi stati. Ogni scena ha un proprio insieme di file: *HTML*, *CSS* e *Javascript*.

In *figura 4.2* si considera ogni scena presente in *ECLAP per Smart TV* come un oggetto e, attraverso la notazione *UML*, si rappresenta l'interazione fra le varie scene e fra le scene e i files *Javascript*. Le scene sono rappresentate con il colore blu, mentre i files *Javascript* con l'arancione ed esistono due tipi di interazioni: `<<open>>` e `<<use>>`. Come si vede nella *figura 4.2* le due scene centrali, *SceneHome* e *SceneContent*, oltre ad essere collegate tra di loro, lo sono con tutte le altre quattro, *SceneLogin*, *SceneSettings*, *SceneImage* e *videoPlayer*. La *SceneHome* è la scena principale, ossia la prima a comparire quando l'applicazione viene avviata; la *SceneContent* è la scena che contiene la matrice 3x5 di contenuti, ognuno rappresentato da un'anteprima e un titolo.

Lo *Scene Manager API*^[36] include le seguenti funzioni:

- *show()* e *hide()*, mostra/nasconde la scena specificata
- *focus()*, assegna il focus sulla scena specificata

CAPITOLO 4. Eclap per Smart TV: progettazione

- *get()*, restituisce l'istanza di una scena
- *getFocused()*, restituisce l'attuale *SceneID* allo stato *focus*
- *getVisible()*, restituisce se una determinata scena è visibile o no
- *returnFocus()*, il *focus* ritorna al *framework* gestore delle scene (*scene manager*).

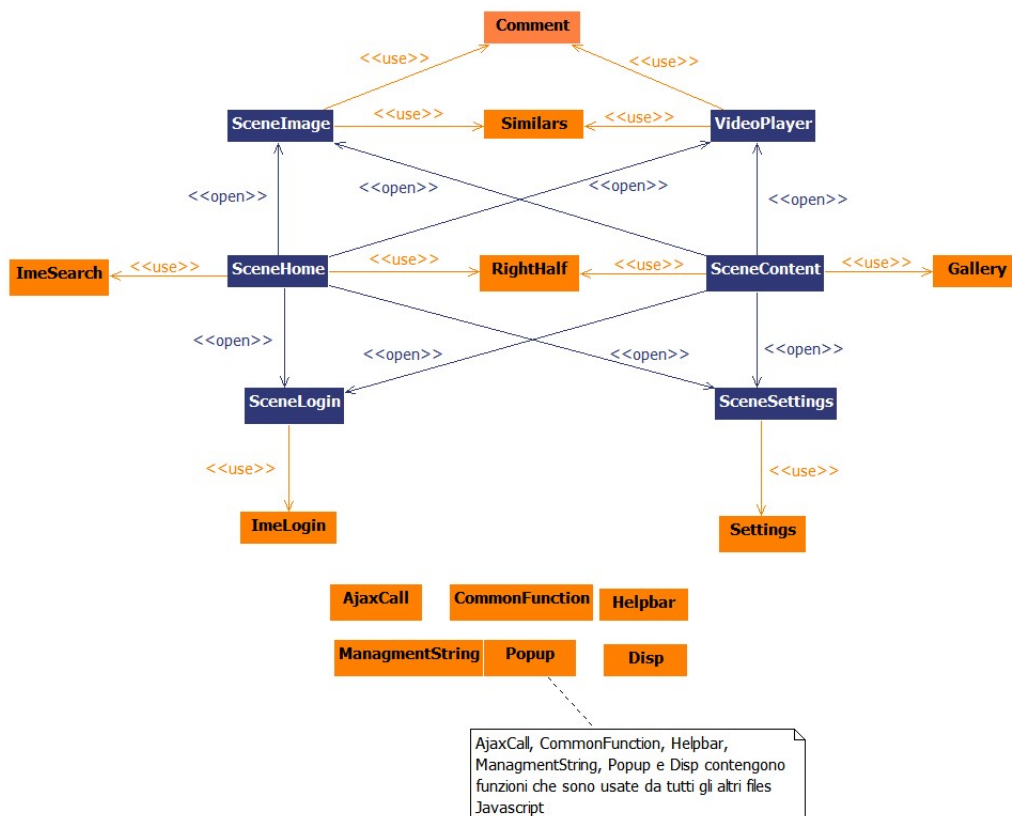


Figura 4.2: Diagramma delle classi ECLAP per Smart TV

Ogni file *Javascript* relativo a una scena presenta le seguenti funzioni, che vengono chiamate al momento dell'uso di una delle funzioni precedentemente descritte:

- *initialize()*, chiamata quando la scena è creata
- *handleShow()*, chiamata quando la scena viene mostrata tramite *show()*
- *handleHide()*, chiamata quando la scena viene nascosta tramite *hide()*

CAPITOLO 4. Eclap per Smart TV: progettazione

- *handleBlur()*, quando una scena passa allo stato *focus*, la scena precedentemente selezionata passa allo stato *blur* e quindi viene chiamata la funzione *handleBlur()*
- *handleKeyDown()*, chiamata quando un tasto del telecomando viene premuto.

I *Components UI*, facenti parte di *AppsFramework*, utilizzano la libreria *jQuery* e sono definiti come plugin nella forma “*sf_*”^[37].

In questa progettazione sono stati usati due *Components UI* facenti parte delle *API Samsung: KeyHelp UI Components*^[38] e *Loading UI Components*^[39]. Il primo serve per la gestione della helpbar presente in ogni scena, in fondo alla schermata, al fine di guidare l’utente nella navigazione e nell’uso delle varie funzionalità dell’applicazione. Il secondo viene usato per gestire l’animazione del popup di caricamento, al passaggio da una scena all’altra, soprattutto quando vengono fatte delle richieste *AJAX (jQuery)* che richiedono un po’ di tempo.

Per usare i *Components UI*, va selezionato l’elemento *HTML (<div>)* in cui si desidera creare il *Component UI* e chiamare il metodo del *plugin Component UI*. Inoltre si possono personalizzare gli stili dei *Components UI* tramite *CSS*^[37].

Il *Common Module TVKeyValue* permette di gestire in modo efficiente qualsiasi evento che si crea premendo un tasto del telecomando. Se un utente della TV preme un qualsiasi tasto del telecomando, si verifica un evento *KeyDown*. Questo evento viene passato ad un elemento, che verifica se a tale evento è stata associata una funzione nella proprietà *onkeydown*; se la risposta è positiva, tale funzione viene eseguita. L’evento viene catturato usando *event.keyCode*^[40].

L’*SDK* di *Samsung* fornisce anche un componente video, *VideoPlayer Service API*^[41], per inserire un video in un’applicazione con un’interfaccia grafica pre-impostata, ben fatta e funzionale. I contenuti possono essere riprodotti a schermo intero, parziale o entrambi (passando da un modo all’altro) e possono essere riprodotti fornendo l’URL.

Le funzioni messe a disposizione dell’utente dal modulo *sf.service.VideoPlayer*^[41] sono le seguenti:

- *init()*, inializza il *VideoPlayer*;
- *show()* e *hide()*, mostra/nasconde il *VideoPlayer*
- *focus()*, assegna il focus al *VideoPlayer*
- *play()*, *pause()* e *stop()* servono, rispettivamente, per avviare, mettere in pausa, in stop il *VideoPlayer* con le opzioni fornite

CAPITOLO 4. Eclap per Smart TV: progettazione

- *resume()*, riprende la riproduzione di un video dopo un'eventuale pausa
- *setPosition()*, imposta il *VideoPlayer* in una posizione precisa con i valori dati
- *setFullScreen()*, imposta il *VideoPlayer* a schermo intero
- *setKeyHandler()*, imposta le funzioni associate alla digitazione dei vari tasti
- *setKeyHelp()*, imposta l'helpbar del *VideoPlayer*
- *getZIndex()* e *setZIndex()* servono, rispettivamente, per restituire e impostare il *z-index* del *VideoPlayer*
- *Skip()*, serve per impostare eventuali salti durante la riproduzione.

Samsung Smart TV fornisce un *Common Module*, chiamato *IME Module*^[18], che permette di inserire del testo nelle applicazioni tramite telecomando. Il modulo, come evidenziato dal nome, utilizza *IME (Input Method Editor)*. Il modulo *IME* può gestire al massimo 256 caratteri, se questo limite viene superato il box di input mostrerà degli spazi vuoti o l'applicazione potrebbe arrestarsi in modo anomalo; per ovviare a questo problema bisogna agire in modo da non consentire l'inserimento di più di 256 caratteri all'interno del box di input, ad esempio con *CSS*^[42].

Relativamente all'interfaccia, *Samsung Smart TV* supporta un sistema multi-language attraverso il menu *OSD(on screen display)*. Per impostare la lingua bisogna agire in questo modo: sul telecomando premere il tasto *MENU* selezionare *System > Menu Language*. In questo modo l'applicazione riceve il codice della lingua selezionata attraverso *sf.core.getEnvValue('lang')* e l'*Apps Framework* carica automaticamente la risorsa della lingua. Per implementare questo sistema di gestione della lingua bisogna aggiungere la proprietà *languages* nel file *app.json*, inserendo i codici delle lingue che si voglio implementare, rispettando lo standard *ISO 639-1*. Per ogni lingua inserita si crea un file *.js* con lo stesso numero e nome di variabili^[43].

ECLAP per Smart TV supporta italiano e inglese.

4.3 Struttura directory e diagrammi di sequenza

Legenda (figura 4.3):

- *app/htmls*, contiene i files *.html* delle varie scene
- *app/scenes*, contiene i files *.js* delle varie scene

CAPITOLO 4. Eclap per Smart TV: progettazione

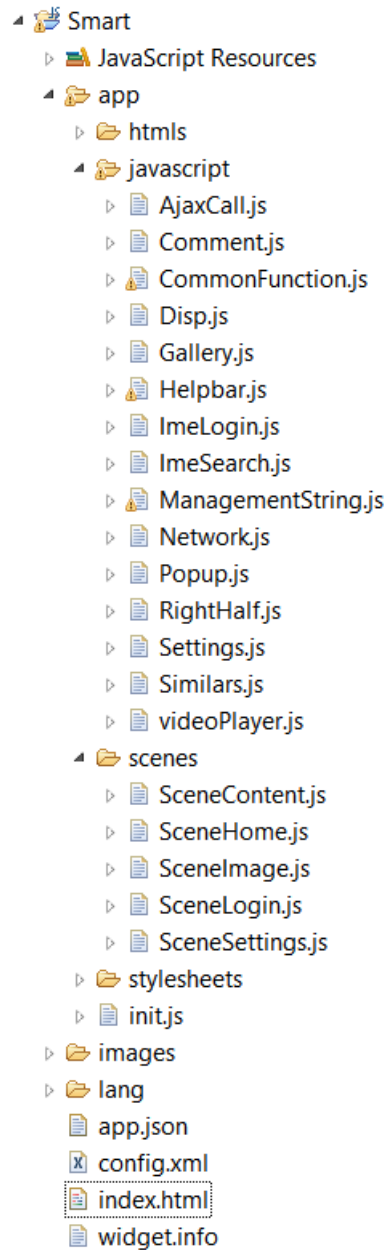


Figura 4.3: Directory ECLAP per Smart TV

CAPITOLO 4. Eclap per Smart TV: progettazione

- *app/stylesheets*, contiene i files *.css* delle varie scene
- *app/init.js*, definisce le funzioni di inizializzazione (*onStart*) e di fine (*onDestroy*)
- *app/javascript*, contiene files *Javascript*, dove ognuno ha al suo interno funzioni chiamate dai files *.js* presenti in *app/scenes*
- *images*, contiene tutte le immagini e icone relative all'applicazione
- *lang*, contiene i files *.js* relativi alla lingua dell'interfaccia
- *app.json*, descrizione dell'applicazione per l'*Apps Framework*
- *config.xml*, descrizione dell'applicazione per *Samsung Smart TV*
- *index.html*, pagina principale dell'applicazione. Tutti gli elementi facenti parte dell'*Apps Framework* sono aggiunti a questo file
- *widget.info*, si specifica la risoluzione dello schermo

Le scene rappresentano ciò che l'utente vede e con cui interagisce. La scena principale, ossia la scena che viene mostrata all'avvio dell'applicazione e che è collegata a tutte le altre è la *SceneHome*. Dopo la *SceneHome*, a livello di importanza e funzionalità offerte, viene la scena *SceneContent*, anch'essa collegata a tutte le altre. La navigazione all'interno delle varie scene si può rappresentare come un albero, dove ogni nodo rappresenta una scena. Usando questa rappresentazione i nodi foglia sono rappresentati dalle restanti scene: *SceneLogin*, *SceneSettings*, *SceneImage* e *videoPlayer*. Per l'utente *videoPlayer* rappresenta una scena vera e propria, ma in realtà si tratta di un riproduttore video che copre la scena da cui viene interpellato.

I files *Javascript*, presenti all'interno della cartella *app/scenes*, sono files relativi alle varie scene e contengono le funzioni *initialize()*, *handleShow()*, *show()*, *handleHide()*, *handleBlur()*, *handleBlur()*, *handleKeyDown()* e ulteriori funzioni sviluppate al fine del regolare funzionamento dell'applicazione. Questi files sono:

- *SceneHome.js* rappresenta la schermata principale
- *SceneContent.js* gestisce la schermata con la matrice dei contenuti
- *SceneImage.js* produce la schermata di visualizzazione delle immagini
- *SceneLogin.js* genera la schermata di login
- *SceneSettings.js* rappresenta la schermata di settings.

CAPITOLO 4. Eclap per Smart TV: progettazione

Oltre il file *.js* ogni scena presenta un file *.html* e un file *.css*, con il primo vengono istanziati alcuni elementi, in genere *div*, presenti nella scena; con il secondo si assegna a ogni elemento, precedentemente istanziato nel file *.html*, delle proprietà a seconda di come lo si vuole presentare all'utente. Le funzionalità fornite dai linguaggi *HTML* e *CSS* possono essere effettuate anche attraverso il linguaggio *Javascript*.

I files *Javascript* presenti all'interno di *app/javascript*, contengono funzioni che vengono chiamate dalle varie scene e servono per separare le varie funzionalità e, quindi, rendere più leggibile il codice. Già dai loro nomi si può intuire qual è la loro utilità:

- *AjaxCall.js* contiene un insieme di alcune richieste *AJAX* (*jQuery*)
- *Comment.js* include le funzioni relative alla gestione dei commenti
- *CommonFunction.js* presenta alcune funzioni generiche che vengono chiamate da vari files *Javascript*
- *Gallery.js* contiene diverse funzioni relative alla gestione della matrice dei contenuti mostrati in *SceneContent*
- *Helpbar.js* include funzioni che gestiscono l'*helpbar* nelle varie scene, che aiuta l'utente nella navigazione e uso dell'applicazione
- *ImeLogin.js* e *ImeSearch.js* gestiscono rispettivamente l'*IME* (*Input Method Editor*) della schermata di login e della ricerca dei contenuti tramite *keyword*
- *ManagementString.js* contiene delle funzioni che restituiscono delle stringhe usate da vari files *Javascript*
- *Network.js* serve per verificare, con una certa frequenza, che la TV sia collegata alla rete
- *Popup.js* include le funzioni relative ai vari popup che si presentano durante l'utilizzo dell'applicazione
- *RightHalf.js* e *Disp.js* gestiscono l'elenco dei canali e delle categorie visualizzate in *SceneHome* e *SceneContent*
- *Settings.js* presenta alcune funzioni relative alla schermata di settings
- *Similar.js* include le funzioni per gestire i simili di un contenuto
- *videoPlayer.js* gestisce le funzioni relative al video player.

CAPITOLO 4. Eclap per Smart TV: progettazione

Dopo aver dato un'idea generale dell'insieme dell'applicazione e dell'organizzazione dei vari files *Javascript*, si passa a definire più in dettaglio, attraverso i diagrammi di sequenza, i vari scenari che possono presentarsi durante l'uso dell'applicazione.

Uno scenario è una determinata sequenza di azioni in cui tutte le scelte sono state già effettuate; in pratica nel diagramma non compaiono scelte, né flussi alternativi. I diagrammi di sequenza descrivono le interazioni che intercorrono, in termini di messaggi, tra l'utente ed entità del sistema che si sta rappresentando. Un messaggio è un'informazione che viene scambiata tra due entità. Solitamente chi invia il messaggio, la parte attiva, è l'utente. Il messaggio che viene generato in risposta ad un precedente messaggio è detto messaggio di risposta^[44].

Riproduzione di video/immagine

Dopo la selezione di un'anteprima si visualizza il contenuto a schermo intero, ciò è rappresentato dai prossimi due diagrammi di sequenza, il primo relativo a un'immagine (figura 4.4) e l'altro a un video (figura 4.5).

Per visualizzare un contenuto a schermo intero bisogna selezionare la sua anteprima dalla scena *SceneHome* o *SceneContent* e premere il tasto *ENTER*. Ciò fa sì che venga richiesto al server, tramite una richiesta *AJAX* (*jQuery*), l'URL del contenuto, per poi creare, per quanto riguarda un'immagine, la scena *SceneImage* in cui verrà mostrata l'immagine a schermo intero dopo aver creato l'elemento che la contiene nascondendo la schermata precedentemente visualizzata.

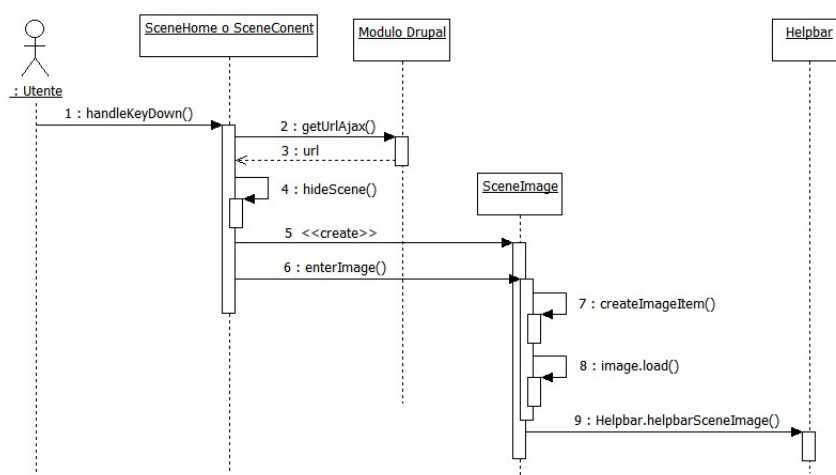


Figura 4.4: Diagramma di sequenza - Riproduzione immagine

CAPITOLO 4. Eclap per Smart TV: progettazione

Per visualizzare un video a schermo intero, la sequenza è uguale fino al passo 4 al caso dell'immagine, dopodiché le funzioni e gli oggetti chiamati in causa sono diversi. Diversamente da prima, non verrà creata nessun'altra scena ma il riproduttore di video viene avviato in modo che occupi l'intera schermata e poggia sulla scena già visualizzata attraverso la chiamata alla funzione `videoPlayer.enterVideo()`.

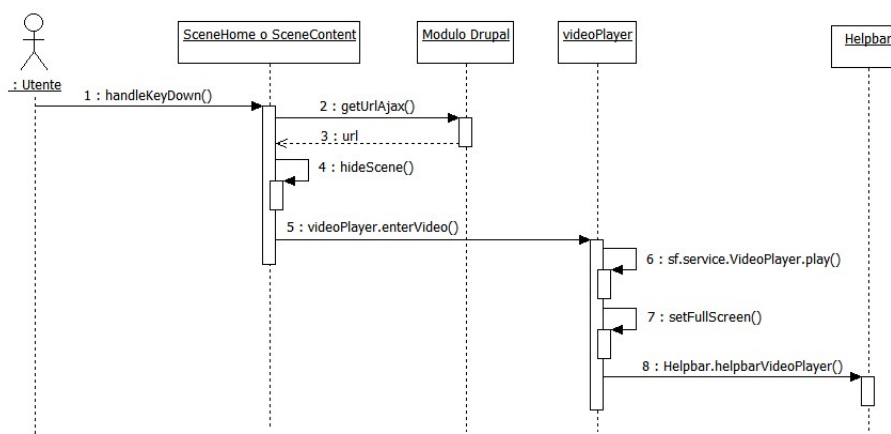


Figura 4.5: Diagramma di sequenza - Riproduzione video

Visualizzazione dei contenuti selezionando canali o categorie

Il primo diagramma di sequenza (*figura 4.6*) considerato riguarda la selezione di un canale a partire dalla scena principale e la conseguente visualizzazione della serie di anteprime dei contenuti. Il canale preso in considerazione è quello *In primo piano* e si può estendere, con lo stesso procedimento, a tutti gli altri canali e a tutte le categorie, con la sola differenza della chiamata della funzione numero 11 che cambia a seconda del *canale/categoria* selezionato/a. Nel caso si digiti una categoria, la funzione chiamata è sempre `getContent()`.

In questo caso, la funzione `handleKeyDown()` si attiva nel momento in cui l'utente preme il tasto *ENTER* dopo aver selezionato un canale o una categoria, il che ha l'effetto di nascondere la scena `SceneHome` e distruggere parte della sua schermata, in particolare tutto ciò che è contenuto nel *div* "contenitore". Viene così creata la scena `SceneContent` e, dopo aver ricevuto i dati attraverso la richiesta *AJAX* (*jQuery*) al modulo *Drupal*, viene riempita la schermata con una matrice 3x5 contenente le anteprime dei contenuti trovati e i relativi titoli con la chiamata alla funzione `Gallery.init()`.

CAPITOLO 4. Eclap per Smart TV: progettazione

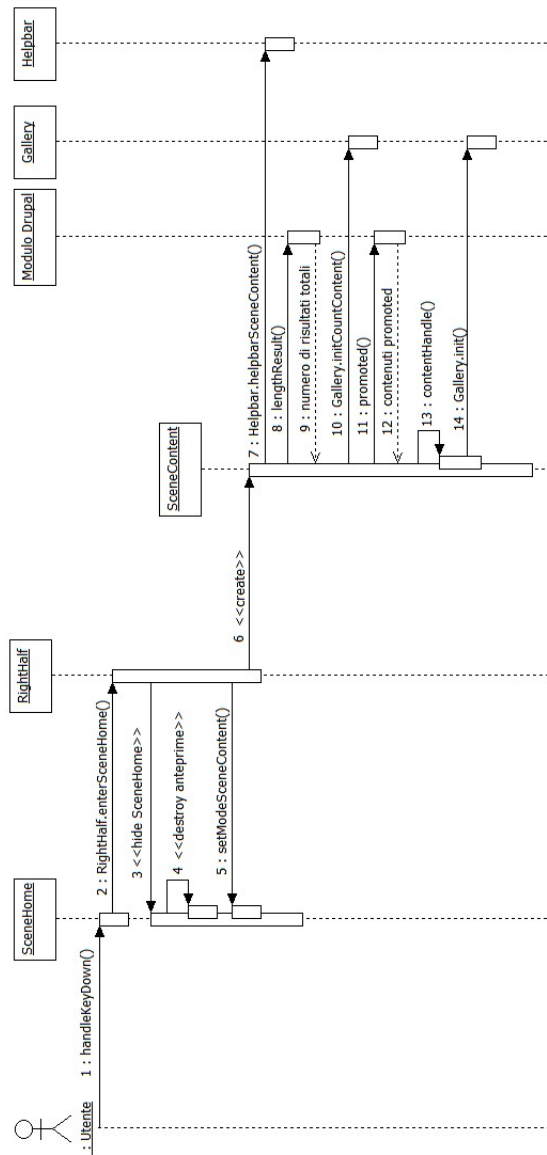


Figura 4.6: Diagramma di sequenza - Visualizzazione dei contenuti selezionando canali o categorie

Visualizzazione dei metadati di un contenuto

Il diagramma di sequenza di *figura 4.7* descrive come avviene la visualizzazione dei metadati, ipotizzando di trovarsi nella schermata principale (*SceneHome*).

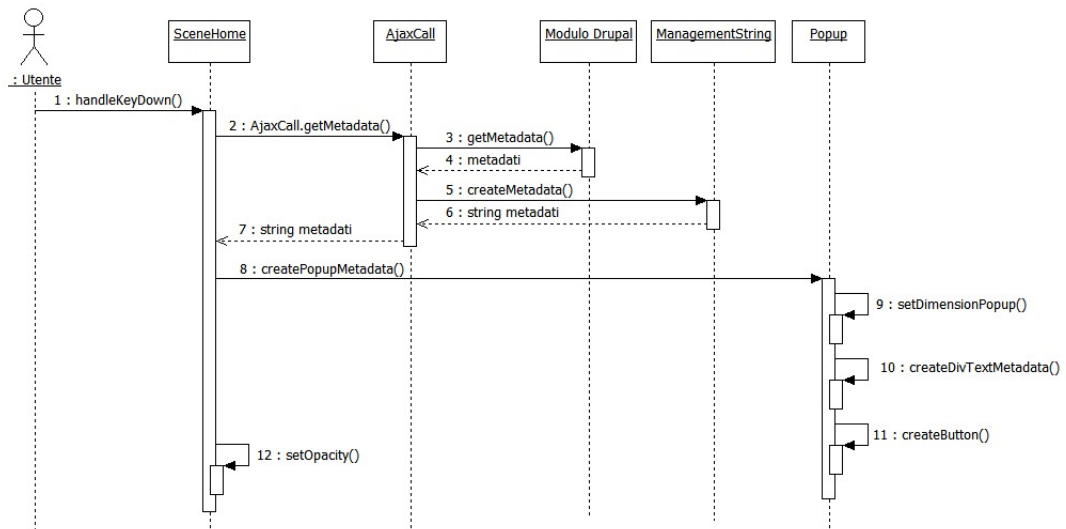


Figura 4.7: Diagramma di sequenza - Visualizzazione dei metadati di un contenuto

Il caso preso in considerazione si svolge selezionando un'anteprima dalla scena *SceneHome* e premendo il tasto *INFO*, ciò fa sì che vengano richiesti al server i metadati del contenuto selezionato. I metadati ricevuti, vengono sistemati in una stringa e viene creato il popup che contiene le informazioni aggiuntive ottenute.

Ricerca

Il diagramma di sequenza di *figura 4.8* riguarda la visualizzazione di una serie di anteprime di contenuti attraverso l'immissione da parte dell'utente di una keyword. Come si può vedere, a partire dalla creazione della *SceneContent* esiste una similarità con il diagramma precedente.

Si ipotizza che l'utente sia in una qualsiasi parte della scena *SceneHome* e il primo passo illustrato corrisponde a quando l'utente preme il tasto *B (verde)*, che attiva la funzione *handleKeyDown()* e assegna il focus al campo di input permettendo all'utente di digitare una *keyword* per la ricerca. L'utente, dopo

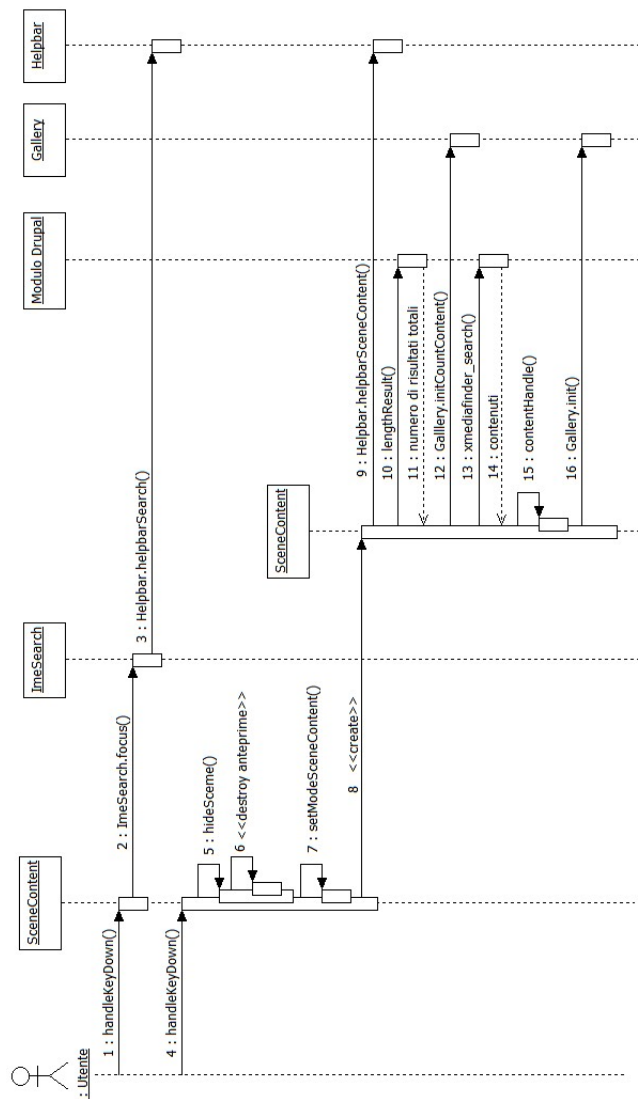


Figura 4.8: Diagramma di sequenza - Ricerca dei contenuti tramite keyword

CAPITOLO 4. Eclap per Smart TV: progettazione

aver composto la *keyword* con uno dei due metodi proposti, premendo il tasto ENTER avvia la ricerca nascondendo la scena *SceneHome*, eliminando alcuni elementi al suo interno e creando la scena *SceneContent*, che viene riempita con i risultati ottenuti dalla richiesta *AJAX* (*jQuery*) fatta al modulo *Drupal* come nel caso precedente.

Visualizzazione dei simili di un contenuto

Il successivo diagramma di sequenza (*figura 4.9*) mostra i passaggi necessari per visualizzare i contenuti simili a un dato contenuto, quando lo si sta visualizzando a schermo intero.

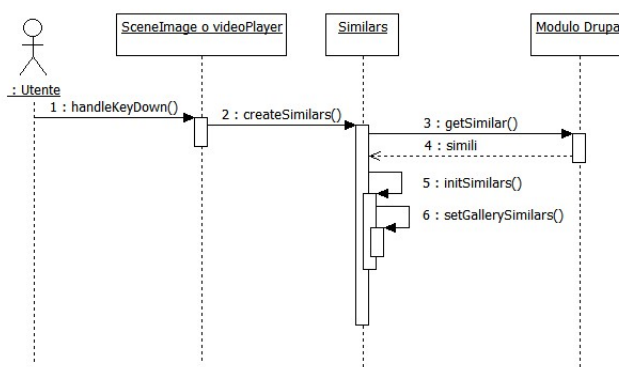


Figura 4.9: Diagramma di sequenza - Visualizzazione dei simili di un contenuto

La funzione *handleKeyDown*, in questo caso, viene interpellata quando viene premuto il tasto *DOWN*, che fa sì che venga chiamata la funzione *createSimilar()* da cui viene fatta la richiesta al server. Dopodiché chiamando le funzioni *initSimilar()* e *setGallerySimilar()* viene inizializzato il popup in cui verrà mostrato ciò che è stato richiesto.

Accesso autenticato

Nel diagramma di sequenza di *figura 4.10* si mostra come avviene il login, in particolare si assume che l'inserimento di nome utente e password siano corretti, in caso contrario al punto 12 viene chiamata la funzione *Popup.createPopupLoginFail()* con cui viene creato un popup che informa l'utente di un suo errore nell'immissione di almeno uno dei due campi e si rimane nella schermata di login.

Se l'utente non è già autenticato, premendo il tasto *C* (*giallo*) dalla scena *SceneHome* o *SceneContent*, viene nascosta la scena visualizzata e creata la scena

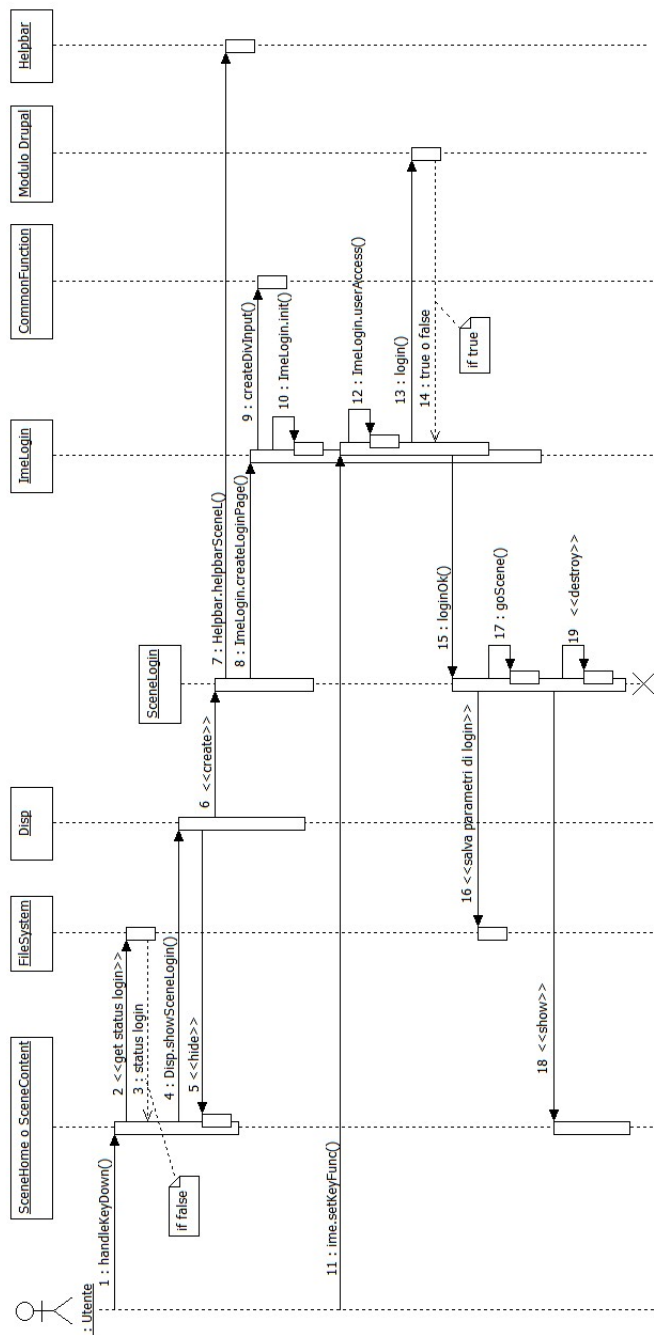


Figura 4.10: Diagramma di sequenza - Accesso autenticato

CAPITOLO 4. Eclap per Smart TV: progettazione

SceneLogin attraverso la funzione *Disp.showSceneLogin()*. Una volta che *SceneLogin* viene mostrata, vengono creati i due campi di input tramite *ImeLogin.createLoginPage()* e inizializzati con la chiamata a *ImeLogin.init()*. Dopo aver immesso il nome utente e la password, premendo il tasto *ENTER* vengono inviate le due immissioni al server per effettuare il login, che risponde con *true* o *false*. Se *false*, viene creato il popup precedentemente descritto in cui si avvisa l'utente di aver sbagliato durante la digitazione, altrimenti si distrugge *SceneLogin* e si torna alla scena precedentemente visualizzata aggiungendo le funzionalità fornite agli utenti autenticati.

Visualizzazione dei commenti di un contenuto

Il diagramma di sequenza di *figura 4.11* descrive come avviene la visualizzazione di eventuali commenti quando si sta visualizzando un contenuto a schermo intero.

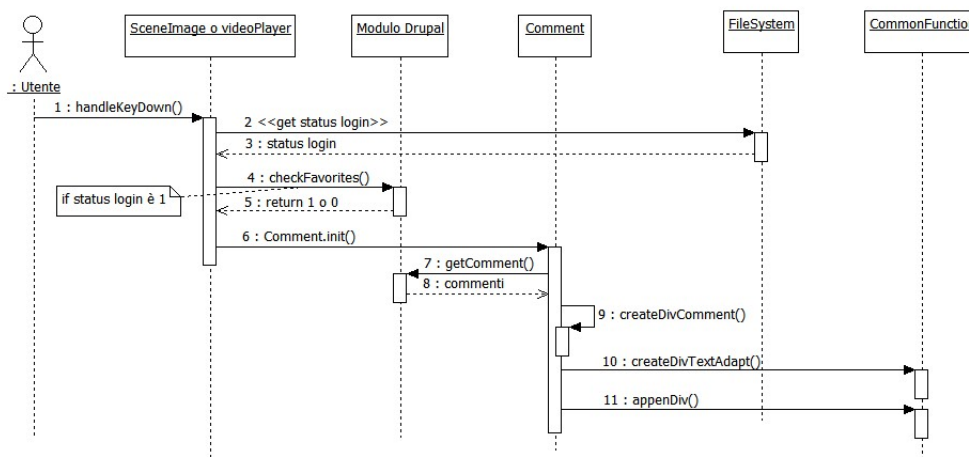


Figura 4.11: Diagramma di sequenza - Visualizzazione dei commenti di un contenuto

La funzione *handleKeyDown*, in questo caso, viene interpellata quando viene premuto il tasto *INFO*, che ha la funzione, rispetto al contenuto visualizzato, di mostrare gli eventuali commenti, di aggiungerlo ai preferiti o commentarlo. I commenti vengono visualizzati all'interno di un popup orizzontale che si attiva in basso alla schermata, appena sopra l'helpbar. Appena l'utente preme il tasto sopra citato, viene verificato se l'utente sia autenticato, in tal caso viene anche verificato se il contenuto fa già parte dei preferiti dell'utente, dopodiché inizializza il popup che contiene i commenti attraverso la funzione *Comment.init()*, che permette di visualizzare gli eventuali commenti nello spazio dedicato con indicati

la data e il nome dell'autore e, nel caso in cui il contenuto faccia già parte dei preferiti, verrà mostrata un'icona di *like*.

Commentare un contenuto

Il diagramma di sequenza di *figura 4.12* mostra i passaggi necessari per aggiungere un commento a un contenuto, ipotizzando di visualizzare il contenuto a schermo intero e il popup dei commenti.

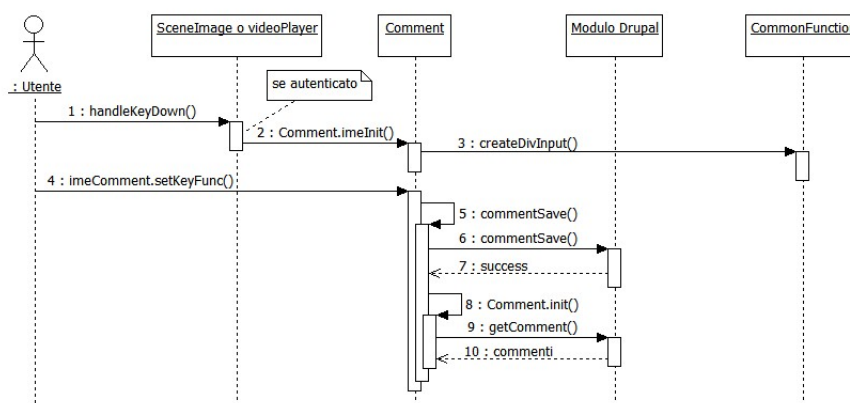


Figura 4.12: Diagramma di sequenza - Commentare un contenuto

Se l'utente è autenticato, nel momento in cui il popup dei commenti è visibile viene creato, premendo il tasto *D (blu)* o *ENTER*, il campo di input per inserire un commento attraverso la funzione *Comment.imeInit()*. Invece, se l'utente non è autenticato, chiamando al passo 2 la funzione *Popup.createPopupRetryCom()*, l'utente viene avvisato che non ha i permessi necessari per effettuare tale operazione. Dopo aver composto il commento, è possibile aggiungerlo fra gli altri premendo il tasto *ENTER* che ha l'effetto di chiamare la funzione *commentSave()* del modulo *Drupal* e di rivisualizzare la lista di commenti comprensiva di quello appena aggiunto, attraverso la funzione *Comment.init()*.

Gestione delle impostazioni

Nel diagramma di sequenza di *figura 4.13* viene illustrato il cambiamento delle impostazioni dal punto in cui l'utente accede alla schermata relativa alle impostazioni al punto in cui abbandona questo scenario dopo un eventuale cambiamento di almeno una delle tre impostazioni.

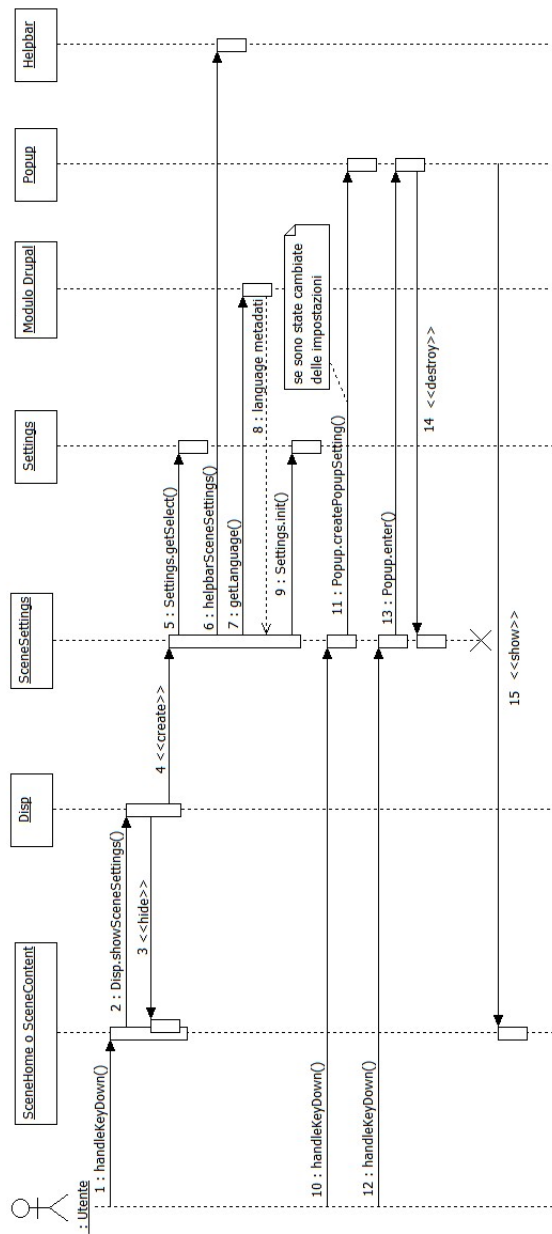


Figura 4.13: Diagramma di sequenza - Gestione delle impostazioni

Ipotizziamo che l'utente sia nella scena *SceneHome* o *SceneContent* e prema il tasto *D (blu)*, viene chiamata la funzione *Disp.showSceneSettings()* che fa sì che venga nascosta la scena visualizzata e creata la scena *SceneSettings* relativa alle impostazioni dell'applicazione, che viene inizializzata con le impostazioni precedentemente scelte. Se non viene cambiata nessuna impostazione e viene premuto il tasto *ENTER* o *RETURN*, dal passo 10 si passa direttamente al passo 14 distruggendo la scena *SceneSettings* e mostrando la scena precedente.

Invece, se è stata cambiata almeno un'impostazione viene creato un popup tramite *Popup.createPopupSettings()* per far confermare all'utente il cambiamento delle impostazioni e, sia se venga confermato o no la modifica, si distrugge la scena *SceneSettings* e viene visualizzata la scena precedente.

Aggiungere ai preferiti un contenuto

Il seguente diagramma (*figura 4.14*) mostra i passaggi necessari per aggiungere un contenuto ai preferiti, ipotizzando di visualizzare il contenuto a schermo intero e il popup dei commenti.

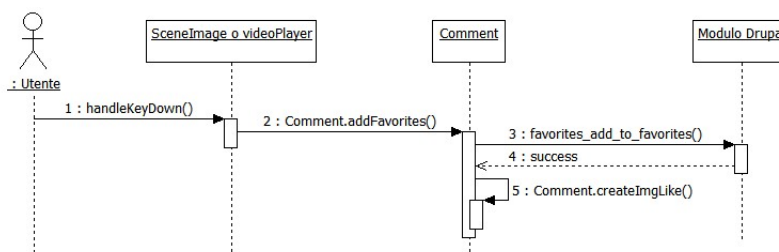


Figura 4.14: Diagramma di sequenza - Aggiungere ai preferiti un contenuto

Se il popup dei commenti risulta visibile all'utente e se l'utente si è autenticato, premendo il tasto *A (rosso)*, il contenuto visualizzato viene aggiunto ai preferiti se non già compreso, altrimenti viene rimosso. Dal diagramma si nota che viene chiamata la funzione *Comment.addFavorites()* da cui parte la richiesta di aggiungere il contenuto ai preferiti dell'utente, e se la richiesta va a buon fine viene aggiunta un'icona di *like* che identifica il contenuto come preferito. Se la richiesta non va a buon fine, al passo 5 verrà chiamata la funzione *Popup.createPopupRetryFav()*, che crea un popup che dà la possibilità all'utente di riprovare a effettuare l'operazione. Invece se l'utente non è autenticato si avvisa l'utente attraverso la funzione *Popup.createPopupFavDenied()*, che non ha i permessi necessari per effettuare tale operazione.

4.4 Comunicazione client-server: modulo Drupal

L'applicazione *ECLAP per Smart TV* installata su un dispositivo rappresenta il client, mentre il server con cui *ECLAP per Smart TV* comunica è gestito tramite *Drupal*^[45], che è un *Content Management System* modulare scritto in *PHP* e distribuito gratuitamente sotto licenza *GNU GPL*. *Drupal* funziona su diversi sistemi operativi, tra cui *Windows*, *Mac OS X*, *Linux* e qualsiasi piattaforma software che supporti i server web *Apache* (versione 1.3 o superiore) o *IIS* (versione 5 o superiore) e il linguaggio *PHP* (versione 4.3.3 o superiore). Inoltre *Drupal* utilizza un database per memorizzare i contenuti, quindi necessita di un software come *MySQL* e *PostgreSQL*, che sono gli unici *DBMS* (*Database Management System*) al momento supportati.

Il «core» di *Drupal*^[45] contiene dei propri moduli, mentre altri moduli possono essere sviluppati dalla comunità e essere scaricati e installati in un'area d'installazione di *Drupal* non destinata al «core». Il «core» di *Drupal* è stato progettato con un sistema di «hook» (ganci), che permettono ai moduli sviluppati di inserire funzioni nel processo di esecuzione di *Drupal*.

L'applicazione *ECLAP per Smart TV* sfrutta un modulo custom scritto in *PHP*, appositamente sviluppato, che permette la comunicazione dell'applicazione con il server e con il database, quest'ultima attraverso interrogazioni *SQL*.

Un modulo è formato da una cartella contenente almeno due files: uno che descrive il modulo stesso (estensione *.info*) e l'altro che contiene le funzioni e gli *hooks* (estensione *.module*).

Lo scambio di dati tra client e server si realizza tramite richieste *AJAX* (*Asynchronous Javascript e XML*), scritte in *Javascript* che permettono di chiamare le funzioni scritte in *PHP* nel file *.module* di *Drupal*. Questa funzione è resa più agevole da *jQuery*, una libreria di funzioni *Javascript*, che, con poche righe di codice, effettua svariate operazioni.

Le richieste possono essere:

- asincrona, il client invia una richiesta e continua la sua esecuzione senza attendere una risposta,
- sincrona, il client invia una richiesta e attende una risposta dal server prima di continuare la propria esecuzione.

In una richiesta, inoltre, si può inviare dei dati tramite due metodi:

- *GET*, le variabili inviate verranno appese all'URL
- *POST*, le variabili verranno inviate in modo invisibile all'utente.

CAPITOLO 4. Eclap per Smart TV: progettazione

Le richieste *AJAX* usate in *ECLAP per Smart TV* con *jQuery* hanno tutte la seguente struttura^[46]:

```
$.ajax({
  type: "POST",
  dataType: "json",
  data: input,
  async: false/true,
  timeout: number,
  crossDomain: true,
  url: URL,
  beforeSend: function(xhr) {
    var cookie = document.cookie;
    xhr.setRequestHeader('Cookie', cookie);
  },
  xhrFields: {
    withCredentials: true
  },
  success: function (data) {
    alert("chiamata a buon fine");
  },
  error: function (msg) {
    alert("chiamata fallita");
  },
});
```

dove:

- *url* è l'URL della risorsa a cui viene inviata la richiesta
- *type* è il tipo di richiesta da inviare
- *data* è una stringa del tipo "chiave=valore&chiave2=valore2&..." e contiene gli eventuali parametri inviati durante la richiesta tramite il metodo specificato in *type*
- *dataType* è il tipo di dato che si riceve
- *async* indica se la chiamata è asincrona (*true*) o sincrona (*false*)
- *cross-domain*, se impostato a *true*, permette di reindirizzare la richiesta su un altro dominio

CAPITOLO 4. Eclap per Smart TV: progettazione

- *timeout* specifica il tempo in millisecondi dopo i quali la richiesta viene considerata scaduta
- *beforeSend* rappresenta una funzione di callback che viene eseguita prima che la richiesta sia inviata
- *success* è la funzione che verrà eseguita se la richiesta avrà successo
- *error* rappresenta la funzione che verrà eseguita in caso di errore
- *xhrFields* permette di impostare una coppia *fieldName-fieldValue* sull'oggetto *XHR*, acronimo di *XMLHttpRequest*.

Ogni richiesta ha un suo messaggio di risposta, il cui formato è *JSON*. *JSON*^[47], acronimo di *Javascript Object Notation*, è un formato adatto per lo scambio di dati client-server e viene usato in *AJAX* come alternativa a *XML/XSLT*. La semplicità di *JSON* ne ha decretato un rapido utilizzo specialmente nella programmazione in *AJAX*. Il suo uso tramite *Javascript* è particolarmente semplice, tanto da renderlo velocemente molto popolare vista la diffusione della programmazione in *Javascript* nel mondo del Web.

I tipi di dati supportati dal formato *JSON*^[47] sono: valori booleani (true e false), interi, reali, virgola mobile, stringhe racchiuse da doppi apici ("), array (sequenze ordinate di valori, separati da virgole e racchiusi in parentesi quadre), array associativi (sequenze coppie chiave-valore separate da virgole racchiuse in parentesi graffe) e null.

Le URL, usate in *ECLAP per Smart TV* per far comunicare il client con il server per l'accesso alle funzionalità offerte dal modulo di *Drupal*, hanno in comune www.eclap.eu/portal/?q=en-US/ a cui segue:

- *webtv/getTaxonomy* restituisce la tassonomia dei contenuti, viene chiamata all'avvio dell'applicazione e ogni volta che si ripresenta la scena principale o quando viene cambiata la lingua dell'interfaccia. Si consideri la tassonomia come un albero, la sua altezza è il massimo fra i livelli di tutti i suoi nodi, un nodo è rappresentato da una categoria e il livello di un nodo in un albero è pari a 1 più il livello del nodo padre, intendendo che la radice ha livello 0. La chiamata restituisce un'array associativo dove la dimensione è data dall'altezza dell'albero e ogni elemento dell'array ha al suo interno altri array, ognuno dei quali ha come chiave il *tid* (codice identificativo) di una categoria. Per convenzione le categorie radice sono sistemate alla prima posizione nell'array. Nella richiesta, il parametro passato *lang* indica la lingua dell'interfaccia impostata (*eng* o *it*).

CAPITOLO 4. Eclap per Smart TV: progettazione

ESEMPIO:

Request URL: http://www.eclap.eu/portal/?q=en-US/webtv/getTaxonomy

Request Method: POST

Form data: lang=it

Response:

```
[
  0: [
    "Genere",
    "Periodo Storico",
    ...
    "Arti dello spettacolo"
  ],
  1: {
    "Genere": [ "Dramma", "Tragedia", "Commedia", ... ],
    "Periodo Storico": [ "Arcaico", "Greco", "Barocco", ... ],
    ...
    "Arti dello spettacolo": [ "Danza", "Teatro", "Altro", ... ]
  },
  2: {
    "Altro": [ "Circo", "Marionette", "Carnevale" ],
    ...
    "Danza": [ "Balletto", "Tradizionale" ],
    "Cinema e Film": [
      "Nazionale", "Internazionale", "Fumetto", ...
    ]
  }
]
```

- *webtv/getHomePromoted* e *webtv/getHomeSimilar* vengono chiamate all'avvio dell'applicazione e ogni volta che si ripresenta la scena principale. Restituiscono le anteprime dei contenuti facenti parte dell'insieme, rispettivamente, *In primo piano* e *Simili* al video in esecuzione nella schermata principale. Per ogni contenuto viene restituito l'axoid, l'URL del contenuto, il nid, la tipologia del contenuto, l'URL dell'anteprima e il titolo. Nella richiesta, il parametro passato *language* indica la lingua dei metadati impostata nel formato standard *ISO 639-1* tra quelle disponibili.

ESEMPIO:

CAPITOLO 4. Eclap per Smart TV: progettazione

Request URL: http://www.eclap.eu/portal/.../getHomePromoted

Request Method: POST

Form data: language=it

Response:

```
[
  0: {
    "axoid": "urn:axmedis:00000:obj...09a90b2",
    "url": "http://www.eclap.eu/axmedis/...-md.flv",
    "nid": "65132",
    "iconurl": "http://bpfe.eclap.eu/eclap/gif/...09a90b2.gif",
    "type": "video",
    "title": "Conferenza ECLAP. Camera laterale palco"
  },
  ...
]
```

- *webtv/promoted*, *webtv/lastPosted*, *webtv/mostDownloaded*, *webtv/mostRanked* vengono chiamate quando si selezionano i canali accessibili a ogni utente e restituiscono, rispettivamente, le anteprime dei contenuti relativi ai canali *In primo piano*, *Più visti*, *Ultimo caricati* e *Più votati*. Per ogni contenuto viene restituito l'axoid, il nid, la tipologia del contenuto, l'URL dell'anteprima e il titolo. Nella richiesta, i parametri passati *language*, *type*, *page* e *contentForPage* indicano, rispettivamente, la lingua dei metadati imposta nel formato standard *ISO 639-1* tra quelle disponibili, il formato di contenuto da restituire (*image*, *video* o *any types*), il numero della pagina richiesta e il numero massimo di contenuti per pagina da restituire.

ESEMPIO:

Request URL: http://www.eclap.eu/portal/?q=en-US/webtv/promoted

Request Method: POST

Form data: language=it&type=any types&page=0&contentForPage=15

Response:

```
[
  0: {
    "axoid": "urn:axmedis:00000:obj...ad6f5",
    "nid": "3870",
    "iconurl": "http://bpfe.eclap.eu/eclap/gif/...8ad6f5.gif",
    "type": "video",
  }
]
```

CAPITOLO 4. Eclap per Smart TV: progettazione

```
    "title": "Hi-Res"  
  },  
  ...  
  14: { ... }  
]
```

- *webtv/lengthResult* restituisce un numero intero, che rappresenta il numero delle pagine contenenti la matrice dei contenuti in *SceneContent*. Nella richiesta, i parametri passati *typeRequest*, *type*, *category* e *search* indicano, rispettivamente, il tipo di richiesta effettuata indicata con numeri interi (*0 - featured*, *1 - mostDownloaded*, *2 - lastPosted*, *3 - mostRanked*, *4 - category*, *6 - search*, *7 - lastViewed*, *8 - lastByFriend* o *9 - myFavorites*), il formato di contenuto da restituire (*image*, *video* o *any types*), il nome della categoria di cui si cercano i contenuti (usata se viene selezionata una categoria) e la keyword della ricerca immessa (usata se viene effettuata una ricerca dei contenuti tramite keyword).

ESEMPIO:

Request URL: <http://www.eclap.eu/portal/?q=en-US/webtv/lengthResult>

Request Method: POST

Form data: typeRequest=4&type=any types&category=Jazz&search="

Response: "28"

- *webtv/getFavorites*, *webtv/lastViewed* e *webtv/lastByFriend* vengono chiamate quando si selezionano i canali accessibili solo ad utenti autenticati e restituiscono, rispettivamente le anteprime dei contenuti relativi ai canali *Miei Preferiti*, *Ultimi visti* e *Visti dagli amici*. Per ogni contenuto viene restituito l'axoid, il nid, la tipologia del contenuto, l'URL dell'anteprima e il titolo. Nella richiesta, i parametri passati *language*, *type*, *page* e *contentForPage* indicano, rispettivamente, la lingua dei metadati impostata nel formato standard *ISO 639-1* tra quelle disponibili, il formato di contenuto da restituire (*image*, *video* o *any types*), il numero della pagina richiesta e il numero massimo di contenuti per pagina da restituire.

ESEMPIO:

Request URL: <http://www.eclap.eu/portal/?q=en-US/webtv/getFavorites>

Request Method: POST

Form data: language=it&type=any types&page=0&contentForPage=15

CAPITOLO 4. Eclap per Smart TV: progettazione

Response:

```
[
  0:{
    "axoid": "urn:axmedis:00000:obj...ad6f5",
    "nid": "3870",
    "iconurl": "http://bpfe.eclap.eu/eclap/gif/...8ad6f5.gif",
    "type": "video",
    "title": "Hi-Res"
  },
  ...
  14: { ... }
]
```

- *webtv/getContent* viene chiamata quando una categoria, che non ha figli, viene selezionata. Restituisce un insieme di anteprime dei contenuti relativi alla categoria selezionata. Per ogni contenuto viene restituito l'axoid, il nid, la tipologia del contenuto, l'URL dell'anteprima e il titolo. Nella richiesta, i parametri passati *category*, *parent*, *language*, *type*, *page*, *contentForPage*, e *langInterface* indicano, rispettivamente, il nome della categoria di cui si cercano i contenuti, il relativo nome della categoria madre, la lingua dei metadati impostata nel formato standard *ISO 639-1* tra quelle disponibili, il formato di contenuto da restituire (*image*, *video* o *any types*), il numero della pagina richiesta, il numero massimo di contenuti per pagina da restituire e la lingua dell'interfaccia impostata (*eng* o *it*).

ESEMPIO:

Request URL: <http://www.eclap.eu/portal/?q=en-US/webtv/getContent>

Request Method: POST

Form data: `category=comico&language=it&type=any types&page=0&contentForPage=15&parent=genere&langInterface=it`

Response:

```
[
  0:{
    "axoid": "urn:axmedis...d6456",
    "nid": "3531",
    "iconurl": "http://bpfe.eclap.eu/eclap/gif/...d6456.gif",
    "type": "video",
    "title": "Briachi (Tuscae Gentes)"
  },
  ...
]
```

CAPITOLO 4. Eclap per Smart TV: progettazione

```
    ...  
    14: { ... }  
  ]
```

- *webtv/searchbase* viene chiamata quando si effettua una ricerca dei contenuti tramite *keyword* e restituisce le anteprime relative ai contenuti che soddisfano i parametri di ricerca. Per ogni contenuto viene restituito l'axoid, il nid, la tipologia del contenuto, l'URL dell'anteprima e il titolo. I parametri passati *search*, *language*, *type*, *page* e *contentForPage* indicano, rispettivamente, la keyword della ricerca immessa, la lingua dei metadati impostata nel formato standard *ISO 639-1* tra quelle disponibili, il formato di contenuto da restituire (*image*, *video* o *any types*), il numero della pagina richiesta e il numero massimo di contenuti per pagina da restituire.

ESEMPIO:

Request URL: `http://www.eclap.eu/portal/?q=en-US/webtv/searchbase`

Request Method: POST

Form data: `search=dario fo&language=it&type=image&page=0&content
ForPage=15`

Response:

```
[  
  0: {  
    "axoid": "urn:axmedis:00000:obj...2 b3ce63e514d",  
    "nid": "41704",  
    "iconurl": "http://bpfe.eclap.eu/eclap/gif/...2 b3ce63e514d  
    .gif",  
    "type": "image",  
    "title": "Abbraccio tra Dario Fo e Franca Rame"  
  },  
  ...  
  14: { ... }  
]
```

- *webtv/similar* permette di ottenere i contenuti simili di un contenuto dato. Per ogni contenuto viene restituito l'axoid, il nid, la tipologia del contenuto, l'URL dell'anteprima e il titolo. Nella richiesta, i parametri passati *nid*, *count* e *language* indicano, rispettivamente, il numero identificativo del contenuto di cui cercare contenuti simili, il numero massimo di contenuti simili da restituire e la lingua dei metadati impostata nel formato standard *ISO 639-1* tra quelle disponibili.

CAPITOLO 4. Eclap per Smart TV: progettazione

ESEMPIO:

Request URL: http://www.eclap.eu/portal/?q=en-US/webtv/similar

Request Method: POST

Form data: nid=4131&count=200&language=it

Response:

```
[
  0: {
    "axoid": "urn:axmedis:00000:obj...22205df",
    "nid": "4126",
    "iconurl": "http://bpfe.eclap.eu/eclap/gif/0/...22205df.gif",
    "type": "video",
    "title": "Il primo miracolo di Gesù Bambino – camera frontale"
  },
  ...
  92: { ... }
]
```

- *webtv/getUrl* viene chiamata quando si vuole vedere un contenuto a tutto schermo. Restituisce l'URL di un dato contenuto. Nel caso in cui il contenuto non sia accessibile all'utente che ha effettuato la richiesta, perché non autenticato o non iscritto a un determinato gruppo, la richiesta restituirà la stringa "access denied". Nella richiesta, i parametri passati *nid*, *axoid*, *format*, *section* e *searchOrder* indicano, rispettivamente, il numero identificativo e l'axoid a del contenuto di cui viene richiesta l'URL, il formato del contenuto (*image*, *video* o *any types*), il canale o la categoria di appartenenza del contenuto (*searchbase*, *category_nomeCategoria*, *channel_myFavorite*, *channel_lastViewed*, *channel_promoted*, *channel_lastPosted*, *channel_mostDownloaded*, *channel_mostRanked* o *channel_lastByFriend*) e il numero della pagina di accesso al contenuto.

ESEMPIO:

Request URL: http://www.eclap.eu/portal/?q=en-US/webtv/getUrl

Request Method: POST

Form data: nid=95874&axoid=urn:axmedis:00000:obj...3b522de1ae&for-

CAPITOLO 4. Eclap per Smart TV: progettazione

mat=video§ion=channel_promoted&searchOrder=0

Response: <http://www.eclap.eu/axmedis/567/000...3b522de1ae-md.flv>

- *webtv/getMetadata* serve per ottenere i metadati relativi a un contenuto nella lingua impostata, se disponibile, altrimenti nella lingua di default del contenuto, che è invece sempre disponibile. Nella richiesta, i parametri passati *axoid* e *language* indicano, rispettivamente l'axoid del contenuto e la lingua dell'interfaccia impostata (*eng* o *it*).

ESEMPIO:

Request URL: <http://www.eclap.eu/portal/?q=en-US/webtv/getMetadata>

Request Method: POST

Form data: axoid=urn:axmedis:00000:obj....16bc85457010&language=it

Response:

```
[
  {"creator": { "language": "it", "it": "cta Uniroma" }},
  {"description": {
    "language": "it", "it": "Prologo ... Torino, 1987."
  }},
  {"subject": {
    "language": "it", "it": "Prologo ... 1980, pp. 63-120."
  }},
  {"title": {
    "language": "it", "it": "Il primo miracolo di ... sinistra"
  }},
  {"rights": {
    "language": "it", "it": "Ferruc ... Università di Roma"
  }},
  {"source": { "language": "it", "it": "Umatic" }},
  {"date": "7 dicembre, 1985"},
  {"httpdownload": "143"},
  {"ranking": "0"},
  {"duration": "00:06:08.1"},
  {"publish": "CTA-UNIROMA"},
  {"deflanguage": "it"},
  {"format": "video"},
  {"timestamp": "1307549194226"},
]
```

- *webtv/commentSave* permette di salvare un commento a un contenuto, scritto da un utente autenticato. Restituisce l'*ID* del commento nel caso l'operazione sia stata effettuata, *false* se l'operazione non è andata a buon

CAPITOLO 4. Eclap per Smart TV: progettazione

fine. Nella richiesta, i parametri passati *nid* e *text* indicano, rispettivamente, il numero identificativo del contenuto oggetto di commento e il testo del commento.

ESEMPIO:

Request URL: `http://www.eclap.eu/portal/?q=en-US/webtv/commentSave`

Request Method: POST

Form data: `nid=3870&text=very nice`

Response: 369

- *webtv/checkFavorites* verifica se un certo contenuto fa parte dell'insieme dei contenuti preferiti di un utente autenticato. Restituisce il numero intero *1* se è un contenuto preferito, *0* se non lo è. Nella richiesta, il parametro passato *nid* indica il numero identificativo del contenuto.

ESEMPIO:

Request URL: `http://www.eclap.eu/portal/?q=en-US/webtv/checkFavorites`

Request Method: POST

Form data: `nid=3870`

Response: 1

- *webtv/addToFavorites* e *webtv/removeFavorites* servono, se l'utente è autenticato, per aggiungere o rimuovere ai/dai propri preferiti un certo contenuto. La loro chiamata comporta la restituzione di un numero intero, *1* se l'operazione è andata a buon fine, *2* in caso contrario. Nella richiesta, il parametro passato *nid* indica il numero identificativo del contenuto.

ESEMPIO:

Request URL: `http://www.eclap.eu/portal/?q=en-US/webtv/addToFavorites`

Request Method: POST

Form data: `nid=3870`

Response: 1

- *webtv/login* serve effettuare il login di un utente dopo che ha immesso i due parametri nome utente e password. Restituisce il valore booleano *true* se il login è andato a buon fine, *false* in caso contrario. Nella richiesta, i parametri passati *u* e *p* indicano, rispettivamente, il nome utente e la password immessa.

CAPITOLO 4. Eclap per Smart TV: progettazione

ESEMPIO:

Request URL: http://www.eclap.eu/portal/?q=en-US/webtv/login

Request Method: POST

Form data: u=username&p=password

Response: [true]

- *webtv/logoutSmartTV*, serve per effettuare il logout di un dato utente e restituisce un numero intero, 0 se il logout è andato a buon fine, un valore diverso da 0 in caso contrario.

ESEMPIO:

Request URL: http://www.eclap.eu/portal/?q=en-US/webtv/logoutSmartTV

Request Method: POST

Response: 0

- *webtv/checkCategorySonArr* verifica se un array di categorie ha figli. Restituisce un array con il valore *true*, se la categoria in quella posizione dell'array ha figli, *false* in caso contrario. Nella richiesta, i parametri passati *lang*, *name* e *parent* indicano, rispettivamente, la lingua dell'interfaccia impostata (*eng* o *it*), il nome delle categorie di cui si vuole verificare l'esistenza di categorie figlie e il nome delle categorie madri.

ESEMPIO:

Request URL: http://www.eclap.eu/portal/.../checkCategorySonArr

Request Method: POST

Form data:

lang: it

name: ["Arti dello spettacolo", "Genere", ..., "Soggetto"]

parent: [0, 0, ..., 0]

Response: [true, true, ..., true]

- *webtv/getLanguage* restituisce tutte le lingue relative ai metadati di un contenuto. Non tutti i contenuti hanno i metadati tradotti in tutte le lingue. Restituisce un array contenente le lingue disponibili nel formato standard *ISO 639-1/ISO 639-2*.

ESEMPIO:

Request URL: http://www.eclap.eu/portal/?q=en-US/webtv/getLanguage

Request Method: POST

Response:

```
[
  "hu", "el", "fr", "sl", "ca", "nl", "pl", "en", "es",
  "it", "de", "pt", "da", "ita", "slo"
]
```

4.5 Database gestito da MySQL

MySQL è un sistema di gestione di un database relazionale composto da un client con interfaccia a riga di comando e un server.

In informatica *SQL*^[48] (*Structured Query Language*) è un linguaggio standardizzato per database basati sul modello relazionale progettato per: creare e modificare schemi di database (*DDL - Data Definition Language*), inserire, modificare e gestire dati memorizzati (*DML - Data Manipulation Language*), interrogare i dati memorizzati (*DQL - Data Query Language*), creare e gestire strumenti di controllo ed accesso ai dati (*DCL - Data Control Language*). Nonostante il nome, non si tratta dunque solo di un semplice linguaggio di interrogazione, ma si occupa anche della creazione, della gestione e dell'amministrazione del database.

Per quanto riguarda le categorie vengono coinvolte le seguenti tabelle:

drupal.term_data, *drupal.term_hierarchy*, *drupal.term_node*, *cache_index.taxonomy_language*.

La tabella *term_data* contiene, tra quelli utilizzati per la realizzazione di questa applicazione *ECLAP per Smart TV*, gli attributi: *tid* (identificativo della categoria), *vid* (identificativo del numero di revisione) e *name* (nome della categoria in inglese). *Term_hierarchy* contiene gli attributi *tid* (identificativo della categoria) e *parent* (identificativo della categoria madre). *Term_node* è formato dagli attributi *nid* (identificativo di un nodo) e *tid* (identificativo categoria).

È possibile ottenere la traduzione del nome di una categoria in un'altra lingua utilizzando la tabella *taxonomy_language*, che contiene la traduzione di ogni categoria in italiano, inglese, sloveno, olandese, francese, danese, tedesco, greco, ungherese, polacco, portoghese, spagnolo e catalano.

CAPITOLO 4. Eclap per Smart TV: progettazione

In *Drupal* è possibile aggiungere una grande varietà di contenuti di diverso tipo, che hanno una caratteristica comune, quella di essere dei nodi. In termini di programmazione ad oggetti, si può affermare che tutti i tipi di contenuto ereditano le caratteristiche comuni dalla “classe” nodo, che rappresenta l’astrazione del contenuto.

Ogni nodo può essere identificato univocamente dall’attributo *nid* o *axoid* e l’associazione a un contenuto di questi due attributi è memorizzata nella tabella *drupal.axmedis_obj*.

Per avere informazioni più dettagliate dei vari nodi si devono consultare le seguenti tabelle: *axdbv4.rootobjectsinfo* e *axdbv4.dcmi*.

La tabella *rootobjectsinfo* contiene i seguenti attributi: *rootObjectAxoid* (axoid del nodo), *httpdownload* (numero di volte che il contenuto viene visto da un utente), *publisher* (nome di chi pubblica il nodo), *ranking* (numero di voti ricevuti), *filesecco* (estensione del contenuto), *resolution* (risoluzione H-L), *dx* e *dy* (eventuali dimensioni del contenuto), *durata* (eventuale durata del contenuto), *sourcefile* (URL contenuto), *deflanguage* (lingua di default) e *format* (tipologia del contenuto).

La tabella *dcmi* serve per memorizzare tutte le informazioni aggiuntive relative ai vari nodi come titolo, descrizione, soggetto, data di creazione, creatore, editore, etc. Gli attributi di tale tabella sono: *id* (numero di identificazione), *axoid* (axoid del nodo a cui si deve associare l’informazione aggiuntiva), *termRefinement* (tipologia dell’informazione aggiuntiva), *termValue* (contenuto dell’informazione aggiuntiva) e *language* (lingua in cui è scritta l’informazione aggiuntiva).

Per la gestione dei commenti viene usata la tabella *drupal.comments* con i seguenti campi: *cid* (numero identificativo del commento), *pid* (cid del commento padre), *nid* (nid a cui è associato il commento), *uid* (numero identificativo di chi ha effettuato il commento), *subject* (soggetto del commento), *comment* (testo del commento), *hostname* (ip da cui viene effettuato il commento), *timestamp* (data in cui viene effettuato il commento) e *name* (nome dell’utente che ha commentato).

Per la gestione dei preferiti vengono coinvolte le tabelle *drupal.user_favorites* e *drupal.user_favorites_log*, la prima è formata dai campi *uid* (numero identificativo di un utente), *nid* (numero identificativo di un contenuto) e *timestamp* (data in cui viene aggiunto ai preferiti il contenuto); la seconda memorizza tutte le aggiunte/rimozioni ai/dai preferiti di un contenuto, per far ciò la tabella contiene gli attributi *uid*, *nid*, *timestamp* e *action* (mark/unmark).

La tabella *drupal.user_promoted* memorizza tutti i contenuti appartenenti all’insieme Promoted, ogni contenuto è identificato dalla colonna *nid*.

CAPITOLO 4. Eclap per Smart TV: progettazione

Con le tabelle *querylog.smarttv_queryrequest* e *querylog.smarttv_activity*, create appositamente per questa applicazione, vengono memorizzate le attività che effettua l'utente. In particolare, con la tabella *smarttv_queryrequest* è possibile tenere traccia di canali o categorie che vengono selezionati; i relativi campi sono: *id* (numero identificativo della richiesta), *uid* (numero identificativo dell'utente), *type_request* (nome del canale/categoria selezionato/a), *tid* (numero identificativo della categoria), *search_order* (numero di pagina richiesta), *ip* (ip da cui viene fatta la richiesta), *agent* (user agent), *timestamp* (data in cui viene fatta la richiesta), *session_id* (id della sessione in corso) e *category* (0 se è stato selezionato un canale, 1 se è stata selezionata una categoria). Invece con la tabella *smarttv_activity* è possibile registrare le attività fatte dagli utenti come il login, logout, aggiunta/rimozione di un contenuto dai preferiti, richiesta dei commenti e aggiunta di un commento; per far ciò i campi usati sono: *id* (numero identificativo dell'attività registrata), *uid* (numero identificativo dell'utente che effettua tale attività), *type_activity* (il nome dell'attività registrata), *nid* (nid del contenuto su cui viene effettuata l'attività), *ip* (ip da cui viene effettuata l'attività), *agent* (user agent), *timestamp* (data in cui viene effettuata l'attività), *session_id* (id delle sessione in corso) e *text_comment* (testo del commento).

Capitolo 5

Eclap per Smart TV: validazione e uso

L'applicazione è stata testata con buoni risultati sulla *TV Samsung....(modello 2013)*

Il test su TV è risultato necessario per verificare l'efficienza delle varie funzioni offerte dall'applicazione, soprattutto quelle che non sono compatibili con l'emulatore fornito da *Samsung*, come il salvataggio di dati in locale sulla TV, l'uso prolungato dell'immissione di lettere all'interno dei campi di input e l'uso prolungato dell'applicazione.

Di seguito si analizzano e descrivono diversi scenari possibili durante l'uso dell'applicazione con allegati i relativi screenshot.

5.1 Screenshot e loro descrizione

All'avvio, l'applicazione mostra un video in esecuzione appartenente al canale *In primo piano* scelto casualmente e alcune anteprime di contenuti scelti in maniera casuale a cui si può accedere direttamente tramite il tasto *ENTER* (*figura 5.1*). Di queste anteprime, sei fanno parte del canale *In primo piano* e tre sono *Simili* a quello in esecuzione. È presente anche il campo di ricerca di contenuti tramite *keyword*. Inizialmente il focus è sul contenuto in basso a sinistra, evidenziato tramite un bordo arancione. Il cambio del focus sui vari elementi presenti sulla schermata è effettuato attraverso le frecce *LEFT/RIGHT* e *UP/DOWN*. Ad esempio, partendo dalla posizione iniziale (*figura 5.1*), tra i tasti *Navigazione* l'unico che ha qualche effetto, come si può intuire, è il tasto *RIGHT*. Supponendo che si voglia passare alla selezione di un'anteprima *Simile*, si deve procedere con il tasto *RIGHT* fino a quando si raggiunge la posizione che permette l'utilizzo del

CAPITOLO 5. Eclap per Smart TV: validazione e uso



Figura 5.1: Screenshot - Home

tasto *UP*. Viceversa, dalle anteprime *Simili* è possibile passare alle anteprime *In primo piano* premendo il tasto *DOWN*.



Figura 5.2: Screenshot - Passaggio ai canali (Prima)

CAPITOLO 5. Eclap per Smart TV: validazione e uso



Figura 5.3: Screenshot - Passaggio ai canali (Dopo)

Ipotizzando di avere il focus su una delle due anteprime più vicine ai canali o categorie, è possibile selezionare i canali o le categorie premendo il tasto *RIGHT*. Viceversa, se il focus è assegnato a un canale o una categoria, premendo il tasto *LEFT*, viene selezionata un'anteprima (figura 5.2-5.3-5.4-5.5).



Figura 5.4: Screenshot - Passaggio alle categorie (Prima)

CAPITOLO 5. Eclap per Smart TV: validazione e uso



Figura 5.5: Screenshot - Passaggio alle categorie (Dopo)

Selezione di un canale o di una categoria

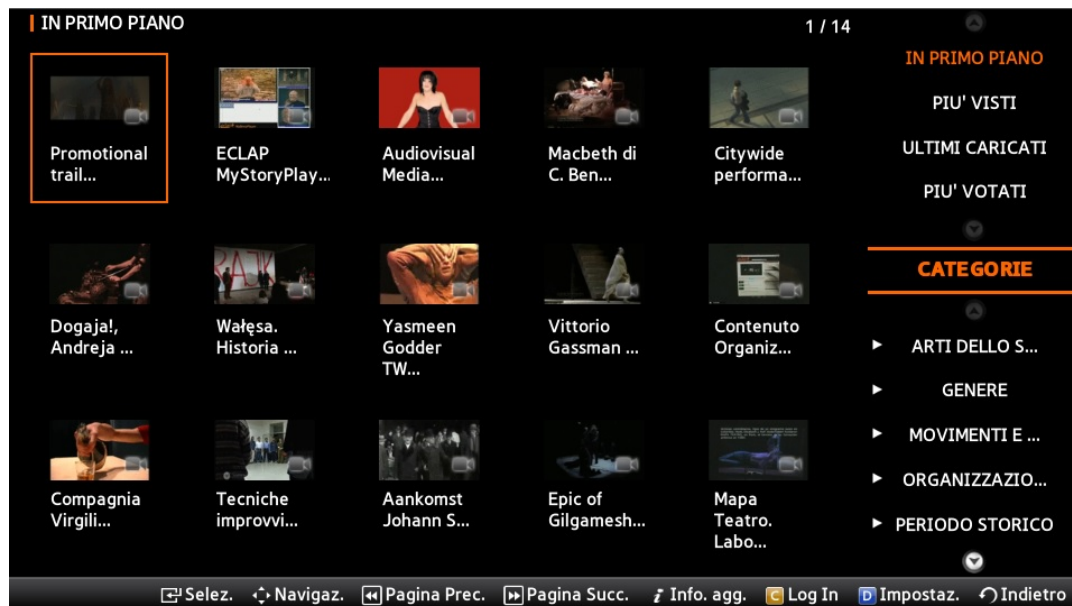


Figura 5.6: Screenshot - Schermata della scena Content

Se, dopo la selezione di un elemento, che sia un canale o una categoria, si preme il tasto *ENTER* si passa alla schermata di visualizzazione delle anteprime

CAPITOLO 5. Eclap per Smart TV: validazione e uso

dei contenuti, che comprende più pagine, dove ogni pagina è costituita da una matrice 3x5 (*figura 5.6*). La navigazione all'interno di questa schermata avviene attraverso le frecce *UP/DOWN* e *LEFT/RIGHT*.

Ipotizzando di avere il focus su una delle tre anteprime più vicine ai canali o categorie, è possibile selezionare i canali o le categorie premendo il tasto *RIGHT*. Viceversa, se il focus è assegnato a un canale o una categoria, premendo il tasto *LEFT*, viene selezionata un'anteprima.

Helpbar

In ogni schermata è presente l'helpbar (*figura 5.7*), una barra orizzontale in basso alla schermata, che aiuta l'utente a sfruttare tutte le funzionalità che l'applicazione offre, ossia guida l'utente durante l'uso dell'applicazione in modo da facilitare la sua esperienza.



Figura 5.7: Screenshot - Helpbar

Ipotizzando di trovarsi nella schermata principale, è possibile utilizzare una serie di tasti colorati, il tasto *A* (*rosso*) permette di vedere a schermo intero il video in esecuzione, il tasto *B* (*verde*) permette di immettere una *keyword* per la ricerca di contenuti collegati alla *keyword* immessa, il tasto *C* (*giallo*) consente di effettuare il login passando alla relativa schermata e il tasto *D* (*blu*) consente di cambiare le impostazioni dell'applicazione. Le funzionalità dovute ai tasti *C* e *D* sono accessibili anche dalla schermata contenente la serie di anteprime.

Ricerca

Allo scenario relativo alla ricerca dei contenuti tramite *keyword* si accede dalla schermata principale premendo il tasto *B* (*verde*), che assegna il focus al campo di input dedicato all'inserimento di una *keyword* per la ricerca.

Dalle *figure 5.8* e *5.9* si può vedere che si possono usare, tra quelli integrati all'applicazione, due sistemi di scrittura scegliendoli con il tasto *B* (*verde*). Il primo usando i numeri del telecomando a cui sono associati tre o quattro lettere ognuno; il secondo usando una tastiera virtuale all'interno della quale ci si può muovere e selezionare la lettera che si vuole digitare.

Dopo aver digitato la *keyword* desiderata, nell'esempio sottostante "DARIO FO", premendo il tasto ENTER, verranno visualizzati i risultati correlati alla

CAPITOLO 5. Eclap per Smart TV: validazione e uso



Figura 5.8: Screenshot - Ricerca tramite tastierino numerico

keyword immessa nella schermata *SceneContent* (schermata matrice anteprime) vista precedentemente (figura 5.10).

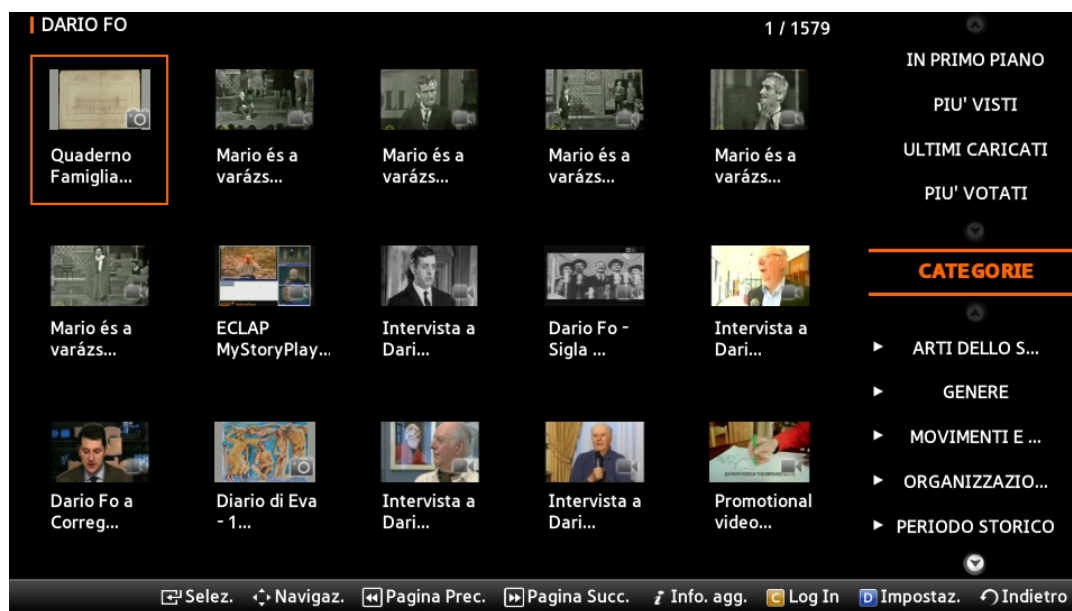


Figura 5.10: Screenshot - Ricerca con la keyword "dario fo"

CAPITOLO 5. Eclap per Smart TV: validazione e uso



Figura 5.9: Screenshot - Ricerca tramite tastiera virtuale

Riproduzione video/immagini

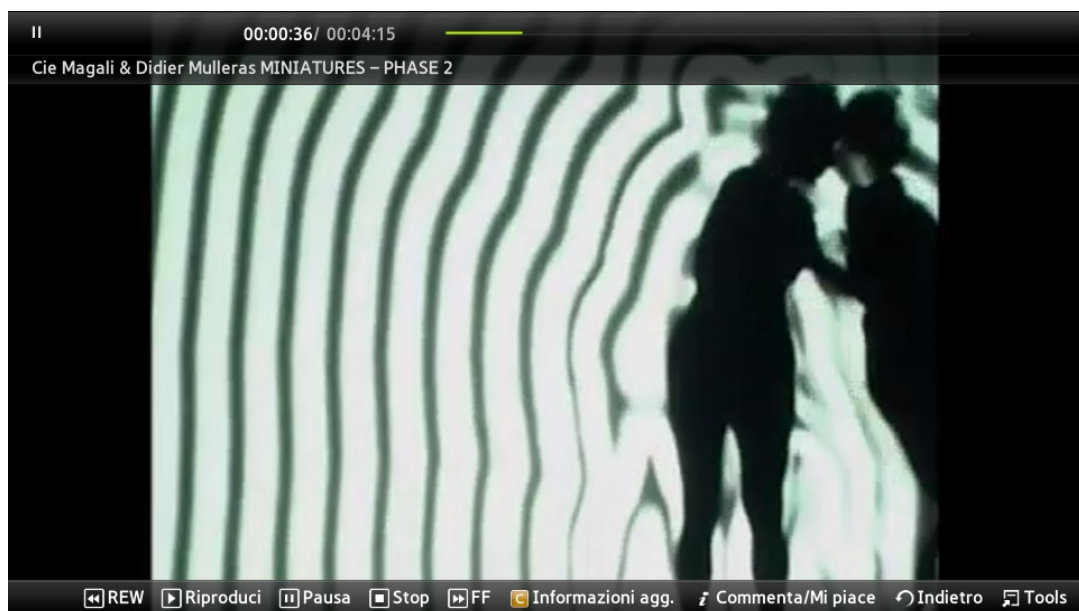


Figura 5.11: Screenshot - Video a schermo intero

Dopo aver visto e descritto la navigazione all'interno della scena principale e diverse funzionalità, ora verranno descritte le funzionalità che l'applicazione offre in relazione a un contenuto specifico, che è lo scopo principale dell'applicazione stessa.

Un video o un'immagine viene visualizzata a schermo intero premendo il tasto *ENTER* dopo aver selezionato la relativa anteprima dalla *SceneHome* (scene principale) o dalla *SceneContent* (figura 5.11- 5.12). Per quanto riguarda il video, si possono usare i classici tasti di controllo (*play*, *pause*, *stop*, *ff* e *rw*). I tasti, che offrono delle funzionalità aggiuntive sono il tasto *C* (*giallo*), che visualizza i metadati del contenuto, e il tasto *INFO*, che visualizza i commenti di un contenuto e permette di aggiungere ai preferiti e di commentare un contenuto se si è un utente autenticato.



Figura 5.12: Screenshot - Immagine a schermo intero

Visualizzazione dei metadati di un contenuto

Il tasto *C* (*giallo*) permette di leggere i metadati relativi al contenuto visualizzato; a tale funzionalità si può accedere anche selezionando la relativa anteprima dalle schermate in cui vengono visualizzate premendo il tasto *INFO*. In ogni caso, i metadati sono mostrati attraverso un popup e vengono visualizzate le seguenti informazioni aggiuntive: titolo, durata (solo per i contenuti video), data di pubblicazione, numero di visualizzazioni, data dell'evento, soggetto, descrizione e chi l'ha pubblicato. (figura 5.13 - 5.14).

CAPITOLO 5. Eclap per Smart TV: validazione e uso



Figura 5.13: Screenshot - Visualizzazione dei metadati

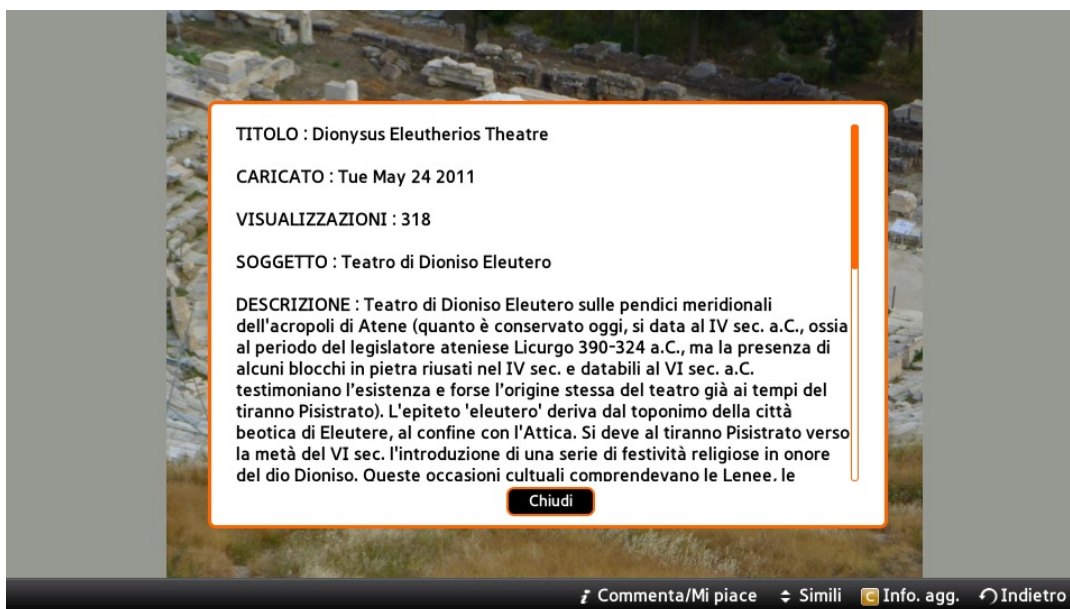


Figura 5.14: Screenshot - Visualizzazione dei metadati (scroll bar attiva)

Simili di un contenuto

Mentre un contenuto è visualizzato a schermo intero, è possibile ottenere i contenuti simili al contenuto visualizzato premendo il tasto *DOWN*. Fatto ciò comparirà un popup (*figura 5.15*) in cui verranno mostrate le anteprime dei contenuti simili, disposti in pagine dove ogni pagina è una matrice 2x4. Se uno di questi contenuti simili interessa l'utente, può essere selezionato e quindi visualizzato a schermo intero.

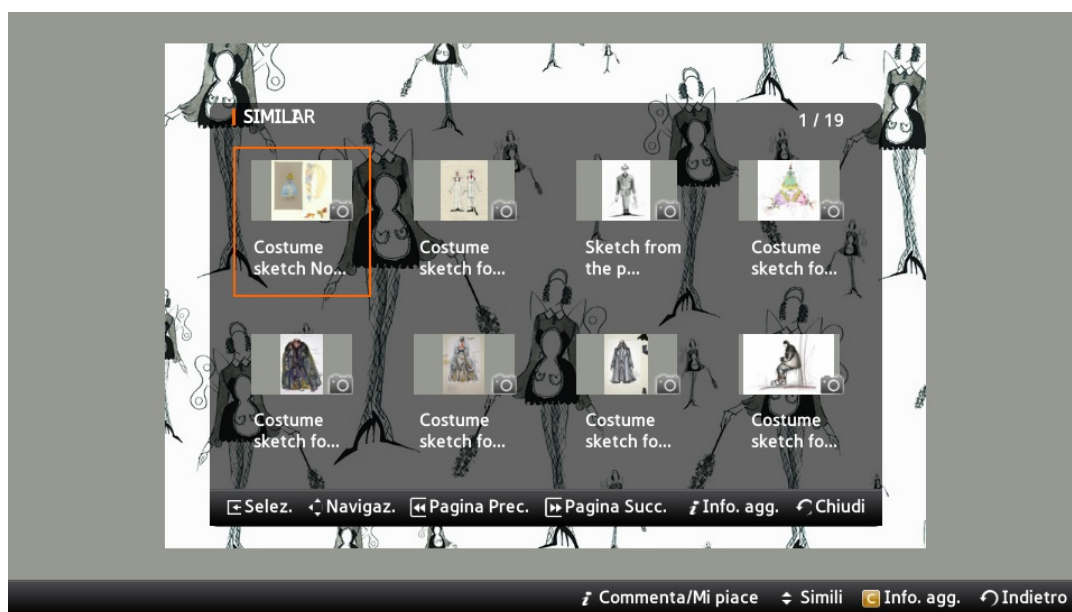


Figura 5.15: Screenshot - Visualizzazione dei simili di un contenuto

Accesso autenticato

Premendo il tasto *C* (*giallo*) dalla *SceneHome* (scena principale) o dalla *SceneContent* (schermata matrice anteprime), si passa alla schermata di login (*figura 5.16*).

Per immettere i dati all'interno dei campi di input, si possono scegliere i due metodi di scrittura descritti prima, scegliendoli attraverso il tasto *B* (*verde*). La selezione del campo di input scelto per inserire il nome utente o la password è evidenziata da un bordo arancione e con le frecce *UP/DOWN* ci si può spostare dall'una all'altra. Dopo aver inserito nome utente e password, per tentare l'accesso autenticato si preme il tasto *ENTER*; premendo il tasto *BACK* si torna alla schermata precedente.

CAPITOLO 5. Eclap per Smart TV: validazione e uso

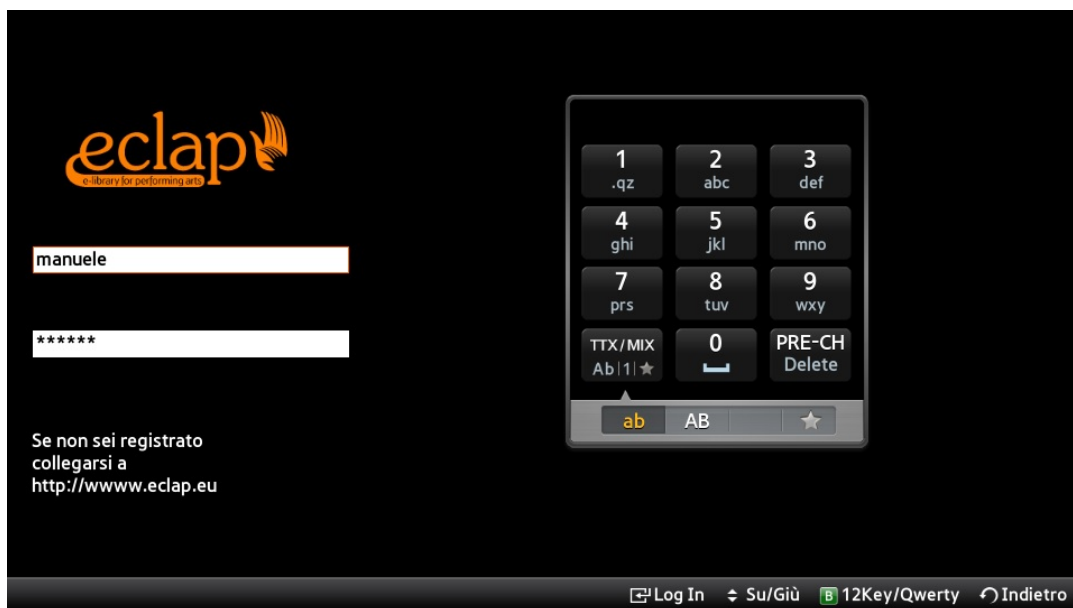


Figura 5.16: Screenshot - Login

Quando si torna nella schermata principale da utente autenticato (figura 5.17), verrà mostrato un messaggio di saluto all'utente e verranno aggiunti i canali *Miei preferiti*, *Ultimi visti* e *Visti dagli amici*.



Figura 5.17: Screenshot - Navigazione con accesso autenticato

CAPITOLO 5. Eclap per Smart TV: validazione e uso

Una volta che un utente si è autenticato, si può uscire da questa modalità premendo sempre il tasto *C* (giallo) dalla *SceneHome* (schermata principale) o dalla *SceneContent* (schermata matrice anteprime), dopodiché verrà chiesto all'utente di confermare tale azione (figura 5.18).



Figura 5.18: Screenshot - Logout

Commentare e aggiungere ai preferiti un contenuto

Mentre un contenuto è visualizzato a schermo intero, il tasto *INFO* permette di accedere alle funzionalità social dell'applicazione, ossia la visualizzazione di eventuali commenti, l'aggiunta ai preferiti e la possibilità di commentare il contenuto visualizzato.

I commenti, se ci sono, verranno visualizzati uno alla volta attraverso un popup, posto nella parte inferiore della schermata sopra l'helpbar. Con i tasti *FF* (<<) e *RW* (>>) è possibile sfogliare tutti i commenti, nel caso ce ne siano più di uno (figura 5.19).

Per attivare l'immissione di un commento bisogna premere il tasto *D* (blu) o il tasto *ENTER* quando è visualizzato il popup dei commenti (figura 5.21). Si possono scegliere due sistemi di scrittura per mezzo del tasto *B* (verde). Il primo usando i numeri del telecomando a cui sono associati tre o quattro lettere ognuno; il secondo usando una tastiera virtuale all'interno della quale ci si può muovere e selezionare la lettera che si vuole digitare. Dopo aver digitato il commento, con il tasto *ENTER* si effettua il commento al contenuto visualizzato.

CAPITOLO 5. Eclap per Smart TV: validazione e uso

Per aggiungere un contenuto ai preferiti bisogna premere il tasto *A* (red) quando è visualizzato il popup dei commenti (figura 5.20).



Figura 5.19: Screenshot - Popup dei commenti

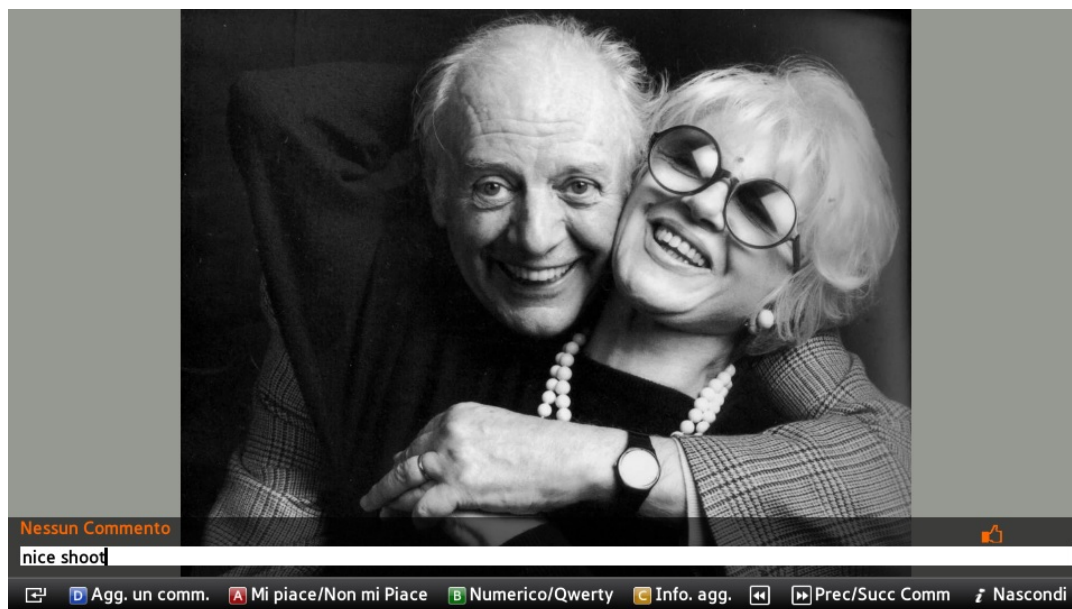


Figura 5.20: Screenshot - Commento abilitato

Gestione delle impostazioni

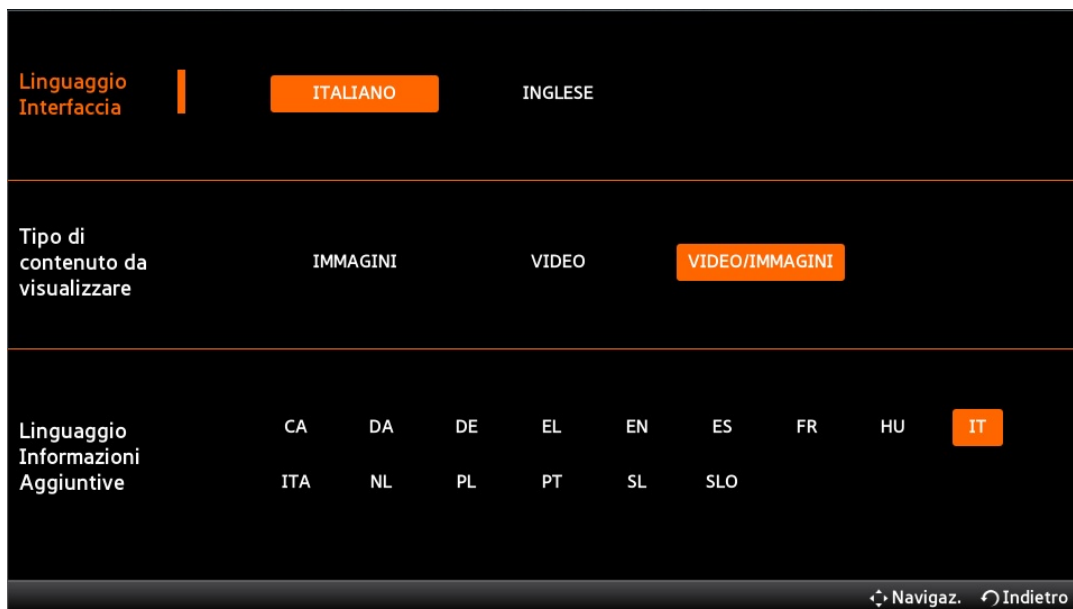


Figura 5.21: Screenshot - Impostazioni



Figura 5.22: Screenshot - Popup di conferma impostazioni

Si può accedere alla schermata delle impostazioni (*figura 5.21*) premendo il tasto *D (blu)* dalla *SceneHome* (scene principale) o dalla *SceneContent* (schermata matrice anteprime).

La navigazione all'interno di questa schermata dovrebbe essere, come negli altri casi, abbastanza intuitiva. La selezione di un'impostazione o della relativa opzione è data dal colore arancione. Usando le frecce *UP/DOWN* si sceglie una delle tre impostazioni: lingua dell'interfaccia, tipo di contenuto da visualizzare e lingua dei metadati. Dopo aver selezionato l'impostazione su cui si vogliono apportare delle modifiche, con i tasti *LEFT/RIGHT* è possibile cambiare la relativa opzione. Per tornare alla schermata precedente si usa il tasto *ENTER o BACK*. Nel caso in cui almeno un'impostazione è stata modificata, si deve confermare tale modifica. In caso di conferma le nuove impostazioni verranno memorizzate, altrimenti si torna alle impostazioni salvate precedentemente (*figura 5.22*).

5.2 Problemi incontrati

Nella fase di sviluppo dell'applicazione sono stati riscontrati principalmente tre problemi.

Uno di questi, in parte risolto, riguarda la visualizzazione di alcuni video sulla TV. In particolare alcuni video non vengono riprodotti a seguito del messaggio di errore "*video non supportato*", situazione che non si verifica visualizzando lo stesso identico video sull'emulatore. Il problema è legato al grado di risoluzione del video, in particolare ciò vale per i video codificati in *H.263*. Questa difficoltà è stata risolta, in parte, scegliendo di visualizzare il video con la maggiore risoluzione, cosa che non è stato possibile con alcuni video per i quali non esistono versioni con risoluzione maggiori di quelli codificati in *H.263*.

Nella fase di sviluppo è stato riscontrato un problema durante l'uso dell'emulatore, che dopo appena qualche decina di secondi si arrestava in modo anomalo, soprattutto durante la visualizzazione di video o l'inserimento di parole tramite il tastierino numerico o la tastiera virtuale. Questo limite dell'emulatore è stato superato dalla *Samsung*, permettendo così agli sviluppatori di emulare una TV attraverso l'installazione e l'uso dello stesso emulatore su macchina virtuale.

Durante i primi test su TV, a differenza di quanto avveniva sull'emulatore, il server non riconosceva il client, non mantenendo così traccia delle scelte fatte dall'utente (ad esempio l'effettuazione del login). Capito che il server captava il cookie del client solo se inviato prima dell'effettuazione della richiesta *AJAX*, per risolvere il problema è stato sufficiente aggiungere tra gli attributi della richiesta *AJAX (jQuery)* il seguente codice in *Javascript*:

```
beforeSend: function(xhr) {
```

CAPITOLO 5. Eclap per Smart TV: validazione e uso

```
var cookie = document.cookie;
xhr.setRequestHeader('Cookie', cookie);
}
```

ECLAP per Smart TV può essere testata su un emulatore solo a condizione che sia rimossa la funzione *Settings.setParameter()* dal file *SceneHome.js*, chiamata dalla funzione *initialize()*. In questo modo l'emulatore non legge più le impostazioni all'avvio dell'applicazione ma quelle fornite di default. È consigliabile, inoltre, rimuovere la funzione *Network.checkNetwork()* dal file *SceneHome.js*, chiamata dalla funzione *initialize()*, perché il plugin *Network* fornito da *Samsung* non funziona sull'emulatore.

Non è stato possibile testare *ECLAP per Smart TV* direttamente su modelli di TV prodotti nel 2012 e 2014, ma solo sui relativi emulatori. Questi test hanno presentato dei piccoli problemi: quello effettuato sull'emulatore 2012 ha evidenziato delle anomalie all'avvio, in pratica la scena principale (home page) si ricarica per 2-3 volte, dopodiché inizia a funzionare correttamente; sull'emulatore 2014 si rilevano delle anomalie durante la visualizzazione dei video, che viene oscurato da una schermata grigia.

Capitolo 6

Conclusioni

6.1 Conclusioni e possibili sviluppi

Nel corso di questa tesi è stata realizzata l'applicazione *ECLAP per Samsung Smart TV*, un'applicazione multimediale della categoria *Social TV*.

ECLAP è un consorzio che riunisce istituzioni europee delle arti dello spettacolo, università e istituti di ricerca e nasce come un progetto di ricerca cofinanziato dal Programma Ict-Psp Cip della Commissione Europea. *ECLAP* fa uso di un database di indicizzazione avanzata e strumenti di distribuzione per la produzione e la diffusione del ricco patrimonio artistico europeo e questo si traduce in un arricchimento e promozione della cultura europea. Attualmente *ECLAP* distribuisce più di un milione di contenuti, di cui circa 125000 provengono da 32 prestigiose istituzioni europee, ed è in grado di riprodurre immagini, video, documenti, audio, ebooks, ePub, animazioni, slides, playlists, raccolte, 3D, musica braille, annotazioni, ecc...

ECLAP per Samsung Smart TV permette di visualizzare contenuti selezionando un canale o una categoria, navigare all'interno della tassonomia dei contenuti, effettuare ricerche di contenuti in base a una keyword immessa, riprodurre immagini e video, visualizzare di un dato contenuto metadati, commenti e contenuti simili, scegliere le impostazioni relative all'uso dell'applicazione (lingua dell'interfaccia, formato dei contenuti da visualizzare e lingua dei metadati) e autenticarsi in modo da usufruire di servizi aggiuntivi, come ad esempio commentare e aggiungere ai preferiti un dato contenuto.

Il progetto dell'applicazione *ECLAP per Samsung Smart TV* si è svolto in quattro fasi:

- *prima fase* - studio dell'ambiente scelto per lo sviluppo dell'applicazione, cioè *Samsung Smart TV*

CAPITOLO 6. Conclusioni

- *seconda fase* - analisi dei requisiti e descrizione dei casi d'uso dell'applicazione
- *terza fase* - sviluppo della parte server dell'applicazione, consistente in un modulo custom scritto in *PHP* che permette la comunicazione dell'applicazione con il server, gestito da *Drupal*, e con il database, gestito da *MySQL*, e sviluppo della parte client basata sullo standard aperto *CE-HTML*
- *quarta fase* - test dell'applicazione su una *Samsung Smart TV* modello 2013.

L'applicazione *ECLAP per Samsung Smart TV* è stata inviata alla *Samsung* ed è in fase di valutazione con l'obiettivo di renderla disponibile, in breve tempo, a tutti gli utenti delle *Samsung Smart TV*.

Per *ECLAP per Smart TV* si può prevedere un ulteriore sviluppo aggiungendo le seguenti funzionalità:

- navigazione all'interno dell'applicazione per mezzo di un pad direzionale *D-pad* integrato ai telecomandi di ultima generazione
- controllo dei comandi che agiscono sulla TV per mezzo di altri due sistemi: riconoscimento vocale e gestuale
- integrazione con un'applicazione per smartphone/tablet appositamente creata con l'obiettivo di migliorare e velocizzare la funzionalità inerente all'immissione di parole.

Appendice

Samsung Smart TV App Development^[49]

Samsung, come altri produttori, dà la possibilità di sviluppare le proprie applicazioni attraverso un *SDK* (*Software Development Kit*) che viene aggiornato ogni anno con nuove funzionalità. Per il download di questo *SDK* e di altre risorse è necessario registrarsi su *Samsung Developers Forum*. L'*SDK* si integra nell'ambiente di sviluppo *Eclipse*.

Come detto in precedenza, le applicazioni sono organizzate in "scene" (con i files *HTML*, *CSS* e *Javascript*), in files di configurazione e in cartelle contenente immagini.

I files di configurazione sono: *index.html*, *app.json*, *init.js*, *config.xml* e *widget.info*.

app.json

```
{
  "scenes" : ["MainScene","SubSceneContent"],
  "files" : [],
  "theme" : "base",
  "languages" : ["en","it"]
}
```

scenes

un *Array<String>*, contiene il nome di tutte le scene usate nell'applicazione.

files

un *Array<String>*, è una proprietà opzionale.

theme

una *String*, definisce il tema dei componenti dell'*Apps Framework UI*. Per

ora

solo il tema "base" è supportato

APPENDICE

languages

un *Array<String>*, è una proprietà opzionale e serve per impostare i linguaggi

supportati dall'applicazione, il primo valore è il linguaggio di default

init.js

```
function onStart() {
    sf.scene.show('MainScene');
    sf.scene.focus('MainScene');
}
function onDestroy() {
}
```

La funzione *onStart()* viene eseguita all'avvio dell'applicazione, che mostra e assegna il focus alla scena principale. La funzione *onDestroy()* viene eseguita alla chiusura dell'applicazione.

index.html

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset=utf-8">
    <script type="text/javascript" language="javascript"
      src="$MANAGER_WIDGET/Common/af/2.0.0/loader.js">
    </script>
    <script type="text/javascript" language="javascript"
      src="$MANAGER_WIDGET/Common/API/Widget.js">
    </script>
    <script type="text/javascript" language="javascript"
      src="$MANAGER_WIDGET/Common/API/TVKeyValue.js">
    </script>
    <object id='pluginNetwork' border='0'
      classid='clsid:SAMSUNG-INFOLINK-NETWORK'>
    </object>
    <object id='pluginObjectTV' border='0'
      classid='clsid:SAMSUNG-INFOLINK-TV'>
    </object>
    <script type="text/javascript" src="app/javascript/Javascript.js">
```

APPENDICE

```
</script>
</head>
<body>
  <script language="javascript" type="text/javascript"
    src="$MANAGER_WIDGET/Common/IME/ime2.js" >
  </script>
  <script type="text/javascript"
    src="$MANAGER_WIDGET/Common/IME_XT9/ime.js" >
  </script>
  <div id="loadingScene" class="sf-ui-loading"
    style="display:block;position:absolute;z-index:200;visibility:hidden;" >
    <div class="sf-ui-loading-circle-1" ></div>
  </div>
</body>
</html>
```

Il file *index.html* contiene tutte le *API* usate nell'applicazione, la versione dell'*Apps Framework* usato, eventuali oggetti forniti da *Samsung* ed eventuali elementi *HTML*.

config.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<widget>
  <ThumbIcon>Resource/image/icon/picasa_106.png</ThumbIcon>
  <BigThumbIcon>Resource/image/icon/picasa_115.png</BigThumbIcon>

  <ListIcon>Resource/image/icon/picasa_85.png</ListIcon>
  <BigListIcon>Resource/image/icon/picasa_95.png</BigListIcon>
  <category></category>
  <autoUpdate>n</autoUpdate>
  <ver>0.930</ver>
  <mgrver>1.000</mgrver>
  <fullwidget>y</fullwidget>
  <srcctl>y</srcctl>
  <audiomute>n</audiomute>
  <videomute>n</videomute>
  <dcont>y</dcont>
  <network>y</network>
```

APPENDICE

```
<widgetname>HelloWorld</widgetname>
<description>Welcome!</description>
<width>960</width>
<height>540</height>
<author>
  <name>Samsung Electronics Co. Ltd.</name>
  <email></email>
  <link>http://www.sec.co.kr/</link>
  <organization>Samsung Electronics Co. Ltd.</organization>
</author>
</widget>
```

<ThumbIcon> <BigThumbIcon> <ListIcon> <BigListIcon>

contengono i percorsi delle icone dell'applicazione e si differenziano per dimensioni

<category>

indica la categoria dell'applicazione, ad esempio video, sport, game, information, education

<autoUpdate>

se 'y' l'applicazione si sincronizza con l'hub site. Se non c'è bisogno di nessuna

sincronizzazione, seleziona 'n'

<ver>

indica la versione dell'applicazione.

<mgrver>

indica la versione dell'*Application Manager* che è richiesta per avviare l'applicazione

<fullwidget>

indica se l'applicazione è a schermo intero o no

<srcctl>

se 'y' permette di passare dai canali TV al media player interno all'applicazione

e viceversa

<audiomute> <videomute>

impostano l'audio o il video muti all'avvio dell'applicazione se settati a 'y'

<dcont>

quando l'applicazione è a schermo intero, settandolo a 'y' rimuove lo sparkling

<widgetname>

APPENDICE

contiene il nome dell'applicazione

`<description>`

contiene una descrizione dell'applicazione

`<network>`

se 'y', all'avvio dell'applicazione viene effettuato un test dell'applicazione, se

fallisce l'applicazione non si avvia e viene visualizzato un messaggio che indica

il fallimento

`<width>` `<height>`

indicano le dimensioni dell'applicazione, rispettivamente larghezza e altezza

`<author>`

tutti i campi contenuti in questo tag contengono informazioni relative a chi l'ha pubblicata

Scene HTML file

I files *HTML* relativi alle scene devono avere la forma `[SceneID].html` ed essere posizionati nella cartella `app/html`.

Supponendo che si stia sviluppando la *MainScene*, il file avrà il seguente nome `MainScene.html` e la seguente struttura:

```
<div id="SceneMain" style="visibility: visible;">
  <!-- Scene HTML contents -->
</div>
```

Scene CSS file

Il nome del file *CSS*, usato per attribuire le proprietà agli elementi definiti nel file *HTML*, deve avere la forma `[SceneID].css` ed essere posizionato nella cartella `app/stylesheets`. Tutti i files *CSS* saranno caricati quando l'applicazione viene avviata. Ad esempio:

```
#SceneMain{
  position: absolute;
  left: 0px;
  top: 0px;
  width: 960px;
  height: 540px;
  background-color: #ffffff;
```

APPENDICE

```
}
```

Da notare che se l'applicazione non occupa tutto lo schermo, la posizione (*top* e *left*) e la dimensione (*width* e *height*) potrebbero cambiare.

Scene Javascript File

Il file *Javascript* deve assumere la forma: *[SceneID].js* ed essere posizionato nella cartella *app/scenes*. Questo file contiene la definizione della classe *SceneID*. Quando viene avviata un'applicazione, un'istanza di questa classe è creata e gestita attraverso lo *Scene Manager API*^[36] e i suoi metodi. Di seguito un esempio di una scena con le funzionalità di base.

```
function SceneMain(options) {  
}  
SceneMain.prototype.initialize = function () {  
    alert("SceneMain.initialize()");  
    //questa funzione sarà chiamata solo una volta, quando lo scene manager  
    mostra questa scena per la prima volta  
    //inizializza gli stili e i controlli della scena, e inizializza le variabili  
    //i files HTML e CSS della scena saranno caricati prima che questa funzione  
    sia chiamata  
}  
SceneMain.prototype.handleShow = function (options) {  
    alert("SceneMain.handleShow()");  
    //questa funzione sarà chiamata quando lo scene manager mostra questa  
scena  
}  
SceneMain.prototype.handleHide = function (options) {  
    alert("SceneMain.handleHide()");  
    //questa funzione sarà chiamata quando lo scene manager nasconde questa  
scena  
}  
SceneMain.prototype.handleFocus = function (options) {  
    alert("SceneMain.handleFocus()");  
    //questa funzione sarà chiamata quando lo scene manager assegna il focus  
a  
    questa scena  
}
```

APPENDICE

```
SceneMain.prototype.handleBlur = function () {
    alert("SceneMain.handleBlur()");
    //questa funzione sarà chiamata quando lo scene manager sposta il focus
da
    questa scena a un'altra scena
}
SceneMain.prototype.handleKeyDown = function (keyCode) {
    alert("SceneMain.handleKeyDown(" + keyCode + ")");
    // TODO: write a key event handler when this scene gets the focus
    switch (keyCode) {
        case sf.key.LEFT:
            break;
        case sf.key.RIGHT:
            break;
        case sf.key.UP:
            break;
        case sf.key.DOWN:
            break;
        case sf.key.ENTER:
            break;
    }
}
```

Riproduzione video

Il sistema adottato per riprodurre video in *ECLAP per Smart TV* è attraverso il componente *VideoPlayer Service API*^[41].

I video possono essere riprodotti a schermo intero, su una parte dello schermo, o in entrambi i modi (passando dall'uno all'altro). Il seguente codice permette di mostrare il video su una parte dello schermo scelta dallo sviluppatore impostando le variabili *left*, *top*, *width* e *height*.

```
// Imposto la posizione e la grandezza del VideoPlayer all'interno della schermata
sf.service.VideoPlayer.setPosition({
    left: 100,
    top: 100,
    width: 500,
    height: 400
});
```

APPENDICE

```
});

//Avvia la riproduzione
sf.service.VideoPlayer.play({
  url: 'http://mydomain.com/TestContent.mp4',
  fullScreen: false // Sets Player to partial mode
});
```

L'avvio del VideoPlayer a schermo intero si effettua nel seguente modo:

```
// riproduttore video a schermo intero
sf.service.VideoPlayer.setFullScreen(true);

// avvio del riproduttore video
sf.service.VideoPlayer.play({
  url: 'http://mydomain.com/TestContent.mp4',
  fullScreen: true
});
```

Se il VideoPlayer è attivo, i tasti del telecomando sono processati dal componente *VideoPlayer ServiceAPI*. Per impostare la funzione di ogni singolo tasto, usare il seguente metodo:

```
sf.service.VideoPlayer.setKeyHandler(sf.key.RETURN, function () {
  //Quando viene premuto il tasto RETURN viene eseguita questa funzione
});
```

Supporto multi-language^[43]

Un'applicazione, per quanto riguarda l'interfaccia, può supportare più di una lingua. Si deve aggiungere la proprietà *"languages"* nel file *app.json*, che contiene i codici delle lingue nell'*ISO 639-1* e, quindi, per ogni lingua si deve creare un file nella cartella *lang* con il nome nella forma *[languagecode].js*, in questo modo il corrispondente file è incluso automaticamente durante l'inizializzazione dell'applicazione.

Se la proprietà *"languages"* nel file *app.json* contiene le lingue *en (inglese)* e *it (italiano)*, per ognuna di queste deve essere creato un file *Javascript* con i nomi *en.js* e *it.js*

APPENDICE

en.js

```
var STR_YES = "Yes";  
var STR_NO = "No";  
...
```

it.js

```
var STR_YES = "Si";  
var STR_NO = "No";  
...
```

Uso degli UI Components^[37]

Gli UI Components dell'*AppsFramework* usano la libreria *jQuery* ed sono definiti come plugin *jQuery* nella forma "*sf_*". Per usare l'*AppsFramework UI Components* va creato e selezionato l'elemento *HTML* in cui creare lo *UI Component* e chiamare il metodo del plugin *UI Component*

L'esempio seguente riguarda l'uso del *Loading UI Components*^[39]. In un file *HTML* va allegato il seguente codice all'interno del tag `<body>`.

```
<div id="loading" class="sf-ui-loading"  
  style="display:block;position:absolute;z-index:200;visibility:hidden;" >  
  <div class="sf-ui-loading-circle-1" ></div>  
</div>
```

Nell'esempio precedente, le proprietà CSS sono state incluse direttamente nel file *HTML*; in alternativa, è possibile includerle in un file *CSS*.

Per usare un metodo dell'*UI Component* creato, va selezionato l'elemento *HTML* e chiamato il metodo desiderato:

```
//mostra il componente di caricamento  
$('#loading').sfLoading('show');  
//imposta la velocità di rotazione  
$('#loading').sfLoading({  
  stepIntervalTime: 250  
});
```

Test di un'applicazione sulla TV^[50]

Una volta completata la fase di test e debug, l'applicazione va caricata su una *Samsung Smart TV* in modo da effettuare una verifica completa sull'effettivo funzionamento dell'applicazione. Un'applicazione può essere caricata direttamente dal PC di sviluppo alla TV attraverso un server locale e riconosciuta come *User Applications*.

Per testare un'applicazione su un televisore è necessario configurare il computer di sviluppo nel seguente modo:

- installare il *Web Server Apache* (versione 2.2.16) contemporaneamente o successivamente all'installazione del *SDK Samsung Smart TV*
- impostare le preferenze dell'SDK:
 - selezionare l'opzione *Samsung Smart TV SDK Preferences* dal menù *Samsung Smart TV SDK*
 - inserire il *Server PC IP* e la cartella *root* (esempio: [*\$Apache install directory*]\htdocs)
- creare il pacchetto dell'applicazione da caricare sulla TV:
 - cliccare su *App Packing* all'interno della voce *Samsung Smart TV SDK*
 - si aprirà un box da compilare con il nome e il numero di versione dell'applicazione, il Paese dove l'applicazione sarà usata e la data
 - selezionare *Update the packaged files on the server*
 - cliccare su *OK* in questo modo il pacchetto verrà caricato sul *Web Server Apache*

Dopo aver caricato l'applicazione sul server web, è necessario configurare la TV:

- configurare la rete:
 - connettere la TV alla stessa rete a cui è connesso il computer tramite wireless o cavo di rete
 - nel menù *Settings* della TV selezionare *Network* ed entrare in *Network Setup*

APPENDICE

- scegliere una configurazione *automatica* (per DHCP) o una configurazione *manuale* (per indirizzi IP statici) immettendo l'indirizzo IP, la subnet mask, il gateway e il server DNS. Infine, verificare che la rete funzioni correttamente
- effettuare il login a *User App* e installare l'applicazione:
 - premere il tasto *MENU* del telecomando
 - cliccare su *Smart Features*
 - cliccare su *Samsung Account*
 - effettuare il *Log In* inserendo “*develop*” nell'*ID* e lasciando vuoto lo spazio riservato alla password
 - tornare nel menù *Apps*, cliccare su *More Apps*, quindi su *Options* e poi su *IP Setting* per immettere l'indirizzo IP del web server che contiene il pacchetto dell'applicazione
 - infine dal menù *Options*, cliccando su *Start App Sync*, viene caricata l'applicazione sulla TV per essere testata e utilizzata

Procedura di sottomissione^[51]

Affinché un'applicazione sviluppata per *Samsung Smart TV* sia distribuita da *Samsung* tramite l'*Apps TV Store*, è necessario registrarsi e inserire le informazioni relative al prodotto da sottomettere in *Samsung Apps Seller Office*.

Il processo di sottomissione generale si divide in tre passaggi: *Basic Information*, in cui si descrivono le informazioni dell'applicazione: dispositivi supportati, categoria di appartenenza dell'applicazione e caricamento dell'applicazione da sottomettere; *Display Information*, in cui si descrive ciò che è visibile agli utilizzatori; *Final Review*, che rappresenta la fase finale del processo di sottomissione.

Nella fase di sottomissione è necessario effettuare la descrizione dell'applicazione da sottomettere basandosi su un modello stabilito:

- *UI Structure*, che consiste nel mostrare la struttura dell'interfaccia utente per mezzo di una rappresentazione ad albero o di un grafo di flusso
- *Usage Scenario*, in cui si descrivono le varie funzionalità dell'applicazione
- *Menu & function description*, che consiste nella descrizione delle schermate dell'applicazione attraverso l'uso di screenshot

APPENDICE

- *Key Policy*, in cui si descrivono le funzionalità dei vari tasti del telecomando nelle varie scene
- *How to change the languages*, che consiste nella descrizione di come l'utente può cambiare la lingua dell'interfaccia, se supportato

Inoltre *Samsung* fornisce agli sviluppatore un *self checklist* per verificare la qualità dell'applicazione prima di iniziare la fase di sottomissione. La *checklist* comprende una serie di verifiche di controllo dell'applicazione organizzate in categorie che sono:

- *Configurazioni Iniziali*, che presenta le seguenti sottocategorie: *App Description*, *Credentials for testing*, *Geo-IP Blocking*, *Ticker (non EU countries)* e serve per verificare le configurazioni iniziali dell'applicazione
- *Operazioni base*, che ha lo scopo di controllare le funzioni base dell'applicazione, in particolare: verifica delle funzionalità dell'applicazione descritte nell'*App Description*, dell'operazione di login, del funzionamento dei tasti *RETURN* ed *EXIT*, dei tasti durante la riproduzione di un video e della sospensione dello screensaver durante la riproduzione di un video
- *Operazioni specifiche / UI*, consistenti nella verifica di operazioni più specifiche: visibilità del focus sugli oggetti presenti sullo schermo, cambiamento della lingua tramite Device Menu Language, funzionamento della navigazione, capacità del tasto *RETURN* di muoversi tra le varie scene o per tornare nello *SmartHub*, funzionamento del tasto *EXIT*, rispetto delle linee guida *Samsung Smart TV SDK UX Guideline*, corrispondenza tra anteprima e relativo contenuto, attivazione della barra di scorrimento quando il testo supera la dimensione del contenitore, visibilità di un indicatore di caricamento durante l'elaborazione dei dati
- *Eccezioni*, in cui si verifica che gli errori siano gestiti correttamente durante l'uso dell'applicazione, ad esempio la notifica all'utente della disconnessione della TV dalla rete, e la sincronizzazione del volume tra dispositivo e applicazione.
- *Funzioni*, che consiste nel verificare che la riproduzione di immagini e video avvenga correttamente e che i popup risultino ben visibili.
- *Input*, che serve per controllare che l'uso del campo di input funzioni correttamente

APPENDICE

A conclusione del processo di sottomissione, l'applicazione sviluppata subirà un processo di certificazione costituito da quattro fasi: *Pre-examination*, in cui l'applicazione viene convalidata sulla base di alcuni requisiti (validazione *API*, integrità del Manifest File, malware, etc); *Content Review*, in cui viene verificata l'integrità etica dei contenuti; *Device Test*, che corrisponde alla fase di testing; *Confirmation*, che consiste nell'approvazione o meno dell'applicazione sulla base dei risultati ottenuti nelle precedenti tre fasi.

Bibliografia

- [1] http://it.wikipedia.org/wiki/Smart_TV
- [2] http://it.wikipedia.org/wiki/Social_TV
- [3] <http://www.samsungdforum.com/Guide/art00017/index.html>
- [4] <http://www.samsungdforum.com/Guide/ref00009/index.html>
- [5] <http://www.samsungdforum.com/Guide/ref00009/uiapi.html>
- [6] http://www.samsungdforum.com/Guide/ref00009/apis_apps_serviceapi_entry.html
- [7] <http://www.samsungdforum.com/Guide/ref00006/index.html>
- [8] <http://www.samsungdforum.com/Guide/d16/index.html>
- [9] <http://www.samsungdforum.com/Guide/ref00003/index.html>
- [10] <http://www.samsungdforum.com/Guide/ref00001/index.html>
- [11] <http://www.samsungdforum.com/Guide/art00105/index.html>
- [12] <http://www.samsungdforum.com/Guide/ref00008/index.html>
- [13] <http://www.samsungdforum.com/Guide/art00102/index.html>
- [14] http://www.samsungdforum.com/Guide/ref00006/common_module_tvkeyvalue_object.html
- [15] http://www.samsungdforum.com/Guide/ref00006/common_module_widget_object.html
- [16] http://www.samsungdforum.com/Guide/ref00006/common_module_plugin_object.html

BIBLIOGRAFIA

- [17] http://www.samsungdforum.com/Guide/ref00006/common_module_cimageviewer.html
- [18] http://www.samsungdforum.com/Guide/ref00006/common_module_ime_object.html
- [19] http://www.samsungdforum.com/Guide/ref00006/common_module_sso_object.html
- [20] <http://www.samsungdforum.com/Guide/tut00053/index.html>
- [21] http://it.wikipedia.org/wiki/Universal_Plug_and_Play
- [22] http://www.samsungdforum.com/Guide/ref00009/service_smartremote.html
- [23] <http://www.samsungdforum.com/Guide/tut000001/index.html>
- [24] <http://www.samsungdforum.com/Guide/d01/index.html>
- [25] <http://www.samsungdforum.com/Guide/art00027/index.html>
- [26] http://www.samsungdforum.com/Guide/ref00008/nservice/dtv_nservice.html
- [27] <http://www.samsungdforum.com/Guide/d30/index.html>
- [28] <http://www.eclap.eu/portal/?q=en-US/node/3727>
- [29] <http://www.eclap.eu/portal/?q=node/94220>
- [30] <http://www.eclap.eu/portal/?q=node/3748>
- [31] <http://www.eclap.eu/portal/?q=node/116088>
- [32] <http://www.eclap.eu/portal/?q=node/3739>
- [33] <http://www.eclap.eu/portal/?q=node/103996>
- [34] http://it.wikipedia.org/wiki/Architettura_three-tier
- [35] http://it.wikipedia.org/wiki/Architettura_multi-tier
- [36] http://www.samsungdforum.com/Guide/ref00009/sfscene_api.html
- [37] <http://www.samsungdforum.com/Guide/art00024/index.html>
- [38] http://www.samsungdforum.com/Guide/ref00009/uiapi_keyhelp.html

BIBLIOGRAFIA

- [39] http://www.samsungdforum.com/Guide/ref00009/uiapi_loading.html
- [40] <http://www.samsungdforum.com/Guide/art00046/index.html>
- [41] http://www.samsungdforum.com/Guide/ref00009/service_videoplayer.html
- [42] <http://www.samsungdforum.com/Guide/tut00118/index.html>
- [43] <http://www.samsungdforum.com/Guide/art00020/index.html>
- [44] http://it.wikipedia.org/wiki/Sequence_Diagram
- [45] <http://it.wikipedia.org/wiki/Drupal>
- [46] <http://api.jquery.com/jquery.ajax/>
- [47] <http://it.wikipedia.org/wiki/JSON>
- [48] <http://it.wikipedia.org/wiki/SQL>
- [49] <http://www.samsungdforum.com/Guide/art00018/index.html>
- [50] <http://www.samsungdforum.com/Guide/d20/index.html>
- [51] <http://developer.samsung.com/distribute/app-submission-guide>