

# Realizing CRM in the broadcast industry — by using the second screen

**Robbie De Sutter and Lode Nachtergaele**

*VRT-medialab*

**Consumers are becoming more and more digital-oriented. Today, most of them have personal and always-connected mobile internet devices, thus allowing them to create a digital footprint anytime, anywhere. In tandem with this, new opportunities arise for the “classic” broadcast industry to set up and maintain a direct relationship with their TV viewers and radio listeners.**

**Until recently, a broadcaster had a one-way connection to its customers ... namely, over a TV or radio distribution channel to a physical TV screen or radio set. Interaction was only possible after implementing and deploying expensive and hard-to-develop software on a set-top box. However, by employing web technology intelligently, a broadcaster can now more easily connect to its customers and build a direct relationship with them.**

**In this article, we discuss how to set up such a system and what the particular needs are in a broadcast context. We use the second screen to collect data from our customers, and enrich it in order to acquire beneficial information for broadcasters and advertisers. We discuss a scalable architecture to process and handle this user information in real time, as well as for offline statistical analysis.**

In the retail industry, it is a given fact that knowledge about your customers and their desires is key to improve your service and retain your consumer base. This had led to the concept of **Customer Relationship Management** (CRM) [1][2][3], which enables a company to create a strategy for managing interactions with its customers. Although originally designed to improve sales, the concepts are nowadays being used to improve customer service and to support the marketing process [4].

Until recently, a media company – in particular TV and radio broadcasters – did not have the possibility to implement CRM systems, as they lacked one crucial piece of information: who is the actual consumer of the product, i.e. who is the person watching the TV and listening to the radio programmes? This is due to the fact that a broadcaster has no efficient means of knowing who is actually sitting in front of the TV screen or radio set. Through the use of a (weekly) newsletter, some broadcasters have tried to overcome this problem; however the net result has been somewhat limited.

Recent technological advances however have created the opportunity to overcome these limitations. Affordable devices are becoming more and more mobile, personal and always internet-connected. And thanks to these devices (such as smartphones and internet tablets) along with cheap and ubiquitous network connectivity through 3G and Wi-Fi, a user can express himself anytime and any-

where, and leave a digital footprint. In the context of a broadcaster, we call these devices the “second screen” – the TV set being the first screen. The second screen allows the user to express his/her emotions through comments, tweets, and rating systems about the content on the TV or radio, whenever he or she is triggered by it.

As broadcasters are interested in which content is being consumed by which consumer, we propose a technique to employ second-screen devices to collect that particular information. Furthermore, by enriching the user’s information with socio-demographic information (such as age, gender, residence, number of family members, etc.), broadcasters can employ this knowledge to improve their programme schedules as well as providing better-targeted advertisements.

Media consumers can also benefit. As their second-screen devices will be aware of their media consumption history and context, personalized services and simplified social interactions about the content become possible, and existing services will improve significantly.

## Measuring the audience

As discussed above, the main issue for broadcasters before CRM systems could be implemented was to figure out who is actually consuming the media content. Currently, audience measurement is done through specialized companies such as the CIM in Belgium<sup>1</sup>, SKO in the Netherlands<sup>2</sup>, Nielsen in the USA<sup>3</sup>, etc. These companies typically use a panel group, equipped with dedicated hardware, to estimate the ratings. For example, in Belgium, the TARIS 5000 PeopleMeter device developed by TNS<sup>4</sup> is used to automatically detect to which channel the TV is tuned, by analyzing the signal that is sent to the television screen. It cannot detect which users are watching; this information has to be entered manually by the participant. Another system is Arbitron’s Portable People Meter<sup>5</sup> which processes the ambient sound and searches for watermarks that identify the channel source. A similar system is used by Nielsen in the USA and Civolution<sup>6</sup> in France.

A third type of system uses a camera to monitor what consumers are watching. Examples of this technology are Cognovision’s AimView<sup>7</sup> and TruMedia AlliO<sup>8</sup>.

Finally, some experiments are being conducted where the TV device itself is made aware of the audience by allowing the users to “log on” to their TV set. For example, Philips’ Aprico<sup>9</sup> system, as well as various set-top boxes, allow users to create a profile that is used to keep track of the channels or programmes they watch, as described in [5][6].

Although these systems are becoming more intelligent, they have several fundamental issues, making it hard to deploy these systems on a larger scale than with a panel population. In particular, the usability of these systems is (i) *low* (e.g., there is almost no incentive why a person would “log on” to a TV screen and even less to update this information if additional people join in) or (ii) *undesirable* (e.g., a camera “spying” in the living room is very intrusive). Furthermore, all these systems pin the users to the room where the device is located. Some solutions require special – expensive and proprietary – devices at the consumer’s side, severely limiting the possible size of the measurement

- 
1. <http://www.cim.be>
  2. <http://www.kijkonderzoek.nl>
  3. <http://www.nielsen.com>
  4. <http://www.tnsglobal.com>
  5. <http://www.arbitron.com>
  6. <http://www.civolution.com>
  7. <http://www.cognovision.com>
  8. <http://www.tru-media.com>
  9. <http://www.aprico.tv/>

panels. Furthermore, most of the existing technologies only monitor the classical TV broadcast signal and ignore internet streaming or other types of playout. Finally, none of the mentioned solutions is able to produce measurement data in real time.

By using the second screen as a measuring device, we can overcome these issues.

## Abbreviations

<b>APG</b>	“As-run” Programme Guide
<b>CRM</b>	Customer Relationship Management
<b>HbbTV</b>	Hybrid Broadband Broadcast Television <a href="http://www.hbbtv.org/">http://www.hbbtv.org/</a>
<b>HTML</b>	HyperText Markup Language
<b>I/O</b>	Input/Output
<b>ID</b>	IDentification / IDentity / IDentifier
<b>PPG</b>	Planned Programme Guide

## Attracting the users

Our starting point is that an average customer is, or soon will be, digitally equipped and hence is using a second-screen device (such as a smartphone, tablet or laptop), while consuming media content on the first screen (e.g., a regular TV or radio set). Some broadcasters perceive this as a threat ... as the eye-balls are moving away from the first screen to the second screen. Other broadcasters (inspired by TV and set-top box vendors) believe that the answer can be found by bringing the internet content to the first screen. These solutions allow users to chat, vote and share their emotions through the TV set.

However, we at VRT are convinced this is not a sufficient solution, mainly because the main screen is a shared device whereas the interactions are personal. Ask yourself: would you like to sacrifice a part of your TV screen in order to display the Facebook friends and comments of one or more of your children? Furthermore, despite initiatives like HbbTV<sup>10</sup>, many different (proprietary) standards and systems exist to bring content to the first screen. Should a broadcaster develop for, and support, all these different devices?

Our approach is to interpret the second screen as an opportunity instead of a threat. A broadcaster should also be present on those devices in order to recover the eye-balls. Furthermore, we can establish a direct communication relationship with the individual person using the device. Finally, the development of applications for these devices can be realized in a more cost-effective manner.

In particular, by using the upcoming HTML5 technology to develop web applications, most – if not all – second-screen devices can be targeted simultaneously by an already-existing vast developer network. Indeed, the number of professionals able to develop in HTML5 is several magnitudes larger than the number of professionals able to develop for particular (proprietary) TV devices or set-top boxes. The latter holds true, even when HbbTV has been standardized.

## VRT MediaSquare

VRT-medialab has developed **MediaSquare**, a second-screen web application to help consumers of prime-time content to interact digitally. *Fig. 1* shows a screenshot of the prototype application, in particular the programme page. On this page, a user can read comments from other MediaSquare users, or tweets using the particular programme hash tag. He or she can also get an overview of the persons who watch, like, or dislike this programme. This information can be filtered so it also displays the information from friends in the MediaSquare environment.

User can participate by, for example, posting comments. They can easily choose whether or not these messages must also be posted on their Twitter or Facebook accounts. Finally, some additional “applications” can be linked on a programme page, such as voting or preview applications. Note that the web application does not stream the content. The audiovisual content remains on the first screen whereas the additional interactivity is on the second screen.

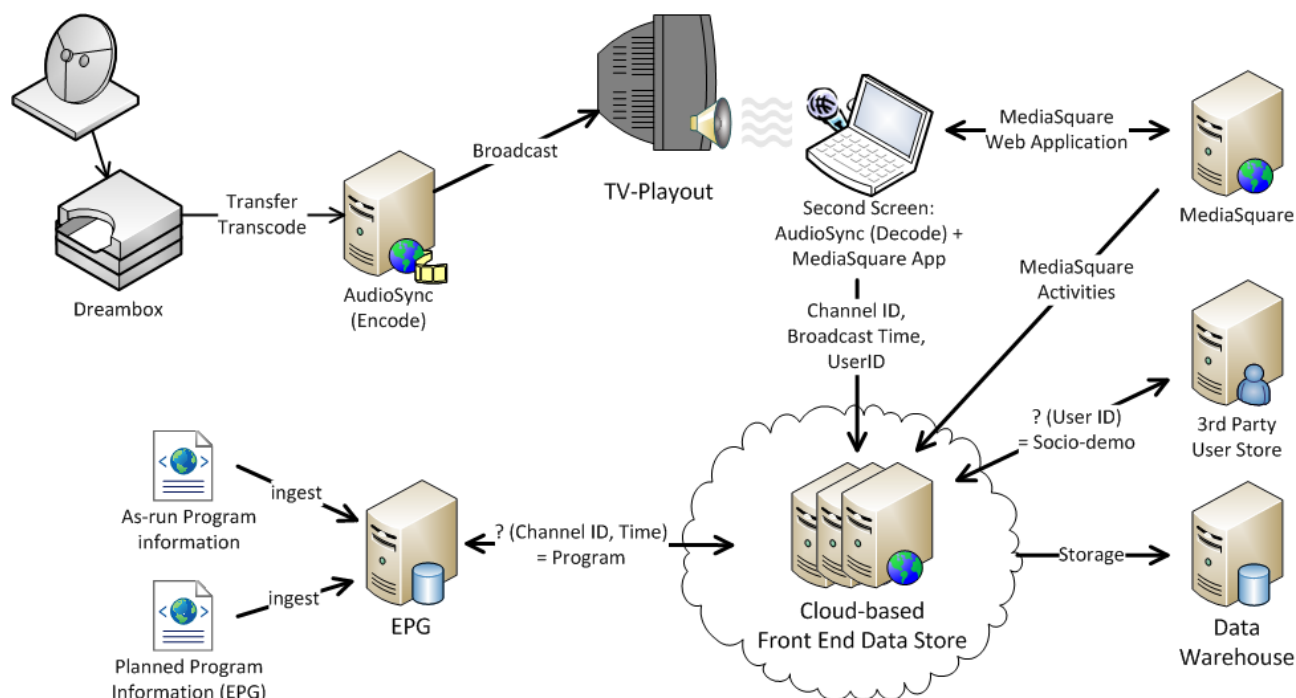
The user benefits, as MediaSquare offers a common marketplace where discussions about TV and radio content can take place. In contrast to Twitter, it is much easier for non-savvy people to use

10. <http://www.hbbtv.org/>





fingerprinting [8] techniques, it is possible to synchronize the second screen with the first screen. As these second-screen devices are personal, we can match a particular user with the consumed TV or radio channel, together with the broadcast time of the programme. After obtaining the explicit consent of the user to meet privacy concerns, this measured behaviour can be transmitted to a central data warehouse system that logs the data and builds up an individual user viewing profile.



**Figure 2**  
Collecting and enriching the consumer data

Fig. 2 give an overview of the set-up we created as a proof of concept. In this set-up, we captured the broadcast TV and radio streams from a satellite by using Dream Multimedia's Dreambox DM 8000 device<sup>13</sup>. This device allows the capturing of six simultaneous broadcast streams from two satellite transponders. The signal is transferred to an encoding station that watermarks the audio signal of the stream. This watermark contains a channel identification code and the broadcast wall-clock time. The watermarked streams are re-multiplexed with the video stream and sent to a regular TV screen for playout.

A second-screen device – represented by a laptop in the diagram – is equipped with watermark-decoding software and is able to extract the embedded watermarks from the audio emitted by the TV set. As soon as the watermark is detected and decoded, and at regular intervals (every five seconds), the Channel ID and Broadcast Time together with the ID of the user is sent to a central system – the Front End Data Store. To handle peak load and scalability, this system is implemented on the Amazon EC2 cloud infrastructure.

### **Enriching the user data**

As explained in the previous section, the data received from the user must be enriched to become beneficial information.

First, the channel information and broadcast time is used to distil the name of the actual programme or advertisement being watched. A first approximation is obtained by using the planned programme

12. <http://www.civolution.com>

13. <http://www.dream-multimedia-tv.de>

guide (PPG). However, as live TV and radio always divert from the PPG, the actual “as-run” programme guide (APG) is required to know the exact start and end time of the programmes or advert spots. Note that the as-run information is available after broadcasting, whereas the planned programme information is available up-front.

The broadcast time is required to handle time-shifted viewing. For example, if a user is watching the seven o'clock news programme the next day at nine o'clock in the morning, then we want to register that the user has watched that particular news programme, and not the show that was actually broadcast at wall-clock time (i.e. nine o'clock the next morning).

Finally, the user information can be enriched with socio-demographic information. This can be done **explicitly** by the user himself/herself (e.g., by filling in an online questionnaire), **implicitly** (e.g., by giving information required to participate in a contest) or by a **third-party service provider** (e.g., in Europe the company Bisnode<sup>14</sup> collects and maintains consumer information, for example, obtained through the retail and banking sector). This additional information will give the broadcaster as well as the advertiser an insight into the profiles of their customers. It will also enable segmentation of the users into different groups which can be individually targeted by, for example, tailor-made adverts.

As illustrated in *Fig. 2*, all collected information is eventually stored in a Data Warehouse. The Data Warehouse can be used to perform statistical analysis on the data and it serves as input data for the available CRM software tools such as Salesforce<sup>15</sup>.

## Architecture

In the previous sections, we described how the information about a consumer at a particular time can be collected and enriched. As a user can turn off the system or change the broadcast channel at any given time, we need to update this information regularly. As we want to use this system to inform advertisers about which commercials are actually being watched, we have chosen to re-send the information every five seconds. Furthermore, we will typically handle more than one consumer at the same time. Hence, if we want to deploy this system in the Flemish region of Belgium, with about 6 million inhabitants, and if we assume that around 1% of them are active users, we can expect about 12,000 information messages every second during peak load. As we are dealing with peak loads (during the evening we expect more messages than during the day as more people are watching TV in the evening), we need to build our architecture in such a way that it can easily scale up and down to handle the load. Until recently, a broadcaster needed to scale the internal IT infrastructure for these peak loads, leaving most computing power unused during off-peak times. However, in these economically-difficult times, a more cost-efficient solution is required. Hence, cloud-based solutions are the appropriate method to realize the scalability and cost-efficient requirement.

A second requirement is to handle these messages in real time such that statistics are available immediately. Furthermore, to unlock targeted advertisement, these real-time statistics must be able to segment the data into different target groups, e.g., all women between the age of 25 and 34 living in a particular area.

With these requirements in mind, we have developed an Amazon EC2 cloud-based solution, which we call the “Front End Data Store”. This component provides data to a “real-time” path which provides segmented information about the users instantaneously, and an “offline” path which stores all information into a Data Warehouse ready to be analyzed with in-depth CRM tools.

*Fig. 3* gives a more detailed overview of the components in each path.

14. <http://www.bisnode.com>

15. <http://www.salesforce.com>

## Real-time path

The real-time path gives real-time information about the consumers and the programmes they are watching. By combining the user information with the socio-demographic data as explained above, it is possible to segment this information into different groups, such as male vs. female viewers, viewers living in a particular region of the country, viewers of a particular age (or age range), and so on.

All user information arrives at the Node.JS server <sup>16</sup>. This is an open-source event-driven I/O framework that allows one to create scalable server-side scripts for processing incoming events. In fact, we use it as a REST endpoint on which the clients can post their messages.

This client information message is converted into a string that structures the channel information as well as the socio-demographics of the user. For example the string

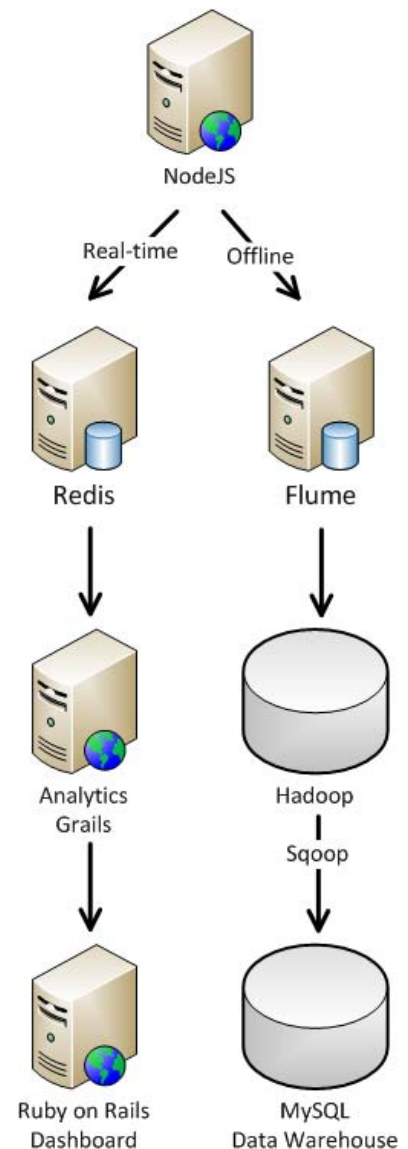
```
VRT_MALE_AGES_25_TO_34_LIMB
```

... structures the information that the user watching the channel VRT is a male between the age of 25 and 34, living in the province "Limburg". This string key is passed on to the Redis System.

Redis <sup>17</sup> is an in-memory key-value store. As such, it can keep track of many different keys in a scalable way. Each key corresponds to a numeric value, containing the number of actual persons that map to that particular key.

The Grails <sup>18</sup> component samples the information from Redis and logically combines the information from the different key-value pairs. For example, Grails can combine the information from all male users in all regions.

Finally the Dashboard created in Ruby on Rails visualizes the information of the Grails in such a way that the Dashboard user can select the data he wants to visualize in real time.



**Figure 3**  
**Front End Data Store**  
**Architecture**

## Offline path

In the offline path, all information from a user is consolidated and stored in a data warehouse for further processing. The Node.JS writes the user data to a log file, which is processed asynchronously by Cloudera Flume <sup>19</sup>. Flume is a log-processing system that works like a queue system. It also performs an initial clean-up of the data and passes it on to the Hadoop <sup>20</sup> store.

All information is stored in an Apache Hadoop system. This component allows reliable, scalable and distributed storage of the information and is a so-called NoSQL system [9]. Within the Hadoop component, we also perform data compacting. For example, if a user is watching a particular TV show from the beginning to the end, we only need to store the fact that the user has watched the complete show, and not every independent message received every five seconds.

16. <http://nodejs.org/>

17. <http://redis.io/>

18. <http://www.grails.org/>

19. <http://www.cloudera.com>

20. <http://hadoop.apache.org/>

The offline path could actually end with the Hadoop system as this contains all the information and hence is the Data Warehouse. However, we have noticed that most statistical analytic tools cannot handle NoSQL systems. Tool vendors such as SAS <sup>21</sup> require a relational SQL interface to a Data Warehouse. In order to realize that we use Cloudera Hadoop Sqoop to convert the Hadoop store into a MySQL database. Again, this is an asynchronous process.

## Conclusions

In this article, we have discussed two systems that allow broadcasters to attract media consumers on the second screen, and to collect information about these consumers in order to bootstrap a Customer Relationship Management approach in the media. The system uses the second screen – i.e., the personalized mobile internet-connected device of the user – to identify and track his or her viewing habits. By enriching this information with socio-demographic data, the broadcaster can get to know its consumers and employ it in real time for additional benefits, such as segmented and targeted advertisements.

Our proposed architecture to support this system is built by keeping the particularities of a broadcast context in mind. Peak load and scalability were taken into account from the start of system construction by implementing the front-end system on cloud services.

The system has already been implemented and will go live during the summer of 2011 (during a live TV show). It will collect information from the TV viewers in a declarative method and partly in a measured approach using audio watermarking. Segmented statistics will be collected and used during the live show.

21. <http://www.sas.com/>



**Robbie De Sutter** received an M.Sc. in Computer Science in 1999 and a Ph.D. in 2006 from Ghent University, Belgium. After a brief period as a full-time professor at the University College of West Flanders, Belgium, he is currently involved in the information management research group at VRT-medialab. The research in this group focuses on the creation, adaptation, integration and exploitation of audiovisual descriptive metadata, and investigates automatic content analysis tools and improved indexing and search algorithms for audiovisual materials.

Since 2010, Dr De Sutter has been vice-chair of the EBU Expert Community on Metadata and is involved in the ECM/SCAIE workgroup which studies and evaluates automatic information extraction analysis tools for audiovisual production and archived content. Recently, his research has focused on realizing the CRM methodology in a media context. He has particularly focused on the opportunities the second screen brings for broadcasters in order to (re)establish a direct connection with their customers.

**Lode Nachtergaele** started his career in 1989 at IMEC in Belgium. In 2001, he became Project Manager Multimedia and was in charge of the eVRT MPEG project. Shortly after he joined the Flemish broadcaster VRT as Business Architect. In 2008, he was in charge of the mobile initiatives at VRT starting with the beta version of the news website “m.dere-dactie.be” and the sports site “m.sporza.be”.

Since 2009, Mr Lode has been working as a researcher at VRT-medialab. At first he worked for the Internet cluster, and afterwards for the second-screen application “de koers” that was launched in 2010 on the Internet and then in full HTML5 on the Apple iPad. Currently he is leading the MediaSquare project team at VRT-medialab. The aim of the project is to pioneer a second screen web app to help prime-time content consumers to interact digitally (express emotions, have a conversation, vote, poll, play along with, ...) on an opt-in basis.





## References

- [1] Shan L. Pan and Jae-Nam Lee: **Using E-CRM for a Unified View of the Customer**  
In Communications of the ACM. Vol. 46, No. 4, April 2003
- [2] Jill Dyché: **The CRM Handbook: A Business Guide to Customer Relationship Management**  
Addison-Wesley Professional, 2001
- [3] Konstantinos Tsipstis: **Data Mining Techniques in CRM: Inside Customer Segmentation**  
Wiley, 2010
- [4] Paul Greenberg: **CRM at the Speed of Light, Fourth Edition: Social CRM 2.0 Strategies, Tools, and Techniques for Engaging Your Customers**  
McGraw-Hill Osborne Media; 4<sup>th</sup> edition, 2009
- [5] Theodoros Bozios, Georgios Lekakos, Victoria Skoularidou, Kostas Chorianopoulos: **Advanced Techniques for Personalized Advertising in a Digital TV Environment: the iMEDIA System**  
Proceedings of the 2001 eBusiness and eWork Conference, Venice, Italy, pp. 1025-1031, 2001
- [6] Jeongyeon Lim, Munjo Kim, Bumshik Lee, Munchurl Kim, Heekyung Lee and Han-kyu Lee: **A target advertisement system based on TV viewer's profile reasoning**  
Multimedia Tools and Applications, Vol. 36. No. 1-2, pp. 11-35, 2008
- [7] Mitchell D. Swanson, Bin Zhua, Ahmed H. Tewfika and Laurence Boney: **Robust audio watermarking using perceptual masking**  
Signal Processing, Vol. 66, Issue 3, pp 337-355, 28 May 1998
- [8] Pedro Cano, Eloi Battle, Emilia Gomez, Leandro de C.T. Gomes and Madeleine Bonnet: **Audio Fingerprinting: Concepts And Applications**  
Proceedings of 1st International Conference on Fuzzy Systems and Knowledge Discovery, 2002
- [9] M. Stonebraker: **SQL databases v. NoSQL databases**  
Communications of the ACM, Vol. 53, Issue 4, April 2010

*This version: 15 June 2011*

---

Published by the European Broadcasting Union, Geneva, Switzerland

ISSN: 1609-1469

Editeur Responsable: Lieven Vermaele

Editor: Mike Meyer

E-mail: [tech@ebu.ch](mailto:tech@ebu.ch)



**The responsibility for views expressed in this article  
rests solely with the authors**