seevl: mining music connections to bring context, search and discovery to the music you like

Alexandre Passant

Seevl Ltd,
c/o Digital Enterprise Research Institute,
National University of Ireland, Galway, Ireland
info@seevl.net

Abstract. seevl mines music connections from the Web to bring context, search and discovery for the music you like, directly within your favorite applications. We rely on the latest Semantic Web / Linked Data technologies in order to (1) aggregate, interlink and consolidate data, (2) deliver meaningful search and discovery services and (3) build plug-ins providing those features in existing applications.

1 System description and architecture

seevl mines music connections from the Web to bring context, search and discovery for the music you like, directly within your favorite applications. To do so, we aggregate multiple data sources from the Web and unify them as Linked Data, building a giant graph of musical entities (Artists, Bands, Labels, Genres, etc.). We then provide additional services (such as search and recommendations) on top of this aggregated data. Our current focus is on artist biographies and fact-sheets (genre, label, influences, etc.), and additional information is to be added at a later stage.

The data sources (MusicBrainz, Wikipedia, Freebase, BBC, NY Times) can be already in RDF (such as Freebase) or are translated to RDF during an extraction process (such as Wikipedia). In both cases, artist data is mapped to the Music Ontology[1] so that heterogeneous sources are eventually represented with the same data model, making the deployment of new services easier.

Once collected, the data is stored in a dedicated RDF store (powered by OpenLink Virtuoso) and hosted on a EC2 cluster instance. This provides us with scalability and the benefits of EC2 architecture (load-balancing, elastic cache, etc.). On top of the data, we developed a dedicated data browser, available at `http://data.seevl.net` and built using django[2], that let people explore the data available, and provides recommendations for every artist in the database. A key feature of these recommendations is their explanation aspect, so that a user can understand why one artist is suggested when browsing another one.

[1] `http://music-ontology.com`
[2] `http://django-project.org`

A screenshot of this feature is provided in Fig. 1, where one can see the relations that link the Grateful Dead[3] and the Other Ones[4], ranging from common members to same genre and label. As shown in the Recommendation Systems literature, explanations provides valuable insights to users - and have been possible thanks to the use of Linked Data in seevl. While our focus remains music, the browser has been developed in a way that it is domain agnostic and new type of entities can be browsed and searched by just editing its configuration parameters.

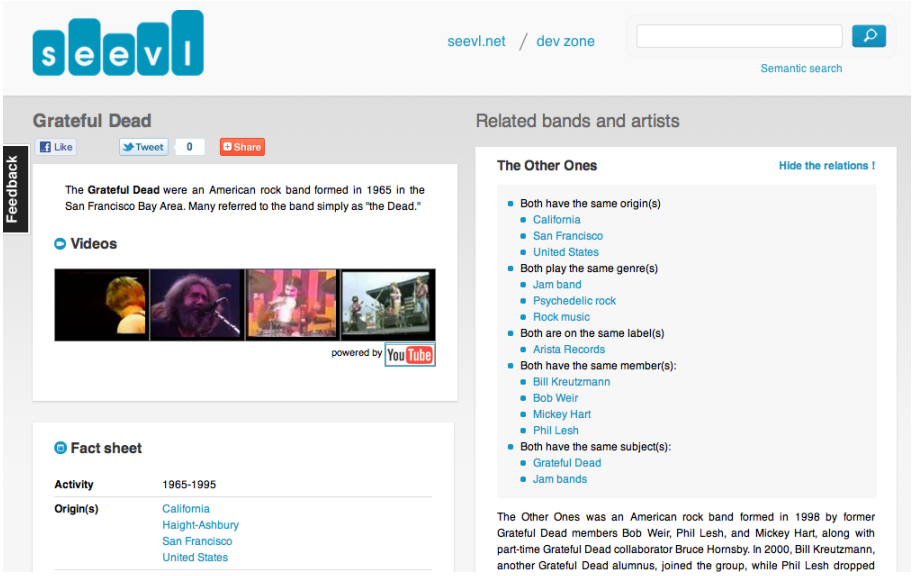

Fig. 1. Recommendations and explanations in seevl

In addition to browsing artist pages, the interface also provides a semantic search capabilities, where up to 12 features can be combined to search for artist and bands. As seen in Fig. 2, these features range from musical ones (genre, collaborations), to biographical ones (birth place, origin). This allows complex queries, such as finding "all psychedelic rock bands from San Francisco that played in the 60's"[5], which are complex to run on classical music search engines and related apps. Both the browsing pages and the search ones are rendered through a set of SPARQL queries, executed over the RDF store using the SPARQL protocol. A video of these search and browsing capabilities is available at `http://vimeo.com/24751799`

[3] `http://data.seevl.net/entity/HnMrsqGy`
[4] `http://data.seevl.net/entity/HnMrsqGy`
[5] `http://data.seevl.net/entity/?genre=6Au4ENbt&origin=xHySCHWQ&activity.time.start=1960-1969`

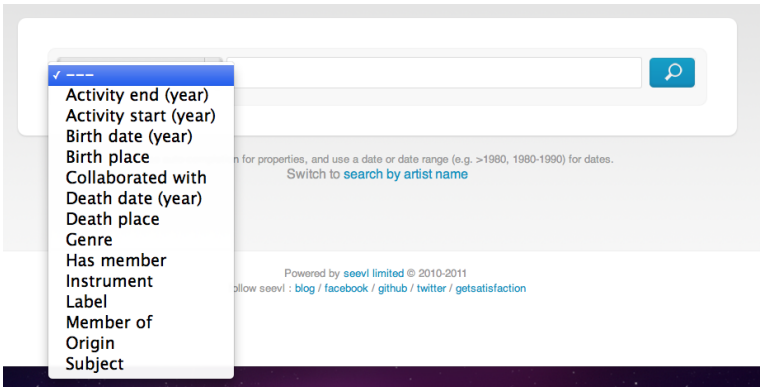

Fig. 2. Semantic search interface

Further, in order to build applications on top of this, we enable Content-Negotiation[6] on `http://data.seevl.net` and provide all the data as JSON-LD[7]. However, we require developers to get a developer key in order to access this data, and we partner with 3scale[8] to provide the required architecture (developer keys, plans, metrics, etc.)

A complete architecture overview is provided on Fig. 3 below.

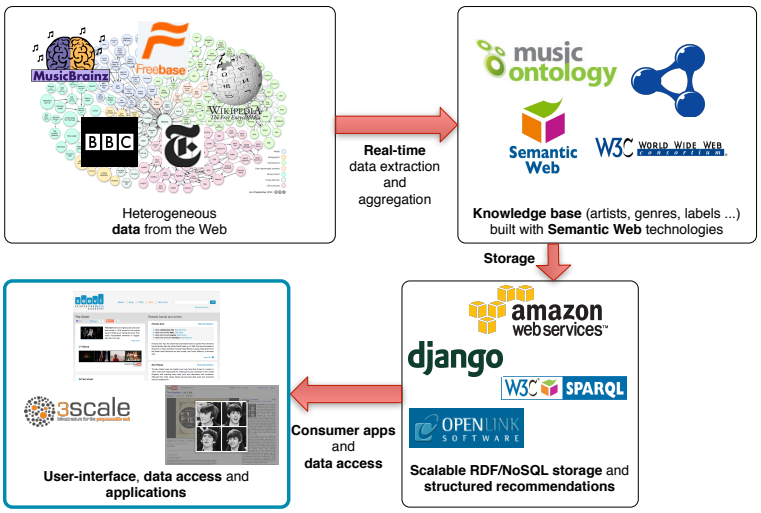

Fig. 3. The seevl architecture

[6] `http://en.wikipedia.org/wiki/Content_negotiation`
[7] `http://json-ld.org`
[8] `http://3scale.net`

2 Seevl for YouTube

To leverage the features provided by this architecture in consumer-facing applications, we first built seevl for YouTube. This Chrome extension[9] delivers biographies, fact-sheet and recommendations for every music video on YouTube. When a music video is played and the plug-in is installed, we first use heuristics to identify the artist name from the page, and then call `http://data.seevl.net` to retrieve related information – combined with the YouTube API to get videos of related artists. Information from `http://data.seevl.net` is retrieved as JSON-LD and directly processed by the extension that renders it in the user-interface (Fig. 3). The plug-in also enables the semantic search feature, so that one can build advanced playlists (e.g. from the previous query) directly within YouTube and enjoy the related videos, without have to run the query on a website (or by googling to find bands) and then switching back to YouTube.

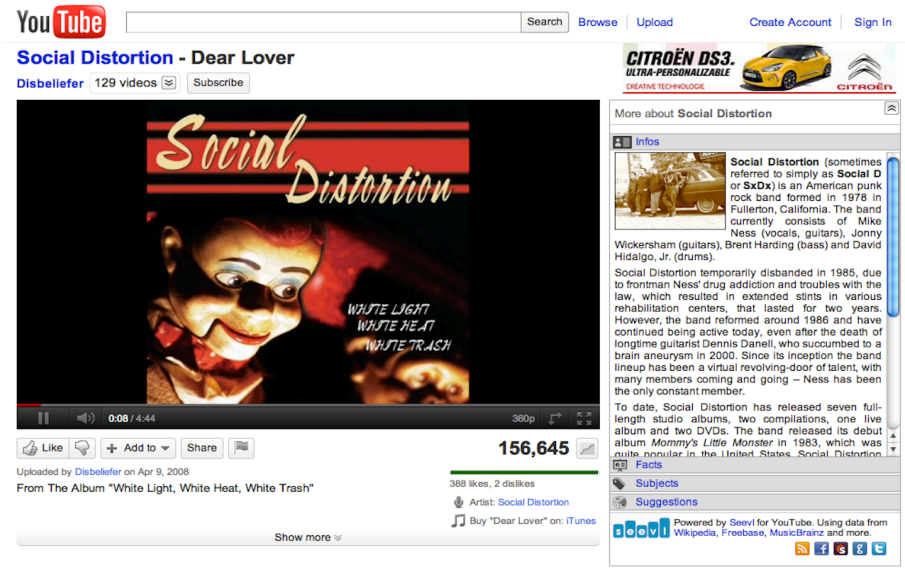

Fig. 4. seevl for YouTube

3 Design choices

Several design considerations about seevl are worth noting:

– **A fully Semantic-Web based architecture**. Every part of seevl is built using Semantic Web / Linked Data technologies. Data is modelled as RDF

[9] `https://chrome.google.com/webstore/detail/ifkepnnmiibjoocopfhnglmcoidfeaho`

from heterogeneous sources, stored in a RDF store, and rendered through SPARQL queries. We found that the graph-model provided by RDF provided us with the most agile way to build the system and make changes when required (such as new properties for an artist).

– **Use of well-known ontologies**. We decided not to reinvent the wheel but reuse - and extend when needed - existing ontologies. We rely on the Music Ontology - that we extended in some parts (e.g. representing activities as event, allowing for artist that played at several intervals, such as bands reunions) - to represent all the music-related features of artists. Other models include DublinCore[10] and SKOS[11].
– **Content-negotiation and JSON-LD**. This enables a *WYSIWYM What You See Is What You Mean* representation of data, since we expose the data as it is initially modelled, albeit using a simple JSON serialisation that is easy to understand for developers. In addition, using content-negotiation rather than a separate API model saves costs since we have a single layer to maintain between users (humans and machines) and our data. That largely simplifies the deployment process, and minimises overhead. Also, every new feature is immediately available from both side with no added cost.

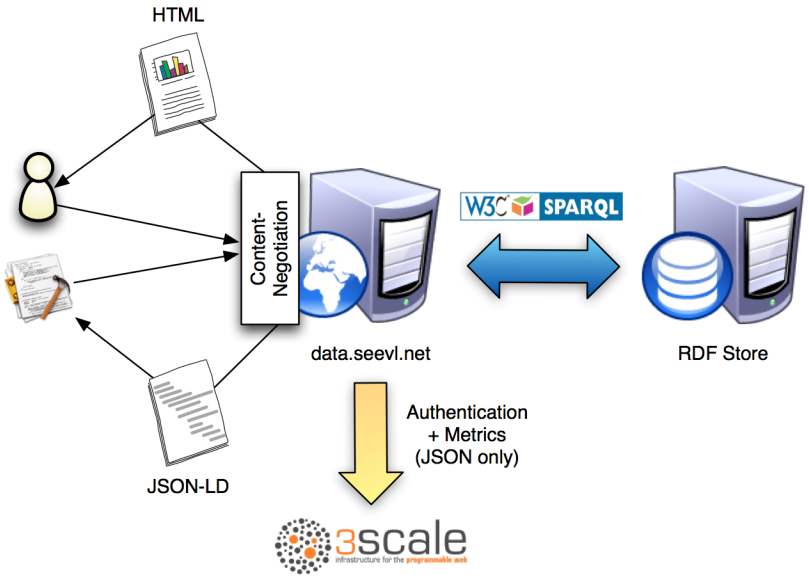

Fig. 5. JSON-LD and content-negotiation

[10] http://dublincore.org/
[11] http://www.w3.org/2004/02/skos/

– **Think of entities, not triples**. The seevl interface - and data access with JSON-LD - is *entity-centric* rather then *triples-centric*. This facilitates the browsing and understanding of data, whether it is for end-users or developers. Triples may be a natural fit when modelling data, not necessarily when rendering it.

Challenge criteria

Mandatory requirements

– *The application has to be an end-user application, i.e. an application that provides a practical value to general Web users or, if this is not the case, at least to domain experts*: seevl targets anyone that listens and/or wants to discover more music.
– *The information sources used should be under diverse ownership or control.* Data originates from several Web sources.
– *The information sources used should be heterogeneous (syntactically, structurally, and semantically).* Initial data is represented with different models and serialisations (*e.g.* Raw data in Wikipedia, RDF on Freebase).
– *The information sources used should contain substantial quantities of real world data (i.e. not toy examples).*. seevl provides information and recommendations for approximately 50K artists.
– *The meaning of data has to play a central role.* Semantic search and discovery features are provided using the meaning of the data, for instance to find "all psychedelic rock bands from San Francisco that played in the 60's"[12].
– *Meaning must be represented using Semantic Web technologies.* All entities and their attributes are represented using well-known ontologies. The seevl architecture is completely Semantic-Web based.
– *Data must be manipulated/processed in interesting ways to derive useful information.* Semantic search, contextual discovery and recommendations are derived from the data.
– *This semantic information processing has to play a central role in achieving things that alternative technologies cannot do as well, or at all.* seevl features are available thanks to the semantic representation of the data and its underlying graph model.

Additional Desirable Features

– *The application provides an attractive and functional Web interface (for human users).* seevl for YouTube has an attractive interface, directly integrated into YouTube.

[12] `http://data.seevl.net/entity/?genre=6Au4ENbt&origin=xHySCHWQ&activity.time.start=1960-1969`

- *The application should be scalable (in terms of the amount of data used and in terms of distributed components working together). Ideally, the application should use all data that is currently published on the Semantic Web.*. data.seevl.net can serve multiple and distributed applications, seevl for YouTube being the first one of a larger series.
- *Rigorous evaluations have taken place that demonstrate the benefits of semantic technologies, or validate the results obtained.* Recommendations have been previously evaluated with dbrec - that uses a previous version of seevl recommendation algorithm.
- *Novelty, in applying semantic technology to a domain or task that have not been considered before.* This is the first consumer-facing application providing recommendations and music discovery using Semantic Web technologies.
- *Functionality is different from or goes beyond pure information retrieval.* In addition to semantic search, seevl provides recommendations and contextual discovery.
- *The application has clear commercial potential and/or large existing user base.* Recent statistics showed that the YouTube VEVO channel reached more than 60M Unique Visitors in 2011[13], showing a large user-base for our products.
- *Contextual information is used for ratings or rankings.* N/A
- *Multimedia documents are used in some way.* seevl for YouTube let users discover new videos on YouTube for related bands
- *There is a use of dynamic data (e.g. workflows), perhaps in combination with static information.* N/A
- *The results should be as accurate as possible (e.g. use a ranking of results according to context).* New artists can be browsed contextually, *e.g.* by related genre or band member.
- *There is support for multiple languages and accessibility on a range of devices.* N/A

[13] `http://tinyurl.com/vevo-youtube-july11`