APACHECON DENVER

WESTIN DENVER DOWNTOWN

APRIL 7-9, 2014

Apache Linked Data Stack in Use

Presented For The Apache Foundation By

LINUX FOUNDATION

Fusepool

APACHECON
DENVER
WESTIN DENVER DOWNTOWN
APRIL 7-9, 2014

Building the Fusepool Platform on Apache Software.

Presented For The Apache Foundation By
LINUX FOUNDATION

About Fusepool

- European Union funded Research Project

Fusepool develops an user-adaptive «Living Knowledge Pool» for product development and re-search. Compared to existing search and knowledge management solutions, Fusepool provides two core benefits: the automated transformation of content from web-harvesting and participating organizations into structured Linked Open Data format and the automated group-specific optimization of knowledge finding and matching based on transfer learning from individual users. Instead of optimizing results only individually per user, Fusepool fuses anonymised user interactions to derive optimizations for specific user groups of users. Information mining and interlinking combine text mining, feature- and entity extraction with semantic web technologies. Content classification and entity identification enable automated enrichment and interlinking of information extracted from internal as well as web-harvested 'raw' content. In addition, Linked Open Data (LOD) from hundreds of data repositories such as Eurostat or DBPedia (Wikipedia) are accessed to pool knowledge related to the information need of the user. Moreover, 'raw' content that is transformed into machine-understandable content can be published as LOD for others to reuse it.

Knowledge finding and matching refers to the semantics-aware search integrating content based on available metadata (e.g. classifications, entities) into a stream-lined application for finding and matching content to support the user's information needs. Advanced search features include refinement and filtering, query intent discovery, and proactive information gathering. In addition, recommendations provide the user with potentially relevant information and user dis/approval optimizes future recommendations. Visual analytics and graphical user interfaces present intuitively the complex information and analytical results. Users can develop and share layouts and even layouts are able to adapt to user needs based on past user interactions.

Fusepool

Linked Data Application

Some rather young members of the Apache family

- Jena
- Clerezza
- Stanbol
- Any23
- Marmotta

RDF and Linked Data

Do I need to explain?

- Serializations <-> data model
- Graphs / Triples
- IRIs / Blank Nodes / Literals
- Datasets
- Triplestore
- SPARQL
- Giant Global Graph (TimBL)
- Linked Open Data

Jena

- RDF API
- Sparql Engine
- Triple Store
    - Embedded (TDB and others)
    - Server (Fuseki)
- Reasoning
    - OWL/RDFS
    - Inference API

- RDF API
  - Multiple backends: Jena, Virtuoso, Sesame
- Framework for building RDF backed Webapps
  - Based on JAX-RS
  - TypeHandlers
  - Typerendering -> ScalaServerPages
  - Content negotiation
  - Security JAAS

- Original goal: reusable components for semantic content management
  - Enhancer
  - Entityhub
  - Contenthub
  - Reasoner
  - Ontologymanager

Or more realistically:

- Enhancer
- Entityhub
- Contenthub
- Reasoner
- Ontologymanager

- Anything to triples
- Extracts RDF from a variety of input formats
- Can be used
  - As a Java library
  - On the command line
  - Via HTTP

Apache Marmotta™

- Aims to implement the Linked Data Platform Standard
- Own Triple store: Kiwi (supports versioning, backed by SQL)
- Started from Kiwi Semantic Wiki Project (2008-2011)
- LDPath: Xpath for RDF
- LDPath Templates: Freemarker to render RDF resources

Fusepool

Fusing it together

- Extracting entities from plain text -> Stanbol Enhancer
- Authentication/Authorization -> RDF based in Clerezza
- Presenting the data -> Clerezza
- Faceted searching -> Stanbol Contenthub

What didn’t work.

Access Control

- Porting Authentication from Clerezza to Stanbol
- User Management in Stanbol
- Ensuring all stanbol modules work when security is enabled

Rendering the data

- Stanbol UI tied to Jersey
- Clerezza TypeRendering needs own JAX-RS impl (later Wink, JAX-RS 2.0)

1. Added RDF Rendering to Stanbol (using LDPath templates)
2. Removed Jersey dependency in Stanbol
3. Ported Clerezza TypeRendering to JAX-RS 2.0

Maven Archetypes

Showing development patterns

- Creating
  - Enhancement Engines
  - Statefull/-less Webapplication
- Goals:
  - Support Content Negotiation
  - Are portable accross JAX-RS implementations

ContentHub

Limit usefulness for fusepool because:

- Facet values (entities) not connected to RDF data
- Duplication of metadata in graph and SOLR
- No security by exposing SOLR endpoints
- No support for structured content
- HTTP API doesn’t speak RDF
- Hard to manage code

Enhanced Content Store

For now apache licensed on Github

- REST API to upload unstructured document
- Documents are assigned dereferenceable HTTP URI
- Enhancer executed on uploaded documents
- Documents as well as well as digested meta-data is stored to content graph
- HTTP-Meta header points to meta-data of documents
- Lucene based CRIS is configured to listen to graph changes and keep index up to date
- Faceted search exposed as RDF-REST-API

Interlinking

For now apache licensed on Github

- Framework for integrating Interlinkning Engine like Silk or Limes
- Datalifecycle taking care of
  - Transformation
  - Enhancemnet
  - Interlinking
  - Smushing

Discussion

- Do we still need language specific RDF APIs?
- How to best deal with overlapping apache projects?
- Research projects and apache communities.