chem-bla-ics

Centralized or decentralized?

Egon Willighagen

Published August 13, 2007

Citation
Willighagen, E. (n.d.). In *chem-bla-ics*. chem-bla-ics. https://doi.org/10.59350/y11ff-5he48

Keywords
Inchi, Semweb

Copyright
Copyright © Egon Willighagen 2007. Distributed under the terms of the Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Peter wondered if data should be stored centralized or decentralized, when Deepak blogged about Freebase and Metaweb. Now, I haven't really looked into these two projects, but the question of centralized versus decentralized is interesting. It's MySQL versus the world wide web; it's the PubChem compound ID versus the InChI; it's http://cb.openmolecules.net/rdf/?InChI=1/CH4/h1H4 versus `info:inchi/InChI=1/CH4/h1H4` (see RDF-ing molecular space).

Both have advantages and disadvantages (everything does). Google has a huge experience with massive data, and is the centralized version of the distributed world wide web. Personally, I tend towards the decentralized version of things. Scales better. The chemical RDF community showed some concerns about scalability of triple stores (see e.g. Taylor et al. *Bringing Chemical Data onto the Semantic Web*, **2006**, DOI 10.1021/ci050378m). Now, their tests went up to some 30M triples, which is barely enough to store the InChI, PubChem compound ID, and one chemical name.

So, how would this work for molecules then? I am leaning towards a system where one can query resources about one molecule, and work ones way through molecular space. Using KEGG, reaction databases, similarity stores, one could move from molecule to molecule, and add bits of RDF along the way, filling a local RDF store around the actual query I have in mind. For example, if I want to verify that the mass spectrum I found really belongs to the molecular structure I have in mind, I would look up in the resources I know about all triples that relate to the putative structure, and do my queries from there. That's what I would do... (and will do, but more on that later...)