



what if? ...our service could answer any question, any time?

Peter Baumann, Jacobs University, Germany

What standards and technologies should the infrastructure be based on?

Following the Unix philosophy of “do one thing and do it well” services should be modular. Service providers should be able to pick the best of breed per data category and task. “No one size fits all” – this holds for the highly complex, diverse geo data in particular. This has been a core idea of OGC and has led to its W*S world, but over the recent years has been weakened due to non-technical reasons. Too often services today compete in functionality. One example is SWE which has been established largely independent from W*S leading to a continued confusion. “All data are sensor data” does not contribute to modularity. Rather, W*S (in particular: WMS, WFS, WCS, WCPS) should be harmonized actively (and modernized considering recent findings from Computer Science).

PS: Open-source and proprietary should co-exist – favouring pure proprietary technology is just as infeasible as allowing *only* open-source.

What architectural pattern would you recommend?

Query languages have been tremendously successful due to their flexibility, boosted by scalable implementations serving huge amounts of data every day. Even the NoSQL movement meantime acknowledges the value of query languages. While query languages traditionally had a focus on administrative and business data this has changed since long – today we have versatile domain-specific languages, even standardized, for both features (SQL Simple Features) and coverages (Web Coverage Processing Service). This asset should be exploited and integrated with well-founded languages like SQL, SPARQL, etc.

What will not fly:

- REST: Generally, there was and is a misguidance in geoinformatics (not generally in Computer Science, notably!) based on the focus on REST and APIs. REST is not an architecture, but just syntax (“/” instead of “&”) – and if used as an architectural principle then it hardwires classical hierarchical file systems. Times and again the limitation has been elaborated, such as in OGC Testbed-12. We need more than hierarchies, such as sets, graphs, arrays (“datacubes”).
- WPS: essentially, it is a complex variant of the 1980s lightweight ASN.1 interface description language. WPS is semanticless (cf. WPS 1.0 introduction) and hence never will allow automatic orchestration – the plethora of papers attempting this (and failing) shows it. We need services rich in semantics to achieve higher quality of service, including performance.

What should be the main components of the infrastructure?

XPath/XQuery (can work for JSON as well!), SQL, SPARQL, etc. for metadata; WFS Query Expression for features; WCPS for coverages; it would not cost more than the existing infrastructure, but it would save immensely because new functionality does not need new coding, just new queries.

How would you organise the implementation process and make it cost-efficient?

Gathering entrepreneurial data centers for establishing lighthouse services and best practice for the various domains; put up massive (national language) training and helpdesks; strictly focus on open standards so that open-source and proprietary solutions can interoperate. In any case, only tools passing OGC conformance tests should be acceptable.