

# Data Models For Interoperability

Rob Atkinson

*social* **change**  
ONLINE

# Contents

---

- Problem Statement
- Conceptual Architecture
- Role of Data Models
- Role of Registries
- Integration of GRID and SDI

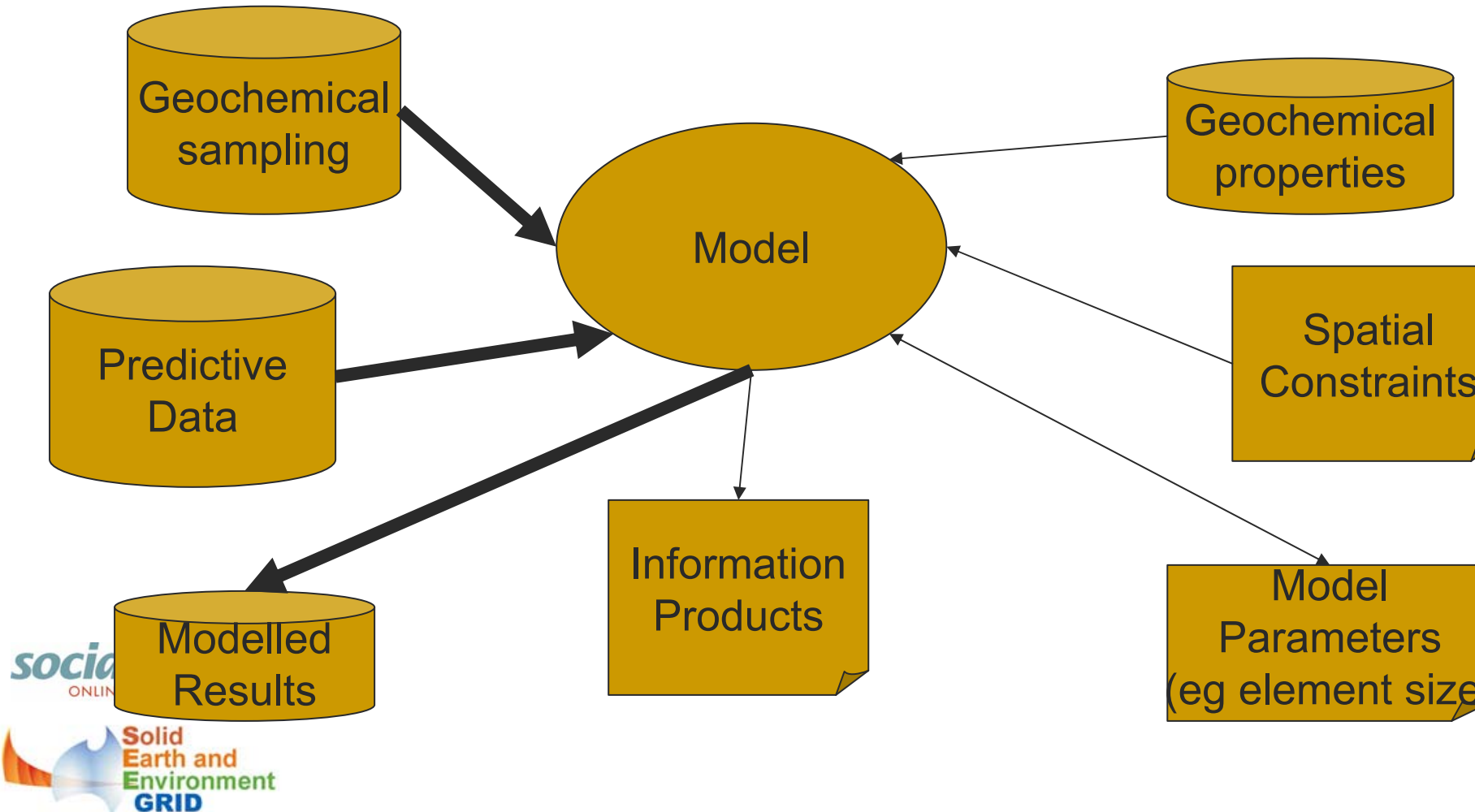
# Problem Statement

---

- How do we derive useful information from sparse sampling and many possible models of behaviour
- In the “solid earth” domain – note many similar issues in atmosphere, marine, land management, ecosystems ...
- Given that data is collected and/or managed by many different agencies

# Problem (Facets)

- Modelling – many possible inputs



# Problem (Facets)

---

- Many possible models
- May want to re-run conditions with new model, or new data
- Models may take time
- Or results may need to be archived for audit trail reasons
- Huge or trivial amounts of data....

# Architecture

---

- *“..the strategic decisions about the structure and behaviour of the system, the collaborations among the system and the physical deployment of the system” –Quatrani*

**social**change  
ONLINE

 **Solid  
Earth and  
Environment  
GRID**

# Conceptual Architecture

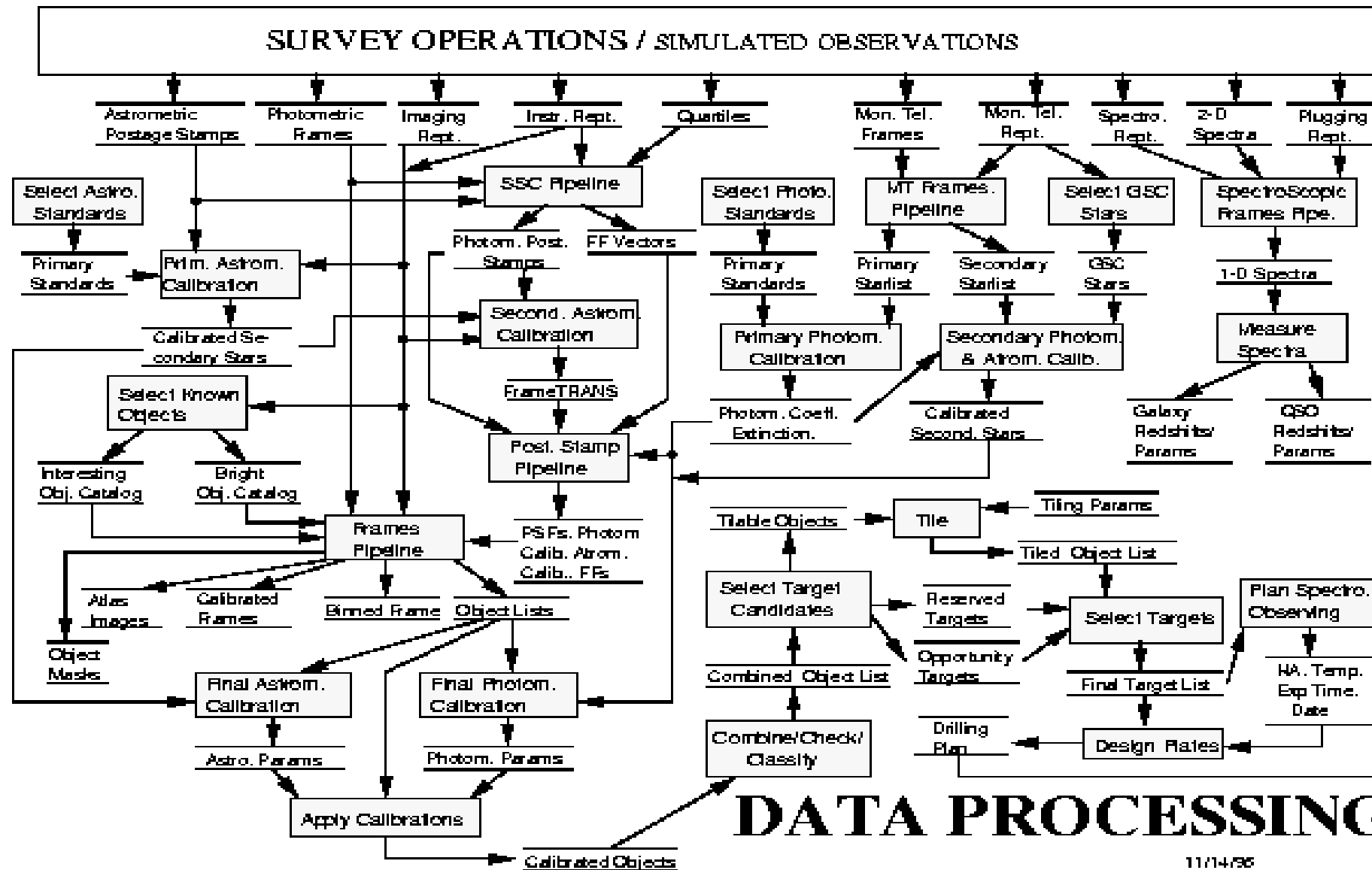
---

- GRID
- Web Services
- Differences?
- Commonalities?

*social*change  
ONLINE



# Sloan Digital Sky Survey Production System

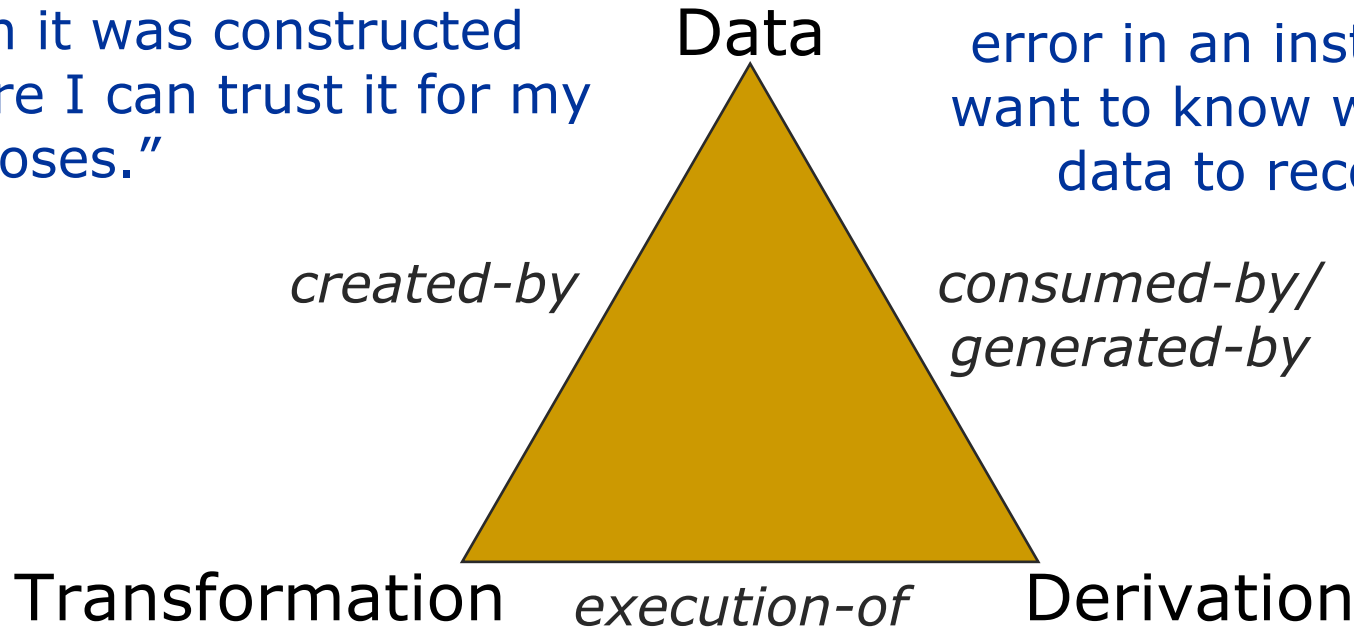




# Motivations

"I've come across some interesting data, but I need to understand the nature of the corrections applied when it was constructed before I can trust it for my purposes."

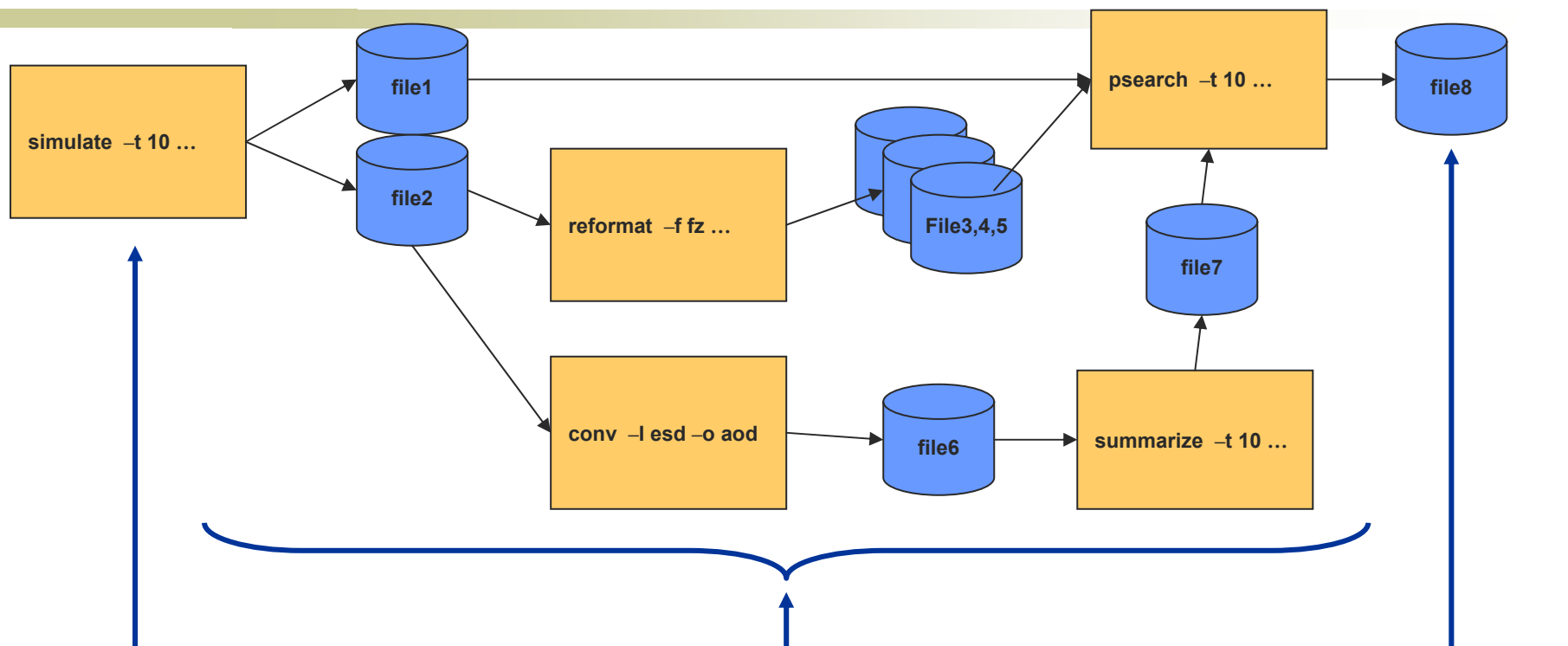
"I've detected a calibration error in an instrument and want to know which derived data to recompute."



"I want to search an astronomical database for galaxies with certain characteristics. If a program that performs this analysis exists, I won't have to write one from scratch."

"I want to apply an astronomical analysis program to millions of objects. If the results already exist, I'll save weeks of computation."

# Motivations



Patch workflow following changes

socialchange  
ONLINE



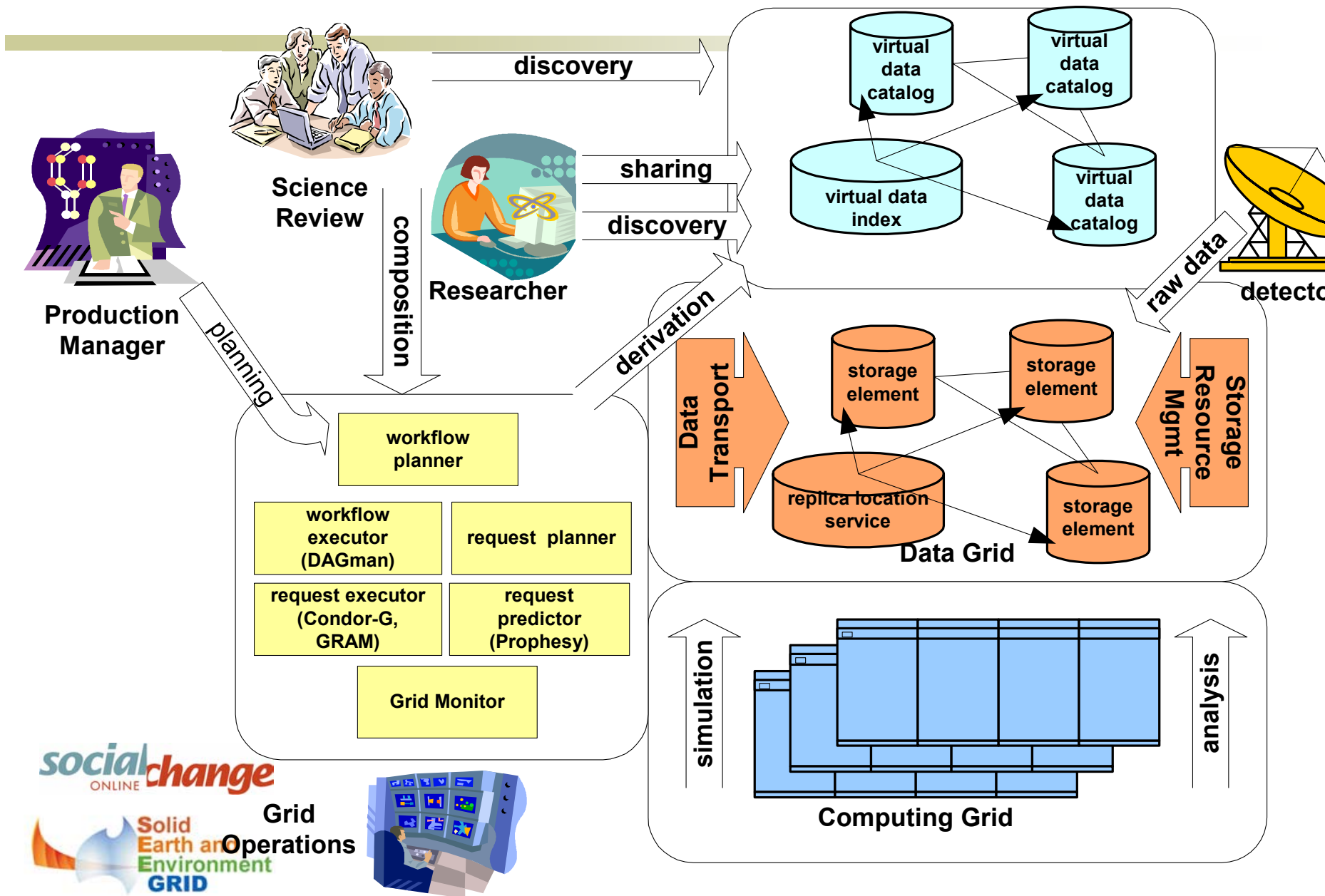
Manage workflow

Explain provenance, e.g. file8:

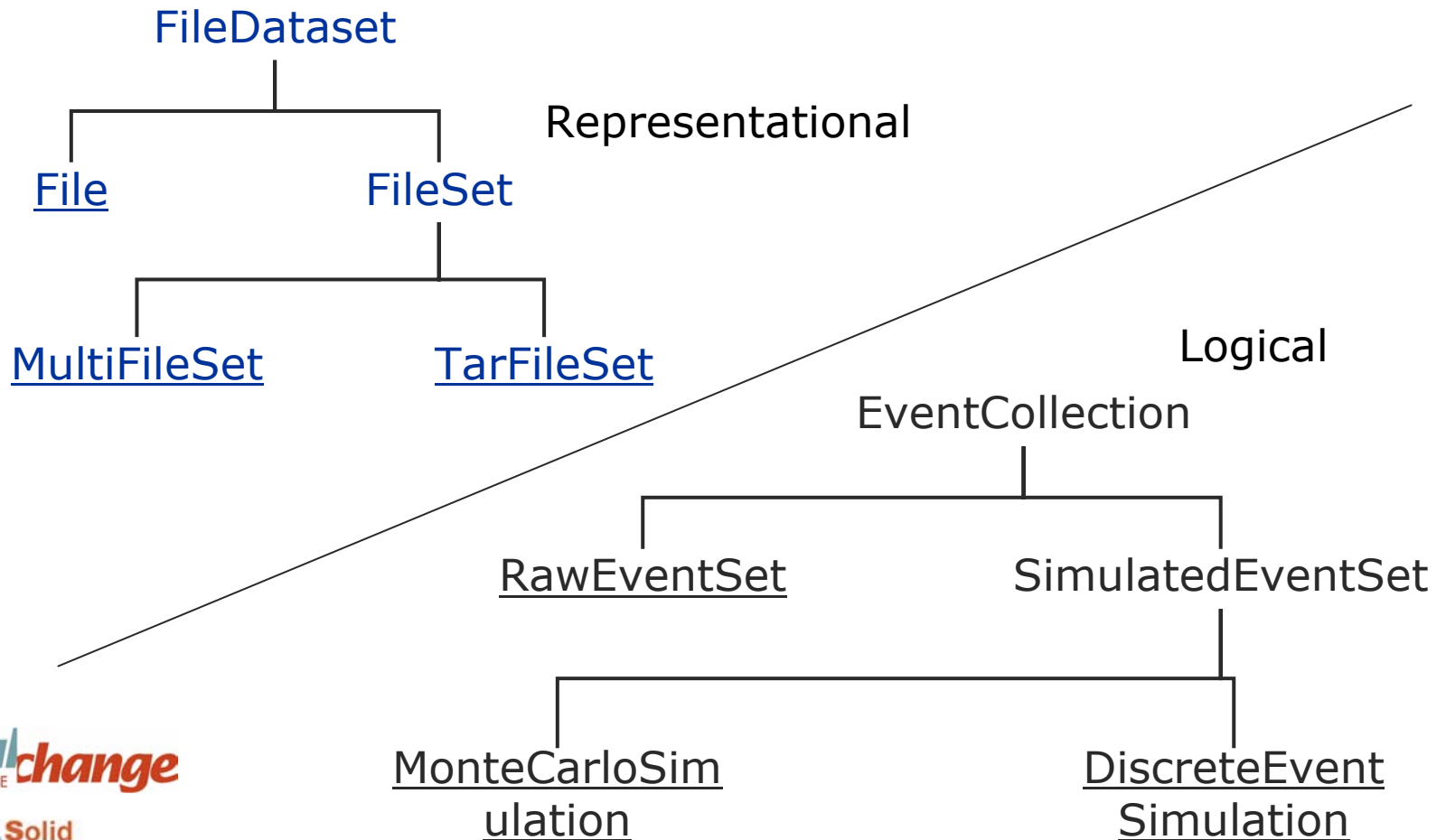
```
psearch -t 10 -i file3 file4 file5 -o file8
summarize -t 10 -i file6 -o file7
reformat -f fz -i file2 -o file3 file4 file5
conv -l esd -o aod -i file 2 -o file6
simulate -t 10 -o file1 file2
```

On-demand data generation

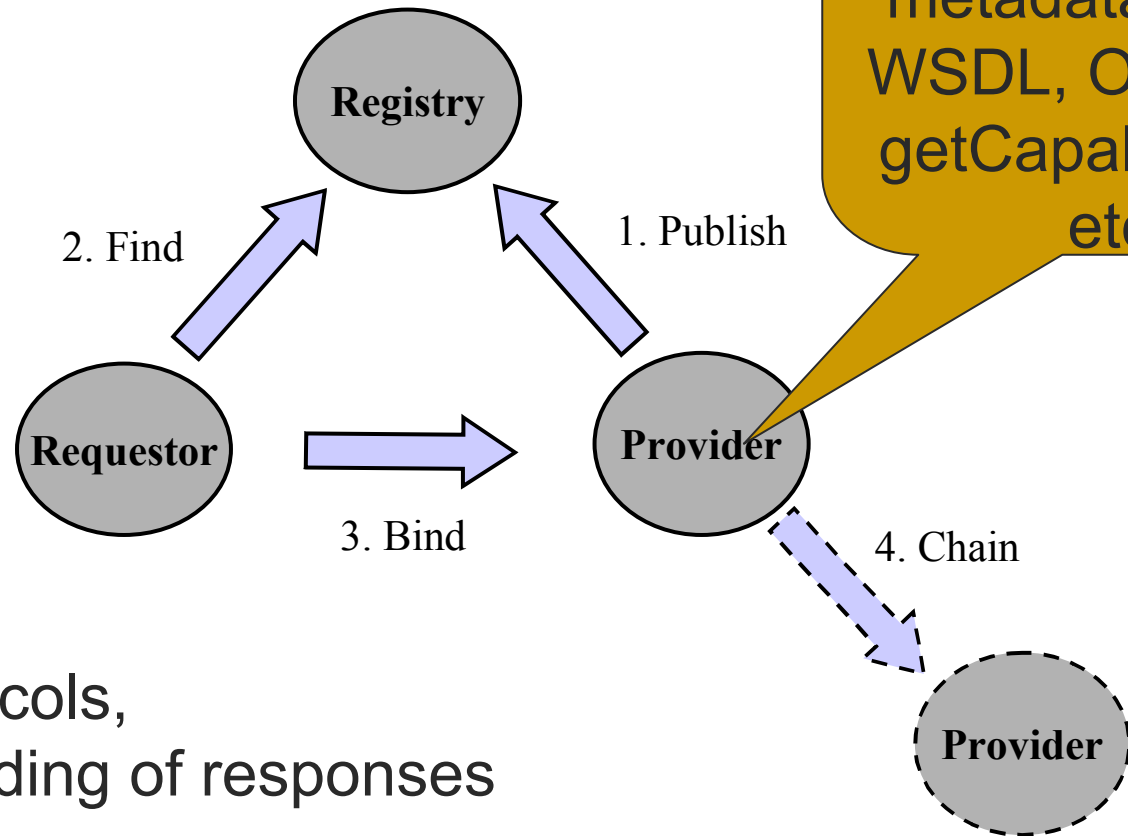
# Virtual Data Grid



# Example GRID Dataset Types

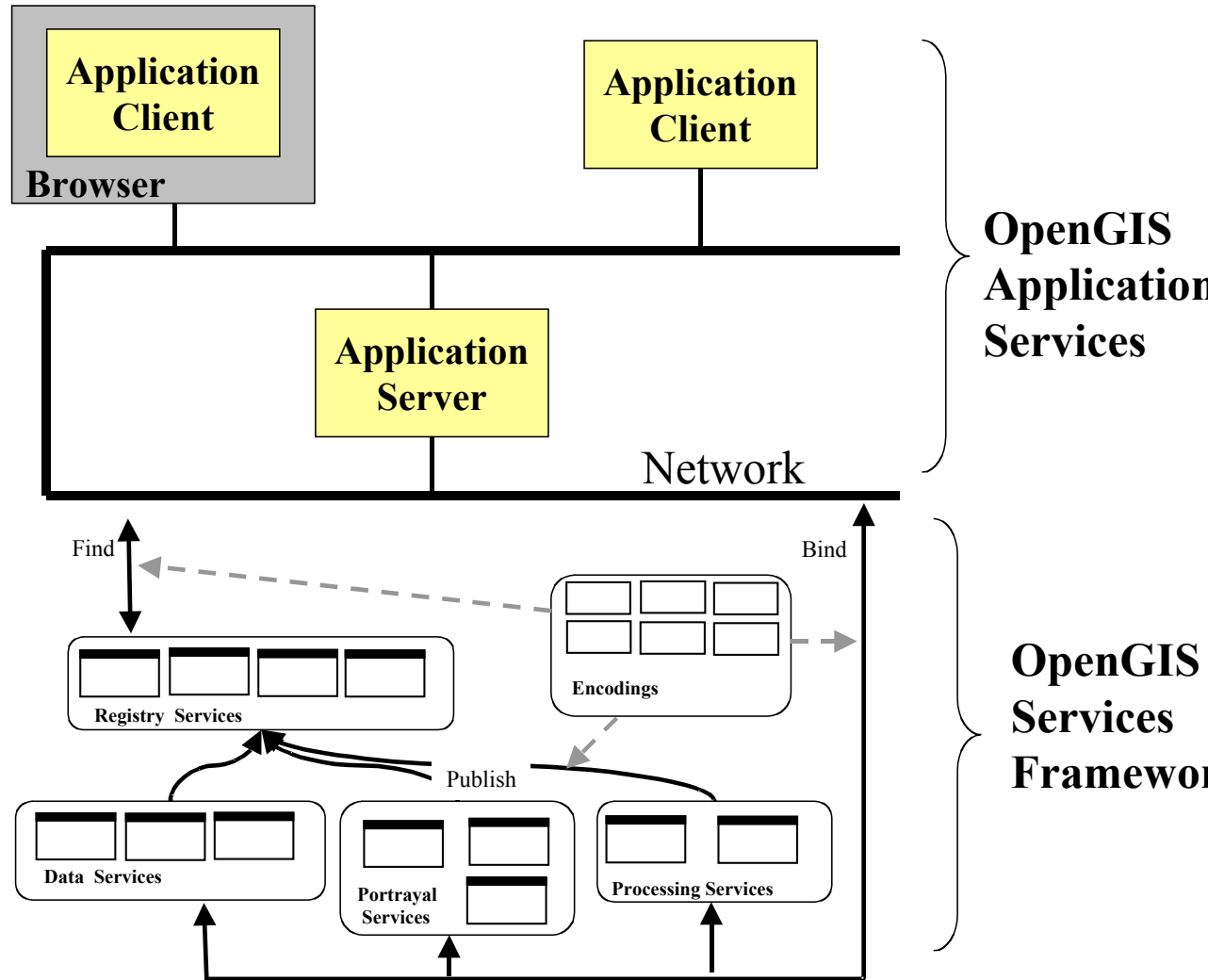


# Web Services

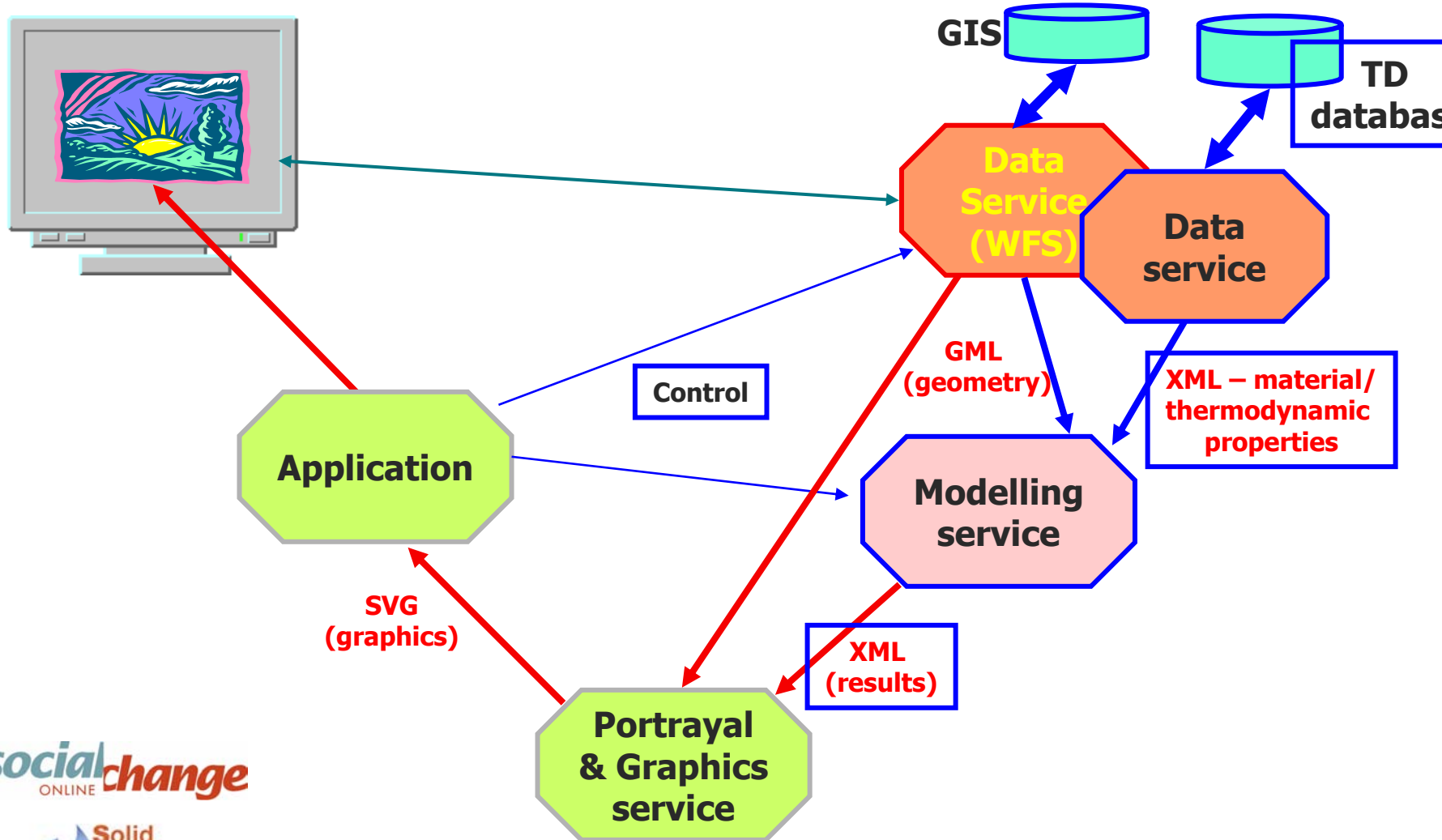


Web protocols,  
XML encoding of responses

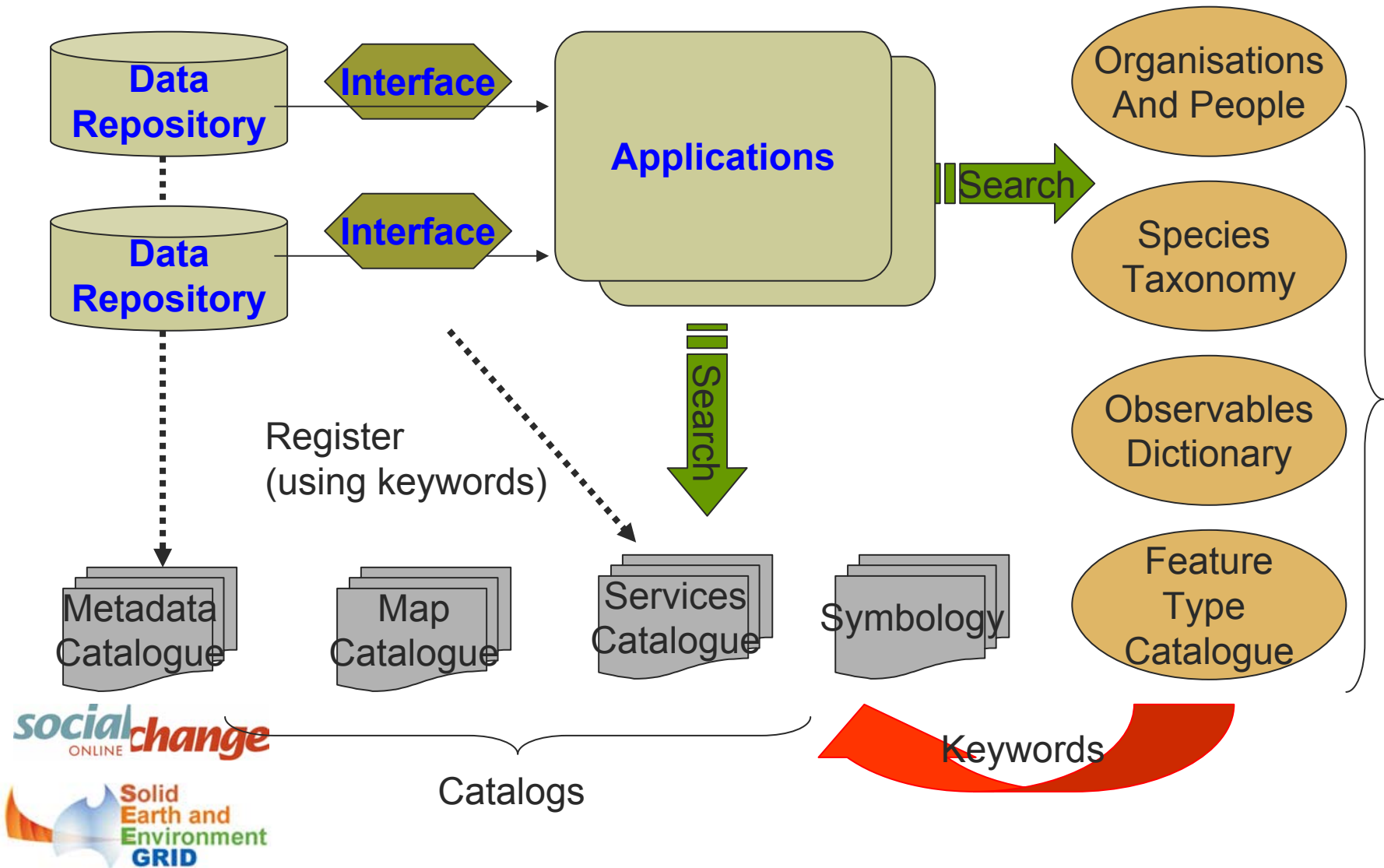
# OpenGIS Web Services



# Example



# SDI Component Model





# Atkinson's instant type hierarchy

---

- Primitives (supplied by environment)
- “standards” – allow technology implementation (GML, RDF etc)
- “profiles” – mandated common patterns (eg metadata requirements)
- “Feature Types” – domain models
- “Service Types”
- “Service Offerings” (introduces content models)

# Data Models

---

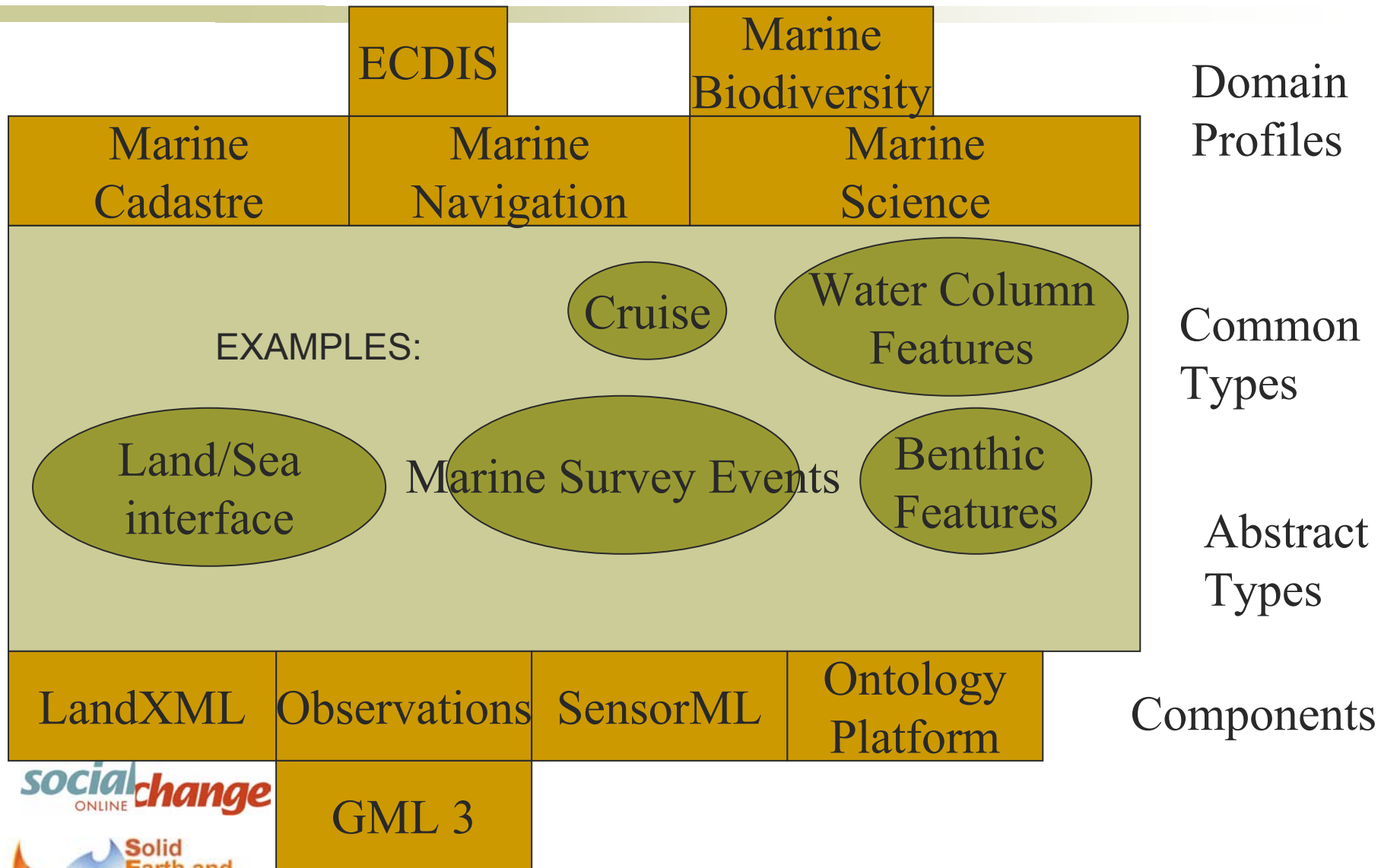
- GRID – “files” – homogeneous data model – interoperable but not flexible
- OpenGIS WS – “feature types” – implementation policies determine level of interoperability
- Can we develop a humungous data model to anticipate all needs and capabilities.....

# Component Data Models

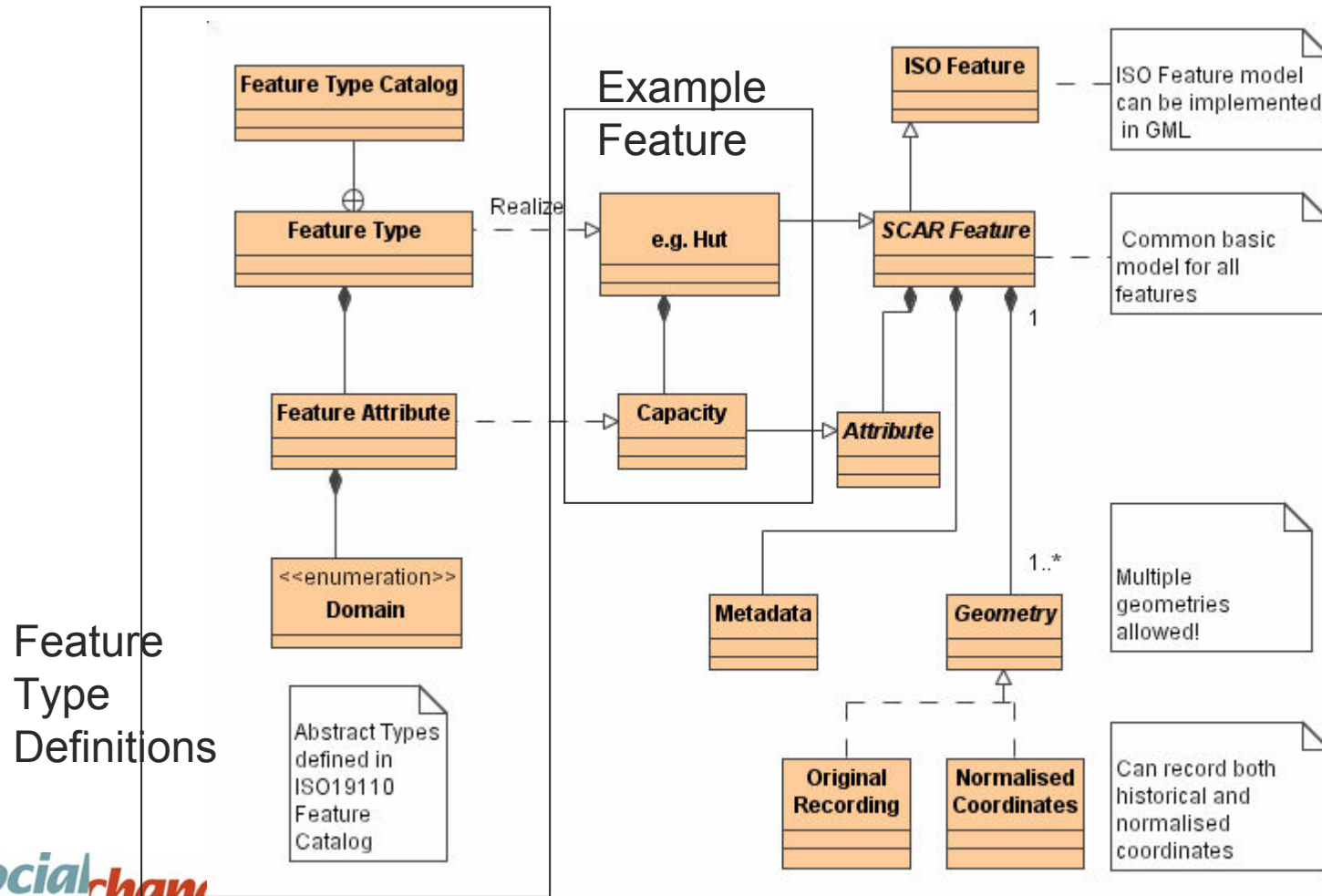
---

- Reusable sub-components
- Allow services to be created to serve reusable data
- Because a service consumer doesn't have to subscribe to the entire enterprise model!
- NB: Service offers described using “types”
  - WSDL, using XML:schema (c.f. GML)

# MarineXML Architecture



# Data Model : Feature Types



# Strong or Weak Typing

---

- Is a feature best described as:

**GeneralFeature** (attribute:type="Hut")

Or

**Building** (attribute:type="Hut", beds=3)

Or

**Hut** (beds=3)

Or

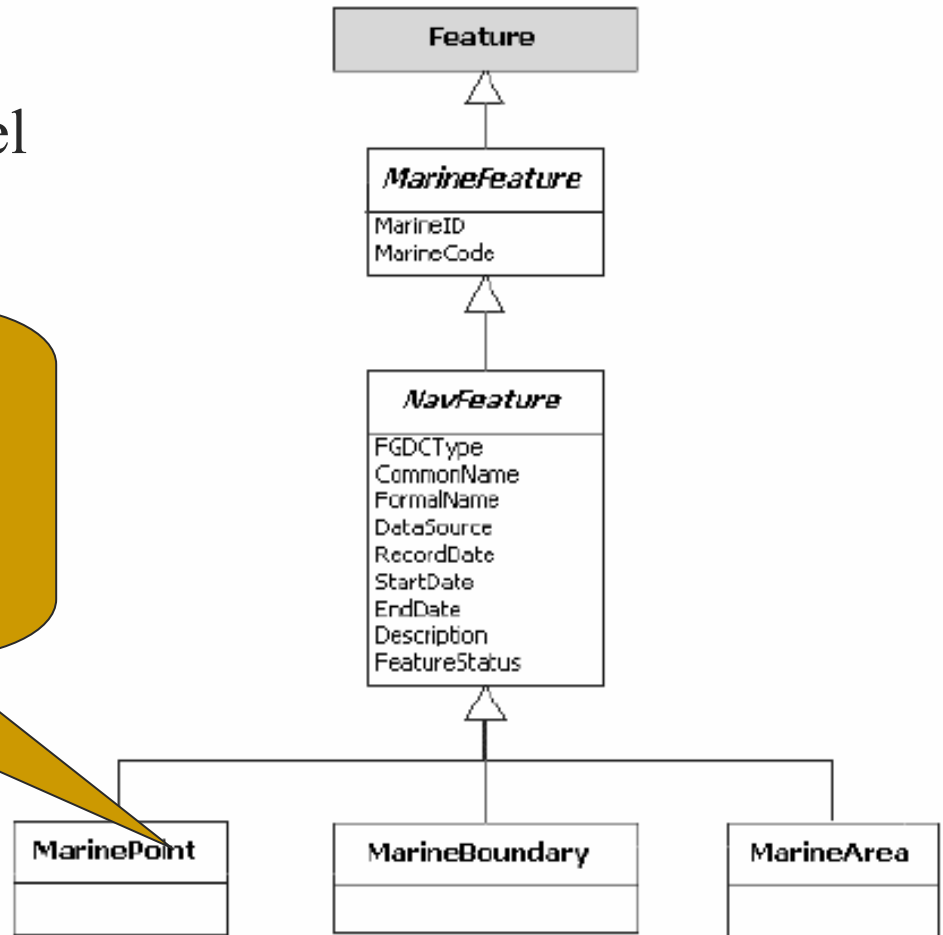
**3 Bed Hut**

- "Granularity" depends on usage requirements

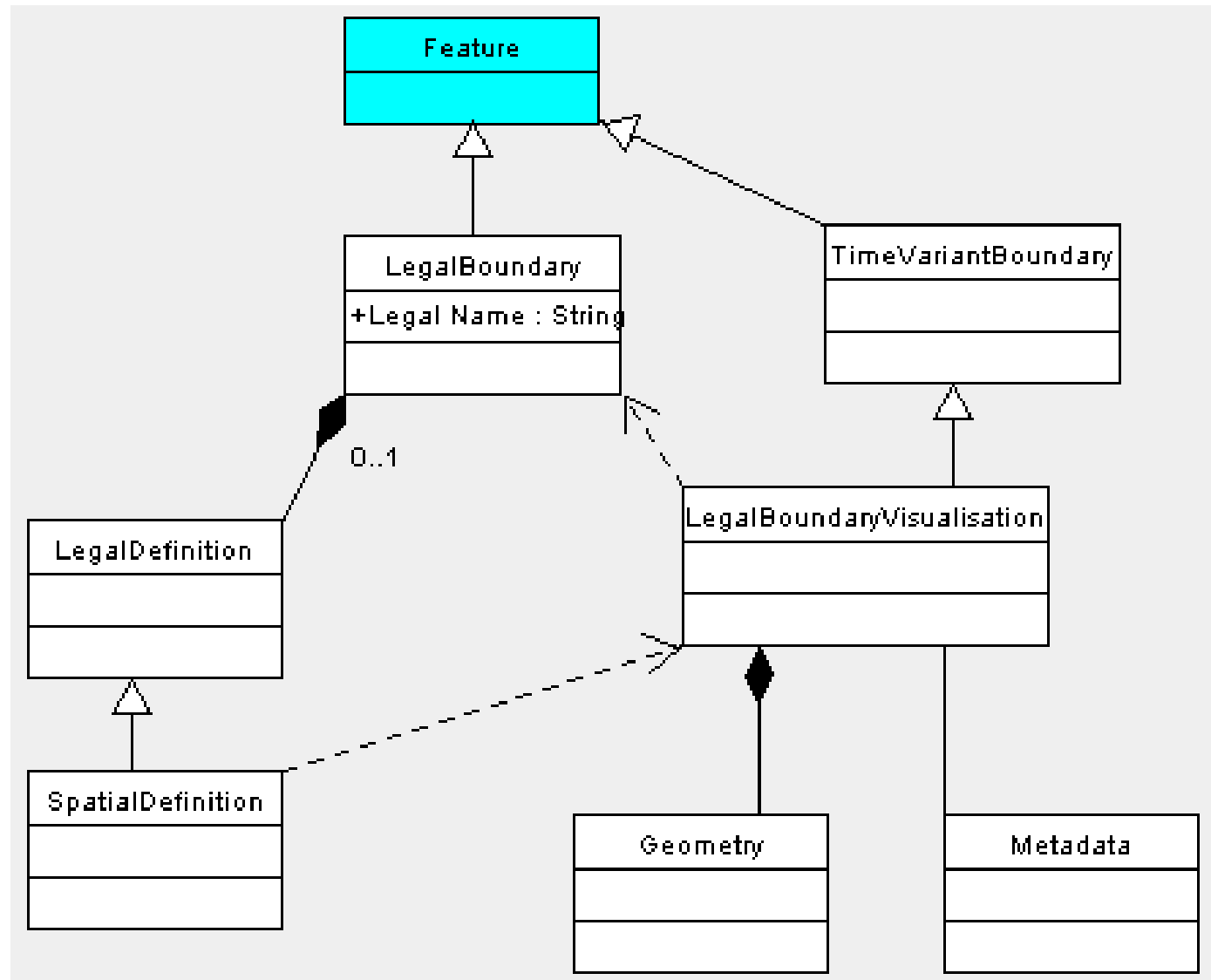
# Logical and Physical Data Models

## ArcGIS Marine Data Model

Physical Model:  
Abstraction from a  
geometry

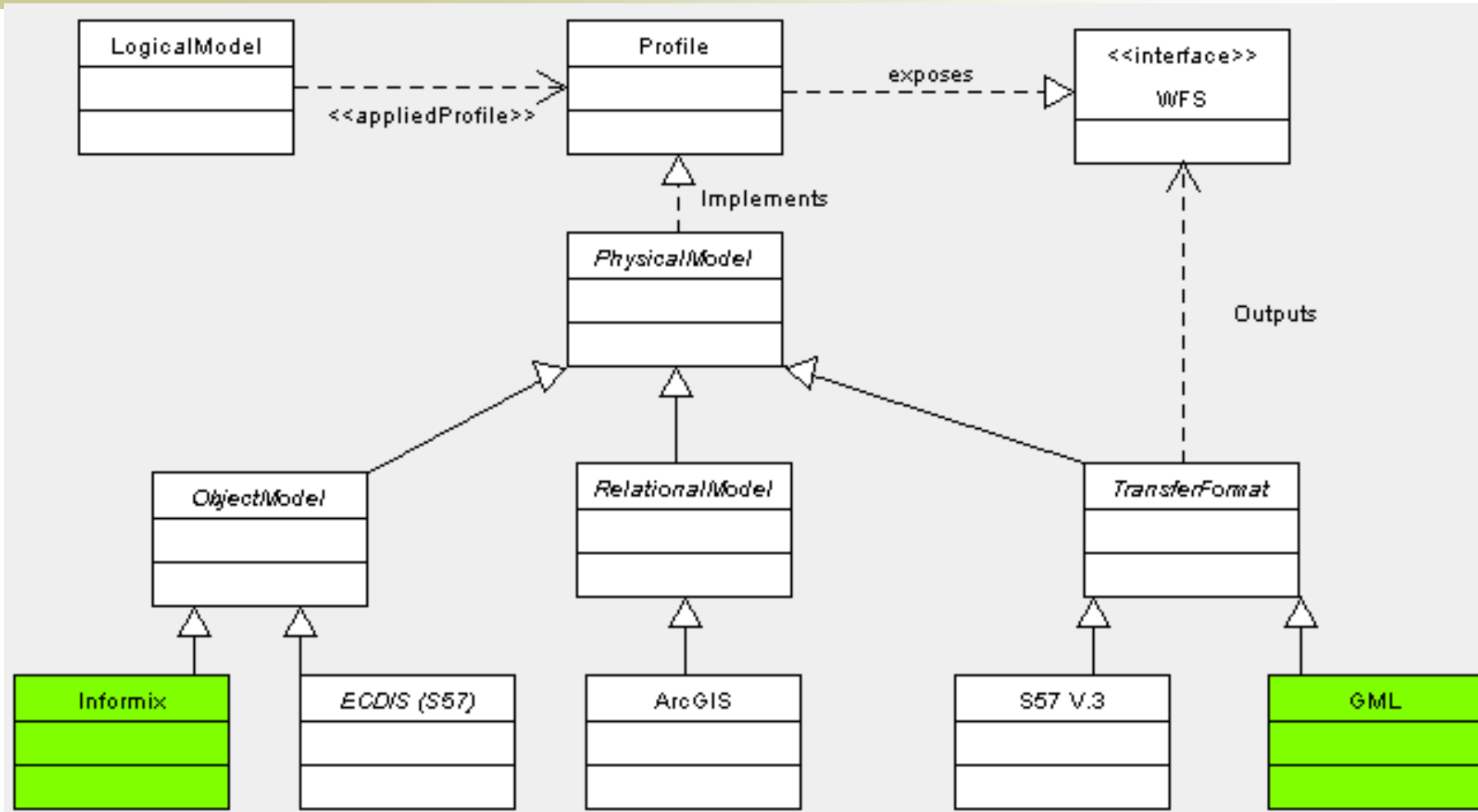


# Example Logical View

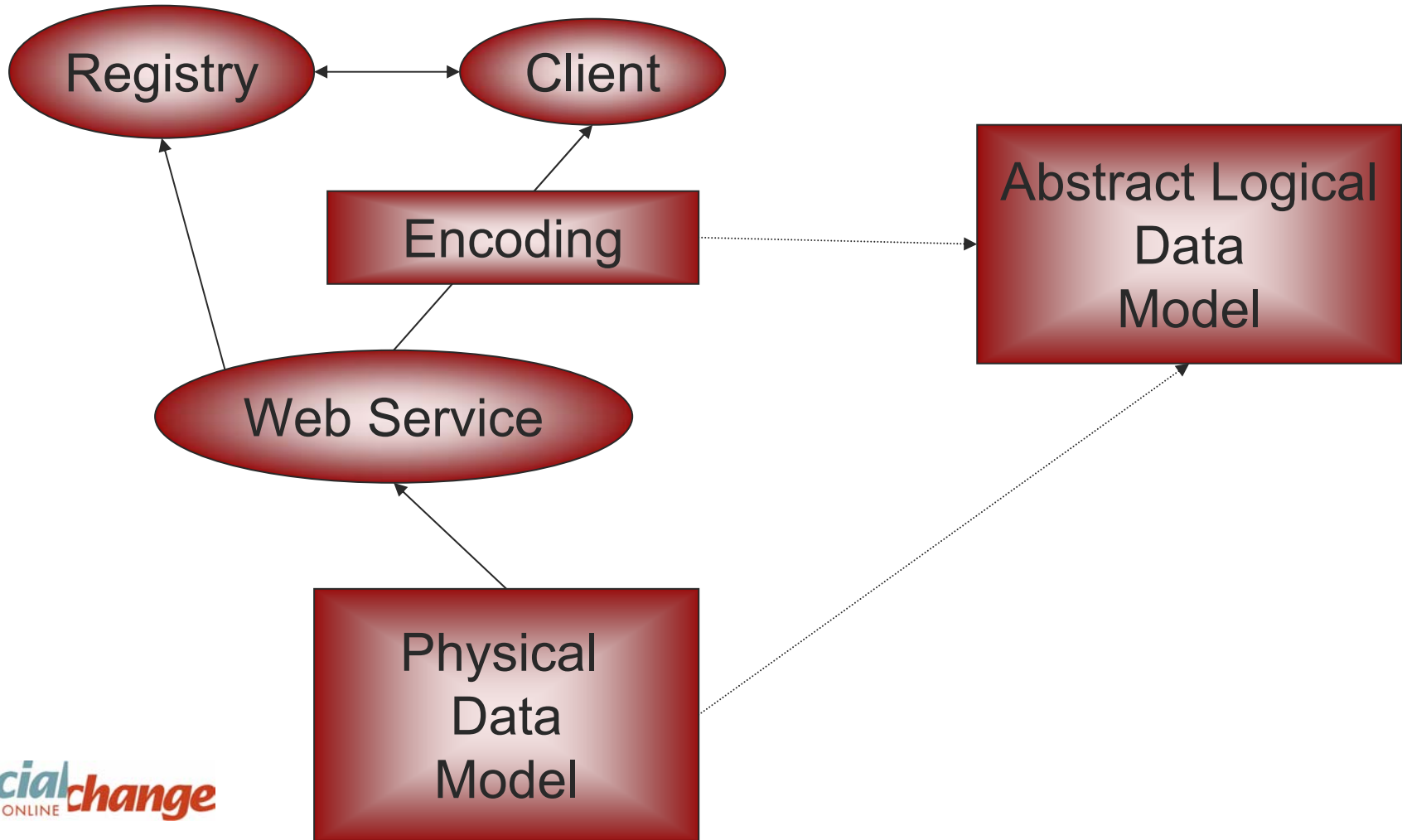




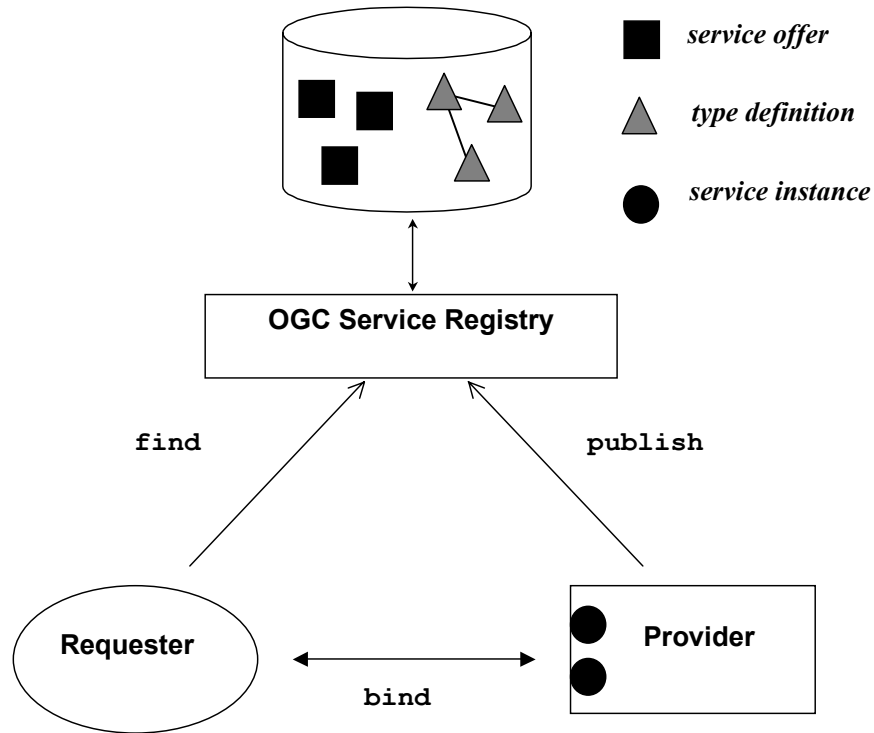
# Implementation



# SDI Information Architecture



# Role of Registries



# Role of Registries

---

- “central purpose is to store service offers and match them with service requests”
- “provide a context in which resources can be discovered and used”
- Data and service “types” are the key units of classification
- *Ipsa facto* registry design is a translation of **policy** into **technology** driven by a **system architecture** that clearly identifies **data model**.

# Registries

---

- Easy to build
  - Many implementations... little consistency
  - Not too hard to connect to if the content is valuable!
- Hard to populate
  - Technically no problems
  - Achieving critical mass
  - And being useful to future applications

**social**change  
ONLINE



# Integration of GRID and Web Services?

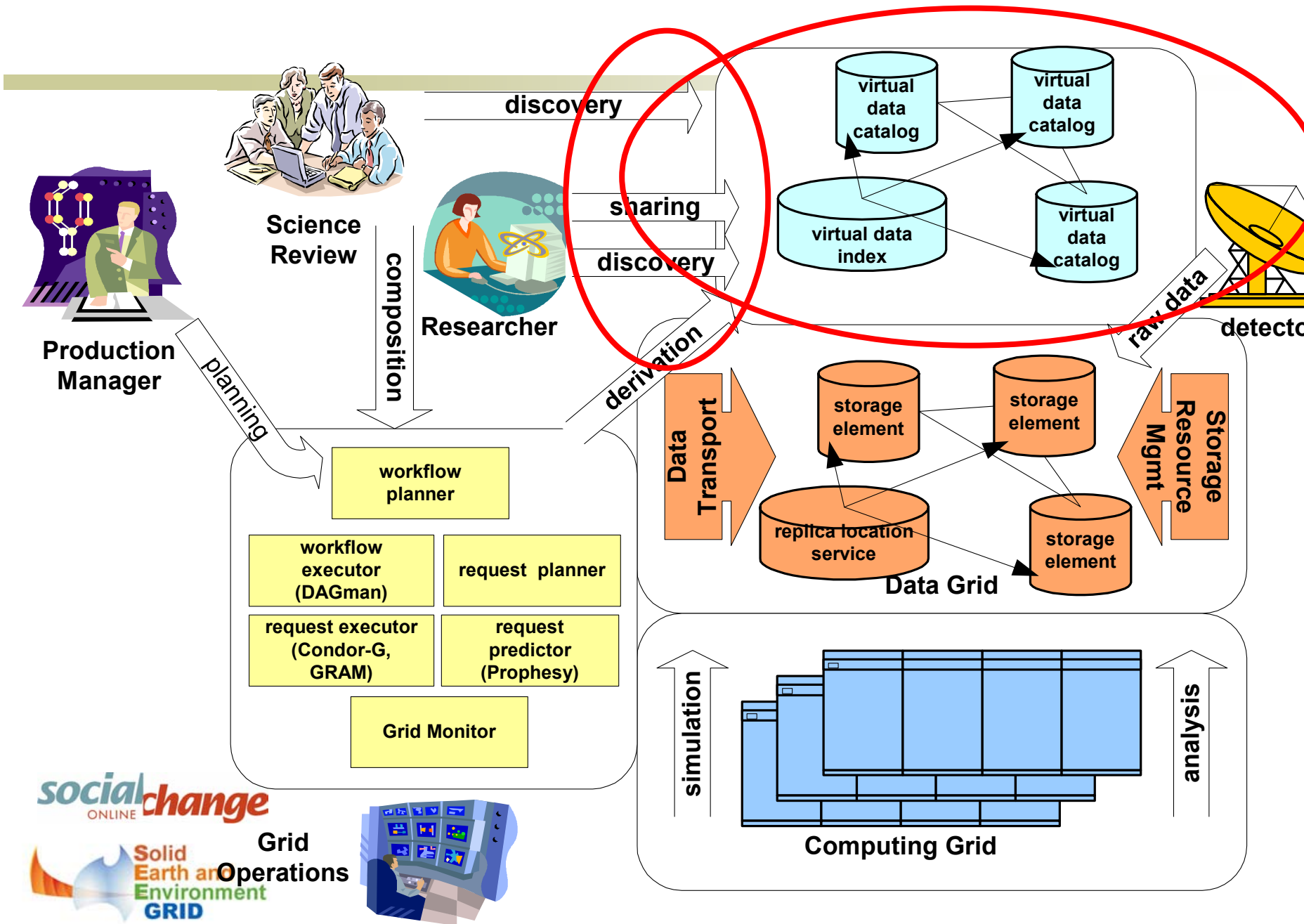
---

- GRID – pre-defined applications
  - Computational and data storage
  - A range of outputs
  - Control functions
- Web Services :
  - Initiation
  - Exploitation
  - Discovery

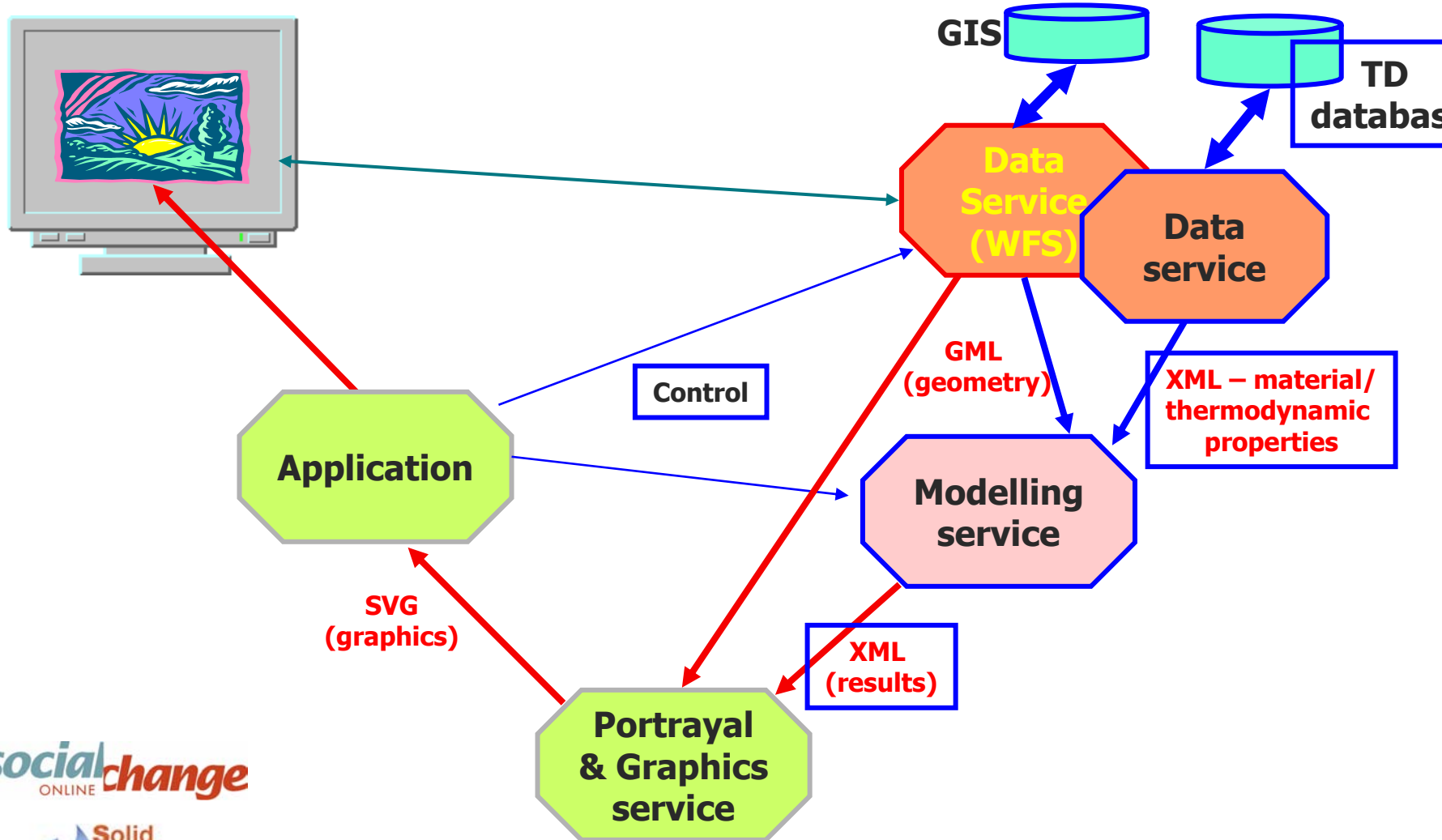
**social**change  
ONLINE



# Virtual Data Grid



# Example





# Going forward

---

Model desired outputs

(are these already well known?)

Know current storage and processing baseline

identify possible GRID applications

System architecture: what goes where

Data Models

Web Services

Registries

Build something and learn!

Deployed services are the performance benchmark

**social**  
ONLINE **change**



# Infrastructure

---

- Grants: will allow us to work out WHAT to do
- Services require a long term home...
- Services drive other people's business plans!
- So we need to address both in parallel