

# Scenarios for the SIS

Mark van Harmelen

[mark@cs.man.ac.uk](mailto:mark@cs.man.ac.uk)

Independent Consultant

and

School of Computer Science, University of Manchester

# Scope

- We are looking at end user ‘applications’
- That are feasible to build using SIS and its APIs
  - By a University
  - With help from yourselves, if you have time
  - Using non-SIS services as may be needed

# Example: XCRI Use

Not SIS, but an example of how a service can be used in a way that will be popular with many users



# Steps Today

- Write candidate applications on yellow stickies
  - Brainstorm, write till ideas exhausted, consider candidates you know or think are feasible
  - Rapid work; no deeply considered judgement
  - Sticky content “User uses x to find/modify/... y” + services used
- Filter in group
  - Check services and APIs needed
  - Retain feasible, discard others
  - Rank 0,1,2, 3, 4 potential user popularity (0 no users, 4 many users)
  - Seek expert advice elsewhere in room on unknown feasibility when ranking done
- Refine results (later slides)

# Refining results (the simple refinement case)

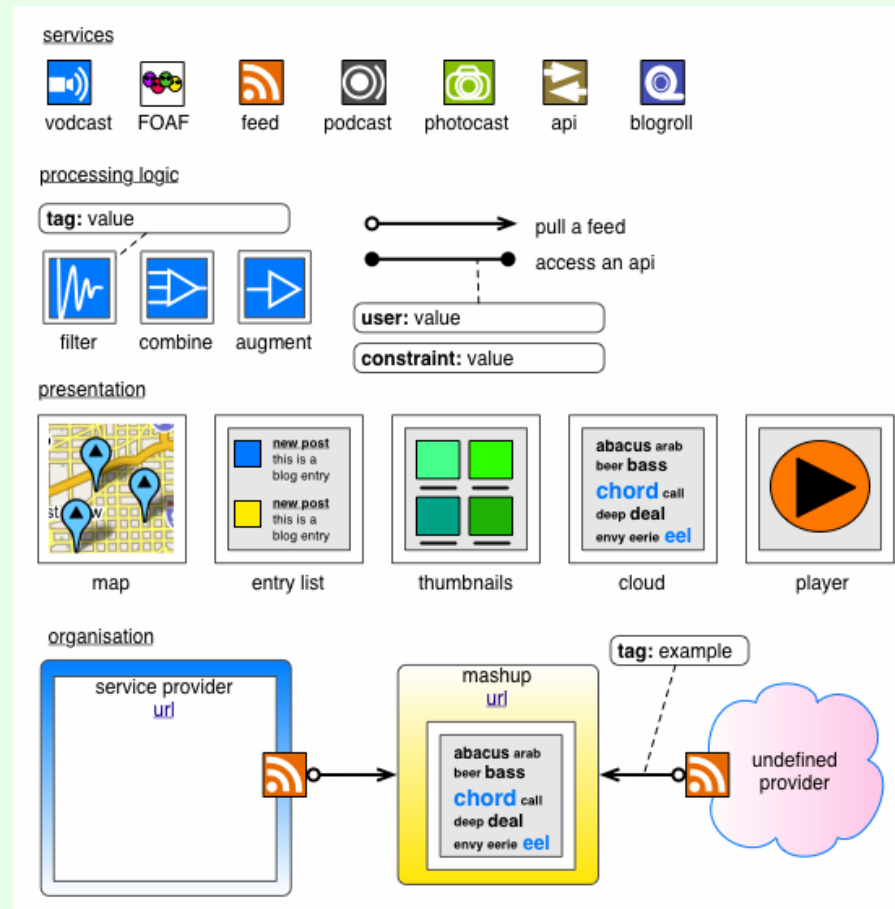
- Write a narrative story
- With an up-front a user description
  
- Eg as sent around by Tom in email

# Refining results (preferable)

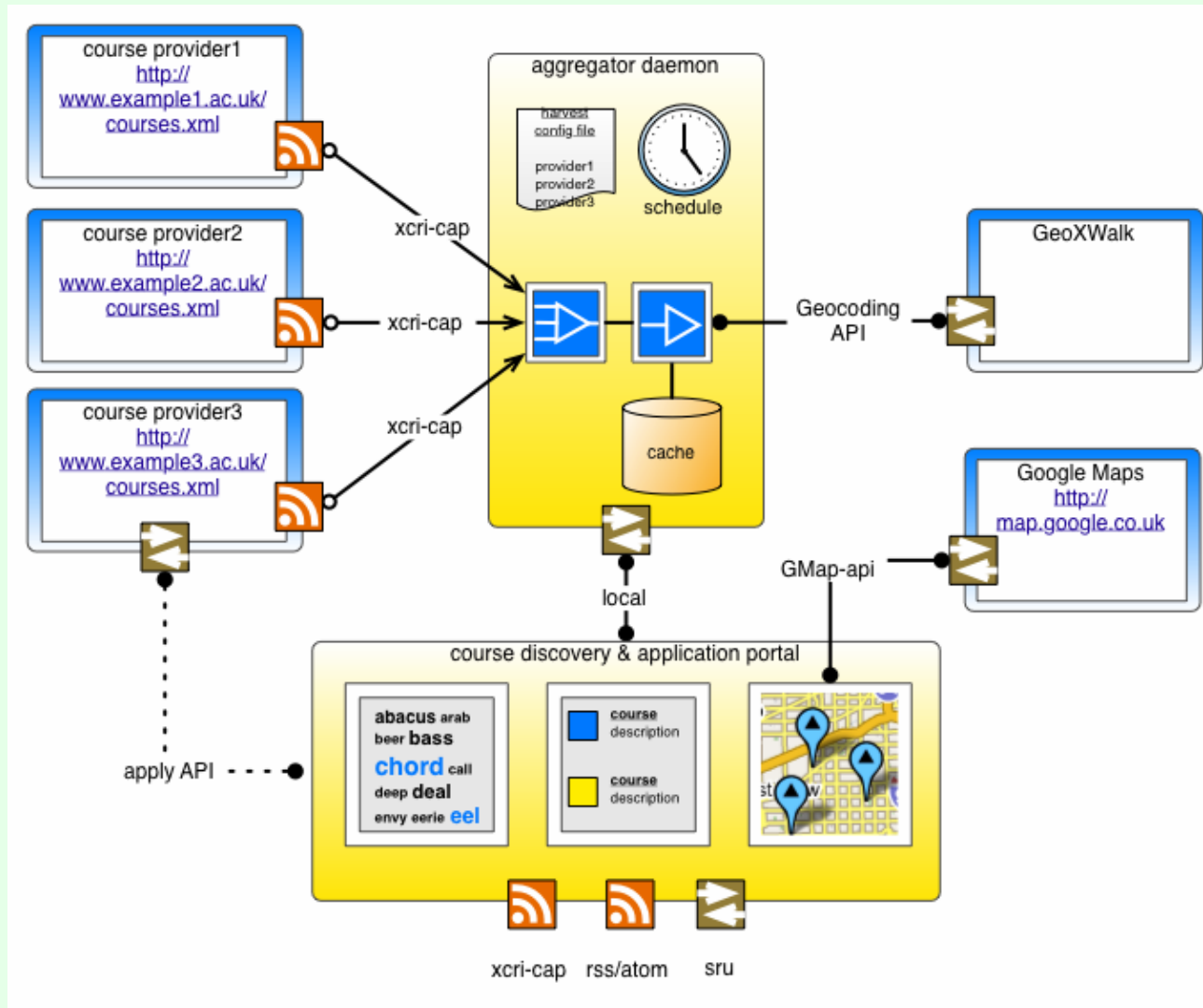
- Specify actor(s)
- Specify aim, input needed and results out
- Sketch user interface
- Draw system diagram (example on a later slide)
- Sequence diagram with protocol and API information
- Specify any minor work needed to make feasible to implement, and availability of service implementation effort (preferably avoidable)
- Please, if you do this, get to the end 😊 !!!

# Components in a system diagram

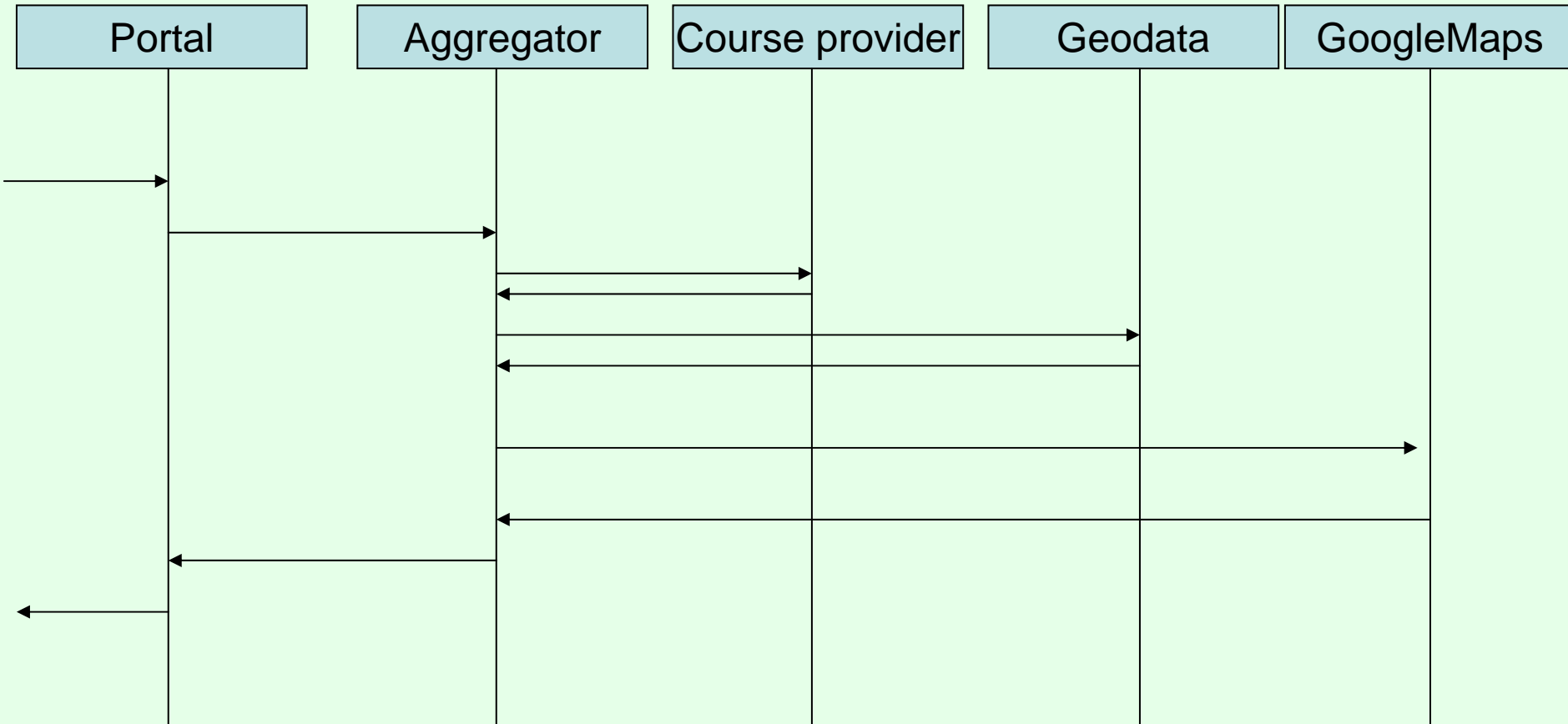
UML, or something roughly equivalent but more visually interesting:



# System diagram for the XCRI example



# Sequence diagram example



Label with 'message' names, protocols and APIs used

# Internally

- Aim; show how to do the techie stuff
- A rough guide to feasibility, should show what interfaces exist
- And for those that don't a guide to difficulty of provision, technically and wrt available effort (e.g. easy for someone who knows x internals and REST, ten days work, no available effort)