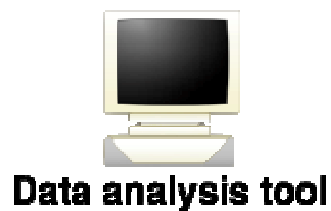
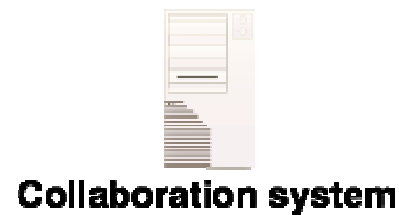


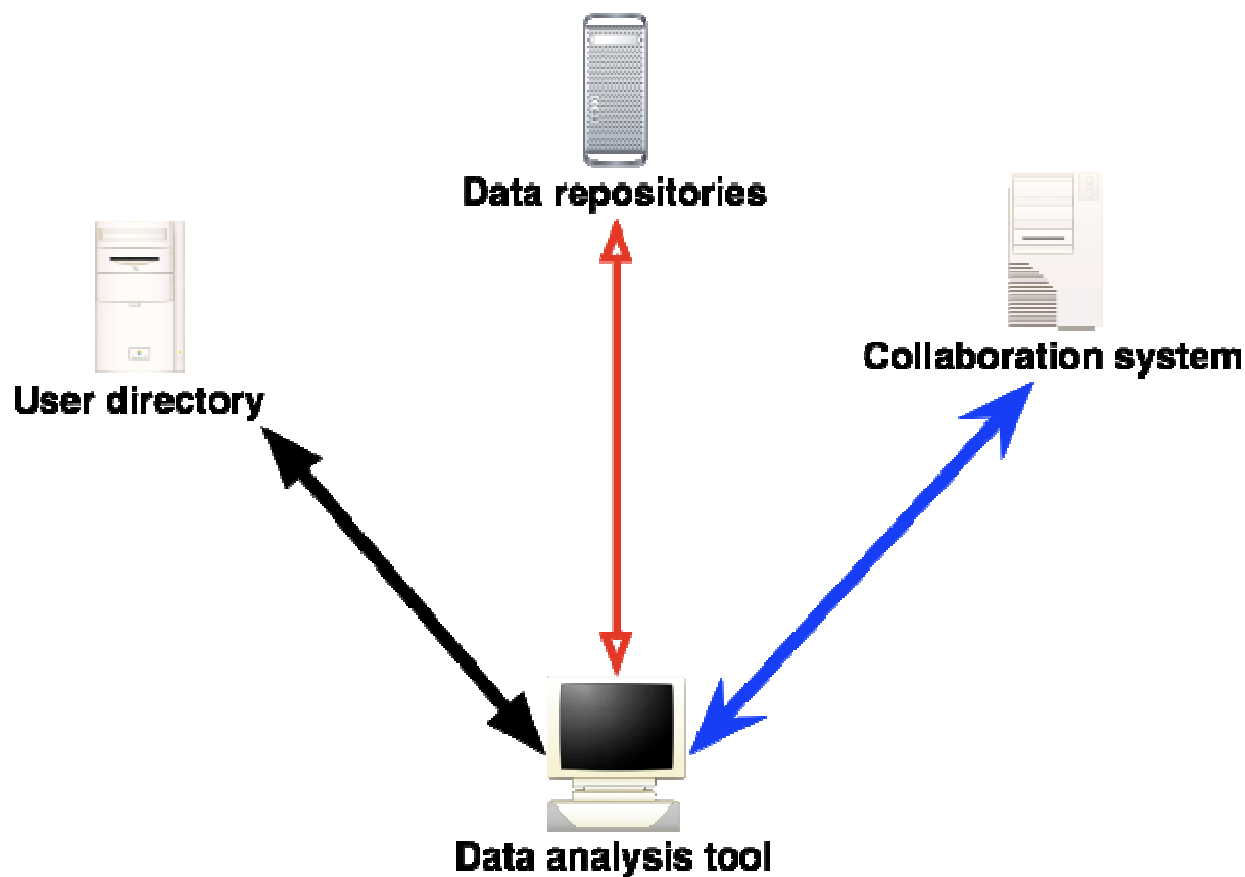
- Advanced general strand:
  - David Chadwick
  - Joan Wright
  - George Inman
  - Alice
  - Mike Mineter
  - Michael Fraser
  - Graham Mason
- Submitting strand:
  - James Farnhill
  - Ann Borda
  - Matthew Mascord
  - David Fergusson
  - Alex Voss
  - Ross Gardler
  - Neil Geddes
  - Caleb Racey

- Purpose: Closer look at the e-framework model and its relation to some use cases
- We'll look at:
  - The motivation behind the e-framework
  - Benefits for projects
  - Use cases

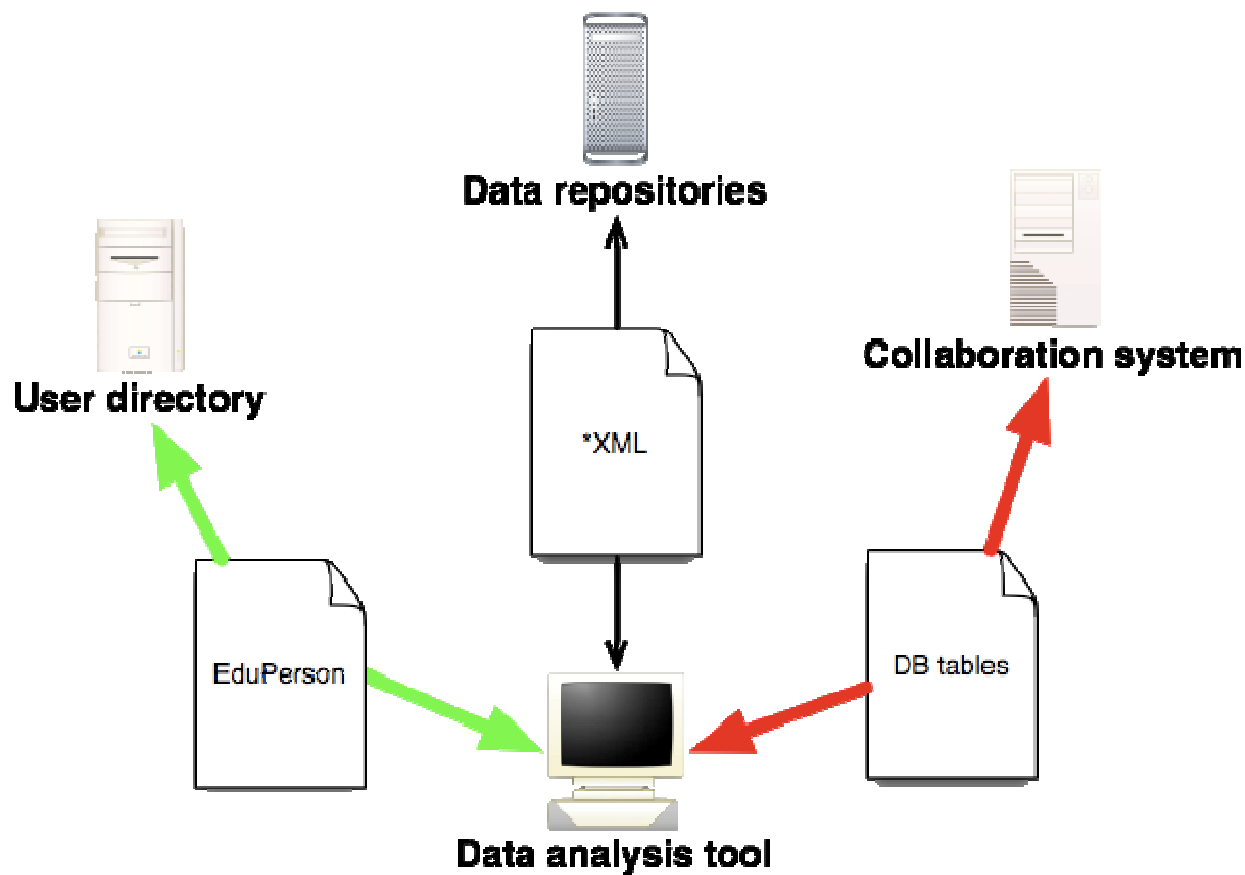
- Stand-alone systems



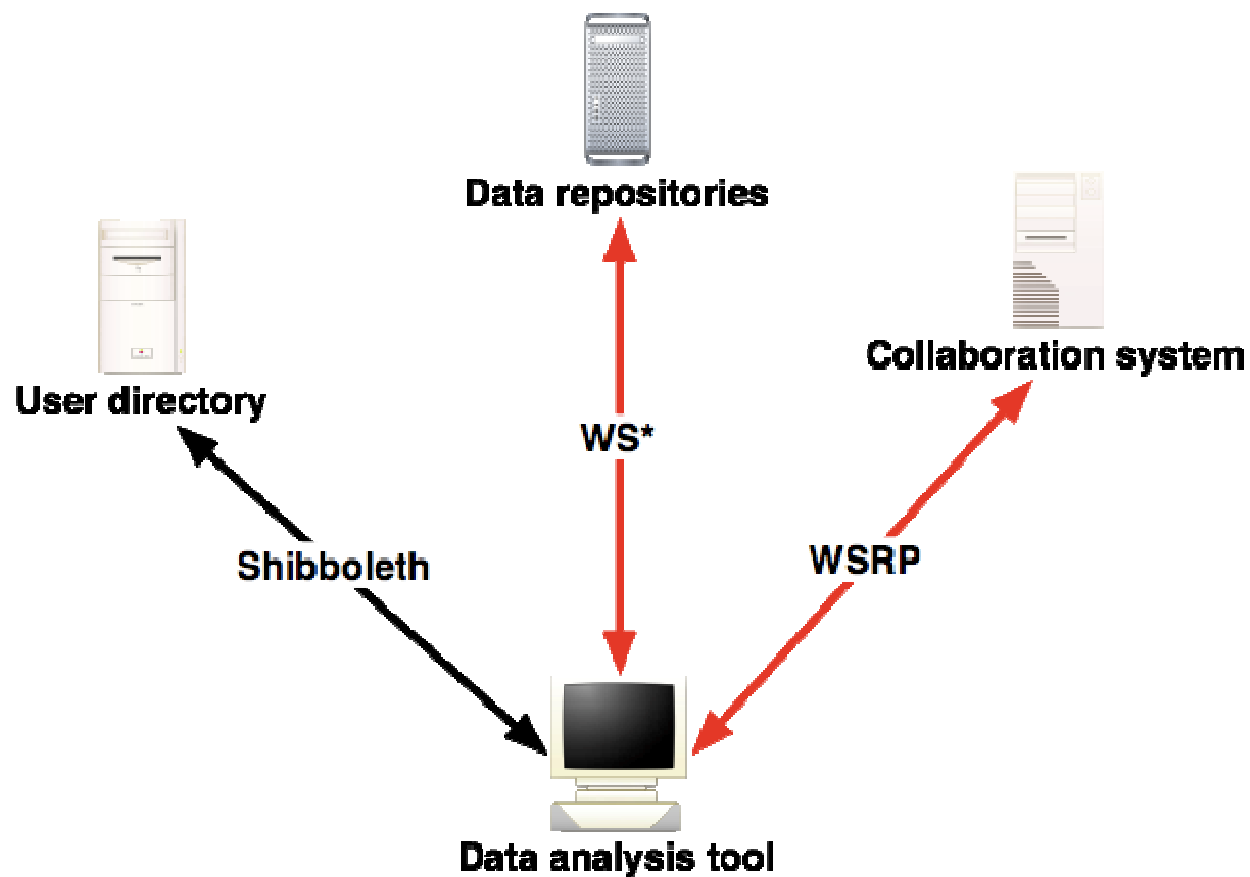
- Custom integration



- Data structure interoperability

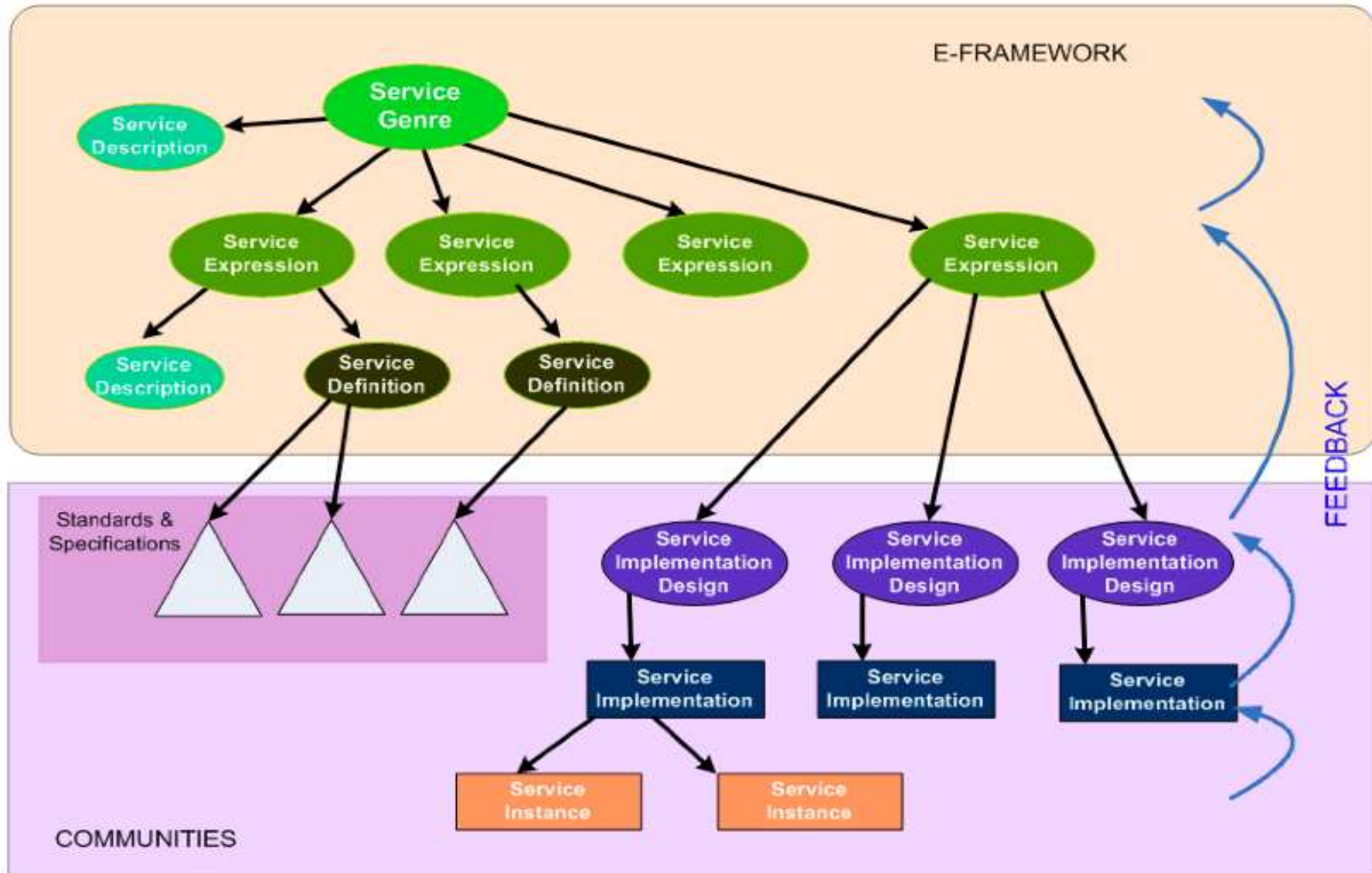


- Webservices



- Combinatorial explosion:
  - many functions/genres
  - at many different levels of granularity
  - with many implementations for each function or genre
  - and little or no indication how to put them together

# Services



- A recognisable type or category of service
  - e.g. *Search, Alert user* etc.
  - Groups together similar specific interfaces or Service Expressions
  - Captures as much as possible of the abstract commonality within the genre

- A specific service interface contract
  - e.g. Z39.50, SRW/U, IETF Atom, SMTP
  - Fulfils part or whole of the functional scope of the Service Genre it belongs to
  - Consists of
    - Service definition (machine readable)
    - Service description (human readable)
  - Can be monolithic, or a composition of different agreements about:
    - Data model
    - Behaviour
    - Binding

- Blueprint for a piece of software that implements a service
  - Platform specific
  - Complete; takes in more requirements than just the interoperability contract
  - Outside the e-Framework proper

- Portable piece of software that implements a service
  - e.g. Dspace, Movable Type binaries

## Service instance

- Actual deployed service endpoint on the network

- Collection of services, combined to support:
  - One or more workflows or processes
  - Using a description of the policies, resources, services and ways to combine them
  - Are a concrete, standardised description of a domain specific model

# Benefits - Institutions



- Alignment of strategies and infrastructure development
- More choice of systems and suppliers
- Improved return on investment in existing systems
- More effective communications between communities through shared understanding
- Interoperability within and across institutions and national boundaries

## Benefits - Developers

- Better understanding between suppliers and customers
- More rapid development cycles through reusable components
- Entry of small innovative players into the market
- Faster response to customer requirements
- Communication and collaboration between developers
- Flexible business models for software development

# Use case: identity management



- A simple SUM: New Zealand's Identity and Access Management ([IAM](#))
- A more extensive SUM: Australia's Meta Access Management System ([MAMS](#))

# A use example



- Simple use case:
  - Several systems need to notify users
  - Users have different ways in which they can or need to be notified
- E-Framework has service genre:
  - 'Alert Users'
    - “Alerting services provide a mechanism by which information relating to an event can be sent to a human for attention”
    - Handles various message formats

# A use example, ctd.

- 'Alert users' genre has one service expression:
  - OASIS Common Alerting Protocol
    - No particular binding
    - For emergency services
    - But: some useful semantics

In short: roll your own, with some guidance...

# A use example, ctd.

- Your 'Alert user' service expression can make use of:
  - Service genre: 'Message user'
    - Service expression: XMPP
      - Service Implementation: Jabber
      - Service Instance: Google Talk server
  - 'E-mail transport'
    - SMTP
      - Sendmail
      - smtp.nottingham.ac.uk:25