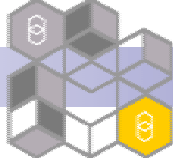


# Service matching in bioinformatics – an introduction to “shims”

Duncan Hull

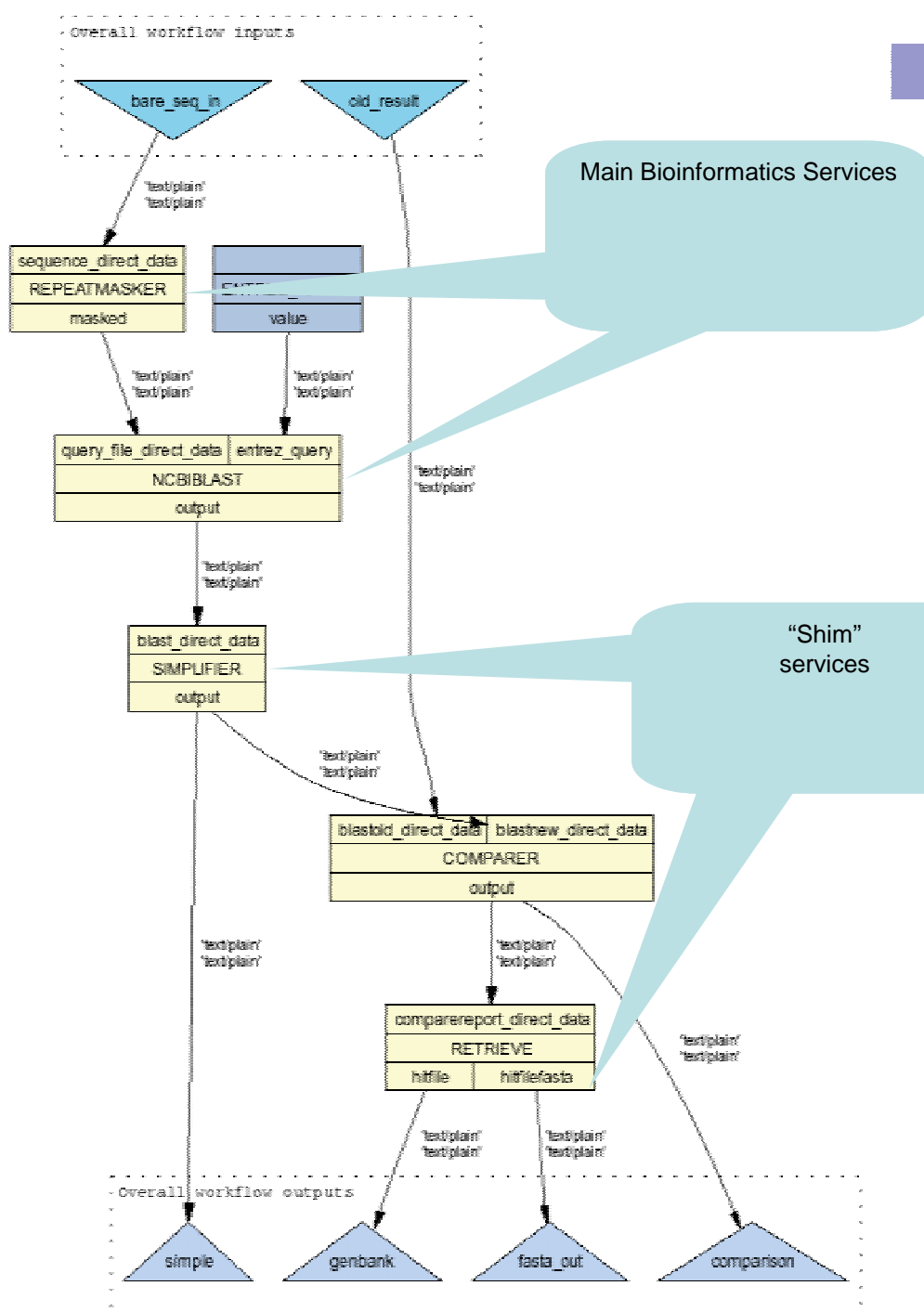
University of Manchester  
17<sup>th</sup> October 2004: SDSC Linkup

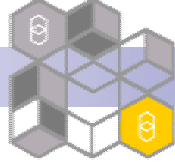


# Introduction

myGrid: Services for biologists to create workflows perform experiments

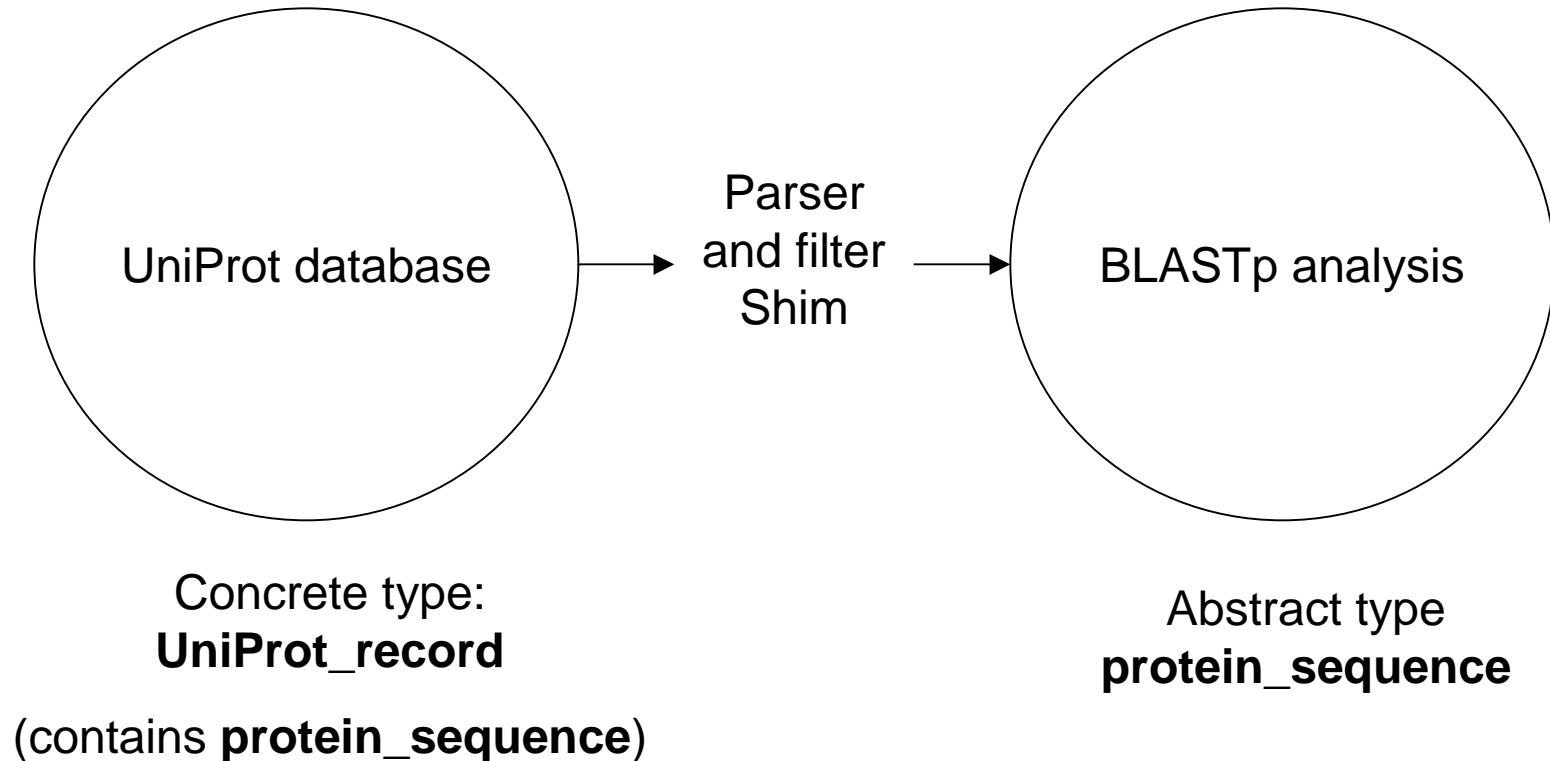
- Matching I/O
- Done by text – no type
- **16 bio, 10 shims**
- **Ubiquitous problem:**

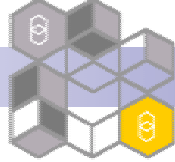




# Shims in detail

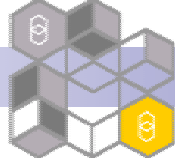
- Semantically compatible, syntactically incompatible





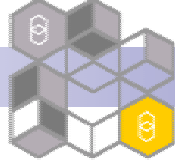
# Shim (working) definition

**Shim:** A shim is a software component who's main purpose is to syntactically match otherwise incompatible resources. It takes some input, performs some task and produces an output. *Depending on usage*, a shim is **experimentally neutral** and possibly automatable.



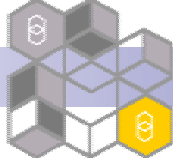
# Type manipulations

	<b>Abstract</b>	<b>Concrete</b>
<b>Type</b>	DNA_sequence	EMBL, GenBank, FASTA
<b>Instance</b>	“My favourite DNA sequence”	1    ttcctctttc... 61   agctctttgt... 121 cccagatcaa...

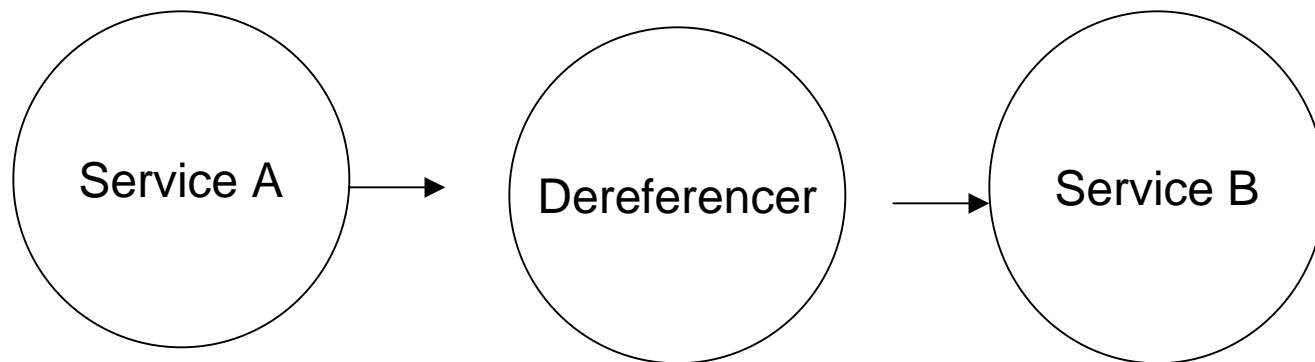


# Some examples

- Parser / Filter
- De-referencer
- Syntax translator
- Mapper
- Iterator

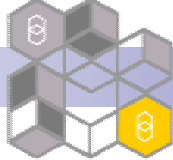


# Dereferencer

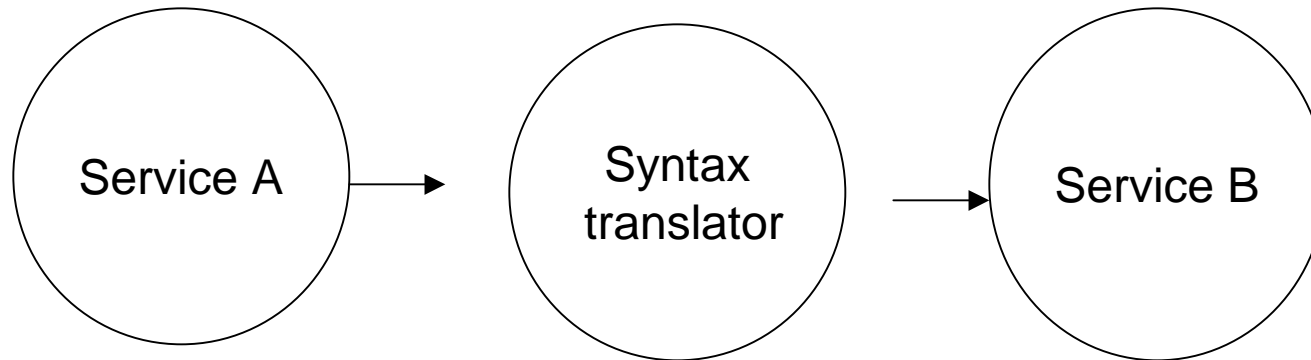


Concrete type: **GenBank identifier**

Concrete type: **GenBank record**



# Syntax translator

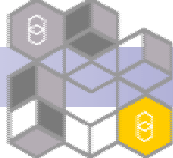


Abstract type: **DNA sequence**

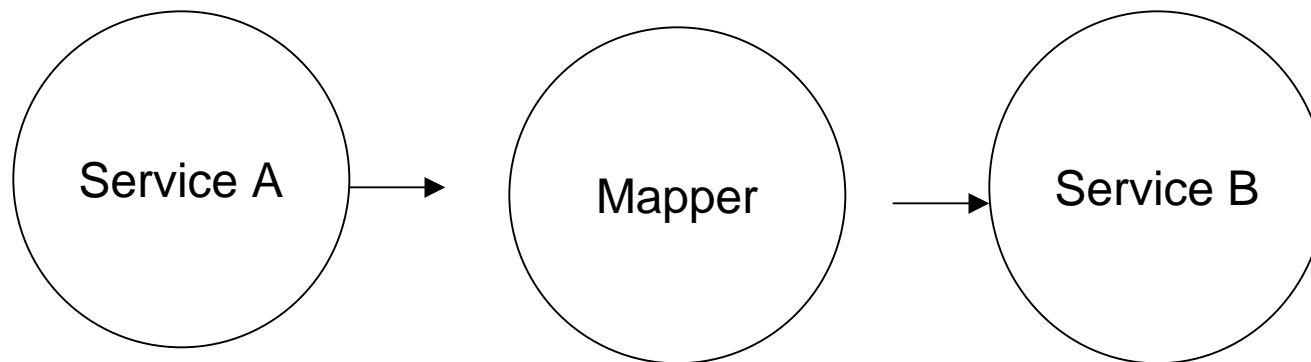
Concrete type: **BSML**

Abstract type: **DNA sequence**

Concrete type: **AGAVE**

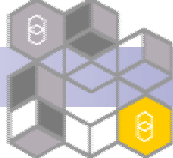


# Mapper

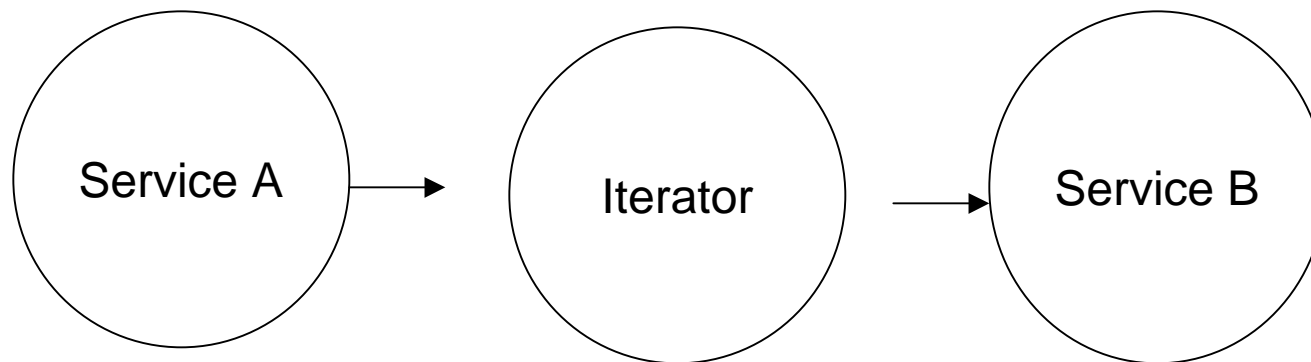


Concrete type: **GenBank identifier**

Concrete type: **EMBL identifier**

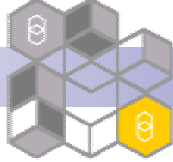


# Iterator



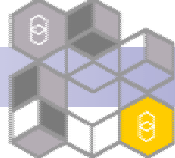
Output type: **Collection of x**

Input type: **a Single x**



# Seven steps to shim nirvana

1. Recognise two services are not compatible
  - Syntactic and semantic levels
2. Recognise the degree of mismatch
  - everything connected to everything
3. Identify what type of shim(s) is/are needed
4. Find or manufacture the shim
5. Advise user on “semantic safety” of the shim
6. Invoke the shim
7. Record provenance



# Conclusions

- Shims are ubiquitous
  - No globally accepted type system to describe all bioinformatics data that services produce and consume
  - (probably never will be!)
  - Not just a problem in bioinformatics...
- Ontologies of web services help matching semantically and can ease syntactically, maybe bridge between the two