(What I have learnt)

Building stateful web services

Matthew J. Graham
CACR, Caltech

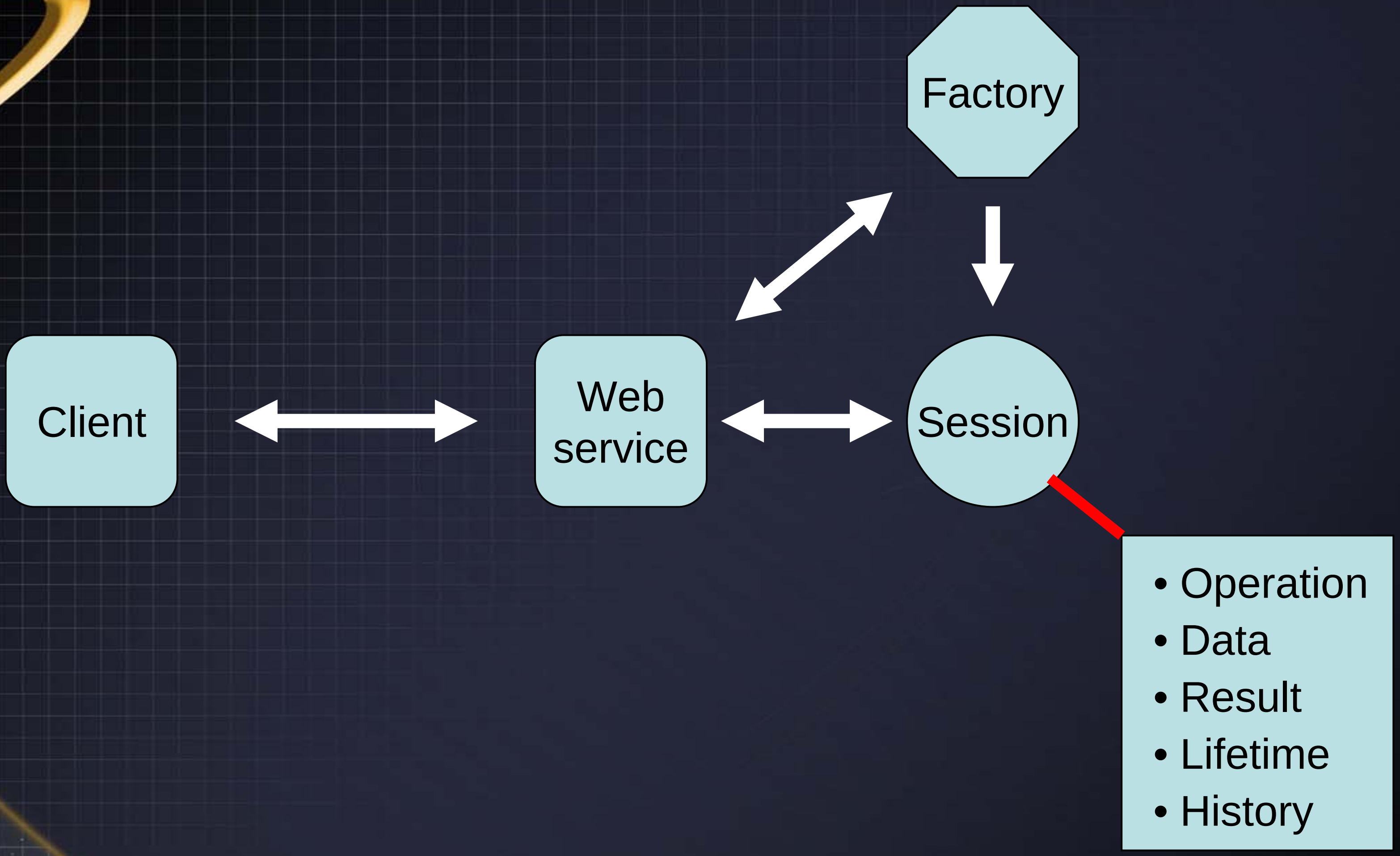
Concept: VOStatistics

- “a prototype knowledge-based statistical toolkit implemented within the VO paradigm for the entire astronomical community.”
- easily extensible (legacy code)
- distributed
- open source client GUI (interactive)
- http://www.vostat.org

Considering state

- What is state?
- Do we really need it?
- Isn’t it just about server-side code?
- How hard is it to do?

Basic model


Factory
Client
Web service
Session
• Operation
• Data
• Result
• Lifetime
• History

Toolkits

- Java (http://www.javaskyline.com/webservices):
  - Apache and Axis (http://www.apache.org)
  - Globus (http://www.globus.org)
- C#:
  - .Net; Mono (http://www.go-mono.com)
  - OGSI.NET / WSRF.NET (http://www.cs.virginia.edu/~gsw2c/wsrf.net.html)
- Perl:
  - SOAP::Lite (http://www.soaplite.com)
  - OGSI::Lite / WSRF::Lite (http://www.sve.man.ac.uk/Research/AtoZ/ILCT)
- Python:
  - SOAPpy / ZSI (http://pywebsvcs.sourceforge.net)
  - PyGlobus

MMPWS

- "Stateless"
  - public double getAnswer(double x, double exp)

- "Stateful"
  - public void setValue(double x)
  - public double getAnswer(double exp)
  - public void setPower(double exp)
  - public double getAnswer()

Clients

- Java (using WSDL2Java):

```
MMPWSLocator mmpwsl = new MMPWSLocator();
mmpwsl.setValue(3.14);
System.out.println(mmpwsl.getAnswer(0.5));
```

- Perl (using SOAP::Lite):

```
my $mmpws = SOAP::Lite -> service(“http://…/blah.wsdl”);
$mmpws->setValue(3.14);
$print $mmpws->getAnswer(0.5);
```

- Python (using SOAPpy):

```
server = SOAPProxy(“http://…/blah.wsdl”)
server.setValue(3.14)
print server.getAnswer(0.5)
```

Discussion points

- Invocation
  - Syntax / Discovery
- Data
  - Virtual
- Workflow
  - Data / Dynamic management
- Asynchronous activity
  - Mechanism
- Security
  - Mechanism

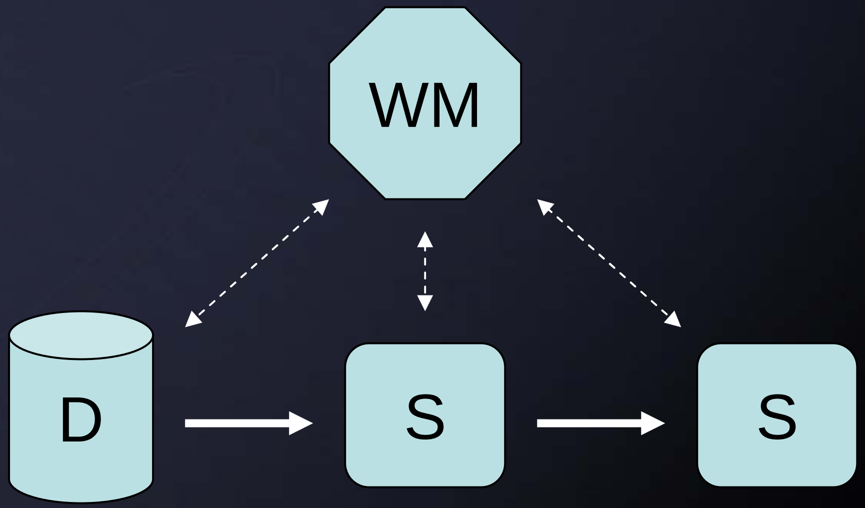
Invocation

-
```
<Invocation>
    <Method>
    <Data>
        <Location>
        <Format>
        <Subset>
            <Row> / <Column> / <Plane>
        <VOTable> / <FITS>
    <Argument>
        <Key> … <Value>
    <Qualifier>
```

- getInvocationDescription(…)

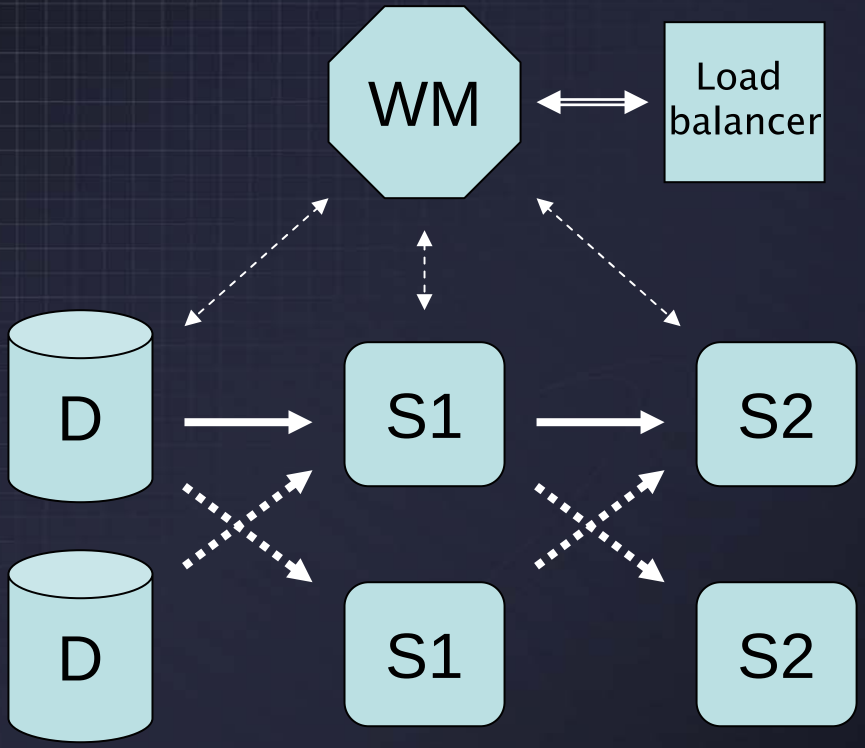
Data

- Location
- Virtual
  - Description
  - Instantiation: When? How? Managed? – “grazing”
  - Persistence
- Workflow
  - Data client location:



Workflow

- Dynamic management


WM
Load balancer
D
S1
S2
D
S1
S2