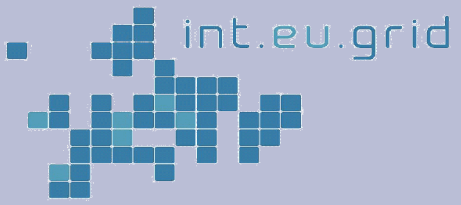
Interactivity and the Grid: the Web Service approach

Why web-services?
 - because the development of middleware points into this direction.

See:
  The Globus Toolkit:
  GT2 – pre WS Gram. With GT3 / GT3: WS GRAM

See also:
  Unicore:
  Unicore 5: no Web Services. Unicore 6: Web Services.

What about glite?
  Not yet. But since glite is based on GT2, who knows in which direction glite will evolve. So better be prepared.



Method 1: using JobFactory

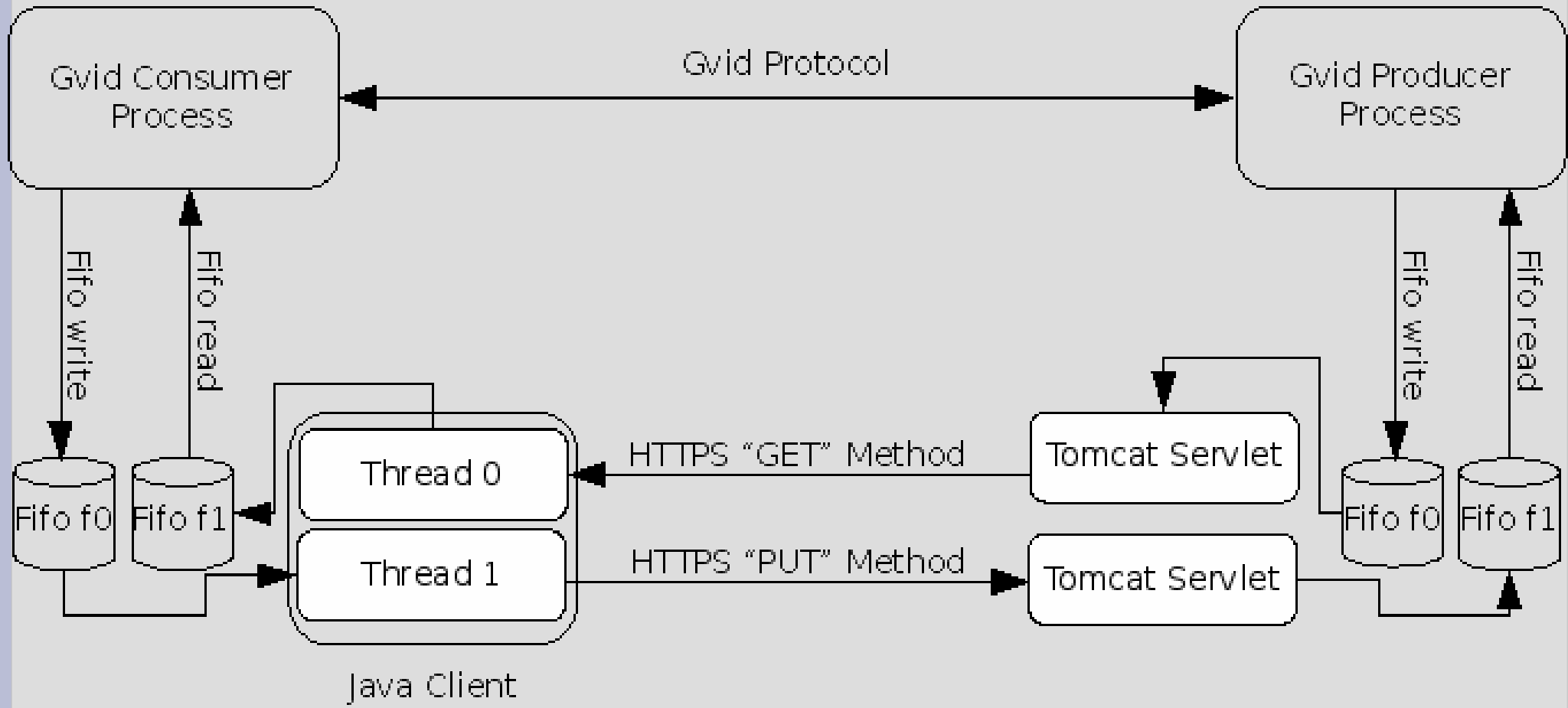
In GT4, WS based job submission is already implemented. Is being used by globusrun-ws: ManagedJobFactoryService, ManagedJobService, ManagedExecutableJobService, DelegationService, SubscriptionManagerService.

Glogin-ws: a Web Service based method to establish a seperate interactive communication channel, same as in pre-WS glogin. Glogin-ws makes use of the same services as globusrun-ws.

Advantage: out of the shelf Web Services can be used.
Disadvanted: slower (up to 10 seconds due to gsipftp polling), possible firewall issues.

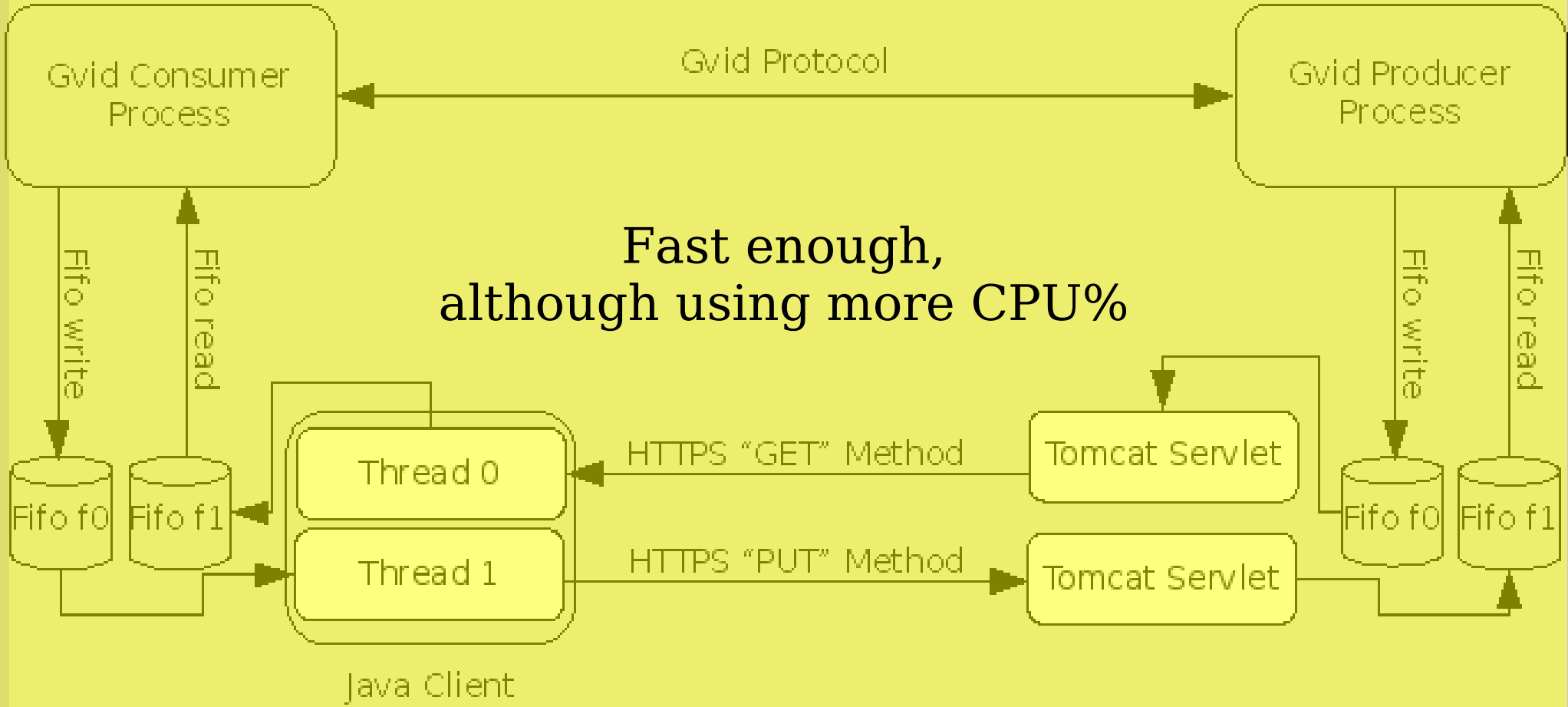
Method 2: a dedicated Web Service for Interactivity.

Preliminary Feasibilty experiment: Interactive connection with Java (Tomcat as Web Service container) using HTTPS for transportation using “Gvid-Doom”.



Method 2: a dedicated Web Service for Interactivity.

Preliminary Feasibilty experiment: Interactive connection with Java (Tomcat as Web Service container) using HTTPS for transportation using “Gvid-Doom”.



Method 2: a dedicated Web Service for Interactivity.

Web Services are different!

Additional layer (“marshalling”): encapsulation in SOAP/XML

each web-service invocation is:

connect()
SSL-Handshake
send SOAP/XML request
receive SOAP/XML reply
disconnect

=> extremely! Slow.

Method 2: a dedicated Web Service for Interactivity.

If each WS-invocation creates a new connection: can we somehow(??) re-use an existing WS-connection?

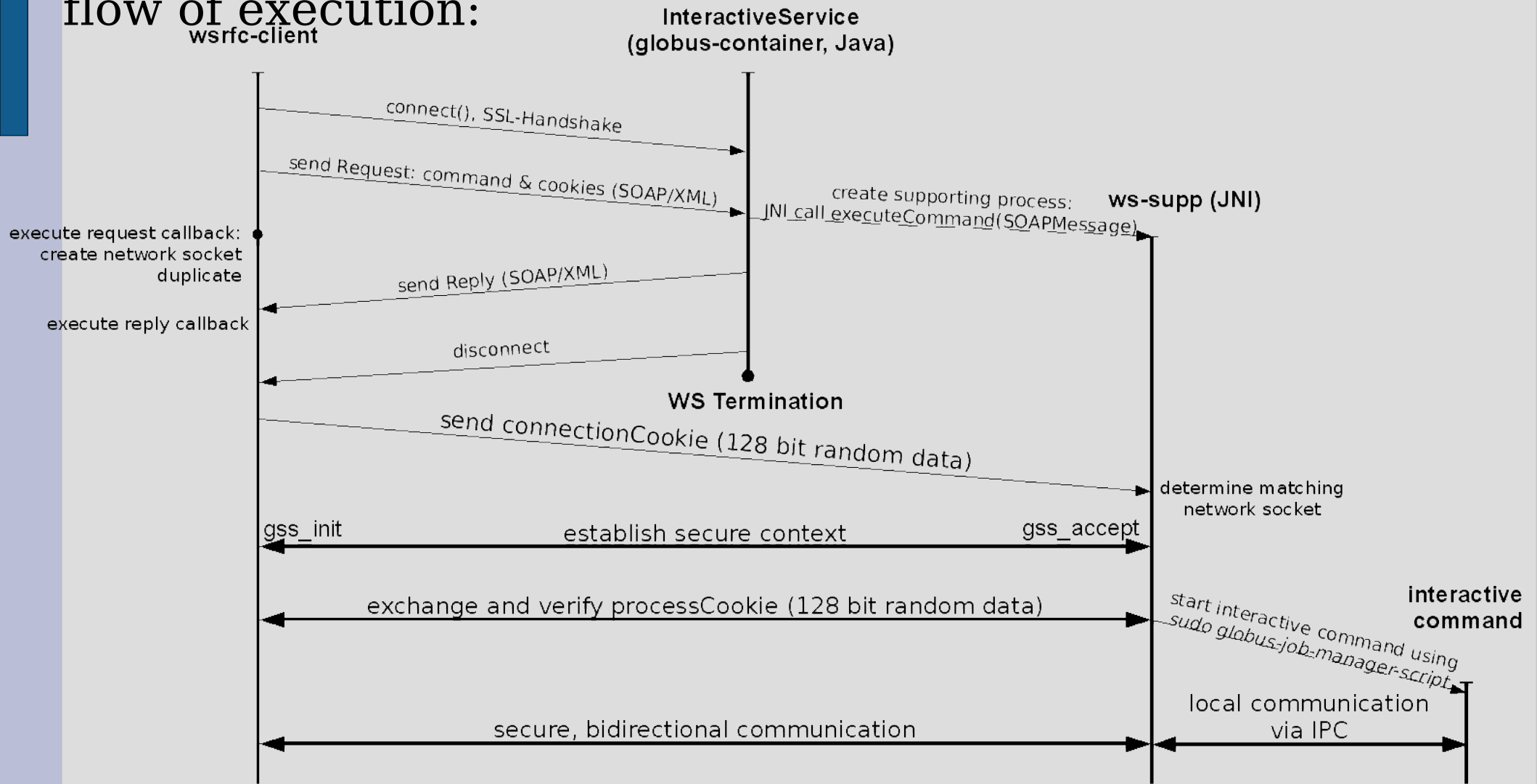
Answer: YES!

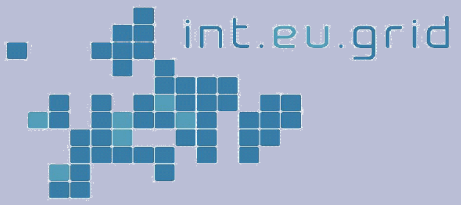
How is this done:
(1) client: asynchronous WS-mechanism with callbacks for request/reply completion: create a copy of the outgoing network descriptor (unix, posix: dup(2))

(2) Web-Service: use a supporiting method to search for the network-descriptor a SOAP/XML message was received on. Unfortunately, the java.net.Socket object is not passed to the WS-method, so it's necessary to examine all connections in order to find out the right one.

Method 2: a dedicated Web Service for Interactivity.

To do this, the ws-client and the WS use cookies. Complete flow of execution:





Conclusion and Future Work

- Conclusion: pretty fast mechanism. Demonstration? Although not pure Java – it's not possible!
- Future Work: Not competing with glogin, but rather: use this Interactive Service as another alternative for glogin's job submission mechansim. (glogin has better optimisation for data transmission)

That's it. Thanks for listening.