Platform for Astronomy Tool InterCommunication

PLASTIC is...

- A specification for how astronomy tools can interoperate on a user's desktop
- Simple
- Extensible
- A collaboration between the Aladin, AstroGrid, Topcat and VisIVO teams

PLASTIC

Platform for Astronomy Tool InterCommunication

- The culpable:
  - Thomas Boch (Aladin)
  - Marco Comparato (VisIVO)
  - Sebastien Derriere
  - Pierre Fernique (Aladin)
  - Bob Mann (IfA)
  - John Taylor (AstroGrid/IfA)
  - Mark Taylor (Topcat)
  - Noel Winstanley (AstroGrid)

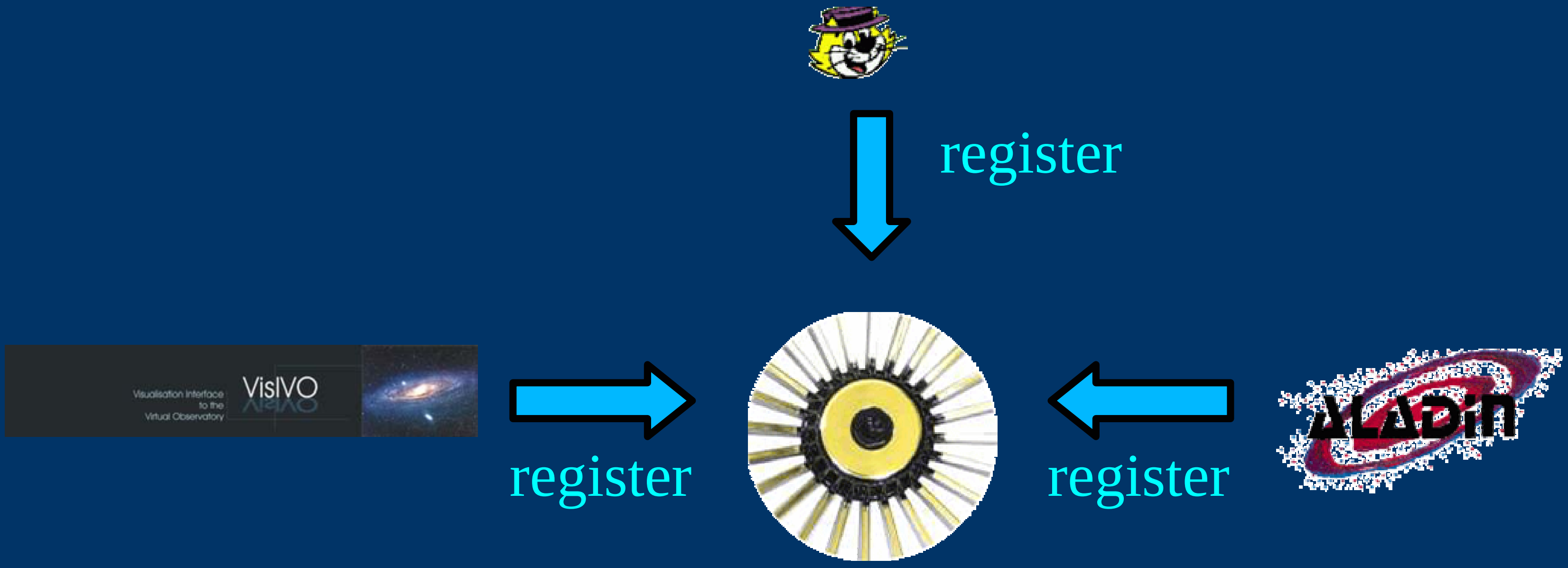
Contents

- PLASTIC – why and how?
- Demo
- Plasticizing your apps
- Where are we now?
- What next?

Some history...

- VisIVO & Aladin interoperability (Becciani, Comparato, Gheller...)
- The Aladin plugin interface (Boch, Fernique...)
- The Astro Client Runtime (Winstanley)

Architecture: publish-subscribe



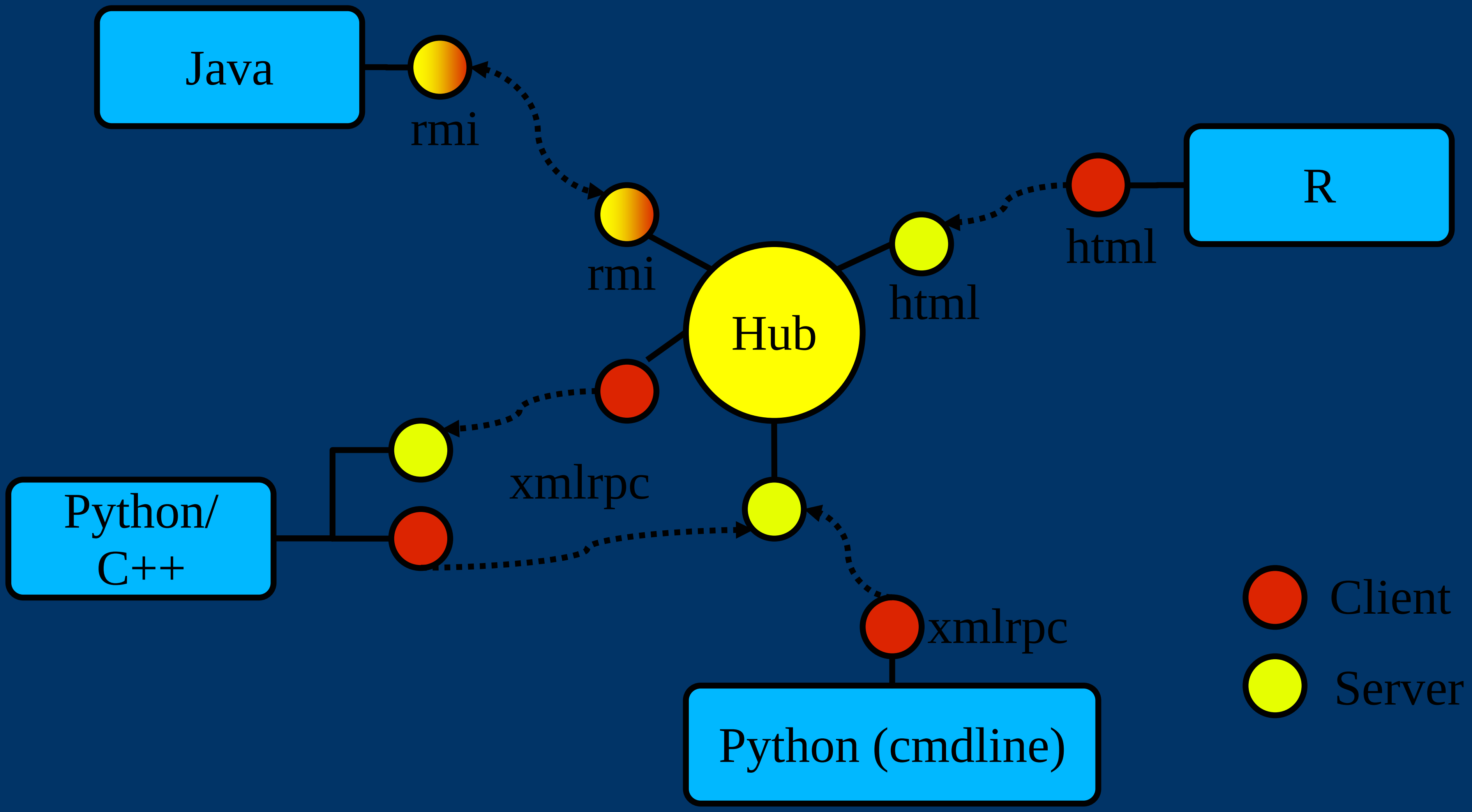
register

register

register

“Plastic Hub”

Architecture: multiple “protocols”



Messages

- ivo://votech.org/votable/load
- ivo://votech.org/votable/loadFromURL
- ivo://votech.org/votable/showObjects

- ivo://votech.org/test/echo
- ivo://votech.org/info/getName

Demo

Plasticizing your apps: Java

(non-programmers, please talk amongst yourselves)

- Implement the PlasticListener interface:
  - `Object perform(URI sender, URI message, Object[] args)`
- Obtain a reference to the PlasticHub
- Register your application
  - `id = hub.registerRMI(“myapp”, messages, this)`
- Send messages to other apps:
  - `Map responses = hub.request(id, message, args)`

Plasticizing your apps: xml-rpc

(non-programmers, please talk amongst yourselves)

- Run an xml-rpc server in your app (with an off-the-shelf-library)
- Obtain the URL of the hub's xml-rpc server
- Register your application
  - `id = s.plastic.hub.registerXMLRPC(“myapp”,[], myURL)`
- Send messages to other apps:
  - `responses = s.plastic.hub.request(id, message, args)`

Plasticizing your apps: scripting

(non-programmers, please talk amongst yourselves)

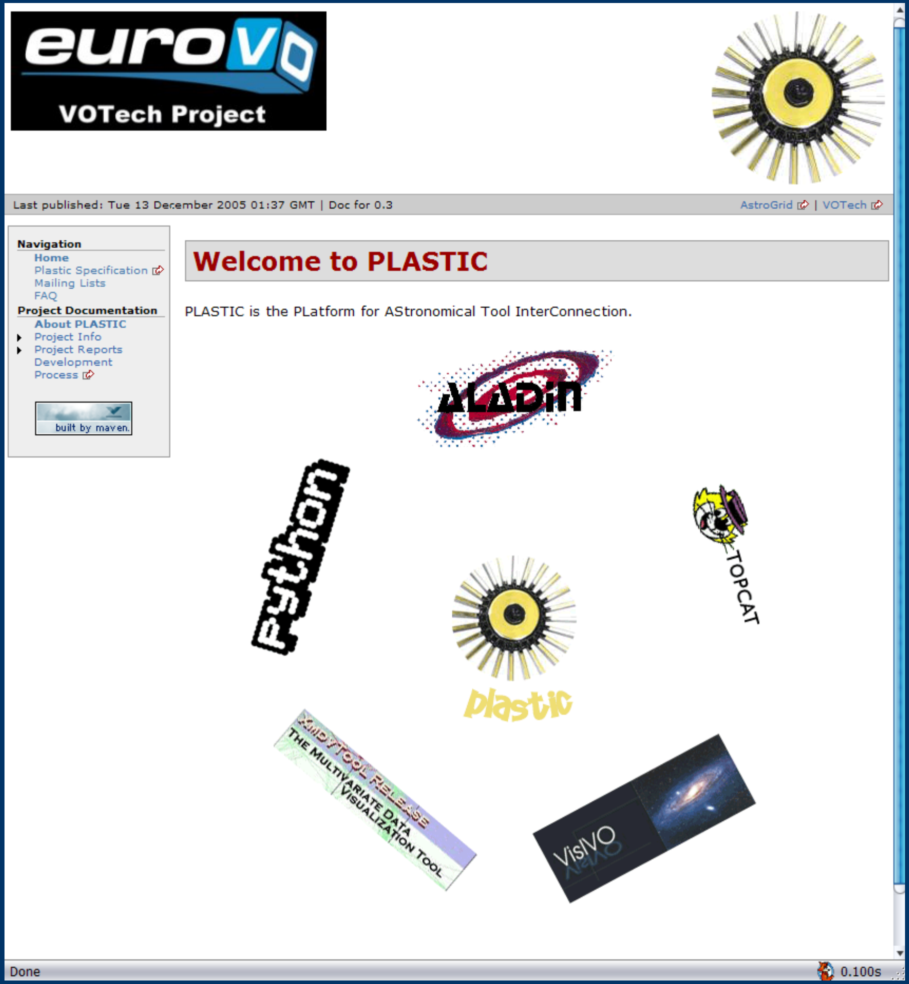
- Obtain the URL of the hub's xml-rpc server
- Register your application
  - `id = s.plastic.hub.registerNoCallback(“myapp”,[], myURL)`
- Send messages to other apps:
  - `responses = s.plastic.hub.request(id, message, args)`

Where are we now?

- Version “1.0RC1” of the spec agreed
- An implementation of a Plastic Hub in the ACR.
- Plastic-compatible tools:
  - Xmdv-lite
  - Aladin
  - TabView
- Coming soon:
  - Topcat
  - VisIVO

Where are we now II?

- Bindings available for R, Python
- Coming soon: IDL?
- Java library for accessing plastic hub
- Website:
  - http://plastic.sourceforge.net
- Mailing lists:
  - http://plastic.sourceforge.net/mail-lists.html



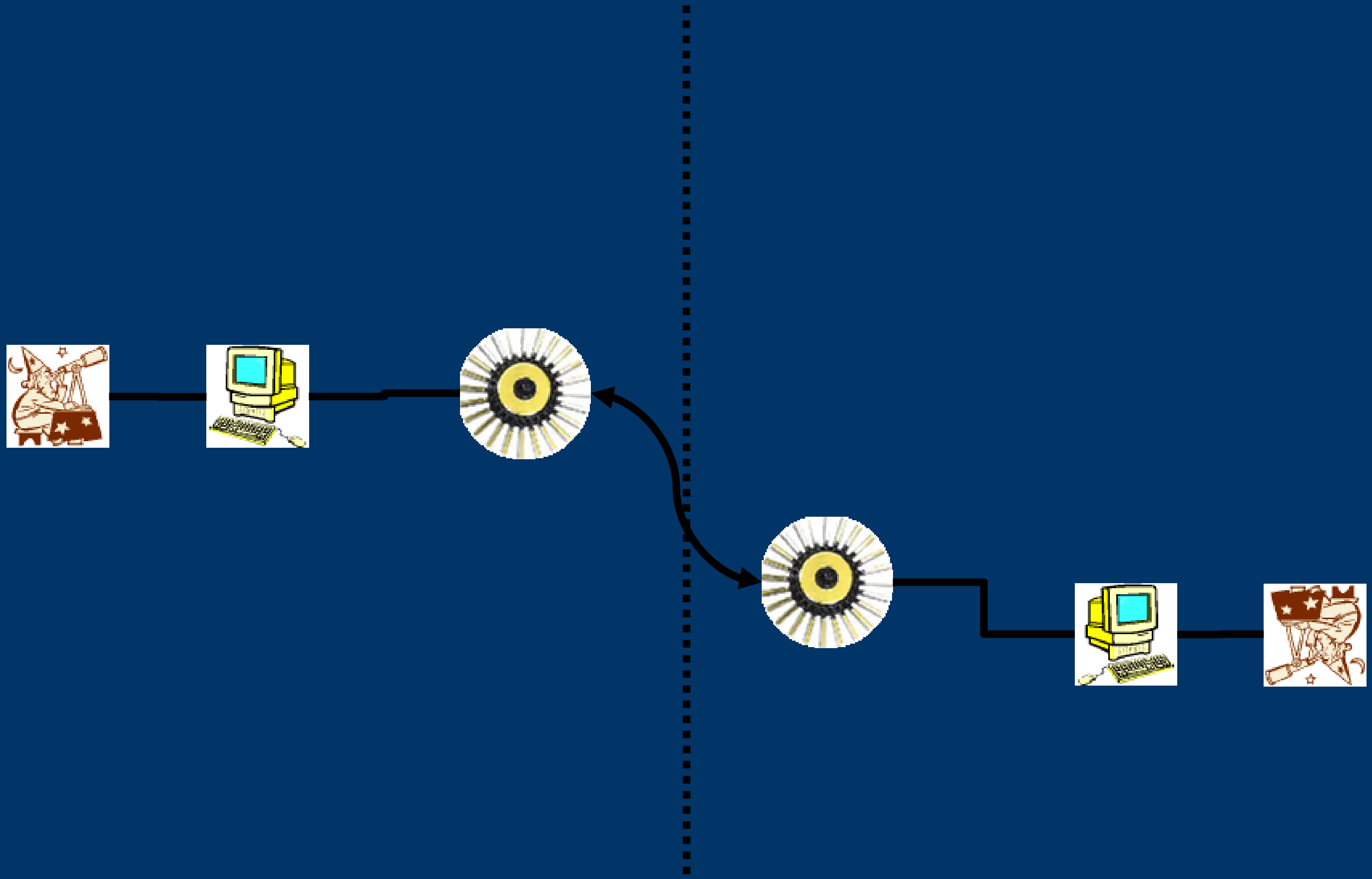
Last published: Tue 13 December 2005 01:37 GMT | Doc for 0.3

Welcome to PLASTIC

PLASTIC is the PLatform for AStronomical Tool InterConnection.

Other stuff we could do...

Collaborative working



Other stuff we could do....

Server side data processing

(white board)

What next?

- Are we on the right lines?
- What other tools should be plasticized?
- What messages would be useful?
- What other applications are there of this technology?