Client side applications and scripts for Astrogrid

Noel Winstanley - nw@jb.man.ac.uk

Aims

- A simple, uniform way to access and integrate VO components
  - remote
    - web services (registry queries, CEA, Myspace, JES)
  - client-side
    - logic (Astrogrid A&A, Tool)
    - gui components (Myspace microbrowser)
  - later
    - utility libraries (STIL)
    - visualizers (VOSpec, Topcat, etc)
- from any programming, scripting or shell language
- on any platform

Noel Winstanley - nw@jb.man.ac.uk

Move VO out of the Browser

- query the registry from the unix commandline
- bulk myspace operations in batch files
- load and save into myspace from emacs
- call CEA applications from IDL scripts
- glue together VO GUI components in Perl
- prototype workflows in Python
- knock together a workflow builder in Tcl/Tk
- even Visual Basic or Exel macros...
- Integrate VO components with the tools that astronomers use at the moment

THE UNIVERSITY of MANCHESTER

Noel Winstanley - nw@jb.man.ac.uk

AstroGrid

Restrictions & Assumptions

- Platform Independence
  - can't use COM, CORBA, DCOP
- Language Independence
  - can't use Java-RMI
- Simple Deployment
- Assume all VO components are written or accessible from Java
  - Java client-side libraries and delegate.

Option 1: Go to the SOAP

- *Write code that calls the VO webservices directly.*
- SOAP is language and platform neutral
- Interoperability is tricky.
- Not all languages provide a SOAP toolkit
- WSDL doesn't define whole contract
  - hidden SOAP headers - A&A
  - protocol
    - order in which methods in same interface are called
    - order on methods on different interfaces
- doesn't expose client-side libraries.

Option 2: Delegates for All

- *VO projects produce client-side libraries for a range of popular languages*
  - e.g. port the astrogrid delegates and client libraries.
- a chore
- testing nightmare
- ports will diverge in behaviour.
- always going to leave someone's favourite language out.

THE UNIVERSITY of MANCHESTER

Noel Winstanley - nw@jb.man.ac.uk

Astro Grid

Option 3: Linkable Library

- *Find ways of linking the existing Java client-side libraries into other languages*
- single implementation,
  - same behaviour everywhere.
- java-IDL bridge, java-Perl bridge.
  - single-language solutions.
- Build a shared library using GCJ / JNI?
  - platform-dependent
  - not all languages are able to load libraries.
- heavy-duty install
- Efficient (apart from being implemented in Java)

Noel Winstanley - nw@jb.man.ac.uk

Option 4: Linkable Component

- *A server that runs on the user's desktop*
  - *accepts requests from other desktop applications*
  - *processes requests using the existing Java client-side libs.*
- Advantages
  - Single sign-on, configuration, preferences
  - Common cache for sharable results (registry resolving, myspace filestructures)
  - a place to maintain state
  - can receive and display notifications from services, even after the initiating application has terminated

How do applications communicate connect to the Server?

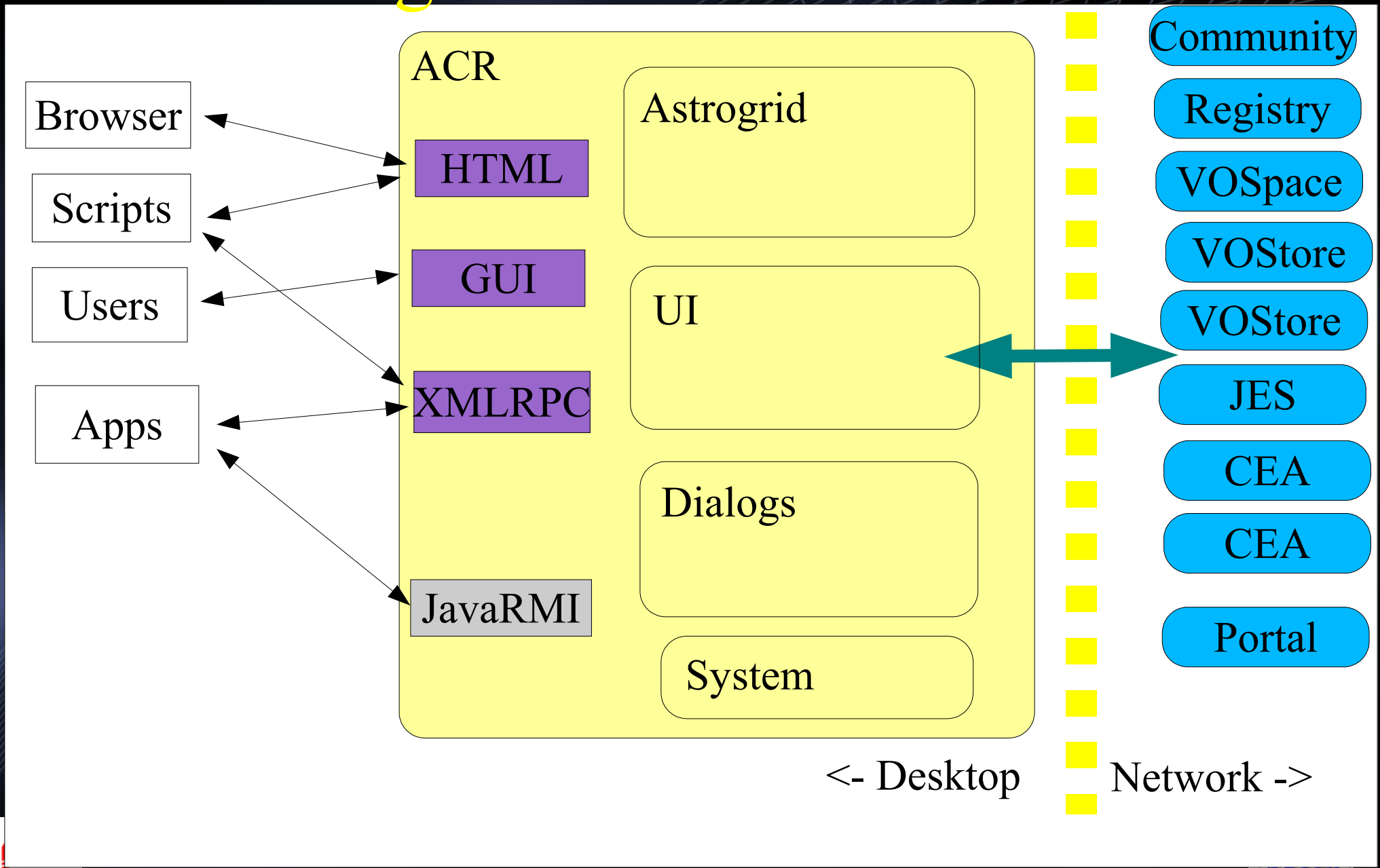
- There's next to no language & platform independent way to do this.
  - Systems for communicating between running processes tend to be tied to a specific desktop
  - Windows – COM, KDE - DCOP Gnome - CORBA (ish), TWM – forget it.
- Unix pipes? - different on Windows
- Lowest common denominator is HTTP-Get.
  - rough-n-ready procedure call – pass parameters in, get a result back.
  - every language supports this, or is superceded by a language that does. even bash, using curl

Noel Winstanley - nw@jb.man.ac.uk

HTTP and XMLRPC

- forerunner of SOAP
  - simpler types than SOAP, but better than raw HTTP-GET
- handy introspection methods
- wide range of implementations - http://xmlrpc.scripting.com/directory/1568/implementations
- So Apps can connect to Server via HTTP-GET and XMLRPC
- Efficiency?
  - not significant when calling remote services.
  - For client-side requests, at least the JVM is already running. Not suited for tightly-coupled libraries.

Astrogrid Common Runtime



ACR as infrastructure for other Applications

- single sign on, config, accessible libraries and vo delegates – makes a great framework for writing and deploying VO applications
- write an ACR plugin
  - tightly coupled, in-process calls to existing VO components
  - plugins can themselves export functionality
- link to ACR from external app
  - possible to deploy & launch ACR as part of webstarting external apps.

Further Work

- Other access methods
  - RMI / SOAP
- late-loading plugins
  - listed in registry?