Future Astronomical Software Environment

FASE

P. Grosbøl, ESO
D. Tody, VAO/NRAO
L. Paioro, INAF
Y. Granet, LAM/OAMP
B. Garilli, INAF
C. Surace, LAM/OAMP
OPTICON FASE Network

Outline

- Main Objectives
- Architecture and Design
- Prototype Implementation
- Conclusions and Outlook

Why a Common Environment?

- Difficult to share software
  - Not trivial to combine applications
  - Legacy systems have different scripts and API
- Reduced support of legacy systems
  - Old, monolithic designs
  - No full support of new IT infrastructures
  - Lack of new applications
- Limited scalability
  - Need to process large data sets
  - Utilize modern hardware options

OPTICON FASE Network

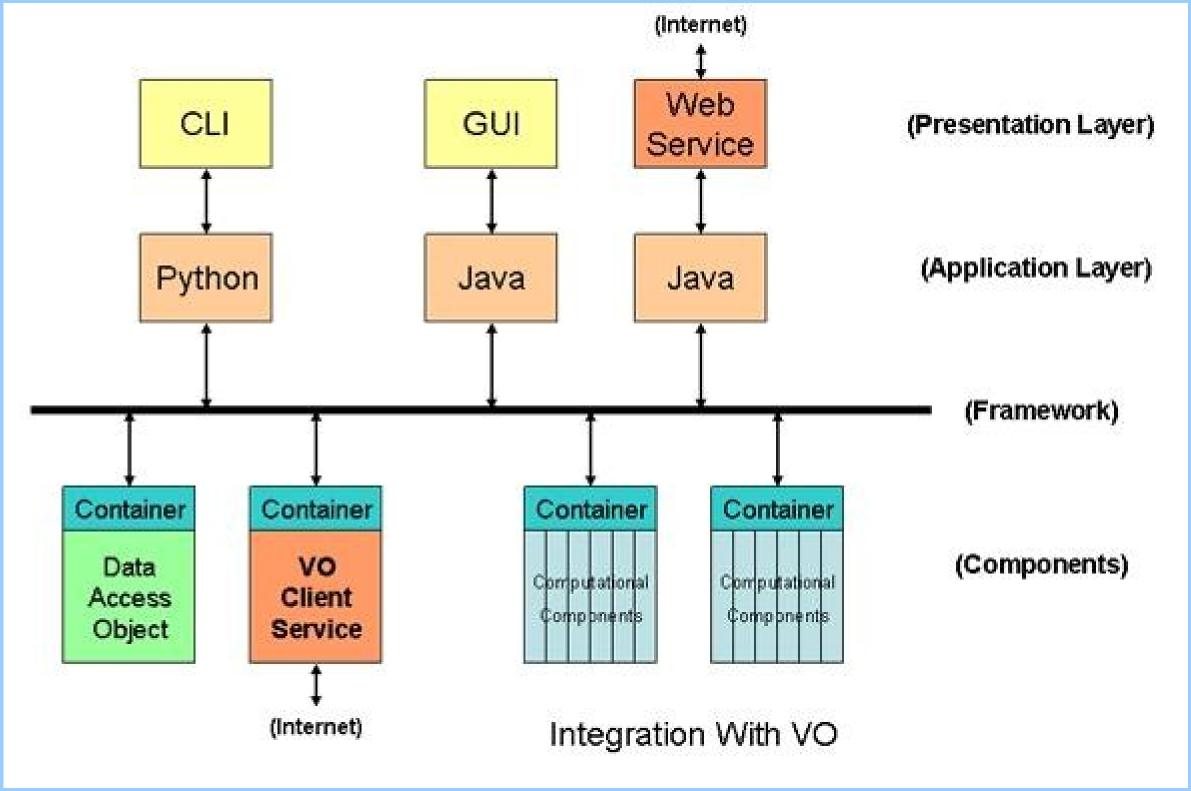
- FASE Network created to
  - Discuss needs for new environment
  - Outline main requirements
  - Propose architecture and design concepts
  - Demonstrate feasibility through prototype
- Funded by EC FP6/FP7 through OPTICON
  - Work 2004 $\rightarrow$ 2011
  - Meetings and 1-2 FTE for prototype (Milan+Marseille)
- Participation
  - Members from major European institutes/systems
  - Associate members from US (VAO and others)
- Twiki: https://www.eso.org/wiki/bin/view/Opticon

Requirements

- Support popular scripting languages
- Support standard compiles languages
- Easy development of new applications
- Separation between algorithms and IT
- Access to VO and Web applications
- Access to legacy applications/systems
- Deployment on laptops $\rightarrow$ clusters $\rightarrow$ …
- Provide scalability
- See e.g BoF at ADASS XVI

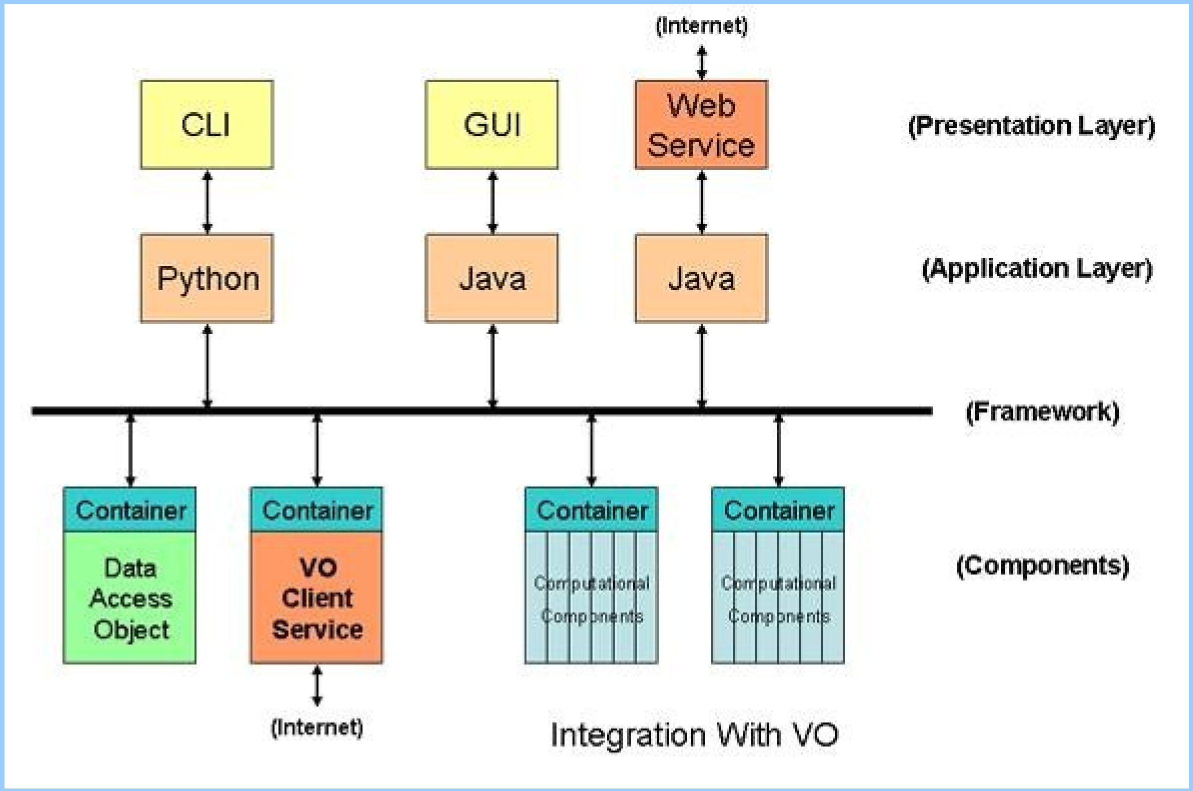
Architecture and Design

- OMG distributed object concept
  - Mature architecture
  - Language neutral
  - Allow multiple implementations of infrastructure
  - Main parts
    - Clients
    - Software bus
    - Services
    - Containers
    - Components



Elements of Environment

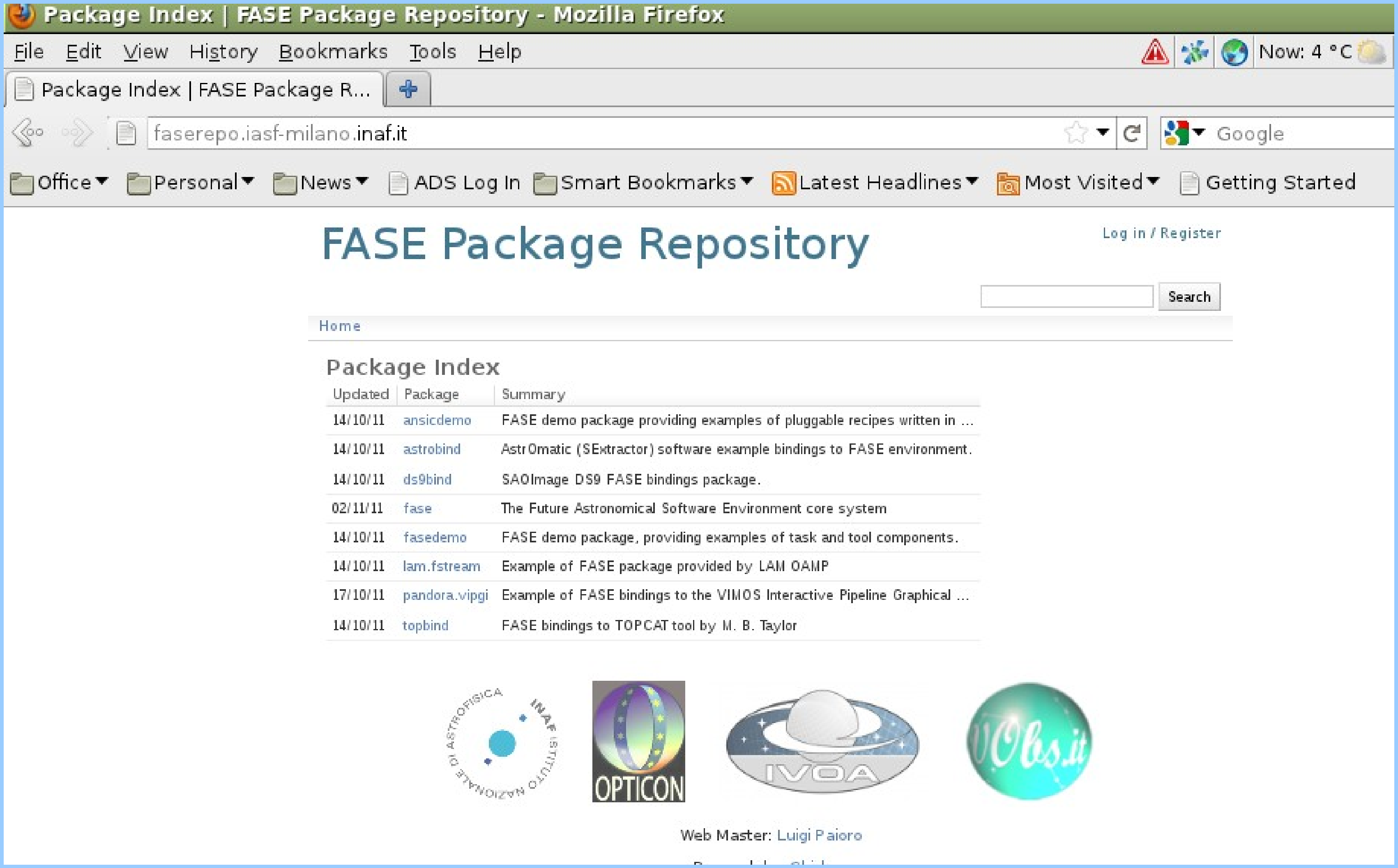
- Main items
  - Client interface
  - Application framework (Tody et al. 2009 ADASS XVIII)
    - Message bus
    - Package Manager
    - General services
    - Containers
  - Component interface



Prototype

- Proof-of-concept
  - Access to VO and legacy apps.
  - Basic scalability
- Choices
  - Python for scripting
  - Framework:
    - SAMP bus, Packaging Tool in Java, general services in Python
  - Container for Python, C/FORTRAN
- Implementation done by
  - Milan (INAF) and Marseille (LAM/OAMP)
- Prototype available for download
  - http://faserepo.iasf-milano.inaf.it/fase
- Details see Poster 109 Luigi Paioro et al.

Prototype Repository



Package Index | FASE Package Repository - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Now: 4 °C
Package Index | FASE Package R...
faserepo.iasf-milano.inaf.it
Google
Office Personal News ADS Log In Smart Bookmarks Latest Headlines Most Visited Getting Started
Log in / Register
Search
Home

FASE Package Repository

Package Index

Updated	Package	Summary
14/10/11	ansicdemo	FASE demo package providing examples of pluggable recipes written in ...
14/10/11	astrobind	AstrOmatic (SExtractor) software example bindings to FASE environment.
14/10/11	ds9bind	SAOImage DS9 FASE bindings package.
02/11/11	fase	The Future Astronomical Software Environment core system
14/10/11	fasedemo	FASE demo package, providing examples of task and tool components.
14/10/11	lam.fstream	Example of FASE package provided by LAM OAMP
17/10/11	pandora.vipgi	Example of FASE bindings to the VIMOS Interactive Pipeline Graphical ...
14/10/11	topbind	FASE bindings to TOPCAT tool by M. B. Taylor

Web Master: Luigi Paioro

Proof of concept

- Feasibility demonstrated by prototype
  - Scripting in Python or Unix shell CLI
  - Framework in Python, SAMP bus, Packaging Tool in Java
  - Basic containers for C and Python
  - Interface to legacy applications e.g. ESO CPL
  - Limited scalability demonstrated for CPU cluster
  - Interaction with VO enabled tasks through SAMP
  - Used for actual pipeline (Milan)
- Different Data Models not resolved by environment
  - Common Data Model e.g. VO
  - Explicit transformation

Conclusions I - Results

- Requirements for environment
  - Document with 200+ explicit requirements (wide review)
- Architectural concept
  - Common US+EU recommendation
  - Based on distributed object concept
- Feasibility demonstrated by prototype
- Compatible with VO
  - ref. VAO Desktop initiative
- Compliments VO

Conclusions II – What now?

- Technical implementation - simple
  - Use of available open-source tools
  - Ready for Reference implementation (<5 FTE)
- Main issues: Political + financial
  - Will to agree on common, shared API
  - Find organization to support environment
- Common API as important for software sharing as FITS has been for data sharing