

A PPARC funded project



Service-to-service security

VOtech kick-off
Cambridge
November 2004

Guy Rixon
AstroGrid Technical Architect
University of Cambridge

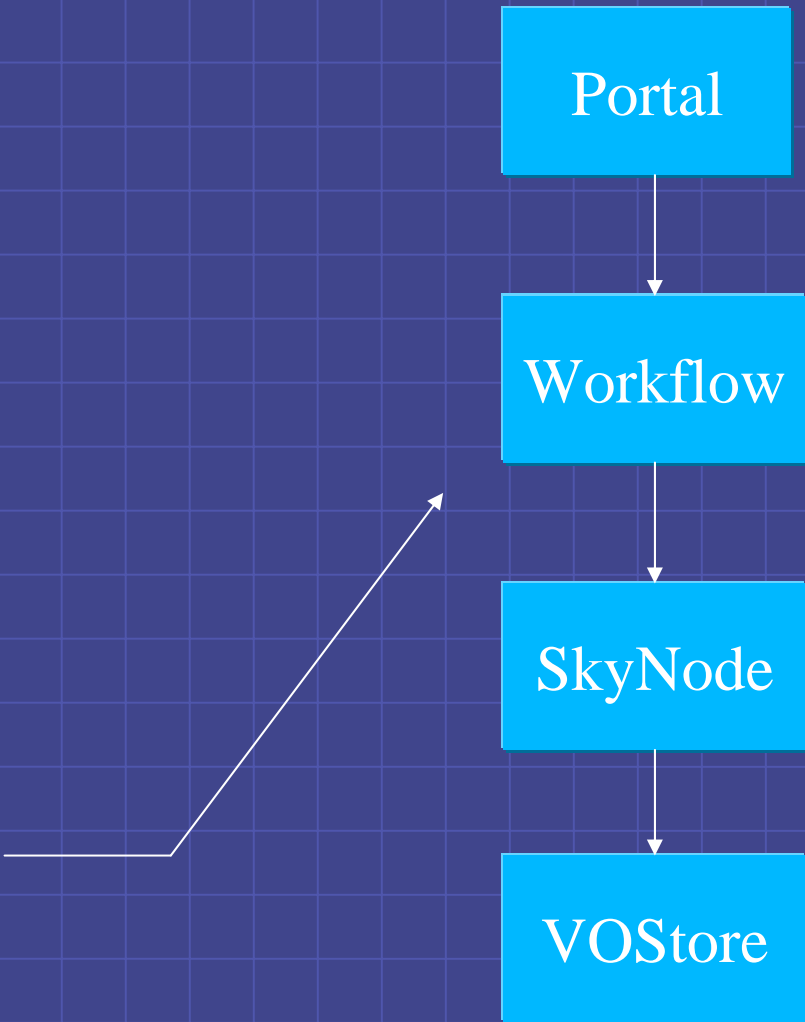


Jodrell Bank
Observatory

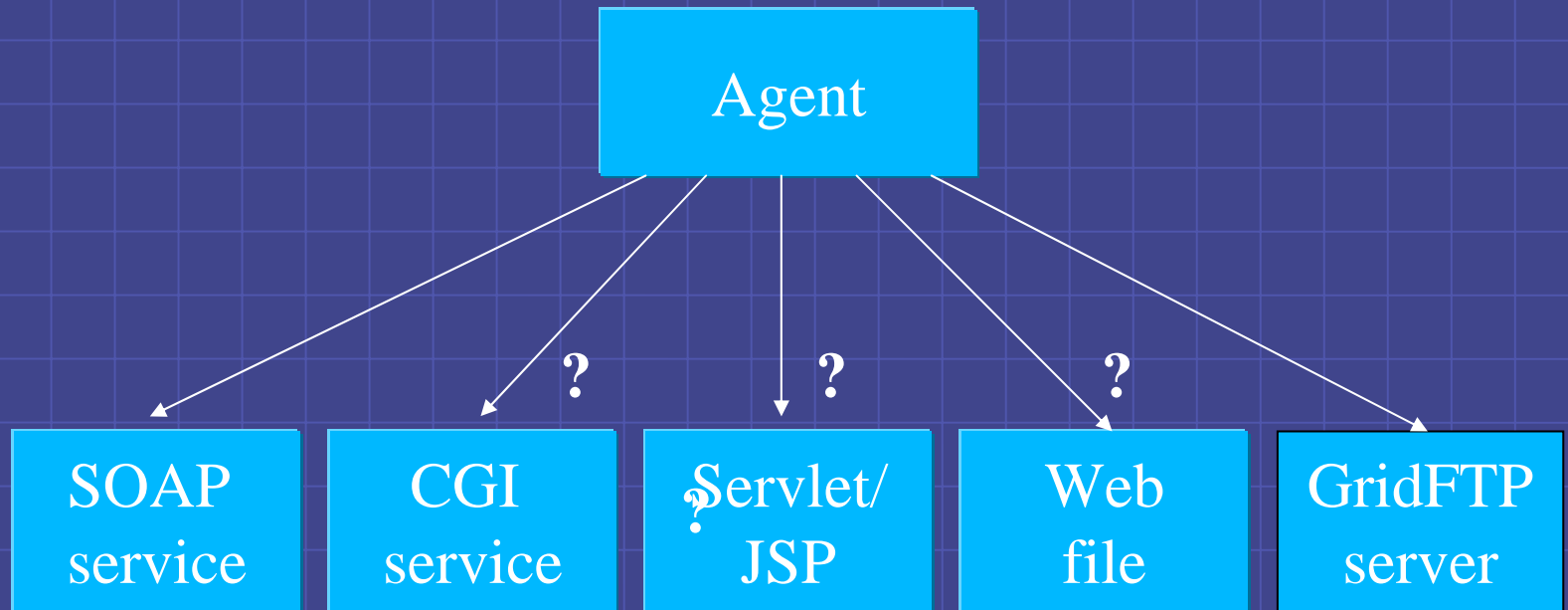


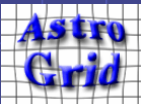
Scope 1: purpose of services

- Authentication, Authorization and Accounting for:
 - Data centres
 - VOStores
 - Registries
 - Web portals
 - Workflow engines
 - [etc]
- ...and chains delegation between the above services



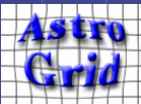
Scope 2: technology of services





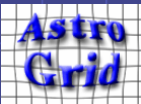
Scope 4: level of security

- Privilege separation between users: yes.
- Secure against accidental mis-use: yes.
- Secure against deliberate mis-use via public i/f: yes.
- Secure against intrusion by message alteration: maybe.
- Secure against intrusion via message capture and replay: maybe.
- Secure against intrusion via compromised host: maybe.



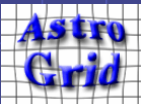
Scope 3: number of entities

- $\sim 10^4$ end-users
- $\sim 10^2$ sites
- $\sim 10^3$ coarse-grained resources
 - E.g. surveys
- $\sim 10^{8+}$ fine-grained resources
 - E.g. individual images



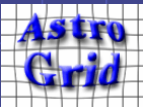
Single-sign-on authentication

- Sign on to entire VO once per interactive session
 - => exchange user-to-system credentials for service-to-service credentials
 - => sign-on point generates/stores service-to-service credentials
- Sign on to same account from multiple UIs/applications
 - => sign-on point is separate from UI/web portal
 - => "community" service to effect SSO
- Register only once, not once per service



Single registration

- Register once for all of VO, not once per service
 - Don't make service providers administer every user
 - Use local knowledge of registrations for checks
 - => community service handles registration
 - => service providers need to trust communities
 - => mapping to local accounts in services should be automatic and hidden.



Authorization: two levels

Coarse grained

Agent/
client

Authenticates to ↓

Data
centre

Checks authorization in ↓

Community
service

Fine grained

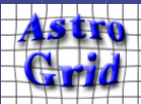
Agent/
client

Authenticates to ↓

Data
centre

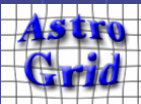
Checks authorization in ↓

Local
DB/service



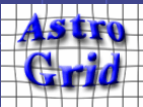
Accounting

- Record improper usage
 - Detect intrusions
 - Cleanse systems
 - Persecute attackers even unto 7th generation
 - Reassure users/partners
- Record proper usage
 - Capacity planning
 - Reports to funding bodies
 - Possible quota tracking
 - Bragging rights
 - => need all/most calls to services to identify user



Software frameworks (1)

- *"It's easy to implement our design for AAA; we provide a software framework."* – toolkit author
- *"We have a local framework for AAA in our services."* – data-centre operator
- *"Frameworks fight each other like pit bulls, you can't have more than one."* – Roy Williams



Frameworks (2)

VO Framework

VO Framework

Service

Gateway

Local Framework

Local Framework

Service



VO Framework

Defined protocol

Agent

Gateway

Service

Local Framework

Defined protocol

