Access Grid 2.0

Virtual Venues

Overview

- The spatial metaphor implies
  - Virtual Rooms
  - Connections between them

In the AG, the rooms are referred to as **Venues**, and the space in which Venues exist and are connected is called the **Venues Fabric**.

Use Cases I

- Actor: User
  - Enter/exit a Venue
  - Add data/services to a Venue
  - Retrieve data from a Venue
  - Receive services from a Venue
  - Remove data/services from a Venue
  - Resolve capabilities

Use Cases II

- Actor: Venue Administrator
  - Create a new Venue
  - Specify Venue owner
  - Specify Venue access policy
  - Add an exit to a Venue

Requirements I

- Virtual Venues
  - Venues are persistent
  - Venues are coherently experienced
  - Venues are collaboration scopes
  - Venues are extensible
  - Venues broker capabilities

Requirements II

- Virtual Venues Fabric
  - Venues fabric is scalable
  - Venues fabric is topological

Architecture I

- Attributes
  - Environment
  - Authorization Policy
  - Entrances and Exits
  - Users
  - Data
  - User Level Services
  - Network Services
  - Streams

Architecture II

- Operations
  - Modifying the environment
  - Modifying authorization policy
  - Modifying topology
  - Entering the venue
  - Exiting the venue

Architecture III

- Operations (continued)
  - Adding and Removing Data
  - Adding and Removing Services
  - Brokering Capabilities
  - Adding and Removing Streams

Technology Strategy

- Venues will be deployed through a web services-based Venues Server
- Identity is outside the scope of Venues

Technology Choices

- The Venues Server will implement a SOAP interface
- The Venues Server interface will be described using WSDL
- Venues security will employ Globus

Design I

- Venues Parts
  - Venue – The persistent, coherent, secure scope for interactions.
  - Venue Service – How the Venue is made network accessible
  - Venue Server – ???

Design II

- Environment: descriptive information that is shared with all participants of the Venue to create a sense of shared reality.
  - Venue Name
  - Venue Description, in some standard Markup Language
  - Optional Items: Pictures, Icons, Background Music
- Authorization Policy: a policy that is enforced by the Venue when users try to perform venue operations.
  - Authorization will default to as open as possible

Design III

- Users: a list of user profiles for participating users.
  - Identification information, e.g. Globus Certificate
  - Capabilities Information
- Connections: a list of entrances and exits that connect this venue to other venues – like hyperlinks in web pages.
  - Description of what's on the other end
  - Location of the other side, e.g. URL

Design IV

- Services: a list of service descriptions for services found in the Venue, e.g. Voyager.
  - Service Information
  - Service Location
- Data: a list of data descriptions for data found in the Venue.
  - Data Description
  - Data Location

  We are planning for data to reside outside of the Venue, what the technology choice for that data server is hasn't been determined.

Design V

- Network Services: a list of service descriptions for services available to the venue during capabilities brokering.
  - Description of interfaces
  - Description of functionality
  - Location of the Service
- Streams: a list of active stream configurations
  - Description of the media stream
  - Location of the media stream, e.g. Network Address

Open Issues

- How do we create Venues?
- How does one author authorization policies?
- Who issues certificates?
- How do you trust a certificate?
- Do Venues need certificates? Other Services?
- How does SOAP enable multiple Services on one Server?

Conclusion

- To Dos:
  - Venues design is not yet complete
  - Current prototype uses XMLRPC instead of SOAP
  - Schemas need to be created
  - WSDL needs to be generated
- Current panics
  - SC02

Final Notes

- Questions?

- We'll meet monthly:
  - The first Tuesday of every month
  - 9AM Central – 11 AM Central
  - Argonne Institutional Venue