Developing Shared Applications

Susanne Lefvert

June 10, 2004

Outline

- Overview
- Shared Application Client
- Example: Question and Answer Tool
  - Use cases/requirements
  - Design
  - Implementation
  - Packaging and distribution
- Documentation

Outline

- **Overview**
- Shared Application Client
- Example: Question and Answer Tool
  - Use cases/requirements
  - Design
  - Implementation
  - Packaging and distribution
- Documentation

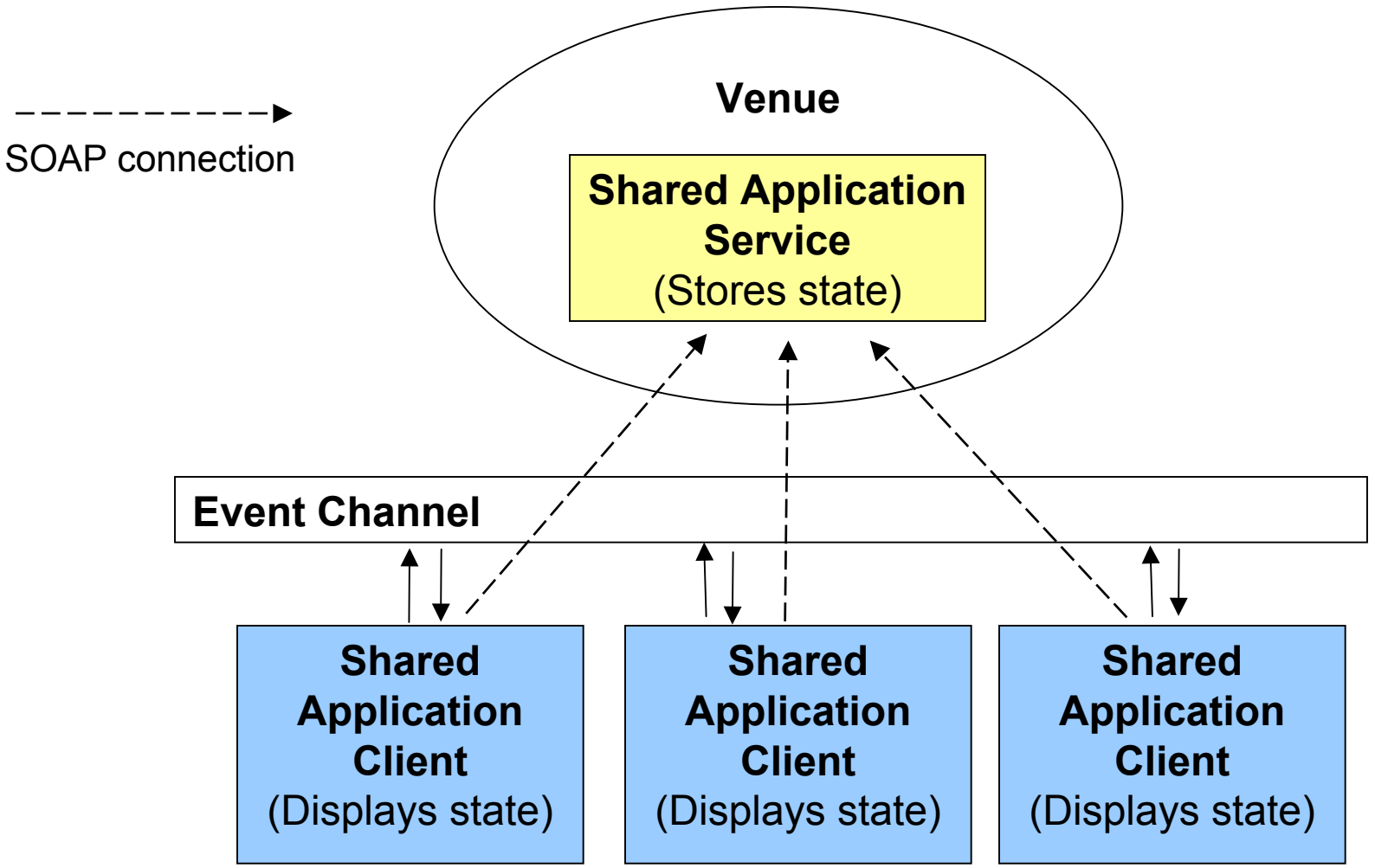
Overview

- Enhance collaboration
- Several applications included in AGTk
  - Shared Browser
  - Shared Presentation Viewer
- Framework to support plug-in applications

AGTk Environment

- Security, authorization, and authentication
- Framework for SOAP messages
- Invocation of the application code

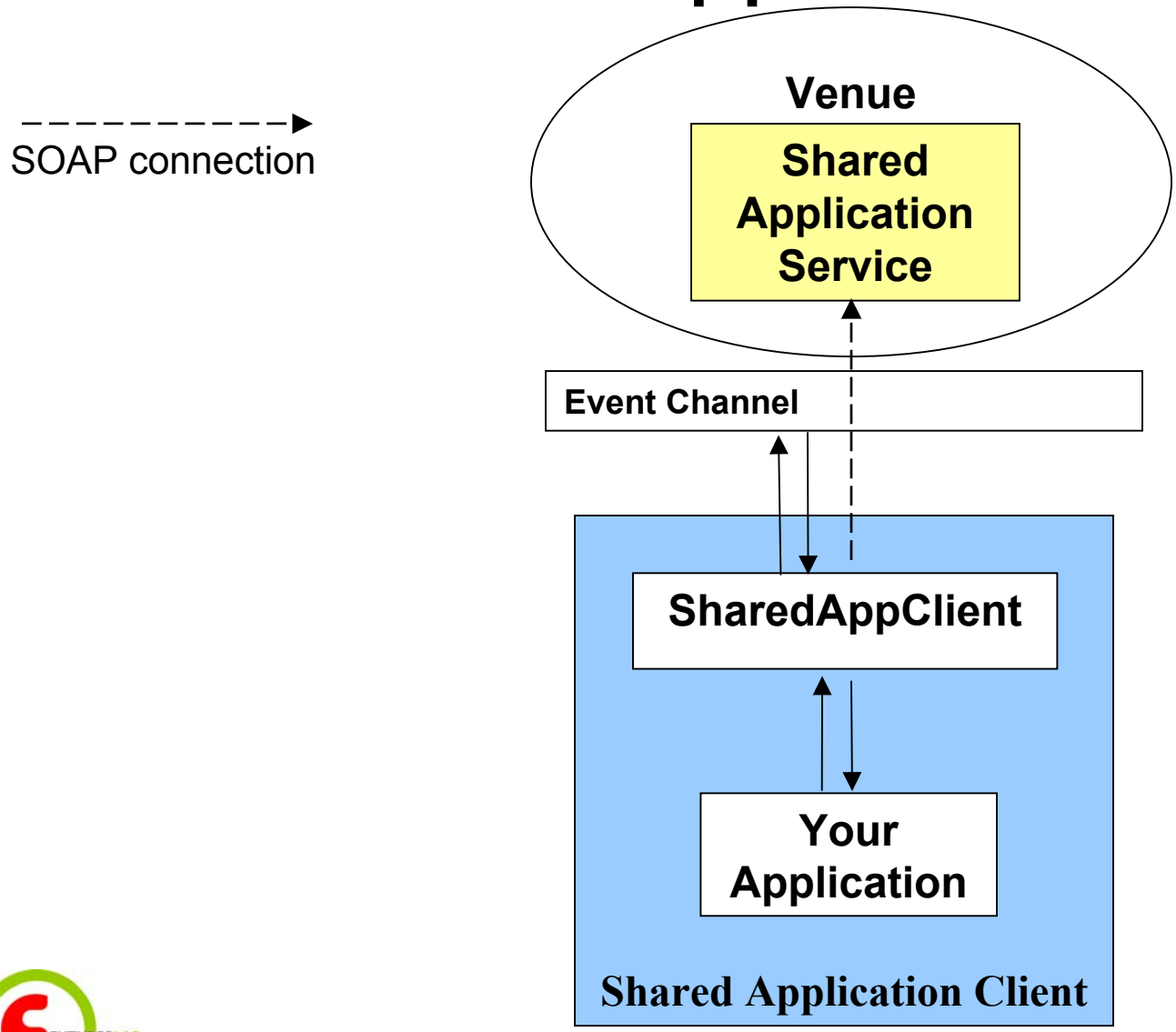
Shared Application

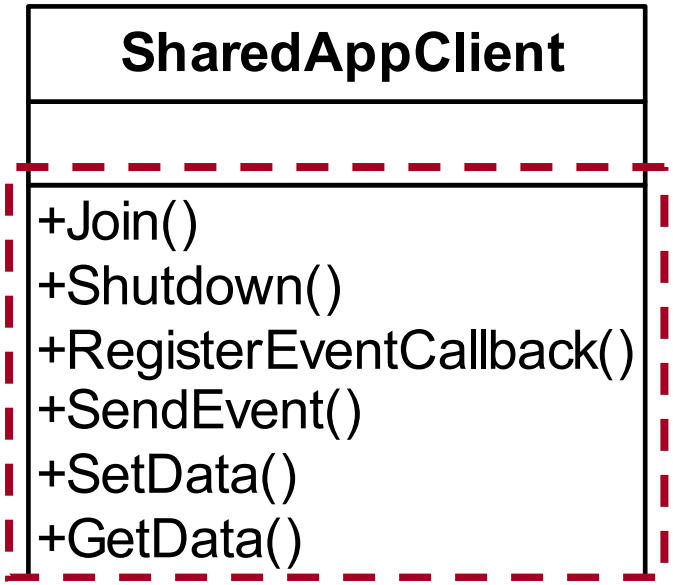


Outline

- Overview
- **Shared Application Client**
- Example: Question and Answer Tool
  - Use cases
  - Design
  - Implementation
  - Packaging and distribution
- Documentation

Shared Application Client




SharedAppClient
+Join()
+Shutdown()
+RegisterEventCallback()
+SendEvent()
+SetData()
+GetData()

Outline

- Overview
- Shared Application Client
- **Example: Question and Answer Tool**
  - Use cases/requirements
  - Design
  - Implementation
  - Packaging and distribution
- Documentation

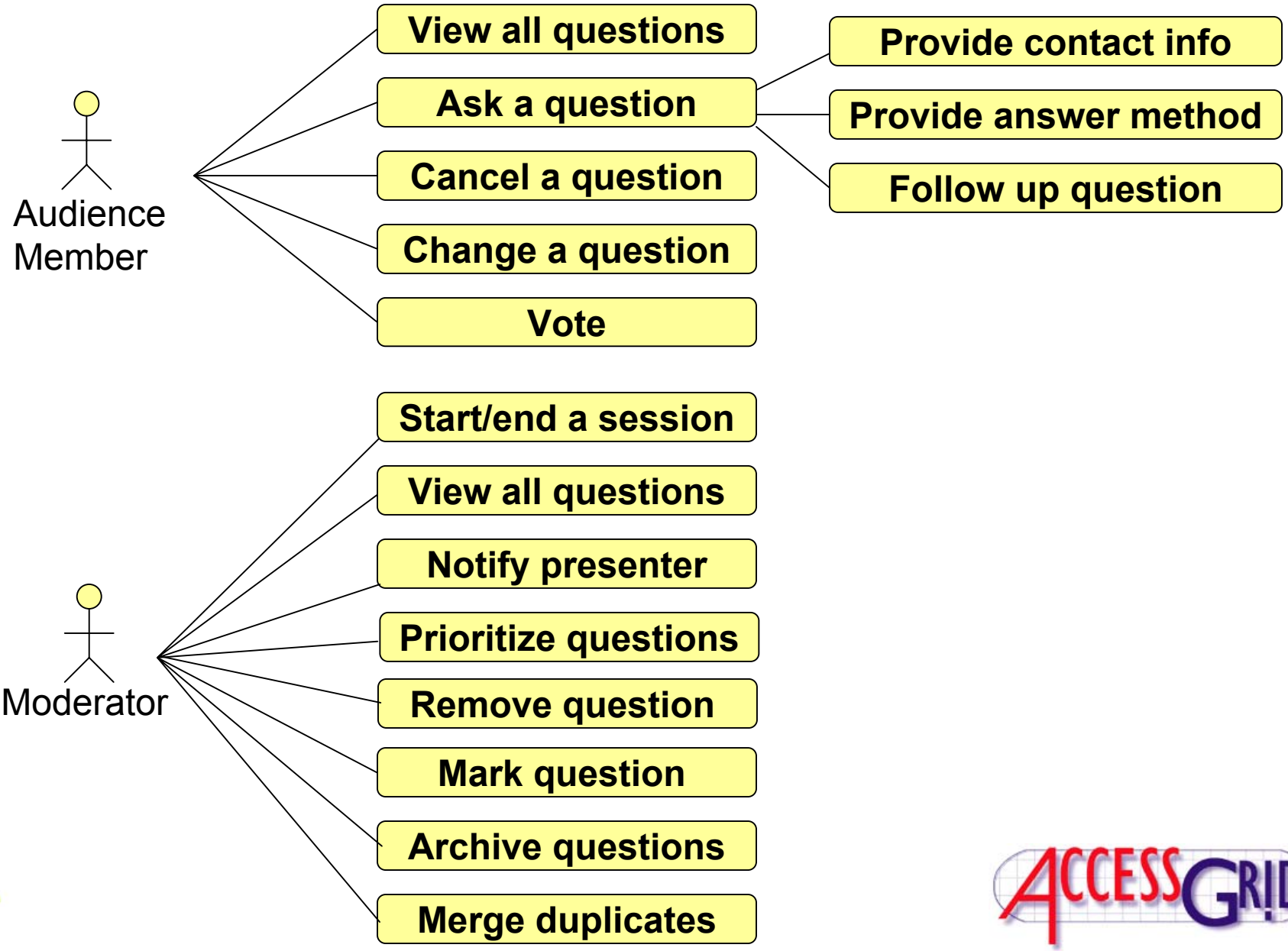
Example Shared Application:

Question and Answer Tool

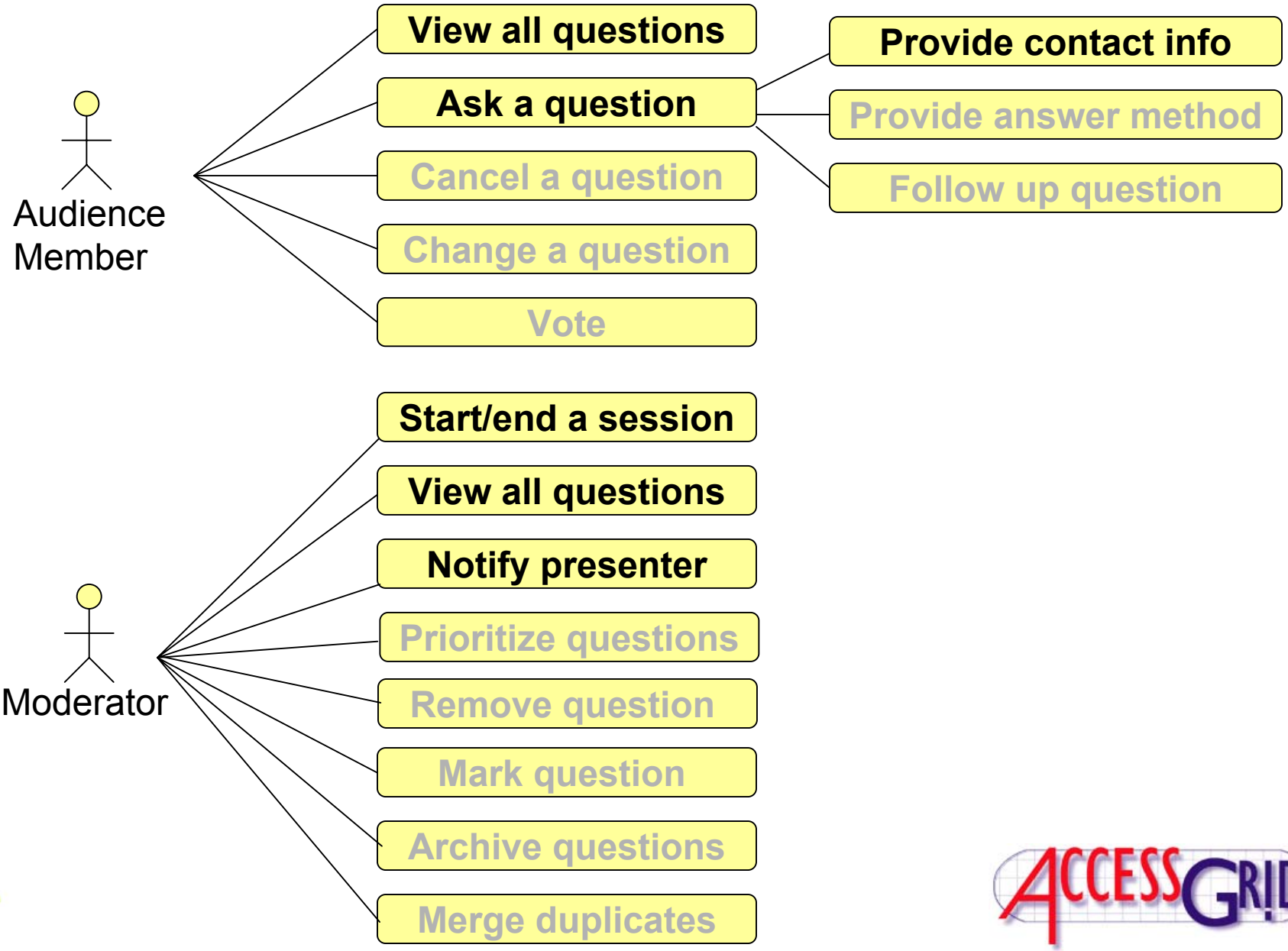
Outline

- Overview
- Shared Application Client
- Example: Question and Answer Tool
  - **Use cases/requirements**
  - Design
  - Implementation
  - Packaging and distribution
- Documentation

Ask a question at a seminar.



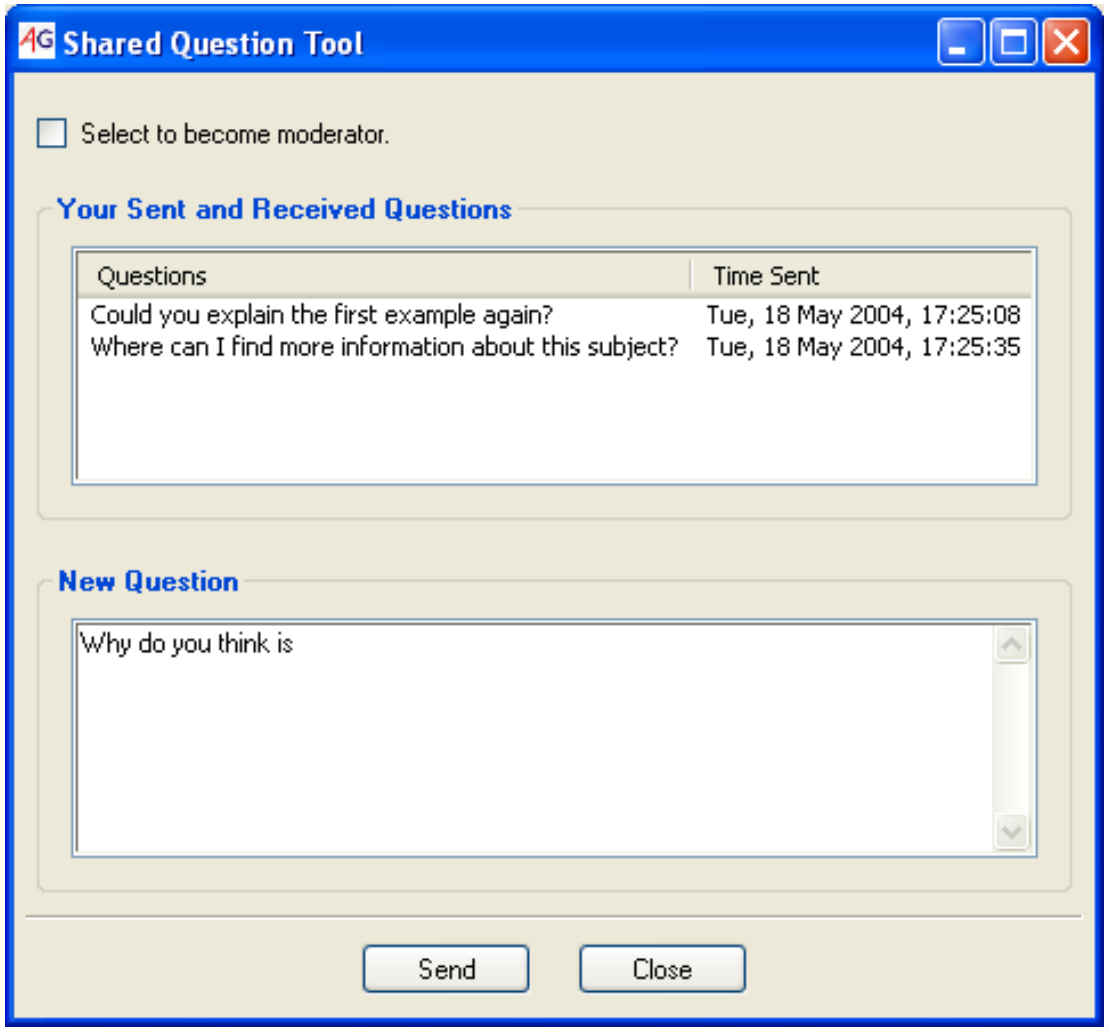
Ask a question at a seminar.



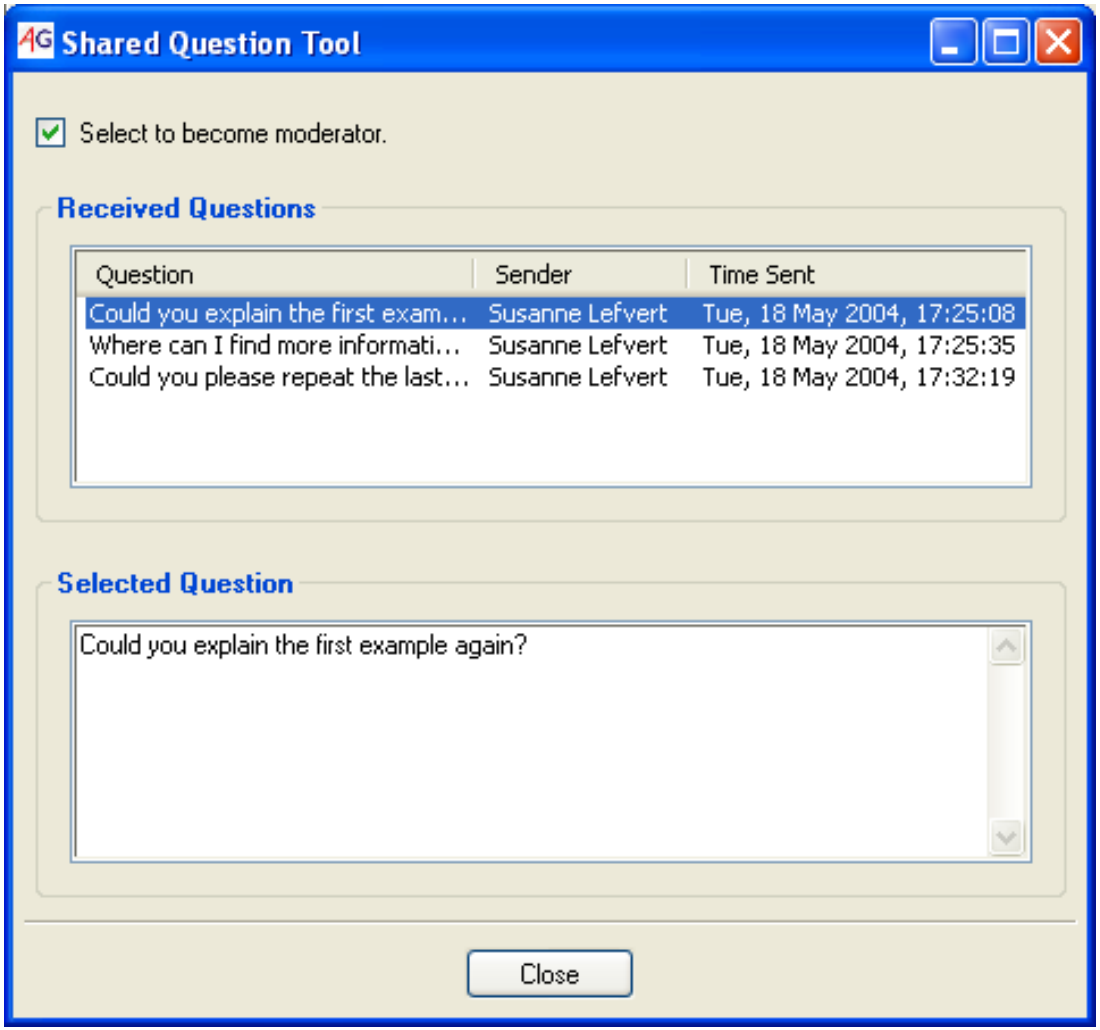
Outline

- Overview
- Shared Application Client
- Example: Question and Answer Tool
  - Use Cases/requirements
  - **Design**
  - Implementation
  - Packaging and distribution
- Documentation

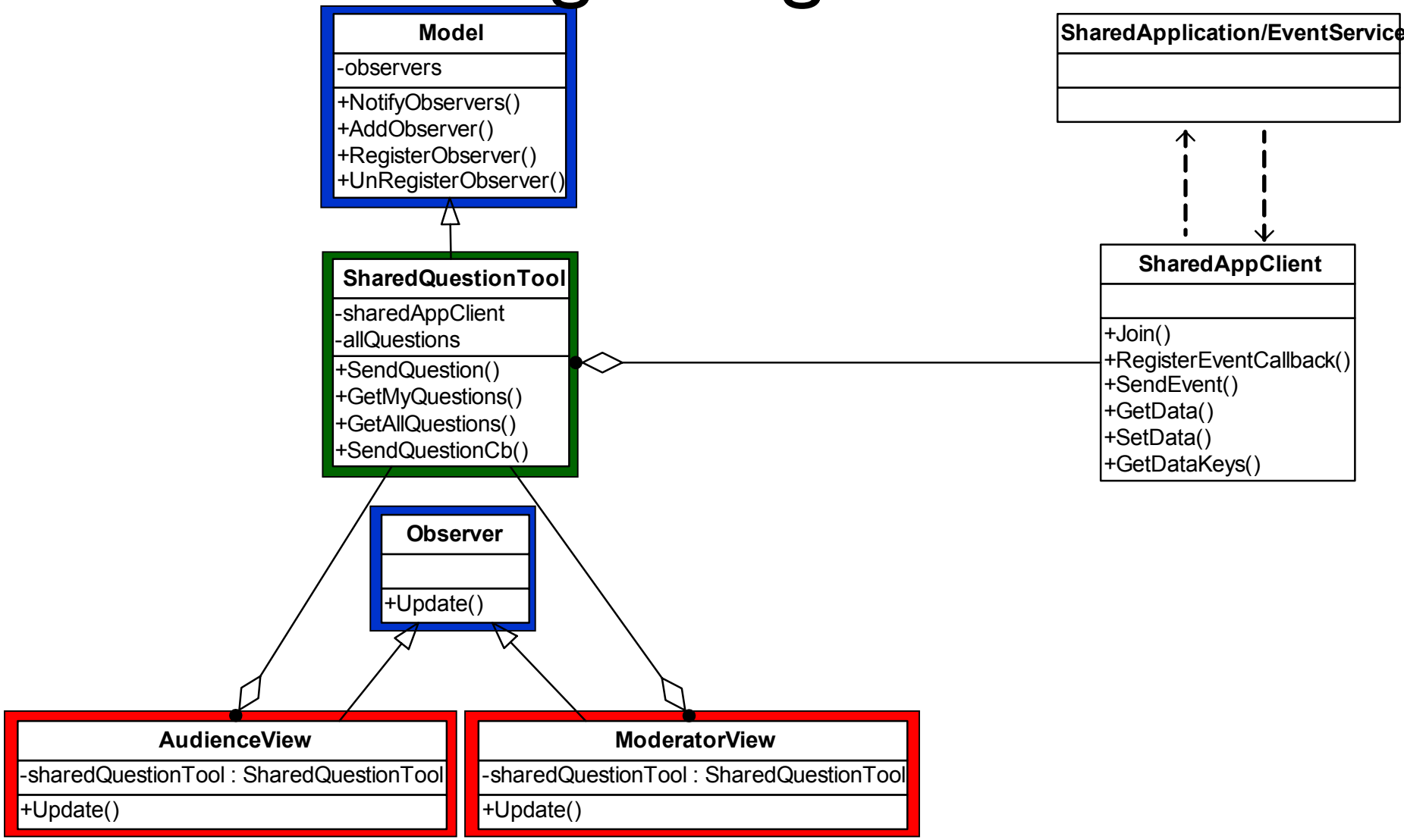
User Interface Design
Audience View



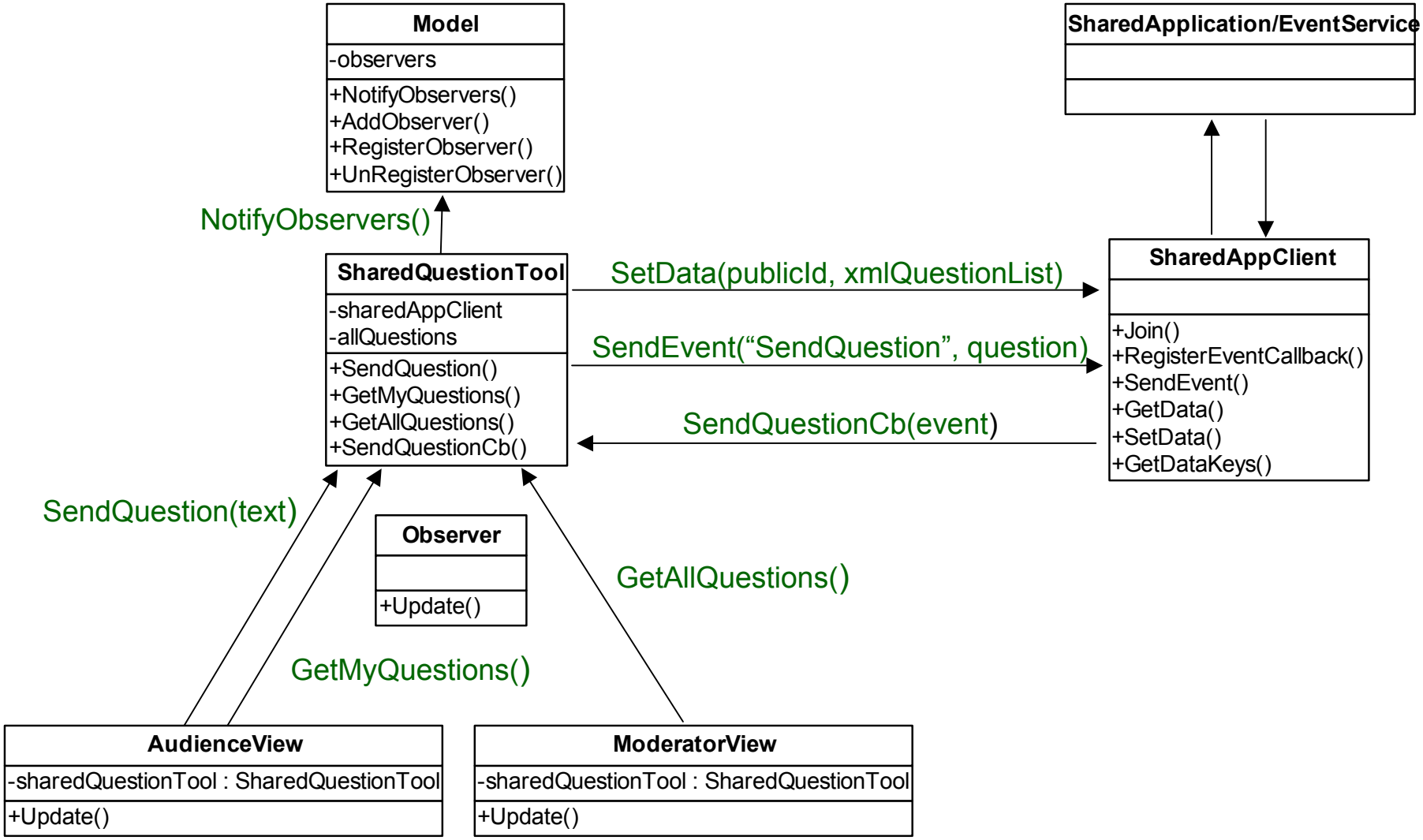
User Interface Design
Moderator View



Design Diagram



Design Diagram



Outline

- Overview
- Shared Application Client
- Example: Question and Answer Tool
  - Use cases/requirements
  - Design
  - **Implementation**
  - Packaging and distribution
- Documentation

```
class SharedQuestionTool(Model):
  SEND_QUESTION = "SendQuestion"

  def __init__( self, appUrl, debugMode = 0, logFile = None):
    Model.__init__(self)
    self.allQuestions = []

    # Create shared application client
    self.sharedAppClient = SharedAppClient("SharedQuestionTool")

    # Get client profile
    configDir = UserConfig.instance().GetConfigDir()
    clientProfileFile = os.path.join(configDir, "profile")
    self.clientProfile = ClientProfile(clientProfileFile)

    # Join the application session
    self.sharedAppClient.Join(appUrl, self.clientProfile)
    self.publicId = self.sharedAppClient.GetPublicId()

     # Register event callback
     self.sharedAppClient.RegisterEventCallback(self.SEND_QUESTION,
     self.SendQuestionCb)

    # Get all questions currently stored in the application service
    clients = self.sharedAppClient.GetDataKeys()

    for clientId in clients:
        list = self.sharedAppClient.GetData(clientId)
        if len(list) > 0:
            qlist = self.__FromXML(list)
            for question in qlist:
                self.allQuestions.append(question)
```

```
def SendQuestion(self, text):
    '''Called when an audience member clicks the send button.'''
    question = self.__CreateQuestion(text)
    myQuestions = self.GetMyQuestions()
    myQuestions.append(question)

    # Store questions in the venue application service.
    questions = self.__ToXML(myQuestions)
    self.sharedAppClient.SetData(self.publicId, questions)

    # Send the event.
    self.sharedAppClient.SendEvent(self.SEND_QUESTION,
     question)
    return question

def SendQuestionCb(self, event):
    '''Callback invoked when SEND_QUESTION events arrive. '‘
     # Store data
    question = event.data
    self.allQuestions.append(question)

   # Notify user interface.
    self.NotifyObservers()

def GetAllQuestions(self, clientId = None):
    '''Returns the entire question list.'''
    return self.allQuestions

def GetMyQuestions(self):
    '''Returns my question list.'''
    myQuestions = []
    for question in self.allQuestions:
       if question["appId"] == self.publicId:
          myQuestions.append(question)
    return myQuestions
```

Outline

- Overview
- Shared Application Client
- Example: Question and Answer Tool
  - Use cases/requirements
  - Design
  - Implementation
  - **Packaging and distribution**
- Documentation

Application Description Files

SharedQuestionTool.app

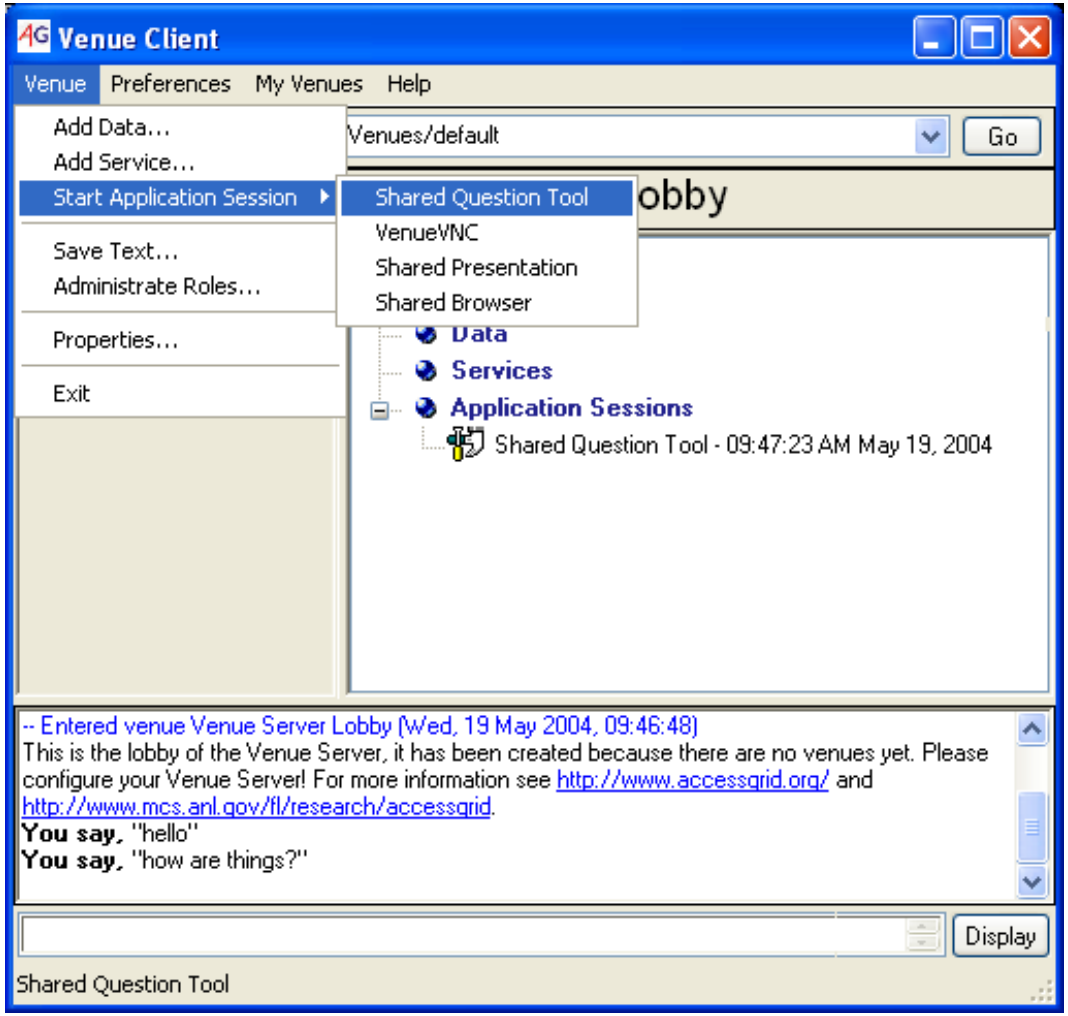
```
[application]
name = Shared Question Tool
mimetype = application/x-ag-question-tool
extension = sqtool
files = SharedQuestionTool.py, ObserverPattern.py

[commands]
Open = %(python)s SharedQuestionTool.py -a %(appUrl)s
```

Access Grid Package Manager (AGPM)

Use AGPM to register the application with your user environment.

- agpm.py –f SharedQuestionTool.app
- agpm.py –h (for more options)

Start Application Session



Access Grid Package Extension (agpkg)

1. Create a zip package of your files
   - SharedQuestionTool.py
   - ObserverPattern.py
   - SharedQuestionTool.app
2. Rename it with extension agpkg
   - SharedQuestionTool.agpkg
3. Opening SharedQuestionTool.agpkg from Venue Client installs the application

Outline

- Overview
- Shared Application Client
- Example: Question and Answer Tool
  - Use cases/requirements
  - Design
  - Implementation
  - Packaging and distribution
- **Documentation**

Documentation

- Contributed Software

http://www.mcs.anl.gov/fl/research/accessgrid/wiki/moin.cgi/ContributedSoftware

- Reference Materials

http://www.mcs.anl.gov/fl/research/accessgrid/wiki/moin.cgi/DevelopingSharedApplications

Credits

- This work was supported in part by the Mathematical, Information, and Computational Sciences Division subprogram of the Office of Advanced Scientific Computing Research, Office of Science, U.S. Department of Energy, under Contract W-31-109-ENG-38.

- This material is based upon work supported in part by the National Science Foundation under Grant No. ANI-0222509.

- This material is supported in part by Microsoft Research.

- This material is supported in part by the National Institute of Health/National Library of Medicine as part of the Advanced Biomedical Collaboratory project.