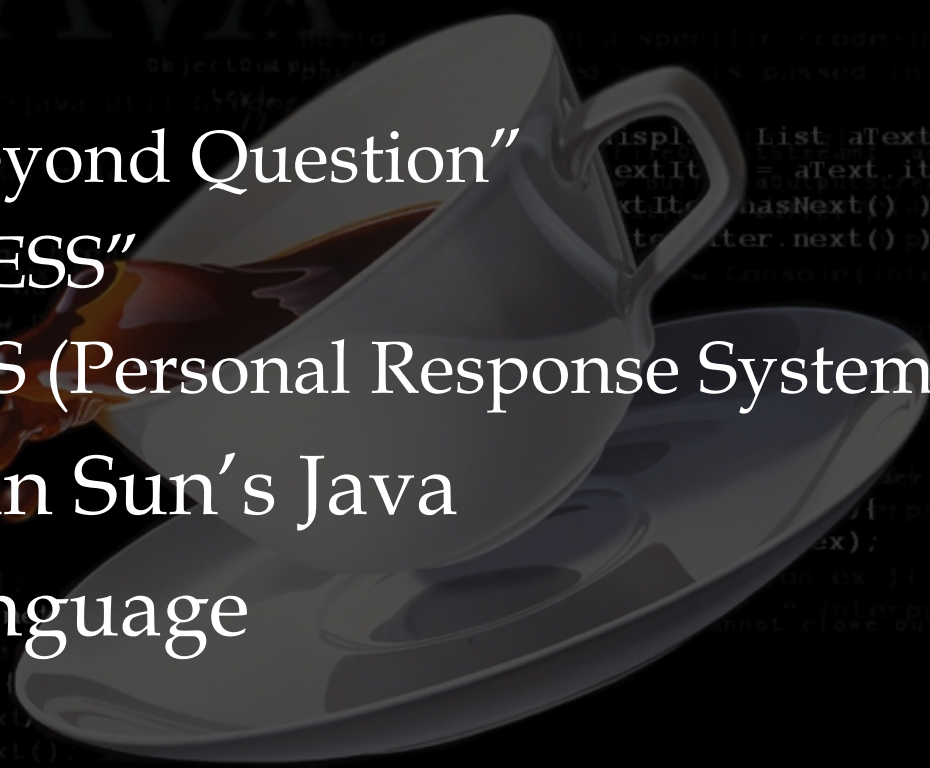




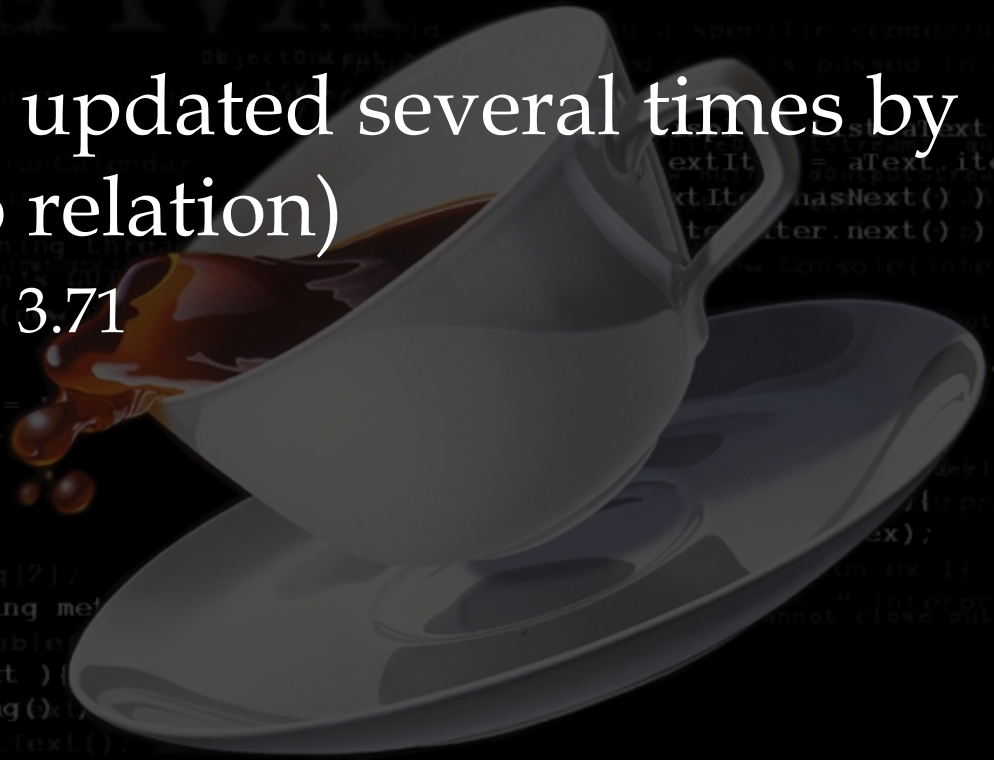
The Current QuizMaster

- Student Response System
- Internet-based “Ask the Audience”
- Similar to:
 - Smartroom’s “Beyond Question”
 - Sun-tech’s “XPRESS”
 - InterWrite’s “PRS (Personal Response System)”
- Written entirely in Sun’s Java programming language



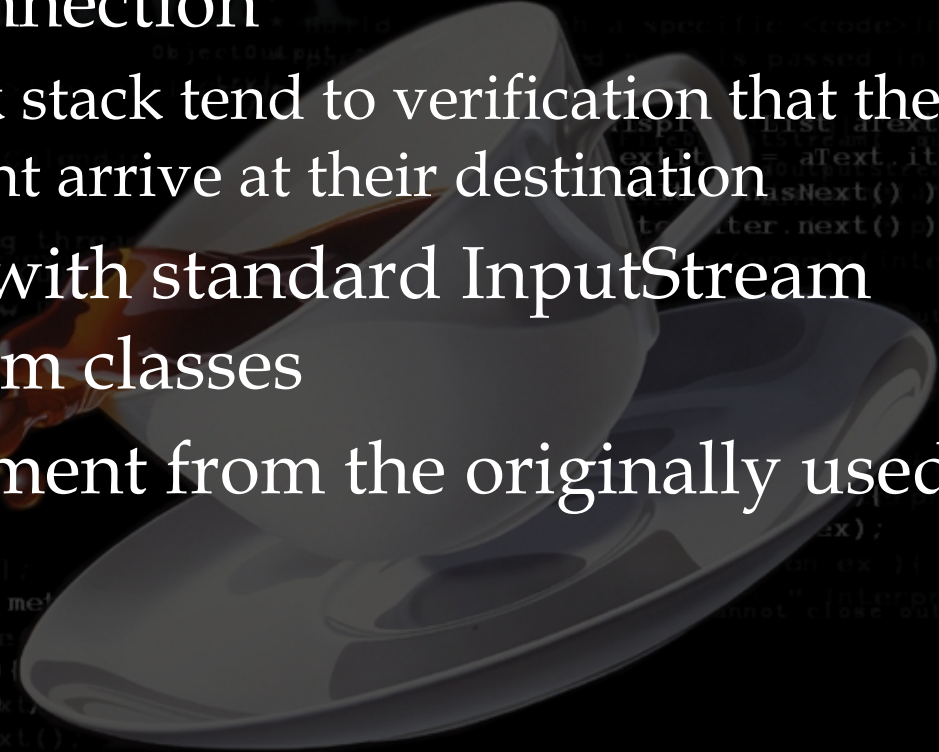
The History

- First developed as the Senior Seminar project of Philip Smith
- Version 1.0
- Taken over and updated several times by Allen Smith (no relation)
- Versions 1.01 – 3.71



The Background

- QuizMaster uses java.net.Socket for communication
 - TCP protocol connection
 - Lets the network stack tend to verification that the bytes that are sent arrive at their destination
 - Easily wrapped with standard InputStream and OutputStream classes
 - Drastic improvement from the originally used java Datagrams



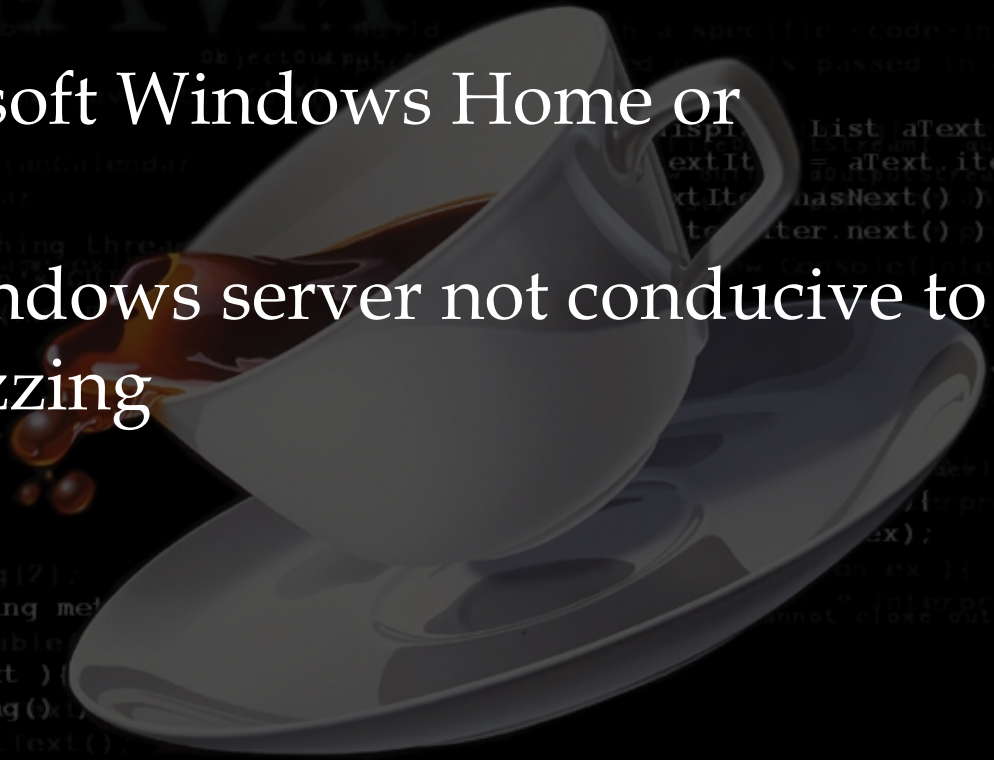
The Background

- Most communication is done through passing of DataPacket objects
 - DataPacket is a custom class implemented in version 2.0 by Allen Smith
 - Another major improvement from version 1.0



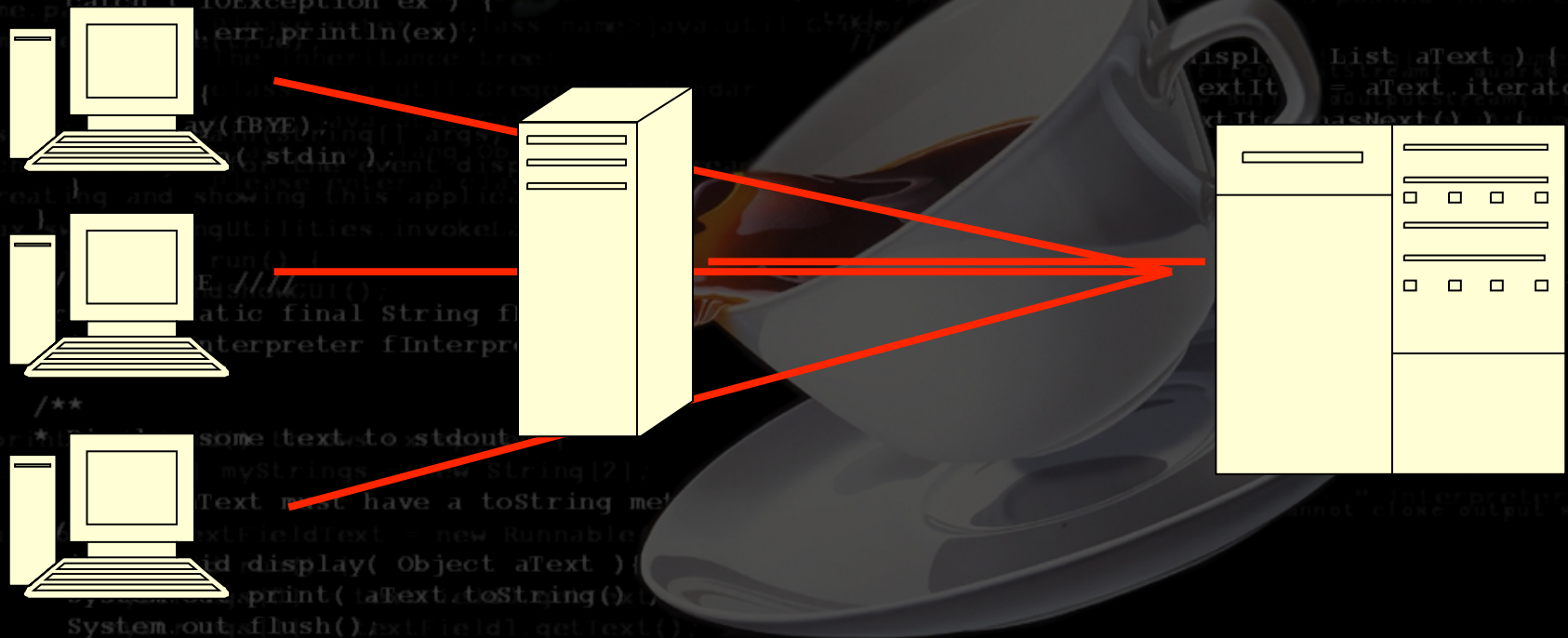
The Idea

- QuizMaster had a major limitation
 - Connection limit on simultaneous incoming requests
 - Only on Microsoft Windows Home or Professional
 - Location of Windows server not conducive to interactive quizzing



The First Attempt

- Rewrite StreamConnection and ClientConnectionThread classes to communicate through a proxy



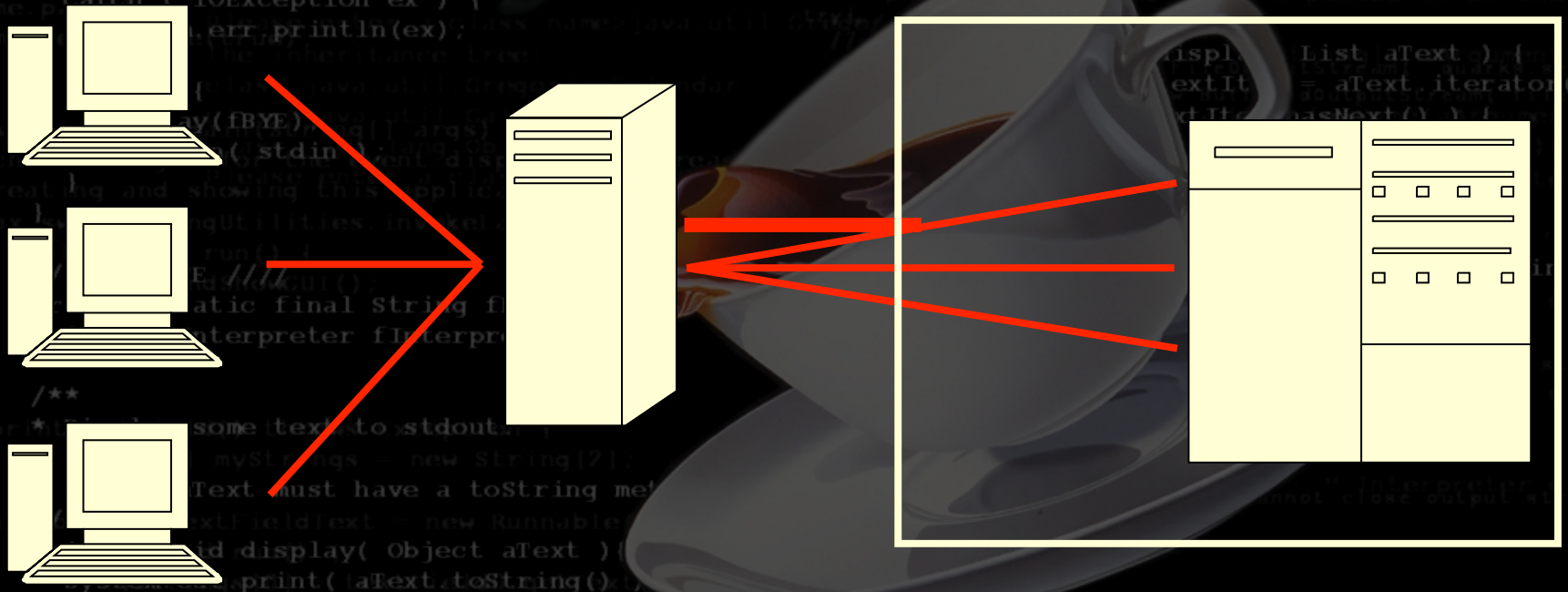
The First Attempt

- Rewrite StreamConnection and ClientConnectionThread classes to communicate through a proxy
- First big snag:

```
104
105 if(acceptDialogChoice == ACCEPT){
106     System.out.println("ACCEPT");
107     //Add this new player as a brand new player
108     connections.addElement( incomingConnection );
109     incomingConnection.start(); //Fire up this thread to start receiving
110     int playerIndex = getNumberOfPlayers() - 1;
111     connections(playerIndex).setPlayerID( playerIndex );
112     System.out.println(playerIndex);
113
114     //this is one of very few places DataPackets are not used
115     connections(playerIndex).writeInt( playerIndex );
116
117     System.out.println("Server writes int: " + playerIndex);
118     parent.status.addPlayer(connections(playerIndex).getPlayerName(), playerIndex);
119     parent.finalReport.addPlayer(!acceptConnections); //if acceptConnections is false, that will fl
120     setEnabled(playerIndex, true);
121     //Only indicate a new connection if the quiz HASN'T started; otherwise, the question text will b
122     if(acceptConnections == true){
123         parent.questionArea.setText("Please begin connecting to " +
124             InetAddress.getLocalHost().getHostAddress() + "\n\n" +
125             "Connections Established: " + getNumberOfPlayers());
126     }
```

The Second Attempt

- Byte level proxy
- Still needed to eliminate the connection problem



```

/**      public void run() {
* Create the JFrame and show it.  (To be called early,
* this method should be invoked from the
* event dispatcher thread.)
*/
//pass each line of input to fInterpreter, and display
//fInterpreter's result
//use buffering
InputStreamReader inputStreamReader = new InputStreamReader ( System.in );
BufferedReader stdin = new BufferedReader ( inputStreamReader );
//Make sure we have a window before we start
//create a JFrame
boolean hasRequestedQuit = false;
String line = null;
//Create and set up the window
List result = new ArrayList();
try {
    while (!hasRequestedQuit) {
        line = stdin.readLine();
        //note that "result" is passed to fInterpreter
        hasRequestedQuit = fInterpreter.interpret(line);
        display(result);
        result.clear();
    }
} catch ( IOException ex ) {
    System.err.println(ex);
} finally {
    display(fBYE);
    shutdown(stdin);
}

//creating and showing this application
javax.swing.SwingUtilities.invokeLater(
    new Runnable() {
        public void run() {
            //PRIVATE
            private static f = new f();
            f.run();
        }
    }
);

/**
void printDisplay some text to stdout {
    final String[] myStrings = new String[2];
    * @param aText must have a toString method
Run*/
private void display( Object aText ) {
    System.out.print( aText.toString() );
    System.out.flush();
}
}

```



QuizMaster Demo

References

- Allen Smith
 - Comments in the existing QuizMaster code
- java.sun.com
 - Extensive online documentation for the Java platform 1.4.2 Software Development Kit
- Kevin Hieb
 - Reminding me how to use Vectors in Java

