



Applets – Belebung von Webseiten

Dipl.-Ing. Wolfgang Beer

Was sind Applets?



- Java Klassen, die spezielle Richtlinien befolgen, um:
"in Internet-Browsern lauffähig zu sein"
- Somit ist,
 - komplexere Funktionalität,
 - dynamische Grafik,

auch in herkömmlichen HTML-Seiten möglich
- Alternativen:
 - JavaScript, JSP, Servlets, CGI, ASP,
 - bei Graphik: Macromedia Flash

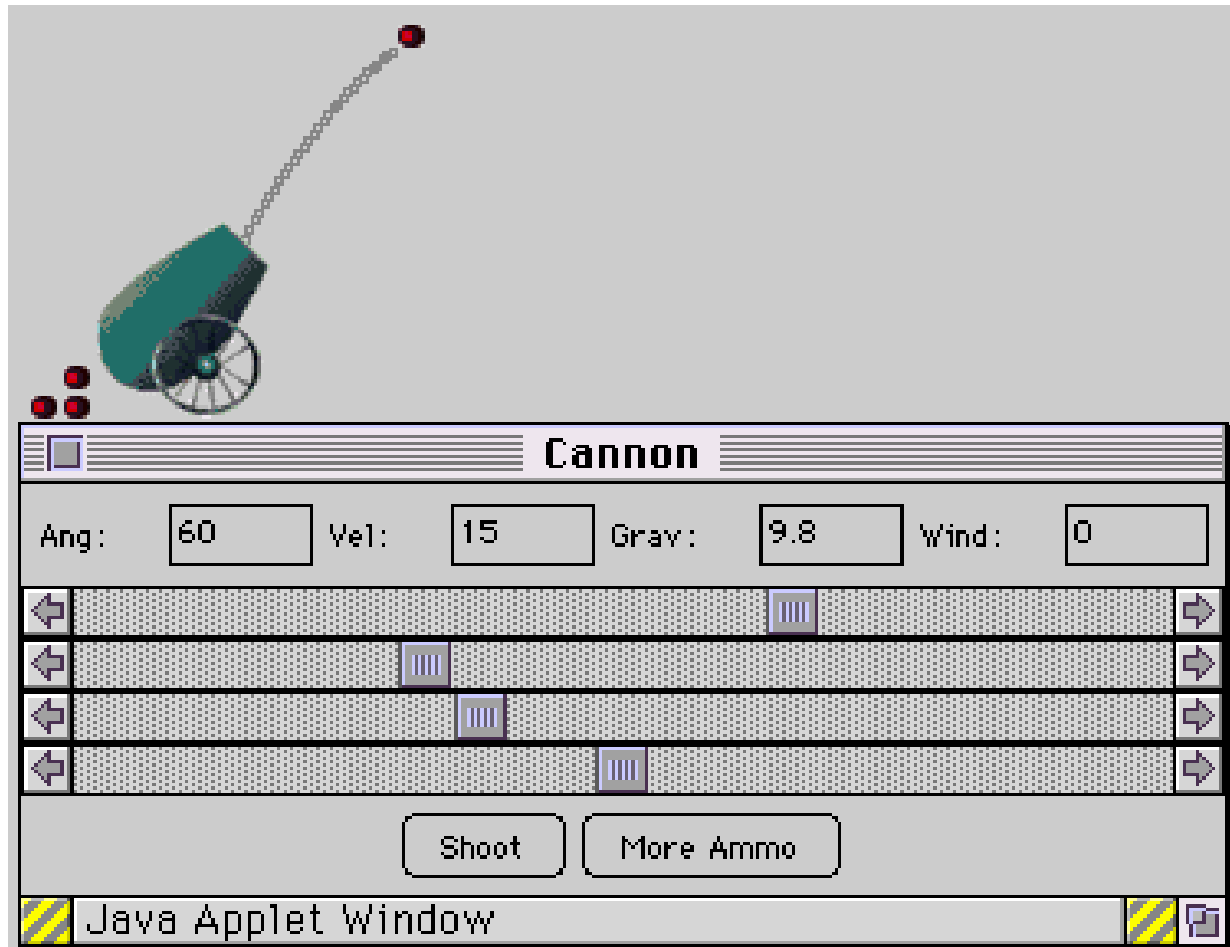
Interaktives, dynamisches Web



- Formulare
- Common Gateway Interface (CGI)
- JavaScript
- Active Server Pages (ASP, JSP)
- Java Applets

Klient	Server
Formulare	CGI
	ASP, JSP
Java Script	
Applets	

Anwendungsgebiete für Applets: Animationen



Anwendungsgebiete für Applets: Applikationen

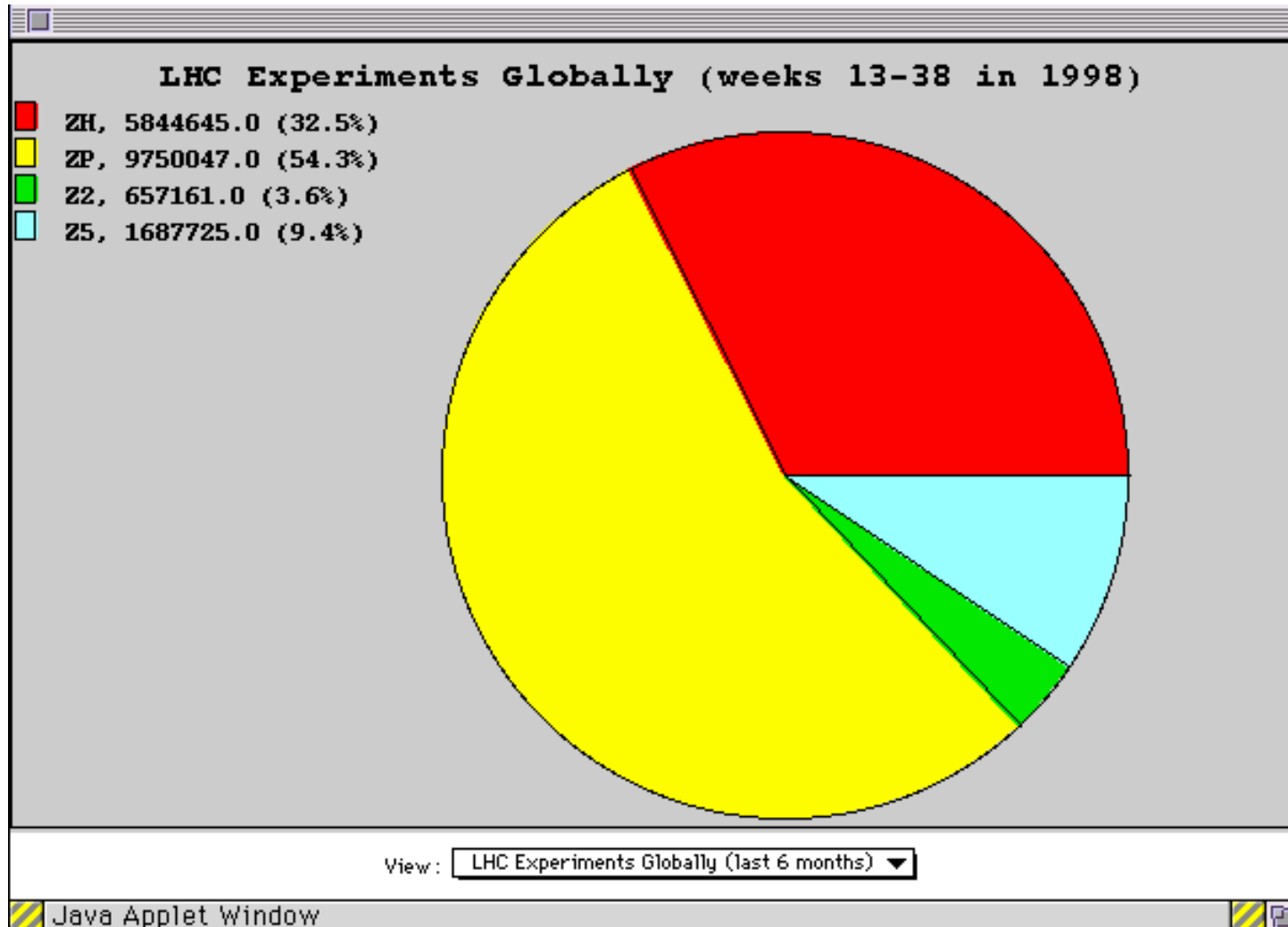


27 - across

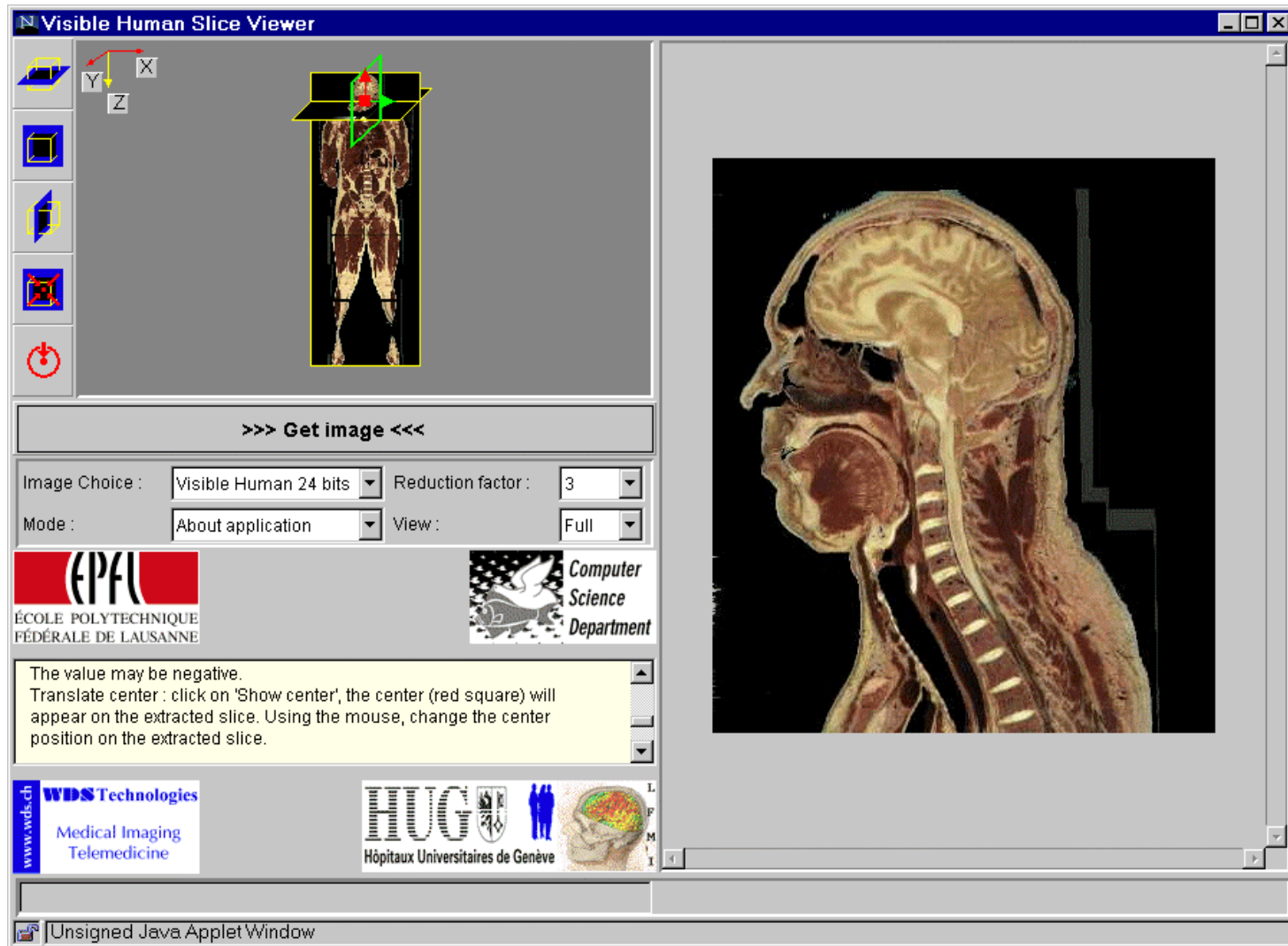
Group of two

1	2	3		4	5	6	7		8	9	10	
			12		13				14			15
				17					18			
	19			20		21		22		23		
		24						25		26		
				27		28	29		30		31	32
33	34	35	36			37		38		39		40
41						42				43		
44					45				46			
	47			48		49		50				
			51					52		53	54	55
56	57	58		59		60	61		62			
63			64			65		66		67		68
69						70				71		
	72					73					74	

Anwendungsgebiete für Applets: Datenpräsentation



Anwendungsgebiete für Applets: Visualisierungen



Anwendungsgebiete für Applets: Komplizierte Dateneingabe



Konten

VerfügerIn: Bert Tester

BLZ	Kontonummer	Kontobezeichnung	Kontostand per 23.09.1998
30025	1-23.456.783	Bert Maiers Gehaltskonto	12.530,54 ATS
30025	12-34.567.890	Bert Maiers Kegelclub	1.230,30 ATS
30025	123-45.678.901	Bert und Clara Maier (Familienkonto)	7.355,20 ATS

Wählen Sie bitte eines Ihrer Konten und die gewünschte Funktion aus

Raiffeisen. Die Bank 

Anwendungsgebiete für Applets: Verteilte Bearbeitung von Daten

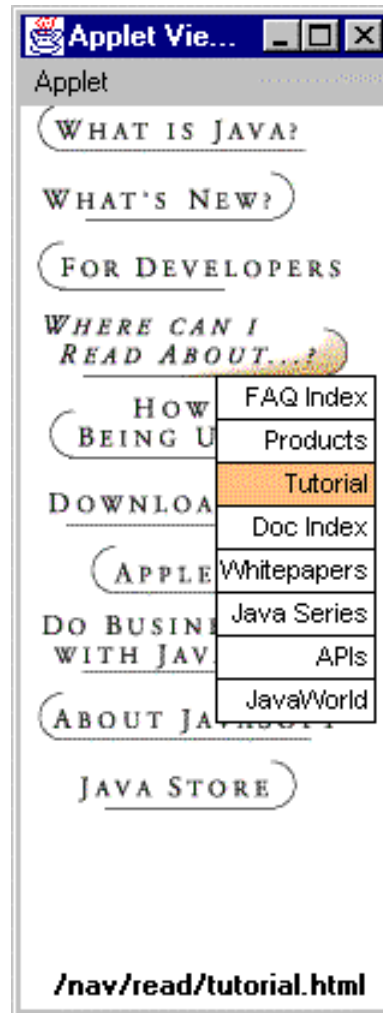


The screenshot shows a Java Applet Viewer window titled "Applet Viewer: JavaNotes.class". The window contains a menu bar with "File", "View", "Options", and "Help". Below the menu is a "User" field with a dropdown menu showing "< no login >". To the right of the user field are several buttons: "Today", "Go To", "New", "View", and "Delete".

The main content area is divided into two sections. On the left is a list of times from 1:00 to 7:30 in 30-minute increments. On the right is a calendar for September 1998. The calendar has columns for days of the week (Mo, Di, Mi, Do, Fr, Sa, So) and rows for weeks. The date 23 is highlighted. Below the calendar are navigation buttons for previous and next months, and the text "September 1998".

At the bottom of the window, a status bar displays the text "Applet started."

Anwendungsgebiete für Applets: Browsererweiterung



Aufbau von Applets



HTML-Seite

Umgebung (Text, Bilder, andere Applets)

Parameter

„Java“ - Teil

Eine oder mehrere Klassendateien

Ein Jar-Archiv

Zusätzliche Dateien (Bilder, Datenbank, ...)

Einbettung in HTML-Seite



...
<applet code="Mein.class" width=300 height=200>

<param name=MeinPara value="Veränderbar ohne
Programmänderung">

<param name=WeitererPara value="unterschiedlich auf
verschiedenen HTML-Seiten">

Diesen Text können Sie sehen, wenn Ihr Browser
keine Applets unterstützt!

</applet>

...

Attribute des Applet-Tag



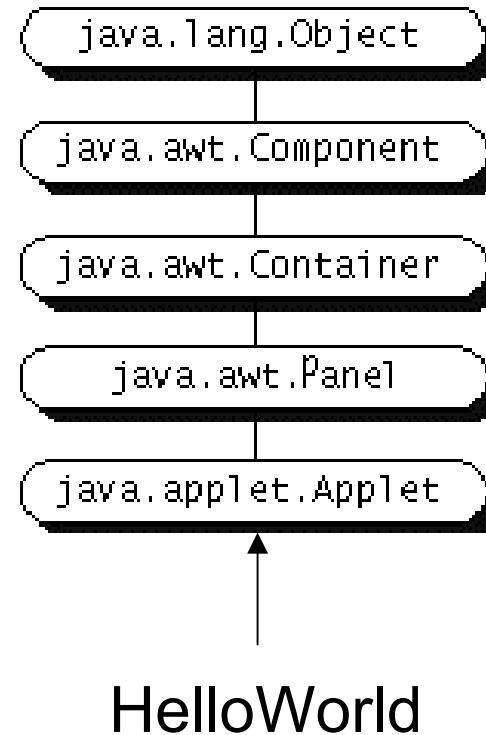
code	Name der auszuführenden Klasse
width	Breite des reservierten Bereichs
height	Höhe des reservierten Bereichs
name	Name des Applets (für Inter-Applet-Kommunikation)
align	Ausrichtung des reservierten Bereichs
codebase	URL des Verzeichnisses wo die Klassen gefunden werden

HelloWorld-Applet



```
import java.applet.*;
import java.awt.Graphics;

public class HelloWorld extends Applet {
    public void paint(Graphics g) {
        g.drawString("Hello world!", 50, 25);
    }
}
```



Wichtige Applet-Methoden



- **paint(Graphics g)**

Zeichnet das Applet und seine Elemente

- **init()**

Browser ruft *init()* des Applets zur Initialisierung auf

- **start()**

Browser ruft *start()* beim Start der HTML-Seite auf

- **stop()**

Browser ruft *stop()* beim Stop der HTML-Seite auf

- **destroy()**

SecurityManager



- Applets laufen in einem *AppletViewer*
- Jeder *AppletViewer* besitzt einen *SecurityManager*
- *SecurityManager* überwachen Applets und lösen gegebenenfalls eine *SecurityException* aus.
- Warum das alles?
 - Applet darf beim Klienten nichts zerstören
 - Und auch keine Informationen ausspionieren!

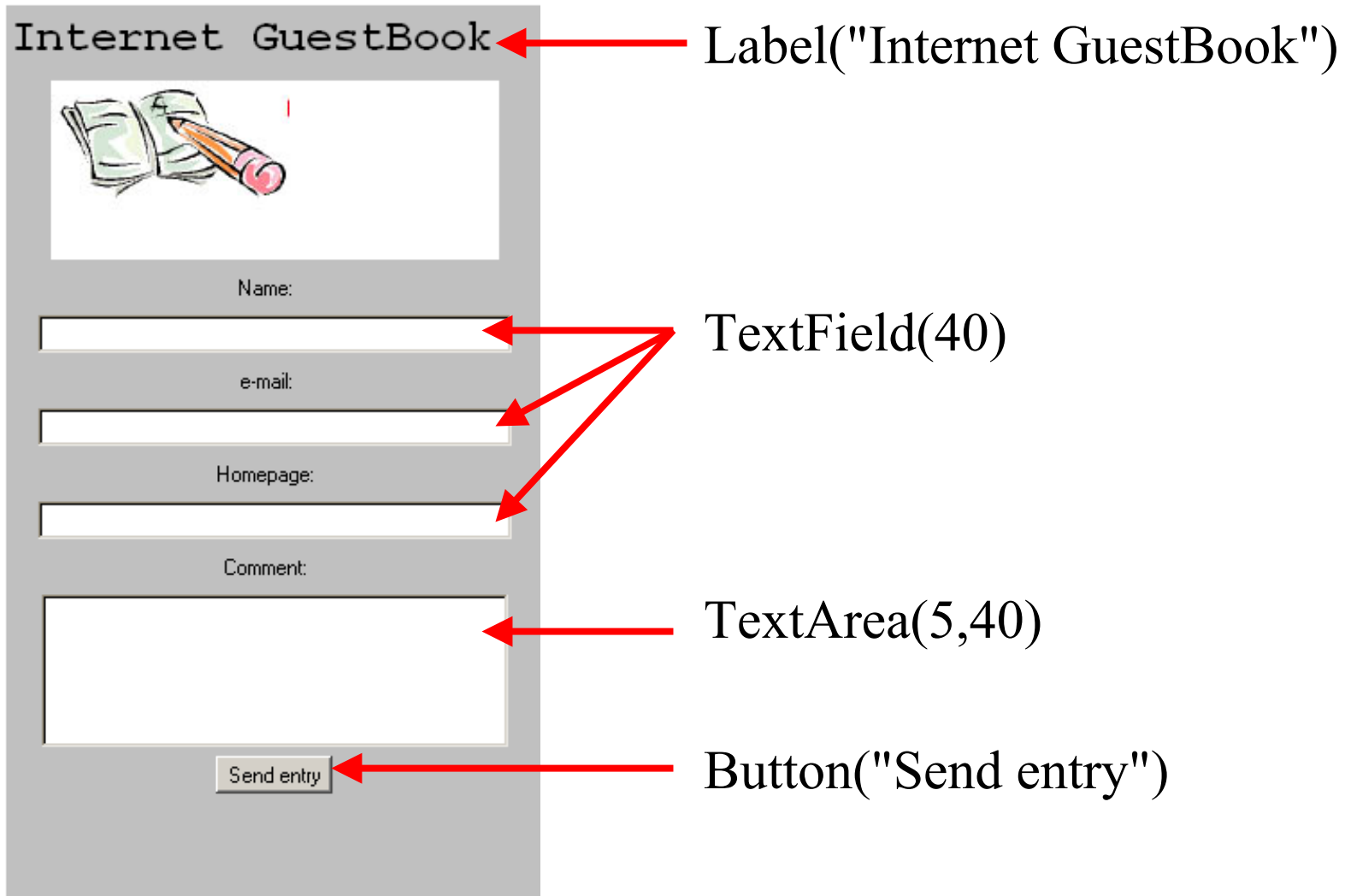
- Verboten:
 - Das Laden von zusätzlichen Bibliotheken und das Ausführen von *native* Methoden.
 - Dateioperationen auf dem Klientenrechner
 - Keine Netzverbindungen außer zum Host von dem das Applet stammt.
 - Das Starten von Programmen auf dem Klientenrechner
 - Das Lesen von Systemeinstellungen auf dem Klientenrechner
 - Fenster die ein Applet öffnet haben ein anderes Erscheinungsbild:





- Erlaubt:
 - Netzwerkverbindung zum Host von dem das Applet stammt (z.B.: *mySQL-DB* Anbindung)
 - Können andere HTML Seiten anzeigen.
 - Können *public* Methoden von anderen Applets der gleichen HTML-Seite aufrufen.
 - Können weiterlaufen auch wenn Seite verlassen wird. (meistens wird aber *stop()* aufgerufen)
 - Für Applets die aus dem lokalen *CLASSPATH* kommen gelten diese Restriktionen nicht.

"GuestBook"-Applet Layout



Applet Komponenten



- Initialisierung der graphischen Oberfläche (GUI)
- Ereignisverbindungen (z.b.: Button-Click)
- Kommunikation mit CGI Skript
 - Lesen und Schreiben der Gästebucheinträge
- Ausgabe der CGI Ergebnisse
 - Ausgabe der einzelnen Gästebucheinträge

Initialisieren der GUI



```
public void init(){
    Panel ppic;
    Image img;
    img = getImage( this.getDocumentBase(), "guestbook.jpg" );
    ppic = new Panel() {
        public void paint(Graphics g) {
            if (img != null) g.drawImage(img, 0 ,0, this);
        }
        public Dimension getPreferredSize(){
            return new Dimension(250,100);
        }
    };
    this.setLayout(new BorderLayout());
    Label l = new Label("Internet GuestBook");
```

Verbinden von GUI Komponenten



- Implementierung einer speziellen Schnittstelle:

```
class GuestBook extends Applet implements
ActionListener

public void actionPerformed( ActionEvent ev ){
    //Reaktion auf das Ereignis
}
```

- Registrierung bei einer GUI-Komponente:

```
Button b = new Button("Send");

b.addActionListener( this );
```

Kommunikation mit CGI Skript



```
private void loadEntries( ){
    String s = null;
    try {
        URL server = new URL( "http://127.0.0.1:8080/examples/servlet/StorageServlet" );
        InputStream in = server.openStream();
        byte[] buf = new byte[3000];
        int i = 0;
        int ch = in.read();
        while (ch >= 0) { buf[ i++ ] = ( byte )ch; ch = in.read(); }
        s = new String( buf, 0, i );
    } catch ( Exception ex ){ ex.printStackTrace(); }
    StringTokenizer tokenizer = new StringTokenizer( s, "|" );
    int count = tokenizer.countTokens();
    while( count > 4 ){
        String name = tokenizer.nextToken();
        String datum = tokenizer.nextToken();
        String email = tokenizer.nextToken();
        String hp = tokenizer.nextToken();
        String text = tokenizer.nextToken();
        Msg m = new Msg( name, email, hp, text);
        entries.addElement( m );
        count -= 5;
    }
}
```

Anzeigen des Bildes



```
public boolean imageUpdate(  
    Image img, int flags,  
    int x, int y, int w, int h) {  
    boolean ready = ((flags & (ERROR | ALLBITS)) != 0);  
    if (ready)  
        imgCanvas.repaint();  
    return !ready;  
}
```


Applets können alles

Aber

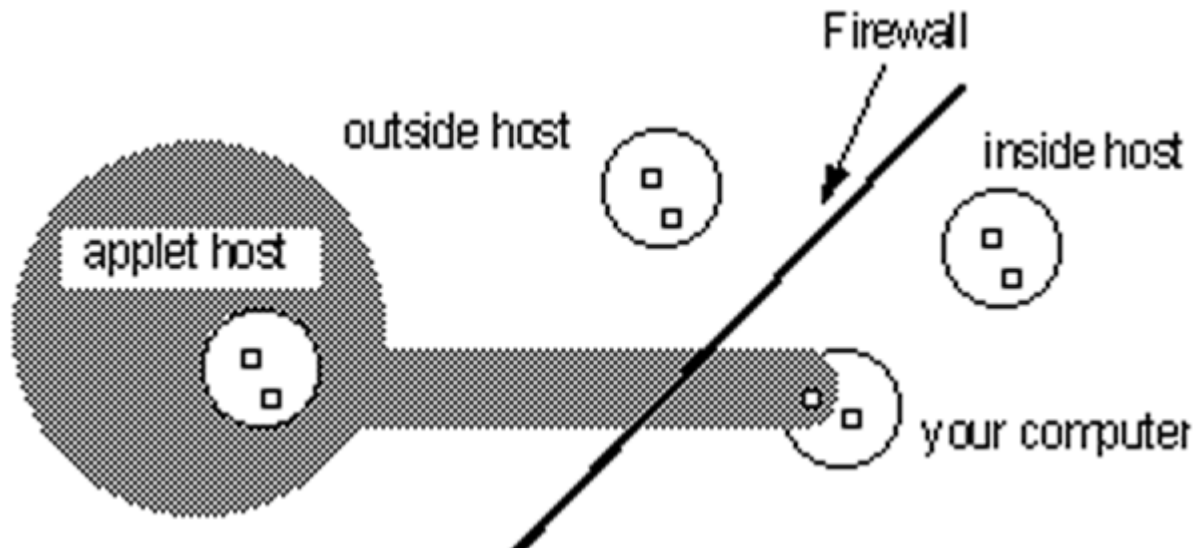
Sicherheit

Komplexität

Applet Security Manager



- Keine lokalen Dateioperationen
- Keine Emails
- Starten von Prozessen
- Eingeschränktes Netzwerk



Komplexität



- Programmieren
- Viele Details
 - Layout
 - Asynchrone Datenübertragung