



# (J)Meterweise integrieren

Integrationstests mit  
JMeter und Eclipse

# Agenda

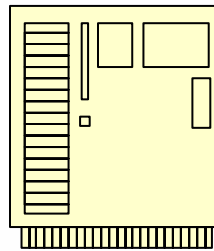
- Ausgangssituation
- Weshalb JMeter?
- JMeter Übersicht
- JMeter – Deep inside
- Erweitern von JMeter in Eclipse
- JMeter im Integrationstest
- Fazit / Zusammenfassung
- Weiterführende Links und Quellen

# Ausgangssituation



# Ausgangssituation

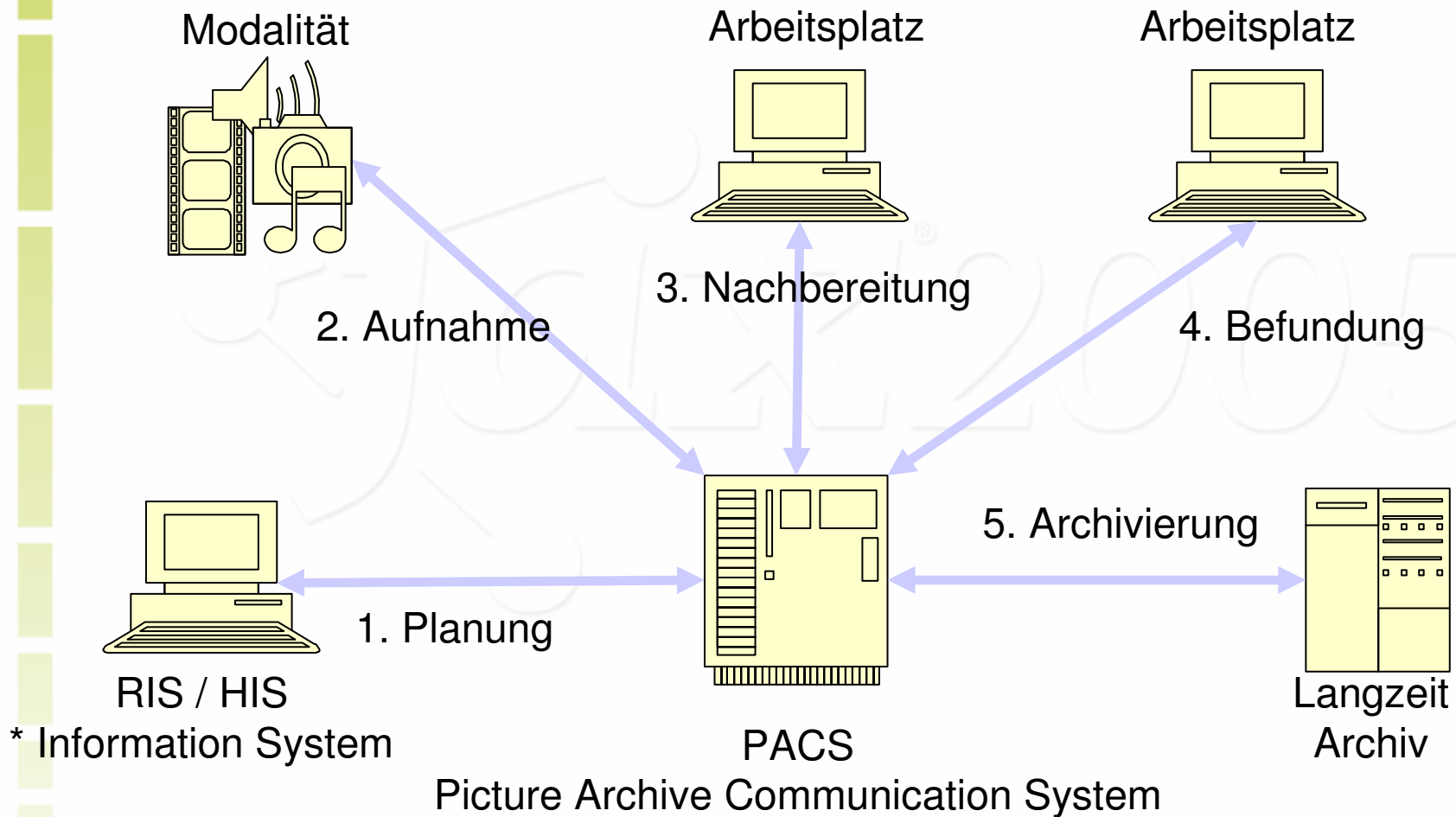
- Integration ist Vision in der Medizintechnik
- PACS stehen im Zentrum dieser Vision



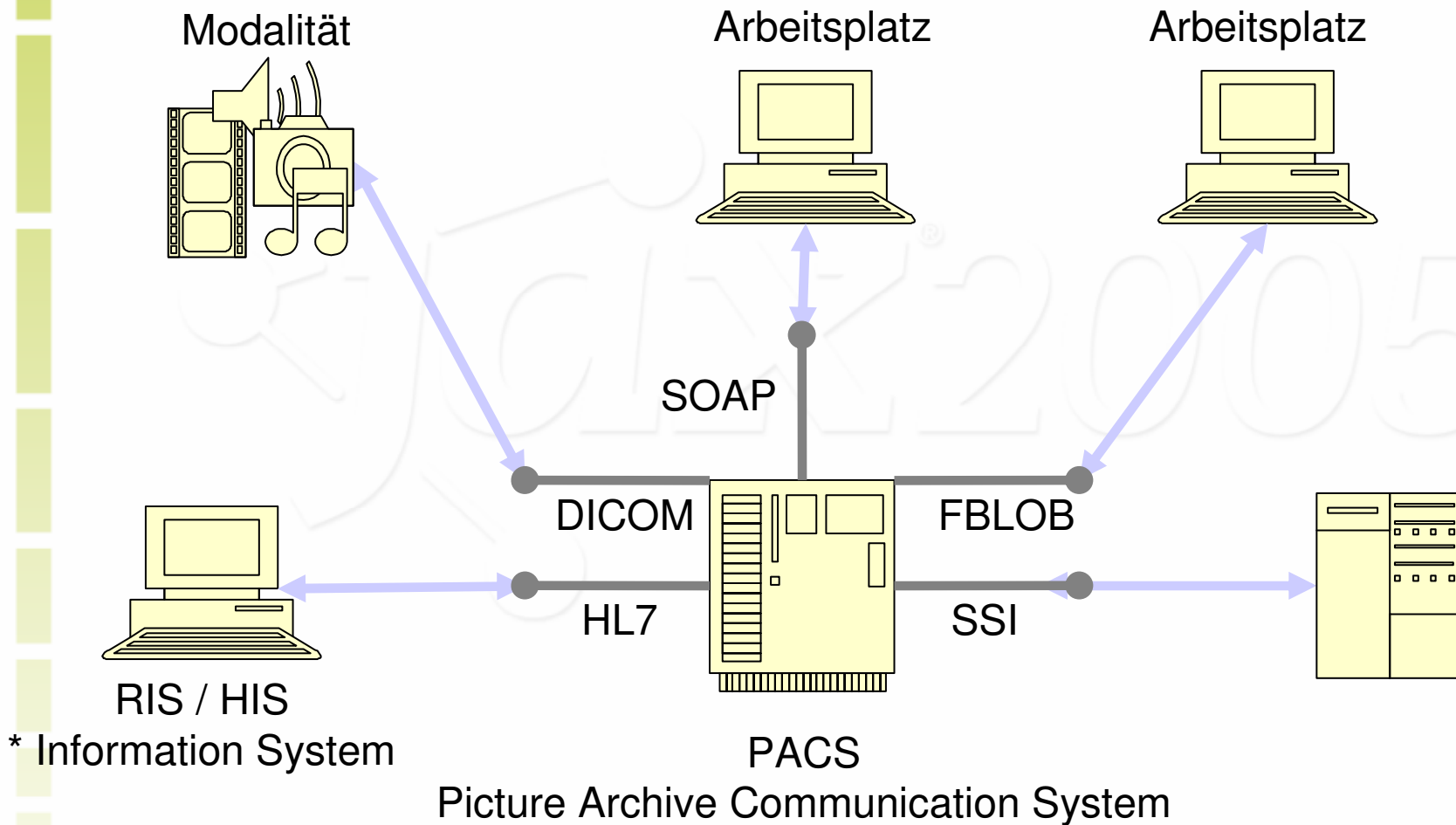
PACS

Picture Archive Communication System

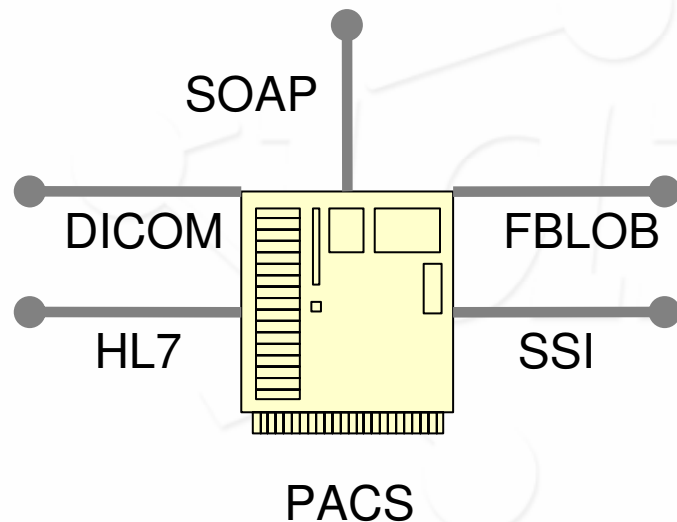
# Ausgangssituation



# Ausgangssituation



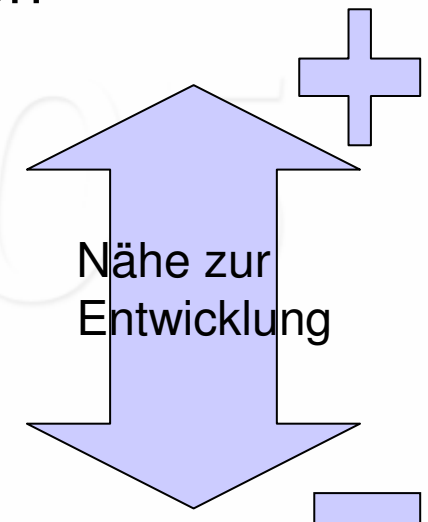
# Ausgangssituation



- Viele Schnittstellen
- Testtreiber für einige Schnittstellen vorhanden
  - Dicom Offis
  - Dicom ADVT
  - MESA
  - FBLOB
- Schnittstellen für übrige HL7 und SOAP TCP- und Socket basiert

# Ausgangssituation

- Medizintechnik unterliegt strengen QM-Auflagen (FDA)
- Viele Testebenen, um Qualität sicherzustellen
  - Unit Tests in der Entwicklung (UT)
  - Produktintegrationstests (PIT)
  - Produktvalidierungstests (PVT)
  - Systemtest (ST)
  - Systemintegrationstest (SIT)
  - CAT (Klinischer Anwendungstest)





# Ausgangssituation

## Produktintegrationstests (PIT)

- Entwicklungsnah
- Fokus auf Prüfung der Schnittstellen gegenüber ihrer funktionalen Spezifikation

## Spezialitäten des PACS PIT

- Keine Oberflächentests
- Größte Anzahl von Schnittstellen
- Interesse an Performance-/Stabilitätsaussagen
- Verwendung der Tests im Feld
- Verfügbarer WinRunner sehr teuer, aber wenig hilfreich

# Weshalb JMeter?



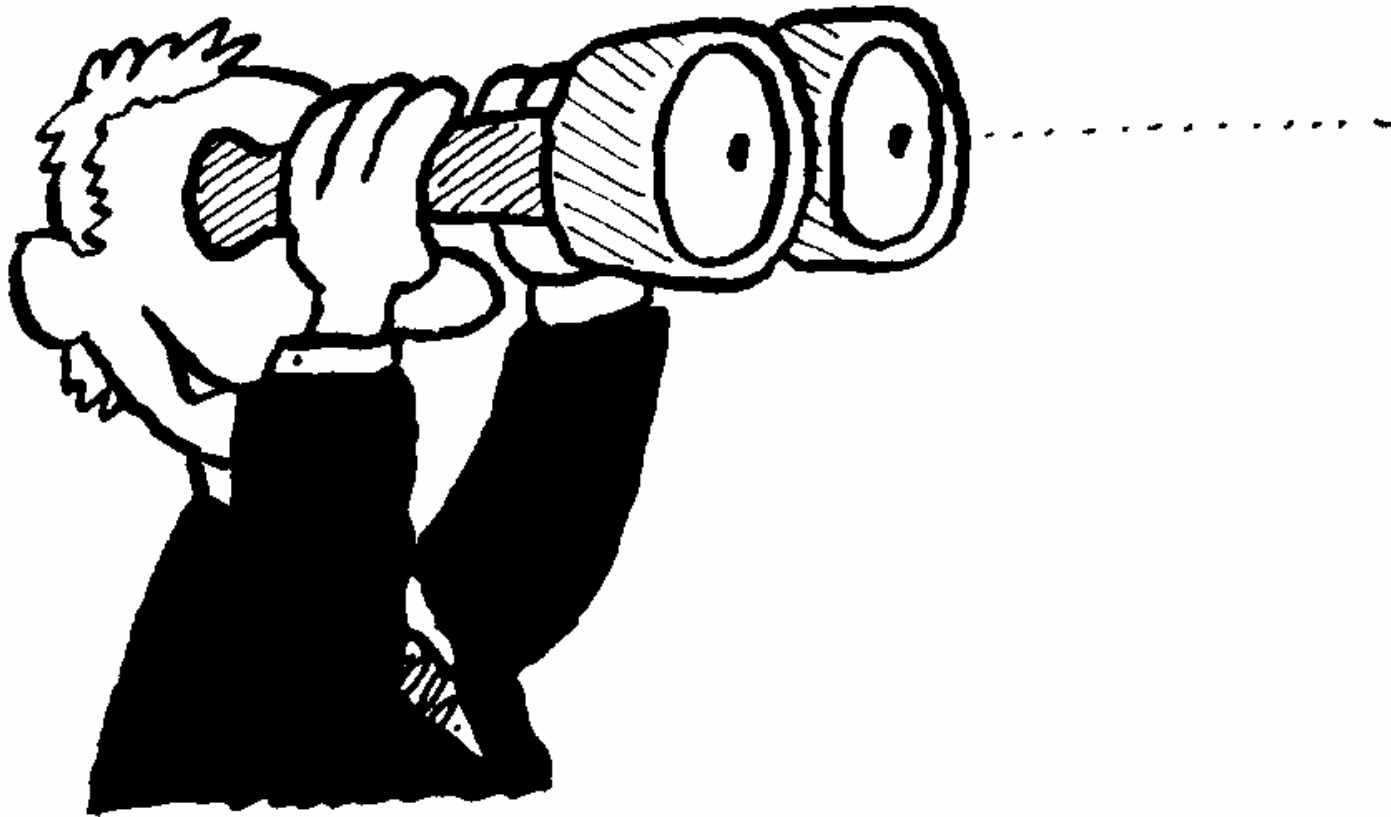
# Weshalb JMeter? (I)

- 100 % Java
  - Plattformunabhängigkeit, Multiplattform-Unterstützung
  - Keine Notwendigkeit für proprietäre Skriptsprache
  - Gleiche Entwicklungssprache wie für zu testendes PACS
  - Umfangreiche Bibliothekenunterstützung (<-> WinRunner TSL)
- OpenSource
  - Ermöglicht Verwendung zur Problemdiagnose im Feld
  - Geringe Kosten im Vergleich zu Mercury WinRunner

# Weshalb JMeter? (II)

- Ermöglicht kurze Turnarounds
  - mit Entwicklern
  - Für geänderte Tests
- Großes Repertoire bereits existierender Treiber
  - SOAP, HTTP, LDAP, ...
- Gute Erweiterungsmöglichkeiten durch
  - „Plugin“-Technik
  - Einbindung von externen Treibern
- Ausbaufähigkeit zu Last- und Stresstestszenarien

# JMeter Übersicht



# JMeter Übersicht

- Historie – aktuelle Entwicklung
- GUI - Oberfläche
- Zentrale Elemente
  - ThreadGruppe
  - Sampler
  - Controller
  - Listener
  - ...
- Gültigkeitsbereiche der Elemente
- Weiteres Wissenswertes

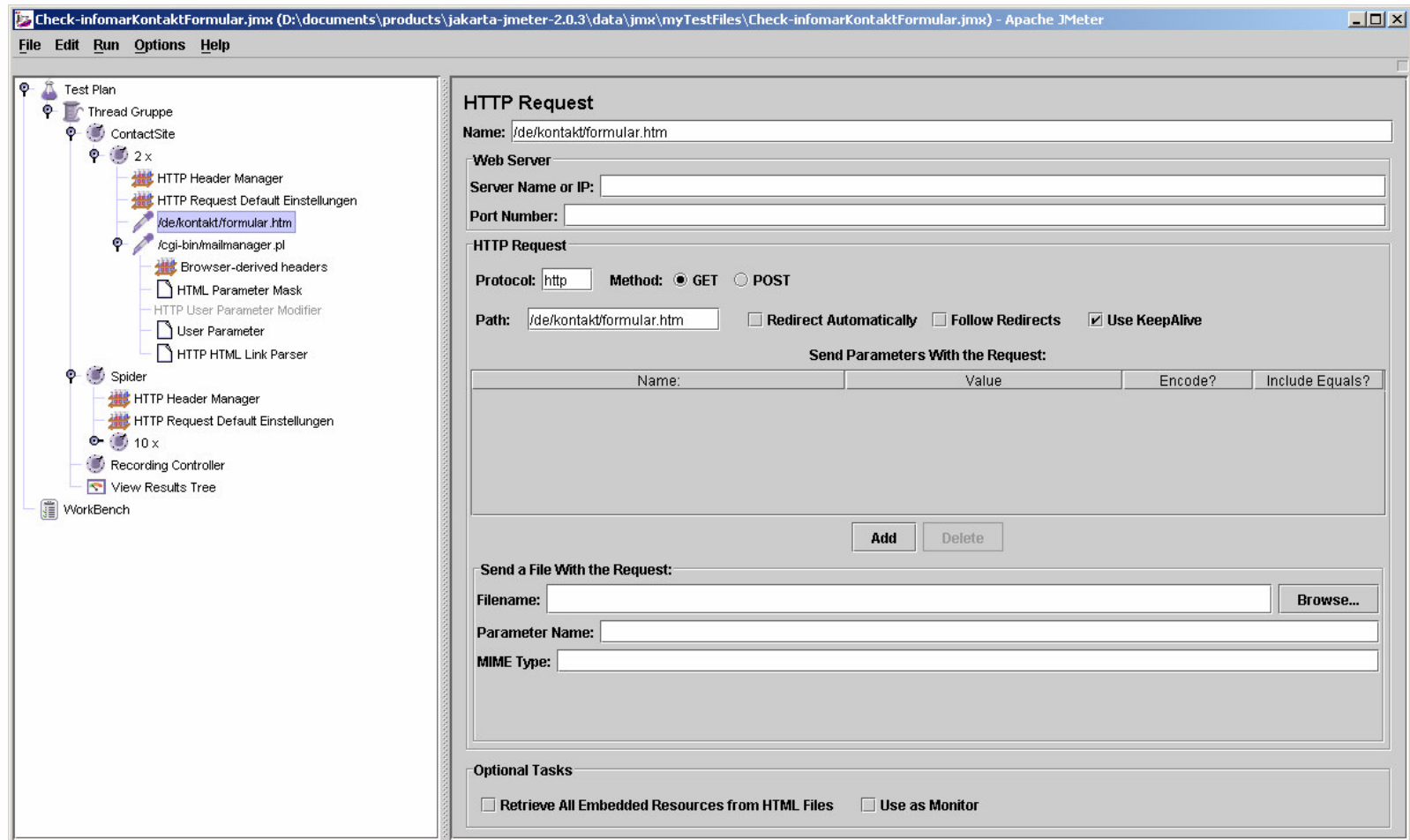
# JMeter Übersicht

## Gestern und heute

- Historie
  - Version 1.0 von Stefano Mazzocchi (02/99)
  - Aktuelle Version 2.0.3 von Michael Stover (03/05)
  - Release in halbjährigem Abstand
- Ziele / Zweck
  - Ursprünglich: Test Performance von Apache JServ
  - Simulation von Lastszenarien
  - Messung von Performance
- Open Source
  - unter <http://jakarta.apache.org/jmeter/>
  - 729 Klassen, ~ 51 Kilo NCSS

# JMeter Übersicht

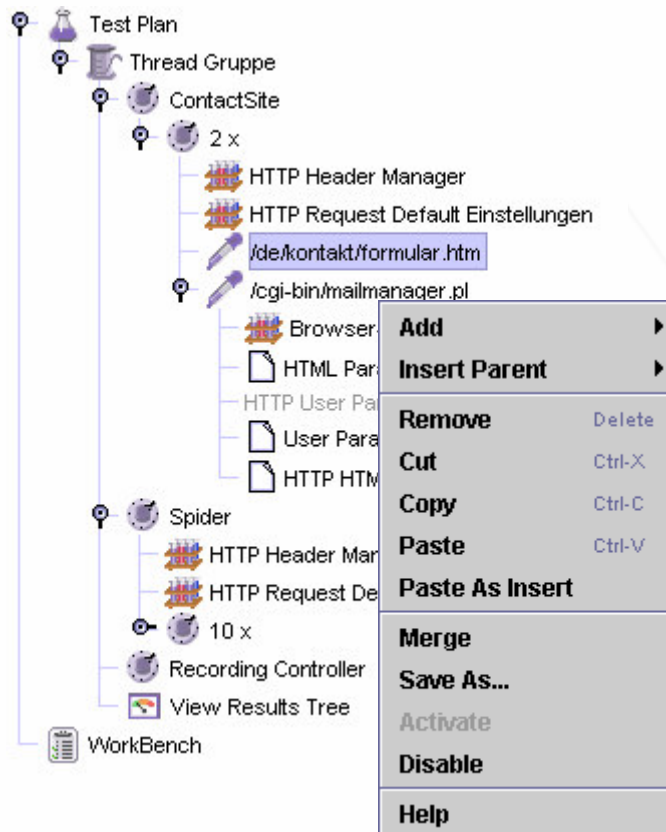
## GUI – antik, aber hilfreich





# JMeter Übersicht

## GUI – antik, aber hilfreich



- Linke Seite
  - Testplan Baumdarstellung
  - Cut - Copy - Paste
  - Kontextsensitives Menu
  - Merge von Testplänen
  - Speicherung von Teilen
  - Immer nur ein Testplan

# JMeter Übersicht

## GUI – antik, aber hilfreich

- Rechte Seite – Detailansicht zur
  - Spezifikation von Testelementen
  - Anzeige von Testergebnissen

**HTTP Request**

Name: /de/kontakt/formular.htm

**Web Server**

Server Name or IP:

Port Number:

**HTTP Request**

Protocol: http Method:  GET  POST

Path: /de/kontakt/formular.htm  Redirect Automatically  Follow Redirects  Use KeepAlive

Send Parameters With the Request:

Name	Value	Encode?	Include Equals?

Add Delete

Send a File With the Request:

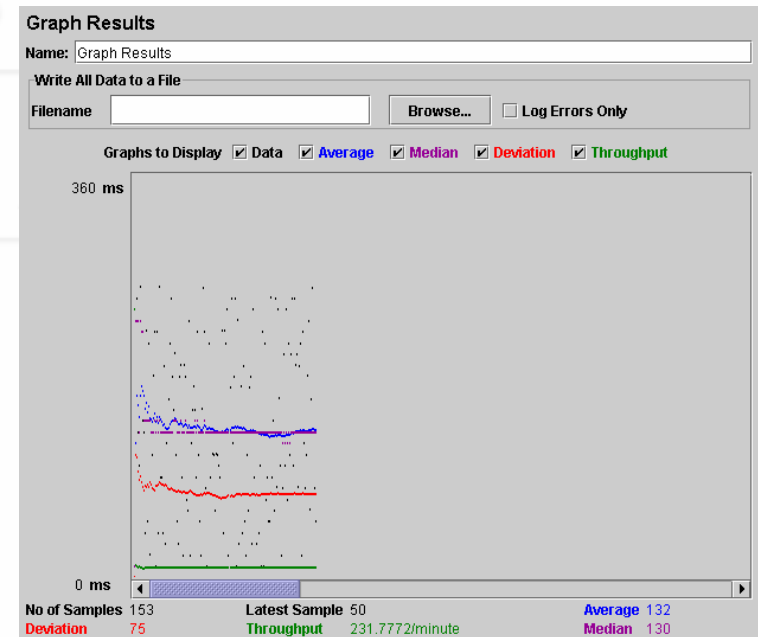
Filename:  Browse...

Parameter Name:

MIME Type:

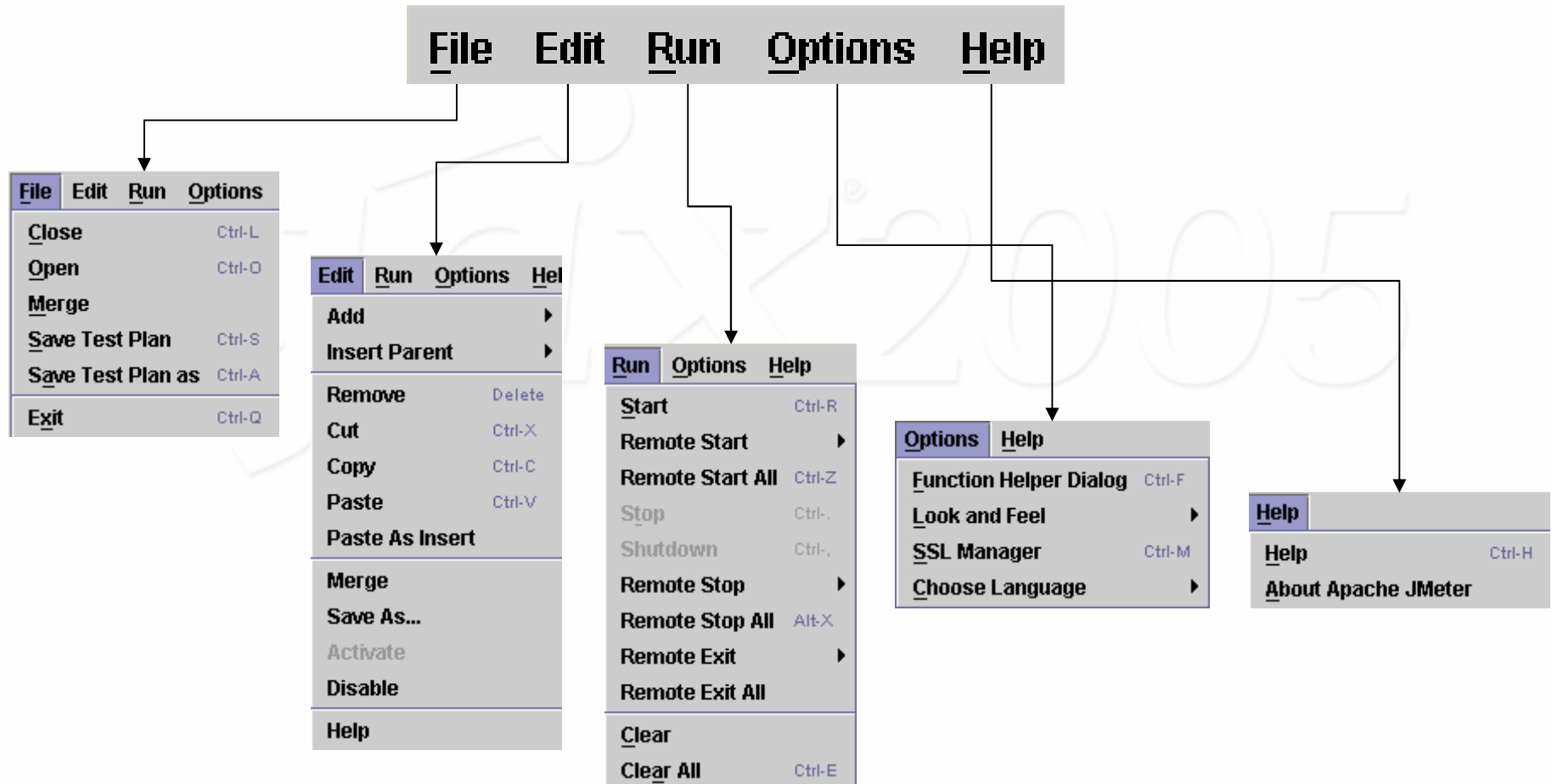
**Optional Tasks**

Retrieve All Embedded Resources from HTML Files  Use as Monitor



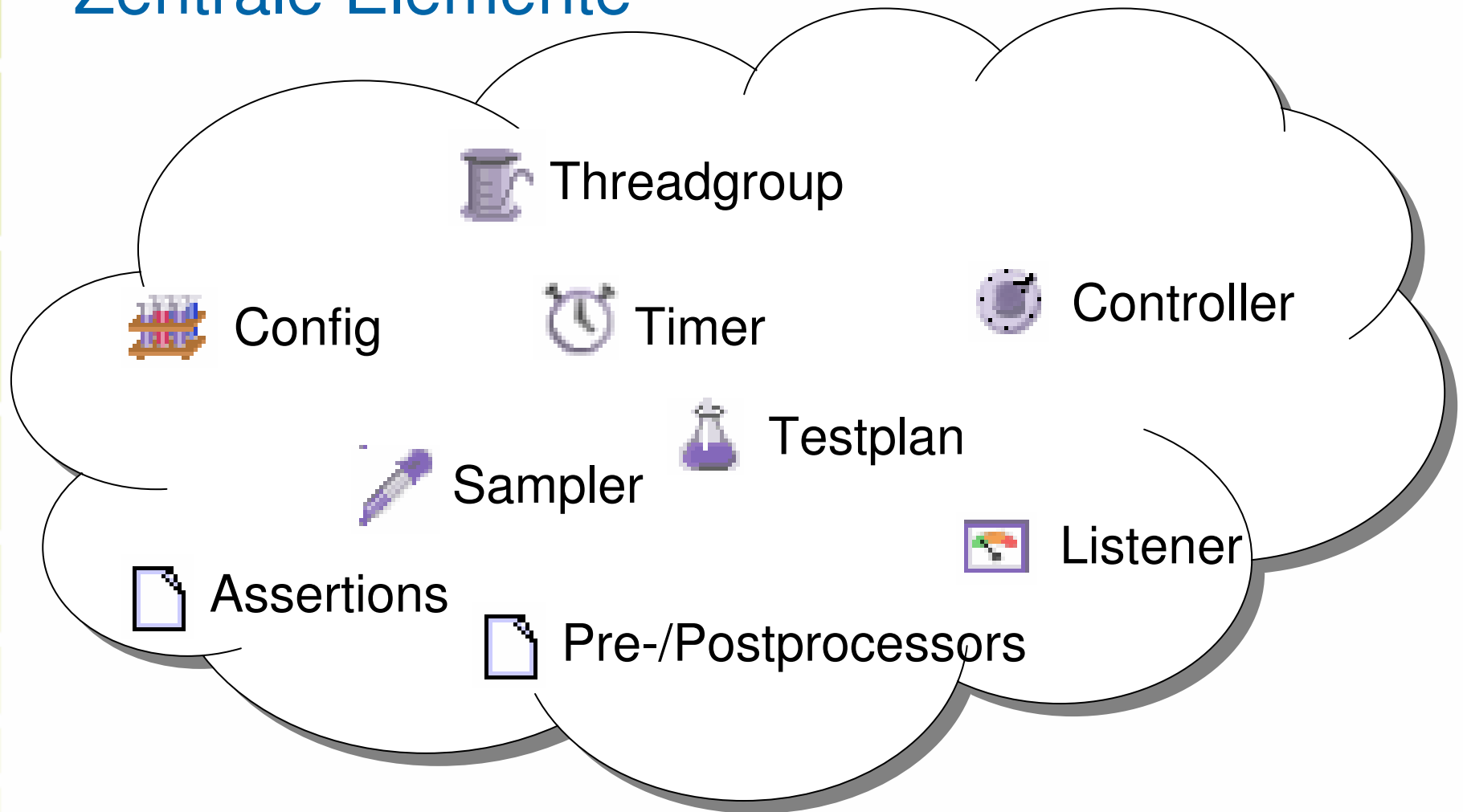
# JMeter Übersicht

## GUI – antik, aber hilfreich



# JMeter Übersicht

## Zentrale Elemente



# JMeter Übersicht

## Zentrale Elemente - Threadgroup

- Steuerung des Rampups
- Parallelisierung des Testablaufs
- Wiederholanzahl
- Scheduling
- Fehlerverhalten

**Thread Group**

Name:

Action to be taken after a Sampler error

Continue  Stop Thread  Stop Test

**Thread Properties**

Number of Threads:

Ramp-Up Period (in seconds):

Loop Count:  Forever

Scheduler

**Scheduler Configuration**

Start Time:

End Time:

Duration (seconds):

Startup delay (seconds):

# JMeter Übersicht

## Zentrale Elemente - Sampler



- Ausführung einer Anfrage
  - **HTTP**
  - SOAP, Webservice
  - LDAP
  - JDBC
  - Java Requests (Test!)
  - TCP
  - Beanshell, BSF
  - (Tomcat Access Log)

The screenshot shows the 'HTTP Request' configuration dialog in JMeter. It includes fields for Name, Web Server (Server Name or IP, Port Number), HTTP Request (Protocol, Method: GET or POST, Path), and options for Redirect Automatically, Follow Redirects, and Use KeepAlive. There is a table for 'Send Parameters With the Request' with columns for Name, Value, Encode?, and Include Equals?. Below this are buttons for 'Add' and 'Delete'. There are also fields for 'Send a File With the Request' (Filename, Parameter Name, MIME Type) and a 'Browse...' button. At the bottom, there are 'Optional Tasks' with checkboxes for 'Retrieve All Embedded Resources from HTML Files' and 'Use as Monitor'.

# JMeter Übersicht

## Zentrale Elemente - Controller



- Ablauf und Durchsatzsteuerung
  - Ablauf (wie oft, wann, etc.)
  - Durchsatz
  - Zeitlich
- Strukturierung (Simple, Include)
- Parametrierung (Zufälligkeit)
- Unterstützte Typen
  - If, While, **For Each**, Switch
  - **Throughput**,
  - Module, Simple
  - Random, Random Order
  - Interleave, OnlyOnce

The screenshot shows the configuration for a 'ForEach Controller'. The fields are as follows:

- Name:** ForEach Controller
- Input variable prefix:** input
- Output variable name:** forEachInput
- Add "\_" before number ?

# JMeter Übersicht

## Zentrale Elemente - Config



- Parametrierung mehrerer Sampler gleichen Typs
  - **HTTP**
  - JDBC
  - LDAP
  - TCP
  - Java
- Definition von Variablen
- Manager
  - HTTP Authorization (Simple)
  - HTTP Headers
  - HTTP Cookies

The screenshot shows the 'HTTP Request Defaults' configuration dialog. It has several input fields: 'Name' (set to 'HTTP Request Defaults'), 'Protocol', 'Server Name or IP', 'Path', and 'Port Number'. Below these fields is a section titled 'Send Parameters With the Request:' which contains a table with columns for 'Name', 'Value', 'Encode?', and 'Include Equals?'. At the bottom of the dialog are 'Add' and 'Delete' buttons.

Name	Value	Encode?	Include Equals?
------	-------	---------	-----------------

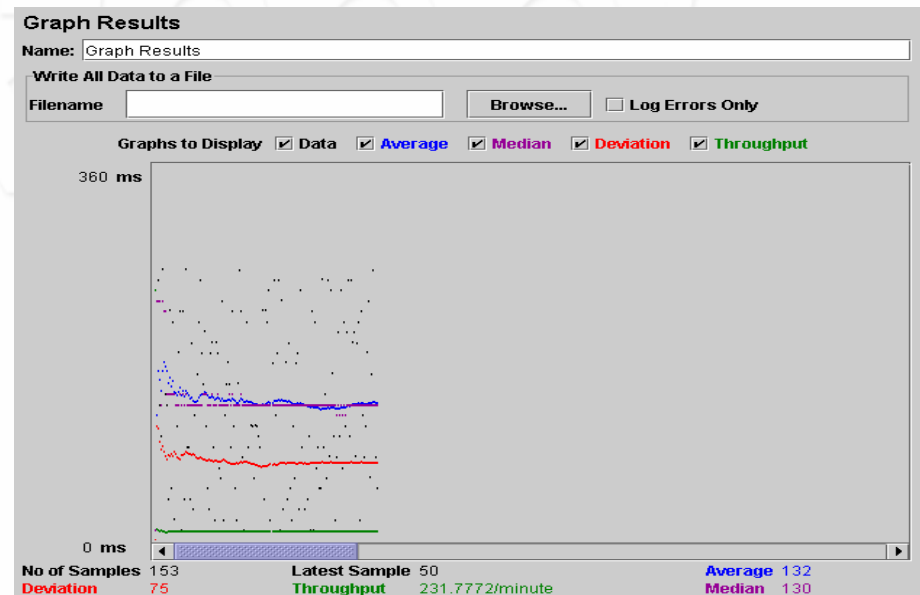


# JMeter Übersicht

## Zentrale Elemente - Listener



- Zusammenfassung von Ergebnissen in
  - tabellarischen Ansichten
  - grafischen Ansichten
  - Dateien
- Monitoring Tomcat (ab 5.0.19)
- Email-Benachrichtigung
- Unterstützte Typen
  - Aggregate Report
  - View Results in Table
  - Graph Results
  - View Results Tree
  - Simple Data Writer
  - Mailer
  - Assertion Results



# JMeter Übersicht

## Zentrale Elemente - Timer



- Steuerung
  - Durchsatz
  - Zeitlich
- Parametrierung (Zufälligkeit)
- Unterstützte Typen
  - Constant Timer
  - Uniform Random Timer
  - Gaussian Random Timer
  - ***Constant Throughput Timer***

Constant Throughput Timer

Name:

Delay before each affected sampler:

Target throughput (in samples per minute):

# JMeter Übersicht

## Zentrale Elemente - Assertions



- Überprüfung der durchgeführten Sampler auf
  - Inhalt
  - Dauer
  - Größe
  - „Wohlgeformtheit“
  - Identität
- Unterstützte Typen
  - Response
  - Beanshell (größte Freiheit ;-)
  - Duration
  - MD5Hex
  - HTML Tidy Prüfung
  - Size
  - XML, **XPath**

The screenshot shows the 'XPath Assertion' configuration window in JMeter. It includes a 'Name' field with the value 'XPath Assertion'. Below it, the 'XML Parsing Options' section contains four checkboxes: 'Tolerant XML/HTML Parser', 'Use Namespaces', 'Validate XML', and 'Ignore Whitespace'. The 'XPath Assertion' section features a text input field containing the XPath expression '//ok[3]' and a 'Validate' button. At the bottom of this section, there is a checkbox labeled 'True if nothing matches'.

# JMeter Übersicht

## Zentrale Elemente - Preprocessors



- Benutzerabhängige
  - Parameterwerte
  - Zähler
- Parametrierung von Requests bzgl.
  - Linkadressen
  - Werten von Eingabemasken
  - Session Ids
- Unterstützte Typen
  - User Parameters
    - **Interaktiv**
    - Aus XML Datei
  - HTML Link Parser
  - HTML Parameter Mask
  - HTTP URL Rewriting Modifier

**User Parameters**

Name:

Update Once Per Iteration

Parameters

Name:	User_1	User_2	User_3
username	Martin	Christine	Andreas
visitedPages	5	15	25

# JMeter Übersicht

## Zentrale Elemente - Postprocessors



- Auswertung und Speicherung von Ergebnissen
- Steuerung der Testfortführung
- Unterstützte Typen
  - **Regular Expression Extractor**
  - Result Status Action Handler
  - Save Responses to file
  - Generate Summary Results

The screenshot shows the configuration window for a 'Regular Expression Extractor' in JMeter. The window has a title bar 'Regular Expression Extractor'. Below the title bar, there is a text field for 'Name' containing 'Regular Expression Extractor'. Underneath is a section 'Response Field to check' with two radio buttons: 'Body' (selected) and 'Headers'. Below this are five input fields: 'Reference Name' with 'whileCondition', 'Regular Expression' with 'Failed', 'Template' (empty), 'Match No. (0 for Random)' (empty), and 'Default Value' with 'false'.

# JMeter Übersicht

## Zentrale Elemente - Sonstiges



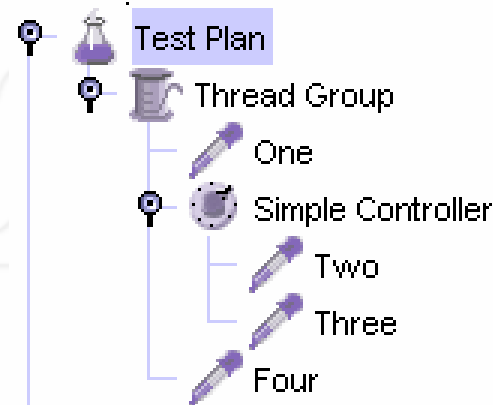
- **HTTP Proxy Server – Recording**
- Workbench – JMeter „Scrapbook“

The screenshot shows the 'HTTP Proxy Server' configuration window in JMeter. The 'Name' field is set to 'HTTP Proxy Server'. The 'Port' is '8080'. There are three checked options: 'Capture HTTP Headers', 'Set Keep-Alive', and 'Add Assertions'. The 'Target Controller' is set to 'Use Recording Controller'. The 'Grouping' is set to 'Do not group samplers'. There are two sections for 'Patterns to Include' and 'Patterns to Exclude', each with an 'Add' and 'Delete' button. At the bottom, there are 'Start', 'Stop', and 'Restart' buttons.

# JMeter Übersicht

## Gültigkeitsbereich der Elemente

- Folgende Elemente betreffen Vaterknoten und all seine Kinder
  - Config Elements
  - Pre-/Postprocessors
  - Assertions
  - Timers
  - Listeners

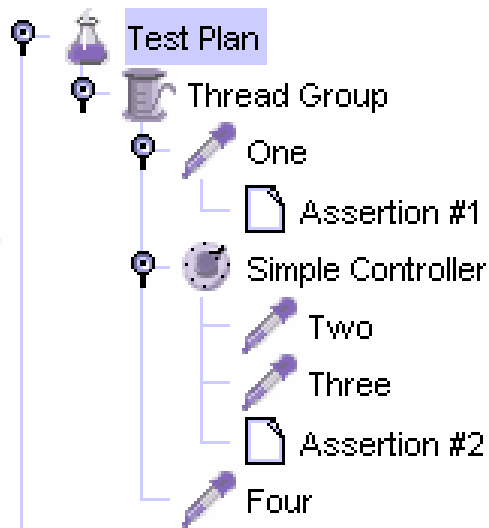


Sampler One  
Sampler Two  
Sampler Three  
Sampler Four

# JMeter Übersicht

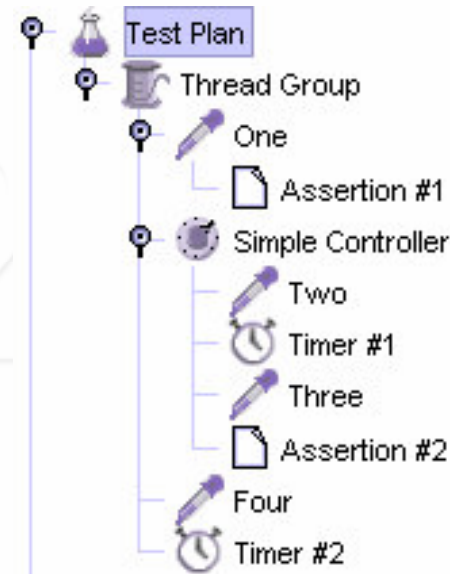
## Gültigkeitsbereich der Elemente

### ● Beispiel 1



One mit A #1  
Two mit A #2  
Three mit A #2  
Four

### ● Beispiel 2



One mit A #1 + T #2  
Two mit A #2 + T #1 + T #2  
Three mit A #2 + T #1 + T #2  
Four mit T #2



# JMeter Übersicht

## Weiteres Wissenswertes

- Dateiformat
  - XML (mit Suffix jmx)
  - Elementtypen node, element, collection, property
- Ausführungsmöglichkeiten
  - Interaktiv
  - BatchMode
  - Verteilt über mehrere Rechner
  - Ant TaskDef
  - Junit Adapter (noch in Arbeit :-)
  - Verwaltung mehrerer Testpläne in Eclipse

# JMeter – Deep inside

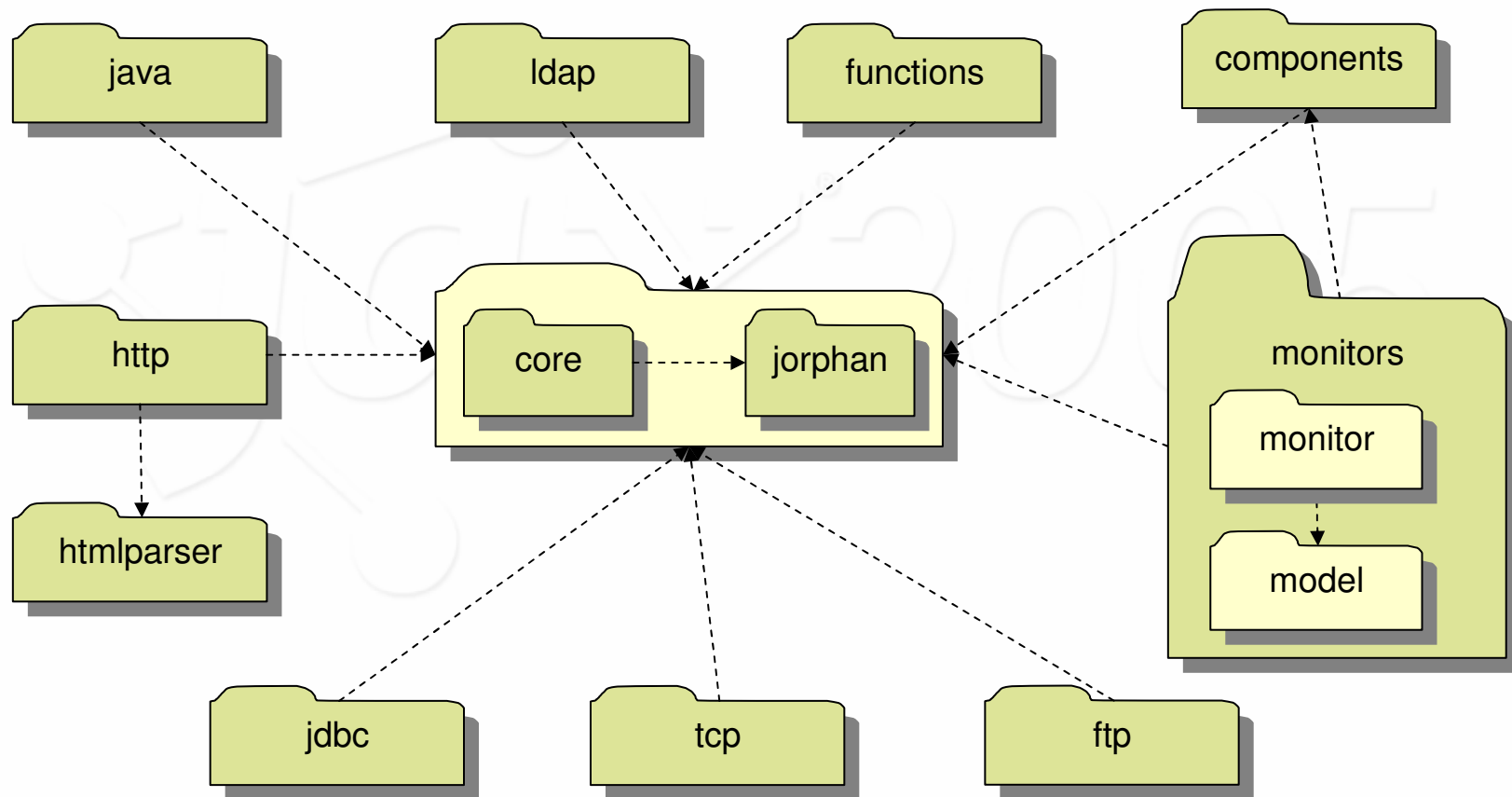


# JMeter – Deep inside

- Package Struktur
- Wichtige Interfaces
- Zusammenspiel GUI und TestElements

JAX2005

# JMeter – Deep inside Package Struktur



# JMeter – Deep inside Zentrale Interfaces

- GUI

```
JMeterGUIComponent  
TestElement createTestElement()  
void modifyTestElement(TestElement)  
void configure(TestElement);
```

- Model

```
TestElement  
...
```

```
Preprocessor  
void process()
```

```
Postprocessor  
void process()
```

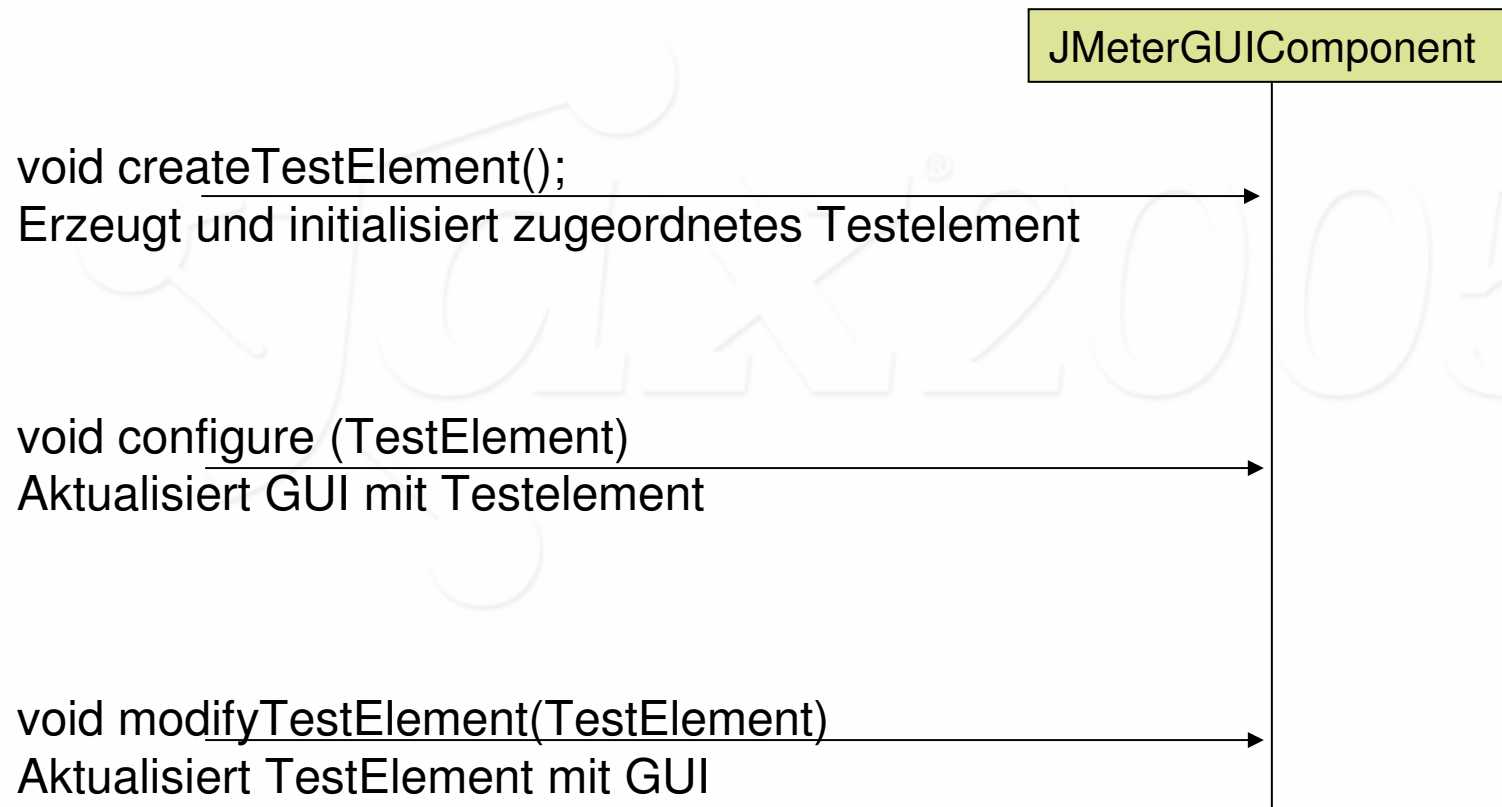
```
Sampler  
SampleResult sample(...)
```

```
Timer  
long delay()
```

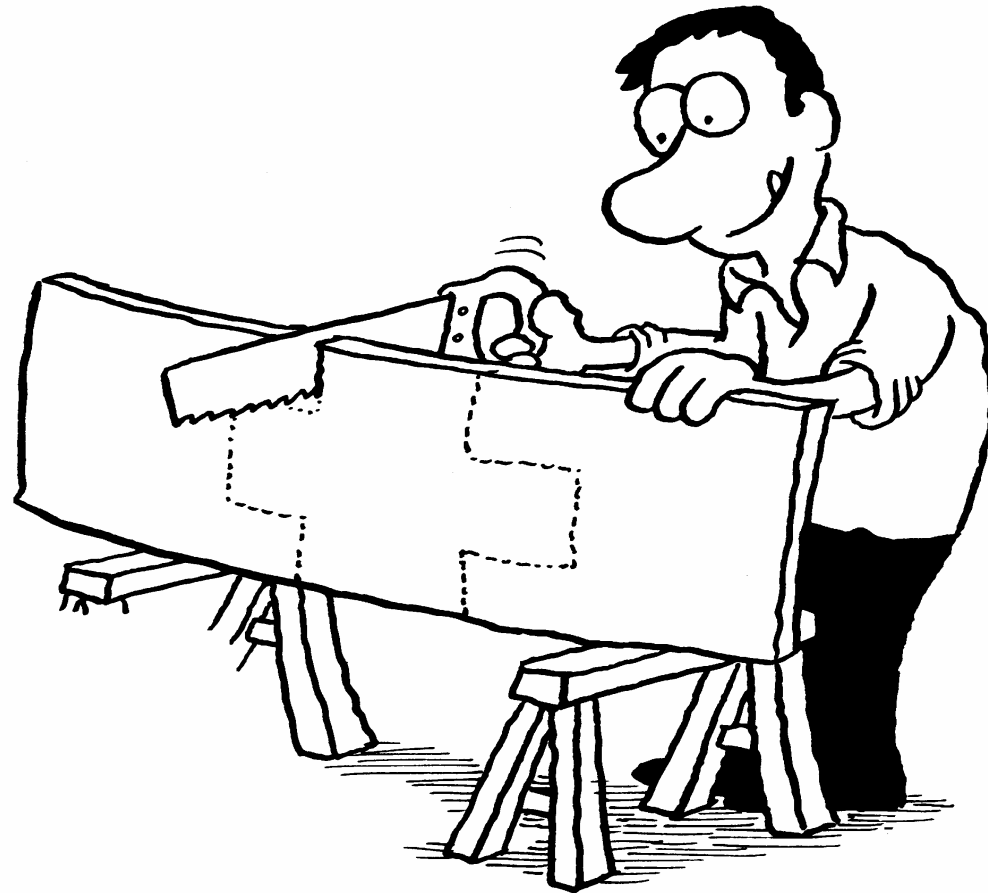
```
Controller  
void initialize  
boolean isDone  
Sampler next()  
...
```

# JMeter – Deep inside

## Zusammenspiel GUI und Testelements



# Erweitern von JMeter in Eclipse



# Erweitern von JMeter in Eclipse

- Vorbereitende Maßnahmen
- Entwicklung eines Hello Samplers
- Im Projekt entwickelte JMeter Plugins



# Erweitern von JMeter in Eclipse

## Vorbereitende Maßnahmen

- Zusätzliche jar Dateien

Jar file	Where to get
bsf.jar	<a href="http://cvs.apache.org/dist/jakarta/bsf/v2.3.0rc1/bin/bsf-bin-2.3.0.zip">http://cvs.apache.org/dist/jakarta/bsf/v2.3.0rc1/bin/bsf-bin-2.3.0.zip</a>
bsh-1.3.0.jar	<a href="http://www.beanshell.org/bsh-1.3.0.jar">http://www.beanshell.org/bsh-1.3.0.jar</a>
activation.jar (1.0.2)	<a href="http://java.sun.com/products/javabeans/glasgow/jaf.html">http://java.sun.com/products/javabeans/glasgow/jaf.html</a>
mail.jar (1.3.2)	<a href="http://java.sun.com/products/javamail/">http://java.sun.com/products/javamail/</a>
jsse.jar (1.4.1)	Bestandteil von jdk seit Version 1.4
iaik_ssl.jar iaik_jce.jar	<a href="http://jce.iaik.tugraz.at/download/evaluation/index.php">http://jce.iaik.tugraz.at/download/evaluation/index.php</a> iaikjce313ev.zip & isasilk306ev.zip
jdbc*.jar	Abhängig von verwendeter Datenbank
velocity-1.4.jar	<a href="http://apache.kgt.org/jakarta/velocity/binaries/velocity-1.4/velocity-1.4.zip">http://apache.kgt.org/jakarta/velocity/binaries/velocity-1.4/velocity-1.4.zip</a> (Generierung Dokumentation)

# Erweitern von JMeter in Eclipse

## Vorbereitende Maßnahmen

- Alternativ folgende Dateien aus Build Pfad herausnehmen
  - org/apache/jmeter/util/JsseSSLManager.java
  - org/apache/jmeter/util/keystore/PKCS12KeyStore.java
  - Org/apache/jmeter/util/keystore/DefaultKeyStore.java
  - org/apache/jmeter/visualizers/MailerVisualizer.java
  - org/apache/jmeter/reporters/MailerModel.java
  - org/apache/jmeter/reporters/MailerResultCollector.java
  - org/apache/jmeter/assertions/\*\*/BeanShell\*.java
  - org/apache/jmeter/protocol/java/\*\*/BSF\*.java
  - org/apache/jmeter/protocol/java/\*\*/Beanshell\*.java
  - org/apache/jmeter/protocol/http/sampler/WebServiceSampler.java
  - org/apache/jmeter/protocol/http/control/gui/WebServiceSamplerGui.java

# Erweitern von JMeter in Eclipse

## Entwicklung eines neuen Samplers

- SourceCode für GUI

```
public TestElement createTestElement() {
    HelloSampler sampler = new HelloSampler();
    modifyTestElement(sampler);
    return sampler;
}

public void configure(TestElement element) {
    super.configure(element);
    txtToGreet.setText(element.getPropertyAsString(HelloSampler.TO_GREET));
}

public void modifyTestElement(TestElement element) {
    configureTestElement(element);
    element.setProperty(HelloSampler.TO_GREET, txtToGreet.getText());
}

public String getStaticLabel() {
    return "Hello Sampler";
}
```

# Erweitern von JMeter in Eclipse

## Entwicklung eines neuen Samplers

- SourceCode für Testelement

```
public class HelloSampler extends AbstractSampler implements Serializable
{
    public static final String TO_GREET = "HelloSampler.ToGreet";
    public HelloSampler() {
        super();
    }

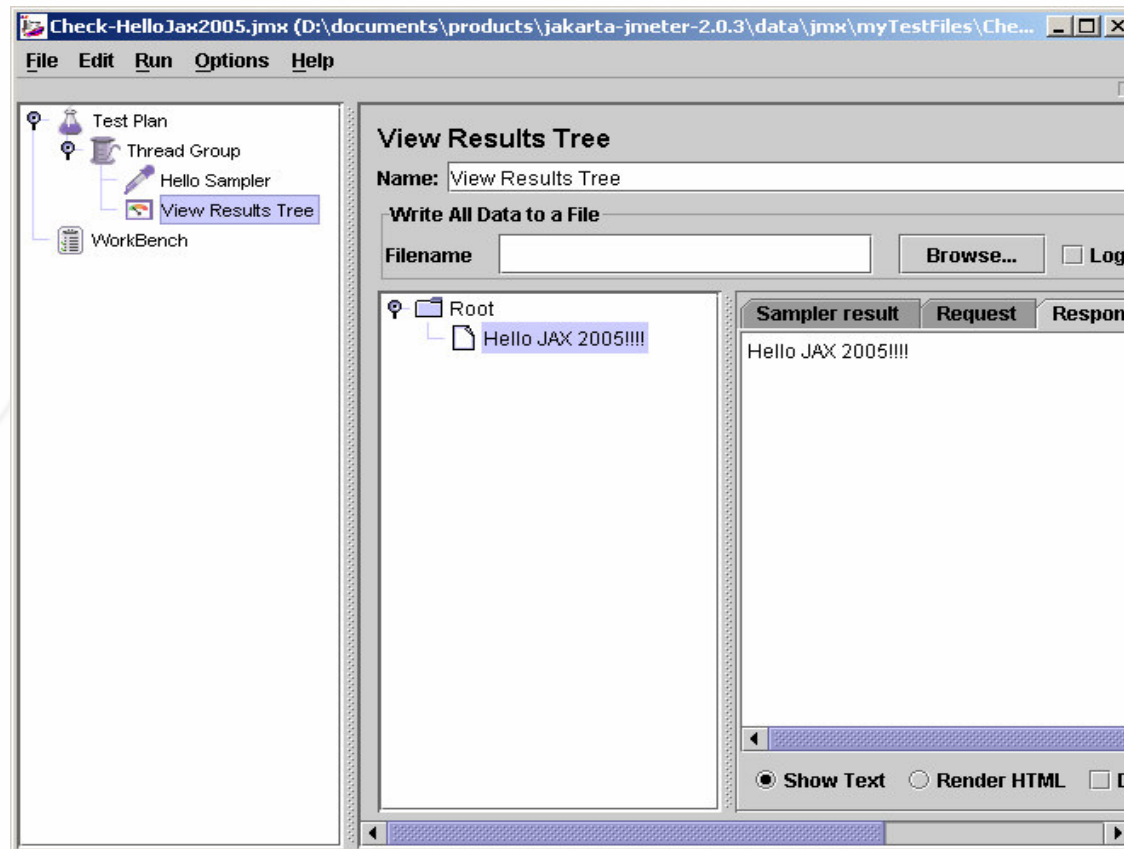
    public SampleResult sample(Entry e) {
        SampleResult res = new SampleResult();

        String hello = "Hello " + this.getPropertyAsString(TO_GREET) + "!";
        res.setDataTypes(SampleResult.TEXT);
        res.setResponseData(hello.getBytes());
        res.setSampleLabel(hello);
        res.setSuccessful(true);

        return res;
    }
}
```

# Erweitern von JMeter in Eclipse Entwicklung eines neuen Samplers

- Live - Präsentation in JMeter ...

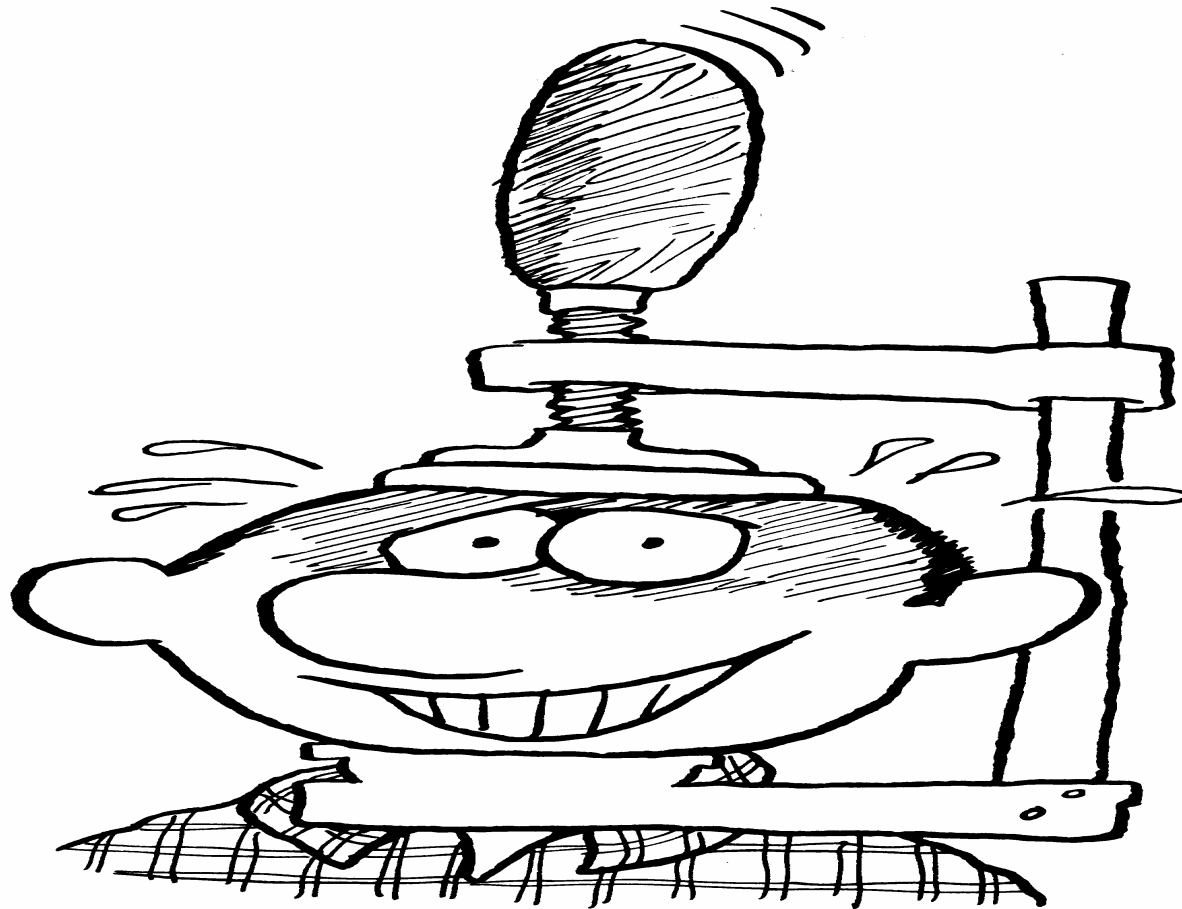


# Erweiterbarkeit von JMeter

## Im Projekt entwickelte JMeter Plugins

- JMX Sampler
- HL7 Sampler
- CommandLine / BatchSampler
- AntScript\*Sampler
- TelnetSampler
- SSH / SFTP Sampler
- Comment Controller
- XPath Postprocessor (analog Regular Expression Extractor)
- Regular Expression Counter
- HL7Assertion
- FormattedActualDateFunction
- ...

# JMeter im Integrationstest



# JMeter im Integrationstest

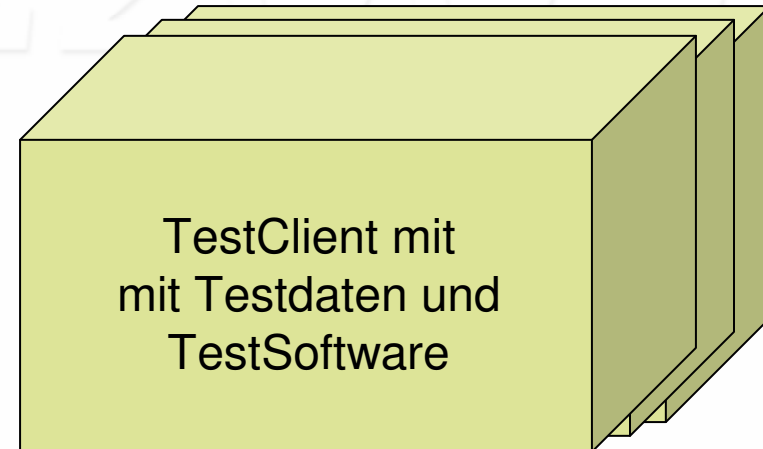
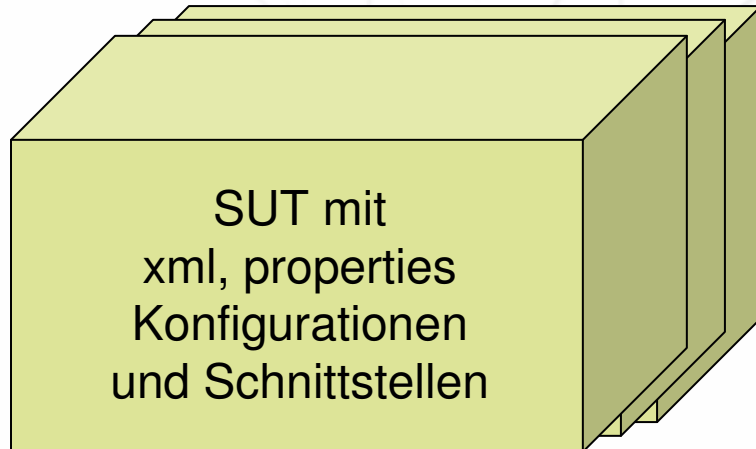
- Übersicht
- Umgebung
  - Testplanung
  - Testdurchführung
  - Vision
- Konfiguration
  - Test Client
  - SUT (System under Test)
- Spezifikation und Implementierung der Testfälle
- Bündelung und Ausführung von Testsets



# JMeter im Integrationstest

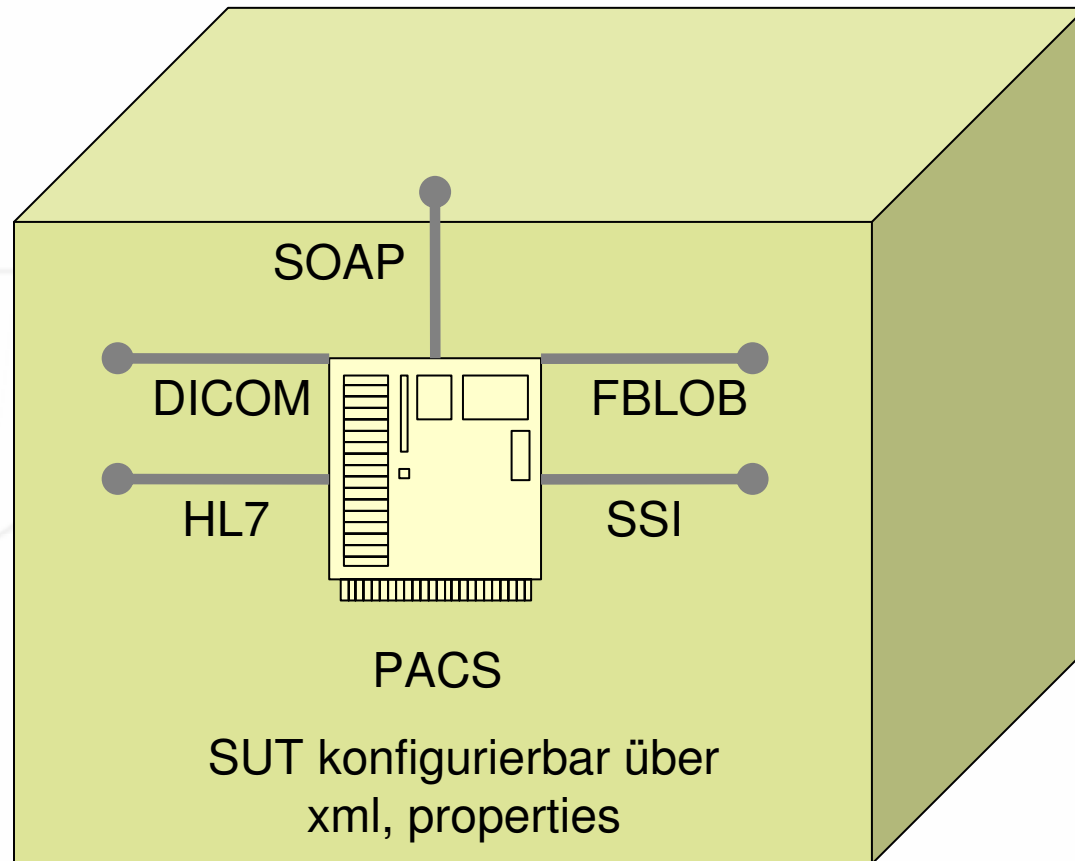
## Übersicht

- Mehrere Server mit installiertem SUT
- Mehrere Testclients mit eingerichteter Testumgebung



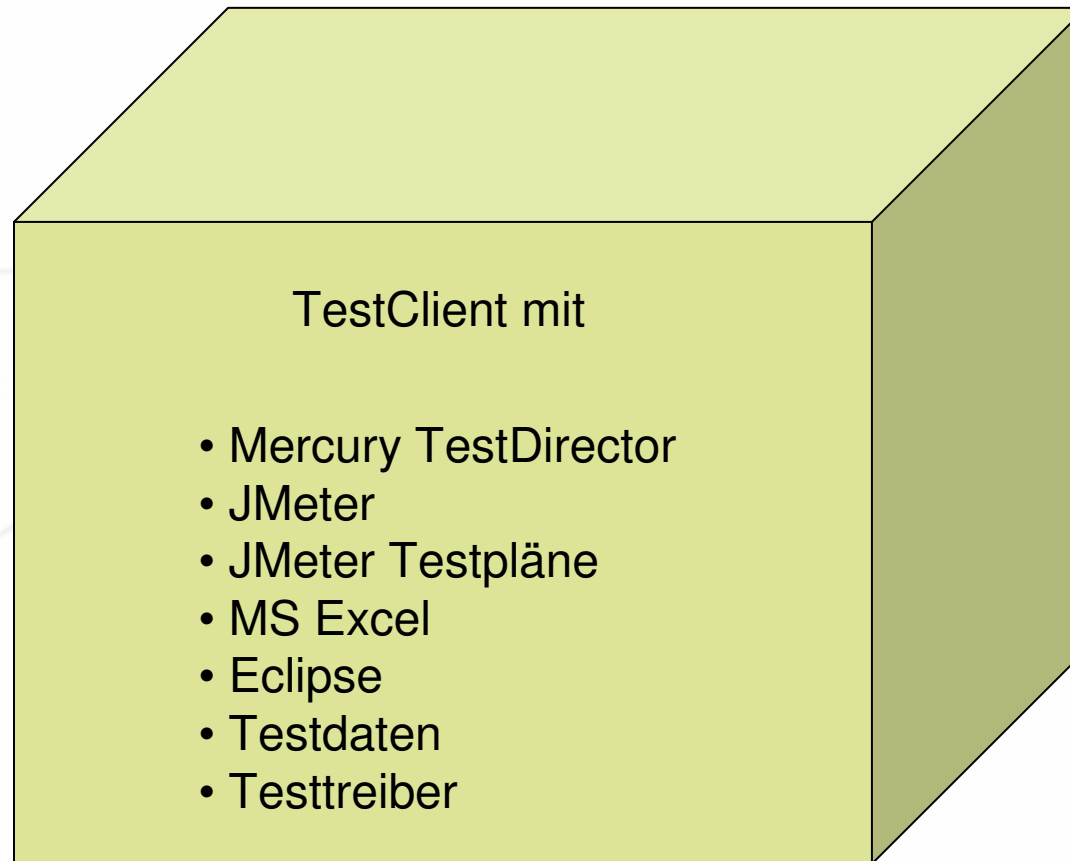
# JMeter im Integrationstest

## Übersicht

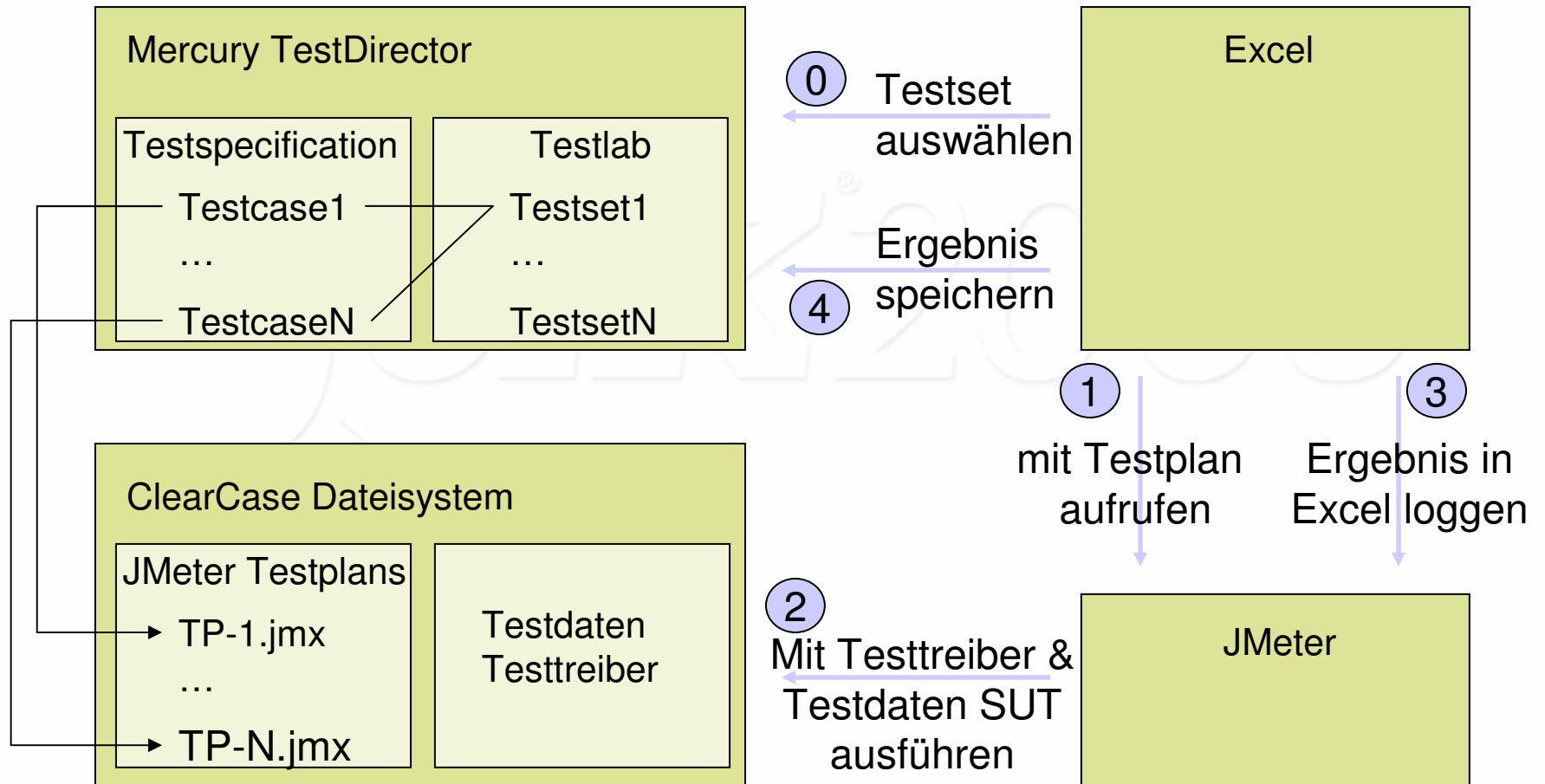


# JMeter im Integrationstest

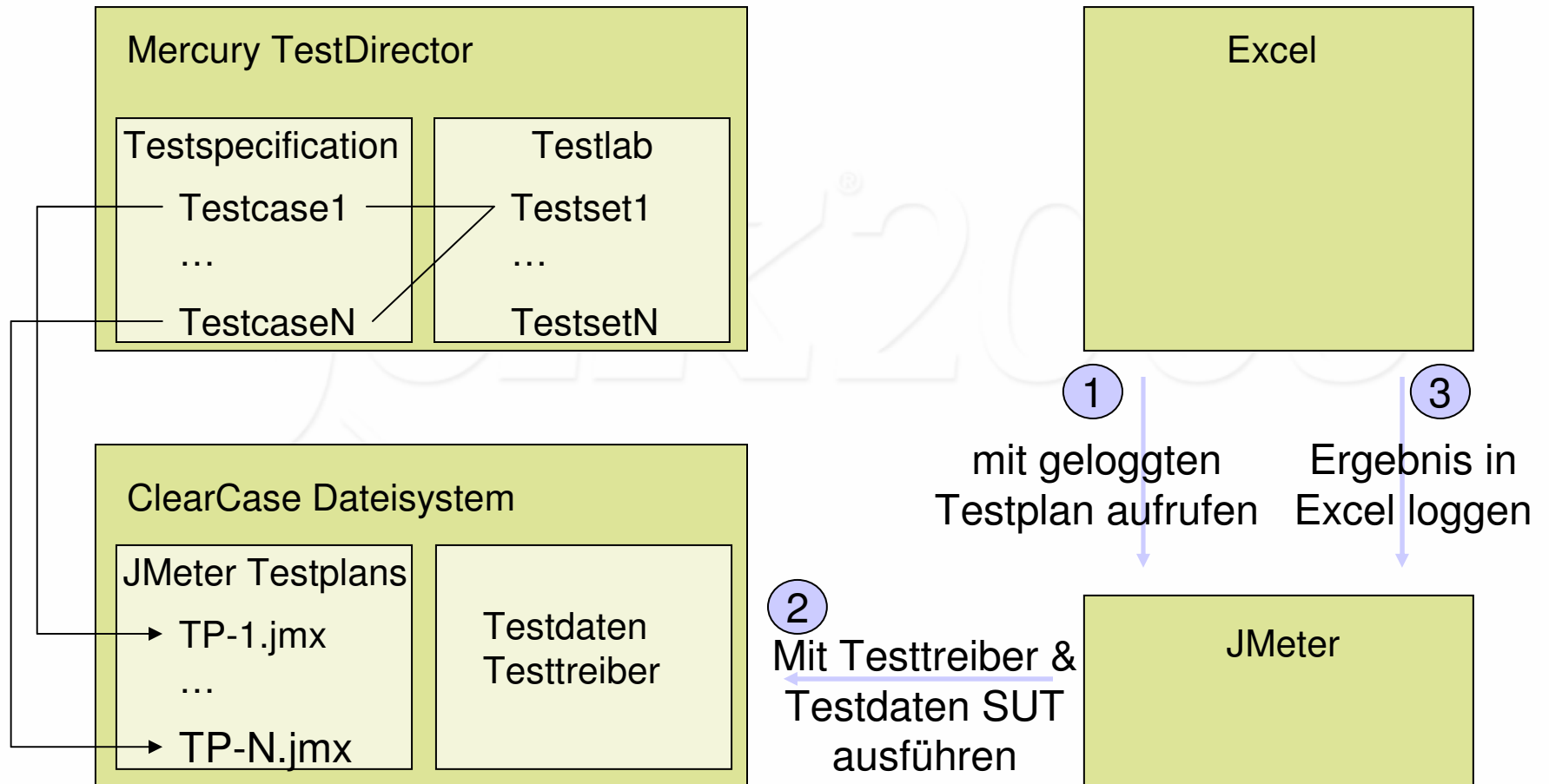
## Übersicht



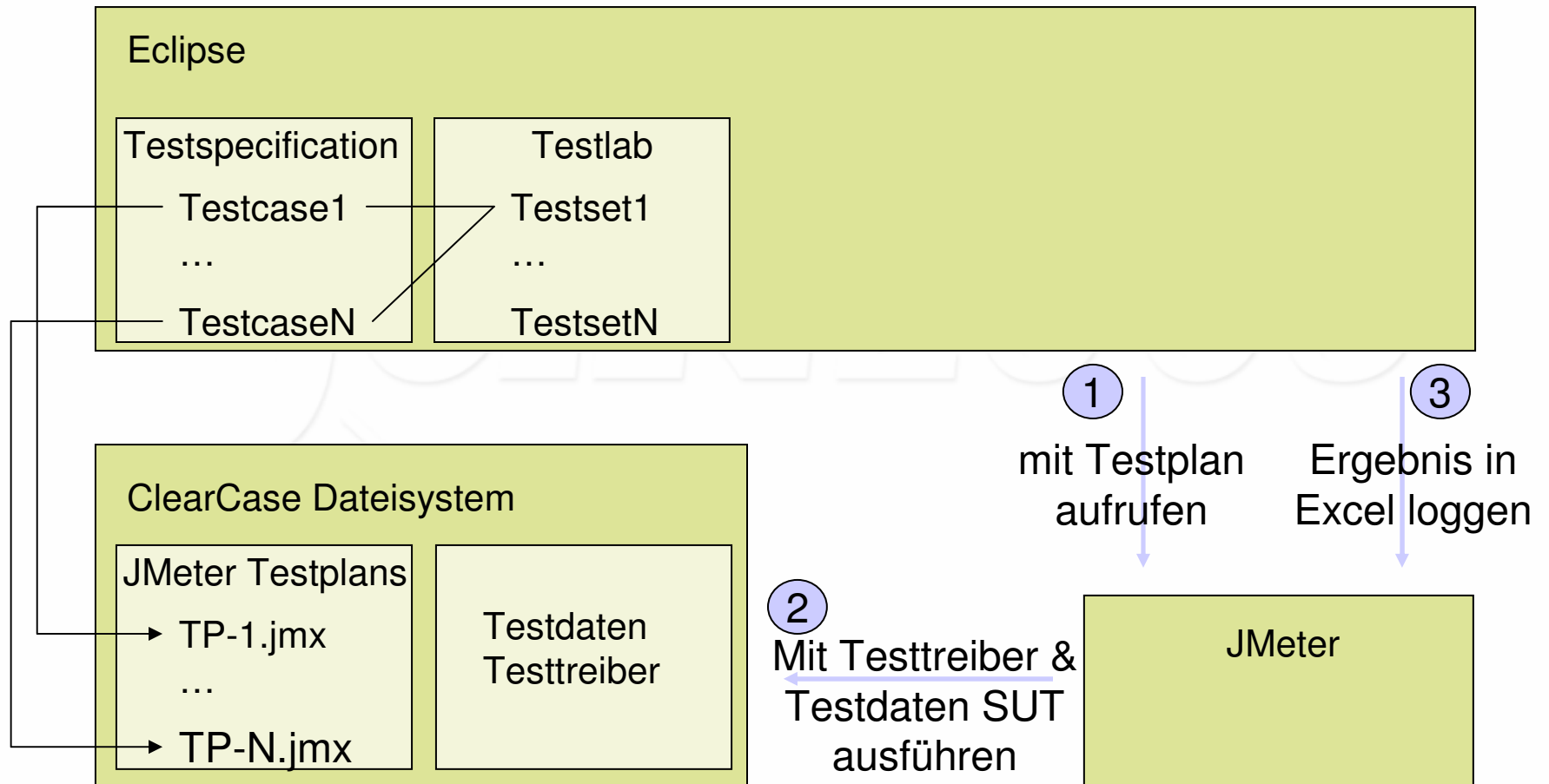
# JMeter im Integrationstest Umgebung – aktuell (I)



# JMeter im Integrationstest Umgebung – aktuell (II)



# JMeter im Integrationstest Umgebung – Vision (Ablösung TestDirector)



# JMeter im Integrationstest Konfiguration

- Notwendige Konfigurationen
  - Test Client
    - Einstellung der Testtreiber auf das Ziel-SUT
    - Rechnerspezifische Einstellungen (Verzeichnisangaben)
  - SUT (System under Test)
    - XML Konfigurationen
    - Properties
    - Systemeinstellungen (z.B. /etc/hosts)
- Durchgeführt durch
  - AntScriptSampler für
    - XSLT Transformationen
    - Properties Ersetzungen

# JMeter im Integrationstest

## Spezifikation und Implementierung der Testfälle

- Sichtung der Schnittstellen- und funktionalen Spezifikationen
- Strukturierung nach Schnittstellen(-methoden)
- Testfallermittlung mit Äquivalenzklassenanalyse
- Vorbedingungen des Testfalls ermitteln
  - SUT Konfiguration
- Extraktion benötigter
  - JMeter Erweiterungen
  - Wiederverwendbarer Teilskripte
- Erstellung eines JMeter Skripts pro
  - Testfall
  - Vorbedingung (z.B. Tapearchiv angeschlossen)



# JMeter im Integrationstest

## Bündelung und Ausführung von Testsets

- Tests mit gleichen Vorbedingungen in Testsets zusammenfassen
- TestDirector Testsets mit Excel auswählen
- Log der Testergebnisse in TestDirector und Excel
- Logs in Excel ermöglichen spätere Ausführung (ohne TestDirector)

# Fazit / Zusammenfassung



# Fazit / Zusammenfassung

- 😊
  - Java Bibliotheksunterstützung (<-> WinRunner TCL)
  - Erweiterungsfähigkeit
  - Reproduzierbarkeit der Fehlerfälle durch Entwickler
  - Ausbarkeit zu Last- und Stresstestszenarien
  - Produktivitätskurve
- ☹️
  - Ressourcenverbrauch von JMeter ist hoch
  - Keine Planübergreifenden Includes
  - Keine Verwaltung mehrerer Pläne

# Fazit / Zusammenfassung

- Mögliche Ausbaustufen
  - Verwaltung von Testplänen in Eclipse
  - Aufruf einer Auswahl von Testplänen aus Eclipse
  - Testwerkzeuge anderer Teststufen einbinden
  - Verwendung als PowerO-Self-Tests im Feld
  - Ausbau der Stress- und Lasttestszenarien
  - Mehr Testtreiber direkt einbinden
  - Spezifischere SOAP-Aufrufe

# Weiterführende Links und Quellen

- Bücher
  - J2EE-Entwicklung mit Open-Source-Tools, Martin Backschat / Stefan Edlich, Spektrum Verlag, 2004 (ein paar Seiten)
- Artikel
  - JMeter Tutorial unter <http://wiki.apache.org/jakarta-jmeter/JMeterLinks> von Mike Stoever und Peter Lin (noch in Entstehung ;-)
- Links
  - JMeter Website <http://jakarta.apache.org/jmeter>
  - Eclipse Website <http://www.eclipse.org>

# Besten Dank



Martin Heider  
Infomar software  
[mh@infomar.de](mailto:mh@infomar.de)  
[www.infomar.de](http://www.infomar.de)