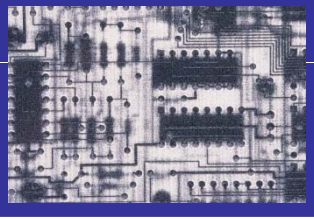
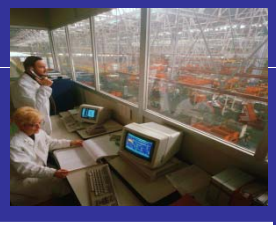
Mod_ooRexx

Final presentation
2010-01-28

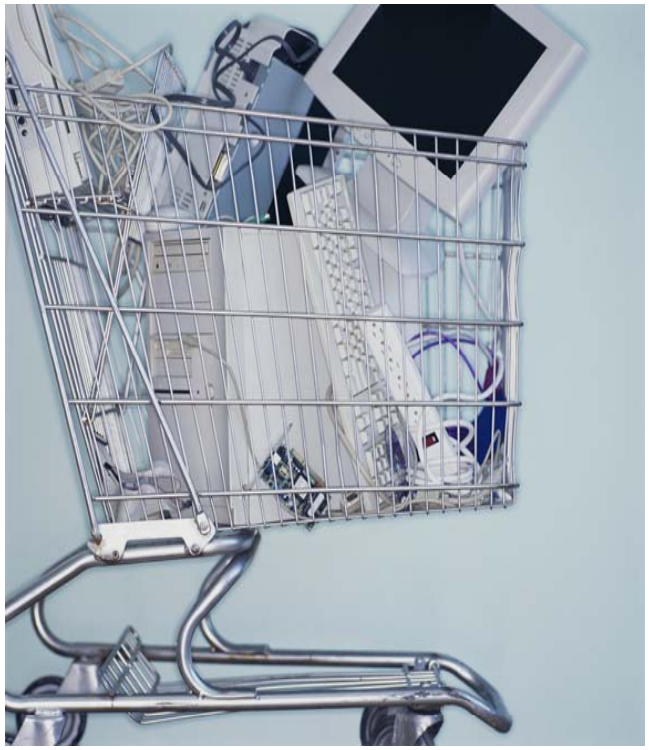





Maschek Robert

ID: 8952711

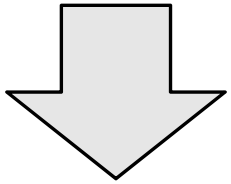
Agenda

- **Introduction**
- **Apache 2 Software architecture**
  - Basic concepts
  - Core objects
  - Request handling
- **Mod_ooRexx**
  - Testsystem
  - Live demonstration: Examples
- **Conclusion**
- **Questions and Answers**



Introduction – I – Focus

“A beginner’s guide for installing and using ooREXX on the Apache webserver.”



Focus on:

- Apache webserver Architecture
- Testsystem (Fedora 12; Windows 2008)
- Mod_ooRexx installation
- Basic examples

Not in focus:

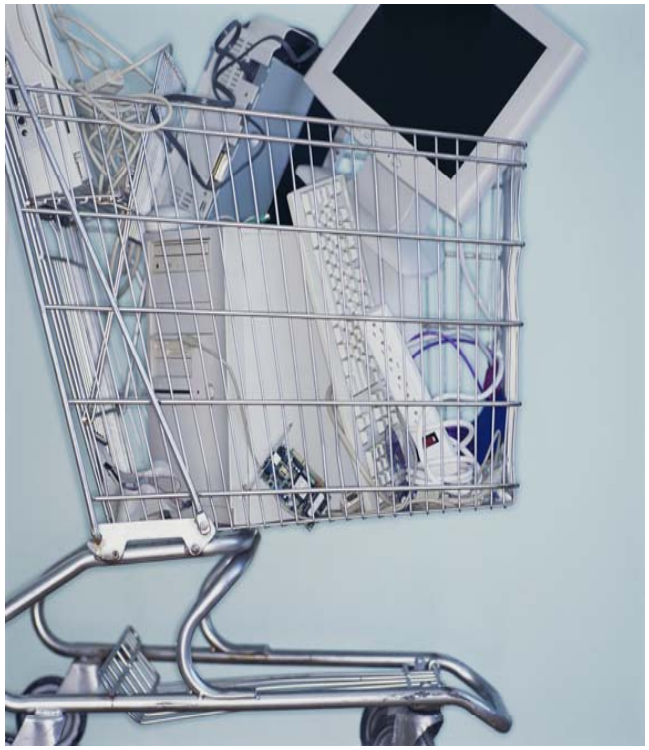
- Nuts and Bolts of Module programming
- Apache Webserver configuration and administration

Introduction – II – Apache - a brief history

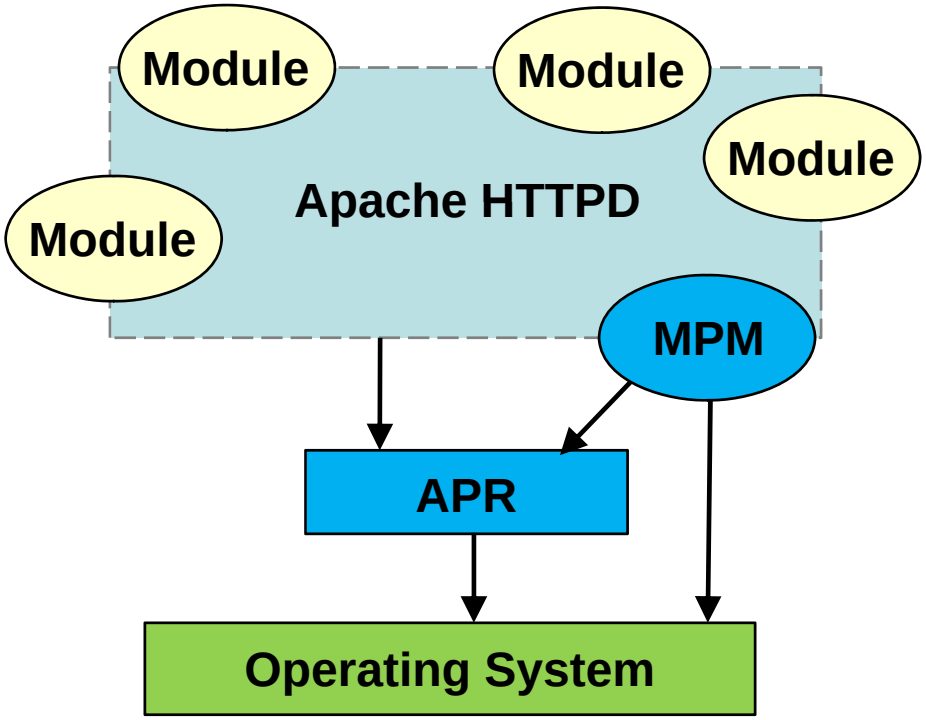
- **Base: NCSA httpd server**
- **April 1995: Version 0.62**
  - Naming: 2 versions (“a patchy server”; “respect for the various Native American nations collectively referred to as Apache”)
- **December 1995: Version 1.0**
  - Redesign of the codebase; adding some features
- **February 1998: Plans for Version 2.0 summarized**
  - The same code for every platform
- **April 2002: Version 2.0.35**
  - First general available version of Apache 2

Agenda

- Introduction
- **Apache 2 Software architecture**
  - Basic concepts
  - Core objects
  - Request handling
- Mod_ooRexx
  - Testsystem
  - Live demonstration: Examples
- Conclusion
- Questions and Answers



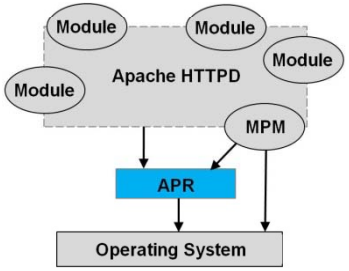
Apache 2 architecture – I – Overview



(Kew 2007, p. 22)

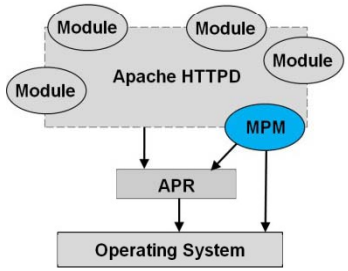
- **relatively small core**
- **most important parts**
  - Apache Portable Runtime (APR)
  - Multi-Processing Module (MPM)
- **Modules**
  - compiled statically into the server
  - Loaded dynamically at runtime

Apache 2 architecture – II – APR



- **Middleware between Operating System and the Apache HTTPD**
- **Standard application programming interface (API)**
- **Consistent interface to underlying platform-specific implementations**
- **maintained by the Apache Software Foundation**
- **APR is not only used for the webserver**
  - Tomcat, Subversion, …

Apache 2 architecture – III – MPM



- **Optimize Apache for the underlying operating system so that incoming requests are mapped onto an execution primitive.**
- **Thread or process**
- **Supports three different kinds of MPM**
  - MPM module: prefork (Unix); beos (BeOS)
  - MPM module: mpm_netware (Netware); mpm_winnt (Microsoft)
  - Hybrid mode: worker, event and mpmt_OS2 (OS2)

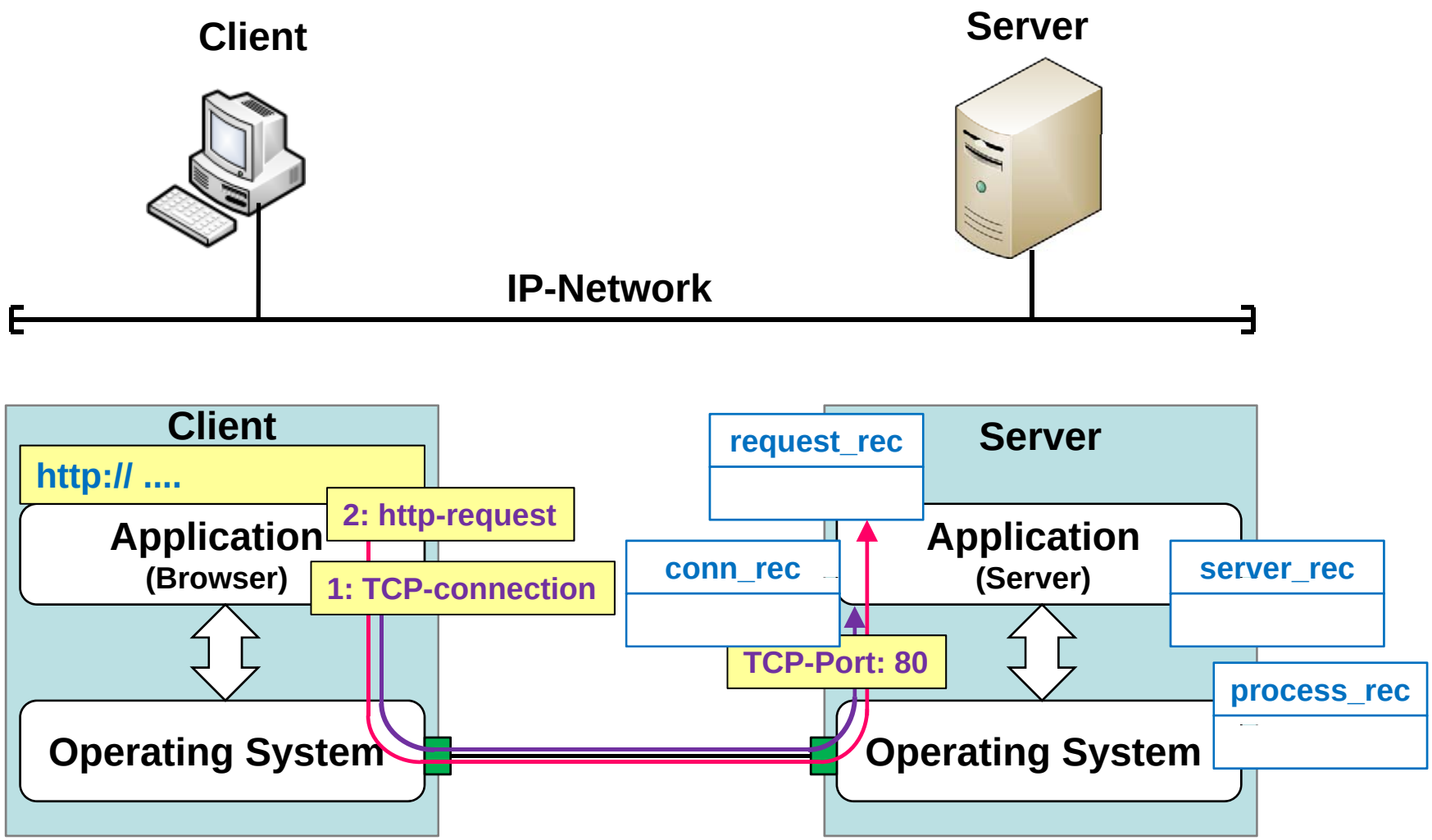
Apache 2 architecture – IV – Pools

- **Grouped collection of resources**
  - file handles, memory, child programs, …
- **Hierarchically structured**
  - Pool → Subpool → Subsubpool and so on
- **During the startup phase Apache creates a pool from which all others are derived.**
  - Configuration information is held in this pool
- **The next level of pools is created for each connection Apache receives and is destroyed when the connection ends**
  - A connection can span several requests and a new pool is created (and destroyed) for each request.

Apache 2 architecture – IV – Operation

- **Startup phase**
  - Reads and verifies the configuration file(s) httpd.conf
  - Loading modules, initialize system resources such as log-files, shared memory segment
  - Open network connections
  - single-process, single-thread program and has full system privileges
  - Before entering the operational phase Apache relinquishes its system privileges.
- **Operational phase**
  - child processes or threads will accept external connections

Apache 2 architecture – V – Core objects



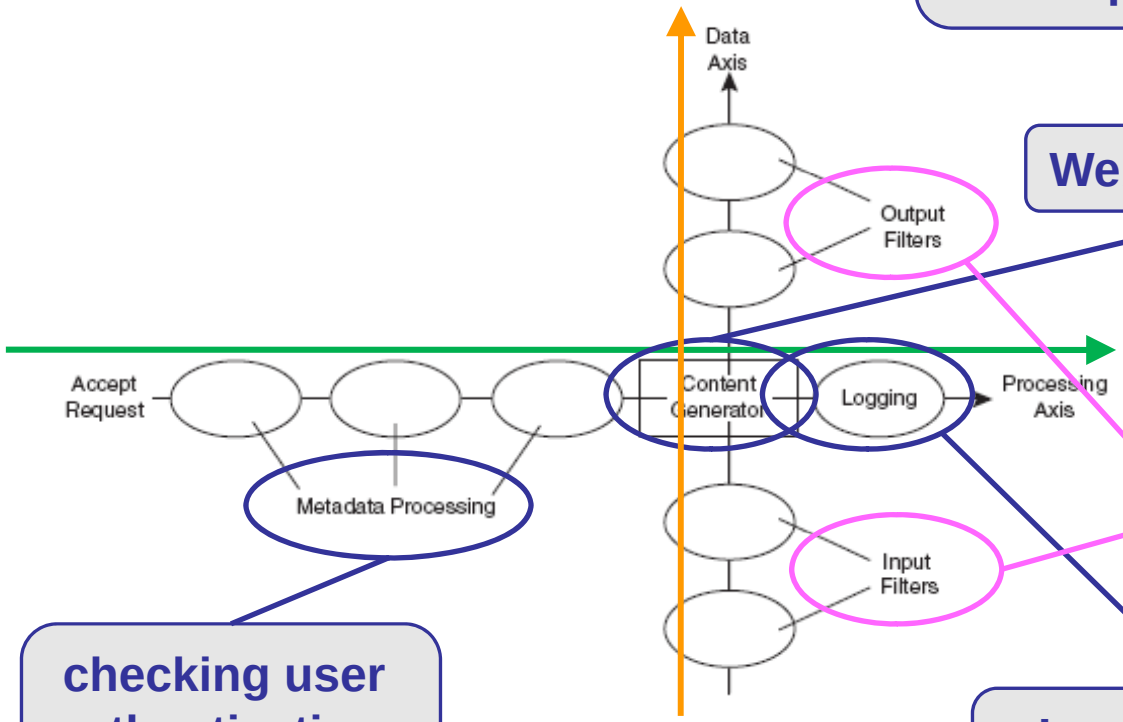
Apache 2 architecture – V – Core objects

- **process_rec**
  - more a part of the operating system
  - main goal is the handling of pools
- **server_rec**
  - created during the startup phase
  - defines a logical webserver (more for virtual hosts)
- **conn_rec**
  - created when a client connects to the webserver
- **request_rec**
  - created whenever a request is accepted
  - stores and processes all the relevant data for all stages of the entire request handling process

Apache 2 architecture – VI – Request handling

New concept:
Filter chain

For a more detailed view:
- **See the paper**
- **The Apache Modeling Project**



Webpage creation

basic operation of a webserver

much cleaner and more efficient data processing

checking user authentication, URI mapping …

Logfile entries created

(Kew 2007, p. 47)

Apache 2 architecture – VII – Hooks

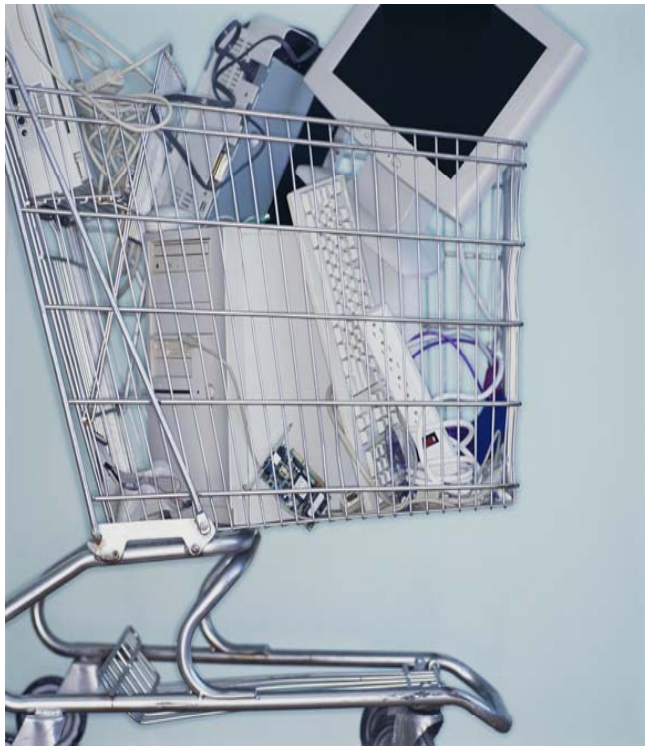
- **A point at which a module can request to be called**
- **Interaction between modules and the webserver**
- **Functions can return either "declined" or "ok" or an error**
- **Offers 10 hooks in total for modules to step in**

Examples:

- **translate_name:**
  - Maps the request URL to the filesystem.
- **access_checker:**
  - Apache checks whether access to the requested resource is permitted according to the server configuration (httpd.conf)

Agenda

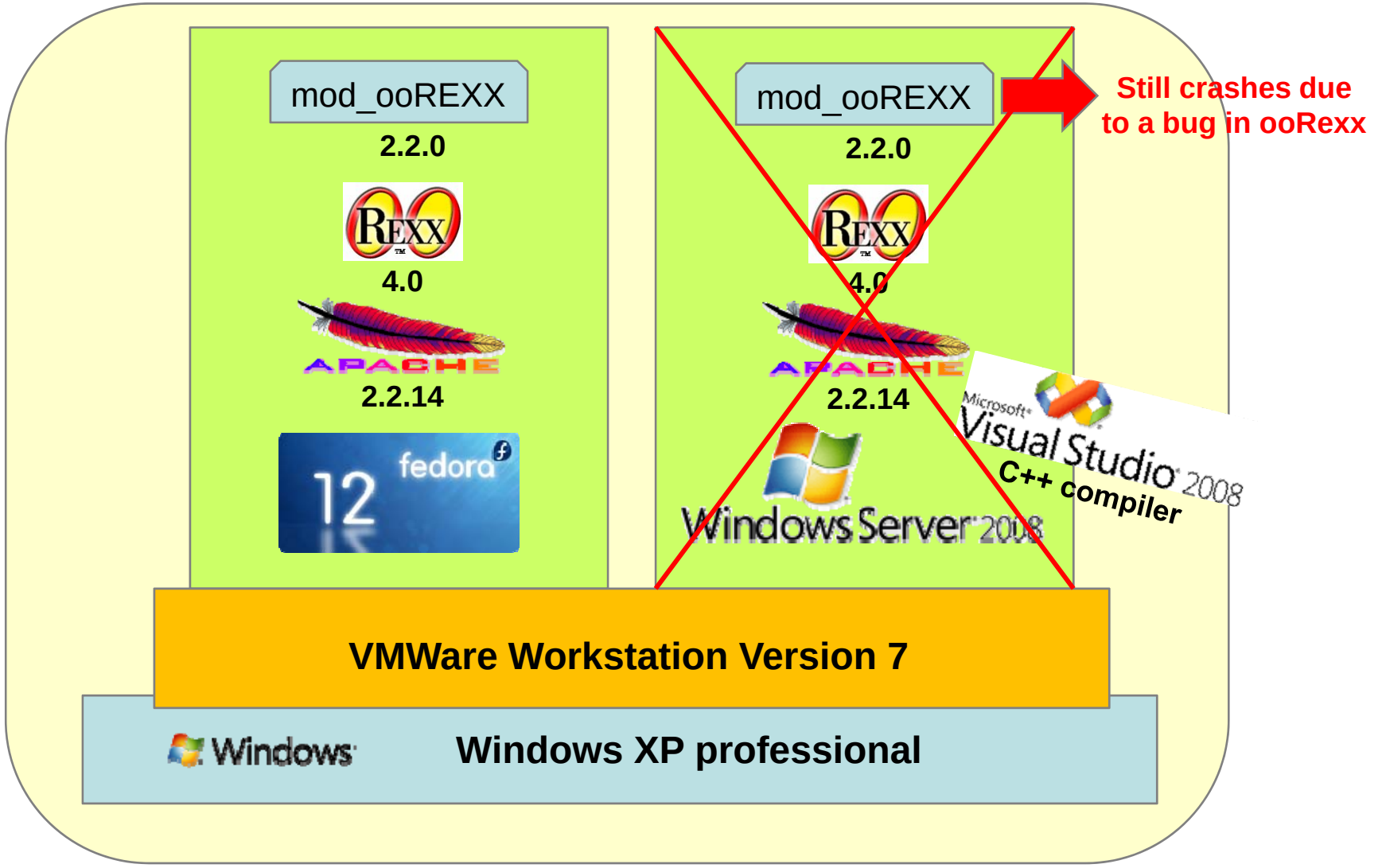
- Introduction
- Apache 2 Software architecture
  - Basic concepts
  - Core objects
  - Request handling
- **Mod_ooRexx**
  - Testsystem
  - Live demonstration: Examples
- Conclusion
- Questions and Answers



Mod_ooRexx – I – A brief history

- **Mid 1990s: IBM offered Object REXX**
  - originally for the OS/2 and OS/390
  - Based on REXX (REstructured eXtended eXecutor)
- **2004: Object REXX was transferred to the Rexx Language Association (RexxLA)**
  - became an open-source project named Open Object REXX
- **REXX and ooREXX are compatible**
- **Mod_Rexx**
  - Include Rexx into the Apache webserver
- **Mod_ooRexx**
  - rewritten to improve the performance and to support the latest version (4.0) of ooREXX

Mod_ooRexx – II – Testsystem

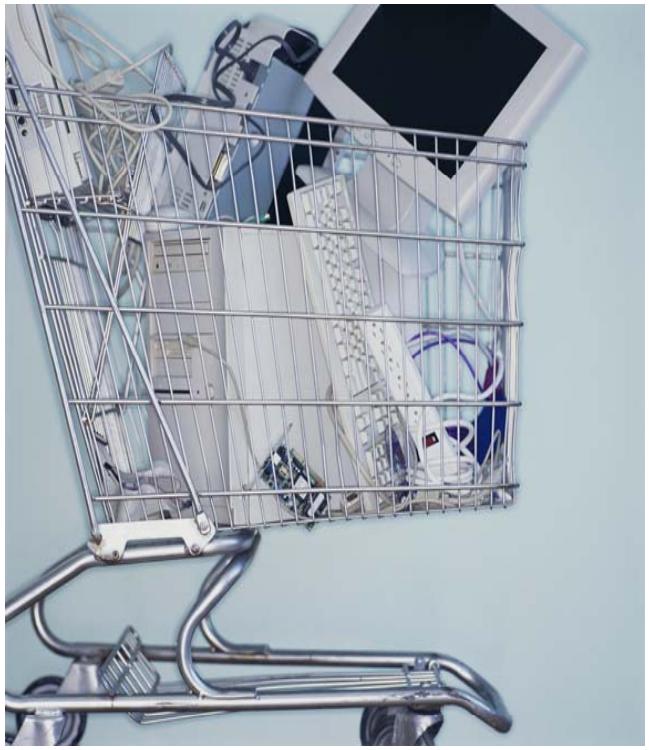


Mod_ooRexx – II – Examples

Running on

Agenda

- Introduction
- Apache 2 Software architecture
  - Basic concepts
  - Core objects
  - Request handling
- Mod_ooRexx
  - Testsystem
  - Live demonstration: Examples
- **Conclusion**
- **Questions and Answers**



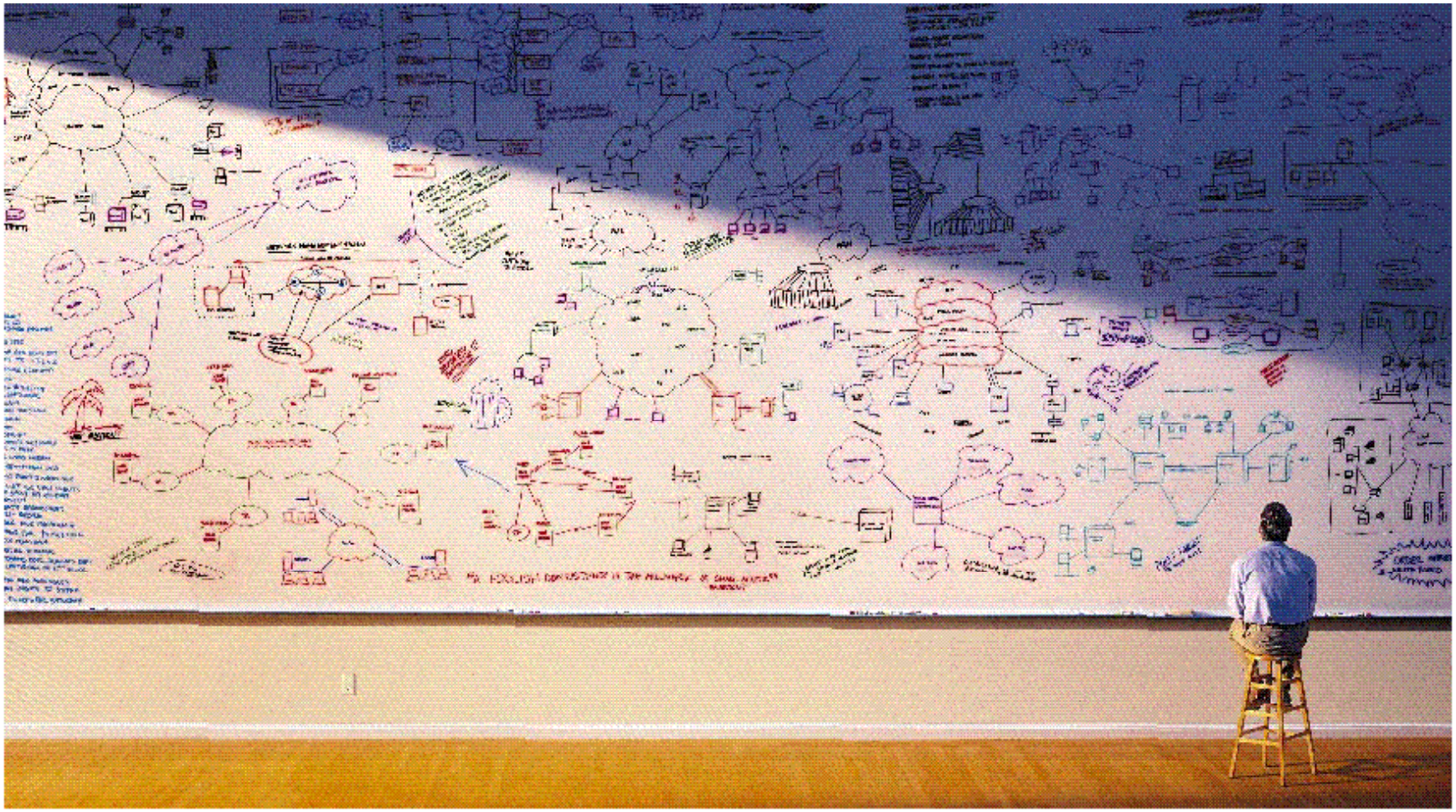
Mod_ooRexx – I – Conclusion

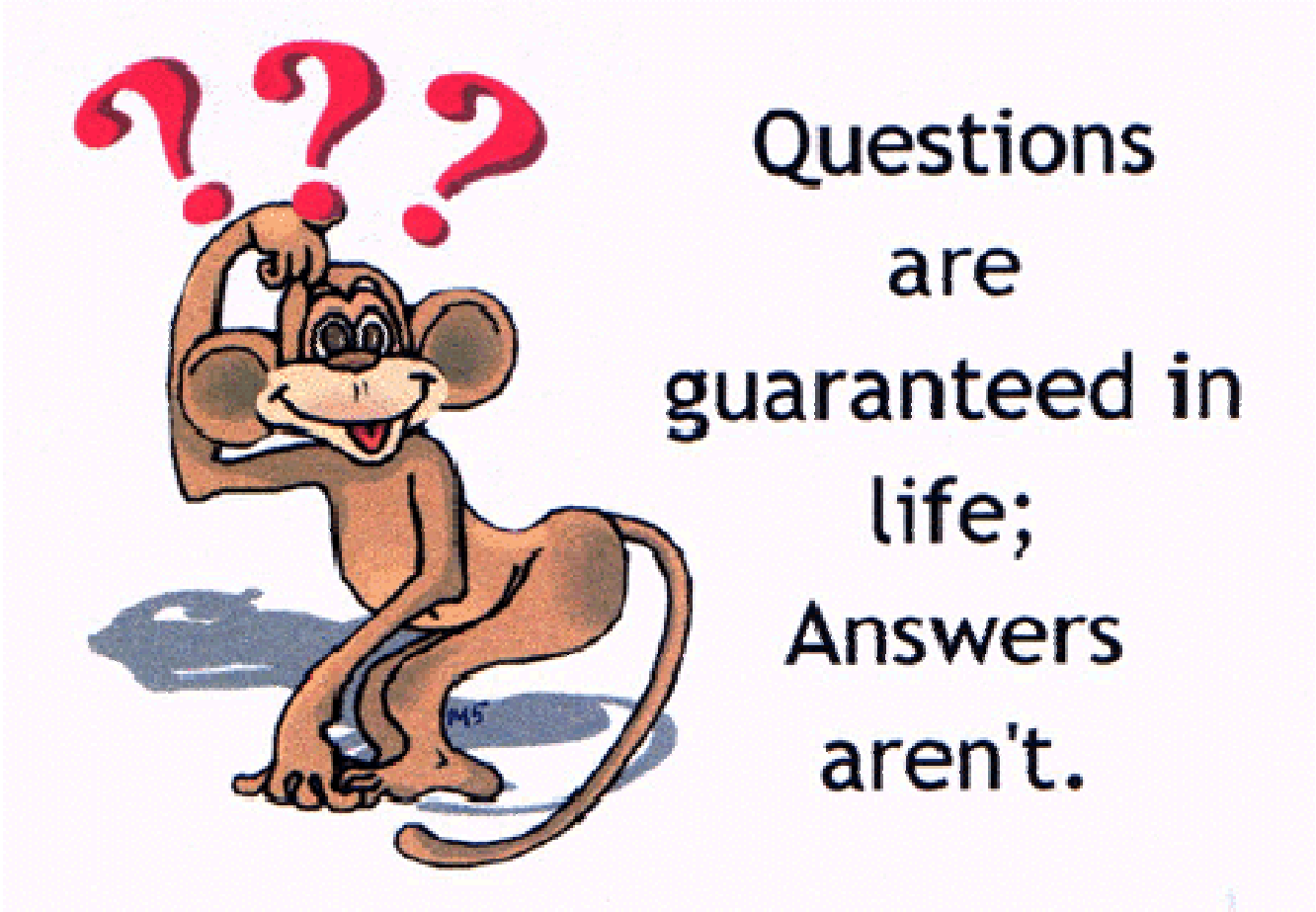
- **Powerful package**

- **Limitations**
  - Lack of resources available results in timeconsuming implementations
  - The existing bug in the Windows version

- **Lot of space for academic work**
  - Examples of using all the Hooks
  - BSF4Rexx and mod_ooRexx in a web environment

Questions and Answers




Questions
are
guaranteed in
life;
Answers
aren't.