

TCL and Content Management Systems

4. European Tcl/Tk User Meeting

2003-05-30

Thomas Witt, Director Business Development

Infopark AG

Facts

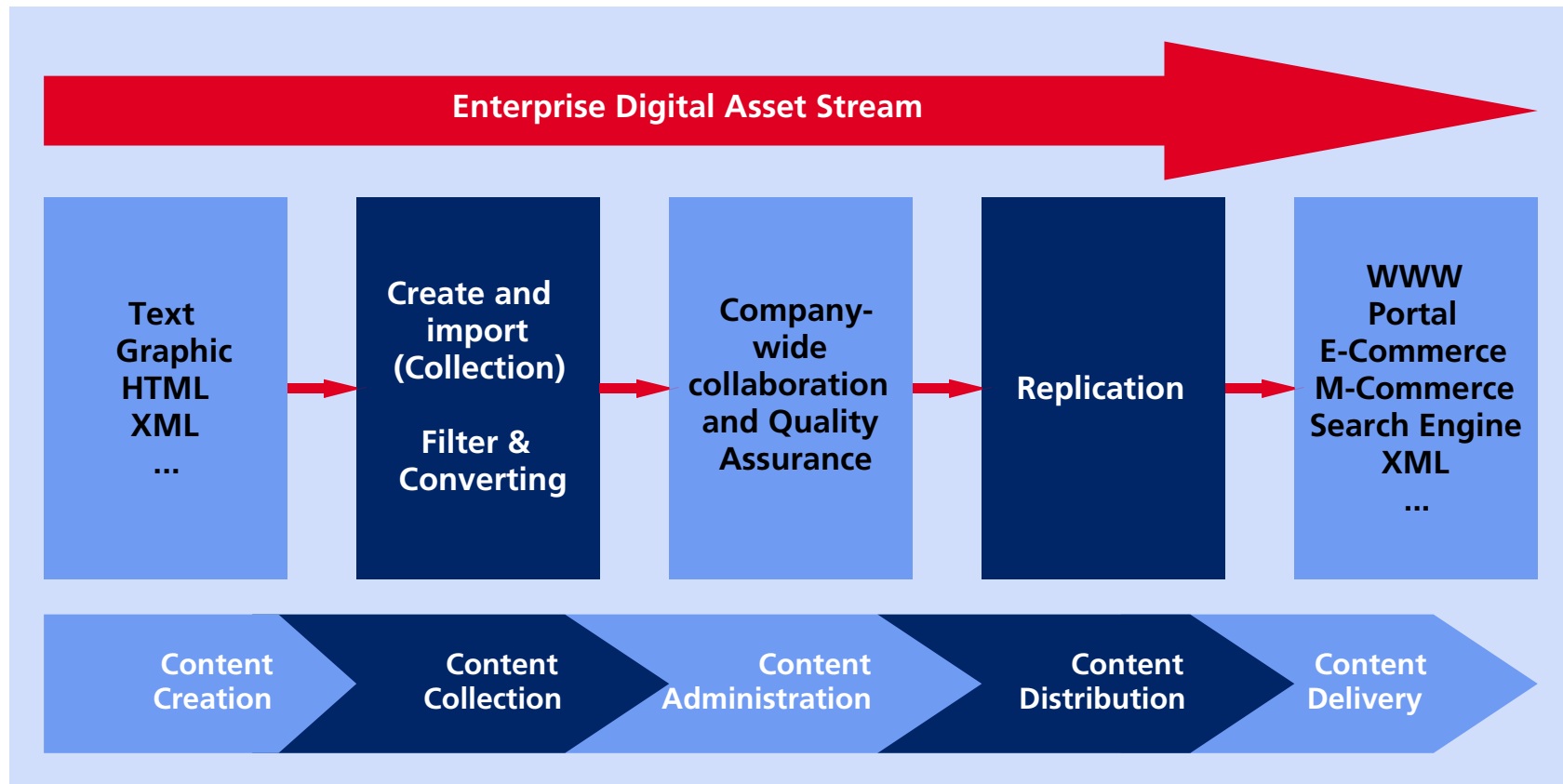
- **Founded: 1994, still privately held**
- **Located in Berlin, Germany**
- **About 80 employees**
- **Profitable business**
- **Member of several standardization committees (W3C)**

Business

- **Content Management Software „NPS“**
- **Flagship Product: Content Management Server NPS**
 - **Using TCL since 1997**
- **Mission: To create a flexible and standards-based CMS**

Infopark is one of the leading european vendors for CMS

What is Content Management Software



Content Management Software reduces maintenance costs

NPS includes TCL - Why?

Open Source

- Nonproprietary
- Broad code base

Easy Integration

- Typical glue language
- Extensible by the vendor using own commands
- Rapid deployment

Built-In Security

- „Safe Interpreters“

Learning curve

- Easy to learn(in opposite to Perl, Python, ...)

TCL is THE choice regarding glue languages

TCL and NPS

Standardized Scripting Environment

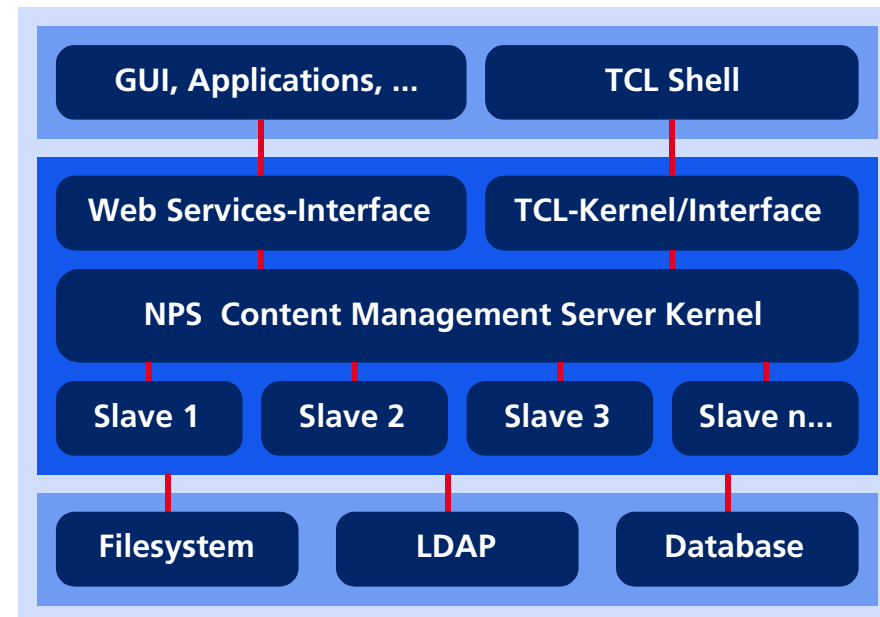
- „SQL for CMS“
- **Automatization**
 - Reports
 - Content import
- **Automated testing**

Callbacks

- **Business rules**
- **Value verification**

APIs

- **User Management**



TCL is the standardized scripting environment of NPS

TCL - How to integrate?

NPS is written in Objective-C

- Includes a complete TCL Interpreter
- Standard command set extended by own commands
- Commands can be executed locally or in the NPS kernel

Client-Server enabled

- NPS can be accessed from any TCL interpreter
 - Remote-Initialization with standard procedures
- All standard commands are local, all other are remote

Integration steps in the server code

- Definition of the command itself
- Mapping („registration“) of the command in TCL

Easy Integration

Command definition/registration

Procedure
for
creating
commands

```
+ (void)createCommand: (const char *)theCmd
  withProc: (TclCmdProc *)theProc isSafe: (BOOL)isSafe
{
  [...]
  Tcl_CreateCommand(globalInterpreter, (char *)theCmd,
    tclCmd, (ClientData)theProc, NULL);

  if (isSafe == YES && safeInterpreter) {
    Tcl_CreateAlias(safeInterpreter, (char *)theCmd,
      globalInterpreter, (char *)theCmd, 0,
    }
}
```

Definition

```
void tclWhoamiCmd(Tcl_Interp *interpreter, int argc, char
{
  [HSBatchProcessor setTclResult: [APP login]
    forInterpreter: interpreter];
}
```

Registra-
tion

```
[HSBatchProcessor createCommand: "whoami"
  withProc: tclWhoamiCmd isSafe: YES];
```

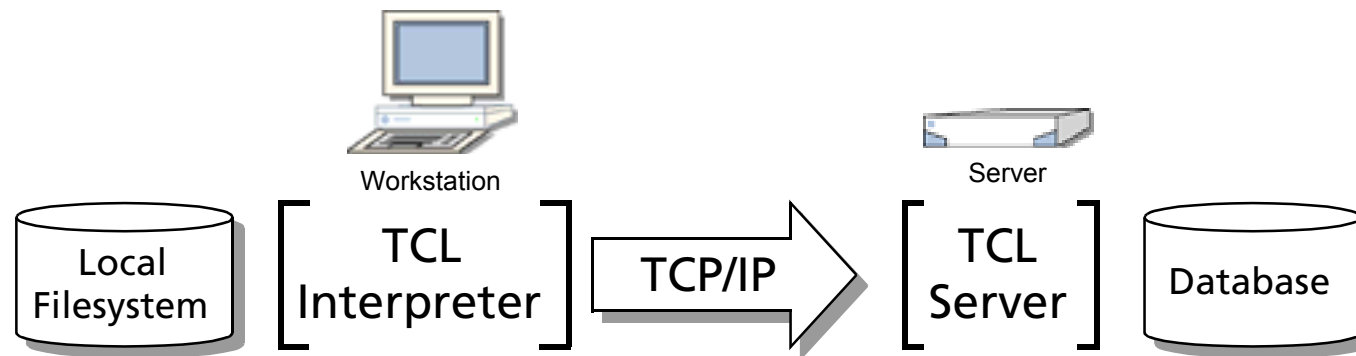
The procedure how to define/map commands

Command execution (client-server)

```
nps5@tristan:~/TclClient> tcl client.tcl
>>> puts "Hello world"
Hello world

>>> whoami
invalid command name "whoami"

>>> connect tristan.infopark.de 3001 root root
CM 5.2.1 - Copyright (C) 1999-2002 Infopark AG. - All rights reserved.
CM> whoami
root
CM> logout
>>>
```



The execution of the command „whoami“

Example: Object modification

```
CM> obj withPath /en/news/press get contentType  
html
```

```
CM> obj withPath /en/news/press get title  
Press Service
```



```
CM> obj withPath /en/news/press edit
```

```
CM> obj withPath /en/news/press editedContent set title "TCL-Test"
```

```
CM> obj withPath /en/news/press release
```

```
CM> obj withPath /en/news/press get title  
TCL-Test
```



```
CM> obj withPath /en/news/press create objClass document n  
7618
```

```
CM> obj withId 7618 editedContent set blob [loadFile C:\ne
```

```
CM> llength [user list]  
78
```

Reading and editing the document hierarchy

Business Rules and Callbacks

Callbacks are used for ...

- verifying a value,
- dynamically modifying the workflow,
- reformat values for displaying,
- modify links on the fly,
- execute procedures within the HTML page, ...

Example: Mail notification on Workflow-Forward

```
proc notificationCmd {logType objectId userLogin
receiverLogin logText} {
    set cId [lindex [obj withId $objectId get contentIds] 0]
    set mail [content withId $cId get Mailbenachrichtigung]
    if {$logType == "action_forward_obj" && "$mail" == "yes"}
        sendMail $objectId $receiverLogin $logText
    }
}
```

With callbacks the functionality can be easily extended

Business Rules and Callbacks (2)

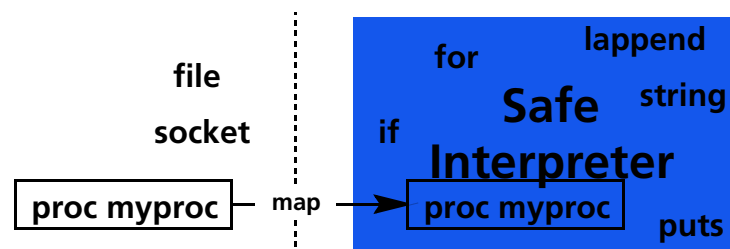
Example of a Safe Interpreter Value-Callback

- Automatically fill up author's E-Mail

```
if {[content withId $contentId get Author_Email] == "" &&
    [lsearch $modifiedAttributes Author_Email] == -1} {
    set email [user withLogin [whoami] get email]
    lappend modifiedAttributes Author_Email $email
}
```

All value callbacks are executed in a Safe Interpreter

- Built-In Security
- Certain commands or procedures can be mapped



The Safe Interpreter offers extensible „built-in“ security

The other side

What's bad? Missing ...

- data typization,
- native XML support,
- Web Services Support,
- customer acceptance (sometimes).

Other CMS-Vendors using TCL

- Vignette
 - Used inline to process pages and access the database

Future

- Infopark's success has also to do with TCL
- Infopark is comitted to TCL in the future as well

CMS and TCL fit perfectly together

Thank you!

CMS Library

- <http://www.infopark.de/> (.com: English language)

CMS Documentation

- White Papers
 - e.g. Scripting, Scalability, XML, Security,
 - Internationalization, Web Services, ...
- Data Sheets
- Complete documentation (TCL-Manual!)

Success stories

- NPS at work

Free 30-days-trial version of NPS

Contact me: Thomas Witt <thomas.witt@infopark.de>