

WakeUpManager

Vortrag zur Studienarbeit

Maximilian Wilhelm
max@rfc2324.org

16.09.2008

Agenda

- Motivation
- Anforderungen
- Architektur
- Benutzerschnittstellen (inkl. Demo)
- Datenhaltung
- Implementation
- Organisatorische Einbindung
- Resumé

Motivation

- Institut für Mathematik
 - Betreibt 110 Mitarbeiter-PCs + 55 Poolrechner
 - 24/7 online
- Arbeitszeit nur $220d * 8h / a$
 - 80% Leerlauf (7000h / a)
 - 73W bis 86W Leerlaufleistung
 - 6W bis 13W Standbyleistung
- Ziel: Rechner bedarfsorientiert starten
- Lösung: Der WakeUpManager

Anforderungen

- Rechner zeitgesteuert verfügbar machen
 - Individuelle Zeitpläne, vom Benutzer änderbar
 - Nicht herunterfahren, wenn Benutzer aktiv!
 - Komfort: Rechner bei Betreten des Büros online
- Rechner manuell startbar
 - Ausserhalb konfigurierter Zeiten
 - Zugriff aus der Ferne
- Organisationsstruktur abbildbar (EIM-M / UPB)
 - Auffinden der Rechner, Verwaltung

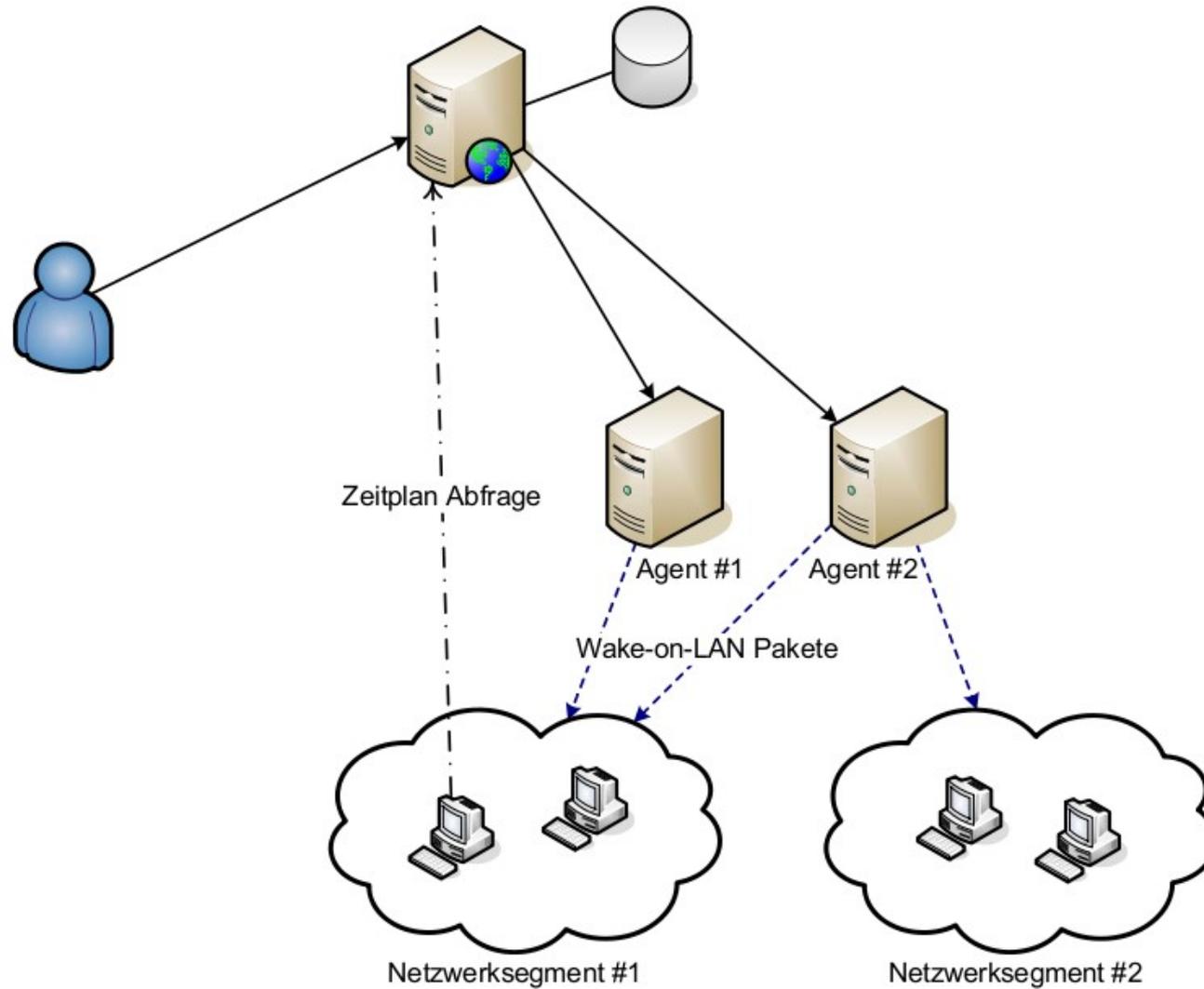
Anforderungen

- Berechtigungssystem
 - Rechte
 - Rechner starten
 - Zeitpläne einsehen/ändern
 - Individuelle Vergabe pro Rechner / Gruppe
- Mandantenfähigkeit (Wunsch)
- Integration in bestehende Netzstruktur
 - Administrative Aufgaben beachten
- Hardwareunterstützung
 - Wake-on-LAN

Architektur

- 3-Tier-Architecture
- Zentraler Server
 - Benutzerschnittstellen
 - Zeitsteuerung
- Boot-Agenten
 - Schnittstelle für zentralen Server
 - Starten Rechner über das Netzwerk
- Klienten
 - Fahren selbstständig herunter

Architekturüberblick



Benutzerschnittstellen

- Webinterface des zentralen Servers
 - Alle Funktionen des WakeUpManagers
 - Standardkonform
 - Mehrsprachig
 - Einheitliches übersichtliches Design
 - Nutzung von Templates und AJAX
- Kommandozeilenprogramm `wum_boot`
 - Rechner starten
 - Mehrsprachig

Vorführung Webinterface

Live Vorführung

Rechner starten, graphisch

WakeUpManager

Rechner starten

- » [Rechner starten](#)
- » [Zeitplan anzeigen](#)
- » [Zeitplan ändern](#)
- » [Aktivierungsstatus](#)

- » [Einstellungen](#)

- » [Über WakeUpManager](#)

Der WakeUpManager bietet die Möglichkeit, einen Rechner bei Bedarf unabhängig von konfigurierten Zeiten zu starten.

Um jetzt einen Rechner über das Netzwerk zu starten, wählen Sie den gewünschten PC aus der Liste aus und klicken Sie auf "Rechner starten."

» Rechner starten

Nur Rechner der Gruppe anzeigen.

» Ergebnis

Host *twoflower.math.uni-paderborn.de* wird gestartet.

Rechner starten, Cmdline

```
max@pandora:~$ echo $LANG
```

```
en_US.UTF-8
```

```
max@pandora:~$ wum_boot twoflower
```

```
Host twoflower.math.uni-paderborn.de will be booted.
```

```
max@pandora:~$ export LANG=de_DE
```

```
max@pandora:~$ wum_boot twoflower
```

```
Rechner twoflower.math.uni-paderborn.de wird gestartet.
```

Datenhaltung

- Relationale Datenbank
 - PostgreSQL
 - Andere RDBMS möglich
- Zentraler Speicher für
 - Netzwerk & Agenten
 - Clientrechner
 - Rechnergruppen / Hierarchie
 - Zeitpläne
 - Berechtigungen

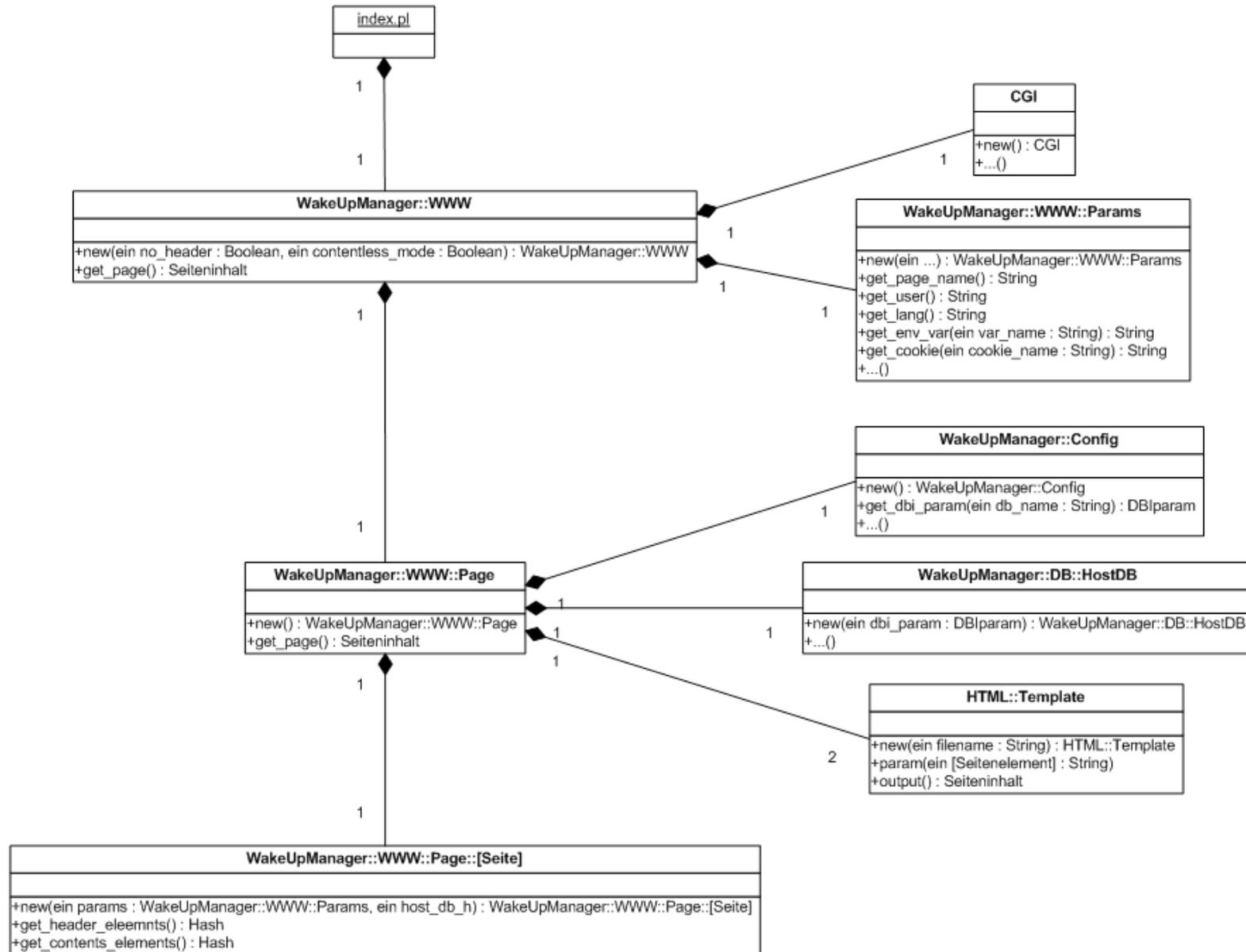
Implementation

- Perl
- Datenbankzugriff via DBI
- Interprozesskommunikation
 - XML-RPC / `Frontier::RPC`
- Webanwendungen
 - Authentifikation via Kerberos
 - CGI
 - `mod_perl`

Implementation des UI

- Mehrschichtenarchitektur
 - Mehrere Abstraktionsschichten
 - Trennung von Logik und Präsentation
- Templating
- Mehrsprachigkeit
- AJAX

Architektur des UI



Implementation RPC

- Server stellt RPC-Dienste für Klienten bereit
- Nutzung von `Frontier::RPC + Wrapper`
 - Weitestgehend unabhängig von RPC Impl.
- Dienste für
 - `wum_shutdown`
 - Wird auf jedem Client ausgeführt
 - Prüft lokale Vorraussetzung und den Zeitplan
 - Fährt Rechner ggf. Herunter
 - `wum_boot`
 - Auf jedem Client installiert

Sicherheit

- Spracheigenschaften
 - `use strict`
 - `use warnings`
 - Taint-Mode
- Verschlüsselte Übertragung (SSL/TLS)
- Testen
 - Unerwartete Eingaben

Organisatorische Einbindung

1. Abbildung der Organisationsstruktur
2. Konfiguration der Netzwerke und Agenten
3. Paketieren und Verteilen der Software
4. Aufnahme und Konfiguration der Clients
5. Festlegen von Benutzerrechten

Resumé

- ✓ System seit über zwei Monaten im Einsatz
 - ✓ ~100 Rechner durch WakeUpManager verwaltet
 - ✓ Keinerlei Beschwerden
 - ✓ Lehretrieb nicht beeinträchtigt
 - ✓ Guter Anklang bei den Benutzern
 - ✓ Bonus: Verringerung des Lärmpegels
- ✓ Funktionale/technische Anforderungen erfüllt
- ✓ Mandantenfähigkeit soweit möglich umgesetzt
- Nutzung der Potentiale des WakeUpManagers
- Alle Ziele im Rahmen der Arbeit erreicht

Fragen?
