

# *Interchange – Die Open Source Onlineshop-Plattform*

*Stefan Hornburg*

*CeBIT 2009, Hannover*



# *History & Motivation*

- 1995: first release of MiniVend
  - 1998: software evaluation for materialboerse.de
  - 2000: Akopia acquired MiniVend
  - 2000: Merged with Tallyman into Interchange
  - 2001: RedHat acquired Akopia
  - 2002: RedHat dropped support for Interchange
  - 2002: foundation of ICDEVGROUP
  - 2007: first booth on Linuxtag in Berlin
- 
-

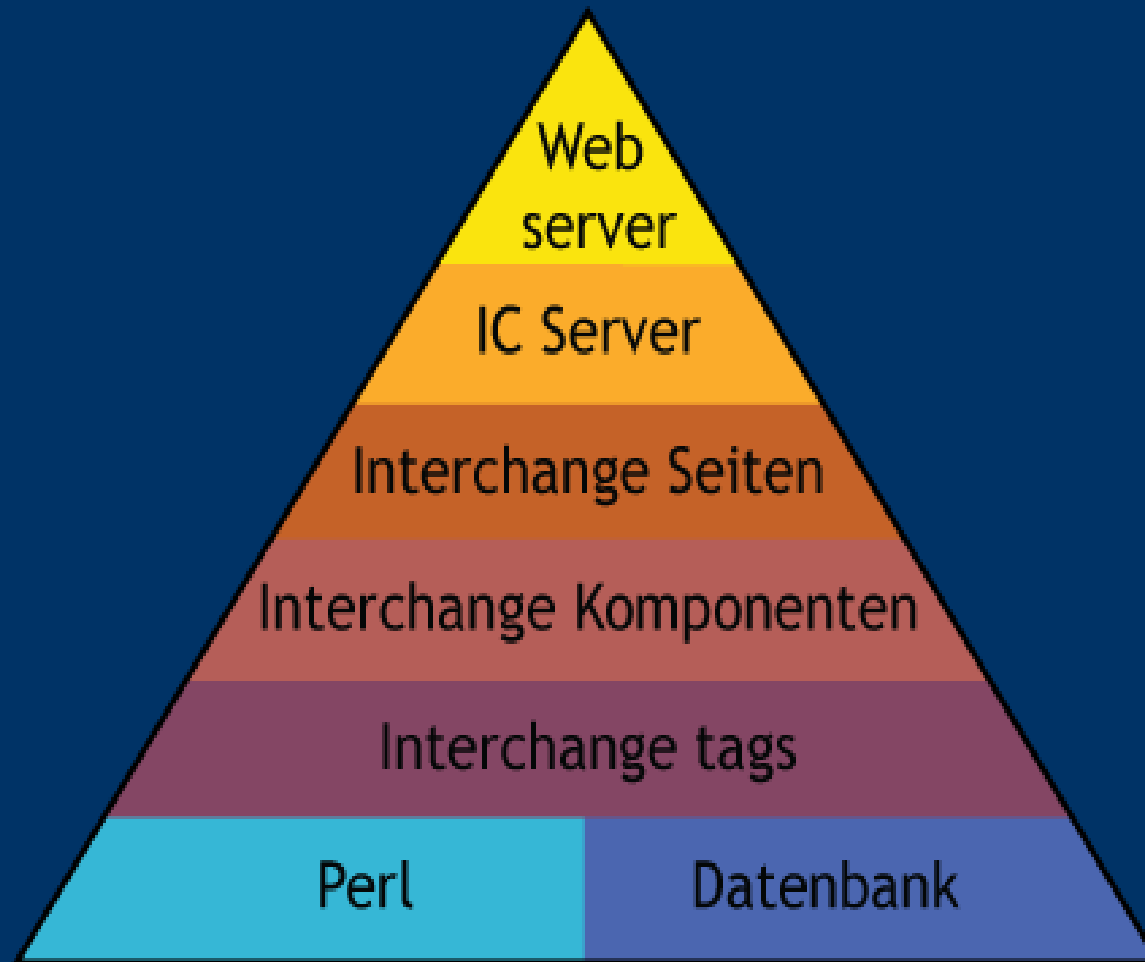
# *Interchange*

- Architektur
- Perl: wie, wo und warum?
- SEO
- Schnittstellen (APIs)
- Entwicklung

# Architektur

- Eigenständiger Server
  - Globale Konfiguration (interchange.cfg)
- Shops (Kataloge)
  - Lokale Konfiguration (catalog.cfg)
- Datenbanken
  - DBI (MySQL, PostgreSQL, Oracle, SQLite, ...)
- Inhalt
  - Seiten, Vorlagen, Komponenten

# *Architektur: Pyramide*



# *Archicecture: Standalone Server*

- Vor- und Nachteile
  - Sicherheit
  - Performanz
    - Threaded Perl
  - Entkopplung
    - Stability pattern, Recursive Proxy, Separate Machine
  - Installation
- Run Modes
  - PreFork
  - PreFork/SingleFork
  - On Demand

# *Architecture: Content*

- HTML with ITL
  - Variables
    - @@MV\_PAGE@@
  - Localization
    - [L]Conference[/L]
  - UserTags
    - [total-cost]
- Embedded Perl
  - Safe compartment
  - UserTags
    - \$Tag->total\_cost()

# *Embedded Perl*

## Sorting Cart by Article Number

```
[calc]
@$Items = sort {$a->{code} cmp $b->{code}}
    @$Items;
[/calc]
```

## Perl Inside UserTag Arguments

```
[assign salestax=`($Tag->subtotal({noformat=>1})
    + $Tag->shipping({noformat=>1}))
    * $Variable->{MWST_SATZ} / 100`]
```



# *Perl: Why, where and how?*

- Perl - why not Java or PHP?
- Perl - where?
- Perl - how?

# *Perl: Warum*

- Aus Mangel an Alternativen 1995
- DBI
- CPAN
  - Business::UPS
  - Business::OnlinePayment

# *Perl: Wo?*

- Server
- UserTags
- ActionMaps
- Filters
- Checks
- Module

# *UserTags: [order] Example*

T-Shirt in den Warenkorb legen:

```
[order item="IC2008"]
```

2 T-Shirts in den Warenkorb legen:

```
[order item="IC2008" quantity="2"]
```

Oranges T-Shirt in den Warenkorb legen:

```
[order code="IC2008" color="orange"]
```

```
[order code="IC2008"  
  form="mv_order_color=orange"]
```

---

---

## *UserTags: [custom-order]*

```
UserTag custom-order Order code quantity
UserTag custom-order AddAttr
UserTag custom-order Routine <<EOR
sub {
    my ($code, $quantity, $opt) = @_ ;

    for (qw/color size/) {
        $opt->{form}
            .= "mv_order_${_}=$opt->{$_}\n";
    }

    return $Tag->order($code, $quantity,
        $opt);
}
EOR
```

---

---

# ActionMap: download

```
ActionMap download <<EOM
sub {
    my ($name, $id) = split('/', shift);

    unless ($Session->{logged_in}) {
        $CGI->{mv_nextpage} = 'index';
        return 1;
    }

    $id =~ s/\.xls$//;

    $Tag->perl({tables => 'uploads'});

    if (($ref = $Db{uploads}->row_hash($id))
        && $ref->{company_idf} == $Scratch->{"company::idf"}) {
        $Tag->deliver({file => $ref->{file}});
        return;
    } else {
        $CGI->{mv_nextpage} = 'intern/upload';
        return 1;
    }
}
EOM
```

---

# Module

## Module laden

```
Require Module Vend::Payment::BusinessPayment  
Require Module Vend::Swish /usr/lib/swish-e/perl
```

---

---

# *Vend::Picture*

```
package Vend::Picture;
```

```
use IO::Scalar;
```

```
use Image::Magick;
```

```
use vars qw($CGI);
```



# *Vend::Picture::magick*

```
sub magick {
    my ($name, $opts) = @_ ;
    my (@args, $image, $msg);

    # first read in image
    if ($opts->{cgi}) {
        tie *IMG, 'IO::Scalar', \ $CGI::file{$name};
        @args = (file => \*IMG);
    } else {
        @args = (filename => $name);
    }

    $image = new Image::Magick;

    if ($msg = $image->Read(@args)) {
        ::logError("failed to read picture: $msg");
        return;
    }

    return $image;
}
```

---

# *Vend::Picture::resize*

```
sub resize {  
    my ($name, $width, $height, $opts) = @_;  
    my ($image, $msg, $size, $outfile);  
  
    $image = magick($name, $opts);
```

```
$outfile = $opts->{outfile} || $name;

# if we want both width and height to fit, we have to crop a bit
if ($width && $height) {

    # Get the size and resize according to it
    my ($orig_width, $orig_height) = (0,0); # $image->[0]->Ping();
    $size = "${width}x$height";
    my $crop = $size;

    $orig_width = $image->[0]->Get('width');
    $orig_height = $image->[0]->Get('height');

    if ($orig_width < $orig_height){ # if landscape
        $size = "${width}x";
    }
    else {
        $size = "x$height";
    }

    $msg = $image->[0]->Thumbnail(geometry => $size);

    $msg = $image->[0]->Set( Gravity => 'Center');

    $msg = $image->[0]->Crop("$crop+0+0");

    return Vend::Picture::write($image, $outfile);
}
```

# *Vend::Picture: Thumbnail Resize*

```
elseif ($width || $height) {
    # shrink it by width or height
    if ($width) {
        $size = "${width}x";
    }
    else{
        $size = "x$height";
    }

    $msg = $image->[0]->Thumbnail(geometry => $size);

    if ($msg) {
        ::logError("Vend::Picture::resize: $msg");
        return;
    }

    return Vend::Picture::write($image, $outfile);
}
```

---

---

# *Vend::Picture: [image-resize]*

```
use Vend::Config;
```

```
use Vend::File;
```

```
Vend::Config::parse_tag('UserTag',  
    'image_resize Order name width height');
```

```
Vend::Config::parse_tag('UserTag',  
    'image_resize AddAttr');
```

```
Vend::Config::parse_tag('UserTag',  
    'image_resize MapRoutine Vend::Picture::resize');
```

```
[image_resize original 200 30]
```

---

---

# *Search Engine Optimization (SEO)*

- Erkennung von Suchrobotern
- Saubere Hyperlinks
  - ActionMaps
  - Paging
- Dynamische Erzeugung von <title> und <meta>
- XING-Gruppe
  - <https://www.xing.com/net/sumo/>

# *Interchange APIs*

- Bezahlung
- Versand
- Optionen
- Datenbank

# *Development*

- Roadmap
  - UTF8
  - WellWell Template
  - Gettext
- Community
  - Organization
  - Contributions



# *Roadmap*

- Store Template (WellWell)
- Full UTF-8 support
- Gettext support



# *UTF-8 Support*

- Interchange 5.6.0
- Regexes in Embedded Perl



# *ICDEVGROUP*

- <http://www.icdevgroup.org/>
- Developers from all around the globe
  - Croatia, Germany, Great Britain, Greece, Netherlands, New Zealand, Slovenia, USA

# *Community*

- Mailingliste
- IRC
- Bug Tracker (RT)
- LinuxTag 2009

