

Skalierung



Renée Bäcker

Skalierung



Renée Bäcker

BESSERE PERFORMANZ DURCH NICHT-BLOCKIERENDE WEBANWENDUNGEN

Renée Bäcker

@reneeb_perl
info@perl-academy.de



Perl-Trainer
-Programmierer
-Magazin

OTRS-Dienstleistungen

Renée Bäcker

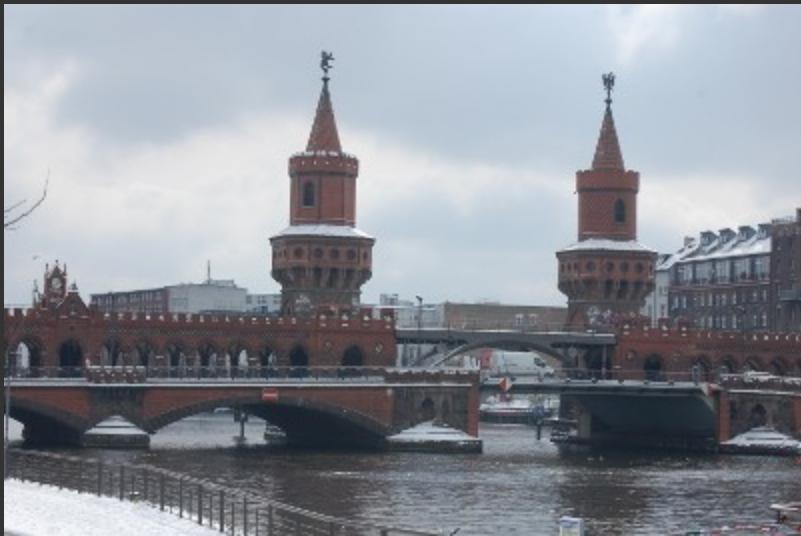
@reneeb_perl
info@perl-academy.de



Seit 2002...
... täglich 10-12 Stunden
Perl
... ganz gerne auf
Konferenzen











Hash-Variante wechseln und bei einem Attribut eben einen anderen Namen bereitstellen.

```
has test2 => (
    is => 'rw',
    isa => 'TestKlasse2',
    handles => {
        tester => 'test',
    },
);
```

Eine vierte Variante gibt es noch, wenn handles einen Namen einer Rolle übergeben wird. Auf Rollen gehe ich in einer späteren Folge des Moose Tutorials ein.

Wurde das Attribut schonmal gesetzt? Diese Frage stellt man sich hin und wieder. Mit dem Parameter predicate kann man einen Subroutinennamen vergeben.

```
has date => (
    is => 'rw',
    isa => 'Str|Undef',
    predicate => 'has_date',
```



PerlyBook

fetch complete release

[EPUB](#)

[MOBI](#)

POD: Pod On Demand.

Pod → EBook

meta::cpan

[Search the CPAN](#)

[I'm Feeling Lucky](#)

PerlyBook

fetch complete release
[EPUB](#) [MOBI](#)

POD: Pod On Demand.



Explore the Map of CPAN by zooming,
panning and clicking with your mouse.

You can search too.

[Need more help?](#)

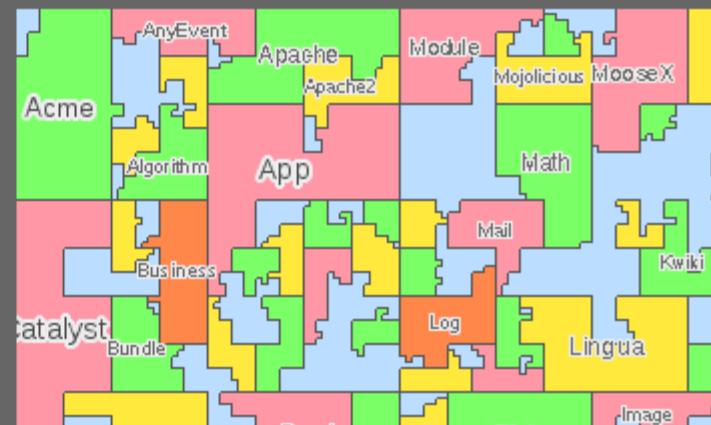
[Sightseeing tours](#)

[Watch the movie](#)



78. [Renee Baecker](#)

- On MetaCPAN: <https://metacpan.org/author/RENEEB>
- On Github: <https://github.com/reneeb>
- Blog: <http://perl-services.de>
- Rank: 232
- [Followers](#): 28
- [Public Repos](#): 94
- Public Gists: 0
- Languages:
 - Perl (85)
 - JavaScript (1)
- Company: Perl-Services.de
- [Coderwall](#) Badges: 6



Pod → EBook

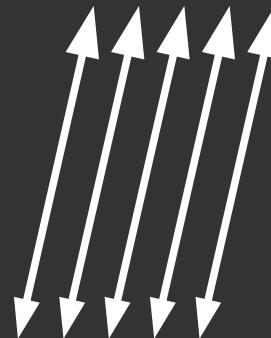
meta::cpan

|

Search the CPAN

I'm Feeling Lucky

HTTP-Request
→ Version, MANIFEST



HTTP-Requests
→ Pod

PerlyBook

fetch complete release
[EPUB](#) [MOBI](#)

POD: Pod On Demand.

Dispatche HTTP-Request

Dispatche HTTP-Request
Hole MANIFEST der Distribution

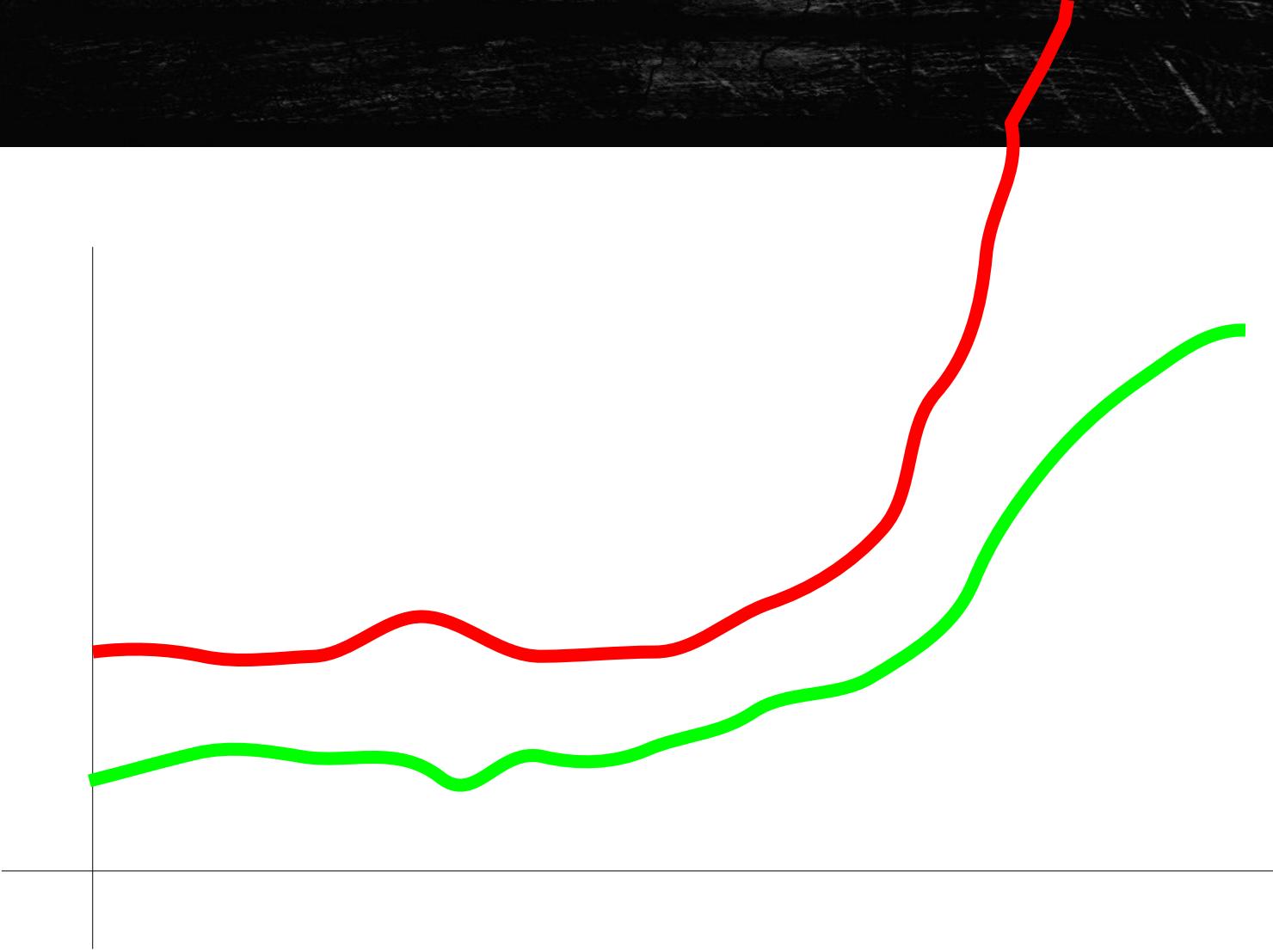
Dispatche HTTP-Request
Hole MANIFEST der Distribution
Für jede Datei im MANIFEST

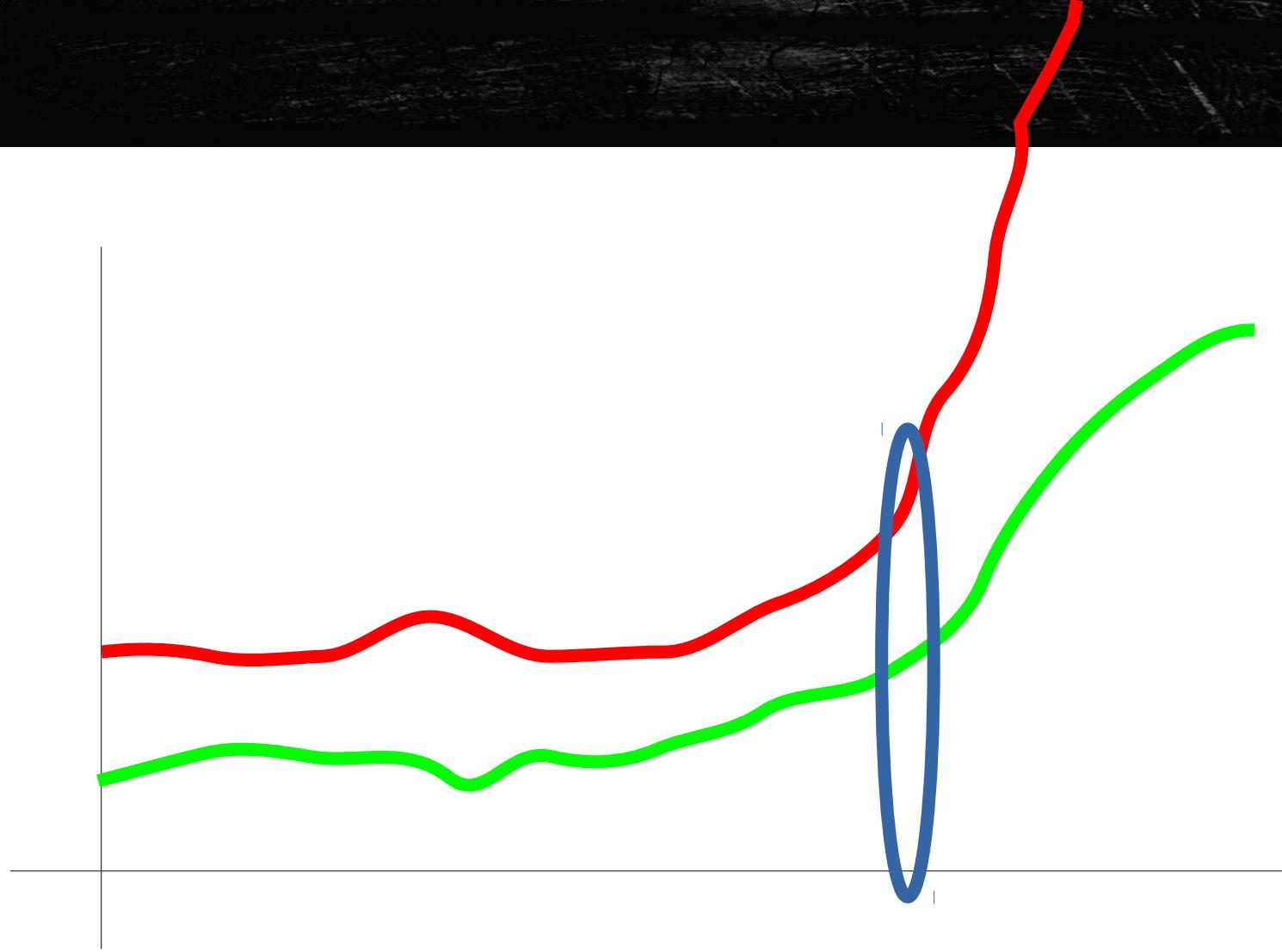
Dispatche HTTP-Request
Hole MANIFEST der Distribution
Für jede Datei im MANIFEST
 Hole Pod

Dispatche HTTP-Request
Hole MANIFEST der Distribution
Für jede Datei im MANIFEST
 Hole Pod
 Speichere Pod

Dispatche HTTP-Request
Hole MANIFEST der Distribution
Für jede Datei im MANIFEST
 Hole Pod
 Speichere Pod
Umwandeln jeder Pod-Datei in XHTML

Dispatche HTTP-Request
Hole MANIFEST der Distribution
Für jede Datei im MANIFEST
 Hole Pod
 Speichere Pod
Umwandeln jeder Pod-Datei in XHTML
Ausliefern EPUB

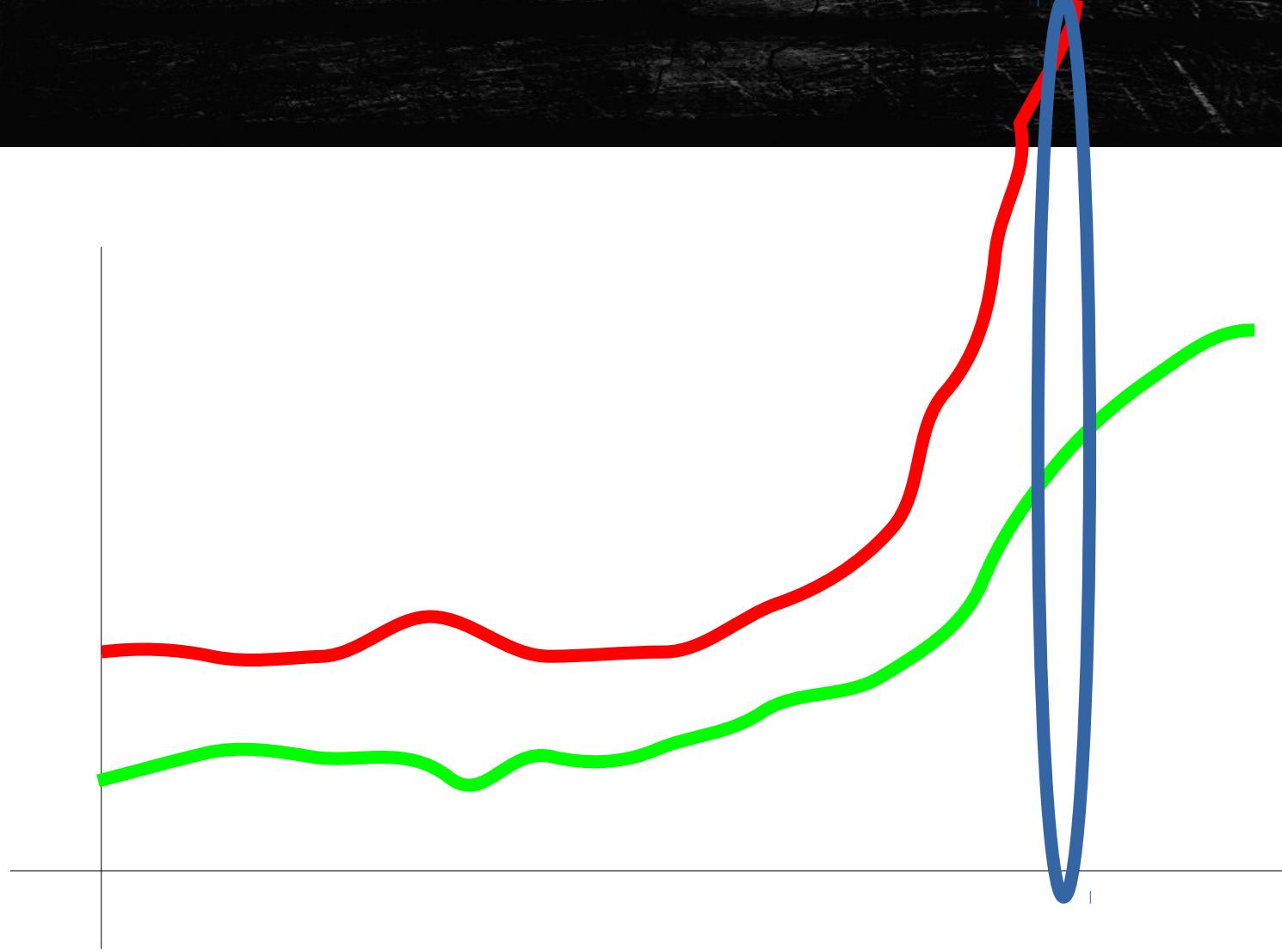




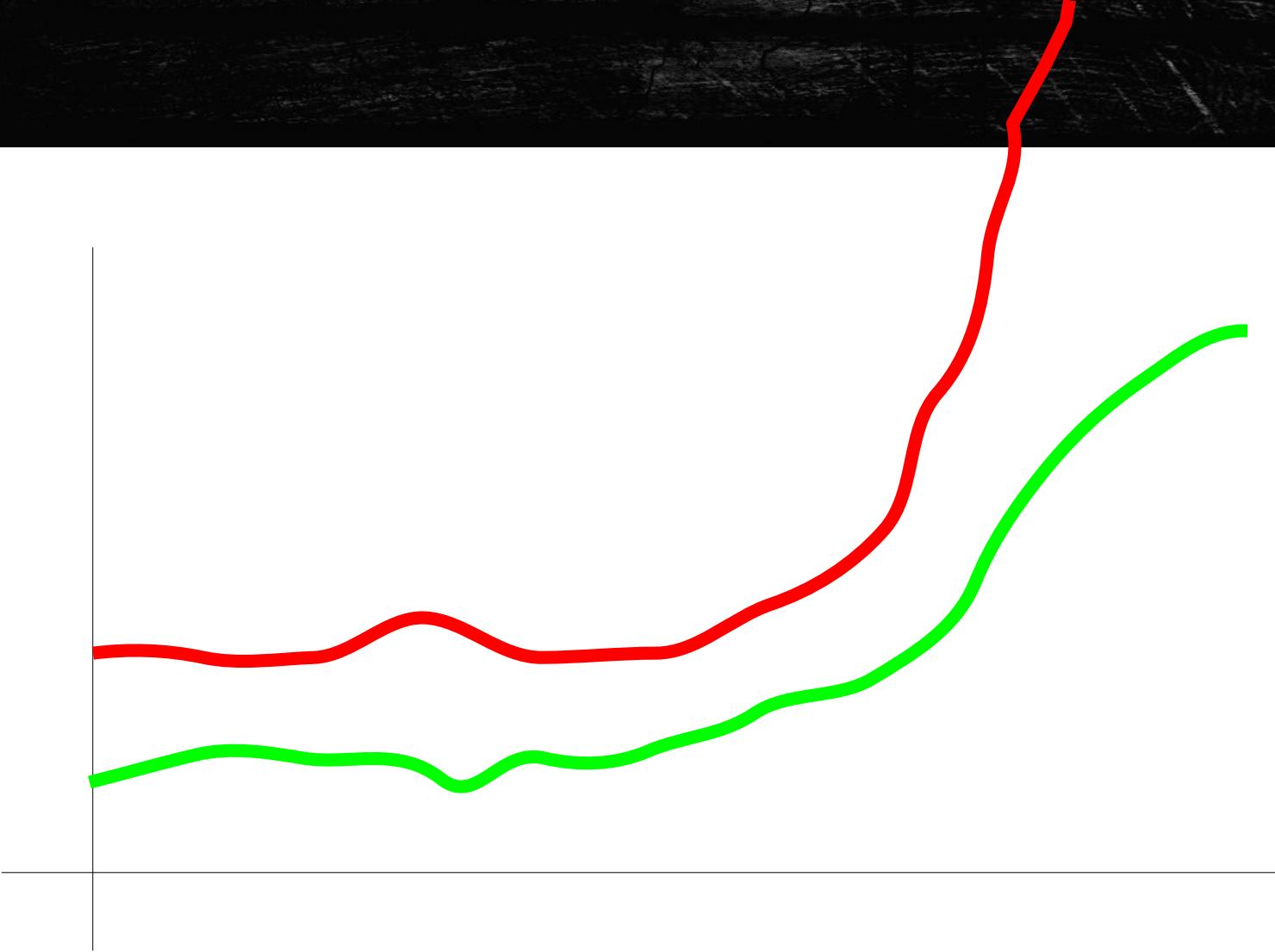
Gabor Szabo

→ 11 nützliche

Perlseiten

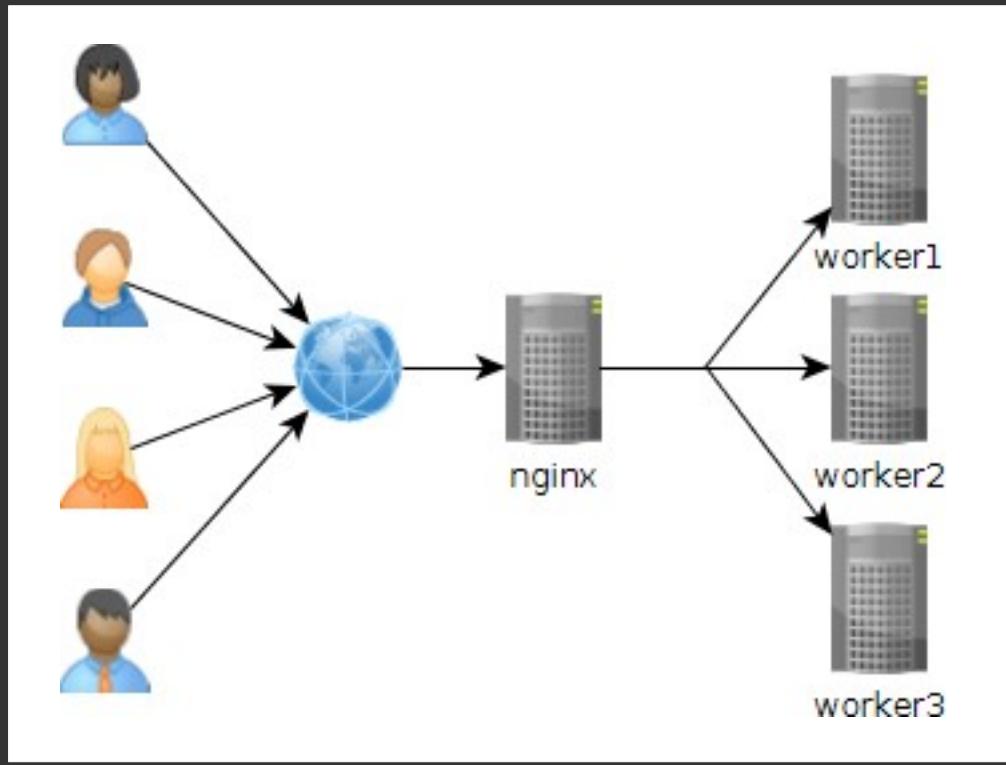


Integration in MetaCPAN

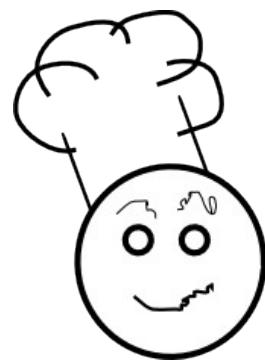




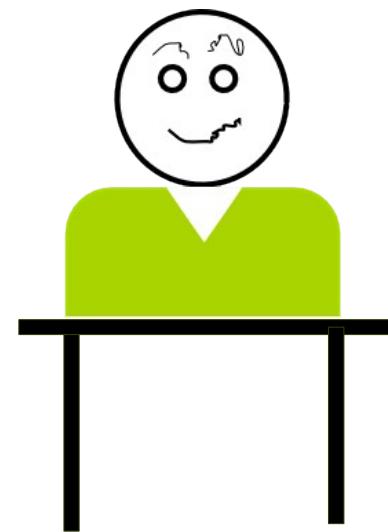
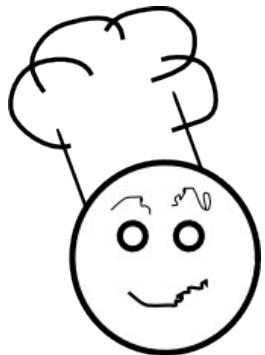




Im Restaurant



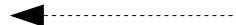
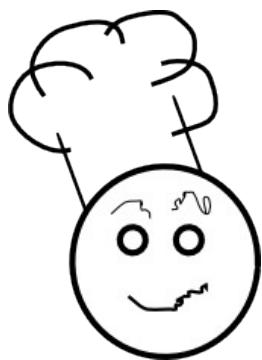
Im Restaurant



Im Restaurant



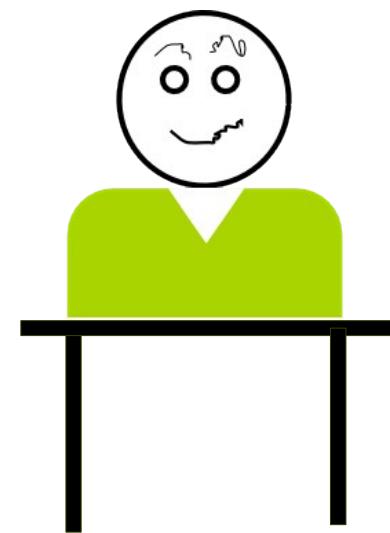
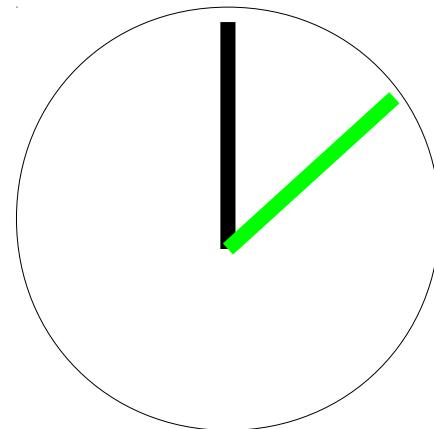
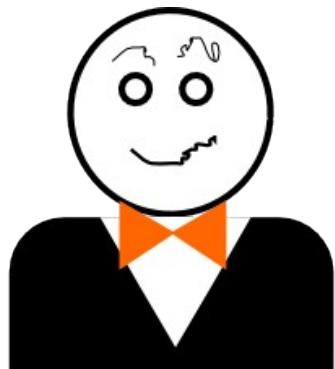
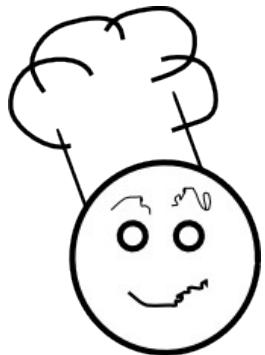
Im Restaurant



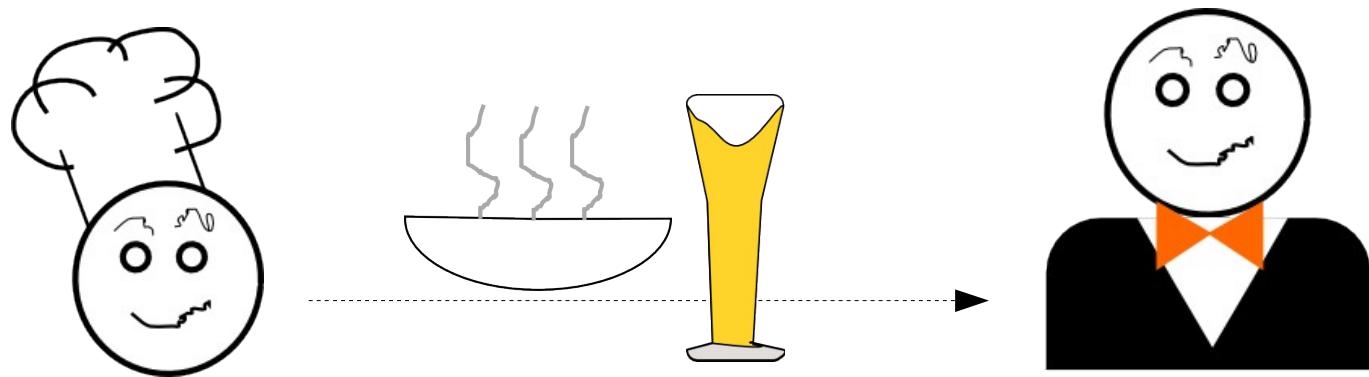
1 Apfelschorle
1 Kartoffelsuppe



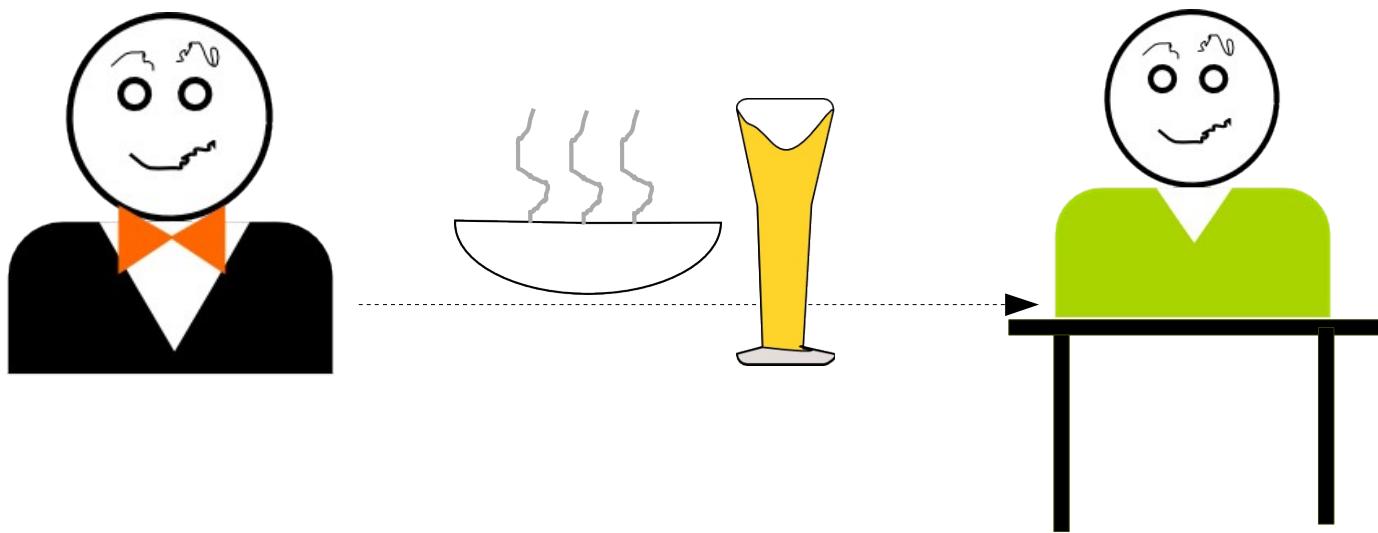
Im Restaurant



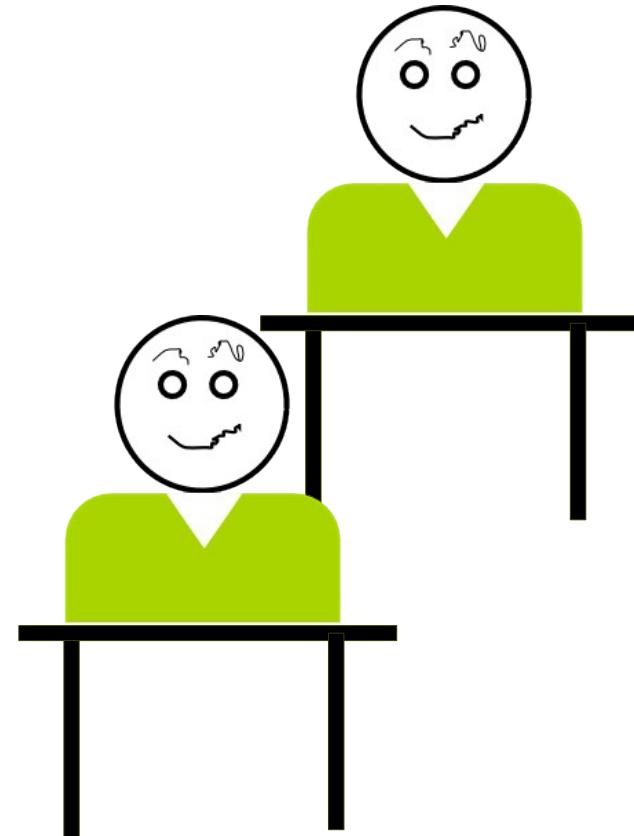
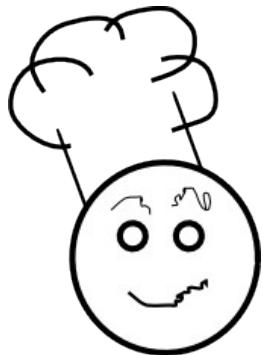
Im Restaurant



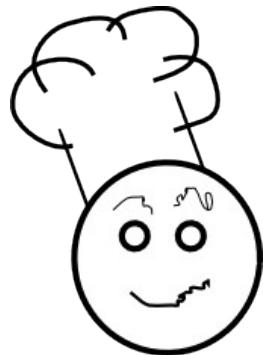
Im Restaurant



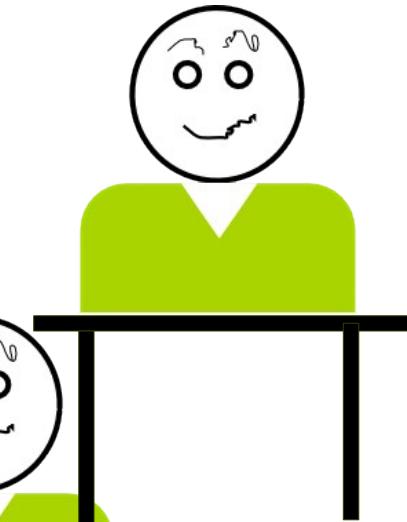
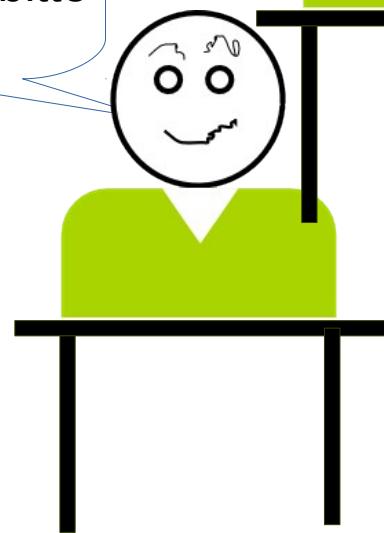
Im Restaurant



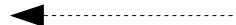
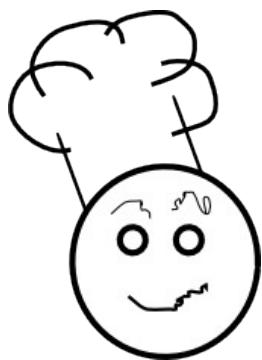
Im Restaurant



1 Apfelschorle und
1 Kartoffelsuppe, bitte



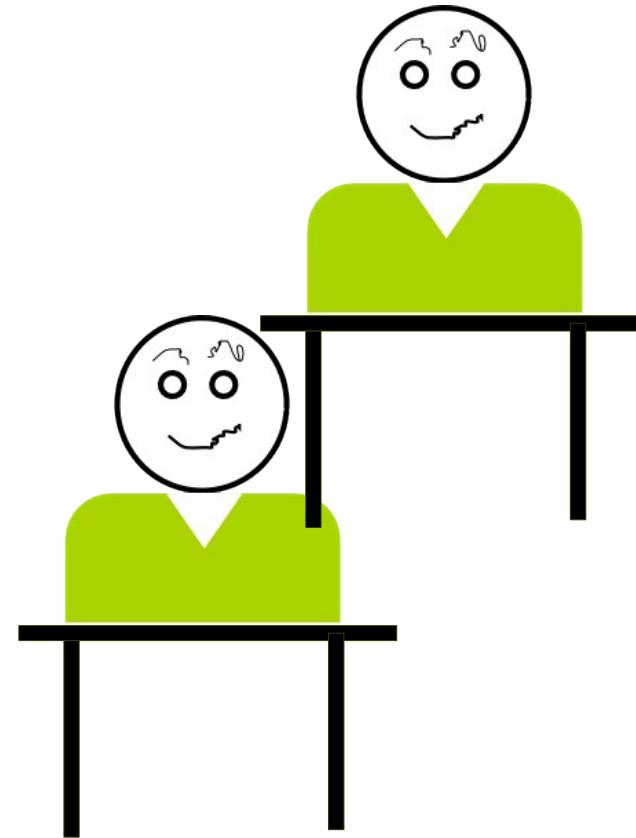
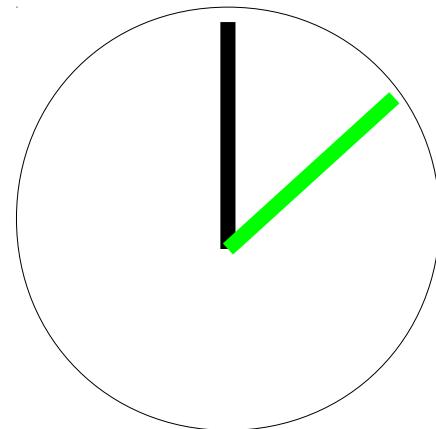
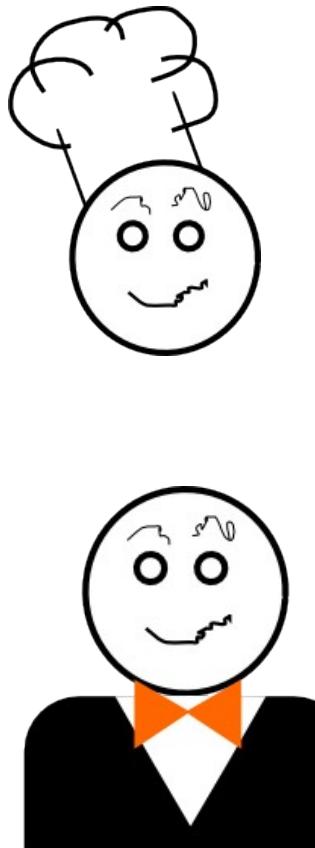
Im Restaurant



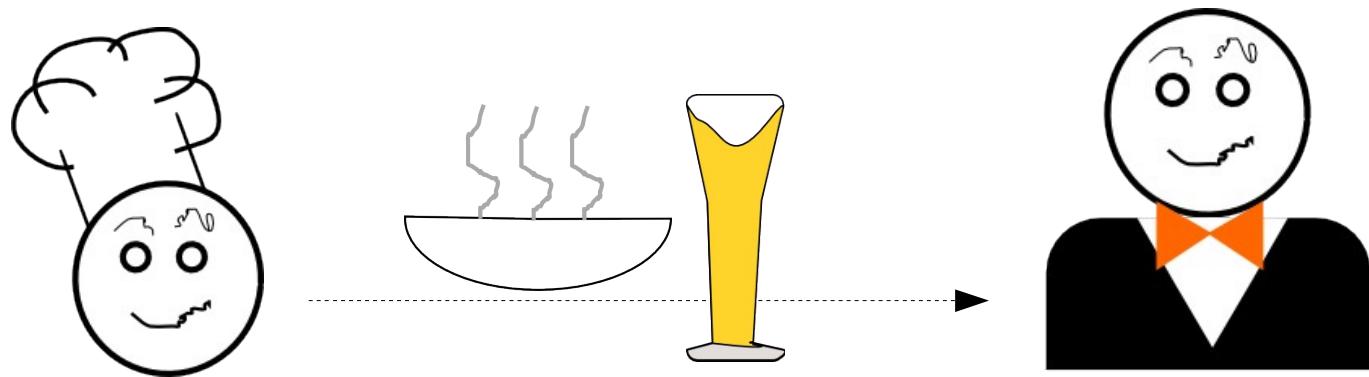
1 Apfelschorle
1 Kartoffelsuppe



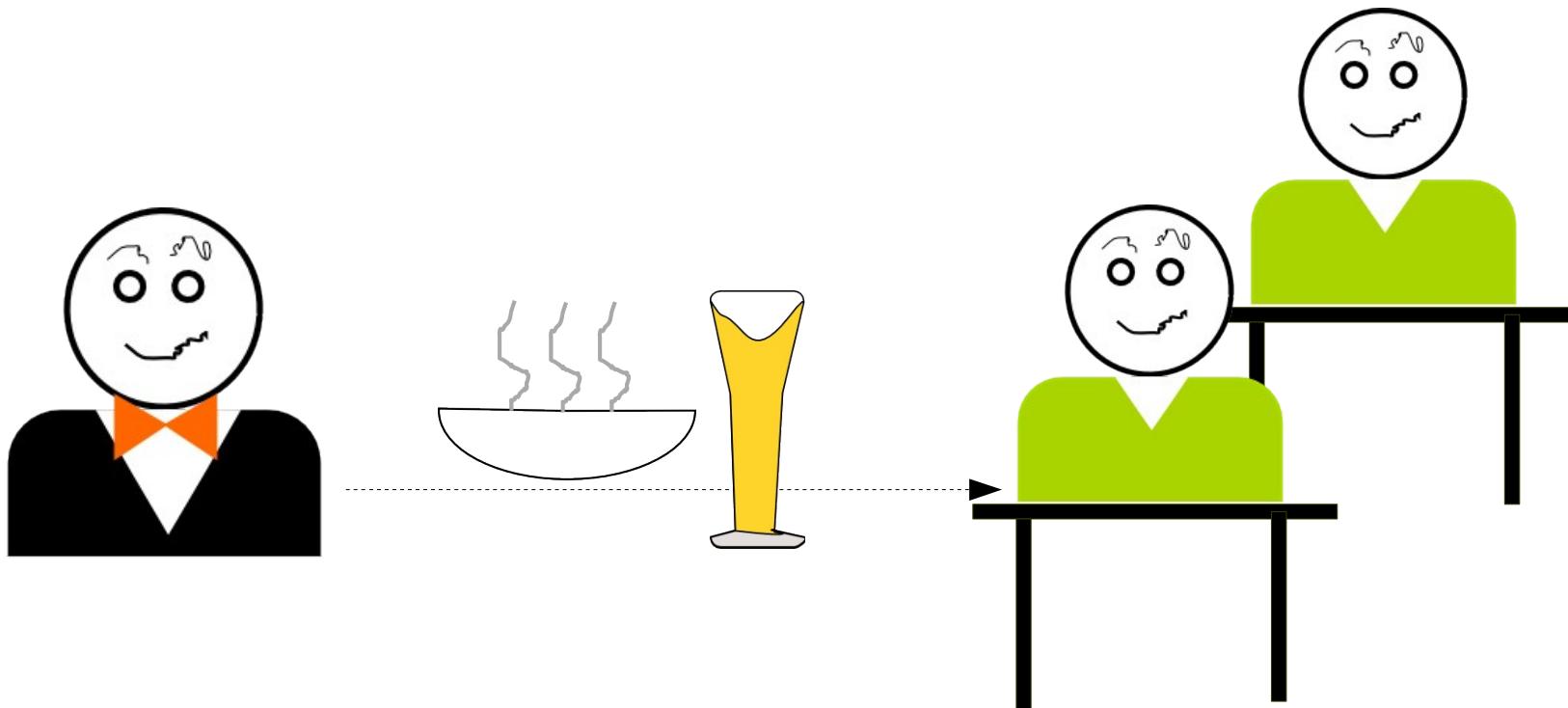
Im Restaurant



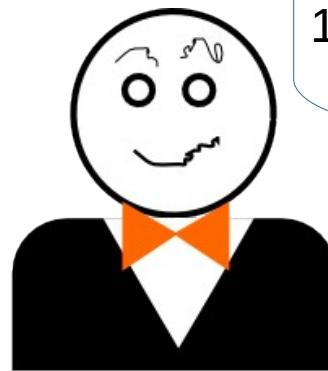
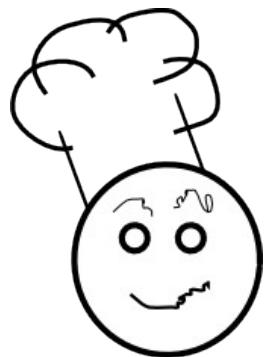
Im Restaurant



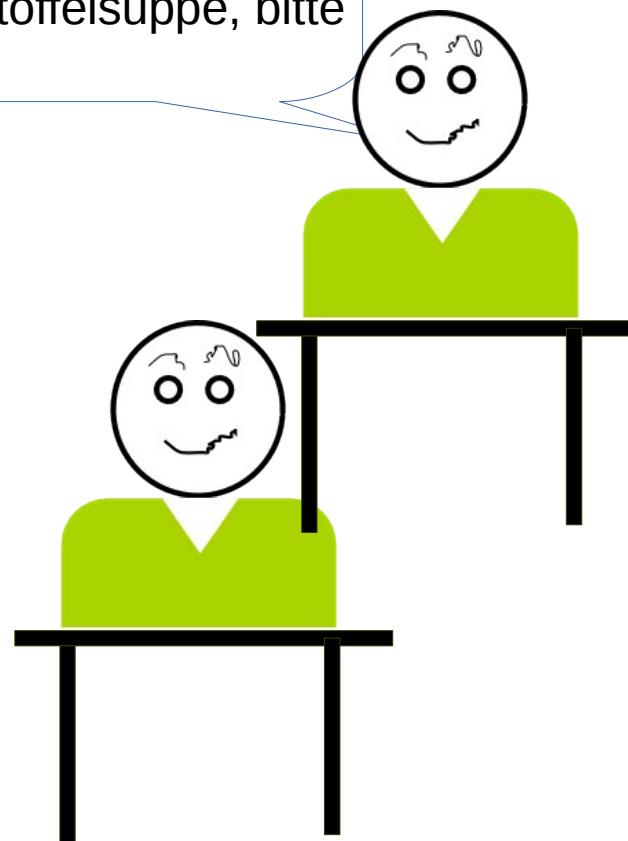
Im Restaurant



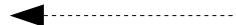
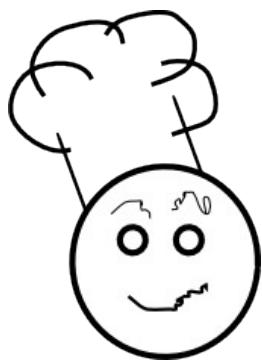
Im Restaurant



1 Apfelschorle und
1 Kartoffelsuppe, bitte



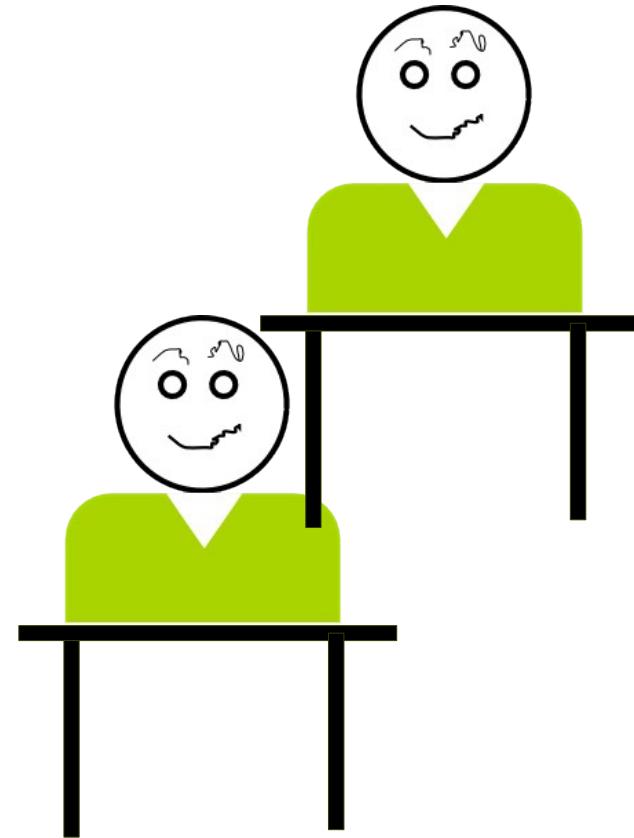
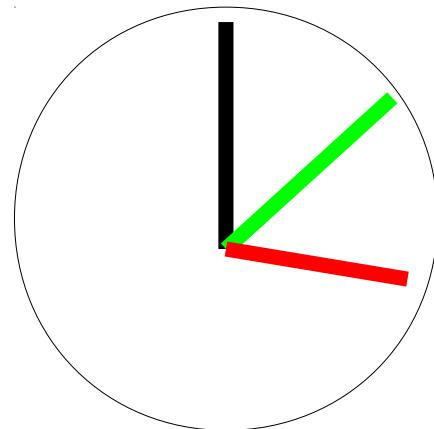
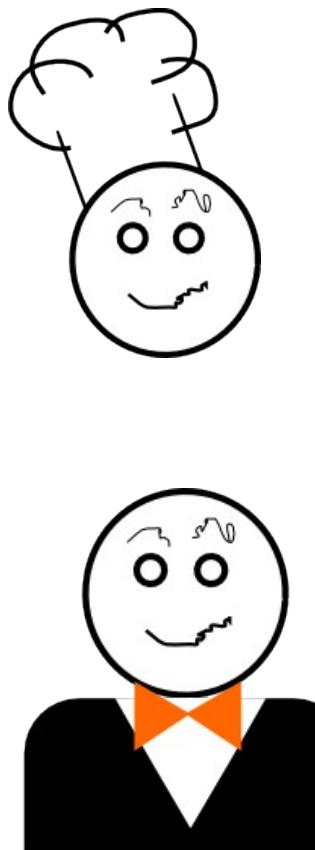
Im Restaurant



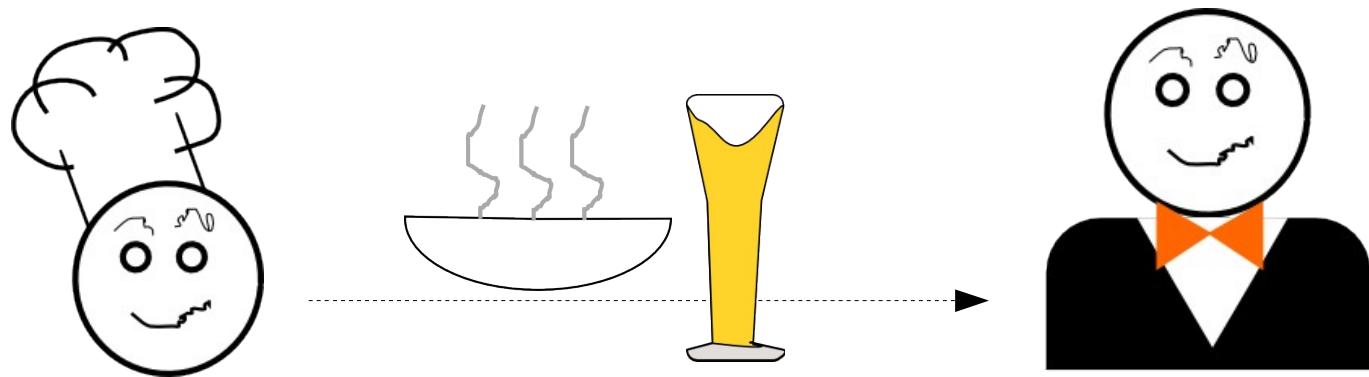
1 Apfelschorle
1 Kartoffelsuppe



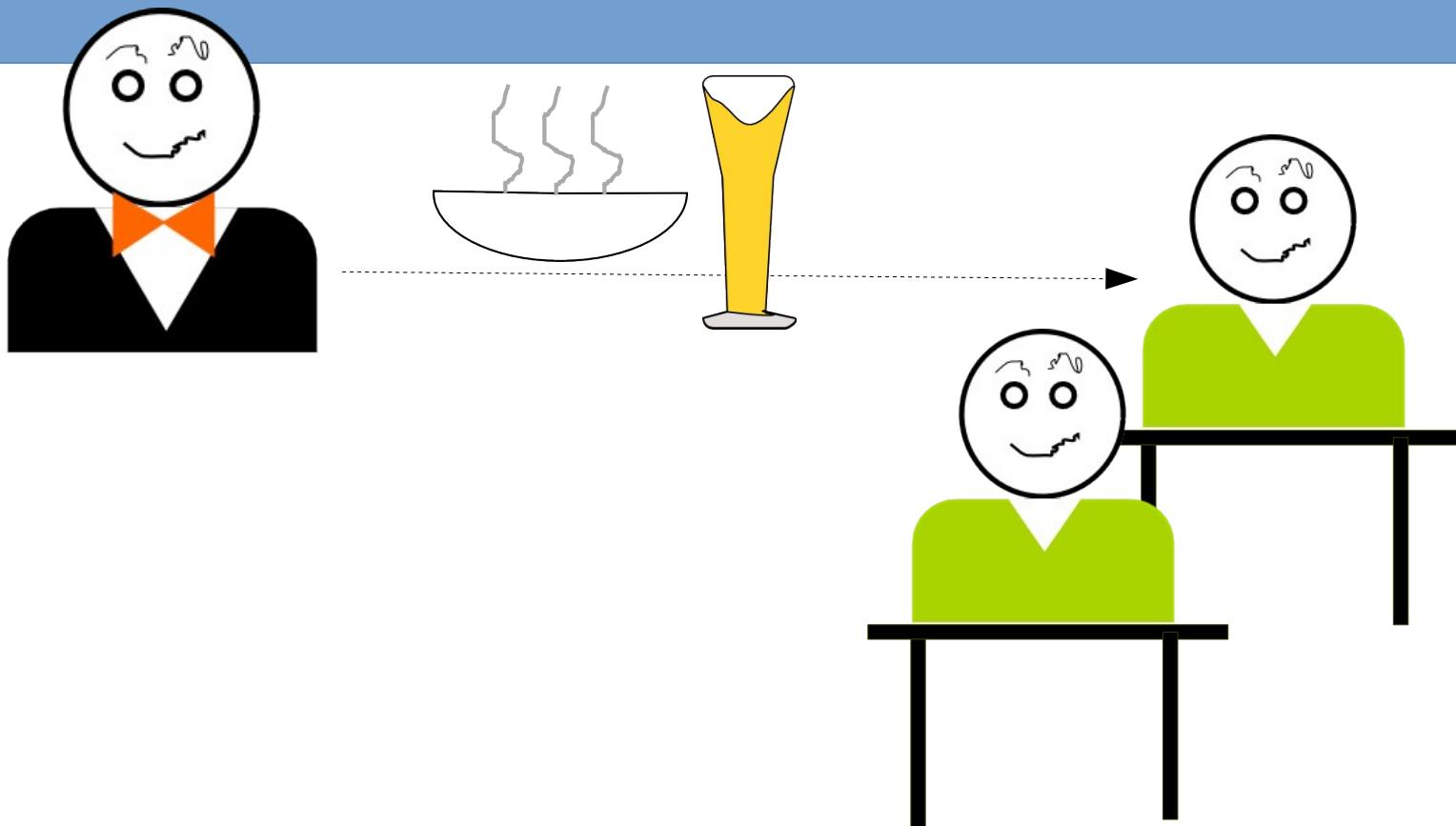
Im Restaurant



Im Restaurant



Im Restaurant



Problem

Lösungsansätze

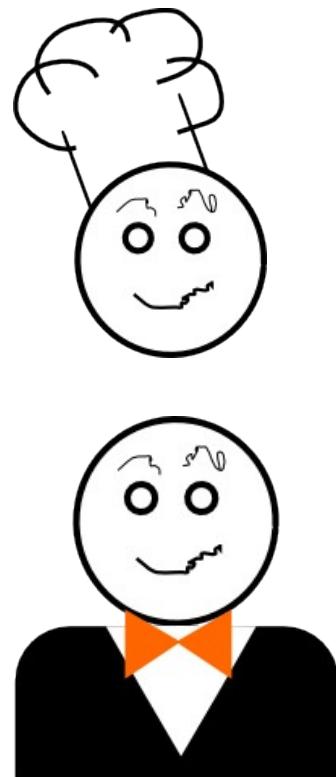
Caching

Caching



Mehr
Hardware

Im Restaurant



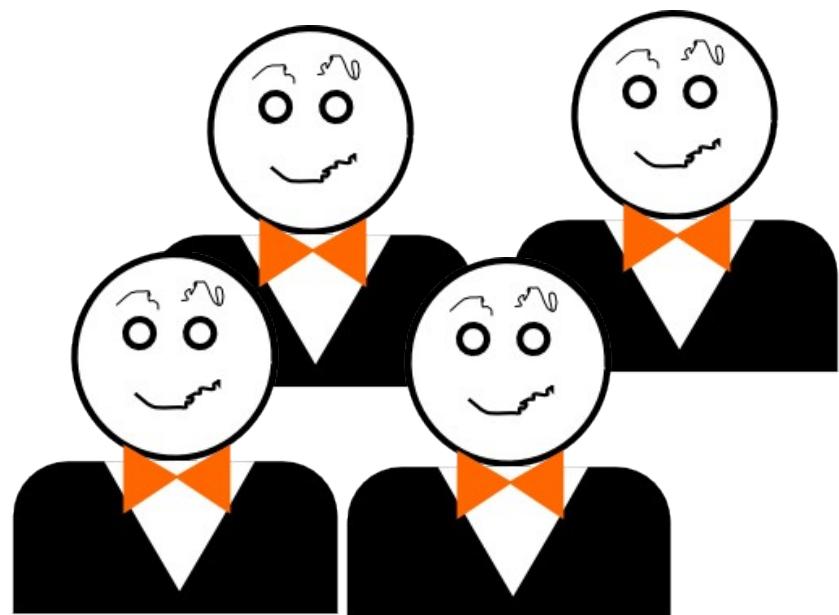
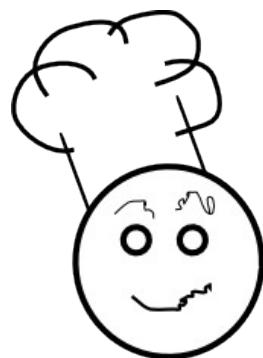


Mehr
Hardware

χ

Mehr
Arbeiter

Im Restaurant





Mehr
Arbeiter

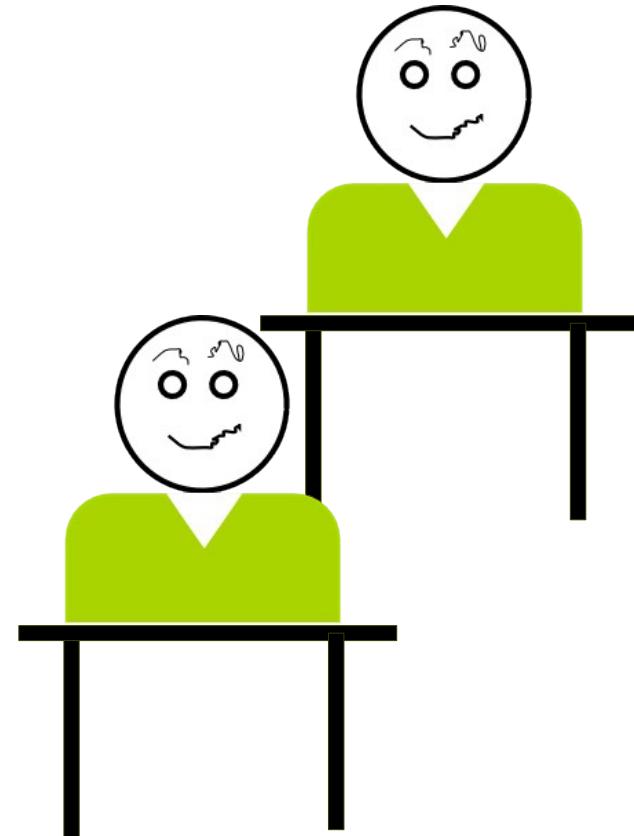
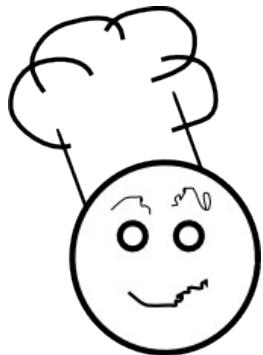
χ

Joel Berger

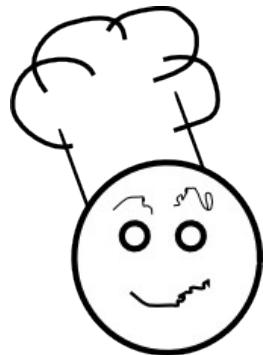
→ blogs.perl.org

Non-Blocking Webapps

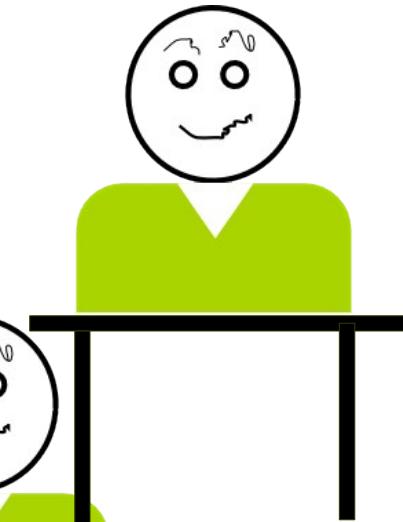
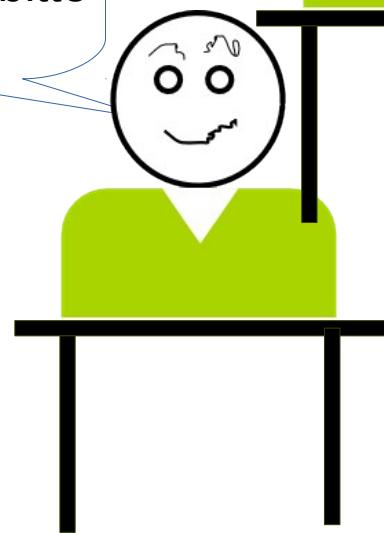
Im Restaurant



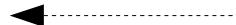
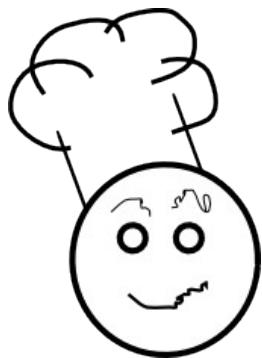
Im Restaurant



1 Apfelschorle und
1 Kartoffelsuppe, bitte



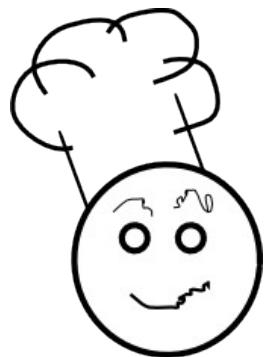
Im Restaurant



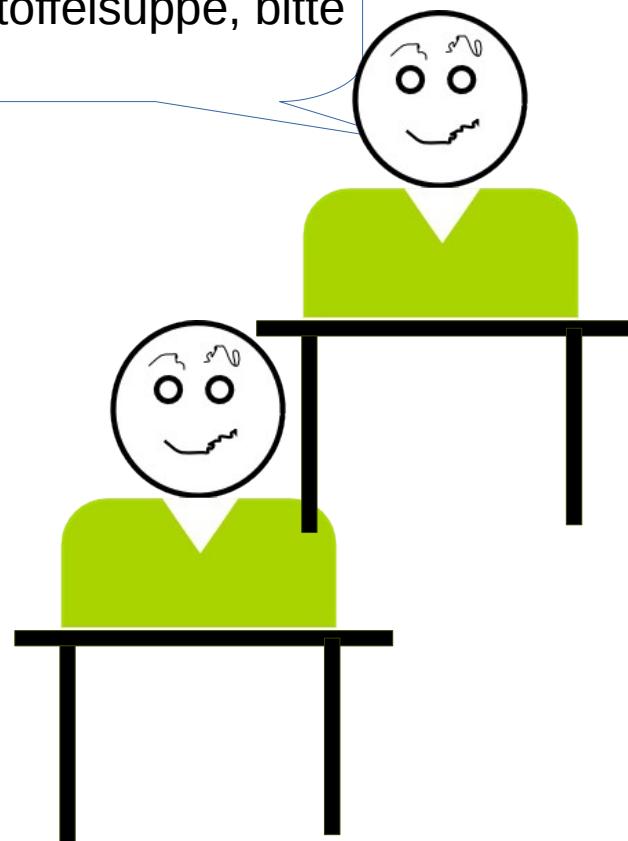
1 Apfelschorle
1 Kartoffelsuppe



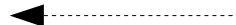
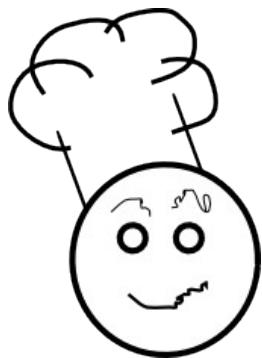
Im Restaurant



1 Apfelschorle und
1 Kartoffelsuppe, bitte



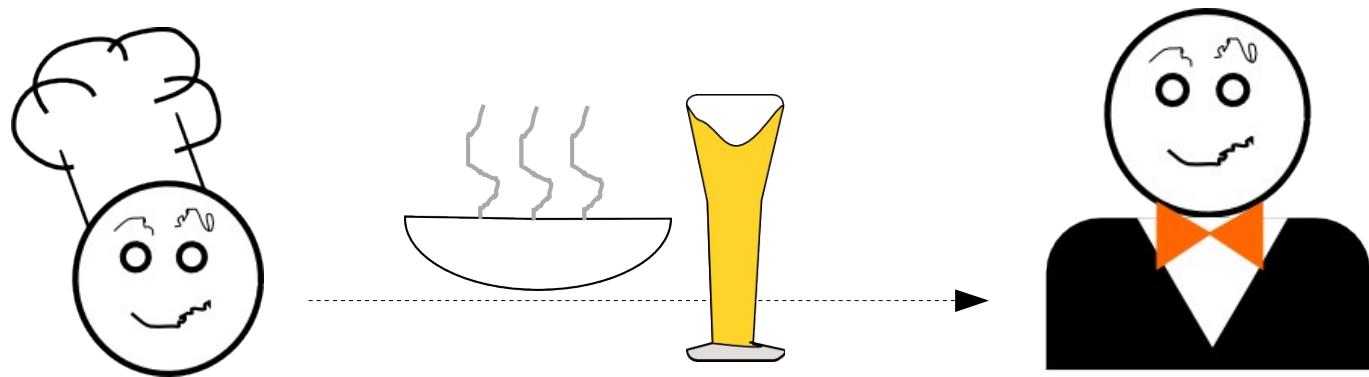
Im Restaurant



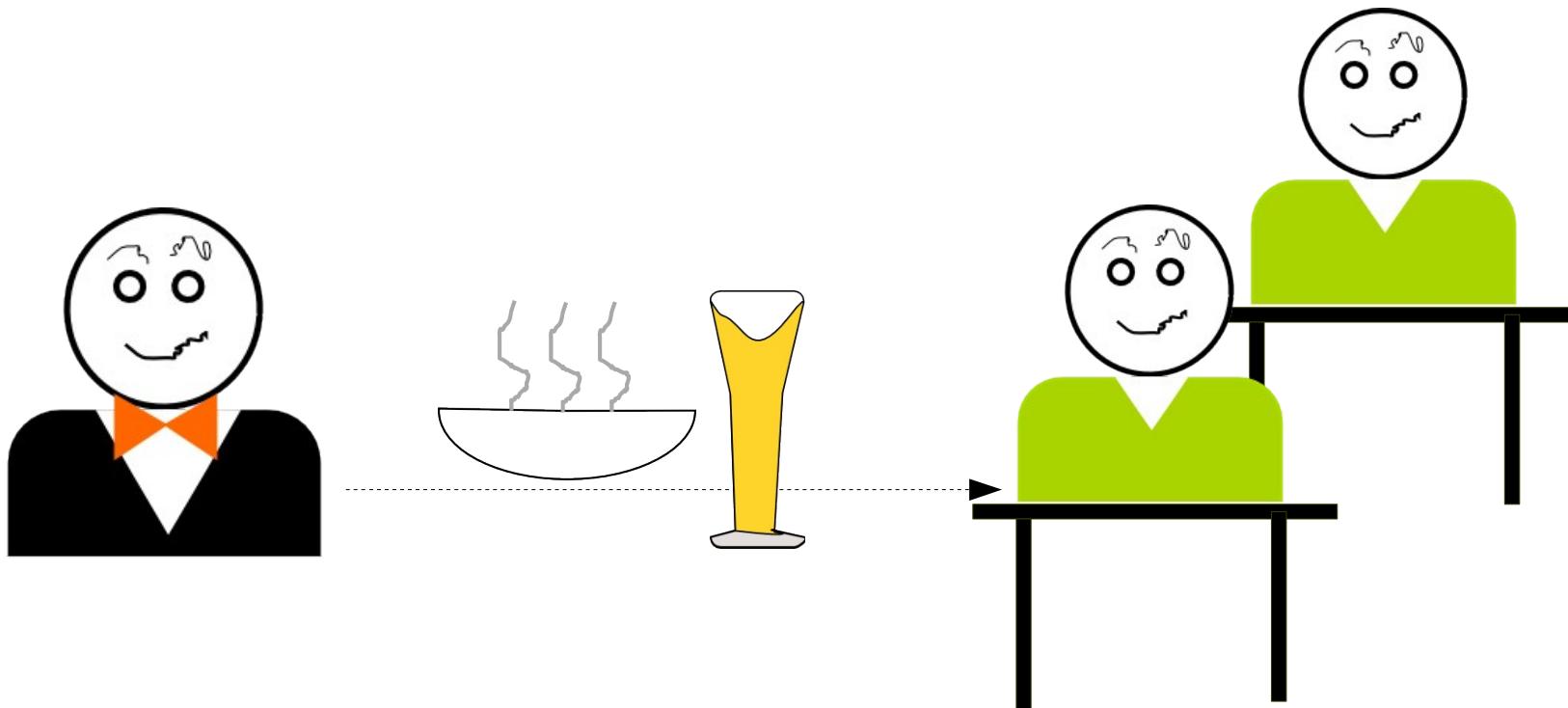
1 Apfelschorle
1 Kartoffelsuppe



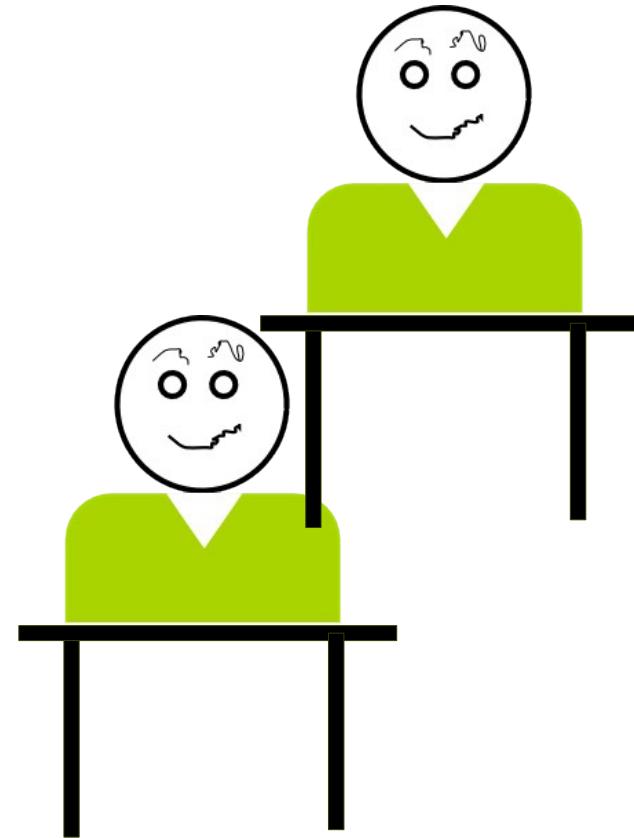
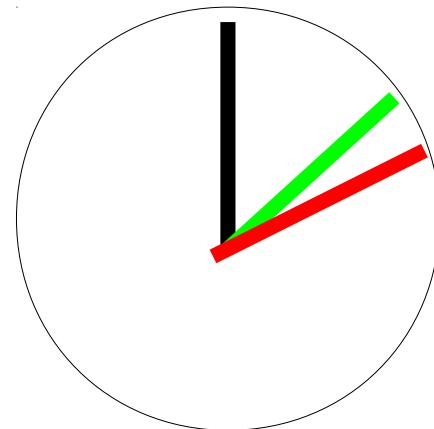
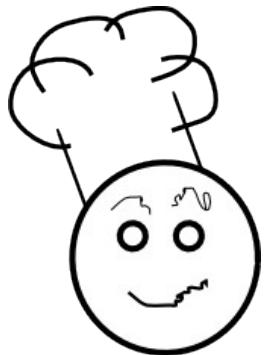
Im Restaurant



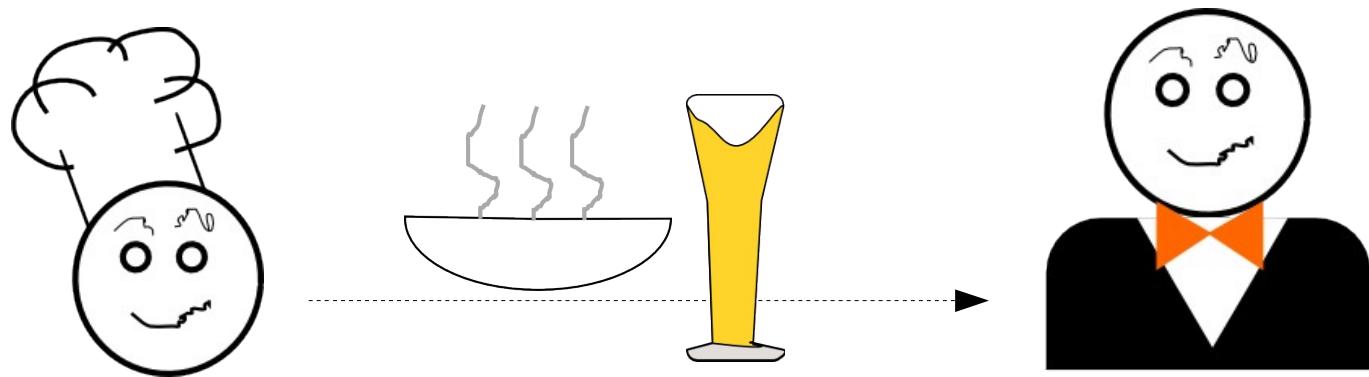
Im Restaurant



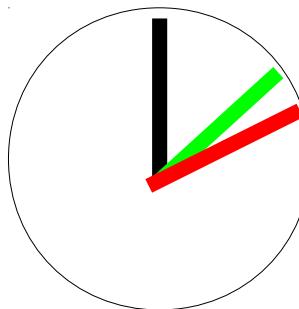
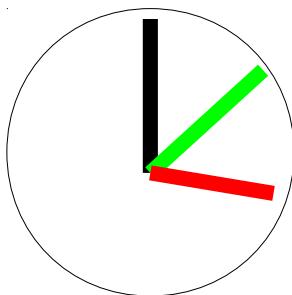
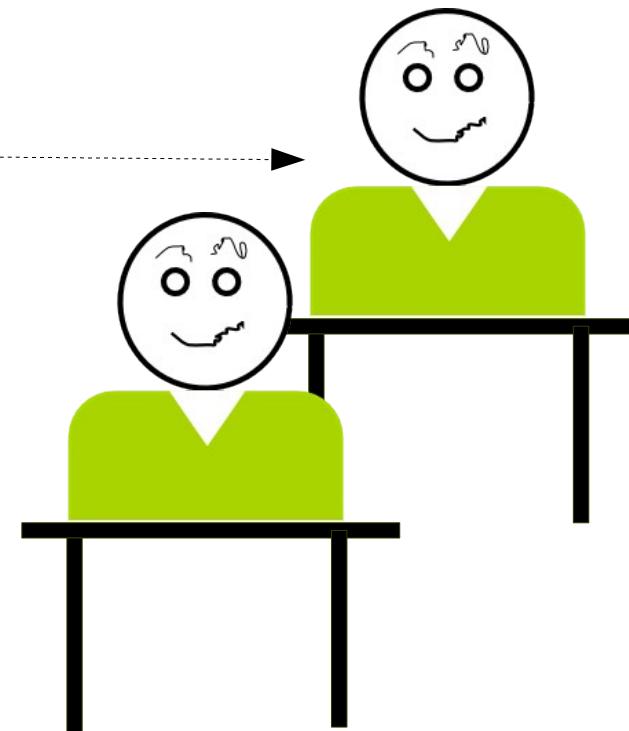
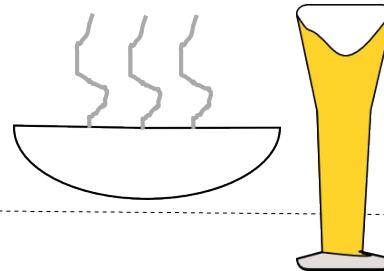
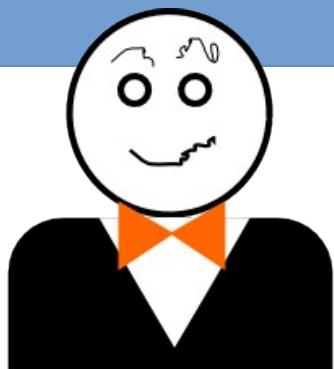
Im Restaurant



Im Restaurant



Im Restaurant



Futures/
Promises

Mojolicious

All in One

Mojolicious::Lite / Mojolicious

```
use Mojolicious::Lite;

get '/' => sub {
    my $self = shift;
    $self->render( text => 'hello world' );
};

app->start;
```

```
$ morbo codetalks.psgi
[Thu Oct 09 07:30:44 2014] [info] Listening at
    "http://*:3000".
Server available at http://127.0.0.1:3000.
```

```
$ wget --quiet -O - http://localhost:3000  
hello world
```

```
get '/event/:name' => sub {
    my $self = shift;
    my $name = $self->param('name');
    $self->render( text => "hello $name" );
};
```

```
$ wget --quiet -O - http://localhost:3000/event/code  
hello code
```

```
get '/user/:name' => sub {
    my $self = shift;
    $self->render( 'user' );
};
```

DATA
@@user.html.ep

```
<html><body>hello <%= $name %></body></html>
```

```
$ wget --quiet -O - http://localhost:3000/user/reneebe  
<html><body>hello reneebe</body></html>
```

```
get '/event/:name' => sub {
    my $self = shift;
    my $name = $self->param('name');
    $self->render( text => "hello $name" );
};
```

```
get '/event/:name' => sub {
    my $self = shift;
    my $name = $self->param('name');
    $self->render( json => {txt =>"hello $name"} );
};
```

```
$ wget --quiet -O - http://localhost:3000/event/code  
{"txt": "hello code"}
```

Initiale Version

Dispatche HTTP-Request
Hole MANIFEST der Distribution
Für jede Datei im MANIFEST
 Hole Pod
 Speichere Pod
Umwandeln jeder Pod-Datei in XHTML
Ausliefern EPUB

```
use Mojo::UserAgent;

my $ua      = Mojo::UserAgent->new;
my $base    = 'http://api.metacpan.org/v0/';
my $perl    = '/SHAY/perl-5.20.1/';
my $source  = $base . 'source' . $perl;
my $request = $ua->get( $source . 'MANIFEST' );
my $manifest = $request->res->body;
my @files   = split /\n/, $manifest;
```

```
my $pod = $base . 'pod' . $perl;
my @pods;
```

FILE:

```
for my $file ( @files ) {
    my $pod_request = $ua->get(
        $pod . $file . '?content-type=text/x-pod' );
    my $pod_text     = $pod_request->res->body;

    next FILE if !$pod_text;
    next FILE if $pod_text !~ /^=head/m;

    push @pods, $pod_text;
}
```

```
my $publisher = EPublisher->new(  
    %config,  
);  
  
my $file = $publisher->convert( 'EPUB', @pods );  
$self->render_file( $file );
```

PerlyBook 2.0

```
use Mojo::UserAgent;

my $ua      = Mojo::UserAgent->new;
my $base    = 'http://api.metacpan.org/v0/';
my $perl    = '/SHAY/perl-5.20.1/';
my $source  = $base . 'source' . $perl;
my $request = $ua->get( $source . 'MANIFEST' );
my $manifest = $request->res->body;
my @files   = split /\n/, $manifest;
```

```
$self->render_later;

my $delay = Mojo::IOLoop->delay;
$delay->on( finish => sub {
    my ($delay, @txs) = @_;
    my @pods;
    FILE:
    for my $file ( @files ) {
        # ...
    }
    $self->render_file( $file );
} );

my @urls = map{
    $pod . $_ . '?content-type=text/x-pod'
}@files;
$ua->get( $_ => $delay->begin ) for @urls;
```

```
$self->render_later;

my $delay = Mojo::IOLoop->delay;
$delay->on( finish => sub {
    my ($delay, @txs) = @_;
    my @pods;
    FILE:
    for my $file ( @files ) {
        # ...
    }
    $self->render_file( $file );
});

my @urls = map{
    $pod . $_ . '?content-type=text/x-pod'
}@files;
$ua->get( $_ => $delay->begin ) for @urls;
```

```
$self->render_later;

my $delay = Mojo::IOLoop->delay;
$delay->on( finish => sub {
    my ($delay, @txs) = @_;
    my @pods;
    FILE:
    for my $file ( @files ) {
        # ...
    }
    $self->render_file( $file );
});

my @urls = map{
    $pod . $_ . '?content-type=text/x-pod'
}@files;
$ua->get( $_ => $delay->begin ) for @urls;
```

```
$self->render_later;

my $delay = Mojo::IOLoop->delay;
$delay->on( finish => sub {
    my ($delay, @txs) = @_;
    my @pods;
    FILE:
    for my $file ( @files ) {
        # ...
    }
    $self->render_file( $file );
} );

my @urls = map{
    $pod . $_ . '?content-type=text/x-pod'
}@files;
$ua->get( $_ => $delay->begin ) for @urls;
```

```
$self->render_later;

my $delay = Mojo::IOLoop->delay;
$delay->on( finish => sub {
    my ($delay, @txs) = @_;
    my @pods;
    FILE:
    for my $file ( @files ) {
        # ...
    }
    $self->render_file( $file );
});

my @urls = map{
    $pod . $_ . '?content-type=text/x-pod'
}@files;
$ua->get( $_ => $delay->begin ) for @urls;
```

```
$ ./wrk -t8 -d30s -c 10 http://127.0.0.1:3000/blocking
```

```
Running 30s test @ http://127.0.0.1:3000/blocking
```

```
8 threads and 10 connections
```

Thread Stats	Avg	Stdev	Max	+/- Stdev
--------------	-----	-------	-----	-----------

Latency	1.44s	1.66s	2.88s	100.00%
---------	-------	-------	-------	---------

Req/Sec	0.00	0.00	0.00	100.00%
---------	------	------	------	---------

```
10 requests in 30.02s, 13.29KB read
```

```
Socket errors: connect 0, read 8, write 0, timeout 101
```

Requests/sec: **0.33**

Transfer/sec: **453.29B**

```
$ ./wrk -t8 -d30s -c 10 http://127.0.0.1:3000/non_blocking
```

```
Running 30s test @ http://127.0.0.1:3000/non_blocking
```

```
8 threads and 10 connections
```

Thread Stats	Avg	Stdev	Max	+/- Stdev
--------------	-----	-------	-----	-----------

Latency	3.47s	1.14s	5.58s	65.38%
---------	-------	-------	-------	--------

Req/Sec	0.00	0.00	0.00	100.00%
---------	------	------	------	---------

```
79 requests in 30.02s, 92.91KB read
```

```
Socket errors: connect 0, read 0, write 0, timeout 42
```

Requests/sec: **2.63**

Transfer/sec: **3.10KB**



Non-Blocking Mojolicious

Mojo::IOLoop

UserAgent

Datenbank

ElasticSearch

render_later

Achtung!

Alles muss
non-blocking
sein

<http://github.com/renneeb/codetalks2014>

**Vielen Dank
für die Aufmerksamkeit**