

Welcome!

# PHP Inside

Tek X - Chicago, US

Derick Rethans - [derick@php.net](mailto:derick@php.net) - twitter: @derickr

<http://derickrethans.nl/talks.php>

<http://joind.in/1574>

## Derick Rethans



- Dutchman living in London
- PHP development
- Author of the `mcrypt`, `input_filter`, `dbus`, `translit` and `date/time` extensions
- Author of `Xdebug`
- Freelancer doing PHP (internals) development



# Am I mad?



# My First Application

- is the new hello world
- Simple web API to get data
- Authentication with HTTP basic auth
- Simple interface: some buttons, an edit box and a list of tweets



- is a PHP binding for the GTK+ widget set
- It comes as an extension for PHP
- It wraps around many GTK+ widgets
- GUI can be created from PHP:

```
<?php
$tb = new GtkToolbar();
$tb->set_show_arrow( false );
$tb->set_property( 'toolbar-style', Gtk::TOOLBAR_BOTH );
$tb->set_property( 'icon-size', 6 );
$this->updatebutton = GtkToolButton::new_from_stock( Gtk::STOCK_REFRESH );
$lbl = new GtkLabel();
$lbl->set_markup( '<span underline="single">U</span>pdate' );
$this->updatebutton->set_label_widget( $lbl );
$this->updatebutton->connect_simple( 'clicked', array( $this, 'update' ) );
$this->updatebutton->add_accelerator( 'clicked', $accels, Gdk::KEY_U, Gdk::MOD1_MASK, 0 );
$this->updatebutton->set_use_underline( true );
$tb->insert( $this->updatebutton, -1 );
```

# Getting PHP on the device

- It's an ARM processor
- Cross compiling vs compiling on the device

## Cross-compiling:

- Requires cross-compilation set-up
- Tricky, and caused libtool problems for me

## Compiling on the device

- All packages need to be found and installed
- Tricky, and caused libtool problems for me
- Compiling PHP is really slow, you really don't want to use `make clean`

- You can't really run a database server on the phone
- SQLite is part of PHP, and self-contained
- Indexes are vital on this platform
- Clever use of SQL/Data manipulation is required for performance

# Twitter App Experiences

- Starting the app is not fast
- Styling with GTK is tricky, especially if you want to run the app on the desktop too
- Twitter API is very unstable, and inaccessible at times; it's also badly designed
- You pull in way too much data than you need
- My fingers are really fat
- Can't really have the app running for a long time

# Twitter Client: The Result

demo

## Tube service update at 10:22pm

<b>Bakerloo</b>	Good service
<b>Central</b>	Good service
<b>Circle</b>	<b>Severe delays</b>
<b>District</b>	<b>Severe delays</b>
<b>H'smith &amp; City</b>	Good service
<b>Jubilee</b>	Good service
<b>Metropolitan</b>	Good service
<b>Northern</b>	Good service
<b>Piccadilly</b>	Good service
<b>Victoria</b>	Good service
<b>Waterloo &amp; City</b>	<b>Planned closure</b>

- Provides status updates through
- If you're traveling, it's really handy to have this information with you
- There is no API, so you need to scrape
- Widget provided for embedding:
- Strict rules about colours etc.:

▼ **Can I modify the colour schemes in any of the feeds to suit my own site design?**

No. All TfL syndicated feeds must use the official colour schemes as supplied within the feed.

Testing has shown that any deviation from the normal TfL colour scheme makes the information feel less trustworthy to customers. Keeping to these rules means people who use your feeds can be confident that the information is from an official source

- It's probably not totally legal

## Scraping the Tube status

- It's a ~25kb download; with GPRS the download time is about 5 seconds
- It has lots of mark-up and JavaScript that we're not interested in
- Using PHP to get the data we want from it is relatively slow
- If the format changes, all the installed apps need to be updated

- The download with status is only 0.5kb
- The API caches the download
- It only provides the data in JSON that we require:

```
{
  "date":1273835118,
  "status":[
    ["Bakerloo",["Good service",""]],["Central",["Good service",""]],
    ["Circle",["Good service",""]],["District",["Good service",""]],
    ["H'smith & City",["Good service",""]],["Jubilee",["Good service",""]],
    ["Metropolitan",["Good service",""]],["Northern",["Good service",""]],
    ["Piccadilly",["Good service",""]],["Victoria",["Good service",""]],
    ["Waterloo & City",["Good service",""]]
  ]
}
```

- Parsing the JSON is fast, very little processing needed in the app, just display
- If the HTML changes, only the scraping algorithm needs to be changed

# Tube Status: The Result

## Talking to the hardware

- Many applications (phonebook, sending SMS) that came with the phone were suboptimal
- Writing a new GUI is possible, but talking to the hardware requires D-BUS
- D-BUS APIs are exposed by the freesmartphone project that the Linux distribution uses
- PHP didn't have a D-BUS extension

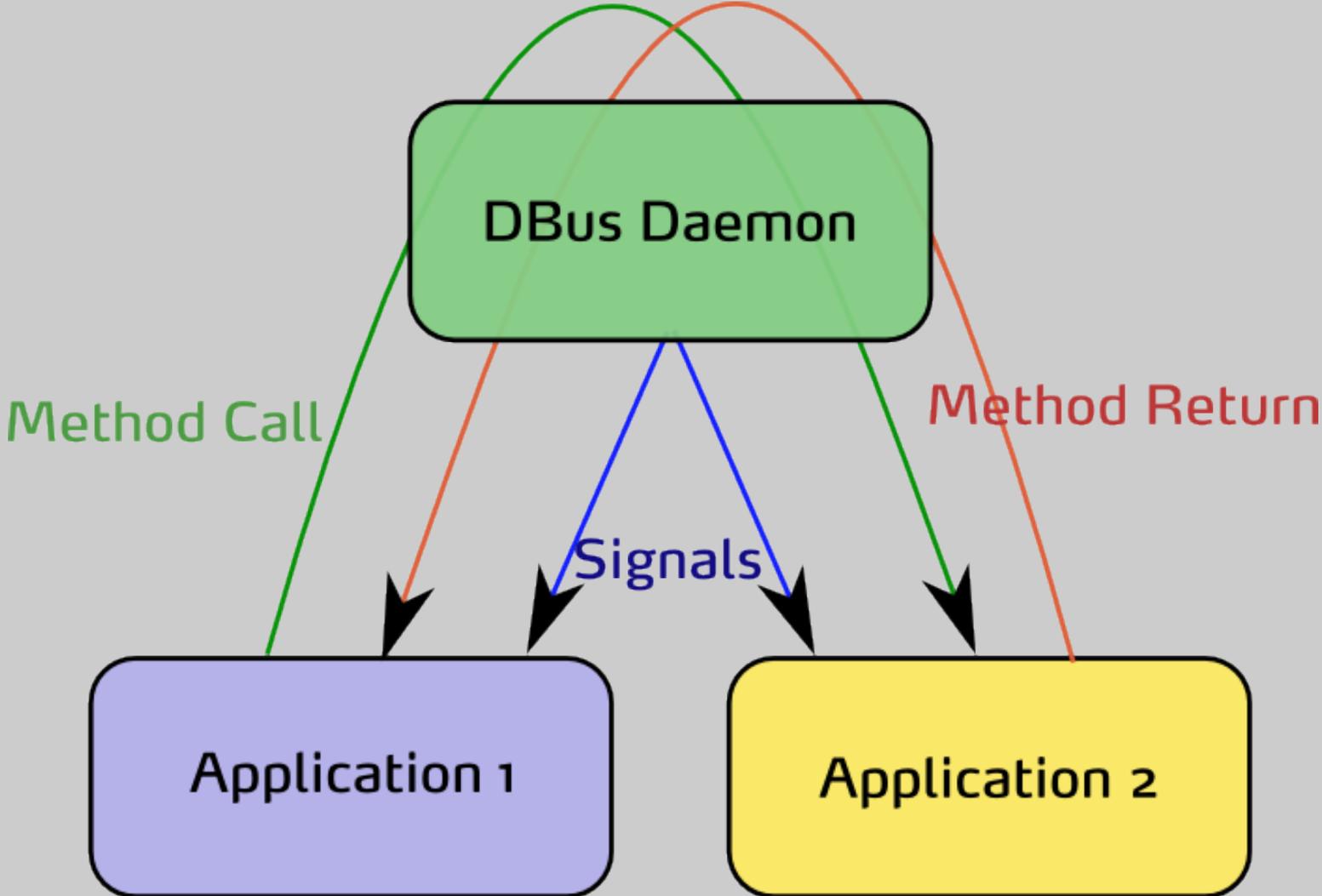
# What is DBUS?

- Part of the freedesktop.org project
- It is meant to replace Corba (GNOME) and DCOP (KDE)
- A message bus system, a simple way for applications to talk to one another
- It consists of a daemon, a library and a protocol
- It is both meant for system level and session level communication



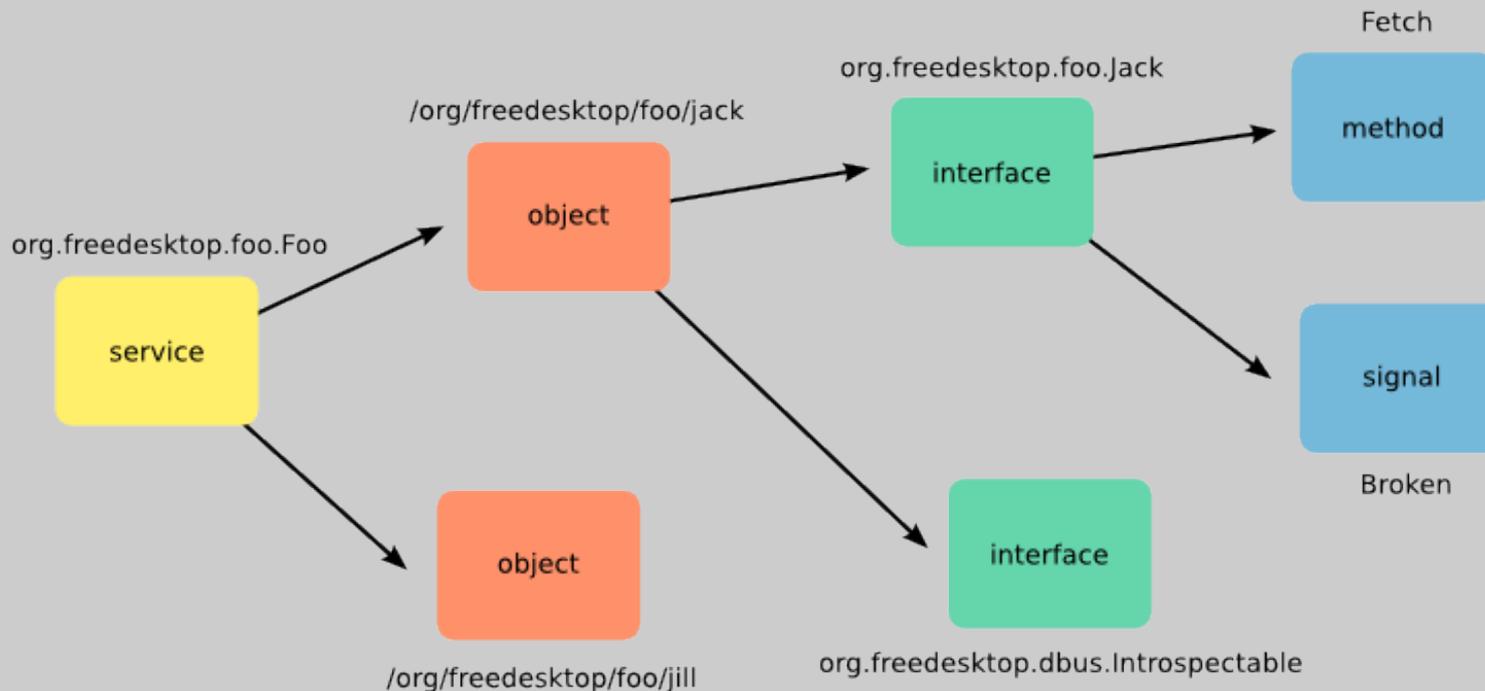
**freedesktop.org**

# Architecture



# DBus Parts

- Bus: Communication pathways that clients can transport messages over
- Proxy: A client-level representation of an object on the bus.



- DBUS bindings using the low-level C library (libdbus)
- Provides proxy objects
- Implements automatic-marshalling of data
- Provides specific classes for full control
- Support for method invocation, method calls, sending and receiving signals
- (Basic) support for introspection

# Talking to a DBUS object from PHP

## Activating the screensaver

### screensaver.php:

```
<?php
$d = new Dbus;
$n = $d->createProxy(
    "org.gnome.ScreenSaver",
    "/org/gnome/ScreenSaver",
    "org.gnome.ScreenSaver"
);

var_dump($n->GetActive());
$n->SetActive( true );
var_dump($n->GetActive());
sleep(5);
$n->SetActive( false );
?>
```

# Phone Services

## Playing Sounds

```
<?php
$d = new Dbus( Dbus::BUS_SYSTEM );
$soundDbus = $d->createProxy(
    'org.freesmartphone.odeviced',
    '/org/freesmartphone/Device/Audio',
    'org.freesmartphone.Device.Audio'
);

$soundDbus->PlaySound(
    '/usr/share/sounds/notify_message.wav',
    0, 0
);
?>
```

# Phone Services

## Activating the shaker

```
<?php
$d = new Dbus( Dbus::BUS_SYSTEM );
$p = $d->createProxy(
    'org.freesmartphone.odeviced',
    '/org/freesmartphone/Device/Vibrator/0',
    'org.freesmartphone.Device.Vibrator'
);

for ( $i =0; $i < 100; $i+=5 )
{
    $p->Stop();
    $p->VibratePattern( 1, 500, 50, $i );
    usleep( 500000 );
}
?>
```

# Phone Services

## Making a call

```
<?php
$d = new Dbus( Dbus::BUS_SYSTEM );
$callInterface = $d->createProxy(
    'org.freesmartphone.ogsmd',
    '/org/freesmartphone/GSM/Device',
    'org.freesmartphone.GSM.Call'
);

$callInterface->Initiate( $nr, 'voice' );

?>
```

# Phone Services

## Sending SMS

```
<?php
$d = new DBus( DBus::BUS_SYSTEM );
$smsInterface = $d->createProxy(
    'org.freesmartphone.ogsmd',
    '/org/freesmartphone/GSM/Device',
    'org.freesmartphone.GSM.SMS'
);

if ( strlen( $text ) <= 160 )
{
    $smsInterface->SendMessage(
        $nr,
        $text,
        new DBusDict( Dbus::VARIANT, array() )
    );
    echo "Sending: $text\n";
}
else
{
    $textParts = chunk_split( $text, 140, "\n" );
    $textParts = explode( "\n", trim( $textParts ) );
    $id = mt_rand( 0, 255 );
    $cnt = count( $textParts );
    $c = 1;
    foreach( $textParts as $textPart )
    {
        $smsInterface->SendMessage(
            $nr,
```

# Phone Services

## Getting GPS coordinates

```
<?php
$d = new Dbus( Dbus::BUS_SYSTEM );
$gpsPos = $d->createProxy(
    'org.freedesktop.Gypsy', '/org/freedesktop/Gypsy',
    'org.freedesktop.Gypsy.Position'
);
$gpsDevice = $d->createProxy(
    'org.freedesktop.Gypsy', '/org/freedesktop/Gypsy',
    'org.freedesktop.Gypsy.Device'
);
$coords = $gpsPos->GetPosition()->getData();
$fix     = $gpsDevice->GetFixStatus();
switch ( $fix )
{
    case 0: $txt = 'Invalid fix'; break;
    case 1: $txt = 'No fix'; break;
    case 2: $txt = '2D fix'; break;
    case 3: $txt = '3D fix'; break;
}
printf(
    "%s | Location: %.2f, %.2f | %s",
    date_create()->format( 'H:i:s' ),
    $coords[2], $coords[3], $txt
);
?>
```

# Application Ideas

- Querying GPS coordinates through SMS
- Activating the shaker through SMS
- ...

# Conclusion

- Mobile devices have little memory and CPU power
- Bandwidth is a real issue
- Lack of APIs
- PHP can run on them, but whether it's smart is to be seen
- Next experiment: PHP on Android/WebOS/Meeego?

Thanks!

Derick Rethans - [derick@php.net](mailto:derick@php.net) - twitter: @derickr

<http://derickrethans.nl/talks.php>

<http://joind.in/1574>