



## OWS Fassadentechnik

---

### Grundgedanken

Die Geodienste einer GDI sind Web Services nach den Spezifikationen der OGC, wie der WMS, WFS, CSW oder WCS. Sie arbeiten mit bestimmten Kommunikationsprotokollen wie HTTP/GET, HTTP/POST oder HTTP/SOAP. Der Kunde erwartet für einen Geodienst eine URL, um den Dienst zu nutzen. An diese URL werden verschiedene standardisierte Anfragen als Operationen (GetCapabilities, GetMap, GetFeature etc.) formuliert. Ergebnis ist eine standardkonforme Antwort oder eine standardkonforme Fehlermeldung.

Soweit zur Theorie der Geodienste. In der Praxis gibt es Abweichungen zwischen der reinen Lehre und der eingekauften Software. Diese Abweichungen können dazu führen, dass die Dienste als nicht interoperabel von außen gesehen werden. Man kann nun lange, intensive und teure Gespräche mit der liefernden Firma führen und irgendwann einen exakten Dienst haben, wenn inzwischen nicht irgendjemand die Profile geändert hat. Oder man stellt zwischen Internet und Geodienst eine Fassade. Sie verhält sich nach außen hin konform, leitet die Anfragen an den internen Dienst weiter und reicht das Ergebnis konform weiter.

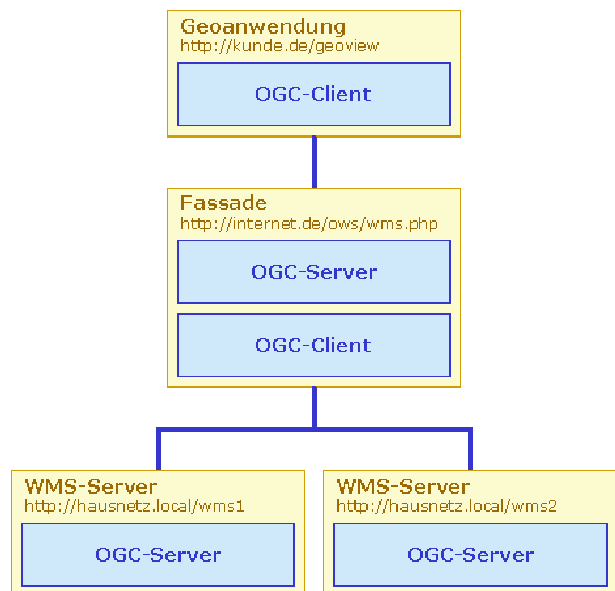
Diese Technik lässt sich sehr einfach mit der Programmiersprache PHP realisieren.

### Architektur einer Fassade

Die Fassade ist ein Softwareprogramm welches sich exakt an die Vorgaben der OGC-Spezifikation und der jeweiligen Anwendungsprofile halten muss. Es muss diese lesen können, um als Geodienst zu agieren und es muss diese schreiben können, um den Geodienst im Hintergrund ansprechen zu können. Eine Fassade ist also OGC-Server und OGC-Client in einem.

Der OGC-Client in der Fassade stellt seine Anfragen an einen oder mehrere OGC-Server. Diese können verschiedene Unzulänglichkeiten haben, welche von der Fassade kaschiert werden.

Die Fassade selbst kann ein Portfolio von Diensten nach Außen hin vollkommen anders darstellen. Es können Layernamen ausgeblendet oder anders benannt werden oder über den Parameter Style bzw. Maßstab können verschiedene Arten von WMS angesteuert werden. Mit der Fassaden-Technik lassen sich alle Geodienste variabler gestalten.



## **Basisimplementierung einer Fassade**

Im Anhang 1 wird ein mögliches Template einer Fassade vorgestellt. Dieses dient als Grundlage für nachfolgende Betrachtungen. Das Template gliedert sich in folgende Bereiche:

- **Konstanten**  
Je nach zu unterstützenden Dienst lassen sich verschiedene Konstanten vordefinieren, wie das aktuelle Datum, Standard-Exceptions oder HTTP-Header-Parameter
- **Funktionen**  
Jedes Template verfügt über verschiedene Arten von Funktionen:
  - Konstante Funktionen (z.B. für die Ausgabe von Fehlermeldungen oder Einlesen des Inputs für HTTP/GET, HTTP/POST, HTTP/SOAP, etc.)
  - Operative Funktionen (für jede unterstützte Version eines Dienstes, je Operation)
  - Implementierungsspezifische Funktionen
- **Input holen**  
Je nach Art der Anfrage (HTTP/GET, HTTP/POST als KVP, HTTP/POST als XML oder HTTP/POST als SOAP) ist eine eigene Funktion zum Einlesen des Inputs notwendig
- **Input analysieren**  
Die Inputanalyse hat verschiedene Aufgaben zu erfüllen:
  - Überführung des Inputs in Variablen
  - Ermittlung der Art des Dienstes, der Operation und der Version
- **Ermittlung des entsprechenden internen Geodienstes**  
Anhand der Angaben "Art des Dienstes", "Operation" und "Version" wird entschieden welcher interne Geodienst für die anschließende Verarbeitung genutzt wird, oder ob eine Fehlermeldung ausgegeben wird (z.B. weil die Operation "PutStyle" beim WMS 1.3.0 nicht implementiert wurde)
- **Evtl. Zulassungsprüfung des Dienstes für den Anfragenden**  
Ist der Anfragende bekannt, kann eine Zulässigkeitsprüfung der Anfrage durchgeführt werden
- **Filtern des Inputs**  
Es gibt verschiedene Anwendungsmöglichkeiten um die Eingabe des Anfragenden zu filtern oder zu verändern. Anbei einige Praxisbeispiele der LGB:
  - Umbenennen von Layern, hinzufügen von Layern
  - Veränderung von EPSG-Codes, Abwandlung der BBOXErgebnis dieses Schrittes ist die eigentliche Anfrage an den eigentlichen Dienst
- **Verarbeitung der Operation**  
Die Anfrage wird abgesendet und das Ergebnis wird in einen Puffer übertragen
- **Filtern des erzeugten Outputs**  
Das zuvor erhaltene Ergebnis kann vor der Ausgabe je nach Zweck überarbeitet werden:
  - Eine Grafik wird durch eine Gamma-Korrektur aufgehellt oder verdunkelt
  - Eine Grafik erhält ein Wasserzeichen oder eine Copyrightangabe
  - In einer XML-Datei werden bestimmte Angaben (z.B. URLs) ausgetauscht
- **Ausgabe des Outputs**  
Der überarbeitete Ausgabebuffer wird auf die Standardausgabe ausgegeben
- **Protokollierung des Zugriffs**  
Der Zugriff wird zum Beispiel mit den Angaben *IP-Adresse des Nutzers, Datum, Uhrzeit, Art des Dienstes, Operation, Version* und *Aufbereitungsdauer* protokolliert

Die gesamte Laufzeit der Fassade sollte 0.1 Sekunden nicht übersteigen, um die Anfrage auf den eigentlichen Dienst nicht unnötig zu bremsen.

## **Ausbau einer Fassade**

In der Landesvermessung und Geobasisinformation Brandenburg (LGB) werden Fassaden seit 2005 eingesetzt um den UMN Mapserver konform zum GDI-BE/BB und neuerdings zum GDI-DE WMS Profil zu machen. Eine Feinabstimmung erfolgte jedoch noch nicht abschließend. Weiterhin wurden Fassaden für einen Gazetteer (WFS) und einen Katalogdienst (CSW) erstellt.

Folgende Gründe sprechen für die Nutzung einer Fassade:

- *Kontrolle:* Der eigentliche Geodienst (vom Softwarehaus) wird nur gefiltert angesprochen
- *Schutz:* Der eigentliche Dienst liegt in einer stärker geschützten DMZ
- *Schutz:* Es werden nur bestimmte Operationen frei angeboten
- *Sicherheit:* Es wird eine unsichtbare Benutzerverwaltung zwischengeschaltet
- *Sparsamkeit:* Es wird ein "neuer" Dienst von einer "älteren" Version simuliert
- *Konformität:* Schwächen/Verarbeitungsfehler des eigentlichen Dienstes werden kaschiert
- *Flexibilität:* Es werden die Ergebnisse verschiedener Dienste zusammenkopiert
- *Flexibilität:* Es werden Parameter anders ausgewertet, um flexiblere Ergebnisse anbieten zu können (z.B. *lenkt der Parameter Style die Ansteuerung unterschiedlicher WMS*)
- *Flexibilität:* Es werden bestimmten Nutzern nicht alle Layer in den Capabilities angeboten

Die Verwendung von Fassaden erspart Ausschreibungen zur speziellen Anpassung von Diensten und gestaltet den Einsatz von Geodiensten allgemein flexibler.

Michael Dreesmann

*Geschäftsstellenleiter der Geodateninfrastruktur Brandenburg*  
[michael.dreesmann@geobasis-bb.de](mailto:michael.dreesmann@geobasis-bb.de)

## Anhang 1 Fassaden - Template

---

```
<?php
/*
 *   Template Fassade                               (C) 2007, LGB
 *
 *   2007-07-05 Michael Dreesmann
 *   Übernahme aus diversen erstellten Fassaden, die Funktionen
 *   fct_removeHTTPHeader() und fct_post_it() von Gordon Stender übernommen
 *   -----
 *
 *   KONSTANTEN
 *   -----
 */
$rm      = $_SERVER[ 'REQUEST_METHOD' ];
$request = "GetCapabilities";
$version = "1.1.1";
$service = "WMS";
$srs      = "EPSG:25833";
$exceptions = "application/vnd.ogc.se_xml";
$year     = date( "Y" );

/*
 *   FUNKTIONEN
 *   -----
 */
function fct_exception_out( $message, $version )
{
    global $head;
    global $exceptions;

    $head = $exceptions;
    $data = sprintf( "<?xml version='1.0' encoding='ISO-8859-1'?>\n
<ExceptionReport version='%s'>\n <Exception code='Unspecified'%s</Exception>\n
</ExceptionReport>\n", $version, $message );

    return $data;
}

function fct_exception( $mime, $message, $version )
{
    $hdr = sprintf( "Content-Type: %s\n\n", $mime );
    header( $hdr );
    printf( "%s", fct_exception_out( $message, $version ) );

    exit;
}

function fct_get_input()
{
    return $_SERVER[ 'QUERY_STRING' ];
}

function fct_post_input()
{
    return file_get_contents("php://input");
}

function fct_WMS111_capa()
{
    global $year;
    global $version;
    global $head;
    global $exceptions;

    $url = sprintf( "./dnm_%d_wms_capa.xml", $year );
}
```

```

if( !($fp=@fopen($url,"r")) )
{
    $head = $exceptions;
    return fct_exception_out( "Capabilities nicht gefunden", $version );
}
else
{
    $head = "application/vnd.ogc.wms_xml";
    fclose( $fp );
    return fct_get_val( $url );
}
}

function fct_WMS111_map()
{
    global $url;
    global $year;
    global $version;
    global $head;
    global $exceptions;
    global $format;
    global $width;
    global $height;
    global $srs;
    global $bbox;
    global $layers;
    global $styles;

    $head = $format;

    $call = sprintf(
"%smap=dnm_%s.map&%s&width=%d&height=%d&%s&srs=%s&bbox=%s&layers=%s&exceptions=%s",
        $url, $year, "service=WMS&version=1.1.1&request=GetMap",
        $width, $height,
        "format=image/png", $srs, $bbox, $layers, $exceptions );
    if( !($data=fct_get_val( $url )) )
    {
        $head = $exceptions;
        return fct_exception_out( "Fehlerhafte GetMap-Anfrage", $version );
    }

    return $data;
}

function fct_WMS111_info()
{
}

function fct_WFS110_capa()
{
}

function fct_WFS110_type()
{
}

function fct_WFS110_feat()
{
}

function fct_get_val( $url )
{
    $data = "";
    if( !($fp=@fopen( $url, "r" )) )
        return 0;
}

```

```

while( $dt = @fread($fp, 512) )
    { $data = $data . $dt; }
@fclose( $fp );

return trim($data);
}

function fct_removeHTTPHeader($httpstream)
{
    $trenner = "\r\n\r\n";

    $pos = strpos ($httpstream, $trenner);
    if ($pos === false) {
        echo "BROKEN HTTP-REQUEST";
    }
    else {
        $httpstream = substr($httpstream,$pos+strlen($trenner));
    }

    return $httpstream;
}

function fct_post_it( $datastream, $url, $port )
{
    $url      = preg_replace("@^http://@i", "", $url);
    $host     = substr($url, 0, strpos($url, "/"));
    $uri      = strstr($url, "/");
    $reqbody = "";

    if(is_array($datastream))
    {
        foreach($datastream as $key=>$val) {
            if (!empty($reqbody)) $reqbody.= "&";
            $reqbody.= $key."=".urlencode($val);
        }
    } else $reqbody.=$datastream;

    $contentlength = strlen($reqbody);
    $reqheader = "POST $uri HTTP/1.0\r\n".
        "Host: $host\r\n". "User-Agent: LGBfassade\r\n".
        "Content-Length: $contentlength\r\n\r\n".
        "$reqbody\r\n";

    $socket = fsockopen($host, $port, $errno, $errstr);

    if (!$socket) {
        $result["errno"] = $errno;
        $result["errstr"] = $errstr;
        return $result;
    }

    fputs($socket, $reqheader);

    $buf = "";
    while( $data = fread($socket, 512) )
        $buf = $buf . $data;
    $buf = fct_removeHTTPHeader($buf);
    fclose($socket);

    return $buf;
}

```

```

/*
 * HOLE INPUT
 ==> $data
 * -----
 */
if( $rm == "GET" )
    $data = fct_get_input();
else
    $data = fct_post_input();

/*
 * ANALYSIERE INPUT
 ==> $oper
 * -----
 */
if( $rm == "GET" )
{
    $q1 = explode( "&", $data );
    $qn = count($q1);
    for( $i=0; $i<$qn; $i++ )
    {
        $q2 = explode( "=", $q1[$i] );
        $vn = strtolower( $q2[0] );
        $trans = array( "%2C" => ",",
                       "%2E" => ".",
                       "%2F" => ":" );
        $$vn = strstr( $q2[1], $trans );
    }

    $oper = sprintf( "%s_%s_%s", $service, $version, $request );
}
else
{
    if( strncmp( $data, "<wfs:GetCapabilities", 20 ) == 0 )
    {
        $service = "WFS";
        $version = "1.1.0";           // GERATEN == INDIVIDUELL
        $request = "GetCapabilities";
    }
    else if( strncmp( $data, "<wfs:DescribeFeatureType", 24 ) == 0 )
    {
        $service = "WFS";
        $version = "1.1.0";           // GERATEN == INDIVIDUELL
        $request = "DescribeFeatureType";
    }
    else if( strncmp( $data, "<wfs:GetFeature", 15 ) == 0 )
    {
        $service = "WFS";
        $version = "1.1.0";           // GERATEN == INDIVIDUELL
        $request = "GetFeature";
    }
    else
    {
        $service = "WFS";
        $version = "1.1.0";           // GERATEN == INDIVIDUELL
        $request = "Unbekannt";
    }
    $oper = sprintf( "%s_%s_%s", $service, $version, $request );
}
}

```

```

/*
 * BESTIMME OWS
 ==> $url, $prt
 * -----
 */
if( $rm == "GET" )
    $url = "http://10.134.11.123:85/cgi-bin/umn4.11.2?";
else
{
    $url = "http://10.134.11.123/cgi-bin/umn4.11.2?";
    $prt = 85;
}

/*
 * VERARBEITUNG
 ==> $resp, $head
 * -----
 */
switch( $oper )
{
    case "WMS_1.1.1_GetCapabilities":
        $resp = fct_WMS111_capa();
        break;
    case "WMS_1.1.1_GetMap":
        $resp = fct_WMS111_map();
        break;
    case "WMS_1.1.1_GetFeatureInfo":
        $resp = fct_WMS111_info();
        break;
    case "WFS_1.1.0_GetCapabilities":
        $resp = fct_WFS110_capa();
        break;
    case "WFS_1.1.0_DescribeFeatureType":
        $resp = fct_WFS110_type();
        break;
    case "WFS_1.1.0_GetFeature":
        $resp = fct_WFS110_feat();
        break;
    default:
        $resp = fct_exception_out( "Unbekannte Operation", $version );
}

/*
 * AUSGABE DES ERGEBNISSES
 ==> STDOUT (UNGEFILTERT)
 * -----
 */
$hdr = sprintf( "Content-Type: %s\n\n", $head );
header( $hdr );
echo $resp;

/*
 * PROTOKOLLIERUNG DER ANFRAGE
 ==> NIL
 * -----
 */

$prtf = "./protokoll.txt";
$timf = time();
$adrfr = $_SERVER[ "REMOTE_ADDRESS" ];

$fp = @fopen( $prtf, "a" );
$linf = sprintf( "%s %10d %20s %s\n", $service, $timf, $adrfr, $request );
fwrite( $fp, $linf );
fclose( $fp );

?>

```