SOAP Bubbles

Radovan Semančík

Čo je SOAP?

- Simple Object Access Protocol
  - Funkcia: Posielanie správ, RPC, ...
  - Cieľ: Komunikácia “systém ↔ systém”
  - Formát: XML
  - Transport: HTTP, HTTPS, SMTP, POP3, TCP, ...

Načo je SOAP?

- Komunikácia “systém ↔ systém”

SOAP

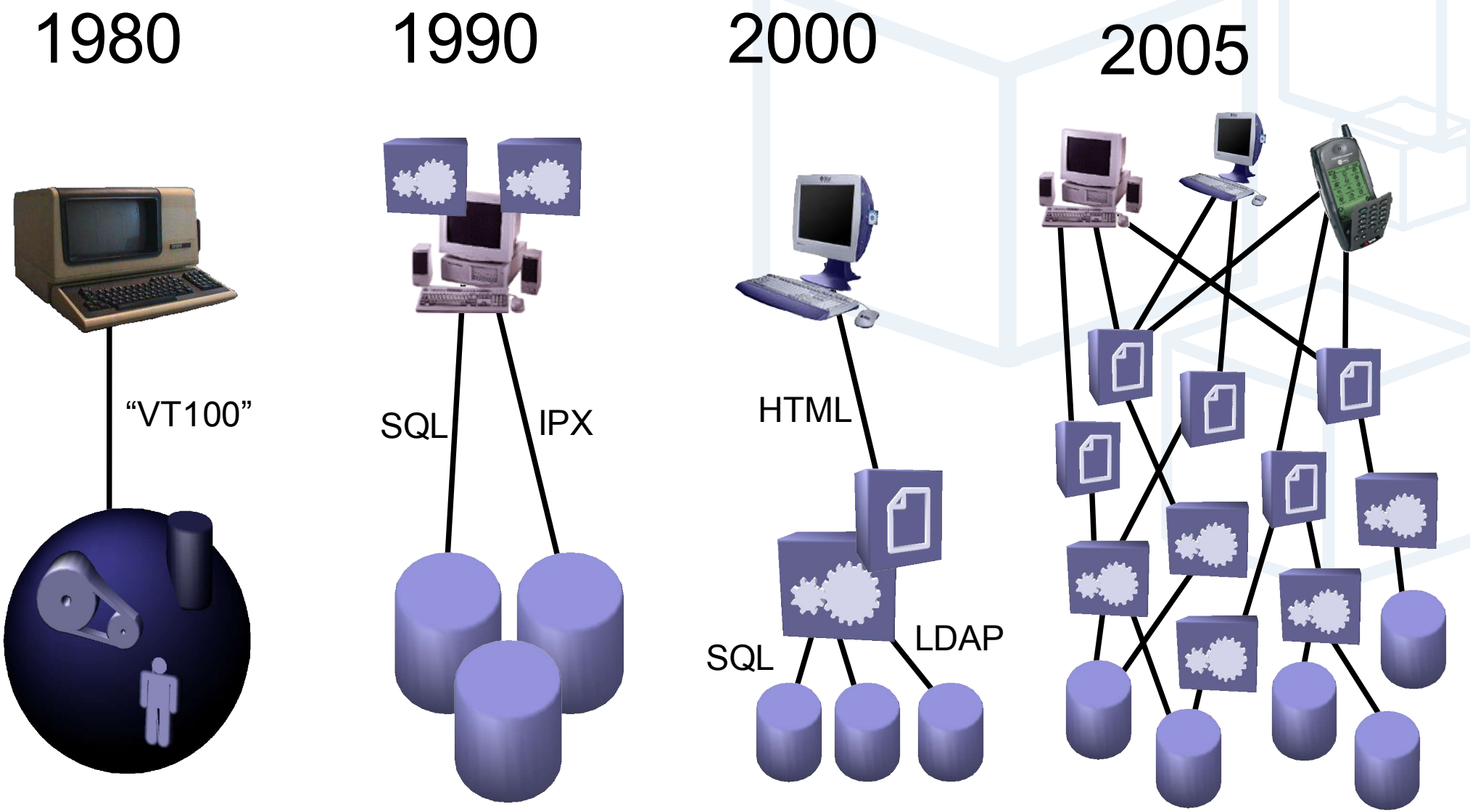
Posielanie správ, RCP, integrácia,
synchronizácia, replikácia, ...

- Porovnanie: HTTP/HTML: Komunikácia “systém ↔ používatel”

HTML

BGS
Business Global Systems

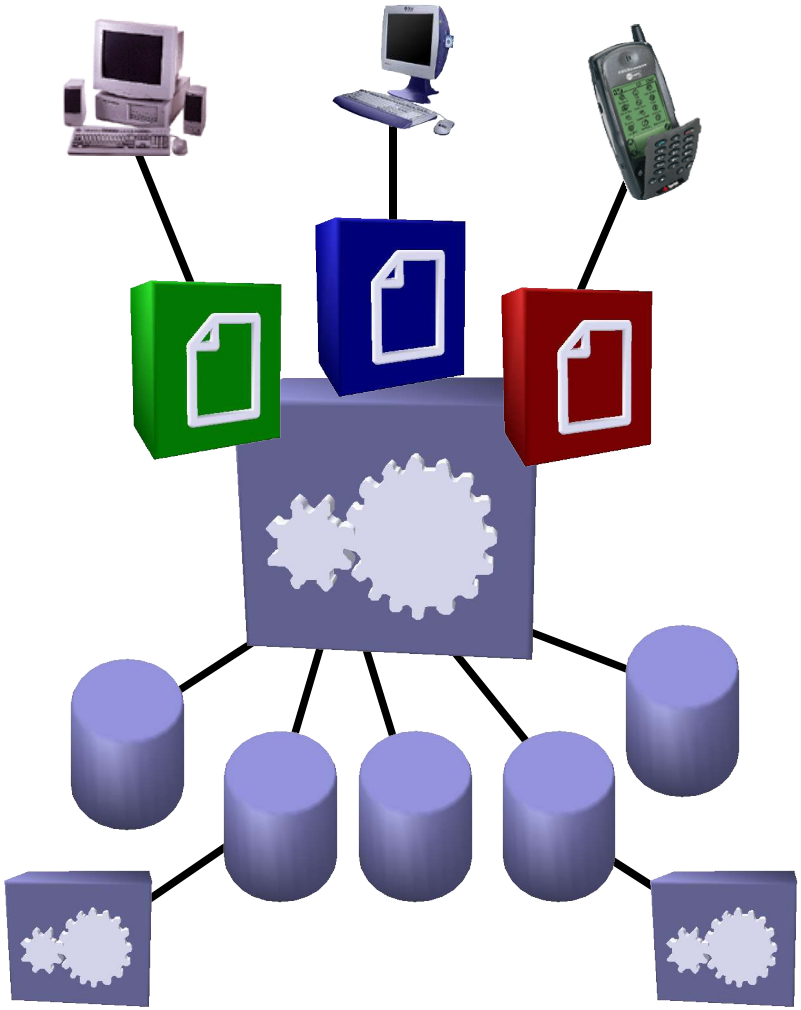
Obdočka: Načo je nám RPC a spol?



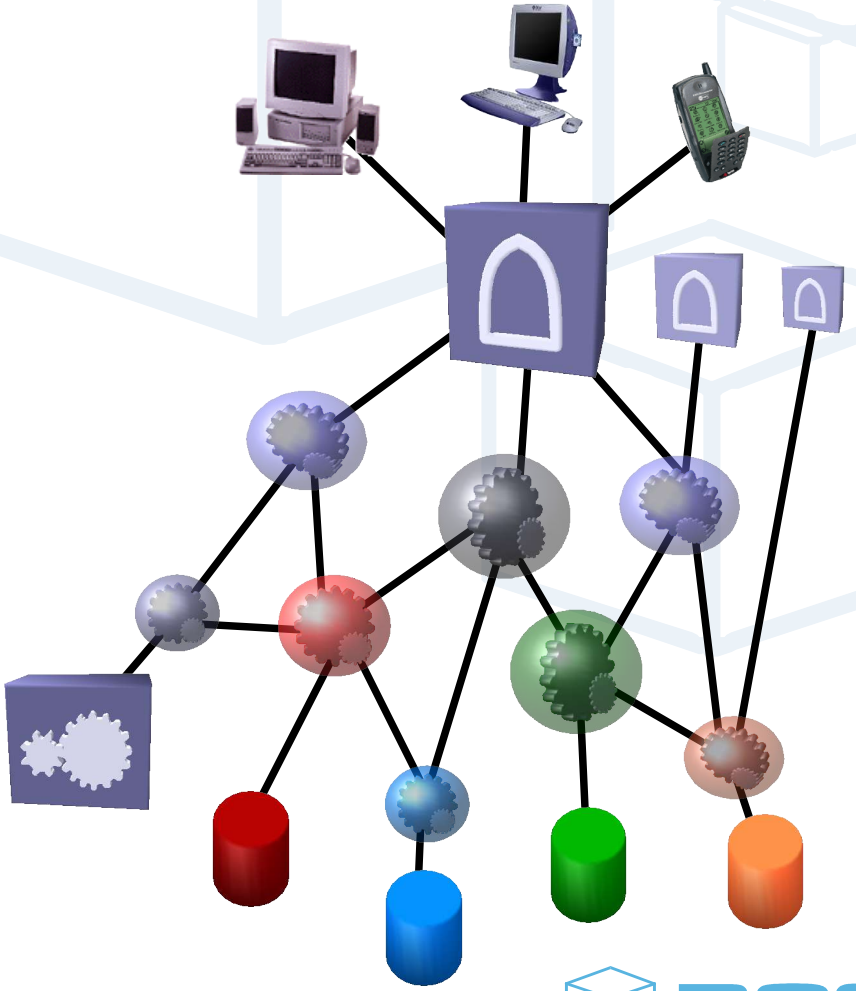
BGS
Business Global Systems

Možnosti riešenia

Pevne viazané
(Tight Coupled)



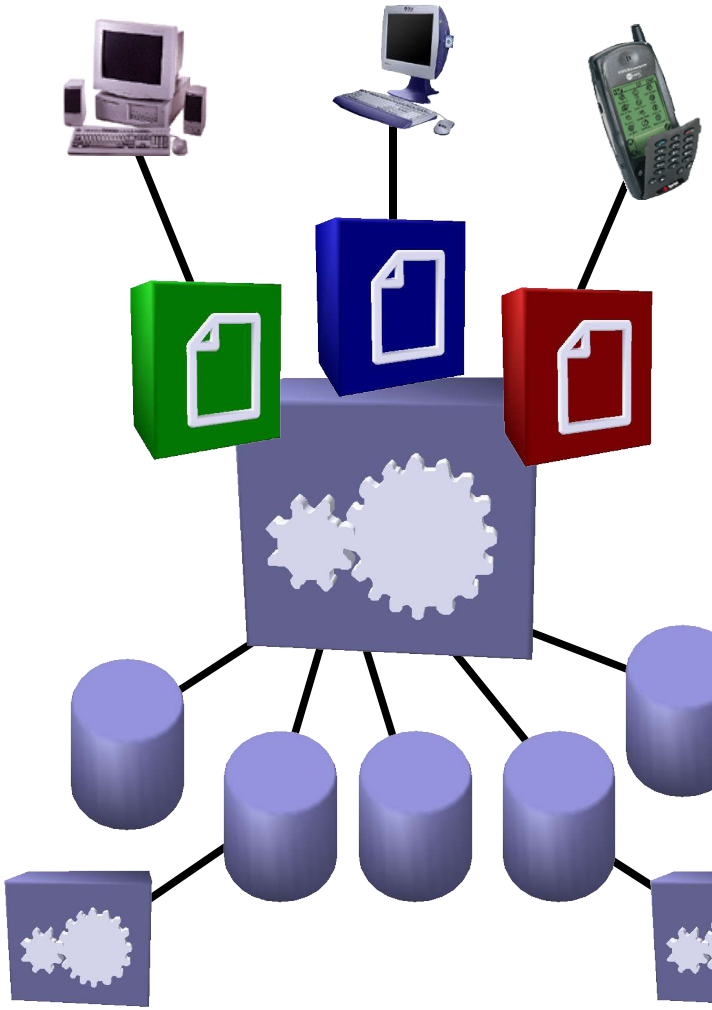
Voľne viazané
(Loose Coupled)



BGS
Business Global Systems

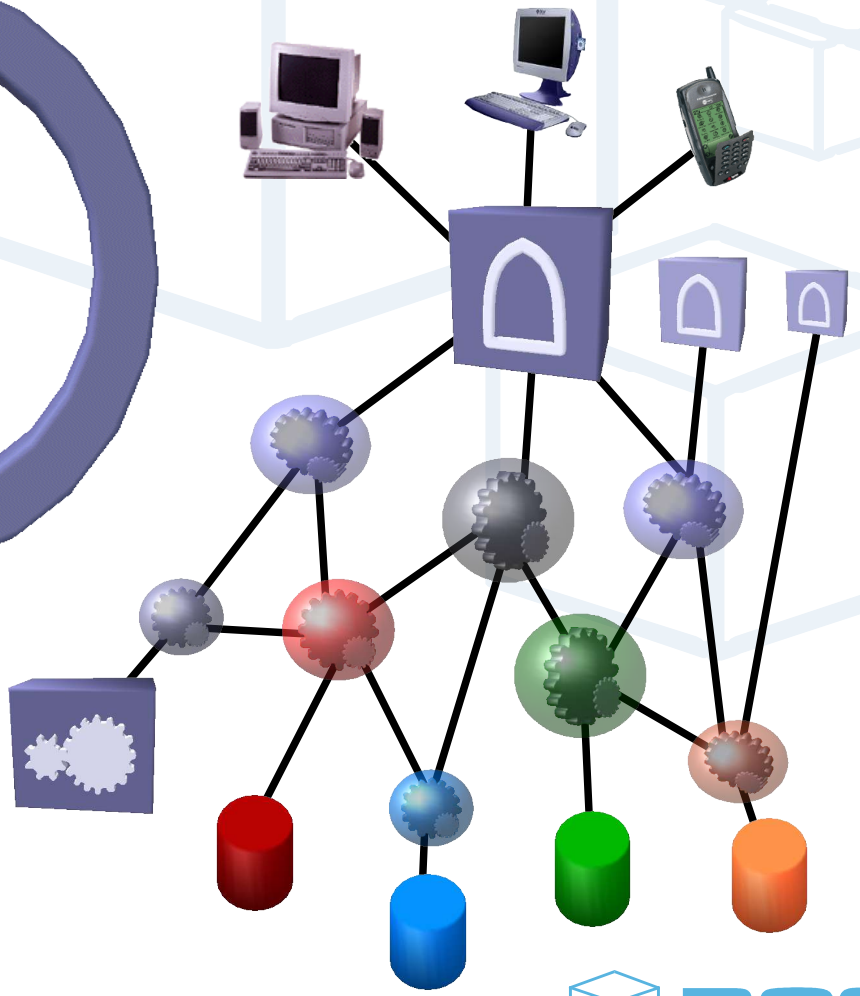
Fight!

Pevne viazané
(Tight Coupled)



vs

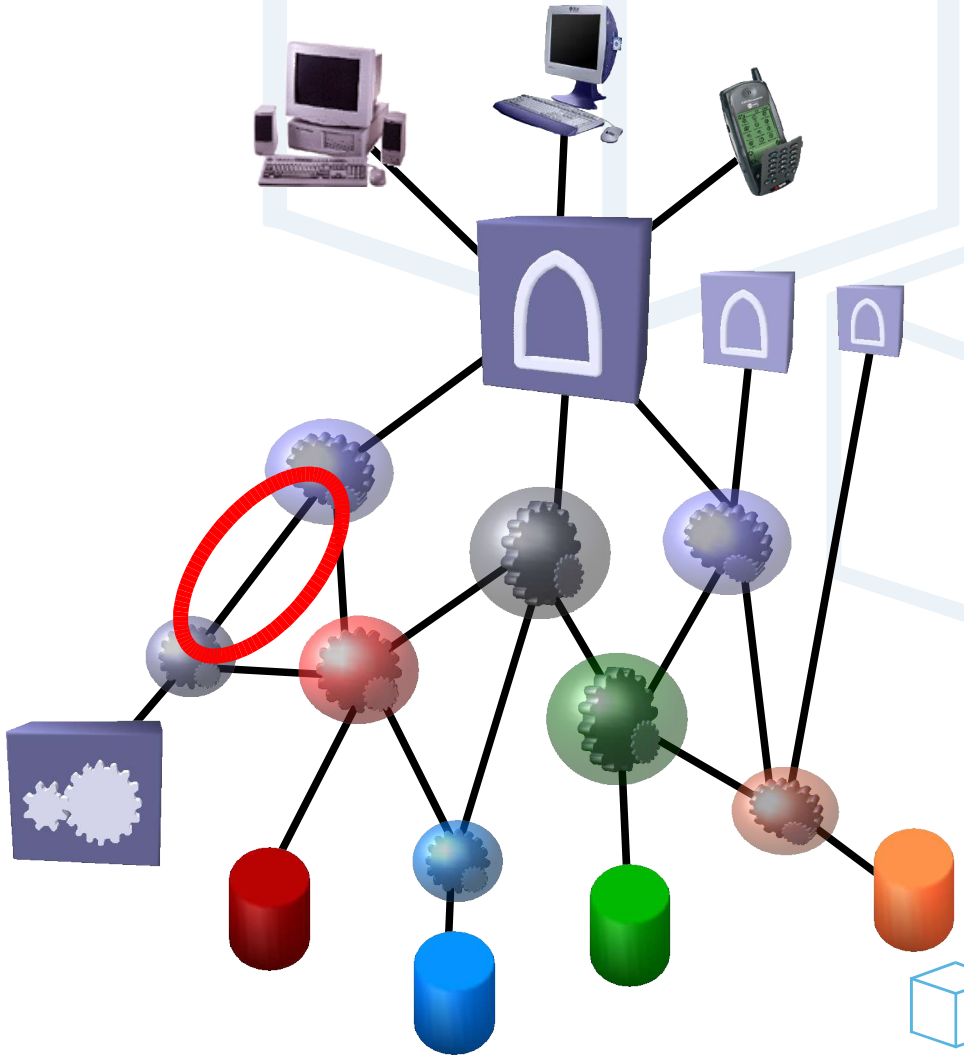
Voľne viazané
(Loose Coupled)



BGS
Business Global Systems

And the winner is:

Voľne viazané
(Loose Coupled)



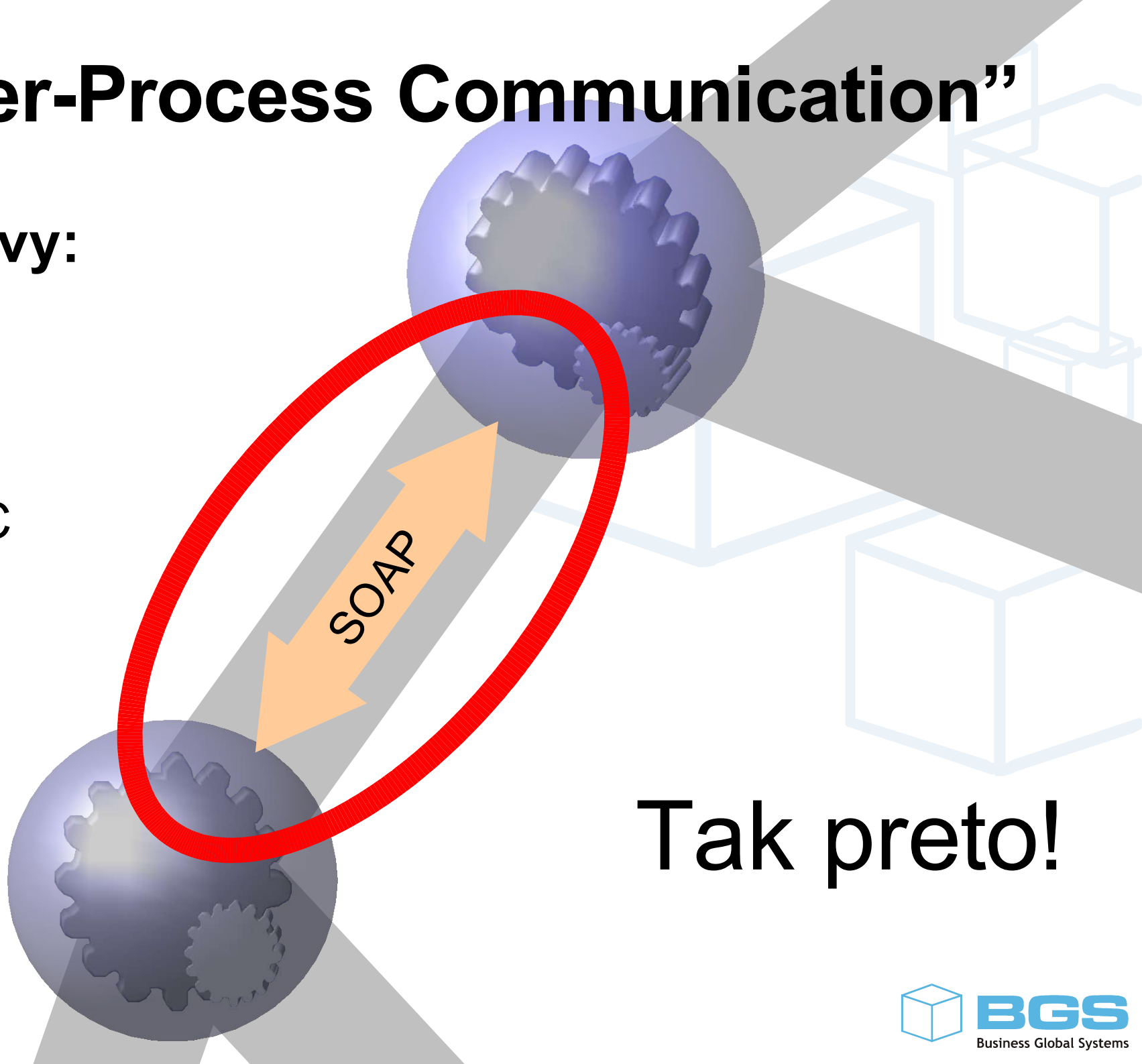
Pevne viazané
(Tight Coupled)

BGS
Business Global Systems

“Inter-Process Communication”

Alternatívy:

- CORBA
- DCOM
- Java RMI
- Sun RPC
- XML-RPC



Tak preto!

BGS
Business Global Systems

Ako pracuje SOAP?

Any technology distinguishable from magic is insufficiently advanced.

-- Anonymous

Ako pracuje SOAP?

SOAP Message (Request)

```
<sayHelloTo>
<name>Slartibartfast</name>
</sayHelloTo>
```

TCP
HTTP
SMTP/POP3
MQ Series ...

SOAP

SOAP Message (Response)

```
<sayHelloToResponse>
<message>Hello Slartibartfast, How are you?</message>
</sayHelloToResponse>
```

BGS
Business Global Systems

Ako pracuje SOAP/HTTP?

SOAP Message (Request)

```
<sayHelloTo>
<name>Slartibartfast</name>
</sayHelloTo>
```

HTTP Request
Content-type: text/xml

HTTP Response
Content-type: text/xml

SOAP Message (Response)

```
<sayHelloToResponse>
<message>Hello Slartibartfast, How are you?</message>
</sayHelloToResponse>
```

BGS
Business Global Systems

SOAP Message

```
<soap:Envelope
   xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
  soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">
   <soap:Header>
     <t:Transaction xmlns:t="http://www.foo.com/transcations/">
       345627483451
     </t:Transaction>
   </soap:Header>
   <soap:Body xmlns:m="http://www.stock.org/stock">
     <m:GetStockPriceRequest>
       <m:StockName>SUNW</m:StockName>
     </m:GetStockPriceRequest>
   </soap:Body>
</soap:Envelope>
```

Obálka

Metadata
(Transakcie, Bezpečnosť, ...)

Data
(Parametre, návratové hodnoty)

BGS
Business Global Systems

<soap:Body>

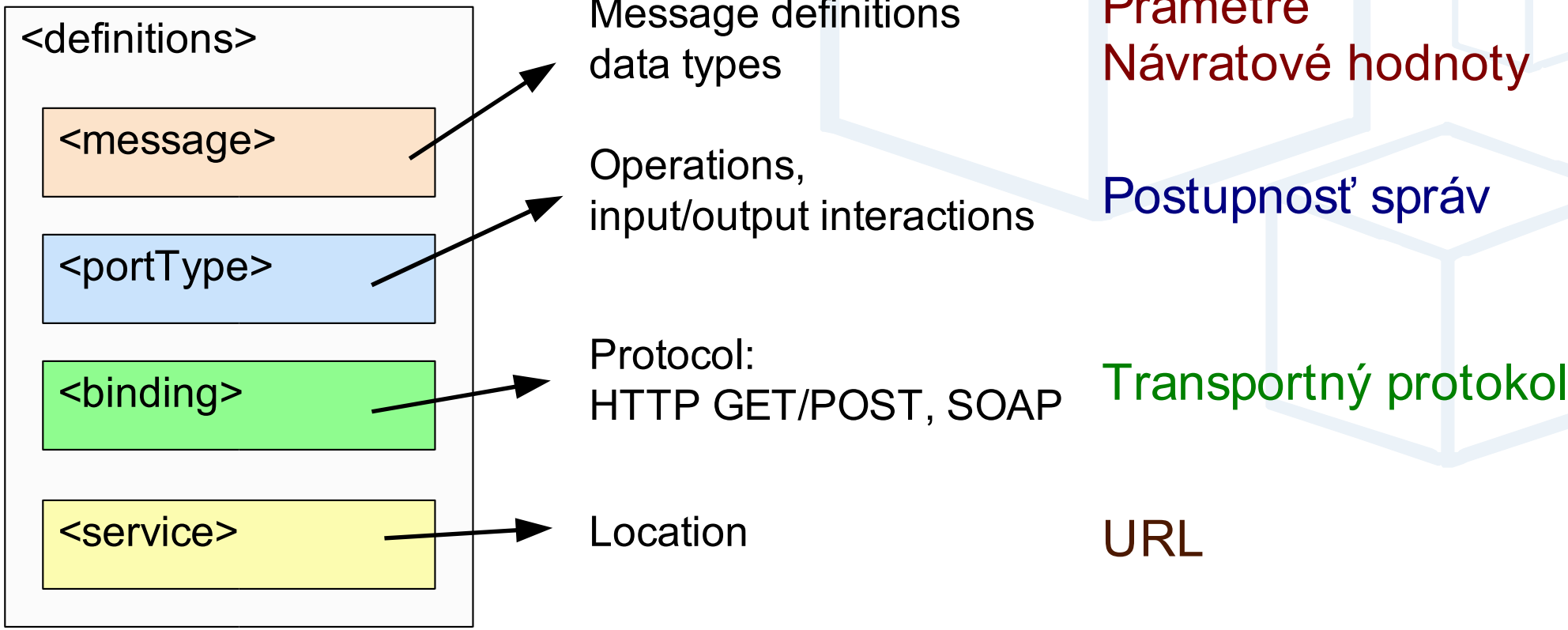
```
<soap:Body xmlns:m="http://www.stock.org/stock">
   <m:GetStockPriceRequest>
    <m:StockSymbol>SUNW</m:StockSymbol>
   </m:GetStockPriceRequest>
</soap:Body>
```

```
<soap:Body xmlns:m="http://www.stock.org/stock">
   <m:GetStockPriceResponse>
    <m:StockPrice>4.35</m:StockPrice>
   </m:GetStockPriceResponse>
</soap:Body>
```

WSDL

Web Services Description Language

- Popis rozhrania



BGS
Business Global Systems

WSDL príklad: <message>

```
<definitions name="StockPriceService"
   targetNamespace="http://storm.alert.sk/StockPriceService"
   xmlns="http://schemas.xmlsoap.org/wsdl/"
   xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
   xmlns:tns="http://storm.alert.sk/StockPriceService"
   xmlns:xsd="http://www.w3.org/2001/XMLSchema">

   <message name="GetStockPriceRequest">
      <part name="stockSymbol" type="xsd:string"/>
   </message>

   <message name="GetStockPriceResponse">
      <part name="stockPrice" type="xsd:decimal"/>
   </message>

...
```

Parameters

Return Values

GetStockPriceRequest
stockSymbol: string

GetStockPriceResponse
stockPrice: decimal

WSDL príklad: <portType>

...

```
<portType name="StockPricePortType">

   <operation name="getStockPrice">
      <input message="tns:GetStockPriceRequest"/>
      <output message="tns:GetStockPriceResponse"/>
   </operation>

</portType>
```

...

WSDL príklad: <binding>

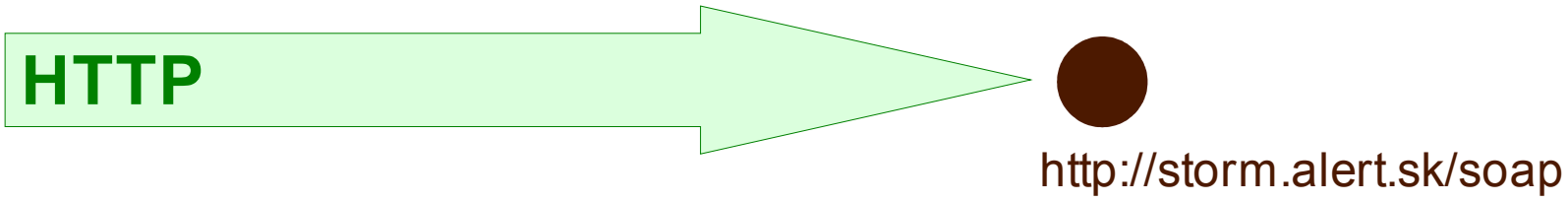
```
…
  <binding name="StockPriceSoapBinding" type="tns:StockPricePortType">
     <soap:binding
           style="rpc"
           transport="http://schemas.xmlsoap.org/soap/http"/>
        <operation name="getStockPrice">
           <soap:operation
        soapAction="http://storm.alert.sk/StockService#getStockPrice"/>
           <input>
              <soap:body
                 encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
                 namespace="http://storm.alert.sk/StockService" use="encoded"/>
           </input>
           <output>
              <soap:body
                 encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
                 namespace="http://storm.alert.sk/StockService" use="encoded"/>
           </output>
        </operation>
  </binding>
…
```

Action URI

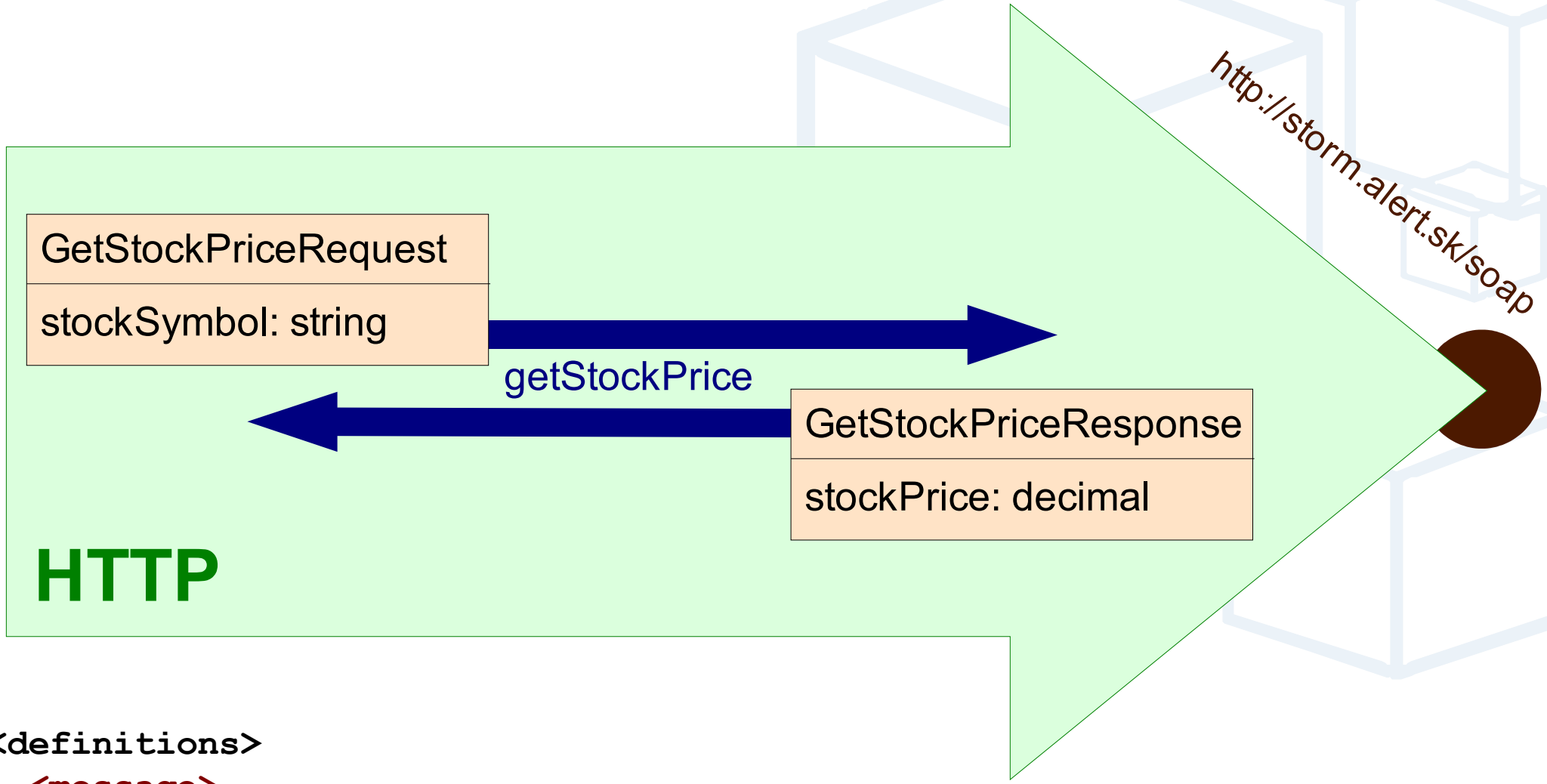
WSDL príklad: <service>

```
...
    <service name="StockService">

        <documentation>Stock Price Service</documentation>

        <port
              binding="tns:Hello_Binding"
              name="Hello_Port">
           <soap:address
              location="http://storm.alert.sk/soap"/>
        </port>

    </service>

</definitions>
```

Service URL



WSDL: Big Picture




```
<definitions>
   <message>
   <portType>
   <binding>
   <service>
</definitions>
```

Ako na to?

I hear and I forget. I see and I remember. I do and I understand.

-- Confucius

Think, think, think, ...

- Čo idem robiť?
  - Ako sa to má správať?
  - Čo je môj cieľ?
- Navrhnúť rozhranie
  - WSDL ... alebo “lokálne” (Java) a WSDL vygenerovať

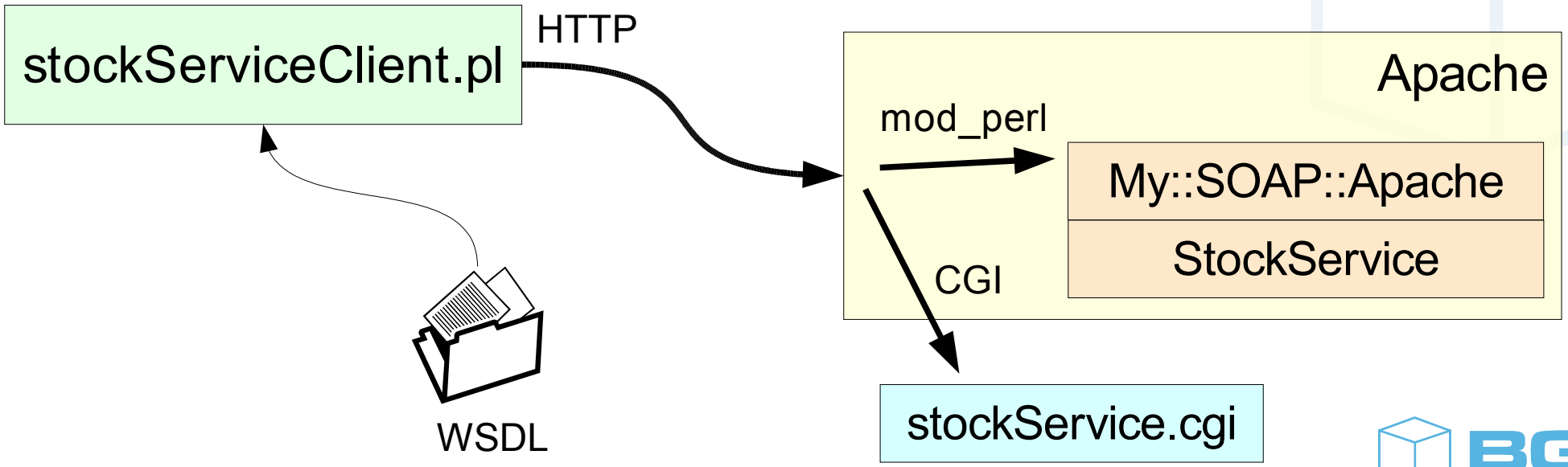
> A couple of months in the laboratory can frequently save a couple of hours in the library.
>
> -- Rule of Discovery

Predpoklady

- Mám WSDL definíciu rozhrania
  - V lokálnom súbore na serveroch aj klientoch
  - Publikovanú cez web (lepšie)
- SOAP HTTP Binding
- RPC mechanizmus
- Zjednodušenie: bez kontroly chýb

Perl SOAP: Základy

- SOAP::Lite (CPAN, soaplite.com)
  - Update: Február 2004
- SOAP (CPAN)
  - Update: September 2000 – zrejme opustený



SOAP in Perl: Client (SOAP::Lite)

```
#!/usr/bin/perl
use strict;

use SOAP::Lite;

my $soap = SOAP::Lite->service("stockservice.wsdl");

my $result = $soap->getStockPrice('SUNW');

print $result->result;
```

Perl SOAP: CGI Server

(SOAP::Lite)

```
#!/usr/bin/perl
use strict;

use SOAP::Transport::HTTP;
SOAP::Transport::HTTP::CGI
    -> dispatch_to('StockService') -> handle;

package StockService;

sub getStockPrice {
    my ($class,$stockSymbol) = @_;

    if ($stockSymbol eq "IBM")
        { return 95 + rand(30) - 15; }
    return rand(80);
}
```

```
<soap:operation soapAction="http://storm.alert.sk/StockService#getStockPrice"/>

<soap:body ... namespace="http://storm.alert.sk/StockService" ... />
```

BGS
Business Global Systems

Perl SOAP: mod_perl Server
(SOAP::Lite)

```
#!/usr/bin/perl
use strict;

package My::SOAP::Apache;
use SOAP::Transport::HTTP;

my $server = SOAP::Transport::HTTP::Apache
    -> dispatch_to('StockService');

sub handler { $server->handler(@_) }

1;
```

Implementácia v `StockService.pm`

```
<Location /soap>
    SetHandler perl-script
    PerlHandler My::SOAP::Apache;
</Location>
```

```
<soap:address location="http://storm.alert.sk/soap"/>
```

BGS
Business Global Systems

Perl SOAP: StockService.pm

(SOAP::Lite)

```
#!/usr/bin/perl
use strict;

package StockService;

use base qw(SOAP::Server::Parameters);

sub getStockPrice {
    my $self = shift;
    my $env = pop;
    my $params = $env->method;

    my $stockSymbol = $params->{stockSymbol};

    ...
}

1;
```

Rozšírené API

```
<Body>
  <getStockPrice>
   <StockSymbol>SUNW<StockSymbol>
  </getStockPrice>
</Body>
```

Parameter podľa mena

BGS
Business Global Systems

Zhodnotenie: Perl

- Problémy
  - Neberie do úvahy <message> definície vo WSDL
    - Potencionálne problémy s interoperabilitou
  - Problematicky sa píše WSDL
    - Nervydrásajúce hladanie chýb
  - Perl je beztypový, SOAP typový
    - Problémy napr. “decimal” vs “float”, atď.
    - Ak to chem spraviť správne, veľa sa nakódujem

Realita

```
<getStockPrice>
  <StockSymbol>SUNW<StockSymbol>
</getStockPrice>
```

vs

WSDL

```
<GetStockPriceRequest>
  <StockSymbol>SUNW<StockSymbol>
</GetStockPriceRequest>
```

Zhodnotenie: Perl

- Doporučenia (cesta najmenšieho odporu)
  - Prispôsobiť WSDL tomu čo chce Perl
    - Prispôsobiť mená správ
    - Prispôsobiť dátové typy
  - KISS: Keep it Simple
  - Zložitejšie služby: mod_perl a vlastný handler

> If it happens, it must be possible.
>
> -- Rule of Practice

PHP SOAP: Základy

- NuSOAP (http://dietrich.ganx4.com/)
  - Library (nusoap.php)
  - Pure PHP
  - Často používaný
- PEAR::SOAP
  - Pure PHP
- PHP-SOAP
  - Použité C moduly
  - Limitovaná funkčnosť

BGS
Business Global Systems

PHP SOAP: Client (NuSOAP)

```
<?php
require_once('nusoap.php');

$client = new soap_client('stockservice.wsdl','wsdl');
$proxy = $client->getProxy();

$price = $proxy->getStockPrice($stockSymbol);

print $price;
?>
```

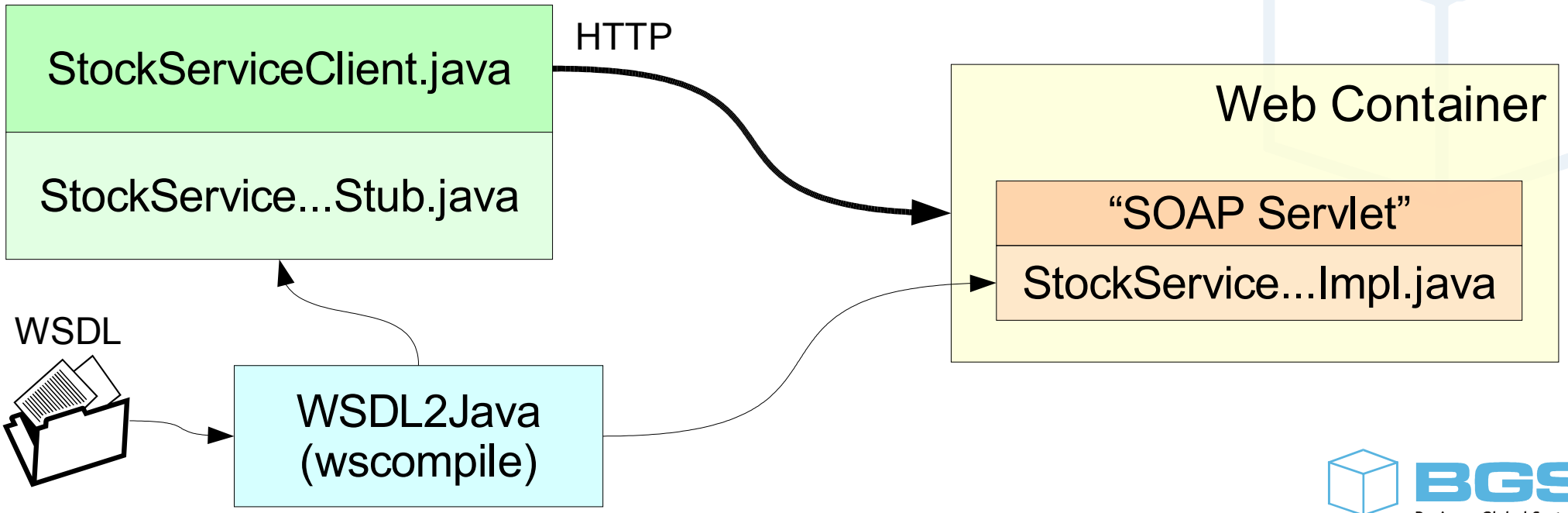
PHP SOAP: Server (NuSOAP)

```
<?php
require_once('nusoap.php');

$server = new soap_server;
$server->register('getStockPrice');

function getStockPrice ($stockSymbol) {
   return (rand(80001)/1000);
}

$server->service($_SERVER['HTTP_RAW_POST_DATA']);
exit();
?>
```

Zhodnotenie: PHP

Java SOAP: Základy

- Web Services Development Kit (Sun)
  - Komerčný
- Axis (Apache project)
  - Free



BGS
Business Global Systems

Java SOAP: Client (Axis)

```
$ java org.apache.axis.wsdl.WSDL2Java stockservice.wsdl
```

```
import ...
import sk.alert.storm.StockService.*;

public class StockServiceClient {

    public static void main(String argv[]) {

        StockService euris = new StockServiceLocator();
        StockServicePortType stub = euris.getStockServicePortType();

        System.out.println(stub.getStockPrice("SUNW"));

    }
}
```

BGS
Business Global Systems

Java SOAP: Server (Axis)

```
$ java org.apache.axis.wsdl.WSDL2Java --server-side stockservice.wsdl
```

```
import java.lang.*;

package sk.alert.storm.StockService;

public class StockServiceSoapBindingImpl implements
                    sk.alert.storm.StockService.StockServicePortType
{

    public float getStockPrice(String stockSymbol)
                        throws java.rmi.RemoteException
    {
        return random()*80;
    }

}
```

Zhodnotenie: Java

- Kompilovaný, typový jazyk
  - “Stub” a “Skeleton” generovaný z WSDL
- Čisté riešenie
  - Zapadá do existujúcej štruktúry (java.rmi.Remote a spol.)
- V budúcnosti priama väzba na EJB
- V budúcnosti: Security, Transakcie, ...

- Axis/Tomcat deployment nie vždy bez problémov

Záver

Záver

- Think “loose coupling”
- Use standards
- Keep it simple

Everything should be as simple as possible, and no simpler.
-- Albert Einstein

Questions?

Ďakujem za pozornosť

... a za pomoc pri tvorbe prednášky:

Pavol Repčík (off): Java príklady
lharp, slnce: PHP príklady

Ing. Radovan Semančík

Business Global Systems, a.s.
Pluhová 2
83248 Bratislava

semancik@bgs.sk