

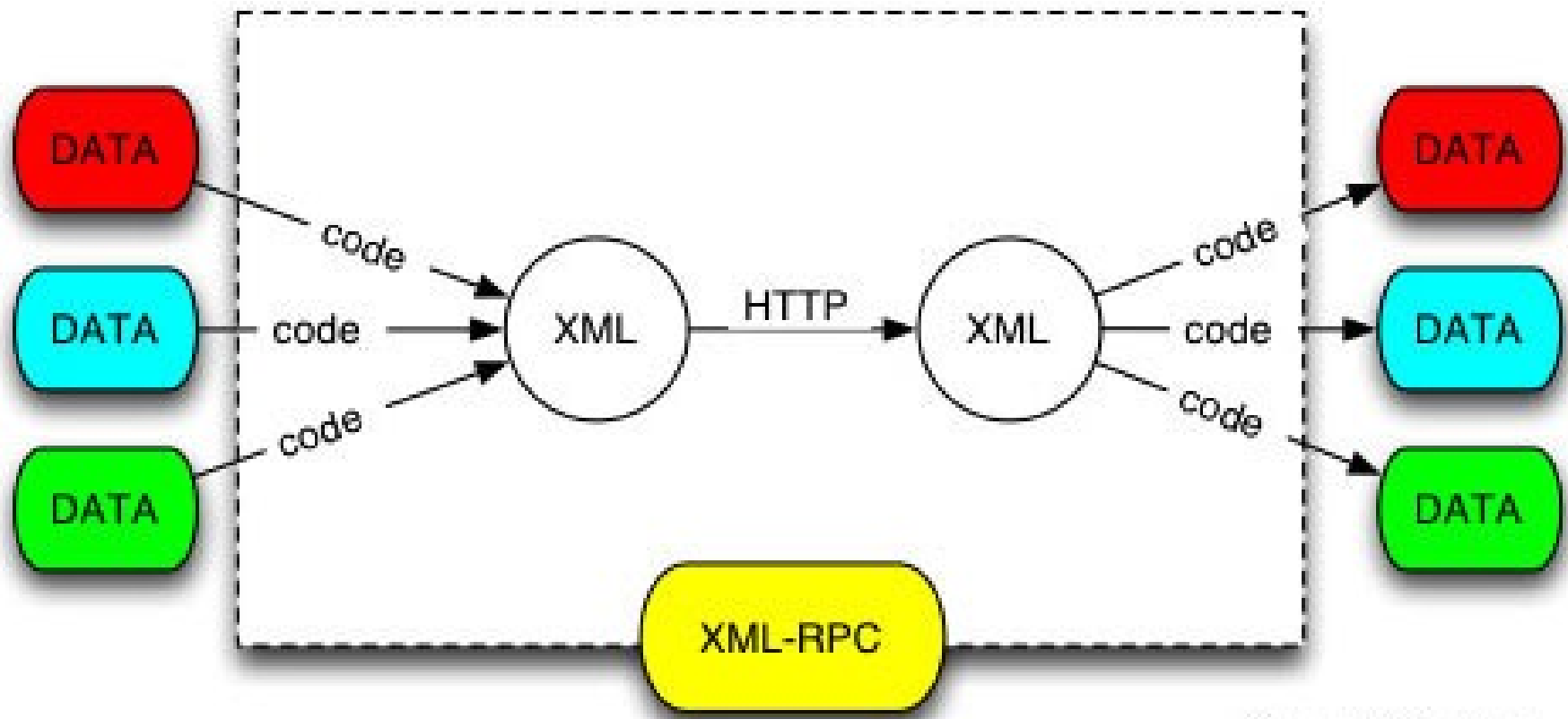


X36DSV – 4. cvičení

XML-RPC (eXtensible Markup Language – Remote Procedure Call)



XML-RPC schéma



Source: JY Stervinou



XML-RPC - naming

- Jmenná služba
 - globální neexistuje
 - jen v rámci serveru
 - mapování 'jméno' → 'třída'
 - částečnou náhradou je introspekce
 - server poskytuje jisté 'neoficiální' metody
 - `system.listMethods`
 - `system.methodSignature`
 - `system.methodHelp`



XML-RPC - typy

- Datové typy
 - základní
 - `int (i4)`
 - `double`
 - `string`
 - `dateTime.iso8601`
 - `base64`
 - `boolean`
 - složené
 - `struct`
 - `array`



XML-RPC request

```
POST /RPC2 HTTP/1.0
User-Agent: Frontier/5.1.2 (WinNT)
Host: betty.userland.com
Content-Type: text/xml
Content-length: 181
```

```
<?xml version="1.0"?>
<methodCall>
  <methodName>examples.getStateName</methodName>
  <params>
    <param>
      <value><i4>41</i4></value>
    </param>
  </params>
</methodCall>
```



XML-RPC response

```
HTTP/1.1 200 OK
Connection: close
Content-Length: 158
Content-Type: text/xml
Date: Fri, 17 Jul 1998 19:55:08 GMT
Server: UserLand Frontier/5.1.2-WinNT
```

```
<?xml version="1.0"?>
<methodResponse>
  <params>
    <param>
      <value><string>South Dakota</string></value>
    </param>
  </params>
</methodResponse>
```



XML-RPC error

```
HTTP/1.1 200 OK  
Connection: close  
...
```

```
<?xml version="1.0"?>  
<methodResponse>  
  <fault>  
    <value>  
      <struct>  
        <member>  
          <name>faultCode</name>  
          <value><int>4</int></value>  
        </member>  
        <member>  
          <name>faultString</name>  
          <value><string>Too many parameters.</string></value>  
        </member>  
      </struct>  
    </value>  
  </fault>  
</methodResponse>
```



XML-RPC server

- XmlRpcServerConfig
 - nastavení vlastností serveru
 - `setEnabledForExtensions`, `setContentLengthOptional`, ...
- PropertyHandlerMapping
 - mapování názvů na samotné třídy
 - `addHandler`, ...



XML-RPC příklad (server)

```
public static void main(String[] args) throws Exception {
    // start web server
    WebServer webServer = new WebServer(port);
    // get the XML-RPC Server
    XmlRpcServer xmlRpcServer = webServer.getXmlRpcServer();
    PropertyHandlerMapping phm = new PropertyHandlerMapping();
    phm.addHandler("CalculatorImpl", CalculatorImpl.class);
    xmlRpcServer.setHandlerMapping(phm);
    XmlRpcServerConfigImpl serverConfig =
        (XmlRpcServerConfigImpl) xmlRpcServer.getConfig();
    serverConfig.setEnabledForExtensions(true);
    serverConfig.setContentLengthOptional(false);
    webServer.start();
}
```



XML-RPC klient

- ClientConfig
 - nastavení vlastností klienta
 - basicUserName, basicPassword, enabledForExtensions, ...
- TransportFactory
 - stará se o samotný přenos
 - XmlRpcSunHttpTransportFactory, XmlRpcCommonsTransportFactory, ...
- XmlWriterFactory
 - stará se o vytvoření XML zprávy



XML-RPC příklad (klient)

```
public static void main(String[] args) throws Exception {
    // create configuration
    XmlRpcClientConfigImpl config = new XmlRpcClientConfigImpl();
    config.setServerURL(new URL("http://127.0.0.1:8080/xmlrpc"));
    config.setEnabledForExtensions(true);
    config.setConnectionTimeout(60 * 1000);
    config.setReplyTimeout(60 * 1000);
    XmlRpcClient client = new XmlRpcClient();
    // use Commons HttpClient as transport
    client.setTransportFactory(
        new XmlRpcCommonsTransportFactory(client));
    // set configuration
    client.setConfig(config);
    // make the a regular call
    Object[] params1 =
        new Object[] { new Integer(2), new Integer(3) };
    Integer result1 =
        (Integer) client.execute("CalculatorImpl.add", params1);
    System.out.println("2 + 3 = " + result1);
}
```



XML-RPC materiály

- <http://en.wikipedia.org/wiki/XML-RPC>
- www.xmlrpc.com
- <http://ws.apache.org/xmlrpc>