



XML applications for a MySQL database

A practical approach

Madrid, November 23th. 2004

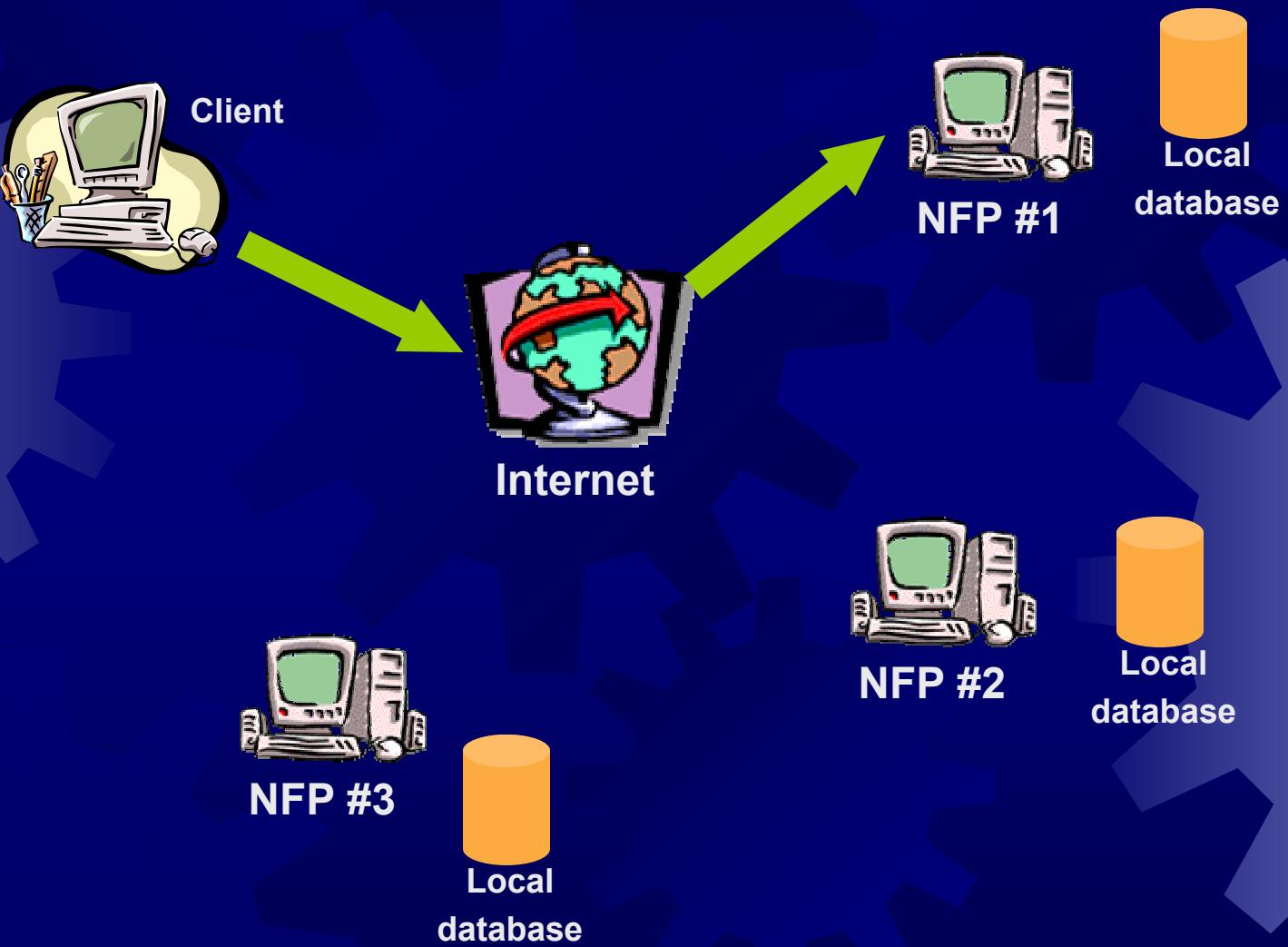
Angel J. Martín Sesma (CEDEX)
IT responsible of the Spanish NFP
<http://hispagua.cedex.es>



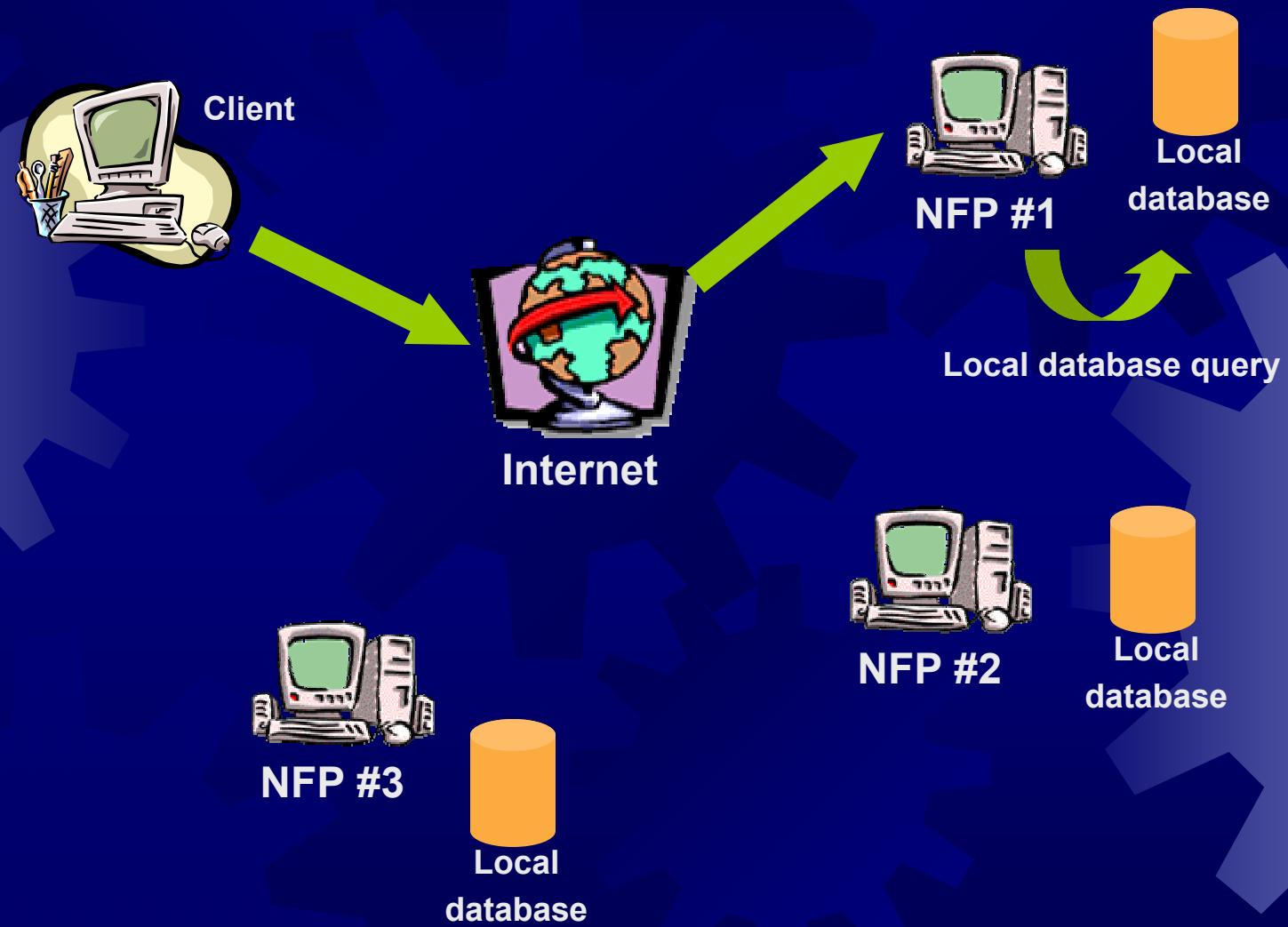
Actual scenario

- ✿ Different hardware & software NFP's web servers
- ✿ National Focal Points using different database engines
- ✿ Non-existent unified database structures and contents
- ✿ Only local database queries can be processed on each NFP

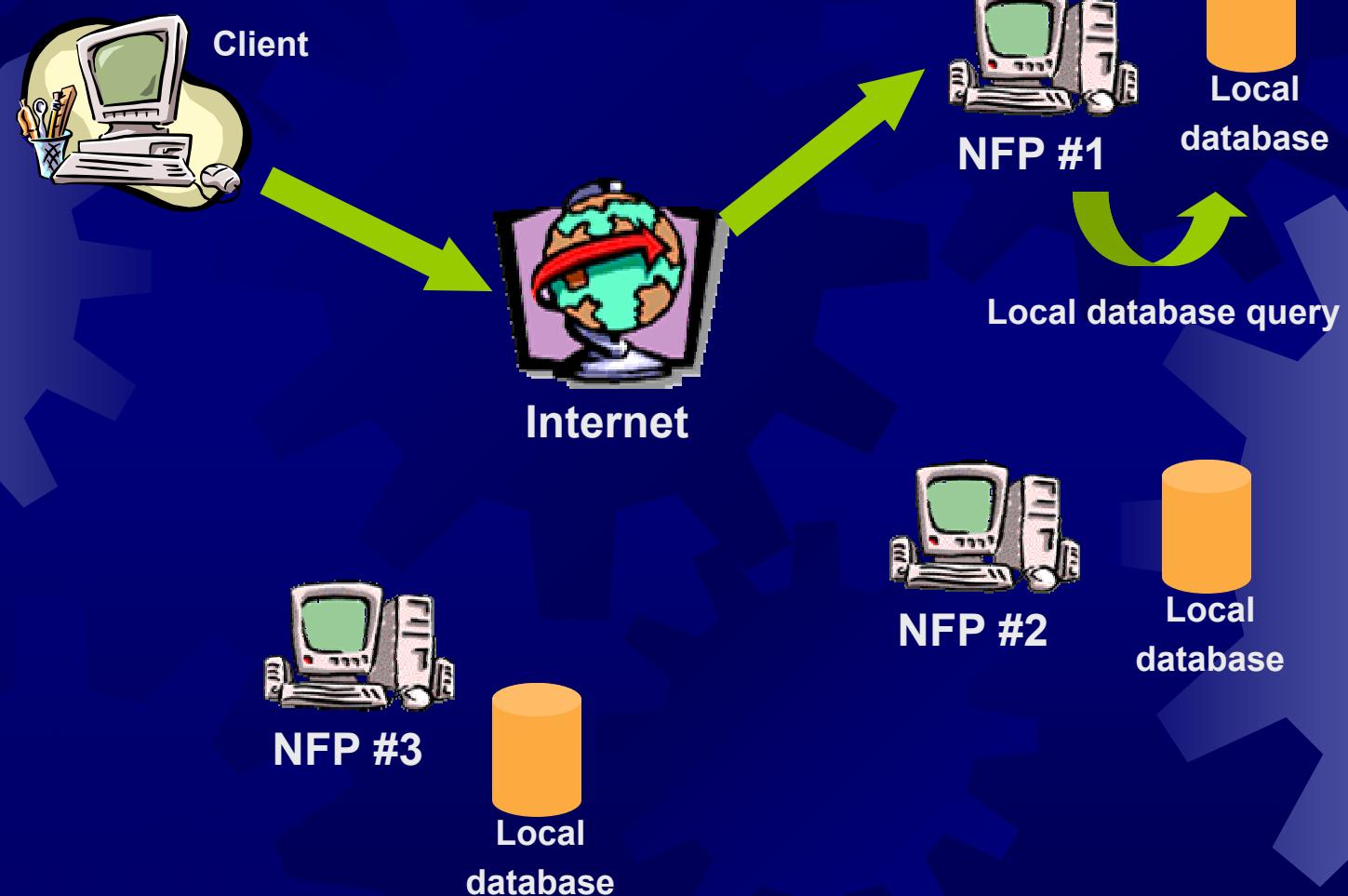
Actual scenario



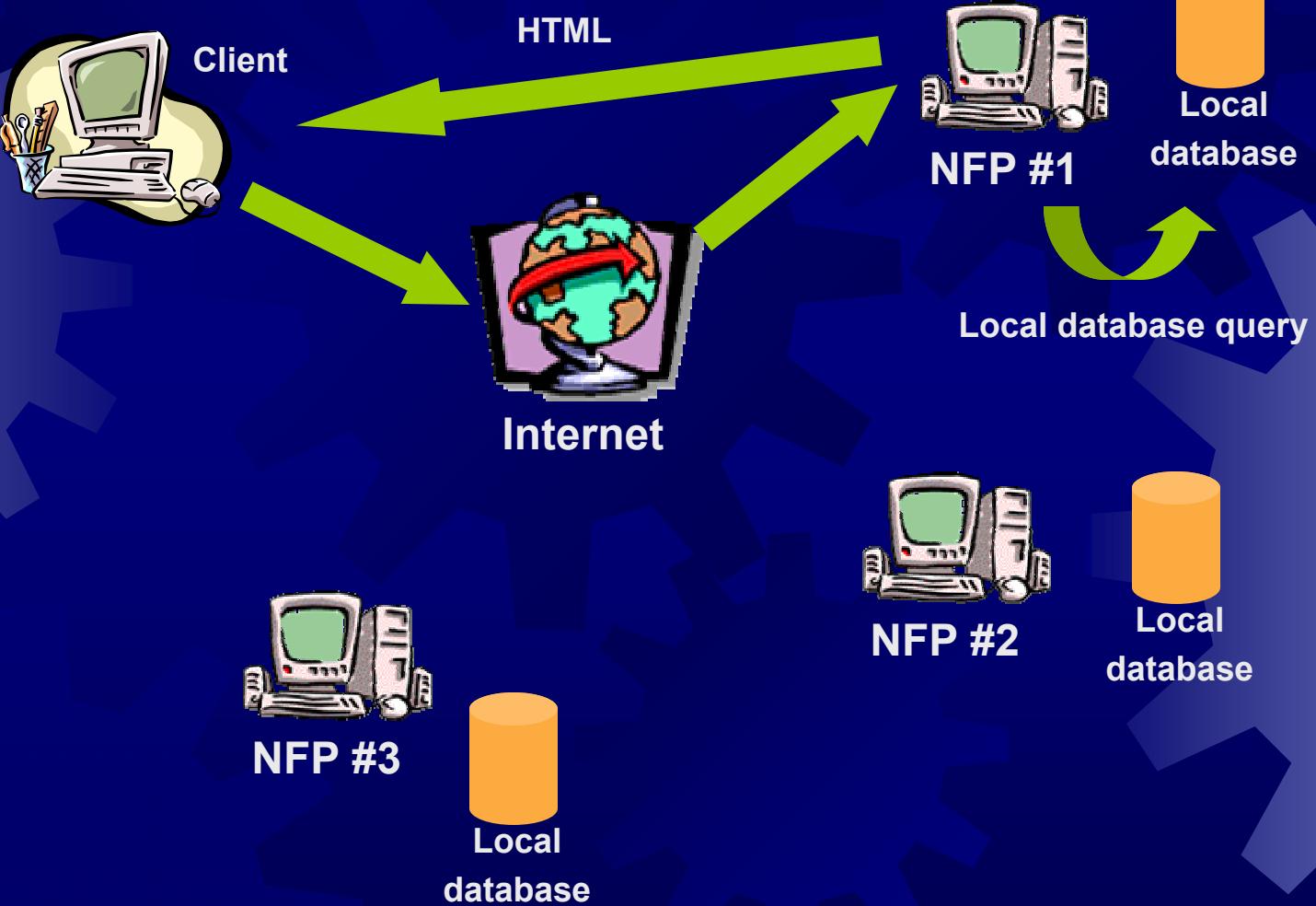
Actual scenario



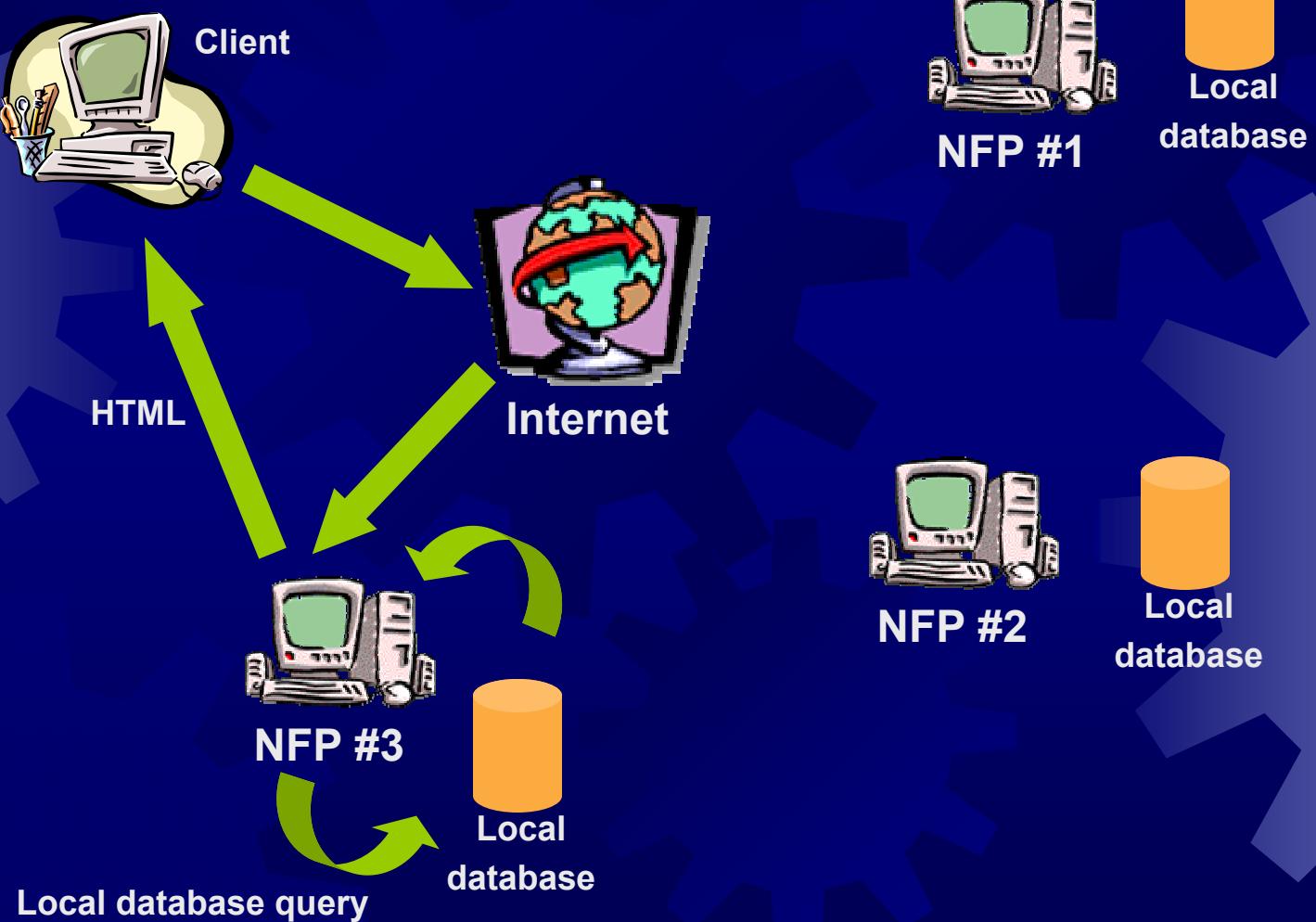
Actual scenario



Actual scenario



Actual scenario



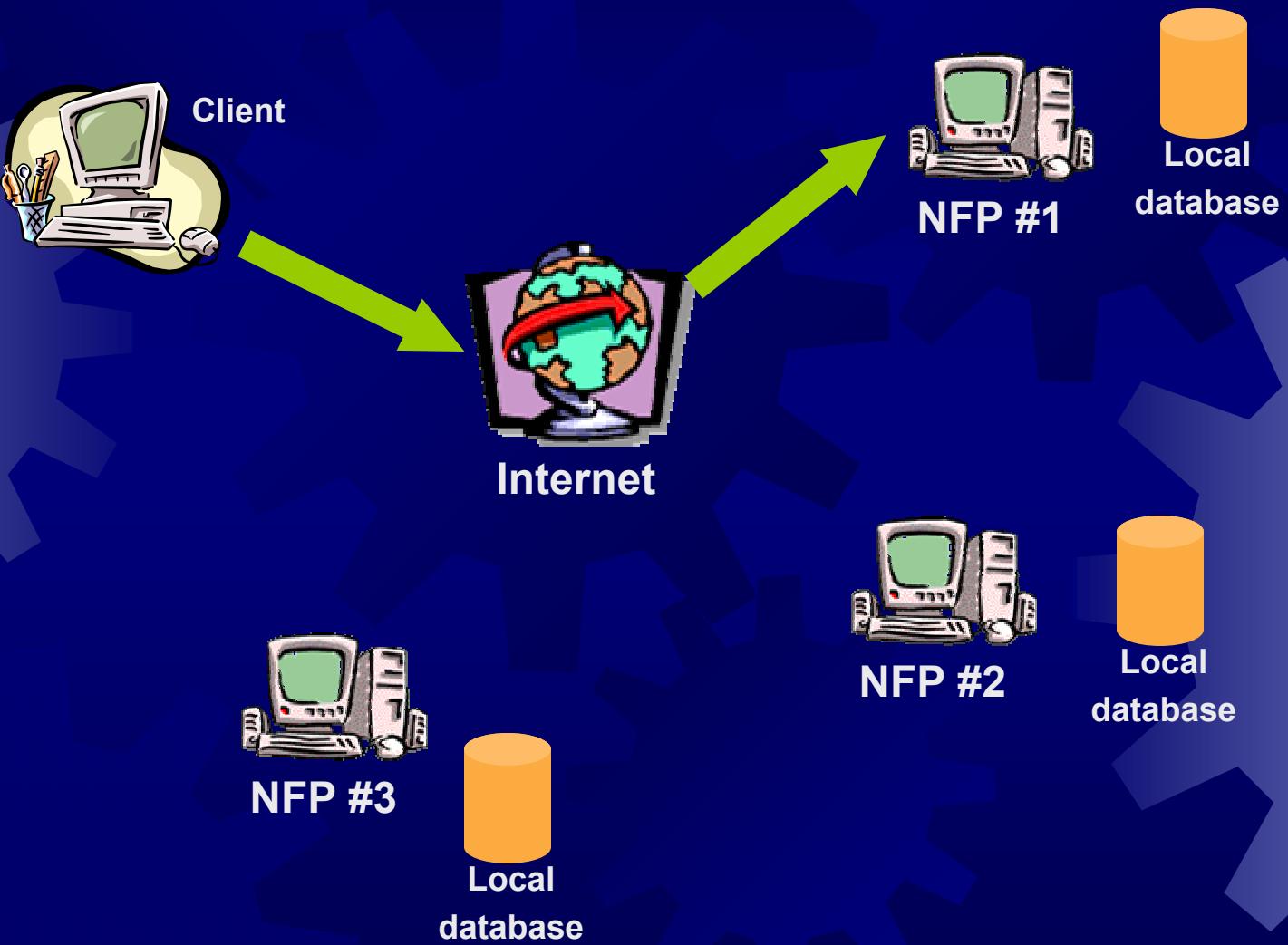


XML as a standard for information exchange

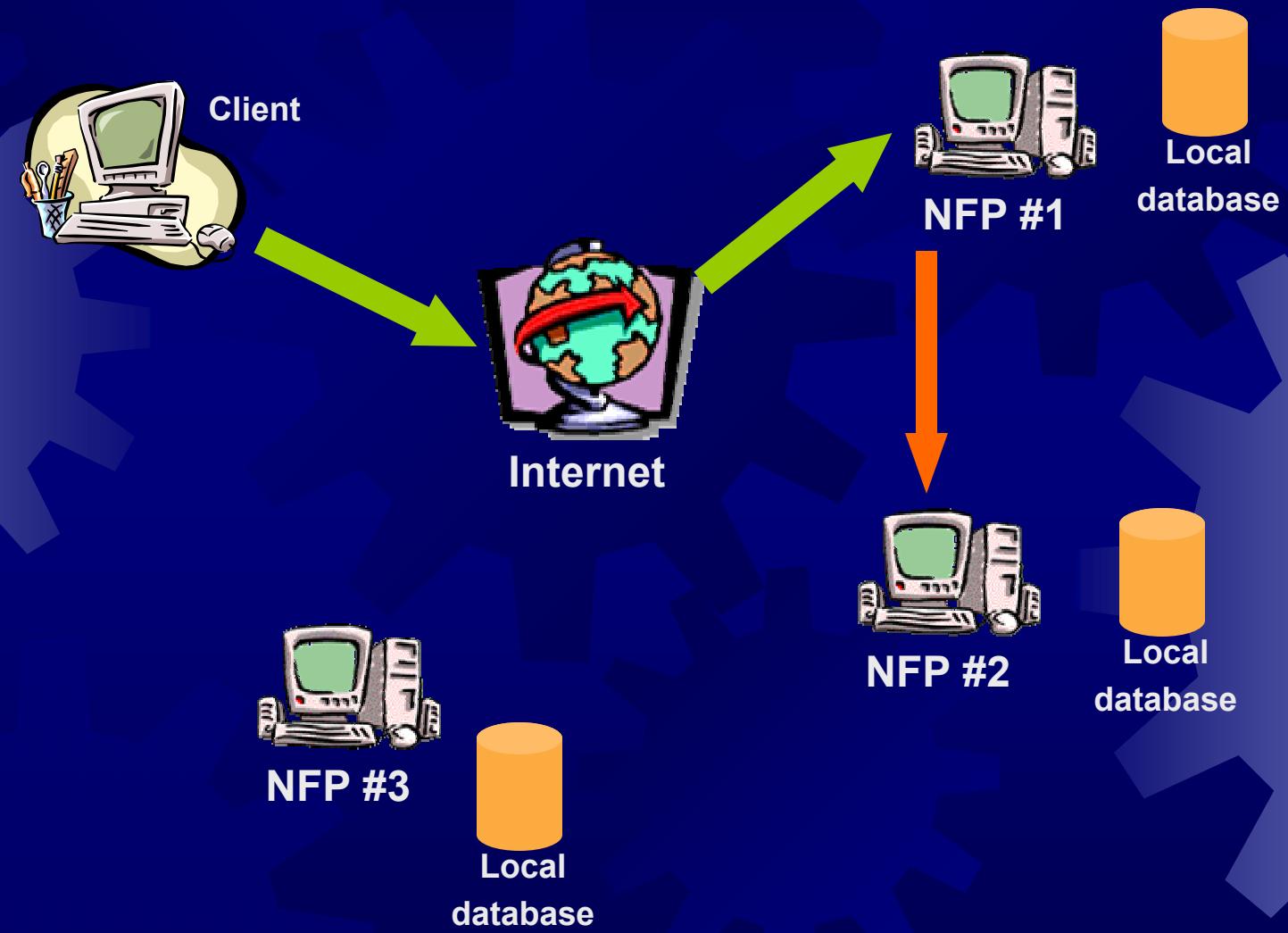
- ★ XML is a international standard
- ★ Most database engines has XML import/export procedures
- ★ Ease to program applications to generate XML files using database queries
- ★ Retrieves data from different NFP's databases using remote queries



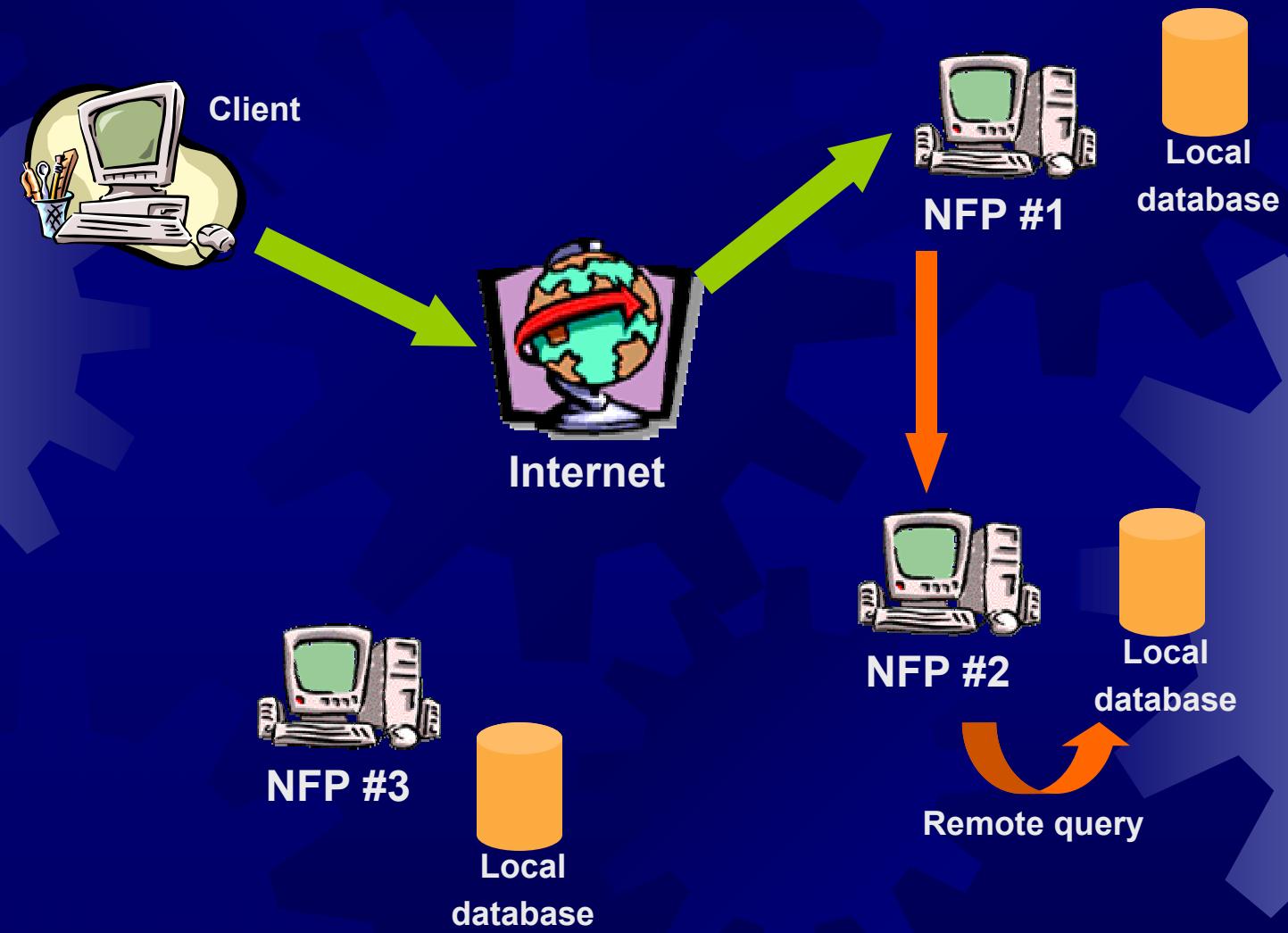
Proposed scenario



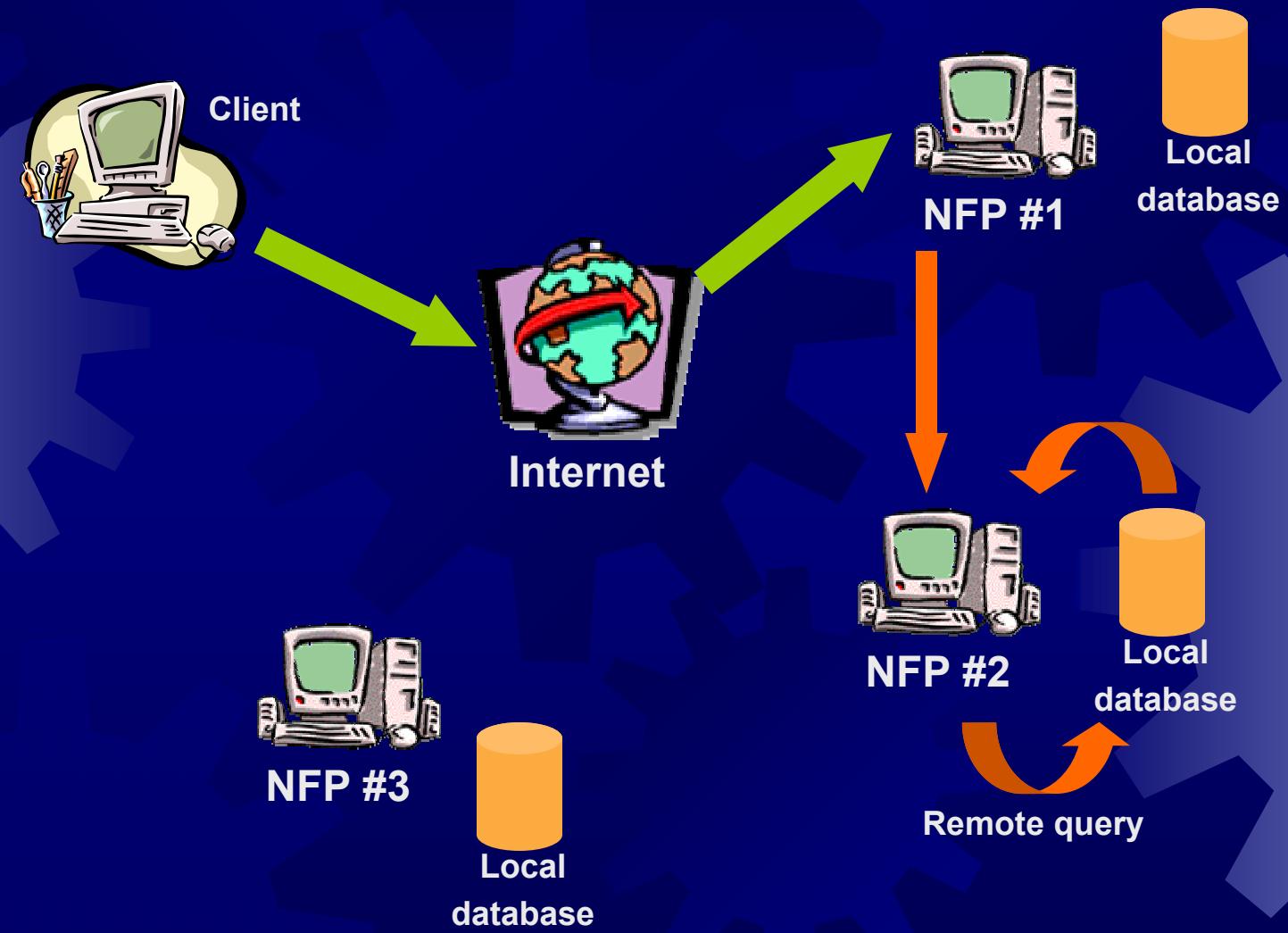
Proposed scenario



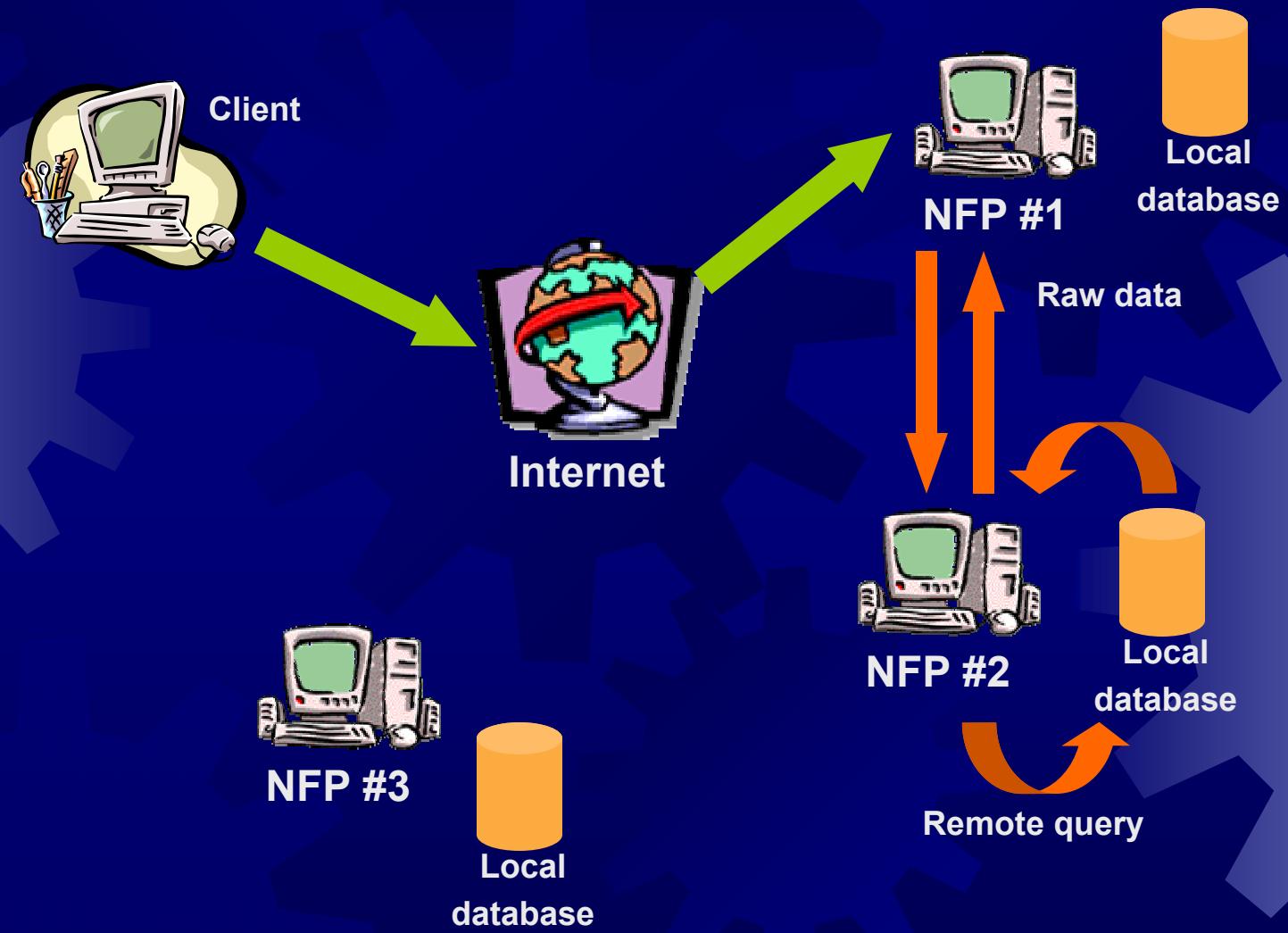
Proposed scenario



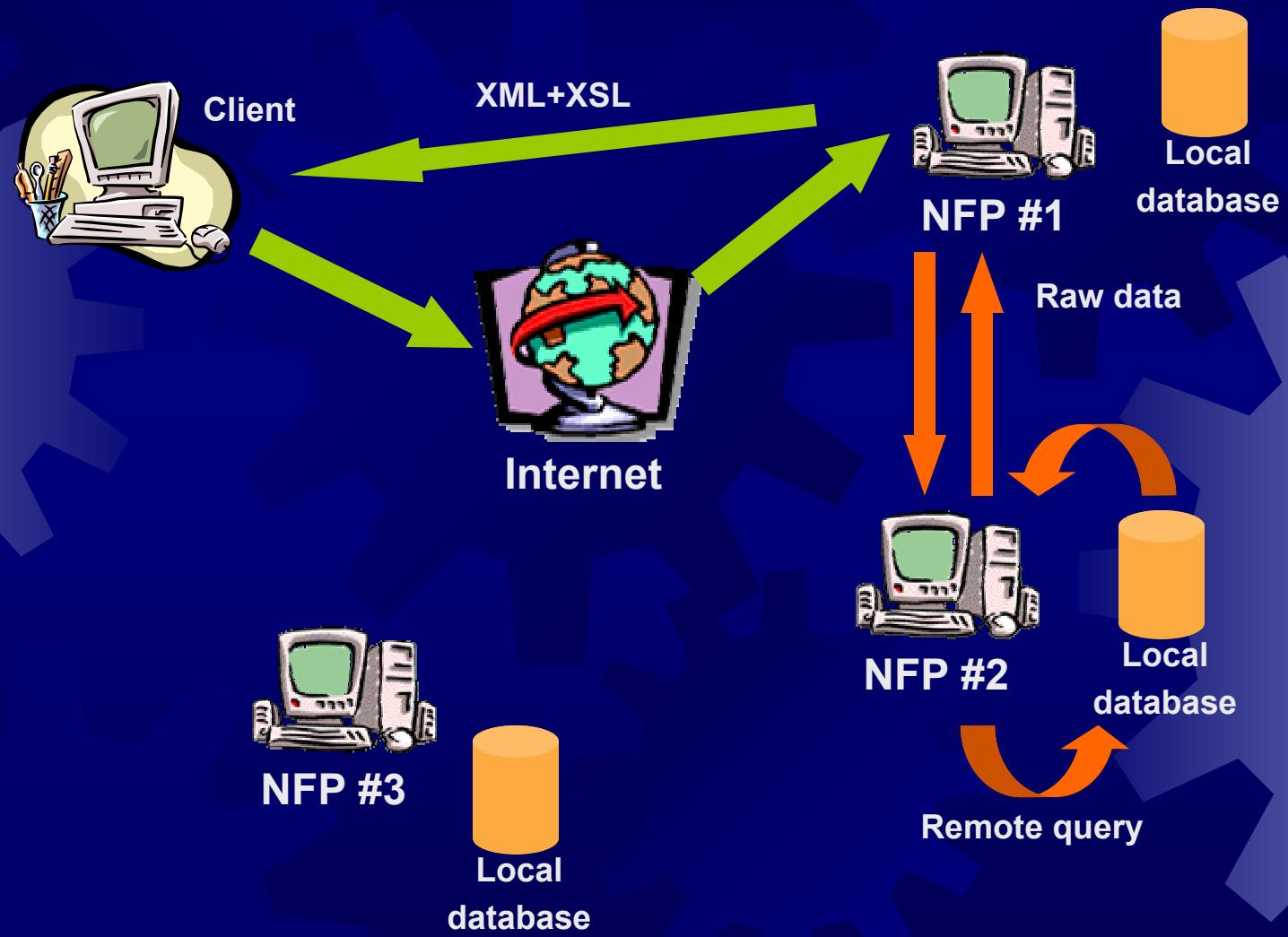
Proposed scenario



Proposed scenario



Proposed scenario





A practical approach

- ★ The goal: Obtain remote data from a MySQL database located at <http://hispagua.cedex.es> and generate a XML file
- ★ Our local system is an Apache web server with PHP and MySQL implemented on it



Obtaining data in XML format

- ★ In our server we have a small application written in PHP to make a remote query over a Hispagua's remote MySQL database
- ★ The results of the query will be converted to a local XML “well formed” file

Obtaining data in XML format

```
<?PHP
```

```
$dbname="hispagua"; //This the MySQL database  
$connection=mysql_connect("hispagua.cedex.es","anonymous") or die  
("Couldn't connect to database."); //Establishes the connection to the  
database  
$table_name="empresas"; //This the table where we are going to search in  
$db=mysql_select_db($dbname,$connection); //Selects the current database  
$query="select * from ".$table_name; //This is the SQL query  
$result=mysql_query($query,$connection) or die("Couldn't execute the  
query."); //Executes the query and stores the result in a variable  
$num=mysql_num_rows($result); //Counts numbers of rows returned by the  
query  
if ($num != 0) //If there are results of the query, generates XML code
```



Obtaining data in XML format

```
{ $file= fopen("results.xml", "w"); //Opens the file
$xml ="<?xml version=\"1.0\" ?>\r\n";
$xml .="<?xml:stylesheet type=\"text/xsl\" href=\"style.xsl\" ?>\r\n";
$xml .="<contacts>\r\n";
while ($row = mysql_fetch_array($result)) //Converts each row of the results in
an array and generates its XML code
{
    if ($row["id"])
    {
        $xml .="\t<entity name=\"" . $row["organismo"] . "\">>\r\n";
        $xml .="\t\t<address>" . $row["direccion"] . "</address>\r\n";
        $xml .="\t\t<city>" . $row["ciudad"] . "</city>\r\n";
        $xml .="\t\t<province>" . $row["provincia"] . "</province>\r\n";
        $xml .="\t\t<country>" . $row["pais"] . "</country>\r\n";
        $xml .="\t</entity>\r\n";
    } else
    {
        $xml .="\t<entity name=\"Nothing Returned\">\r\n";
        $xml .="\t\t<address>none</address>\r\n";
        $xml .="\t\t<city>none</city>\r\n";
        $xml .="\t\t<province>none</province>\r\n";
        $xml .="\t\t<country>none</country>\r\n";
        $xml .="\t</entity>\r\n";
    }
}
```

Obtaining data in XML format

```
$_xml .="</contacts>"; //Finishes the XML code  
fwrite($file, $_xml); //Writes and closes the file  
fclose($file);  
echo "XML has been written. <a href=\"results.xml\">View the XML.</a>";  
}  
else //If there are no results of the query  
{  
    echo "No Records found";  
}  
?>
```





What's next?

- ★ Now we have the requested data in our local system, we can process it on several ways
 - ★ Data can be shown on a web browser using a local XSL style sheet from our web server
 - ★ Retrieved data can be transferred to a local database in our MySQL engine using a third party tool as PremiumSoft Navicat





Showing XML data on a web browser

- ★ When the XML code was generated, we inserted this line on it:

```
<?xml:stylesheet type="text/xsl" href="style.xsl"?>
```

- ★ This declares the style sheet that must be used to show the file on a web browser
- ★ This technique converts the XML file in a HTML file on the client side



Showing XML data on a web browser



Converting XML into HTML using a XSL sheet

```
<?xml version="1.0" ?>
<html xmlns:xsl="http://www.w3.org/TR/WD-xsl">
<body>
<h1>Results of the query on database empresas@hispania.cedex.es</h1>
    <xsl:for-each select="site/page">
        <h2>Entity name: <xsl:value-of select="name"/></h2>
        <ul>
            <li>Address: <xsl:value-of select="address"/></li>
            <li>City: <xsl:value-of select="city"/></li>
            <li>Province: <xsl:value-of select="province"/></li>
            <li>Country: <xsl:value-of select="country"/></li>
            <li>Phone: <xsl:value-of select="phone"/></li>
        </ul>
        <hr/>
    </xsl:for-each>
</body>
</html>
```



Converting XML into HTML using a XSL sheet

```
<?xml version="1.0" ?>
<html xmlns:xsl="http://www.w3.org/TR/WD-xsl">
<body>
<h1>Results of the query on database empresas@hispania.cedex.es</h1>
    <xsl:for-each select="site/page">
        <h2>Entity name: <xsl:value-of select="name"/></h2>
        <ul>
            <li>Address: <xsl:value-of select="address"/></li>
            <li>City: <xsl:value-of select="city"/></li>
            <li>Province: <xsl:value-of select="province"/></li>
            <li>Country: <xsl:value-of select="country"/></li>
            <li>Phone: <xsl:value-of select="phone"/></li>
        </ul>
        <hr/>
    </xsl:for-each>
</body>
</html>
```





Practical use of this technique

- ★ This is a easy and fast way to access to remote NFP's contents using a client/server structure. This preserves databases of external manipulations
- ★ Allows a NFP to show other NFP's information using its own web style



Final Considerations

- ★ The implementation of this technique requires some additional work:
 - ★ Every NFP has to publish a index containing the databases that can be accessed this way and its internal structure
 - ★ The security rules of every NFP needs to be modified to allow remote queries, opening the corresponding ports on its firewalls
 - ★ It would be recommendable to have a common DTD specification defining the structure of these XML files