Web + Stata

Alexander Zlotnik

Technical University of Madrid, ETSIT, DIE
Ramon y Cajal University Hospital

Stata is a registered trademark of StataCorp LP, College Station, TX, USA.

User-contributed **programs**

ssc install <program>

findit <program>
(runs both **search** and **net search**)

net from http://www.website.com/

manually copy program files to
C:\ado\plus\<subdir>

Sometimes this is not enough

Sometimes your **program**...

... requires **complex interactions** with **external software packages**
(ex: WinBUGS, MATLAB, Maxima, AnyLogic)

... uses **proprietary data sources**
(ex: real-time currency exchange rates)

... uses **proprietary source code**

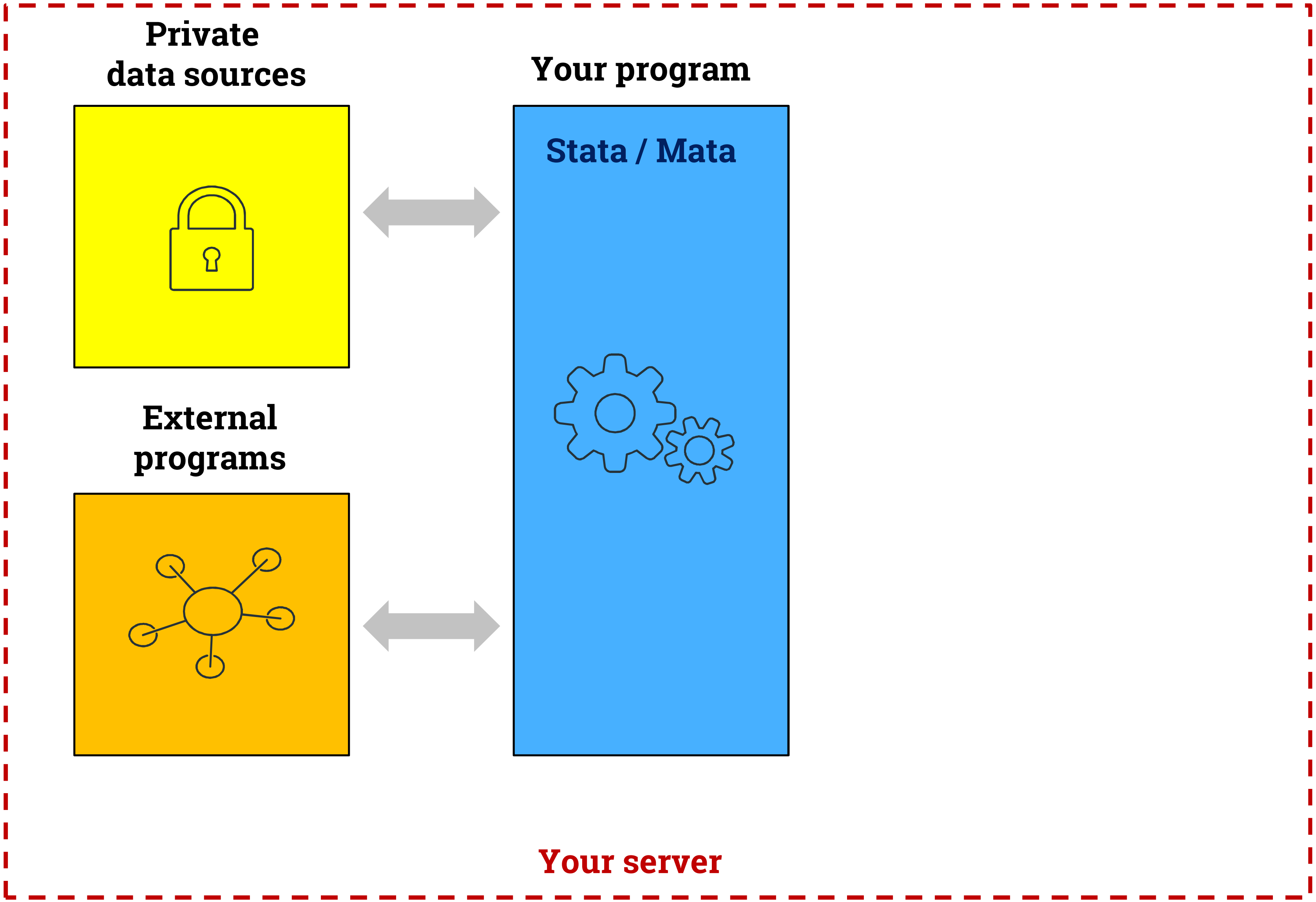
Sometimes your **audience**…

… does **not** have the **version of Stata** your program requires

… does **not** have **Stata at all**

… does **not** have a PC, but may have a *smartphone* with a web browser

What if…?



Private data sources

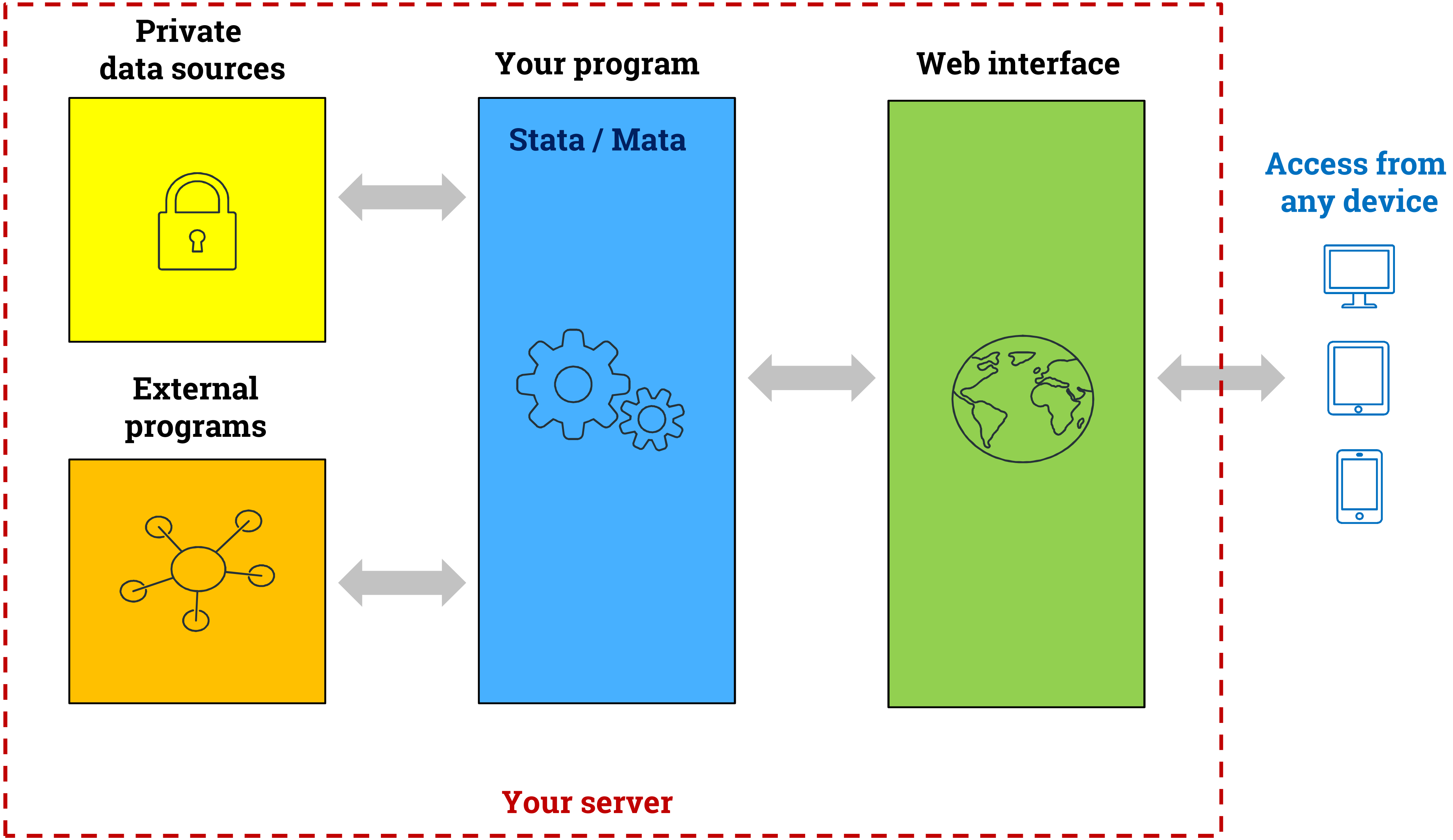
Your program

Stata / Mata

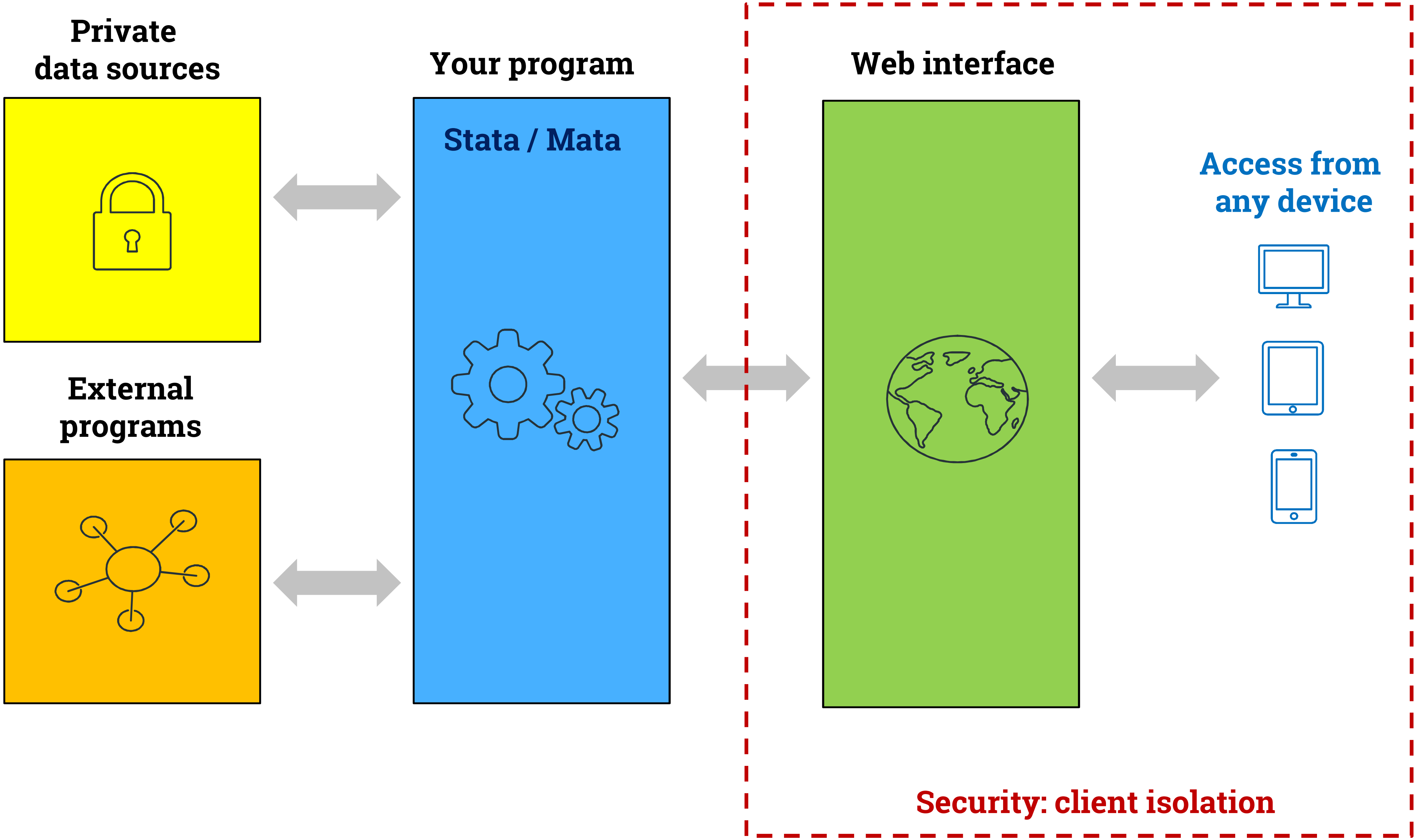
External programs

Your server

What if…?


Private data sources
Your program
Web interface
Stata / Mata
Access from any device
External programs
Your server

What if...?



Private data sources

External programs

Your program

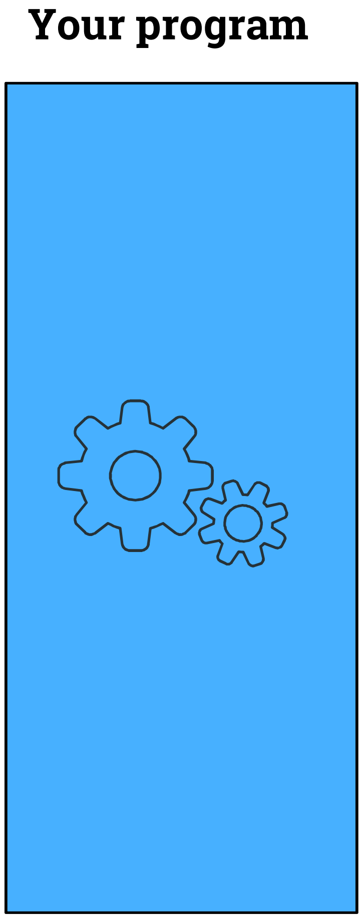
Stata / Mata

Web interface

Access from any device

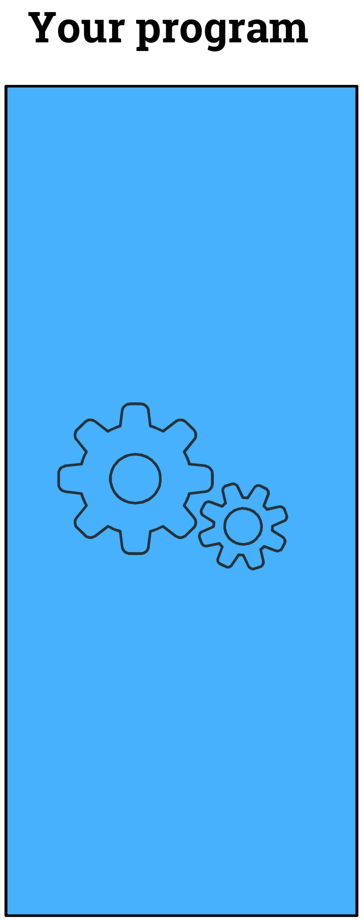
Security: client isolation

How?



Option 1:
Translate Stata / Mata program into
a **general-purpose programming language** used
in web applications.
Ex: Java, C / C++, C#, ASP.net + VB.net, Python,
Ruby, etc

How?



Option 1:
Translate Stata / Mata program into
a **general-purpose programming language** used
in web applications.
Ex: Java, C / C++, C#, ASP.net + VB.net, Python,
Ruby, etc

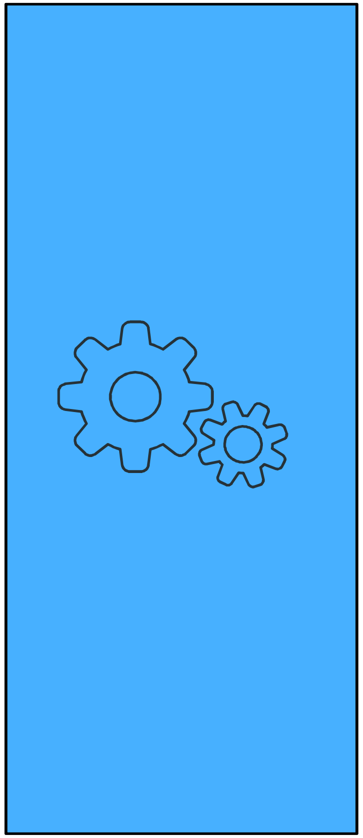
- **Few numerical libraries**

- **May not have the same functions**

- **Functions may not be implemented in the same way**
  - **subtle errors**
  - **numerical precision issues**
  - **performance issues**

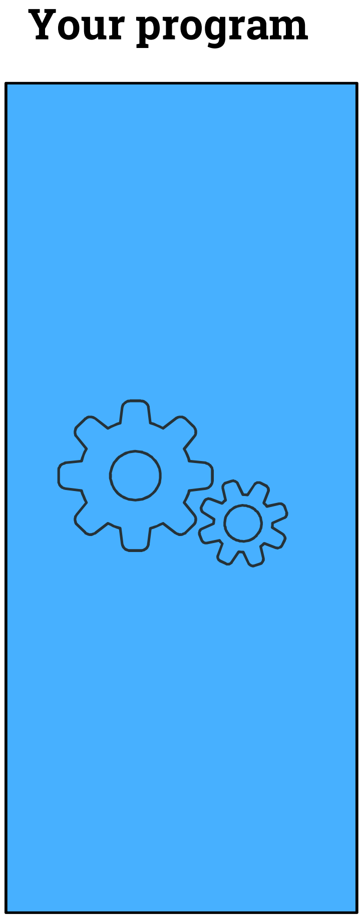
How?

Your program



Option 2:
Translate Stata / Mata program into
R & RShiny or **SAS Stored Process Web Application**

How?



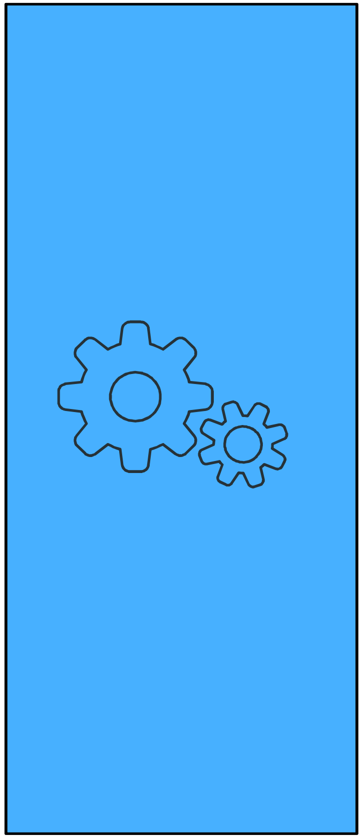
Option 2:
Translate Stata / Mata program into
R & RShiny or **SAS Stored Process Web Application**

- Still requires a laborious translation in most cases

- Again, functions may not be implemented in the same way

- RShiny is a nice alternative but the free version only supports one concurrent session

How?



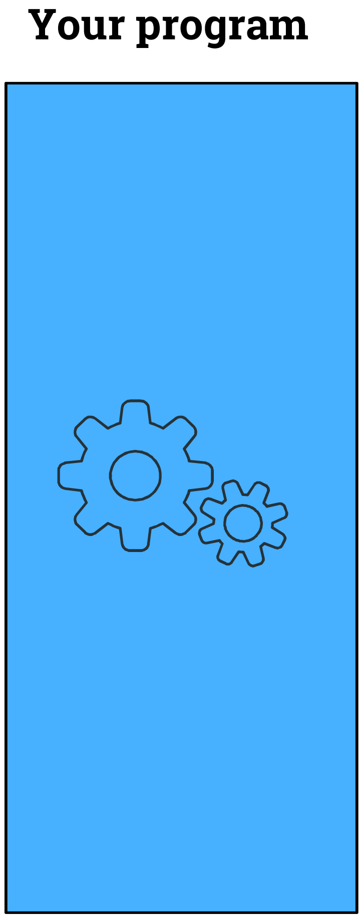
Option 3:
Use a slightly modified version of your existing **Stata** program in a **web application**.

Stata/MP Stata/SE Stata/IC

NUMERICS by Stata

Stata is a registered trademark of StataCorp LP, College Station, TX, USA, and the Stata logo is used with the permission of StataCorp.

How?



Option 3:
Use a slightly modified version of your existing **Stata** program in a **web application**.

-- In this presentation, we will see how to build a **web application** using **your Stata program**, with **minimal modifications** based on Stata/IC, Stata/SE or Stata/MP.

-- Very similar techniques can be used with Numerics for Stata.

Technologies

Program core: **Stata + Mata**

Web application language: **PHP**

Web server: **Apache**

Operating system: **Windows**

Technologies

Program core: **Stata + Mata**

Web application language: **PHP**

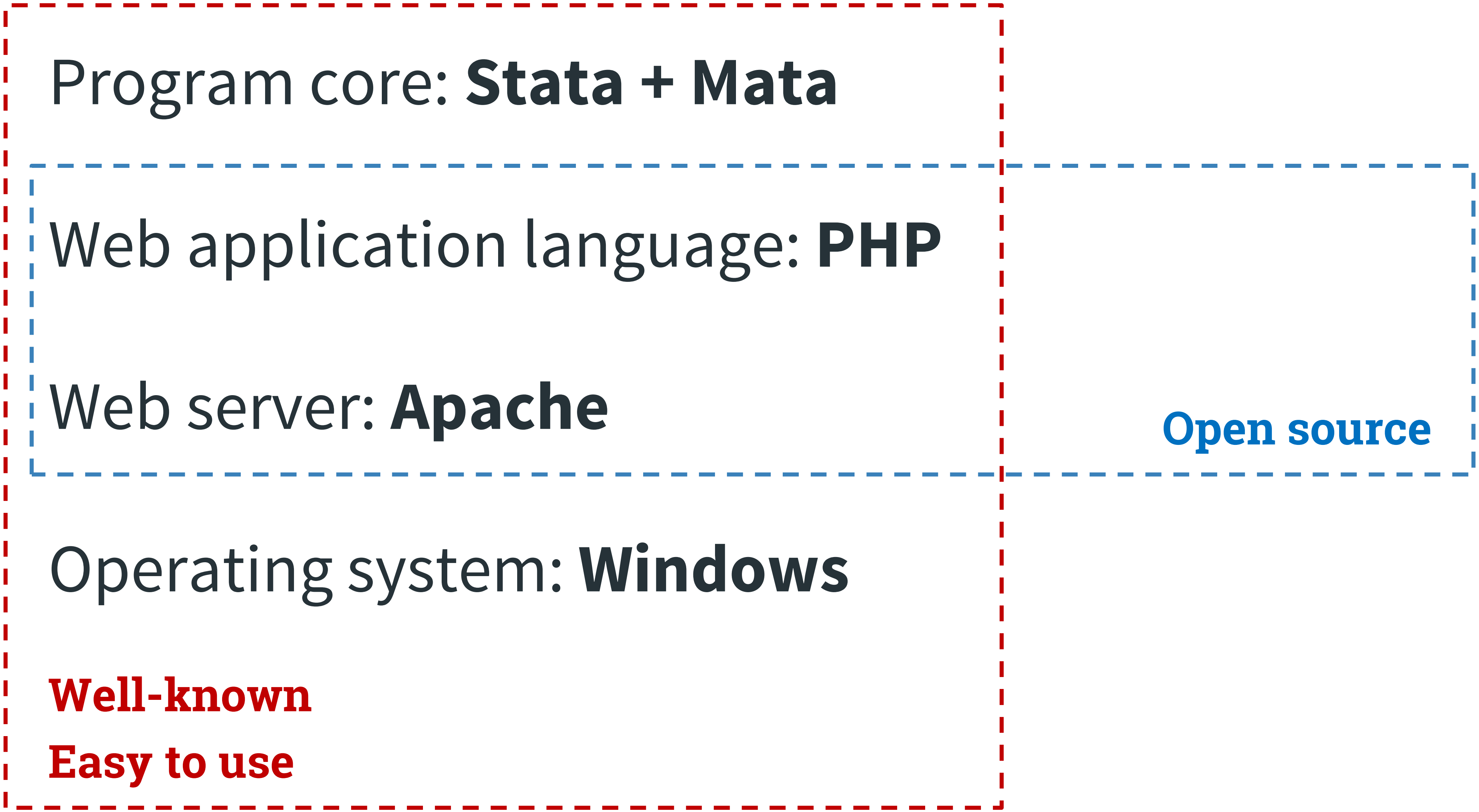
Web server: **Apache**

Operating system: **Windows**

Well-known

Easy to use

Technologies



Program core: **Stata + Mata**

Web application language: **PHP**

Web server: **Apache**

Open source

Operating system: **Windows**

Well-known

Easy to use

Web application language

PHP implementation **example**

Other languages may also be used:
- Java (servlets, JSPs)
- Python
- ASP / ASP.net + C# / VB.net
- C/C++, Perl (CGI interface)
- *et cetera*

Web server

Apache implementation **example**

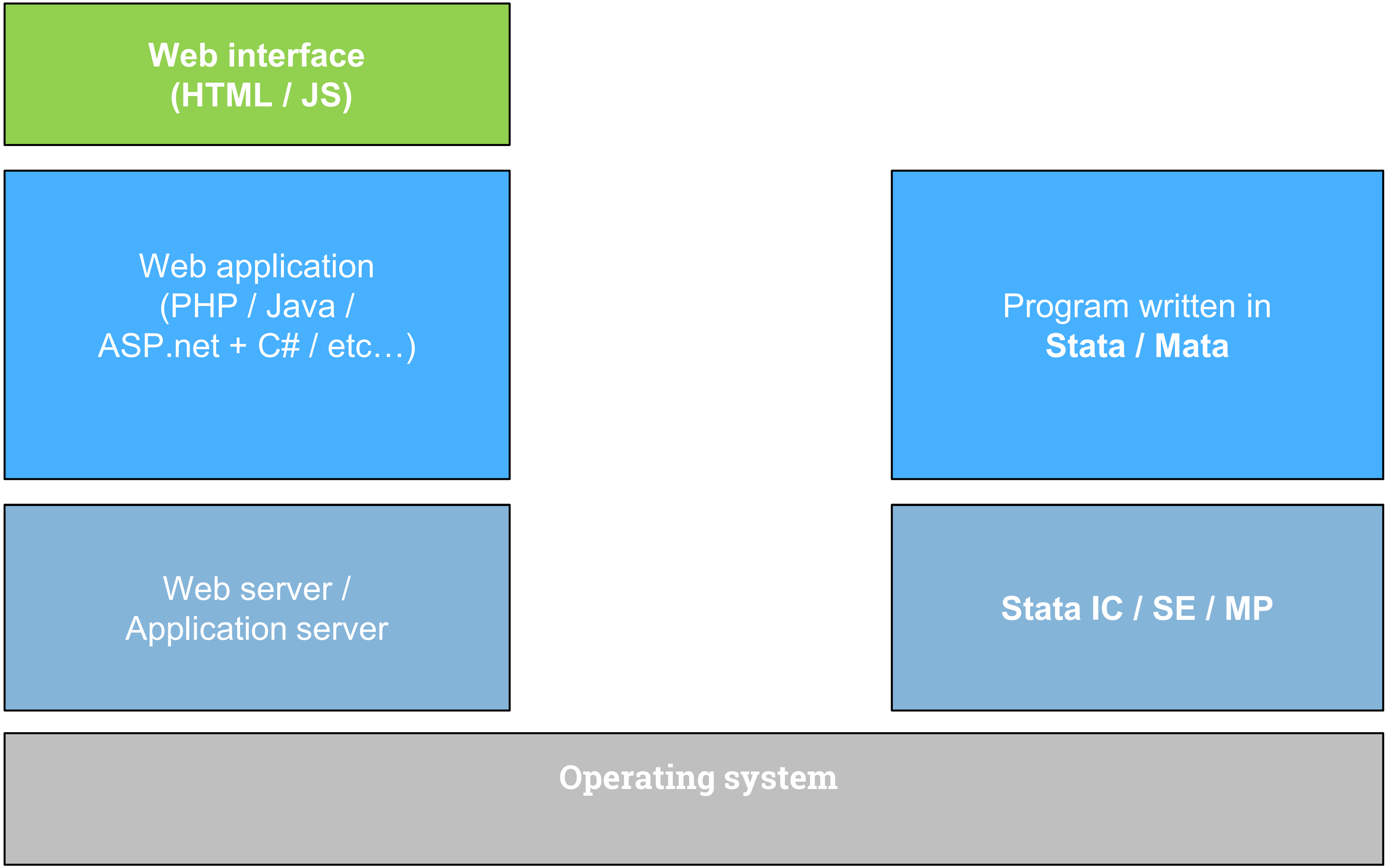
Other web servers, application containers and application servers may also be used:

- Tomcat
- JBoss
- Oracle WebLogic
- IBM WebSphere
- Magic xpa
- *et cetera*

Operating system

It should be possible to do this on **any operating system** that supports Stata (i.e. Windows, Unix/Linux, Mac OS X).

General idea



Web interface
(HTML / JS)

Web application
(PHP / Java /
ASP.net + C# / etc…)

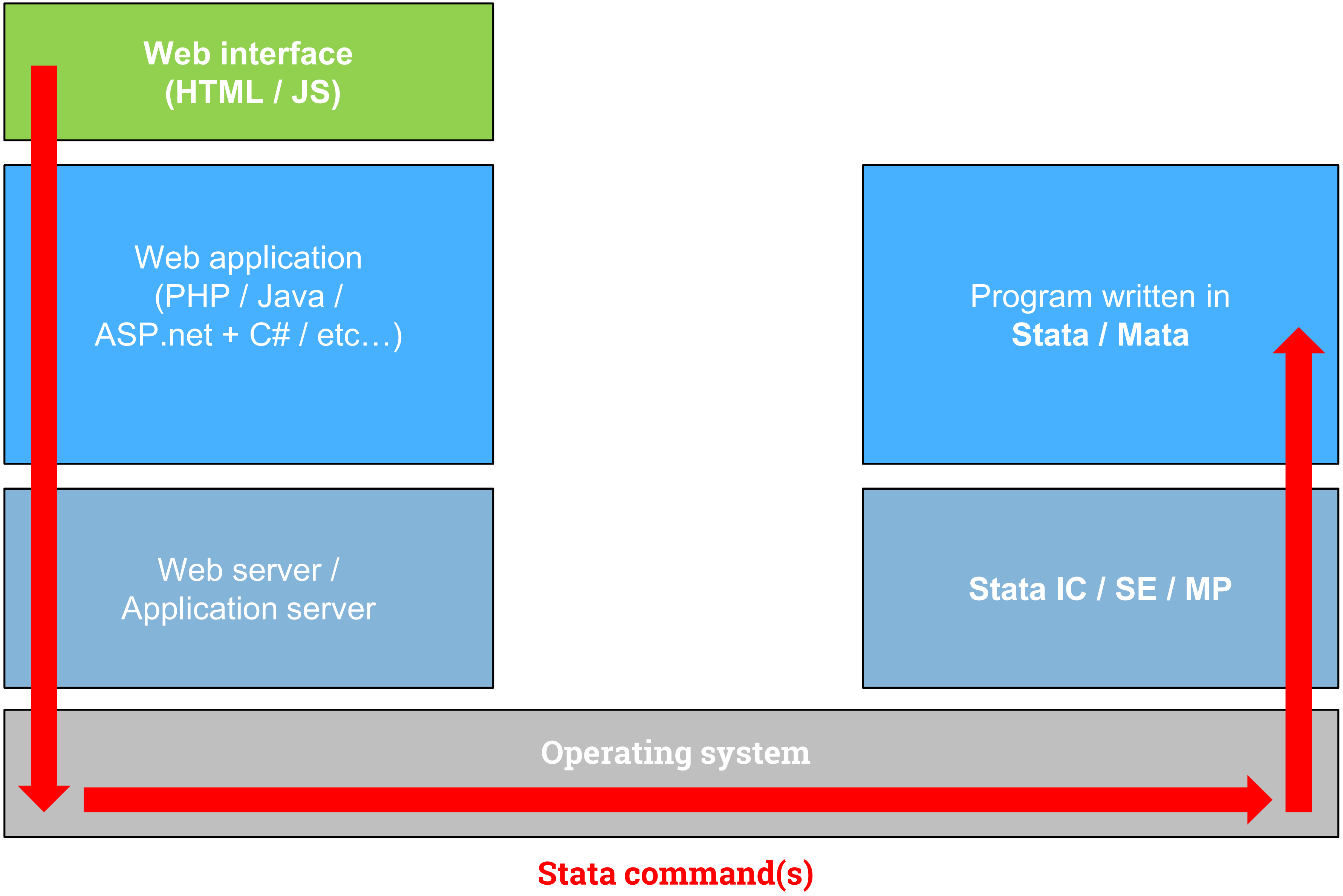
Program written in
Stata / Mata

Web server /
Application server

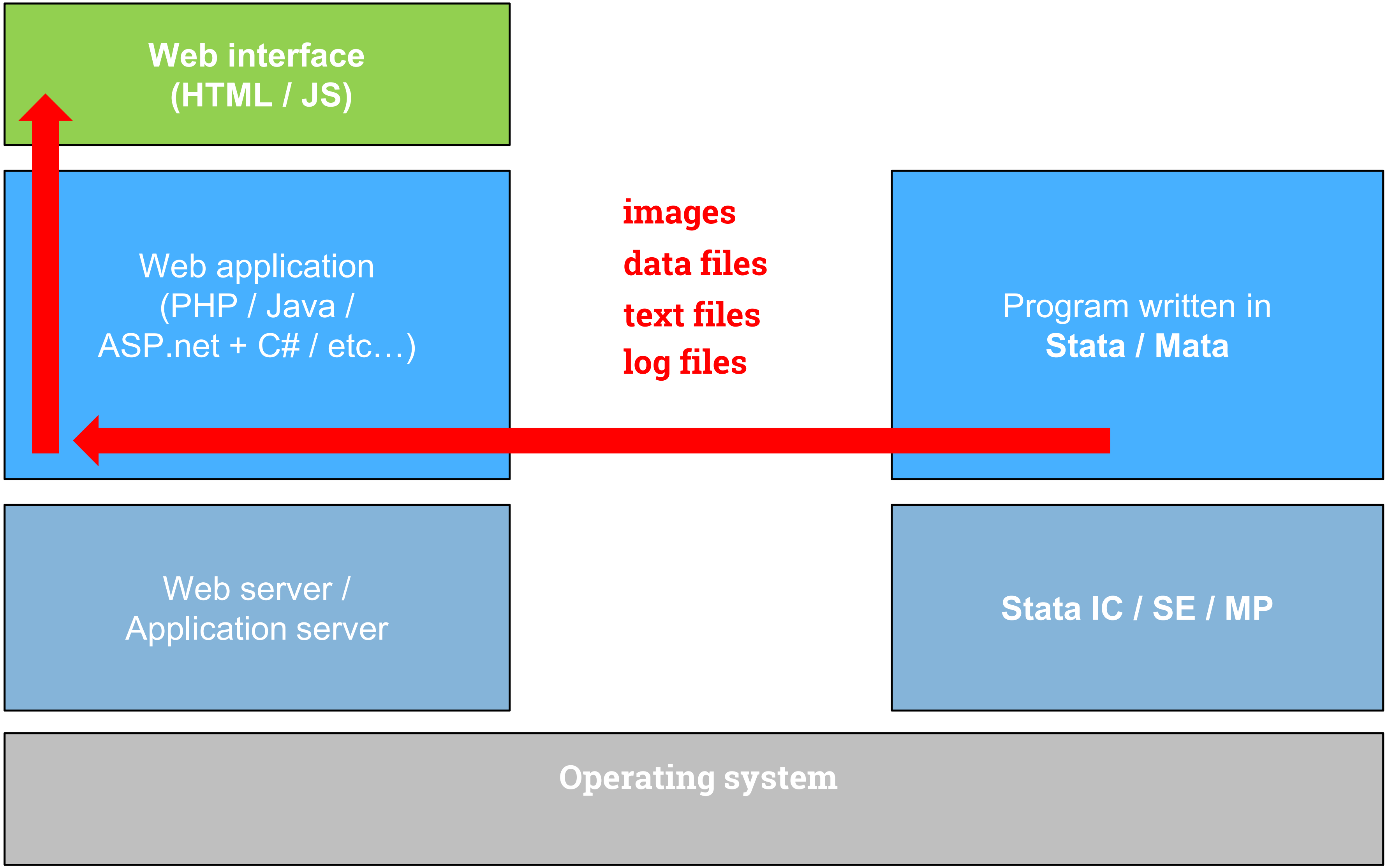
Stata IC / SE / MP

Operating system

Calling Stata



Getting a response from Stata



Web interface (HTML / JS)

Web application (PHP / Java / ASP.net + C# / etc…)

images

data files

text files

log files

Program written in **Stata / Mata**

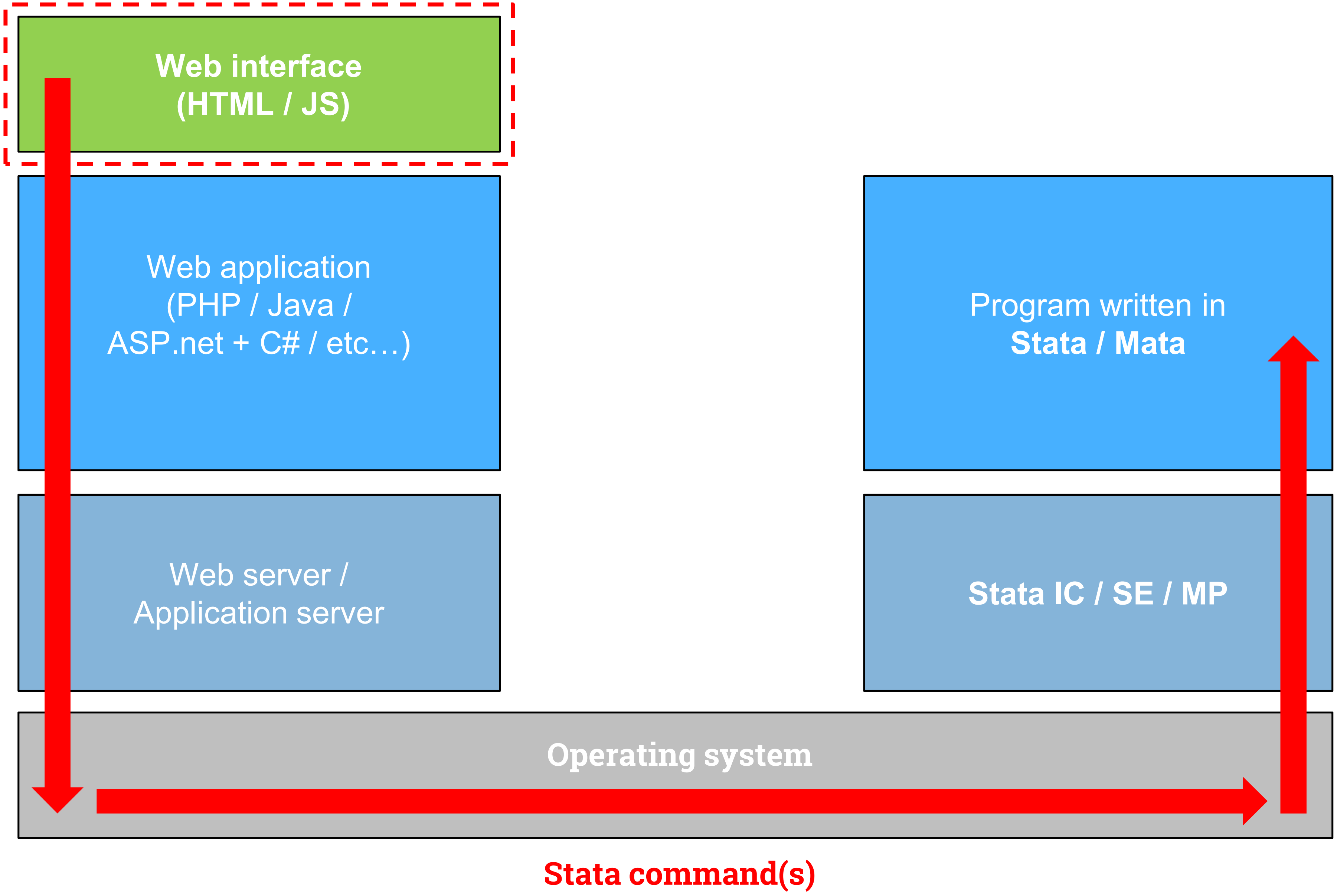
Web server / Application server

Stata IC / SE / MP

Operating system

Simplified example

Calling Stata



Web interface
(HTML / JS)

Web application
(PHP / Java /
ASP.net + C# / etc…)

Web server /
Application server

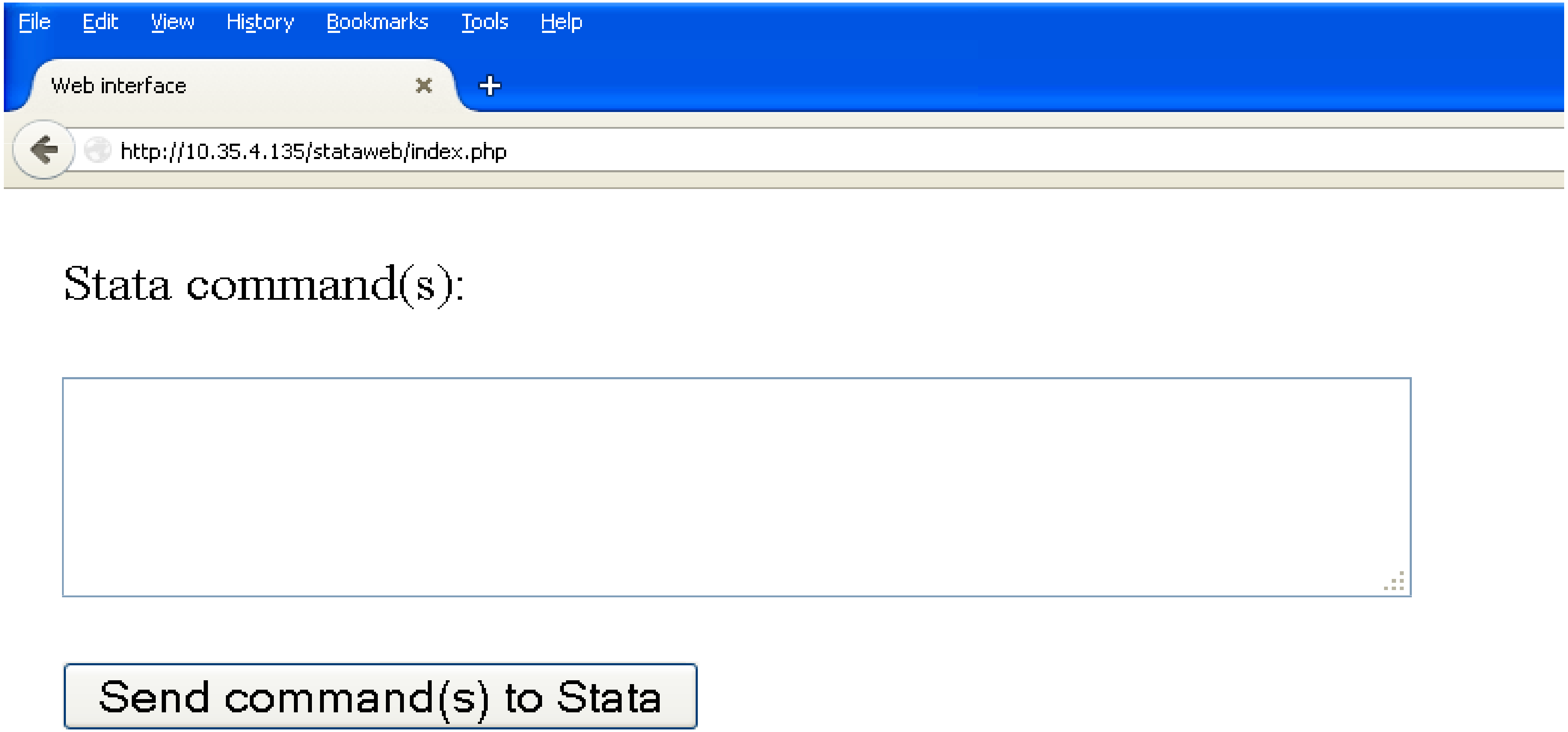
Program written in
Stata / Mata

Stata IC / SE / MP

Operating system

Stata command(s)

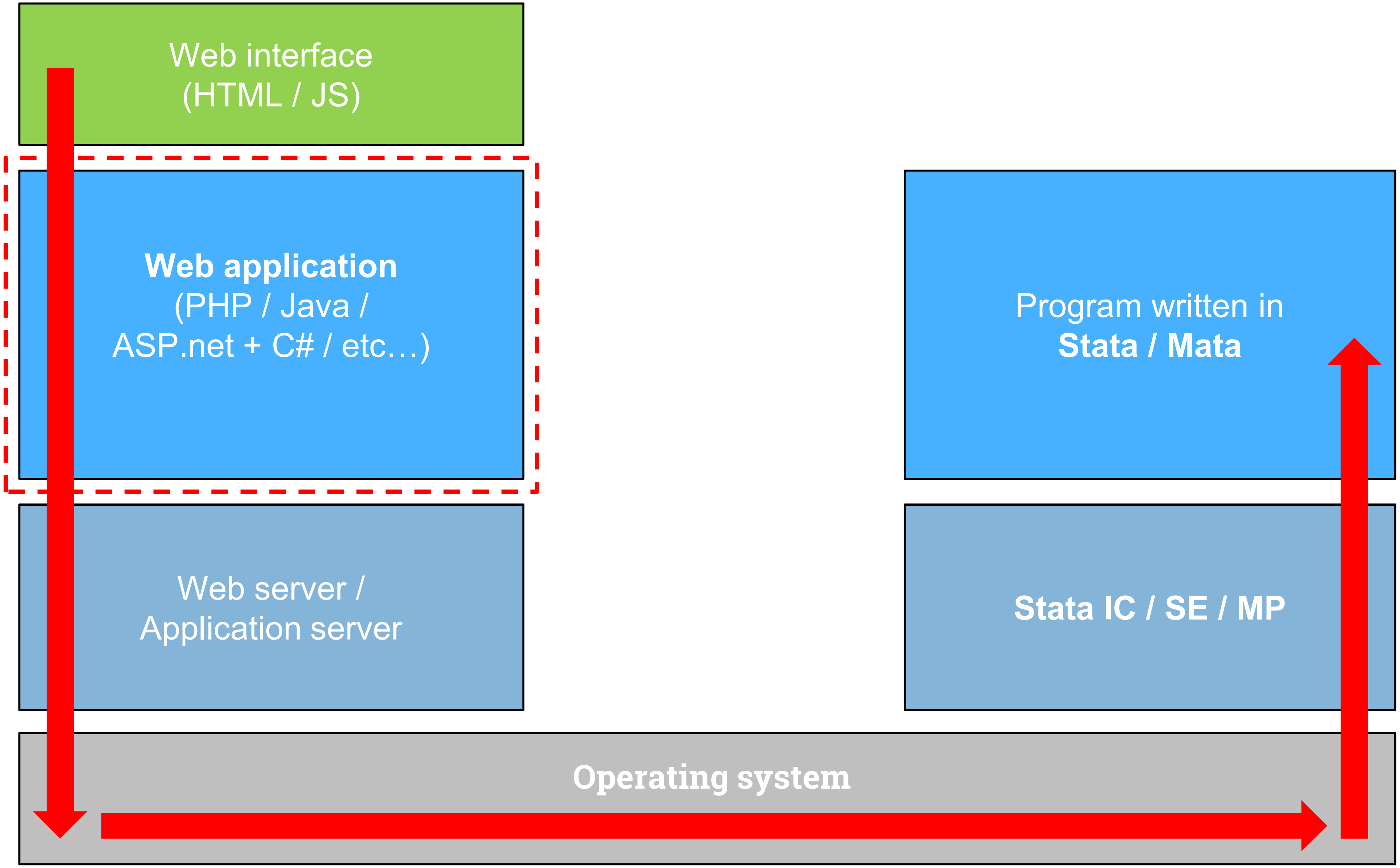
Calling Stata



Calling Stata

```
<html>
 <head> Web interface </head>
 <body>
 <form action=“call_stata.php" method="post">
   Stata command(s):<br><br>
   <textarea name="stata_commands" ><br><br>
   <input type="submit"
    value="Send command(s) to Stata" >
 </form>
 </body>
</html>
```

Calling Stata



Web interface
(HTML / JS)

Web application
(PHP / Java /
ASP.net + C# / etc…)

Web server /
Application server

Program written in
Stata / Mata

Stata IC / SE / MP

Operating system

Stata command(s)

Calling Stata

call_stata.php

```
<?php
...

$stata_commands = $_POST[“stata_commands”];

write_stata_do_file($stata_commands);

execute_stata_do_file();

...
>
```

Calling Stata

Our web application will execute:

```
<path_to_Stata>/Stata.exe /q /e  do "commands.do"
```

$stata_commands

We'll previously write our commands here

Calling Stata

Our web application will execute:

```
<path_to_Stata>/Stata.exe /q /e  do “commands.do”
```

$stata_commands

We’ll previously write our commands here

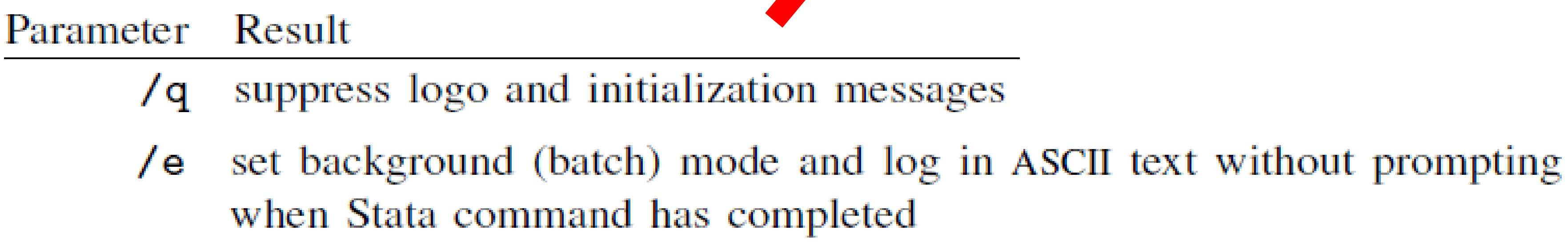
Example:

```
 cd <path_to_temp_folder>
 sysuse auto
 histogram price
```

Calling Stata

Our web application will execute:

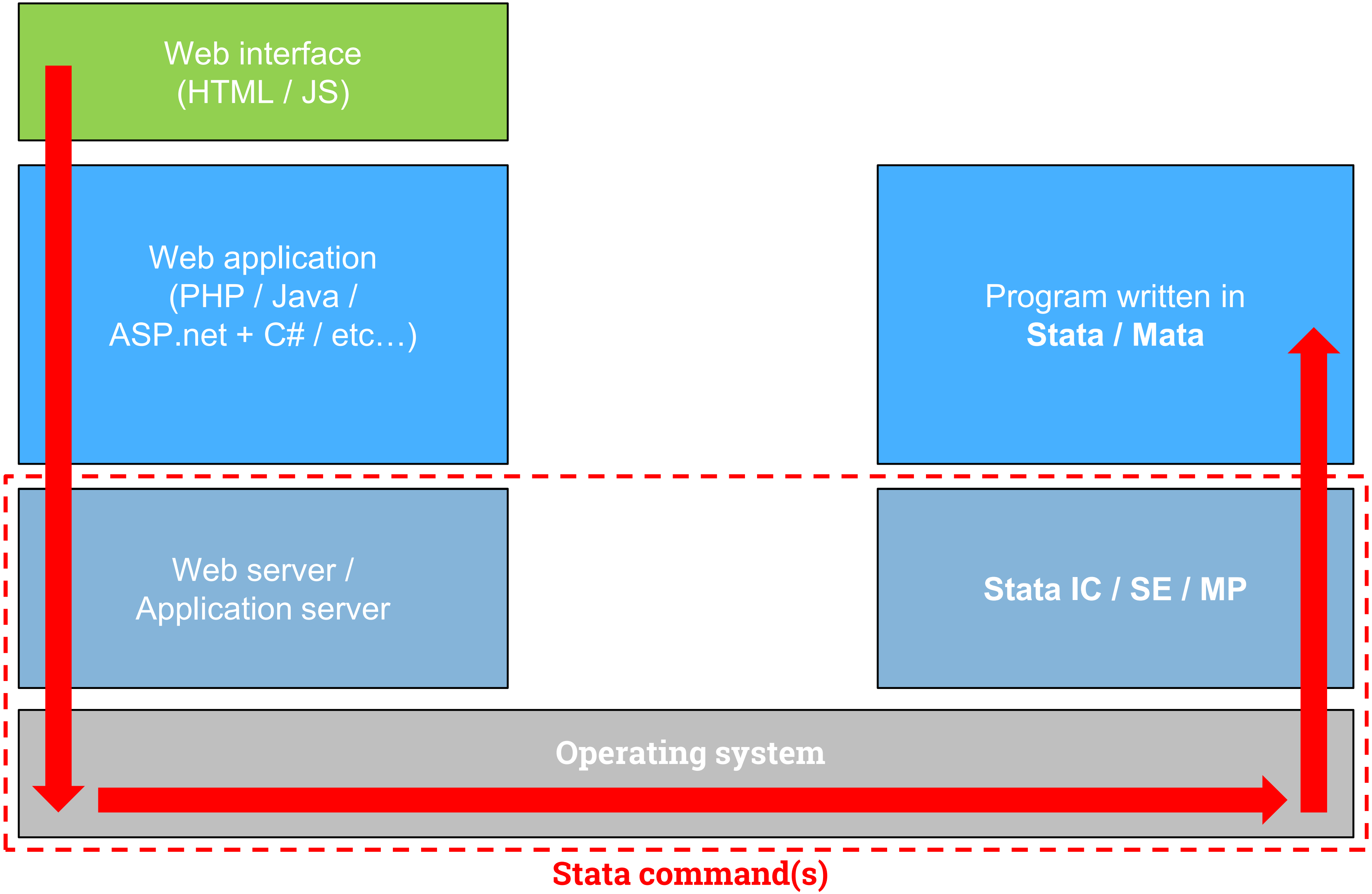
```
<path_to_Stata>/Stata.exe /q /e do “commands.do”
```



Parameter	Result
/q	suppress logo and initialization messages
/e	set background (batch) mode and log in ASCII text without prompting when Stata command has completed

*(Stata User’s Guide, section [**B.5**])*

Calling Stata



Calling Stata

Ex: PHP

-- synchronous execution

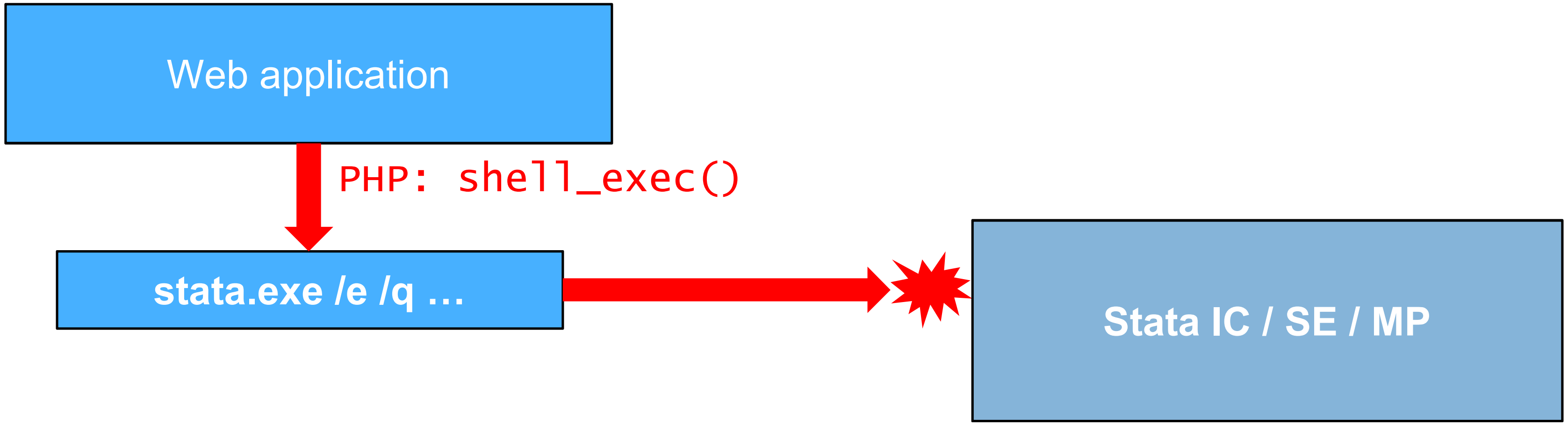
```
shell_exec(...);
```

-- asynchronous execution

```
pclose(popen(...,"r"));
```

Calling Stata

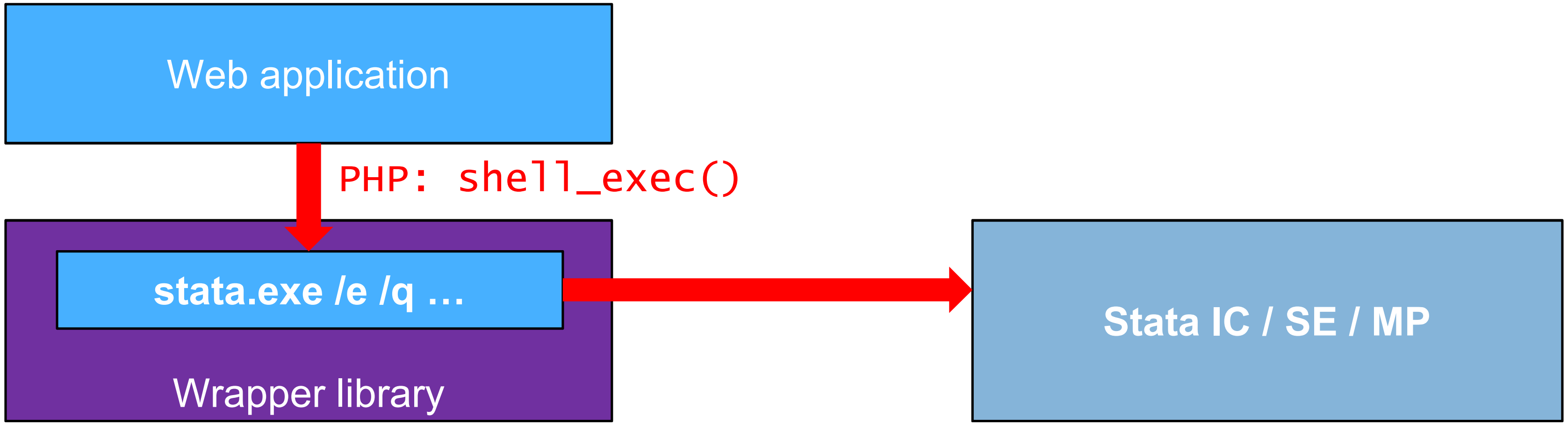
Problem: modern versions of Stata will **not** work if called directly from a web server (SYSTEM user).



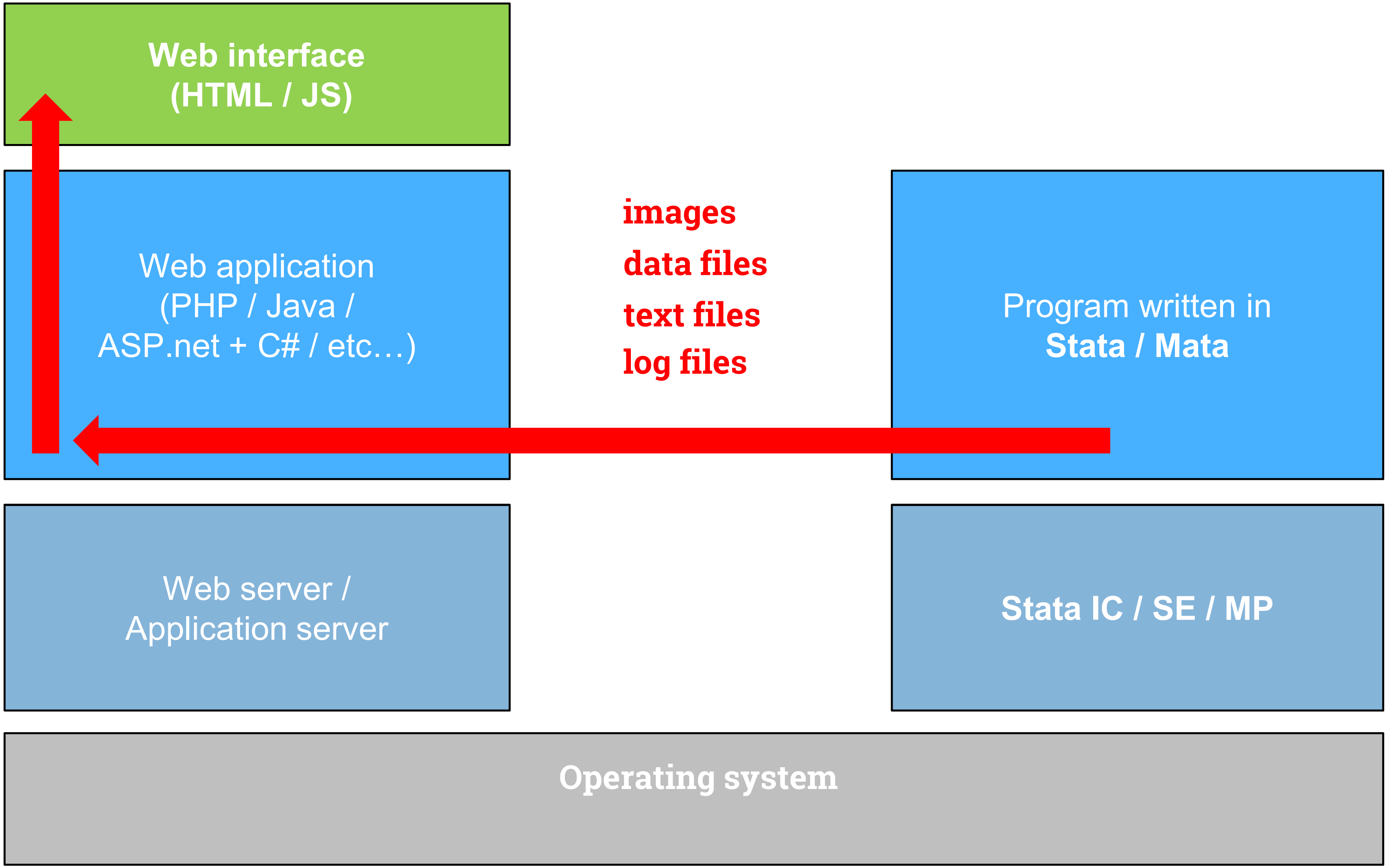
Calling Stata

Problem: modern versions of Stata will **not** work if called directly from a web server (SYSTEM user).

Solution: wrapper + user impersonation



Getting a response from Stata



Web interface (HTML / JS)

Web application (PHP / Java / ASP.net + C# / etc…)

images

data files

text files

log files

Program written in **Stata / Mata**

Web server / Application server

Stata IC / SE / MP

Operating system

Getting a response from Stata

Our web application will execute:

```
<path_to_Stata>/Stata.exe /q /e  do “commands.do”
```

We’ll previously write our commands here

Example:

```
cd <path_to_temp_folder>
sysuse auto
histogram price, normal saving(graph01, replace)
graph export graph01.png, replace
```

Getting a response from Stata

Our web application will execute:

```
<path_to_Stata>/Stata.exe /q /e  do “commands.do”
```

We’ll previously write our commands here

Example:

```
 cd <path_to_web_folder>/img/
 sysuse auto
 histogram price, normal saving(graph01, replace)
 graph export graph01.png, replace
```

Now our web application will be able to display

```
<path_to_web_folder>/img/graph01.png
```

Getting a response from Stata

call_stata.php

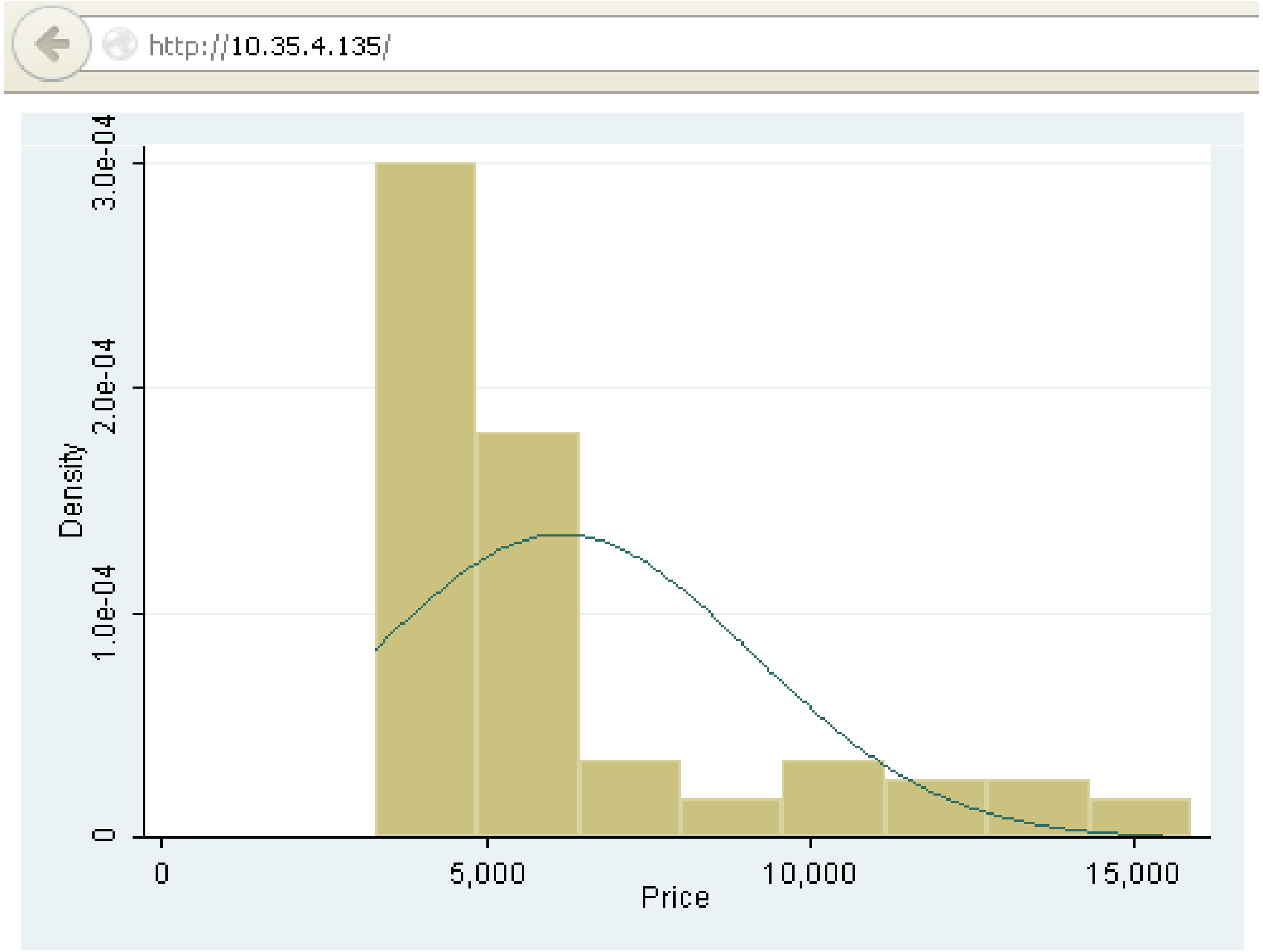
```
<?php
...

$stata_commands = $_POST[“stata_commands”];

write_stata_do_file($stata_commands);

execute_stata_do_file();

display_results(); //display graph01.png

...
?>
```

Getting a response from Stata

call_stata.php

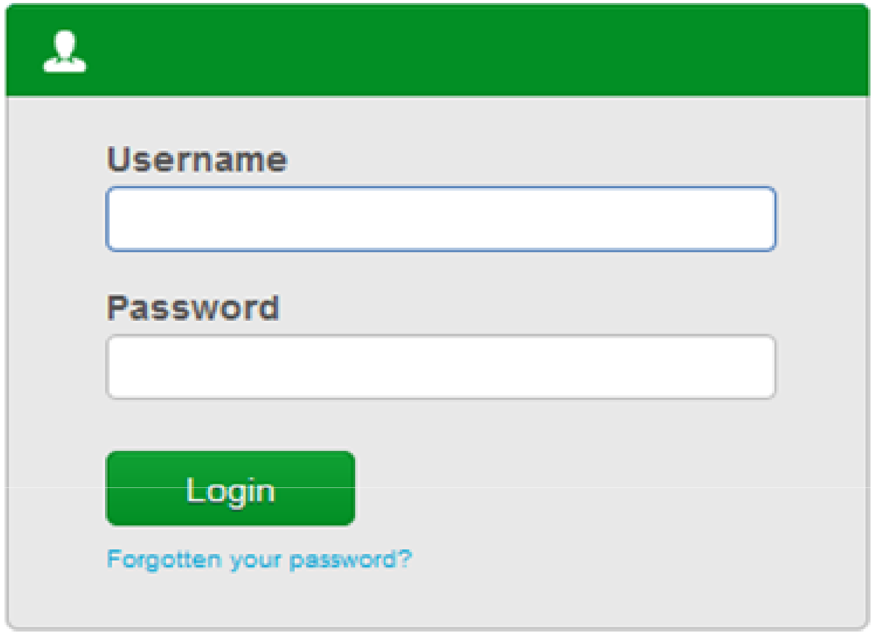
```
<?php
...

function display_results() {
echo “<html>”;
echo “ <head>Result</head>”;
echo “ <body>”;
echo “  <img src=img/graph01.png>”;
echo “ </body>”;
echo “</html>”;
}
...
?>
```

Getting a response from Stata



Basic security

SQL injection attack:



'; DROP TABLE users;

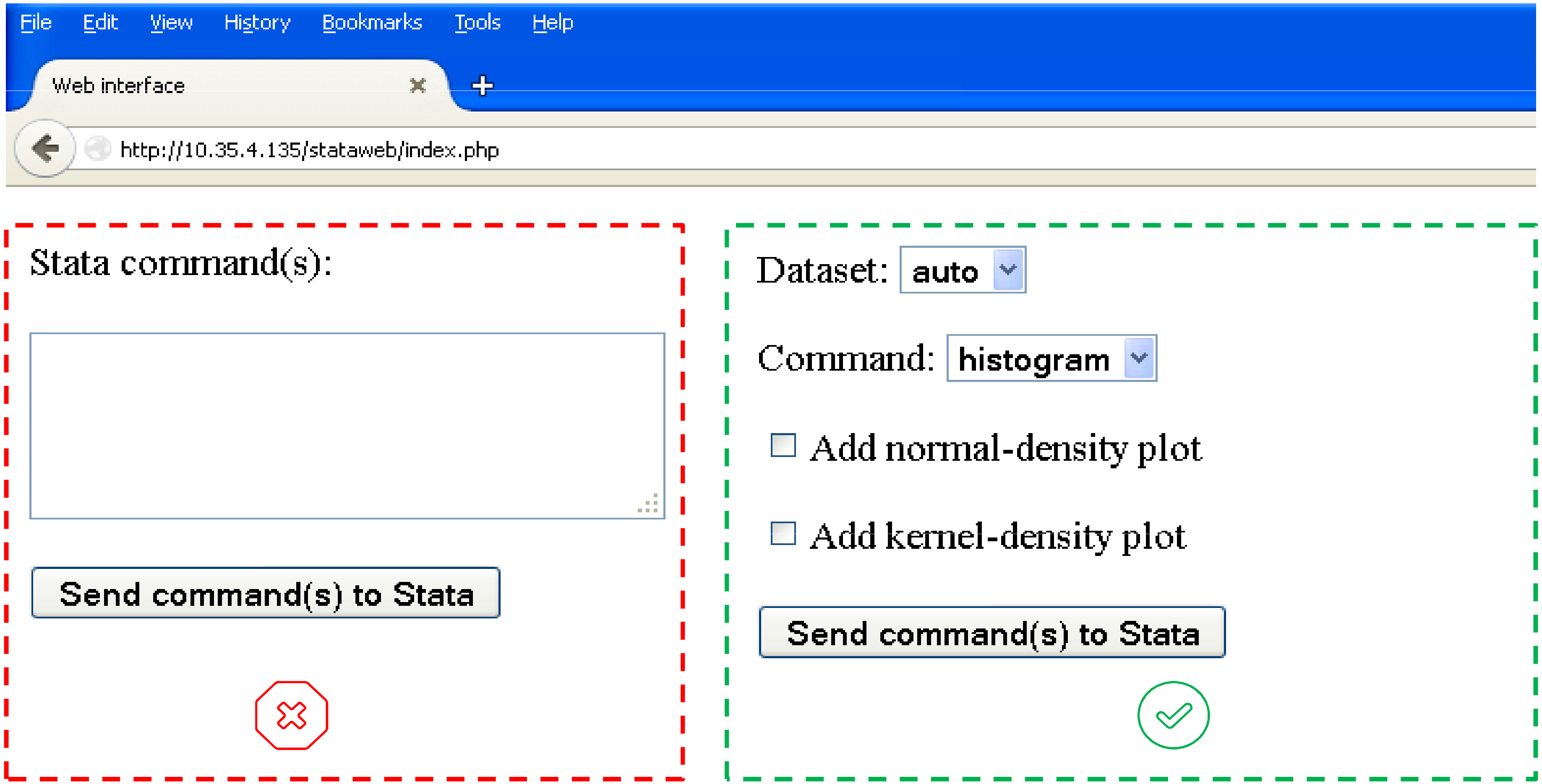
Basic security

Prevent “**Stata injection**” attacks:

--Limited, sanitized inputs,
Ideally, **no free text** fields on the web interface

--Avoid or restrict **shell(), xshell(), winexec()**
in your Stata program

Basic security



http://10.35.4.135/stataweb/index.php

Stata command(s):

Dataset: auto

Command: histogram

□ Add normal-density plot

□ Add kernel-density plot

Bad practice

Better practice

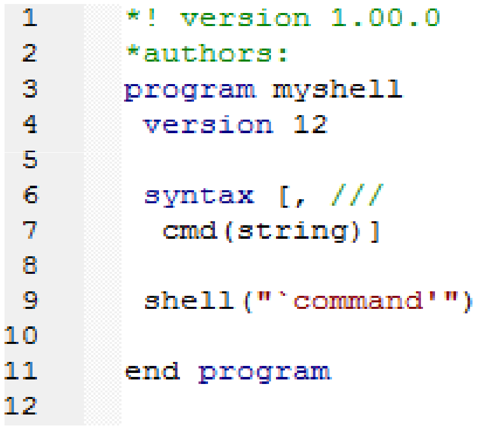
Basic security

Prevent “**Stata injection**” attacks:

--Limited, sanitized inputs,
Ideally, **no free text** fields on the web interface

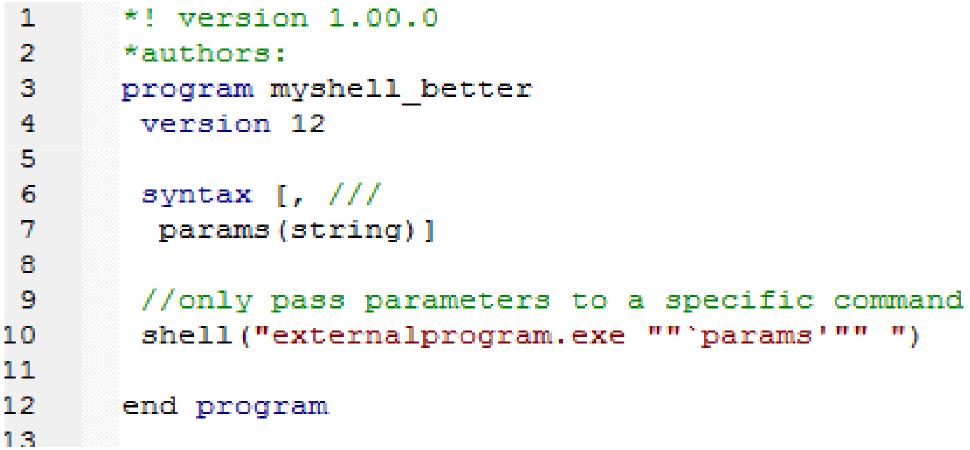
--Avoid or restrict **shell(), xshell(), winexec()**
in your Stata program

Basic security



```
*! version 1.00.0
*authors:
program myshell
 version 12

 syntax [, ///
  cmd(string)]

 shell("`command'")

end program
```

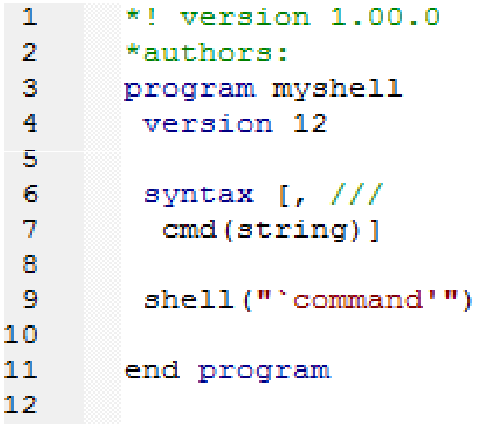
Bad practice



```
*! version 1.00.0
*authors:
program myshell_better
 version 12

 syntax [, ///
  params(string)]

 //only pass parameters to a specific command
 shell("externalprogram.exe ""`params'"" ")

end program
```

Better practice

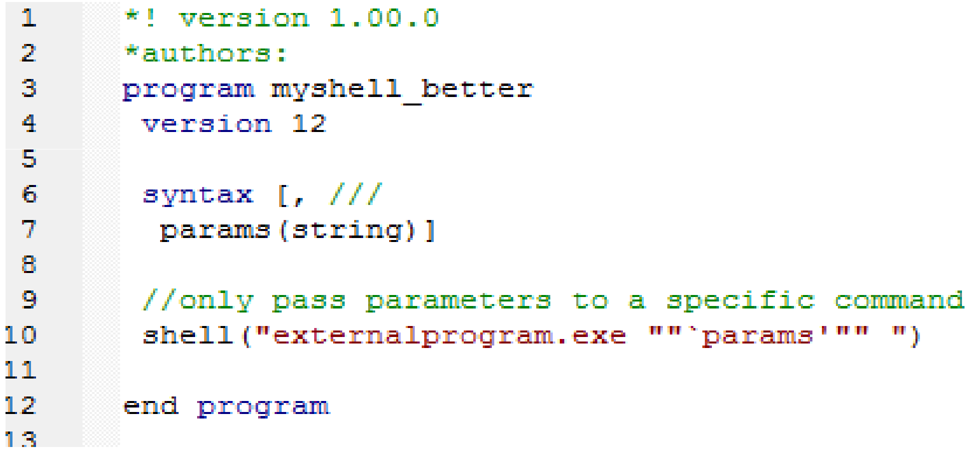
Basic security



```
*! version 1.00.0
*authors:
program myshell
 version 12

 syntax [, ///
  cmd(string)]

 shell("`command'")

end program
```

✗

Bad practice



```
*! version 1.00.0
*authors:
program myshell_better
 version 12

 syntax [, ///
  params(string)]

 //only pass parameters to a specific command
 shell("externalprogram.exe ""`params'"" ")

end program
```

It's even better to avoid dynamic shell() commands if Stata is executed through a web interface

✓

Better practice

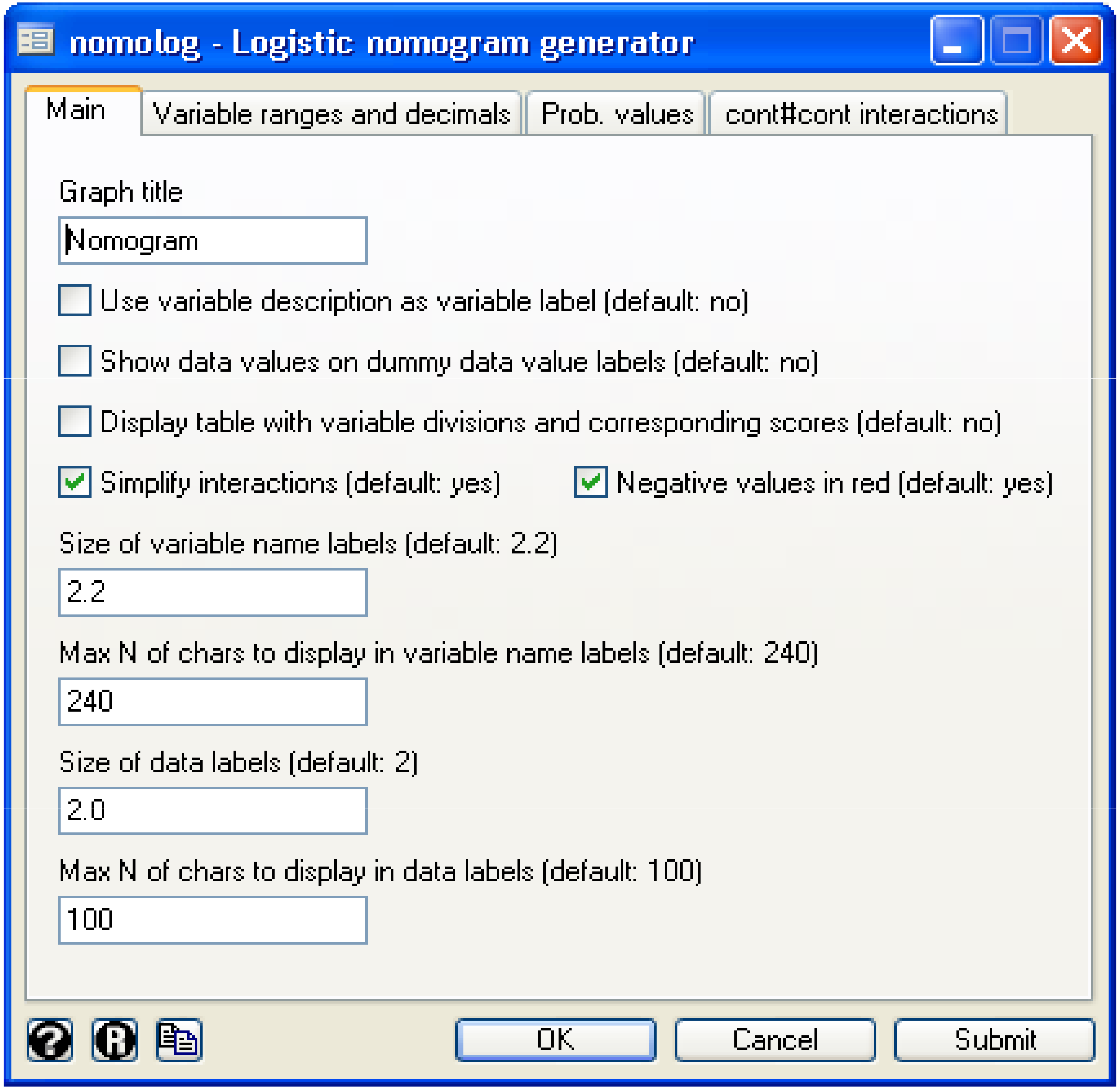
Implementation example

Web interface for –nomolog–

A general-purpose nomogram generator for predictive logistic regression models

Zlotnik A, Abraira V. Stata Journal. 2015. Volume 15, Number 2

URL: http://www.zlotnik.net/stata/nomograms



nomolog - Logistic nomogram generator
Main
Variable ranges and decimals
Prob. values
cont#cont interactions
OK
Cancel
Submit

Graph title

Nomogram

□ Use variable description as variable label (default: no)

□ Show data values on dummy data value labels (default: no)

□ Display table with variable divisions and corresponding scores (default: no)

☑ Simplify interactions (default: yes) ☑ Negative values in red (default: yes)

Size of variable name labels (default: 2.2)

2.2

Max N of chars to display in variable name labels (default: 240)

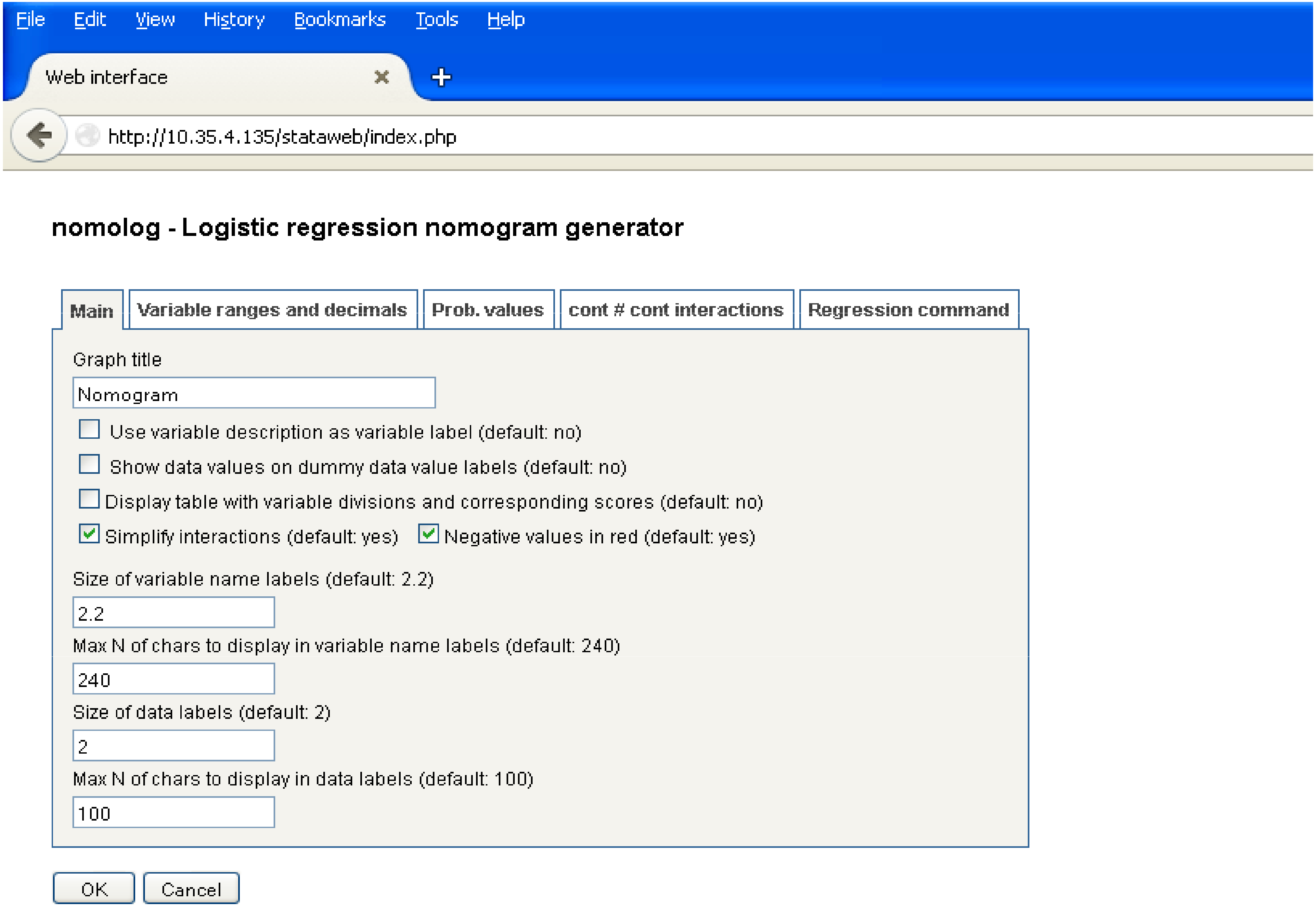
240

Size of data labels (default: 2)

2.0

Max N of chars to display in data labels (default: 100)

100


File Edit View History Bookmarks Tools Help
Web interface
http://10.35.4.135/stataweb/index.php

nomolog - Logistic regression nomogram generator

Main
Variable ranges and decimals
Prob. values
cont # cont interactions
Regression command

Graph title

Nomogram

□ Use variable description as variable label (default: no)

□ Show data values on dummy data value labels (default: no)

□ Display table with variable divisions and corresponding scores (default: no)

☑ Simplify interactions (default: yes) ☑ Negative values in red (default: yes)

Size of variable name labels (default: 2.2)

2.2

Max N of chars to display in variable name labels (default: 240)

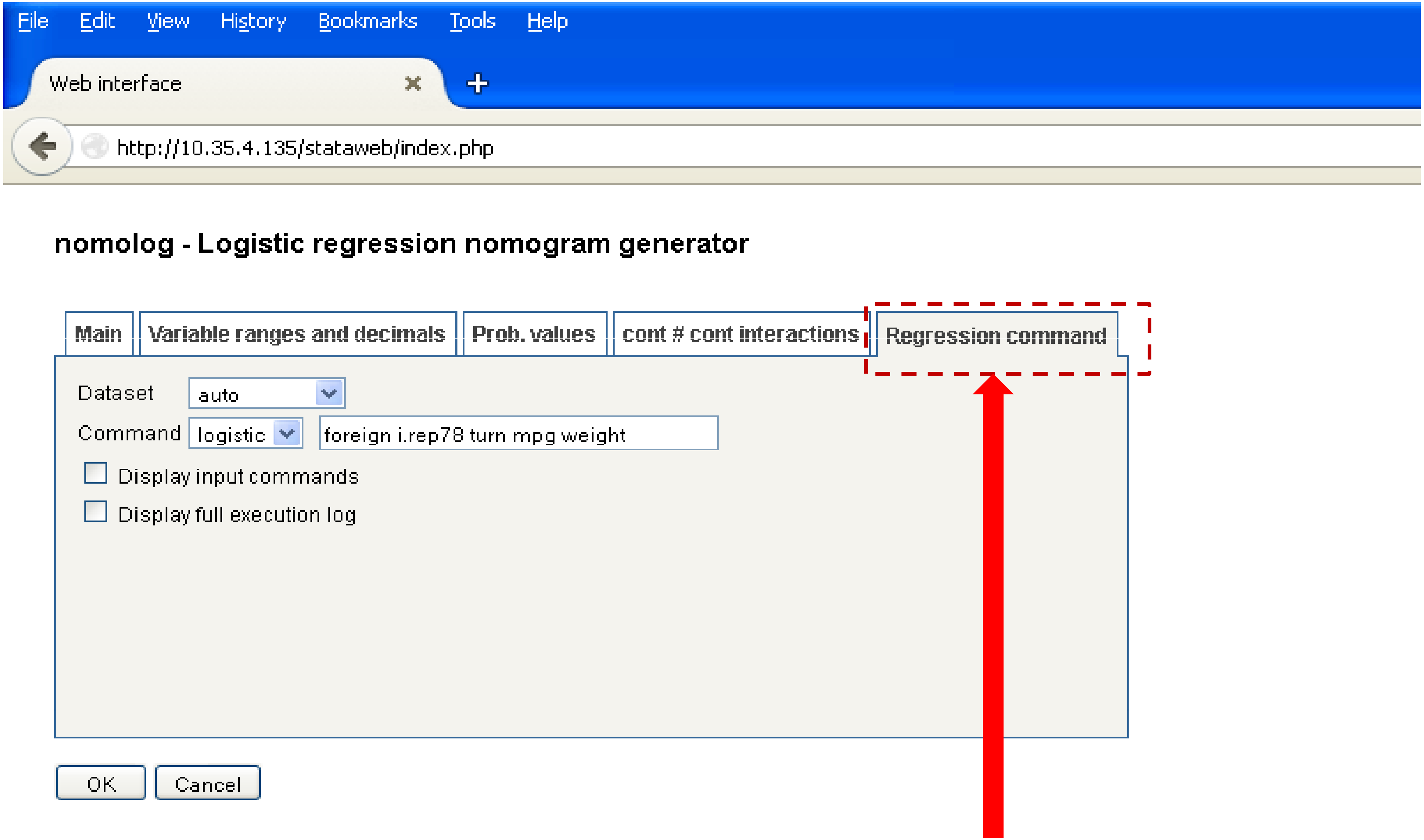
240

Size of data labels (default: 2)

2

Max N of chars to display in data labels (default: 100)

100



nomolog - Logistic regression nomogram generator

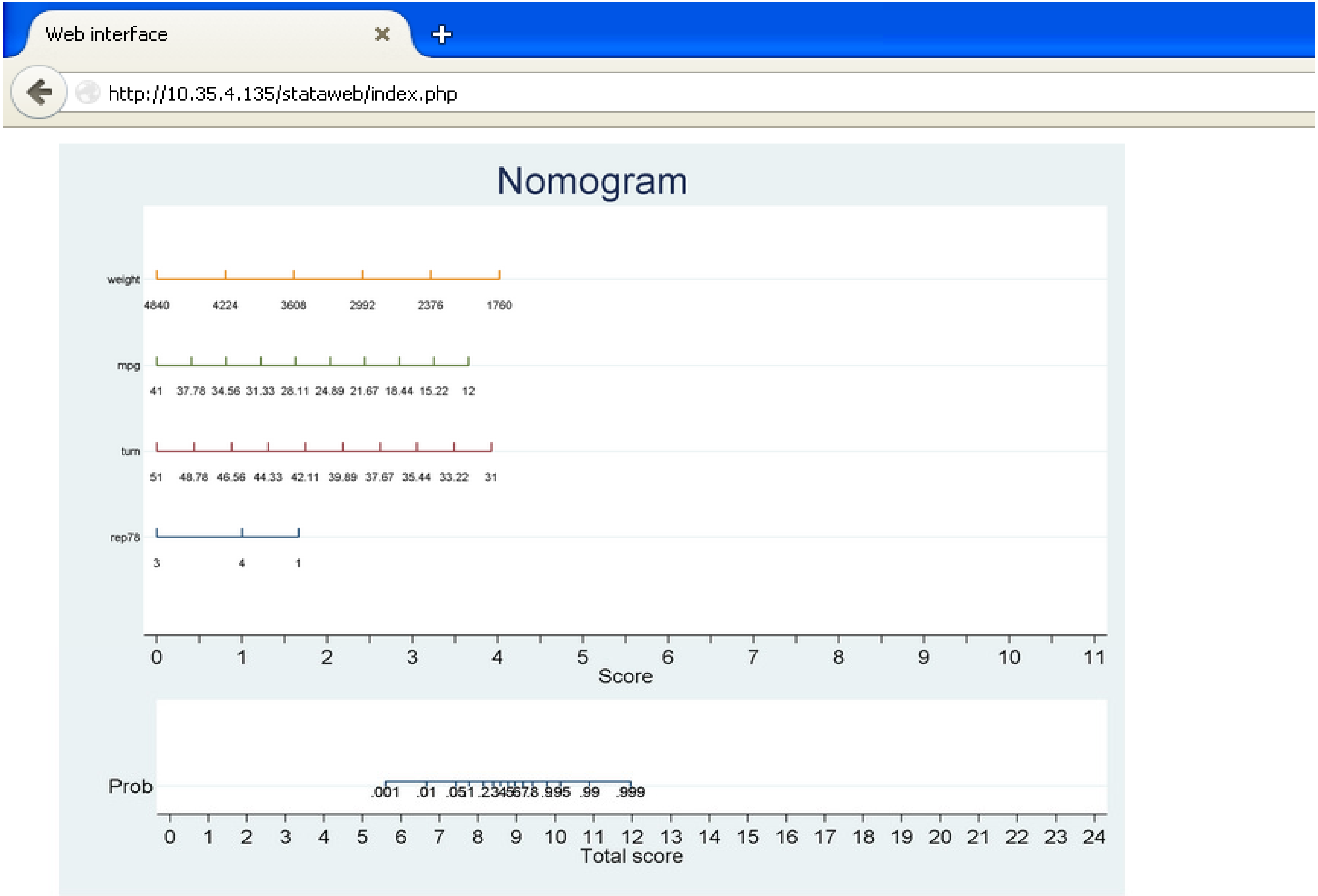
Dataset: auto

Command: logistic foreign i.rep78 turn mpg weight

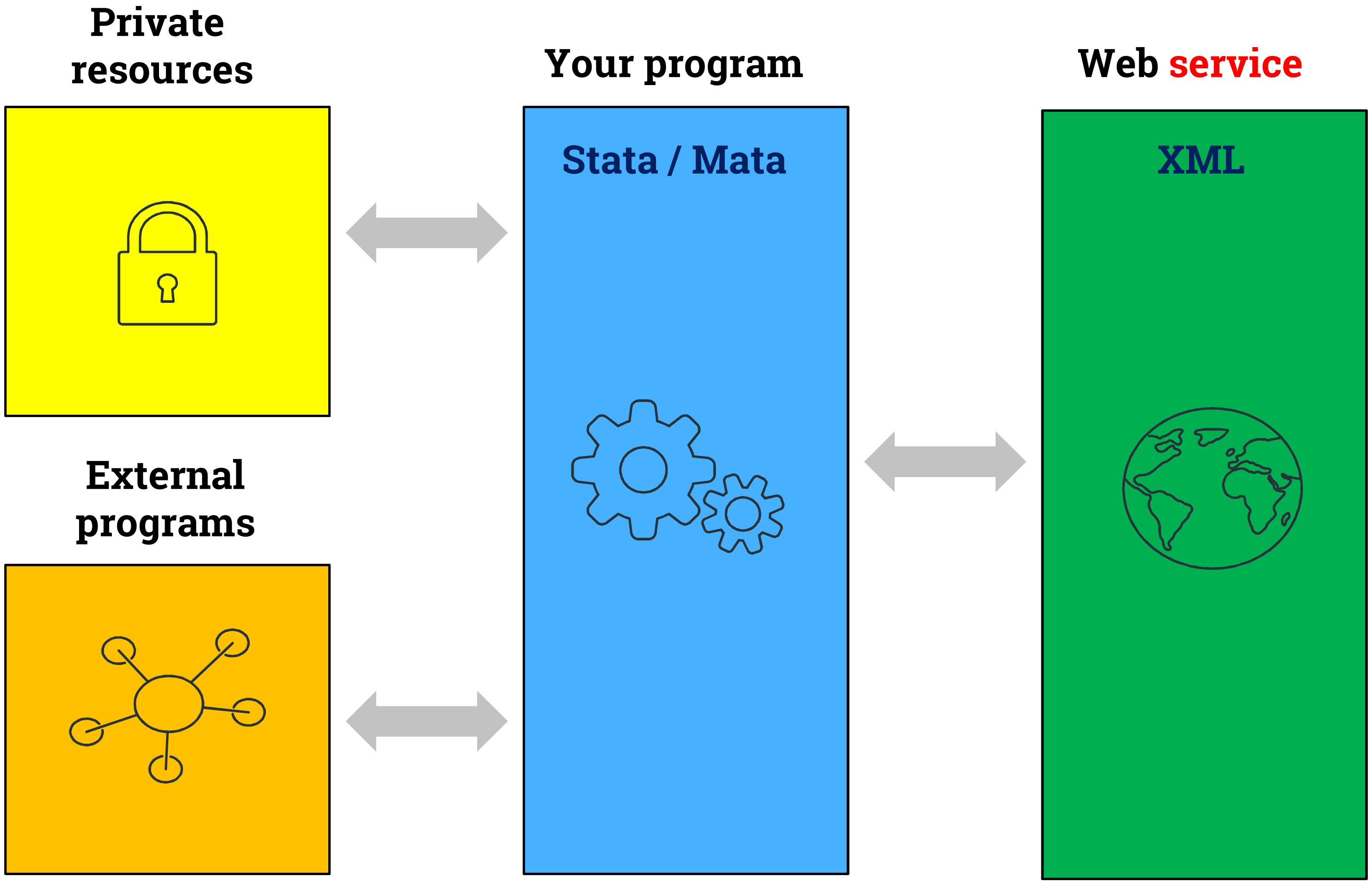
□ Display input commands

□ Display full execution log

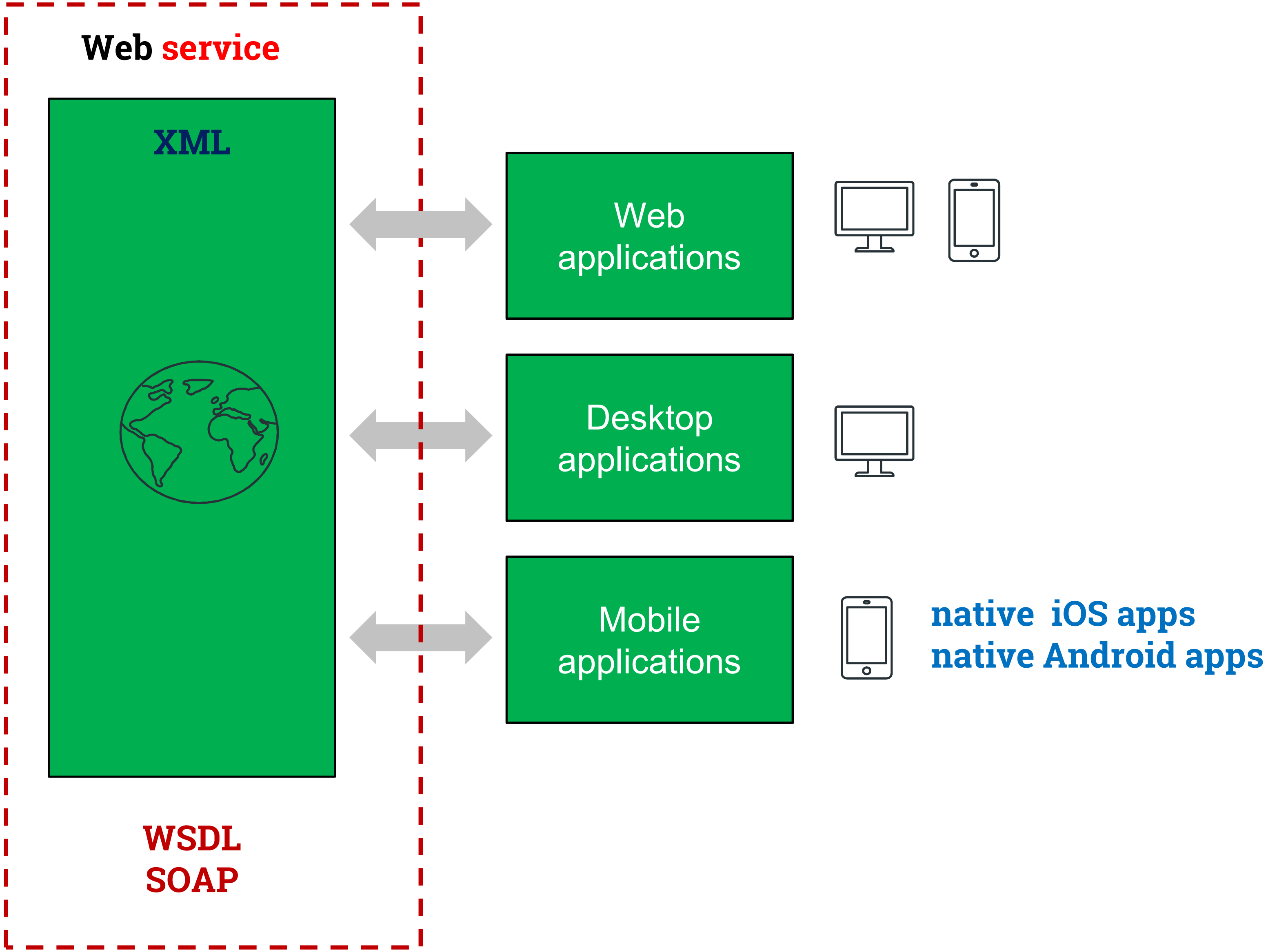
In the web implementation, we must add a tab for loading the dataset and executing the logistic regression command.


Web interface
http://10.35.4.135/stataweb/index.php
Nomogram
weight
4840 4224 3608 2992 2376 1760
mpg
41 37.78 34.56 31.33 28.11 24.89 21.67 18.44 15.22 12
turn
51 48.78 46.56 44.33 42.11 39.89 37.67 35.44 33.22 31
rep78
3 4 1
0 1 2 3 4 5 6 7 8 9 10 11
Score
Prob
.001 .01 .051.2345678.995 .99 .999
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
Total score

Stata web services



Stata web services



Web service

XML

WSDL
SOAP

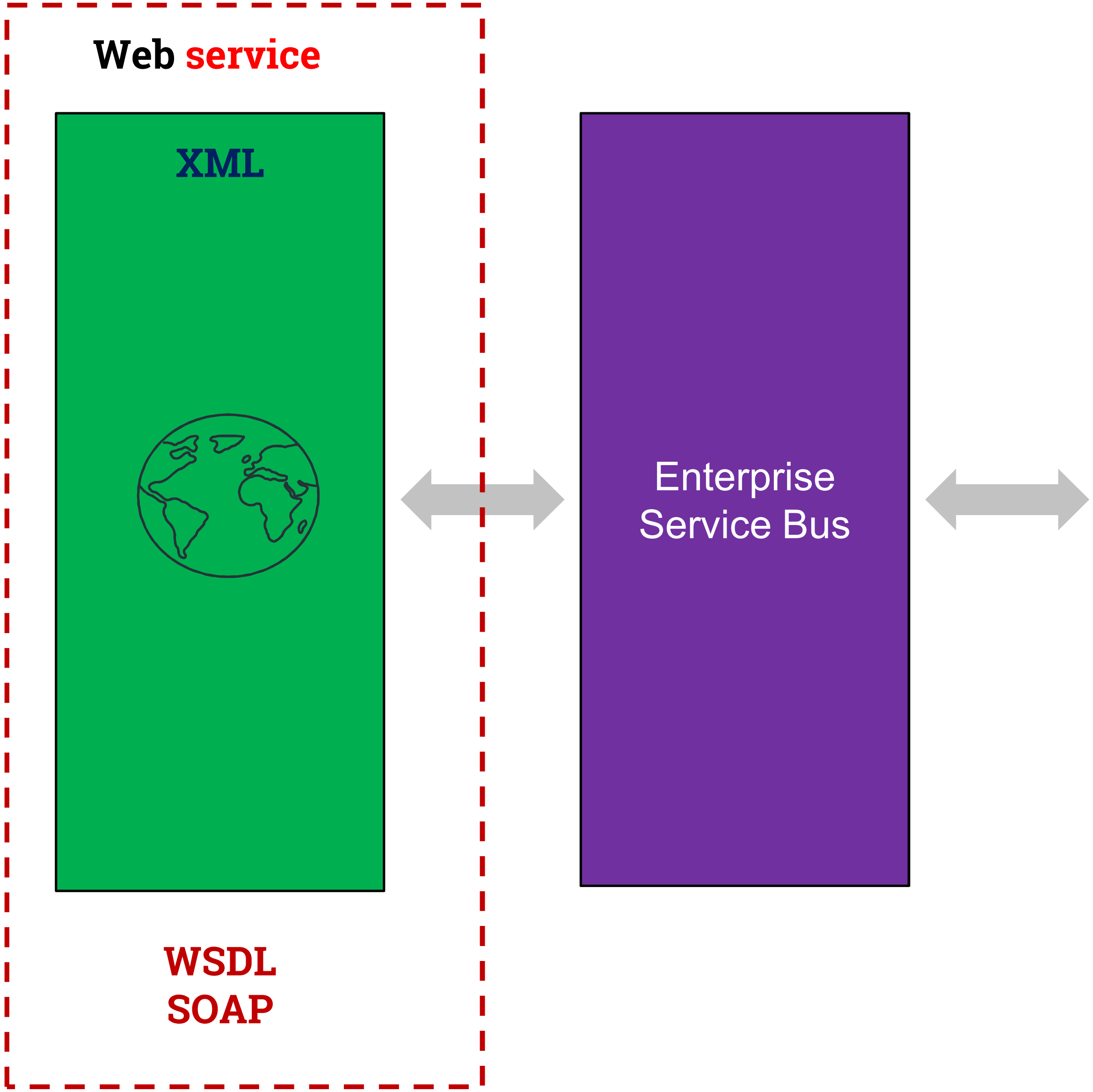
Web applications

Desktop applications

Mobile applications

native iOS apps
native Android apps

Service-oriented architecture



Web service

XML

WSDL
SOAP

Enterprise Service Bus

Other web services

Credits

Credits

Special thanks to all the people who made and released these design resources for free:

- Presentation template by SlidesCarnival
- Photographs by Unsplash & Death to the Stock Photo (license)