



Stata web services: Toward Stata-based healthcare informatics applications integrated in a service-oriented architecture (SOA)

Alexander Zlotnik

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

Modesto Escobar

Universidad de Salamanca

Ascensión Gallardo-Antolín

UC3M, Department of Signals and Systems, Madrid

Juan Manuel Montero Martínez

Technical University of Madrid, ETSIT, DIE

A decorative network diagram in the top-left corner, consisting of various sized circles (nodes) connected by thin lines (edges). Some nodes are solid grey, while others are hollow with a grey outline. The connections form a complex, interconnected web.

Why?

A decorative network diagram in the bottom-right corner, similar to the one in the top-left. It features a cluster of nodes connected by lines, with some nodes being solid grey and others hollow with a grey outline.

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>

A decorative background featuring a network diagram of interconnected nodes and lines, rendered in light gray, positioned in the corners of the slide.

**Sometimes this is
not enough**

Sometimes your program...

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

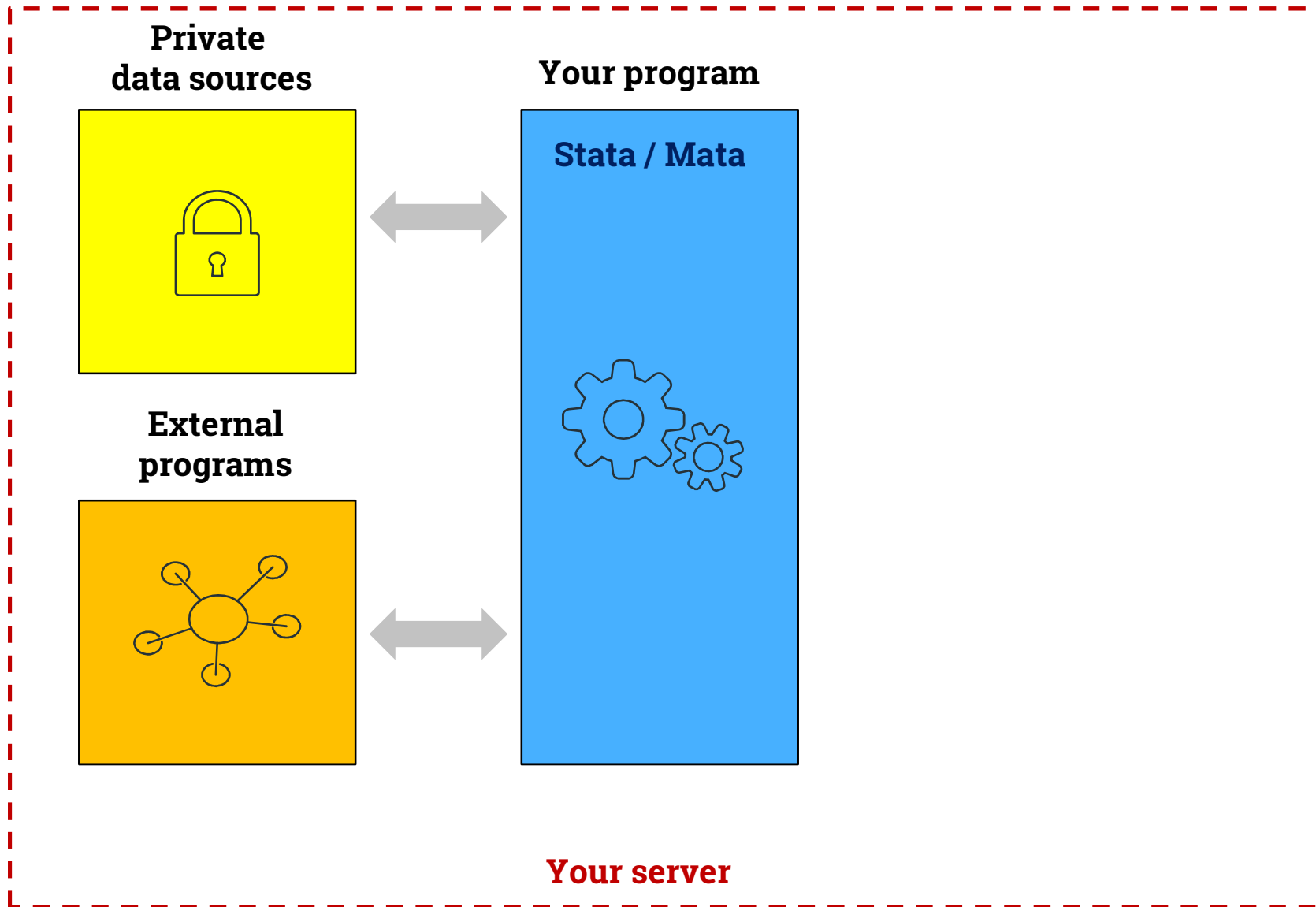
... uses **proprietary data sources**
(ex: real-time currency exchange rates)

... uses **proprietary source code**

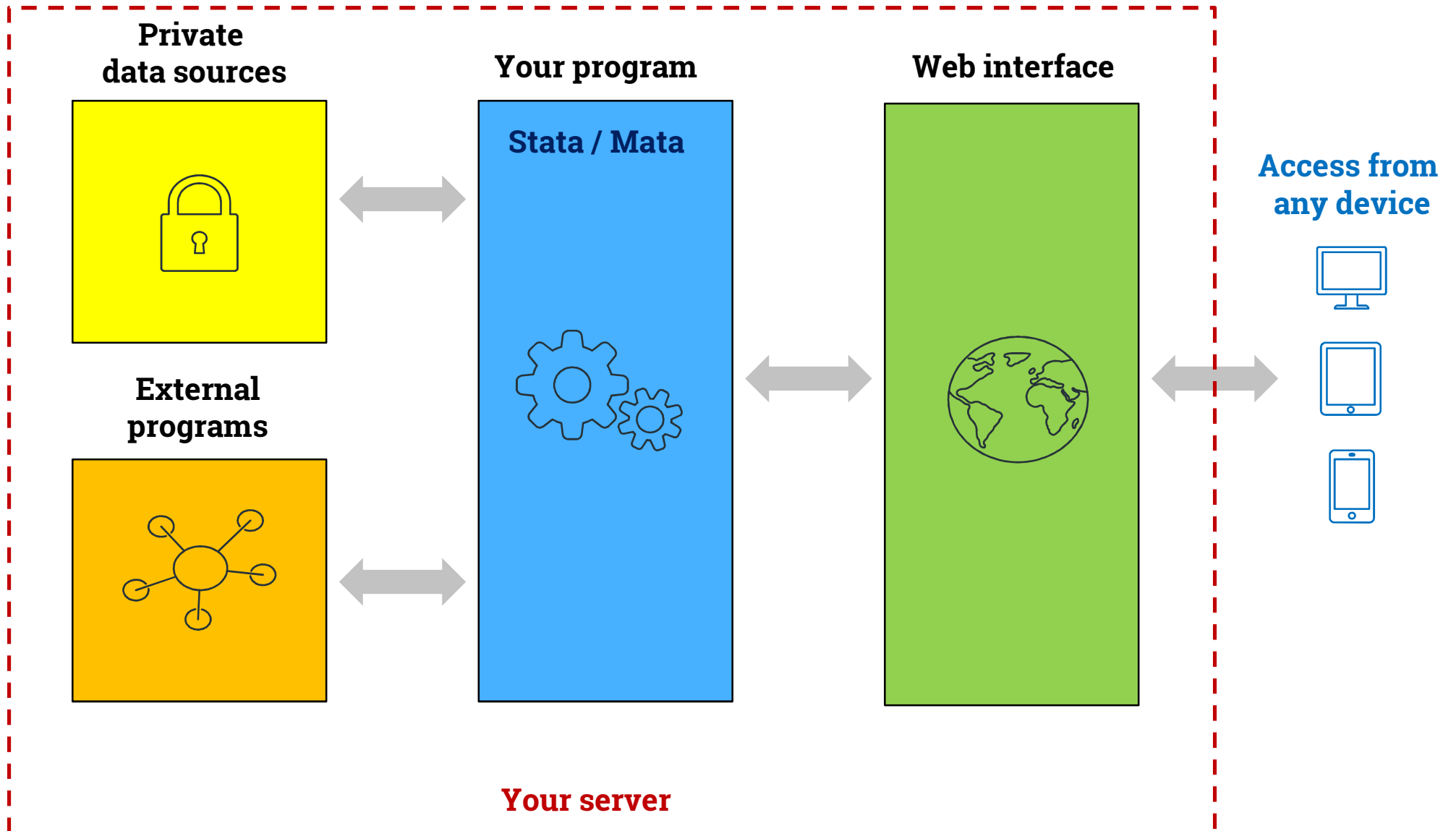
Sometimes your users...

- ... does **not** have the **version of Stata** your program requires (*ex: it may require v14 and they may only have v12*)
- ... does **not** have **Stata at all** (*Stata is not very common in some fields*)
- ... does **not** have a PC, but may have a *smartphone* with a web browser (*ex: developing countries*)

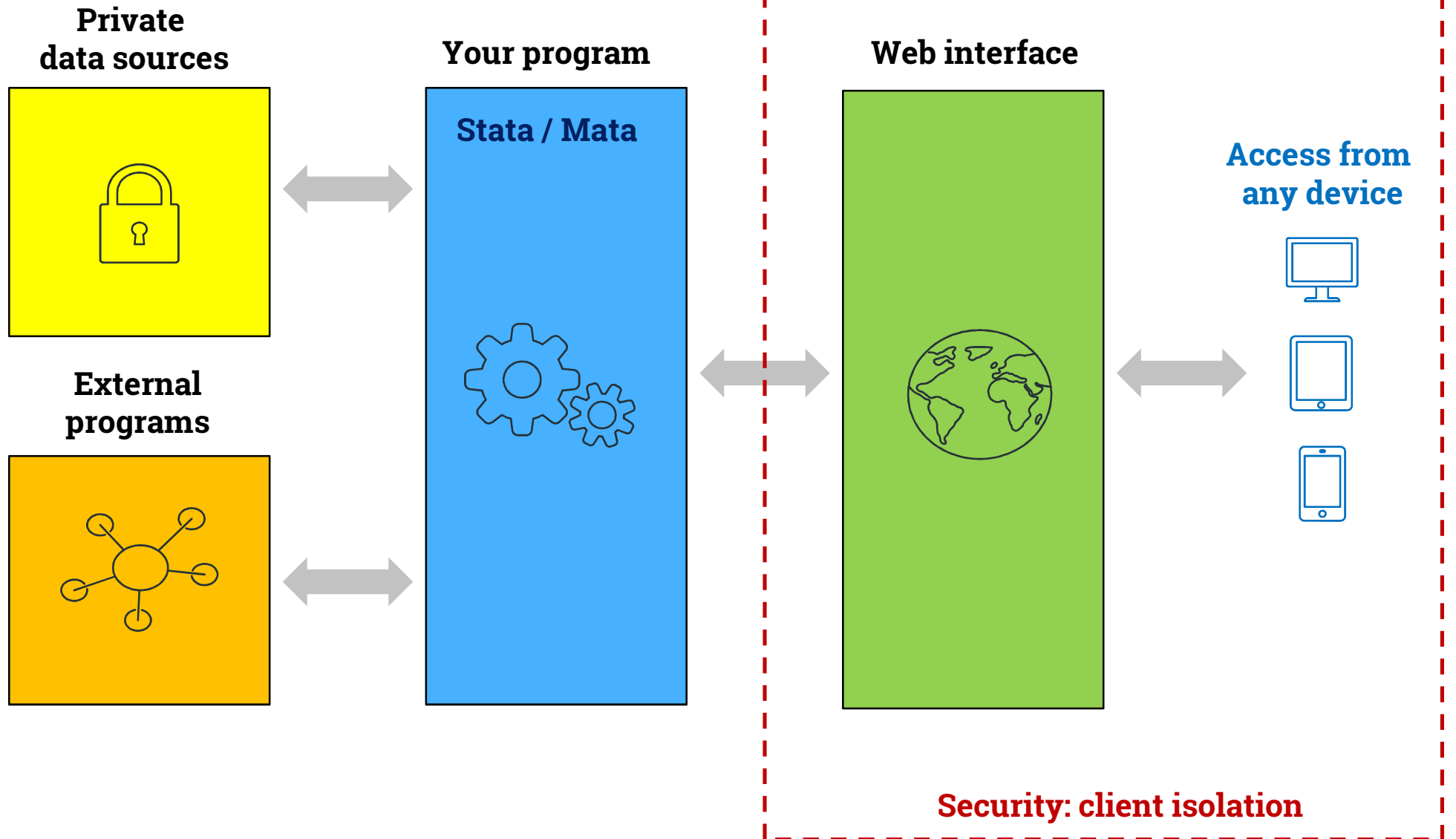
What if...?



What if...?



What if...?



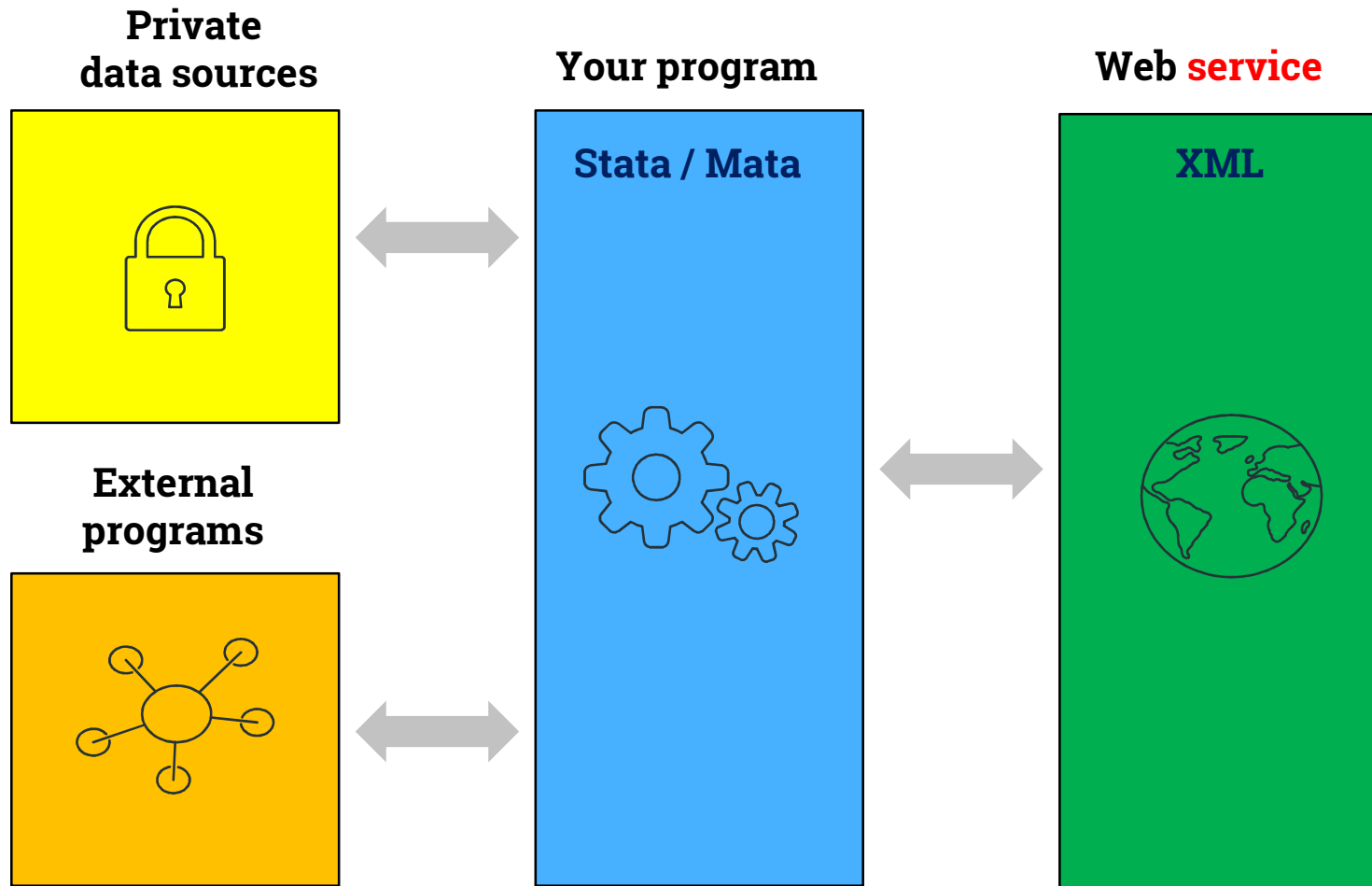
Sometimes your program...

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

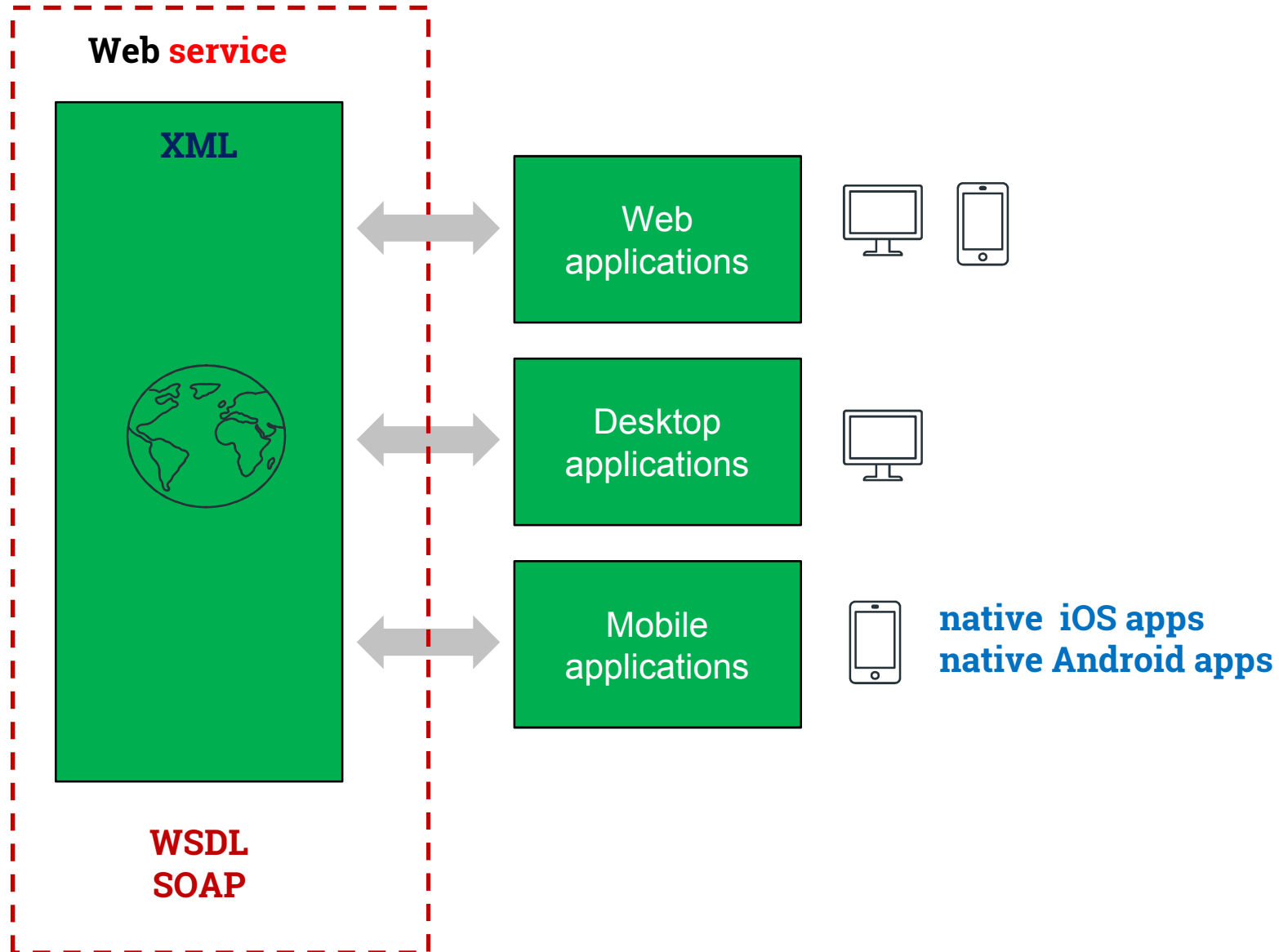
... uses **proprietary data sources**
(ex: real-time currency exchange rates)

... uses **proprietary source code**

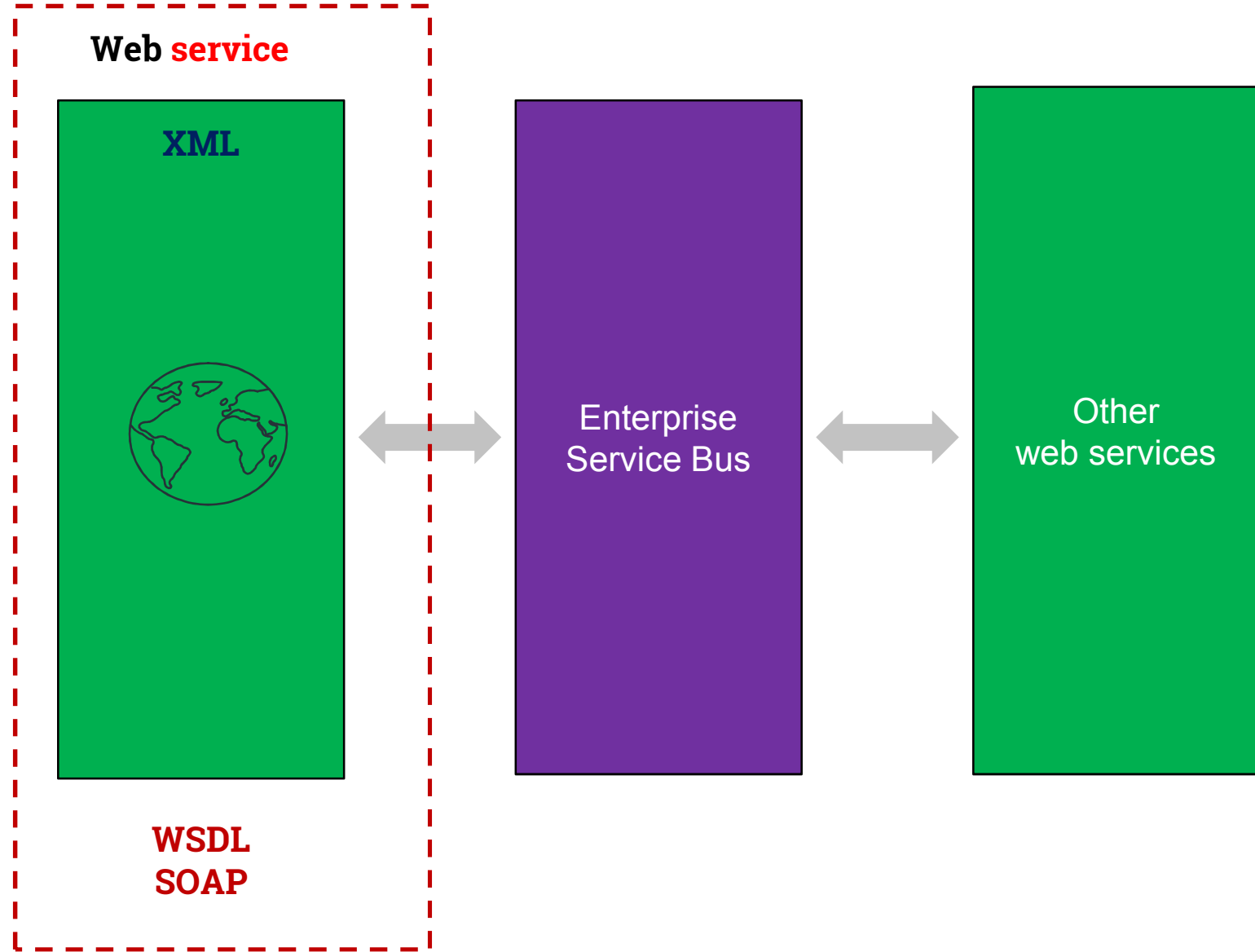
What if...?



What if...?



Service-oriented architecture



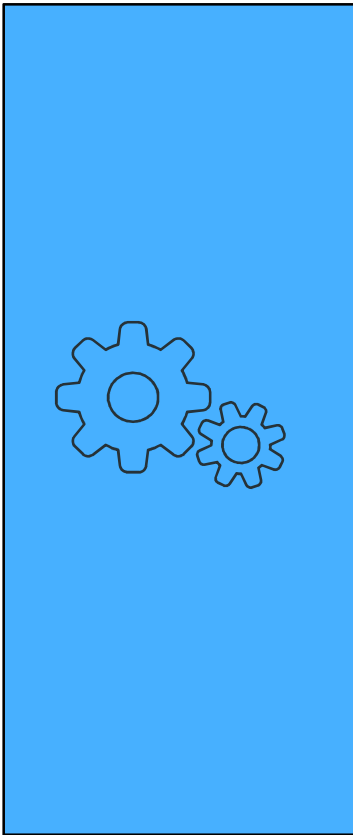
A decorative network diagram in the top-left corner, consisting of various sized circles (nodes) connected by thin lines (edges). Some nodes are solid grey, while others are hollow with a grey outline. The connections form a complex, branching structure.

How?

A decorative network diagram in the bottom-right corner, similar to the one in the top-left. It features a cluster of nodes connected by lines, with some nodes being solid grey and others hollow with a grey outline.

How?

Your program



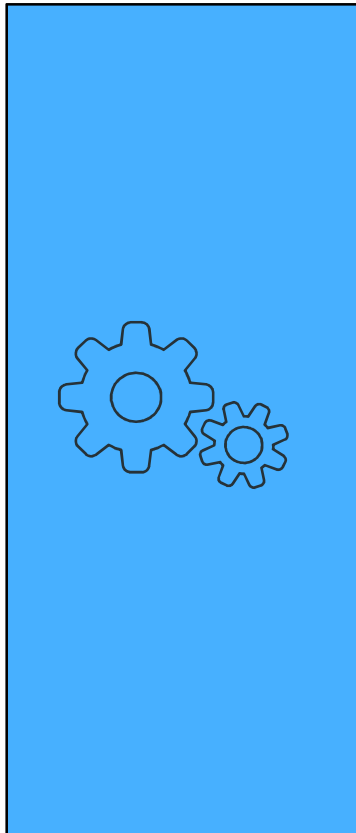
Option 1:

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?

Your program



Option 1:

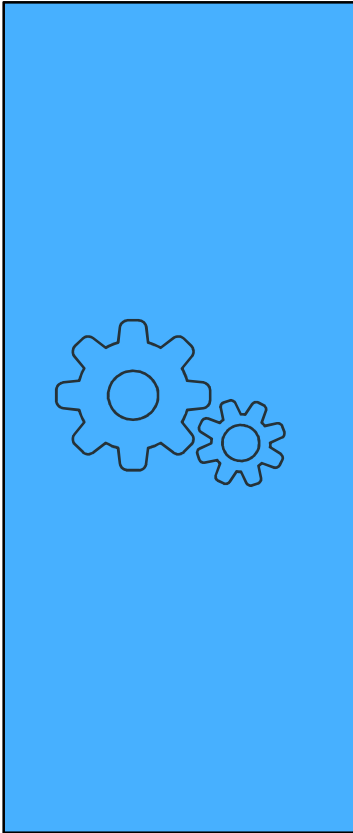
Translate Stata / Mata program into a **general-purpose programming language** used in web applications.

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



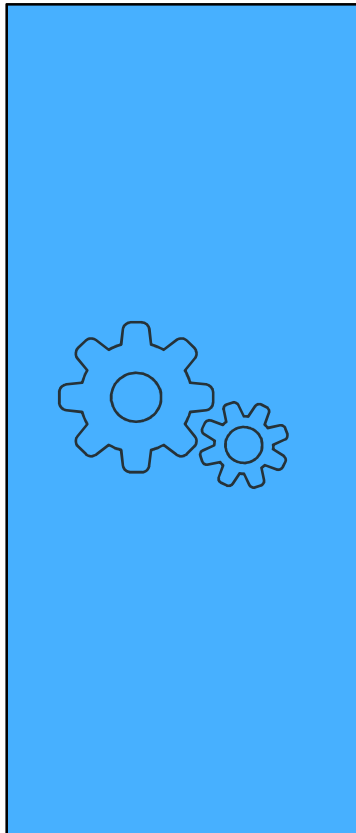
Option 2:

Translate Stata / Mata program into

R & RShiny or **SAS Stored Process Web Application**

How?

Your program



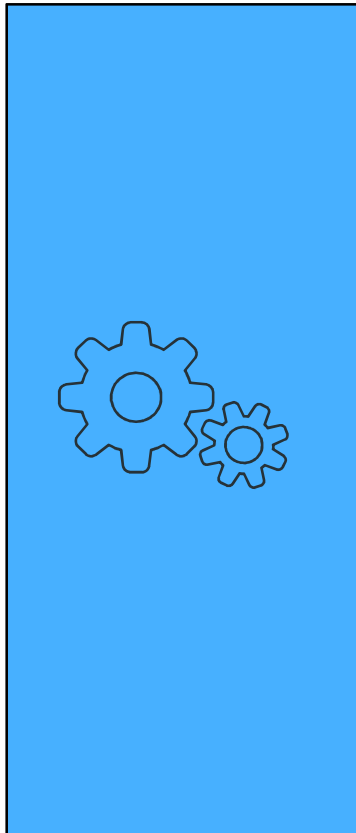
Option 2:

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?

Your program



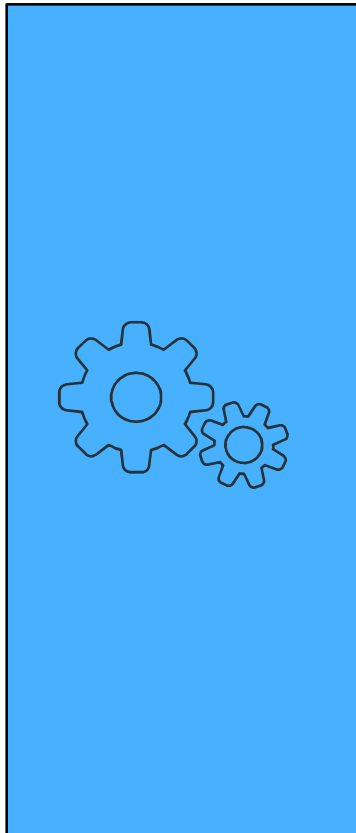
Option 3:

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



How?

Your program



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/web service** 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**

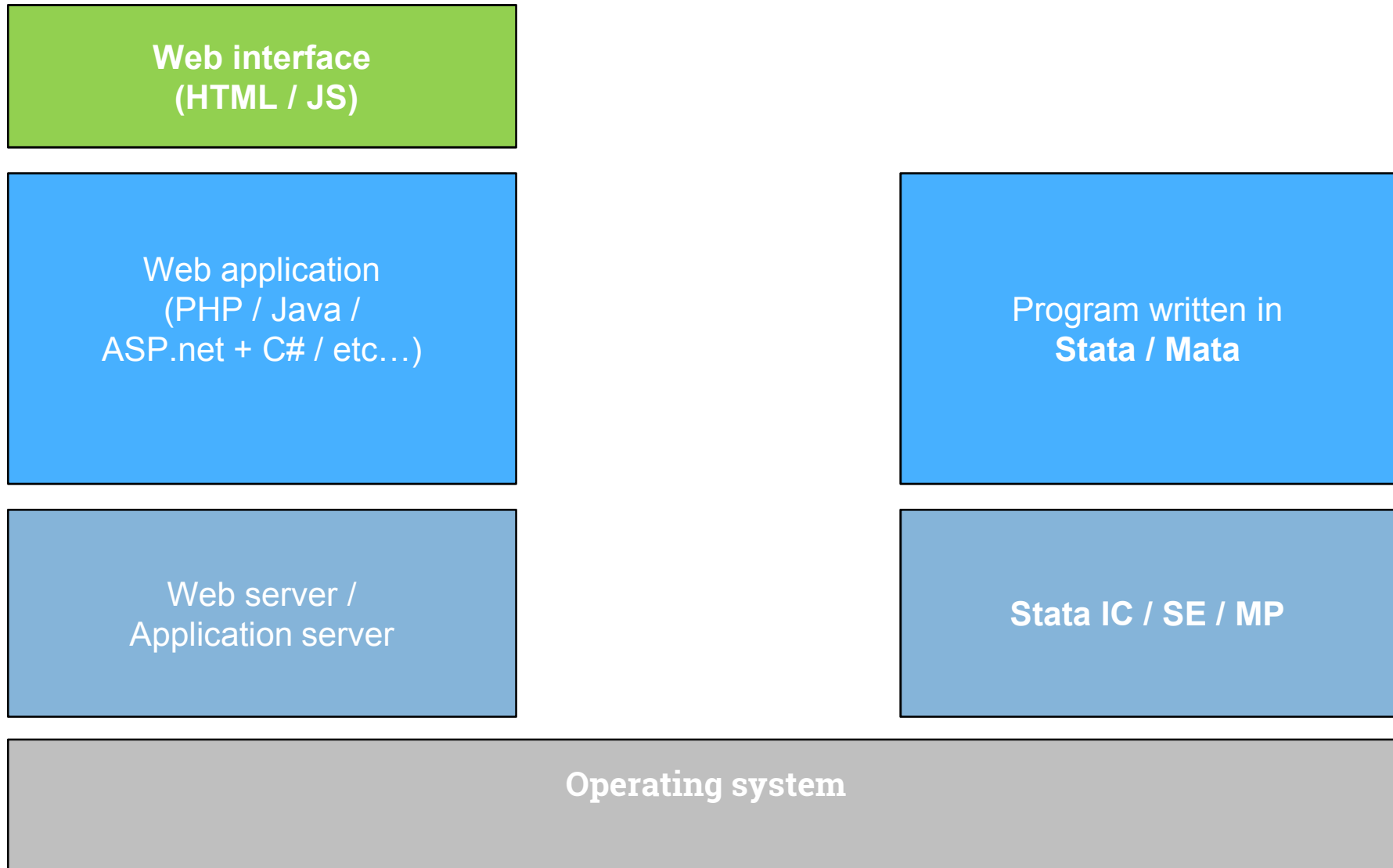
Other web servers, application containers and application servers may also be used:

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

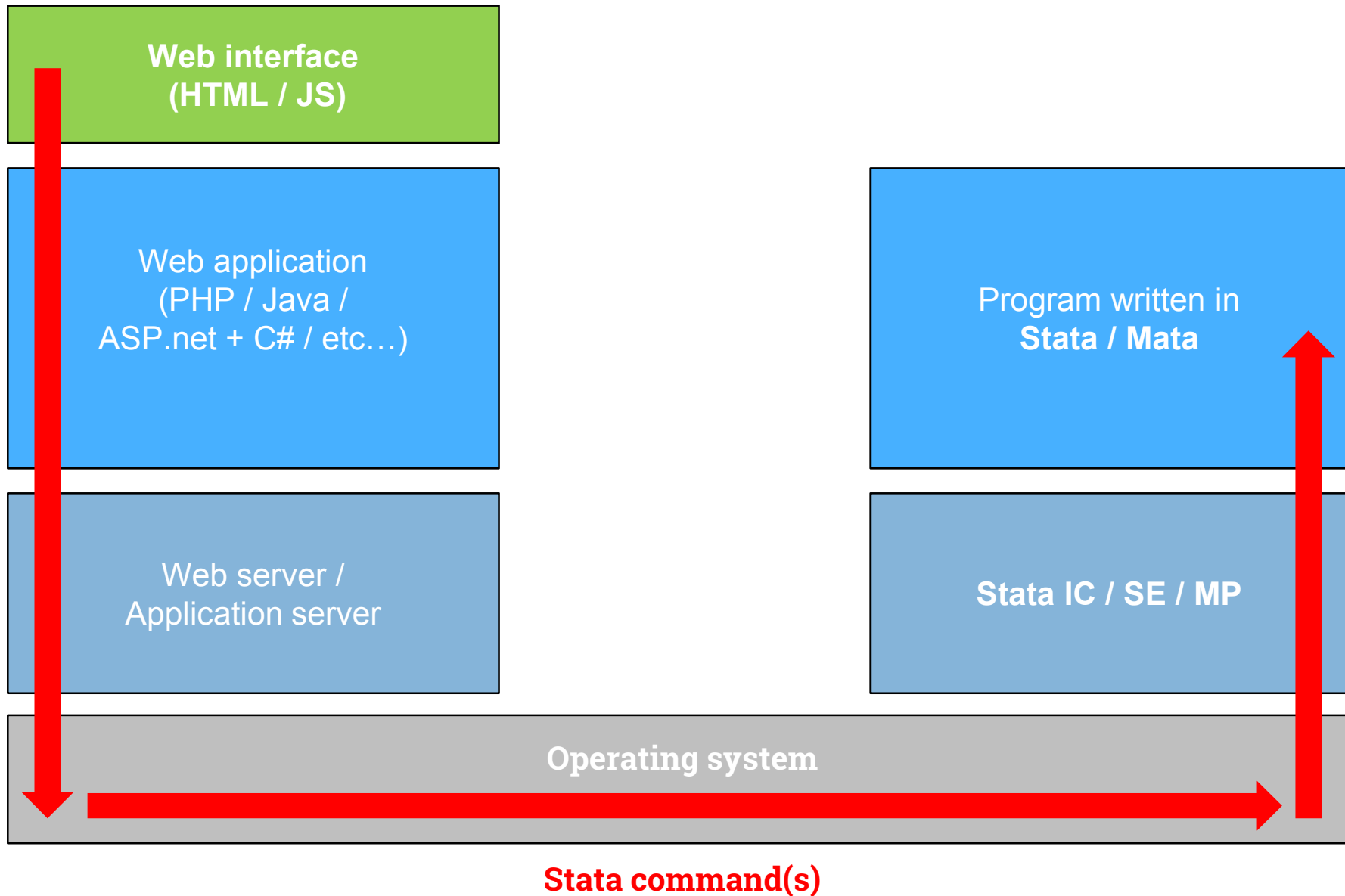
Operating system

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

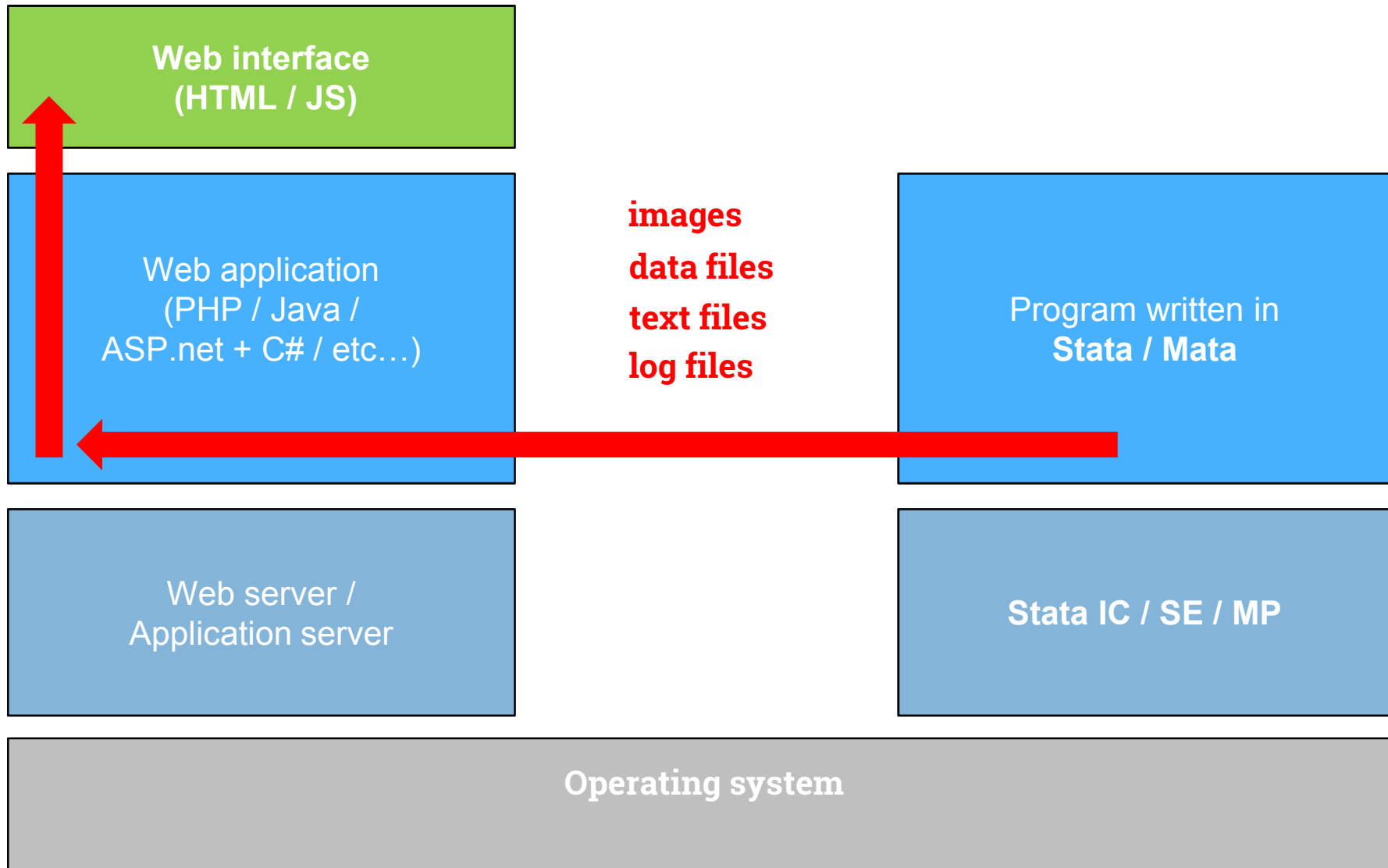
General idea



Calling Stata



Getting a response from Stata

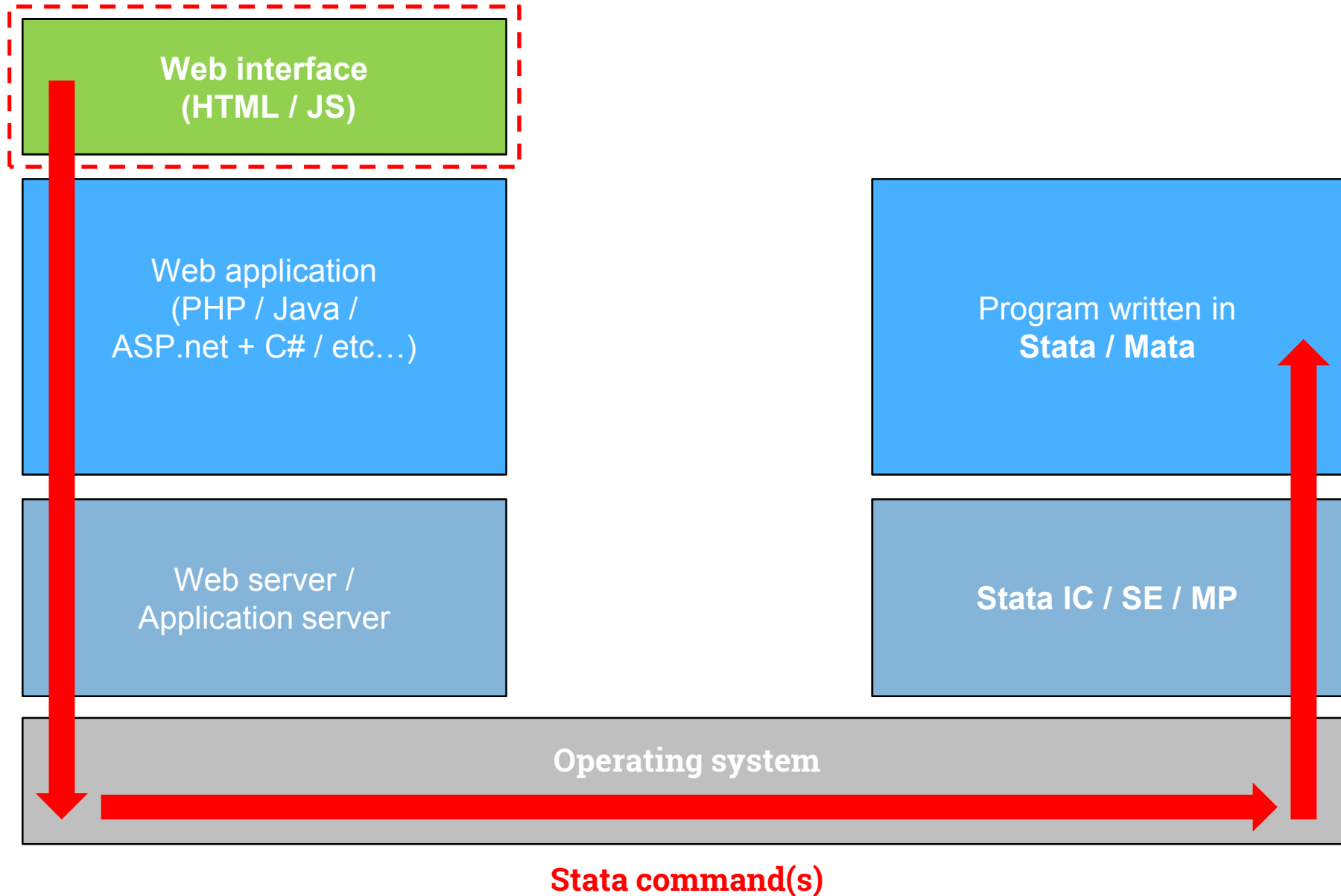


A decorative network diagram in the top-left corner, consisting of various nodes (some solid grey, some hollow white) connected by thin grey lines. The nodes are arranged in a complex, interconnected pattern.

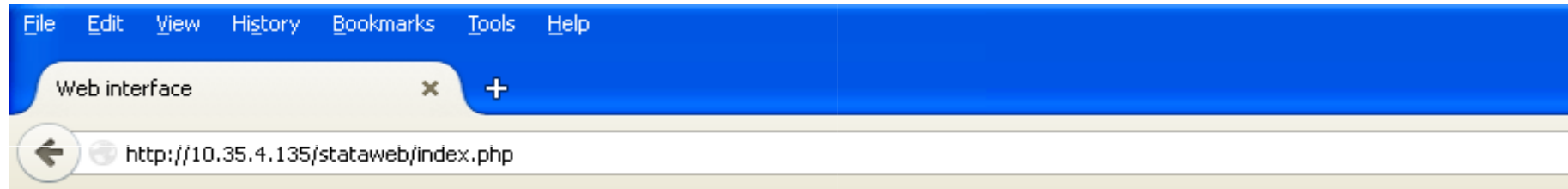
Simplified example

A decorative network diagram in the bottom-right corner, similar to the one in the top-left, with nodes and connecting lines.

Calling Stata



Calling Stata



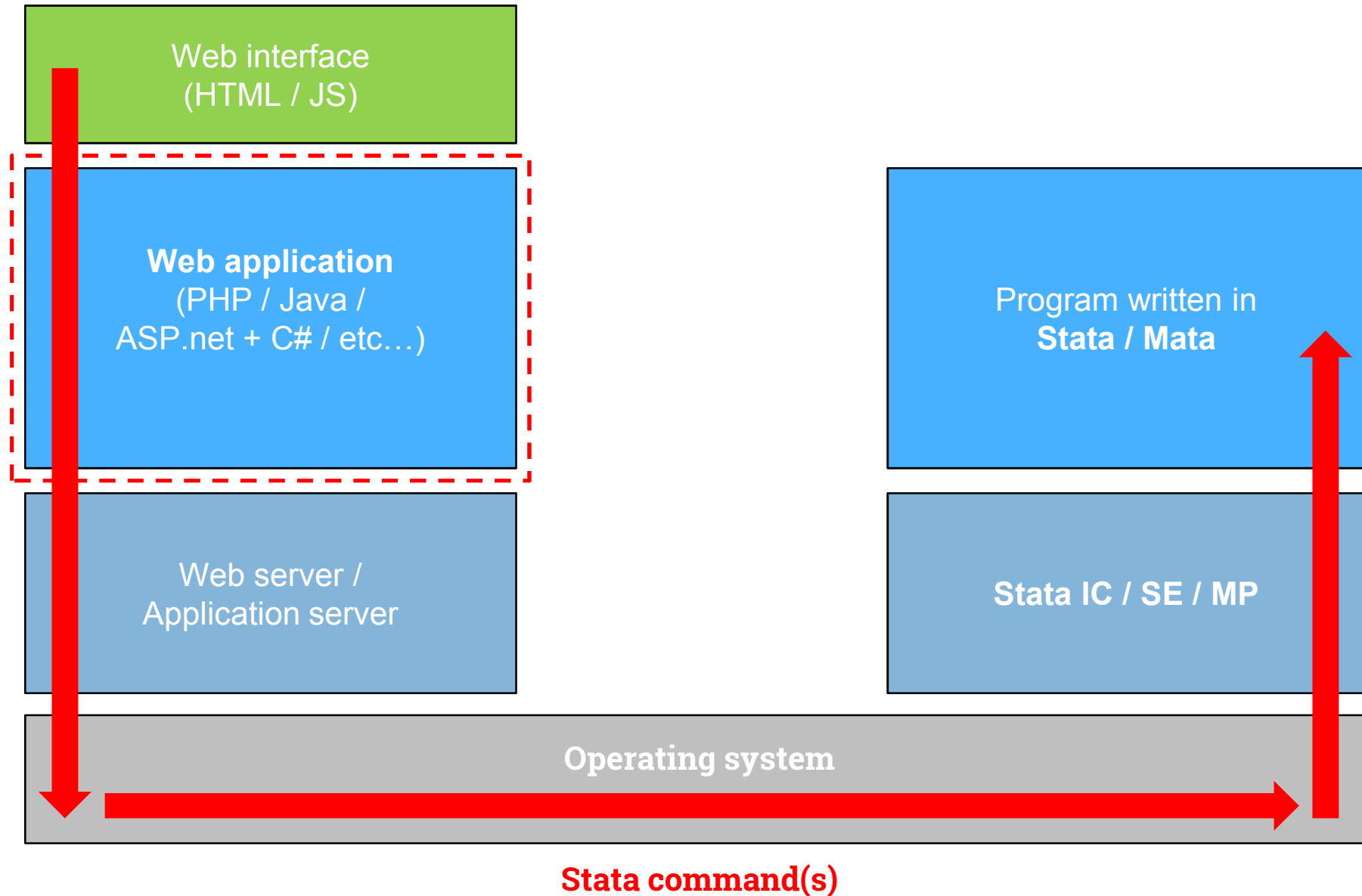
Stata command(s):

Send command(s) to 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



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"
```



Parameter	Result
<code>/q</code>	suppress logo and initialization messages
<code>/e</code>	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

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"
```

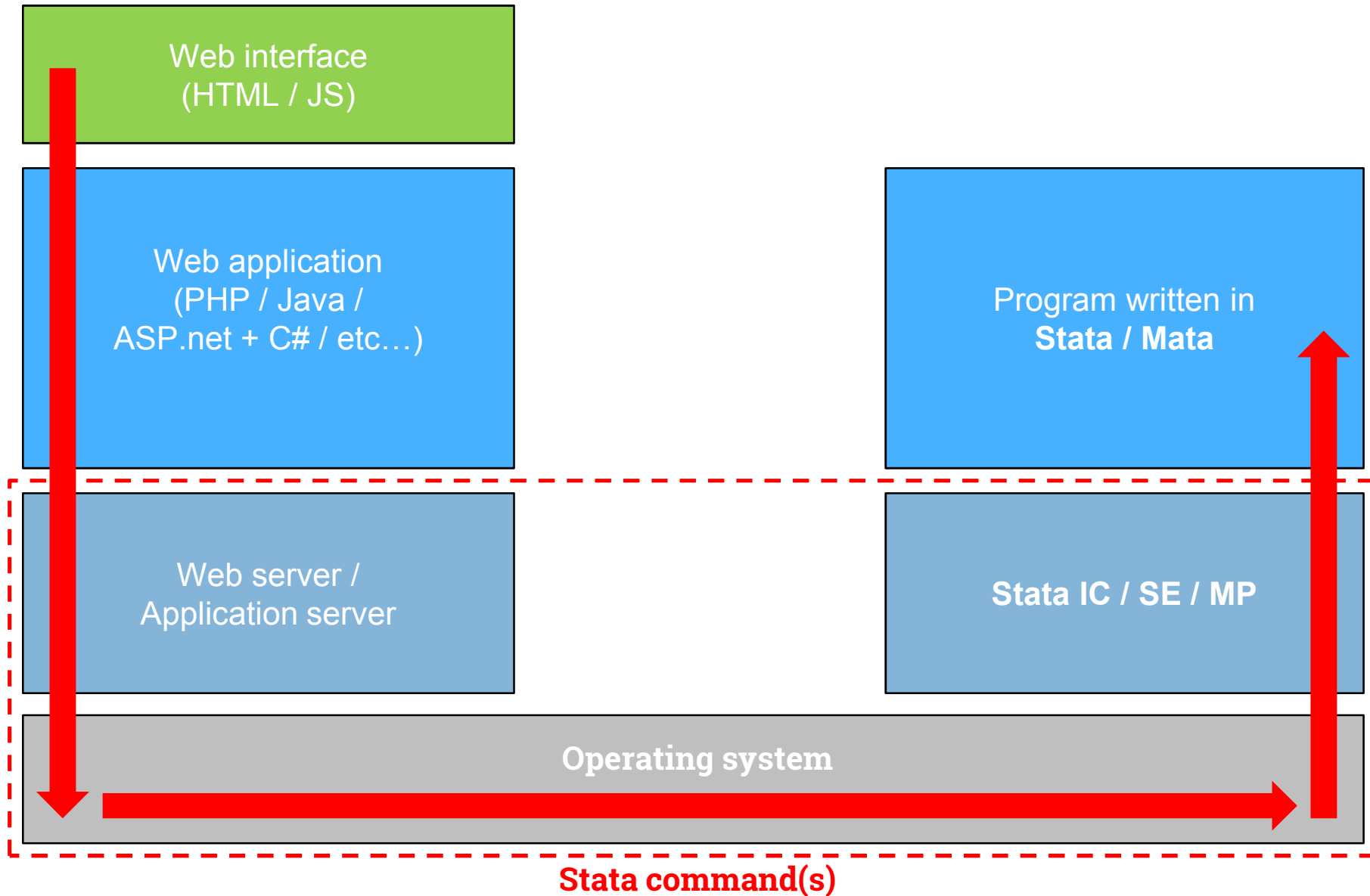
`$stata_commands`

We'll previously write our commands **here**

Example:

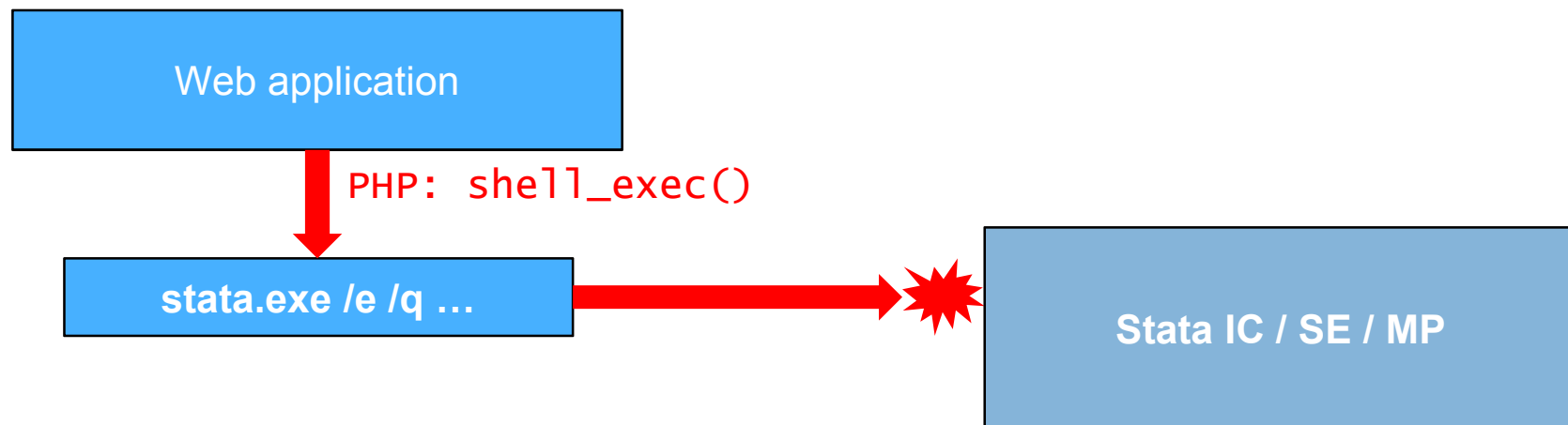
```
cd <path_to_temp_folder>  
sysuse auto  
histogram price
```

Calling Stata



Calling Stata

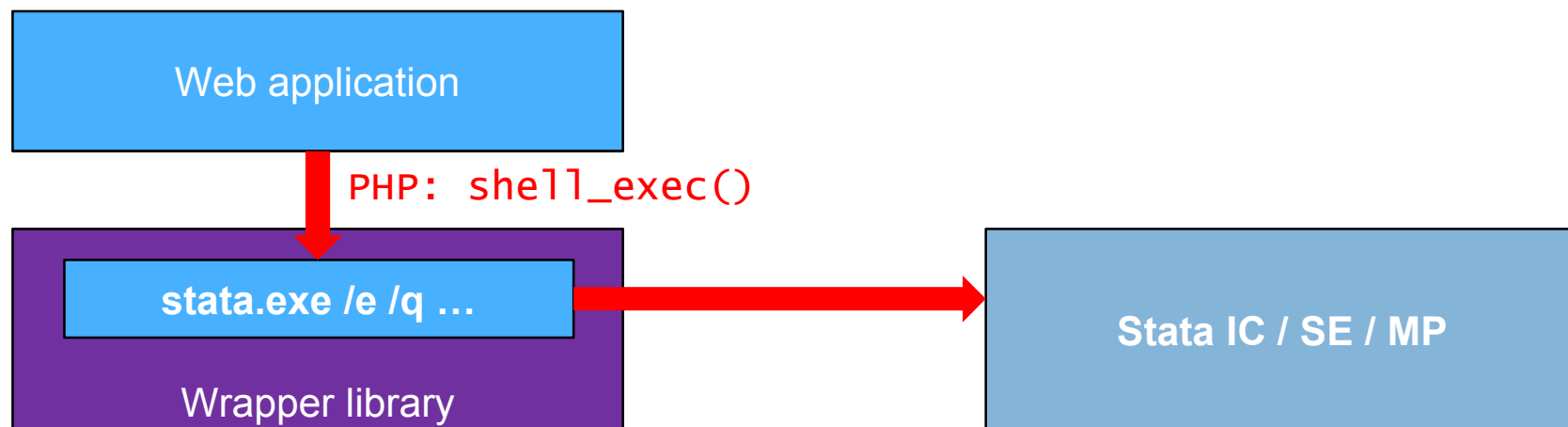
Problem: modern versions of Stata will **not** work if called directly from a web server (SYSTEM user).



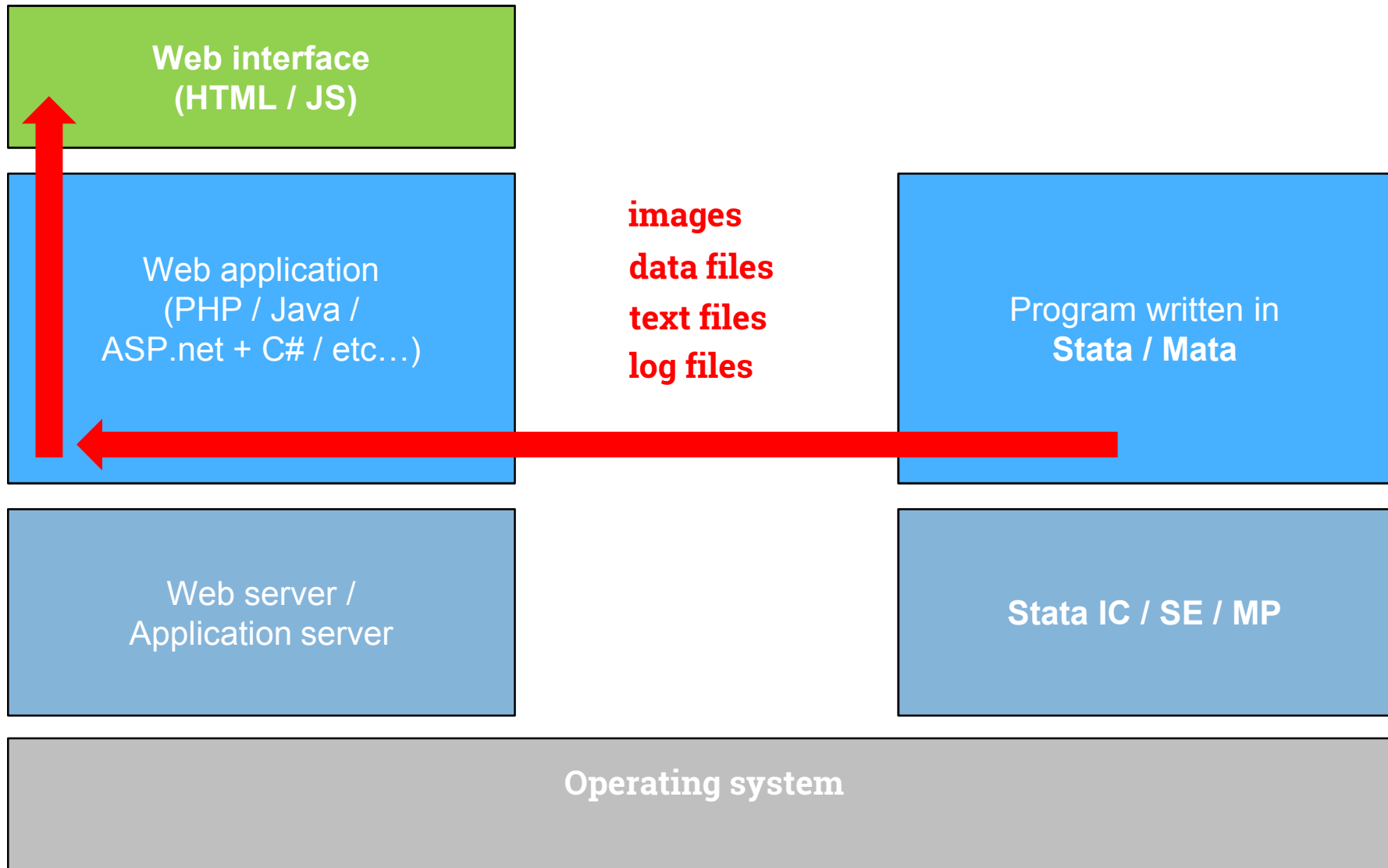
Calling Stata

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



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
```

```
...
```

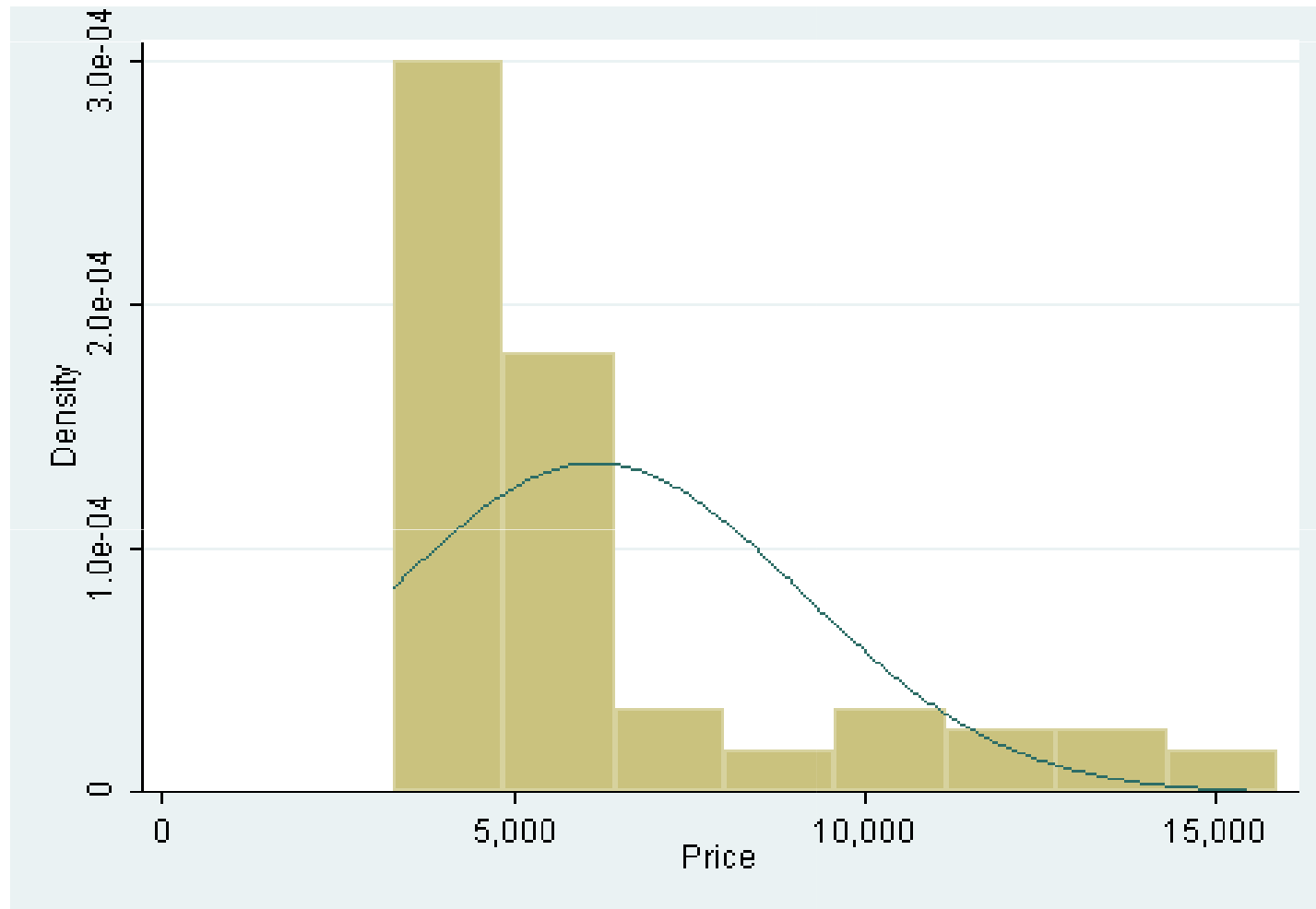
```
function display_results() {  
echo "<html>";  
echo " <head>Result</head>";  
echo " <body>";  
echo " <img src=img/graph01.png>";  
echo " </body>";  
echo "</html>";  
}
```

```
...
```

```
?>
```

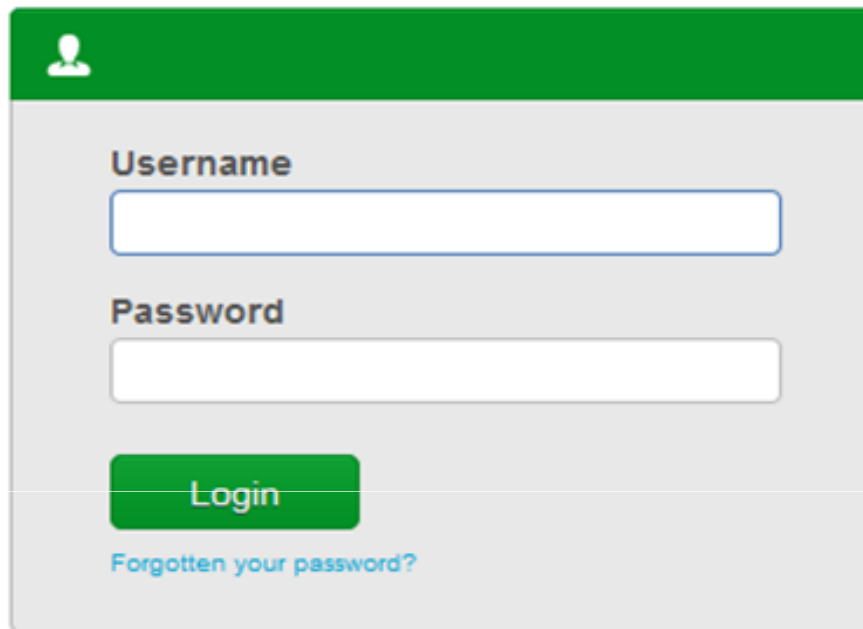
Getting a response from Stata

← <http://10.35.4.135/>



Basic security

SQL injection attack:



Username

Password

Login

[Forgotten your password?](#)

← **' ; DROP TABLE users ;**

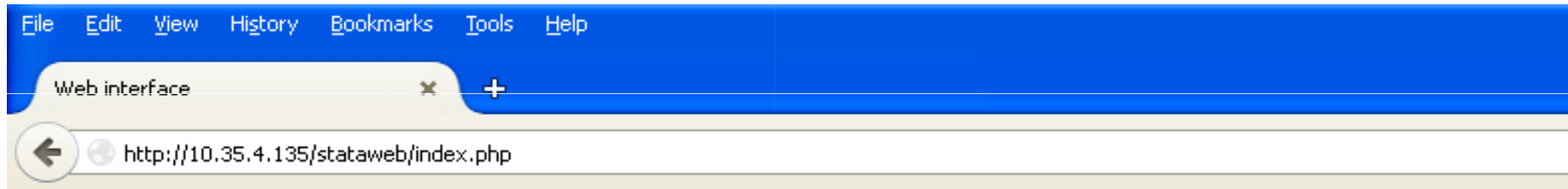
Basic security

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



Stata command(s):




Bad practice

Dataset:

Command:

Add normal-density plot

Add kernel-density plot



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

```
1  *! version 1.00.0
2  *authors:
3  program myshell
4  version 12
5
6  syntax [, ///  
7  cmd(string)
8
9  shell("`command`")
10
11 end program
12
```



Bad practice

```
1  *! version 1.00.0
2  *authors:
3  program myshell_better
4  version 12
5
6  syntax [, ///  
7  params(string)
8
9  //only pass parameters to a specific command
10 shell("externalprogram.exe ""`params`"" ")
11
12 end program
13
```



Better practice

Basic security

```
1  *! version 1.00.0
2  *authors:
3  program myshell
4  version 12
5
6  syntax [, ///
7  cmd(string) ]
8
9  shell("`command'")
10
11 end program
12
```



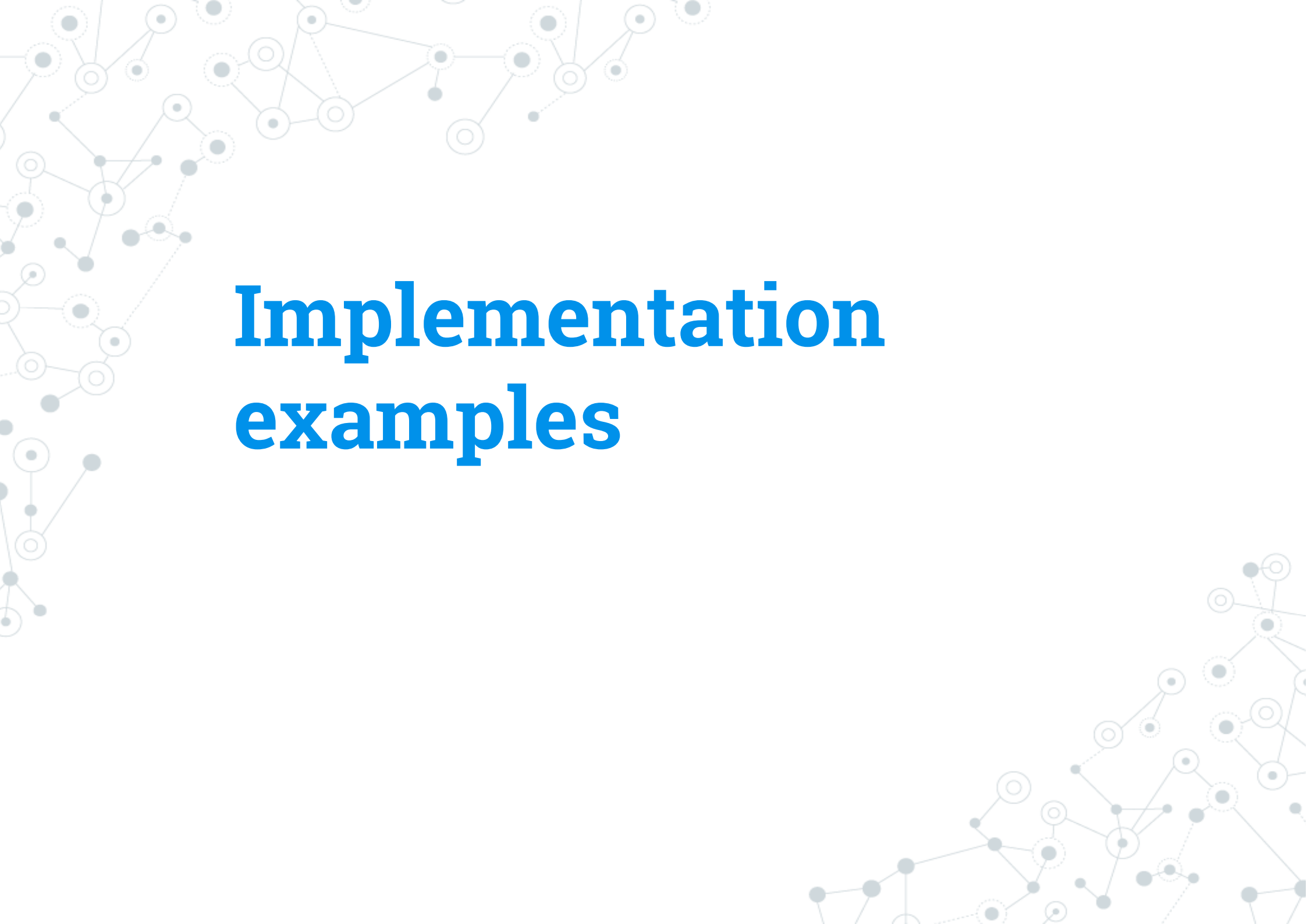
Bad practice

```
1  *! version 1.00.0
2  *authors:
3  program myshell_better
4  version 12
5
6  syntax [, ///
7  params(string) ]
8
9  //only pass parameters to a specific command
10 shell("externalprogram.exe ""`params'"" ")
11
12 end program
13
```



Better practice

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

A decorative background featuring a network diagram with nodes and connecting lines, primarily visible on the left and bottom-right sides. The nodes are represented by circles of varying sizes and colors (grey, white, and blue), connected by thin grey lines. Some nodes are highlighted with a double-circle effect.

Implementation examples

Web interface for –coin–

**Studying coincidences with network analysis
and other multivariate tools**

Modesto Escobar. Stata Journal. 2015 (*in press*)

coin - Analysis of coincidences

Main Plots if/in Weights Export

Dichotomous or factor variables:

Outputs

- Frequencies Minimum:
- Column percentages
- Row percentages
- Relative frequencies (%)
- Relative frequencies (%) - missing included
- Relative frequencies (%) - controlled for over var
- Odds ratios
- Tetrachoric correlations

Residuals and significances

- Expected values
- Residuals
- Standard residuals
- Normalized
- Significance of normalized residuals
- Fisher exact test
- Significance of odd ratios
- Standard errors of ln(odd ratios)

Distance matrix habeman Adjacency matrix p <= Support:

Similarity matrix Centrality measures Bonferroni adjusted

Coordinates Sorted list of

Sort

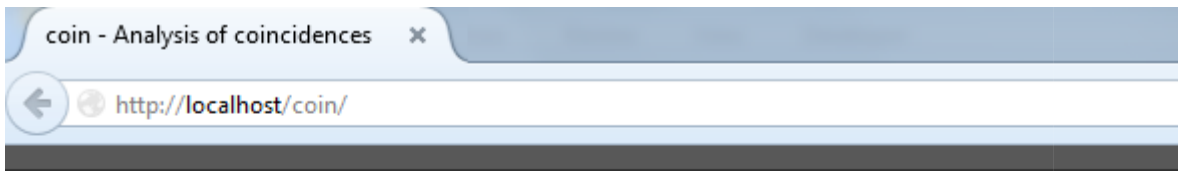
List order Ascending order Descending order

Over

Variables head:

Where...

OK Cancel Submit



coin - Analysis of coincidences

Main Plots Output options

Variables:

Vars. head (Omni, Axiom, Affi, ExomeLOF):

Legend:

Outputs

Frequencies

Column percentages

Row percentages

Odds ratios

Residuals and significances

Residuals

Standardized residuals

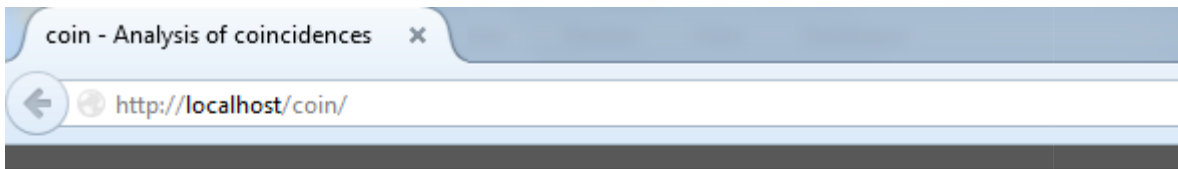
Normalized

Significance or normalized residuals

Distance matrix haberman Adjacency matrix p<= .50 Support 0

Similarity matrix Centrality measures Bonferroni-adjusted

OK Cancel



coin - Analysis of coincidences

coin - Analysis of coincidences

Main Plots Output options

Variables:

Vars. head (Omni, Axiom, Affi, ExomeLOF):

Legend:

Outputs

Frequencies

Column percentages

Row percentages

Odds ratios

Residuals and significances

Residuals

Standardized residuals

Normalized

Significance or normalized residuals

Distance matrix haberman Adjacency matrix p<= .50 Support 0

Similarity matrix Centrality measures Bonferroni-adjusted

OK Cancel

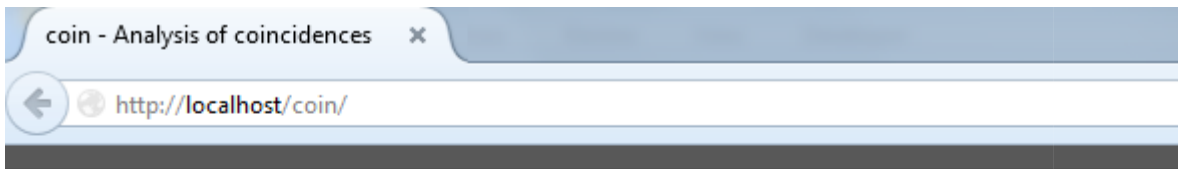
Execution output:



Omni, Axiom, Affy, Exome/LOF

Has Omni Genotypes, Has Axiom Genotypes, Has Affy 6 0 Genotypes, Has Exome/LOF Genotypes

Frequencies				
Has Omni Genotypes	2098			
Has Axiom Genotypes	880	977		
Has Affy 6 0 Genotypes	3	0	1195	
Has Exome/LOF Genotypes	2055	851	3	2063
Col. percentages				
Has Omni Genotypes	100.0	90.1	0.3	99.6
Has Axiom Genotypes	41.9	100.0	0.0	41.3
Has Affy 6 0 Genotypes	0.1	0.0	100.0	0.1
Has Exome/LOF Genotypes	98.0	87.1	0.3	100.0



coin - Analysis of coincidences

coin - Analysis of coincidences

Main Plots Output options

Variables: Gender

Vars. head (Omni, Axiom, Affi, ExomeLOF):

Legend:

Outputs

Frequencies

Column percentages

Row percentages

Odds ratios

Residuals and significances

Residuals

Standarized residuals

Normalized

Significance or normalized residuals

Distance matrix haberman

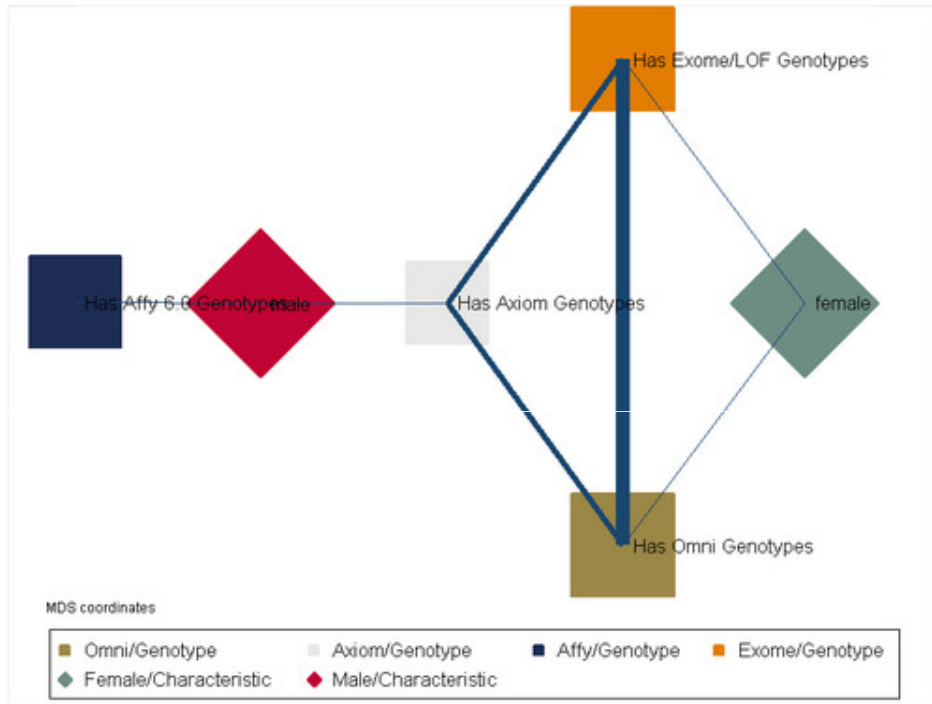
Adjacency matrix p<= .50 Support 0

Similarity matrix

Centrality measures Bonferroni-adjusted

OK Cancel

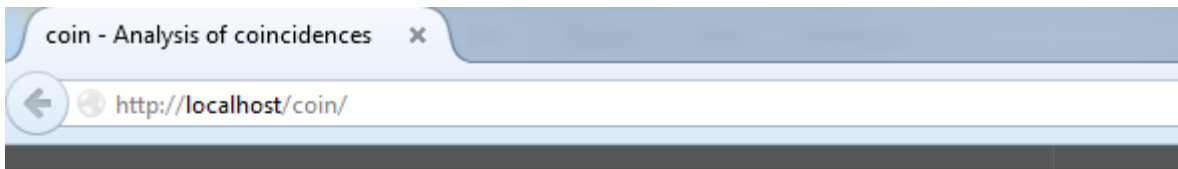
Execution output:



female,male,Omni,Axiom,Affy,ExomeLOF

female, male, Has Omni Genotypes, Has Axiom Genotypes, Has Affy 6 0 Genotypes, Has Exome/LOF Genotypes

Frequencies						
female	1760					
male	0	1740				
Has Omni Genotypes	1070	1028	2098			
Has Axiom Genotypes	488	489	880	977		
Has Affy 6 0 Genotypes	581	614	3	0	1195	
Has Exome/LOF Genotypes	1062	1001	2055	851	3	2063
Col. percentages						
female	100.0	0.0	51.0	49.9	48.6	51.5
male	0.0	100.0	49.0	50.1	51.4	48.5
Has Omni Genotypes	60.8	59.1	100.0	90.1	0.3	99.6
Has Axiom Genotypes	27.7	28.1	41.9	100.0	0.0	41.3
Has Affy 6 0 Genotypes	33.0	35.3	0.1	0.0	100.0	0.1
Has Exome/LOF Genotypes	60.3	57.5	98.0	87.1	0.3	100.0



coin - Analysis of coincidences

Main | Plots | Output options

Variables:

Vars. head (Omni, Axiom, Affi, ExomeLOF):

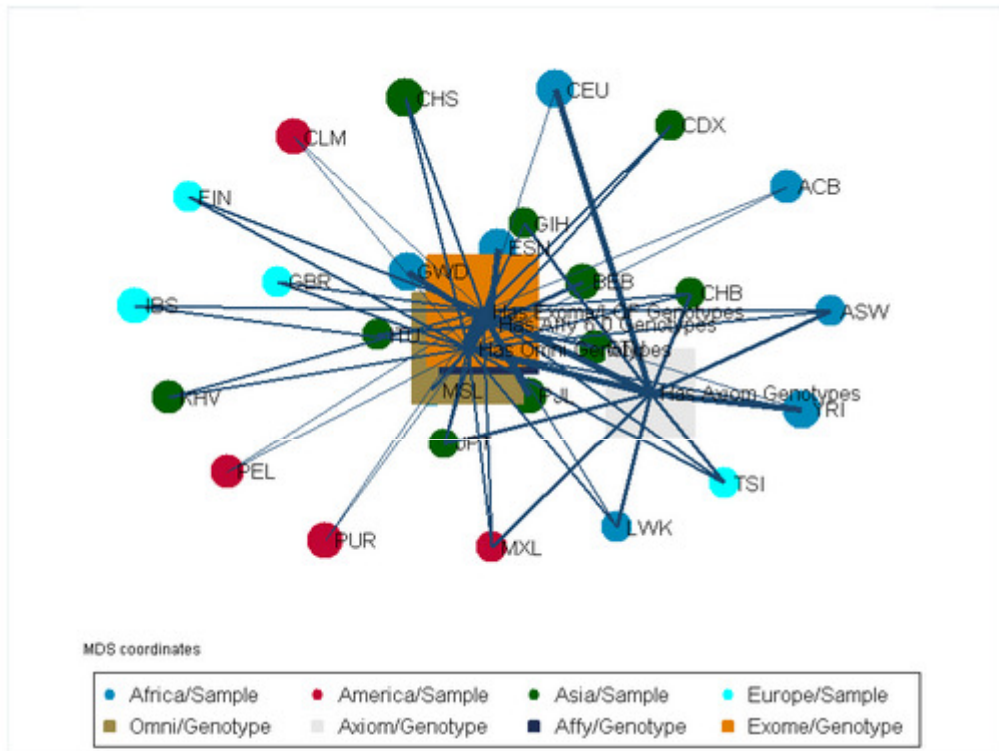
Legend:

Outputs	Residuals and significances
<input checked="" type="checkbox"/> Frequencies	<input type="checkbox"/> Residuals
<input checked="" type="checkbox"/> Column percentages	<input type="checkbox"/> Standardized residuals
<input type="checkbox"/> Row percentages	<input type="checkbox"/> Normalized
<input type="checkbox"/> Odds ratios	<input type="checkbox"/> Significance or normalized residuals

Distance matrix Adjacency matrix

Similarity matrix Centrality measures Bonferroni-adjusted

Execution output:



ACB, ASW, BEB, CDX, CEU, CHB, CHS, CLM, ESN, FIN, GBR, GIH, GWD, IBS, ITU, JPT, KHV, LWK, MSL, MXL, PEL, PJI, PUR, STU, ACB, ASW, BEB, CDX, CEU, CHB, CHS, CLM, ESN, FIN, GBR, GIH, GWD, IBS, ITU, JPT, KHV, LWK, MSL, M

Frequencies |

ACB	123						
ASW	0	112					
BEB	0	0	144				
CDX	0	0	0	109			
CEU	0	0	0	0	183		
CHB	0	0	0	0	0	108	
CHS	0	0	0	0	0	0	171
CLM	0	0	0	0	0	0	0
ESN	0	0	0	0	0	0	0
FIN	0	0	0	0	0	0	0
GBR	0	0	0	0	0	0	0
GIH	0	0	0	0	0	0	0
GWD	0	0	0	0	0	0	0
IBS	0	0	0	0	0	0	0

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

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

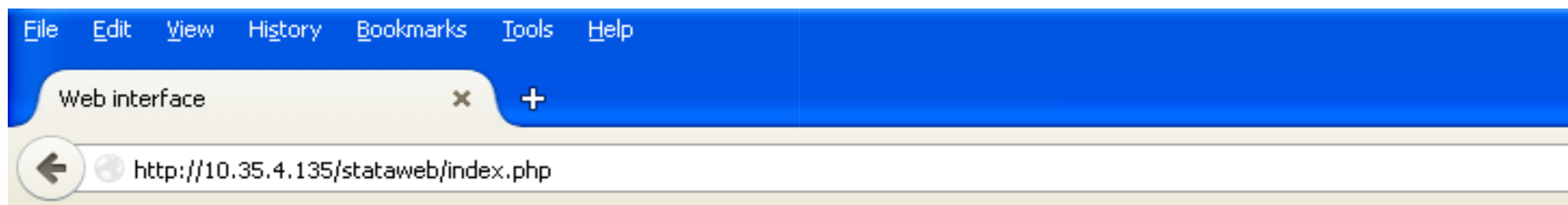
Max N of chars to display in variable name labels (default: 240)
240

Size of data labels (default: 2)
2.0

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

? P [Print] [Help]

OK Cancel Submit



nomolog - Logistic regression nomogram generator

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

Graph title

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)

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

Size of data labels (default: 2)

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



nomolog - Logistic regression nomogram generator

nomolog - Logistic regression nomogram generator

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


Dataset auto

Command logistic foreign i.rep78 turn mpg weight

Display input commands

Display full execution log

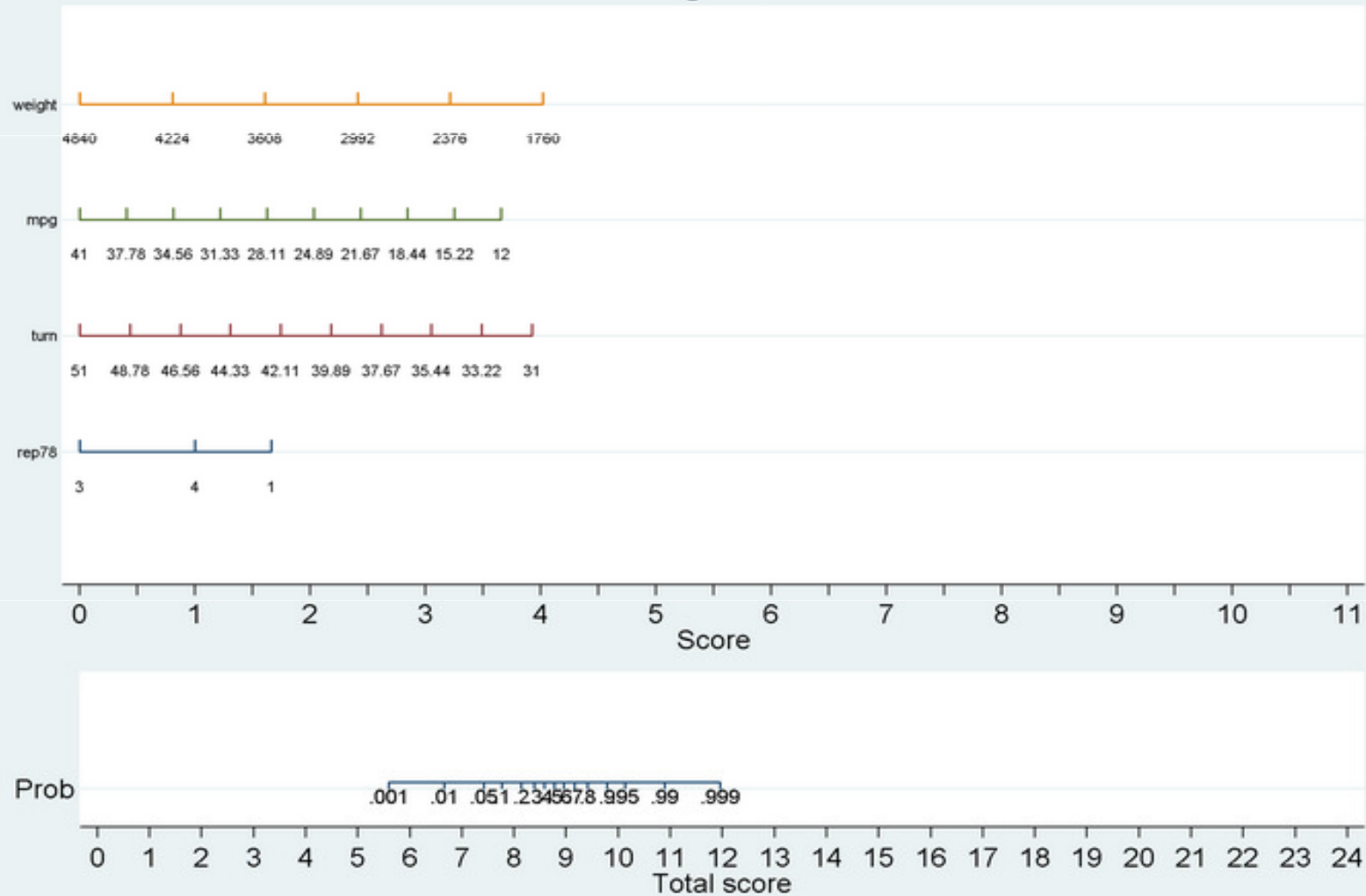
OK Cancel



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



Nomogram



The background of the image is a light gray grid with a pattern of interconnected nodes and lines, resembling a network or molecular structure. The nodes are represented by small circles, some of which are solid dark gray, while others are hollow with a dark gray outline. The lines connecting the nodes are thin and light gray, creating a complex web of connections across the entire page.

Questions?

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 \(license\)](#)