



Debugging - Bug stories

Steven Costiou

steven.costiou@inria.fr

RMoD / Inria Lille - Nord Europe

September 2022

Evil coffee machines

Debugging: The 9 indispensable rules for finding even the most elusive software and hardware problems, David J. Agans, 2002 (p.34)

Evil coffee machines

Debugging: The 9 indispensable rules for finding even the most elusive software and hardware problems, David J. Agans, 2002



Crashing every day
around the same time



Mainframe computer (source: wikipedia)

Evil coffee machines

Debugging: The 9 indispensable rules for finding even the most elusive software and hardware problems, David J. Agans, 2002



Crashing every day
around the same time



But in different places
in the program

Mainframe computer (source: wikipedia)

Evil coffee machines

Debugging: The 9 indispensable rules for finding even the most elusive software and hardware problems, David J. Agans, 2002



Around the same
time, in the
cafeteria...




Crowded cafeteria (source: wikipedia)

Evil coffee machines

Debugging: The 9 indispensable rules for finding even the most elusive software and hardware problems, David J. Agans, 2002



Around the same time, in the cafeteria...

During the 3pm coffee break, people started ordering coffees 




Crowded cafeteria (source: wikipedia) ⁶

Evil coffee machines

Debugging: The 9 indispensable rules for finding even the most elusive software and hardware problems, David J. Agans, 2002



Around the same time, in the cafeteria...

During the 3pm coffee break, people started ordering coffees 

Lots of coffees...



Crowded cafeteria (source: wikipedia) ⁷

Evil coffee machines

Debugging: The 9 indispensable rules for finding even the most elusive software and hardware problems, David J. Agans, 2002



Random bugs

Power drain



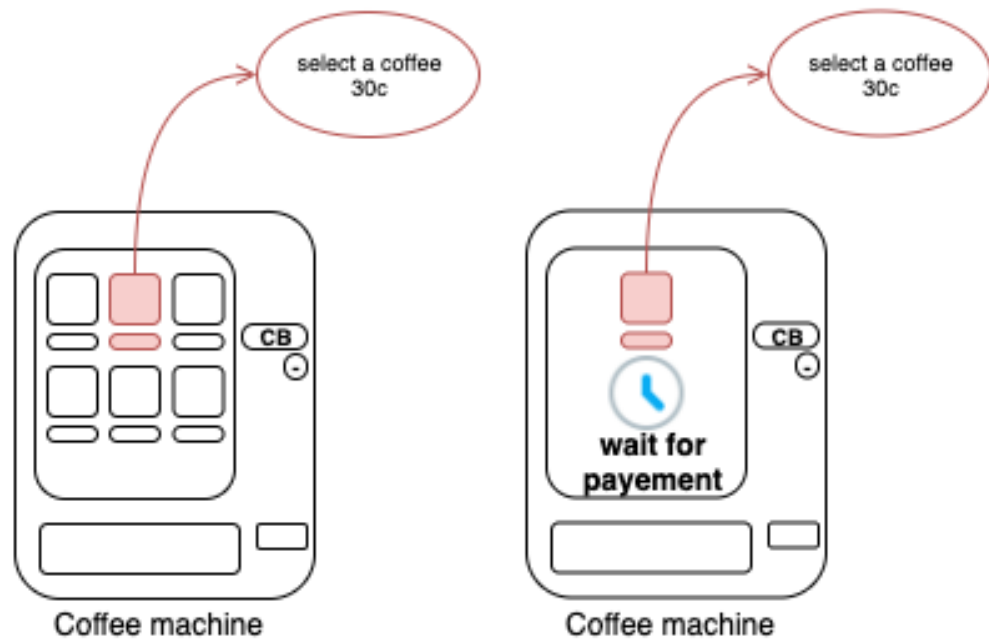
Coffee machines
(source: wikipedia)

From real experience.

The coffee machines timeout at Univ. Lille

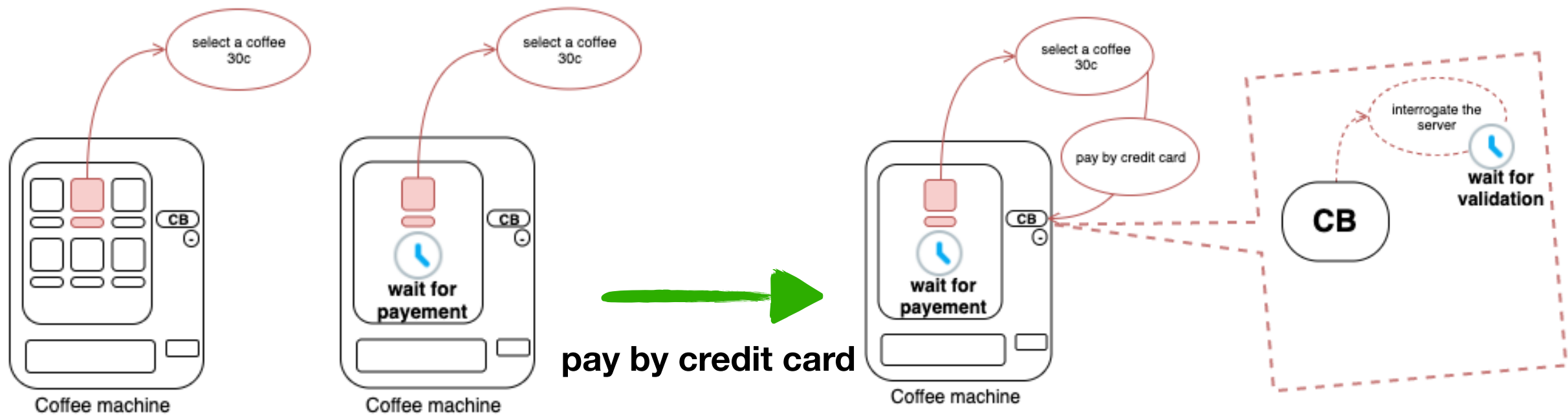
(you can even try it!)

The coffee machines timeout at Univ. Lille



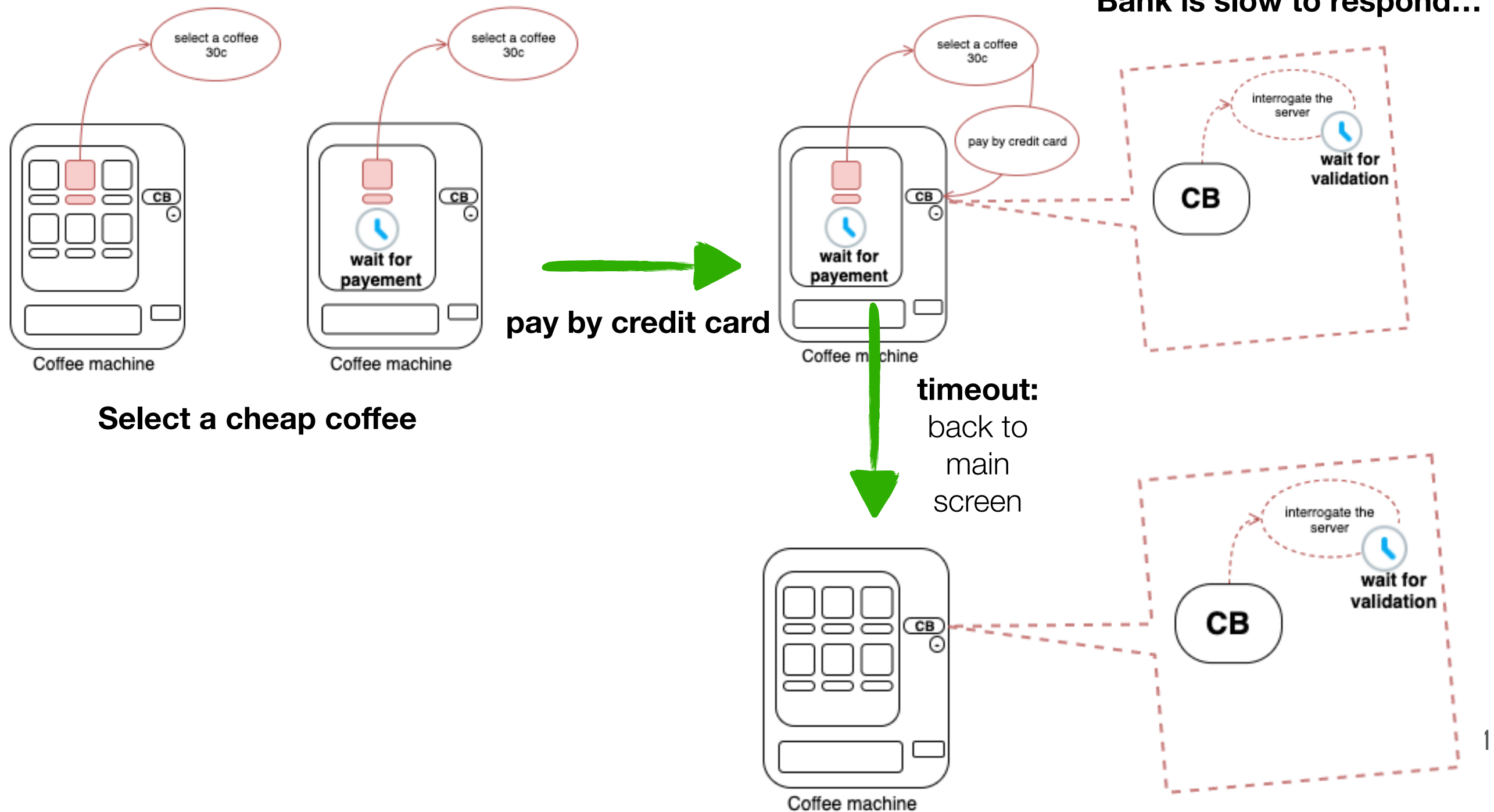
Select a cheap coffee

The coffee machines timeout at Univ. Lille

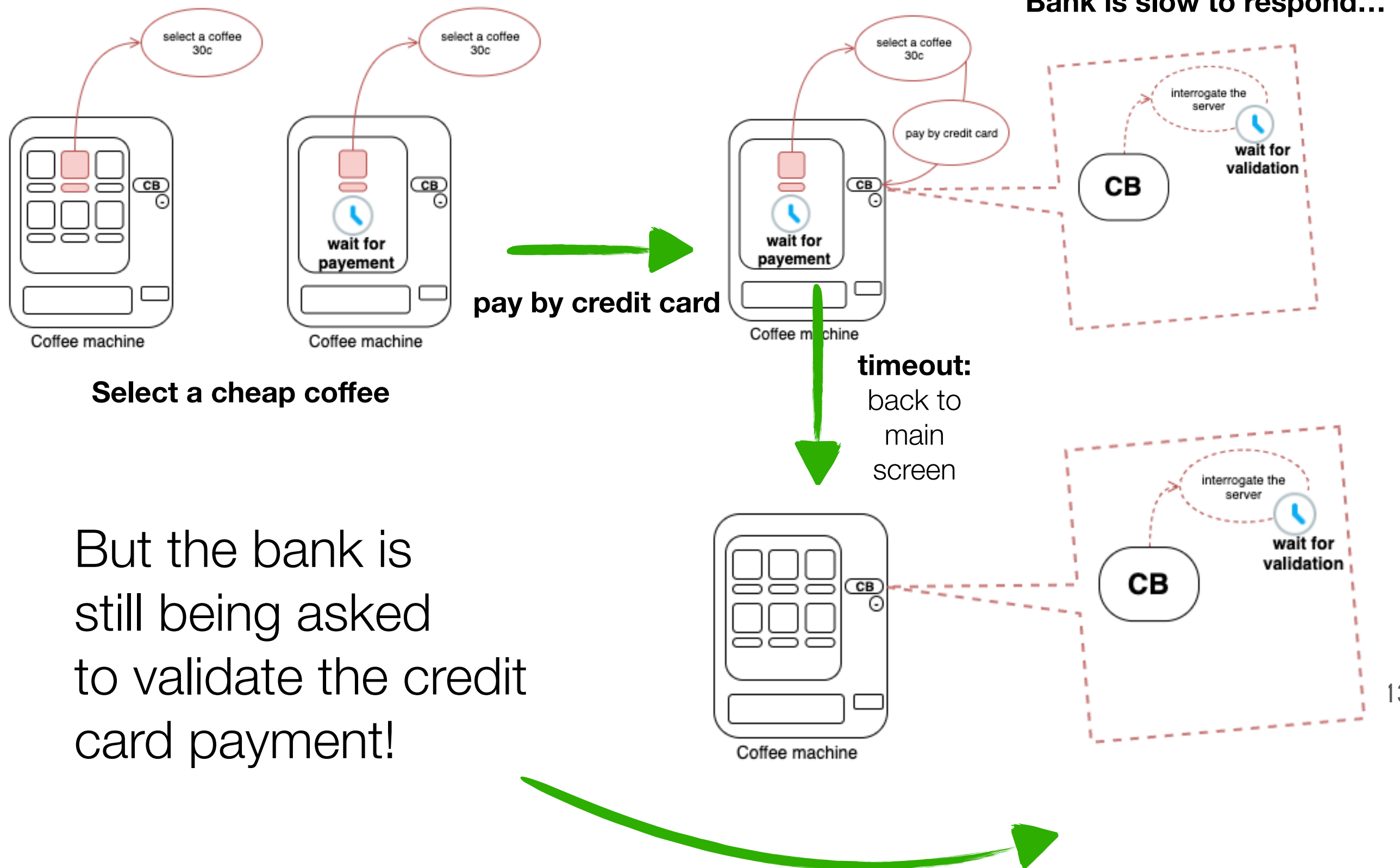


Select a cheap coffee

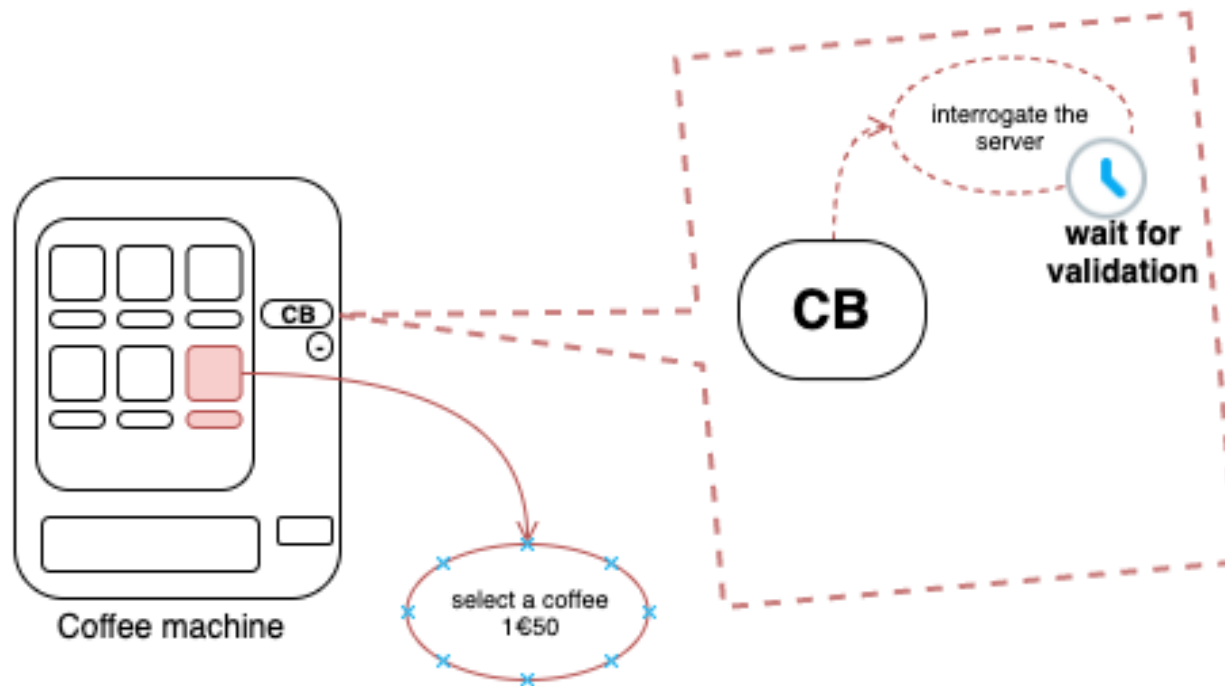
The coffee machines timeout at Univ. Lille



The coffee machines timeout at Univ. Lille

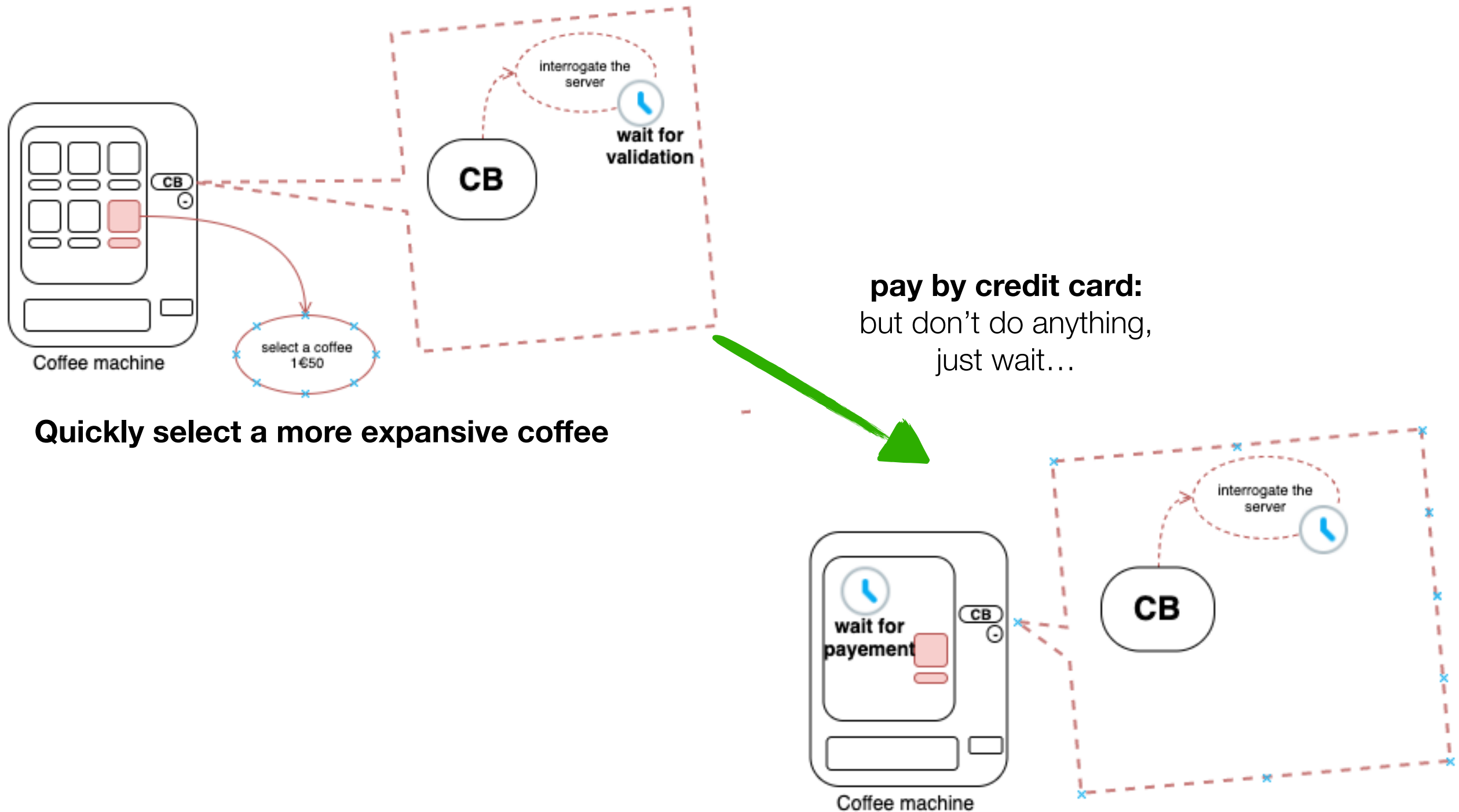


The coffee machines timeout at Univ. Lille

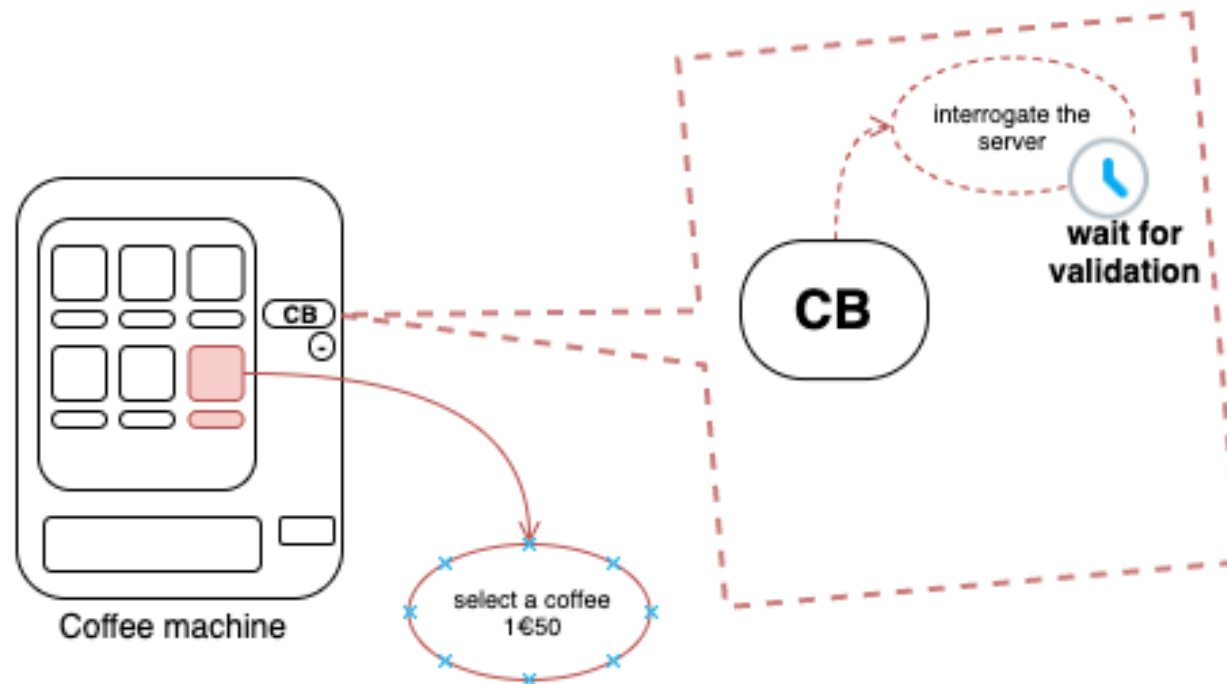


Quickly select a more expensive coffee

The coffee machines timeout at Univ. Lille

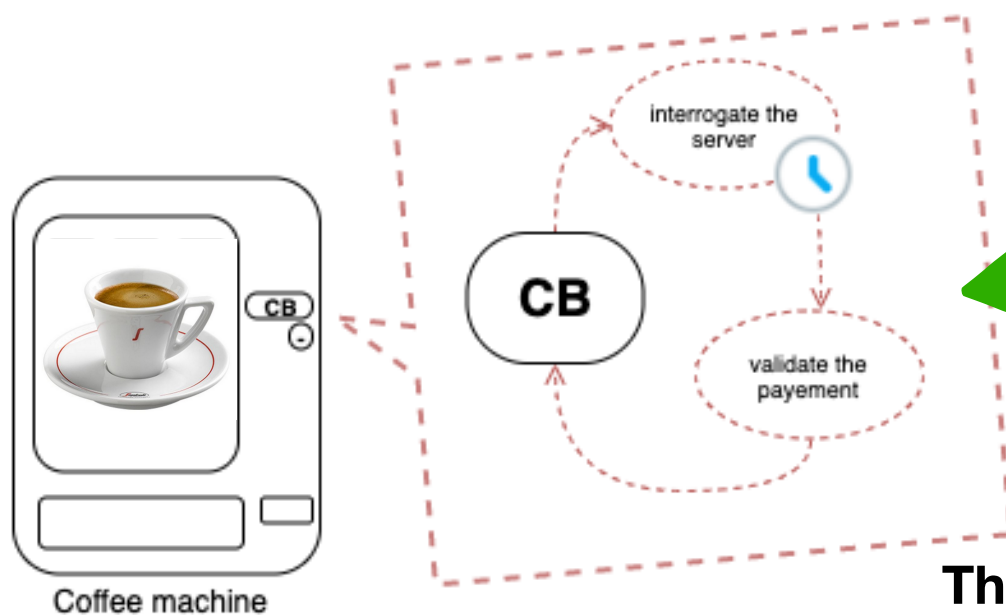


The coffee machines timeout at Univ. Lille



Quickly select a more expansive coffee

pay by credit card:
but don't do anything,
just wait...



The bank finally responds:
cheap payment is validated
but you get served the expansive coffee!

Only wednesdays

My Hairiest Bug War Stories,
Mark Eisenstadt, 1997 (Story B)

Only Wednesdays

My Hairiest Bug War Stories,
Mark Eisenstadt, 1997

- A program was only working properly on wednesdays...
- **Documentation stated that day of the week was returned in a doubleword — 8 bytes**
 - But as Wednesday is 9 characters long, the system was expecting a three words array — 12 bytes
 - For the six other days of the week, the additional 4 bytes were programmatically filled out with blanks, as only 8 bytes were necessary

M	O	N	D	A	Y						
T	U	E	S	D	A	Y					
W	E	D	N	E	S	D	A	Y			
T	H	U	R	S	D	A	Y				
F	R	I	D	A	Y						
S	A	T	U	R	D	A	Y				
S	U	N	D	A	Y						

Only Wednesdays

My Hairiest Bug War Stories,
Mark Eisenstadt, 1997

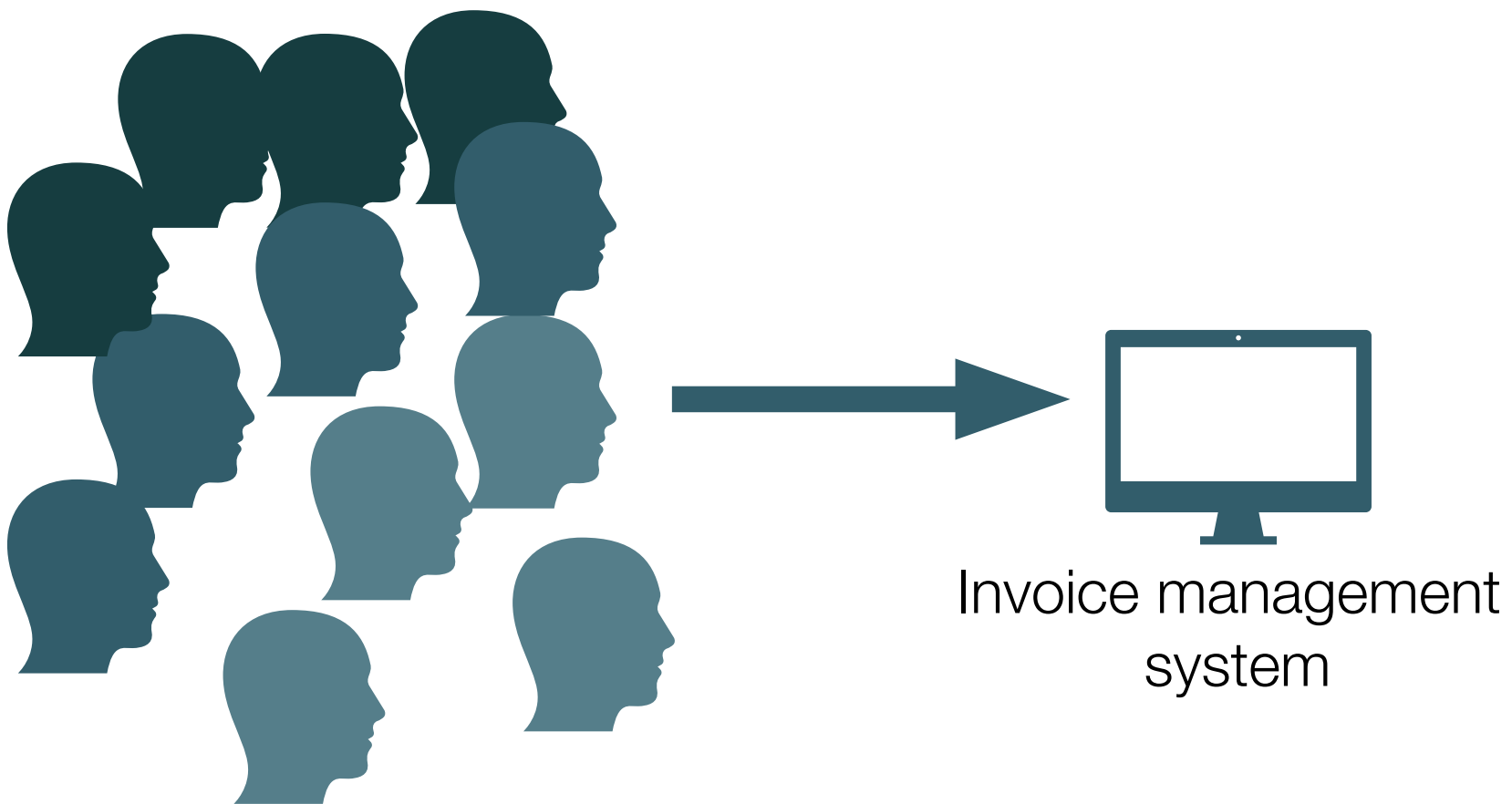
- A program was only working properly on wednesdays...
- **Documentation stated that day of the week was returned in a doubleword — 8 bytes**
 - But as Wednesday is 9 characters long, the system was expecting a three words array — 12 bytes
 - For the six other days of the week, the additional 4 bytes were programmatically filled out with blanks, as only 8 bytes were necessary
- **Unfortunately, those 4 additional bytes were also used to store another character «y», to compare with users' answers (« yes » or « no »)**
 - This value was wiped out 6 days out of 7
 - On Wednesdays, the value was stored in its correct place because it is the last character of «Wednesday»

M	O	N	D	A	Y			Y			
T	U	E	S	D	A	Y		Y			
W	E	D	N	E	S	D	A	Y			
T	H	U	R	S	D	A	Y	Y			
F	R	I	D	A	Y			Y			
S	A	T	U	R	D	A	Y	Y			
S	U	N	D	A	Y			Y			

From real experience.

The double invoice

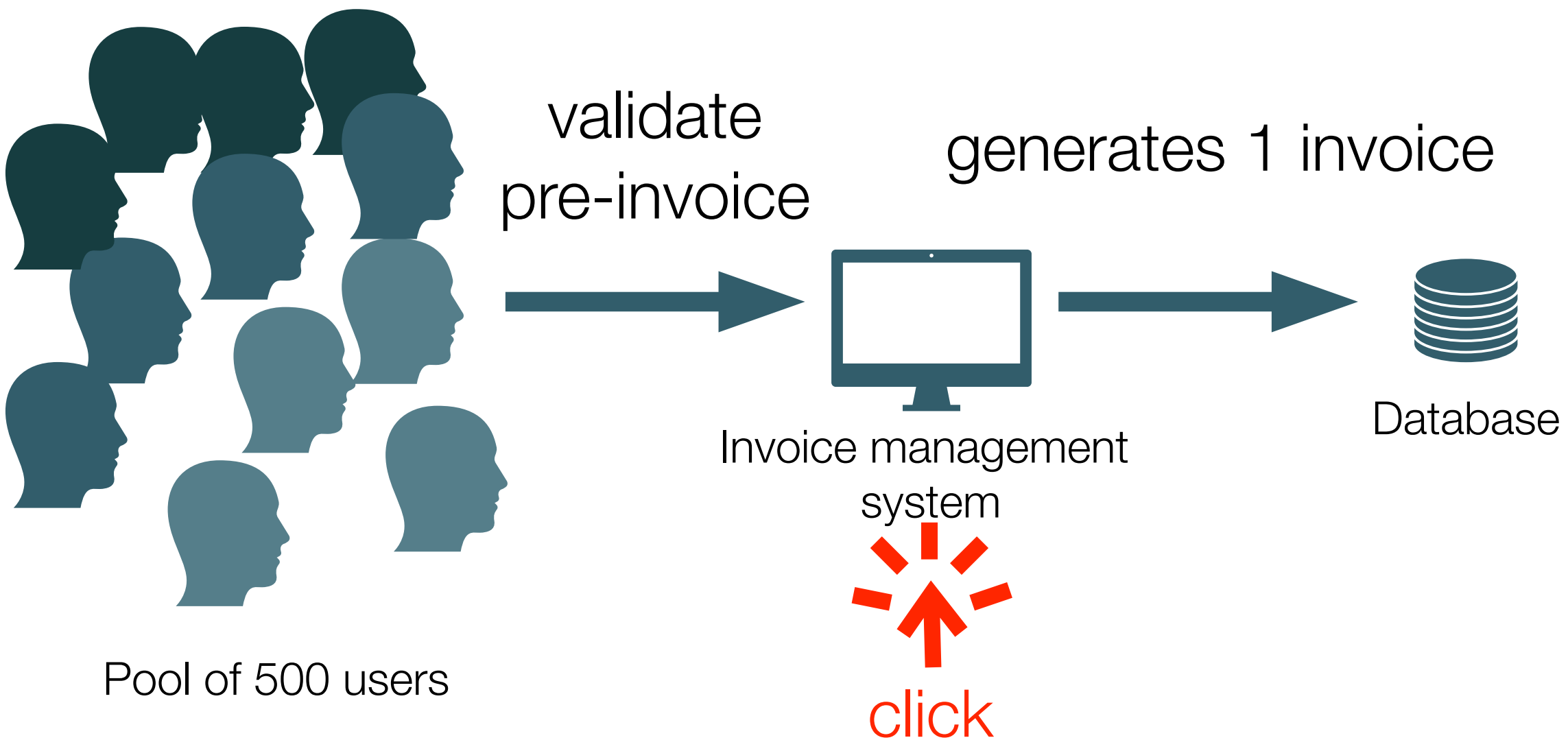
The double invoice



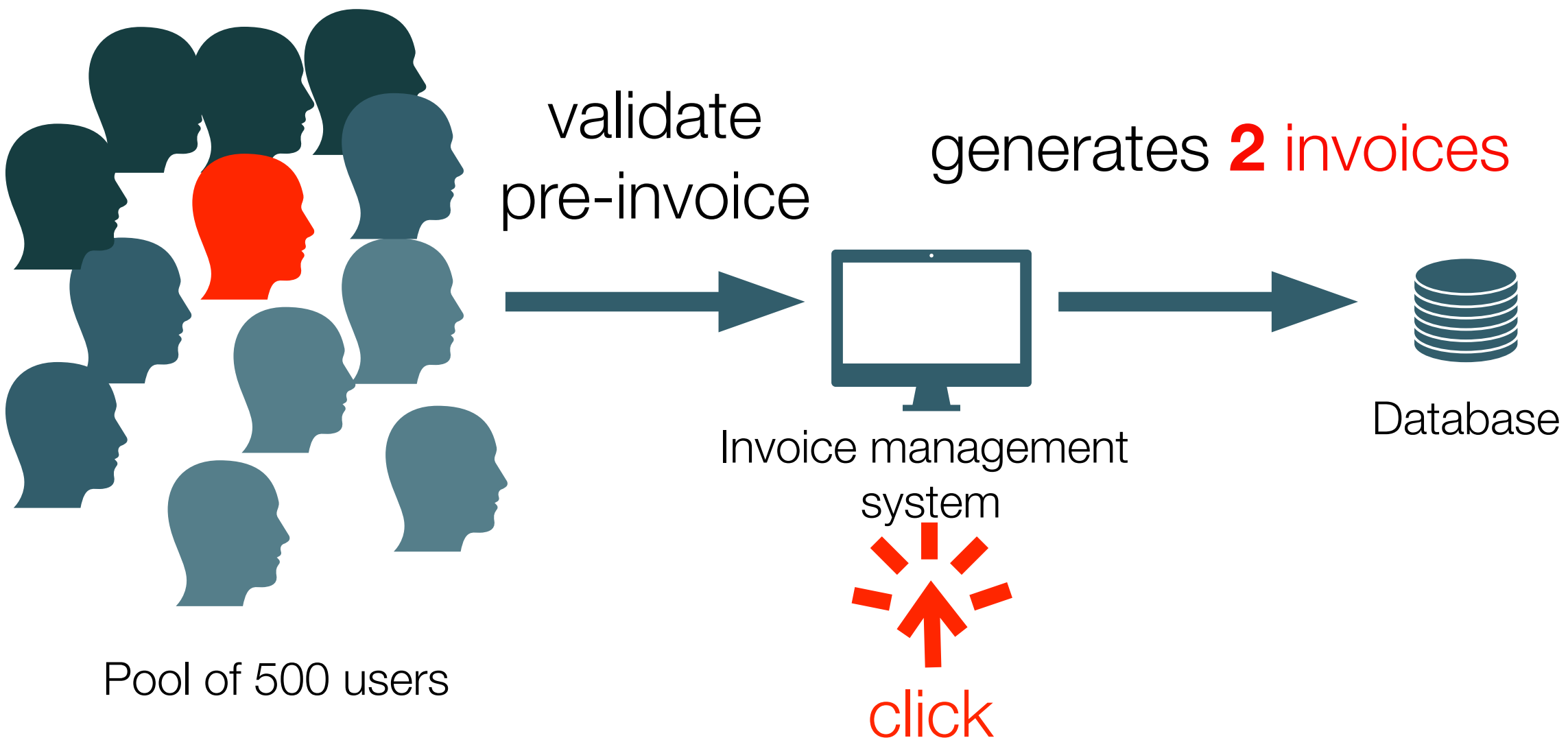
Pool of 500 users

Invoice management
system

The double invoice

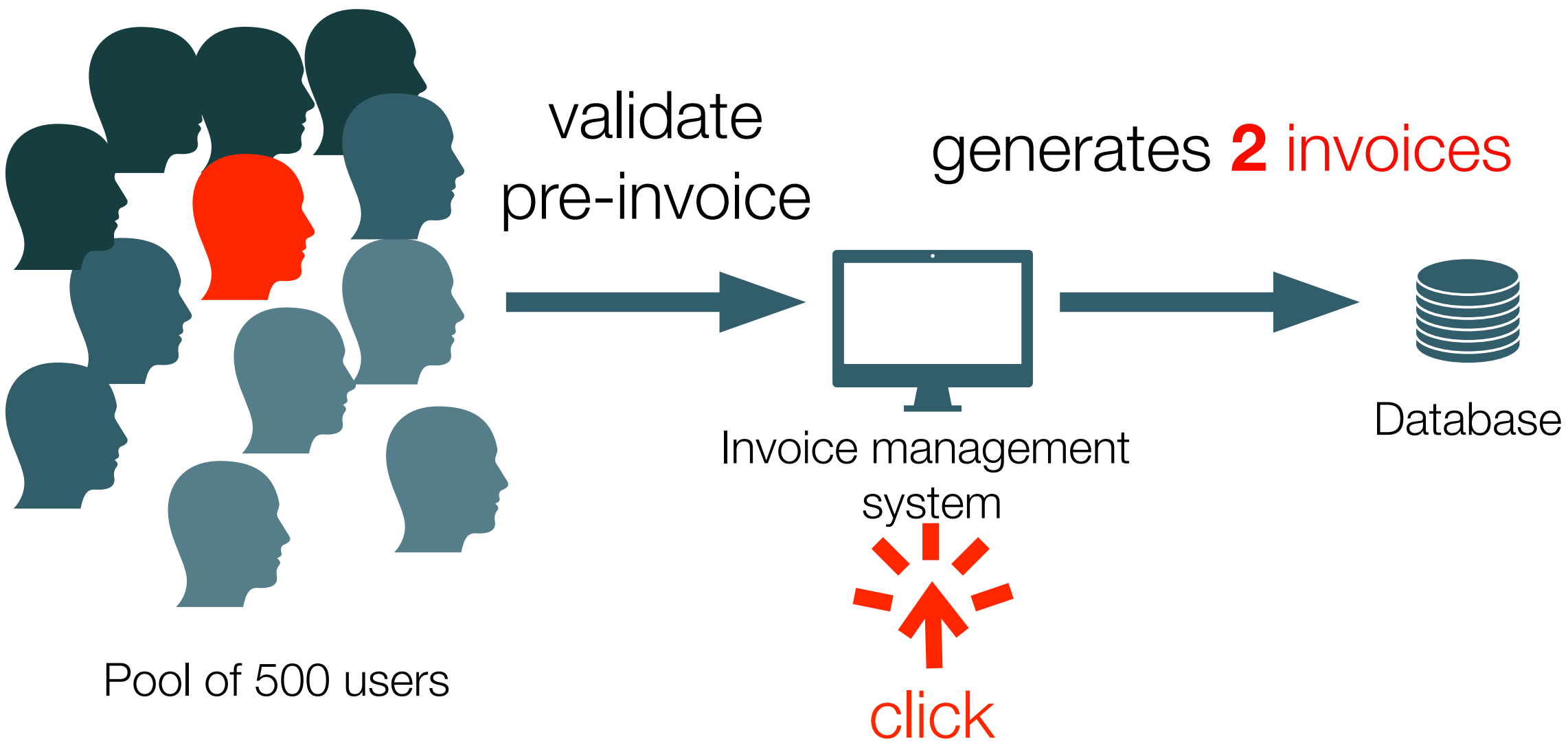


The double invoice



French Law: only one single invoice can be generated from a pre-invoice

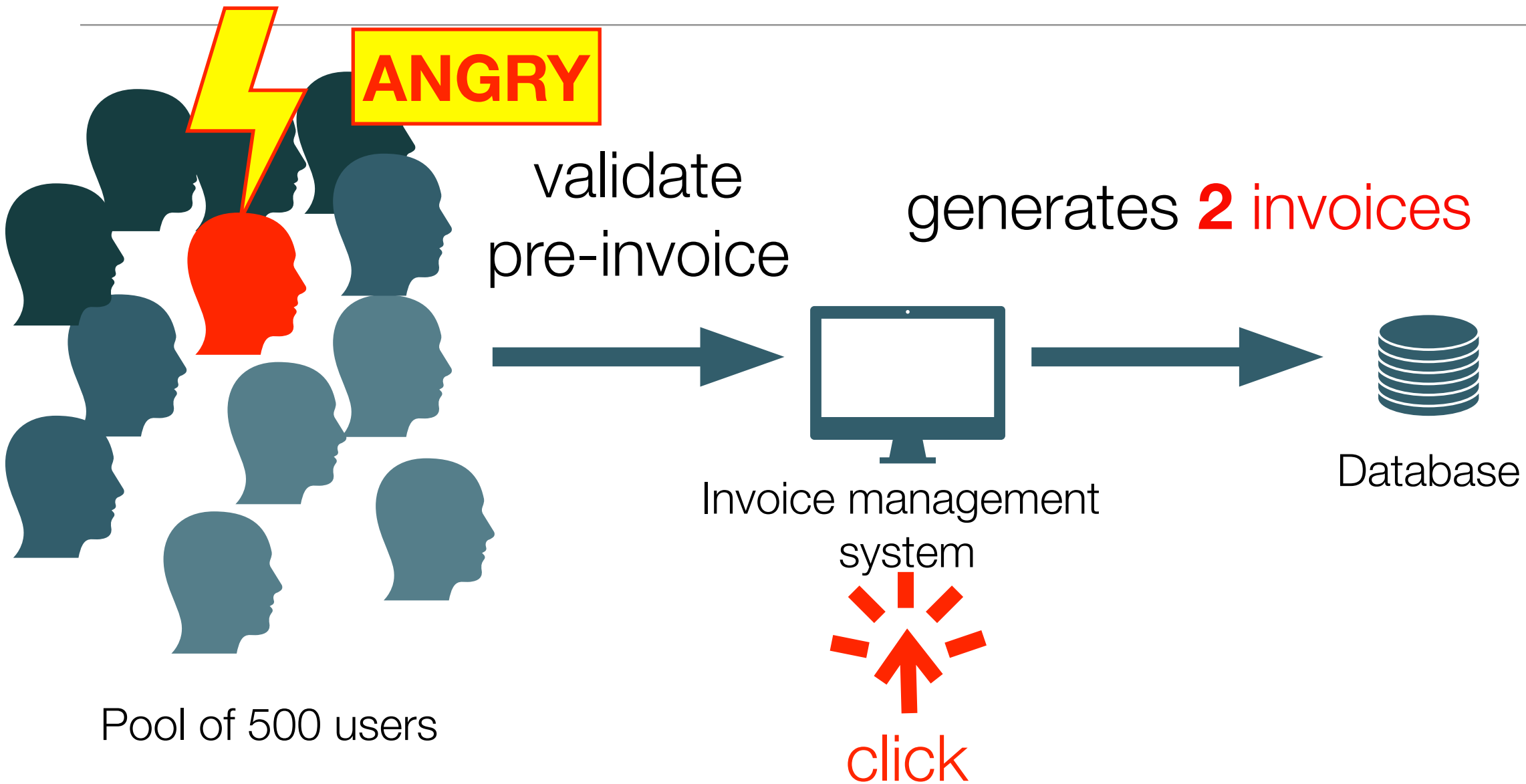
The double invoice



French Law: only one single invoice can be generated from a pre-invoice

French Law: once written in the database, you cannot modify invoices without leaving a trace in your accounts (considered as cheating)

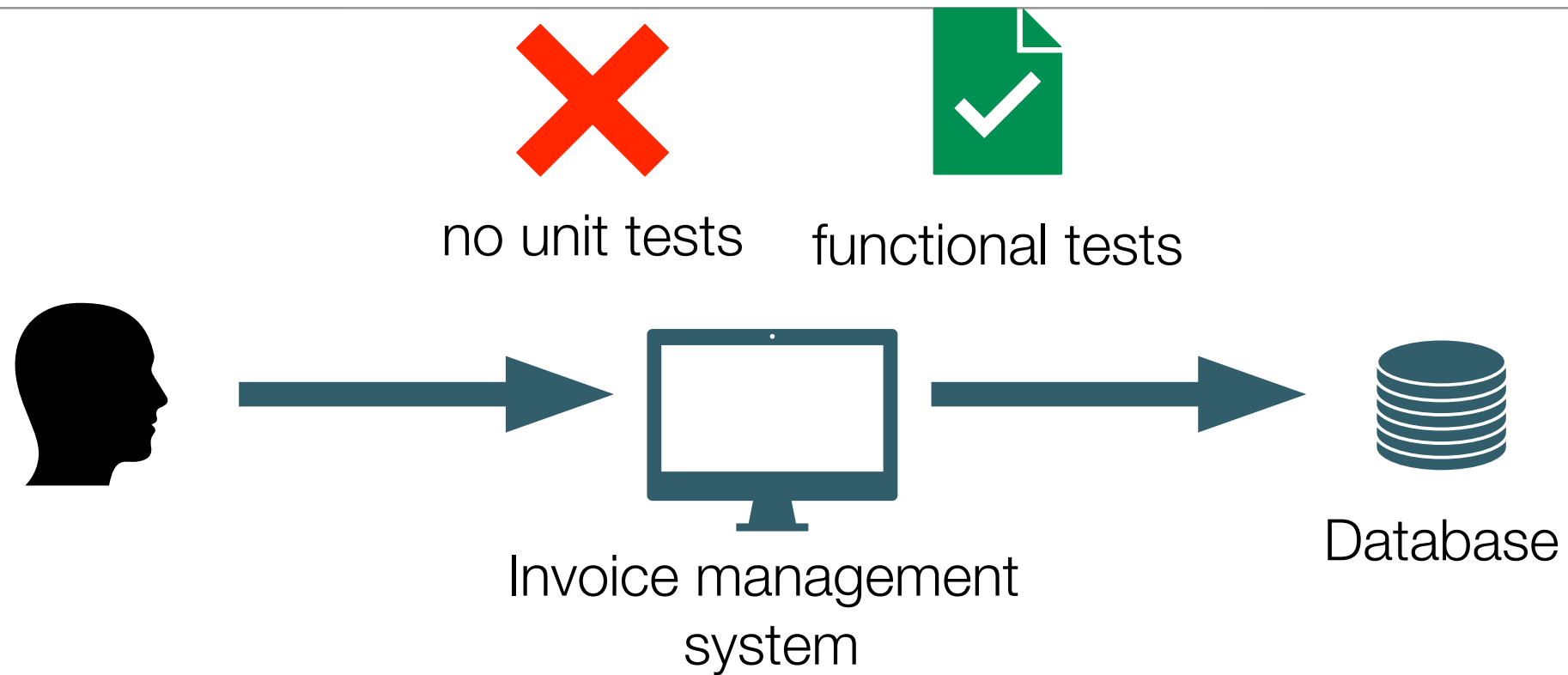
The double invoice



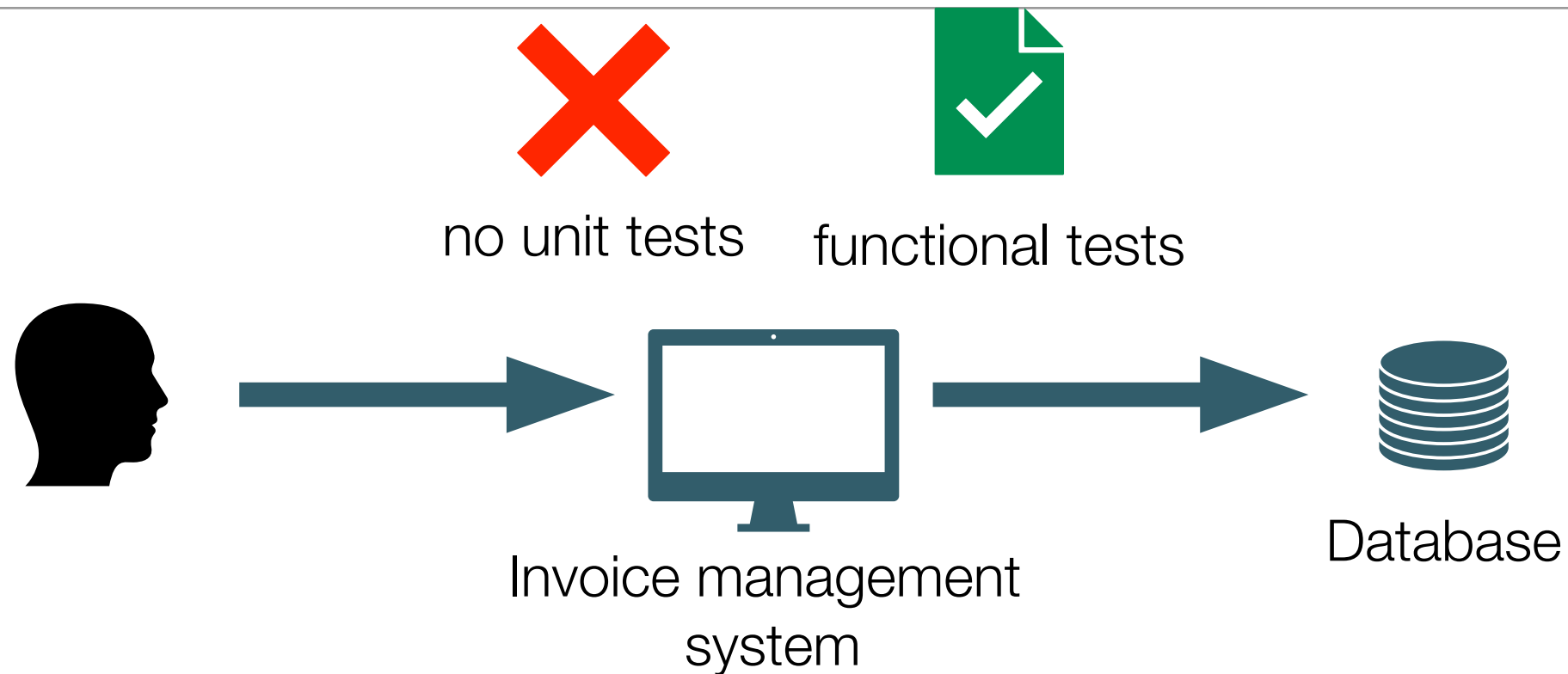
French Law: only one single invoice can be generated from a pre-invoice

French Law: once written in the database, you cannot modify invoices without leaving a trace in your accounts (considered as cheating)

The double invoice

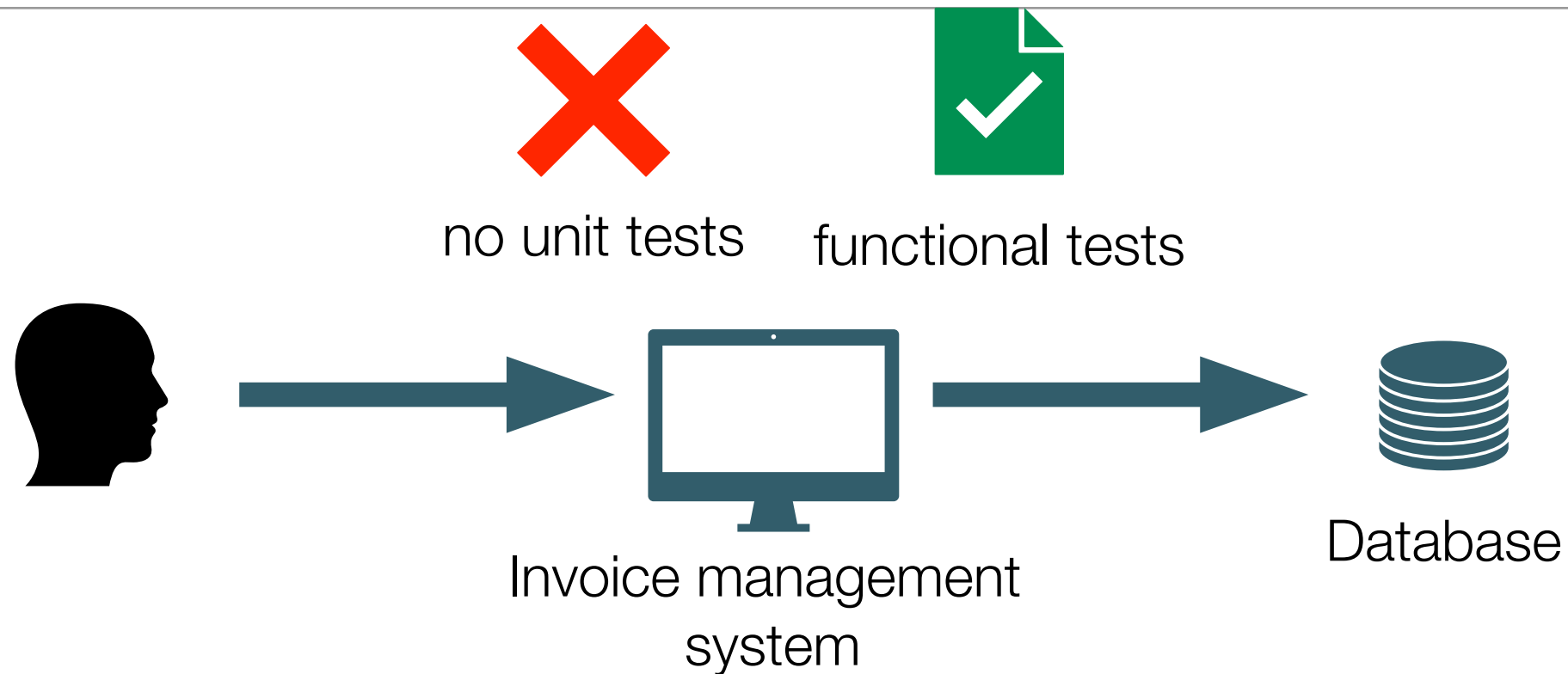


The double invoice



- We isolated the user and made him show us **remotely** how he used the system:

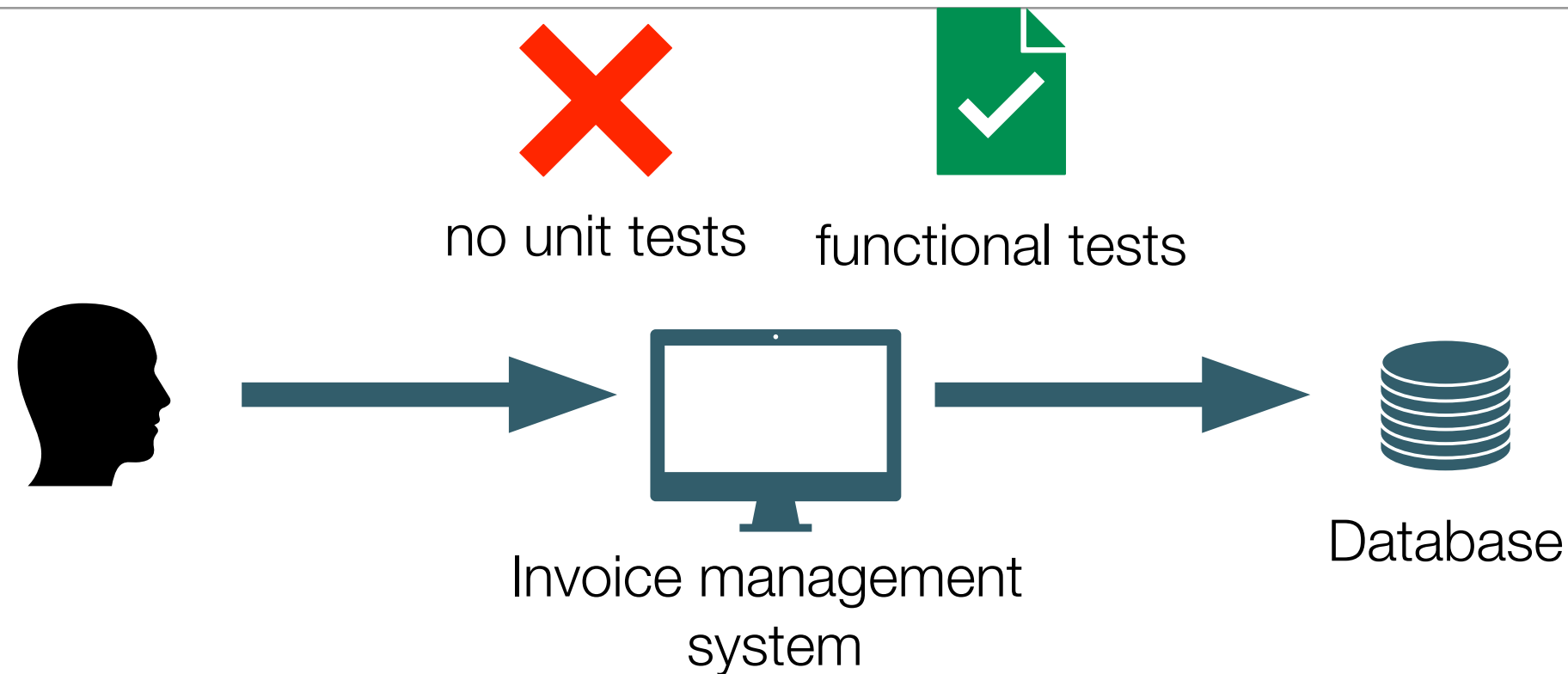
The double invoice



- We isolated the user and made him show us **remotely** how he used the system:

 everything was **always fine when we were looking!**

The double invoice



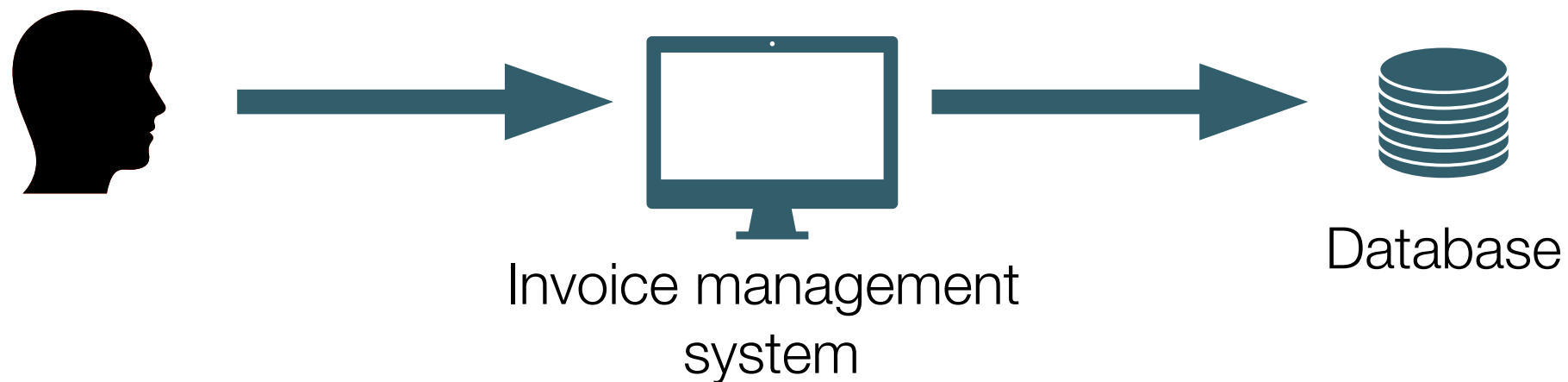
- We isolated the user and made him show us **remotely** how he used the system:

➡ everything was **always fine when we were looking!**

➡ but after, the angry **user kept phoning the hotline** because of the double invoices generation!

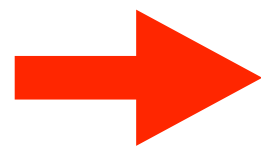
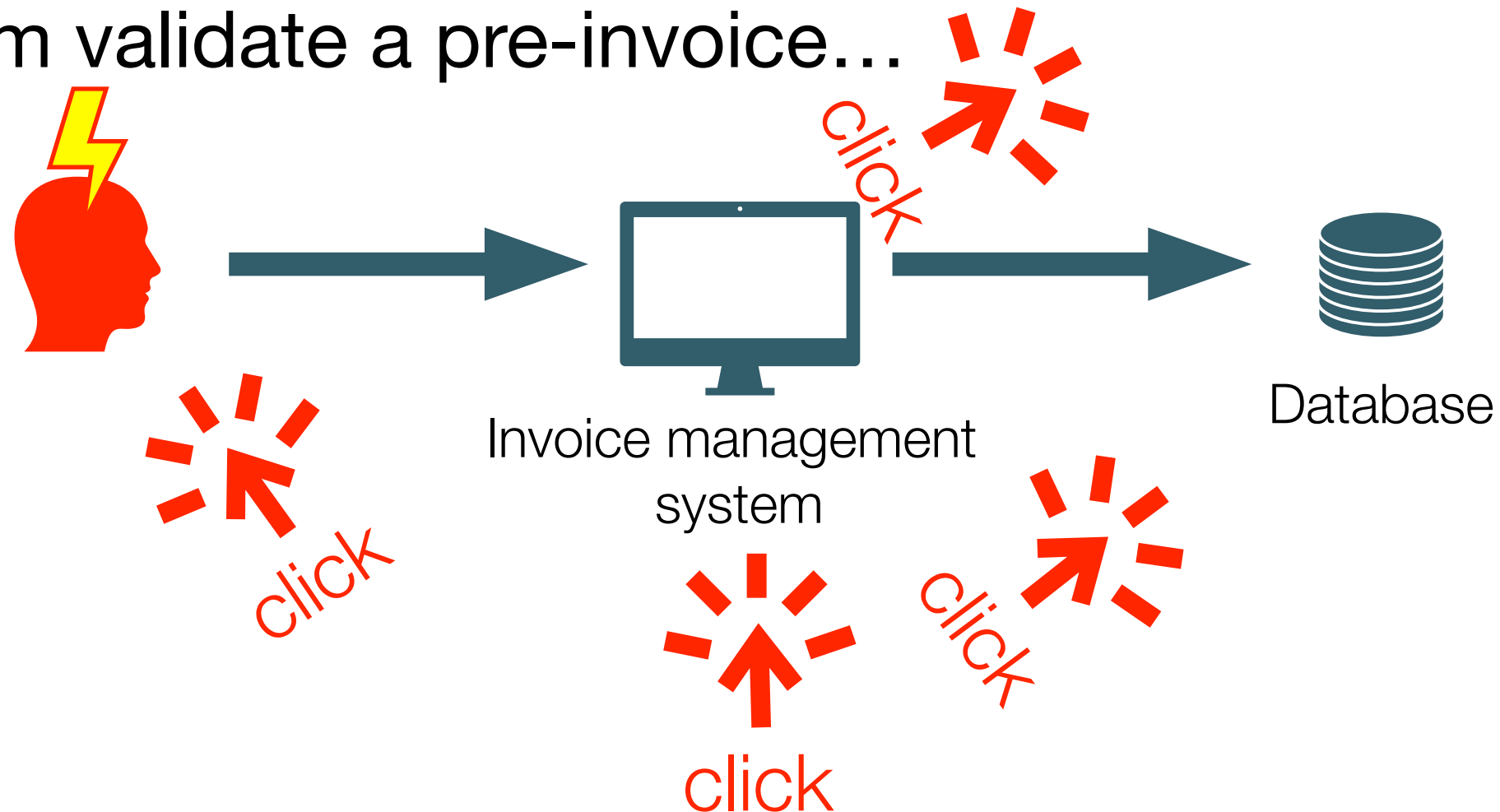
The double invoice

One day, we went to see him for something else and saw him validate a pre-invoice...



The double invoice

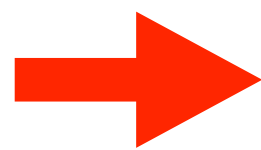
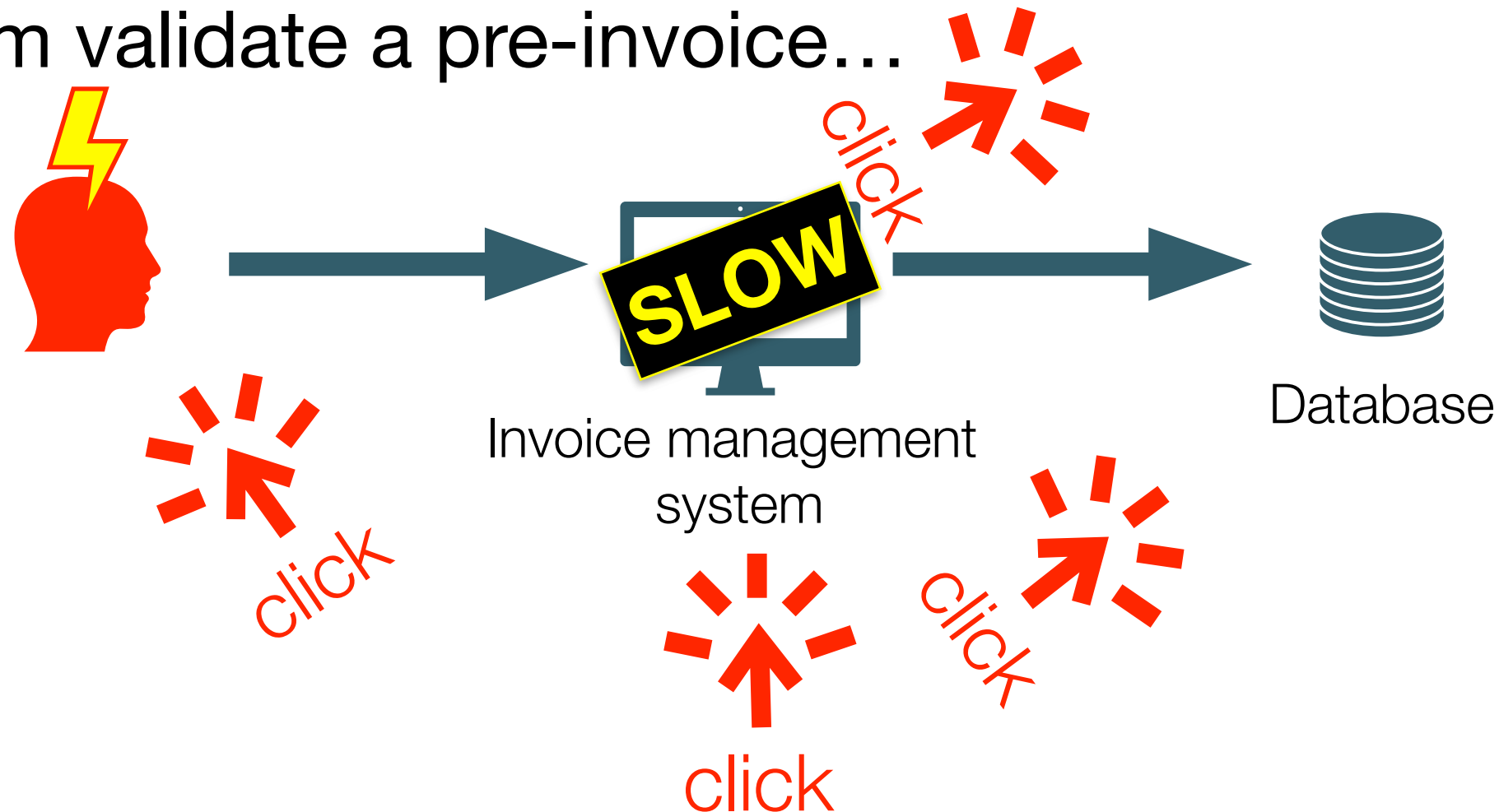
One day, we went to see him for something else and saw him validate a pre-invoice...



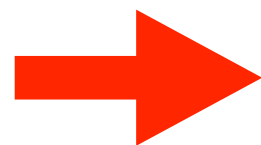
Turns out the user was angry all the time
When performing an action, he clicked madly on the GUI buttons

The double invoice

One day, we went to see him for something else and saw him validate a pre-invoice...



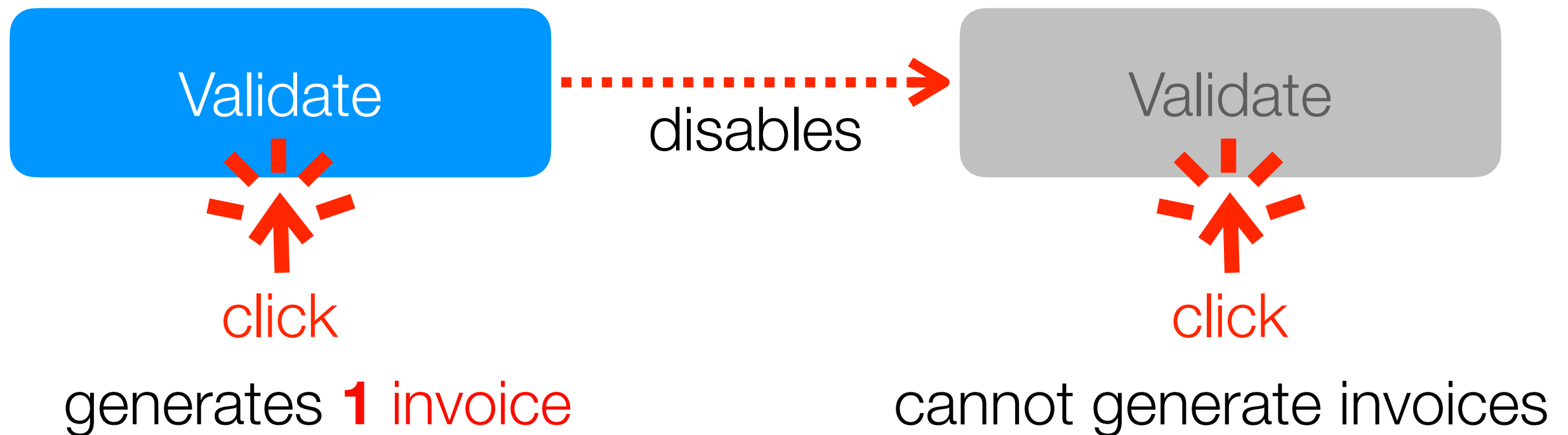
Turns out the user was angry all the time
When performing an action, he clicked madly on the GUI buttons



Turns out his computer was so slow that he managed to click two times in a row on the « validate » button

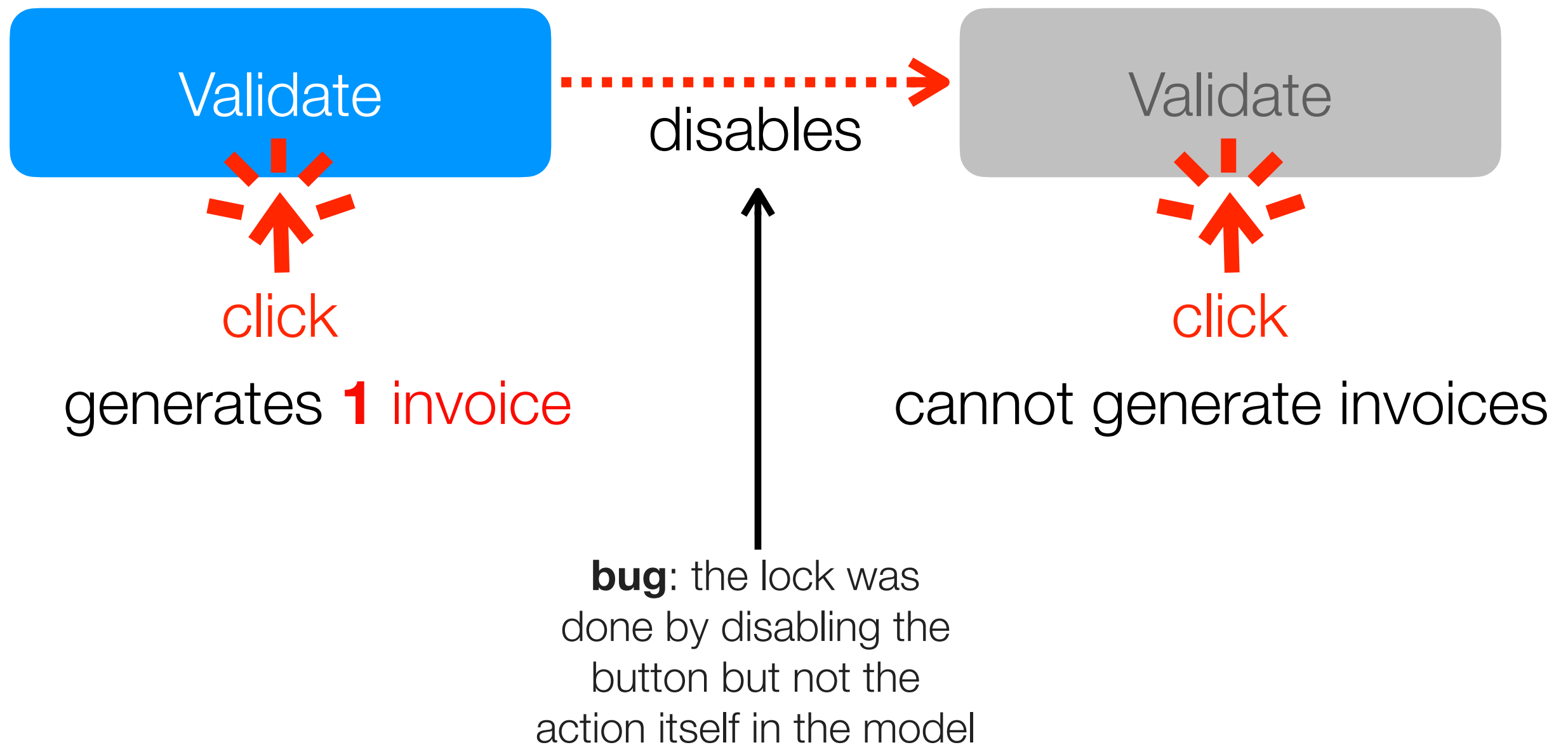
The double invoice

The bug explained:



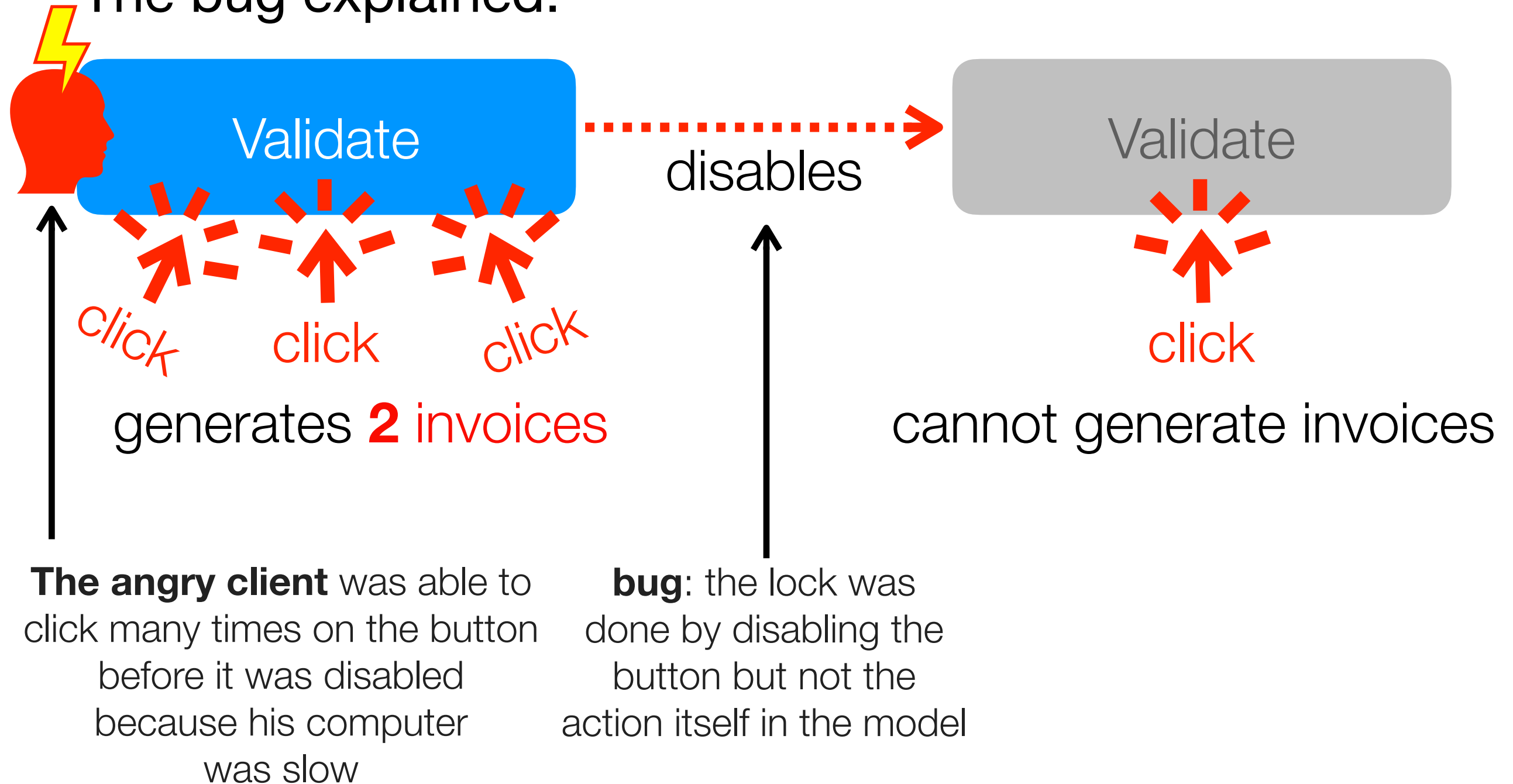
The double invoice

The bug explained:



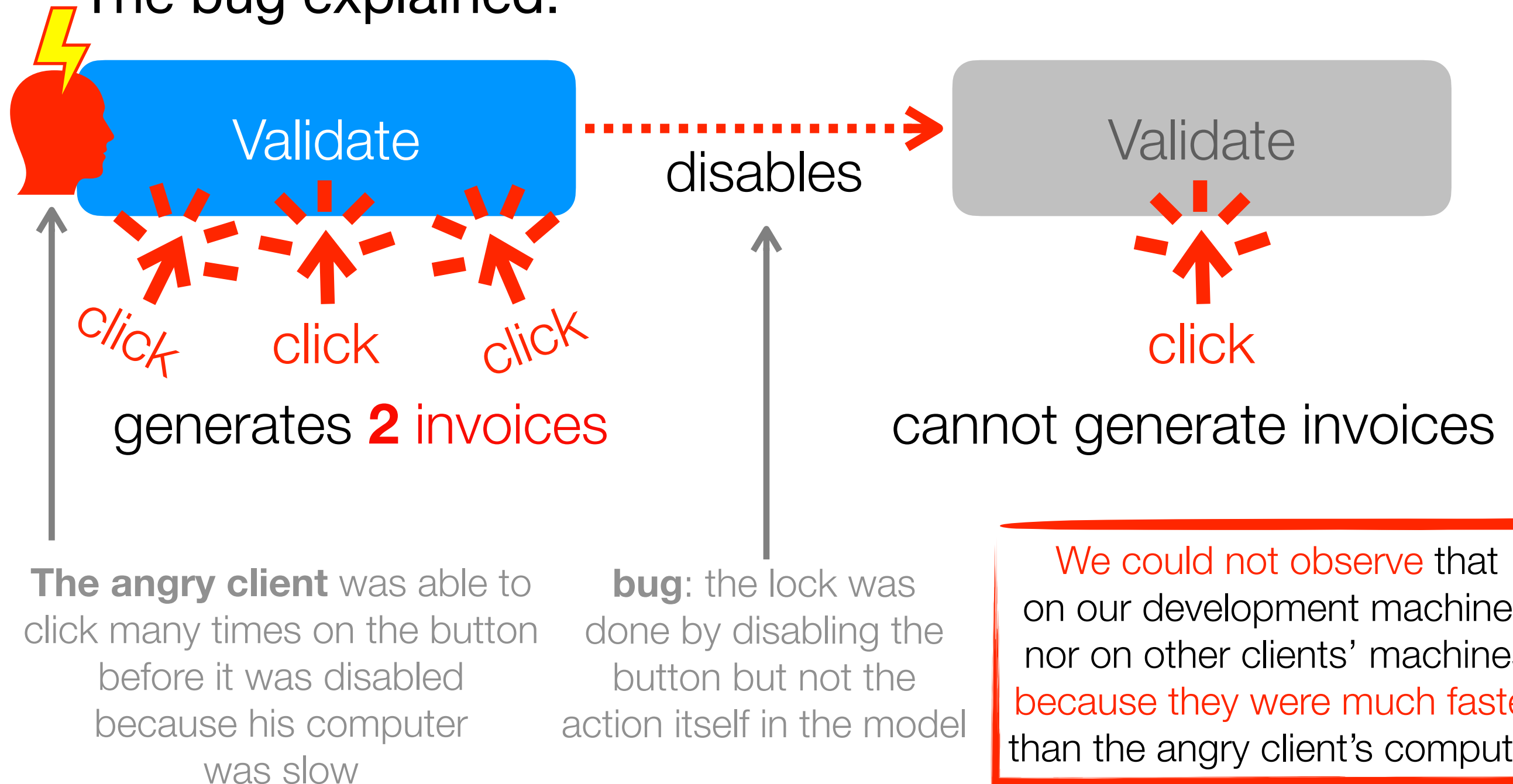
The double invoice

The bug explained:



The double invoice

The bug explained:



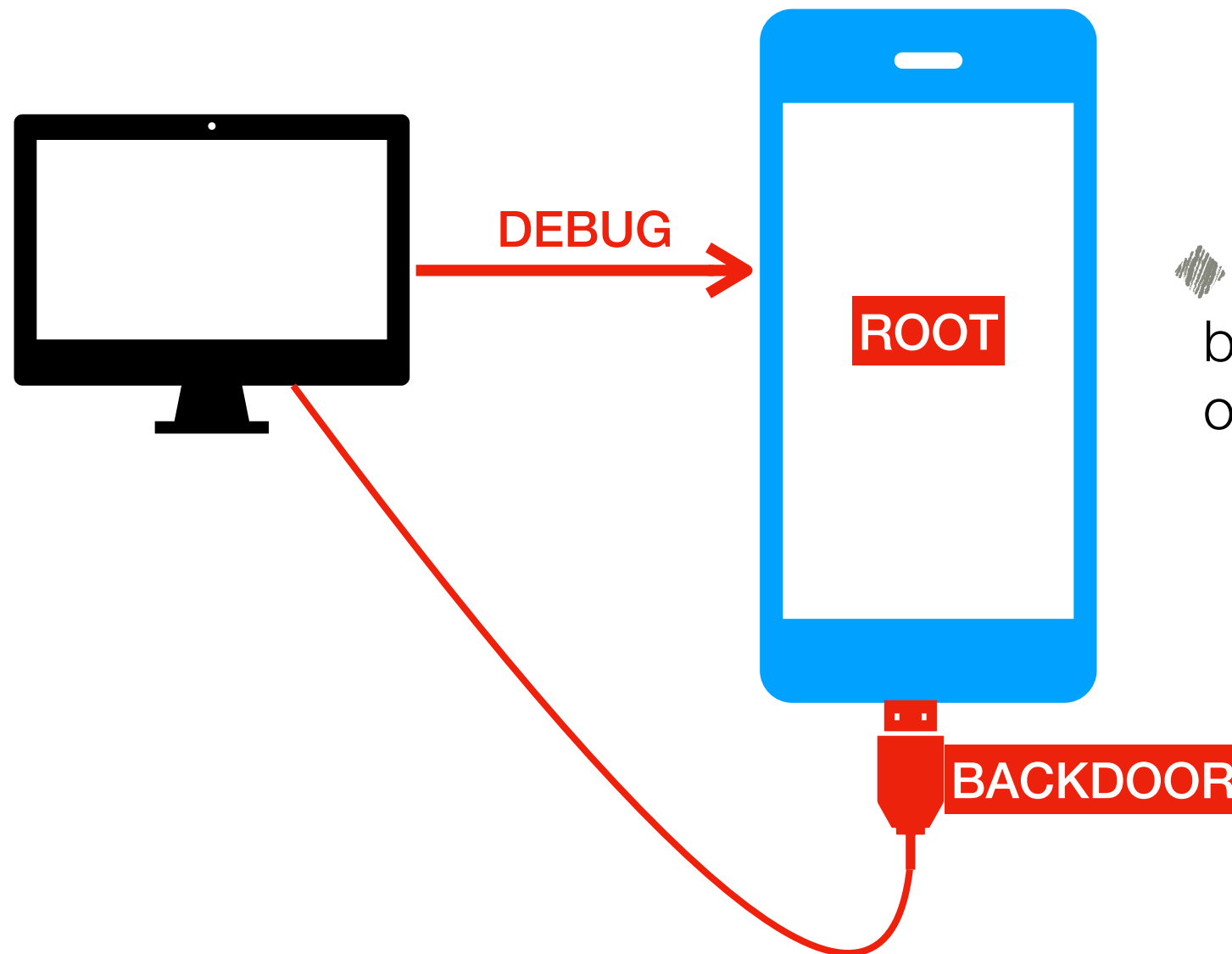
The Android

« root-console »

problem

The Android « root-console » bug

Why Programs Fail, Andreas Zeller, 2009

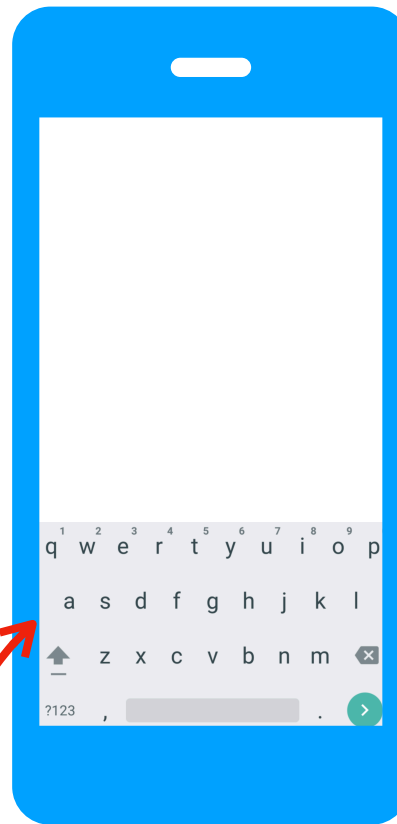


Google engineers implemented a backdoor to connect remote devices over the serial port for debugging

The Android « root-console » bug

Why Programs Fail, Andreas Zeller, 2009

However, when there was no device connected, the backdoor would listen to the system keyboard...



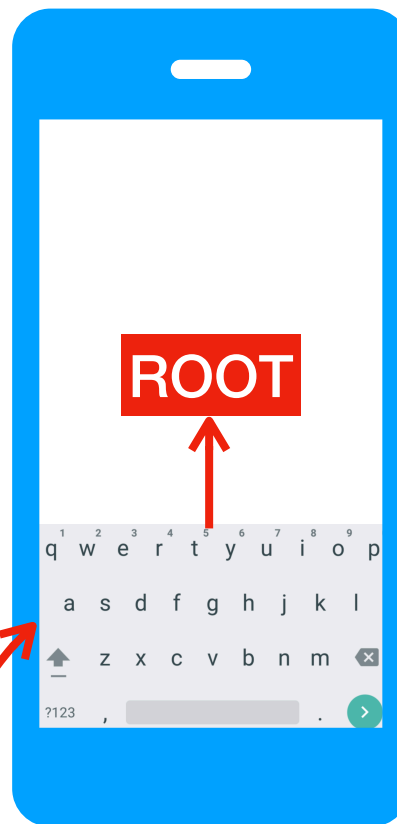
BACKDOOR

The Android « root-console » bug

Why Programs Fail, Andreas Zeller, 2009

However, when there was no device connected, the backdoor would listen to the system keyboard...

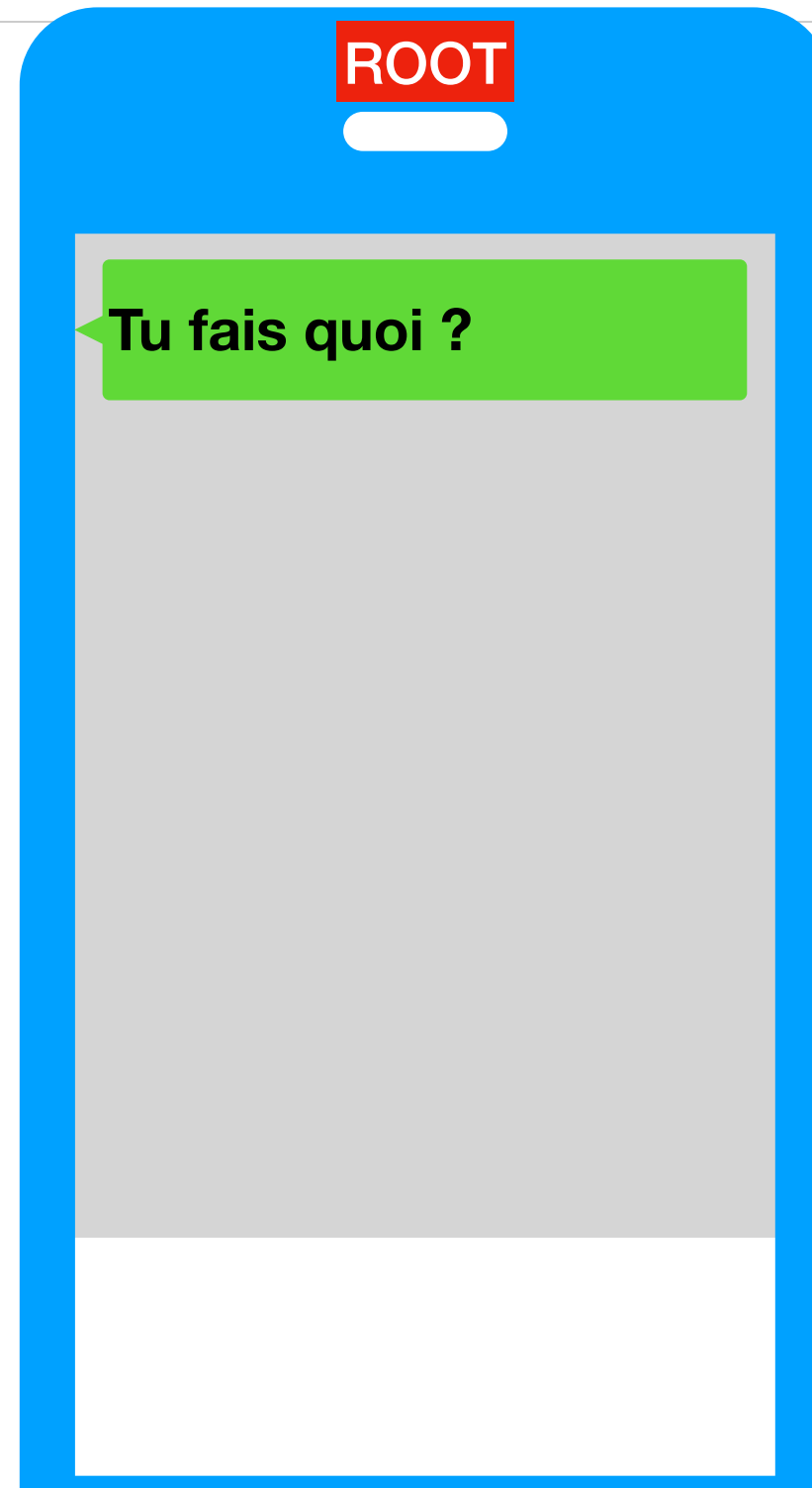
BACKDOOR



Any text that people typed was then executed as a command with superuser privileges...

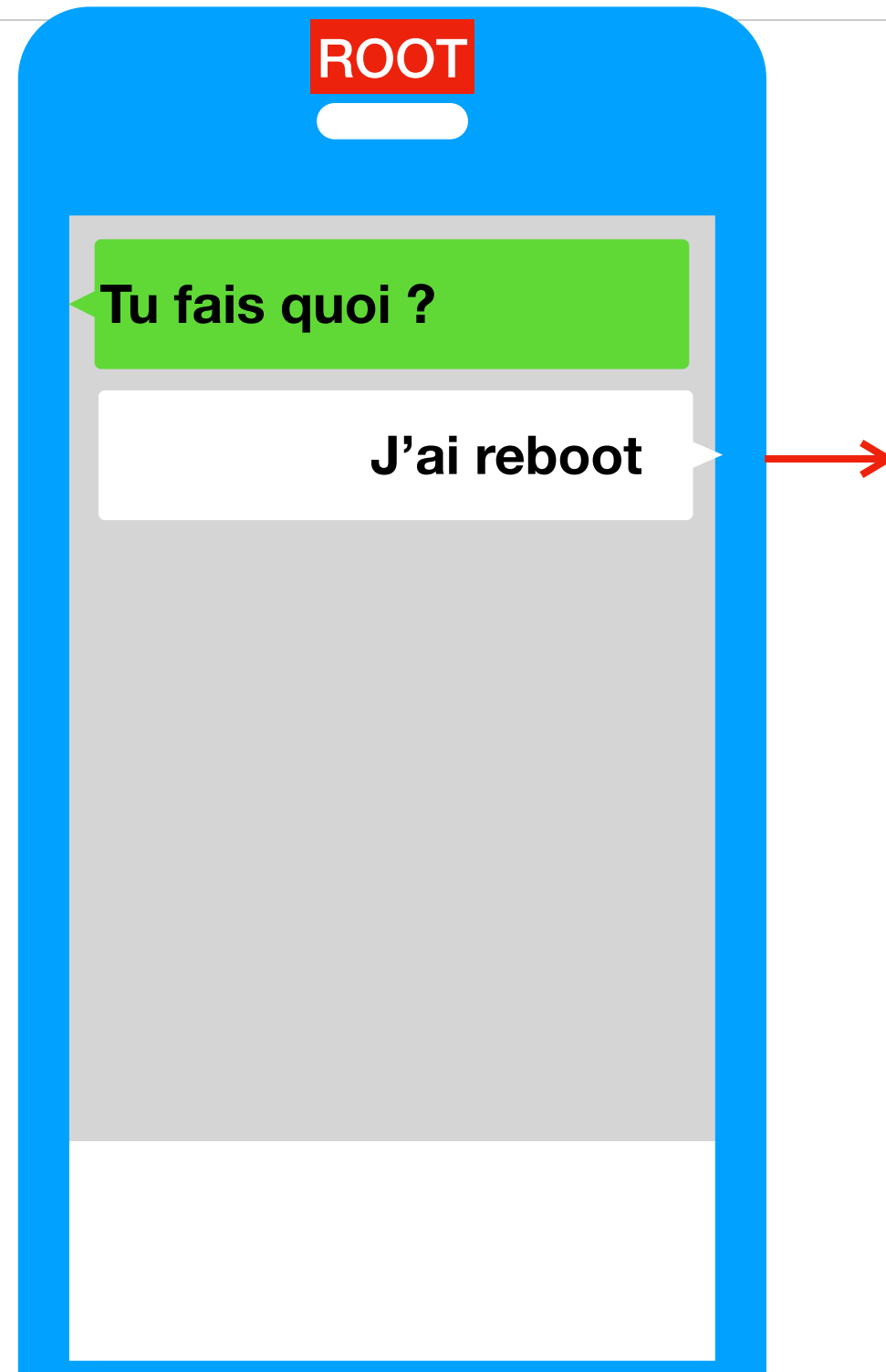
The Android « root-console » bug

Why Programs Fail, Andreas Zeller, 2009



The Android « root-console » bug

Why Programs Fail, Andreas Zeller, 2009



Redémarre le
téléphone

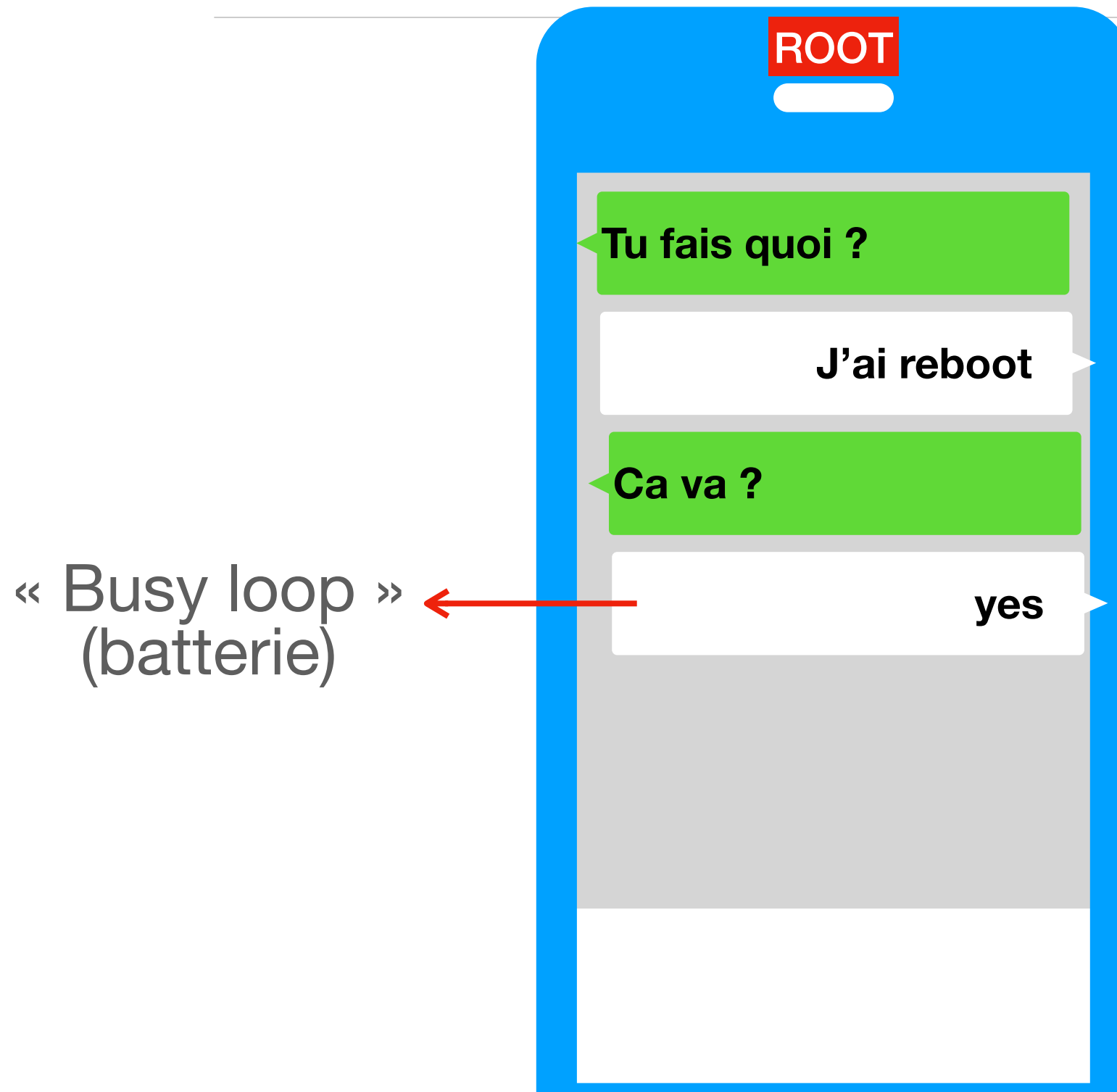
The Android « root-console » bug

Why Programs Fail, Andreas Zeller, 2009



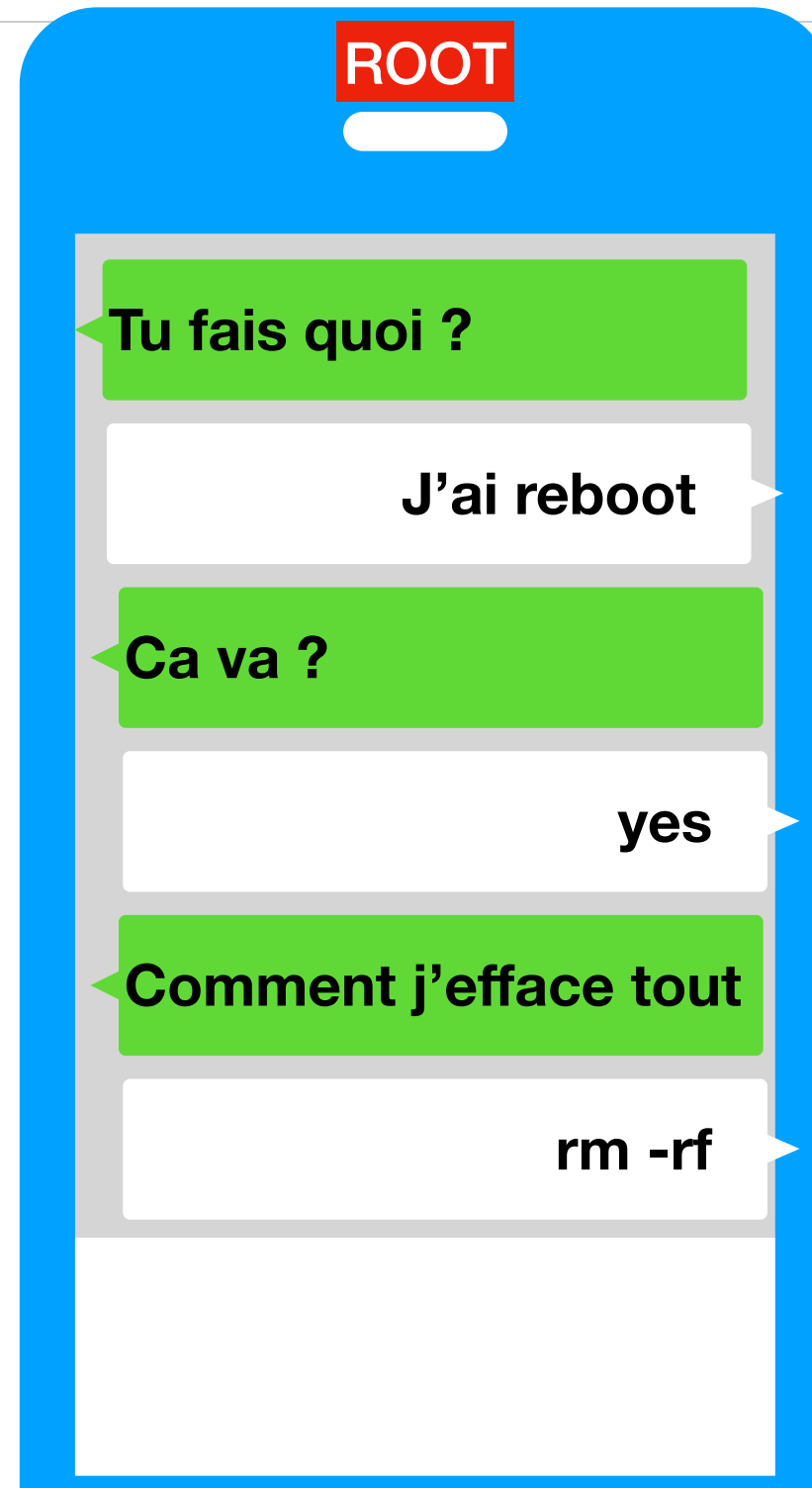
The Android « root-console » bug

Why Programs Fail, Andreas Zeller, 2009



The Android « root-console » bug

Why Programs Fail, Andreas Zeller, 2009



The Android « root-console » bug

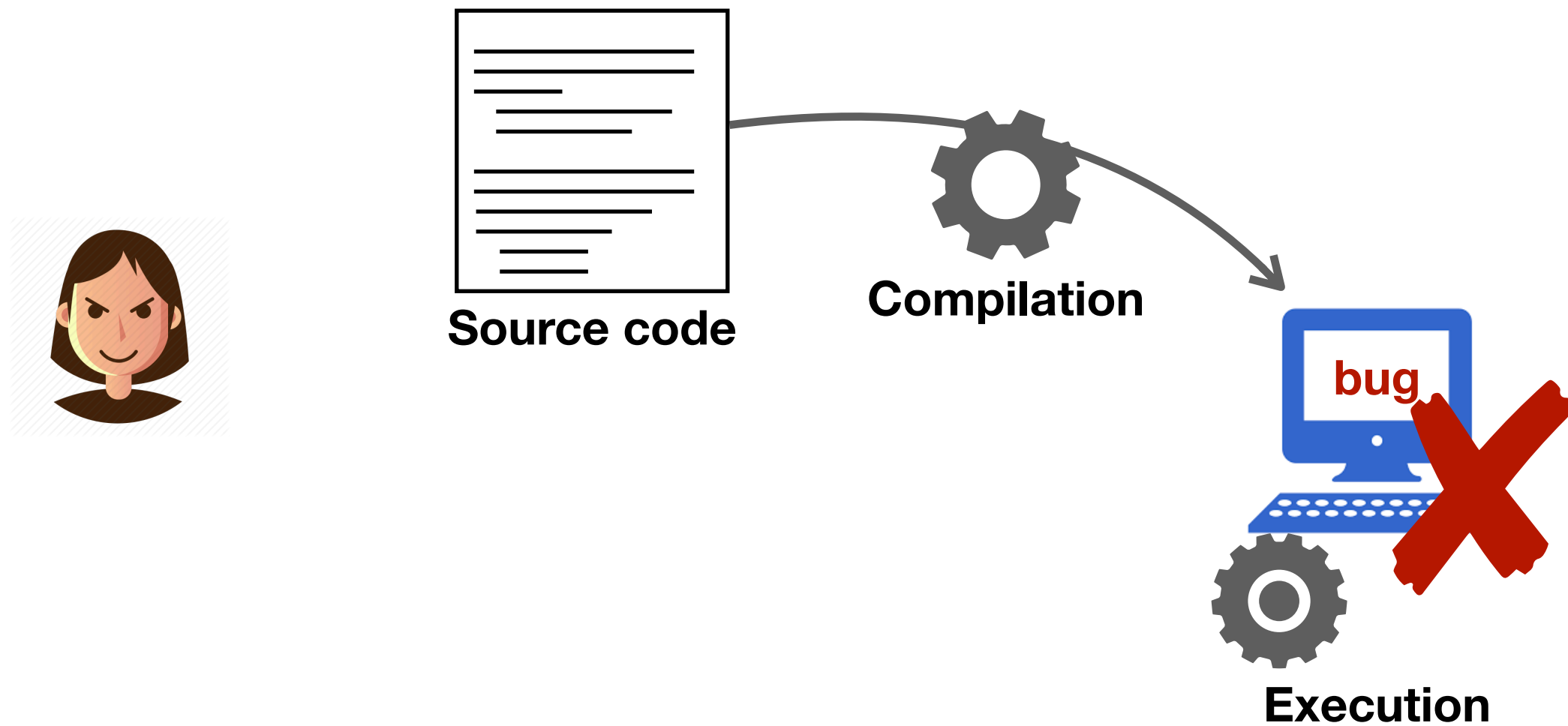
Why Programs Fail, Andreas Zeller, 2009



Tool interference

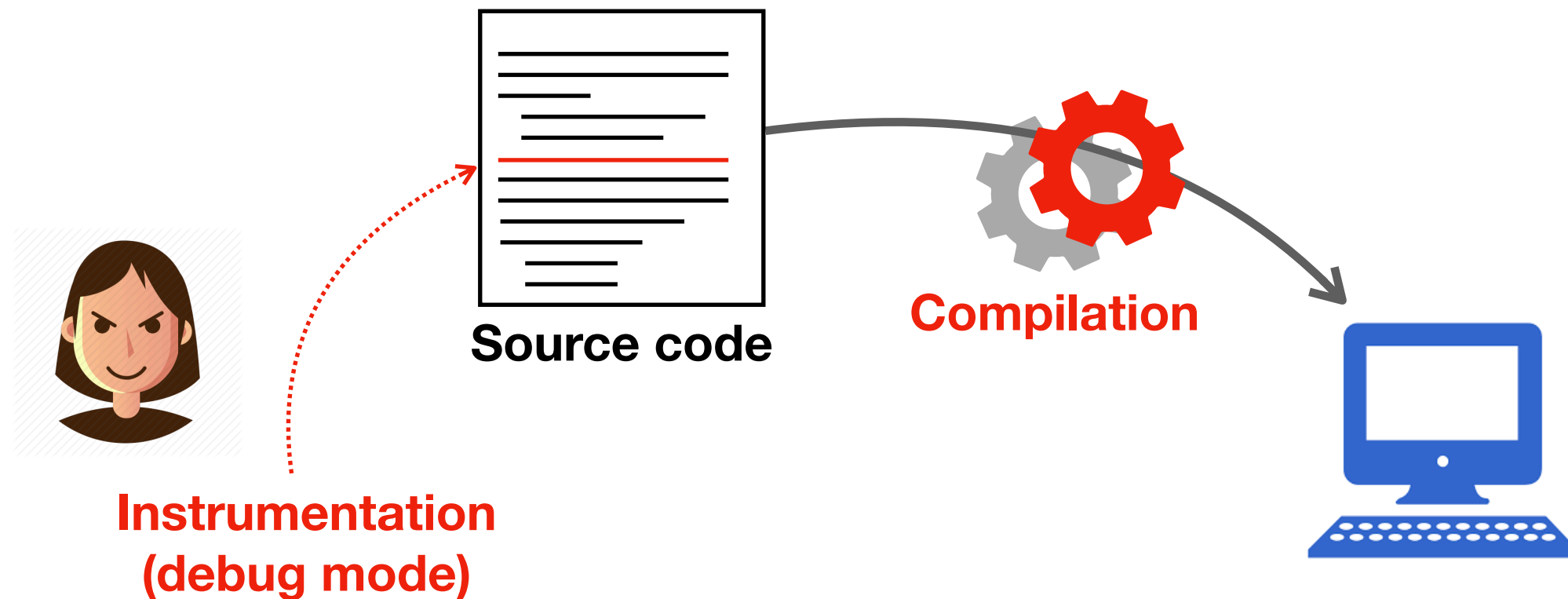
Tool interference

Why Programs Fail, Andreas Zeller, 2009



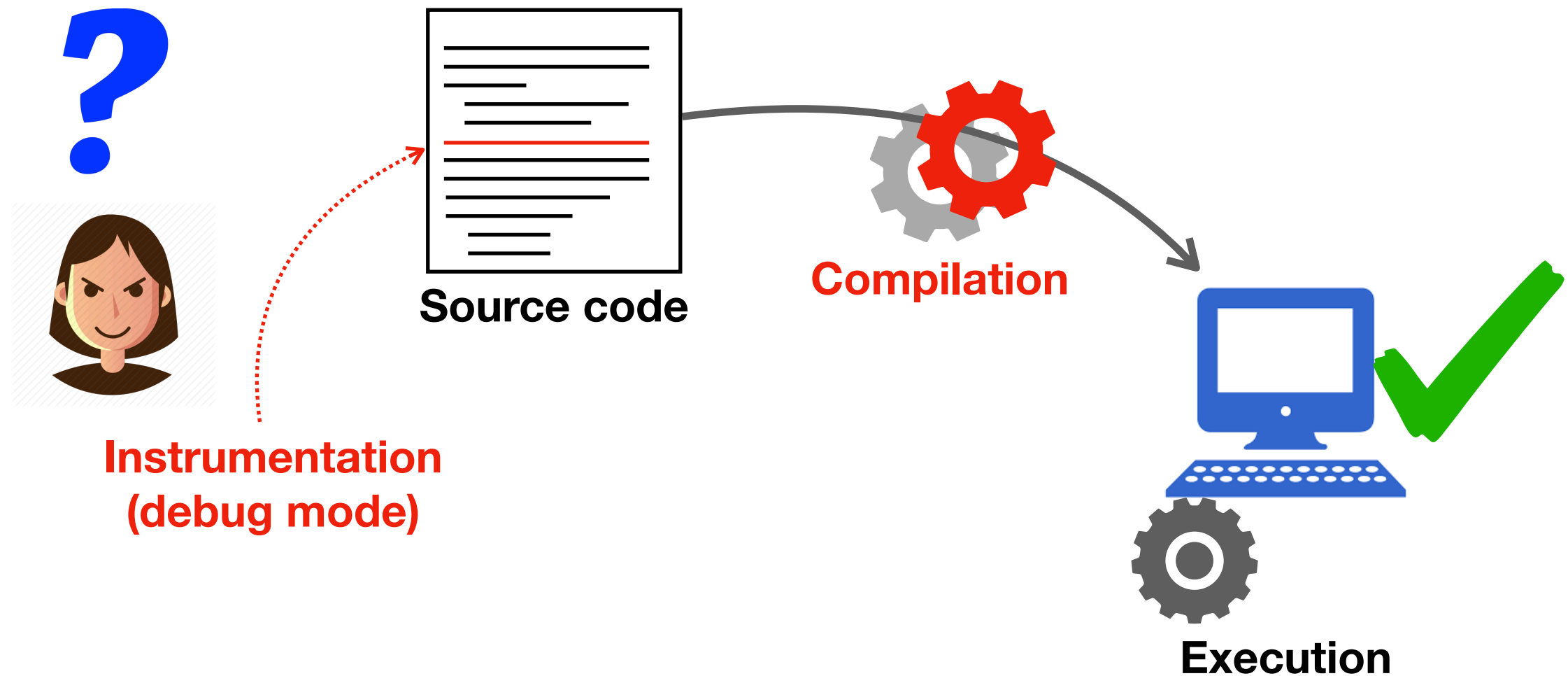
Tool interference

Why Programs Fail, Andreas Zeller, 2009



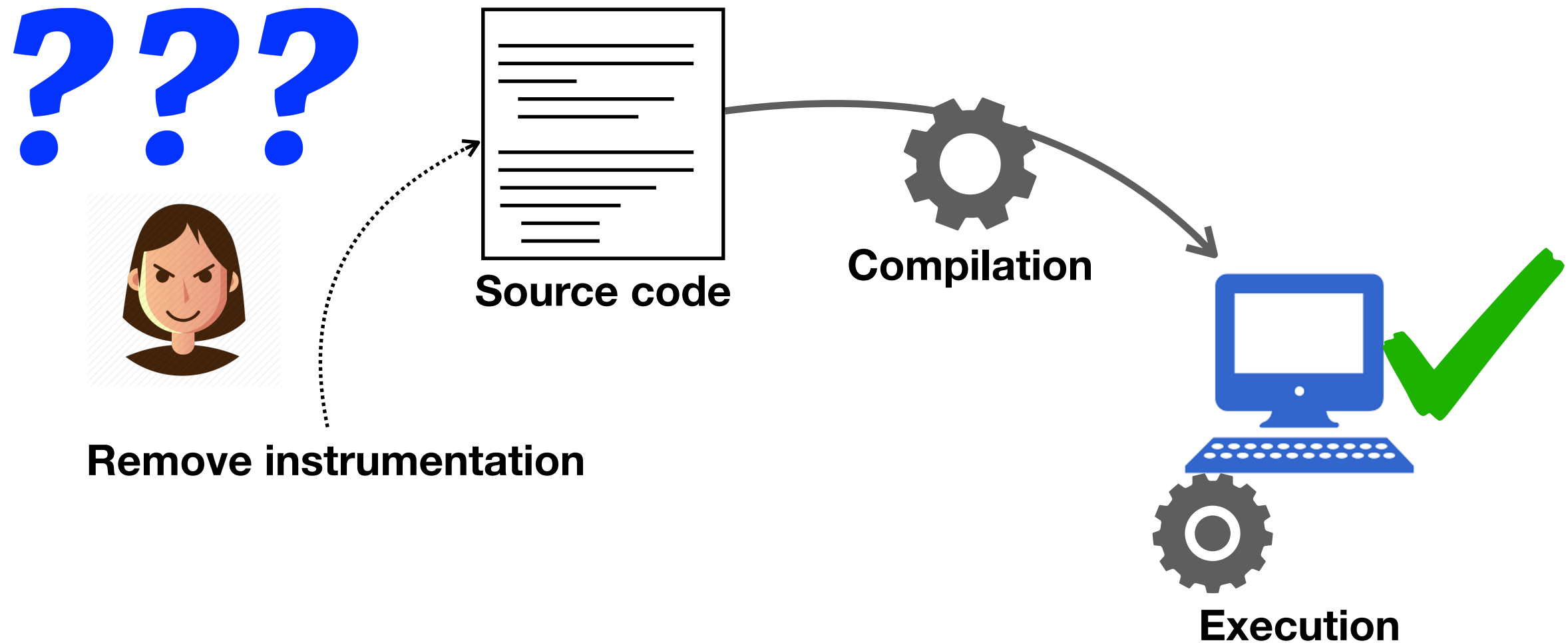
Tool interference

Why Programs Fail, Andreas Zeller, 2009



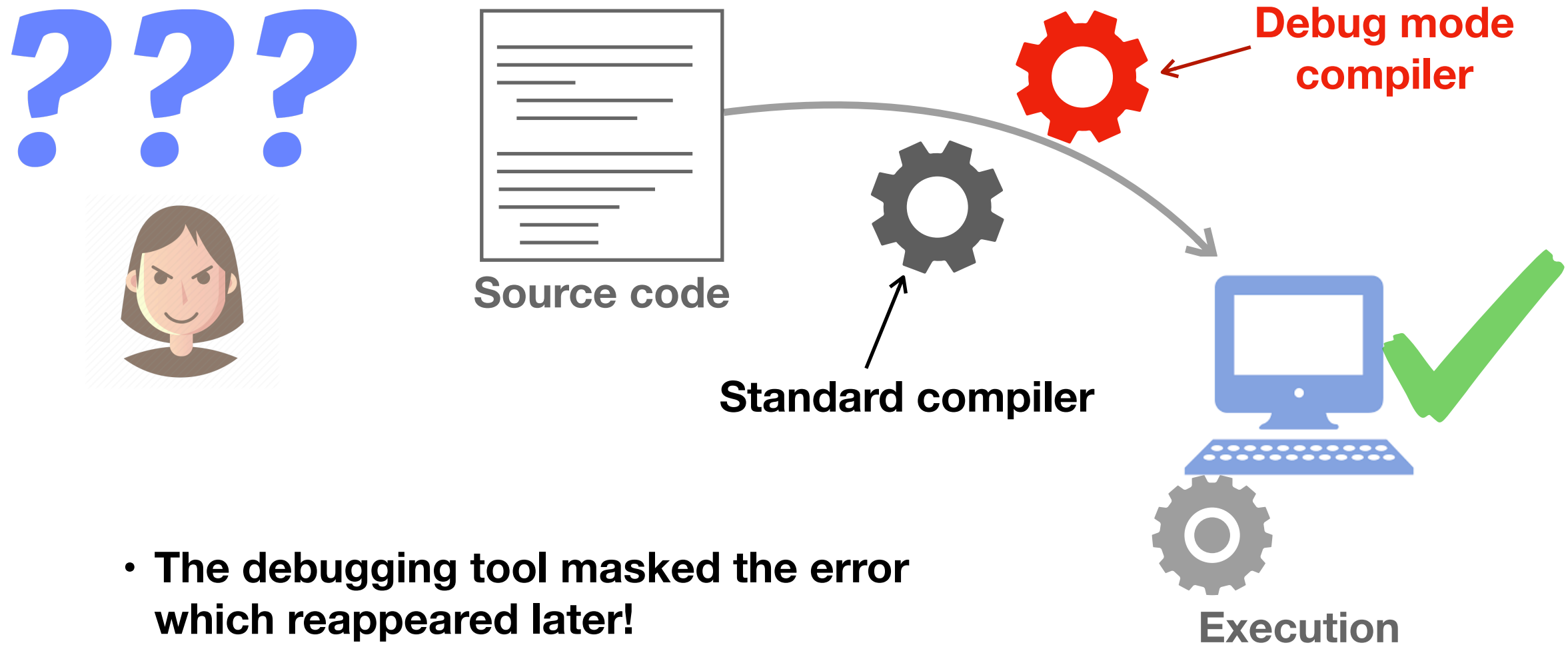
Tool interference

Why Programs Fail, Andreas Zeller, 2009



Tool interference

Why Programs Fail, Andreas Zeller, 2009



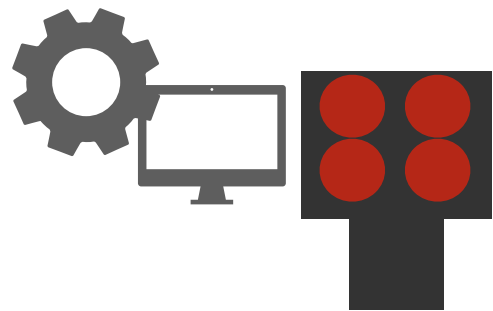
The patriot interception system bug

**Fighting bugs: Remove, retry, replicate,
and rejuvenate, Grotte and Trivedi, 2007**

The patriot interception system bug

The Patriot missile-defense system bug (1991):

Patriot system



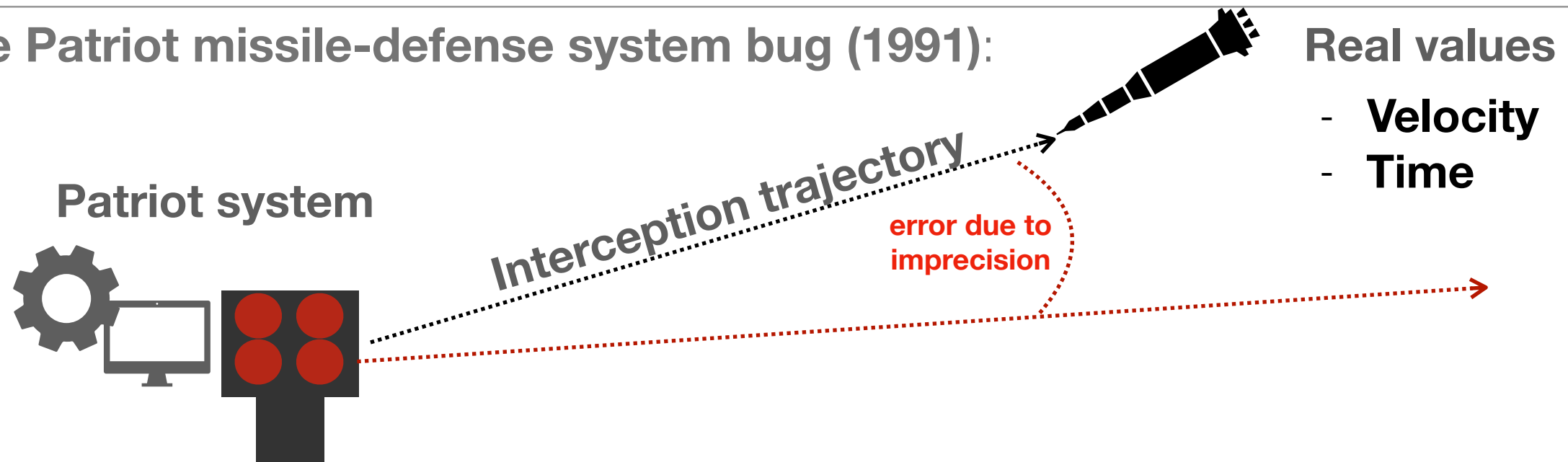
Interception trajectory

Real values

- **Velocity**
- **Time**

The patriot interception system bug

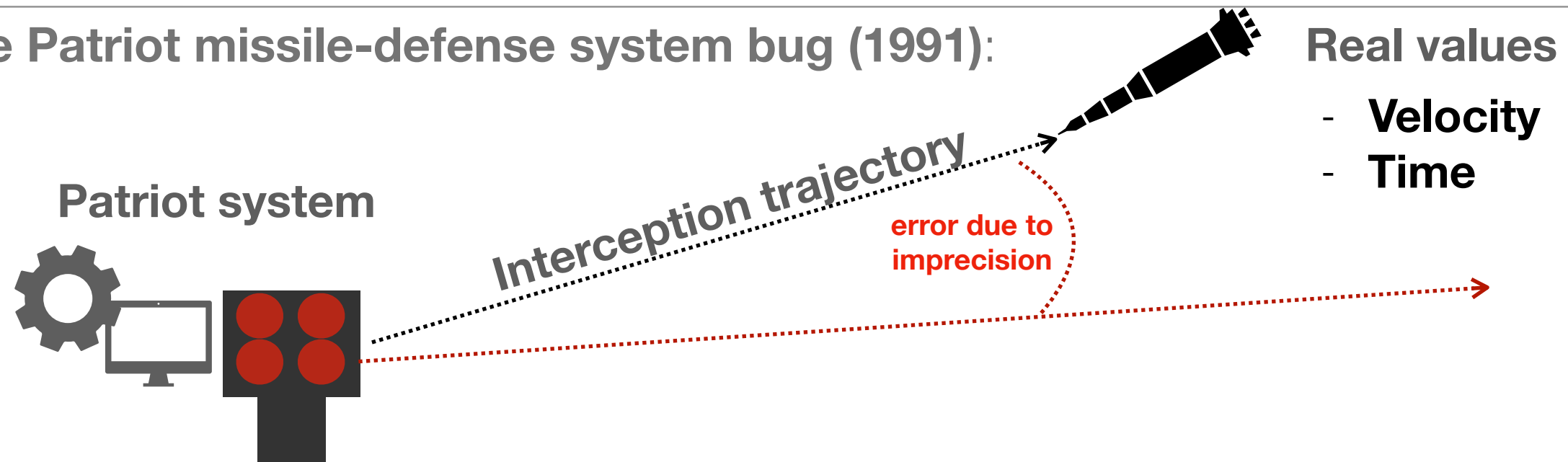
The Patriot missile-defense system bug (1991):



- Time internally represented as an integer!
- Conversion to a real value was imprecise
- Imprecision augmented with time!

The patriot interception system bug

The Patriot missile-defense system bug (1991):



- Time internally represented as an integer!
- Conversion to a real value was imprecise
- Imprecision augmented with time!

1991, Dharhan, Saudi Arabia: after 20 hours of run time, the system missed an interception because of this bug (28 dead, 97 wounded)

The Therac-25 bugs

The Therac-25 bug (one of them...)

**Radiation
therapy machine
used in the 80's**

**Entirely
controlled
by software**



**Bugs:
radiation
overdose of
6 patients, 3
died**

The Therac-25 bug (one of them...)

Error counter:
0 = SAFE



The Therac-25 bug (one of them...)

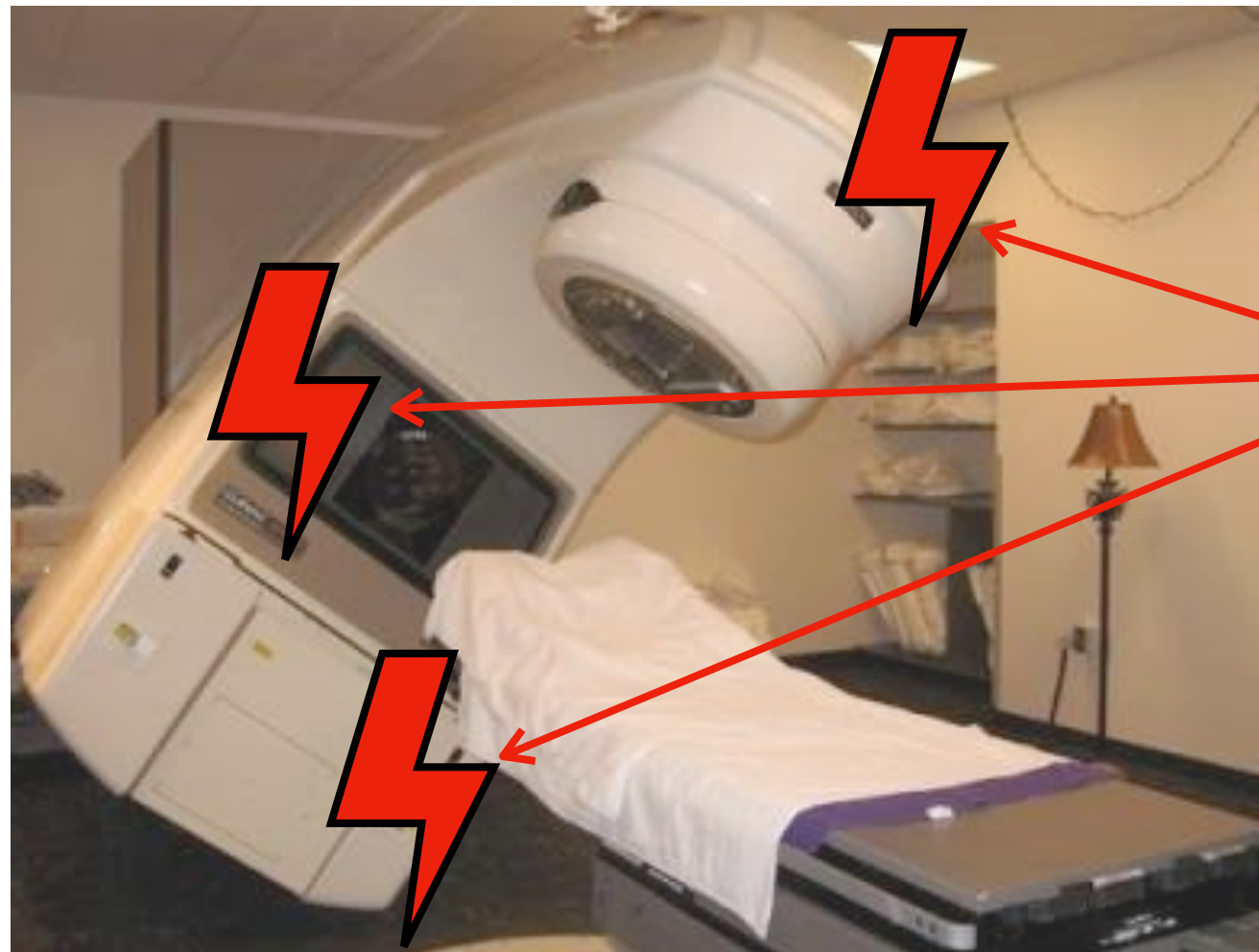
Error counter:

>0 = DANGER



The Therac-25 bug (one of them...)

counter = 3



System events

The Therac-25 bug (one of them...)

counter = 75



The Therac-25 bug (one of them...)

counter = 197



The Therac-25 bug (one of them...)

counter = 255



The Therac-25 bug (one of them...)

counter = 0



The Therac-25 bug (one of them...)

Counter variable
encoded on one
byte!

counter = 0



The Therac-25 bug (one of them...)

Counter variable
encoded on one
byte!

counter = 0



Pressing « **start** »:
if done at the **precise**
« **rollover** » moment,
delivers the **full**
radiation dose to the
patient!



Bibliography

References

- 1.<https://www.ibiblio.org/harris/500milemail.html>
- 2.**Debugging: The 9 indispensable rules for finding even the most elusive software and hardware problems**, David J. Agans, 2002
- 3.**Why Programs Fail**, Andreas Zeller, 2009
- 4.**My Hairiest Bug War Stories**, Mark Eisenstadt, 1997
- 5.**Fighting bugs: Remove, retry, replicate, and rejuvenate**, Grotte and Trivedi, 2007
- 6.**The Science of Debugging**, Telles and Hsieh, 2001