DEFCON XXI

How my Botnet Purchased Millions of Dollars in Cars & Defeated the Russian Hackers

I want to tell you a story about...

- Hacking
- Cars
- Russian Hackers
- Screwing with the system

I want to tell you a story about...

- Commercial Botnets
- Creating competitive advantages
- Not using technology as directed

What you'll learn

- What makes a good Botnet / Webbot project?
- How Bots create competitive advantages for business (example)
- What I would do differently today

(the example happened 6 years ago)

What makes a good Bot project?

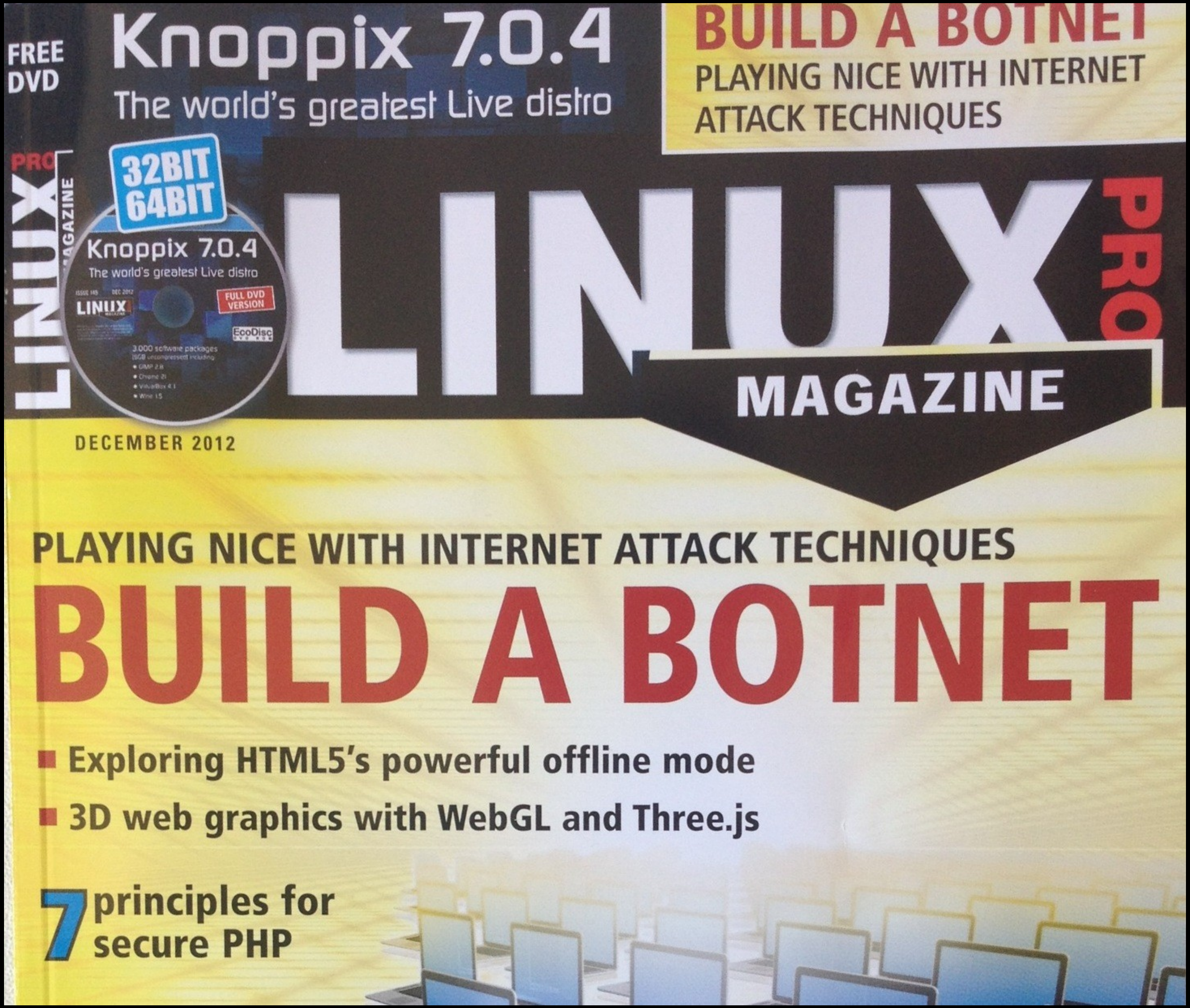
- The Bot...
  - Can't recreate Google
  - Must solve a problem
  - Be viable for it's service life
  - Doesn't “show it's hand”

I have permission to tell this story.

- It's rare that I get to mention specific projects.

- I've been writing about bots since 1999
  - Medical diagnostics
  - Privacy
  - Fraud detection
  - Private investigations
  - Governments

So the 1st thing I did, was write this..



FREE DVD

Knoppix 7.0.4

The world's greatest Live distro

32BIT 64BIT

BUILD A BOTNET

PLAYING NICE WITH INTERNET ATTACK TECHNIQUES

LINUX PRO MAGAZINE

DECEMBER 2012

PLAYING NICE WITH INTERNET ATTACK TECHNIQUES

BUILD A BOTNET

- Exploring HTML5's powerful offline mode
- 3D web graphics with WebGL and Three.js

7 principles for secure PHP

PLAYING NICE WITH INTERNET ATTACK TECHNIQUES

UK Nov 2012
US Dec 2012

The Problem to be Solved

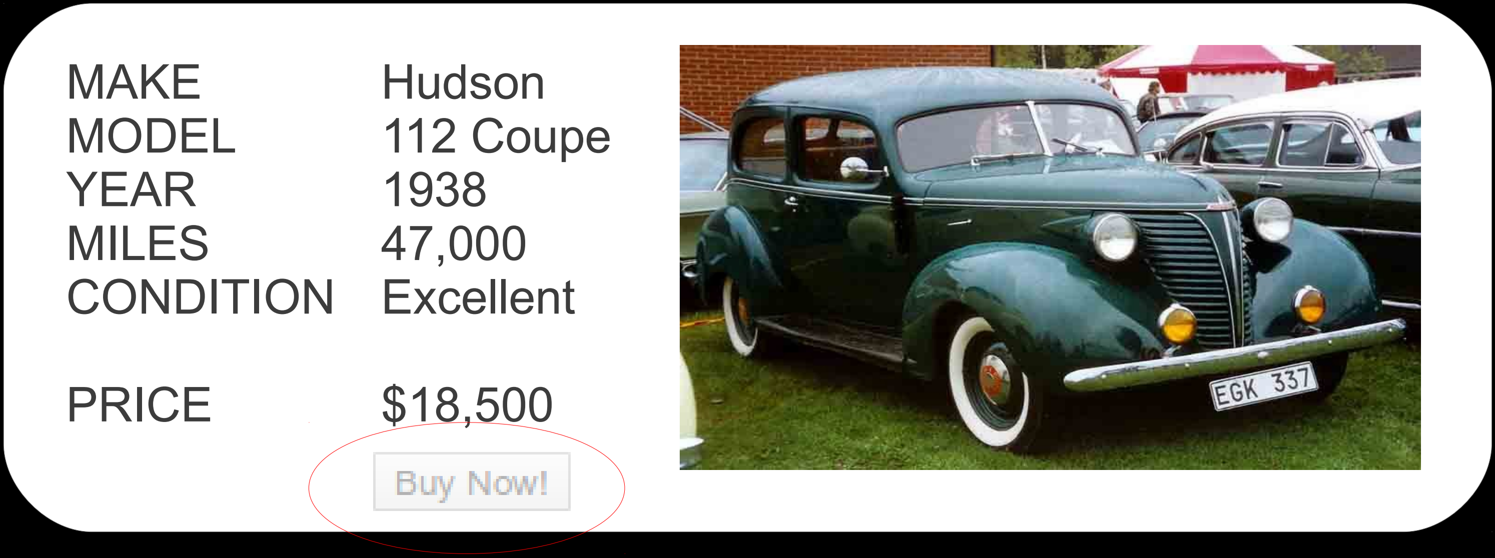
- Dealerships that sell new cars make most of their money on used vehicles.
- Automobile dealers spend a lot of time & money acquiring (previously owned) inventory.
- A client found a website that had great cars for sale.
- Unfortunately, due a lot of competition (and bad web design) he wasn't able to buy the cars he wanted.

The Opportunity

- Daily, a national franchise posted about 300 rental returns for dealerships to purchase.
- They could view the cars before the sale.
- No car could be purchased before 10AM PDT.

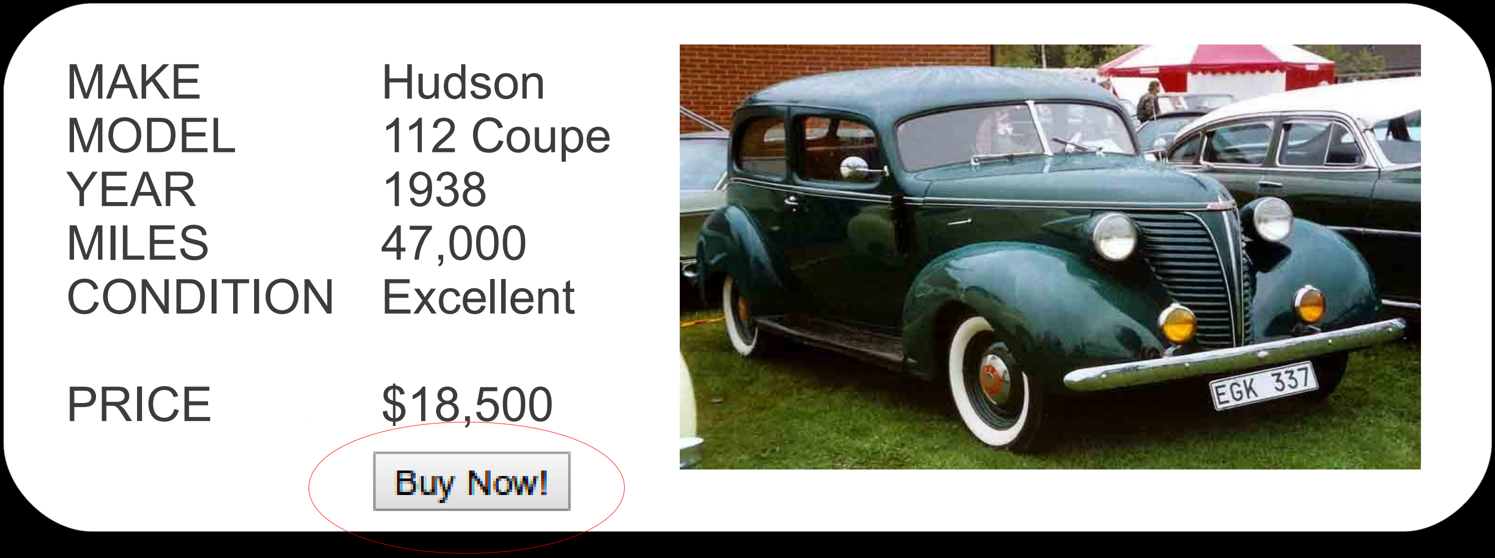
The Opportunity

Before 10:00am



MAKE	Hudson
MODEL	112 Coupe
YEAR	1938
MILES	47,000
CONDITION	Excellent
PRICE	$18,500

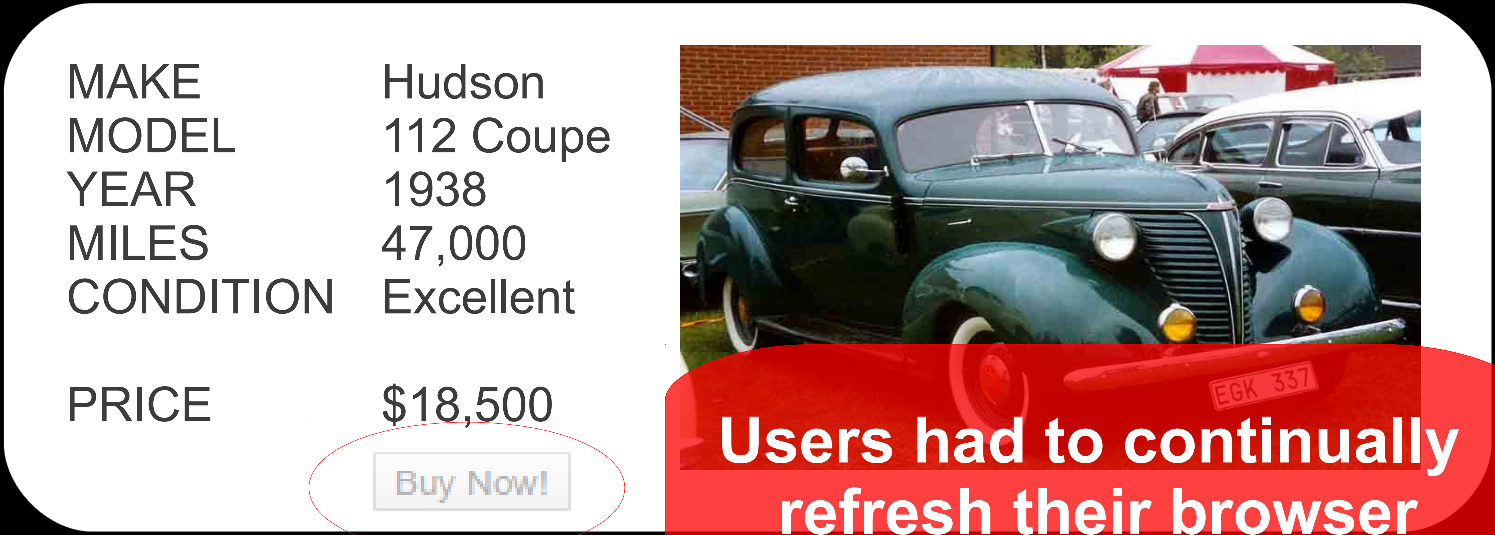
After 10:00am



MAKE	Hudson
MODEL	112 Coupe
YEAR	1938
MILES	47,000
CONDITION	Excellent
PRICE	$18,500

The Opportunity

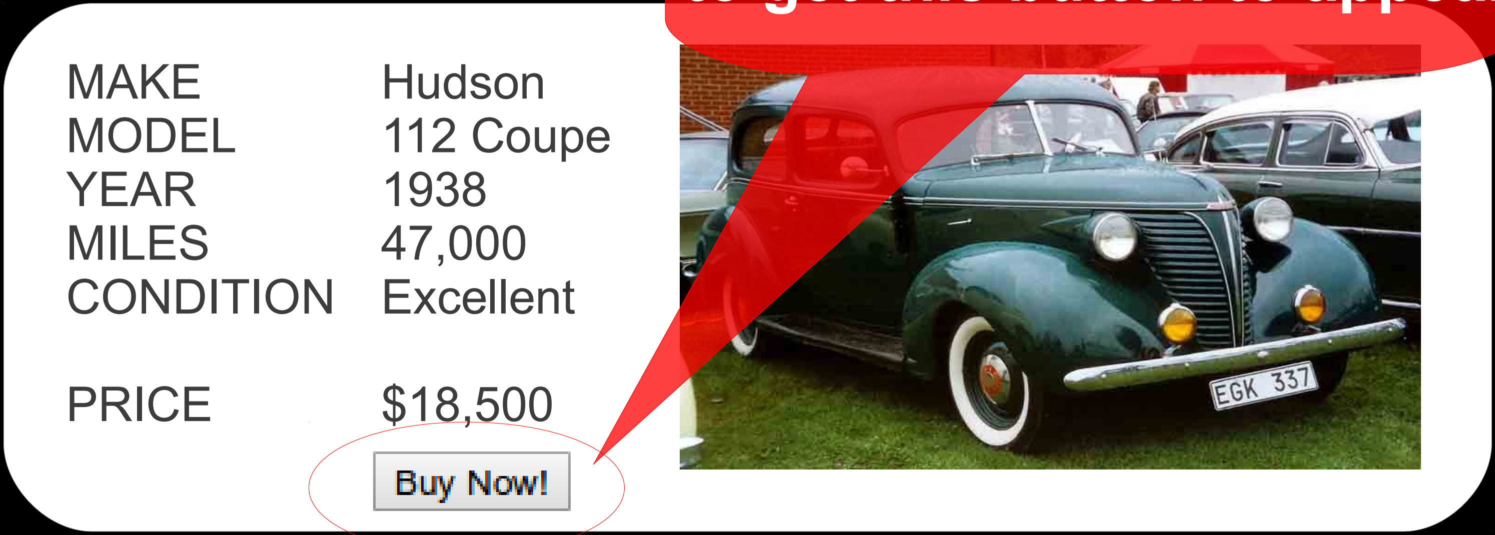
Before 10:00am



MAKE	Hudson
MODEL	112 Coupe
YEAR	1938
MILES	47,000
CONDITION	Excellent
PRICE	$18,500

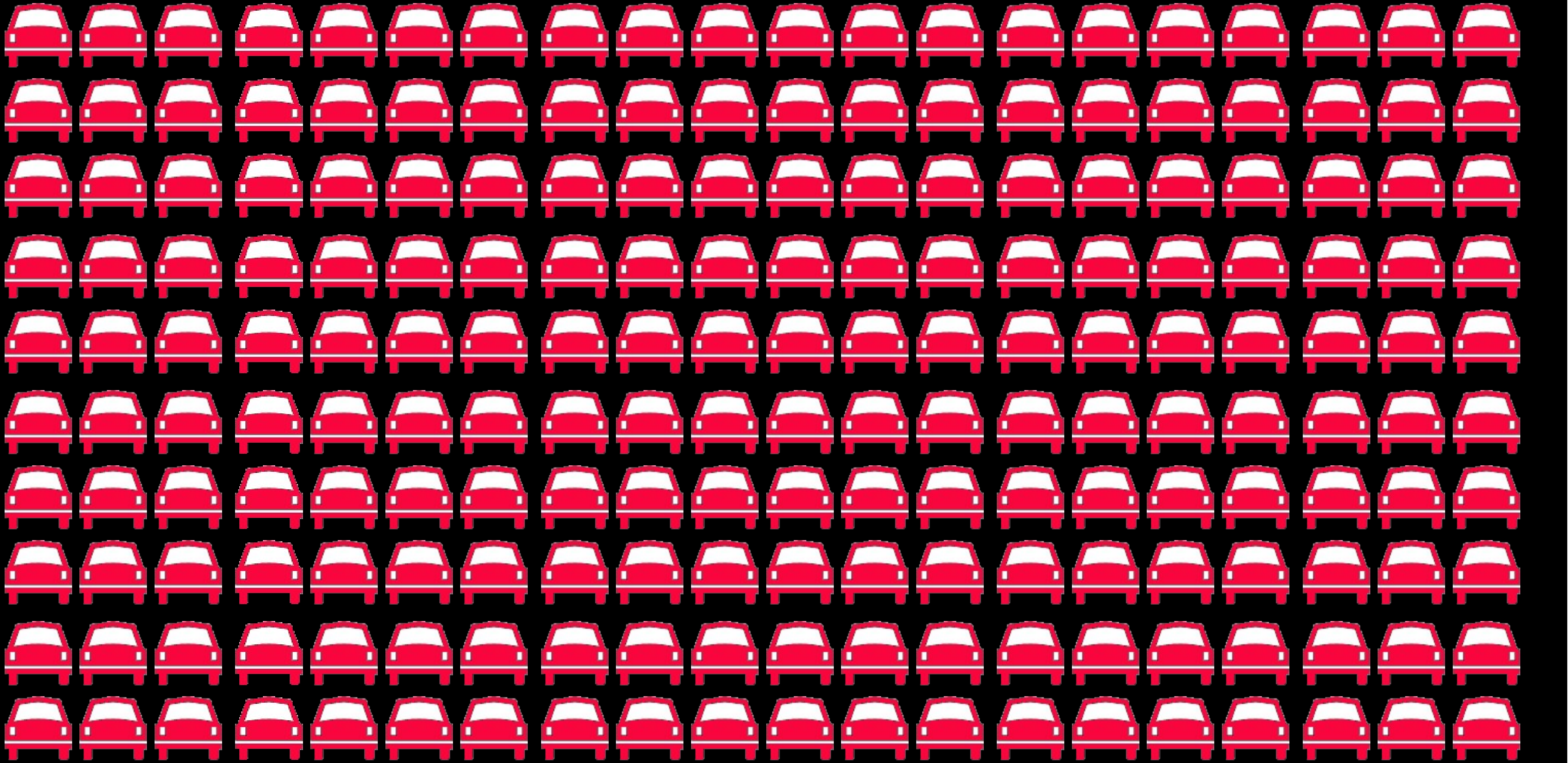
Users had to continually refresh their browser to get this button to appear

After 10:00am



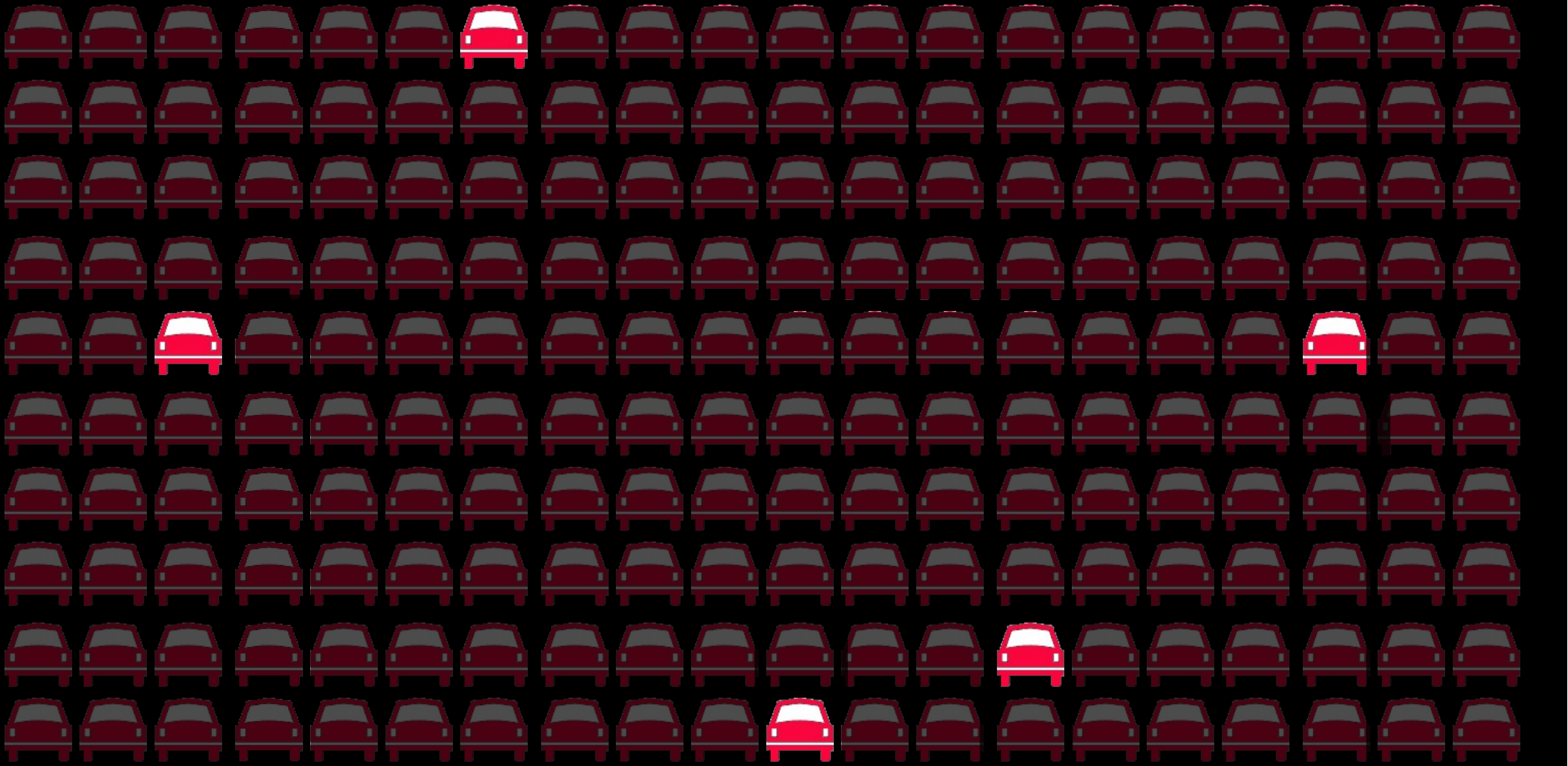
MAKE	Hudson
MODEL	112 Coupe
YEAR	1938
MILES	47,000
CONDITION	Excellent
PRICE	$18,500

And here's the real problem.



If there are two hundred available cars...

And here's the real problem.

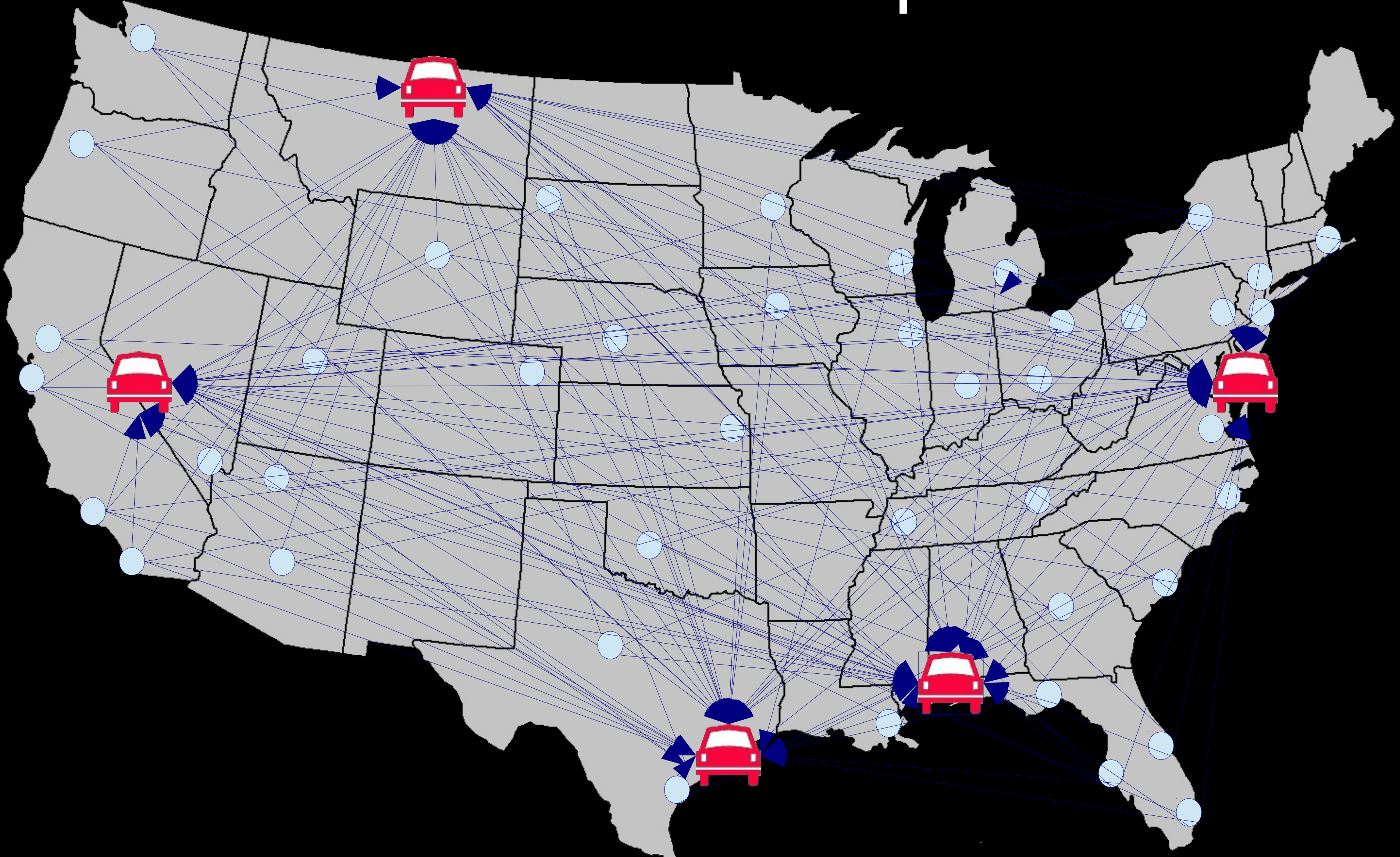


About **five** cars will be highly desirable

And here's the real problem.

Every franchised dealership...

And here's the real problem.



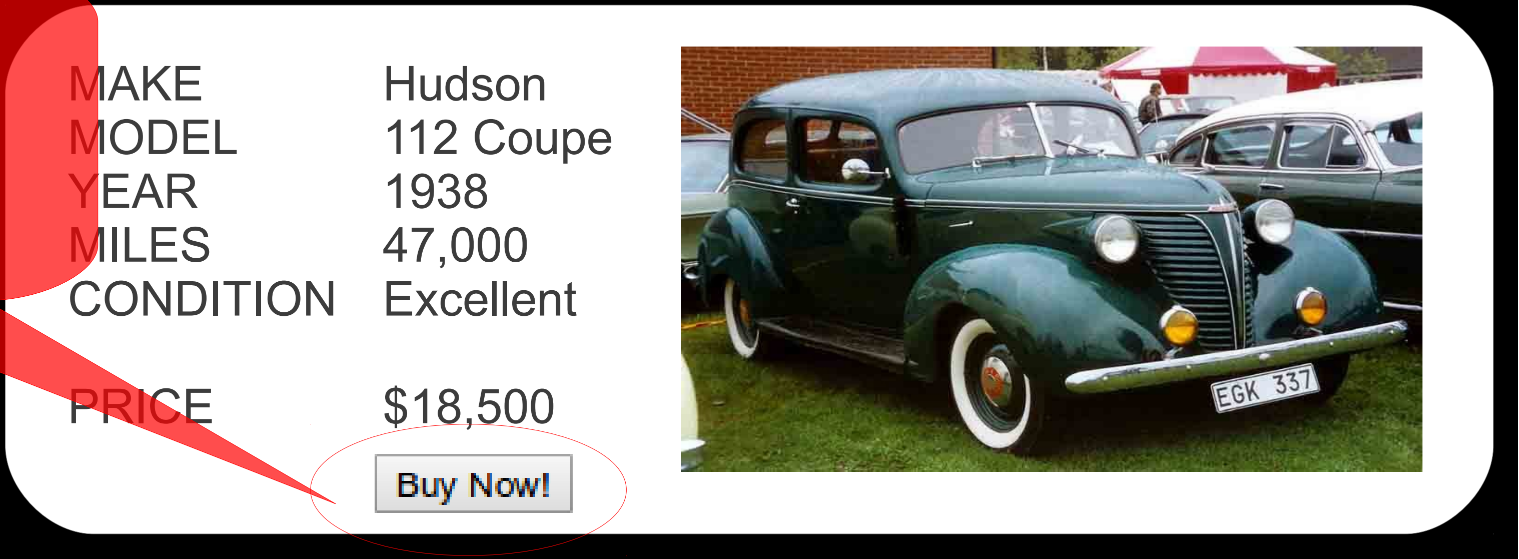
Attempted to buy the same cars.

What my client faced

- There were a limited number of “real deals”
- Every dealership wanted the same cars
- The website's design created peak bandwidth demands, that made the website very hard to use.

How my client coped

While waiting for this button to appear...

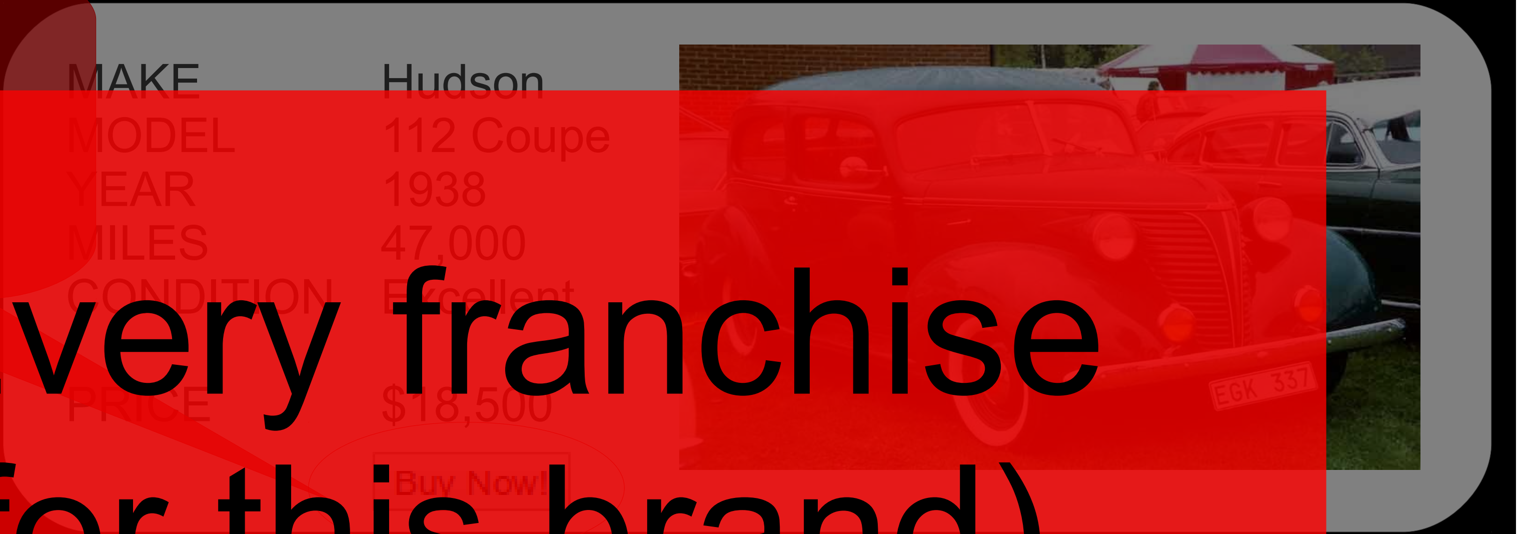


MAKE	Hudson
MODEL	112 Coupe
YEAR	1938
MILES	47,000
CONDITION	Excellent
PRICE	$18,500

- Car dealers would recruit every available person for this task.
- These people would be assigned to specific cars
- They'd have multiple browser windows open
- They'd keep hitting the browser refresh button until it was enabled

How my client coped

While waiting for this button to appear...

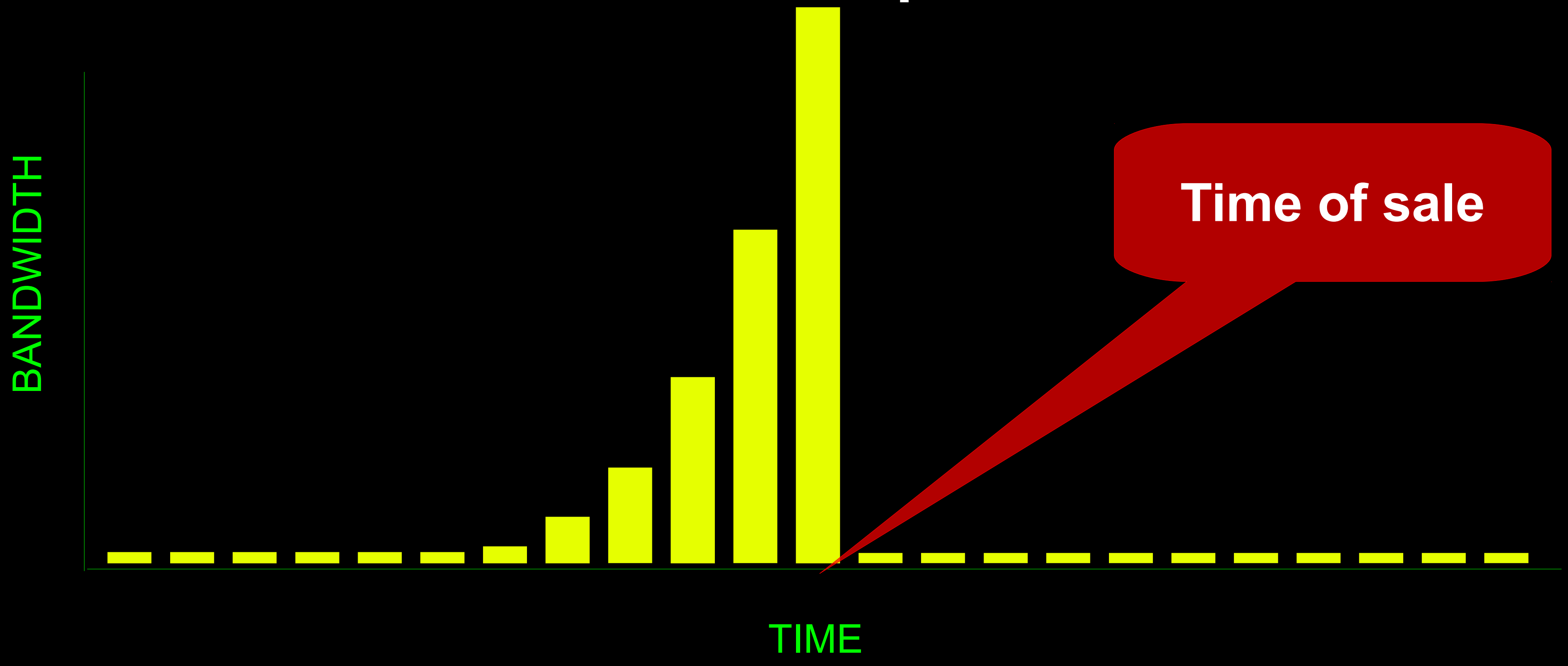


MAKE	Hudson
MODEL	112 Coupe
YEAR	1938
MILES	47,000
CONDITION	Excellent
PRICE	$18,500

- Car dealers would recruit every available person for this task
- These people would be assigned to specific cars
- They'd have multiple browser windows open
- They'd keep hitting the browser refresh button until it was enabled

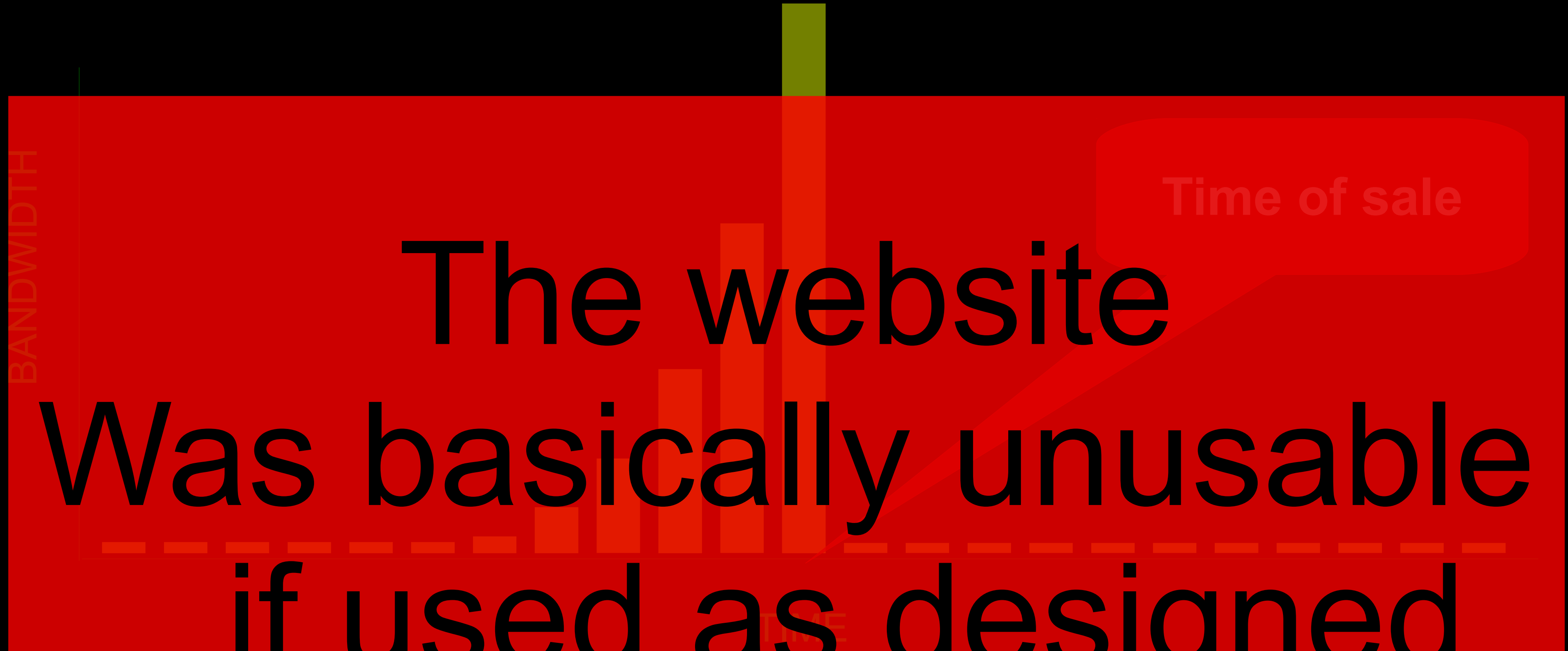
Every franchise (for this brand) did these things

Site design caused a technical problem



- The bandwidth/server lag peaked at the sale time
- It could take as long as 30 seconds for screens to refresh

Site design caused a technical problem



The website
Was basically unusable
...if used as designed

- The bandwidth/server lag peaked at the sale time
- It could take as long as 30 seconds for screens to refresh

The client called and asked if a webbot could help.

Solution, Part 1

Problems with existing system:

1. Was too manual
2. The “Buy” button took too long to appear

Solution, Part 1

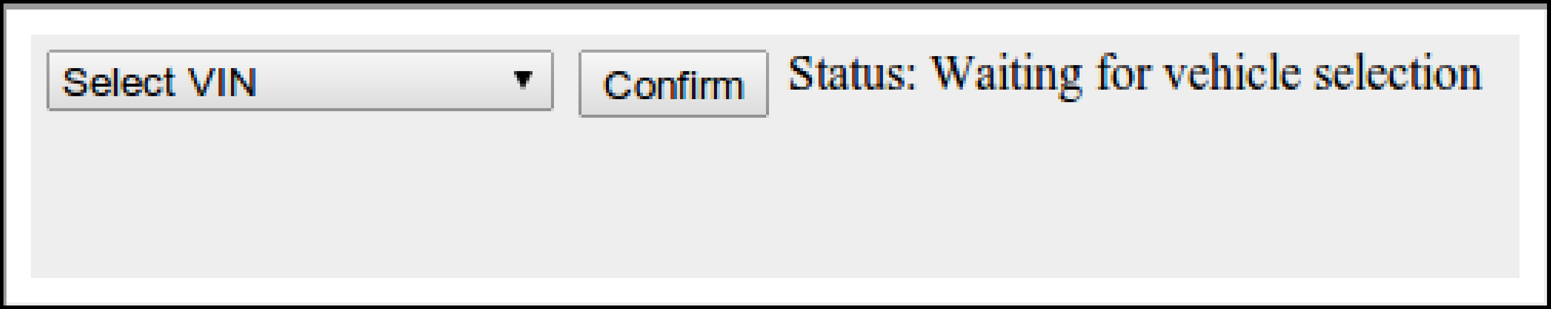
Problems with existing system:

1. Was too manual
2. The “Buy” button took too long to appear

Note:

This was seven years ago, and I don't develop like this anymore.

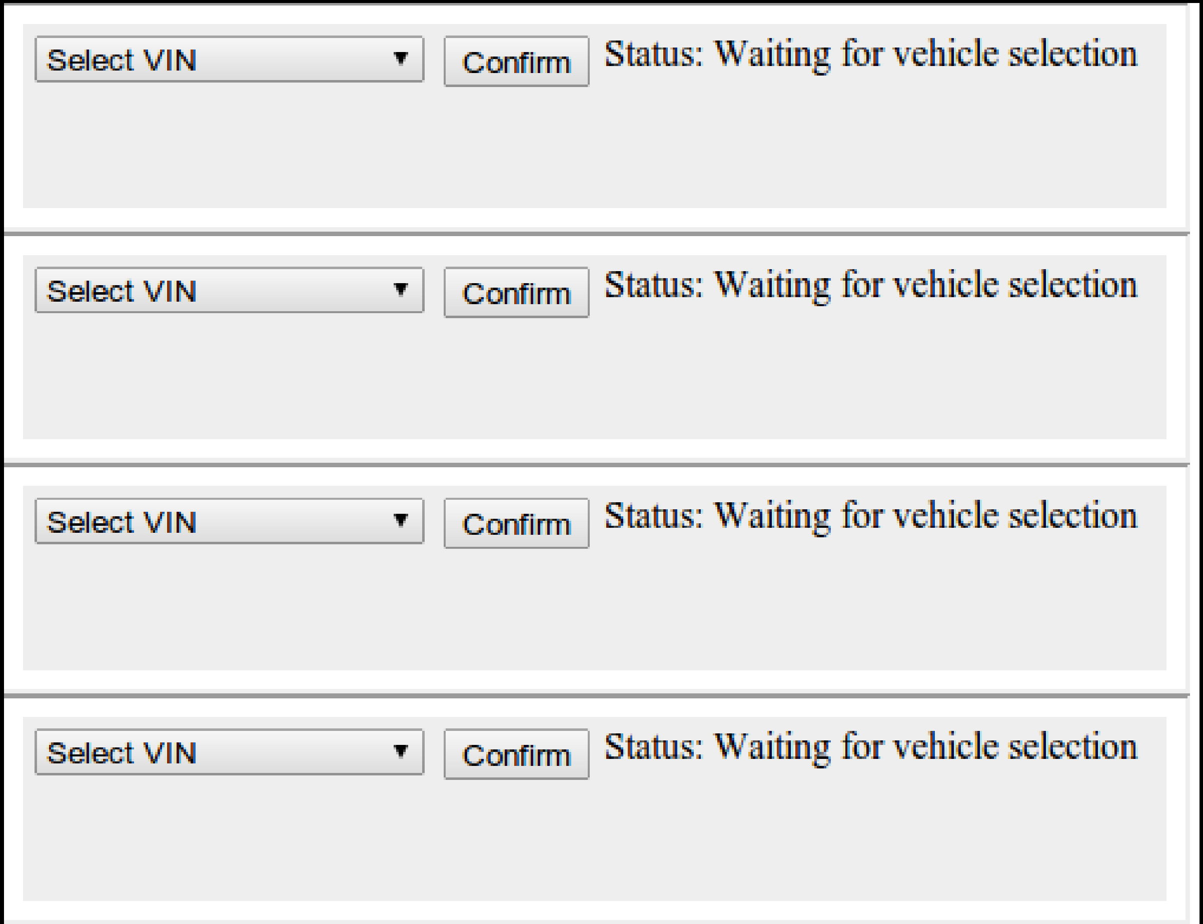
Solution, Part 1



Status: Waiting for vehicle selection

I developed a light weight web interface like this.

Solution, Part 1



Status: Waiting for vehicle selection

Status: Waiting for vehicle selection

Status: Waiting for vehicle selection

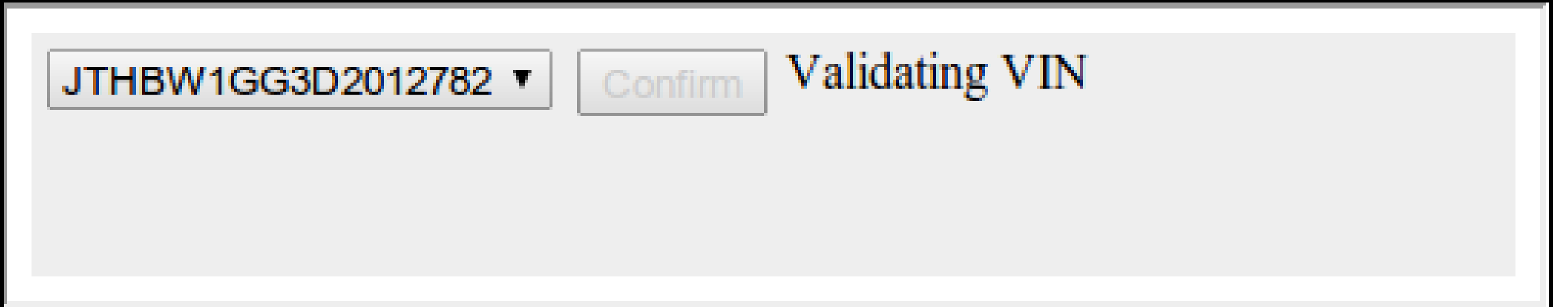
Status: Waiting for vehicle selection

Each client was in an HTML frame.

Initially, we used four instances.

The client would often load the bot on multiple computers.

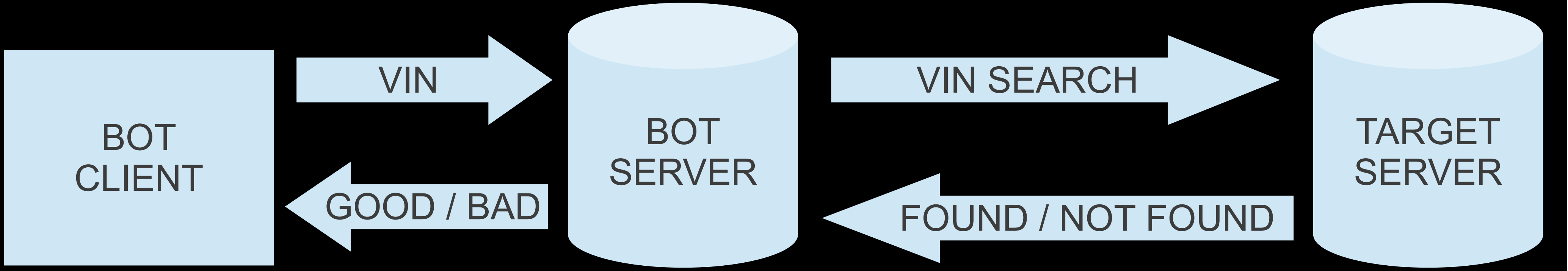
Solution, Part 1



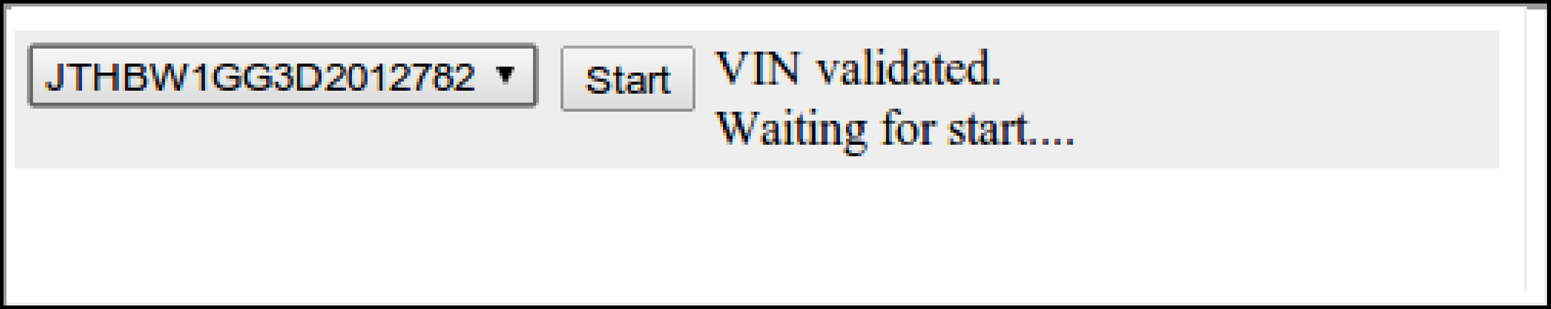
JTHBW1GG3D2012782 ▼

Validating VIN

The first step was to validate the VIN



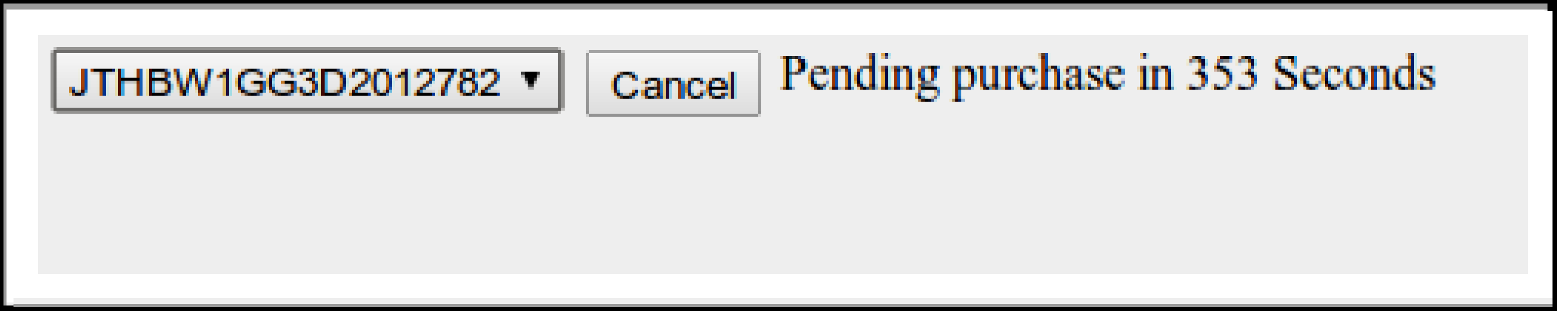
Solution, Part 1



VIN validated.
Waiting for start....

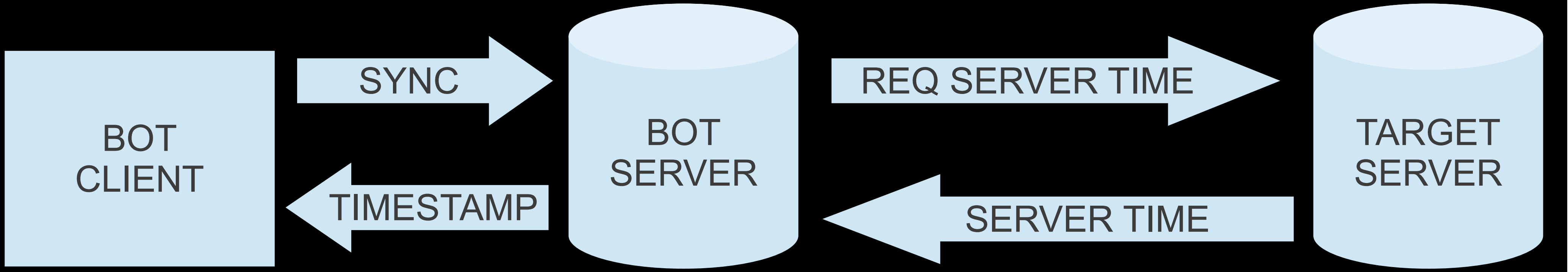
Once the VIN was validated, it waited for the client to tell it to start.

Solution, Part 1

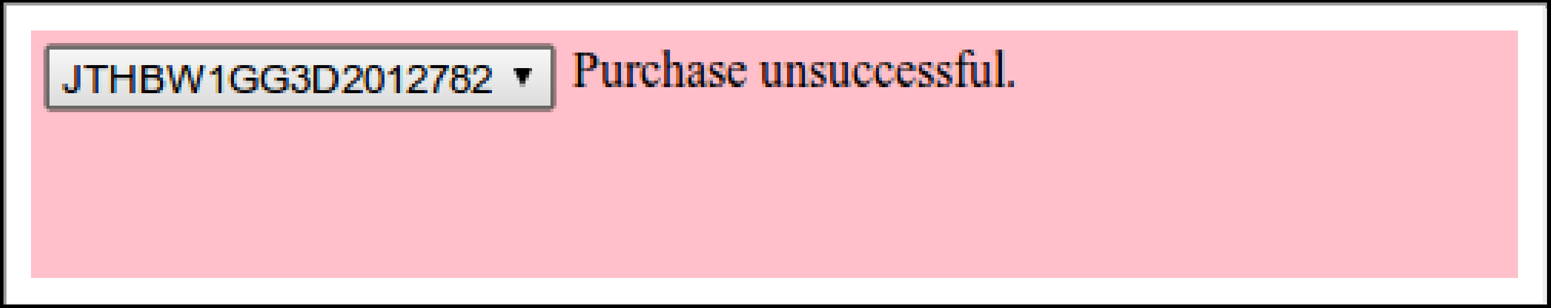


Pending purchase in 353 Seconds

The Client Bot, then synchronized clocks with the Target
And started the count down to purchase.



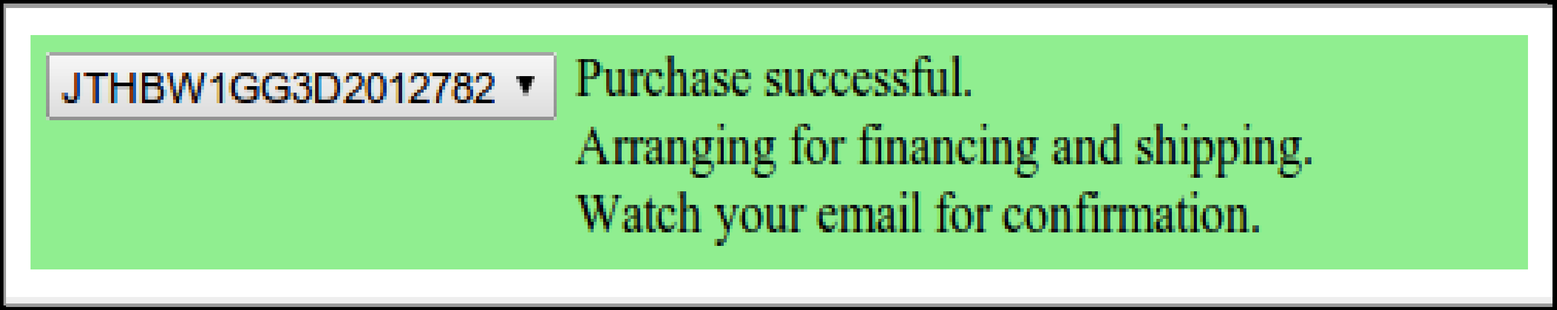
Solution, Part 1



Purchase unsuccessful.

Every now and then, we'd miss one.

Solution, Part 1

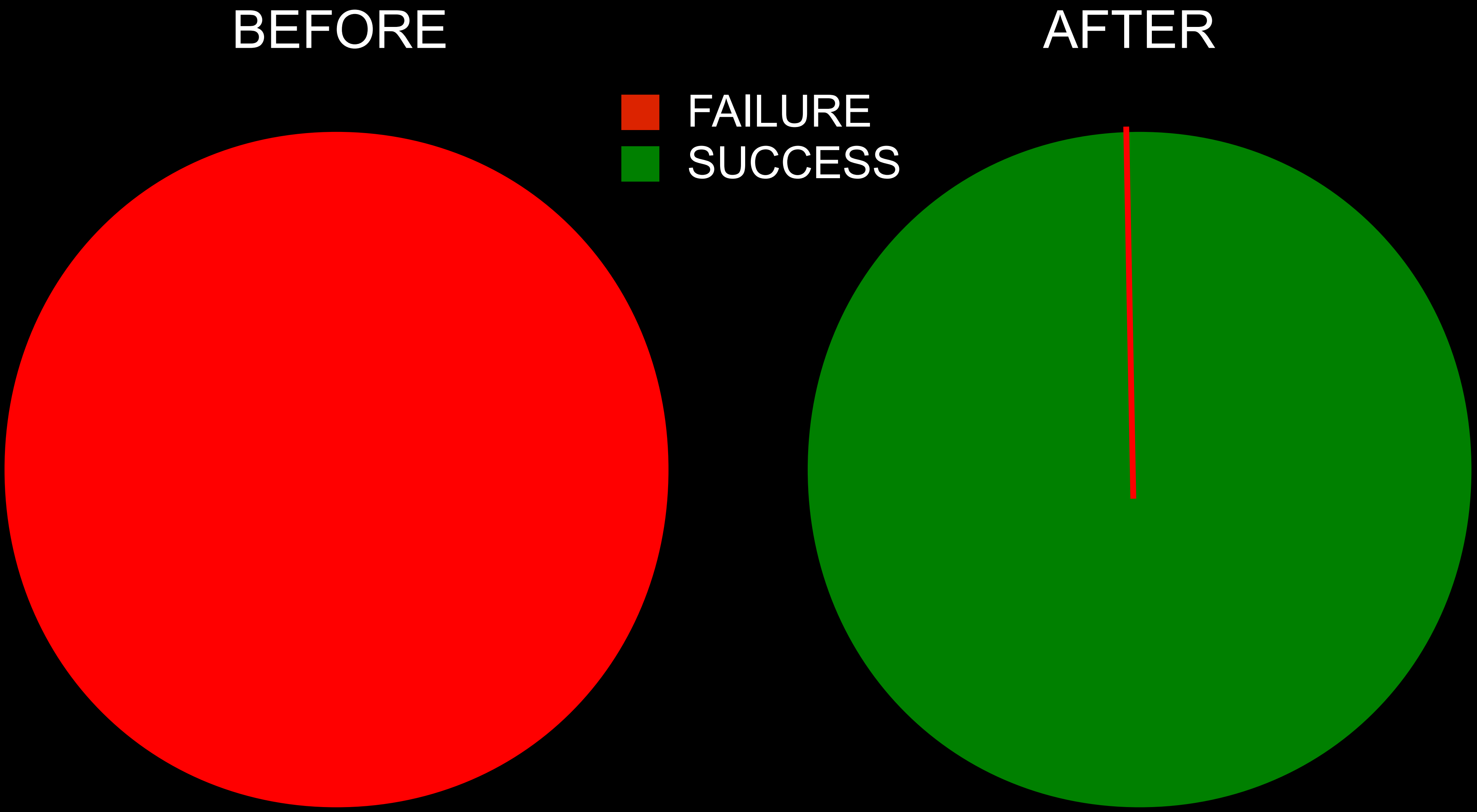


Purchase successful.
Arranging for financing and shipping.
Watch your email for confirmation.

But more often:

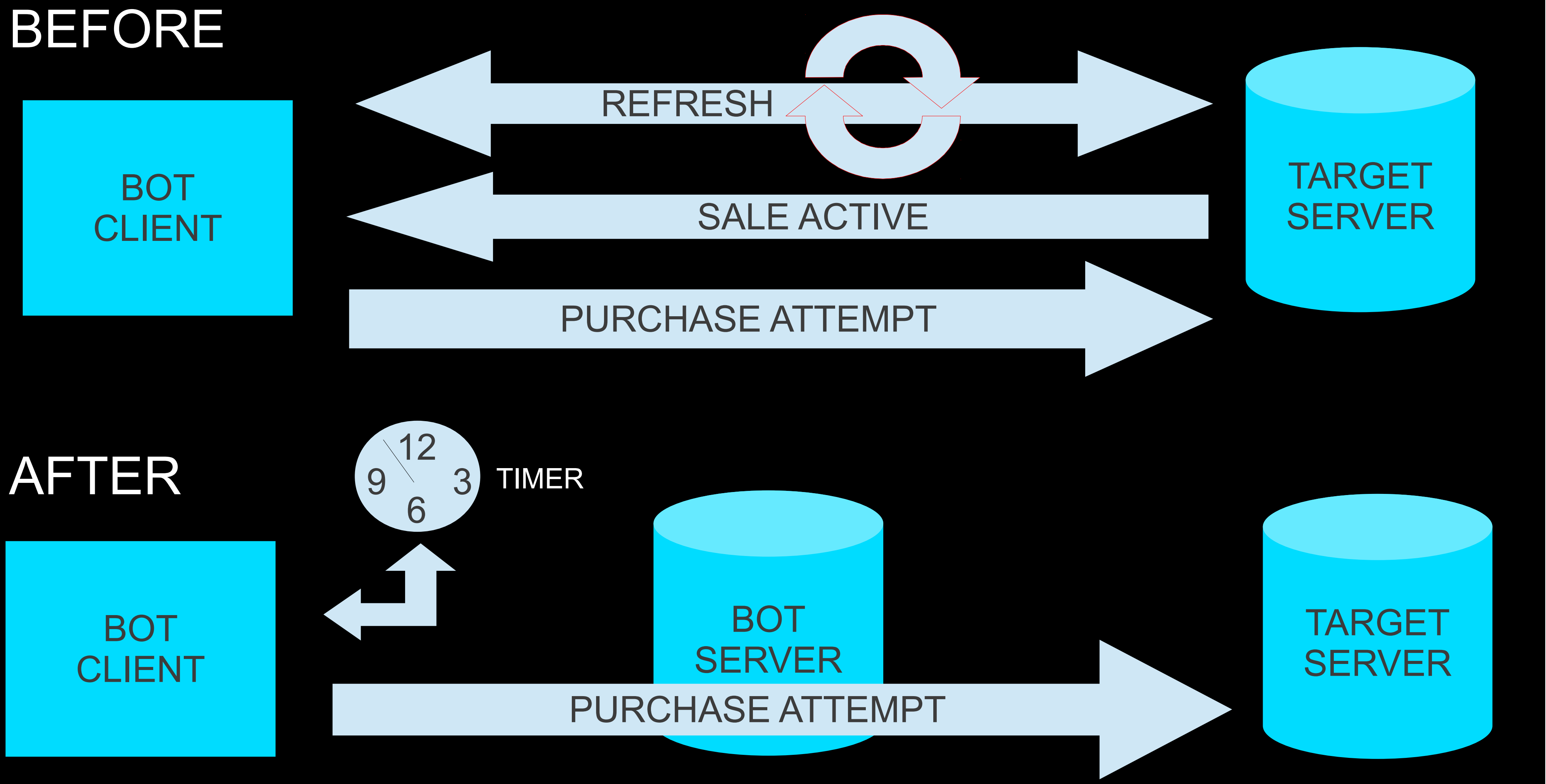
- The sale was successful, and
- The Bot arranged for financing & shipping

Success, Part 1

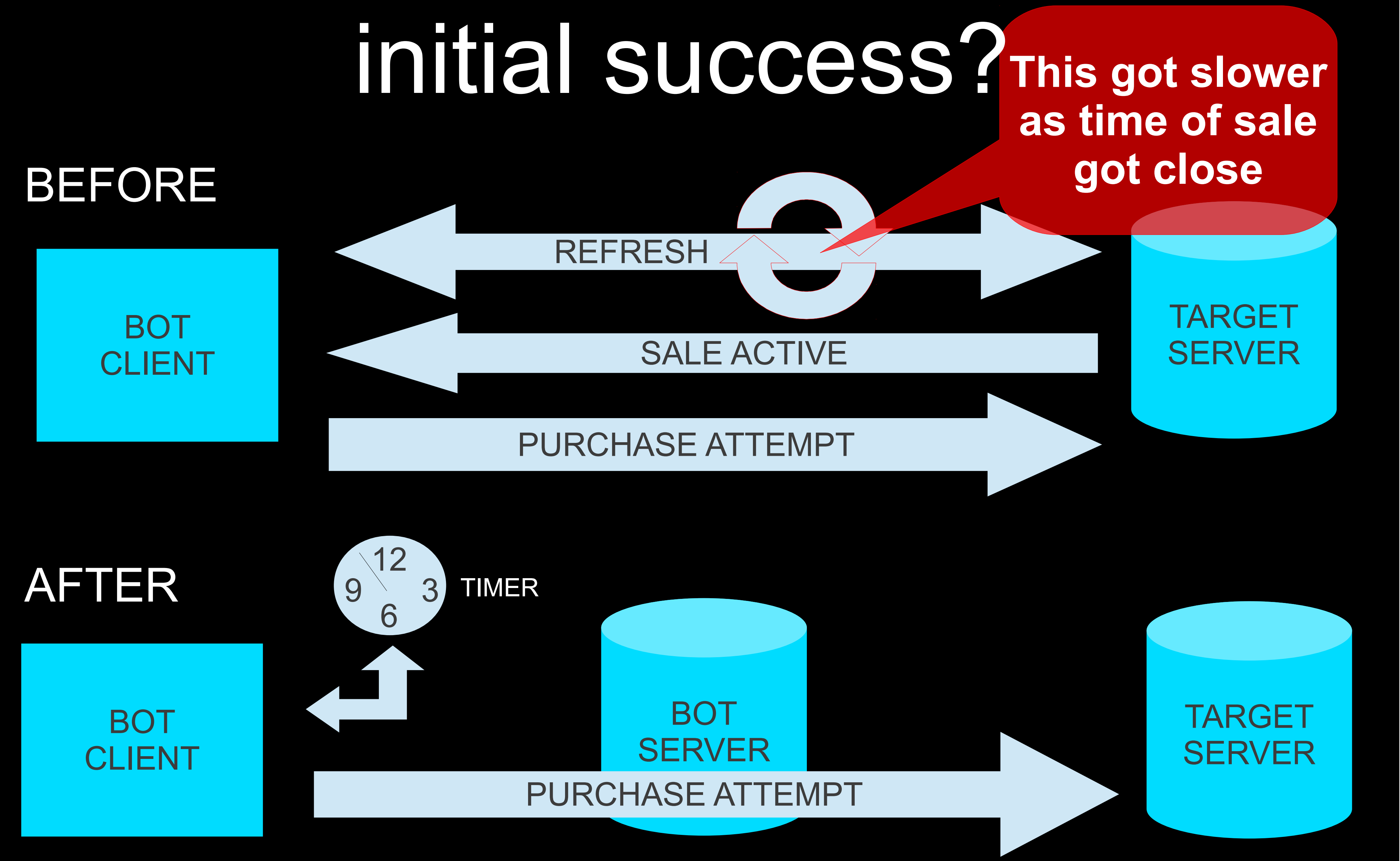


The client's purchase success rate went from 0% to near 100%

What accounted for the initial success?

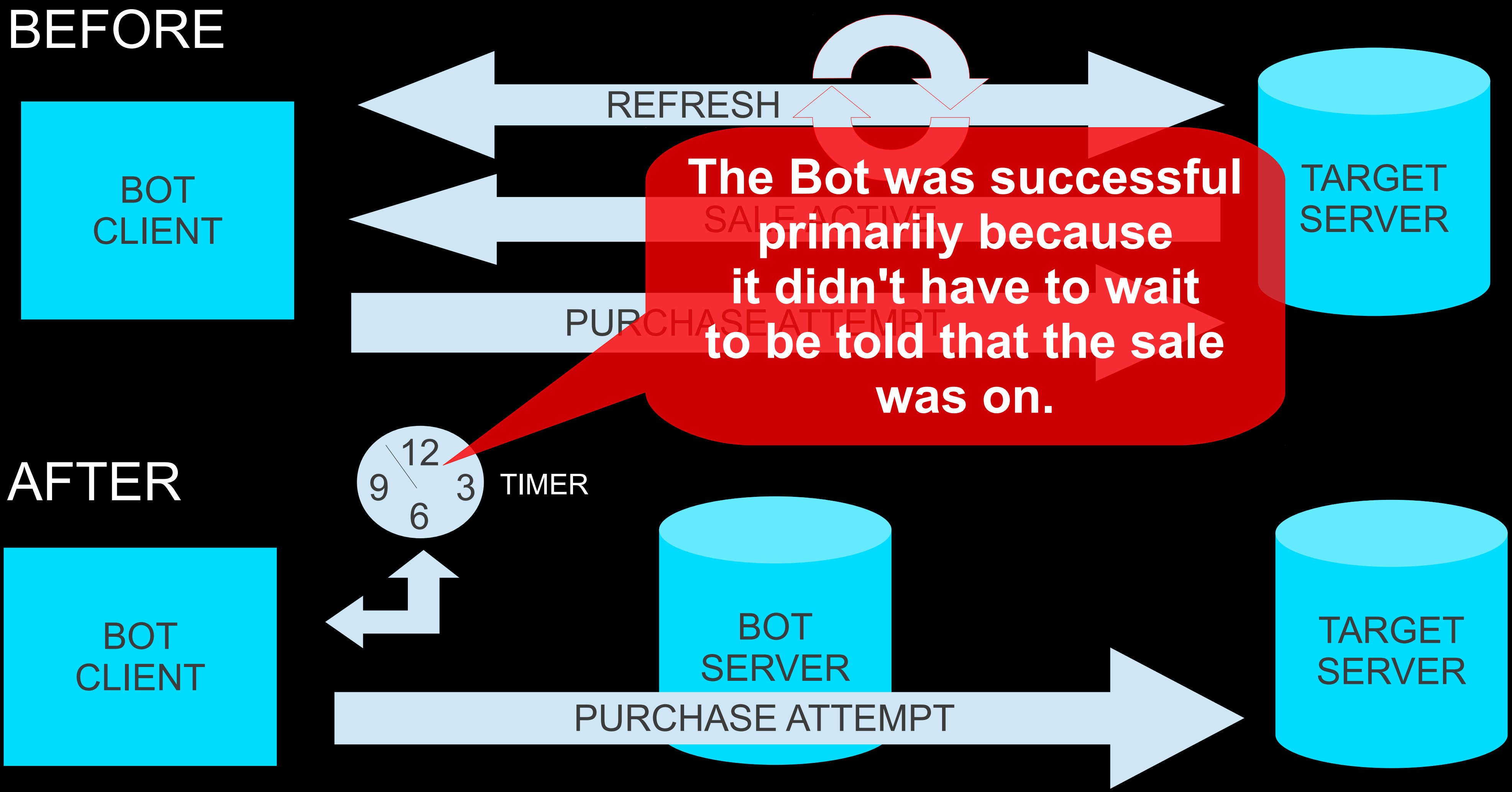


What accounted for the initial success?



This got slower as time of sale got close

What accounted for the initial success?



The Bot was successful primarily because it didn't have to wait to be told that the sale was on.

A sign of problems

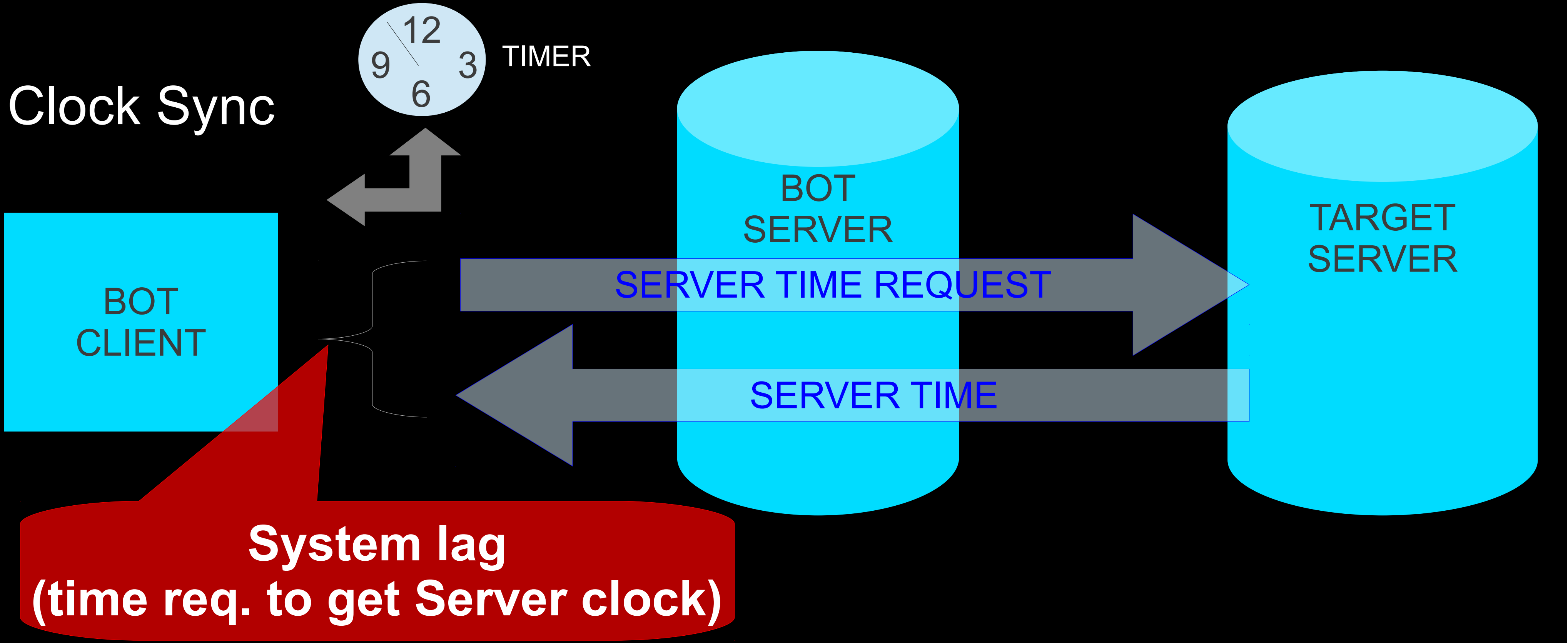
The client continued to have success for about six months.

- Suddenly, success rates dropped to about 50%

- My client discovered a competing bot developed by a group of Russian Hackers

- Competition is good, and leads to innovation.

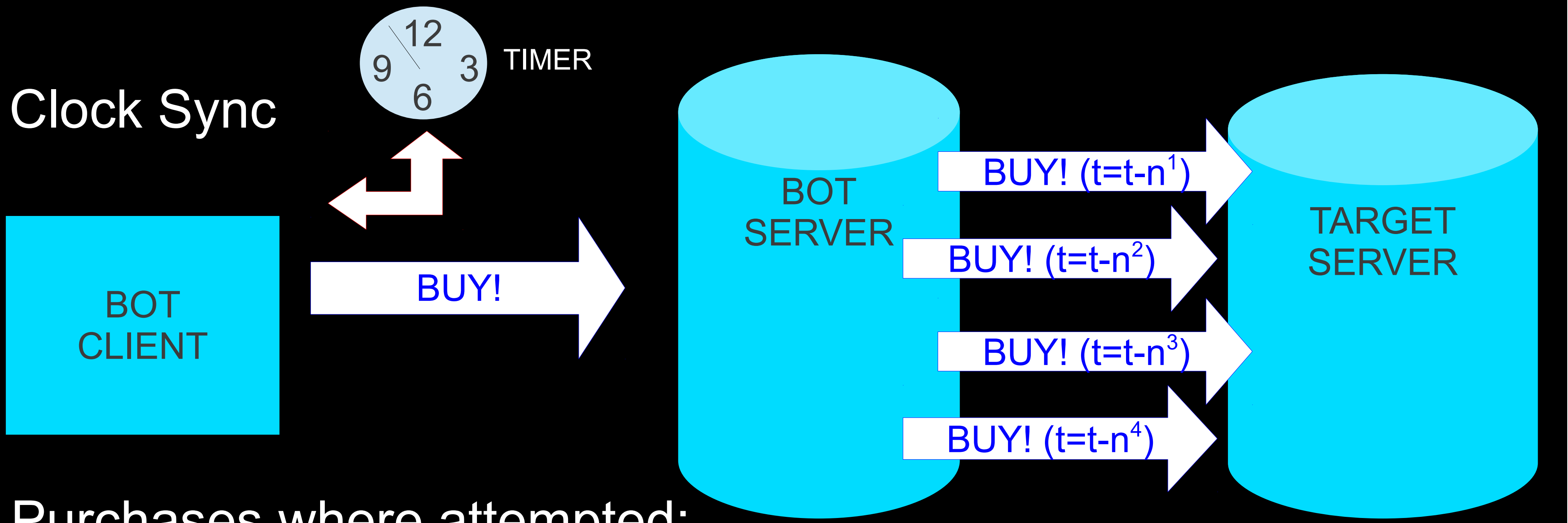
Solution, Part 2a



System lag
(time req. to get Server clock)

The clock synchronization was modified to make checks more often as the sale neared, and also calculated **system lag**.

Solution, Part 2b



Purchases where attempted:

- The Bot Client triggered the Bot Server to make multiple attempts to buy the vehicle.
- Each attempt was made slightly prior to the actual sale time and based on calculated system delays.

How successful was this Bot?

- The Bot operated for about 40 weeks
- The client bought approximately 20 cars a week (estimated)
- Total cars purchased **800** (estimated)
- If the average wholesale cost ~$16,000
- The Bot purchased **$12,800,00**
  (estimated wholesale value)

What would I do differently?

- What did it do well?
  - More successful than anticipated
  - Very lightweight clients
    - Easily updated
    - Easily distributed

What would I do differently?

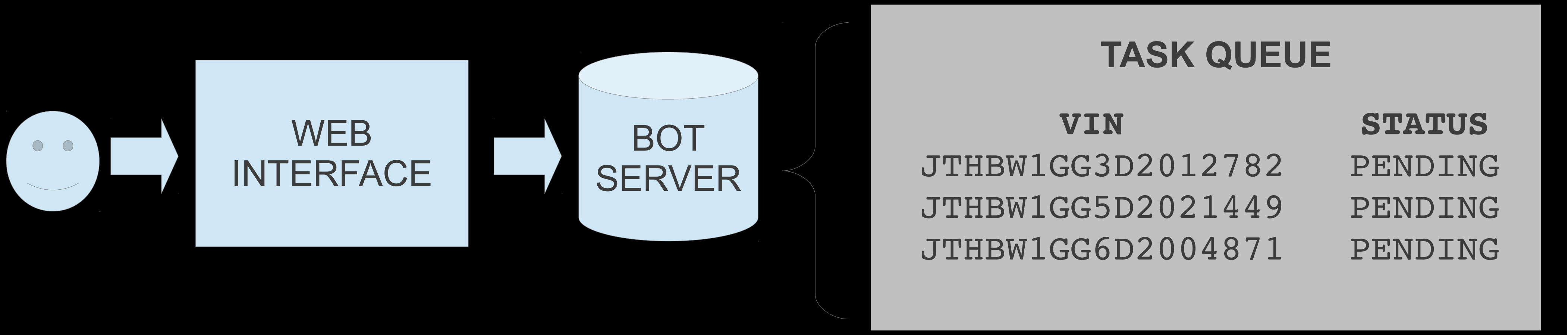
- Fairly stealthy
  - It required authenticated users
    - Affects stealthiness
    - Tried to simulate human behavior
    - Using multiple accounts
  - The expected behavior was pretty weird

What would I do differently?

- Today, it would have to accommodate newer technology
  - The sale website used standard HTML forms, which are easy to emulate (submit) with simple PHP scripts
  - Today's websites are more suffocated
    - AJAX
    - Complex forms, etc.

What would I do differently?

- Vehicles could be written into a “task queue” via a web interface to the BOT SERVER

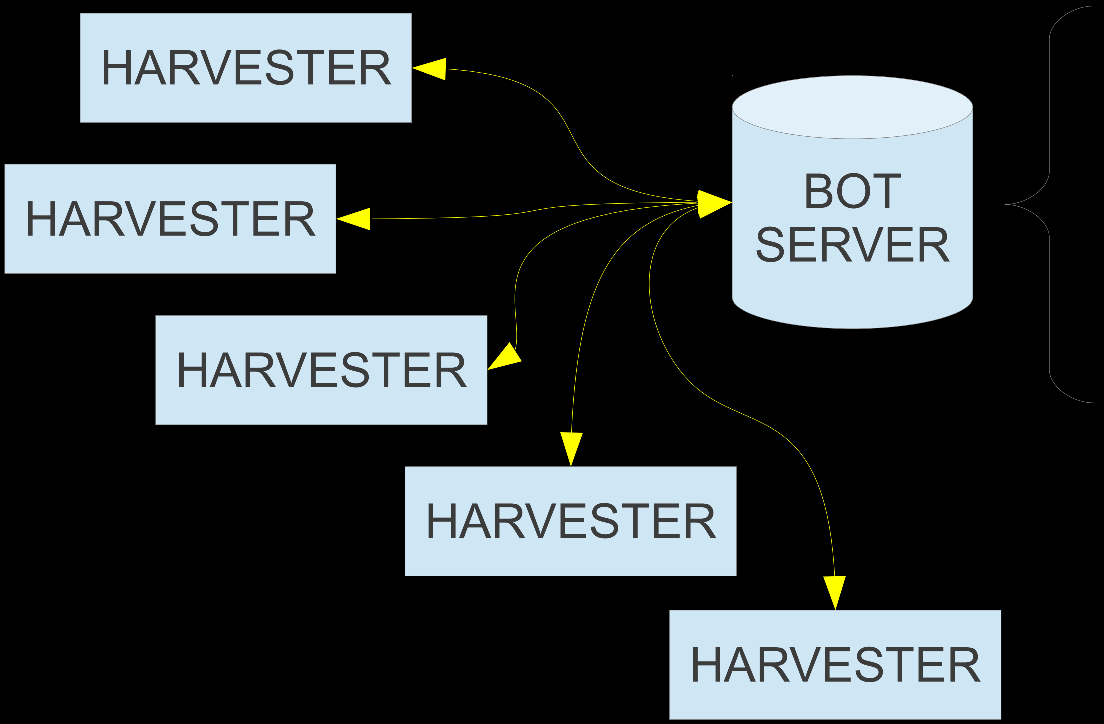


TASK QUEUE

VIN	STATUS
JTHBW1GG3D2012782	PENDING
JTHBW1GG5D2021449	PENDING
JTHBW1GG6D2004871	PENDING

What would I do differently?

- Tasks in the Task Queue are distributed to individual computers that I call *harvesters*.



TASK QUEUE

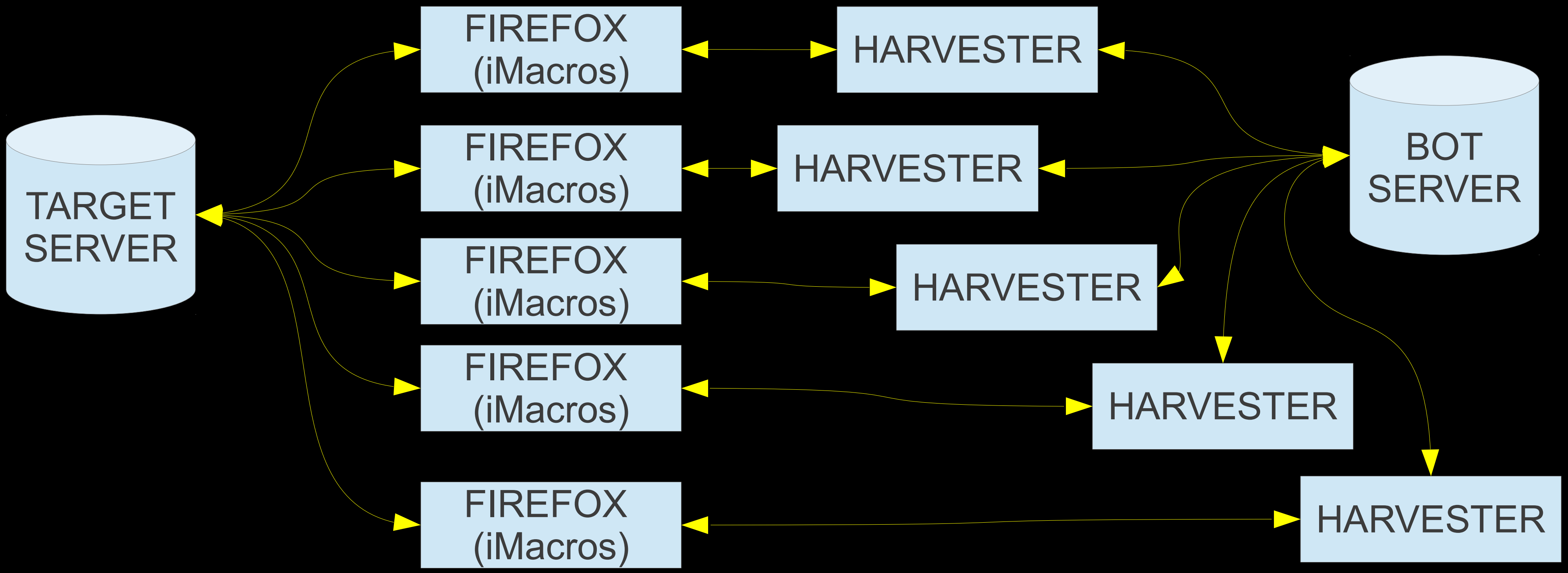
VIN	STATUS
JTHBW1GG3D2012782	PENDING
JTHBW1GG5D2021449	PENDING
JTHBW1GG6D2004871	PENDING
...	...

The harvesters may be located anywhere

- Data Center
- Office
- Cloud

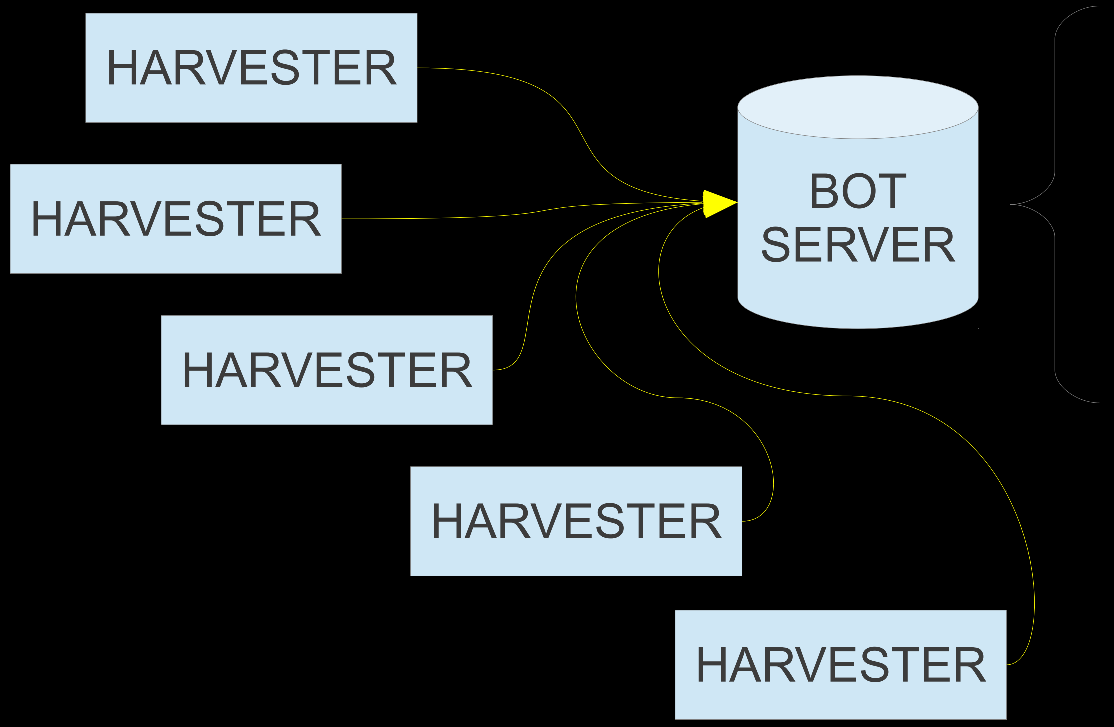
What would I do differently?

- The harvesters create (iMacros) **browser macros** on-the-fly and execute commands directly in Firefox



What would I do differently?

Once the task is completed, the harvesters communicate back to the Bot Server, which updates the Task Queue.



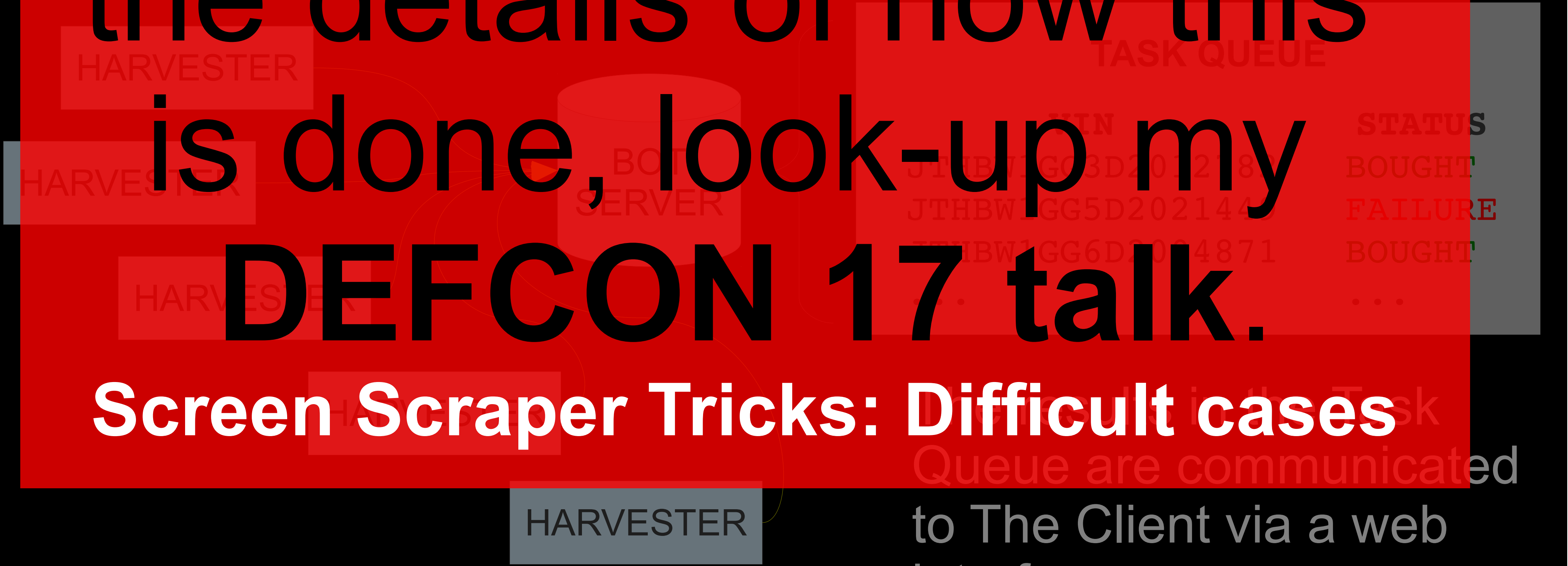
TASK QUEUE

VIN	STATUS
JTHBW1GG3D2012782	BOUGHT
JTHBW1GG5D2021449	FAILURE
JTHBW1GG6D2004871	BOUGHT
...	...

The results in the Task Queue are communicated to The Client via a web interface.

What would I do differently?

Once the task is completed, the harvesters communicate back to the Bot Server, which updates the Task Queue.



VIN	STATUS
JTHBW1GG3D2012182	BOUGHT
JTHBW1GG5D2021440	FAILURE
JTHBW1GG6D2004871	BOUGHT
...	...

Results in the Task Queue are communicated to The Client via a web interface.

If you're interested in the details of how this is done, look-up my **DEFCON 17 talk.**

Screen Scraper Tricks: Difficult cases

SHOUTS

Thanks to **All of You**, the **DEFCON CFP Goons** (particularly **Nikita**) for allowing me to give my **5th** DEFCON talk.

If you like this type of talk, the rest of my talks are on YouTube.

- DEFCON X Developing Webbots & Agents
- DEFCON XI Corporate Intelligence
- DEFCON XV The Incredible Executable Image Exploit
- DEFCON XVII Advanced Scraping: Difficult Cases

(search “schrenk DEFCON”)

mgschrenk@gmail.com *www.schrenk.com*