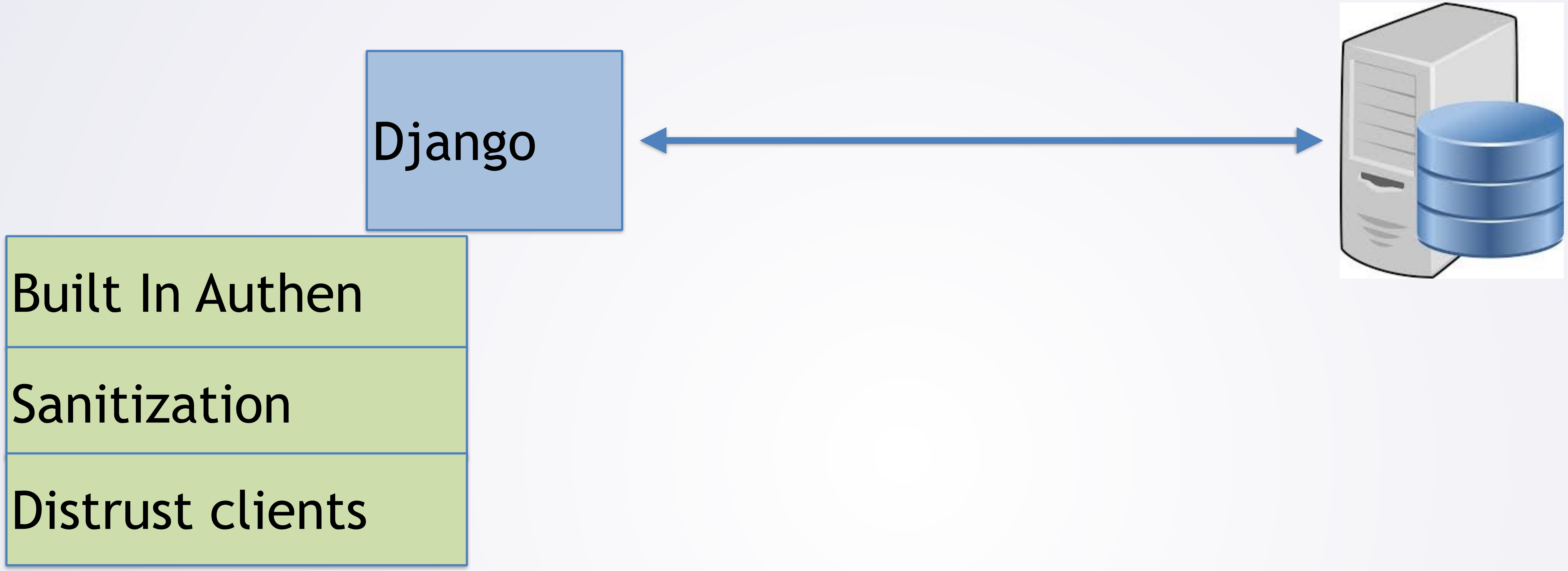
# Engineering Robust Server Software

## Defense In Depth

Brian Rogers &
Tyler Bletsch / Duke ECE
Used with permission from Drew Hilton

Duke
UNIVERSITY

# You Are Building YourAwesomeSite.com



Django

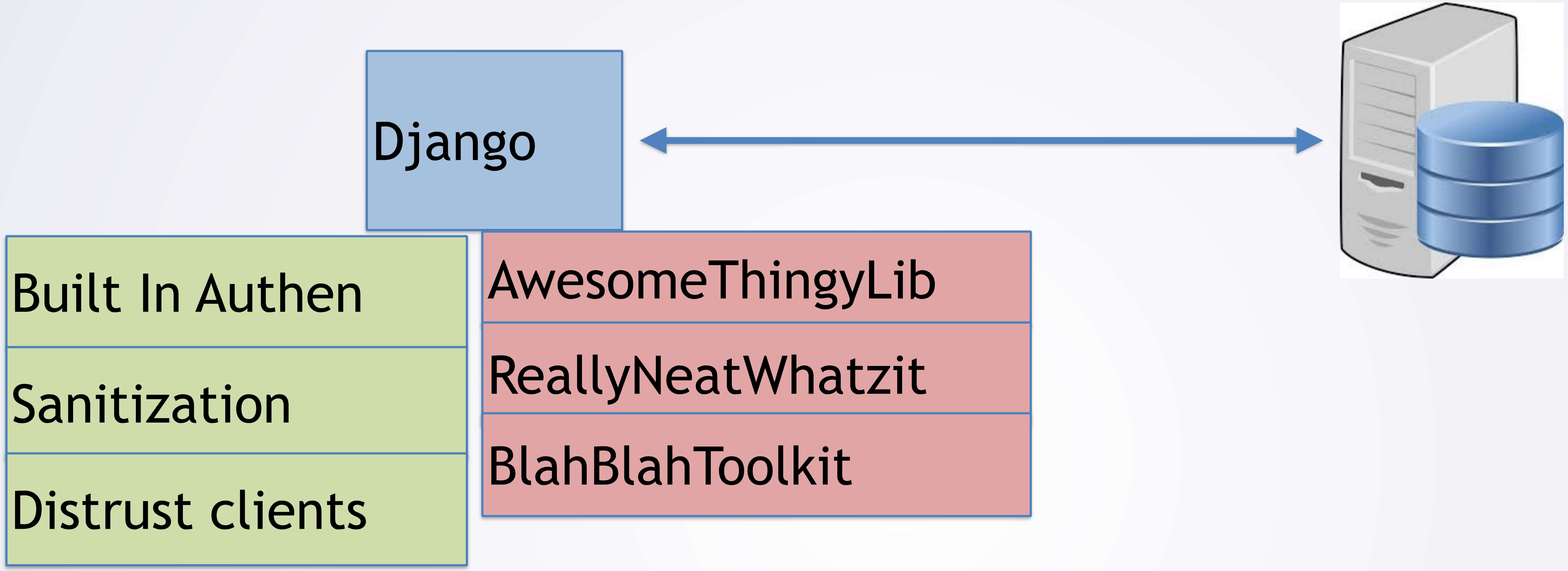
Built In Authen

Sanitization

Distrust clients

- Use all the best practices you know

# You Are Building YourAwesomeSite.com



Django

Built In Authen

Sanitization

Distrust clients

AwesomeThingyLib

ReallyNeatWhatzit

BlahBlahToolkit

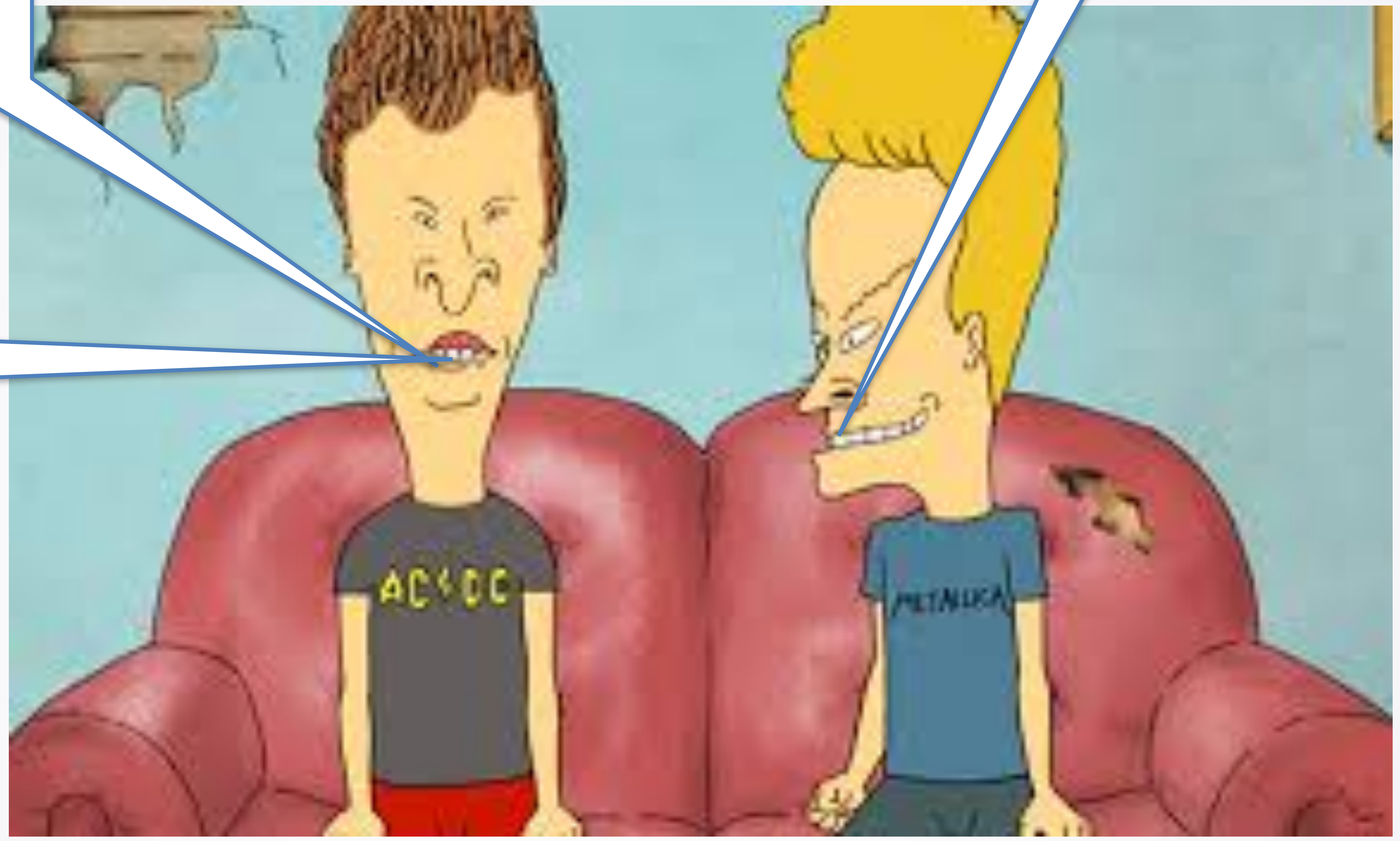
- …But also lots of things you didn't write
  - Adds a lot of complexity…

# You Are Building YourAwesomeSite.com

Hey Beavis,
I found this code on StackOverflow

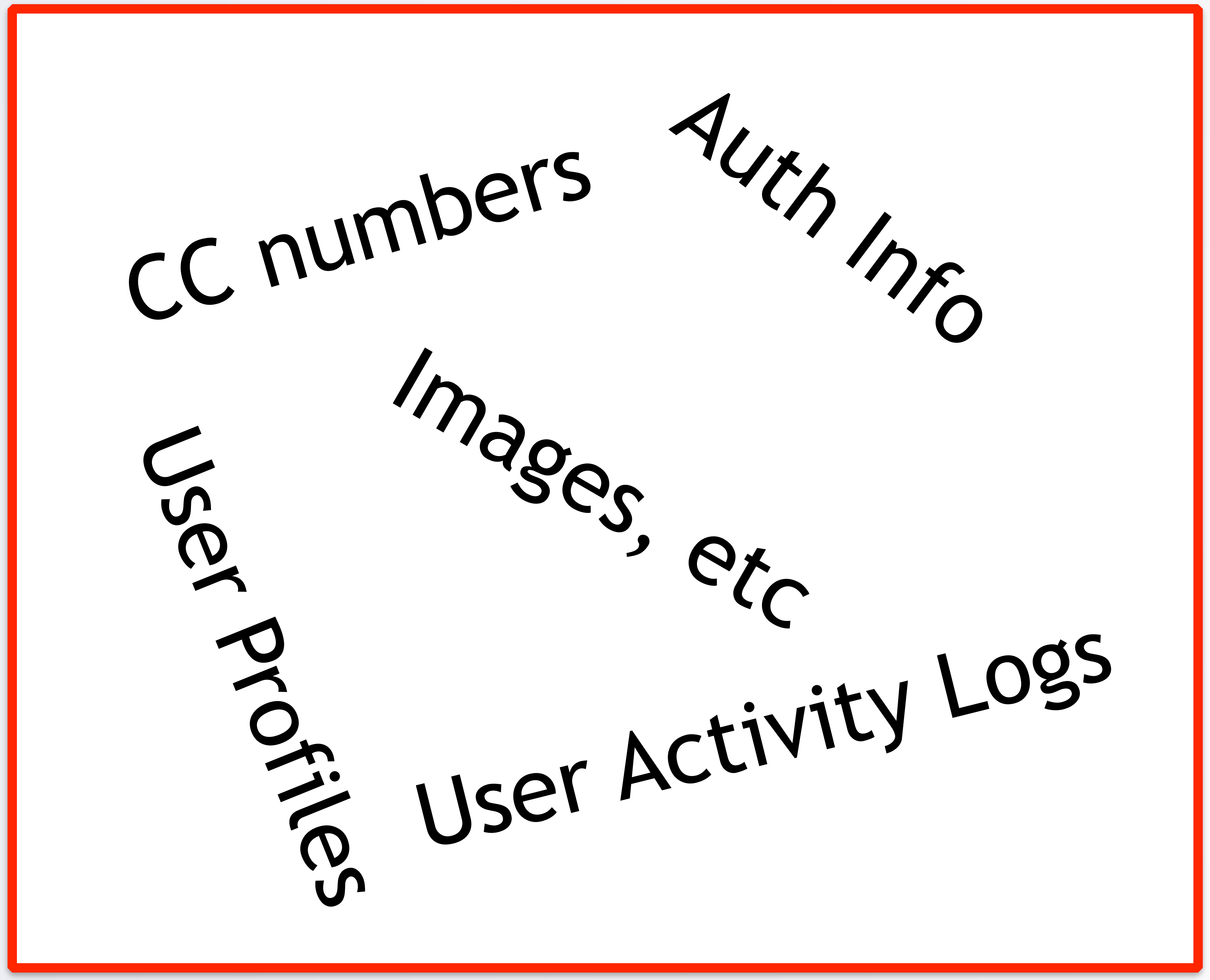
Heheh Whats it do Butthead? Heheh

I dont know but they said its really awesome.



- Oh plus what about the other developers on your team?

# What Happens If Something Goes Wrong?


CC numbers

Auth Info

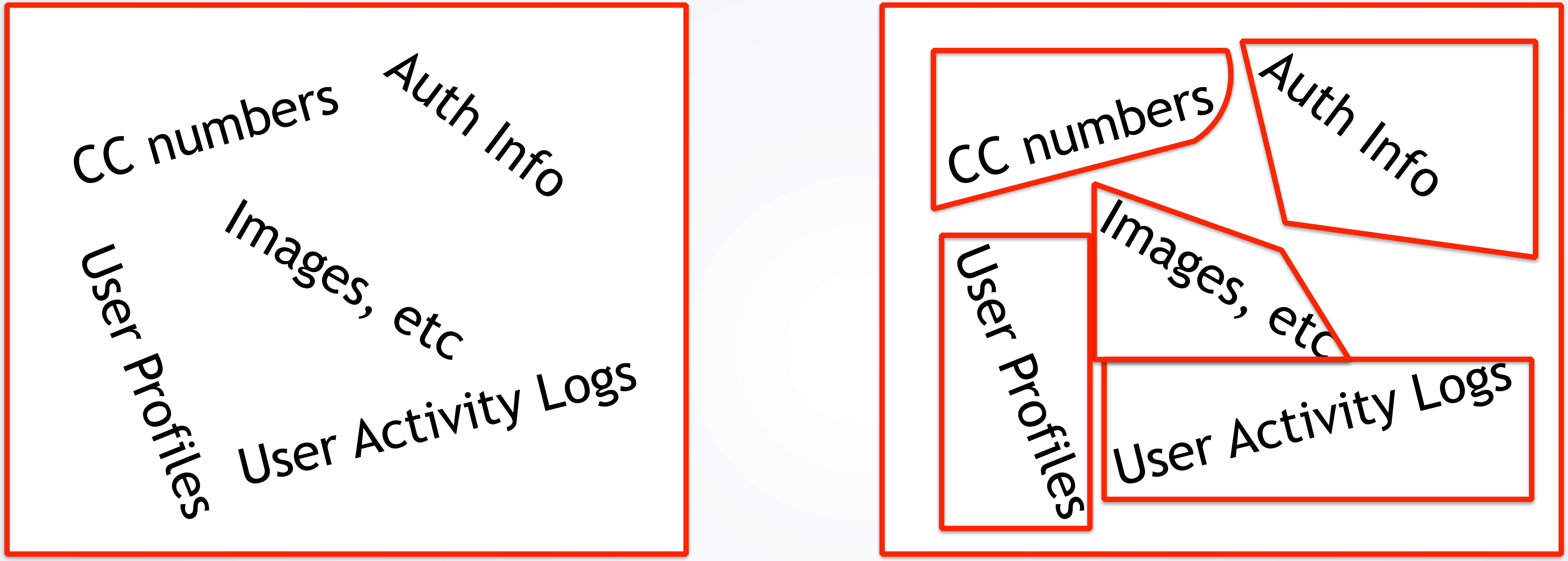
Images, etc

User Profiles

User Activity Logs

- Suppose a vulnerability exists: what is the damage?

# Defense In Depth



- Idea: Assume one layer of security might fail
  - Multiple layers of security
  - Minimize damage if one layer is compromised

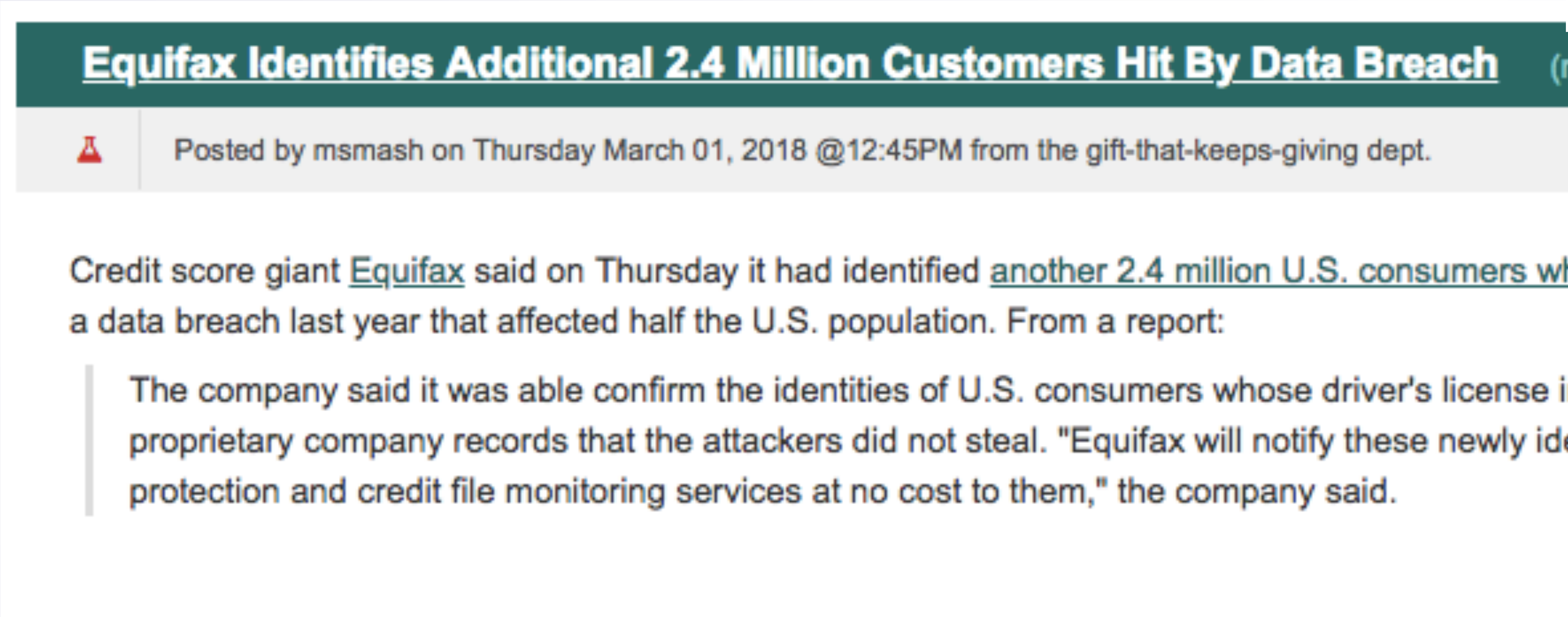
Duke UNIVERSITY

# Example of This That We Have Seen?

- What have we already seen that is an example of mitigating damage if compromised?
  - **A**: nop slide
  - **B**: CSRF token
  - **C**: Diffie-Hellman Key Exchange
  - [✓ **D**: Salt and hash passwords]

# Famous Example of NOT Defense In Depth

- Equifax got hacked
  - Bug in web library they were using
  - Many users' personal data (SSNs, etc) stolen
- Why/how?
- What should they have done?

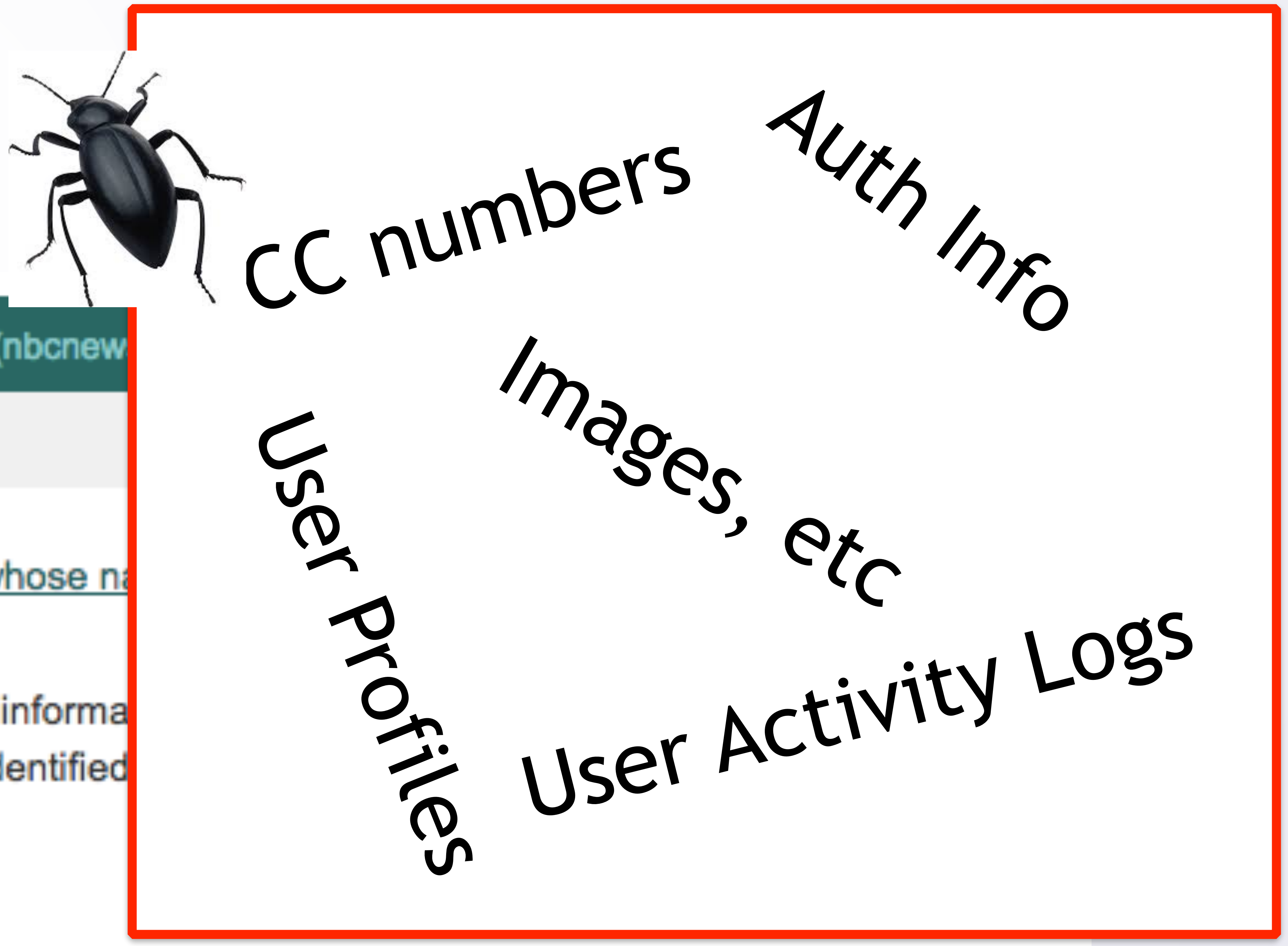


Equifax Identifies Additional 2.4 Million Customers Hit By Data Breach (nbcnew

Posted by msmash on Thursday March 01, 2018 @12:45PM from the gift-that-keeps-giving dept.

Credit score giant Equifax said on Thursday it had identified another 2.4 million U.S. consumers whose na a data breach last year that affected half the U.S. population. From a report:

> The company said it was able confirm the identities of U.S. consumers whose driver's license informa proprietary company records that the attackers did not steal. "Equifax will notify these newly identified protection and credit file monitoring services at no cost to them," the company said.



CC numbers

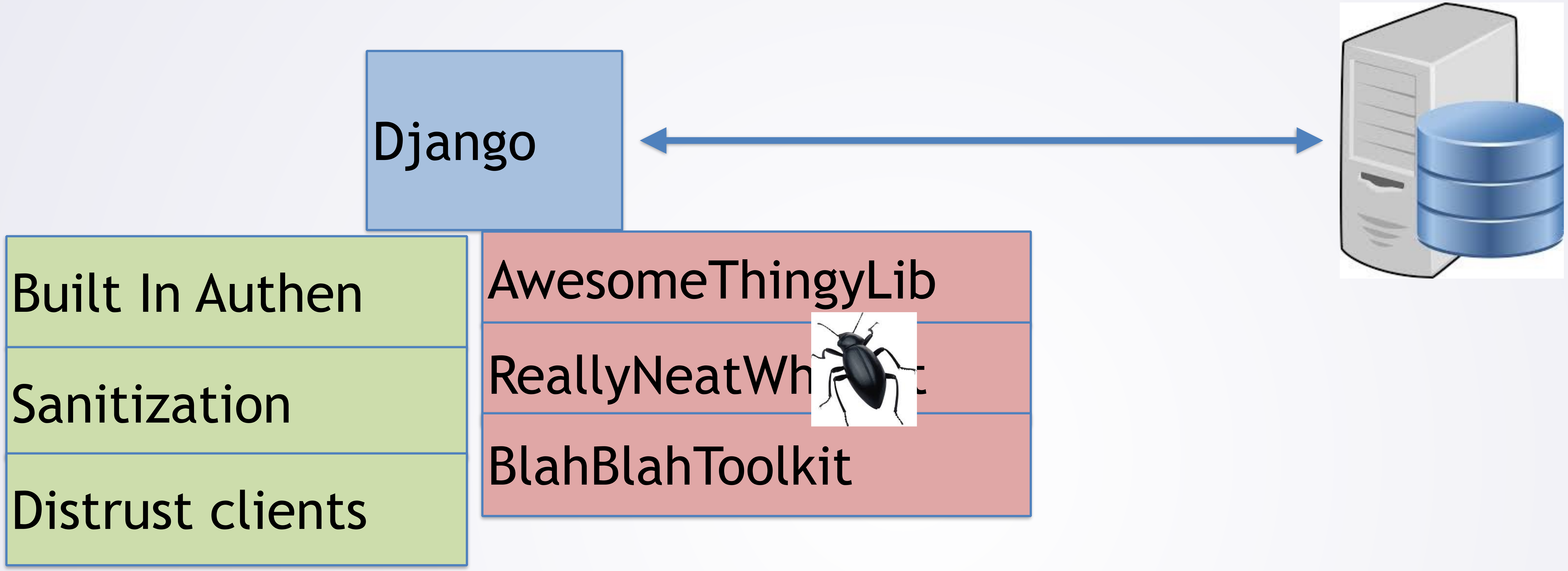
Auth Info

Images, etc

User Profiles

User Activity Logs

# Vulnerability -> Access To All Things



Django

Built In Authen

Sanitization

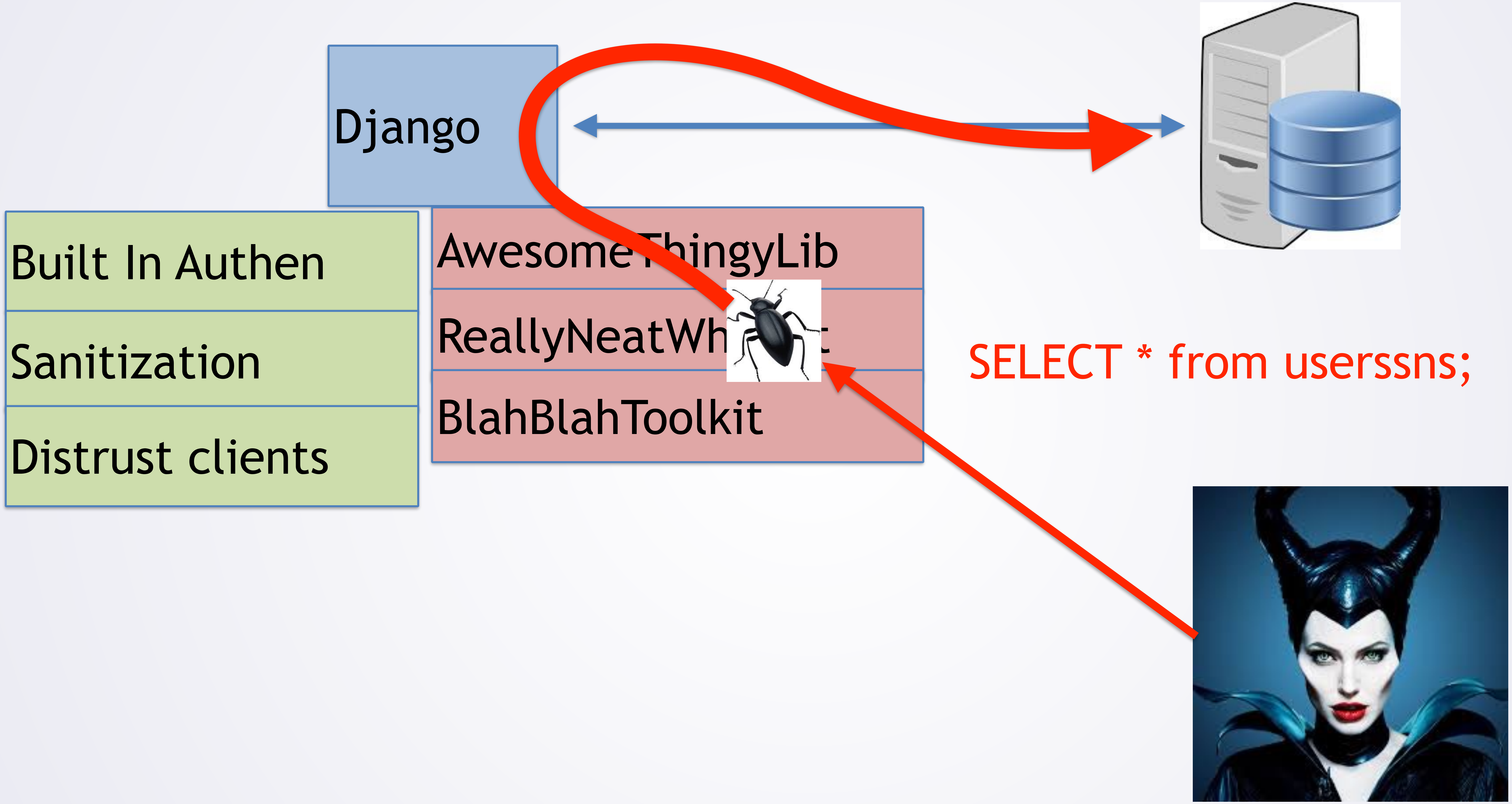
Distrust clients

AwesomeThingyLib

ReallyNeatWh t

BlahBlahToolkit

# Vulnerability -> Access To All Things



Django

Built In Authen

Sanitization

Distrust clients

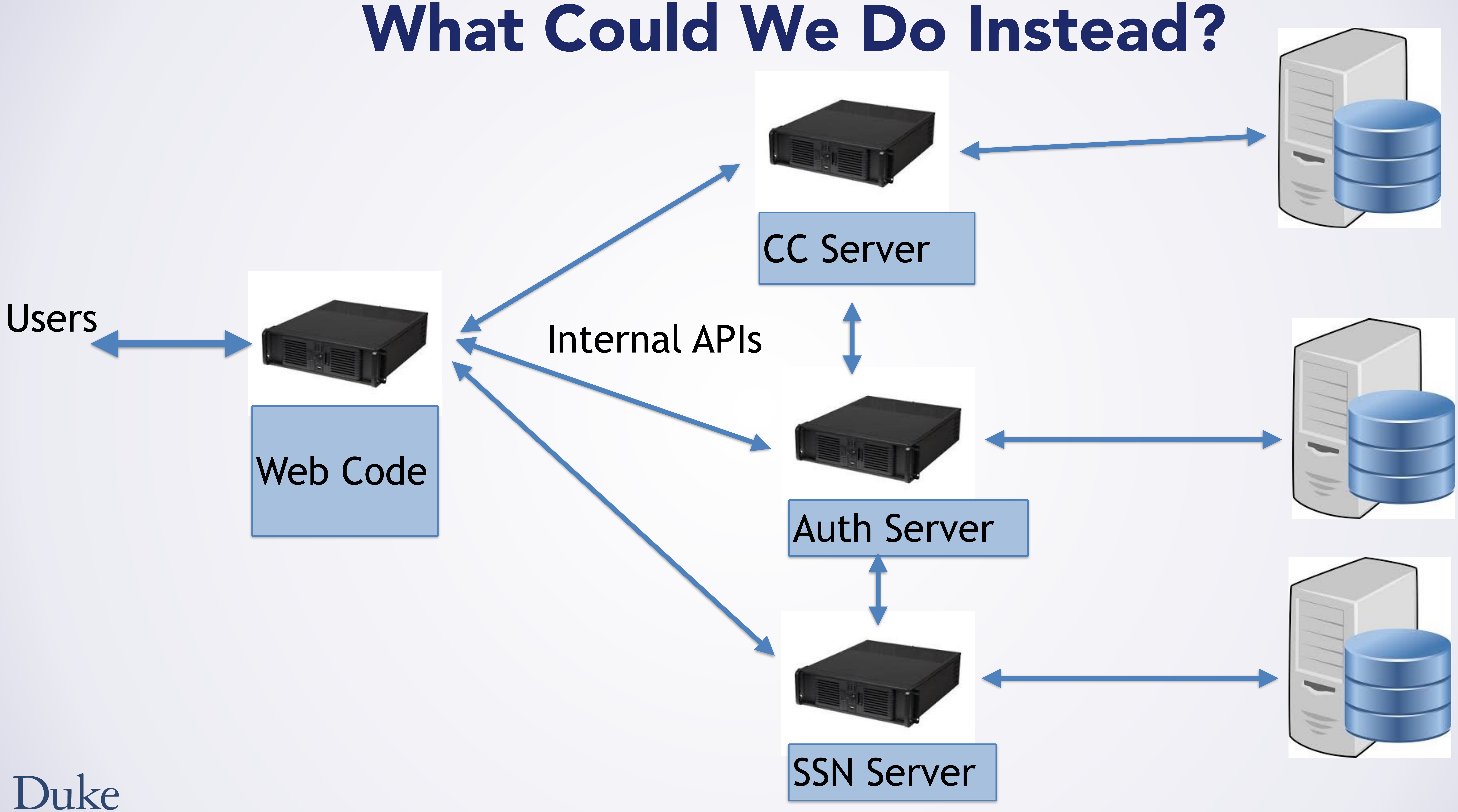
AwesomeThingyLib

ReallyNeatWh t

BlahBlahToolkit

SELECT * from userssns;

# What Could We Do Instead?

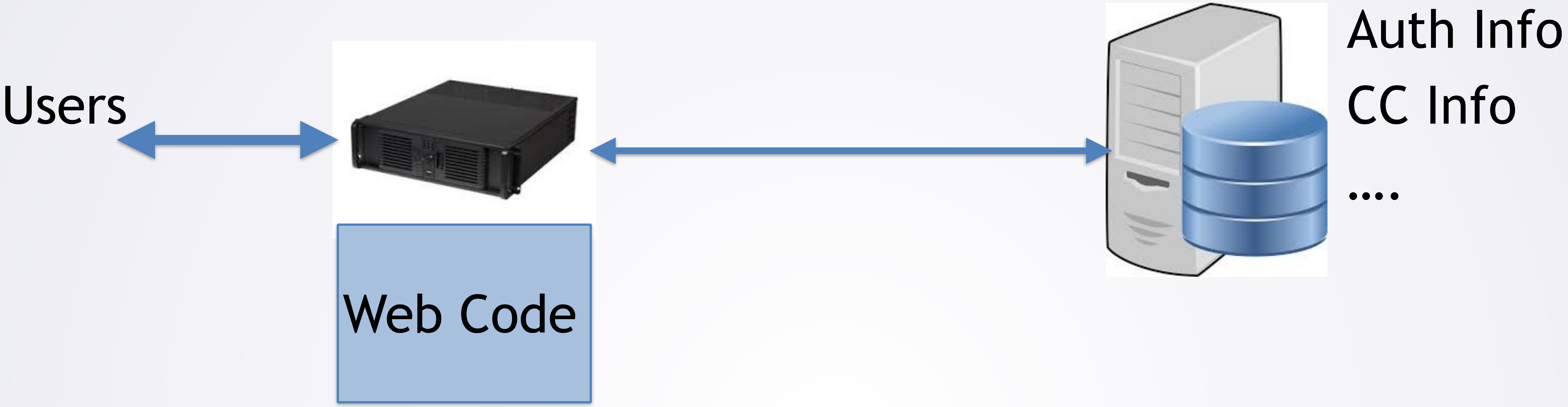


# What Could We Do Instead?

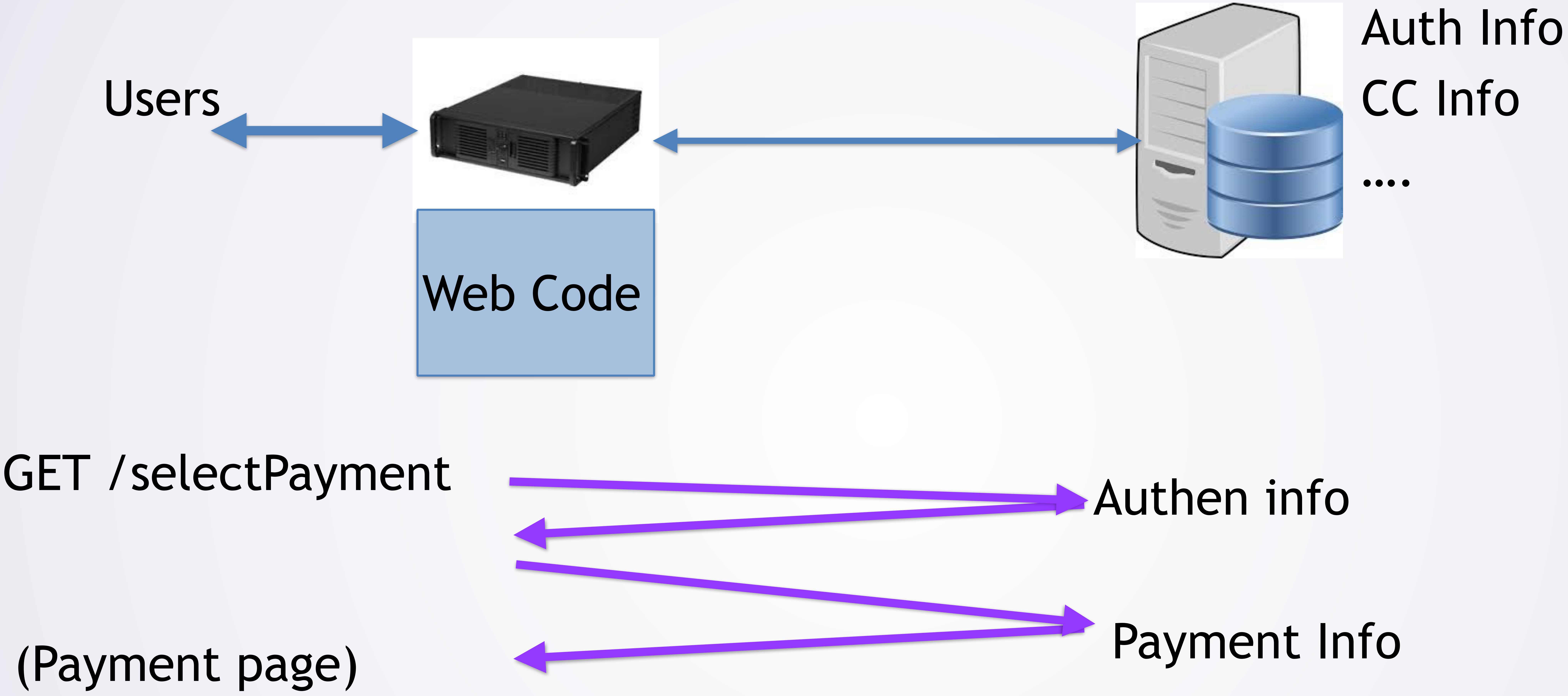
+Firewalls
Users
Web Code
Internal APIs
CC Server
Auth Server
SSN Server

Duke University

# Contrast: Make a Purchase



# Contrast: Make a Purchase



Users

Web Code

Auth Info

CC Info

….

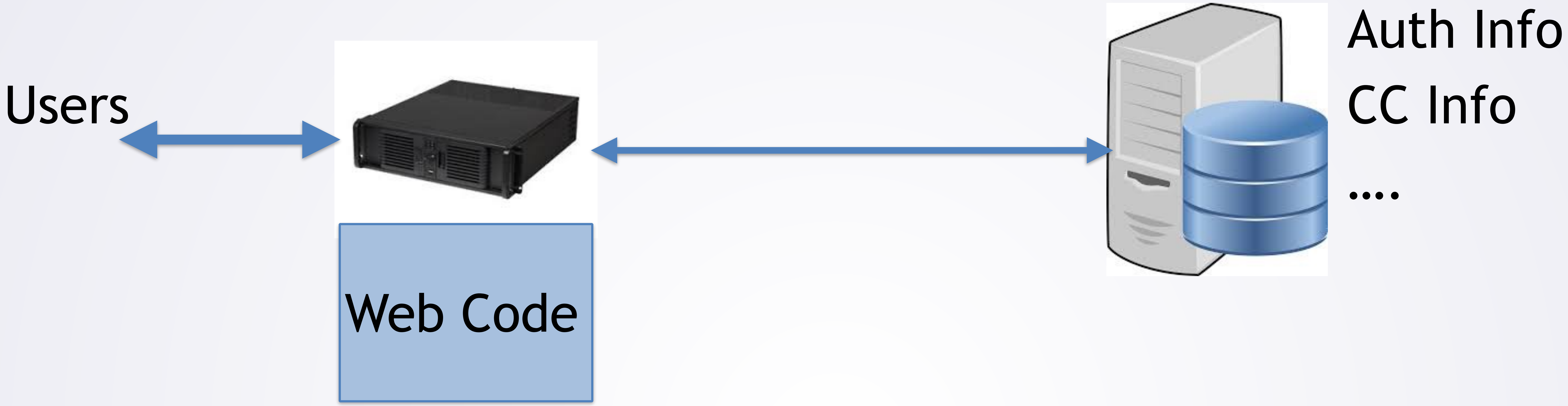
GET /selectPayment

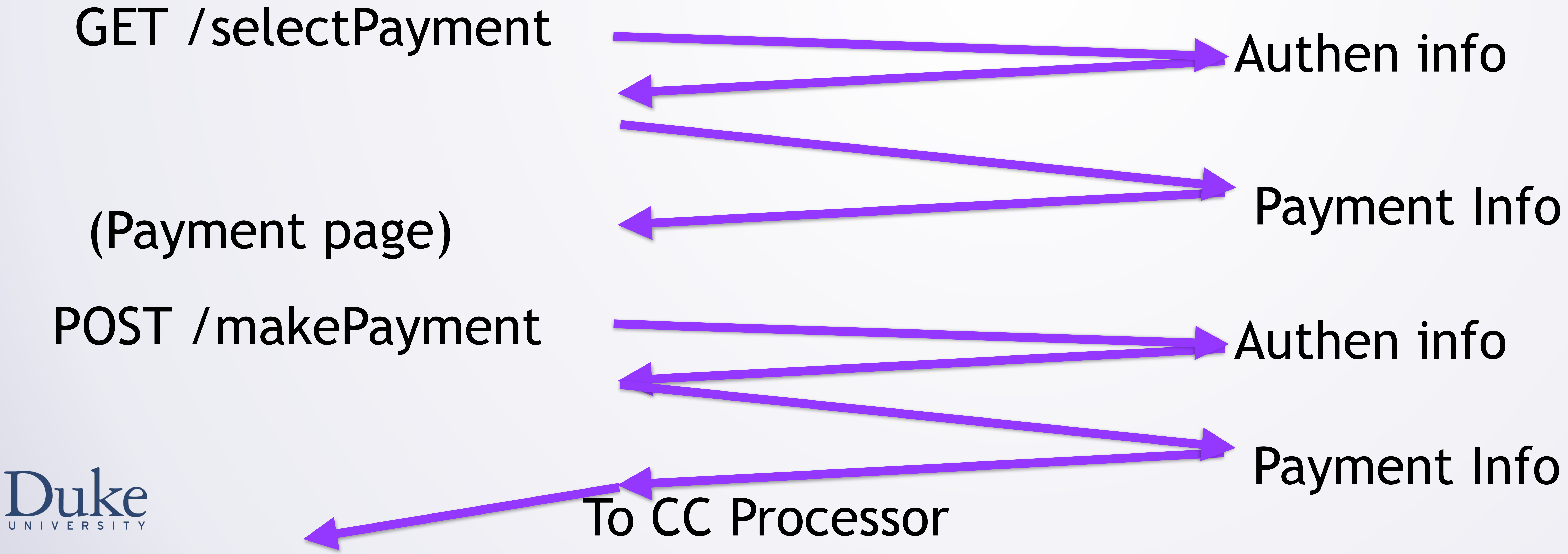
Authen info

Payment Info

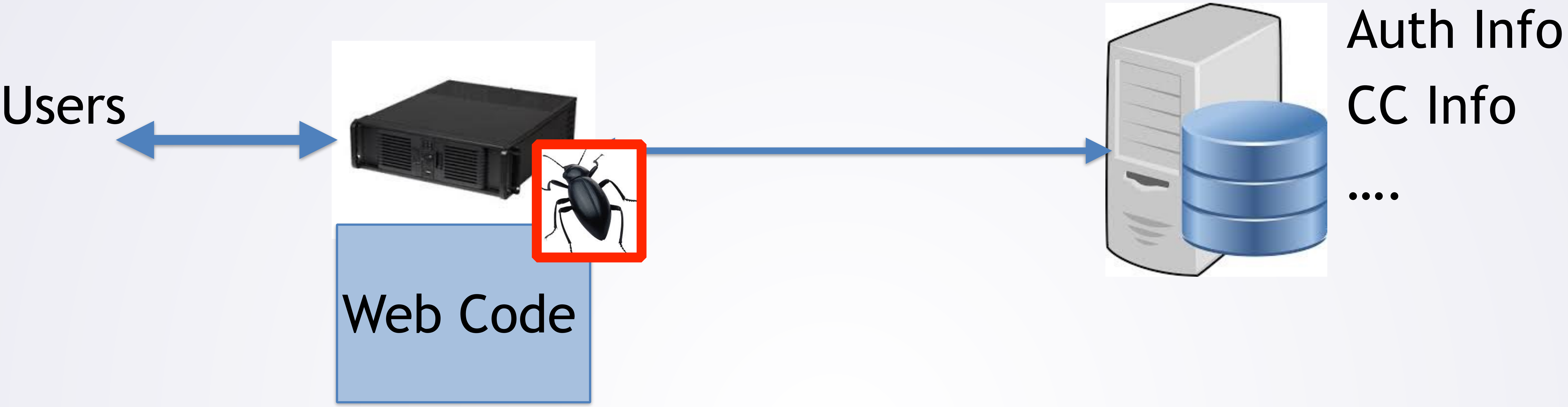
(Payment page)

# Contrast: Make a Purchase





# Contrast: Make a Purchase



Users

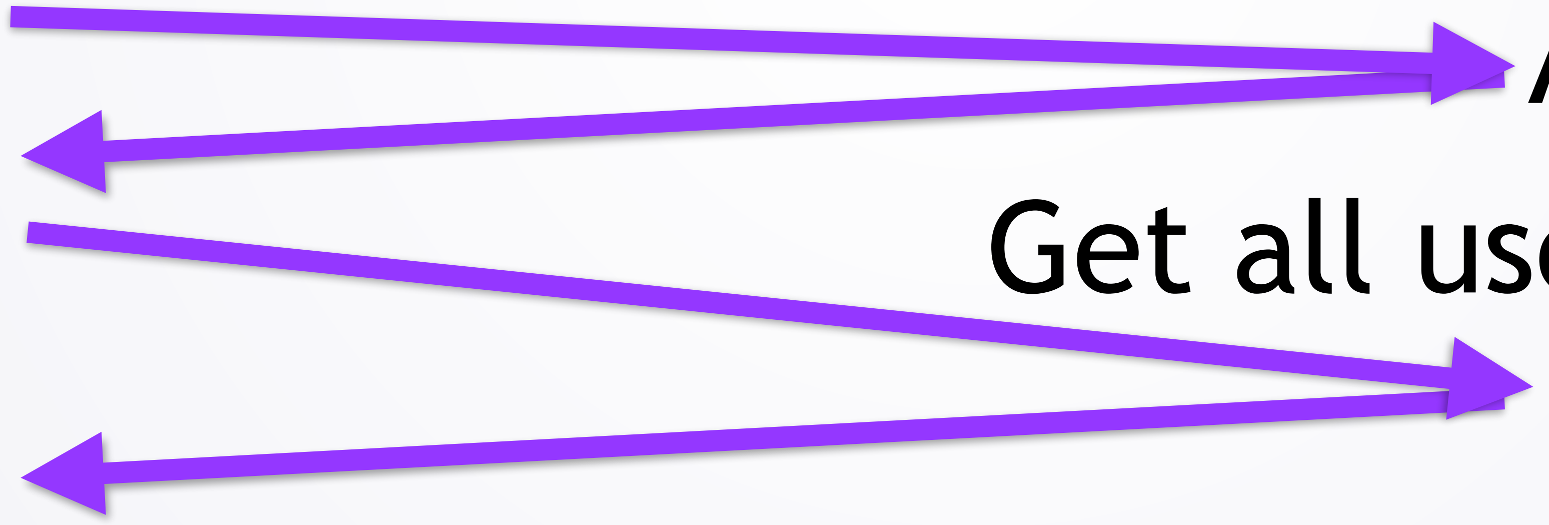
Web Code

Auth Info

CC Info

….

GET /selectPayment



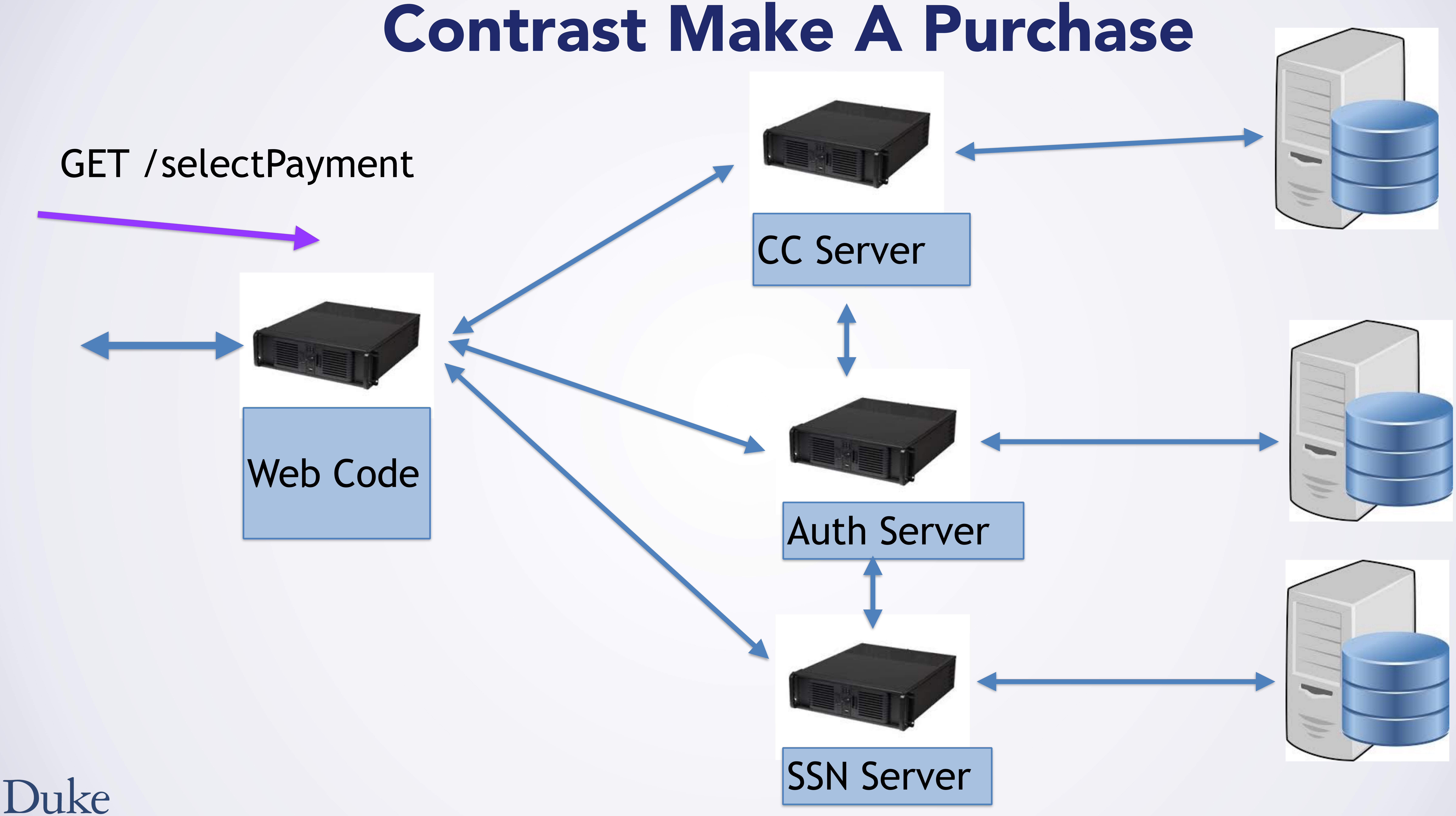
Authen info

Get all user's info

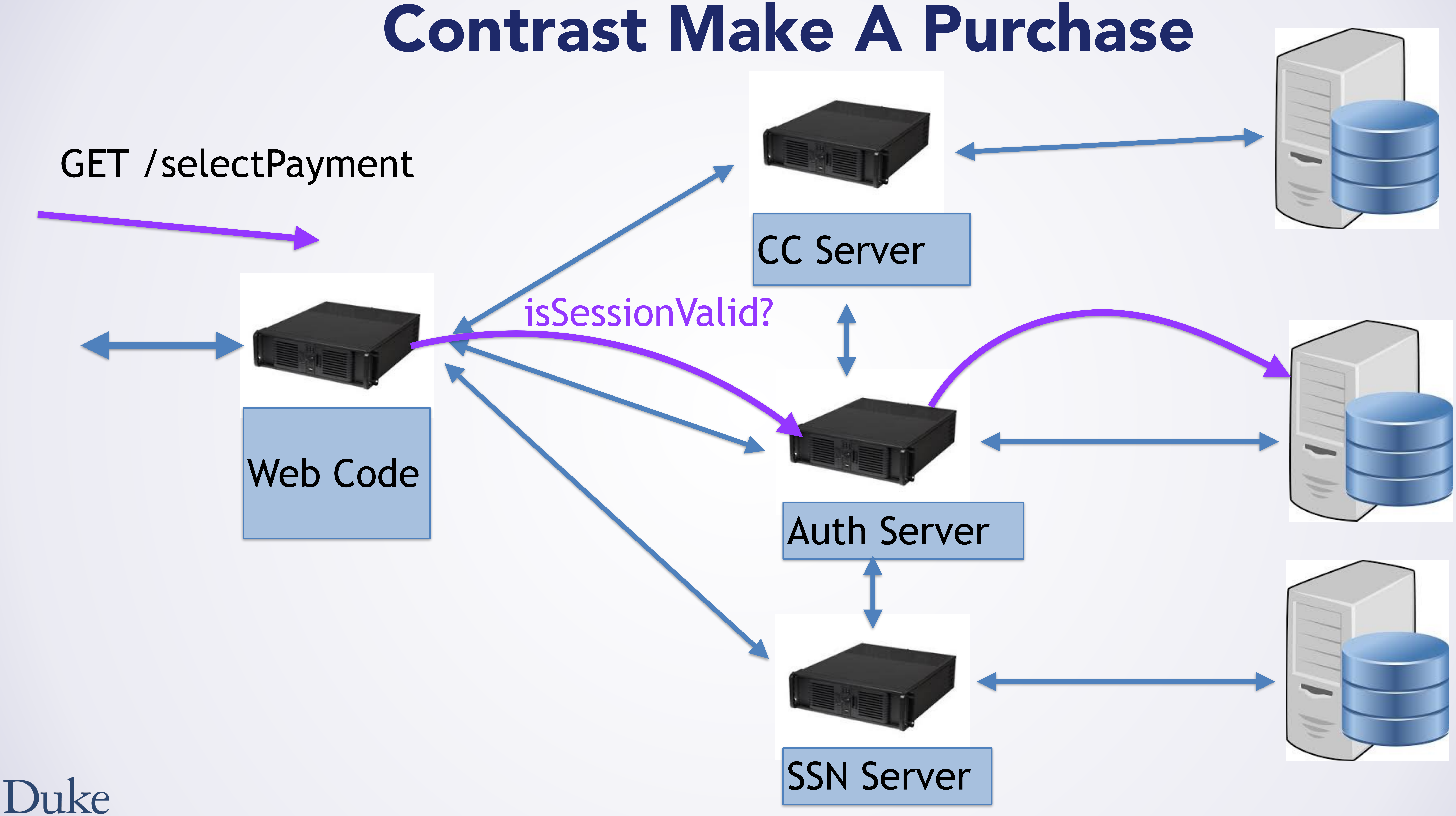
(all the info)

(everyone's
credit card #s
hashed pwds
SSNs…)

# Contrast Make A Purchase

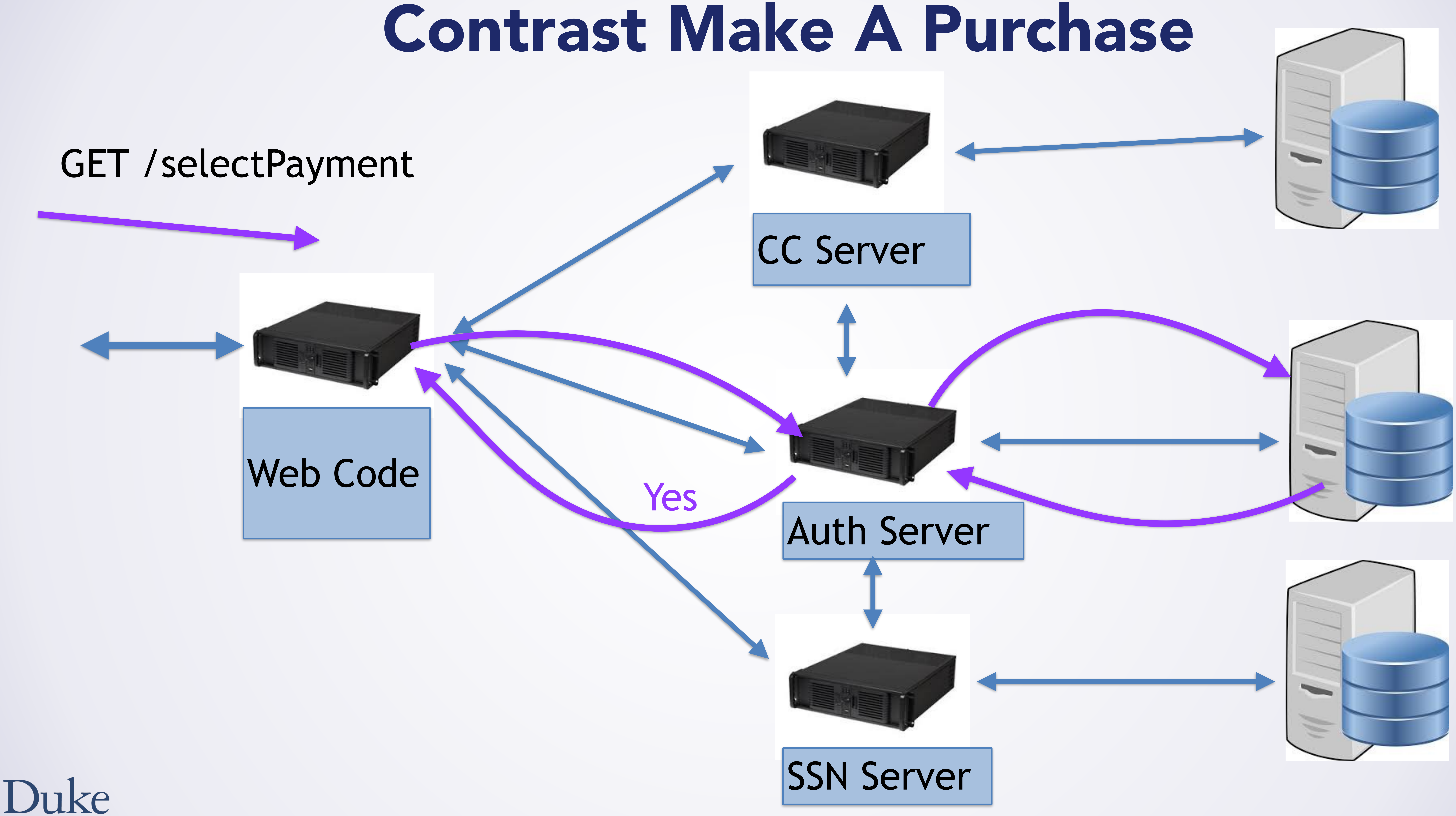


# Contrast Make A Purchase

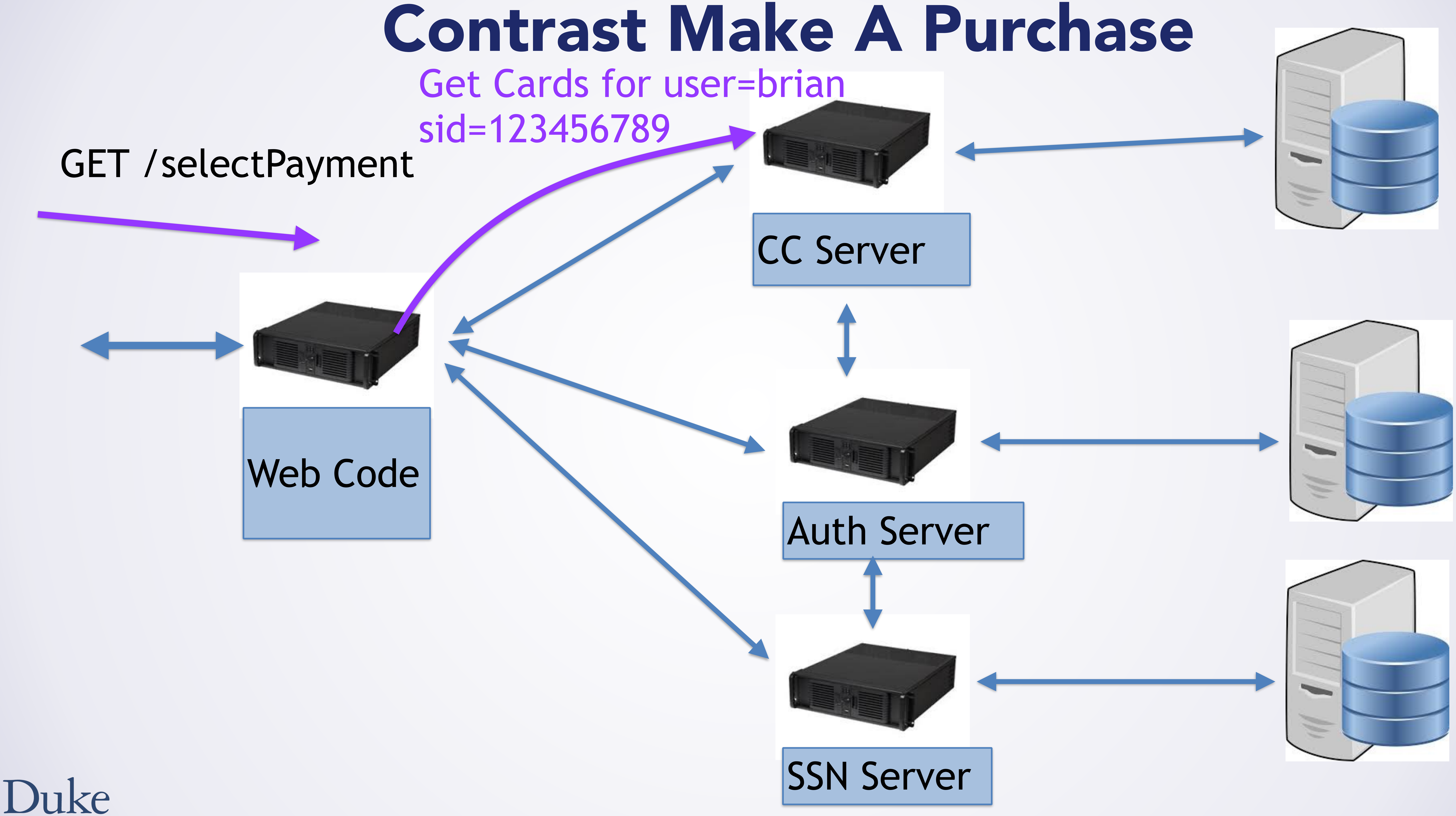


Duke UNIVERSITY

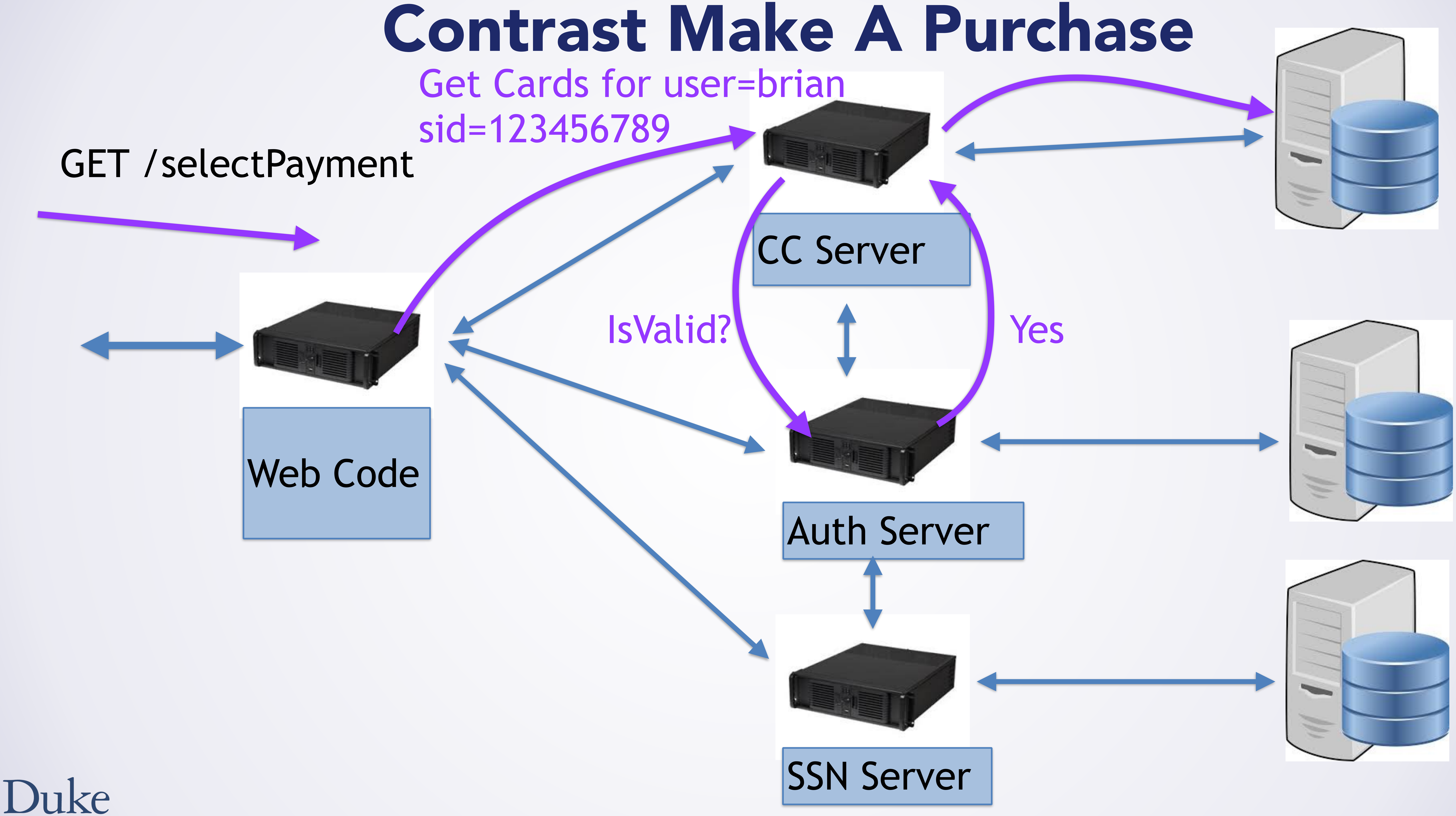
# Contrast Make A Purchase



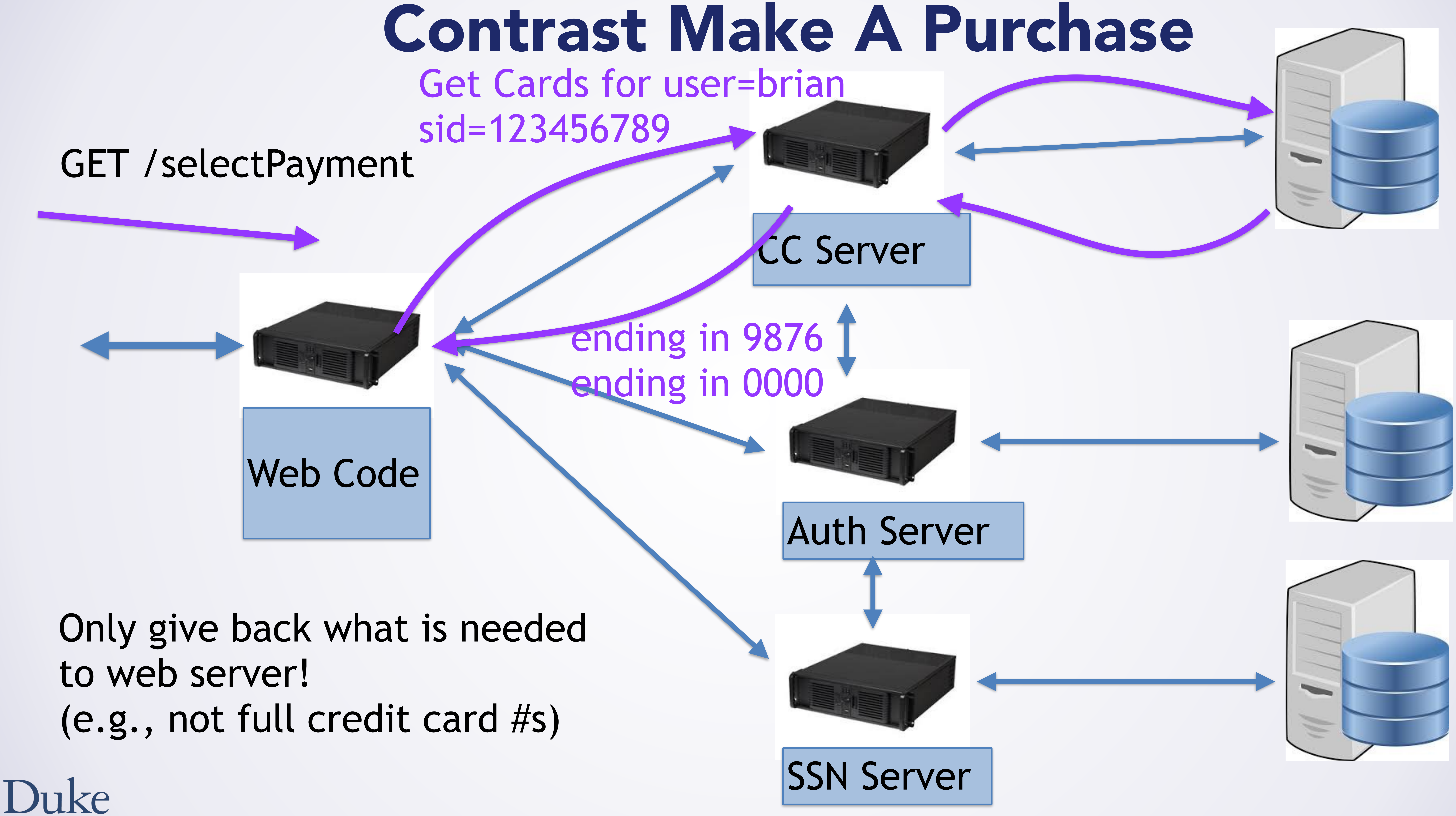
# Contrast Make A Purchase



# Contrast Make A Purchase



# Contrast Make A Purchase



Get Cards for user=brian
sid=123456789

GET /selectPayment

CC Server

ending in 9876
ending in 0000

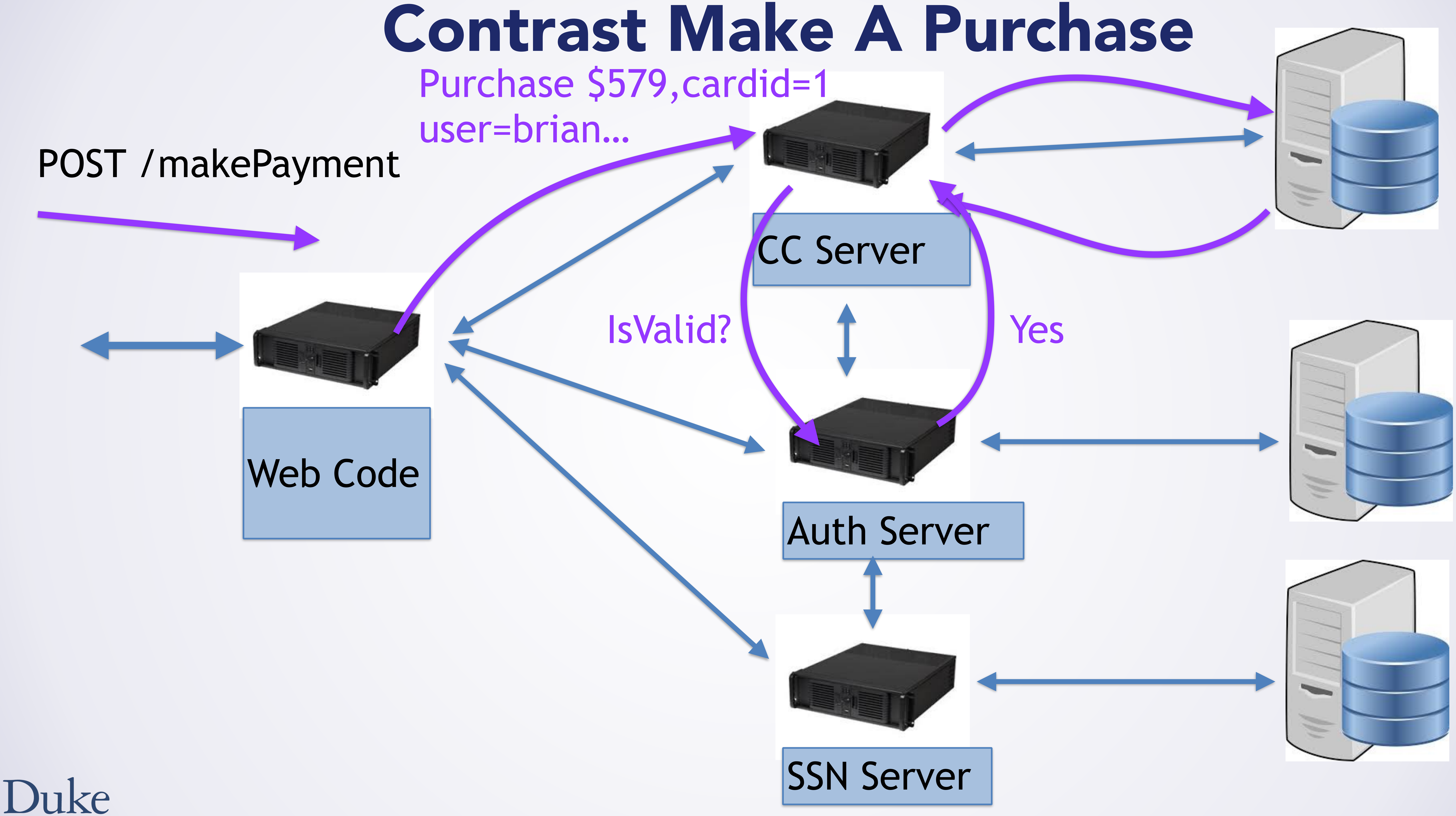
Web Code

Auth Server

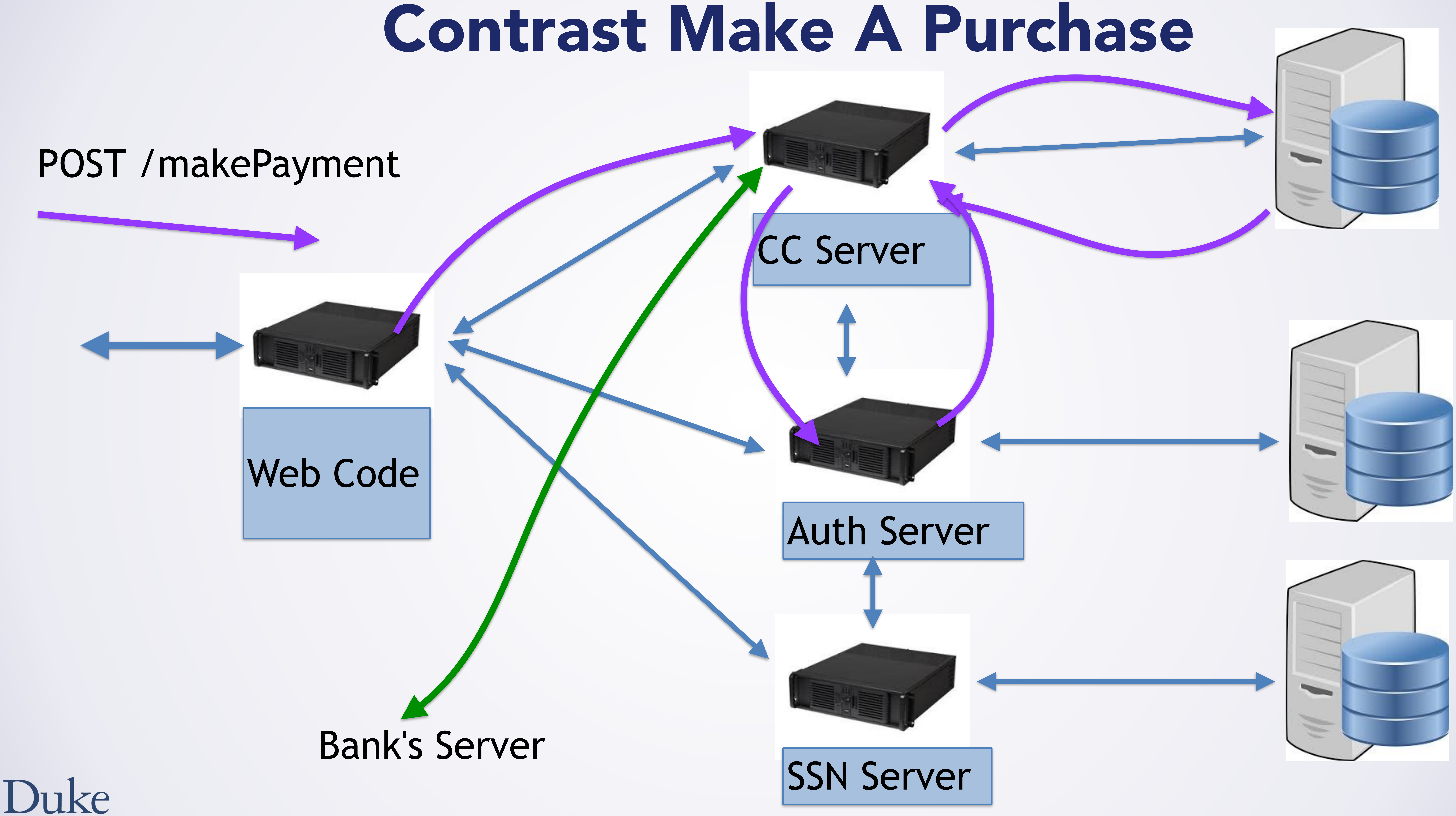
SSN Server

Only give back what is needed to web server!
(e.g., not full credit card #s)

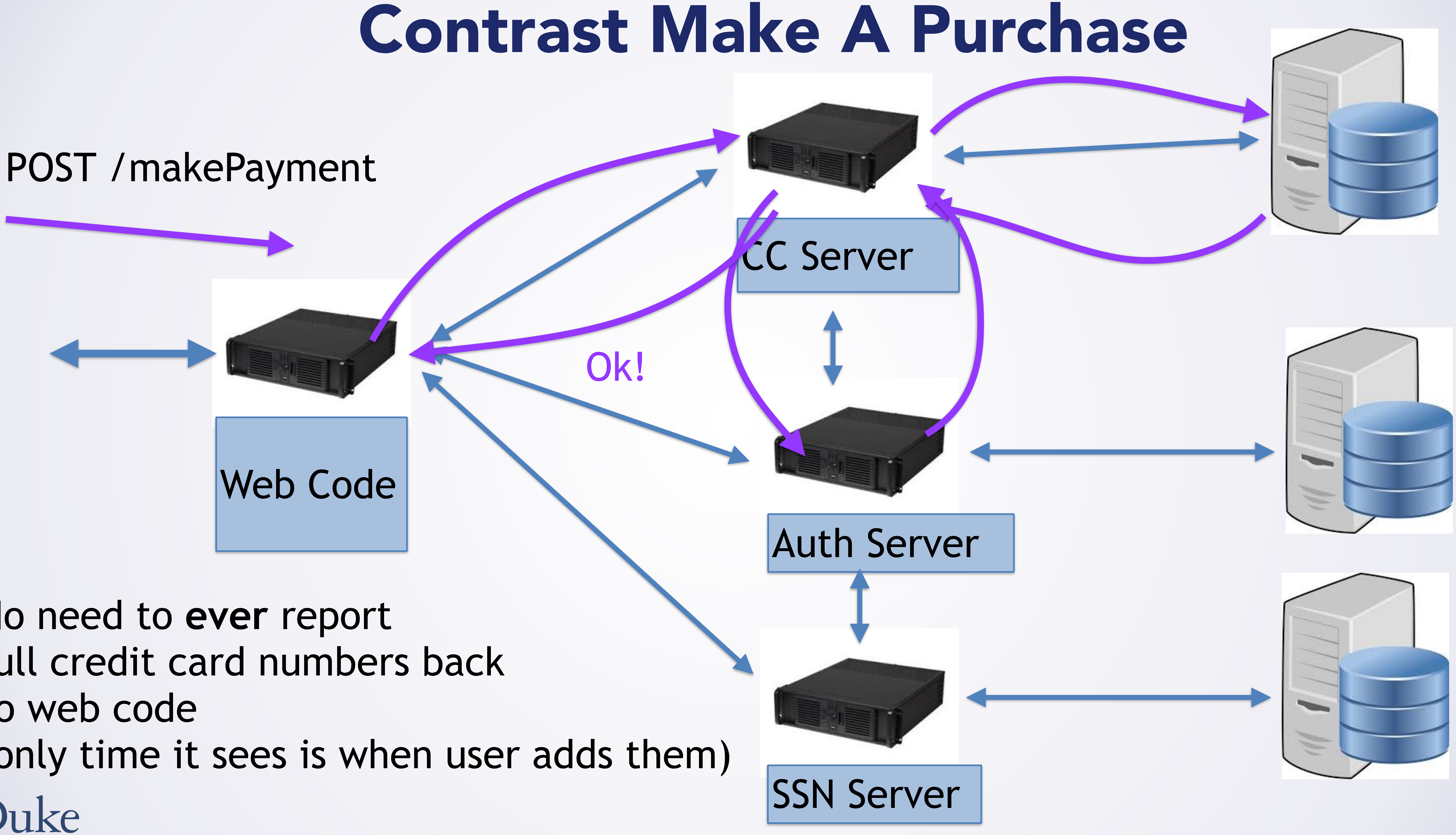
# Contrast Make A Purchase



# Contrast Make A Purchase



# Contrast Make A Purchase



POST /makePayment

Web Code

CC Server

Ok!

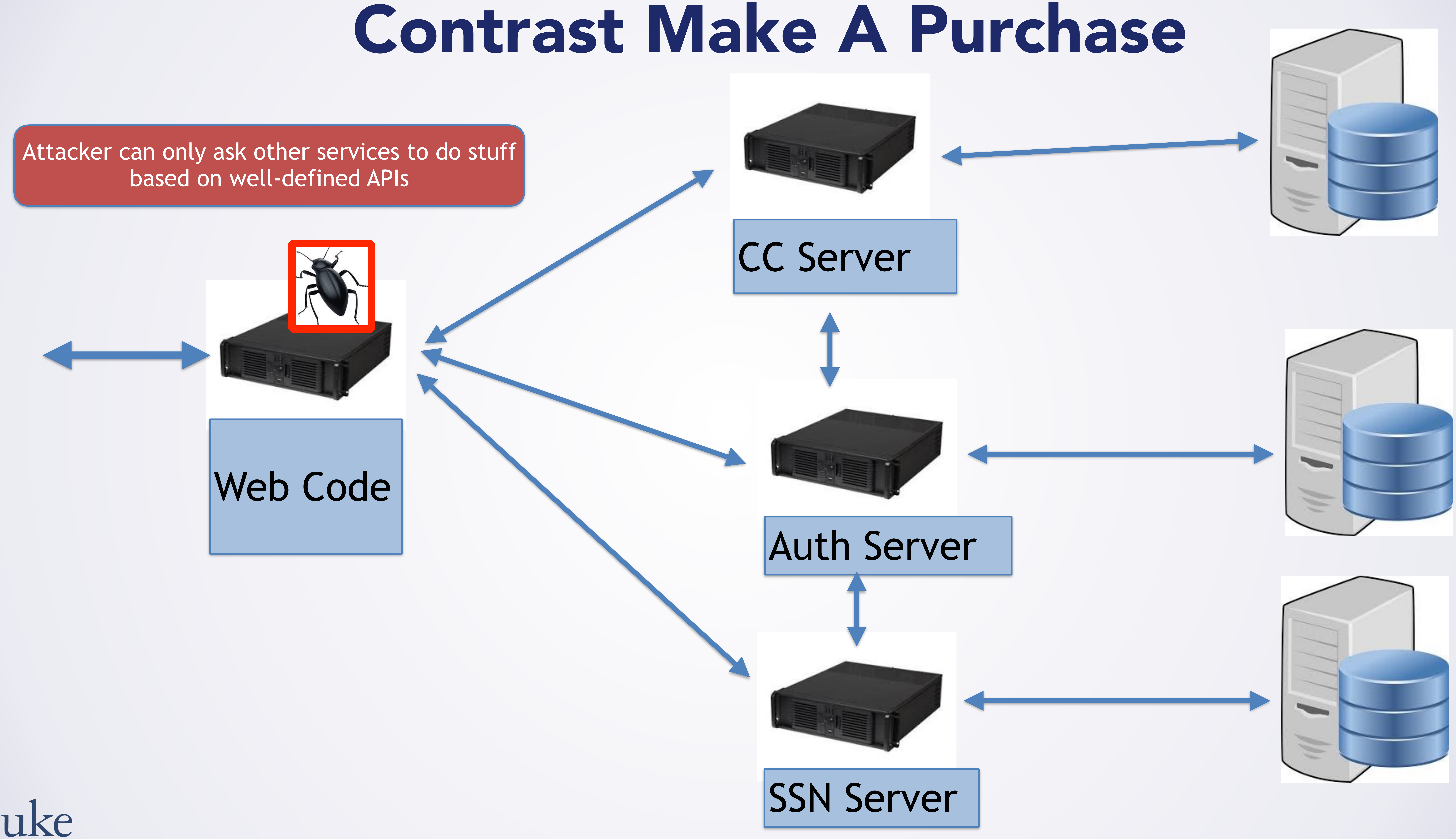
Auth Server

SSN Server

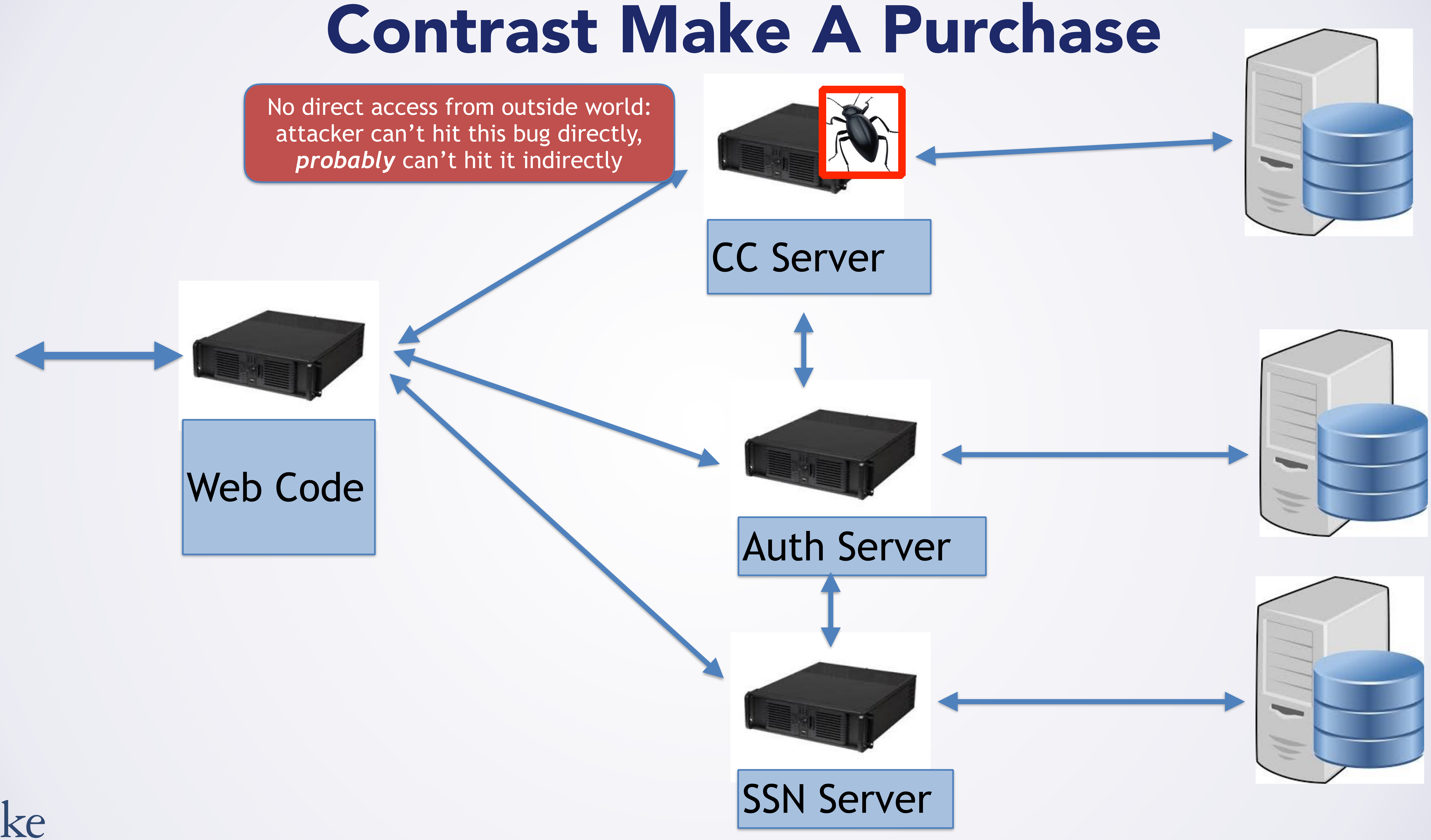
No need to **ever** report
full credit card numbers back
to web code
(only time it sees is when user adds them)

# Contrast Make A Purchase

Attacker can only ask other services to do stuff based on well-defined APIs

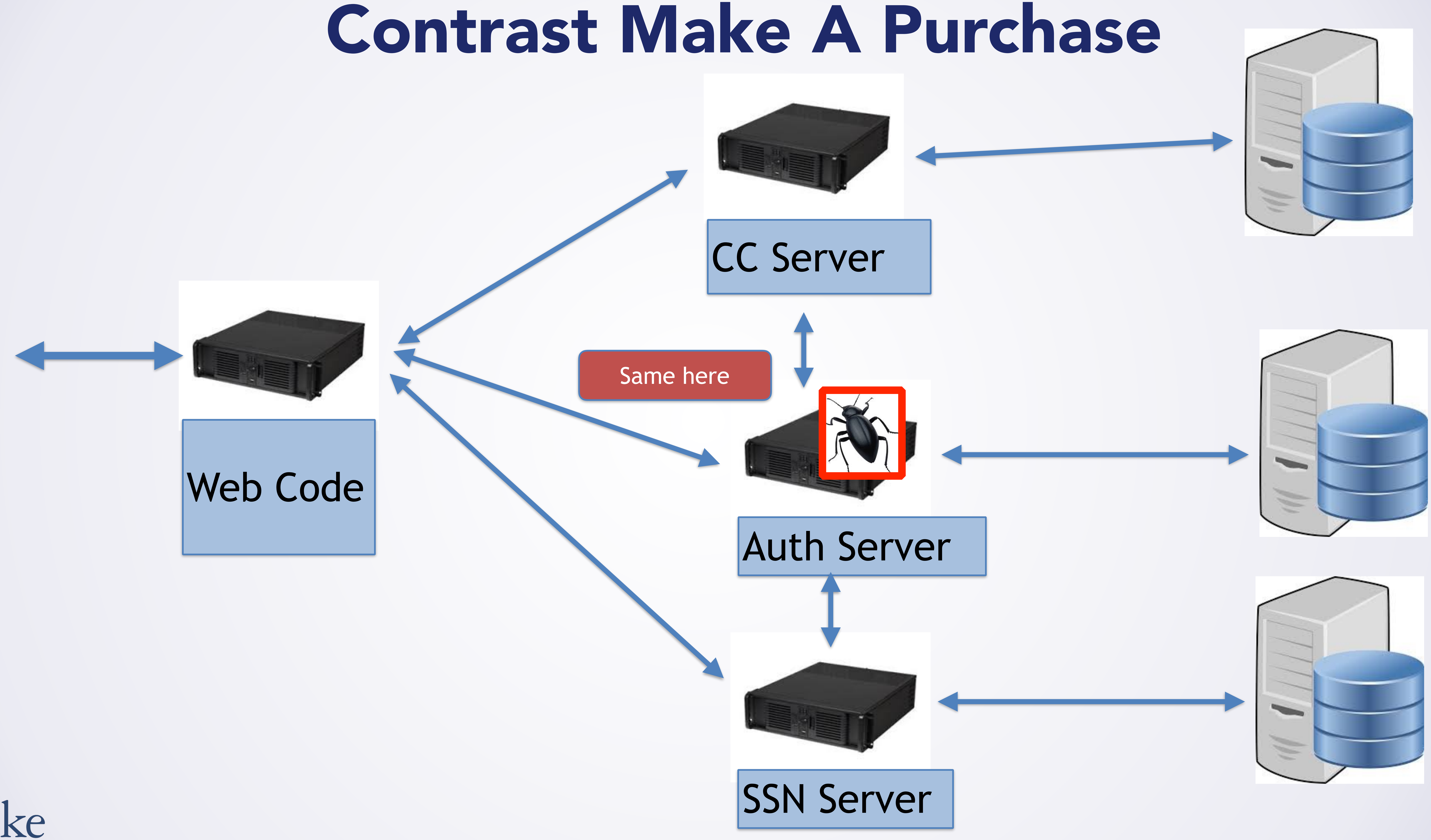


# Contrast Make A Purchase



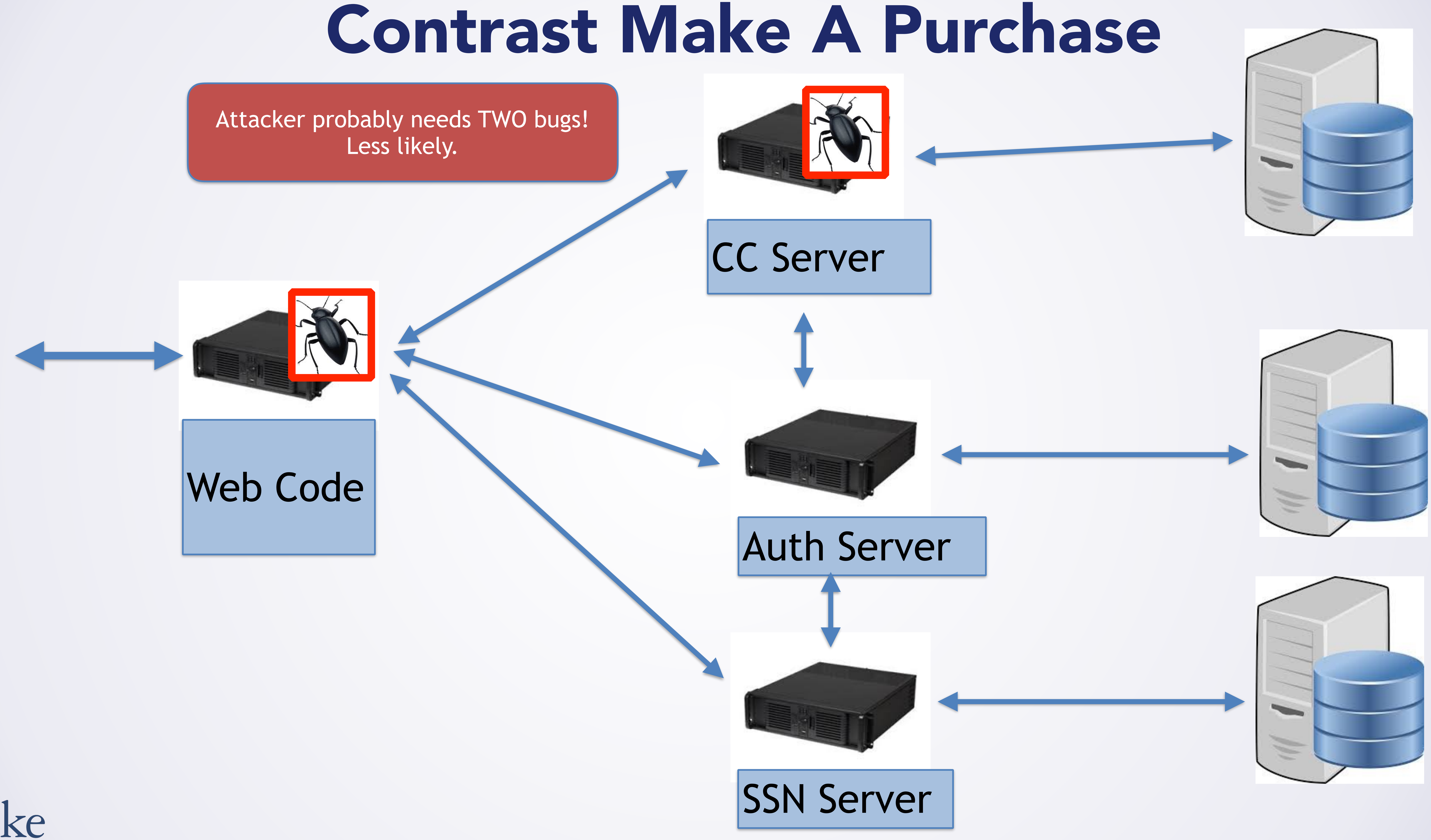
No direct access from outside world: attacker can't hit this bug directly, ***probably*** can't hit it indirectly

# Contrast Make A Purchase



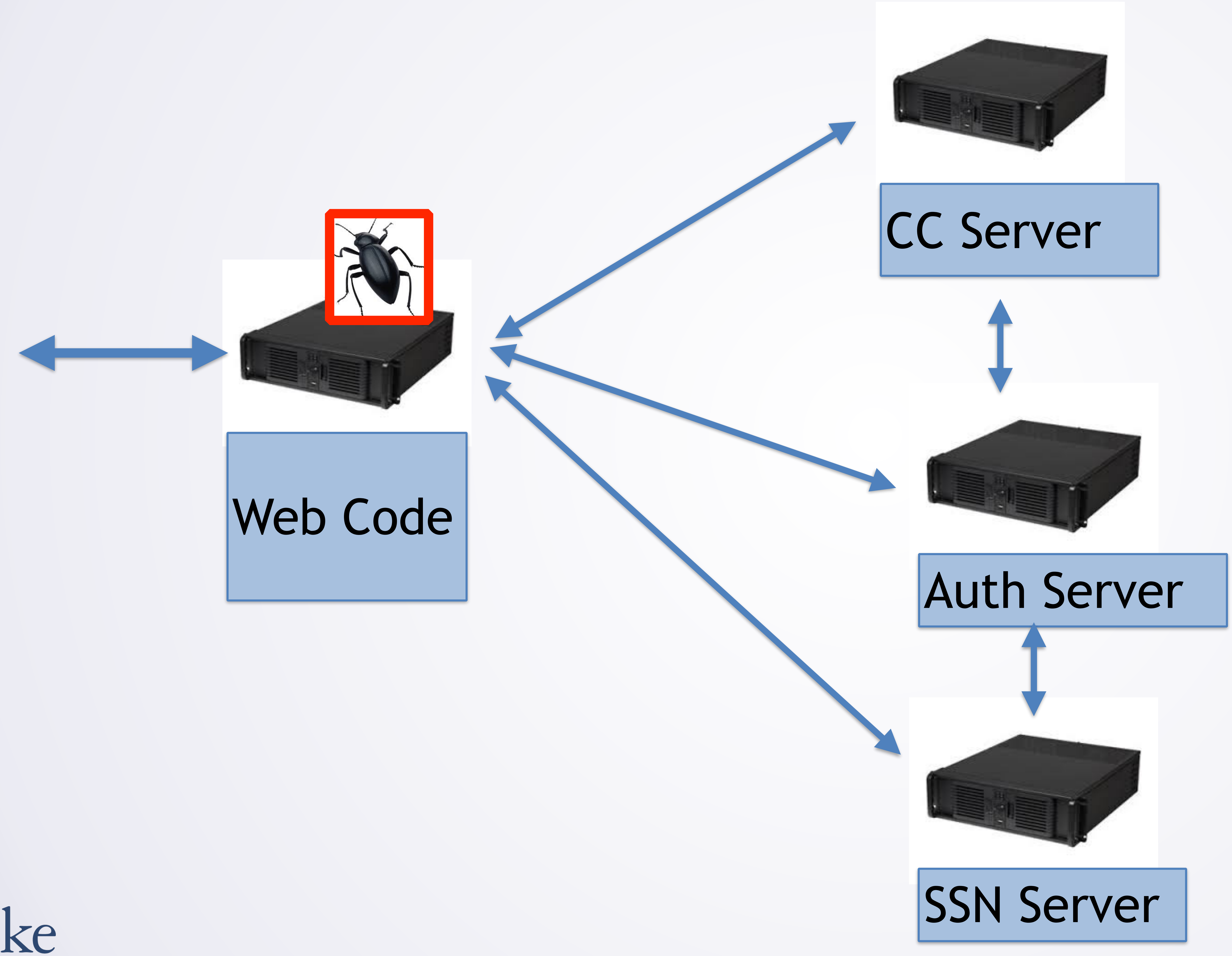
Duke UNIVERSITY

# Contrast Make A Purchase



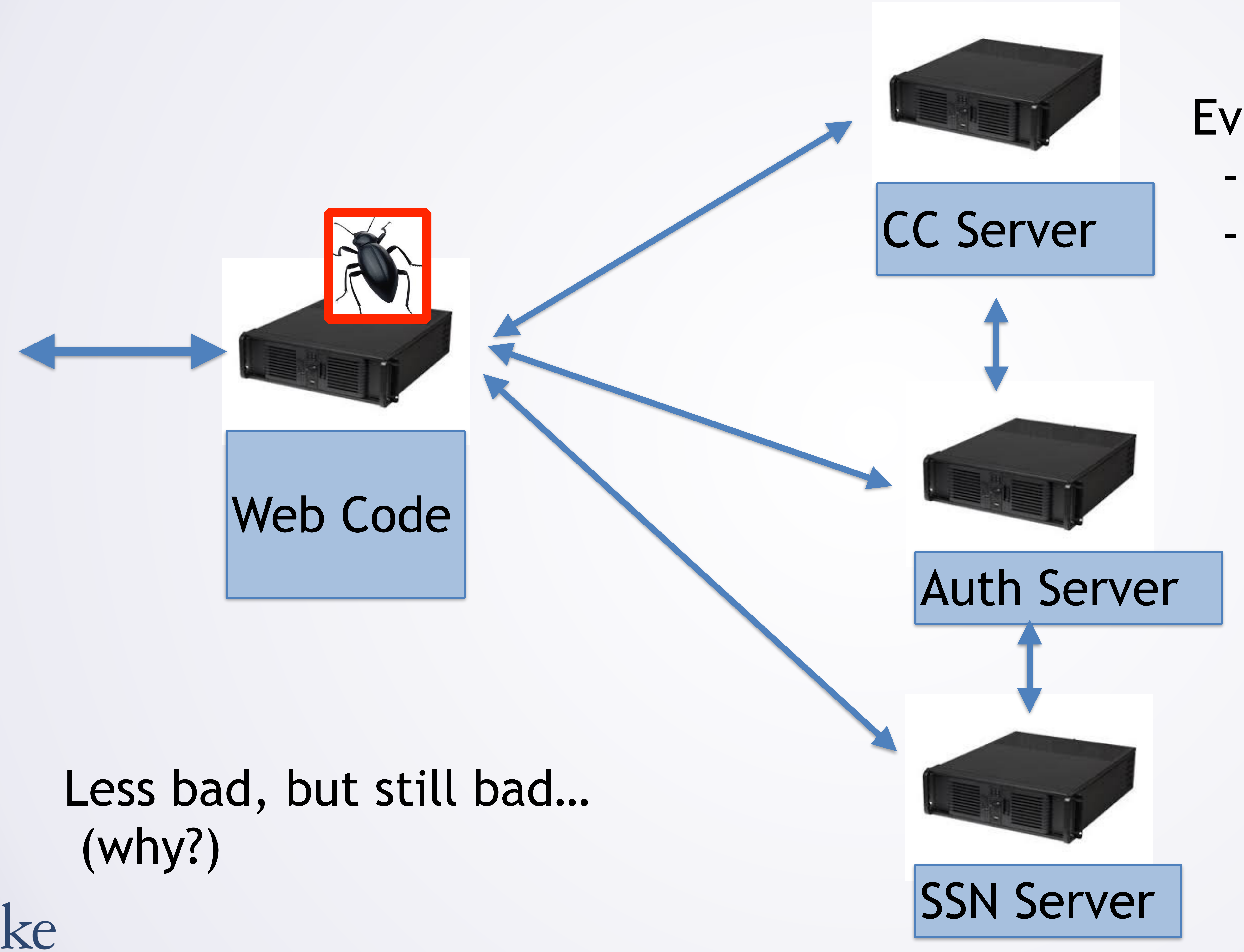
Attacker probably needs TWO bugs! Less likely.

# Let Us Revisit This



I'm going to play a longer game…

# Let Us Revisit This

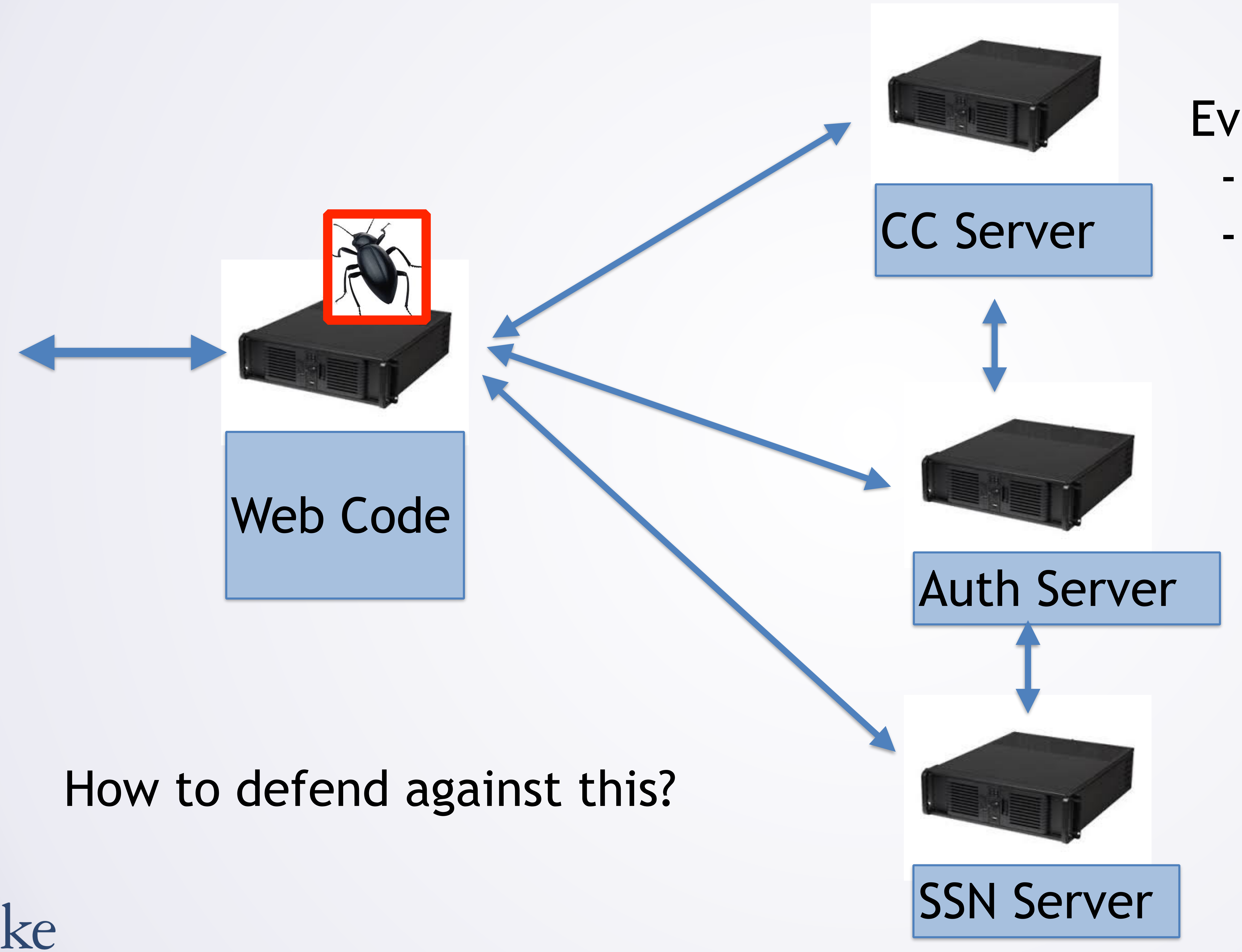


Every time someone logs in
- Get their auth info
- Send request to CC server
  Purchase something
  with their card

Less bad, but still bad…
(why?)

Duke
UNIVERSITY

# Let Us Revisit This



Every time someone logs in
- Get their auth info
- Send request to CC server
  Purchase something
  with their card

How to defend against this?

# Remember this "plan"?

- Most secure:
  - ~~Run program~~ Handle web request on a computer
  - Throw away computer
  - Buy new computer
  - ~~Run next program~~ Handle next web request on it

# Ok that plan was bad… but

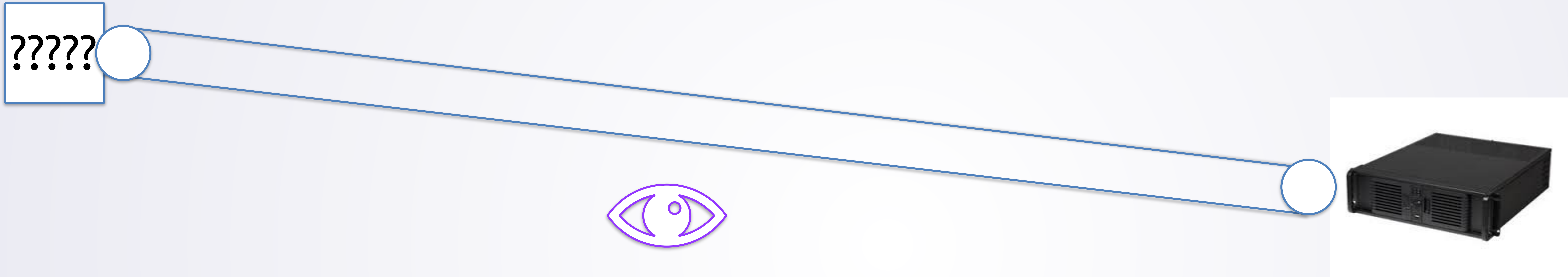
- That plan was bad, but what did we decide we could do instead?

Containers

# Prevention + Detection + Response

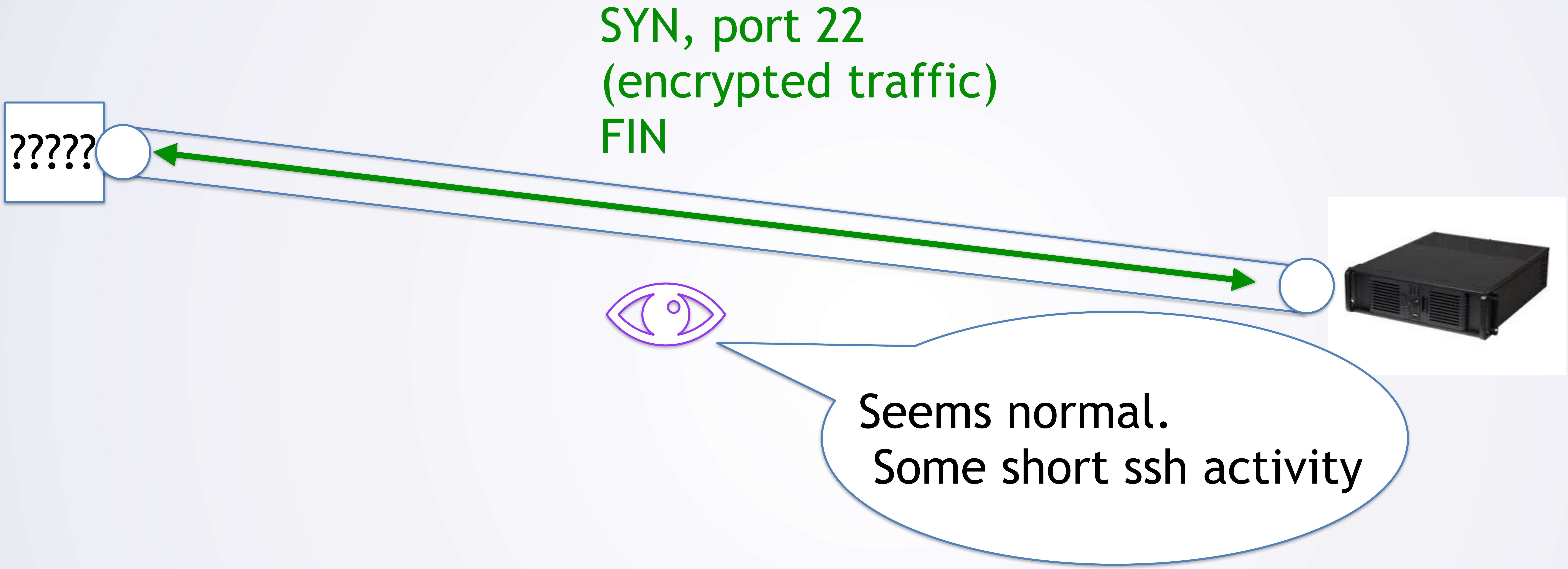
- So far have talked about **prevention**
  - Keep bad things from happening
  - Reduce badness if they do happen
- Also want **detection**
  - Know when a bad thing has happened / is happening
- …and to be able to **respond** to the attack
  - Nice if we can do something about it…

# Intrusion Detection



- Monitor system for suspicious activity

# Intrusion Detection

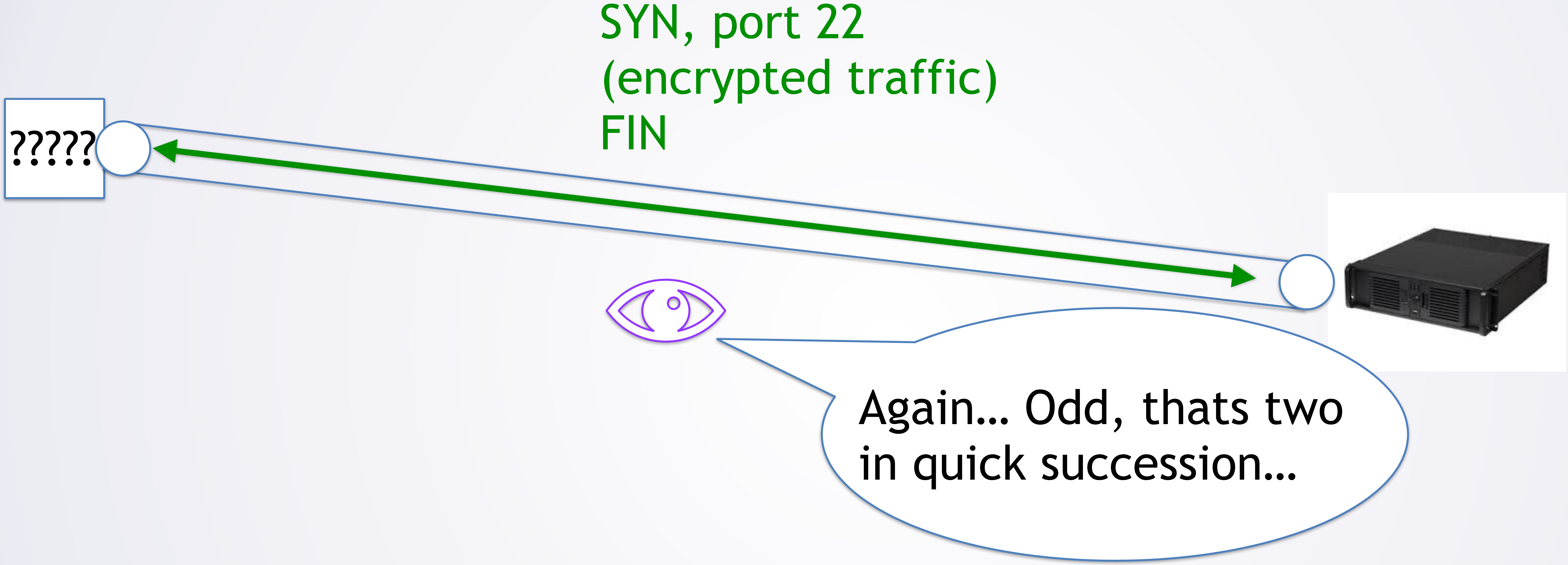


SYN, port 22
(encrypted traffic)
FIN

Seems normal.
Some short ssh activity

- Monitor system for suspicious activity

# Intrusion Detection



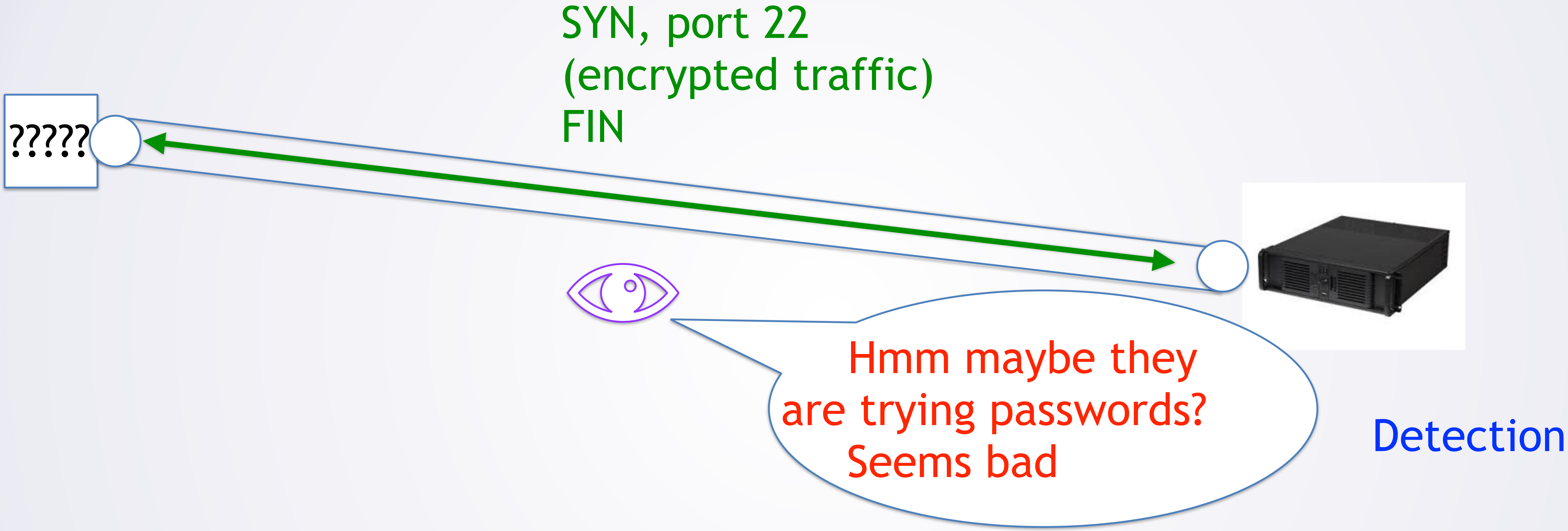
SYN, port 22
(encrypted traffic)
FIN

?????

Again... Odd, thats two in quick succession...

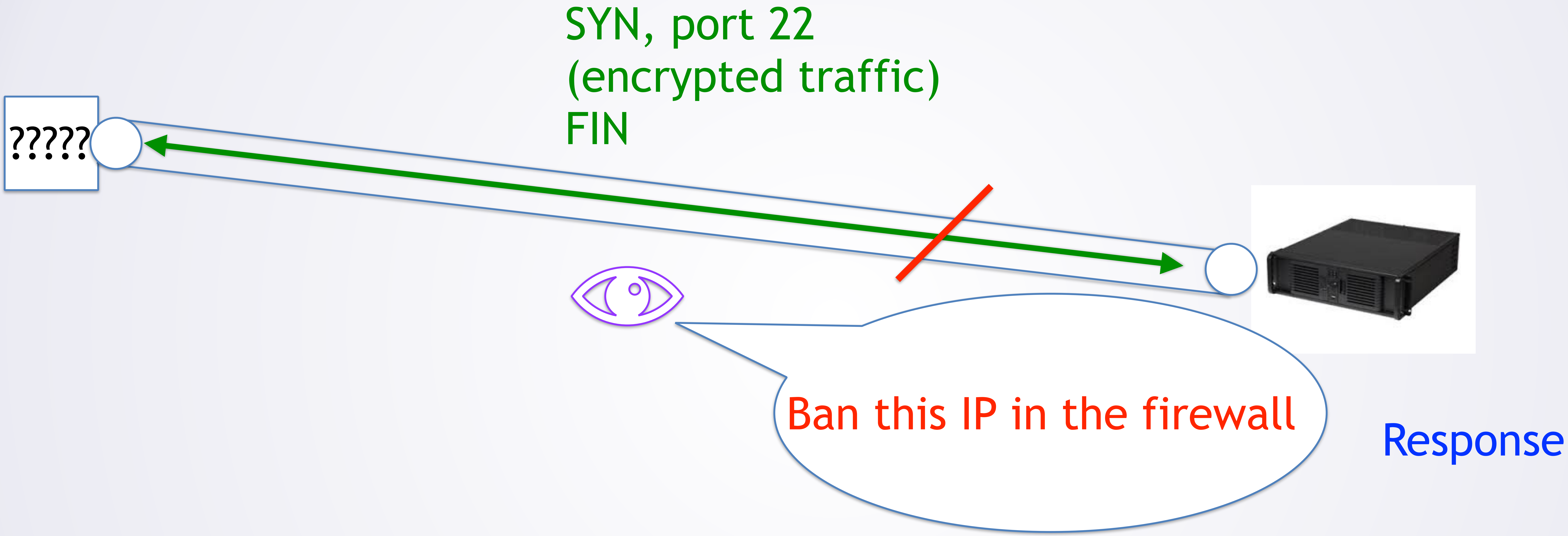
- Monitor system for suspicious activity

# Intrusion Detection



- Monitor system for suspicious activity

# Intrusion Detection



- Monitor system for suspicious activity

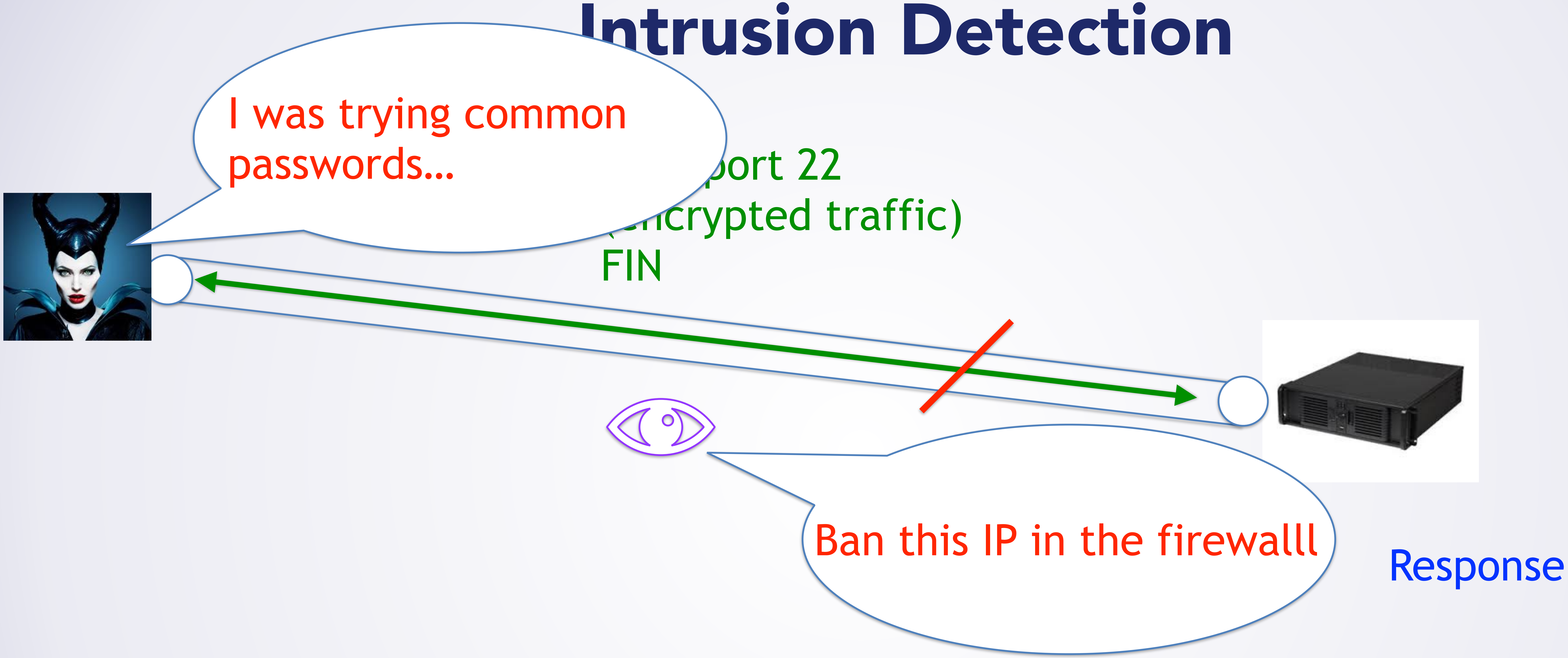
# Was this response good?

- Detected something suspicious
- Responded strongly:
  - Blocked traffic from originating site
- Good or bad?

# Was this response good?

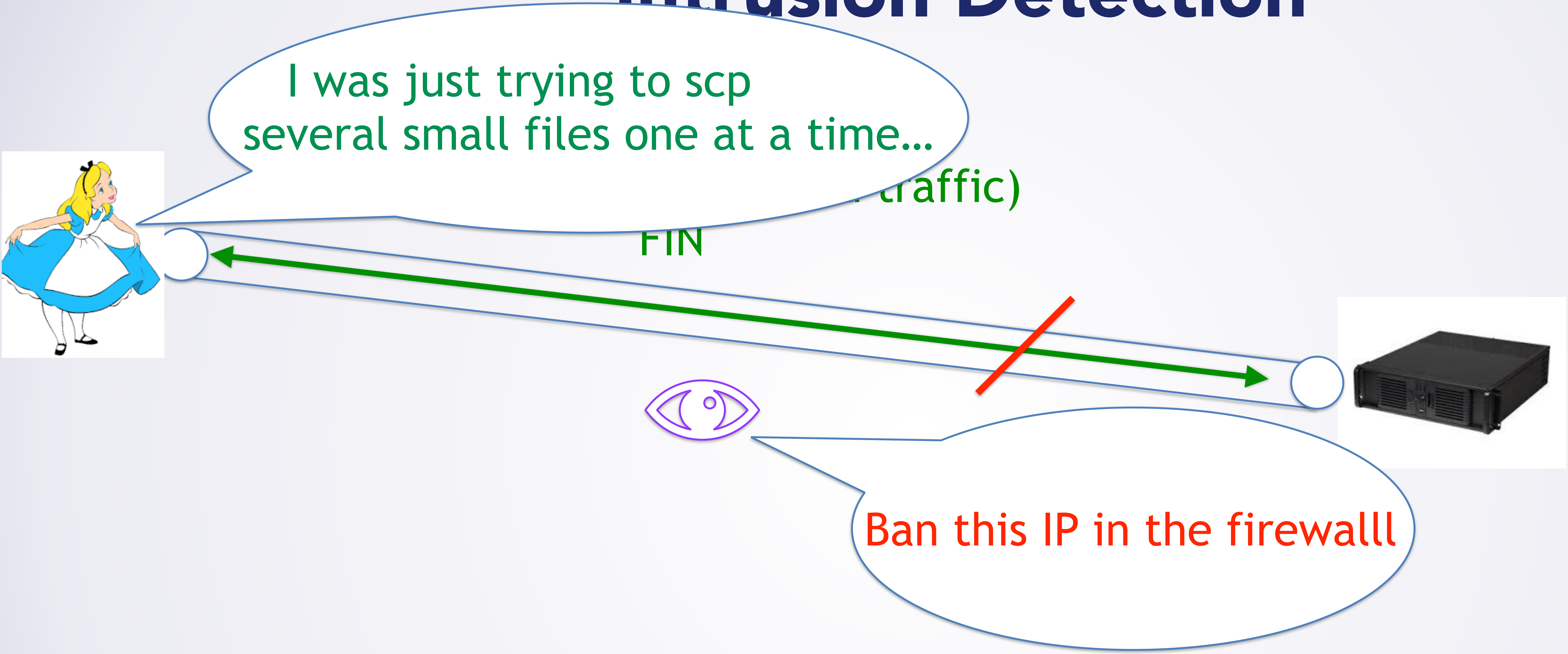
- Detected something suspicious
- Responded strongly:
  - Blocked traffic from originating site
- Good or bad?
  - **It depends!**

# Intrusion Detection


I was trying common passwords...
port 22
ncrypted traffic)
FIN
Ban this IP in the firewalll
Response

- If true positive, outcome was good

# Intrusion Detection



I was just trying to scp several small files one at a time…

traffic)

FIN

Ban this IP in the firewalll

- If **false** positive, then it was bad
    - Abnormal is not always bad

# Detection

- **Automated**: Algorithmic analysis + detection
  - Signature based: look for patterns
    - This seems to be trying many passwords
    - This seems to be port scanning
  - Anomaly detection:
    - Develop ML model of normal behavior
    - Find things that deviate
- **Human**:
  - Look at logs, system behavior etc

# Detection

- Not limited to network activity
  - These aren't queries that we ever run…
  - This return address has been overwritten
  - This pattern of system calls is unusual
  - There have been 4 failed login attempts for user “brian"
  - …
- Similar ideas in non-computer security
  - Bank watches credit card purchases for suspicious activity
  - Unattended bags at airport
  - …

# Responses

- Notify administrators
  - **Send email**: Hey something is strange… Here is what is up!
  - Pros and cons?

# Responses

- Notify administrators
  - Send email, text, etc: Hey something is strange… Here is what is up!
  - Pros and cons?
- Block suspicious behavior
  - Lock account, firewall traffic, ….
  - Pros and cons?

# Responses

- Notify administrators
  - Send email, text, etc: Hey something is strange… Here is what is up!
  - Pros and cons?
- Block suspicious behavior
  - Lock account, firewall traffic, ….
  - Pros and cons?
- Shutdown affected system
  - Power that machine off
  - Pros and cons?

# Responses

- Notify administrators
  - Send email, text, etc: Hey something is strange… Here is what is up!
  - Pros and cons?
- Block suspicious behavior
  - Lock account, firewall traffic, ….
  - Pros and cons?
- Shutdown affected system
  - Power that machine off
  - Pros and cons?
- Nuke and restore from backup? (or even throw away hw?)

# Factors in Choosing Response

- False positive rate
  - How certain are we that suspicious = bad?
- Severity of suspected attack
  - How bad is it?
    - Someone trying to find a vulnerability vs
    - Server was rooted
- Impacts of response on "good" users/ how many affected
  - Bad impacts: services temporarily unavailable, …
  - Good impacts: prevent leakage of sensitive info,…

# Wrap Up

- Assume security measures will fail!
  - Multiple levels: mitigate damage if one fails
- Detect suspicious activity
  - Don't just assume everything is good, look for bad stuff
- Respond to threats
  - What to do: it depends…