



TIPS FOR LOGGING

June-2019

Who we are?

Axon Active Vietnam

CEO MARKUS BAUR



URI

CAO Truc Nguyen

HR General
Phuong Nguyen

FIN

Thao Luong

Recruitment
Truc Nguyen



Zürich

CIO Sebastian Sussmann

ICT

CTO Khanh Tran

R & D

Giang Le

Knight

CTO Khanh Tran

Online Marketing & Design
Hung Do

Ho Chi Minh Branch

Branch Director - Long Nguyen

Genève Nhan Nguyen	Schwyz Cuong Le	Wallis Thu Dinh	Glarus An Doan	Zug Bang Ngo	Facility Management Hieu Nguyen
Neon An Tran	Avatar Diem Nguyen	Up Nhung Ngo	Alibaba Dung Tran	5Star Cuong Do	Amigo Huy Nguyen
Jaguar Kiet Pham	Next Nam Ha	Wawa Tung Le	Dilbert Hung Nguyen	iKool Hoang Tran	
Slash Tien Hoang	Wow Thong Nguyen	Kepler Cuong Phan	Doremon Anh Do	Chamelizard Bang Ngo	
Shark Hien Nguyen	Golden Nguyen Bui	Nova Trung Ly			
Iris Vu Tran	Pixels Duc Le	Vision Ngan Tran			
Hermes Hieu Nguyen	Smurfs Bao Phan	Samurai Tuan Pham			
Minos Han Huynh	X Loc Nguyen	Arrow Binh Nguyen			
	Helios Tuan Ta				
	Hacka Chau Le				

Da Nang Branch

Branch Director - Hai Dang

Aargau Tung Nguyen	Bern Lien Vo	Basel Nam Dang	Thurgau Thuan Ho	Office Management Minh Phan
WIN3 Hoang Nguyen	Sparta Hung Truong	Anonymous Tho Le	Bolt Loc Nguyen	ICT Nam Nguyen
Bamboo Ngoc Phan	Uranus Vuong Huynh	Fusion Linh Do	Sunwheel Phat Vu	
Bingo Linh Hoang	Europa Ha Ton	Crocodile Nhat Le	Everest Mao Nguyen	
Falcon Hai Nguyen	Galaxy Phuc Thai	Apollo Hoang Nguyen	Gun8 Tung Nguyen	
Burning Phuong Pham	Nemo Thai Hoang	Skynet Khoa Tran	Boeing Luan Vo	
	Kangaroo Truong Vo	Bee12 Quang Vo		
	Mercury Minh Nguyen	Hulk Tuan Nguyen		
	Scuti Hai Nguyen	Dragon Thuy Nguyen		
	Koala Lien Vo			

Can Tho Branch

Branch Director – Muon Nguyen

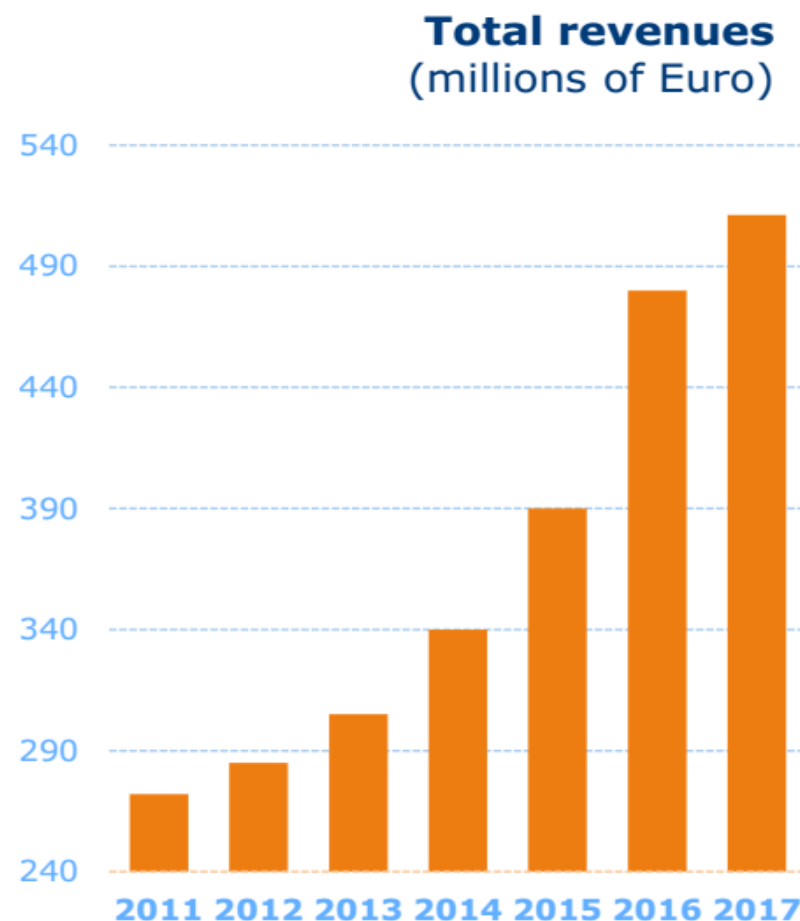
Lucerne Muon Nguyen	Office Management Trang Dao
Pioneer Hung Nguyen	
Infinity Khoa Tang	
Innovation Khanh Vo	
Brothers Ngo Dinh	
Brotherhood Ngo Dinh	
Rocket Trung Le	
Sunshine Muon Nguyen	
BlueBirds Tin Tran	
Wave Muon Nguyen	

Myanmar Branch

Branch Director – Dominik Sauter

Tessin Dominik Sauter
Zenith Giang Le

CRIF AT A GLANCE

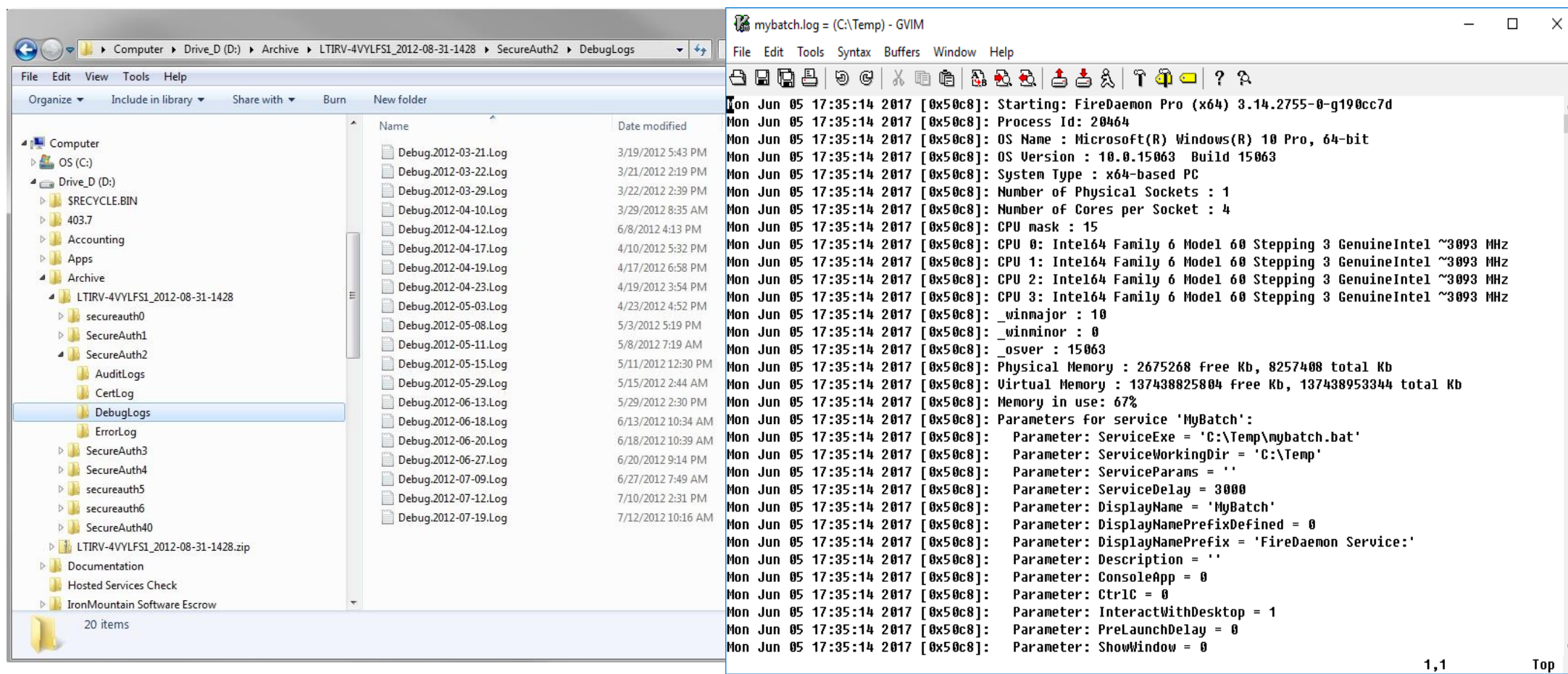


Agenda

- ❑ What is logging in programming?
- ❑ Why logging is important?
- ❑ **Not Good** & Good habits in logging
- ❑ More practical tips
- ❑ Log analysis

What is logging in programming?

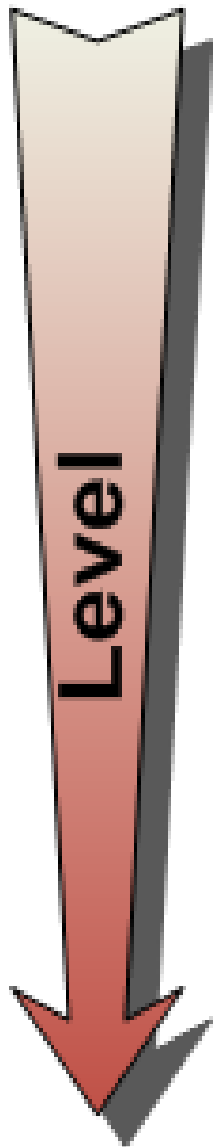
What is logging in programming?



What is logging in programming?

Logging is the process of recording application actions and state to a secondary interface.

What is logging in programming?



DEBUG

fine-grained informational events that are most useful to debug an application

INFO

informational messages that highlight the progress of the application at coarse-grained level

WARN

potentially harmful situations

ERROR

error events that might still allow the application to continue running

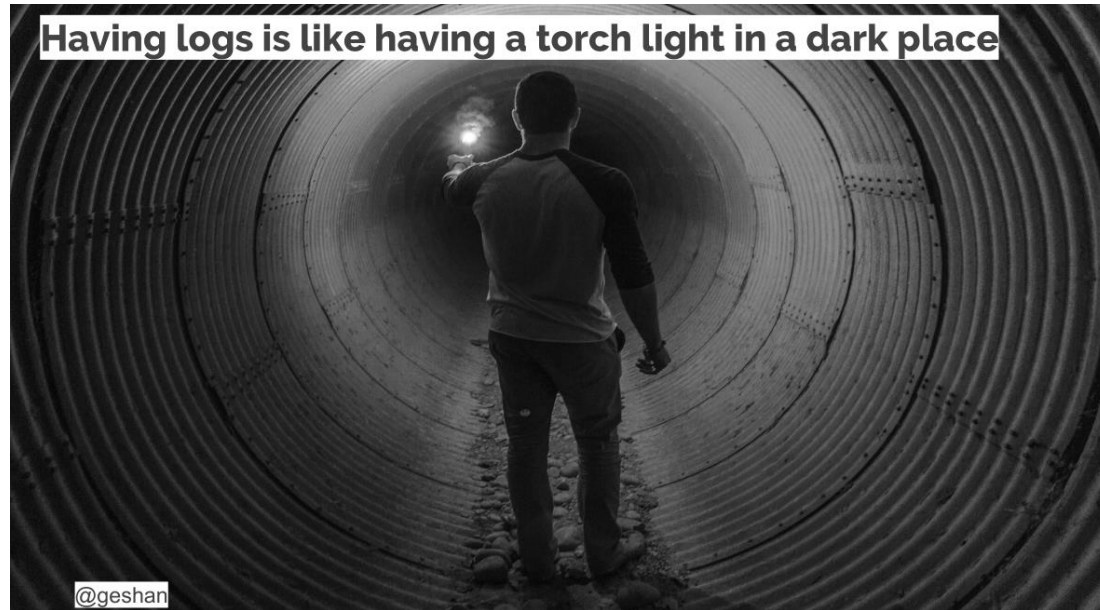
FATAL

very severe error events that will presumably lead the application to abort

What is logging in programming?



What is logging in programming?



“Logs are like car insurance. Nobody wants to pay for it, but when something goes wrong everyone wants the best available” .

Pablo Kraan

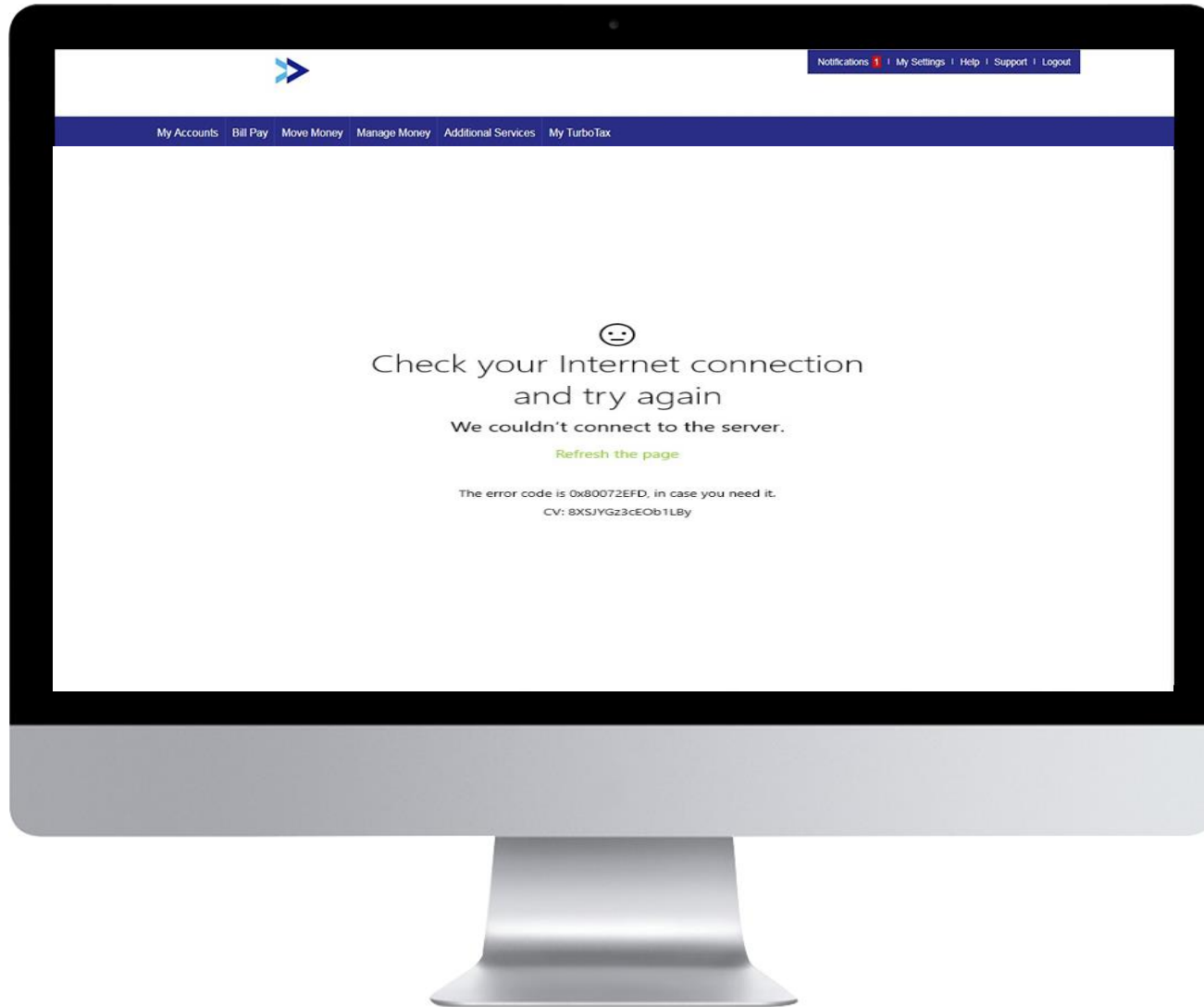
“

If dog is a man's best friend, logs are software engineer's best friend.

@geshan

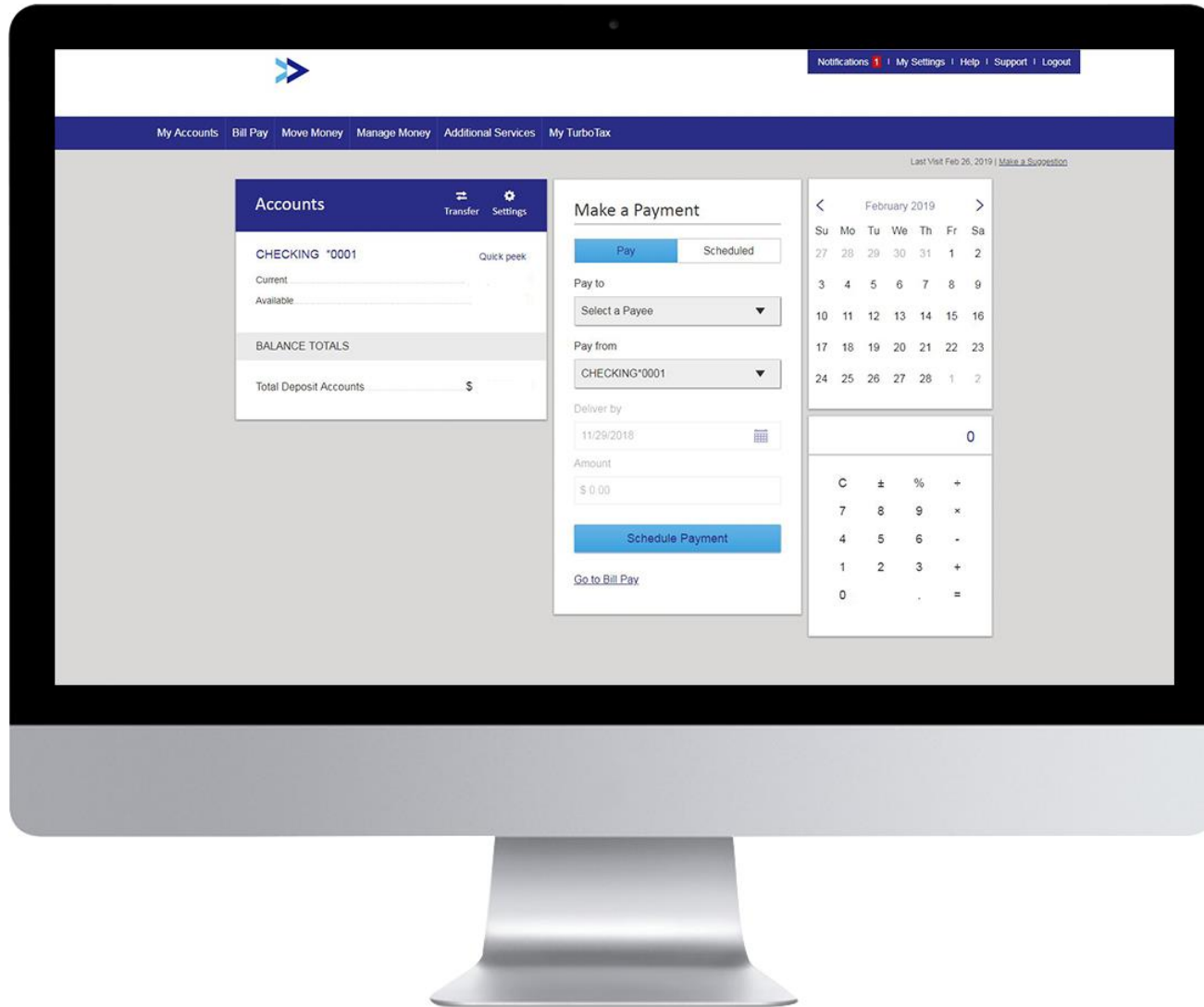
Why logging is important?

Why logging is important?



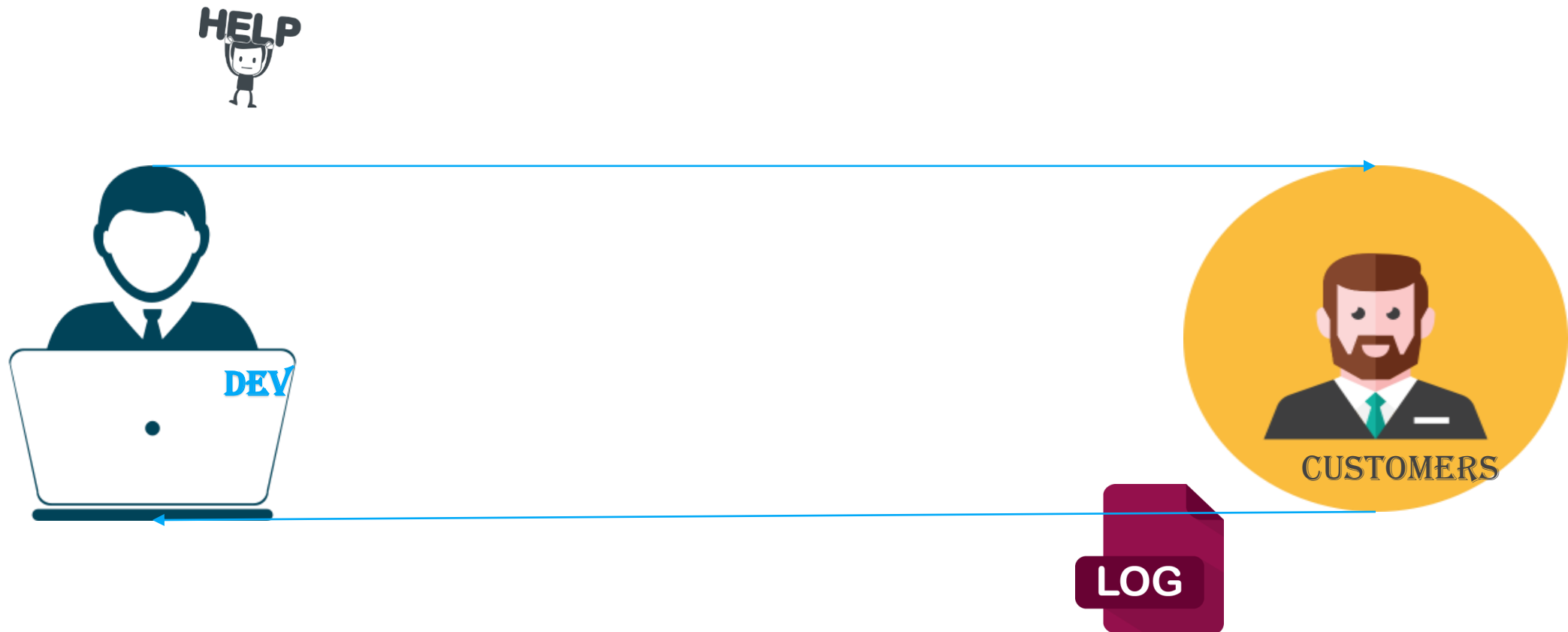
Error happens on
customers' side

Why logging is important?



No bug found on Dev's environments!!!

Why logging is important?



Why logging is important?

```
04-18 07:12:27.590 9224-9395/com.example.android.displayingbitmaps I/OpenGLRenderer: Initialized EGL, version 1.4
04-18 07:12:27.590 9224-9395/com.example.android.displayingbitmaps D/OpenGLRenderer: Swap behavior 1
04-18 07:12:27.827 9224-9229/com.example.android.displayingbitmaps I/art: Do partial code cache collection, code=29KB, data=28KB
04-18 07:12:27.827 9224-9229/com.example.android.displayingbitmaps I/art: After code cache collection, code=27KB, data=27KB
04-18 07:12:27.827 9224-9229/com.example.android.displayingbitmaps I/art: Increasing code cache capacity to 128KB
04-18 07:12:27.916 9224-9229/com.example.android.displayingbitmaps I/art: Do partial code cache collection, code=61KB, data=53KB
04-18 07:12:27.917 9224-9229/com.example.android.displayingbitmaps I/art: After code cache collection, code=61KB, data=53KB
04-18 07:12:27.917 9224-9229/com.example.android.displayingbitmaps I/art: Increasing code cache capacity to 256KB
04-18 07:12:28.098 9224-9224/com.example.android.displayingbitmaps I/art: Starting a blocking GC Explicit
04-18 07:12:28.121 9224-9224/com.example.android.displayingbitmaps I/art: Explicit concurrent mark sweep GC freed 1933(106KB)
  AllocSpace objects, 0(0B) LOS objects, 40% free, 13MB/21MB, paused 333us total 22.300ms
04-18 07:12:28.123 9224-9224/com.example.android.displayingbitmaps I/art: Starting a blocking GC Explicit
04-18 07:12:28.145 9224-9224/com.example.android.displayingbitmaps I/art: Explicit concurrent mark sweep GC freed 187(8KB)
  AllocSpace objects, 0(0B) LOS objects, 39% free, 13MB/21MB, paused 318us total 21.597ms
04-18 07:12:28.377 9224-9388/com.example.android.displayingbitmaps E/GRPC: tcp_client_posix.c:173] failed to connect to
  'ipv4:127.0.0.1:12389': socket error: connection refused
04-18 07:12:29.374 9224-9389/com.example.android.displayingbitmaps E/GRPC: tcp_client_posix.c:173] failed to connect to
  'ipv4:127.0.0.1:12389': socket error: connection refused
04-18 07:12:30.375 9224-9389/com.example.android.displayingbitmaps E/GRPC: tcp_client_posix.c:173] failed to connect to
  'ipv4:127.0.0.1:12389': socket error: connection refused
```

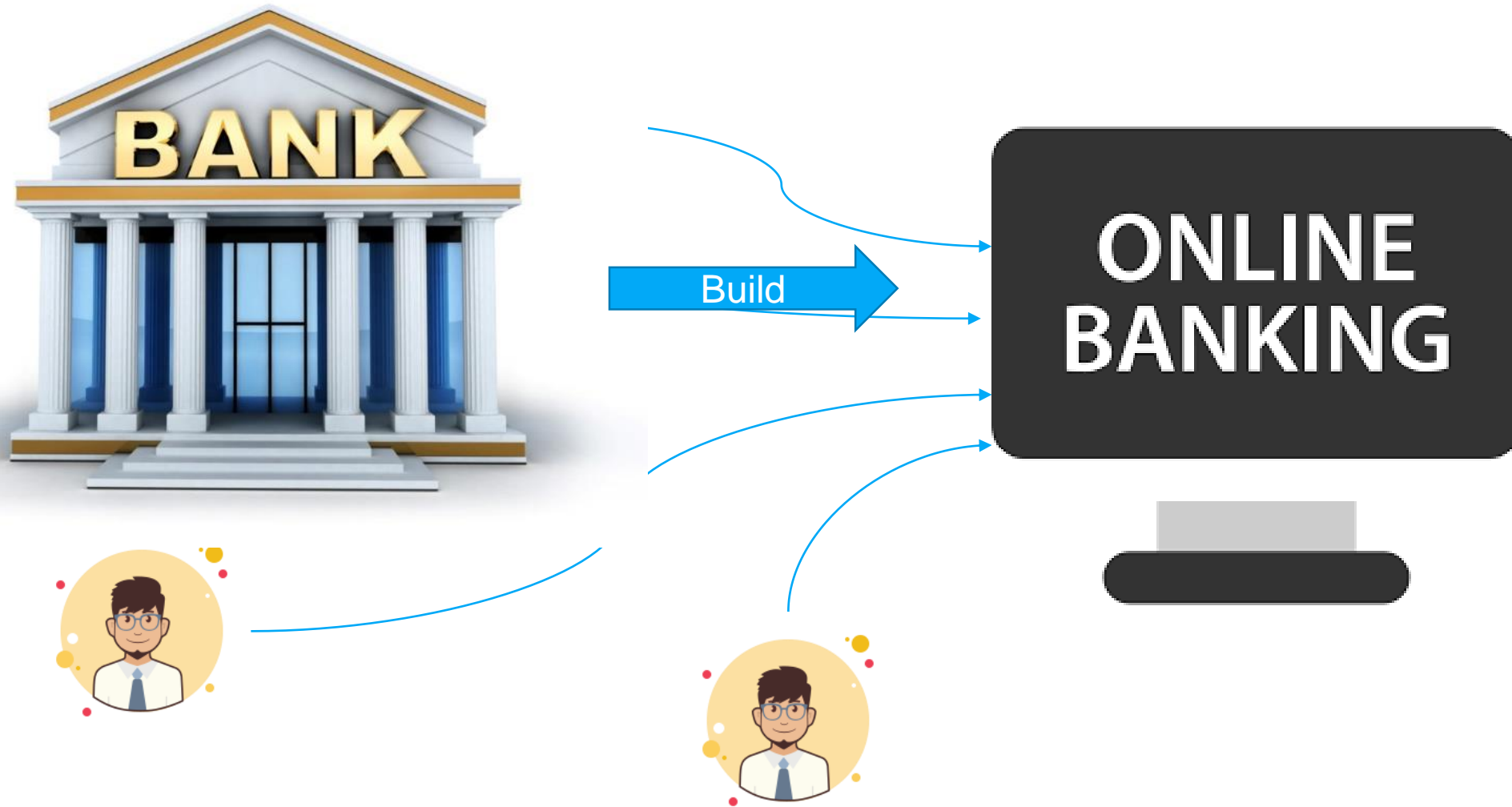
Why logging is important?

One more example

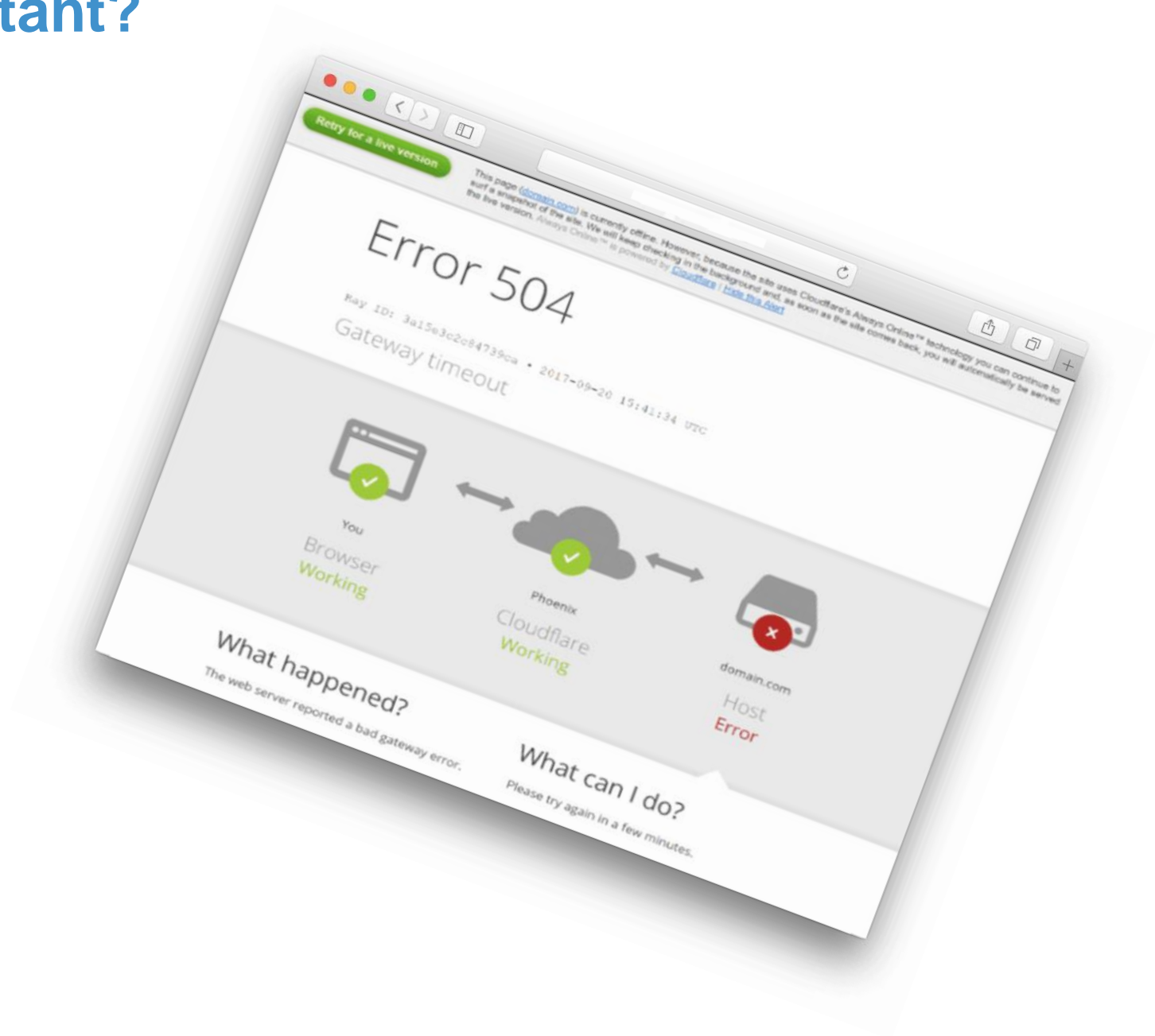
Why logging is important?



Why logging is important?



Why logging is important?



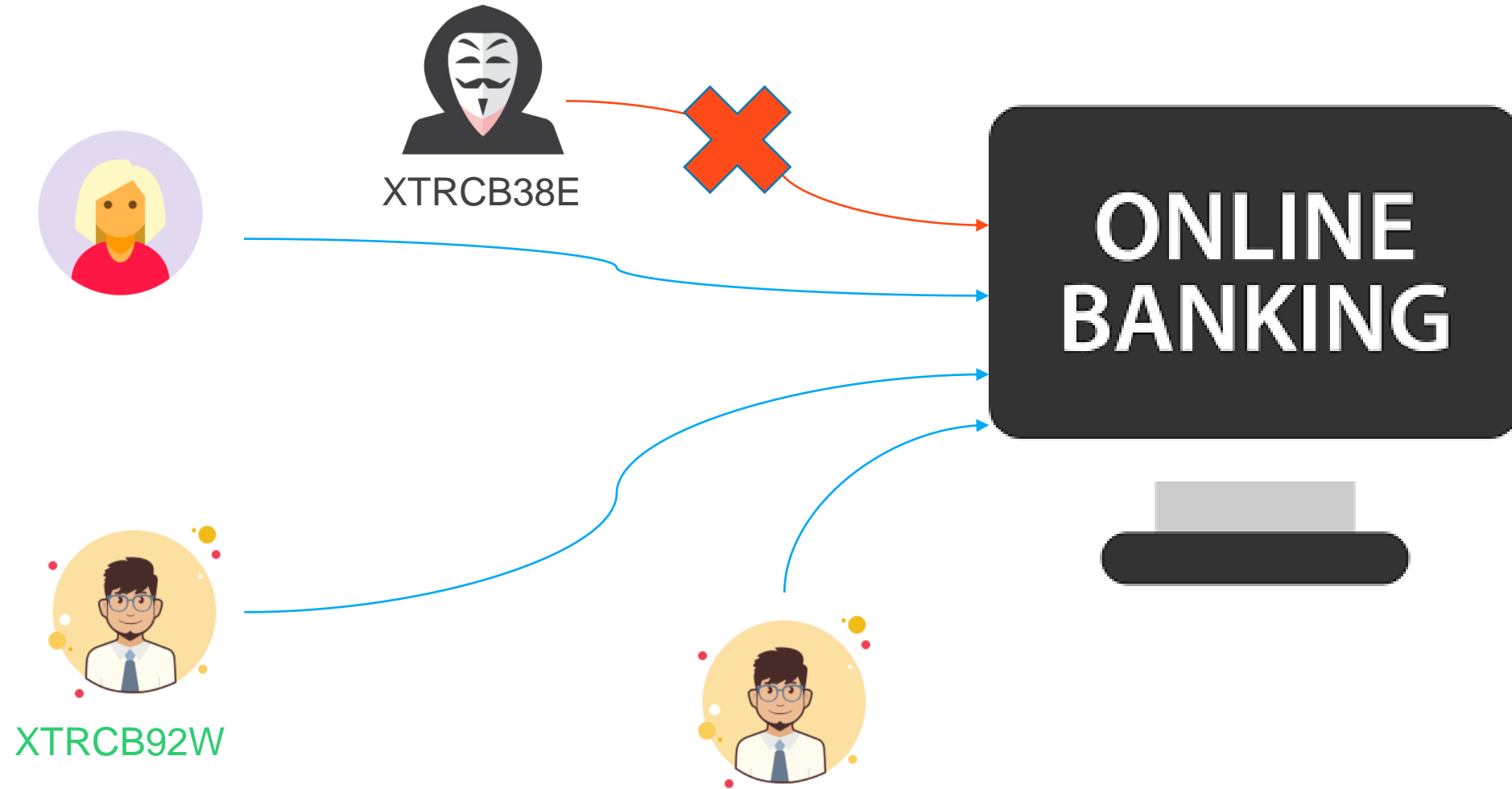
Why logging is important?

```
- - [10/Jan/2019:10:51:36 +0700] " User XTRCB38E login successfully...." - -
- - [10/Jan/2019:10:51:36 +0700] " User XTRCB92W login successfully...." - -
- - [10/Jan/2019:10:51:36 +0700] " [XTRCB38E] POST " 200 5103 10 "http://stdbank.com:8088/accountInfo" 537.36" - -
- - [10/Jan/2019:10:51:36 +0700] " [XTRCB38E] POST " 200 5103 12 "http://stdbank.com:8088/accountInfo" 537.36" - -
- - [10/Jan/2019:10:51:36 +0700] " [XTRCB92W] POST " 200 5103 10 "http://stdbank.com:8088/withdrawal" 537.36" - -
- - [10/Jan/2019:10:51:36 +0700] " [XTRCB38E] POST " 200 5103 18 "http://stdbank.com:8088/accountInfo" 537.36" - -
- - [10/Jan/2019:10:51:36 +0700] " [XTRCB38E] POST " 200 5103 10 "http://stdbank.com:8088/accountInfo" 537.36" - -
- - [10/Jan/2019:10:51:36 +0700] " [XTRCB38E] POST " 200 5103 10 "http://stdbank.com:8088/accountInfo" 537.36" - -
- - [10/Jan/2019:10:51:36 +0700] " [XTRCB92W] GET " 200 5103 11 "http://stdbank.com:8088/bill" 537.36" - -
- - [10/Jan/2019:10:51:40 +0700] " [XTRCB38E] POST " 200 5103 10 "http://stdbank.com:8088/accountInfo" 537.36" - -
- - [10/Jan/2019:10:51:40 +0700] " [XTRCB38E] POST " 200 5103 22 "http://stdbank.com:8088/accountInfo" 537.36" - -
- - [10/Jan/2019:10:51:40 +0700] " [XTRCB38E] POST " 200 5103 10 "http://stdbank.com:8088/accountInfo" 537.36" - -
- - [10/Jan/2019:10:51:40 +0700] " [XTRCB92W] GET " 200 5103 12 "http://stdbank.com:8088/bill" 537.36" - -
- - [10/Jan/2019:10:51:40 +0700] "User XTRCB92W logout successfully...." " - -
- - [10/Jan/2019:10:51:40 +0700] " [XTRCB38E] POST " 200 5103 10 "http://stdbank.com:8088/accountInfo" 537.36" - -
- - [10/Jan/2019:10:51:40 +0700] " [XTRCB38E] POST " 200 5103 10 "http://stdbank.com:8088/accountInfo" 537.36" - -
- ...
- ..
- .
```

Why logging is important?



Why logging is important?



Why logging is important?



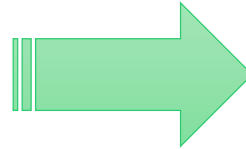
- Know what happen
- Save time
- Overview about process
- Control access
- ...

Not Good & Good **habits in logging**

Using Log framework



```
</> [ ] X
1
2
3
4 String logMessage = "This is a log message";
5
6 SomeWriter writer = new SomeWriter(filename);
7
8 writer.write(logMessage);
9
10 writer.close();
11
12
13
```



```
</> [ ] X
1
2
3
4 private static final Log = Logger.getLog();
5
6 Log.error("Error message");
7
8 Log.info("Information");
9
10 Log.debug("Debug purpose");
11
12
13
```

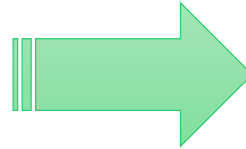
Using log framework instead of writing your own code.

Lack of information

Lack of information



```
</> [ ] X
1
2  if (id == null) {
3
4      log.error("Invalid input.");
5
6  }
7
8
9  if (name == null) {
10
11      log.error("Invalid input.");
12
13  }
```

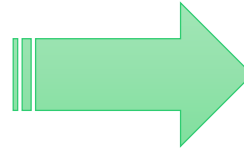


```
</> [ ] X
1
2  if (id == null) {
3
4      log.info("ID must not be null.");
5
6  }
7
8  if (name == null) {
9
10      log.info("Name must not be null.");
11
12  }
13 }
```

Lack of information



```
</> [ ] X
1
2
3 try {
4     Connection c = server.connect();
5 } catch (Exception e) {
6     log.error("Cannot connect to server.", e);
7 }
8
9
10
11
12
13
```

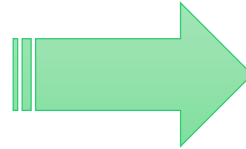


```
</> [ ] X
1
2
3 try {
4     Connection c = server.connect();
5 } catch (Exception e) {
6     log.error("Cannot connect to server " +
7 host + ":", e);
8 }
9
10
11
12
13
```


Lack of information



```
</> [ ] X
1  try {
2
3      Reader reader = new Reader("file.txt");
4
5      while (c != null) {
6
7          c = reader.readLine();
8
9          processLine(c);
10     }
11 } catch (Exception e) {
12     log.error("Error occurred.", e);
13 }
```



```
</> [ ] X
1  int i = 0;
2
3  try {
4      Reader reader = new Reader("file.txt");
5
6      while (c != null) {
7
8          c = reader.readLine();
9
10         processLine(c);
11         i++;
12     }
13 } catch (Exception e) {
14     log.error("Error occurred at line " + i, e);
15 }
```

Redundant information

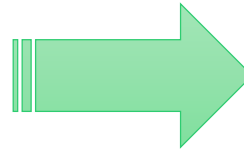
Redundant information



```
...  
catch (ValidationException e) {  
    log.error("Validation detail message here", e);  
}
```

```
...  
catch (PermissionException e) {  
    log.error("Permission detail message here", e);  
}
```

```
...  
catch (EntityNotFoundException e) {  
    log.error("Entity not found message here", e);  
}
```

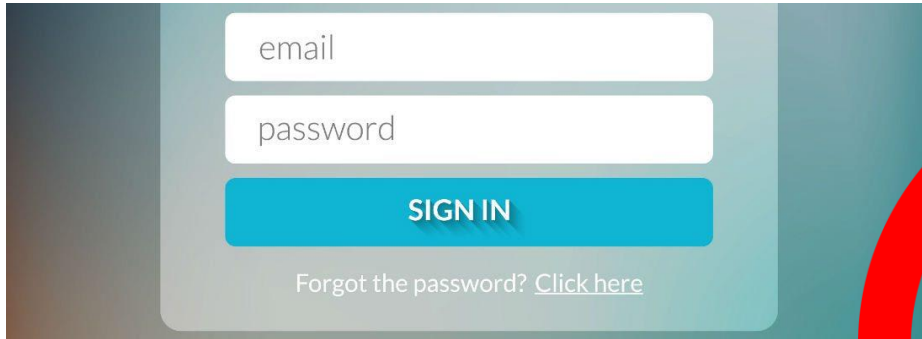


```
...  
catch (ValidationException e) {  
    log.info("Validation detail message here");  
}
```

```
...  
catch (PermissionException e) {  
    log.info("Permission detail message here");  
}
```

```
...  
catch (EntityNotFoundException e) {  
    log.info("Entity not found message here");  
}
```

Sensitive information



email

password

SIGN IN

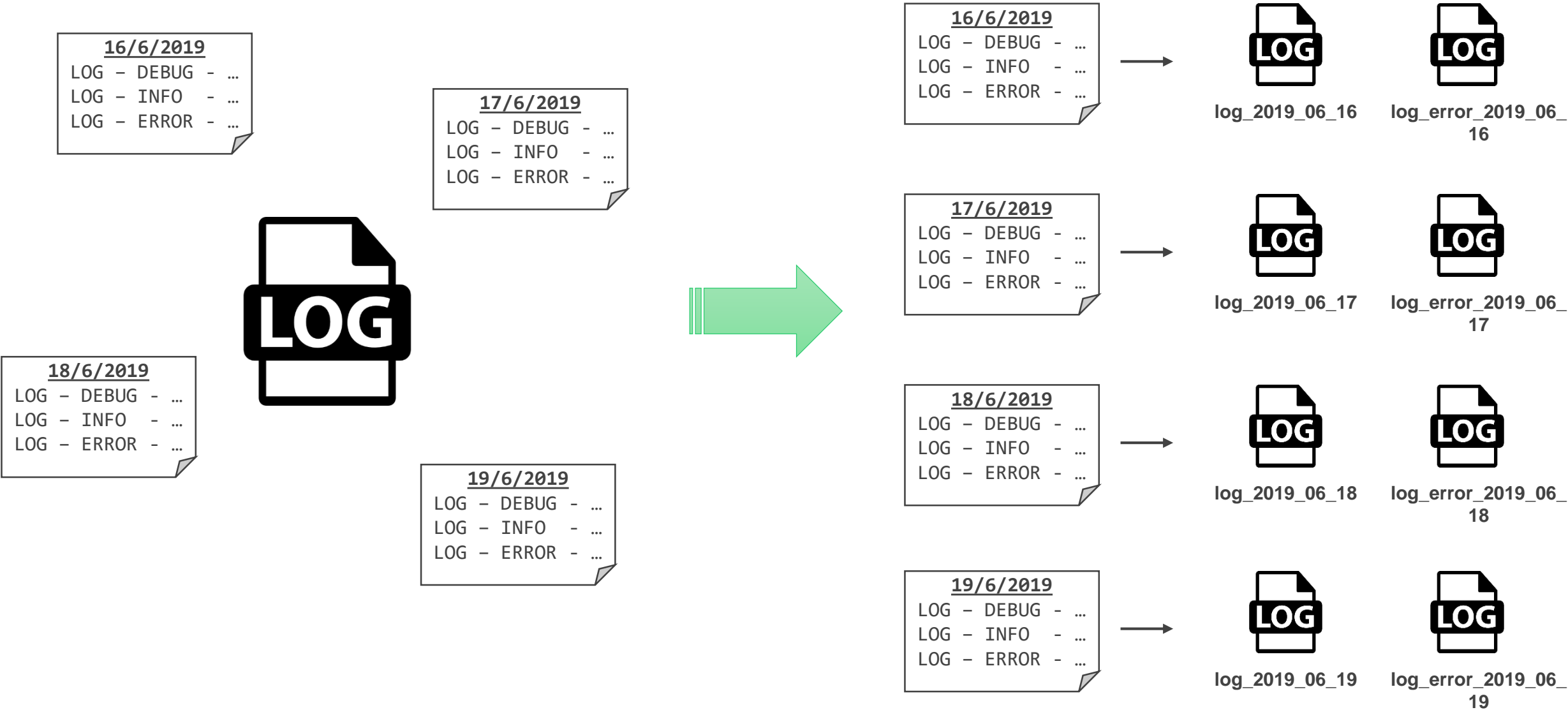
Forgot the password? [Click here](#)



LOG - INFO : username - dilbert
LOG - INFO : password - password

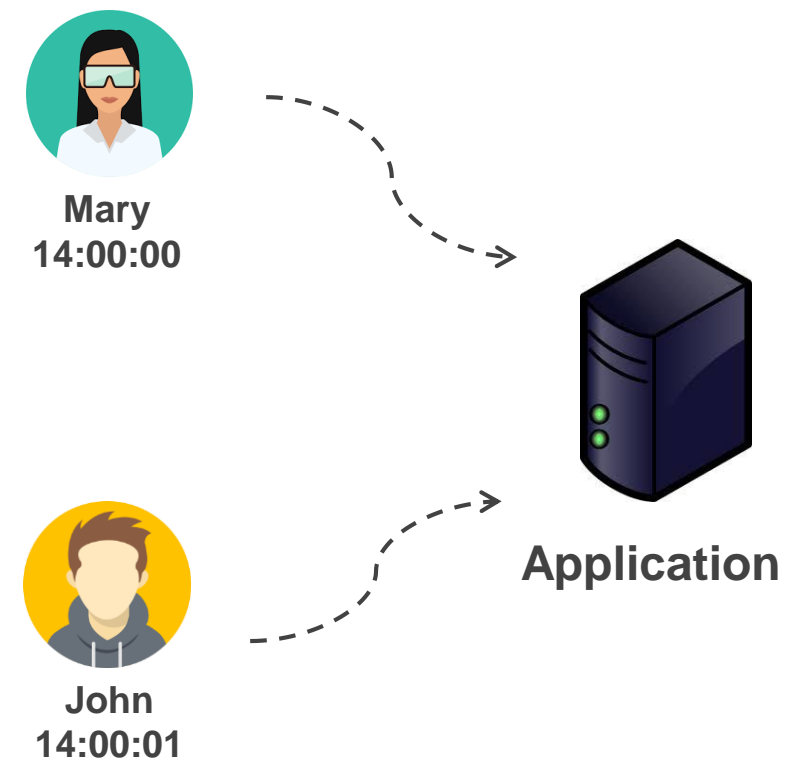
LOG - INFO : number - 5412-7501-2345-0987
LOG - INFO : name - Peter Hans Muller
LOG - INFO : validTill - 12/20
LOG - INFO : CVV - 696

Separated log file



More practical tips

Using unique ID per request

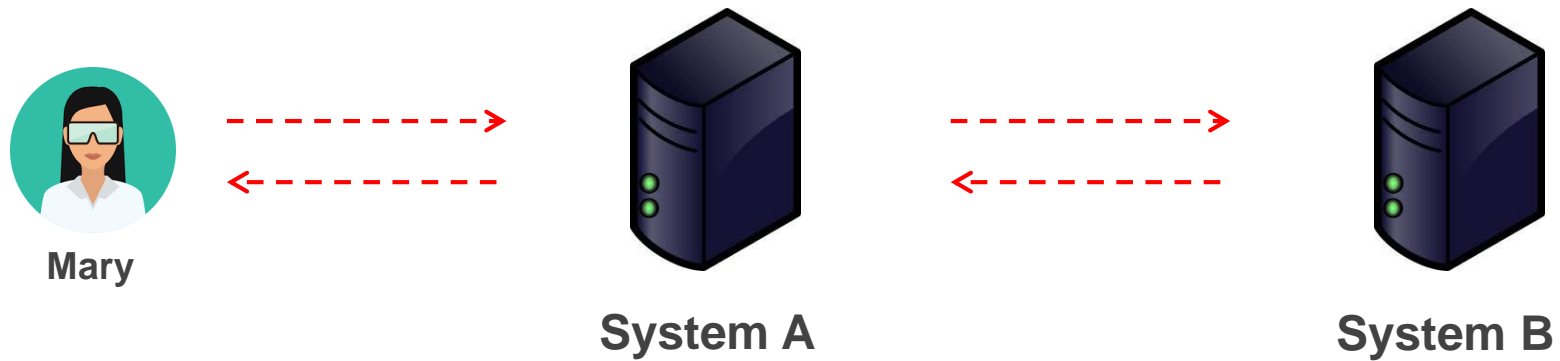


16-6-2019-14:00:00		Process request coming...
16-6-2019-14:00:00		Process step 1
16-6-2019-14:00:01		Process step 2
16-6-2019-14:00:01		Process request coming...
16-6-2019-14:00:01		Process step 1
16-6-2019-14:00:01		Process step 2
16-6-2019-14:00:02		Process step 3 ????
16-6-2019-14:00:02		Process step 3 ????

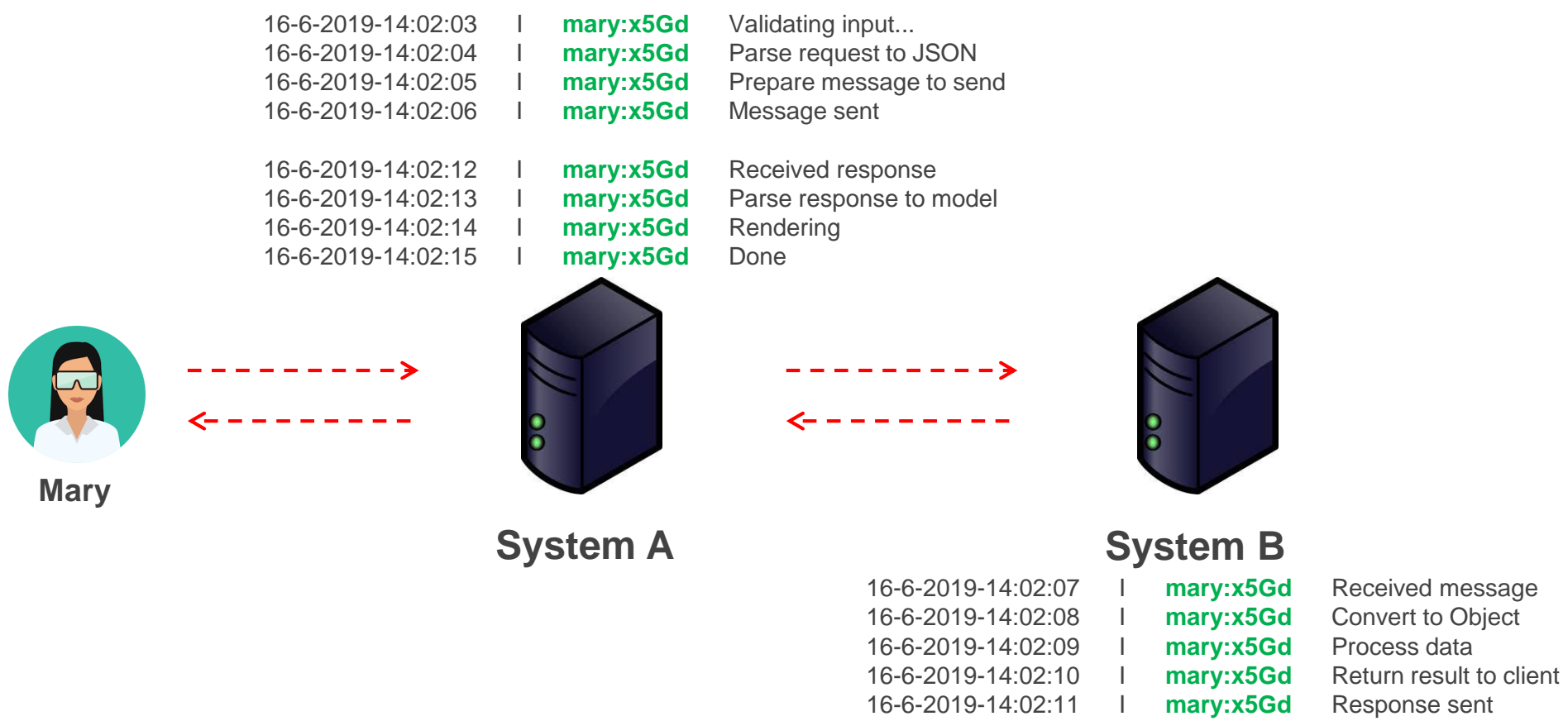


16-6-2019-14:00:00		mary:x5Gd	Process request coming...
16-6-2019-14:00:00		mary:x5Gd	Process step 1
16-6-2019-14:00:01		mary:x5Gd	Process step 2
16-6-2019-14:00:01		john:r4RN	Process request coming...
16-6-2019-14:00:01		john:r4RN	Process step 1
16-6-2019-14:00:01		john:r4RN	Process step 2
16-6-2019-14:00:02		mary:x5Gd	Process step 3
16-6-2019-14:00:02		john:r4RN	Process step 3

Using unique ID per request



Using unique ID per request



Using DEBUG log

```
String url = getUrl();
log.debug("Url: " + url);
HttpResponse response = getHttpClient().execute();
log.debug("Response: " + response);

try {
    getObjectMapper().readTree(response);
} catch (Exception e) {
    log.error("Error occurred while parsing response: " + e.getMessage(), e)
}
```

```
D - Url: http://www.google.com/search?q=developer
D - Response: <html>...</html>
E - Error occurred while parsing response: Unexpected character ('<' (code 60)) at [Source:
java.io.StringReader@39aeed2f; line: 1, column: 2]
    at...
    at...
    at...
```

Log process status while processing big data

```
public void processData() {  
    int total = fileService.getTotalRecords();  
    int bulkSize = 0;  
    for (int i = 0; i < total; i++) {  
        bulkRequest.add(createRequest(fileService.readLine(i)));  
        ++bulkSize;  
        //process block 100 records  
        if (bulkSize == 100 || i == (total - 1)) {  
            BulkResponse response = bulkRequest.execute();  
            bulkSize = 0;  
            int failedRecs = response.getFailedRecords();  
            log.info(format("Total records: %d, processed records: %d, failed records: %d", total, i, failedRecs));  
        }  
    }  
}
```

```
I - Start importing process.  
I - Total records: 125000, processed records: 100, failed records: 0  
I - Total records: 125000, processed records: 200, failed records: 6  
I - Total records: 125000, processed records: 300, failed records: 9  
...  
I - Total records: 125000, processed records: 125000, failed records: 69  
I - Finish processing file 'schools.txt'.  
I - End importing process after 69 ms.
```

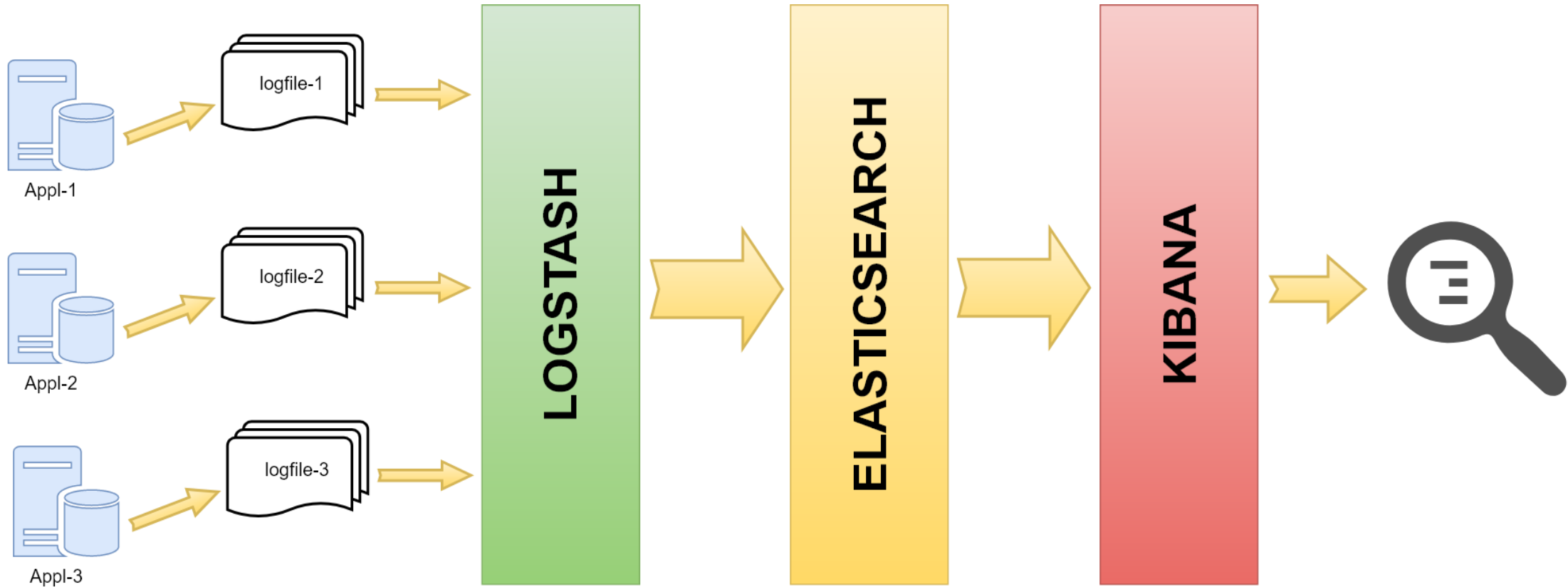

Log analysis

Why analyzing log is important?

- Production monitoring and debugging
- Resource usage
- Tracking client requests
- Performance analysis
- Identifying key trends
- ...

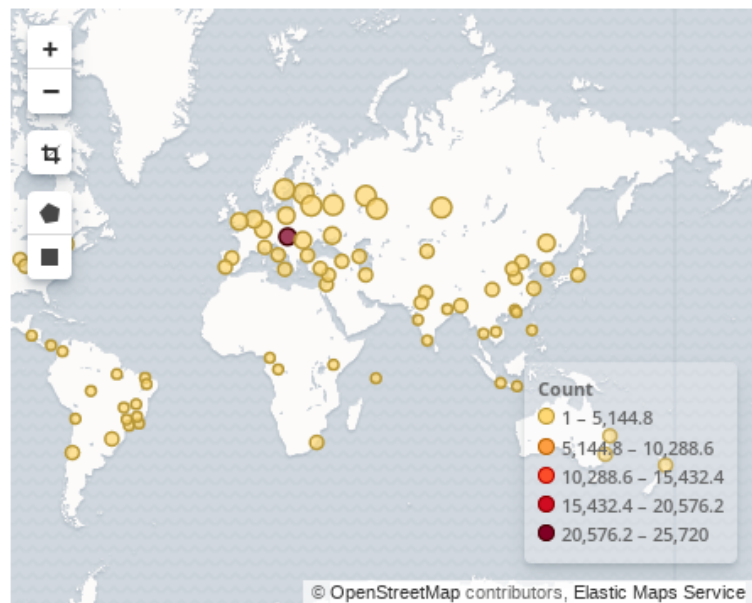


Using Kibana for log analysis

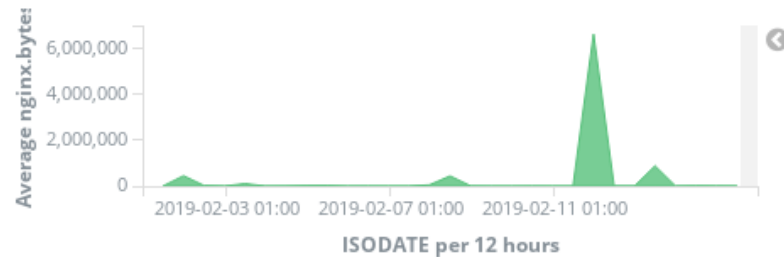


Track client requests

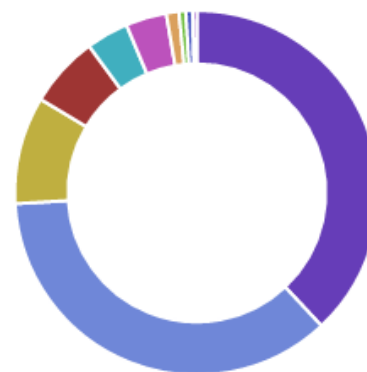
NGINX Client IP Map



NGINX Average bytes



NGINX TOP 10 Response codes



NGINX Top5 Agents

List of Agents	Count
Mozilla/5.0 (Macintosh; Intel Mac OS X 10.14; rv:65.0) Gecko/20100101 Firefox/65.0	13,791
Mozilla/5.0 (Linux) mirall/2.3.3 (Nextcloud)	13,166
Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/73.0.3683.103 Safari/537.36	7,868

NGINX Top5 404 resp. / IP

Client IP	404s
172.20.0.1	497
111.231. [REDACTED]	356
123.207. [REDACTED]	356
47.98. [REDACTED]	355
106.13. [REDACTED]	355

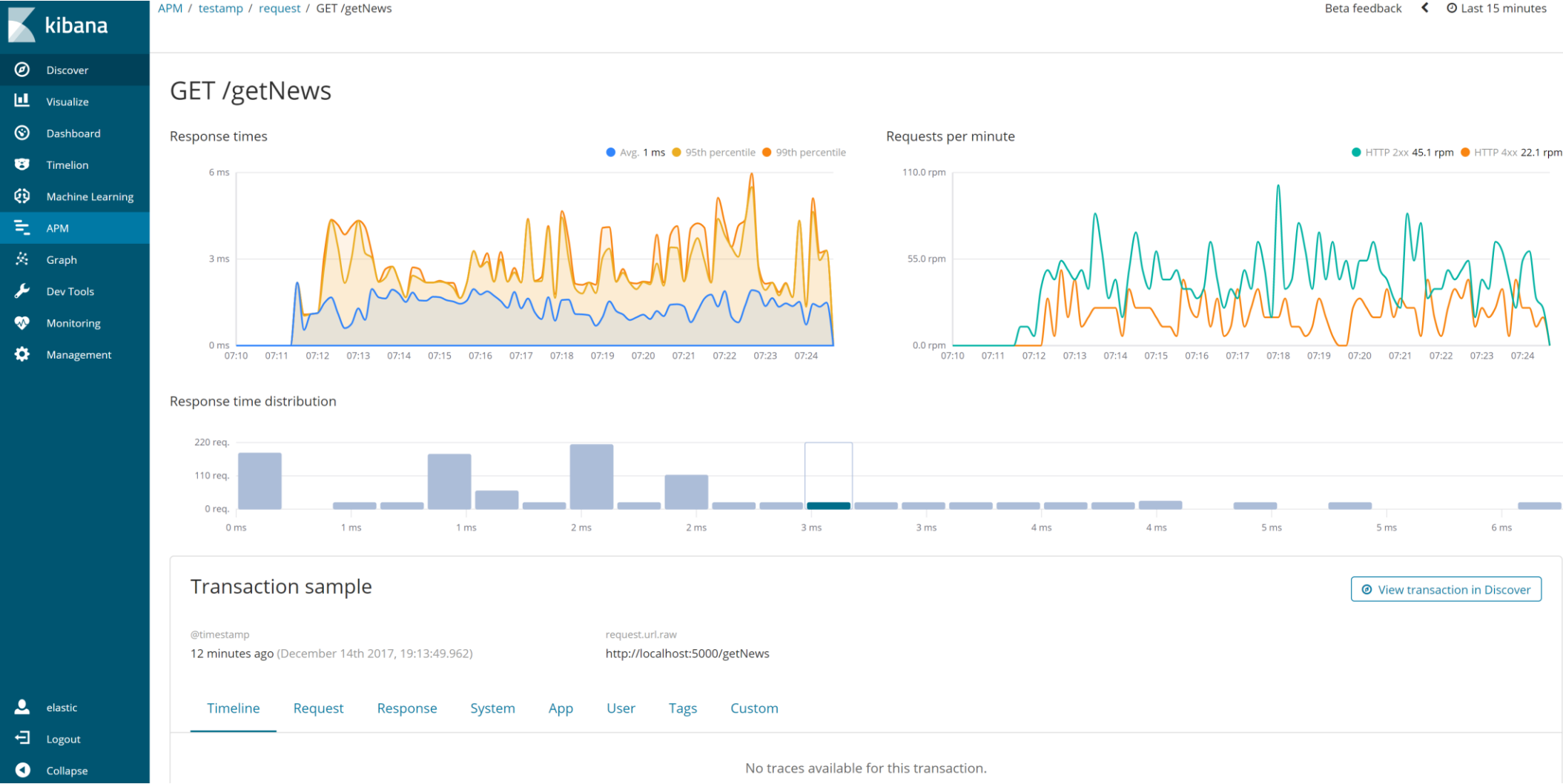
NGINX Top10 HTTP user

Username	Count
Missing	43,238
-	19,611
[REDACTED]	16,644
[REDACTED]	7,960
[REDACTED]	4,772
[REDACTED]	2,699

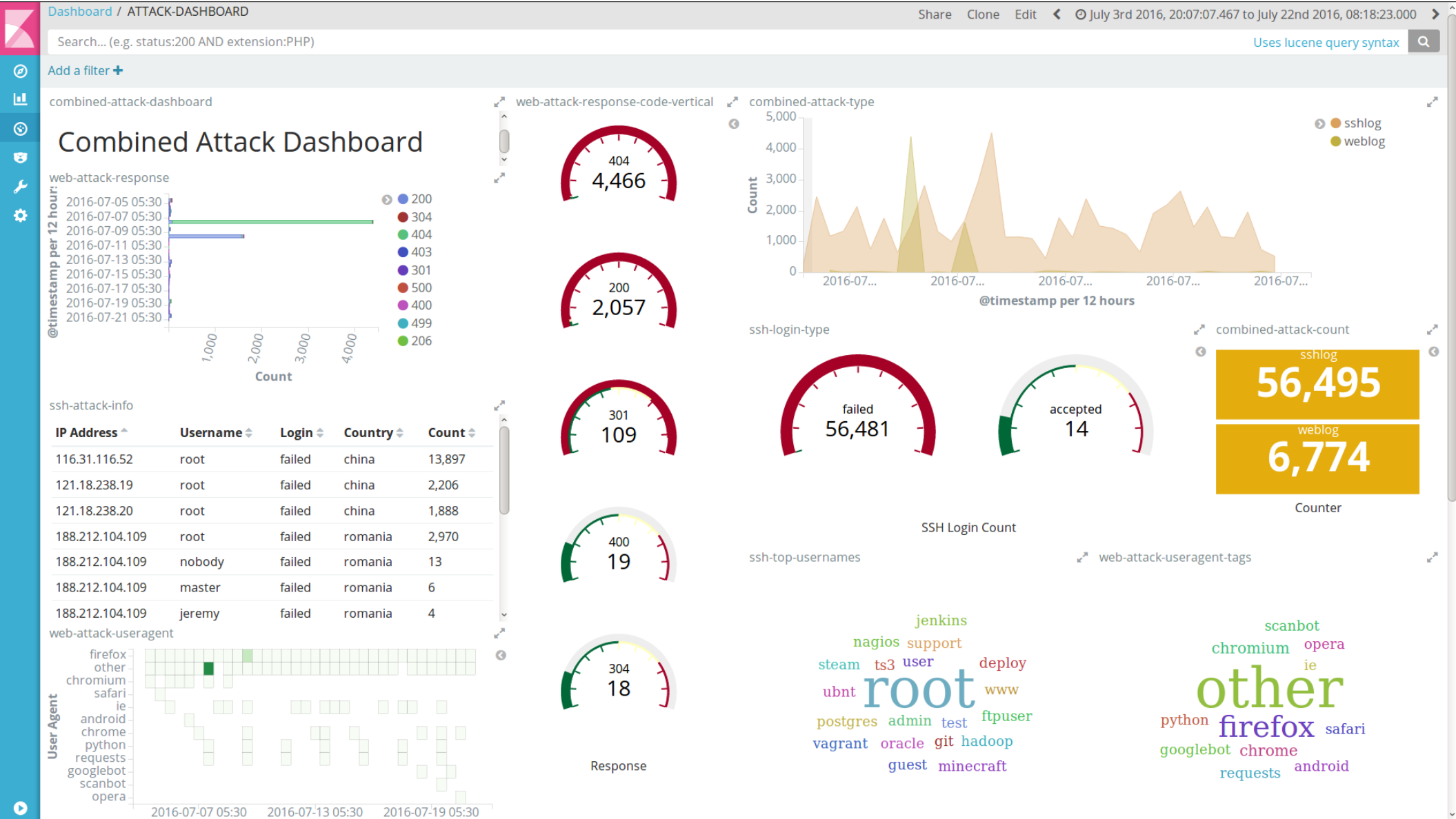
NGINX Top5 Requested URLs

Requested URLs	Count
/oc/remote.php/dav/files/[REDACTED]	7,012
/oc/ocs/v2.php/apps/notifications/api/v2/notifications	5,959
/oc/remote.php/dav/files/[REDACTED]	4,603
/oc/remote.php/dav/files/[REDACTED]	4,161
/oc/remote.php/dav/files/[REDACTED]	2,389

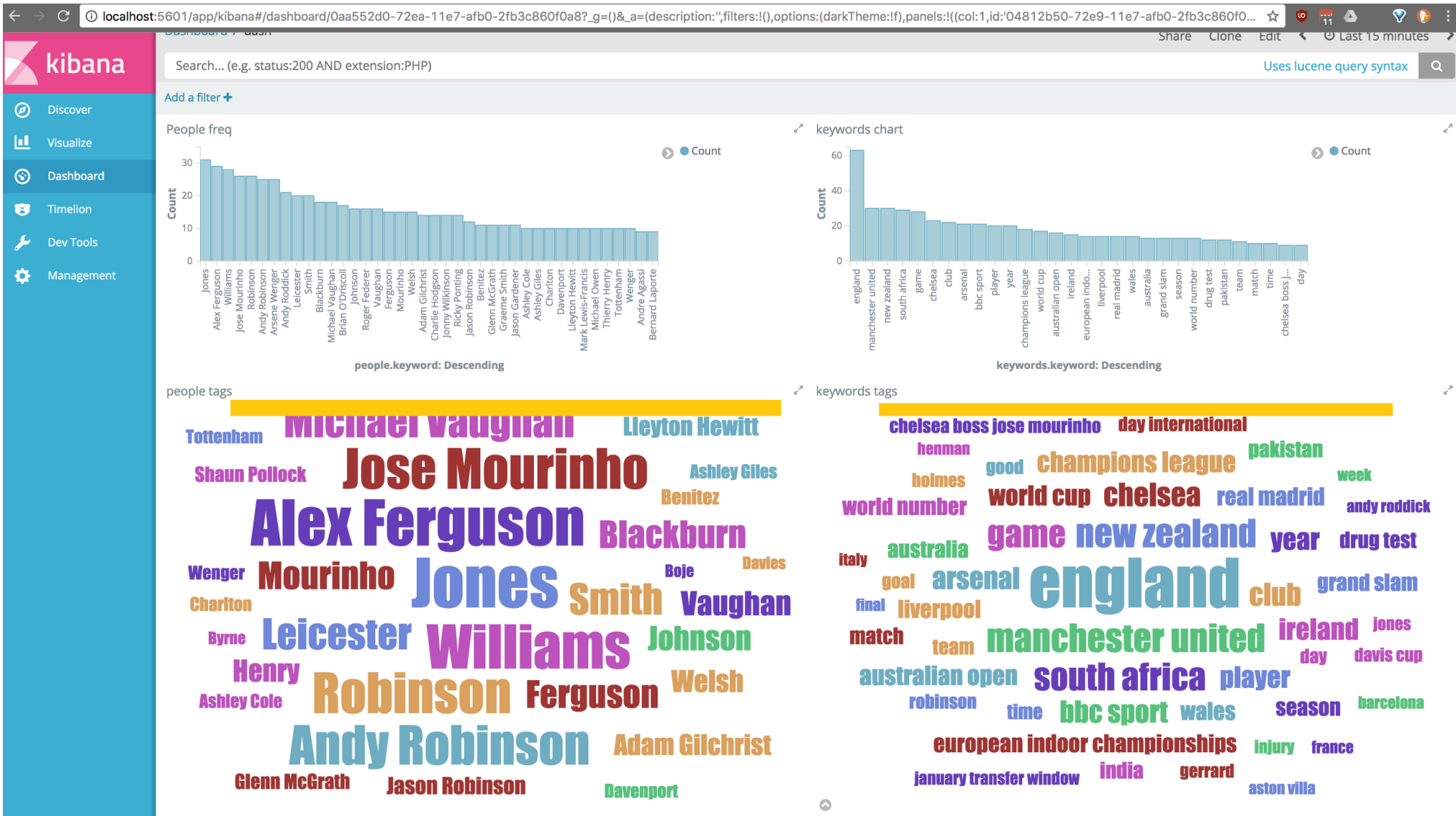
Monitor application performance



Monitor web attack



Identify key trends



Q & A



AXON ACTIVE

👁 Focusing on your agility



THANK YOU!