Scrify: an online tool to validate Bitcoin script

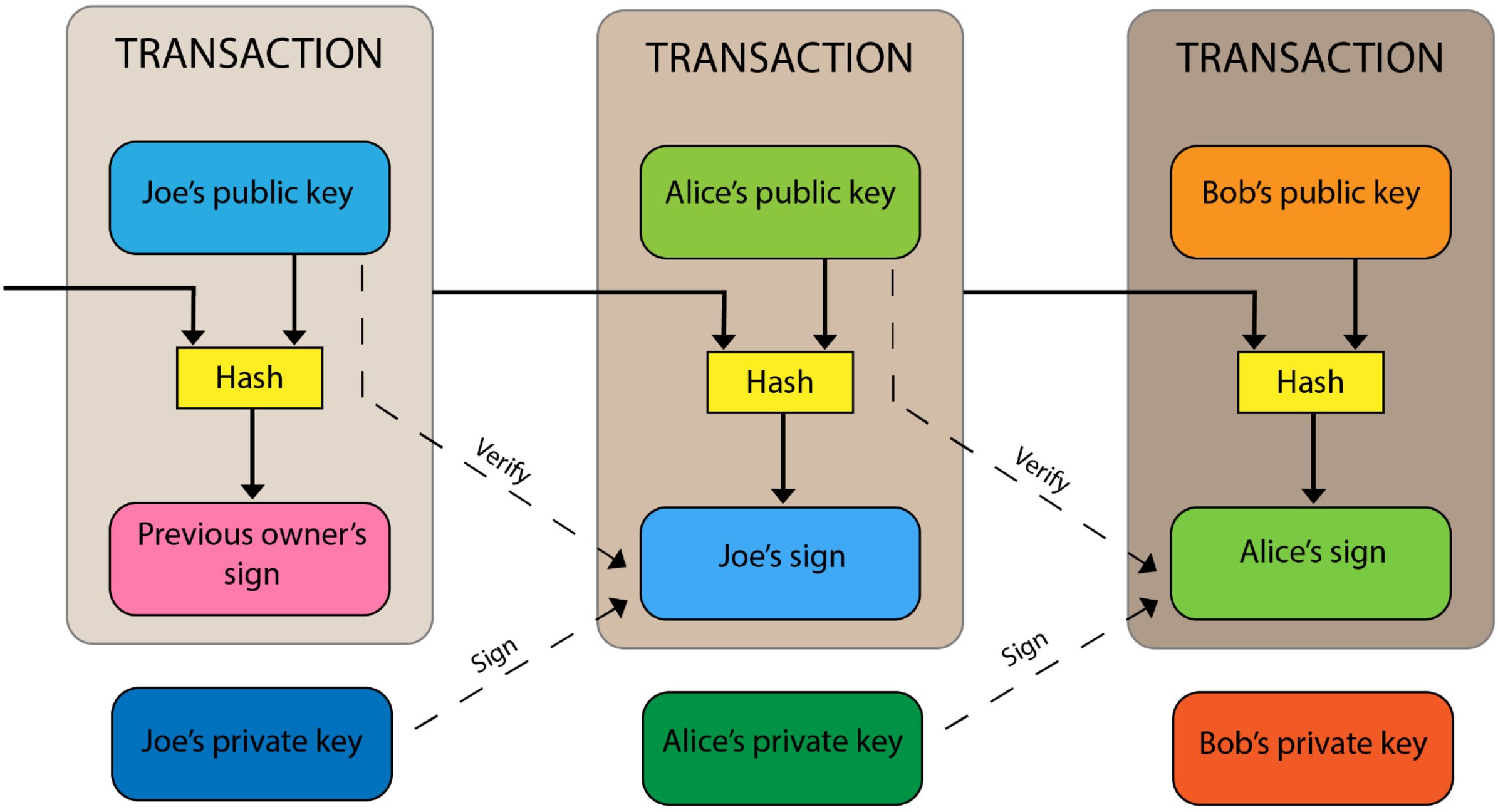
Stefano Bistarelli, Andrea Bracciali, **Ivan Mercanti**

A.D. 1308
unipg
UNIVERSITÀ DEGLI STUDI
DI PERUGIA

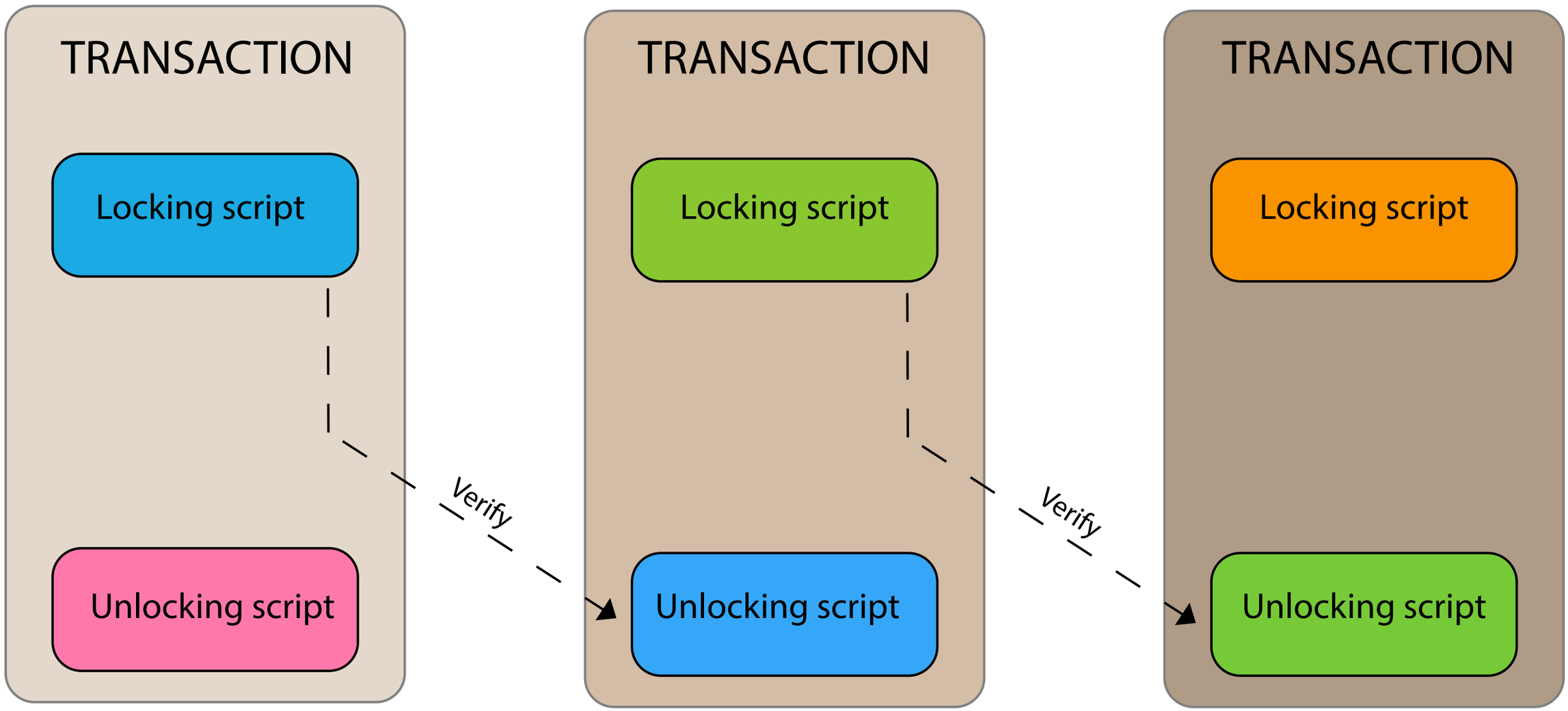
UNIVERSITY of
STIRLING

DLT 23 - 26/05/2023

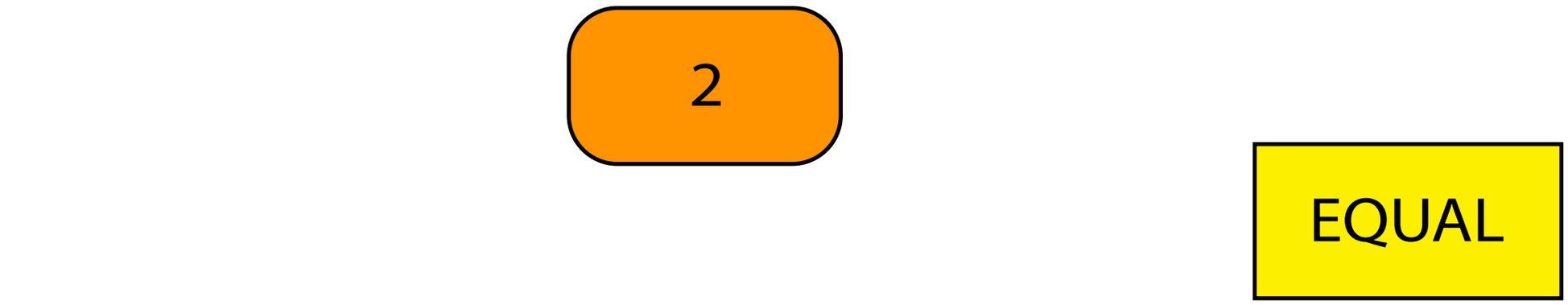
HOW TRANSACTIONS WORK



HOW TRANSACTIONS WORK

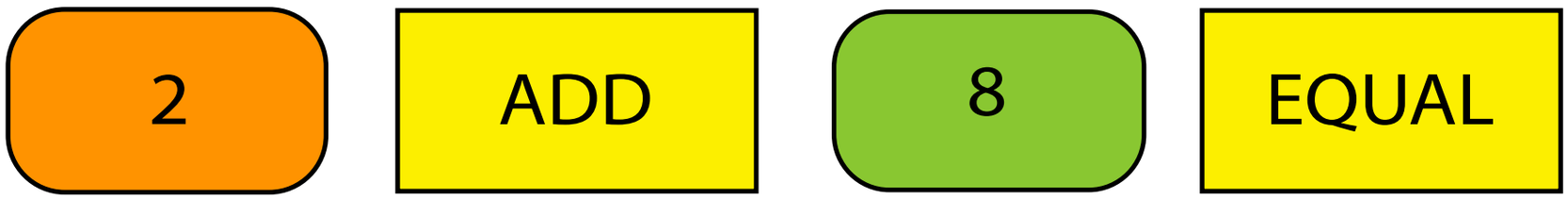


SCRIPT



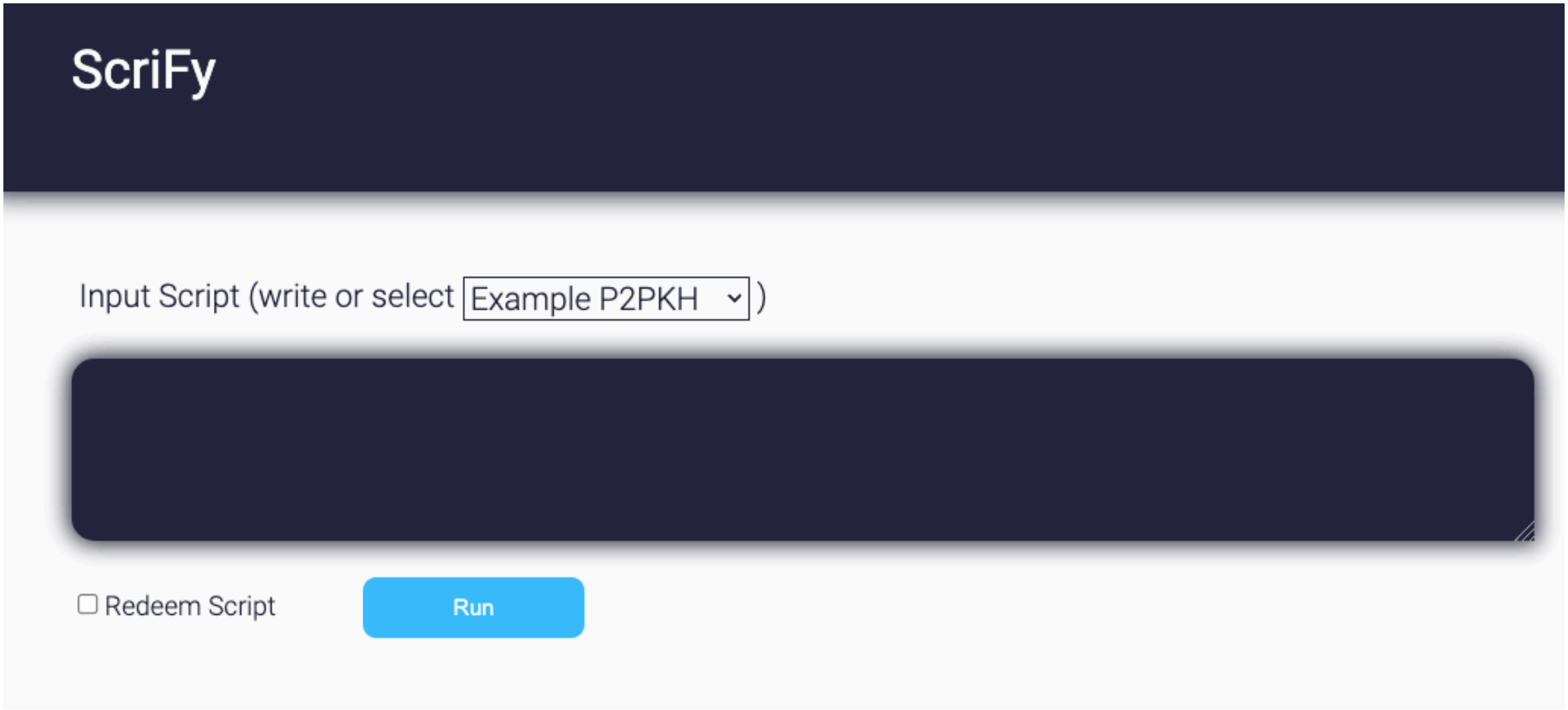
SCRIPT

Locking Script

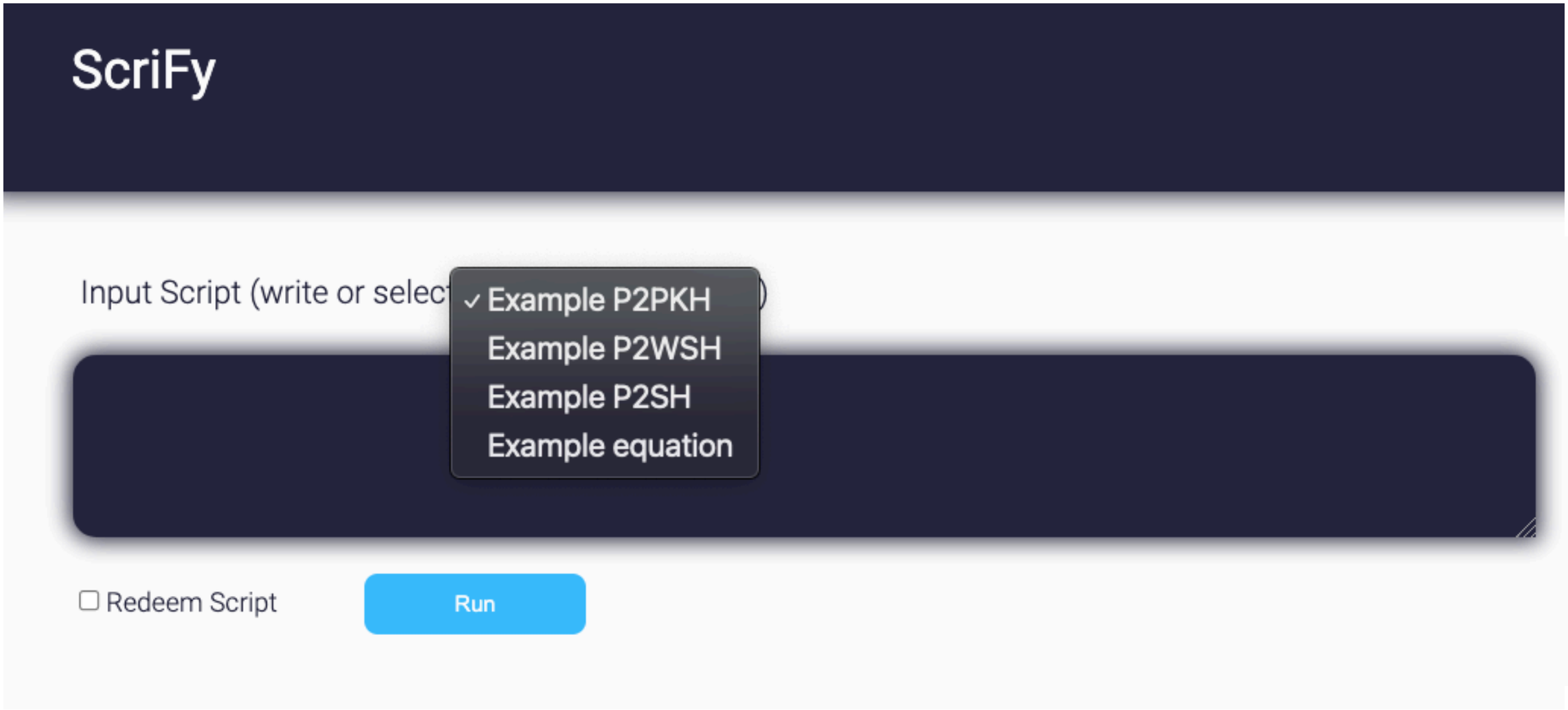


Unlocking Script

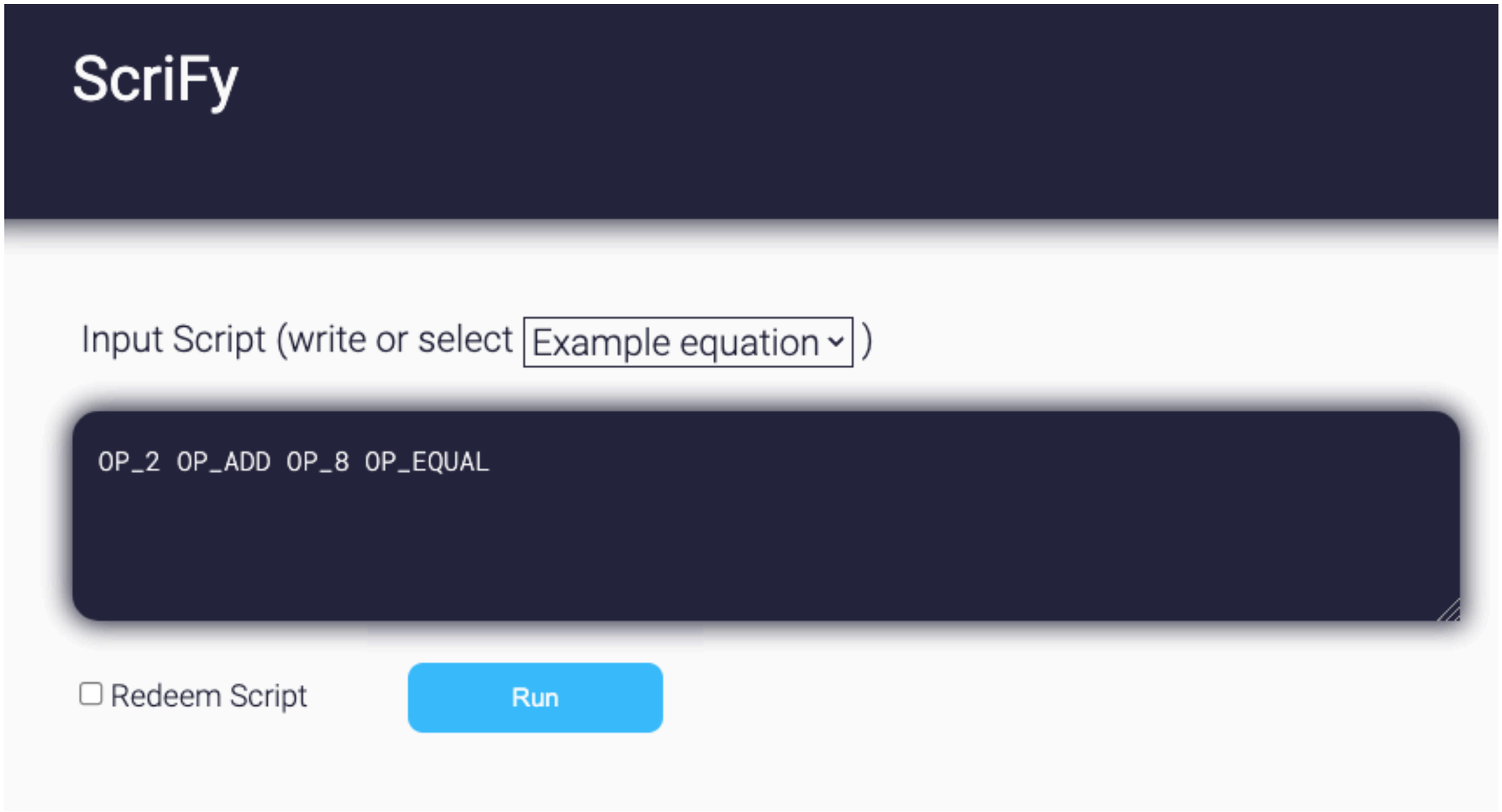
SCRIFY


ScriFy
Input Script (write or select Example P2PKH)
Redeem Script
Run

SCRIFY


ScriFy
Input Script (write or selec
Example P2PKH
Example P2WSH
Example P2SH
Example equation
Redeem Script
Run

SCRIFY

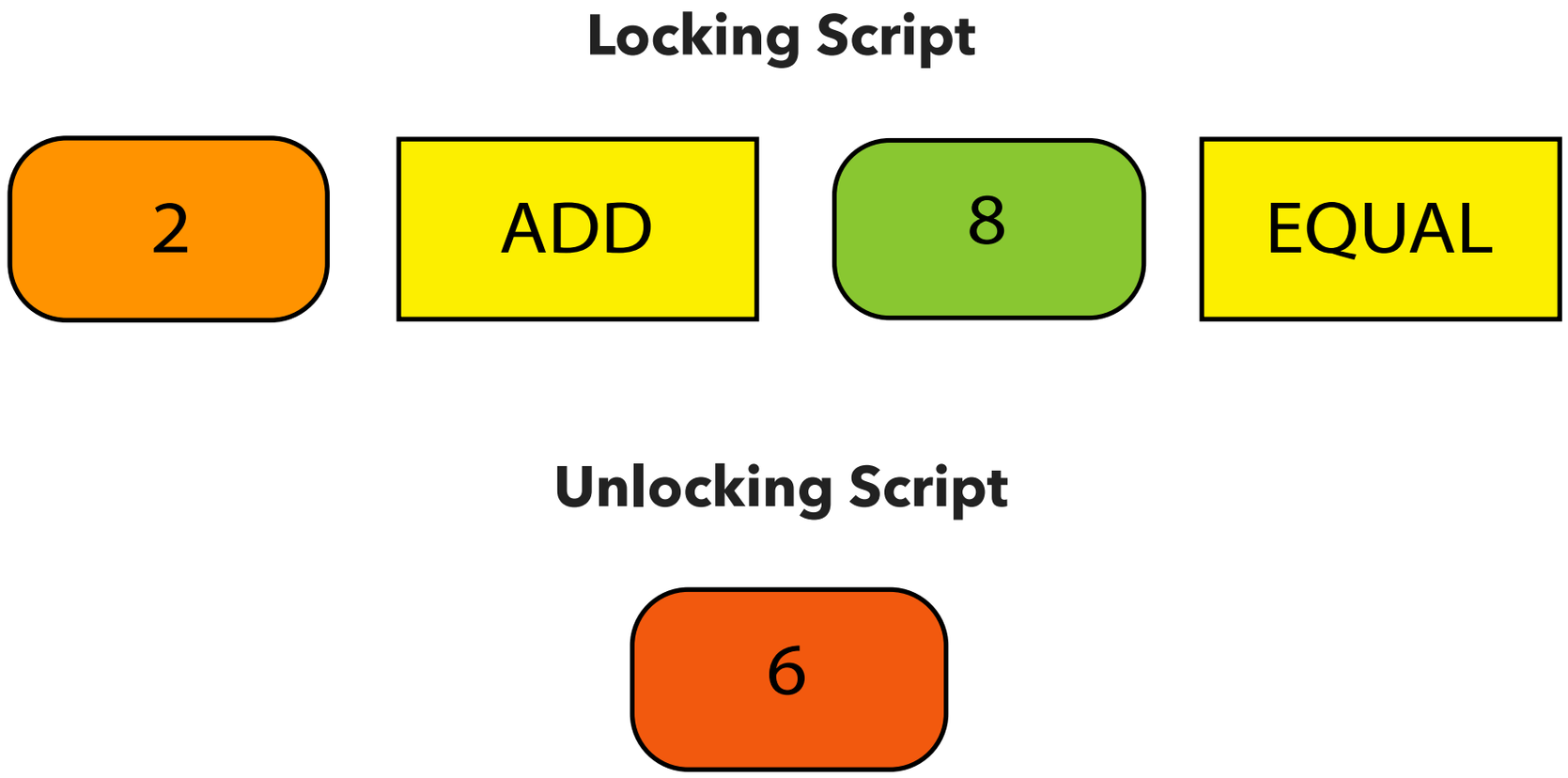

ScriFy
Input Script (write or select Example equation)
OP_2 OP_ADD OP_8 OP_EQUAL
Redeem Script
Run

SCRIFY

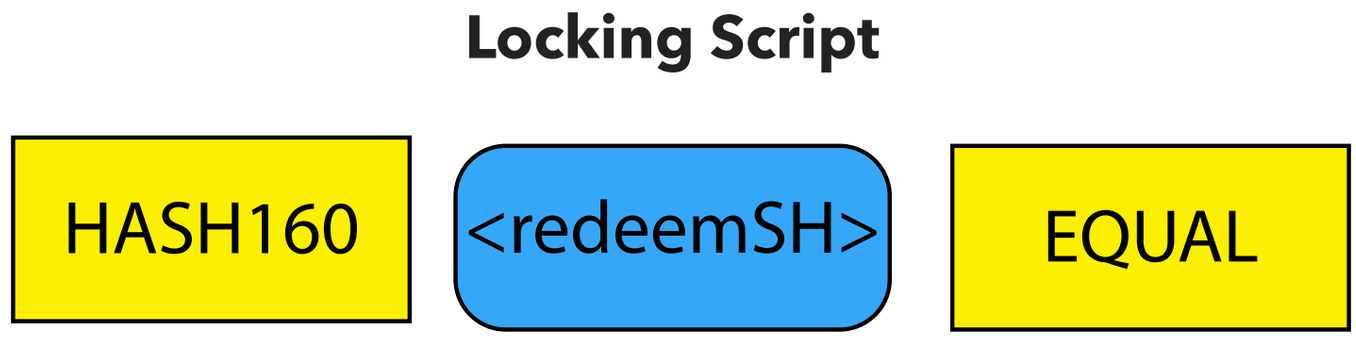
Program Output

```
-------------------------
---------- AST ----------
0:      OP_2;
1:      OP_ADD;
2:      OP_8;
3:      OP_EQUAL;
-------------------------
-------- Inferred -------
---
--- Symbolic evaluation report of execution branch 0
---
Branch's decision points:

Required initial main stack:
        |------------------|
head -> | X_(0)
        |------------------|
Required initial alternative stack:
head -> |------------------|
Inferred constraints:
       Constraint: ((Int 2) + (X_(0))) == (Int 8)
Resulting symbolic stack:
        []

------------------------
-------- Verdict -------
types correct, 1 branch(es) viable
```

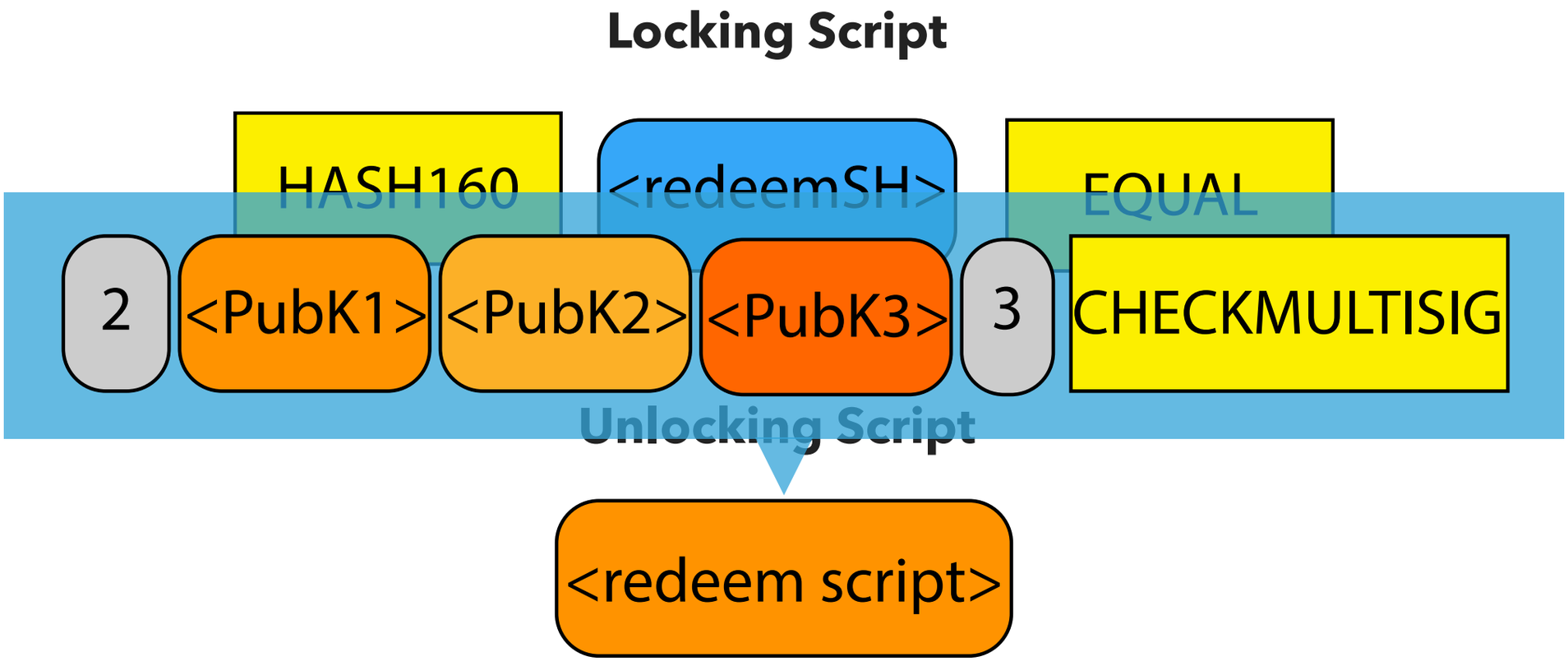
SCRIPT



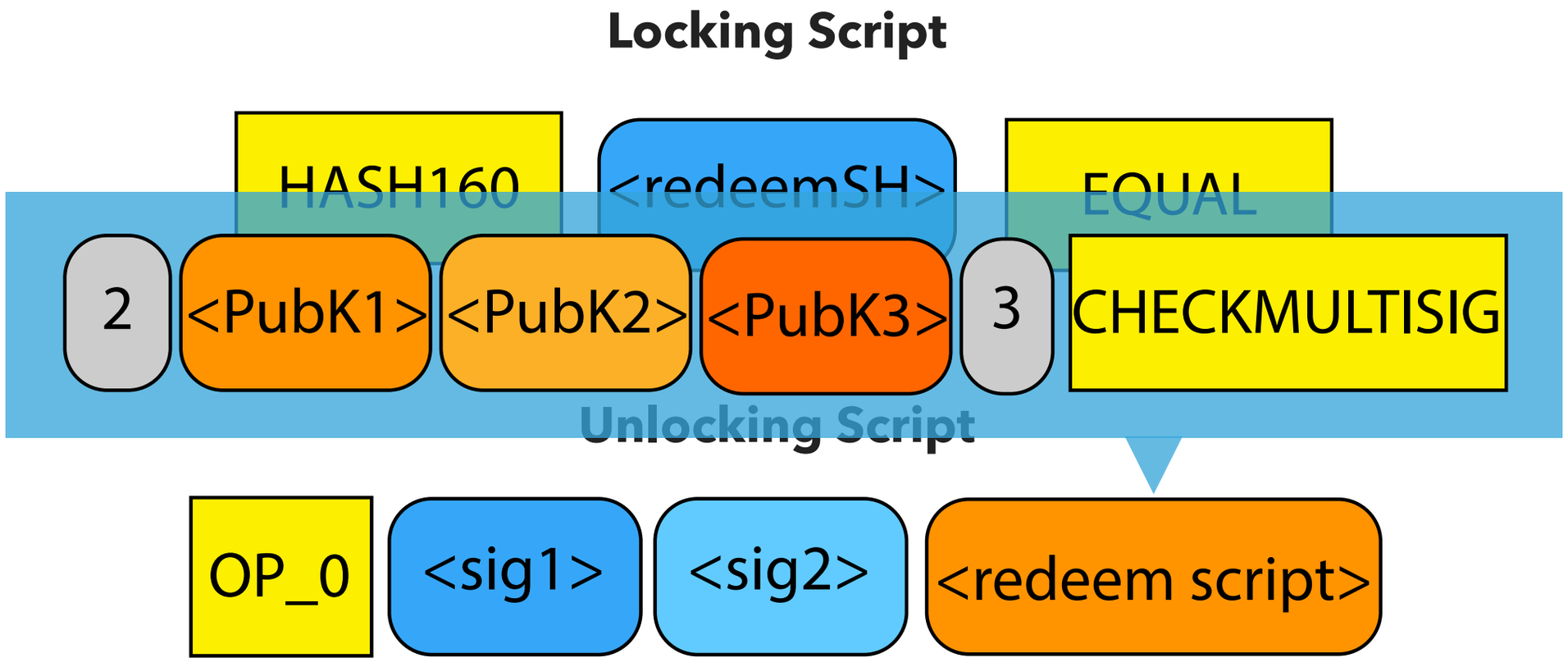
PAY TO SCRIPT HASH (P2SH)



PAY TO SCRIPT HASH (P2SH)



PAY TO SCRIPT HASH (P2SH)



ONLINE TOOL

```
"SPECIAL EXECUTION"
-------------------------
---------- AST ----------
0: OP_P2SH;
1: OP_HASH160;
```

```
OP_HASH160 PUSH a7690b6478b372940e63794204a7690b6478b372 OP_EQUAL
```

```
Required initial main stack:
|------------------|
head -> | X_(0)
| X_(-1)
|------------------|
Required initial alternative stack:
head -> |------------------|
Inferred constraints:
Constraint: (Hash_20 (X_(0))) == (BS_20 "a7690b6478b372940e63794204a7690b6478b372")
Resulting symbolic stack:
[X_(-1) = ExeVerifier_(X_(0))]

------------------------
-------- Verdict -------
types correct, 1 branch(es) viable
```

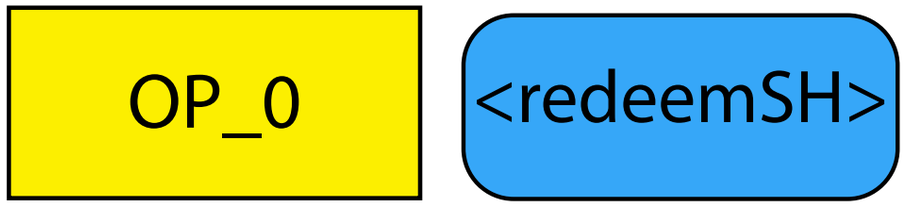
ONLINE TOOL

```
OP_HASH160 PUSH a7690b6478b372940e63794204a7690b6478b372 OP_EQUAL
```

```
"SPECIAL EXECUTION"
-------------------------
---------- AST ----------
0: OP_P2SH;
1: OP_HASH160;
2: BS_20 "a7690b6478b372940e63794204a7690b6478b372";
3: OP_EQUAL;
-------------------------
-------- Inferred -------
---
--- Symbolic evaluation report of execution branch 0
---
Branch's decision points:

Required initial main stack:
|------------------|
head -> | X_(0)
| X_(-1)
|------------------|
Required initial alternative stack:
head -> |------------------|
Inferred constraints:
Constraint: (Hash_20 (X_(0))) == (BS_20 "a7690b6478b372940e63794204a7690b6478b372")
Resulting symbolic stack:
[X_(-1) = ExeVerifier_(X_(0))]

------------------------
-------- Verdict -------
types correct, 1 branch(es) viable
```

PAY TO WITNESS SCRIPT HASH (P2WSH)

Locking Script

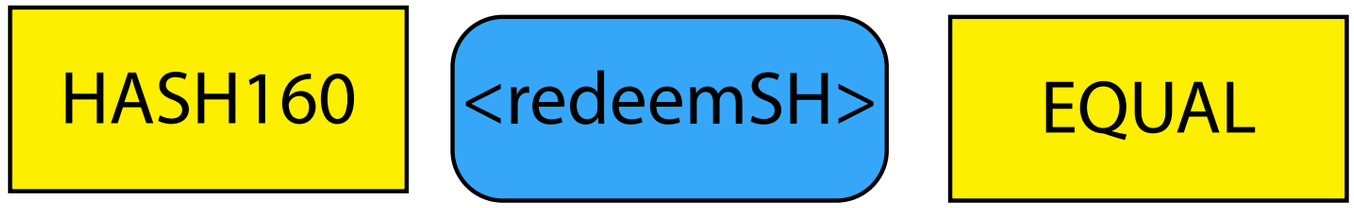


Unlocking Script

Witness script

PAY TO SCRIPT HASH (P2SH)

Locking Script



Unlocking Script

P2WSH

```
"SPECIAL EXECUTION"
-------------------------
---------- AST ----------
0:      OP_P2SH;
1:      OP_SHA256;
```

```
OP_0 PUSH 701a8d401c84fb13e6baf169d59684e17abd9fa216c8cc5b9fc63d622ff8c58d
```

```
Required initial main stack:
        |------------------|
head -> | X_(0)
        | X_(-1)
        |------------------|
Required initial alternative stack:
head -> |------------------|
Inferred constraints:
        Constraint: (Hash_32 (X_(0))) == (BS_32 "701a8d401c84fb13e6baf169d59684e17abd9fa216c8cc5b9fc63d622ff8c58d")
Resulting symbolic stack:
        [X_(-1) = ExeVerifier_(X_(0))]

------------------------
-------- Verdict -------
types correct, 1 branch(es) viable
```

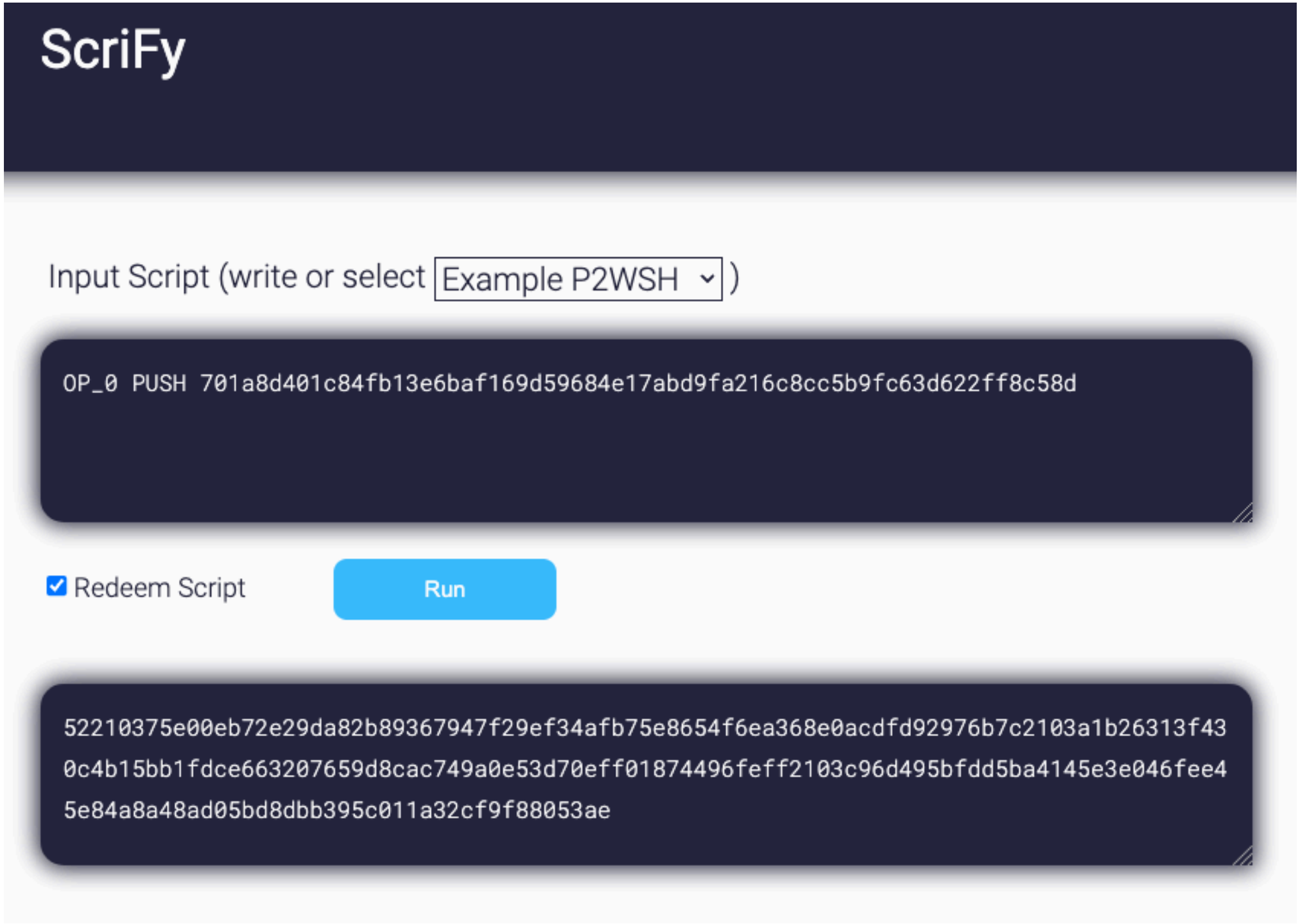
EXAMPLES

P2WSH

```
OP_0 PUSH 701a8d401c84fb13e6baf169d59684e17abd9fa216c8cc5b9fc63d622ff8c58d
```

```
"SPECIAL EXECUTION"
-------------------------
---------- AST ----------
0:      OP_P2SH;
1:      OP_SHA256;
2:      BS_32 "701a8d401c84fb13e6baf169d59684e17abd9fa216c8cc5b9fc63d622ff8c58d";
3:      OP_EQUAL;
-------------------------
-------- Inferred -------
---
--- Symbolic evaluation report of execution branch 0
---
Branch's decision points:

Required initial main stack:
        |------------------|
head -> | X_(0)
        | X_(-1)
        |------------------|
Required initial alternative stack:
head -> |------------------|
Inferred constraints:
        Constraint: (Hash_32 (X_(0))) == (BS_32 "701a8d401c84fb13e6baf169d59684e17abd9fa216c8cc5b9fc63d622ff8c58d")
Resulting symbolic stack:
        [X_(-1) = ExeVerifier_(X_(0))]

------------------------
-------- Verdict -------
types correct, 1 branch(es) viable
```

REDEEM SCRIPT



EXAMPLES

```
------------------------
---------- AST ----------
0:      OP_2;
1:      BS_33 "0375e00eb72e29da82b89367947f29ef34afb75e8654f6ea368e0acdfd92976b7c";
2:      BS_33 "03a1b26313f430c4b15bb1fdce663207659d8cac749a0e53d70eff01874496feff";
3:      BS_33 "03c96d495bfdd5ba4145e3e046fee45e84a8a48ad05bd8dbb395c011a32cf9f880";
4:      OP_3;
5:      OP_CHECKMULTISIG;
6:      BS_32 "701a8d401c84fb13e6baf169d59684e17abd9fa216c8cc5b9fc63d622ff8c58d";
7:      OP_SHA256;
8:      BS_32 "701a8d401c84fb13e6baf169d59684e17abd9fa216c8cc5b9fc63d622ff8c58d";
9:      OP_EQUAL;
------------------------
-------- Inferred -------
---
--- Symbolic evaluation report of execution branch 0
---
Branch's decision points:

Required initial main stack:
        |-----------------|
head -> | X_(0)
        | X_(-1)
        | X_(-2)
        |-----------------|
Required initial alternative stack:
head -> |-----------------|
Inferred constraints:
        Constraint: (Hash_32 (BS_32 "701a8d401c84fb13e6baf169d59684e17abd9fa216c8cc5b9fc63d622ff8c58d")) == (BS_32
"701a8d401c84fb13e6baf169d59684e17abd9fa216c8cc5b9fc63d622ff8c58d")
        Constraint: (X_(-2)) == (Int 0)
Resulting symbolic stack:
        [MultiSig [X_(-1),X_(0)] [BS_33 "0375e00eb72e29da82b89367947f29ef34afb75e8654f6ea368e0acdfd92976b7c",BS_33
"03a1b26313f430c4b15bb1fdce663207659d8cac749a0e53d70eff01874496feff",BS_33
"03c96d495bfdd5ba4145e3e046fee45e84a8a48ad05bd8dbb395c011a32cf9f880"]]

------------------------
-------- Verdict -------
types correct, 1 branch(es) viable
```

CONCLUSION AND FUTURE WORKS

CONCLUSION AND FUTURE WORKS

- We design a on online tool for symbolic model checker script

scrify.dmi.unipg.it

CONCLUSION AND FUTURE WORKS

- We design a on online tool for symbolic model checker script
- We includes several special cases

scrify.dmi.unipg.it

CONCLUSION AND FUTURE WORKS

- We design a on online tool for symbolic model checker script
- We includes several special cases
- Extend to multi-step protocols

scrify.dmi.unipg.it

Scrify: an online tool to validate Bitcoin script

Stefano Bistarelli, Andrea Bracciali, **Ivan Mercanti**

THANKS FOR THE ATTENTION.

QUESTIONS?

DLT 23 - 26/05/2023