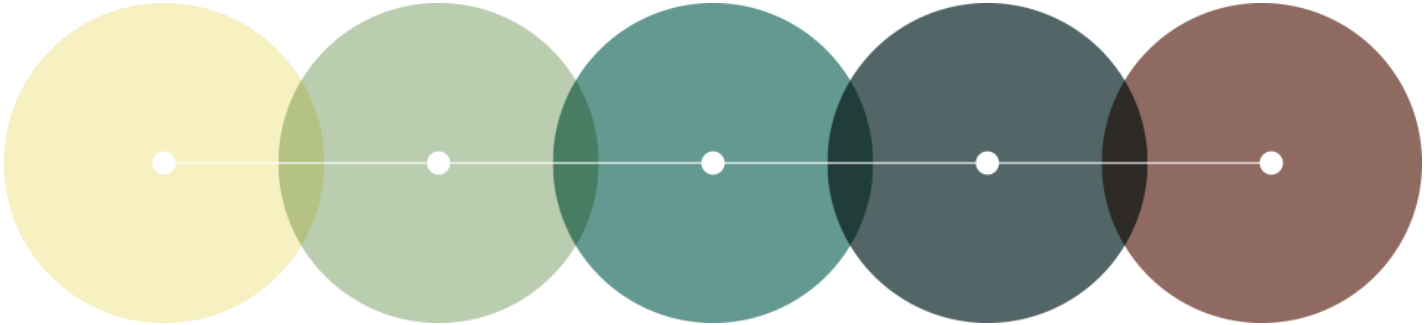
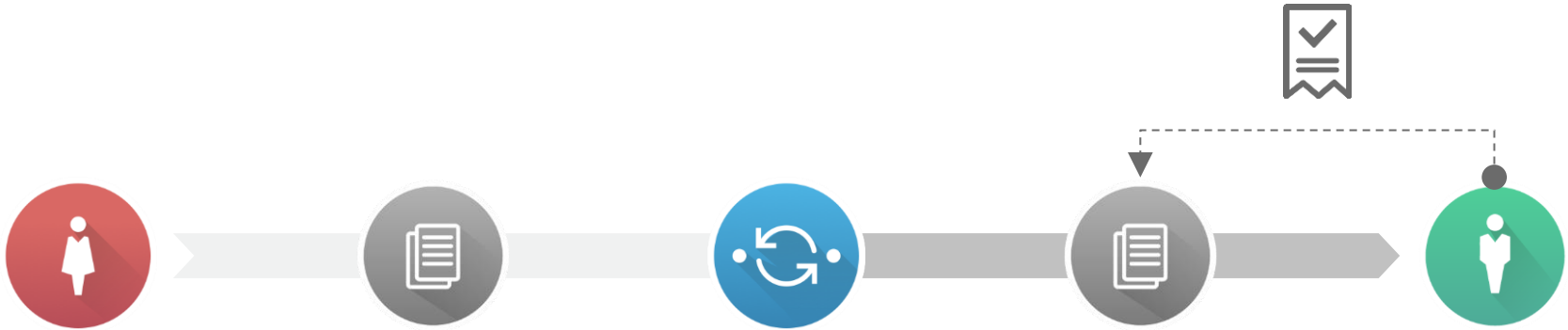


Crypto Conditions

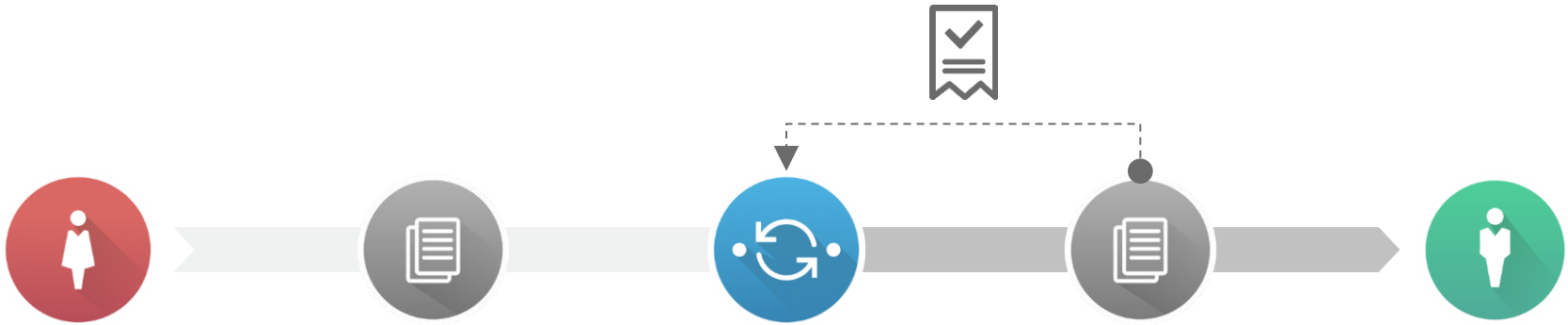
A Solid Foundation for ILP

Stefan Thomas

The Receipt

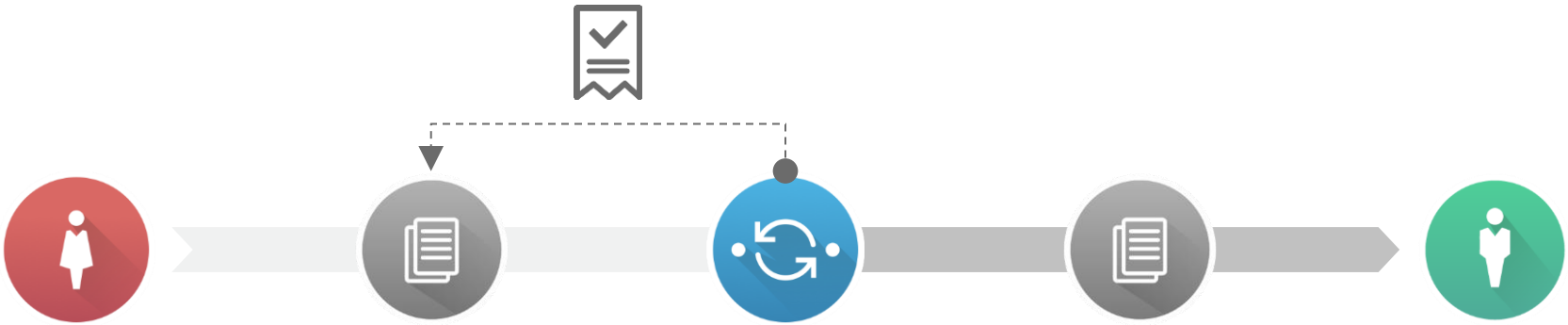


Connector gets receipt from ledger



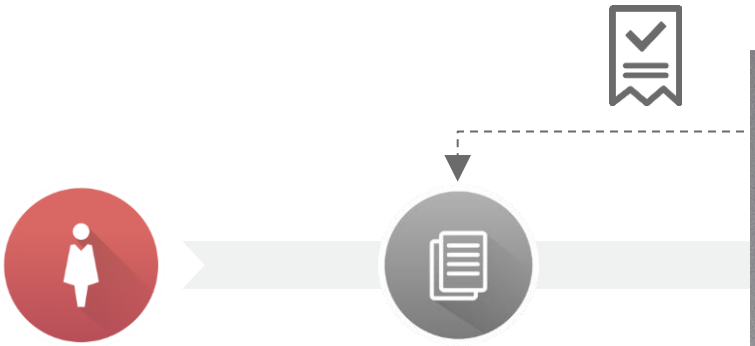
Umm...

Connector passes on the receipt

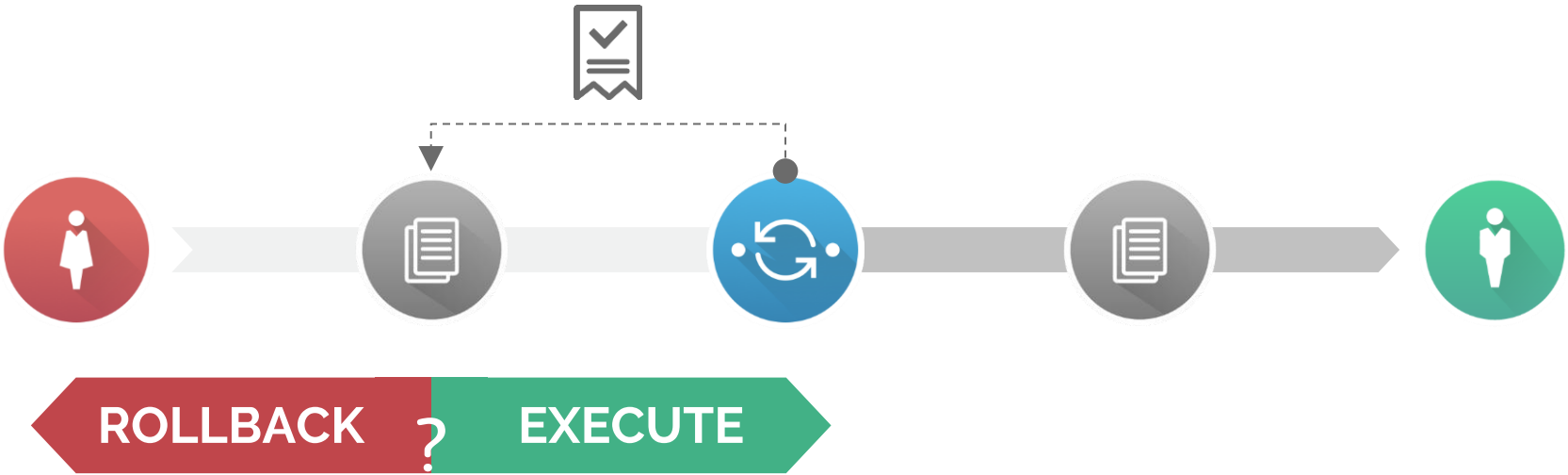


Excuse me…

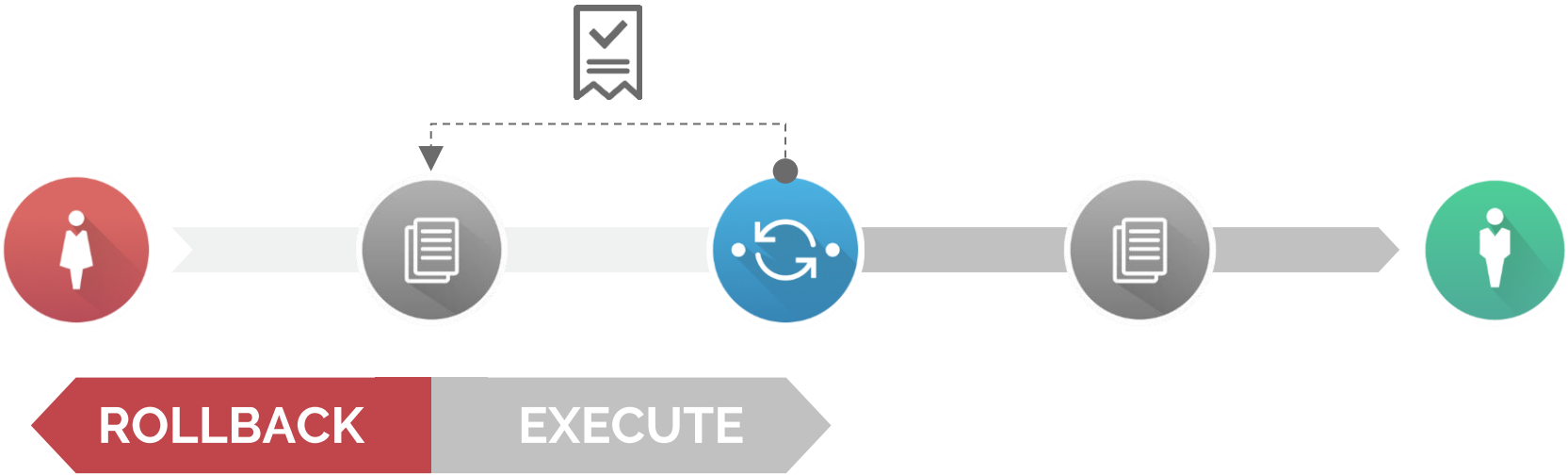
Connector passes on the receipt



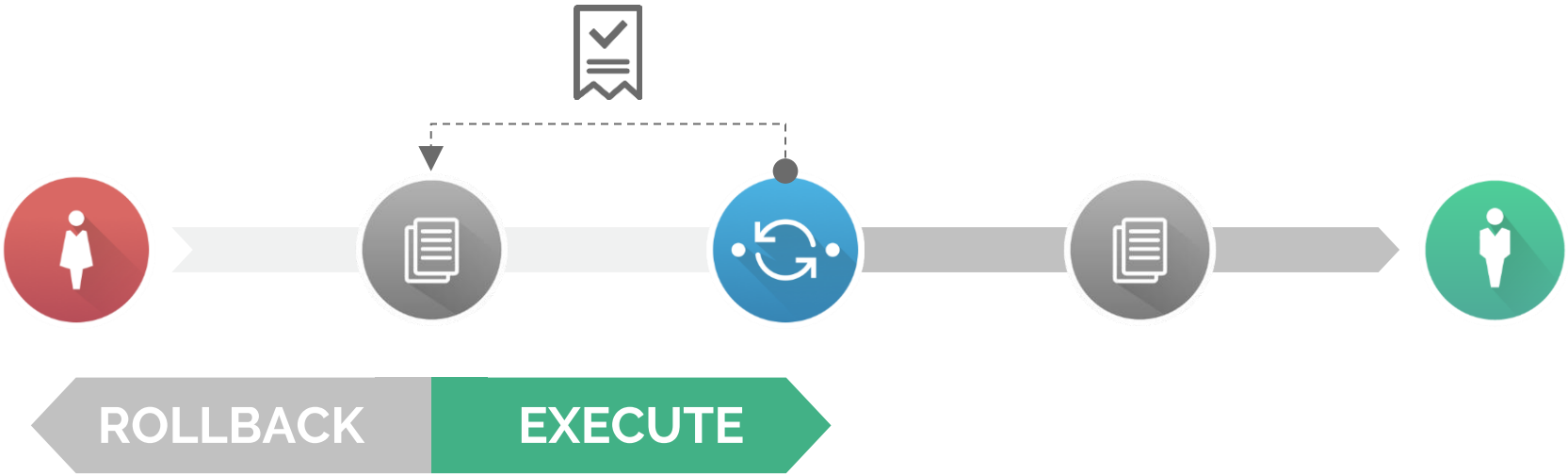
Ledger Needs to Decide



Rollback on **failure**



Execute on **success**



What is **success**?

As a sender I want to be certain

As a sender I want to be certain
that my donation was received

As a sender I want to be certain
that the invoice was paid

As a sender I want to be certain

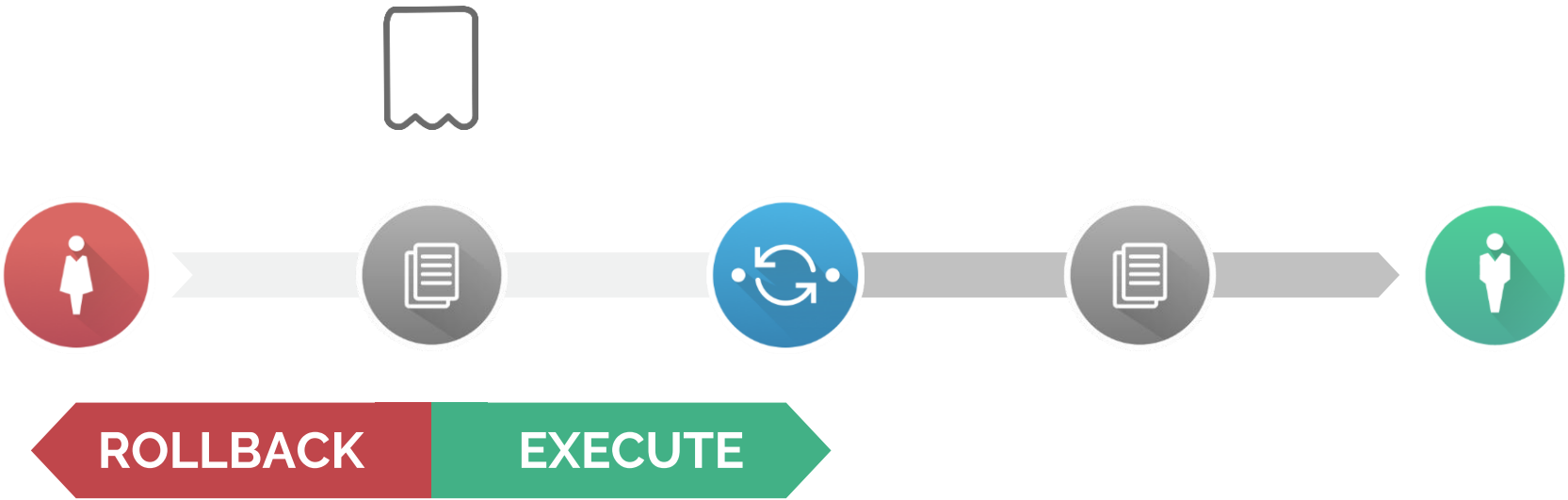
that my debt is settled

As a sender I want
non-repudiable
proof-of-settlement

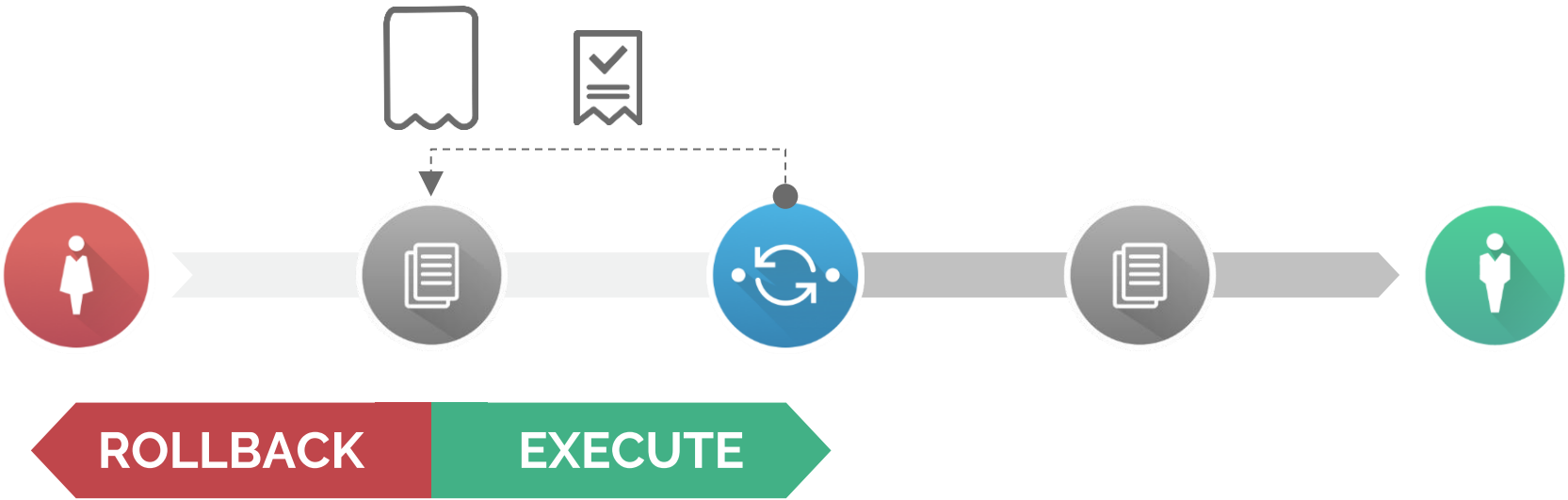
The Receipt

- Signed by recipient
- Proves receipt-of-funds
- Non-repudiable
- Might have data attached
- Pre-agreed before payment

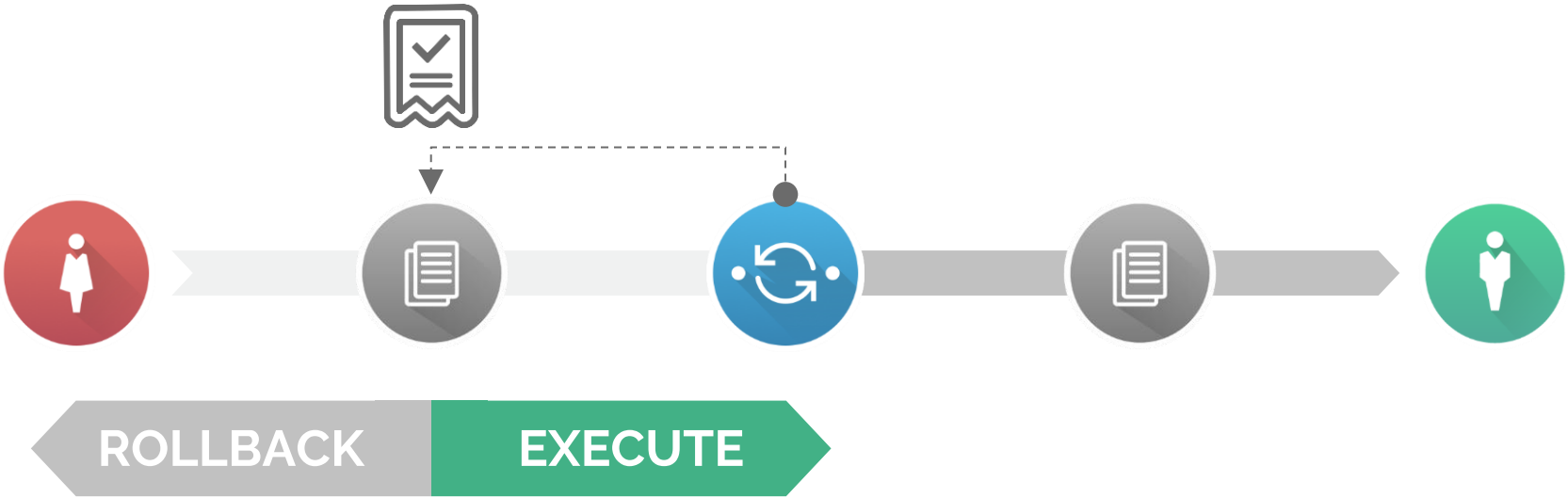
How do we describe the receipt?



How do we describe the receipt?

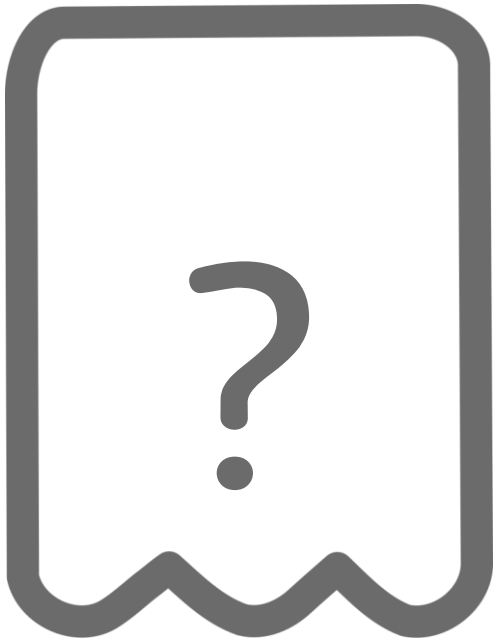


How do we describe the receipt?



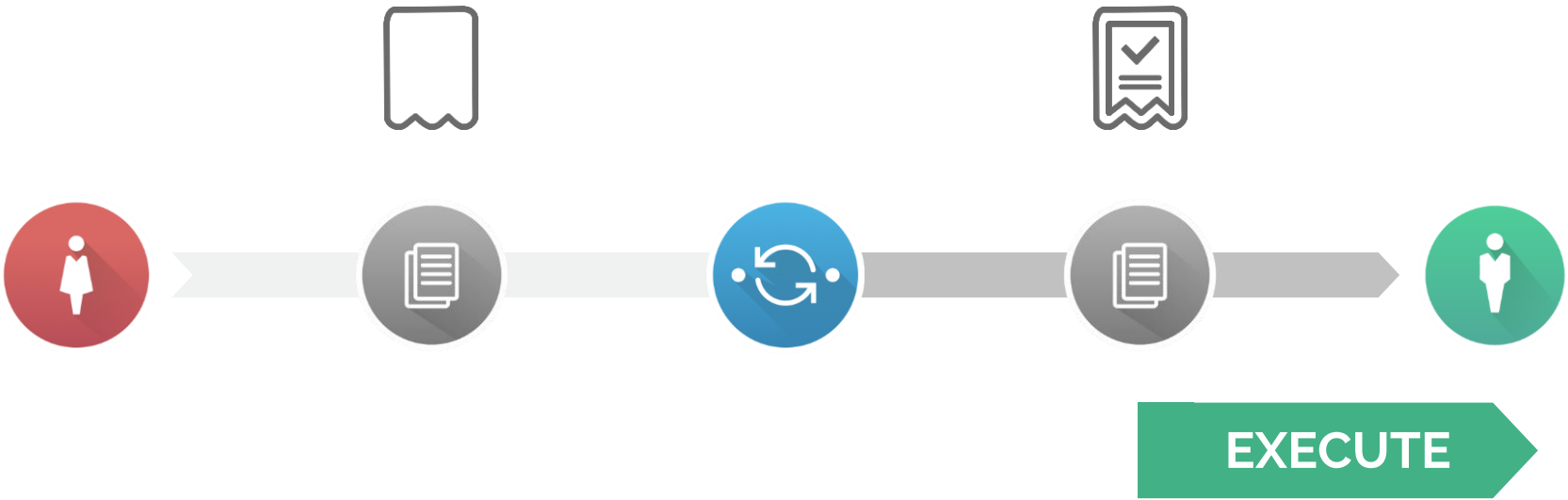
Condition

- Describes some signed message
- *Fulfilled* by that message

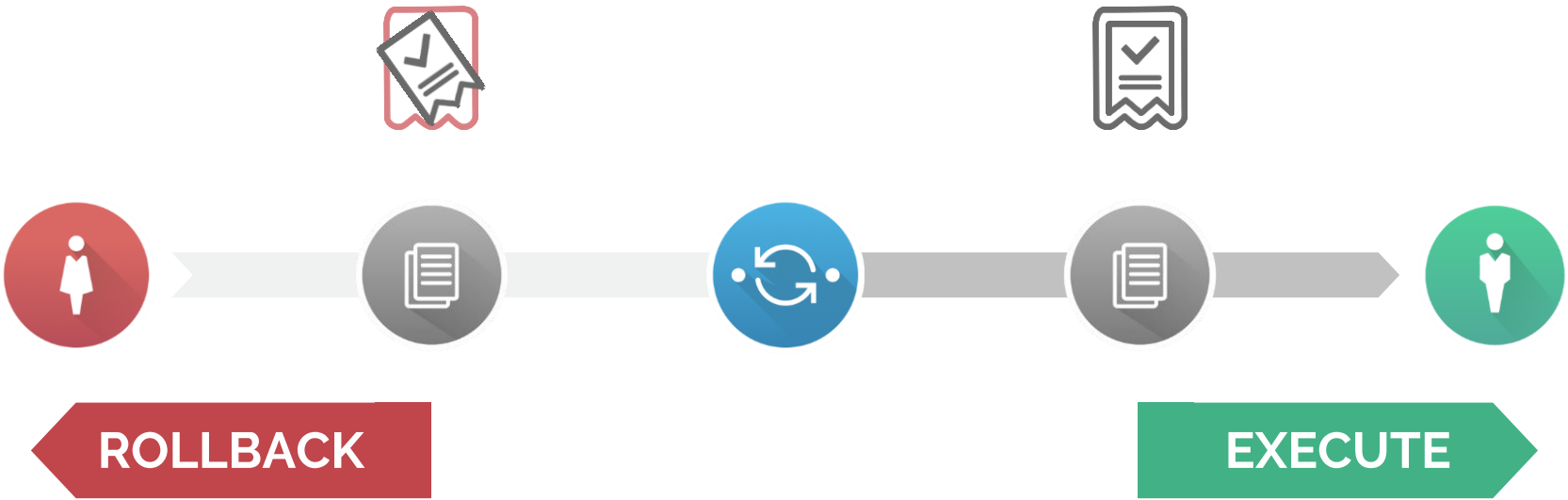


Why Are Conditions so Important?

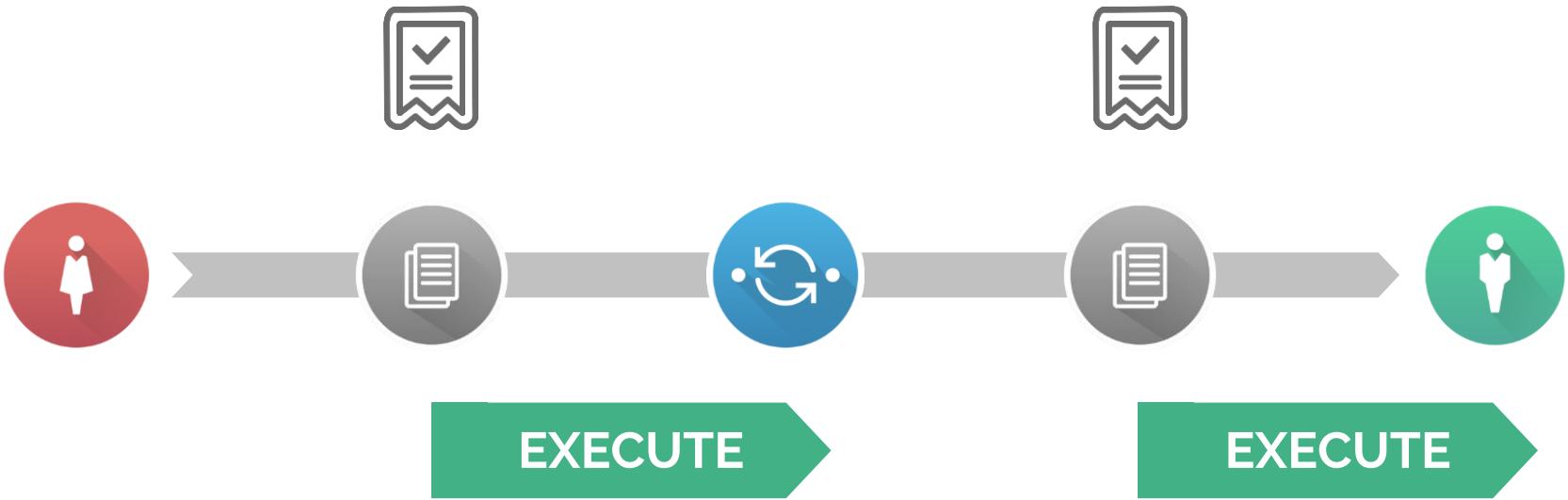
Why Are Conditions so Important?



Why Are Conditions so Important?



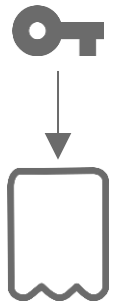
Conditions Must Be **Bit-Perfect**



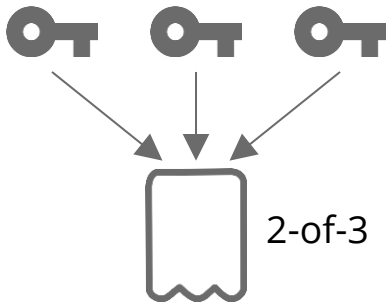
Others Have Done The Hard Part

Are Simple Signatures Enough?

Single Signature Condition



Multi Signature Condition



Prior Art: Bitcoin Scripts

```
2 <K1> <K2> <K3> 3 CHECKMULTISIGVERIFY
```

- Forth-like language
- Many opcodes disabled
- Primary use case: m-of-n multi-signature

Other Ideas

- Pay-to-script-hash — Andresen
- Tree Signatures — Wuille
- Merkleized Abstract Syntax Trees (MAST) — Rubin et al
- Script2 — Blockstream
- Smart Signatures — Allen et al
- State Channels — Coleman

Security vs Flexibility

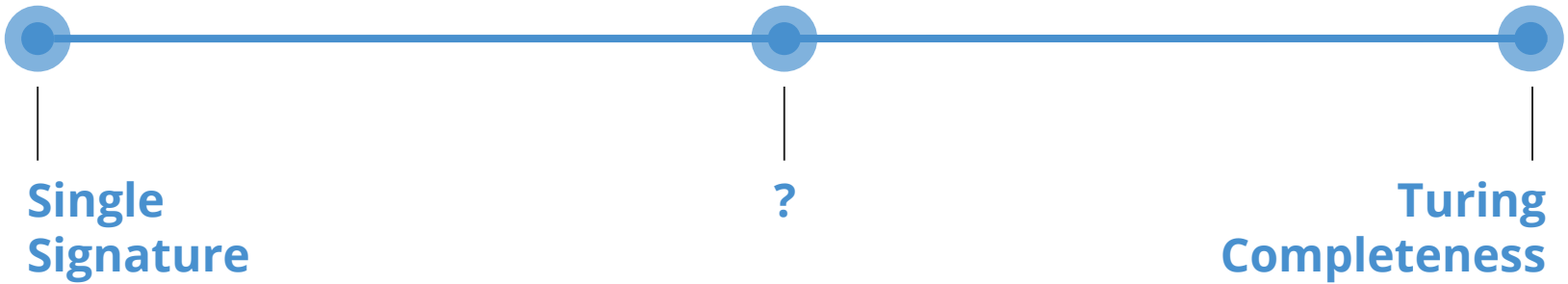
Security vs Flexibility

Bit-perfect, standardized programming language

Security vs Flexibility

"ActiveX of blockchain"

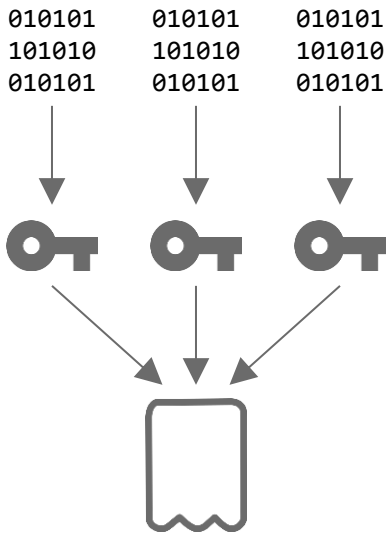
Security vs Flexibility



Smart Oracles

Smart Oracle

Smart Oracle



Delegation

Single Signature

If **Bob** says **yes**,
then **yes**.

2-of-2 Multi Signature

If **Bob** and **Lina** say **yes**,
then **yes**.

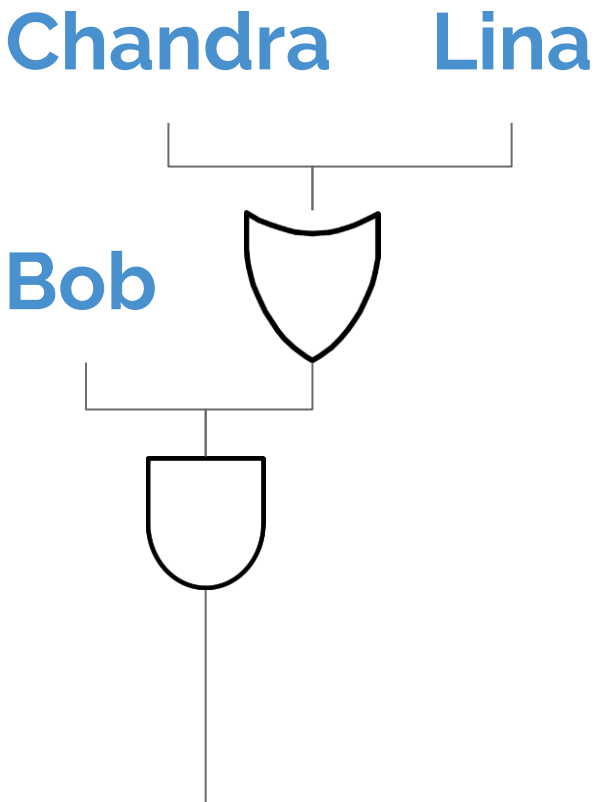
2-of-3 Multi Signature

If **any two** of **Bob**, **Chandra**, **Lina** say **yes**, then **yes**.

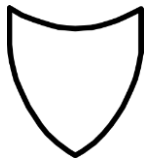
2-of-3 Multi Signature?

If **Bob** and (**Chandra** or **Lina**) say **yes**, then **yes**.

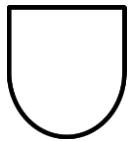
Boolean Circuits



Boolean Circuits

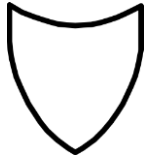


OR

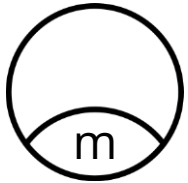


AND

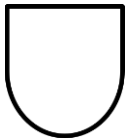
Boolean Circuits



OR

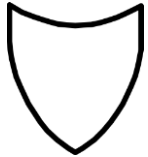


m-of-n

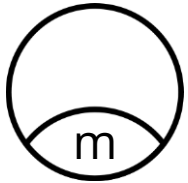


AND

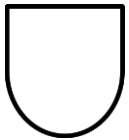
Boolean Circuits



1-of-n

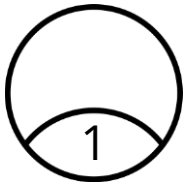


m-of-n

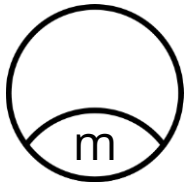


n-of-n

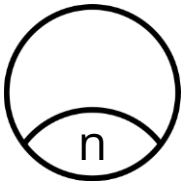
Boolean Circuits



1-of-n



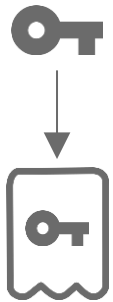
m-of-n



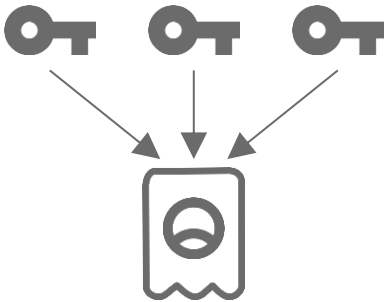
n-of-n

Putting it all together

Condition Types

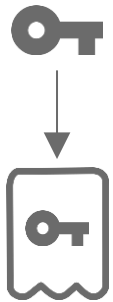


Signatures

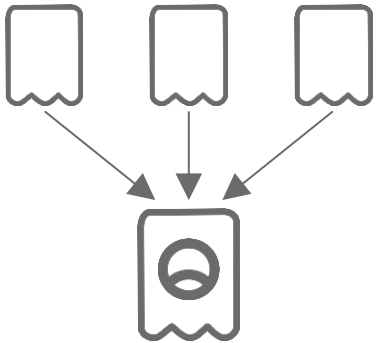


Thresholds

Condition Types



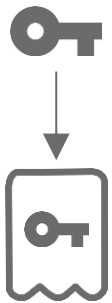
Signatures



Thresholds

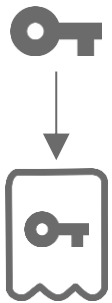
Fulfillment

Signature Condition



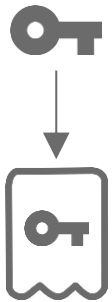
```
CONDITION =
  VARUINT BITMASK = 2
  UINT256 HASH
  VARUINT MAX_FULLFILLMENT_LENGTH
```

Signature Condition



```
CONDITION =
  VARUINT BITMASK = 2
  UINT256 HASH
  VARUINT MAX_FULLFILLMENT_LENGTH

HASH = SHA256(
  SHA256("https://...#rsa-sha-256")
  VARSTR MODULUS
  VARSTR MESSAGE_PREFIX
)
```

Signature Condition



```
cc:1:Aml_0BgDRVk32fXTVMQ6V1SSKFrxAF7XHegeI16vTaexsgI
```

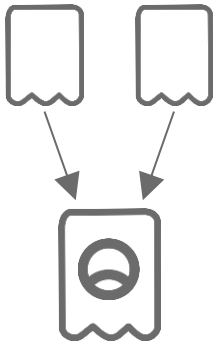
Signature Fulfillment

```
FULFILLMENT =
  VARUINT BITMASK = 2
  VARSTR MODULUS
  VARSTR MESSAGE_PREFIX
  VARSTR MESSAGE
  UINT8[LENGTH(MODULUS)] SIGNATURE
  VARUINT BYTES_UNUSED
```

Signature Fulfillment

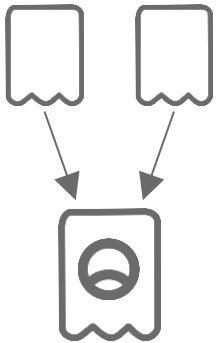
cf:1:AoABYTEyNWNiZGFmNWI3NDk0MzQ5YjE2NGUxMmRjZTRiND
BkMTI4MTNkYTY1ZDM4YTEyOTNmZDFhOWMwMTk2YzJlZjRmYWRhN
jI2OWNjYzFhNzdjMTZhYjc2NmRhMGU0NzYxYzQ4Mjc1Y2U4MzNm
OGE5MzdkOWMyOWQzZDVlNmQyZTkMSGVsbG8gd29ybGQhFSBDb25
kaXRpb25zIGFyZSBoZXJlIUFLExFQC5PXoLli_CBpIEE0WOypkZ
L7wCf_eIWExJJEmdrt00ubso94DO7gIOLUTh5sRCuij29Yos1b_
yE79gwA

Threshold Condition



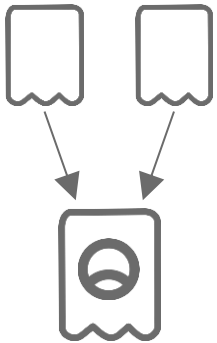
```
CONDITION =
  VARUINT BITMASK
  UINT256 HASH
  VARUINT MAX_FULLFILLMENT_LENGTH
```

Threshold Condition



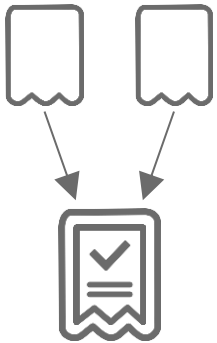
```
CONDITION =
  VARUINT BITMASK
  UINT256 HASH
  VARUINT MAX_FULLFILLMENT_LENGTH

HASH = SHA256(
  SHA-256("https://...#threshold-sha-256")
  VARUINT THRESHOLD
  VARUINT NUM_ELEMENTS
  FOR EACH ELEMENT
    ELEMENT_CONDITION
)
```

Threshold Condition



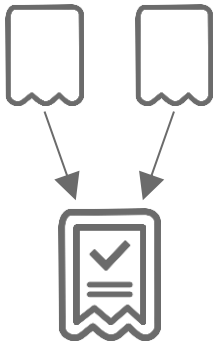
`cc:1:AR9QCVepDTNSct6SGWTx6cWfLgBrp8tNqWIjylAiEy4vtAI`

Threshold Fulfillment



```
FULFILLMENT =
  VARUINT BITMASK
  VARUINT THRESHOLD
  VARUINT NUM_ELEMENTS
  FOR EACH ELEMENT
    VARUINT PARAMS
    ELEMENT_CONDITION/FULFILLMENT
```

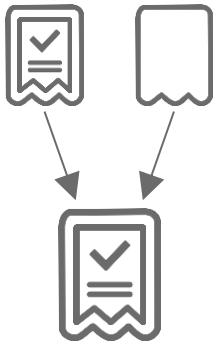
Threshold Fulfillment




```
cf:1:AQICjYwvnzkRBOa8QDKcPC_c6Sh-LnKPfu-
COmv9j2e3ScuzAgKAAWExMjVjYmRhZjViNzQ5NDM0OWIxNjRlMTJk
Y2U0YjQwZDEyODEzZGE2NWQzOGExMjkzZmQxYTljMDE5NmMyZWY0Z
mFkYTYyNjljY2MxYTc3YzE2YWI3NjZkYTBlNDc2MWM0ODI3NWNlOD
MzZjhhOTM3ZDljMjlkM2Q1ZTZkMmU5DEhlbGxvIHdvcmxkIRUgQ29
uZGl0aW9ucyBhcmUgaGVyZSFBSxMRUAuT16C5YvwgaSBBNFjsqZGS
-
8An_3iFhMSSRJna7dNLm7KPeAzu4CDi1E4ebEQroo9vWKLNW_8hO_
YMAA
```

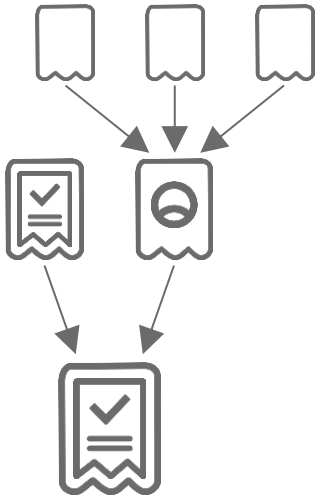
Merkle Circuits

Threshold Fulfillment



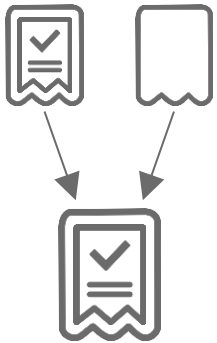

```
cf:1:AQICjYwvnzkRBOa8QDKcPC_c6Sh-LnKPfu-
COmv9j2e3ScuzAgKAAWExMjVjYmRhZjViNzQ5NDM0OWIxNjRlMTJk
Y2U0YjQwZDEyODEzZGE2NWQzOGExMjkzZmQxYTljMDE5NmMyZWY0Z
mFkYTYyNjljY2MxYTc3YzE2YWI3NjZkYTBlNDc2MWM0ODI3NWNlOD
MzZjhhOTM3ZDljMjlkM2Q1ZTZkMmU5DEhlbGxvIHdvcmxkIRUgQ29
uZGl0aW9ucyBhcmUgaGVyZSFBSxMRUAuT16C5YvwgaSBBNFjsqZGS
-
8An_3iFhMSSRJna7dNLm7KPeAzu4CDi1E4ebEQroo9vWKLNW_8hO_
YMAA
```

Threshold Fulfillment




```
cf:1:AQICjYwvnzkRBOa8QDKcPC_c6Sh-LnKPfu-
COmv9j2e3ScuzAgKAAWExMjVjYmRhZjViNzQ5NDM0OWIxNjRlMTJk
Y2U0YjQwZDEyODEzZGE2NWQzOGExMjkzZmQxYTljMDE5NmMyZWY0Z
mFkYTYyNjljY2MxYTc3YzE2YWI3NjZkYTBlNDc2MWM0ODI3NWNlOD
MzZjhhOTM3ZDljMjlkM2Q1ZTZkMmU5DEhlbGxvIHdvcmxkIRUgQ29
uZGl0aW9ucyBhcmUgaGVyZSFBSxMRUAuT16C5YvwgaSBBNFjsqZGS
-
8An_3iFhMSSRJna7dNLm7KPeAzu4CDi1E4ebEQroo9vWKLNW_8hO_
YMAA
```

Threshold Fulfillment



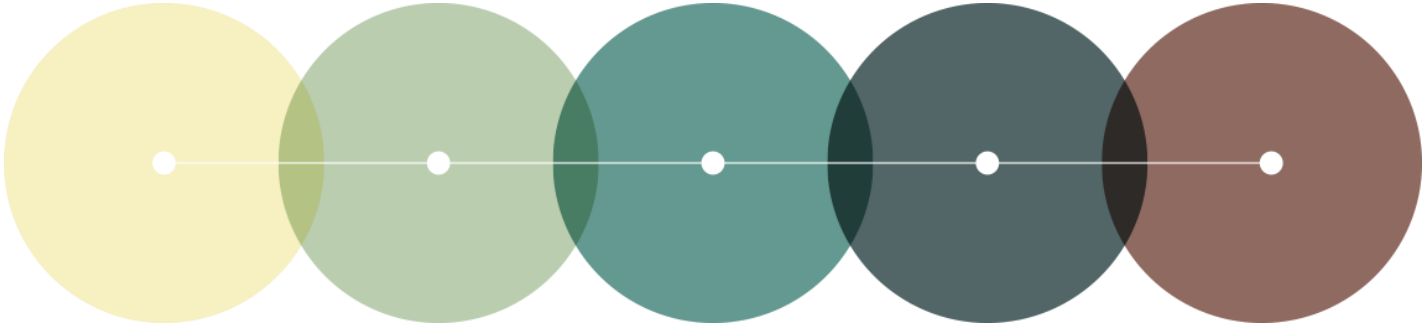

```
cf:1:AQICjYwvnzkRBOa8QDKcPC_c6Sh-LnKPfu-
COmv9j2e3ScuzAgKAAWExMjVjYmRhZjViNzQ5NDM0OWIxNjRlMTJk
Y2U0YjQwZDEyODEzZGE2NWQzOGExMjkzZmQxYTljMDE5NmMyZWY0Z
mFkYTYyNjljY2MxYTc3YzE2YWI3NjZkYTBlNDc2MWM0ODI3NWNlOD
MzZjhhOTM3ZDljMjlkM2Q1ZTZkMmU5DEhlbGxvIHdvcmxkIRUgQ29
uZGl0aW9ucyBhcmUgaGVyZSFBSxMRUAuT16C5YvwgaSBBNFjsqZGS
-
8An_3iFhMSSRJna7dNLm7KPeAzu4CDi1E4ebEQroo9vWKLNW_8hO_
YMAA
```

Recap

- Two Condition Types
  - Signature
  - Threshold
- Conditions are constant size
- Falsy branches can be omitted
- Complex logic is delegated

Other Features

- Conditions can be generated from fulfillments
- Max fulfillment length in condition
- Extensible with new crypto primitives
- Required feature set in condition



First Editor's Draft Coming Soon

Interledger.org