



BLOCKCHAIN COMMONS

**GORDIAN ENVELOPE
REQUEST & RESPONSE**

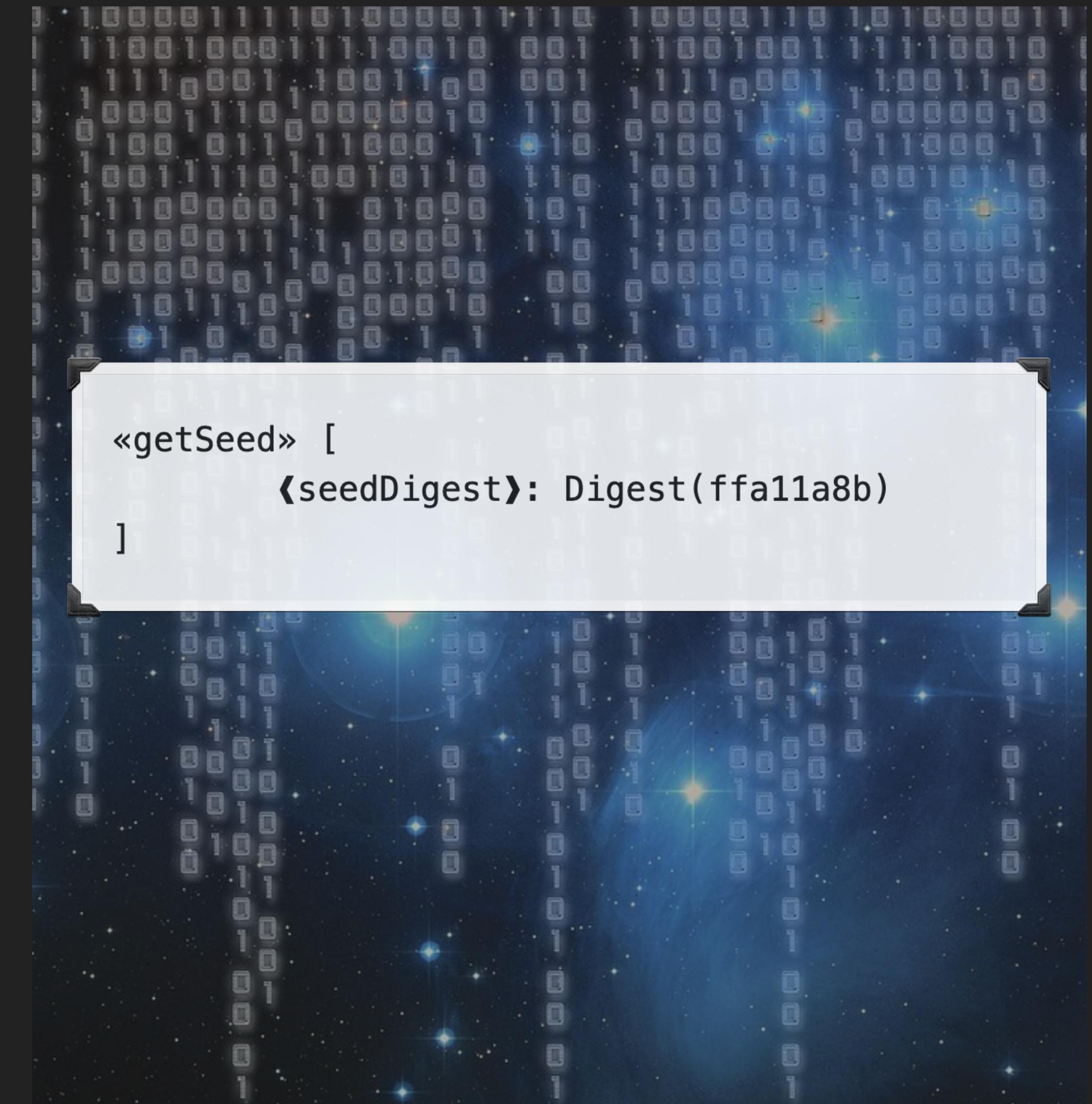
WHAT IS REQUEST & RESPONSE?

- ▶ Gordian Envelope Functionality
- ▶ One Device Can Request from Another
- ▶ See the Implementation Guide
 - ▶ BCR 2024-04
 - ▶ <https://tinyurl.com/bcr-2024-004>



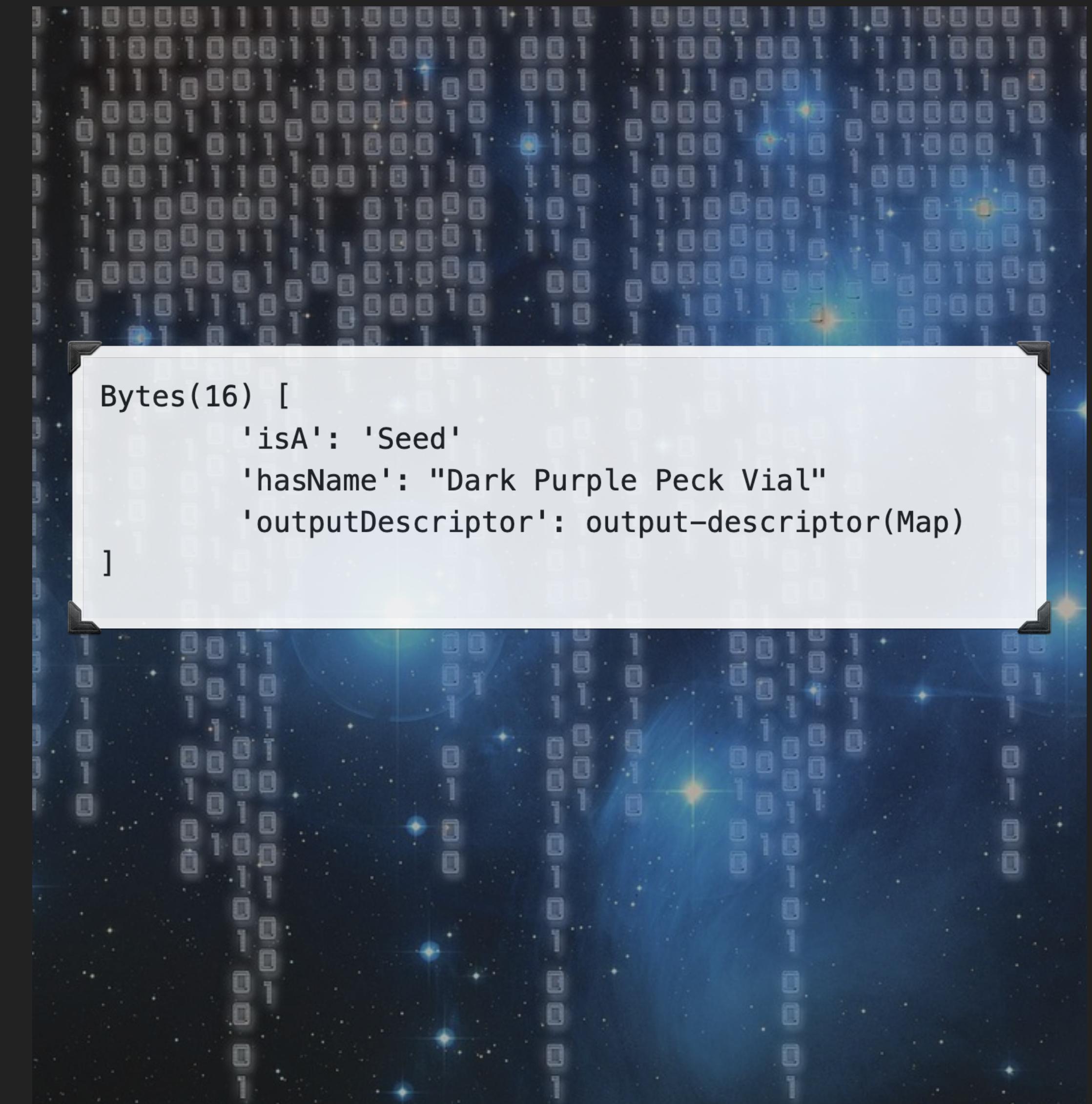
REQUESTS ARE EXPRESSIONS

- ▶ Requests are Built on Expressions
- ▶ A Function
- ▶ With Zero or More Parameters
- ▶ Each of Which Has an Argument



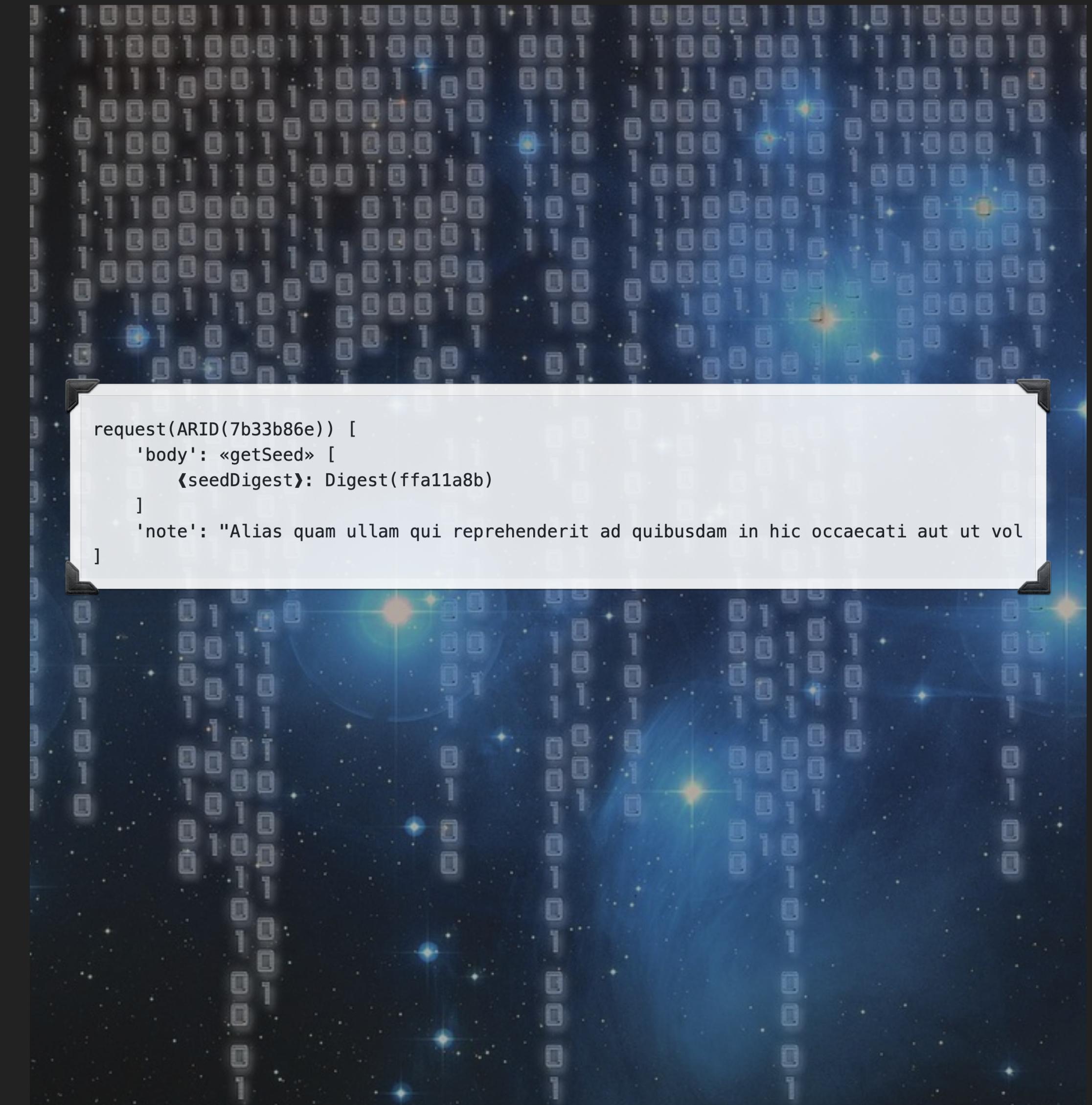
RESULTS ARE DATA

- ▶ Results Return the Requested Object
 - ▶ With a Defining `isA`
 - ▶ And Maybe Other Metadata



A WRAPPED REQUEST

- ▶ Expressions and Results are Wrapped
- ▶ Marked with an ARID (Random ID)
- ▶ Expression is `body`
- ▶ Maybe Metadata Like a `note`
- ▶ This is How Expression & Data
- ▶ Become a Request & Response



A WRAPPED RESPONSE

- ▶ Responses are Wrapped Too
 - ▶ The Same ARID is Used
 - ▶ Data is `result`

```
response(ARID(7b33b86e)) [  
    'result': Bytes(16) [  
        'isA': 'Seed'  
        'hasName': "Dark Purple Peck Vial"  
        'outputDescriptor': output-descriptor(Map)  
    ]  
]
```

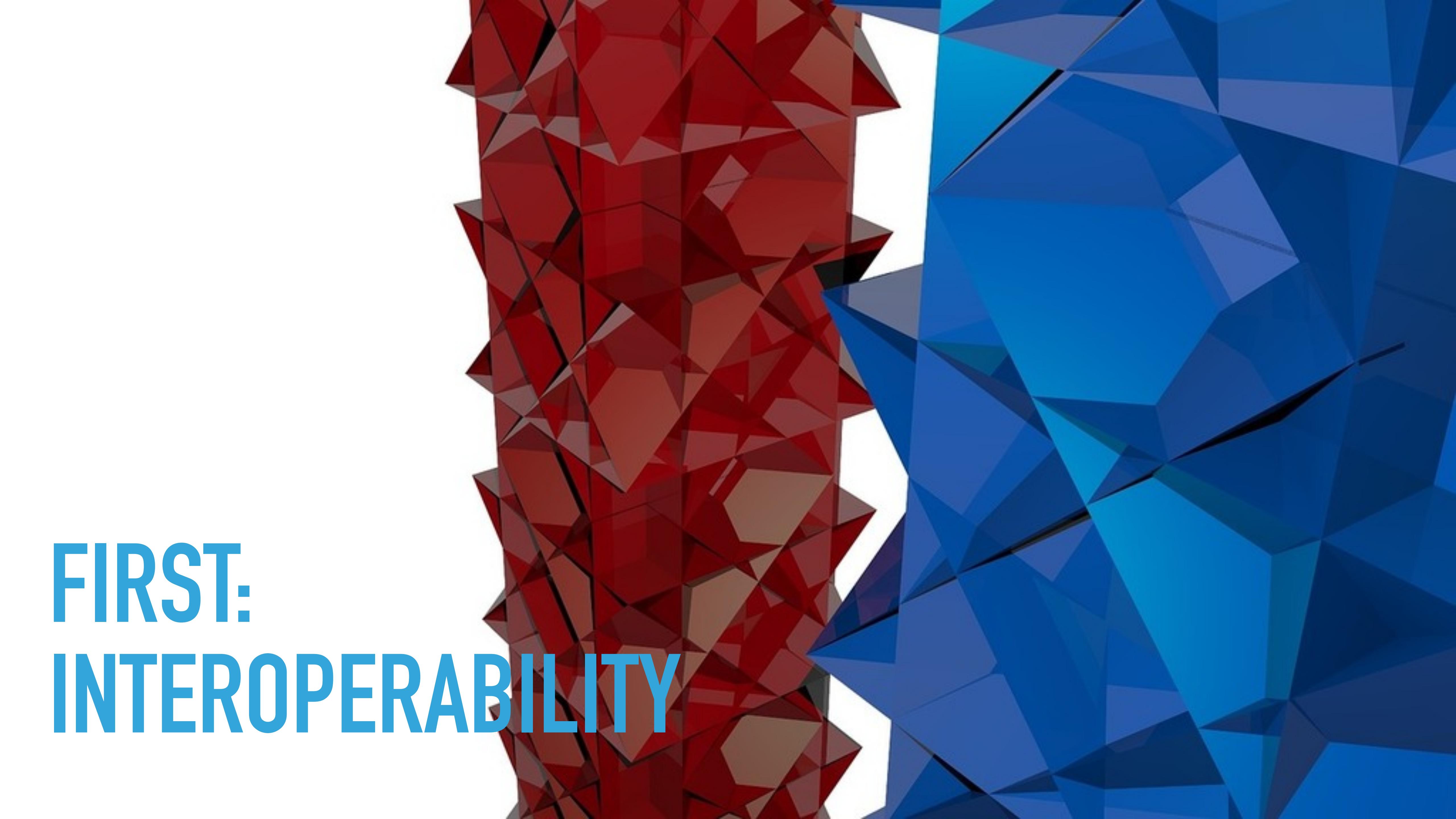
THERE ARE DETAILS

- ▶ Request/Response are Coded with CBOR tags
- ▶ `function`/`parameter` are Defined with Values
- ▶ Everything Uses Known Values
- ▶ It's All in the Implementation Guide!
- ▶ <https://tinyurl.com/bcr-2024-004>

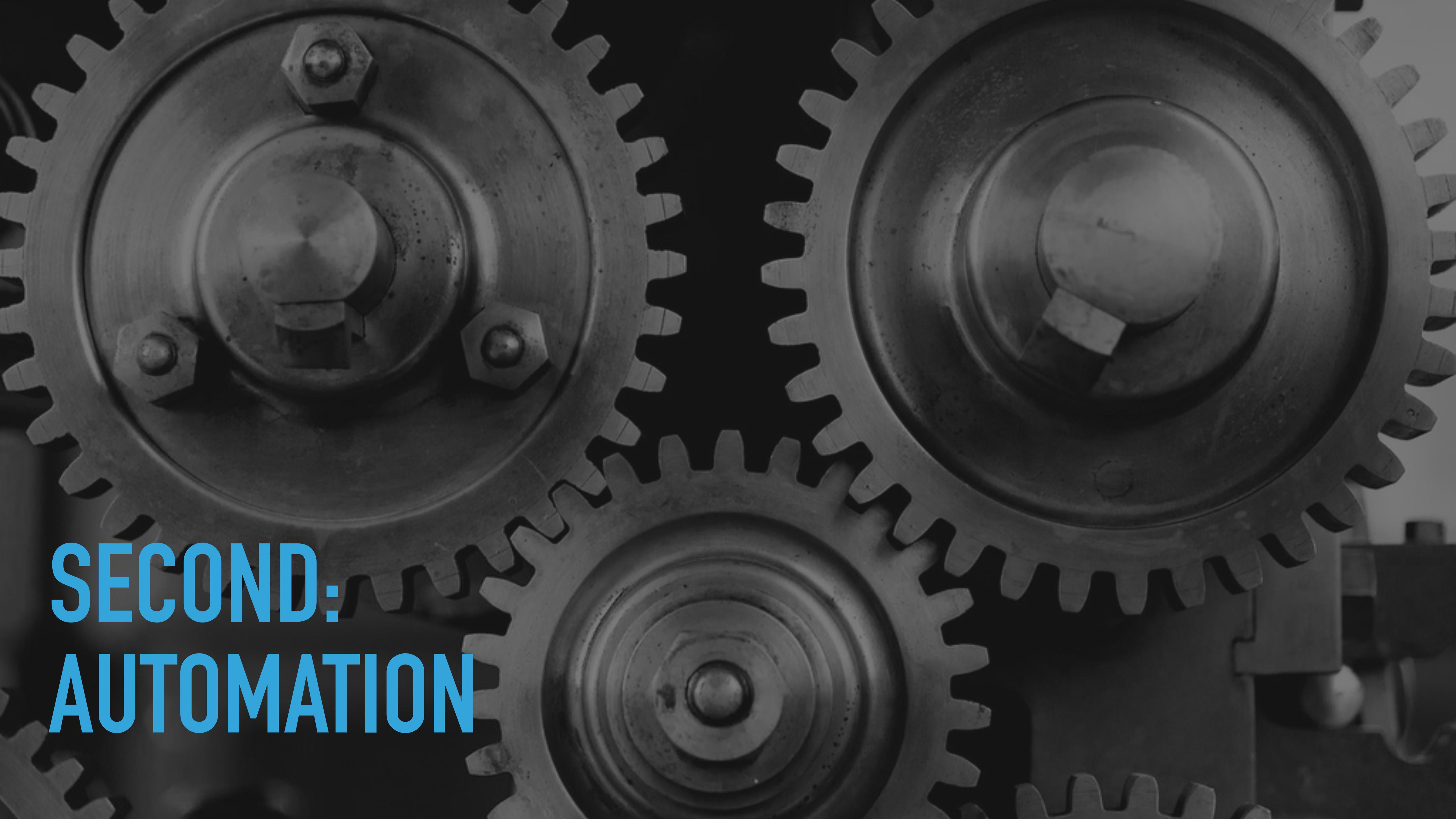


A photograph of a large satellite dish antenna. The dish is a light-colored, parabolic reflector mounted on a dark, cylindrical support structure. It is set against a clear blue sky with some faint clouds. The background shows a complex metal lattice structure of a tower or antenna system.

WHY WOULD YOU USE
REQUEST & RESPONSE?

The background features a large, abstract geometric shape composed of numerous triangles. The left side of the shape is filled with a pattern of red and orange triangles, while the right side is filled with a pattern of blue and teal triangles. The overall effect is a sense of depth and movement.

**FIRST:
INTEROPERABILITY**



SECOND: AUTOMATION

NON-AUTOMATED COMPLEXITIES

1. Research Points (🧠)

- ▶ User figures out how things work

2. Decision Points (💡)

- ▶ User decides what to do

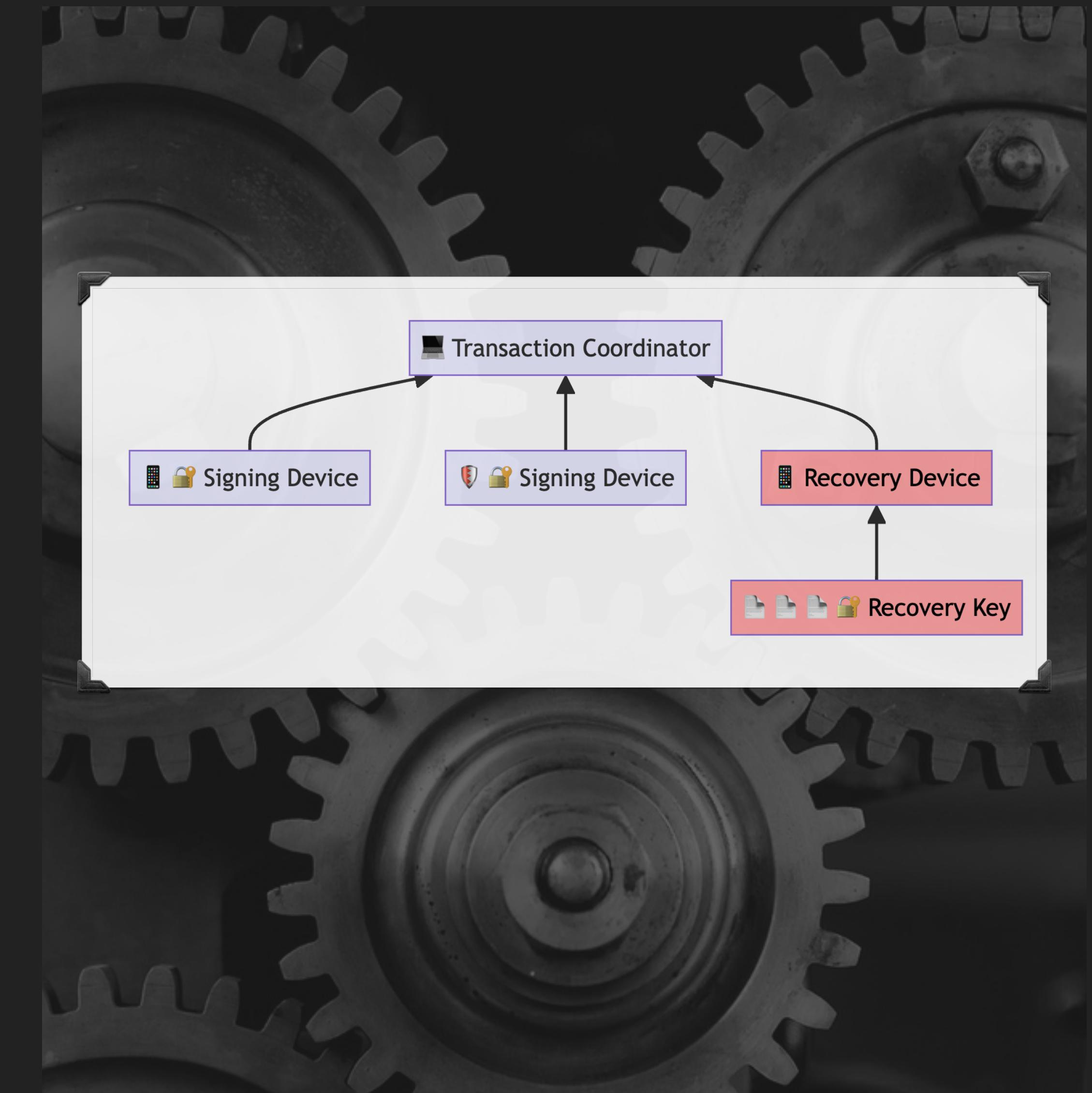
3. Human Actions (👤)

- ▶ User pushes buttons



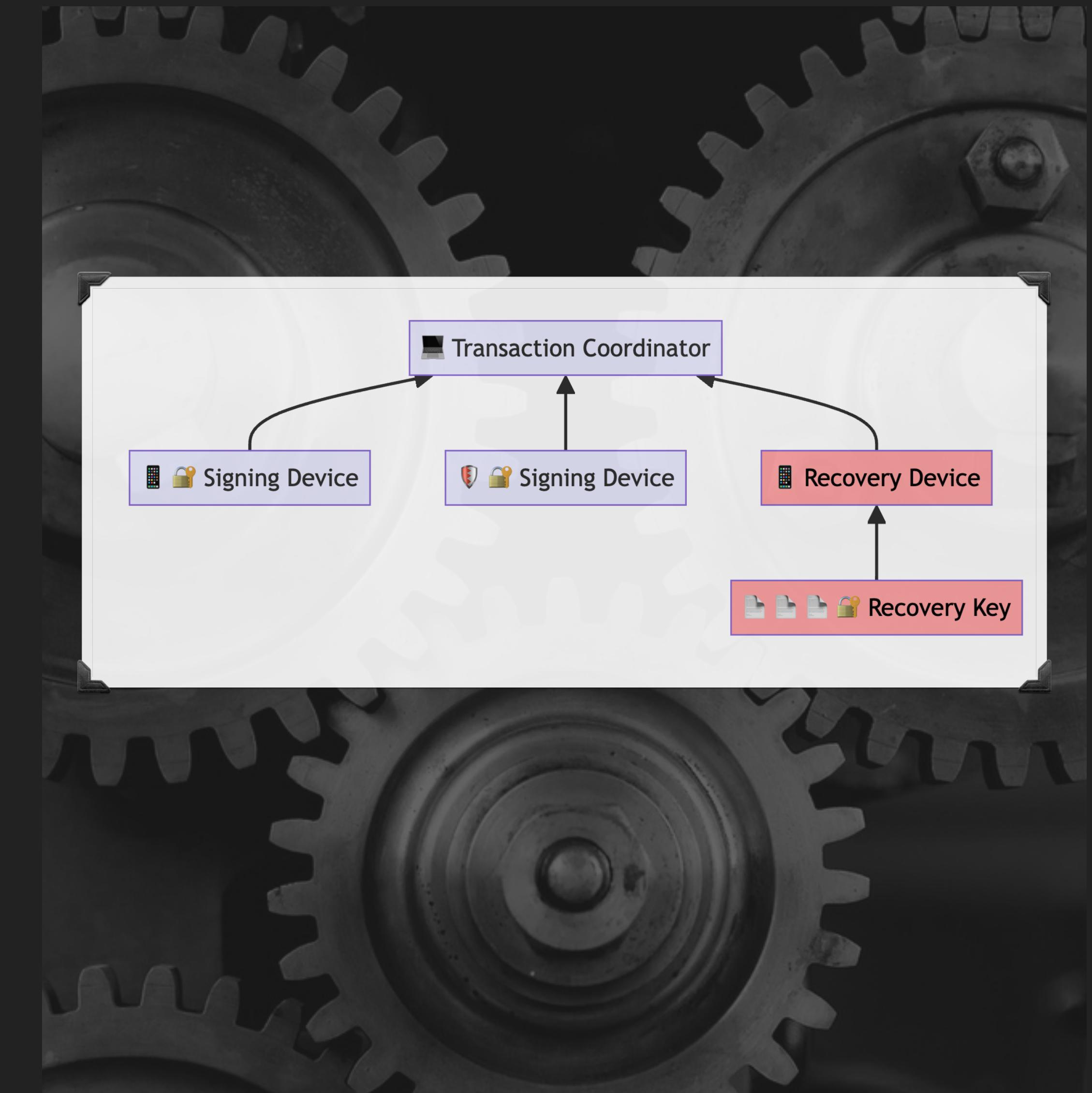
AUTOMATING TASKS

- ▶ Creating Multisigs is a Clear Use Case
- ▶ We Want People to Use Multisigs
 - ▶ More Secure
 - ▶ No Single Points of Failure!
 - ▶ No Single Points of Compromise



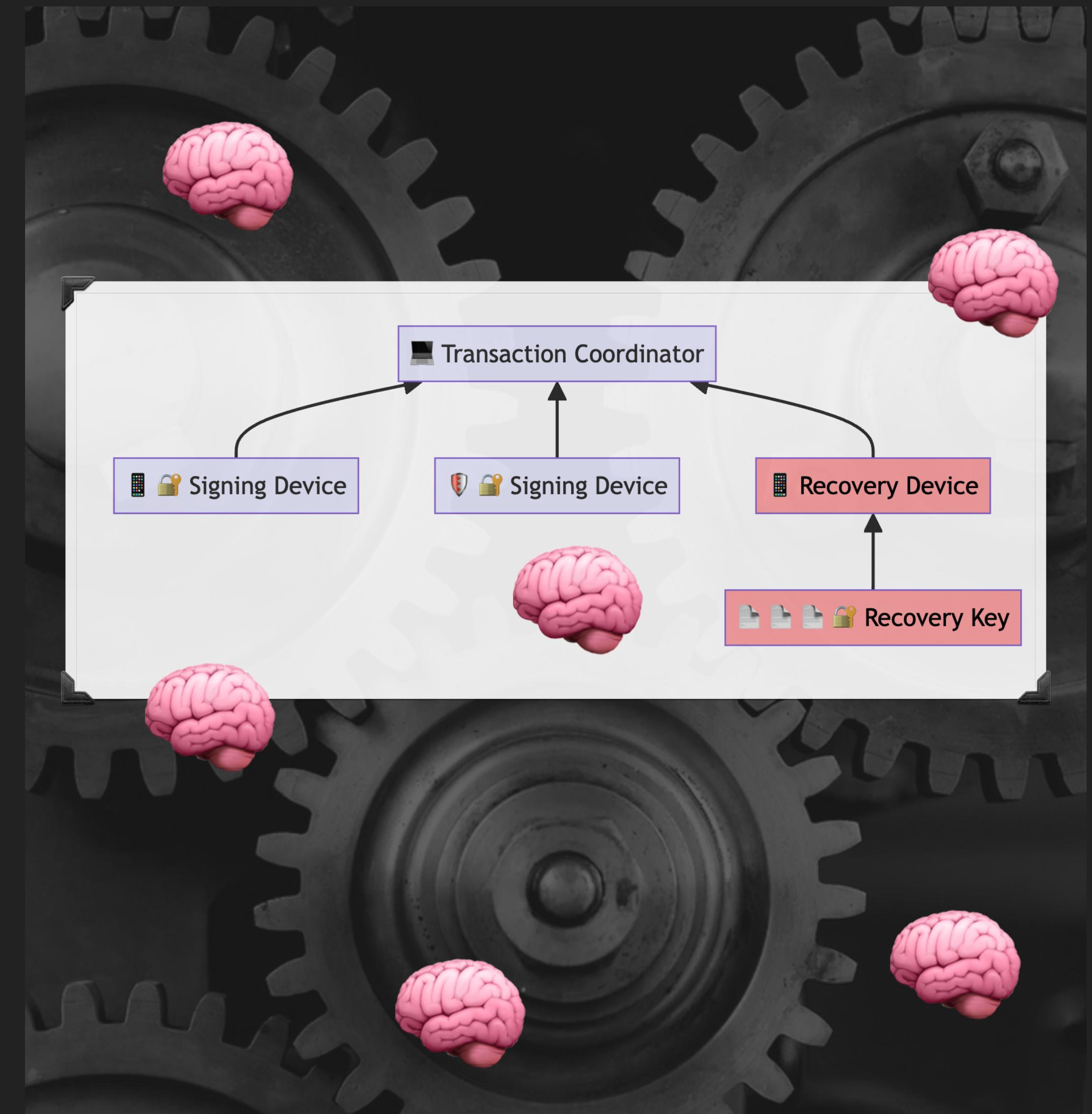
AUTOMATING MULTISIGS

- ▶ We Wrote a Whole Scenario on Creating Multisigs
 - ▶ <https://tinyurl.com/multisigs>
- ▶ But It's Too Complex!



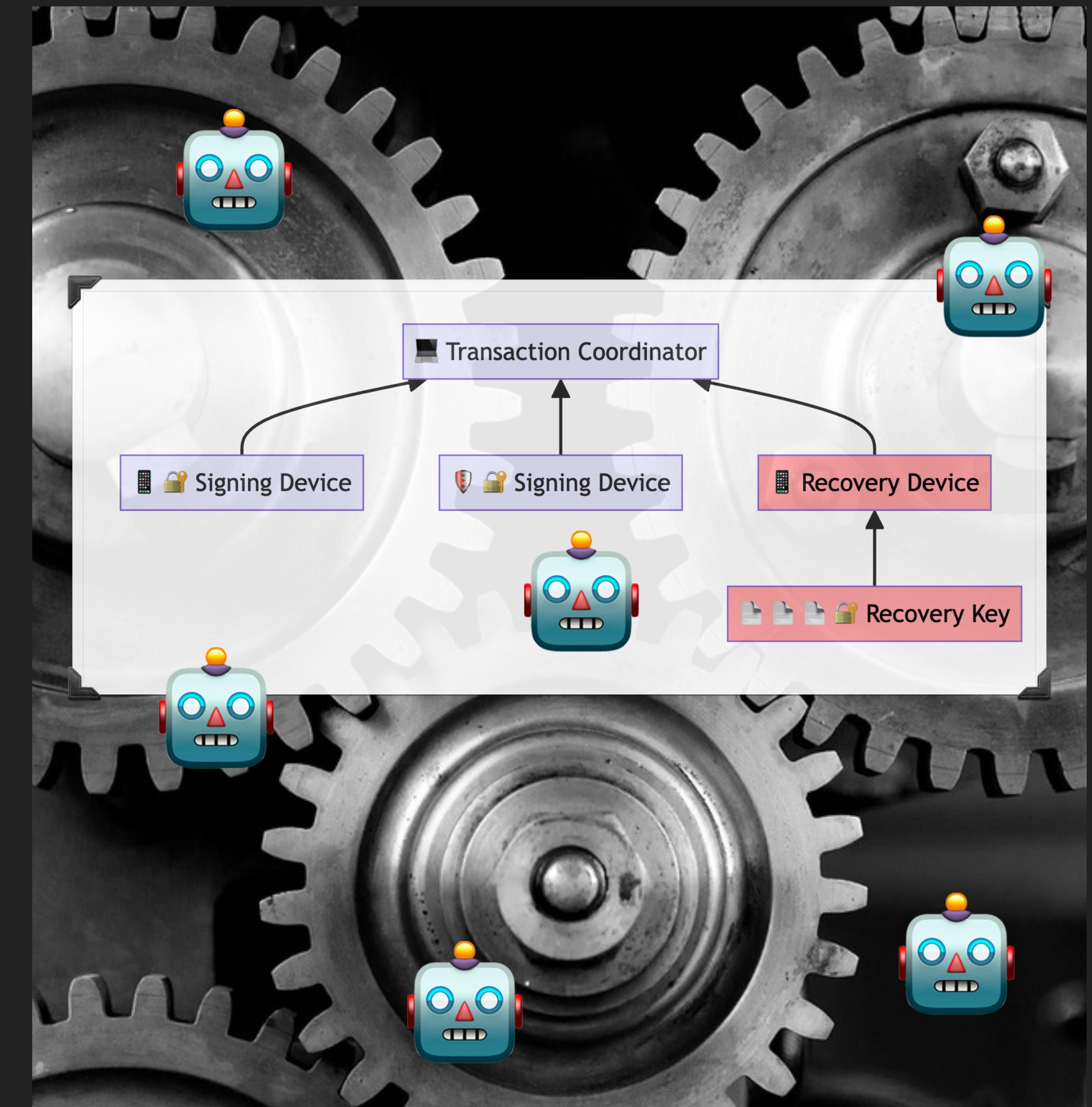
NON-AUTOMATED MULTISIG

- ▶ User Has to **Know** a Procedure
- ▶ User Has to **Know** How Devices Work
- ▶ User Is a Knowledgeable **Conductor**



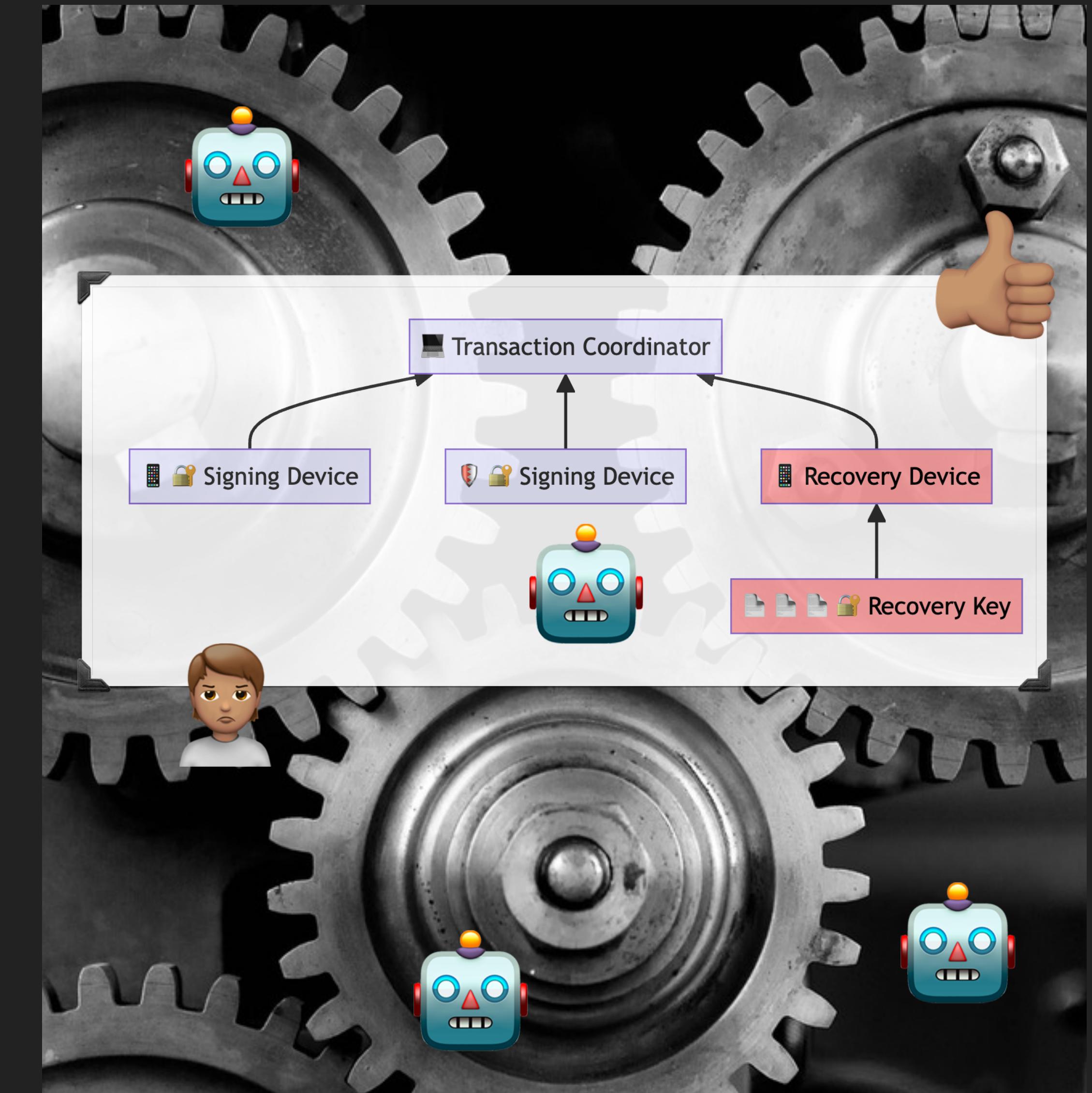
AUTOMATED MULTISIG

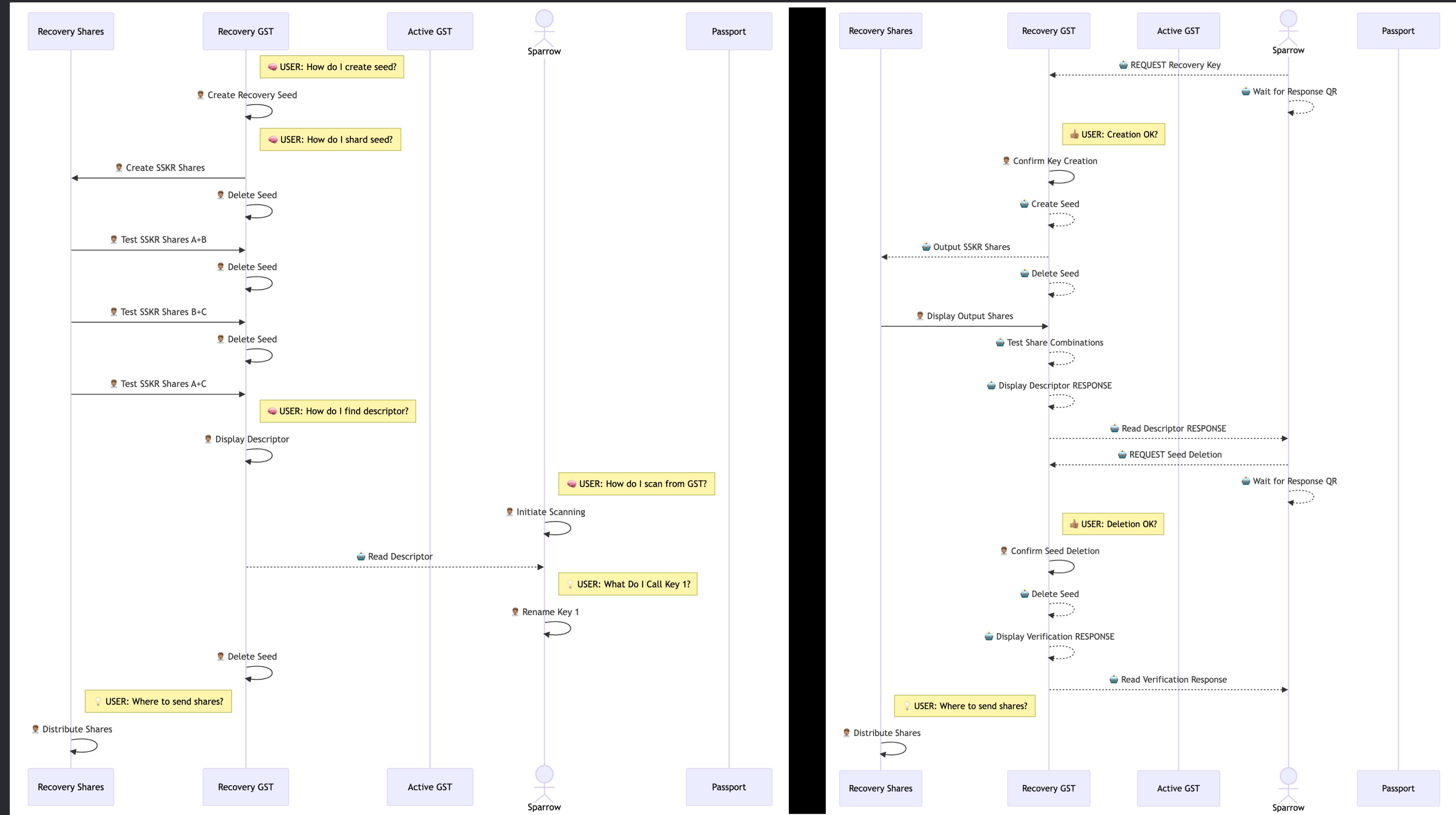
- ▶ Transaction Coordinator is Conductor
- ▶ Centralizes Decisions
- ▶ Tells The User What to Do
- ▶ Communicates with Devices
 - ▶ Rather than the User Jumping Around
 - ▶ Tells the Devices What to Do
 - ▶ Rather than the User Having to Know



AUTOMATED MULTISIG WITH USER

- ▶ Transaction Coordinator is Conductor
- ▶ But ...
- ▶ The User Confirms
- ▶ The User Takes Physical Actions





TASK AUTOMATION

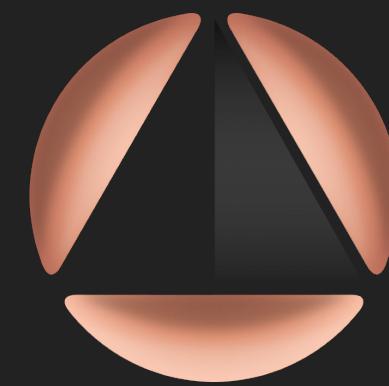
	Classic	R/R
Decision Points (💡)	5	2
Confirmation Points (👍)	0	6
Research Points (🧠)	11	1
Human Actions (👤)	31	14
Automated Actions (🤖)	5	33

TASK SIMPLIFICATION

THE MULTISIG USE CASE FOR ENVELOPE REQUEST & RESPONSE

R/R ECOSYSTEM STARTS TODAY!

- ▶ Foundation Devices is Supporting Envelope & Request/Response for an Upcoming Project



F O U N D A T I O N

- ▶ Join Us in Creating an Automated & Interoperable Ecosystem



FOR MORE ON REQUEST & RESPONSE

- ▶ Take a Look at our Full Use Case:
 - ▶ In our Smart Custody Repo
 - ▶ <https://tinyurl.com/multisigs-rr>
- ▶ Implement with the Implementation Guide:
 - ▶ In our Research Repo
 - ▶ <https://tinyurl.com/multisigs>



