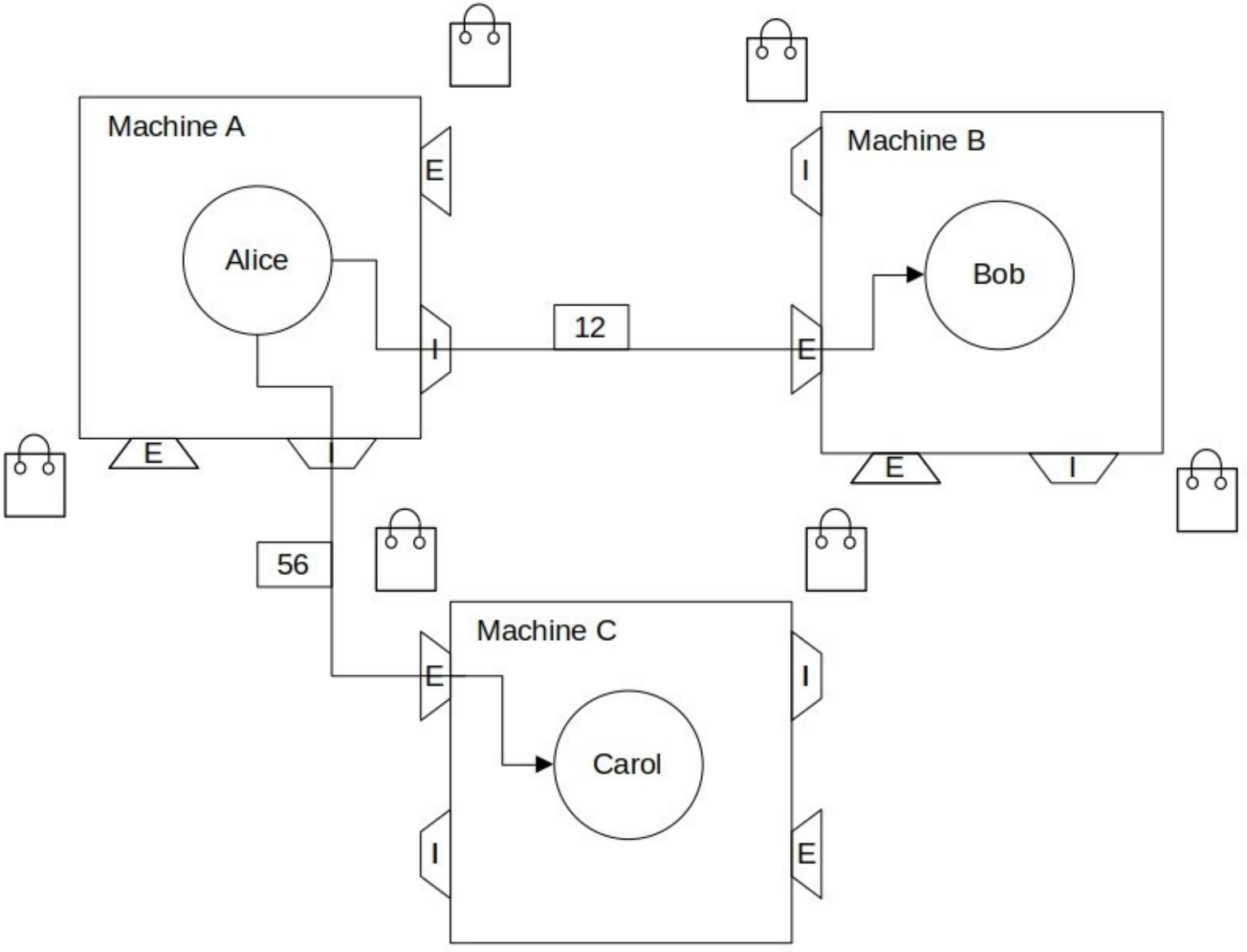

Machine A
Alice
E
I
E
I
12
56
Machine B
Bob
I
E
E
I
Machine C
Carol
E
I
I
E

What do sessions have?

Each side of each session:

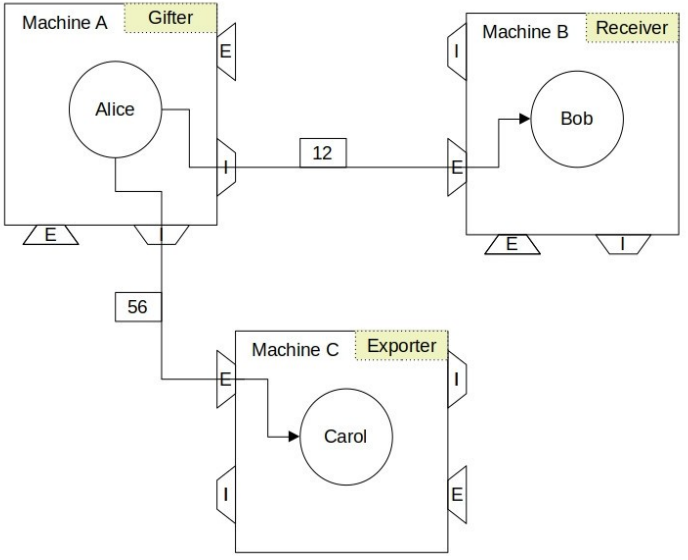
- Session ID (both public keys, sorted and hashed)
- A keypair (public key, private key)
- Import / export table
- A gift table

Alice sending a message to Bob

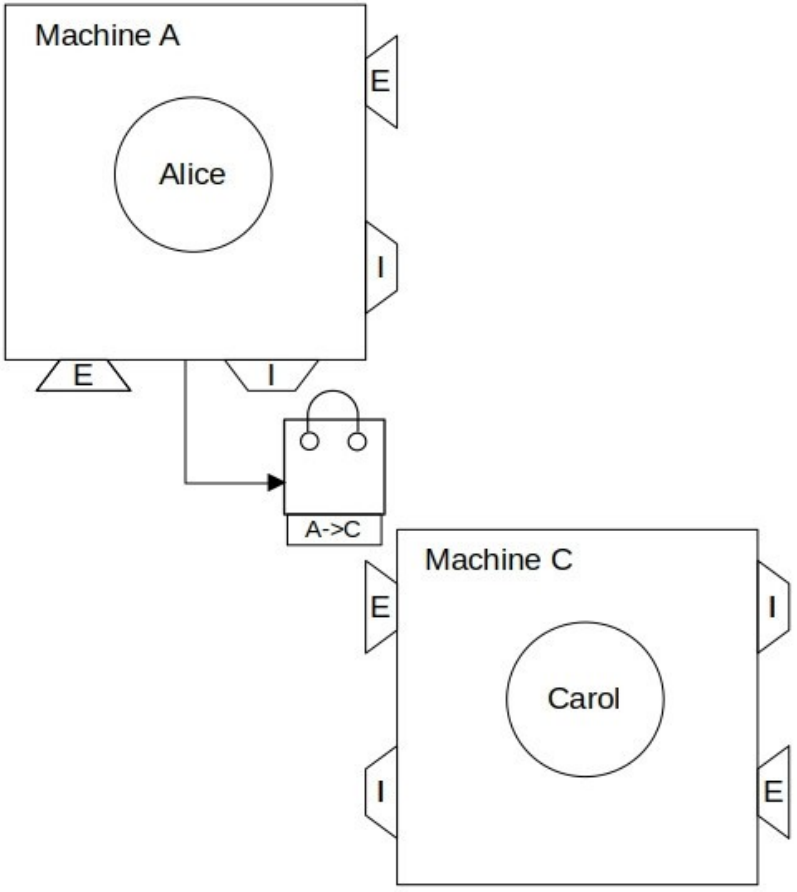
(← bob ’say-hello carol)

Terminology

- Gifter — Who is giving the handoff (A in this case)
- Receiver — Who is getting the reference (B in this case)
- Exporter — Location of gift (C in this case)



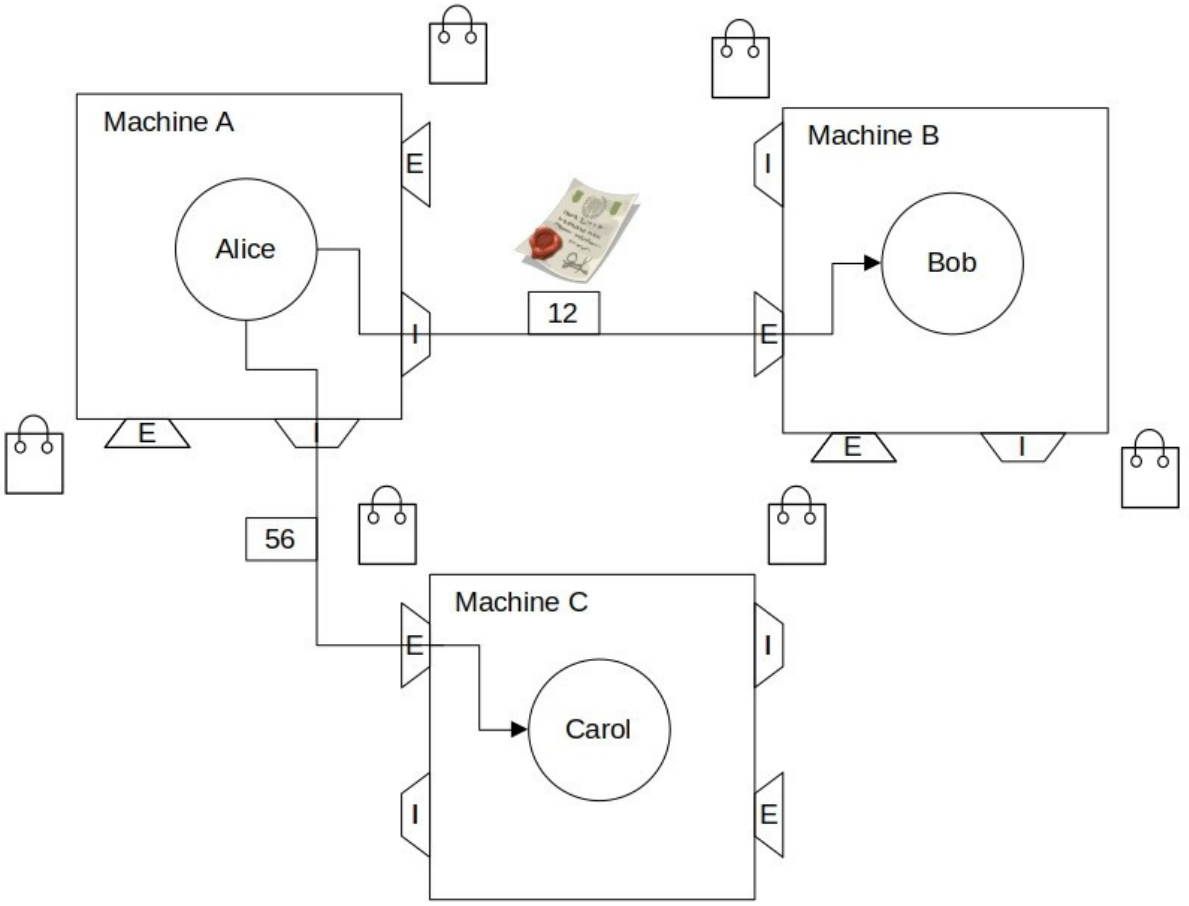
Machine A depositing a gift on C (for B)




```
(← <C-bootstrap-refr> 'deposit-gift 25
<carol-refr>)


(op:deliver-only
     (desc:export 0)
     ('deposit-gift
          25
          (desc:export 56)))
```

Machine A sending certificate to B



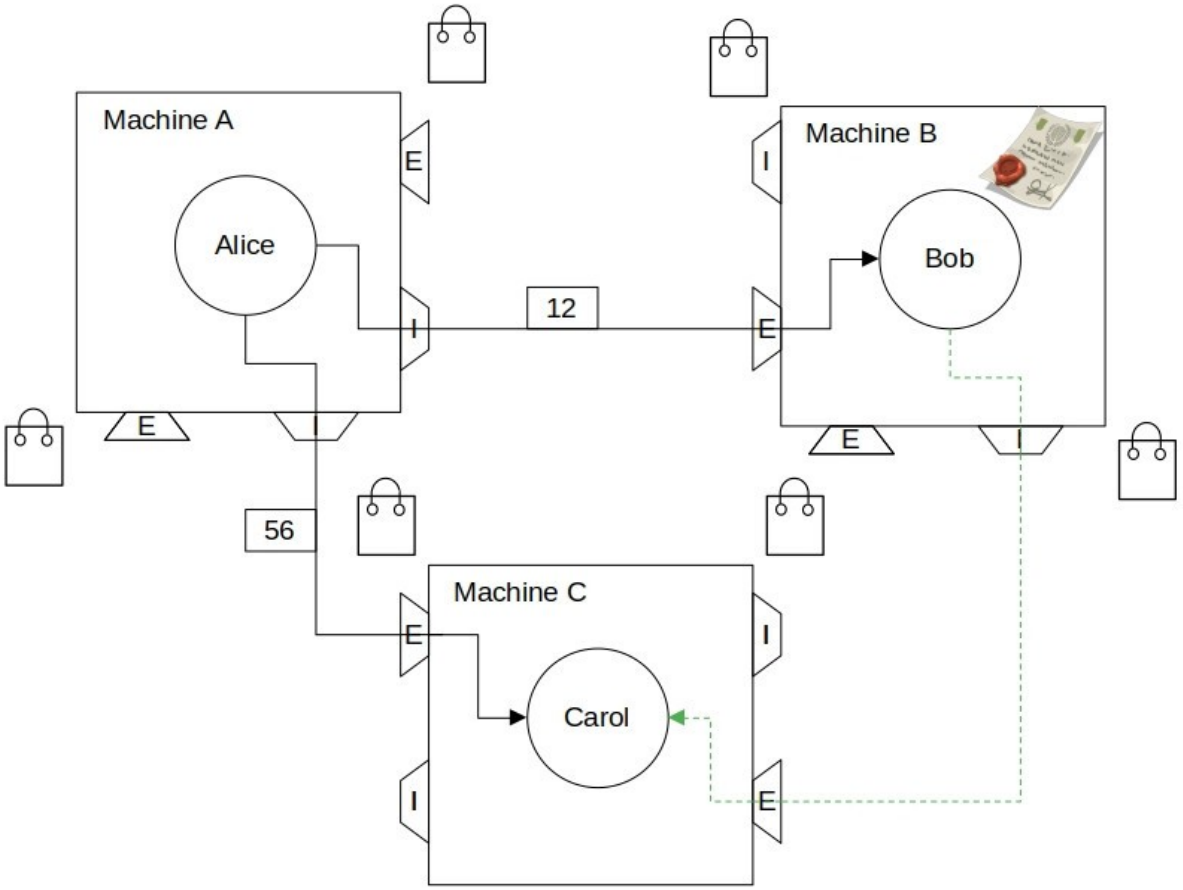
Certificate

`(desc:handoff-give`	Certificate
`recipient-key`	B’s public key in A ↔ B
`exporter-location`	Machine address to C
`session`	Session ID for A ↔ C session
`gifter-side`	Machine A’s public key in A ↔ C session
`gift-id)`	ID of gift (carol) in C’s A ↔ C gift table

Certificate that’s on the wire

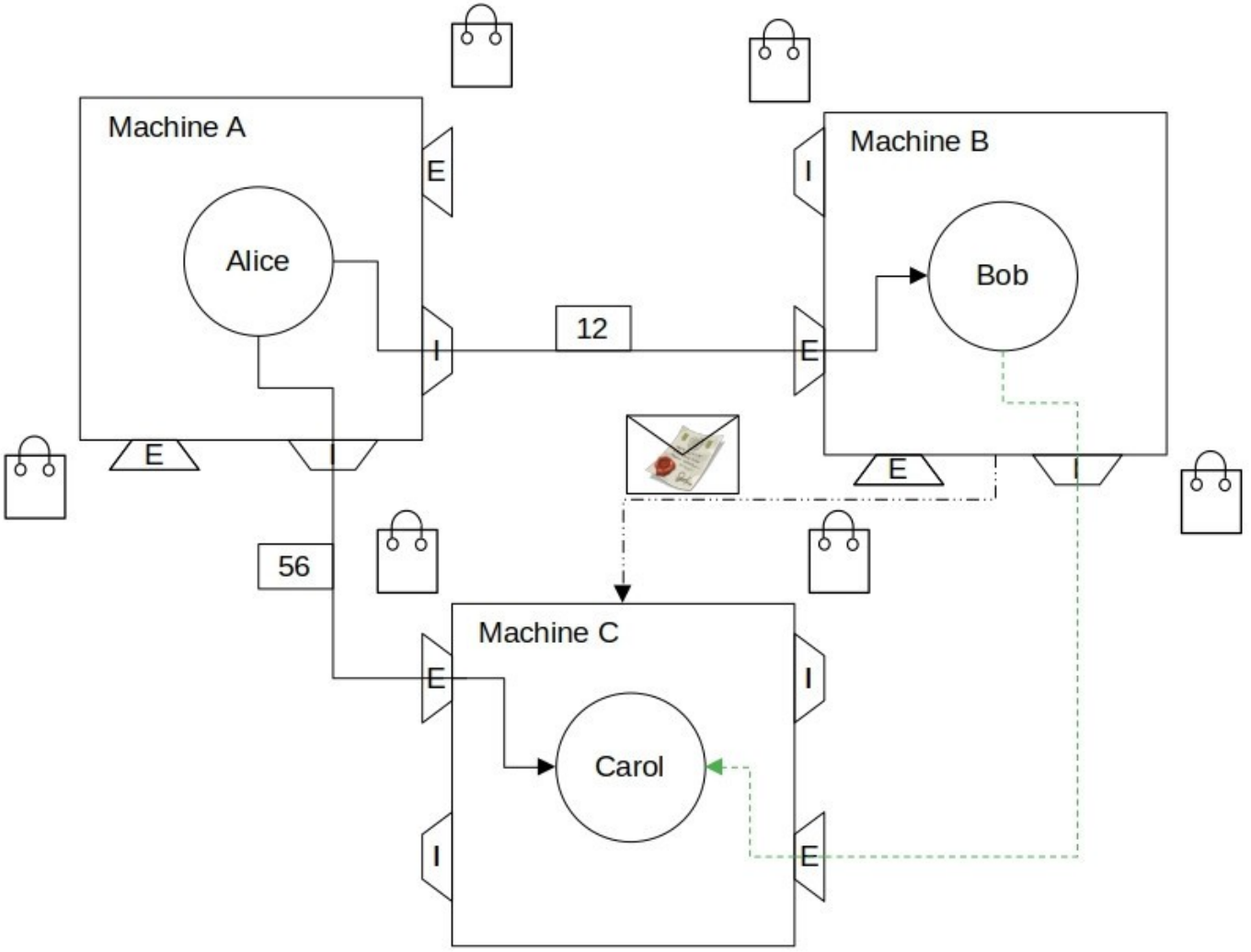
```
(desc:sig-envelope                    Signed envelope wrapping message
  (desc:handoff-give                  Certificate
    <B-key-of-AtoB>
    ”ocapn://machine-c.foo”
    AtoC-session
    A-key-of-Ato-C
    25)                               The ID where A left the gift on C for B
 <signature-by-A-key-of-AtoC>)        The signature of the desc:handoff-give
```

What does Machine B do with the certificate?



What does B do with the certificate?

- Gives Bob a promise which eventually will resolve to Carol
- Establishes connection to machine C (if one doesn’t exist)
- B makes a desc:handoff-receive based on the certificate
- B gives the desc:handoff-receive to machine C’s bootstrap object


Machine A
Alice
E
I
E
I
12
56
Machine B
Bob
I
E
E
I
Machine C
Carol
E
I
I
E

What is a desc:handoff-receive?

```
(desc:handoff-receive
    receiving-session   B ↔ C session
    receiving-side      B's key in B ↔ C session
    handoff-count       Integer to prevent replay attacks
    signed-give)        Certificate a.k.a handoff-give
```

```
(desc:sig-envelope
  (desc:handoff-receive
    <BtoC-session>                    receiving-session: Name of session in B ↔ C
    <B-key-of-BtoC>                   receiving-side: The B's public key in B ↔ C
    4                                 handoff-count: Integer to prevent replay attacks
    (desc:sig-envelope                signed-give: Same certificate & signed envelope
       (desc:handoff-give
         <B-key-of-AtoB>              recipient-key: B's public key in A ↔ B
         "ocapn://machine-c.foo"      exporter-location: Machine address to C
         AtoC-session                 session: Session ID for A ↔ C session
         A-key-of-AtoC                gifter-side: A's public key in A ↔ C
         25)                          gift-id: key in A ↔ C's gift table
       <signature-by-A-key-of-AtoC>))  Signature A made using their key in A ↔ C
   <signature-by-B-key-of-AtoB>)       Signature B has made using their A ↔ B key
```

Identify which session

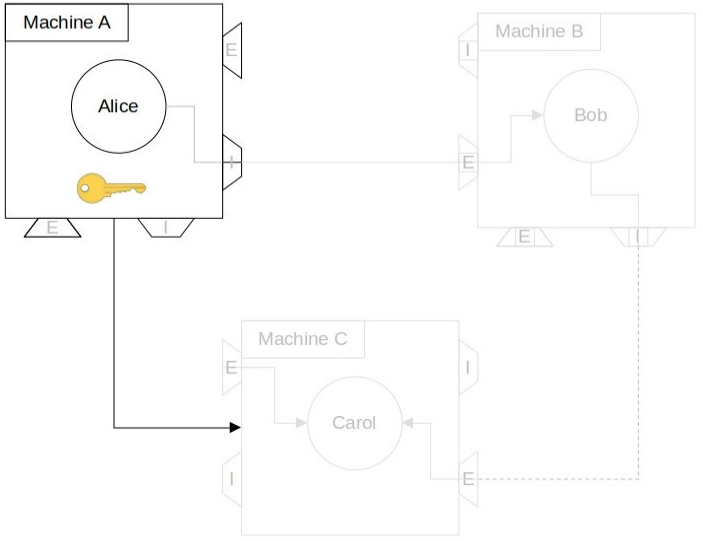
```
(sig-envelope
  (handoff-receive
    <BtoC-session>
    <B-key-of-BtoC>
    4
    (sig-envelope
       (handoff-give
         <B-key-of-AtoB>
         "ocapn://machine-c.foo"
         AtoC-session
         A-key-of-AtoC
         25)
       <signature-by-A-key-of-AtoC>))
   <signature-by-B-key-of-AtoB>)
```

Check the signature on the cert

```
(sig-envelope
  (handoff-receive
    <BtoC-session>
    <B-key-of-BtoC>
    4
    (sig-envelope
       (handoff-give
         <B-key-of-AtoB>
         "ocapn://machine-c.foo"
         AtoC-session
         A-key-of-AtoC
         25)
       <signature-by-A-key-of-AtoC>))
   <signature-by-B-key-of-AtoB>)
```

This is signed using the public key of A in the A ↔ C session.

This proves that A is the A, c thinks they are and that they did in fact make this certificate.

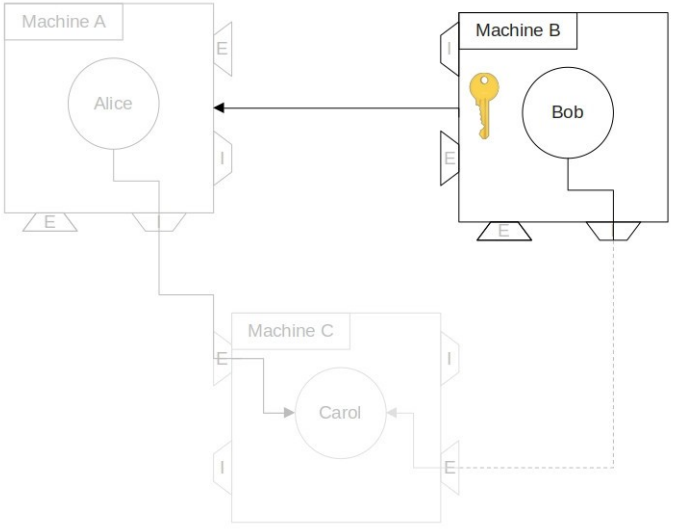


Check the signature on the handoff-receive

```
(sig-envelope
  (handoff-receive
    <BtoC-session>
    <B-key-of-BtoC>
    4
    (sig-envelope
       (handoff-give
         <B-key-of-AtoB>
         ”ocapn://machine-c.foo”
         AtoC-session
         A-key-of-AtoC
         25)
       <signature-by-A-key-of-AtoC>))
   <signature-by-B-key-of-AtoB>)
```

Check the signature on the handoff-receive using the key provided by A in the handoff-give.

This proves to C that that B is the B that machine A thinks B is and that B really did make this handoff-receive

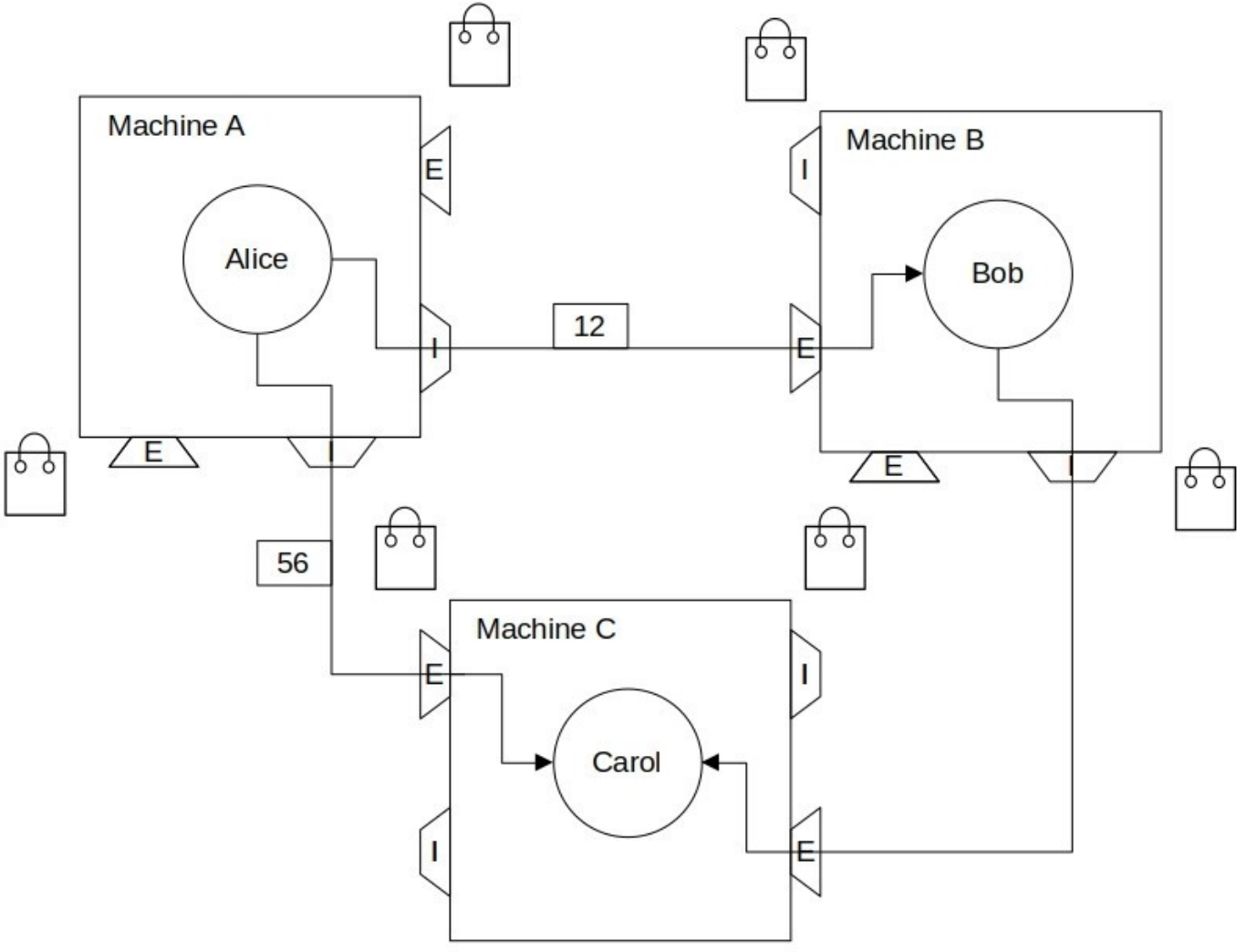


What does C do with the certificate

- After C checks that the certificate is valid, C knows:
  - A in the A ↔ C relationship actually did give B in the A ↔ B relationship the certificate
  - B, who is presenting this to C is the same B that A thinks B is
  - This isn’t being replayed

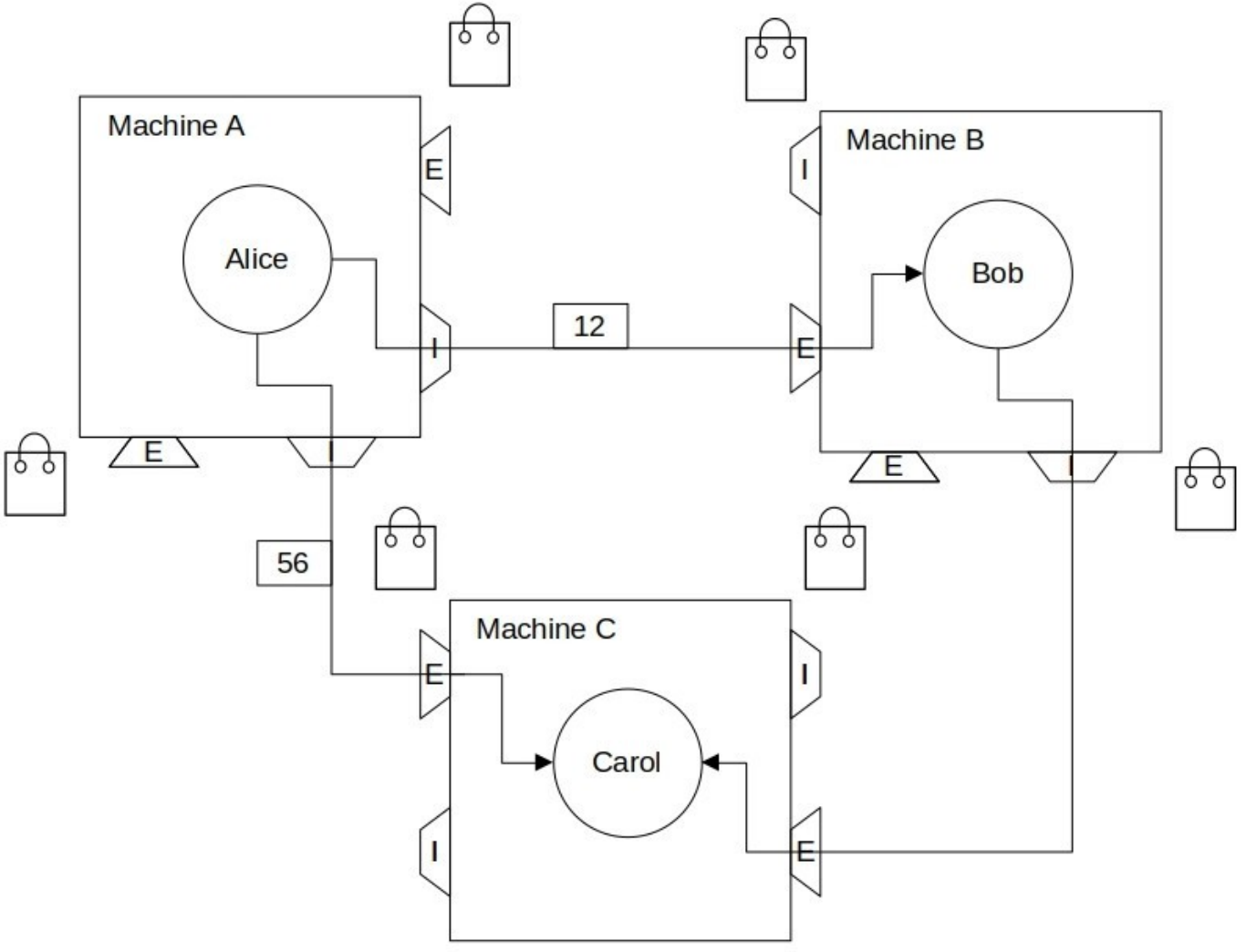
What does C do with the certificate

- C gets the gift at `gift-id` left for B in the A ↔ C gift table
- It exports this gift to B


Machine A
Alice
E
I
E
I
12
Machine B
I
Bob
E
E
I
56
Machine C
E
I
Carol
I
E

B gets the reference to Carol from C

- B fulfills the promise B gave to Bob.


Machine A
Alice
E
I
E
I
12
Machine B
I
Bob
E
E
I
56
Machine C
E
I
Carol
I
E