

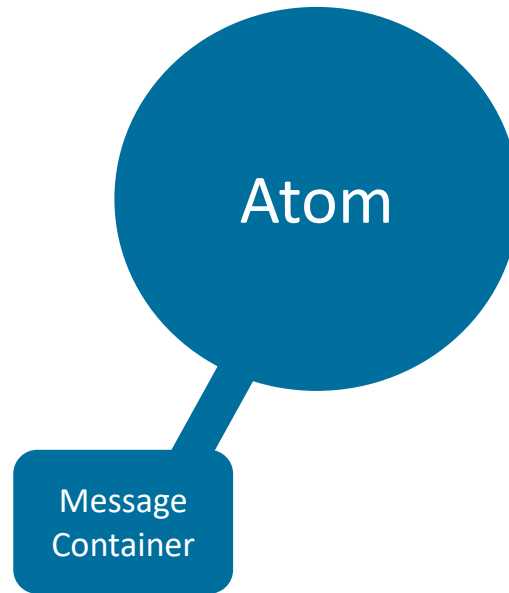


Research Studio SAT

# WoN Data Model & Architecture



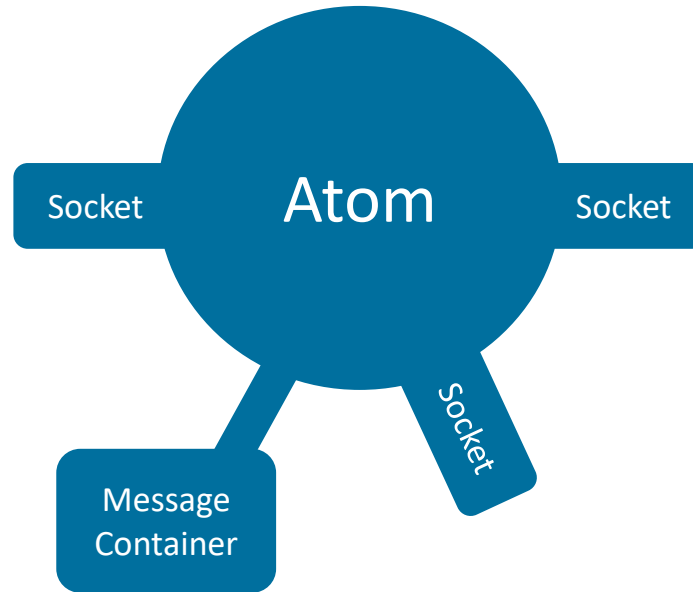
# Atom



# Atom

- Public Key
- Self-Description (what am I, what is the offer, etc.)
- Description of desired counterparts (what am I looking for?)
- State: Active/Inactive/Deleted
- Matcher Control
  - Embedded SPARQL query
  - DoNotMatchBefore/DoNotMatchAfter [Date]
  - Flag: NoHintForMe/NoHintForCounterpart
  - Flag: UsedForTesting

# Socket



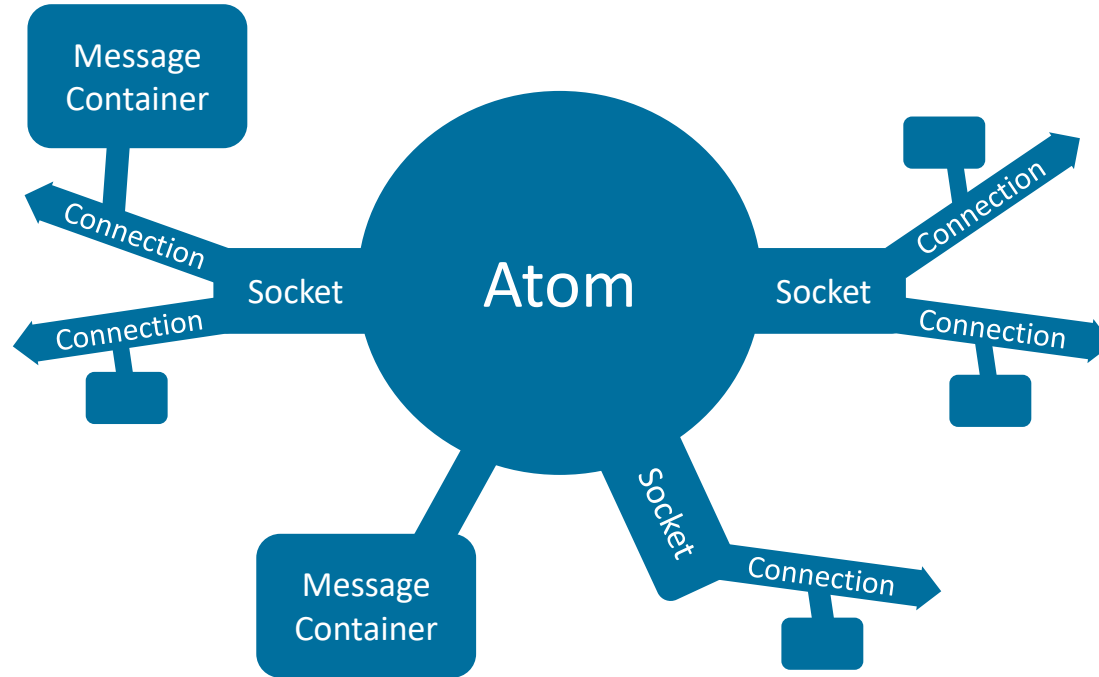
# Socket

Part of the atom, allows connecting to other atoms

Refers to a published SocketDefinition, interpreted by WoN node

- Capacity (number of connections)
- Auto-open
- Generate atom property when connected
- Compatible socket

# Connection



# Connection

- Belongs to the atom
- Has a message container (private)
- Points to *target atom* and *target connection*
- Is established in a connect/connect handshake

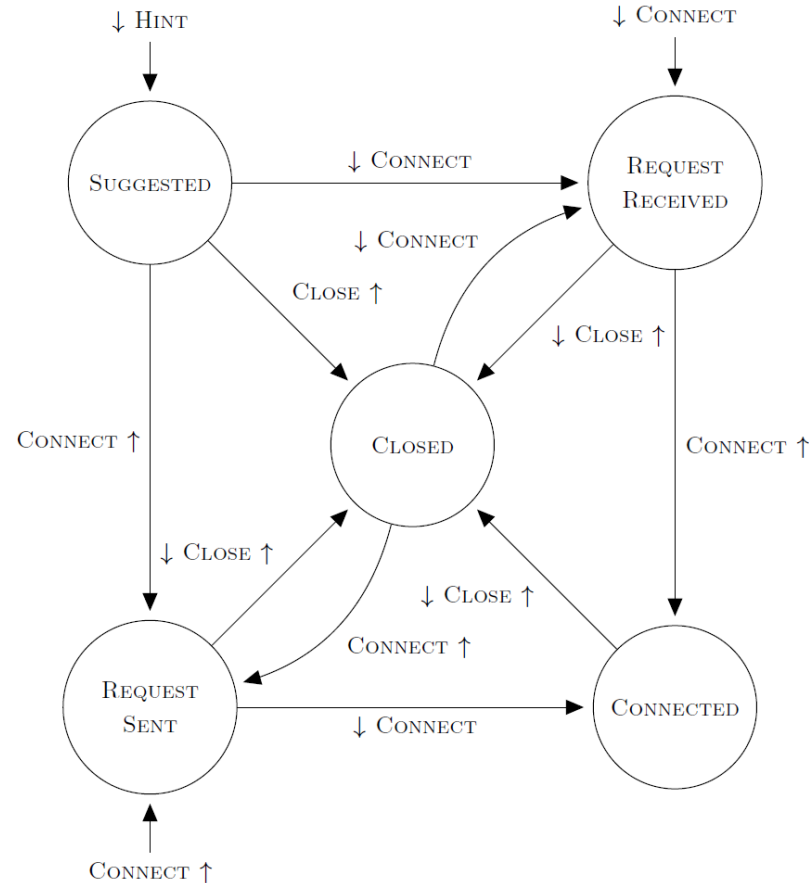
# Connection State

○ connection state

↓[type] incoming message of [type]

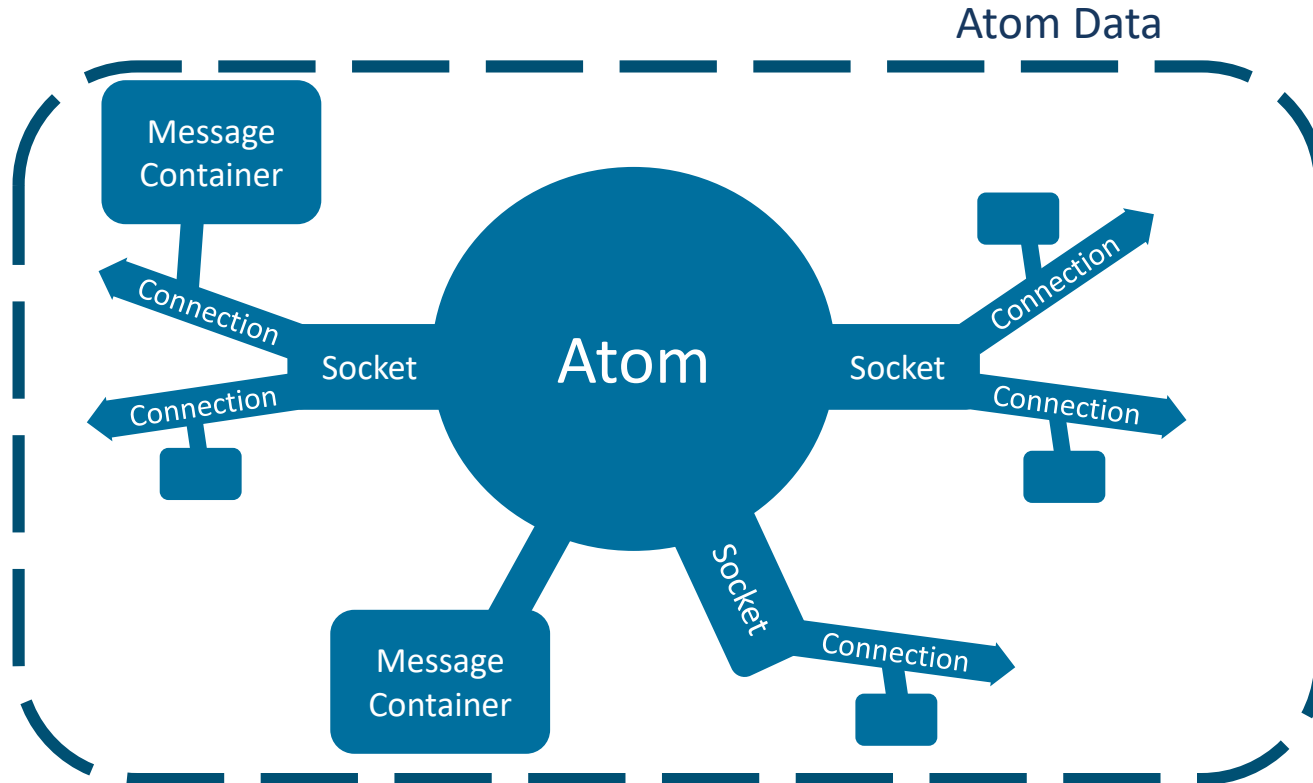
[type]↑ outgoing message of [type]

↓[type]↑ in or out message of [type]



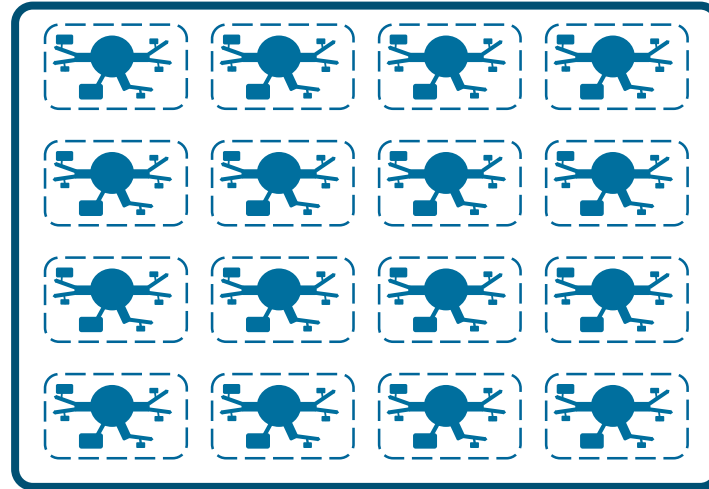


# Atom Data



# WoN Node

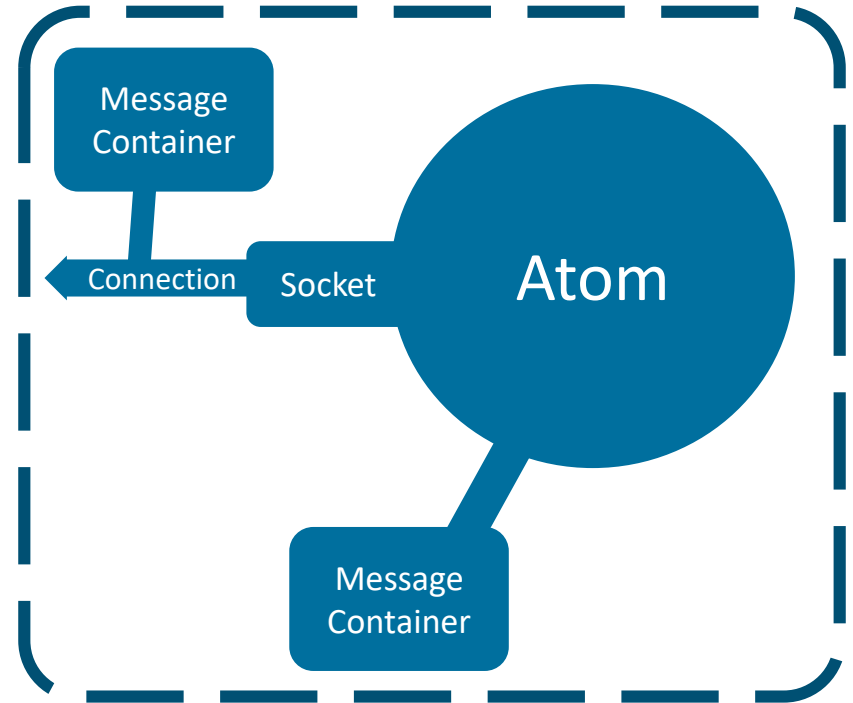
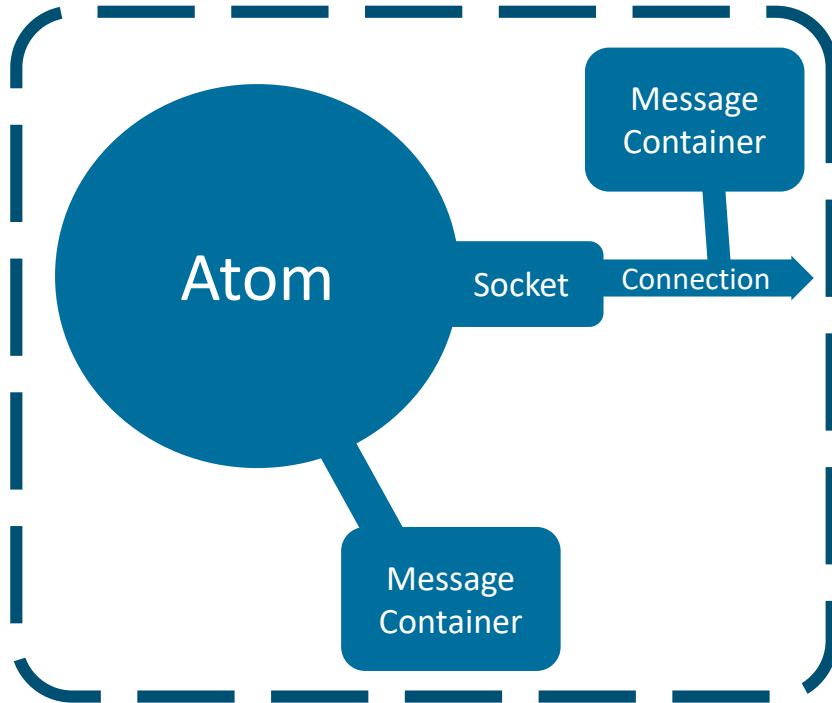
## WoN Node



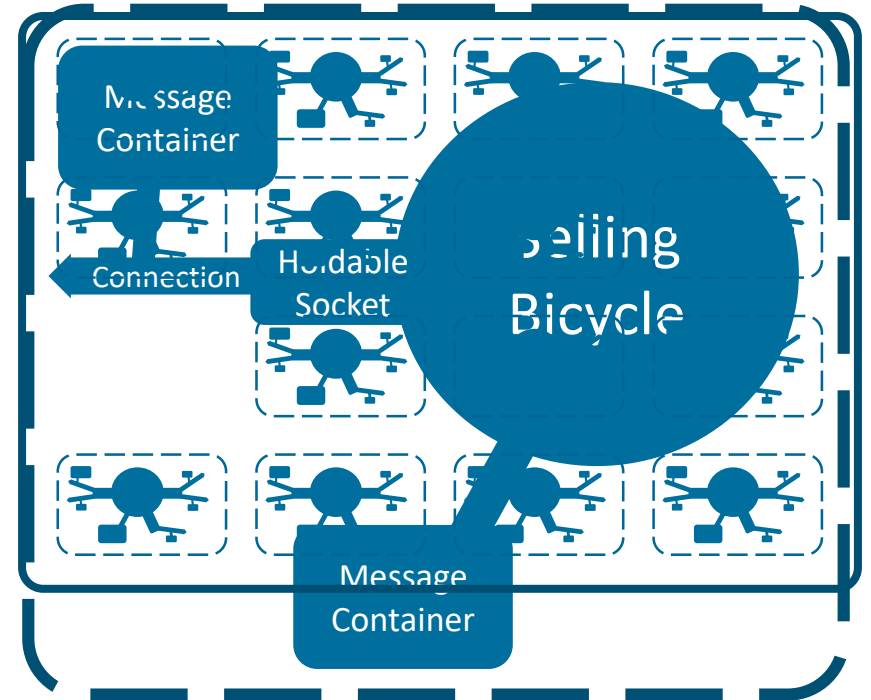
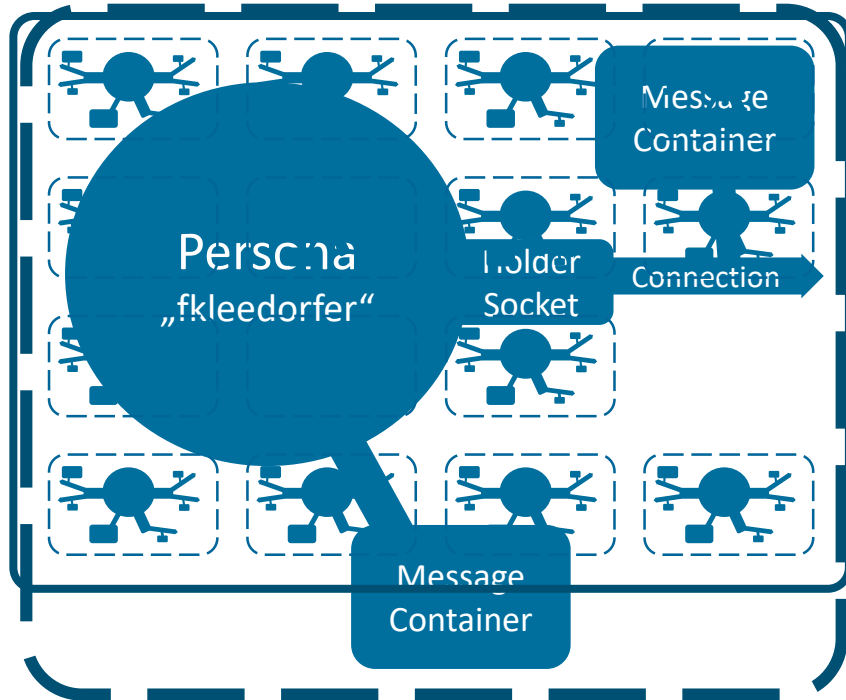
# WoN Node

- Public key
- Holds all data of N atoms
- Crawlable
- Processes messages from/to Owner Applications
- Publish-Subscribe queue for updates (→ matchers)

# Composition of Atoms



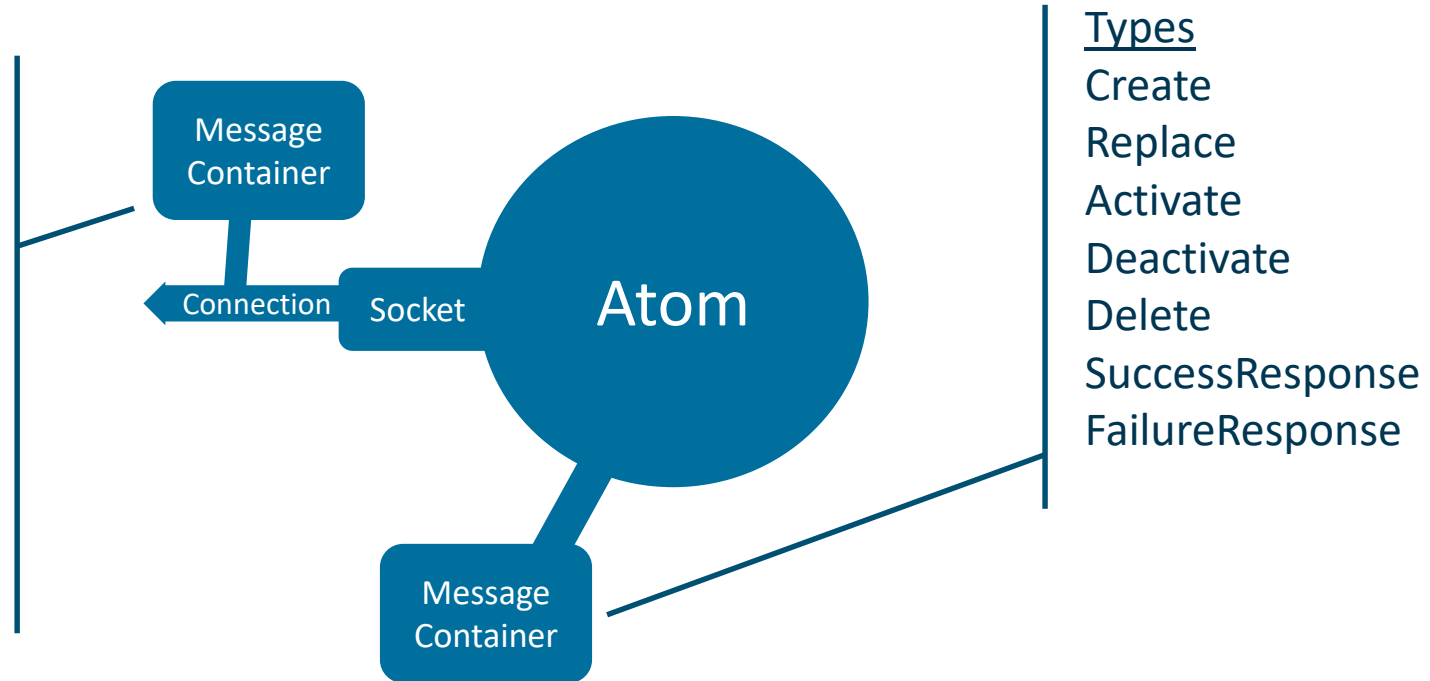
# Example



# Message

## Types

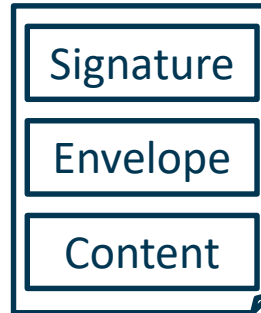
Hint  
Connect  
ConnectionMessage  
Close  
SuccessResponse  
FailureResponse



## Types

Create  
Replace  
Activate  
Deactivate  
Delete  
SuccessResponse  
FailureResponse

# Message: Overall Structure



Identifier: content-addressed URI: **wm:/[hash]**

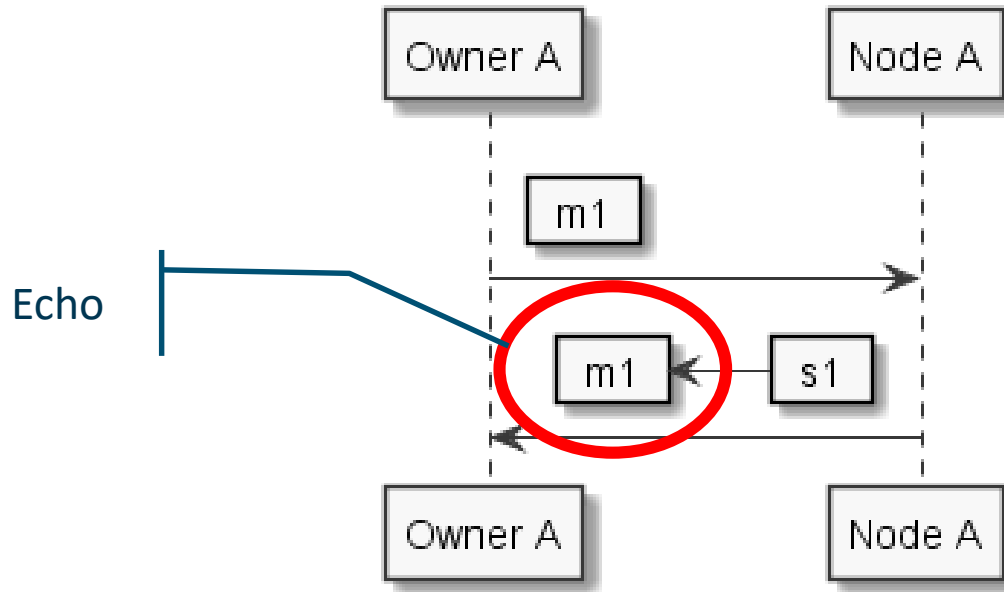
For example: **wm:/W1eXBGNeBvmvcvpdffjb5YjypjRLp2Ti2itfUWsdN1JDs7**

# Delivery Chain: Atom

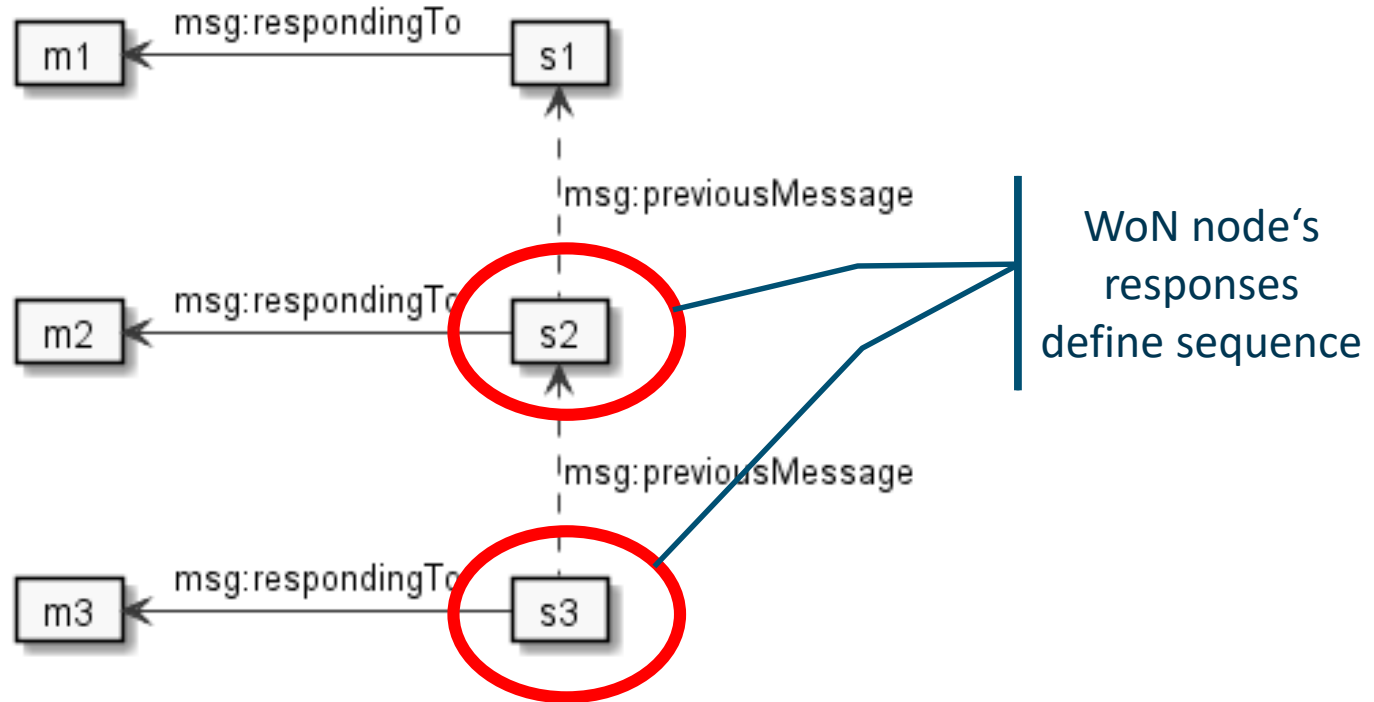




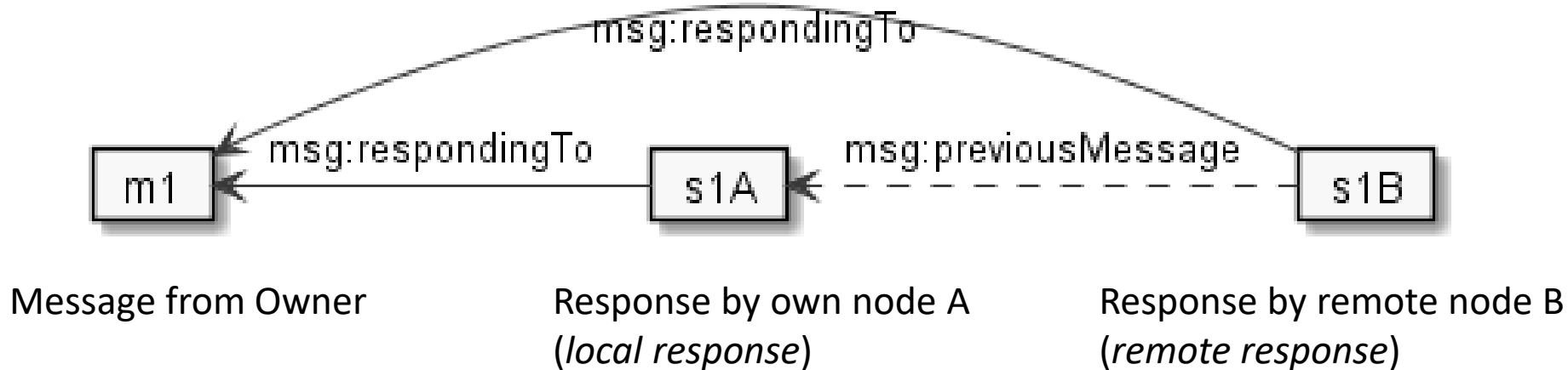
# Delivery Chain: Atom



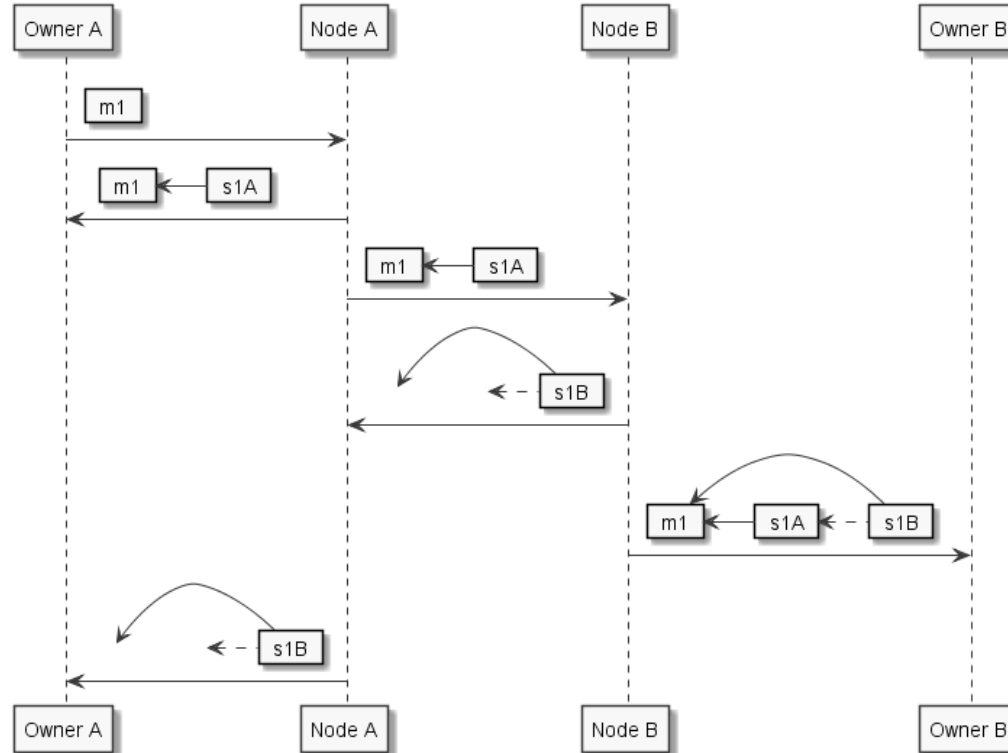
# Message Sequence



# Delivery Chain: Connection

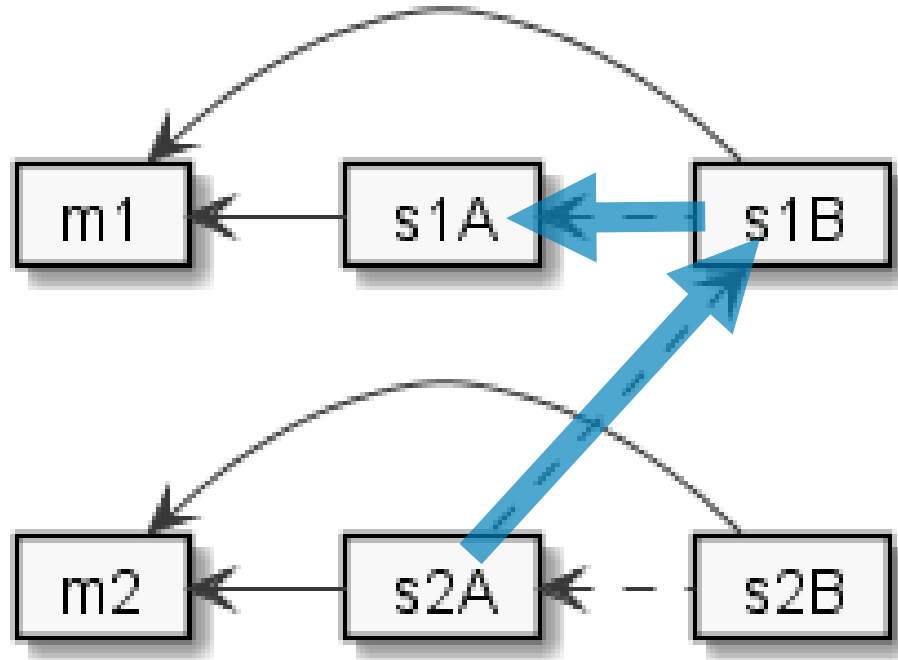


# Delivery Chain: Connection



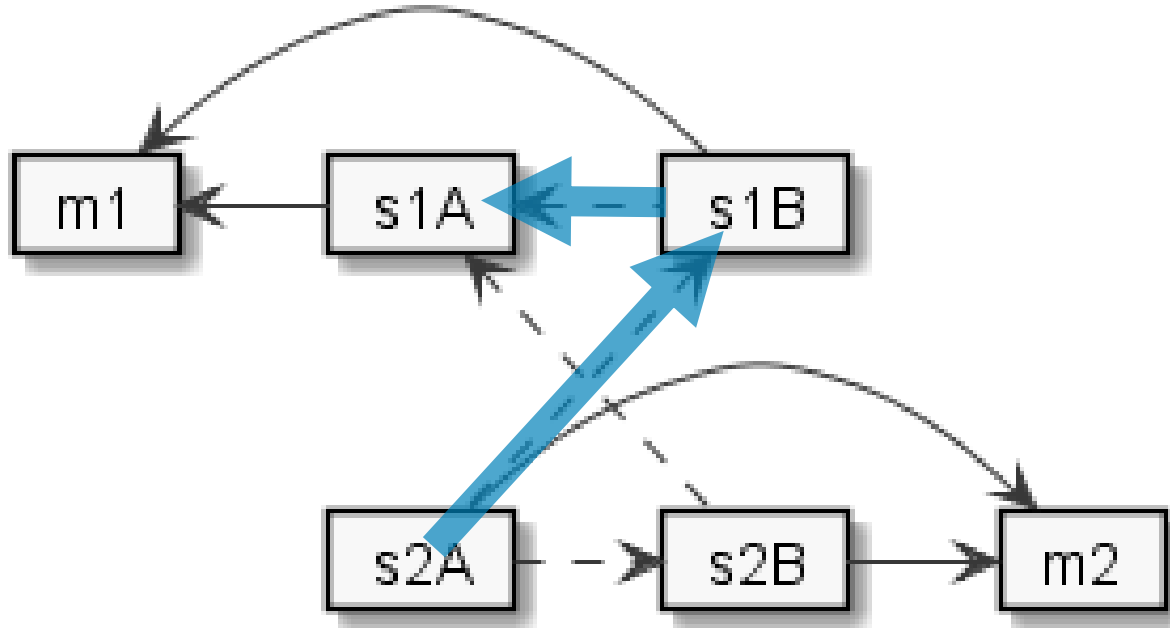
Goal:  
Same verifiable  
message sequence  
in both connections

# Message Sequence: messages from same side

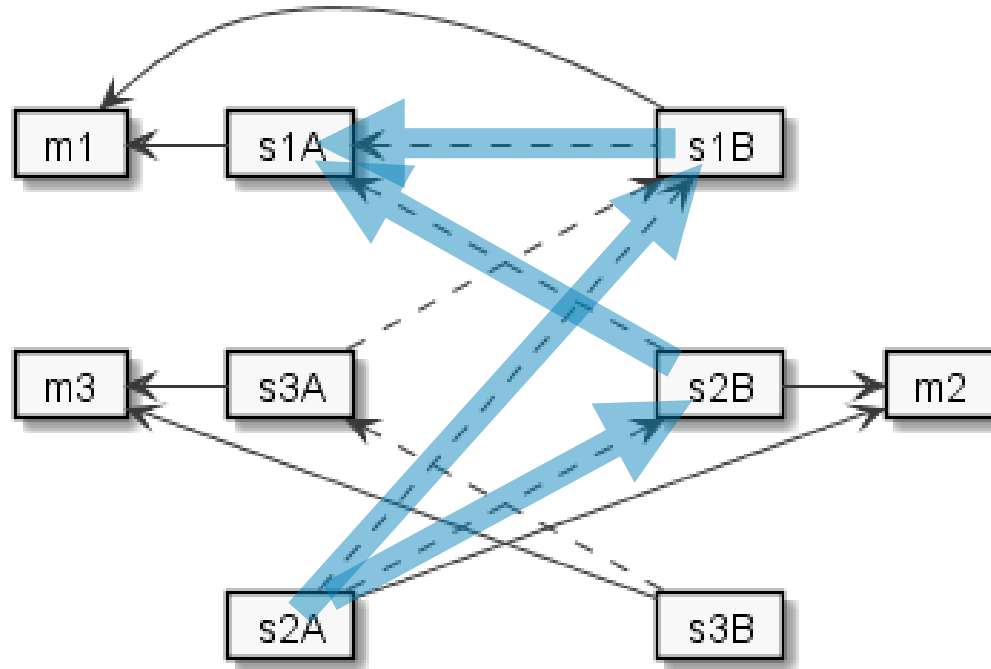


m1 after m2 if Path:  
any response of m1 →  
both responses of m2

# Message Sequence: message from each side



# Message Sequence: simultaneous messages



# Message Sequence: Lost Response

