SOAP

What Is SOAP?

- SOAP 1.1

  Simple Object Access Protocol
- SOAP 1.2

  Huh? Us, work for Microsoft? No way!

  XML Protocol (XMLP or XP)

SOAP 1.2 Is…

- A “wrapper” protocol
- Written in XML
- Independent of the wrapped data
- Independent of the transport protocol
- Efficient (according to the W3C)
- A uni-directional message exchange paradigm

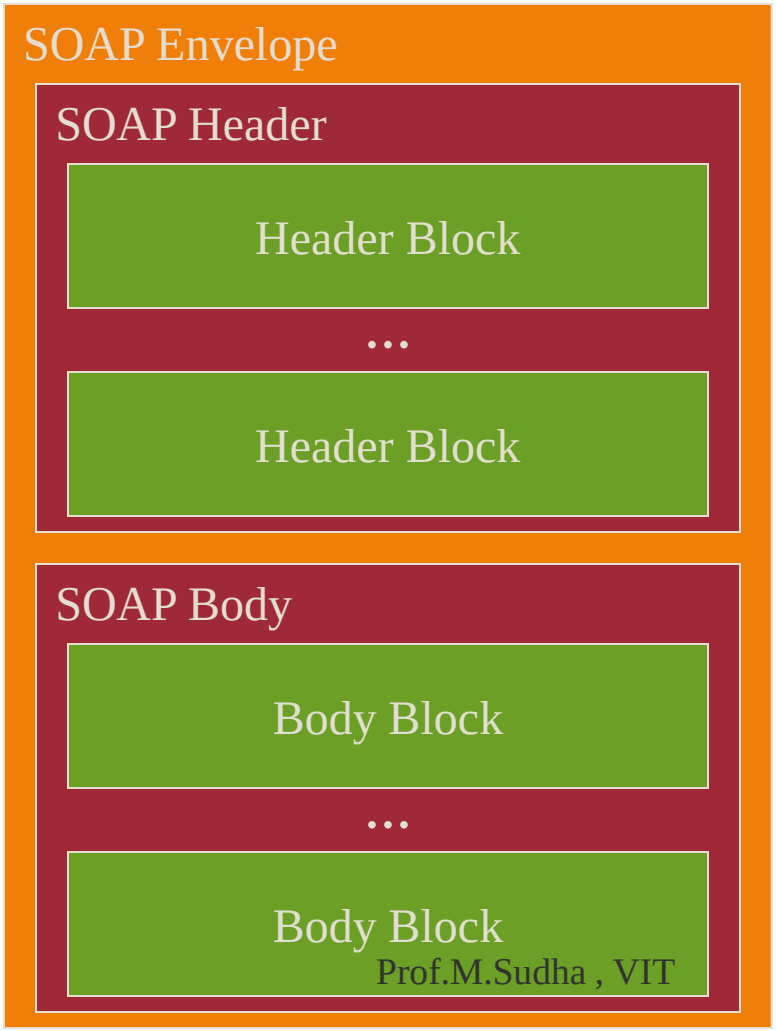
SOAP 1.2 Is *Not*…

- A transport protocol
- Written in *valid* XML
- Independent of the wrapped data
- Independent of the transport protocol
- Efficient (according to me)
- A uni-directional message exchange paradigm

Classification

- “Wrapper” protocol versus transport protocol
- Data is placed in header blocks and body blocks, as desired
- Transport is handled by another mechanism
  - HTTP 1.1 is the only binding specified, though others are possible

Message Anatomy



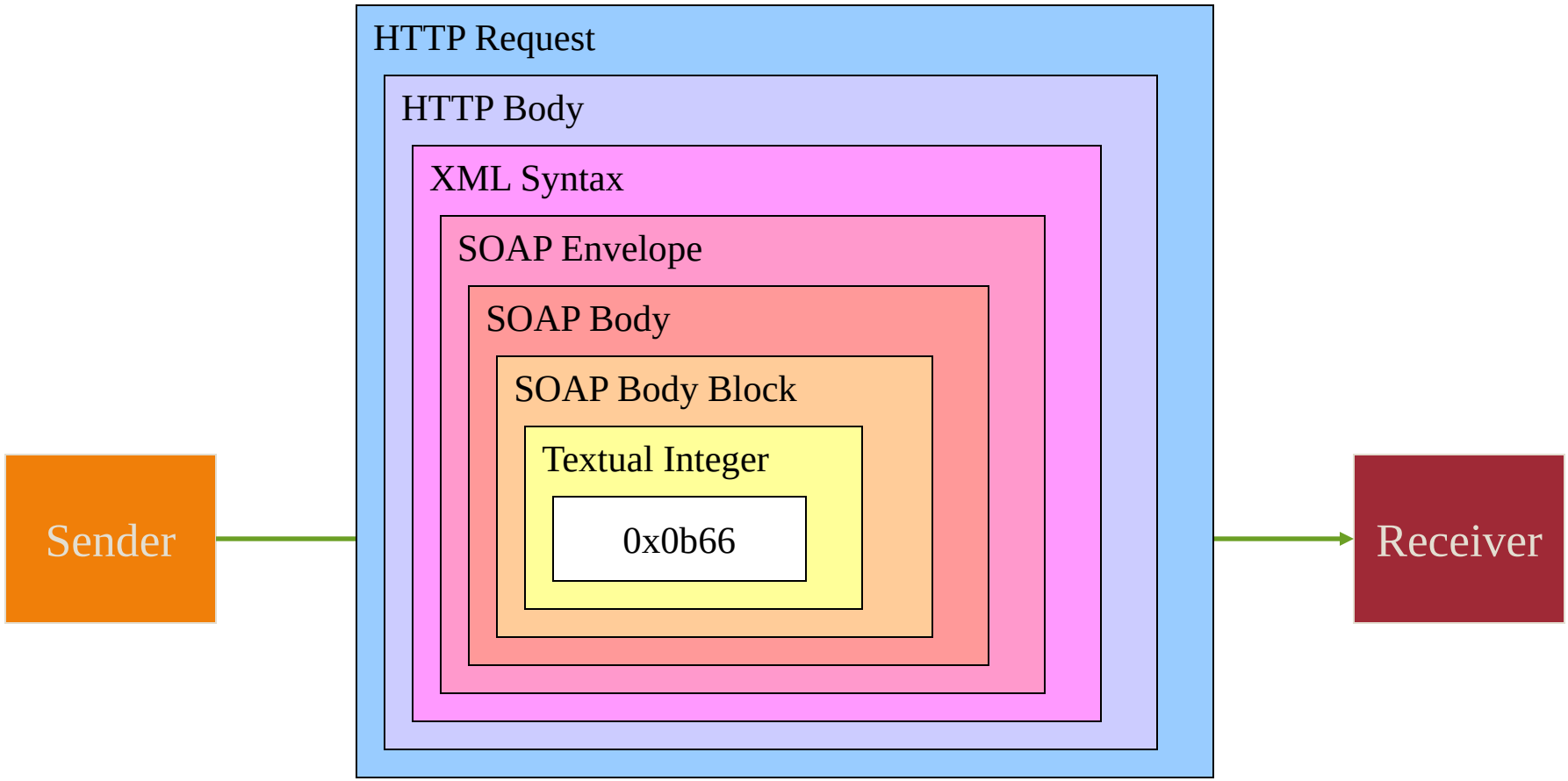
Message Representation


```
<?xml version="1.0" ?>
<env:Envelope xmlns:env="http://www.w3.org/soap-envelope">
  <env:Header>
    <data:headerBlock
      xmlns:data="http://example.com/header"
      env:actor="http://example.com/actor"
      env:mustUnderstand="true">
      ...
    </data:headerBlock>
    ...
  </env:Header>
  <env:Body>
    <data:bodyBlock xmlns:data="http://example.com/header">
      ...
    </data:bodyBlock>
    ...
  </env:Body>
</env:Envelope>
```

Independence

- Independent of the wrapped data
  - True, but…
    - Only text data is allowed
    - Some data structures are difficult to represent
- Independent of the transport protocol
  - True, but…
    - The XML Protocol Working Group has requested additions to the HTTP 1.1 specification

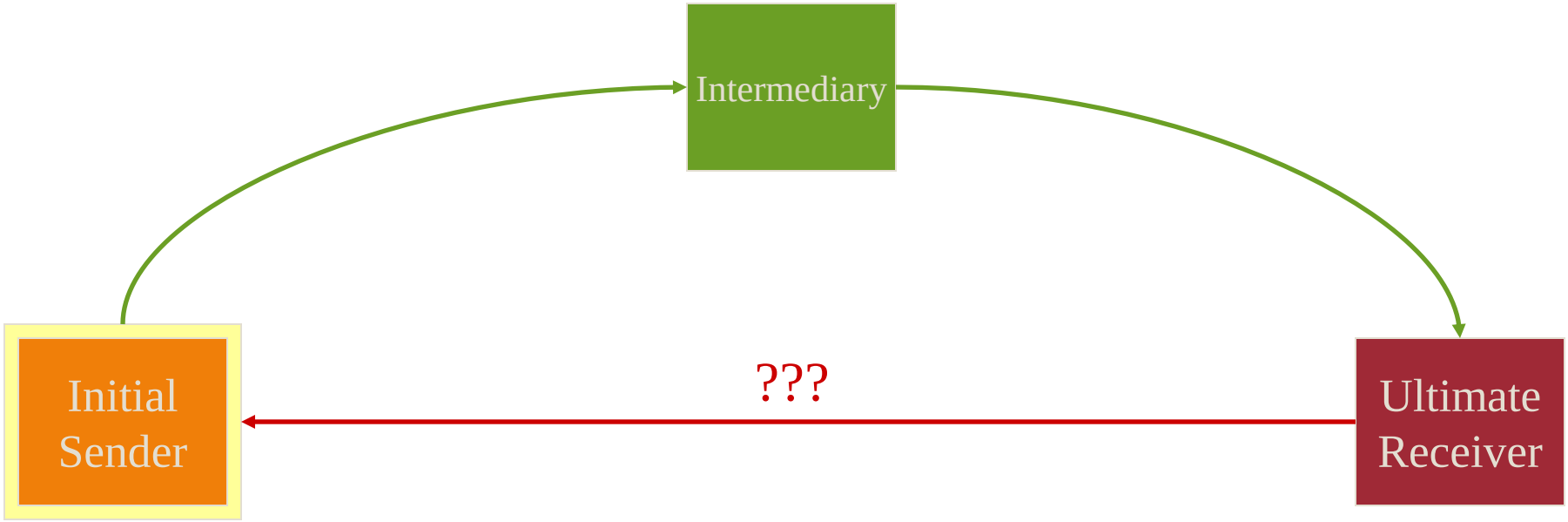
(In)Efficiency



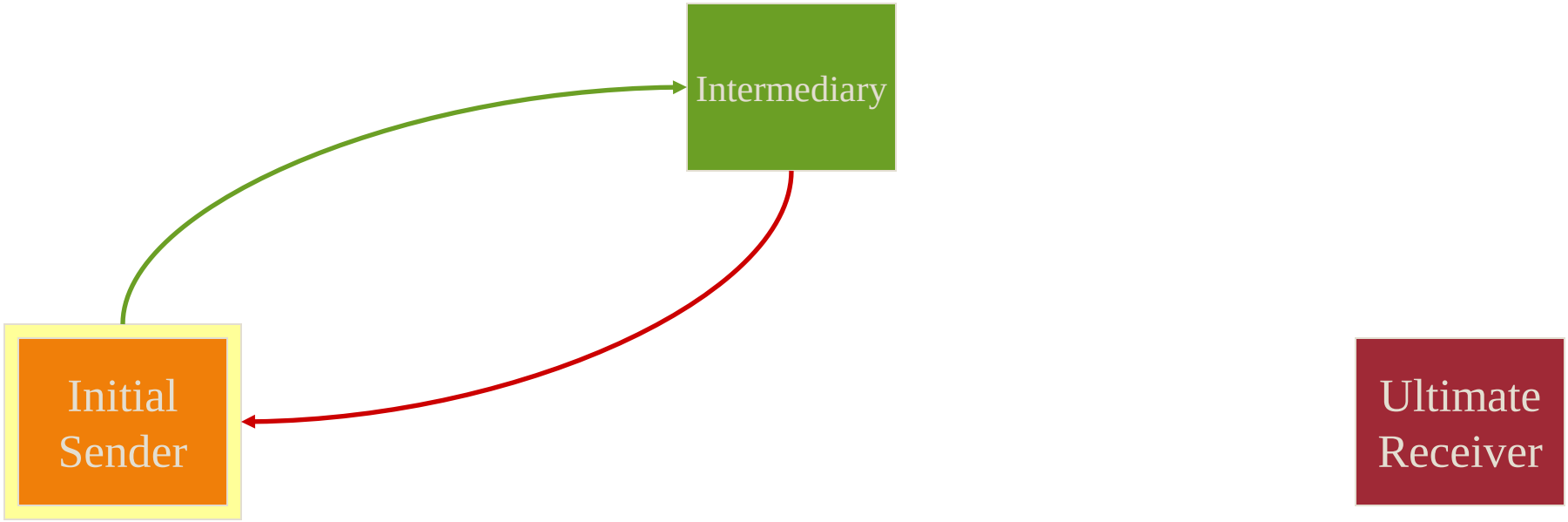
Exchange Paradigm

- Point-to-point exchange
  - Sender, receiver, possible intermediaries
- Uni-directional message exchange
  - True, but…
    - Specification includes semantics for dealing with faults
    - Faults cannot be ignored
    - Faults must be reported to the sending node

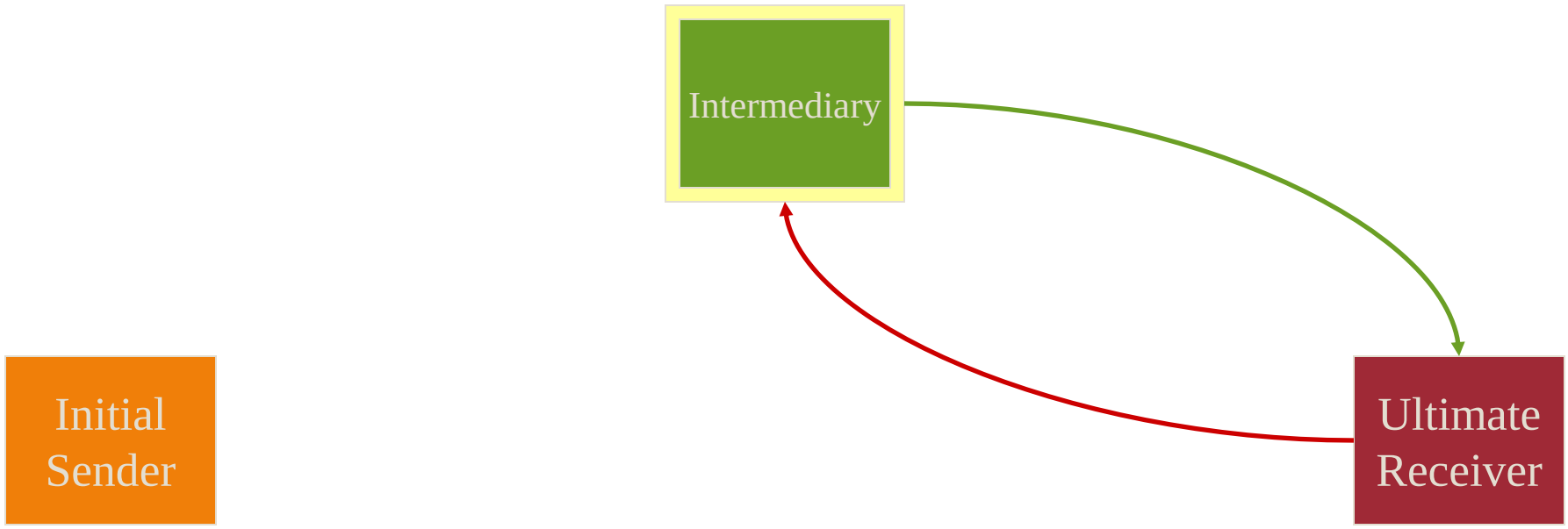
Uni-directional Exchange



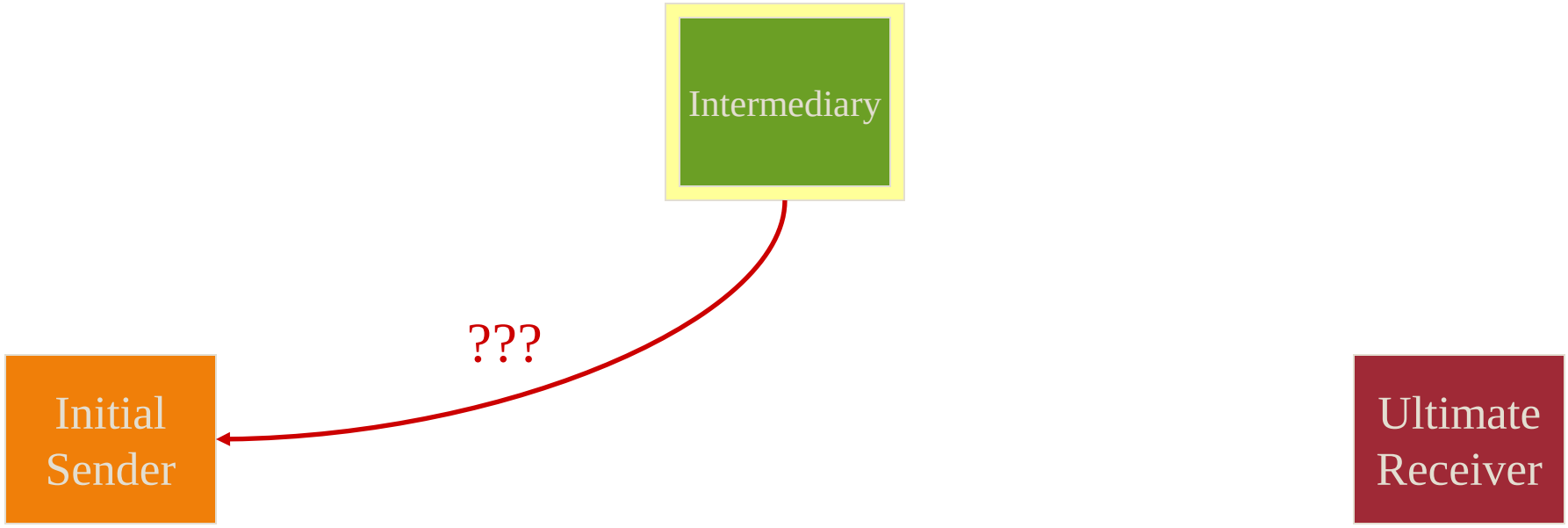
Bi-directional Exchange (Series)



Bi-directional Exchange (Series)



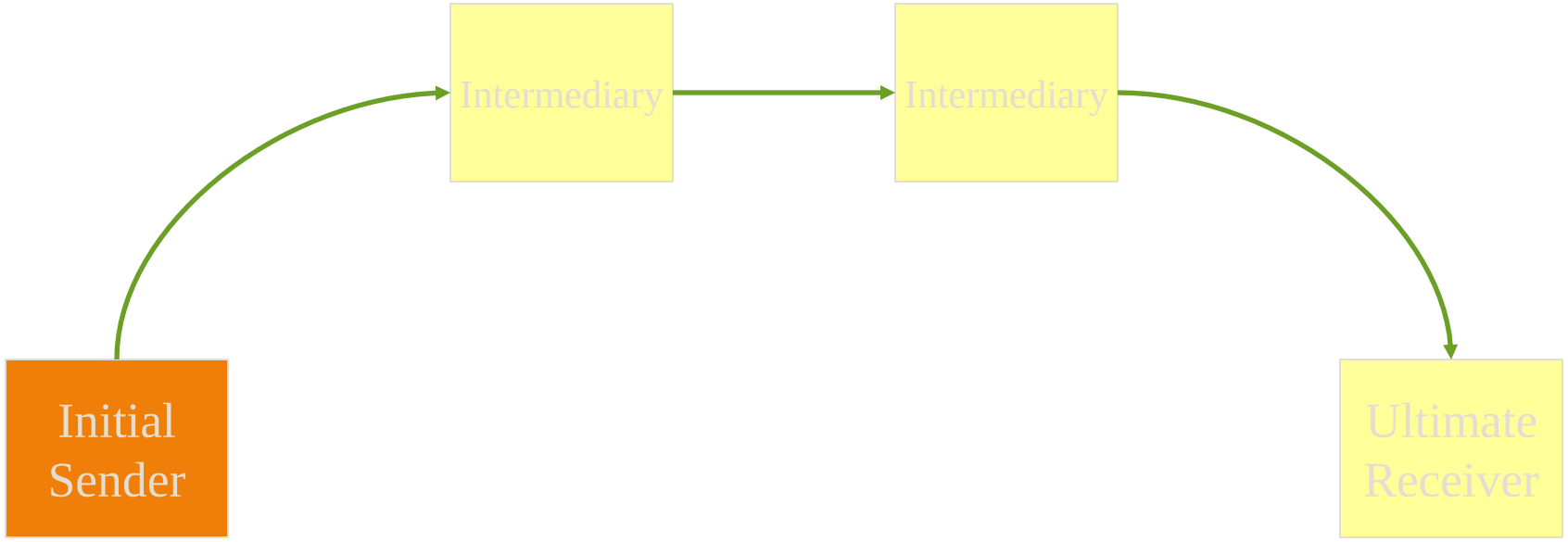
Bi-directional Exchange (Series)



Processing Model

- Point-to-point (sender-to-receiver) exchange, possibly via intermediaries
- Receivers assume “roles” as actors
- Header blocks can be specific to actors
  - Body blocks are always specific to the ultimate receiver
- Actors can be required to understand header blocks

Nodes


Intermediary
Intermediary
Initial Sender
Ultimate Receiver

Actors

- Standard actors
  - none
  - next
  - Anonymous
- Application-specific actors
  - Can be anything
  - Semantics implied by a URI

Actor-specific Header Blocks


```
<data:headerBlock
  xmlns:data="http://example.com/header"
  env:actor="http://example.com/actor1"
  env:mustUnderstand="true">
  ...
</data:headerBlock>
<data:headerBlock
  xmlns:data="http://example.com/header"
  env:actor="http://example.com/actor2"
  env:mustUnderstand="true">
  ...
</data:headerBlock>
<data:headerBlock
  xmlns:data="http://example.com/header"
  env:actor="http://example.com/actor2"
  ...
</data:headerBlock>
```

Intermediary Algorithm

- Receive message
- Process appropriate header blocks
  - Processing possibly produces a fault
- Remove processed headers
- Add new headers
- Send message

Ultimate Recipient Algorithm

- Receive message
- Process appropriate header blocks
  - Processing possibly produces a fault
- Process all body blocks
  - Processing possibly produces a fault

Higher-level Exchange Paradigms

- RPC
  - Fits well with HTTP 1.1 binding
  - Current activity within the XML Protocol Working Group
- Conversational
  - Fits well with general message passing, but awkward with HTTP 1.1 binding

Normative References

- http://www.w3.org/2000/09/XML-Protocol-Charter
- http://www.w3.org/2002/ws/Activity.html
- http://www.w3.org/TR/xmlp-reqs/
- http://www.w3.org/TR/xmlp-am/
- http://www.w3.org/TR/xmlp-scenarios/
- http://www.w3.org/TR/soap12-part0/
- http://www.w3.org/TR/soap12-part1/
- http://www.w3.org/TR/soap12-part2/