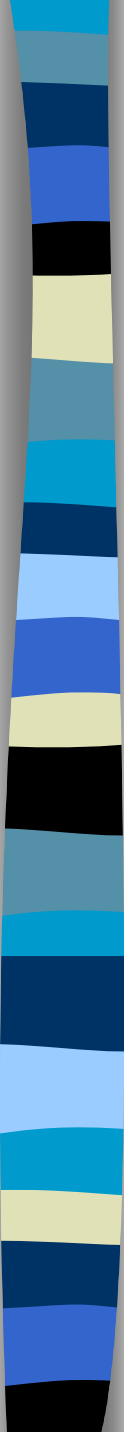


# Abstract Model Subgroup



Stuart Williams  
skw@hplb.hpl.hp.com

Mark Baker, Sun Microsystems, ([mark.baker@canada.sun.com](mailto:mark.baker@canada.sun.com))  
Martin Gudgin, DevelopMentor, ([marting@develop.com](mailto:marting@develop.com))  
Oisin Hurley, Iona, ([ohurley@iona.com](mailto:ohurley@iona.com))  
Marc Hadley, Sun, ([marc.hadley@uk.sun.com](mailto:marc.hadley@uk.sun.com))  
John Ibbotson, IBM Corporation, ([john\\_ibbotson@uk.ibm.com](mailto:john_ibbotson@uk.ibm.com))  
Scott Isaacson, Novell Inc. ([SISAACSON@novell.com](mailto:SISAACSON@novell.com))  
Yves Lafon, W3C, ([ylafon@w3.org](mailto:ylafon@w3.org))  
Jean-Jacques Moreau, Canon, ([moreau@crf.canon.fr](mailto:moreau@crf.canon.fr))  
Henrik Frystk Nielsen, Microsoft Corporation, ([frystyk@microsoft.com](mailto:frystyk@microsoft.com))  
Krishna Sankar, Cisco Systems, ([ksankar@cisco.com](mailto:ksankar@cisco.com))  
Nick Smilonich, Unisys, ([nick.smilonich@unisys.com](mailto:nick.smilonich@unisys.com))  
Lynne Thompson, Unisys, ([Lynne.Thompson@unisys.com](mailto:Lynne.Thompson@unisys.com))  
Stuart Williams, Hewlett-Packard Company, ([skw@hplb.hpl.hp.com](mailto:skw@hplb.hpl.hp.com))



- Overview of Model
- Status
- Plans
- Issues
- Questions



# Overview of Model

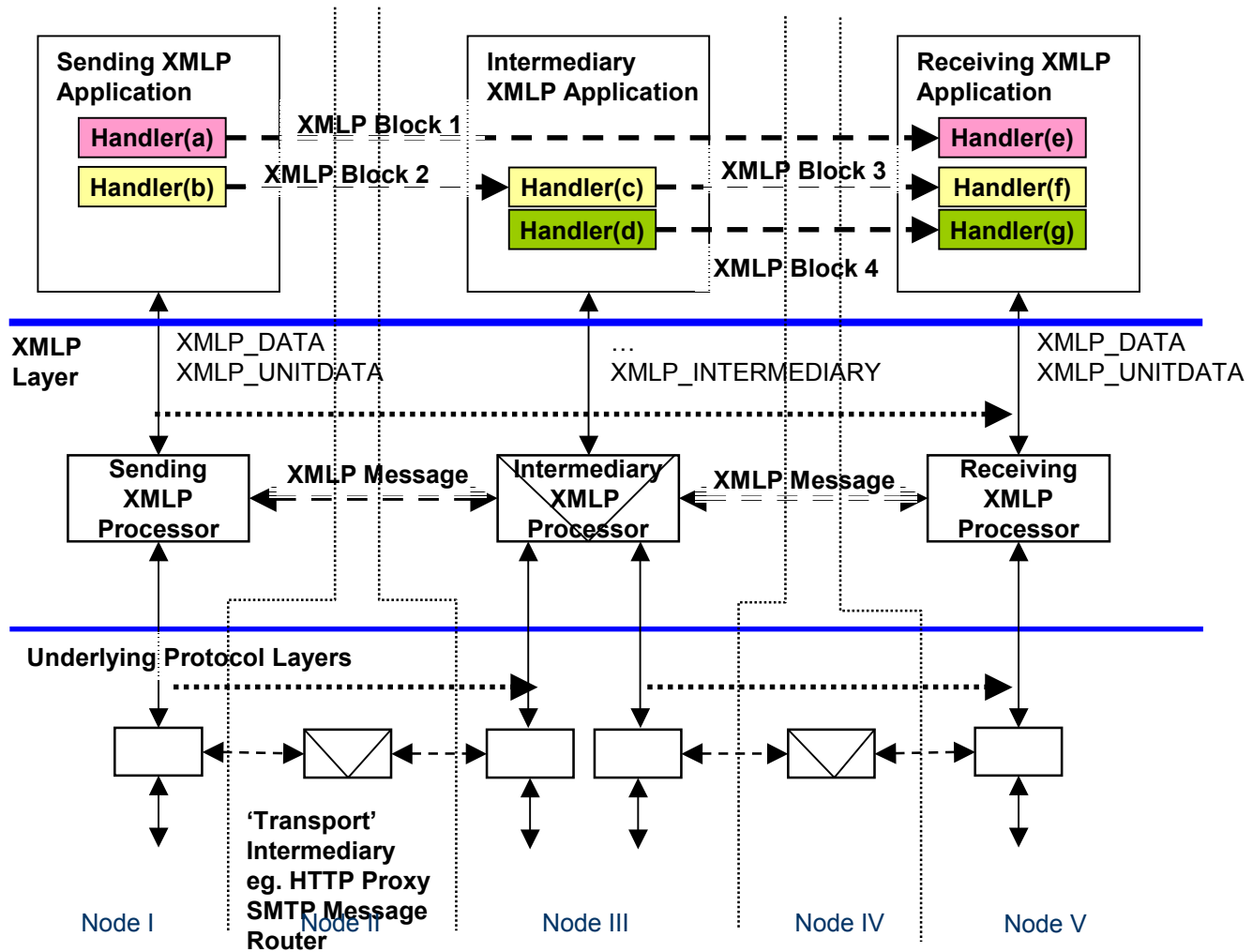
- Descriptive Tool
- Focus on WHAT, not HOW
- A means to develop clarity over the WHAT XMLP is and what (functionally) it does.
- A potential means to partition the design task.
- A potential means to structure a document collection.



# Model Outline

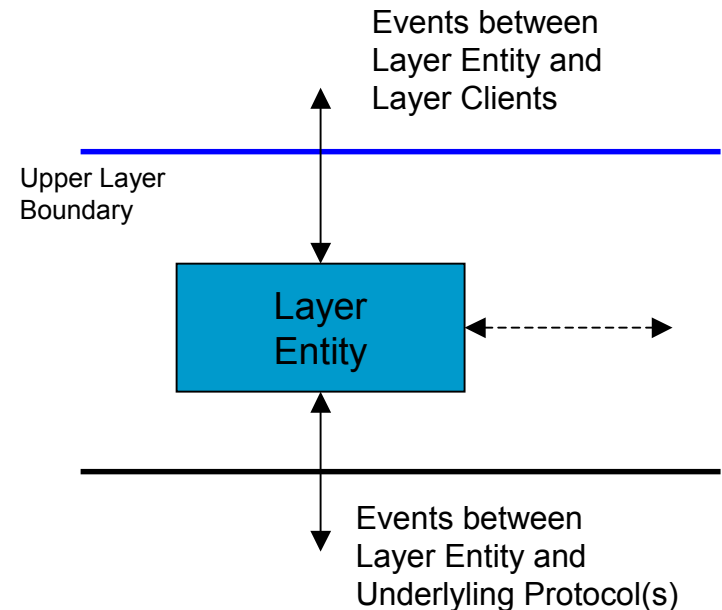
- Top Level Overview
- Service Abstraction
  - One-way, Two-way request/response and Intermediary operation.
- Modules and Applications
  - Path, Targetting
- Binding
  - BindingContext and Arbitrary Attachments
- Security
  - Mostly a placeholder defers to Bindings and Modules

# Top-Level Diagram

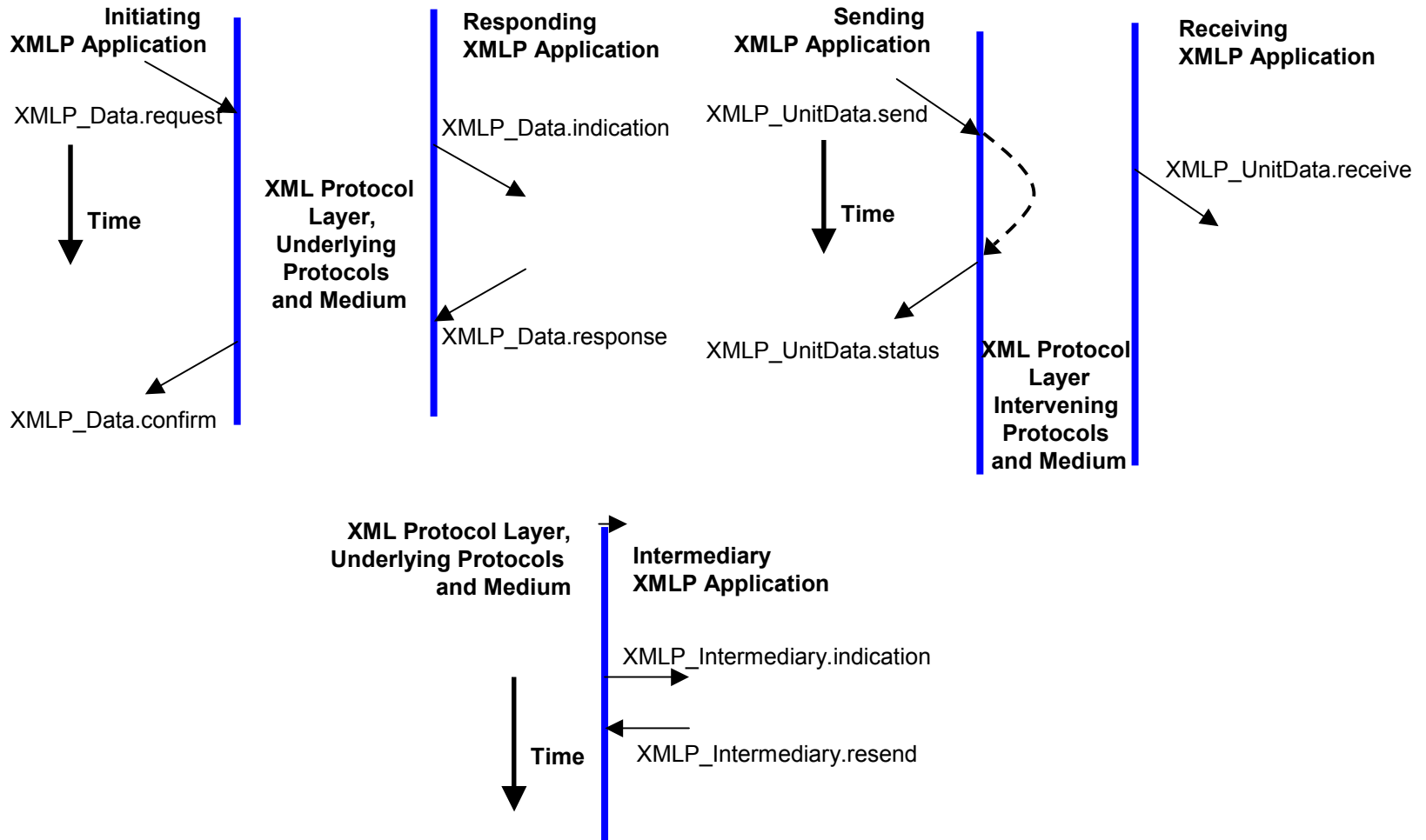


# Service Abstraction

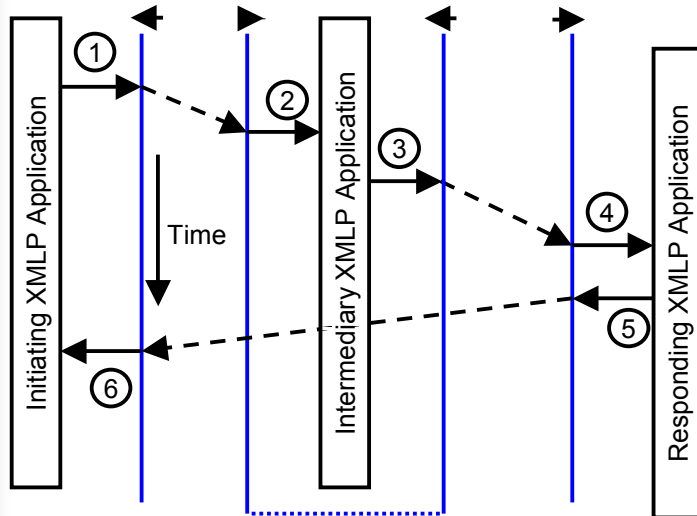
- It's 'hard' to abstract the functionality of arbitrary modules!
- XMLP Handlers 'positioned' above the XMLP Layer into XMLP Applications
- The core abstraction is then around XMLP message exchange (through processing intermediaries).
- Service Abstraction describes protocol operations as event patterns seen from 'above'.
- Layer Entity (XMLP Processor) implements the procedural rules of the protocol (which have yet to be designed).
- Layer Supports 3 operations for the exchange of XMLP messages
  - XMLP\_Data (Two-way req/resp.) (4 events)
  - XMLP\_UnitData (one-way) (3 events)
  - XMLP\_Intermediary (2 events)



# Operations Summary

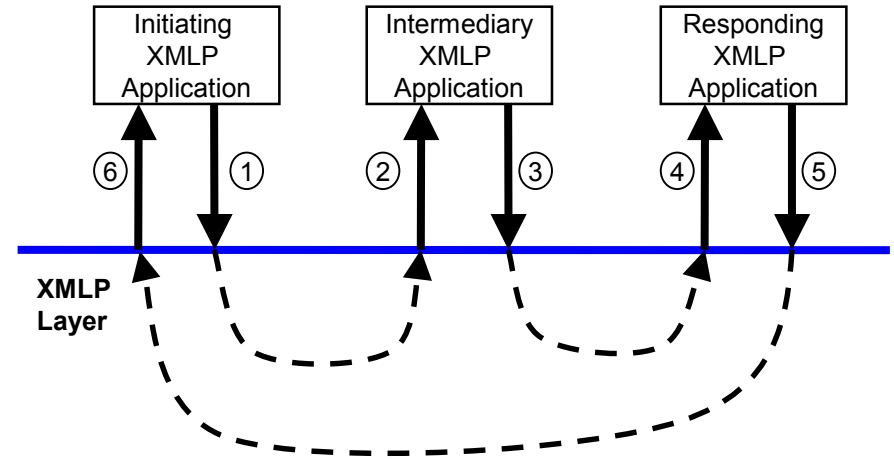


# Operation through Intermediaries



## Layer Primitives Key

1. XMLP\_Data.request
2. XMLP\_Intermediary.indication
3. XMLP\_Intermediary.resend
4. XMLP\_Data.indication
5. XMLP\_Data.response
6. XMLP\_Data.confirm



## Layer Primitives Key

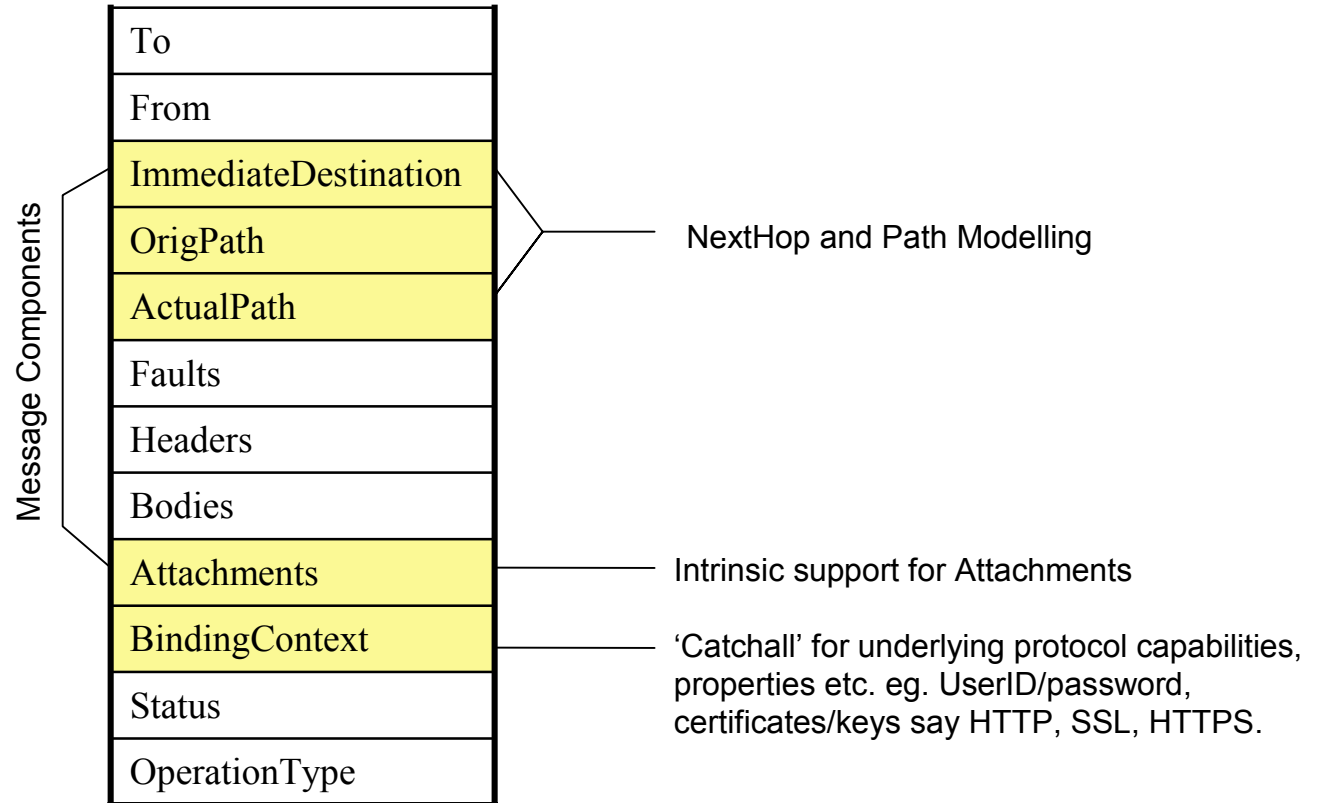
1. XMLP\_Data.request
2. XMLP\_Intermediary.indication
3. XMLP\_Intermediary.resend
4. XMLP\_Data.indication
5. XMLP\_Data.response
6. XMLP\_Data.confirm

Numerical ordering indicates time sequence

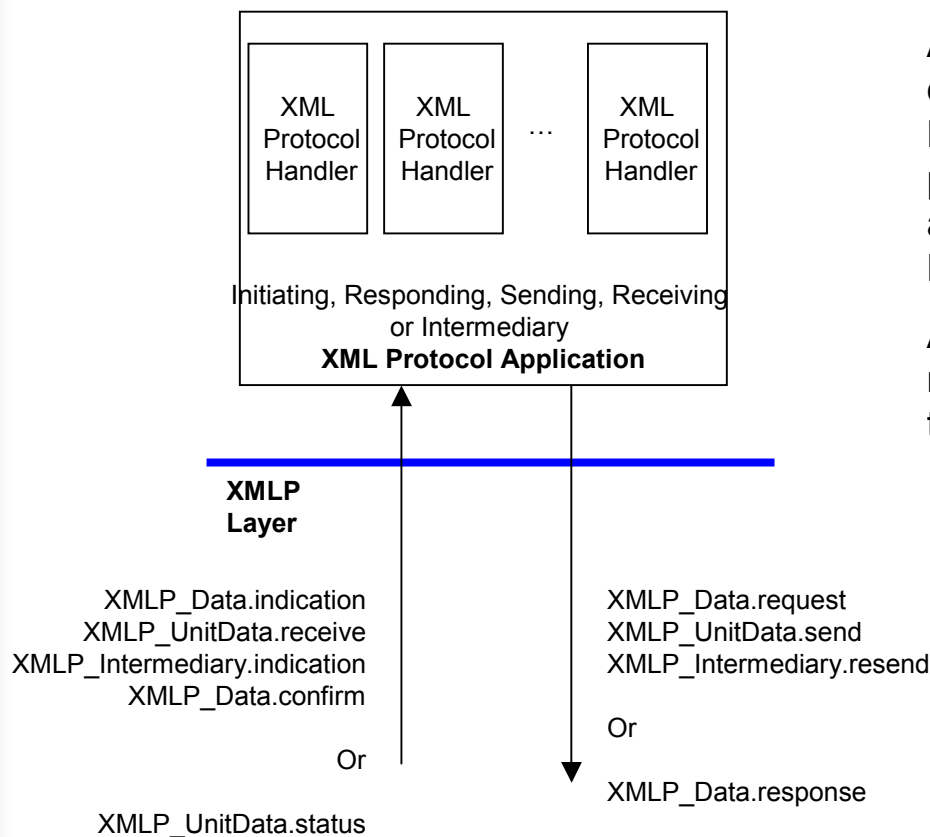
NB. Response could also be processed by an Intermediary



# Event Parameters



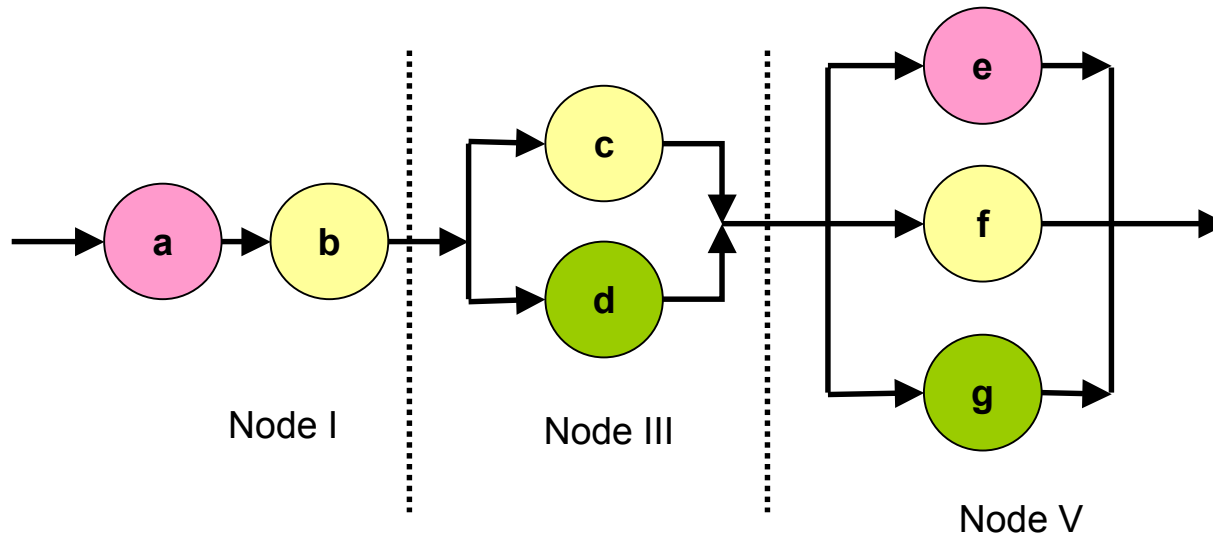
# Applications, Modules and Handlers



An XML Protocol Module encapsulates the definition of one or more related **XML Protocol Blocks** and their associated processing rules. These processing rules are realised in one or more **XML Protocol Handlers**.

An **XML protocol application** is responsible for identifying and dispatching targeted **XML protocol handlers**

# Message Path and Targeting



Node numbering from figure 2.1

- This is a topic of active debate – how/whether to support message path as 1<sup>st</sup> class notion in XMLP
- *ImmediateDestination*, *OrigPath* and *ActualPath* make current draft neutral to the resolution of this discussion.
- Need for terminology around types of targeting: Directed, Functional, Group...
- Issues of handler processing order semantics.



# Binding and Attachments

- Service Abstraction provides intrinsic support for Arbitrary Attachments
  - Implications:
    1. An intrinsic mechanism for carrying and referencing arbitrary attachments carried (somewhere) within the outermost XML construct (for bindings to protocols that don't have intrinsic support for arbitrary attachments).
    2. Binding specific mechanisms to carry attachments in more 'efficient' ways.
- ebXML/SOAP with Attachments discussion.
- Topic of active discussion
- BindingContext – mentioned earlier.
  - Container for parameters/properties of underlying protocols
  - Eg. UserID/Passwd, Key(Refs)/Certificates, QoS...
  - Means to deliver binding specific context information to XMLP Application and Handlers.



# Security

- Refers to BindingContext to exploit any underlying security features.
- Refers to XMLP Modules (extensions) for Application layer security.



# AMG Document Status

- Work of a subgroup of the XMLP-WG
  - 4 Phone Conferences – One major rewrite.
- Good handle on basic message exchanges and model of XMLP processing intermediaries.
- Section 3. (Service Abstraction) is problematic for some.
- Treatment of Fault Handling, Paths, Targeting and Attachments is still open (require work).



# AMG Plans

- Solicit Feedback from WG
  - Need to know whether WG regarded this as a useful contribution.
- Participate in and synthesis from Path/Targeting and Arbitrary Attachment discussions
- Seed further discussion of Fault handling.
- ‘Road-test’ Model against DS’s, S’s, SOAP 1.1 and SOAP with Attachments.



# AMG Issues

- Requirements Glossary and AM Definition of Terms – These are closely related and in most cases aligned. Need to converge to a single reference.
- Section 3. Stuart regards it as a crucial part of abstraction of XMLP - the WHAT of XMLP. Henrik regards it as being too close to an API definition (larger note in Issues section).
- Intrinsic support for BOTH two-way request/response and one-way operations... do we need both?
- Questions...