

Proposed IEEE 1451.0 to P1451.5 Interface

Jeff Burch

Agilent Technologies

jeff_burch@agilent.com

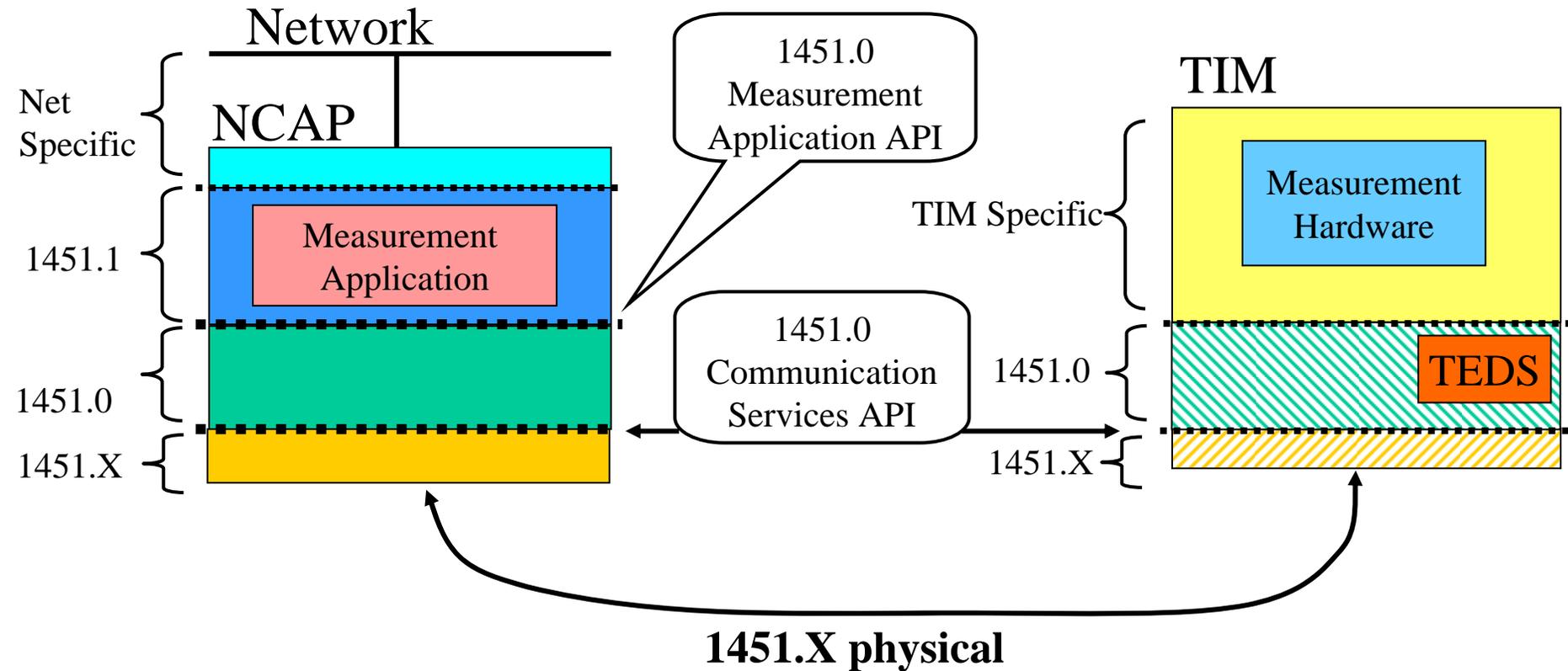


Agilent Technologies

Agenda

- Introduction
- Interface Goals
- Modes of Communication
- APIs
- Sequence Diagram
- Outstanding Issues
- Summary

Introduction



The 1451.0 standard:

- Provides a communication abstraction between NCAP and TIMs.
- Support many physical communication technologies.
- Defines "Communication Services API" ← **Subject of this presentation**
- Defines "Measurement Application API (not covered in this presentation)"

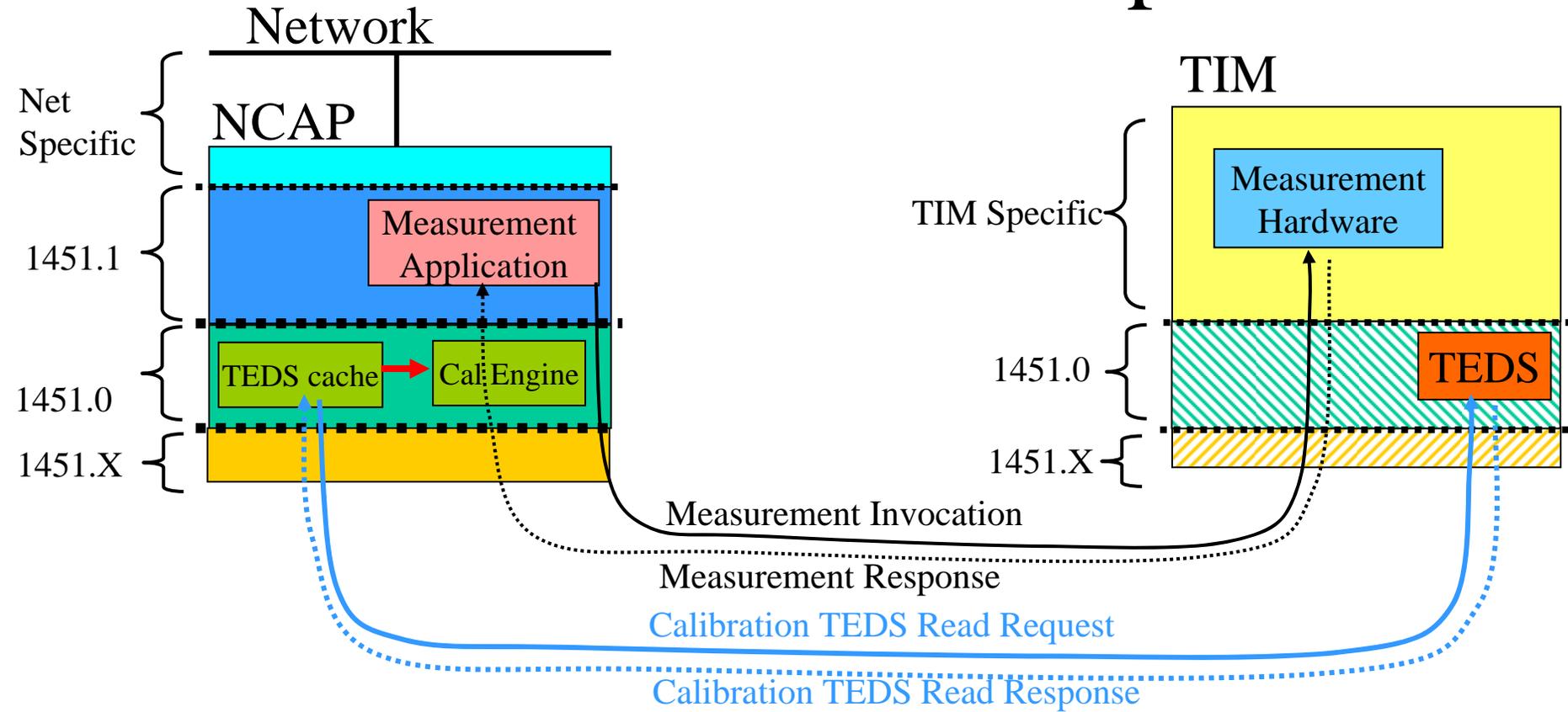
Communication API Goals

- Make it easy to add a new physical layer between NCAP and TIM
- Provide a useful abstraction for communications between NCAP and TIMs
- Define a language-neutral API and interface semantics
- Allow implementers the ability to use native communication features and mechanisms

Modes of Communication

- Client / Server style
 - Always One-to-One
 - Bi-directional information flow
 - Optional one-way
 - Synchronous and asynchronous forms
- Publish / Subscribe style
 - One-to-Many, may be One-to-One or One-to-All
 - Unidirectional information flow
 - Non-acknowledged
- Fully Symmetric Communication
 - NCAP and TIMs can be Client, Server, Publisher, and Subscriber simultaneously

Client / Server Example

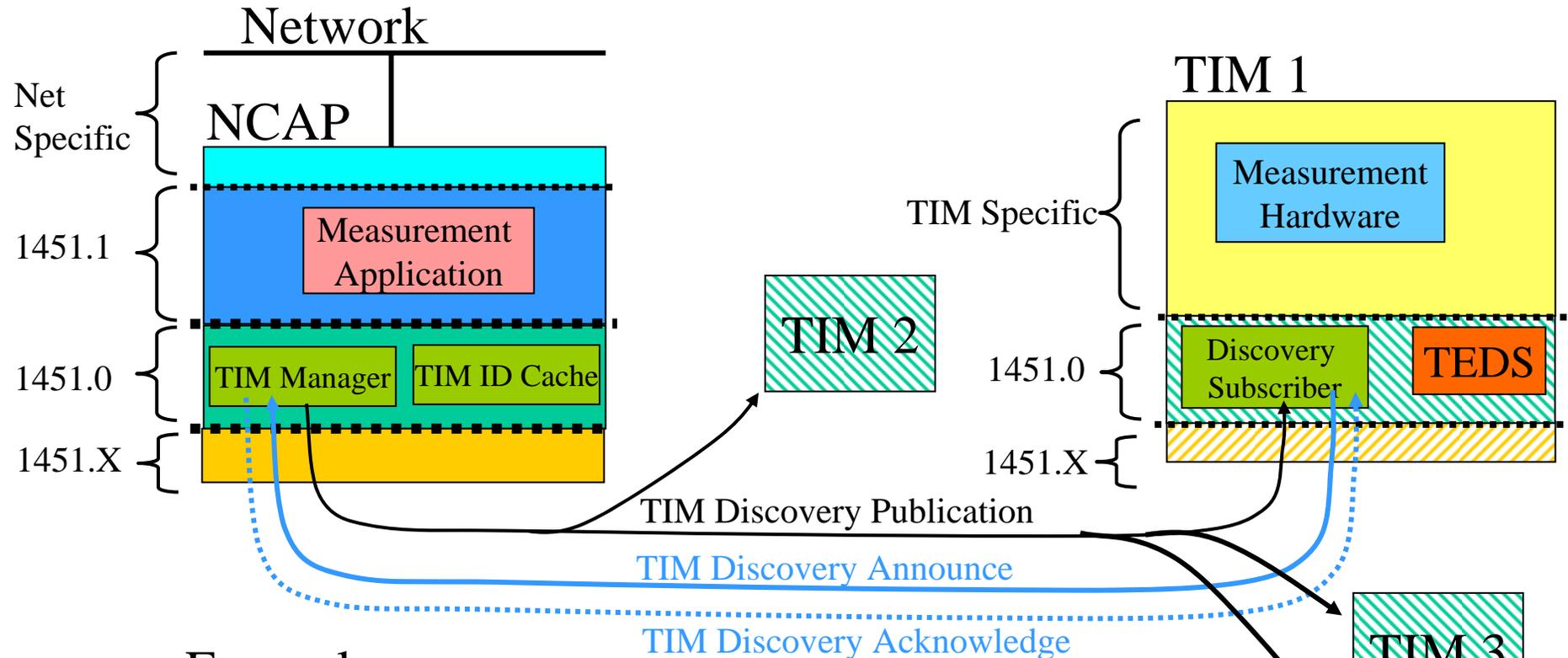


Examples:

- TEDS Read / Write operation: NCAP is client, TIM is server
- Polled Measurement: NCAP is client, TIM is server

Publish / Subscribe

TIM Discover Example

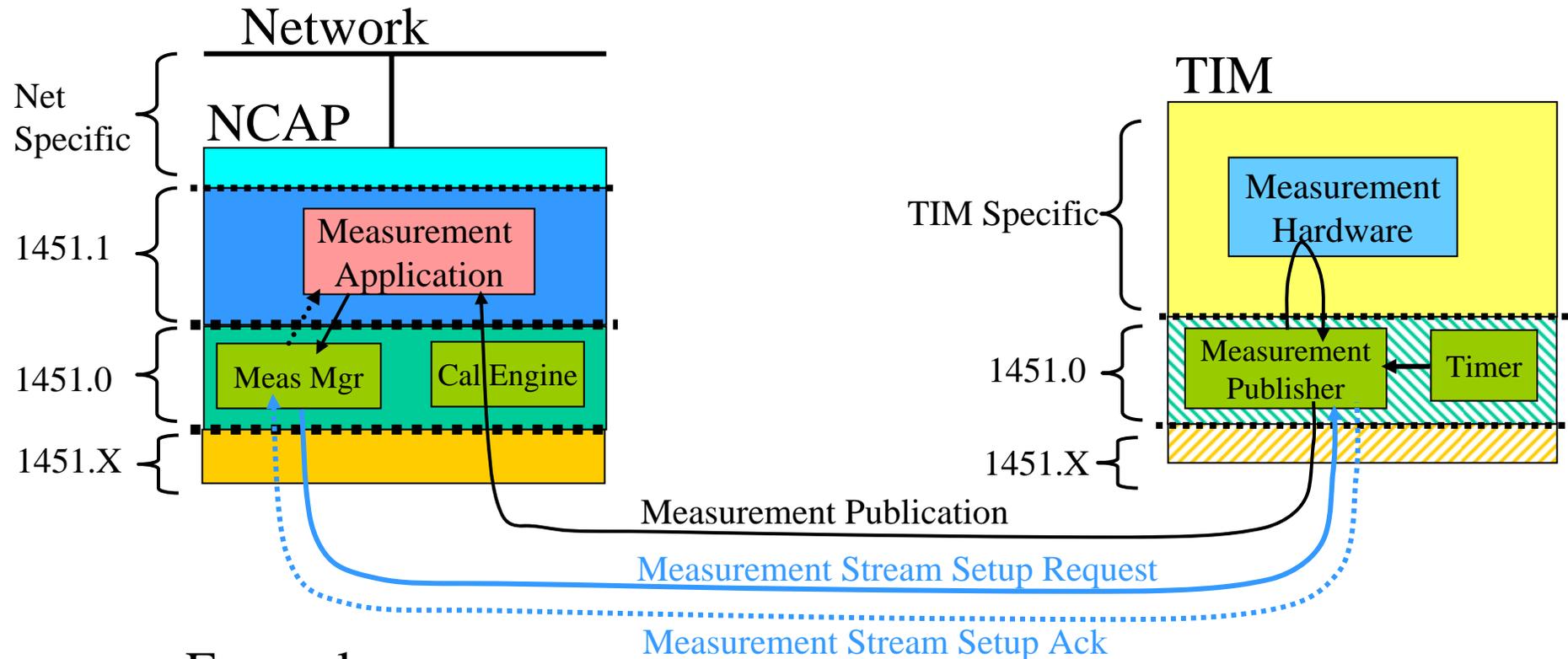


Example:

- NCAP attempts to discover available TIMs by publishing to all "Discovery Subscribers"
- Each TIM responds by sending "Discovery Announce" information back to the NCAP by a Client / Server message

Publish / Subscribe

Measurement Stream Example



Example:

- NCAP sets up streaming measurement with client / server message
- TIM's periodic timer triggers measurement process
- Measurement results are published back to NCAP

Client / Server APIs

ServerReceiveInterface

(from ieee1451dot0::CommServicesInterface)

+receive(messageID:UInt16,payload:OctetArray):OctetArray

AsyncResponseInterface

(from ieee1451dot0::CommServicesInterface)

+receiveResponse(messageID:UInt16,payload:OctetArray):void

MessageDispatchInterface

(from ieee1451dot0::CommServicesInterface)

+openClient(nodId:UInt16,objId:UInt8,replyTo:AsyncResponseInterface):UInt16

+registerServer(objId:UInt8,callback:ServerReceiveInterface):UInt16

+close(id:UInt16):void

+send(id:UInt16,payload:OctetArray):UInt16

+setQoS(id:UInt16,mode:UInt16,args:ArgumentArray): Boolean

Publish / Subscribe API

SubscriptionReceiveInterface

(from ieee1451dot0::CommServicesInterface)

+notify(id:UInt16,payload:OctetArray):void

PublicationDispatchInterface

(from ieee1451dot0::CommServicesInterface)

+openPub(domain:UInt8,groupId:UInt8):UInt16

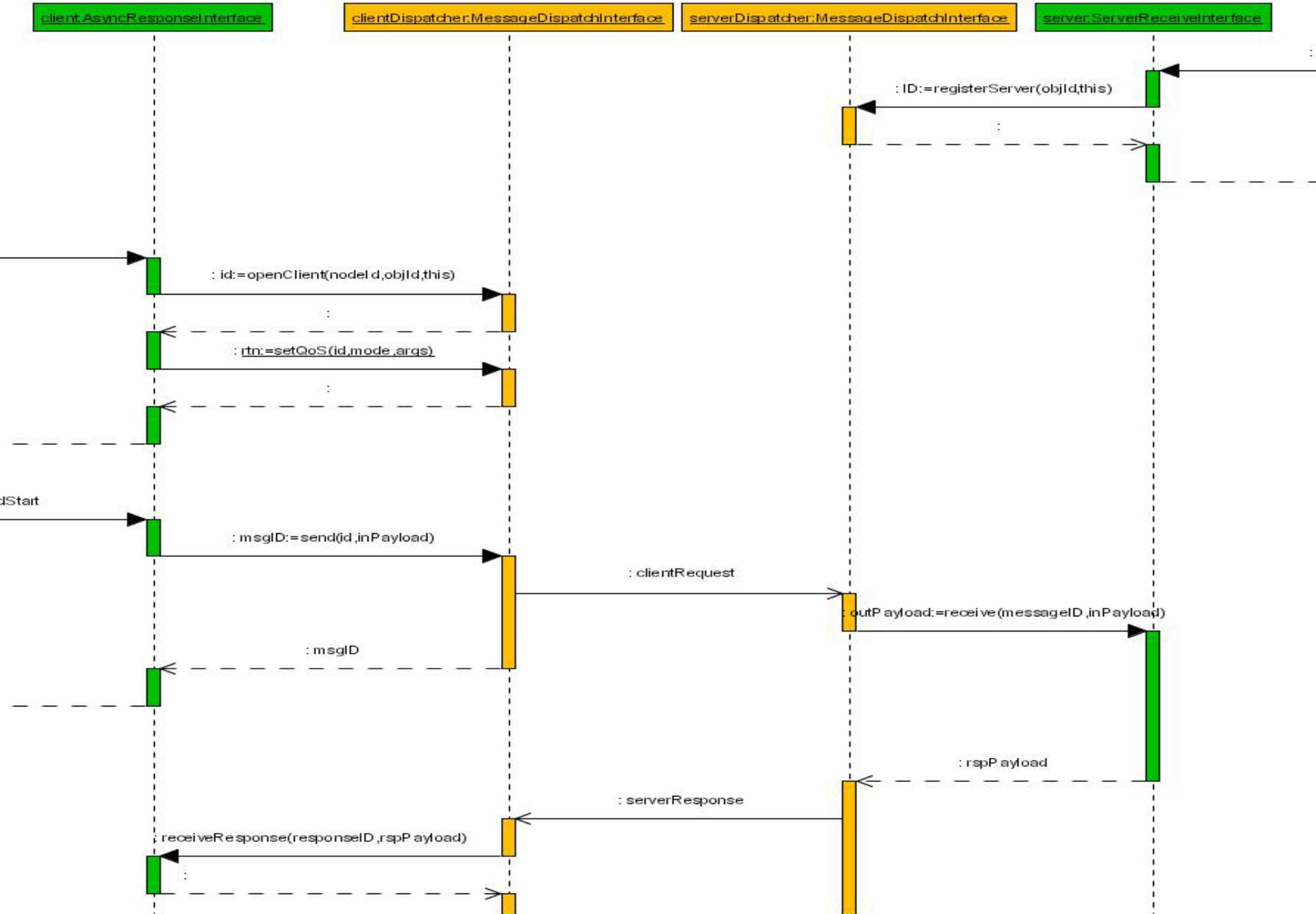
+openSub(domain:UInt8,groupId:UInt8,callback:SubscriptionReceiveInterface):UInt16

+close(id:UInt16):void

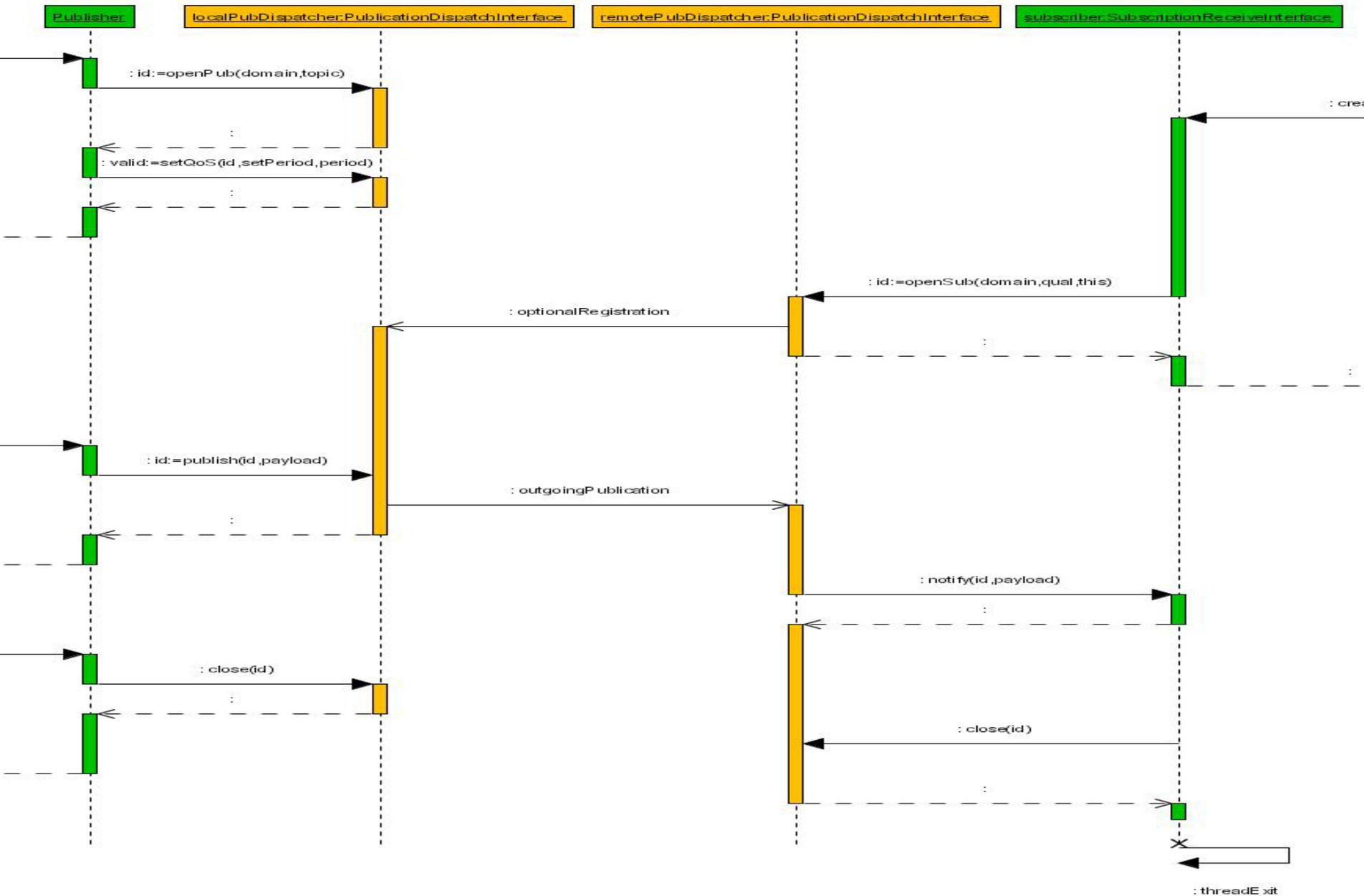
+publish(id:UInt16,payload:OctetArray):UInt16

+setQoS(id:UInt16,mode:UInt16,args:ArgumentArray): Boolean

Client / Server Sequence



Publish / Subscribe Sequence



Outstanding API Issues

- Representation of Communication Endpoints
 - UInt16 NodeID, UInt8 ObjId
- Quality of Service Parameters
 - Repetition Rate (e.g. 200 Hz or 5 msec/sample)
 - Payload Size (e.g. 1024 UInt16 values)
 - Latency Requirement (e.g. 2 msec maximum delay)
- Examples for Specific 1451.X Technologies
 - 1451.5 / 802.11 using UDP/IP and TCP/IP

Communication API Goals Review

- Make it easy to add a new physical layer between NCAP and TIM
 - ← ~10 functions to implement, ~3 functions to call
- Provide a useful abstraction for communications between NCAP and TIMs
 - ← Client/Server and Pub/Sub models of communication with QoS mechanism tailored for measurement applications
- Define a language-neutral API and interface semantics
 - ← API specified in IDL with examples in UML, Java, C++, and C
- Allow implementers the ability to use native communication features and mechanisms
 - ← Logical communication specified. Implementers provide conversion from logical to physical communication. No assumptions made on available native mechanisms.