

# Concurrent Types as Engineering Principles for Large Distributed Systems



<http://mrg.doc.ic.ac.uk/>

Nobuko Yoshida  
Imperial College London

# The Kohei Honda Prize for Distributed Systems Queen Mary, University of London

Posted with permission from QMUL on 17<sup>th</sup> Dec 2013. [Original article](#) written by Edmund Robinson.

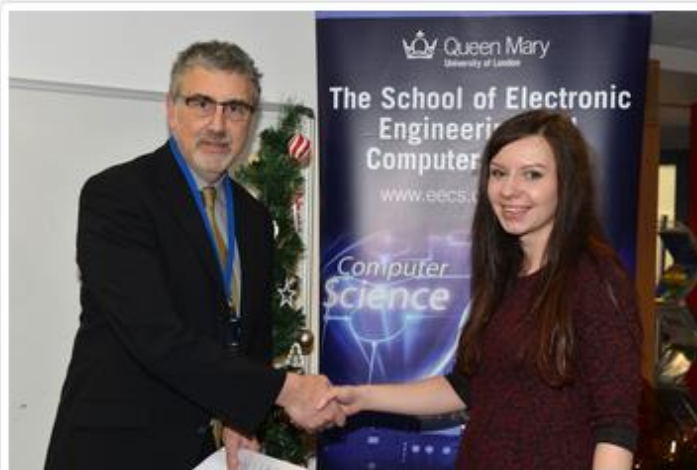
This prize was instituted in 2013 and is awarded annually to one undergraduate student and one postgraduate student in recognition of their achievement in applying the highest quality scientific and engineering principles in the broad area of Distributed Systems. This is the area in which Dr Honda concentrated most of his teaching, and it is also the area in which he conducted his research. Its primary funding comes from a donation from his family, who wished to commemorate Dr Honda in this way. Additional funding has come from Dr Honda's own ETAPS Award. This prize is sponsored by Springer Verlag, and awarded annually by the ETAPS committee in recognition of an individual's research contribution. Dr Honda received the first such award posthumously, and the awarding panel expressed a wish that the funding be used to supplement this prize fund. The laudation for this award, written by Dr Honda's colleague, Prof Vladimiro Sassone is included later.

## About Dr Honda

Kohei Honda was born and lived the first part of his life in Japan. Like many scientists he was fascinated by the idea of finding basic explanatory theories, like the physicists looking for grand unified theories of the universe. Kohei, though, was passionately interested in finding the right basic explanatory theory for the process of computation. Most academics agree that the basic theory



## Winners 2013



**Ms Anna Pawlicka**

2013 winner (Undergraduate) source: QMUL



**Mr. Valdmir Negacevshi**

2013 winner (Postgraduate) source: QMUL

# Open Problems

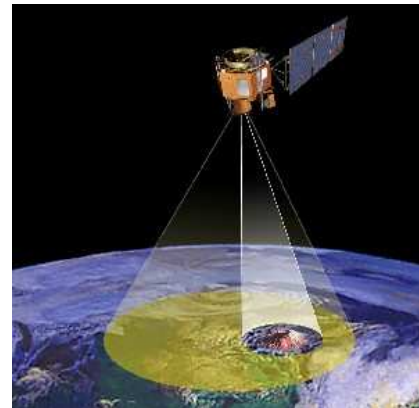
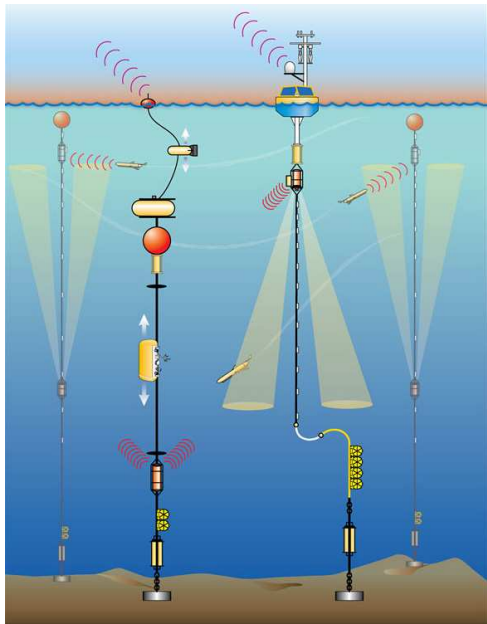
- The way to organise software is increasingly based on communications (Cloud Computing, Many Cores,...)
- **Question**
  - How to **formally** abstract/specify/implement/control communications?
  - How to apply mobile processes and their type theories to real distributed applications and programming languages?

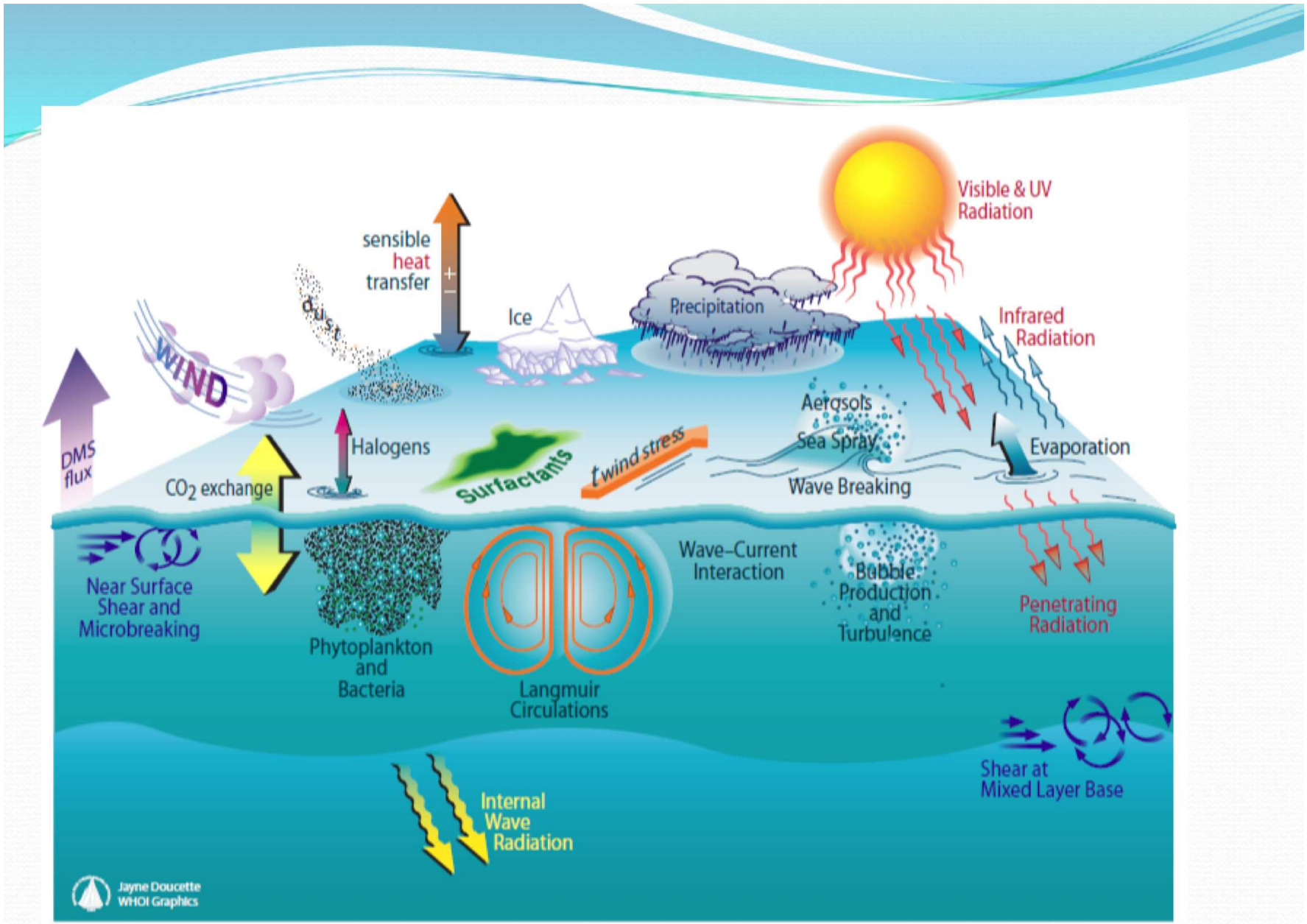
# Open Problems

- The way to organise software is increasingly based on communications (Cloud Computing, Many Cores,...)
- **Question**  $\implies$  **Multiparty session type theory**
  - How to **formally** abstract/specify/implement/control communications?
  - How to apply mobile processes and their type theories to real distributed applications and programming languages?  
 $\implies$  **large-scale cyberinfrastructure for e-Science**

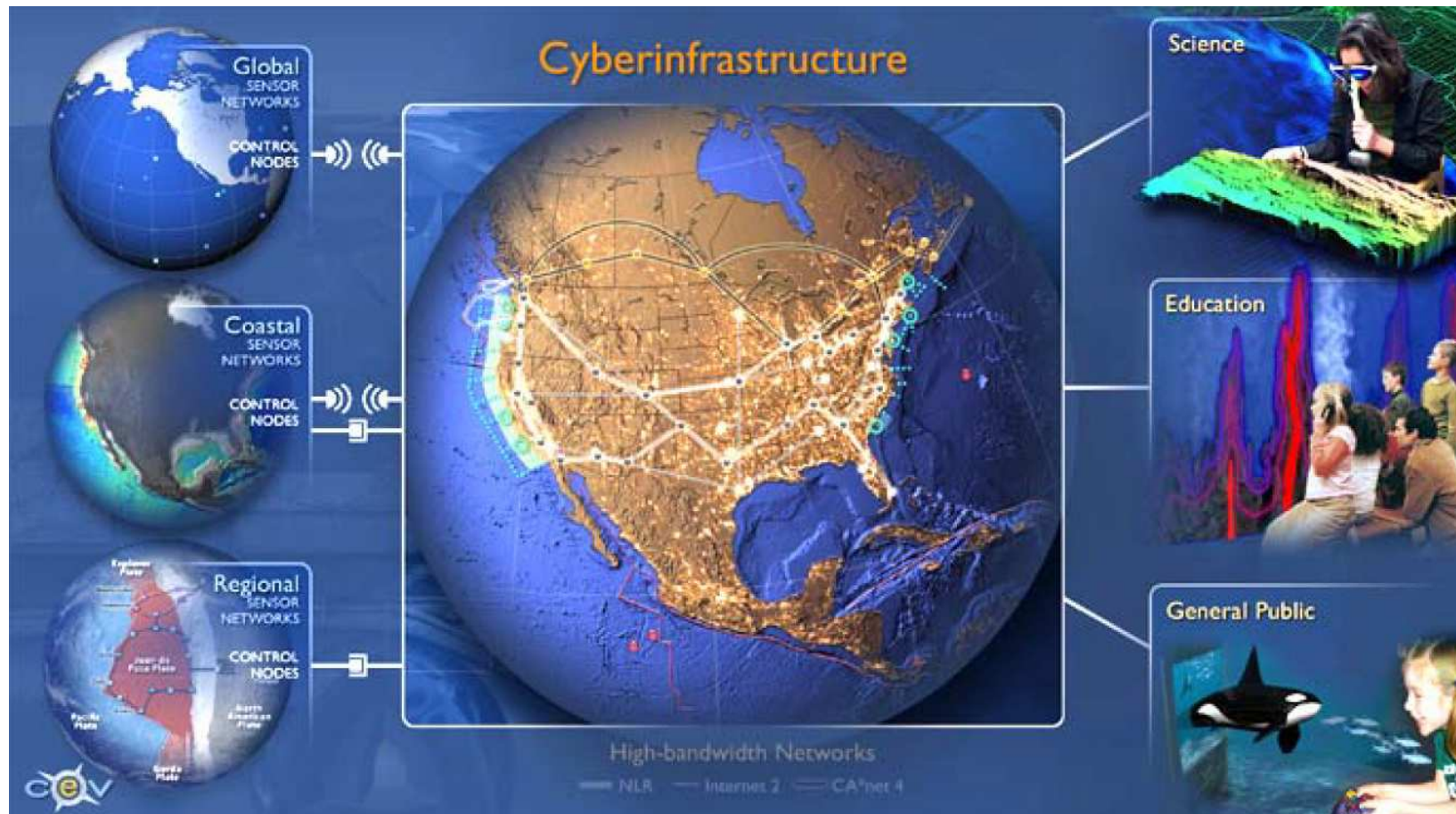
# Ocean Observatories Initiative

- A NSF project (400M\$, 5 Years) to build a cyberinfrastructure for observing oceans around US and beyond.
- Real-time sensor data constantly coming from both off-shore and on-shore (e.g. buoys, submarines, under-water cameras, satellites), transmitted via high-speed networks.





# Ocean Observatories Initiative





## Challenges

- The need to specify, catalogue, program, implement and manage *multiparty message passing protocols*.
- Communication assurance
  - Correct message ordering and synchronisation
  - Deadlock-freedom, progress and liveness
  - Dynamic message monitoring and recovery
  - Logical constraints on message values
- Shared and used over a long-term period (e.g. 30 years in OOI).

## Why Multiparty Session Types?

- Robin Milner (2002): *Types are the leaven of computer programming; they make it digestible.*
  - ⇒ Can describe communication protocols as *types*
  - ⇒ Can be materialised as *new communications programming languages* and *tool chains*.
- *Scalable* automatic verifications (deadlock-freedom, safety and liveness) without *state-space explosion problems* (*polynomial time complexity*).
- Extendable to *logical verifications* and flexible *dynamic monitoring*.

# Dialogue between Industry and Academia

Binary Session Types [PARL'94, ESOP'98]



Milner, Honda and Yoshida joined W3C WS-CDL (2002)



Formalisation of W3C WS-CDL [ESOP'07]



Scribble at  $\pi^4$  Technology

# Pi calculus versus Petri nets: Let us eat “humble pie” rather than further inflate the “Pi hype”

W.M.P. van der Aalst

**Abstract.** In the context of *Web Service Composition Languages* (WSCLs) there is an ongoing debate on the best foundation for *Process-Aware Information Systems* (PAISs): *Petri nets* or *Pi calculus*. Examples of PAISs are Workflow Management (WFM), Business Process Management (BPM), Business-to-Business (B2B), Customer Relationship Management (CRM), Enterprise Resource Planning (ERP) systems. Clearly, the web-service paradigm will change the architecture of these systems dramatically. Therefore, triggered by industry standards such as SOAP, WSDL, UDDI, etc., standards are being proposed for orchestrating web services. Examples of such WSCLs are BPEL4WS, BPML, WSFL, WSCI, and XLANG. In the debate on Petri nets versus Pi calculus many players in the “WSCL marketplace” are using demagogic arguments not based

**Petri-Pi** Working Group led by R. Milner and W.M.P van der Aalst started in 2003

# Beginning: Petri-Pi

From: Robin Milner

Date: Wed, February 11, 2004 1:02 pm

Steve

Thanks for that. I believe the pi-calculus team ought to be able to do something with it -- you seem to be taking it in that direction already.

Nobuko, Kohei: I thought we ought to try to model use-cases in pi-calculus, with copious explanations in natural language, aiming at seeing how various concepts like role, transaction, .. would be modelled in pi. I am hoping to try this one when I get time; you might like to try too, and see if we agree!

Robin

# CDL Equivalent

- Basic example:

```
package HelloWorld {
    roleType YouRole, WorldRole;
    participantType You{YouRole}, World{WorldRole};
    relationshipType YouWorldRel between YouRole and WorldRole;
    channelType WorldChannelType with roleType WorldRole;

    choreography Main {
        WorldChannelType worldChannel;

        interaction operation=hello from=YouRole to=WorldRole
            relationship=YouWorldRel channel=worldChannel {
            request messageType=Hello;
        }
    }
}
```

# Scribble Protocol

- *"Scribbling is necessary for architects, either physical or computing, since all great ideas of architectural construction come from that unconscious moment, when you do not realise what it is, when there is no concrete shape, only a whisper which is not a whisper, an image which is not an image, somehow it starts to urge you in your mind, in so small a voice but how persistent it is, at that point you start scribbling" - Kohei Honda 2007*
- **Basic example:**

```
protocol HelloWorld {  
  role You, World;  
  Hello from You to World;  
}
```

# Dialogue between Industry and Academia

Binary Session Types [PARL'94, ESOP'98]



Milner, Honda and Yoshida joined W3C WS-CDL (2002)



Formalisation of W3C WS-CDL [ESOP'07]



Scribble at  $\pi^4$  Technology



Multiparty Session Types [POPL'08]



# Dialogue between Industry and Academia

Binary Session Types [PARL'94, ESOP'98]



Milner, Honda and Yoshida joined W3C WS-CDL (2002)



Formalisation of W3C WS-CDL [ESOP'07]



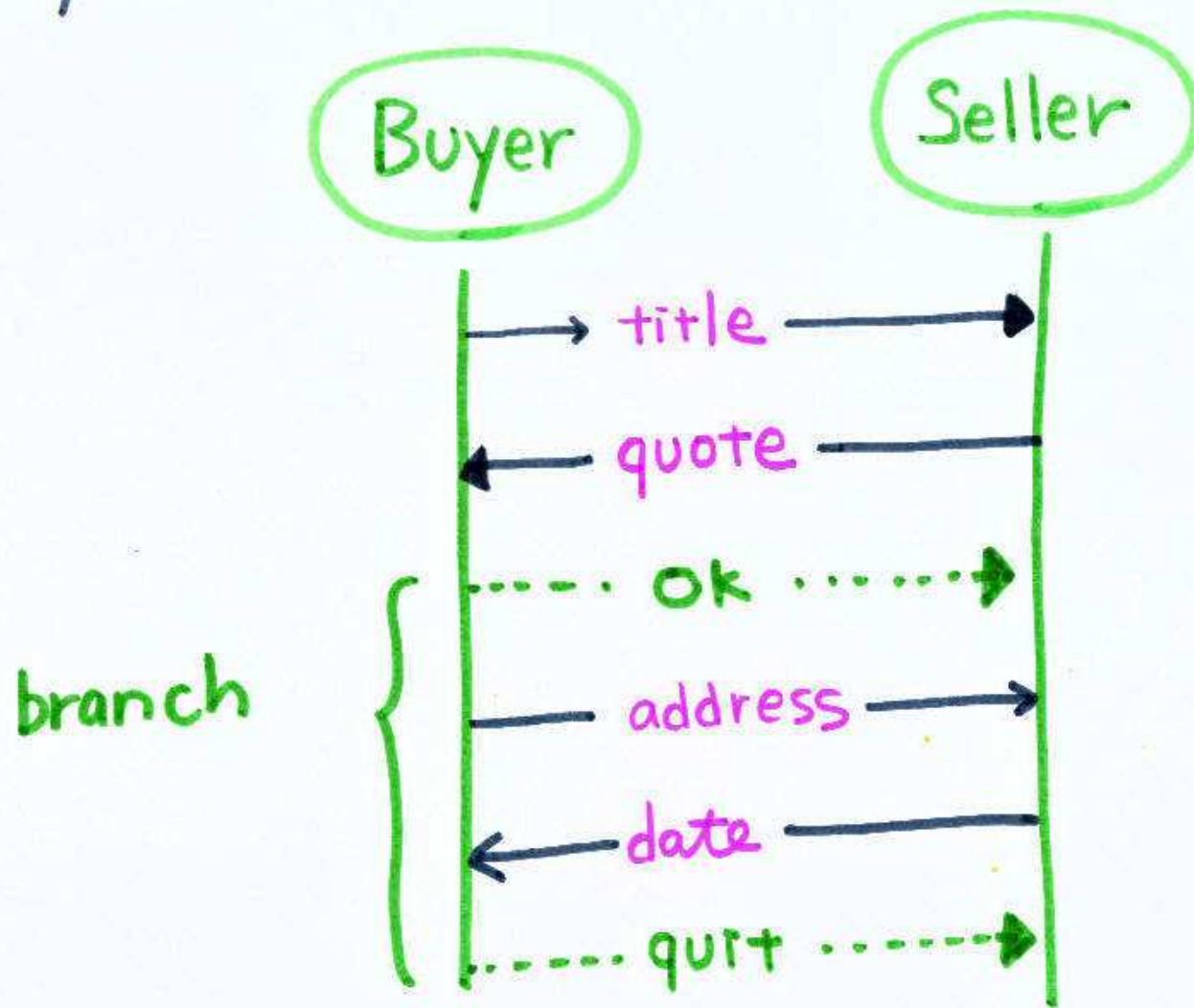
Scribble at  $\pi^4$  Technology

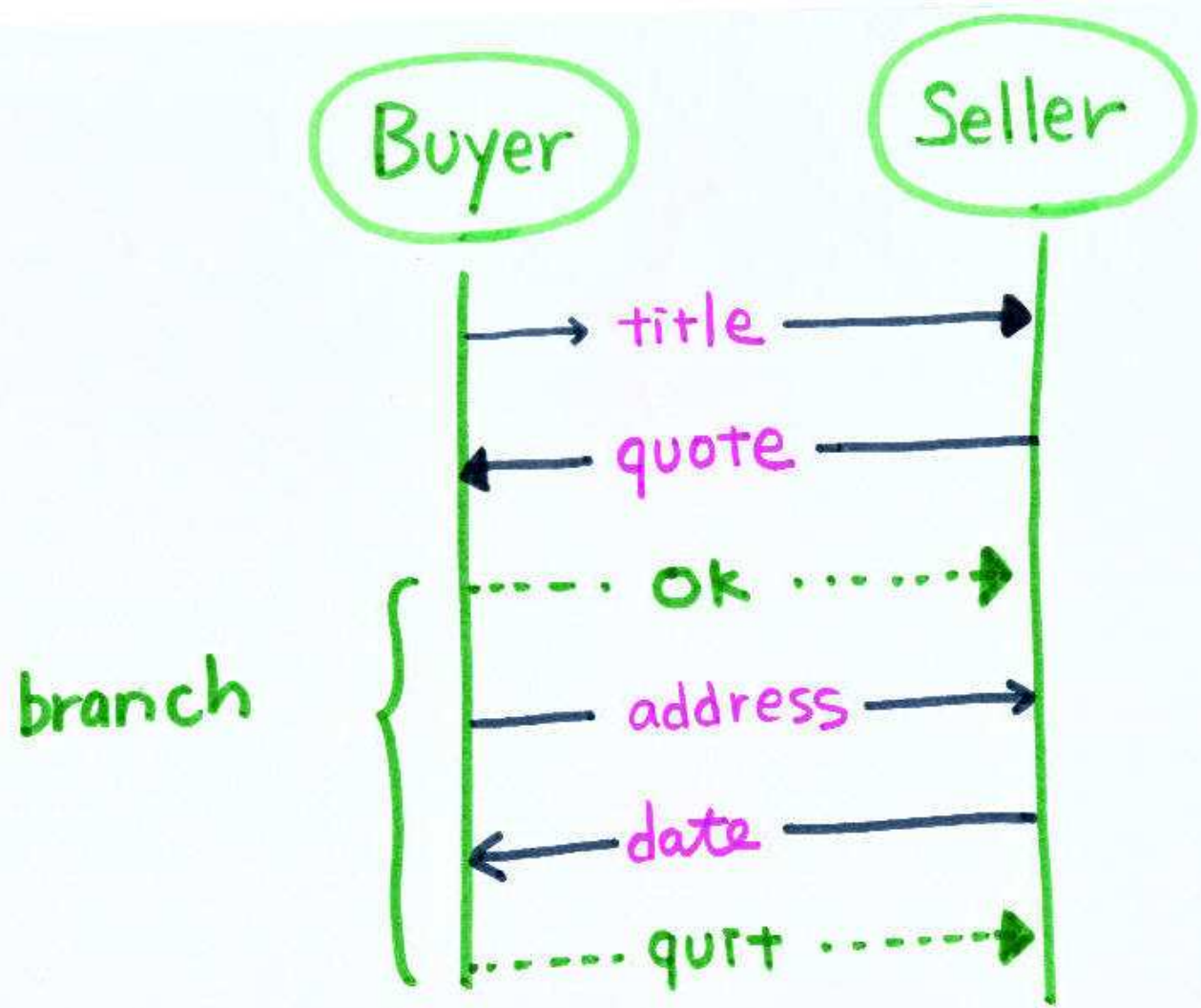


Multiparty Session Types [POPL'08]



# Binary Session Types : Buyer-Seller Protocol





! String ; ? Int ; ⊕ { ok : !String ; ? Date ; end , quit : end }

branch

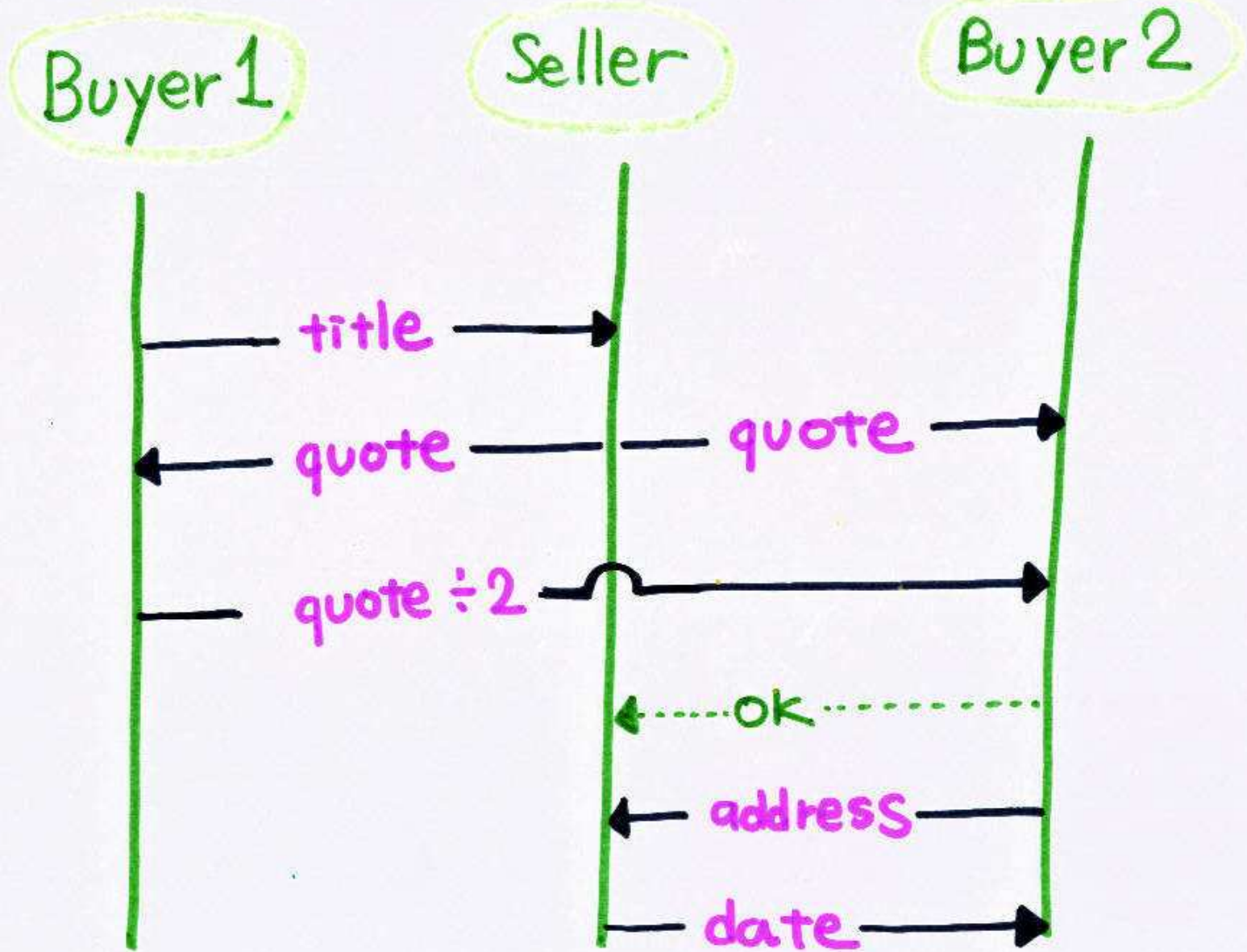


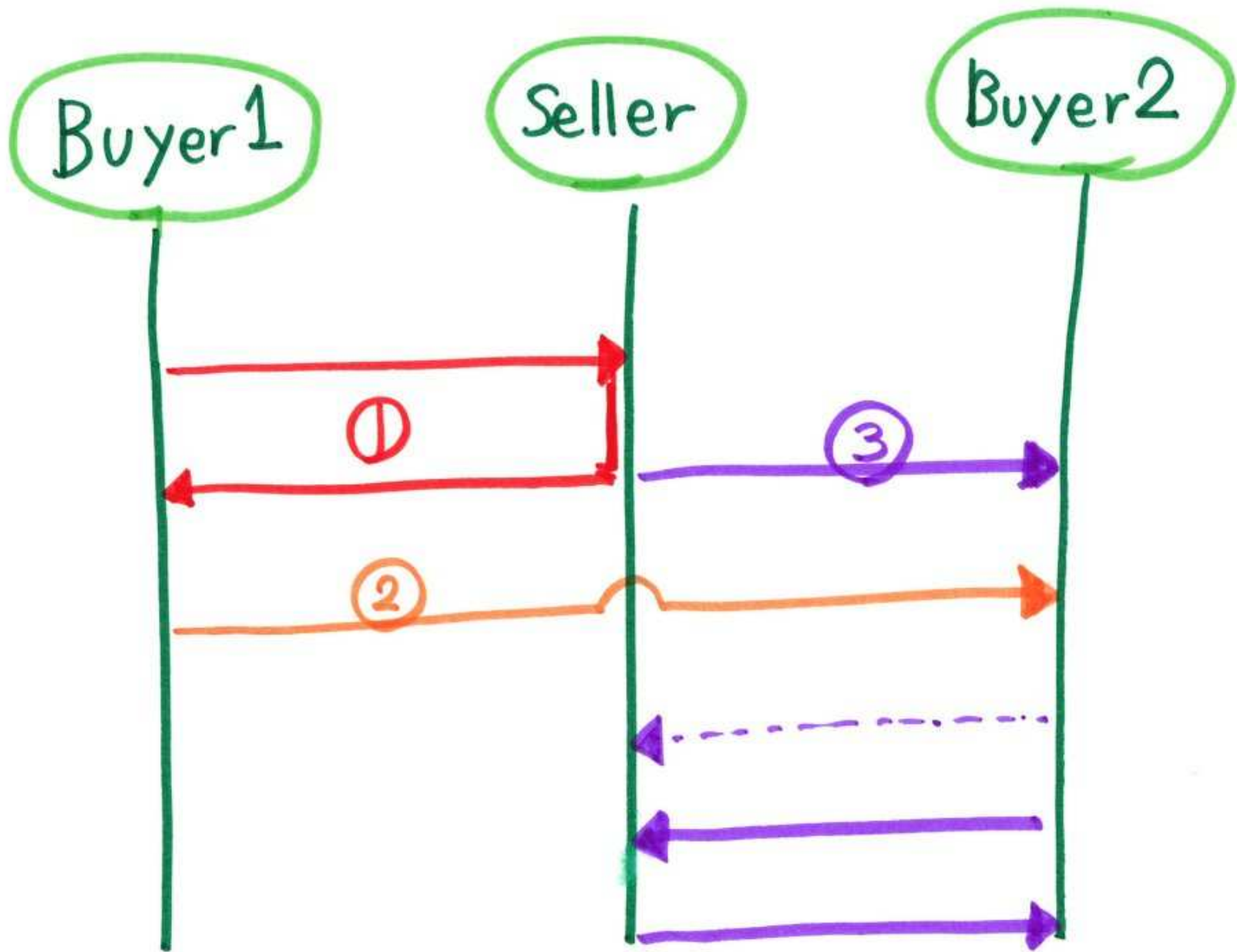
P has T  
Q has T dual  
P | Q typable

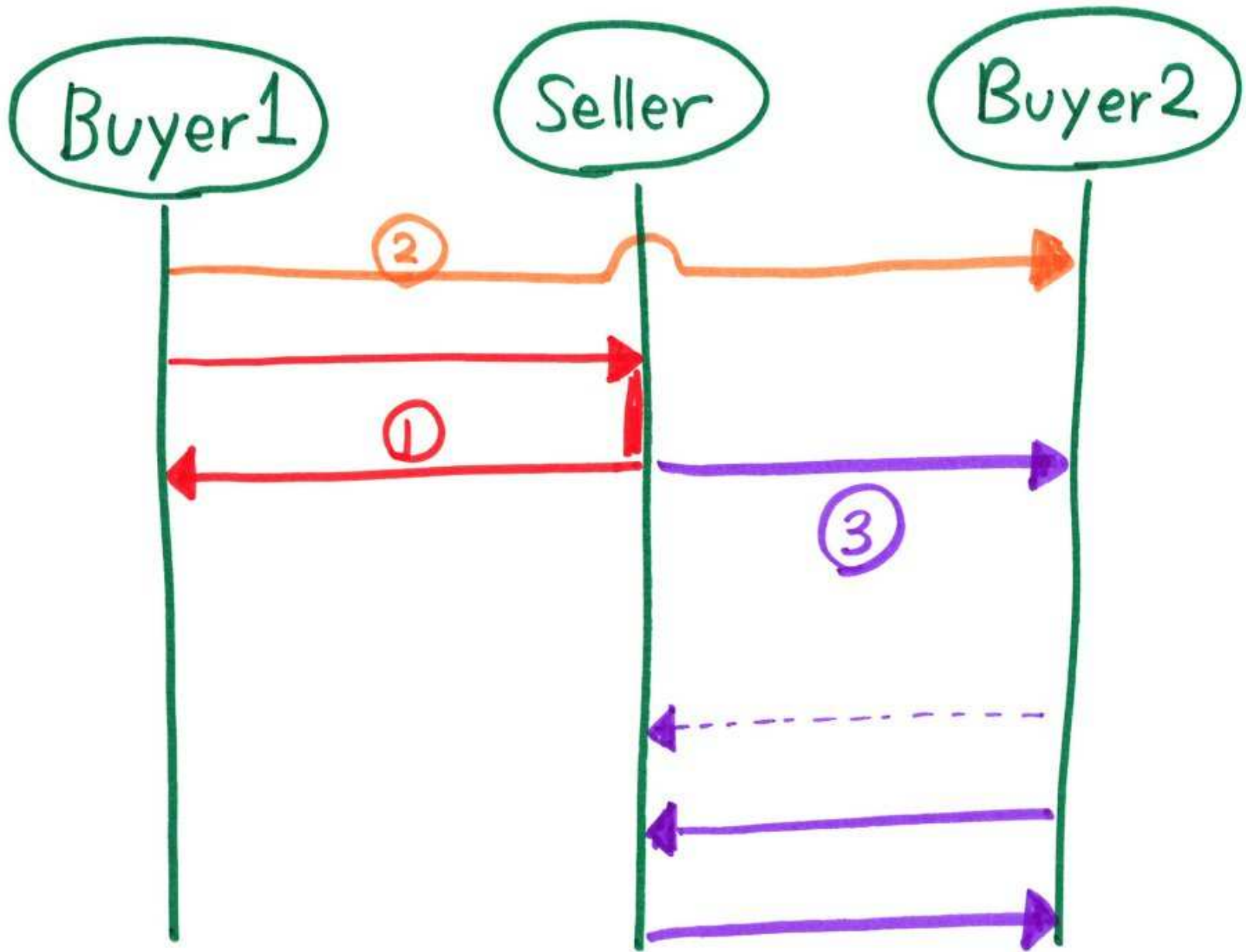
! String ; ? Int ;  $\oplus$  { ok : !String ; ? Date ; end , quit : end }

dual ? String ; ! Int ;  $\otimes$  { ok : ?String ; ! Date ; end , quit : end }

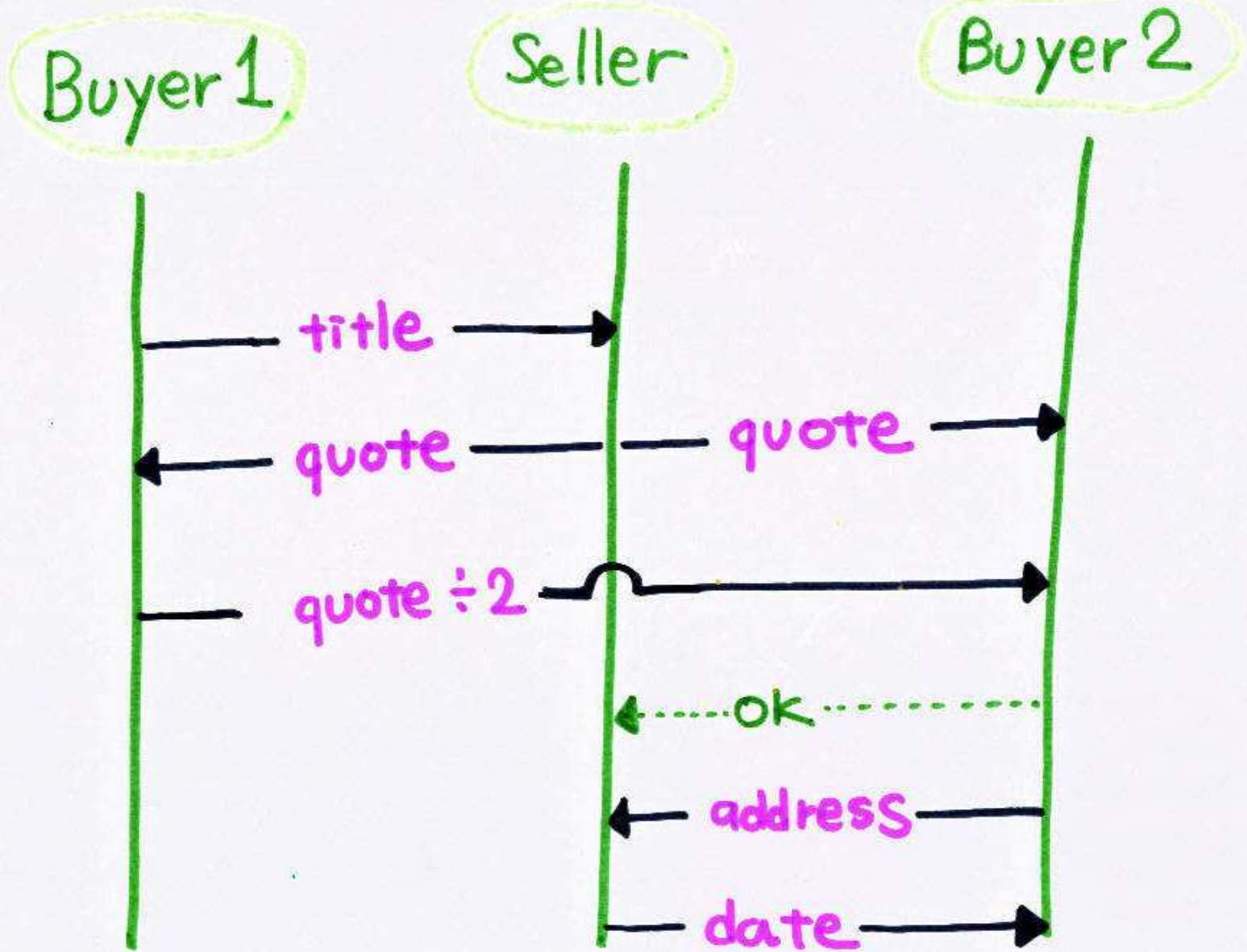
# Multiparty Session Types



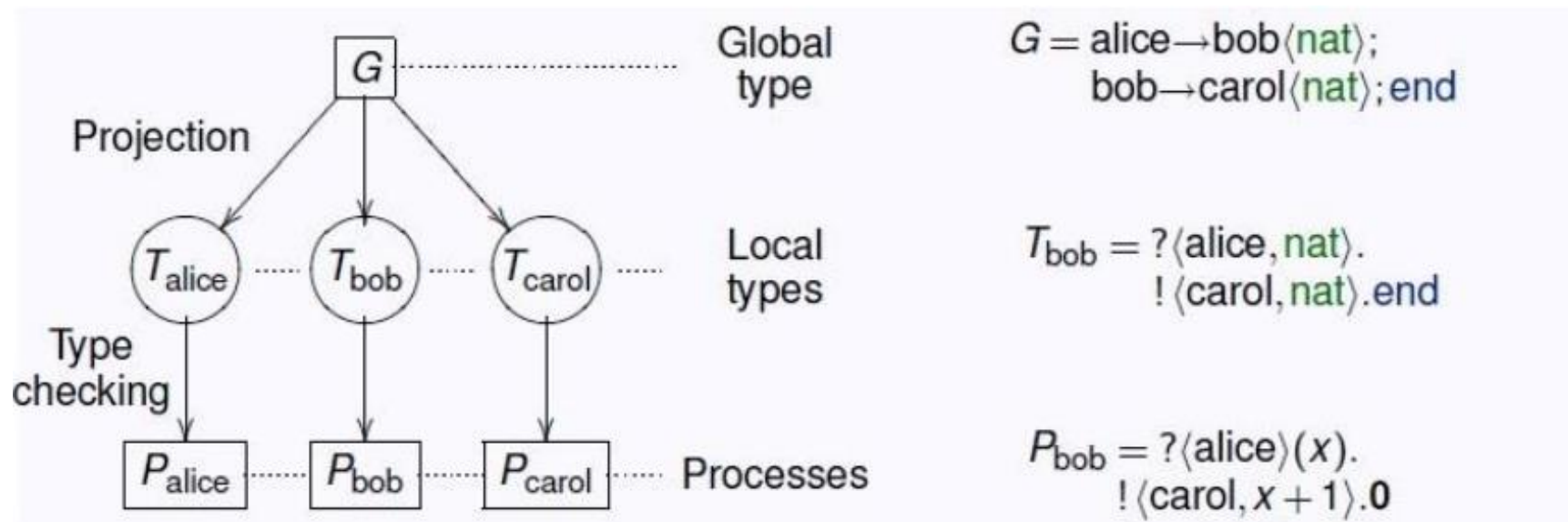




# Multiparty Session Types



# Session Types Overview



## ► Properties

- Communication safety (no communication mismatch)
- Communication fidelity (the communication follow the protocol)
- Progress (no deadlock/stuck in a session)



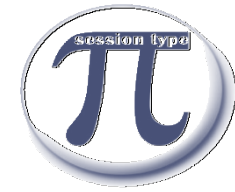












# Evolution Of MPST

---

- ▶ Binary Session Types [THK98, HVK98]



- ▶ Multiparty Session Types [POPL'08]



- ▶ A Theory of Design-by-Contract for Distributed Multiparty Interactions [Concur'11]



- ▶ Multiparty Session Types Meet Communicating Automata [ESOP'12, ICALP'13]



- ▶ Network Monitoring through Multiparty Session Types [FMOODS'13]



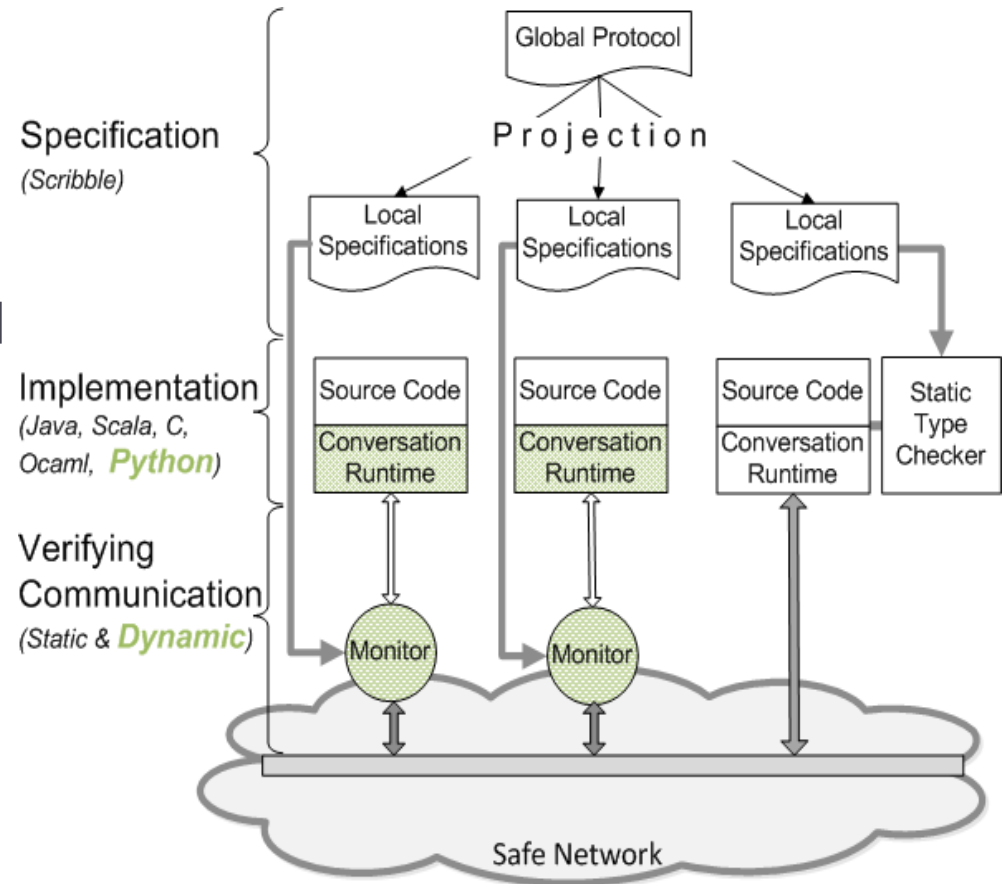
- ▶ SPY: Local Verification of Global Protocols [RV'13]
- ▶ Distributed Runtime Verification with Session Types and Python [RV'13]



# Session Types for Runtime Verification

## ▶ Methodology

- ▶ Developers design protocols in a dedicated language - Scribble
- ▶ Well-formedness is checked by Scribble tools
- ▶ Protocols are projected into local types
- ▶ Local types generate monitors





## What is Scribble?

Scribble is a language to describe application-level protocols among communicating systems. A protocol represents an agreement on how participating systems interact with each other. Without a protocol, it is hard to do meaningful interaction: participants simply cannot communicate effectively, since they do not know when to expect the other parties to send data, or whether the other party is ready to receive data.

However, having a description of a protocol has further benefits. It enables verification to ensure that the protocol can be implemented without resulting in unintended consequences, such as deadlocks.

Find out more ...

[Language Guide](#)[Tools -](#)[Specification](#)[Forum](#)

## An example

```
module examples;

global protocol HelloWorld(role Me, role World) {
  hello(Greetings) from Me to World;
  choice at World {
    hello(GoodMorning) from World to Me;
  } or {
    hello(GoodAfternoon) from World to Me;
  }
}
```

A very simply example, but this illustrates the basic syntax for a hello world interaction, where a party performing the role *Me* sends a message of type *Greetings* to another party performing the role "World", who subsequently makes a decision which determines which path of the choice will be followed, resulting in a *GoodMorning* or *GoodAfternoon* message being exchanged.

### Describe

Scribble is a language for describing multiparty protocols

### Verify

Scribble has a theoretical foundation, based on the Pi Calculus and Session Types, to ensure that protocols

### Project

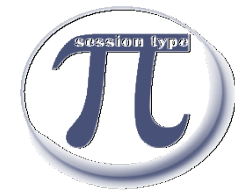
Endpoint projection is the term used for identifying the

### Implement

Various options exist, including (a) using the endpoint projection for a role to generate a skeleton code, (b)

### Monitor

Use the endpoint projection for roles defined within a

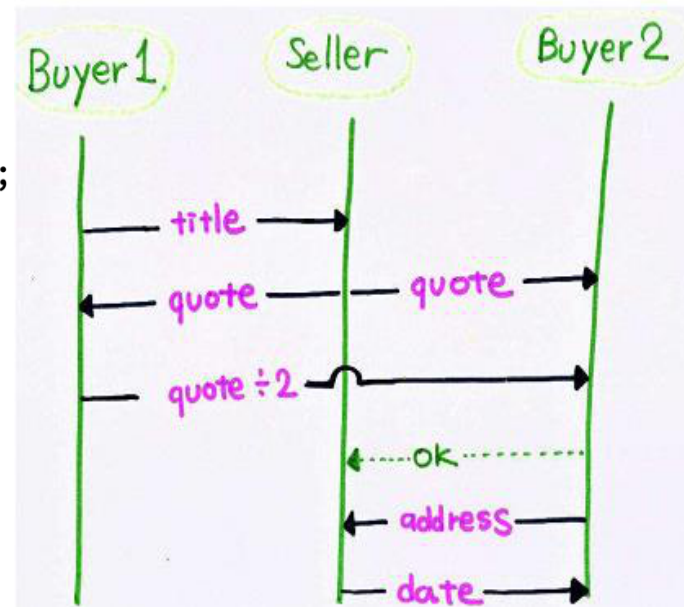


# Two Buyer Protocol in Scribble

```
module Bookstore;

type <java> "java.lang.Integer" from "rt.jar" as Integer;
type <java> "java.lang.String" from "rt.jar" as String;

global protocol TwoBuyers(role A, role B, role S) {
  title(String) from A to S;
  quote(Integer) from S to A, B;
  rec LOOP {
    share(Integer) from A to B;
    choice at B {
      accept(address:String) from B to A, S;
      date(String) from S to B;
    } or {
      retry() from B to A, S;
      continue LOOP;
    } or {
      quit() from B to A, S;
    }
  }
}
```





# Buyer: A local projection

---

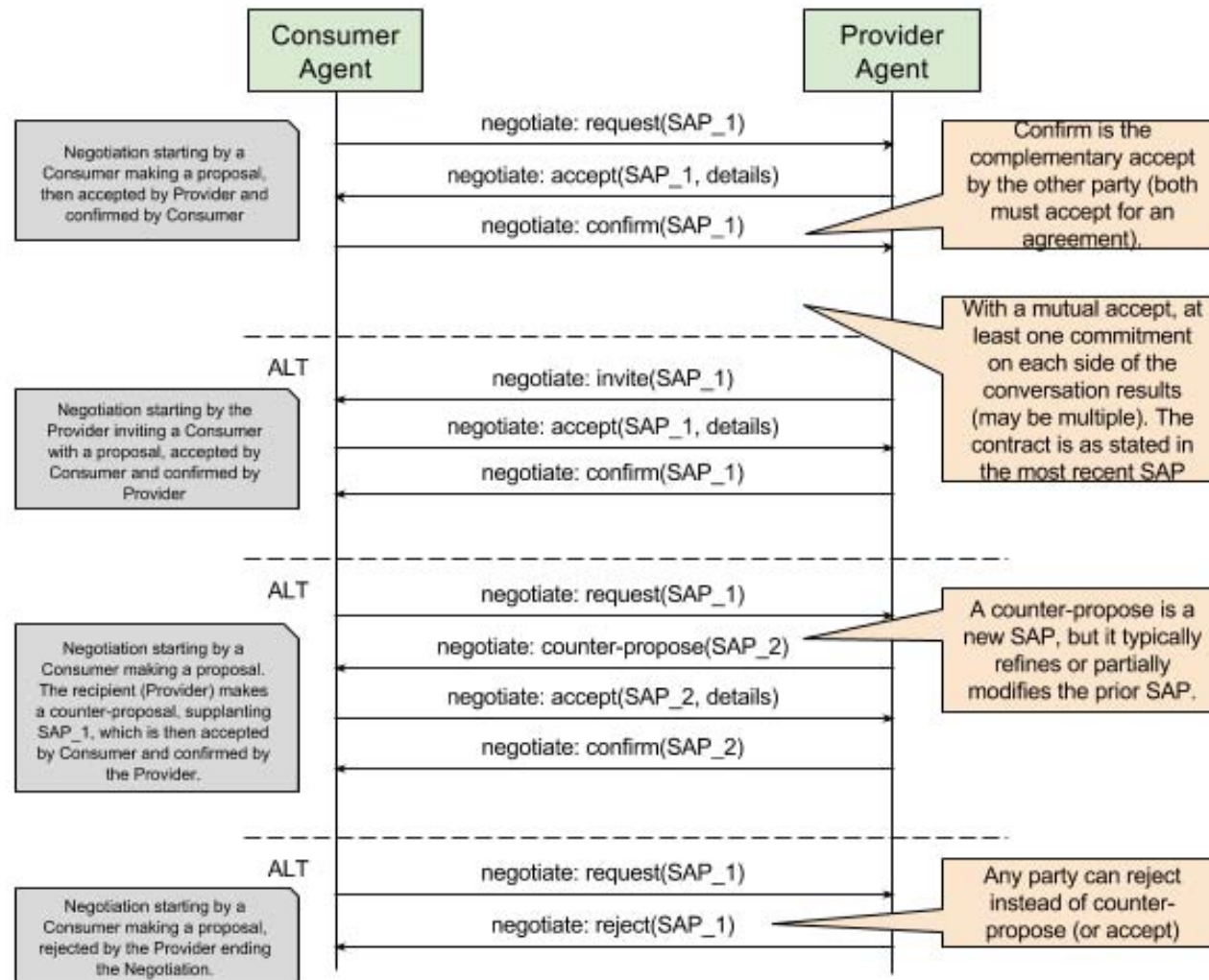
```
module Bookstore_TwoBuyers_A;

type <java> "java.lang.Integer" from "rt.jar" as Integer;
type <java> "java.lang.String" from "rt.jar" as String;

local protocol TwoBuyers_A at A(role A, role B, role S) {
  title(String) to S;
  quote(Integer) from S;
  rec LOOP {
    share(Integer) to B;
    choice at B {
      accept(address:String) from B;
    } or {
      retry() from B;
      continue LOOP;
    } or {
      quit() from B;
    }
  }
}
```



# OOI agent negotiation 1/5

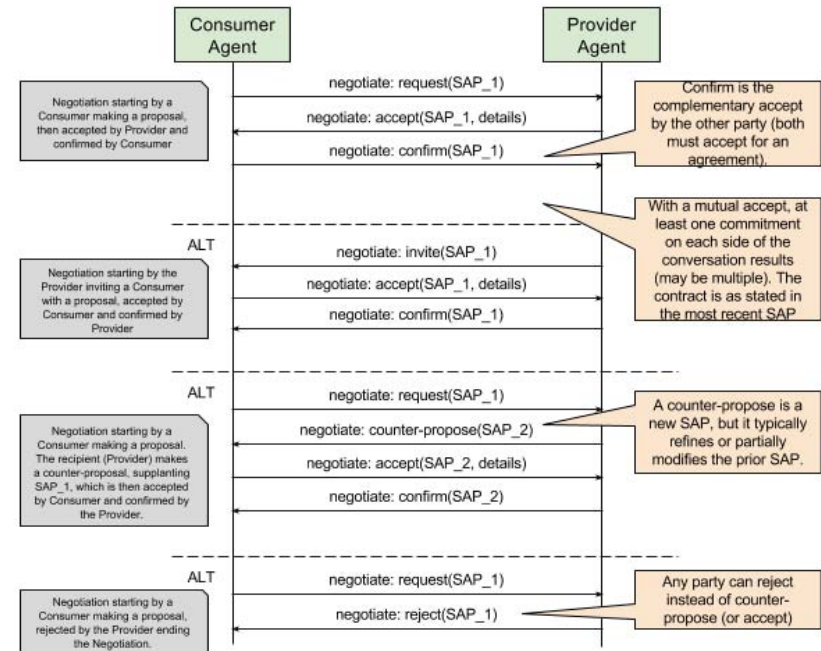


- ▶ <https://confluence.oceanobservatories.org/display/syseng/CIAD+COI+OV+Negotiate+Protocol>

# OOI agent negotiation 2/5

```
type <yml> "SAPDoc1" from "SAPDoc1.yml" as SAP;
```

```
global protocol Negotiate(role Consumer as C, role Producer as P) {
```



```
}
```

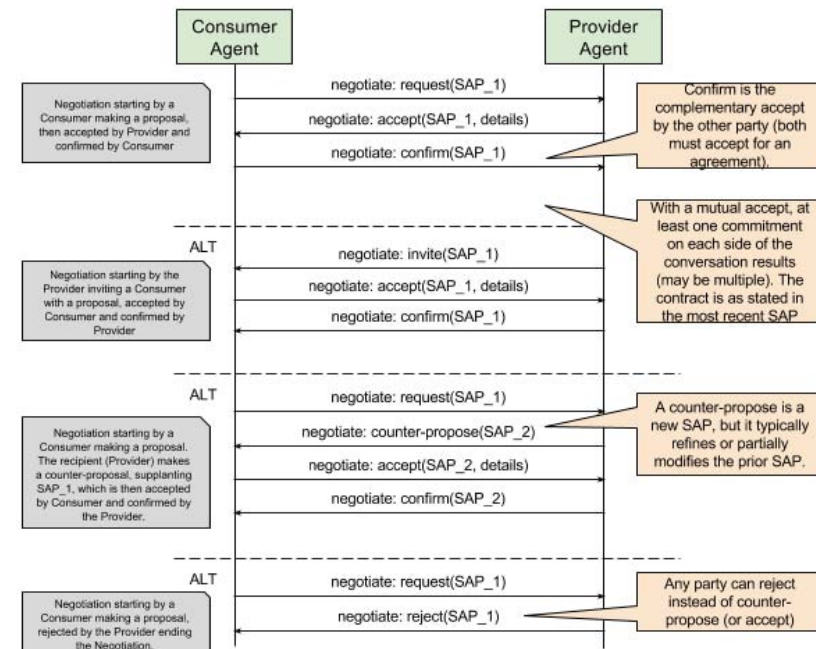
# OOI agent negotiation 3/5 (choice)

```
type <yml> "SAPDoc1" from "SAPDoc1.yml" as SAP;
```

```
global protocol Negotiate(role Consumer as C, role Producer as P) {  
  propose(SAP) from C to P;
```

```
  choice at P {  
    accept() from P to C;  
    confirm() from C to P;  
  } or {  
    reject() from P to C;  
  } or {  
    propose(SAP) from P to C;
```

```
  } }  
}
```

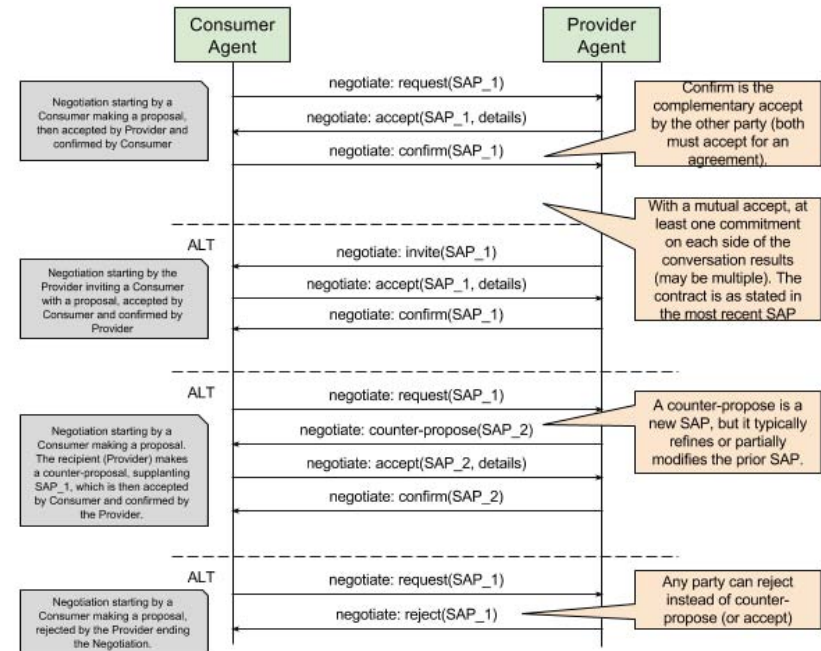


# OOI agent negotiation 4/5

```
type <yml> "SAPDoc1" from "SAPDoc1.yml" as SAP;
```

```
global protocol Negotiate(role Consumer as C, role Producer as P) {
  propose(SAP) from C to P;
```

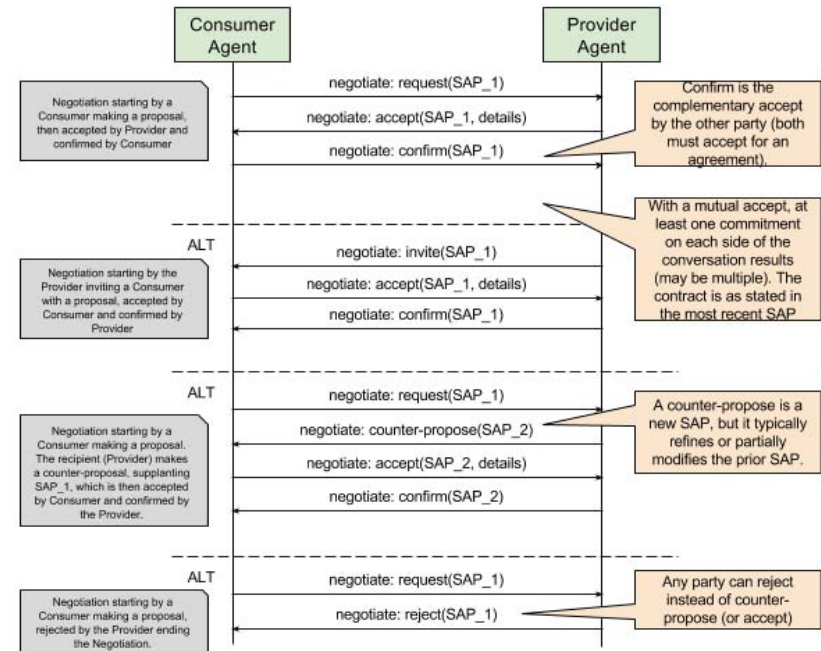
```
  choice at P {
    accept() from P to C;
    confirm() from C to P;
  } or {
    reject() from P to C;
  } or {
    propose(SAP) from P to C;
    choice at C {
      accept() from C to P;
      confirm() from P to C;
    } or {
      reject() from C to P;
    } or {
      propose(SAP) from C to P;
    }
  }
}
```



# OOI agent negotiation 5/5 (recursion)

```
type <yml> "SAPDoc1" from "SAPDoc1.yml" as SAP;
```

```
global protocol Negotiate(role Consumer as C, role Producer as P) {
  propose(SAP) from C to P;
  rec X {
    choice at P {
      accept() from P to C;
      confirm() from C to P;
    } or {
      reject() from P to C;
    } or {
      propose(SAP) from P to C;
      choice at C {
        accept() from C to P;
        confirm() from P to C;
      } or {
        reject() from C to P;
      } or {
        propose(SAP) from C to P;
        continue X;
      }
    }
  }
}
```

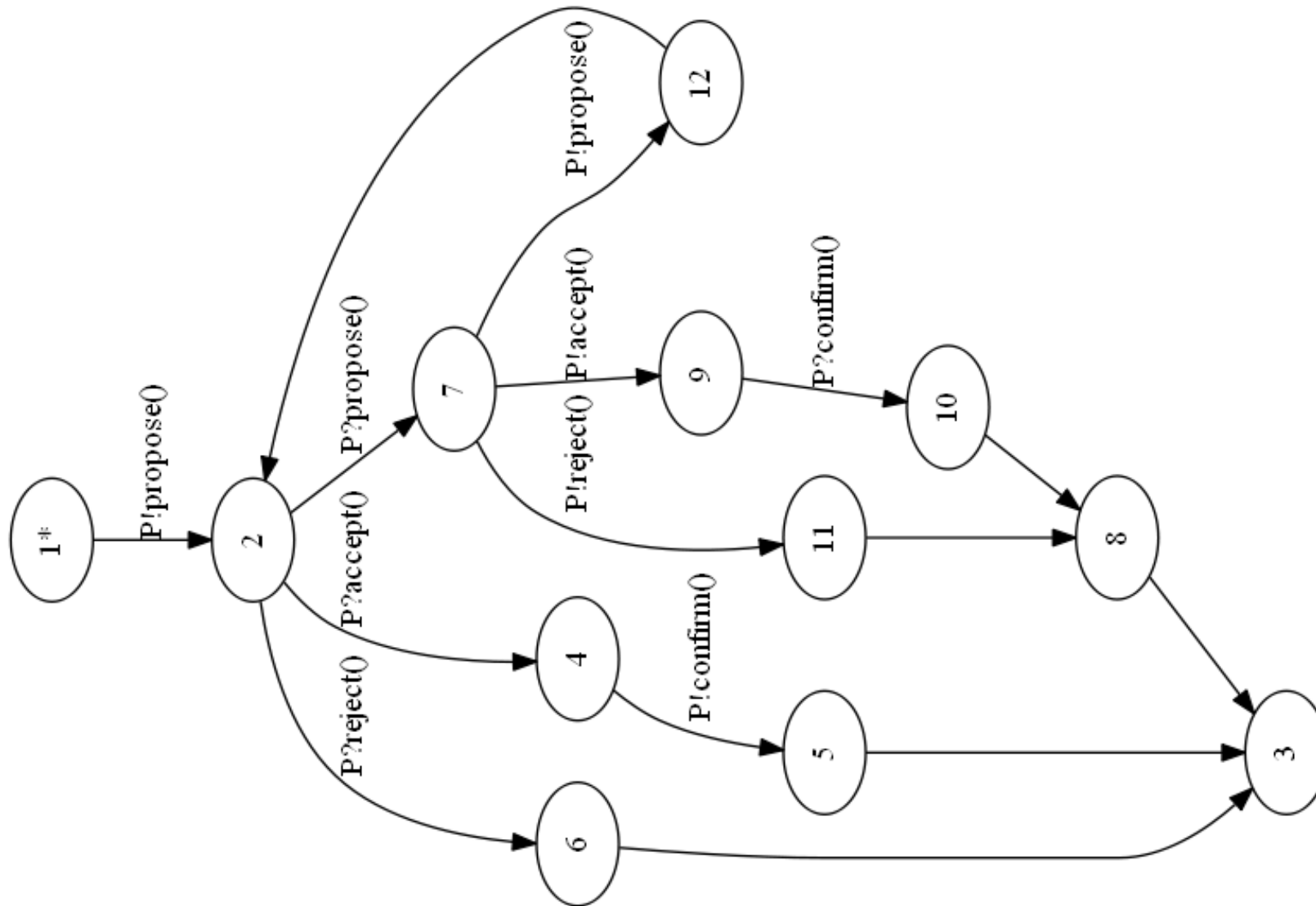


# Local protocol projection (Negotiation Consumer)

```
// Global
propose(SAP) from C to P;
rec START {
  choice at P {
    accept() from P to C;
    confirm() from C to P;
  } or {
    reject() from P to C;
  } or {
    propose(SAP) from P to C;
    choice at C {
      accept() from C to P;
      confirm() from P to C;
    } or {
      reject() from C to P;
    } or {
      propose(SAP) from C to P;
      continue START;
    }
  }
}
```

```
// Projection for Consumer
propose(SAP) to P;
rec START {
  choice at P {
    accept() from P;
    confirm() to P;
  } or {
    reject() from P;
  } or {
    propose(SAP) from P;
    choice at C {
      accept() to P;
      confirm() from P;
    } or {
      reject() to P;
    } or {
      propose(SAP) to P;
      continue START;
    }
  }
}
```

# FSM generation (Negotiation Consumer)





# Scribble Community

---

- ▶ **Webpage:**

- ▶ [www.scribble.org](http://www.scribble.org)

- ▶ **GitHub:**

- ▶ <https://github.com/scribble>

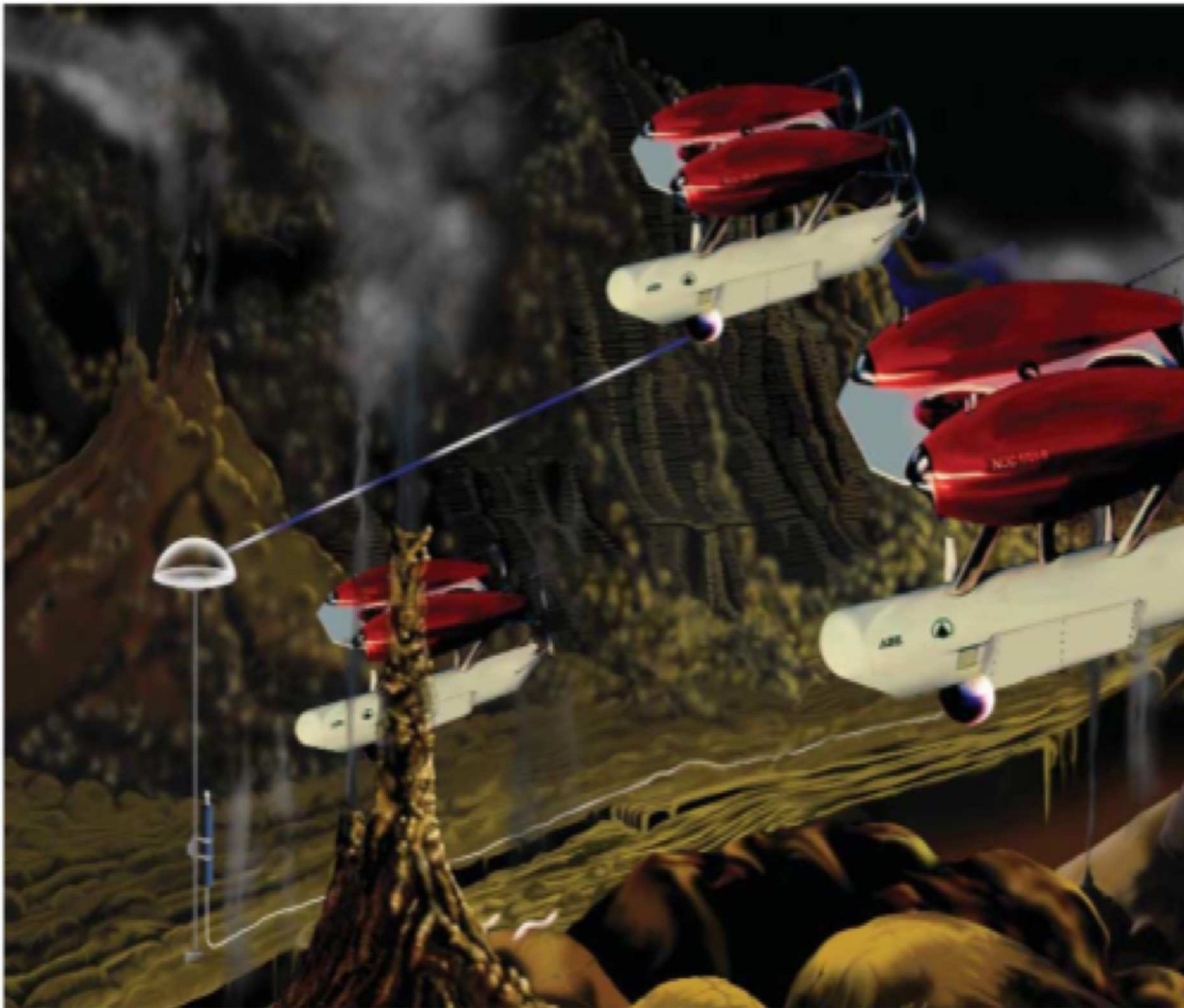
- ▶ **Tutorial:**

- ▶ [www.doc.ic.ac.uk/~rhu/scribble/tutorial.html](http://www.doc.ic.ac.uk/~rhu/scribble/tutorial.html)

- ▶ **Specification (0.3)**

- ▶ [www.doc.ic.ac.uk/~rhu/scribble/langref.html](http://www.doc.ic.ac.uk/~rhu/scribble/langref.html)





**Figure 5: A coordinated set of autonomous underwater vehicles**



**Figure 3: Observatory comprised of ships, aircraft and autonomous vehicles linked to assimilation modeling capabilities on shore**



SEARCH

RESOURCES

- All Resources
- Data Products
- Observatories
- Platforms
- Instruments

Welcome to Release 2 of the Ocean Observatories Initiative Observatory (OOI). You already have access to many OOI features and real-time data. Just click on something that looks interesting on this page to start using the OOI as our Guest.

For personalized services, such as setting up notifications and preserving settings for your next visit, create a free account by clicking on "Create Account" at the top of the page.



National Science Foundation working with Consortium for Ocean Leadership

Funding for the Ocean Observatories Initiative is provided by the National Science Foundation through a Cooperative Agreement with the Consortium for Ocean Leadership. The OOI Program Implementing Organizations are funded through sub-awards from the Consortium for Ocean Leadership.

Location

CURRENT LOCATION

FILTER



DATA LEGEND

- Temperature
- Salinity
- Oxygen
- Density
- Currents
- Sea Surface Height (SSH)
- Chlorophyll
- Turbidity
- pH
- Seismology
- Other

REQUENCY

- 1 Hour
- 2 hours
- 3 hours
- 5 hours
- 8 hours
- 12 hours
- 18 hours
- 24 hours
- 48 Hours
- 72 Hours

RECENT UPDATES

NAME	DATE	TYPE	EVENT	DESCRIPTION	NOTE
01 m Oregon Coast North Salinity	2012-01-10 23:55:55	Type	Event	Description goes here	Note goes here
01 m California South 100m pH	2012-01-10 23:55:55	Type	Event	Description goes here	Note goes here
01 m California South salinity	2012-01-10 23:55:55	Type	Event	Description goes here	Note goes here
03 m Oregon North Turbidity	2012-01-10 23:55:55	Type	Event	Description goes here	Note goes here
05 m Oregon South Temperature	2012-01-10 23:55:55	Type	Event	Description goes here	Note goes here
20 m Oregon Coast Currents	2012-01-10 23:55:55	Type	Event	Description goes here	Note goes here
01 h California South Seismology	2012-01-10 23:55:55	Type	Event	Description goes here	Note goes here
01 h Oregon Coast South 1000m Ox	2012-01-10 23:55:55	Type	Event	Description goes here	Note goes here
02 h California Coast Seismology	2012-01-10 23:55:55	Type	Event	Description goes here	Note goes here
04 h California North Seismology	2012-01-10 23:55:55	Type	Event	Description goes here	Note goes here

Dashboard

RECENT IMAGES

- Glider**  
Last Modified: 2011-06-15  
Last Viewed: 2011-12-15  
Last Updated: 2011-12-30, 13.24
- Gorgonian Coral**  
Last Modified: 2011-06-15  
Last Viewed: 2011-12-15  
Last Updated: 2011-12-30, 13.24
- Acoustic Release**  
Last Modified: 2011-06-15  
Last Viewed: 2011-12-15  
Last Updated: 2011-12-30, 13.24

POPULAR RESOURCES

- SeaBird CDT**  
Last Modified: 2011-06-15  
Last Viewed: 2011-12-15  
Last Updated: 2011-12-30, 13.24
- Marine caption**  
Last Modified: 2011-06-15  
Last Viewed: 2011-12-15  
Last Updated: 2011-12-30, 13.24
- Surface Buoy**  
Last Modified: 2011-06-15  
Last Viewed: 2011-12-15  
Last Updated: 2011-12-30, 13.24

UNUSUAL EVENTS

- Oregon Coast Wave Height**  
Last Modified: 2011-06-15  
Last Viewed: 2011-12-15  
Last Updated: 2011-12-30, 13.24
- Water Surface Elevation**  
Last Modified: 2011-06-15  
Last Viewed: 2011-12-15  
Last Updated: 2011-12-30, 13.24

# Language and Implementations

- ▶ **Carrying out large-scale experiences** with OOI, Pivotal, Red Hat, Cognizant, UNIFI, TrustCare
- JBoss **SCRIBBLE** [ICDCIT'10, COB'12] and **SAVARA** projects
- ▶ **High-performance computing**  
Session Java [ECOOP'08, ECOOP'10, Coordination'11]  
⇒ Session C & MPI [TOOLS'12][Hearts'12][EuroMPI'12][PDP'14]
- ▶ **Multiparty session languages** Ocaml, Java, C, Python, Scala, Jolie
  - Trustworthy Pervasive Healthcare Services via Multiparty Session Types [FHIES'12]
  - Practical interruptible conversations: Distributed dynamic verification with session types and Python [RV'13]
  - Multiparty Session Actors [Coordination'14]

<http://www.zdlc.co/faq/>

---

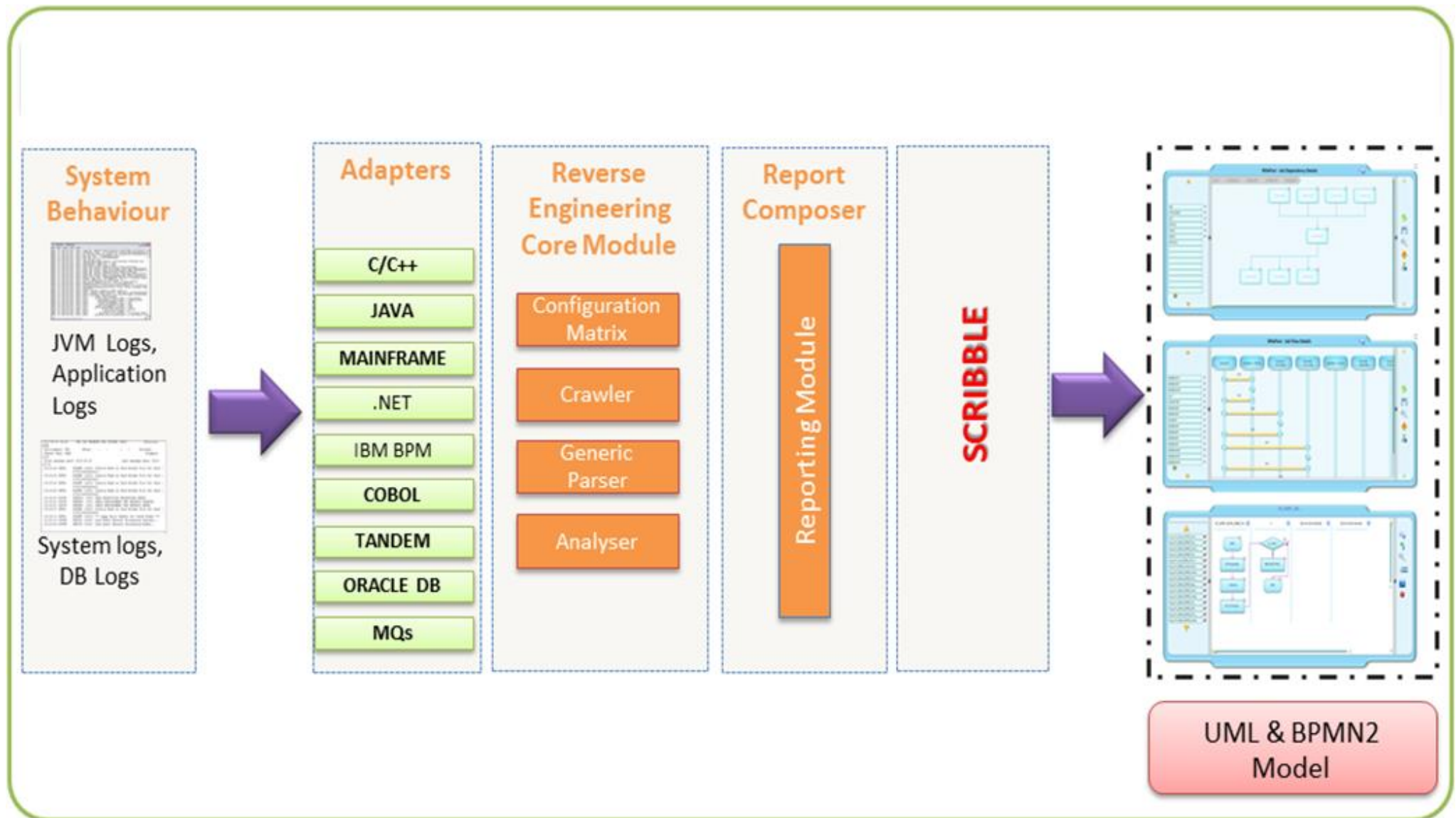


[Home](#) [ZDLC Solutions](#) [FAQ](#) [Resources](#) [Events](#) [Blog](#) [Contact](#) [Partners](#) [Cognizant](#)

---

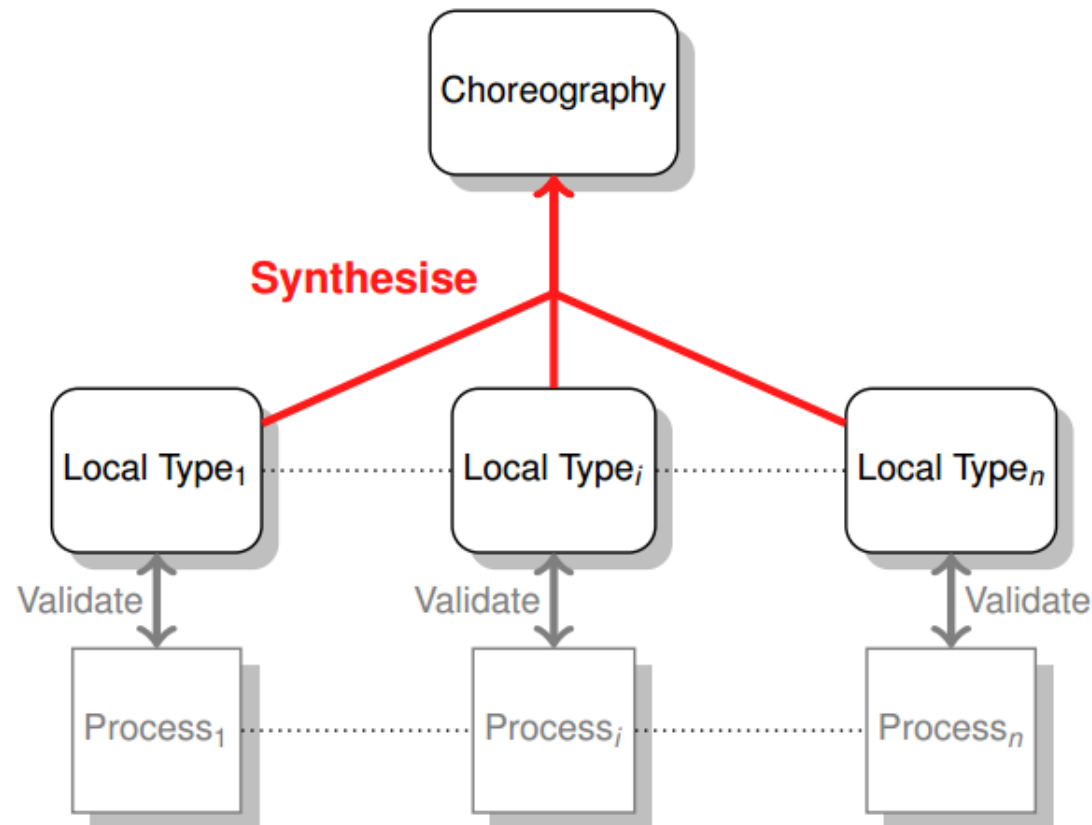


# Zero Deviation Life Cycle Platform



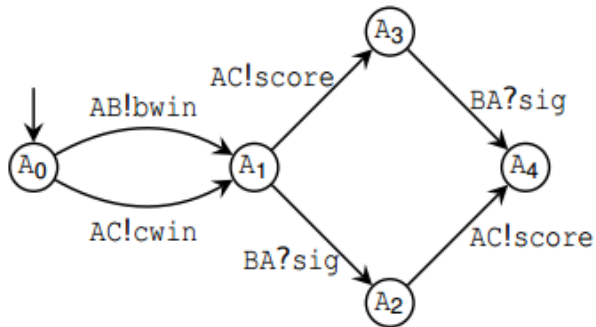
# Synthesis of Graphical Choreographies 1/2

---

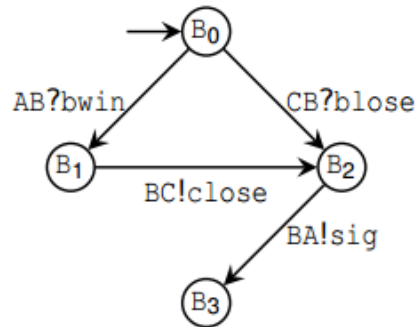


- Multiparty Session Types top-down approach (cf. POPL'08 & ESOP'12)
- Not applicable without *a priori knowledge* of a choreography
- Synthesise a choreography from a set of local specifications
- Concretely: from *Communicating Finite-State Machines* to *Global Graphs*

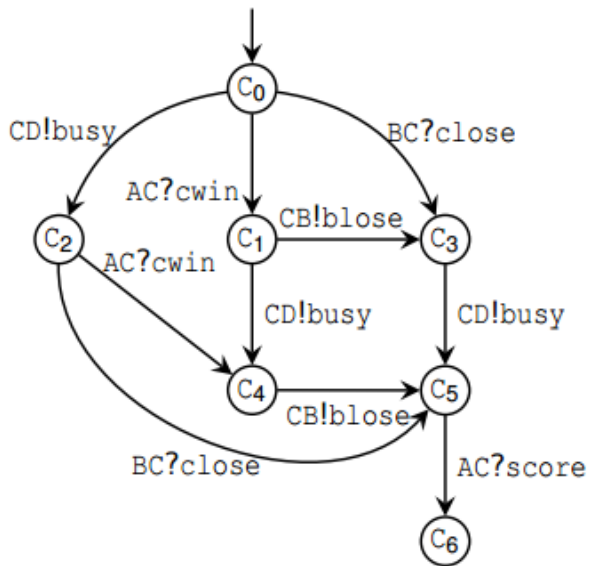
# Synthesis of Graphical Choreographies 2/2



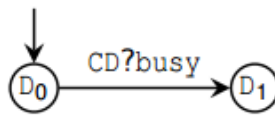
Alice



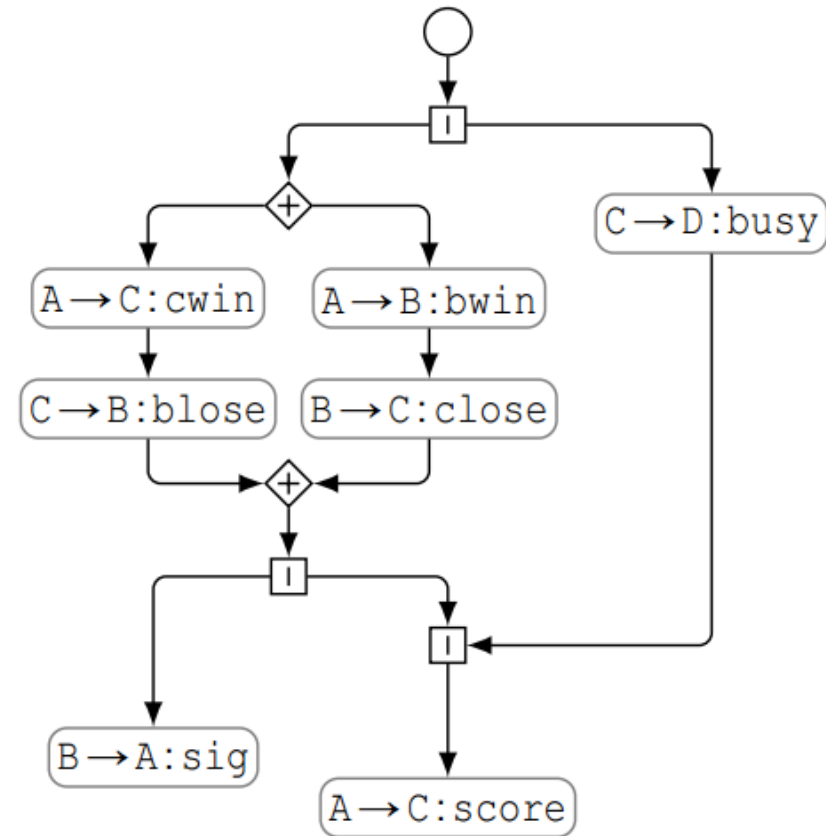
Bob



Carol

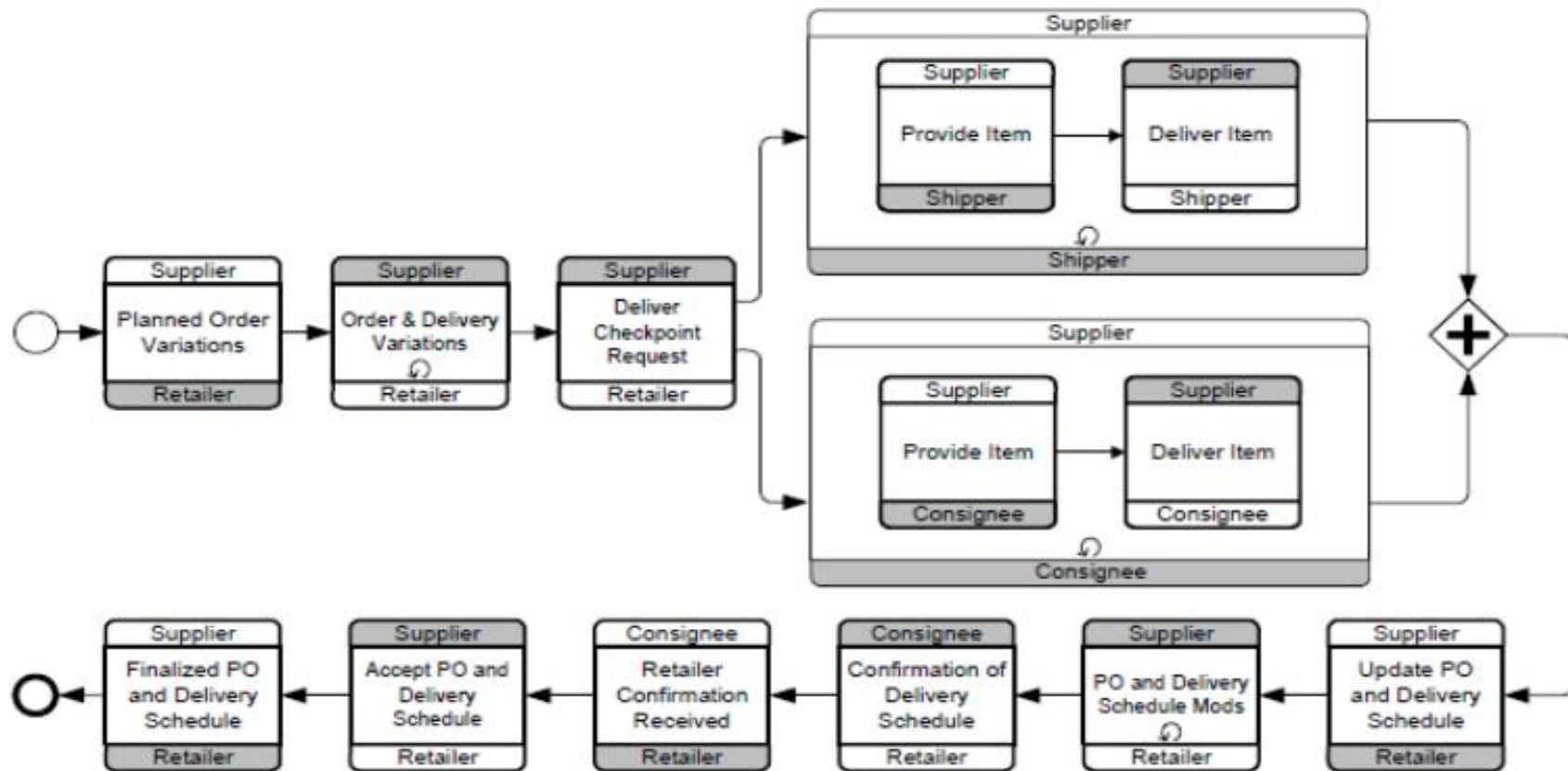


Dave



# Session Nets 1/2

Graphical global specification based on Petri Nets that cannot be directly represented in the MPST linear syntax



▶ An application of the Petri Nets token dynamics to a conformance validation



# Proposals

- Building and understanding computations and interactions
- Link with other disciplines
- Look at use cases and examples
- Talk with, guide and motivate industry people
- Strong confidence: theories should be useful
- Don't hesitate to waste your time

## A rare cluster of qualities

From the team of OOI CI:

*Kohei has lead us deep into the nature of communication and processing. His esthetics, precision and enthusiasm for our mutual pursuit of formal Session (Conversation) Types and specifically for our OOI collaboration to realize this vision in very concrete terms were, as penned by Henry James, lessons in seeing the nuances of both beauty and craft, through a rare cluster of qualities - curiosity, patience and perception; all at the perfect pitch of passion and expression.*