Scribble, Runtime Verification and Multiparty Session Types

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

Nobuko Yoshida
Imperial College London

In collaboration with:

Matthew Arrott (OOI)

Gary Brown (Red Hat)

Stephen Henrie (OOI)

Bippin Makoond (Cognizant/Qualit-e)

Michael Meisinger (OOI)

Matthew Rawlings (ISOTC68/USB)

Alexis Richardson (RabbitMQ/Pivotal)

Steve Ross-Talbot (Cognizant/Qualit-e)

and all our academic colleagues

Laura Bocchi, Tzu-Chun Chen, Tiago Cogumbreiro, Romain Demangeon, Pierre-Malo Denielóu, Juliana Franco, Luca Fossati, Dimitrios Kouzapas, Julien Lange, Rumyana Neykova, Nicholas Ng, Weizhen Yang

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

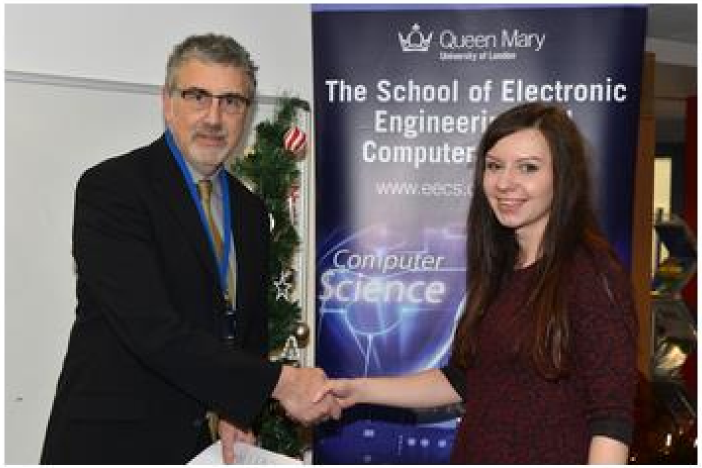
Posted with permission from QMUL on 17th Dec 2013. Original article written by Edmund Robinson.

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

Outline

- Background
- Multiparty Session Types
- Scribble and Applications to a Large-scale Cyberinfrastructure
- Recent Works

Communication is Ubiquitous

- Internet, the WWW, Cloud Computing, the next-generation manycore chips, message-passing parallel computations, large-scale cyberinfrastructure for e-Science.
- The way to organise software is increasingly based on communications.
- Applications need *structured* series of communications.

- Question
  - How to formally abstract/specify/implement/control communications?

Communication is Ubiquitous

- Internet, the WWW, Cloud Computing, the next-generation manycore chips, message-passing parallel computations, large-scale cyberinfrastructure for e-Science.
- The way to organise software is increasingly based on communications.
- Applications need *structured* series of communications.

- Question
  - How to formally abstract/specify/implement/control communications?

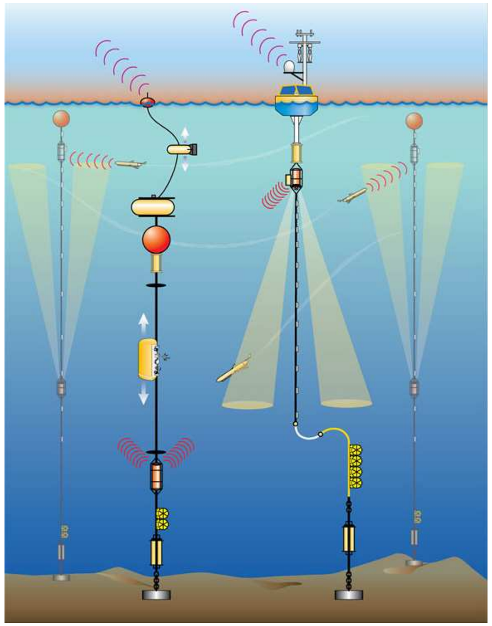
Communication is Ubiquitous

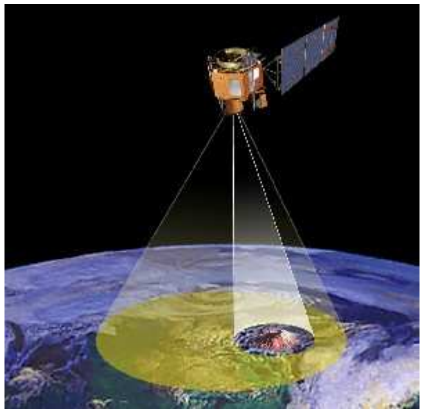
- Internet, the WWW, Cloud Computing, the next-generation manycore chips, message-passing parallel computations, large-scale cyberinfrastructure for e-Science .
- The way to organise software is increasingly based on communications.
- Applications need *structured* series of communications.

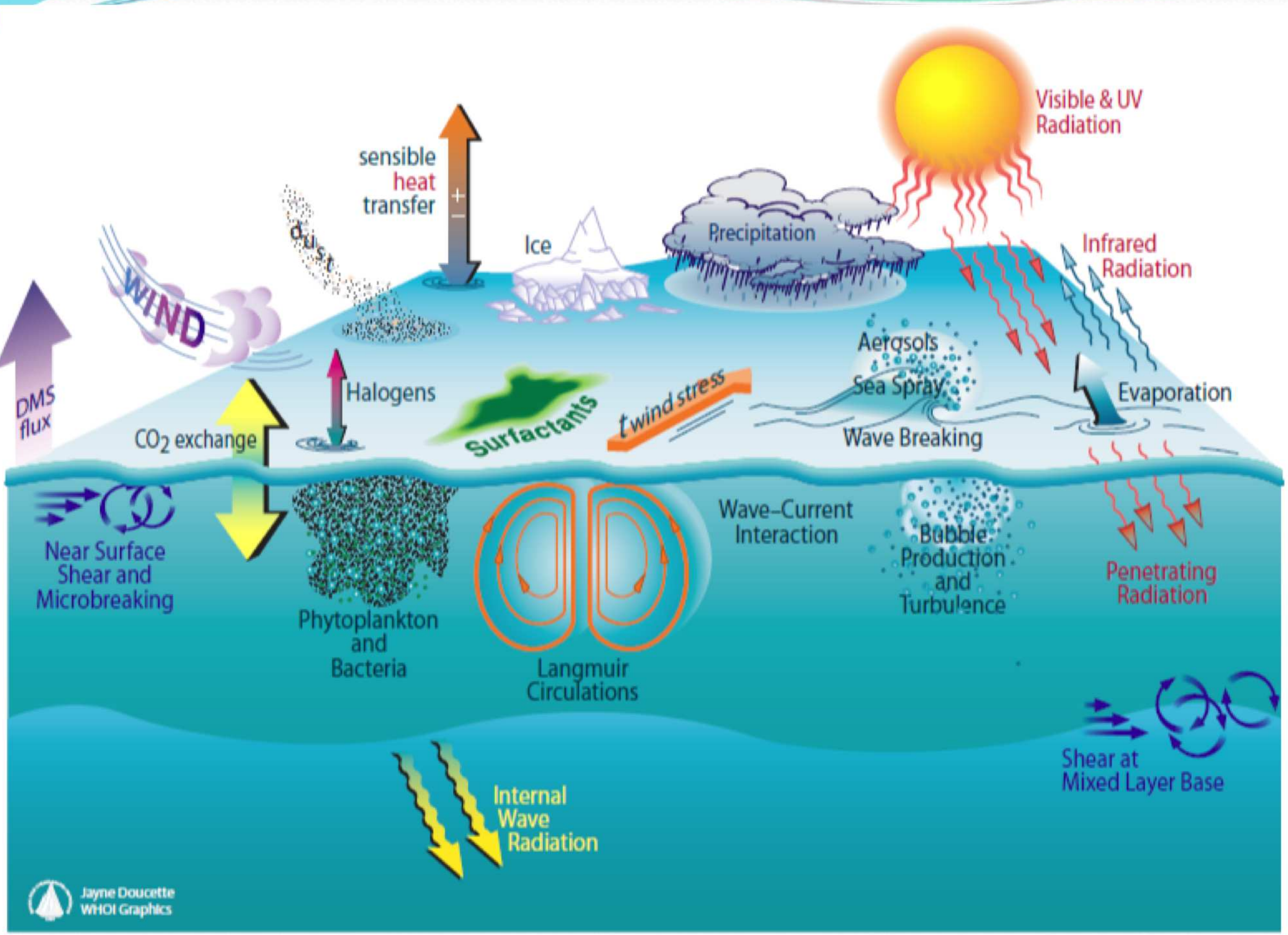
- Question $\Longrightarrow$ Multiparty session type theory
  - How to formally abstract/specify/implement/control communications?

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.

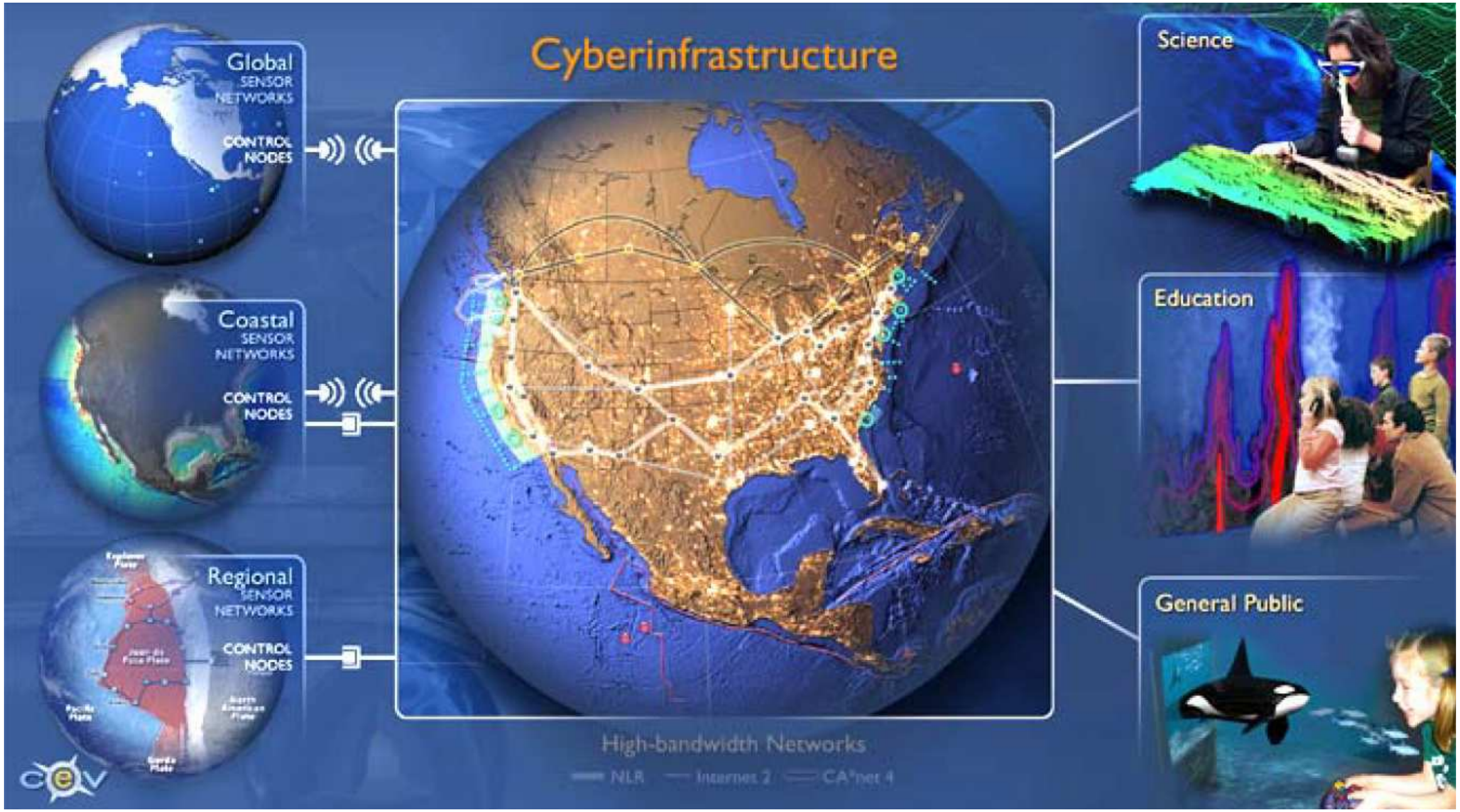


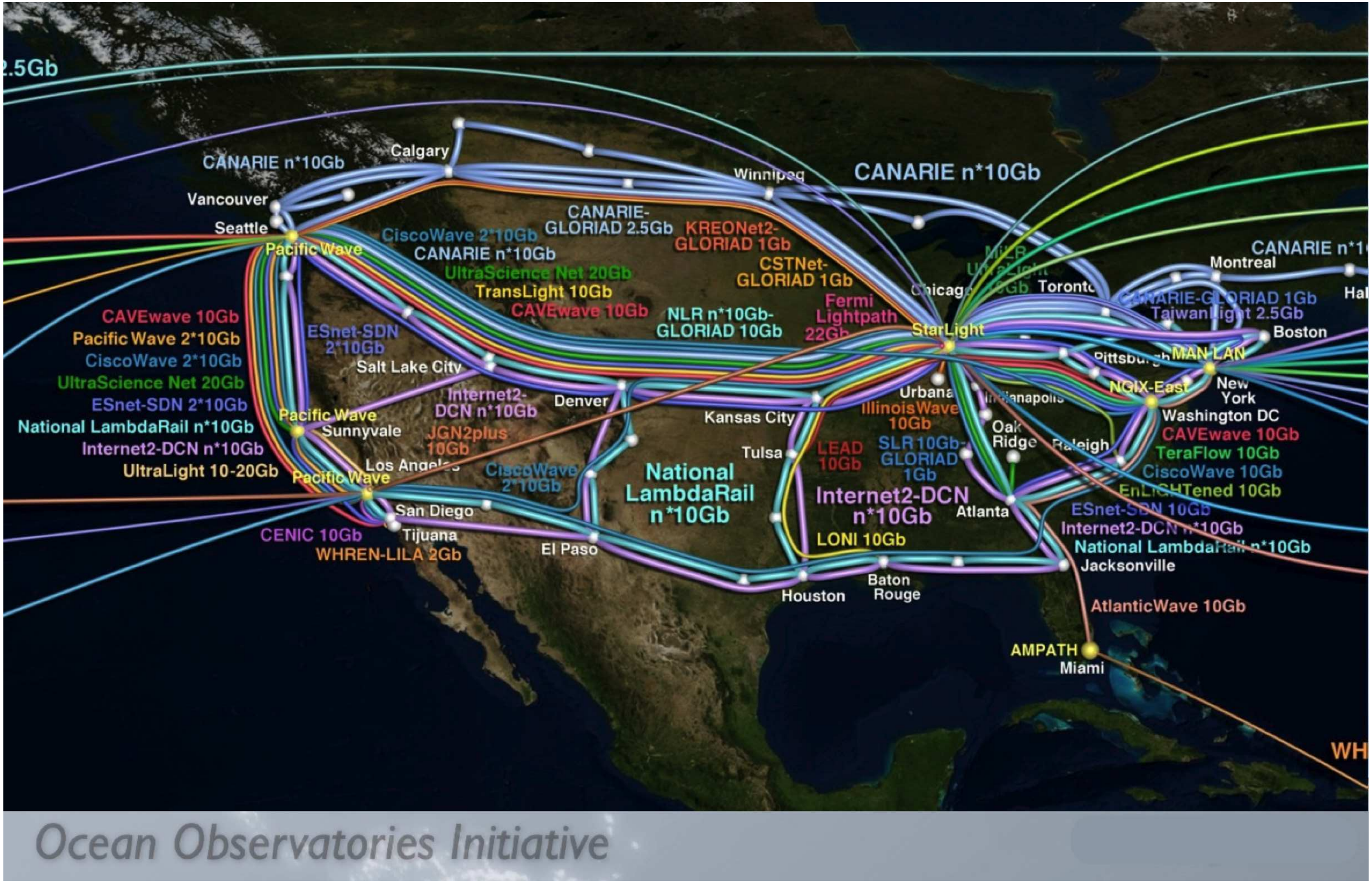




Visible & UV Radiation
sensible heat transfer
dust
Ice
Precipitation
Infrared Radiation
WIND
Aerosols
Sea Spray
Evaporation
DMS flux
Halogens
Surfactants
τ wind stress
Wave Breaking
CO_2 exchange
Near Surface Shear and Microbreaking
Wave–Current Interaction
Bubble Production and Turbulence
Penetrating Radiation
Phytoplankton and Bacteria
Langmuir Circulations
Shear at Mixed Layer Base
Internal Wave Radiation
Jayne Doucette WHOI Graphics

Ocean Observatories Initiative




CANARIE n*10Gb
Calgary
Vancouver
Seattle
Pacific Wave
Winnipeg
CANARIE n*10Gb
CiscoWave 2*10Gb
CANARIE-GLORIAD 2.5Gb
KREONet2-GLORIAD 1Gb
CANARIE n*10Gb
UltraScience Net 20Gb
TransLight 10Gb
CSTNet-GLORIAD 1Gb
CAVEwave 10Gb
NLR n*10Gb-GLORIAD 10Gb
Fermi Lightpath 22Gb
StarLight
Toronto
Montreal
CANARIE-GLORIAD 1Gb
TaiwanLight 2.5Gb
Boston
MAN LAN
New York
NGIX-East
Washington DC
CAVEwave 10Gb
TeraFlow 10Gb
CiscoWave 10Gb
EnLIGHTened 10Gb
ESnet-SDN 10Gb
Internet2-DCN n*10Gb
National LambdaRail n*10Gb
Jacksonville
AtlanticWave 10Gb
AMPATH
Miami
CAVEwave 10Gb
Pacific Wave 2*10Gb
CiscoWave 2*10Gb
UltraScience Net 20Gb
ESnet-SDN 2*10Gb
National LambdaRail n*10Gb
Internet2-DCN n*10Gb
UltraLight 10-20Gb
ESnet-SDN 2*10Gb
Salt Lake City
Internet2-DCN n*10Gb
Denver
Pacific Wave
Sunnyvale
JGN2plus 10Gb
Los Angeles
Pacific Wave
CiscoWave 2*10Gb
San Diego
Tijuana
CENIC 10Gb
WHREN-LILA 2Gb
El Paso
National LambdaRail n*10Gb
Kansas City
Tulsa
Urbana
IllinoisWave 10Gb
LEAD 10Gb
SLR 10Gb-GLORIAD 1Gb
Oak Ridge
Raleigh
Internet2-DCN n*10Gb
Atlanta
LONI 10Gb
Houston
Baton Rouge
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*.

  $\Longrightarrow$ Can describe communication protocols as *types*

  $\Longrightarrow$ 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]

$\Downarrow$

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

$\Downarrow$

Formalisation of W3C WS-CDL [ESOP’07]

$\Downarrow$

Scribble at π4 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* (WS-CLs) there is on ongoing debate on the best foundation for *Process-Aware Information Systems* (PAISs): *Petri nets* or *Pi calculus*. Example 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;
        }
    }
}
```

Dr Gary Brown (Pi4 Tech) in 2007

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]

$\Downarrow$

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

$\Downarrow$

Formalisation of W3C WS-CDL [ESOP'07]

$\Downarrow$

Scribble at π4 Technology

$\Downarrow$

Multiparty Session Types [POPL'08]

$\Downarrow$

Scribble redhat

Dialogue between Industry and Academia

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

$\Downarrow$

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

$\Downarrow$

Formalisation of W3C WS-CDL [ESOP'07]

$\Downarrow$

Scribble at π4 Technology

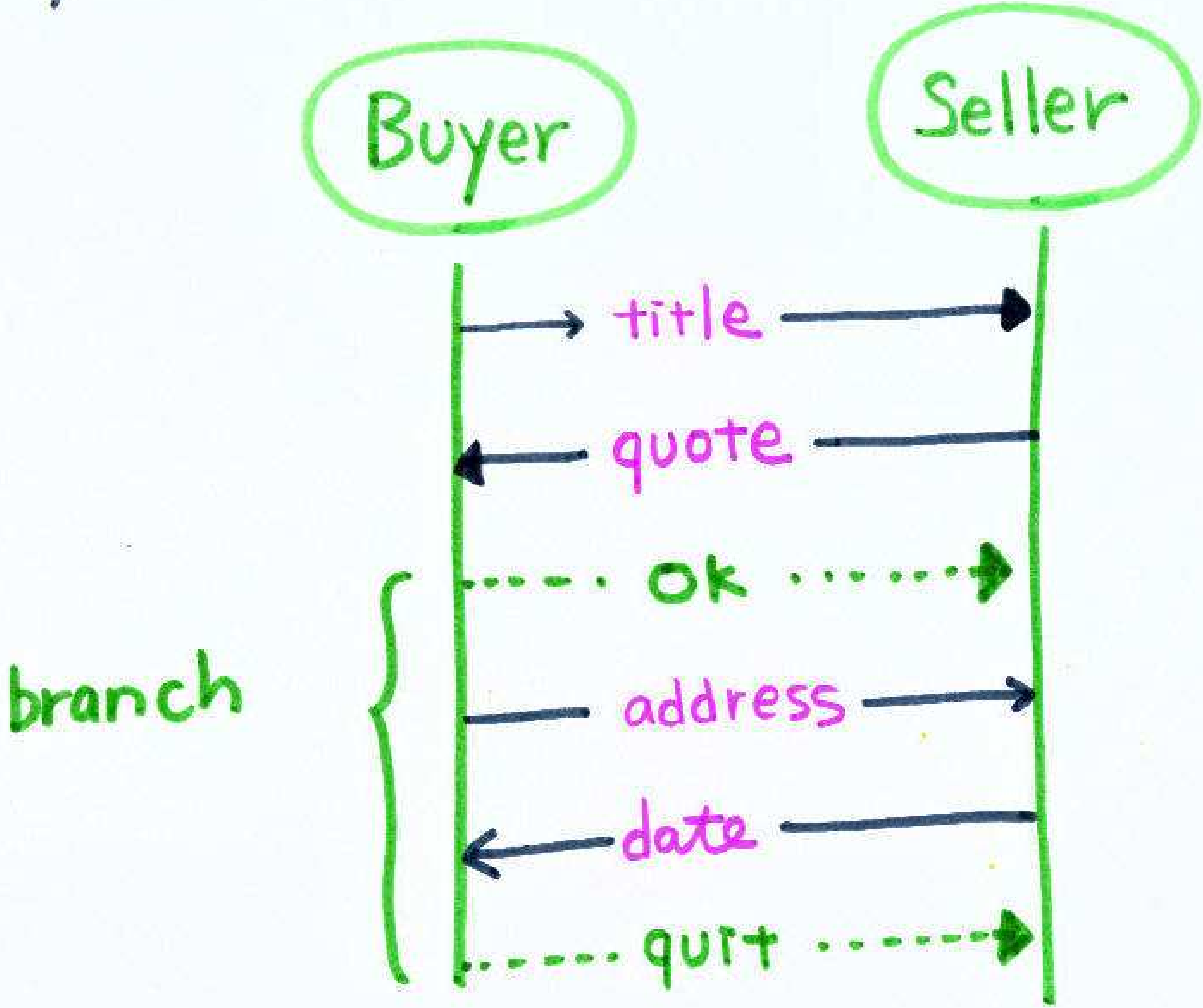
$\Downarrow$

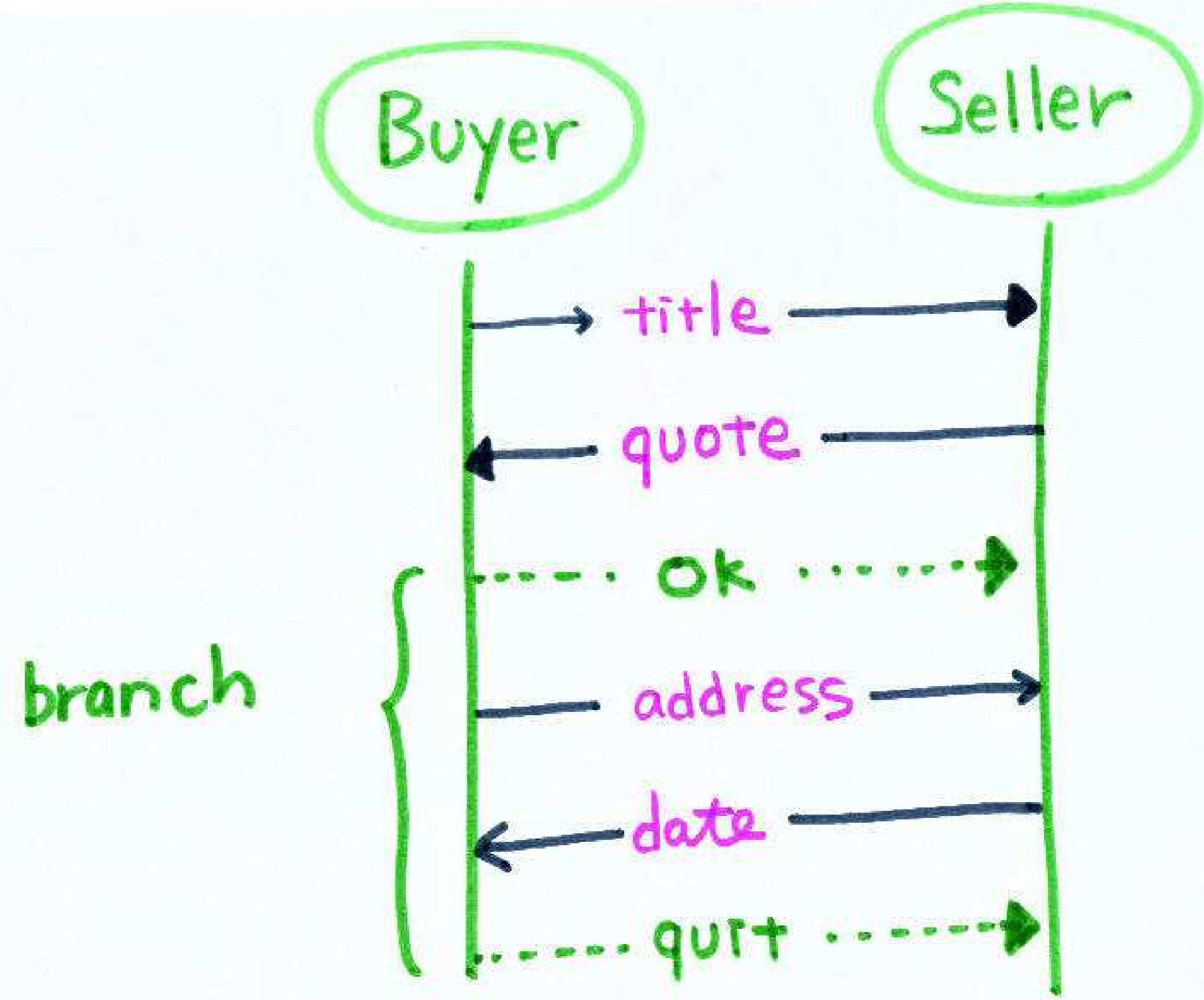
Multiparty Session Types [POPL'08]

$\Downarrow$

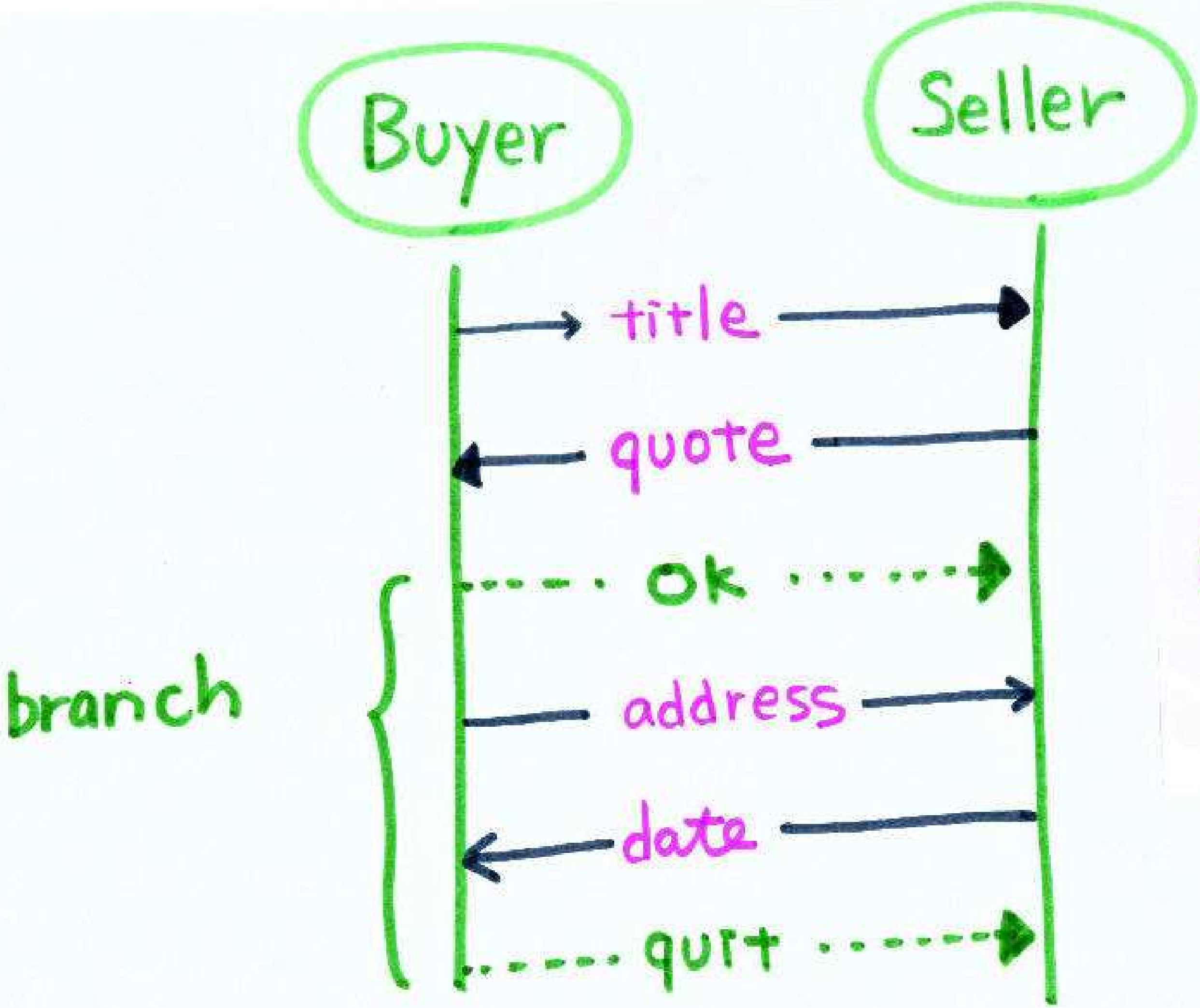
OOI OCEAN OBSERVATORIES INITIATIVE Scribble redhat SAVARA Cognizant

Binary Session Types : Buyer - Seller Protocol





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



P has T

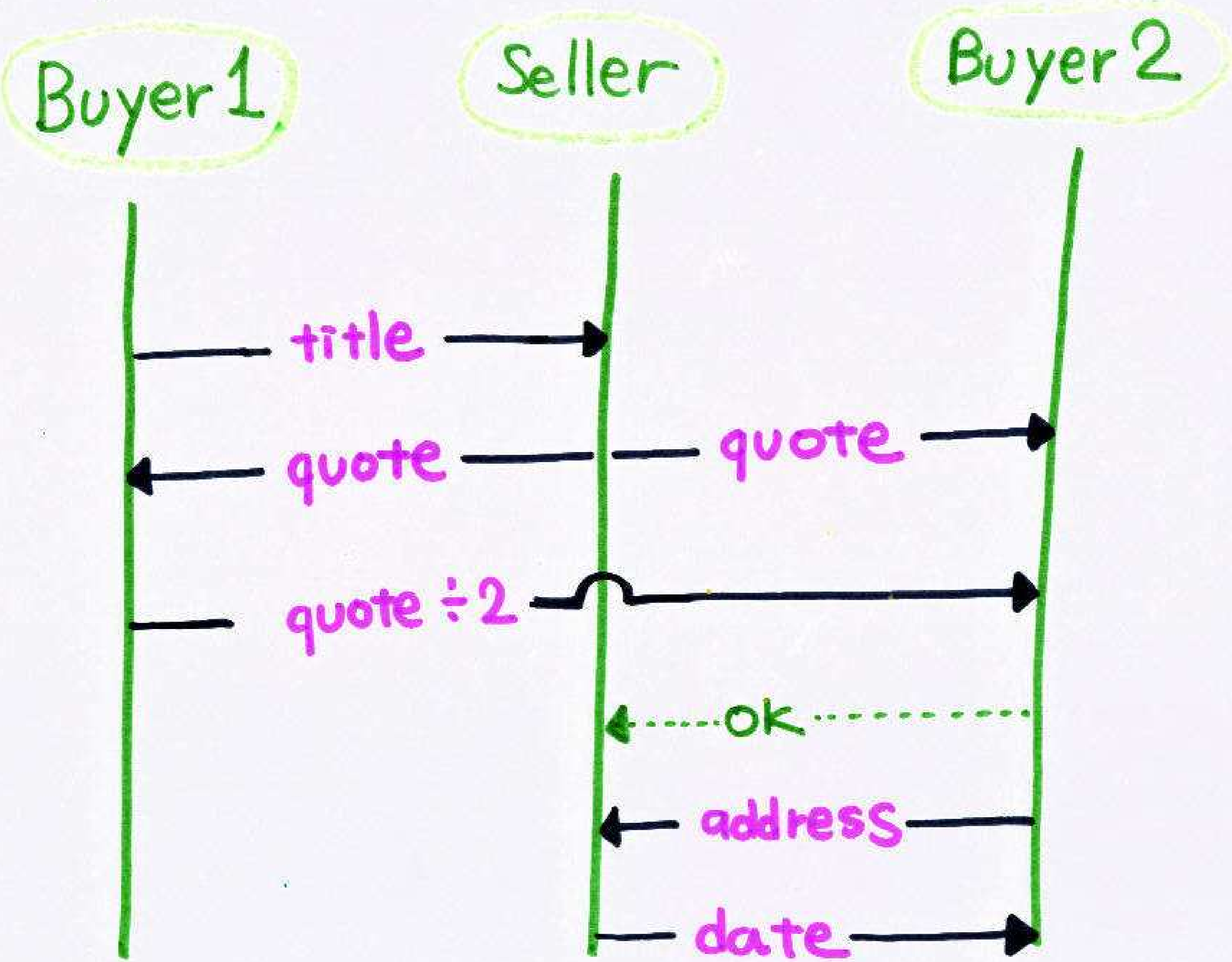
Q has $\overline{T}$ dual

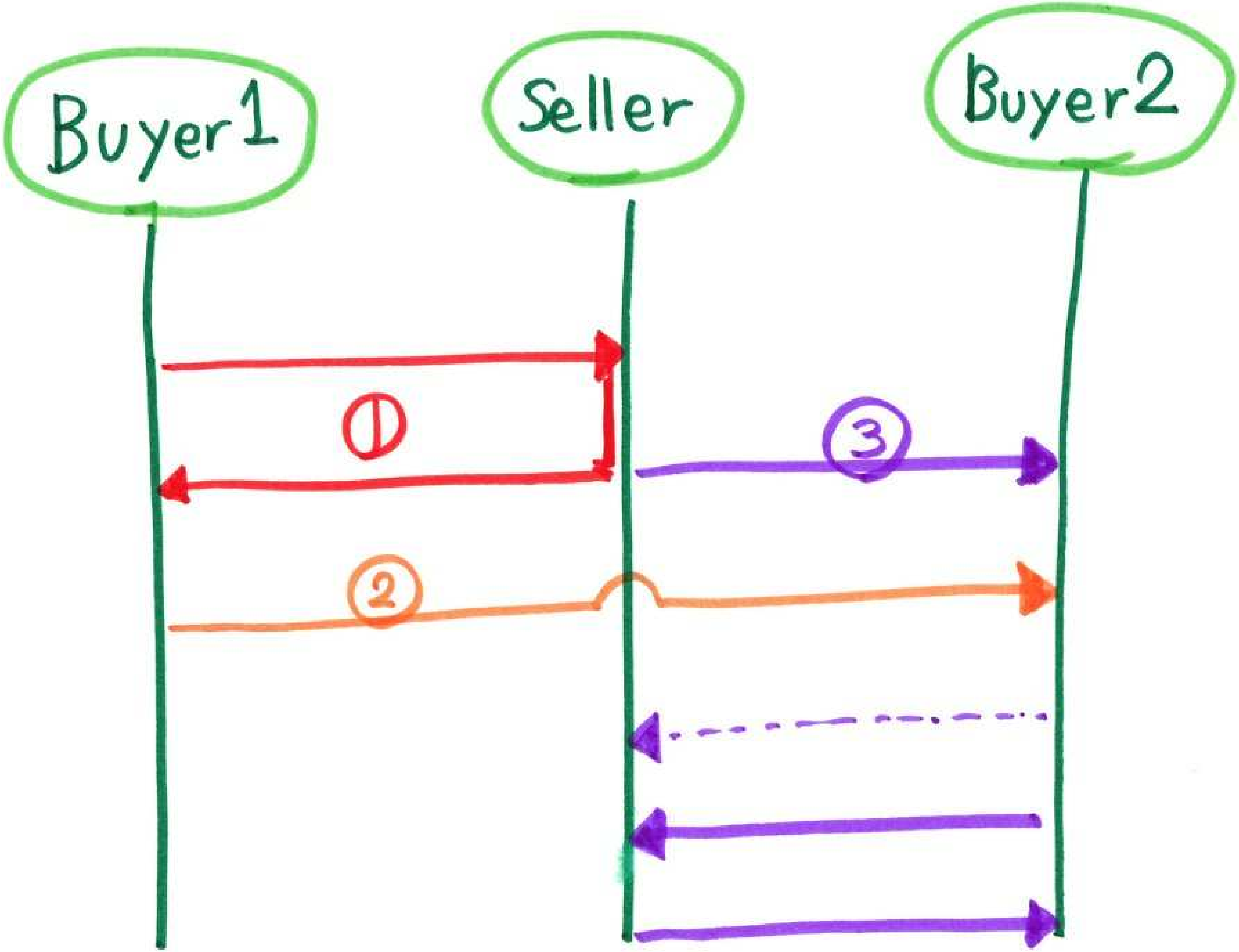
P | Q typable

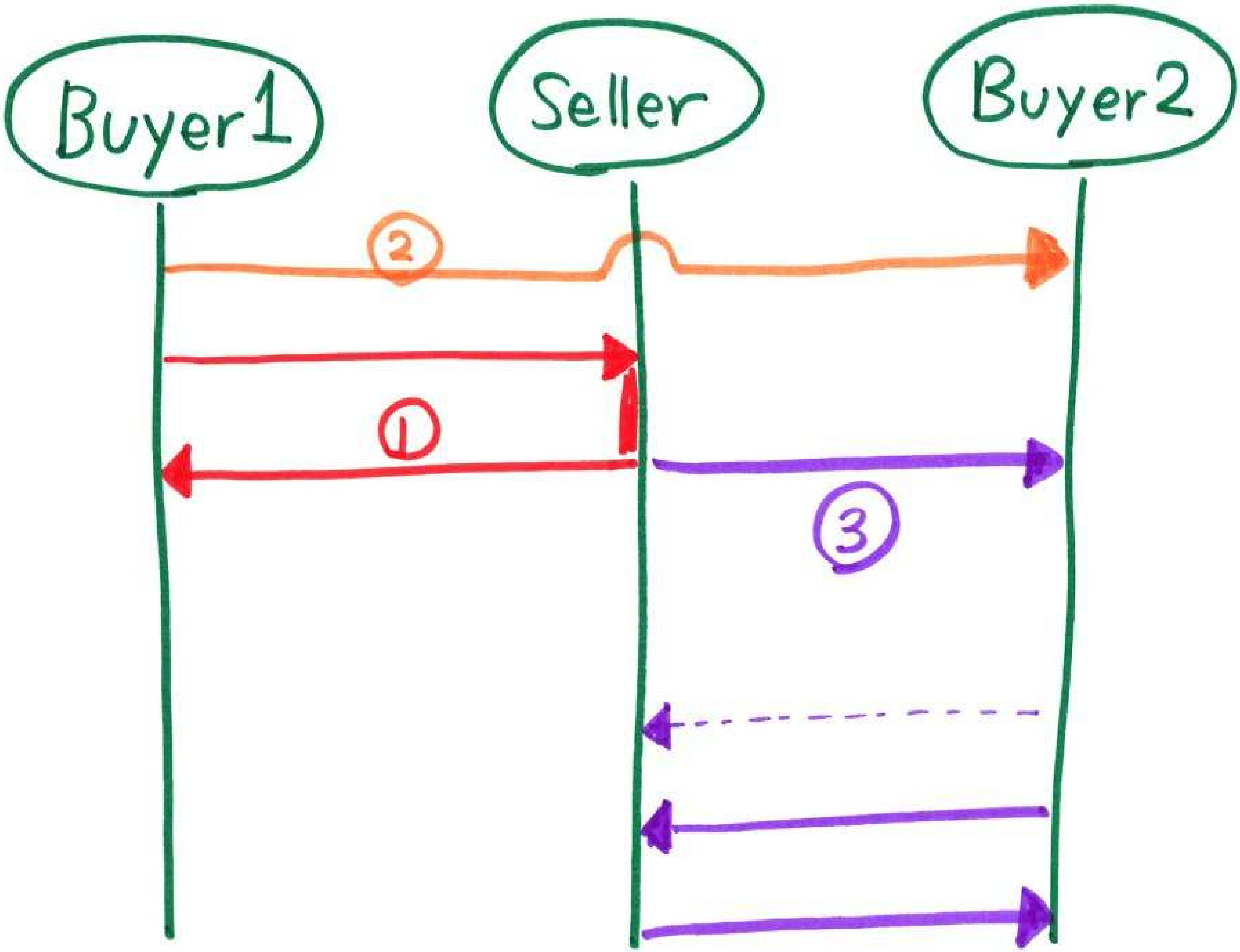
$$!\,\mathsf{String}\,;\,?\,\mathsf{Int}\,;\,\oplus\{\mathsf{ok}: !\,\mathsf{String}\,;\,?\,\mathsf{Date}\,;\,\mathsf{end},\ \mathsf{quit}: \mathsf{end}\}$$

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

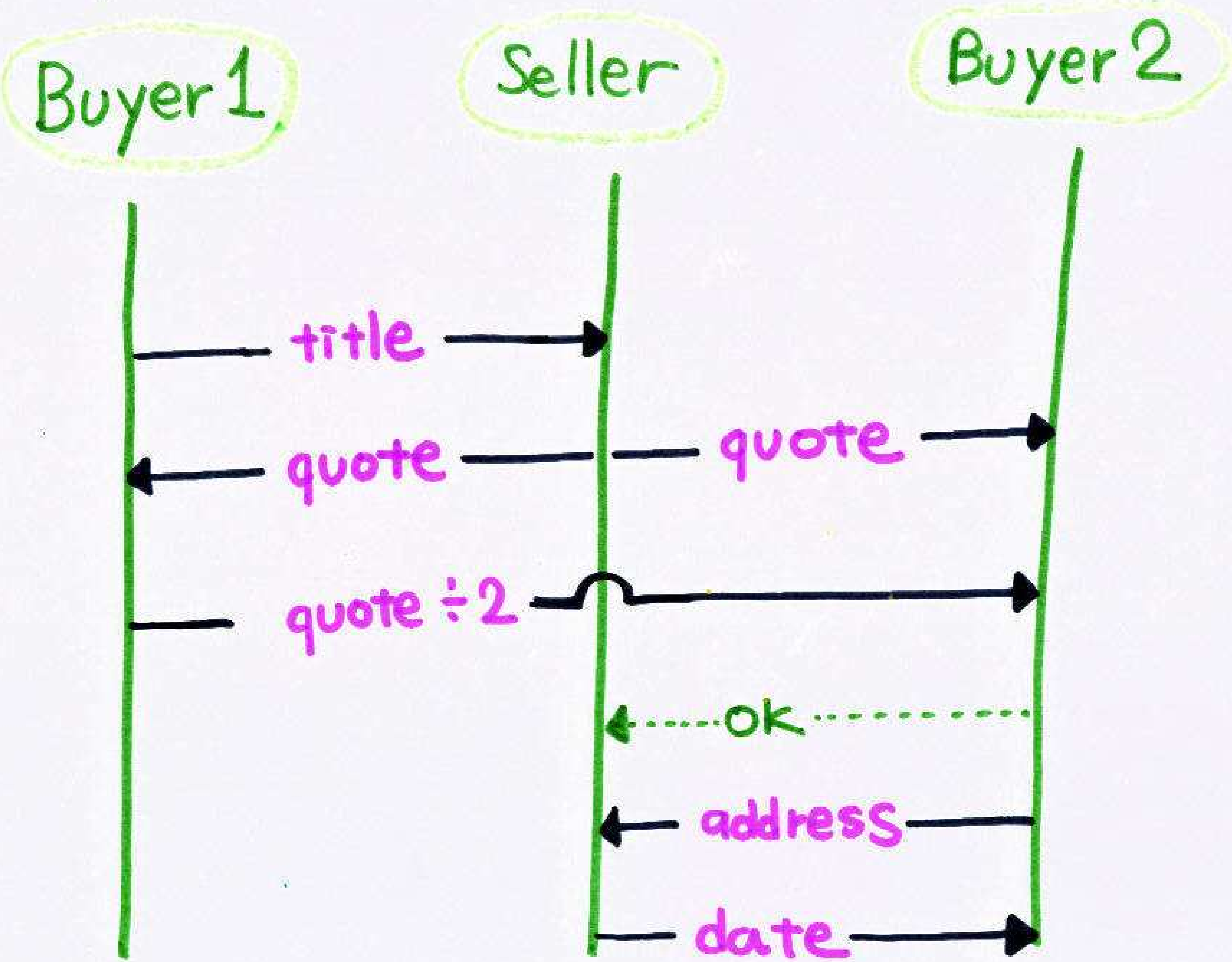
Multiparty Session Types



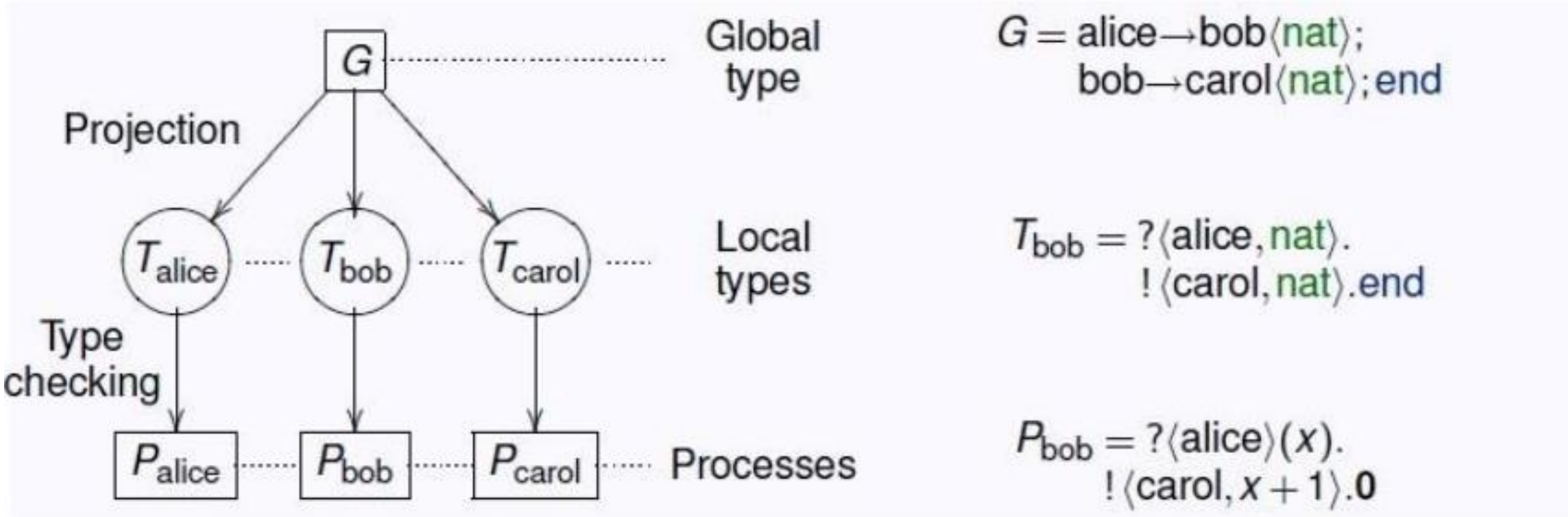

Buyer1
Seller
Buyer2
1
2
3


Buyer1
Seller
Buyer2
2
1
3

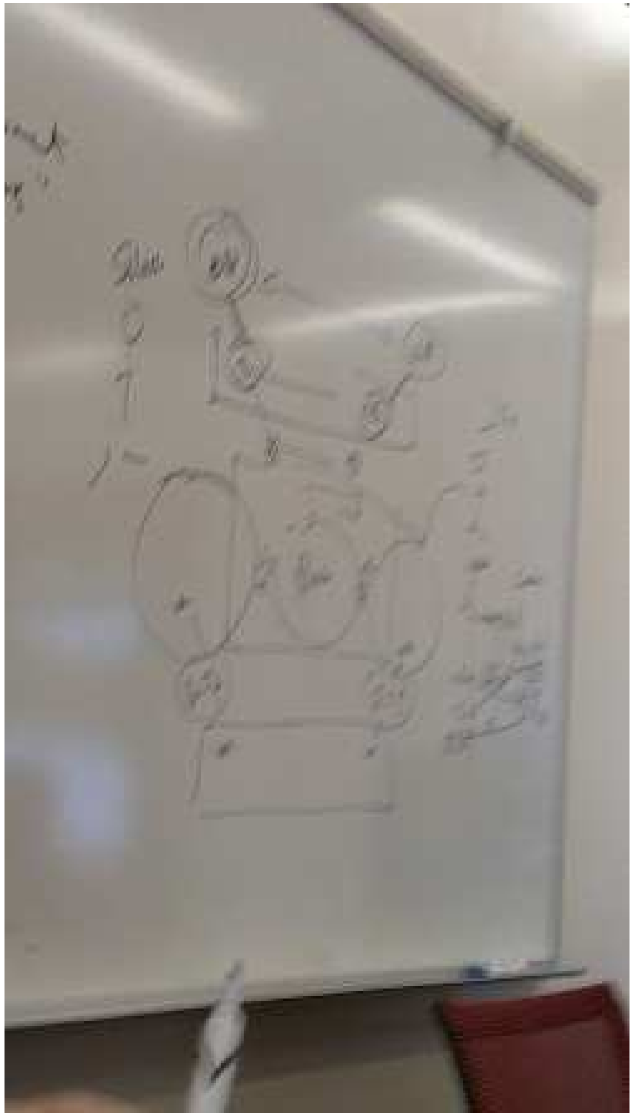
Multiparty Session Types



Session Types Overview



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



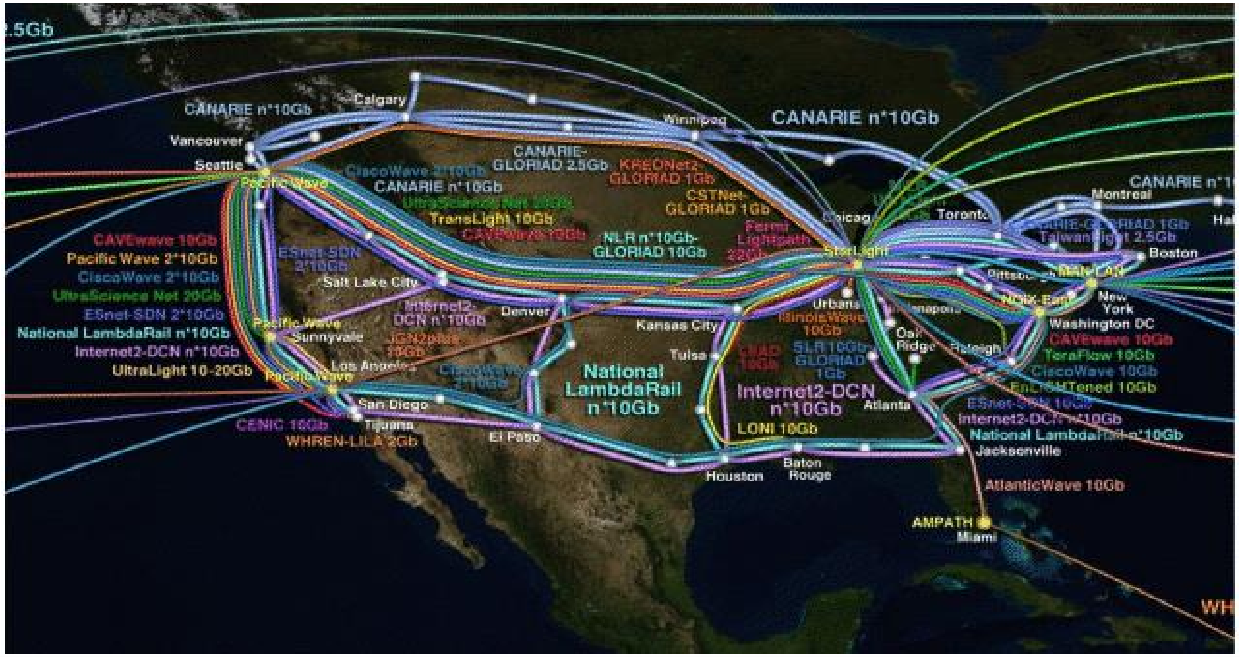
Evolution Of MPST

- 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]

Ocean Observatory Initiative (OOI)

OOI aims: to deploy an infrastructure (global network) to expand the scientists' ability to remotely study the ocean



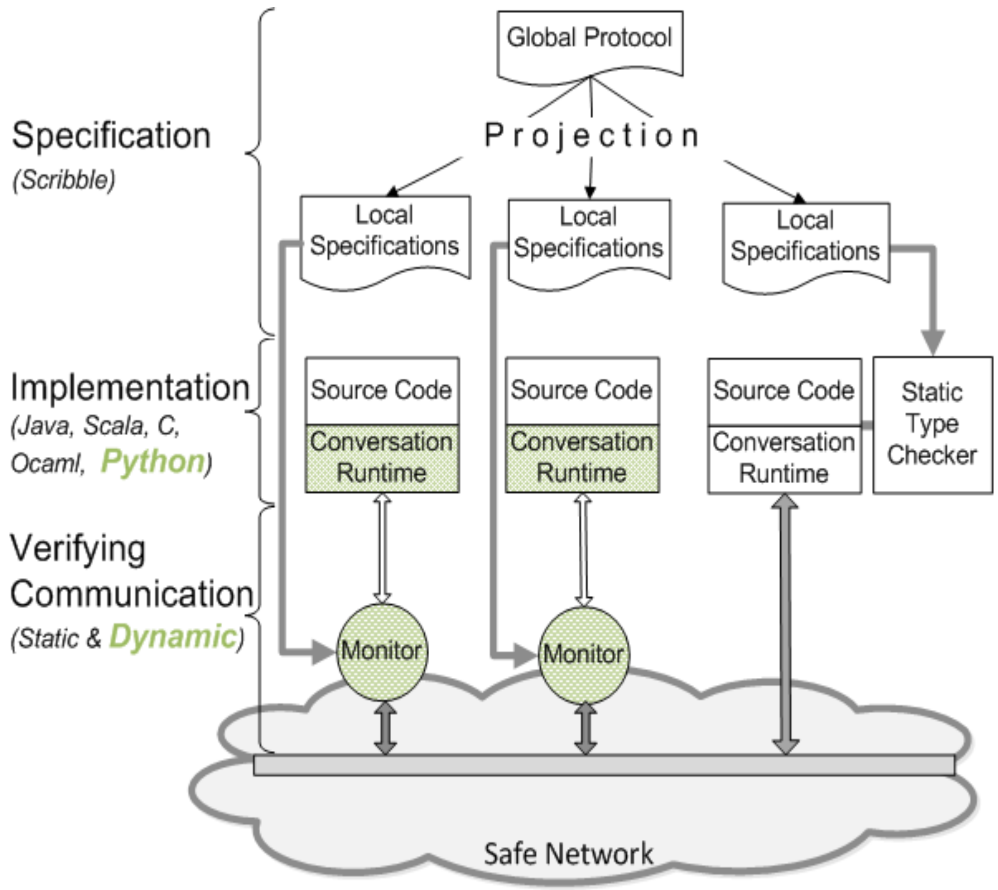
Usage: Integrate real-time data acquisition, processing and data storage for ocean research,…

OOI: verification challenges

- applications written in **different** languages, running on **heterogeneous** hardware in an **asynchronous** network.
- different authentication domains, external **untrusted** applications
- various distributed protocols
- requires correct, safe interactions

Session Types for Runtime Verification

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



2-level Verification

1. Writing correct global protocols with **Scribble Compiler**

2. Verify programs via ***local monitors***

2-level Verification

1. Writing correct global protocols with **Scribble Compiler**

2. Verify programs via ***local monitors***

www.scribble.org

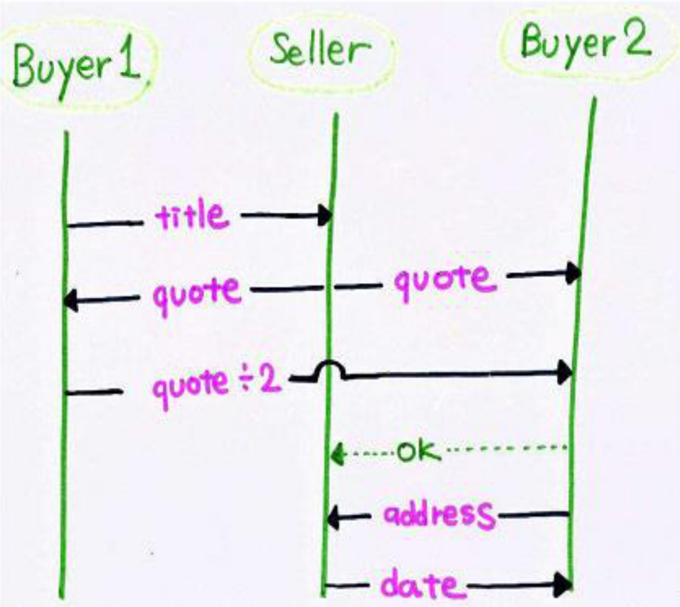
A Global Protocol

Import type → protocol def → send-receive → recursion → choice

```
type <python> "StringType" from "Lib/types.py" as str;

global protocol Negotiation(role P, role R, role A) {
    offer(string) from P to R;
    offer(string) from R to A;
    (string) from A to R;
    rec START {
        choice at R {
            accept() from R to P;
            confirm() from P to R;
        } or {
            offer(string) from R to P;
            (conditions:string) from P to R;
            continue START;
        } or {
            reject() from R to P;
            confirm() from R to P;}}}
```

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;
} } }
```

Global protocol well-formedness 1/2

```
global protocol ChoiceAmbiguous(role A, role B, role C) {
  choice at A {
    m1() from A to B; // X
    m2() from B to C;
    m3() from C to A;
  } or {
    m1() from A to B; // X
    m5() from B to C;
    m6() from C to A;
} }
```

```
global protocol ChoiceNotCommunicated(role A, role B, role C) {
  choice at A {
    m1() from A to B;
    m2() from B to C; // X
  } or {
    m4() from A to B;
} }
```

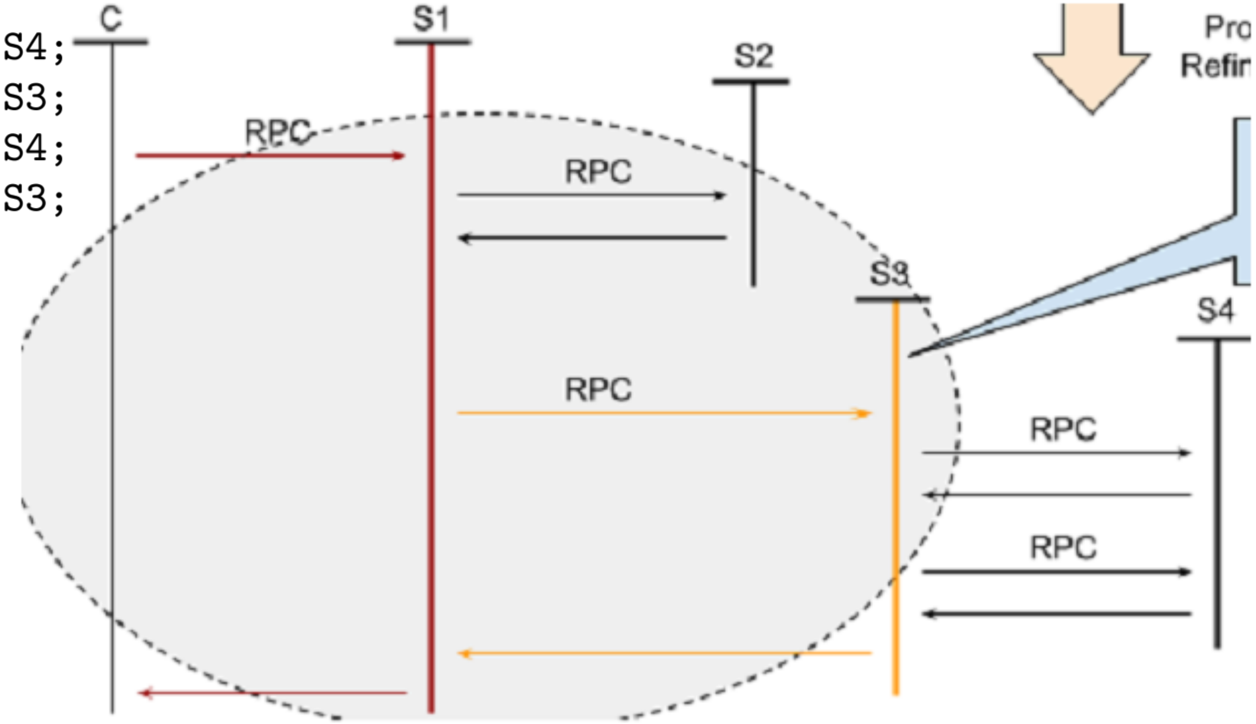
Global protocol well-formedness 2/2

```
global protocol ParallelNotLinear(role A, role B, role C) {
  par {
    m1() from A to B; // X
    m2() from B to C;
  } and {
    m1() from A to B; // X
    m4() from B to C;
} }
```

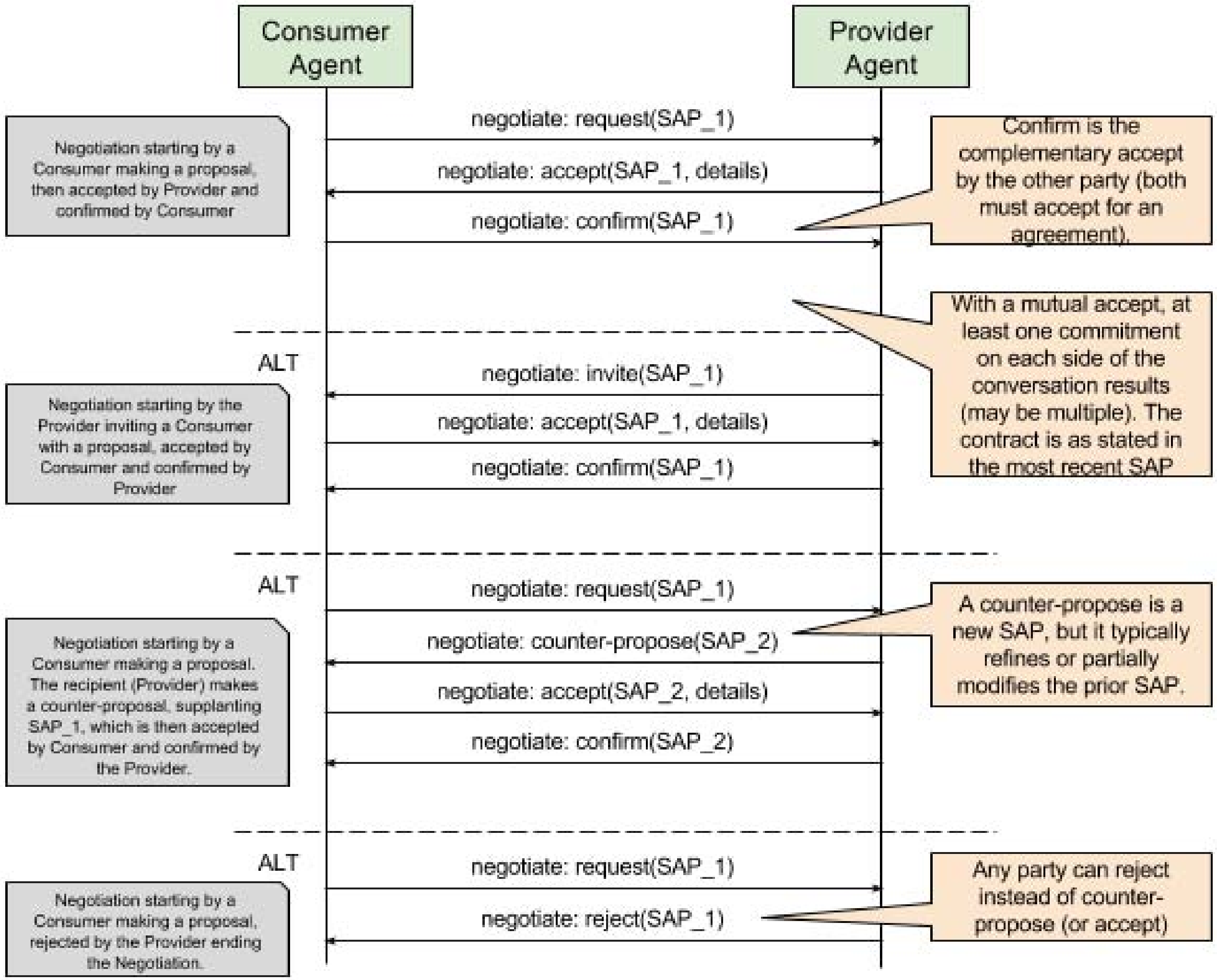
```
global protocol RecursionNoExit(role A, role B, role C, role D) {
  rec X {
    m1() from A to B;
    continue X;
  }
  m2() from A to B; // Unreachable for A, B
  m3() from C to D;
}
```

Application-level service call composition

```
// Direct specification
global protocol P3(role C, role S1, role S2, role S3, role S4)
{
    () from C to S1;
        () from S1 to S2;
        () from S2 to S1;
        () from S1 to S3;
            () from S3 to S4;
            () from S4 to S3;
            () from S3 to S4;
            () from S4 to S3;
        () from S3 to S1;
    () from S1 to C;
}
```



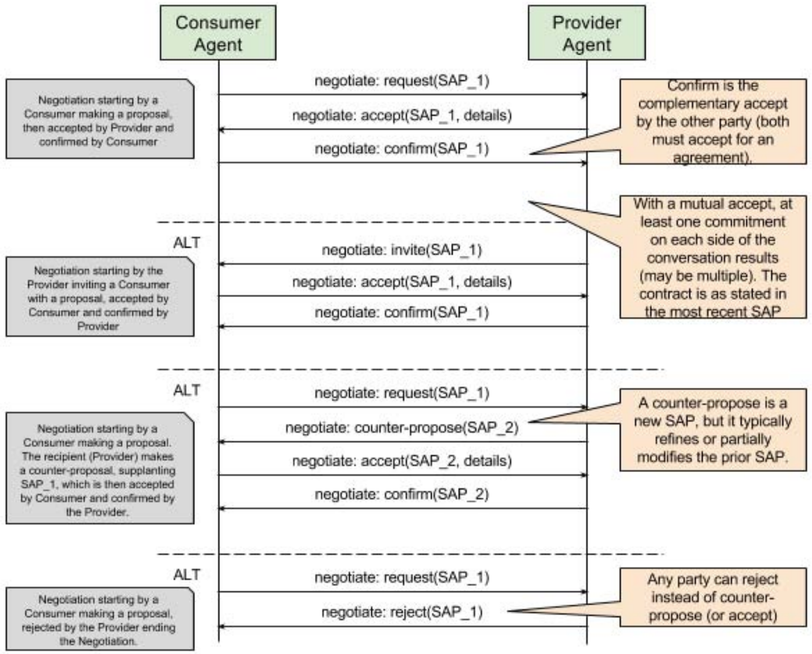
OOI agent negotiation 1/5



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

OOI agent negotiation 2/5

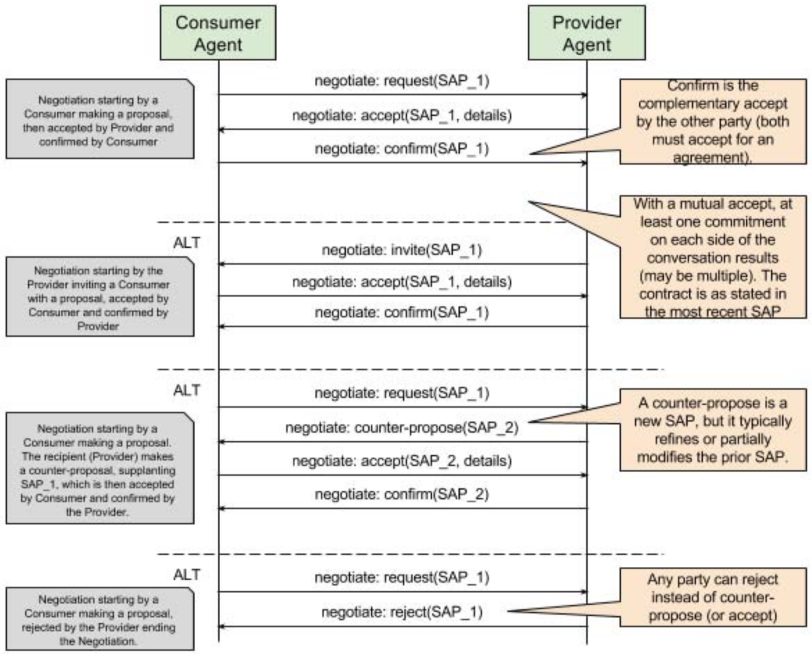
```
type <yml> "SAPDoc1" from "SAPDoc1.yml" as SAP;

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




```
}
```

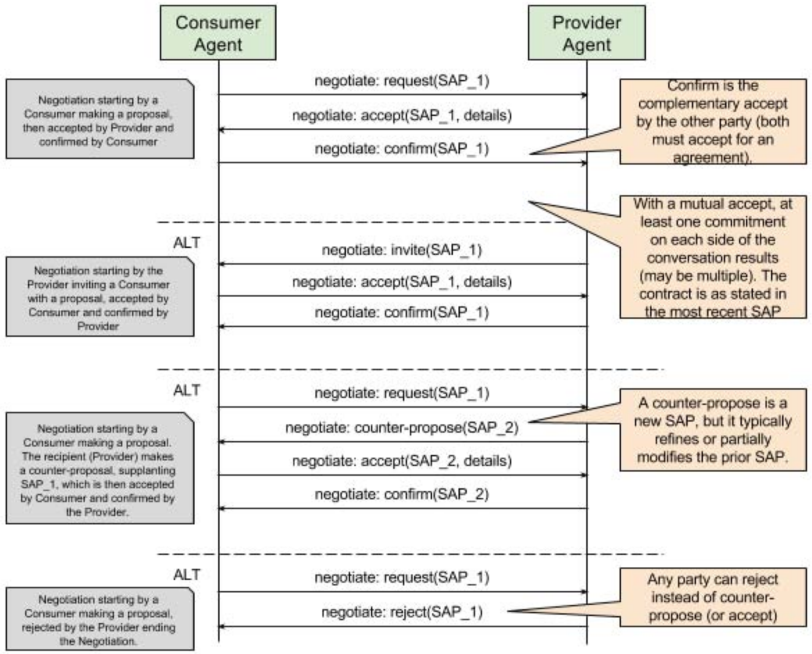
OOI agent negotiation 3/5 (choice)

```
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;

} }
```



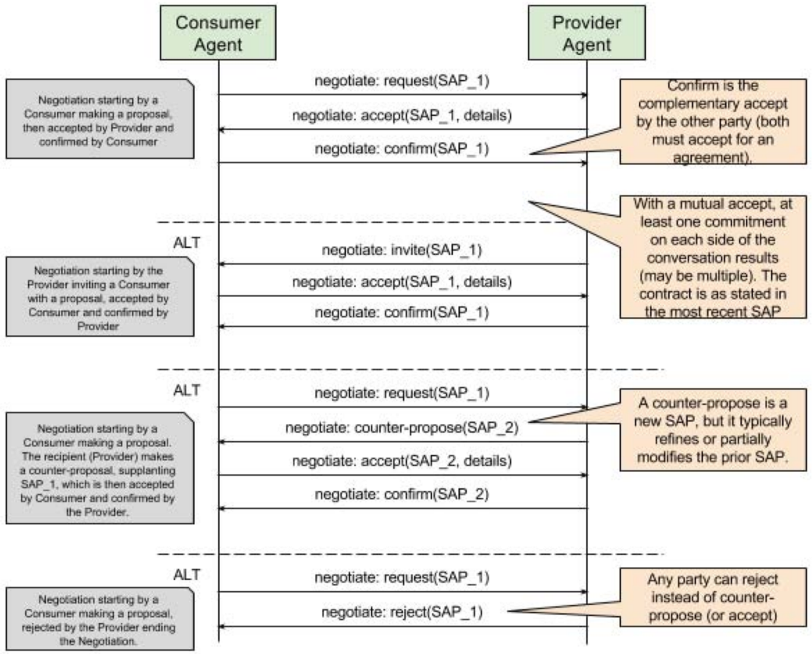
OOI agent negotiation 4/5

```
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;
      }
    }
```



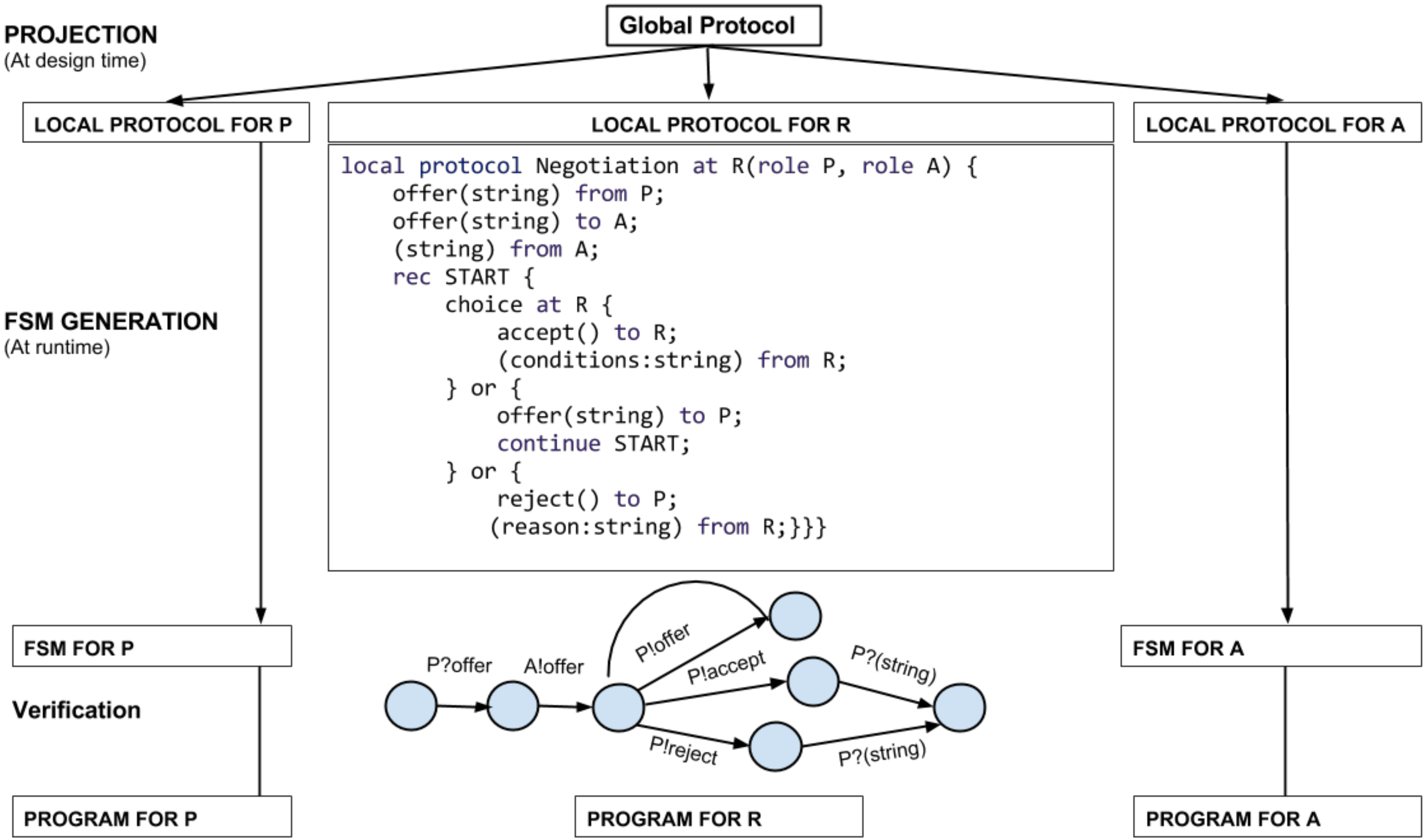

```
  }
```

2-level Verification

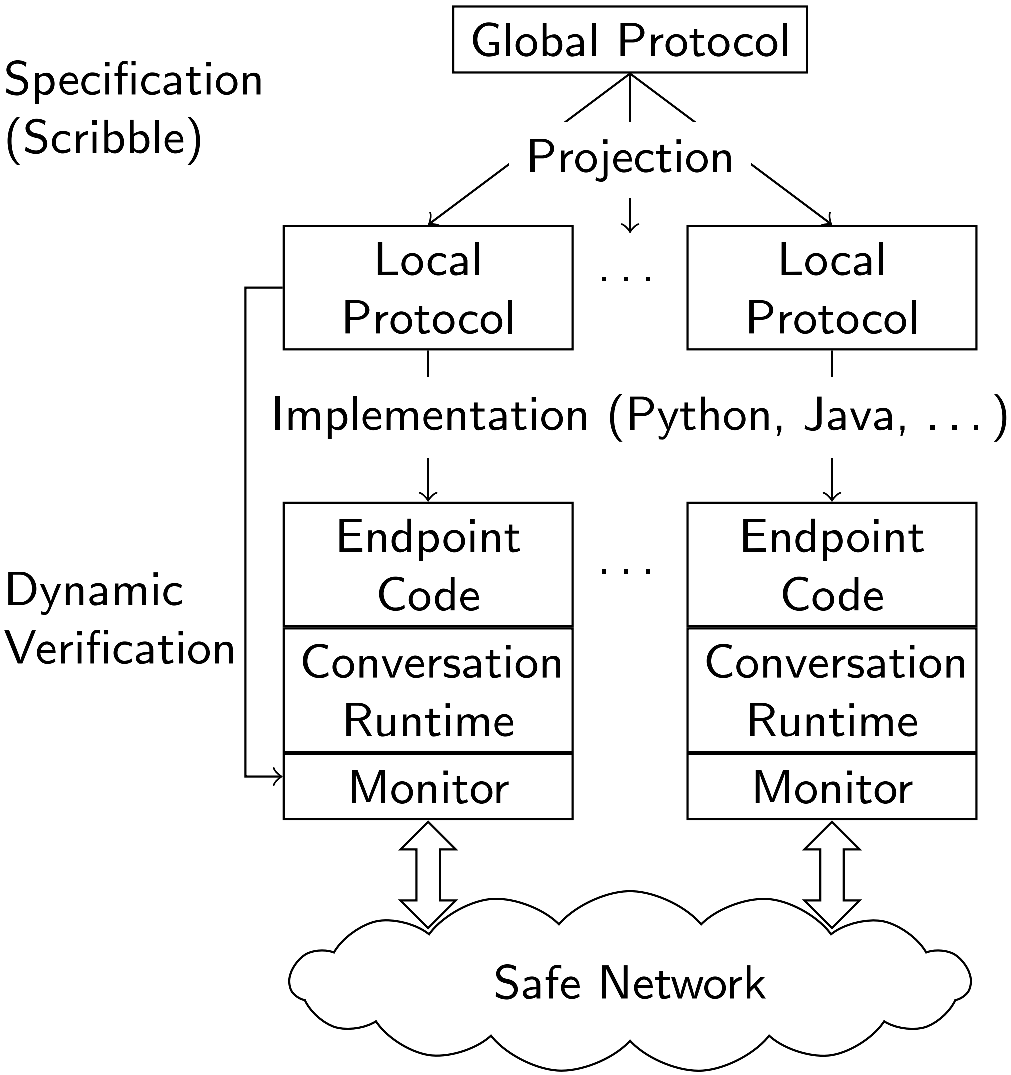
1. Writing correct global protocols with **Scribble Compiler**

2. Verify programs via ***local monitors***

Local Protocol Conformance



The Scribble Framework



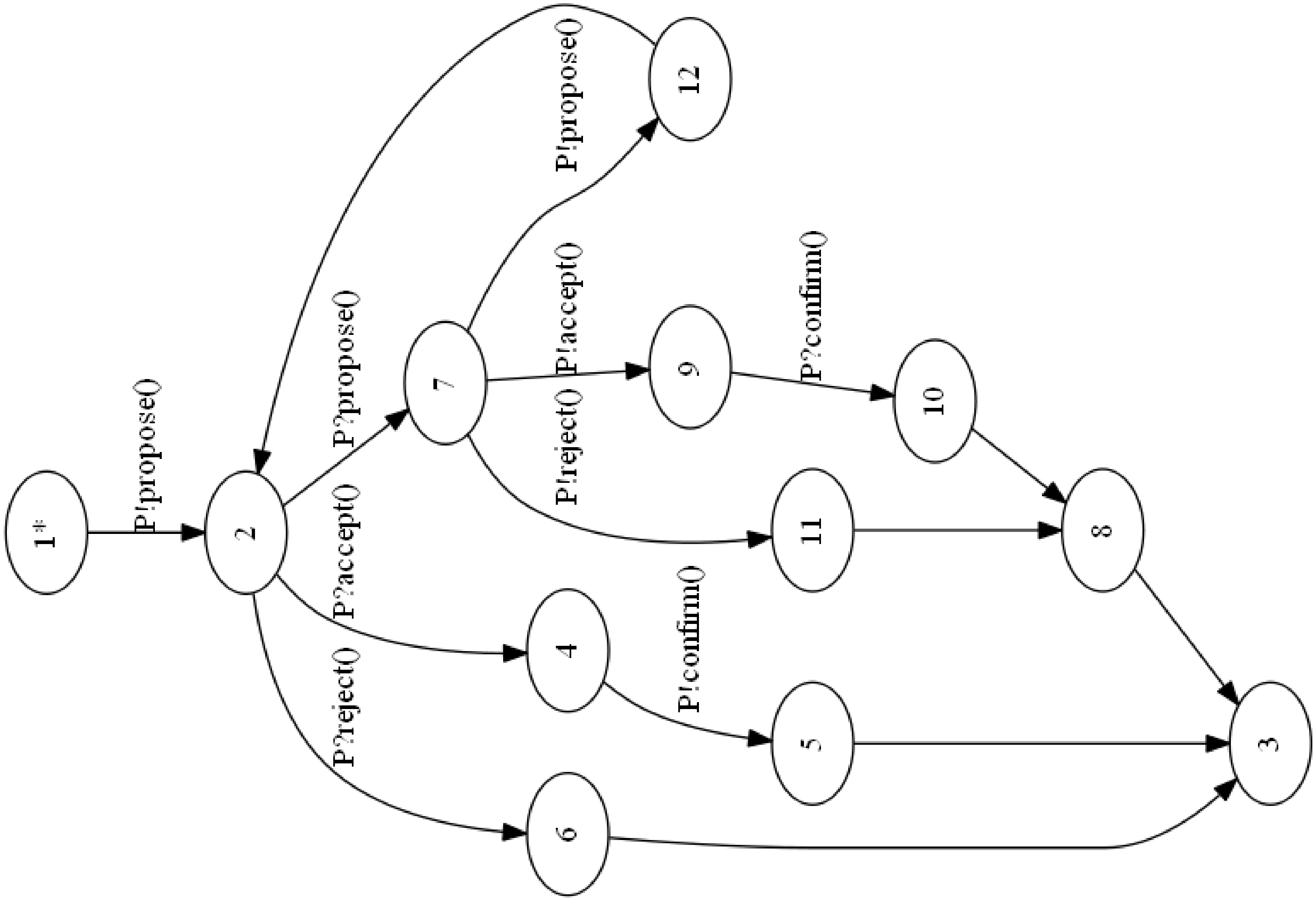
- Scribble global protocols
  - Well-formedness validation
- Scribble local protocols
  - FSM generation (for endpoint monitoring)
- (Heterogeneous) endpoint programs
  - Scribble Conversation API
  - (Interoperable) Distributed Conversation Runtime

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;
} } }
```

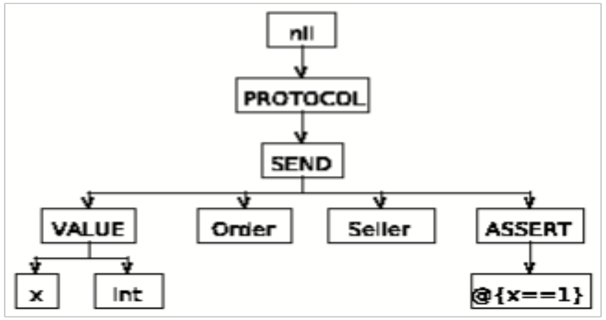
FSM generation (Negotiation Consumer)



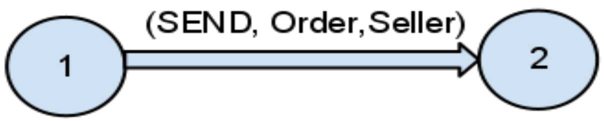
FSM Generator

Scribble: Order(x:int) to Seller @{x==1}

AST:

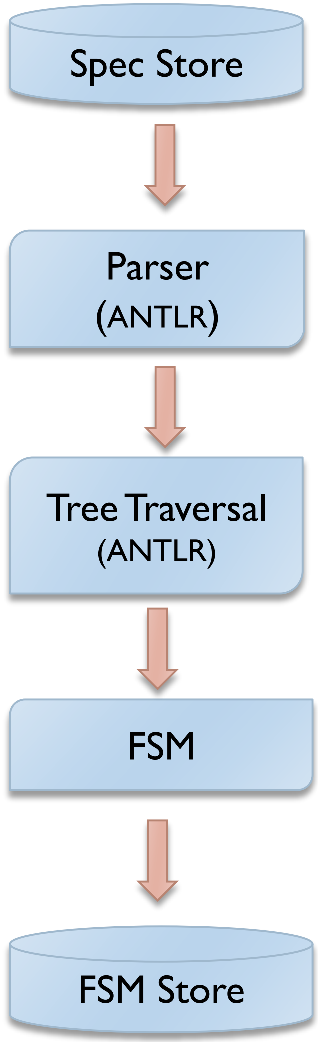


FSM:

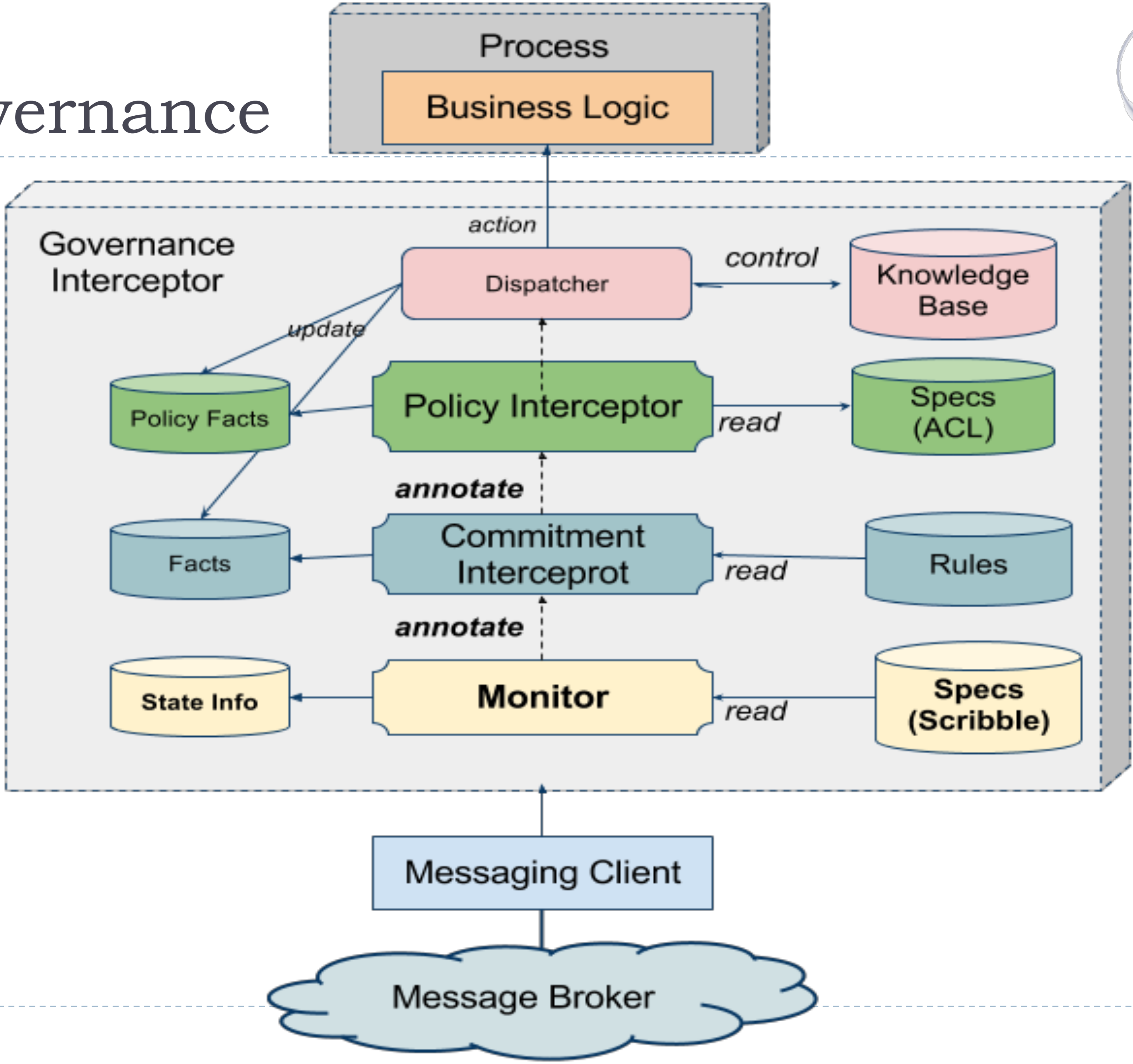


FSM transition_table:

(1, (send, order, seller) ->
(2, assertion_object, {'x':'int'})



Governance


Process
Business Logic
action
Governance Interceptor
Dispatcher
control
Knowledge Base
update
Policy Facts
Policy Interceptor
read
Specs (ACL)
annotate
Facts
Commitment Interceprot
read
Rules
annotate
State Info
Monitor
read
Specs (Scribble)
Messaging Client
Message Broker

Scribble Community

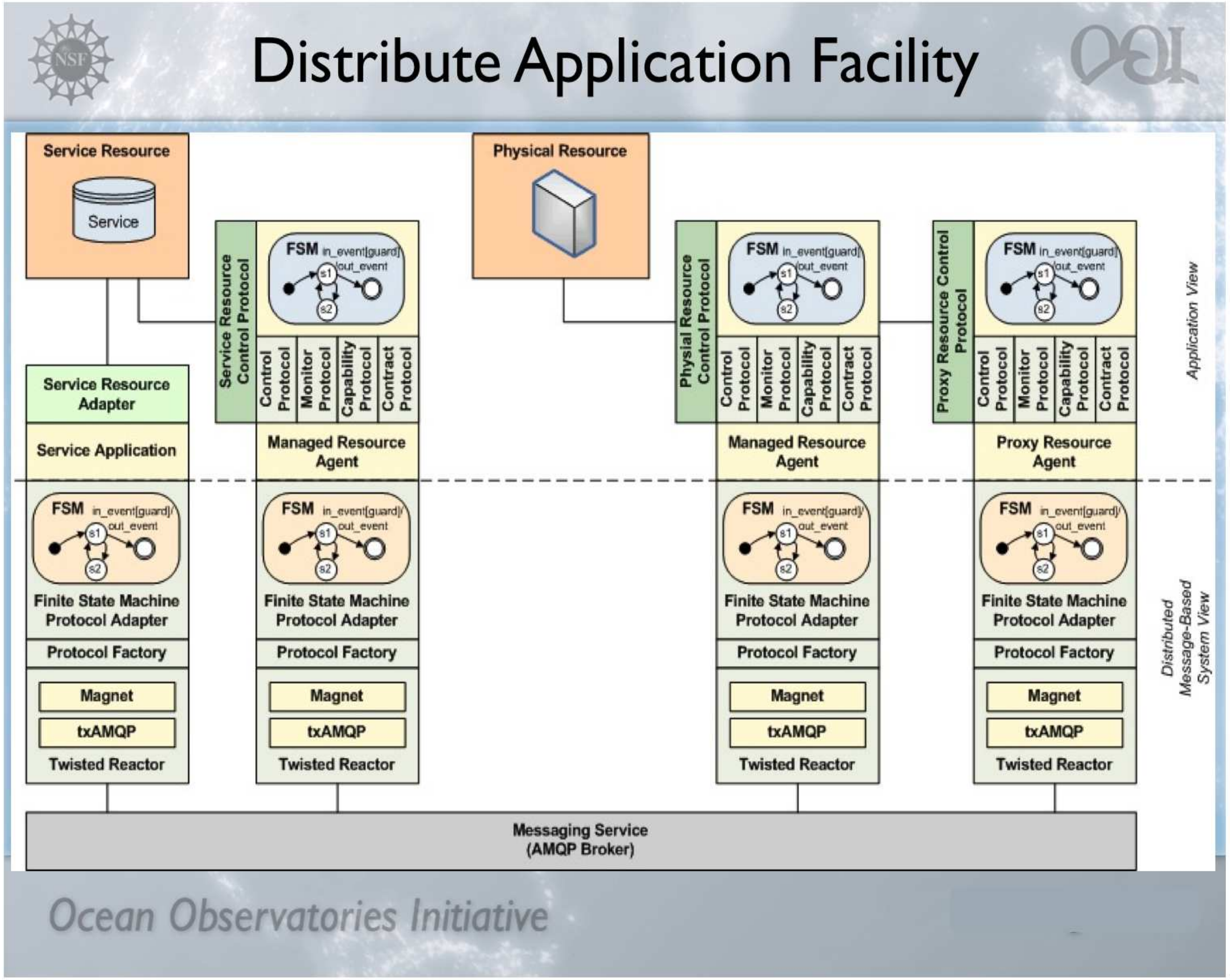
- Webpage:
  - www.scribble.org
- GitHub:
  - https://github.com/scribble
- Tutorial:
  - www.doc.ic.ac.uk/~rhu/scribble/tutorial.html
- Specification (0.3)
  - www.doc.ic.ac.uk/~rhu/scribble/langref.html

A theory for network monitoring

- Formalise MPST-monitoring and asynchronous networks.
- Introduce monitors as first-class objects in the theory
- Justify monitoring by soundness theorems.
  - Safety
    - monitors enforces specification conformance.
  - Transparency
    - monitors does not affect correct behaviours.
  - Fidelity
    - correspondence to global types is maintained.

Multiparty Sessions for Runtime Monitors

$$
\begin{array}{lcl}
A & ::= & \mathsf{tt} \mid \mathsf{ff} \mid e_1 = e_2 \mid e_1 < e_2 \mid \neg A \mid A_1 \wedge A_2 \mid A_1 \vee A_2 \\
e & ::= & v \mid e_1 + e_2 \mid e_1 - e_2 \mid e_1 * e_2 \mid e_1 \,\mathtt{mod}\, e_2 \qquad S ::= \mathsf{bool} \mid \mathsf{int} \mid \mathsf{string} \\
G & ::= & \mathtt{r}_1 \to \mathtt{r}_2 : \{l_i(x_i : S_i)\{A_i\}.G_i\}_{i \in I} \mid G_1 \mid G_2 \mid G_1; G_2 \mid \mu \mathtt{t}.G \mid \mathtt{t} \mid \epsilon \mid \mathsf{end} \\
T & ::= & \mathtt{r}!\{l_i(x_i : S_i)\{A_i\}.T_i\}_{i \in I} \mid \mathtt{r}?\{l_i(x_i : S_i)\{A_i\}.T_i\}_{i \in I} \mid T_1 \mid T_2 \mid T_1; T_2 \mid \\
 & & \mu \mathtt{t}.T \mid \mathtt{t} \mid \epsilon \mid \mathsf{end} \\
P & ::= & \overline{a}\langle s[\mathtt{r}] : T \rangle \mid a(y[\mathtt{r}] : T).P \mid k[\mathtt{r}_1, \mathtt{r}_2]!l\langle e \rangle \mid k[\mathtt{r}_1, \mathtt{r}_2]?\{l_i(x_i).P_i\}_{i \in I} \mid \\
 & & \mathsf{if}\ e\ \mathsf{then}\ P\ \mathsf{else}\ Q \mid P \mid Q \mid \mathbf{0} \mid \mu X.P \mid X \mid P; Q \mid (\nu a)\ P \mid (\nu s) P \\
N & ::= & [P]_\alpha \mid N_1 | N_2 \mid \mathbf{0} \mid (\nu a) N \mid (\nu s) N \mid \langle r\ ;\ h \rangle \\
r & ::= & a \mapsto \alpha \mid s[\mathtt{r}] \mapsto \alpha \qquad h ::= m \cdot h \mid \emptyset \qquad m ::= \overline{a}\langle s[\mathtt{r}] : T \rangle \mid s\langle \mathtt{r}_1, \mathtt{r}_2, l\langle v \rangle \rangle
\end{array}
$$


Distribute Application Facility
Service Resource
Service
Physical Resource
Service Resource Control Protocol
Physical Resource Control Protocol
Proxy Resource Control Protocol
FSM in_event[guard]/out_event
s1
s2
Control Protocol
Monitor Protocol
Capability Protocol
Contract Protocol
Service Resource Adapter
Service Application
Managed Resource Agent
Proxy Resource Agent
Finite State Machine Protocol Adapter
Protocol Factory
Magnet
txAMQP
Twisted Reactor
Messaging Service (AMQP Broker)
Application View
Distributed Message-Based System View
Ocean Observatories Initiative

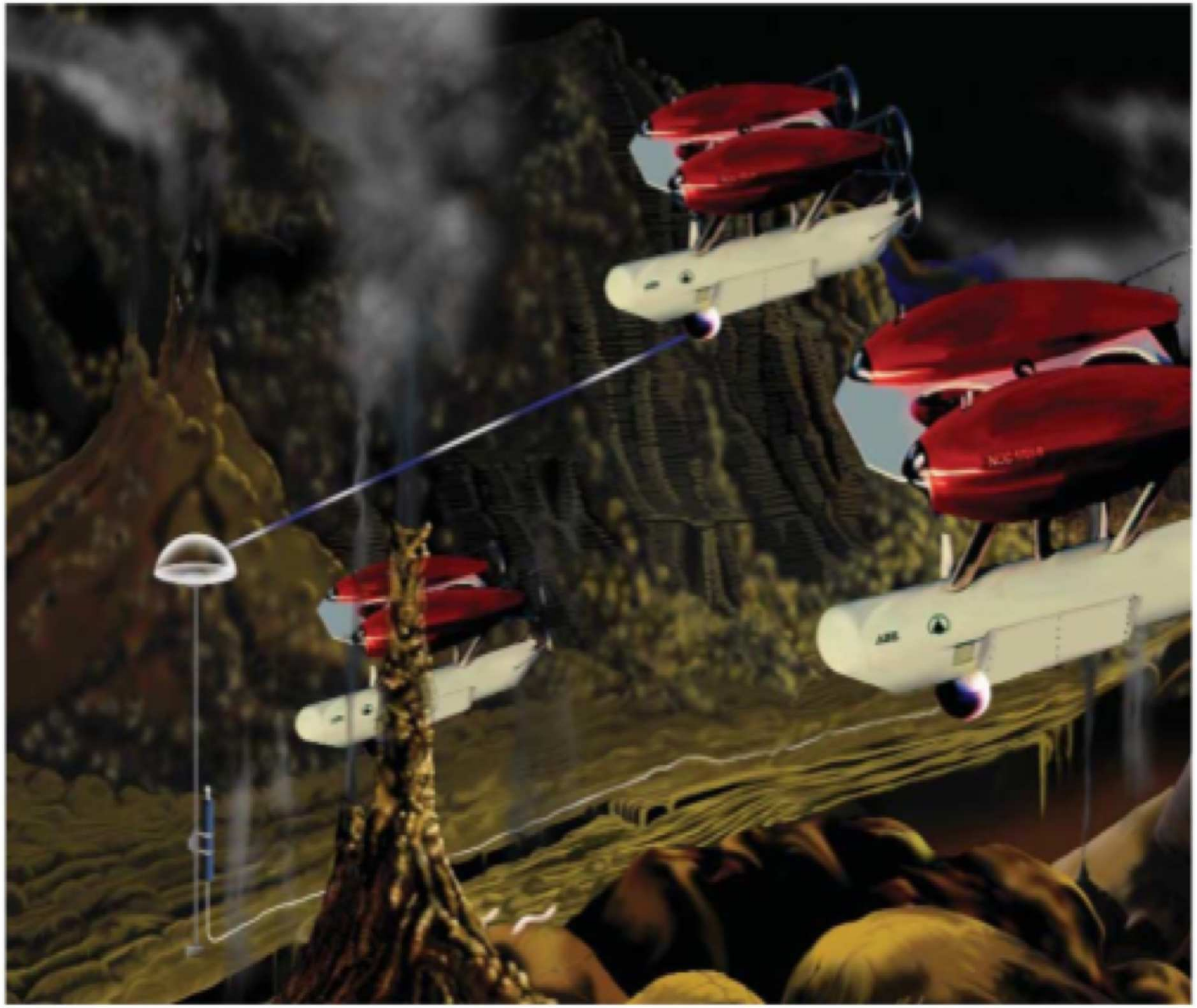


Figure 5: A coordinated set of autonomous underwater vehicles

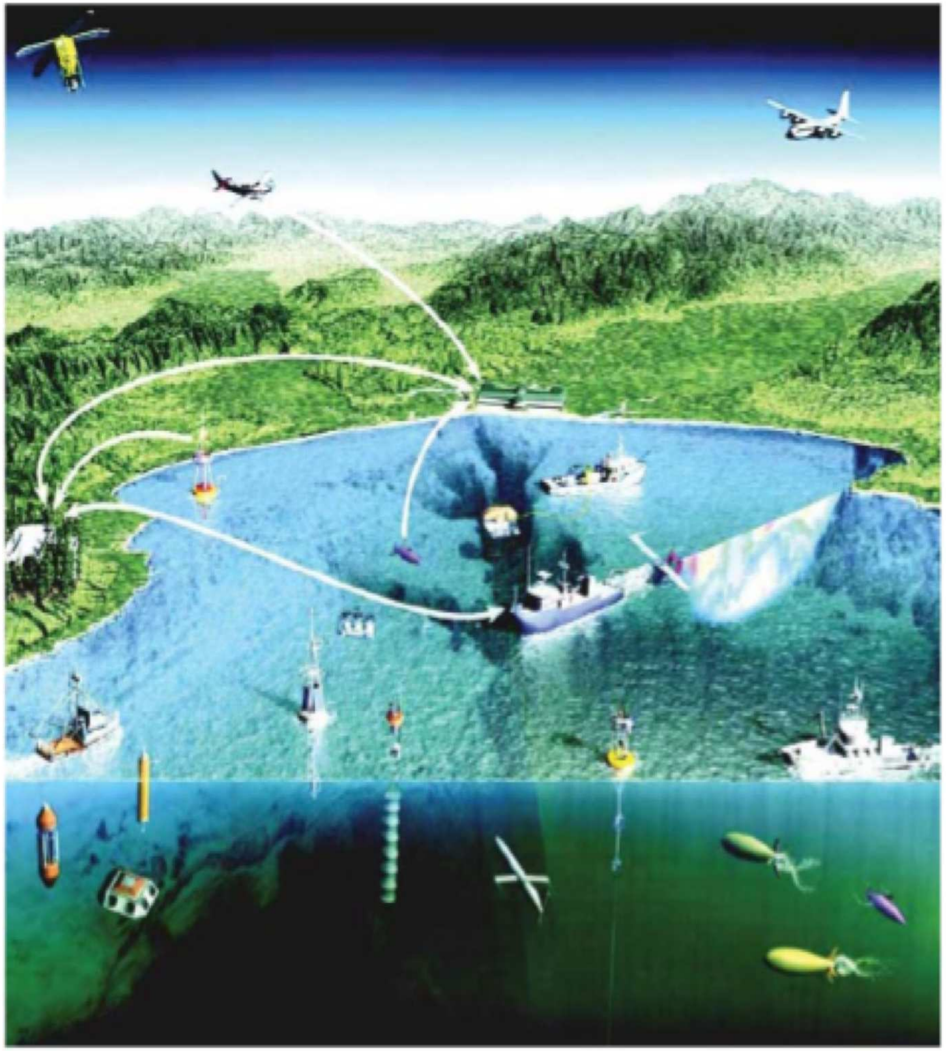
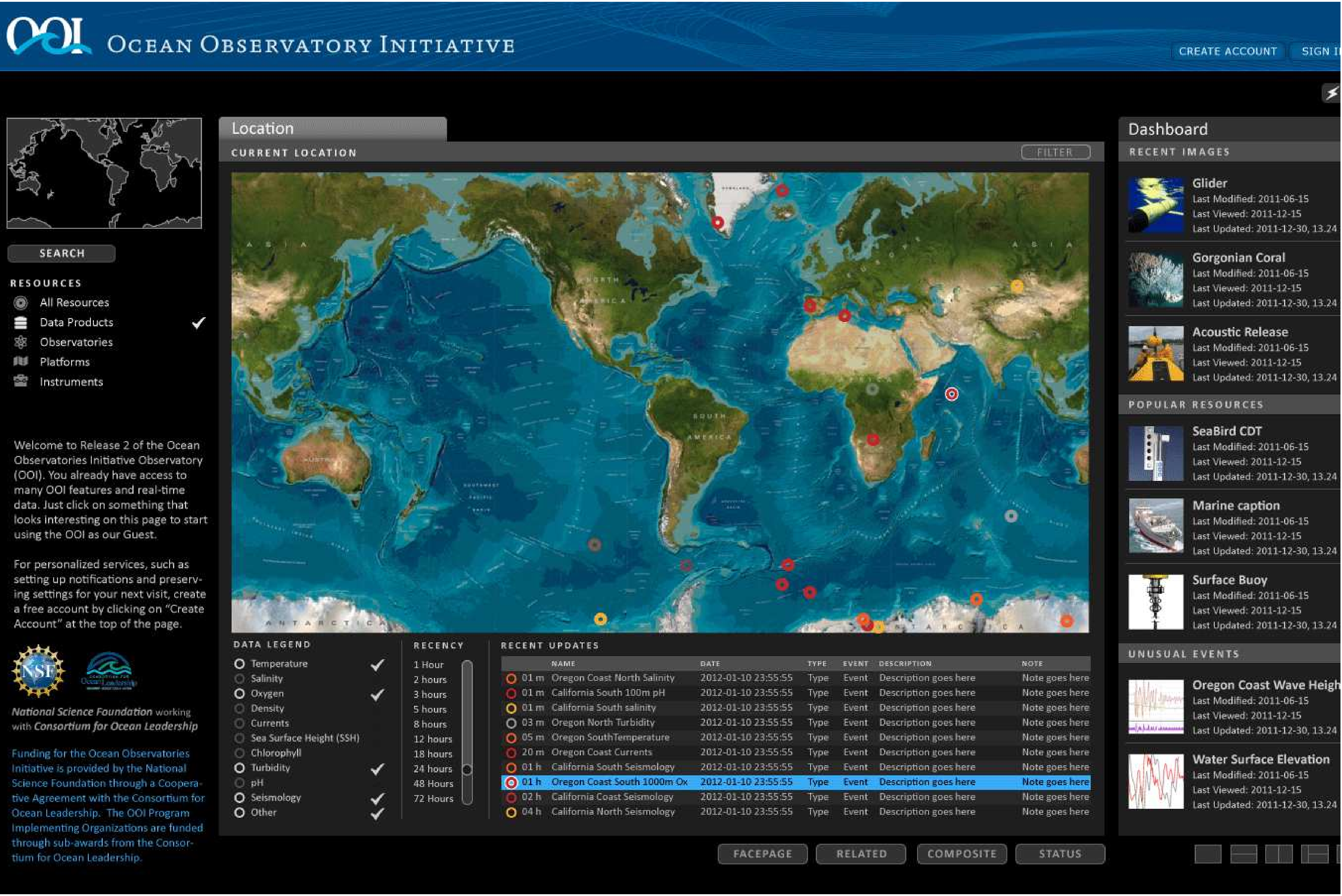


Figure 3: Observatory comprised of ships, aircraft and autonomous vehicles linked to assimilation modeling capabilities on shore


OCEAN OBSERVATORY INITIATIVE
CREATE ACCOUNT
SIGN I
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
RECENCY
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 SouthTemperature 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
FACEPAGE
RELATED
COMPOSITE
STATUS
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 Heigh
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, Congnizant, UNIFI, TrustCare
  - JBoss SCRIBBLE [ICDCIT'10, COB'12] and SAVARA projects
- High-performance computing

  Session Java [ECOOP'08,ECOOP'10,Coordination'11]

  $\Longrightarrow$ 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]

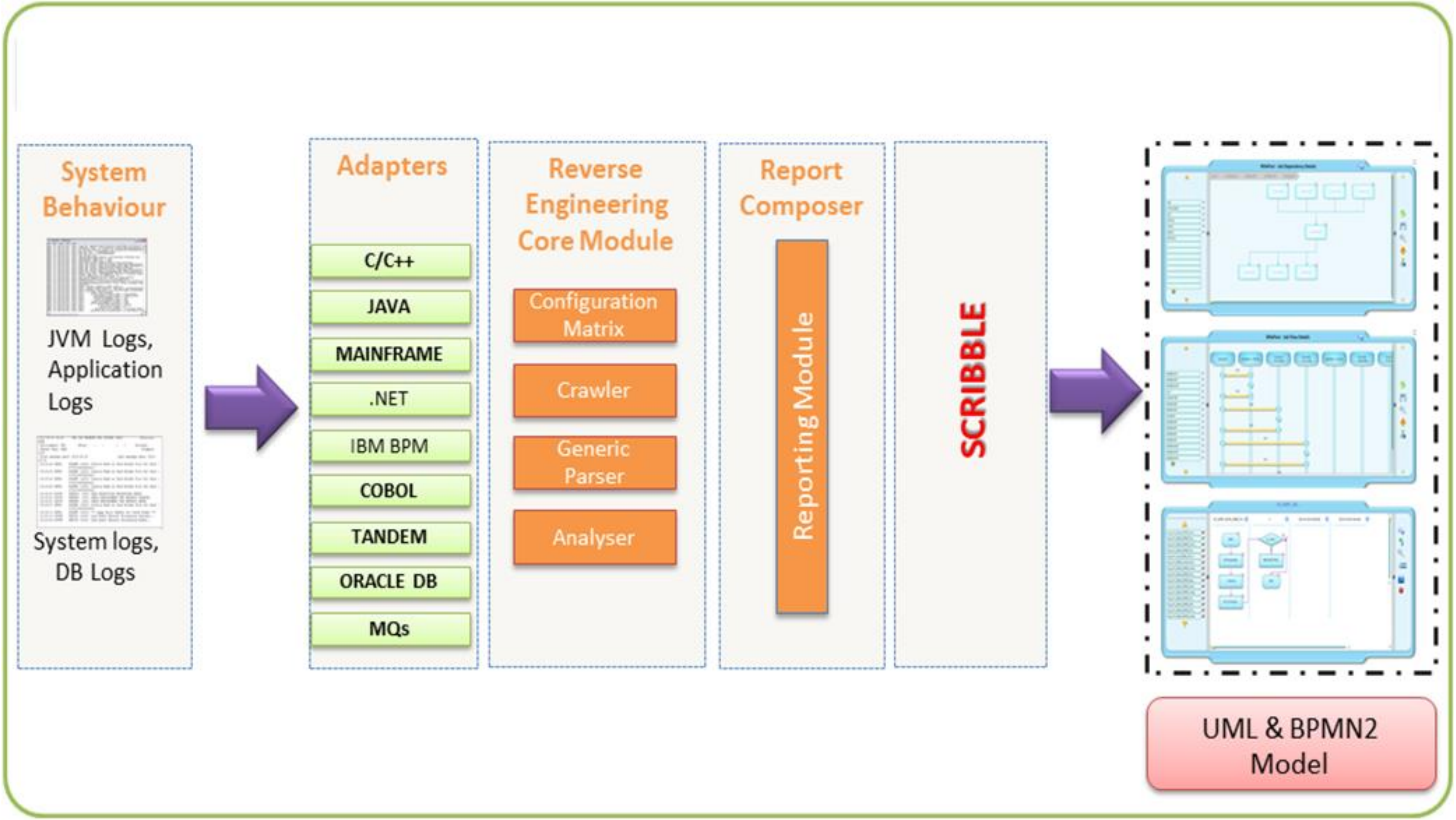
Timed Multiparty Session Types based on Communicating Timed Automata

```
global protocol TempAvg(
    role M, role W, role A) {
  [x@M: x=0; reset(x)]
  [x@W: 400≤ x<800]
  msg(task) from M to W;
  rec Loop{
     [x@W: x≤ 300800; reset(x)]
     [x@M: x==300800]
     msg(data) from W to M
     choice at M{
        [x@M: x==300800]
        [x@A: 301200≤ x; reset(x)]
        more(data) from M to A;
        [x@M: x==300800; reset(x)]
        [x@W: x==800; reset(x)]
        more(task) from M to W;
        continue Loop;}
     or{
        [x@M: x==300800]
        [x@A: 301200≤ x; reset(x)]
        stop(data) from M to A;
        [x@M: x==300800; reset(x)]
        [x@W: x==800; reset(x)]
        stop from M to W; }
     }
  }
```

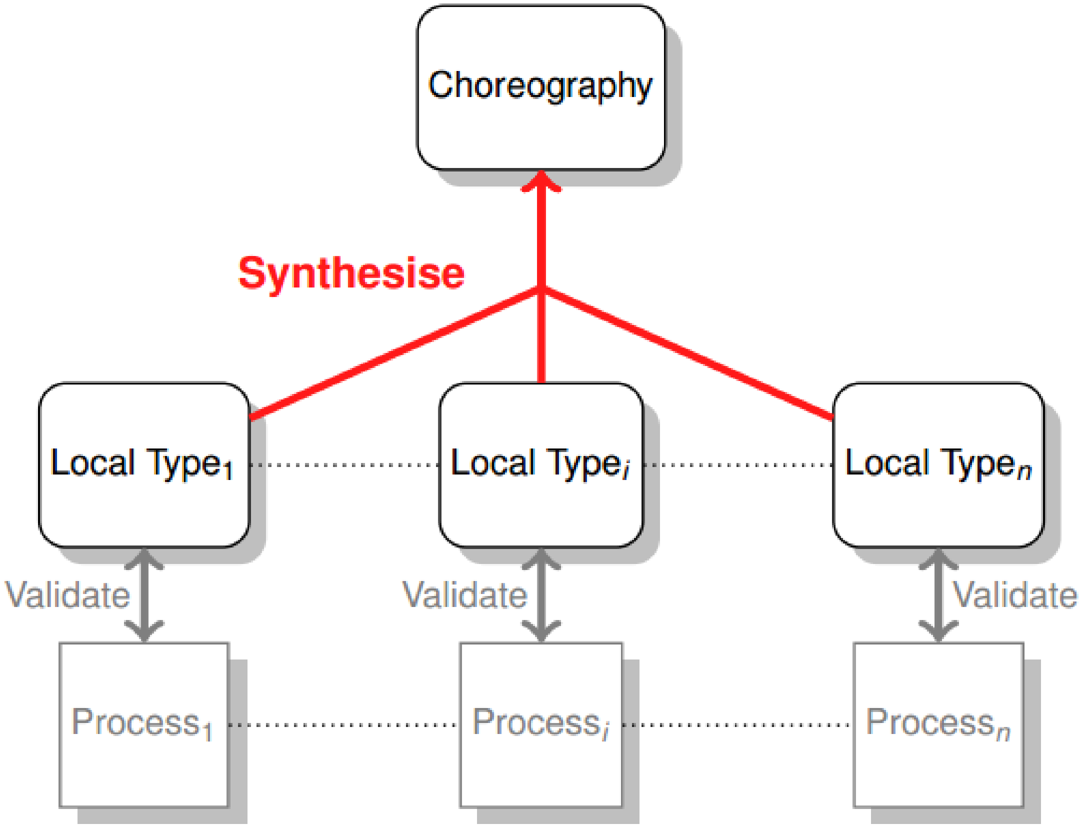
TGs	π-calculus	session CTAs
untimed	type safety, error-freedom, progress [13]	Sound, complete characterisation, progress [11]
timed	type safety (Thm 4.3) error-freedom (Thm 4.4)	Sound, complete characterisation, (Thm 5.6)
feasible, wait-free	progress (Thm 5.4, Thm 5.5)	progress (Thm 5.7)

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

Zero Deviation Life Cycle Platform

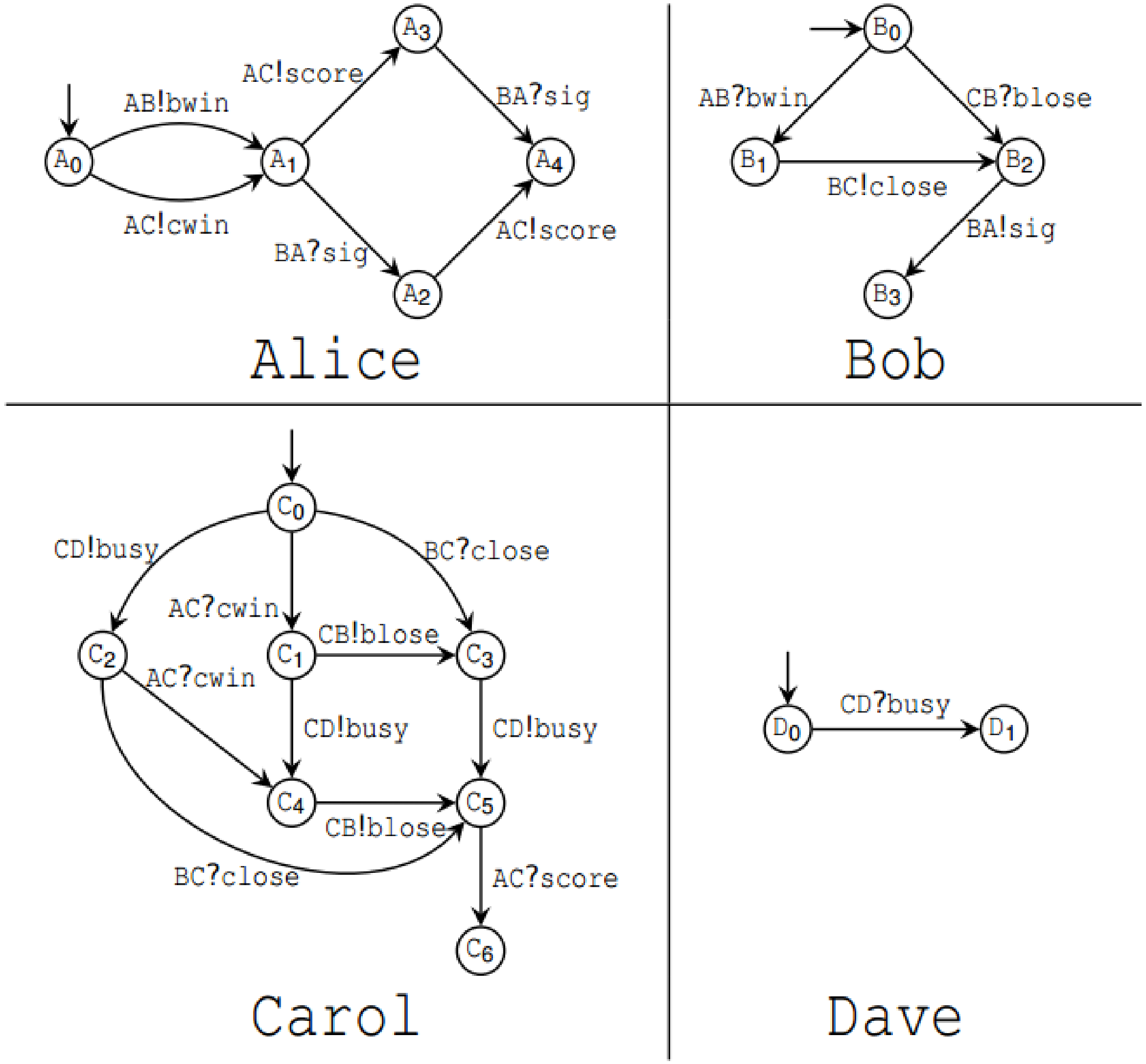


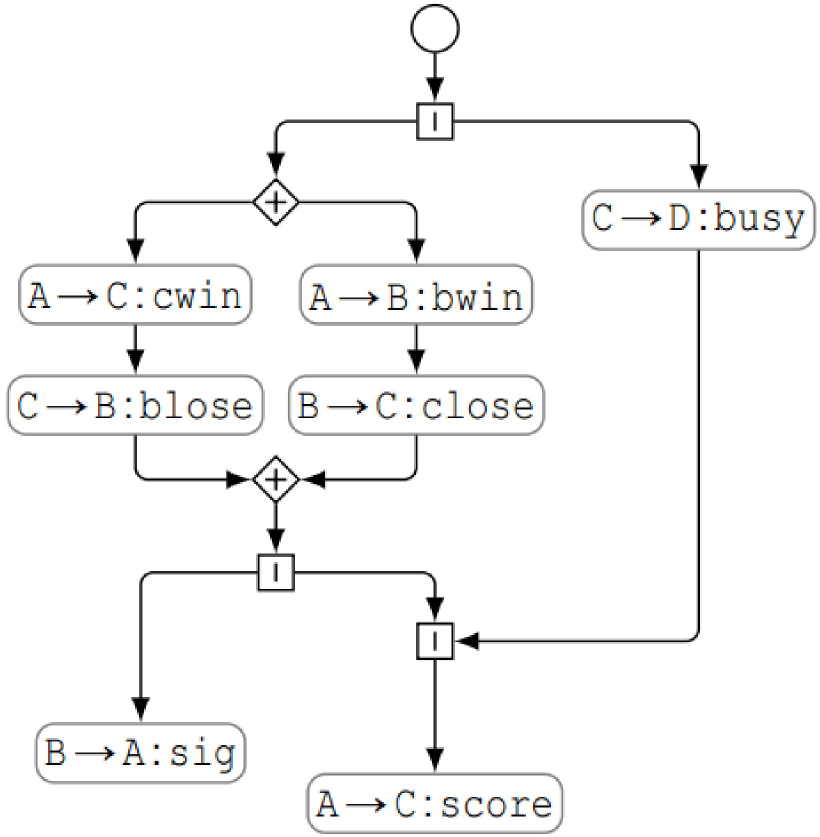
Synthesis of Graphical Choreographies 1/2



- 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*

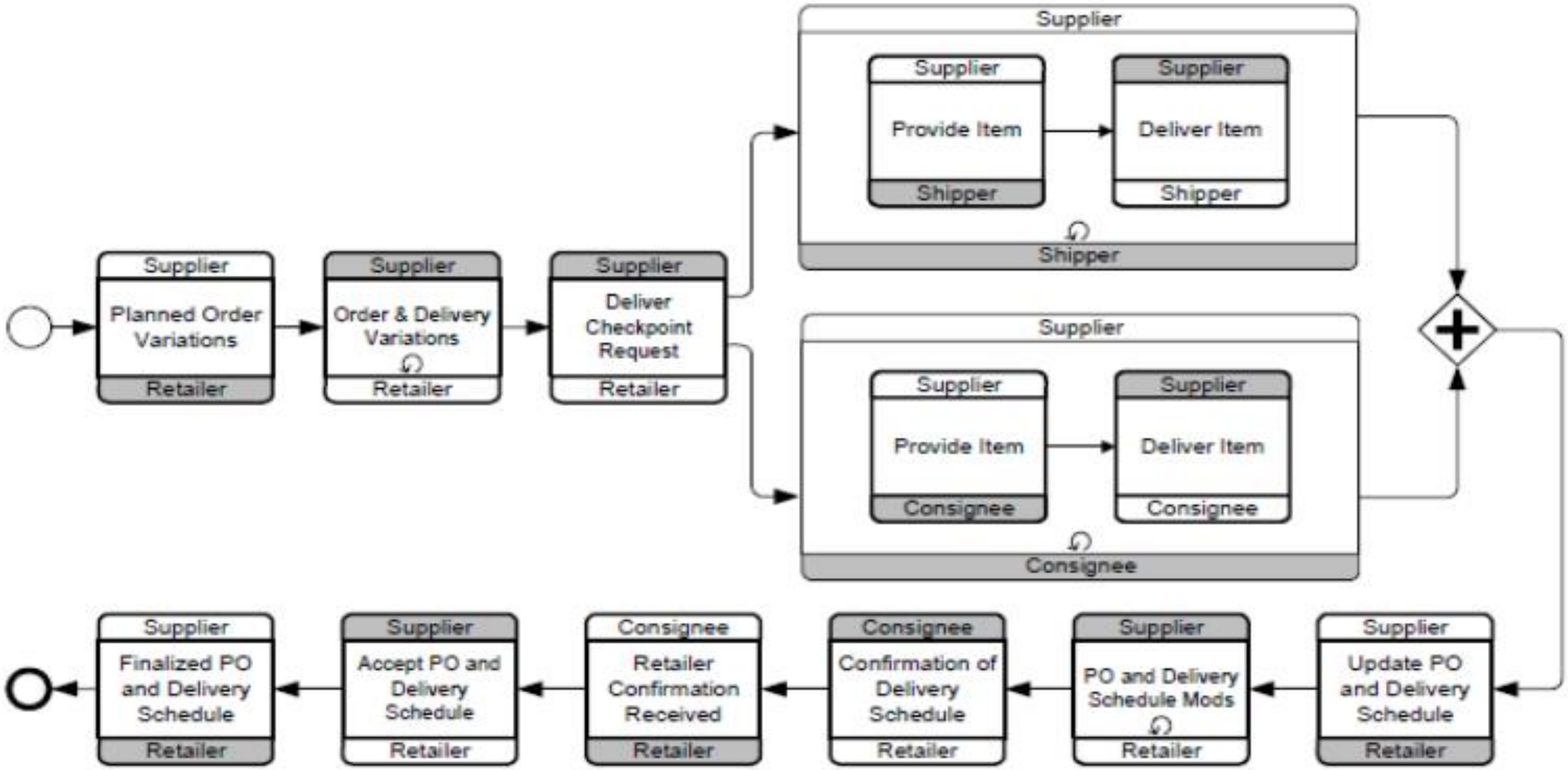
Synthesis of Graphical Choreographies 2/2





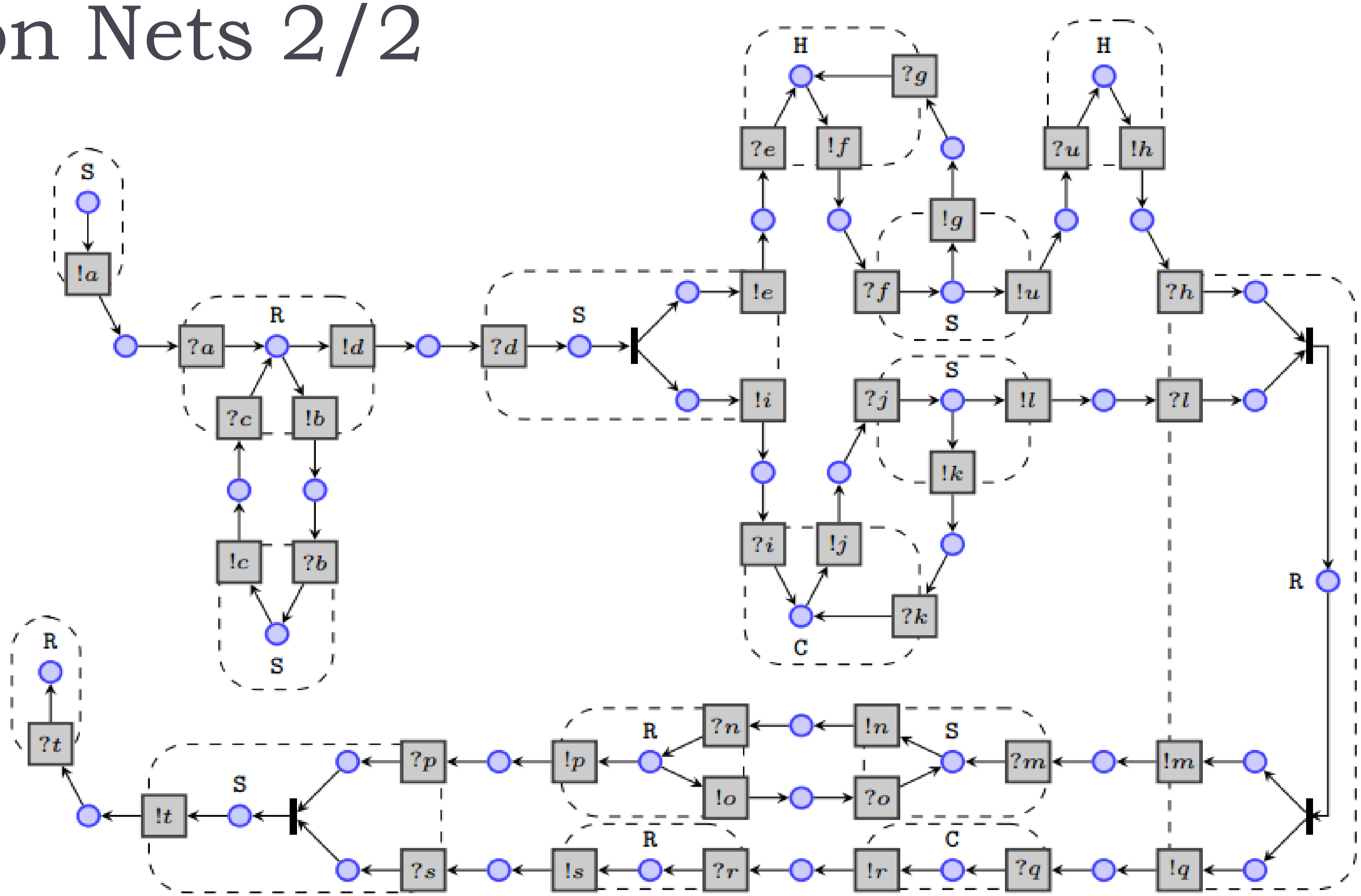
Session Nets 1/2

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

Session Nets 2/2



$$g = \{a \mapsto \texttt{Planned}, b \mapsto \texttt{Order}, c \mapsto \texttt{OrderEnd}, d \mapsto \texttt{Checkpoint}, e \mapsto \texttt{Provide},$$
$$f \mapsto \texttt{Deliver}, g \mapsto \texttt{Provide}, h \mapsto \texttt{Update}_1, i \mapsto \texttt{Provide}, j \mapsto \texttt{Deliver}, k \mapsto \texttt{Provide},$$
$$l \mapsto \texttt{Update}_2, m \mapsto \texttt{PO}, n \mapsto \texttt{POAck}, o \mapsto \texttt{PO}, p \mapsto \texttt{Accept}_1, q \mapsto \texttt{Confirmation},$$
$$r \mapsto \texttt{Retailer}, s \mapsto \texttt{Accept}_2, t \mapsto \texttt{Finalized}, u \mapsto \texttt{ProvideEnd}\}$$

Session Type Projects

- EPSRC *Conversation-Based Governance for Distributed Systems by Multiparty Session Types*
- SADEA EPSRC *Exploiting Parallelism through Type Transformations for Hybrid Manycore Systems*, with Vanderbauwhede, Scholz, Gay and Luk
- Programme Grant *From Data Types to Session Types: A Basis for Concurrency and Distribution*, with Wadler and Gay
- EU FP7 FETOpenX UpScale with de Boer (CWI), Clark, Wrigstad (Uppsala), Johnsen (Oslo) and Drossopoulou
- Pivotal Dynamic Assurance based on Multiparty Session Types
- Cognizant/Qualit-e EPSRC Knowledge Transfer Secondments

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.

Multiparty Session Type Theory

- Multiparty Asynchronous Session Types [POPL'08]
- Progress
  - Global Progress in Dynamically Interleaved Multiparty Sessions [CONCUR'08], [Math. Struct. Comp. Sci.]
  - Inference of Progress Typing [Coordination'13]
- Asynchronous Optimisations and Resource Analysis
  - Global Principal Typing in Partially Commutative Asynchronous Sessions [ESOP'09]
  - Higher-Order Pi-Calculus [TLCA'07,TLCA'09]
  - Buffered Communication Analysis in Distributed Multiparty Sessions [CONCUR'10]

- Logics
  - Design-by-Contract for Distributed Multiparty Interactions [CONCUR'10]
  - Specifying Stateful Asynchronous Properties for Distributed Programs [CONCUR'12]
  - Multiparty, Multi-session Logic [TGC'12]
- Extensions of Multiparty Session Types
  - Multiparty Symmetric Sum Types [Express'10]
  - Parameterised Multiparty Session Types [FoSSaCs'10, LMCS]
  - Global Escape in Multiparty Sessions [FSTTCS'10] [Math. Struct. Comp. Sci.]
  - Dynamic Multirole Session Types [POPL'11]
  - Nested Multiparty Sessions [CONCUR'12]

- Dynamic Monitoring
  - Asynchronous Distributed Monitoring for Multiparty Session Enforcement [TGC'11]
  - Monitoring Networks through Multiparty Sessions [FORTE'13]
- Automata Theories
  - Multiparty Session Automata [ESOP'12]
  - Synthesis in Communicating Automata [ICALP'13]
- Typed Behavioural Theories
  - On Asynchronous Eventful Session Semantics [FORTE'11] [Math. Struct. Comp. Sci.]
  - Governed Session Semantics [CONCUR'13]
- Choreography Languages
  - Compositional Choreographies [CONCUR'13]