Monitoring Protocol Conformance with Multiparty Session Types and OpenTelemetry

David Castro-Perez, Francisco Ferreira, Nobuko Yoshida, Fangyi Zhou

Imperial College London

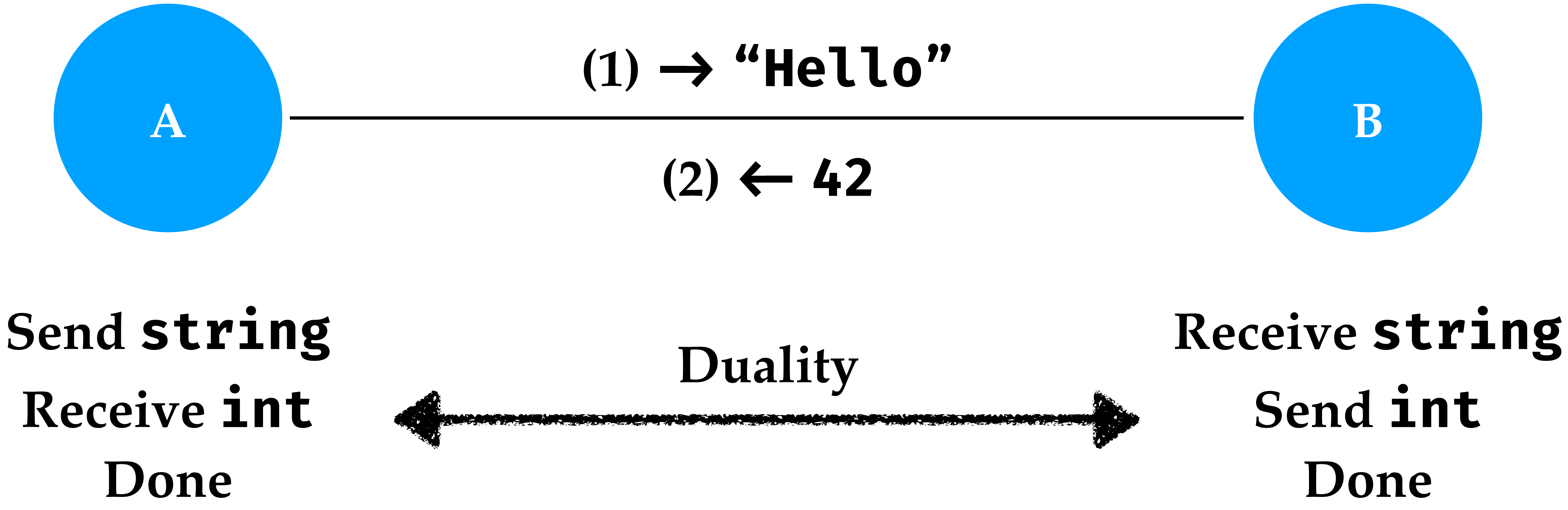
Telemetry Working Group - 16 March 2021

Quick Primer on Session Types

Concurrency

Shared Memory

Message Passing



A

B

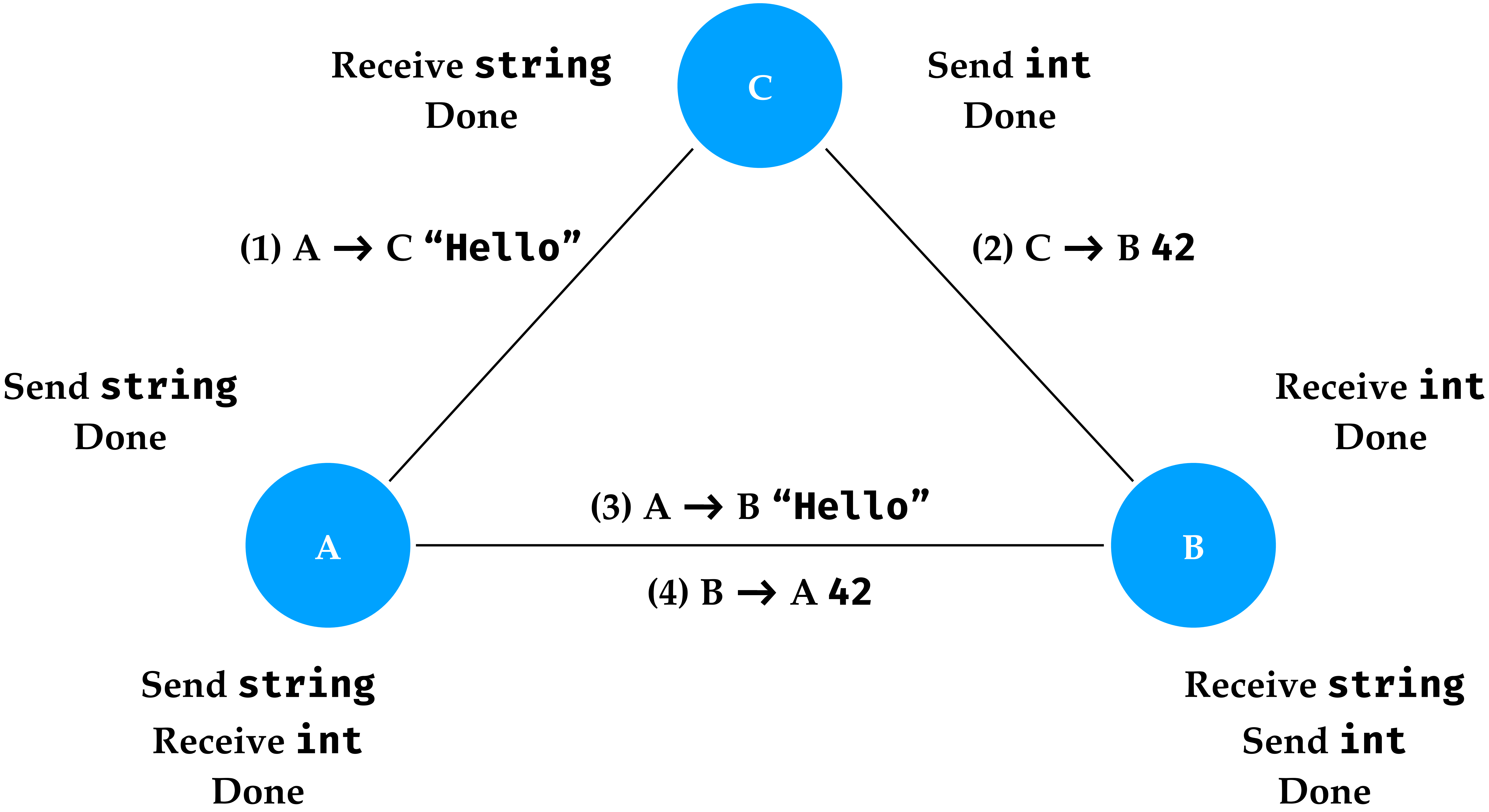
(1) → “`Hello`”

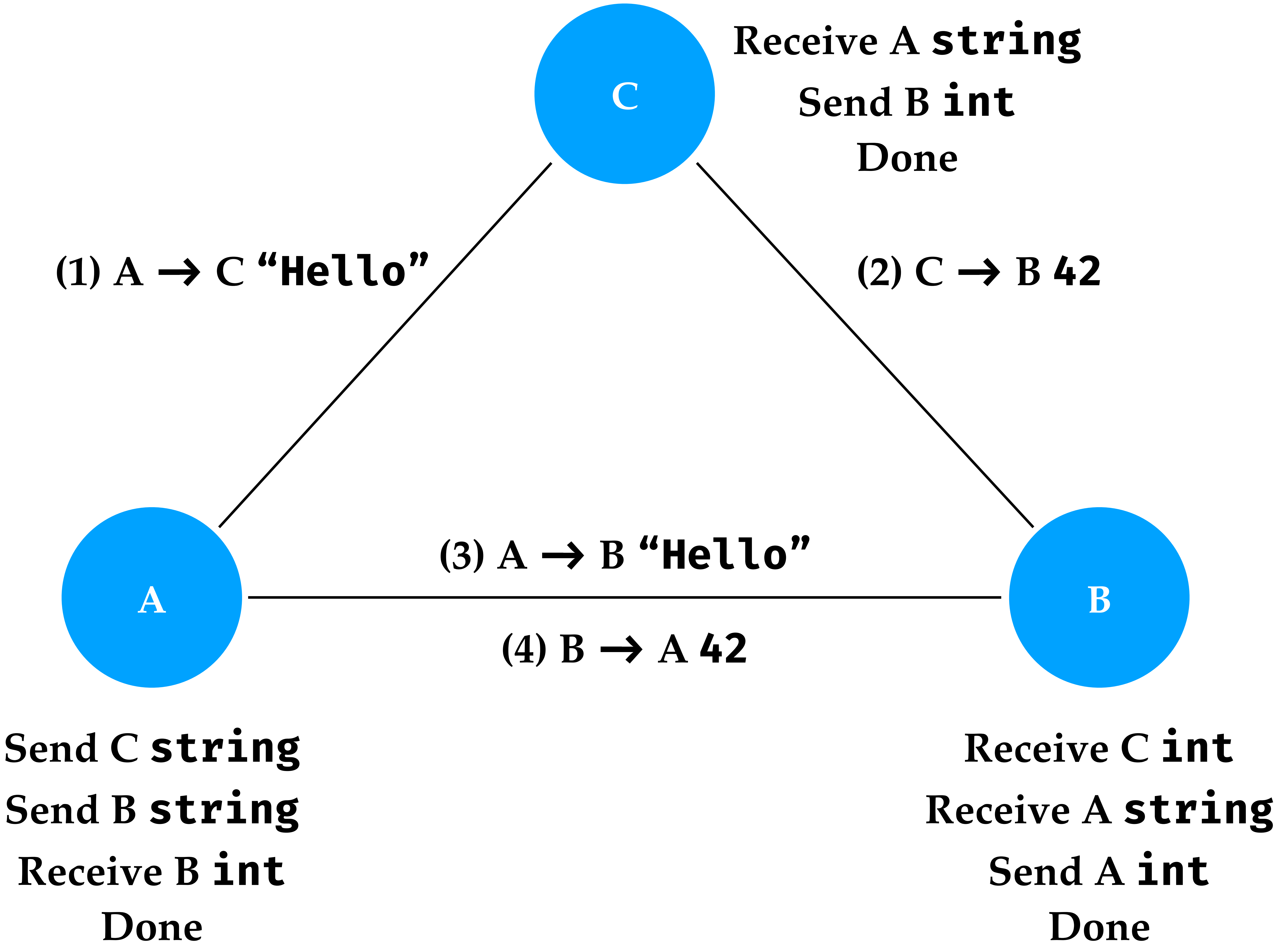
(2) ← `42`

Send `string`
Receive `int`
Done

Duality

Receive `string`
Send `int`
Done


Receive string
Done
C
Send int
Done
(1) A → C "Hello"
(2) C → B 42
Send string
Done
Receive int
Done
(3) A → B "Hello"
A
B
(4) B → A 42
Send string
Receive int
Done
Receive string
Send int
Done



C

Receive A `string`
Send B `int`
Done

(1) A → C “`Hello`”

(2) C → B `42`

A

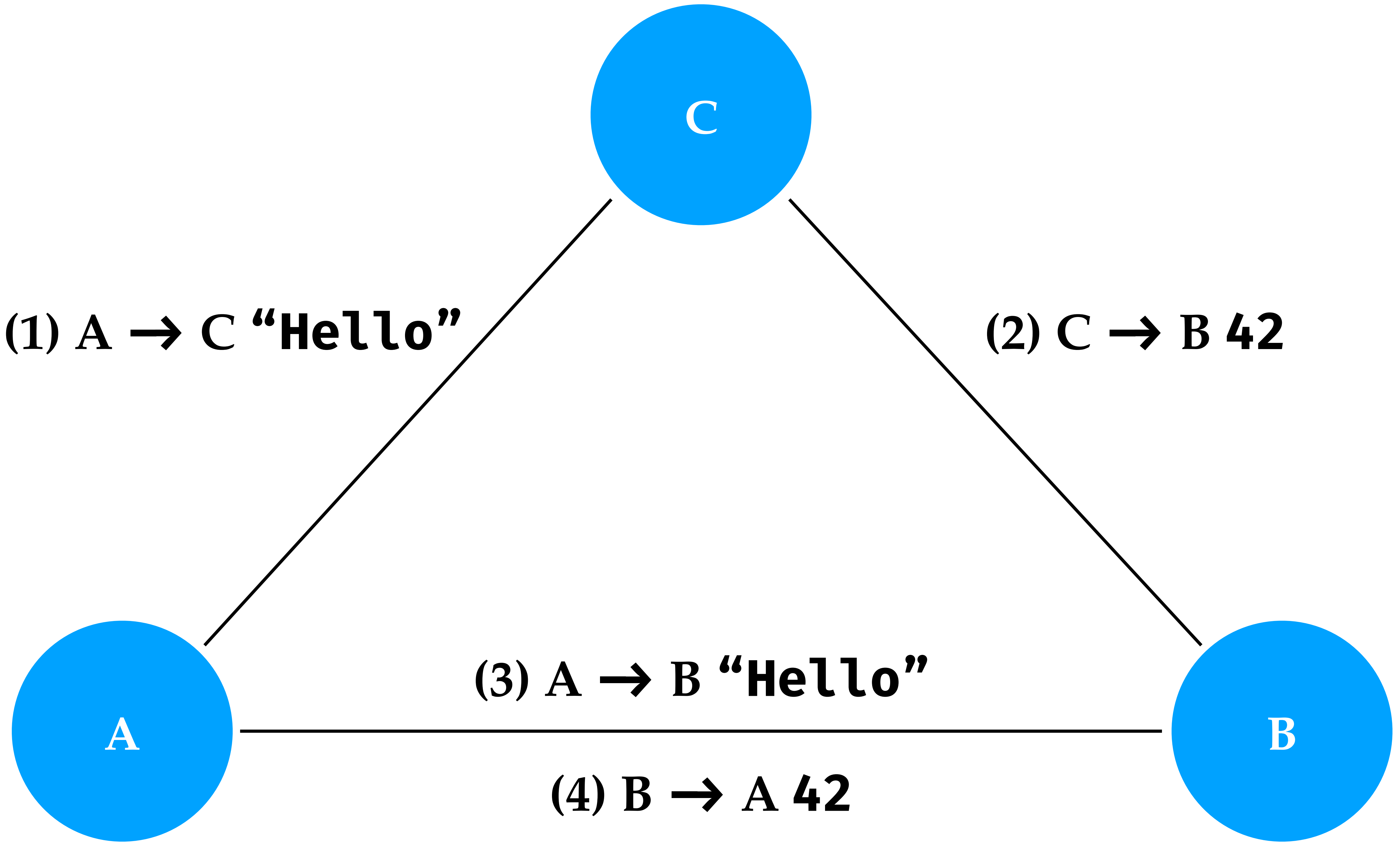
(3) A → B “`Hello`”

(4) B → A `42`

B

Send C `string`
Send B `string`
Receive B `int`
Done

Receive C `int`
Receive A `string`
Send A `int`
Done



Global Protocol

```
string from A to C;
int    from C to B;
string from A to B;
int    from B to A;
```

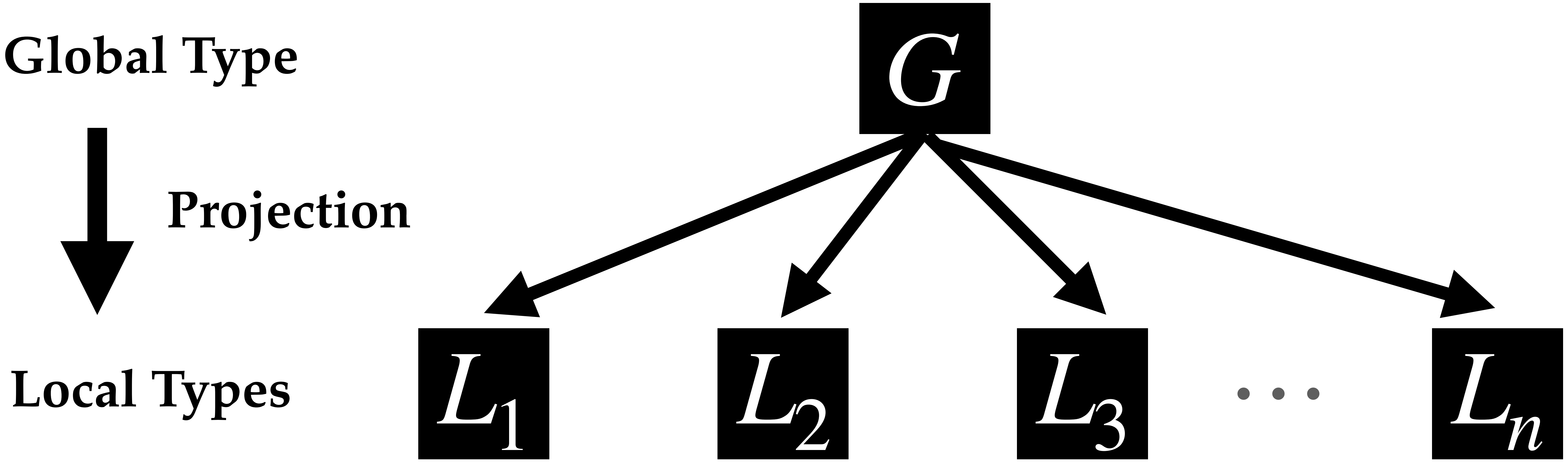
Multiparty Session Types (MPST)

A typing discipline for message passing concurrency

Global Type $\boxed{G}$

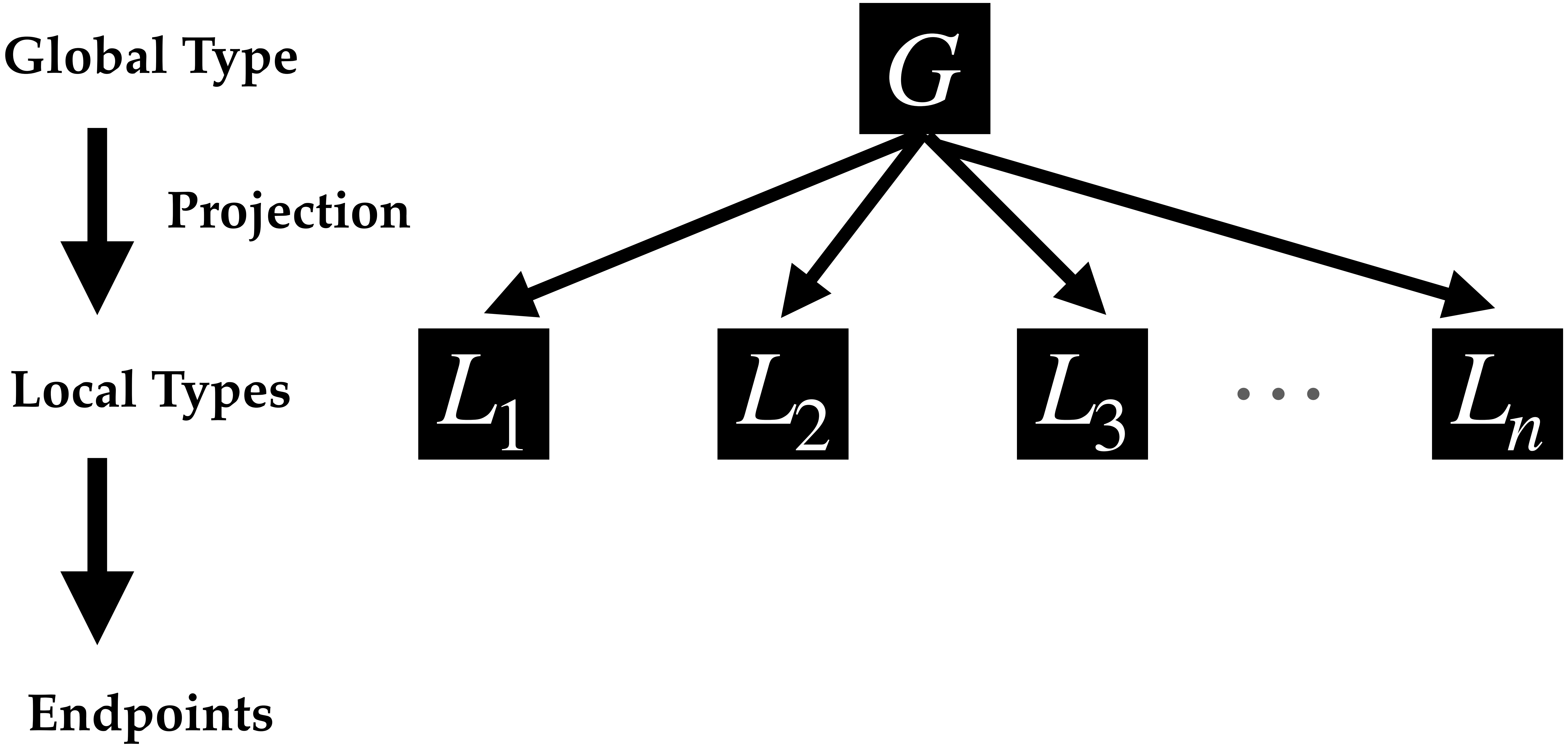
Multiparty Session Types (MPST)

A typing discipline for message passing concurrency



Multiparty Session Types (MPST)

A typing discipline for message passing concurrency



Global Type

Projection

Local Types

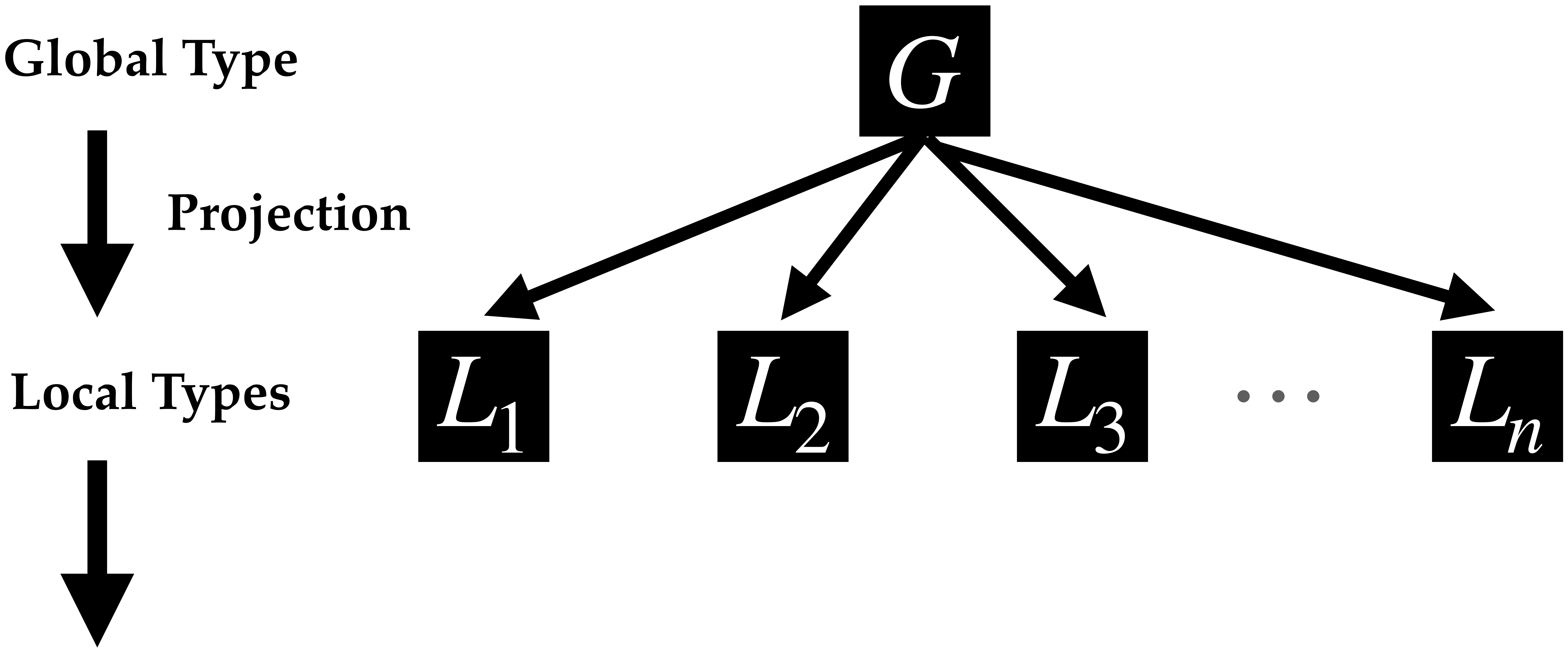
Endpoints

G

L_1 L_2 L_3 $\cdots$ L_n

Multiparty Session Types (MPST)

A typing discipline for message passing concurrency



Global Type

Projection

Local Types

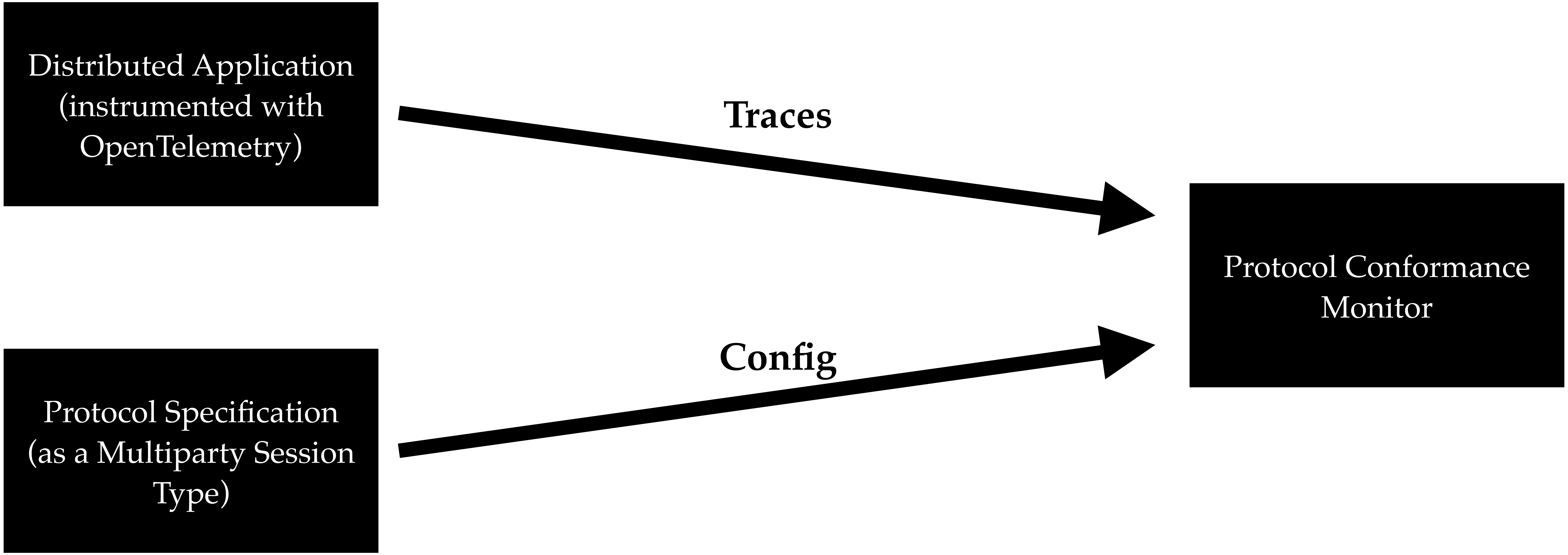
Endpoints

Free from deadlocks or communication mismatches

Goal of this Work

- Monitor protocol conformance for existing distributed system
  - Monitor process behaviour via **tracing** (using OpenTelemetry)
  - **Instrument** existing applications, instead of asking developers to use generated APIs

Workflow



Distributed Application (instrumented with OpenTelemetry)

Traces

Protocol Specification (as a Multiparty Session Type)

Config

Protocol Conformance Monitor

Demo of an (Early) Prototype

Next Steps…

- Develop a more expressive semantic model for global protocols
- Integrate with existing distributed systems

Thank you!

https://github.com/fangyi-zhou/mpst-tracing

Read our blog post at https://tinyurl.com/opentelemetry-mpst