

Why do we actually need the Pi-Calculus for Business Process Management?

Frank Puhlmann

Business Process Technology Group
Hasso-Plattner-Institute
Potsdam, Germany

Outline

- Shifting Theories:
 - From Lambda-Calculus to Pi-Calculus
- Shifting Focus:
 - From Workflow to BPM
- Conclusion

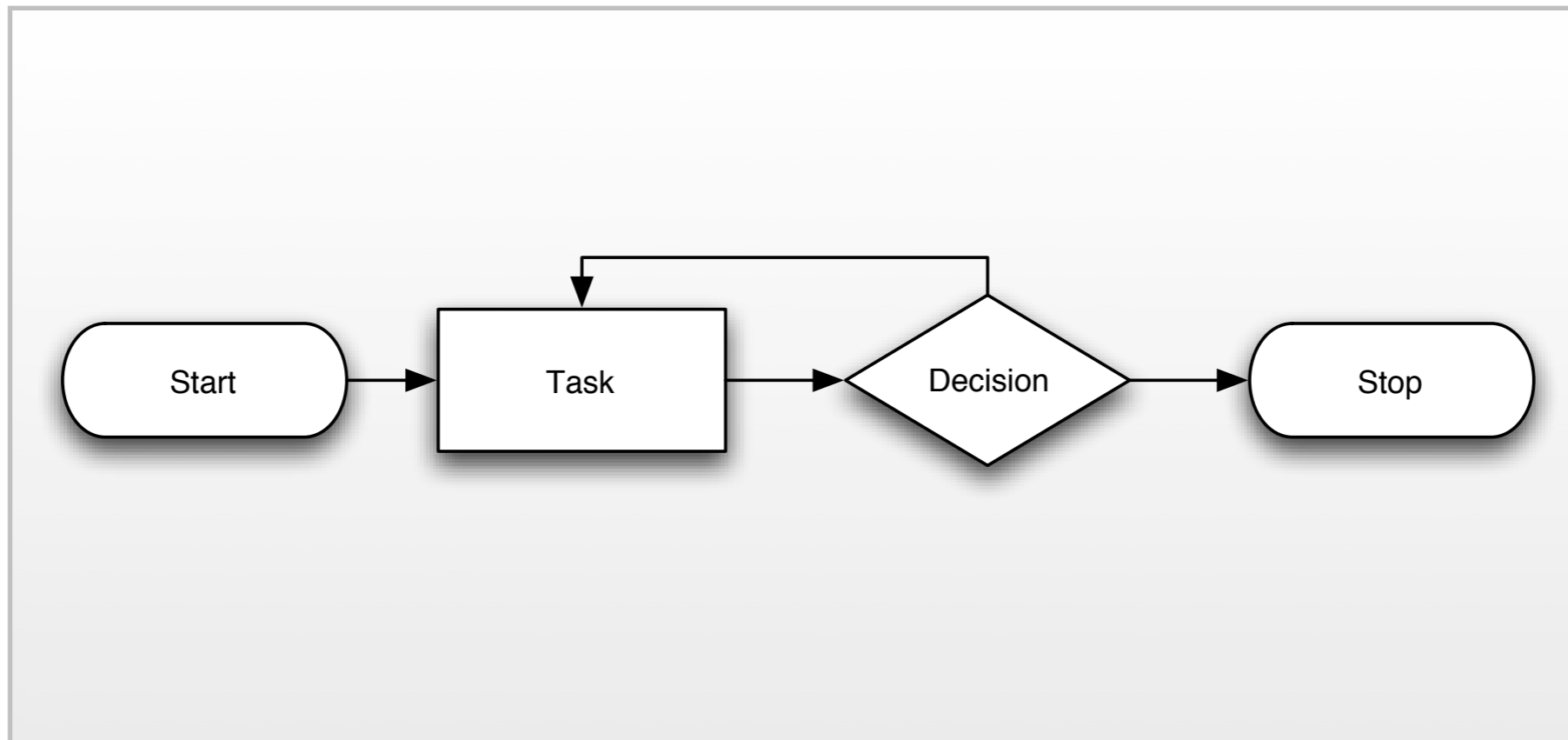
Shifting Theories

Shifts in Theoretical Foundations

- From: Sequential systems
 - Lambda-Calculus (Church, Kleene, ≈ 1930)
- Over: Parallel systems
 - Petri nets (Petri, ≈ 1960)
- To: Mobile systems
 - Pi-Calculus (Milner, Parrow, Walker ≈ 1990)

The Lambda-Calculus

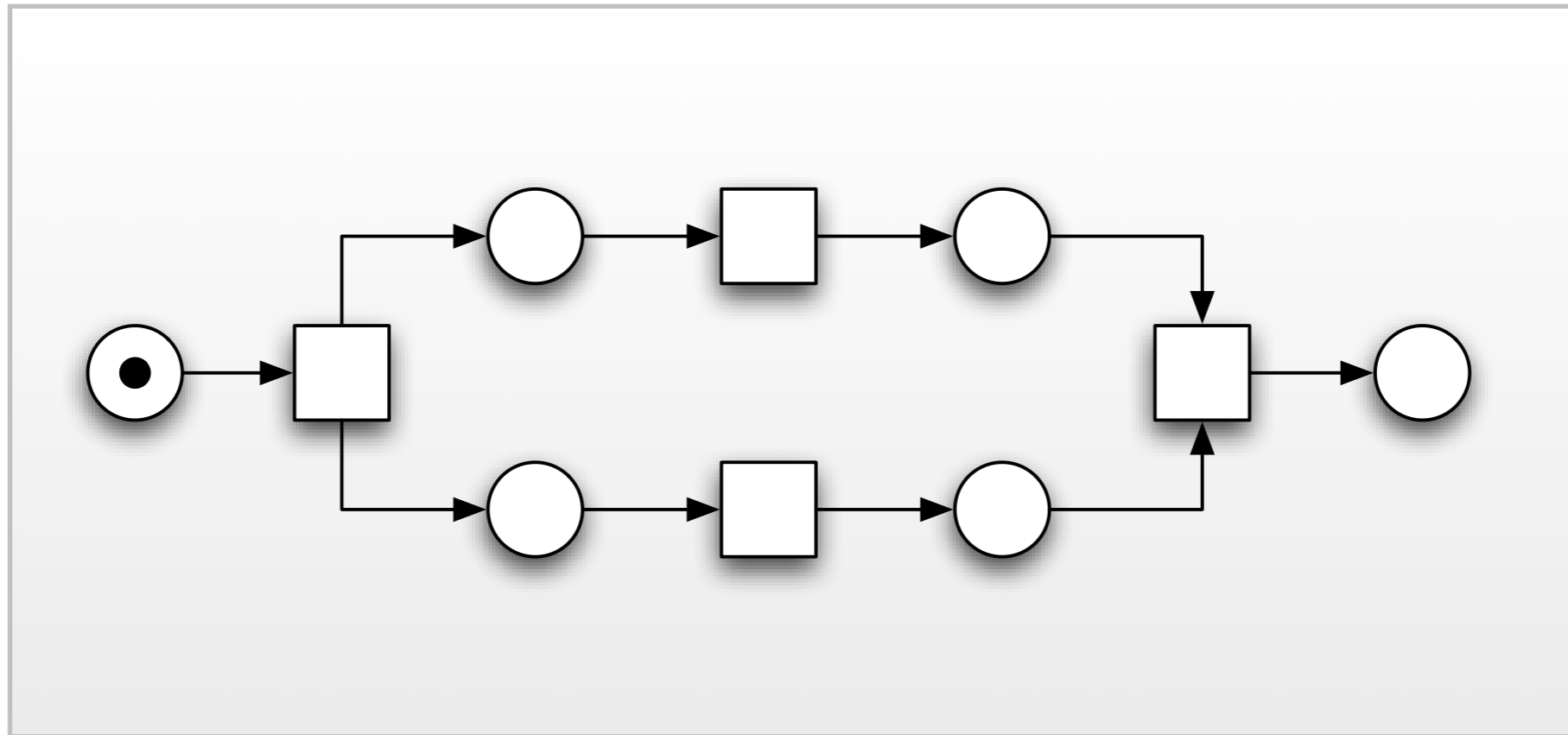
- Defined to investigate the definition of functions which are used for sequential computing
 - Precise definition of a computable function
 - Recursion
- Algebra: Compositional Structure
- Smallest universal programming language



Sequential System

Petri nets

- Business processes require parallelism
 - Split, Joins
 - Dependencies
- Petri nets build a foundation for Workflow
 - Explicit states and structure
 - Strong visualization



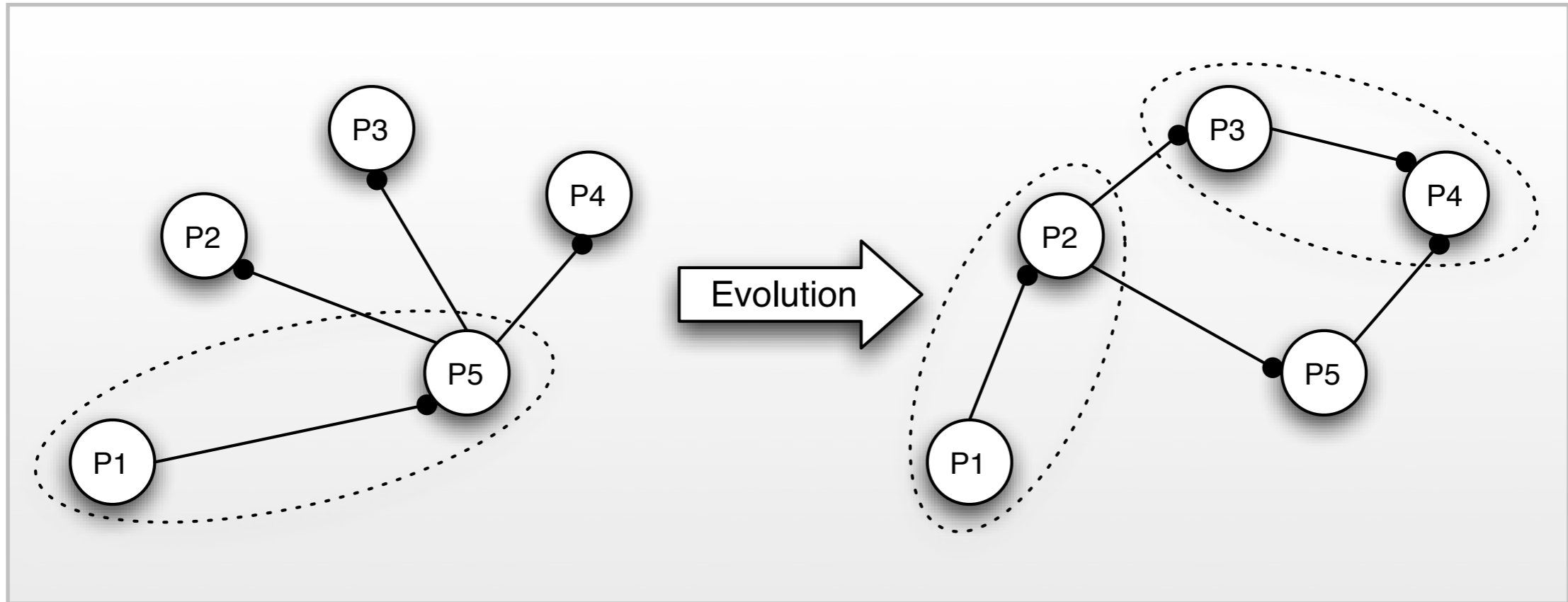
Parallel System

Petri net drawbacks

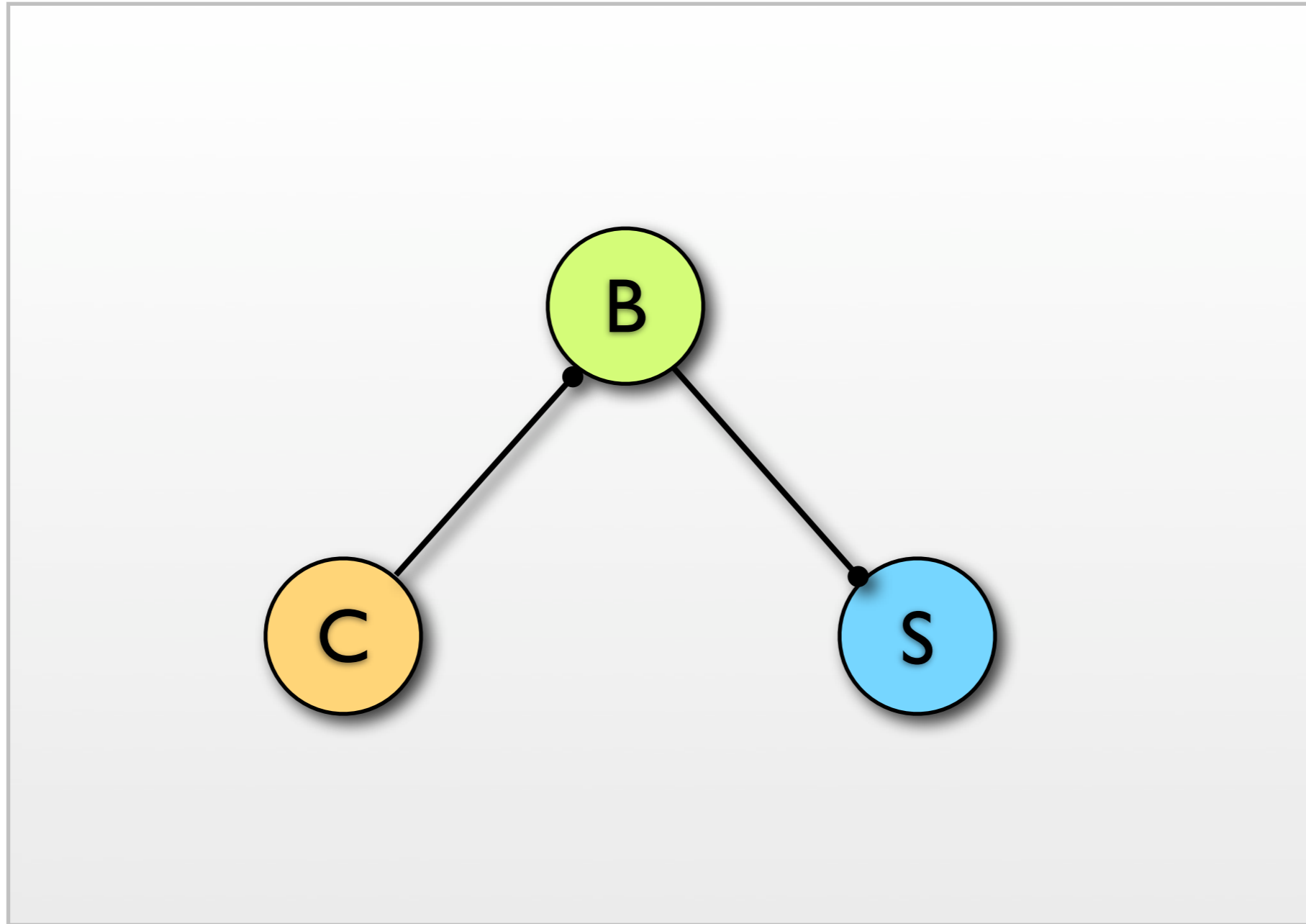
- Good and Bad: Static structure
- No advanced composition
- Regarding behavioral workflow patterns:
 - Excellent support for basic tasks
 - Poor support for advanced tasks

The Pi-Calculus

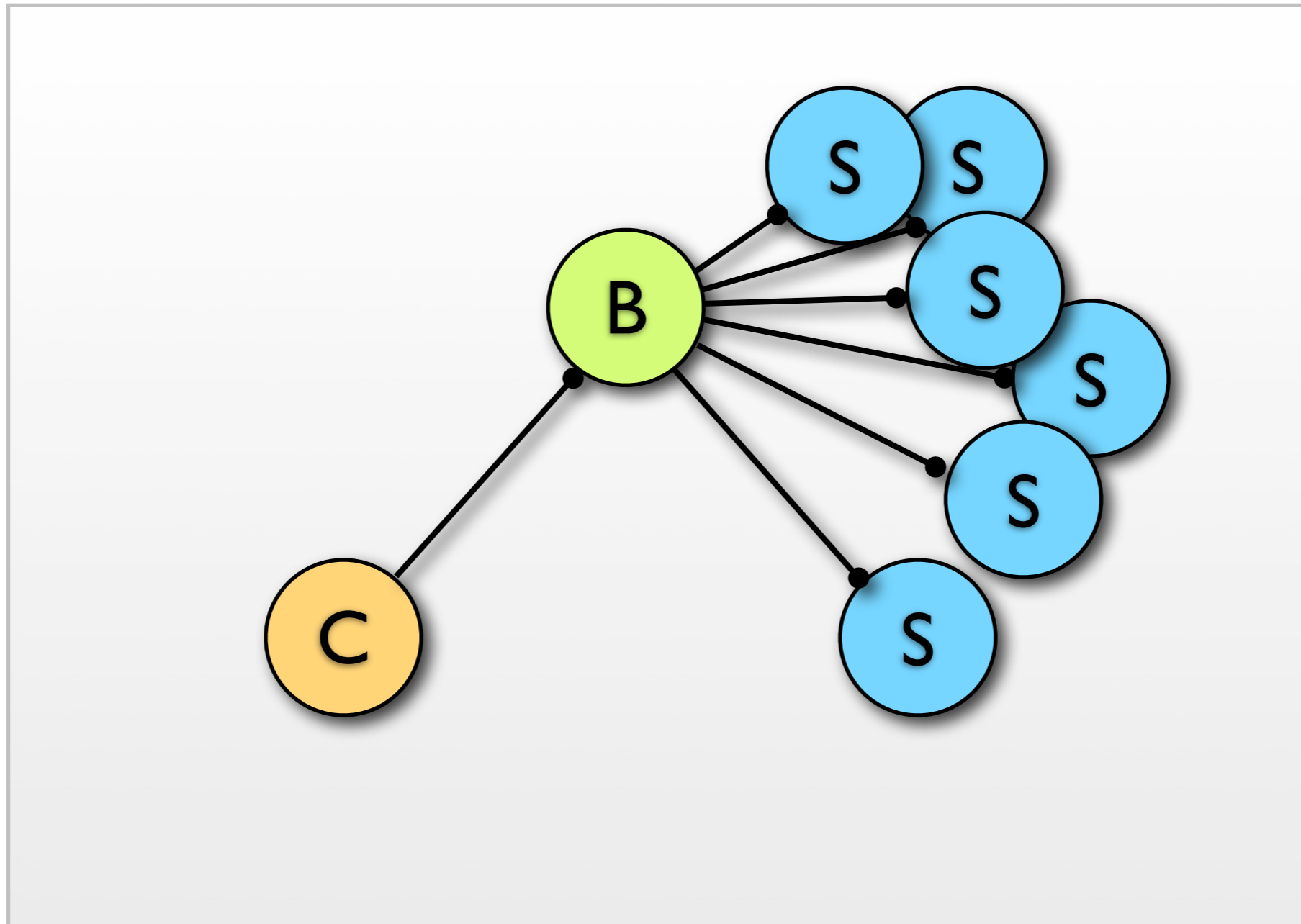
- Describes mobile systems
 - Processes interacting by
 - Names with agile scopes and connections
- Is an algebra



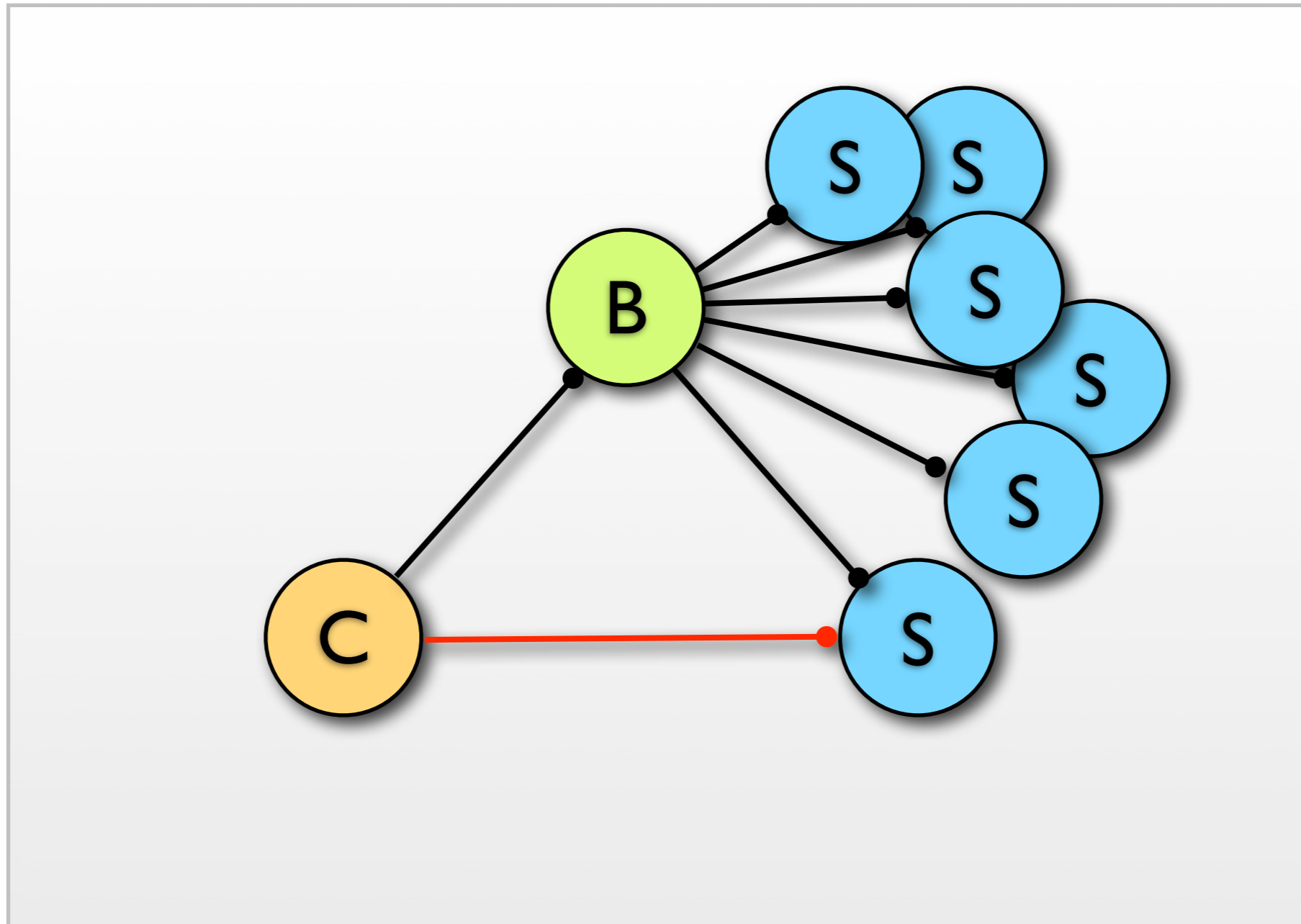
Mobile System



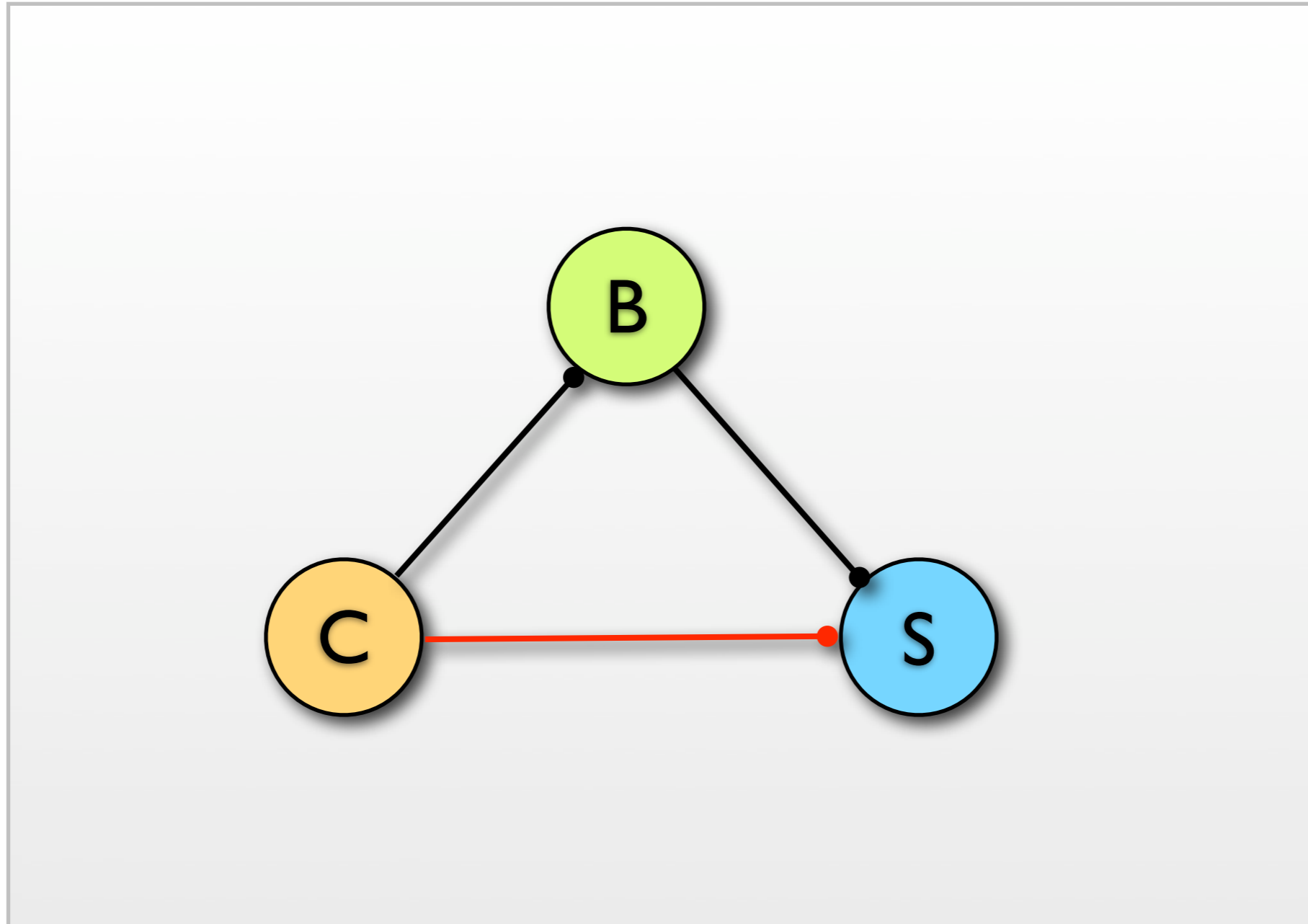
Link Passing Mobility



Link Passing Mobility



Link Passing Mobility



Link Passing Mobility

The Pi-Calculus Advantage

- Overcomes the limitations of static structures
- Has the pros and cons of an algebra
- Supports all behavioral workflow patterns

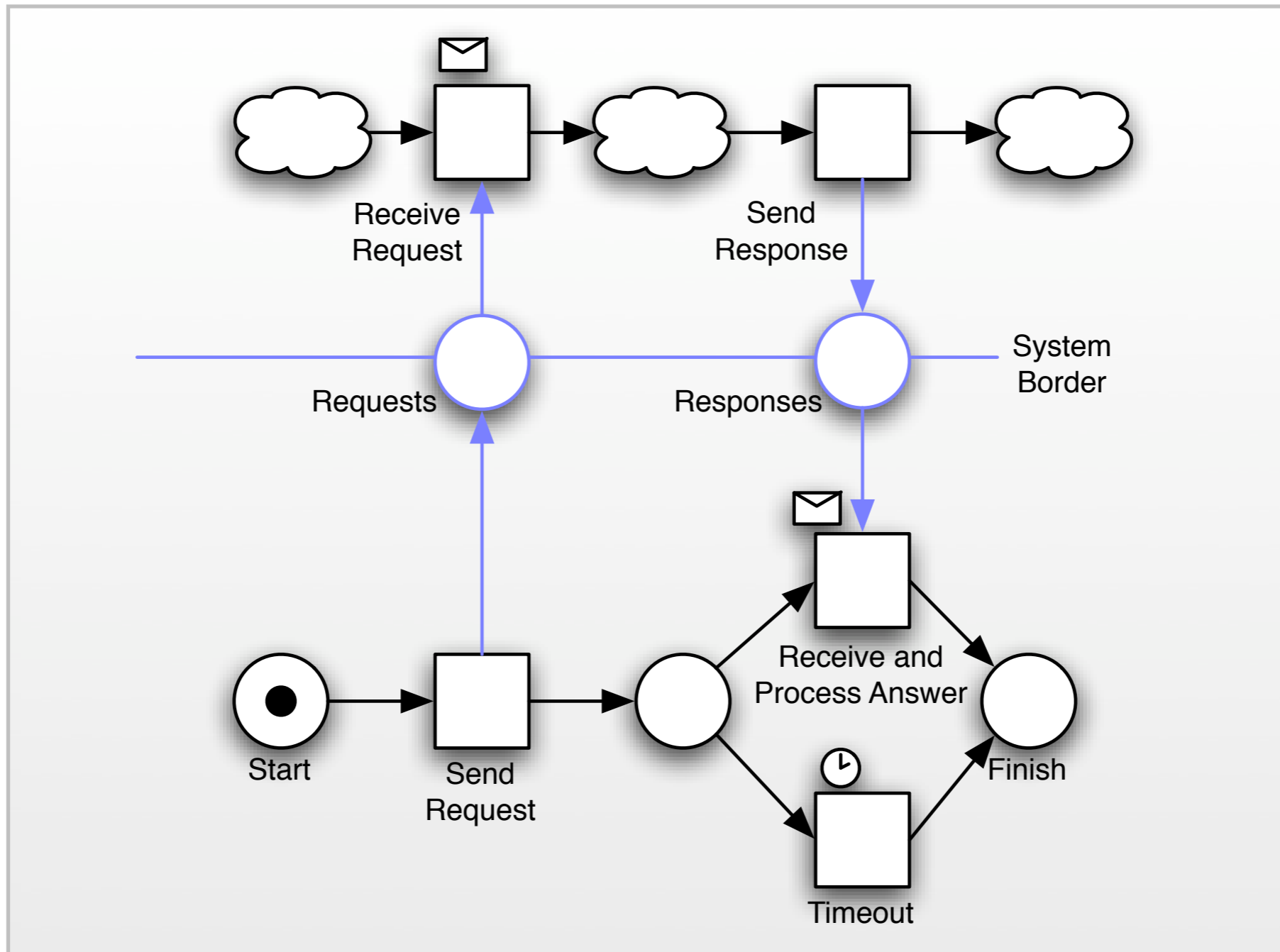
The Shifting Focus

The Shifting Focus: From Workflow to BPM

- Workflow Management:
 - Computerized execution of business processes
 - Company, department internal
- Business Process Management:
 - Extension to interaction and collaboration between different business processes
 - Cross company/enterprise wide

BPM Shift I: From State to Message-based Systems

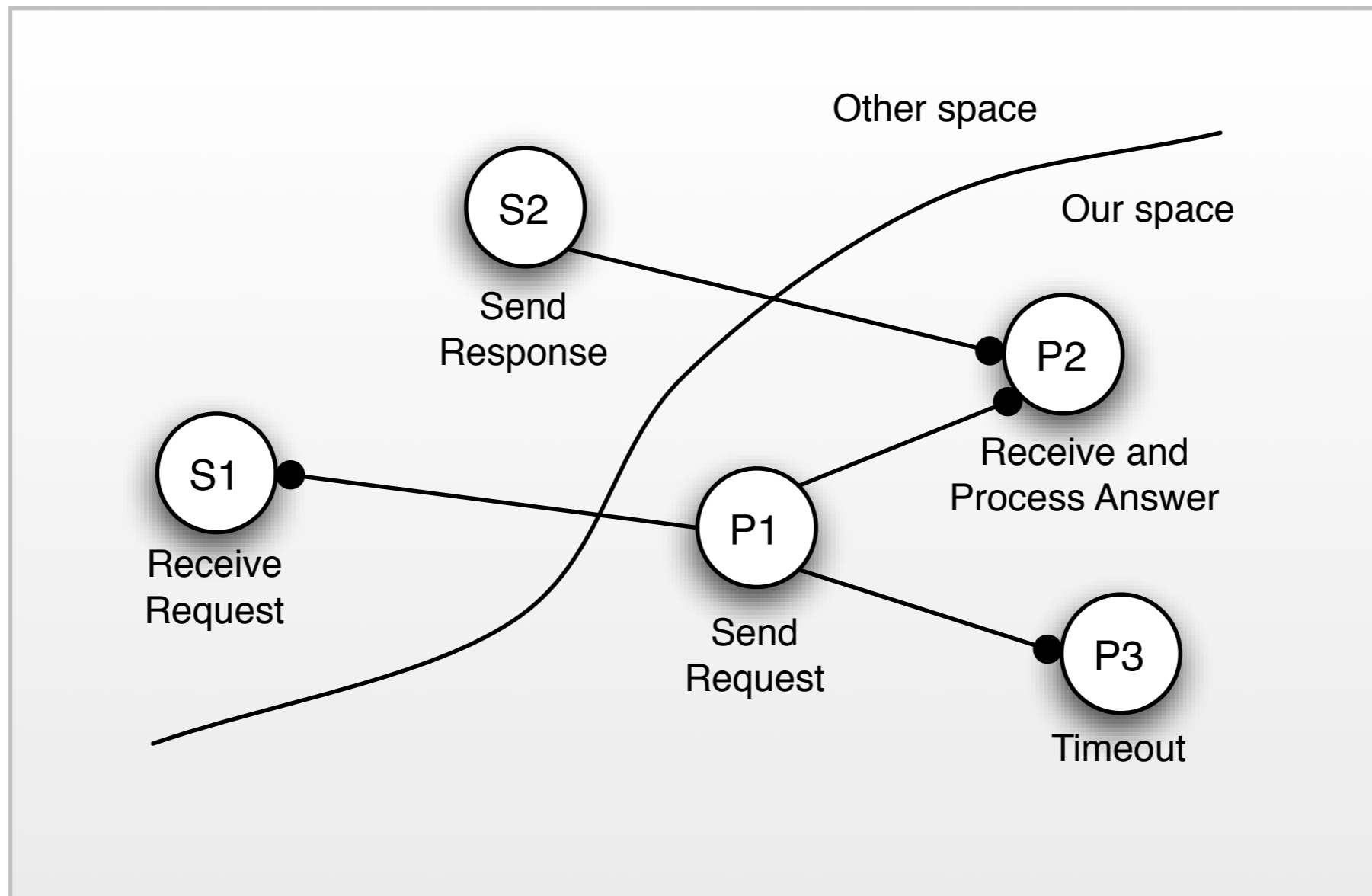
- Traditional: Hard-wired, state-based systems
 - e.g. Workflow nets, Activity Diagrams, BPMN (Token/Place concept)
- Today: Agile, inter-organizational business processes
- “Hard to manage” with states in contrast to message-based systems



State-based Interaction

Message-based Systems

- No explicit state description
 - No static connections
- Each task is mapped to a service:
 - Each task has pre- and postconditions (i.e. in- and outgoing messages)
 - All tasks are “swimming” inside a service-oriented environment



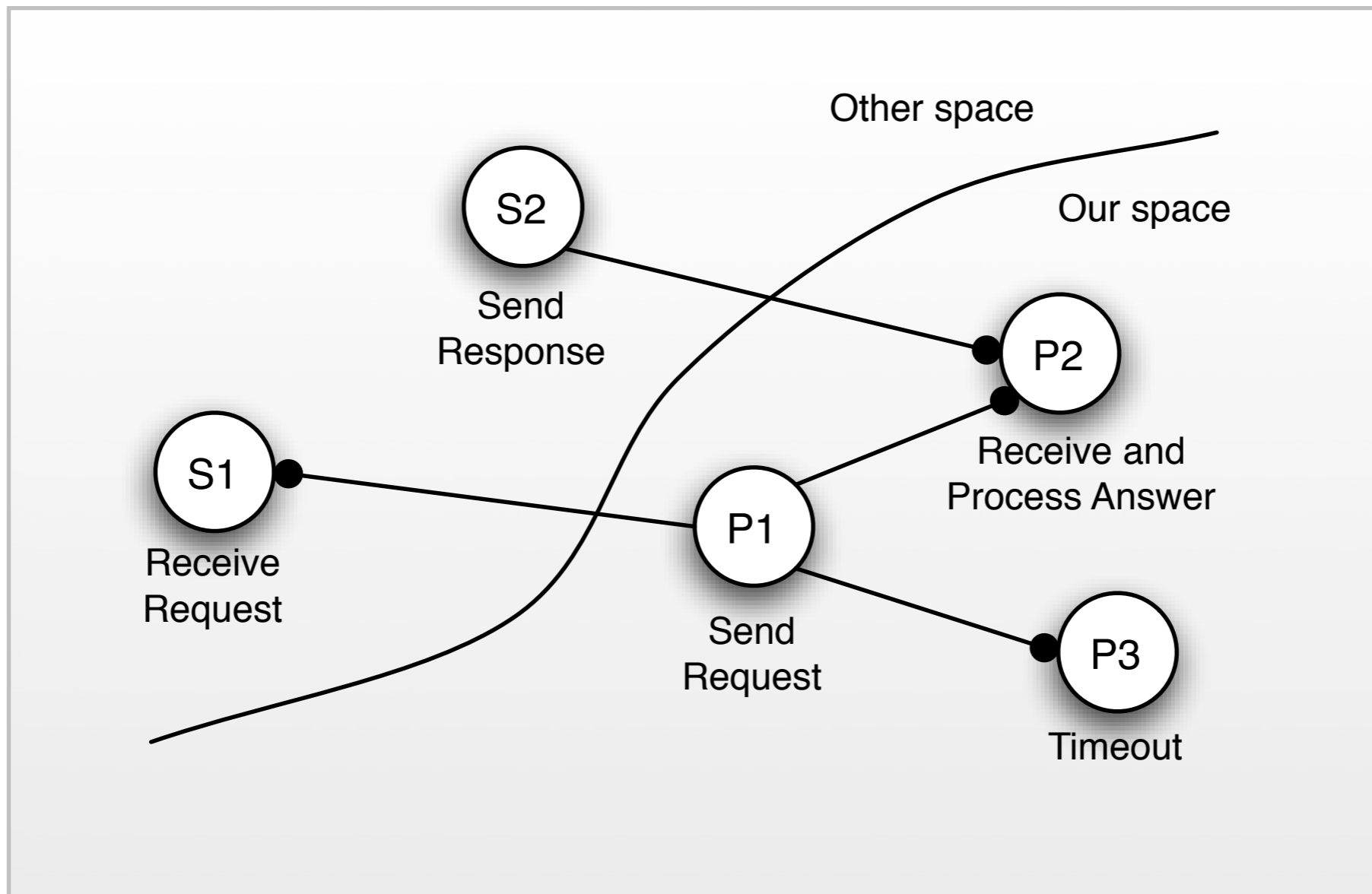
Message-based System

Reason 1:

- Mobile systems are based on the idea of interaction by messages instead of state transitions
- Support intra- as well as inter-organizational business processes

BPM Shift II: From Central Engines to Distributed Services

- Follows direct from the last example:
- No more centralized engine as for intra-organizational “workflow”
- Instead distributed services of different granularity



Distributed Services

Reason II:

- Mobile systems support advanced composition and visibility of their parts
- Support distribution and the service-oriented idea for BPM

BPM Shift III: From Closed to Open Environments

- The environment where processes are executed is shifting strongly from closed to open, which means:
 - Less accessibility
 - More uncertainty
 - Constant change regardless of us
 - Number of possible interaction partners rises fast

Issues regarding Open Environments

- Constant change requires dynamic adaption
- Flexible discovery and integration
- Leading to more agile interaction

Reasons III:

- Mobile systems describe dynamic process structures
- Support “flexibility” regarding discovery and interaction for BPM

Conclusion:

Motivation in a Nutshell

- Mobile systems support advanced key concepts of BPM:
 - Messages
 - Distribution
 - Agile Interactions
- The Pi-Calculus is a process algebra for mobile systems

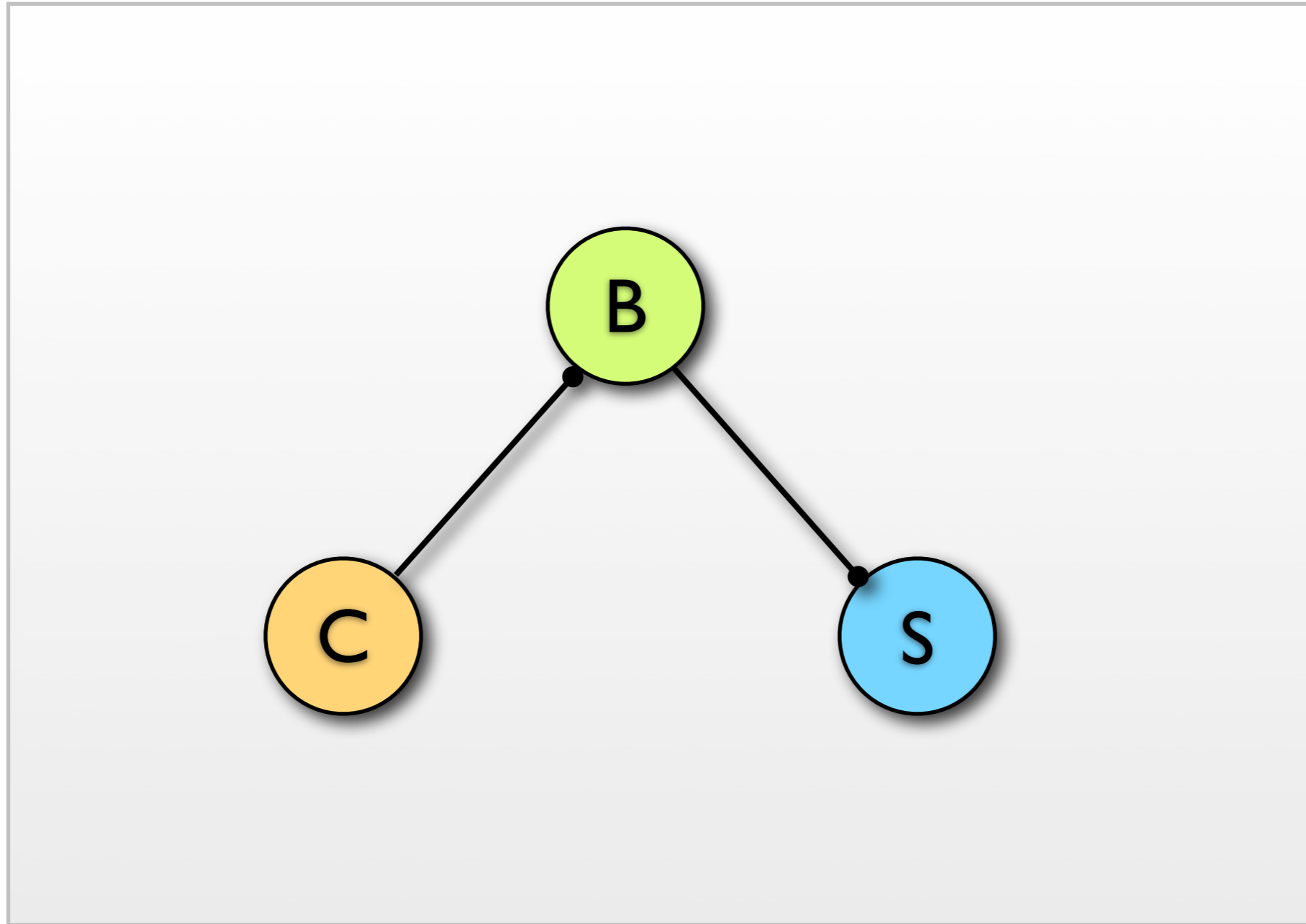
Thank you!

<http://pi-workflow.org>

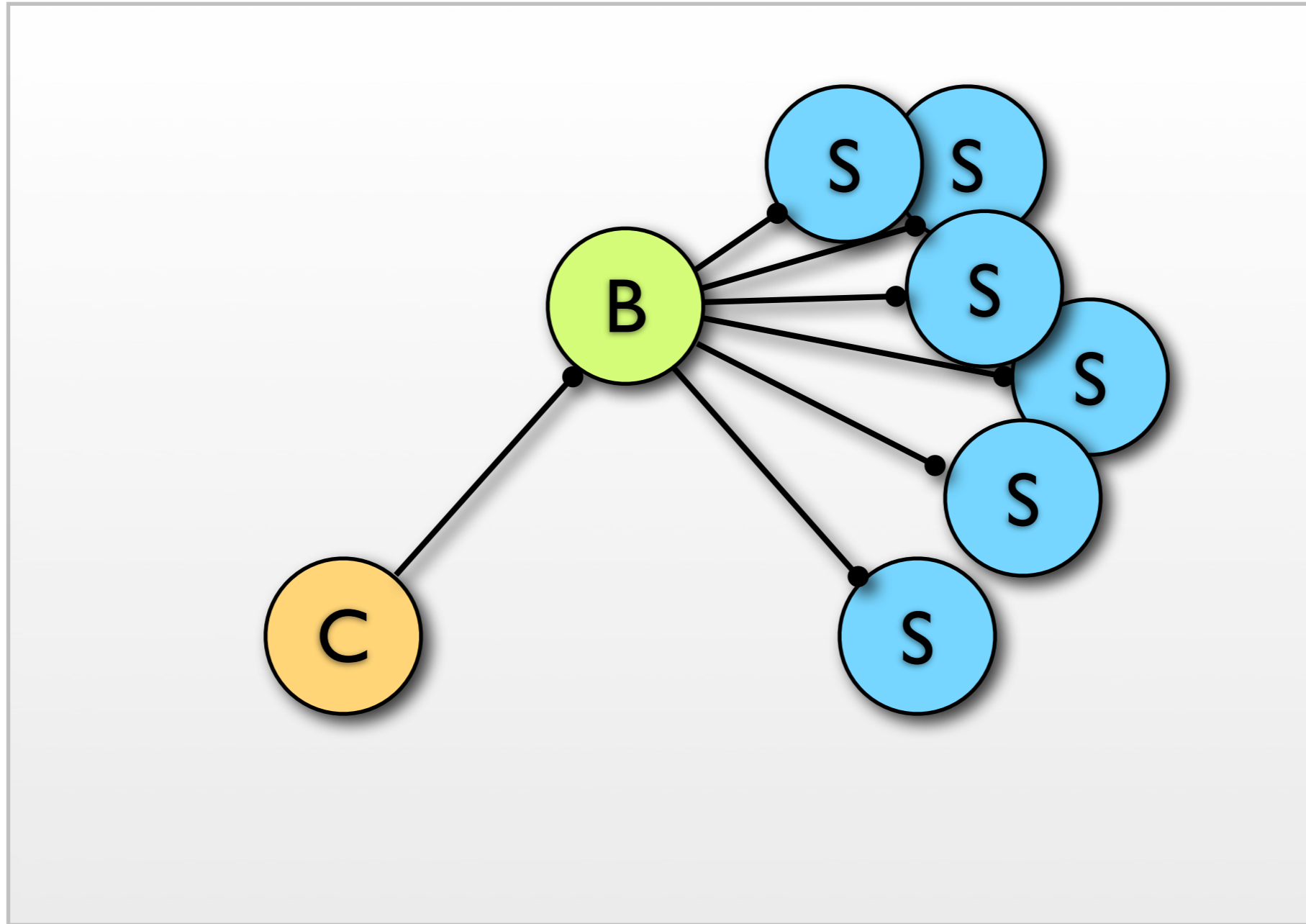
**Actually:
Pi-Calculus was SOA
long before SOA!**

(And SOA is the realization for BPM!)

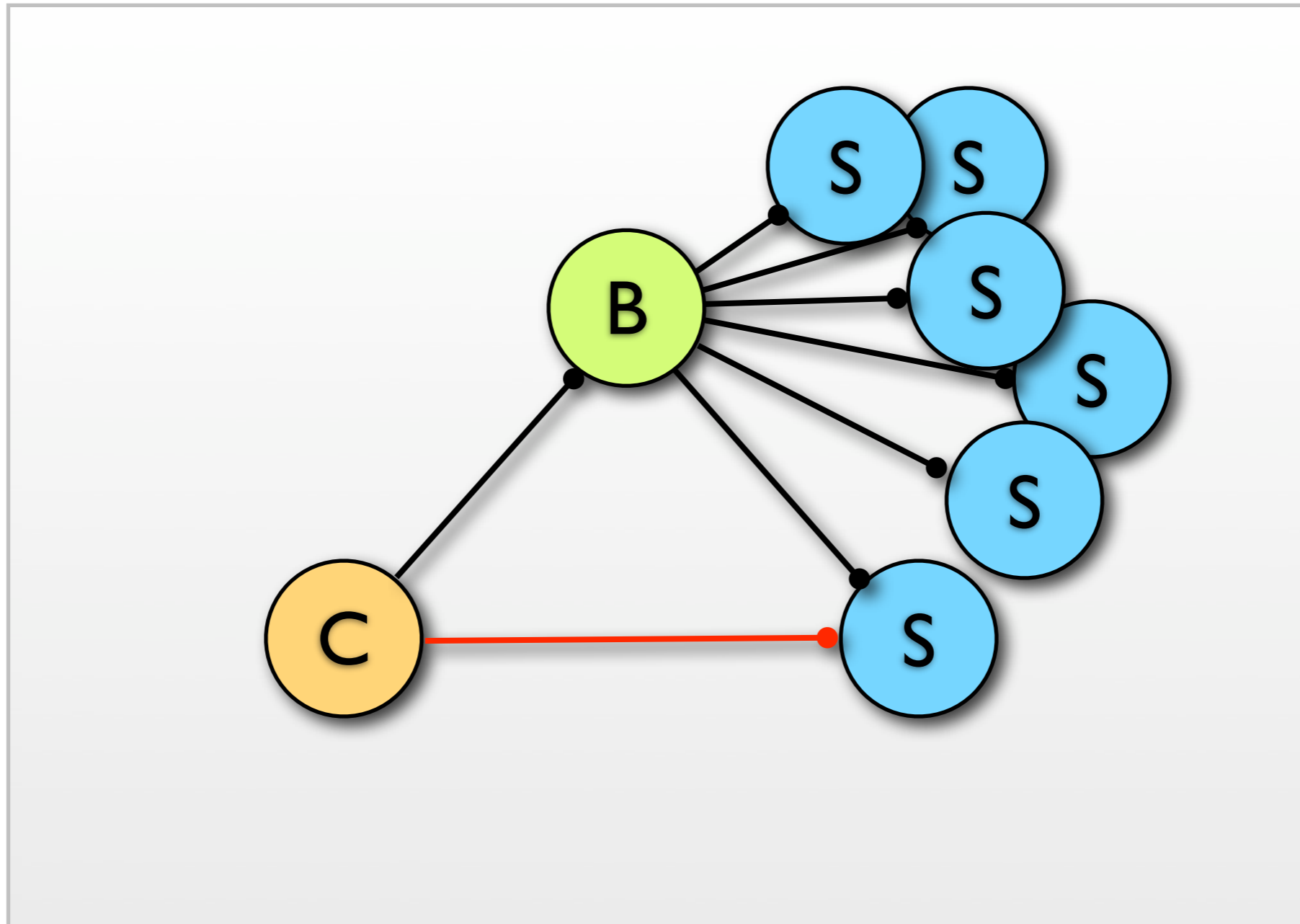
1989



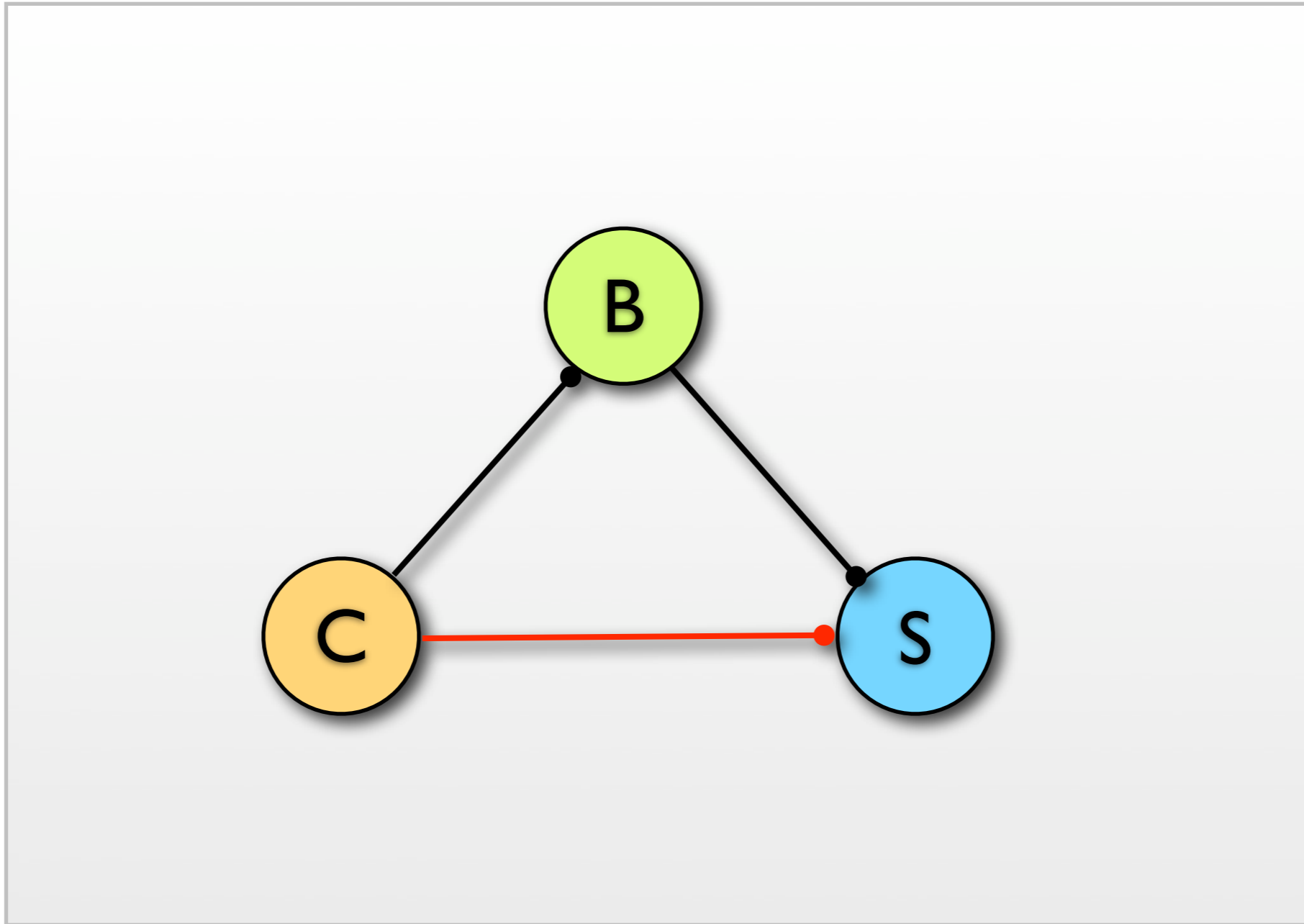
Link Passing Mobility



Link Passing Mobility

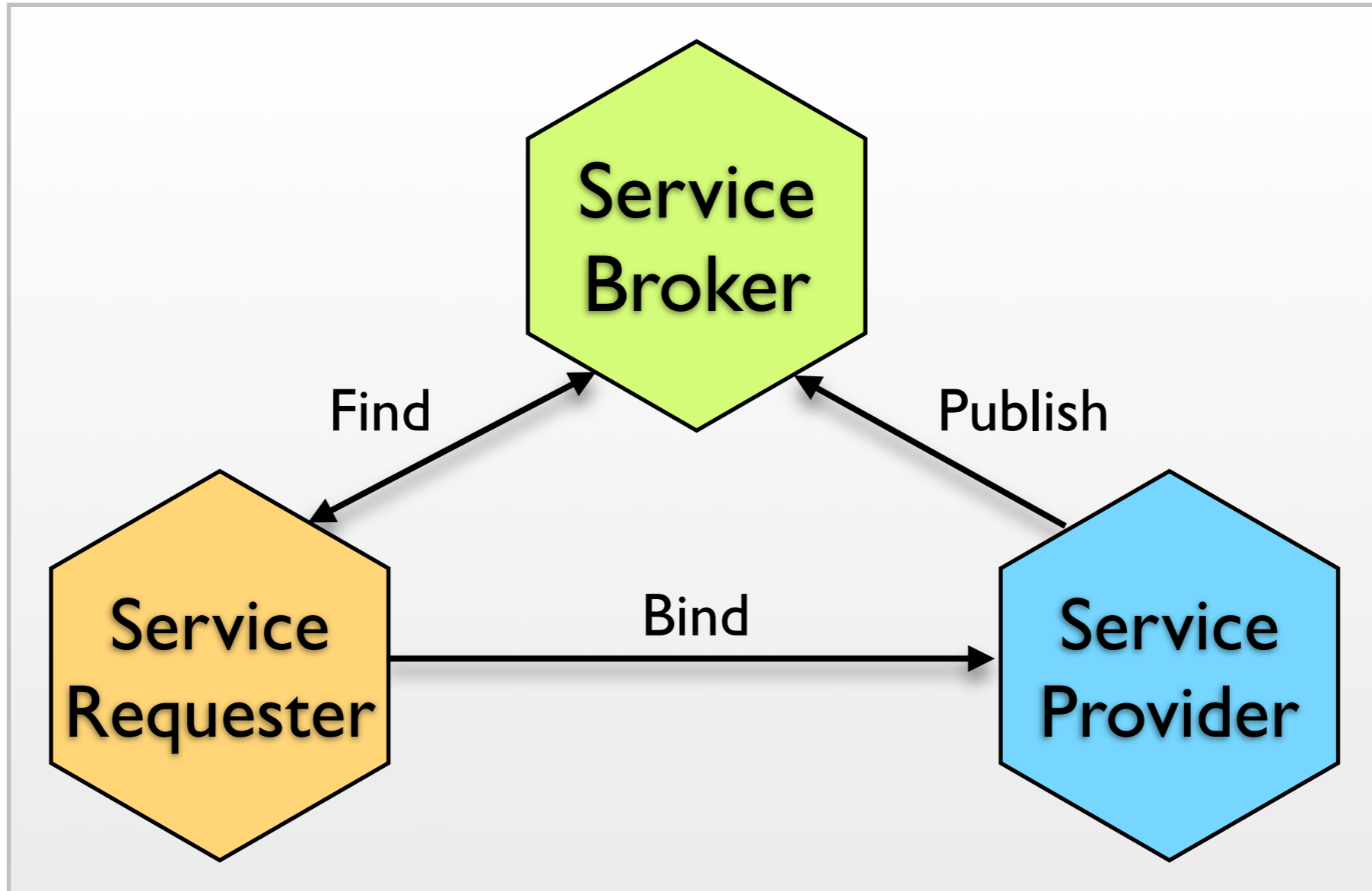


Link Passing Mobility



Link Passing Mobility

Since 2000



SOA