Manufacturing Engineering

Orchestration and Choreography for Business Change

Conrad Bock

NIST

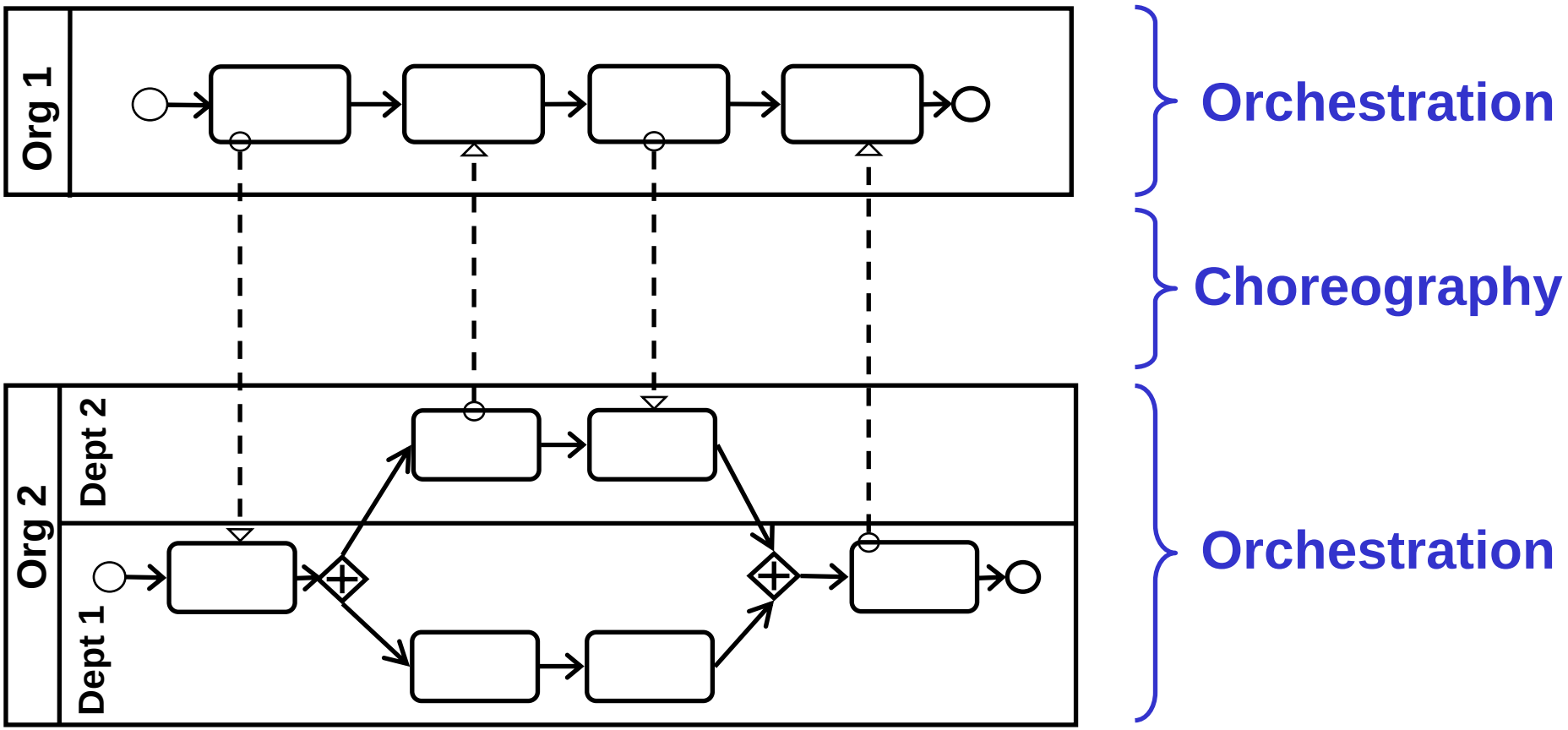
June 26, 2006

NIST

Overview

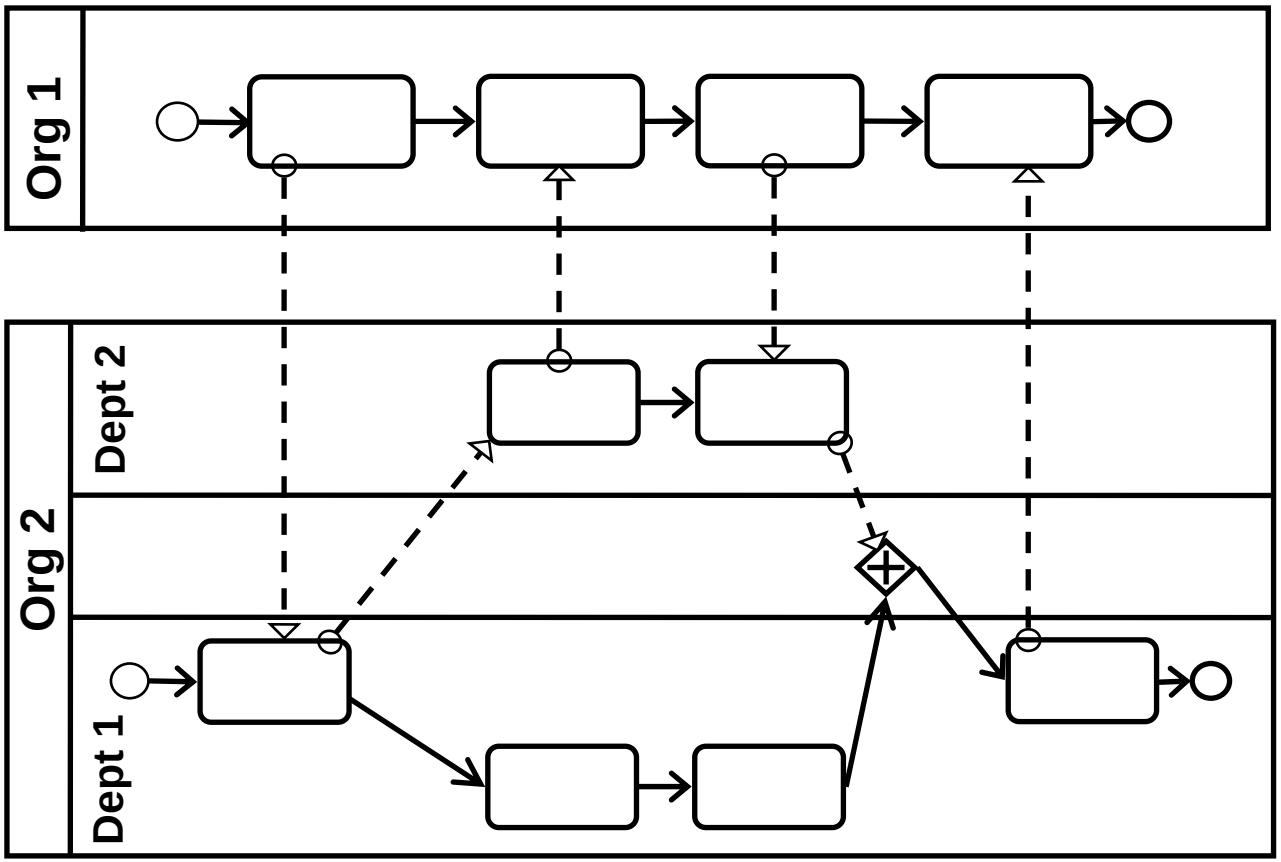
- **What are orchestration and choreography?**
- **Effect of business change on O/C.**
- **Business process models without O/C.**
- **Segmented process specification.**
- **Protocol processes.**

O/C = Organization Boundary ?



- **Orchestration inside organization boundary, choreography across it.**

O/C ≠ Organization Boundary



Orchestration

Choreography

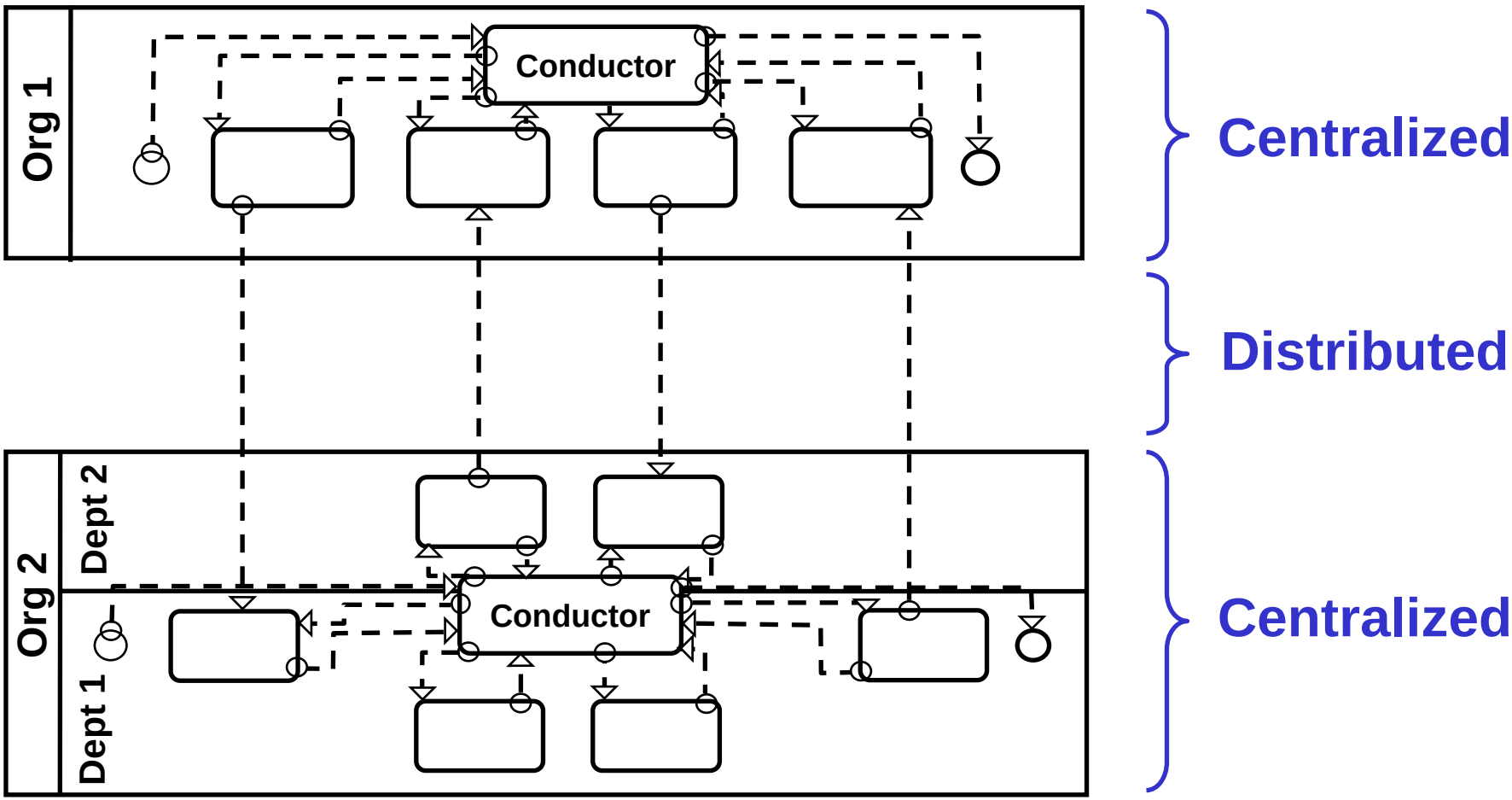
Orchestration

Choreography?

Orchestration

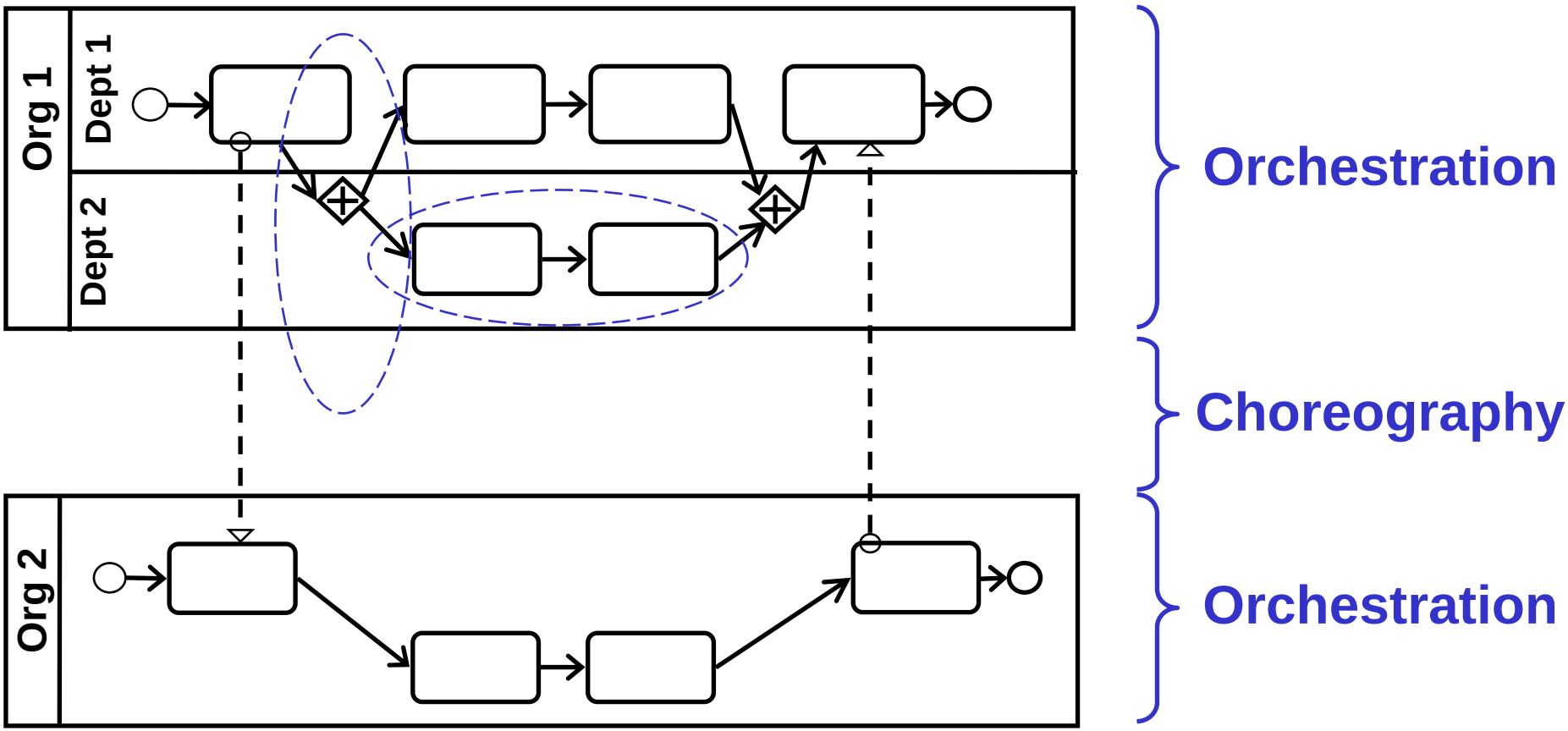
- **Distributed coordination inside organization.**
- **Is it orchestration or choreography?**

O/C = Coordination Pattern



- **Centralized coordination inside organization boundary, distributed coordination across it.**

Organization Change

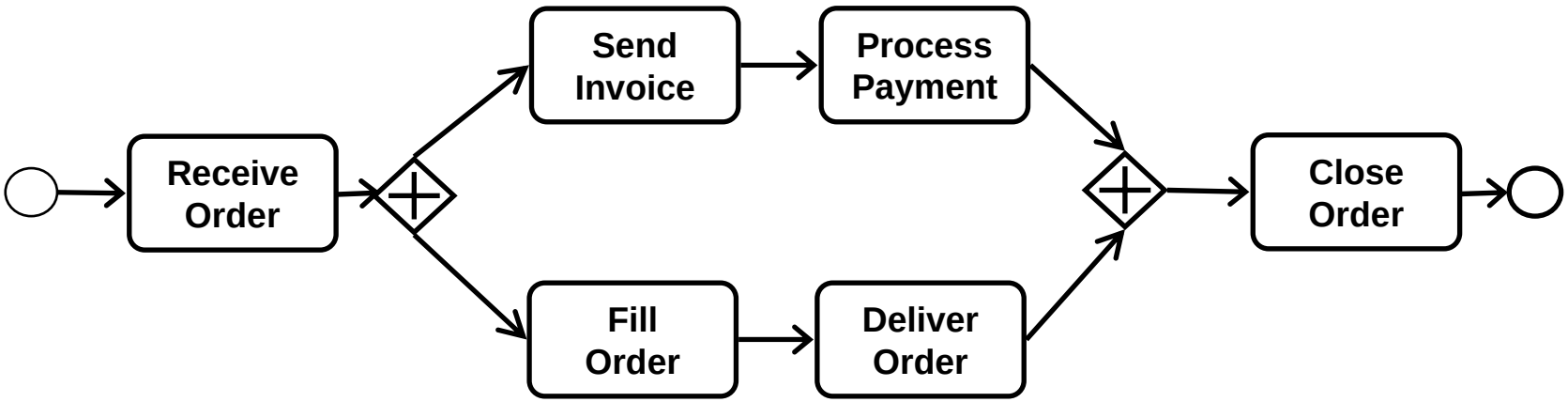


- **Change the boundary, change the model.**
- **Do the steps happen the same way as before?**

High-level Business Viewpoint

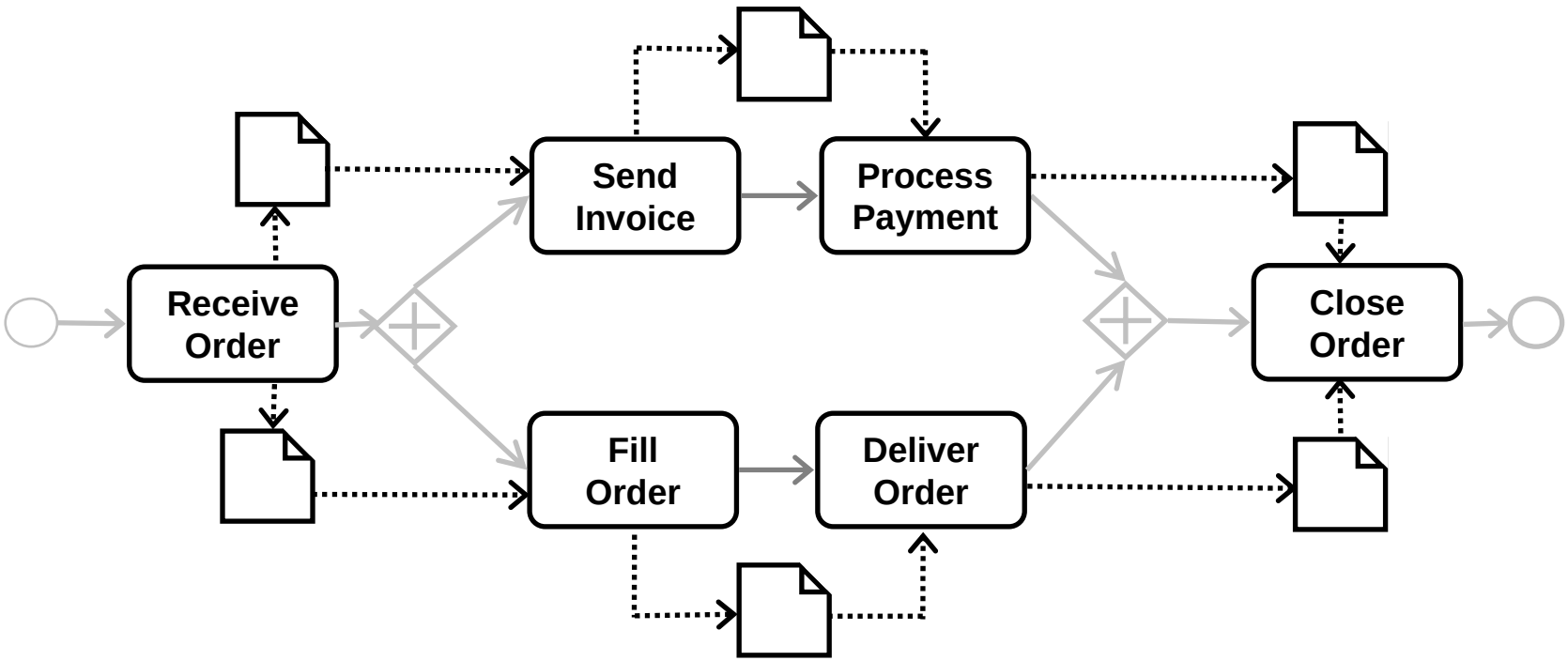
- **A high-level business view does not need to commit to orchestration or choreography.**
- **It might only specify timing of the steps, and what things / people are involved:**
  - **What steps happens after which others (sequence).**
  - **What things are used in common by which steps (output / input).**
  - **Who performs each step.**

Sequence



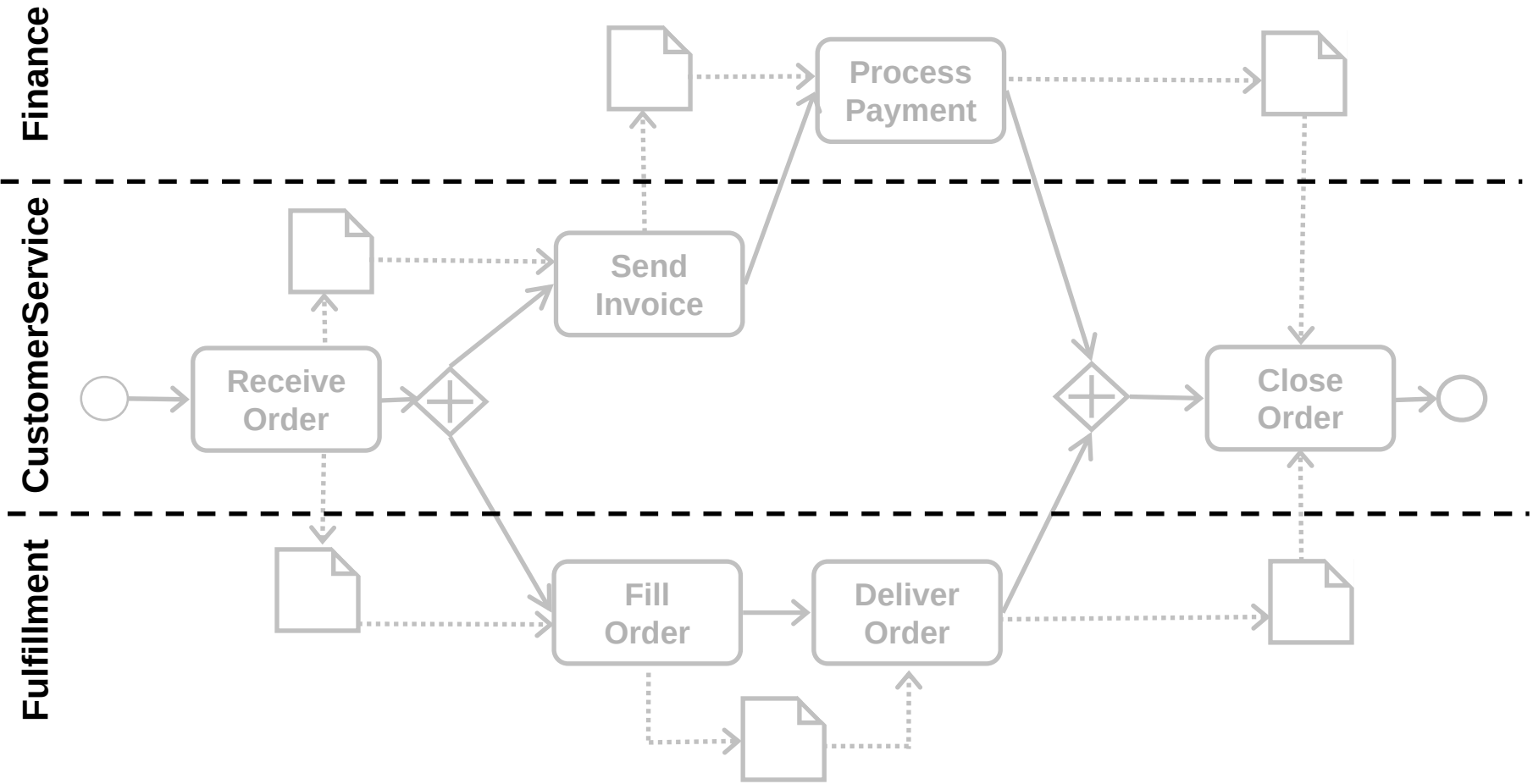
- **What happens after what:**
  - **The first step is ReceiveOrder.**
  - **SendInvoice starts after ReceiveOrder is finished.**
  - **Fill Order starts after ReceiveOrder is finished.**
  - **Process Payment starts after SendInvoice is finished …**
- **No commitment to how this is ensured (coordination).**

Objects in Common



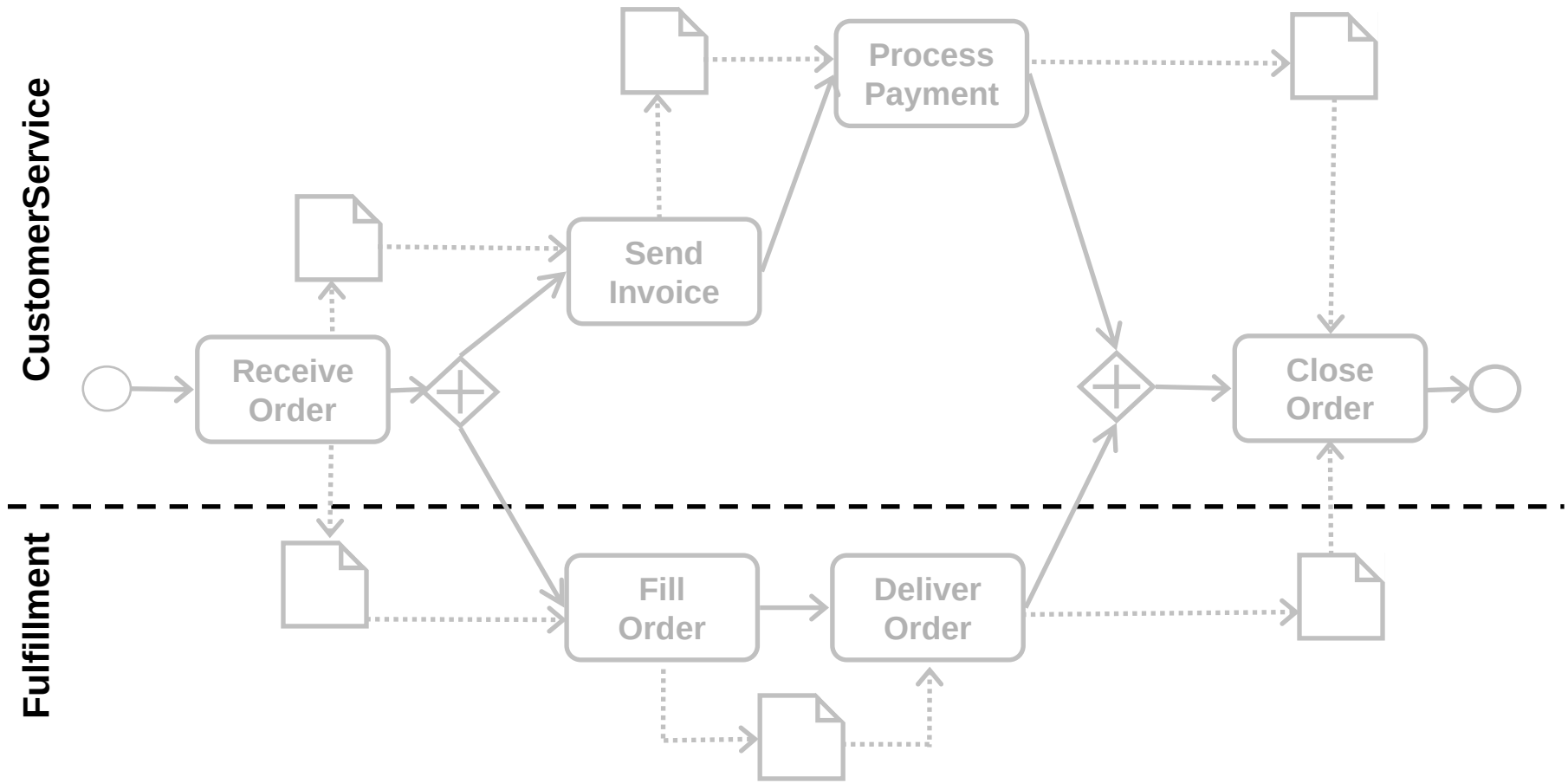
- **What objects are used in common:**
  - **The input document to SendInvoice is the same as the output document of ReceiveOrder.**
  - **The input document to FillOrder is the same as the output document of ReceiveOrder …**
- **No commitment to how this is ensured (coordinatio**

Performers



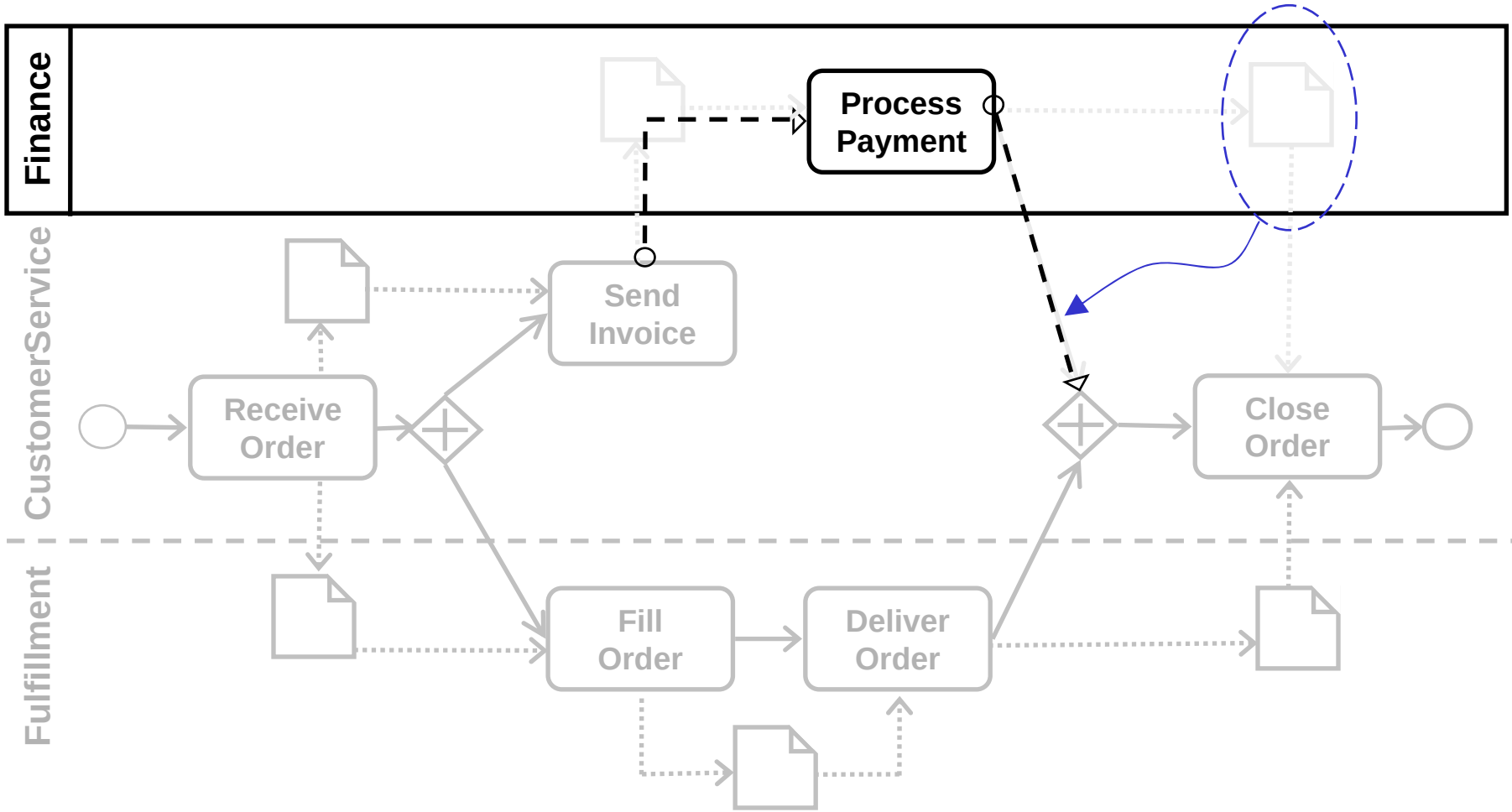
- CustomerService performs ReceiveOrder.
- Fulfillment performs FillOrder …
- Still no coordination pattern committed.

Organization change



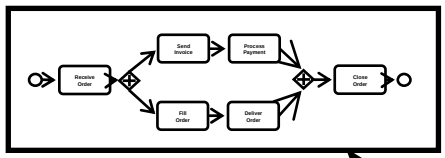
- **CustomerService performs ReceiveOrder, Process Payment, …**
- **Fulfillment performs FillOrder, DeliverOrder.**

Coordination



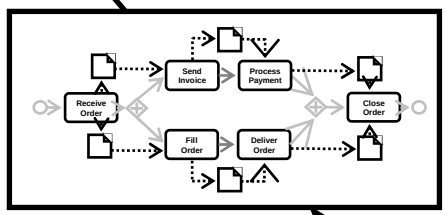
- **Coordination pattern must conform to constraints of the previous slides.**

Process Refinement

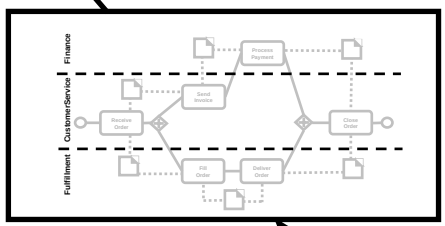


Sequence

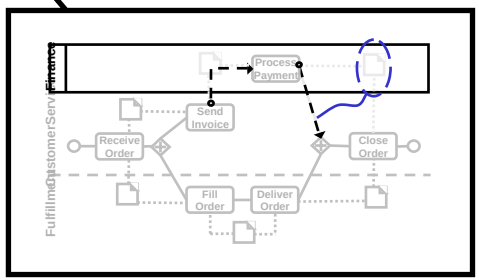
More commitment (not waterfall)



Object I/O



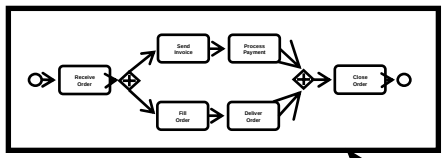
Performers



Coordination

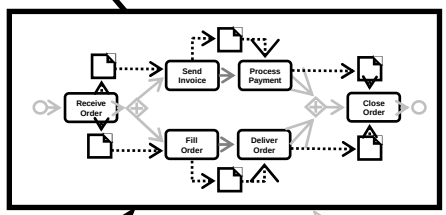
- **Refinement = more commitments.**

Refinement Rollback

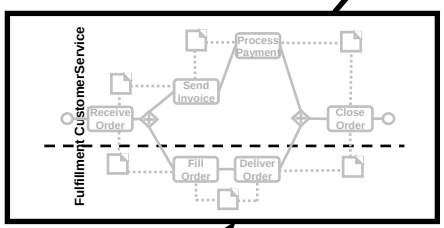


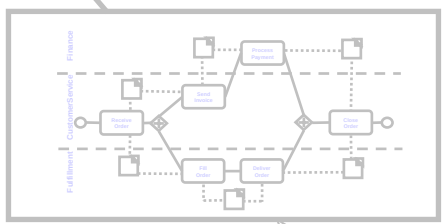
Sequence

Less commitment

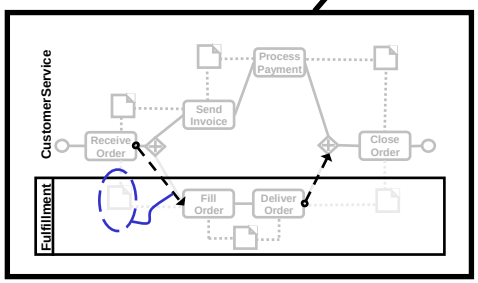


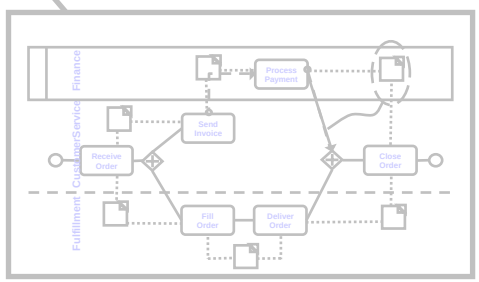
Object I/O





Performers

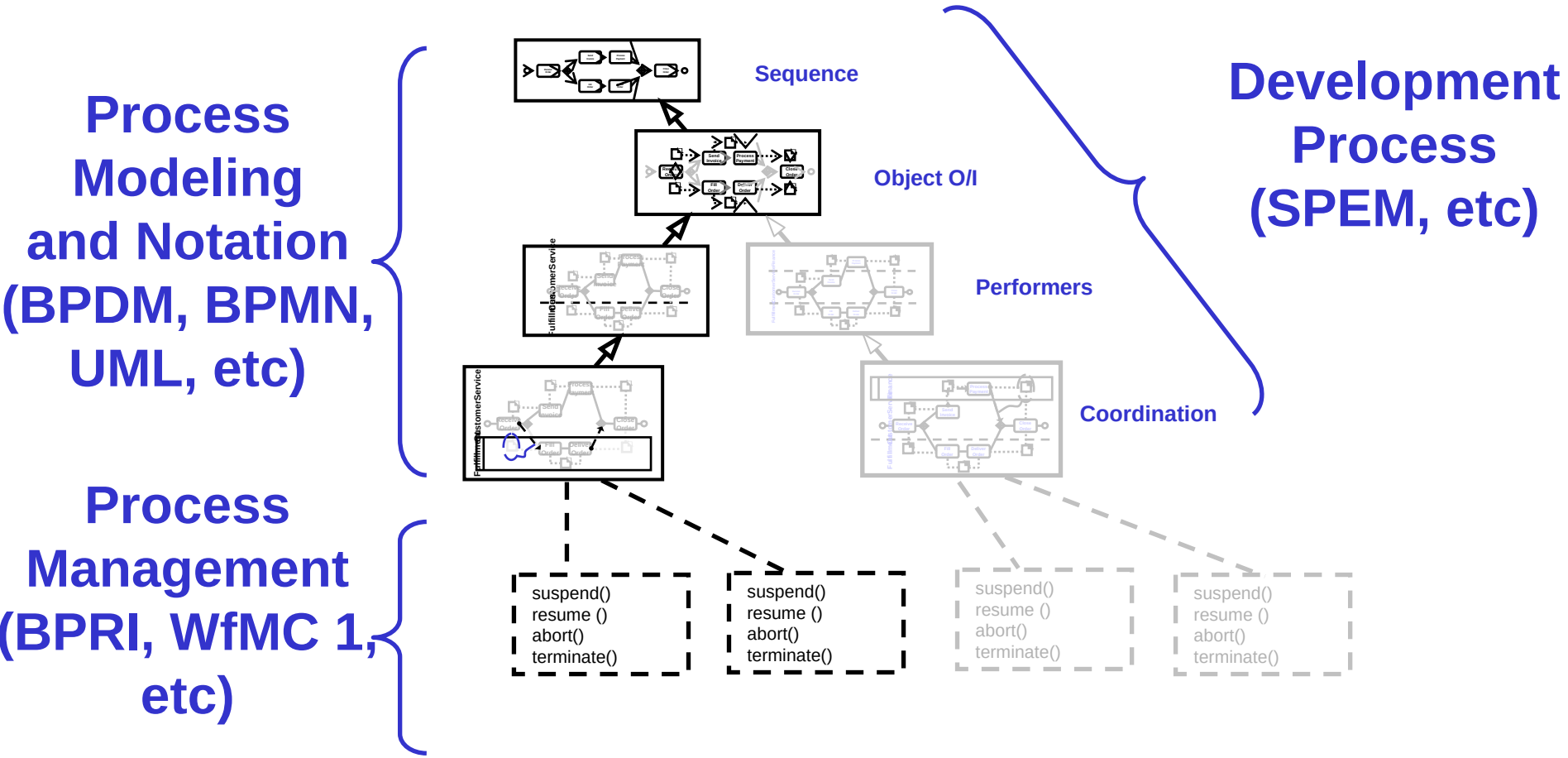




Coordination

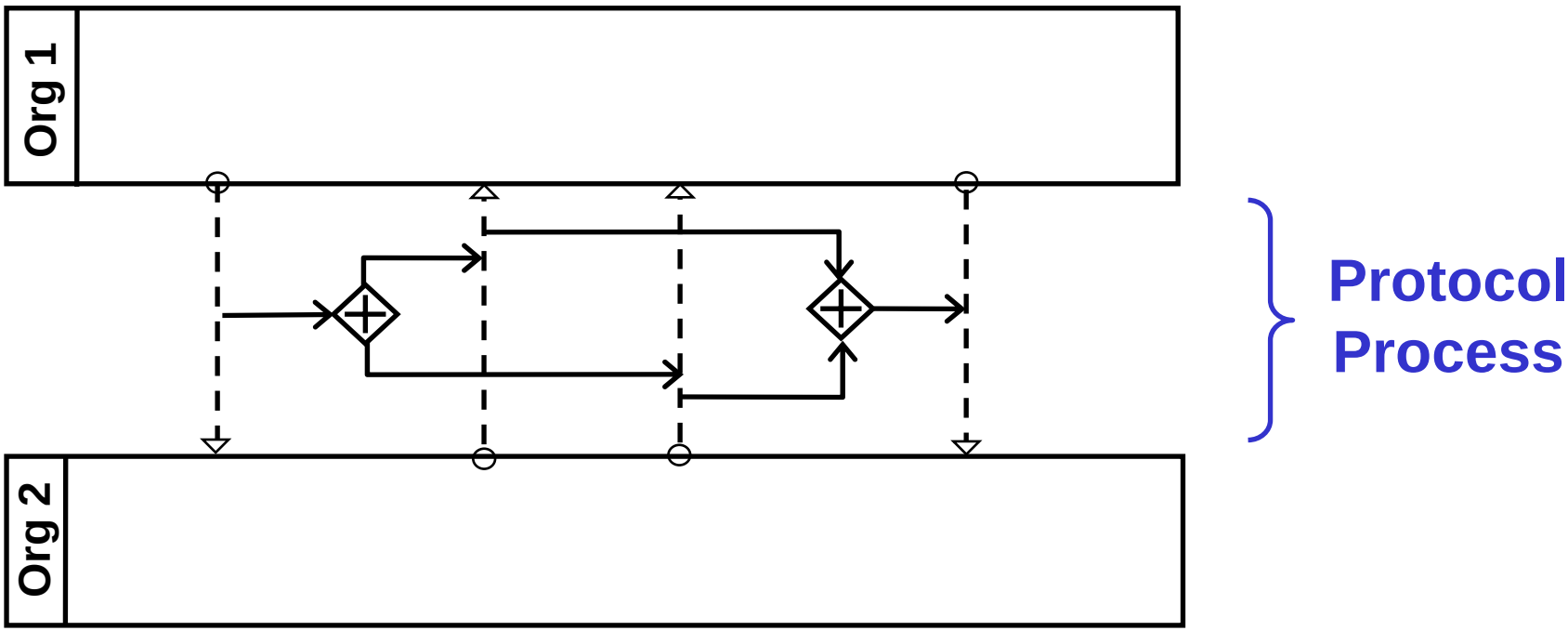
- **Alternative commitments (keep for what-if).**

Relation Between Standards



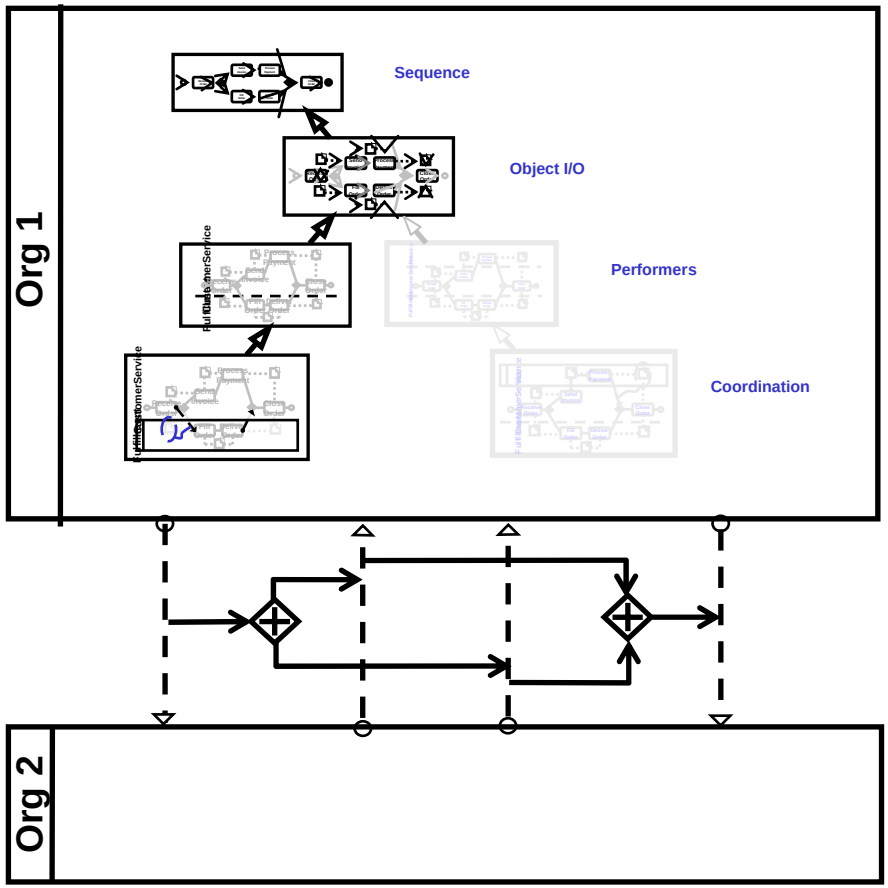
- **Process instances conform to constraints "inherited" from specifications.**

Protocol Processes



- **Use sequencing and object I/O constraints on messages.**
  - **After first message arrives at Org 2, second two are sent in parallel to Org 1.**
  - **After those both arrive at Org 1, last message is sent to Org2.**
- **For defining standard or contractual interactions (eg, RosettaNet).**

Adapting to Protocols



Rolling back coordination commitments

New or changed protocol

- **Specialization rollback for adapting to new and changing protocols.**

Summary

- **Orchestration and choreography are coordination / messaging patterns.**
- **Committing to them makes process specifications brittle under org change.**
- **Segmented process specialization gives access to other commitments when organizations change.**
- **Executing processes should conform to inherited constraints.**
- **Protocol processes can specify standard industry interactions.**