Naiad: A Timely Dataflow System

Derek G. Murray, Frank McSherry, Rebecca Isaacs
Michael Isard, Paul Barham, Martin Abadi

“Historic” Context

- The world in 2013
  - Hadoop, Spark, etc.: General purpose batch data processing (high latency)
  - Pregel, GraphLab, etc.: Graph processing (iterative)
  - Impala, F1, etc.: Fastish SQL queries
  - MillWheel, Storm, etc.: Stream processing systems (low latency)

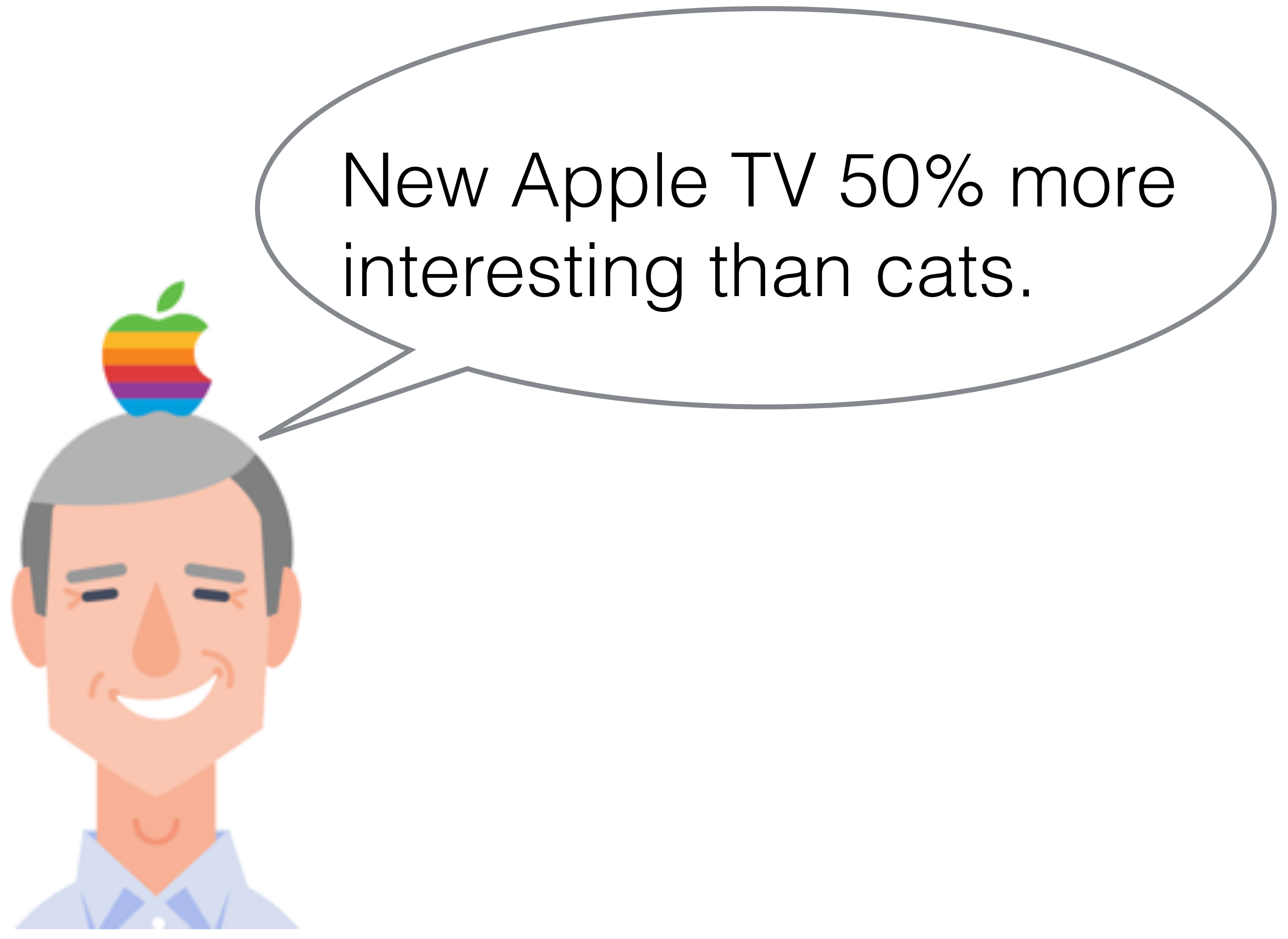
“Historic” Context

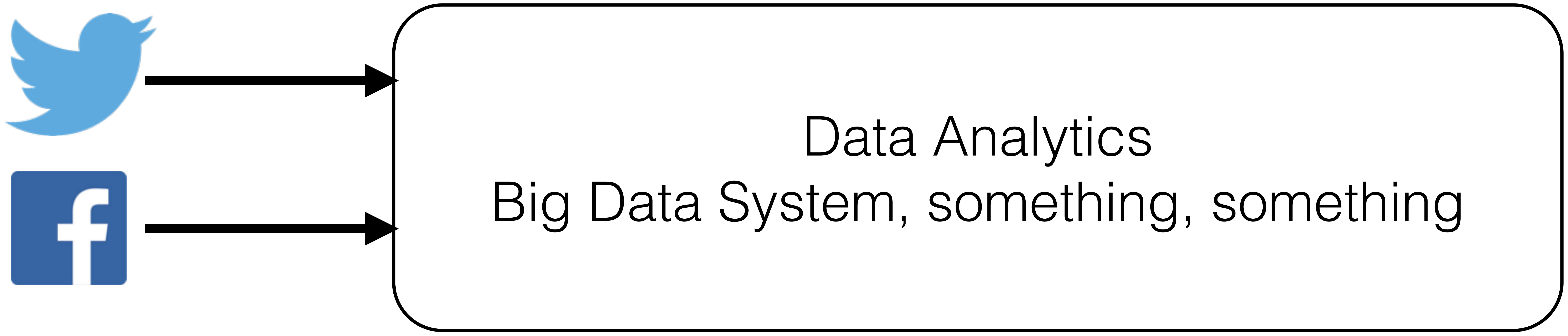
- The world in 2013
    - Hadoop, Spark, etc.: General purpose batch data processing (high latency)
    - Pregel, GraphLab, etc.: Graph processing (iterative)
    - Impala, F1, etc.: Fastish SQL queries
    - MillWheel, Storm, etc.: Stream processing systems (low latency)
- Lots of special purpose frameworks.

“Historic” Context

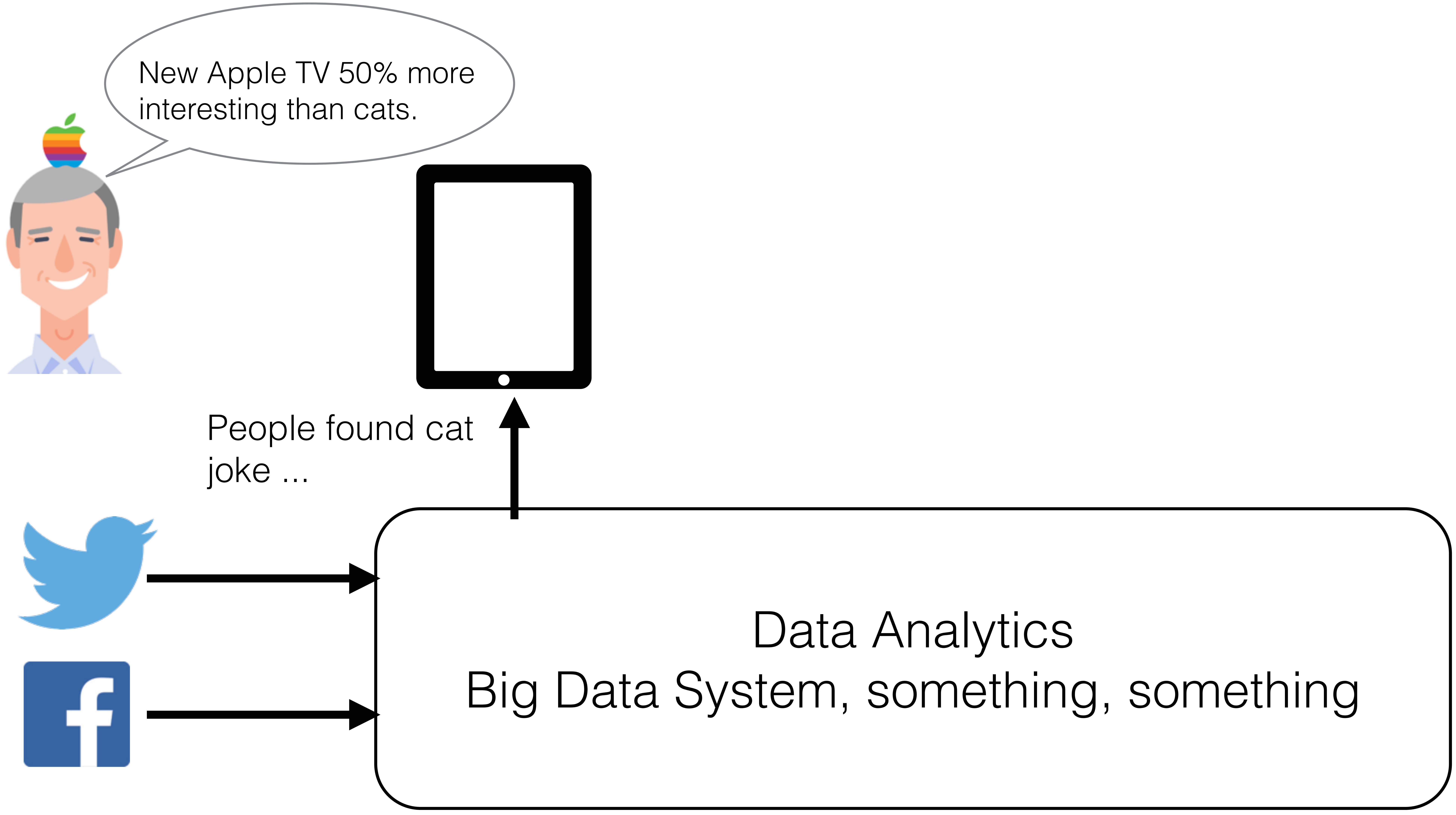
- The world in 2013
  - Hadoop, Spark, etc.: General purpose batch data processing (high latency)
  - Pregel, GraphLab, etc.: Graph processing (iterative)
  - Impala, F1, etc.: Fastish SQL queries
  - MillWheel, Storm, etc.: Stream processing systems (low latency)
- Lots of special purpose frameworks.
- A desire to allow these many paradigms to coexist in one system.
  - See also: Spark Streaming (next week), GraphX (later in the semester).

Why Multi-Paradigm?

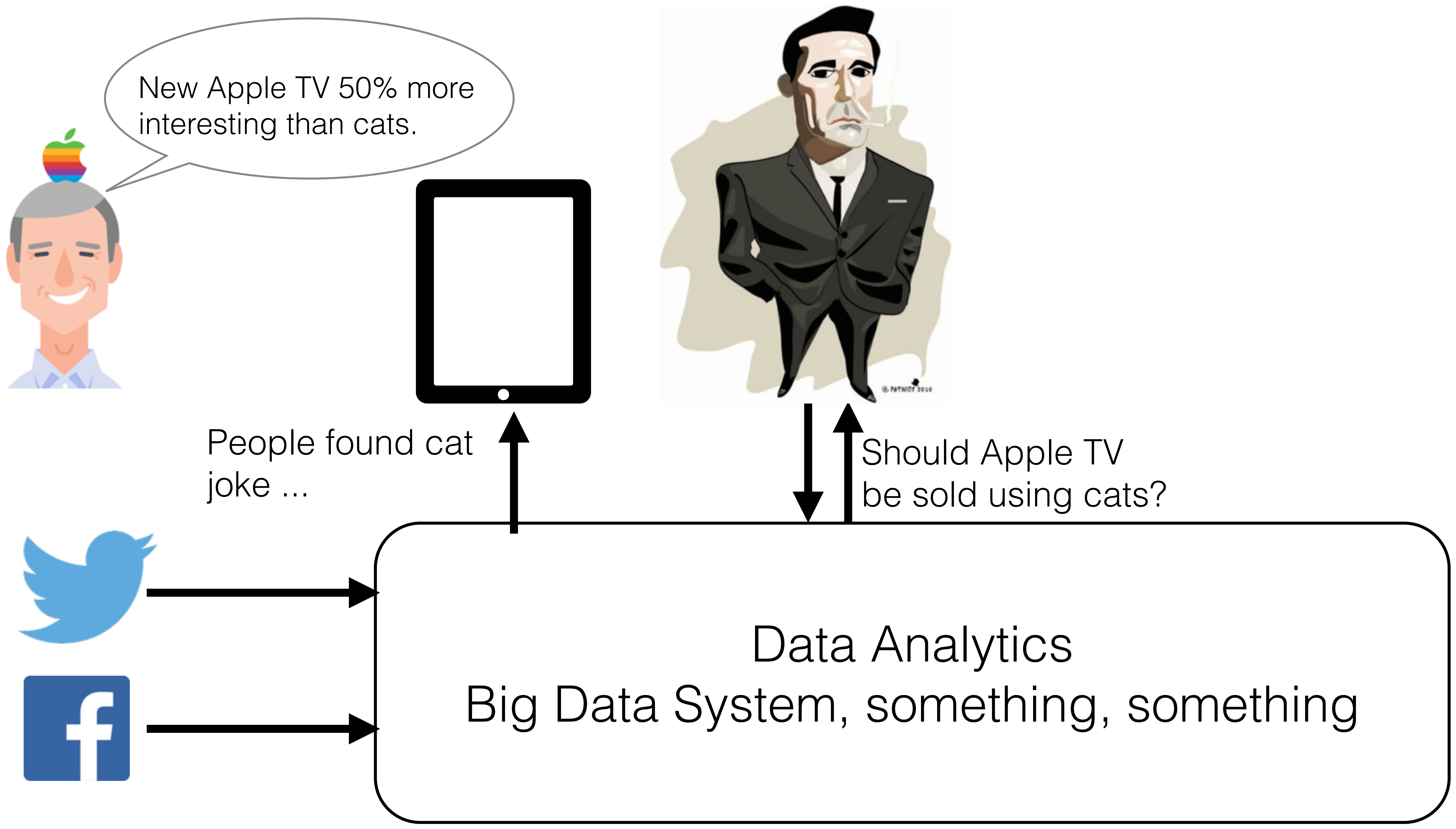




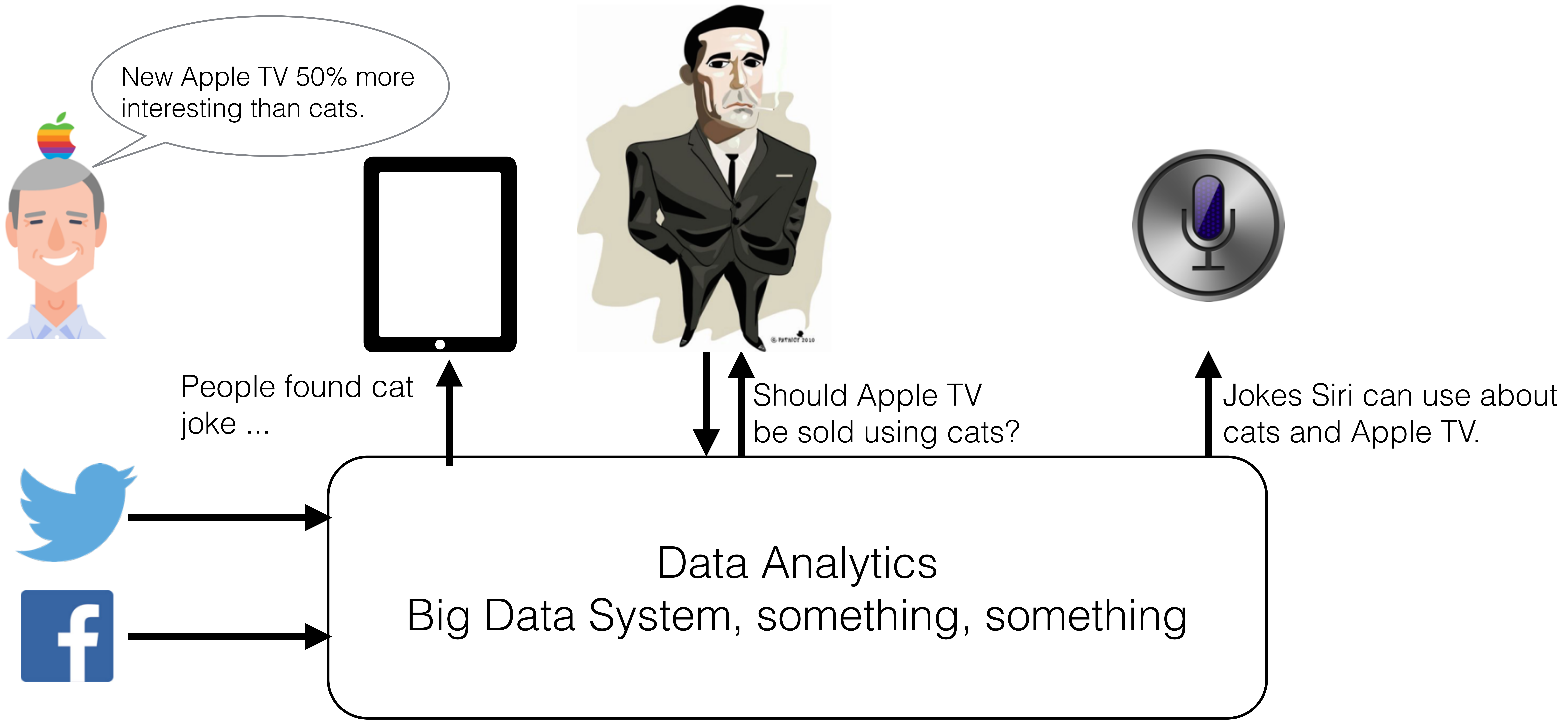
Why Multi-Paradigm?


New Apple TV 50% more interesting than cats.
People found cat joke ...
Data Analytics
Big Data System, something, something

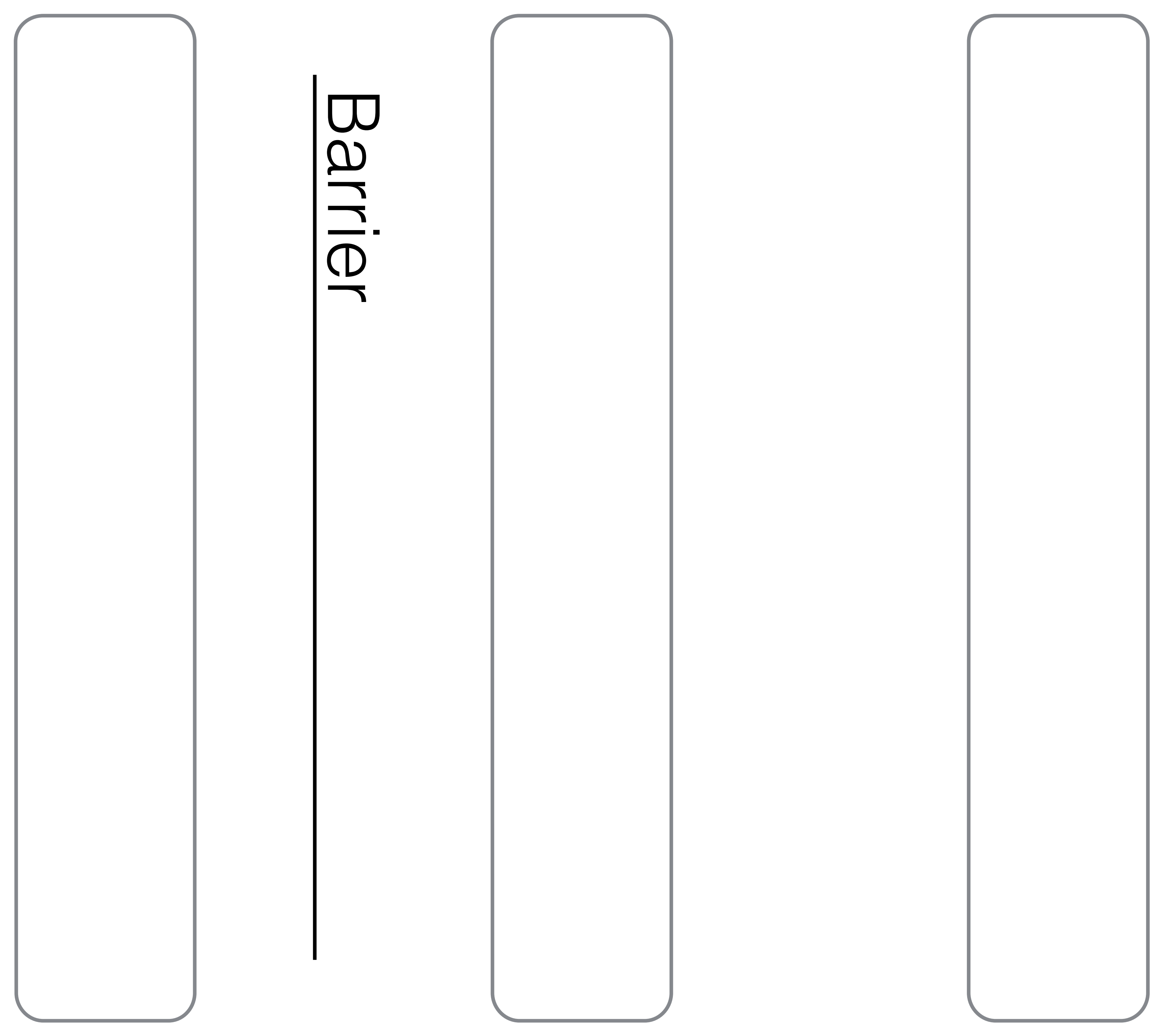
Why Multi-Paradigm?


New Apple TV 50% more interesting than cats.
People found cat joke ...
Should Apple TV be sold using cats?
Data Analytics
Big Data System, something, something

Why Multi-Paradigm?

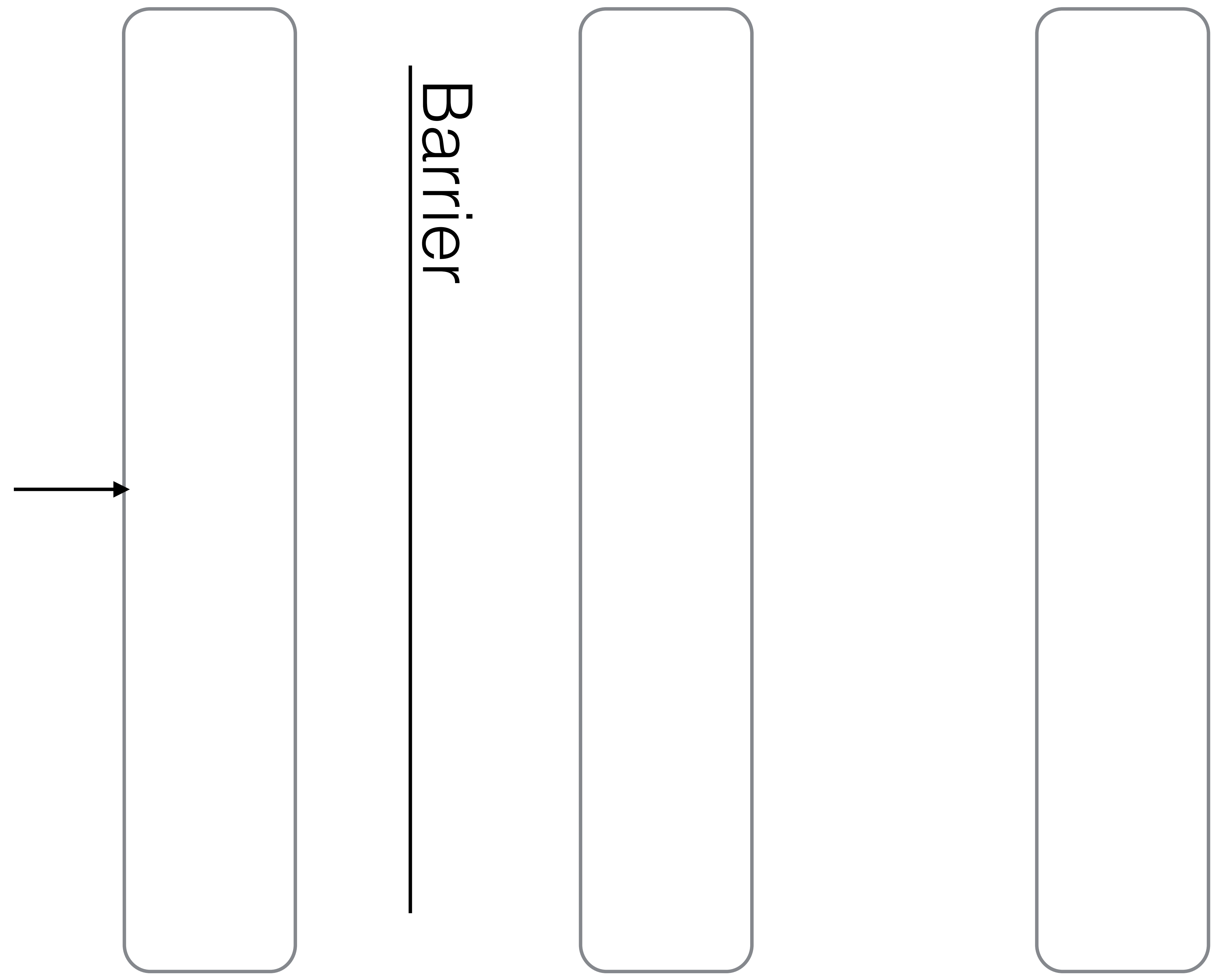

New Apple TV 50% more interesting than cats.
People found cat joke ...
Should Apple TV be sold using cats?
Jokes Siri can use about cats and Apple TV.
Data Analytics
Big Data System, something, something

Question: What Execution Model to Use?



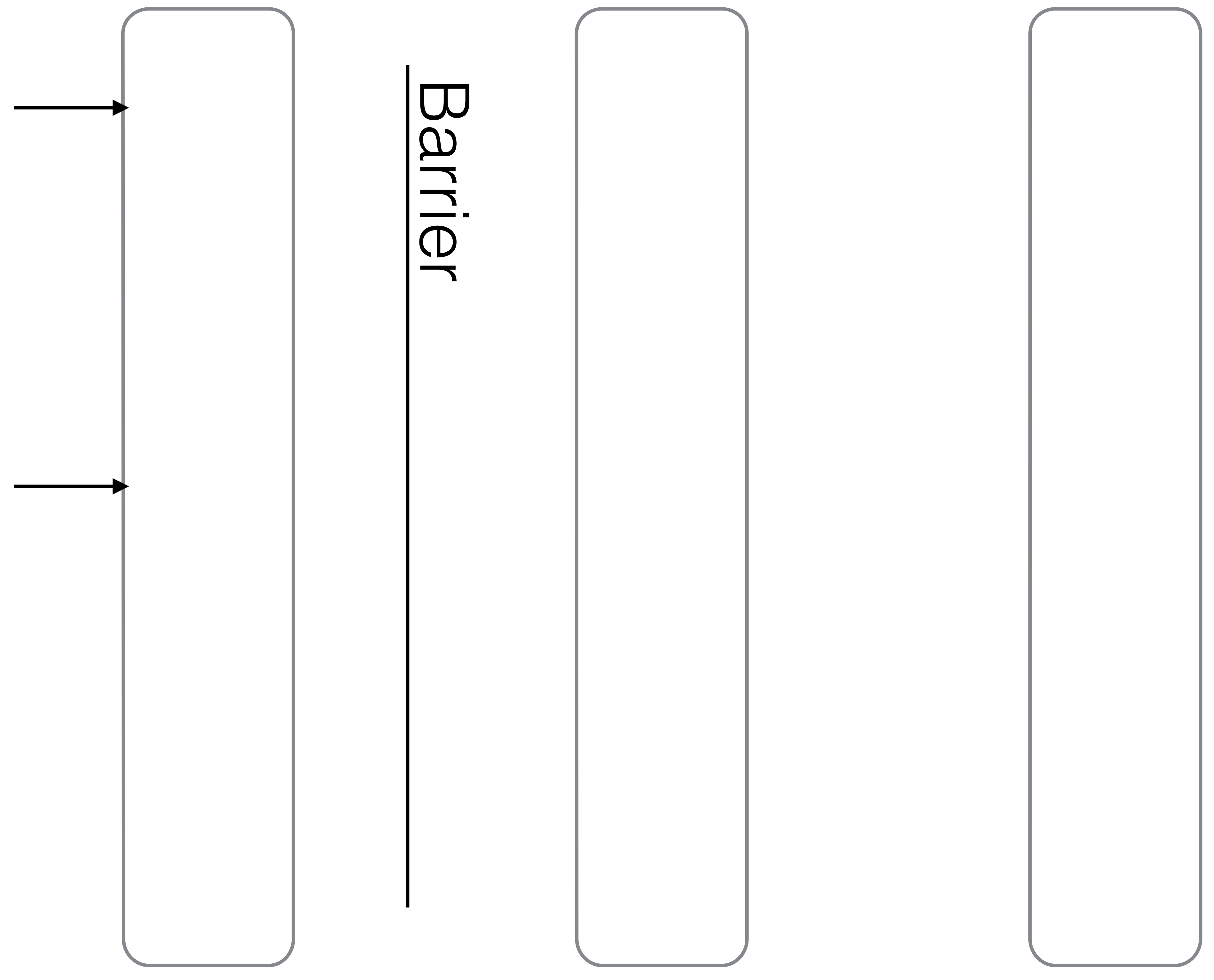
BSP

Question: What Execution Model to Use?



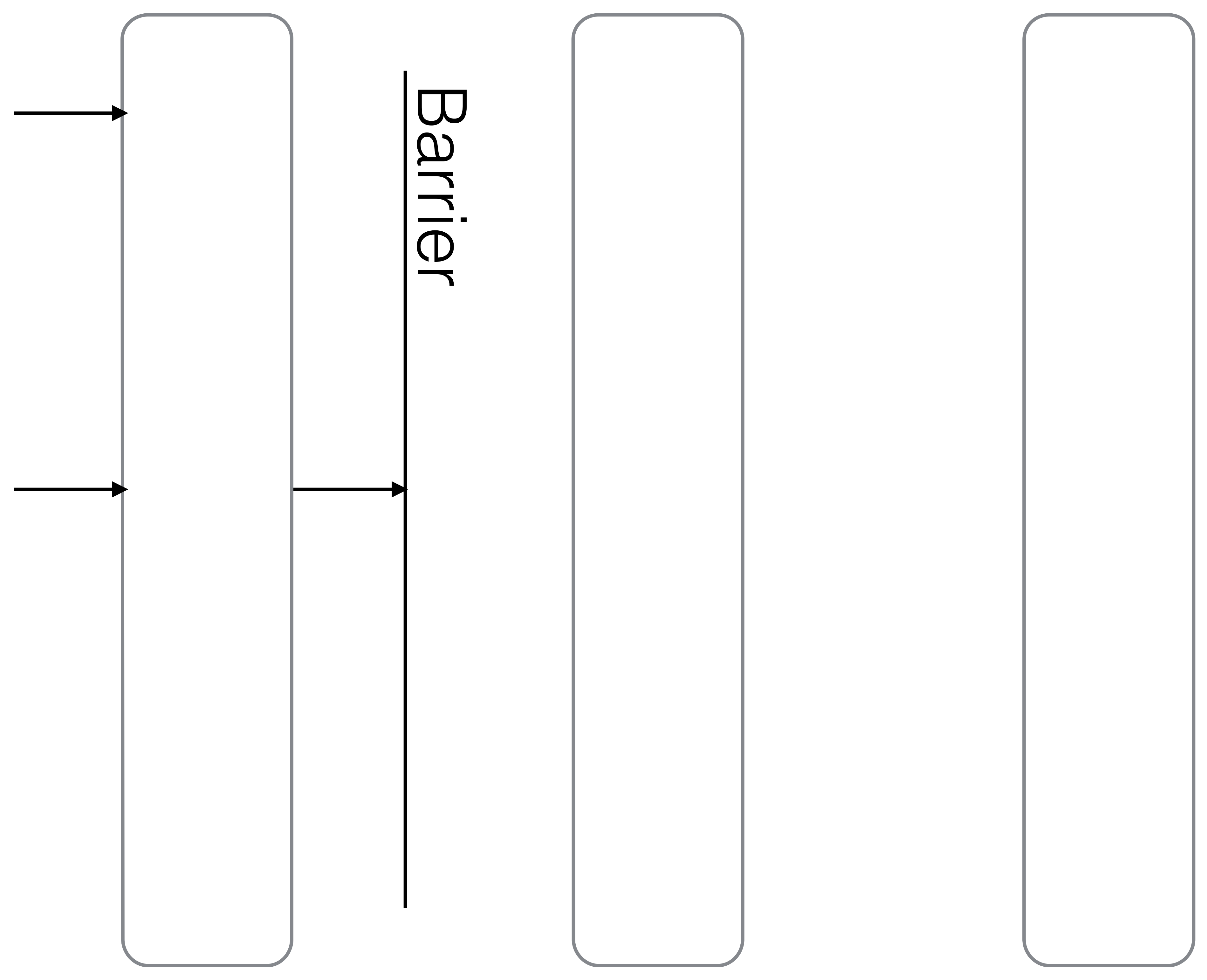
BSP

Question: What Execution Model to Use?



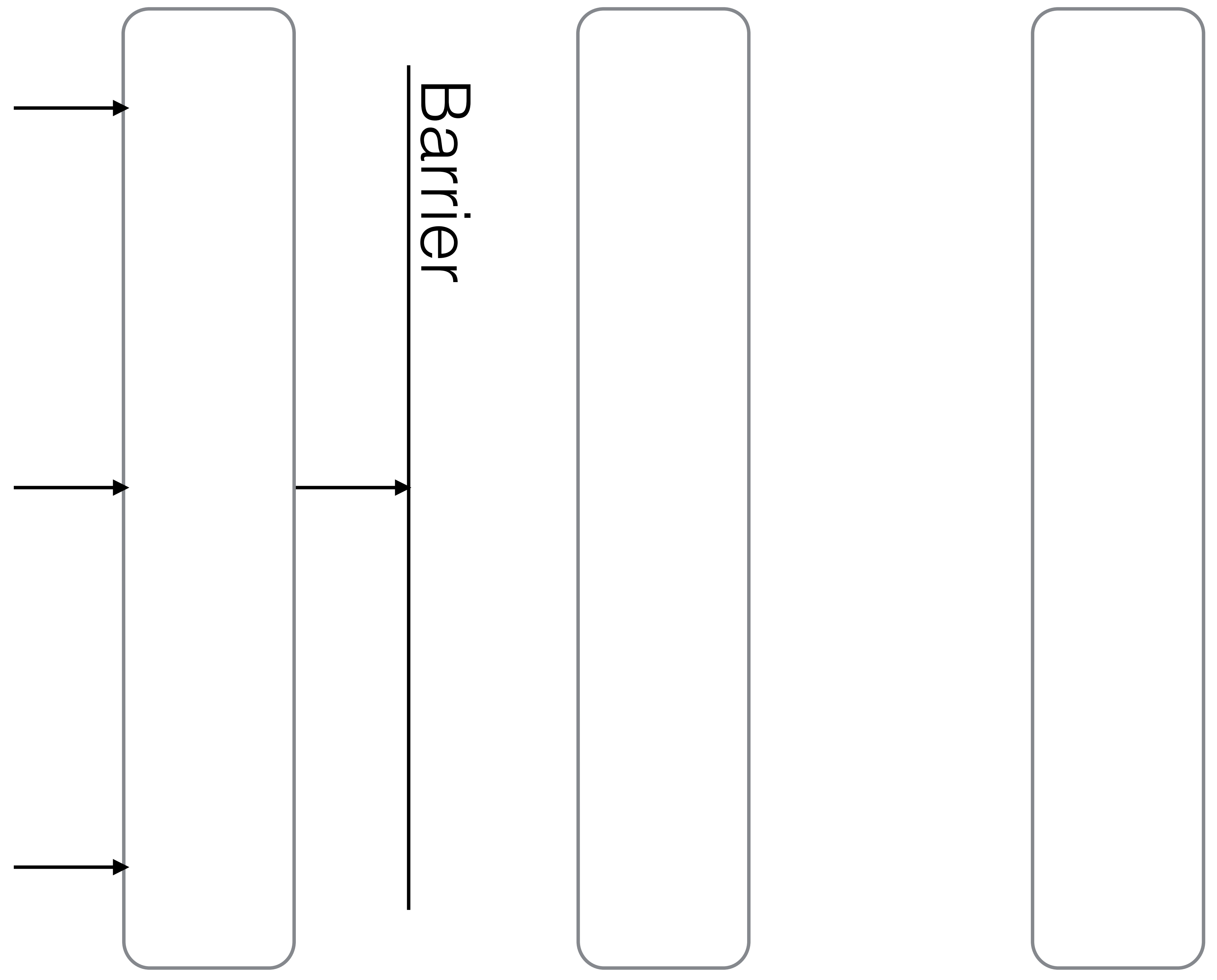
BSP

Question: What Execution Model to Use?



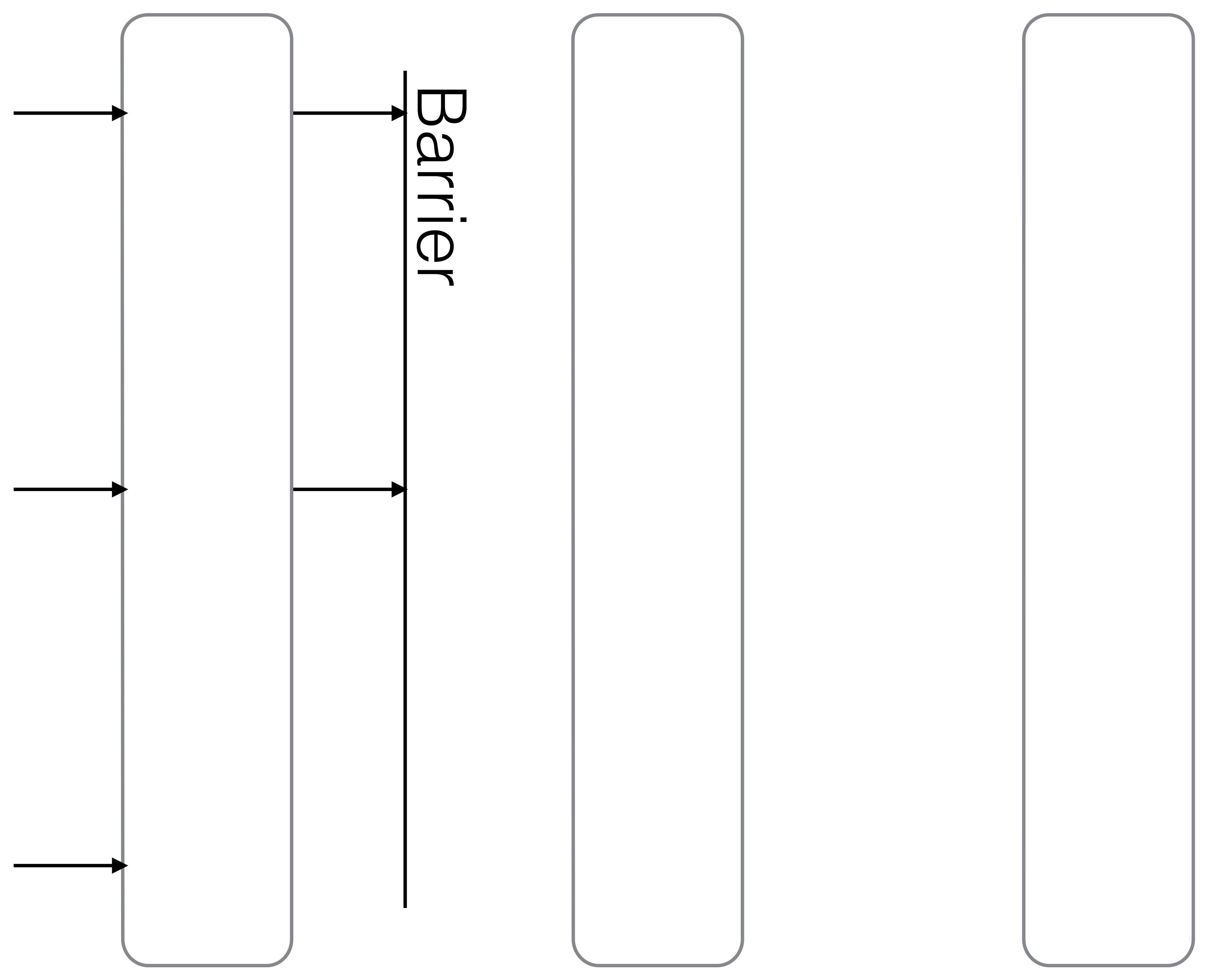
BSP

Question: What Execution Model to Use?



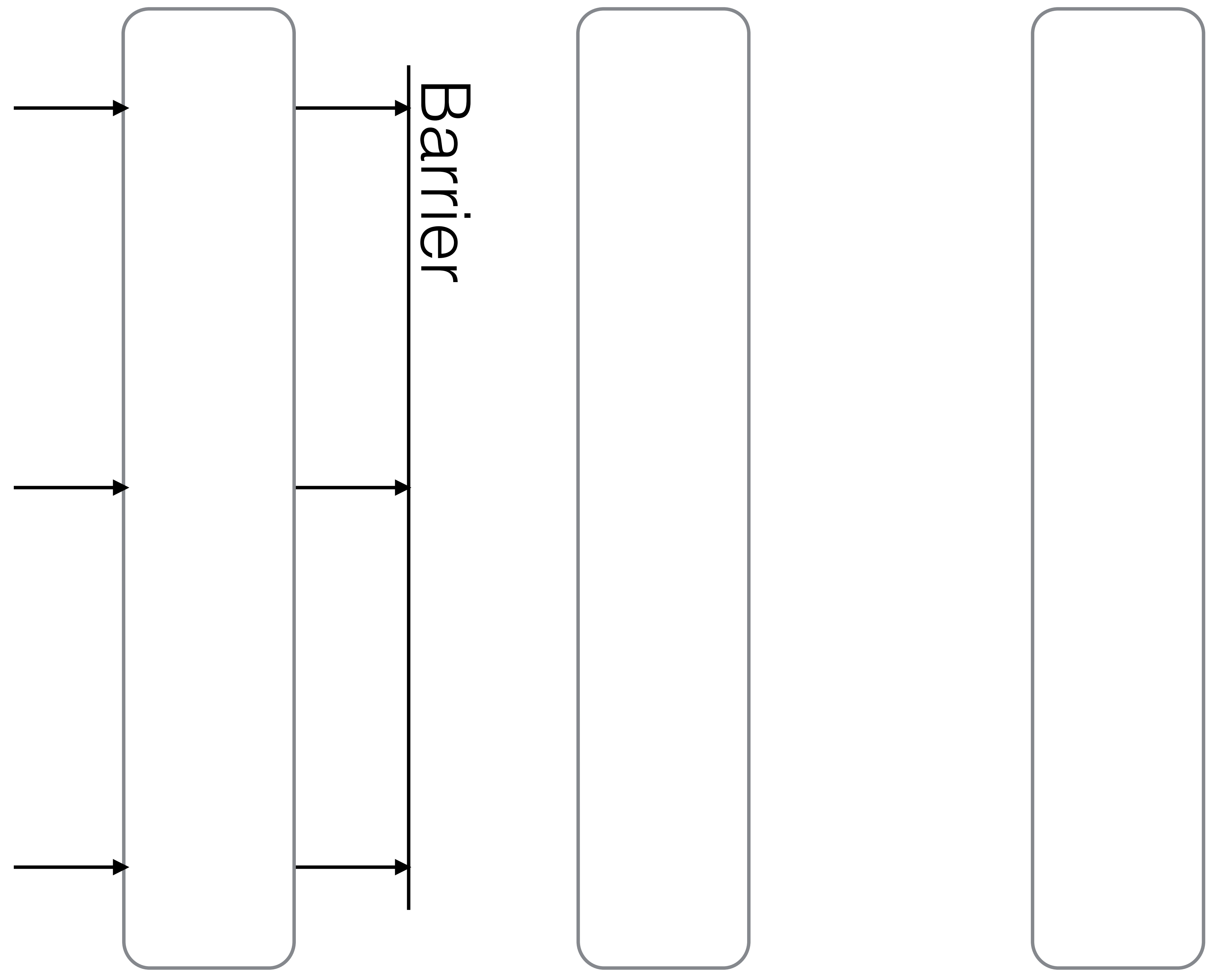
BSP

Question: What Execution Model to Use?



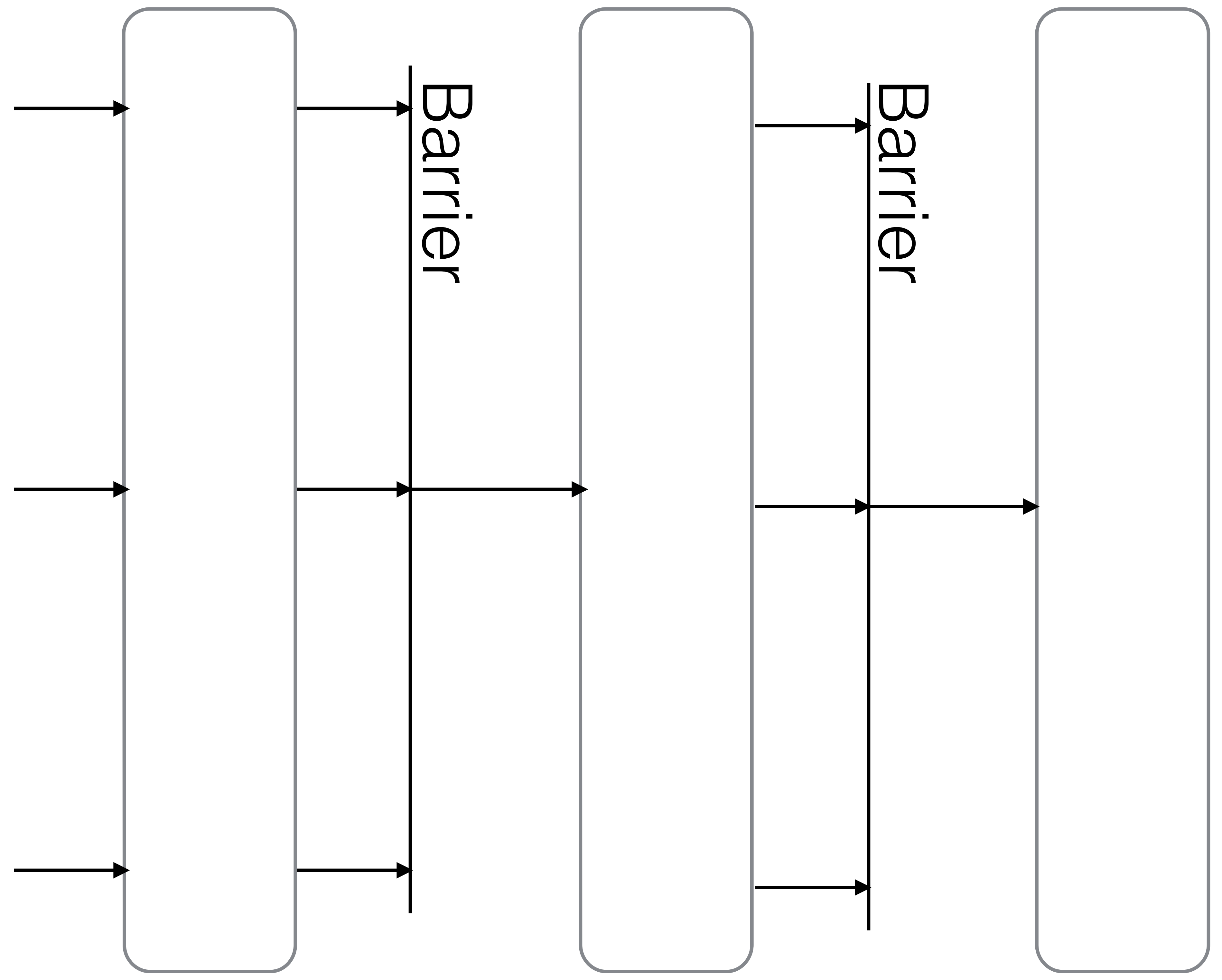
BSP

Question: What Execution Model to Use?



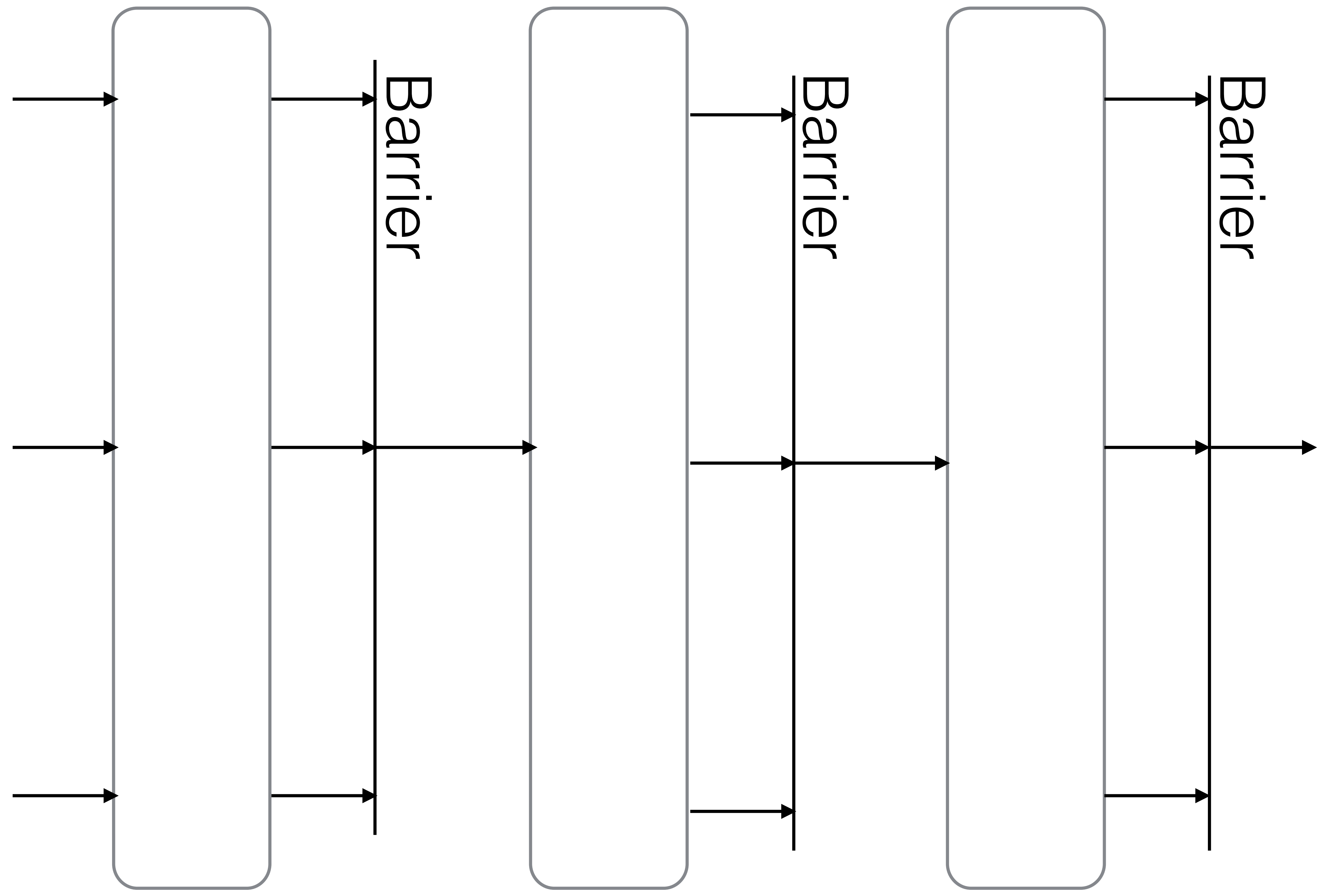
BSP

Question: What Execution Model to Use?



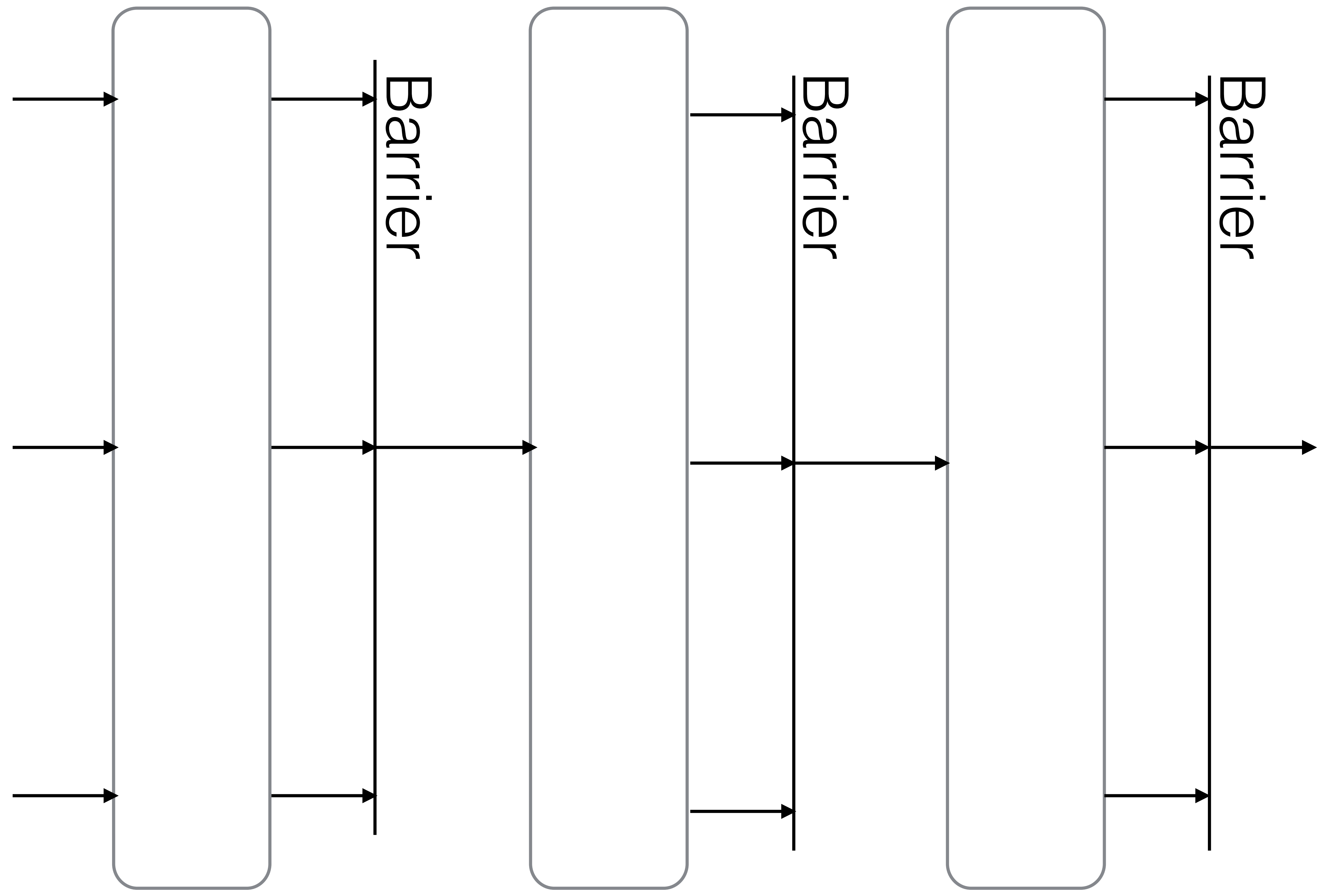
BSP

Question: What Execution Model to Use?

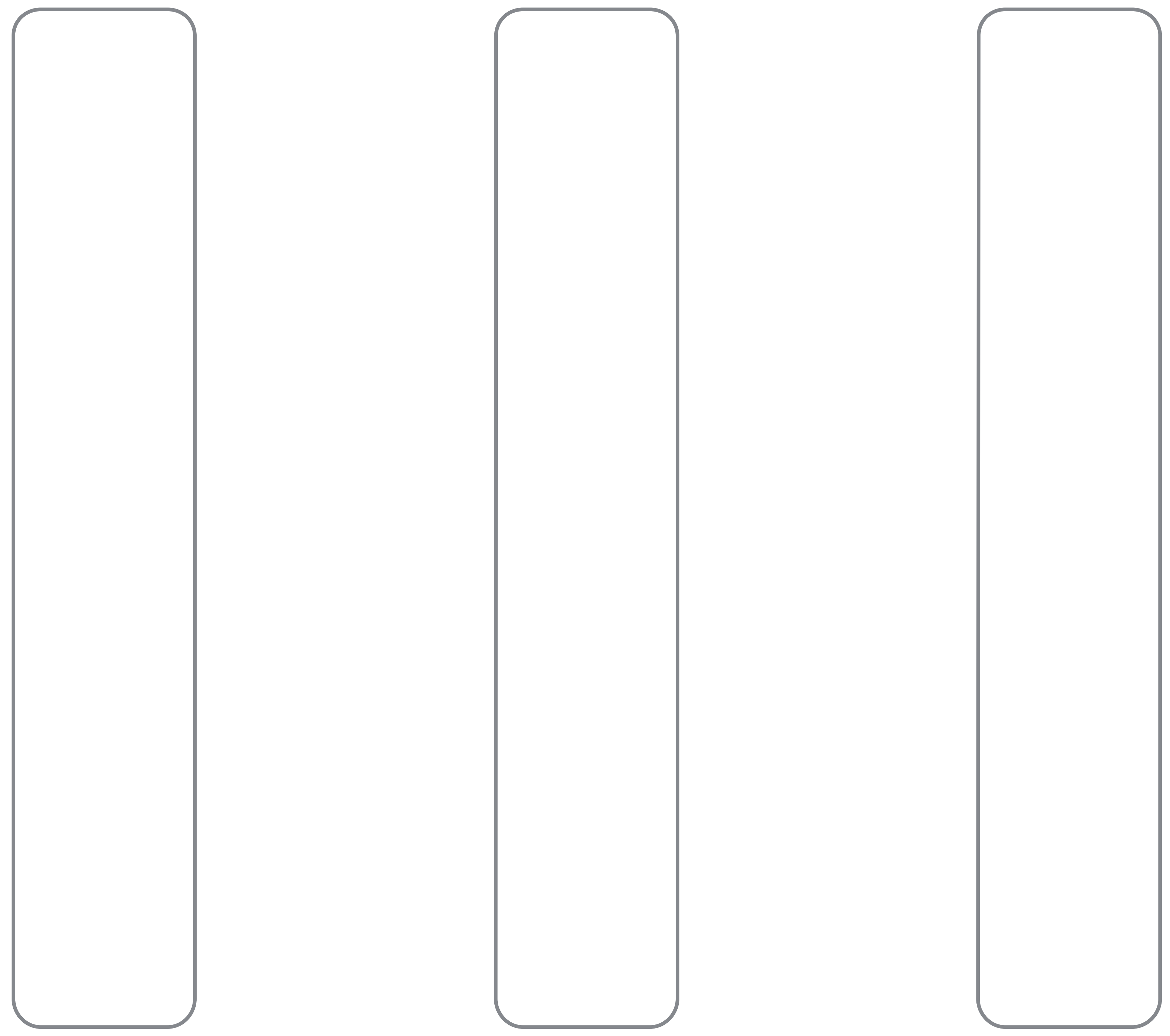


BSP

Question: What Execution Model to Use?

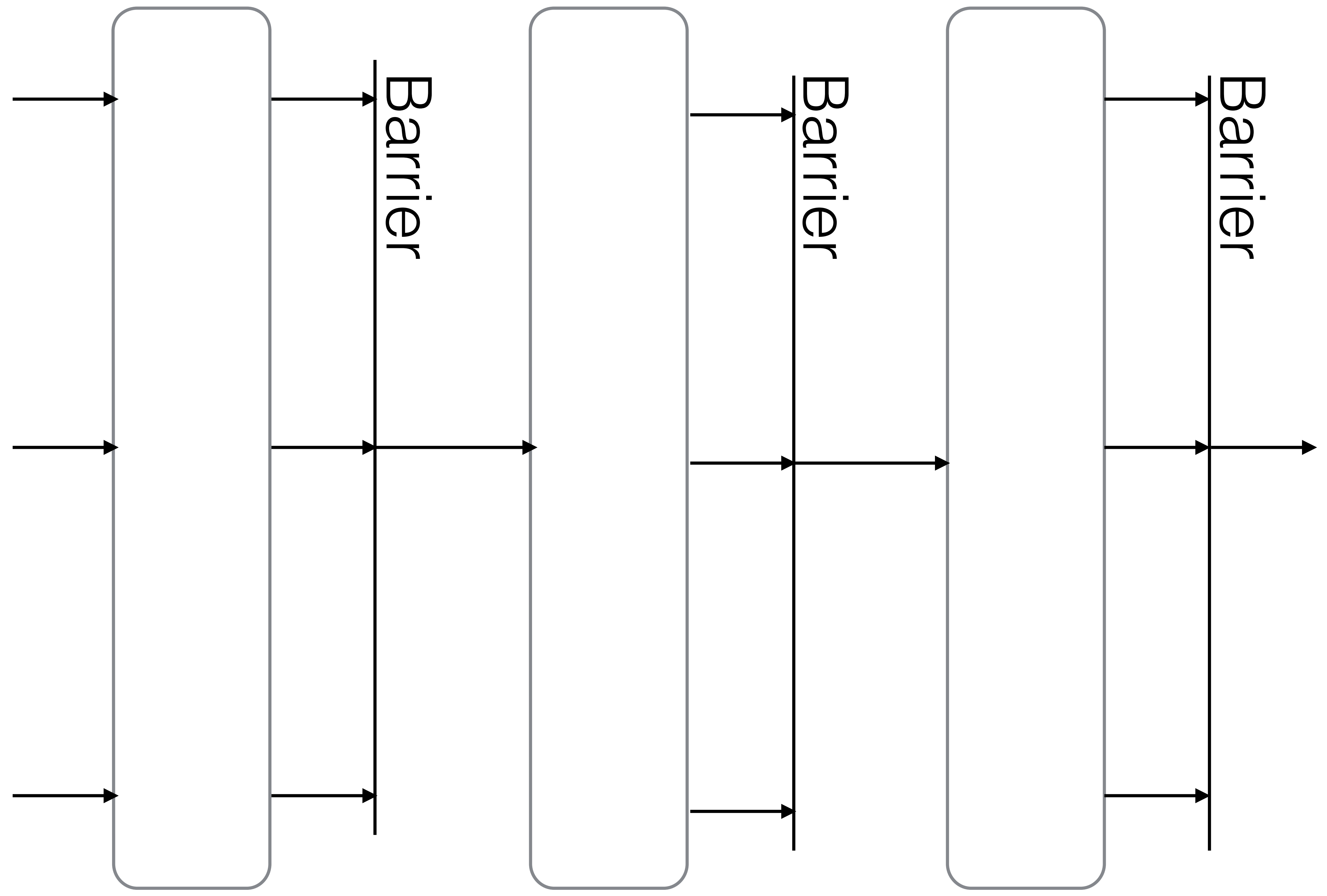


BSP

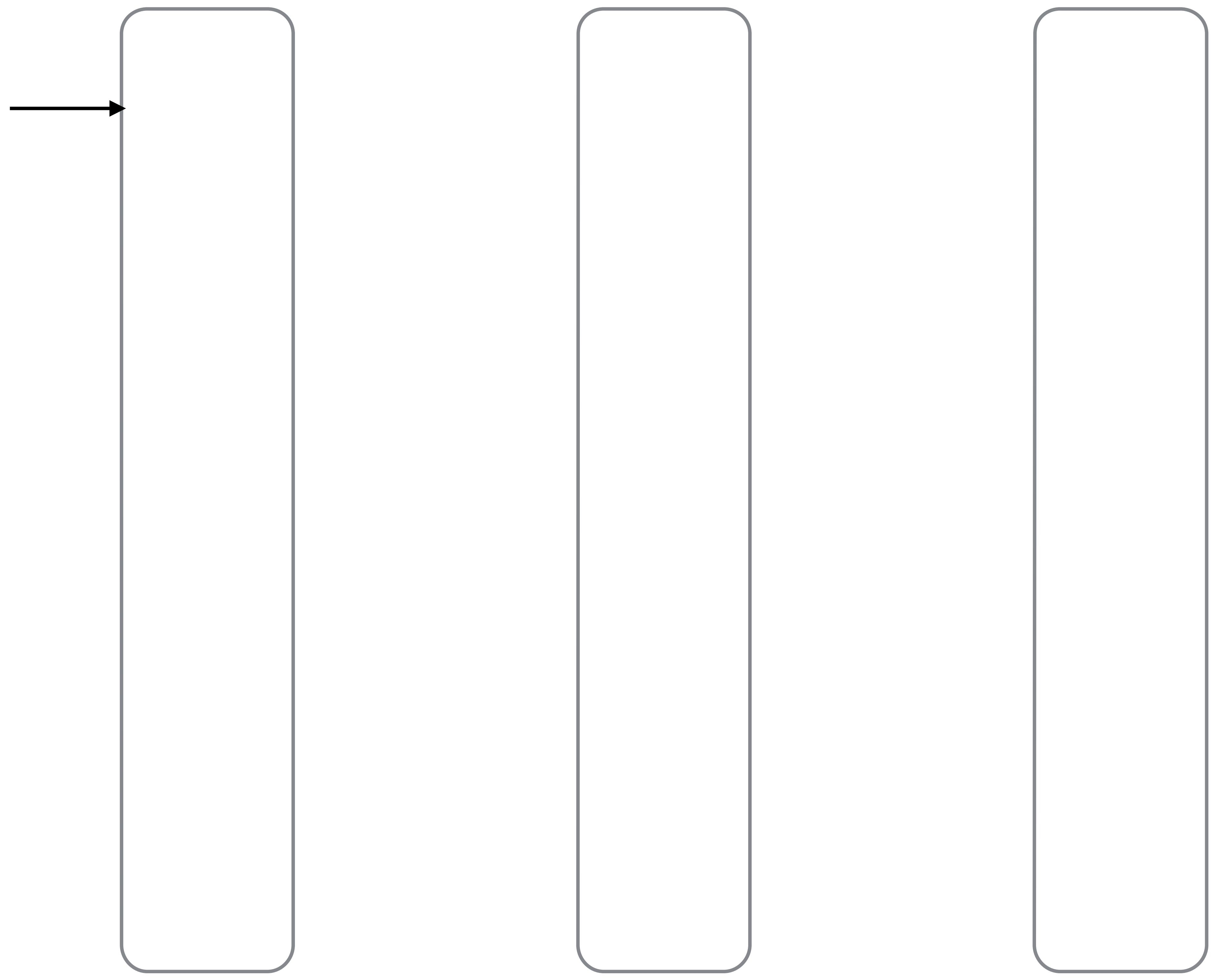


General Dataflow
Out-of-Order Processing

Question: What Execution Model to Use?

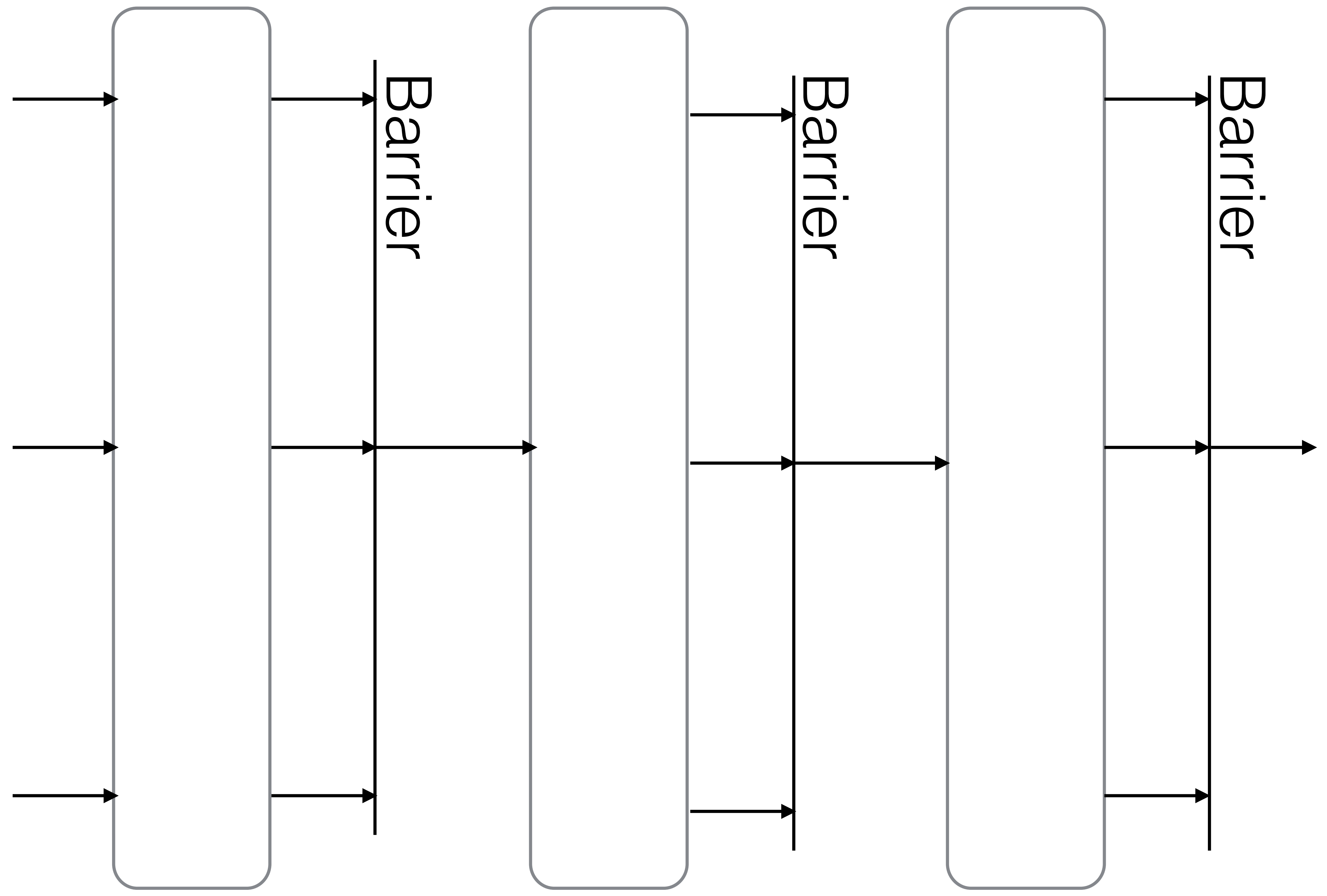


BSP

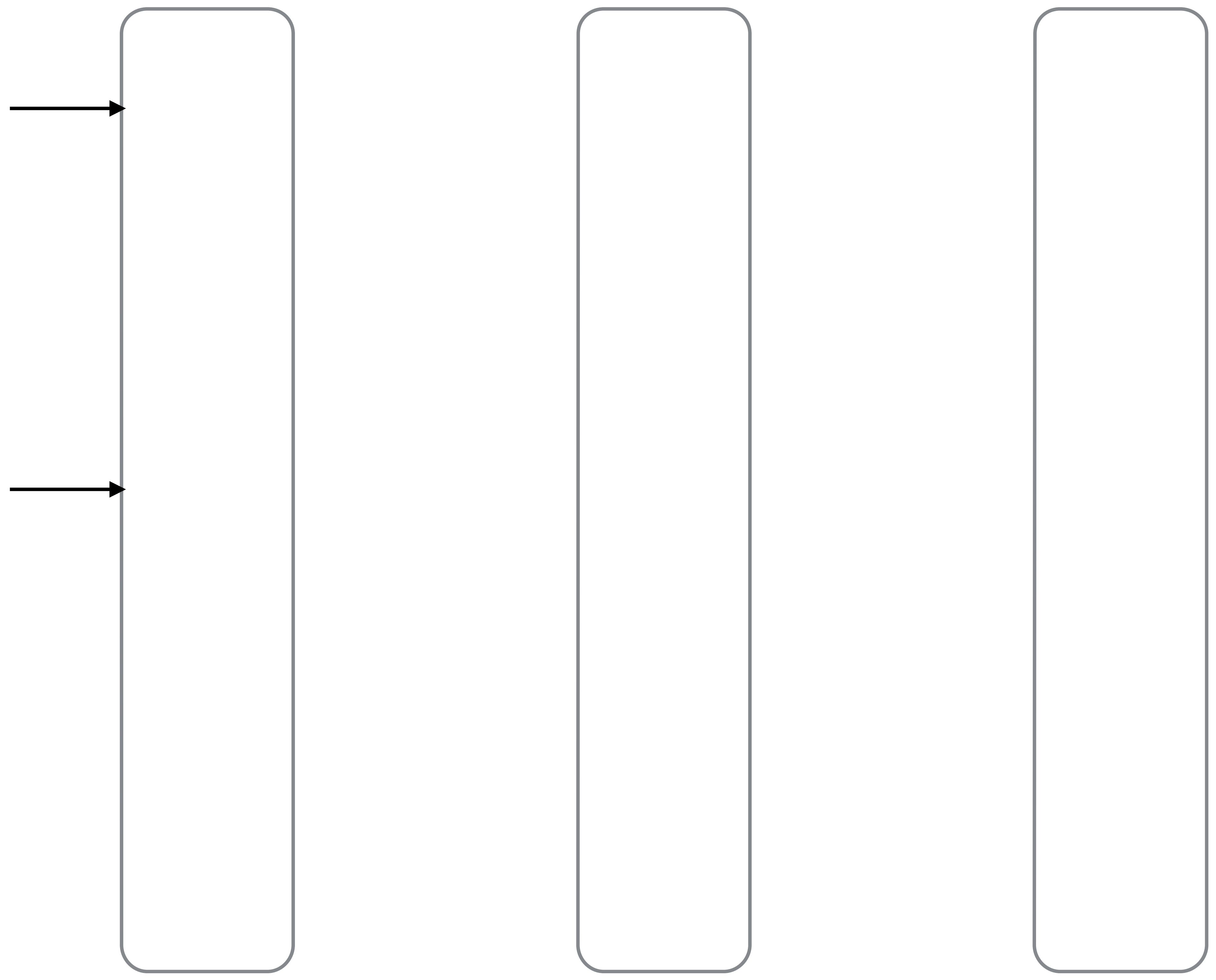


General Dataflow
Out-of-Order Processing

Question: What Execution Model to Use?

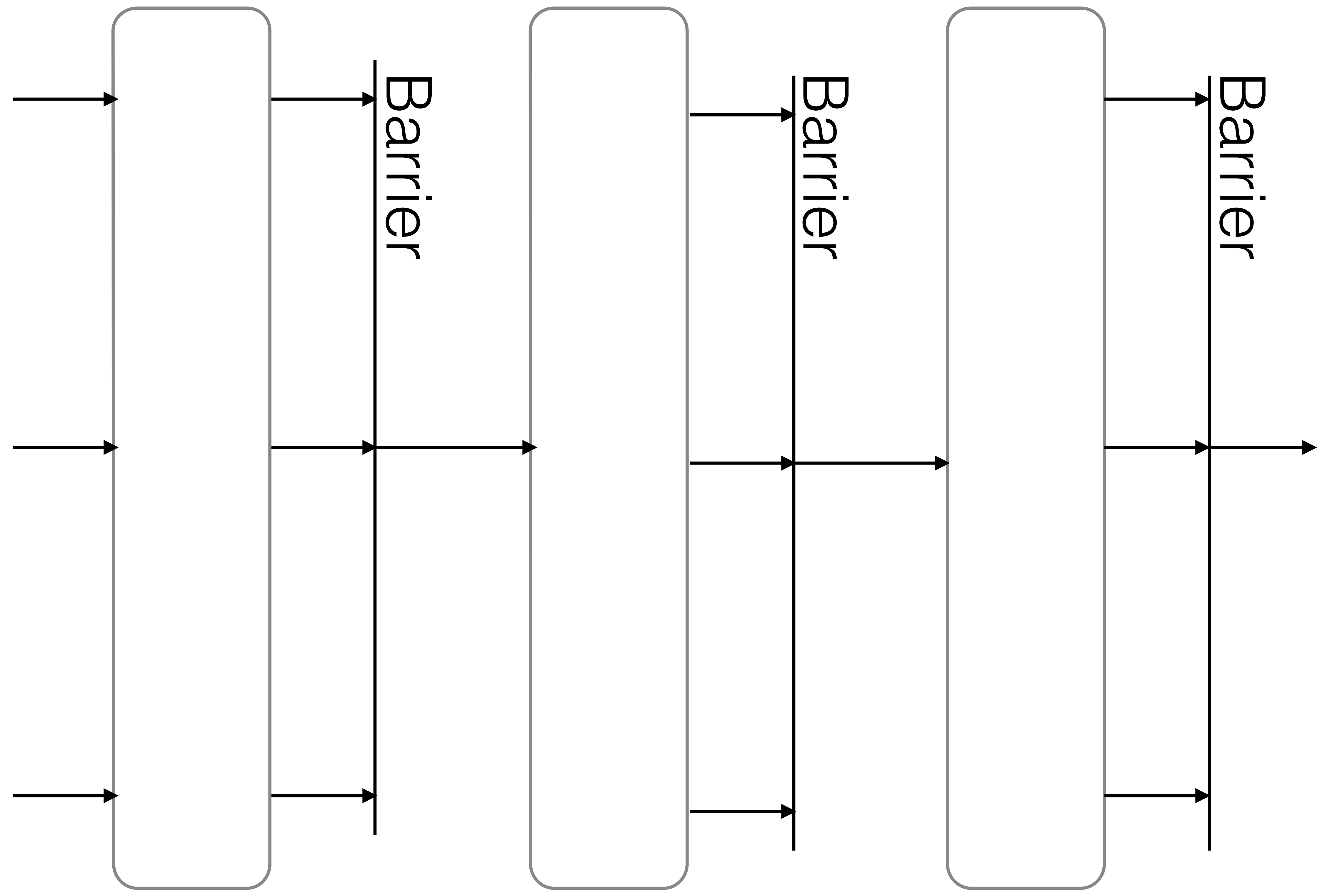


BSP

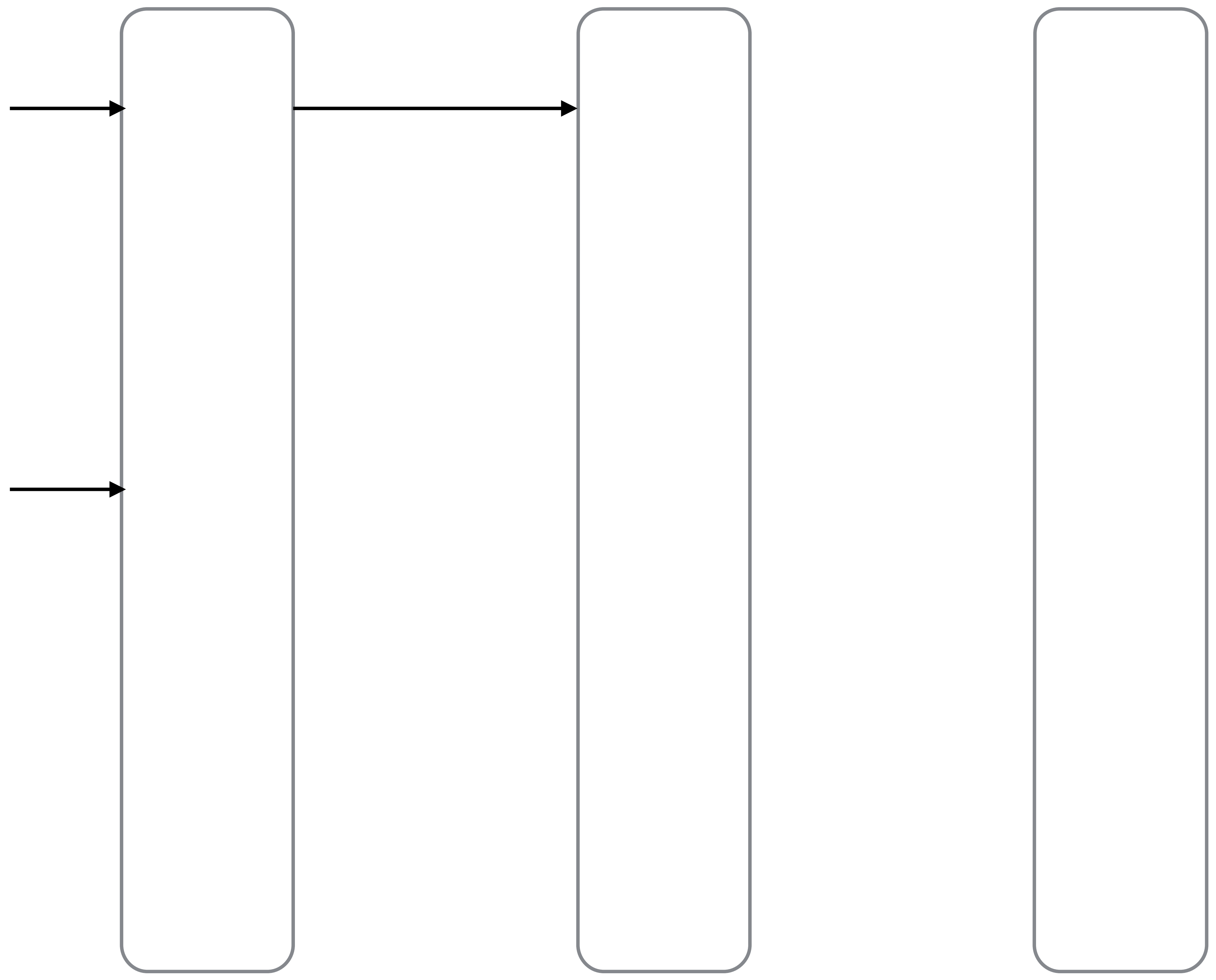


General Dataflow
Out-of-Order Processing

Question: What Execution Model to Use?

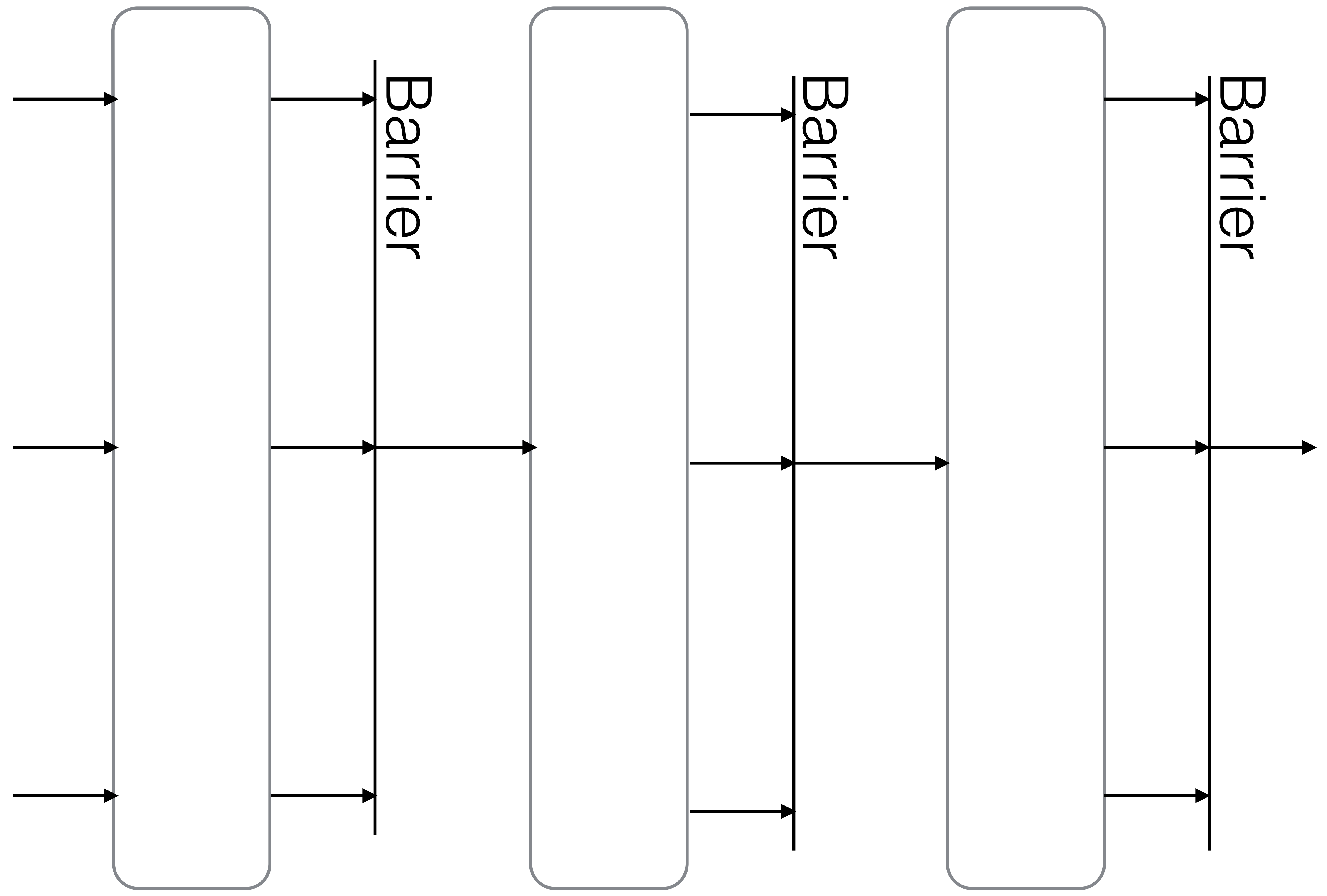


BSP

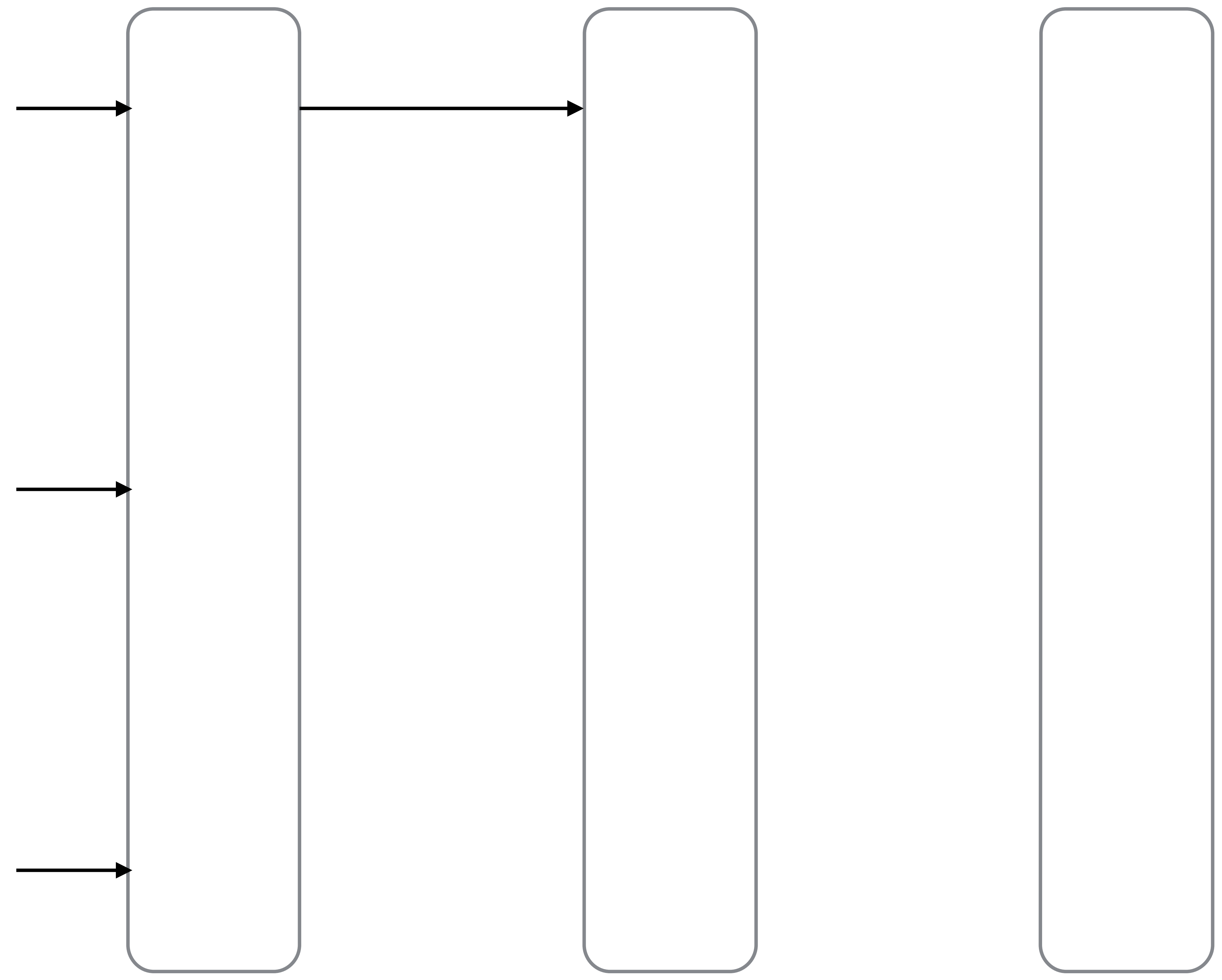


General Dataflow
Out-of-Order Processing

Question: What Execution Model to Use?

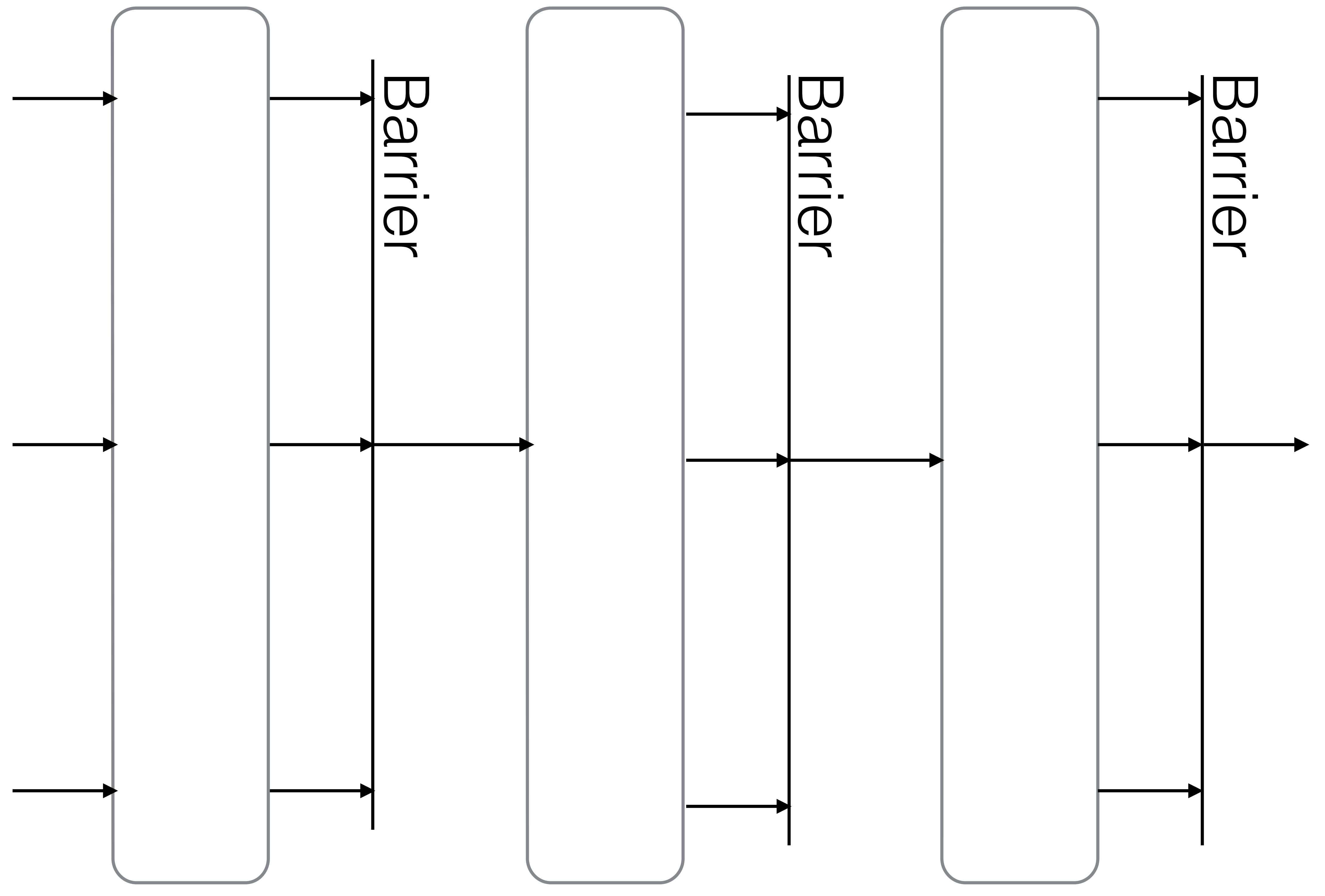


BSP

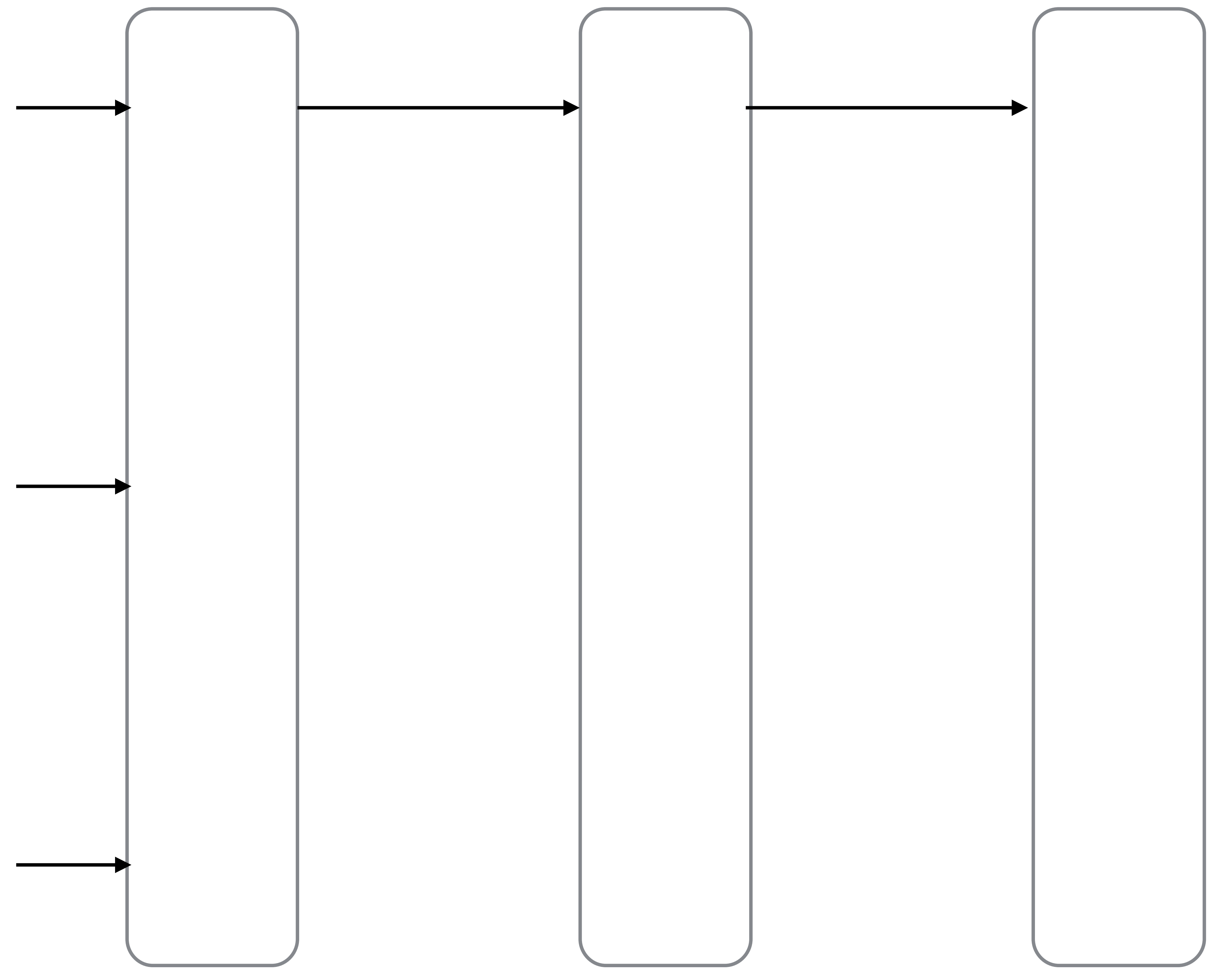


General Dataflow
Out-of-Order Processing

Question: What Execution Model to Use?

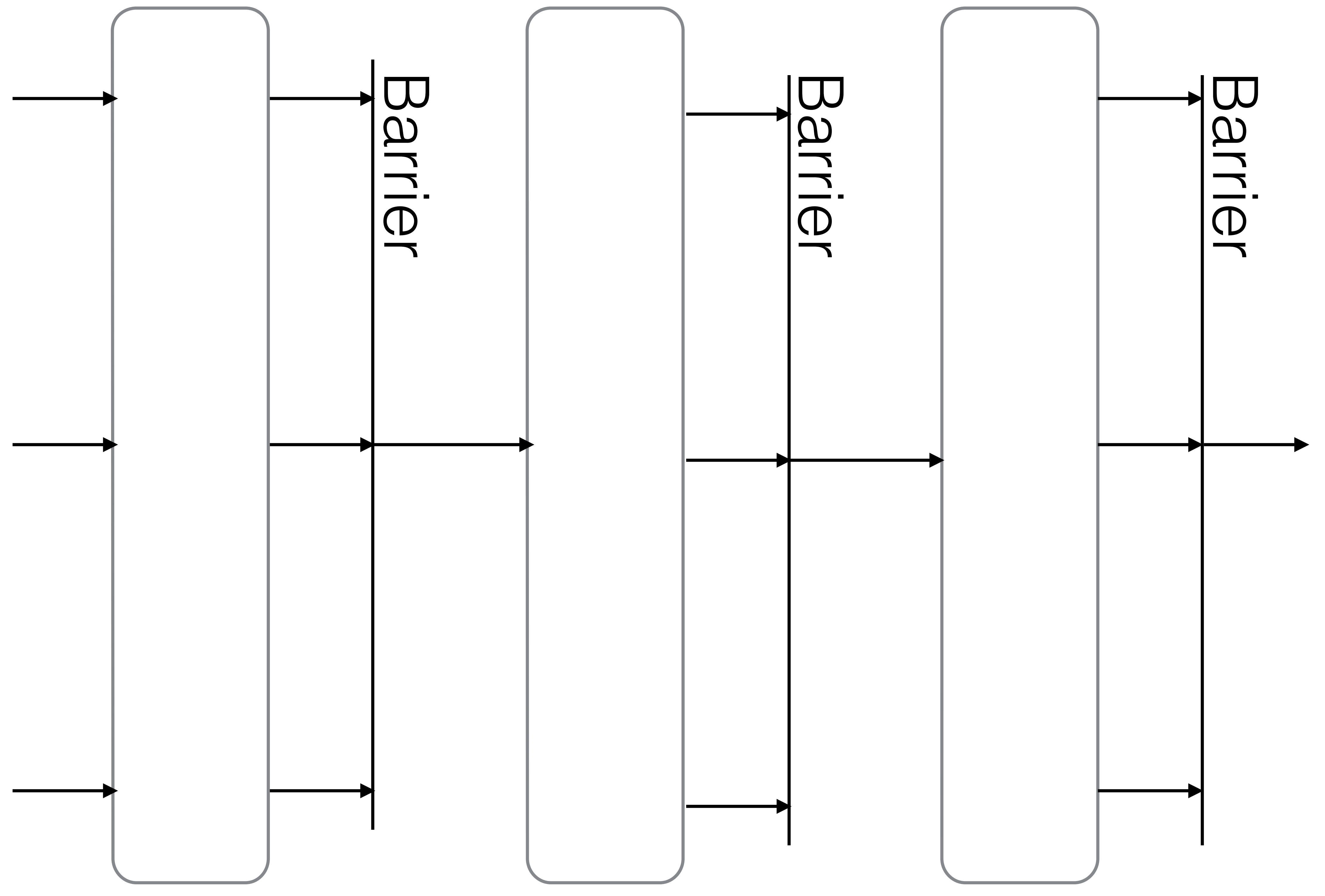


BSP

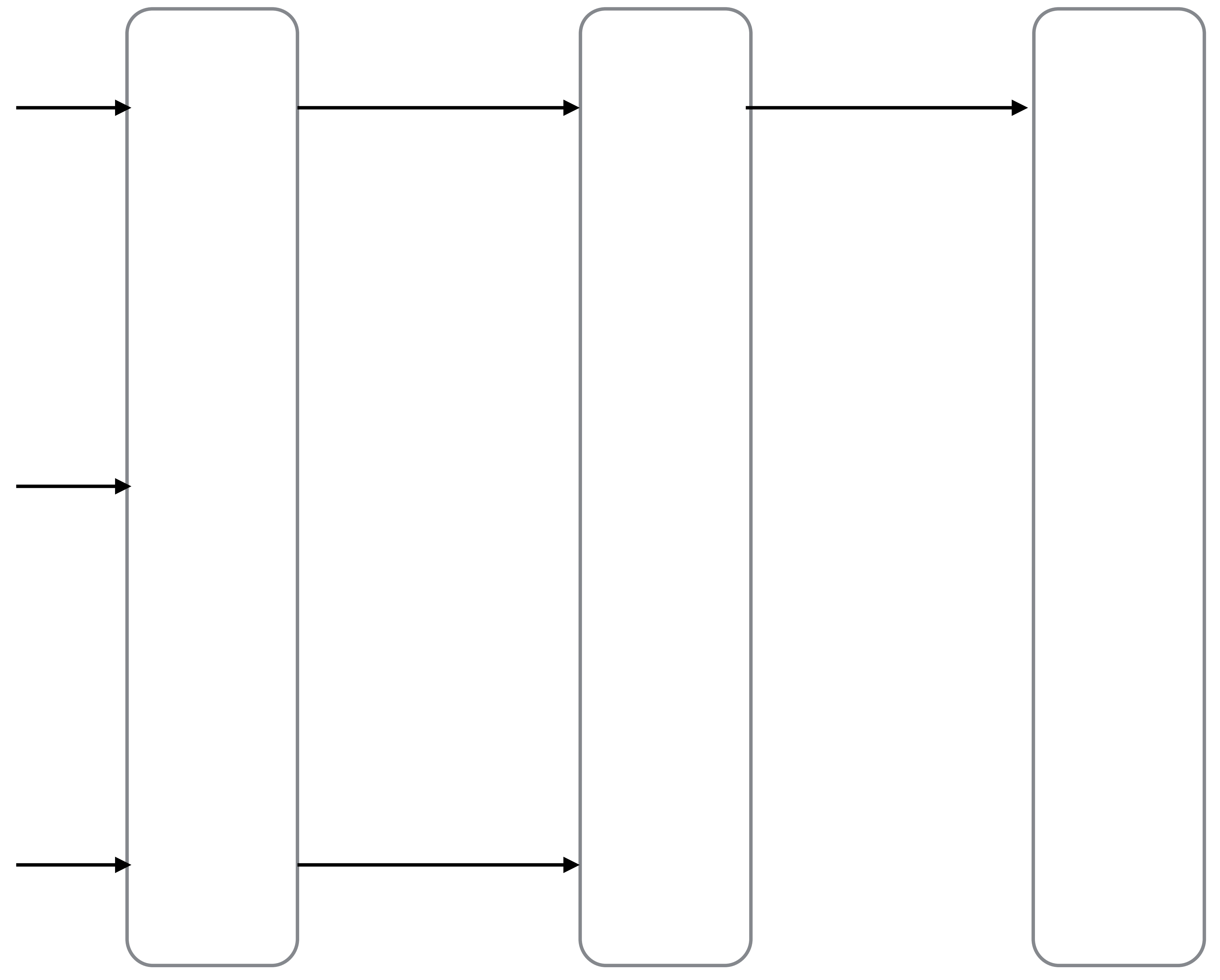


General Dataflow
Out-of-Order Processing

Question: What Execution Model to Use?

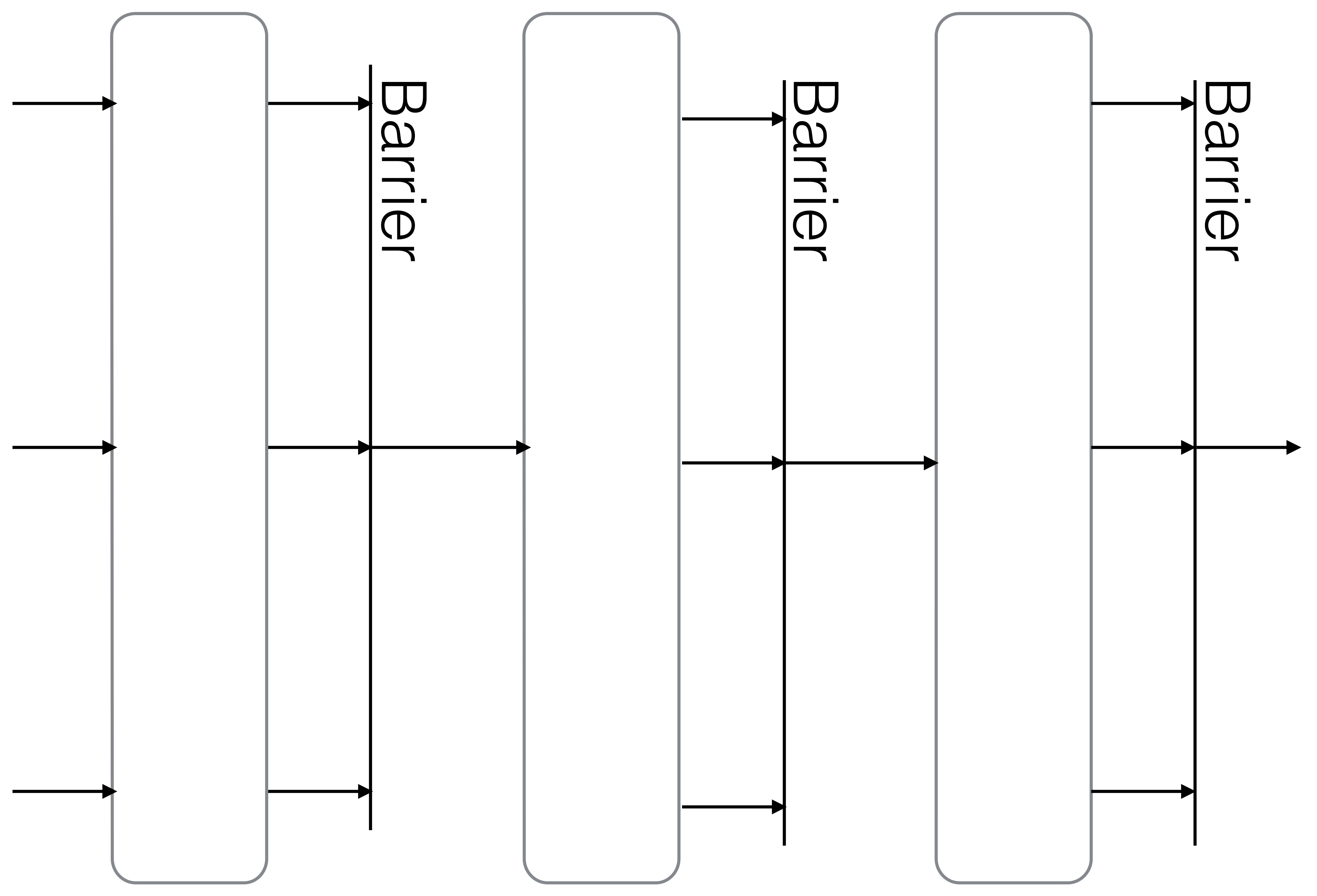


BSP

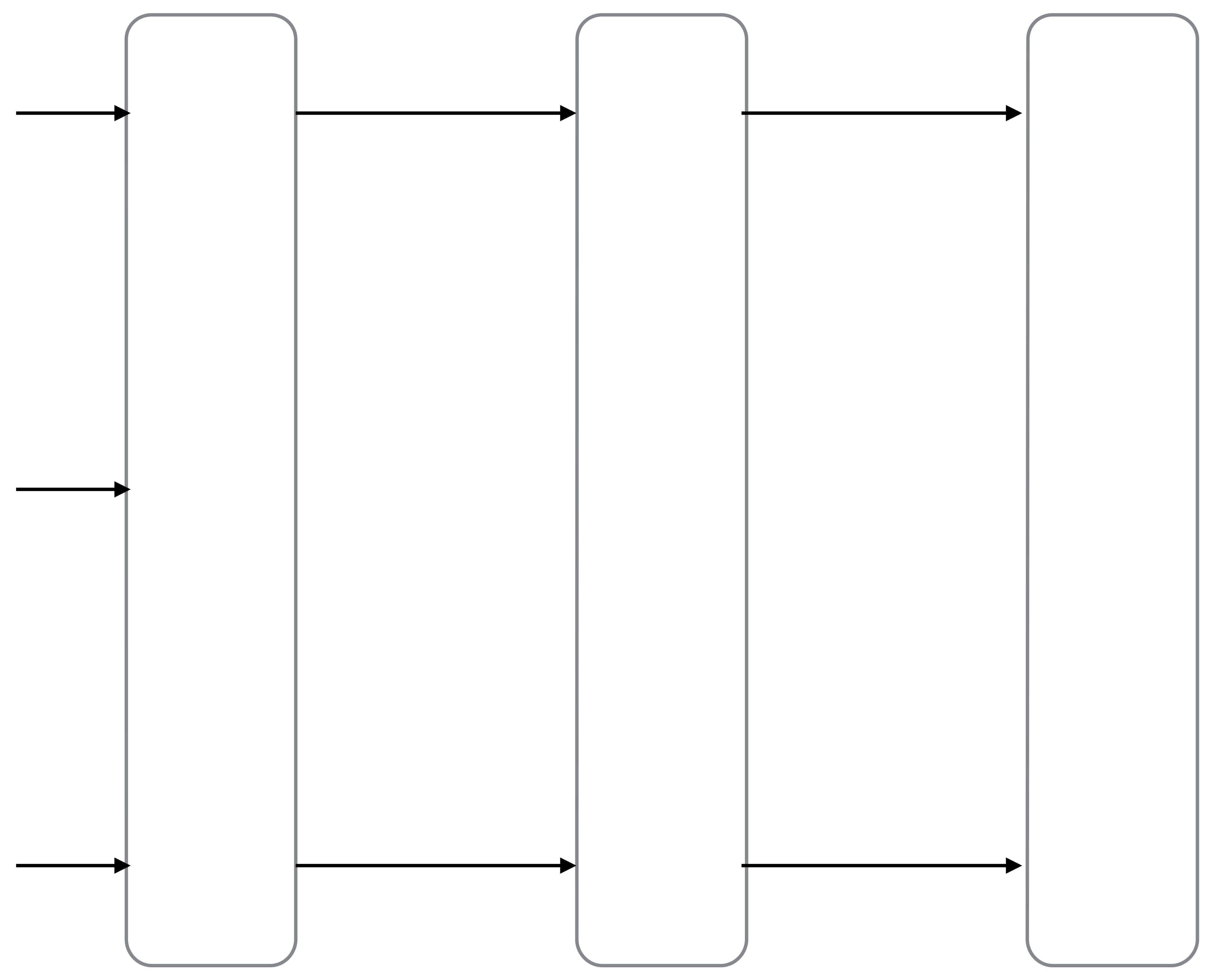


General Dataflow
Out-of-Order Processing

Question: What Execution Model to Use?

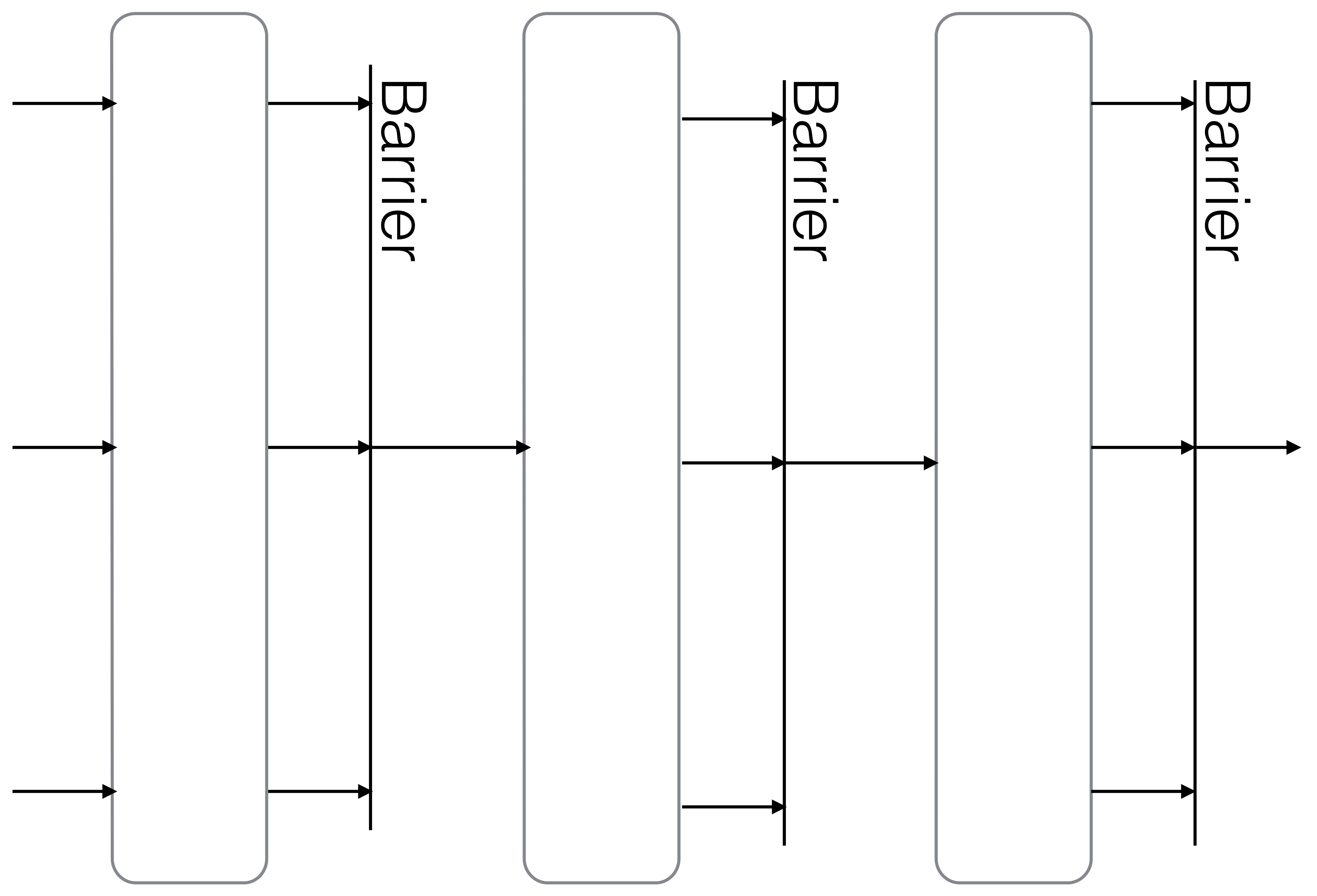


BSP

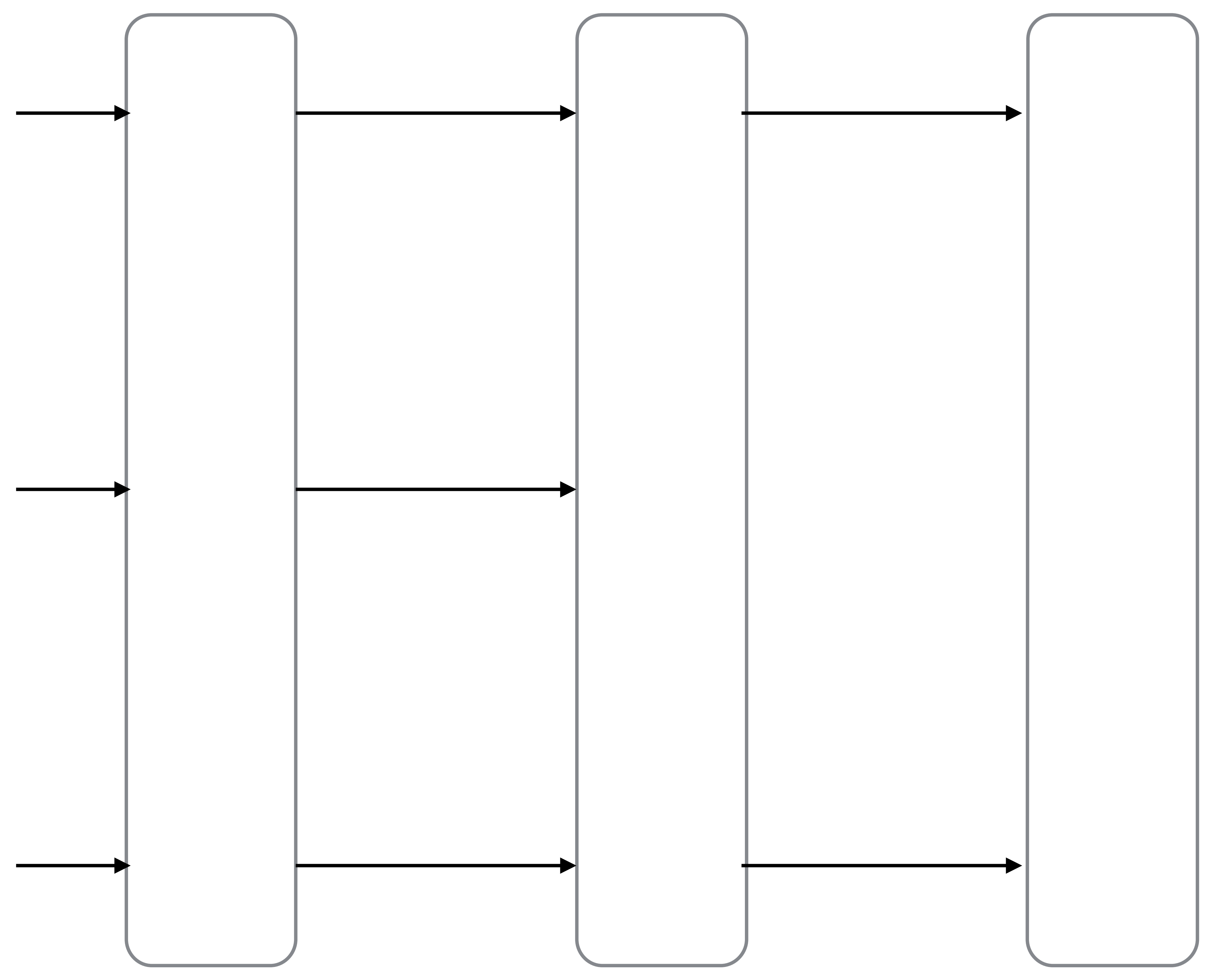


General Dataflow
Out-of-Order Processing

Question: What Execution Model to Use?

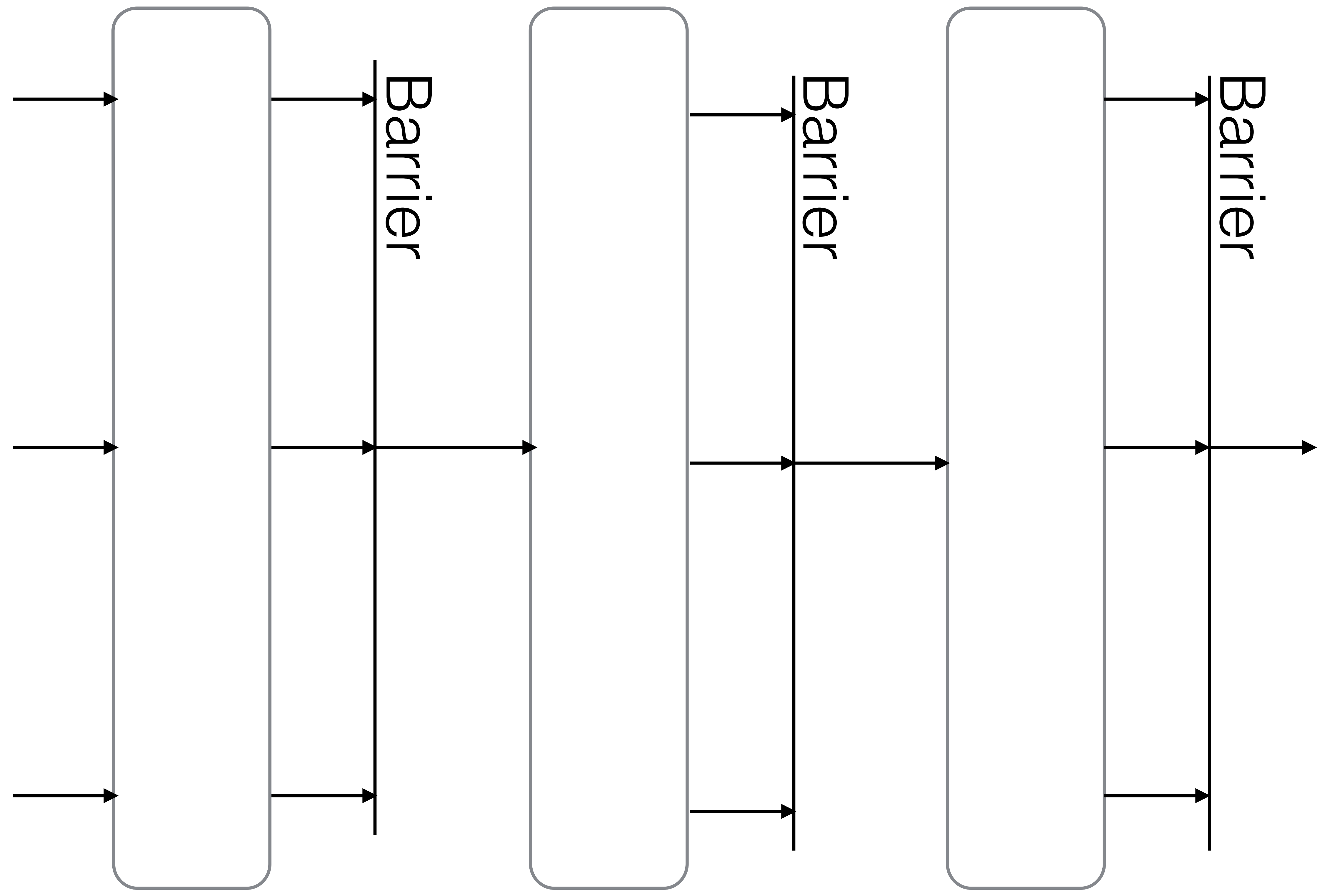


BSP

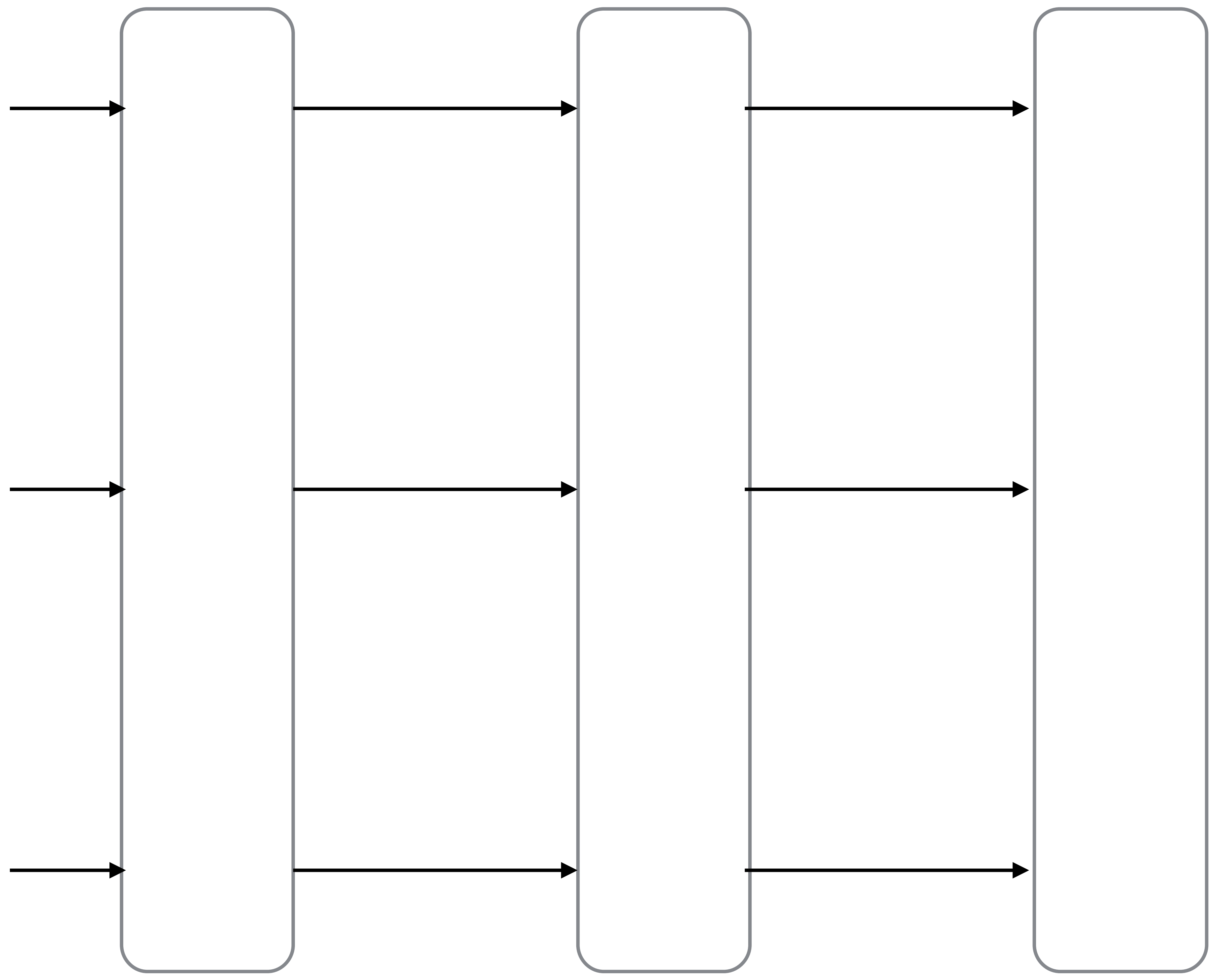


General Dataflow
Out-of-Order Processing

Question: What Execution Model to Use?

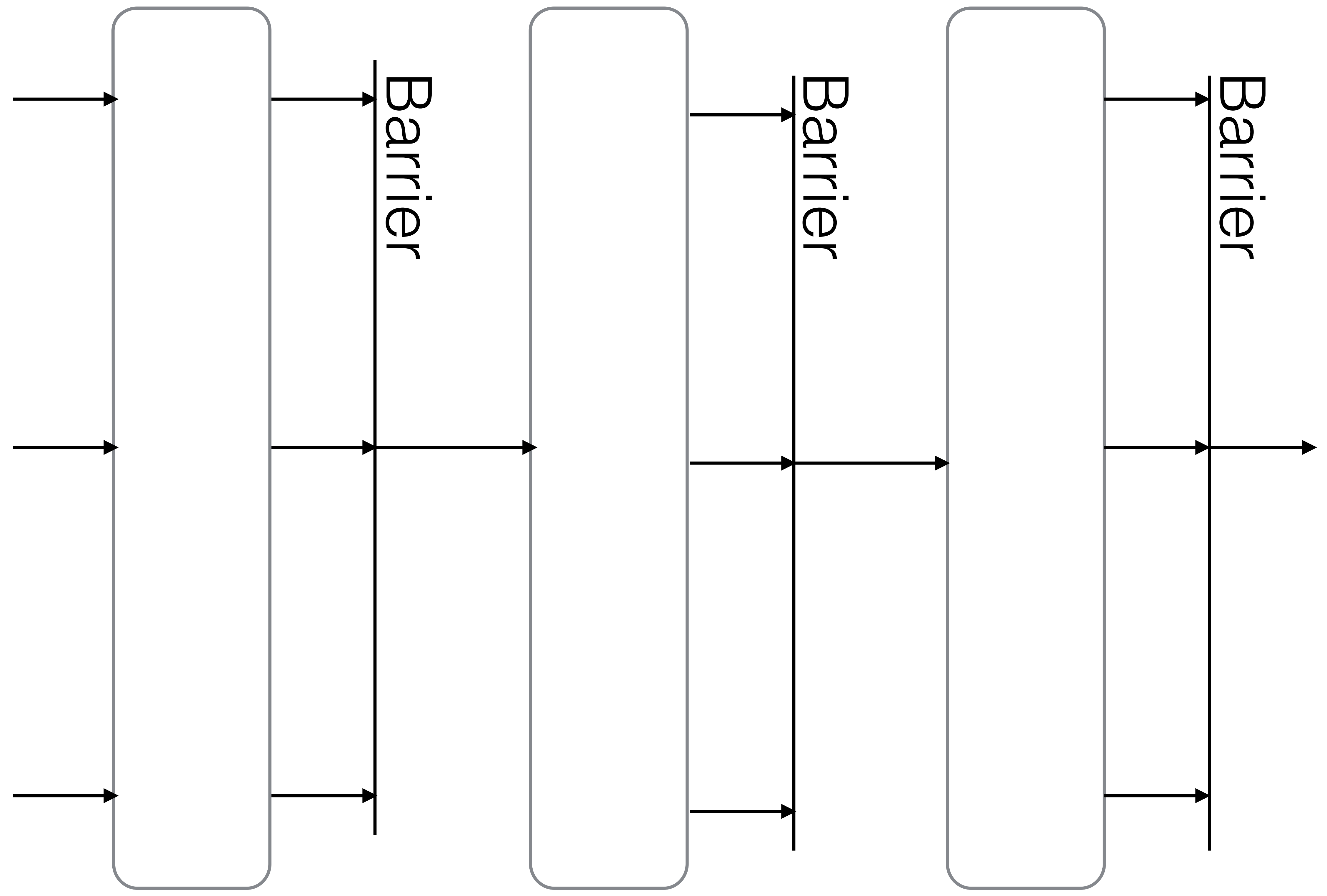


BSP

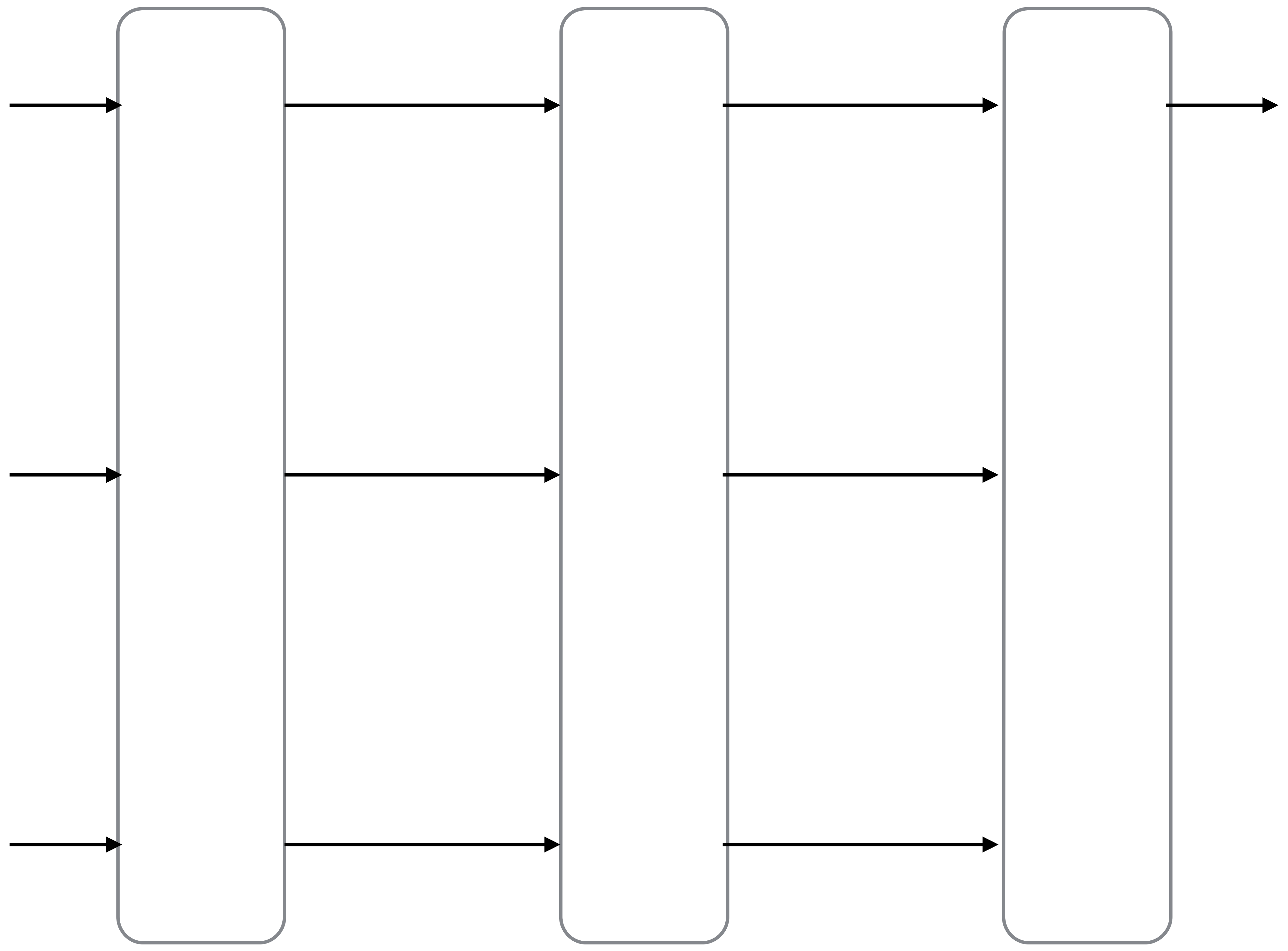


General Dataflow
Out-of-Order Processing

Question: What Execution Model to Use?

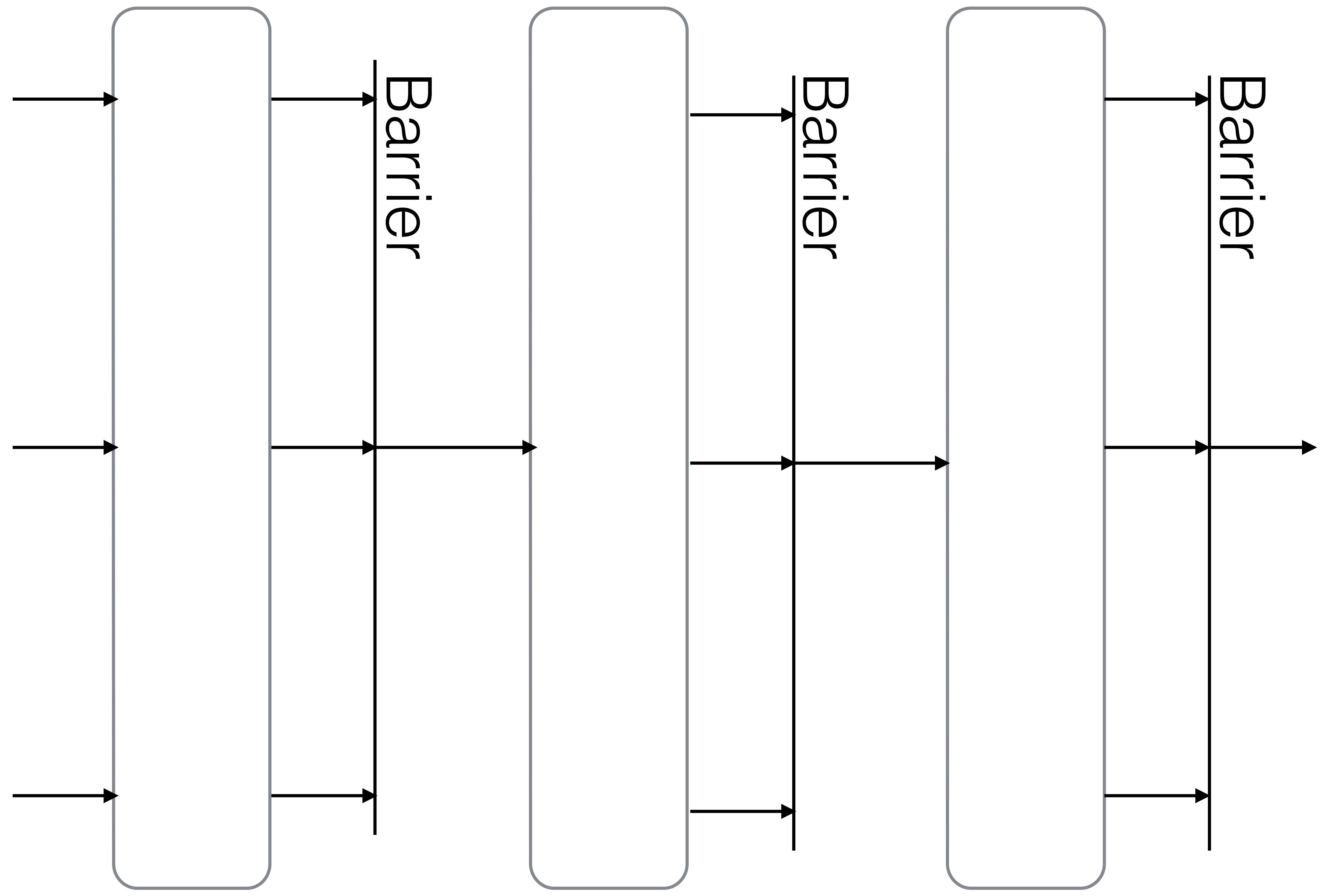


BSP

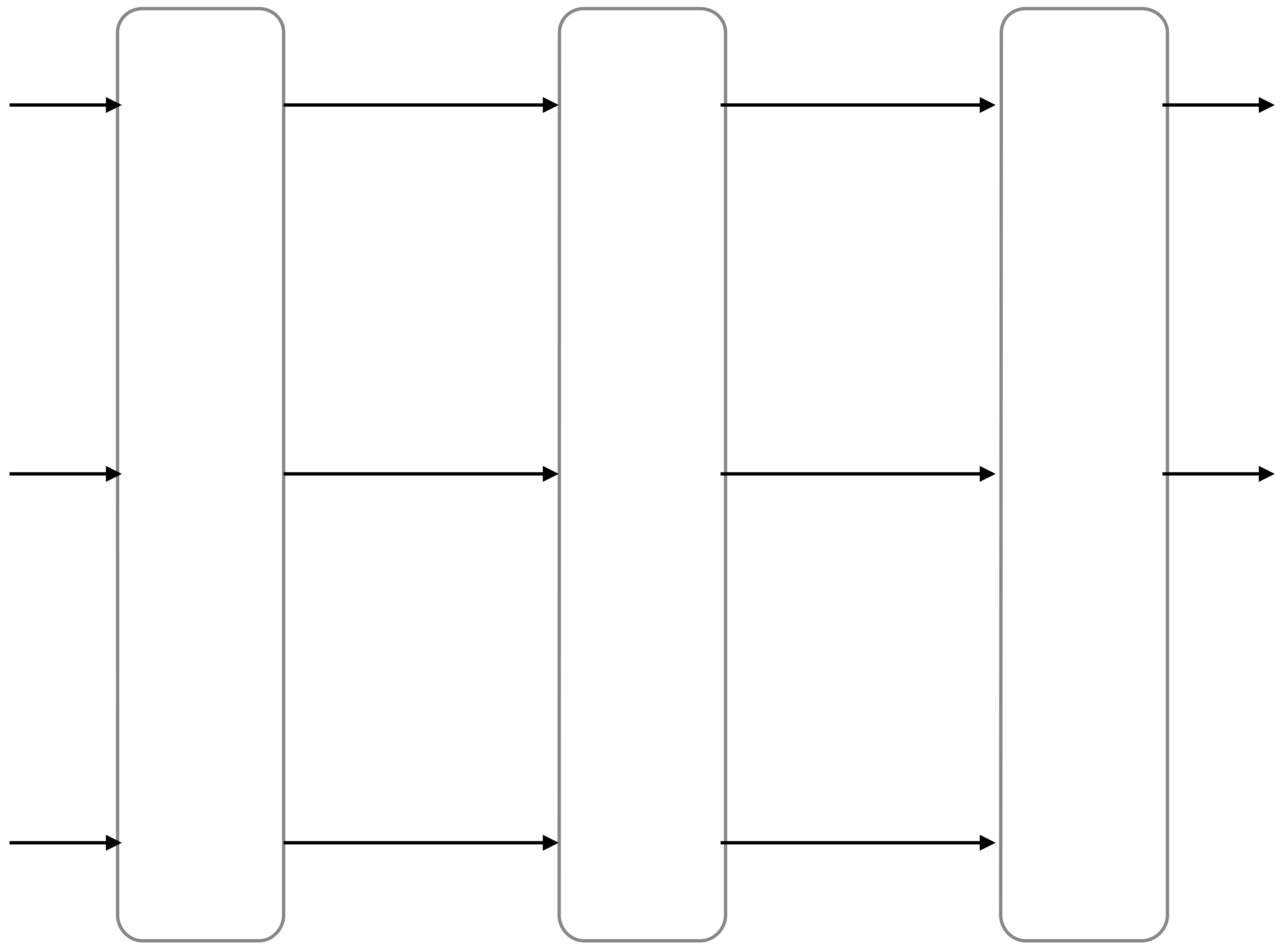


General Dataflow
Out-of-Order Processing

Question: What Execution Model to Use?

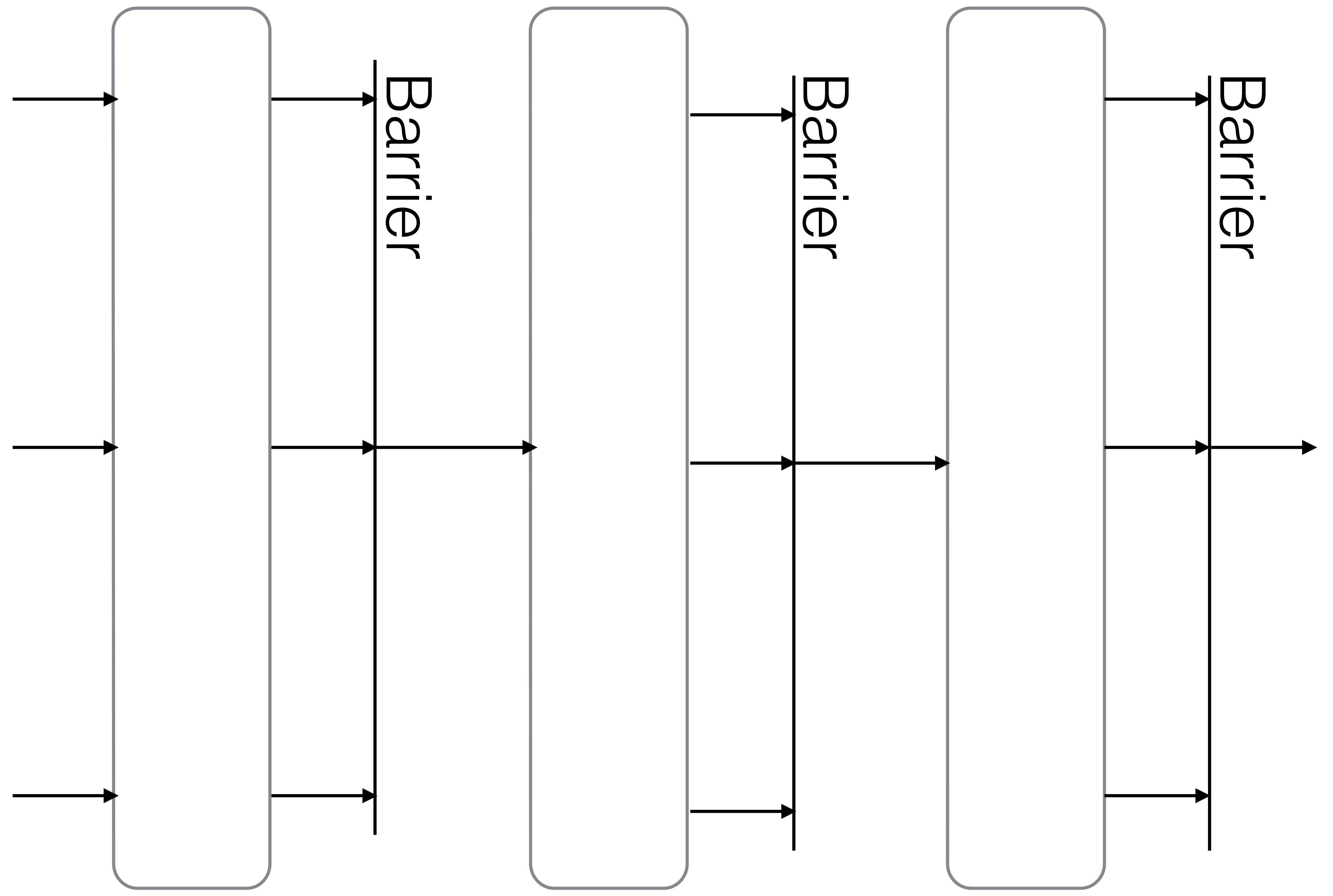


BSP

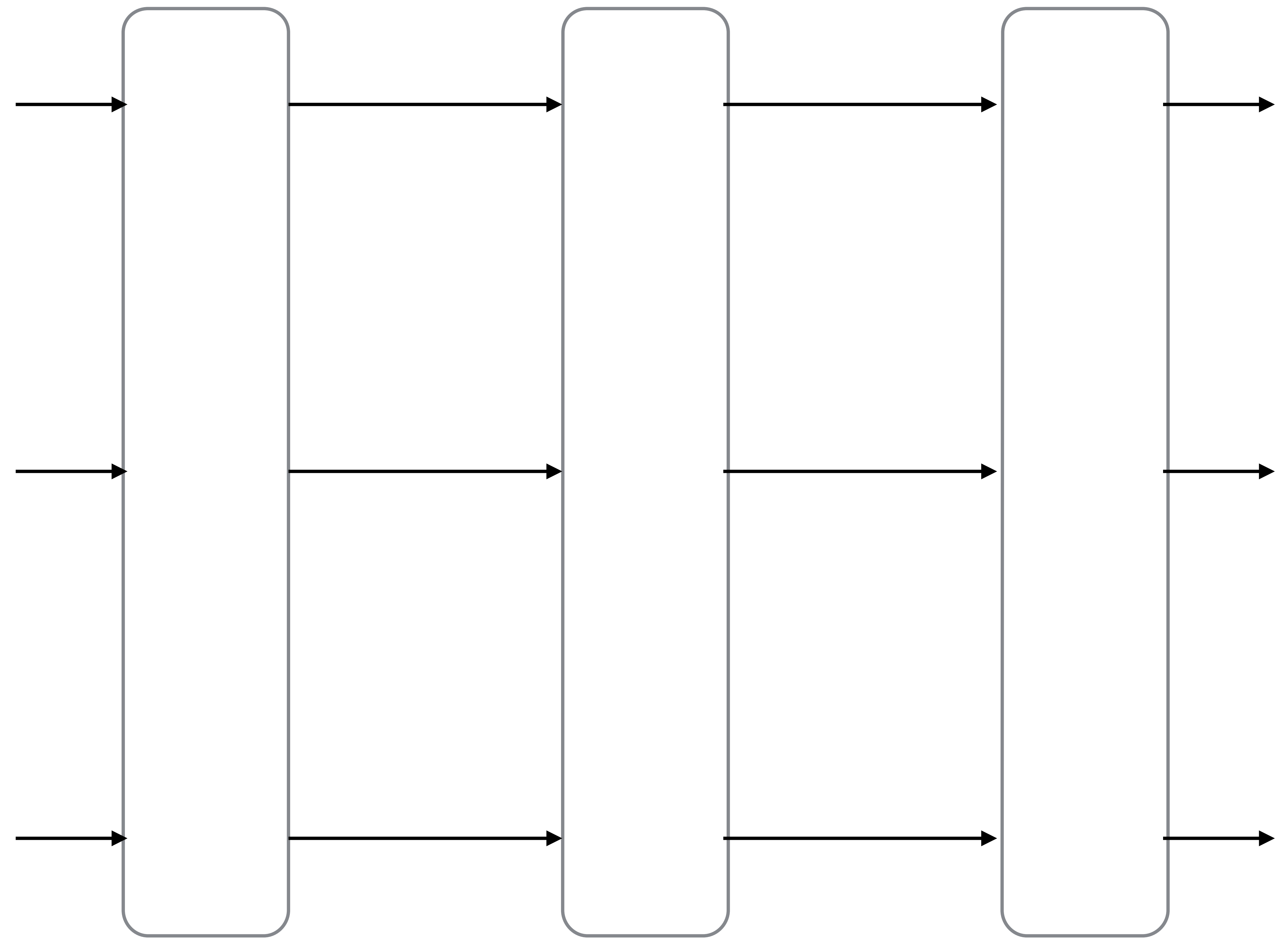


General Dataflow
Out-of-Order Processing

Question: What Execution Model to Use?



BSP



General Dataflow
Out-of-Order Processing

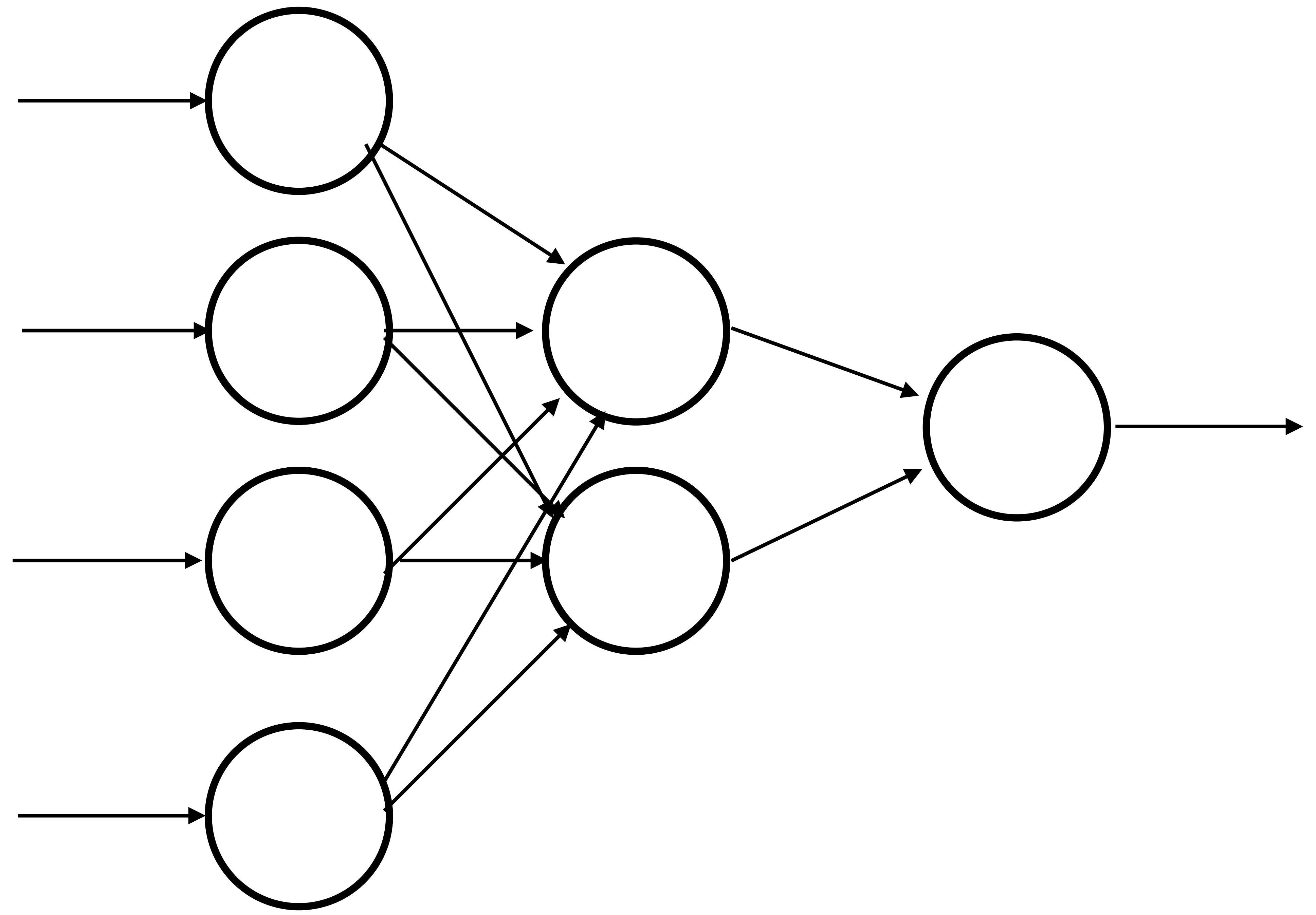
Question: What Execution Model to Use?

	BSP	Out-of-Order Processing
Batch	✔	✔
Iterative	✔	✔
Streaming	? (latency)	✔
Graph Processing	?	?

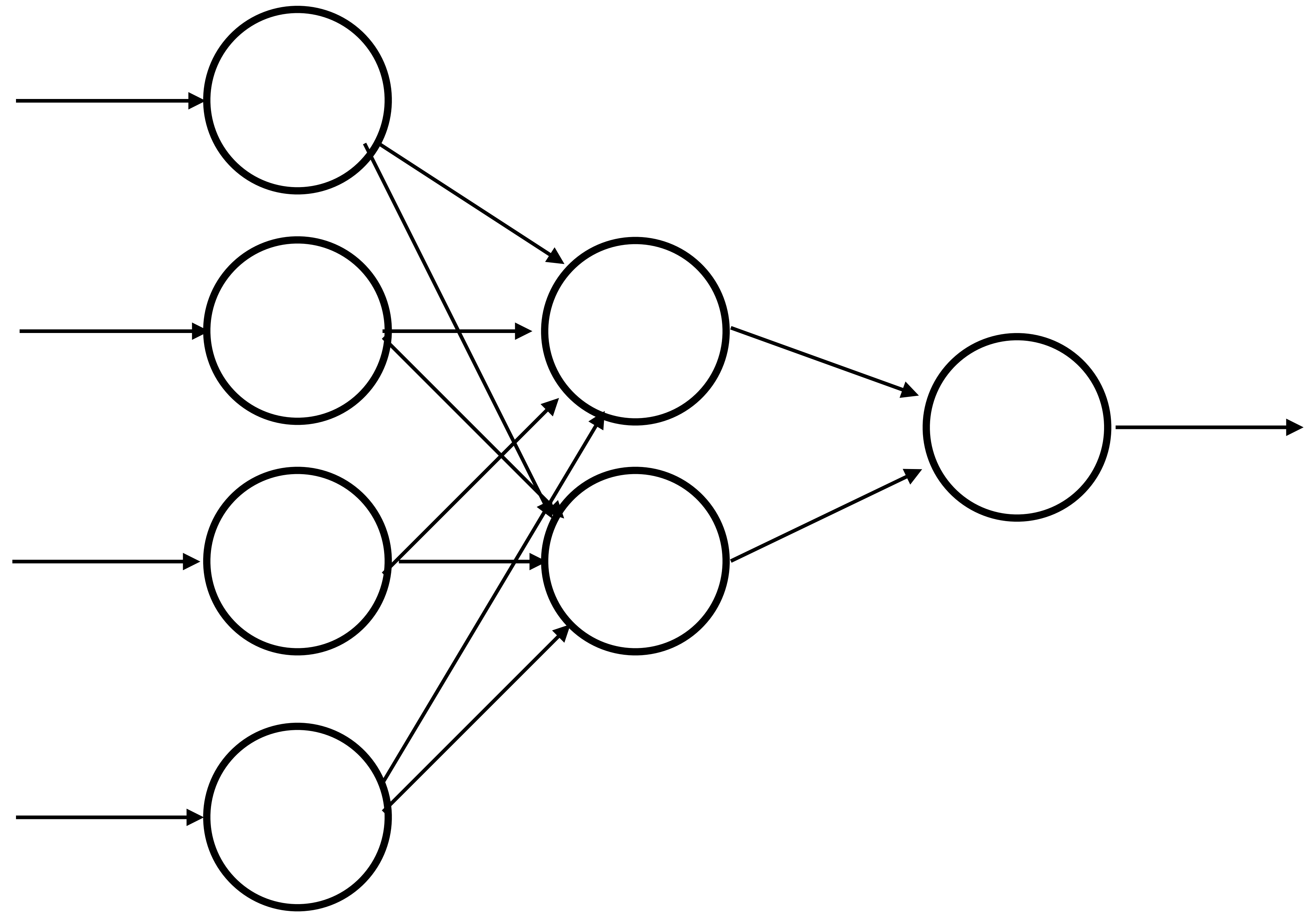
Question: What Execution Model to Use?

	BSP	**Out-of-Order Processing**
Fault Tolerance	Simple: lineage for stages, stateless nodes	Hard: Stateful nodes, need to checkpoint.
Programming Model	Simple?	Don't program directly, execution engine.
Performance	Potentially slow: wait at barriers.	Potentially faster: Avoid synchronization and ordering when possible.

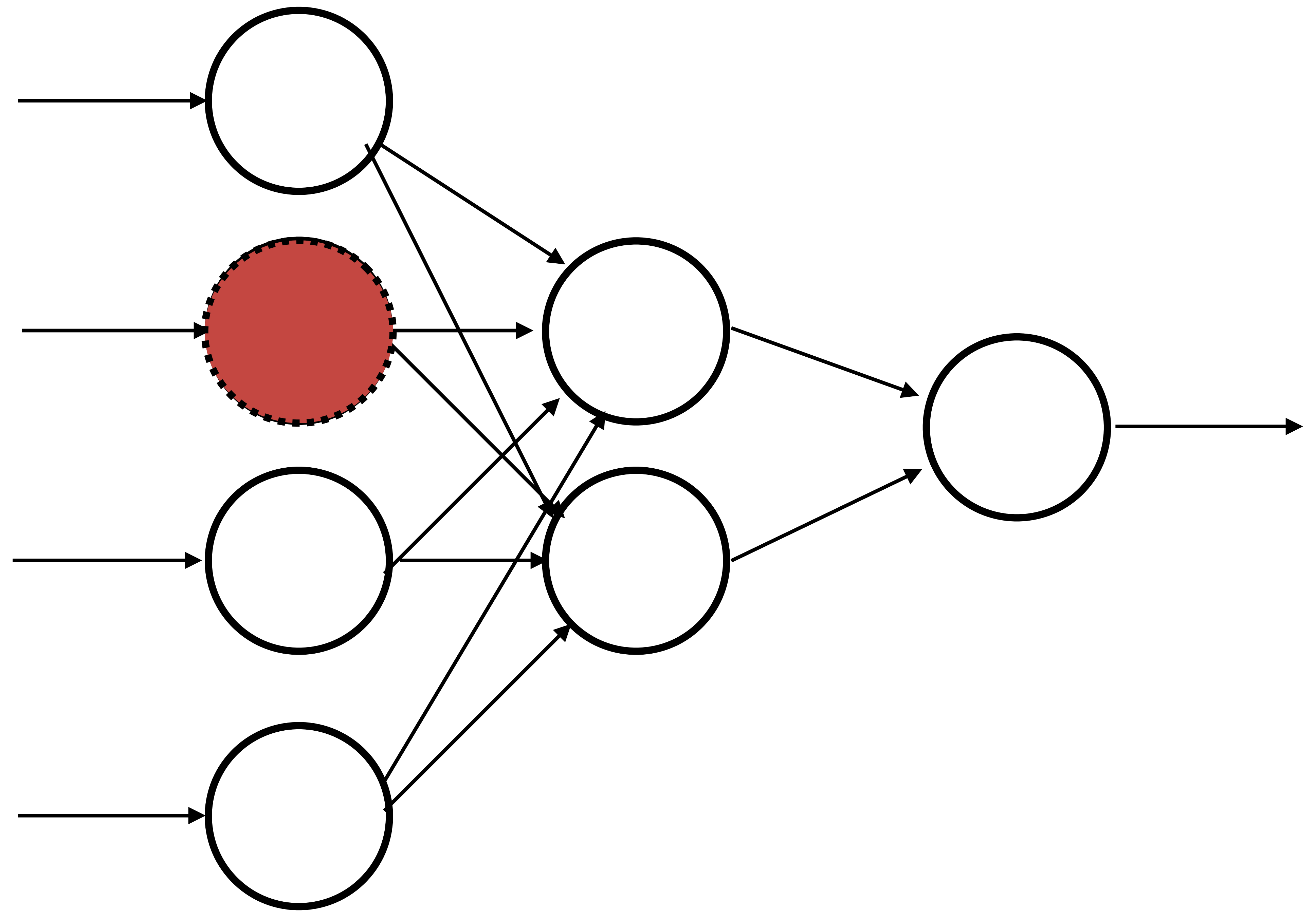
A Closer Look at Performance



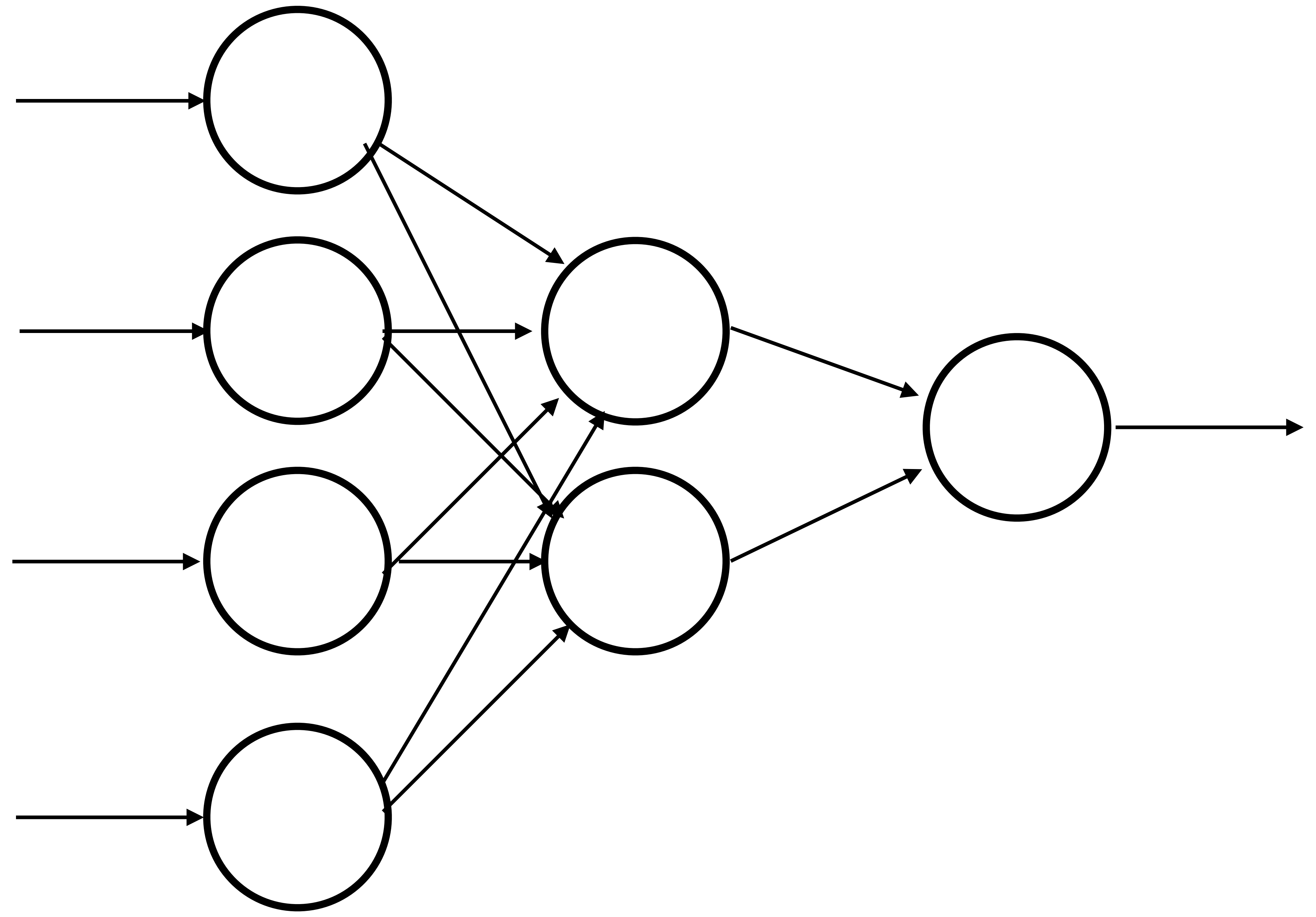
BSP



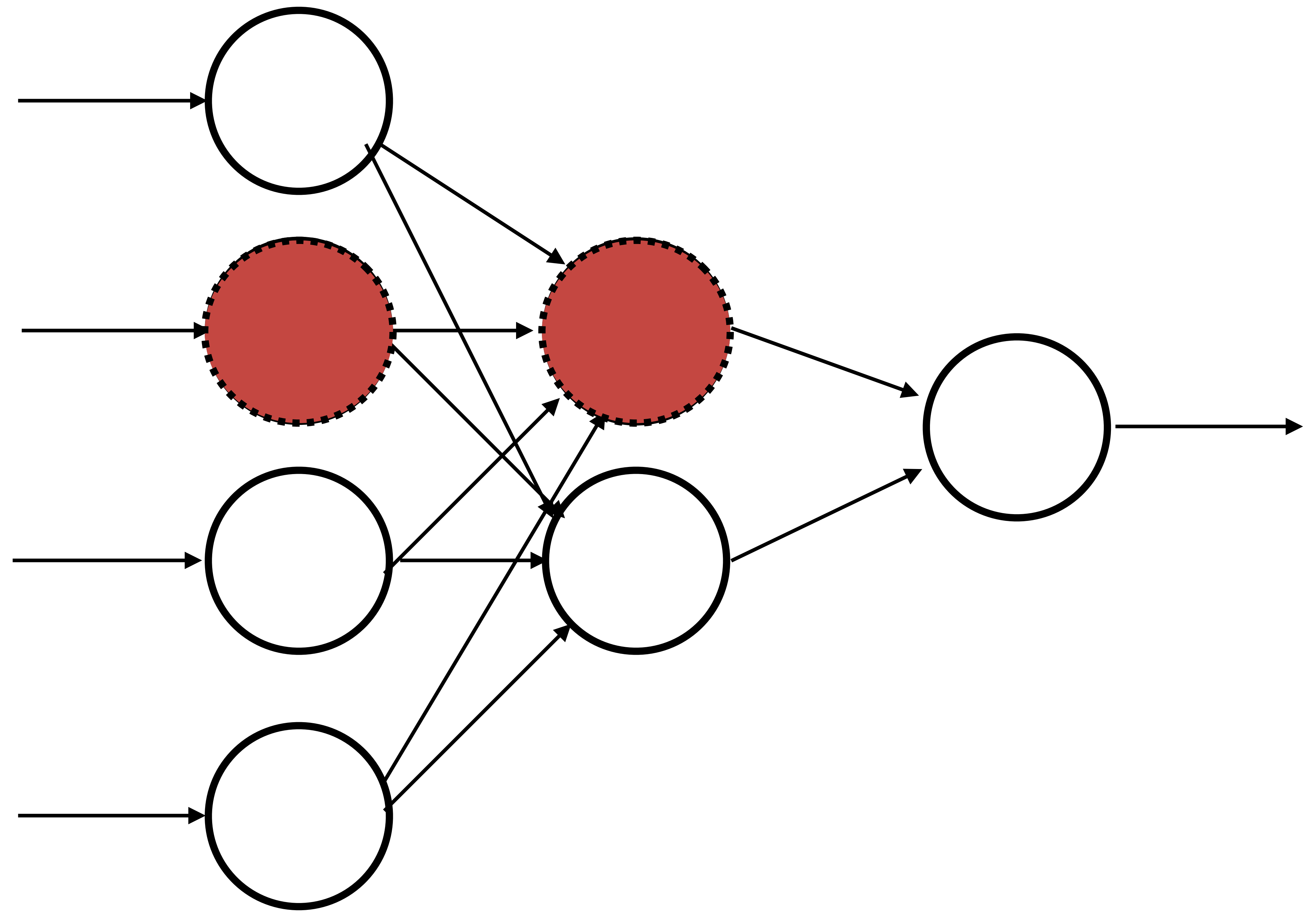
A Closer Look at Performance



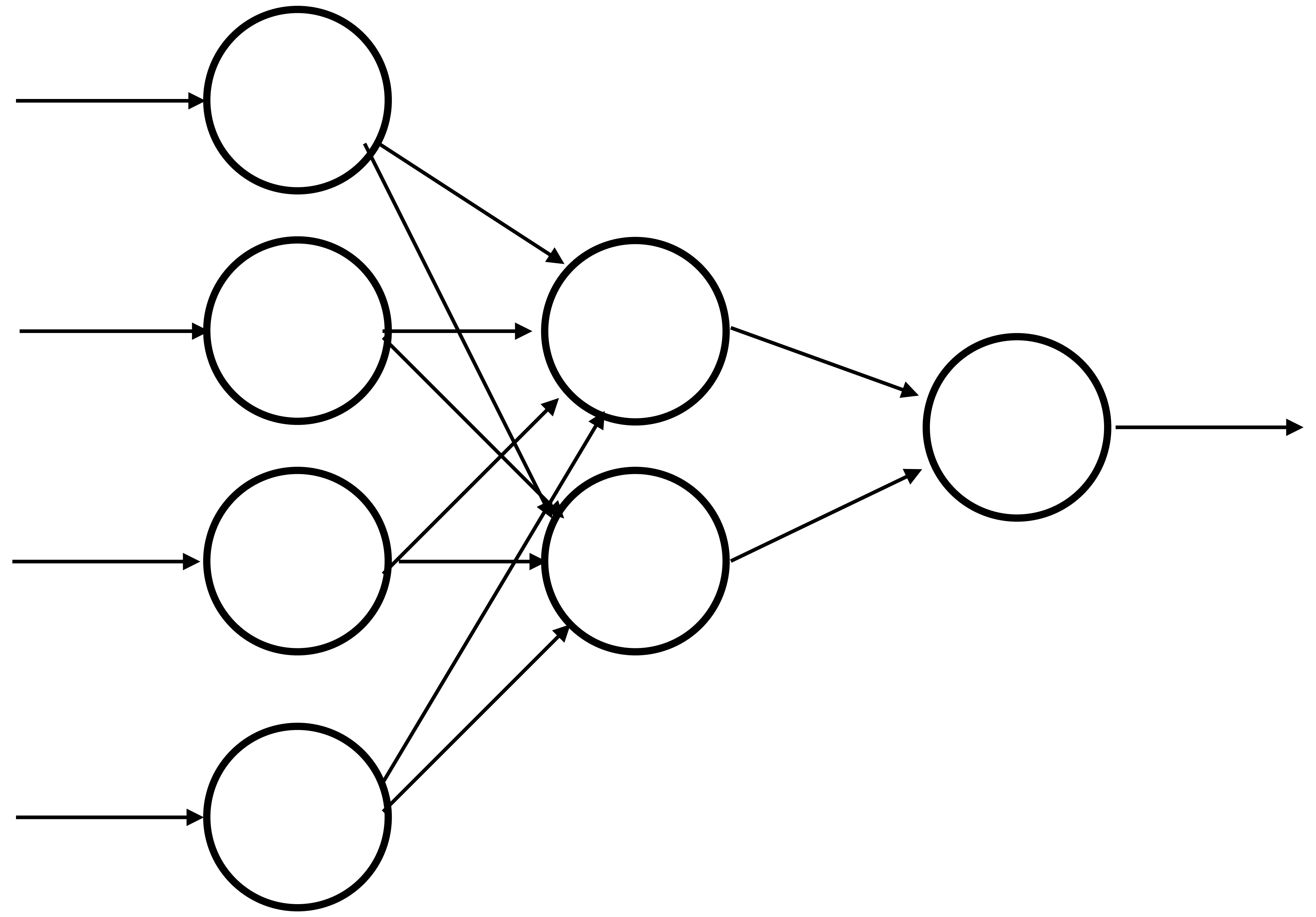
BSP



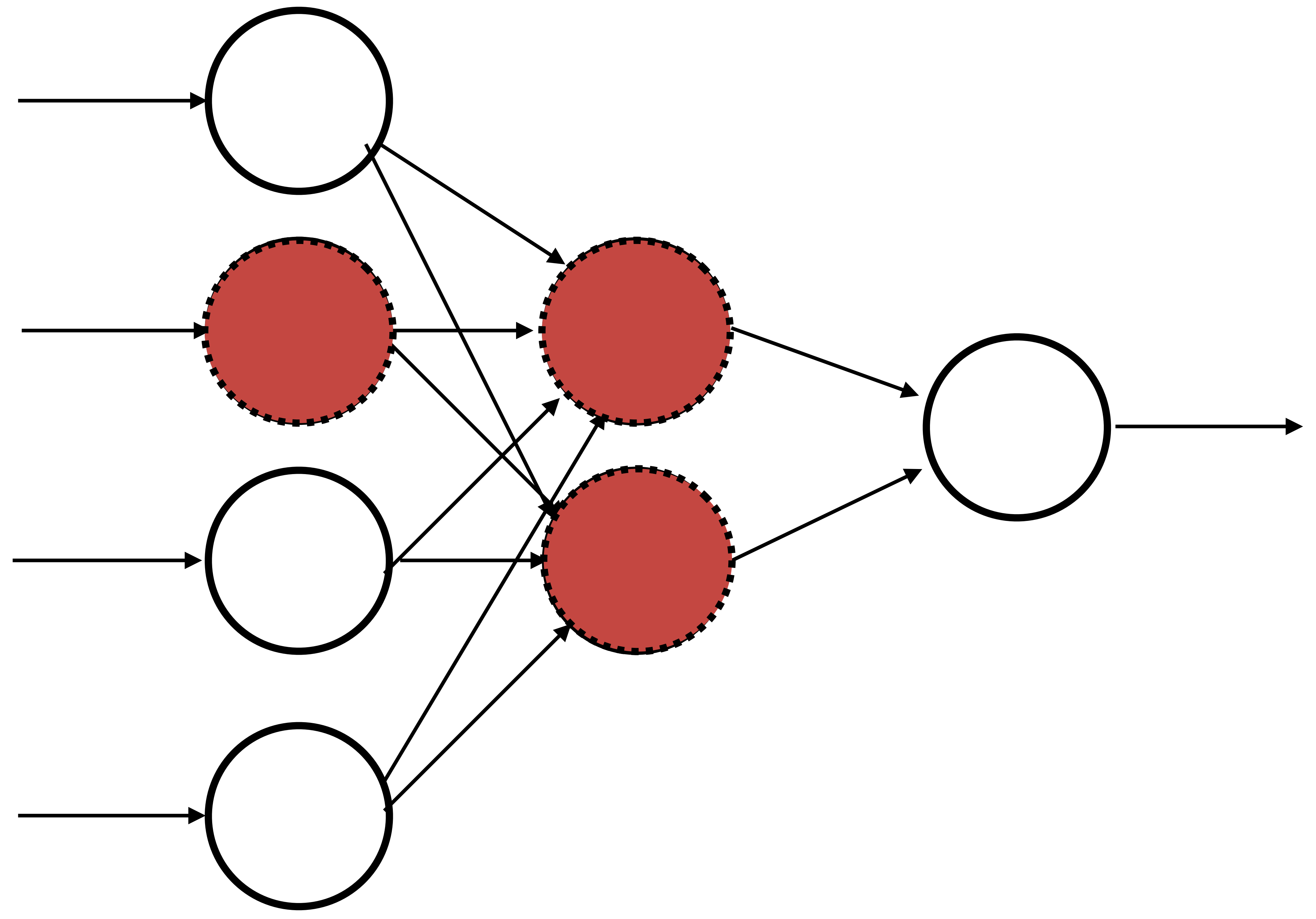
A Closer Look at Performance



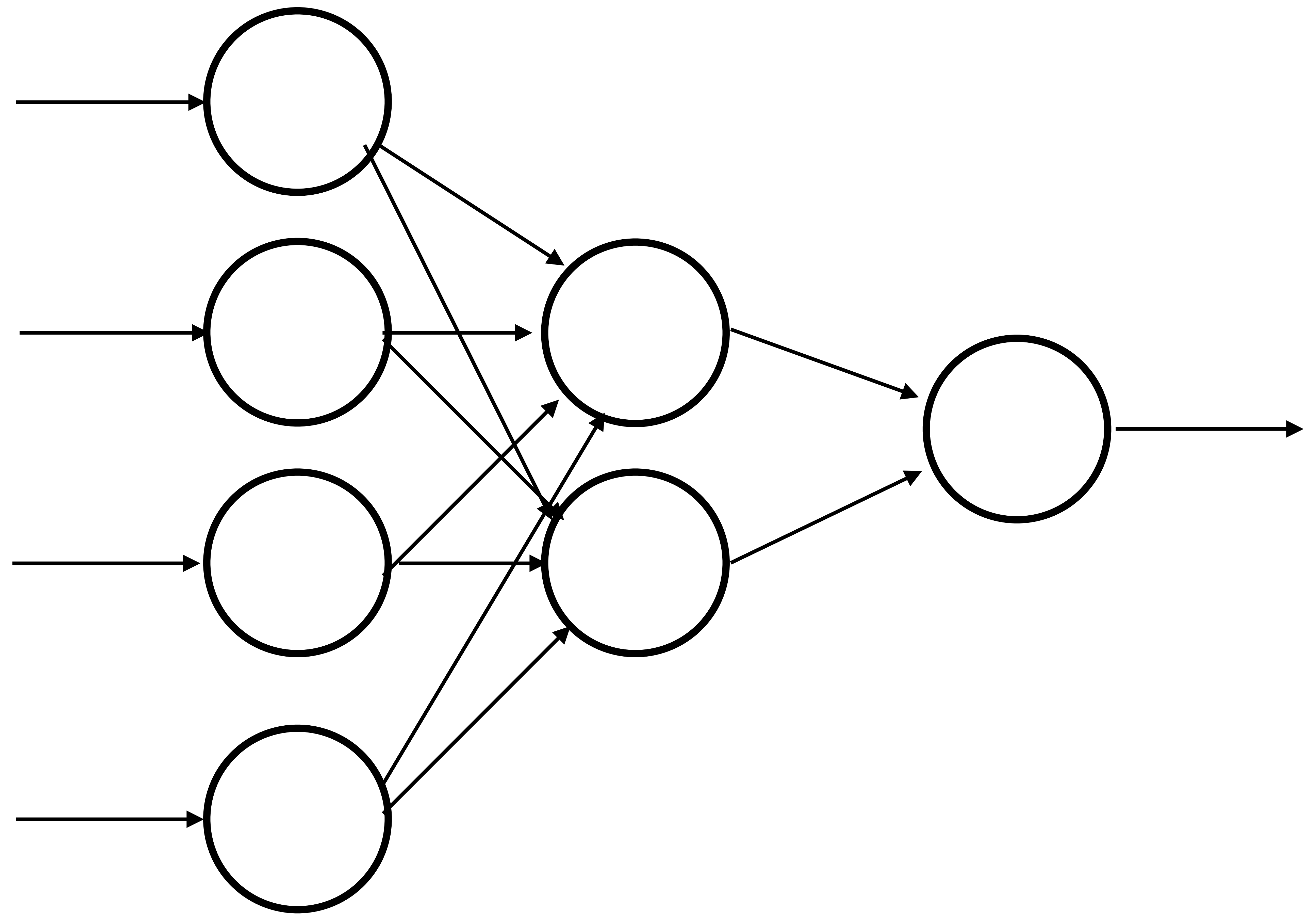
BSP



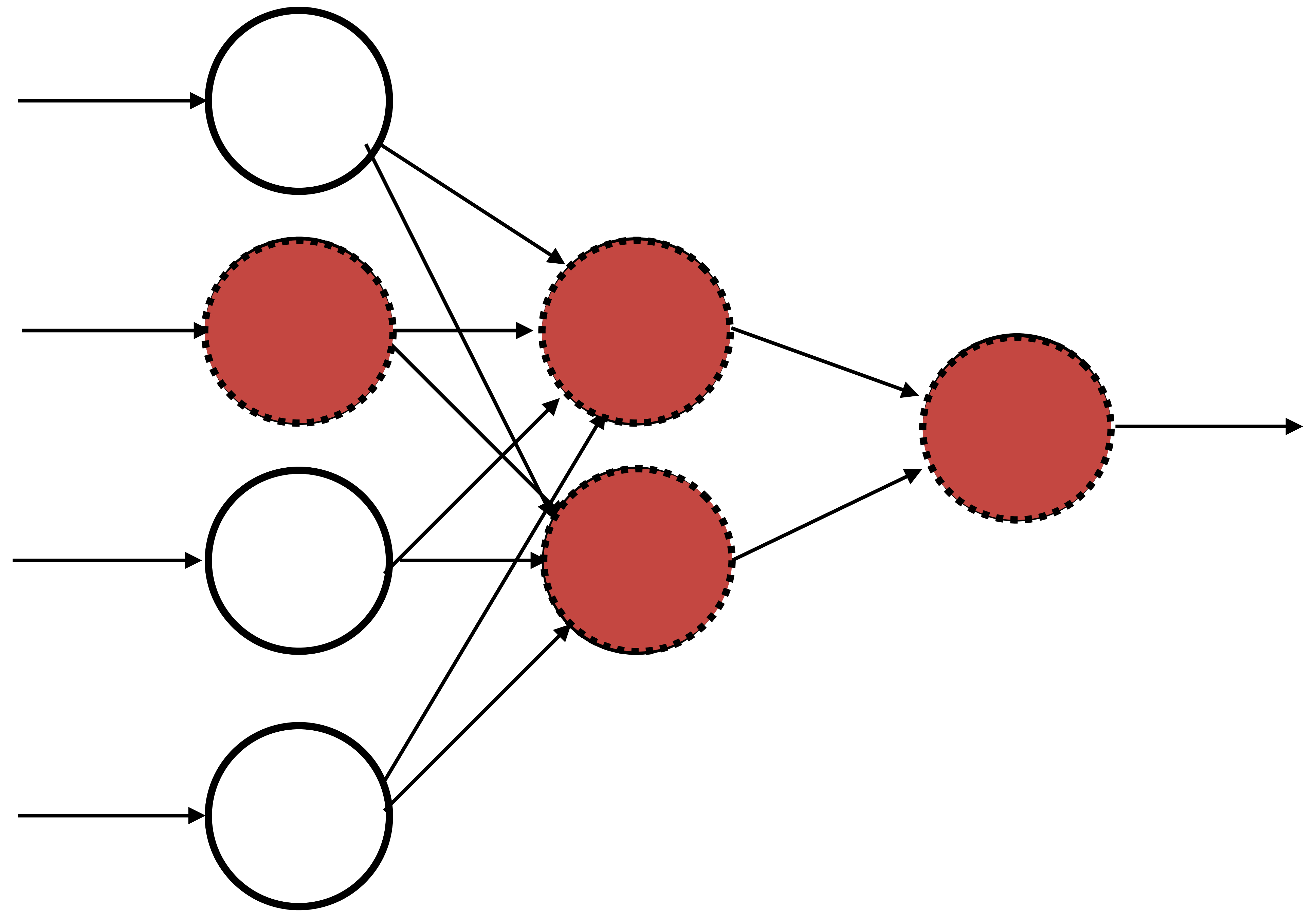
A Closer Look at Performance



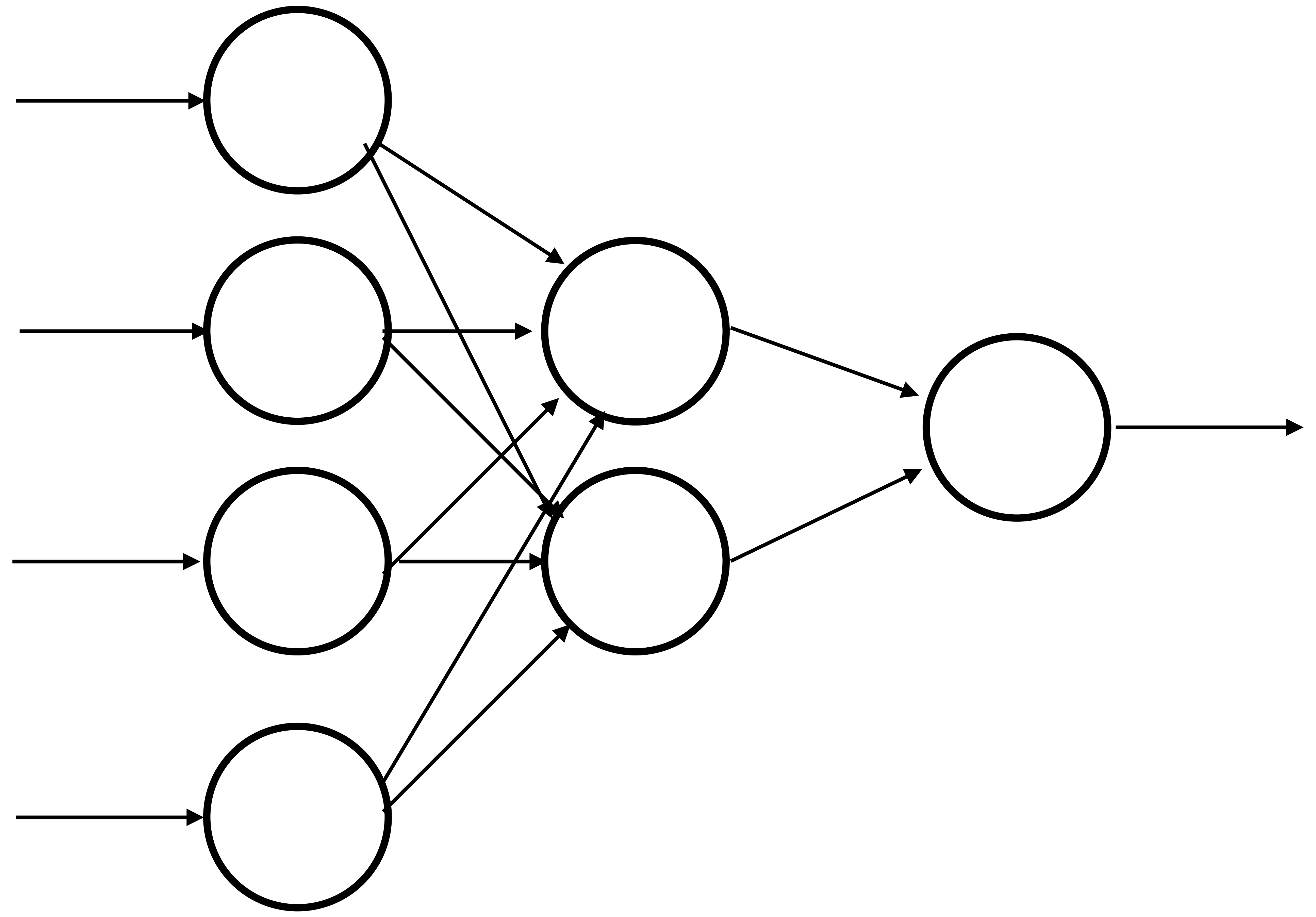
BSP



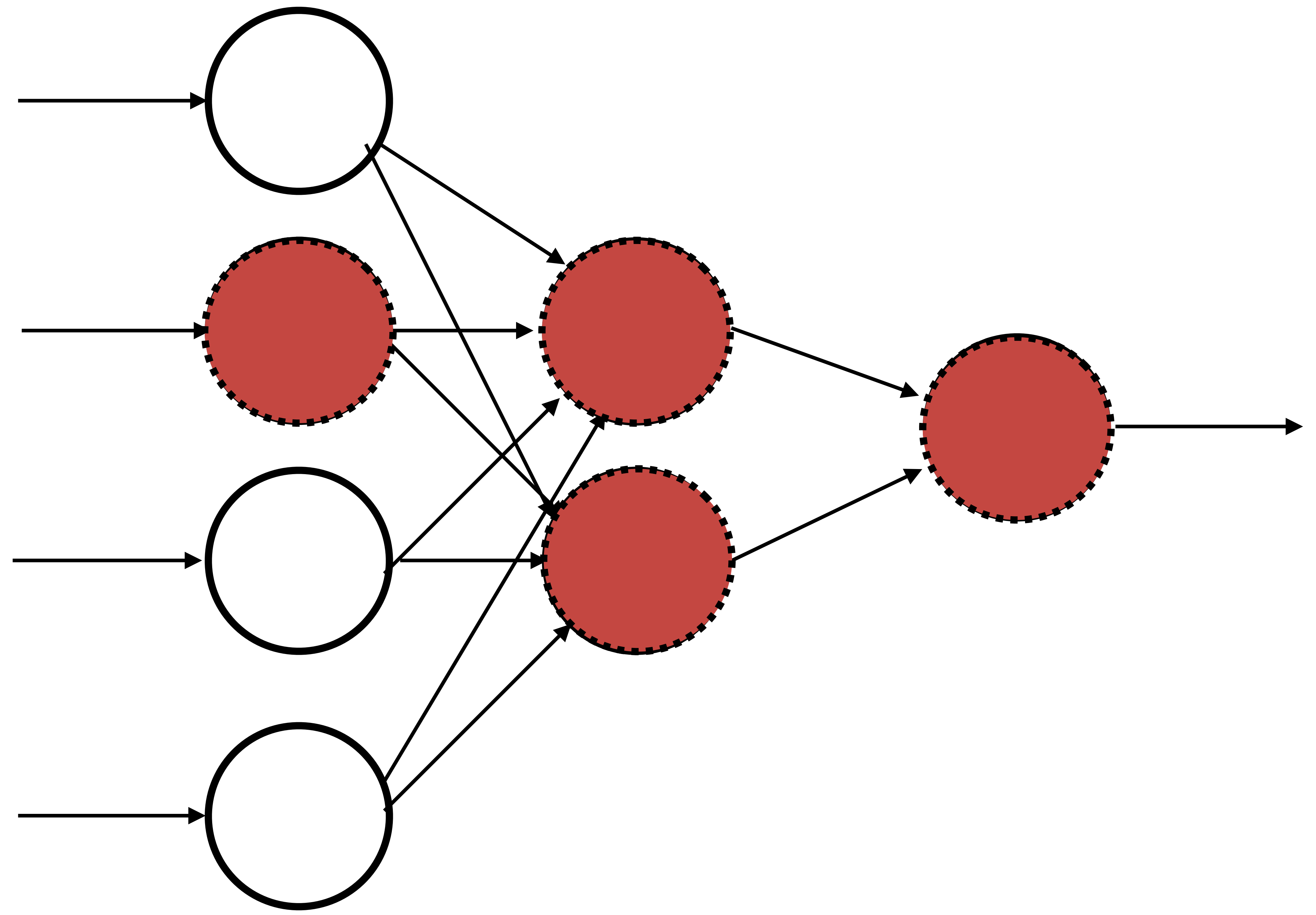
A Closer Look at Performance



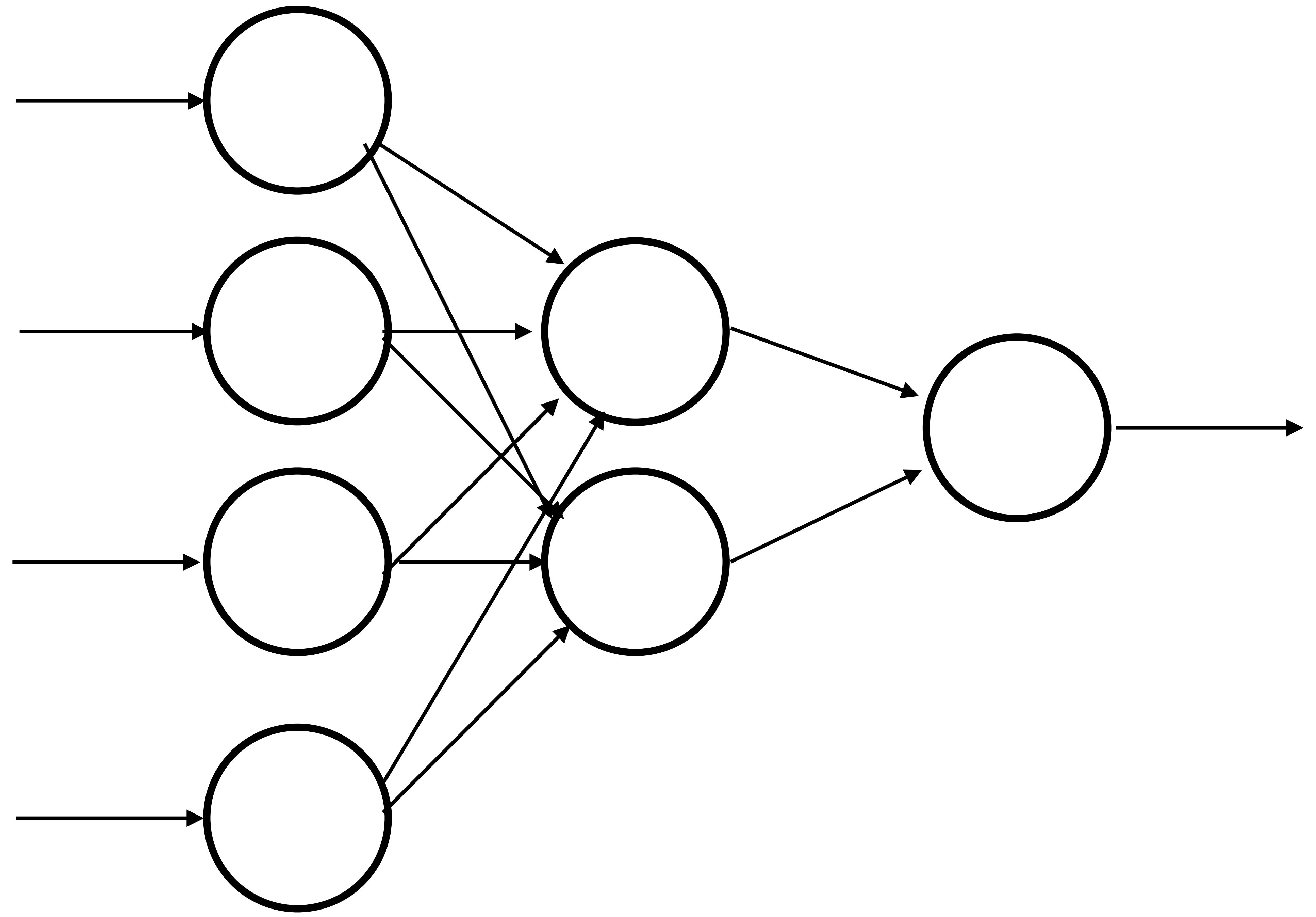
BSP



A Closer Look at Performance

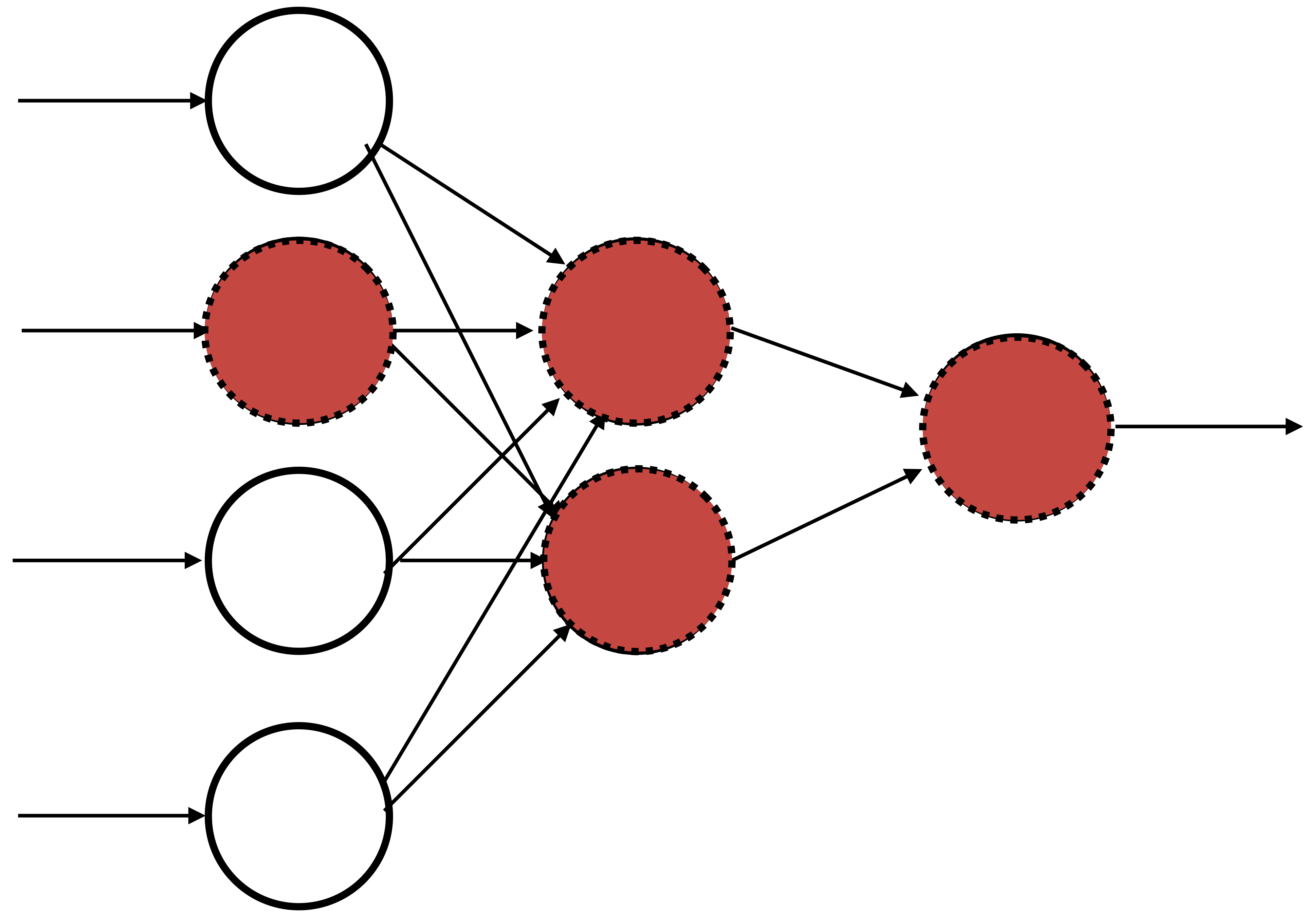


BSP

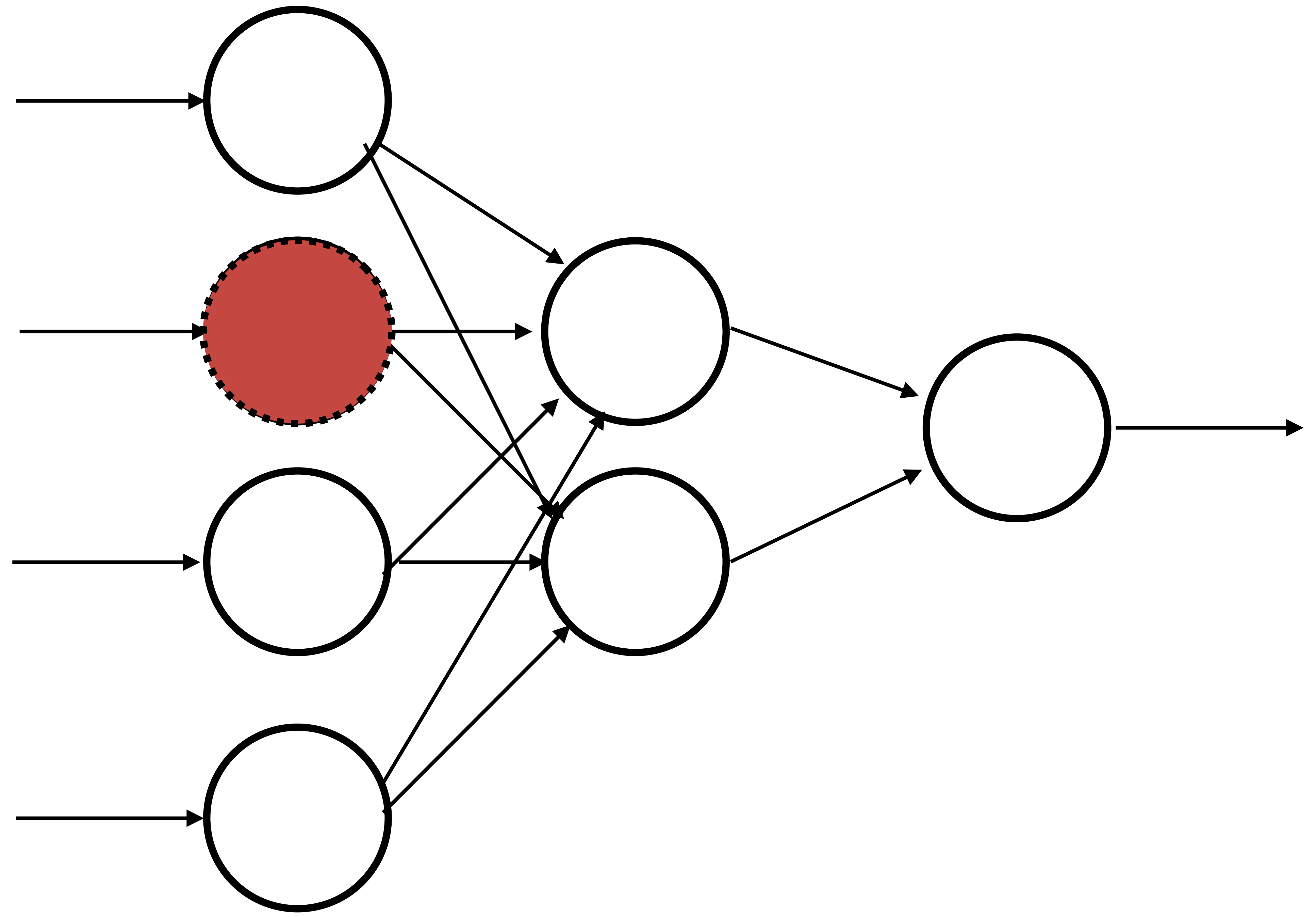


Out-of-Order Processing
(Assuming partial results are still useful)

A Closer Look at Performance

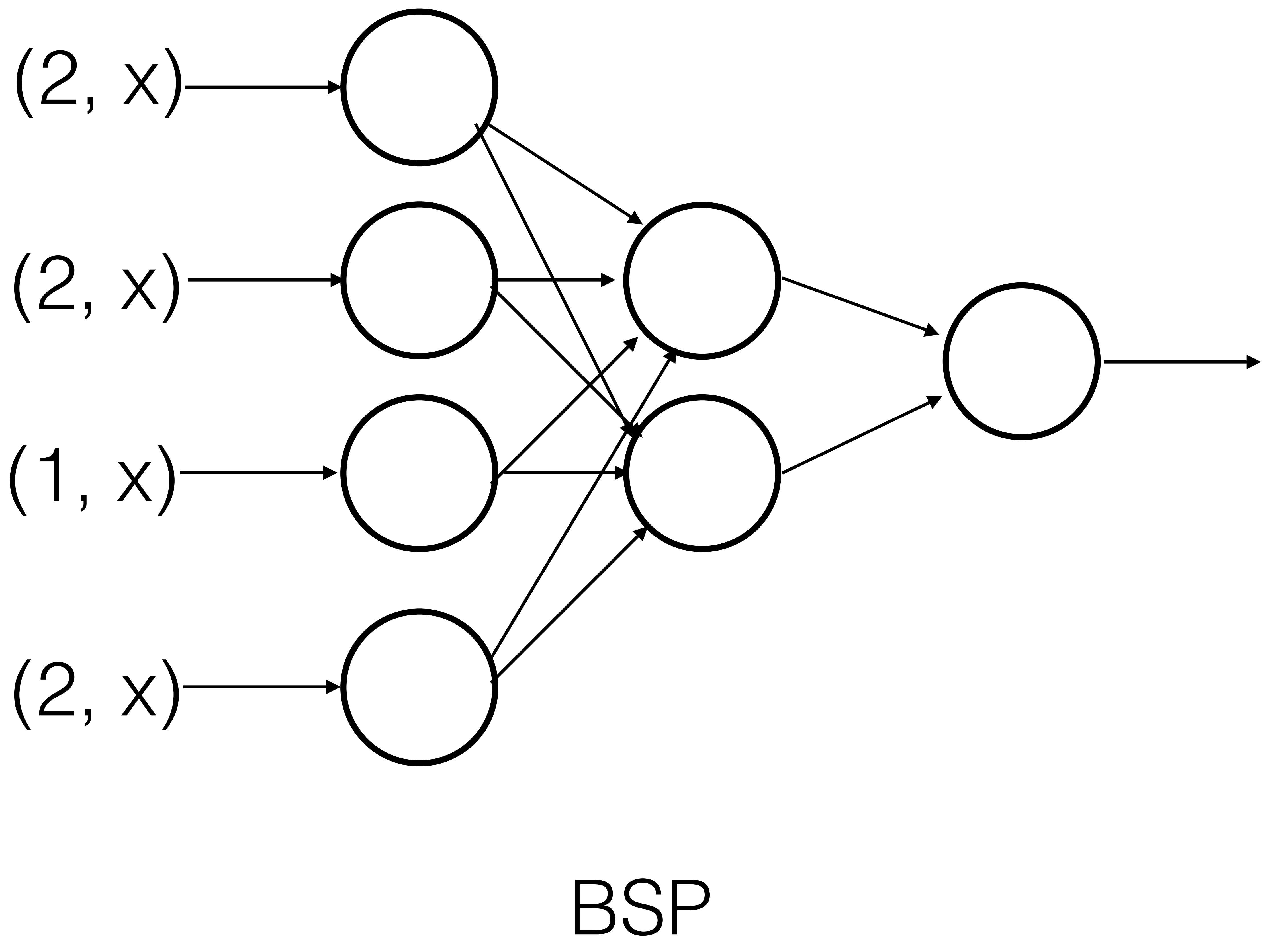


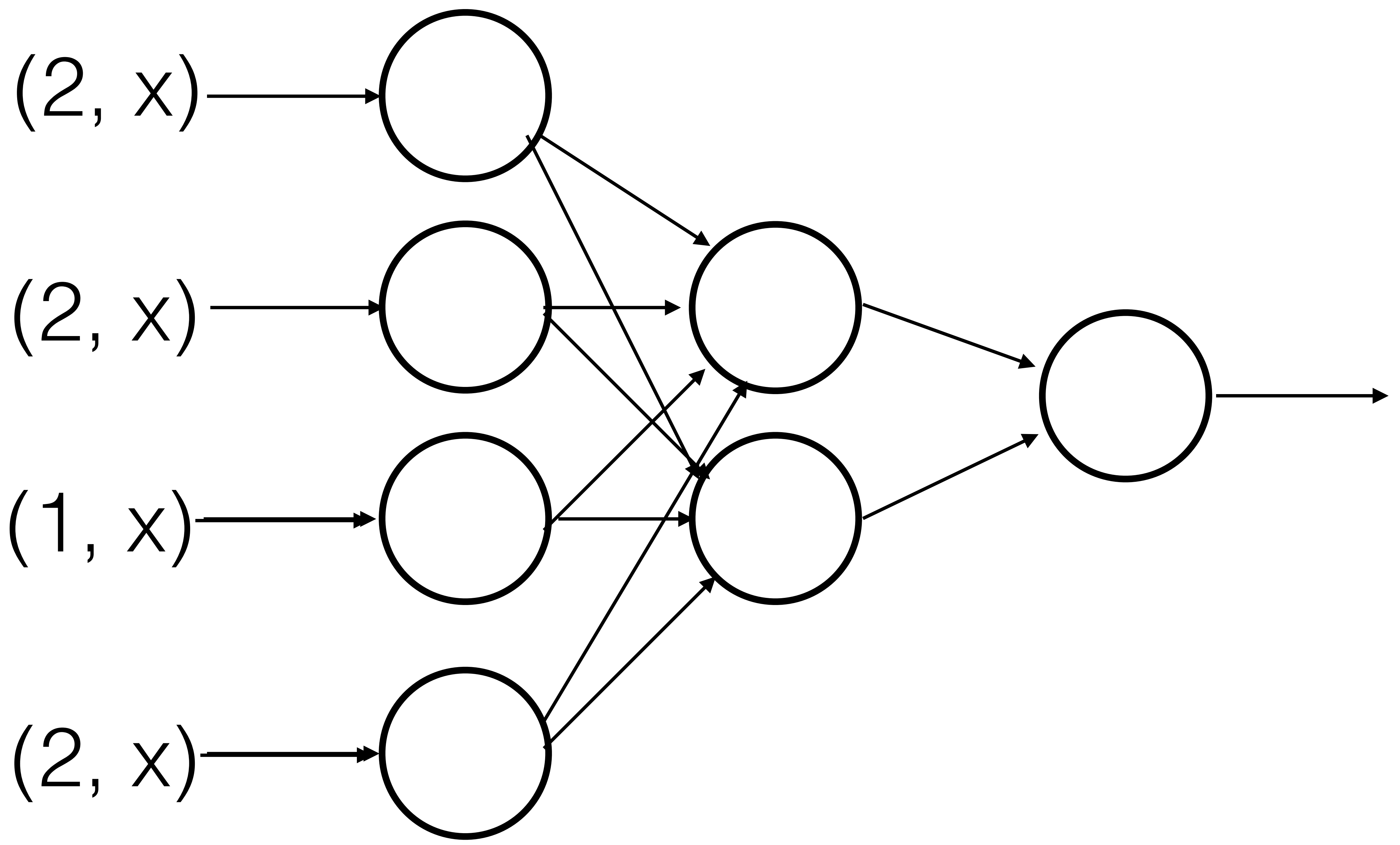
BSP



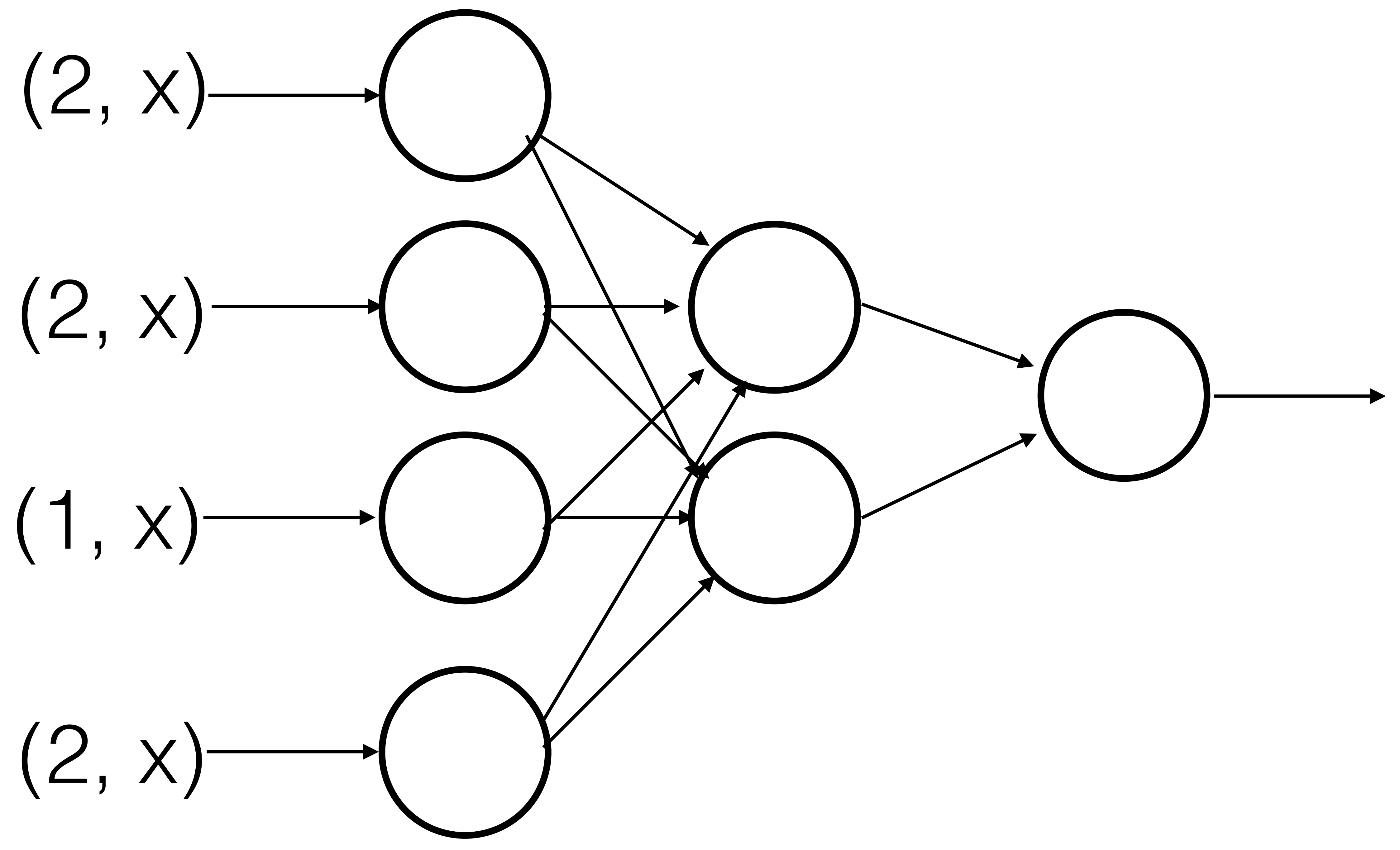
Out-of-Order Processing
(Assuming partial results are still useful)

A Closer Look at Performance

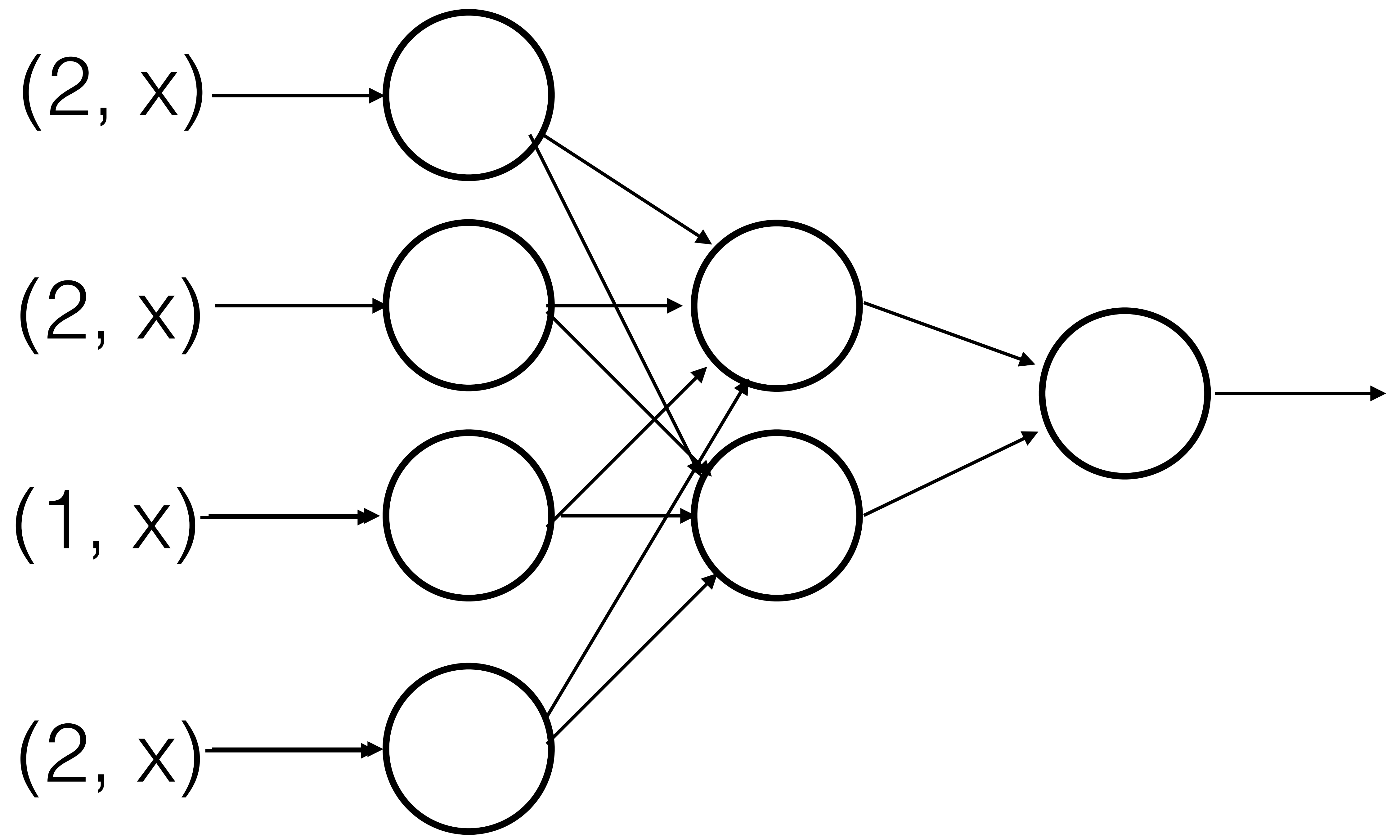




A Closer Look at Performance

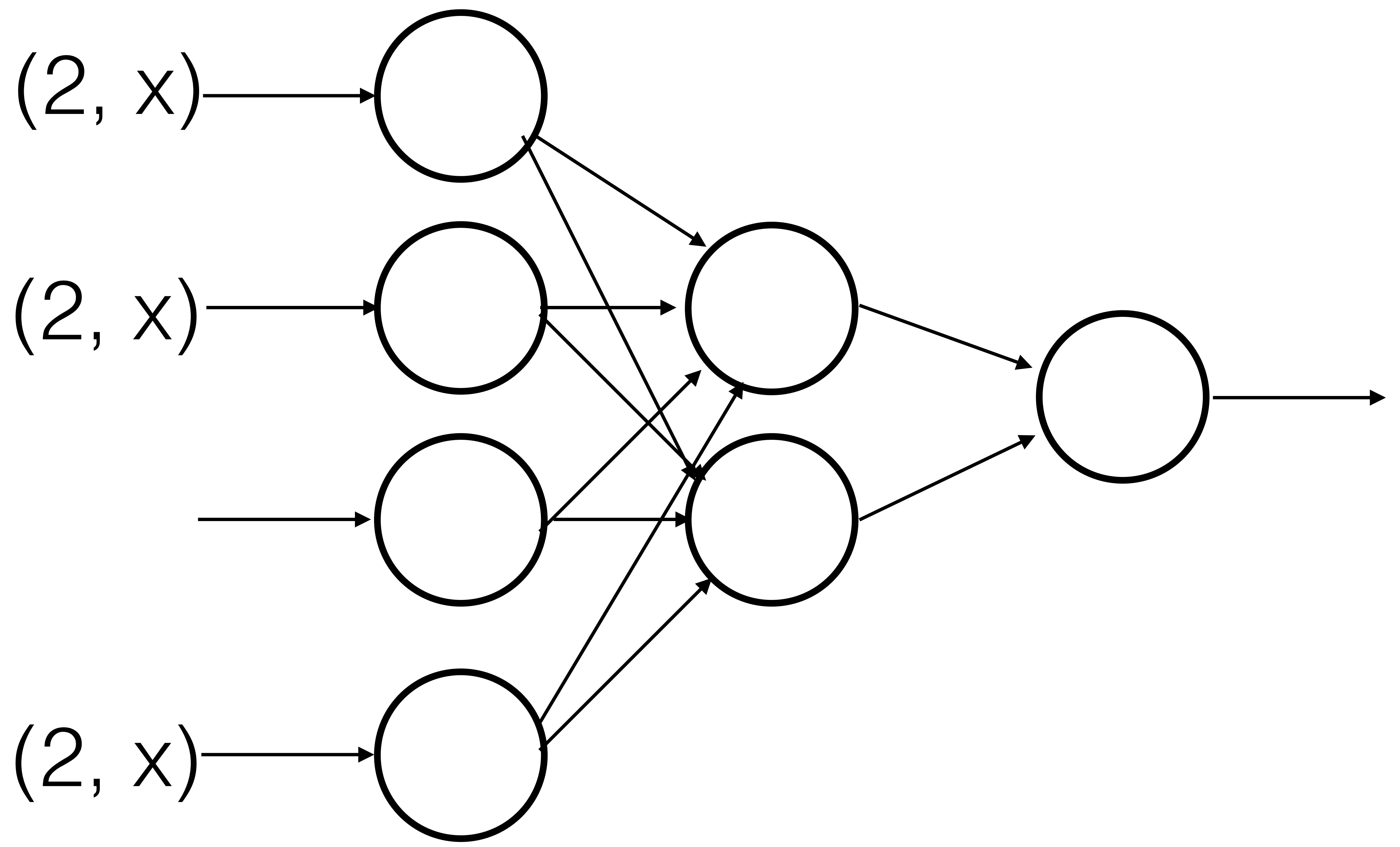


BSP

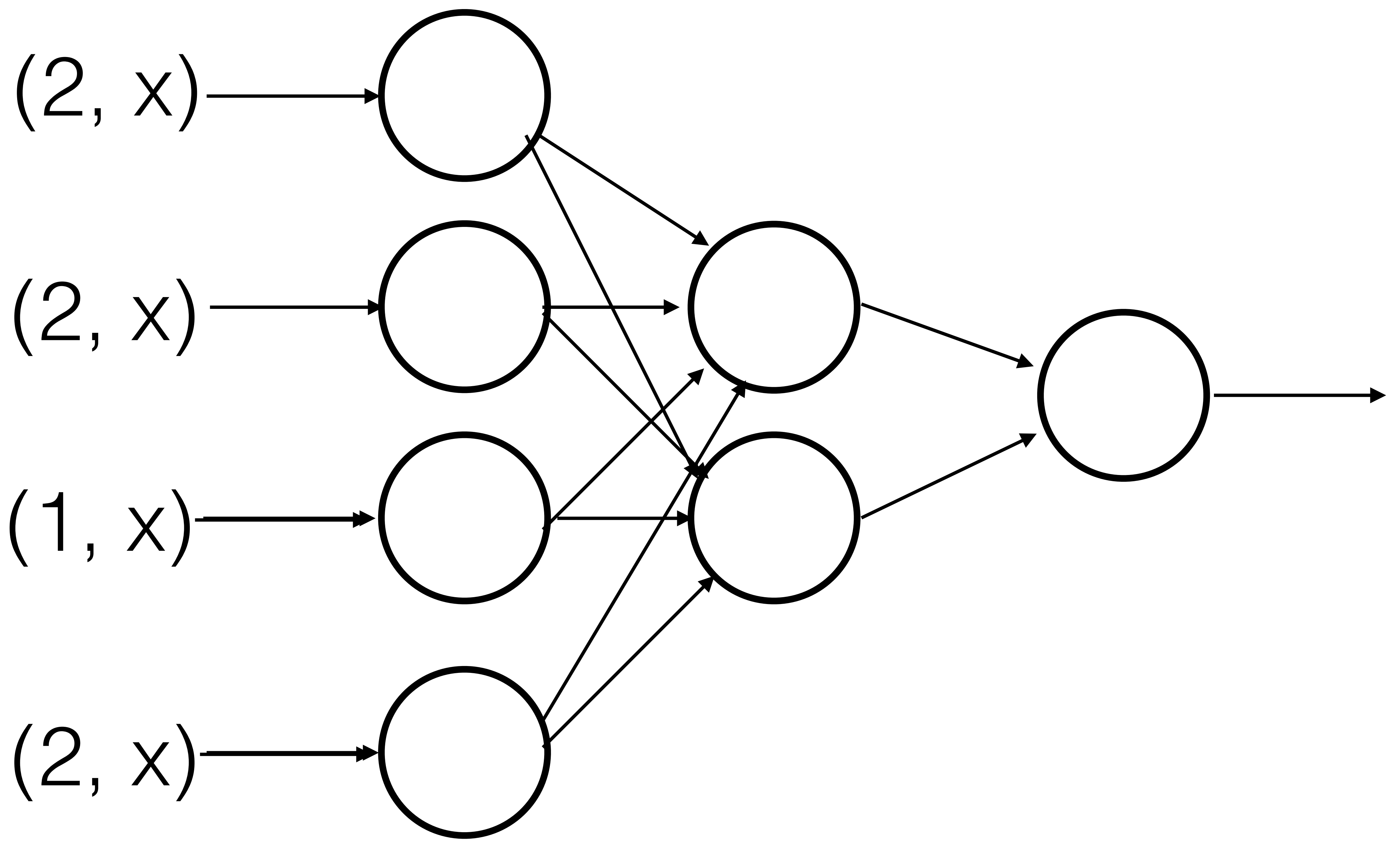


Out-of-Order Processing
(Assuming no interdependence between different epochs)

A Closer Look at Performance

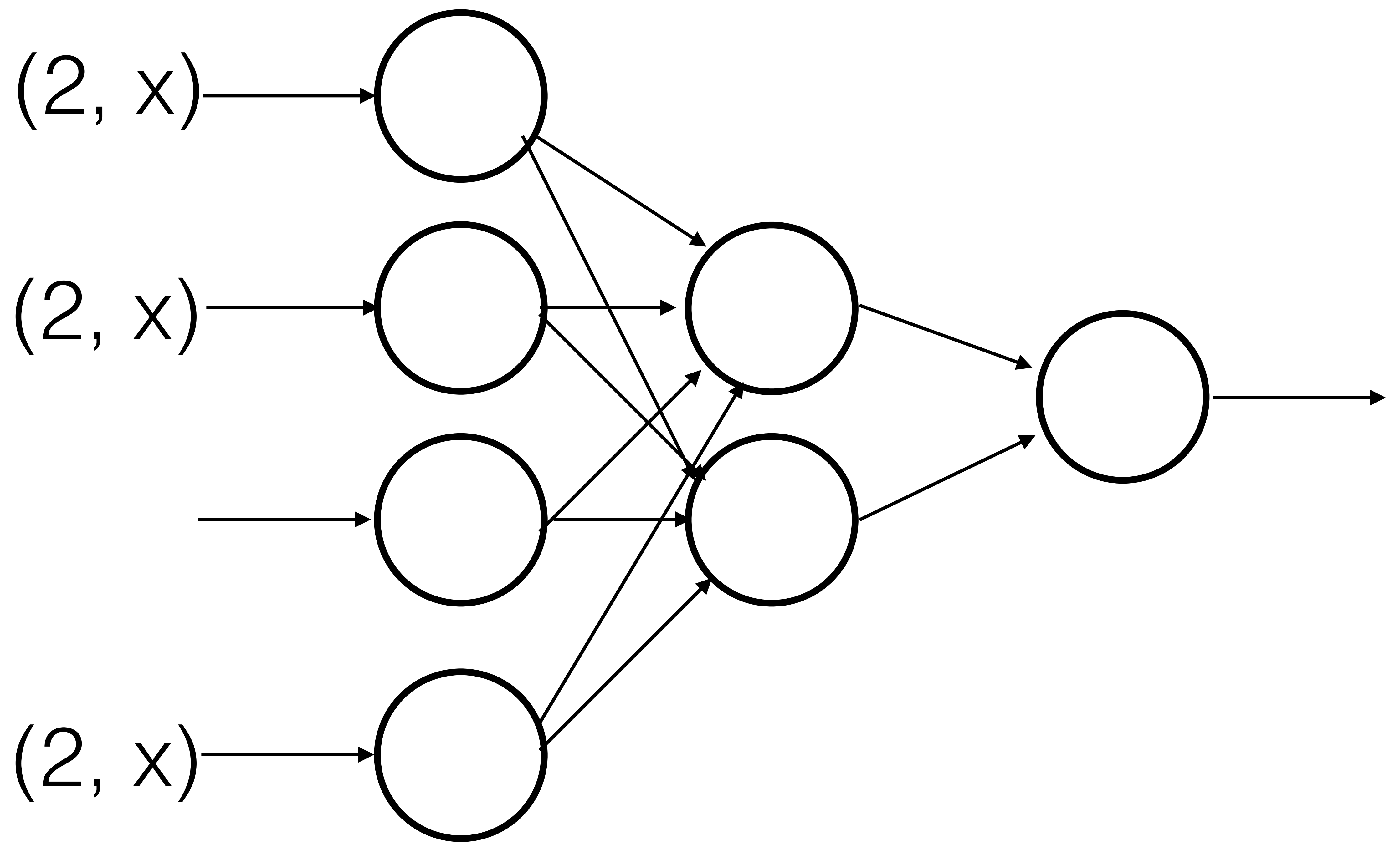


BSP

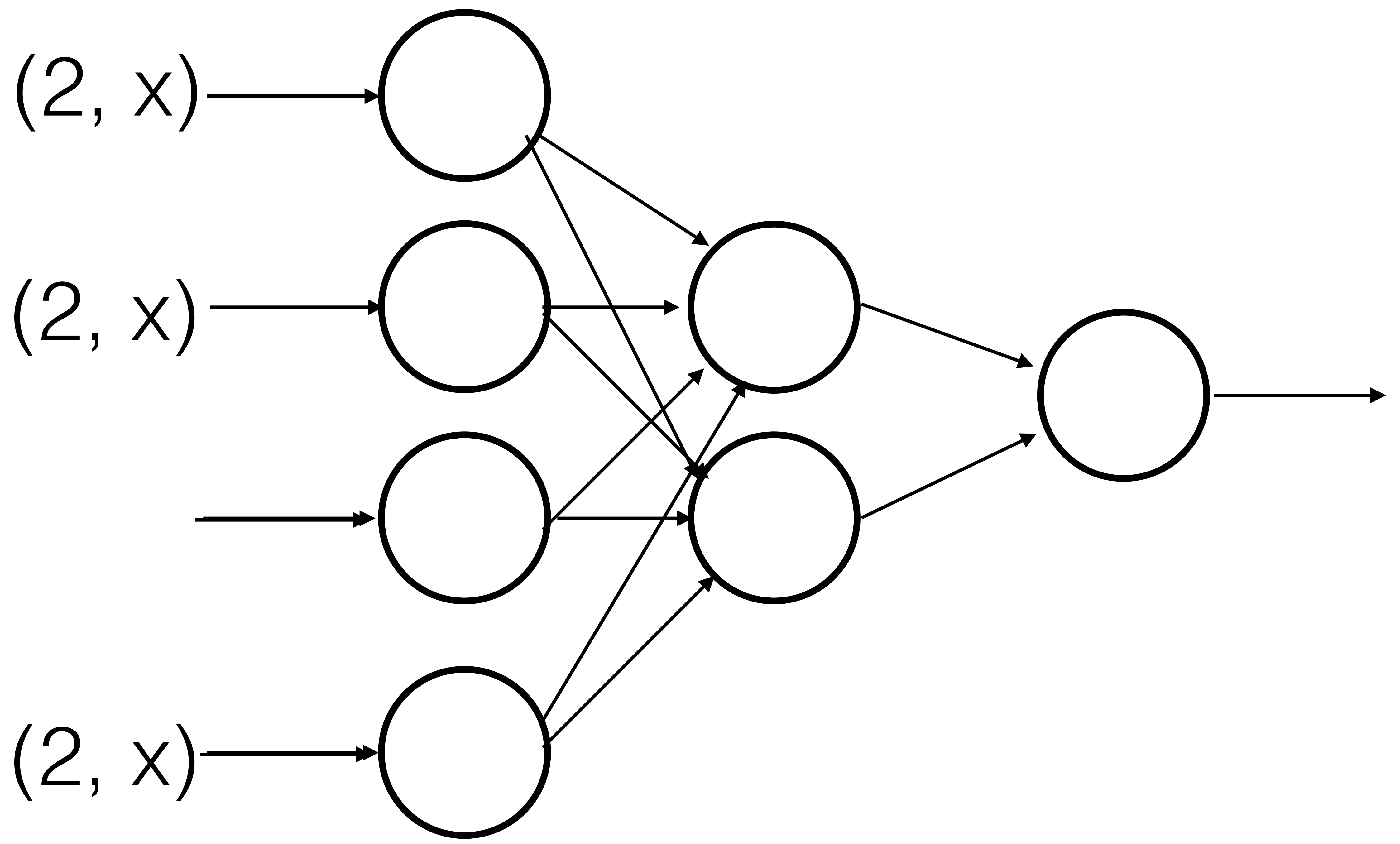


Out-of-Order Processing
(Assuming no interdependence between different epochs)

A Closer Look at Performance

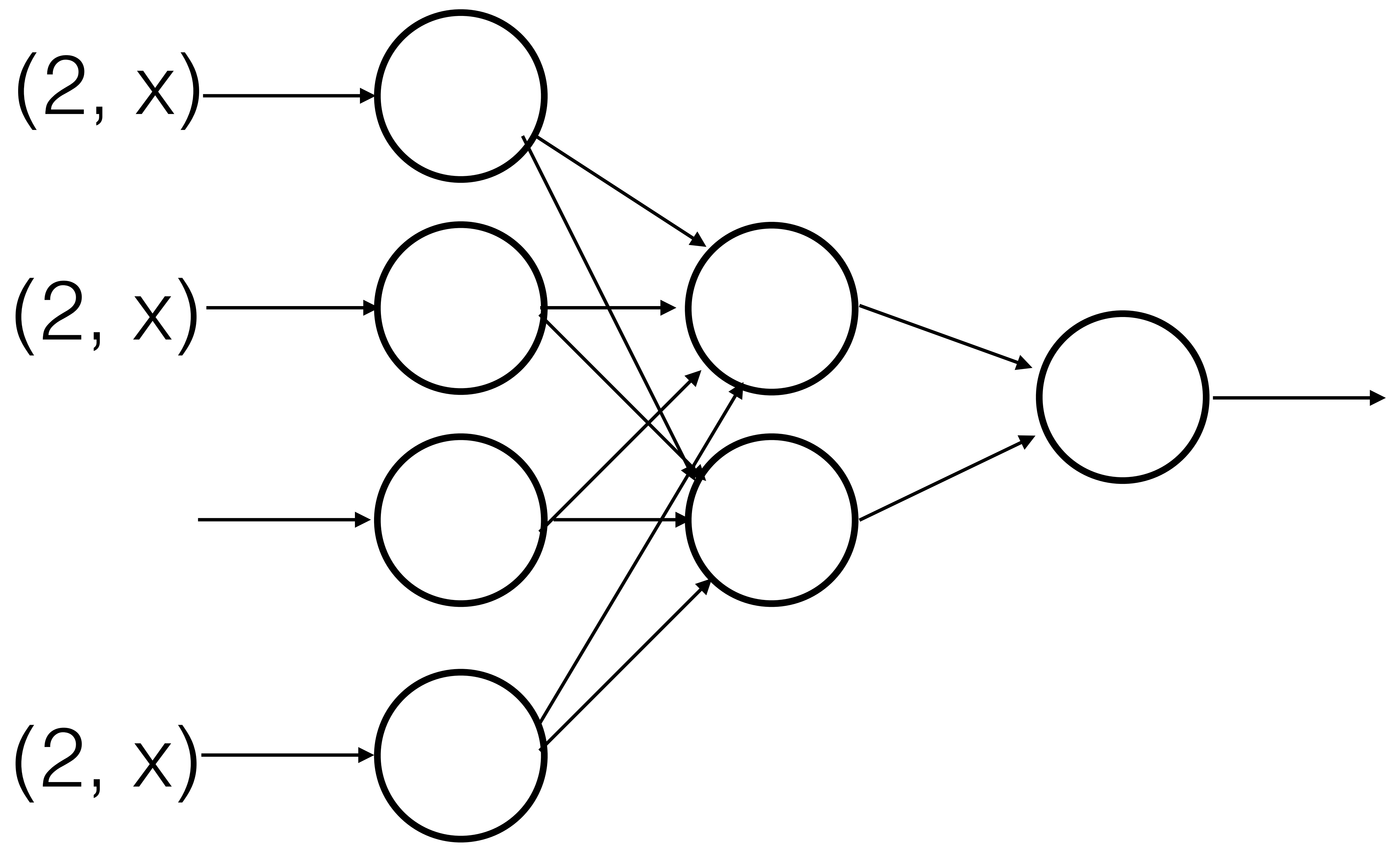


BSP

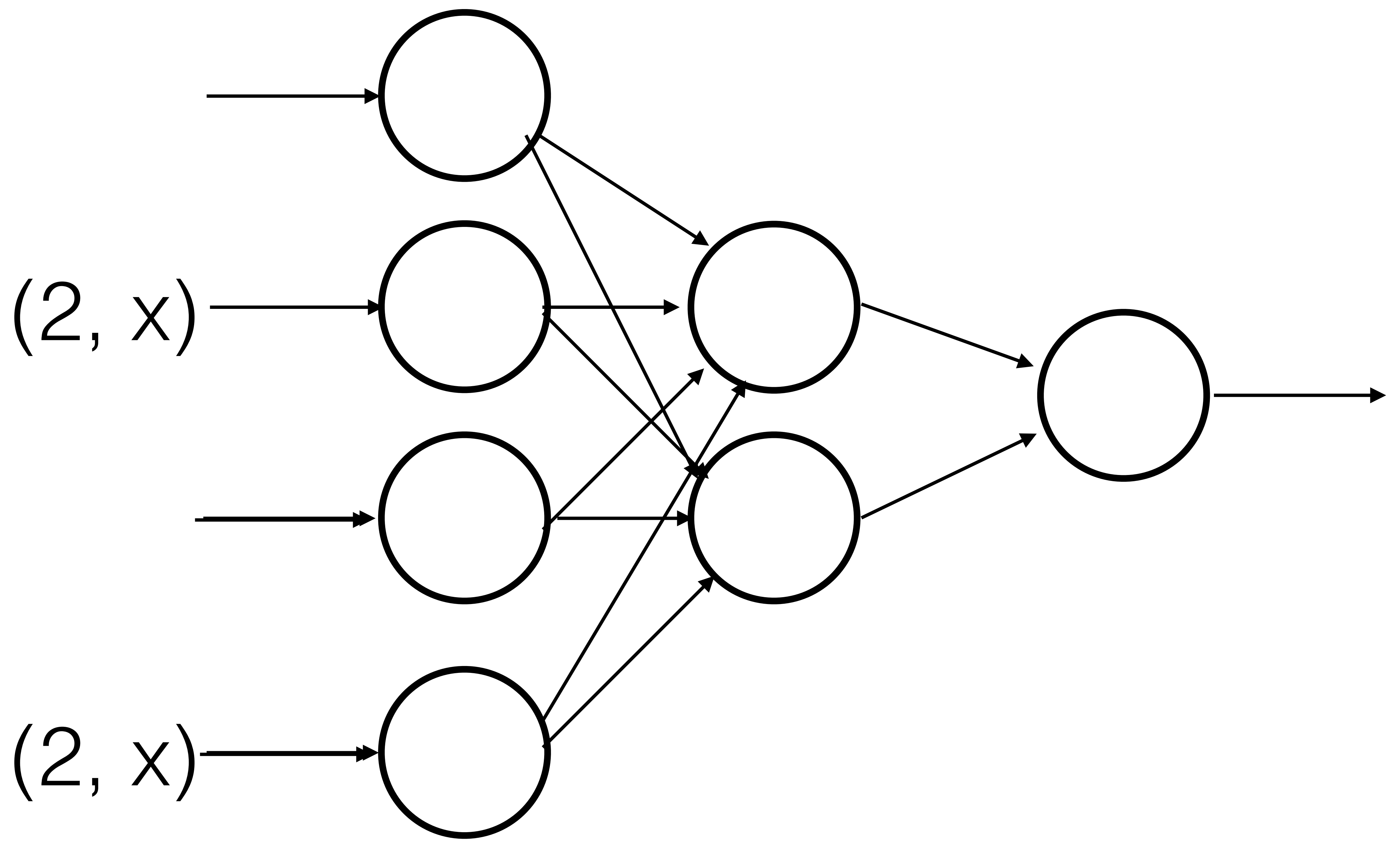


Out-of-Order Processing
(Assuming no interdependence between different epochs)

A Closer Look at Performance

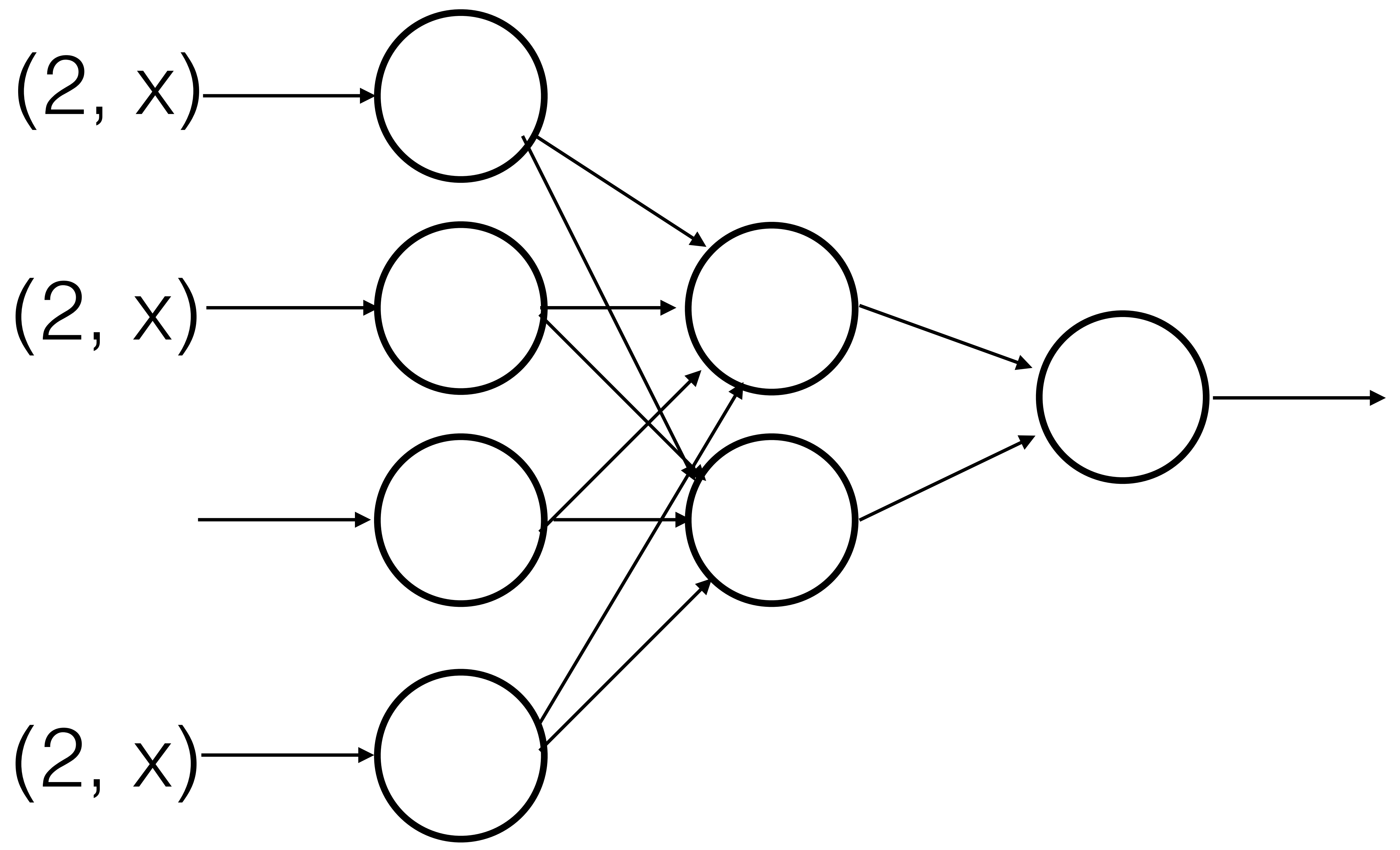


BSP

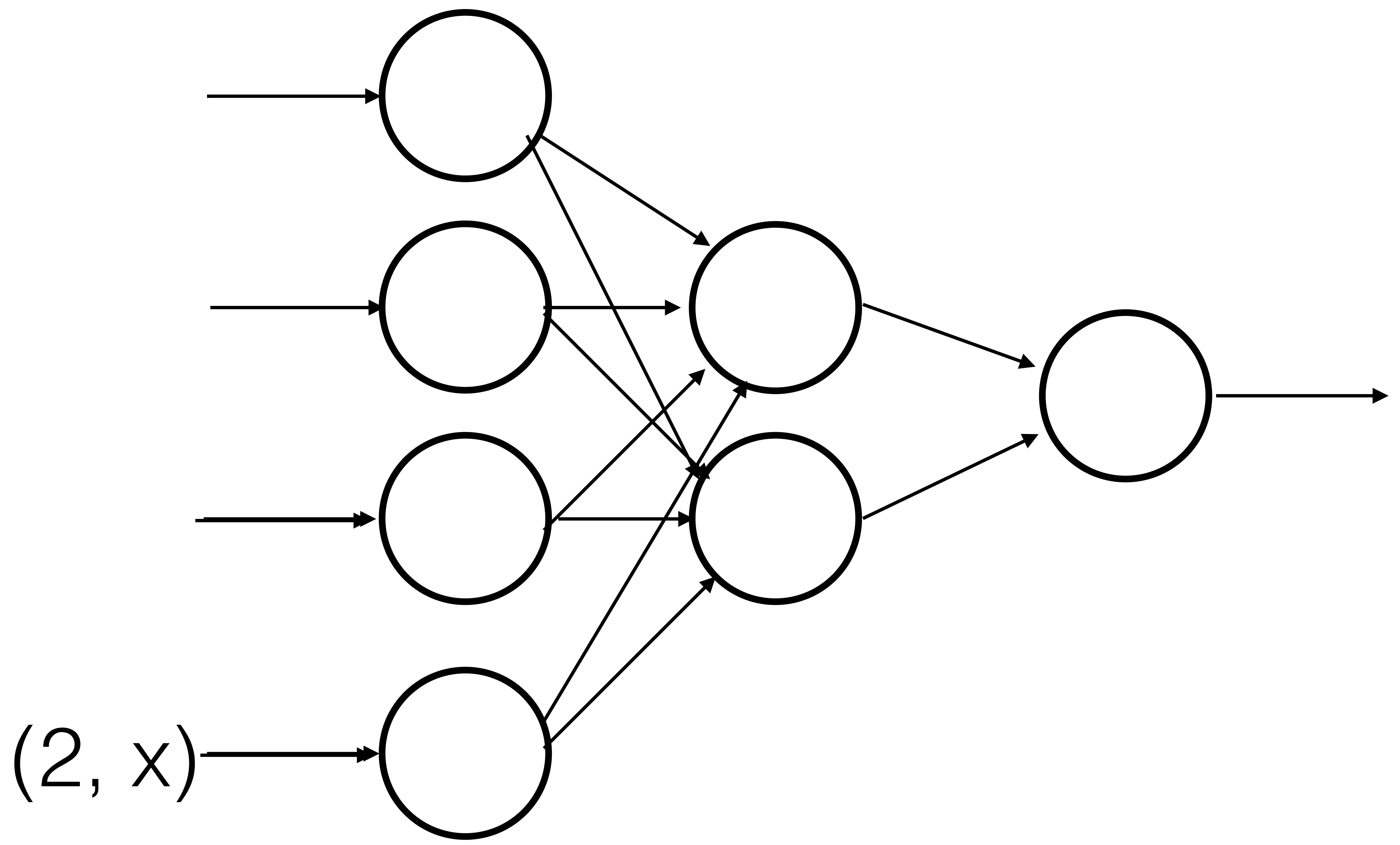


Out-of-Order Processing
(Assuming no interdependence between different epochs)

A Closer Look at Performance

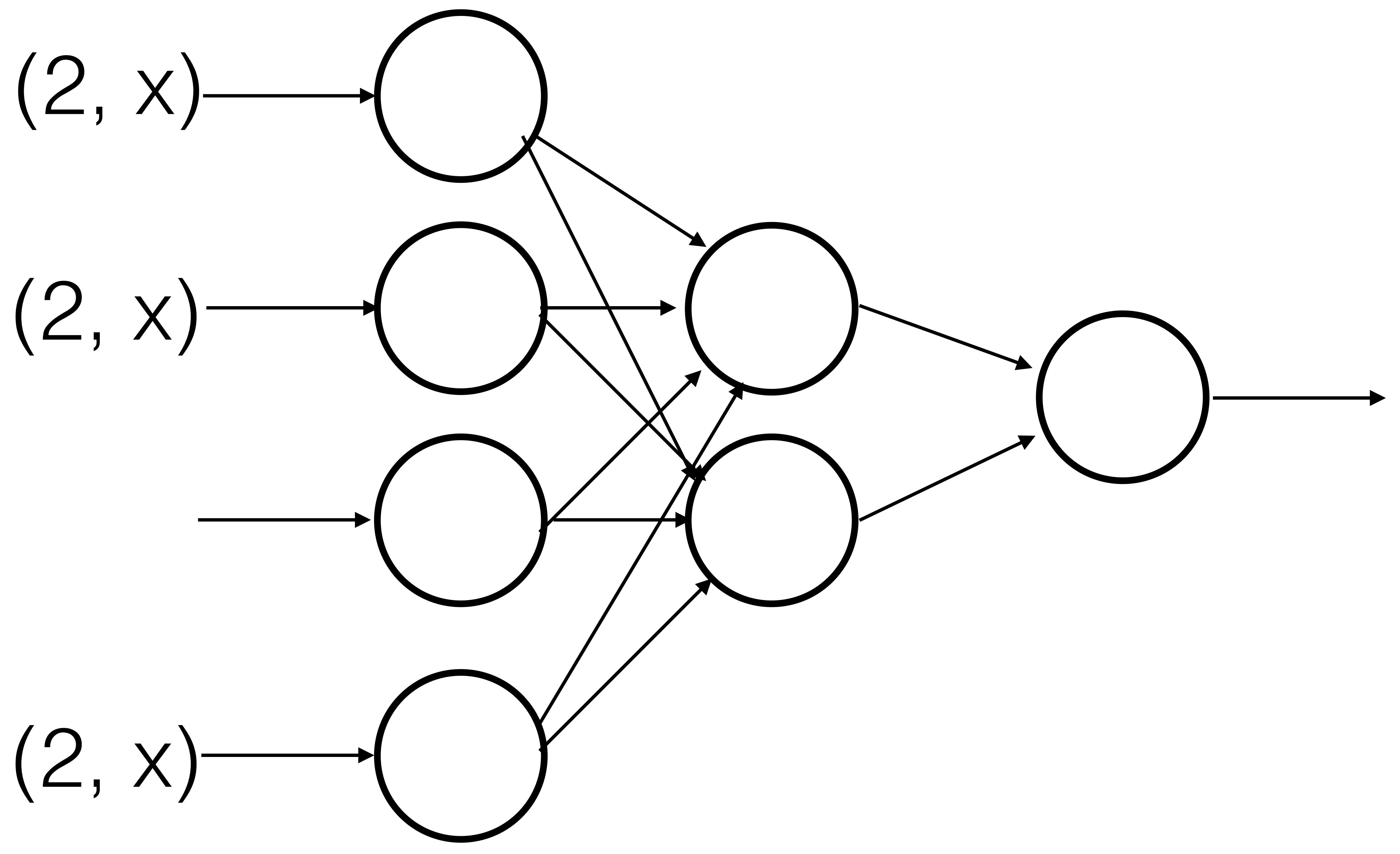


BSP

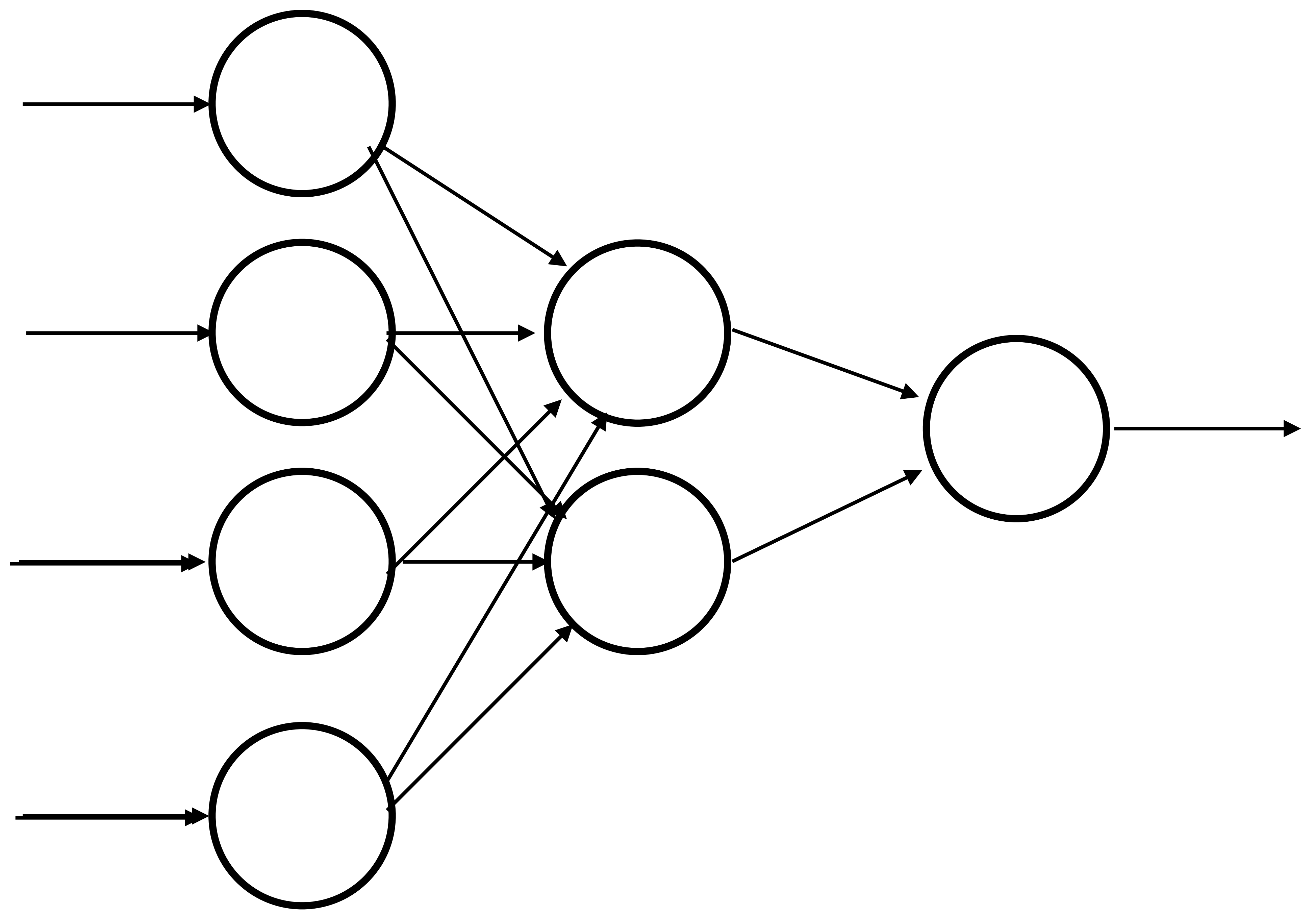


Out-of-Order Processing
(Assuming no interdependence between different epochs)

A Closer Look at Performance



BSP



Out-of-Order Processing
(Assuming no interdependence between different epochs)

Problems to Solve

- Out-of-order is great, but also need to be able to implement BSP.
  - Implemented using progress tracking/notification mechanism.
  - Assume data has timestamps, timestamps have partial order.
  - Processing data with time t results in output with t’ => t’ ≥ t
  - Really time is a logical vector, components for loop counters
- Need loop counters for synchronization in iterative computation.

Problems to Solve

- Stateful Nodes: Revisit all the old problems
    - Fault tolerance: Use checkpointing, but a major weakness.
    - Straggler Mitigation: Use better systems, existing things don’t work

Where are we now?

- Recently: Frank McSherry implemented Naiad in Rust

Where are we now?

- Recently: Frank McSherry implemented Naiad in Rust
- Lots of blog posts claiming faster performance than Spark, etc.
    - See Kay’s talk in a little bit.

Where are we now?

- Recently: Frank McSherry implemented Naiad in Rust
- Lots of blog posts claiming faster performance than Spark, etc.
  - See Kay’s talk in a little bit.
- Performance improved because: Naiad, Rust, the really unsafe serialization?

Where are we now?

- Recently: Frank McSherry implemented Naiad in Rust
- Lots of blog posts claiming faster performance than Spark, etc.
    - See Kay’s talk in a little bit.
- Performance improved because: Naiad, Rust, the really unsafe serialization?
- Can someone who is not Frank McSherry build using Naiad?
    - Will it be as efficient?

More Serious Questions

- Do we really need these multi-paradigm systems?
  - This is quiet complex.
  - Spark, since Spark Streaming and GraphX has gotten complex.
  - Worth it?
- What should BSP systems learn from Naiad?
- Should we reimplement Spark like systems on top of Naiad?
- Scheduling?