Virtual Earthquake and seismology Research Community e-science environment in Europe
Project 283543 – FP7-INFRASTRUCTURES-2011-2 – www.verce.eu – info@verce.eu

# dispel4py in detail

(dispel4py training for OSDC workshop)
session 6 - part II

9 June 2015, Amsterdam

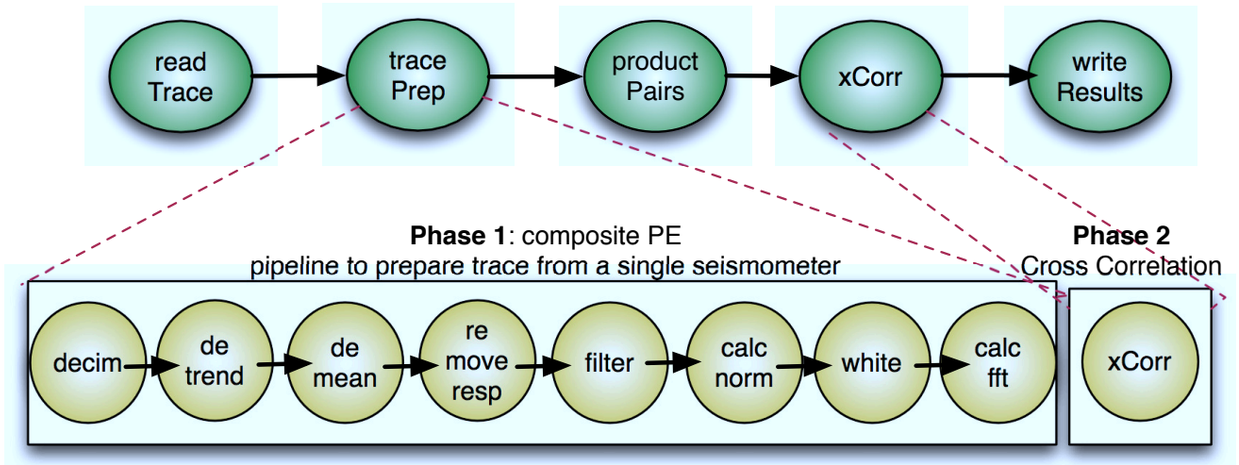
# Outline

- Installation and links
- dispel4py workflows
    - Seismology
    - Astrophysics
    - Social Computing
- Graph examples
- Composite PEs
- Chains
- Groupings
- Mappings to execution platforms

# Installation & links

- This is all you need:
  - **pip install dispel4py**
- Web site: http://dispel4py.org/
- GitHub: https://github.com/dispel4py/dispel4py
- Documentation: http://dispel4py.org/documentation/
- **Exercises**: http://effort.is.ed.ac.uk/osdc2015/osdc2015_dispel4py_exercises.tar

# dispel4py workflows- Seismology

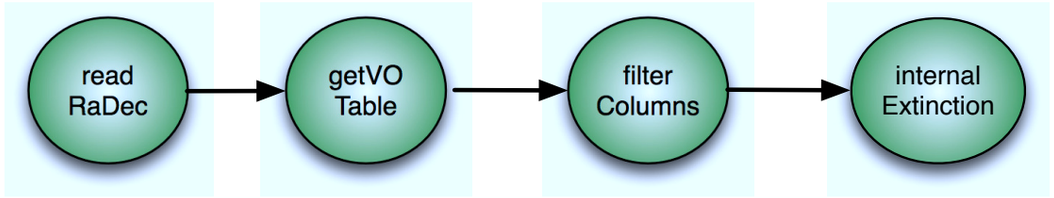



- **Phase 1- Preprocess**: Time series data (traces) from seismic stations are preprocessed in parallel
- **Phase 2: Cross-Correlation**: Pairs all of the stations and calculates the cross-correlation for each pair (complexity O(n2)).
- **Input data**: 1000 stations as input data  (150 MB)
- **Output data**: 499,500 cross-correlations (39GB)

11th IEEE eScience 2015 paper

**dispel4py**: An Agile Framework for Data-Intensive eScience

# dispel4py workflows- Astrophysics

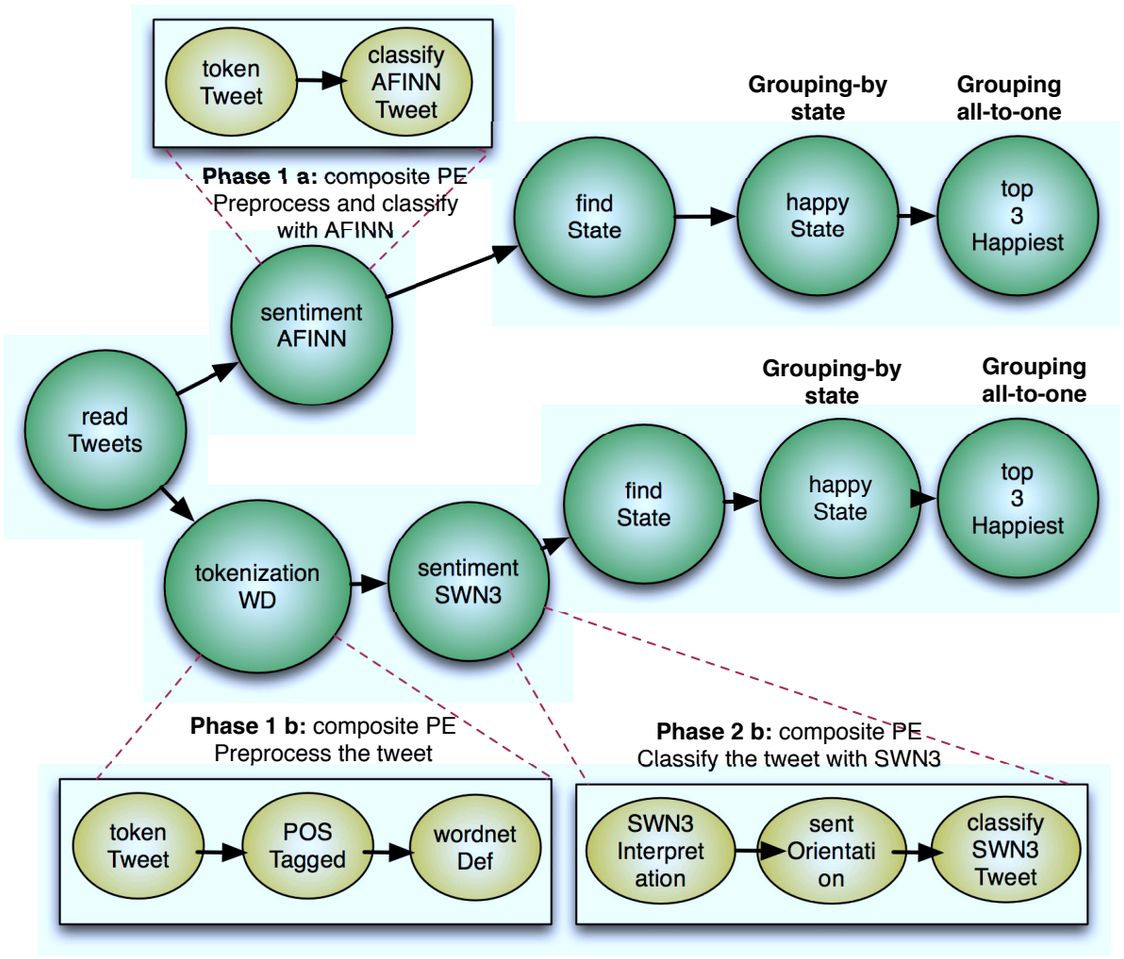



- Calculates the *Internal Extinction of the Galaxies* from the AMIGA catalogue[1]
- This property represents the dust extinction within the galaxies and is a correction coefficient needed to calculate the optical luminosity of a galaxy.
- The first PE **reads an input file** (*coordinates.txt* size 19KB) that contains the **right ascension (Ra) and declination** (Dec) values for **1051 galaxies.**
- The second PE **queries** a **VO service for each galaxy** in the input file using the Ra and Dec values.
- The results of these queries are filtered by filterColumns PE. It selects:
  - morphological type (Mtype)
  - apparent flattening (logr25)
- Their internal extinction is calculated by the internalExtinction PE.

11th IEEE eScience 2015 paper

**dispel4py**: An Agile Framework for Data-Intensive eScience

# dispel4py workflows- Social Computing

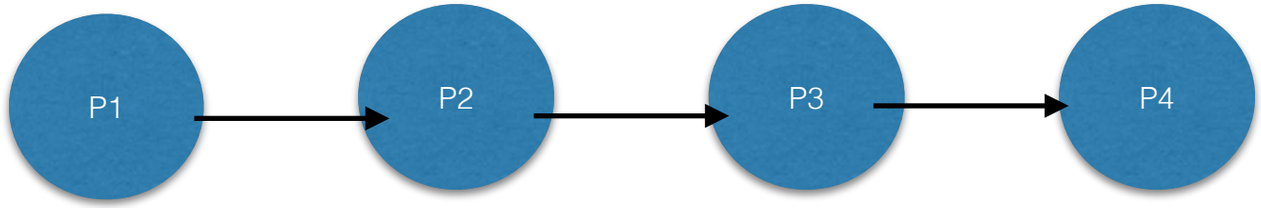



- Two basic sentiment analyses by applying two lexicons:
  - **AFINN** (2477 English words)
  - **SentiWordNet** (155,287 English words and 117,659 synsets)
- After tweets findState PE, which searches the US state from which the tweet originated
- The HappyState PE applies a grouping by based on the state and aggregates the sentiment scores of tweets from the same state
- The last PE applies all-to-one grouping and determines which are the top three happiest states.
- 126,826 tweets (500MB)

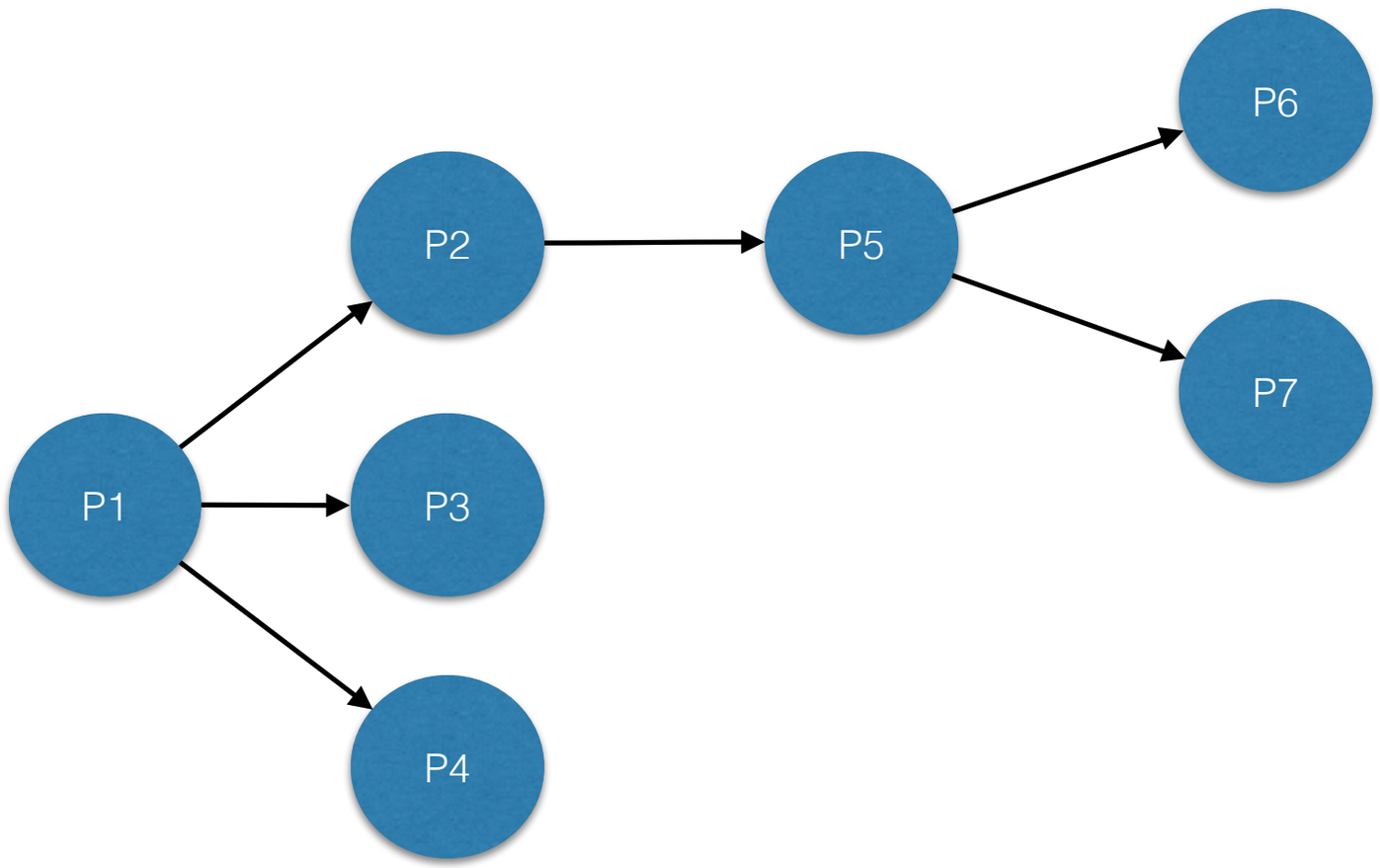
## 11th IEEE eScience 2015 paper

**dispel4py**: An Agile Framework for Data-Intensive eScience

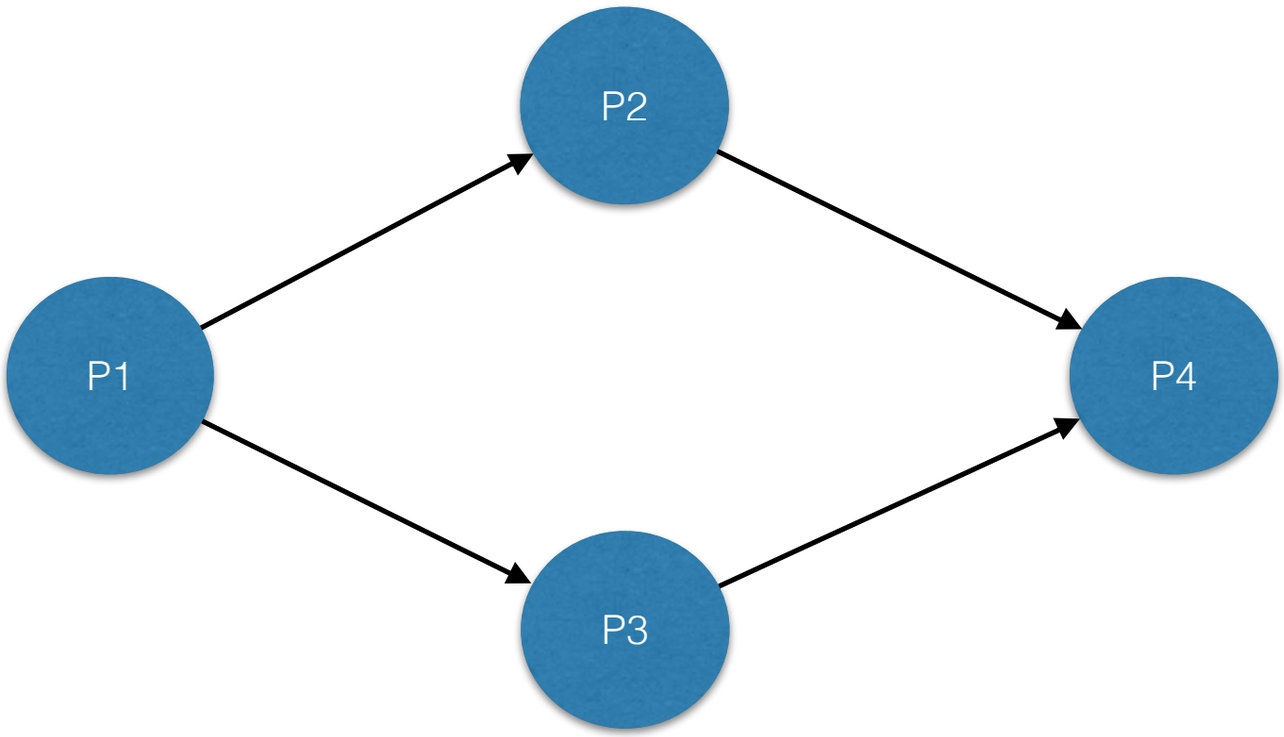
# Pipeline



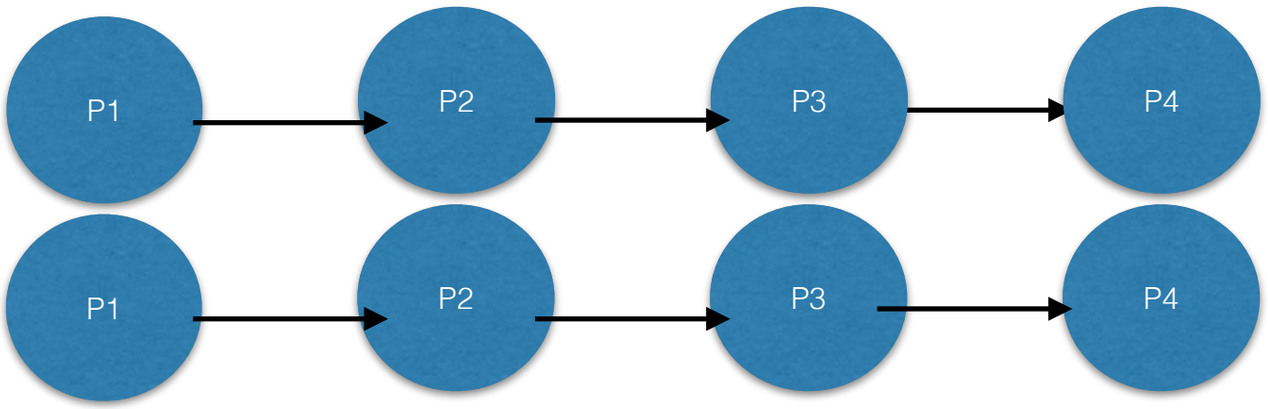
# Tree



# Split and Merge

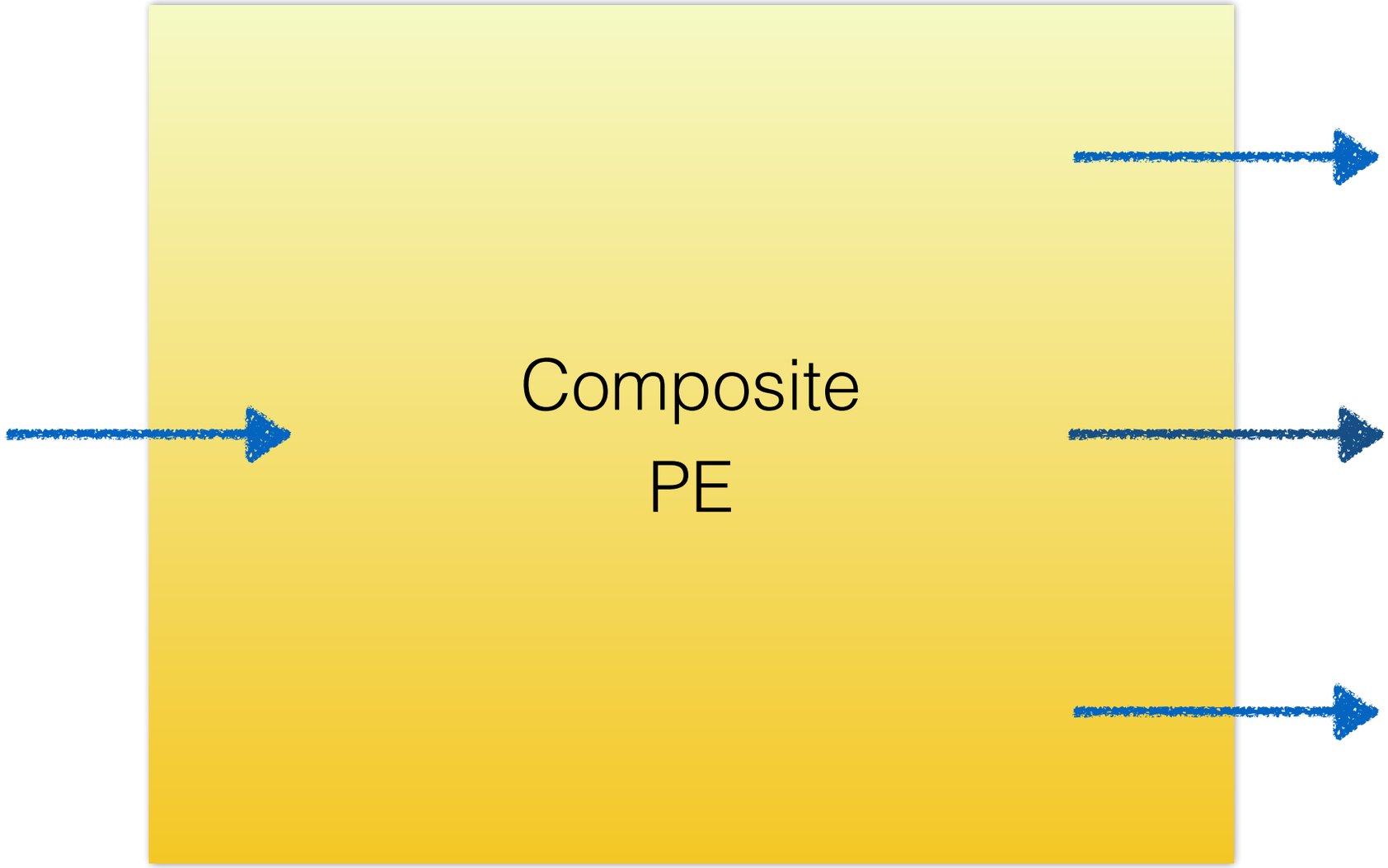


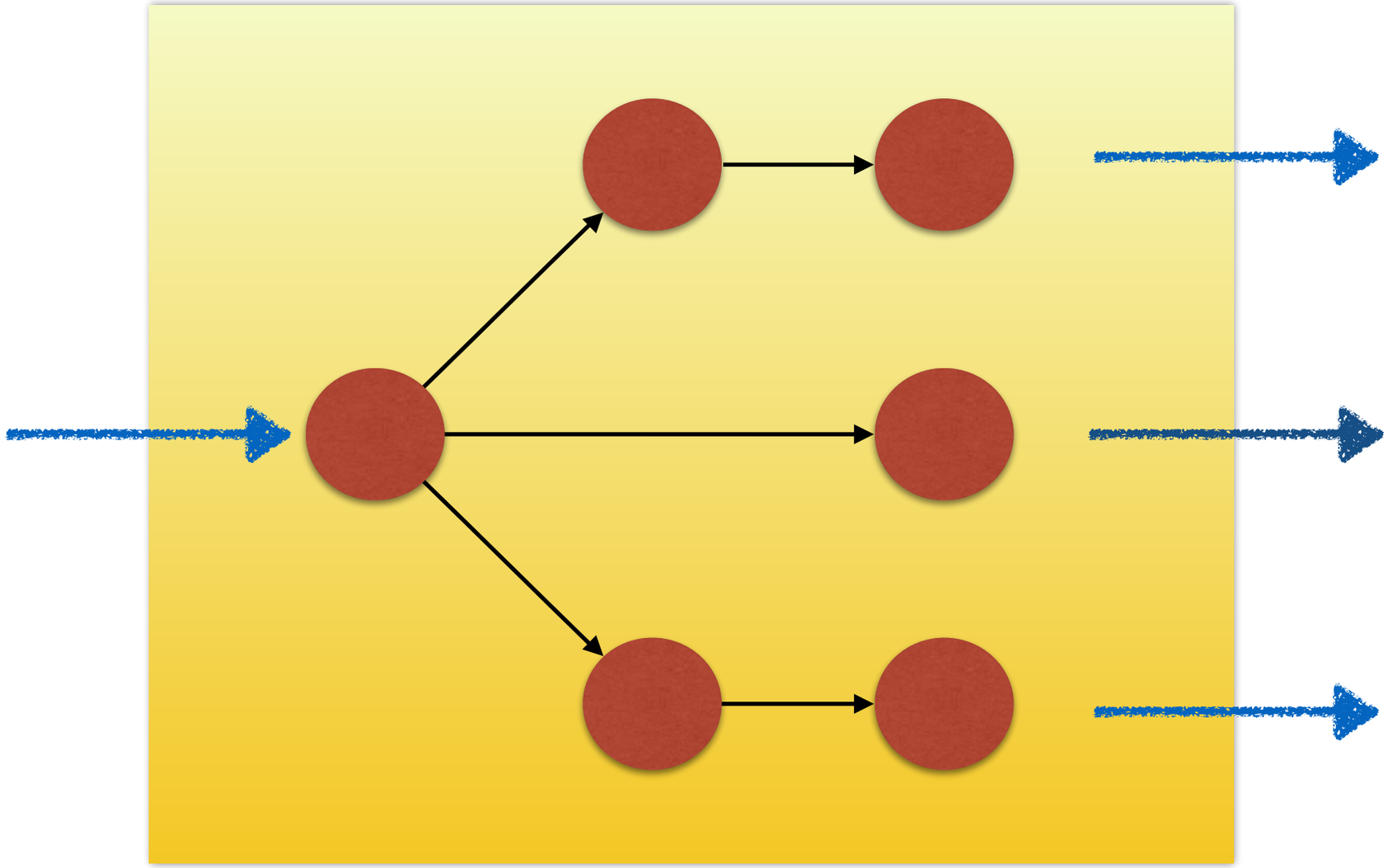
# Unconnected

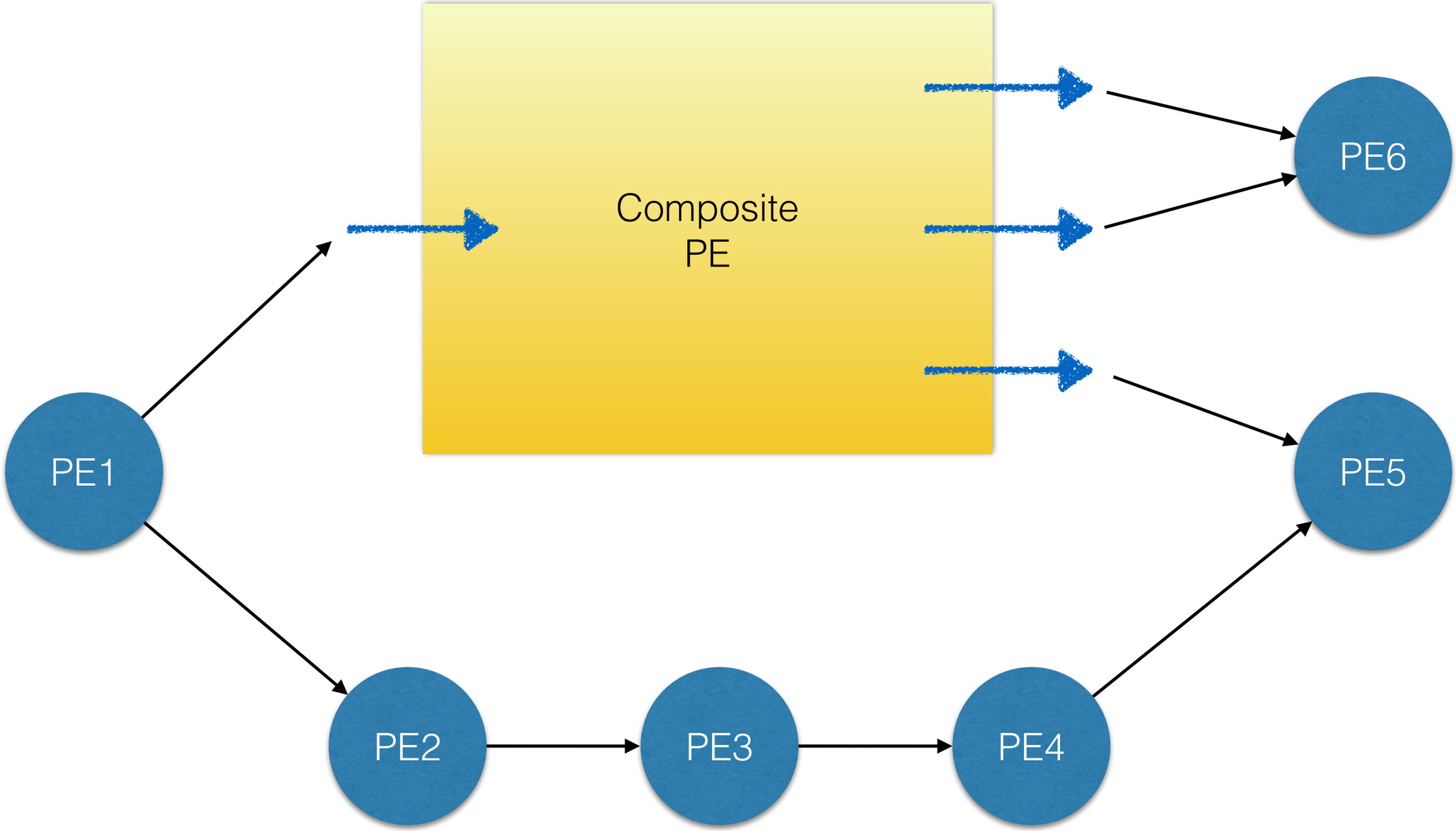


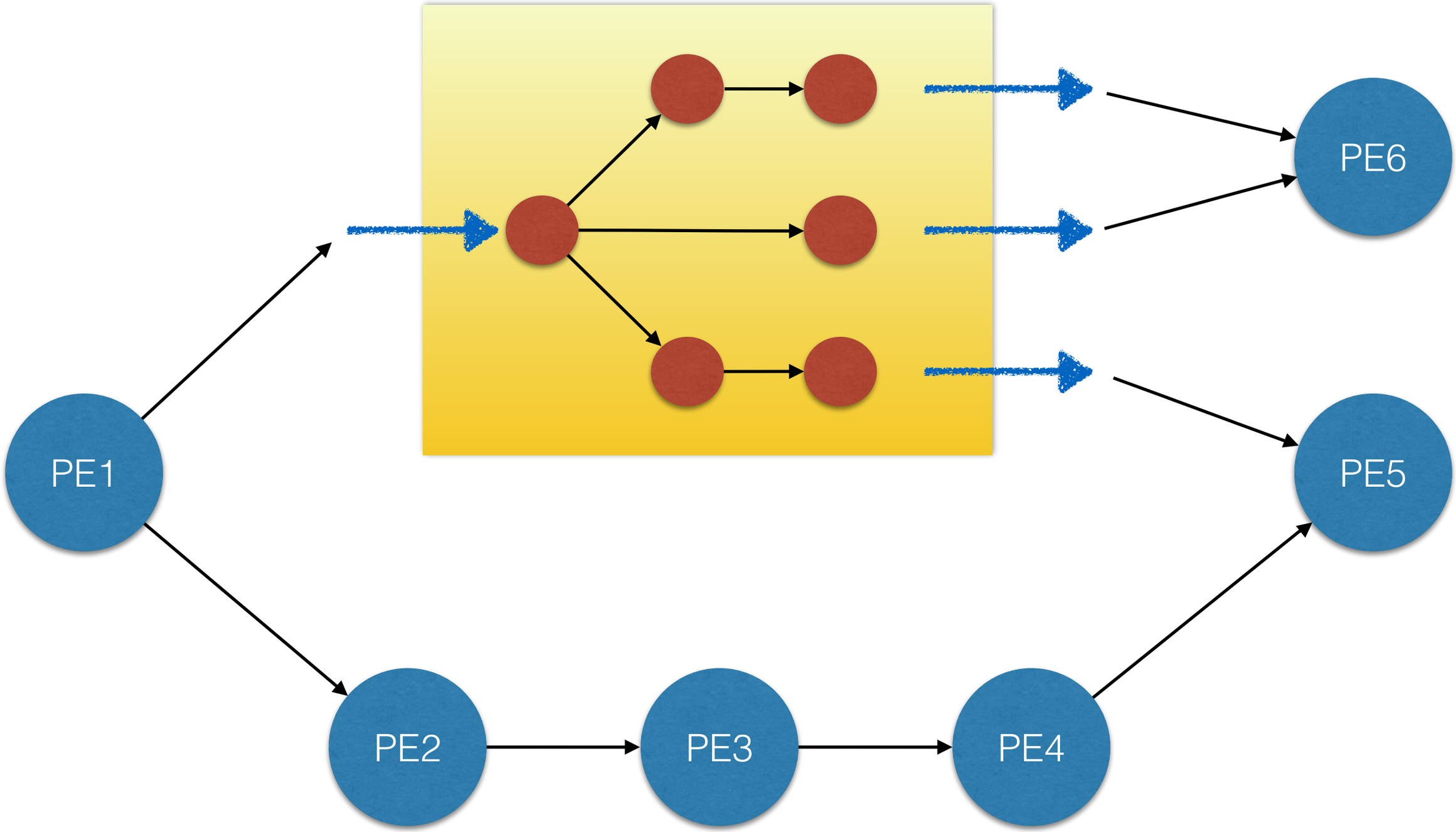
# Composite PEs

- A composite PE is a nested graph
- Looks like a PE but contains other PEs
- Hides the complexity of an underlying process
- When creating a graph, a composite PE is treated like any other PE


Composite
PE

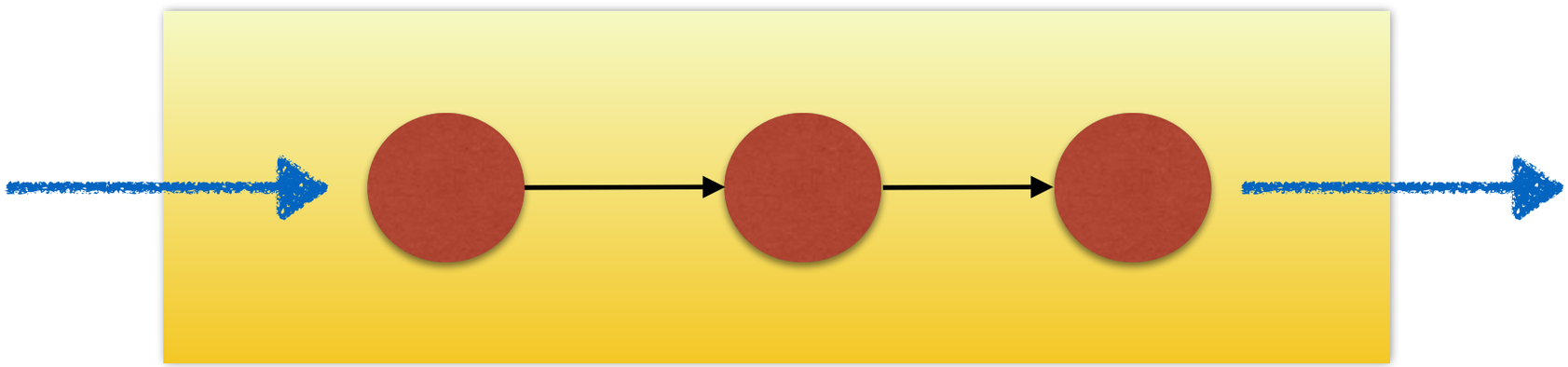



Composite
PE
PE1
PE2
PE3
PE4
PE5
PE6

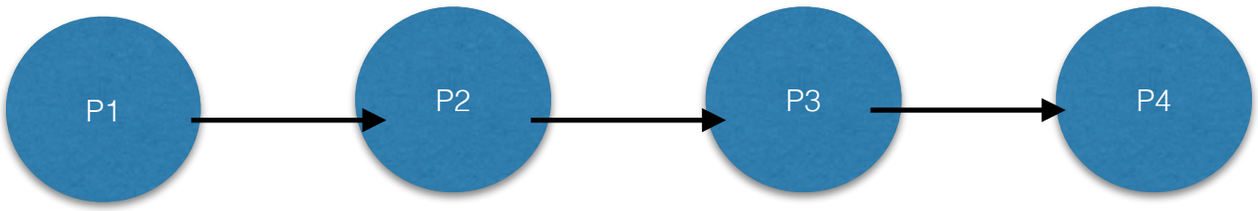

PE1
PE2
PE3
PE4
PE5
PE6

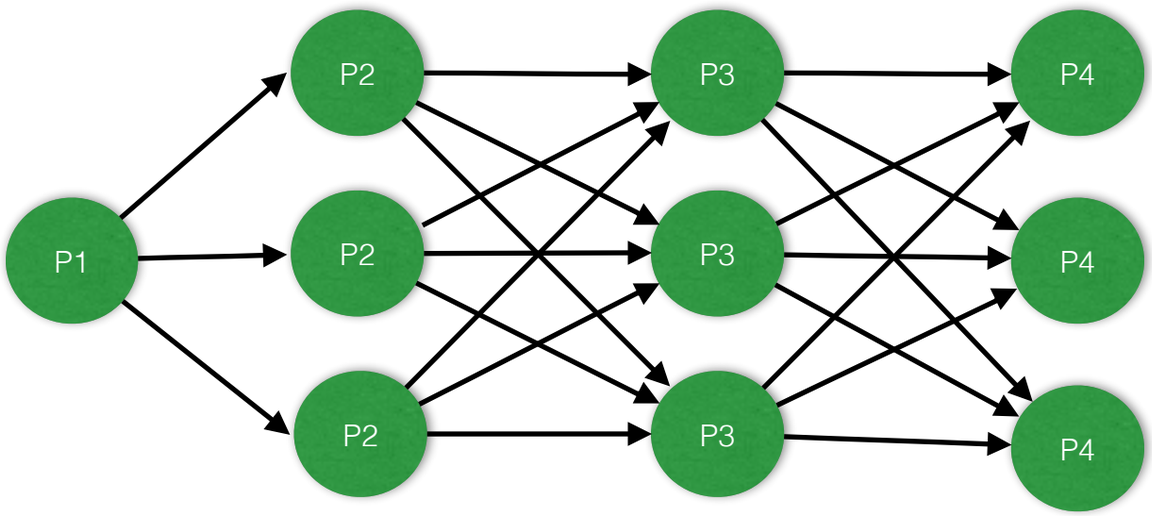
```
def add_value(num, value):
    num += value
    return num
def subtract_value(num, value):
     num -= value
     return num
pipeline = [ add_value, subtract_value, … ]
```

```
compositePE = create_iterative_chain(pipeline)
```



# Executing graphs

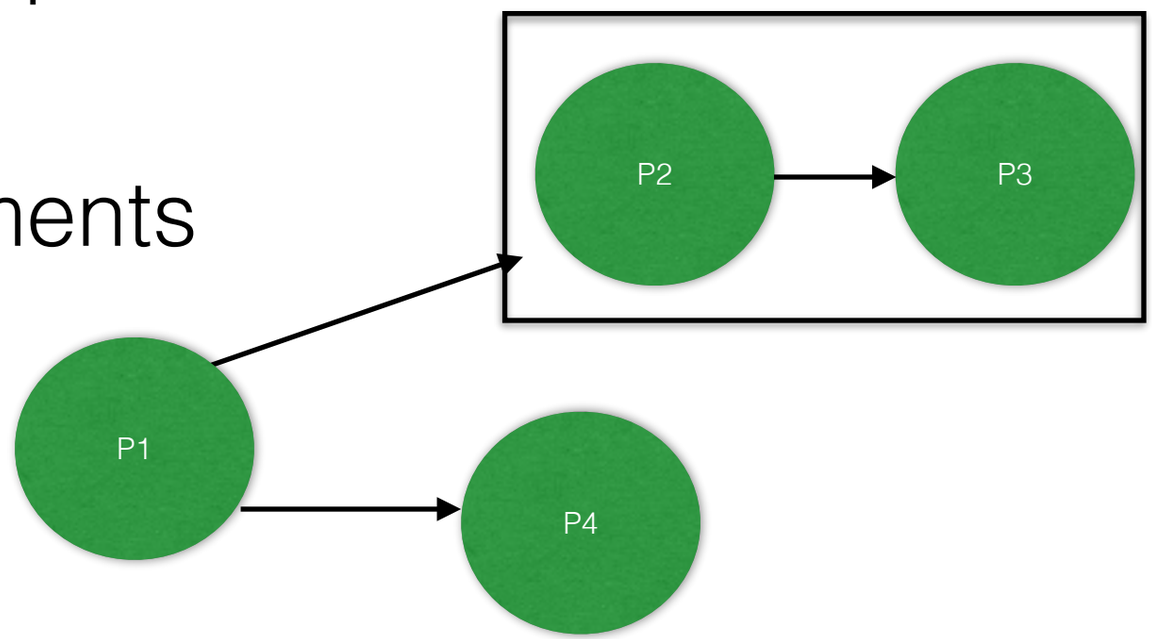


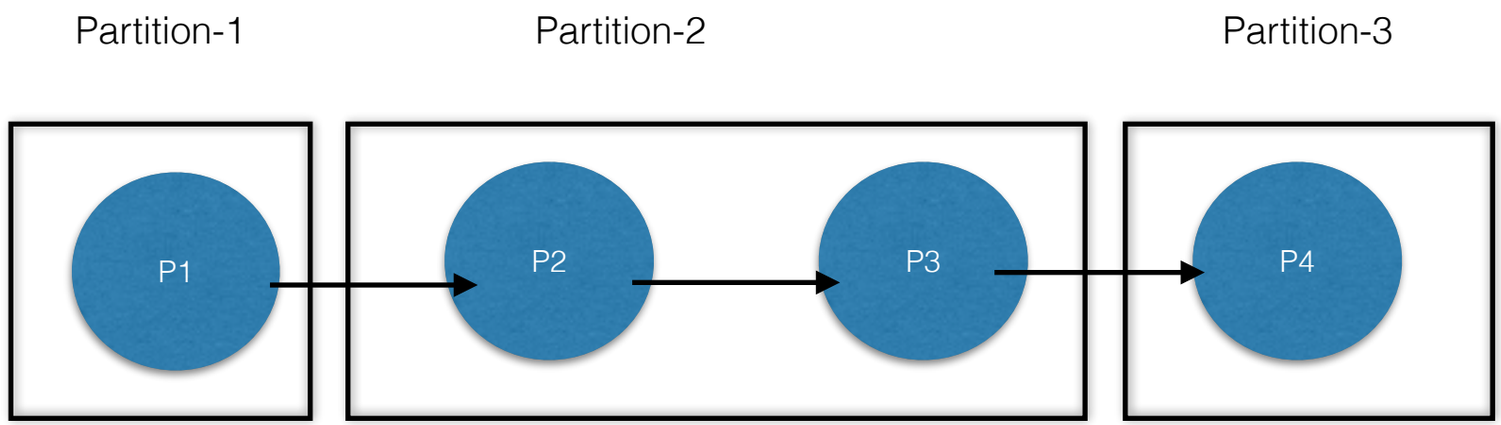



PE Instance

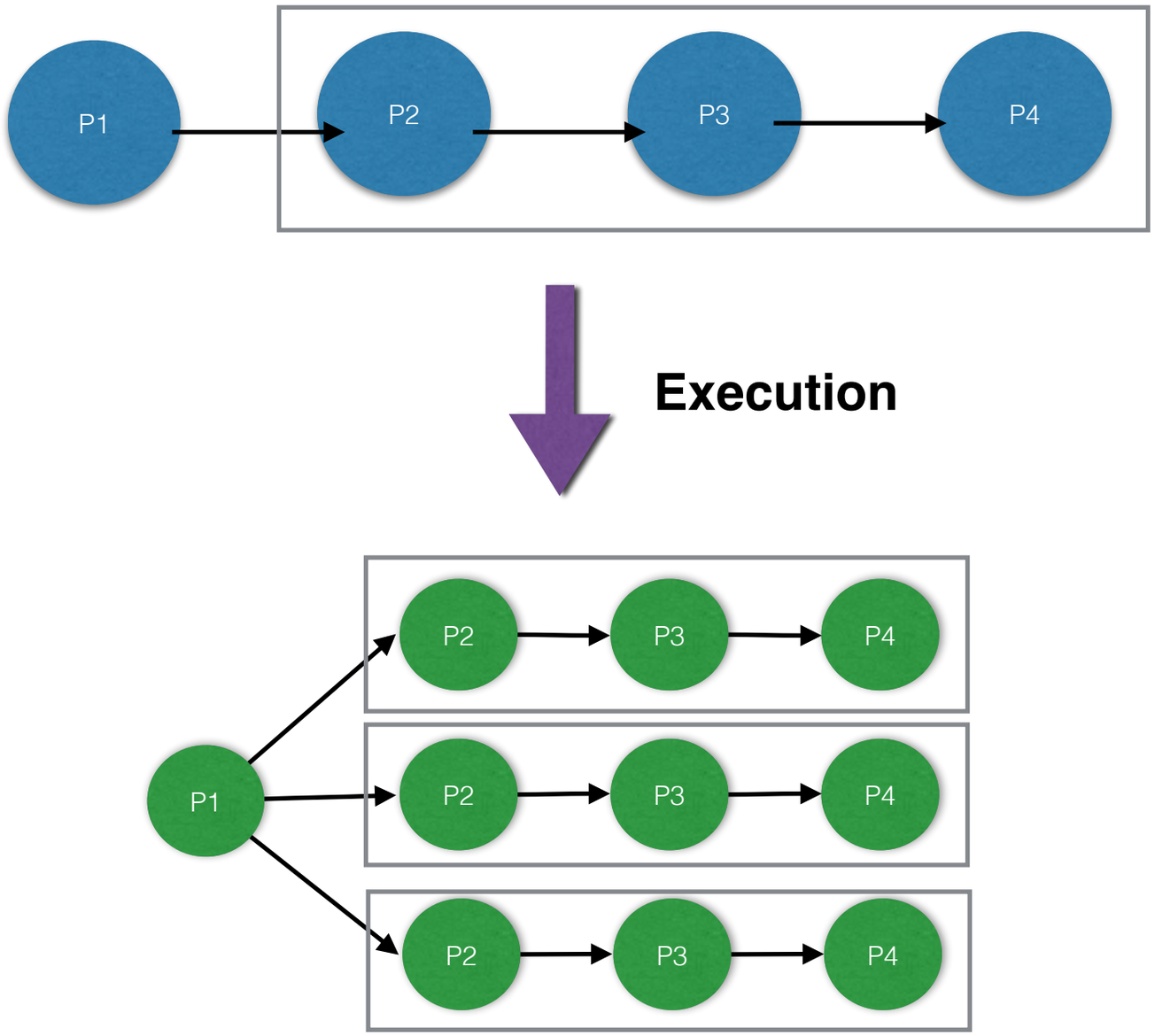
# Partitions

- Run several PEs in a single process
- User defined
- Applies to parallel environments



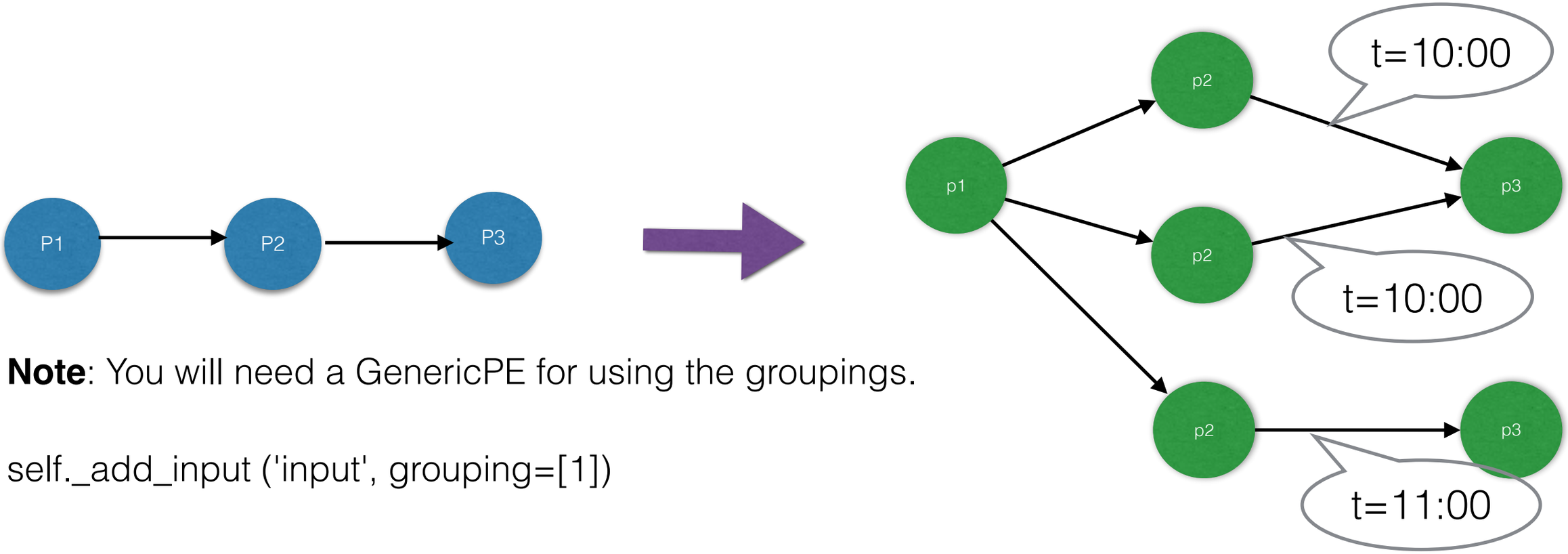
# Partitioned Pipeline



# Groupings

**“Grouping by” a feature (MapReduce)**

All data items that satisfy the same feature are guaranteed to be delivered to the same **instance** of a PE



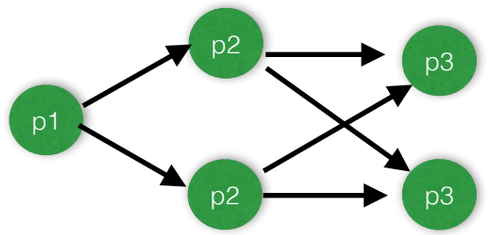

**Note**: You will need a GenericPE for using the groupings.

self._add_input ('input', grouping=[1])

# Other groupings

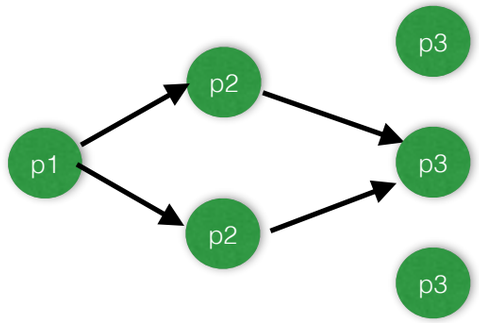
## One-To-All

**P3 - grouping “all”:**
P2 instances send copies of their output data to **all** the connected instances




## Global

**P3 - grouping “global”:**
All the instances of P2 send all the data to **one** instance of P3



# Mappings

**Simple process**

- Sequential mapping for local testing

**Multi process**

- Parallelism based on processes, using Python's multiprocessing library
- Shared memory

**MPI**

- Distributed Memory, message-passing parallel programming model
- Practical, portable, efficient, flexible and stable
- Many HPC centres support it, and it has been widely used in the HPC community

**STORM:**

- Distributed Real-Time computation System
- Fault-tolerant and scalable

# Running graphs

**Sequential mapping**
>> dispel4py simple <name_dispy_graph> <-f input_file in JSON format>
**E.g: dispel4py simple dispel4py.examples.graph_testing.pipeline_test**
**Multi-process mapping**
>> dispel4py multi <name_dispy_graph> -n <number mpi_processes> <-f input_file in JSON format> <-s>
**E.g : dispel4py multi dispel4py.examples.graph_testing.pipeline_test -n 6**

**MPI mapping**
>> mpiexec -n <number mpi_processes> dispel4py mpi <name_dispy_graph> <-f input_file in JSON format> <-s>
**E.g : mpiexec -n 6 dispel4py mpi dispel4py.examples.graph_testing.pipeline_test**

**STORM mapping**
>> python storm_submission.py <name_dispy_graph> <-m mode: [remote, local, create]>
**E.g : dispel4py storm dispel4py.examples.graph_testing.pipeline_test -m remote**

# Extra useful dispel4py information

- dispel4py commands:
  - dispel4py -h
  - -h, --help show this help message and exit
  - -a attribute, --attr attribute >> name of graph variable in the module
  - -f inputfile, --file inputfile >> file containing input dataset in JSON format
  - -d inputdata, --data inputdata >> input dataset in JSON format
  - -i iterations, --iter iterations >> number of iterations

# Extra useful dispel4py information

- inputs to the workflow:
  - dispel4py simple workflow.py -d ‘{“PE NAME CLASS" : [{"input" : “Hello World World”}]}’
  - dispel4py simple workflow.py -d ‘{"PE NAME CLASS" : [ {"input" : "coordinates.txt"} ]}’
  - dispel4py simple workflow.py -f coordinates.txt

# Thanks to

* **Amy Krause**
* **Malcolm Atkinson**
* **Alessandro Spinuso**

# Questions

* **Rosa Filgueira -> rosa.filgueira@ed.ac.uk**
* **Amy Krause -> a.krause@epcc.ed.ac.uk**
* **Malcolm Atkinson -> malcolm.atkinson@ed.ac.uk**
* **Alessandro Spinuso -> Alessandro.Spinuso@knmi.nl**