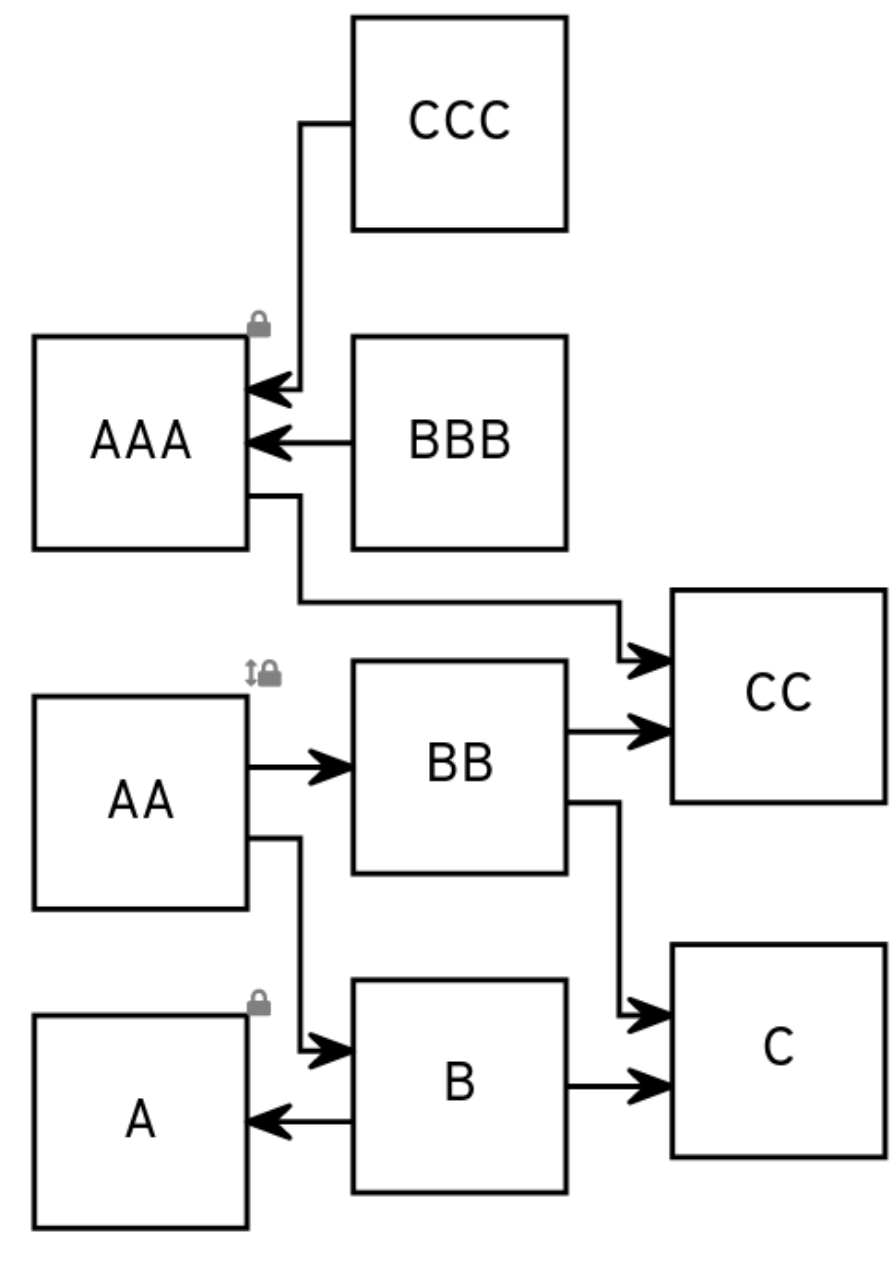
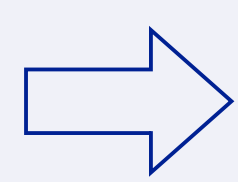


# An Interactive Graph Layout Constraint Framework

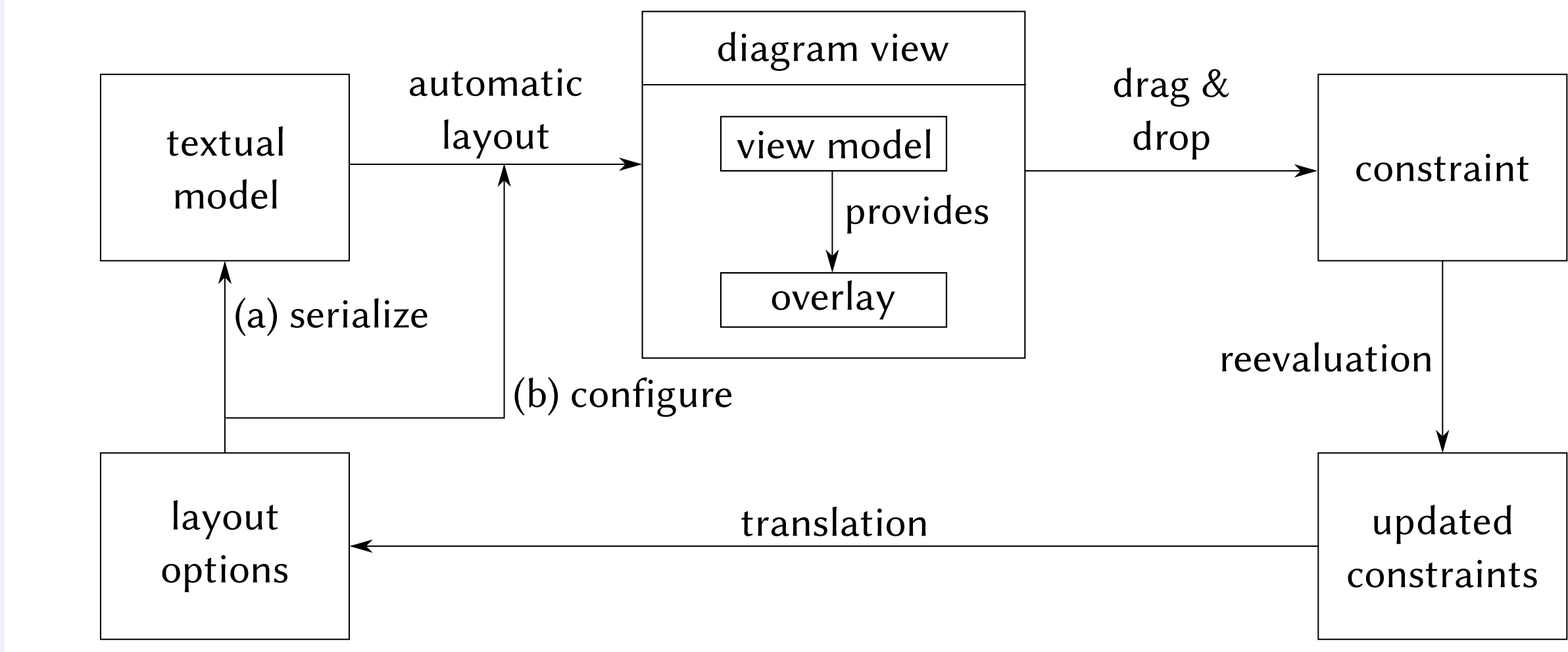
## Automatic Graph Drawing

```
interactiveLayout: true
node A {
  positionChoiceConstraint: 3
  layerChoiceConstraint: 0
}
node AA {
  positionChoiceConstraint: 2
}
node AAA {
  positionChoiceConstraint: 1
  layerChoiceConstraint: 0
}
node B
node BB
edge B -> A
...
```



Example of constraints in the textual source that are respected in the layout of the graph. Serializing constraints is language dependent and might not always be possible.

## Framework



The textual model is synthesized into a view model. This view model serves as a basis for an interactive overlay which allows setting layout constraints for the current algorithm via drag & drop. Introducing the constraints in a controlled manner allows updating existing constraints based on the intention of the user. Translating the constraints into layout options to use for the current algorithm leaves two options: Serializing them or using them to configure the layout.

## Example - Layered Layout Constraints

### Absolute Constraints

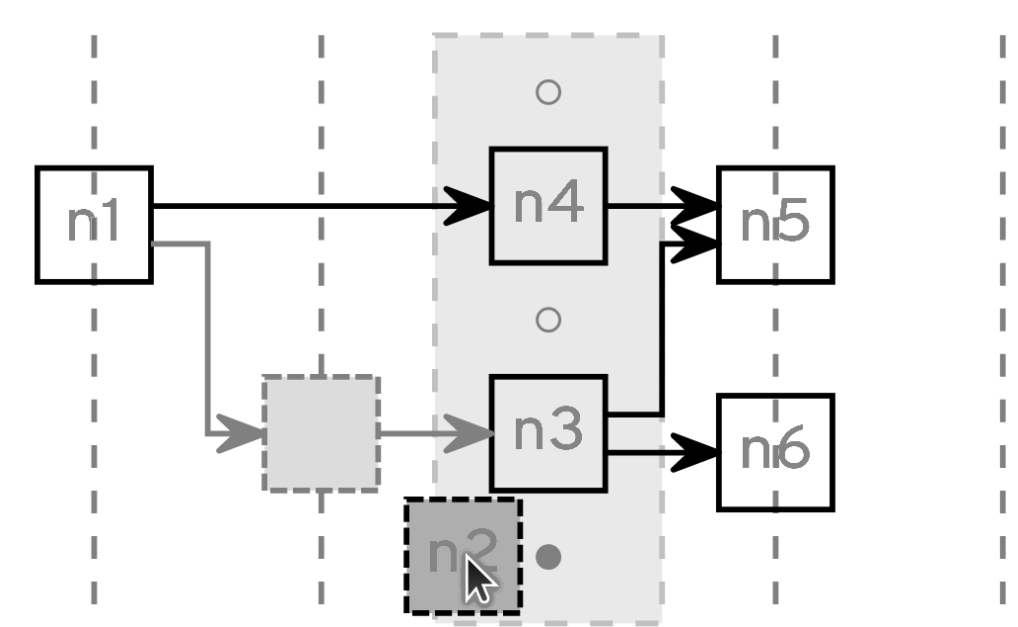
#### Layer Constraint

The layer of a node, where 0 is the first layer.

#### Position Constraint

The position of a node in the layer, where 0 indicates the first position.

#### Overlay



### Relative Constraints

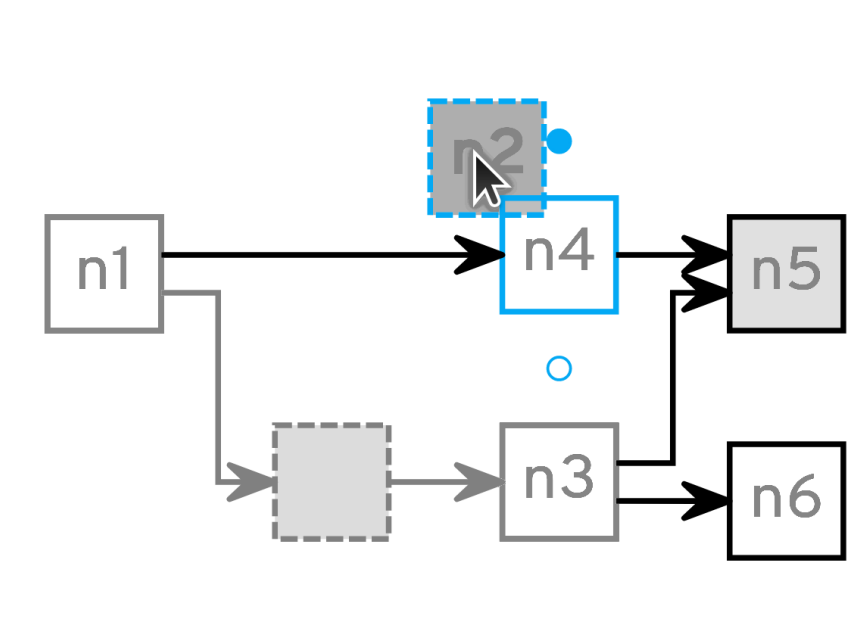
#### Predecessor-of Constraint

The node is placed above the target node.

#### Successor-of Constraint

The node is placed below the target node.

#### Overlay

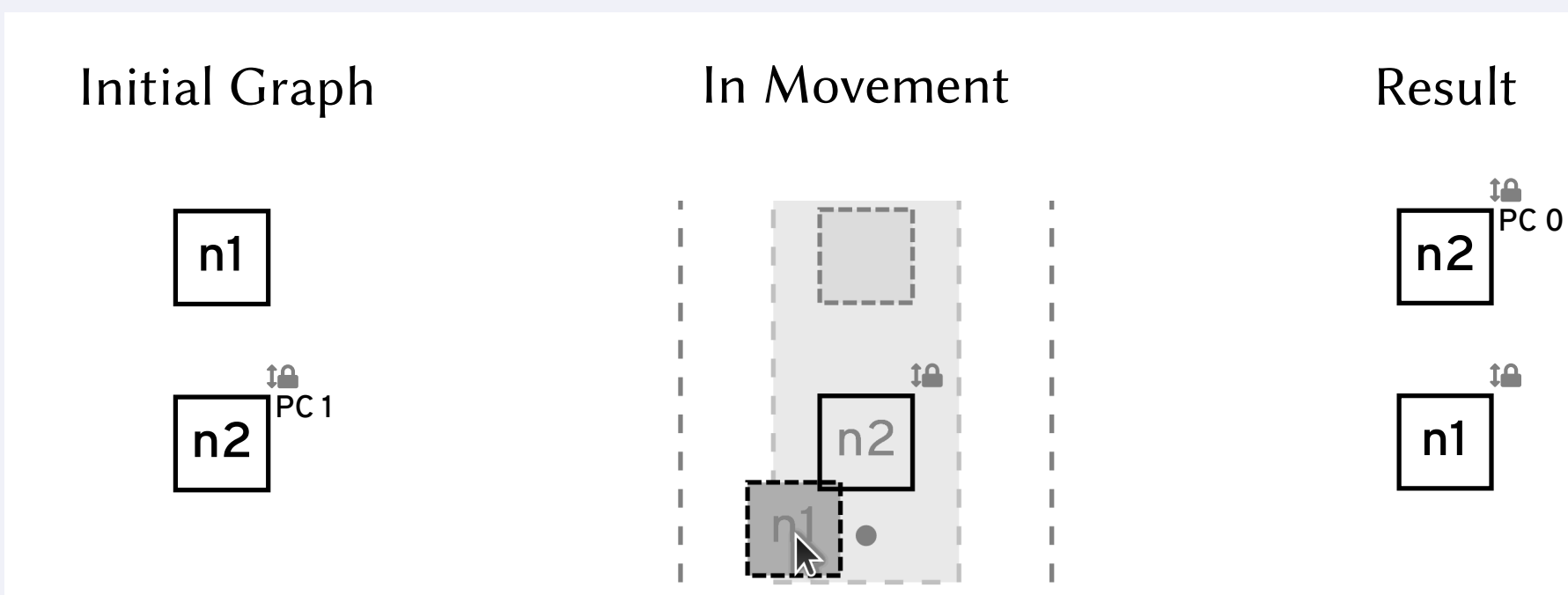
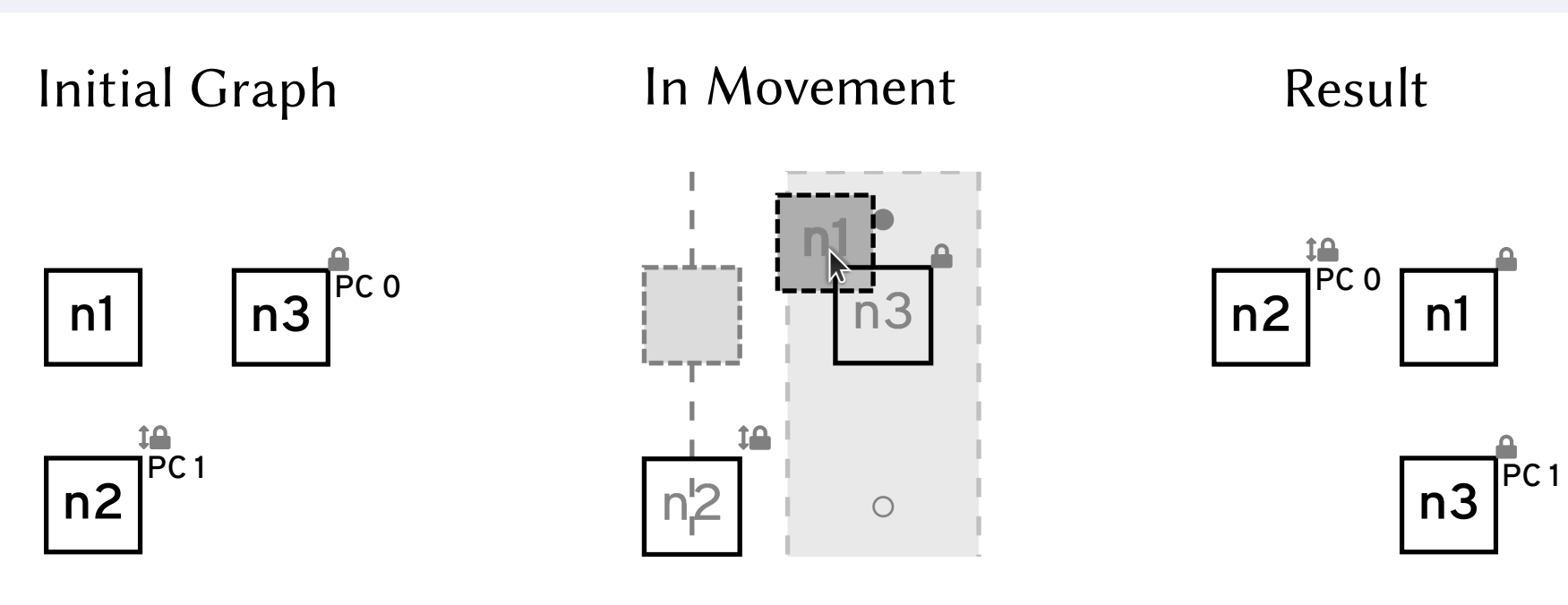


Available position constraints are visualized by the circles and available layer constraints by dashed lines. The current layer is highlighted as a rectangle and the current position is indicated by the filled circle. A shadow shows the original position of the moved node.

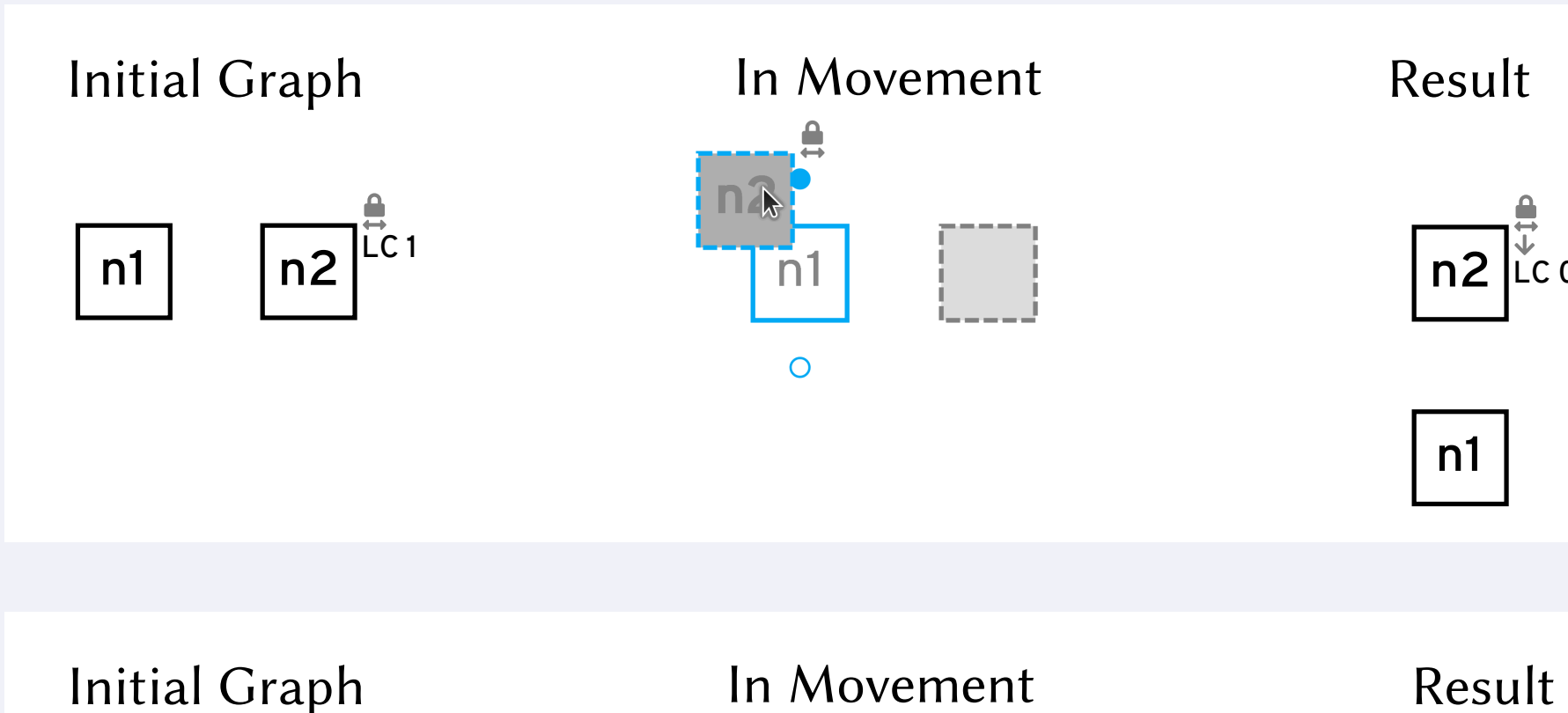
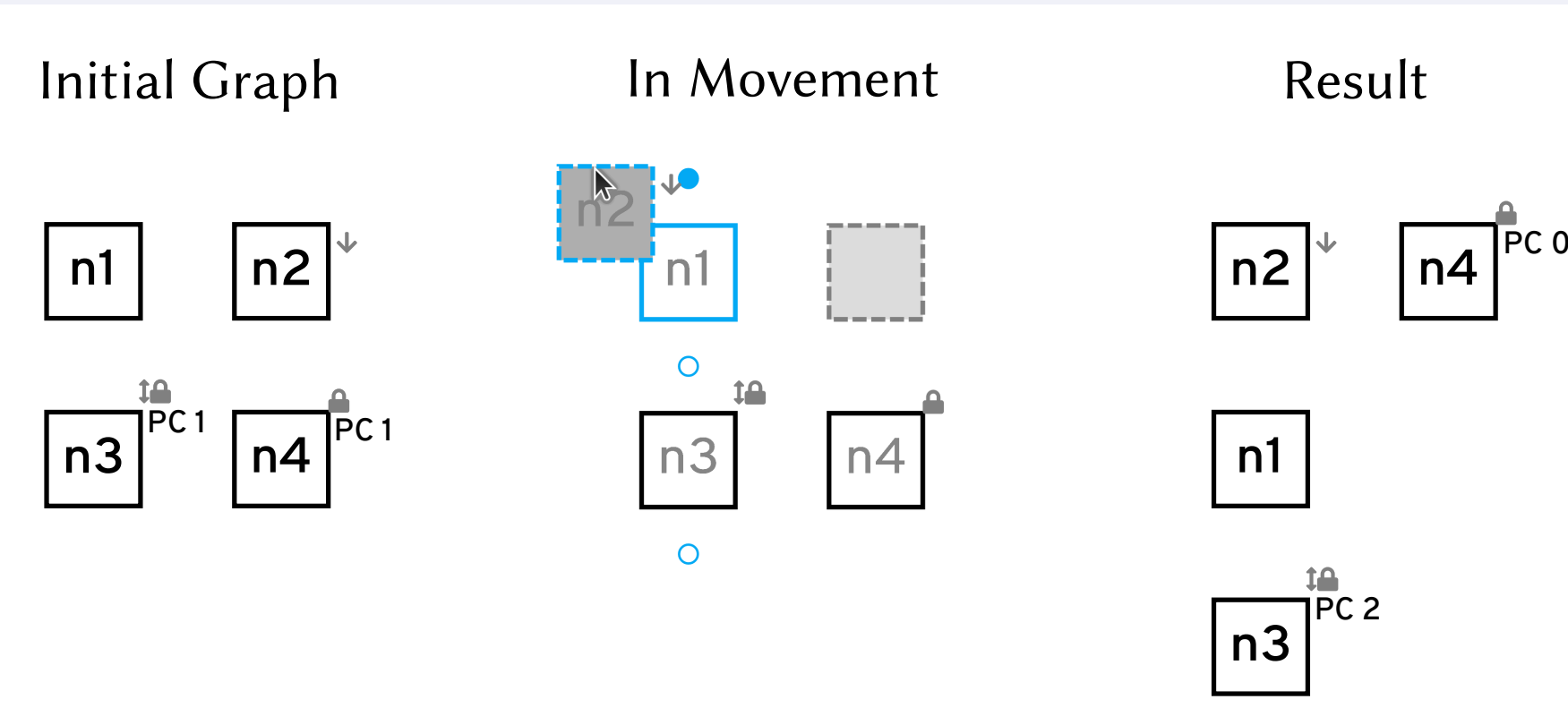
The target node is highlighted in blue and the constraint is indicated by the circles. If the circle above the target is filled, a predecessor-of constraint is set. If the circle below the target is filled, a successor-of constraint is set.

## Example - Layered Layout Reevaluation

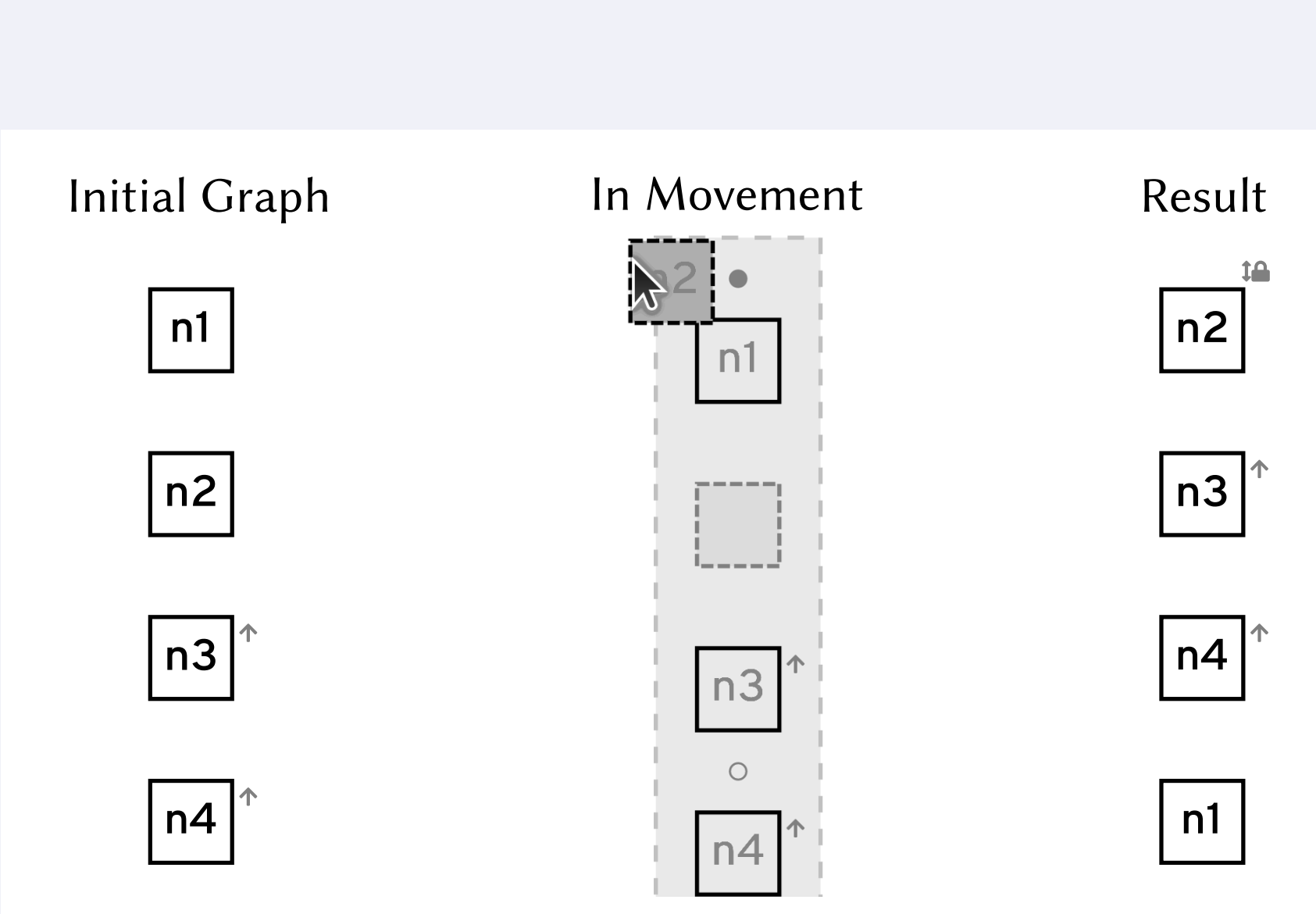
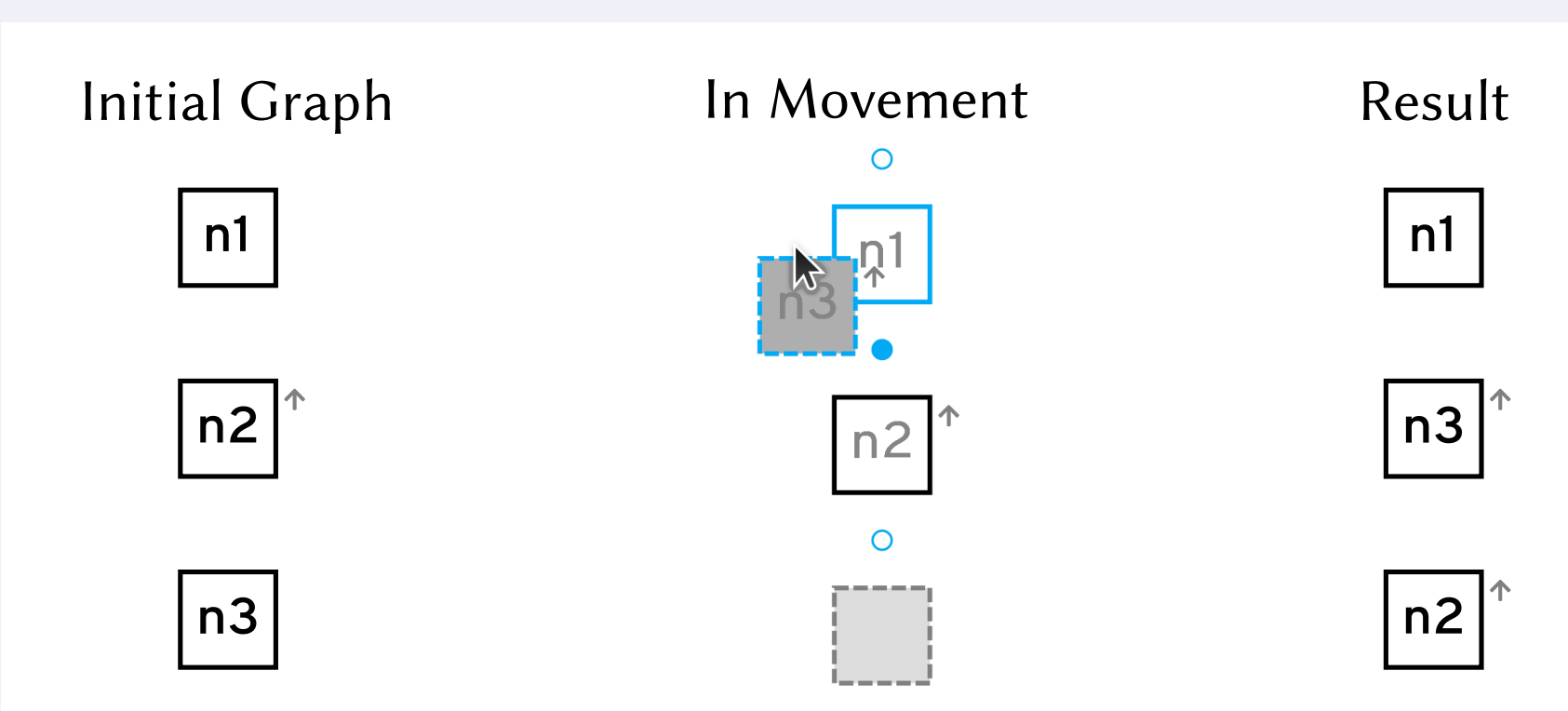
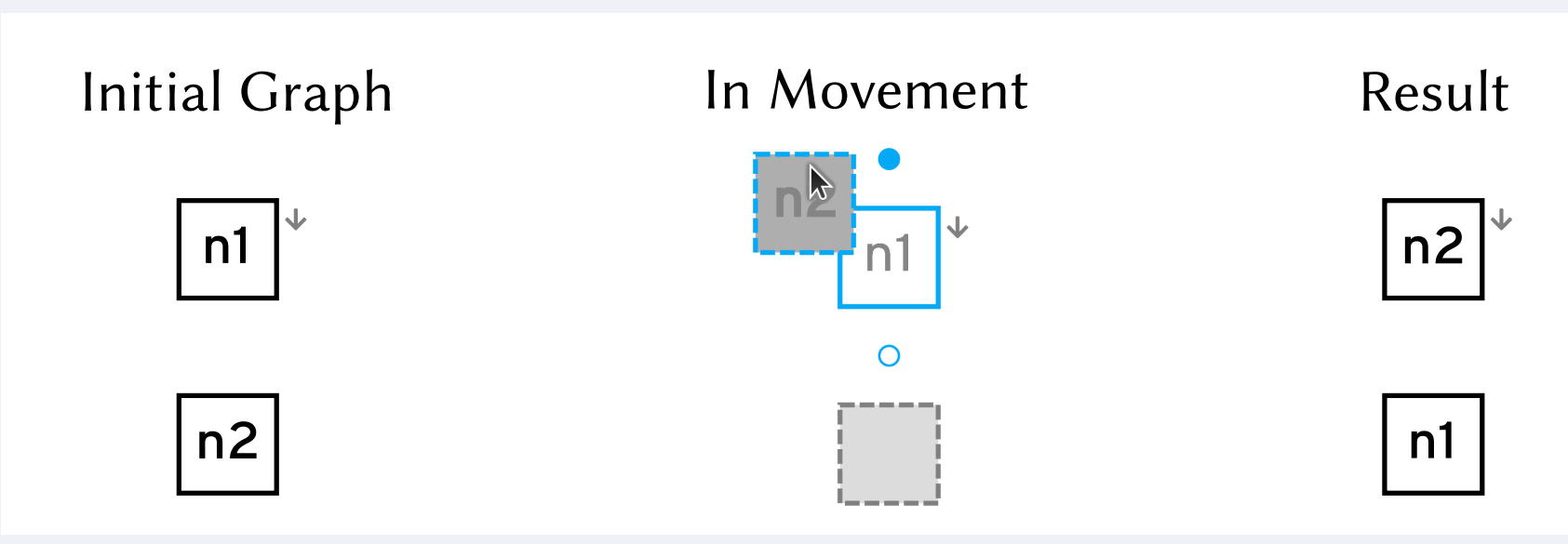
### Absolute Constraints



### both



### Relative Constraints



## Contact Persons

Jette Petzold, Sören Domrös, Connor Schönberger,  
Prof. Dr. Reinhard von Hanxleden  
Department of Computer Science, Kiel University,  
Olshausenstr. 40, 24098 Kiel, Germany  
Phone: +49 (0) 431 880-7525  
jep@sdo@cos@rvh@informatik.uni-kiel.de  
<http://rtsys.informatik.uni-kiel.de>

## Related Publications

- [1] K. F. Böhringer, F. N. Paulisch. Using constraints to achieve stability in automatic graph layout algorithms. In Proc. of the SIGCHI Conference on Human Factors in Computing Systems, New York, 1990
- [2] M. J. McGuffin, I. Juristica. Interaction Techniques for Selecting and Manipulating Subgraphs in Network Visualizations. IEEE transactions on visualization and computer graphics, 2009

## The Framework is part of



**KIeler**  
The Key to Efficient Modeling

On the web:  
<https://github.com/kieler>