

# A model for social news streams & time indices in graph data bases

Institute for Web Science and Technologies, University of Koblenz-Landau

## Goal

In a social network application social news streams that aggregate status updates from your social circle are a key feature.

We have looked into the data model behind the social news streams of Twitter and Facebook and propose a new model that is based on graphs and is more efficient than Facebook's model but also respects the advantages of Twitter's approach.

## Requirements

- Fast reading
- Scaling
- Flexibility

## Assumptions

- Reading >> writing
- Node degree >> stream length
- Content items are independent

## Twitter ( FlockDB )

- 💡 Reading ok
- 💡 Flexible
- 💡 Low storage
- ⚡ Joins don't scale

## Facebook ( Files )

- 💡 Scaling
- 💡 Reading is fast
- ⚡ Low flexibility
- ⚡ High storage

## Results:

### Reading

- $O(1)$  in network size
- $O(1)$  in node degree (!)
- $O(n)$  in stream length

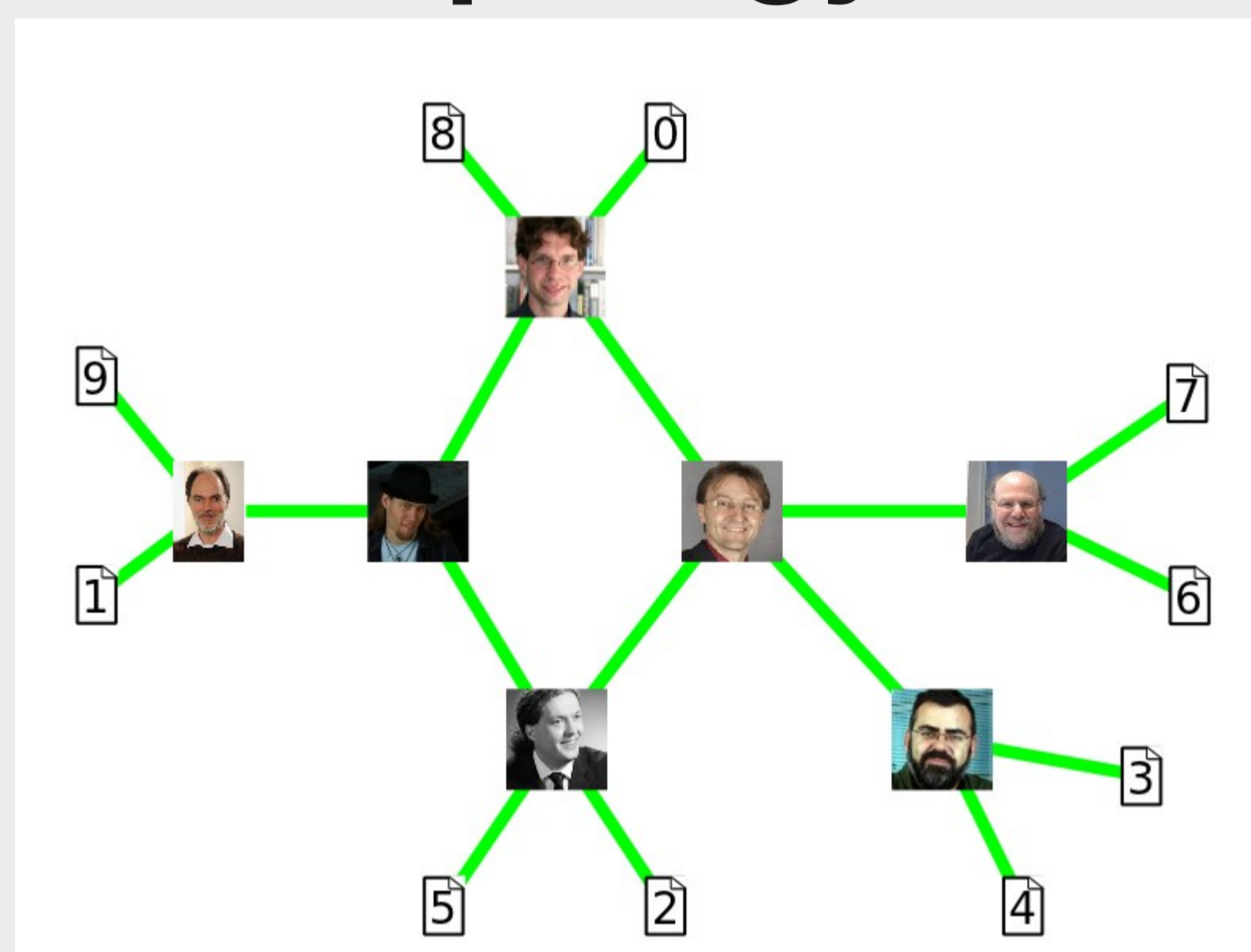
### Writing

- $O(n)$  in node degree

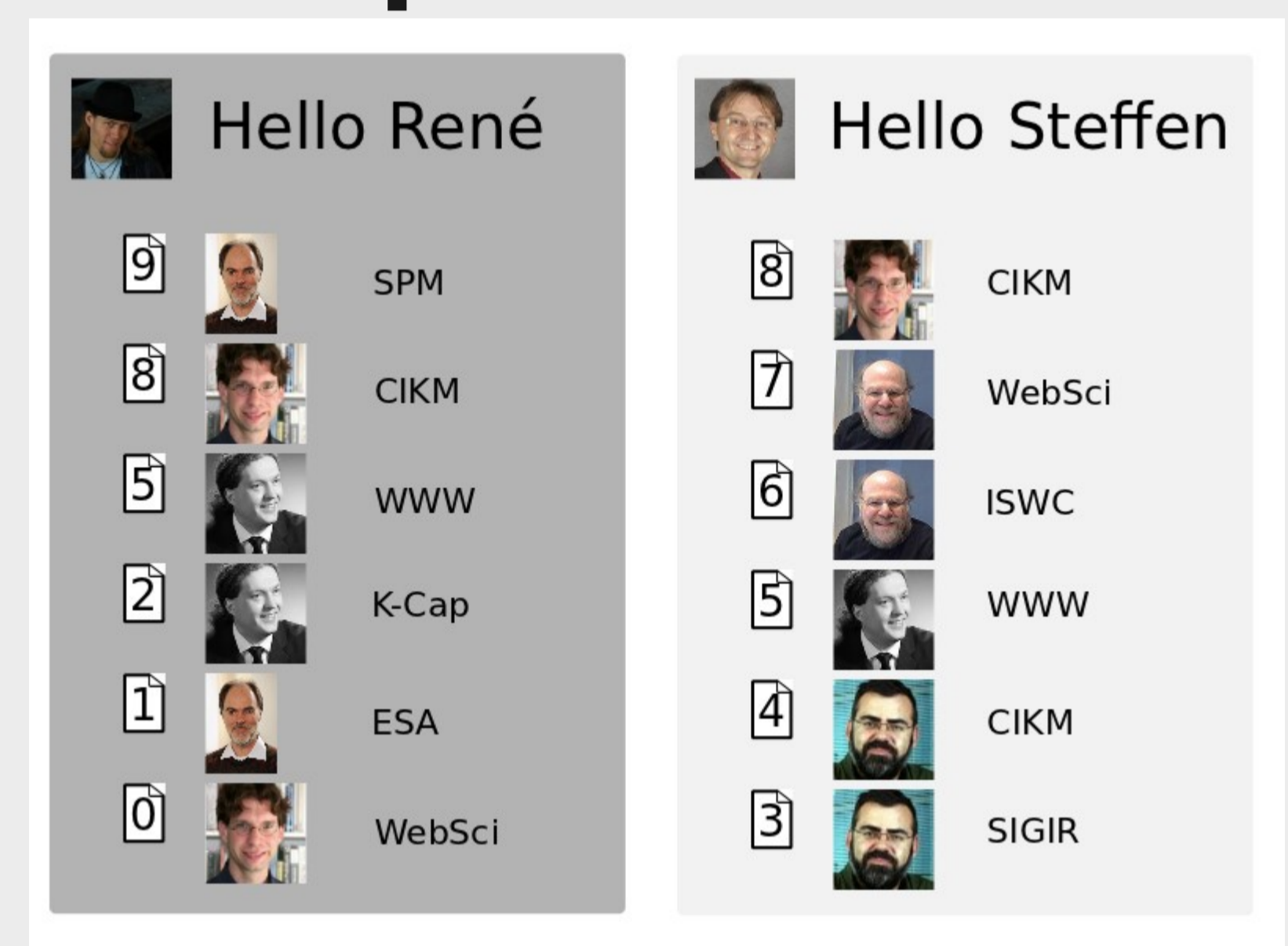
### Storage

- $O(n)$  in network size
- Low redundancy
- Twice as many edges required

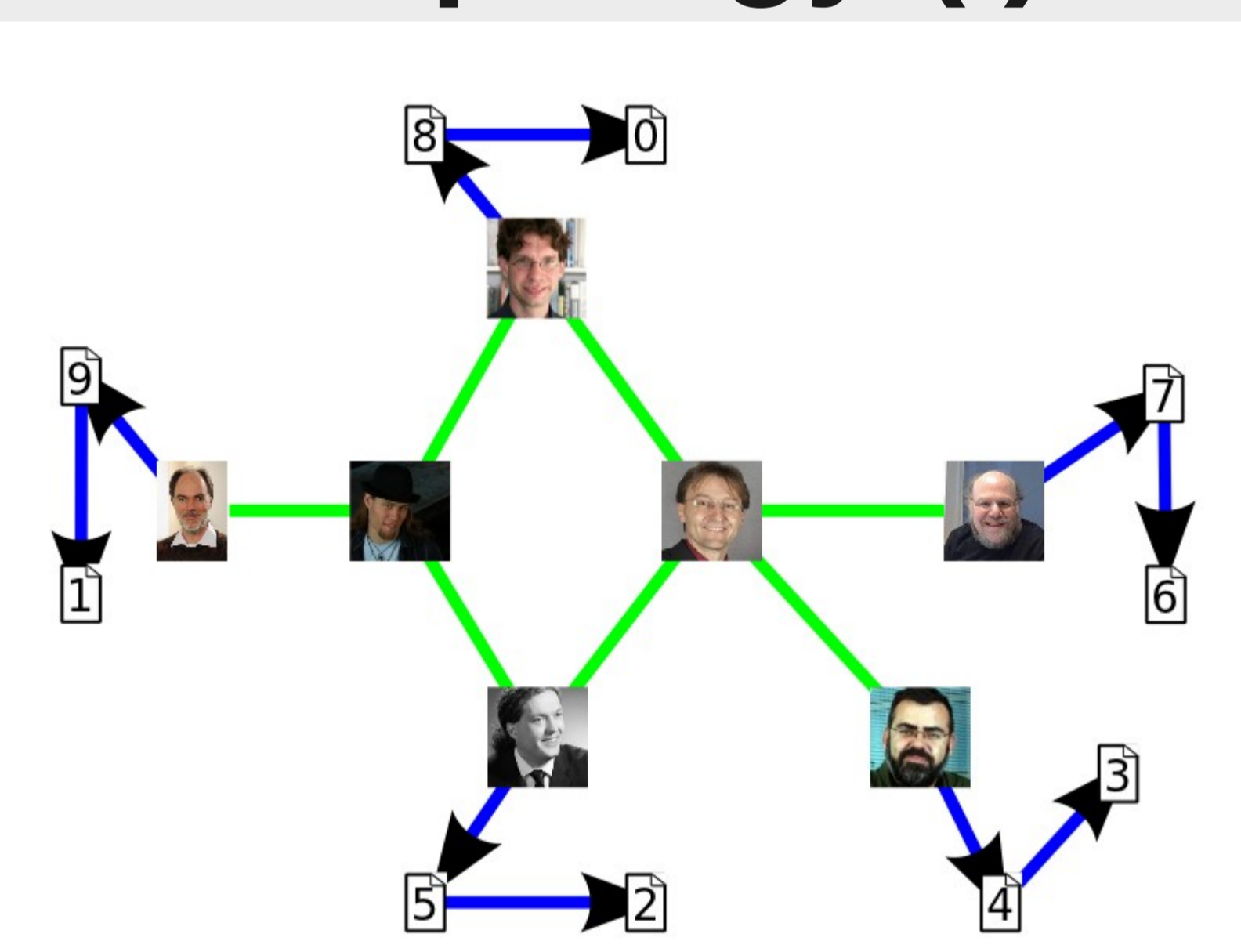
## Star topology



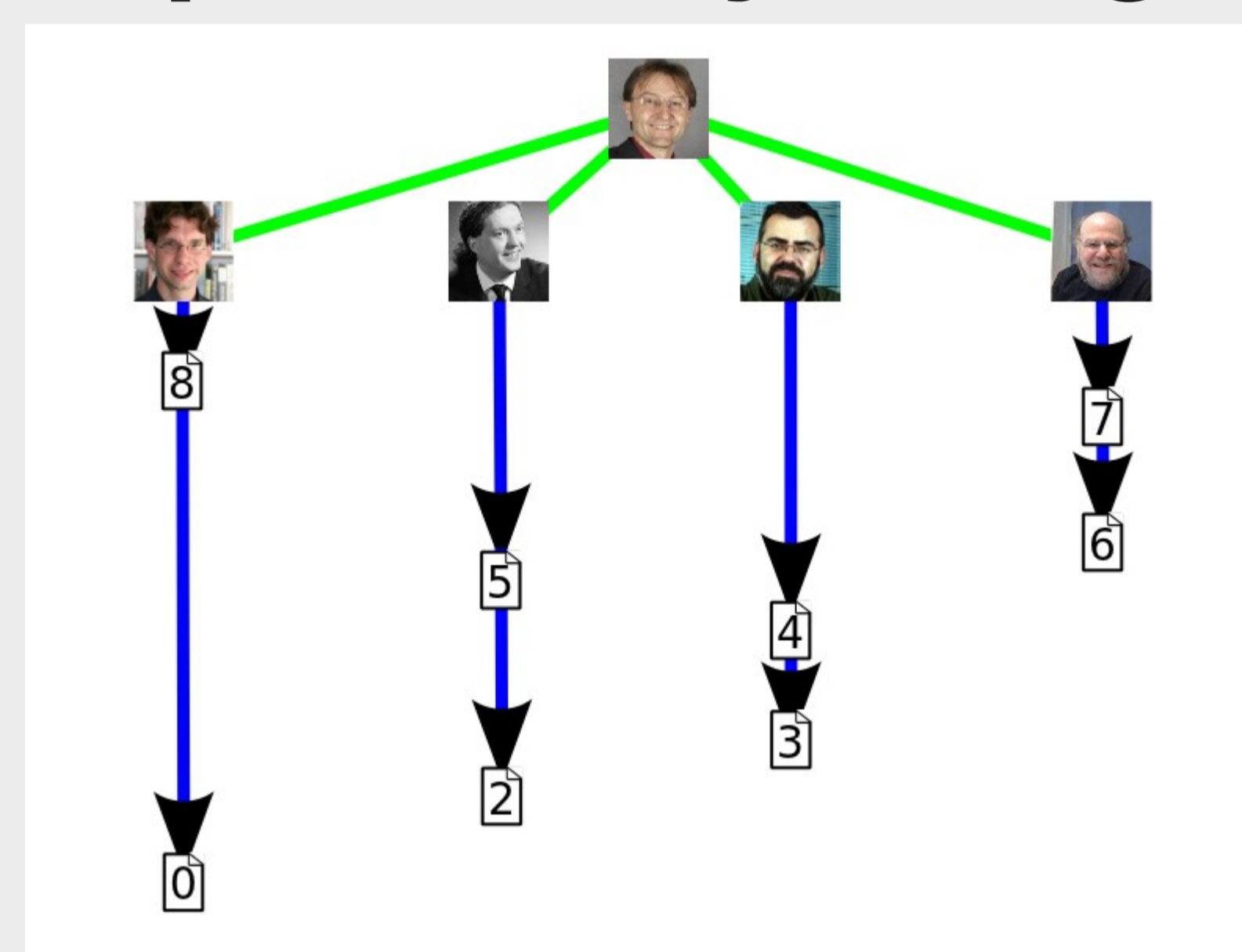
## Example



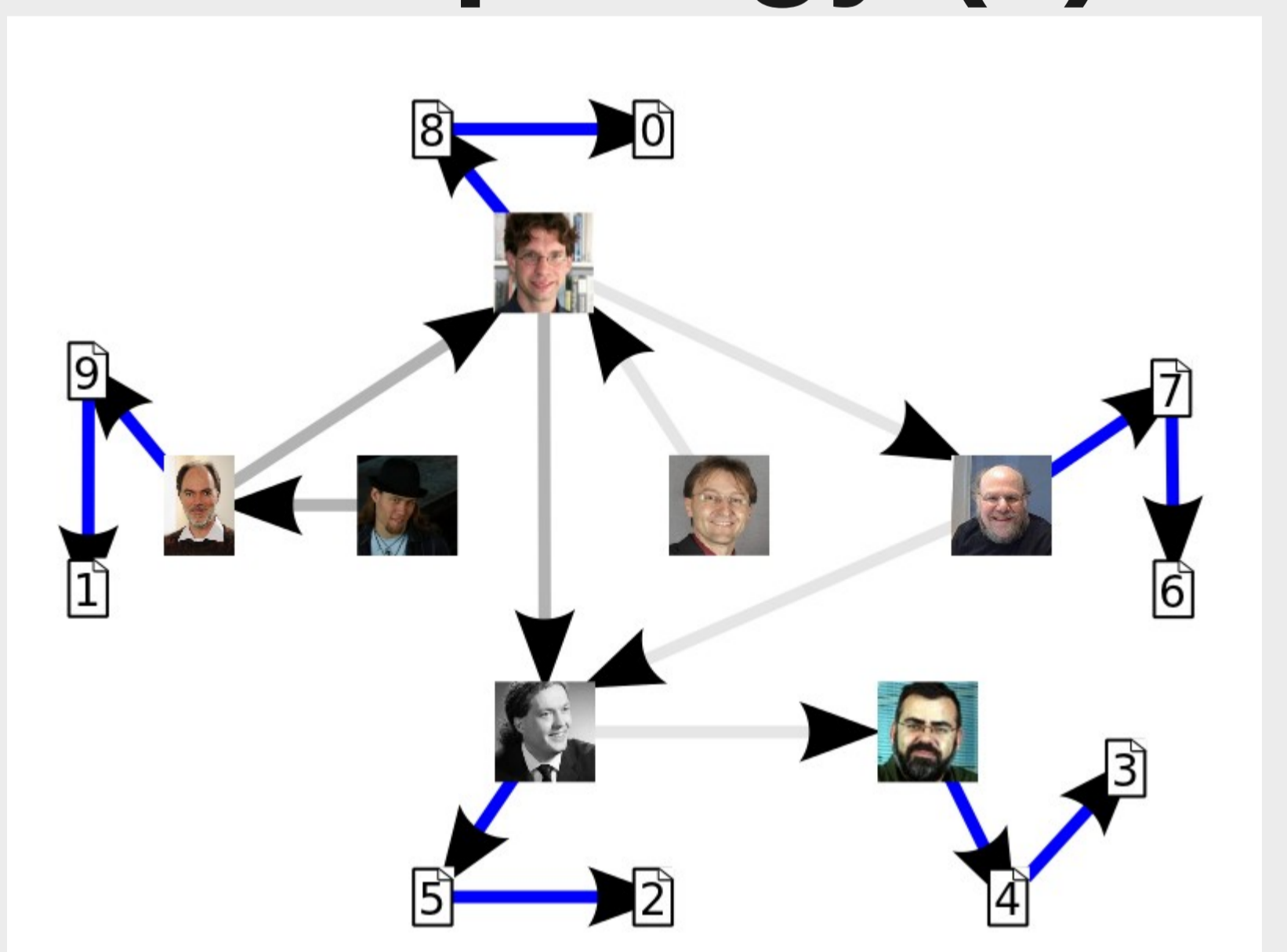
## List topology (I)



## Top-k n-way merge



## List Topology (II)



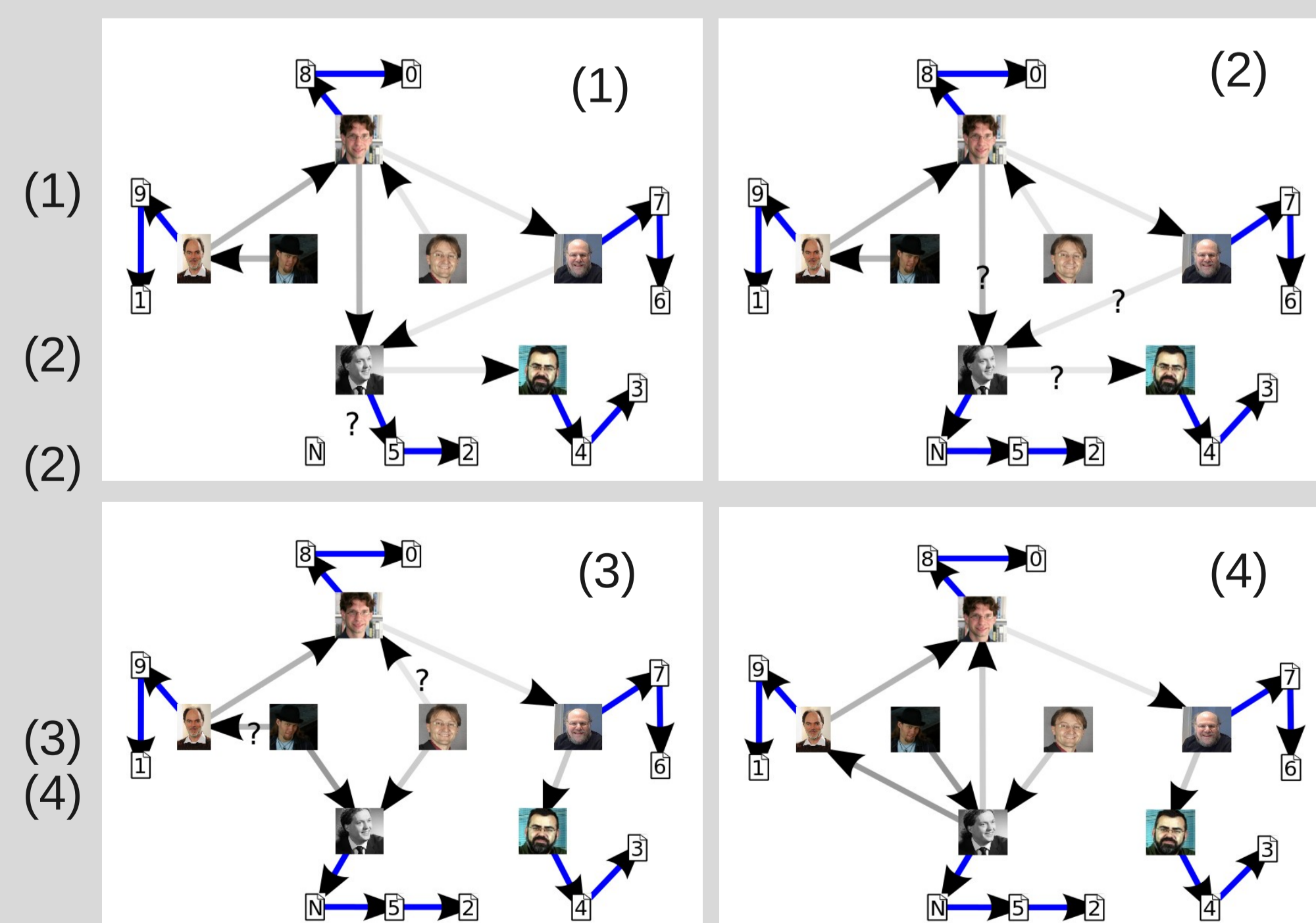
## Maintain the Index

Let us assume a new Content Item N is created and shall be linked to an entity X. How is the time index to be maintained?

Link Content to entity X  
use star topology to calculate EgoNet(X)  
(Same as BFS depth 1)

for Y in EgoNet(X)

- ET = edge type of Y
- On X with Respect to ET
- P = predecessor(X); S = successor(X)
- Link P to S
- With respect to ET insert X to the beginning of EgoNet(Y)



## more Assumptions

- Named graphs
- Choosing edges from same edge type can be done in  $O(1)$
- Graph data base can be parallelized (scaling)
- Star topology still available
- New edges in the Graph will also produce a new Content item at the nodes