

# Constellation: Programming decentralised social networks

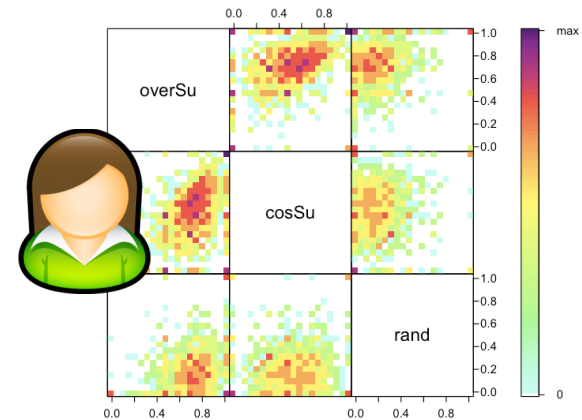
Anne-Marie Kermarrec, François Taïani



# Summary

## ■ Constellation: DSL for self-organising social networks

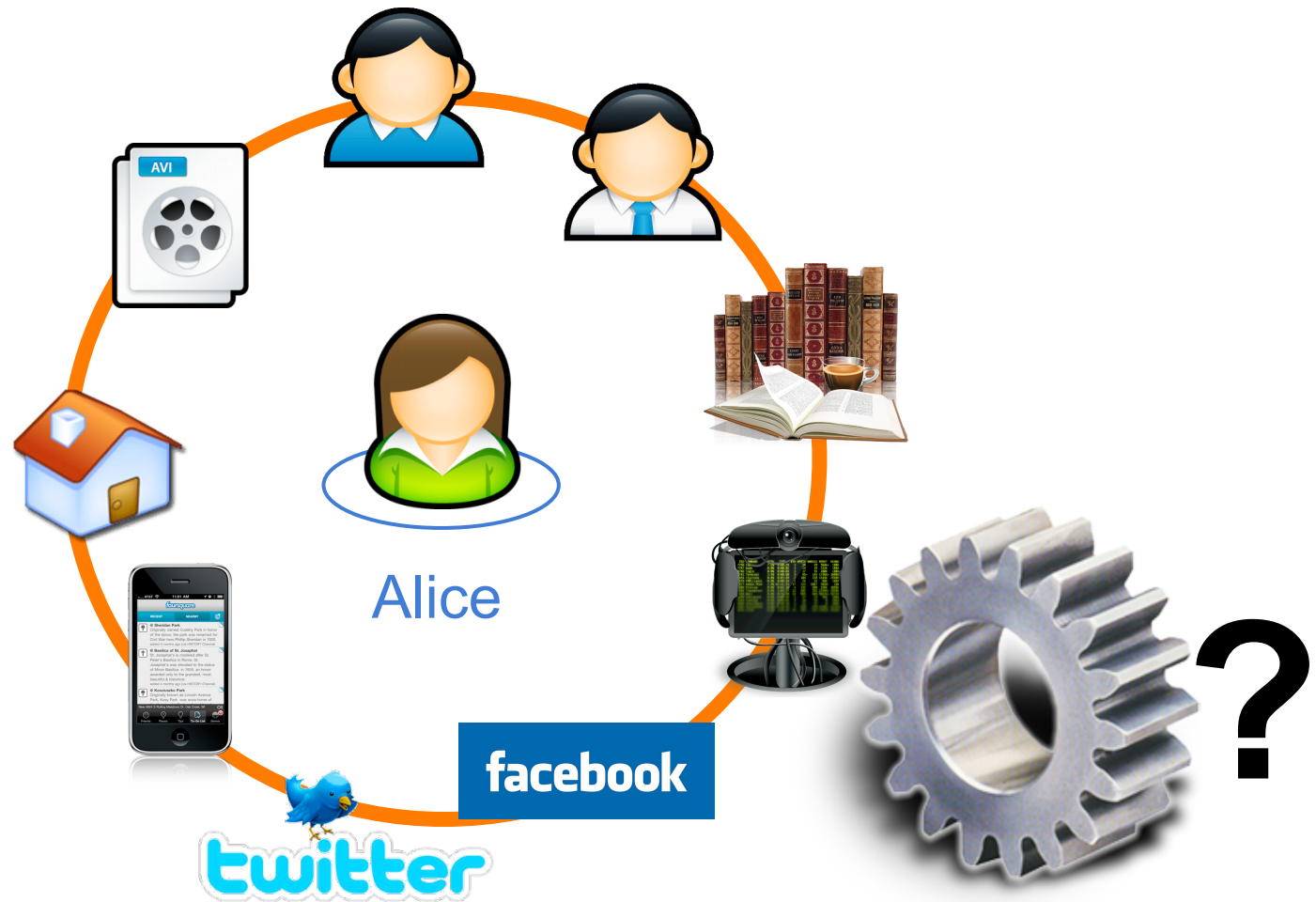
- **Rapid** experimentation
- **Composable**, recursive
- Still under development



## ■ Benefits

- Not much in terms of language research (?)
- Dist. Sys. : organise **design space**, help **exploration**
- Not limited to social networks (embryomorphing eng.)

# Towards a User-Centred Web

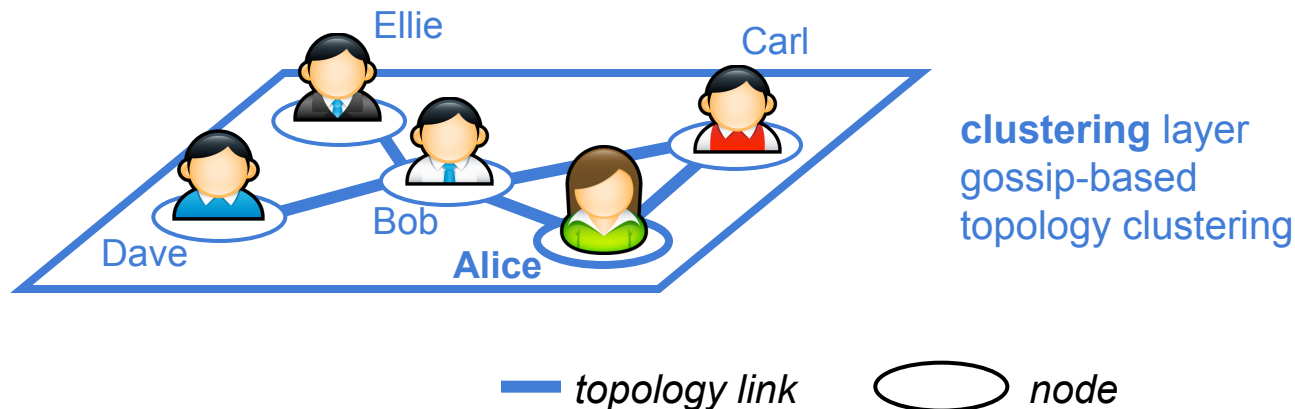


- Privacy + Personalisation → Decentralisation

# Gossip-based Social Networks

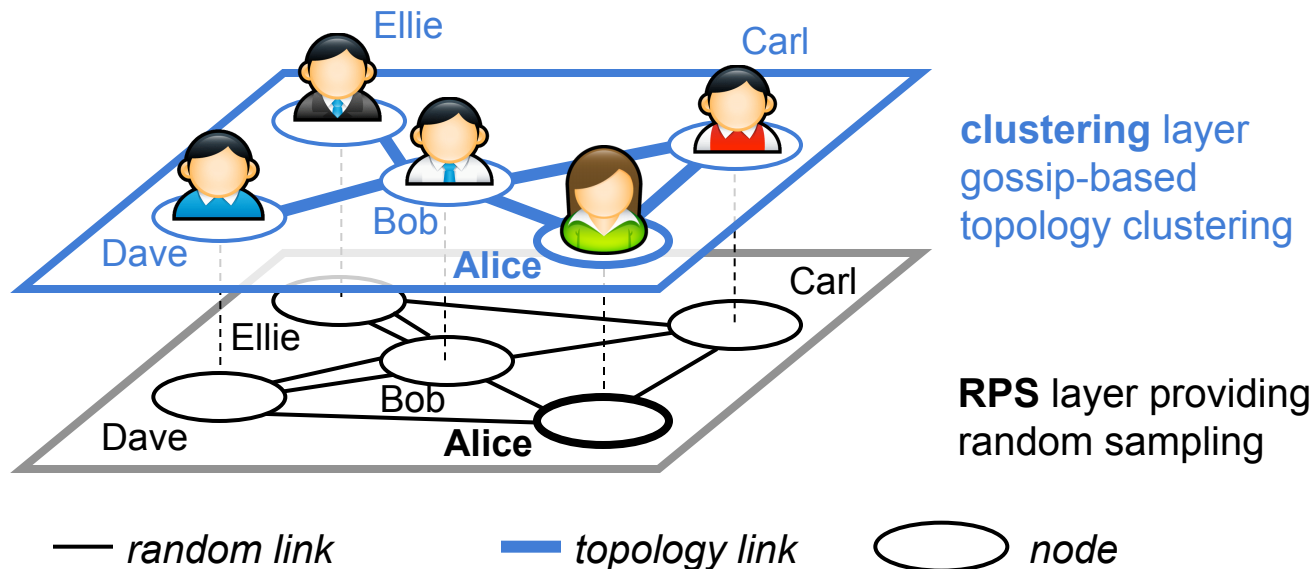
*Builds on T-Man [JMB09], Vicinity [VS05]*

- Each users = 1 node = a profile + a set of neighbours
- Metrics: “similarity” measure between nodes
- Goal: find  $n$  closest neighbours (decentralised)



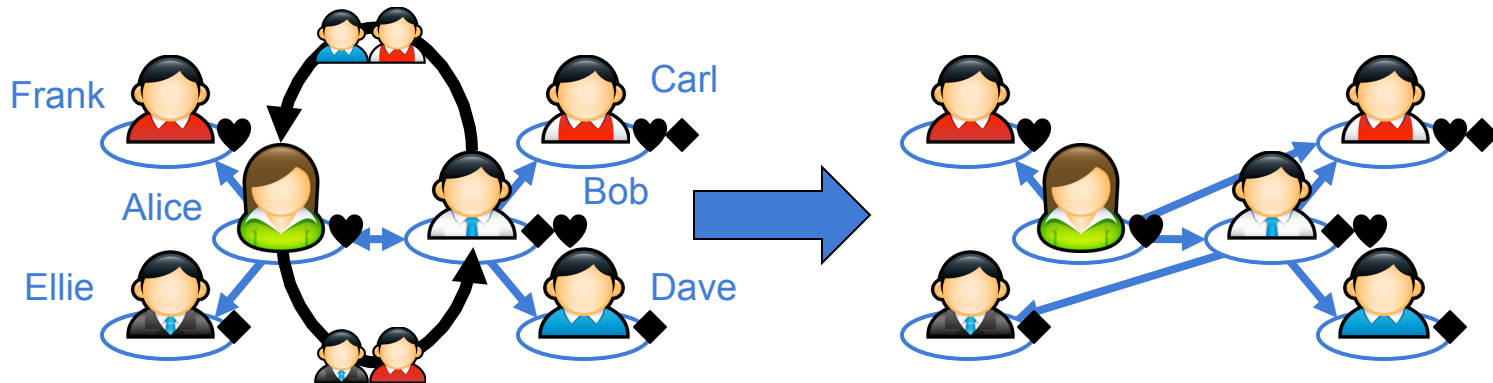
# Gossip-based Social Networks

- Uses randomising layer (RPS, Cyclon)
- Periodically: each node  $n$ 
  - **merges** RPS neighbours with clustering neighbours
  - **sorts** nodes in list based on similarity
  - **keeps**  $k$  closest neighbours



# Example: Overlap Metrics

- User profile: list of shapes
  - ➔ Alice loves heart, Bob prefers diamonds over hearts

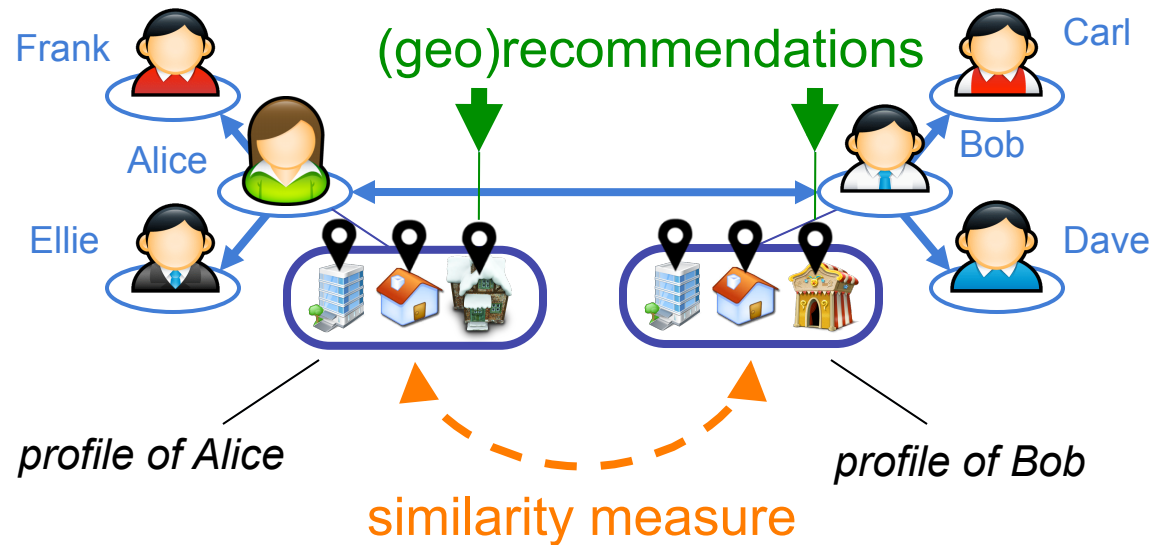


**1** exchange of neighbors lists

**2** neighborhood optimization

# Exploitation: recommendation

- E.g. (geo)recommendations



- Other uses: search, query extension, news filtering

**Can we build a (simple, elegant)  
language to rapidly experiment with  
these self-stabilising social networks?**



# Challenges

- Large design space (profile, metrics)
- Recursive behaviours (neighbourhood comp)
- Monolithic approaches / low reuse

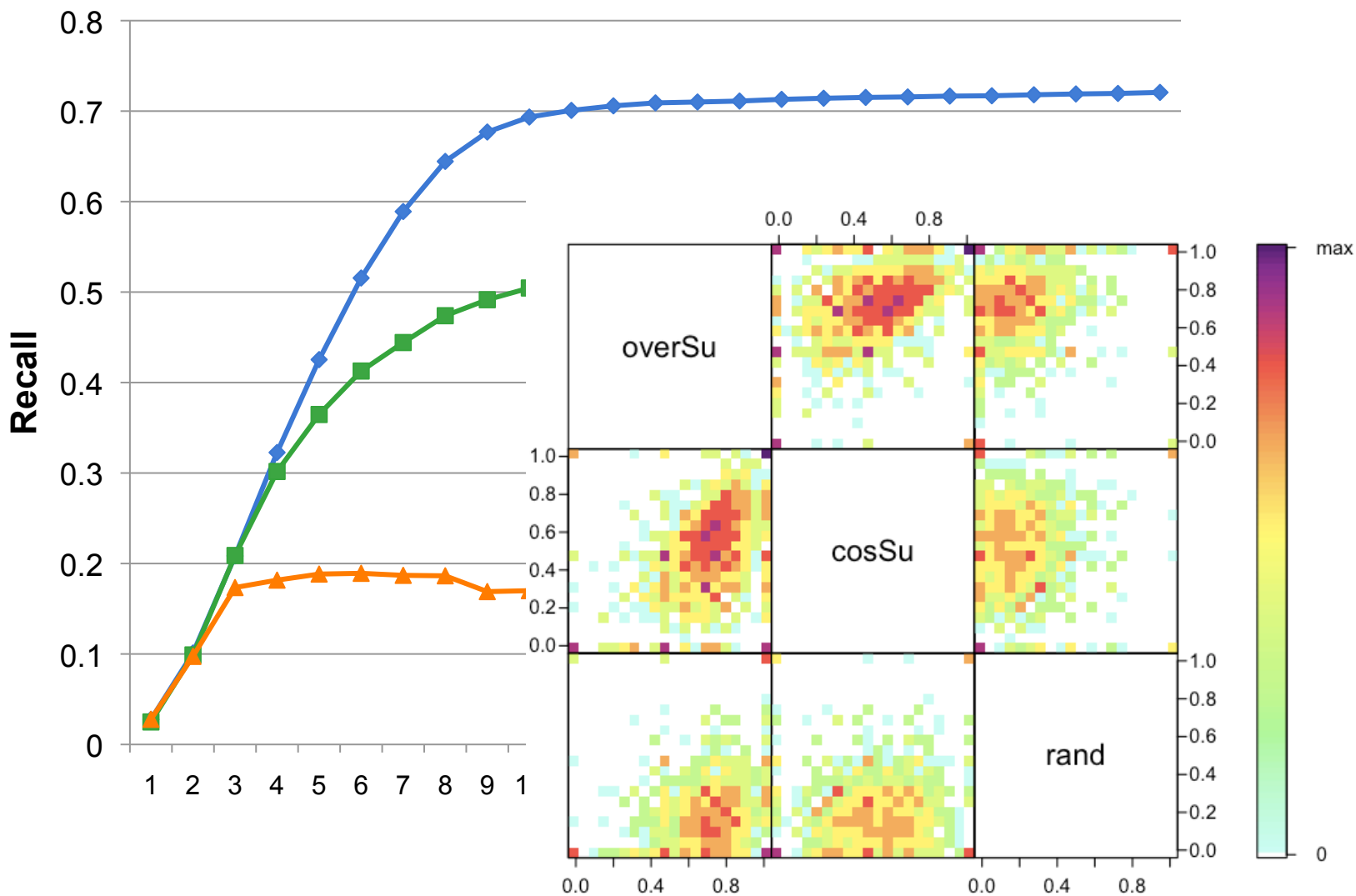
# Our Take: Constellation

- Simple, declarative, (composable)
- Predicting twitter subscriptions using past subs (:s)

```
node User {  
  data :name  
  data :s  
  clusters :cosSu { |u| cos_sim(self->:s, u->:s) }  
  clusters :overSu { |u| overlap(self->:s, u->:s) }  
  clusters :rand { |u| rand }  
}
```

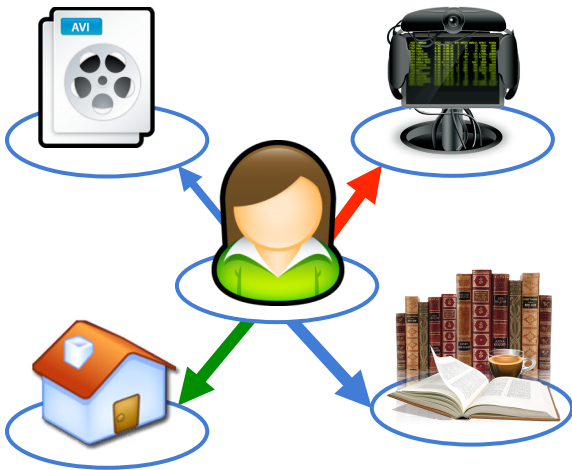
- Ruby-based prototype
- 2446 users crawled from twitter

# Results



# Tackling more: Heterogeneity

- Different types of nodes, asymmetric similarities (\*)

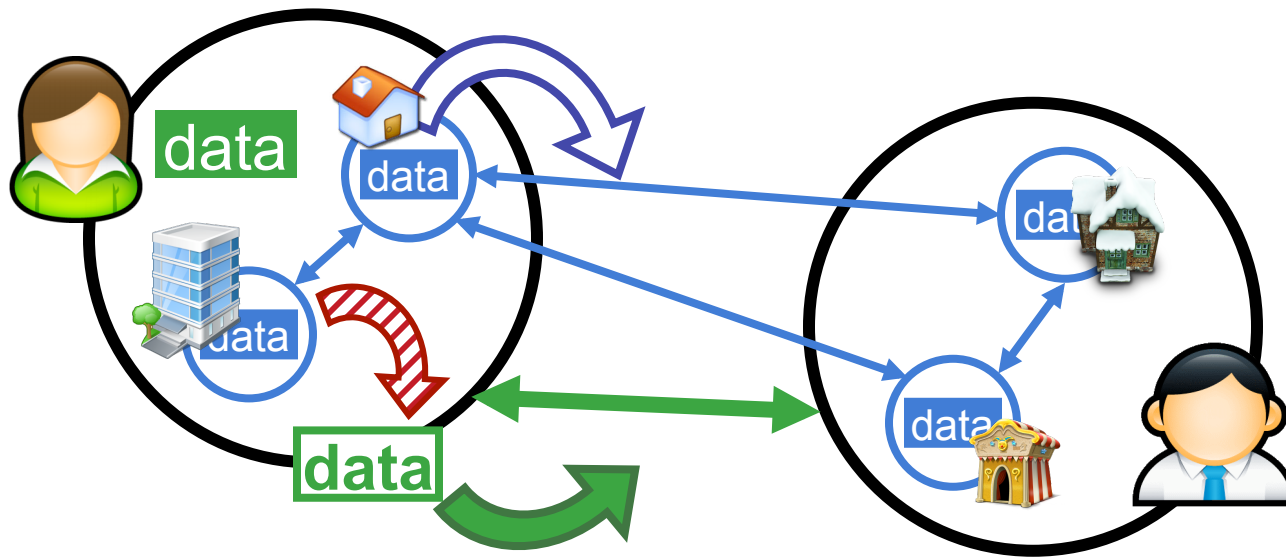


```
node Website { data :k }
node User    {
  data :i
  ..
  clusters :sites with Website {
    |s| cos_sim( this->:i, s->:k ) }
  ..
}
```

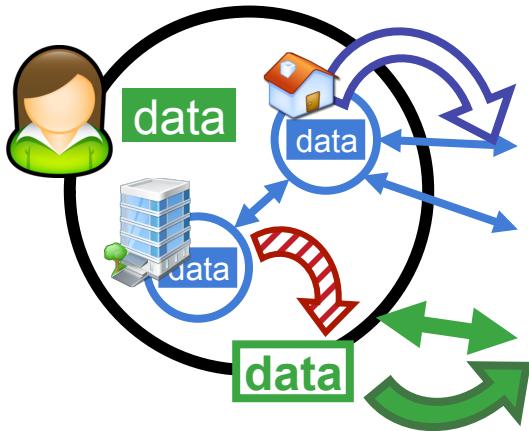
(\*) under development

# Tackling more: Recursion

- Recursive clustering [BBGKL10, BFGKL10]
  - containment relationships
  - clustering applied to containee
  - neighbourhood computation



# Tackling more: Recursion



```
node Checkin {
  data :t
  clusters :similarSites { |c|
    overlap(this->:t,c->:t) }
}
node User {
  contains Checkin :c
  clusters :similarUsers { |u|
    overlap(this->:c->:t,u->:c->:t) }
  set_scope :similarUsers for :c
}
```

( under development)

# Summary

- Constellation: DSL for self-organising social networks
  - **Rapid** experimentation
  - **Composable**, recursive
  - Still under development
- Benefits
  - Not much in terms of language research?
  - Dist. Sys. : organise **design space**, help **exploration**
  - Not limited to social networks (embryomorphic eng.)



[Doursat, R. (2008)]

Doursat, R. (2008). Programmable architectures that are complex and self-organized: from morphogenesis to engineering. (ALIFE XI)

# Some Refs: Gossip

- [BHO99] Birman, K. P., Hayden, M., Ozkasap, O., Xiao, Z., Budiu, M., and Minsky, Y. (1999). **Bimodal multicast**. ACM Trans. Comput. Syst., 17:41–88.
- [GRB01] Gupta, I., Renesse, R. v., and Birman, K. P. (2001). **Scalable fault-tolerant aggregation in large process groups**. DSN '01, pages 433– 442
- [KMG03] Kermarrec A.-M., Massoulie L., Ganesh, A.J., **Reliable Probabilistic Communication in Large-Scale Information Dissemination Systems**, *IEEE Transactions on Parallel and Distributed Systems*, March 2003, (14:3)
- [JGK04] Jelasity, M., Guerraoui, R., Kermarrec, A.-M., and van Steen, M. (2004). **The peer sampling service: experimental evaluation of unstructured gossip-based implementations**. Middleware '04, pages 79–98, New York, NY, USA. Springer- Verlag New York, Inc.



# Some Refs: Gossip

- [HHL06] Haas, Z. J., Halpern, J. Y., and Li, L. (2006). **Gossip-based ad hoc routing**. *IEEE/ACM Trans. Netw.*, 14:479–491.
- [JK06] Jelasity, M. and Kermarrec, A.-M. (2006). **Ordered slicing of very large-scale overlay networks**. In *Proceedings of the Sixth IEEE International Conference on Peer-to-Peer Computing*, pages 117–124, Cambridge, United Kingdom. IEEE Computer Society.
- [JMB09] Mark Jelasity, Alberto Montresor, and Ozalp Babaoglu. 2009. **T-Man: Gossip-based fast overlay topology construction**. *Comput. Netw.* 53, 13 (August 2009), 2321-2339.
- [LGKVB06] Erwan Le Merrer, Vincent Gramoli, Anne-Marie Kermarrec, Aline C. Viana, and Marin Bertier. 2006. **Energy aware self-organizing density management in wireless sensor networks**. *1st Int. workshop on Decentralized resource sharing in mobile computing and networking (MobiShare '06)*. ACM

# Some Refs: Gossip + SocNet

- [BBGKL10] X. Bai, M. Bertier, R. Guerraoui, A.-M. Kermarrec, and V. Leroy. **Gossiping personalized queries**. In EDBT'10.
- [BFGKL10] M. Bertier, D. Frey, R. Guerraoui, A.-M. Kermarrec, and V. Leroy. **The Gossple anonymous social network**. In Middleware'2010.

# Some Refs: Frameworks

- [LTBBK11] Lin S., Taiani F., Bertier M., Blair G. S., Kermarrec A.-M. (2011). **Transparent componentisation: high-level (re)configurable programming for evolving distributed systems**. ACM SAC '11, pp. 203–208
- [PB2010] Lonnie Princehouse and Ken Birman. **Code-partitioning gossip**. SIGOPS Oper. Syst. Rev., 43:40–44, January 2010.
- [LTB08] Lin, S., Taiani, F., and Blair, G. S. (2008). **Facilitating gossip programming with the gossipkit framework**, DAIS'08, pages 238–252
- [KS07] Kermarrec, A.-M. and van Steen, M. (2007). **Gossiping in distributed systems**. SIGOPS Oper. Syst. Rev., 41:2–7
- [EFL07] Eugster, P., Felber, P., and Le Fessant, F. (2007). **The "art" of programming gossip-based systems**. SIGOPS Oper. Syst. Rev., 41:37–42.

# T-Man in Action

- (taken from [JMB09])

Result → structured overlay

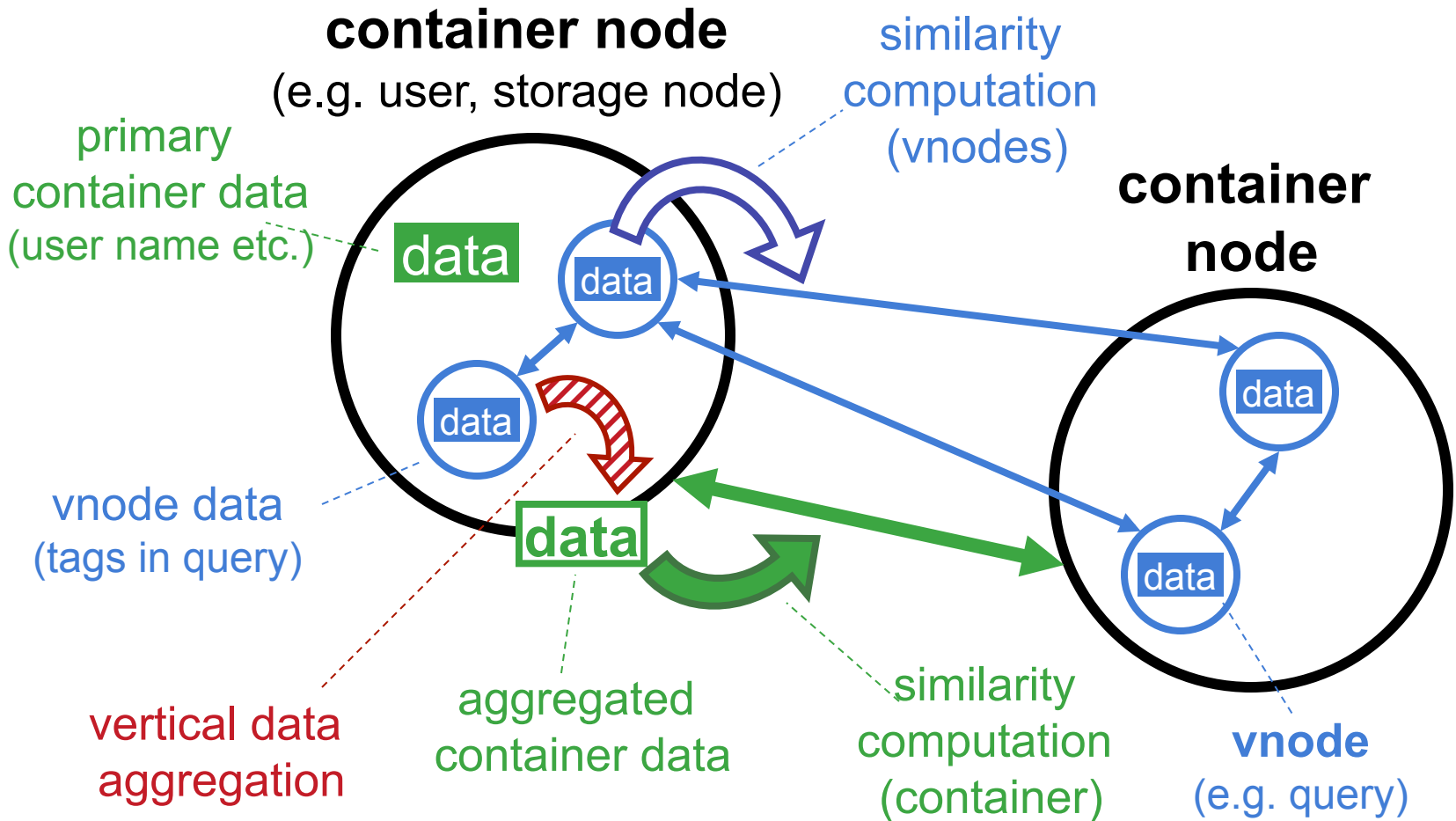
Highly resilient against churn, partition (RPS)

after 2 cycles

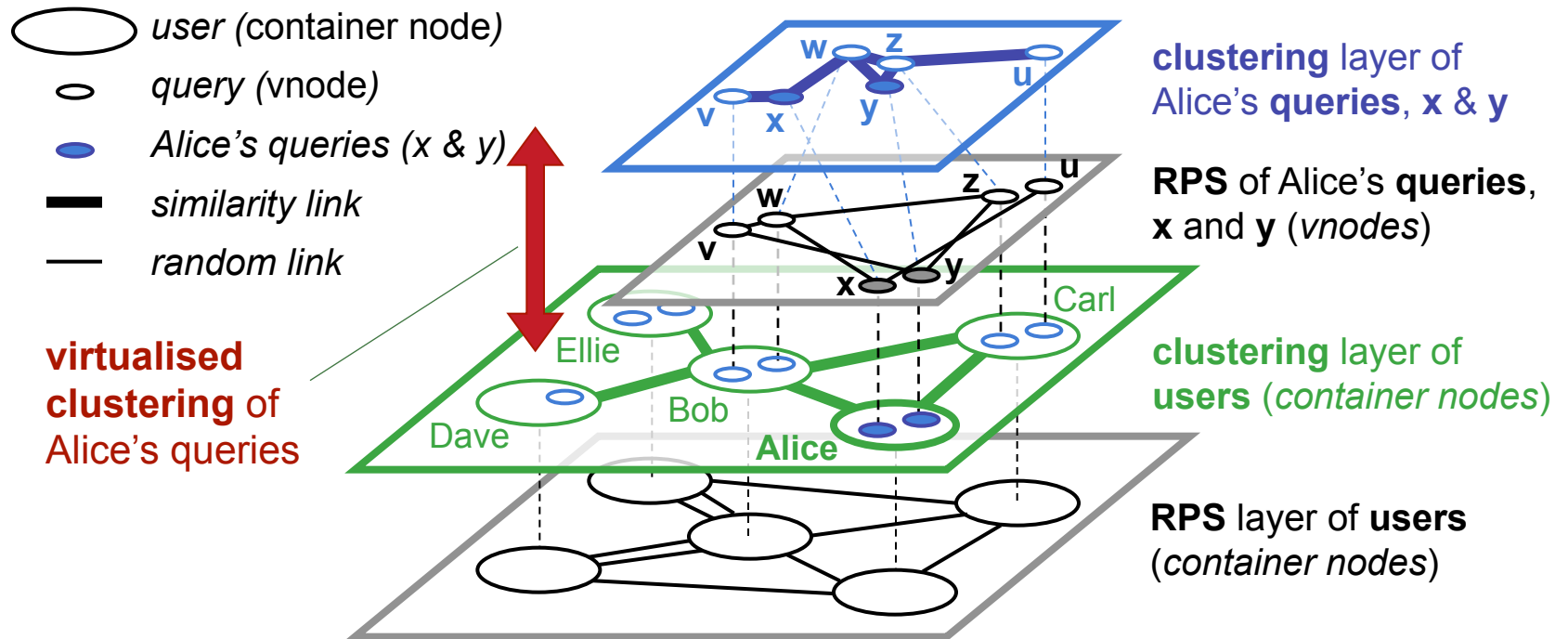
Fast convergence (Swap)

after 7 cycles

# Capturing Neighbourhood Comp



# Capturing neighbourhood Comp



# Self-organising networks

- To speed convergence each node  $n$ 
  - picks one neighbour  $i$  in clustering layer
  - sends own list of neighbours, receives that of  $i$
  - $i$  and  $n$  merge, **sort**, keep  $k$  closest neighbours

