



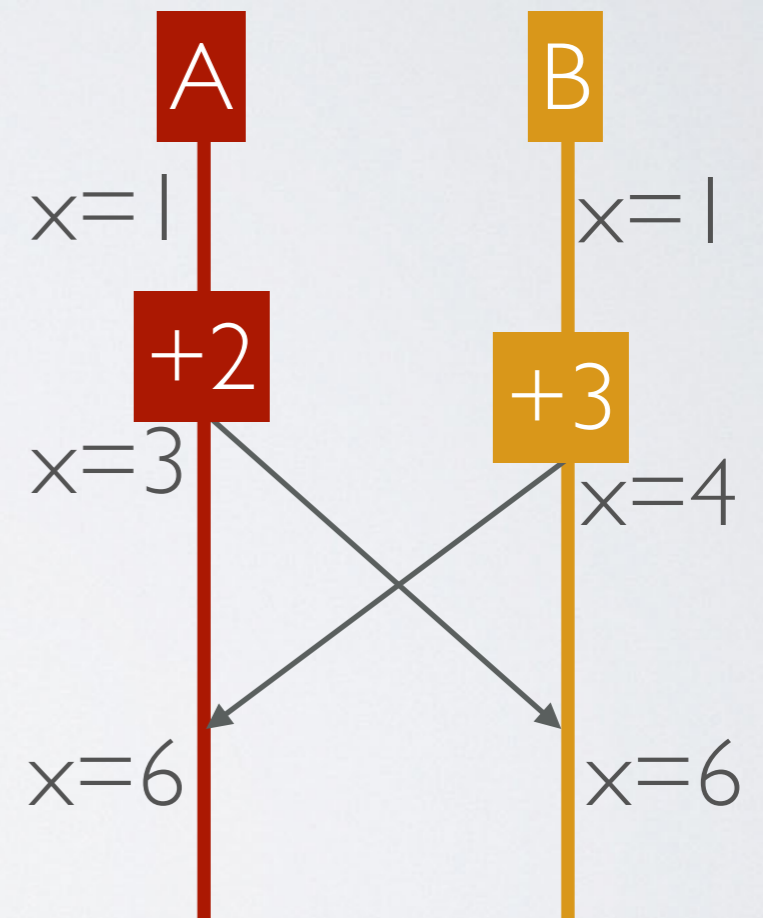
# DYNAMIC CONSISTENCY

João Neto  
Universitat Politècnica de Catalunya

Huge thanks to:  
João Barreto (IST, U. Lisboa)  
Luís Veiga (IST, U. Lisboa)  
Emmanouil Dimogerontakis (UPC Barcelona & IST, U.Lisboa)  
Leandro Navarro (UPC Barcelona)

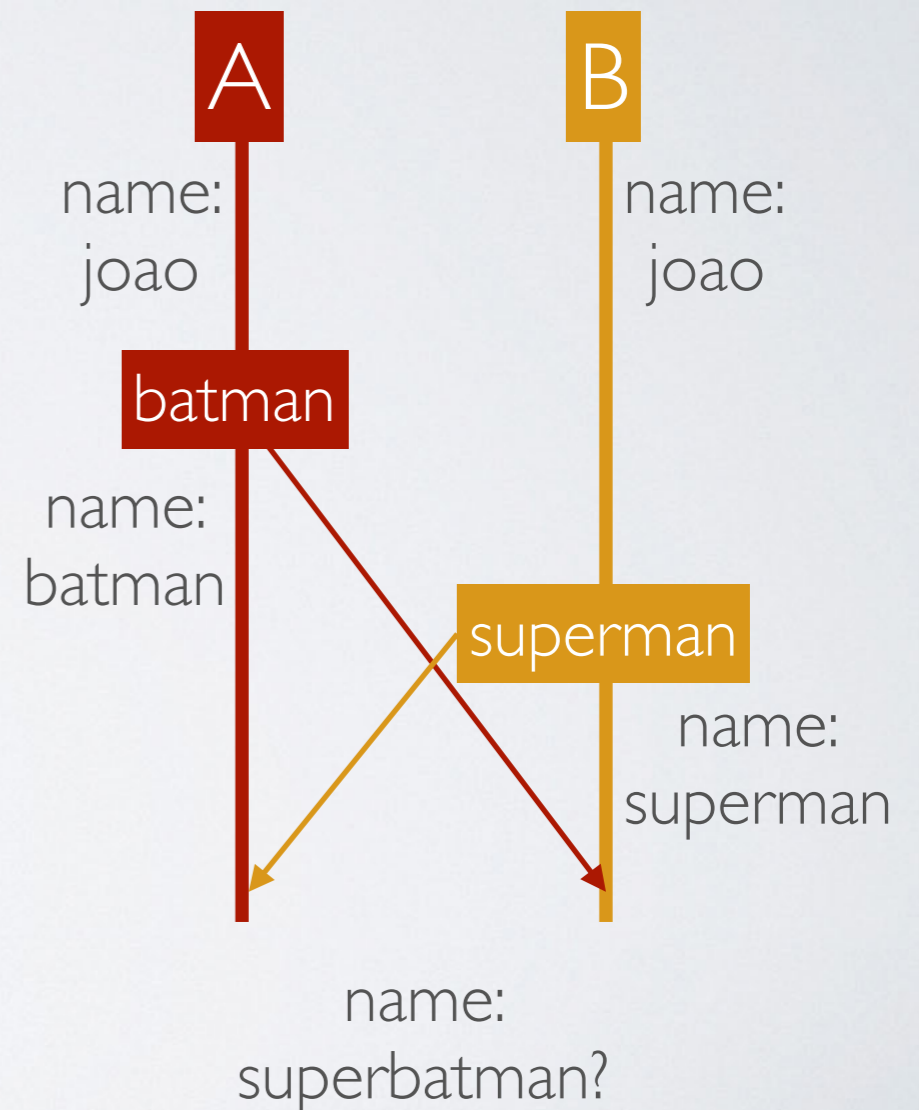
# OPTIMISTIC REPLICATION

- All replicas can act as a 'master' and commit without synchronization
- Available under network partitions, but weaker consistency — conflicts, anomalies.
- Correctly handling concurrent updates: CRDTs



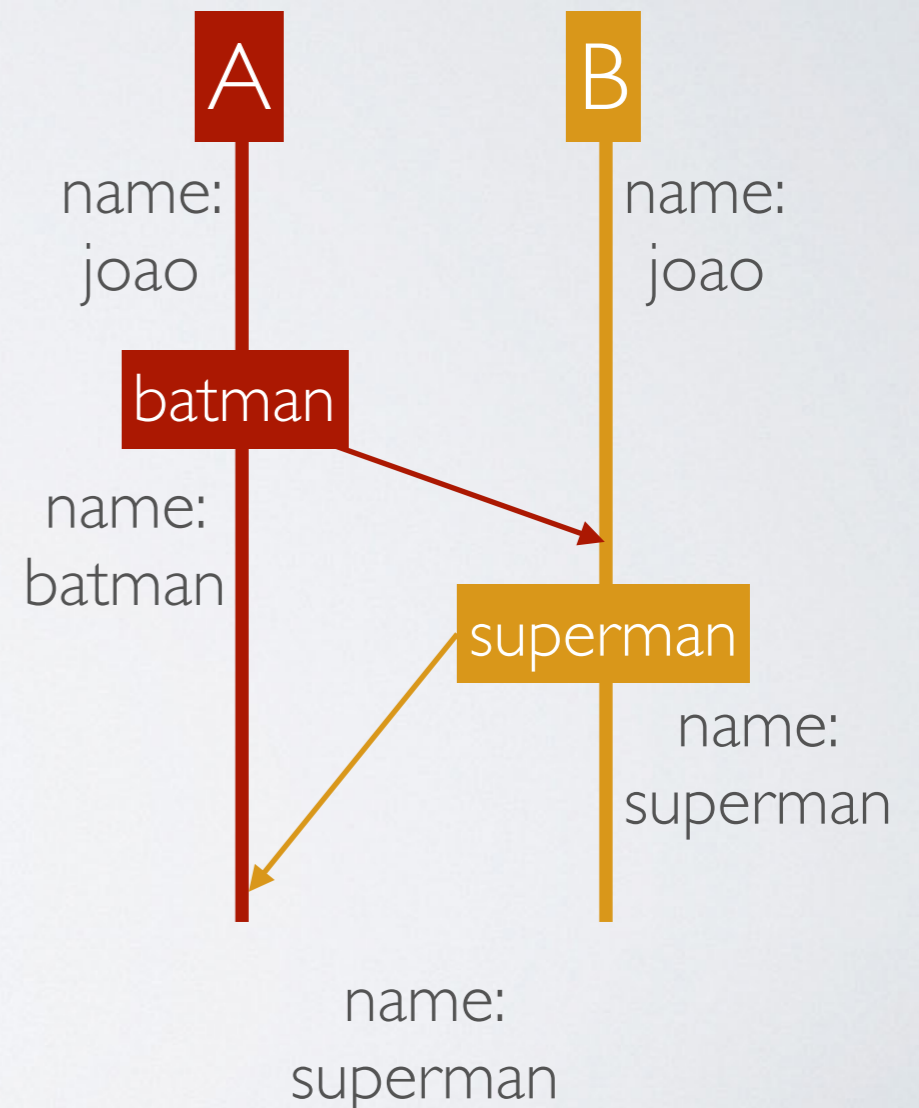
# OPTIMISTIC REPLICATION

- Tradeoff in Weak Consistency:  
**Laziness VS Anomalies**
  - Register:  
assign(batman) || assign(superman)
  - Set:  
add(item) || remove(item)
  - Graph:  
addEdge(v1,v2) || removeVertex(v1)
  - ...



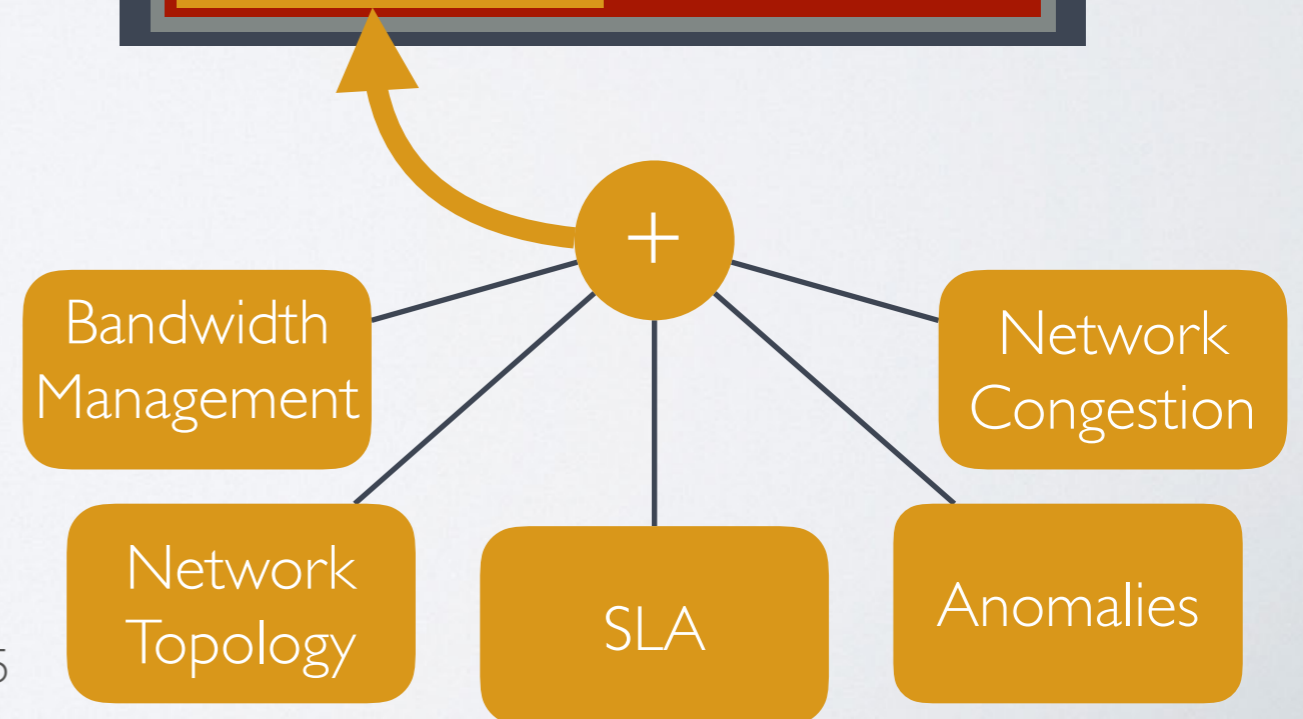
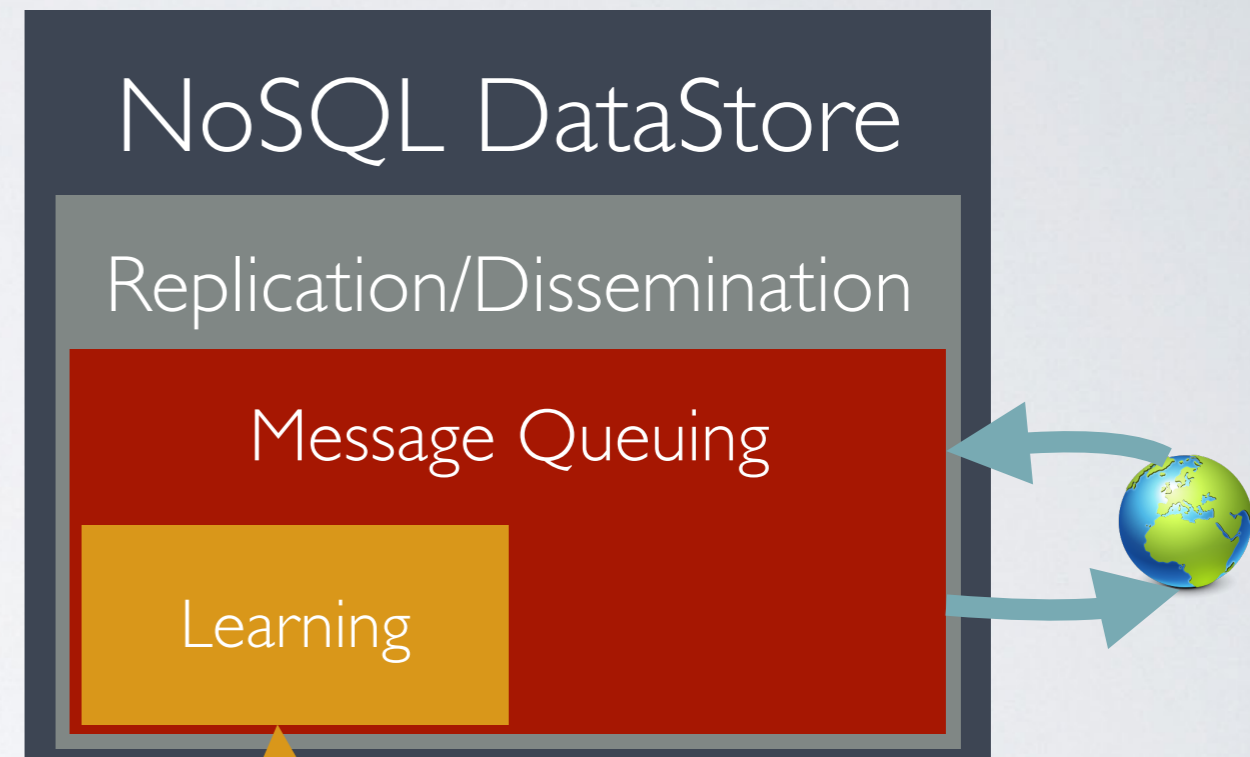
# OPTIMISTIC REPLICATION

- Tradeoff in Weak Consistency: Laziness VS **Anomalies**
- Not all updates are created equal — **expedite important updates.**
- **Upgrade/Downgrade Consistency** per-operation based on available resources



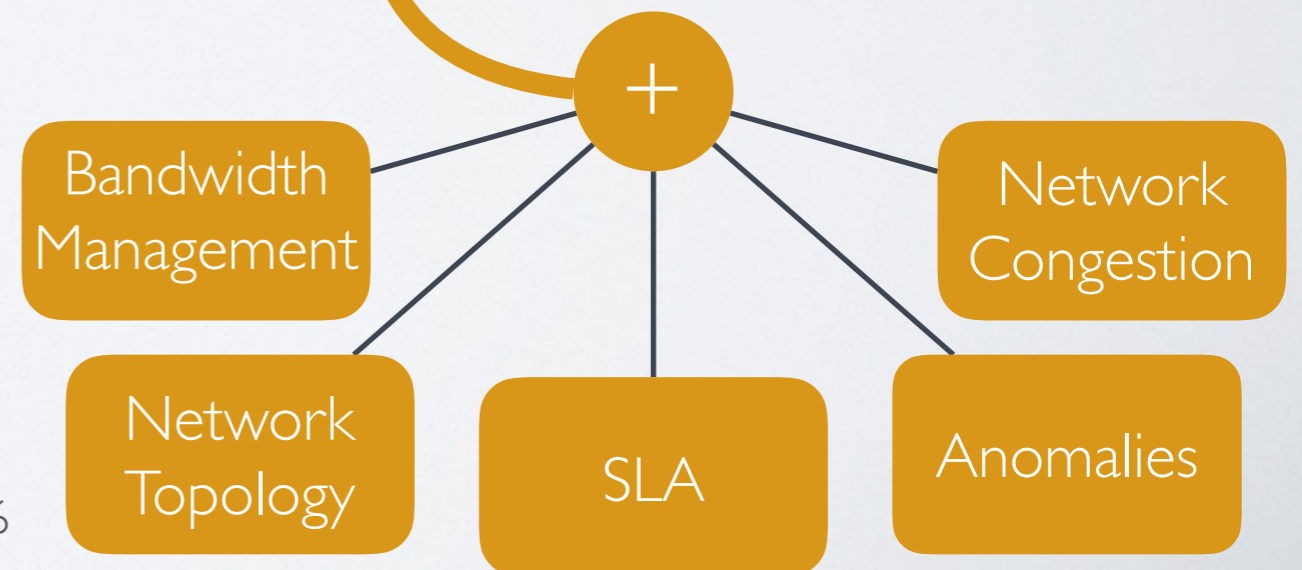
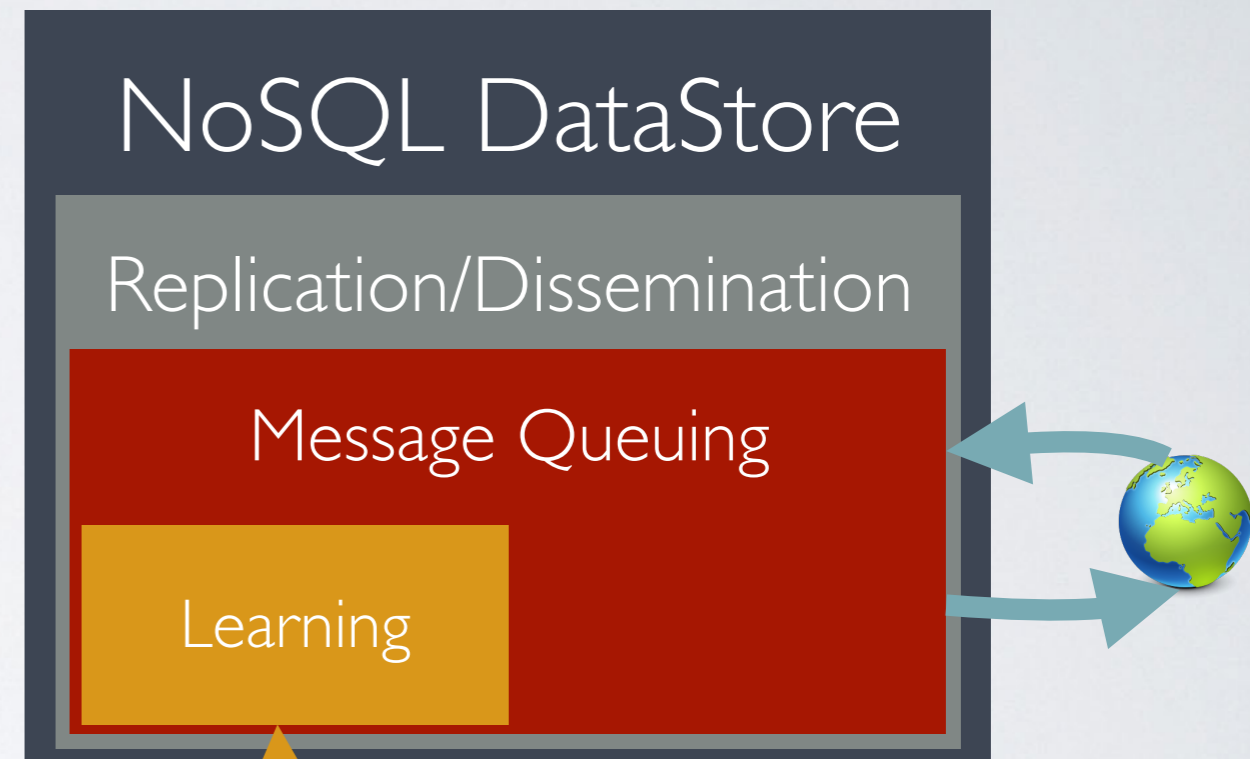
# PROPOSED APPROACH

- **Dynamic Consistency:**  
Consistency can be fluid
- **Message Queuing and Reordering:**  
Not all updates are created equal
- **Self-Adapting:**  
Resources and Demand are Dynamic



# PROPOSED APPROACH

- **Machine Learning**
  - Predict where anomalies might happen, improving user experience
  - Don't forget your SLAs and cost management
  - Disseminate updates according to available network resources

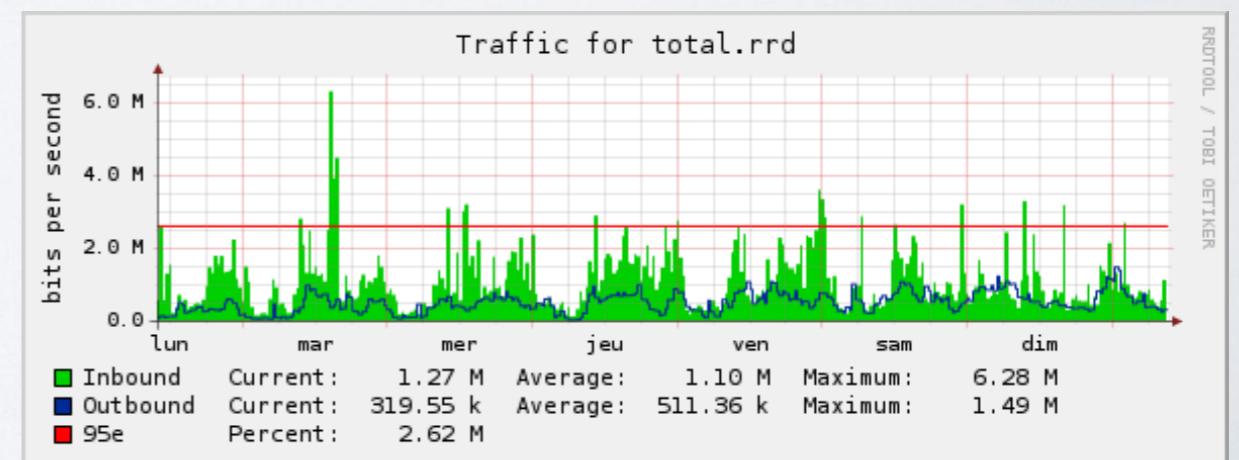
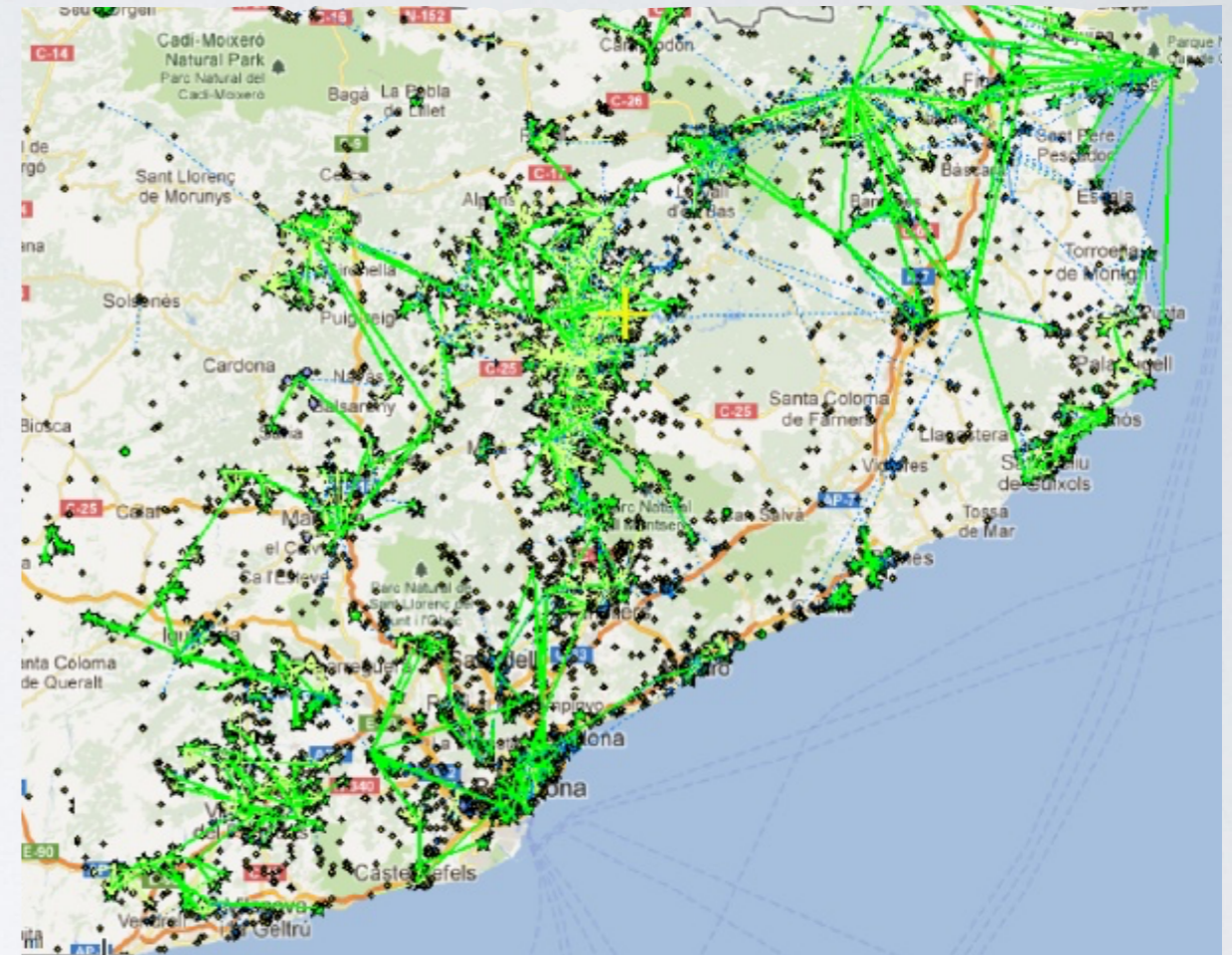


# MAIN CHALLENGES

- How to correctly perform the consistency level switch?
- Are programs easily made to work in both consistencies?
- How to compose optimization objectives?
- How to collect comprehensive enough information without too much overhead?

# EVALUATION

- Development?
- AntidoteDB
- Deployment?
- Traditional multi-DC scenario
- Edge/Fog Networks — [guifi.net](http://guifi.net)





# CONCLUSIONS

- **Anomalies** are caused by **concurrent updates** to the same object, and will always happen on weak consistency models.
- **Smart queuing** reduces the **vulnerability window** where anomalies can happen.
- If you stop treating operations as **black-boxes**, you can make smarter decisions.