Availability in Partition-tolerant Systems with Data Constraints

Mikael Asplund, Simin Nadjm-Tehrani

Real-time Systems Laboratory

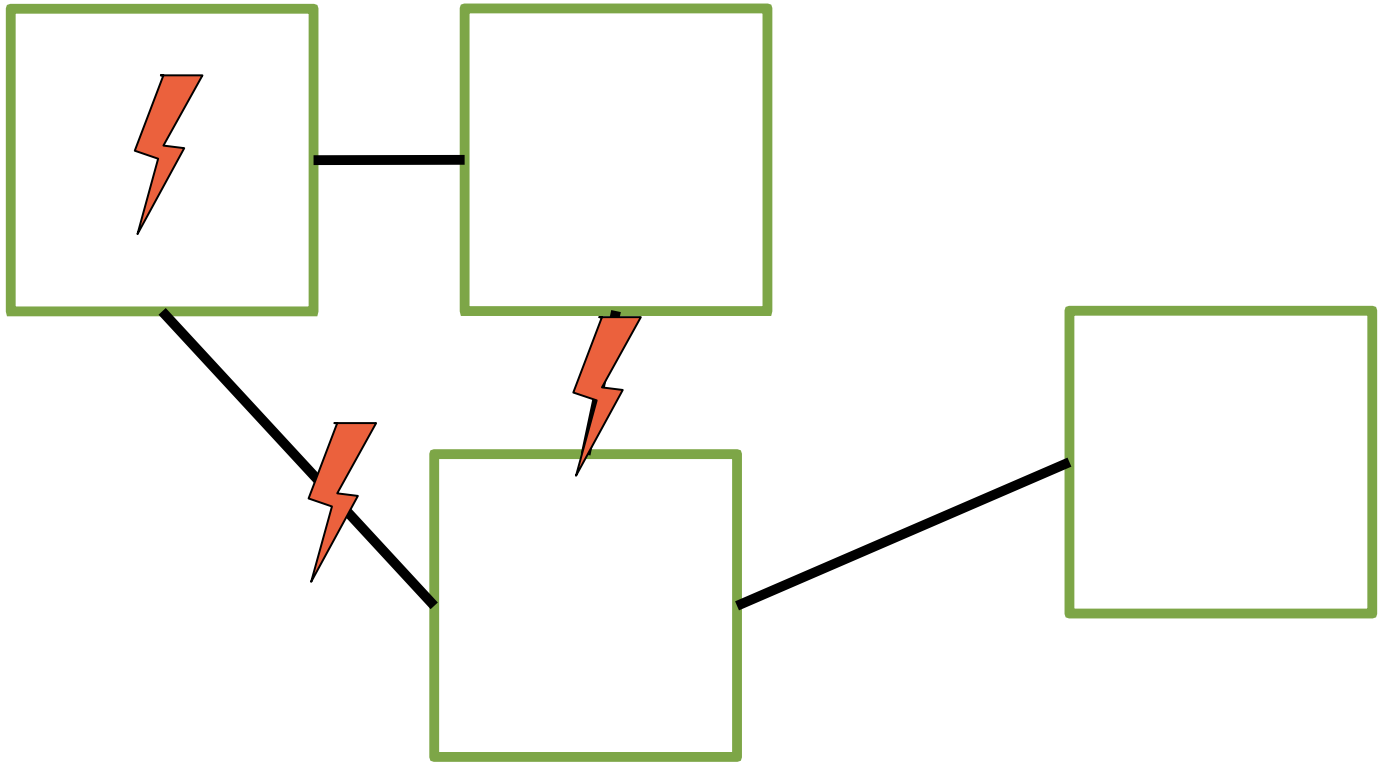
Dept. of Computer & Information Science

www.ida.liu.se/~rtslab

Distributed Object Systems

Fault Model

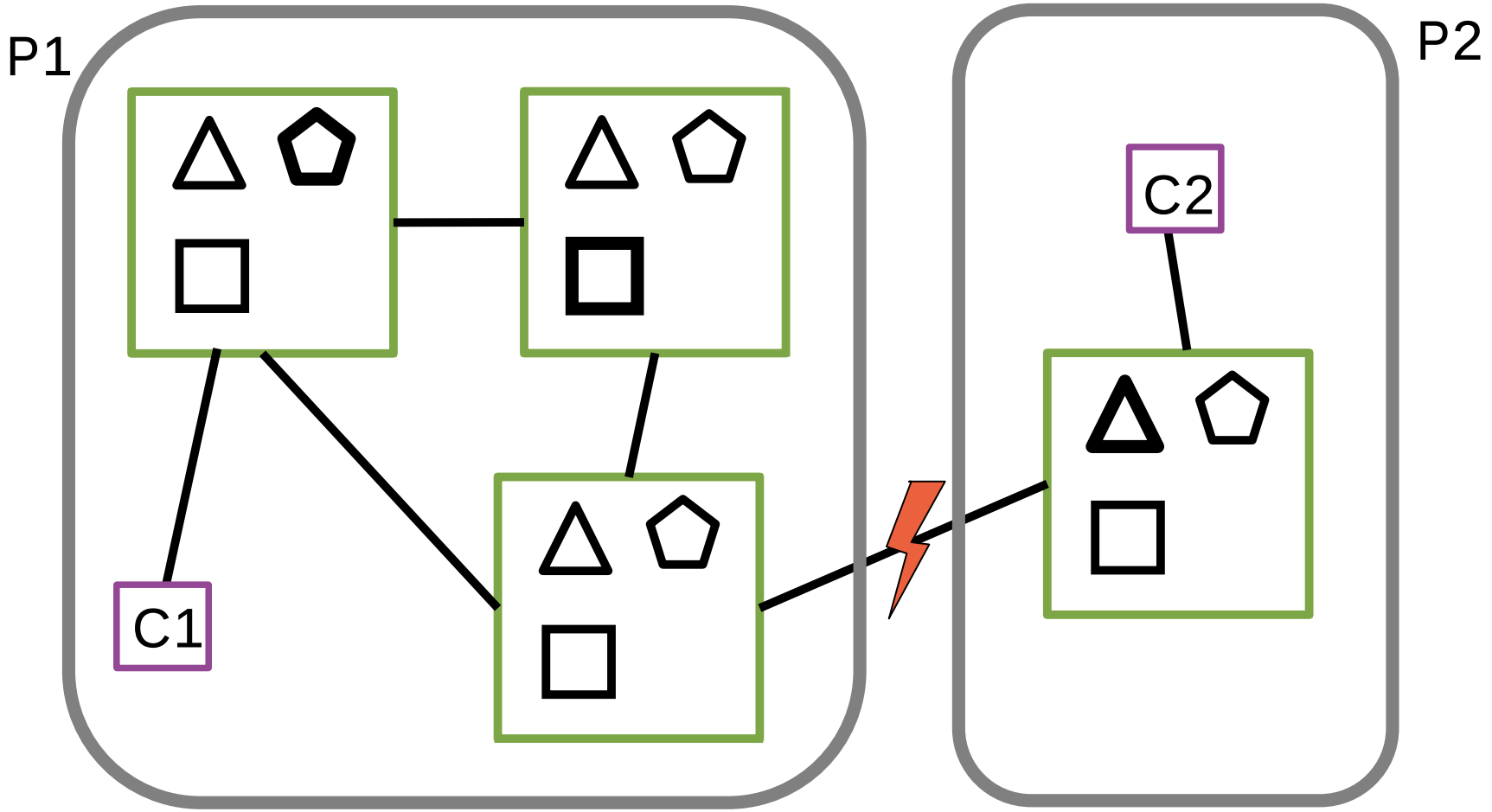
- Faults: Node crash, link failure, network partition



Integrity Constraints

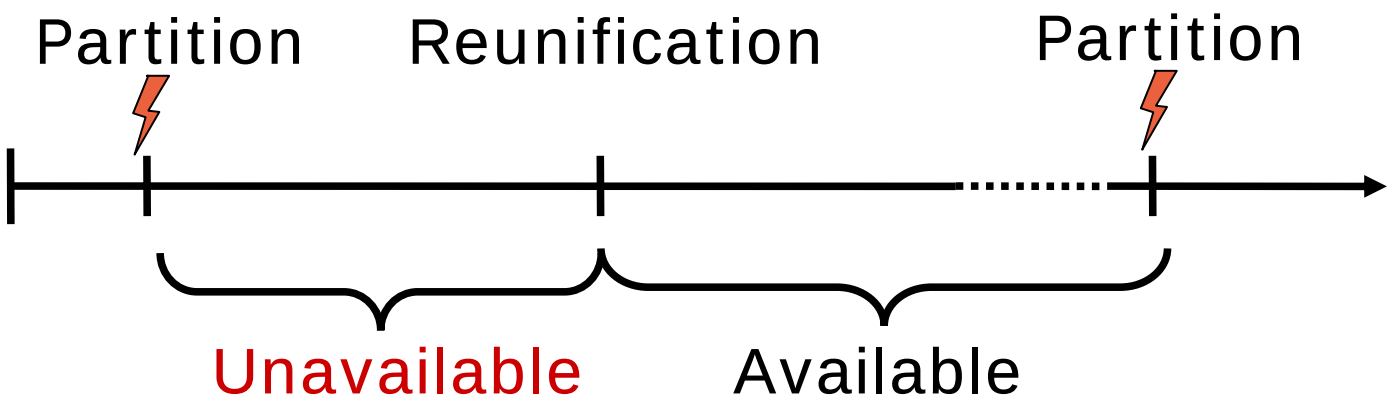
- Bank: you cannot withdraw more money than you have in your account
- Booking a flight ticket: Number of booked tickets must be less than number of seats

Distributed services



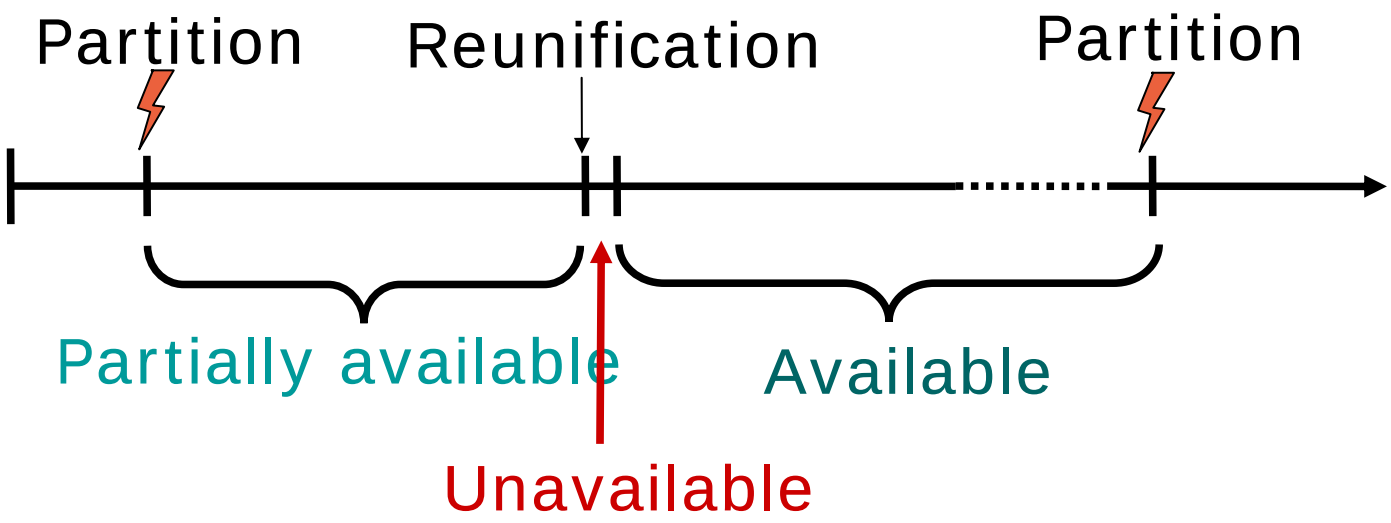
How can we provide availability to C1 and C2?

Pessimistic Approach



Possible solution

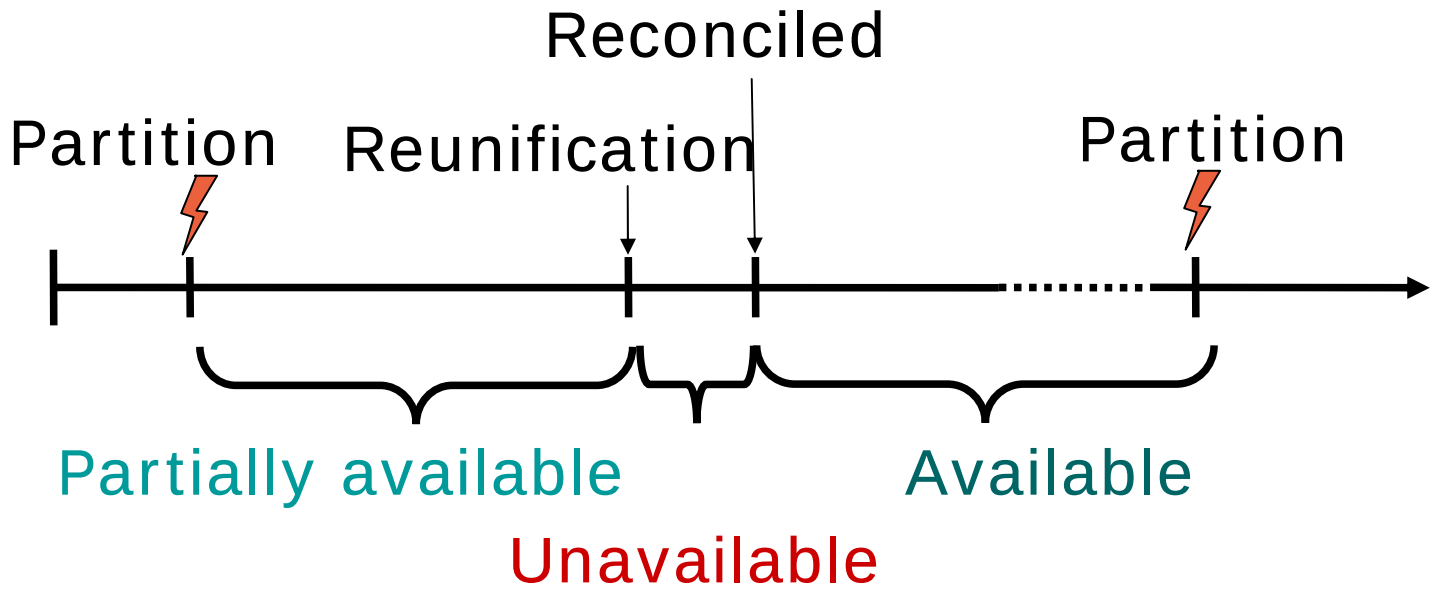
Majority partition:



Alternative: Be Optimistic!

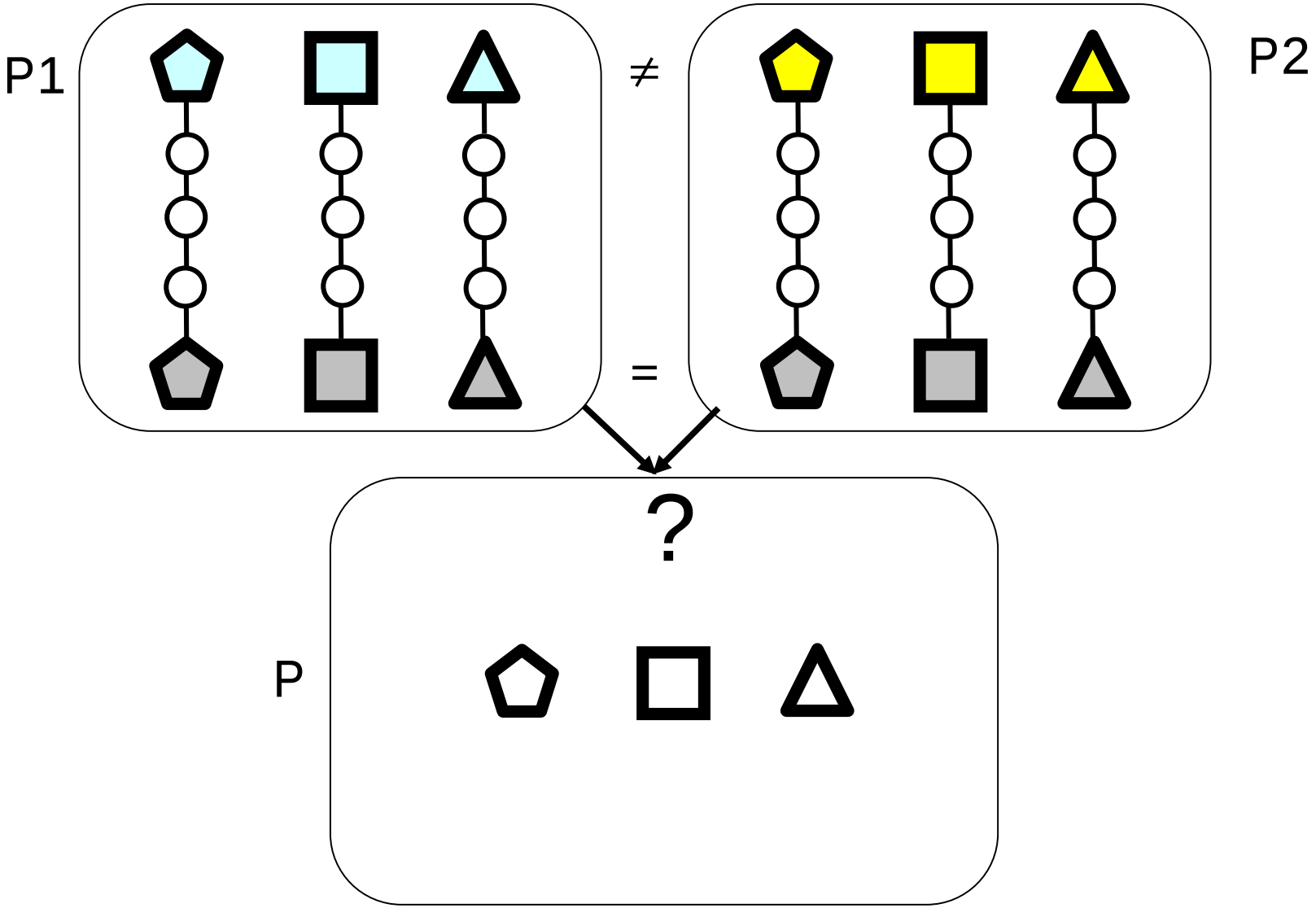
- “Where a pessimistic system waits, an optimistic system speculates” [Saito and Shapiro]
- European project, DeDiSys
  - Three platforms: CORBA, EJB, .Net
- Being optimistic requires fixing afterwards: ***Reconciliation***

Our approach

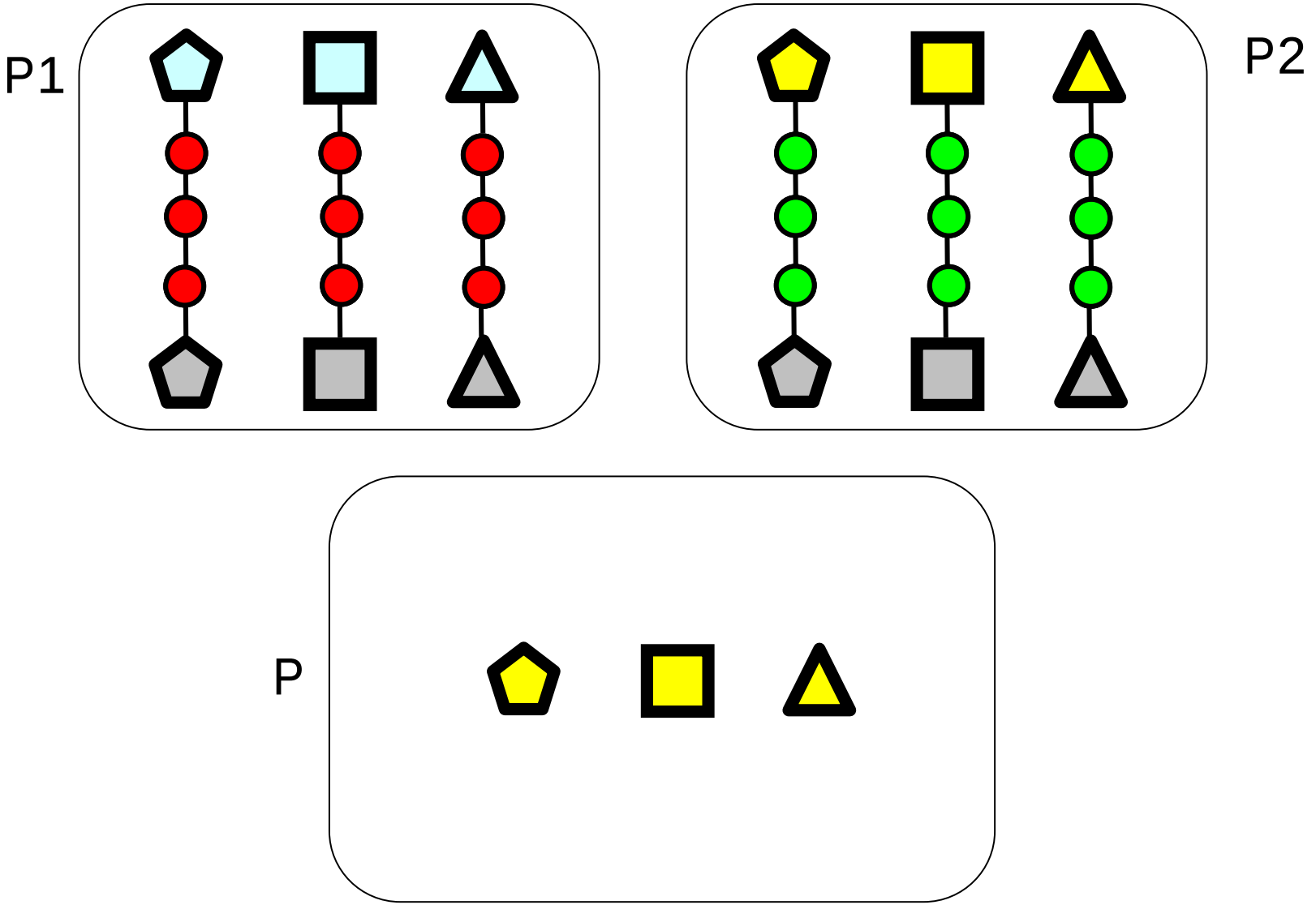


- Primary per partition
- Critical vs. Non-critical constraints
- Stop-the-world during reconciliation

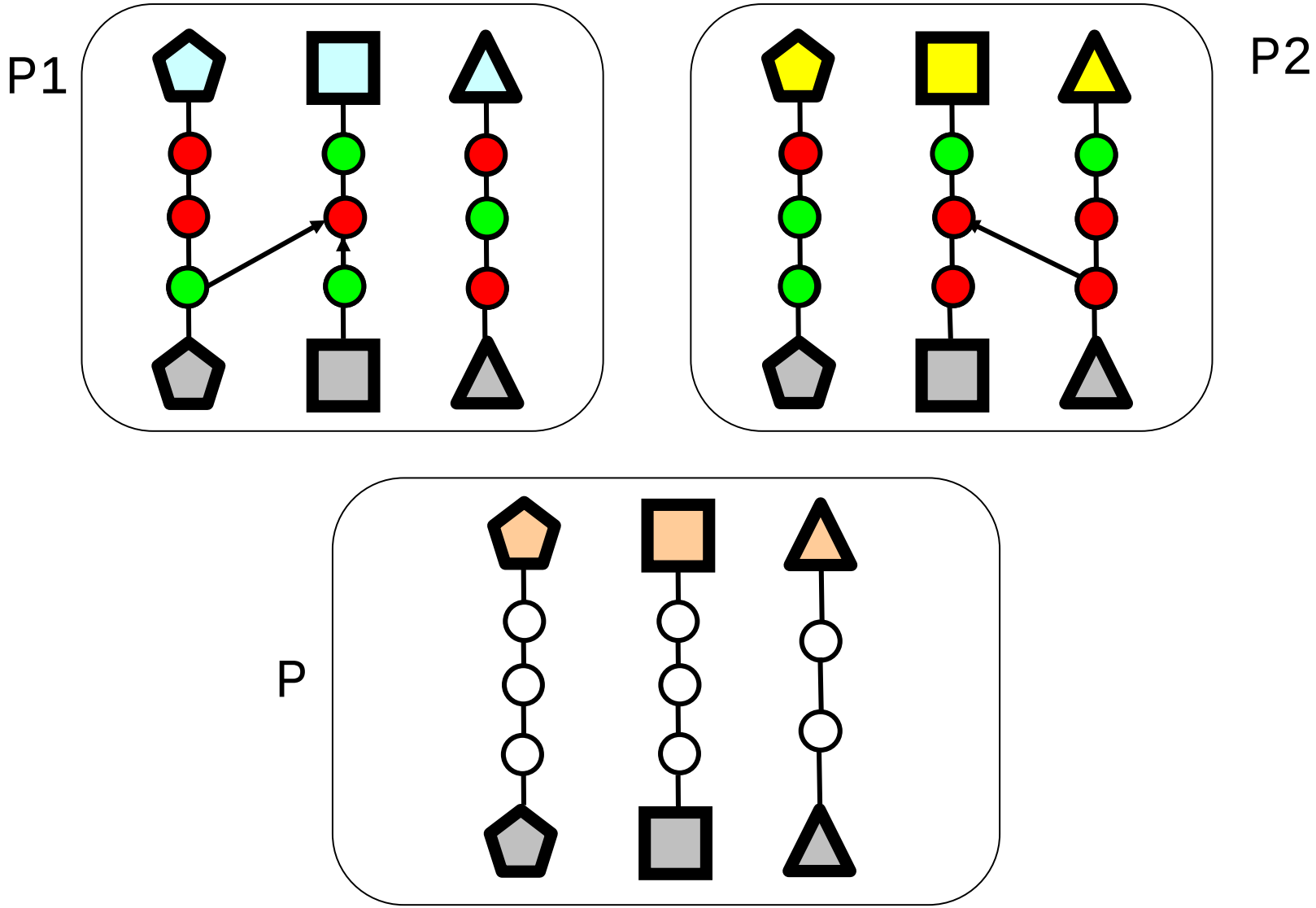
Reconciliation: Goal



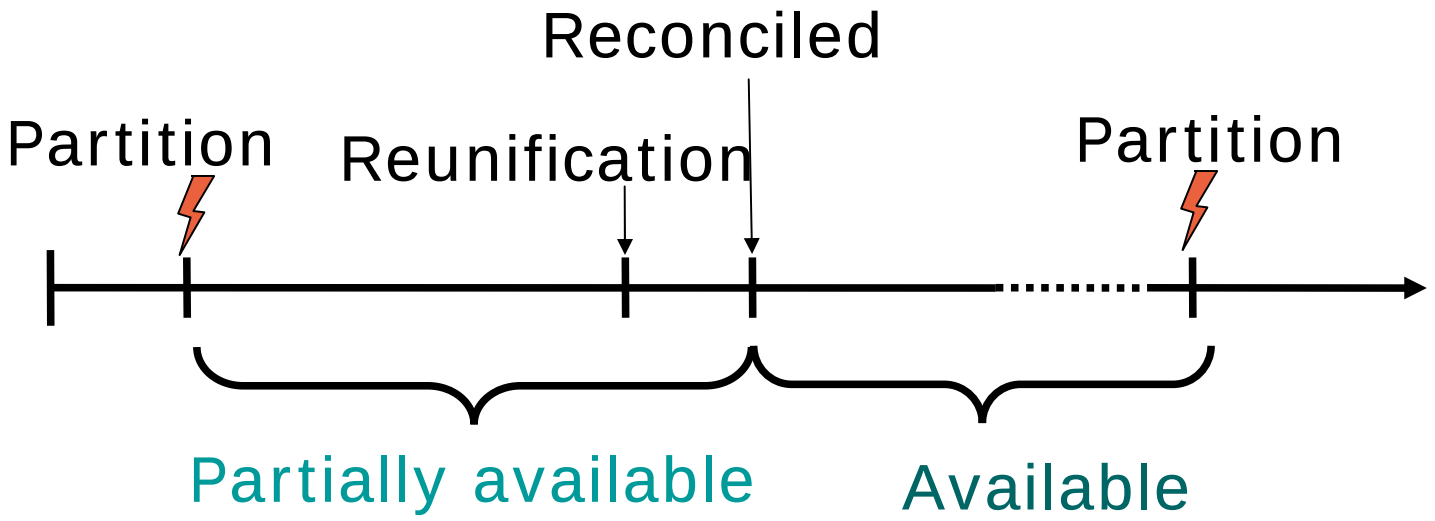
CHOOSE1



MERGE



Continuous Service Protocol

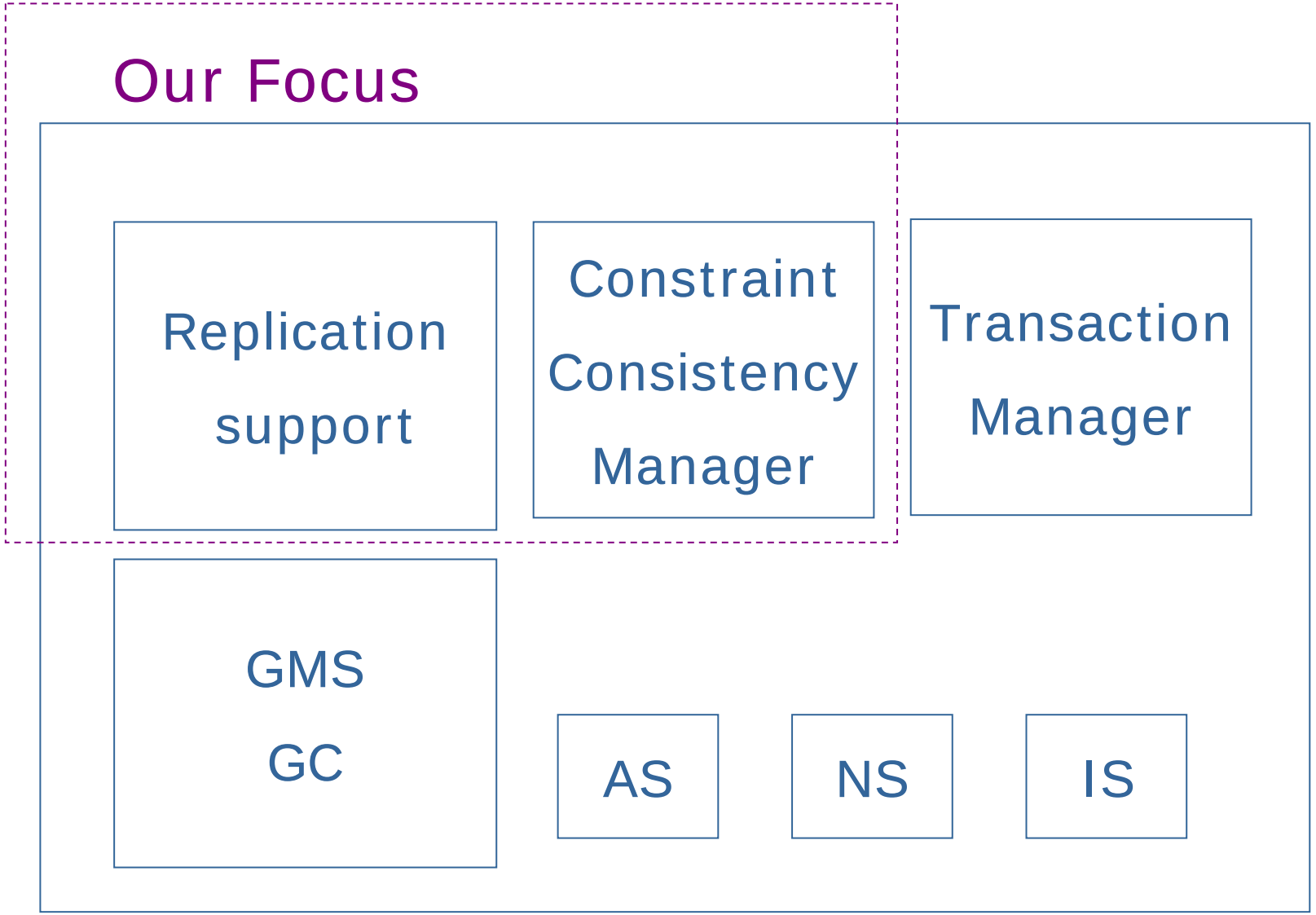


- Available at all times (almost)

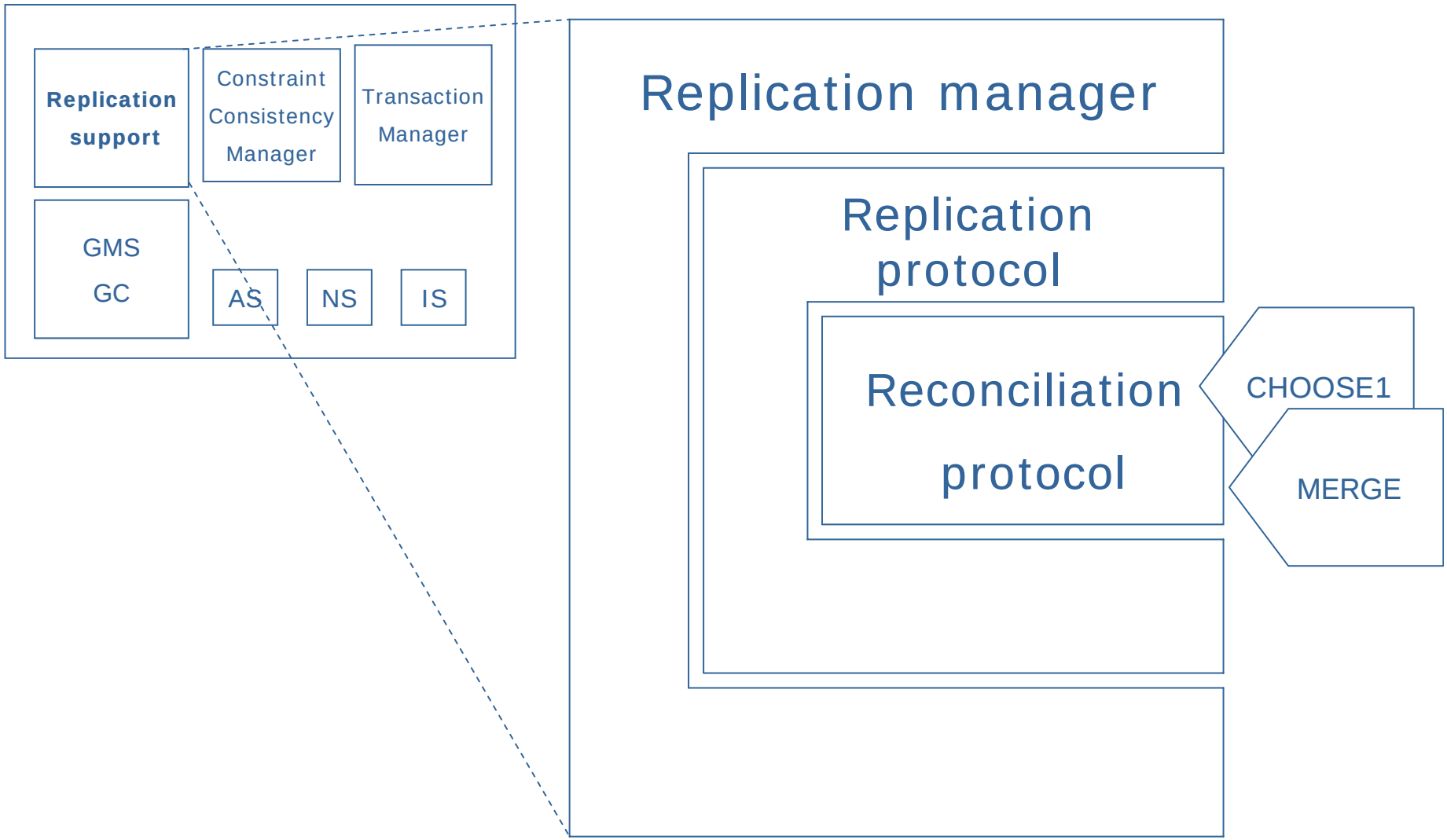
Novelty

- Maintains virtual partitions during reconciliation
- System continues to provisionally accept operations during reconciliation
- Formally described and proved correct using I/O Automata

Middleware support



Reconfigurable support



Measuring Availability

- "The simplest measure of availability is the probability A(t), that the system is *operational* at an arbitrary point in time." [Helal Heddaya and Bhargava]

- What does it mean to be operational?
  - Partial operationality
  - Apparent operationality
  - We need to reflect the penalty of later revoked operations