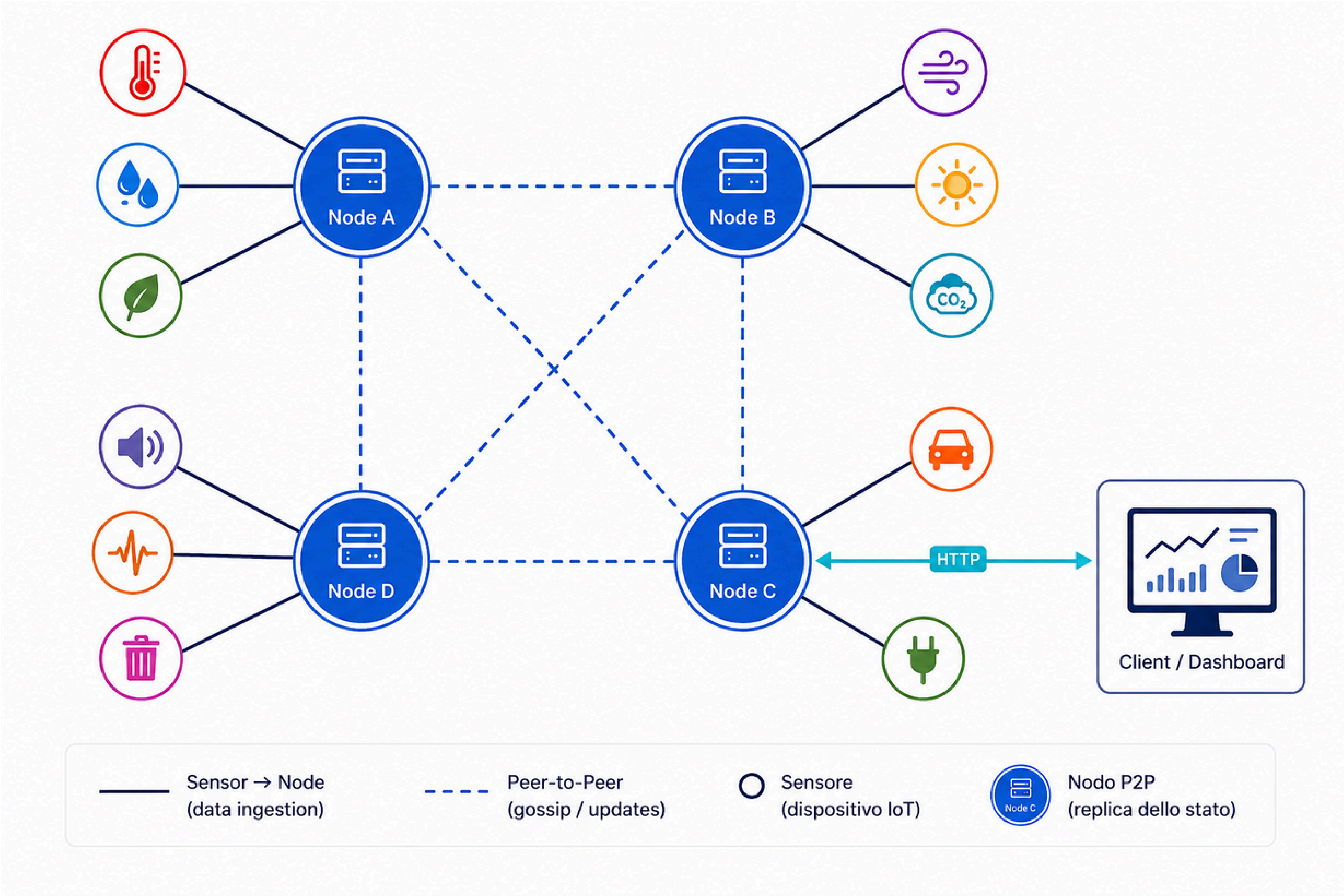
UNIVERSITÀ DI BOLOGNA
CDL-M IN INGEGNERIA E SCIENZE INFORMATICHE, A.A. 2025/2026

P2P DISTRIBUTED SENSOR HUB FOR SMART CITY MONITORING

Alex Santini

Monitoraggio distribuito di sensori in una smart city



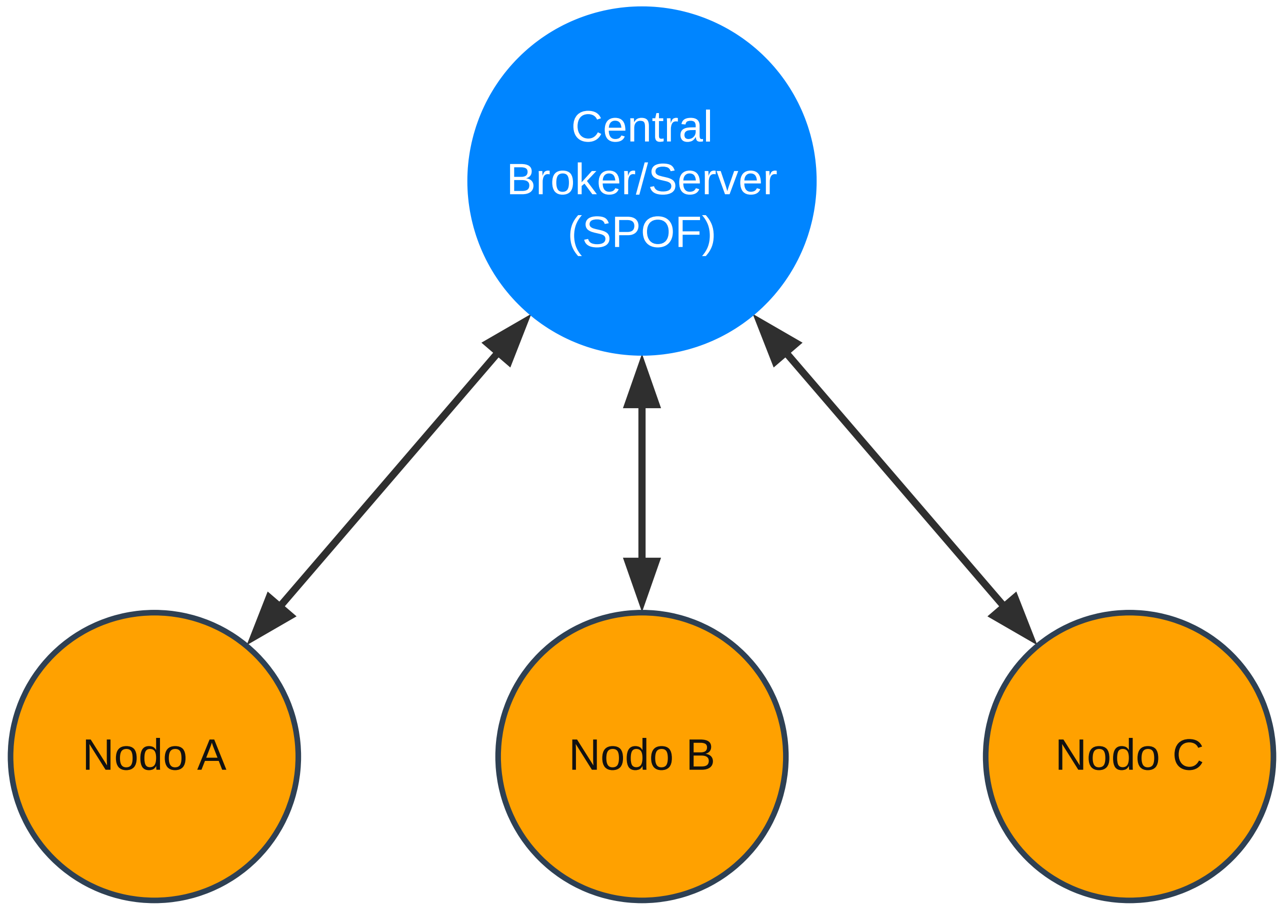
Ogni nodo

- riceve dati da un insieme locale di sensori
- mantiene una replica dello stato globale

- comunicazione gossip p2p
- consistenza eventuale
- client può connettersi ad un nodo qualsiasi

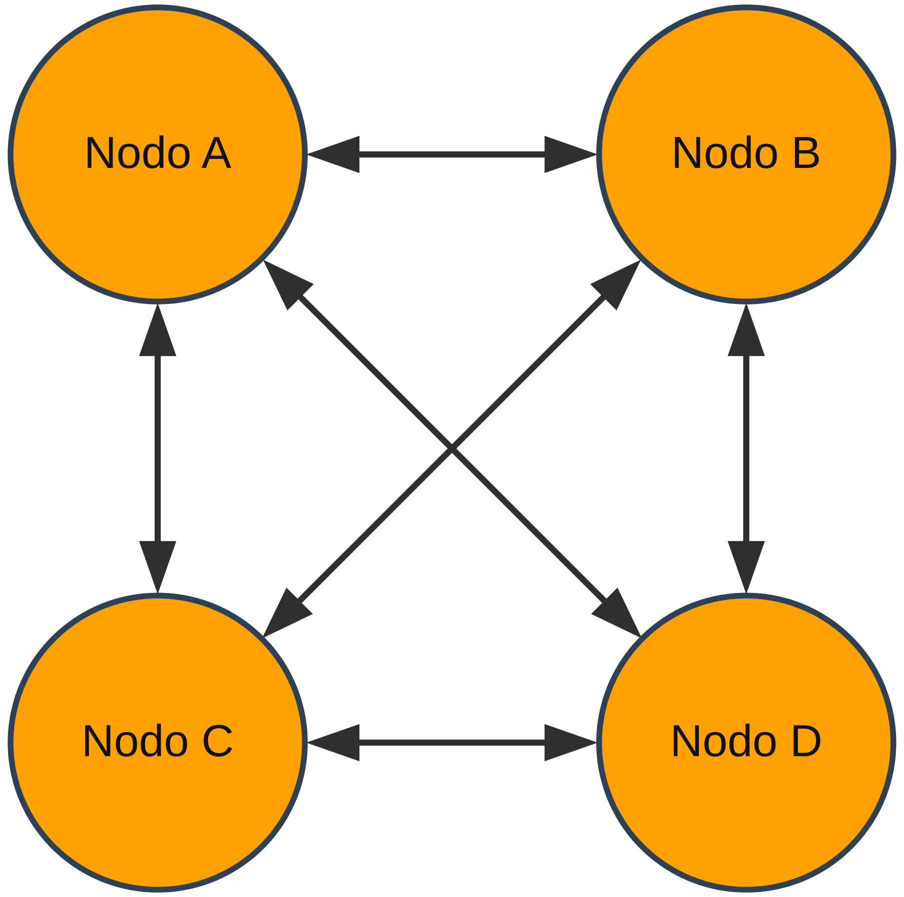
LIMITI DEL MODELLO CENTRALIZZATO

- single point of failure (SPOF)
- bottleneck e scalabilità limitata
- ridotta resilienza a fault



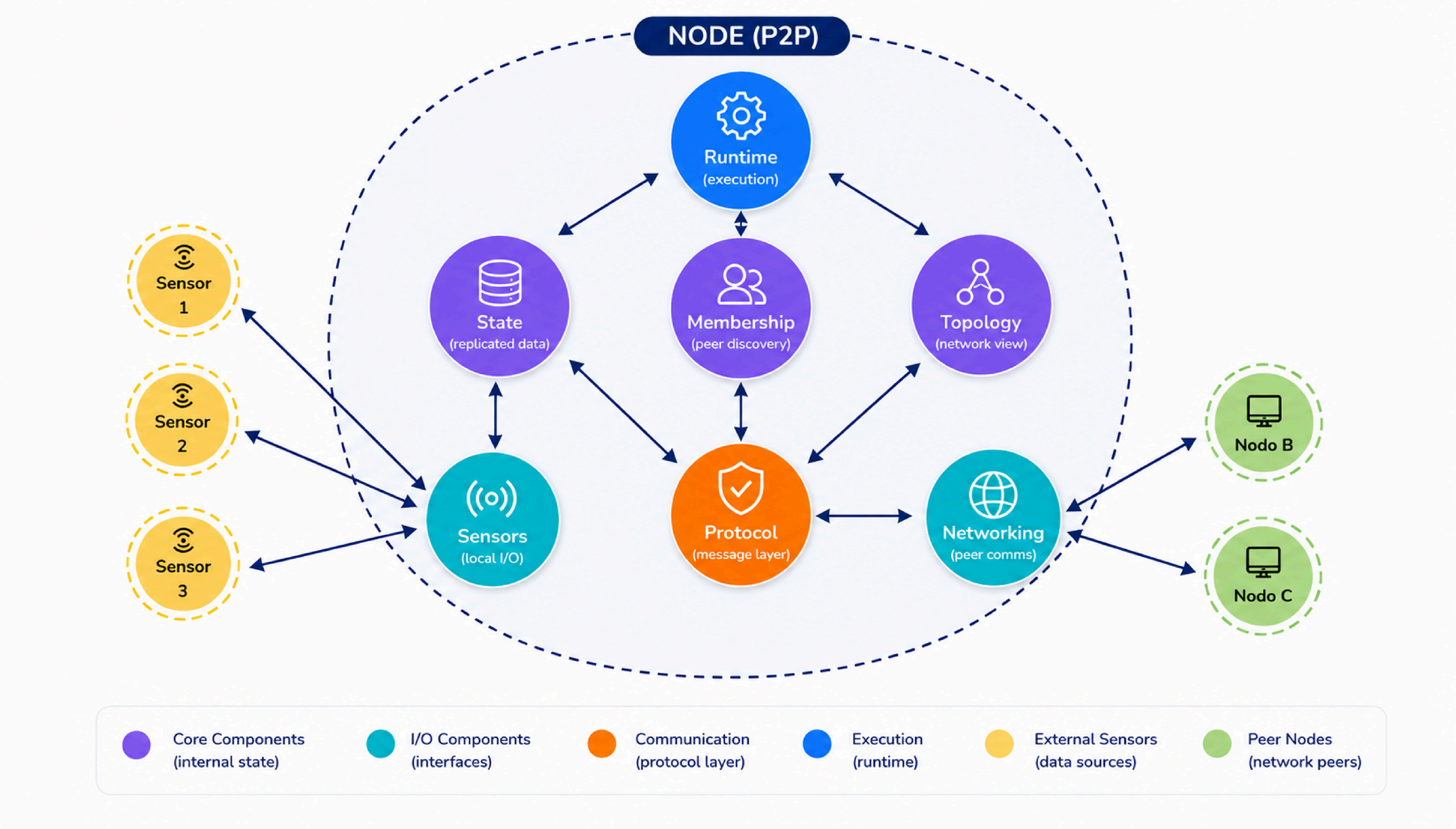
VANTAGGI DEL MODELLO DECENTRALIZZATO

- maggiore **disponibilità**
- eliminazione SPOF
- migliore **scalabilità**
- migliore **resilienza** operativa sotto failure e disconnessioni



No central coordinator

Architettura di un nodo



Protocollo & Tipi di Messaggi

Il sistema utilizza un protocollo JSON custom per la comunicazione tra peer.

Messaggi principali

- JOIN_REQUEST
- PEER_LIST
- PING / PONG
- SENSOR_UPDATE
- GOSSIP_STATE
- GET_DELTA
- DELTA_UNAVAILABLE
- FULL_SYNC_REQUEST / RESPONSE

Ogni messaggio trasporta

- identificativo
- timestamp
- payload applicativo


```
{
    "type": "SENSOR_UPDATE",
    "sender_id": "node-2",
    "timestamp": 1779185730000,
    "payload": {
        "sensor_id": "room-a-temp-01",
        "value": 22.7,
        "ts_ms": 1779185729500,
        "origin": "node-2",
        "meta": {
            "unit": "C",
            "period_ms": 1000
        },
        "seq": 8421
    }
}
```

Gossip-Based State Replication

Ogni nodo periodicamente

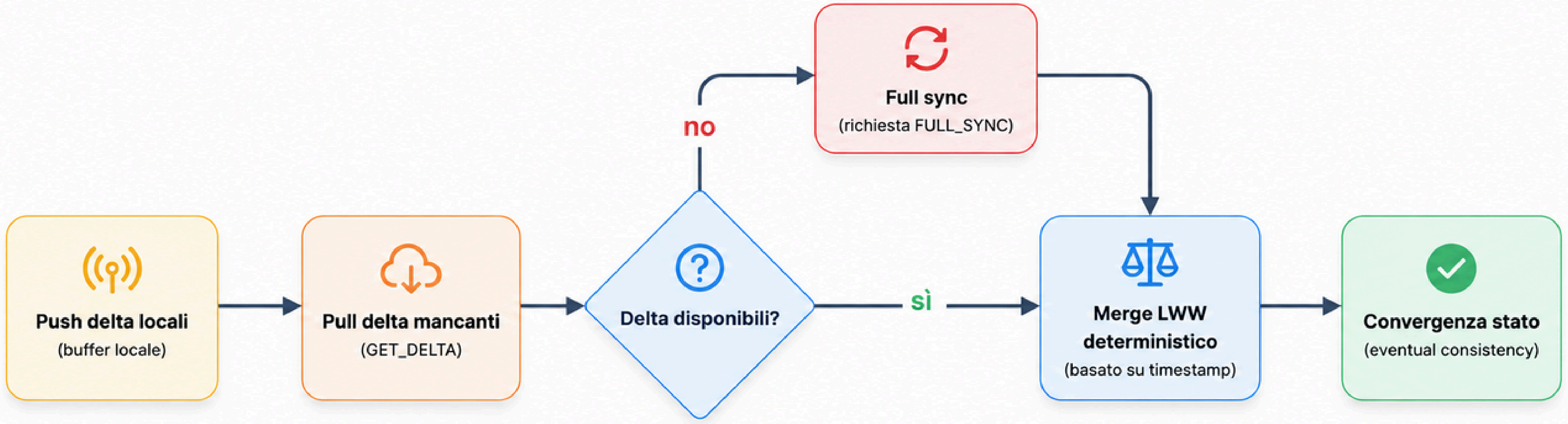
- invia **delta locali** ai peer (push)
- richiede **delta mancanti** ai peer (pull, GET_DELTA)
- se la storia non è più disponibile, esegue **full sync**

Merge policy

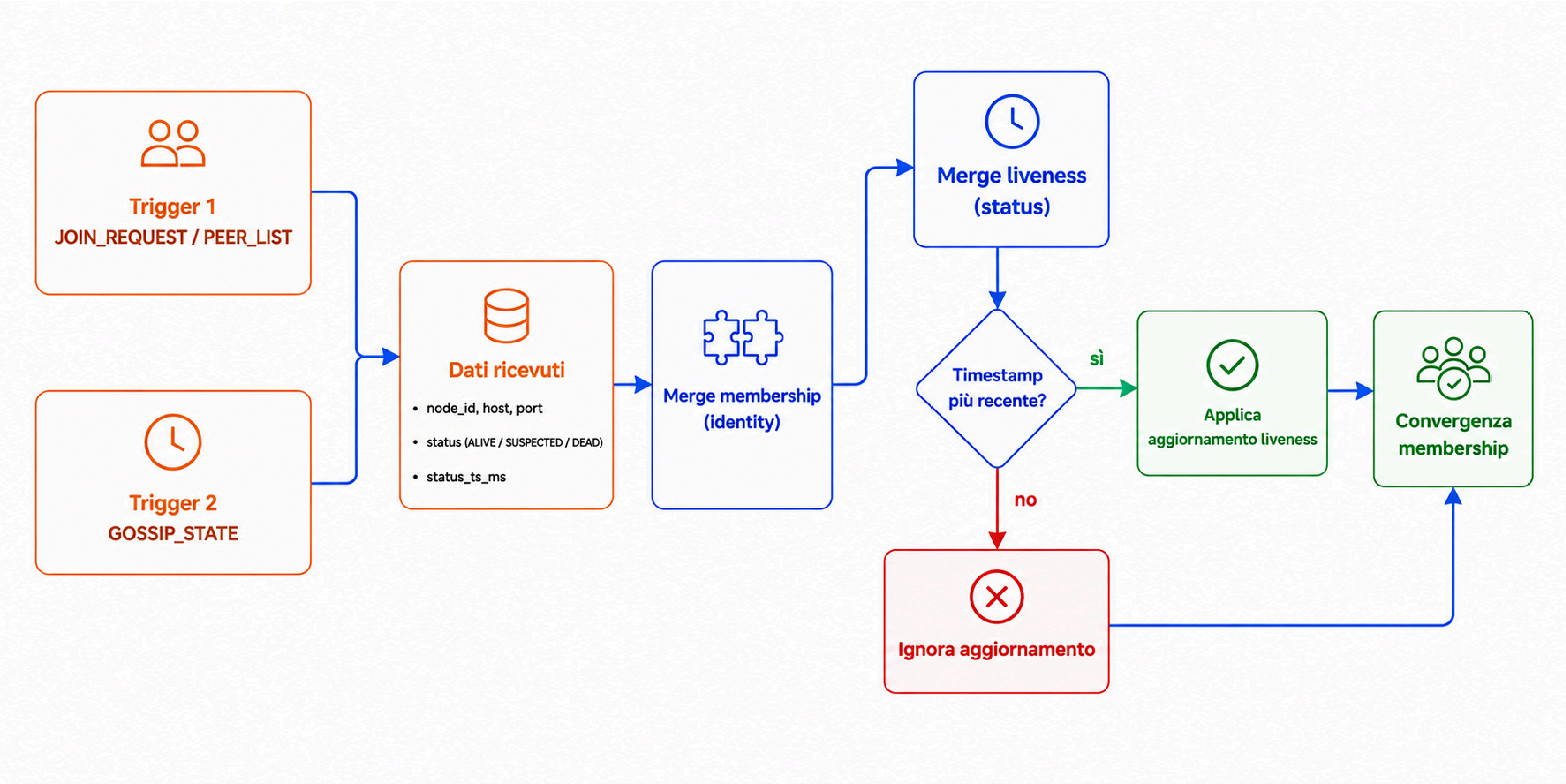
- last-writer-wins (LWW)
- merge deterministico

Proprietà

- **convergenza** dello **stato**
- ***eventual consistency***
- nessun coordinatore centrale

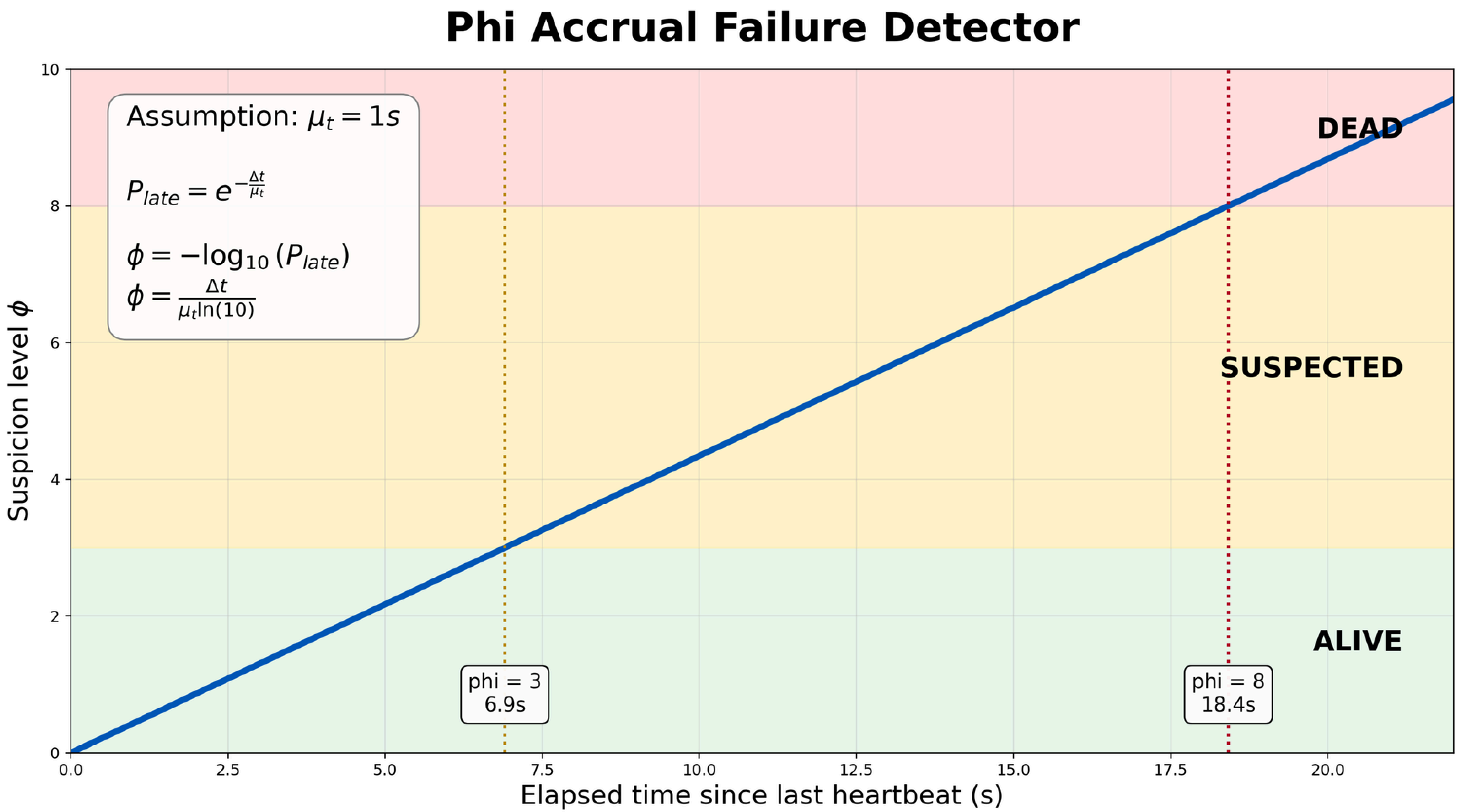


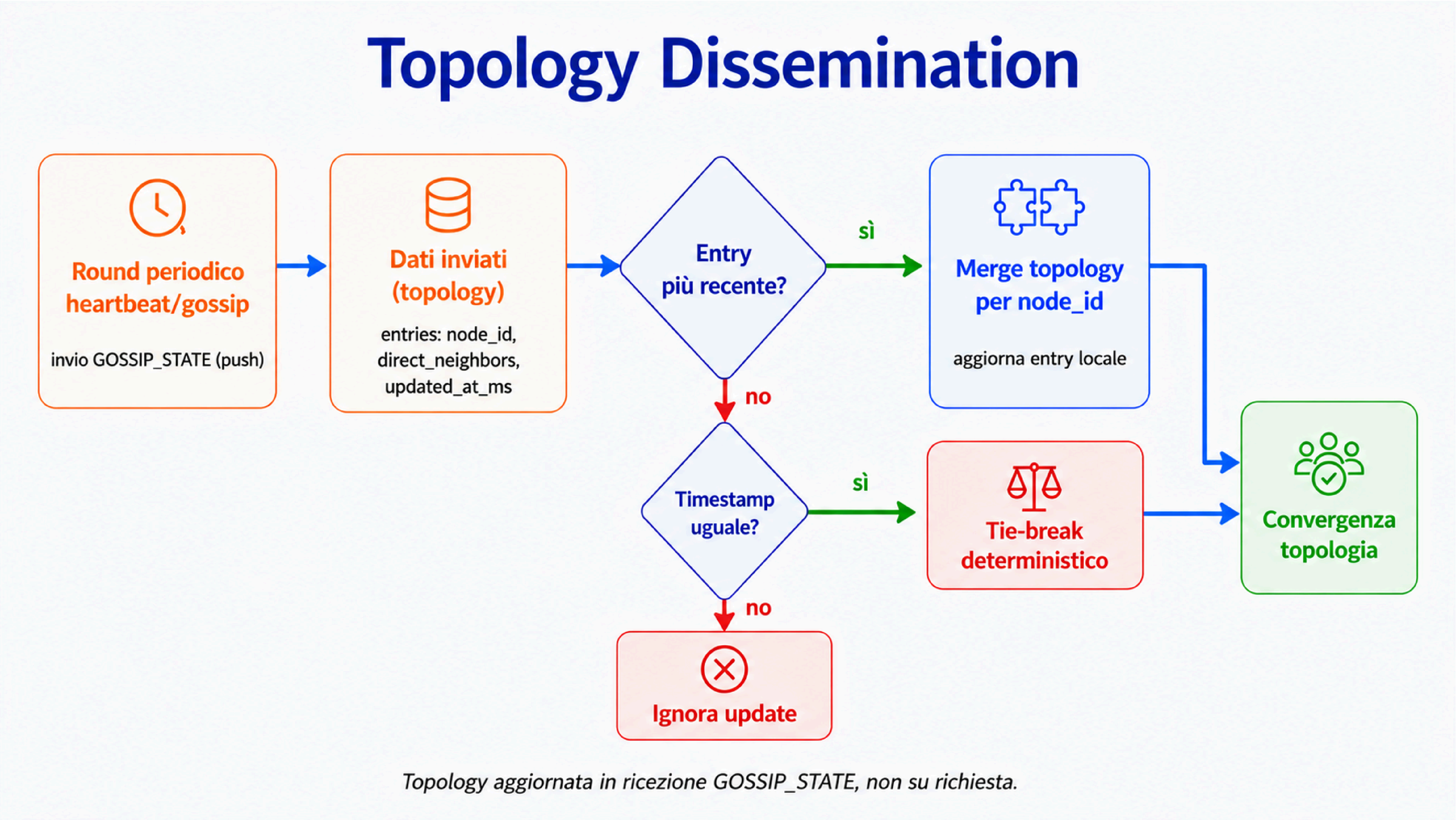
Membership Replication



Phi-accrual failure detector

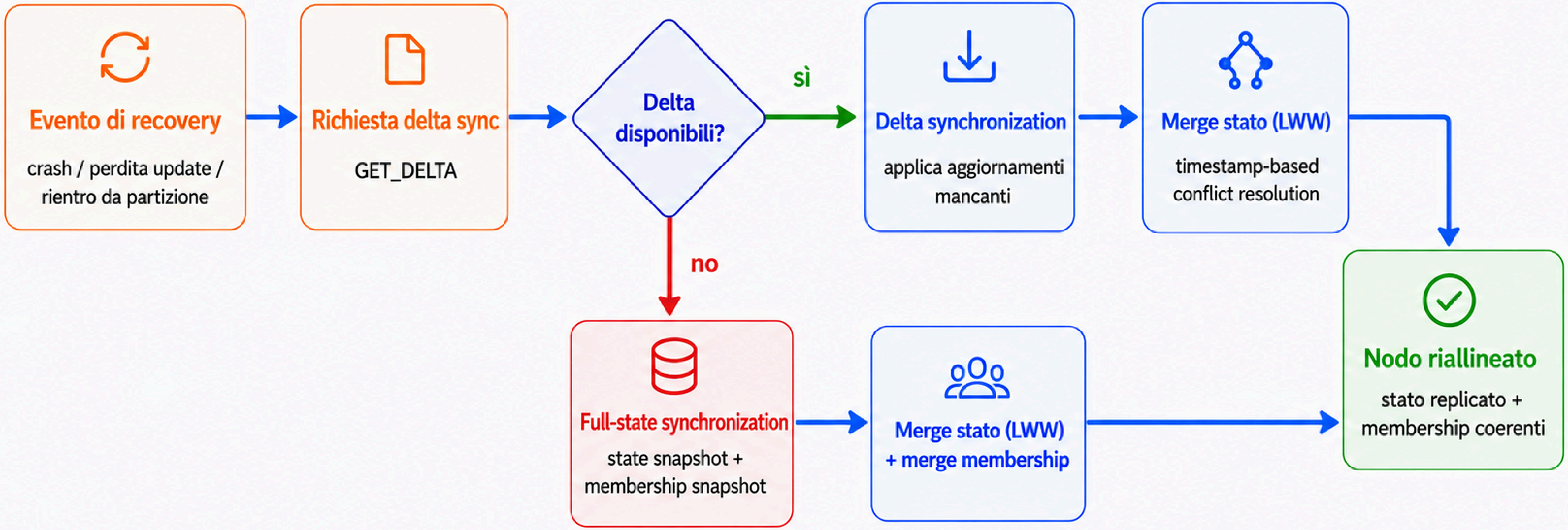
- Detector adattivo (basato su heartbeat)
- diminuisce falsi positivi, sensibilità progressiva
- sospetto cresce con il ritardo degli hb





Usata solo per osservabilità

Recovery & Synchronization



Topology riallineata tramite gossip topology exchange.

Modello di Consistenza & CAP trade off

Il sistema implementa

- **eventual consistency**
- **merge** basato su timestamp
- **last-writer-wins** (LWW) per la risoluzione dei conflitti

Secondo il modello **CAP**, il sistema privilegia **AP** sotto network partition

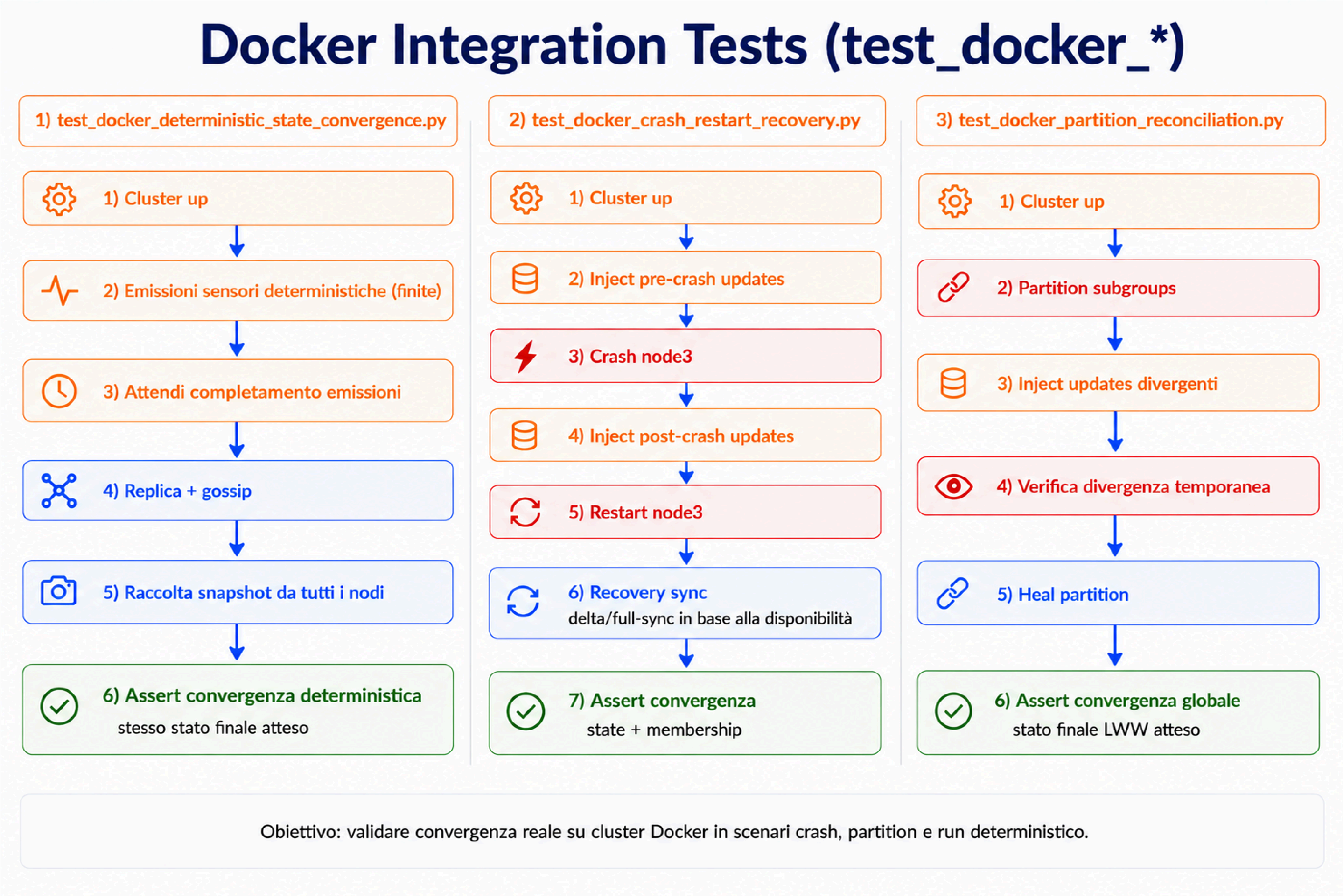
- **availability**
- **partition tolerance**

La **strong consistency** non viene adottata per evitare coordinamento globale ad ogni update

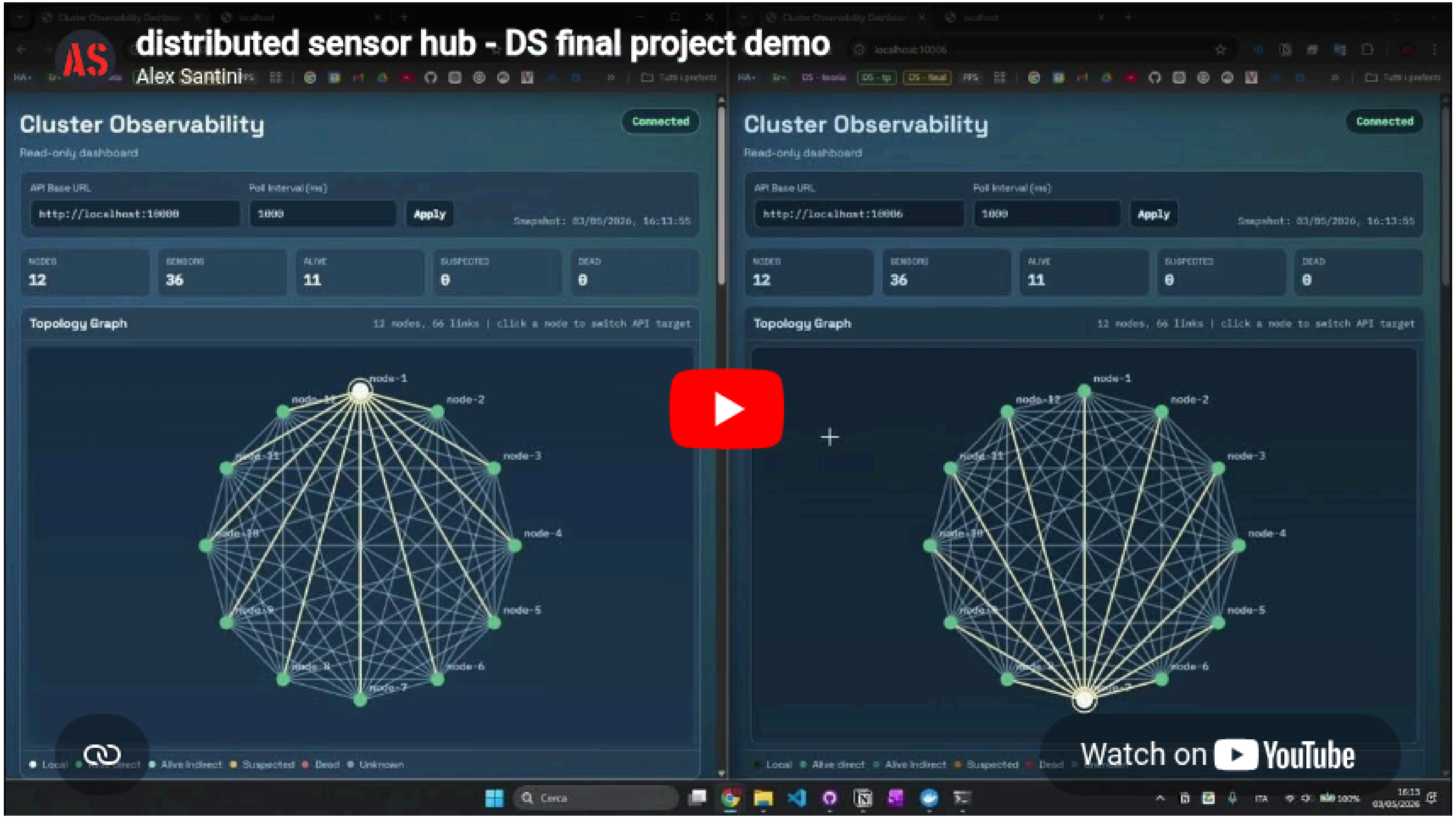
Implementazione & Tecnologie

Tecnologie utilizzate

- Python
- TCP sockets (comunicazione peer-to-peer)
- Multithreading (runtime concorrente per heartbeat, gossip, sync)
- Docker compose (orchestrazione cluster multi-nodo)
- JSON


Docker Integration Tests (test_docker_*)
1) test_docker_deterministic_state_convergence.py
1) Cluster up
2) Emissioni sensori deterministiche (finite)
3) Attendi completamento emissioni
4) Replica + gossip
5) Raccolta snapshot da tutti i nodi
6) Assert convergenza deterministica
stesso stato finale atteso
2) test_docker_crash_restart_recovery.py
1) Cluster up
2) Inject pre-crash updates
3) Crash node3
4) Inject post-crash updates
5) Restart node3
6) Recovery sync
delta/full-sync in base alla disponibilità
7) Assert convergenza
state + membership
3) test_docker_partition_reconciliation.py
1) Cluster up
2) Partition subgroups
3) Inject updates divergenti
4) Verifica divergenza temporanea
5) Heal partition
6) Assert convergenza globale
stato finale LWW atteso
Obiettivo: validare convergenza reale su cluster Docker in scenari crash, partition e run deterministico.

Video Demo



Link al video

Limiti e Sviluppi futuri

Limiti attuali

- nessuna **Byzantine fault tolerance**
- overhead gossip crescente al crescere dei nodi (in topologia **full mesh**)
- **stato** mantenuto in **memoria**
- **sensor ingestion non replicato nativamente** (possibile perdita dati se cade il nodo gestore)

Possibili sviluppi

- persistent storage per stato e log delta
- **adaptive gossip intervals**
- **autenticazione e autorizzazione peer**
- monitoring dashboard avanzata
- **replicated sensor ingestion** (multi-writer su più nodi con deduplica)

Grazie per l’attenzione