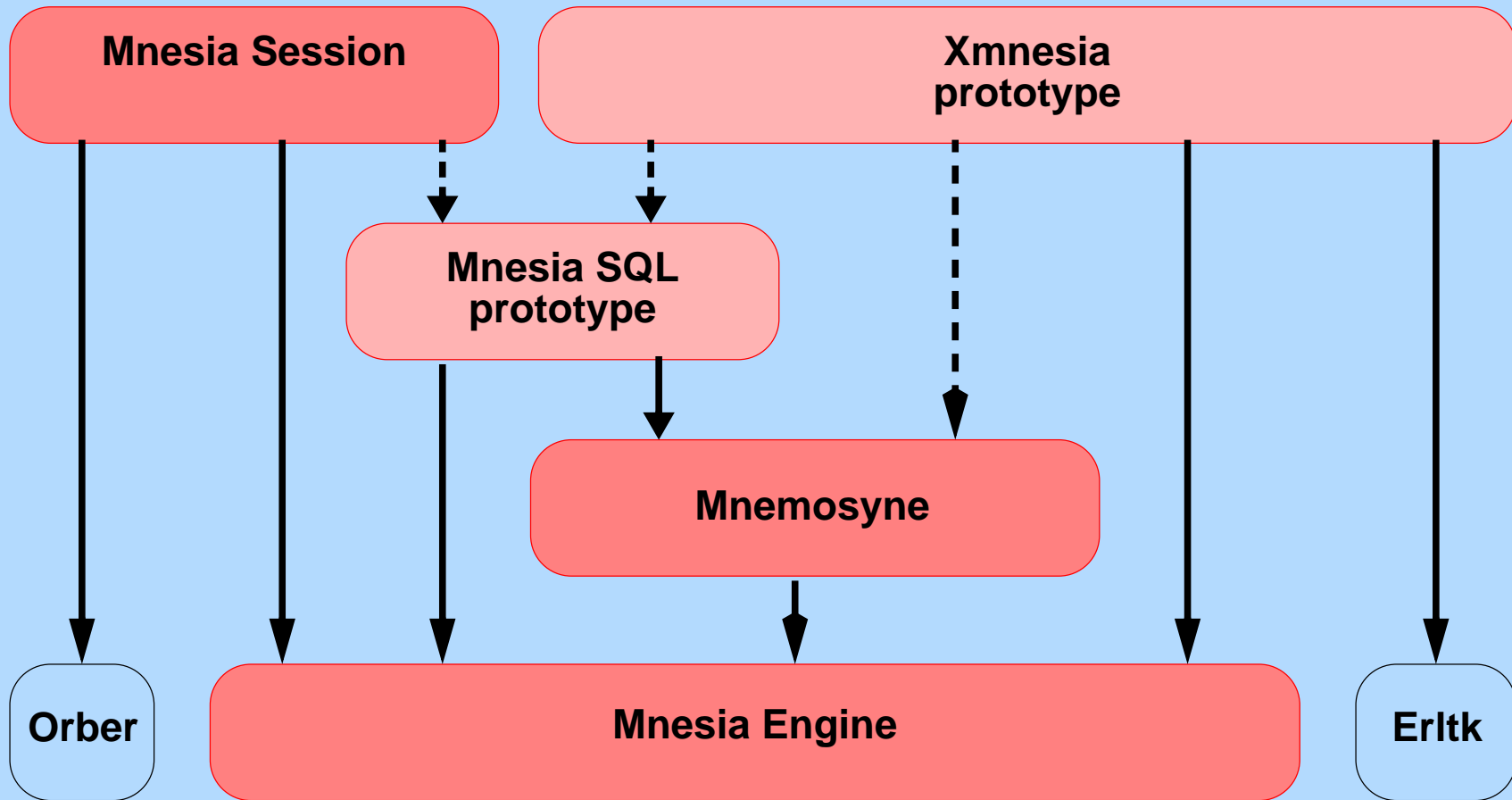


MNESIA

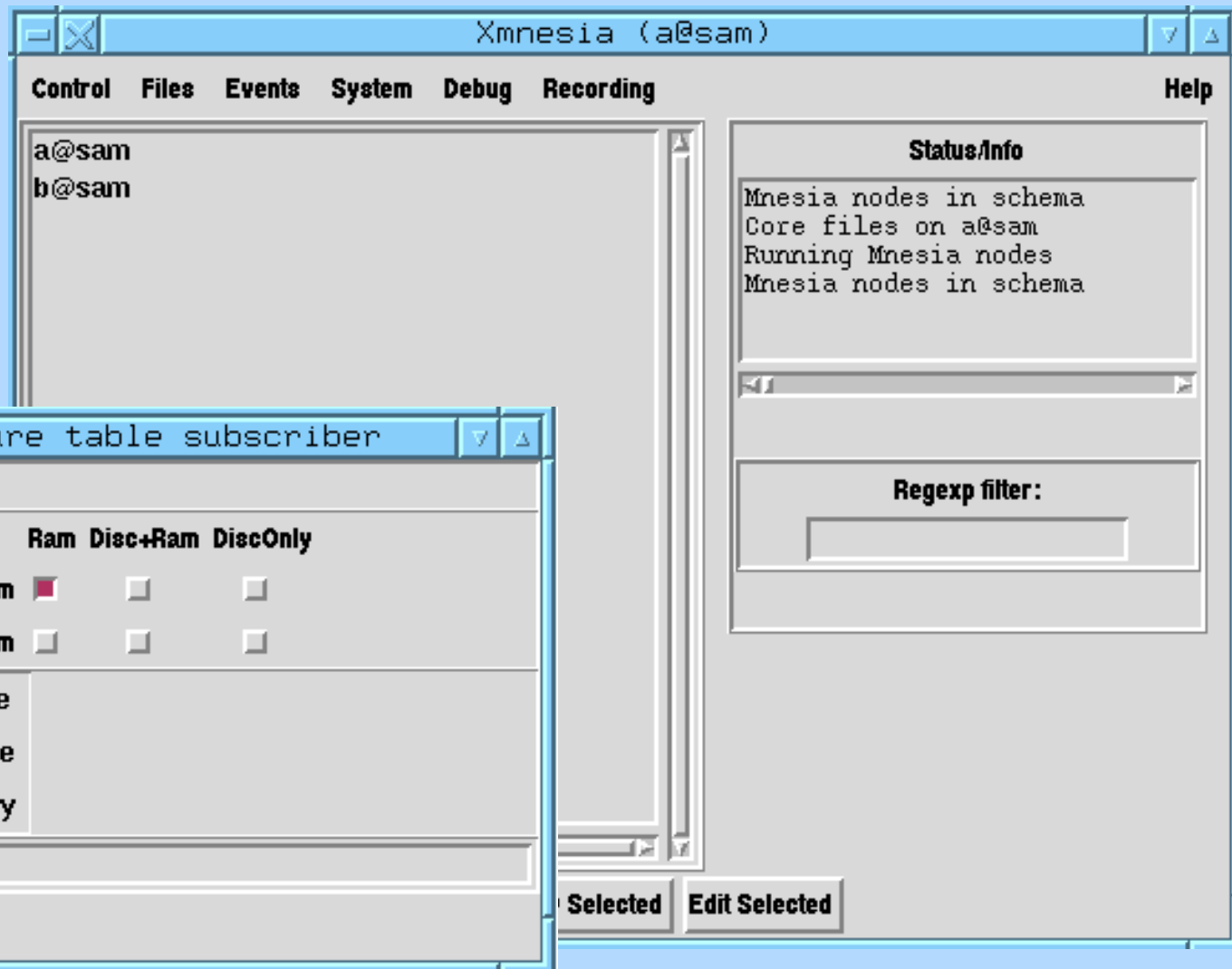
Internals



The Mnesia application family

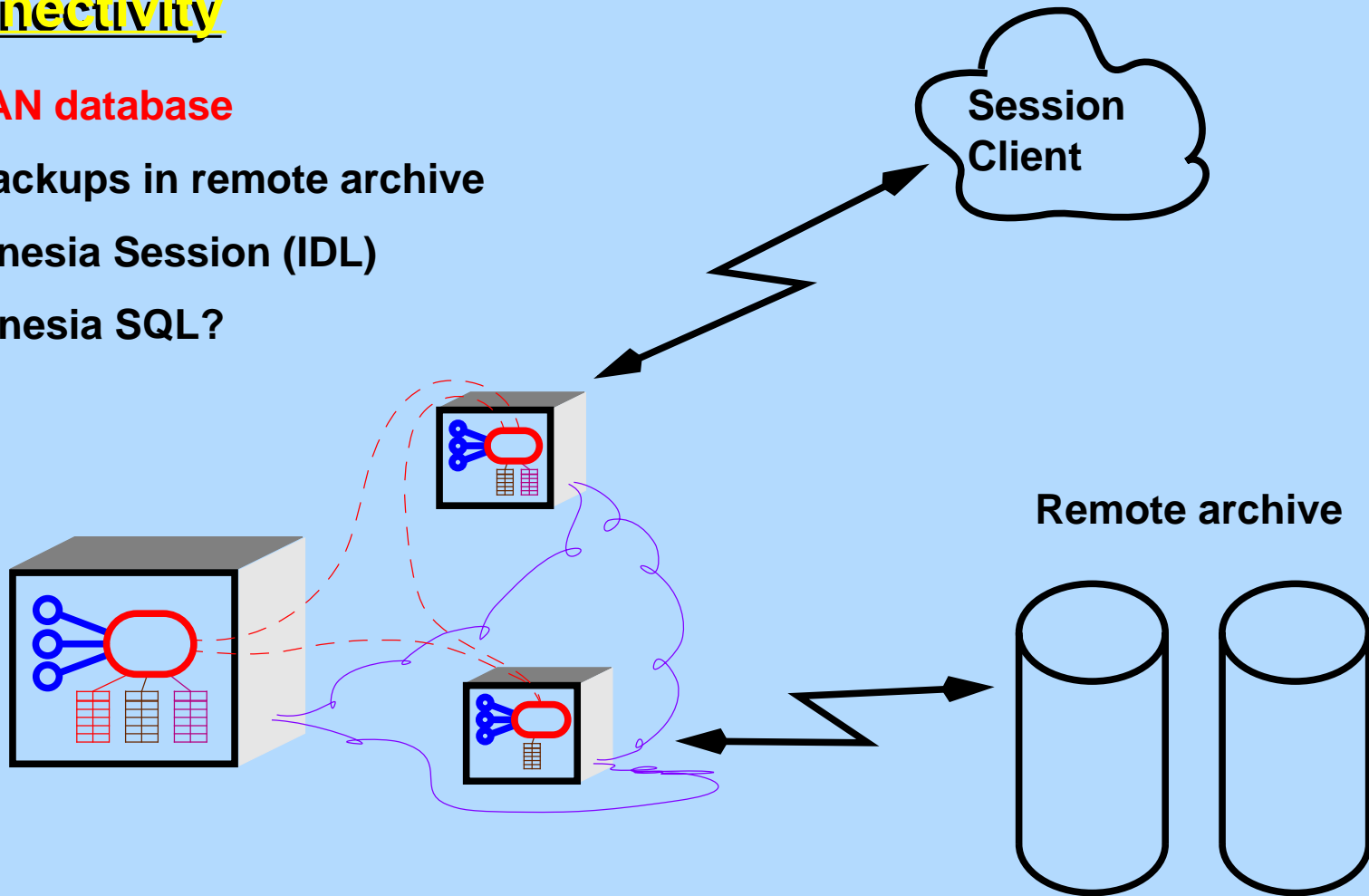


Xmnesia



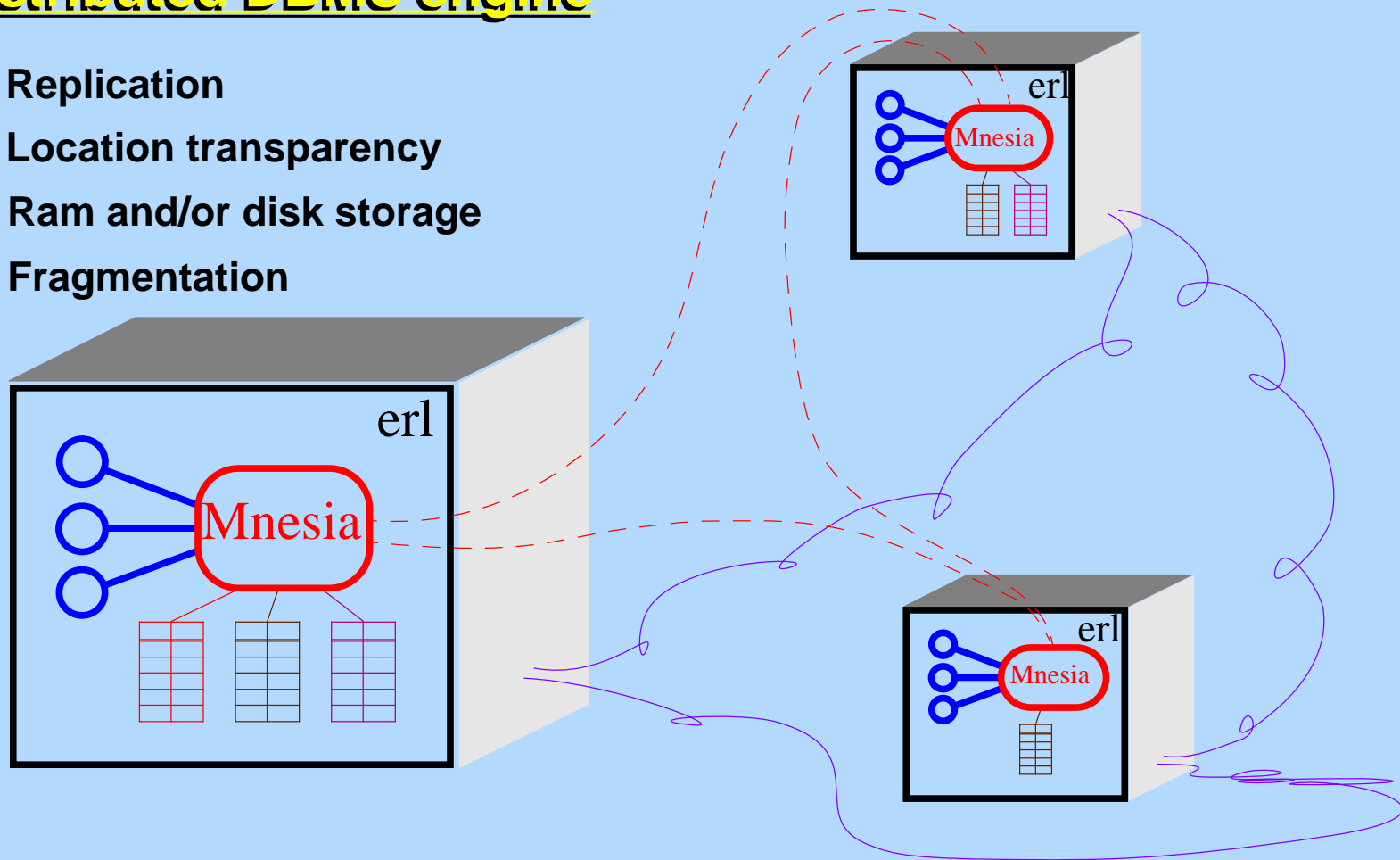
Connectivity

- LAN database
- Backups in remote archive
- Mnesia Session (IDL)
- Mnesia SQL?



Distributed DBMS engine

- Replication
- Location transparency
- Ram and/or disk storage
- Fragmentation



Evaluation contexts

- mnesia: ets(Fun, Args)
- mnesia: async_dirty(Fun, Args)
- mnesia: sync_dirty(Fun, Args)
- mnesia: transaction(Fun, Args)

```
-record(employee, {name, phone, salary}).
```

```
Fun = fun(Who, Raise) ->
```

```
    [Emp] = mnesia:read({employee, Who}),
```

```
    Salary = Emp#employee.salary + Raise,
```

```
    mnesia:write(Emp#employee{salary = Salary})
```

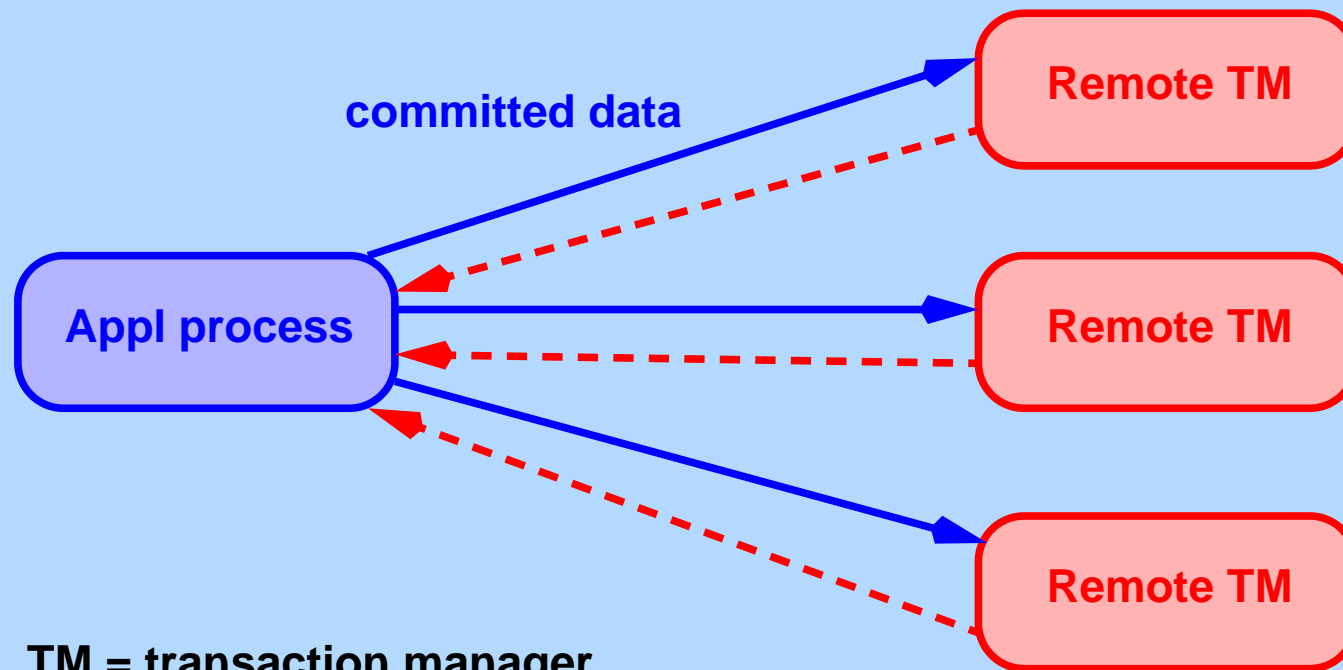
```
end.
```

```
Args = [Who, Raise].
```

mnesia_access callback module

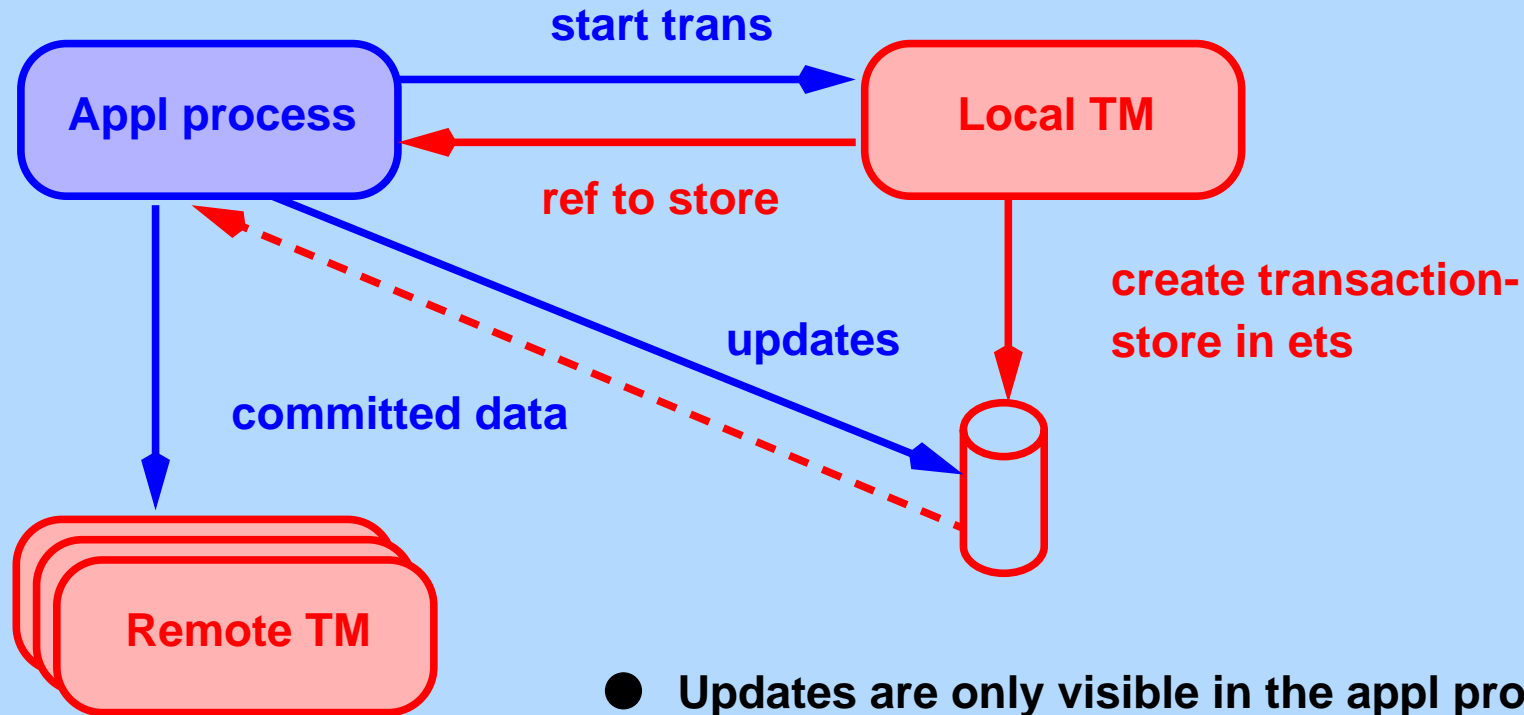
Dirty commit protocols

- Commit work is performed in the application process + remote TM's
- ets, async_dirty and sync_dirty



TM = transaction manager

Storage of transaction data



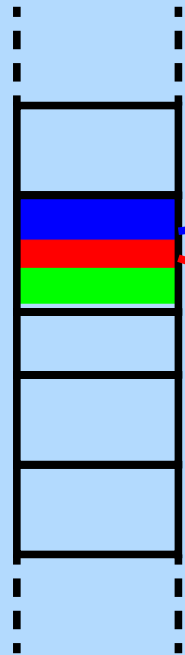
- Updates are only visible in the appl process
- Commit work is performed in the application process + remote TM's

Commit work needed on each node

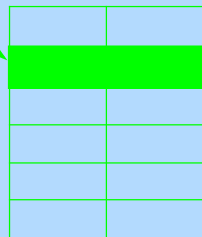
- Optionally append to transaction log - disk_log
 - Optionally update memory resident B-tree for SNMP
 - Optionally store old values in checkpoint retainer - (d)ets
 - Optionally send messages to subscribing processes
 - Optionally update secondary indecies - (d)ets
 - **Mandatory update of the main table - (d)ets**
- Optionally send message to locker to release locks

Dump of transaction log

disk_log

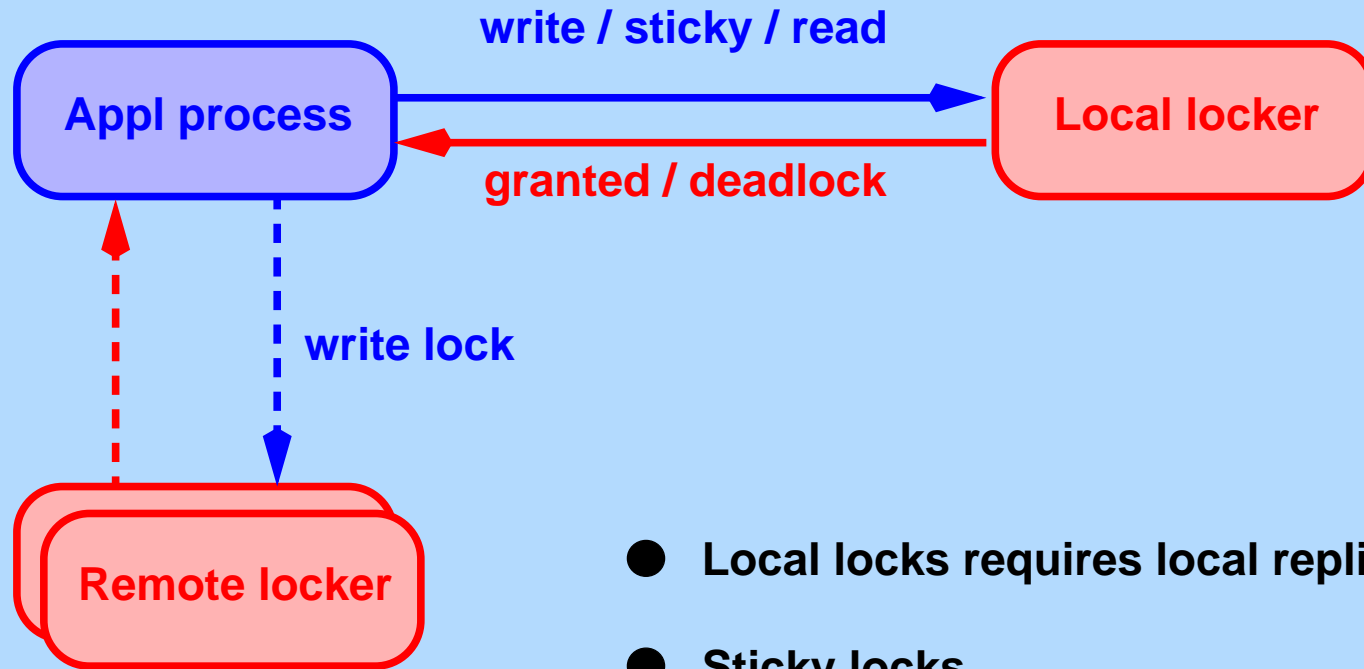


dets



- Thresholds, **startup**
- Dual log files
- Move records to dets
- **Ensure atomicity**

Lock acquisition



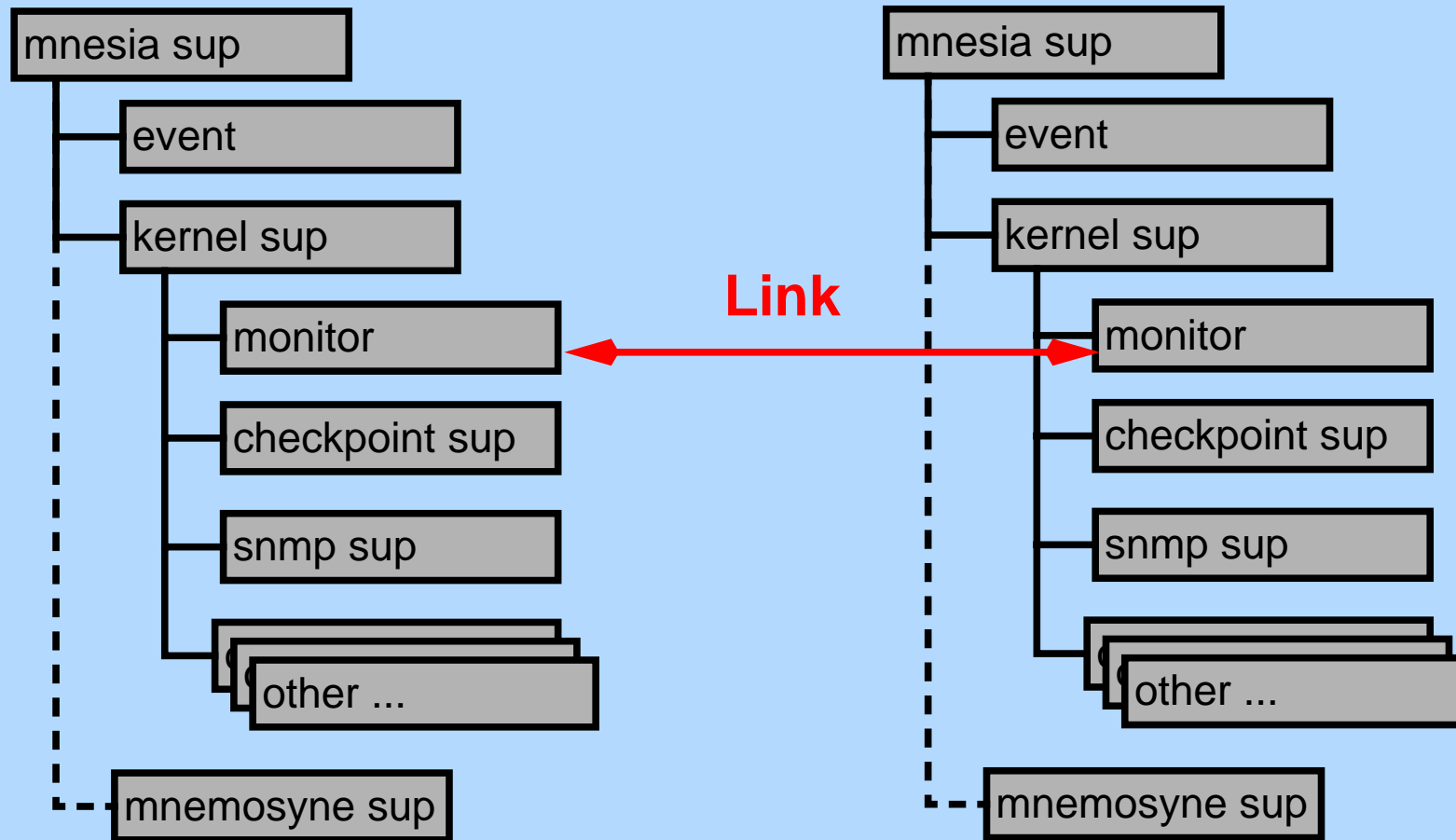
- Local locks requires local replica
- Sticky locks
- Multi-granular locks on records or tables

Lock conflicts

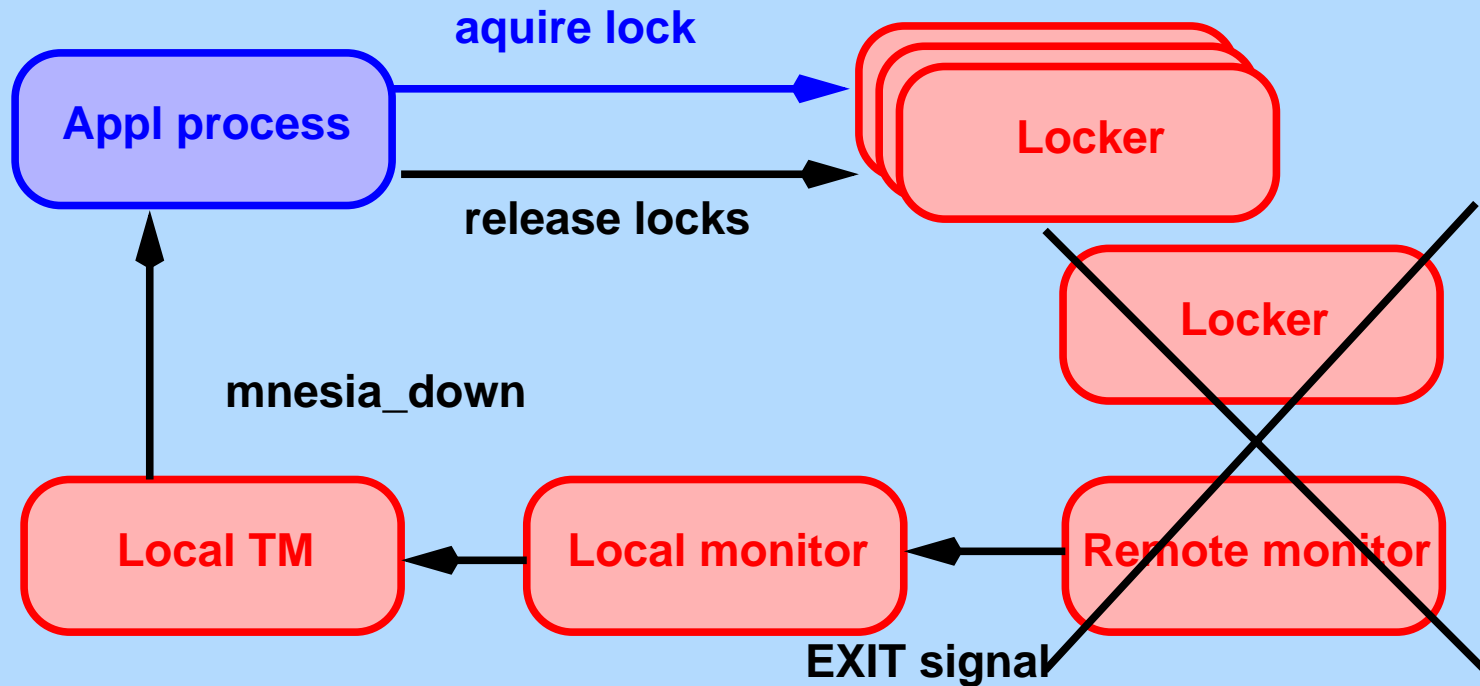


- Queue of conflicting lock requests
- Unfair index locks :-)
- Deadlock prevention
- Lamport clock

Supervision and event handling



Handling of node loss



- Re-run transaction at node loss or deadlock
- Log mnesia_down to disk

Lightweight transaction protocol



- Non-blocking commit protocol
- Local TM becomes new coordinator if appl process dies
- Rely on "recovery by table load"
- Only used for symmetrically replicated tables

Recovery by table load

- 1 ● Dump transaction log (twice) at startup
- 2 ● **Presume commit** if the "transaction" has been logged
- 3 ● Copy table from active replica if there is any
- 4 ● Load table from local disk if we have got
mnesia_down's from all other nodes with a replica
- 5 ● Select an arbitrary loader if no one decided to load

Heavyweight transaction protocol

- Non-blocking commit protocol
- Local TM becomes new coordinator if appl process dies
- May be used for any tables, but is avoided for performance reasons
- Currently used for schema and asymmetrically replicated tables



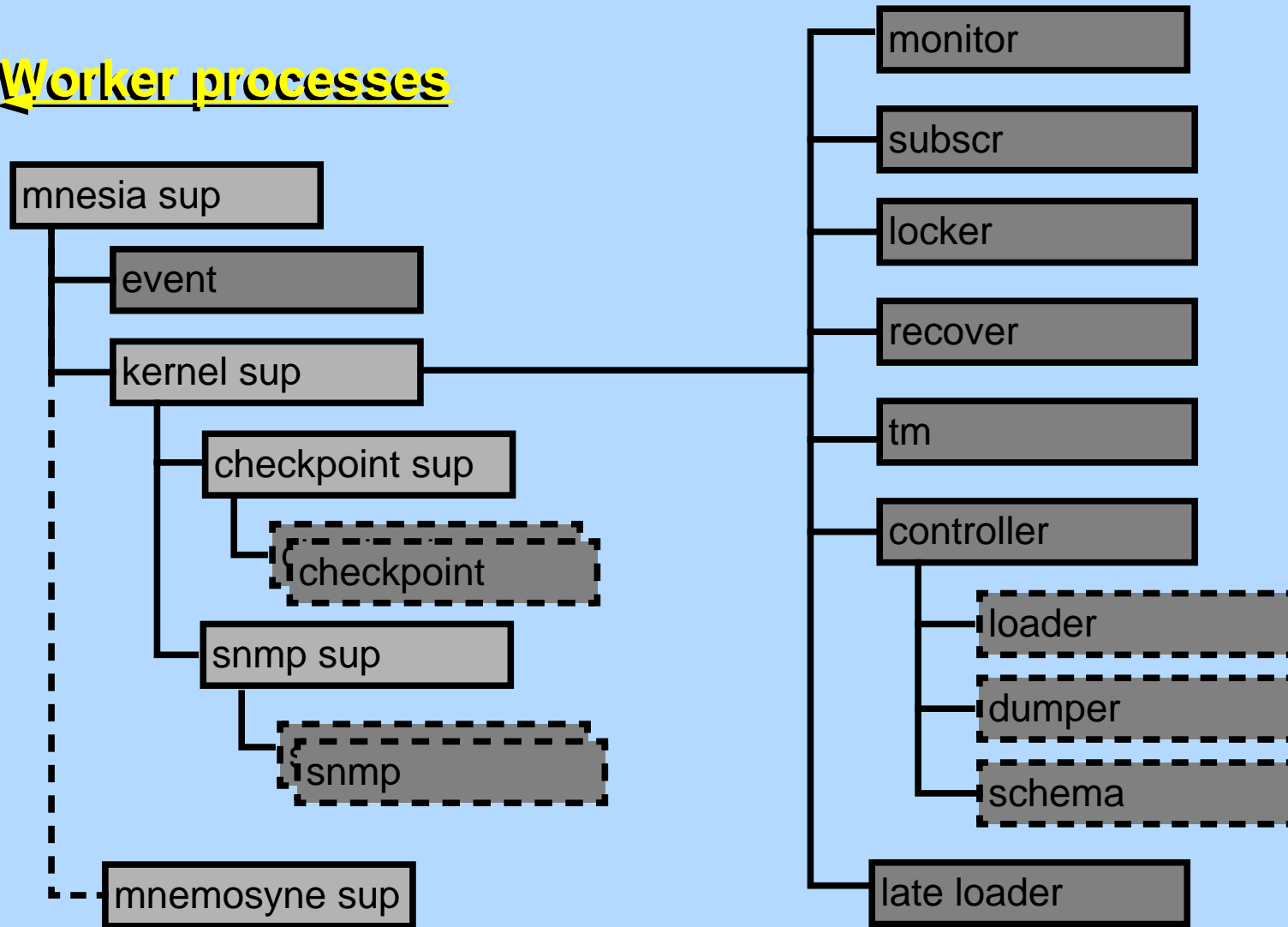
Heavyweight transaction protocol



Recovery of heavyweight transactions

- 1 ● Dump transaction log (twice) at startup
- 2 ● Send local decisions to other nodes at startup
- 3 ● Consult decision log about transaction outcome
- 4 ● Presume abort if all involved nodes are up, but unclear
- 5 ● Wait until we know the outcome, this may in case of double error, be until all involved nodes are up
- 6 ● Presume abort after `max_wait_for_decision` (default infinity)

Worker processes



Lines Of Code (October 1999)

- Mnesia Engine ~15.600 LOC
- Mnemosyne ~4.900 LOC
- Mnesia SQL prototype ~6.000 LOC (+generated parser ~38.000 LOC)
- Xmnesia prototype ~3.000 LOC
- Mnesia Session ~1.400 LOC
- Test suite (Mnesia Engine) ~14.000 LOC

Test suite (Mnesia Engine)

- Acceptance (install)
- Nice coverage, Evil coverage
- Examples
- Configuration parameters
- Atomicity, Consistency, Isolation, Durability, Recovery
- Prediction, Fairness, Benchmarks, Resource Consumption, Scalability
- **Light weight test server. Simple to run single test cases in debugger.**

Documentation roadmap

- Mnesia Overview, presented at PADL' 99, written for Erlang novices
http://www.ericsson.se/cslab/~hakan/mnesia_overview.pdf
- Erlang Book
- Erlang Language Extensions
- Mnesia User's Guide
- Mnesia Reference Manual

Erlang/OTP including Mnesia is available as open source at

<http://www.erlang.org>