

| Lost rite

A gentle kick in the nuts by an angry mule

Me ?!

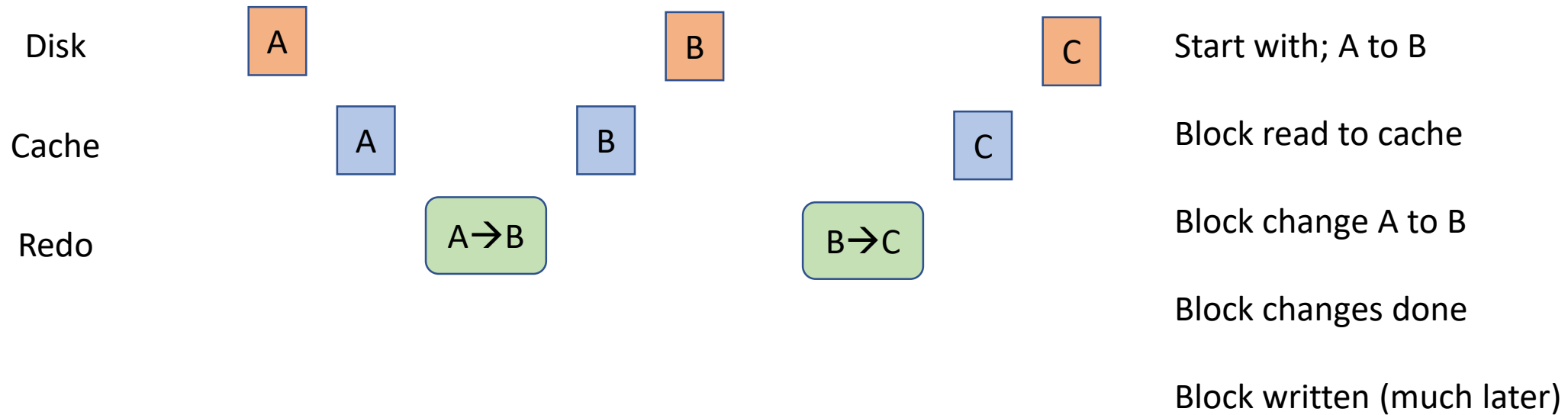
- **Michael Möller** (aka. M-square or M²)
- Worked with IT last 50 years (started early)
- Oracle DBA with Oracle and Miracle for 20+ years.
 - M2@miracle.dk
- Oracle Certified Master, OakTable Member
- Although I'd rather be a steampunk Tesla-coil gunner on a steam-airship ...



Lost Write – what happens

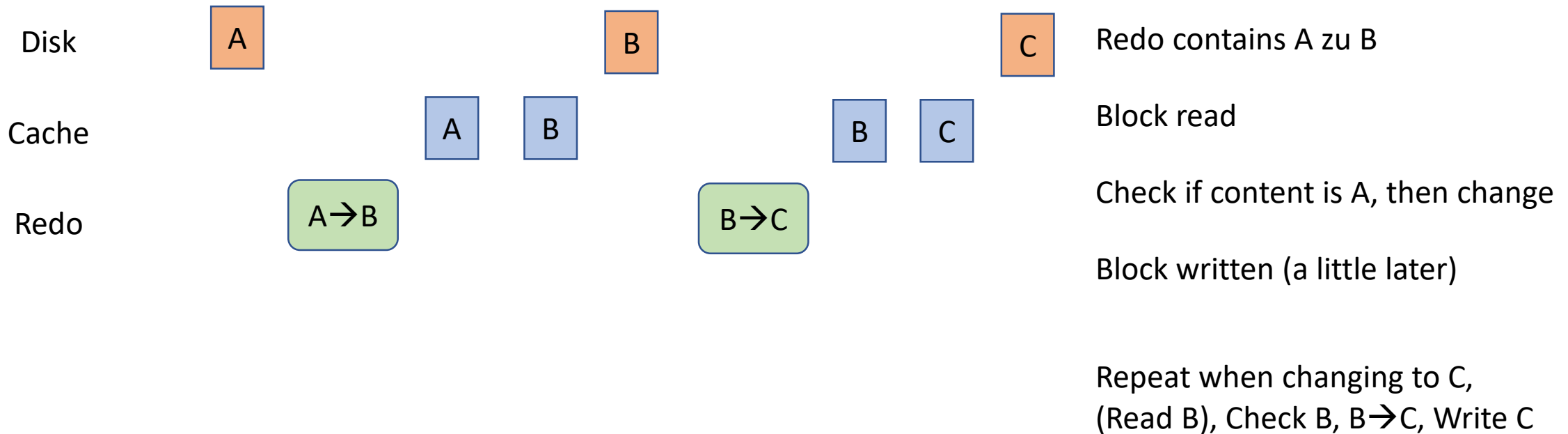
- A data block is changed
- Oracle requests write of block to disk
- The block is **not** written (or in the wrong place)
- The O/S acknowledges „Write Succeeded“
- You get Data loss and Recovery becomes impossible (past this point)

Normal operation

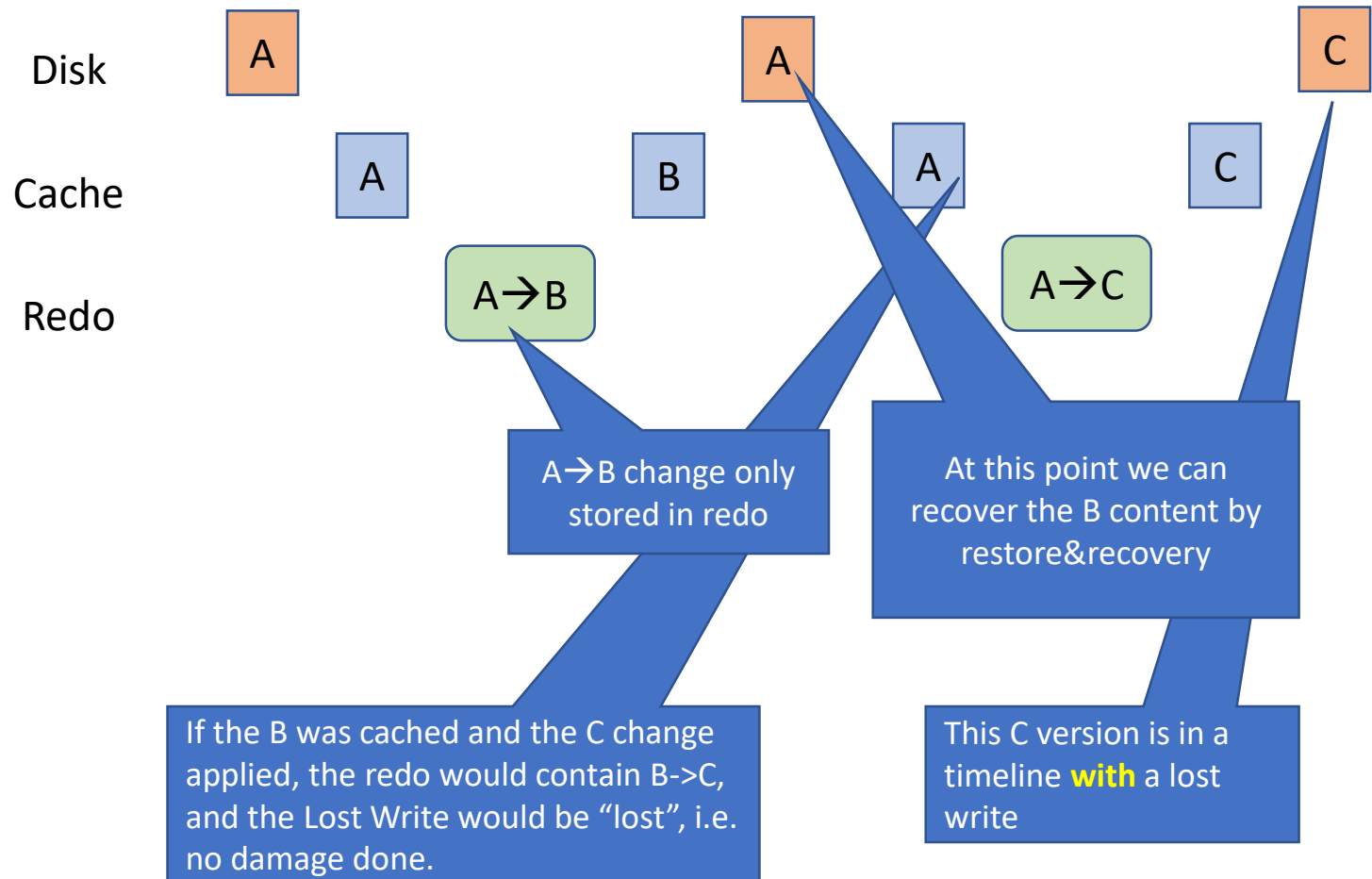


Repeats when changing to C,
Read the B, B→C Vector, C written

Normal Standby or Recovery



Lost Write scenario



First change, something to B

Block read to cache

Block changes for A to B

Block changes done

Block write *lost*

Second change, something to C

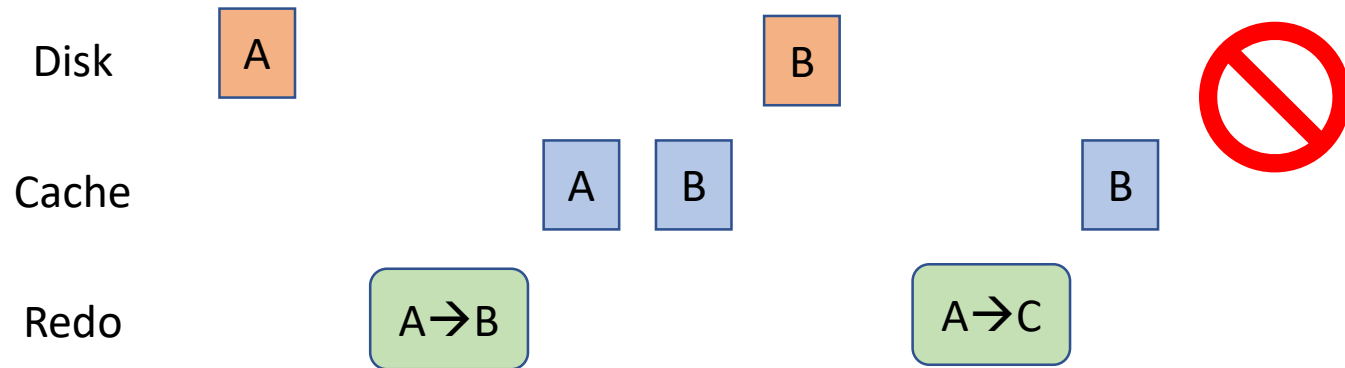
Block read to cache

Block changes for A to C

Block changes done

Block written (much later)

Standby or Recovery fails



Redo contains A to B

Block read

Check it is A, apply change

Block written (later)

Redo contains A to C

Block read

ORA 600 [3020]; A was expected

Likely causes

- Hardware – Memory, disk, connections (disks not local to node)
- Software – O/S, Filesystem, Virtualization
- Many components has Caches, Load sharing and "striping".
- Zillions of writes issued, even with "parity" and "checksum" on many levels – because **faults do occur**.
 - Faults are usually autocorrected or at least reported.
- Sometimes it is a double/triple error, which slips through the checks
 - Is not detected at the occurrence but later by Oracle

What happens in Oracle

- As always the redo block contains the change with the SCN
- As always, the new block content in memory is later written "in batch" (Scatter write) to disk
- That write went well (except it **did not**)
- When you read the block all is well, except you see old data.
- Further updates to rows in the block work well, too
- BUT THEN you have the irrevocable **Lost Write**

How does Oracle detects it? (Standard)

- If you never write to the block again: remains undiscovered
 - But you have lost data!
- If you do further updates : still remains undiscovered

BUT

- ***IF*** you do recovery when processing the 2nd redo, Redo and Data inconsistency – **stuck recovery**
- Which also happens in (physical) Dataguard, as it is a constant recover

How Oracle detects it (with Lost Write Detect)

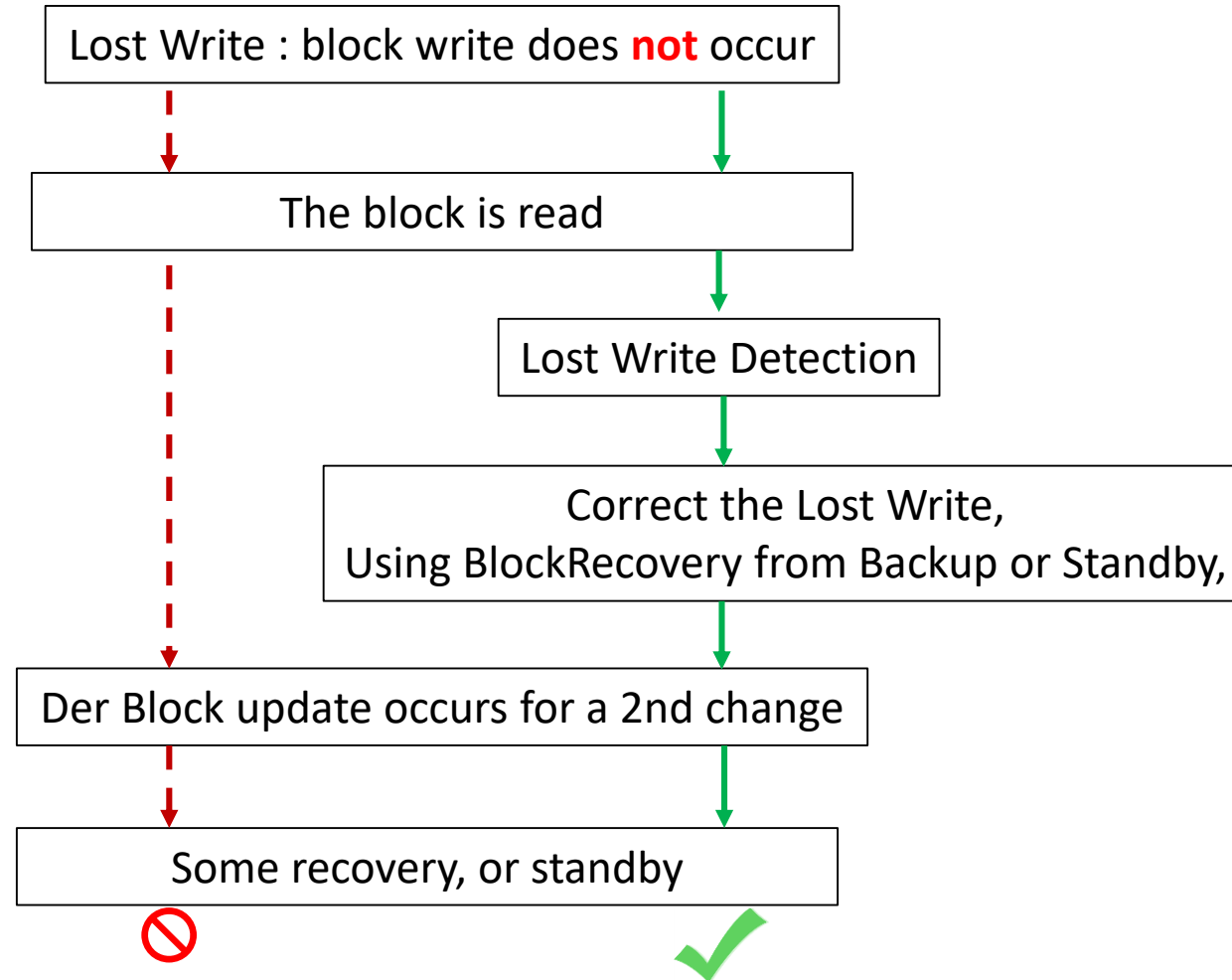
- **DB_LOST_WRITE_PROTECT=TYPICAL** (NONE, FULL)
- Adds to Redo what it read (DBA,SCN)
- Which means: nothing is detected (!)

- At any recovery the first Read in redostream discovers the Lost Write
 - (without DB_LOST_WRITE_PROTECT, discovery occurs after the 2nd write)
- **THEREFORE:** Do a running recovery (like Standby) for early detection

Data Guard Lost Write Detection

- The LostWrite Detection is dependent on the usage pattern of the block (i.e. at least read once)
- In short, when the primary reads a block (with a lost write, the “A”), the standby does the same read, (it’s block was written with the redo and thus correct with the “B”) and does a comparison
 - Just the SCN/DBA, not whole block.
- More IO, more Redo
- 18c LostWrite detection without Standby: For choice tables Oracle will keep a shadow copy and do the same comparison on read

Timelines, bad and good



Oracle Demos

LostWrite Detection on a Standby

<http://www.oracle.com/technetwork/database/availability/lostwriteprotection-1867677.exe>

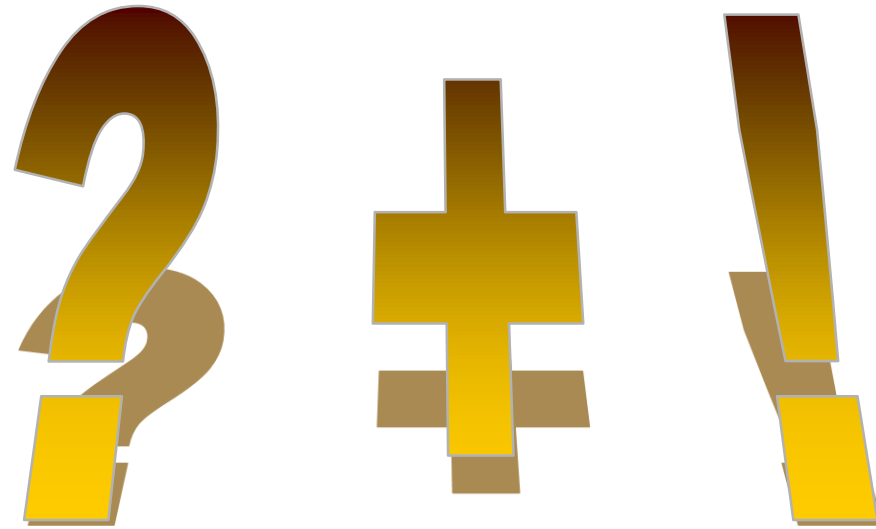
Block Corruption, but not Lost Write

[activedataguard-autoblockrepair-180595.exe](#)

Block corruption correction with Standby

[standby-autoblockrepair-1881372.exe](#)

... and finally,



Thank you,
M2@Miracle.DK