Getting Serious About Redundancy!

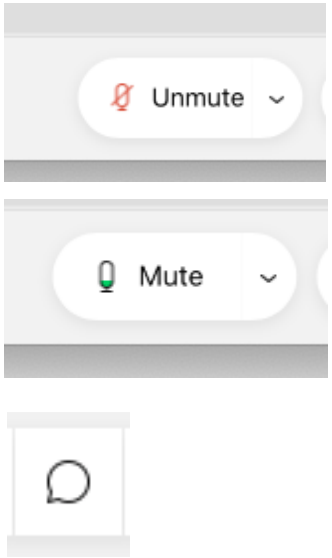
by Tom Girsch

Informix Tech Talks by the IIUG

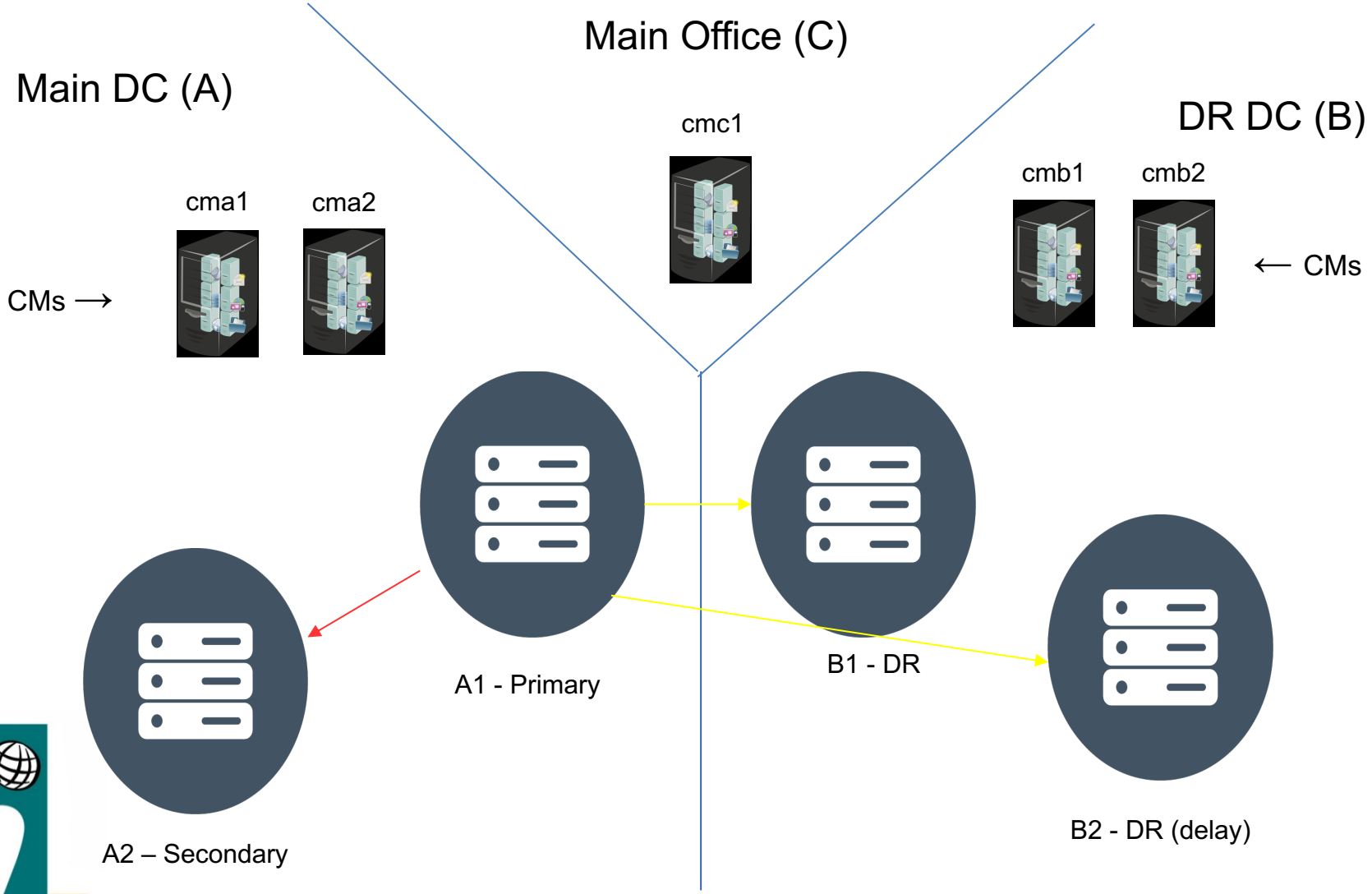
We are launching a new channel on YouTube for Informix Users! This will be a place for Informix how-to videos. More information will be coming soon.

Webcast Guidelines

- **The Webcast is pre-recorded**. The replay and slides will be available on the IIUG Website
- **Please Mute your line.** Background sounds will distract everyone
- **Use the Chat Button** to ask questions



Spoiler Alert!



Why Be Redundant?

- Why Be Redundant?
  - Minimize Downtime
  - Prevent Data Loss
  - Workload Distribution
  - “Things” fail!

- Downsides of Redundancy
  - Considerably More Complicated
  - Vastly More Expensive
  - Trades One Set Of Problems For Another

Repeat the Mantra!

Eliminate Single Points of Failure!

"Things" fail.

Redundancy Starts At The Host

- Multiple CPUs
- Not all hosts will survive the loss of a CPU, but some will
- More memory than you need
- Most modern hosts can disable failed memory DIMMs and keep running
- Redundant Power Supplies
  - Plugged into separate breakers!
  - Battery backup
- Redundant Storage
- Redundant Network

Redundant, Dedicated Storage

- Full Mirroring
  - Striping can increase speed but at a cost
  - Parity RAID modes like 5/6 are less reliable and cost write speed
- Redundant RAID Controllers
  - What happens if your RAID controller goes out?
- Dual Paths To Disk
  - Better throughput when everything is up and working
  - Protection against a failed controller
- No Shared Storage: a SAN is a “thing.”

True Girschywood Stories: The SAN Incident!

Redundant Network

- Two NICs
  - Two ports on one NIC won't help you: a NIC is a "thing"
  - Better throughput when everything is up and working

- Network "Teaming"
  - Each NIC is connected to a separate switch (a "thing")
  - Both listen on same address (driver handles the dups)
  - If a cable or switch dies, you're still connected
  - If configured in round-robin or load-balance mode, you double your network throughput

Redundant Servers

- What if the whole server fails?
  - Non-recoverable CPU failure
  - Power surge
  - ID-10-T error
- Second, identical server
  - Preferably in a separate rack
  - Connected to separate switches
  - On separate power breakers
  - With its own storage
- Physical hosts, not VMs!
- Each server is shared nothing

True Girschywood Stories: The VM Incident

Behind Redundant Connection Routers

- Without a Connection Manager (CM), how will clients know which host is Primary?
- But if you've only got one CM, *it* is now a single point of failure
- CMs should *not* be on the same host as the DB
- CMs *do* make sense to virtualize
  - *However*, the CMs should be on separate VM clusters
  - As we just discussed, the VM cluster is still "a thing!"
- With multiple CMs, failover becomes an issue
  - Make sure each CM has a unique PRIORITY value
  - If you have multiple CMs from the same host/VM, only one should have failover enabled

Review So Far

- We've got internally-redundant hosts with redundant power
- Each connected to dedicated, redundant storage
- Each connected to redundant switches
  - With redundant cables
  - From redundant NICs
- Behind redundant connection routers
  - Running on redundant hosts or VM clusters

What Are We Missing?

Redundant Data Centers

- The data center is a “thing,” and can fail!
  - Lightning Strike
  - Catastrophic Weather Event
  - Long-term Power Failure
  - ID-10-T error

- Second, identical data center (DR site)
  - Repeat everything we’ve done above at a second site
  - Should match as closely as possible
  - With redundant connections between data centers!
  - Location Matters!

Review So Far, Redux

- We've got internally-redundant hosts with redundant power
- Each connected to dedicated, redundant storage
- Each connected to redundant switches
  - With redundant cables, from redundant NICs
- Behind redundant connection routers
  - Running on redundant hosts or VM clusters
- With redundant data centers
  - In sufficiently distant locations

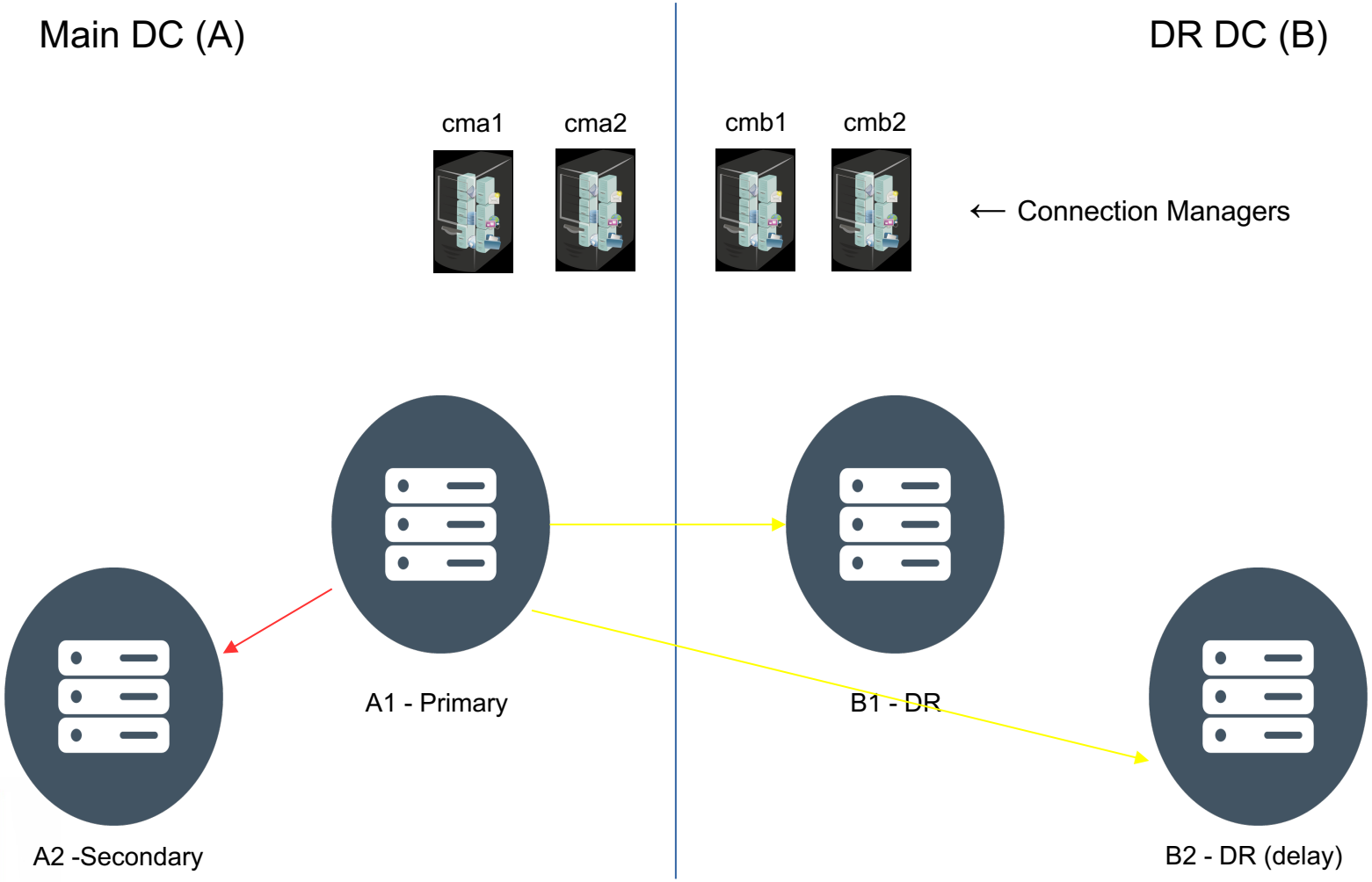
What Are We Still Missing?

True Girschywood Stories: The WAN Incident!

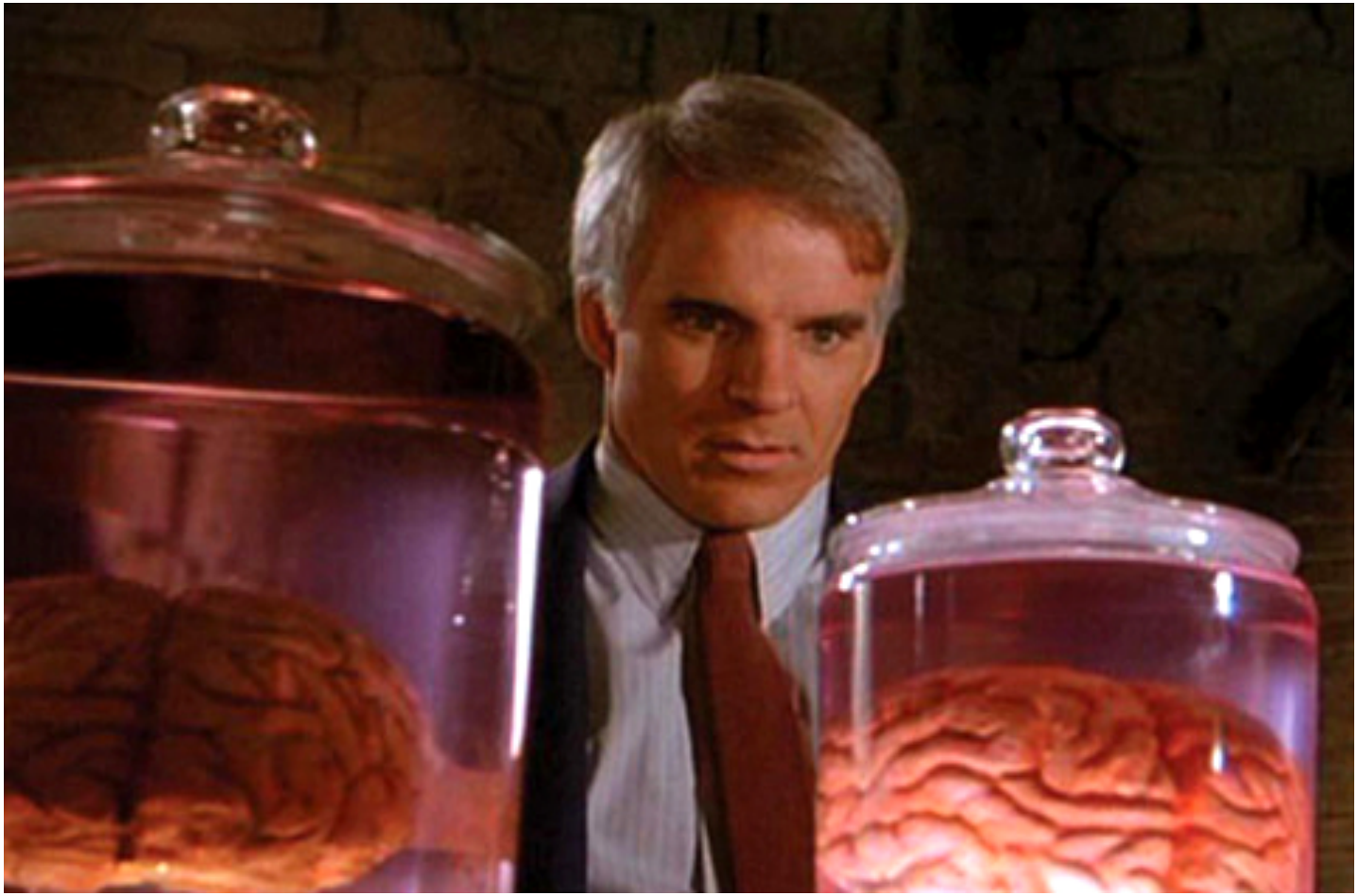
Failover!

- Your redundancy is only as good as your ability to use it!
- Can you successfully fail from one host to another with minimal interruption of service?
- Can you successfully fail from one *data center* to another with minimal interruption of service?
- The more you obscure all this redundancy from your clients / customers / users, the better
  - They should neither know nor care what the cluster looks like
  - They shouldn't have to change anything at all when a failover occurs, other than *maybe* to disconnect and reconnect

Redundant Configuration Review (So Far)



True Girschywood Stories: Automated Failover



How Automated Failover Works

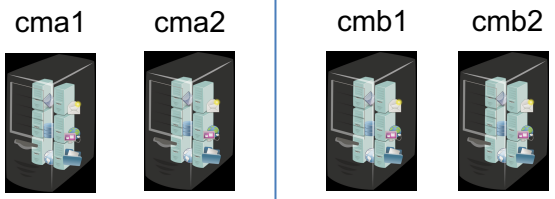
- Highest-priority CM “wins” – it’s the “failover arbiter”
- If highest-priority CM loses contact with primary:
  - Checks secondary; if secondary can still see primary, it does nothing
  - If secondary cannot see primary, it promotes the secondary
  - If it can’t see secondary either, it works its way down through the nodes in HA_FOC_ORDER
- So far, so good. But what if you lose your data center link?

DC Link Lost (1)

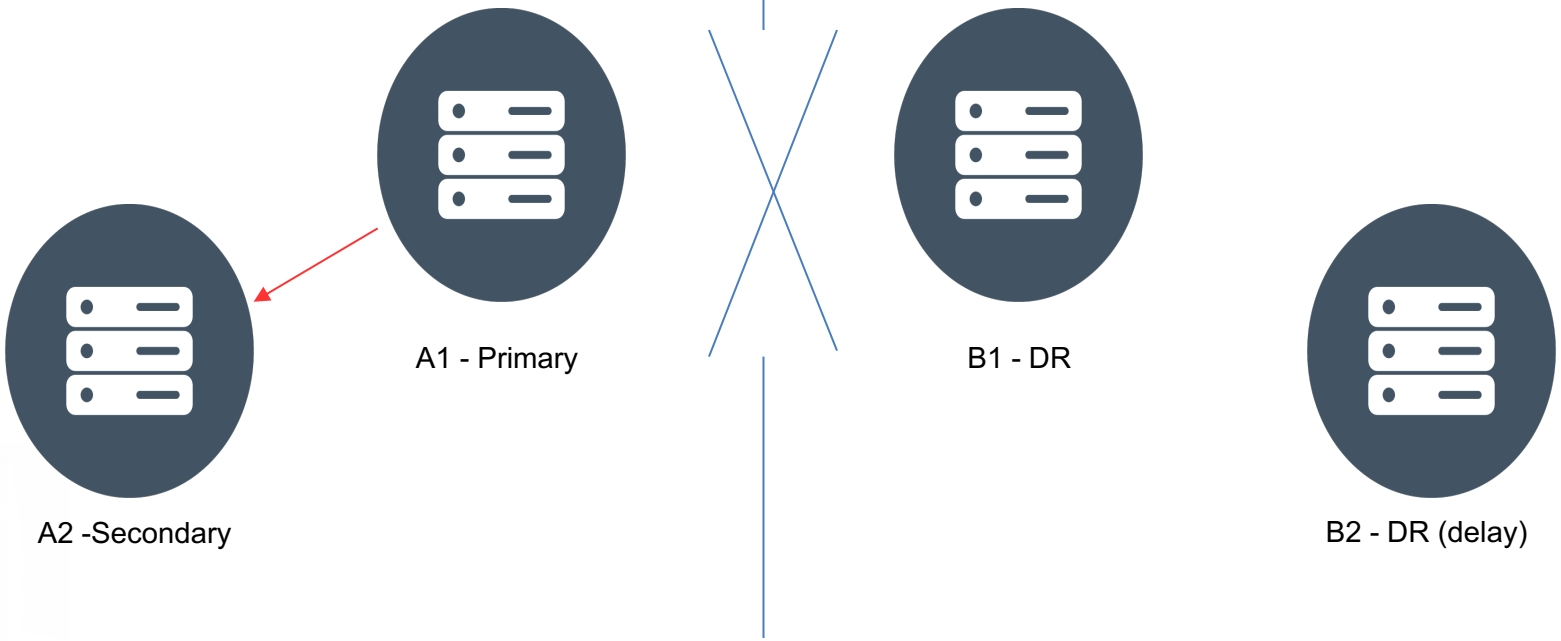
Main DC (A)

DR DC (B)

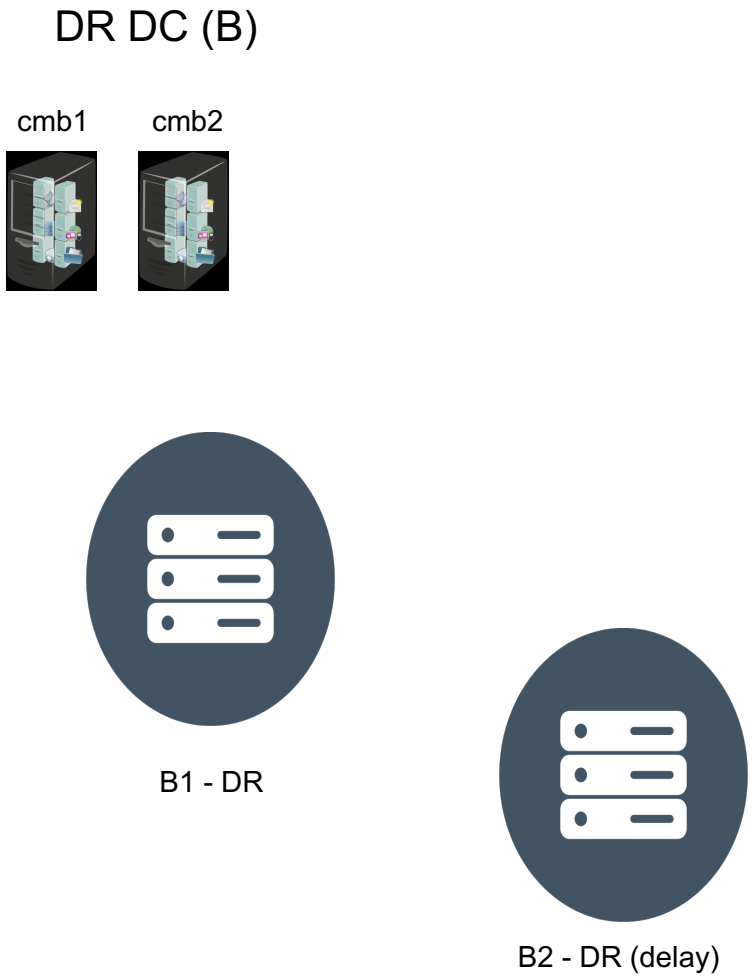
Everything's Fine Here!



But What About Here?

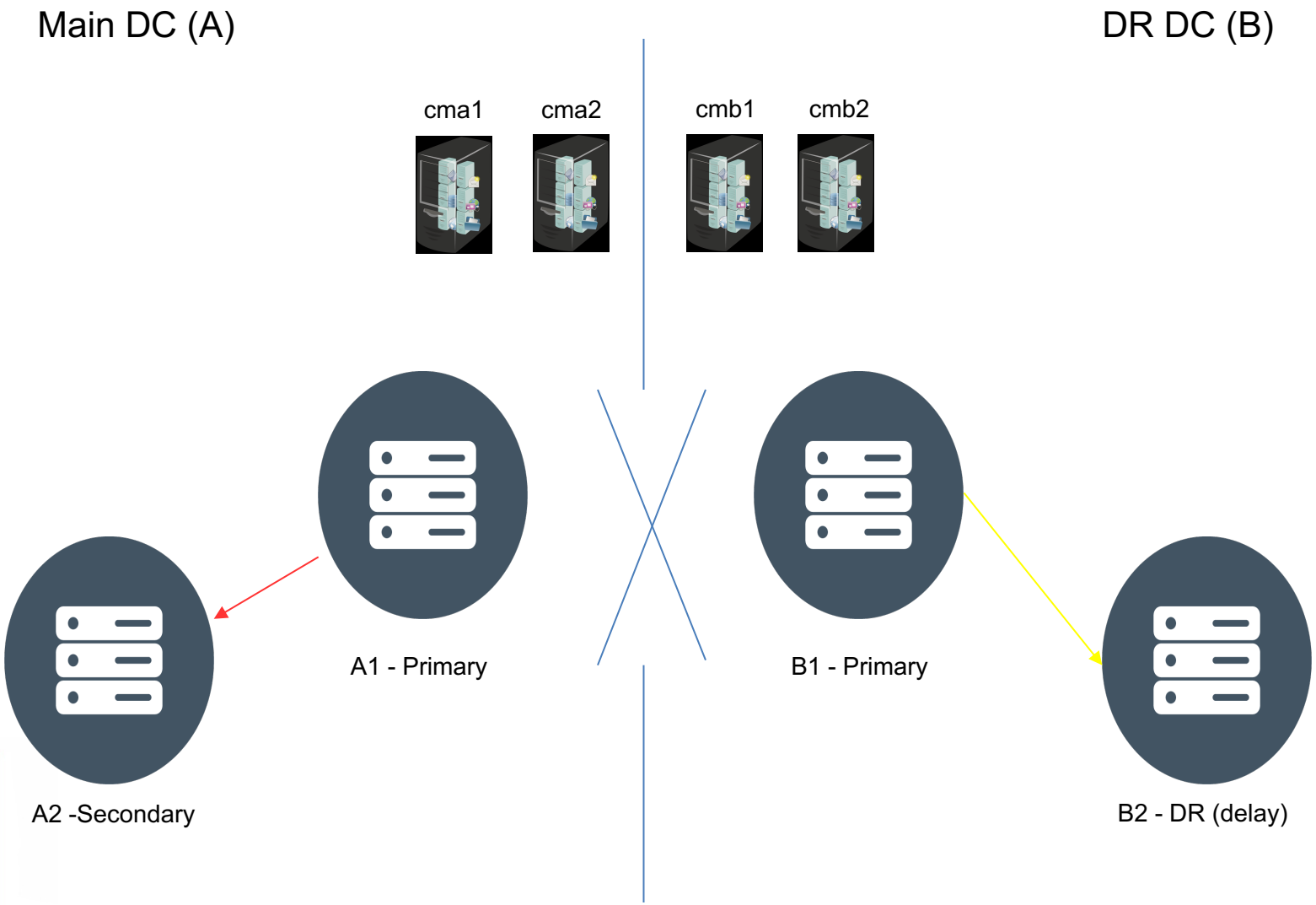


DC Link Lost (2) – DC B's View of the Universe



- All contact to DC A lost
- CM cmb1 is now highest priority
  - cmb1 can't see primary
  - cmb1 can't see secondary
  - cmb1 *can* see RSS
  - Asks RSS: Can *you* see primary?
  - RSS says "nope."
  - cmb1 says, "Welp, better initiate failover!"
  - DB node b1 – formerly RSS – becomes primary

DC Link Lost (3) – Overall View of the Universe



Automated Failover Lesson

- In a multi-data center setup, you should only enable automated failover at the primary data center
- That means failing over from one data center to another requires manual intervention
- Is multi-DC auto-failover feasible? Stay tuned.

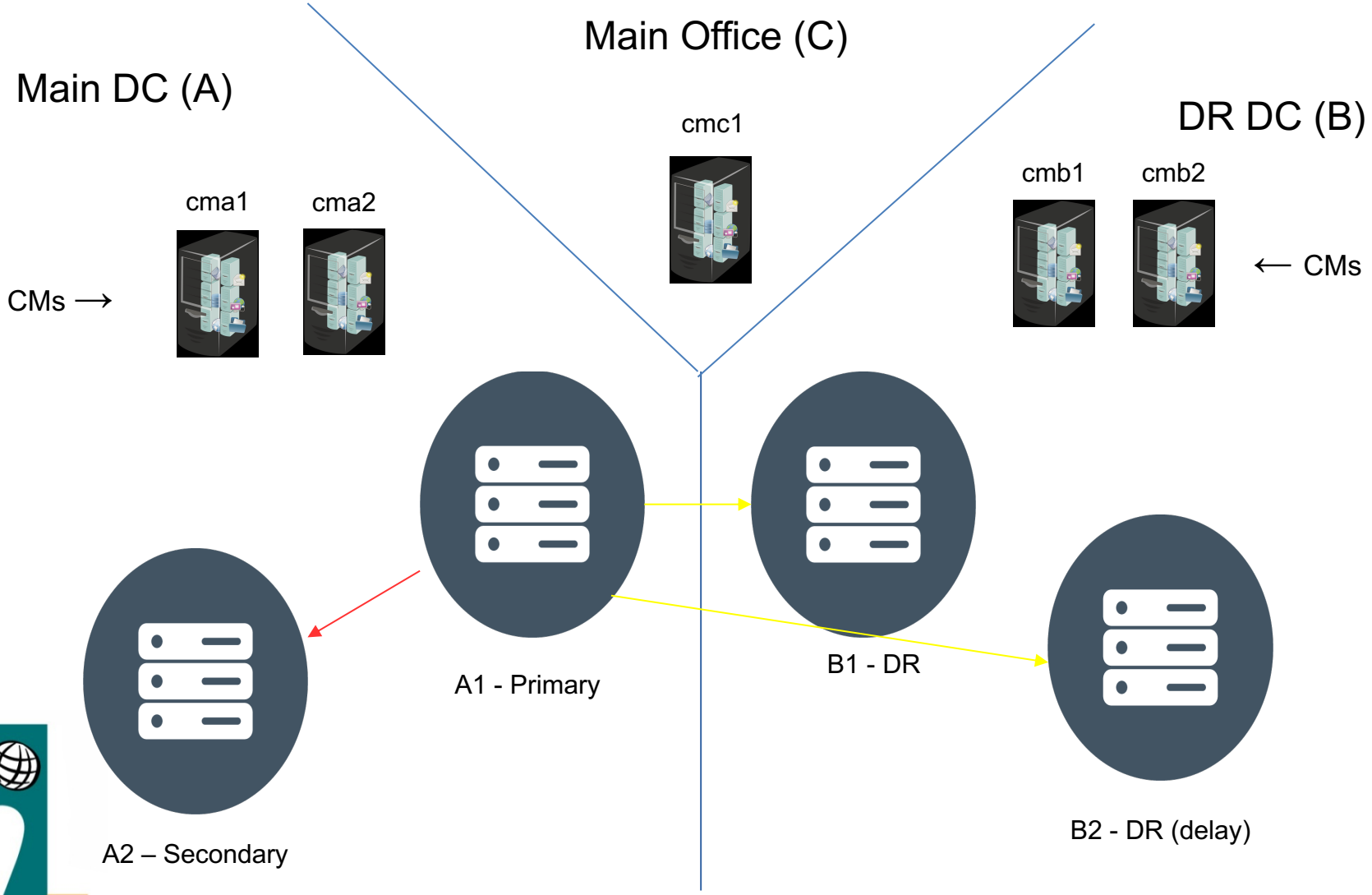
Management Issues

- Are your secondary / DR servers ready to become Primary?
  - Are all the appropriate user accounts set up?
  - Are all the necessary scripts present?
  - Are cron jobs configured?
    - Are they set to run only on the primary, when appropriate?
  - Do configurations (both physical and software) match?
- **You should have procedures in place to make sure that when you update one node in a cluster to add or change any of the above, you update them all**

Three Is the Magic Number

- Other databases (e.g., MariaDB/Galera, MongoDB) don't use priority for failover
- Instead, they use a majority vote algorithm
  - Requires an odd number of nodes (at least three)
- Ideally, this requires a third data center
  - Two DCs with three nodes each = six nodes; there can be "ties"
  - Three DCs with three nodes each works, but seems overkill
  - Can do 3 + 3 + 1; bonus is that the third DC doesn't have to be fully built out (Informix: 2+2+1)

Final Configuration



Let's Talk Licensing

- All that hardware is already expensive, but now we have to license the software
- Per IBM/Informix license agreement, you must license any engines that take user transactions
- If no users connect (i.e., true DR server), no license is required
- What's more important to you?
  - The ability to distribute your production workload?
  - The need to keep licensing costs down?

Repeat the Mantra!

Eliminate Single Points of Failure!

We've Still Missed One! (At Least!)

Arguably The Most Important Single Point of Failure

Thank You

Informix Tech Talks by the IIUG on YouTube

We are launching a new channel on YouTube for Informix Users! Please subscribe to our channel on YouTube to stay informed. This will be a place for Informix how-to videos. More videos will be coming soon.

Subscribe at:

https://www.youtube.com/c/InformixTechTalksbytheIIUG