UKSUG

SAP DATABASE & TECHNOLOGY USER GROUP

Headline sponsor

MLOGICA powered by aws

TechSelect 2019

SAP Database & Technology User Show

Sponsored by SAP

HADR isn't that HARD

A case study & lessons learned

Joe Woodhouse, Prima Donna Consulting

September 10, 2019

Contents

A mercifully brief history of DR & HA solutions from Sybase & SAP

A pleasingly short review of the ASE HADR architecture

Some disturbing changes if you are an expert Replication Server DBA

HADR Limitations

Lessons learned

Impossible client requirements, and transcending the limitations

Q &A if I haven't talked too long

A brief history of DR and HA through the ages

Disk mirroring and RAID

ASE log shipping

Replication Server Warm Standby

ASE HA + OS clustering

Replication Server MSA

ASE + hardware disk replication

ASE log shipping + online … for standby_access

ASE CE

ASE + OS clustering (+ hardware disk replication?)

The issues they were trying to solve (in no particular order)

Hardware risks – I need a second server, uninterruptible power, and redundant disks

Corruption risks – I need a logical copy of my data that isn't a physical copy

Mean time to recover – I can't afford to wait to boot a server and ASE

Sunk costs – I'd really like to be able to put my idle DR hardware to use

Datacentre risks – I need to survive loss of a building's infrastructure (or the building itself)

DR distance – I need to survive loss of district/city power (or the city itself?!)

Performance impact – I can't have my Production impacted by my DR

Latency – I need my redundant database(s) to be up to date

Automation – I need it to take care of itself and Just Work

Zero data loss – I can't afford to lose a single committed transaction

Moral of the story

Some of these goals are opposed

- Logical copy probably imposes latency
- Zero data loss requires synchronous copying which harms Production performance

Some increasingly elaborate combination solutions could (mostly) overcome these trade-offs

- e.g. ASE + RAID + OS clustering + SRDF disk replication + SRS MSA to long distance DR
- Each new technology added to the stack increases the costs
- … and increases the complexity (which is itself a risk when most disasters are caused by human error)

Even combination solutions still can't tick all the boxes

- Single points of failure: shared disk (OS clustering), SRS itself (Warm Standby, MSA), etc.
- The holy grail is zero data loss with zero latency and zero Production impact
  - … oh and I'd like to be able to use my DR for reporting
  - … and maybe DBA housekeeping
  - … without compromising latency

ASE HADR “Always On” to the rescue

Promises to have the least trade-offs and tick the most boxes

- And to be fair does actually do what it says on the tin

Start with Sybase Replication MSA

- We know it; it works; logical copy; destination host and ASE are already booted

Use Replication CI mode instead of standard (“legacy”) Rep Agent processing

- Much faster; less intrusion on Production; lower latency; many OS native threads

Make it synchronous (same datacentre only)

- This is new to Replication Server – can now offer zero latency and true zero data loss

Tune it out of the box (and require some very fast disk hardware)

- So synchronous operation doesn’t impact Production

Make it automated (via optional but critical Fault Manager)

- Also new to Replication Server

ASE HADR – does it really work?

If you follow all the rules, yes it does

Zero data loss = synchronous

- If you follow all the rules, this won't impact Production
- Yes really

So what are the rules?

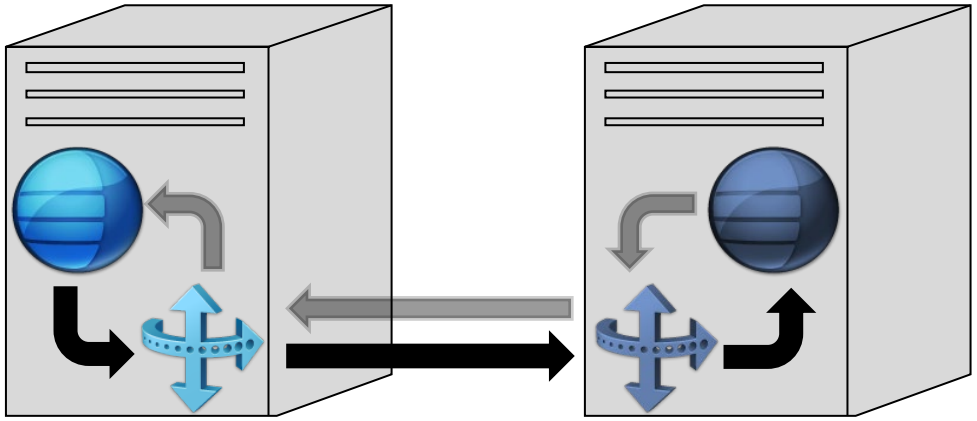
- Fast hardware for the new SPQ (Simple Persistent Queue) used for CI (Canonical Interface)
- They aren't kidding, go buy PCIe/NVMe disks for SPQ at a minimum, and preferably all SRS partitions
- 10+GbE NICs, ideally one set dedicated to HADR nodes
- Read all the documentation and follow all the rules for OS memory/disk/network tuning
  – … even if it makes your Sybase boxes special snowflakes, ruining your OS configuration management
- You can no longer get away with not having a PK or unique index on *every* table
  – Forget any hope of zero data loss without this

Same technology but a whole new world

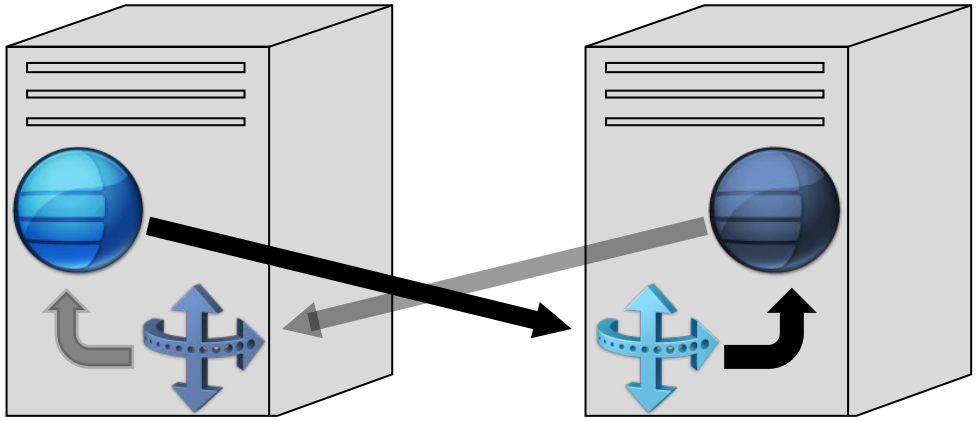
OK so you're a Replication Server expert

- … you're in trouble now

You're familiar with this architecture:



But HADR is like this:



Each ASE connects to and is managed by the SRS on the *other* node

This changes everything

Each ASE connects to and is managed by the SRS on the *other* node

- Repdefs and subscriptions are in the “wrong” place
- If you connect to the Replication Server on the same host, “threads are down” (no they’re not)
- All your habits about suspending and resuming threads are exactly wrong for HADR

It’s even worse than that

- HADR gives both ASEs and SRSs internal names
- In fact it gives the ASEs two names each! (to handle connections in each direction)

Actually it’s even worse than *that*

- You no longer have a choice… you have to start using the RMA interface
- In fact a lot of the commands you’re used to running really should be run in the RMA instead
- The HADR Replication Servers now live in $SYBASE/DM, not $SYBASE/REP-16_0

Anything else different??

The HADR Replication Servers now live in $SYBASE/DM, not $SYBASE/REP-16_0

Actually they need to be defined as their own $SYBASE

- So you have two different SYBASE.sh files on each node
- This is a problem if you're flipping back and forth… think about it…
  - LD_LIBRARY_PATH="$SYBASE/[whatever]":$LD_LIBRARY_PATH
  - Environment variables will grow and grow and eventually break
- The answer is to fully reset the variables in each SYBASE.sh

ASE's interface file is full of (hopefully) sensible names as chosen by you

- HADR SRS's interfaces file has all of the HADR internal names
- The answer is to add each file's entries to the other file

Licenses also need to be copied from $SYBASE to $SYBASE/DM

Strongly recommend sensible names and many command aliases!

But it’s not all bad

The RMAs take care of most things for you

- If you run RMA commands it sorts it all out and runs everything in the right place
- In fact you can fully resync a database from Primary to Standby in just three (!!) RMA commands
  - Yes it does the database dump and load for you
  - Admittedly this is probably too slow for VLDB
  - The VLDB solution is beyond the scope of this presentation
  - Hint: dump database + multiple dump tran, final dump tran uses dump marker
- RMA understands what is supposed to be up and down

Seriously, consult the documentation before you resume any suspended thread

- Then double-check which Replication Server you’re in

Limitations

All versions must match *exactly* within HADR

- Everything: ASE and Replication Server versions, character sets, sort orders
- Yes everything: OS version & patches
- They're not kidding: CPUs & the same FPU implementation on the CPUs if you use *float* or *real* datatypes

HADR *must* be its own replication domain

- Its own IDserver
- Can't be used for any replication other than with the HADR pair

You are asking for trouble if you run any version below 16.0 SP03 PL06

- No "3rd node DR" or "external replication" until PL05
- No SQL statement replication or excluding tables until PL06
- And PL07 fixes a few important bugs
- … Just use PL06 or PL07

More limitations

HADR cannot be part of a Replication Server Warm Standby

- It can however participate in MSA as source, destination, or both
- More on this later

HADR not supported as part of an OS cluster

- It's complicated; it has to do with two different failover managers and potential for "split brain"

HADR cannot use MPR multi-path replication

- The intention is that CI Rep Agent replaces MPR anyway

Rep Agent recovery mode

- Hands up anyone who's done this ever

Stored procedure replication the old way – pretend it's a table

Object-level repdefs within HADR (external replication OK)

A few more limitations

HADR is strictly two nodes only… for synchronous mode

- You can do two-node synchronous within the datacentre, and async to long distance "3rd node DR"
- This is less exciting that you think
  – More on this (and "external replication") next slide

No stored procedure replication within HADR

- But can do stored procedure replication with external replication

Automation is not (ahem) installed automatically

- You must separately install and configure the Fault Manager
- It's not very user-friendly

Everything assumes you will use DBA Cockpit to manage the system

- You don't have to though
- And really, you probably *shouldn't* as you don't have version control without scripts
  – And that means command line FTW

But I want to do synchronous zero data loss between datacenters! (✹)

Previous slides made it sound like synchronous HADR can *only* be done within a datacenter

- "It depends"

Synchronous HADR *can* be done with a node in two different datacenters

- It all comes down to latency, because the round-trip latency will be added to every COMMIT in Production

A good rule of thumb:

- If Production can cope with synchronous disk replication, it can probably cope with synchronous HADR
- … and for all the same reasons

There are no t-shirt sizes about maximum distance or maximum latency

- Dark fiber between two datacenters in the same city: most sites can achieve this without issues
- Dedicated & isolated dark fiber between different cities: some sites have achieved this
- <2ms = no problem; 2-5ms = probably OK; 5-20ms = performance impact; >20ms = probably not OK

(✹) This slide added after the presentation because three of you asked me about this

More about synchronous HADR between datacenters (✹)

Much also depends on your workload

- Single-threaded batch jobs (using set-based processing) don't care about even 10s latency
- Highly concurrent OLTP is in serious trouble with even 10ms latency

Remember that HADR will drop back from synchronous to asynchronous if latency "too high"

- You can define what is "too high" – this is configurable

However! Beware on all versions prior to ASE 16.0 SP03 PL07!

- Before PL07, HADR dropped back to async mode *silently*
- As of PL07, HADR will now stamp a message in its logs letting you know that this has happened
- Monitor your logs!
- Patch to PL07!

(✹) This slide added after the presentation because three of you asked me about this

But I want more than two nodes

HADR 3rd node

- Synchronous HA local cluster
- Asynchronous long distance DR
- DR node is known to HADR and managed by it
- Resync to DR *can* be done by RMA
  - But probably too slow for any serious database(s)
- Cannot failover to DR
  - DR cannot be promoted to primary or even standby

HADR + external replication

- Synchronous HA local cluster
- If replicating out of HADR… asynchronous MSA
- External replication is unknown to HADR
- Resync in any direction cannot be done by RMA
  - Resync is greatly complicated by HADR
- All external replication sees HADR as one ASE
  - Maybe not instantly obvious but that includes both Replication Servers as part of HADR
  - Just treat the entire HADR as one ASE
  - HADR is its own Replication domain
- This gets a bit weird when replicating out of HADR
  - The entire HADR gets a Rep Agent
  - SPQRA, and it changes a few things
  - Read the documentation

I hope I can save you my pain

Replicate the master database within HADR (and if using it, to external replication)

- Everything is much easier if logins are always completely in sync
- Get ready for a lot of scripted bcp's otherwise

Maybe read up on Rep Agent CI or "streaming mode"

- Many previous SRS threads are now external OS threads
- This means they don't have a *spid* in "admin who" output
  – Lack of a spid used to indicate a thread was down, but not any more

Start-up and shutdown things in the correct order!

- Start-up:
  – ASE2 (yes the standby), SRS2, ASE1, SRS1, RMAs x2, DBA Cockpit x2, Fault Manager
- Shutdown
  – FM, DBA Cockpit x2, RMAs x2, ASE1 (yes the primary & use "shutdown with wait nowait_hadr"), SRS1, SRS2, ASE2 (yes the standby last)

What else did Joe lose sleep from

There is a new rs_ticket_history table in every database known to HADR

- This grows and nothing will ever delete or truncate it, so you must do it “often enough”
- This is especially a problem in the master database
- This is even more of a problem in master..syslogs because all those inserts are logged

While most RMA “sap_%” commands are synchronous, some are not

- sap_status will report on the most recent asynchronous command…
- … only on the most recent and not any older
- … and on the “most recent” even if that was a week or more ago

Use resource files for everything – I had to rebuild a few servers

External replication *must have* the ASE_SECDIRS license

- (aka ASE_SECDIR2 aka ASE_PRIVACY) license
- If you don’t have it and won’t buy it then you need some truly gnarly workarounds

Documentation bugs? That never happens!

Documentation incomplete

- It says ER maint login must not have sa_role, sso_role, mon_role, and some HADR permissions
  – But fails to mention that it also must not have oper_role
- It has a good checklist for replicating out of HADR
  – But fails to mention creating the maint user in the external database
  – And also fails to mention that this will fail unless you are replicating master or syncing logins
- It appears to say it's OK to not have PK or unique index on every table
  – It says performance will suffer
  – But fails to mention you can't do synchronous zero latency without it
- It recommends fast disks for SPQ
  – But fails to mention you probably can't achieve synchronous HADR without PCIe/NVMe

Documentation bugs? That never happens!

Documentation entirely silent on some points

- External replication DSIs to HADR *must* have dsi_replication_ddl = ‘on’
- DDL statement replication (within or to/from HADR) DSI needs deferred_name_resolution = ‘on’
- Doesn’t discuss why ASE_SECDIRS and “enable granular permissions” is so essential to external replication
  – Very briefly: cannot alias maint user to dbo; it needs sa_role but is not allowed sa_role

Work would be so much easier without clients…

A client's requirements:

- HADR on Solaris
- External replication to AWS
- Cannot have any outage in Production, at all
- Replication delivery to AWS must be deliberately delayed to allow recovery from human error
- Databases must be encrypted

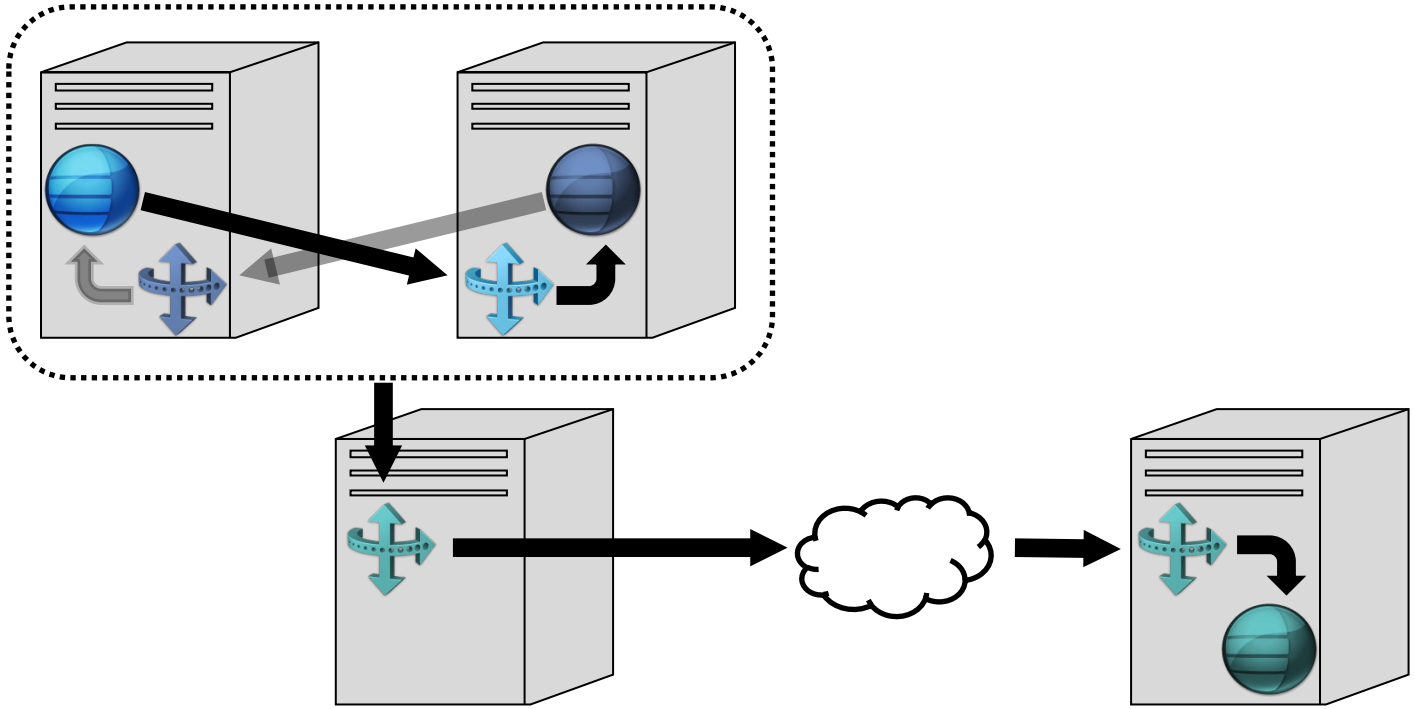
Why is this a problem?

- HADR + external replication cannot cross platforms
- Even if it could, think about resync to AWS…
  – XPDL needs an outage in the source database

SAP said it can't be done

- Opinion was divided

Eureka! Solution version 1



It is true that HADR + ER can’t cross platforms

But there is nothing stopping us having an intermediate point

HADR on Solaris, external replication to Solaris, replication from ER to AWS

Eureka! Solution version 1

It is true that HADR

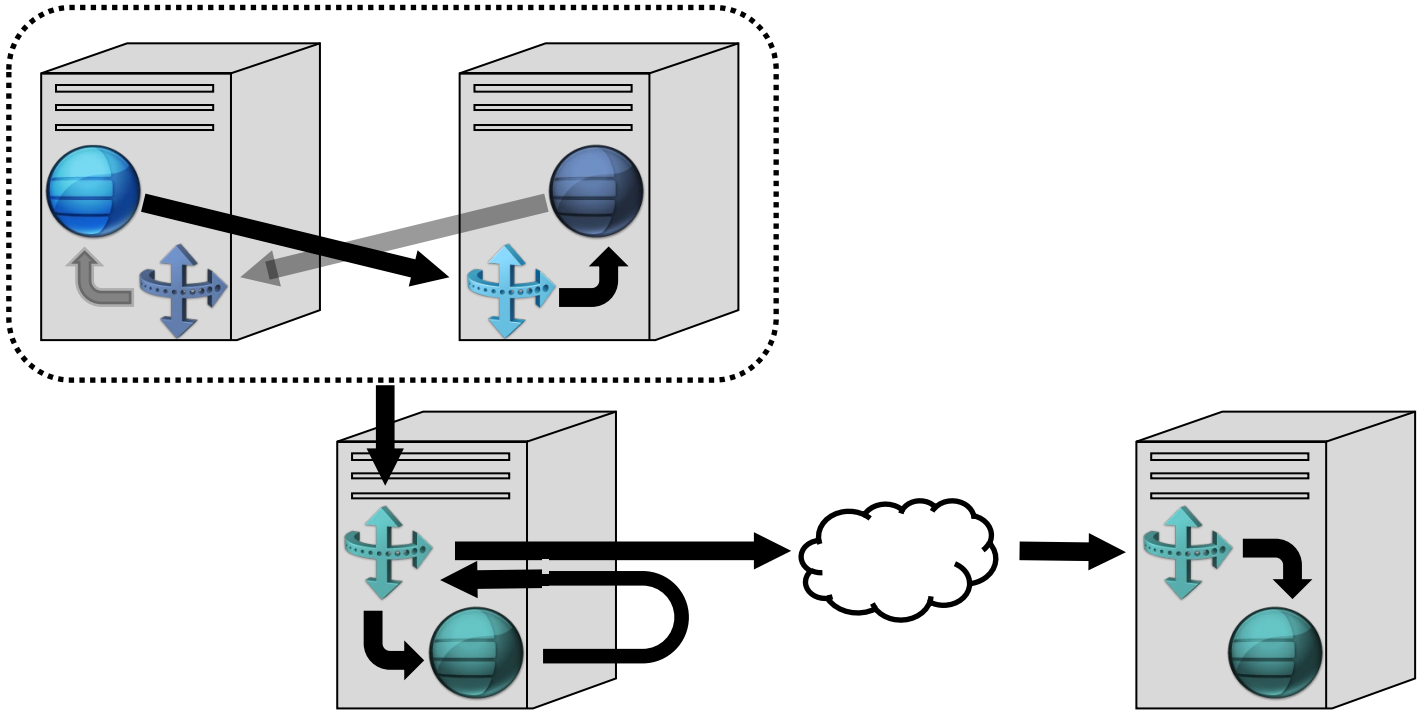
But there is nothing stopping us having an intermediate point

HADR on Solaris, external replication to Solaris, replication from ER to AWS

Not quite, back to the drawing board, version 2

Doesn’t deal with XPDL needing an outage we’re not allowed

OK, add a third Solaris ASE



Version 2

Resync now decoupled from AWS

- Resync HADR to external replication on-premises ASE (“ASE3”)
- This is same platform, so fully online and no outage in Production
- HADR adds some complexity but this is well documented

Allow replication to catch up to ASE3

Suspend replication to ASE3

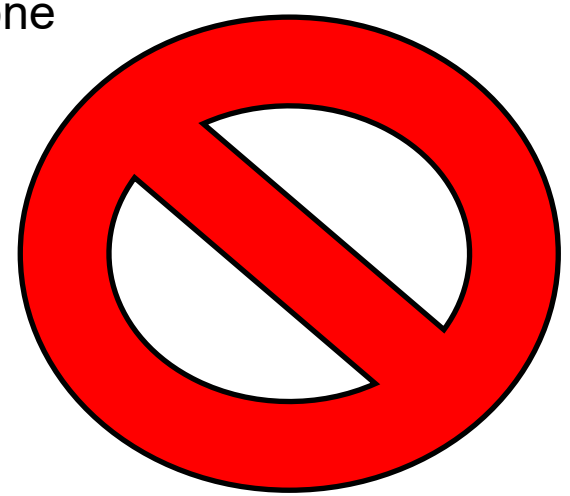
Once ASE3 is quiet, resync to AWS

- This is XPDL but only needs outage in ASE3, not in HADR
- Queues will grow during slow XPDL to the cloud
- Need substantially more queue space
- Need good HADR tuning, WAN tuning to AWS, and replication tuning across WAN

Never mind the intermediate version 1.5

This involved some wacky stuff

- Replicate to AWS and then back again to ASE3
- Hack the RS interfaces file in AWS to repoint to AWS ASE once XPDL resync was done



Intention was to reduce queues and speed things up

- But roundtrip replication “there and back again” was too complex

Version 2 is much cleaner

- It is effectively now two separate resyncs
- HADR to external replication: well understood, and doesn’t know or care about AWS
- Solaris replication to AWS RHEL: well understood, and doesn’t know or care about HADR

All requirements met, everyone happy

- Well, almost…

Ah, that's right, encryption

This part of the solution is on hold while we investigate further

HADR is fine with encryption, and key changes will replicate (if you replicate master)

HADR to external replication is fine with encryption

- RS has some restrictions with encrypted PKs but it's doable
- Issue: CI streaming replication not supported cross-platform with Endian swap

Replication cross-platform… not clear if that works

- Possible issue: SRS replicates cypher text, and if encryption is platform-dependent this won't work
- Possible workaround: replicate with msg_encryption, SSL, and disk encryption both sides

XPDL with encrypted database… not clear if that works

- Possible issue: Endian conversion needs plaintext not cyphertext
- Possible workaround: decrypt prior to XPDL resync, re-encrypt both sides after resync
  – … assuming cross-platform replication of encryption is OK

A last word about HADR futures (✹)

Disclaimers!

- I don't work for SAP and don't control their Engineering priorities
- All this based on hearsay
- This is *not* a prediction, let alone a promise, let alone a commitment from SAP
- Timelines totally unknown

HADR futures (maybe)

- Cross-platform external replication (making my entire solution unnecessary; oh well)
- On-premises HADR with external replication to the cloud: supported out of the box by SAP (ditto)
- Cross-platform CI-mode streaming replication (faster cross-platform external replication)
- Faster & more automated failover of HADR when replicating out to external replication
  – Currently the SPQRA must manually be dropped & recreated

(✹) This slide added after the presentation because three of you asked me about this

Questions and thank you

Joe Woodhouse

Prima Donna Consulting

joe.woodhouse@primadonnaconsulting.com