



Education

Trends in Application Recovery

Andreas Schwegmann, HP

- The material contained in this tutorial is copyrighted by the SNIA unless otherwise noted.
- Member companies and individual members may use this material in presentations and literature under the following conditions:
 - ◆ Any slide or slides used must be reproduced in their entirety without modification
 - ◆ The SNIA must be acknowledged as the source of any material used in the body of any document containing material from these presentations.
- This presentation is a project of the SNIA Education Committee.
- Neither the author nor the presenter is an attorney and nothing in this presentation is intended to be, or should be construed as legal advice or an opinion of counsel. If you need legal advice or a legal opinion please contact your attorney.
- The information presented herein represents the author's personal opinion and current understanding of the relevant issues involved. The author, the presenter, and the SNIA do not assume any responsibility or liability for damages arising out of any reliance on or use of this information.

NO WARRANTIES, EXPRESS OR IMPLIED. USE AT YOUR OWN RISK.

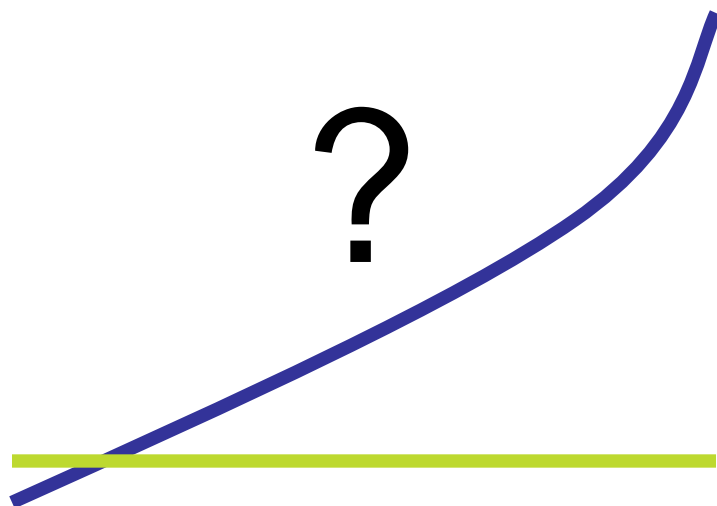
➤ Trends in Application Recovery

- ◆ This session will appeal to Data Center Managers, Backup Administrators, Application Administrators and those that are concerned how data lost in an application or database could be recovered.
 - › Challenges and trends in application backup and recovery.
 - › You will be guided through all layers from the service down to the physical hardware.
 - › You will learn how to recover individual lost pieces of information up to the recovery of the entire application distributed across complex and virtualized environments.
 - › You will get some ideas how to deal with the human factor in IT environments with distributed responsibilities.
 - › Finally the session discusses how to balance service levels against cost.

Trends in Application Recovery



- Challenges & trends
- What to recover and by whom?
- Application backup
 - ◆ Consistency
 - ◆ Backup window
- Interfaces
- Recovery
 - ◆ Application entirely
 - ◆ Single items
- Virtualization & Cloud
- Conclusion



- ▶ Blue line – exponential data volume growth & complexity growth = the bullets on this slide
- ▶ Green line – IT budget is flat

▶ Exponential data growth

- ◆ Cheaper storage
- ◆ More performance
- ◆ Long term retention
- ◆ Duplicate data

▶ Complexity growth

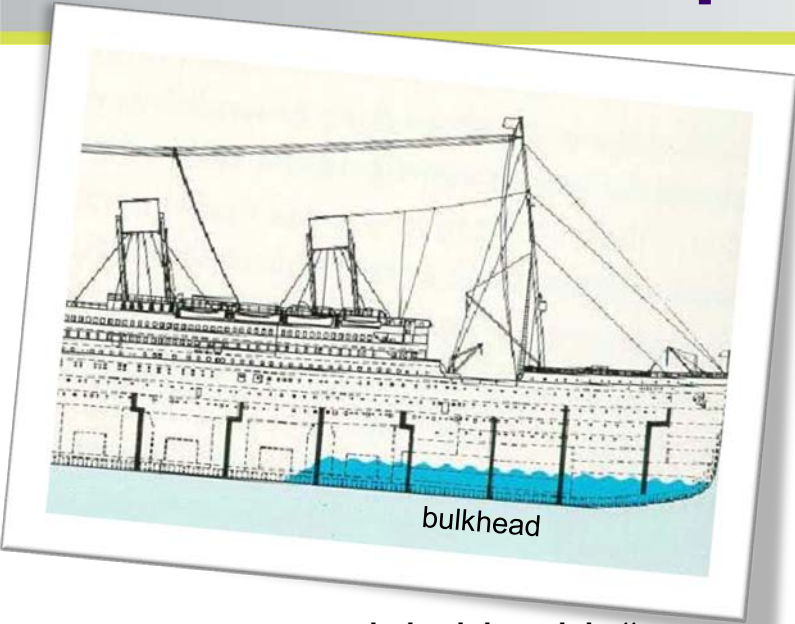
- ◆ Scale up & out
- ◆ High availability
- ◆ Virtualization
- ◆ Cloud – „EaaS“ (Everything as a Service)
- ◆ Search
- ◆ Security & compliance

Where Does Recovery Live?



- Application resilience
- Application build in backup tools
- Templates
 - ◆ „Golden“ images
 - ◆ Virtual Machines
- Application specific 3rd party tools
- Storage specific tools
- Backup software
 - ◆ Across multiple applications & storage
- Operations management software
 - ◆ Across all IT
- The cloud

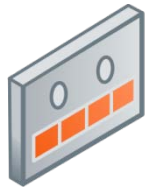
The “No Backup” Trend



„unsinkable ship“
vs.
„rescue boat“

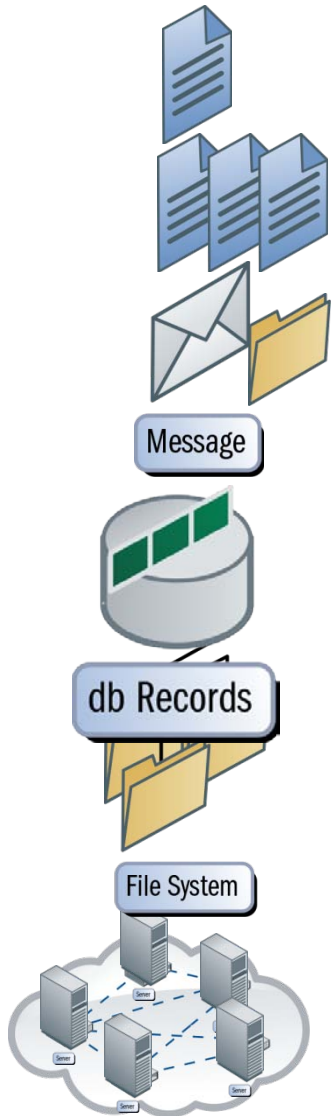


- Application build in database replication
 - ◆ Optional automatic fail over
 - ◆ Lagged databases
- Transaction logging & replication
- Self healing
 - ◆ Consistency ckeck and repair
 - ◆ Defect page detection and repair
- Versioning
- Dumpster & 2nd level Dumpster
- Build in archive



- Different backup to disk service levels
 - ◆ Full copy backup versus backup of changes
 - ◆ Same disk quality as original versus cheaper disk
 - ◆ Retention: days versus months versus years (archiving)
- No performance degradation during backup
 - ◆ Backup from replica & proxy server
- Recovery to any point in time
- Data reduction
 - ◆ Deduplication, primarily of repeated full backups
 - ◆ Incremental and consolidation of incremental backups
- Seamlessly link disk with tape
- Recovery automation & simplification
 - ◆ The know how to recover is build into the software
- Server farm down to single document, mail, ...
 - ◆ Catalog browsing -> Index based full content search

What to Recover?



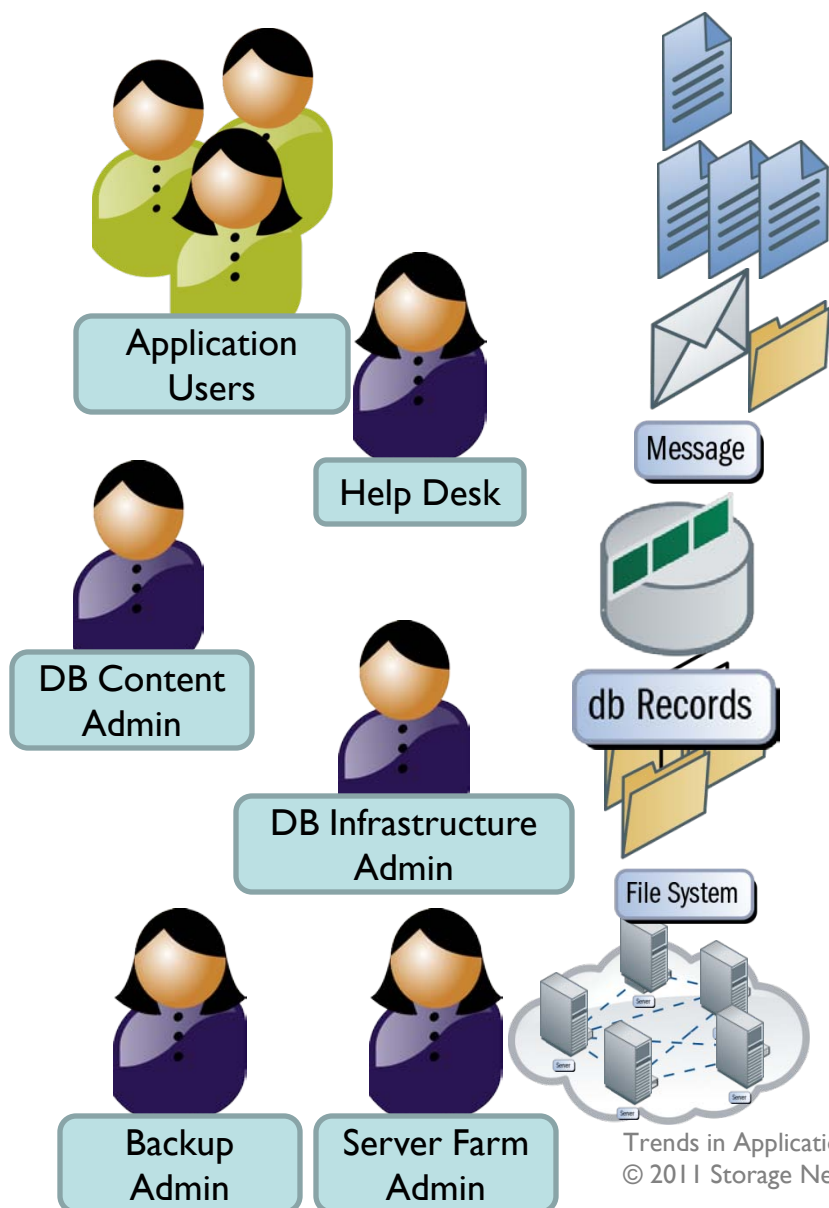
- Single items / end user domain
 - ◆ Files & Directories / Libraries
 - ◆ Entry, record, transaction
 - ◆ Document, e-mail, blog
 - ◆ Calendar, tasks, contacts
 - ◆ Table, list, tree, folder, wiki
 - ◆ E-mail box, user site
- Database
- Search index & services
- Application configuration
- Binaries, OS, configurations
- Physical servers
- Virtual servers
- Server farm
- Disk array

Use Cases Beyond Recovery



- ◆ Testing
 - ◆ Recovery
 - ◆ Development test data
- ◆ Data Migration
 - ◆ On premise -> cloud
 - ◆ Cloud -> on premise
 - ◆ Service provider A -> B
- ◆ Deployment
- ◆ Data warehousing
- ◆ E-Discovery
- ◆ Forensic analysis
- ◆ Archiving
 - ◆ Tax review, ...

Who Does What?



- Different users groups use different user interfaces
 - ◆ Users and administrators don't want more tools, they want to manage from their tools.
- Security
 - ◆ User A should not be able to see data from user B.
 - ◆ The administrator should not be forced to break into the end user privacy.
 - ◆ DB administrator might not have the right to restore from backup.
- Processes
 - ◆ Application administrators might not be connected with backup administrators.

How to reduce the backup volume?

➤ Full backup – file based or block based

◆ Database

- › Data files = “tables“ mapped to files
- › Control files to find data files & log files
- › Transaction log files, optional: move / delete

◆ Trend features

- › Storing data outside of the DB: Files, Binary Large Objects
- › Search Index, services, encryption keys, ...

➤ Incremental backup – changes since last backup

◆ File based

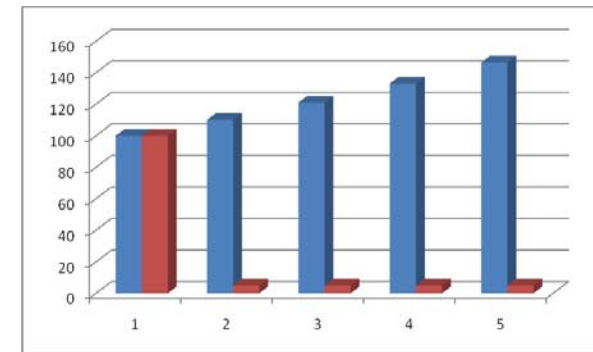
- › Transaction log backup and move / delete
- › Files: File system incremental backup of single instance file store

◆ Changed blocks

➤ Differential backup – changes since last full backup

➤ Compression

➤ Deduplication



➤ When is an application consistent?

- ◆ Data is valid at the same point in time
- ◆ Data is complete



➤ How to accomplish consistency for backup?

- ◆ Offline backup - application shutdown
- ◆ Crash consistent backup
= snapshot without interaction
- ◆ Online backup – application interaction

Consistency - Offline Backup

- Shutdown the application / database
 - ◆ Guarantees application consistency
 - ◆ All cache data copied to disk
 - ◆ All transactions closed
 - ◆ Optional: database consistency check
- Backup to another disk / tape
 - ◆ OR create a snapshot
- Optional: move / delete the transaction logs
 - ◆ Frees disk space
 - ◆ Enables incremental backup based on transaction logs
- Start the application
- Optional: backup the snapshot to another disk or tape
- Recycle the snapshot
 - ◆ Keep the last N snapshots
 - ◆ Snapshot rotation



Crash Consistent Backup

- Create a snapshot while the application runs
 - ◆ Consistency has the same quality as after a system crash
 - ◆ Most applications / databases can survive system crashes
 - › But some don't and some not always.
 - ◆ Recovery can not be guaranteed
- Use cases
 - ◆ 7 x 24 operations -> no backup window
 - ◆ Virtual Machine backup without agent or service API
 - ◆ Application lacks online backup mode feature
 - ◆ No resources for transaction logging during backup
 - ◆ Snapshots enable more points in time
 - › Might reduce the risk





- Database(s) are in “backup mode“ during backup
 - ◆ Data files don't change while in backup mode
 - ◆ Changes during backup happen in the cache and go into logs
 - ◆ After backup all changes are applied to the data files
 - ◆ Optional: backup of the transaction logs & delete logs afterwards
 - ◆ Optional: ongoing log file backup after database backup -> “CDP“
- Consistent search index
 - ◆ All databases need to go into the backup mode
 - ◆ Across the server farm
- Use cases
 - ◆ 7 x 24 operations -> no backup window
 - ◆ Guaranteed & fully supported consistent recovery

Application Backup Interfaces

- Application specific tools via GUI, CLI
- General purpose API
- Streaming backup API
 - ◆ Direct copy
 - › Access to in-memory copy of data, cached by the application
 - › Minimizes redundant memory copies
 - ◆ Incremental backup
 - › Access to changed blocks / pages or transactions
 - ◆ Optional features
 - › Granularity below database level
 - › Compression
 - › Encryption
 - ◆ Sequential access is optimal for streaming media



➤ Snapshot focused

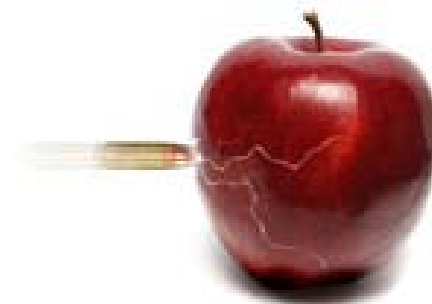
- ◆ Creates application consistent volumes ready for backup

➤ Use cases

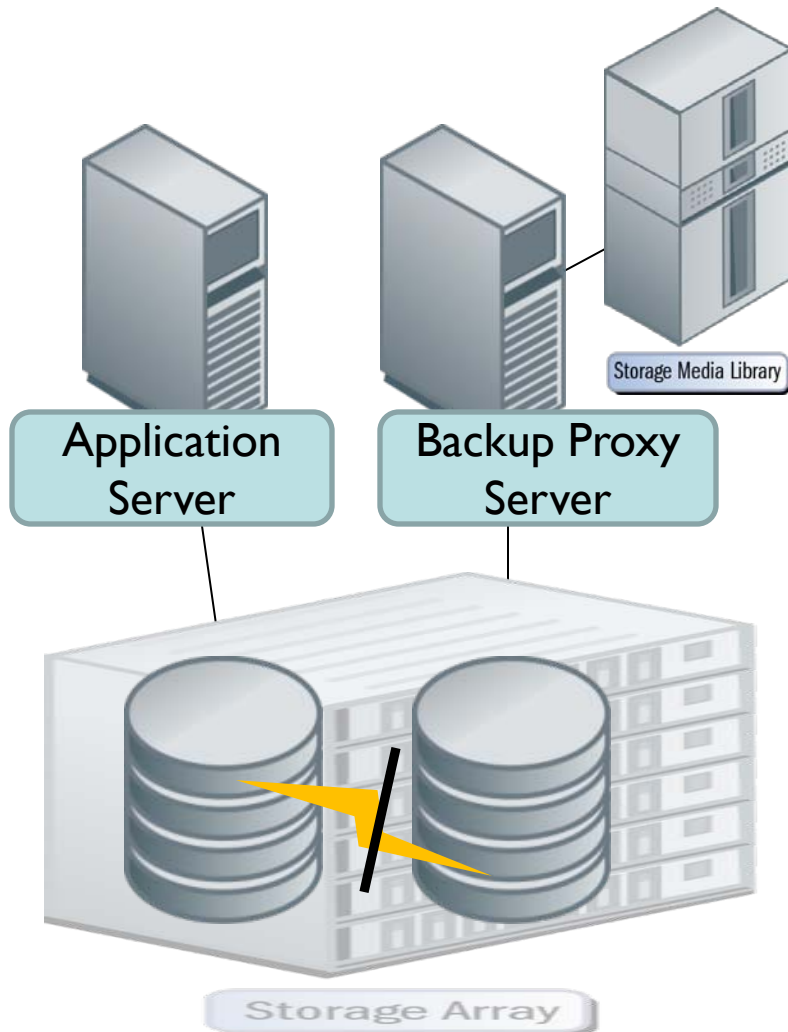
- ◆ Copy the entire volume via snapshot
- ◆ Copy all files needed to recover the application
- ◆ Incremental backup
 - Changed blocks
 - Changed files
- ◆ Feature set might be different compared to streaming API
- ◆ Backup to disk & restore from disk

➤ Trend: volume based backup

- ◆ Better for backup to disk
- ◆ Better for virtualization

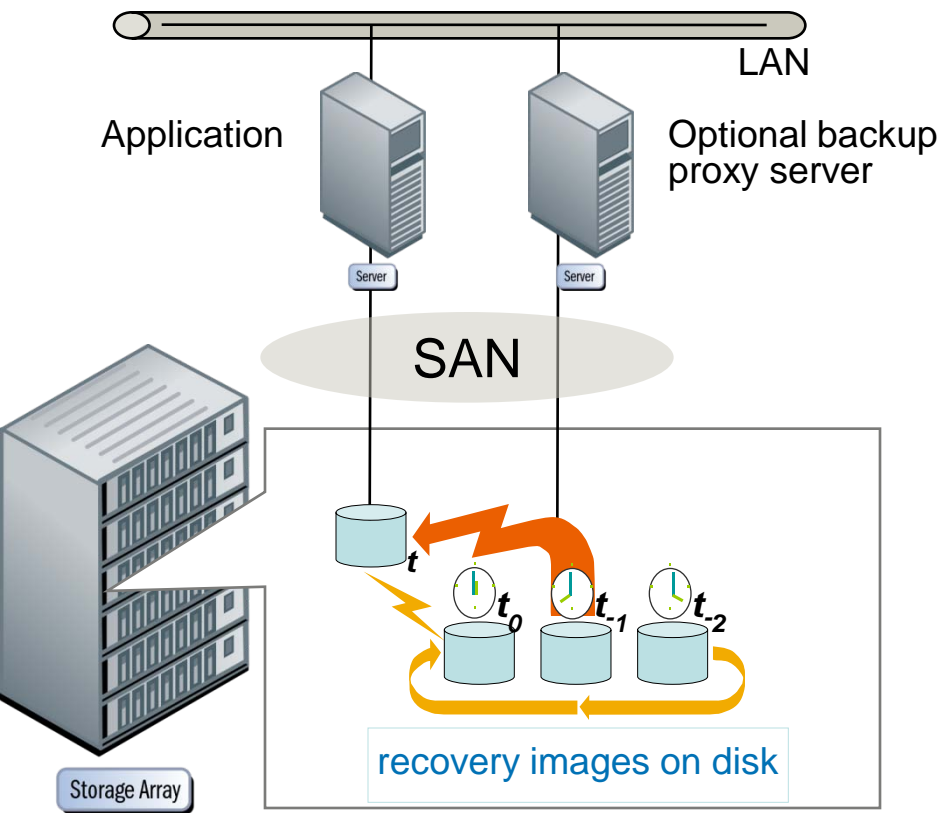


Reduced Performance Impact



- Separate backup proxy server
- Backup from full copy snapshot (mirror)
 - ◆ Application switches into backup mode
 - ◆ Split the snapshot
 - ◆ Back to normal mode
 - ◆ Separate backup proxy server copies the data from split mirror
 - ◆ Resync the mirror after backup
 - › Copies changed blocks only

Application Recovery from Snapshot



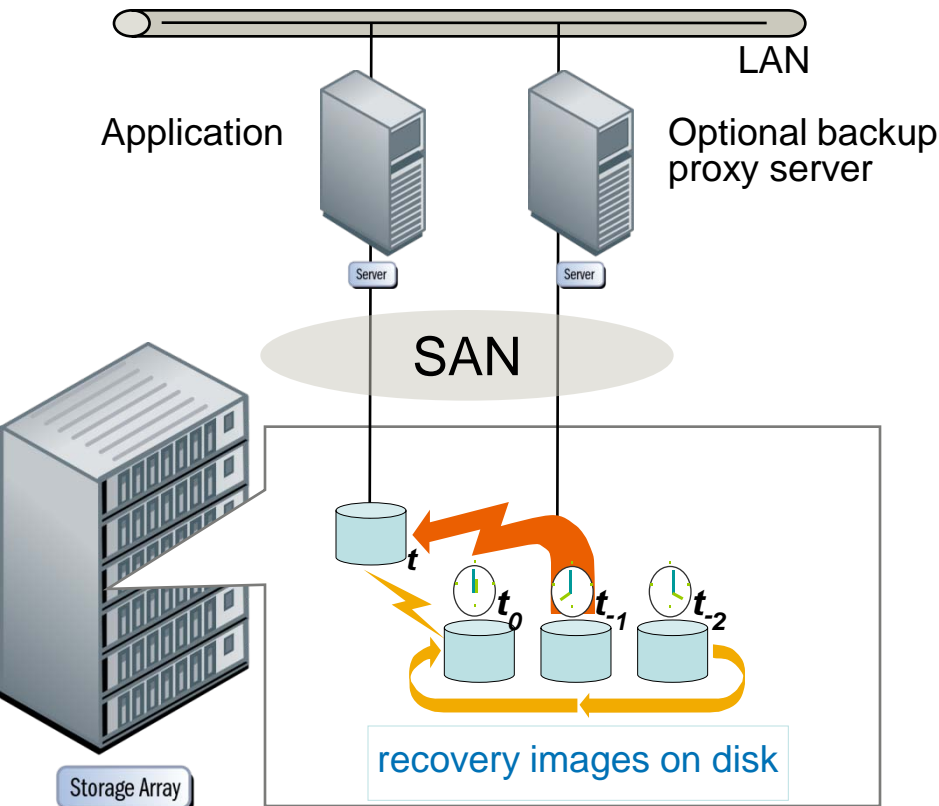
- Application shutdown
- Optional incremental transaction log backup
- Switch to selected snapshot
 - ◆ Instead of restore from tape
- Transaction log roll forward from backup or original disk up to the most recent point in time
- Application back online

Single Item Recovery Options

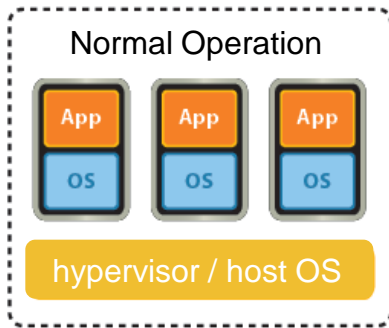
- Dumpster, 2nd level dumpster, versioning, archive
- Lagged database replicas maintained by the application
- Full blown recovery environment & copy back
 - ◆ Spare systems
 - ◆ Virtual Machines
- The application can be used to extract single items from backup
 - ◆ Copy database from backup & mount as recovery database
 - › Needs extra space and time to copy the entire database
 - ◆ Mount the database from the backup directly into the application
- Open the backup database with a separate tool & extract
- Extract single items directly from the backup
 - ◆ Catalog of all single items during or shortly after backup
- Single item recovery from single item backup
 - ◆ Needs a separate “brick level” backup



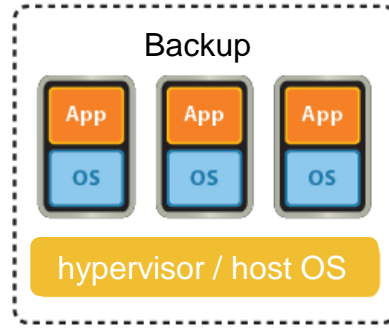
Single Item Recovery from Snapshot



- Mount the database from the snapshot
- Browse & search through the database
 - ◆ Using 3rd party tool
 - ◆ Using the application
- Unmount the snapshot
- Issues
 - ◆ Which snapshot to use?
 - ◆ Snapshot retention



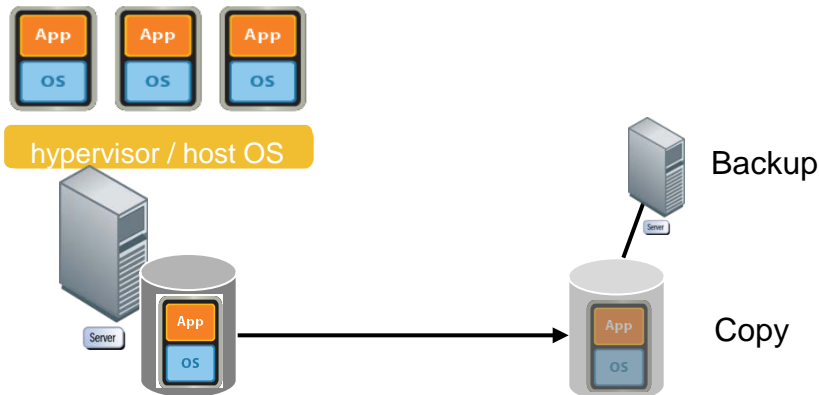
Superman



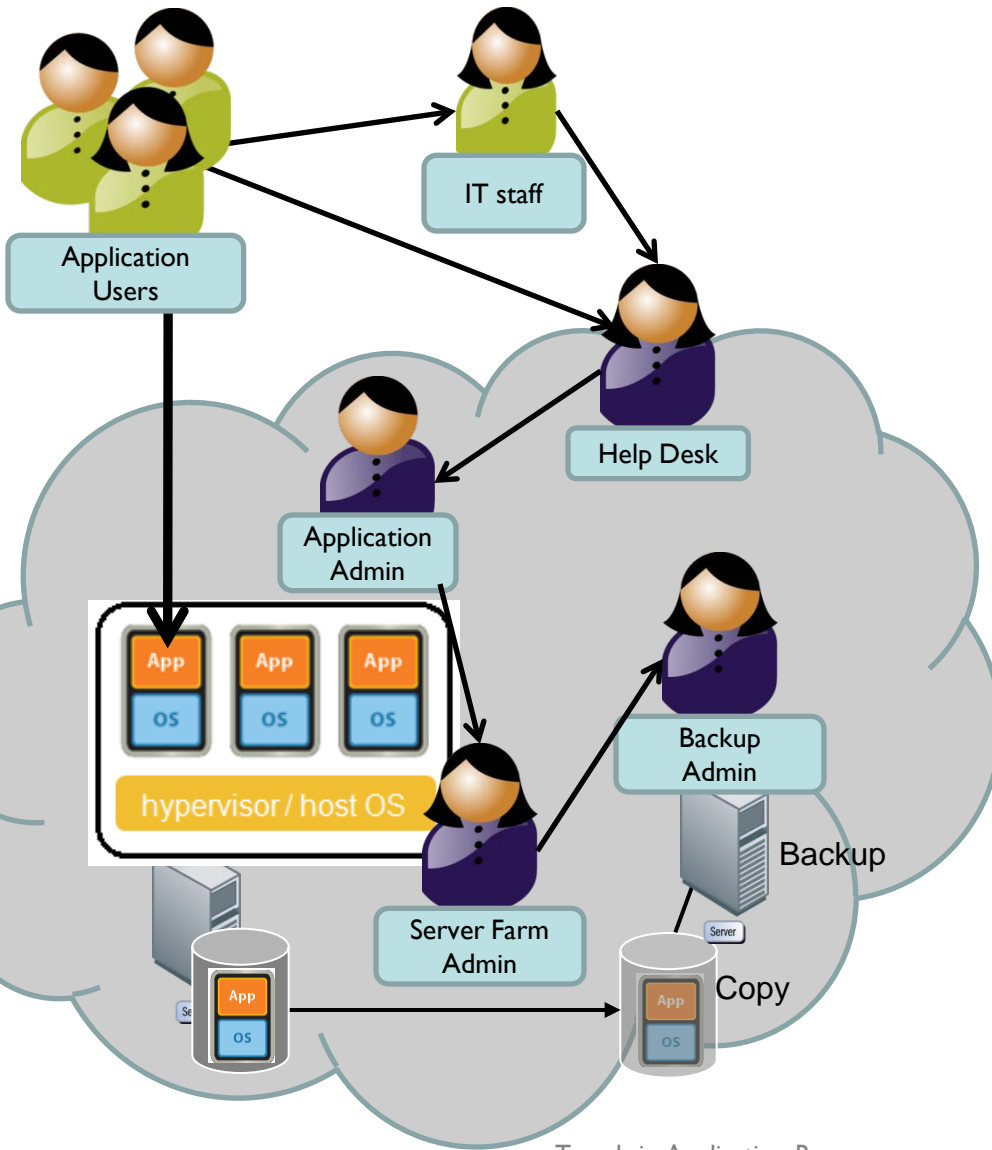
Donkey cart



- Resources shared among different applications
 - ◆ Normal load spreads evenly across day / week / month
 - ◆ Backup load is exceptional
- Resources on physical server often not enough for backup load
- Offload backup via dedicated physical machine
 - ◆ Utilize replication



Application Recovery in the Cloud



- Who does the backup?
 - ◆ Same SLA for all VMs?
 - ◆ Who can define the SLA?
- Who recovers what?
 - ◆ Hypervisor / host
 - ◆ Individual Virtual Machine
 - ◆ Single file from the VM
 - ◆ Application
 - ◆ Application data object
 - > E-mail
 - > Document
 - > Tablespace
 - > Record
- Security?
- Backup targets?
- Empowered end user
 - ◆ Self service



- Even an „unsinkable“ ship needs rescueboots
 - ◆ How many passengers?
 - ◆ Buffer?
- What are the data loss scenarios?
 - ◆ What can happen?
 - ◆ What are your recovery use cases?
 - ◆ Who is involved into the recovery process?
- Cost versus Risk
 - ◆ How much data do you accept to loose?
 - ◆ What is the backup time window?
 - ◆ How long is your acceptable recovery window?
- Backup stays the last line of defense

Refer to Other Tutorials



Check out SNIA Tutorials:

Trends in Data Protection and Restoration Technologies

Introduction to Data Protection: Backup to Tape, Disk and Beyond



Check out Hands On LAB:

Enterprise Content Management

- Please send any questions or comments on this presentation to SNIA:
trackdatamgmt@snia.org

**Many thanks to the following individuals
for their contributions to this tutorial.**

- SNIA Education Committee

**Andreas Schwegmann – this slide deck
Antal Nemes – application backup interfaces
Nancy Clay – tutorial program management
Klaus Bloecher**