Auditing Non TMF Applications

Escort AutoTMF

Product Overview

CARR SCOTT

SOFTWARE INCORPORATED

www.CarrScott.com

Carr Scott Software Inc.

- Founded in 1995 as Tandem spin-off company
  - Founders: Dr. Richard Carr and Harry Scott
  - R&D and support co-located in Compaq Cupertino campus
- Separate corporation - no financial ties to Compaq
- Continued very close working relationship with Compaq
- Growing customer base (approx. 50 NSK installations)
- Carr Scott Software is 100% Himalaya focused
- Both end-users and 3rd parties benefit from Escort technology
- Four main products
  - Escort❖SQL
  - Escort❖AutoTMF
  - Escort❖AutoSYNC
  - Ranger

CARR SCOTT
SOFTWARE INCORPORATED

Escort Technology Benefits

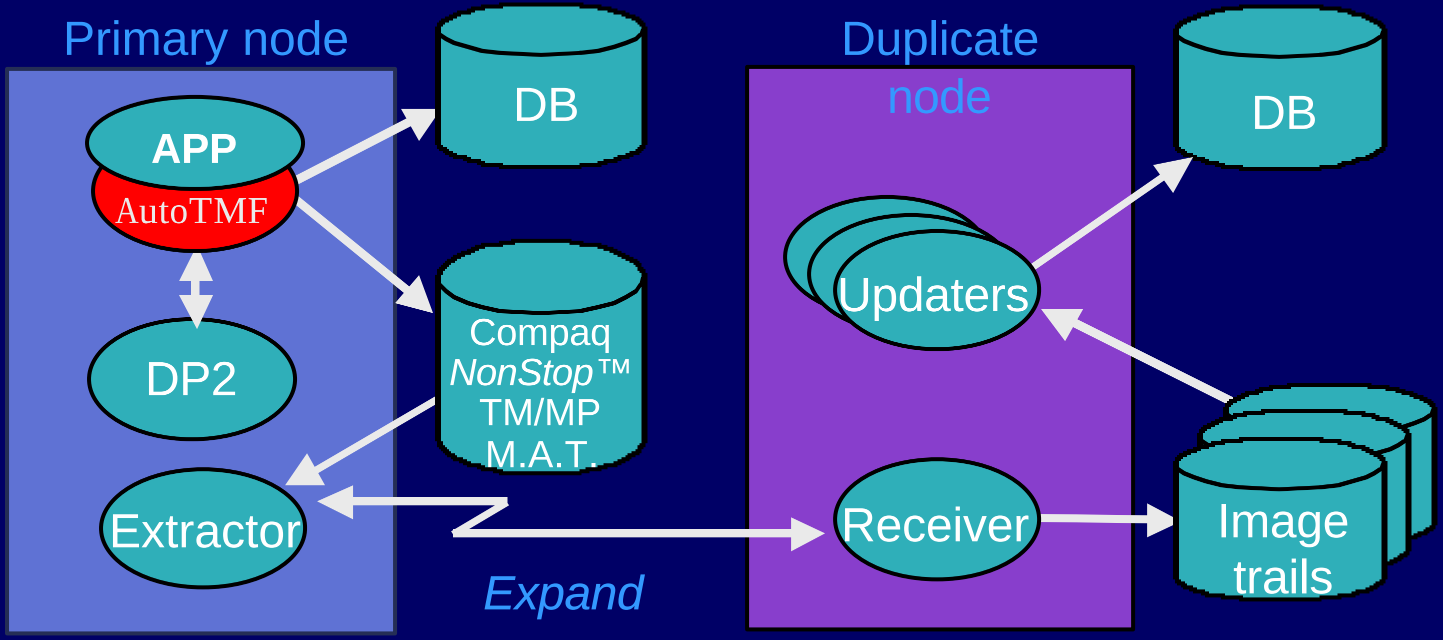
- Upgrade existing applications without rewriting
  - No source code required
  - Low risk, fast implementation using User Library intercept technology
- Advanced use of existing technology
  - Use only Himalaya standard interfaces and products
  - No privileged code, no SUPER access required
  - Not OS version dependent (any OS that is D30 or higher)
  - Any RISC platform (K or S-series Himalaya)
- Proven approach
  - Over 50 application environments

The Problem – Non or Partially Audited Applications

- NonStop TM/MP (a.k.a. TMF) has great advantages
  - Online backups for 24 X 7 availability
  - Data recovery
  - Improved application performance
  - Efficient and reliable replication with RDF, Shadowbase, etc.
- Nearly half of the existing applications on Himalaya systems are not coded to use TMF
- Even most TMF-aware applications do not protect all files
  - Incomplete replication and disaster protection
- Re-programming to use TMF is daunting and risky
- Development $$ better used for new features

CARR SCOTT
SOFTWARE INCORPORATED

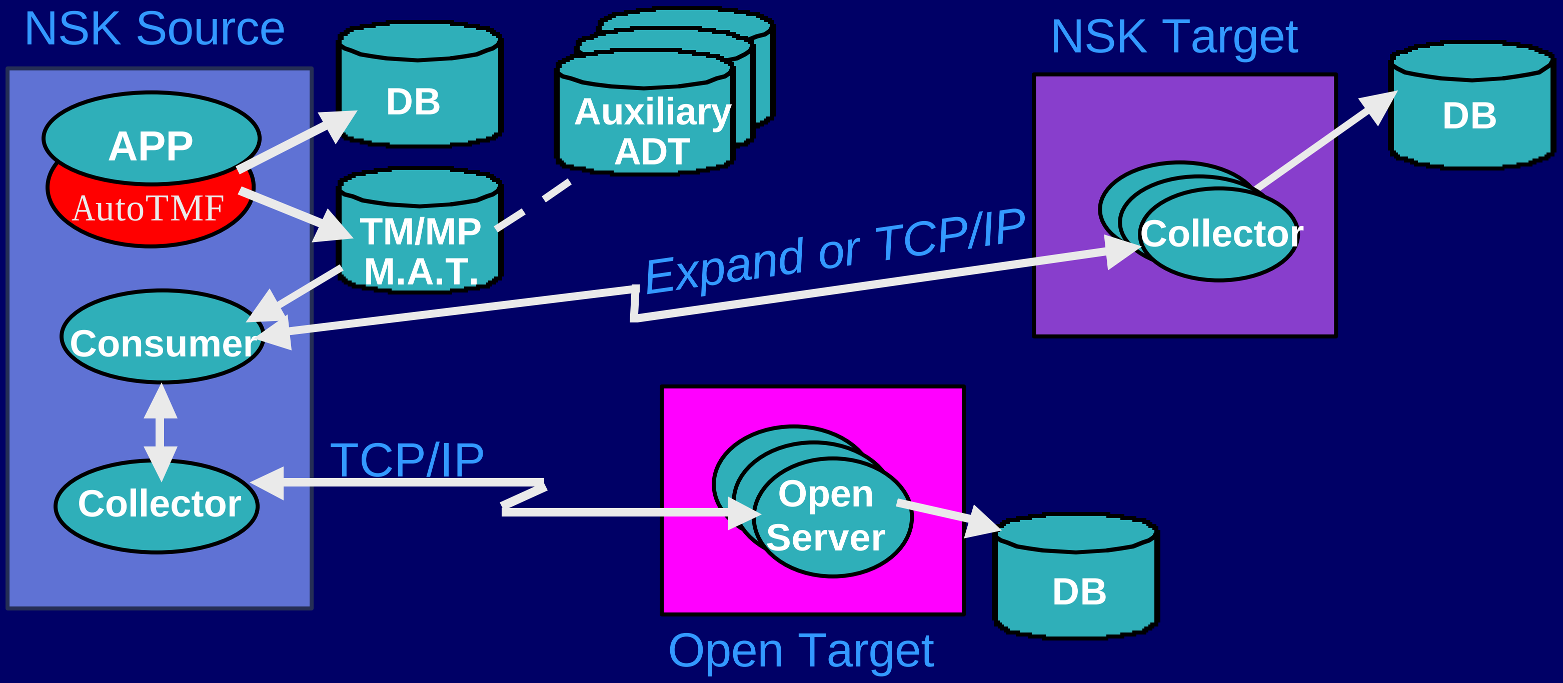
TMF / RDF
Himalaya to Himalaya replication



- I/Os done block at a time, direct to DP2 (not record by record thru file system)
- Low overhead, implemented near the metal (TMF, RDF & DP2 all low level)
- High-performance, real-time update to backup (2.5MB per second updaters)
- Compaq +25 years of data integrity experience

CARR SCOTT
SOFTWARE INCORPORATED

TMF / Shadowbase3 – Loosely coupled, low latency, flexible data replication.



- Loosely coupled replication (replication not bound to application)
- Extremely low latency (no extract file or intermediate data store)
- Flexible, application level replication (heterogeneous source/target support)
- Bi-directional replication (NSK/NSK, NSK/Oracle, Oracle/Oracle)

CARR SCOTT
SOFTWARE INCORPORATED

The Solution – AutoTMF

- AutoTMF enables applications for TMF quickly
  - A few minutes to install
  - A few minutes to prepare object files
    - No programming or re-compiling or binding
  - A few minutes to audit desired files
- Low overhead and minimal operations requirements
- Gain TMF benefits immediately
  - Replication, online dumping and performance
  - Data consistency same or better than pre-AutoTMF application
    - Transaction boundaries are not “business transactions” but locking behaviour tends to align that way
    - “Business transactions” can be implemented incrementally

CARR SCOTT
SOFTWARE INCORPORATED

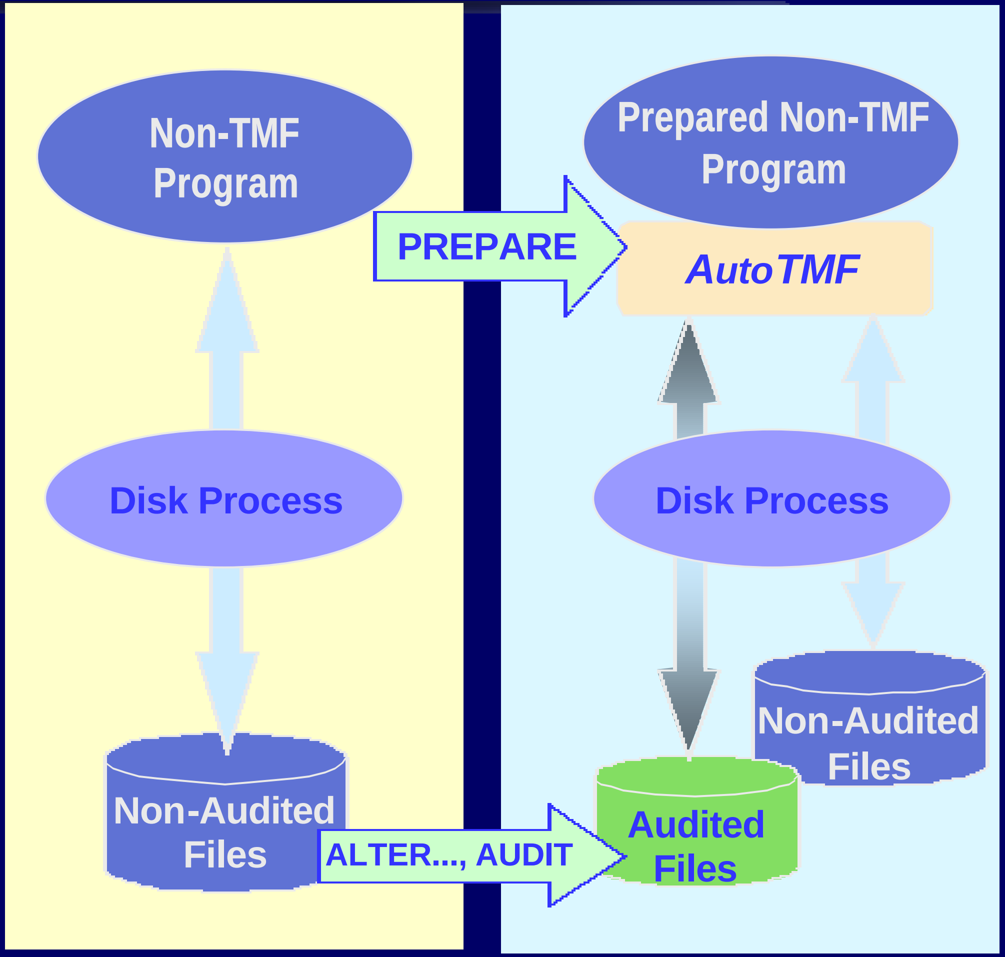
AutoTMF Technical attributes

- Not privileged, requires no sysgen
  - Looks like an application program to TMF
    - Uses standard APIs
  - Distributed & supported by Compaq NonStop AutoTMF (SA45V1)
- Supports NonStop Kernel releases D38 to Gxx
- Requires no changes to programs
- Requires no changes to Pathway configurations or batch scripts
- Simple set of configuration parameters to adapt to particular environments or special requirements
- Fault tolerant, scaleable and high performance

CARR SCOTT
SOFTWARE INCORPORATED

Migration Process

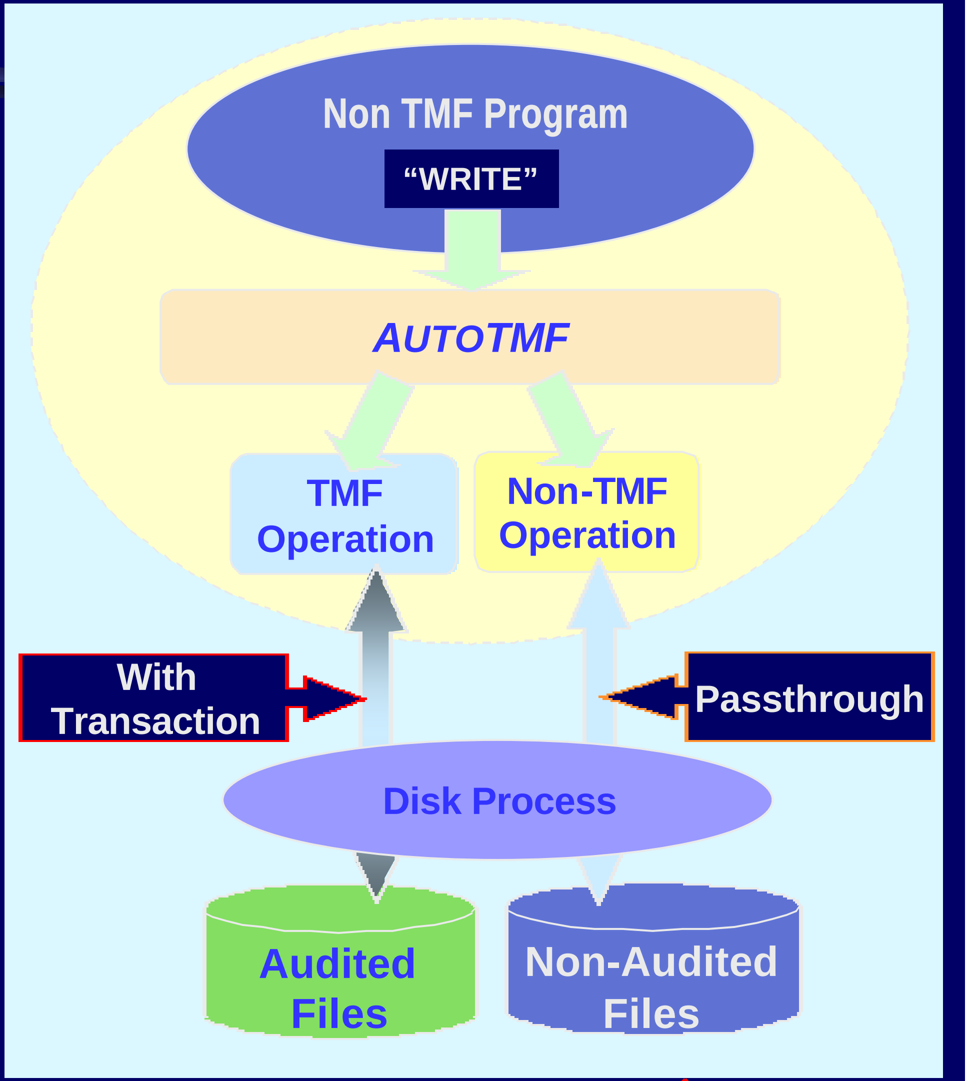
- Programs
  - One command (Prepare)
  - Operates on object file only
  - No changes to source or logic
  - No recompile, SQL compile, bind or axcelerate
- Database
  - Turn on audit
  - Incrementally migrate files



CARR SCOTT
SOFTWARE INCORPORATED

AutoTMF Runtime

- User library intercepts file system and TMF calls
  - Keeps a "state" of program
- Automatic transactions are triggered by operations that require a transaction
- Operations are then passed through to the file system
- Designed for performance
  - Many updates per transaction
  - Files are buffered
  - Efficient block splits



CARR SCOTT
SOFTWARE INCORPORATED

Automatic Transactions

- AutoTMF tracks all audited file accesses
  - Intercepts OPEN, READLOCK, WRITEUPDATEUNLOCK, etc.
  - Intercepts BEGIN/ END/ ABORTTRANSACTION
  - Maintains logical (i.e, unaudited) lock state
- AutoTMF assumes correct “unaudited” program logic
  - Consistent locking and unlocking behavior
  - No sharing of locks with servers
- Issues BEGINTRANSACTION when required
  - Locks and updates when application has no Transaction
- Issues ENDTRANSACTION when appropriate
  - Preserve updates and lock protocols
  - Maximize performance

CARR SCOTT
SOFTWARE INCORPORATED

Automatic Transactions Rules

- All AutoTMF transactions must be committed
  - AutoTMF never ABORTs automatic transactions
  - Emulates file operations on a unaudited database
  - A program cannot stop without committing automatic transactions
- Automatic transactions preserve locking protocol
  - Cannot commit an automatic transaction if unaudited file access still has a record / file locked
- Automatic transactions are local to one process
  - Cannot be exported or inherited

Automatic Transactions Start

- First transactional access determines use of automatic transactions for the open:
  - The first access occurs when the application has not started or inherited its own transaction (default)
  - Or when automatic transactions are configured for the file
- Well-known operations on an audited file which start an automatic transaction
  - READLOCK
  - WRITE
  - LOCKFILE

Automatic Transactions Commit

- Application must have released locks
- Logical points in the processing
  - READUPDATE $RECEIVE
  - REPLY
  - File close and process stop
- Isolation driven (configurable)
  - Server SEND
  - Other external communication (spooler e.g.)
- Activity-driven (configurable)
  - Elapsed time
  - Number of updates
- Balance performance and consistency

CARR SCOTT
SOFTWARE INCORPORATED

Automatic Transactions Abort

- AutoTMF never aborts an automatic transaction
  - Emulates non-audited behavior
  - Prevents process termination with outstanding automatic transactions
  - Does not export automatic transactions
- External events could cause a unilateral abort
  - Cannot be prevented but rare (CPU failure, TMF autoaborts, etc.)
- Rollback limited to the recent updates
  - Effects can be controlled with configuring isolation levels

CARR SCOTT
SOFTWARE INCORPORATED

Automatic Transaction Configuration

- Typical non-TMF application requires no special configuration
  - Audit the files
  - AutoTMF generates the transactions when needed
- Parallel transactions (separate or common)
  - For TMF applications with some non-audited files
  - Parallel transactions configured for newly audited files
  - Program still manages transactions for the previously audited files
- NOWAIT transactions
  - Automatic transaction commit happens “nowait”
- Isolation
  - Controls outside awareness of uncommitted automatic transactions
    - Processes, non audited files and devices

Creating and Renaming Audited Files

- Files created programmatically can be audited automatically
- AUTOTMF emulates RENAME operations on audited files
  - If the file is not opened by another process
  - Normally not possible with audited files
  - Rename is currently not replicated

TMF / AutoTMF Performance

- AutoTMF is optimized
  - No additional processes; no extra I/O
  - One configuration message on process start and audited file opens
    - Configuration cached in memory
  - Automatic transaction generation requires very little processing
  - Passthrough is virtually free (a few microseconds)
- AutoTMF performance is TMF performance
- TMF activity is optimized
  - Several inserts/updates per transaction
  - Parallelism when multiple transactions are configured
  - Nowait transactions
  - User configurable

CARR SCOTT
SOFTWARE INCORPORATED

Escort❖ Tools

- Tracing and debugging
  - Configured via a simple command without altering process startup
    - No changes to Pathway server configuration needed
  - Shows every procedure call
  - Show process startup and termination state (defines, assigns, startup message)
  - Shows all AutoTMF operations
  - Optionally shows input and output data
- Lock management tools
  - One command to display all record and file locks (reworked LISTLOCK command)
  - Deadlock detection
  - Long transaction detection

AutoTMF Futures

- Implementation dates subject to customer commitments
  - AutoTMF support for unaudited SQL tables
  - Super-fast takeover support (Wait_for_RDF)
  - AutoTMF support for native mode applications
  - RDF/TMF Lock-step transaction support (eliminate reprogramming)
  - RDF file RENAME and PURGE support

CARR SCOTT
SOFTWARE INCORPORATED

Replication of non-Database files

Escort❖AutoSYNC

Product Overview

CARR SCOTT
SOFTWARE INCORPORATED

www.CarrScott.com

Non-Database File Replication

- Database replication does not provide complete disaster recovery

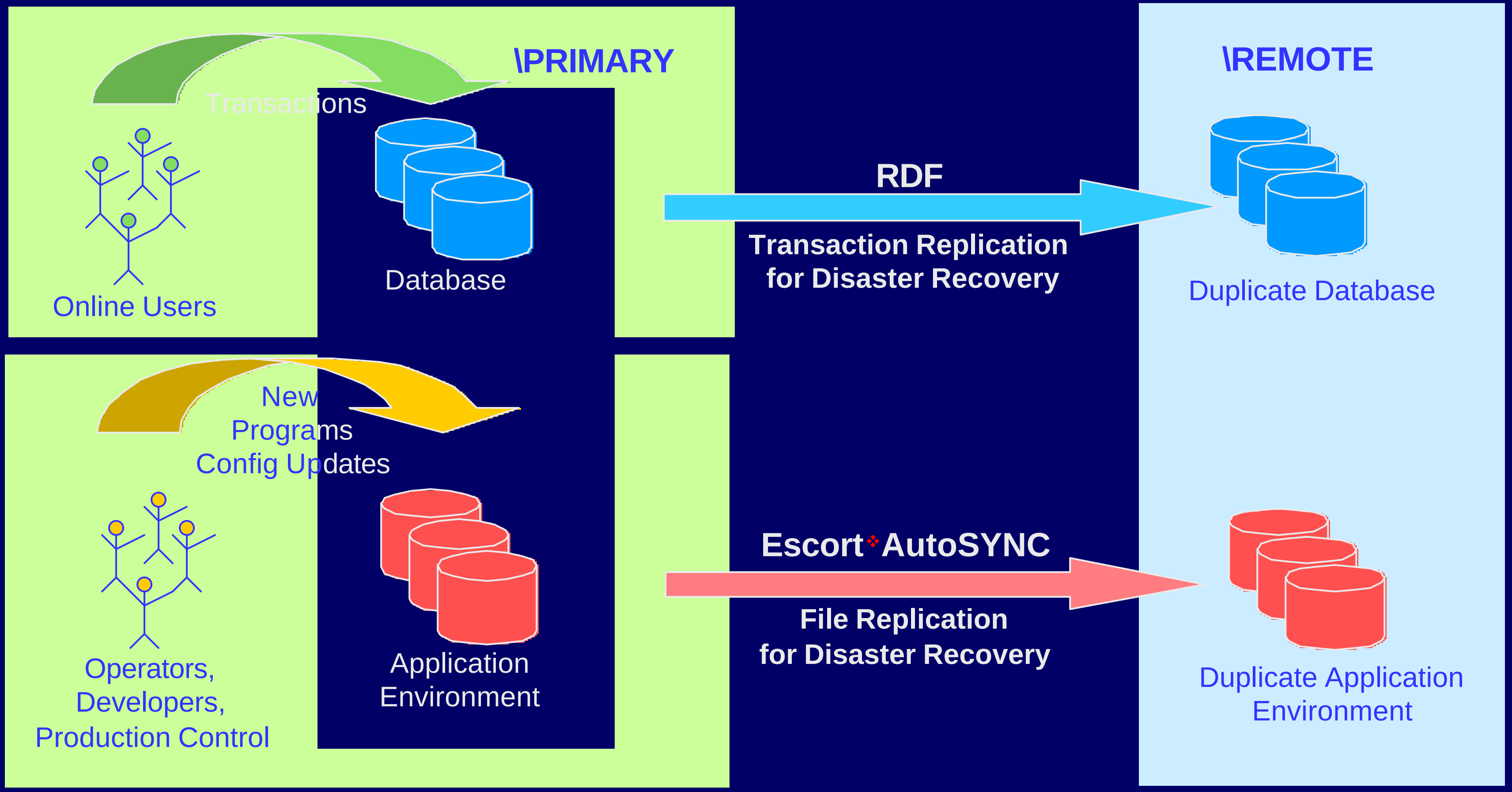
- How do you manage the “Application Environment”?
  - TACL macros and Edit files
  - Configuration files (Pathway, Batch, Spooler, etc...)
  - Object / Source files
  - Report files, BLOBs, etc...
  - Non-audited Enscribe files / SQL tables

- Auditing of such files is either impractical or impossible

EscortvAutoSYNC Overview

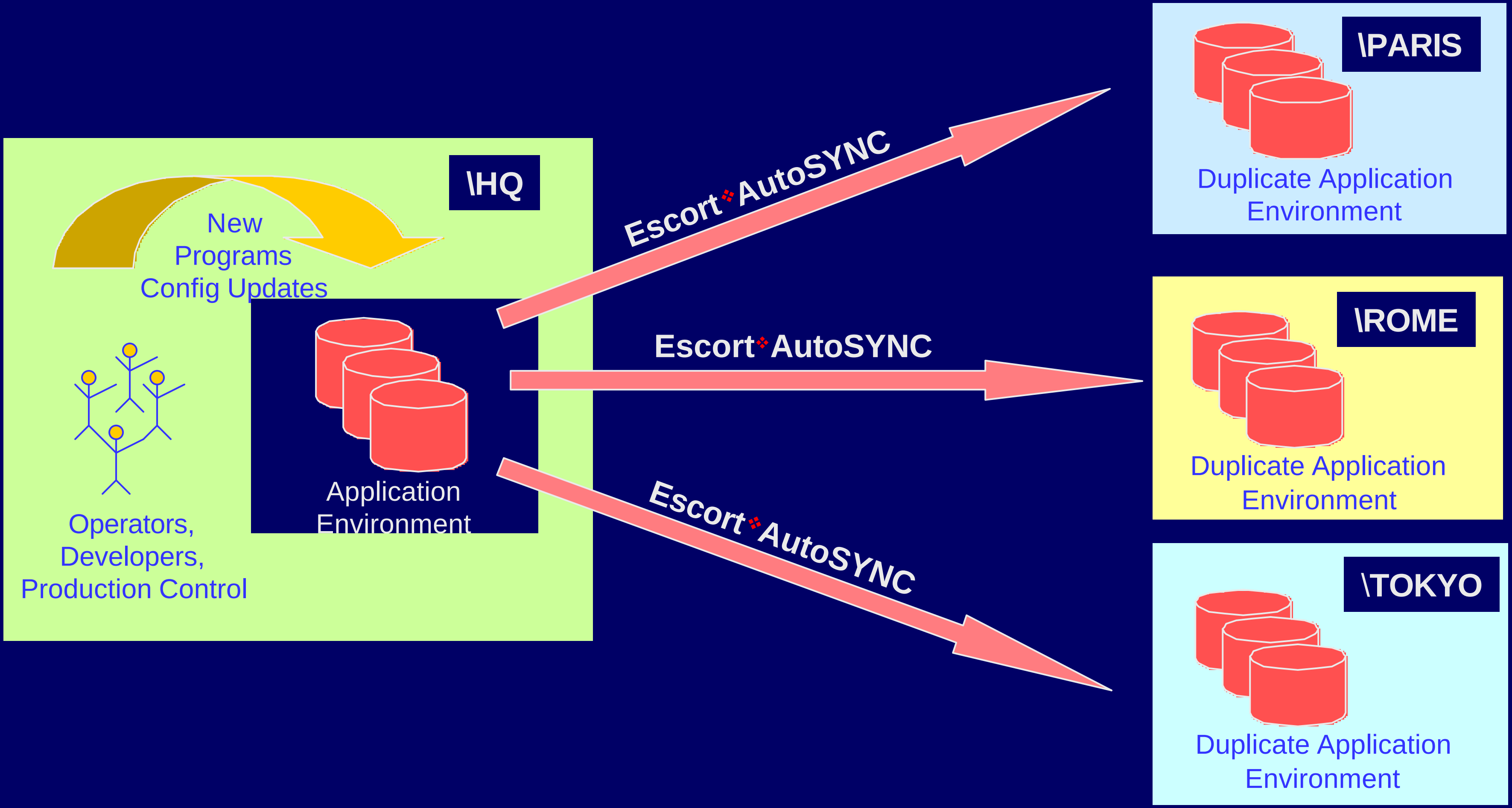
- Synchronizes filesets between Himalaya systems
  - Replication of entire files, not individual records
- Complements RDF and other replication products
- Easy to install; easy to configure and manage
  - Completely automatic; set it and forget it
  - Fault tolerant; highly reliable
  - Uses standard security facilities
- Primary uses:
  - Complete Disaster Recovery preparation
  - Automated Operations support (SW distribution)

AutoSYNC for non-database file replication



CARR SCOTT
SOFTWARE INCORPORATED

AutoSYNC for Software Distribution



CARR SCOTT
SOFTWARE INCORPORATED

AutoSYNC Technical attributes

- Not privileged, requires no sysgen
  - Looks like an application program
  - Uses standard APIs and Compaq facilities
- Supports NonStop Kernel releases D38 to Gxx
- Requires no changes to programs or environment
- Scaleable and high performance
- Simple, yet flexible configuration

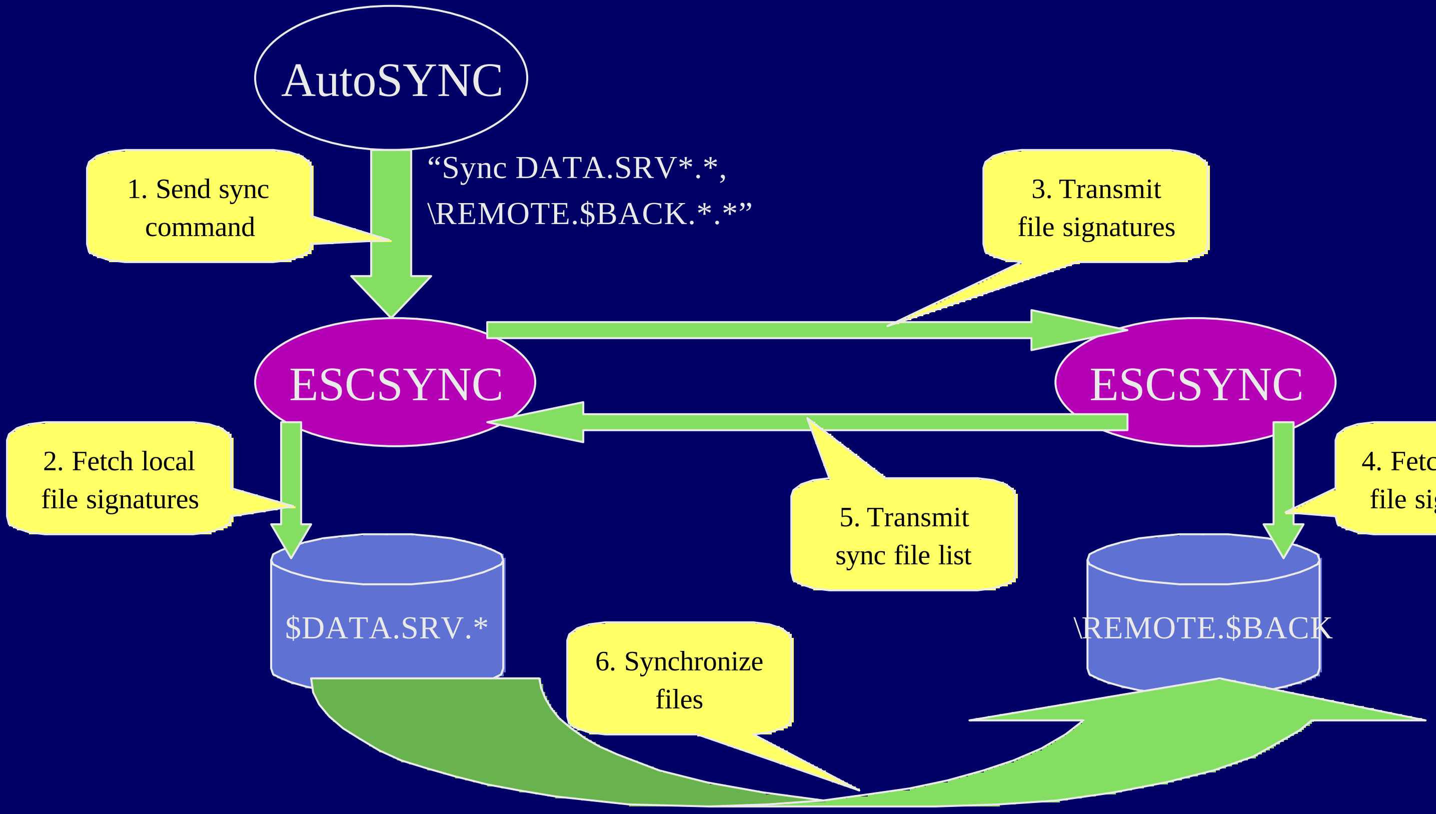
CARR SCOTT
SOFTWARE INCORPORATED

Building blocks

- An Administrator adds SyncUsers
  - SyncUsers can schedule fileset synchronizations
  - Each SyncUser (user id) can be assigned priority and cpus for the synchronizing processes
  - Administrator can SUSPEND and ACTIVATE a SyncUser
- A SyncUser specifies Sync Filesets
  - Source file pattern (ex. $DATA.SRV*.ACCT*)
  - Destination file pattern (\REMOTE.$BACK.*.*)
  - Security and ownership options
  - Scheduling options
  - Compression and performance options
  - Destination file purge option

CARR SCOTT
SOFTWARE INCORPORATED

AutoSYNC Processing Architecture


AutoSYNC
1. Send sync command
"Sync DATA.SRV*.*, \REMOTE.$BACK.*.*"
3. Transmit file signatures
ESCSYNC
ESCSYNC
2. Fetch local file signatures
4. Fetch remote file signatures
5. Transmit sync file list
$DATA.SRV.*
\REMOTE.$BACK
6. Synchronize files

CARR SCOTT
SOFTWARE INCORPORATED

Selecting Files for Synchronization

- AutoSYNC synchronizes files in the Fileset that:
  - Do not exist on the remote system
  - Are “older” on the remote system
    - User may update files on the remote system to change system/volume names, etc.
    - Difference in system clocks is factored in
    - “Exact” synchronization can be specified
    - Are not open for exclusive or update access
  - Are not TMF audited
  - Are not “ZZ” files (unless specified)

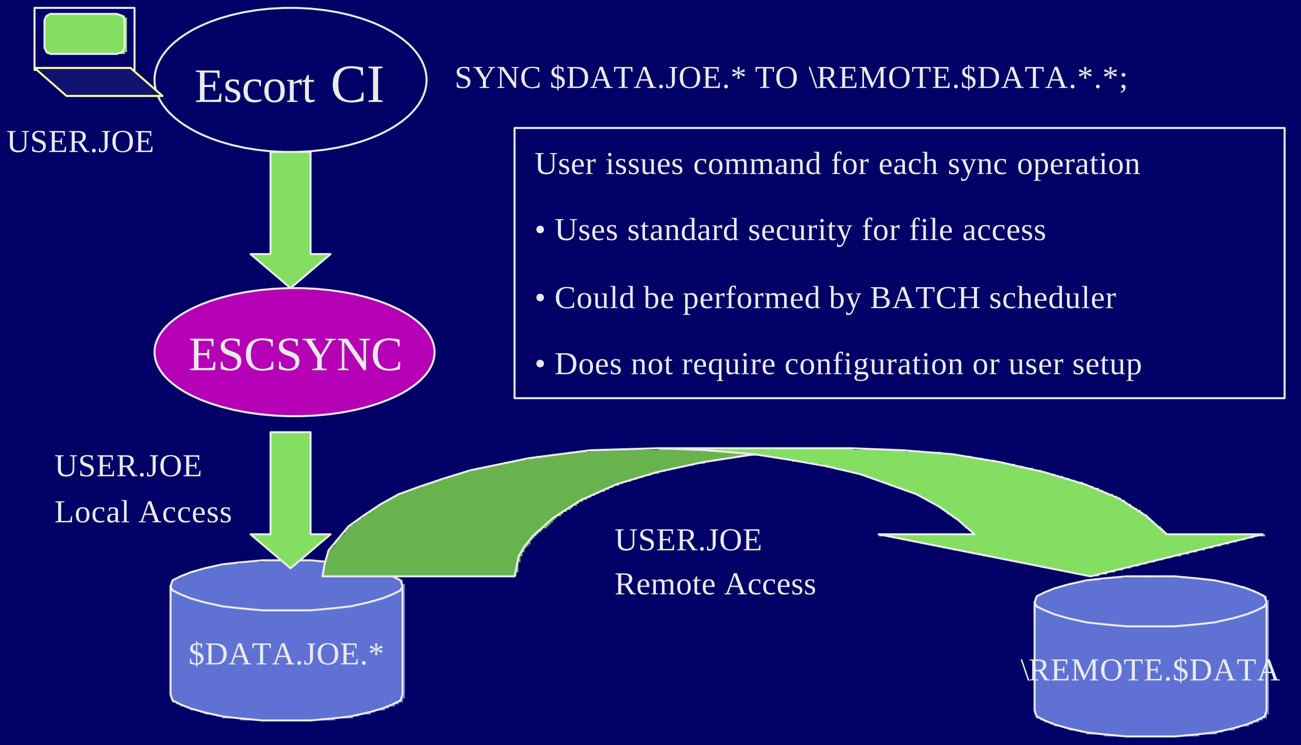
CARR SCOTT
SOFTWARE INCORPORATED

Synchronization Modes

- Online Synchronization
  - Immediate, command-driven synchronization
  - Employs current user’s local and remote access rights
  - Same security requirements as a FUP DUP

- Scheduled Synchronization
  - Multiple users can configure synchronization
  - Automatic; no user intervention
  - Reliable, fault-tolerant
  - Requires security authorization

Online Sync Architecture



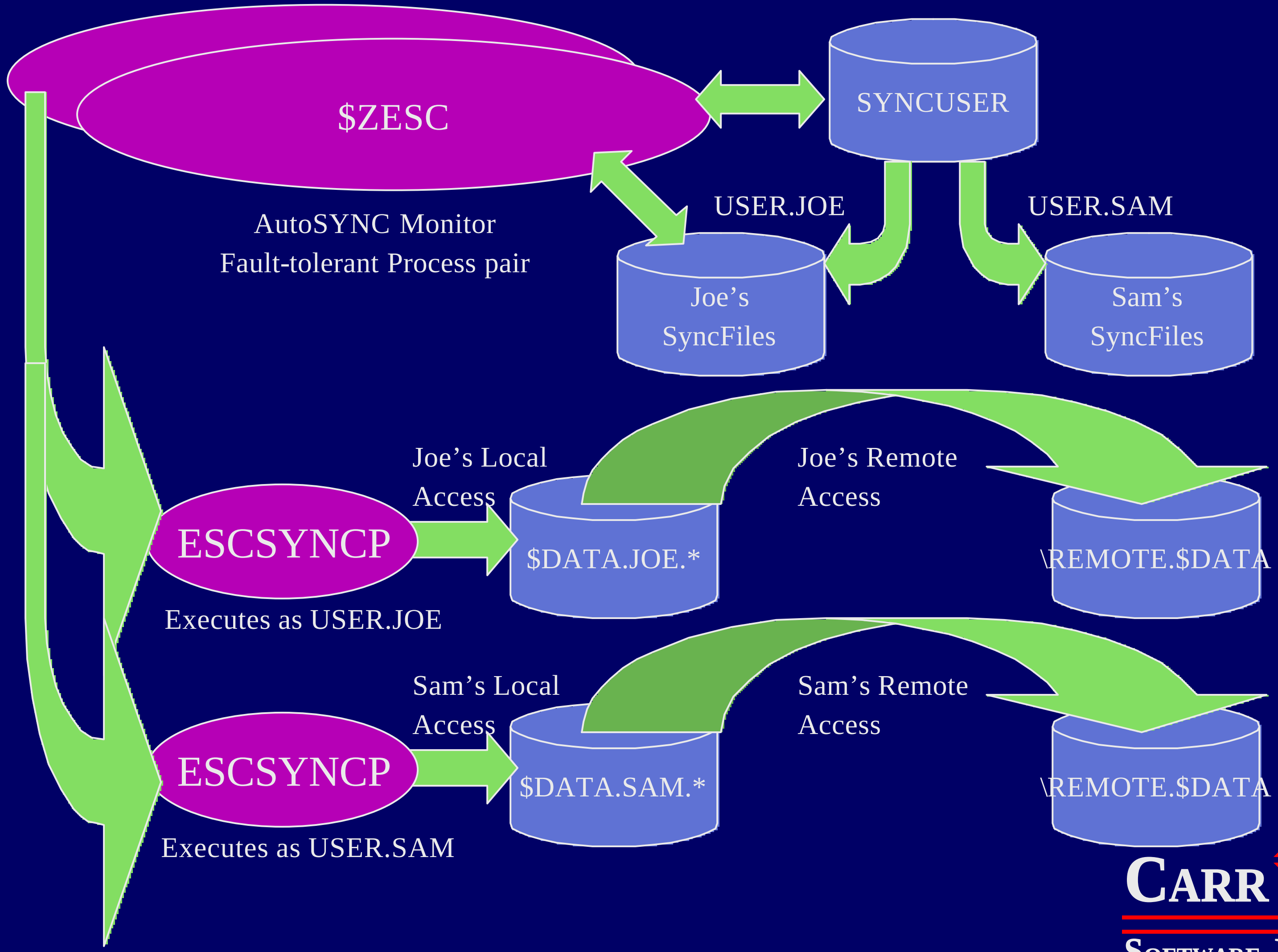
SYNC $DATA.JOE.* TO \REMOTE.$DATA.*.*;

User issues command for each sync operation

- Uses standard security for file access
- Could be performed by BATCH scheduler
- Does not require configuration or user setup

CARR SCOTT SOFTWARE INCORPORATED

Scheduled Sync Architecture



CARR SCOTT
SOFTWARE INCORPORATED

Fileset Scheduling Options

- Default is every 5 minutes
  - User can specify interval from 1 minute to n days.
- Daily START and STOP times
  - Can specify “sync every day starting at 6PM until midnight”
- SUSPEND and ACTIVATE
  - Can temporarily suspend synchronization without deleting fileset configuration

Compression

- Uses LZW algorithm
- Eliminates 60% to 80% of network traffic
- CPU-intensive
  - Compression may not increase throughput if lines are fast
  - AutoSYNC uses compression if it improves throughput (dynamic calculation)
- Compression can be enabled/disabled to fit customer

CARR SCOTT
SOFTWARE INCORPORATED

Performance Options

- Automatic synchronization parallelism for different users and/or different remote systems
- Configured parallelism by specifying “batchid”
- Priority and Throttling
  - Synchronization should not affect ongoing production work
  - User can specify process priority
  - User can specify maximum percentage of busy time

Destination File Purge Option

- Files that do not exist in the source fileset can be purged in the destination fileset
- Used to maintain subvolumes in parallel
- Does not purge files in a destination subvolume if the source subvolume does not exist

Security and Ownership Options

- Automatic upgrade to network security
  - Local security could prevent access after remote sync
  - Local security changed to network security (“A” becomes “N”)
  - Can be disabled with NO NETWORKSECURE option
- SyncUser can specify owner of replicated files
- SyncUser can specify security of replicated files
- Purge security could prevent repeated syncs
  - Changed to allow sync user to purge file
    - Example: N” security if SyncUser and file owner are in different groups
  - Might allow anyone to purge a replicated file
  - Does not allow any new access to file contents

CARR SCOTT
SOFTWARE INCORPORATED

Synchronization Security

- Scheduled Synchronization Security
  - Individual users configure synchronization
    but
  - AutoSYNC monitor process manages ESCSYNC processes
  - Ensures that SYNCs are performed under SyncUser's user id
  - Requires authorization
    - Prevents users from accessing files without authorization
- Online Synchronization Security
  - All operations use current user’s file access rights

CARR SCOTT
SOFTWARE INCORPORATED

AutoSYNC Futures

- Implementation dates subject to customer commitments
- Triggers
  - Execute a user-specified program or macro when file is synchronized
  - Examples of functions that Triggers could perform:
    - Execute EDIT scripts to adjust system / volume names
    - SQLCOMP programs
    - Run batch jobs or other user-defined processes
    - etc...
- OSS support (investigation)

CARR SCOTT
SOFTWARE INCORPORATED

Want More Information?

- Product Information
  - http://www.CarrScott.com/products_autotmf.html
    http://www.CarrScott.com/products_autosync.html
  - info@carrscott.com
- Contact Information
  - Harry Scott, Co-founder +1.781.934.0989
    harry.scott@CarrScott.com