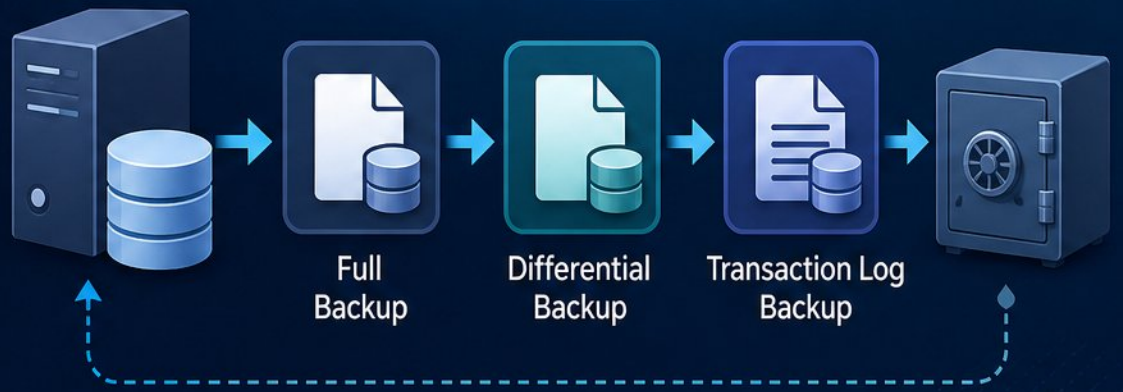
How to Backup SQL Database:

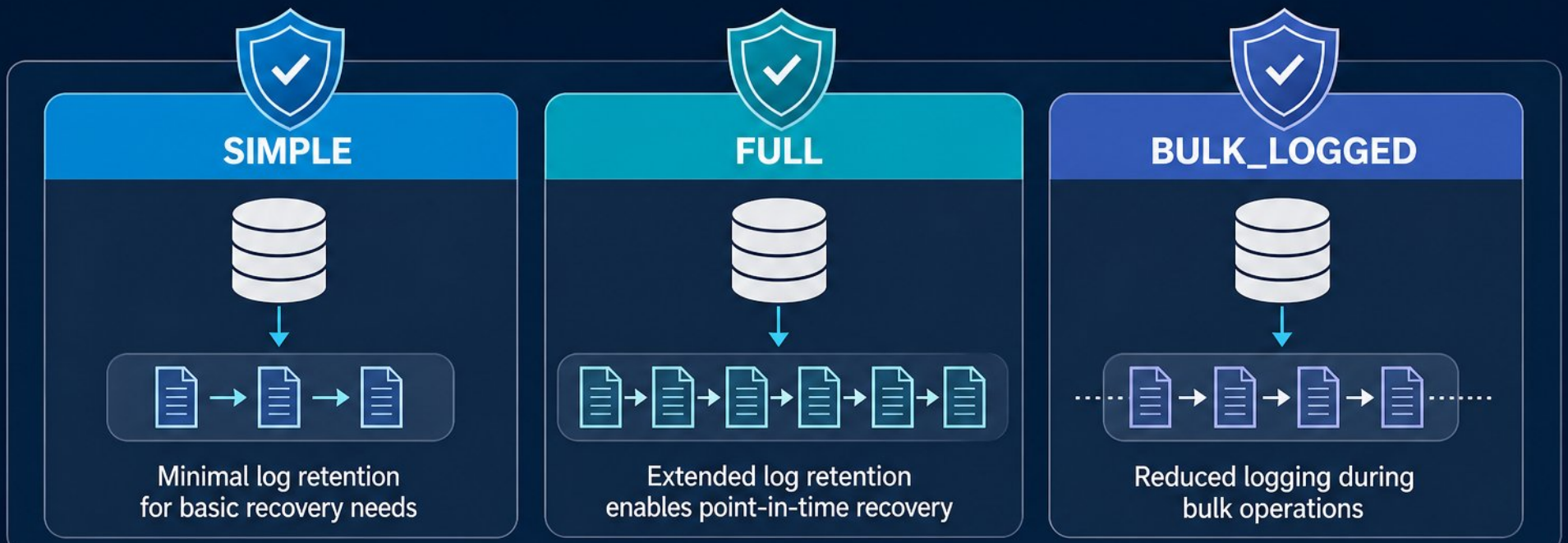
A Practical Workflow

- Database backups are the core safeguard against catastrophic data loss
- RPO defines acceptable data loss; RTO defines acceptable downtime
- RPO and RTO drive backup frequency, type, and recovery strategy
- Core backup types: full, differential, and transaction log backups
- Training covers recovery models, restore testing, and backup monitoring



Understanding the Recovery Model Decision

- Recovery models control transaction log management and backup options
- SIMPLE: minimal administration, recovery only to last full or differential backup
- FULL: enables point-in-time recovery, requires regular log backups
- BULK_LOGGED: temporary state for large operations, trades point-in-time flexibility



Choosing the Right Recovery Model

- Decision matrix: data loss tolerance vs. administrative overhead
- FULL recovery without initial full backup: a false sense of security
- Back up system databases during model changes
- ALTER DATABASE SET RECOVERY FULL after switch
- Changing to FULL requires a full or differential backup immediately

Backup Types and Their Roles

Full backups:
base of every backup chain

Differential:
changes since last full backup

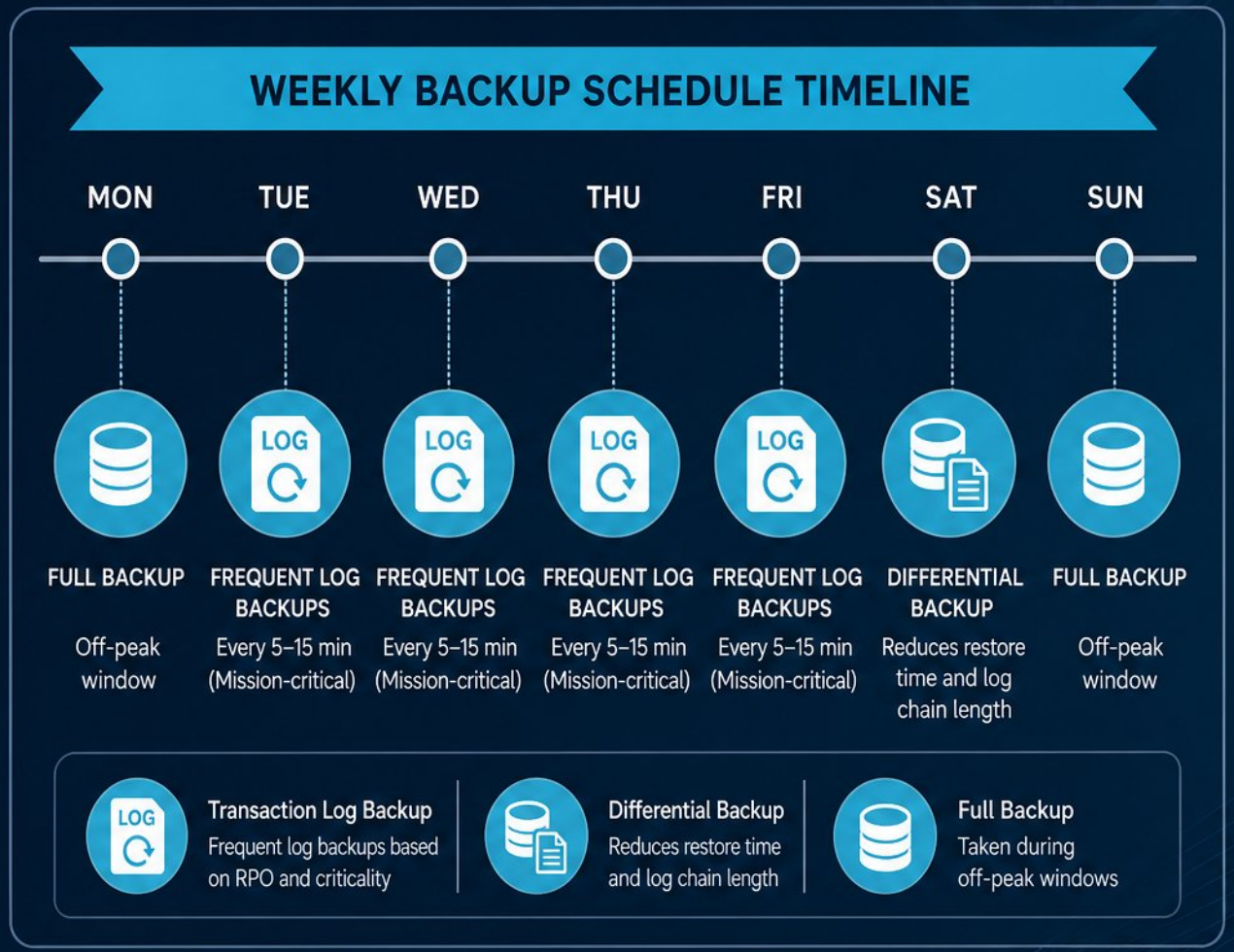
Log backups:
enable point-in-time recovery

Copy-only:
ad-hoc without breaking the chain

File/filegroup:
targeted backup for large DBs

Designing a Practical Backup Schedule

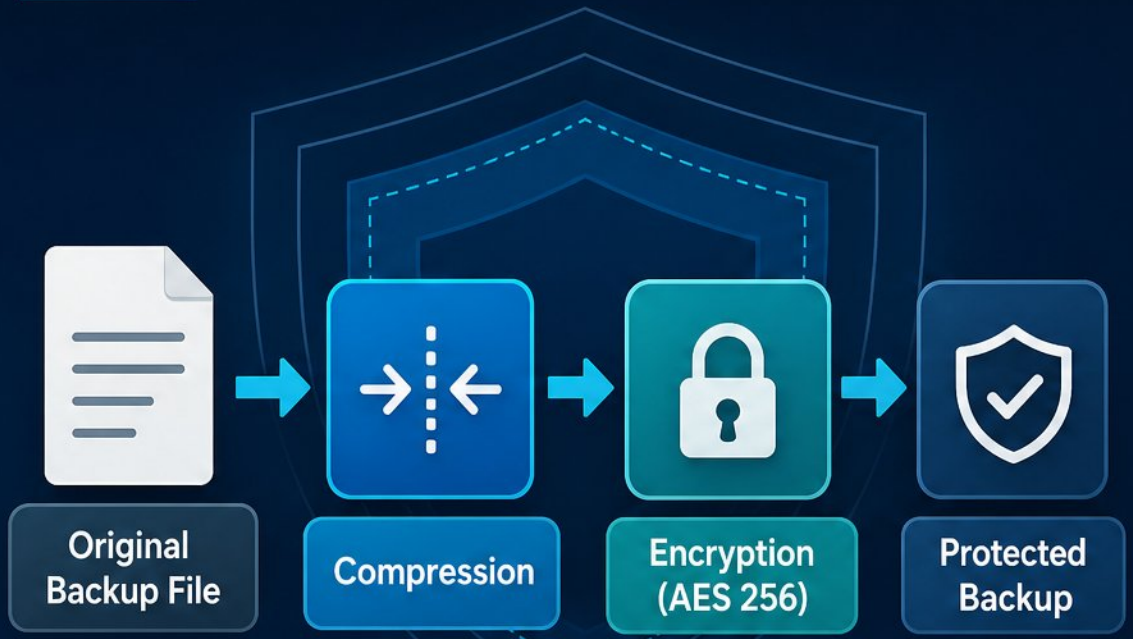
- RPO determines backup frequency: the tighter the RPO, the more frequent the transaction log backups
- Sample schedules: mission-critical (log every 5–15 min), business-critical (log every 30–60 min), non-critical (daily full only)
- Full backups during off-peak windows; differentials reduce restore time and log chain length
- Automate with SQL Server Agent jobs or maintenance plans; alert on failures immediately



Running Backups with T-SQL and SSMS

- Use BACKUP DATABASE and BACKUP LOG for full and log backups
- Employ SSMS dialogs for quick, ad-hoc backup operations
- Validate backup integrity with RESTORE VERIFYONLY
- Query msdb backup tables to confirm job success and coverage

Backup Storage, Compression, and Encryption



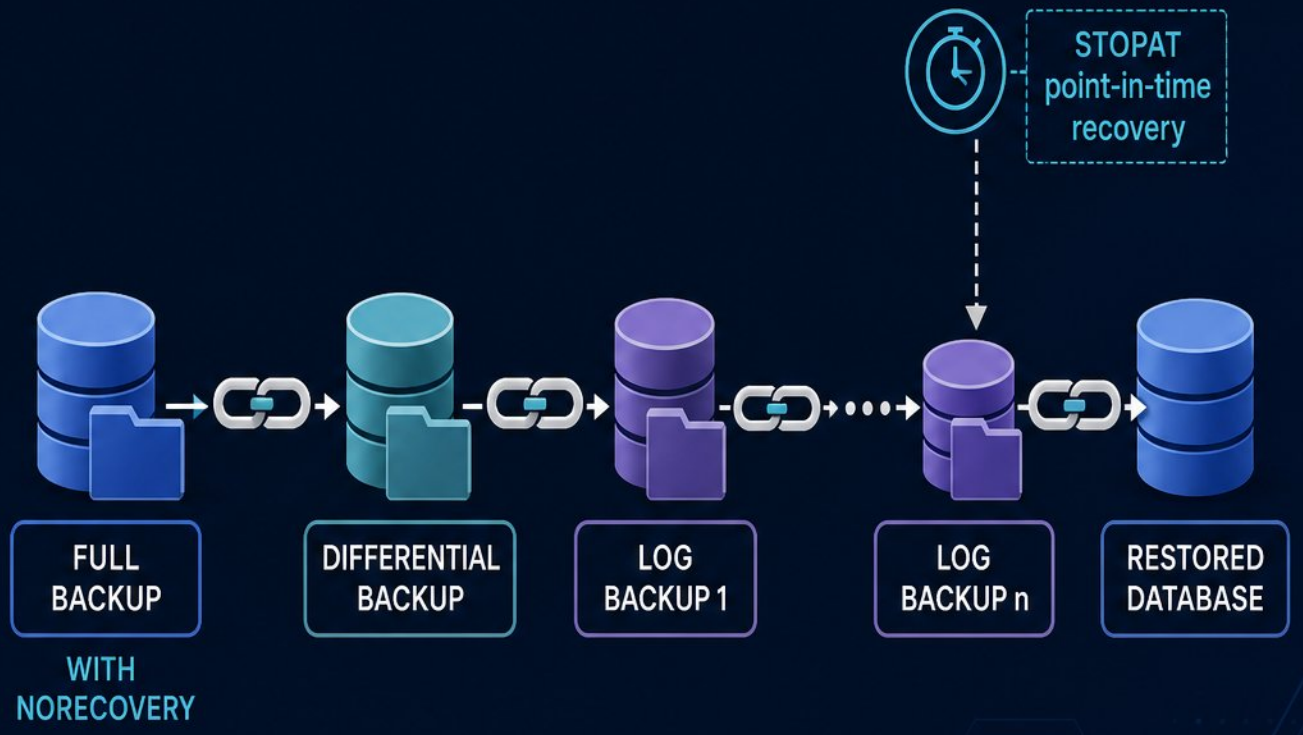
- Destinations: local disk, network share, or cloud storage
- Compression reduces I/O but increases CPU usage
- Encrypt with AES 256 using a certificate or asymmetric key
- Always back up the encryption certificate separately
- For TDE: compression after decryption is key

Naming, Retention, and Storage Strategy

- Use a consistent naming convention: DBName_Type_YYYYMMDD_HHMM.bak
- Set retention based on business requirements, not just disk space
- Keep multiple copies: local disk, offsite, and cloud storage
- Track backups in msdb and periodically clean history with sp_delete_backuphistory

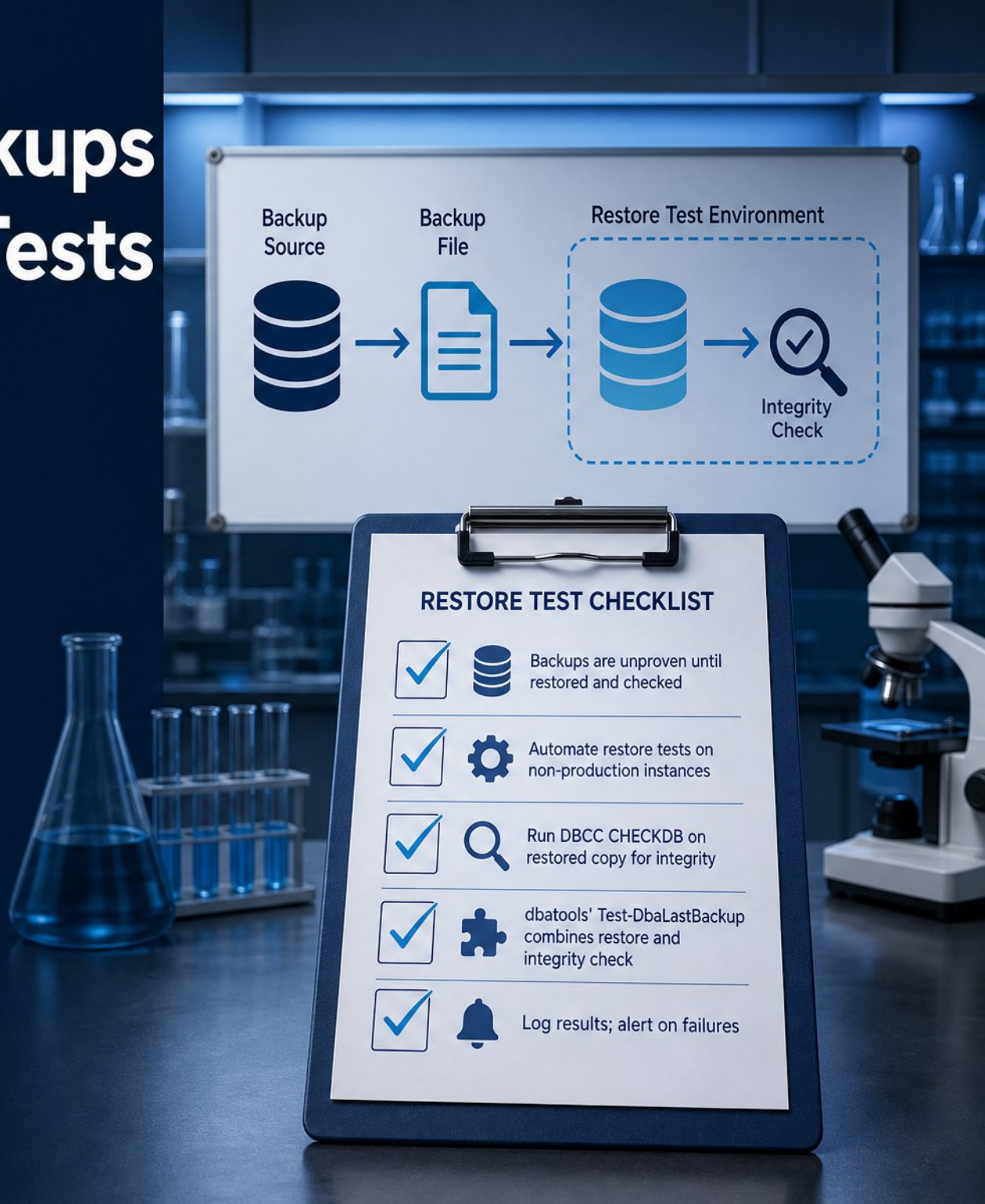
The Restore Workflow You Will Actually Use

- Restore full backup WITH NORECOVERY to apply more files
- Apply differential, then each log backup in order
- Use STOPAT for point-in-time recovery
- Inspect contents first: RESTORE HEADERONLY, FILELISTONLY



Verifying Backups with Restore Tests

- Backups are unproven until restored and checked
- Automate restore tests on non-production instances
- Run DBCC CHECKDB on restored copy for integrity
- dbatools' Test-DbaLastBackup combines restore and integrity check
- Log results; alert on failures



Monitoring Backup Health and Alerting

- Query msdb backup history for success, age, and failures
- Configure SQL Server Agent alerts for job failures
- Alert on overdue backups, for example, no full backup in 24 hours
- Track key metrics: last backup time, size, and duration
- Use built-in reports or third-party tools for proactive monitoring

Common Backup Mistakes and Their Impact

- Backups on same disk as database fail with the database
- Ignored log growth in FULL recovery causes disk-full outages
- Untested backups lead to failed recoveries in disasters
- System database backups (master, msdb, model) are often missed

Learning from Real Backup Failures

- Rogue third-party log backups break restore chains with LSN mismatches
- Service account permissions denied on backup folders
- Full transaction logs block snapshot-based backups
- Corruption hidden until restore attempts fail
- Verify with sp_CheckBackup or DBCC CHECKDB first

Putting It All Together: Your Backup Playbook

- Review the full checklist: recovery models, backup types, schedules
- Document RPO and RTO commitments signed off by stakeholders
- Write the recovery runbook with steps, contacts, and locations
- Test restores regularly, timing them against the RTO target
- Automate restore tests and drills to build confidence

PersonWise.

PersonWise • AI digital-human course platform

Scan the code or click anywhere to watch this course live, with voice Q&A

personwise.ai/t/48867ebd/public