

Code Layout

- > Only perform writes via helper funcs, never update models directly
- > Put all transactions in one file for easier auditing & testing

banking/transactions.py:

```
def transfer_money(src, dst, amt):  
    with transaction.atomic():  
        ...  
  
def merge_accounts(user_a, user_b):  
    with transaction.atomic():  
        ...  
  
def archive_account(user):  
    with transaction.atomic():  
        ...
```

```
from banking.transactions import transfer_money  
  
...
```

Storage Layer Concerns

- > The database can't guarantee data integrity on its own
- > Bit flips are common, use ECC RAM (and ZFS!)
- > Streaming replication or snapshots to do offsite-backups

OFFSITE BACKUPS. OFFSITE BACKUPS. SET UP YOUR OFFSITE BACKUPS.
OFFSITE BACKUPS. OFFSITE BACKUPS. OFFSITE BACKUPS.
OFFSITE BACKUPS. OFFSITE BACKUPS. OFFSITE BACKUPS.
OFFSITE BACKUPS. OFFSITE BACKUPS. OFFSITE BACKUPS.
OFFSITE BACKUPS. OFFSITE BACKUPS. OFFSITE BACKUPS.
OFFSITE BACKUPS. OFFSITE BACKUPS. OFFSITE BACKUPS.
OFFSITE BACKUPS. OFFSITE BACKUPS. OFFSITE BACKUPS.
OFFSITE BACKUPS. OFFSITE BACKUPS. OFFSITE BACKUPS.
OFFSITE BACKUPS. OFFSITE BACKUPS. OFFSITE BACKUPS.
OFFSITE BACKUPS. OFFSITE BACKUPS. OFFSITE BACKUPS.
OFFSITE BACKUPS. OFFSITE BACKUPS. OFFSITE BACKUPS.
OFFSITE BACKUPS. OFFSITE BACKUPS. OFFSITE BACKUPS.
OFFSITE BACKUPS. OFFSITE BACKUPS. OFFSITE BACKUPS.
OFFSITE BACKUPS. OFFSITE BACKUPS. OFFSITE BACKUPS.
OFFSITE BACKUPS. OFFSITE BACKUPS. OFFSITE BACKUPS.
OFFSITE BACKUPS. OFFSITE BACKUPS. OFFSITE BACKUPS.

