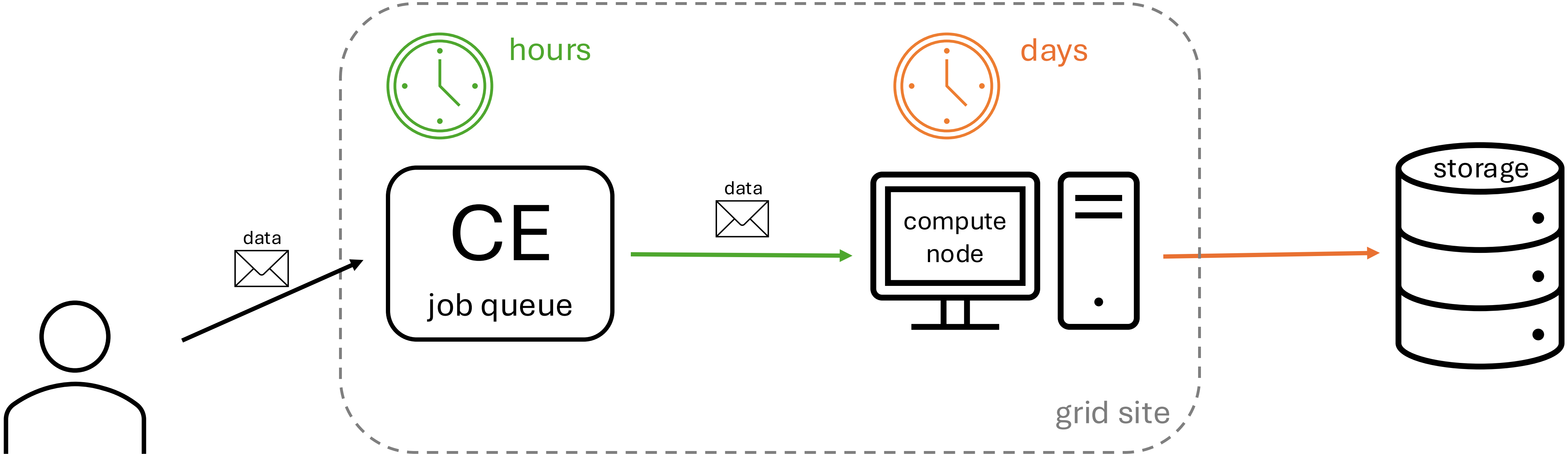
Token Lifetime for Queued Scenarios

February 4th, 2025

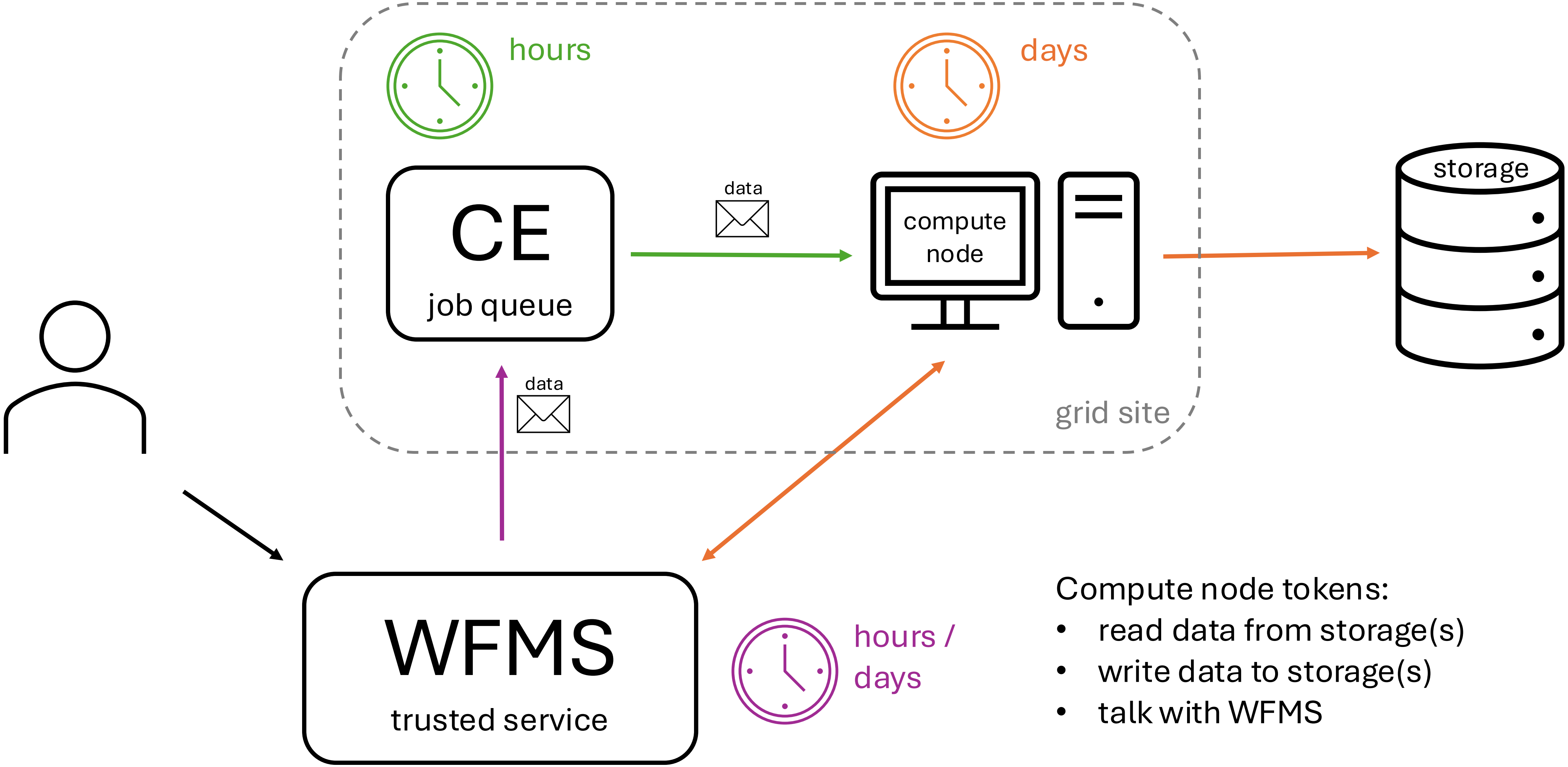
Grid batch job processing



Compute node tokens:

- read data from storage(s)
- write data to storage(s)

WLCG batch job processing



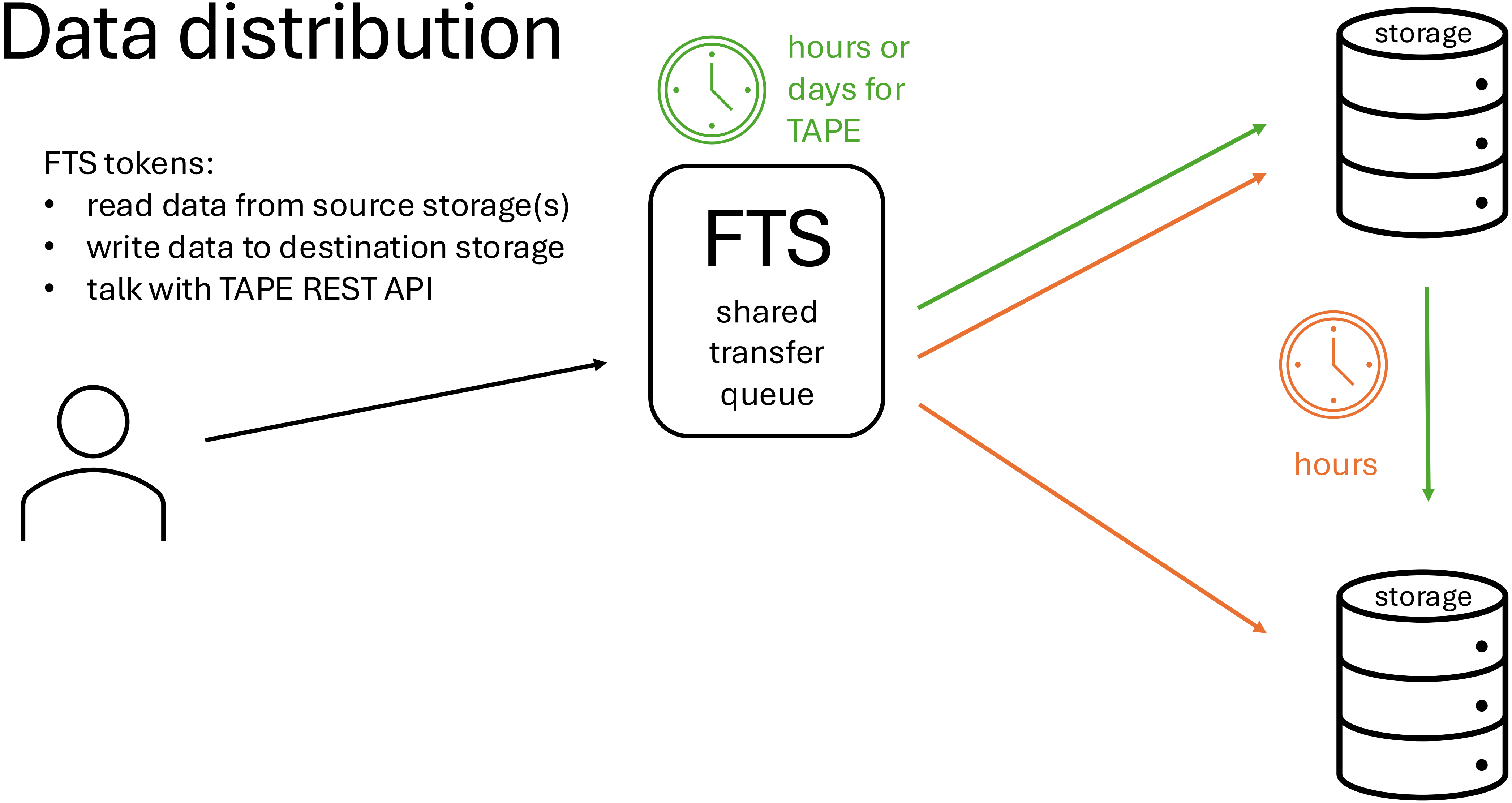
Compute node tokens:

- read data from storage(s)
- write data to storage(s)
- talk with WFMS

Data distribution

FTS tokens:

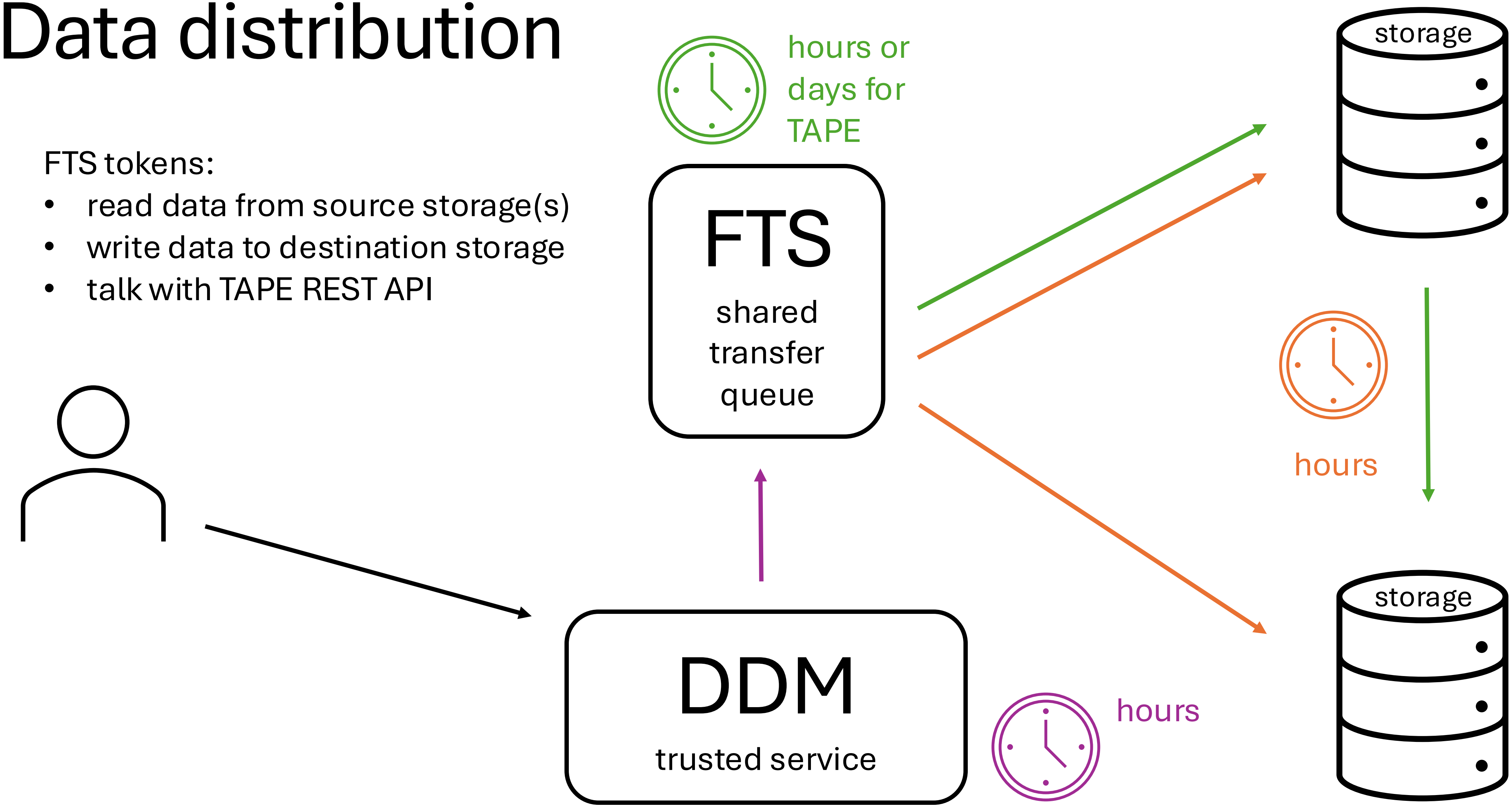
- read data from source storage(s)
- write data to destination storage
- talk with TAPE REST API



Data distribution

FTS tokens:

- read data from source storage(s)
- write data to destination storage
- talk with TAPE REST API



hours or days for TAPE

FTS

shared transfer queue

DDM

trusted service

hours

hours

storage

storage

Tokens & async workflows

- Access tokens lifetime ~ minutes
- Experiments generally use long-lived proprietary credentials
    - Queued tasks (hours / days)
    - Task processing (hours / days)
- OIDC tokens
    - Long lived access tokens not allowed (token profile lifetimes)
    - Ersatz client would kill token issuer (performance)
    - Refresh tokens not good idea and could still cause troubles to issuer
- WLCG experiments – millions of transfers and jobs per day
- Are these "queued" use-cases completely outside of OIDC tokens scope / does it mean software developers always needs to come with their own original solutions

BACKUP

Design

- GlideIn WFMS
- ALICE / VOBox(?)
- Panda WFMS
- FTS
    - token-exchange mode
    - long-lifetime mode for source and destination AT
        - dedicated AT sub-issuer (avoid performance problems)