



# SRM 2.2 in General

Patrick Fuhrmann

support and funding by



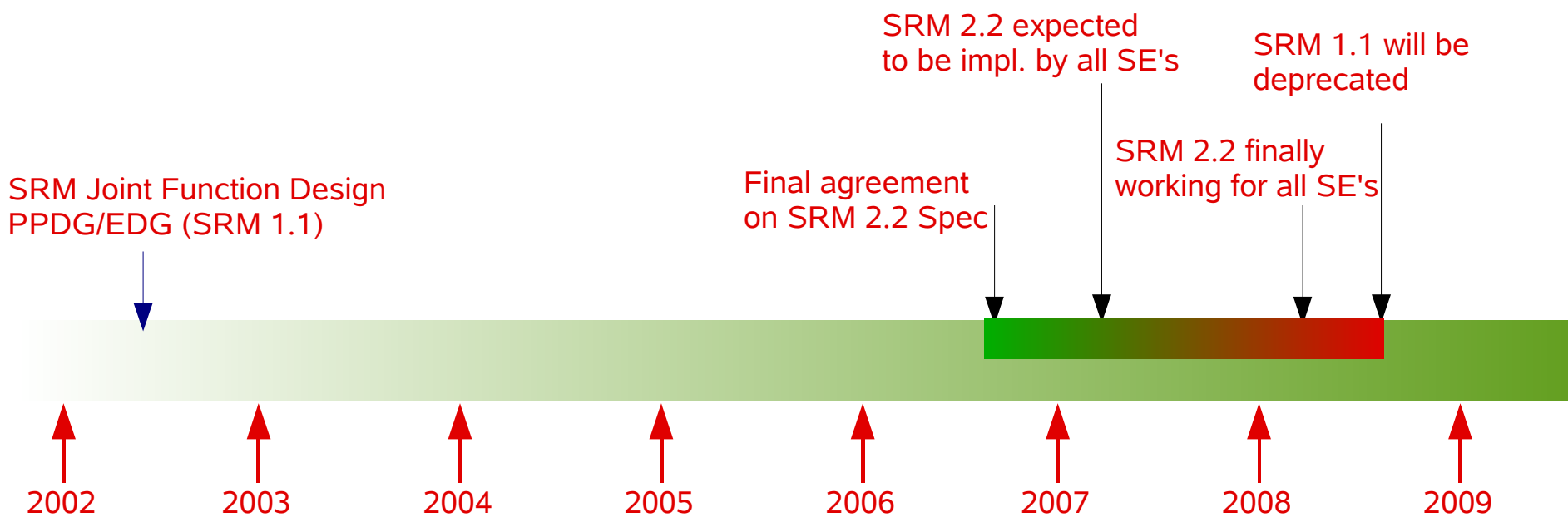


# SRM : The LCG Storage Management Story



dCache.ORG

dCache.ORG



Please find all the details at : <http://sdm.lbl.gov/srm-wg/>





# Preliminary Remarks :

The SRM 2.2 behavior described here, reflect the WLCG MoU only. The original specification provides more options, features ....



# *SRM, Space Attributes and Space Tokens in general.*





# Main features of the SRM 1.1 specification

- Prepares Storage Element to receive data and provides 'Transfer URL' to be subsequently used for the transfer. (see later)
- Prepares a Storage Element to deliver data and provides 'Transfer URL' to be subsequently used for the transfer. (see later)
- Negotiates data transfer protocol(s) with the SRM client.
- Limited file name space operations.
- *Transferring data is NOT part of the protocol specification.*



*Files have Storage Attributes (see later)*

- *Retention Policy*
- *Access Latency*

*which can be specified when writing a file.*

*A certain amount of space can be reserved and Storage Attributes can be assigned (Space Tokens)*

*Space can be reserved to guarantee space availability*

*Variety of directory name functions*

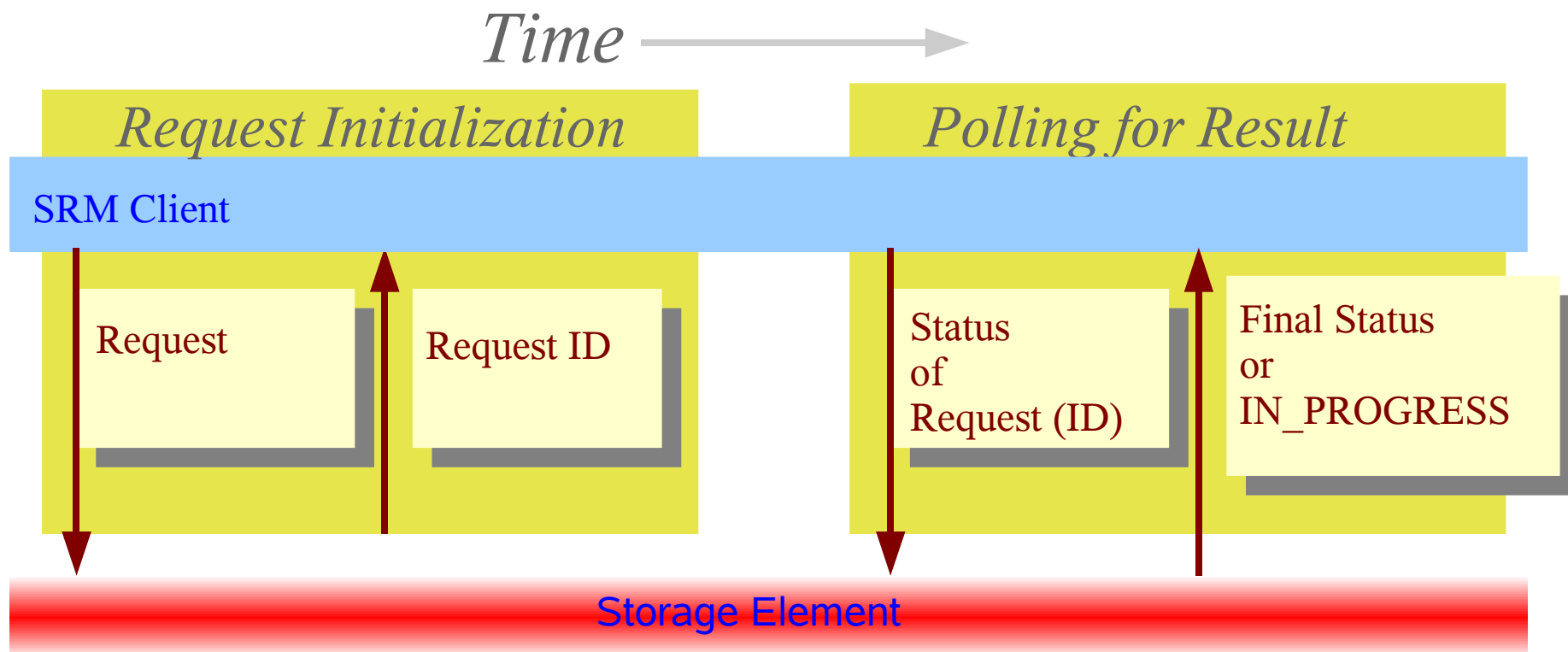


- *Using Space Tokens on read and restore/stage*
- *Full VOMS awareness of SE*
- *Protecting Space Token operations by ACL's*



## SRM asynchronous operations

- The client performs a request for which the server returns an request ID.
- This ID is used to obtain the 'request' result by polling the server.





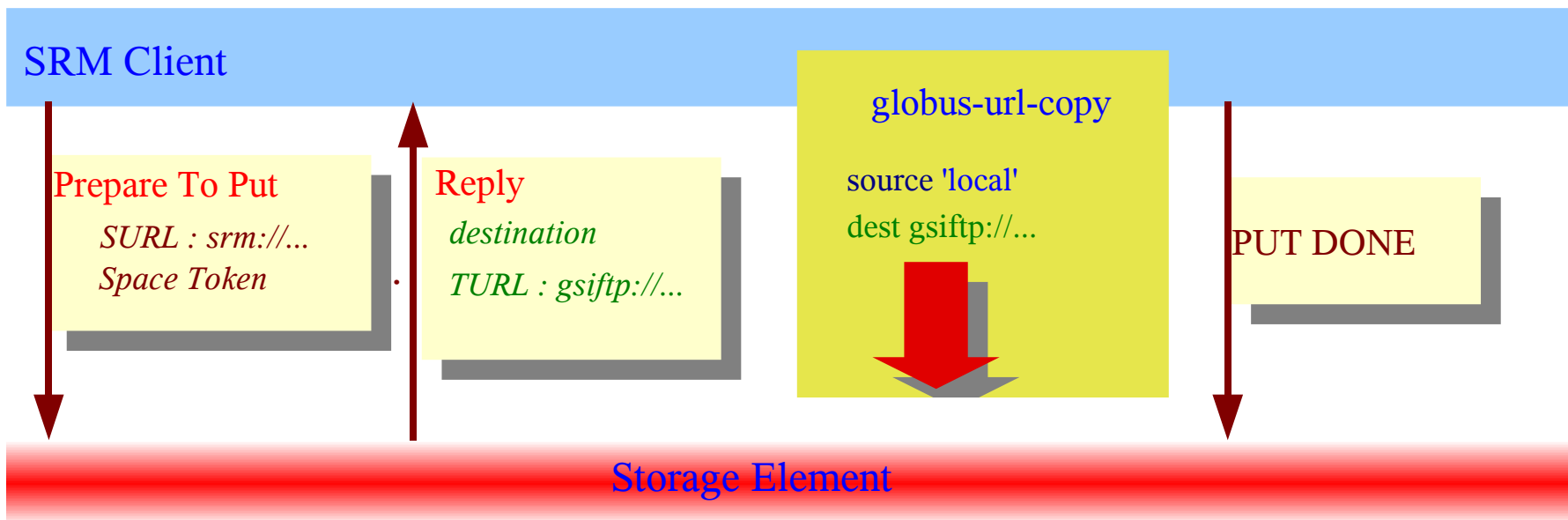


# Initiating data transfers with SRM

## SRM Put

- Client performs SRM Put Request with SURL and storage requirements.
  - which could be **Space Token** or **Access Latency** and **Retention Policy**
- SRM prepares the space in dCache and returns the Transfer URL
- TURL is used by the client to transfer the data into dcache (e.g.: gsiftp://)
- Client has to do a SRM PUT DONE to finish the transaction

Time →



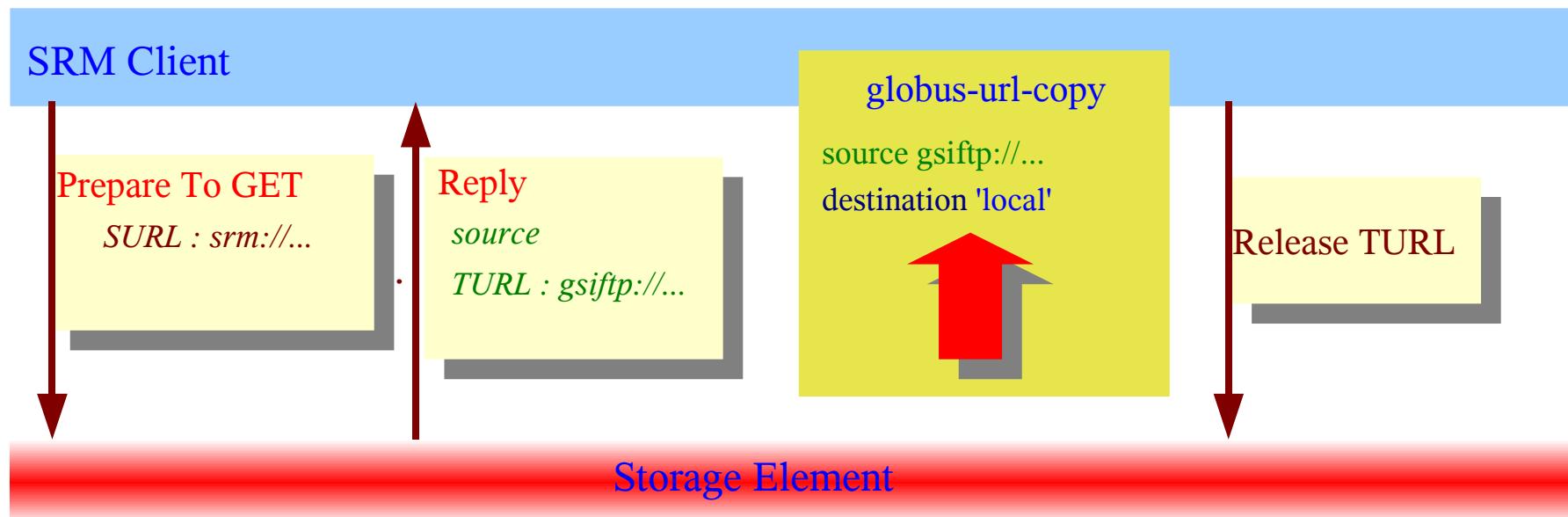


# Initiating data transfers with SRM

## SRM Get

- › Client does SRM Get Request with SURL
- › SRM prepares the file in dCache for transfer and returns the source TURL.
- › TURL is used by the client to transfer the data out of dcache (e.g.: gsiftp://)
- › Client has to do a SRM RELEASE to finish the transaction

Time →



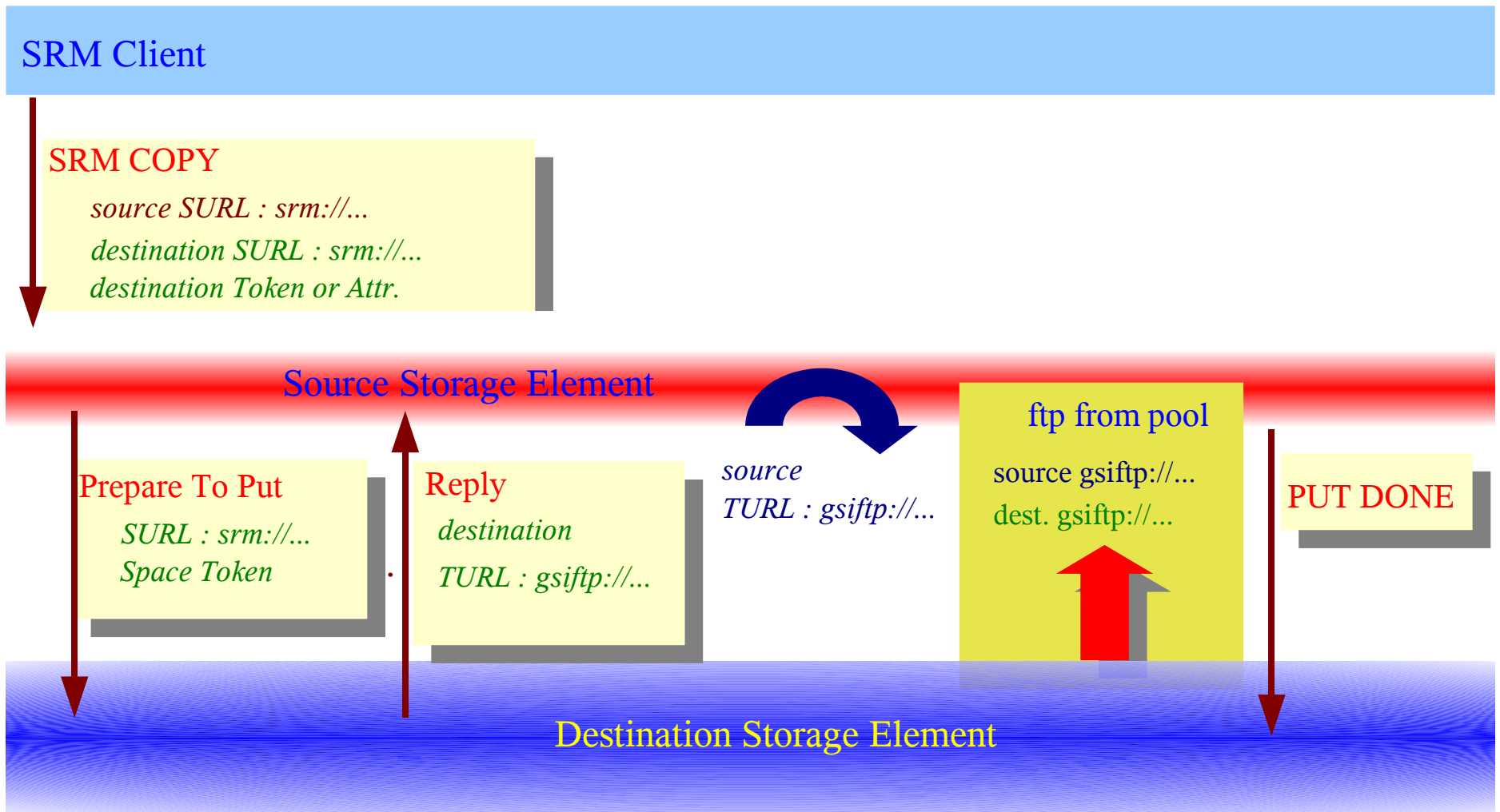


# SRM Copy (Push Mode)

dCache.ORG

dCache.ORG

Time →



GEFÖRDERT VOM



## *SRM 2.2 Properties (Storage Attributes and Reservation)*

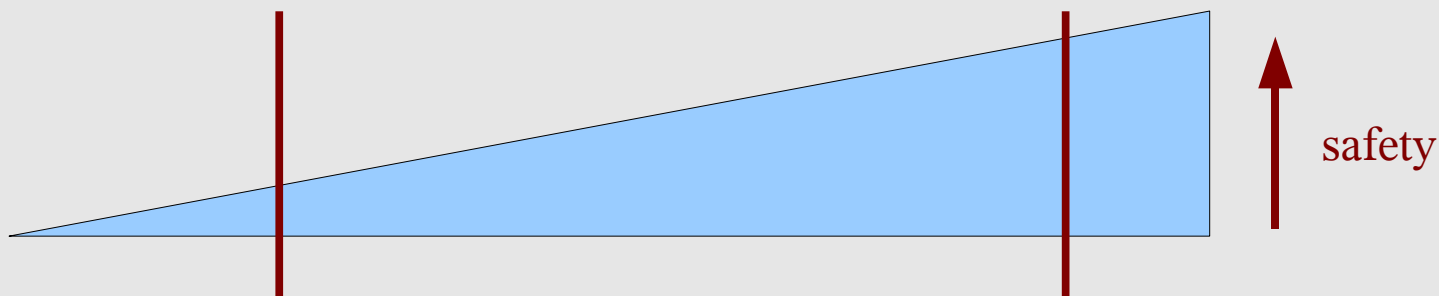


# What are Storage Attributes ?

SRM 2.2 introduces two storage attributes

## Retention Policy

*How safe is my file within the system ?  
or What is the likelihood of a file loss ?*

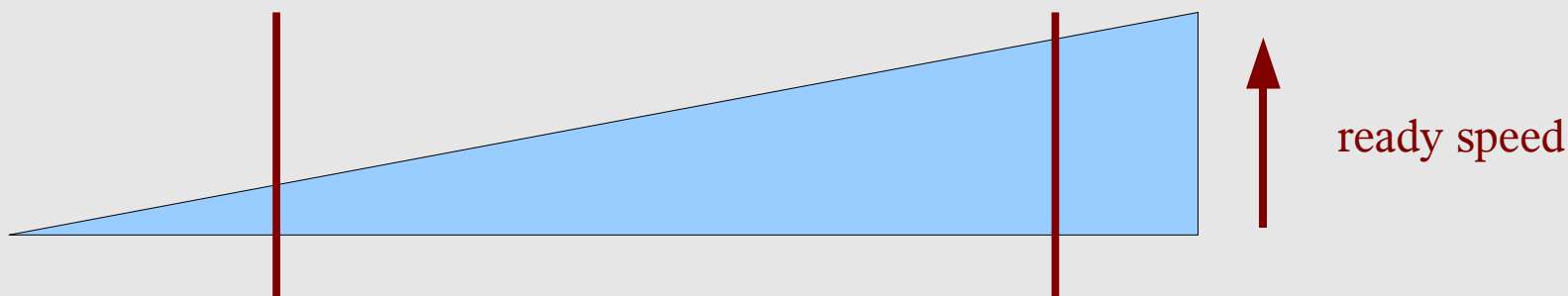


unsafe = **REPLICA** (e.g. JBOD)

Very safe = **CUSTODIAL** (e.g. Tape)

## Access Latency

*How long does it take to make a  
file ready to transfer ?*



fast = **ONLINE** (e.g. Disk)

slow = **NEARLINE** (e.g. Tape)

dCache.ORG

dCache.ORG



# Combining Storage Attributes

dCache.ORG

dCache.ORG

*Developers*

*Users*

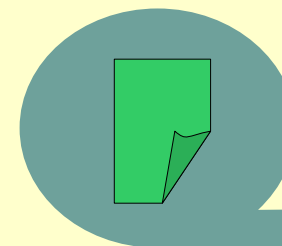
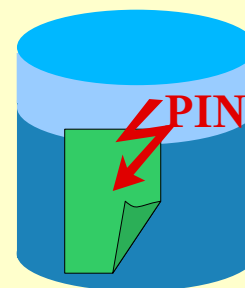
*System*

DISK

TAPE

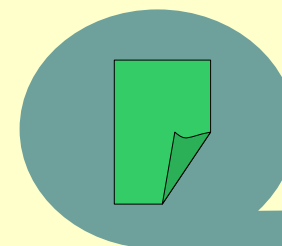
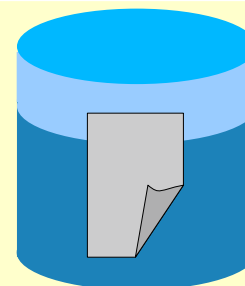
Custodial/Online

T1D1



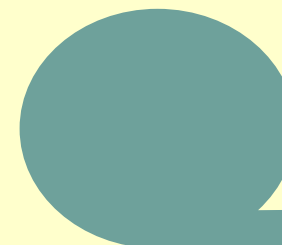
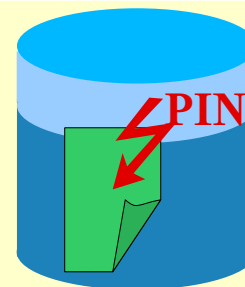
Custodial/Nearline

T1D0



Replica/Online

T0D1



GEFÖRDERT VOM



- With SRM 2.2, storage attributes can be specified when using srm-prepare-to-put, prepare-to-get and bring-online.
- In the current MoU, SE's are **ONLY** required to honor storage attributes on srm-prepare-to-put.
- Those attributes become a property of the file/copy after being written.
- If no attributes are specified, defaults depend on the SE implementation.
- From the SRM specifications point of view, SRM2.2 Storage Attributes may or may not be related to physical areas on the disk system. e.g: A single disk system may hold files with different storage attributes.



# What are Space Tokens ?

Space Tokens have **two** independent purposes ?

- *Space Tokens* represent a **reserved disk space** in the system. SRM-Prepare-to-Put requests, using a *Space Token*, are rejected after the space is completely filled with files.
- *Space tokens* have **Storage Attributes** attached. Files written, using a *Space Token*, inherit those Storage Attributes.

Space Tokens Descriptions ?

- Space Tokens may have human readable *Space Token Descriptions* attached. (e.g. ATLASMCDISK)
- The same *Space Token Description* may be used for one or more Space Tokens.
- SRM Prepare-I/O commands work on *Space Tokens* and not on *Space Token Descriptions*.





# Reserved Spaces and Space Tokens.

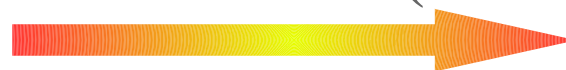
- Writing a file into a Space Token reduces the usable space in this token by the amount of the size of the file.
- After a file has been removed from the name space, its size is returned to the Space Token, it belonged to.
- The usable space in a Space Token, representing the Custodial/Nearline Space Attributes, will grow again, after a file has been migrated to tape. Though the file may stay on disk until space is needed. (See next slide)



# Complete Nonsense

Flush to tape of custodial/nearline (T1D0)

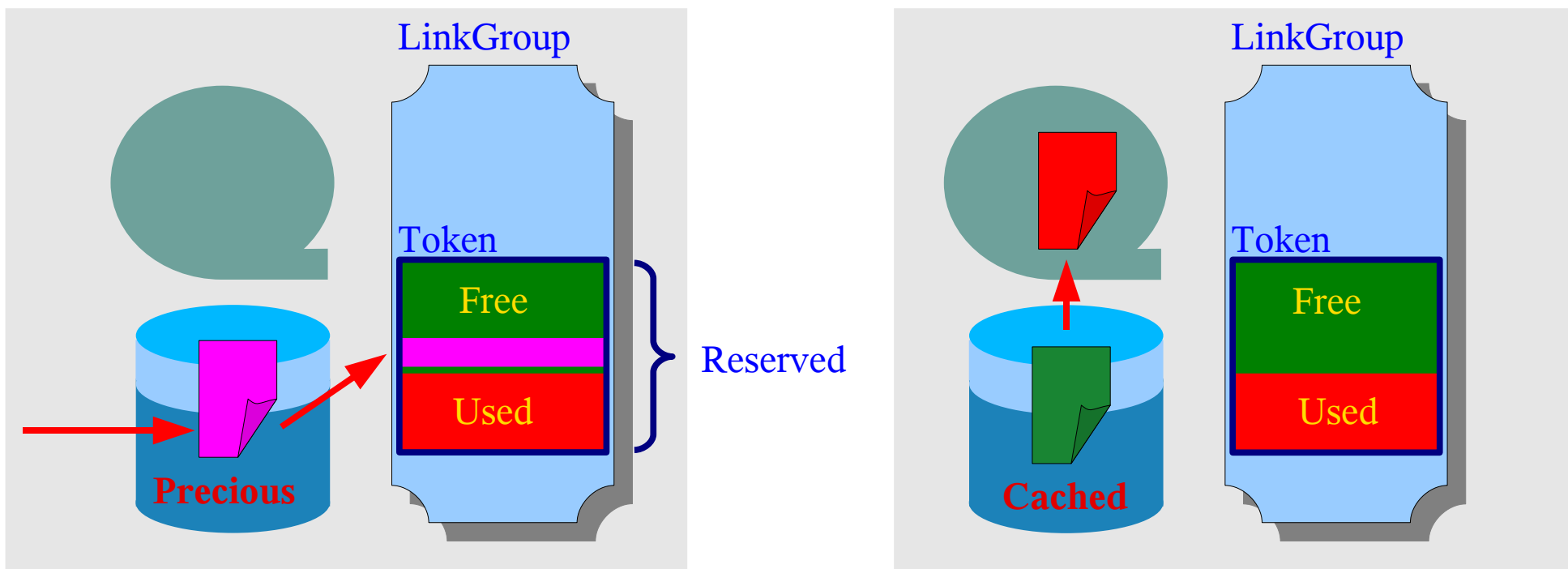
*PUT*



*Flush to tape*

dCache.ORG

dCache.ORG



## SRM 2.2 WLCG Space Management doesn't manage Tape Spaces.

After a T1D0 file has been written to disk and before it is flushed to tape, the state of the file is *precious* and the size of the file is added to *used space* of the space token. After the file has been flushed to tape, the file becomes *cached* and the size of the file is removed again from the *used space* of the token. The file becomes unrelated to the original Space Token.



# *Further reading*

*[www.dCache.ORG](http://www.dCache.ORG)*

dCache.ORG

dCache.ORG

GEFÖRDERT VOM