Replicating Tabular Data with Merkle DAGs

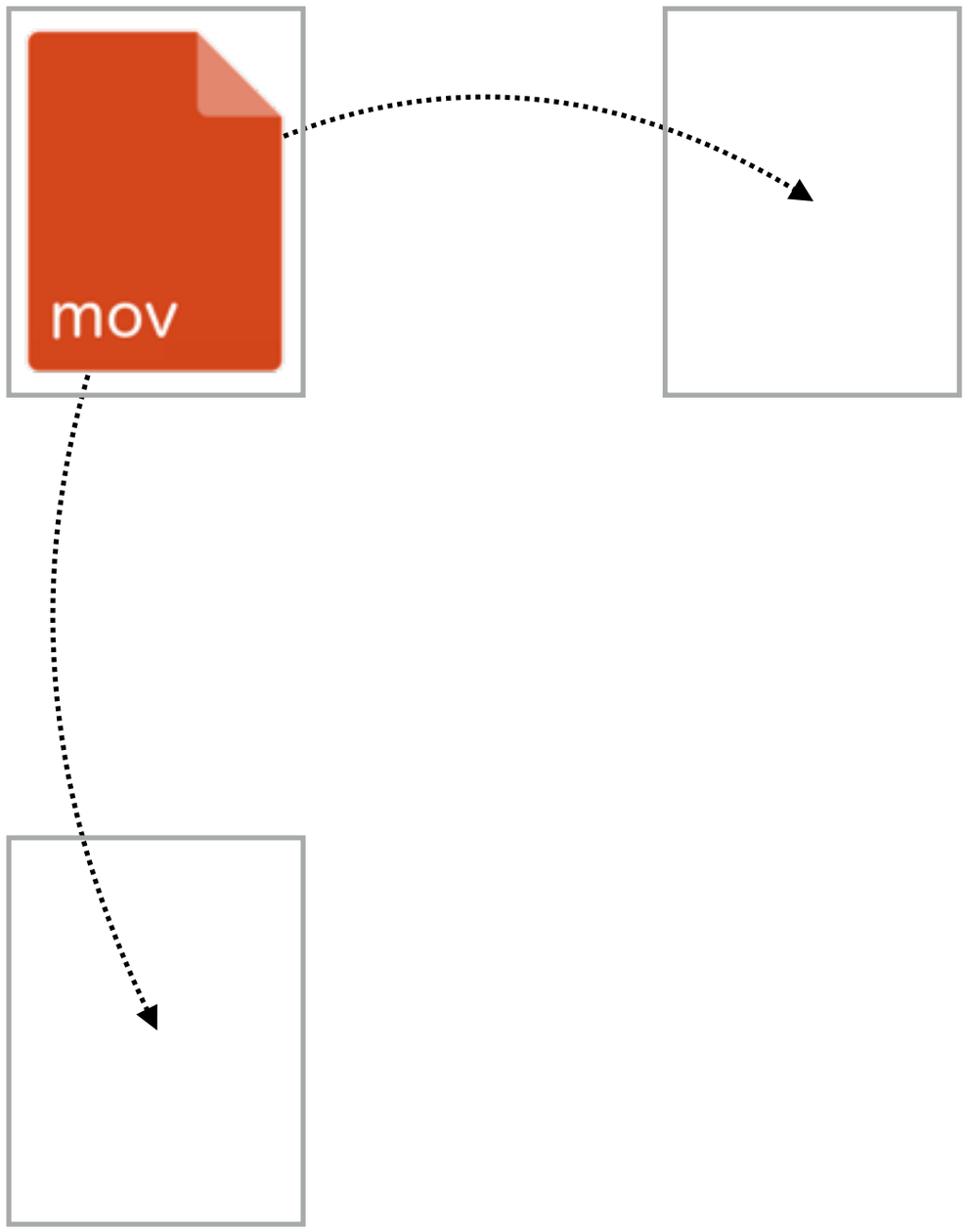
Basics of Data Replication with Merkle DAGs and how we might use SSTables to do it

Prepared by Matt Zumwalt for the dat jawn team 20 April 2016

Replicating Files with bit torrent

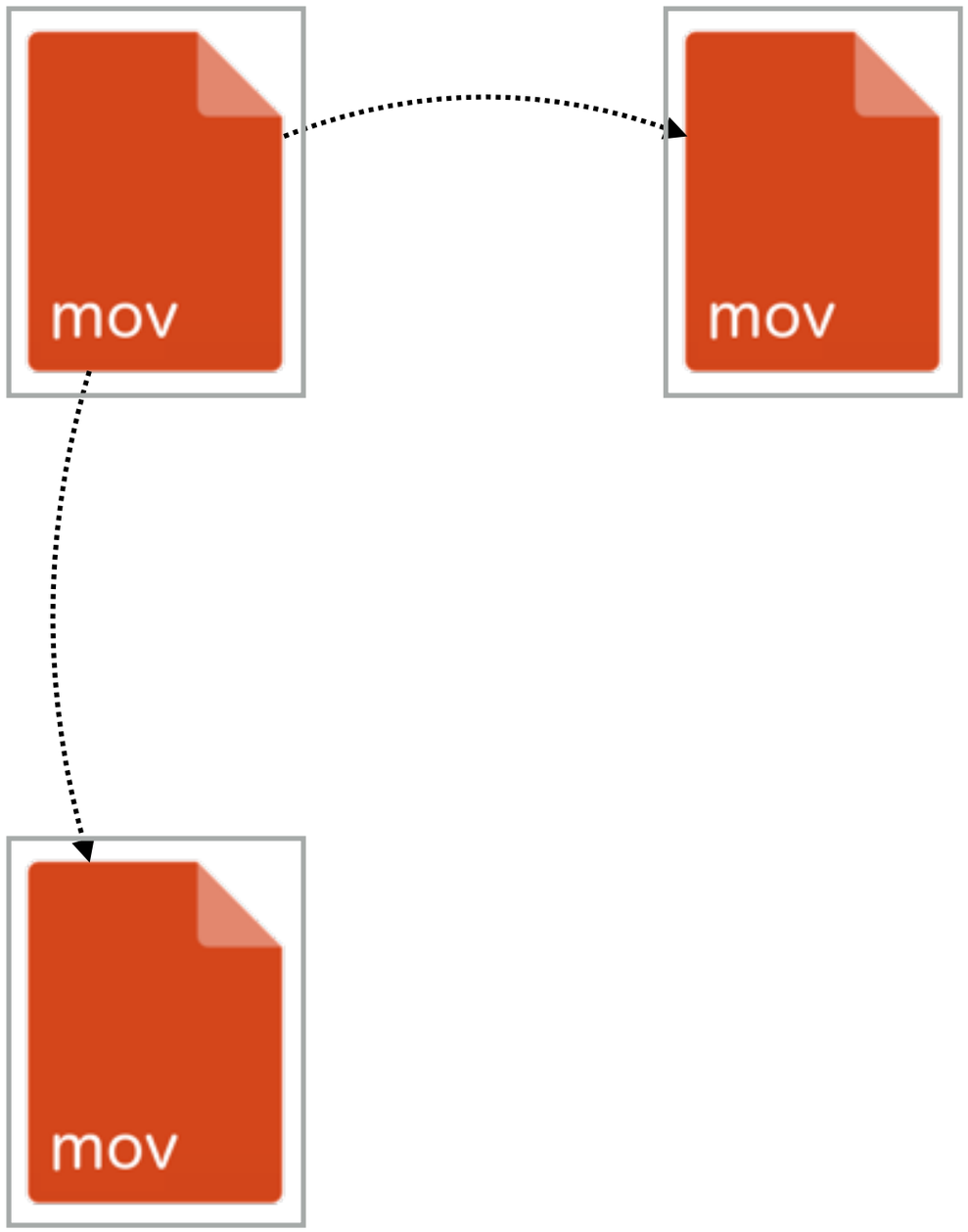
Replicating Files with bit torrent

Bit torrent lets you replicate files



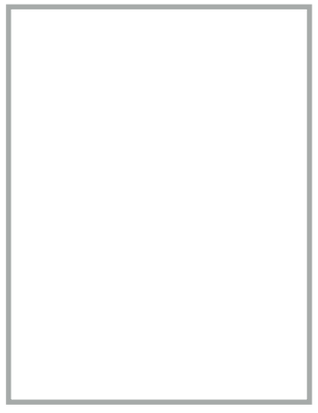
Replicating Files with bit torrent

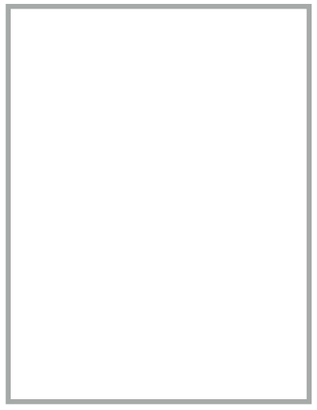
Bit torrent lets you replicate files

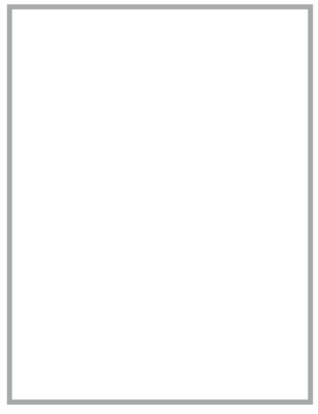


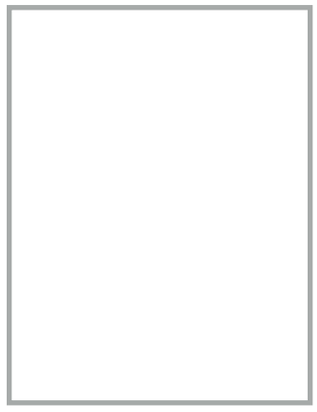
Replicating Files with bit torrent

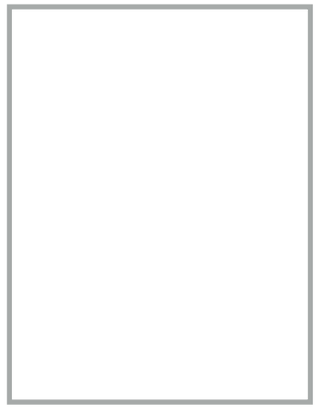
Bit torrent swarms share the burden of replication





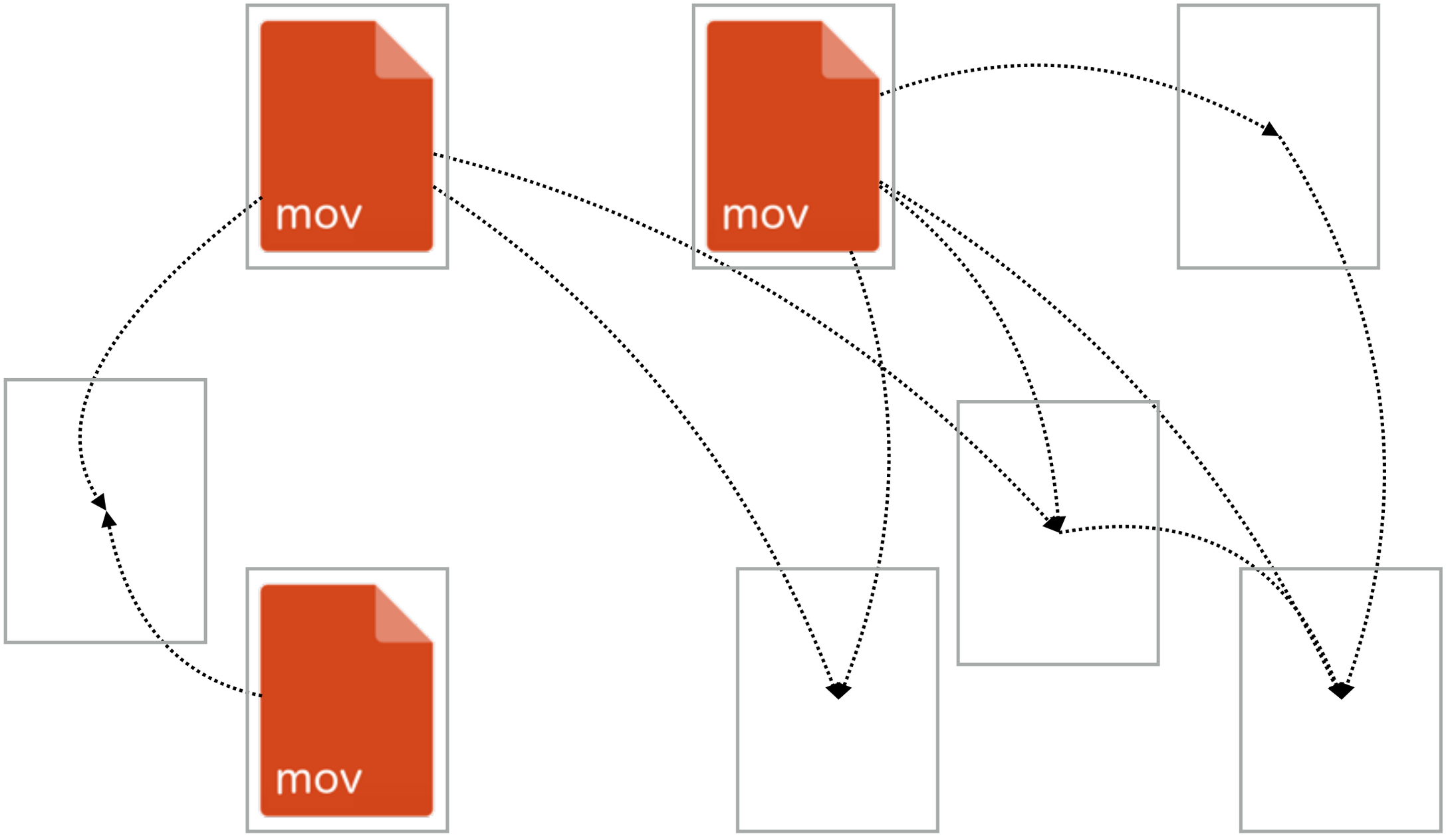






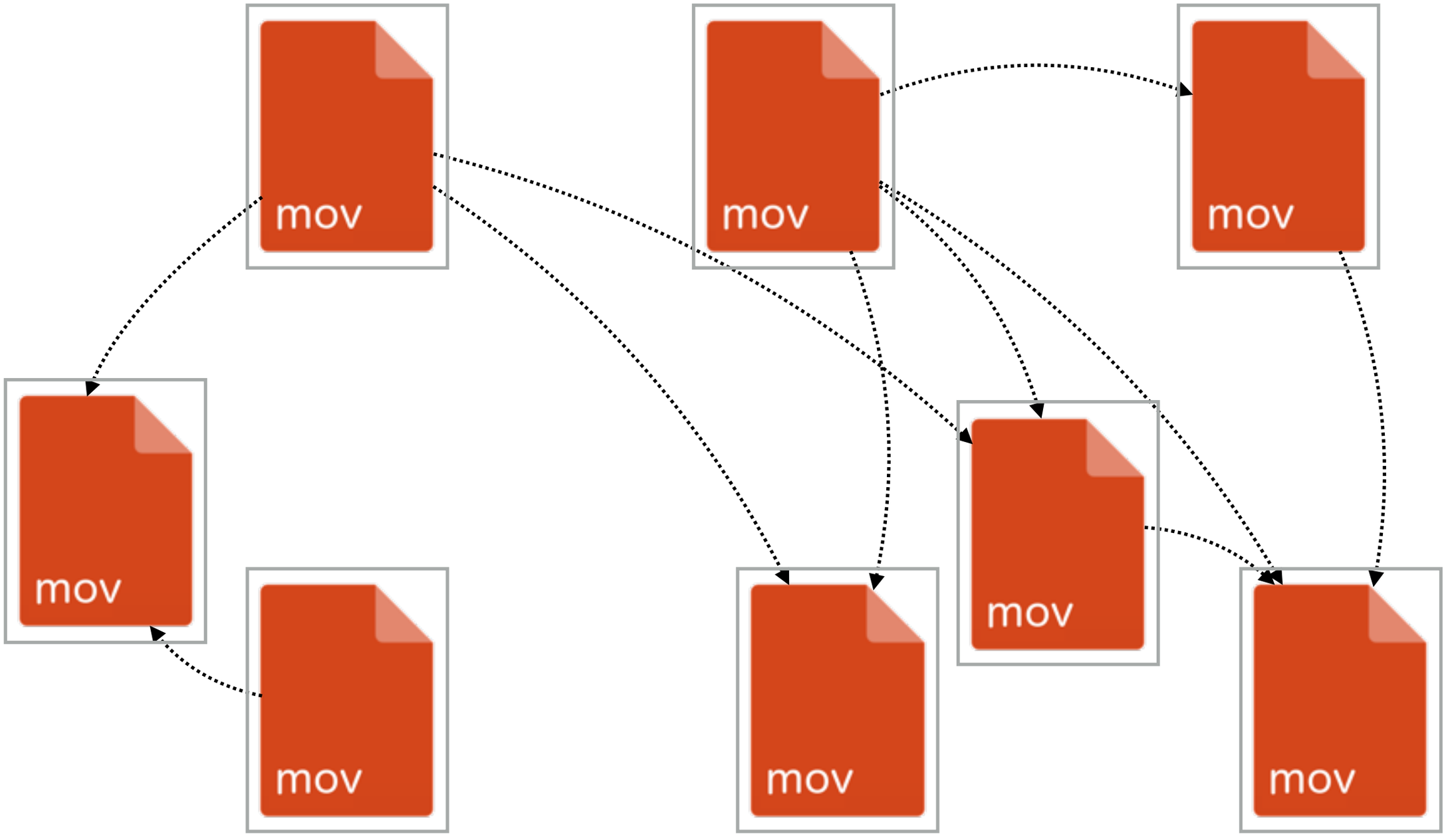
Replicating Files with bit torrent

Bit torrent swarms share the burden of replication



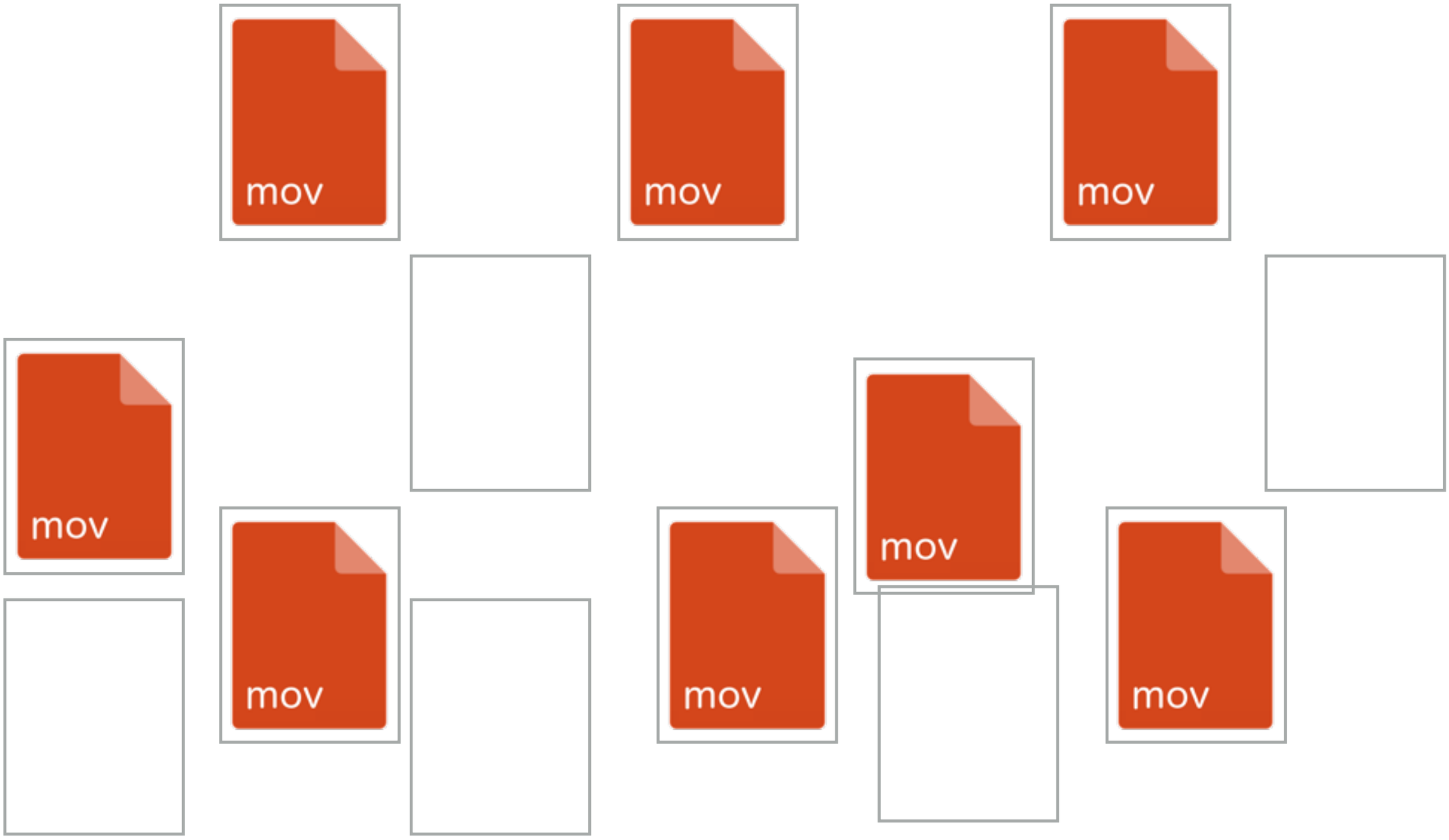
Replicating Files with bit torrent

Bit torrent swarms share the burden of replication



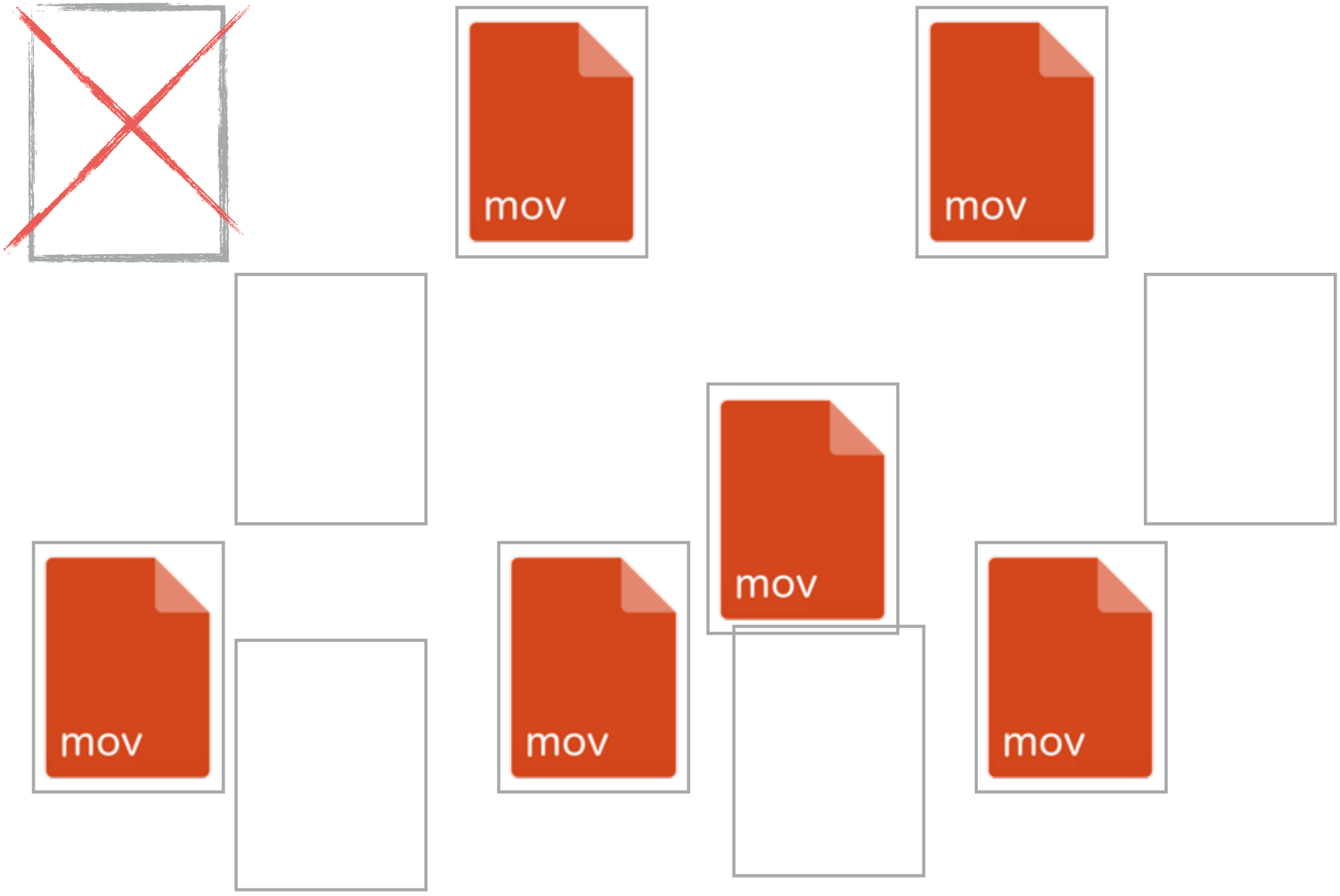
Replicating Files with bit torrent

Bit torrent swarms share the burden of replication



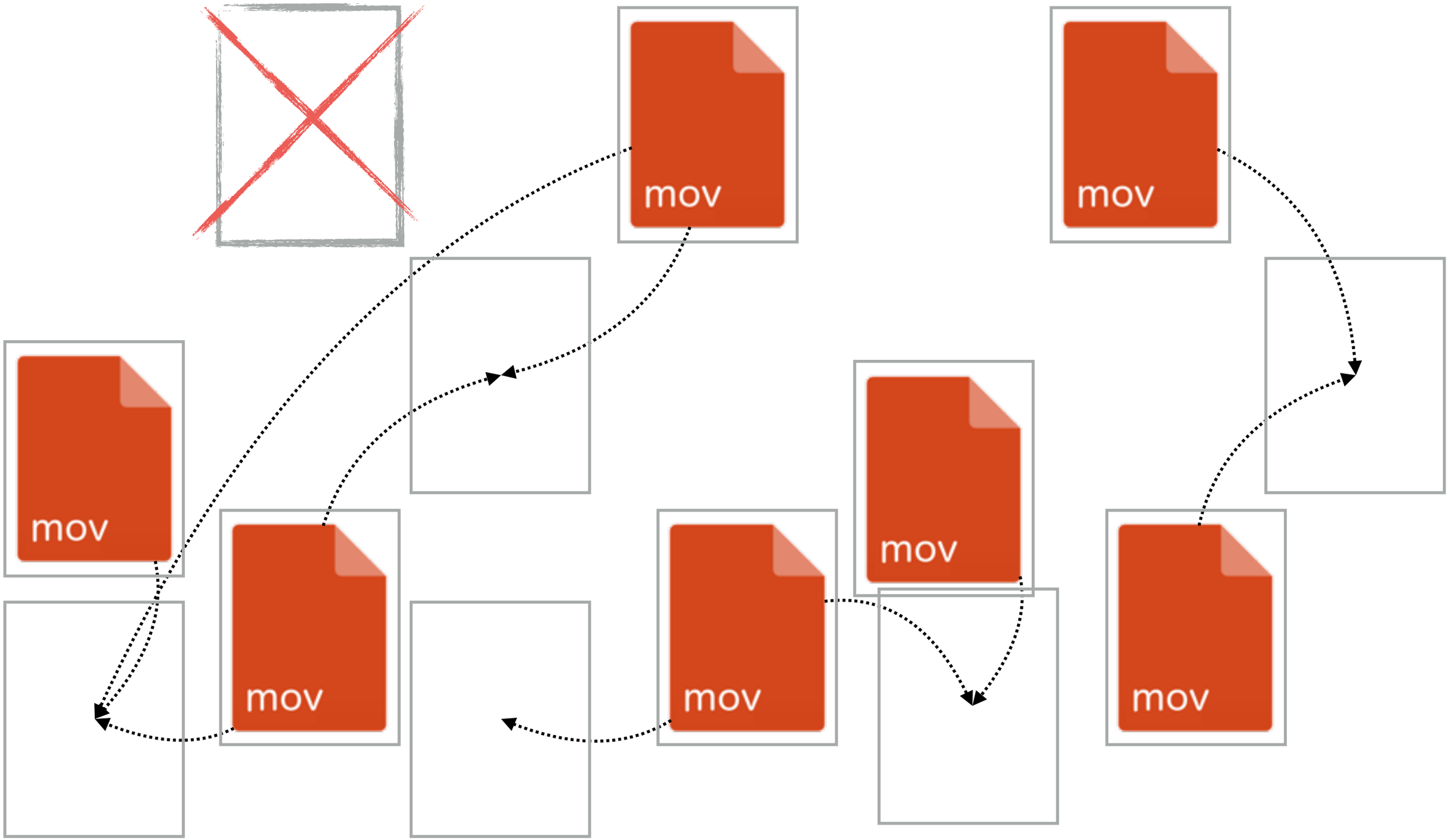
Replicating Files with bit torrent

Even if the original seed goes away



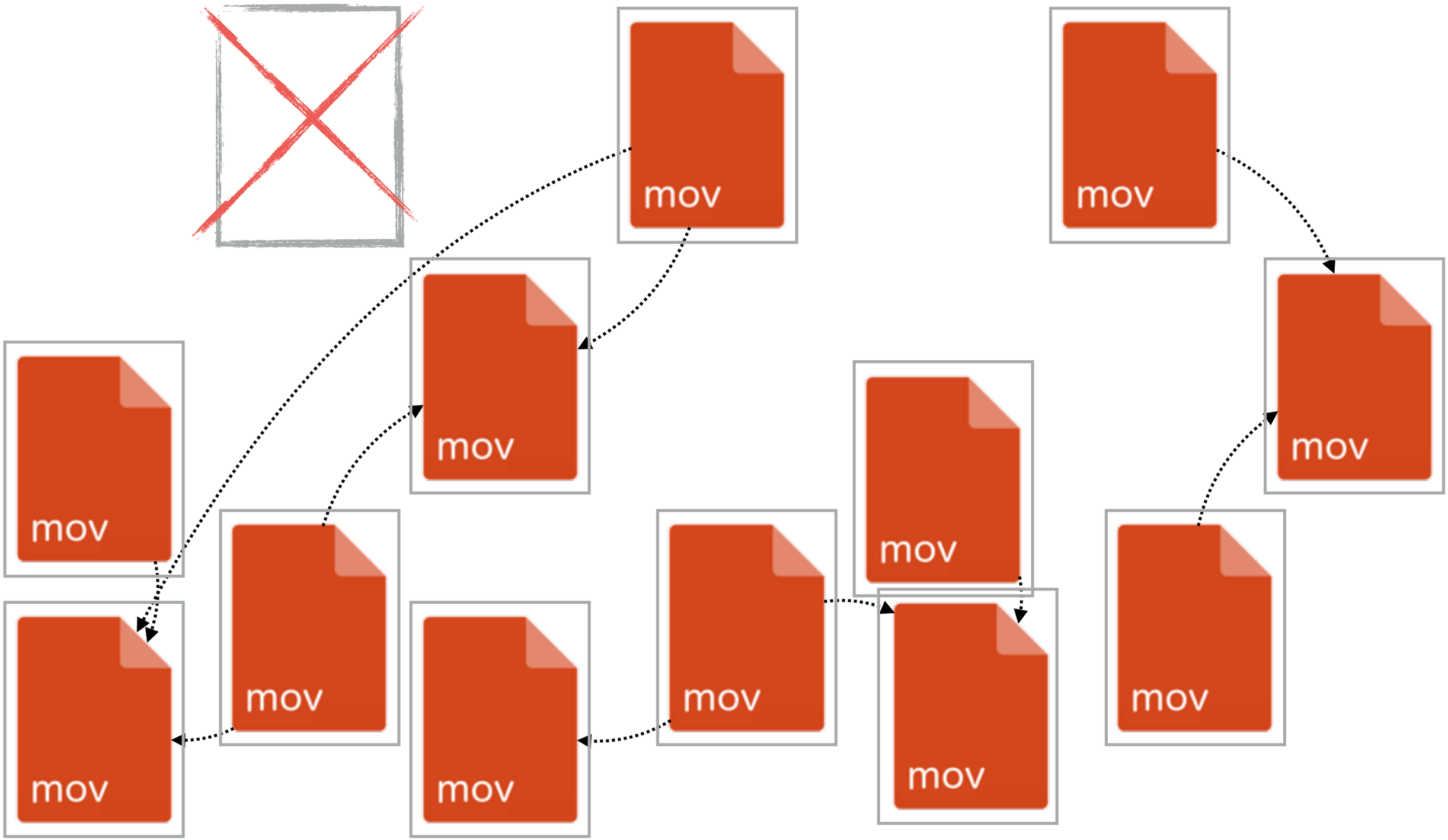
Replicating Files with bit torrent

the swarm can still replicate the file



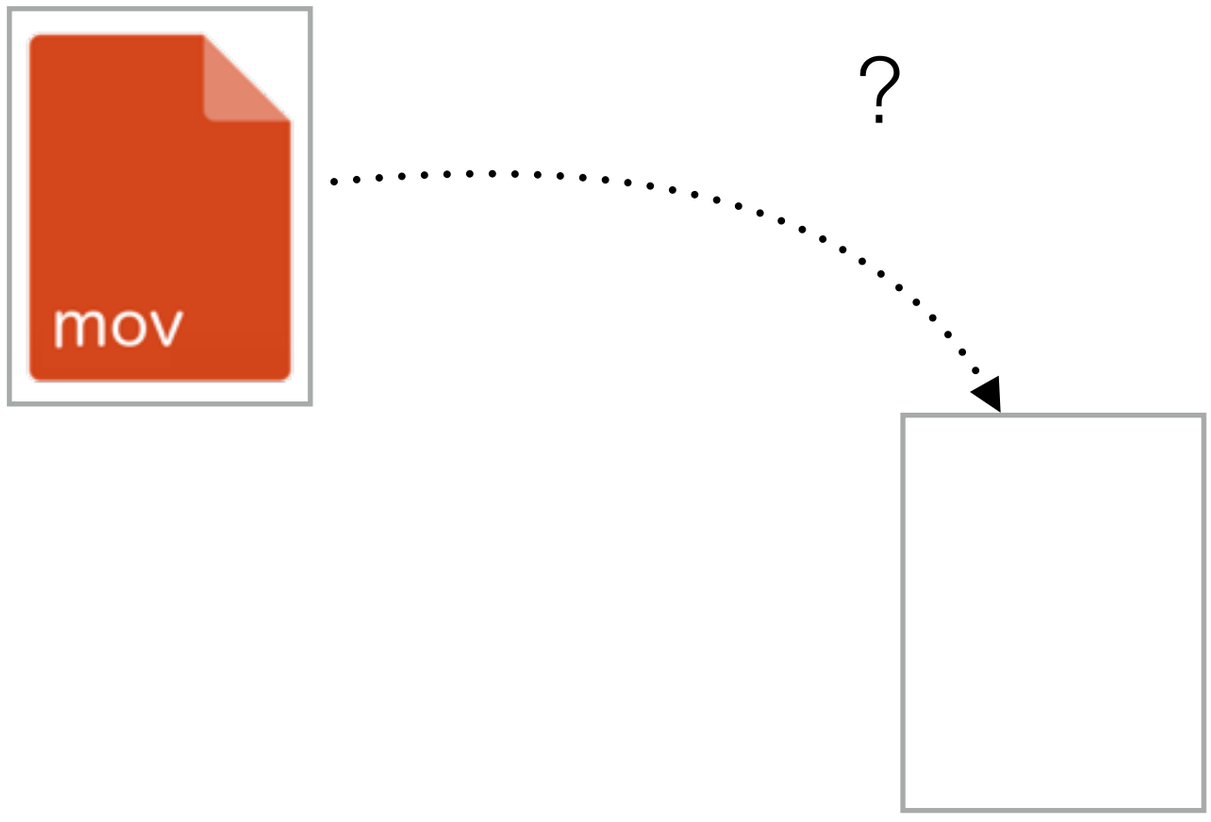
Replicating Files with bit torrent

the swarm can still replicate the file



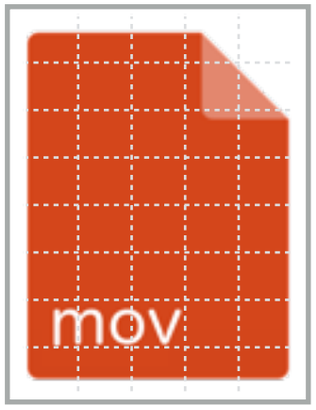
Replicating Files with bit torrent

How does bit torrent replicate a file?



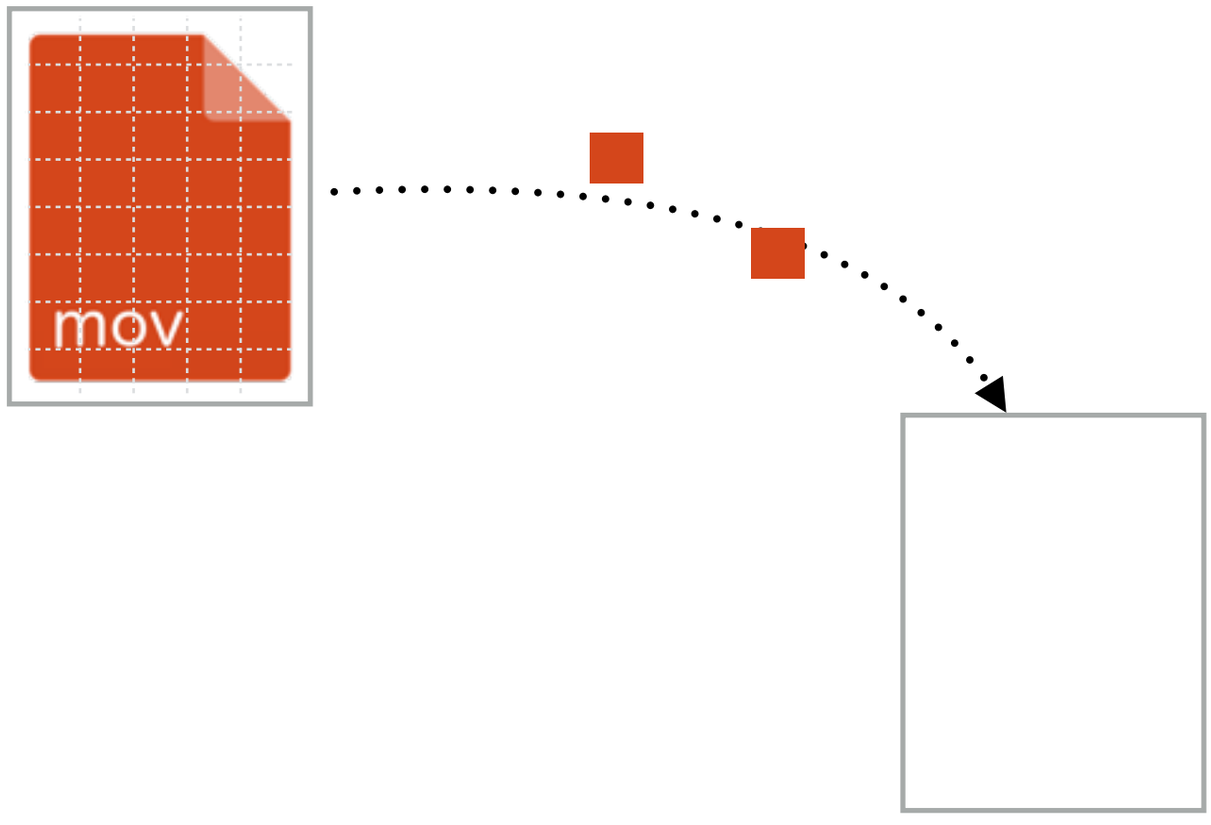
Replicating Files with bit torrent

Bit torrent breaks the file into “pieces”



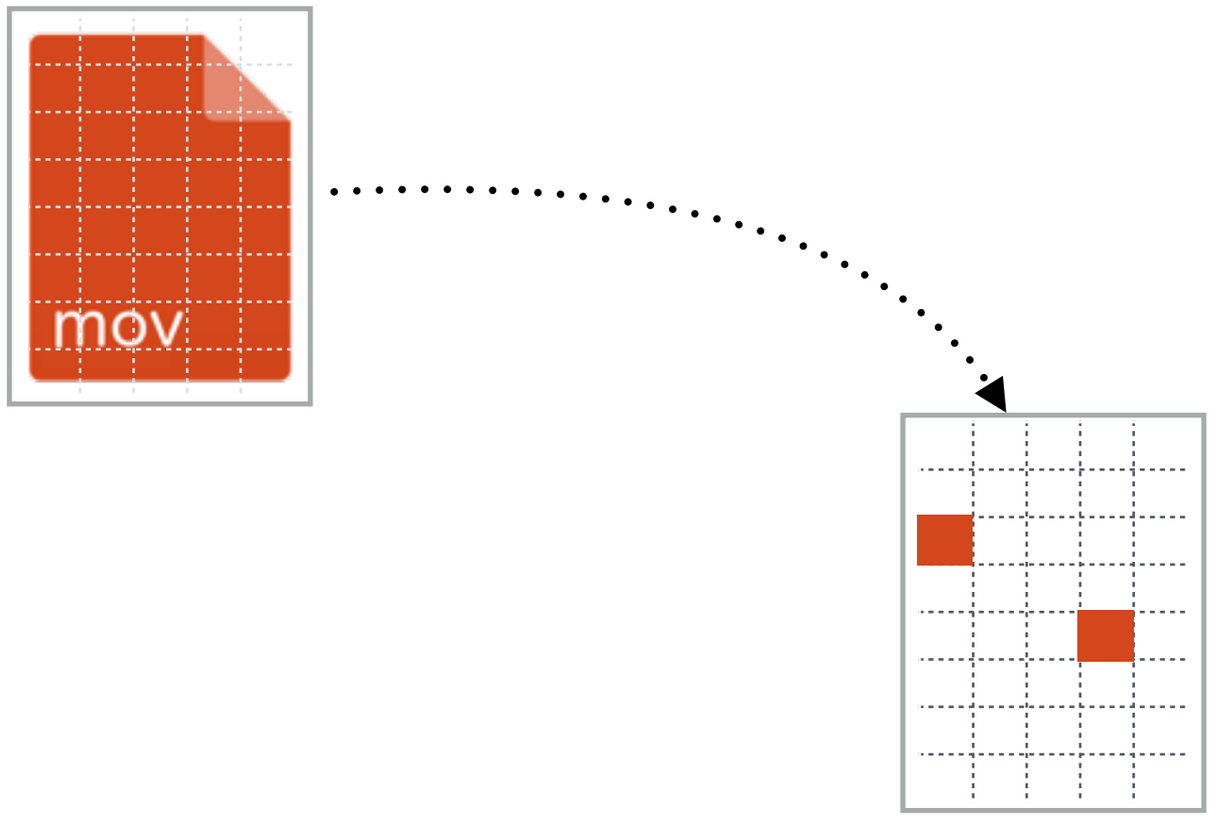
Replicating Files with bit torrent

The bit torrent protocol lets you pass pieces

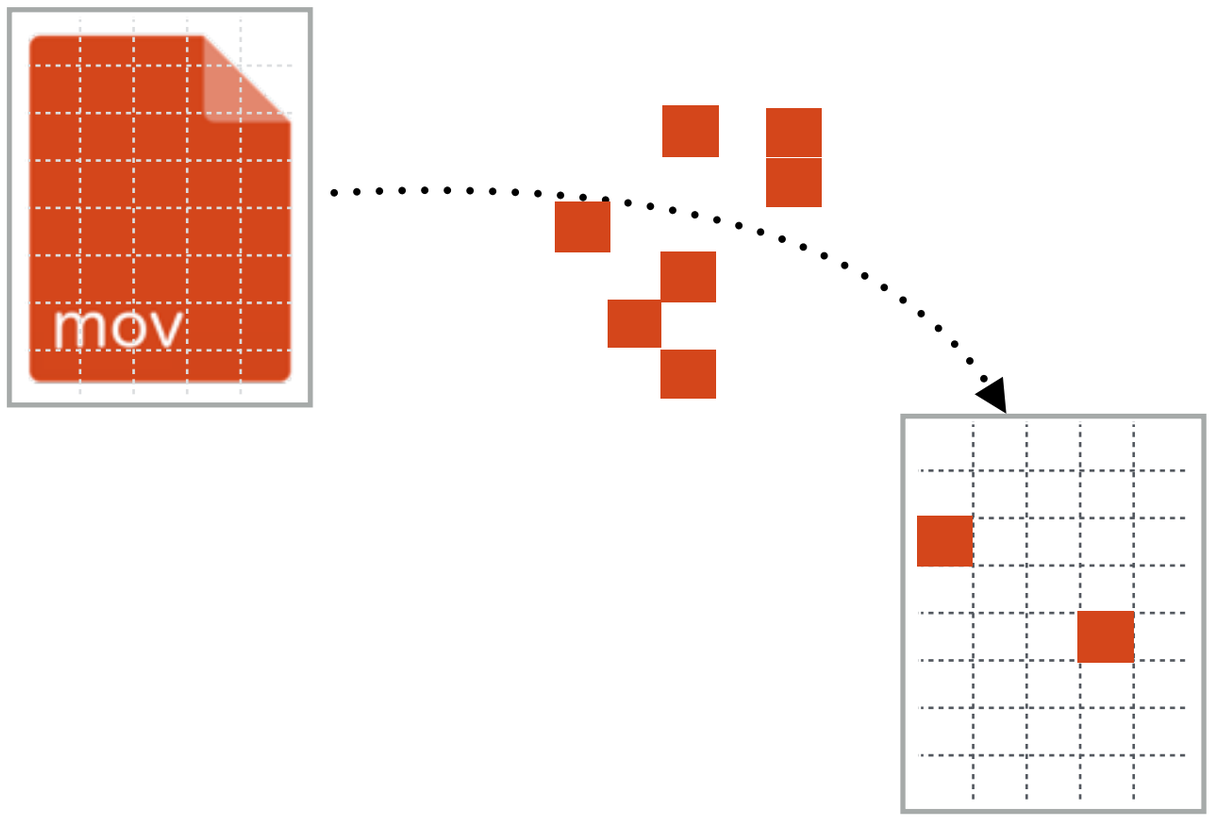


Replicating Files with bit torrent

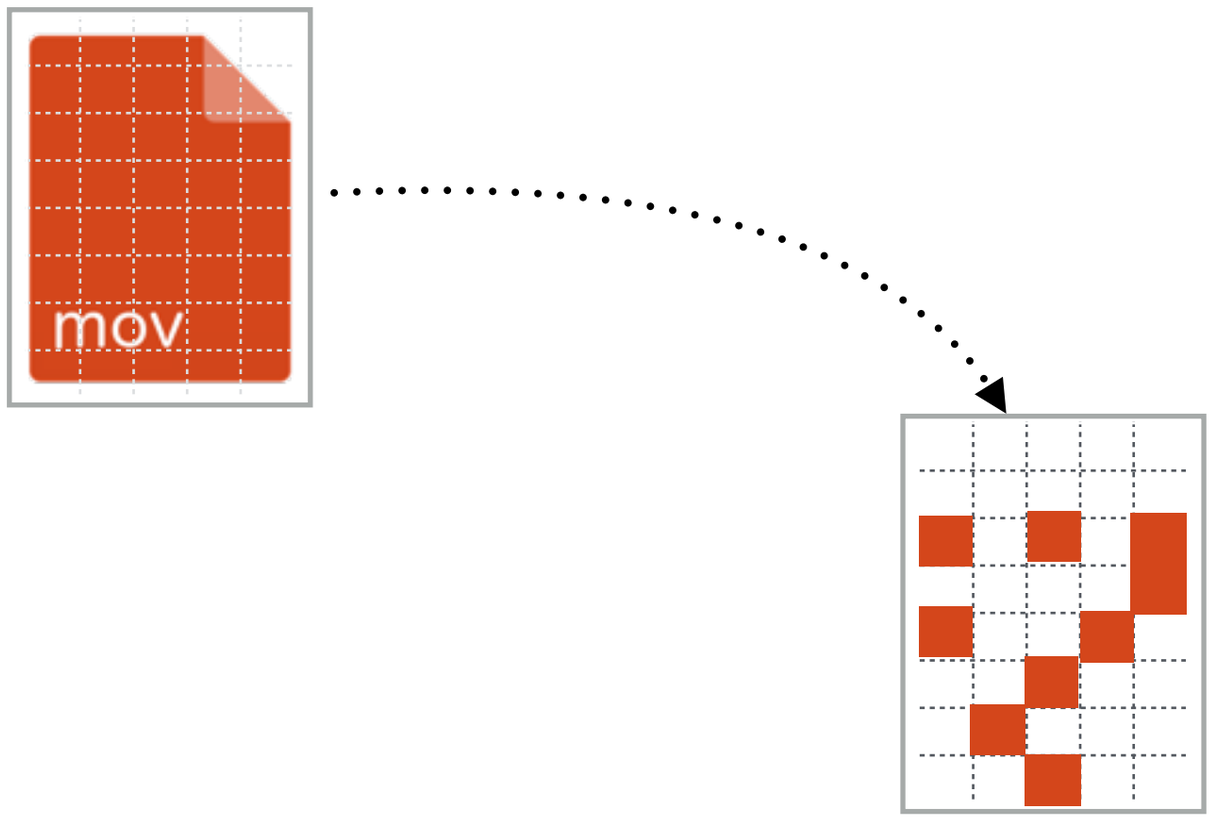
and use those pieces to assemble the file



Replicating Files with bit torrent

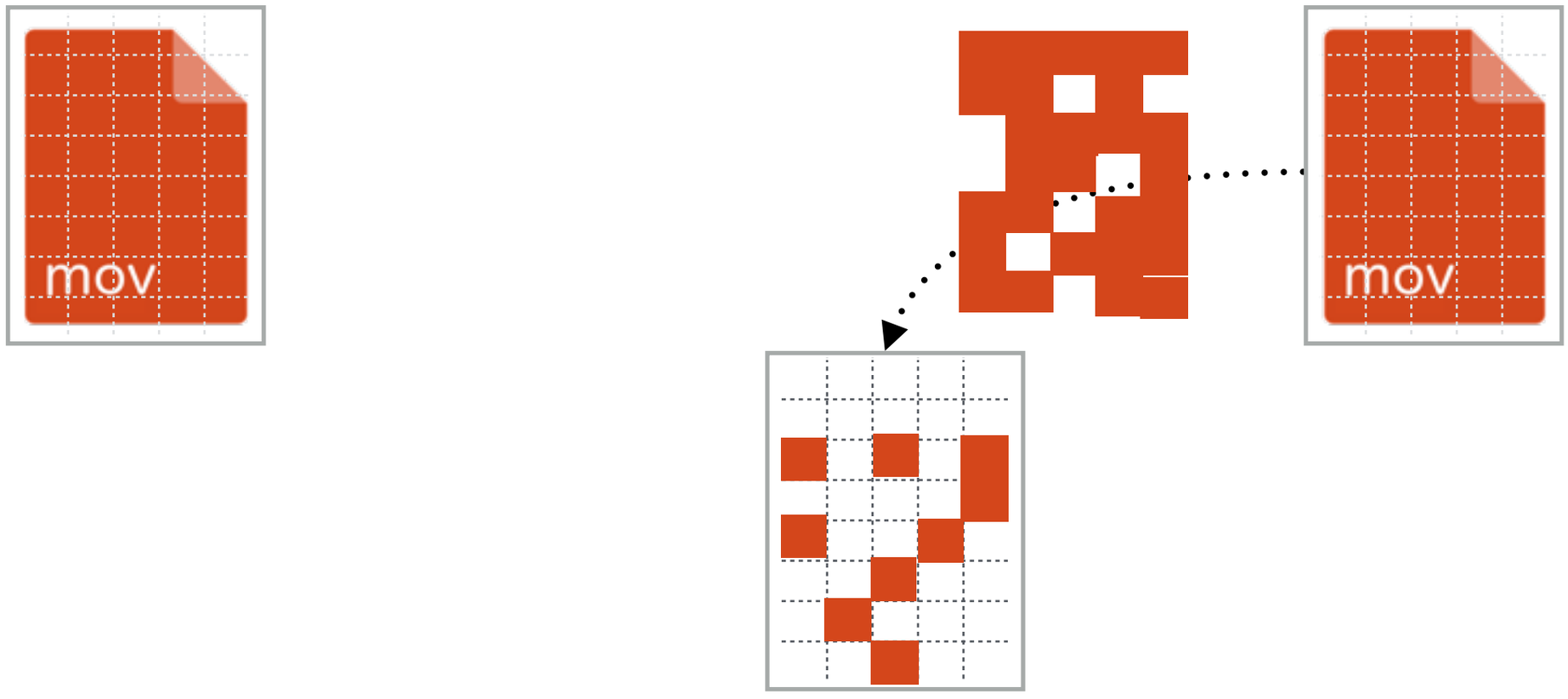


Replicating Files with bit torrent



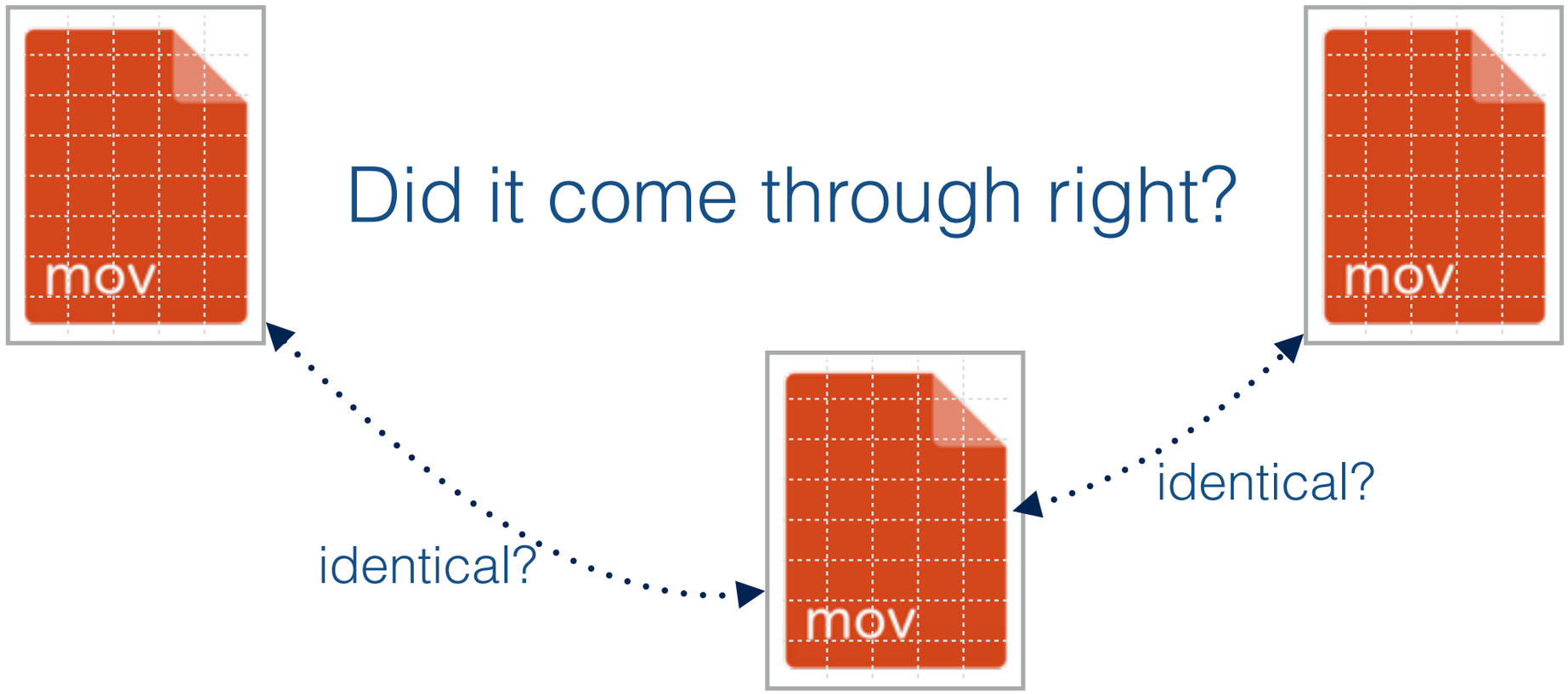
Replicating Files with bit torrent

Any peer can provide pieces



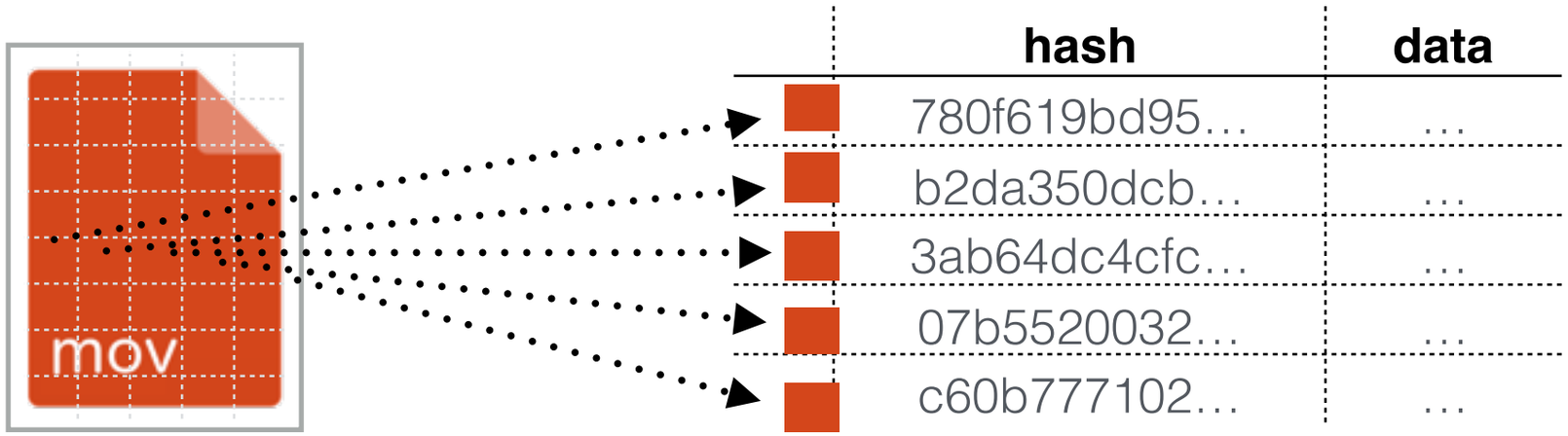
Replicating Files with bit torrent

How do we know if they're identical?



Replicating Files with bit torrent

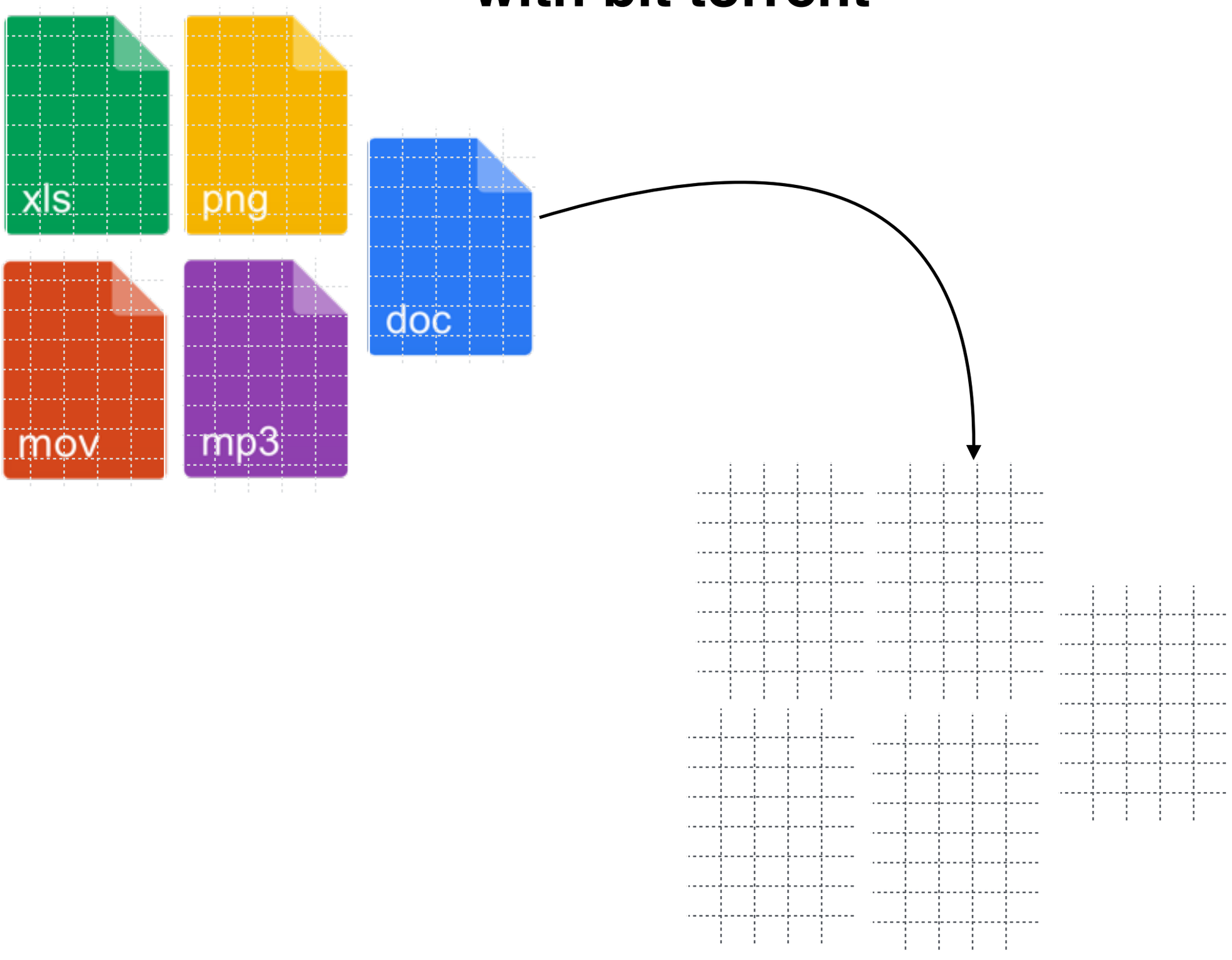
Bit torrent calculates the cryptographic hash for each piece, storing them as a *hash table*



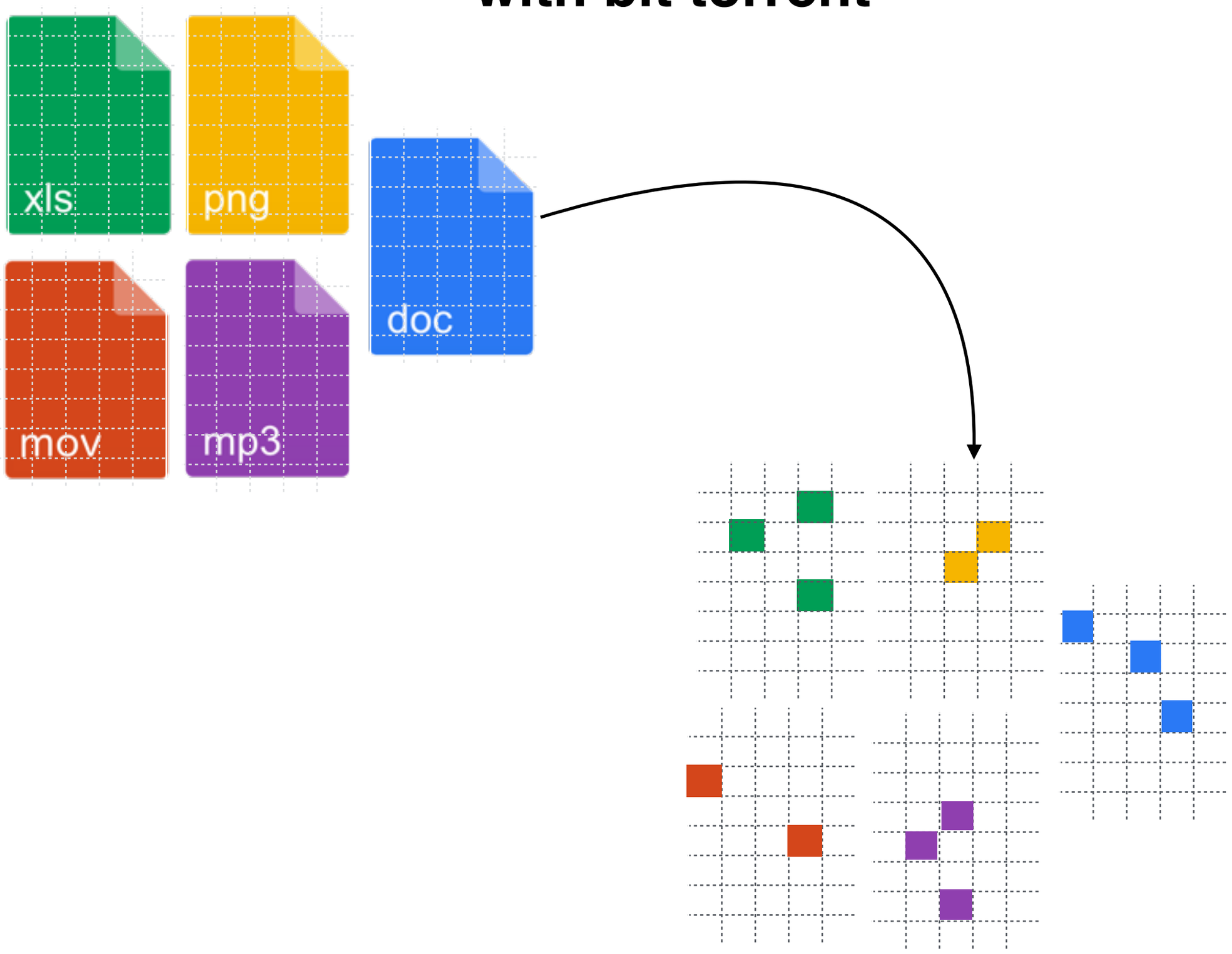
…so you can validate it when it arrives.

Replicating Files with bit torrent

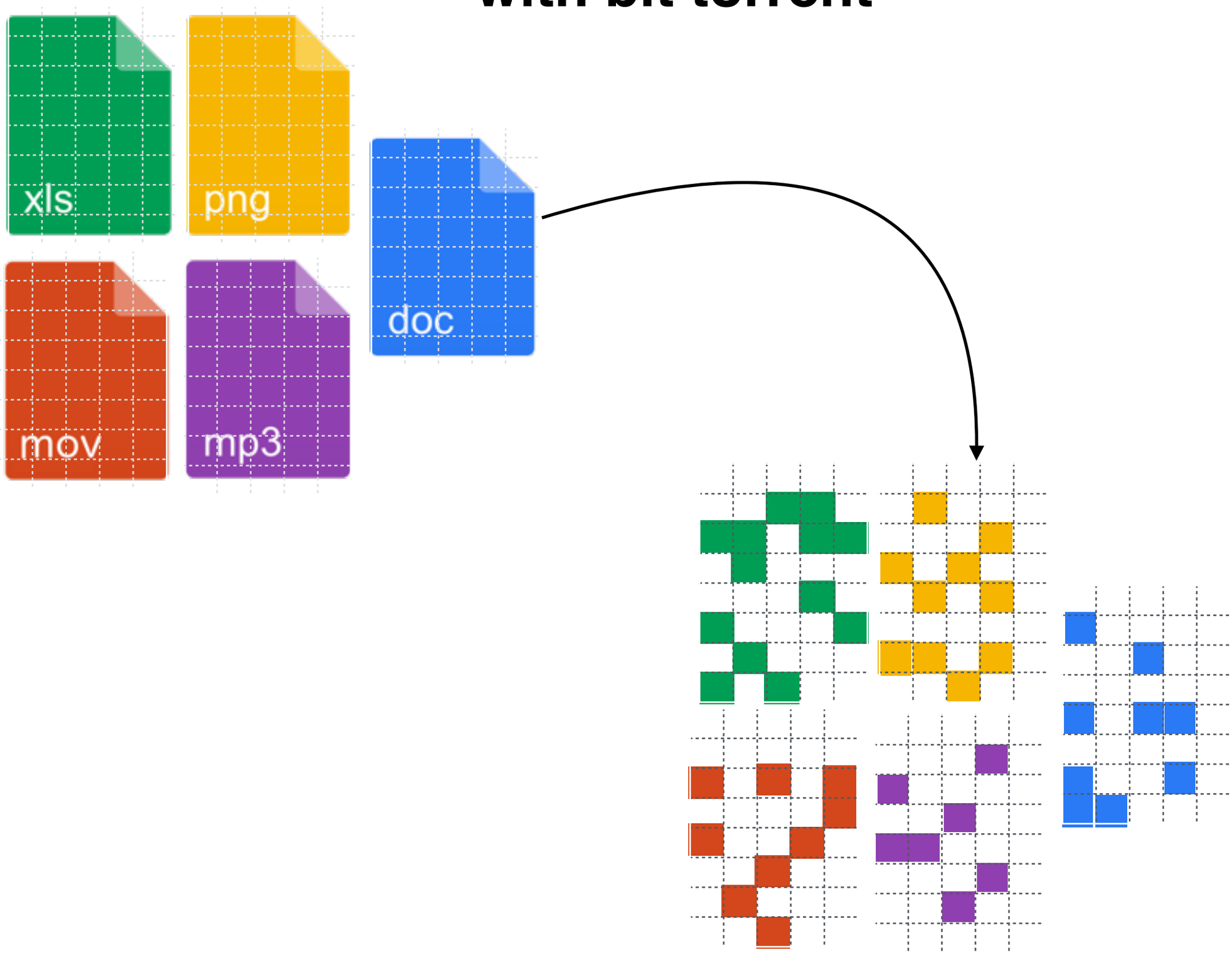
Replicating Files with bit torrent



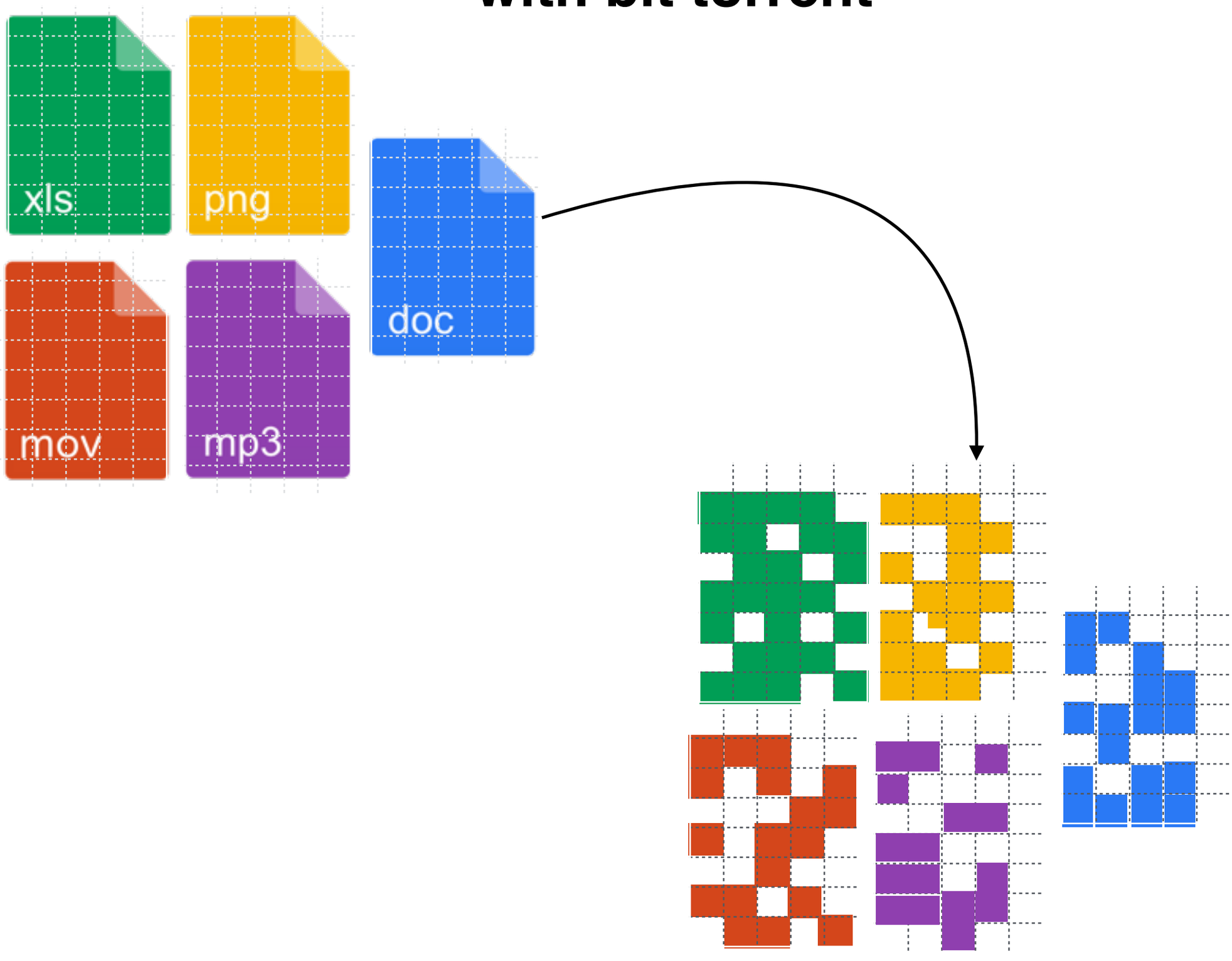
Replicating Files with bit torrent



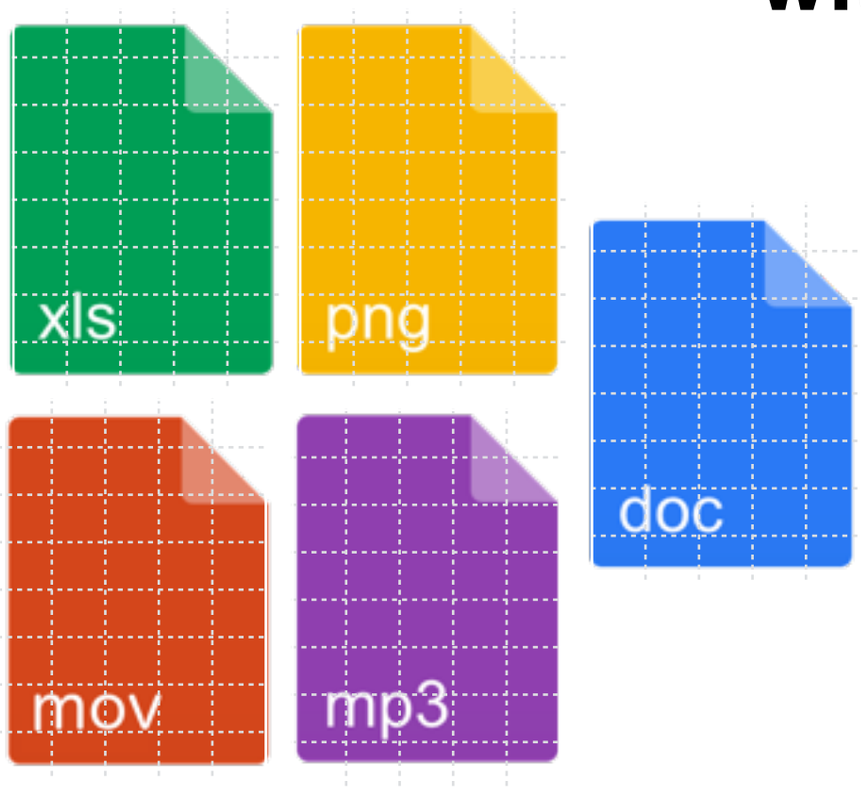
Replicating Files with bit torrent

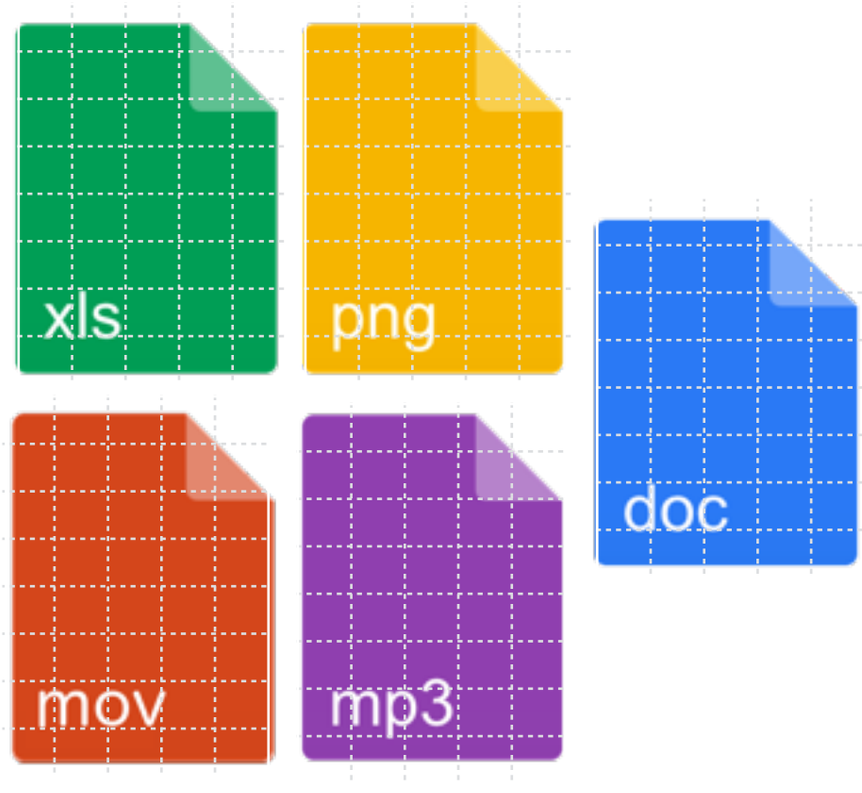


Replicating Files with bit torrent



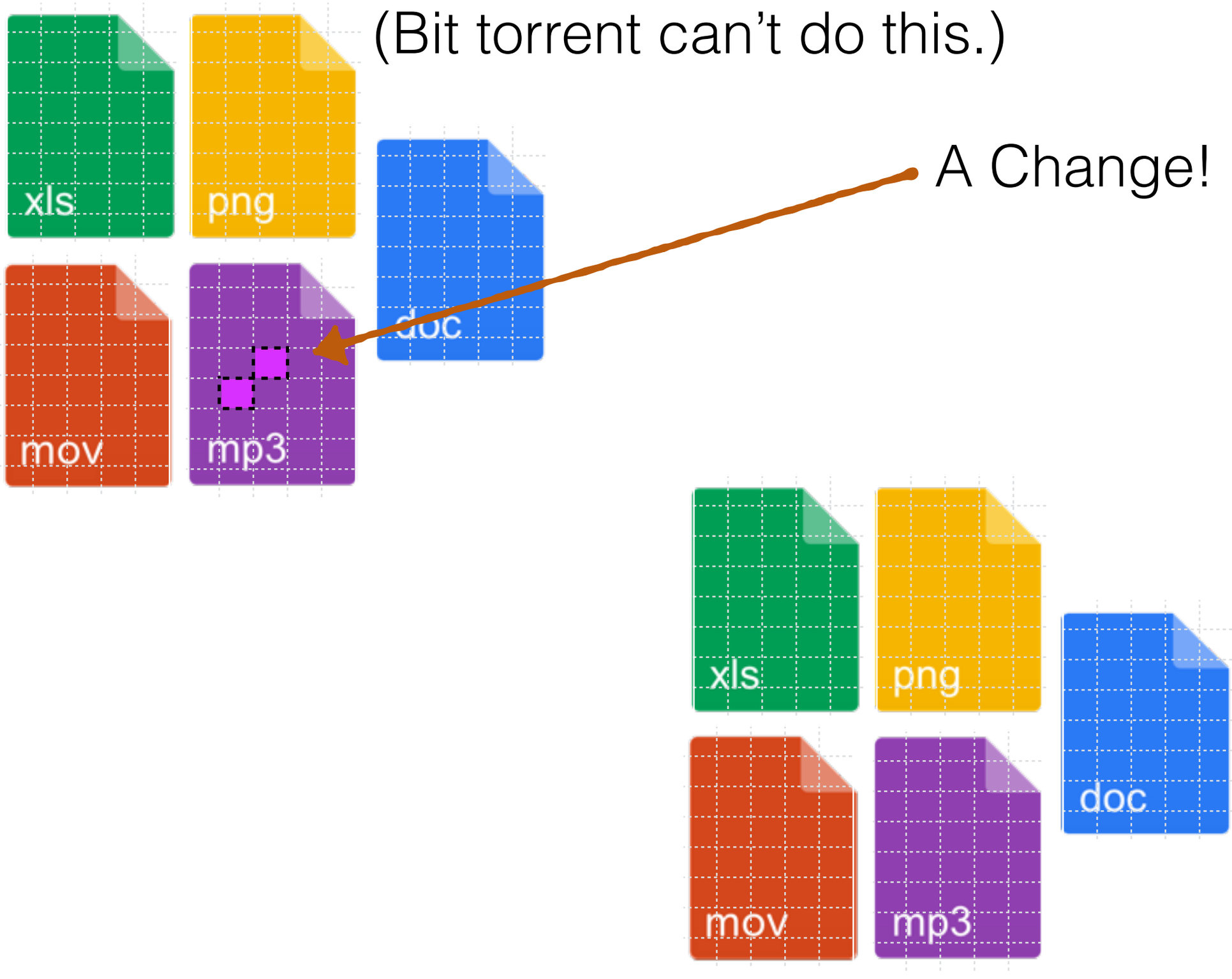
Replicating Files with bit torrent





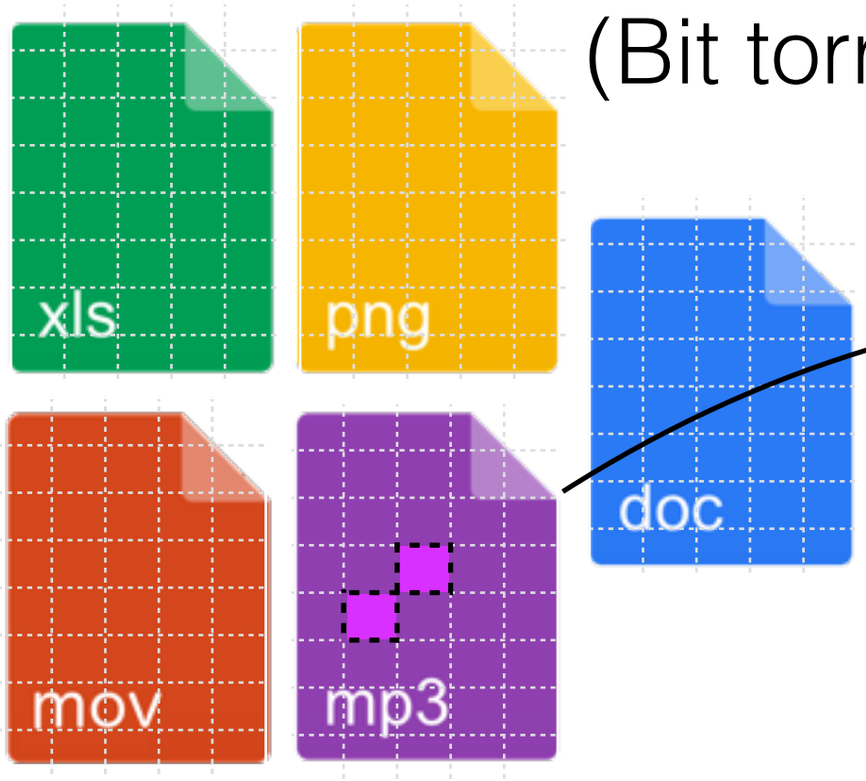
Replicating Changes

(Bit torrent can't do this.)

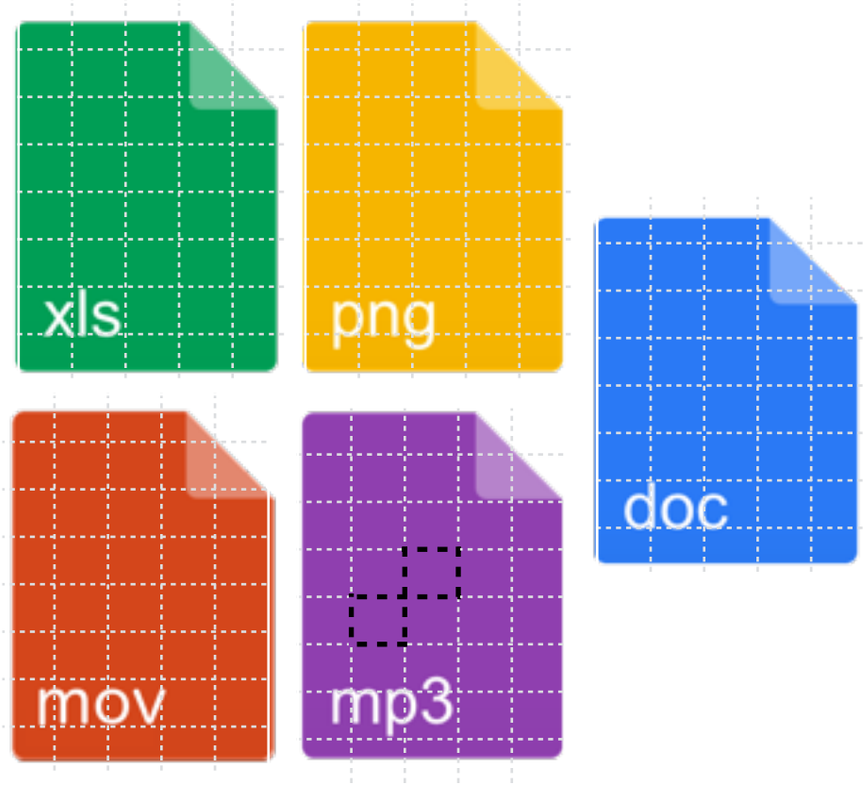


Replicating Changes

(Bit torrent can't do this.)

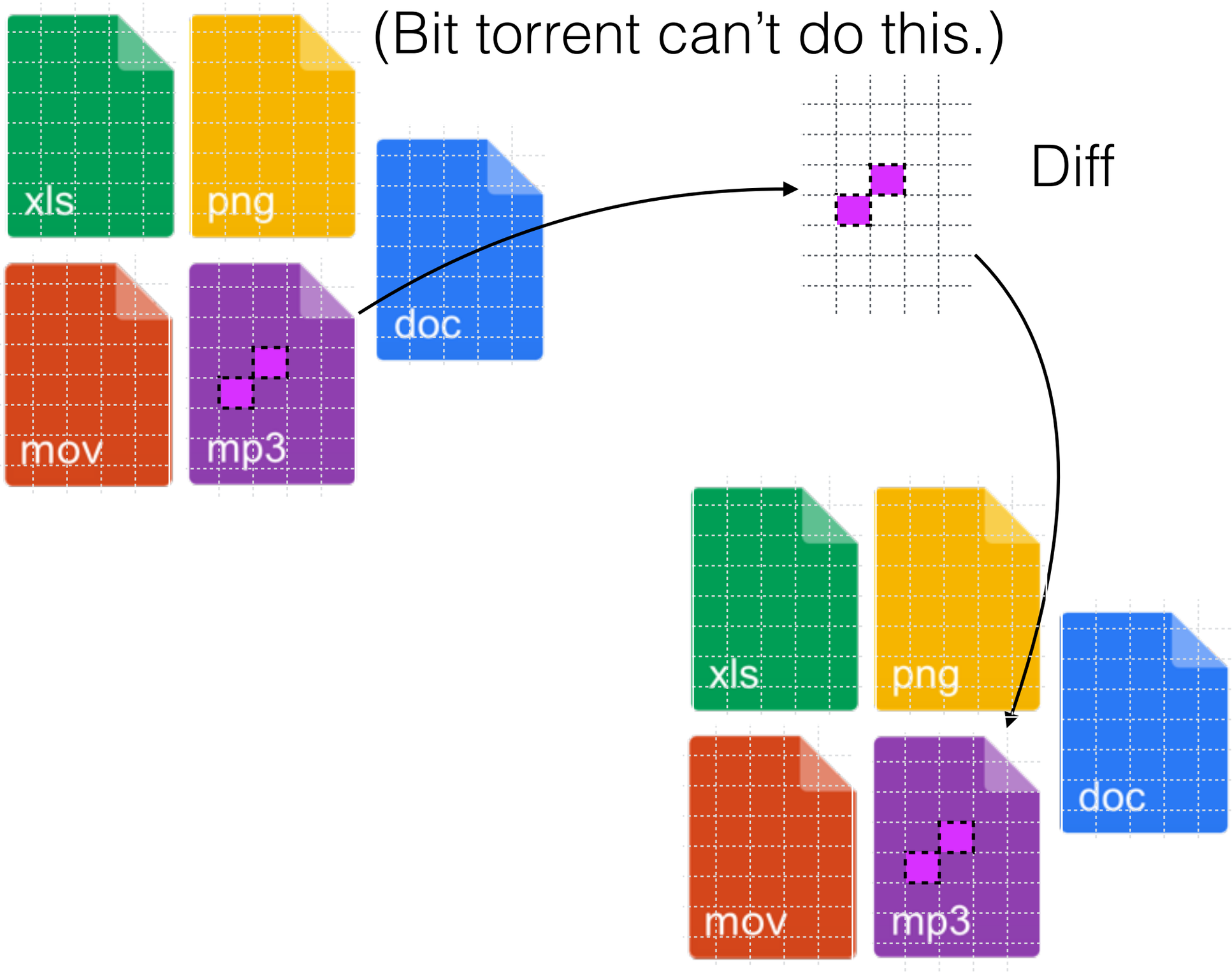


Diff



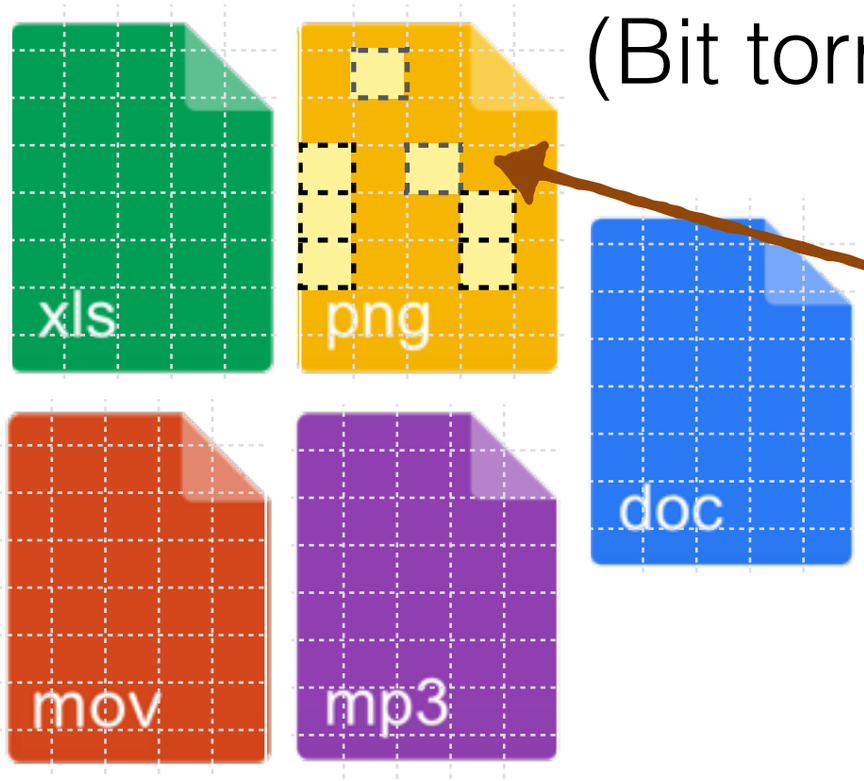
Replicating Changes

(Bit torrent can't do this.)

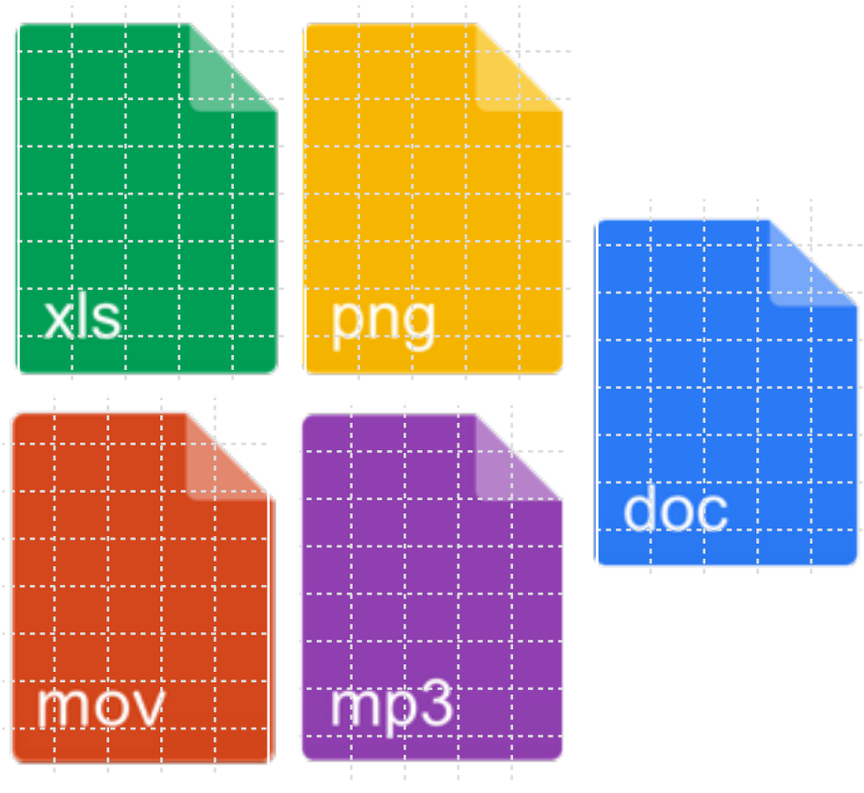


Replicating Changes

(Bit torrent can't do this.)

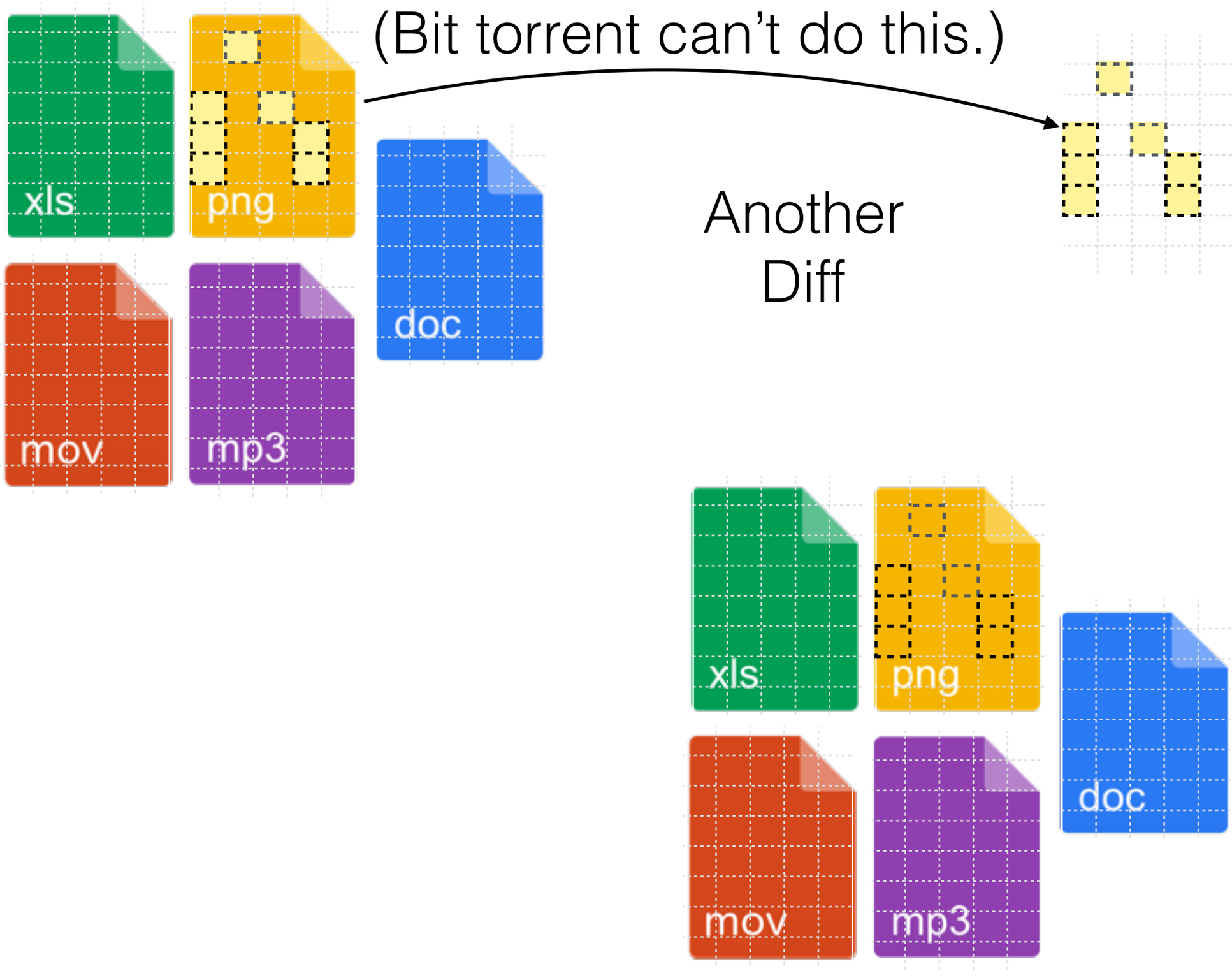


Another Change!



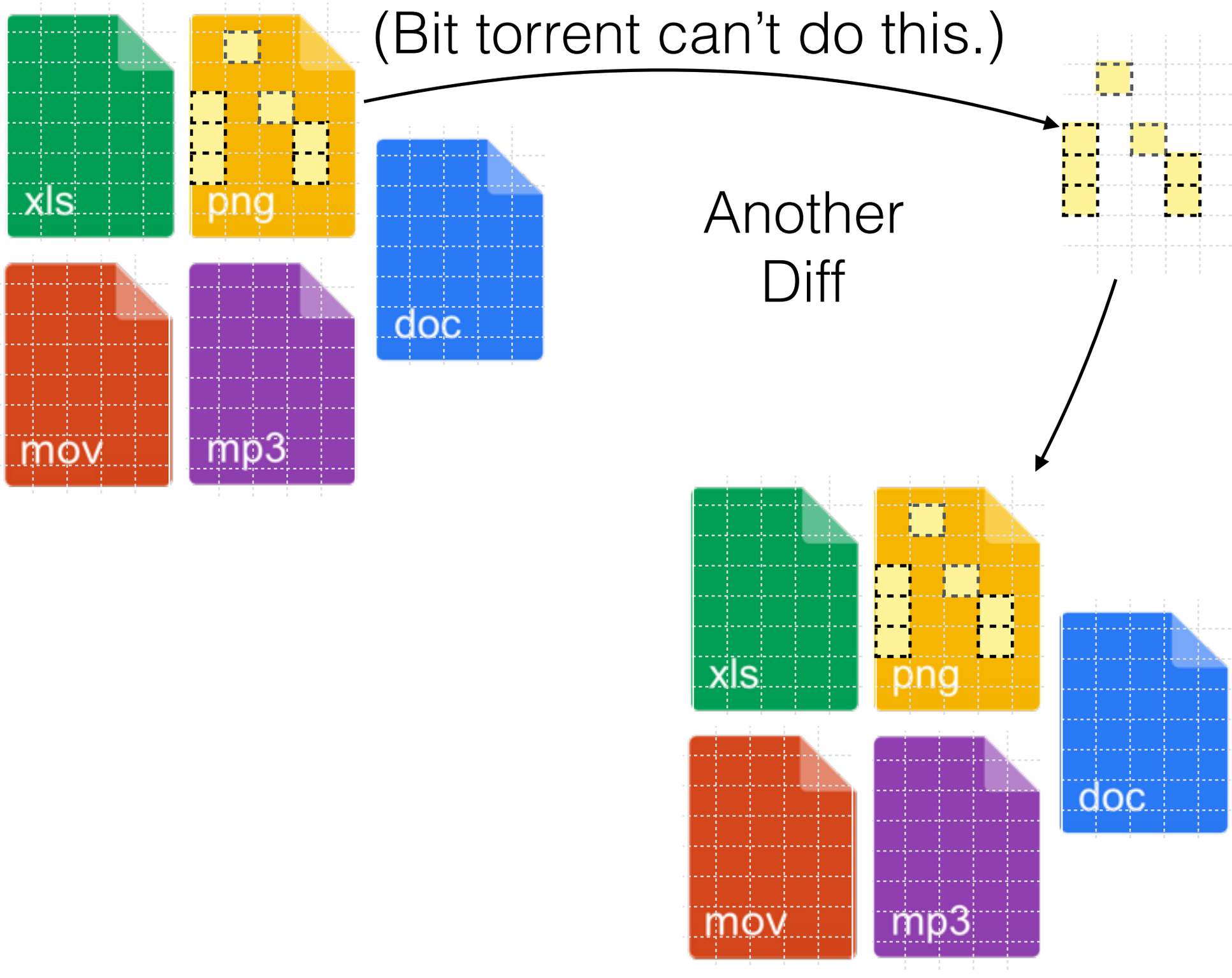
Replicating Changes

(Bit torrent can't do this.)



Replicating Changes

(Bit torrent can't do this.)



Representing Files and their Changes

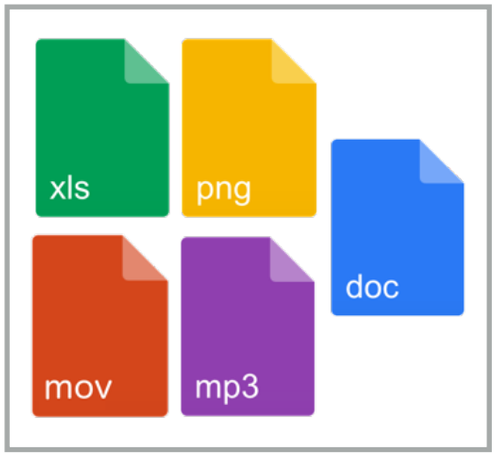
content

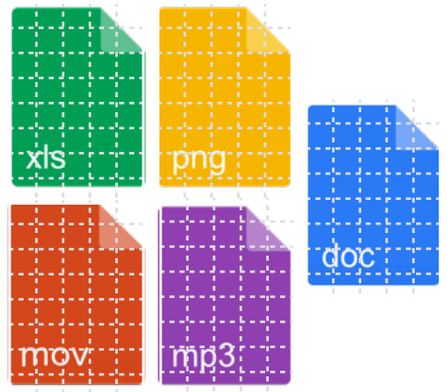
diff

which memory blocks changed?

Initial Commit

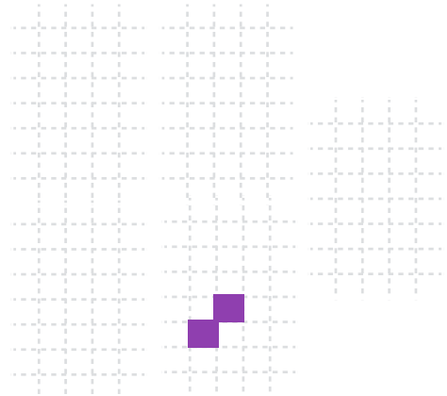
hash: 780f619bd95…





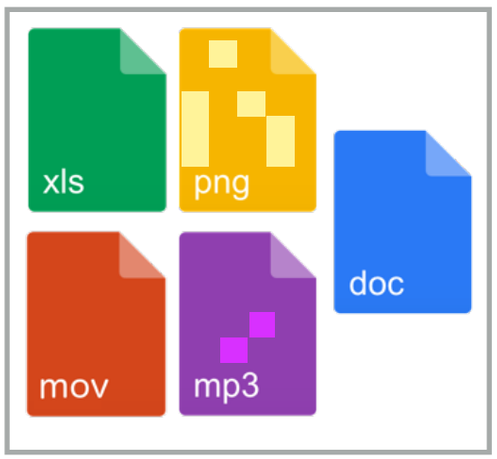
Commit 2

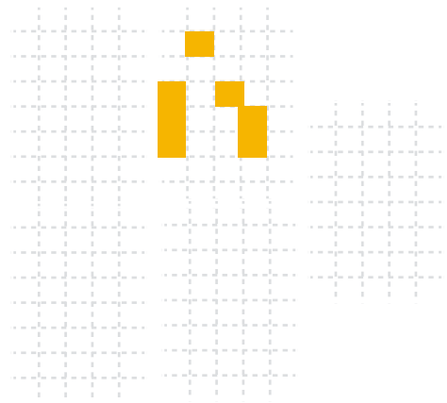
hash: b2da350dcb…



Commit 3

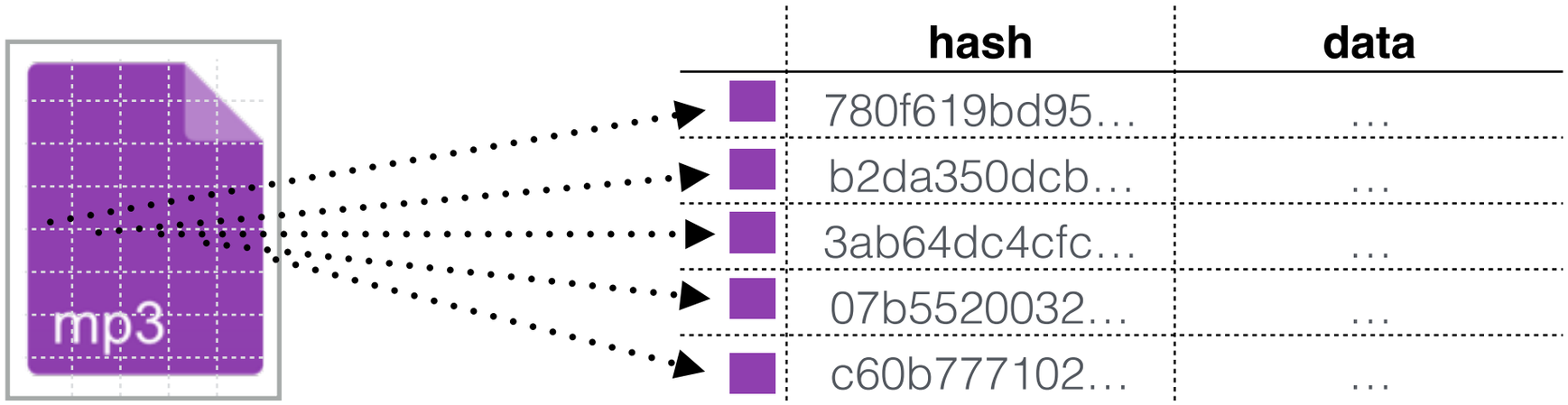
hash: 3ab64dc4cfc…





Representing Files as Merkle DAGs

Bit torrent represents the content as a Hash Table

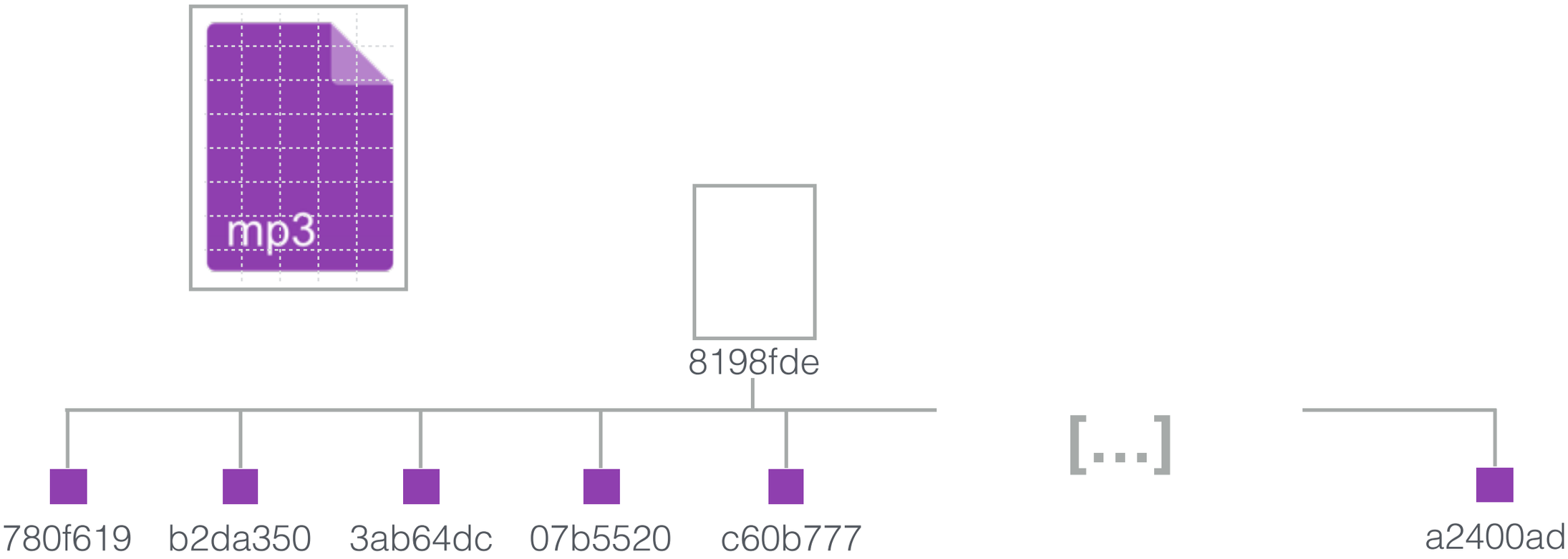


“Unfortunately by dividing a file into fixed size blocks results in all blocks being changed if we to insert a single new byte into the beginning of a file. “

- Hyperdrive Specification

Representing Files as Merkle DAGs

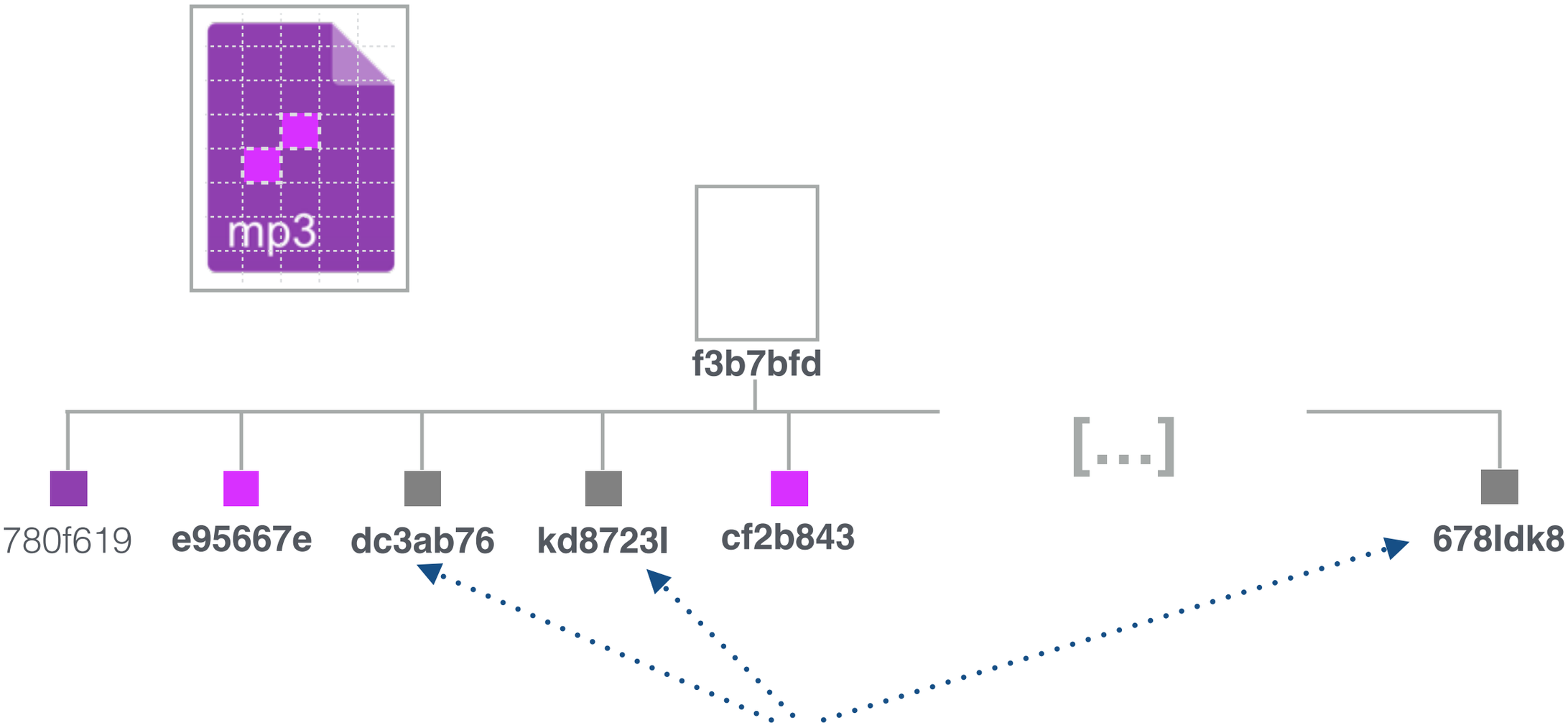
hyperdrive represents the content as a *Hash Tree*



Note: this is a slight simplification of the DAG that hyperdrive builds

Representing Changes to Files

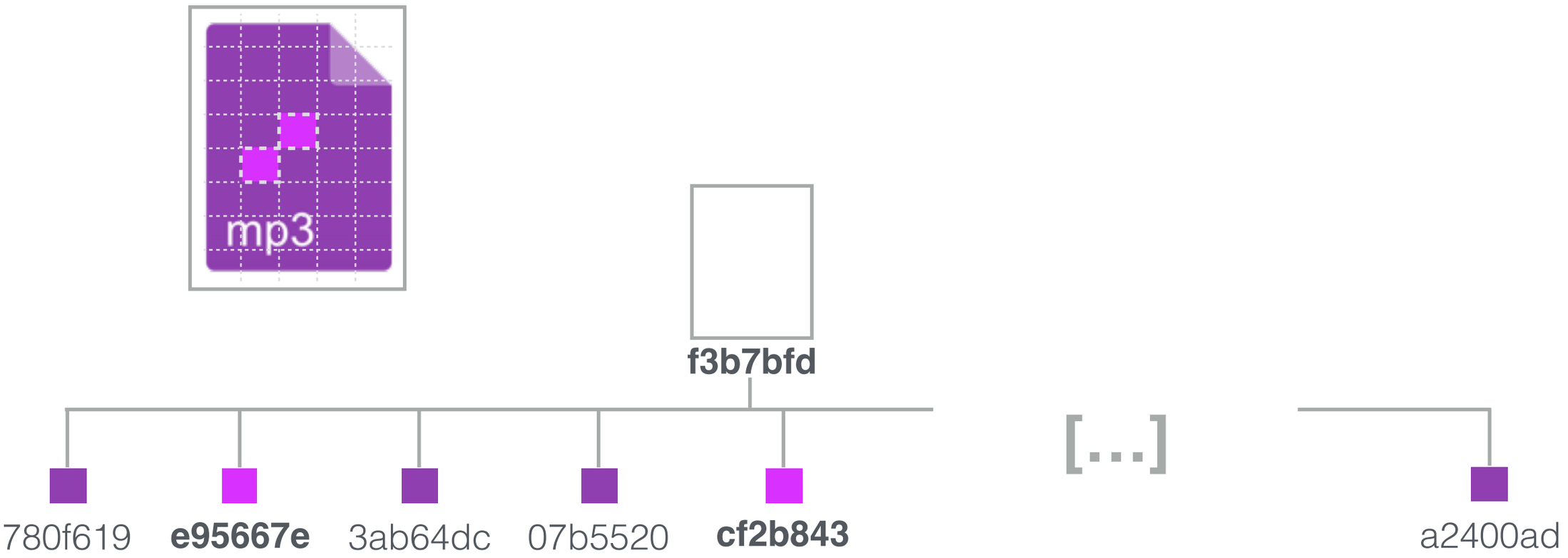
Changing one block changes everything



"Unfortunately by dividing a file into fixed size blocks **results in all blocks being changed** if we to insert a single new byte into the beginning of a file." - Hyperdrive Specification

Representing Changes to Files

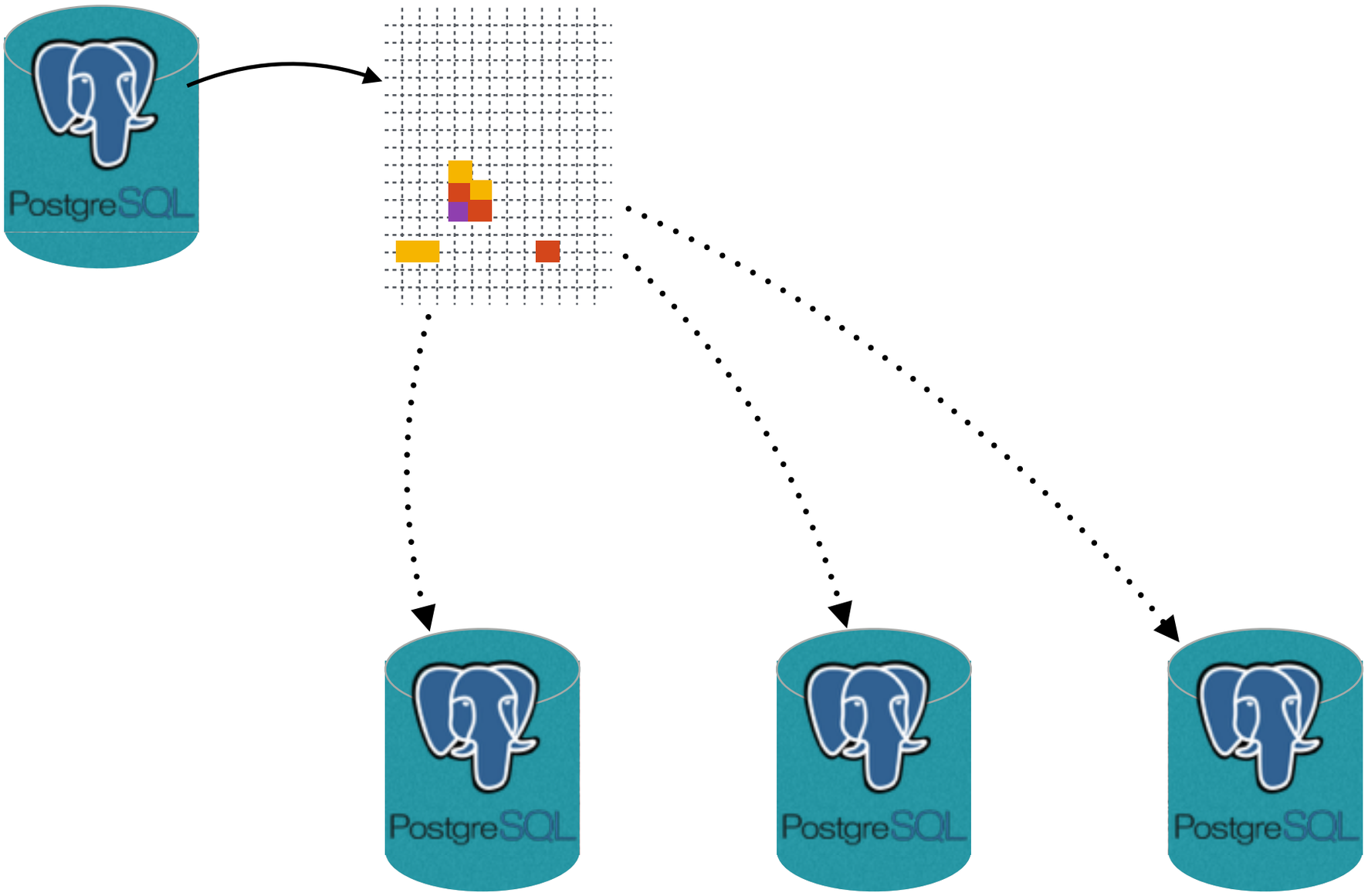
To fix this, hyperdrive uses *Rabin fingerprinting*



"*Rabin fingerprinting* is a content-defined chunking strategy that splits a file into varied size blocks based on the file content. […] Content defined chunking means that **only the neighboring blocks are likely to change** when changing a file." - Hyperdrive Specification

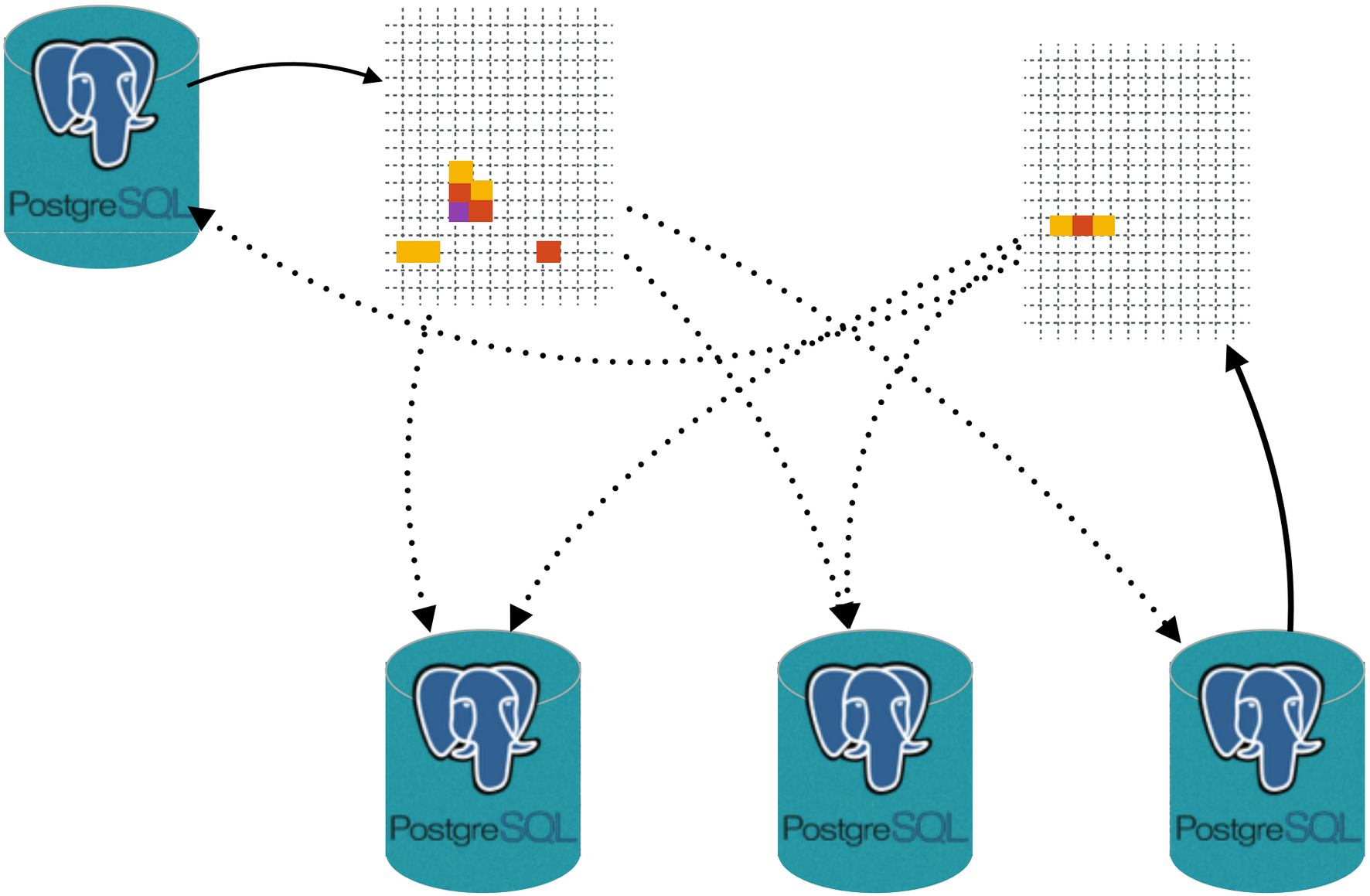
Replicating Data

Parallel: Database Replication



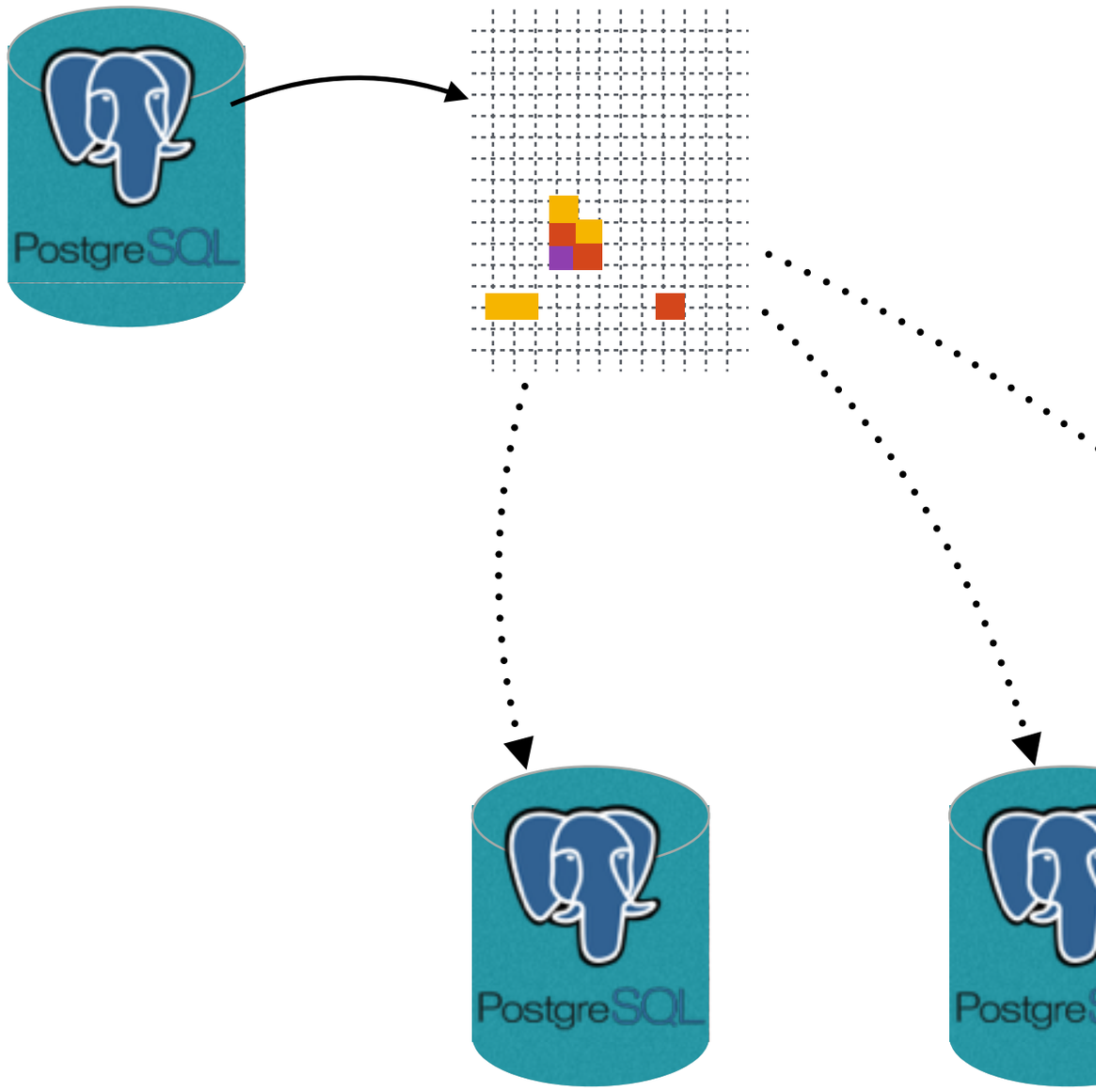
Replicating Data

Database Replication gets Complicated



Replicating Data

Database Replication gets Complicated



Major Factors:
Incomplete Transactions
Merge Conflicts
CAP theorem

Hypothetical: Raw RDBMS Tables in hyperdrive

Initial Commit

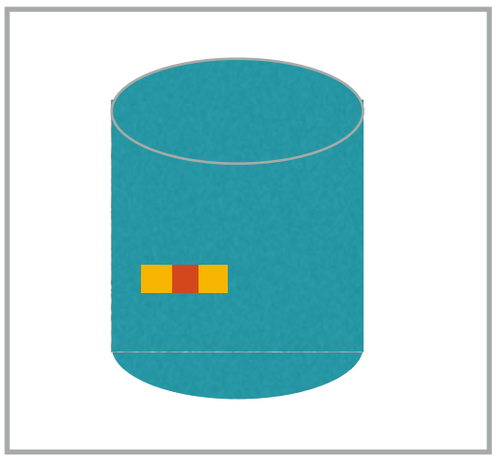
Commit 2

Commit 3

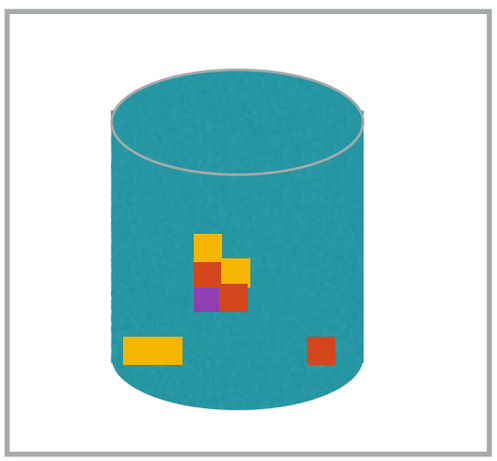
content

hash: 780f619bd95…

hash: b2da350dcb…

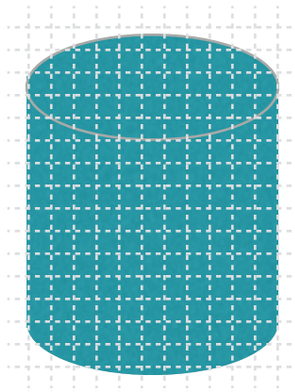


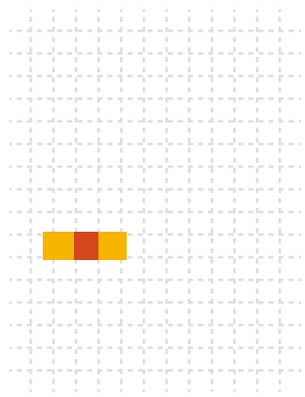
hash: 3ab64dc4cfc…

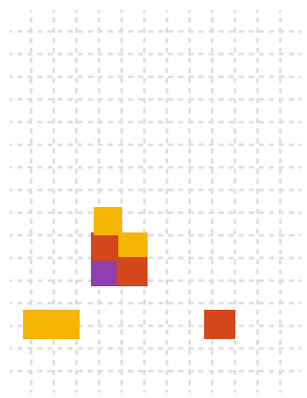


diff

which memory blocks changed?

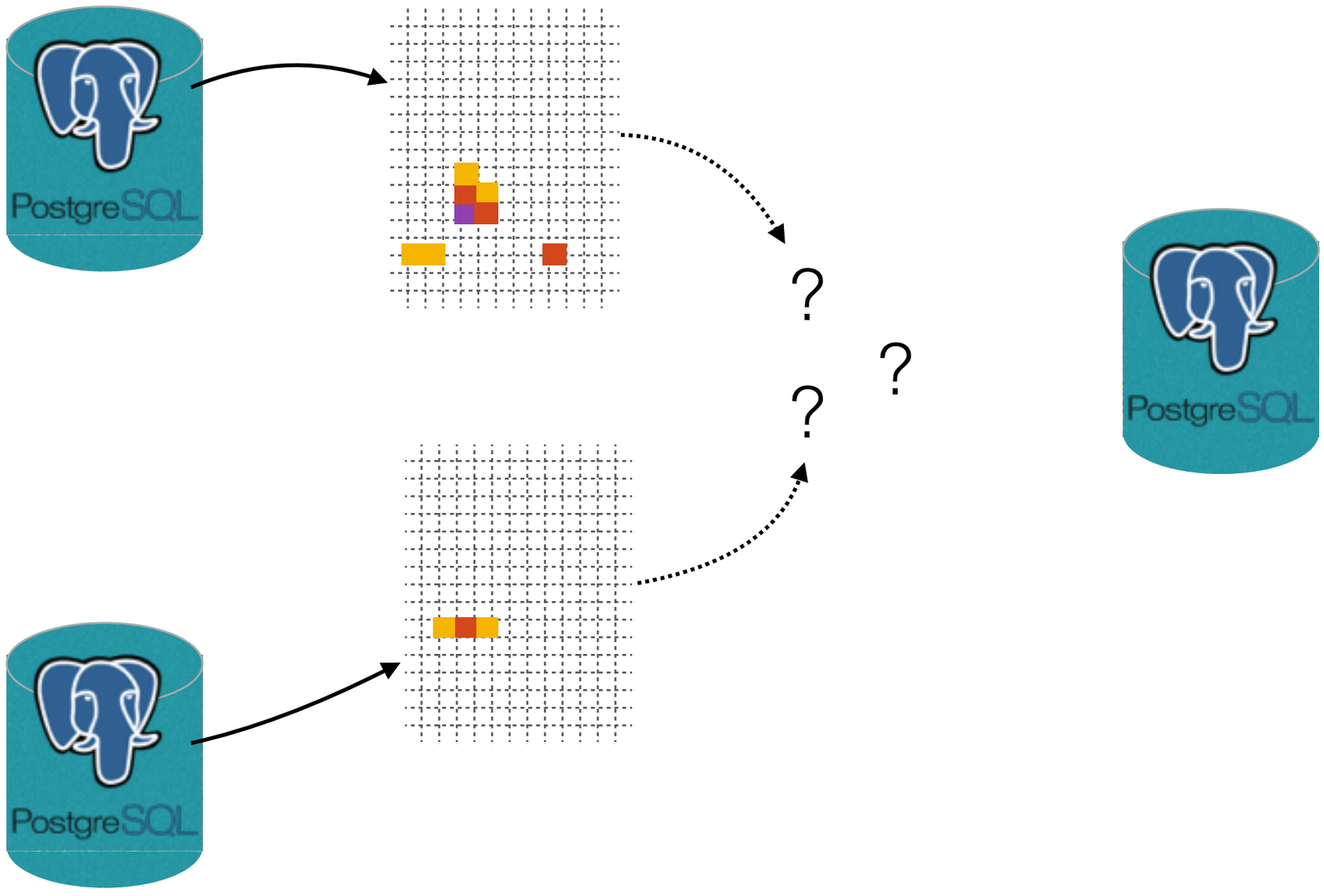






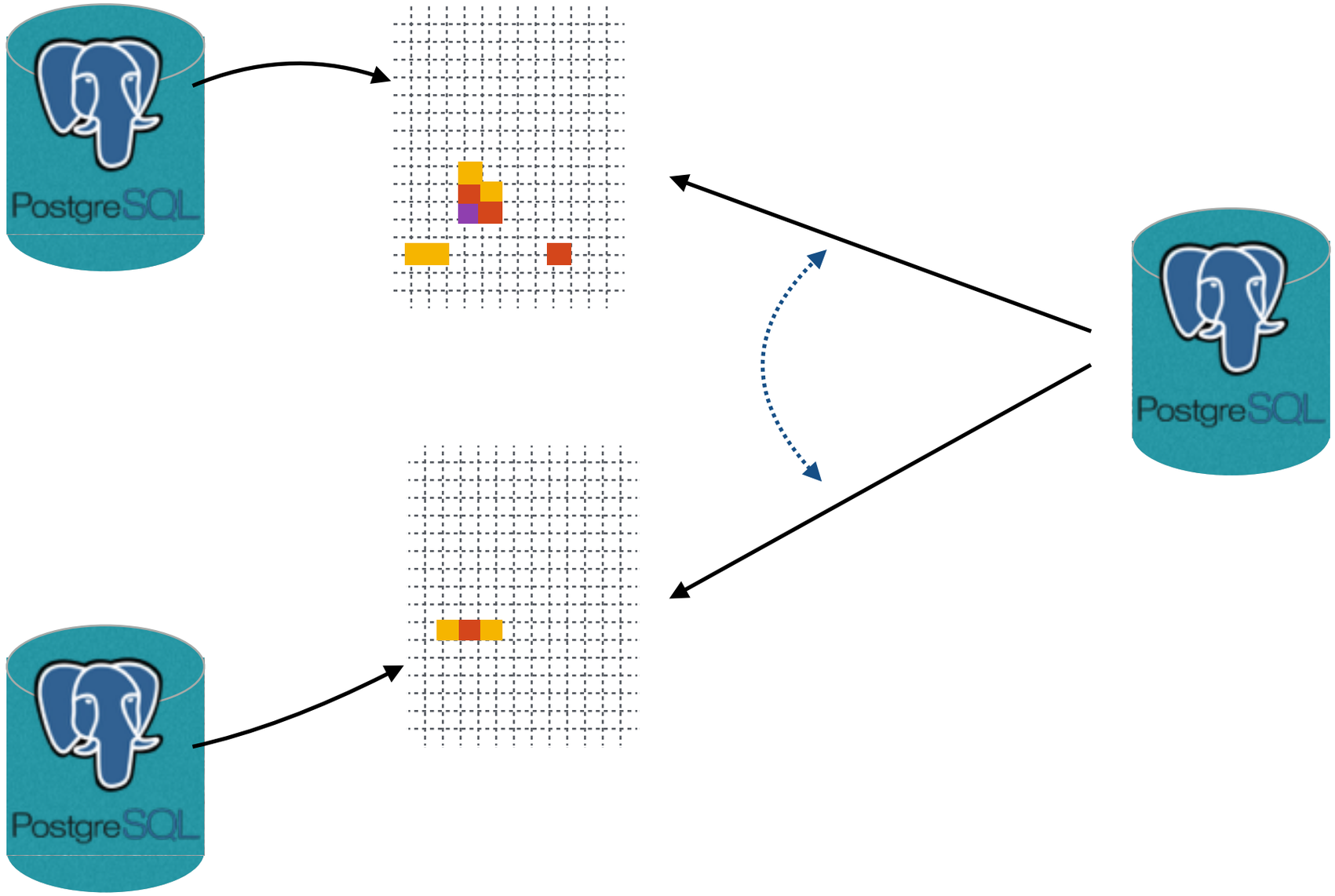
Hypothetical: Raw RDBMS Tables in hyperdrive

How do you diff 2 change sets?



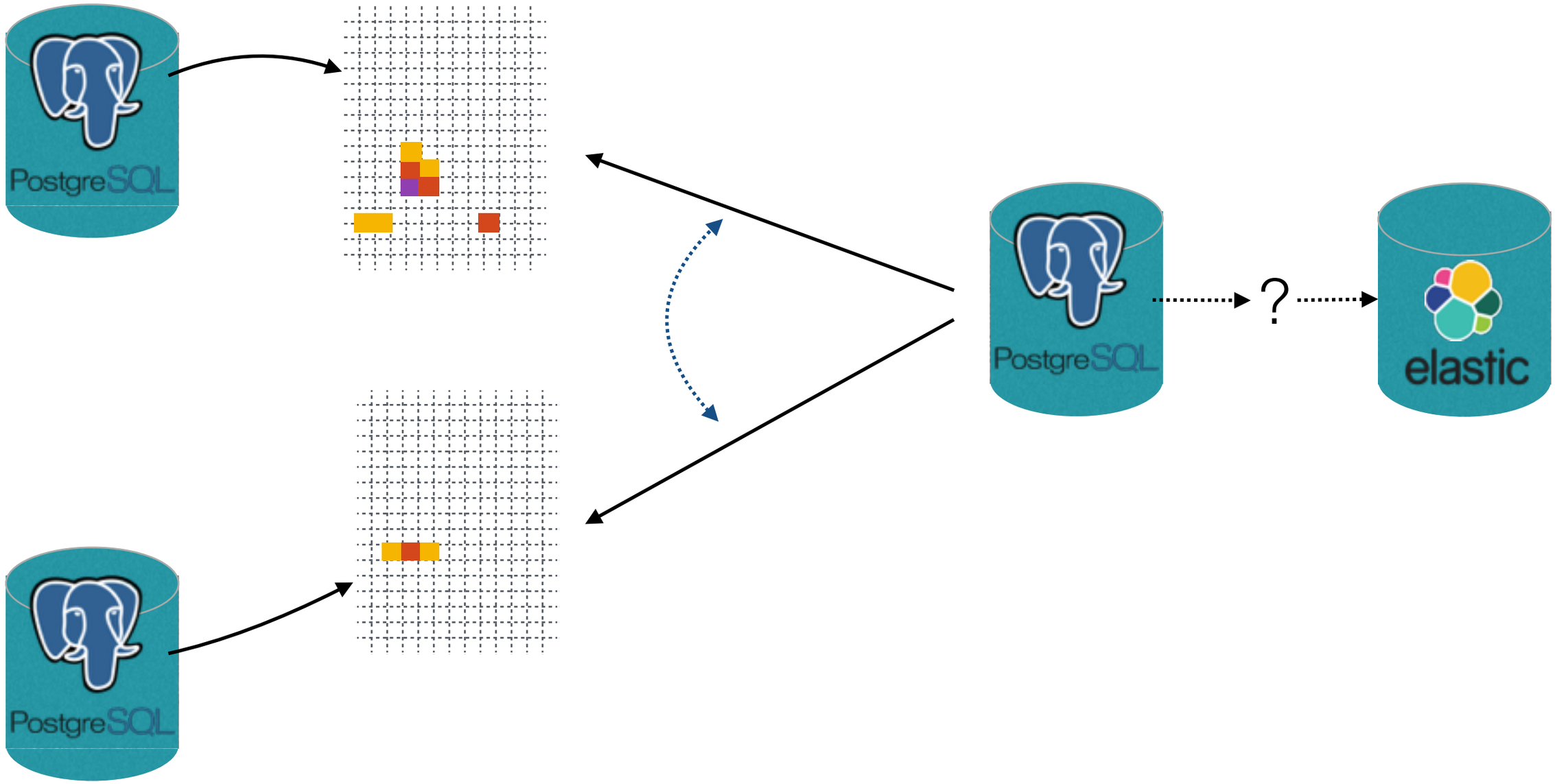
Hypothetical: Raw RDBMS Tables in hyperdrive

How do you switch between change sets?



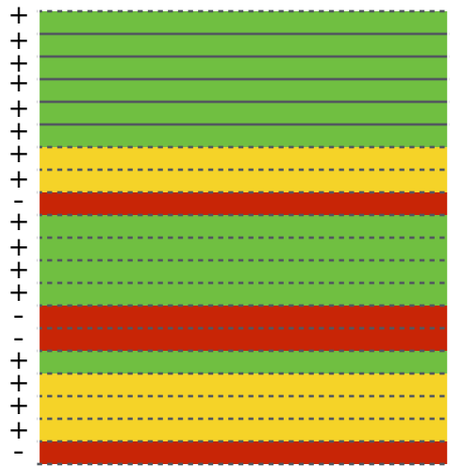
Hypothetical: Raw RDBMS Tables in hyperdrive

What if you have a secondary index?



Keeping a Journal/Log

Track Row Additions, Deletions and Updates



Appending to a Log

content

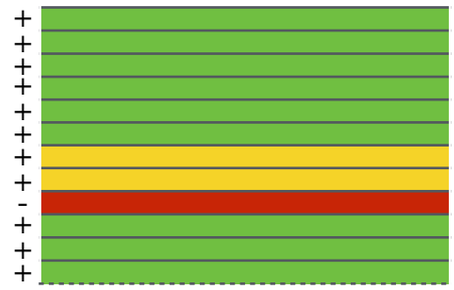
add 7 entries

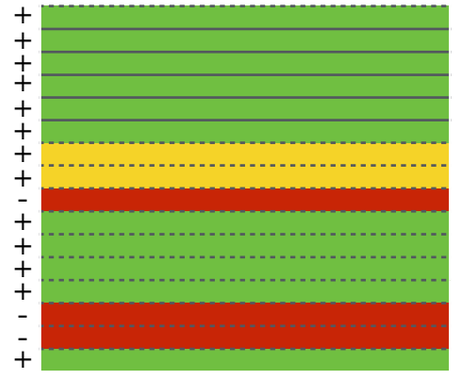
update 2 entries
delete 1 entry
add 3 entries

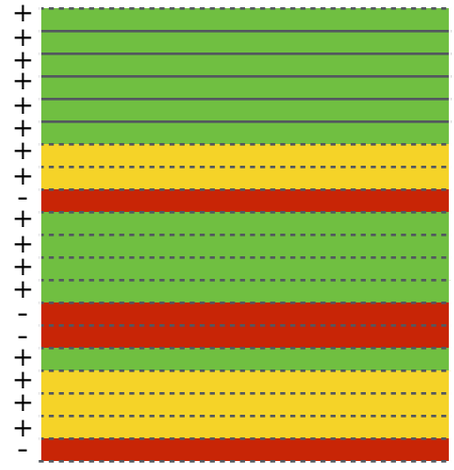
add 1 entry
delete 2 entries
add 1 entry

update 3 entries
delete 1 entry

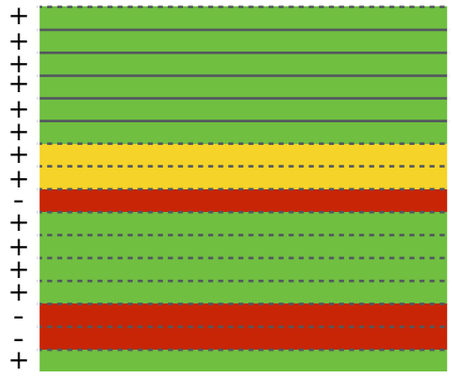
log







Log-Structured Databases: SSTables



A "Sorted String Table" then is exactly what it sounds like, it is a file which contains a set of arbitrary, sorted key-value pairs inside.

Once the SSTable is on disk, it is **immutable**, hence *updates and deletes can't touch the data*. Instead, a more recent value is simply stored […] in case of update, and a "tombstone" record is appended for deletes.

- lilya Gregorik

Log-Structured Databases: leveldb

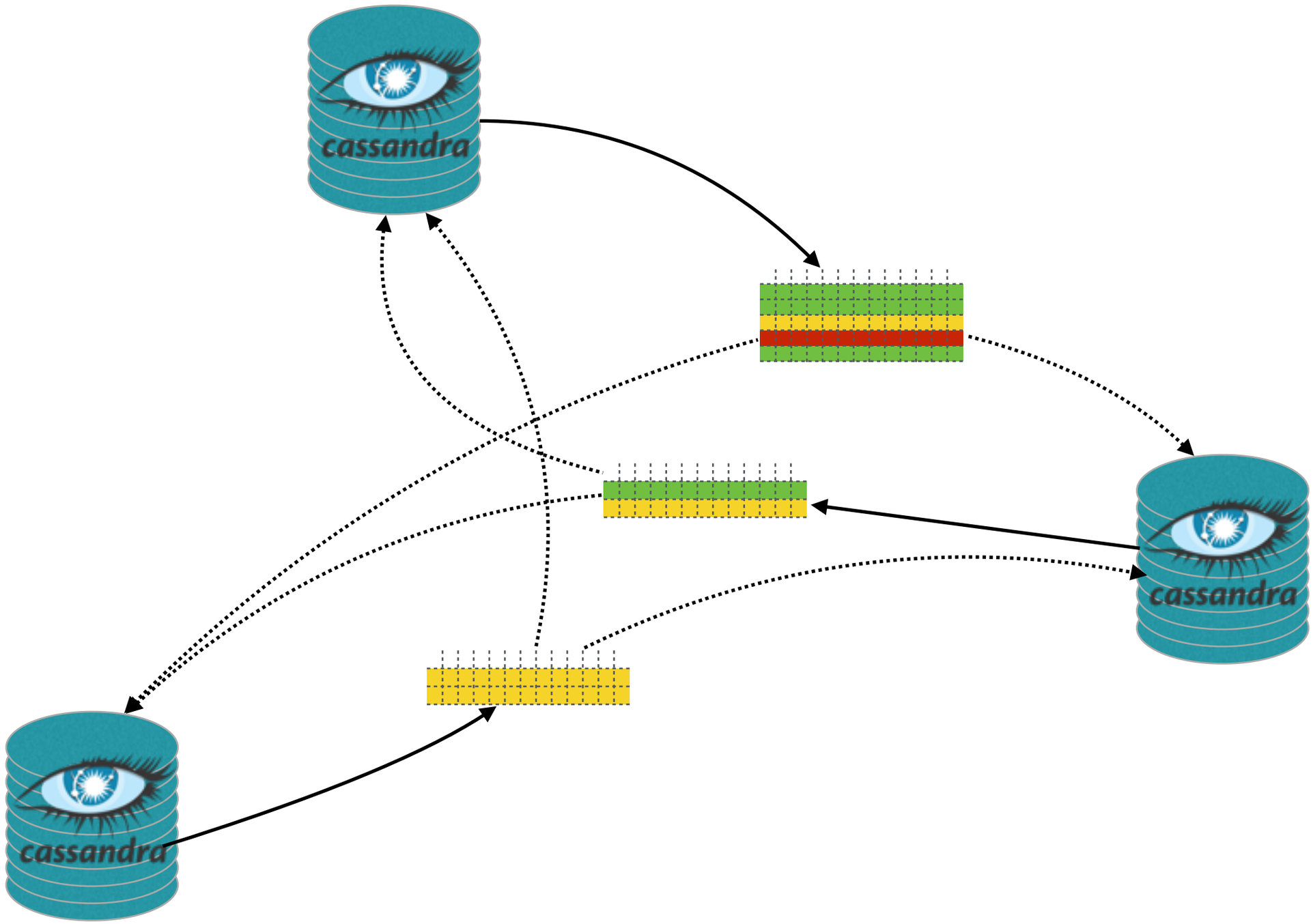
leveldb is a log-structured key-value store

= SSTables + memtable

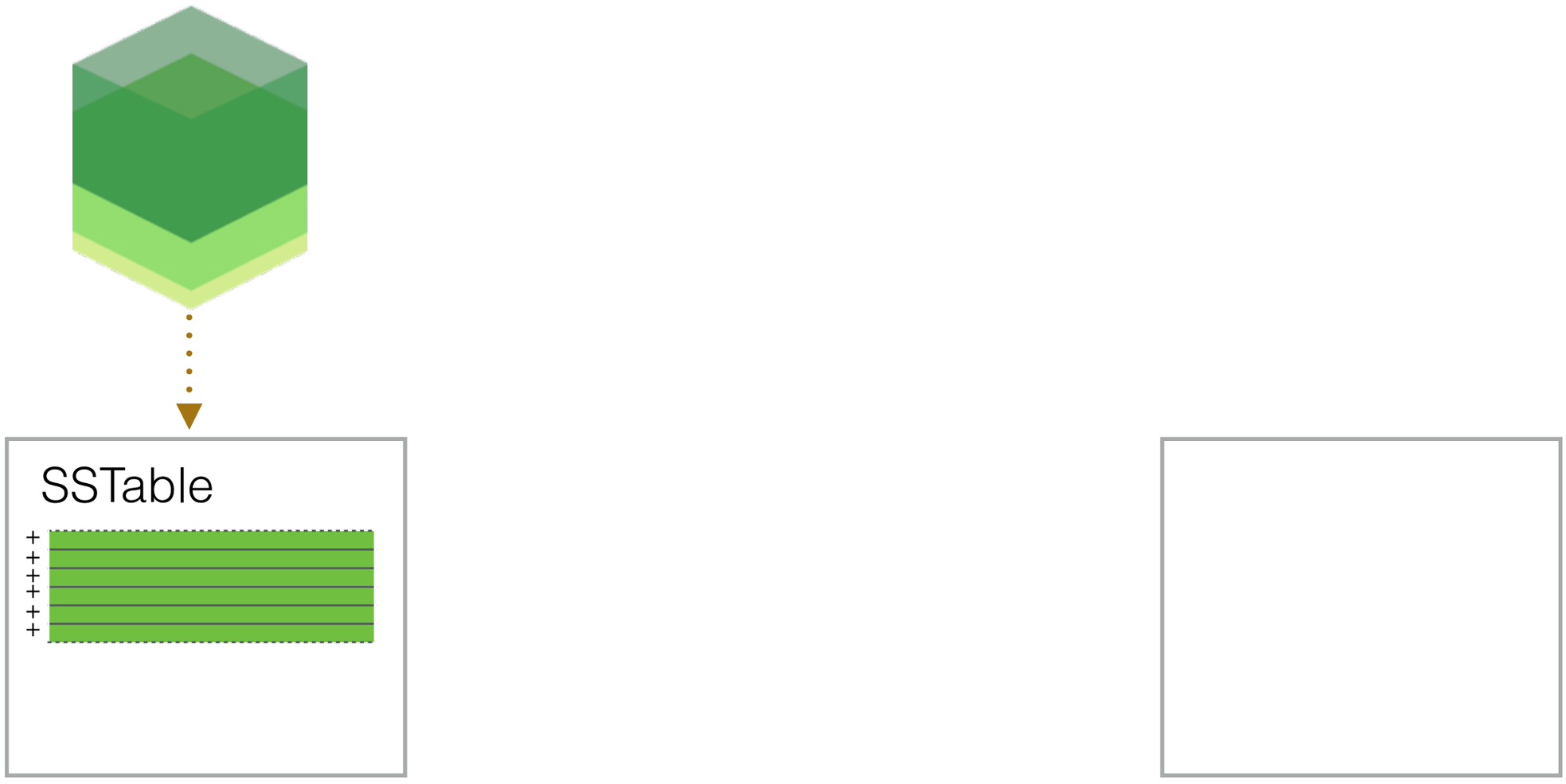
“Take an **SSTable**, add a **MemTable** and apply a set of processing conventions and what you get is a nice database engine for certain type of workloads. In fact, Google's BigTable, Hadoop's HBase, and Cassandra amongst others are all using a variant or a direct copy of this very architecture.” - Ilya Gregorik

Log-Structured Database Replication

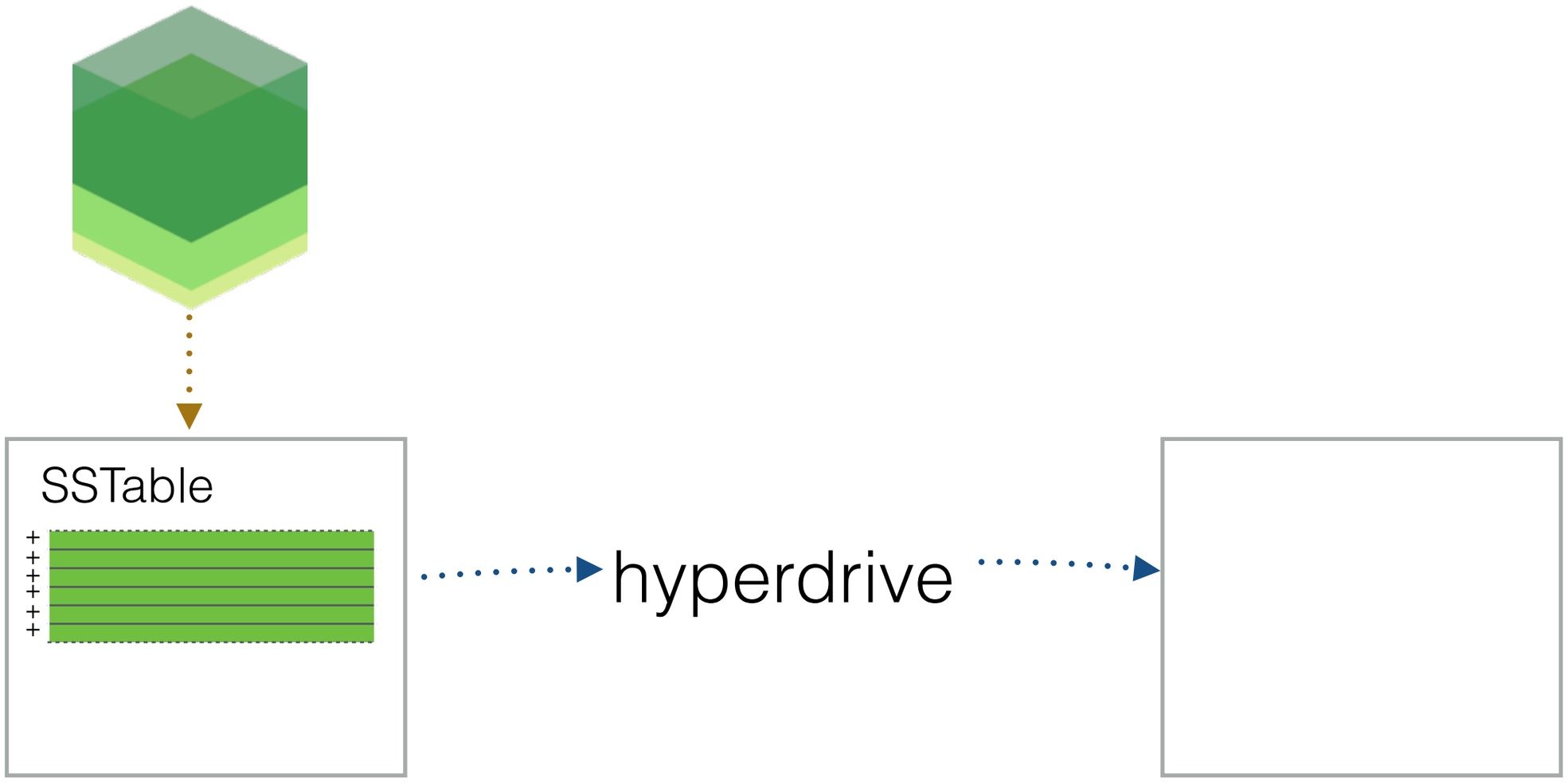
Apache Cassandra uses SSTables to Sync Nodes



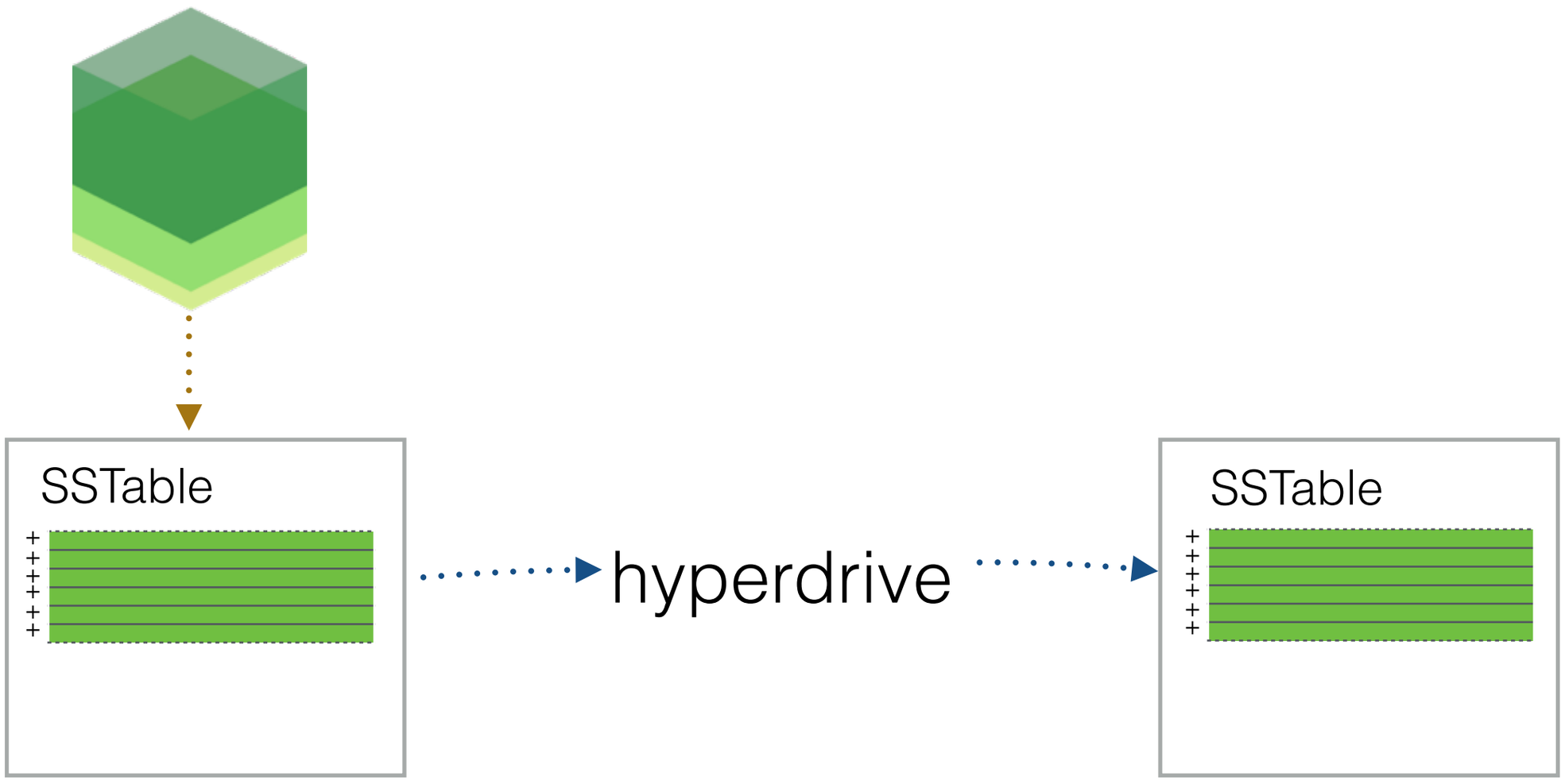
Replicating SStables with hyperdrive



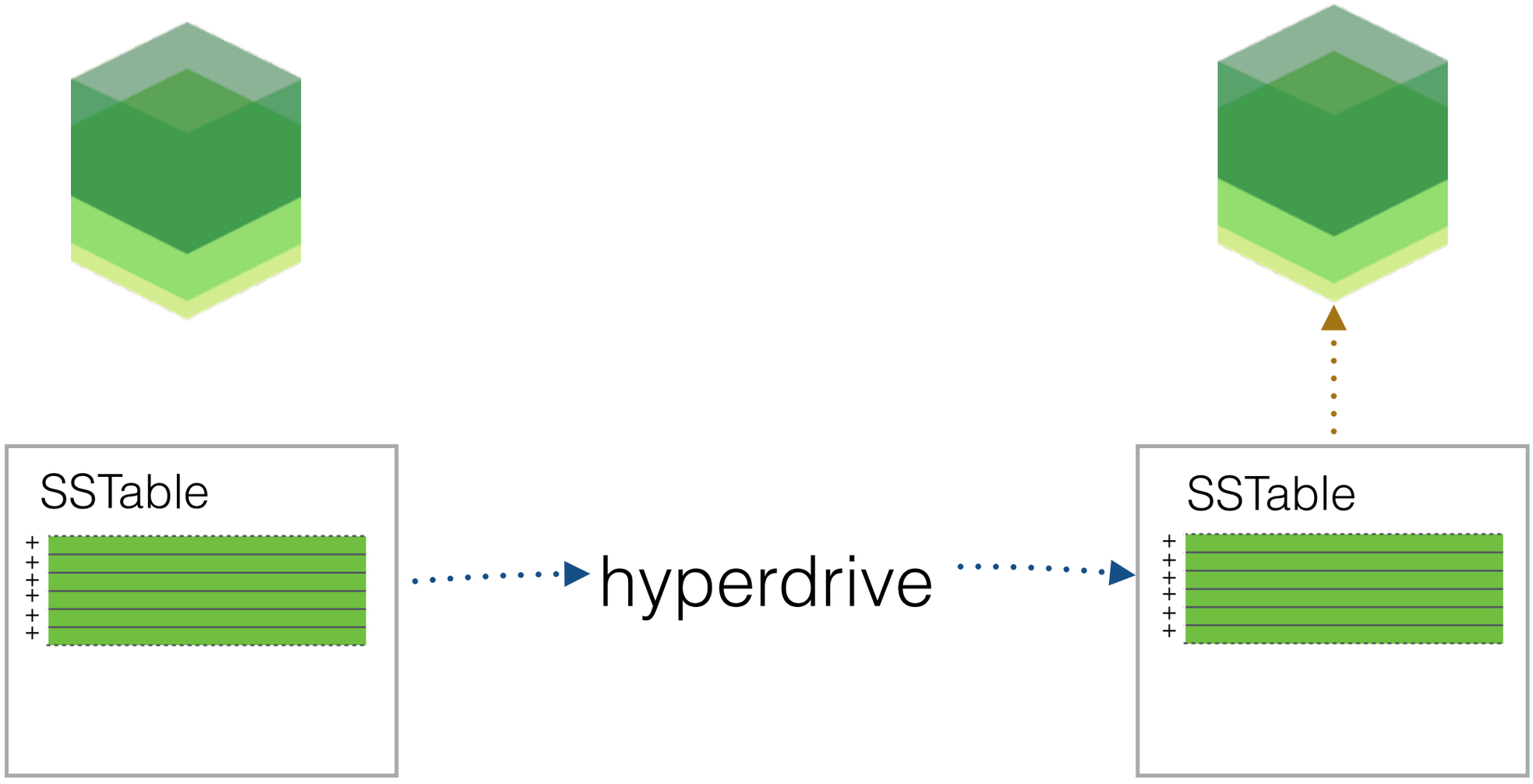
Replicating SStables with hyperdrive



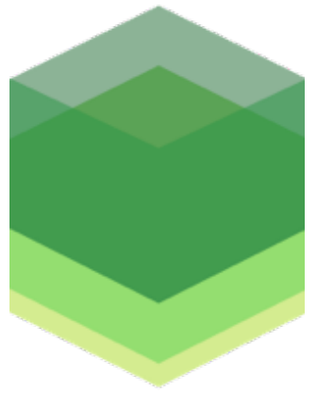
Replicating SStables with hyperdrive

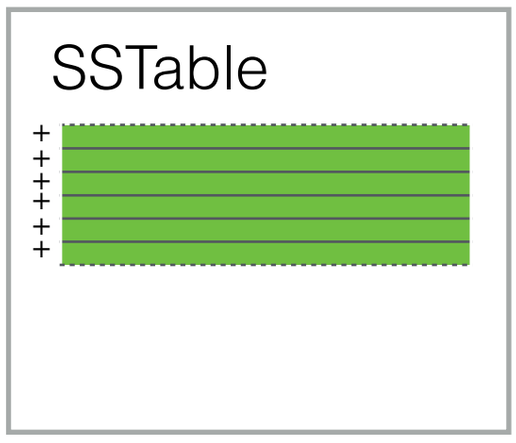


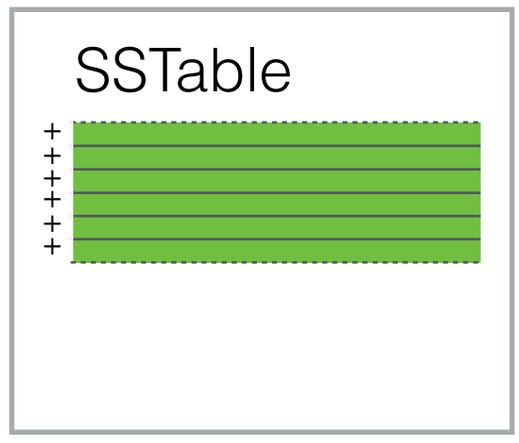
Replicating SStables with hyperdrive



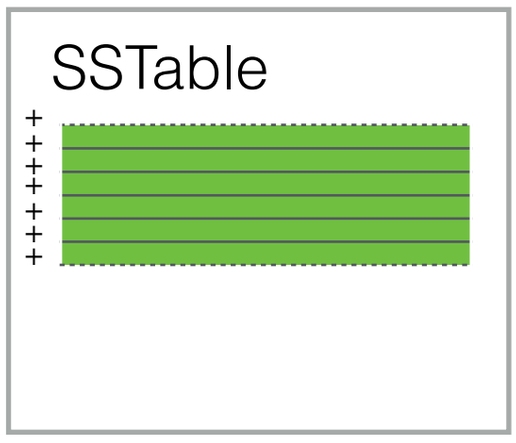
Replicating SStables with hyperdrive

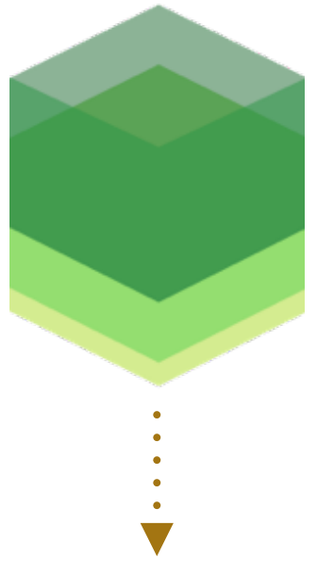


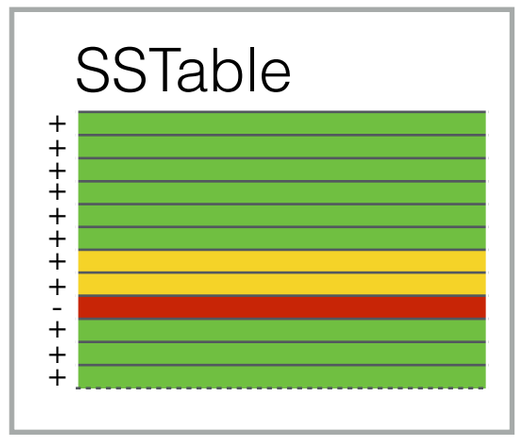




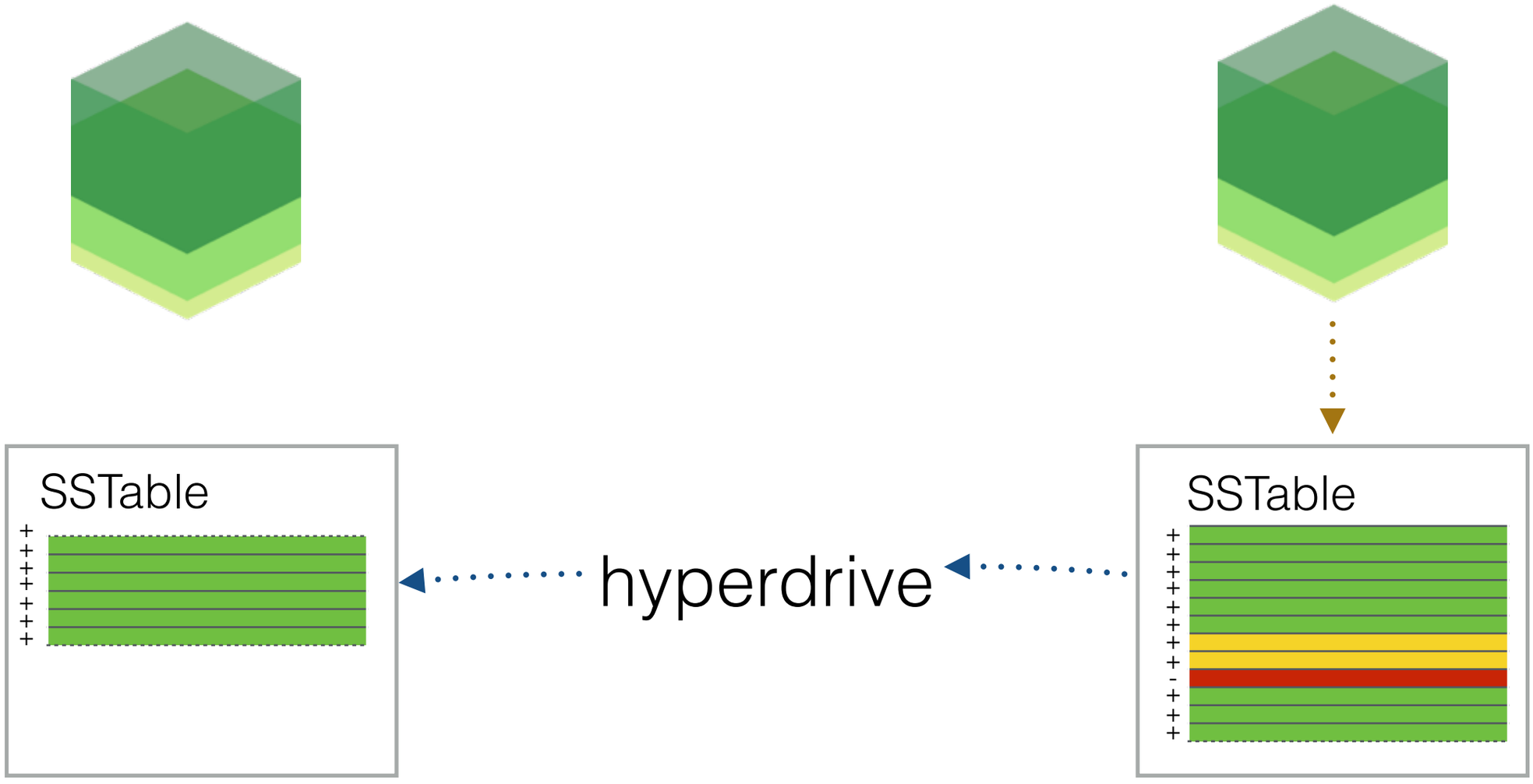
Replicating SStables with hyperdrive



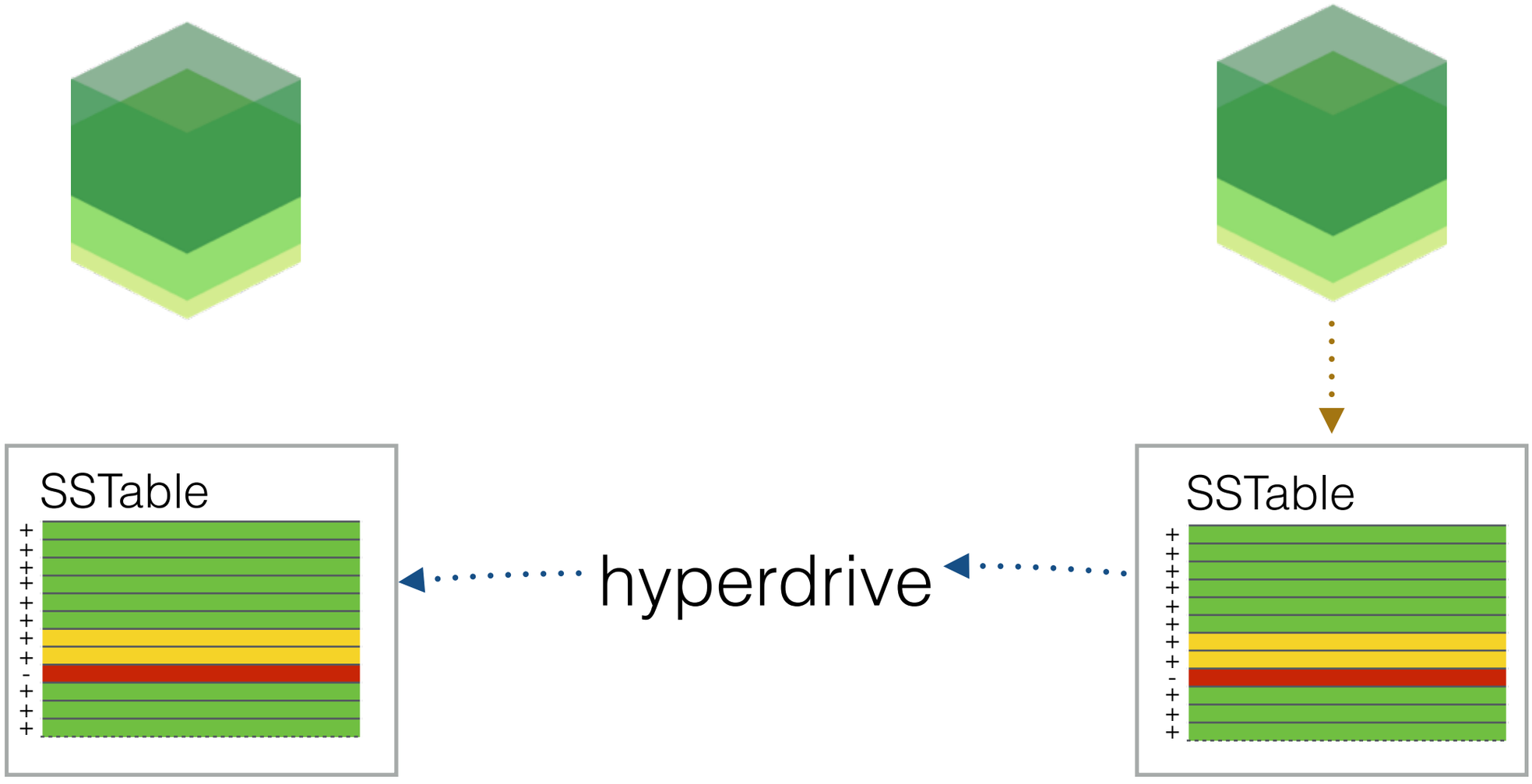




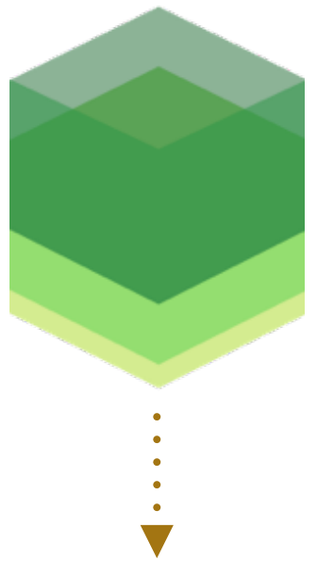
Replicating SStables with hyperdrive

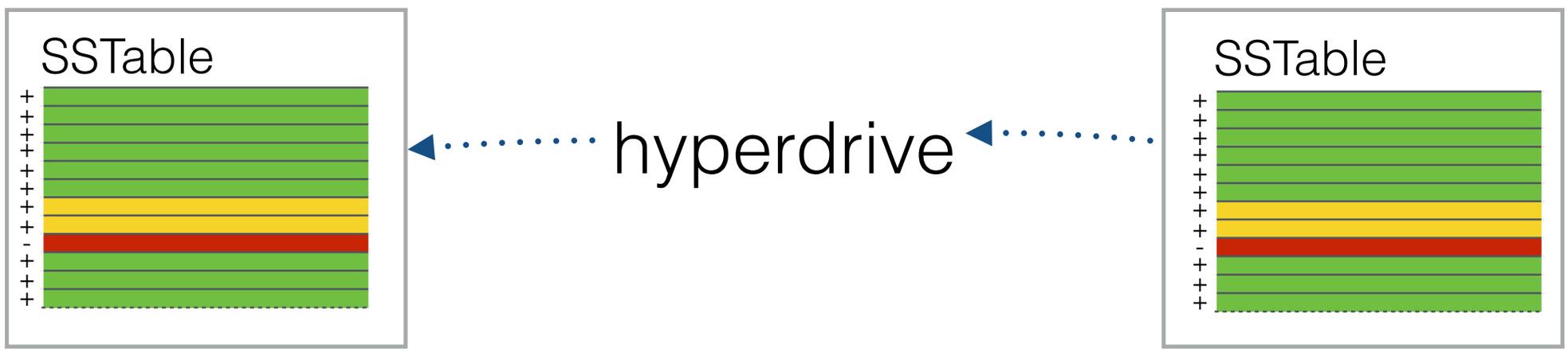


Replicating SStables with hyperdrive



Replicating SStables with hyperdrive





Replicating SSTables with hyperdrive