# A proposed new IGS data flow model

A. Moore, C. Bruyninx, C. Noll, M. Scharber

This plan addresses *actual deficiencies* in the current IGS data flow.

It requires agreement and action from all stations' operational centers, all Data Centers, the Central Bureau, and probably Analysis Centers.

*However,* the logical rules and technical implementation are fairly simple and not burdensome.

Stay with me for a brief tour of the implementation rules, followed by some examples!

# Real operational services have real problems

Goal: An outage of a single DC must not result in data unavailability at other DCs.

- Careful micromanagement of data flow by the DCWG chair has minimized this recently, but there is no coherent scheme.

Goal: When a data file is later revised, all DCs must end up with the proper latest version.

- Revisions should be minimized by design! However, we acknowledge that a zero-tolerance policy is not feasible and design data flow to accommodate.

*We think our data flow design solves these issues.*

# A couple definitions clarified here

Operational Center (OC): makes a station's data available to the IGS community. A station may be its own OC in some cases. An OC pushes data to DCs.

Operational Data Center (ODC): An OC that offers public access to data from a set of stations.

*More definitions are in the position paper.*

*The upstream direction* →

Stations     OCs     RDCs     GDCs

← *The downstream direction*
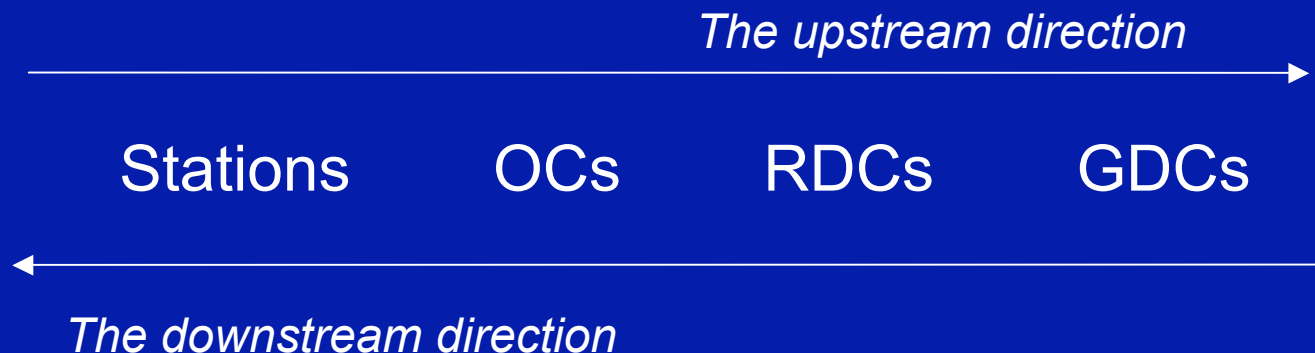
# Preliminaries

- OCs will select 2 DCs of any level: one primary and one secondary.  The most upstream one will be primary. They will be listed in section 13 of the site log.

```
13.     More Information


        Primary Data Center        : GDC1

        Secondary Data Center      : GDC2
```

- RDCs will select 2 GDCs to be their primary and secondary upstream DCs.

# Preliminaries

- The CB will collect and combine this information, and publish a table giving the primary and secondary DC for each station/OC and RDC.

```
OC        Primary        Secondary
          DC             DC
----      -------        ---------
AAAA      GDC1           RDC1
BBBB      RDC1           RDC2
OC1       GDC2           RDC2
OC2       RDC2           ODC2


DC        Primary        Secondary
          DC             DC
----      -------        ---------
RDC1      GDC1           GDC2
RDC2      GDC1           GDC3
ODC1      RDC1           RDC2
ODC2      GDC1           RDC1
```

```xml
<station name="AAAA">

  <primary DC>GDC1</primaryDC>

  <secondaryDC>RDC1</secondaryDC>

</station>

<station name="BBBB">

  <primaryDC>RDC1</primaryDC>

  <secondaryDC>RDC2</secondaryDC>

</station>

<OC name="OC1">

  <primaryDC>GDC2</primaryDC>

  <secondaryDC>RDC2</secondaryDC>
```
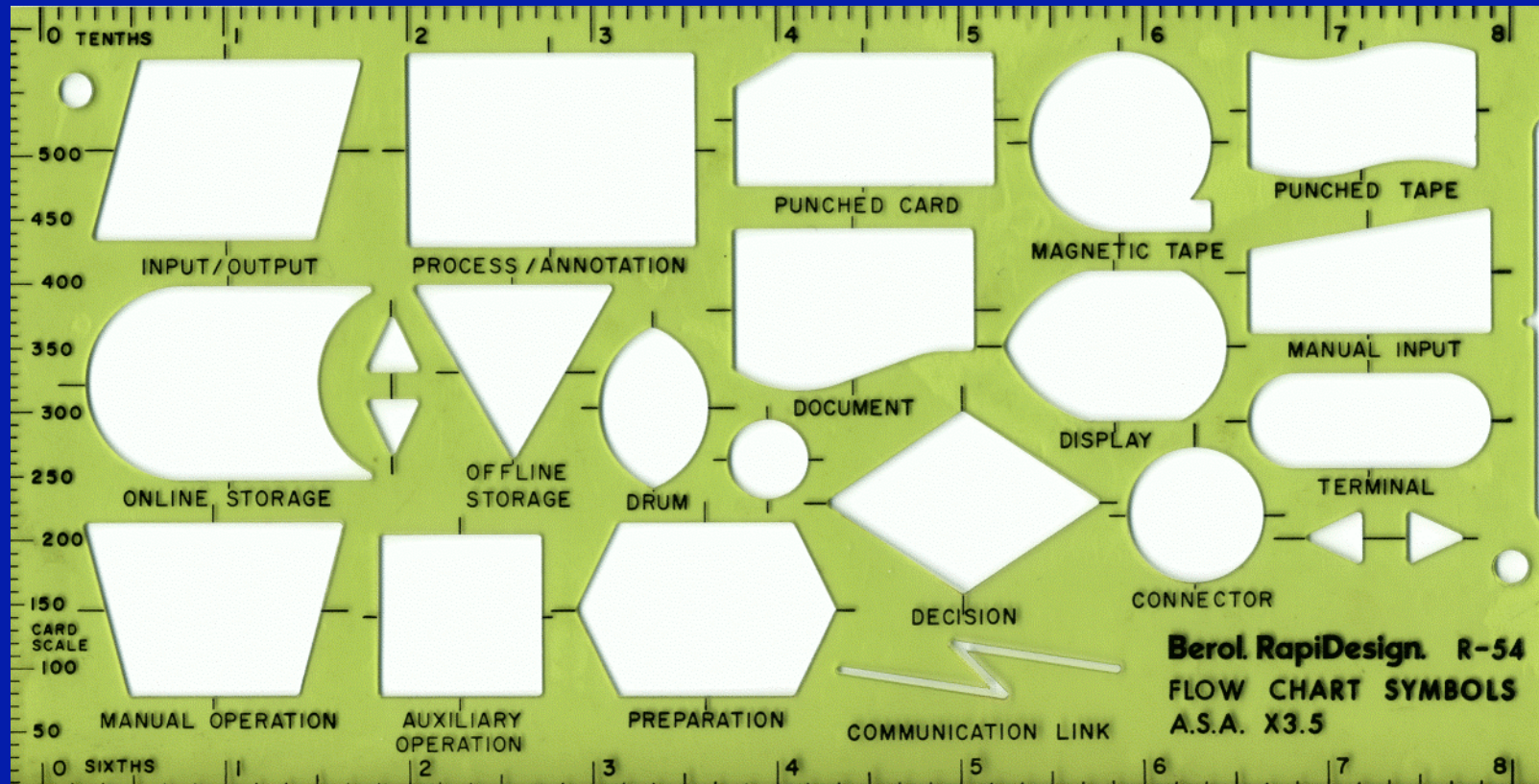
# Rules

- All DCs sleep (wait) 5 minutes before acting on a file pushed to it on a secondary path.

- Any data *pulled* becomes a *private copy* and must not be published or pushed.
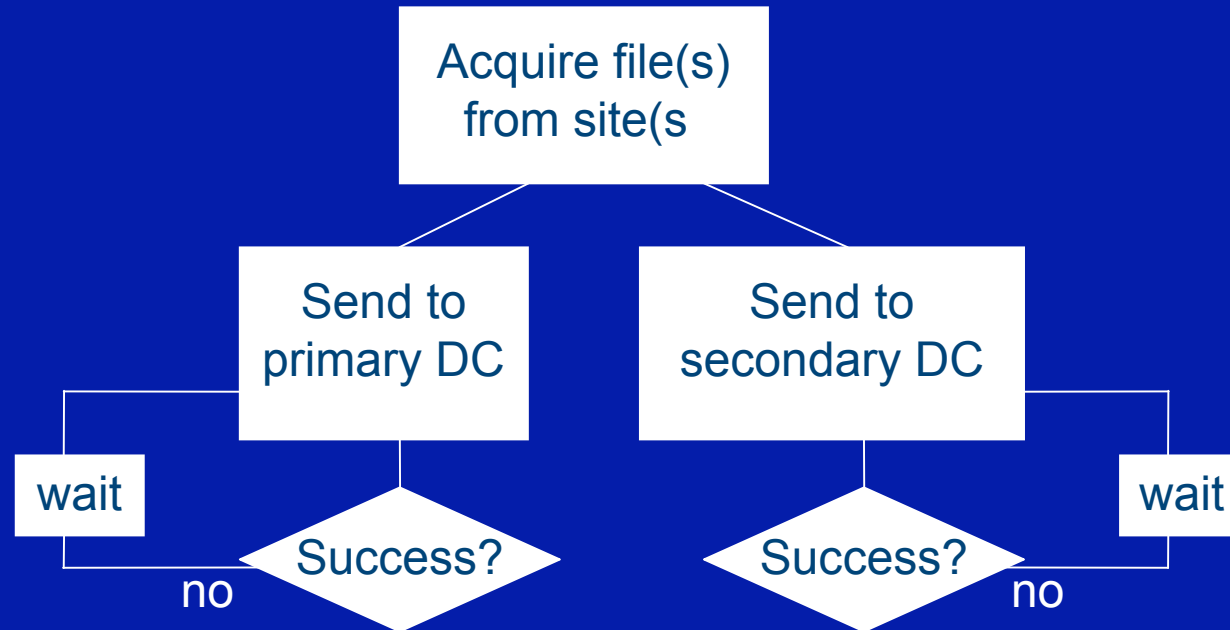
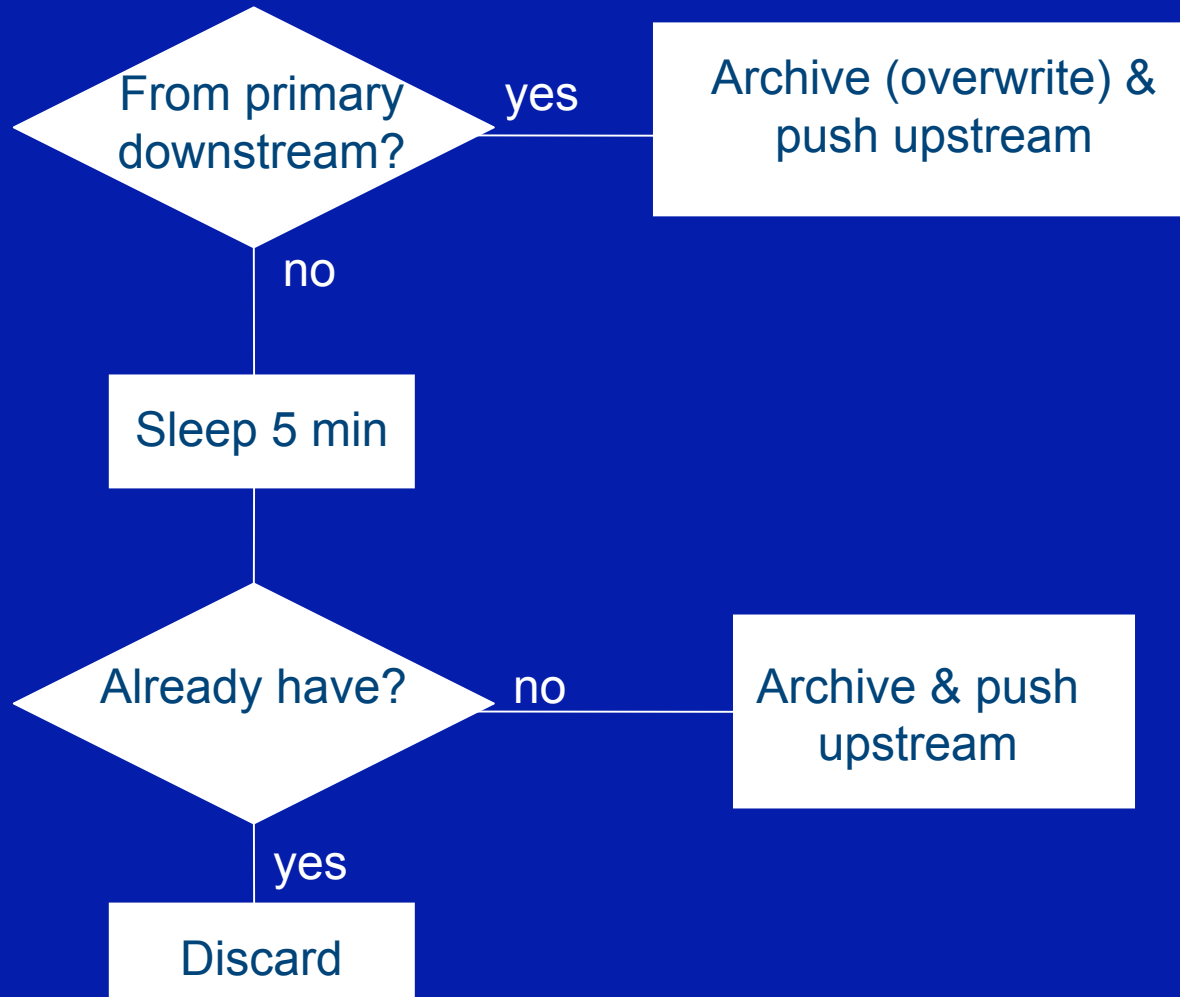- Data is regarded as *read-only*.
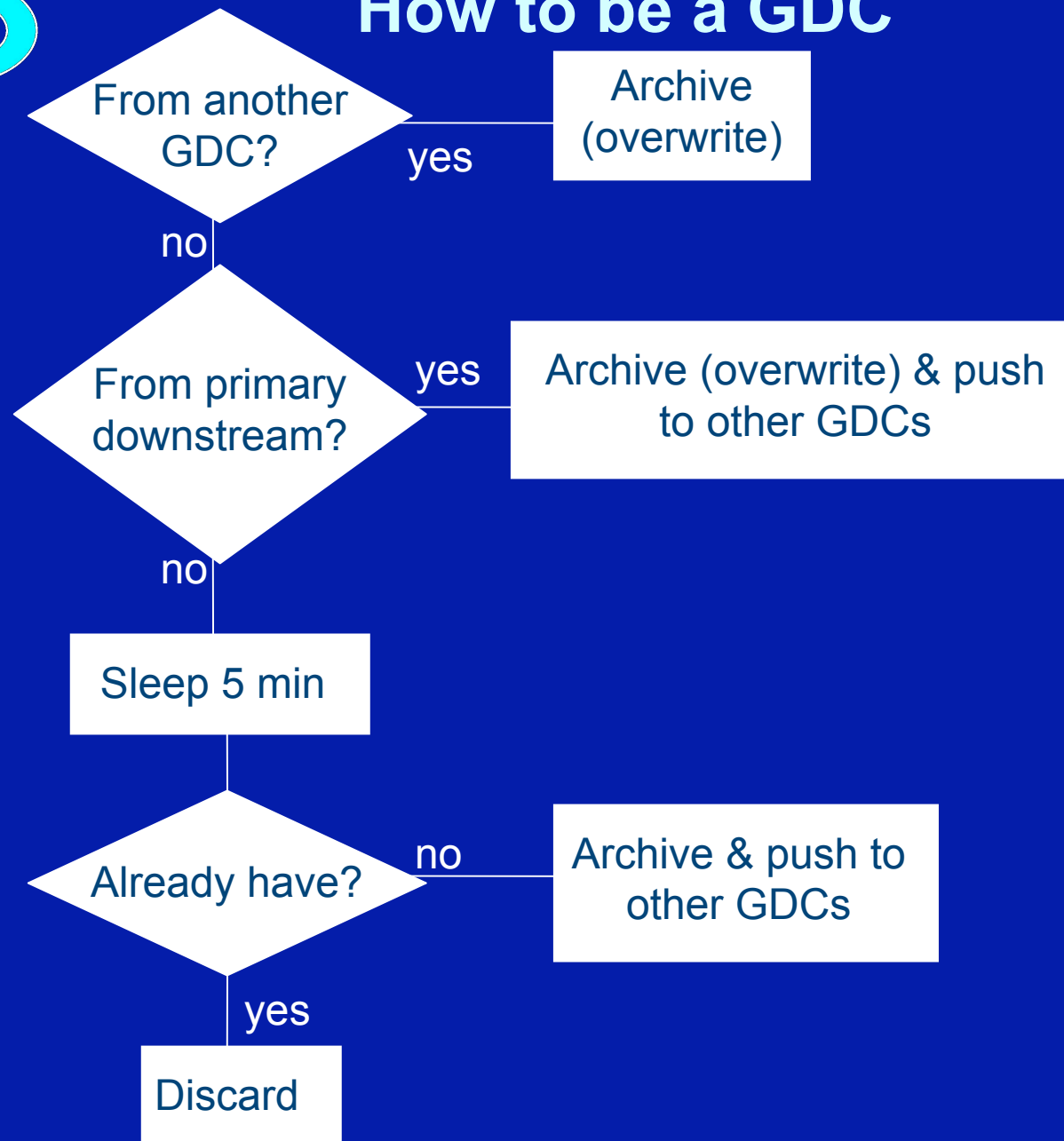
# Here come the flowcharts

# How to be an OC

Acquire file(s) from site(s

Send to primary DC

Send to secondary DC

wait

Success?

no

Success?

no

wait

# How to be an RDC

```
         ┌─────────────────┐   yes   ┌──────────────────────┐
         │ From primary    │─────────│ Archive (overwrite) &│
         │ downstream?     │         │ push upstream        │
         └─────────────────┘         └──────────────────────┘
                  │ no
                  │
         ┌─────────────────┐
         │ Sleep 5 min     │
         └─────────────────┘
                  │
         ┌─────────────────┐   no    ┌──────────────────────┐
         │ Already have?   │─────────│ Archive & push       │
         │                 │         │ upstream             │
         └─────────────────┘         └──────────────────────┘
                  │ yes
         ┌─────────────────┐
         │ Discard         │
         └─────────────────┘
```

# How to be a GDC

**From another GDC?** — yes → Archive (overwrite)

no ↓

**From primary downstream?** — yes → Archive (overwrite) & push to other GDCs

no ↓

Sleep 5 min

↓

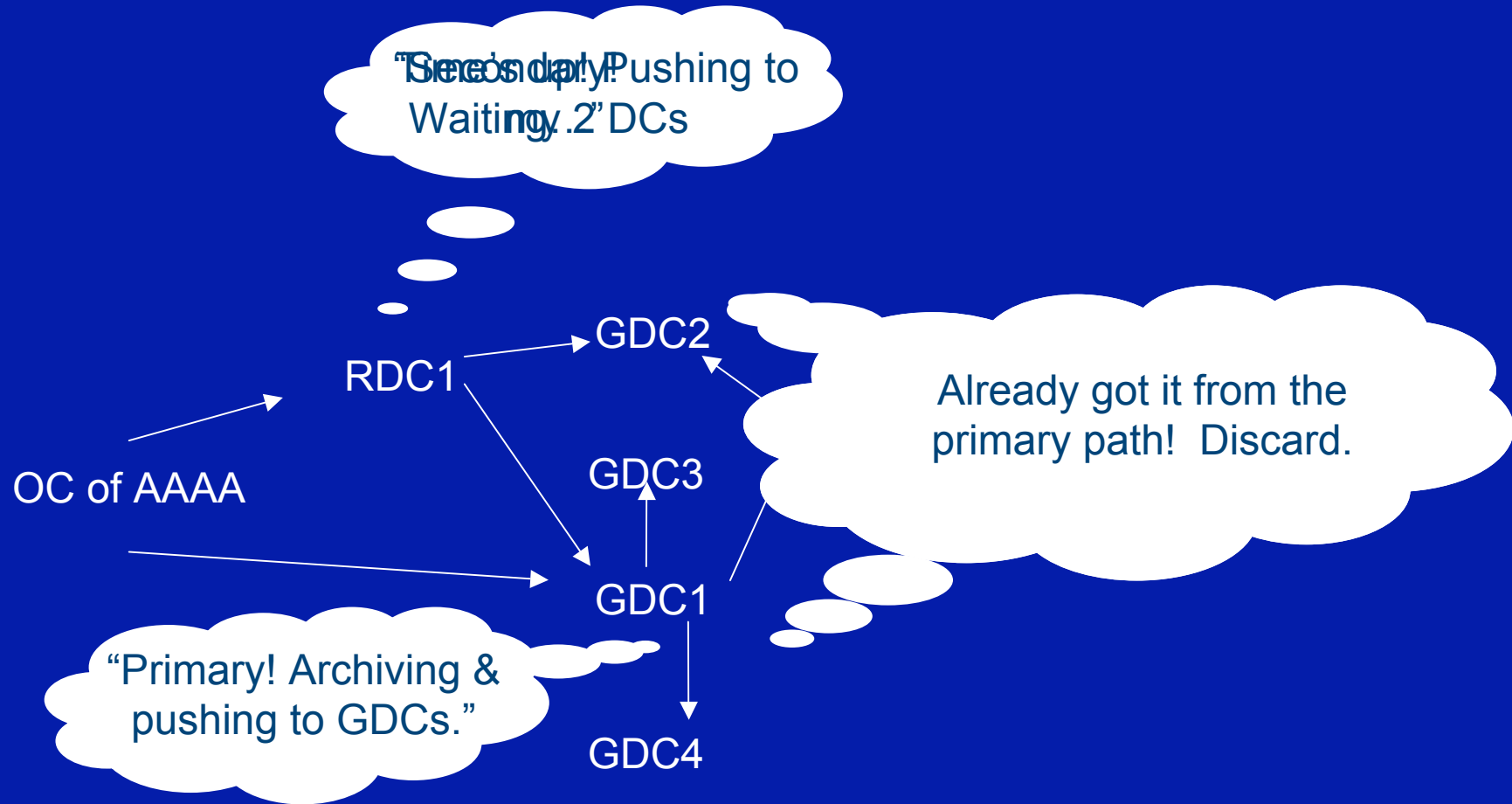**Already have?** — no → Archive & push to other GDCs

yes ↓

Discard

# Example

# On file replacement

The scheme ensures the latest version is archived everywhere.  But how do users learn about a replacement?

- The OC can send out a standardized message on a special purpose mailing list, noting the reason, or

- The OC can put a comment in the RINEX header noting the reason, or

- The OC can transmit a standardized separate file noting the reason.


- DCs can log all replacements with the stated reason (or "unknown".  The entire log can be made available, and additions can be sent to subscribed users.

# The downside(s)

- There *are* some extra transmissions which get discarded.

- There *is* some overwriting of files with possibly equivalent versions, especially after a DC outage.
  - Cannot figure out how to prevent this without going to a unique filename (version) approach or requiring the DC to open up and examine the file.

- These are negatives associated with a high degree of certainty that all DCs have the "correct" file, and that data is available when a DC is down.

- Note that most hourly data is doubly transmitted now.

# Nominal implementation

- Confirm current data flow for each station/OC

- Define primary & secondary DCs for each station/OC

- OCs, ODCs, RDCs, GDCs prepare to transmit and/or receive data on primary and secondary paths

- GDCs implement data equalization rules

- Test

- Go

Recommendation: Pursue this model.