HTCondor Data Story

Greg Thain
Center for High Throughput Computing
University of Wisconsin - Madison

CENTER FOR HIGH THROUGHPUT COMPUTING

PARTNERSHIP to ADVANCE THROUGHPUT COMPUTING

HTCondor Software Suite

Outline

Why file transfer?

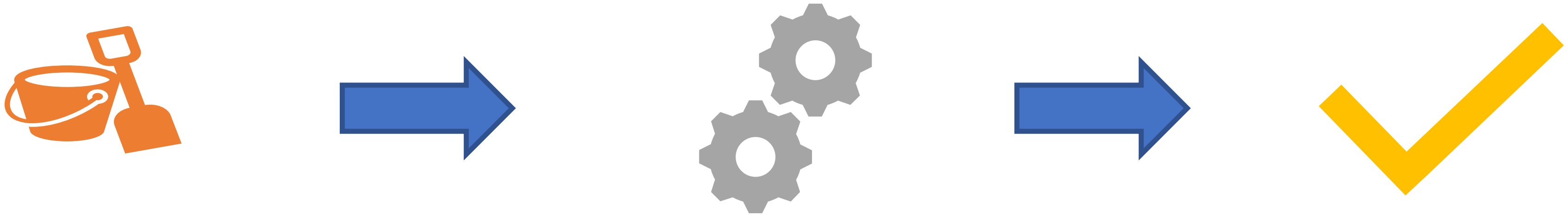
HTCondor built-in or "Cedar" file transfer

Tricks with directories and file transfer

How to use plugins

How to write plugins

Review of HTCondor Job Lifetime



Input Sandbox
(Usually on AP)

Job
(on EP)

Output Sandbox
(back to AP)

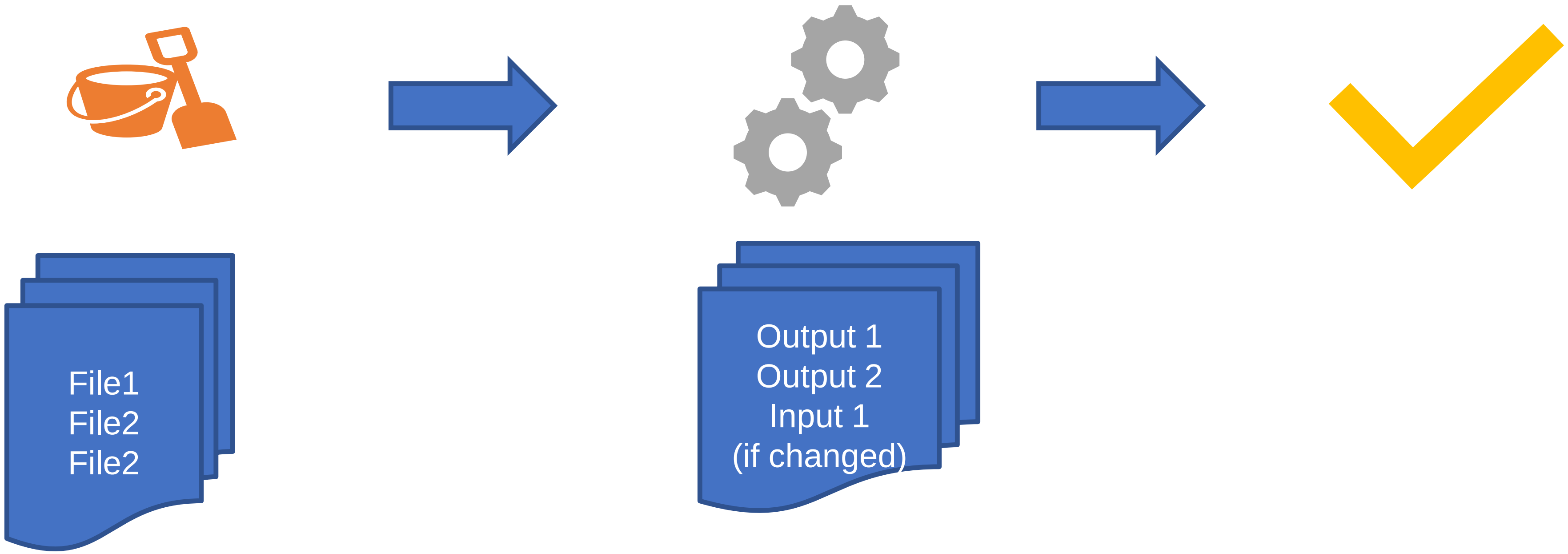
Simple sample HTCondor submit file

```
executable = calculate
arguments = go fast

Should_transfer_files = true
When_to_transfer_files = on_exit
Transfer_input_files = file1, file2, file3

queue
```

Simple HTCondor Job with file xfer



File1
File2
File2

Output 1
Output 2
Input 1
(if changed)

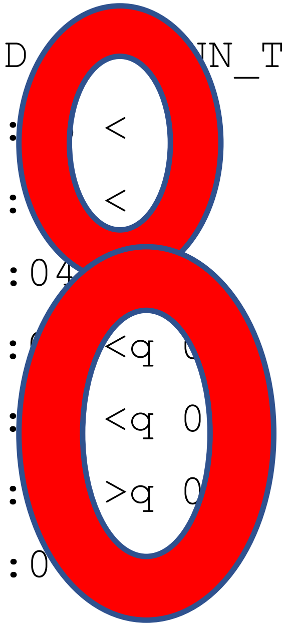
What could go wrong?

- Input file doesn't exist (or is typo!)
  - Result: HTCondor notices, puts job on hold
- EP sandbox dir runs out of disk space
  - Result: HTCondor notices, retries job
- AP has too many concurrent transfers, might overload
  - HTCondor throttles transfers (inputs and outputs)

Aside: symptom of xfer queue backing up…

```
$ condor_q -all -nobatch | head


-- Schedd: submit-1.chtc.wisc.edu : <128.105.244.191:9618?... @ 10/05/22 16:10:12
 ID            OWNER            SUBMITTED       N_TIME ST PRI SIZE      CMD
130.14 g44         10/5  13:48   0+00:00:    <           0.0 generate_part_3
130.15 g44         10/5  13:48   0+00:00:    <           0.0 generate_part_3
130.16 g44         10/5  13:48   0+00:00:04              0.0 generate_part_3
130.17 g44         10/5  13:48   0+00:00:    <q          0.0 generate_part_3
130.18 g44         10/5  13:48   0+00:00:    <q 0        0.0 generate_part_3
130.19 g44         10/5  13:48   0+00:00:    >q 0        0.0 generate_part_3
130.20 g44         10/5  13:48   0+00:00:0               0.0 generate_part_3
```



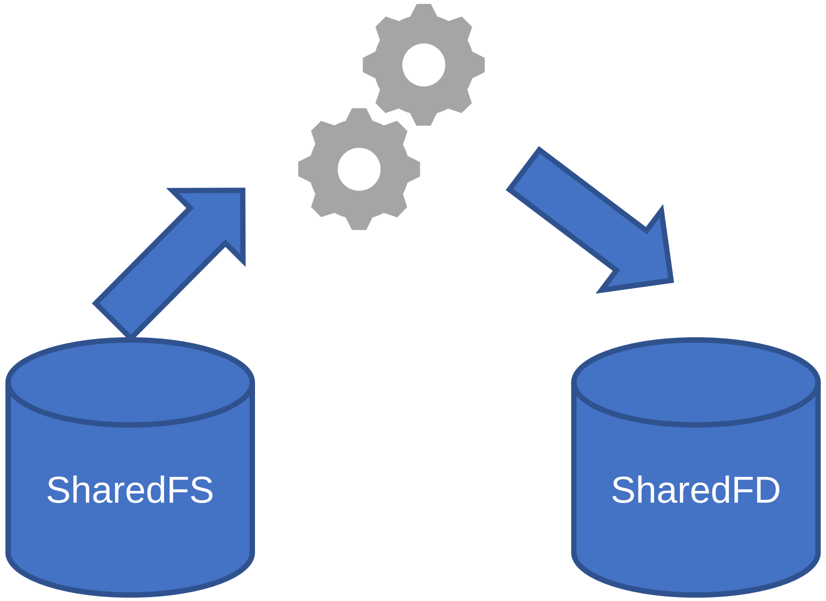
Simple sample HTCondor submit file

```
executable = calculate
arguments = go fast

Should_transfer_files = NEVER
# When_to_transfer_files = on_exit
# Transfer_input_files = file1, file2, file3

queue
```

Simple HTCondor Job with shared fs



What could go wrong?

- Input file doesn't exist (or is typo!)
    - Result: HTCondor doesn't notice, job errors out at runtime
        - Probably leaves queue!
- Shared FS dir runs out of disk space
    - Result: HTCondor doesn't notice, job errors at runtime,
    - Might break other jobs, too
- Shared FS has too many concurrent transfers, might overload
    - NFS / Shared FS breaks, potentially breaking all jobs in pool

Simple sample HTCondor submit file

```
executable = calculate
arguments = go fast

Should_transfer_files = true
When_to_transfer_files = ON_EXIT_OR_EVICT
Transfer_input_files = file1, file2, file3


queue
```

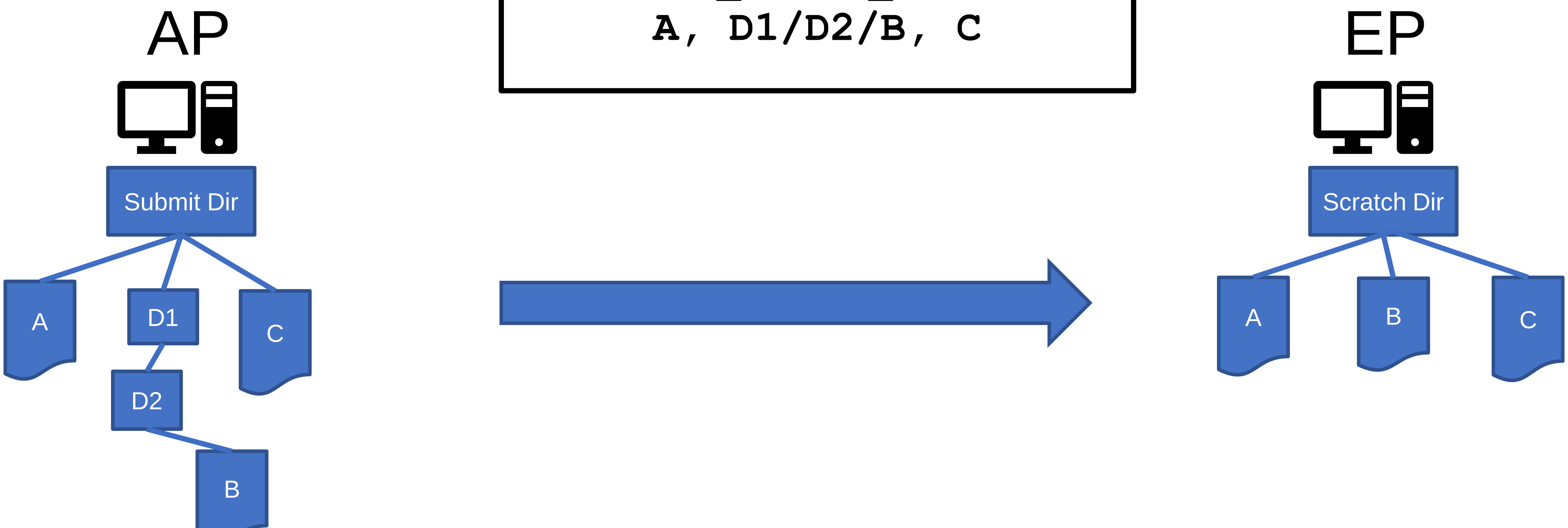
What could go wrong?

- Input file doesn't exist (or is typo!)
  - Result: HTCondor doesn't notice, job errors out at runtime
    - Probably leaves queue!
- Shared FS dir runs out of disk space
  - Result: HTCondor doesn't notice, job errors at runtime,
  - Might break other jobs, too
- Shared FS has too many concurrent transfers, might overload
  - NFS / Shared FS breaks, potentially breaking all jobs in pool

File xfer and directories (input)

HTCondor "flattens" files in directories, by default

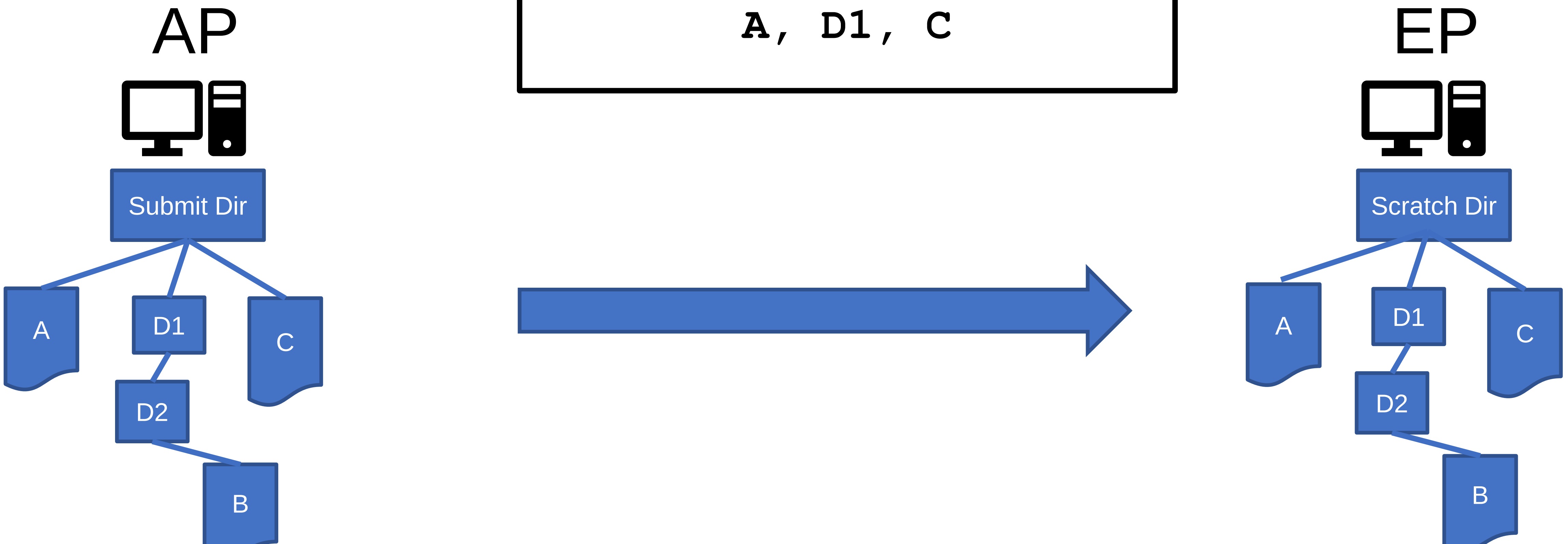
```
transfer_input_files = \
    A, D1/D2/B, C
```



File xfer and directories

But not if directory EXPLICITLY transferred by name:

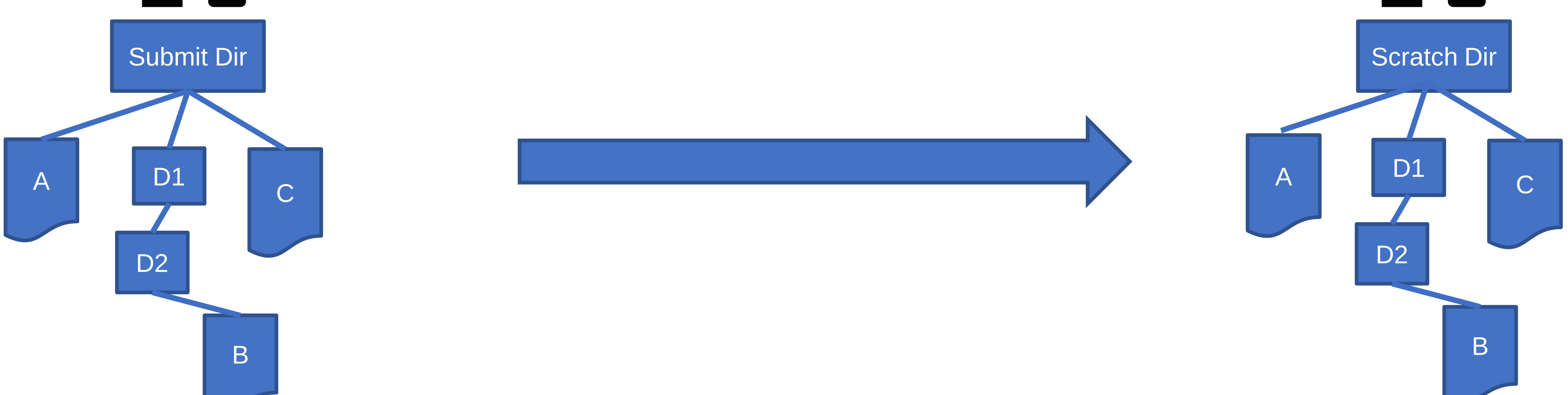
```
transfer_input_files = \
          A, D1, C
```



File xfer and directories (inputs)

OR if preserve_relative_path is true:

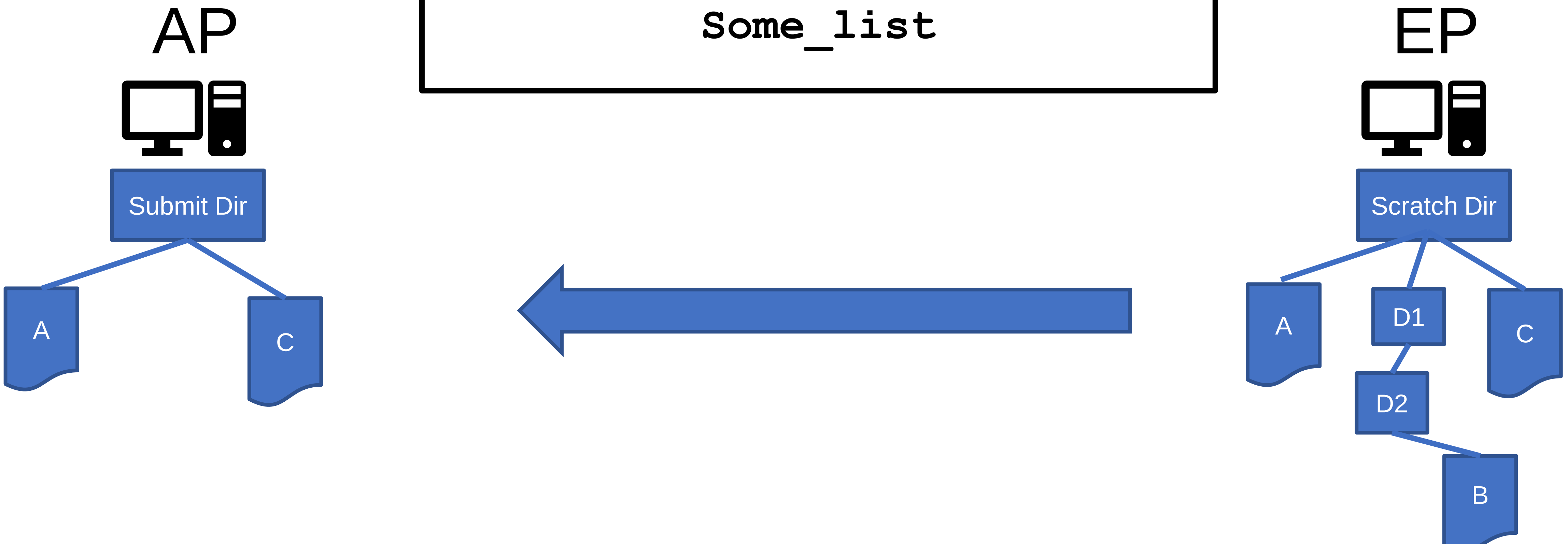
```
preserve_relative_path = true
transfer_input_files = \
            A, D1/D2/B, C
```



File xfer and directories (output)

For OUTPUT, subdirs always skipped…

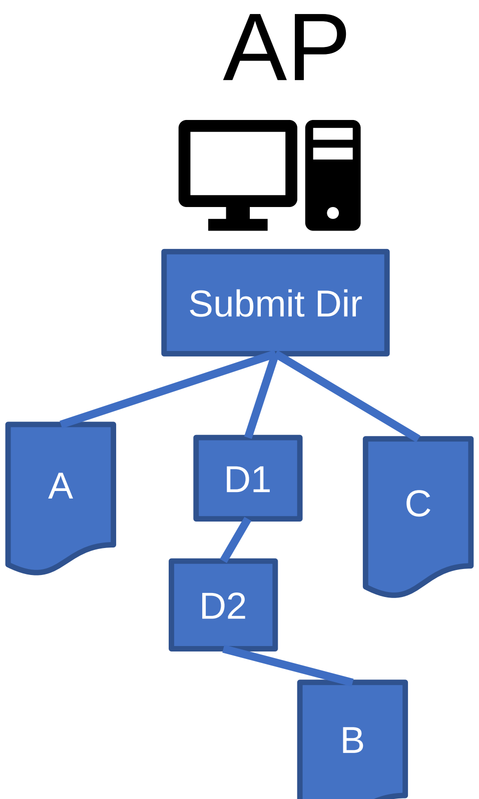
```
transfer_input_files = \
         Some_list
```

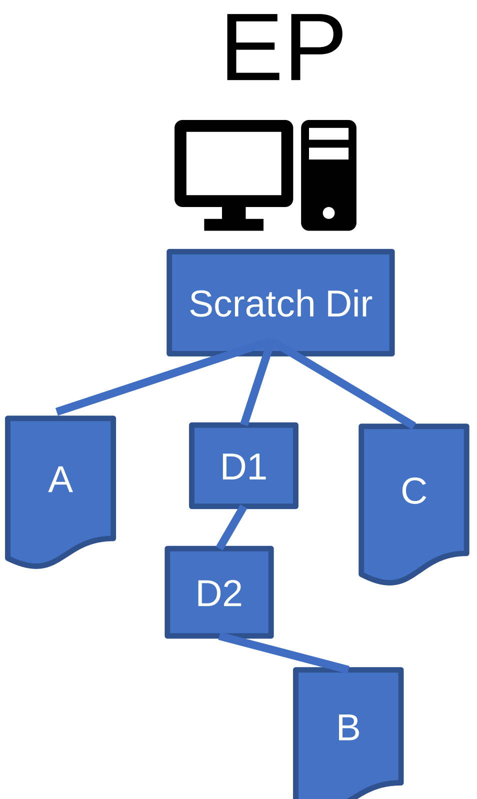


File xfer and directories

Unless output directory explicitly mentioned

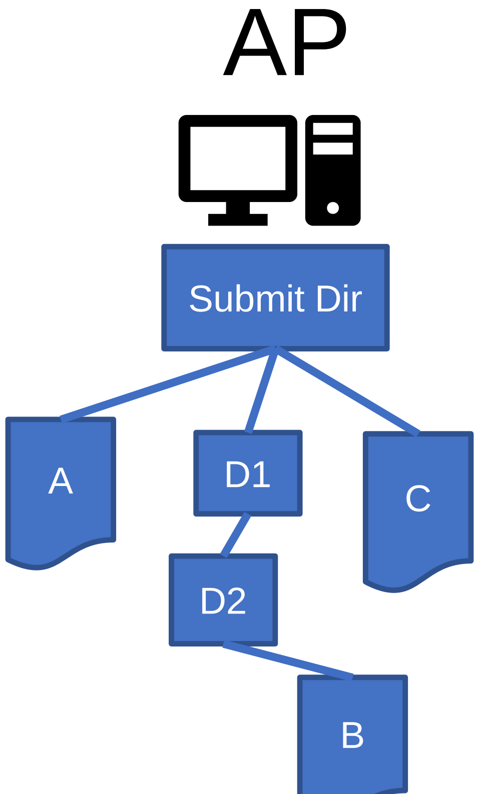
```
transfer_output_files = \
    A, D1, C
```





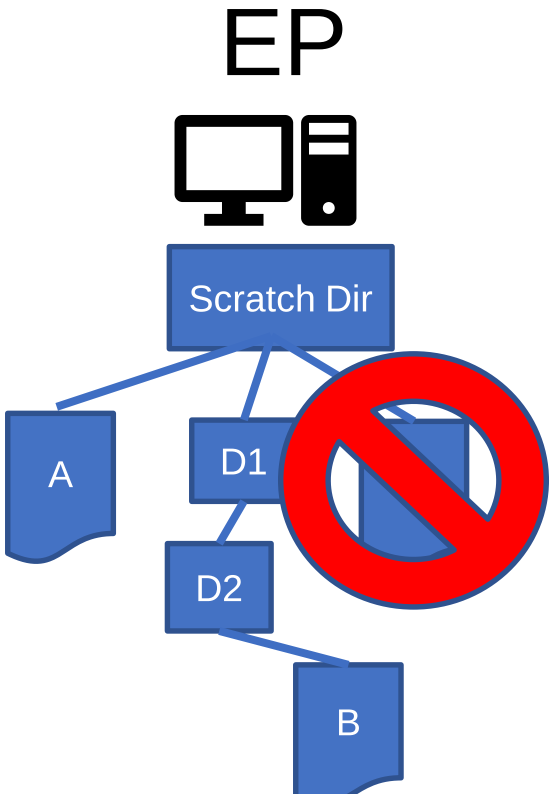
Best practice: omit transfer_output_files

Unless output directory explicitly mentioned



```
transfer_output_files = \
          A, D1, C
```

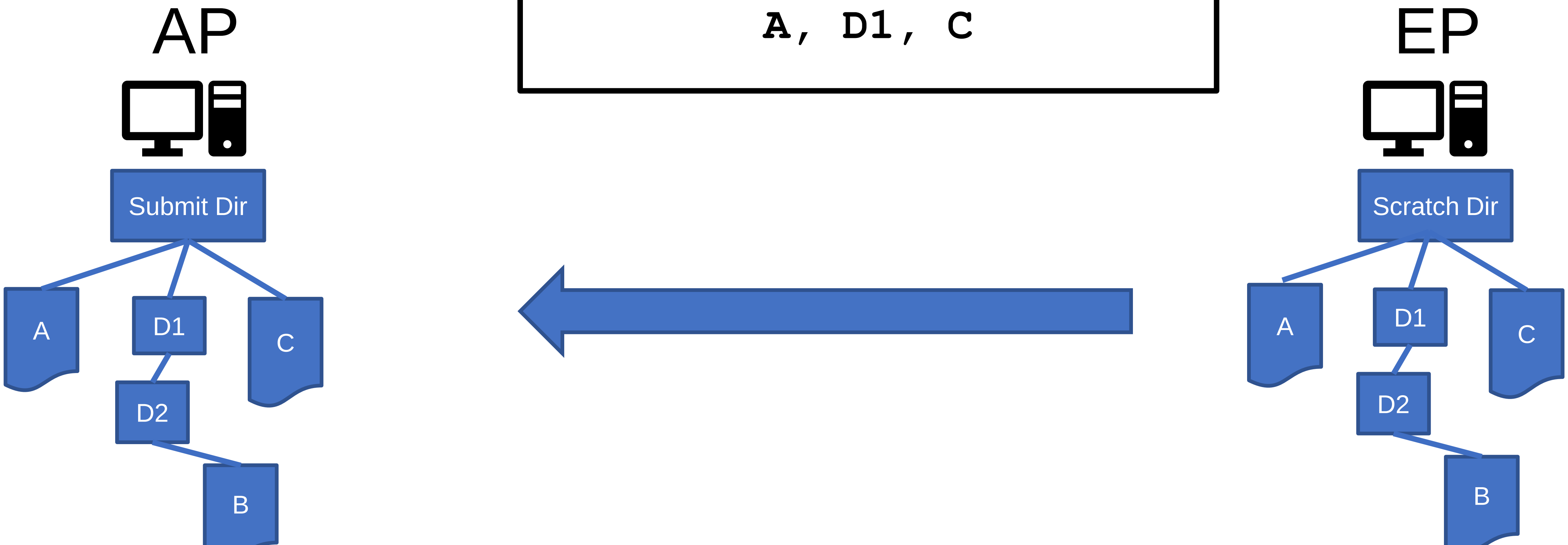
```
    Job goes on HOLD
  "No such file: C"
```



File xfer and directories

Unless output directory explicitly mentioned

```
transfer_output_files = \
        A, D1, C
```



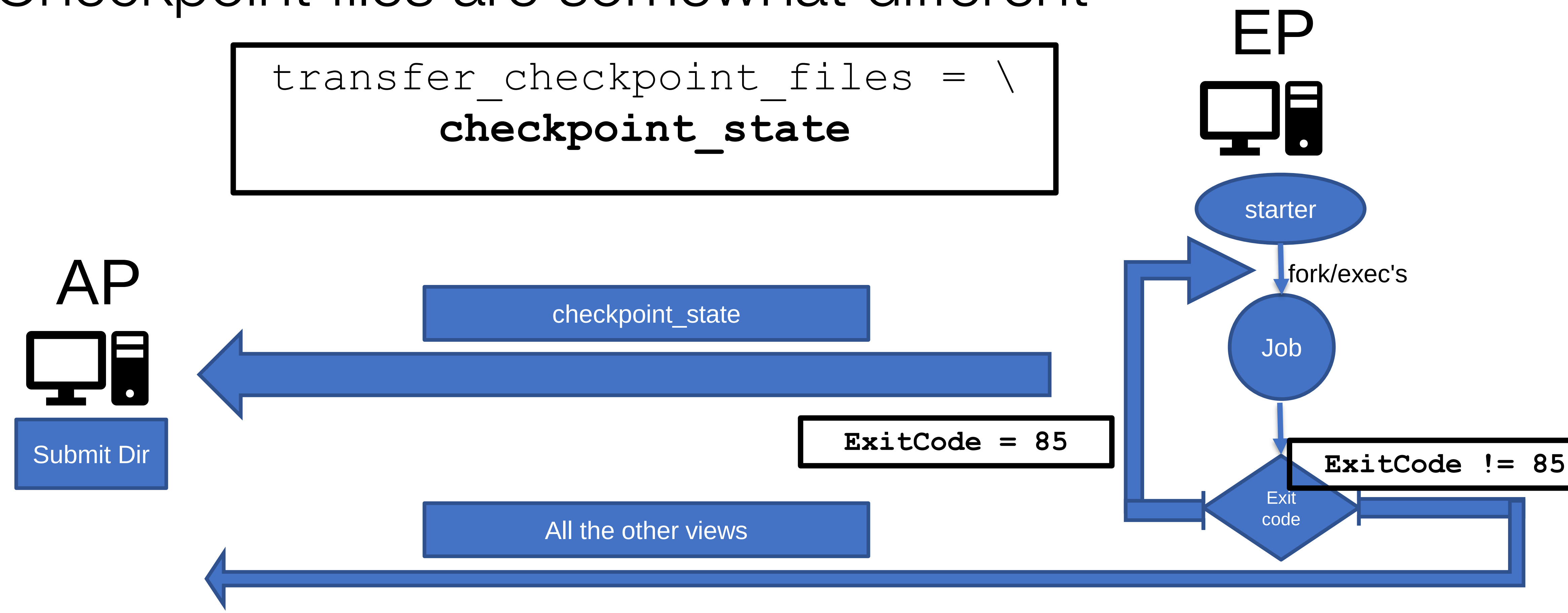
For stdout and stderr only, real time stream

```
# "output" really means "stdout"
output = my_output_file
# "error" really means "stderr"
error = my_error_file
stream_output = true
stream_error  = true
queue
```

User level checkpoint review

Checkpoint files are somewhat different

```
transfer_checkpoint_files = \
    checkpoint_state
```



Third party Transfers (aka file xfer plugins)

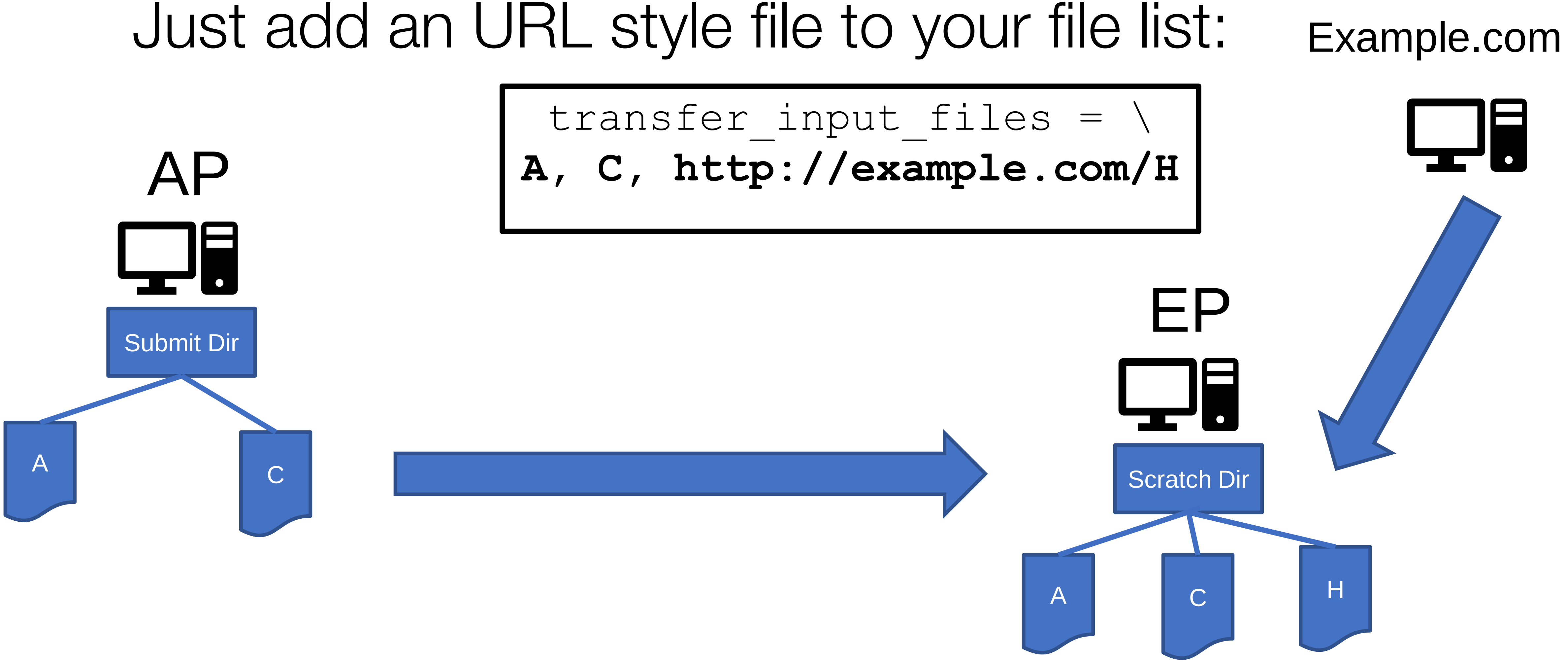
- Sometimes want to xfer from a 3rd party server
  - Instead of manually copying files to AP
- NOT just for performance, but ease of use

- HTCondor calls this "file transfer plugins", or 3rd party transfers

- HTCondor provides "plugins", and you can write your own
- Can mix and match with condor file transfer

File transfer plugins

Just add an URL style file to your file list:

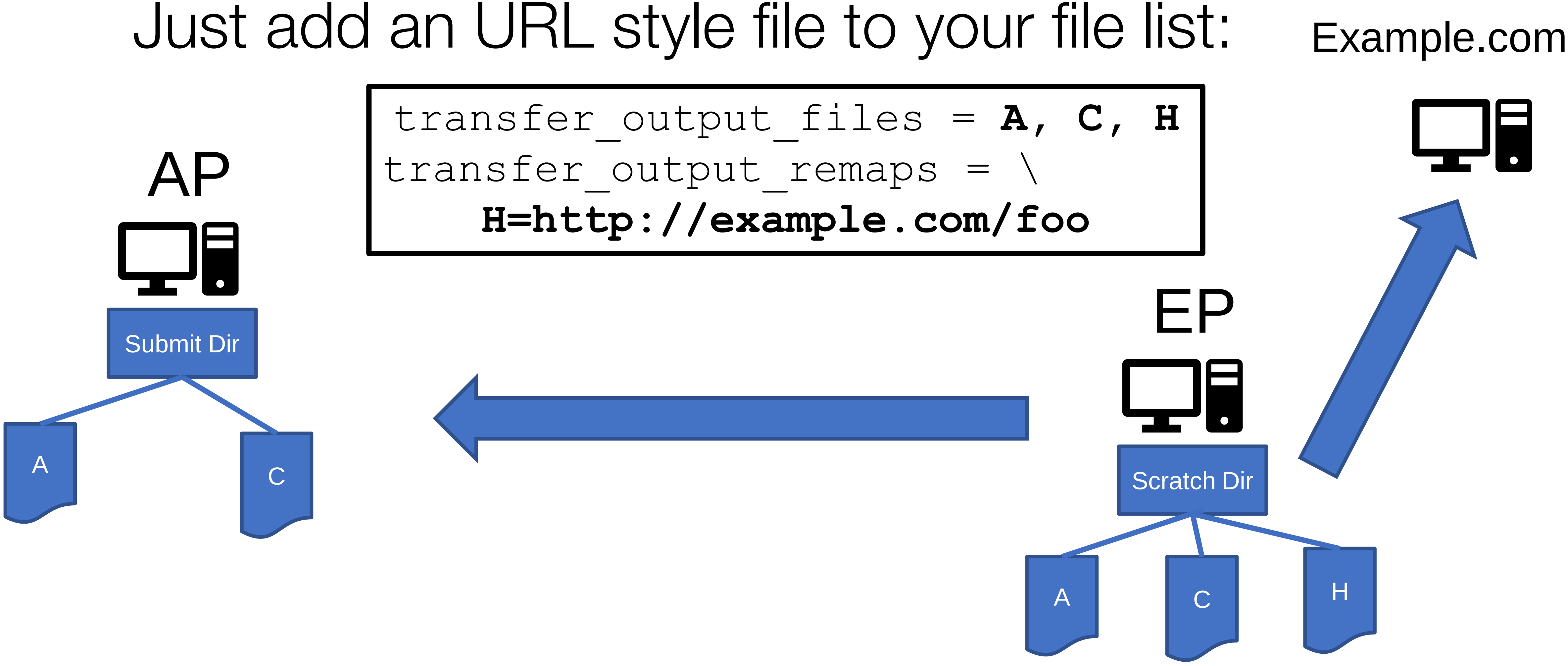


```
transfer_input_files = \
A, C, http://example.com/H
```

Output is a little tricky…

Just add an URL style file to your file list:

```
 transfer_output_files = A, C, H
transfer_output_remaps = \
     H=http://example.com/foo
```



File xfer plugins notes

- Native HTCondor transfer go first (so you can send a credential)
- If plugin fails, job goes on hold
- Plugins work for input and output
- Plugins are one important way to work around wrapper scripts
    - See talk on Friday

file:// xfer plugins notes file:

- file:///
  - "Transfers" a file by copying it from one directory to another
  - Often very useful when you do have a shared fs you want to use
  - Allows HTCondor to check for file xfer errors up front
  - If you absolutely need a shared filesystem, this can be a good compromise

Existing file transfer plugins

- file://
- https://
- http://
- dav://
- davs://
- data://
- ftp://
- s3://
- gs://

Writing your own --detection

```
FILE_TRANSFER_PLUGINS = yours, $(FILE_TRANSFER_PLUGINS)
```

```
$ yours -classad
MultipleFileSupport = true
PluginVersion = "0.2"
PluginType = "FileTransfer"
SupportedMethods = "yours, mine, foo"
```

Writing your own – actually doing the xfer

```
<starter> $ yours -infile .foo.in -outfile .foo.out
```

```
[ LocalFileName = "/scratchd/from_submit";
Url ="yours://path/from_submit" ]
[ LocalFileName =
"/scratchd/from_submit2";
Url = "yours://path/from_submit2" ]
```

Writing your own – telling HTCondor news

```
<starter> $ yours -infile .foo.in -outfile .foo.out
```

```
[ TransferUrl = "yours://from_submit/file";
  TransferSuccess = true;
]
[ TransferUrl = "yours://from_submit/file2";
TransferSuccess = false;
TransferError = "Error: Non capisco niente";
]
```

BYOP: Bring your own plugin

```
transfer_plugins = yours
transfer_input_file = yours://path_from/submit
```

Non-transfer transfer plugins

- Sometimes you want to hold a job if something is **wrong**
- Something that you can only detect on EP before job starts
- But something only the job knows
  - Can't get license?
  - Shared fs is broken?
- Can do this with non-transfer file transfer

Future work

- File transfer plugins are mature, and heavily used

- Especially the shipped one, like http

- But error reporting can't be Boolean + Human string
- And result can't be either success or "go on hold"

- Thinking about using HTTP error codes for better reporting

Summary

- HTCondor provide rich methods for transferring data
- Explicit file transfer is a bit more effort to set up than shared fs
- But often provides better quality of life

Thank you and questions

Thank you – Questions?

This work is supported by the NSF under Cooperative Agreement OAC-2030508. Any options, findings, conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the NSF.