Partial success scenarios with P2300

The following slides were presented during LEWG/SG1 discussion of P2300 to illustrate the problem of delivering “partial success” results. An operation with “partial success” semantics delivers both the reason for failing to achieve complete success, and an indication of what partial side effects have been established. Due to the limitations of the set_error channel (which has a single “error” argument) and set_done channel (which takes no arguments), partial results must be communicated down the set_value channel. When one operation is composed in terms of another, a completion on any of the set_value, set_error, or set_done (i.e. cancellation) channels can be a trigger for a partial result.
async_read_some

set_value
completion scheduler = S1

set_error
completion scheduler = S2
noexcept

set_done
completion scheduler = S3
noexcept

async_read
wants 1024 bytes
has 42 so far

set_value
completion scheduler = ?

set_error
completion scheduler = ?
noexcept

set_done
completion scheduler = ?
noexcept

user code
async_read_some success, async_read complete
async_read_some failure

- async_read_some
  - set_value
    - completion scheduler = S1
  - set_error
    - completion scheduler = S2
    - noexcept
  - set_done
    - completion scheduler = S3
    - noexcept

- async_read
  - wants 1024 bytes
  - has 42 so far
  - set_value
    - completion scheduler = ?

- user code
  - set_value
    - completion scheduler = ?
  - set_error
    - completion scheduler = ?
    - noexcept
  - set_done
    - completion scheduler = ?
    - noexcept
async_read_some cancelled

async_read_some
  - set_value
    - completion scheduler = S1
  - set_error
    - completion scheduler = S2
      - noexcept
  - set_done
    - completion scheduler = S3
      - noexcept

async_read
  - wants 1024 bytes
  - has 42 so far

set_value
  - completion scheduler = ?

set_error
  - completion scheduler = ?
    - noexcept

set_done
  - completion scheduler = ?
    - noexcept

user code
async_read_some cancel attempted, success wins race, check stop_token explicitly

- async_read_some
  - set_value (completion scheduler = S1)
  - set_error (completion scheduler = S2, noexcept)
  - set_done (completion scheduler = S3, noexcept)

- async_read
  - wants 1024 bytes
  - has 42 so far

- set_value (completion scheduler = ?)
- set_error (completion scheduler = ?, noexcept)
- set_done (completion scheduler = ?, noexcept)

- user code
async_read_some cancel attempted, success wins race, check stop_token implicitly