Fall 2022
Earth Radiation
Budget Workshop
Hamburg, Germany

CERES DATA MANAGEMENT TEAM WORKING GROUP REPORT

KATHLEEN DEJWAKH
OCTOBER 12, 2022

OUTLINE

Team & Highlights

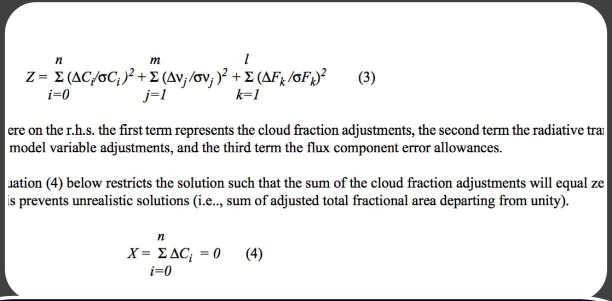
Processing Status

Code Re-architecture

External Interfaces

TEAM & HIGHLIGHTS

RESPONSIBILITIES



$$Z = \sum_{i=0}^{n} (\Delta C_i/\sigma C_i)^2 + \sum_{j=1}^{m} (\Delta v_j/\sigma v_j)^2 + \sum_{k=1}^{l} (\Delta F_k/\sigma F_k)^2 \qquad (3)$$

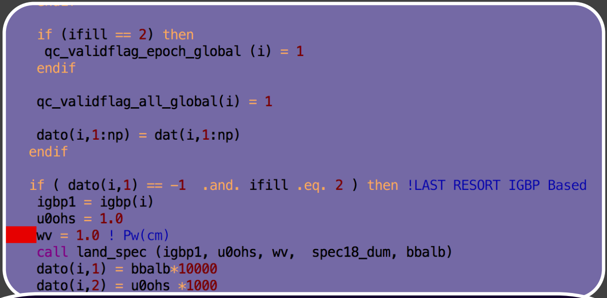
ere on the r.h.s. the first term represents the cloud fraction adjustments, the second term the radiative tra model variable adjustments, and the third term the flux component error allowances.

ıation (4) below restricts the solution such that the sum of the cloud fraction adjustments will equal ze ıs prevents unrealistic solutions (i.e.., sum of adjusted total fractional area departing from unity).

$$X = \sum_{i=0}^{n} \Delta C_i = 0 \qquad (4)$$

Algorithms

- Implementation
- Verification
- Validation assistance



```
 if (ifill == 2) then
  qc_validflag_epoch_global (i) = 1
 endif

 qc_validflag_all_global(i) = 1

 dato(i,1:np) = dat(i,1:np)
endif

if ( dato(i,1) == -1  .and. ifill .eq. 2 ) then !LAST RESORT IGBP Based
 igbp1 = igbp(i)
 u0ohs = 1.0
 wv = 1.0 ! Pw(cm)
 call land_spec (igbp1, u0ohs, wv,  spec18_dum, bbalb)
 dato(i,1) = bbalb*10000
 dato(i,2) = u0ohs *1000
```

Software

- Configuration management

- Maintenance
  - Updates
  - Infrastructure

- Website development

DATA MANAGEMENT TEAM

SSAI Management:

Walter Miller
Susan Thomas

Tom Caldwell

Sunny Sun-Mack
Ricky Brown
Steve Kohler
Yan Chen
Elizabeth Heckert
Rita Smith
Walt Miller

Dale Walikainen
Jeremie Lande

Victor Sothcott
Igor Antropov

Josh Wilkins
Cathy Nguyen
Ed Kizer
Beau Branch

Denise Cooper
Thomas Grepiotis
Hunter Winecoff
Dianne Snyder
Dale Walikainen

PC Sawaengphokhai
Jay Garg
Hunter Winecoff

Nelson Hillyer
Tammy Ayers
Dennis Keyes
Willinda Evans

Carla Grune
Elizabeth Heckert

HIGHLIGHTS

- NOAA-20 one-month gain deliveries (from two-month)

- TISA Averaging
  - "NOGEO" SYN1deg deliveries supporting EBAF 4.2
  - TSIB deliveries supporting Edition 4B

- Clouds/CERESlib deliveries: snow-and-ice data input change

- CATALYST and archiver support:
  - Dark Horse ingest alterations
  - Adjusting for Level-0 data arrival discrepancies

HIGHLIGHTS

- FLASHFlux fully integrated into CATALYST
  - Performance meets or exceeds that of previous system
    - More reliably captures available data
    - Robust to missing data
  - Currently incorporating PGEs supporting NOAA-20 production

- Coming soon:

  PHP 7.x security support ends Nov. 28, 2022 – various PHP 8.x deliveries

PROCESSING STATUS

EDITION 4 PRODUCT AVAILABILITY

Product	Platform	Processed Thru	Publicly Available?
BDS	Terra, Aqua	June ’22	Yes
SSF			
SSF1deg-Hour			
SSF1deg-Day/-Month			
SYN1deg-1Hour/MHour	Terra+Aqua	March ‘22	
SYN1deg-Day/-Month			

EDITION 4 PRODUCT AVAILABILITY

Product	Platform	Processed Thru	Publicly Available?
FluxByCldTyp	Terra+Aqua	June ‘22	Yes
CldTypHist		March ‘22	
EBAF			
EBAF ToA			

EDITION 2 S-NPP PRODUCT AVAILABILITY

Edition	Product	Platform	Processed Thru	Publicly Available?
Ed2A	BDS	S-NPP	May '22	Yes
	SSF			
	SSF1deg-Hour		Sept. '19*	
	SSF1deg-Day/-Month			

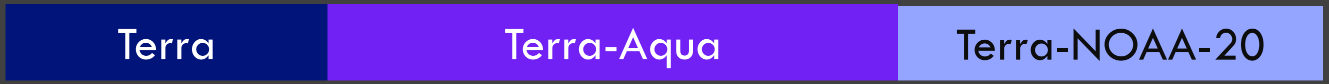
* L3 processing stopped. Instrument in RAPS mode.

EDITION 1B NOAA-20 PRODUCT AVAILABILITY

Product	Platform	Processed Thru	Publicly Available?
BDS	NOAA-20	July ‘22	Yes
SSF			
SSF1deg-Hour			
SSF1deg-Day/-Month			

EDITION 4 REPROCESSING

- Edition4A

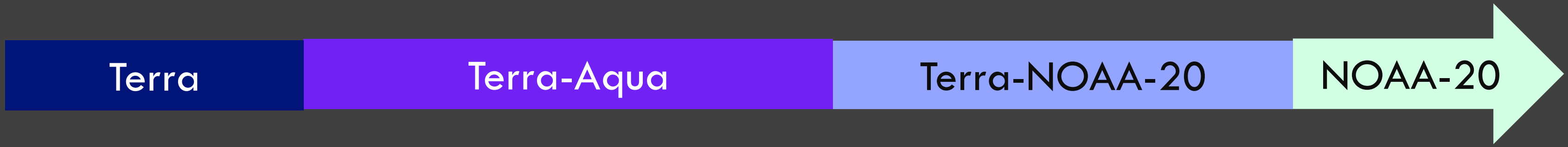


Terra

Terra-Aqua

Terra-NOAA-20

- Edition4B



Terra

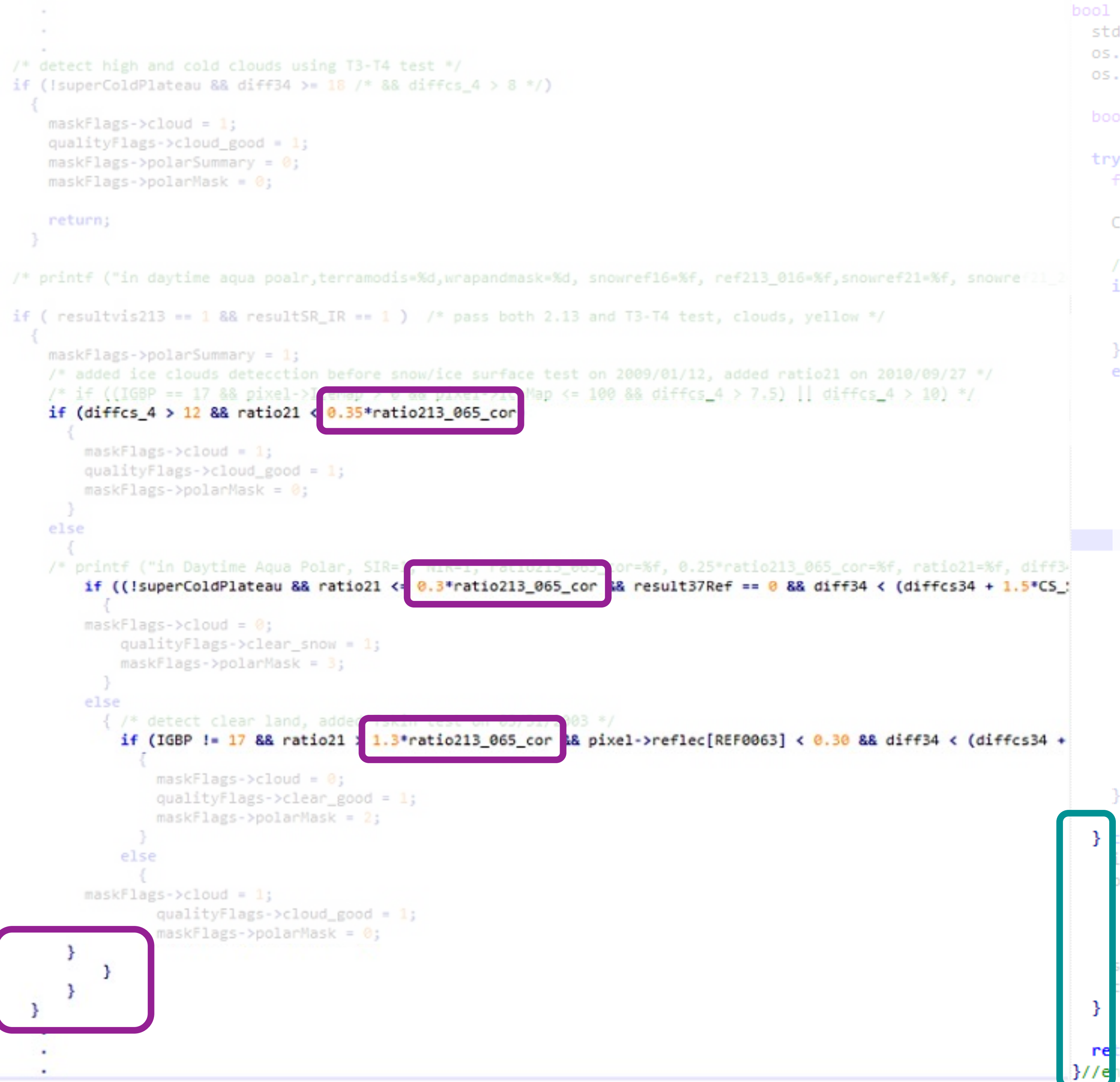
Terra-Aqua

Terra-NOAA-20

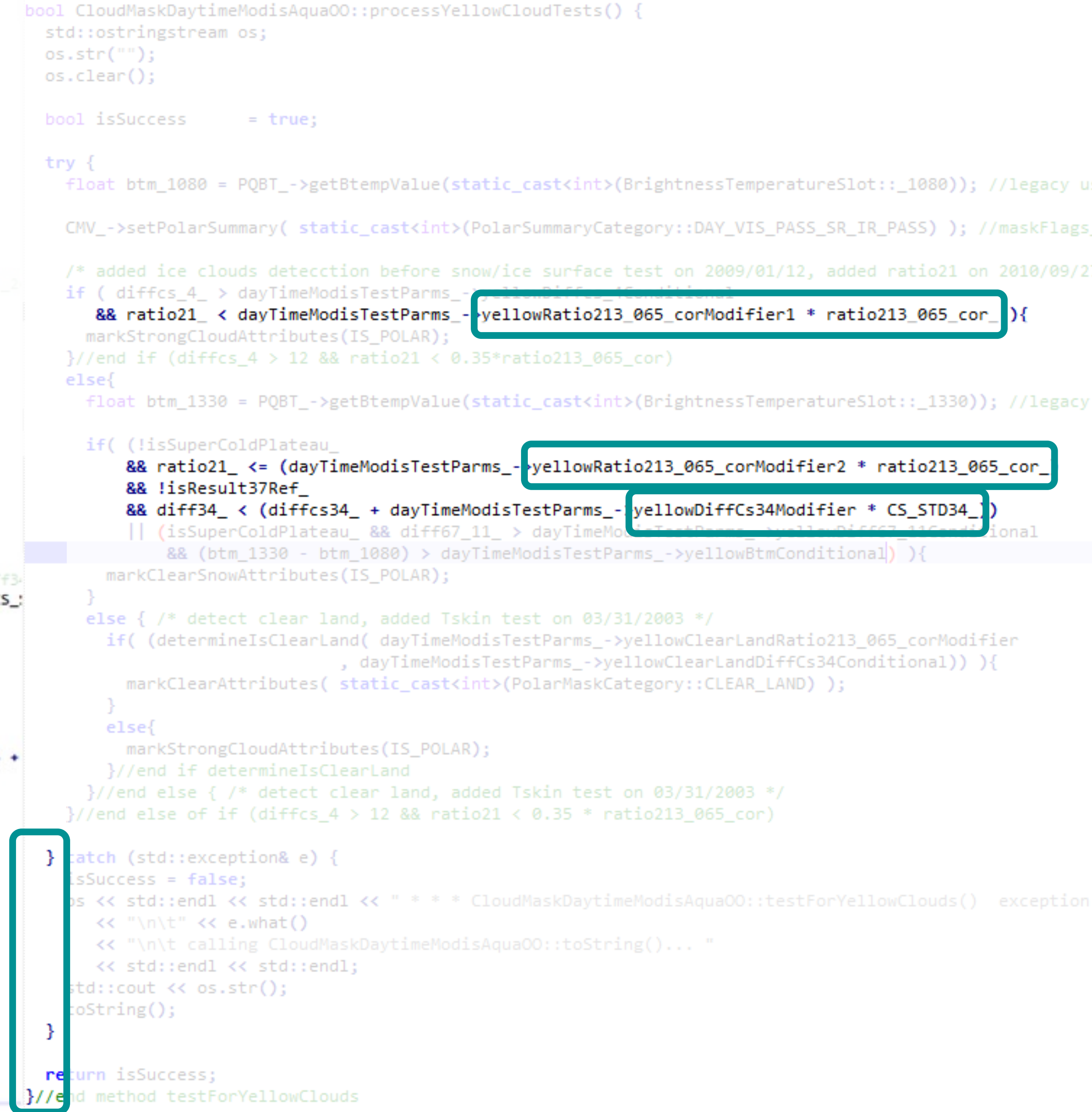
NOAA-20

CODE RE-ARCHITECTURE

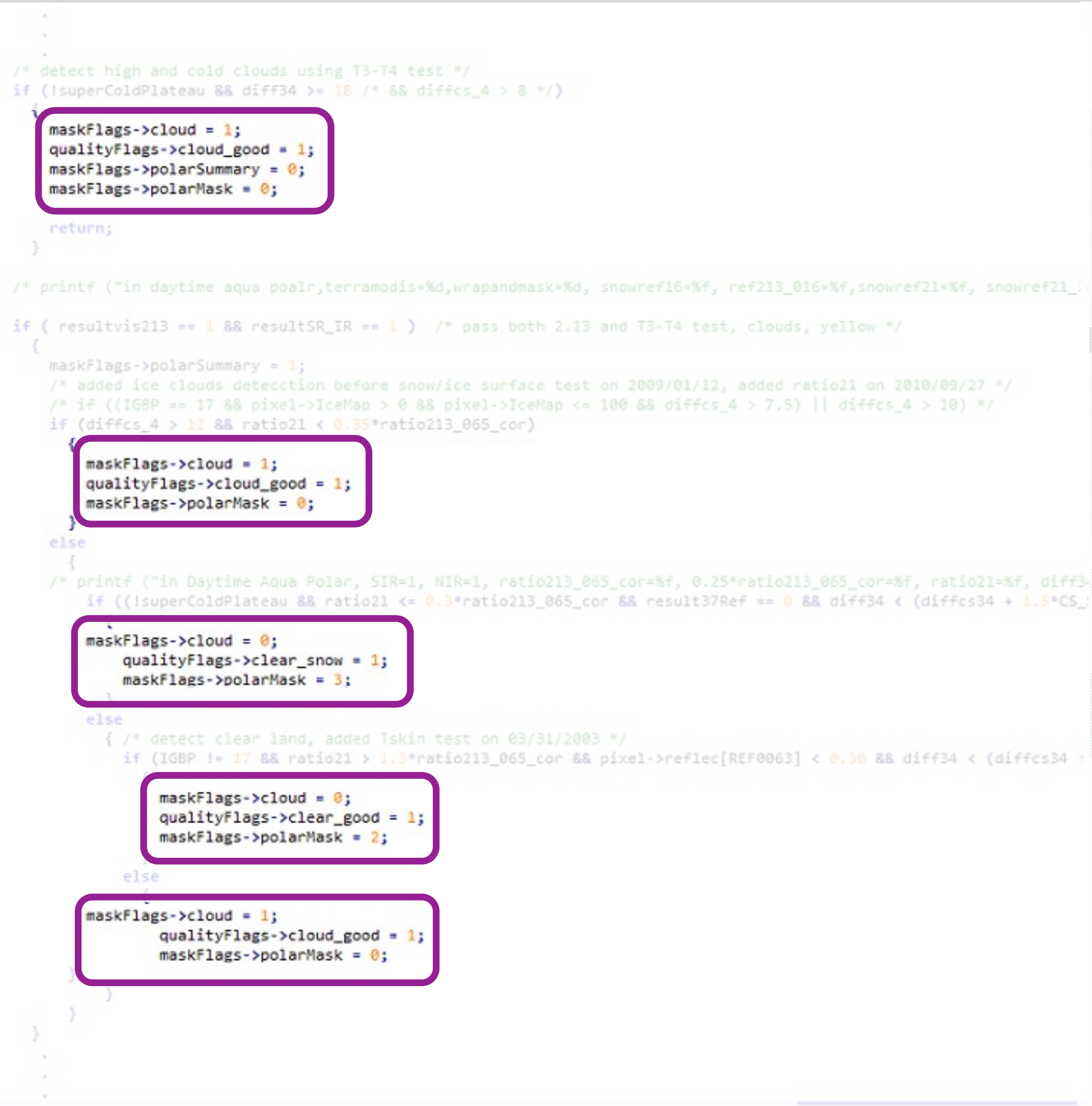
Clouds



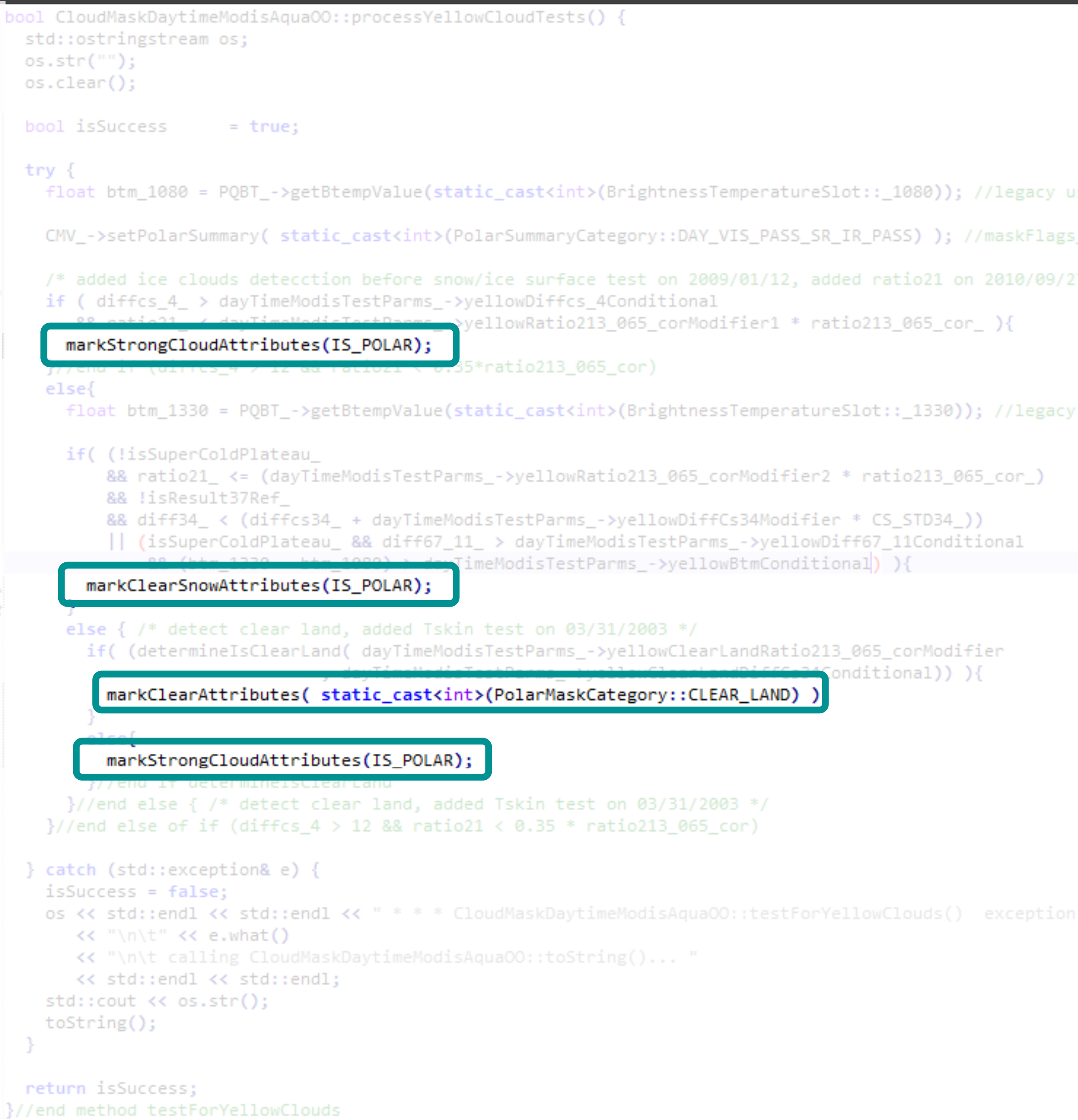
```
  .
  .
  .
/* detect high and cold clouds using T3-T4 test */
if (!superColdPlateau && diff34 >= 18 /* && diffcs_4 > 8 */)
  {
    maskFlags->cloud = 1;
    qualityFlags->cloud_good = 1;
    maskFlags->polarSummary = 0;
    maskFlags->polarMask = 0;

    return;
  }

/* printf ("in daytime aqua poalr,terramodis=%d,wrapandmask=%d, snowref16=%f, ref213_016=%f,snowref21=%f, snowre

if ( resultvis213 == 1 && resultSR_IR == 1 )  /* pass both 2.13 and T3-T4 test, clouds, yellow */
  {
    maskFlags->polarSummary = 1;
    /* added ice clouds detecction before snow/ice surface test on 2009/01/12, added ratio21 on 2010/09/27 */
    /* if ((IGBP == 17 && pixel->Icemap > 0 && pixel->IceMap <= 100 && diffcs_4 > 7.5) || diffcs_4 > 10) */
    if (diffcs_4 > 12 && ratio21 < 0.35*ratio213_065_cor
      {
        maskFlags->cloud = 1;
        qualityFlags->cloud_good = 1;
        maskFlags->polarMask = 0;
      }
    else
      {
    /* printf ("in Daytime Aqua Polar, SIR=1, NIR=1, ratio213_065_cor=%f, 0.25*ratio213_065_cor=%f, ratio21=%f, diff3
        if ((!superColdPlateau && ratio21 <= 0.3*ratio213_065_cor && result37Ref == 0 && diff34 < (diffcs34 + 1.5*CS_
          {
        maskFlags->cloud = 0;
            qualityFlags->clear_snow = 1;
            maskFlags->polarMask = 3;
          }
        else
          { /* detect clear land, added Tskin test on 03/31/2003 */
            if (IGBP != 17 && ratio21 > 1.3*ratio213_065_cor && pixel->reflec[REF0063] < 0.30 && diff34 < (diffcs34 +
              {
                maskFlags->cloud = 0;
                qualityFlags->clear_good = 1;
                maskFlags->polarMask = 2;
              }
            else
              {
        maskFlags->cloud = 1;
                qualityFlags->cloud_good = 1;
                maskFlags->polarMask = 0;
        }
          }
        }
  }
  .
  .
  .
```



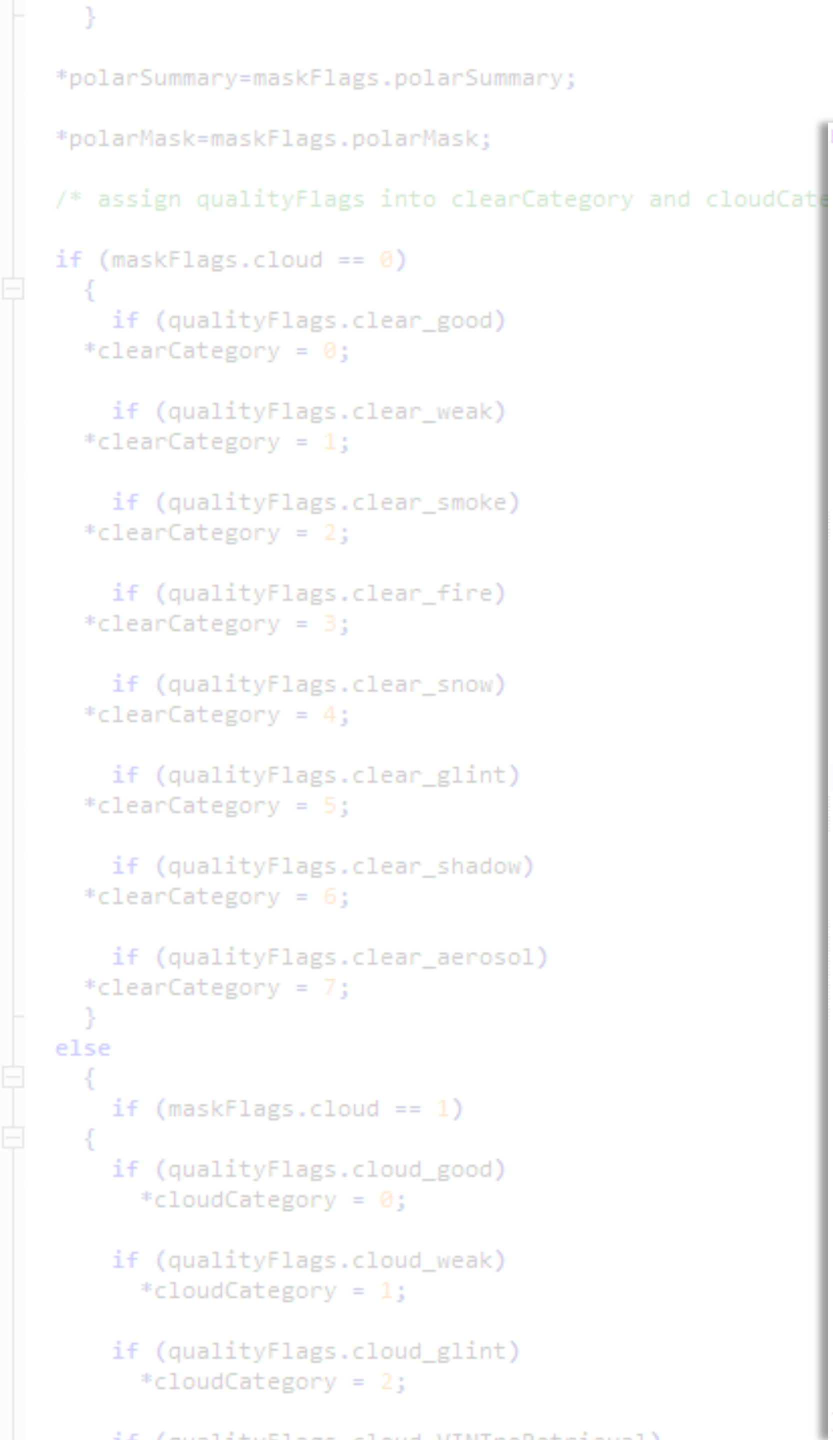
```
bool CloudMaskDaytimeModisAquaOO::processYellowCloudTests() {
  std::ostringstream os;
  os.str("");
  os.clear();

  bool isSuccess      = true;

  try {
    float btm_1080 = PQBT_->getBtempValue(static_cast<int>(BrightnessTemperatureSlot::_1080)); //legacy u

    CMV_->setPolarSummary( static_cast<int>(PolarSummaryCategory::DAY_VIS_PASS_SR_IR_PASS) ); //maskFlags

    /* added ice clouds detecction before snow/ice surface test on 2009/01/12, added ratio21 on 2010/09/2
    if ( diffcs_4_ > dayTimeModisTestParms_->yellowDiffcs_4Conditional
       && ratio21_ < dayTimeModisTestParms_->yellowRatio213_065_corModifier1 * ratio213_065_cor_ ){
      markStrongCloudAttributes(IS_POLAR);
    }//end if (diffcs_4 > 12 && ratio21 < 0.35*ratio213_065_cor)
    else{
      float btm_1330 = PQBT_->getBtempValue(static_cast<int>(BrightnessTemperatureSlot::_1330)); //legacy

      if( (!isSuperColdPlateau_
          && ratio21_ <= (dayTimeModisTestParms_->yellowRatio213_065_corModifier2 * ratio213_065_cor_
          && !isResult37Ref_
          && diff34_ < (diffcs34_ + dayTimeModisTestParms_->yellowDiffCs34Modifier * CS_STD34_))
          || (isSuperColdPlateau_ && diff67_11_ > dayTimeModisTestParms_->yellowDiff67_11Conditional
             && (btm_1330 - btm_1080) > dayTimeModisTestParms_->yellowBtmConditional) ){
        markClearSnowAttributes(IS_POLAR);
      }
      else { /* detect clear land, added Tskin test on 03/31/2003 */
        if( (determineIsClearLand( dayTimeModisTestParms_->yellowClearLandRatio213_065_corModifier
                                 , dayTimeModisTestParms_->yellowClearLandDiffCs34Conditional)) ){
          markClearAttributes( static_cast<int>(PolarMaskCategory::CLEAR_LAND) );
        }
        else{
          markStrongCloudAttributes(IS_POLAR);
        }//end if determineIsClearLand
      }//end else { /* detect clear land, added Tskin test on 03/31/2003 */
    }//end else of if (diffcs_4 > 12 && ratio21 < 0.35 * ratio213_065_cor)

  } catch (std::exception& e) {
    isSuccess = false;
    os << std::endl << std::endl << " * * * CloudMaskDaytimeModisAquaOO::testForYellowClouds()  exception
       << "\n\t" << e.what()
       << "\n\t calling CloudMaskDaytimeModisAquaOO::toString()... "
       << std::endl << std::endl;
    std::cout << os.str();
    toString();
  }

  return isSuccess;
}//end method testForYellowClouds
```



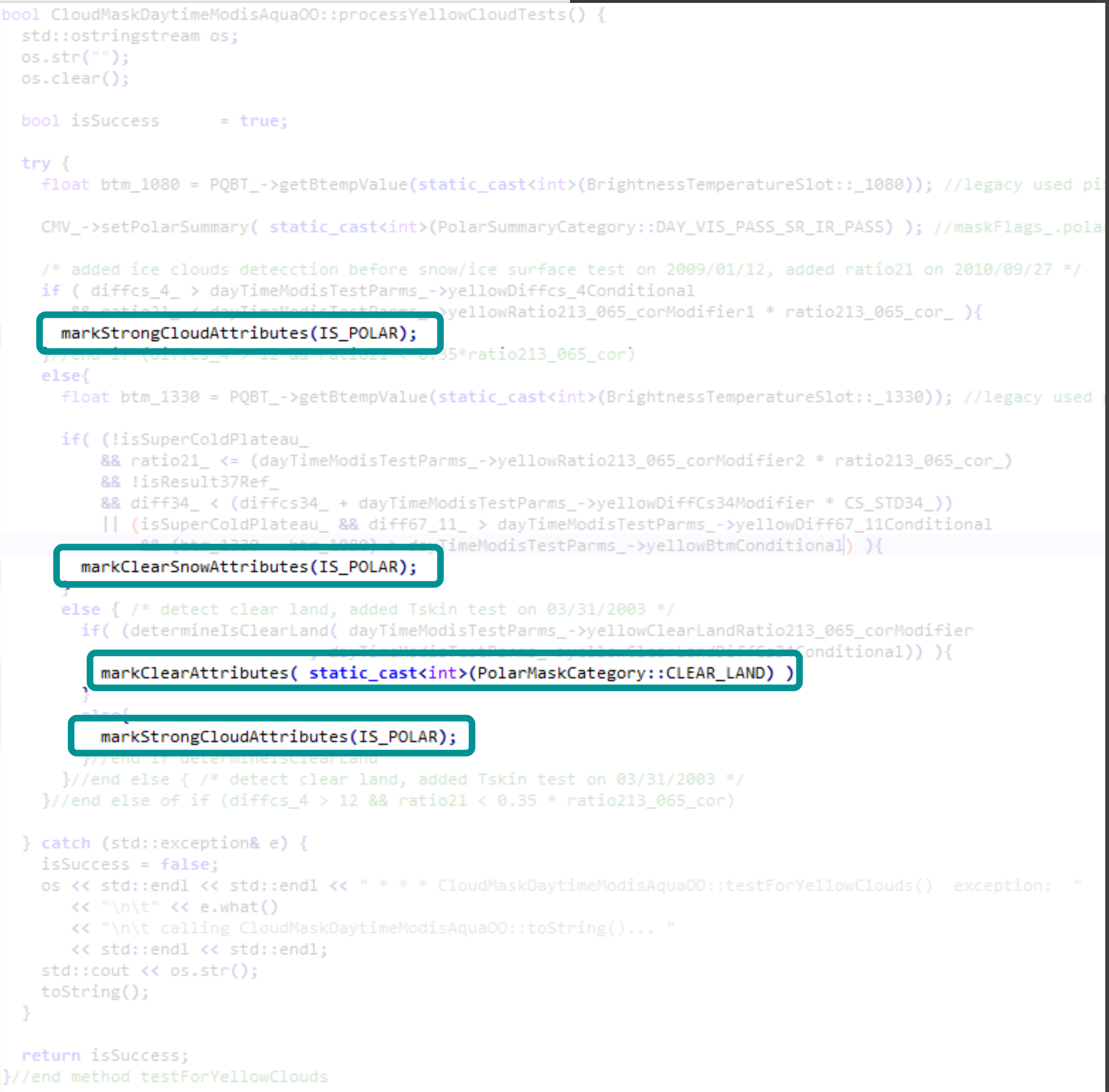
```
    .
    .
    .
/* detect high and cold clouds using T3-T4 test */
if (!superColdPlateau && diff34 >= 18 /* && diffcs_4 > 8 */)
  {
    maskFlags->cloud = 1;
    qualityFlags->cloud_good = 1;
    maskFlags->polarSummary = 0;
    maskFlags->polarMask = 0;

    return;
  }

/* printf ("in daytime aqua poalr,terramodis=%d,wrapandmask=%d, snowref16=%f, ref213_016=%f,snowref21=%f, snowref21_

if ( resultvis213 == 1 && resultSR_IR == 1 )  /* pass both 2.13 and T3-T4 test, clouds, yellow */
  {
    maskFlags->polarSummary = 1;
    /* added ice clouds deteccTion before snow/ice surface test on 2009/01/12, added ratio21 on 2010/09/27 */
    /* if ((IGBP == 17 && pixel->IceMap > 0 && pixel->IceMap <= 100 && diffcs_4 > 7.5) || diffcs_4 > 10) */
    if (diffcs_4 > 12 && ratio21 < 0.35*ratio213_065_cor)
      {
        maskFlags->cloud = 1;
        qualityFlags->cloud_good = 1;
        maskFlags->polarMask = 0;
      }
    else
      {
    /* printf ("in Daytime Aqua Polar, SIR=1, NIR=1, ratio213_065_cor=%f, 0.25*ratio213_065_cor=%f, ratio21=%f, diff3
        if ((!superColdPlateau && ratio21 <= 0.3*ratio213_065_cor && result37Ref == 0 && diff34 < (diffcs34 + 1.5*CS_
          {
        maskFlags->cloud = 0;
            qualityFlags->clear_snow = 1;
            maskFlags->polarMask = 3;
          }
        else
          { /* detect clear land, added Tskin test on 03/31/2003 */
            if (IGBP != 17 && ratio21 > 1.3*ratio213_065_cor && pixel->reflec[REF0063] < 0.30 && diff34 < (diffcs34 +
              {
                maskFlags->cloud = 0;
                qualityFlags->clear_good = 1;
                maskFlags->polarMask = 2;
              }
            else
              {
        maskFlags->cloud = 1;
                qualityFlags->cloud_good = 1;
                maskFlags->polarMask = 0;
              }
          }
      }
  }
    .
    .
    .
```



```
bool CloudMaskDaytimeModisAquaOO::processYellowCloudTests() {
  std::ostringstream os;
  os.str("");
  os.clear();

  bool isSuccess      = true;

  try {
    float btm_1080 = PQBT_->getBtempValue(static_cast<int>(BrightnessTemperatureSlot::_1080)); //legacy u

    CMV_->setPolarSummary( static_cast<int>(PolarSummaryCategory::DAY_VIS_PASS_SR_IR_PASS) ); //maskFlags

    /* added ice clouds deteccTion before snow/ice surface test on 2009/01/12, added ratio21 on 2010/09/2
    if ( diffcs_4_ > dayTimeModisTestParms_->yellowDiffcs_4Conditional
      && ratio21_ < dayTimeModisTestParms_->yellowRatio213_065_corModifier1 * ratio213_065_cor_ ){
      markStrongCloudAttributes(IS_POLAR);
    }//end if (diffcs_4 > 12 && ratio21 < 0.35*ratio213_065_cor)
    else{
      float btm_1330 = PQBT_->getBtempValue(static_cast<int>(BrightnessTemperatureSlot::_1330)); //legacy

      if( (!isSuperColdPlateau_
          && ratio21_ <= (dayTimeModisTestParms_->yellowRatio213_065_corModifier2 * ratio213_065_cor_)
          && !isResult37Ref_
          && diff34_ < (diffcs34_ + dayTimeModisTestParms_->yellowDiffCs34Modifier * CS_STD34_))
          || (isSuperColdPlateau_ && diff67_11_ > dayTimeModisTestParms_->yellowDiff67_11Conditional
          && (btm_1330 - btm_1080) > dayTimeModisTestParms_->yellowBtmConditional) ){
        markClearSnowAttributes(IS_POLAR);
      }
      else { /* detect clear land, added Tskin test on 03/31/2003 */
        if( (determineIsClearLand( dayTimeModisTestParms_->yellowClearLandRatio213_065_corModifier
                                 , dayTimeModisTestParms_->yellowClearLandDiffCs34Conditional)) ){
          markClearAttributes( static_cast<int>(PolarMaskCategory::CLEAR_LAND) )
        }
        else{
          markStrongCloudAttributes(IS_POLAR);
        }//end if determineIsClearLand
      }//end else { /* detect clear land, added Tskin test on 03/31/2003 */
    }//end else of if (diffcs_4 > 12 && ratio21 < 0.35 * ratio213_065_cor)

  } catch (std::exception& e) {
    isSuccess = false;
    os << std::endl << std::endl << " * * * CloudMaskDaytimeModisAquaOO::testForYellowClouds()  exception
       << "\n\t" << e.what()
       << "\n\t calling CloudMaskDaytimeModisAquaOO::toString()... "
       << std::endl << std::endl;
    std::cout << os.str();
    toString();
  }

  return isSuccess;
}//end method testForYellowClouds
```



```
    }

  *polarSummary=maskFlags.polarSummary;

  *polarMask=maskFlags.polarMask;

  /* assign qualityFlags into clearCategory and cloudCate

  if (maskFlags.cloud == 0)
   {
     if (qualityFlags.clear_good)
   *clearCategory = 0;

     if (qualityFlags.clear_weak)
   *clearCategory = 1;

     if (qualityFlags.clear_smoke)
   *clearCategory = 2;

     if (qualityFlags.clear_fire)
   *clearCategory = 3;

     if (qualityFlags.clear_snow)
   *clearCategory = 4;

     if (qualityFlags.clear_glint)
   *clearCategory = 5;

     if (qualityFlags.clear_shadow)
   *clearCategory = 6;

     if (qualityFlags.clear_aerosol)
   *clearCategory = 7;
   }
  else
   {
     if (maskFlags.cloud == 1)
   {
     if (qualityFlags.cloud_good)
       *cloudCategory = 0;

     if (qualityFlags.cloud_weak)
       *cloudCategory = 1;

     if (qualityFlags.cloud_glint)
       *cloudCategory = 2;

     if (qualityFlags.cloud_VIIRSnoRetrieval)
```



```
bool CloudMaskDaytimeModisAquaOO::processYellowCloudTests() {
  std::ostringstream os;
  os.str("");
  os.clear();

  bool isSuccess      = true;

  try {
    float btm_1080 = PQBT_->getBtempValue(static_cast<int>(BrightnessTemperatureSlot::_1080)); //legacy used pi

    CMV_->setPolarSummary( static_cast<int>(PolarSummaryCategory::DAY_VIS_PASS_SR_IR_PASS) ); //maskFlags_.pola

    /* added ice clouds detecction before snow/ice surface test on 2009/01/12, added ratio21 on 2010/09/27 */
    if ( diffcs_4_ > dayTimeModisTestParms_->yellowDiffcs_4Conditional
      && ratio21_ < dayTimeModisTestParms_->yellowRatio213_065_corModifier1 * ratio213_065_cor_ ){
     markStrongCloudAttributes(IS_POLAR);
    }//end if (diffcs_4 > 12 && ratio21 < 0.35*ratio213_065_cor)
    else{
      float btm_1330 = PQBT_->getBtempValue(static_cast<int>(BrightnessTemperatureSlot::_1330)); //legacy used

      if( (!isSuperColdPlateau_
          && ratio21_ <= (dayTimeModisTestParms_->yellowRatio213_065_corModifier2 * ratio213_065_cor_)
          && !isResult37Ref_
          && diff34_ < (diffcs34_ + dayTimeModisTestParms_->yellowDiffCs34Modifier * CS_STD34_))
          || (isSuperColdPlateau_ && diff67_11_ > dayTimeModisTestParms_->yellowDiff67_11Conditional
            && (btm_1330 - btm_1080) > dayTimeModisTestParms_->yellowBtmConditional) ){
        markClearSnowAttributes(IS_POLAR);
      }
      else { /* detect clear land, added Tskin test on 03/31/2003 */
        if( (determineIsClearLand( dayTimeModisTestParms_->yellowClearLandRatio213_065_corModifier
                                 , dayTimeModisTestParms_->yellowClearLandDiffCs34Conditional)) ){
          markClearAttributes( static_cast<int>(PolarMaskCategory::CLEAR_LAND) )
        }
        else{
          markStrongCloudAttributes(IS_POLAR);
        }//end if determineIsClearLand
      }//end else { /* detect clear land, added Tskin test on 03/31/2003 */
    }//end else of if (diffcs_4 > 12 && ratio21 < 0.35 * ratio213_065_cor)

  } catch (std::exception& e) {
    isSuccess = false;
    os << std::endl << std::endl << " * * * CloudMaskDaytimeModisAquaOO::testForYellowClouds()  exception:  "
       << "\n\t" << e.what()
       << "\n\t calling CloudMaskDaytimeModisAquaOO::toString()... "
       << std::endl << std::endl;
    std::cout << os.str();
    toString();
  }

  return isSuccess;
}//end method testForYellowClouds
```

TISA

TISA

Longwave (LW) Flux Computation

Refactoring nb2bb (lw and wn) radiance-to-flux calculations:

- Completed LW and WN
- Standardized netCDF4 files:
  - LUT coefficients
  - Bin bounds and centers

Common

Generic subroutines for:

- Bin index calculations
- Cross-bin interpolations

Shortwave (SW) Flux Computation

Clear/cloudy land, ocean snow scene type calculations

Standardized netCDF4 files:

- ADM LUT coefficients
- Bin bounds and centers

TISA

- xGLB coefficients:
  - Converted to netCDF4 from binary (LW, SW, WN)
  - Developed Python plotting package to visualize statistics

- Redesigned code framework for Edition 5
  - Decoupled code bases: data preparation and algorithm code.
  - SW ADM interpolation code: reduced cyclomatic complexity
  - Unraveled interdependencies among scene type inputs

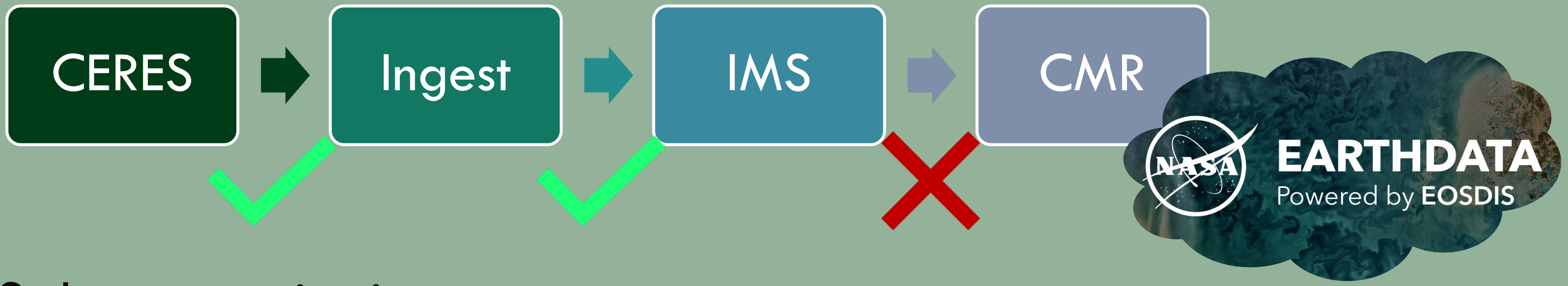
- Also decoupled: Edition 4 LUT access from algorithm code bases
  - Narrowband-to-broadband
  - TRMM ADM

- Externally callable:
  - SW ADM Scene Type identification
  - IGBP map access functions

EXTERNAL INTERFACES

CMR METADATA PROBLEM

- Common Metadata Repository (CMR) for ESDS not getting collection metadata

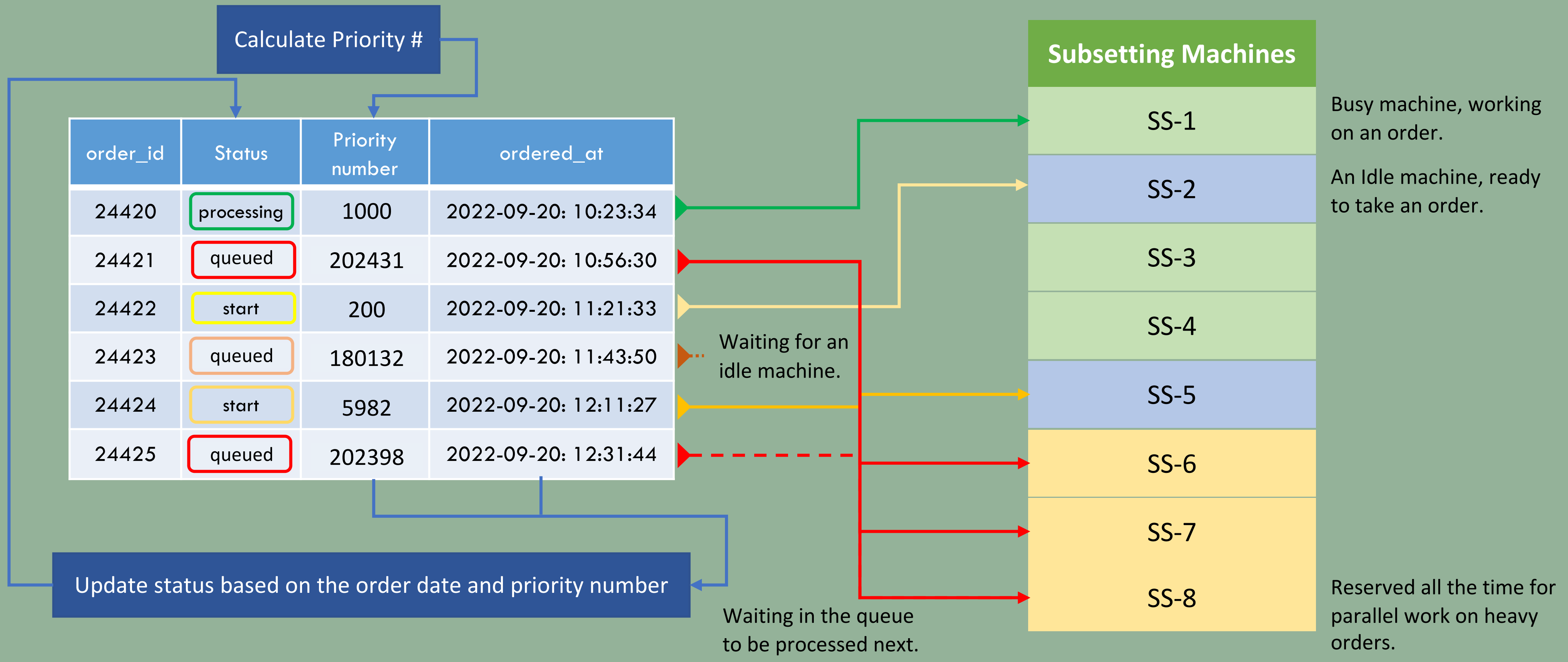


CERES → Ingest → IMS → CMR

✓ ✓ ✗

- Affects ESDS data repositories
- Dark Horse (ingest/archive) at ASDC – stricter metadata standards
- Affected data: CERES – January 2022; FLASHFlux –June 2022

CERES PUBLIC ORDERING TOOL



Calculate Priority #

order_id	Status	Priority number	ordered_at
24420	processing	1000	2022-09-20: 10:23:34
24421	queued	202431	2022-09-20: 10:56:30
24422	start	200	2022-09-20: 11:21:33
24423	queued	180132	2022-09-20: 11:43:50
24424	start	5982	2022-09-20: 12:11:27
24425	queued	202398	2022-09-20: 12:31:44

Update status based on the order date and priority number

Waiting for an idle machine.

Waiting in the queue to be processed next.

Subsetting Machines
SS-1
SS-2
SS-3
SS-4
SS-5
SS-6
SS-7
SS-8

Busy machine, working on an order.

An Idle machine, ready to take an order.

Reserved all the time for parallel work on heavy orders.

LASP-RBSP
LIBERA DATA MANAGEMENT WORKING GROUP

- Bi-weekly meetings
  - L1B data product(s) processing strategy
  - Code curation

- First in-person meeting: September 8-9, 2022 at NASA Langley Research Center

- RBSP given access to LASP Bitbucket repository

- Atmospheric Science Data Center (ASDC) beginning work on ICDs:
  - LASP-ASDC
  - RBSP-ASDC