ITK Video

ITK Extensions for Video Processing

Amitha Perera, Patrick Reynolds, Matt Leotta,
Luis Ibanez, Gabe Hart

Real-Time Image Capture for ITK through a Video Grabber

Kevin Cleary

Methods in Medical Image Analysis: An ITK-Based Course with Deliverable Algorithms that extends and evaluates ITK while broadening its developer base

John Galeotti

Goal

- Make video a first-class citizen of ITK
  - Video I/O
- Allow ITK to be used with Computer Vision libraries (OpenCV, vxl, etc.)
- Allow other libraries to be used with ITK
  - Implement ITK filters using other libraries
- Ensure framework allows real-world video problems to be *researched*
  - E.g. don't try solve hard real-time, but make sure real-time issues can be solved

Assumptions and Choices

- Time is not another dimension
  - Video != 3-d pixel volume
  - Unbounded (live cameras)
  - Memory (2 hrs 1920x1080 color video = 44 GB)
  - On-line processing (streaming applications)
    - We haven't invented time travel yet
- No explicit tagging of color spaces and other metadata (except for time)
  - Video readers will produce grayscale or RGB
  - Similar to CT and Hounsfield

Status: Real-time capture

- Using avacap for frame grabbing (GPL)
  - Created an ITK version which is not GPL
- Ring buffer implemented (duplicated)

Status: Video Extensions

- 4-hr tutorial at CVPR
  - Presented: ITK; using ITK with OpenCV; using OpenCV with ITK; direct video support in ITK
  - ~33 people stayed for whole session
  - Raffle ticket histogram:
    - 1: 22
    - 2: 6
    - 3: 15
    - 4: 5
    - 5: 7
    - 6: 4
    - 7: 2

Video in ITK

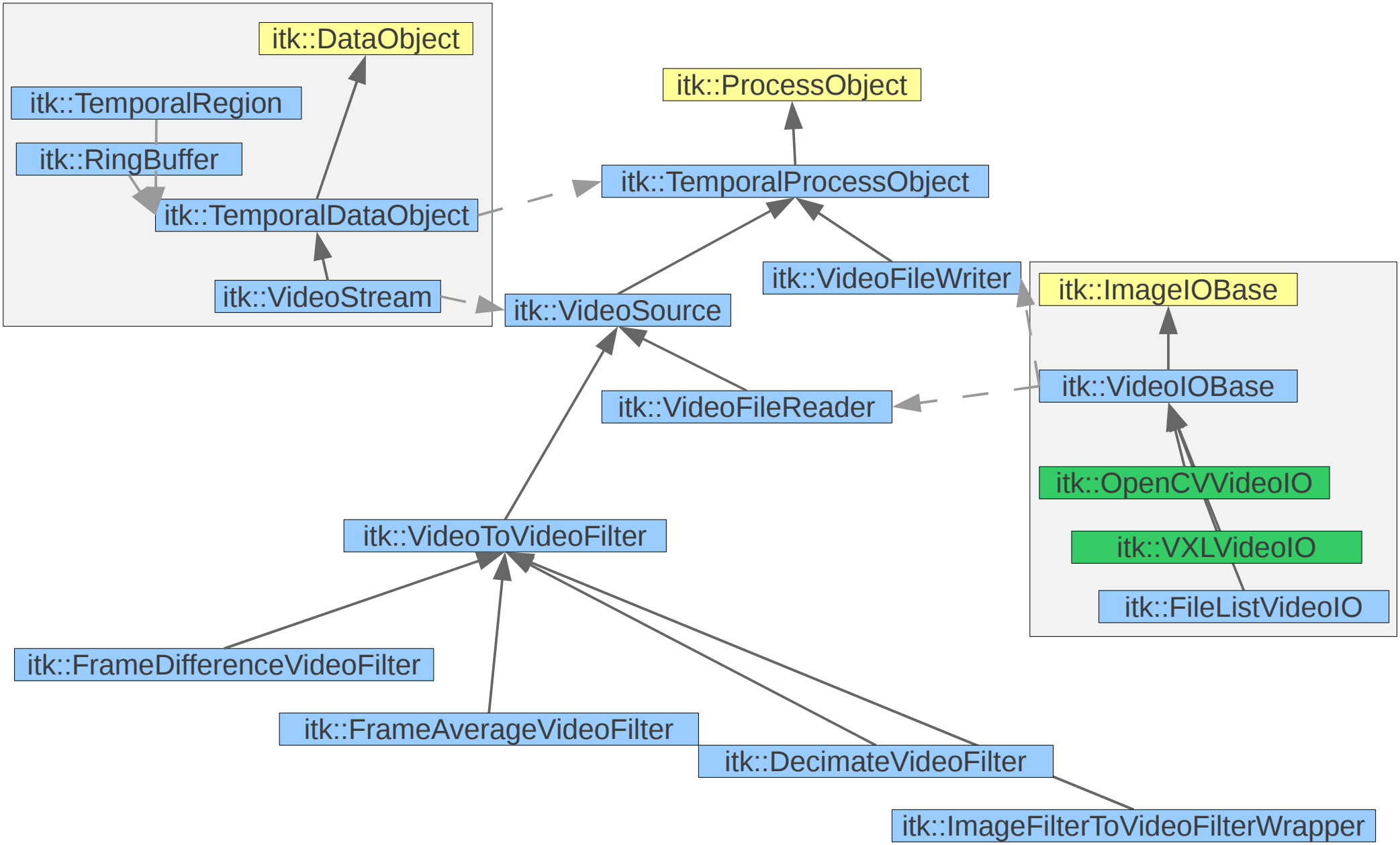
```
reader->SetFileName( argv[1] ); // video reader
writer->SetFileName( argv[2] ); // video writer
frameDifferenceFilter->SetFrameOffset(1);
videoCaster->SetImageFilter( imageCaster );
imageThresh->ThresholdBelow( 128 );
videoThresh->SetImageFilter( imageThresh );
imageCurvatureFlowFilter->SetTimeStep( 0.5 );
imageCurvatureFlowFilter->SetNumberOfIterations( 20 );
videoCurvatureFlowFilter->SetImageFilter( imageFilter );

videoCurvatureFlowFilter->SetInput( reader->GetOutput() );
videoCaster->SetInput( videoFilter->GetOutput() );
frameDifferenceFilter->SetInput( videoCaster->GetOutput() );
videoThresh->SetInput( frameDifferenceFilter->GetOutput() );
writer->SetInput( videoThresh->GetOutput() );

writer->Update(); // generate output video frame by frame, without reading whole source video at once
```

New Classes

New classes

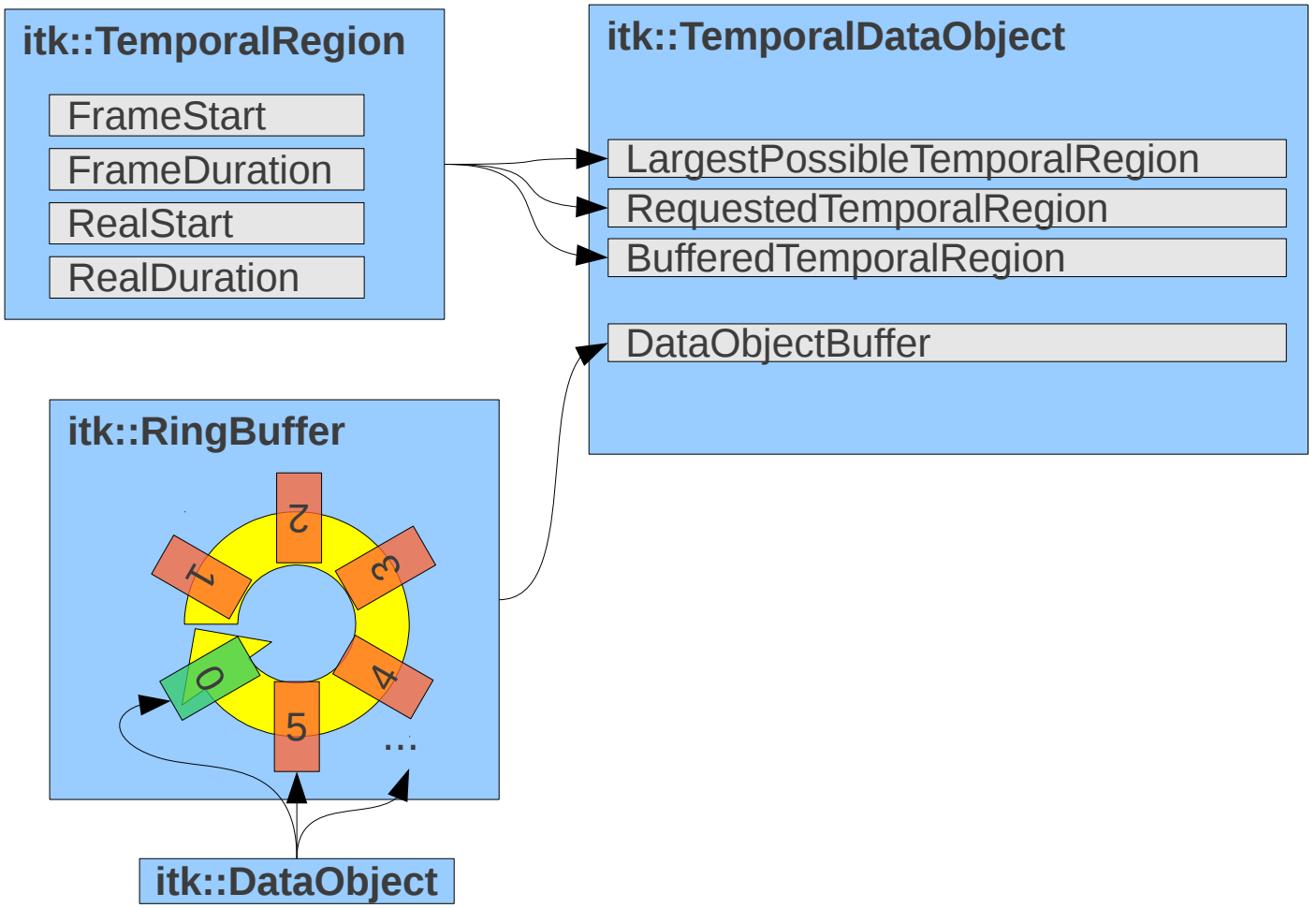
- Real-time capture:
  - A new video I/O classes wrapping a capture library
- Course:
  - ITK video filters
  - ITK representations of non-pixel data (tracks, etc.)

Class Hierarchy

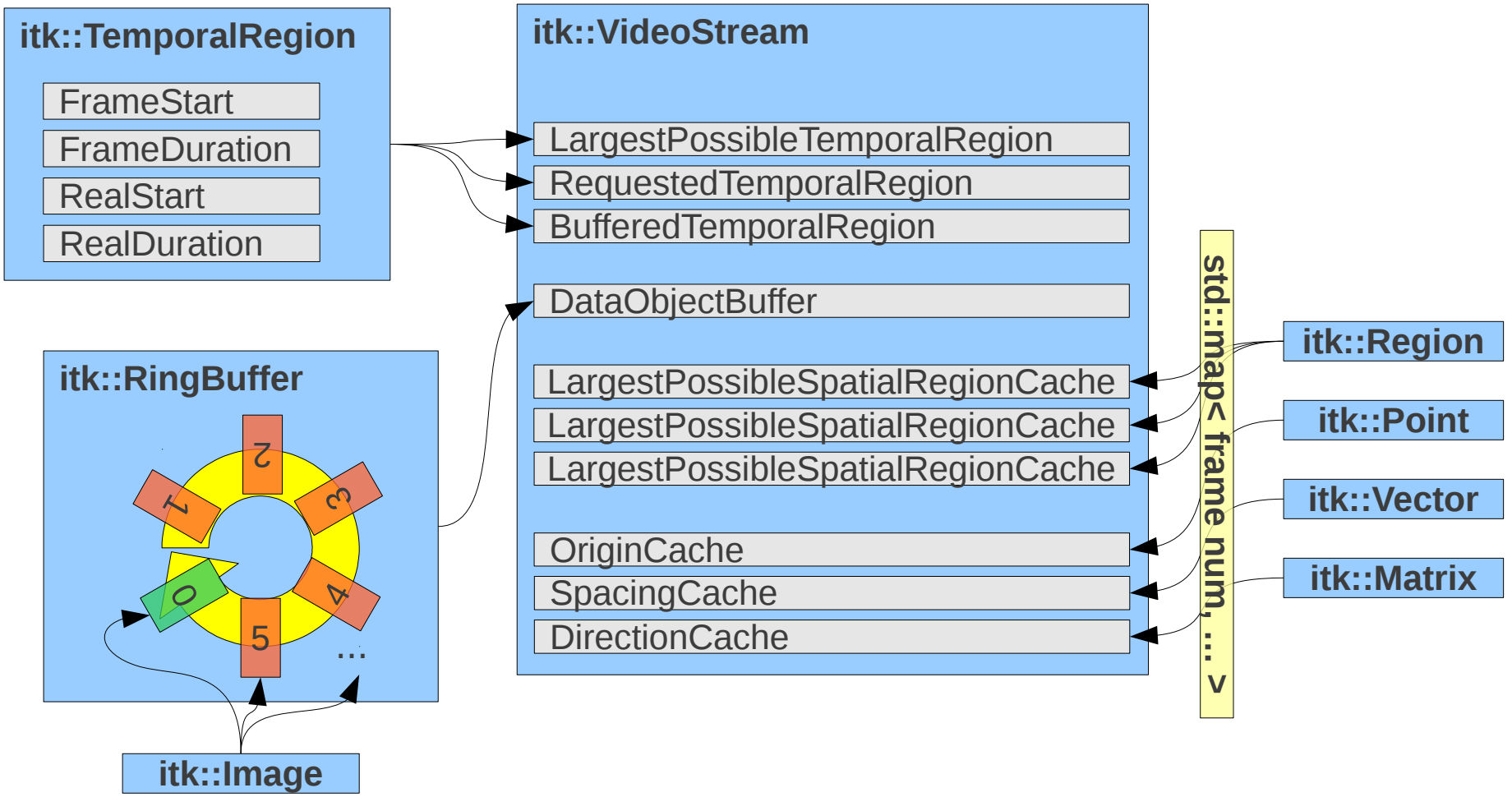


Temporal Data Objects

itk::TemporalDataObject

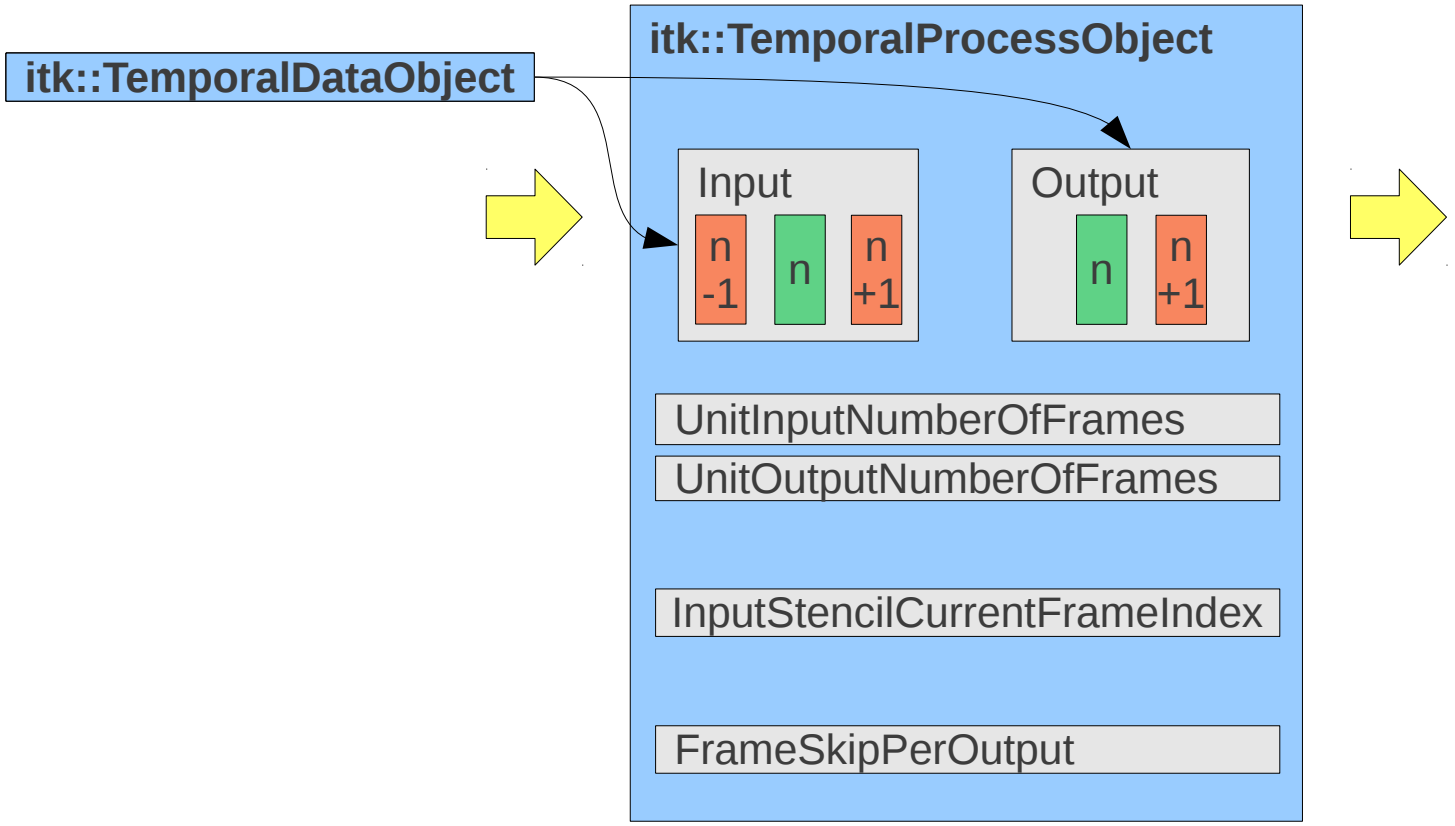


itk::VideoStream

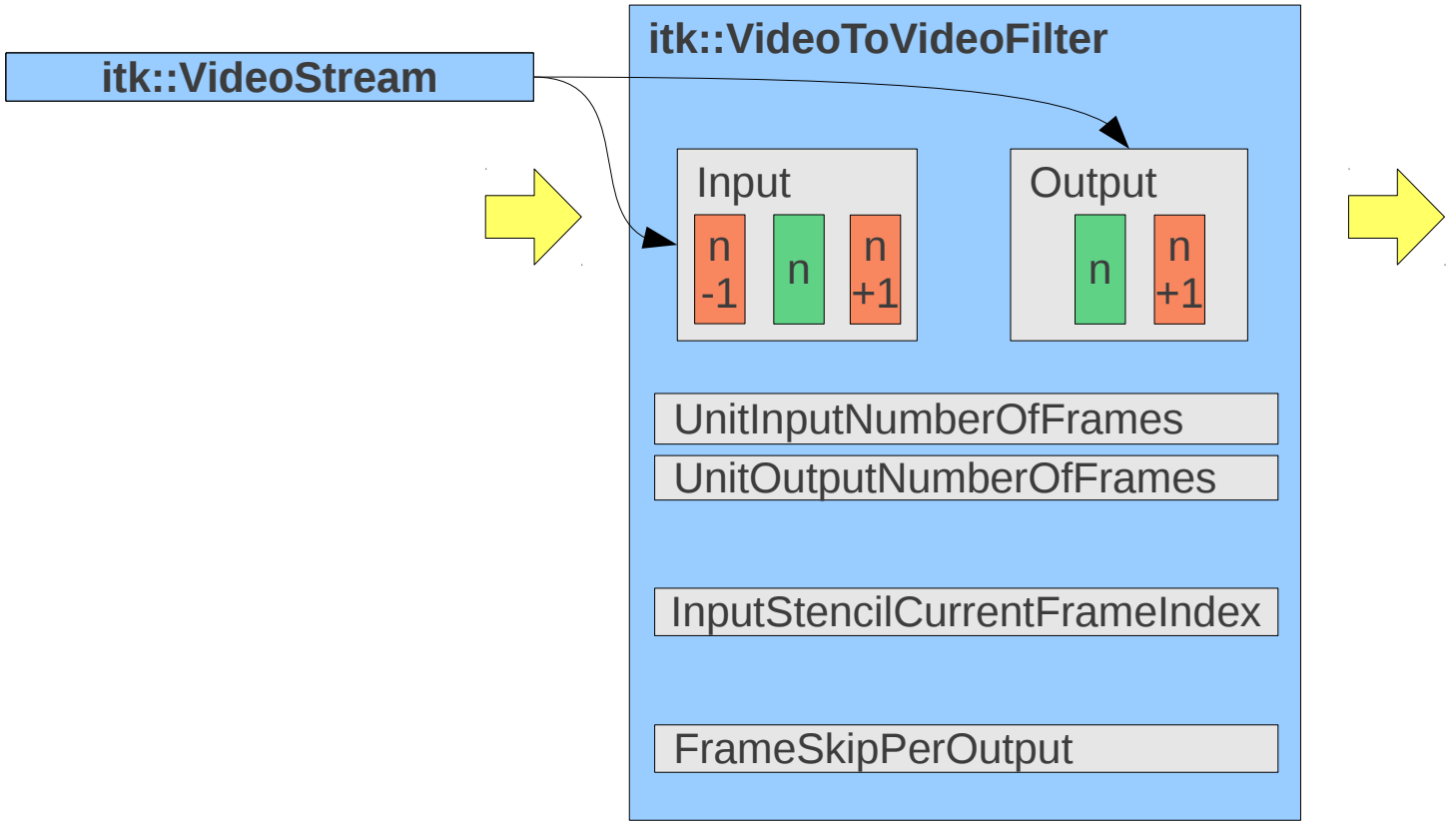


Temporal Process Objects

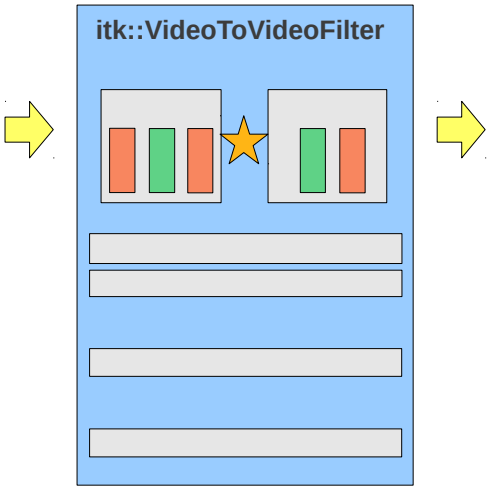
itk::TemporalProcessObject



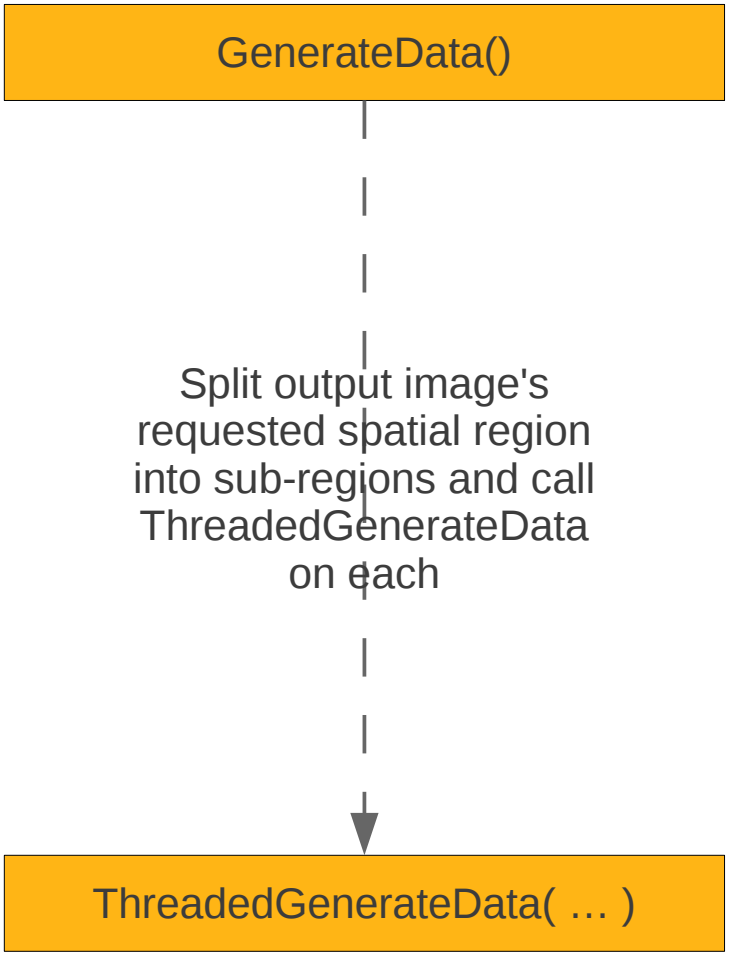
itk::VideoToVideoFilter



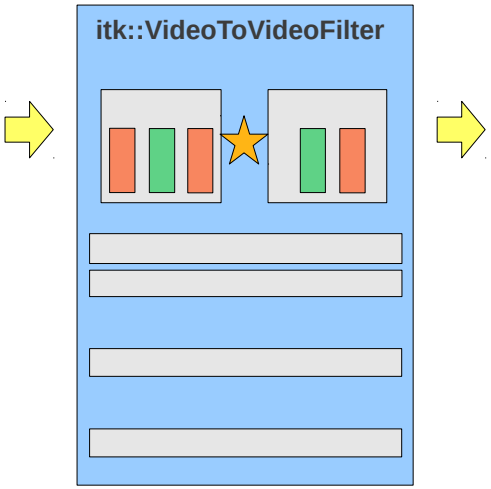
itk::ImageToImageFilter



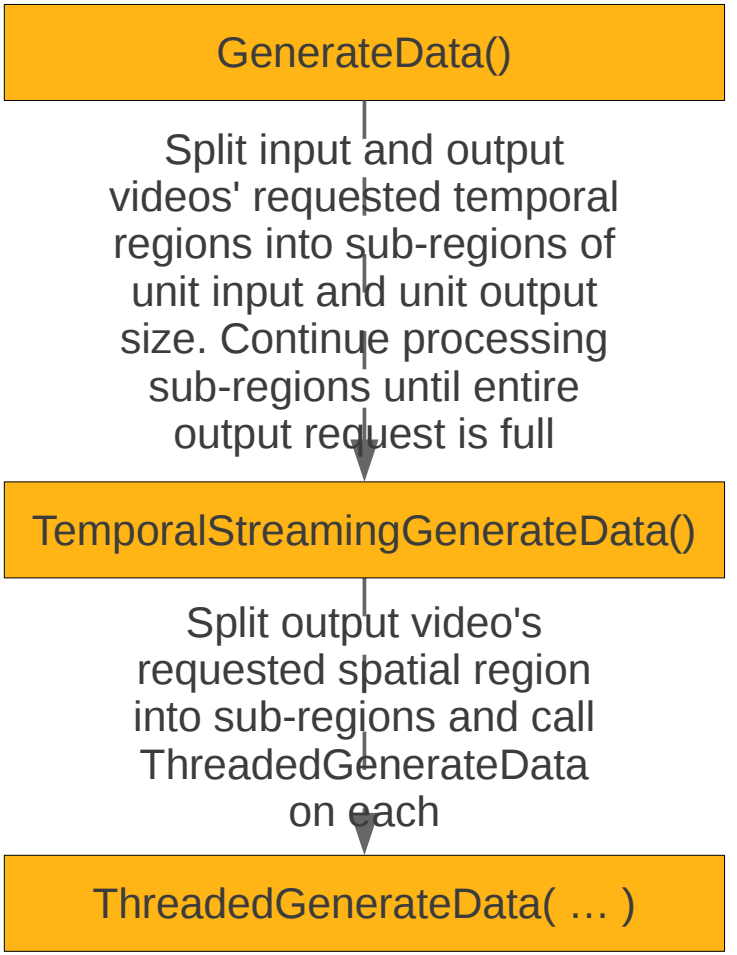
itk::ImageToImageFilter → **GenerateData** Process



itk::VideoToVideoFilter



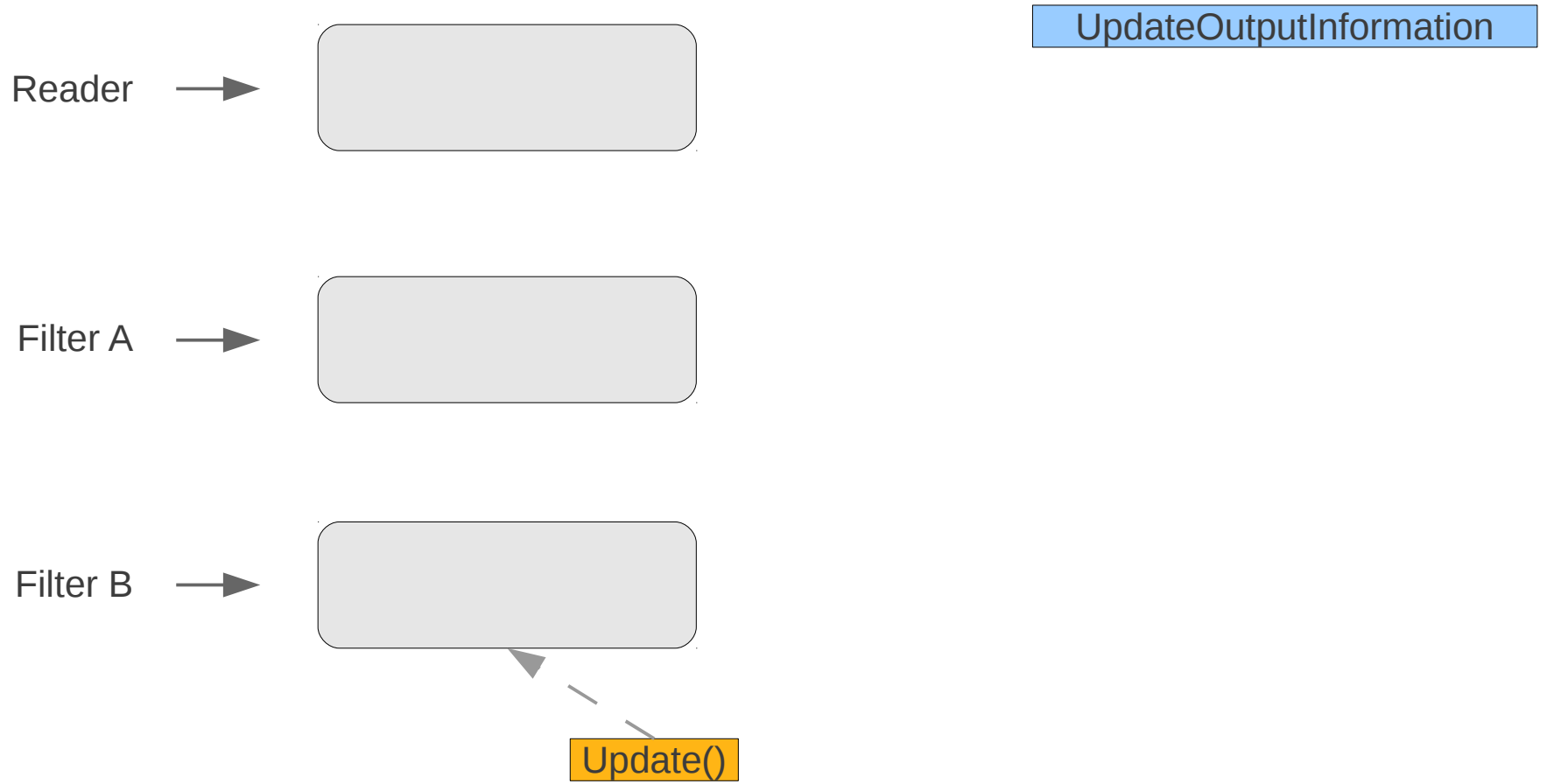
itk::VideoToVideoFilter → **GenerateData** Process



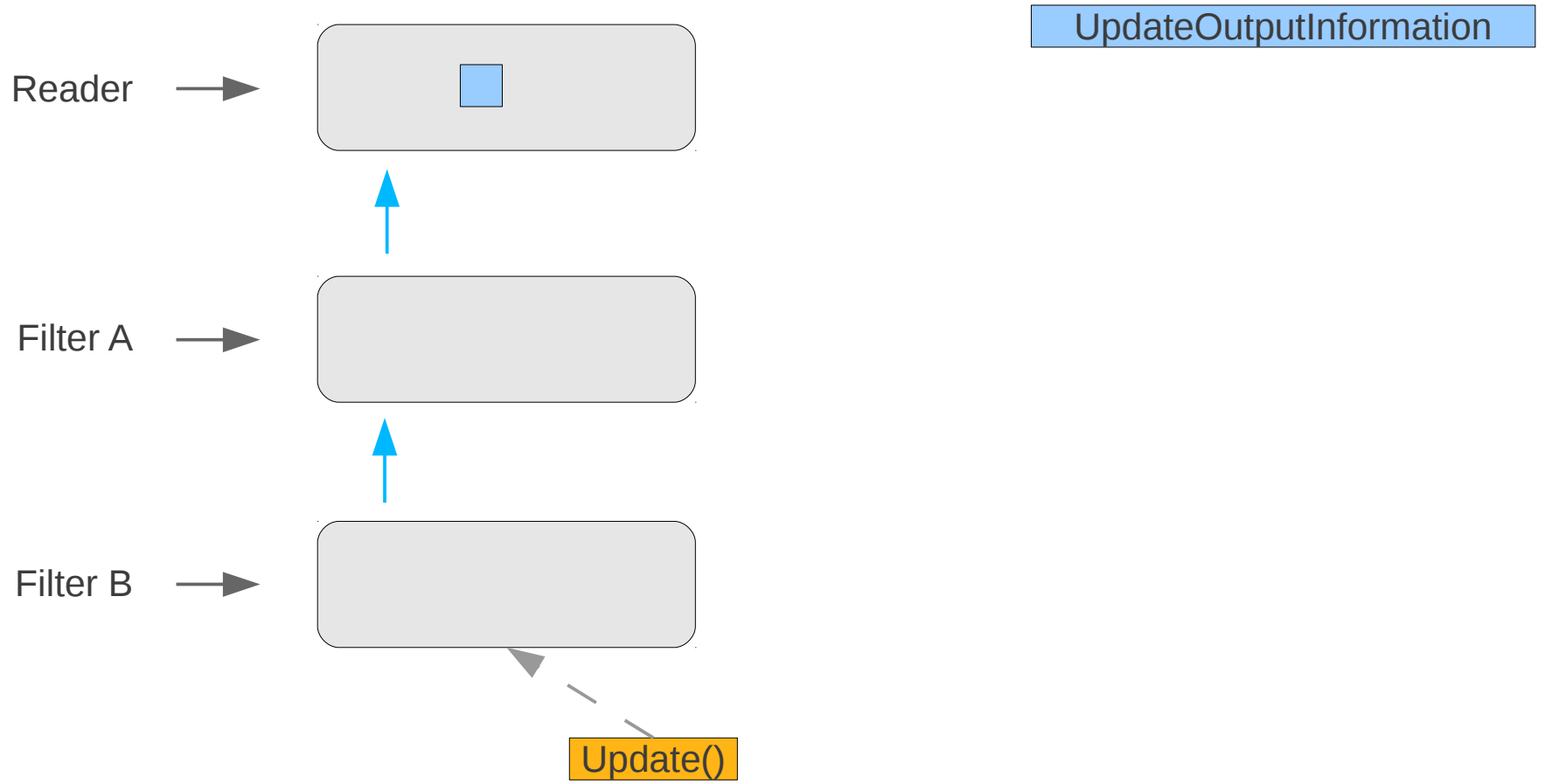
Pipeline Architecture

ITK Pipeline

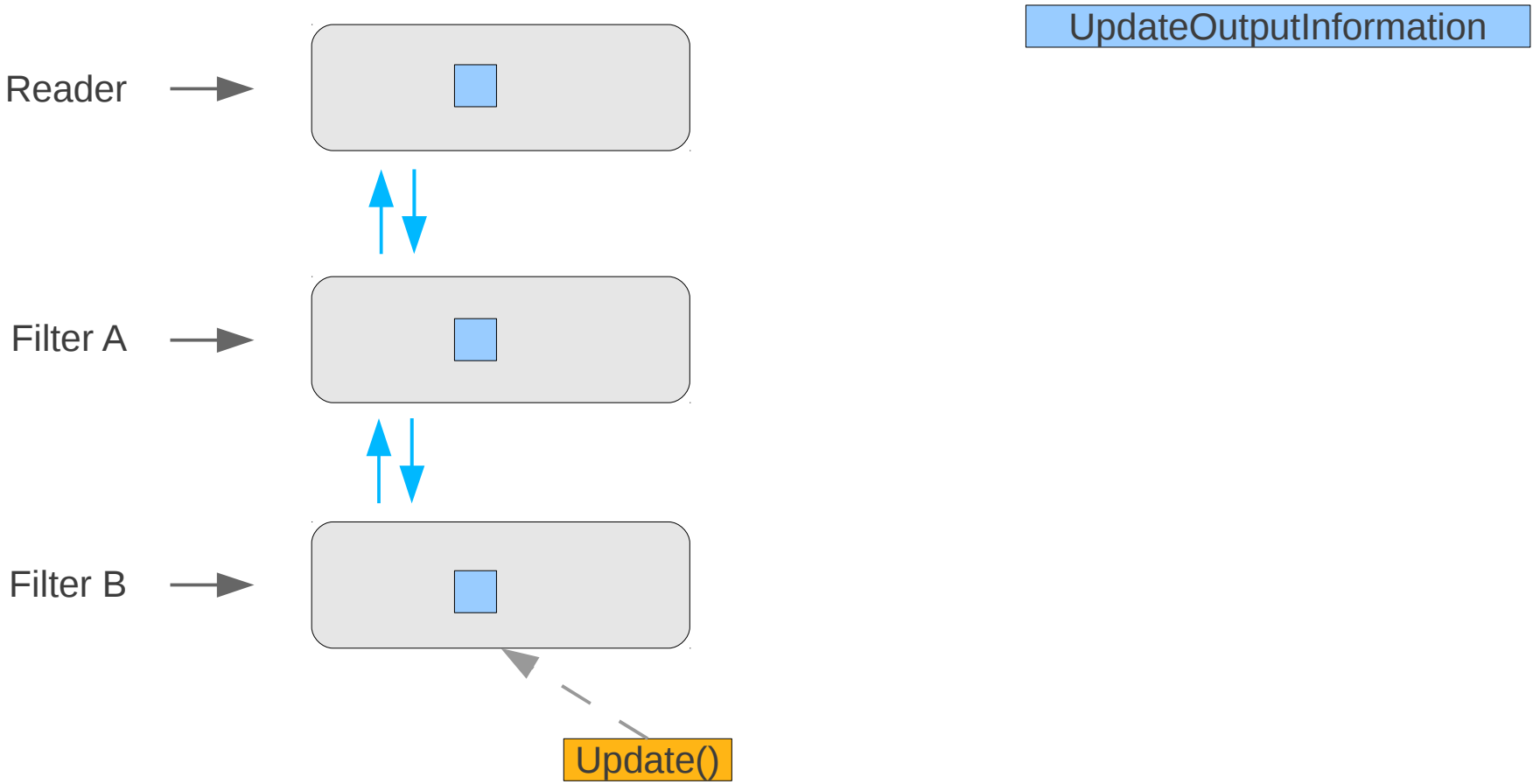
ITK Pipeline



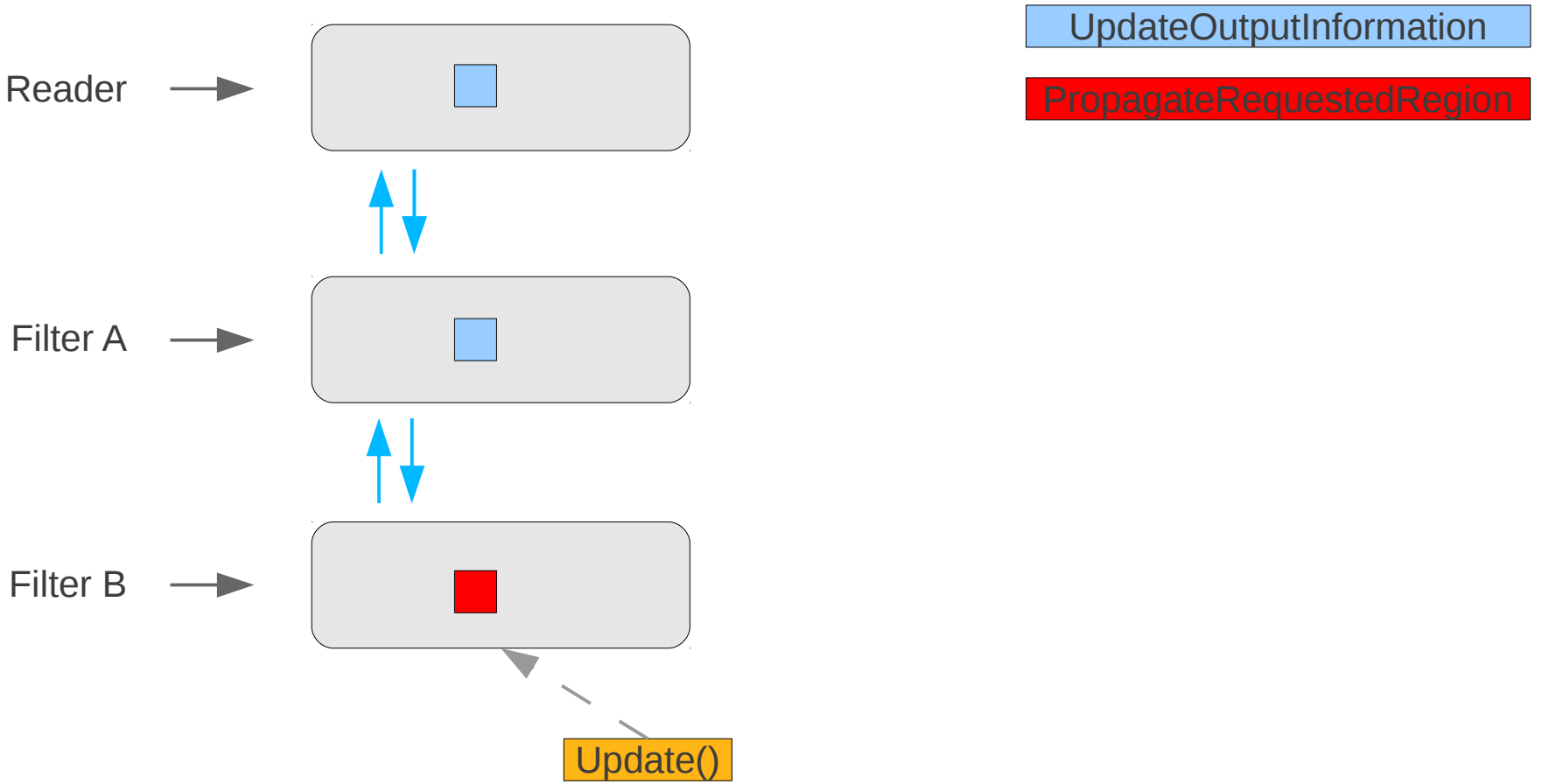
ITK Pipeline



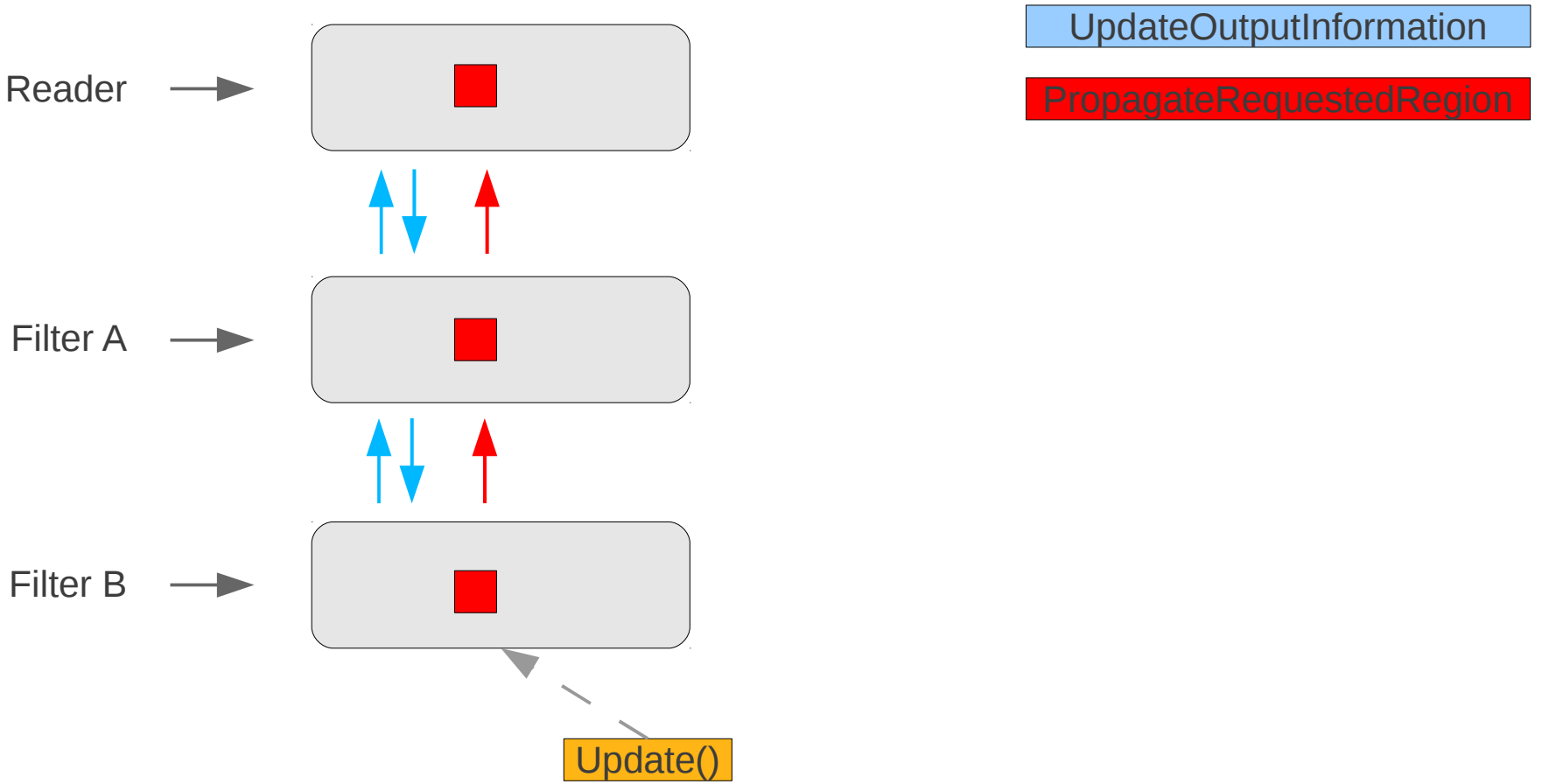
ITK Pipeline



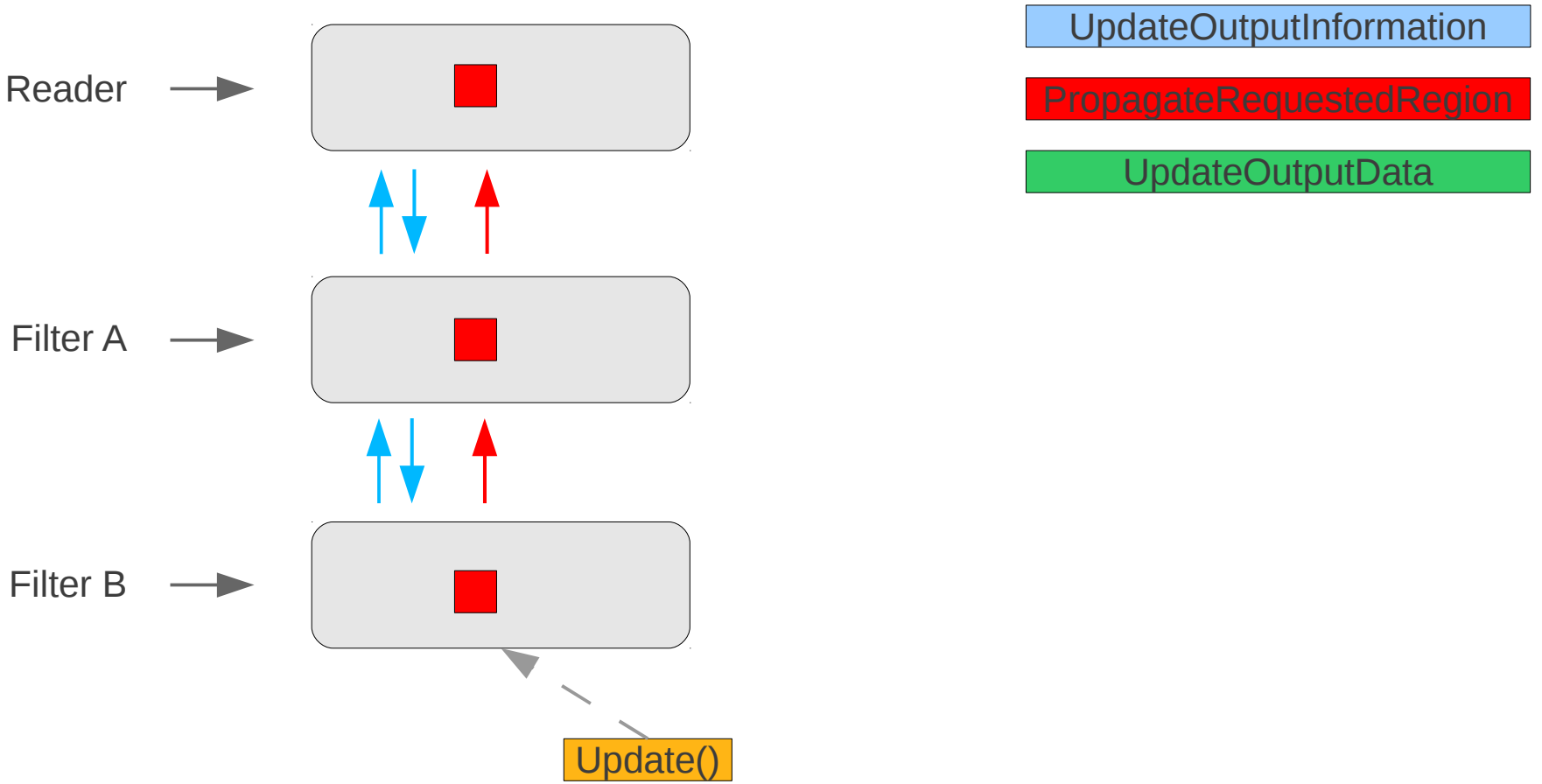
ITK Pipeline



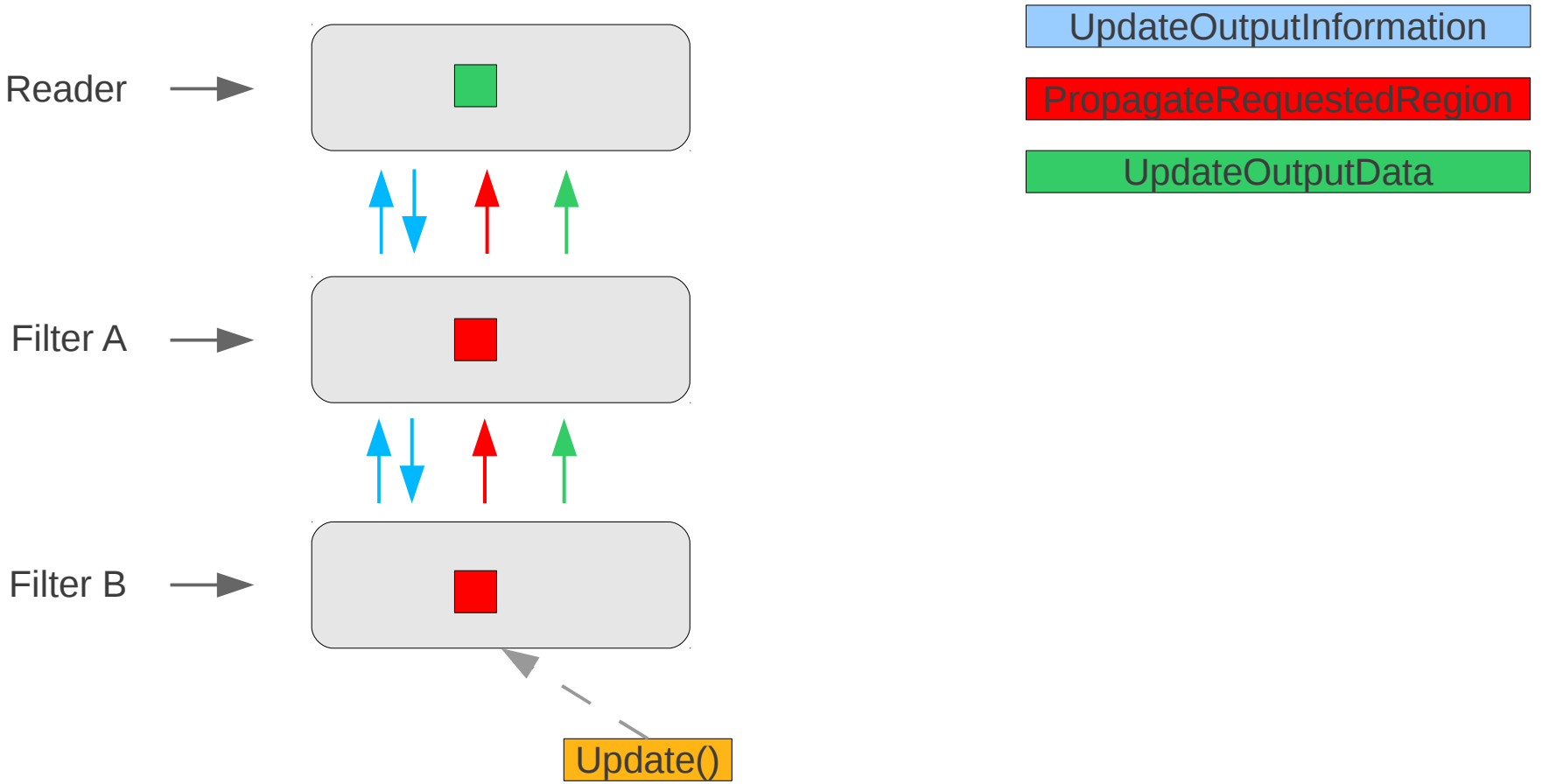
ITK Pipeline



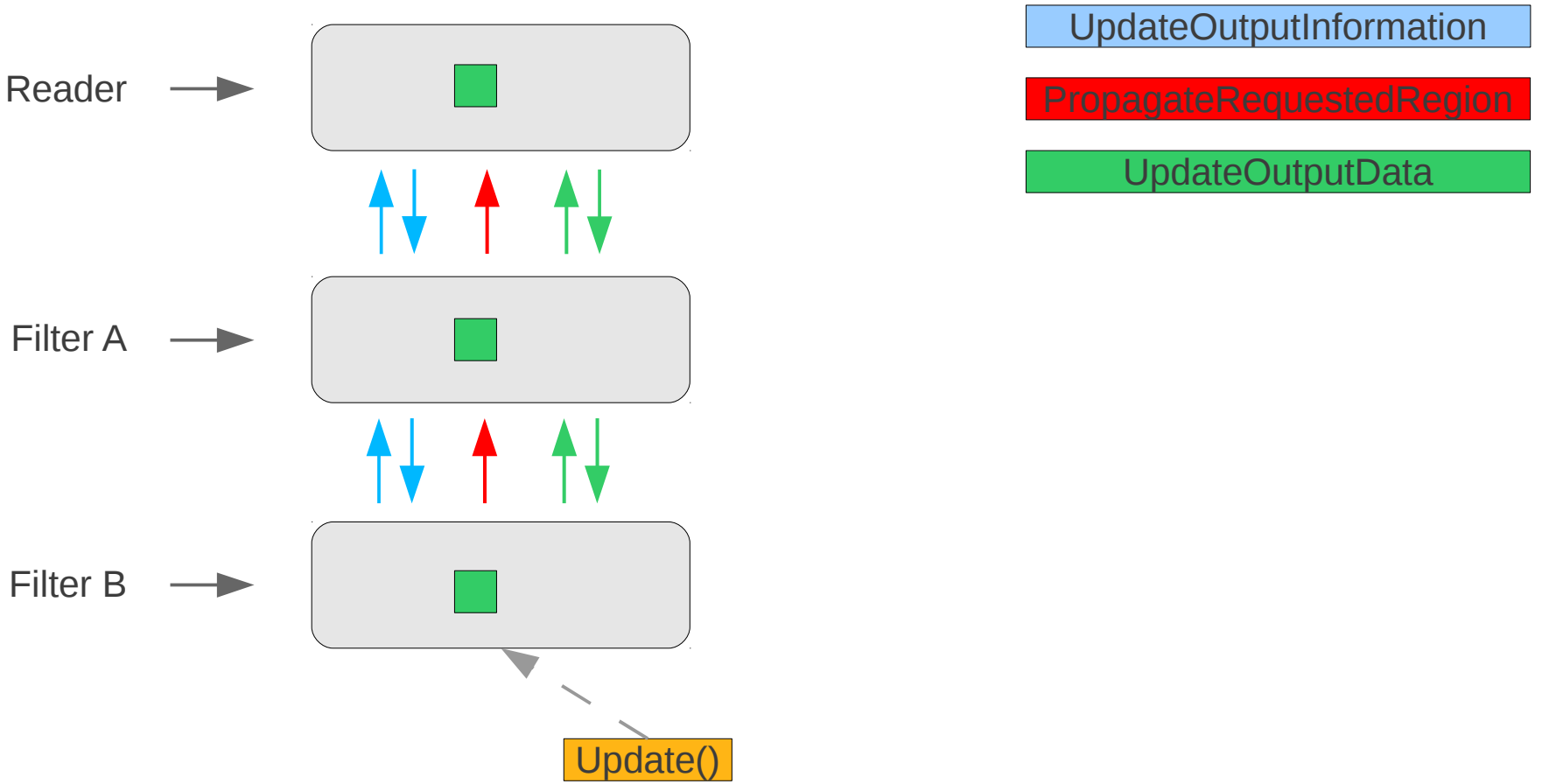
ITK Pipeline



ITK Pipeline

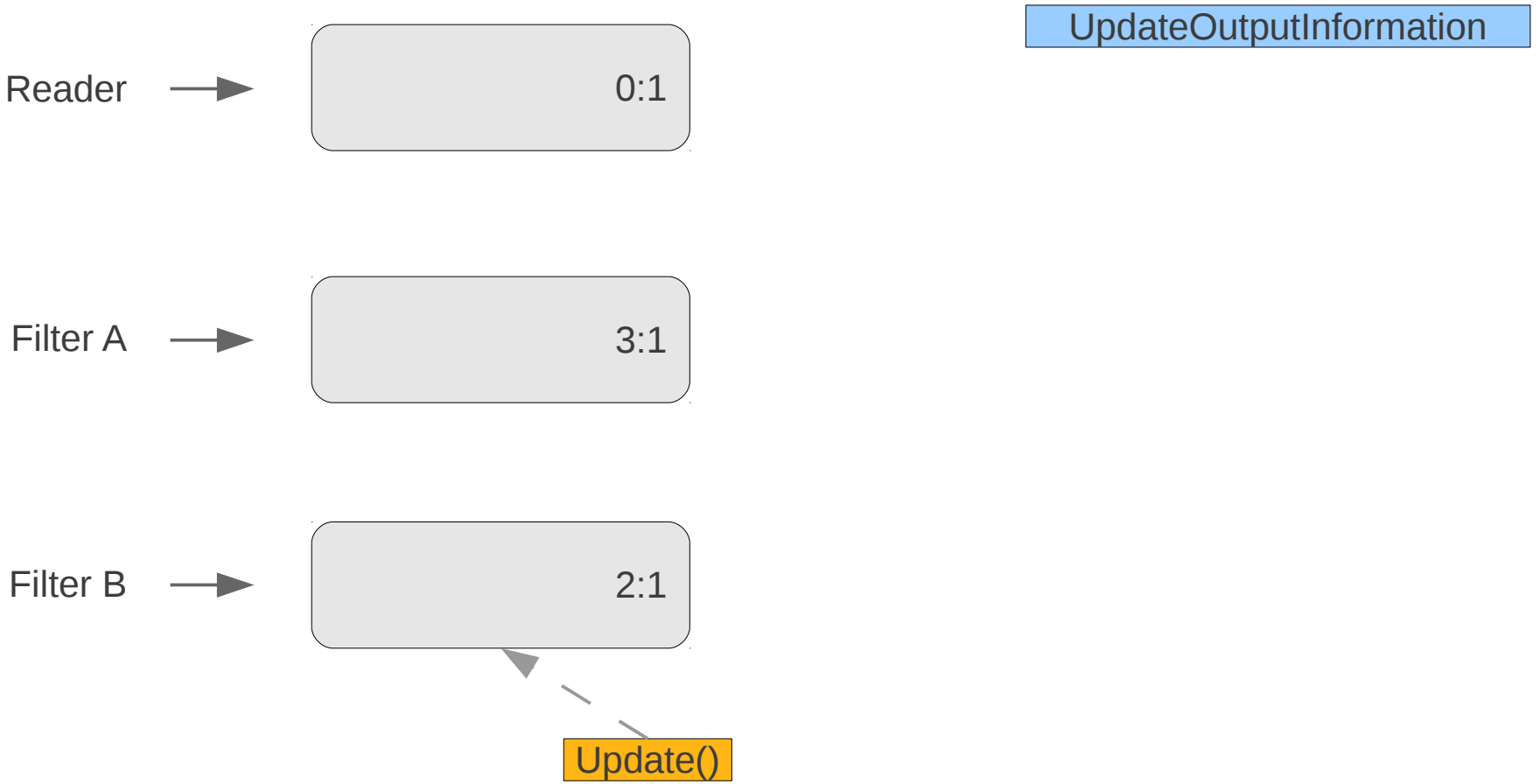


ITK Pipeline

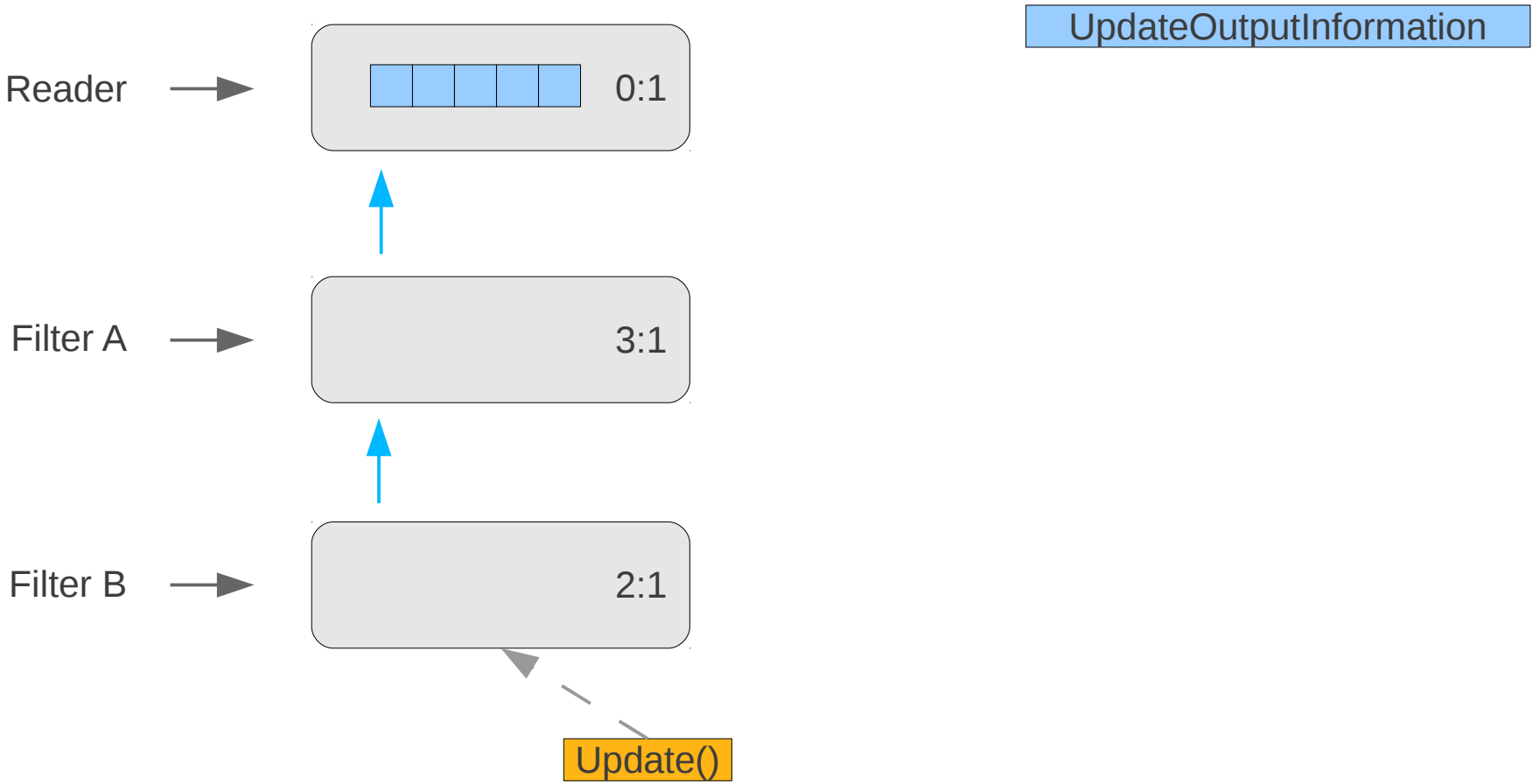


ITK Video Pipeline

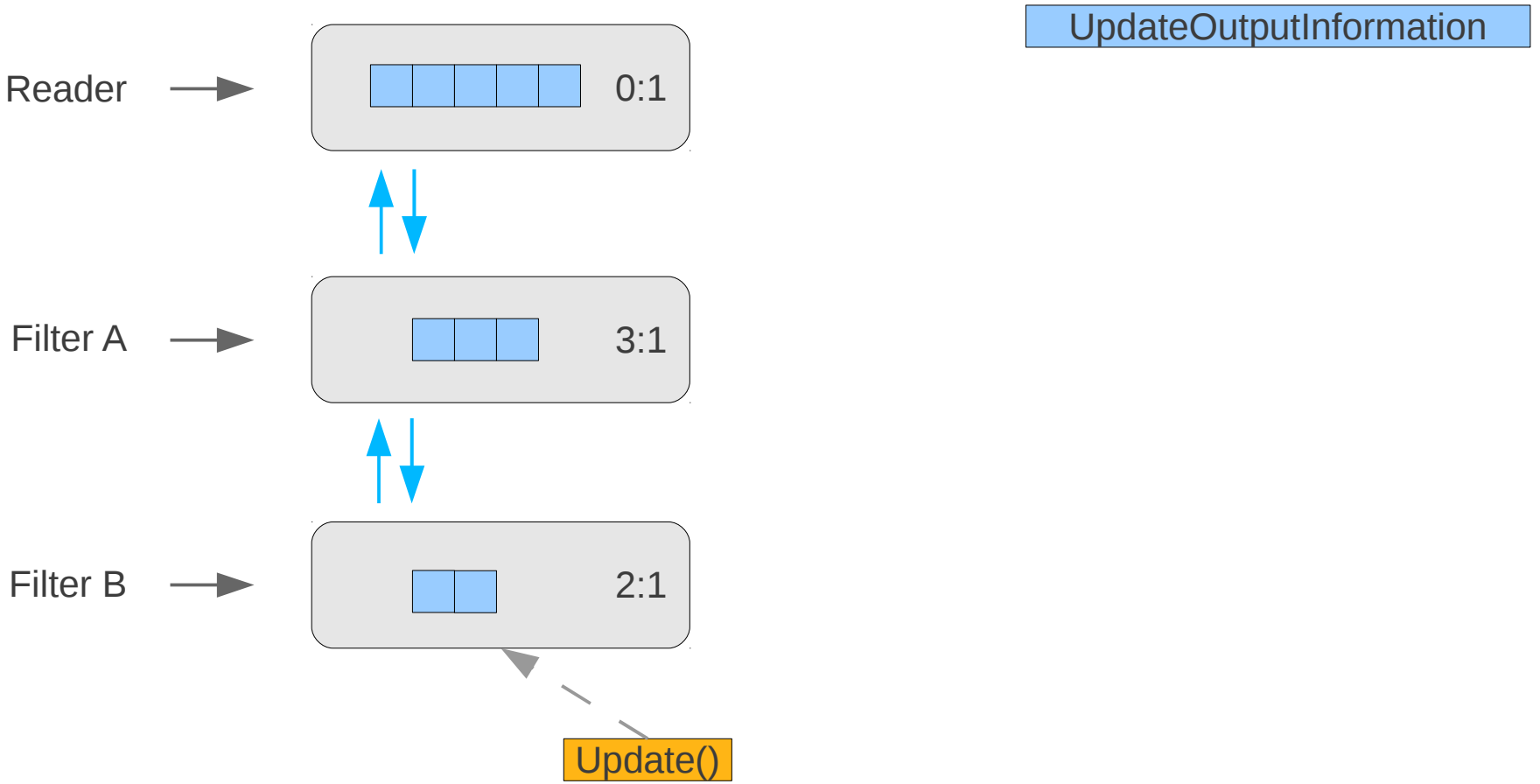
ITK Video Pipeline



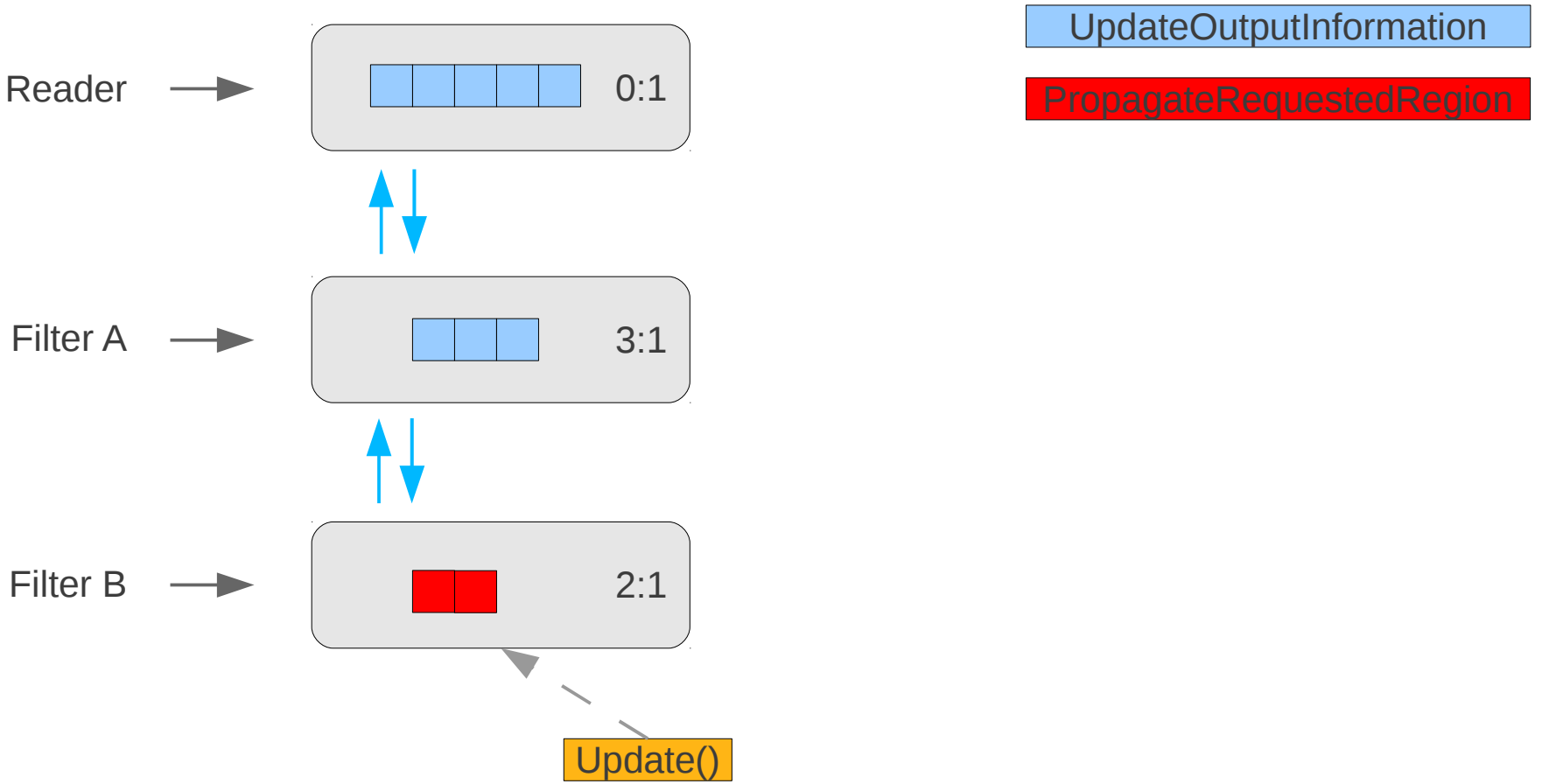
ITK Video Pipeline



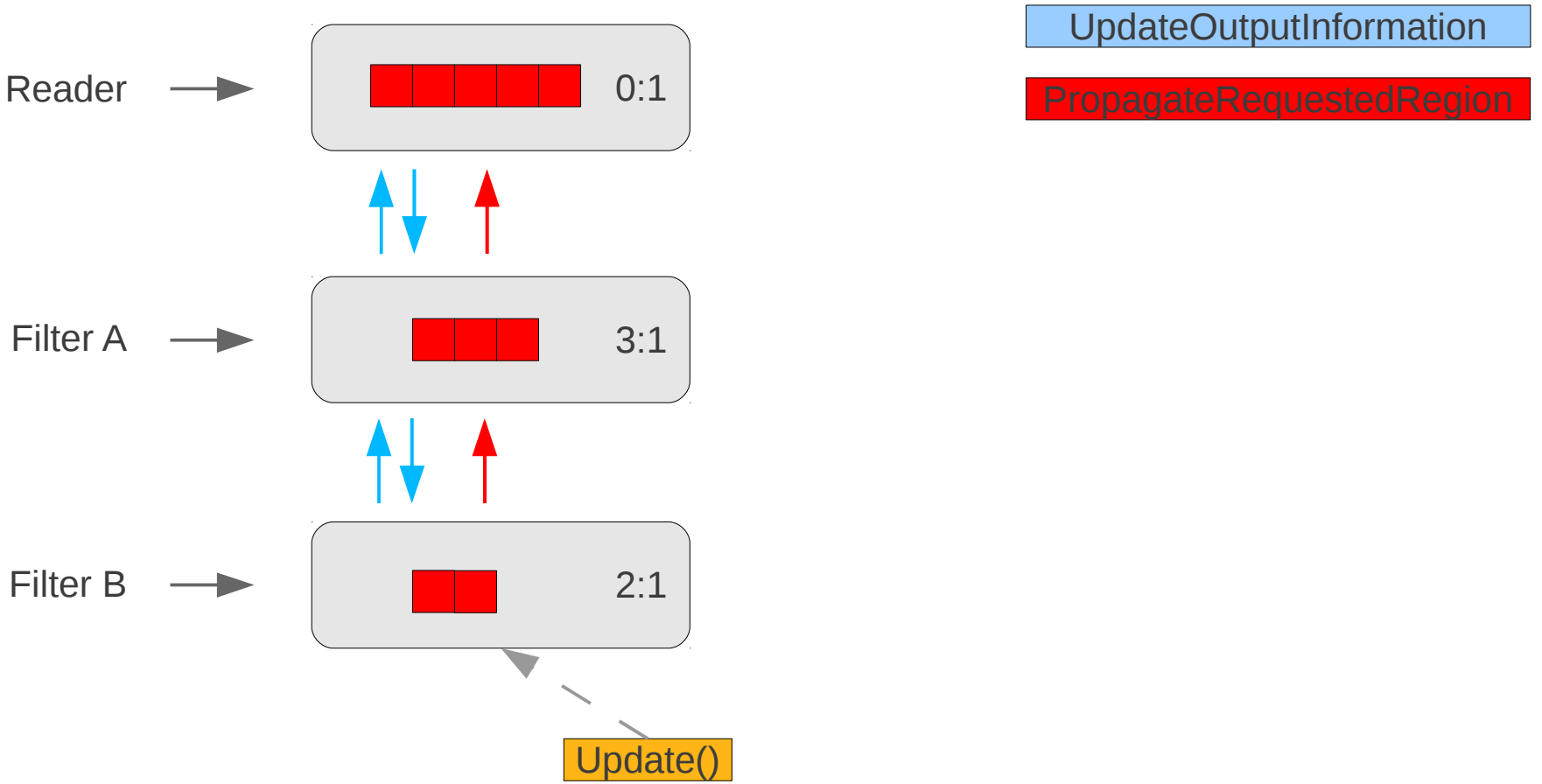
ITK Video Pipeline



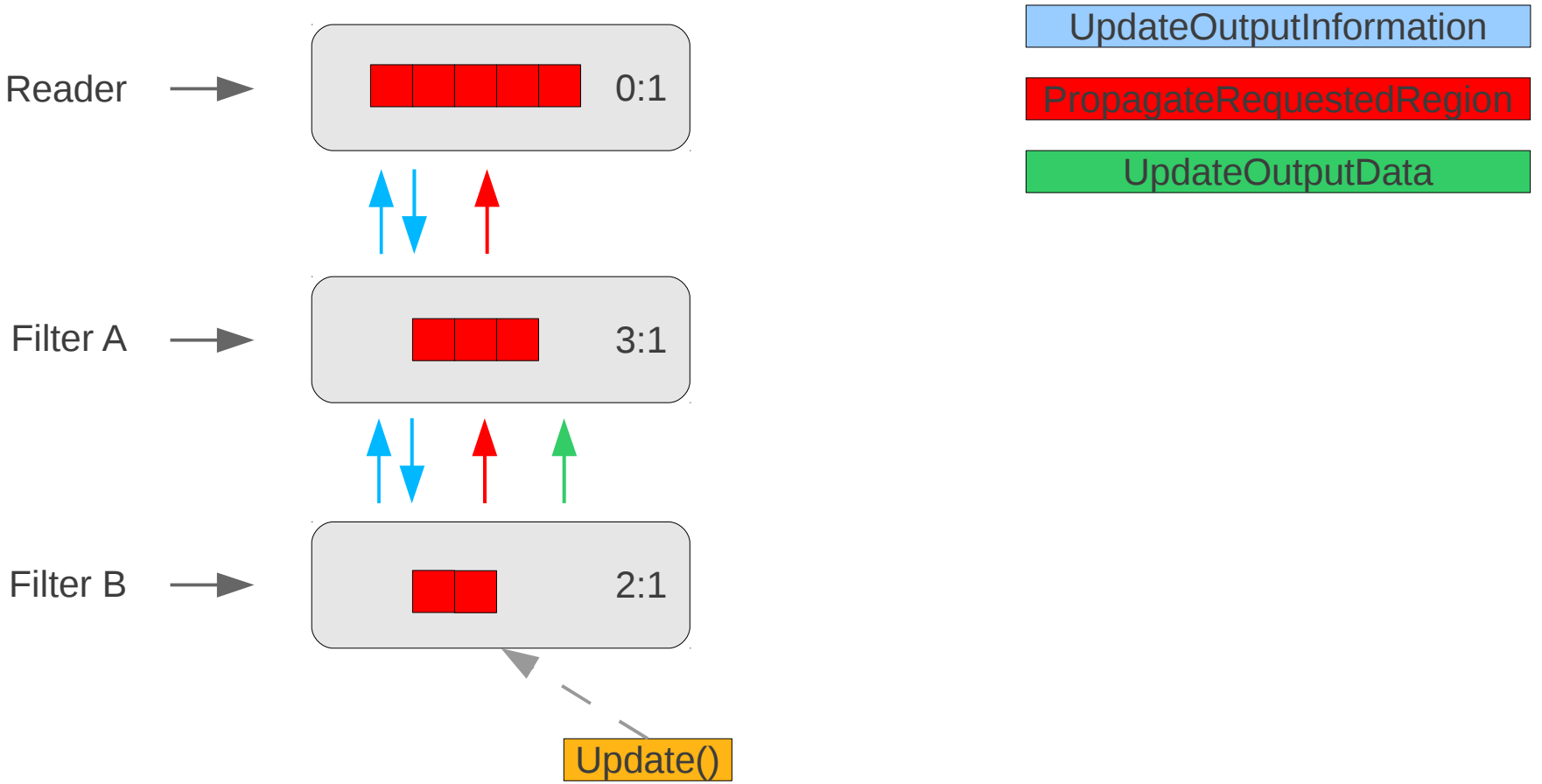
ITK Video Pipeline



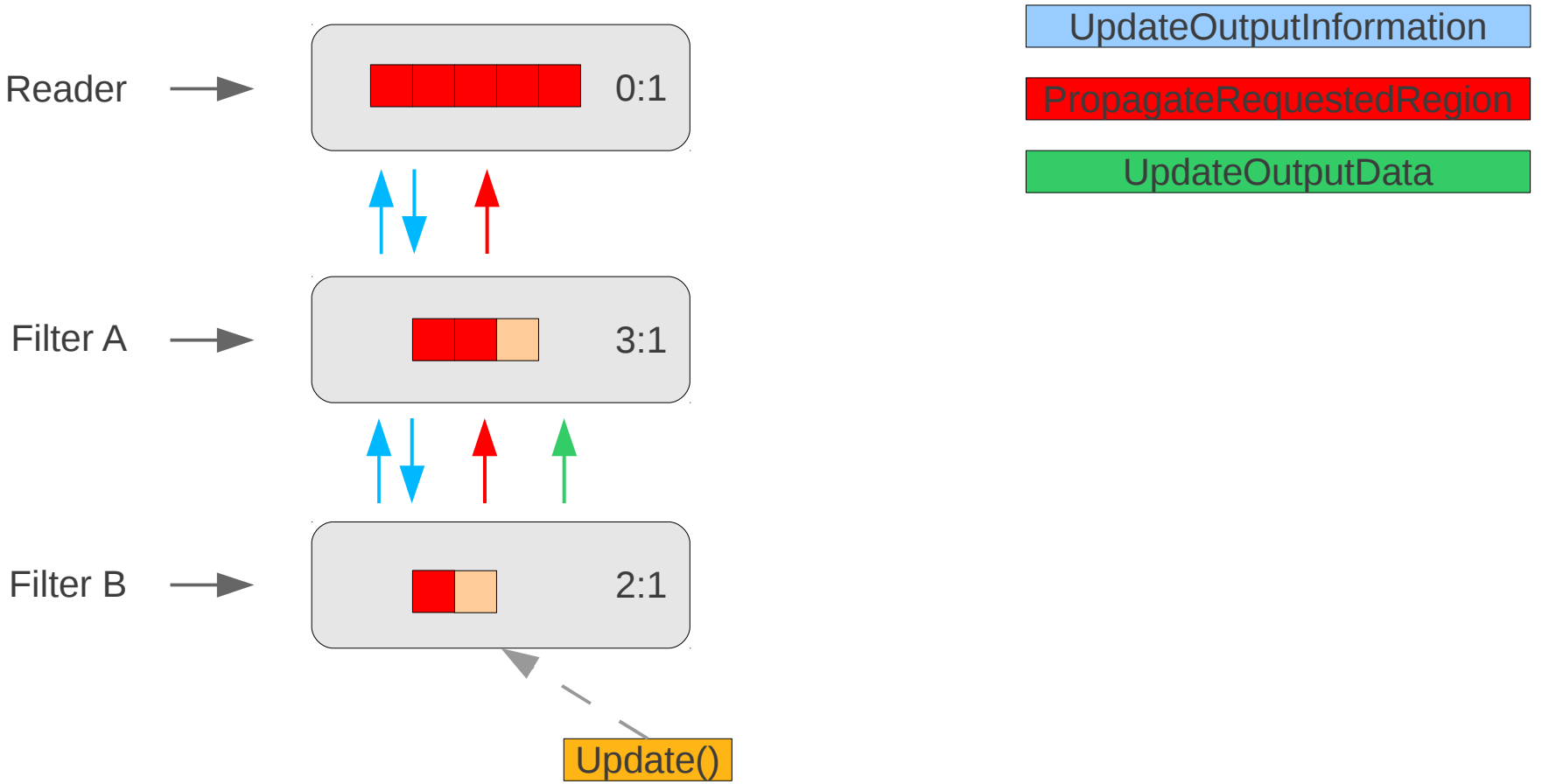
ITK Video Pipeline



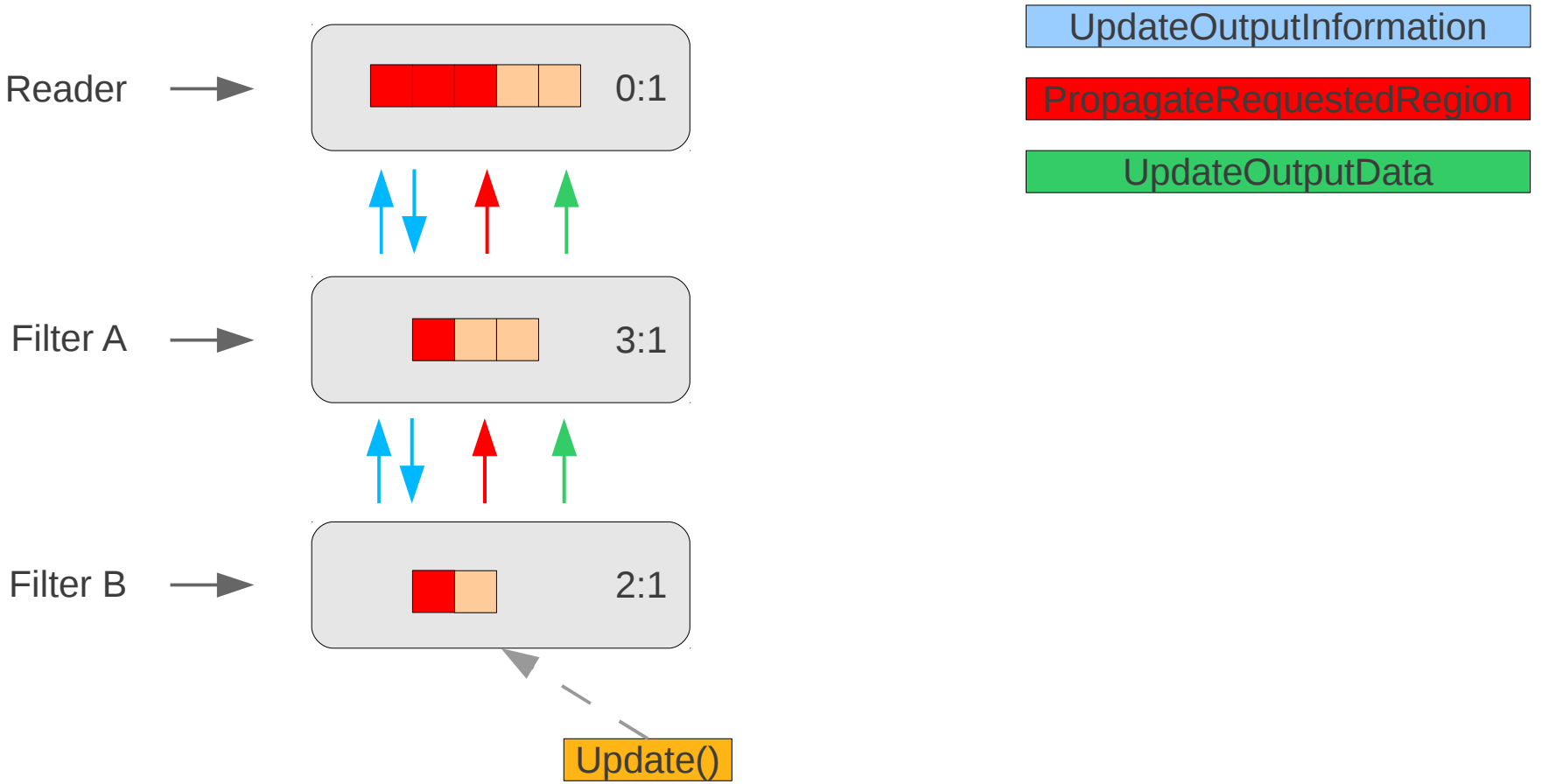
ITK Video Pipeline



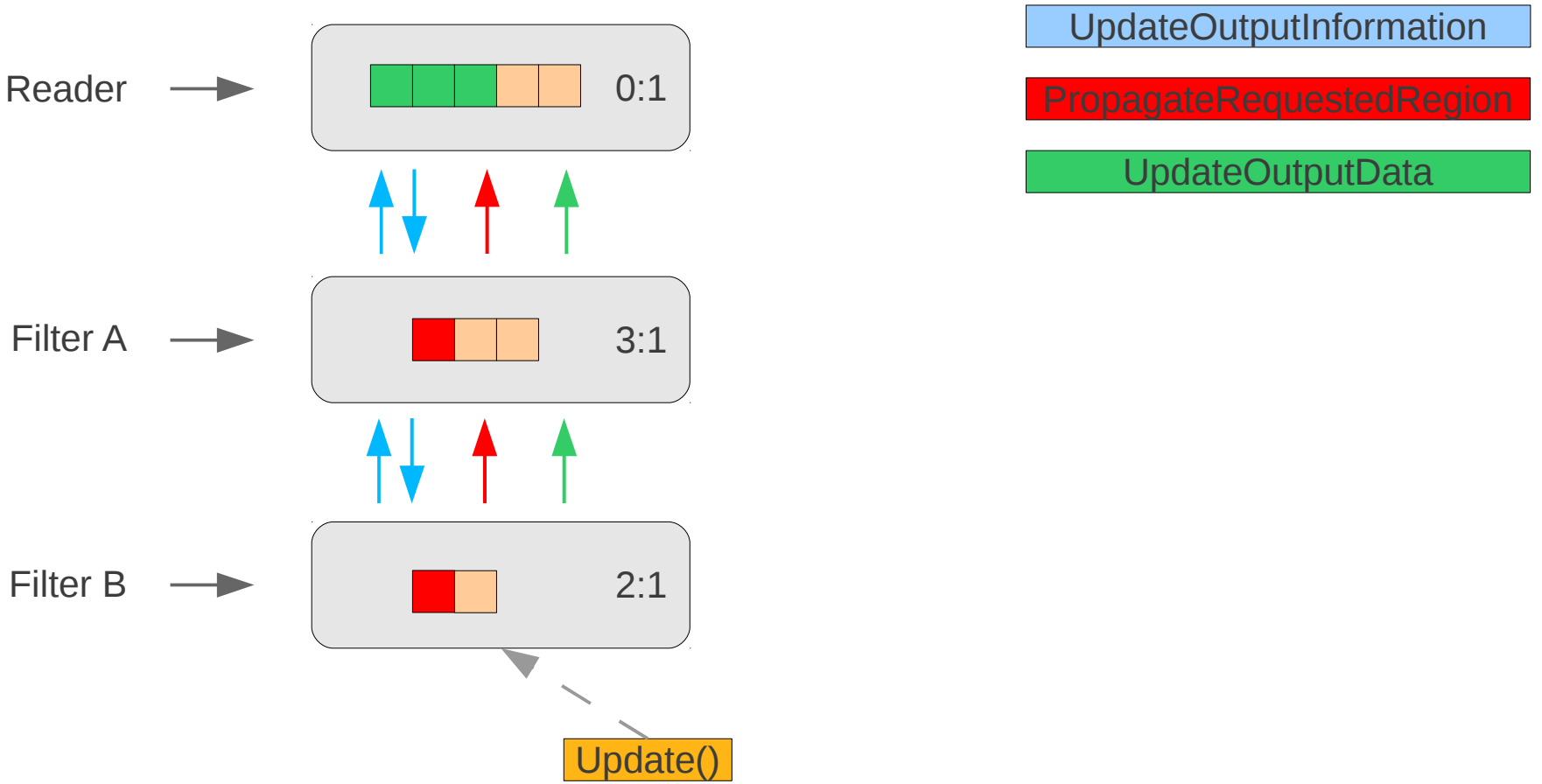
ITK Video Pipeline



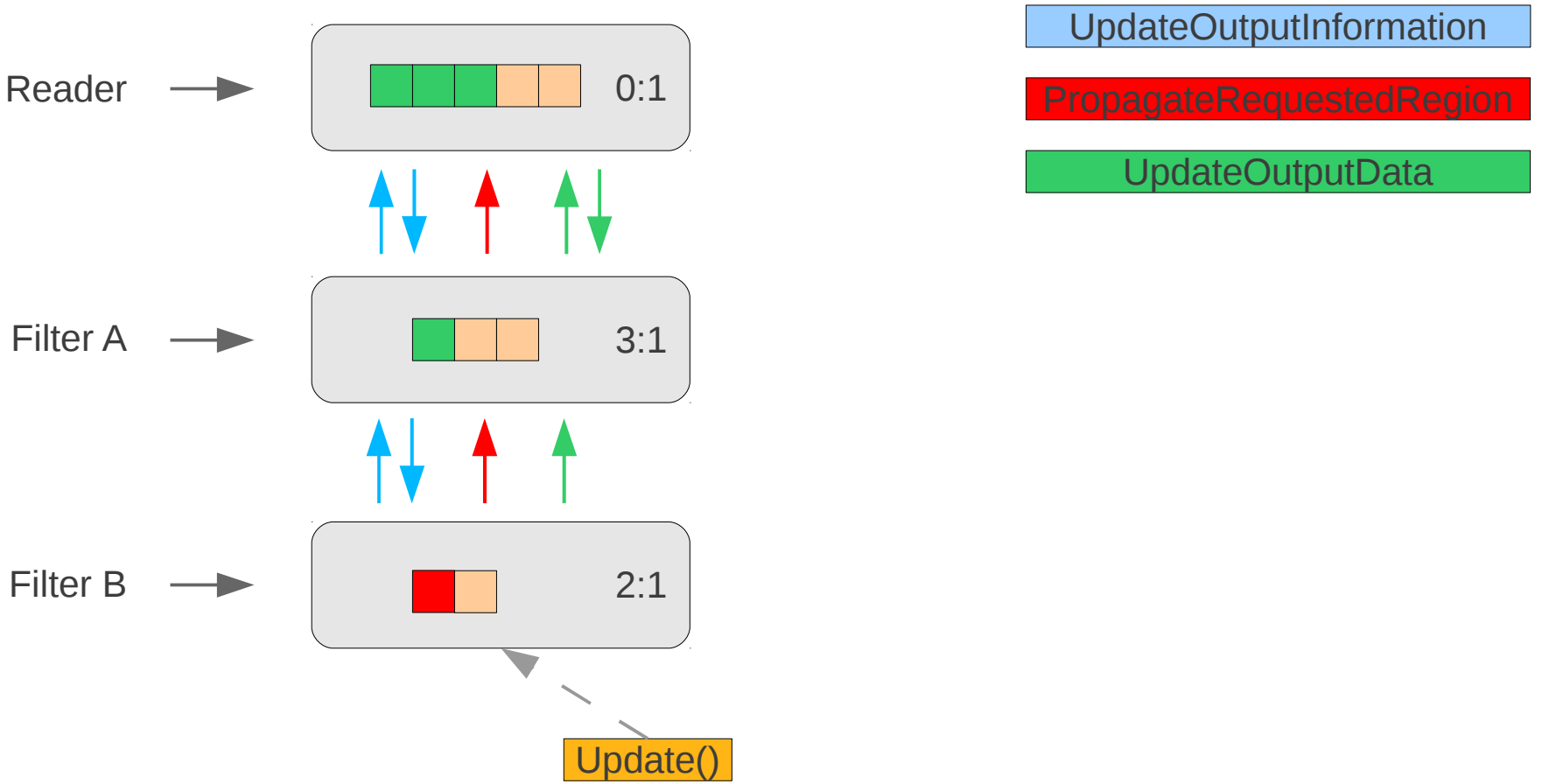
ITK Video Pipeline



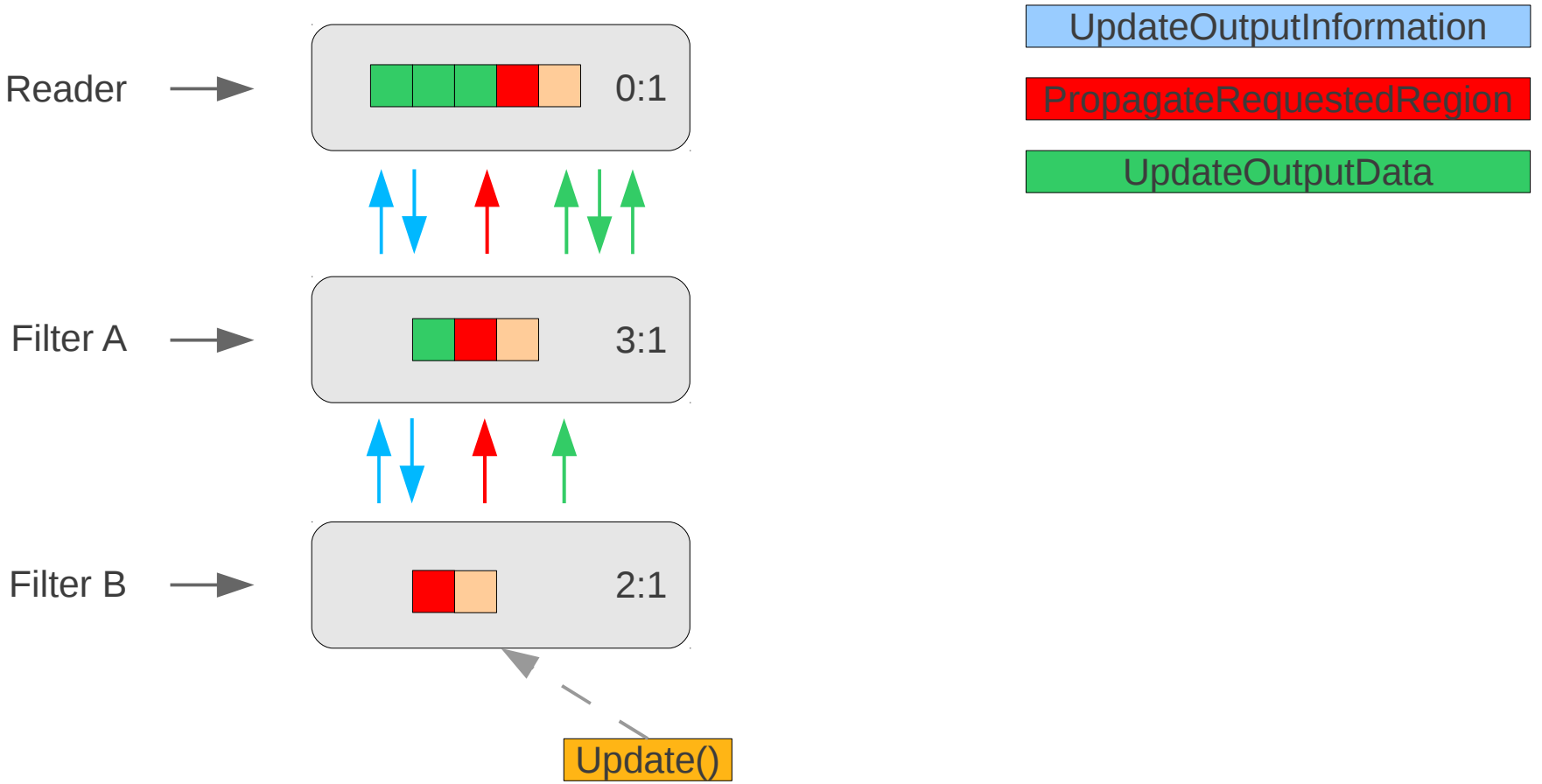
ITK Video Pipeline



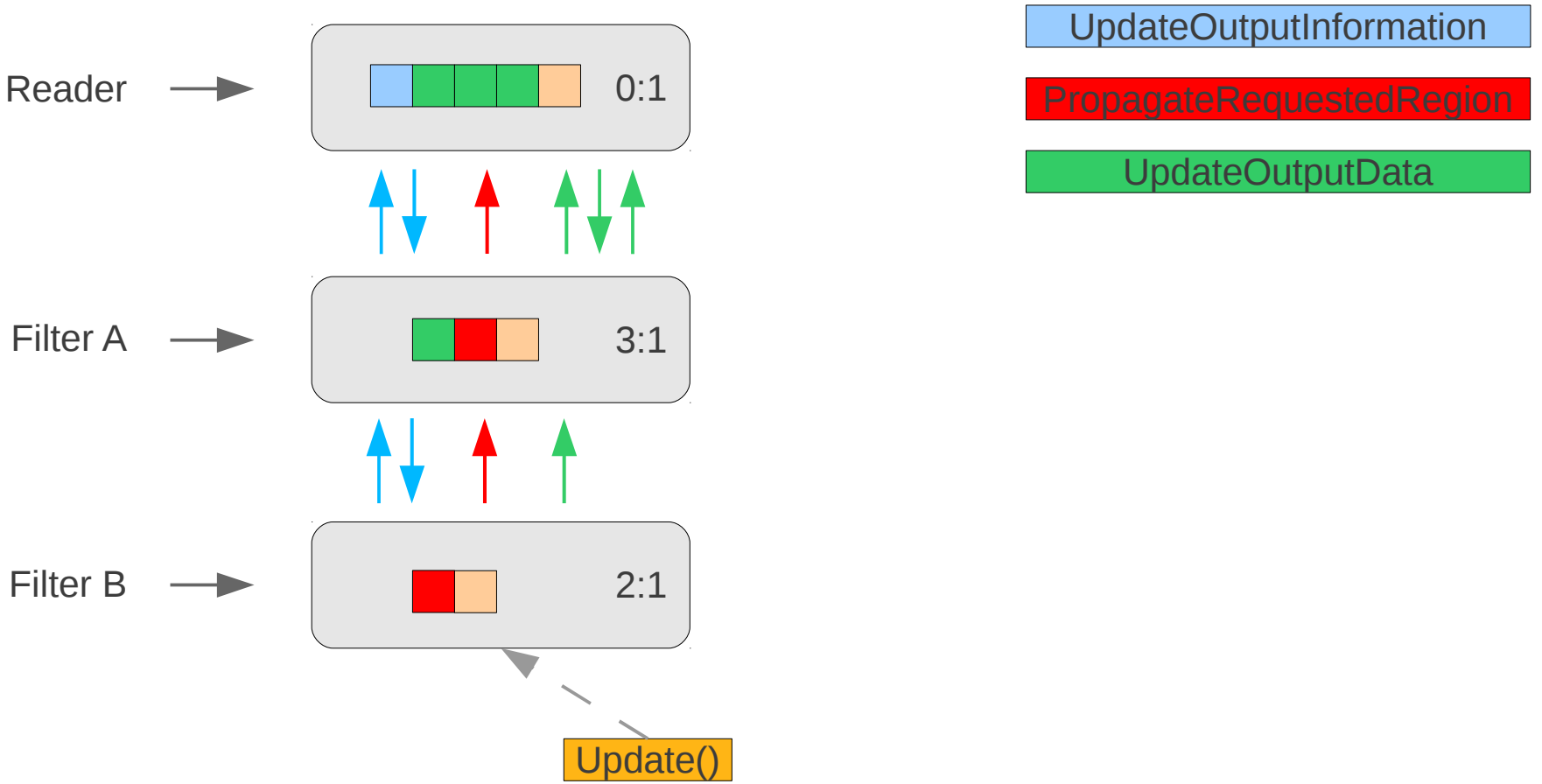
ITK Video Pipeline



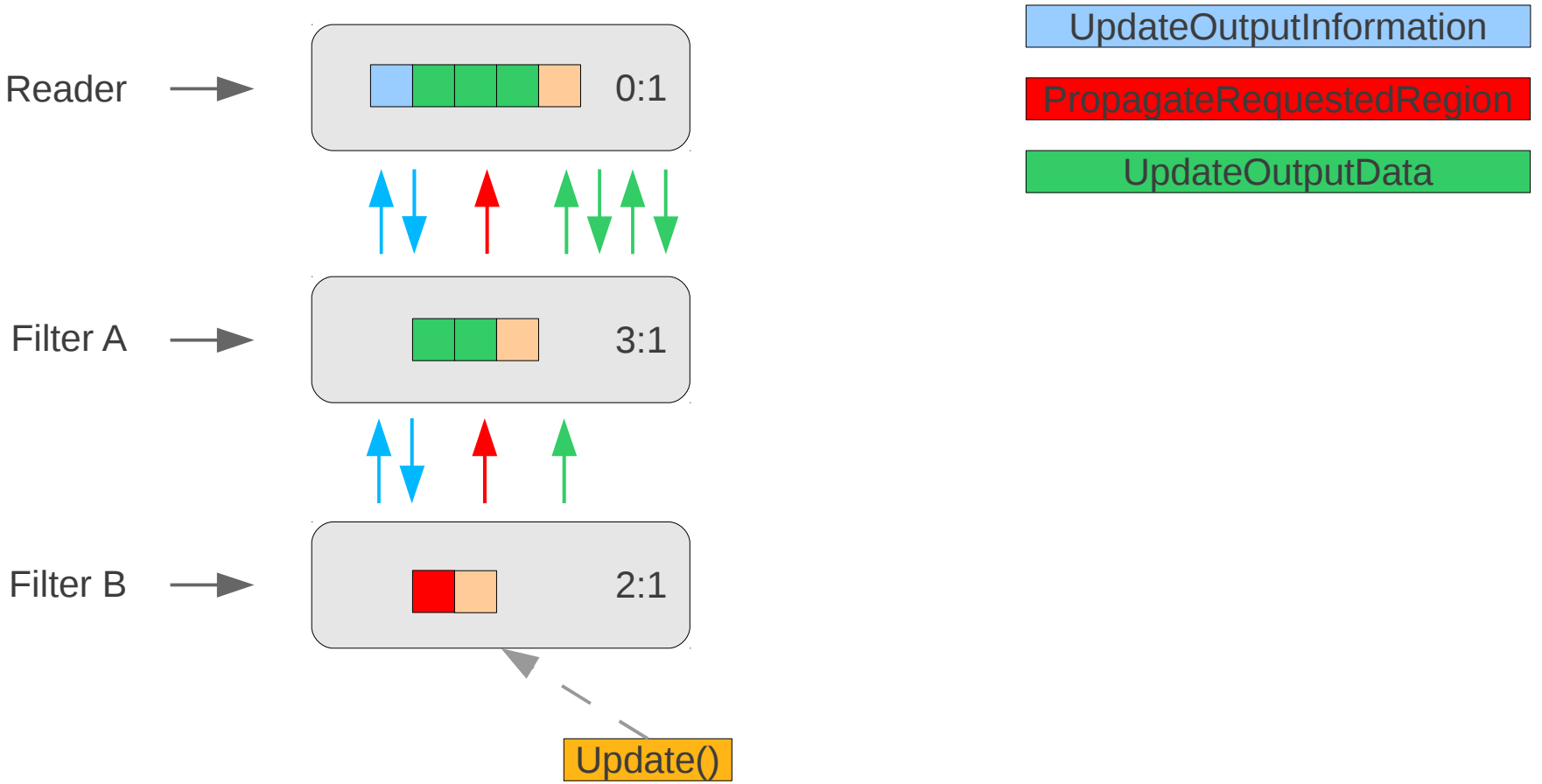
ITK Video Pipeline



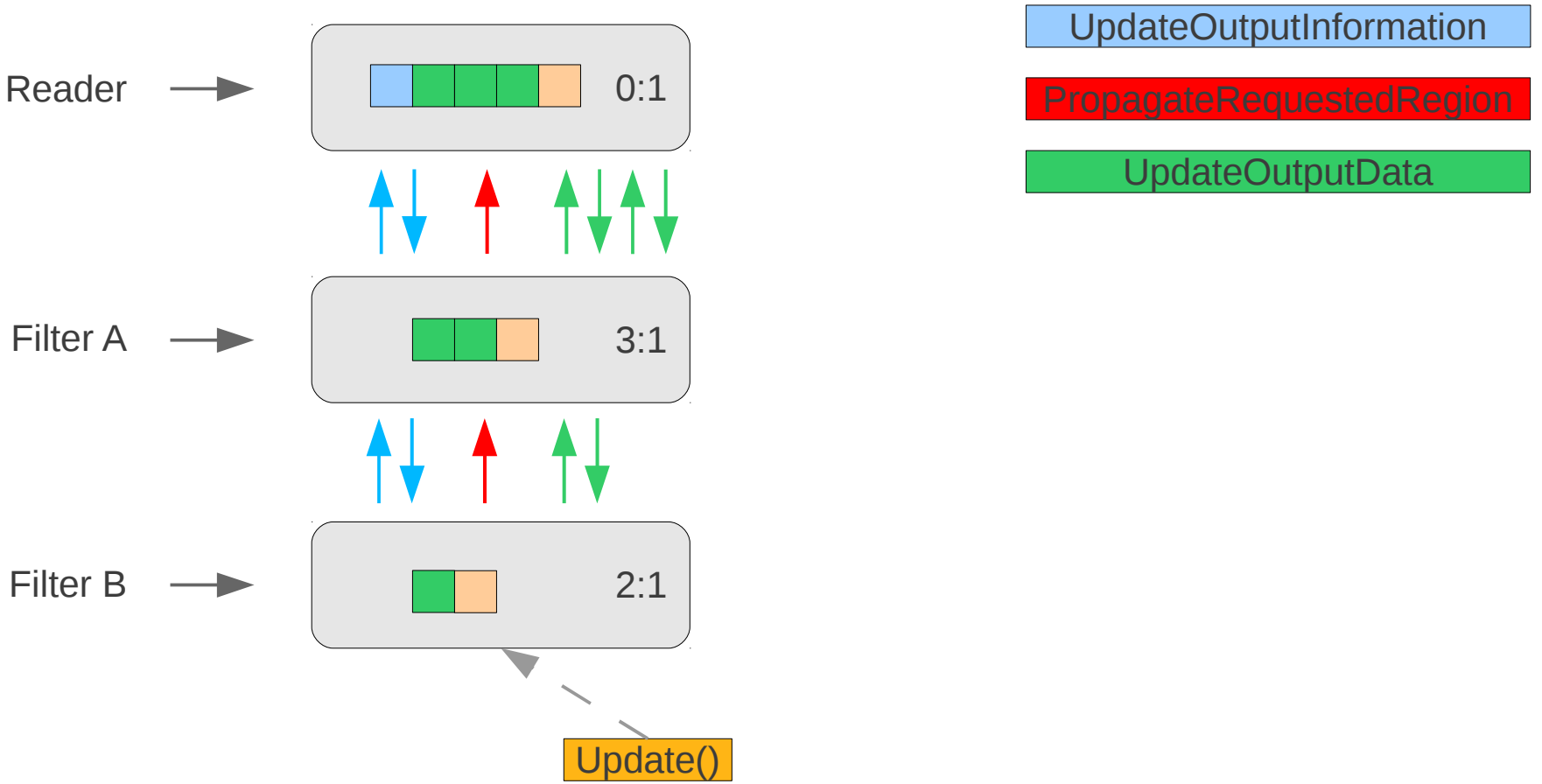
ITK Video Pipeline



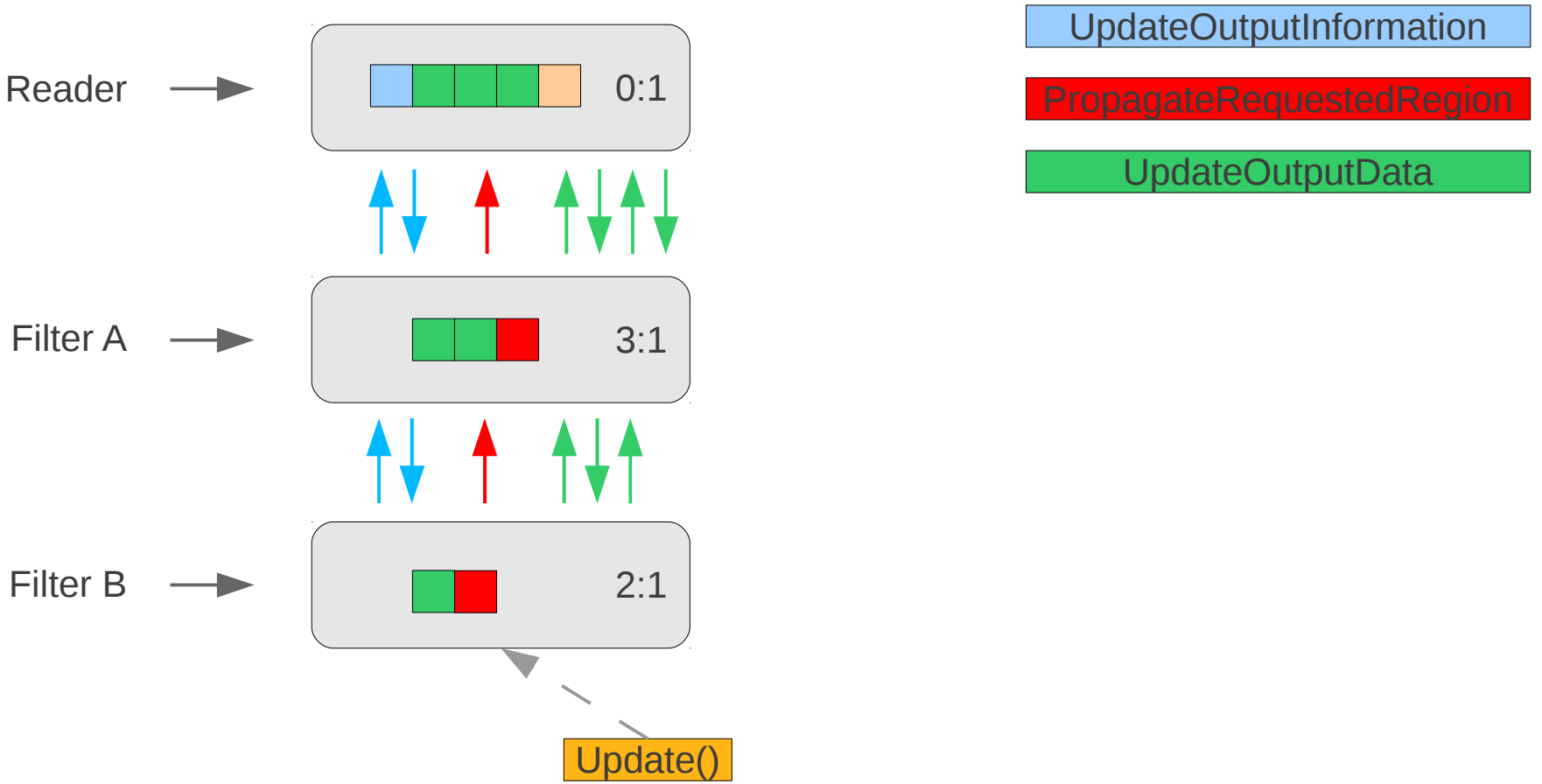
ITK Video Pipeline



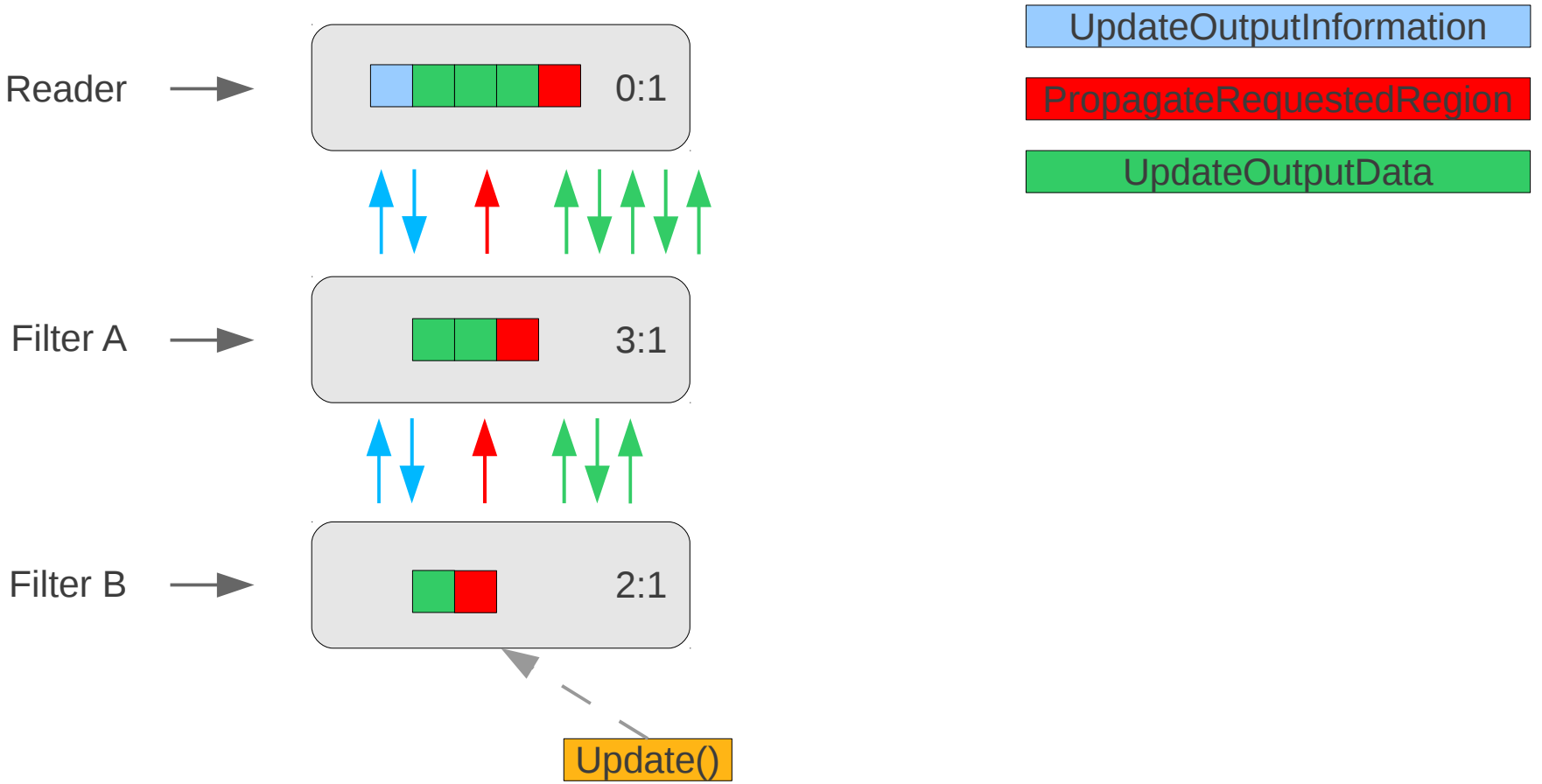
ITK Video Pipeline



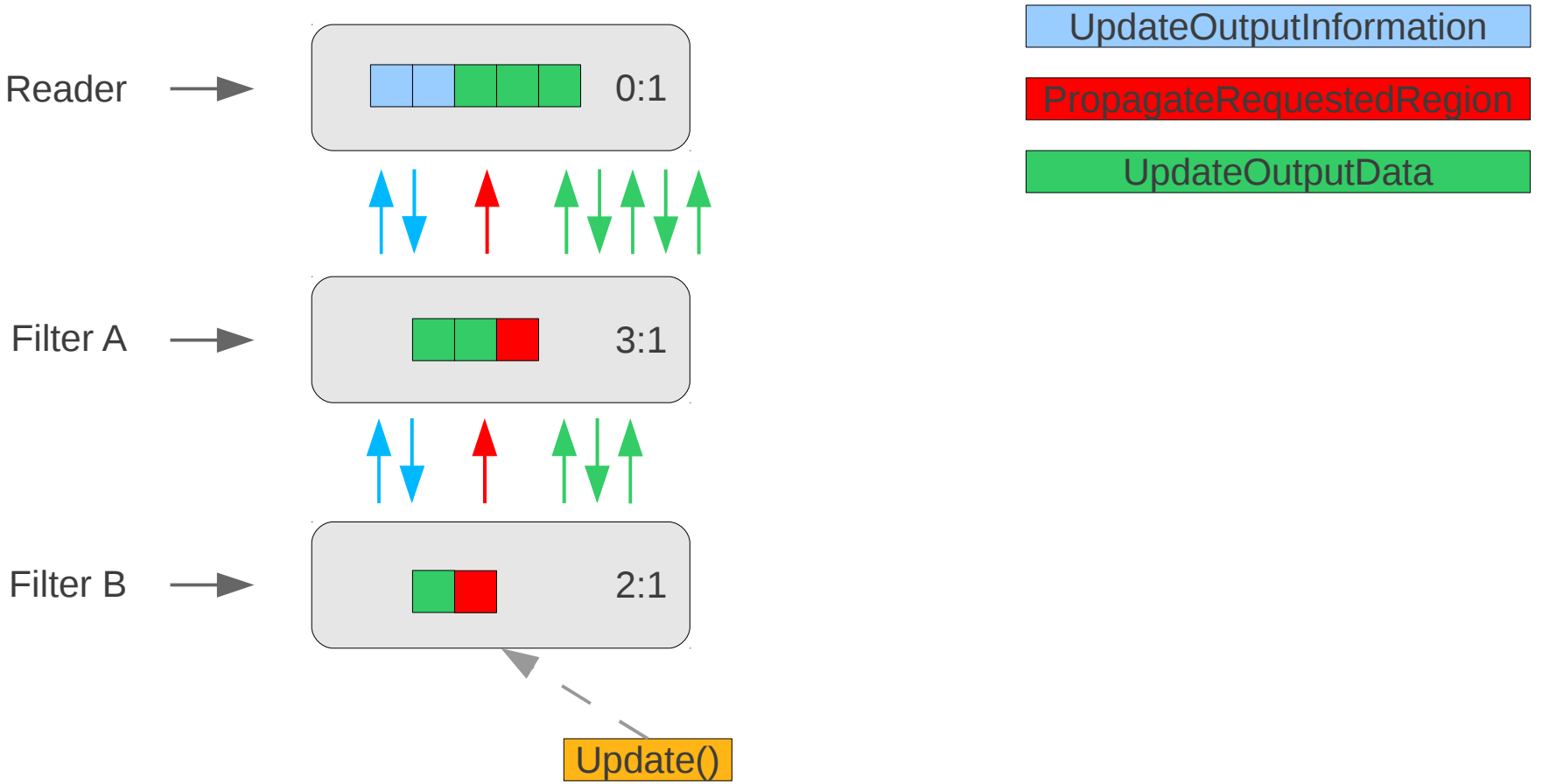
ITK Video Pipeline



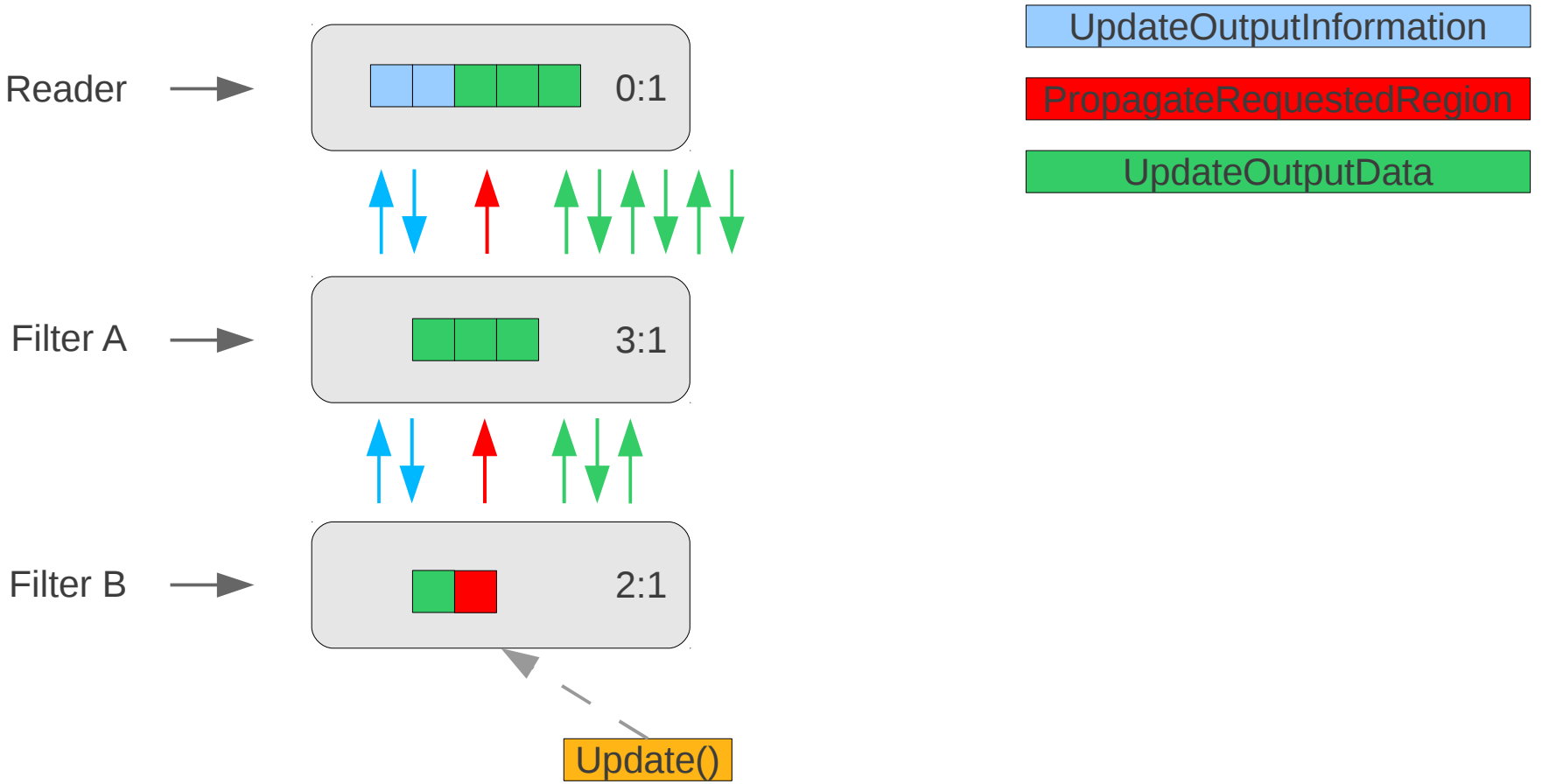
ITK Video Pipeline



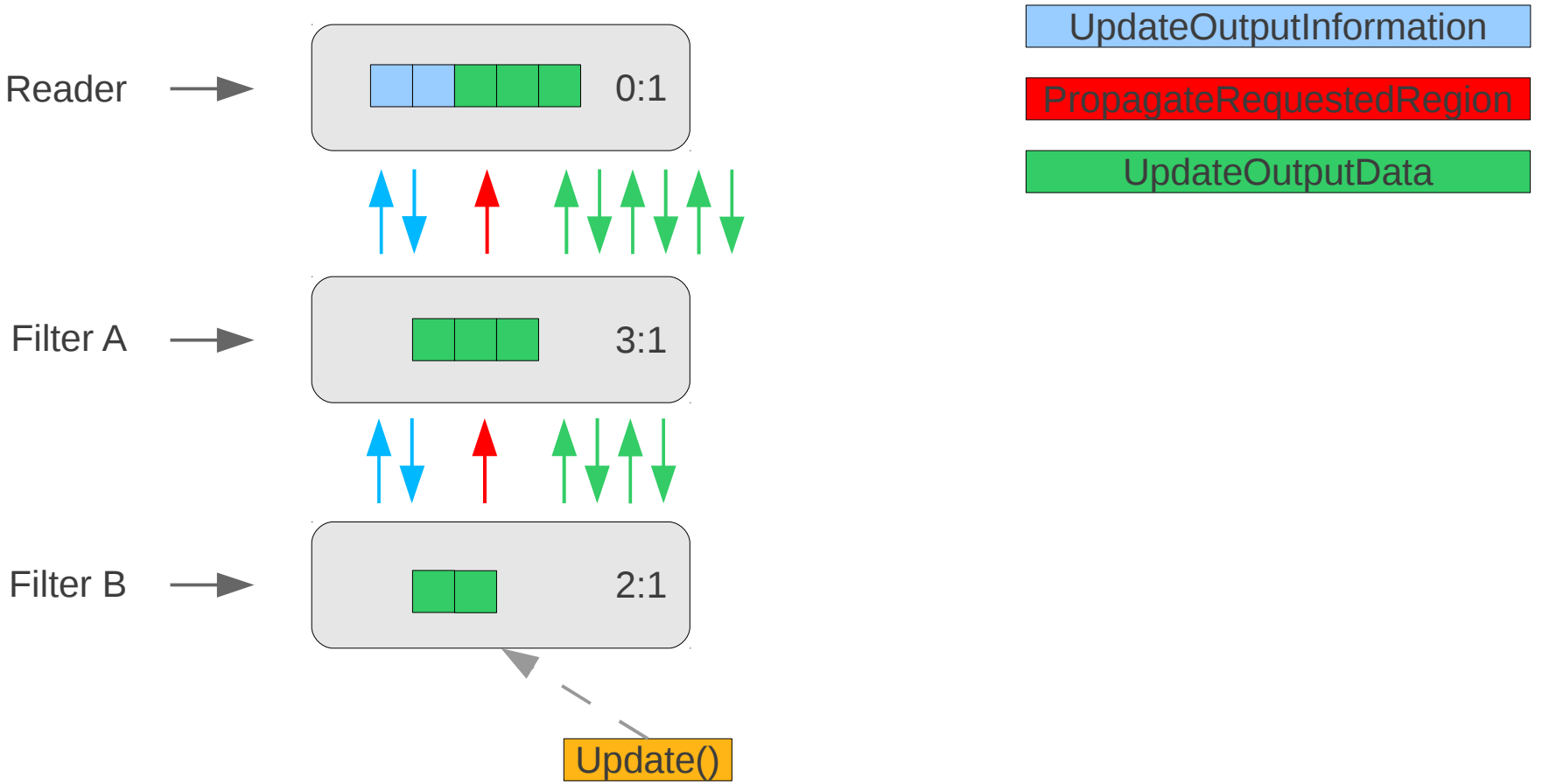
ITK Video Pipeline



ITK Video Pipeline



ITK Video Pipeline



OpenCV / VXL Bridges

Single Image Bridges

- itk::OpenCVImageBridge
  - IplImage ↔ itk::Image
  - cv::Mat ↔ itk::Image

- itk::VXLImageBridge (forthcoming)
  - vil_image ↔ itk::Image
  - vidl_frame ↔ itk::Image

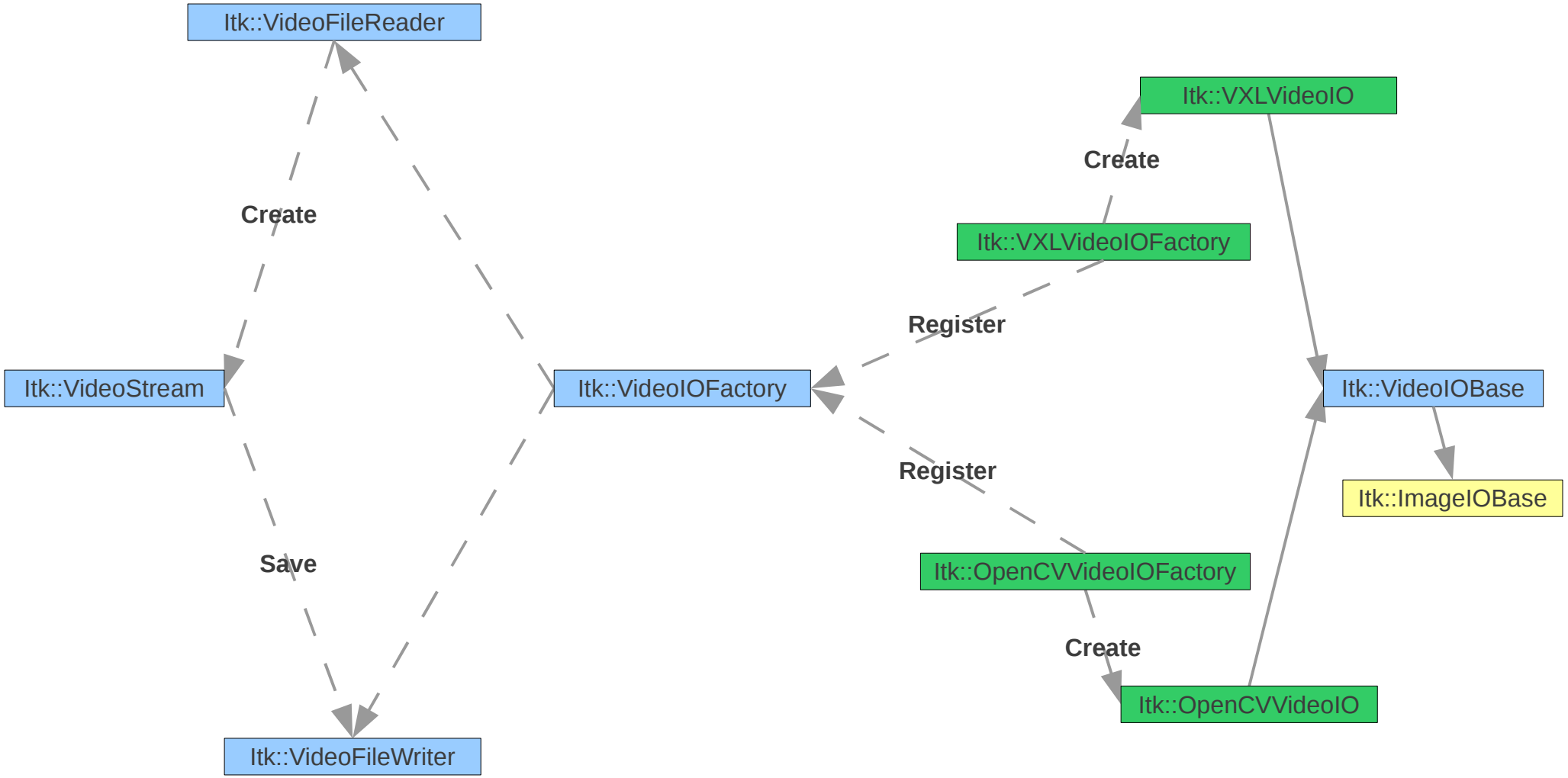
Video Bridges

- itk::OpenCVVideoCapture
  - Subclass of OpenCV's cv::VideoCapture
  - Takes itk::VideoStream as input
  - Triggers pipeline attached to input if necessary

- itk::vidl_itk_istream
  - Subclass of VXL's vidl_istream
  - Takes itk::VideoStream as input
  - Triggers pipeline attached to input if necessary

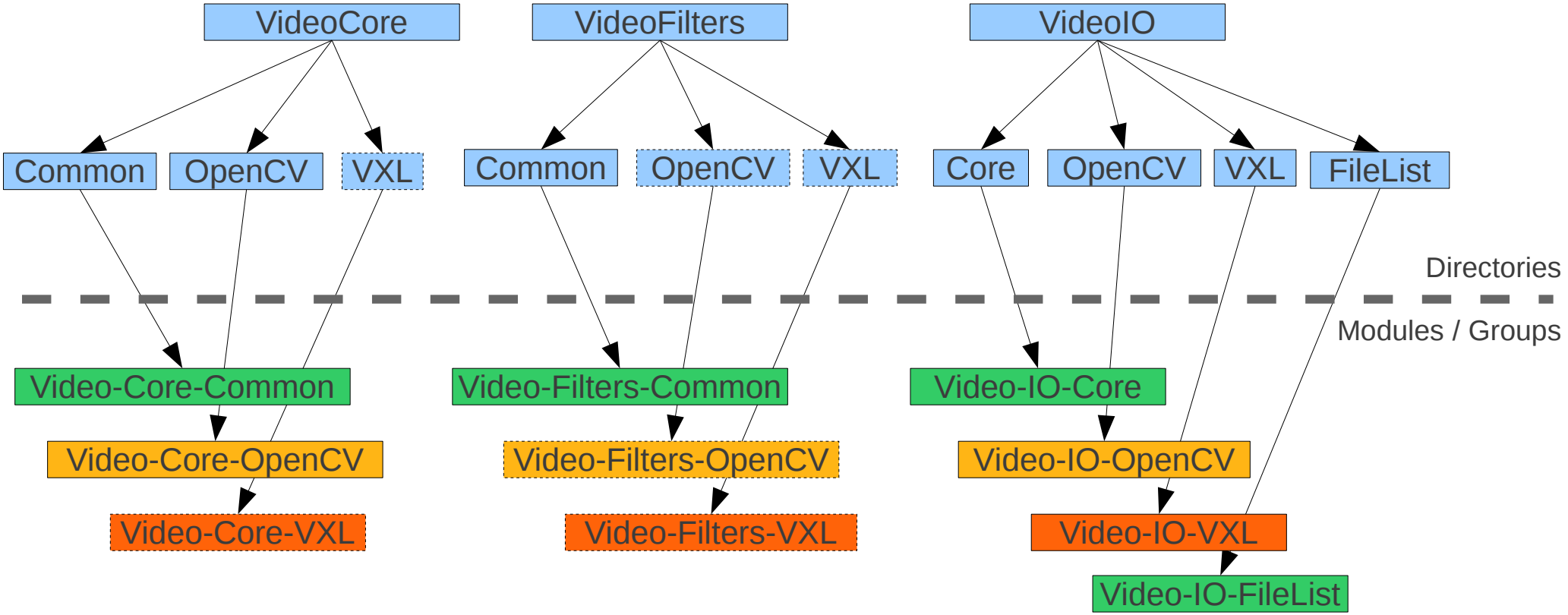
IO Mechanism

IO Mechanism



Module Structure

Module Structure - Current

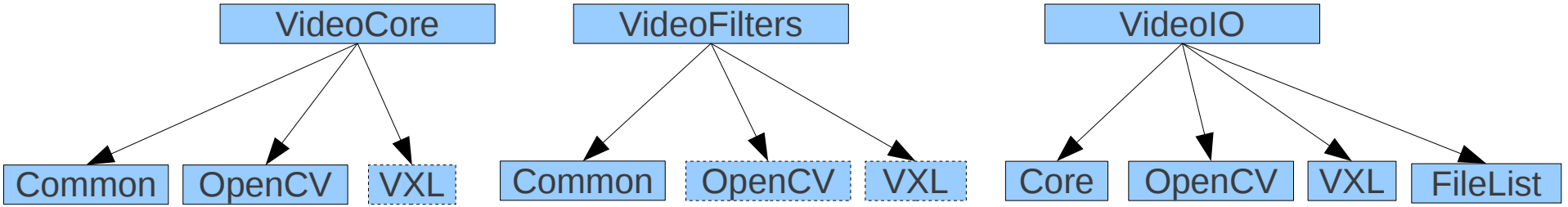


Group_VideoCommon

Group_VideoOpenCV

Group_VideoVXL

Module Structure – Desired (?)



? ? ?

Directories

Modules / Groups

Video-Core-Common

Video-Core-OpenCV

Video-Core-VXL

Video-Filters-Common

Video-Filters-OpenCV

Video-Filters-VXL

Video-IO-Core

Video-IO-OpenCV

Video-IO-VXL

Video-IO-FileList

Group_VideoCommon

Group_VideoOpenCV

Group_VideoVXL

Issues/ToDo

- RealTimeStamp is not a real time stamp
- I/O readers
  - Image sequence as a video (done via file list)
  - Multi-frame DICOM file as a video
  - Video capture
    - sync with Children's National on video reader and on using framework
  - Direct video I/O (using ffmpeg)
- VideoStream → 3-d volume converter
- Handle infinite duration
- Implement timestamp-based flow (vs frame-based flow)
- Implement video filter base classes for multiple inputs and outputs
  - Synchronization of multiple streams
- Avoid memcpy where possible (when wrapping other libraries)
- Ensure frame size is not an intrinsic problem
  - Working with SD video now, but will need to work with HD video
- Implement filters in new video framework (e.g. using OpenCV)
- Translators for other data structures (e.g. OpenCV sequences, tracks, etc.)

END