



ITK Architecture

Luis Ibáñez
William Schroeder
Insight Software Consortium

ITK Basics

- C++ Generic Programming
- Data Pipeline
- Multi-threading
- Streaming
- Exceptions
- Events / Observers
- Tcl and Python wrapping

Generic Programming

Example: STL Standard Template Library

Abstraction of Types and Behaviors

```
std::vector< T >
```

```
std::vector< int >
```

```
std::vector< double >
```

```
std::vector< char * >
```

```
std::vector< Point >
```

```
std::vector< Image >
```

itk::Image

itk::Image< PixelType , Dimension >

itk::Image< char , 2 >

itk::Image< char , 3 >

itk::Image< char , 4 >

itk::Image< float , 2 >

itk::Image< RGB , 3 >

itk::Image< unsigned short , 2 >

itk::Image< itk::Vector<float,2> , 2 >

namespaces

Avoid naming collisions

itk::

itk::Statistics::

itk::fem::

itk::fem::itpack

itk::bio

Your favorite keyword

typedef

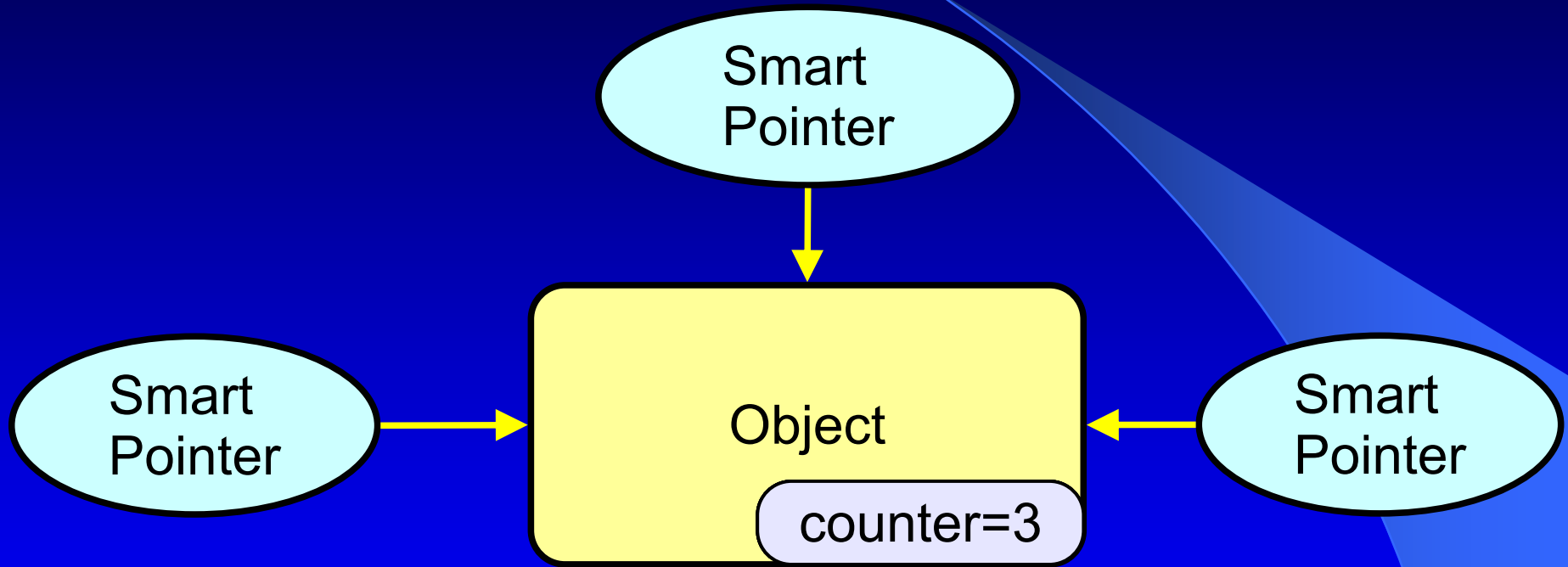
```
typedef itk::Image< char , 2 > ImageType
```

```
typedef itk::ImageFilter< ImageType , ImageType > FilterType
```

otherwise...

```
itk::ImageFilter< Image< char , 2 > ,  
                Image< char , 2 > > FilterType
```

Smart Pointers



Self - Delete

SmartPointers

```
typedef itk::Image< char , 2 > ImageType  
typedef itk::ImageFilter< ImageType , ImageType > FilterType
```

```
FilterType::Pointer filter = FilterType::New();
```

```
ImageType::Pointer image = filter->GetOutput();
```

NO NEED FOR

```
filter->Delete();
```

Const Correctness

Knowing constancy is Insight.

Not knowing constancy leads to disaster.

Tao Te Ching, XVI. Lao Tsu

Const Smart Pointers

```
typedef itk::Image< char , 2 > ImageType
```

```
typedef itk::ImageFilter< ImageType , ImageType > FilterType
```

```
FilterType::Pointer filter = FilterType::New();
```

```
ImageType::ConstPointer image = filter->GetOutput();
```

Can only invoke “const” methods

```
image->GetSpacing ();
```

Compiler error for “non-const” methods

```
image->SetSpacing ( spacing );
```

Data Pipeline

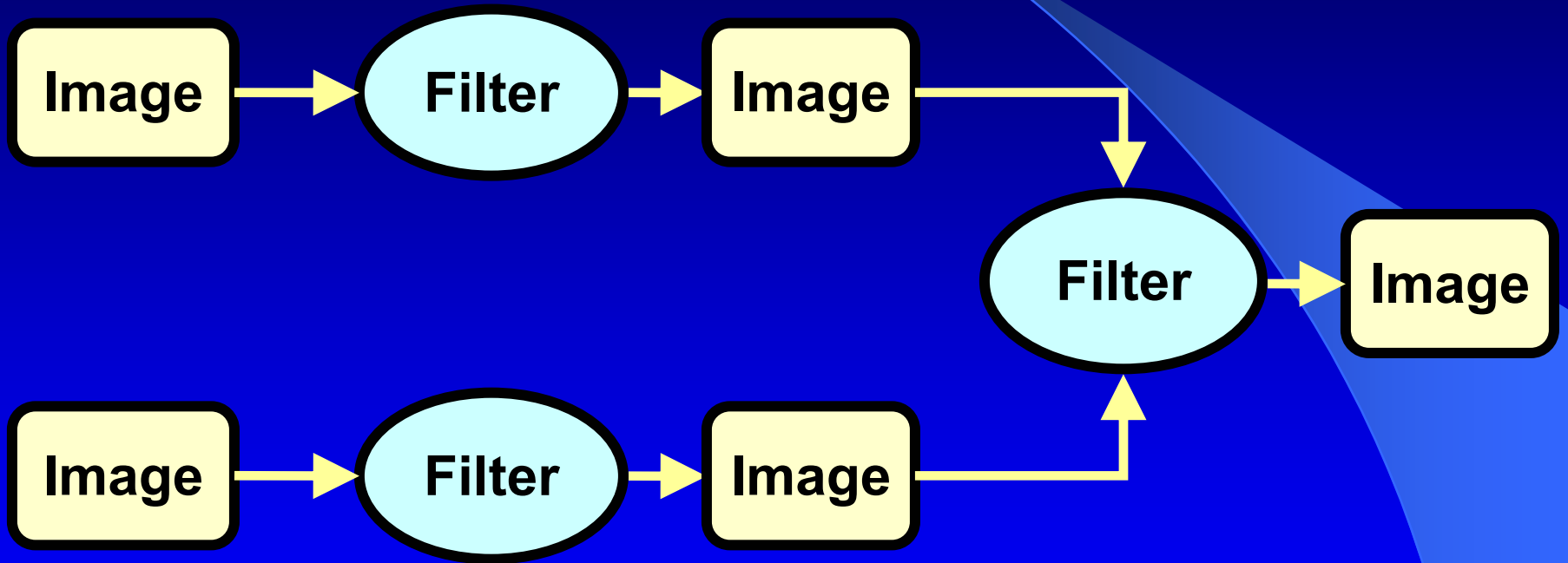
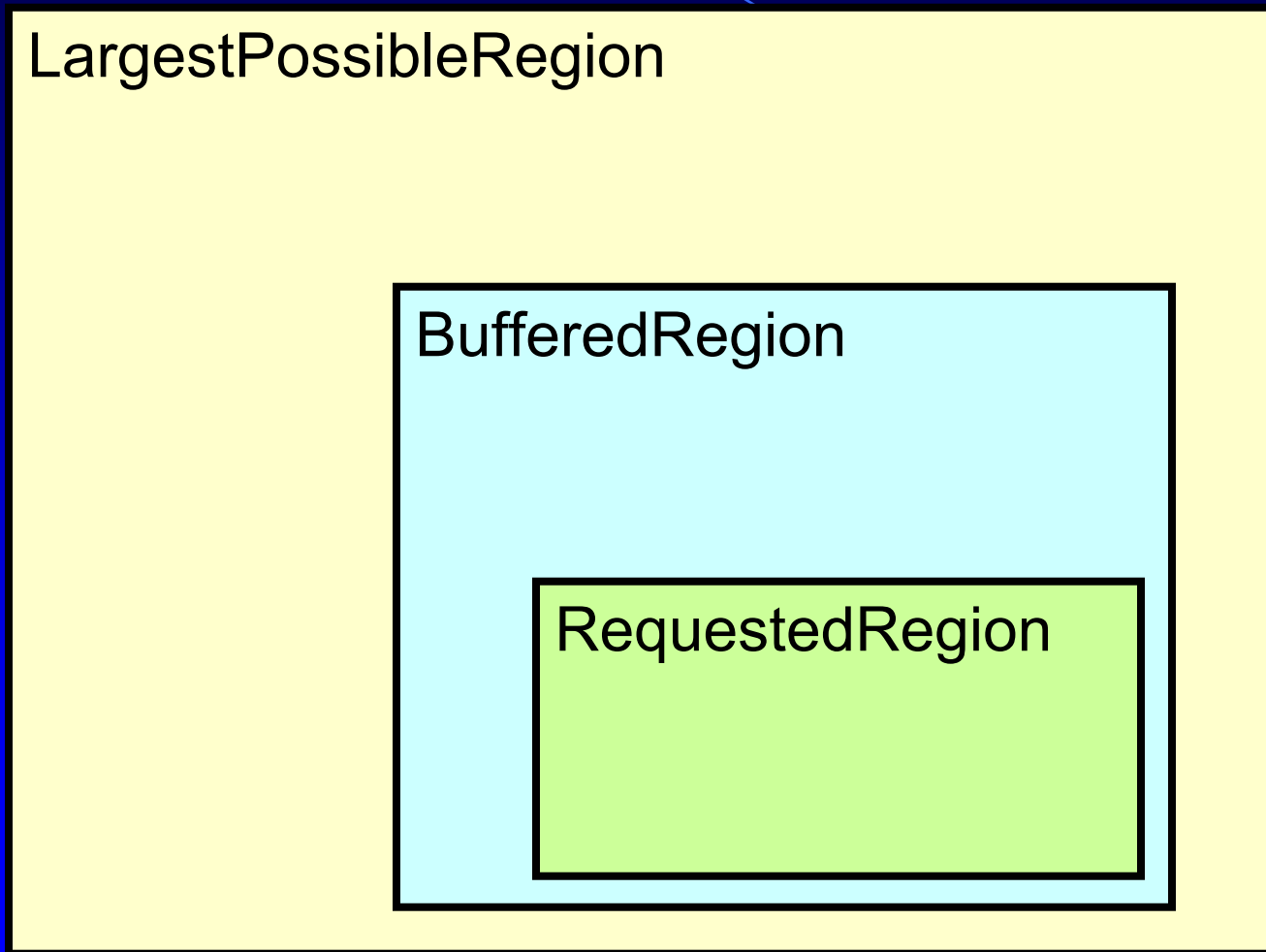
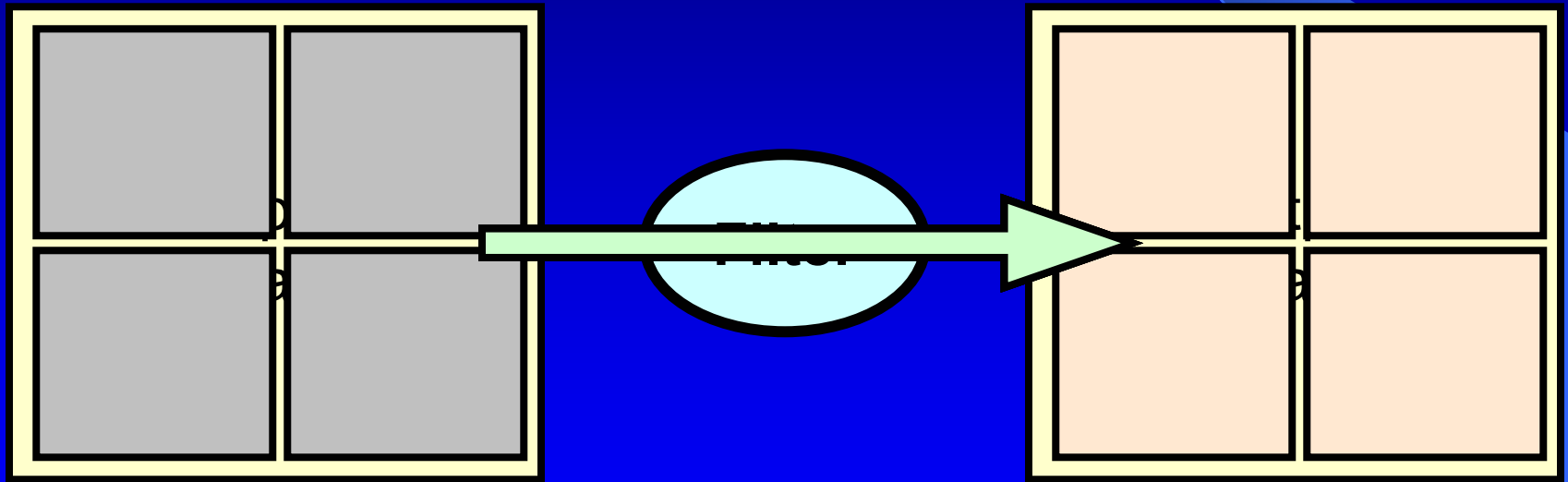


Image Regions

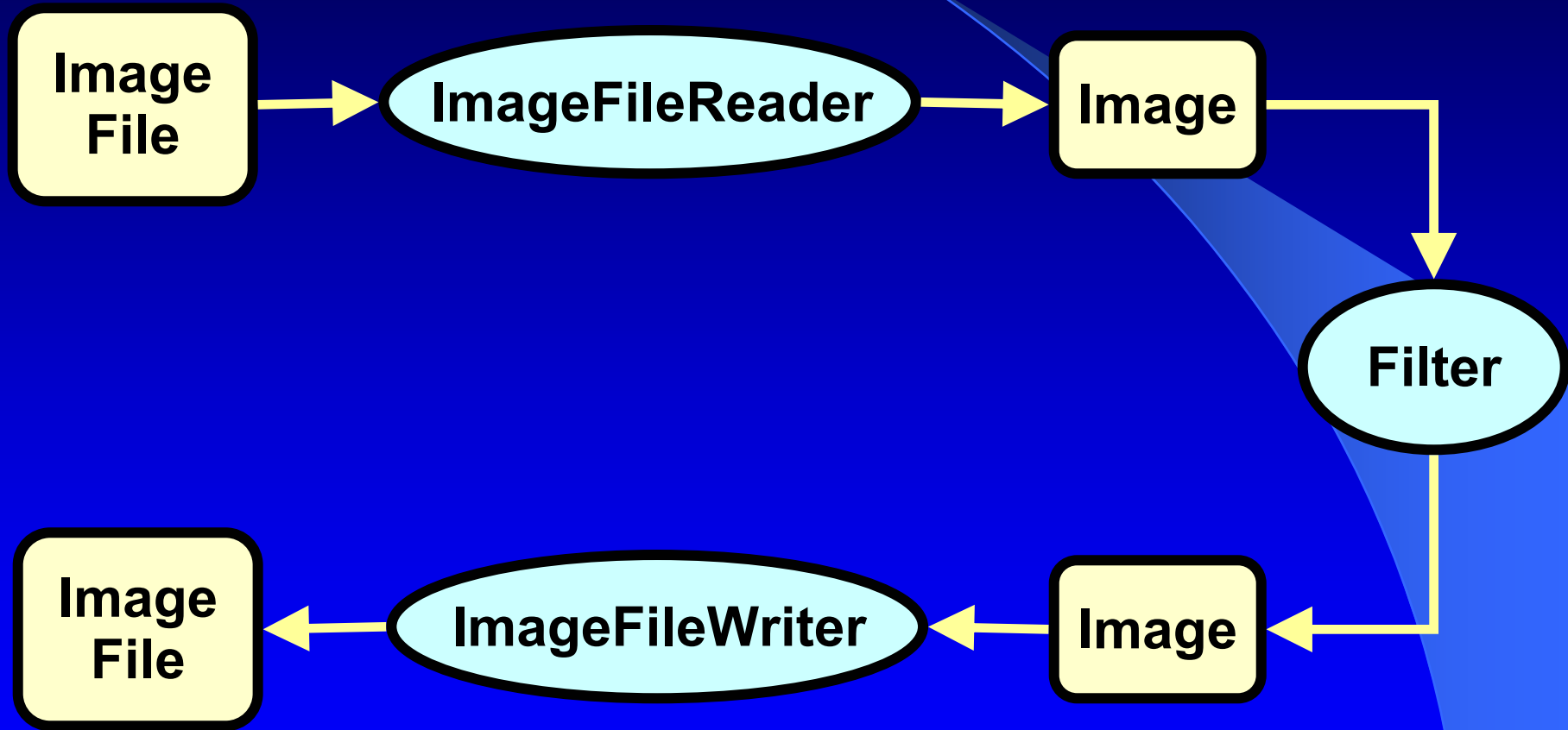


Streaming

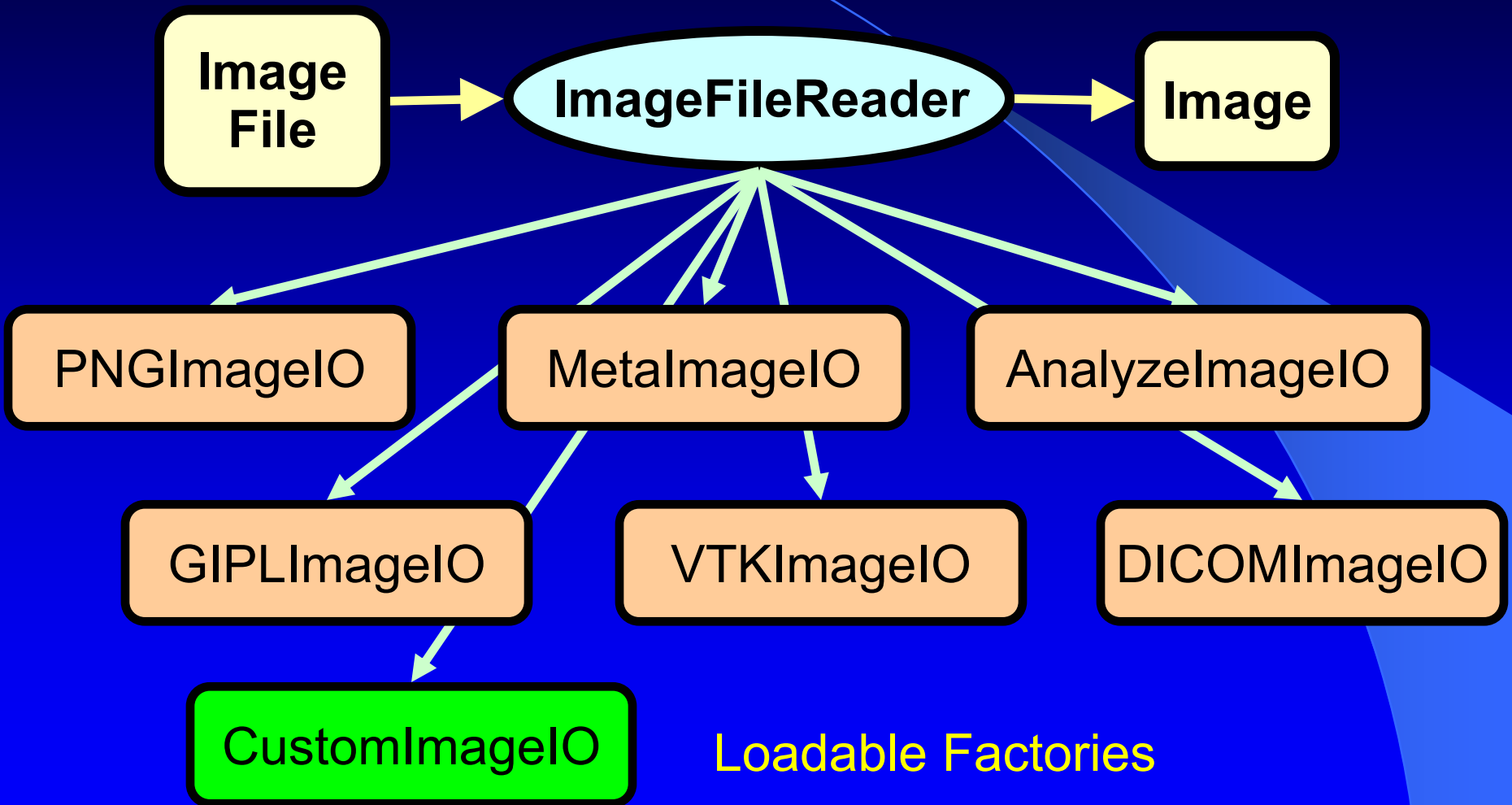
Processing Large Images



Simple Image IO



Simple Image IO



Simple Image IO

```
#include "itkImage.h"
```

```
#include "itkImageFileReader.h"
```

```
#include "itkImageFileWriter.h"
```

```
typedef itk::Image< char , 2 > ImageType;
```

```
typedef itk::ImageFileReader< ImageType > ReaderType;
```

```
typedef itk::ImageFileWriter< ImageType > WriterType;
```

```
ReaderType::Pointer reader = ReaderType::New();
```

```
WriterType::Pointer writer = WriterType::New();
```

```
reader->SetFileName( "inputImage.dcm" ); // DICOM
```

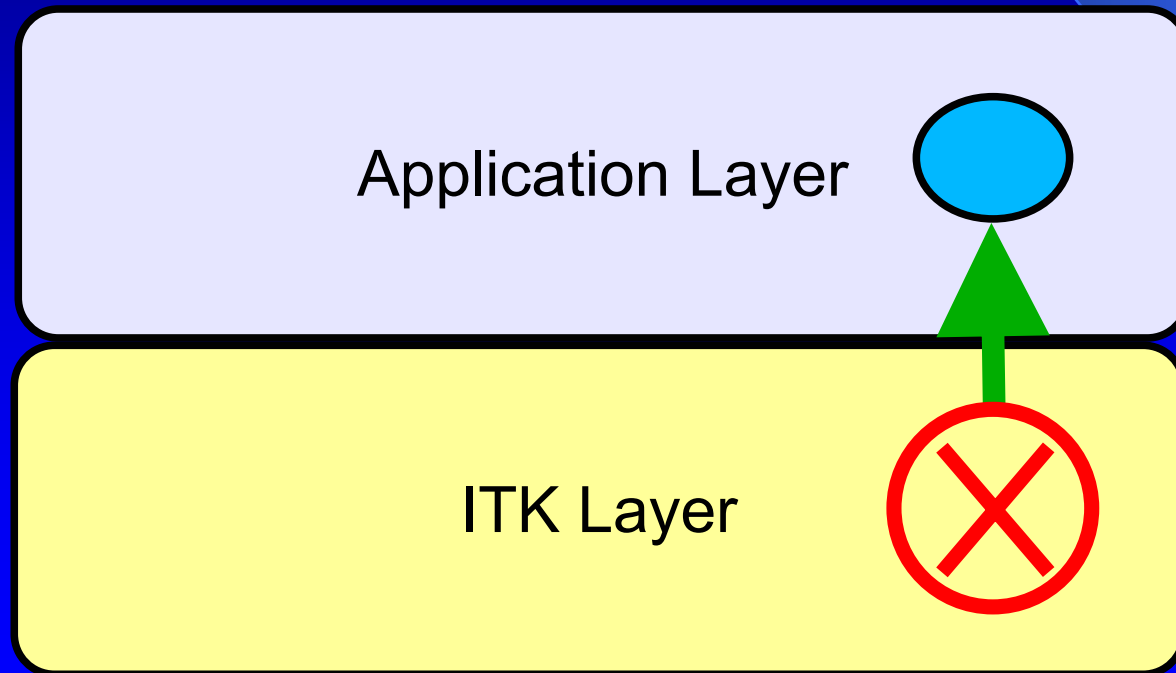
```
writer->SetFileName( "outputImage.hdr" ); // Analyze
```

```
writer->SetInput( reader->GetOutput() );
```

```
writer->Update();
```

Exceptions

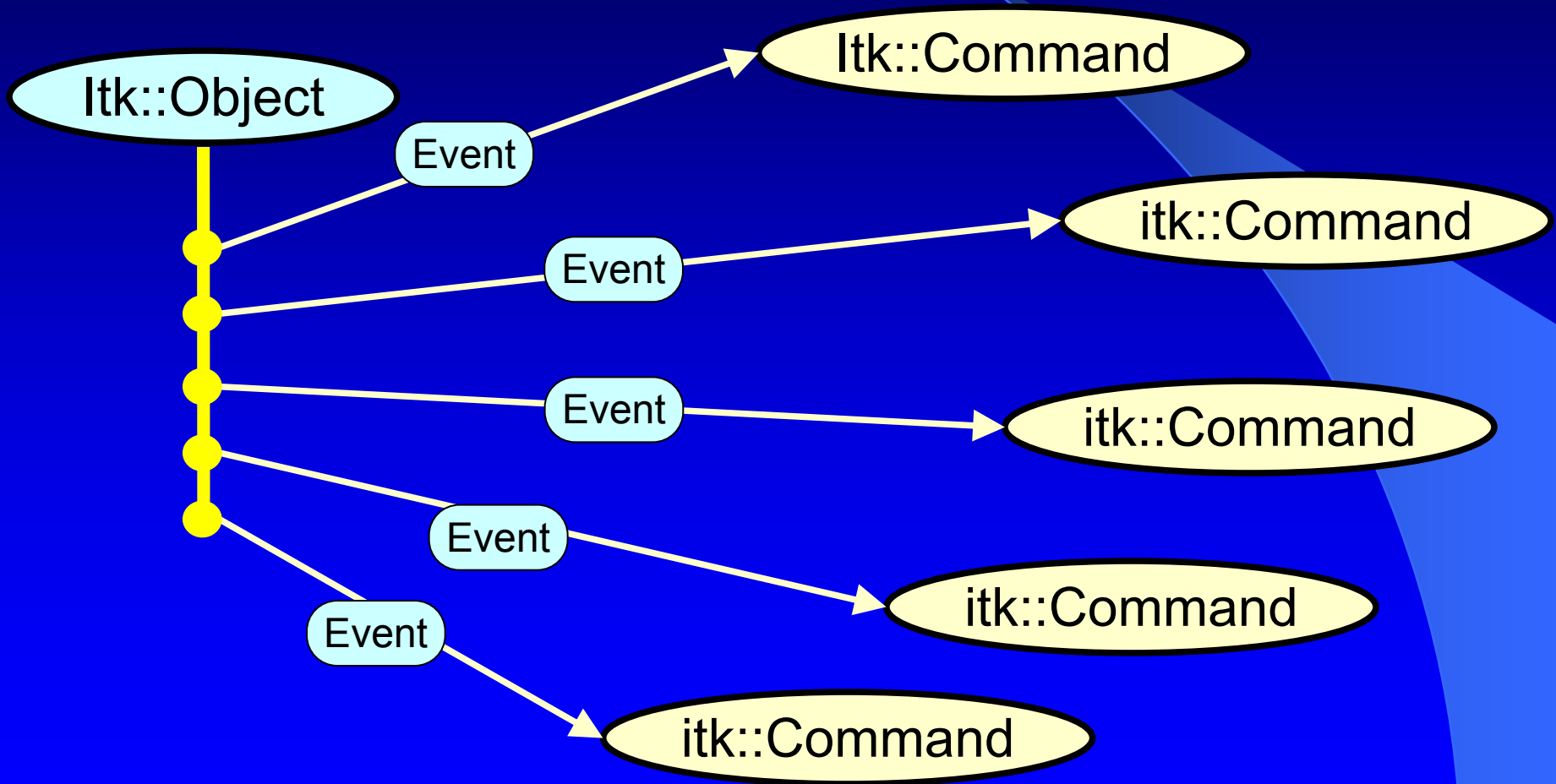
Error Management



Exceptions

```
try
{
    filter->Update();
}
catch( itk::ExceptionObject & exp )
{
    std::cout << exp << std::endl;
}
```

Events and Observers



Events and Observers

Common Events

AnyEvent()

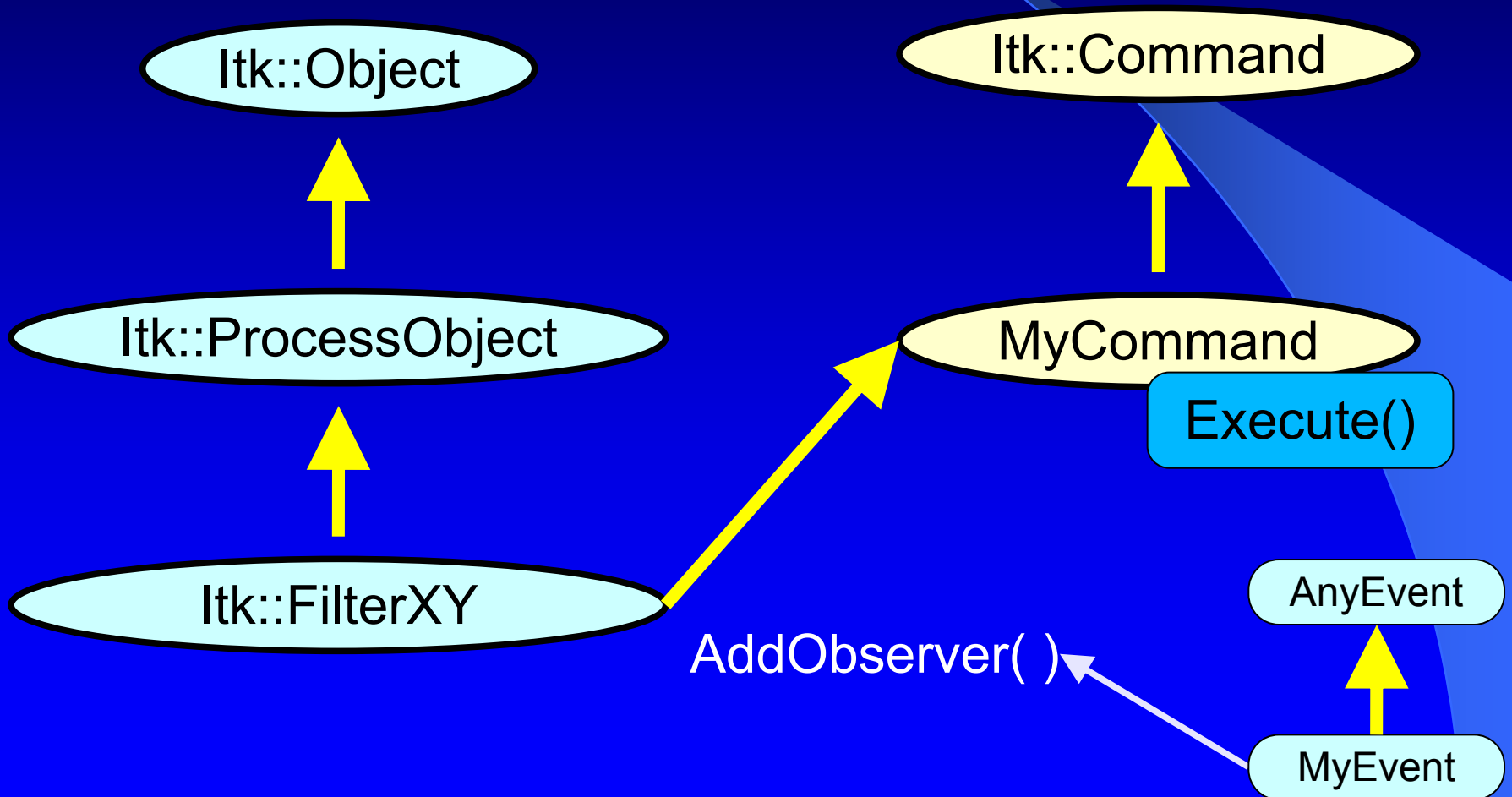
StartEvent()

EndEvent()

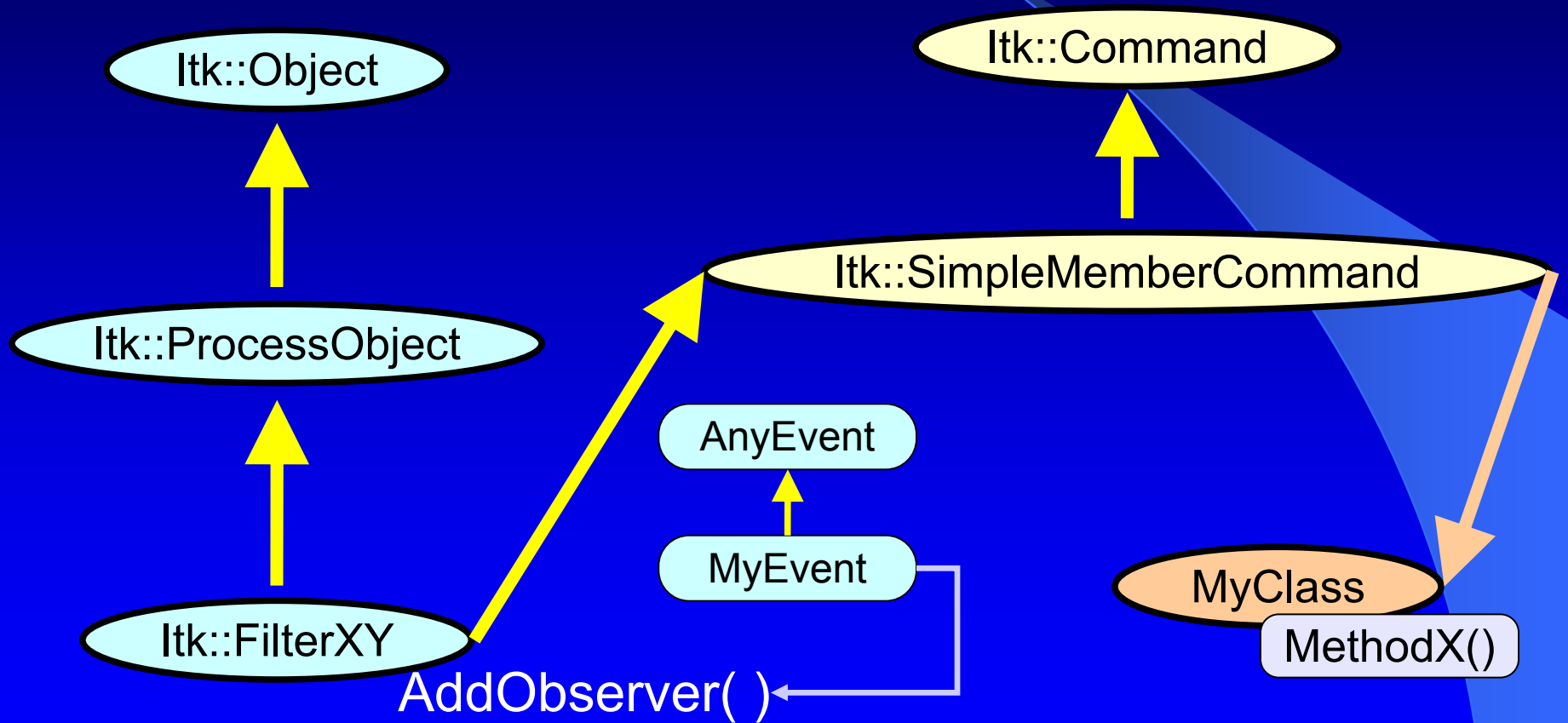
ProgressEvent()

IterationEvent()

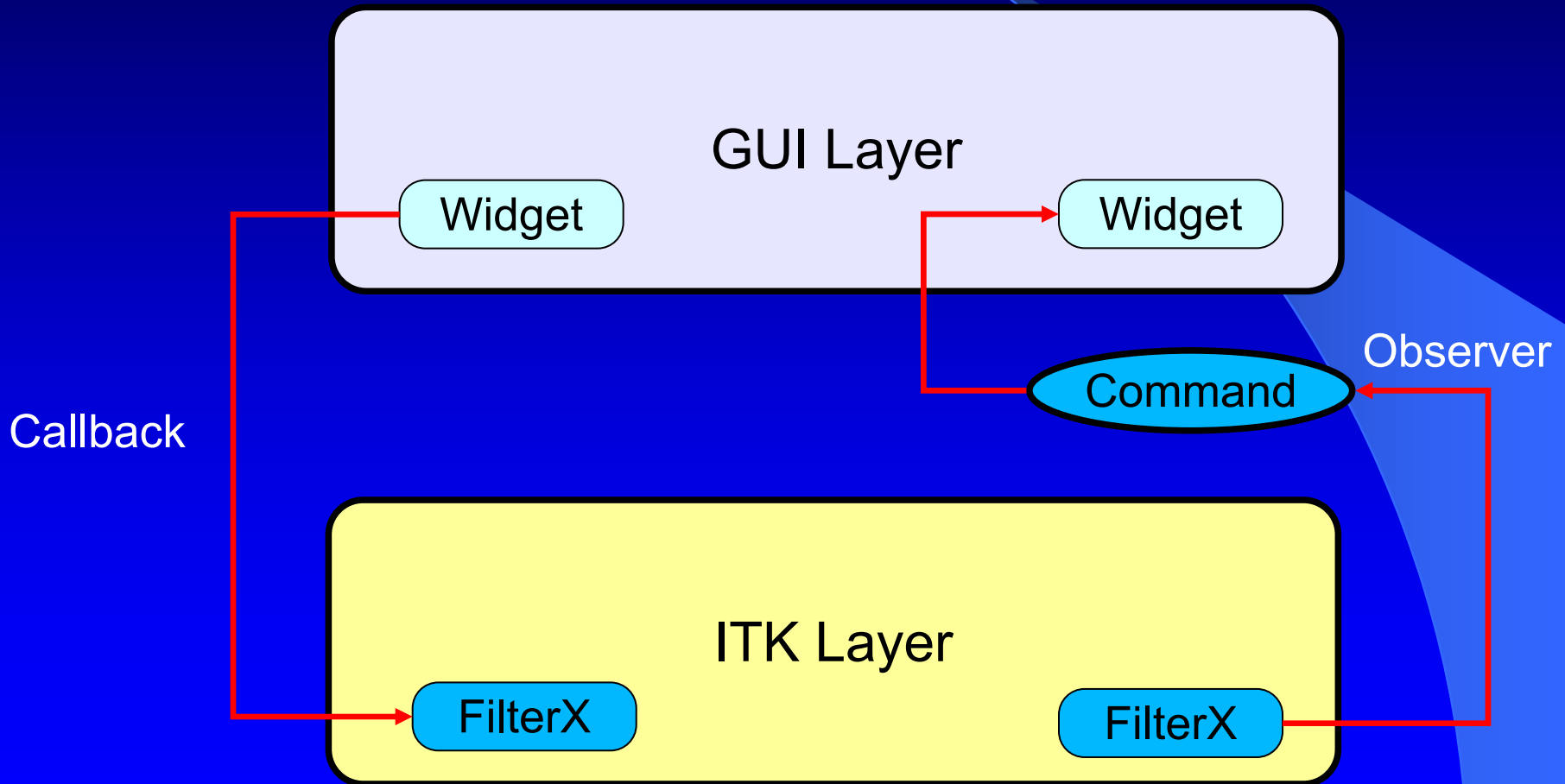
Events and Observers



Events and Observers



GUI Communication



Enjoy ITK !