4/8/2009

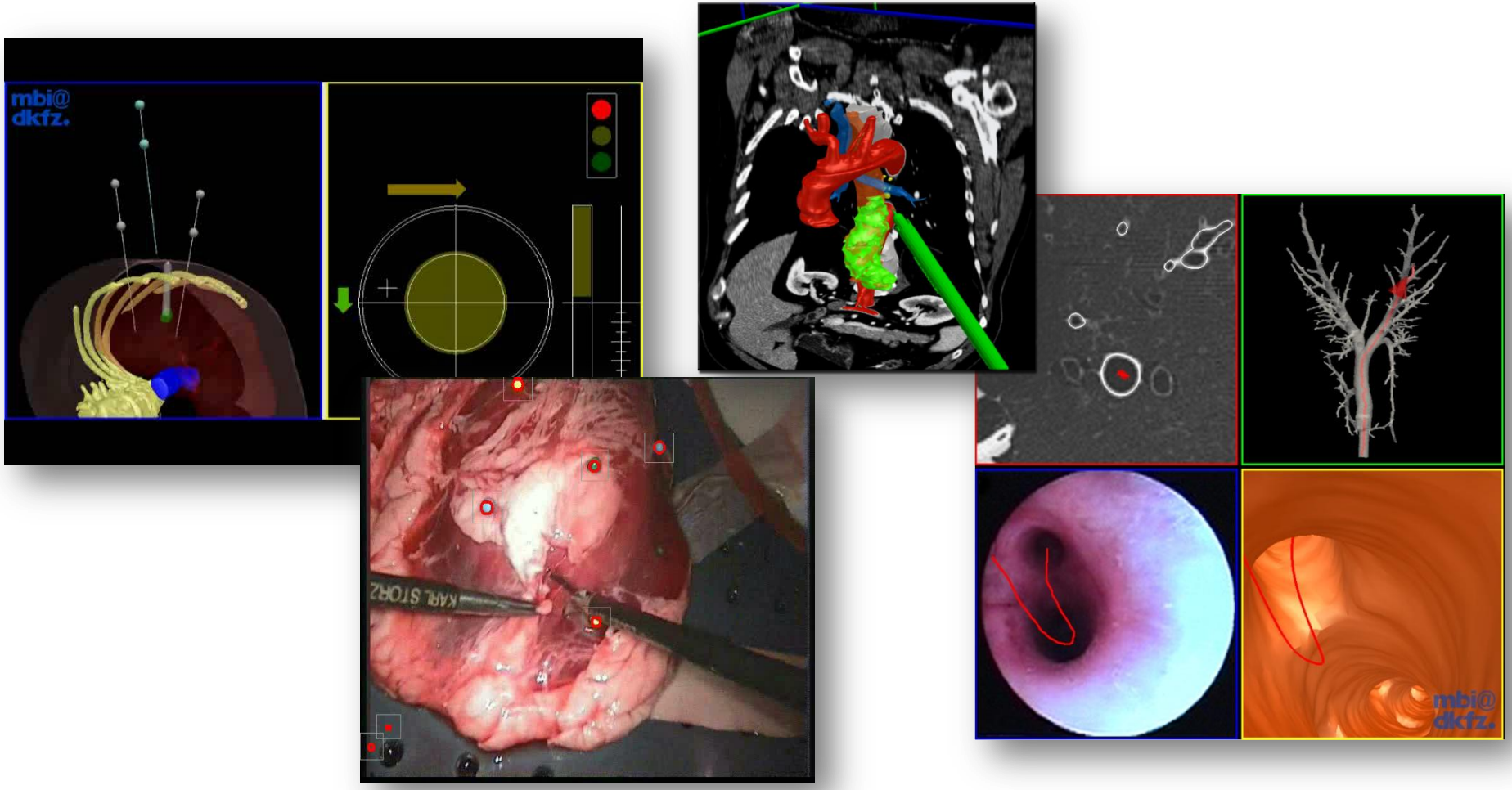
MITK-IGT

Introduction to the image guided therapy component of the Medical Imaging Interaction Toolkit

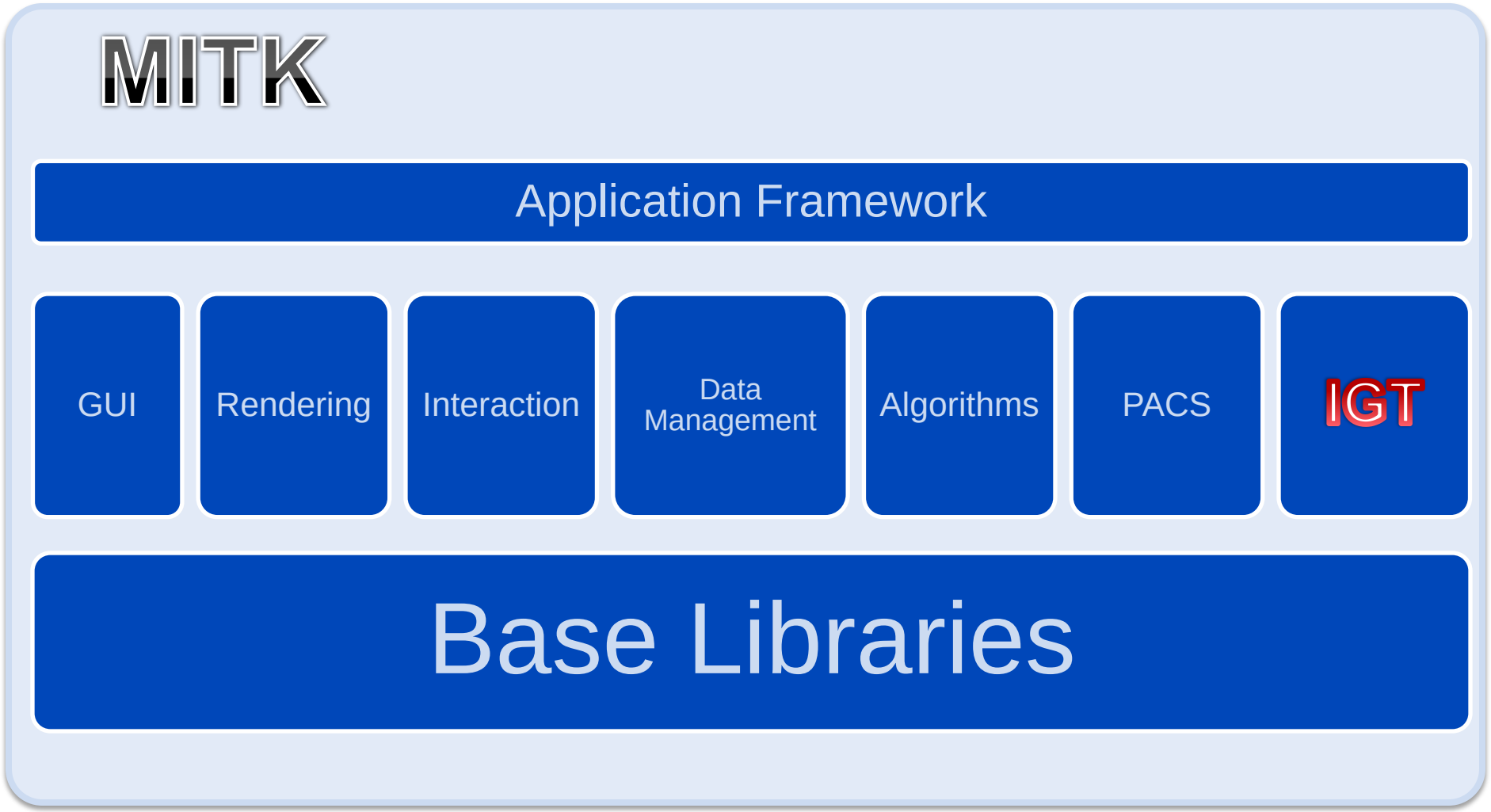
dkfz.
GERMAN
CANCER RESEARCH CENTER
IN THE HELMHOLTZ ASSOCIATION

Image Guided Therapy

Intra-intervential visualization of instruments in relation to patient using preoperative or intra operative images

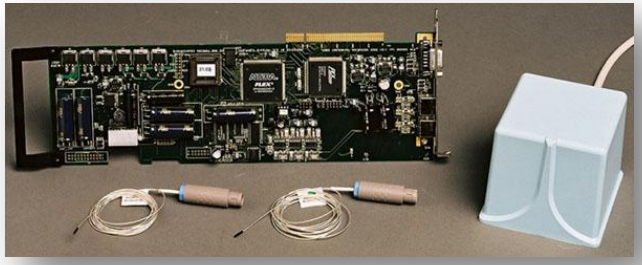


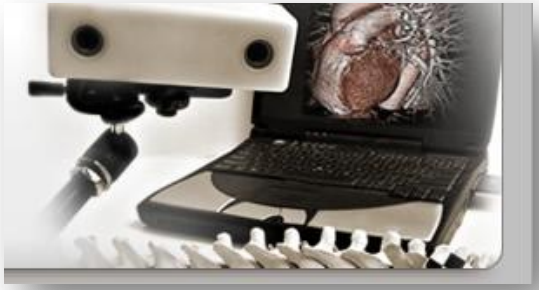
MITK Architecture



MITK-IGT: Supported Tracking Devices

- NDI Polaris
- NDI Aurora
- Claron Technologies MicronTracker (Windows only)
- Ascension Microbird (Windows only)
- Virtual tracking device

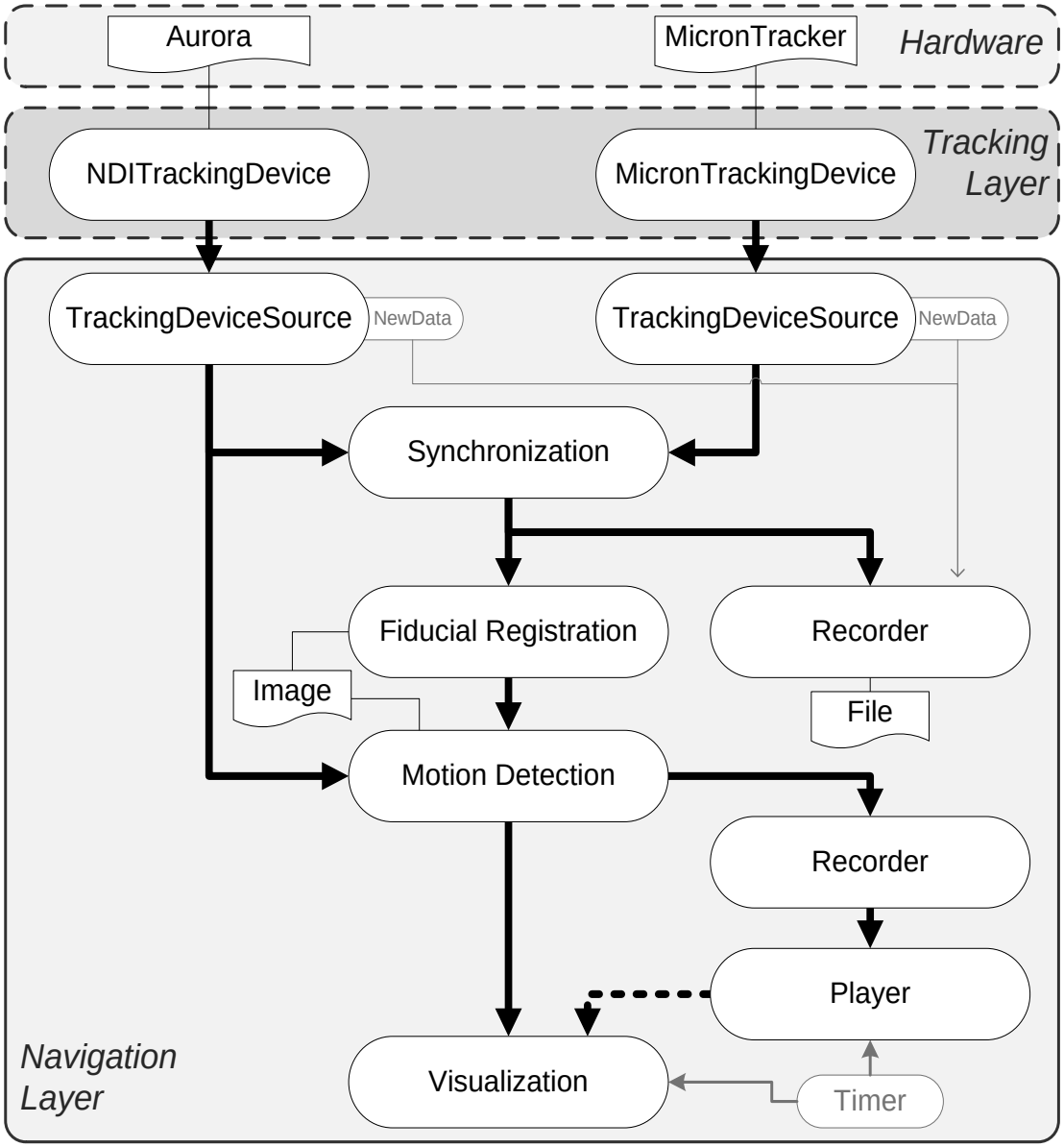




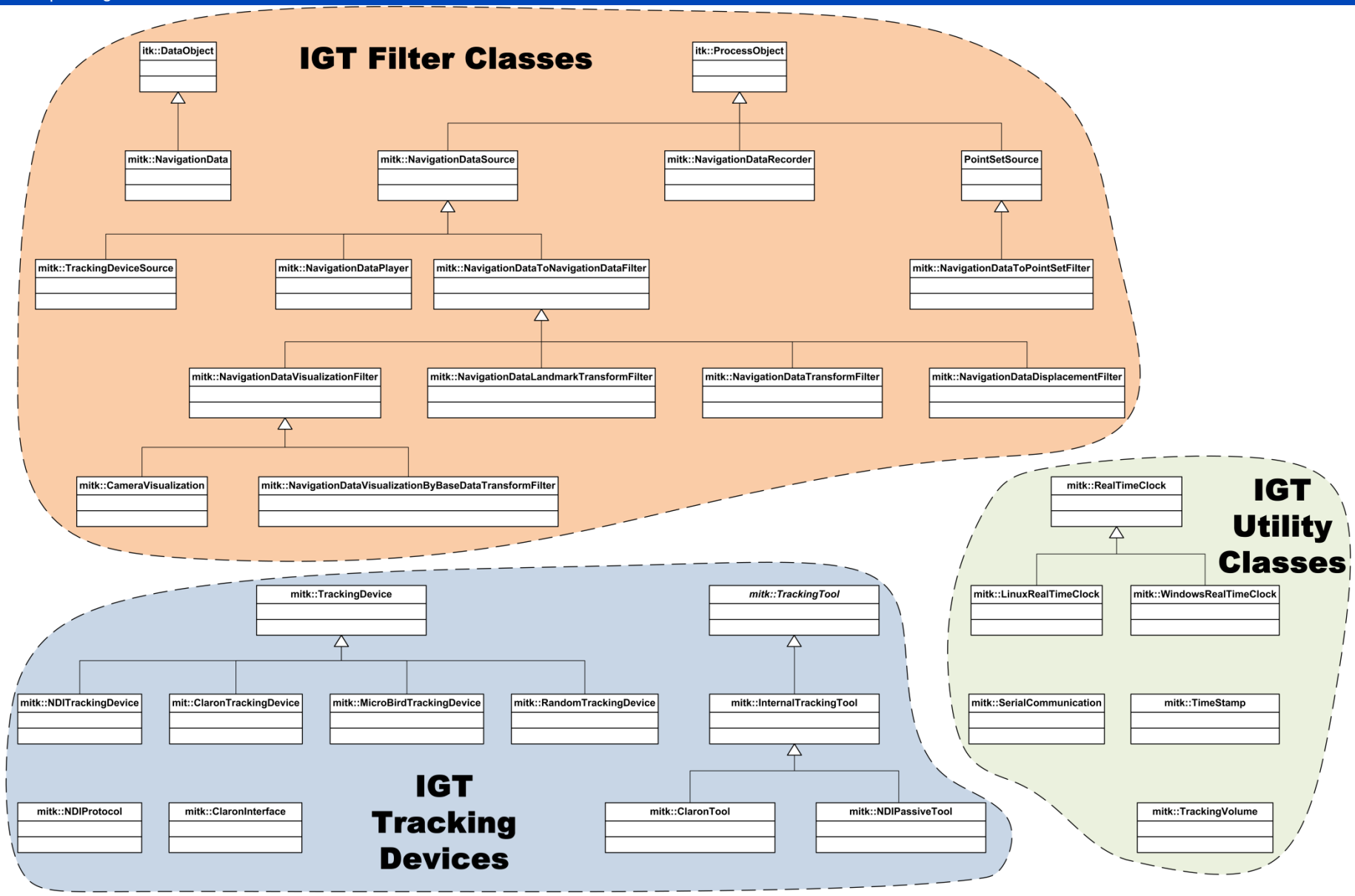
Features of MITK-IGT

- Supports Windows, Linux (and MacOS) plattforms
- Seamless integration with MITK
- Manages communication with tracking devices
- Provides Hardware abstraction of tracking devices
- Processing of tracking data with Pipeline architecture
- Ready to use modules for:
  - Recording and Replaying of tracking data
  - Coordinate transformations, including Landmark based transformation
  - Synchronization of multiple tracking devices
  - visualization of instrument movement
  - Visualization of camera movement
  - Visualization of movement trajectories
- Easily extensible

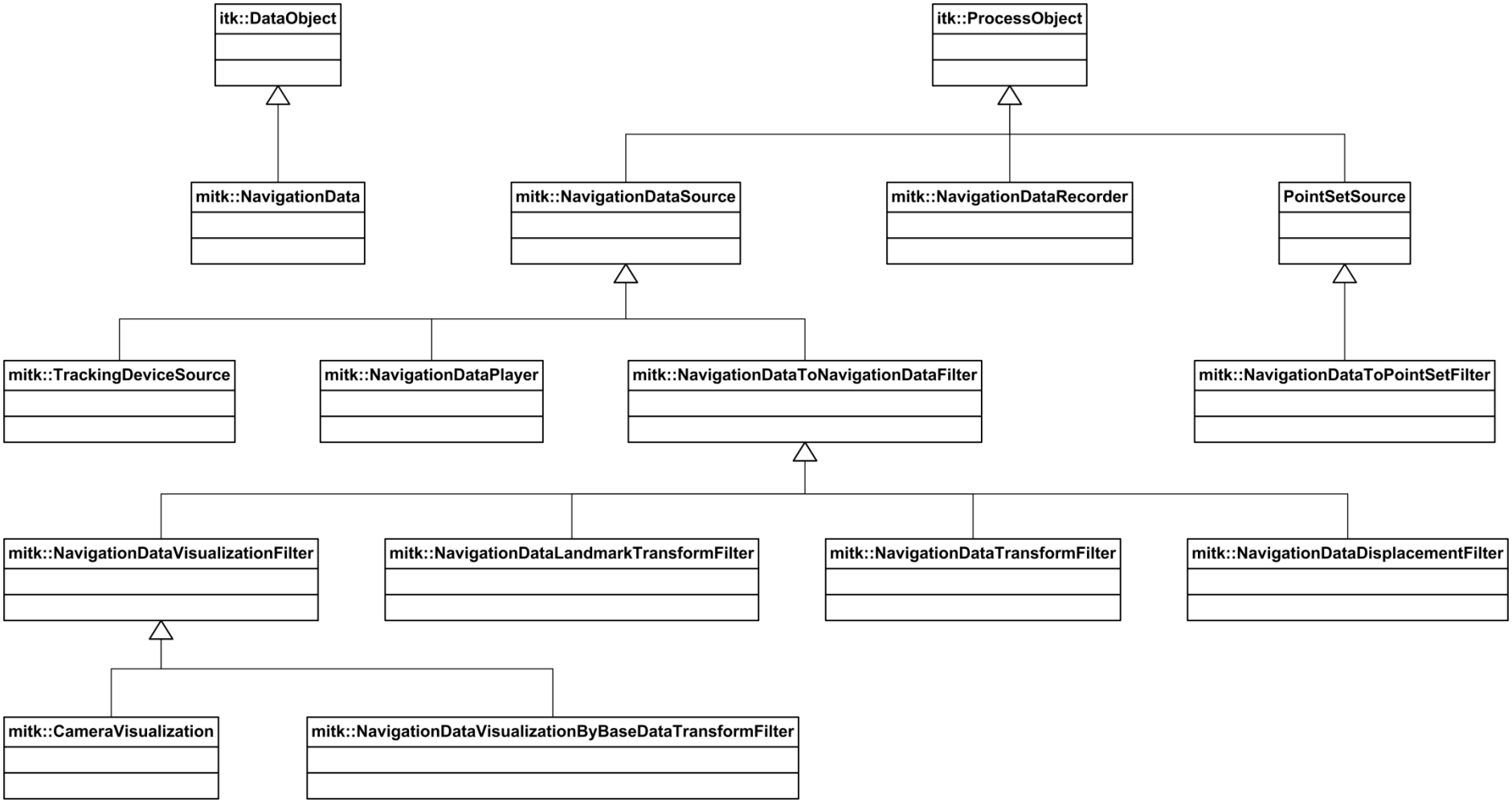
Example MITK-IGT Filter Pipeline



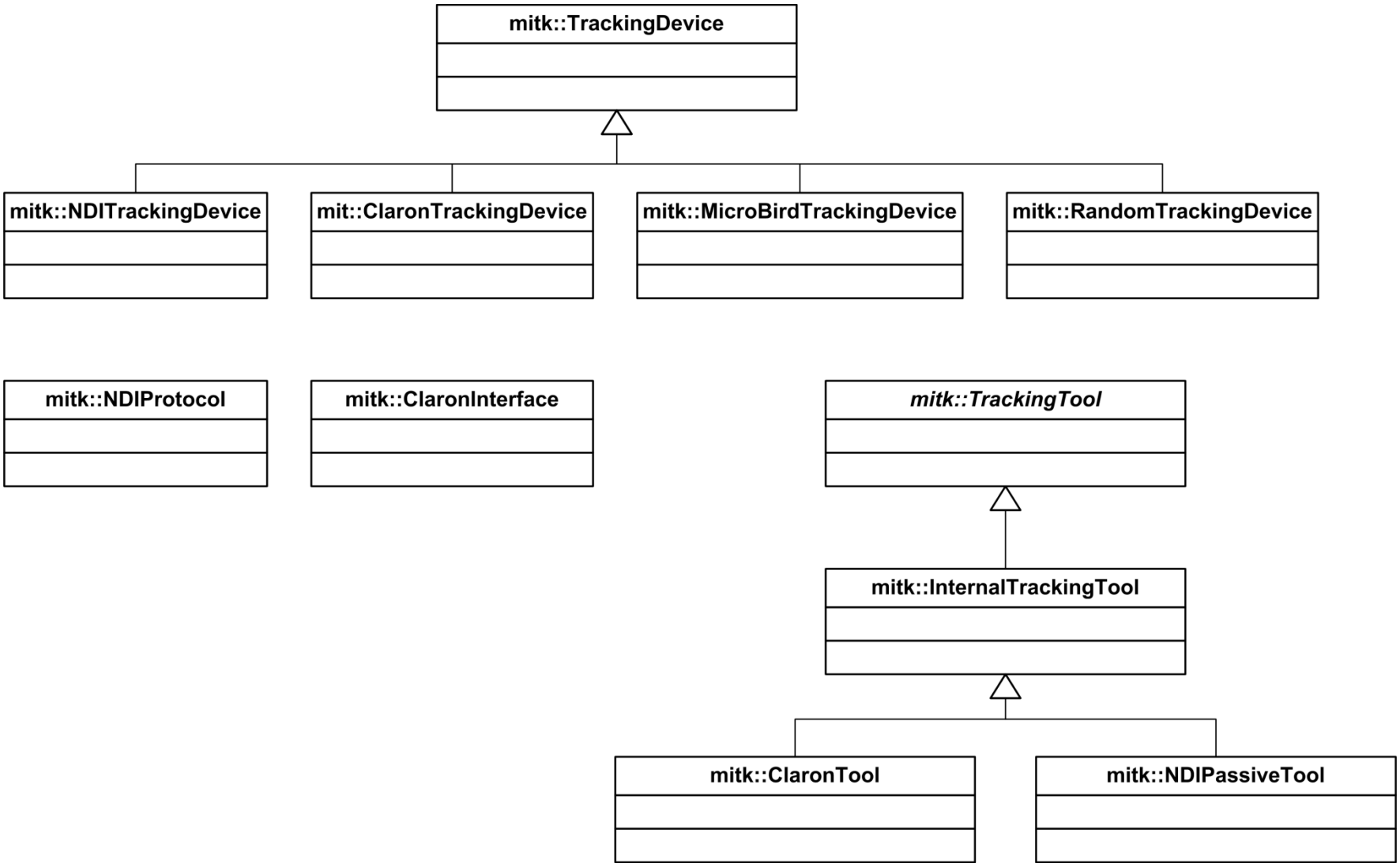
Available Classes in MITK-IGT Release 1



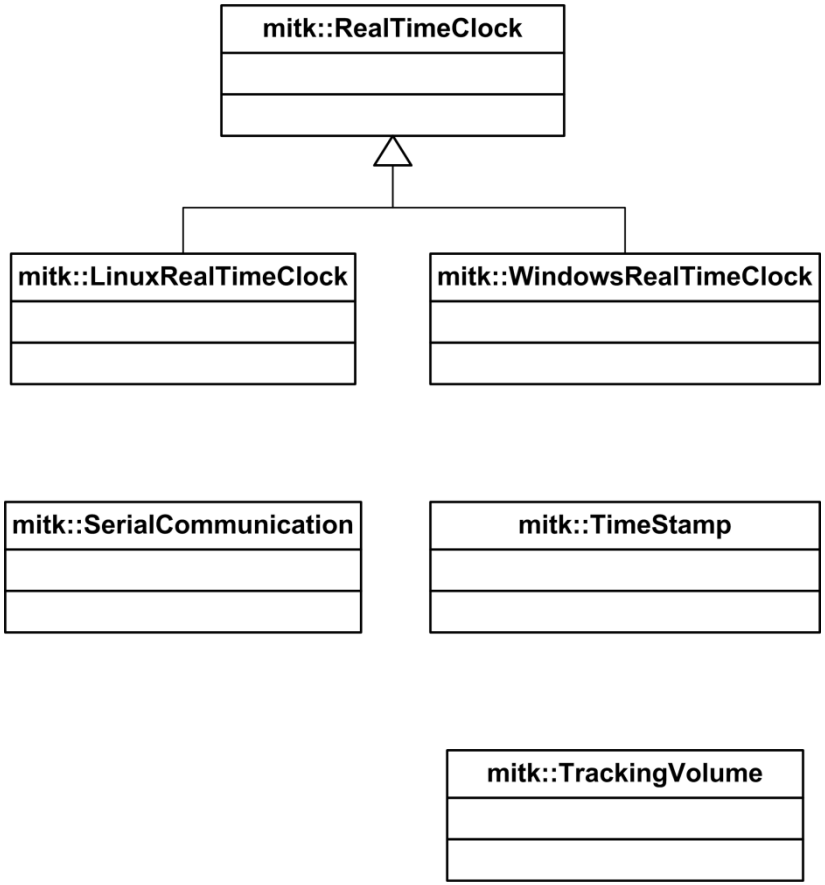
IGT Filter classes



IGT tracking device classes



IGT utility classes

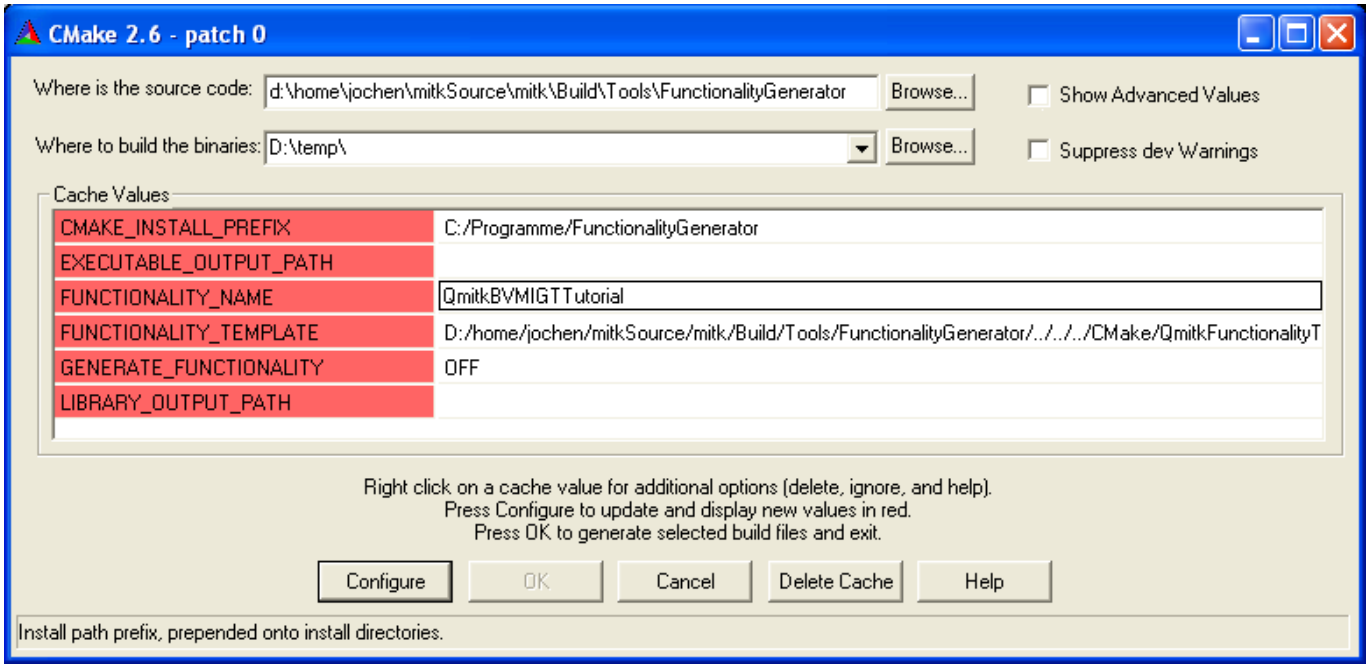


Example Image Guided Therapy Application

- Aim of the example:
  - Graphical application
  - read tracking data from tracking device
  - visualize movement of tracked object

Build an IGT application Part I

- Tutorial is based on (old) Qt3 MainApp
- Create Functionality:



- Move to /mitk/QFunctionalities/

Build an IGT application Part II

- Edit /mitk/QFunctionalities/QmitkBVMIGTTutorial/CMakeLists.txt

```
CREATE_QFUNCTIONALITY(QmitkBVMIGTTutorial)
INCLUDE_DIRECTORIES(${MITK_IGT_INCLUDE_DIRS})
LINK_LIB_TO_FUNCTIONALITY(QmitkBVMIGTTutorial mitkIGT)
```

- Activate Functionality using Cmake
  - `BUILD_QmitkBVMIGTTutorial` must be set to `ON`
  - `MITK_BUILD_IGT` must be set to `ON`
- Re-generate MITK-Project files using Cmake
- load MITK-Project in development environment

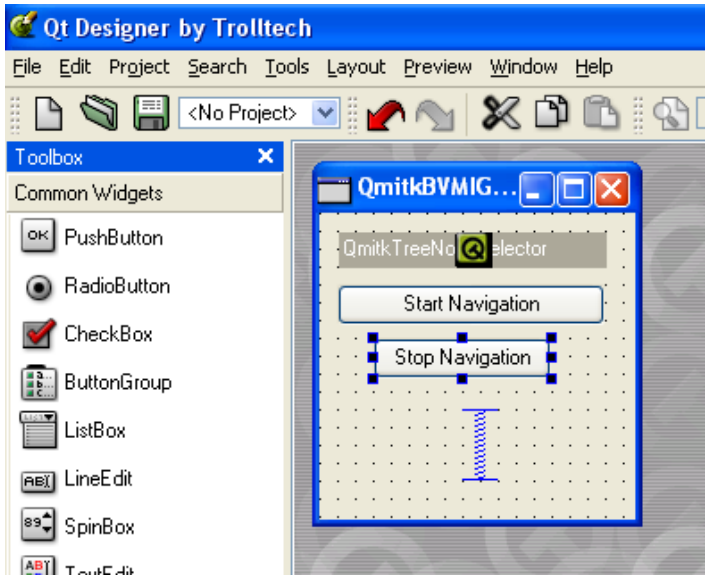
Build an IGT application Part III

- Add members to class definition:

```
class QmitkIGTTutorial : public QmitkFunctionality
{
[…]
protected slots:
  void OnSetupIGT();
  void OnTimer();
  void OnStop();
protected:
  mitk::TrackingDeviceSource::Pointer m_Source;
  mitk::NavigationDataVisualizationByBaseDataTransformFilter
   ::Pointer m_Visualizer;
  QTimer* m_Timer;
}
```

Build an IGT application Part IV

- Add Start and Stop buttons to user interface



- Connect buttons to methods

```
void QmitkBVMIGTTutorialTest::CreateConnections()
{
  if (m_Controls)
  {
    connect((QObject*)(m_Controls->m_StartButton),
            SIGNAL(clicked()),(QObject*) this, SLOT(OnSetupIGT()));
    connect((QObject*)(m_Controls->m_StopButton),
            SIGNAL(clicked()),(QObject*) this, SLOT(OnStop()));
  }
}
```

Build an IGT application Part V

- Implement Setup method: Setup tracking device

```
void QmitkIGTTutorial::OnSetupIGT()
{
  mitk::NDITrackingDevice::Pointer tracker =
                                    mitk::NDITrackingDevice::New();
  tracker->SetPortNumber(mitk::SerialCommunication::COM4);
  tracker->SetBaudRate(mitk::SerialCommunication::BaudRate115200);
  tracker->SetType(mitk::NDIPolaris);

  mitk::NDIPassiveTool::Pointer tool = mitk::NDIPassiveTool::New();
  tool->SetToolName("MyInstrument");
  tool->LoadSROMFile("c:\\myinstrument.rom");
  tracker->Add6DTool(tool);
…
```

Build an IGT application Part VI

- Implement Setup method: Setup IGT filter pipeline

```
…
m_Source = mitk::TrackingDeviceSource::New();
m_Source->SetTrackingDevice(tracker);
m_Source->Connect();

m_Visualizer =
  mitk::NavigationDataVisualizationByBaseDataTransformFilter::New();
m_Visualizer->SetInput(m_Source->GetOutput());
mitk::Cone::Pointer instrument = mitk::Cone::New();
m_Visualizer->SetBaseData(m_Source->GetOutput(), instrument);

mitk::DataTreeNode::Pointer node = mitk::DataTreeNode::New();
node->SetData(instrument);
node->SetName("My tracked object");
node->SetColor(1.0, 0.0, 0.0);
mitk::DataStorage::GetInstance()->Add(node);
…
```

Build an IGT application Part VII

- Implement Setup method: start continuous update

```
…
  m_Source->StartTracking();
  m_Timer = new QTimer(this);
  connect(m_Timer, SIGNAL(timeout()), this, SLOT(OnTimer()));
  m_Timer->start(100);
}
```

Build an IGT application Part VIII

- Implement Update method: execute continuous update

```
void QmitkIGTTutorial::OnTimer()
{
  m_Visualizer->Update();
  mitk::RenderingManager::GetInstance()->RequestUpdateAll();
}
```

- Implement Stop method: cleanup everything

```
void QmitkIGTTutorial::OnStop()
{
  m_Timer->stop();
  disconnect(m_Timer, SIGNAL(timeout()), this, SLOT(OnTimer()));
  m_Source->StopTracking();
  m_Source->Disconnect();
}
```

Build an IGT application Part IX

- Add include files to header:
- `#include "mitkTrackingDeviceSource.h"`
- `#include "mitkNavigationDataVisualizationByBaseDataTransformFilter.h"`
- `#include "qtimer.h"`

- Add include files to .cpp:

```
#include "mitkNDITrackingDevice.h"
#include "mitkNDIPassiveTool.h"
#include "mitkCone.h"
```

Build an IGT application Part X

- Compile
- Execute
- Navigate!

Thank you

- MITK-IGT will be available for download within the next days at:

```
svn co http://svn.mitk.org/
```

- A Software demo will be held at **Monday 16:40** in the Foyer
- More documentation is available:
  - BVM Proceedings Paper **1399**: *MITK-IGT: Eine Navigationskomponente für das Medical Imaging Interaction Toolkit*
  - MITK Documentation: http://mitk.org/wiki/Documentation
  - 2 Tutorials and an example application module available:
    - */mitk/Core/IGT/IGTTutorial/IGTTutorialStep1.cpp*
    - */mitk/Qfunctionalities/QmitkIGTTutorial*
    - */mitk/Qfunctionalities/QmitkIGTExample*