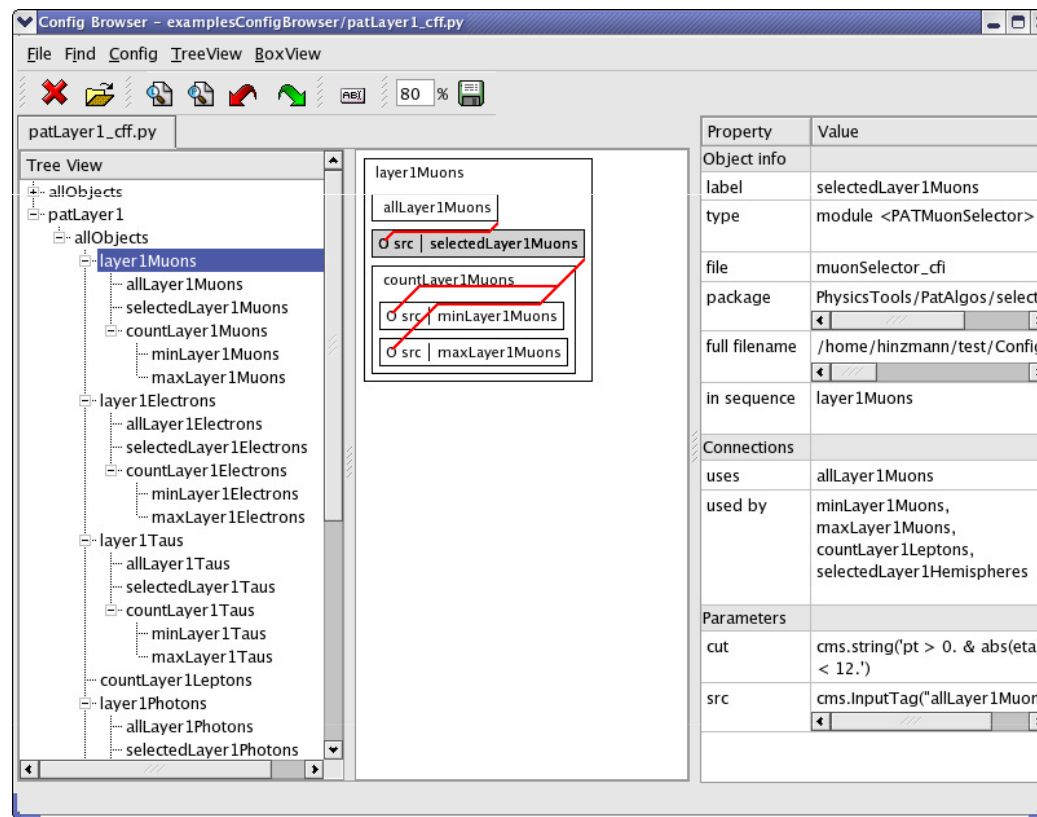


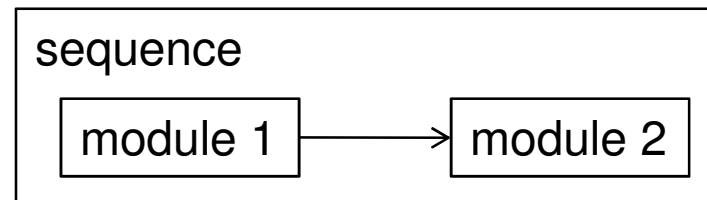
Visualization of the CMS Python Configuration System



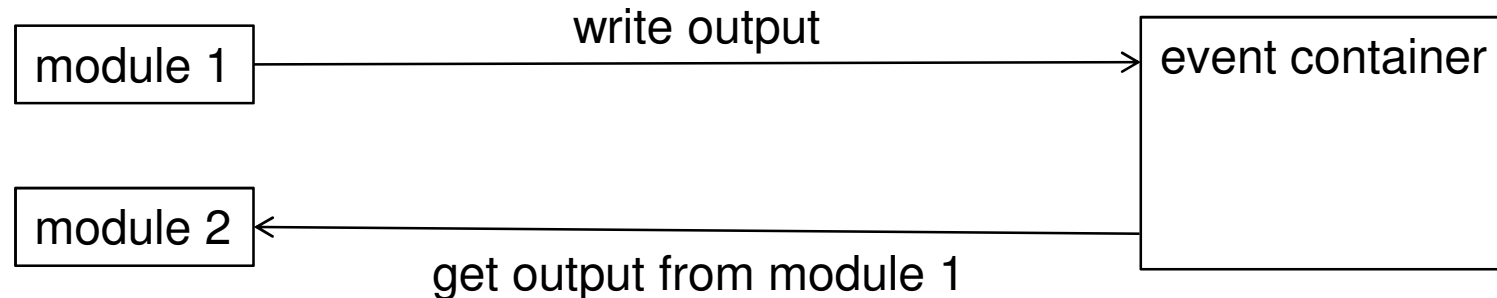
Andreas Hinzmann, O. Actis, M. Erdmann, R. Fischer,
T. Klimkovich, G. Müller, J. Steggemann (RWTH Aachen University),
B. Hegner (CERN)

CMS software framework

- CMS has **common software framework** for simulation, high level trigger, reconstruction and analysis
- **Modular FW:** Modules organized in sequences, defining order of execution



- **Information flow** between modules only via a data container:

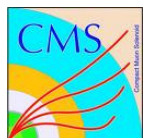


CMS python configuration system

- **Python configuration** system used for **steering of the C++ based FW**
- Python configuration
 - **defines the sequences** and the **parameters** of the modules
 - defines the setup: geometry, conditions, etc.
 - in principle contains all information on the processing flow except for the algorithms
 - **spread over many files connected via python import**

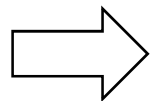
```
import FWCore.ParameterSet.Config as cms
from PhysicsTools.PatAlgos.producers.muonProducer_cfi import *
from PhysicsTools.PatAlgos.selection.muonSelector_cfi import *
from PhysicsTools.PatAlgos.selection.muonCountFilter_cff import *

Muons = cms.Sequence(allMuons * selectedMuons * countMuons)
```



Aims of visualization

- Problems:
 - Configuration spread over many files
 - difficult to find/edit certain modules/parameters
 - Indirect information flow between modules
 - difficult to understand dependencies
- Requirements for a browsing tool for the configuration system
 - Give **quick but detailed overview** over simulation / trigger / reconstruction / analysis
 - Allow **debugging / verification** of configuration files
 - Allow **documentation** of simulation / trigger / reconstruction / analysis



CMS Configuration Browser

Key functionality

- Visualization of the **sequence structure**
- Visualization of the **information flow** between modules
- **Locating module definitions** in configuration files
- **Interactive editing** of the configuration
- **Exporting graphical representations** of the configuration structure

CMS Configuration Browser

1. Open a config file
(and automatically all files imported by it)

The screenshot shows the CMS Configuration Browser interface. The main window displays a tree view of configuration files. A callout box points to the 'File' menu icon, indicating the first step: '1. Open a config file (and automatically all files imported by it)'. The tree view shows a hierarchy of files, including 'allLayer1Muons', 'selectedLayer1Muons', 'countLayer1Muons', 'minLayer1Muons', 'maxLayer1Muons', and similar structures for Electrons, Taus, and Photons. The 'selectedLayer1Muons' file is selected, and its details are shown in the right-hand pane. The details pane includes a table of properties and values, and a section for connections and parameters.

Property	Value
Object info	
label	selectedLayer1Muons
type	module <PATMuonSelector>
file	muonSelector_cfi
package	PhysicsTools/PatAlgos/selectic
full filename	/home/hinzmann/test/Configt
in sequence	layer1Muons
Connections	
uses	allLayer1Muons
used by	minLayer1Muons, maxLayer1Muons, countLayer1Leptons, selectedLayer1Hemispheres
Parameters	
cut	cms.string('pt > 0. & abs(eta) < 12.')
src	cms.InputTag("allLayer1Muons

CMS Configuration Browser

The screenshot shows the CMS Configuration Browser interface. The window title is "Config Browser - examplesConfigBrowser/patLayer1_cff.py". The menu bar includes "File", "Find", "Config", "TreeView", and "BoxView". The toolbar contains icons for file operations and a zoom level of 80%. The main area is divided into three panes:

- Tree View:** A hierarchical tree of configuration objects. The "layer1Muons" object is selected and highlighted in blue. It contains sub-objects like "allLayer1Muons", "selectedLayer1Muons", "countLayer1Muons", "minLayer1Muons", "maxLayer1Muons", and similar objects for "Electrons" and "Taus".
- Object View:** A detailed view of the selected "layer1Muons" object. It shows a diagram of the object's structure with red arrows pointing to "selectedLayer1Muons", "countLayer1Muons", "minLayer1Muons", and "maxLayer1Muons".
- Property View:** A table showing the properties and values of the selected object.

Property	Value
Object info	
label	selectedLayer1Muons
type	module <PATMuonSelector>
file	muonSelector_cfi
package	PhysicsTools/PatAlgos/selectic
full filename	/home/hinzmann/test/Config8
in sequence	layer1Muons
Connections	
uses	allLayer1Muons
used by	minLayer1Muons, maxLayer1Muons, countLayer1Leptons, selectedLayer1Hemispheres
Parameters	
cut	cms.string('pt > 0. & abs(eta) < 12.')
src	cms.InputTag("allLayer1Muons

2. Browse sequence and module content

CMS Configuration Browser

The screenshot shows the CMS Configuration Browser interface. On the left, a 'Tree View' displays a hierarchy of configuration objects under 'patLayer1'. The 'layer1Muons' object is selected. In the center, a diagram visualizes the structure of a sequence, showing 'selectedLayer1Muons' as the source for 'countLayer1Muons', which in turn sources 'minLayer1Muons' and 'maxLayer1Muons'. On the right, a 'Property Value' table provides details for the selected object.

Property	Value
Object info	
label	selectedLayer1Muons
type	module <PATMuonSelector>
file	muonSelector_cfi
package	PhysicsTools/PatAlgos/selectic
full filename	/home/hinzmann/test/Configt
in sequence	layer1Muons
Connections	
uses	allLayer1Muons
used by	minLayer1Muons, maxLayer1Muons, countLayer1Leptons, selectedLayer1Hemispheres
Parameters	
cut	cms.string('pt > 0. & abs(eta) < 12.')
src	cms.InputTag("allLayer1Muons

3. Visualize structure of a sequence

CMS Configuration Browser

The screenshot shows the CMS Configuration Browser interface. On the left, a 'Tree View' displays a hierarchy of configuration objects under 'patLayer1'. The 'layer1Muons' object is selected. In the center, a preview window shows the structure of 'layer1Muons', with 'selectedLayer1Muons' highlighted. On the right, a 'Property Value' table provides details for the selected module.

Property	Value
Object info	
label	selectedLayer1Muons
type	module <PATMuonSelector>
file	muonSelector_cfi
package	PhysicsTools/PatAlgos/selectic
full filename	/home/hinzmann/test/Configf
in sequence	layer1Muons
Connections	
uses	allLayer1Muons
used by	minLa maxL count select
Parameters	
cut	cms.string('pt > 0. & abs(eta) < 12.')
src	cms.InputTag("allLayer1Muons

4. Inspect parameters of selected module

CMS Configuration Browser

The screenshot shows the CMS Configuration Browser interface. The title bar reads 'Config Browser - examplesConfigBrowser/patLayer1_cff.py'. The menu bar includes 'File', 'Find', 'Config', 'TreeView', and 'BoxView'. The toolbar contains icons for file operations and a zoom level of 80%. The main area is divided into three panes:

- Tree View:** A hierarchical tree of configuration objects. The path 'patLayer1 > allObjects > layer1Muons' is selected and highlighted in blue.
- Object Info:** A diagram showing the internal structure of the selected 'layer1Muons' object. It contains sub-objects: 'allLayer1Muons', 'selectedLayer1Muons' (highlighted with a red box), 'countLayer1Muons', 'minLayer1Muons', and 'maxLayer1Muons'. Red lines indicate connections between these sub-objects.
- Property Table:** A table displaying the properties of the selected 'selectedLayer1Muons' module.

Property	Value
Object info	
label	selectedLayer1Muons
type	module <PATMuonSelector>
file	muonSelector_cfi
package	PhysicsTools/PatAlgos/selectic
full filename	/home/hinzmann/test/Config
in sequence	layer1M
Connections	
uses	allLayer
used by	minLayer1Muons, maxLayer1Muons, countLayer1Leptons, selectedLayer1Hemispheres
Parameters	
cut	cms.string('pt > 0. & abs(eta) < 12.')
src	cms.InputTag("allLayer1Muons

A callout box with the text "5. Locate config file of selected module" points to the 'full filename' property in the table.

CMS Configuration Browser

The screenshot shows the CMS Configuration Browser window titled "Config Browser - examplesConfigBrowser/patLayer1_cff.py". The interface includes a menu bar (File, Find, Config, TreeView, BoxView), a toolbar with icons for file operations and zooming (80%), and a tree view on the left. The tree view shows a hierarchy of objects under "patLayer1_cff.py", including "layer1Muons" and its sub-objects like "selectedLayer1Muons", "countLayer1Muons", "minLayer1Muons", and "maxLayer1Muons". A callout box with the text "6. Open config file of selected module for editing" points to the "minLayer1Muons" object in the tree. The right pane displays a table of properties for the selected object.

Property	Value
Object info	
label	selectedLayer1Muons
type	module <PATMuonSelector>
file	muonSelector_cfi
package	PhysicsTools/PatAlgos/selectic
full filename	/home/hinzmann/test/Config8
in sequence	layer1Muons
Connections	
uses	allLayer1Muons
used by	minLayer1Muons, maxLayer1Muons, countLayer1Leptons, selectedLayer1Hemispheres
Parameters	
cut	cms.string('pt > 0. & abs(eta) < 12.')
src	cms.InputTag("allLayer1Muons

CMS Configuration Browser

The screenshot shows the CMS Configuration Browser interface. The title bar reads "Config Browser - examplesConfigBrowser/patLayer1_cff.py". The menu bar includes "File", "Find", "Config", "TreeView", and "BoxView". The toolbar contains various icons, including a save icon, and a zoom level of 80%. A callout box with an arrow pointing to the save icon contains the text "7. Save graphic to file".

The interface is divided into three main sections:

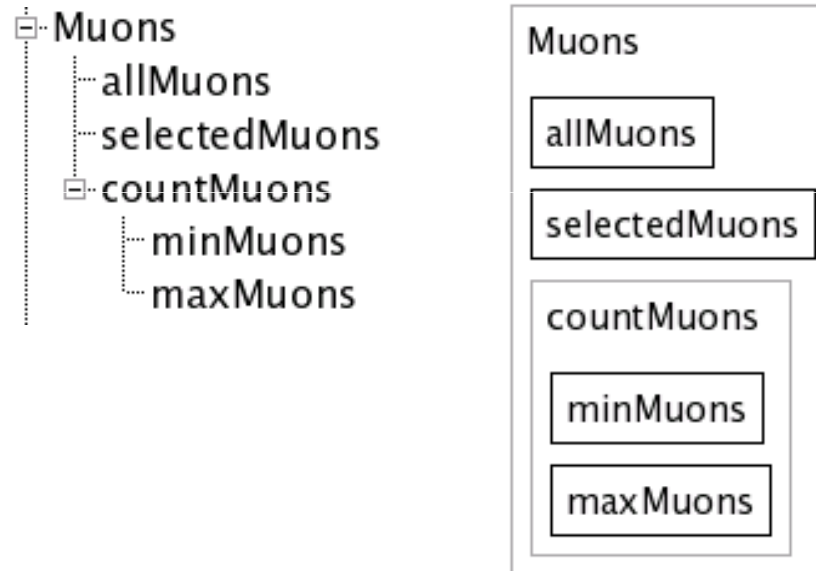
- Tree View:** A hierarchical tree structure on the left. The selected path is `patLayer1 > allObjects > layer1Muons`. Other visible nodes include `allLayer1Muons`, `selectedLayer1Muons`, `countLayer1Muons`, `minLayer1Muons`, `maxLayer1Muons`, and similar structures for Electrons, Taus, and Photons.
- Diagram:** A central diagram showing the relationships between objects. A box labeled "layer1Muons" contains sub-objects: "allLayer1Muons", "selectedLayer1Muons" (highlighted with a red circle), "countLayer1Muons", "minLayer1Muons" (highlighted with a red circle), and "maxLayer1Muons" (highlighted with a red circle). Red lines indicate connections between these objects.
- Property Table:** A table on the right showing the configuration details for the selected object.

Property	Value
Object info	
label	selectedLayer1Muons
type	module <PATMuonSelector>
file	muonSelector_cfi
package	PhysicsTools/PatAlgos/selectic
full filename	/home/hinzmann/test/Config8
in sequence	layer1Muons
Connections	
uses	allLayer1Muons
used by	minLayer1Muons, maxLayer1Muons, countLayer1Leptons, selectedLayer1Hemispheres
Parameters	
cut	cms.string('pt > 0. & abs(eta) < 12.')
src	cms.InputTag("allLayer1Muons

Visualization of sequence structure

- Sequence structure of a config file visualized

- in form of a tree
- as a block diagram



```
countMuons = cms.Sequence(minMuons * maxMuons)

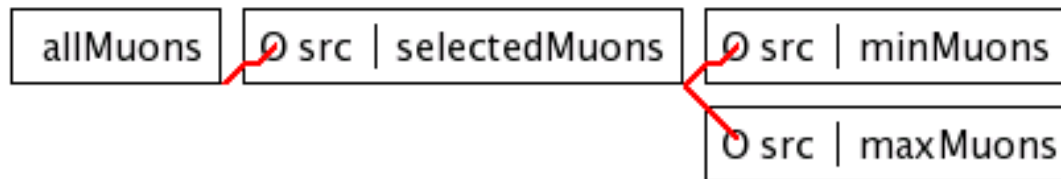
Muons = cms.Sequence(allMuons * selectedMuons * countMuons)
```

- Sequence structure visualizes the **full chain of modules** in a config file

Visualization of information flow

- Input from other modules visualized by **connecting lines**

- Arrange modules according to information flow

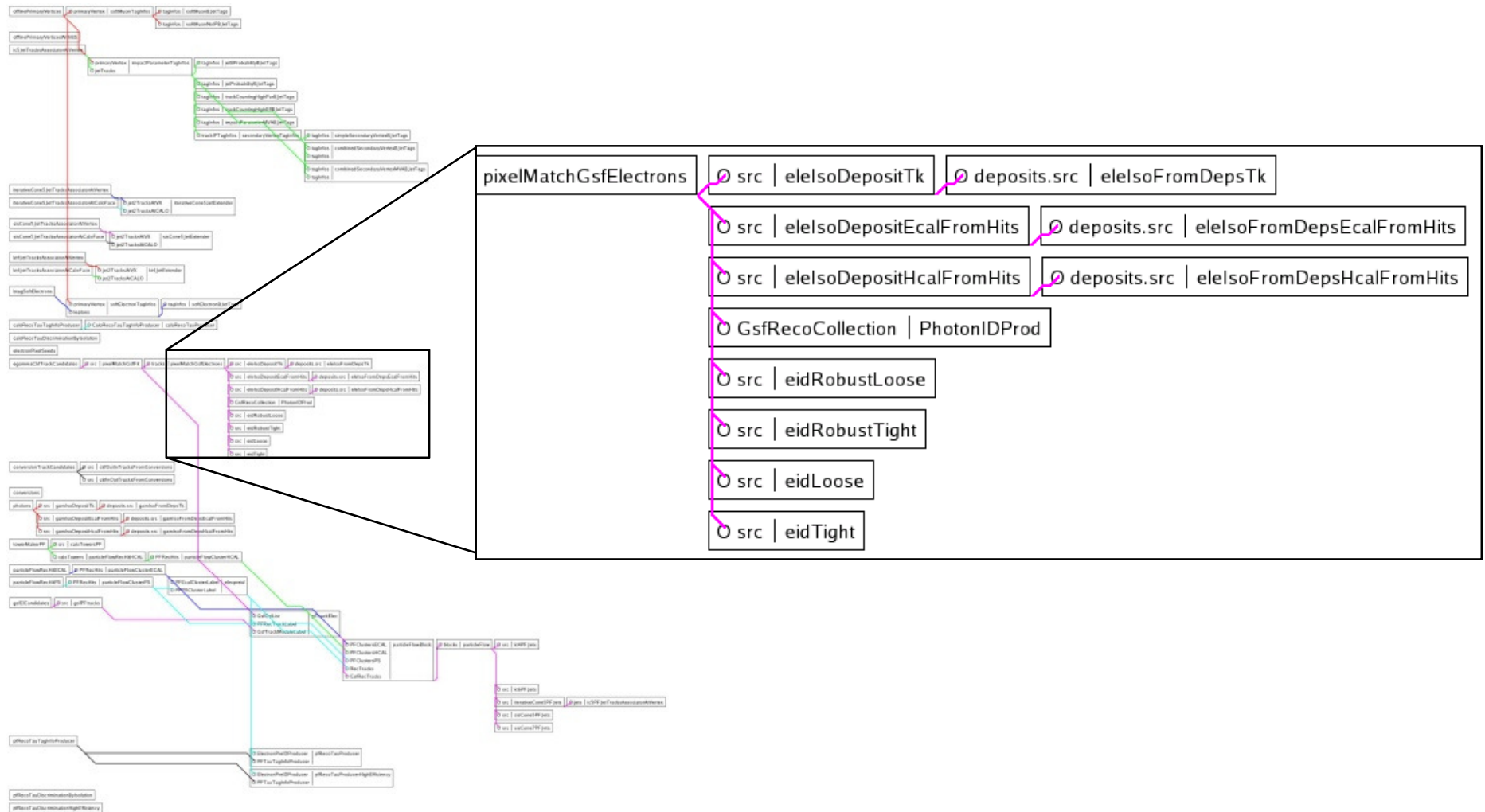


- Modules which depend on the output of selected module **listed in properties** →
- Allows user to understand the dependencies between modules

Property	Value
Object info	
label	selectedMuons
type	EDFilter <PATMuonSelector>
file	muonSelector_cfi : 10
package	PhysicsTools/PatAlgos/se ◀ ▶
full filename	/.automount/home/home ◀ ▶
in sequence	Muons
Connections	
uses	allMuons
used by	minMuons, maxMuons
Parameters	
cut	cms.string('pt > 0. & abs(eta) < 12.')
src	cms.InputTag("allMuons") ◀ ▶

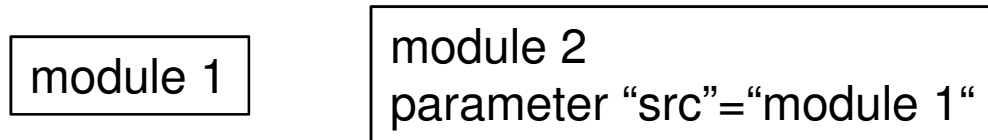
A more sophisticated example

- Graphical representation of the high-level reconstruction in CMS

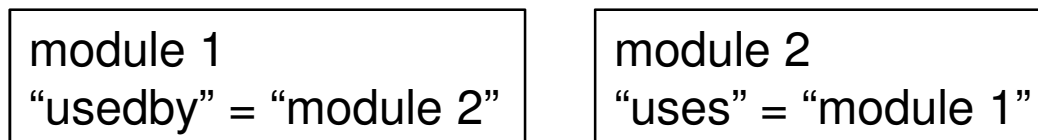


Information in the configuration

- Sequence structure is contained **directly** in python configuration
- Information flow structure however contained only **unidirectional**
 - Input from another module is indicated in the parameters of module by the name of module whose output is used

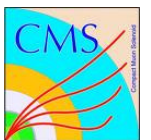


- Names are mapped to the modules themselves
- Entire configuration is scanned in order to calculate "usedby" dependency



The underlying software concept

- Configuration Browser programmed in Python, using Qt for the GUI
- Python:
 - Python configuration can be **directly analyzed via import and introspection**
 - **Fast development turnaround** due to dynamic typing, automatic memory management
- Qt:
 - Widely spread cross-platform application development framework
 - Other usage in CMS (Iguana event display)



The underlying software concept

- Configuration Browser is a plug-in of an underlying software framework
 - **designed for maximum reusability**
- New browsers / editors can be added to the application using a **plug-in mechanism**
- Plug-ins **share graphical compounds**: tree view, center view, properties view, etc.
- Example for further plug-ins:
 - Event browser for the CMS data files (EdmBrowser)
Browses entire content of CMS data files event by event using introspection
- Common code of the two browsers 2000 lines, individual 1000 each

Summary

- Key features of CMS Configuration Browser
 - Visualization of the sequence structure and information flow of CMS configuration files
 - Interactive editing and documentation of the configuration
 - Implementation in Python gives direct access to the configuration
 - GUI designed for maximum reusability

