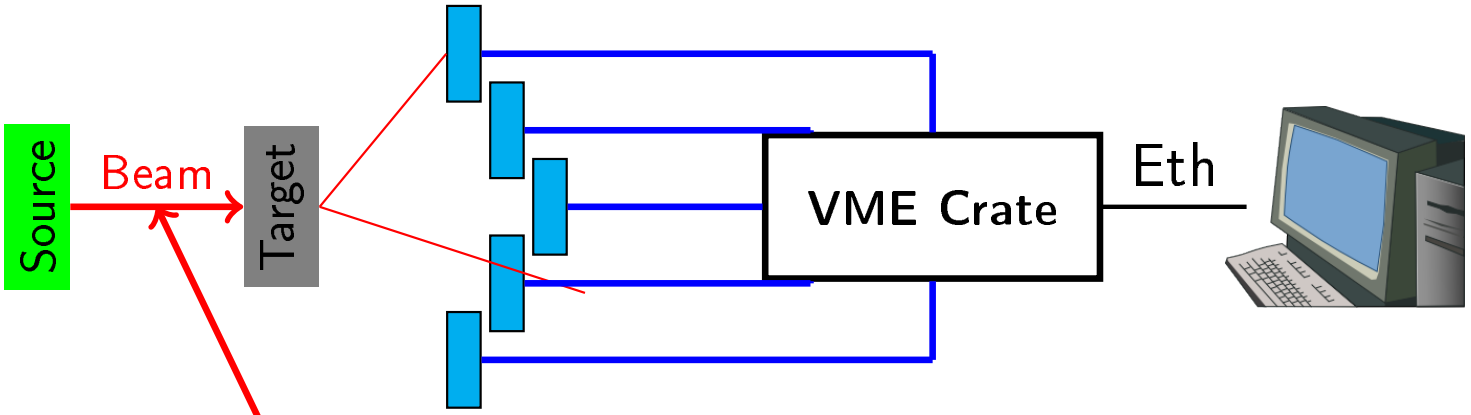
ANSiP 2011 - Narval framework

Xavier Grave

On behalf of Narval team
CSNSM - GANIL - **IPNO**

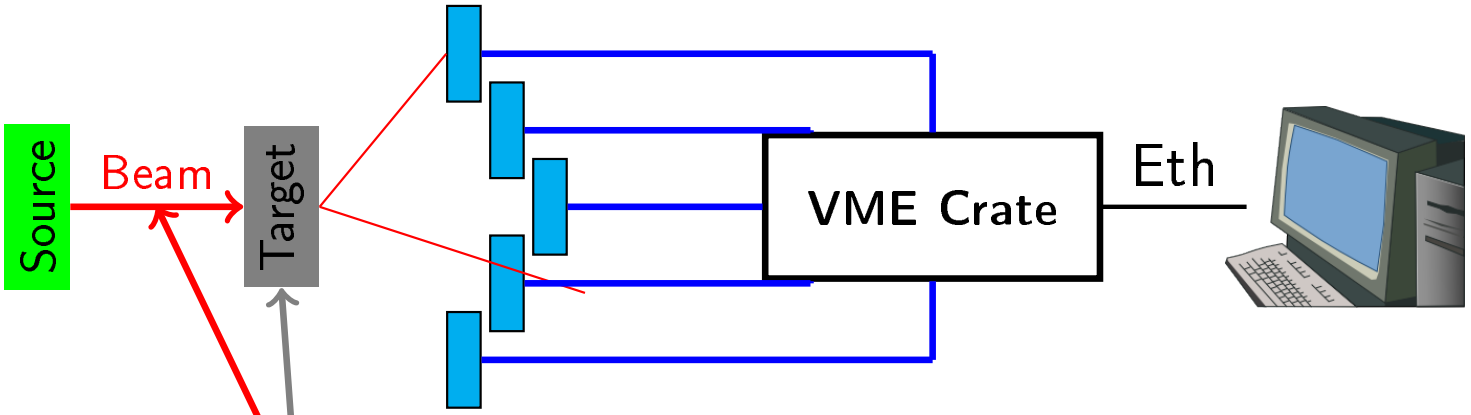
24^{th} November 2011

DAQ basics



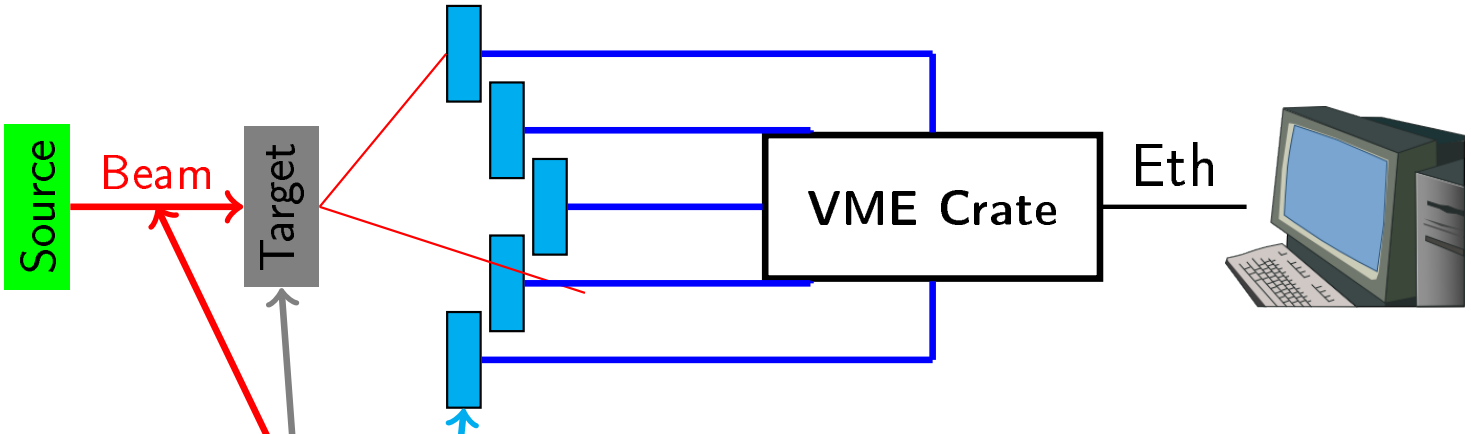
- Beam
  - electrons, protons, ions, molecules (C_{60}, ...)

DAQ basics



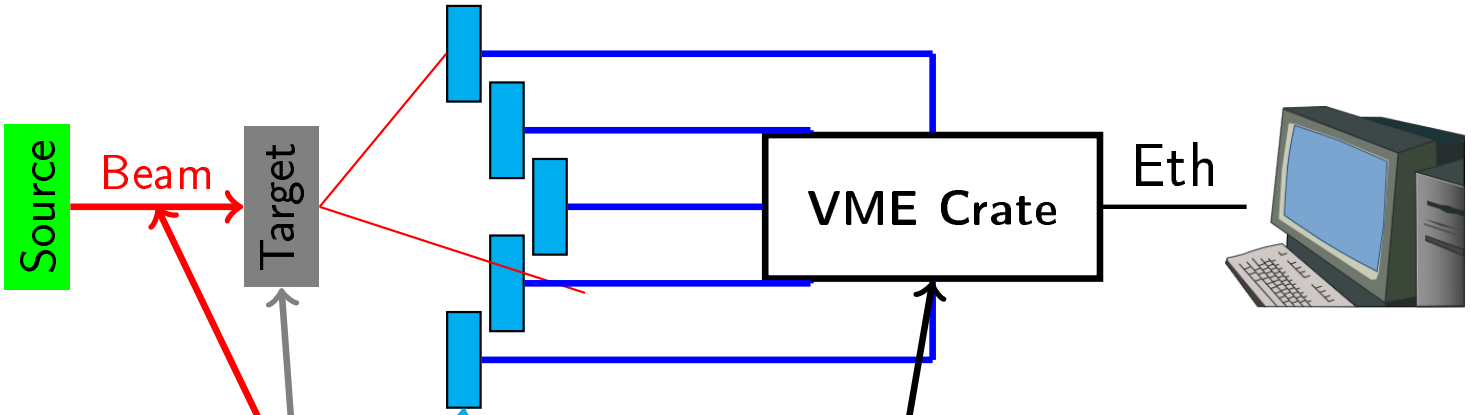
- Beam
  - electrons, protons, ions, molecules (C_{60}, ...)
- Target
  - liquid hydrogen, lead, gold, ...

DAQ basics



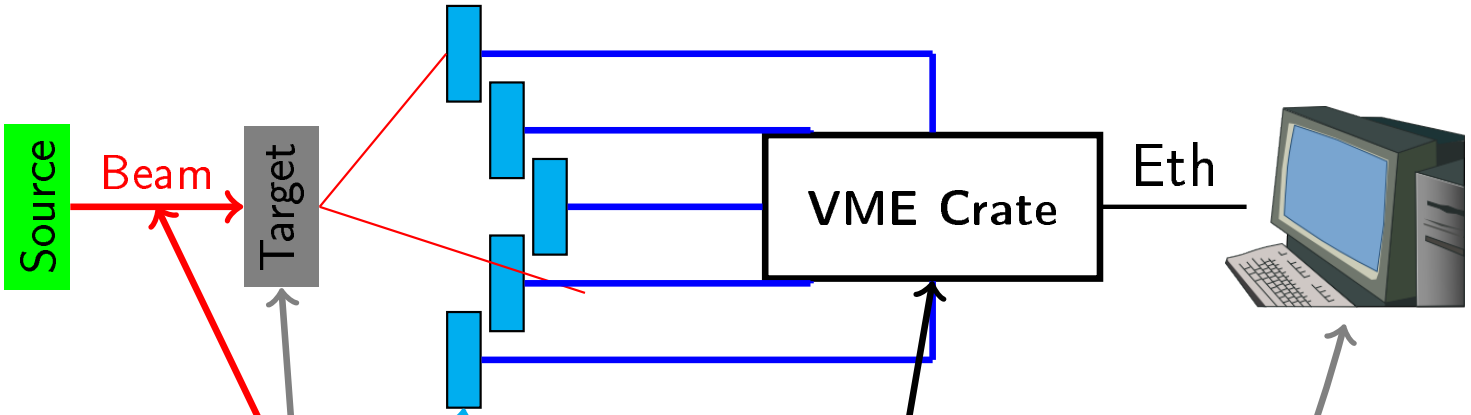
- Beam
  - electrons, protons, ions, molecules (C_{60}, ...)
- Target
  - liquid hydrogen, lead, gold, ...
- Detector(s)
  - silicon based, scintillators, ultra cold germanium mono crystals, ...

DAQ basics



- Beam
  - electrons, protons, ions, molecules (C_{60}, ...)
- Target
  - liquid hydrogen, lead, gold, ...
- Detector(s)
  - silicon based, scintillators, ultra cold germanium mono crystals, ...
- Readout electronics boards
  - VME, PCI, VXI, USB, Eth, ATCA, μTCA ...

DAQ basics

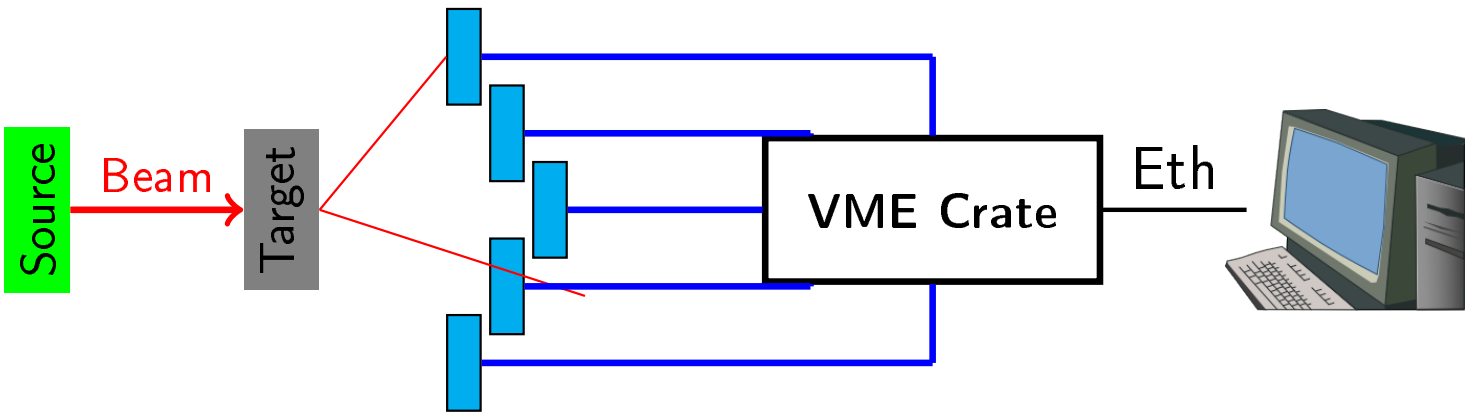


- Beam
  - electrons, protons, ions, molecules (C_{60}, ...)
- Target
  - liquid hydrogen, lead, gold, ...
- Detector(s)
  - silicon based, scintillators, ultra cold germanium mono crystals, ...
- Readout electronics boards
  - VME, PCI, VXI, USB, Eth, ATCA, μTCA ...
- Data acquisition

From OASIS to Narval

OASIS was a monobloc system based on the client/server paradigm :

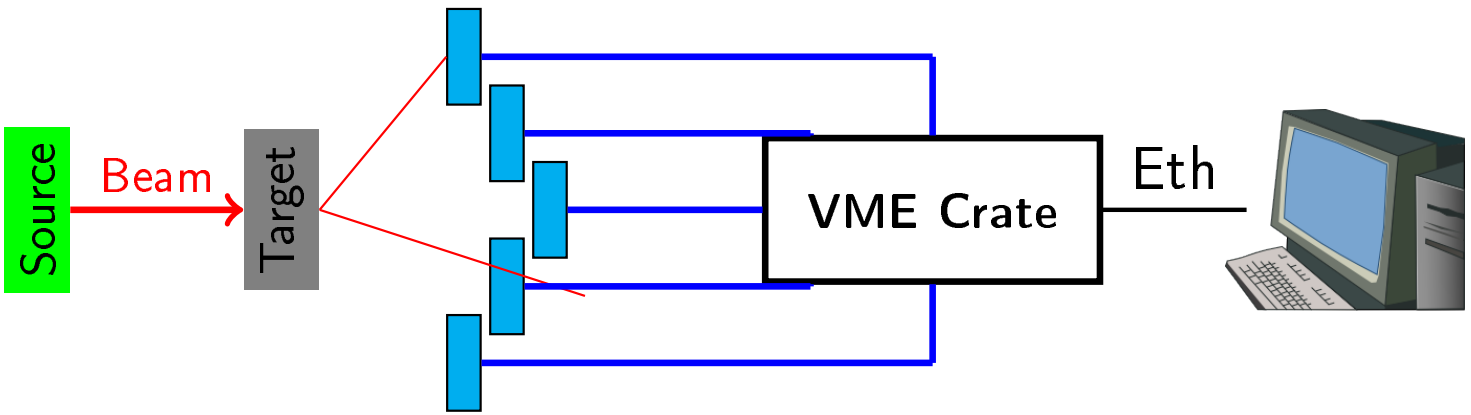
- Embedded client in mostly VME crates under vxWorks



From OASIS to Narval

OASIS was a monobloc system based on the client/server paradigm :

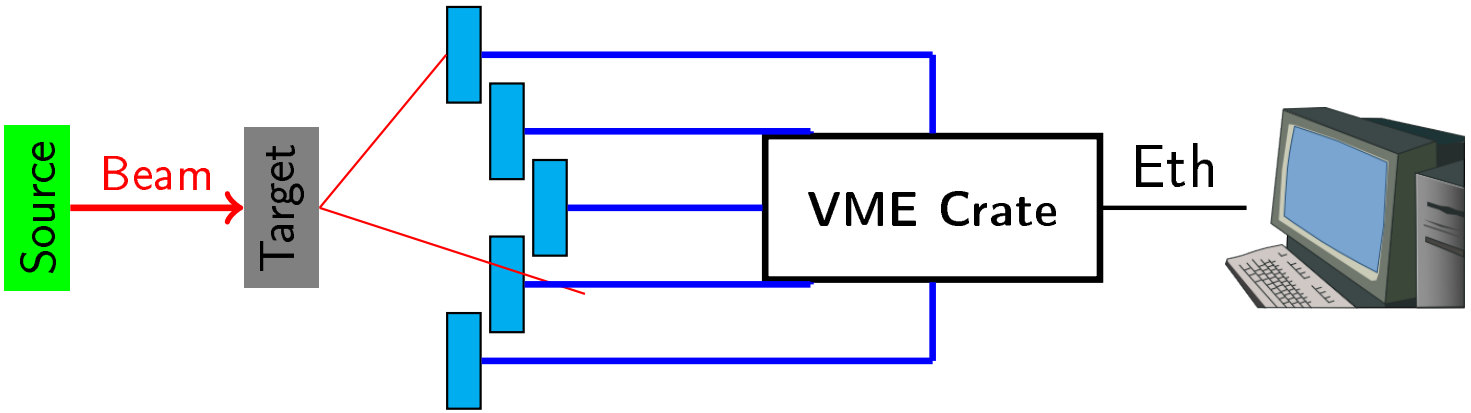
- Embedded client in mostly VME crates under vxWorks
- Server on Solaris Sun hardware



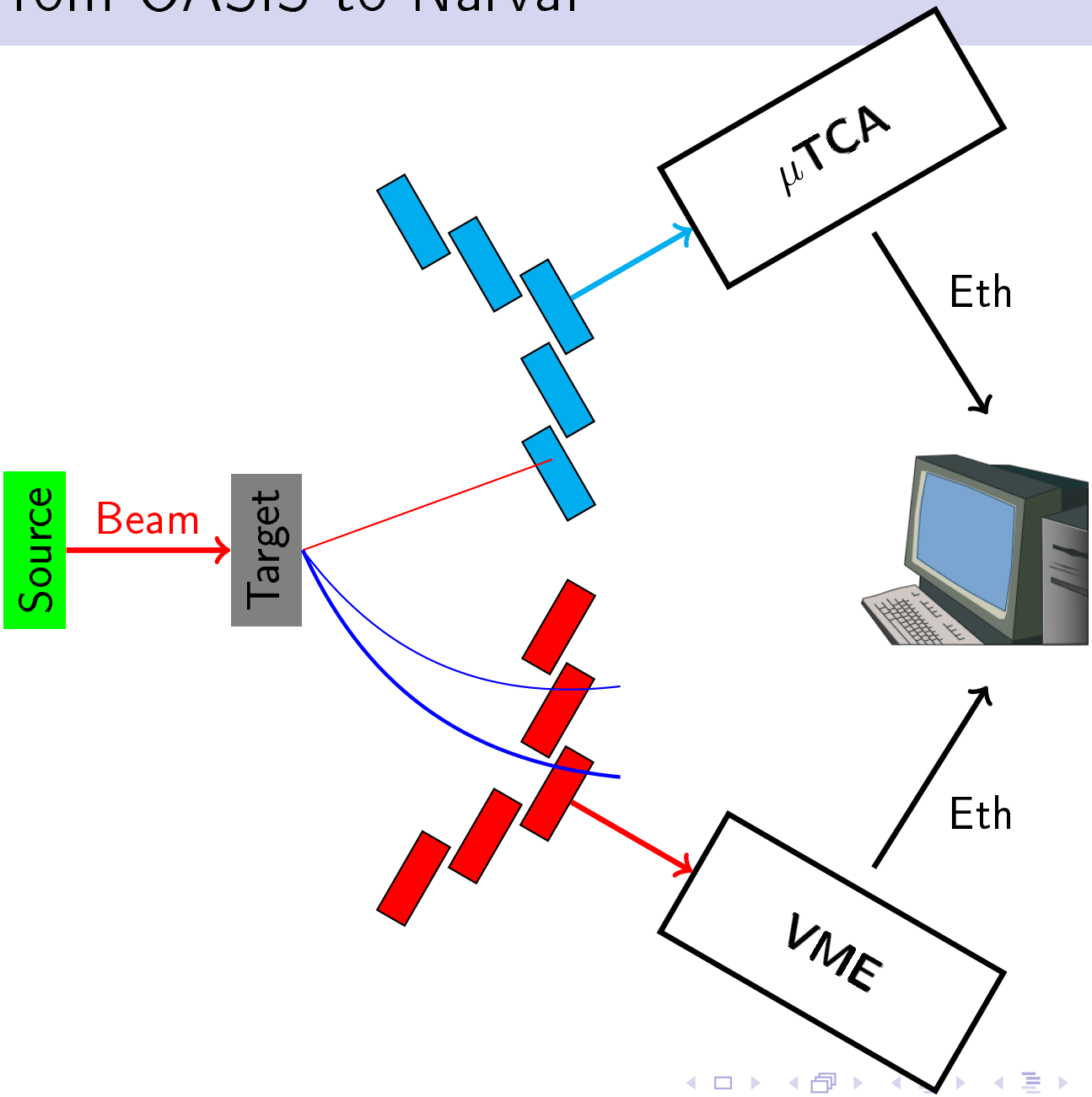
From OASIS to Narval

OASIS was a monobloc system based on the client/server paradigm :

- Embedded client in mostly VME crates under vxWorks
- Server on Solaris Sun hardware
- Communication done using
  - RPC protocol for configuration and walking through the state machine
  - TCP/IP for high speed data transfer



From OASIS to Narval



Narval

Narval stands for :

Nouvelle Acquisition temps Réel Version 1.14.2 Avec Linux

Narval

Narval stands for :

Nouvelle Acquisition temps Réel Version 1.14.2 Avec Linux

It is a distributed system also based on the client/server paradigm :

Narval

Narval stands for :

Nouvelle Acquisition temps Réel Version 1.14.2 Avec Linux

It is a distributed system also based on the client/server paradigm :

- Unique data link replaced by **Data Flow** concept
  - TCP/IP
  - Infiniband
  - Unix Fifo
  - Shared Memory (with the Pool Buffer System)
  - Ada 95 Annex E for system's configuration

Narval

Narval stands for :

Nouvelle Acquisition temps Réel Version 1.14.2 Avec Linux

It is a distributed system also based on the client/server paradigm :

- Unique data link replaced by **Data Flow** concept
  - TCP/IP
  - Infiniband
  - Unix Fifo
  - Shared Memory (with the Pool Buffer System)
  - Ada 95 Annex E for system's configuration
- **Narval is an Open Software** :

Narval

Narval stands for :

Nouvelle Acquisition temps Réel Version 1.14.2 Avec Linux

It is a distributed system also based on the client/server paradigm :

- Unique data link replaced by **Data Flow** concept
  - TCP/IP
  - Infiniband
  - Unix Fifo
  - Shared Memory (with the Pool Buffer System)
  - Ada 95 Annex E for system's configuration
- **Narval is an Open Software** :
  - Operating system : Linux, kFreeBSD

Narval

Narval stands for :

Nouvelle Acquisition temps Réel Version 1.14.2 Avec Linux

It is a distributed system also based on the client/server paradigm :

- Unique data link replaced by **Data Flow** concept
  - TCP/IP
  - Infiniband
  - Unix Fifo
  - Shared Memory (with the Pool Buffer System)
  - Ada 95 Annex E for system's configuration
- **Narval is an Open Software** :
  - Operating system : Linux, kFreeBSD
  - **Ada**, C, C++ Compiler : GNAT, gcc, g++

Narval

Narval stands for :

Nouvelle Acquisition temps Réel Version 1.14.2 Avec Linux

It is a distributed system also based on the client/server paradigm :

- Unique data link replaced by **Data Flow** concept
  - TCP/IP
  - Infiniband
  - Unix Fifo
  - Shared Memory (with the Pool Buffer System)
  - Ada 95 Annex E for system's configuration
- **Narval is an Open Software** :
  - Operating system : Linux, kFreeBSD
  - **Ada**, C, C++ Compiler : GNAT, gcc, g++
  - Narval sources released under **GPL**

Narval

Narval stands for :

Nouvelle Acquisition temps Réel Version 1.14.2 Avec Linux

It is a distributed system also based on the client/server paradigm :

- Unique data link replaced by **Data Flow** concept
  - TCP/IP
  - Infiniband
  - Unix Fifo
  - Shared Memory (with the Pool Buffer System)
  - Ada 95 Annex E for system's configuration
- **Narval is an Open Software** :
  - Operating system : Linux, kFreeBSD
  - **Ada**, C, C++ Compiler : GNAT, gcc, g++
  - Narval sources released under **GPL**
  - **Debian** packages available since Squeeze release

Narval localisations

Since 2001 Narval is used at (not an exhaustive list) :

- ALTO / IPN Orsay
- Scanning table / CSNSM
- Data Flow manager for some GANIL experiments (soon for SPIRAL2 experiments)
- AGATA Detector / Legnaro (soon GSI)
- Chaco VME transportable system / CEA B3
- LAG64 (VME TDC) in a few labs (Germany, USA)
- LPNHE for some electronic test bench (ILC)
- ...

Ada language

Narval is developed in Ada 2005 :

- Strong typing (enhance code stability)

Ada language

Narval is developed in Ada 2005 :

- Strong typing (enhance code stability)
- Native tasking (no need for multi threading libraries)
  - Rendez Vous mechanisms
  - Protected Objects

Ada language

Narval is developed in Ada 2005 :

- Strong typing (enhance code stability)
- Native tasking (no need for multi threading libraries)
  - Rendez Vous mechanisms
  - Protected Objects
  - $\Rightarrow$ no need for mutex or semaphore, code is "naturally" reentrant

Ada language

Narval is developed in Ada 2005 :

- Strong typing (enhance code stability)
- Native tasking (no need for multi threading libraries)
  - Rendez Vous mechanisms
  - Protected Objects
  - $\Rightarrow$ no need for mutex or semaphore, code is "naturally" reentrant
- Distributed Application System (Annex E)
  - provides one distributed application across network and architectures

Ada language

Narval is developed in Ada 2005 :

- Strong typing (enhance code stability)
- Native tasking (no need for multi threading libraries)
  - Rendez Vous mechanisms
  - Protected Objects
  - $\Rightarrow$ no need for mutex or semaphore, code is "naturally" reentrant
- Distributed Application System (Annex E)
  - provides one distributed application across network and architectures
- OO Programming
  - including interfaces (same paradigm as in Java)

Ada language

Narval is developed in Ada 2005 :

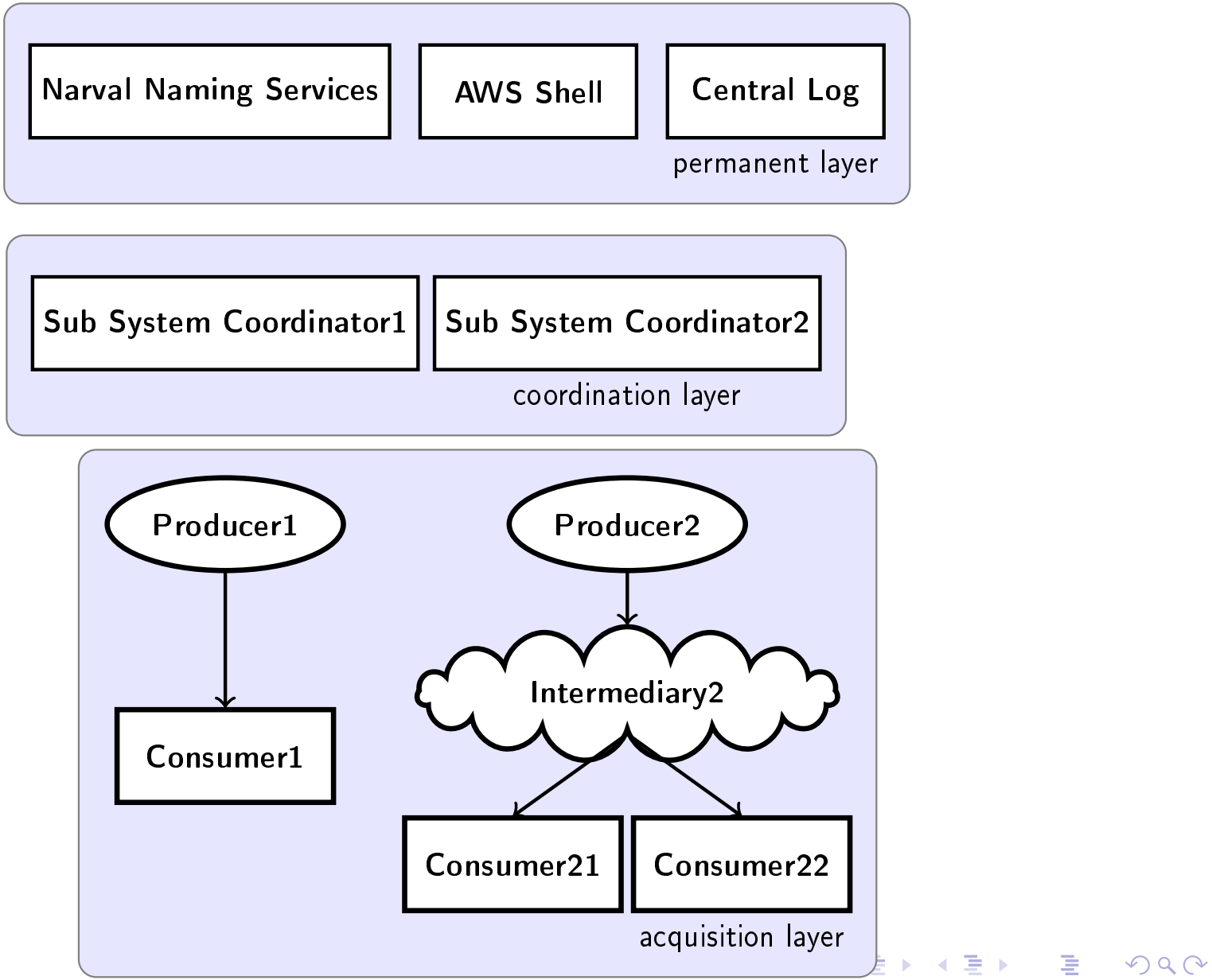
- Strong typing (enhance code stability)
- Native tasking (no need for multi threading libraries)
  - Rendez Vous mechanisms
  - Protected Objects
  - $\Rightarrow$ no need for mutex or semaphore, code is "naturally" reentrant
- Distributed Application System (Annex E)
  - provides one distributed application across network and architectures
- OO Programming
  - including interfaces (same paradigm as in Java)
- ISO Standard since 1983

Ada language

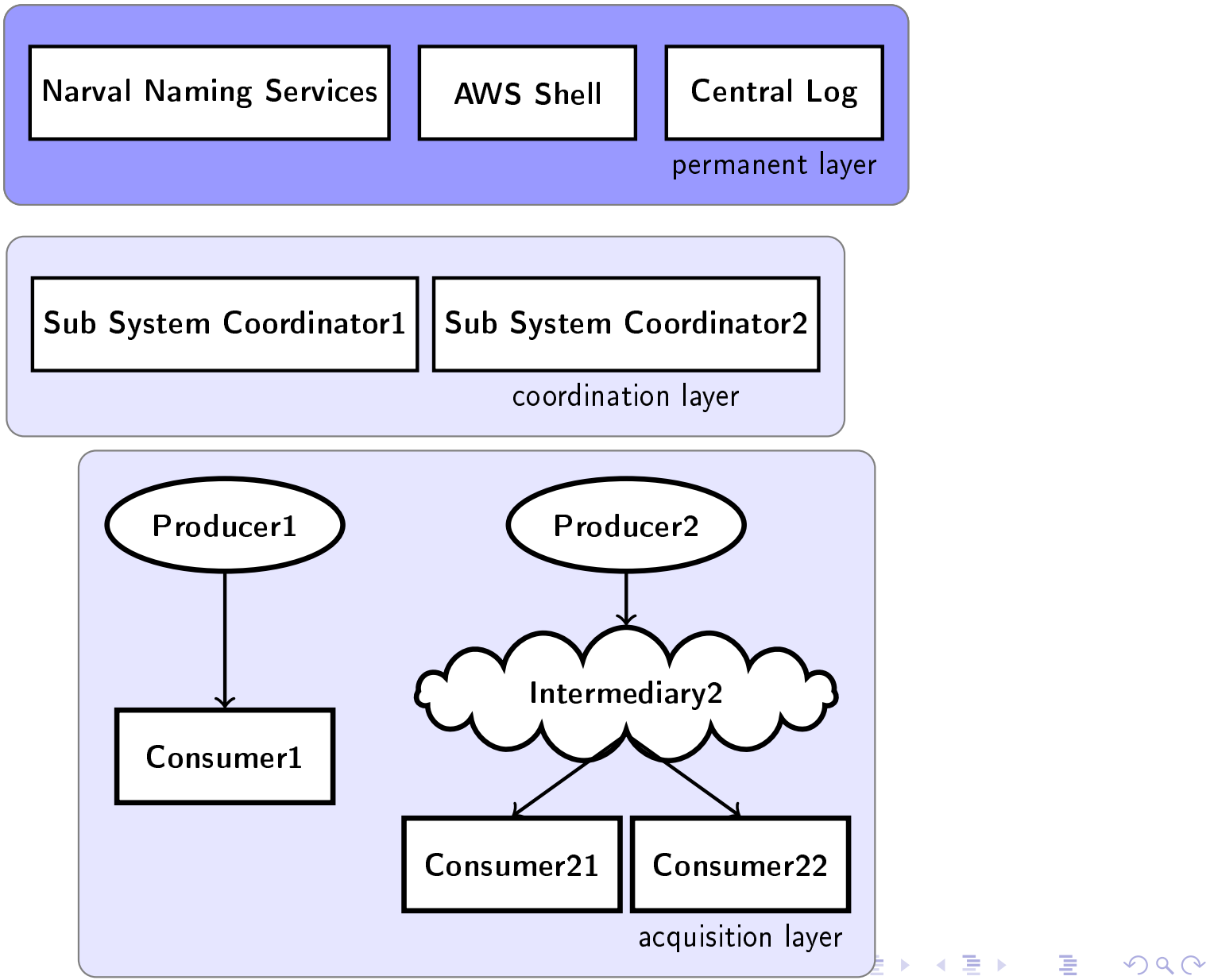
Narval is developed in Ada 2005 :

- Strong typing (enhance code stability)
- Native tasking (no need for multi threading libraries)
  - Rendez Vous mechanisms
  - Protected Objects
  - $\Rightarrow$ no need for mutex or semaphore, code is "naturally" reentrant
- Distributed Application System (Annex E)
  - provides one distributed application across network and architectures
- OO Programming
  - including interfaces (same paradigm as in Java)
- ISO Standard since 1983
  - new standard version in 1995
  - addendum in 2005

A three layers framework



A three layers framework



Coordination processes 1/2

- `narval_naming_services`
  - register and list all sub systems (Name ⇔ fat pointer)
  - "Annex E" Remote Call Interface (a library promoted on the network)

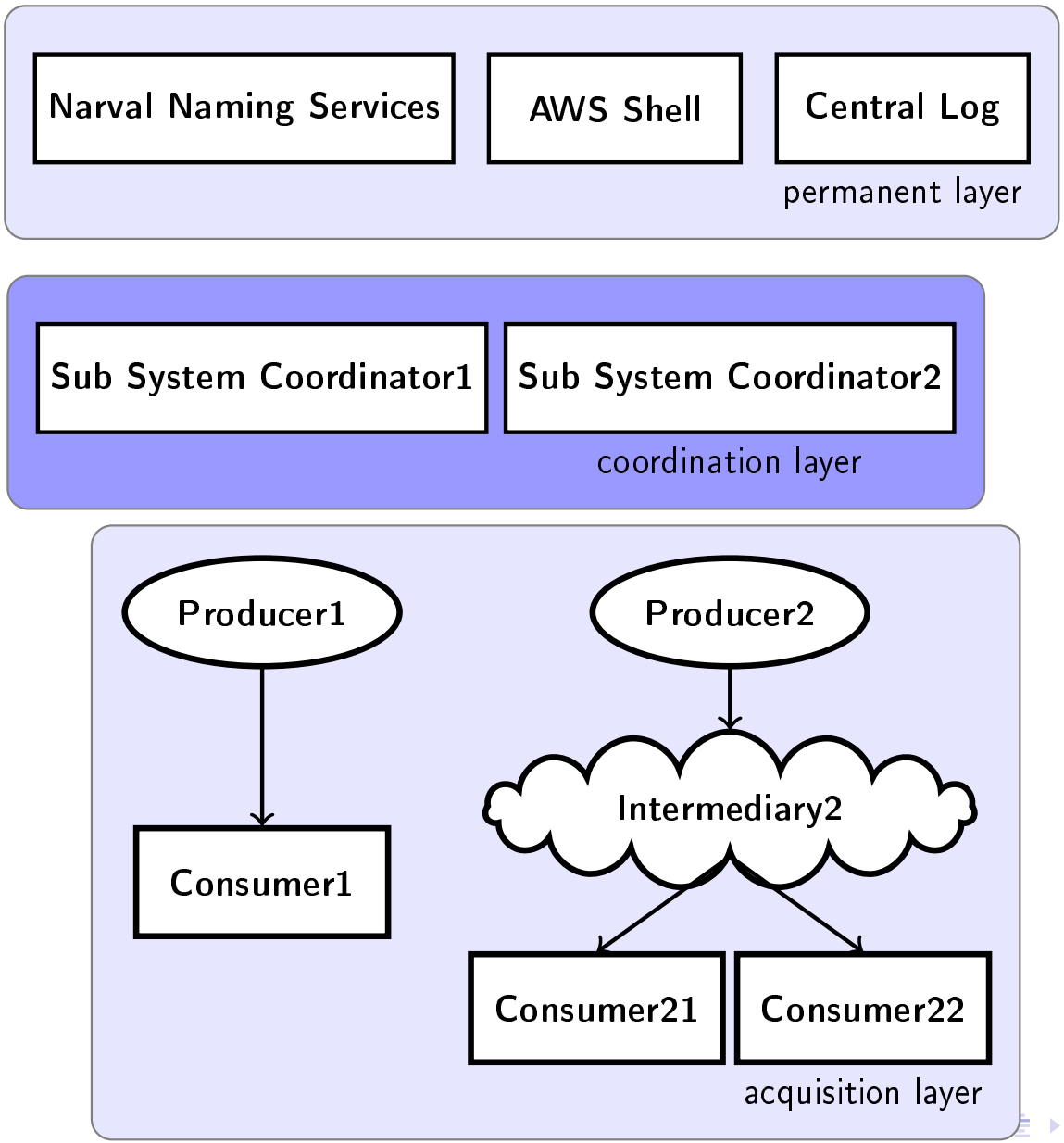
Coordination processes 1/2

- `narval_naming_services`
  - register and list all sub systems (Name ⇔ fat pointer)
  - "Annex E" Remote Call Interface (a library promoted on the network)
- `aws_shell`
  - bridge between Narval Framework and the world as a Web Services interface (using SOAP protocol)

Coordination processes 1/2

- `narval_naming_services`
  - register and list all sub systems (Name ⇔ fat pointer)
  - "Annex E" Remote Call Interface (a library promoted on the network)
- `aws_shell`
  - bridge between Narval Framework and the world as a Web Services interface (using SOAP protocol)
- `central_log`
  - centralise all logs of the Narval Framework
  - Can redirect them to files, to chainsaw or any log4j compatible system
  - "Annex E" Remote type (an object promoted on the network)

Coordination Layer



Coordination processes 2/2

- `sub_system_coordinator`
  - List all actors of a sub system (Name ⇔ Fat pointer)

Coordination processes 2/2

- `sub_system_coordinator`
  - List all actors of a sub system (Name ⇔ Fat pointer)
  - Launch all the actors

Coordination processes 2/2

- `sub_system_coordinator`
  - List all actors of a sub system (Name $\Leftrightarrow$ Fat pointer)
  - Launch all the actors
  - Take care of the State Machine of a given sub system

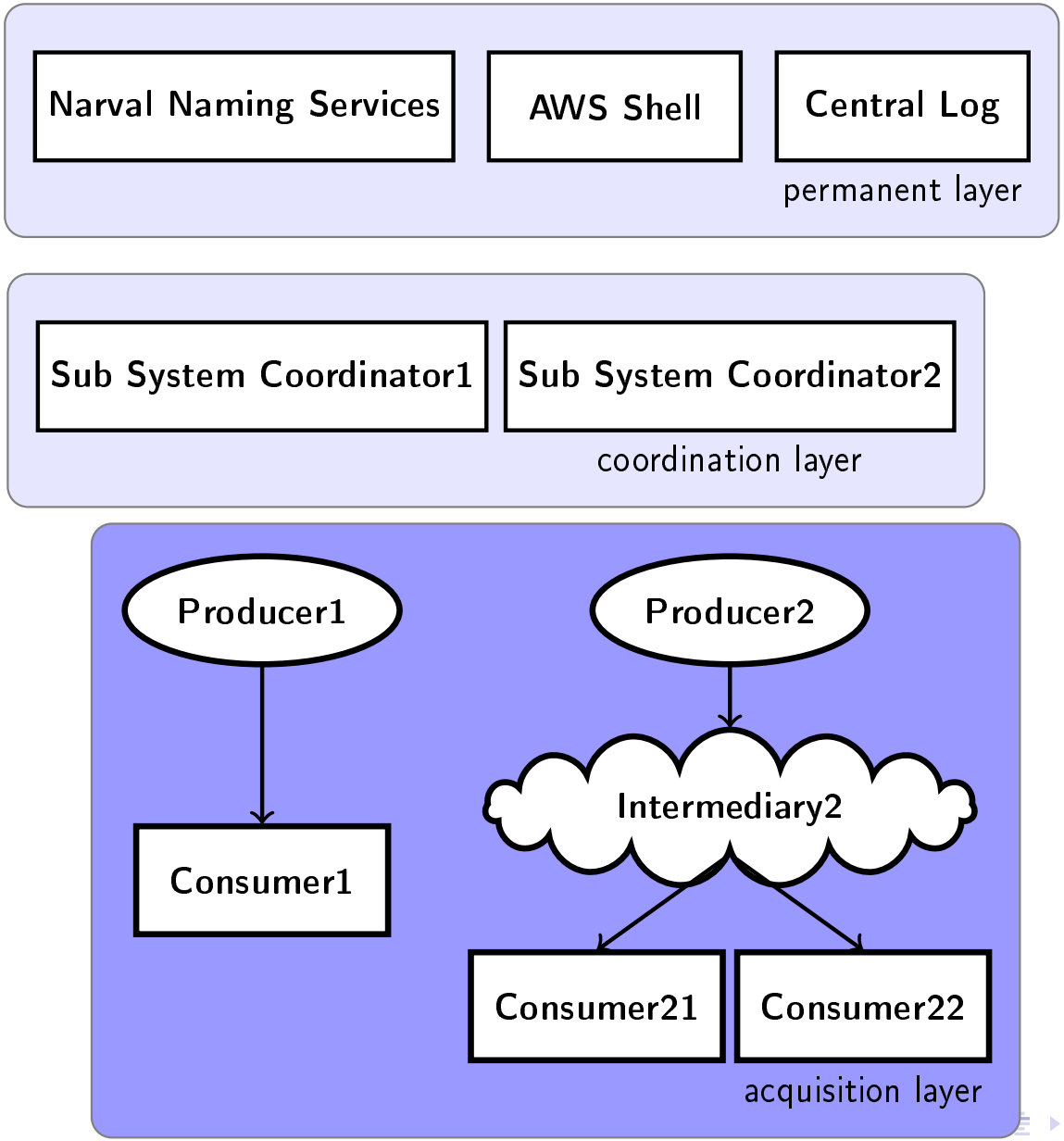
Coordination processes 2/2

- `sub_system_coordinator`
  - List all actors of a sub system (Name $\Leftrightarrow$ Fat pointer)
  - Launch all the actors
  - Take care of the State Machine of a given sub system
  - Dispatch orders to the actors when there is a change in the state machine

Coordination processes 2/2

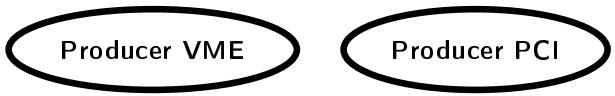
- `sub_system_coordinator`
  - List all actors of a sub system (Name ⇔ Fat pointer)
  - Launch all the actors
  - Take care of the State Machine of a given sub system
  - Dispatch orders to the actors when there is a change in the state machine
  - "Annex E" Remote Type

Acquisition Layer



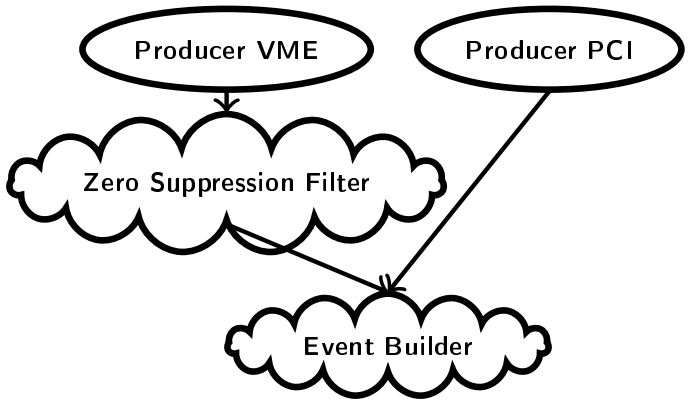
Actors

- Producer
  - `generic_producer`
  - `data_rate_producer`
  - `generic_ada_actor`



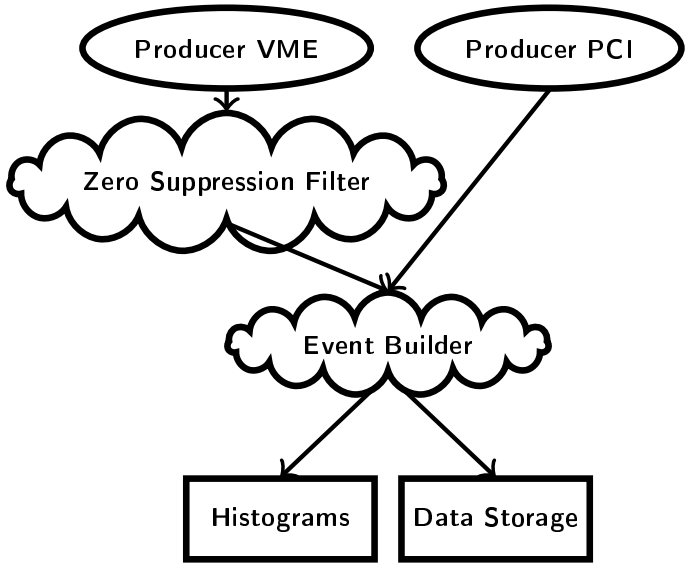
Actors

- Producer
  - `generic_producer`
  - `data_rate_producer`
  - `generic_ada_actor`
- Intermediary
  - `generic_filter`
  - `generic_ada_actor`

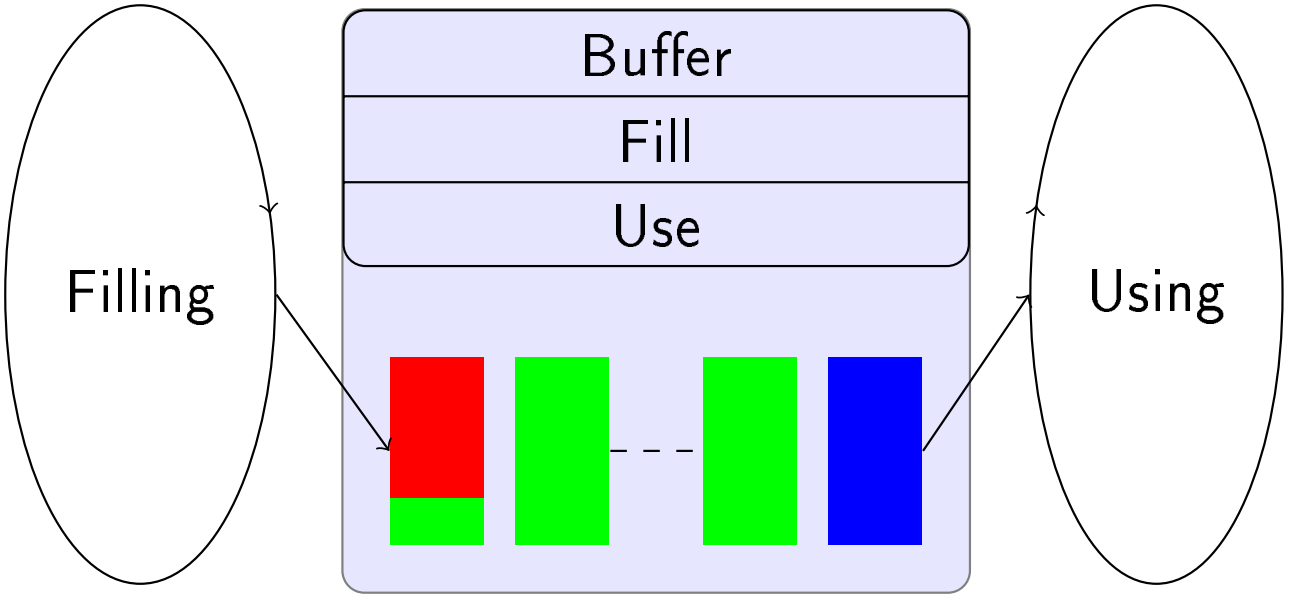


Actors

- Producer
  - `generic_producer`
  - `data_rate_producer`
  - `generic_ada_actor`
- Intermediary
  - `generic_filter`
  - `generic_ada_actor`
- Consumer
  - `generic_consumer`
  - `data_rate_consumer`
  - `generic_ada_actor`



Buffers in actors



Where Filling is :

- Network (consumer)
- Buffer Handling (producer)

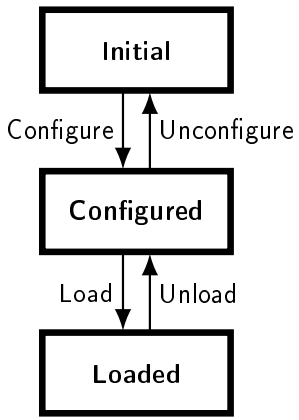
Where Using is :

- Network (producer)
- Buffer Handling (consumer)

The State Machine

Actors available actions :

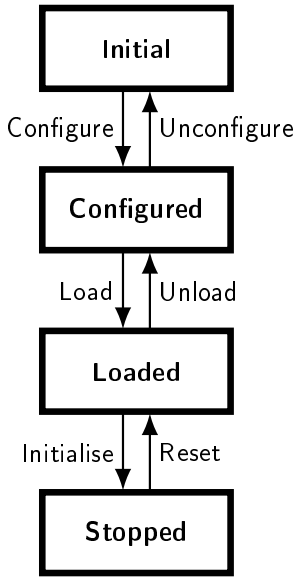
- Initialise when the actor is loaded



The State Machine

Actors available actions :

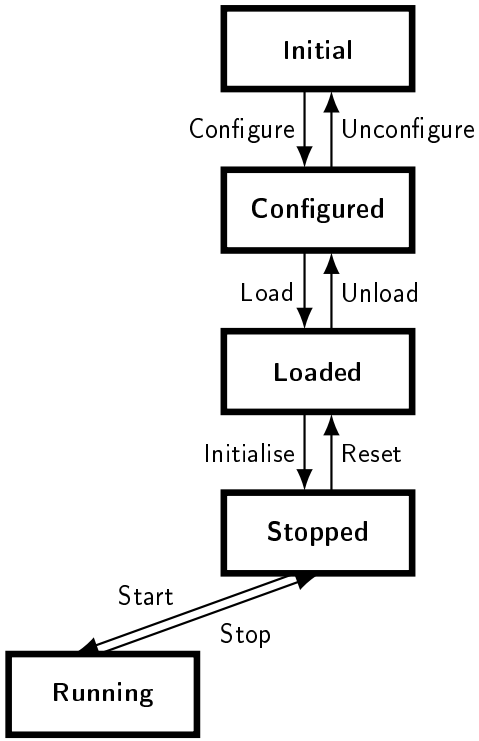
- Initialise when the actor is loaded
- On_Initialise
- On_Reset_Com



The State Machine

Actors available actions :

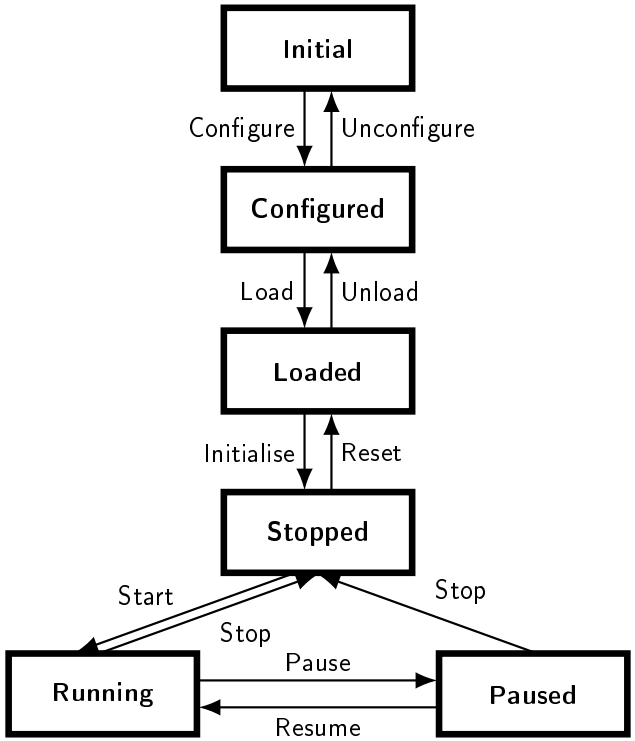
- Initialise when the actor is loaded
- On_Initialise
- On_Reset_Com
- On_Start
- On_Stop



The State Machine

Actors available actions :

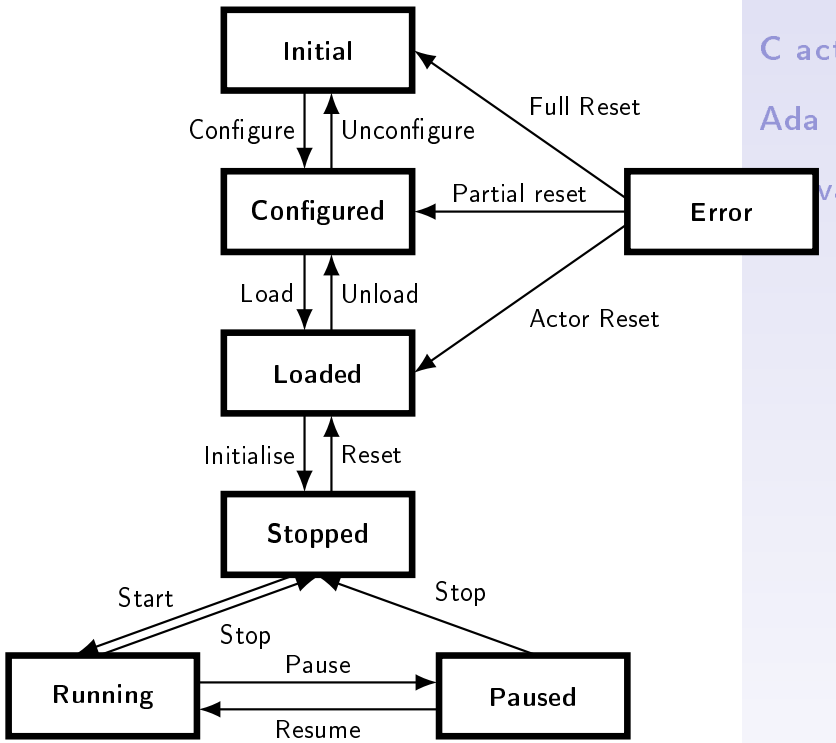
- Initialise when the actor is loaded
- On_Initialise
- On_Reset_Com
- On_Start
- On_Stop
- On_Pause
- On_Resume



The State Machine

Actors available actions :

- Initialise when the actor is loaded
- On_Initialise
- On_Reset_Com
- On_Start
- On_Stop
- On_Pause
- On_Resume
- On_Reset for actor reset

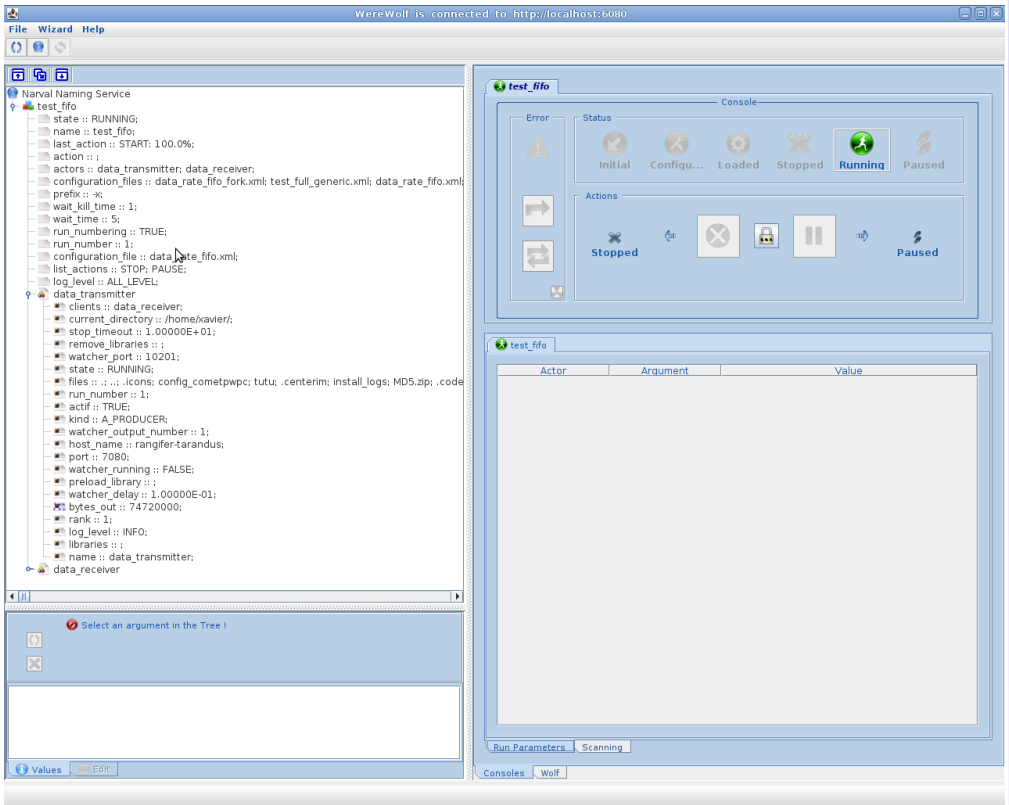


Control processes

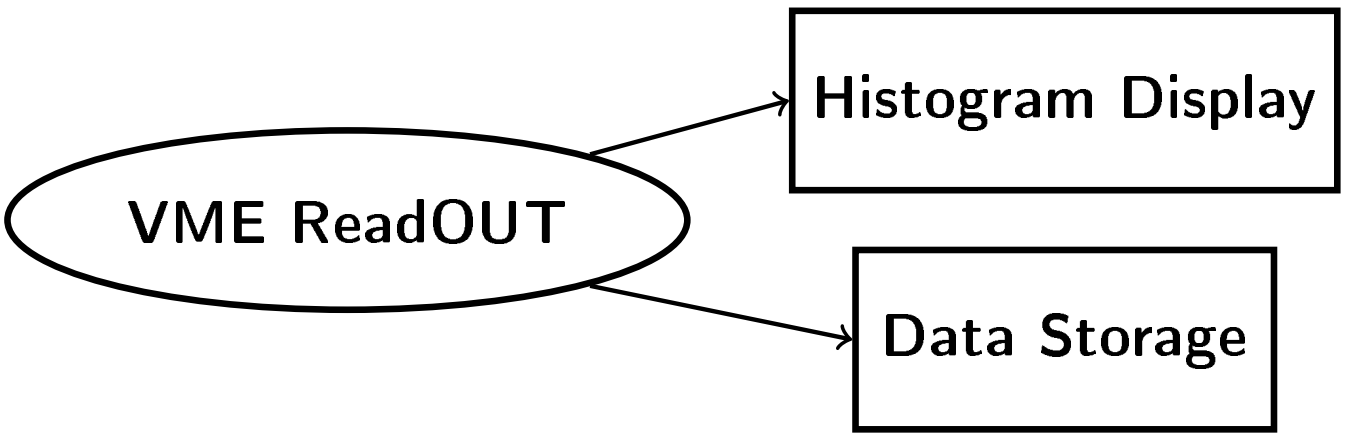
- `narval_shell`
  - command line tool to control narval sub systems

Control processes

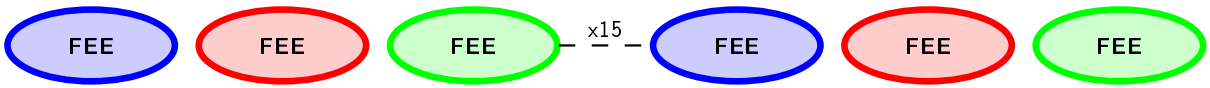
- `narval_shell`
  - command line tool to control narval sub systems
- werewolf
  - graphical tool to control narval sub systems
  - command line available in expert mode



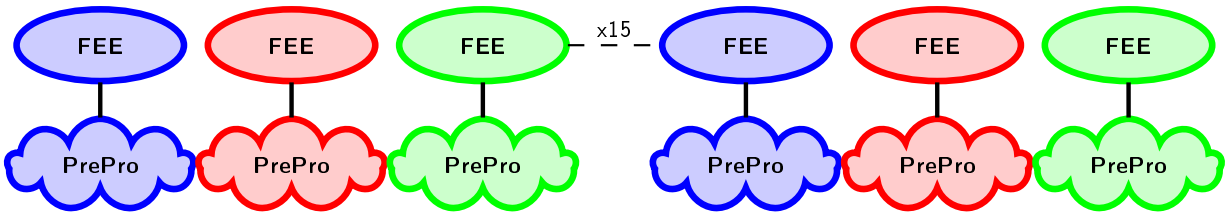
Topologies examples 1/2 : test bench



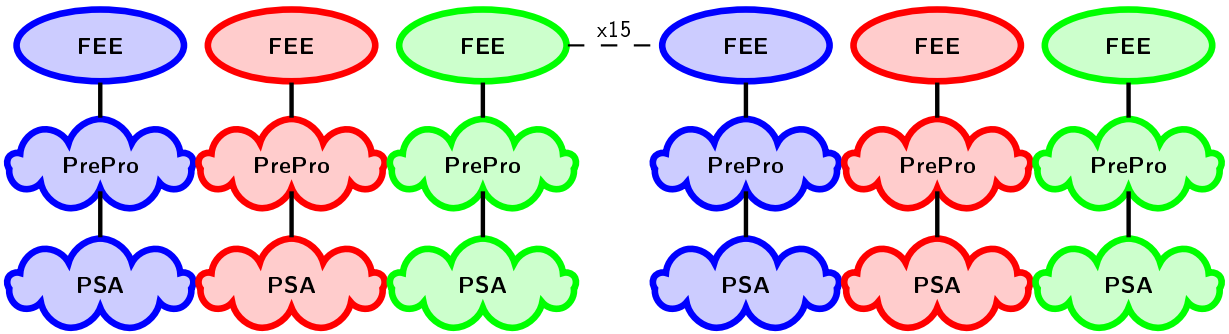
Topologies examples 2/2 : AGATA



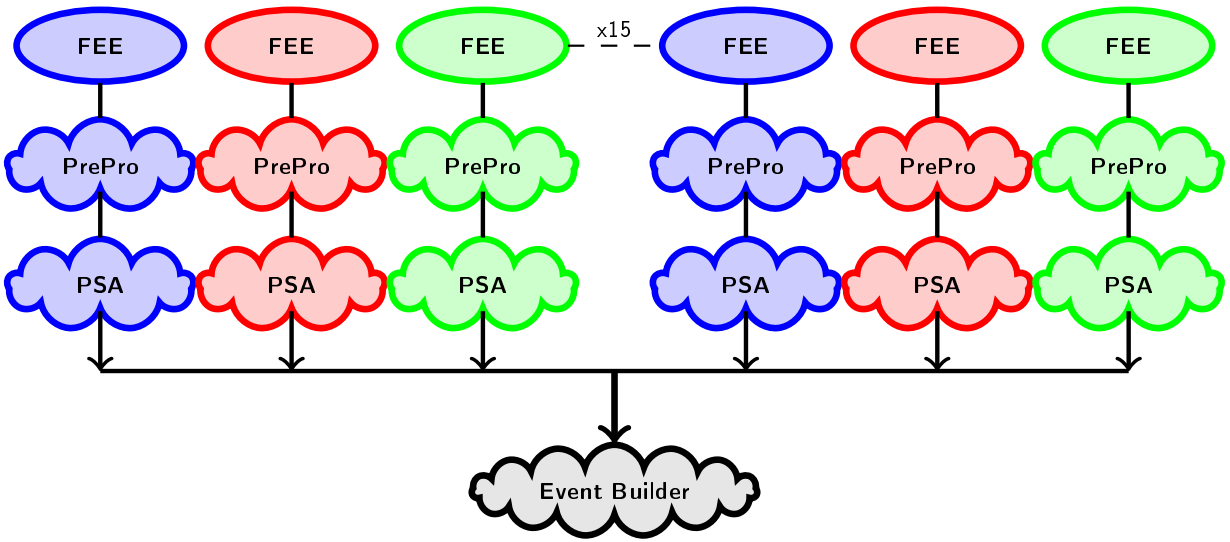
Topologies examples 2/2 : AGATA



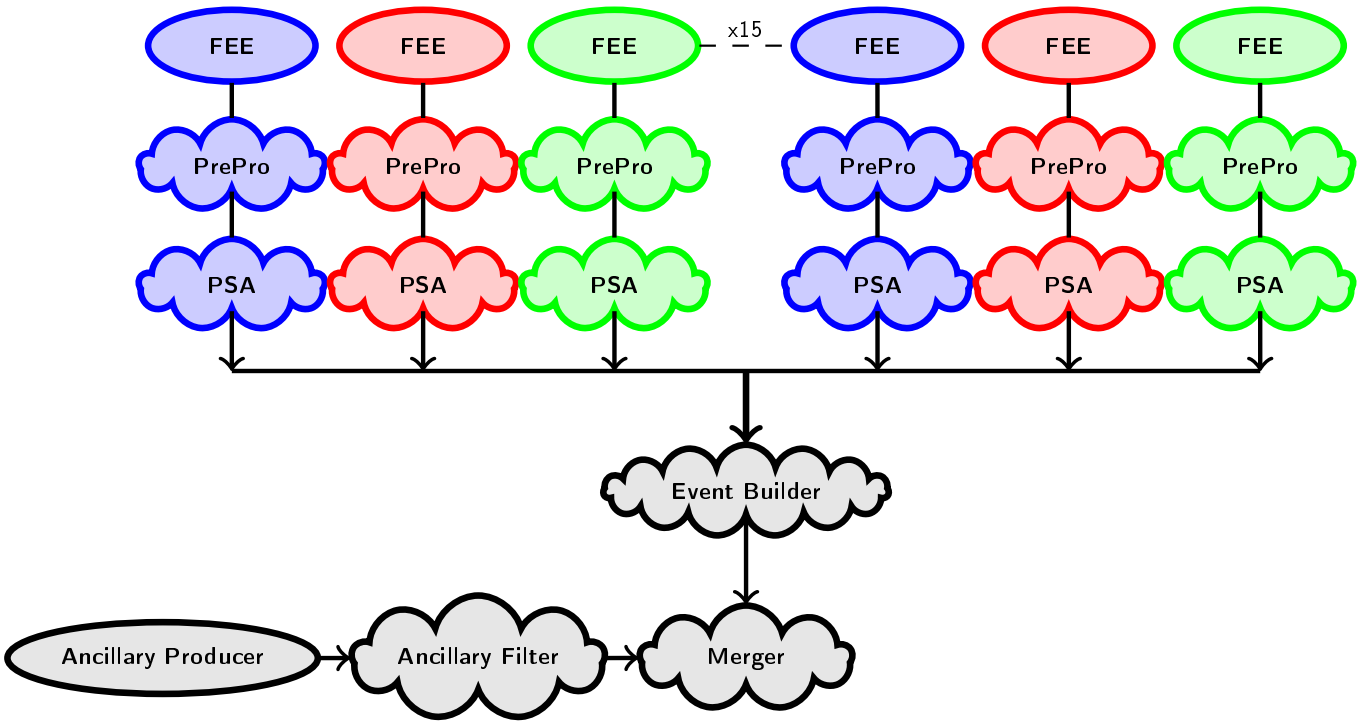
Topologies examples 2/2 : AGATA



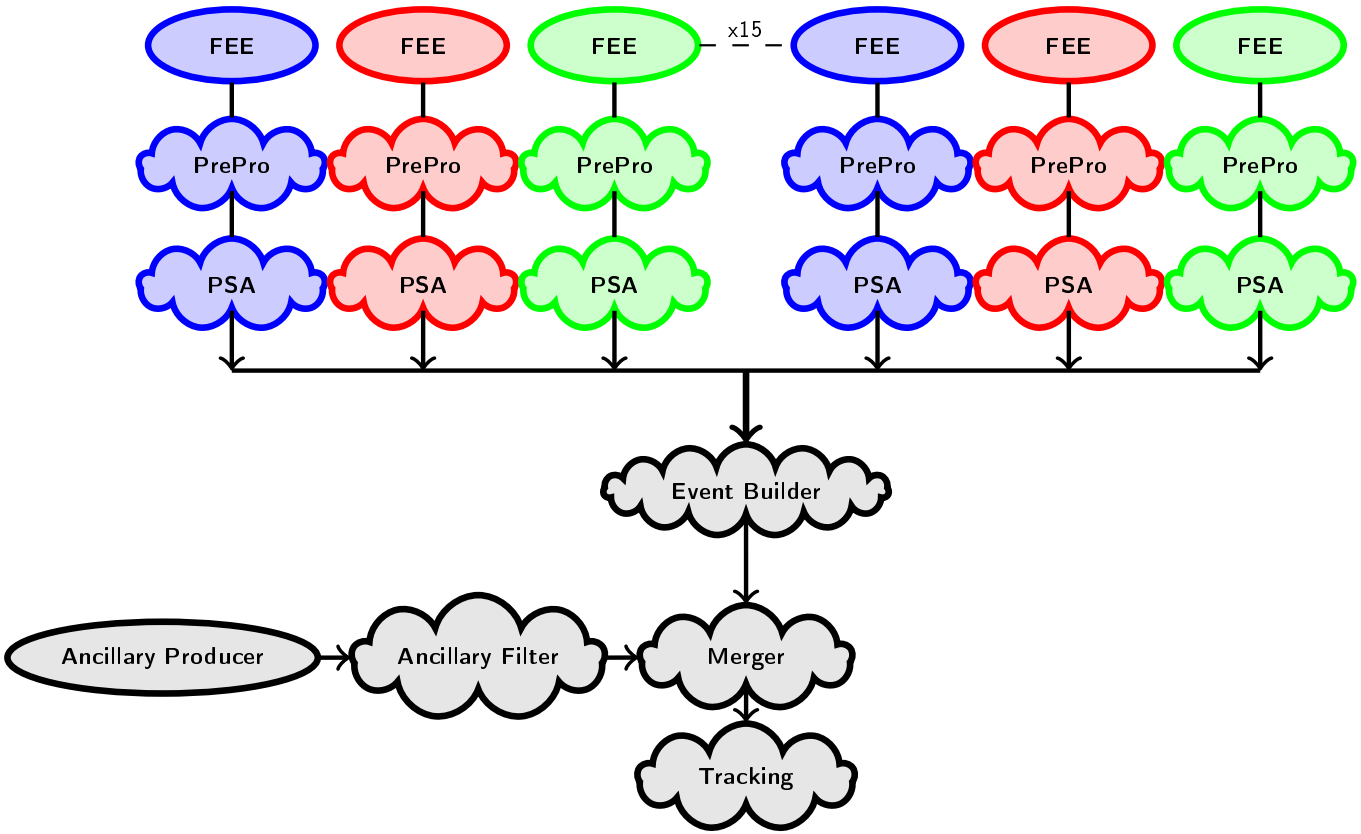
Topologies examples 2/2 : AGATA



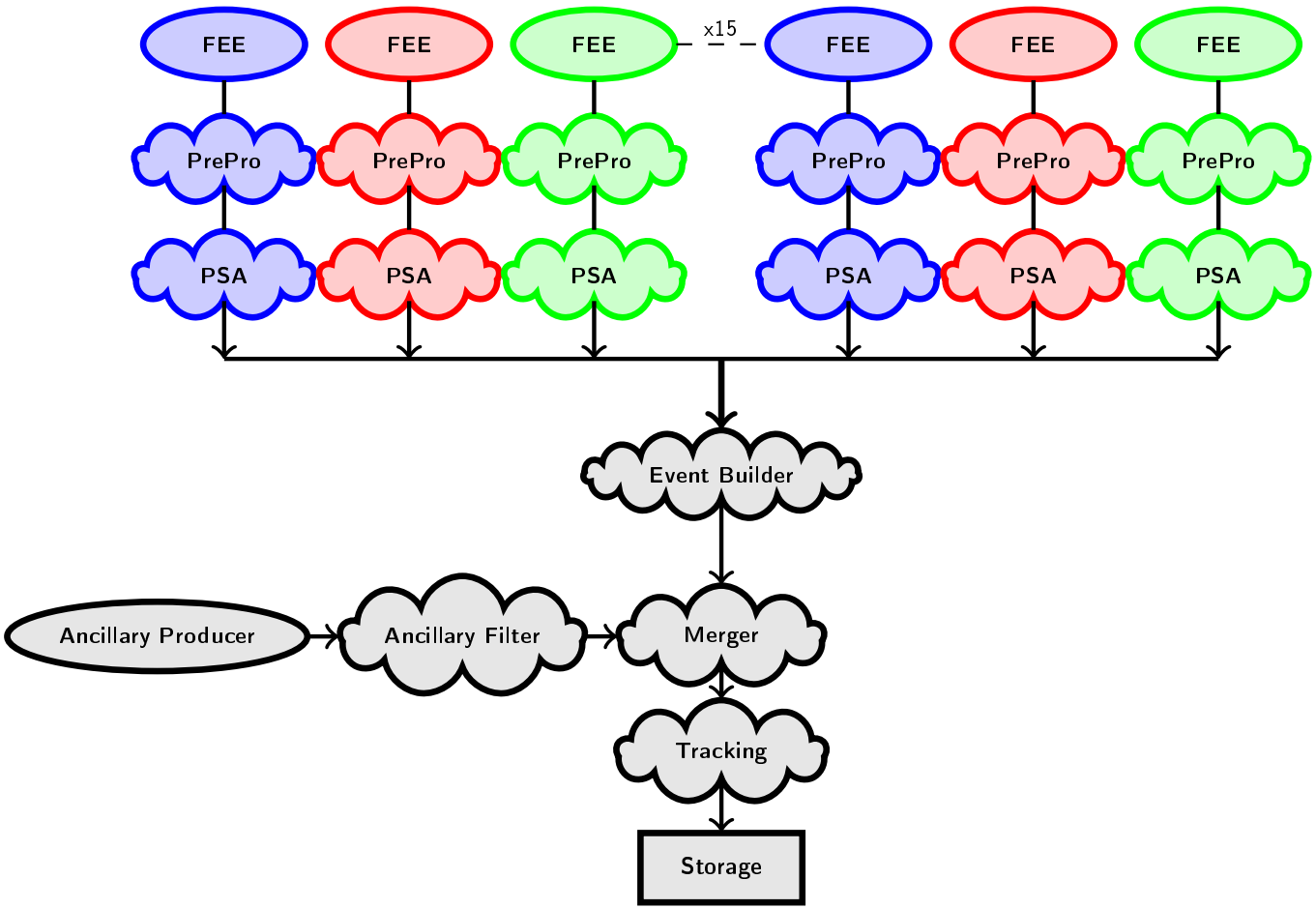
Topologies examples 2/2 : AGATA



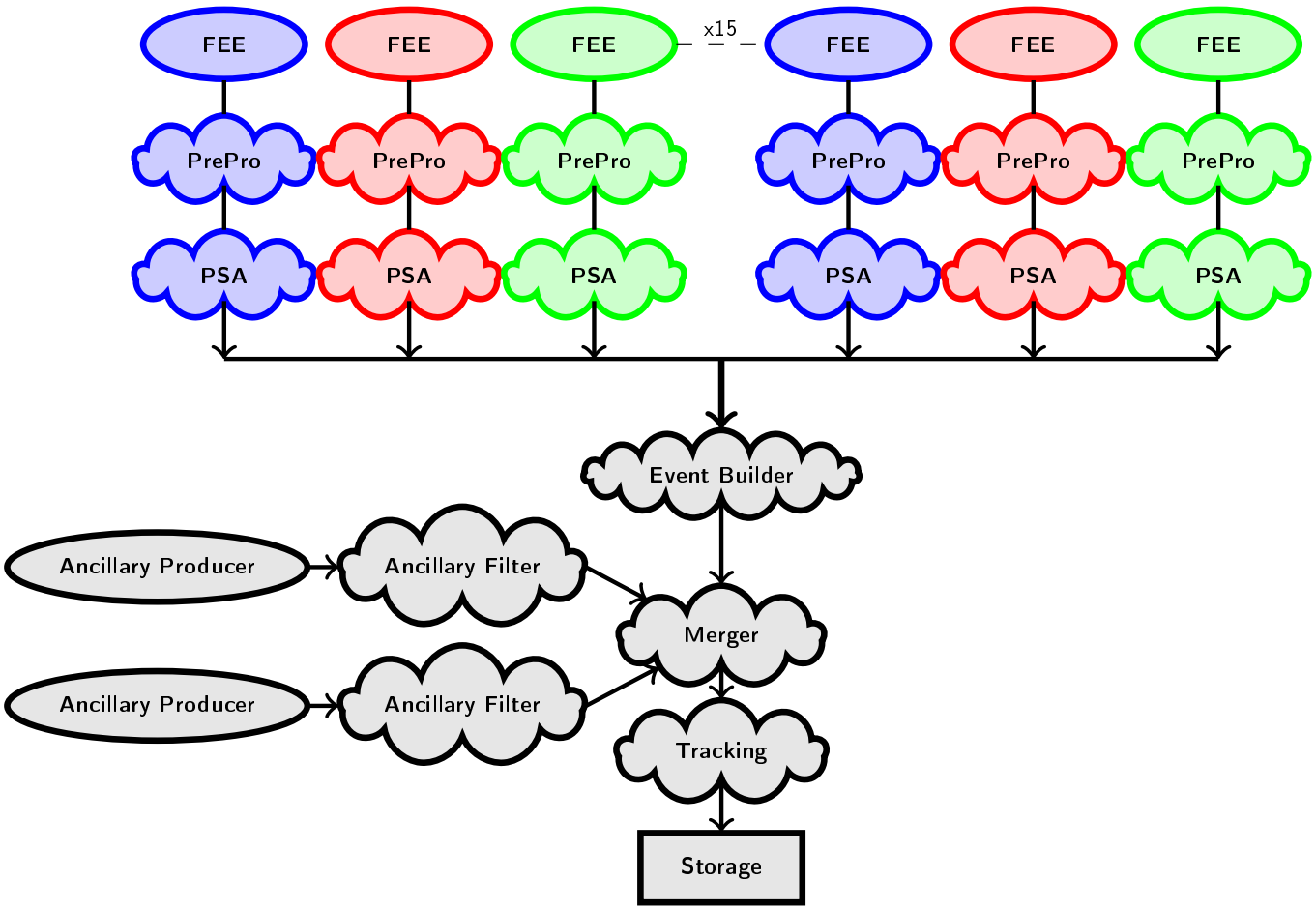
Topologies examples 2/2 : AGATA



Topologies examples 2/2 : AGATA



Topologies examples 2/2 : AGATA



Actor as a shared library

- C or C++ shared library

Actor as a shared library

- C or C++ shared library
  - producer : one output

Actor as a shared library

- C or C++ shared library
  - producer : one output
  - filter : one input to process and inject the result in one output

Actor as a shared library

- C or C++ shared library
  - producer : one output
  - filter : one input to process and inject the result in one output
  - consumer : one input to process

Actor as a shared library

- C or C++ shared library
  - producer : one output
  - filter : one input to process and inject the result in one output
  - consumer : one input to process
- loaded by the Narval shell command :
  - `set library libMy_C_Code.so subsys actor`

Actor as a shared library

- C or C++ shared library
  - producer : one output
  - filter : one input to process and inject the result in one output
  - consumer : one input to process
- loaded by the Narval shell command :
  - `set library libMy_C_Code.so subsys actor`
- one can preload others shared libraries
  - ROOT libs, GRU (Ganil Root Utilities), etc...
  - `set preload_library libGRU.so subsys actor`

Actor as a shared library

- C or C++ shared library
  - producer : one output
  - filter : one input to process and inject the result in one output
  - consumer : one input to process
- loaded by the Narval shell command :
  - `set library libMy_C_Code.so subsys actor`
- one can preload others shared libraries
  - ROOT libs, GRU (Ganil Root Utilities), etc...
  - `set preload_library libGRU.so subsys actor`
- No event builder possibilities

Ada Actor as a process

- Principal interest : access to full narval API
  - Buffers

Ada Actor as a process

- Principal interest : access to full narval API
  - Buffers
  - Tasking

Ada Actor as a process

- Principal interest : access to full narval API
  - Buffers
  - Tasking
  - Network

Ada Actor as a process

- Principal interest : access to full narval API
  - Buffers
  - Tasking
  - Network
  - Parameters

Ada Actor as a process

- Principal interest : access to full narval API
  - Buffers
  - Tasking
  - Network
  - Parameters
  - Log4Ada

Ada Actor as a process

- Principal interest : access to full narval API
  - Buffers
  - Tasking
  - Network
  - Parameters
  - Log4Ada
- Inherit from `Active_Actor_Type` tagged record ($\simeq$ C++ class)
- Must override `Buffer_handling`

Ada Actor as a process

- Principal interest : access to full narval API
  - Buffers
  - Tasking
  - Network
  - Parameters
  - Log4Ada
- Inherit from `Active_Actor_Type` tagged record ($\simeq$ C++ class)
- Must override `Buffer_handling`
- Should override (priority order) :
  - `On_Start`, `On_Stop`
  - `On_Initialise`, `On_Reset_Com`
  - `On_Pause`, `On_Resume`
  - `On_Reset`

Ada Actor as a shared library

- Principal interest : easy to compile (doesn't need full narval development environment)

Ada Actor as a shared library

- Principal interest : easy to compile (doesn't need full narval development environment)
- Partial access to narval API
  - Buffers
  - Parameters
  - Log4Ada

Ada Actor as a shared library

- Principal interest : easy to compile (doesn't need full narval development environment)
- Partial access to narval API
  - Buffers
  - Parameters
  - Log4Ada
- Inherit from `Actor_Interface` and must override/provide all the Interface API

Ada Actor as a shared library

- Principal interest : easy to compile (doesn't need full narval development environment)
- Partial access to narval API
  - Buffers
  - Parameters
  - Log4Ada
- Inherit from `Actor_Interface` and must override/provide all the Interface API
- State Machine
  - `Buffer_handling`, `On_Start`, `On_Stop`, `On_Initialise`, `On_Reset_Com`, `On_Pause`, `On_Resume`, `On_Reset`, `On_Unload`
- IOs
  - `Add_Input`, `Add_Output`, `Clear_IO`, `Append_Parameters`, `Set_Logger`

Narval Support

- Narval School
  - first one $17^{th} - 21^{th}$ of October 2011
  - end user dedicated certainly during next spring

Narval Support

- Narval School
  - first one $17^{th} - 21^{th}$ of October 2011
  - end user dedicated certainly during next spring
- http://forge.in2p3.fr/projects/narval

Narval Support

- Narval School
  - first one $17^{th} - 21^{th}$ of October 2011
  - end user dedicated certainly during next spring
- http://forge.in2p3.fr/projects/narval
- http://narval.in2p3.fr

Narval Support

- Narval School
  - first one $17^{th} - 21^{th}$ of October 2011
  - end user dedicated certainly during next spring
- http://forge.in2p3.fr/projects/narval
- http://narval.in2p3.fr
- Under Debian Squeeze or Ubuntu :
  - `apt-get install narval-doc`
  - `apt-get install narval-servers narval-utils narval-generic-actors`

Thanks for your attention
Questions ?