

Online Control and Configuration KM3NeT Design

VLVnT 2011

**S. Anvar, H. Le Provost, F. Château, F. Louis,
P. Sizun, Y. Moudden, V. Gautard, K. Ménager , E. Zonca & B. Vallage**

CEA Irfu – Saclay

On behalf of the KM3NeT Consortium

KM3NeT Detection Units

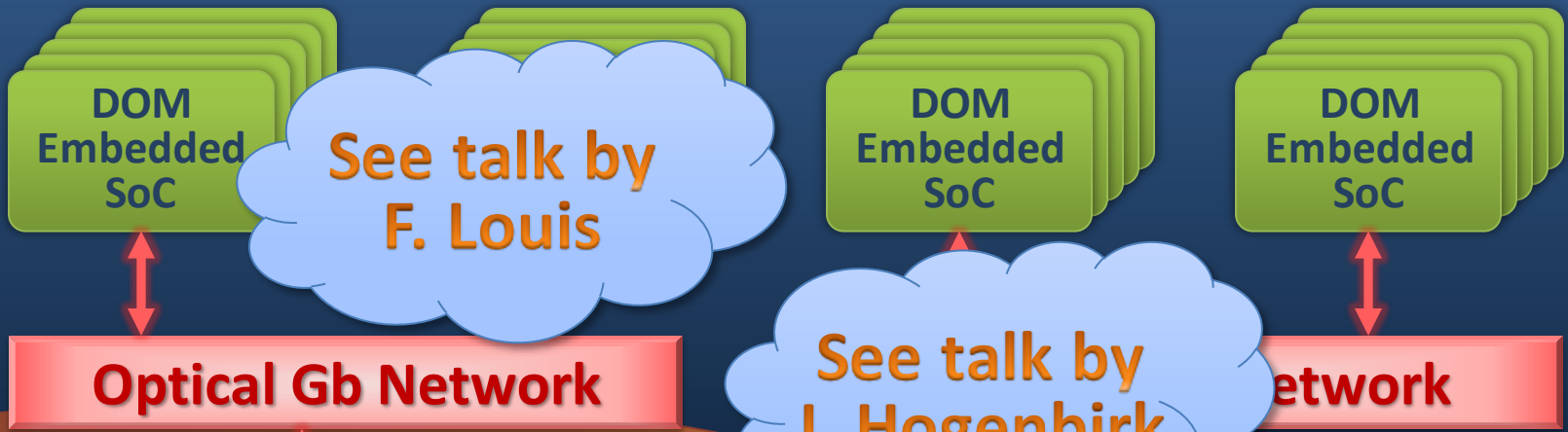
Storey

Digital Optical
Module
(DOM)

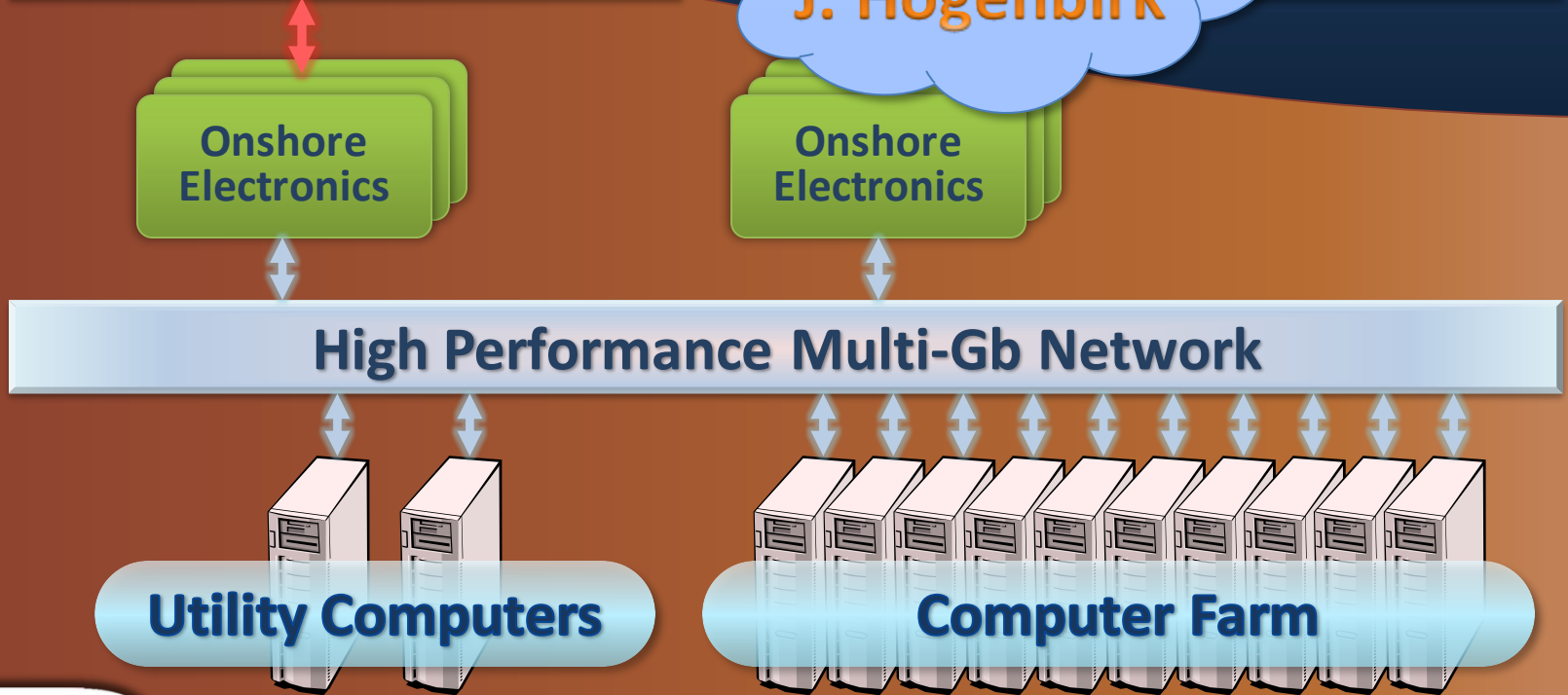
Complete telescope
~ 10000 DOMs

Network Topology

Offshore



Onshore



Online Processes

Offshore

Offshore
DAQ Threads
(VxWorks)

Offshore
DAQ Threads
(VxWorks)

Offshore
DAQ Threads
(VxWorks)

Offshore
DAQ Threads
(VxWorks)

Optical Gb Network

Optical Gb Network

Onshore
Electronics

Onshore
Electronics

High Performance Multi-Gb

Control &
Configurati...

Monitoring
(Linux,
Smartphone...)

Data Routers
(Linux)

Data Filters
(Linux)

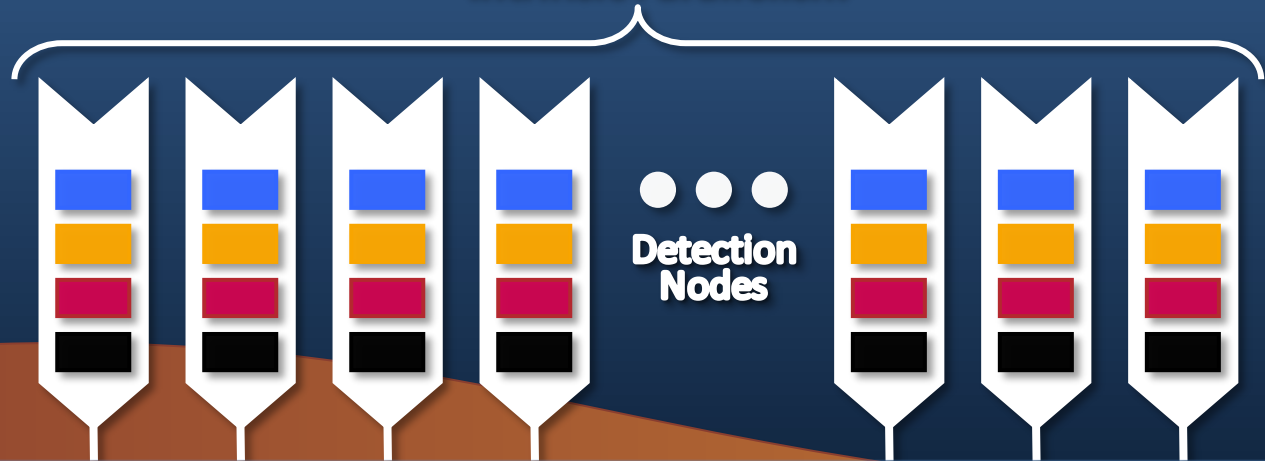
Data Managers
(Linux)

See talk by
A. Papaikonomou

Time-slice building

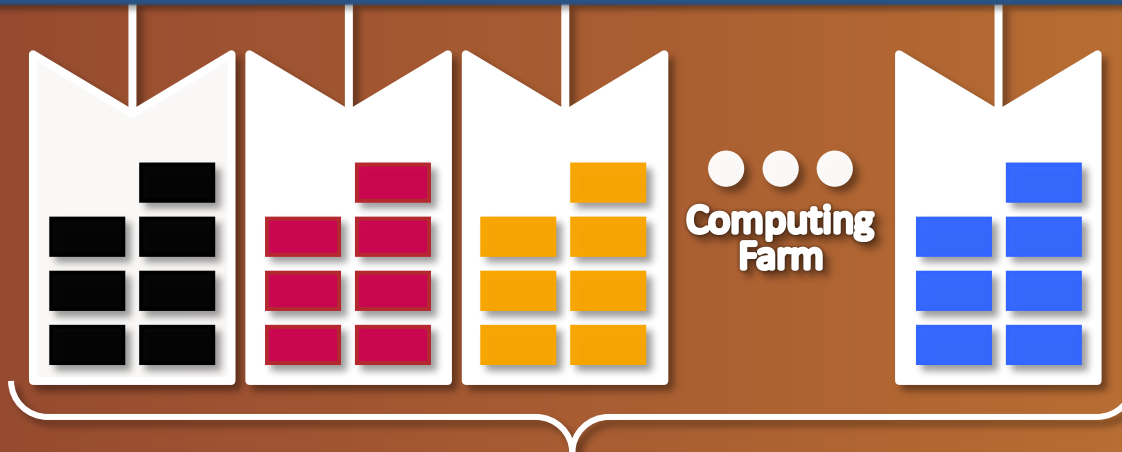
Offshore

Intrinsic Parallelism



Onshore

Switch Fabric



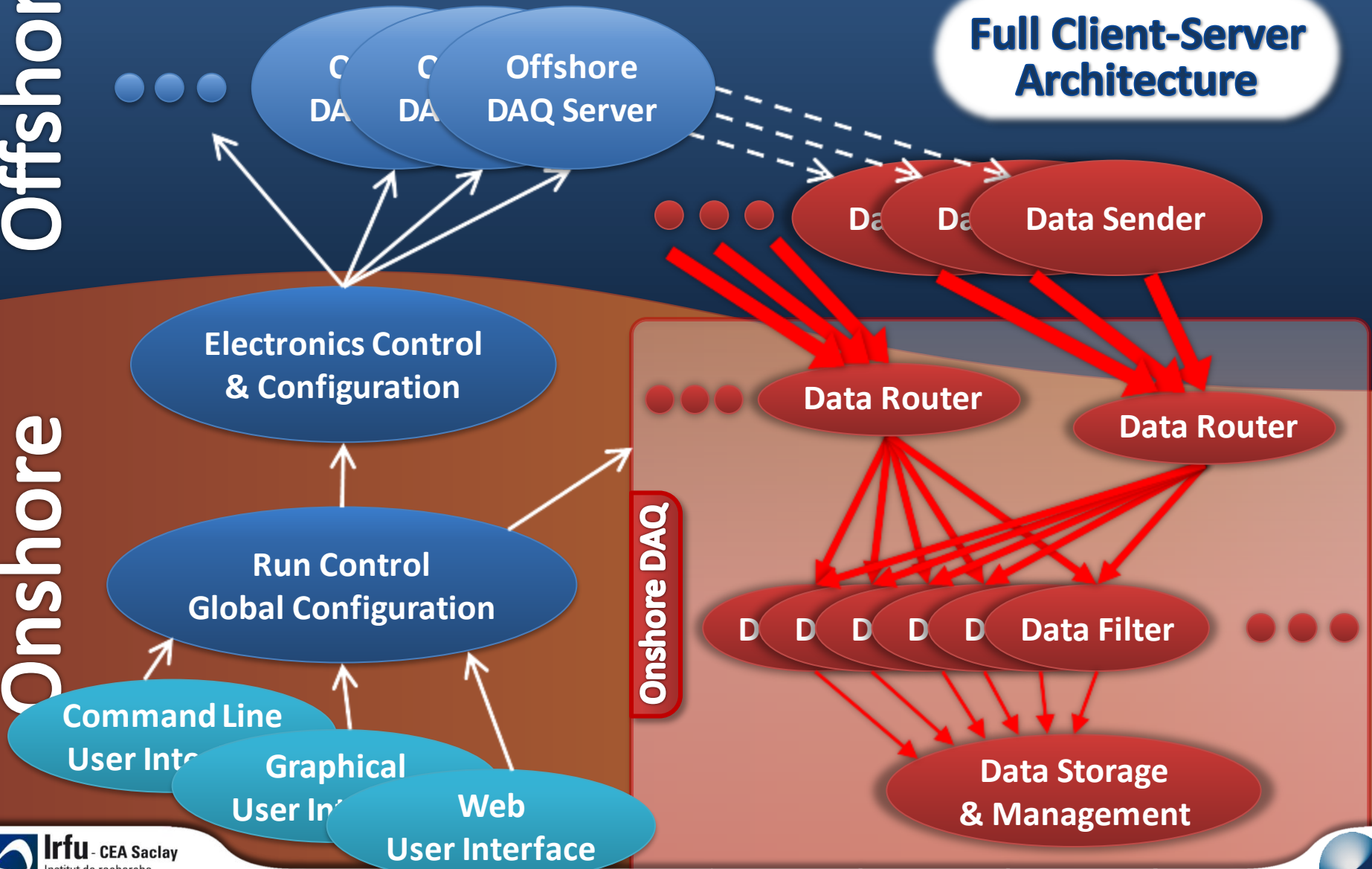
Performance Parallelism

Online Process Architecture

Offshore

Onshore

Full Client-Server Architecture



Onshore DAQ

Massively Distributed Application

~10000 processes

Client-Server Architecture

Middleware: Internet Communication Engine

TCP Protocol

IP Networking

Switched Ethernet

ICE Middleware

~10000 processes

Client-Server Architecture

Middleware: Internet Communication Engine

Open Source

Embedded

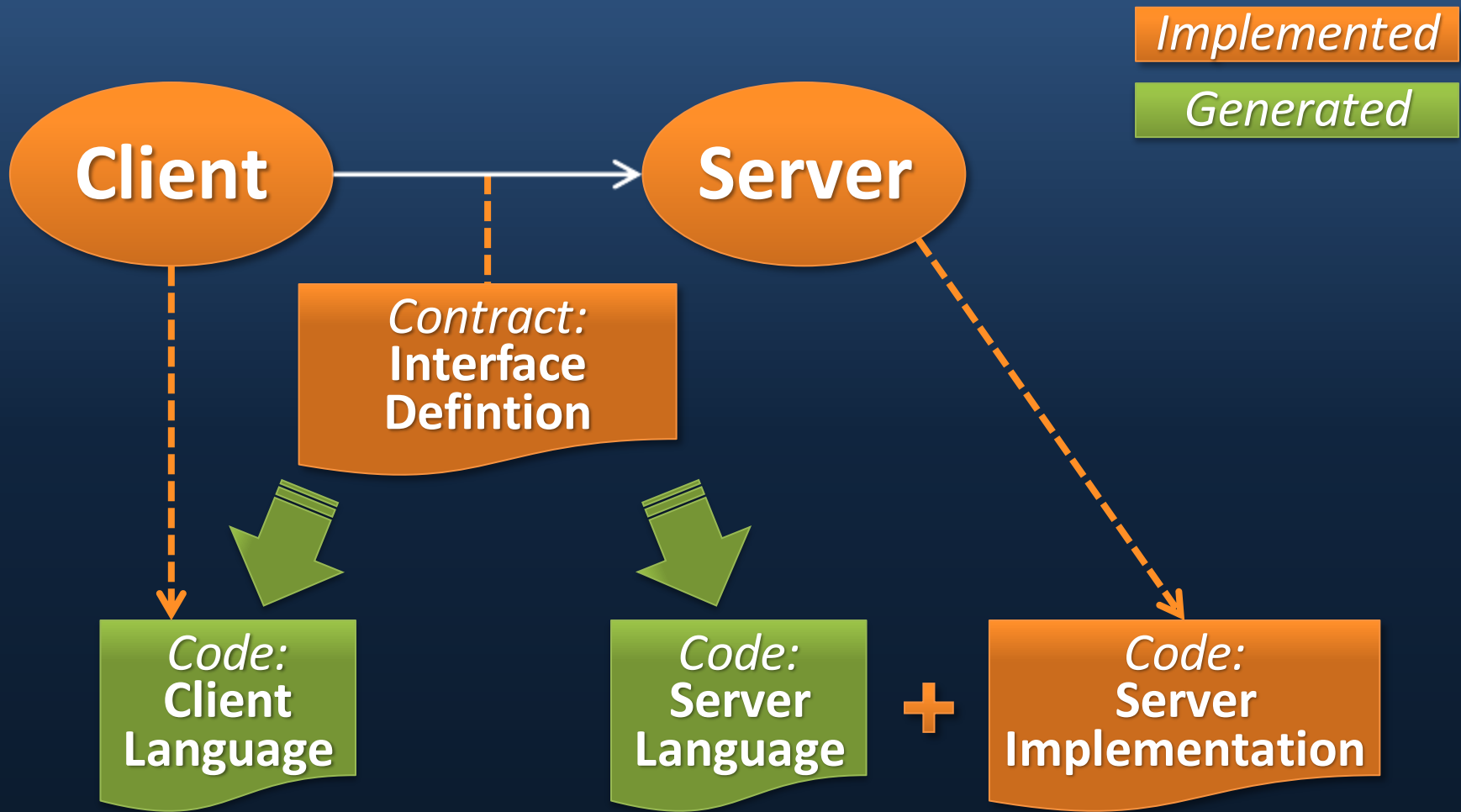
Object
Oriented

Multi
Language:
C++, Java, Python,
Objective-C, C#,
Ruby, PHP...

Ported to VxWorks
for KM3NeT (5 days)

Network Performance
Control & Acquisition

Client-Server over ICE



Client-Server over ICE

Offshore

Onshore

C++ VxWorks
Offshore
DAQ Server

No Network
Programming

C++ VxWorks
Data Sender

TCP/IP
Performance

C++ Linux
Electronics Control
& Configuration

Development by
Multiple Teams

C++ Linux
Data Router

C++ Linux
Run Control
Global Configuration

Management
Tools

C++ Linux
Data Filter

See talk by
A. Papaikonomou

Python PVM
Command Line
User Interface

Java Jvarkit
Graphical
User Interface

C++ Linux
Data Storage
& Management



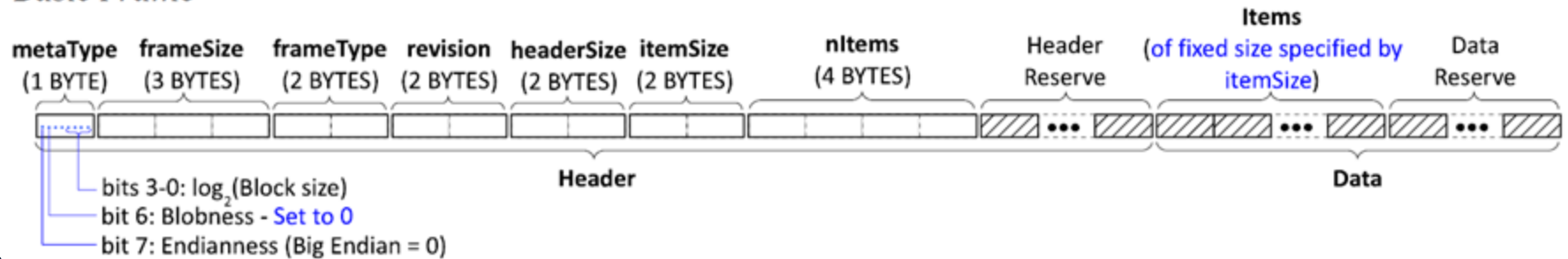
Acquisition Format

Multiframe Metaformat

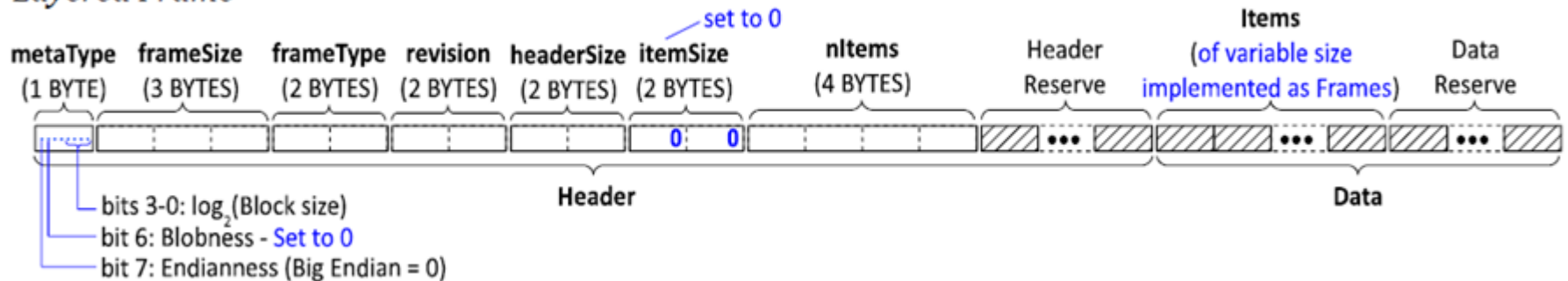
Generic, binary, versionable DAQ data format

Allows backward & forward compatible format evolution

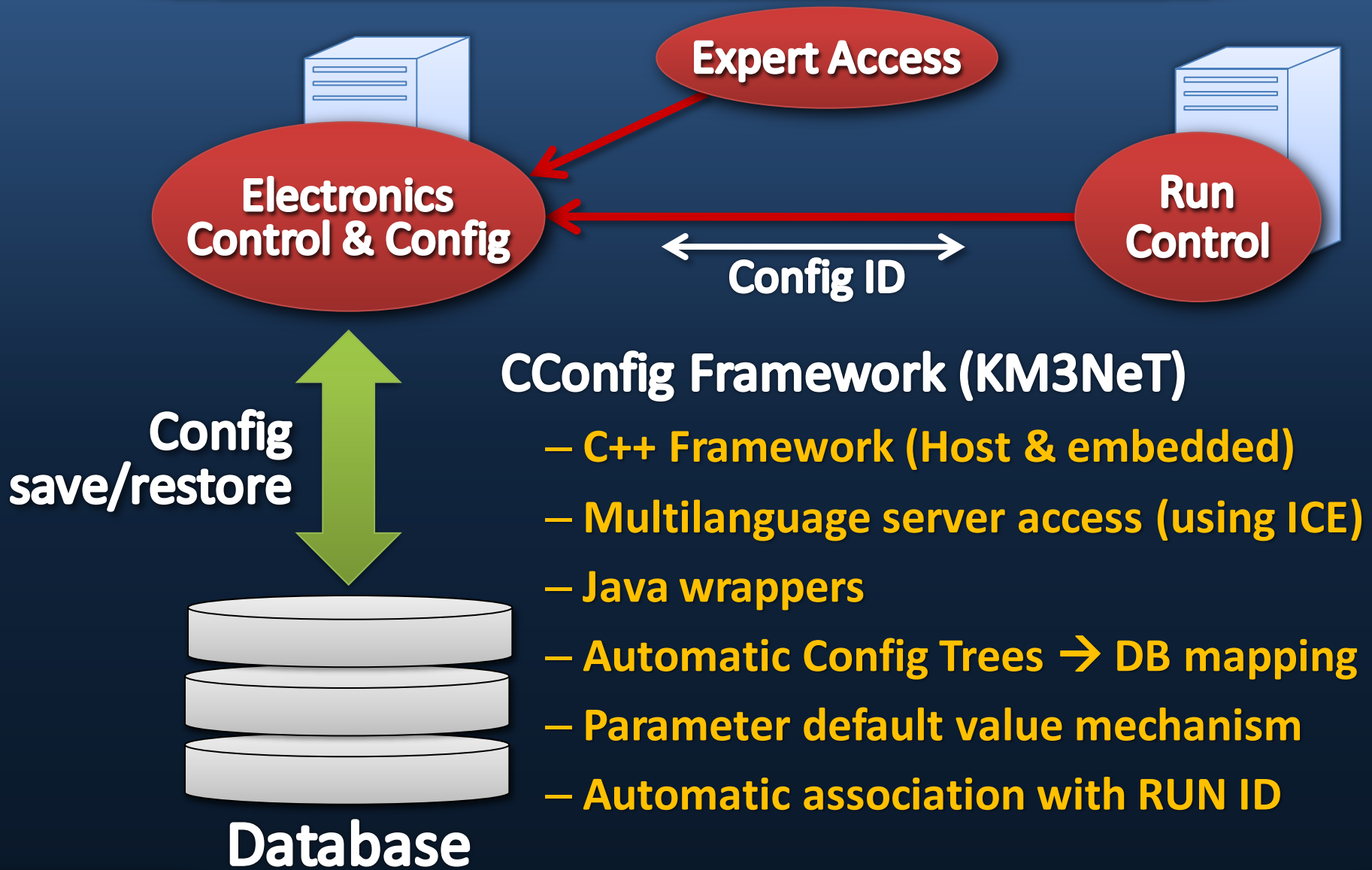
Basic Frame



Layered Frame



Configuration Framework



CConfig Framework (KM3NeT)

- C++ Framework (Host & embedded)
- Multilanguage server access (using ICE)
- Java wrappers
- Automatic Config Trees → DB mapping
- Parameter default value mechanism
- Automatic association with RUN ID

...

Configuration Framework

The image shows two windows from a software application. The 'Configuration editor' window displays a table of configuration parameters for 'KM3NetBench.xcfg'. The table has columns for Name, Value, Unit, Range, BL, and Include path. A blue arrow points to the 'eventRate[PMT]' value of '50.0 kHz' under 'Node[102] TDC_Chan[0]', labeled 'Default value'. A red arrow points to the 'isActive' value of 'true' under 'Node[102] TDC_Chan[1]', labeled 'Overriden value'. The 'KDAQ Run Control' window shows acquisition control buttons (Disconnect, STANDBY, Reset, Init, Configure, Start, Stop), a checkbox for 'Automatic Runs', and a 'Run Monitor' section with fields for Run time (00:01:23), Data (89.8 MB), and Events (3171). The 'Run ID' section includes Run Site (KoalaBench), Run Number (00037), and Date & Time (11.03.25@18:06:35).

Name	Value	Unit	Range	BL	Include path
PMT_SlowControl	true				
useZeroCopy	true				
daqSenderPrio	10		[1, 127]		
sampling	1.0				
doBarrelShifting	true				
stopTimeout	3000 ms		[10, 600000]		
NodeConfig					
Node[*]					
TDC_Chan[*]					
isActive	false				
eventRate[PMT]	50.0 kHz				
eventRate[ACOU]	0.0 kHz				
Node[102]					
TDC_Chan[0]					
isActive	true				
eventRate[PMT]	50.0 kHz				
eventRate[ACOU]	0.0 kHz				
TDC_Chan[1]					
isActive	true				
eventRate[PMT]	50.0 kHz			2	
eventRate[ACOU]	0.0 kHz			2	

```
#include "CCfg/CConfig.h"  
#include "CCfg/Document.h"
```

```
Ccfg::Document doc("/DataPath/KM3NetBench.xcfg");  
CCfg::CConfig cfg(doc.getConfig());
```

```
bool isActive = cfg("NodeConfig")("Node", 102)("TDC_Chan", 1);
```


Conclusion

- **Complete Control & Configuration Framework**
- **Tightly integrated with Data Acquisition**
- **Multi-language / Multi-OS**
- **Highly Modular**
 - Client Server Components
 - Rigorous Interfacing
 - Multiple Development Teams
 - Progressive / Continuous Integration
- **Ready by Spring 2012 for PPM-DU**