

Join Institute for Nuclear Research
Laboratory of High Energy Physics



School of Engineering
Dubna University

Benefits of a Microkernel Architecture for Slow Control Systems Software in Research Activities

Case Studies from LHEP, JINR

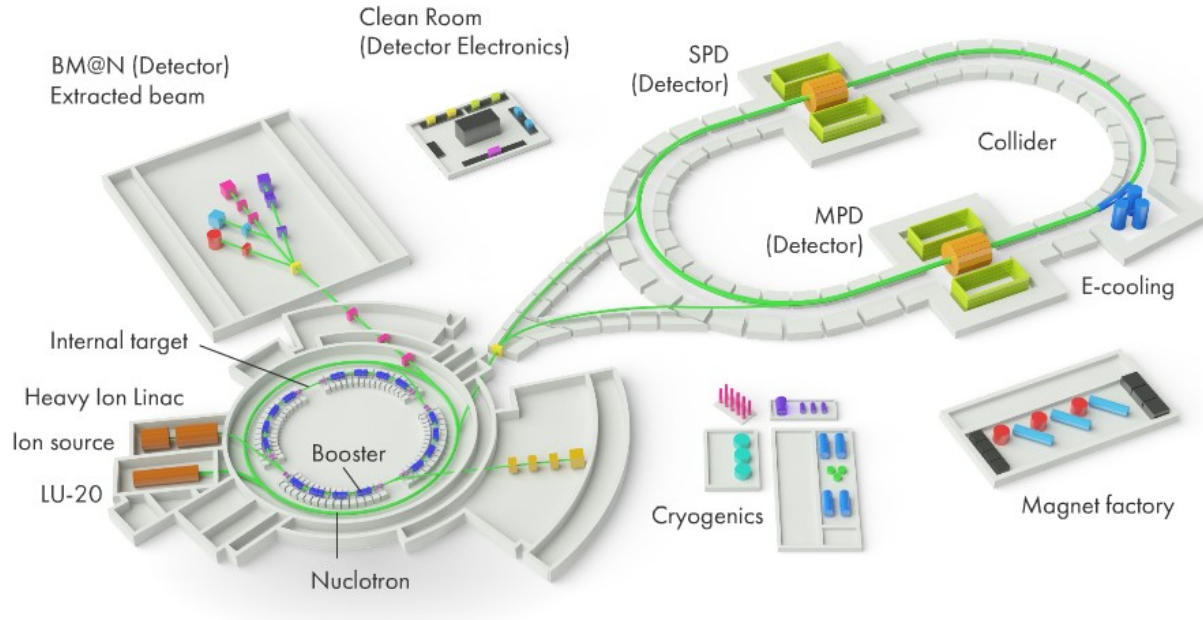
Authors: A. Kozlovsky, A. Bukharin, D. Ivliev, O. Melnikova, A. Noskov, Y. Smirnova, I. Shirikov,
I. Zhabin

Oral: Alexey Kozlovsky

AYSS-2025

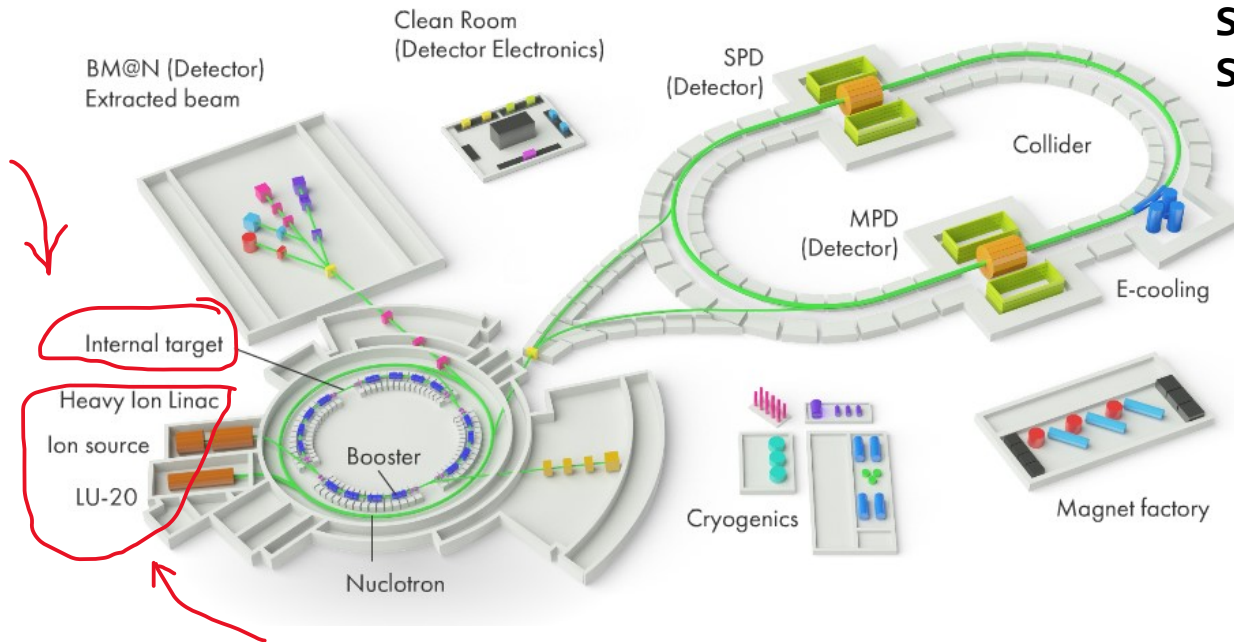
Context

Our team is developing and maintaining hardware and software for various systems at LHEP JINR



Context

Our team is developing and maintaining hardware and software for various systems at LHEP JINR

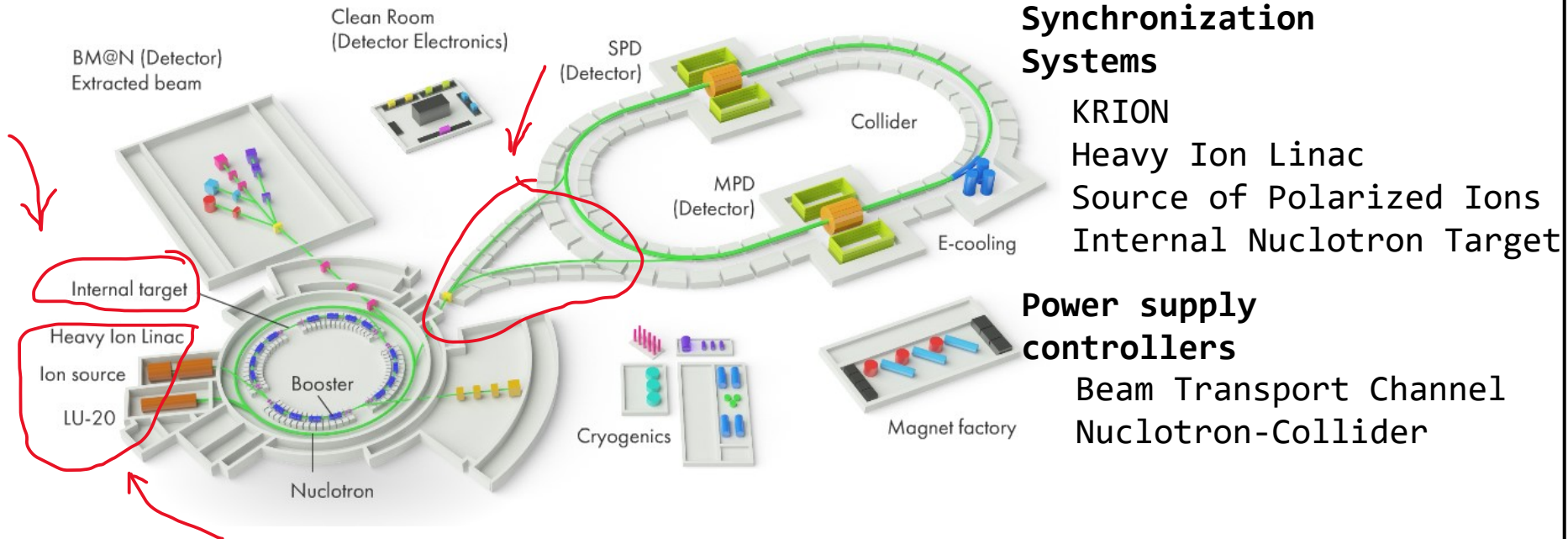


Synchronization Systems

KRION
Heavy Ion Linac
Source of Polarized Ions
Internal Nuclotron Target

Context

Our team is developing and maintaining hardware and software for various systems at LHEP JINR



Context



The hardware is **various** according to the aim, but a majority of modules are built on the **unified** platform

Software should be **unified** as well

Problem

Research activities **require specific hardware** maintained by staff hardware developers

The **complexity** of modern systems arouses the question of the need for the **bespoke software**

Both software and hardware need to be **flexible** to adapt to various constantly-changing requirements from end-users: **physicists**

Slow control systems

Slow control systems

Despite the fact that there is no general definition for a **slow control system** it can be described as the **opposite** of a **Real-Time system**

Slow control systems

Despite the fact that there is no general definition for a **slow control system** it can be described as the **opposite** of a **Real-Time system**

We have a lot of tasks related to slow control systems at LHEP

Slow control systems

Despite the fact that there is no general definition for a **slow control system** it can be described as the **opposite** of a **Real-Time system**

We have a lot of tasks related to slow control systems at LHEP

Often we used to build the standalone software for such kind of systems from scratch

But can we unify the software developing process in such context?

Our past experience

Система синхронизации (Jera)

Таблица синхронизации Настройки

Конфигурация

Открыть конфигурацию

Название конфигурации

Ера

☐ Скрывать окна

Описание конфигурации

Egor

Открыть логи

Настройки контроллера

IP крейта Порт крейта

192.168.127.1 4001

Подключить Отключить

Чтение конф-ии Перезапись конф-ии

Сохранить изменения Вернуть настройки

Параметры блока питания

Вкл. Выкл.

Канал +24 В Канал -24 В

Напряжение Ток

Источник

PoF ☐ Сеть ☐ Авария

OVP ☐ ОСР ☐

Настройки модулей

	Тип	Адрес
1	БПК-31	1
2	ОТ-2	2
3	T8-31	3
4	T8-31	4

Редактировать список

Настройки модуля

Автоконфиг

Статус подключения

We have started with the maintaining of **KRION** synchronization system software in 2021

But it **was not scalable** enough

The idea of unifying the software and implementing it into other systems came to mind

Solution

Solution

We decide to build our own platform

Solution

We decide to build our own platform

1. Integration

Solution

We decide to build our own platform

1. Integration
2. Evolutionary

Solution

We decide to build our own platform

1. Integration
2. Evolutionary
3. Simplicity

Solution

We decide to build our own platform

According to Neal Ford and Mark Richards, Microkernel Architecture would satisfy our quality attribute requirements

1. Integration
2. Evolutionary
3. Simplicity

Architecture characteristic	Star rating
Partitioning type	Domain and technical
Number of quanta	1
Deployability	★ ★ ★
Elasticity	★
Evolutionary	★ ★ ★
Fault tolerance	★
Modularity	★ ★ ★
Overall cost	★ ★ ★ ★ ★
Performance	★ ★ ★
Reliability	★ ★ ★
Scalability	★
Simplicity	★ ★ ★ ★
Testability	★ ★ ★

Solution

We decide to build our own platform

According to Neal Ford and Mark Richards, Microkernel Architecture would satisfy our quality attribute requirements

1. Integration
2. Evolutionary
3. Simplicity

Microkernel Architecture == Plugin Architecture

Architecture characteristic	Star rating
Partitioning type	Domain and technical
Number of quanta	1
Deployability	★ ★ ★
Elasticity	★
Evolutionary	★ ★ ★
Fault tolerance	★
Modularity	★ ★ ★
Overall cost	★ ★ ★ ★ ★
Performance	★ ★ ★
Reliability	★ ★ ★
Scalability	★
Simplicity	★ ★ ★ ★
Testability	★ ★ ★

Solution

We decide to build our own platform

According to Neal Ford and Mark Richards, Microkernel Architecture would satisfy our quality attribute requirements

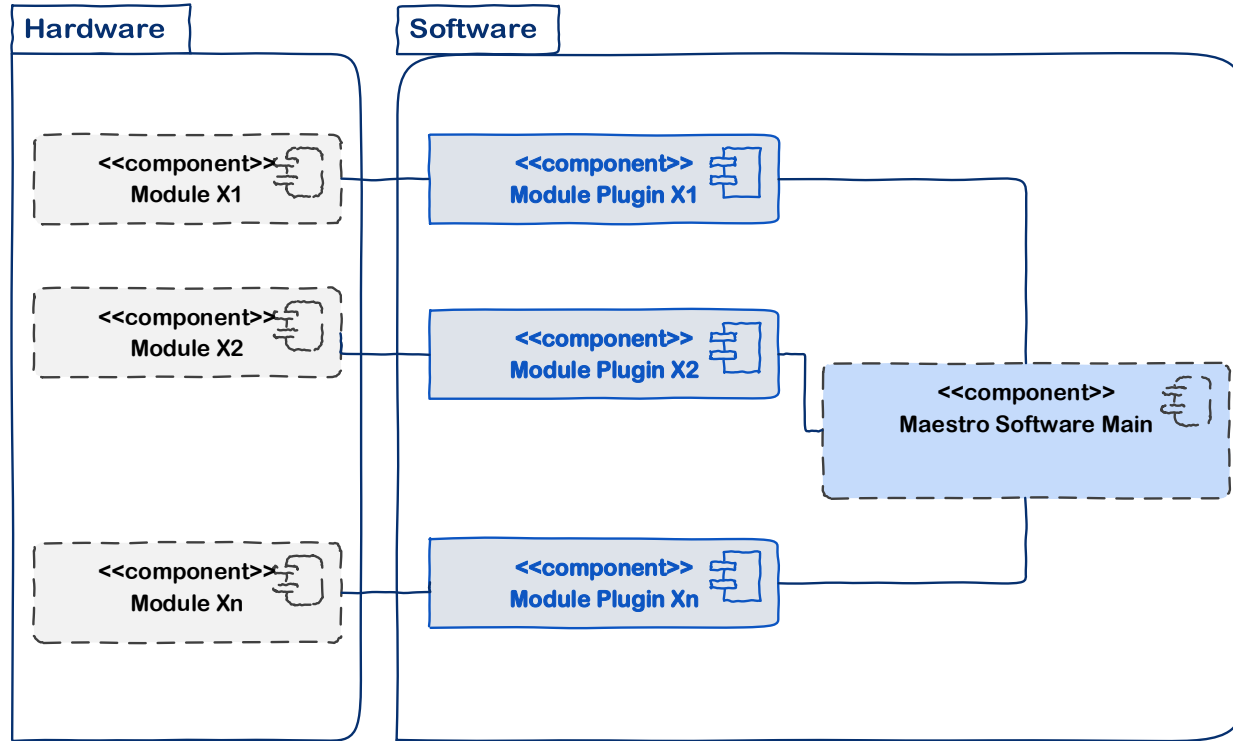
1. Integration
2. Evolutionary
3. Simplicity

Microkernel Architecture == Plugin Architecture

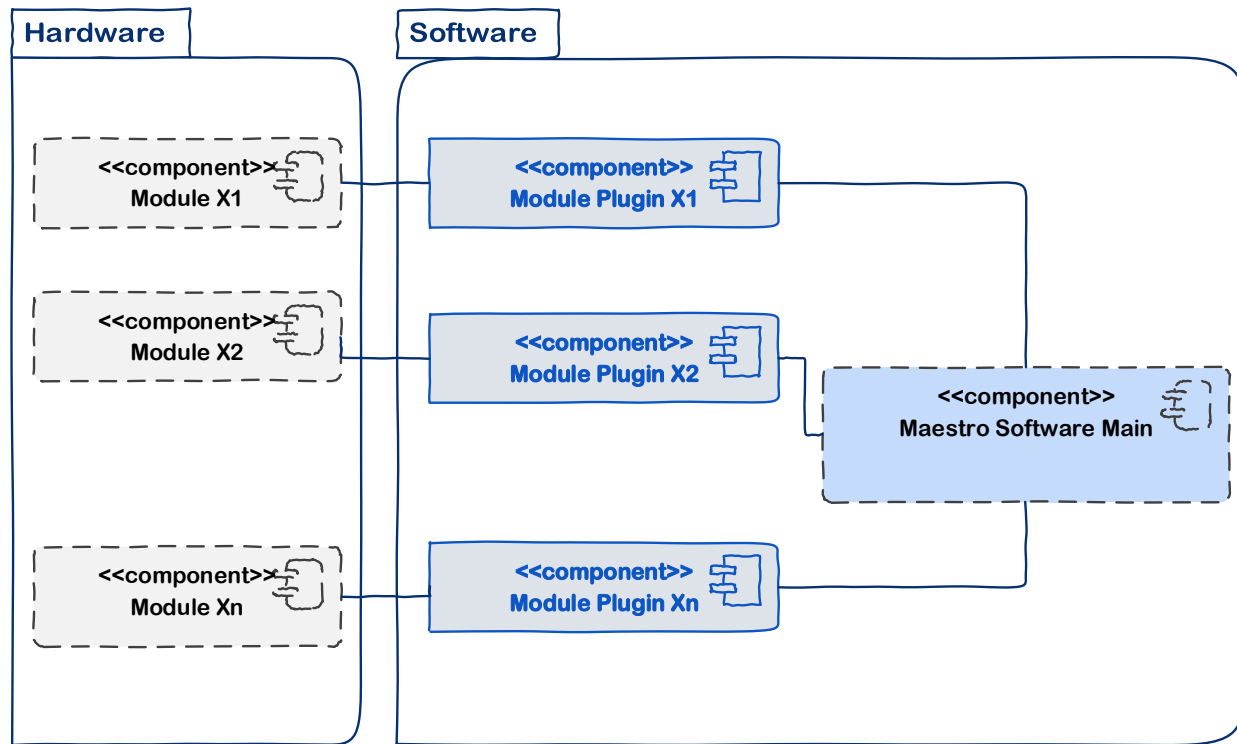
Software module for controlling hardware device
== Plugin

Architecture characteristic	Star rating
Partitioning type	Domain and technical
Number of quanta	1
Deployability	★ ★ ★
Elasticity	★
Evolutionary	★ ★ ★
Fault tolerance	★
Modularity	★ ★ ★
Overall cost	★ ★ ★ ★ ★
Performance	★ ★ ★
Reliability	★ ★ ★
Scalability	★
Simplicity	★ ★ ★ ★
Testability	★ ★ ★

Implementation - Maestro Software

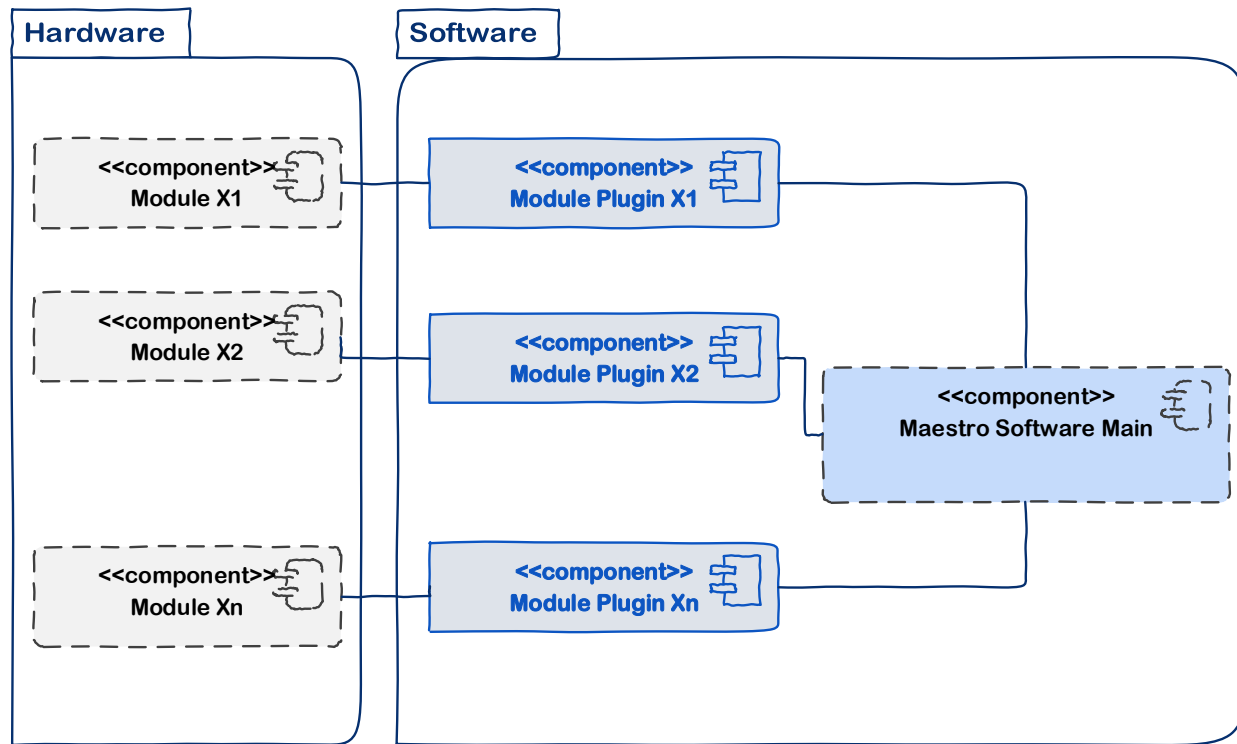


Implementation - Maestro Software



Every plugin is a dynamic library (.so or .dll)

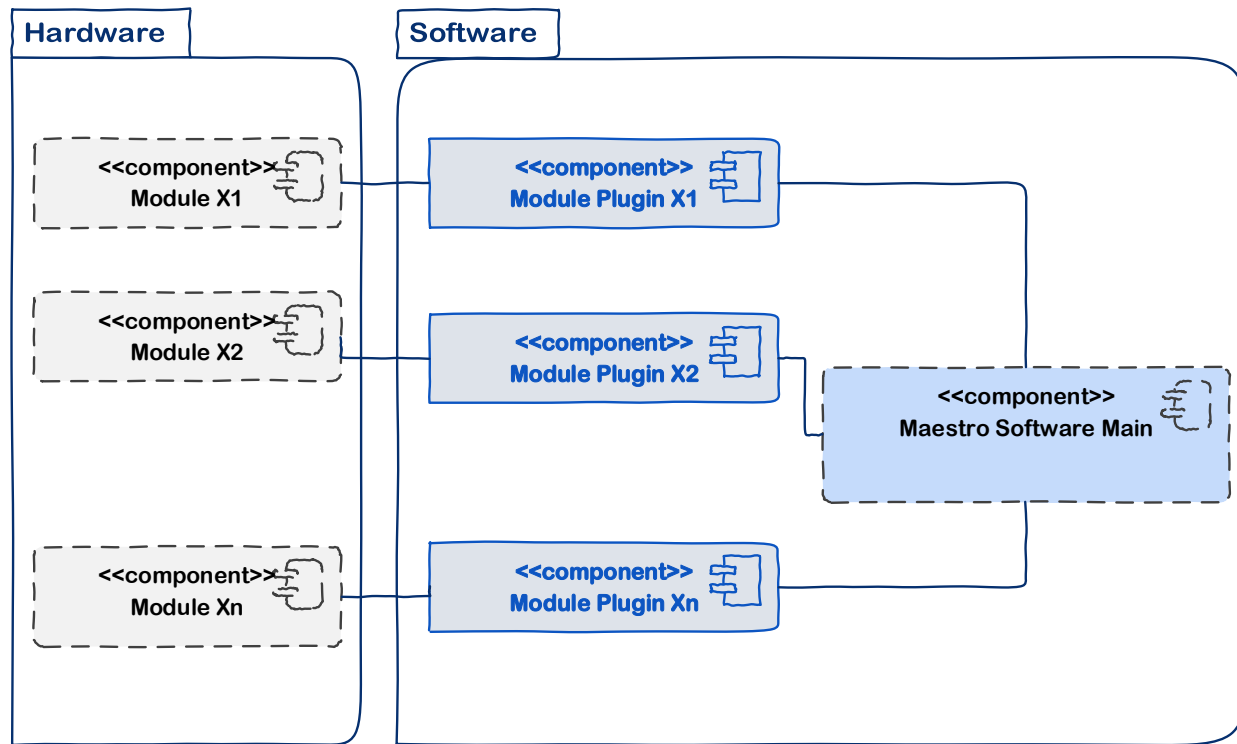
Implementation - Maestro Software



Every plugin is a dynamic library (.so or .dll)

For **each** hardware module we have its software **adaptor**

Implementation - Maestro Software

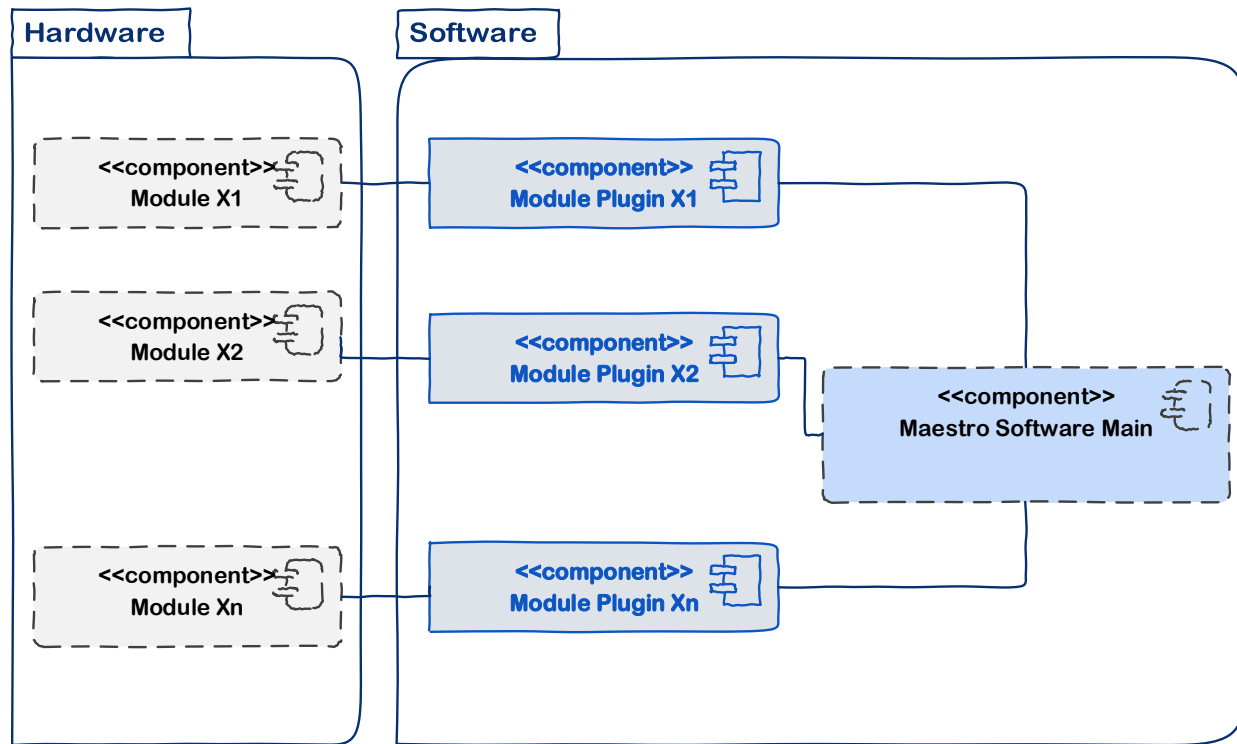


Every plugin is a dynamic library (.so or .dll)

For **each** hardware module we have its software **adaptor**

The plugins should simply be placed into a specific directory

Implementation - Maestro Software



Every plugin is a dynamic library (.so or .dll)

For **each** hardware module we have its software **adaptor**

The plugins should simply be placed into a specific directory



Implementation – Maestro Software

Конфиг Управление подключениями Настройки Инструменты

Устройства

- K1 ID1 БПК-31
- K1 ID10 Timer PM25/Крион
- K1 ID2 OY
- K1 ID3 Timer L1,L2
- K1 ID4 Timer D1,D2
- K1 ID5 Timer T1,T2
- K1 ID6 Timer T3
- K1 ID7 Timer T4
- K1 ID8 Timer Ш.диагн./Инф.пл

Имя устройства K1 ID2 OY

Описание устройства

Разработчик устройства

Версия прошивки unknown

Имя плагина OUGUPluginKRION

Описание плагина OUGU Standalone Plug

Разработчик плагина Alexey Kozlovsky: a)

Версия плагина undefined

Панель мониторинга

K1 ID2 OY

Номер раба 0 Номер цикла 0 Период цикла 0 с

Цикл разрешен: ■

К1 ID2 OY

Таблица синхронизации Управление устройством

	№	Устройство	Имя	Задержка	Длительность	Вкл.	Инов.
1	A1	K1 ID10	Синх. ...	93 мс	0.025 мс	да	нет
2	A2	K1 ID10	СОЧИ Зап...	100 нс	1 мс	да	нет
3	A3	K1 ID10	1.15.3	25 мкс	25 мкс	нет	нет
4	A4	K1 ID10	1.15.4	25 мкс	25 мкс	нет	нет
5	B1	K1 ID10	Крион ...	13.3 мс	10 мкс	да	нет
6	B2	K1 ID10	1.15.6	25 мкс	10 мкс	нет	нет
7	B3	K1 ID10	1.15.7	25 мкс	25 мкс	нет	нет
8	B4	K1 ID10	1.15.8	25 мкс	0.025 мс	нет	нет
9	A1	K1 ID3	L1*	89.3 мс	0.01 мс	да	нет
10	A2	K1 ID3	Старт АЦ...	88 мс	0.01 мс	да	нет
11	A3	K1 ID3	Изм. ток...	93.315 мс	0.01 мс	да	нет
12	A4	K1 ID3	--	2.3 мс	0.01 мс	нет	нет
13	B1	K1 ID3	L2*	90.8 мс	0.01 мс	да	нет
14	B2	K1 ID3	Старт АЦ...	90 мс	0.01 мс	да	нет
15	B3	K1 ID3	Изм. ток...	93.315 мс	0.01 мс	да	нет
16	B4	K1 ID3	--	111.67 мс	0.01 мс	нет	нет
17	A1	K1 ID4	Старт АЦ...	91.9 мс	10 мкс	да	нет
18	A2	K1 ID4	Изм. ток...	93.315 мс	10 мкс	да	нет
19	A3	K1 ID4	D1.1*	92.1 мс	10 мкс	да	нет
20	A4	K1 ID4	D1.2*	92.1 мс	10 мкс	да	нет

Software examples from KRION and HILAC Synchronization Systems

Конфиг Управление подключениями Настройки Инструменты

Панель мониторинга

K1 ID2 OY

Номер раба 0 Номер цикла 0 Период цикла 0 с

Цикл разрешен: ■

К1 ID2 OY

Таблица синхронизации Управление устройством

Период 1800 мс Включение ■

Установить

Формирователь внутренних

Длительность 100000 мс

Установить

Количество импульсов 0 шт.

Период серии 100075 мс

Установить

Задержка Криона общ.

Источник запуска 3. INO_PRE_SER

Задержка от старта 300 мс

Установить

Вкл. ■

1

К1. Инов. ионов в ЗП

Задержка вывода ионов

Задержка от старта 400 мс

Установить

Длительность 100000 мс

Установить

Автом.

или

ГЛУ

1

К2. Экстракция ионов из ЗП

Challenges

As we built **our SDK** with 7 libraries there were an issue with the complexity of maintaining which consumes a lot of time resources

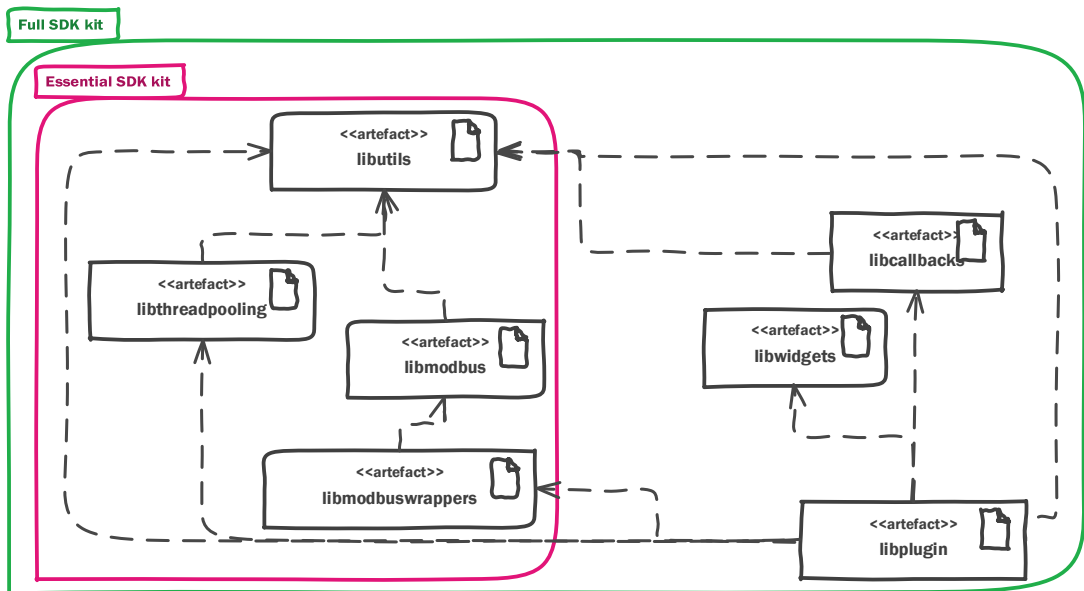
For solving this problem we implemented our **own** self-hosted conan **package manager***

* See the talk of
Denis Ivliev

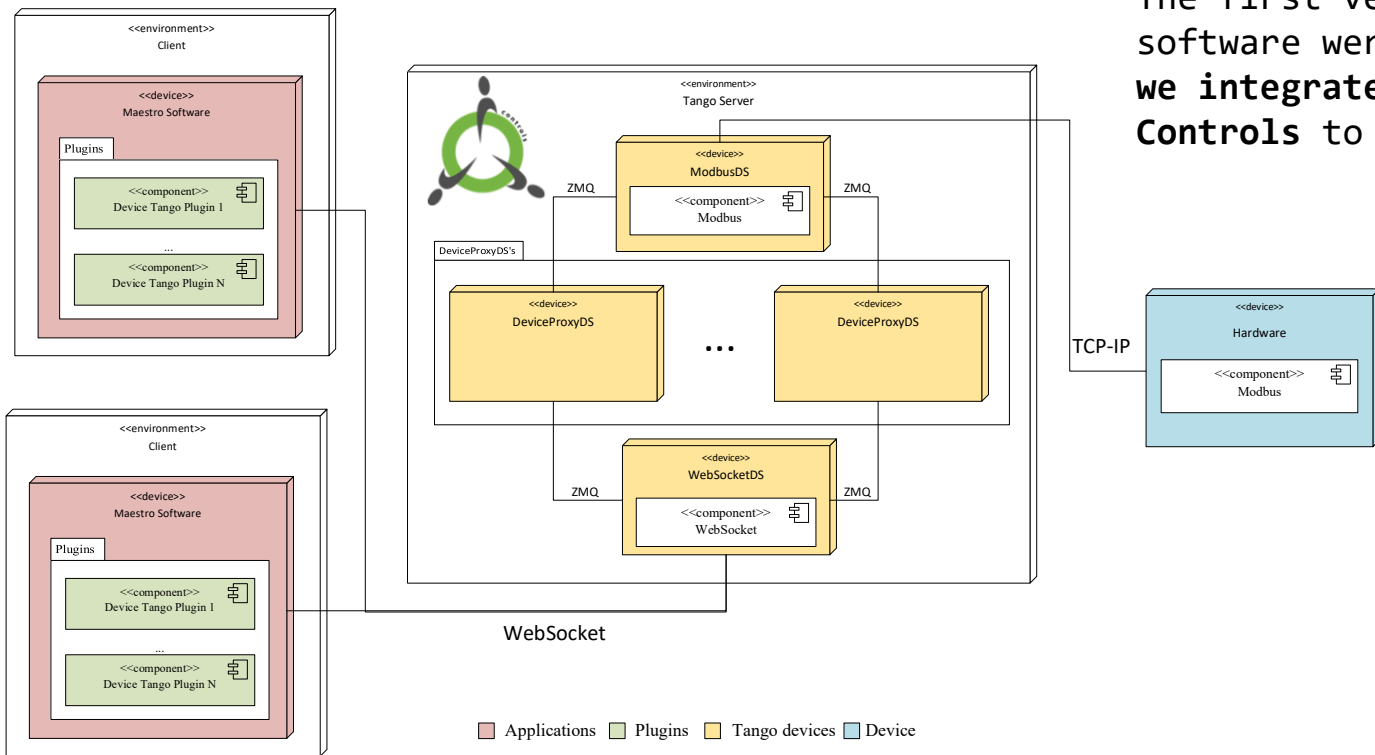


CONAN 2.0

C/C++ Package Manager



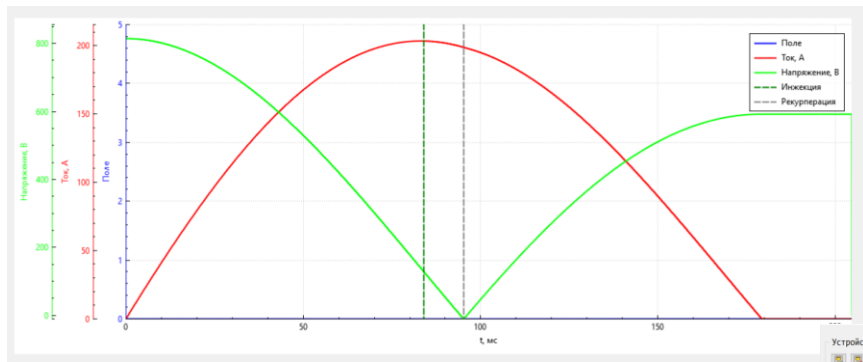
Challenges



The first versions of our software were standalone, but we integrated it with Tango Controls to make it distribute

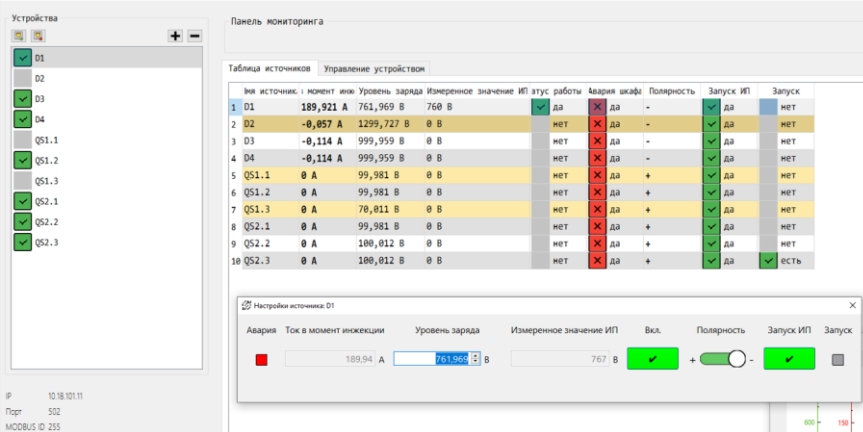
* See the talk of Alexey Bukharin

What about other systems?



Despite the initial orientation towards the synchronization system, our unified platform can be customized to fit the aims of controlling the power supplies of the Nuclotron-Collider Beam Transport Channel

* See the talk of
Artyom Noskov



Have we satisfied our QA?

Have we satisfied our QA?

Integration



We have integrated our software with Tango Controls, and it may be used with various hardware

Have we satisfied our QA?

Integration



We have integrated our software with Tango Controls, and it may be used with various hardware

Evolutionary



We have adapted the software for Nuclotron Collider Beam Transport Channel Power Supplies control and our system has grown functional

Have we satisfied our QA?

Integration



We have integrated our software with Tango Controls, and it may be used with various hardware

Evolutionary



We have adapted the software for Nuclotron Collider Beam Transport Channel Power Supplies control and our system have grown functionally

What about SIMPLICITY?

School of Engineering for LHEP



School of Engineering
Dubna University

We have employed three students from the School of Engineering 1.5 year ago and now they are **indispensable** members of our team

School of Engineering for LHEP



School of Engineering
Dubna University

We have employed three students from the School of Engineering 1.5 year ago and now they are **indispensable** members of our team

Educational course has been developed for engaging the students into the process of slow control system software development

УНИВЕРСИТЕТ «ДУБНА»

26.02 - 12.03

**Введение
в разработку ПО управления
физическими установками**

КУРС ПОВЫШЕНИЯ КВАЛИФИКАЦИИ

26/02 - 12/03

Период обучения

ПН - ПТ

Дни обучения

18:00 - 20:00

Время обучения

+79057040929
smirnova.ya.v@uni-dubna.ru

КОНТАКТЫ

УНИВЕРСИТЕТ «ДУБНА»

School of Engineering for LHEP



School of Engineering
Dubna University

We have employed three students from the School of Engineering 1.5 year ago and now they are **indispensable** members of our team

Educational course has been developed for engaging the students into the process of slow control system software development

Maestro Software Platform architecture has been built on such principles that have made it possible to describe the future plugin easy even for students

УНИВЕРСИТЕТ «ДУБНА»

26.02 - 12.03

**Введение
в разработку ПО управления
физическими установками**

КУРС ПОВЫШЕНИЯ КВАЛИФИКАЦИИ

26/02 - 12/03

Период обучения

ПН - ПТ

Дни обучения

18:00 - 20:00

Время обучения

+79057040929
smirnova.ya.v@uni-dubna.ru

КОНТАКТЫ

УНИВЕРСИТЕТ «ДУБНА»

School of Engineering for LHEP



School of Engineering
Dubna University

We have employed three students from the School of Engineering 1.5 year ago and now they are **indispensable** members of our team

Educational course has been developed for engaging the students into the process of slow control system software development

Maestro Software Platform architecture has been built on such principles that have made it possible to describe the future plugin easy even for students

It has become possible to delegate certain tasks of developing software for hardware module control to a student!

УНИВЕРСИТЕТ «ДУБНА» — 26.02 - 12.03

**Введение
в разработку ПО управления
физическими установками**

КУРС ПОВЫШЕНИЯ КВАЛИФИКАЦИИ

26/02 - 12/03

Период обучения

ПН - ПТ

Дни обучения

18:00 - 20:00

Время обучения

+79057040929

smirnova.ya.v@uni-dubna.ru

КОНТАКТЫ



УНИВЕРСИТЕТ «ДУБНА»

School of Engineering for LHEP



School of Engineering
Dubna University

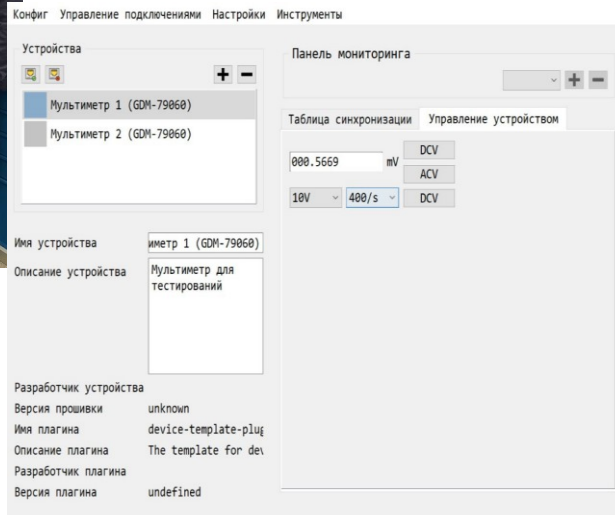
The new generation of SoE
students now is working on the
software for Measure Unit
Hardware control

School of Engineering for LHEP



School of Engineering
Dubna University

The new generation of SoE students now is working on the software for Measure Unit Hardware control



They are developing the **Maestro Software plugins** with SCPI commands support by themselves

Results

The unified platform for client or standalone software for slow control system management has been developed and implemented in different parts of NICA Complex

The software has been integrated with Tango Controls

The system of Maestro Software Plugins **description** has been composed and successfully tested within the framework of the **School of Engineering**



School of Engineering
Dubna University

Thank you for your attention

Alexey Kozlovsky kozlovsky@jinr.ru