Next-Generation MADOCA for The SPring-8 Control Framework

T. Matsumoto, Y. Furukawa, M. Ishii
JASRI/SPring-8

Outline

- Introduction to MADOCA
- MADOCA II Messaging
- MADOCA II Applications
- Summary

Introduction to MADOCA

- Control Framework originally developed at SPring-8, Japan
- Adopted in several facilities for various control applications
  - SPring-8, SACLA, NewSUBARU, HiSOR
  - Accelerator, beamline and experimental station controls
- Successfully utilized since 1997

*Photos courtesy of RIKEN/JASRI

Introduction to MADOCA

- Control Framework originally developed at SPring-8, Japan
  - **M**essage **A**nd **D**atabase **O**riented **C**ontrol **A**rchitecture
- Adopted in several facilities for various control applications
  - SPring-8, SACLA, NewSUBARU, HiSOR
  - Accelerator, beamline and experimental station controls
- Successfully utilized since 1997

*Photos courtesy of RIKEN/JASRI

MADOCA Messaging

- Based on a text command with S/V/O/C syntax
  - **S**ubject, **V**erb, **O**bject, **C**omplement
  - Examples:
    - <S>/put/sr_mag_ps_b/on
    - <S>/get/sr_vac_ivg_19_ab3/pressure
  - Unique for object name to identify control equipment
  - <S> = 123_matumot_oprgui_opcon01 (example)
    → from PID, user name, application name and host name
  - Replies are returned from the equipment

MADOCA Messaging

- Based on a text command with S/V/O/C syntax
  - **S**ubject, **V**erb, **O**bject, **C**omplement
  - Examples:
    - <S>/put/sr_mag_ps_b/on
    - <S>/get/sr_vac_ivg_19_ab3/pressure
  - Unique for object name to identify control equipment
  - <S> = 123_matumot_oprgui_opcon01 (example)
    → from PID, user name, application name and host name
  - Replies are returned from the equipment
    - sr_mag_ps_b/put/<S>/ok
    - sr_vac_ivg_19_abs/get/<S>/1.23E-09Pa

MADOCA Messaging

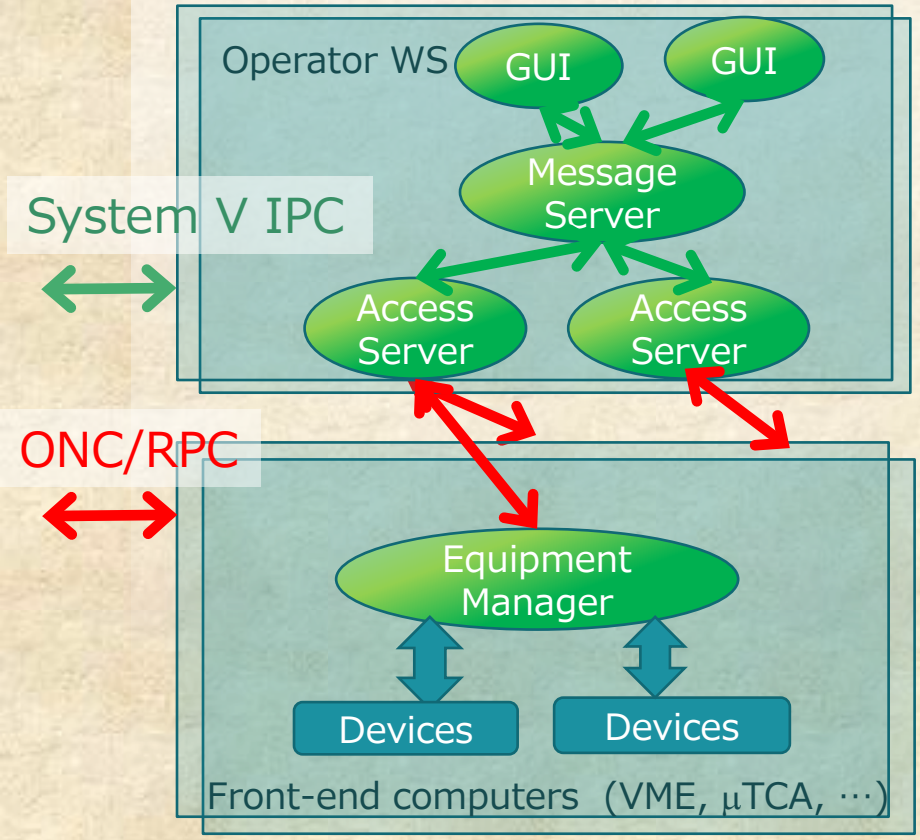
- Based on a text command with S/V/O/C syntax
  - **S**ubject, **V**erb, **O**bject, **C**omplement
  - Examples:
    - <S>/put/sr_mag_ps_b/on
    - <S>/get/sr_vac_ivg_19_ab3/pressure
  - Unique for object name to identify control equipment
  - <S> = 123_matumot_oprgui_opcon01 (example)
    → from PID, user name, application name and host name
  - Replies are returned from the equipment
    - sr_mag_ps_b/put/<S>/ok
    - sr_vac_ivg_19_abs/get/<S>/1.23E-09Pa
- Human-readable messages (abstracted)
  - Do not include specific controls on each device

MADOCA Messaging

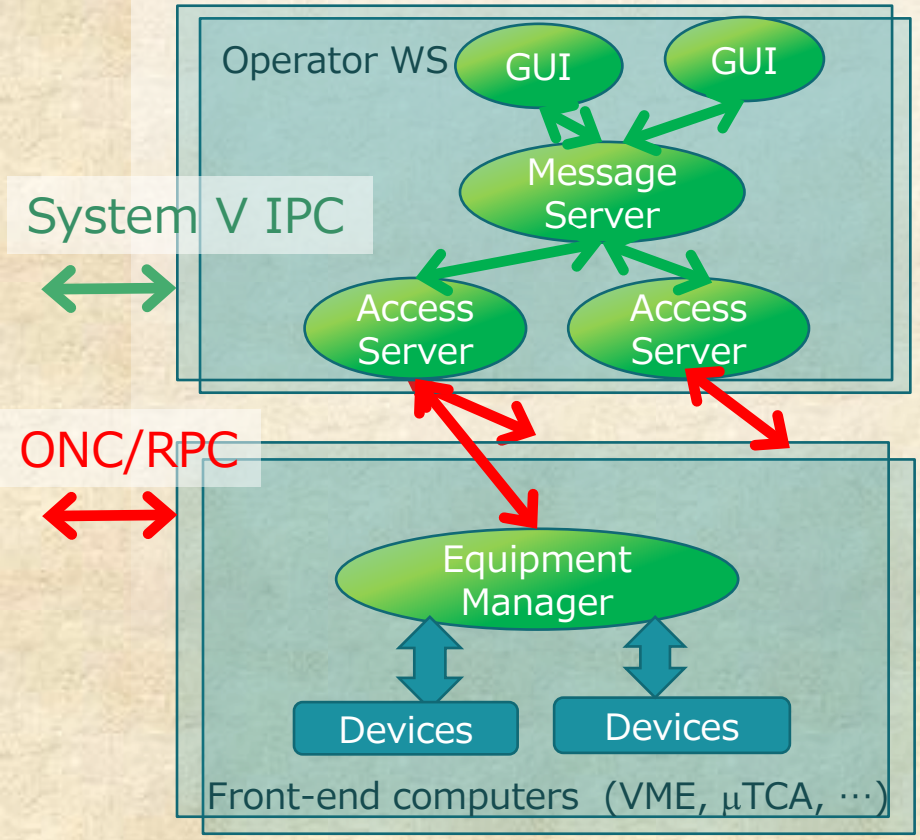
- Based on a text command with S/V/O/C syntax
  - **S**ubject, **V**erb, **O**bject, **C**omplement
  - Examples:
    - <S>/put/sr_mag_ps_b/on
    - <S>/get/sr_vac_ivg_19_ab3/pressure
  - Unique for object name to identify control equipment
  - <S> = 123_matumot_oprgui_opcon01 (example)
    → from PID, user name, application name and host name
  - Replies are returned from the equipment
    - sr_mag_ps_b/put/<S>/ok
    - sr_vac_ivg_19_abs/get/<S>/1.23E-09Pa
- Human-readable messages (abstracted)
  - Do not include specific controls on each device

→ These features are essential in our control system

MADOCA Control Framework

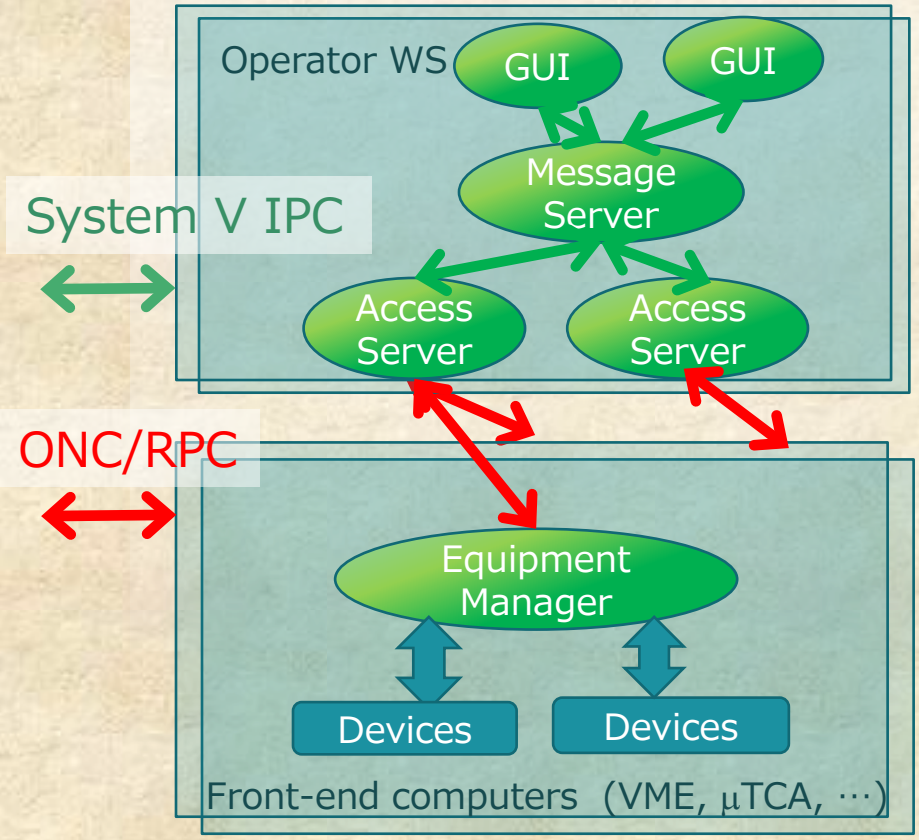


MADOCA Control Framework



Shortcomings

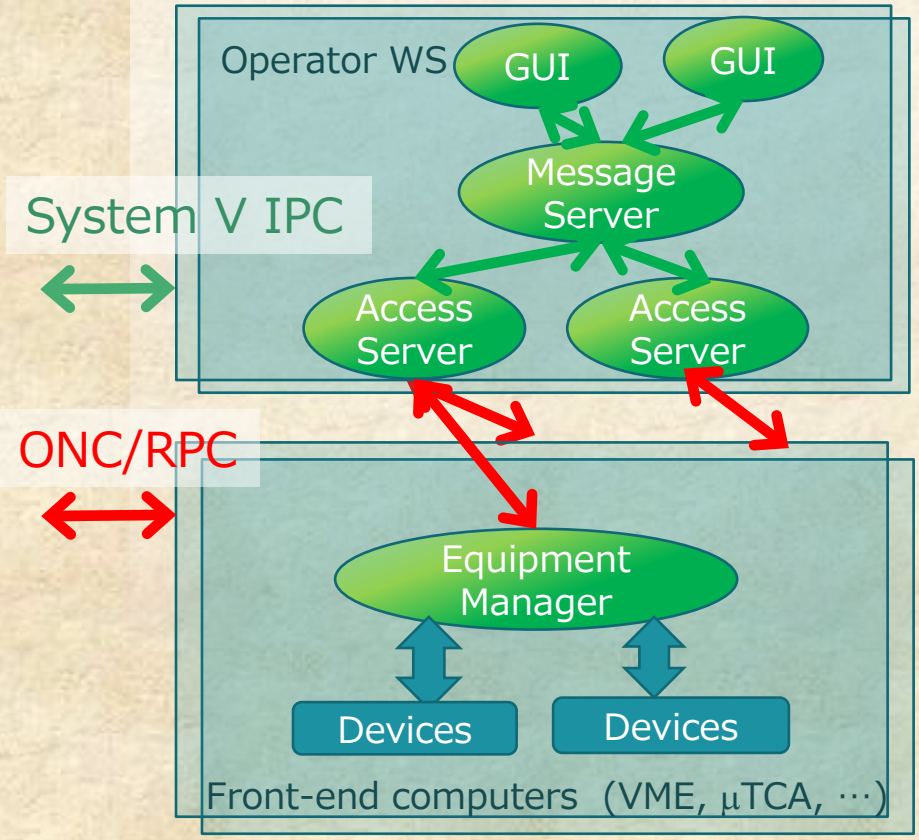
MADOCA Control Framework



Shortcomings

- SVOC length ≦ 255 characters
  - Not suitable to transmit variable length-data
    - Waveform, Image data

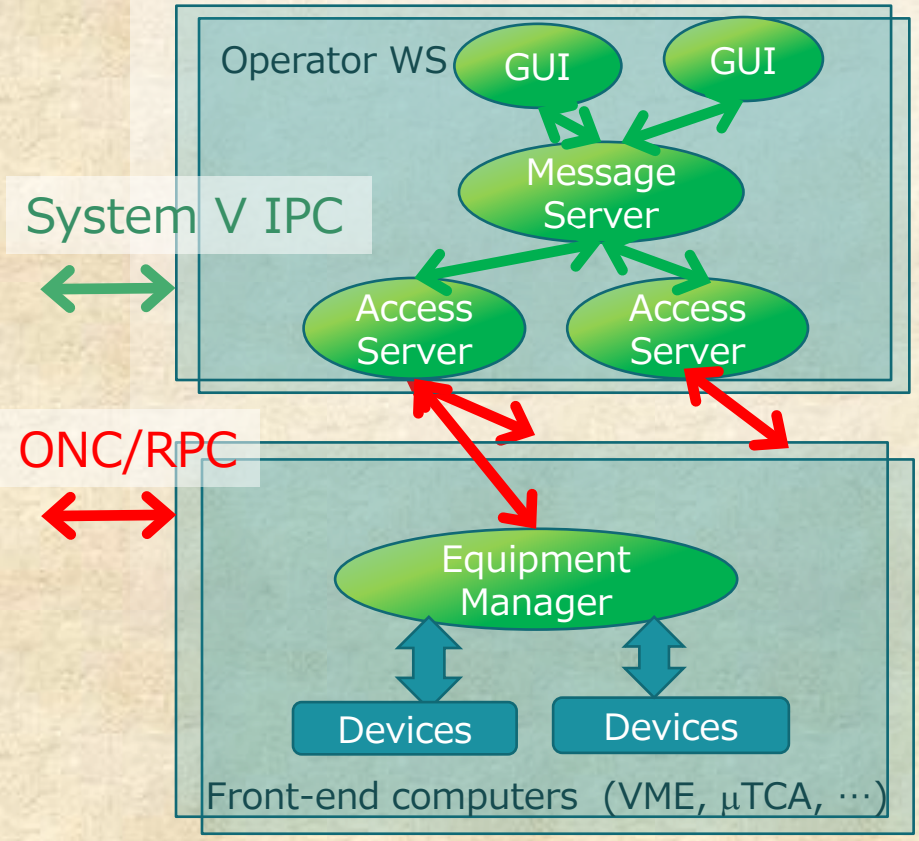
MADOCA Control Framework



Shortcomings

- SVOC length ≦ 255 characters
  - Not suitable to transmit variable length-data
    - Waveform, Image data
- Lack of controls on Windows
  - Used only for Linux, Solaris
    - Due to System V IPC, ONC/RPC

MADOCA Control Framework



Shortcomings

- SVOC length ≦ 255 characters
  - Not suitable to transmit variable length-data
    - Waveform, Image data
- Lack of controls on Windows
  - Used only for Linux, Solaris
    - Due to System V IPC, ONC/RPC
- Synchronous communication in ONC/RPC
  - Need to wait to finish processing of each message

Next-Generation MADOCA, MADOCA II

- Keep messaging format of MADOCA
- Fix shortcomings in MADOCA

Next-Generation MADOCA, MADOCA II

- Keep messaging format of MADOCA
- Fix shortcomings in MADOCA

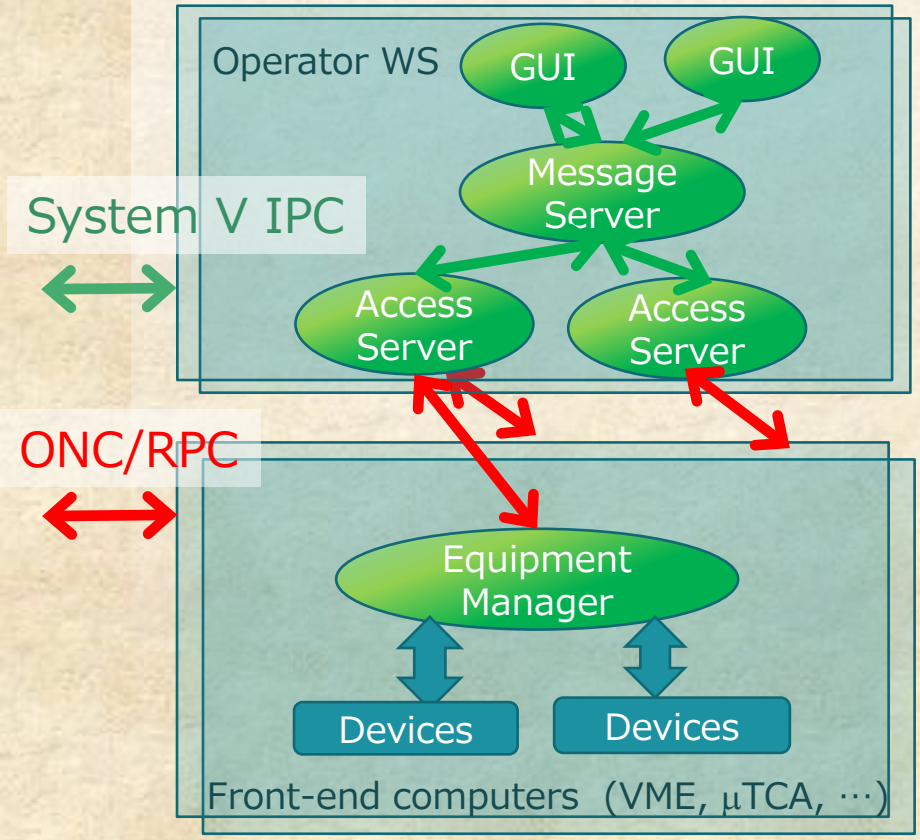
✓ Messaging with ZeroMQ/MessagePack

✓ Data logging with NoSQL (Cassandara, Redis)

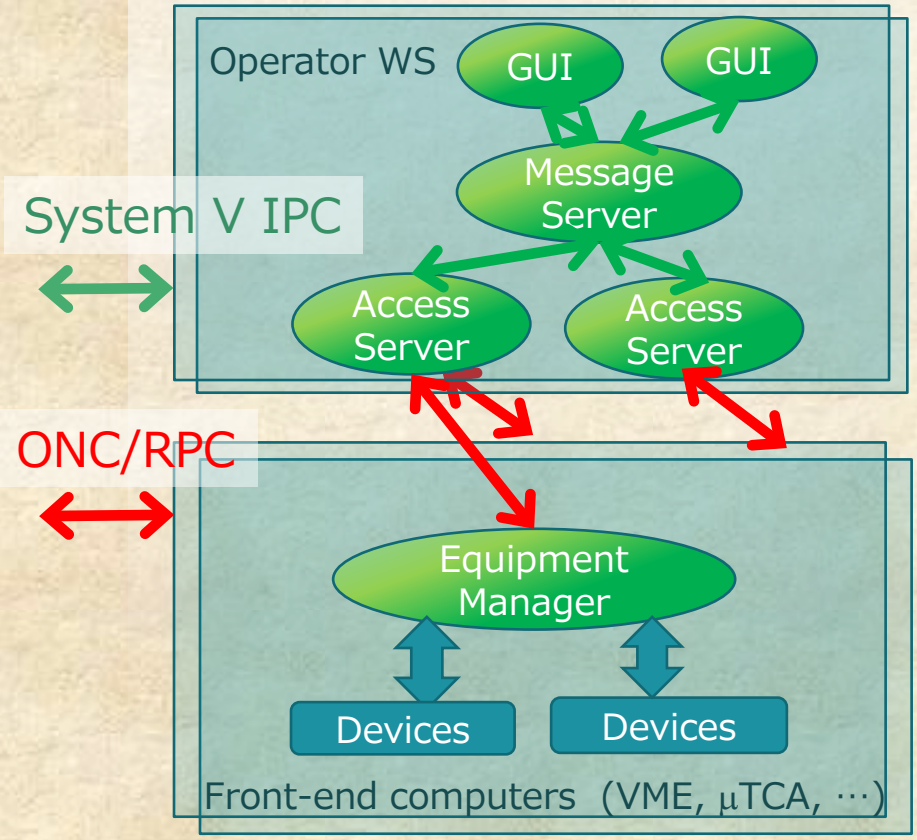
→ Refer M.Kago et al. on TUPPC08

– "Development of a Scalable and Flexible Data Logging System Using NoSQL Databases"

MADOCA

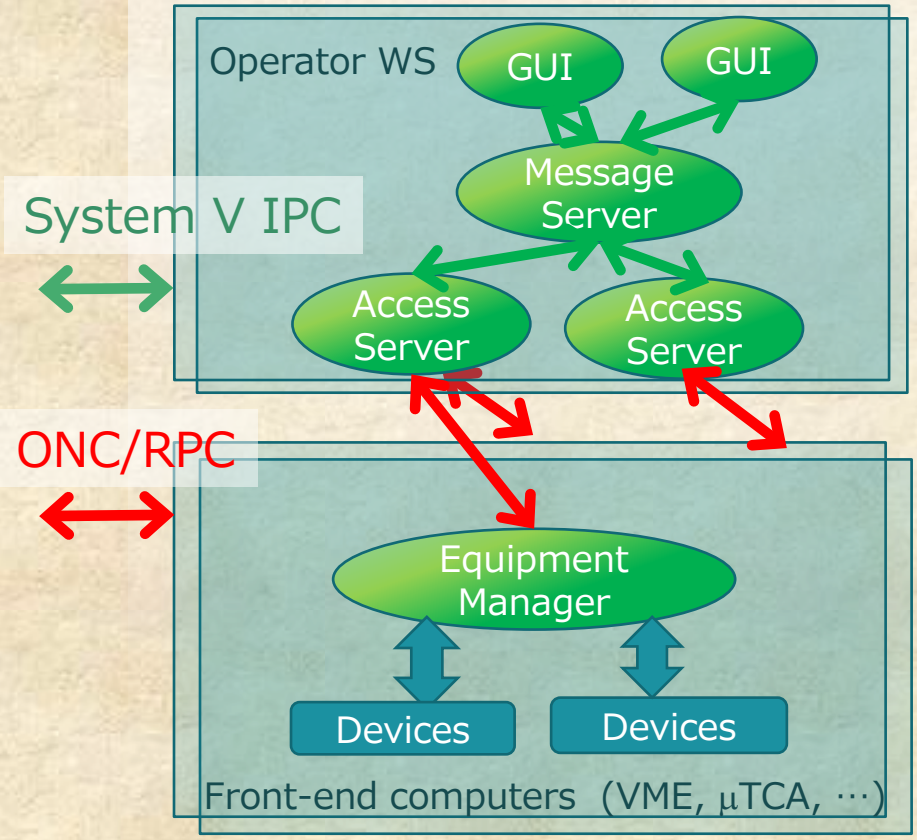

Operator WS
GUI
GUI
Message Server
Access Server
Access Server
System V IPC
ONC/RPC
Equipment Manager
Devices
Devices
Front-end computers (VME, μTCA, ⋯)

MADOCA



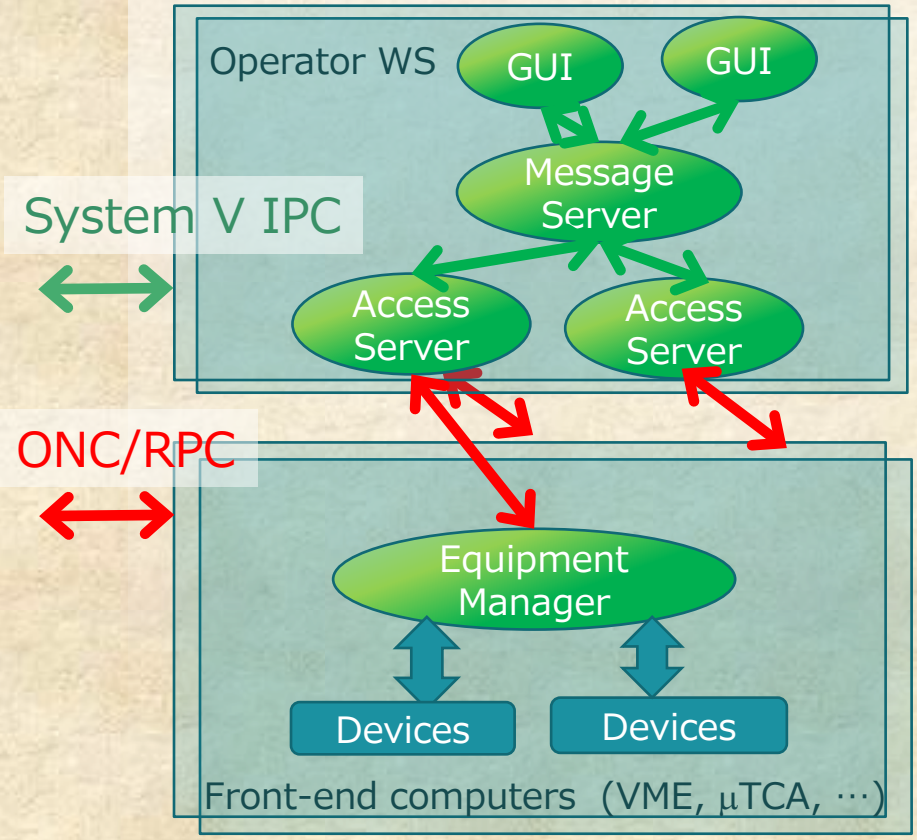
MADOCA II

MADOCA



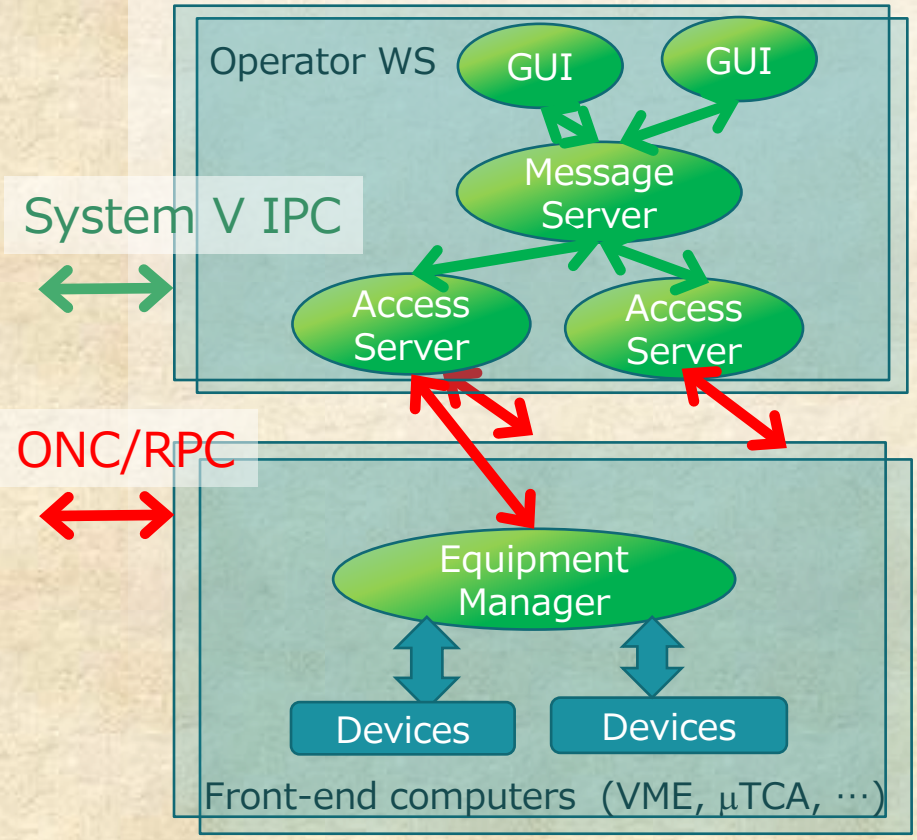
MADOCA II

MADOCA



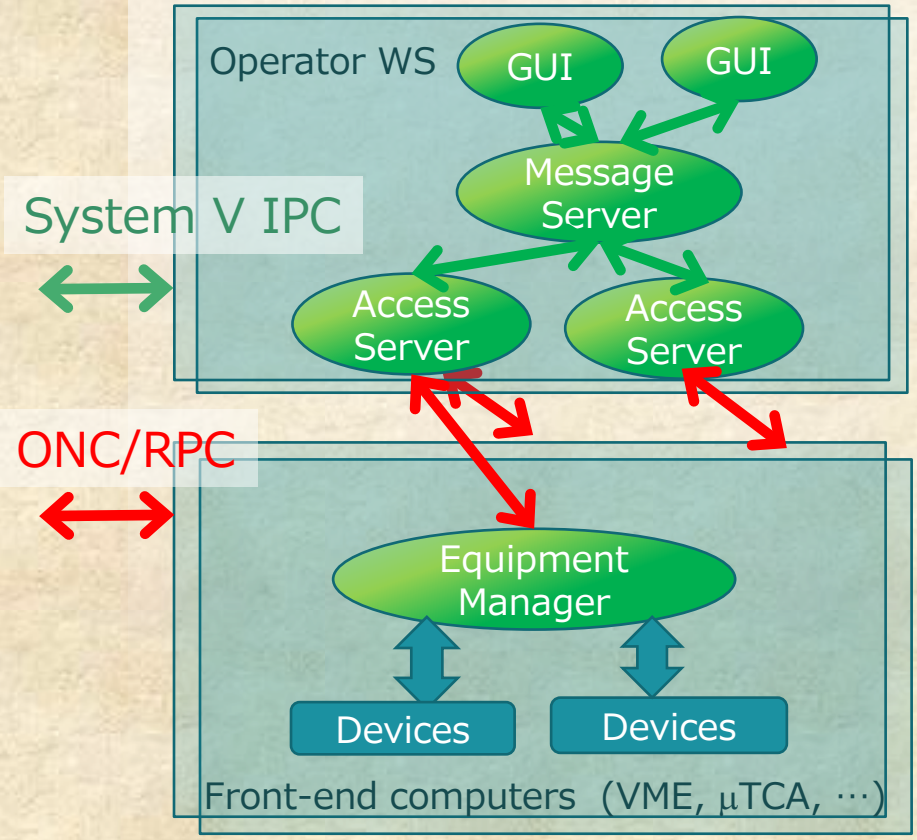
MADOCA II

MADOCA

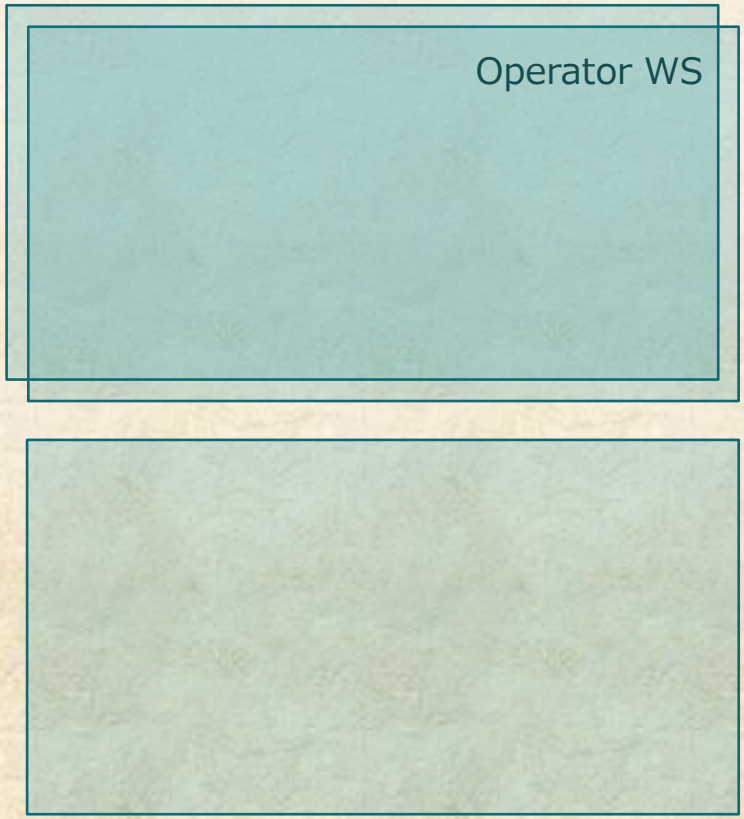


MADOCA II

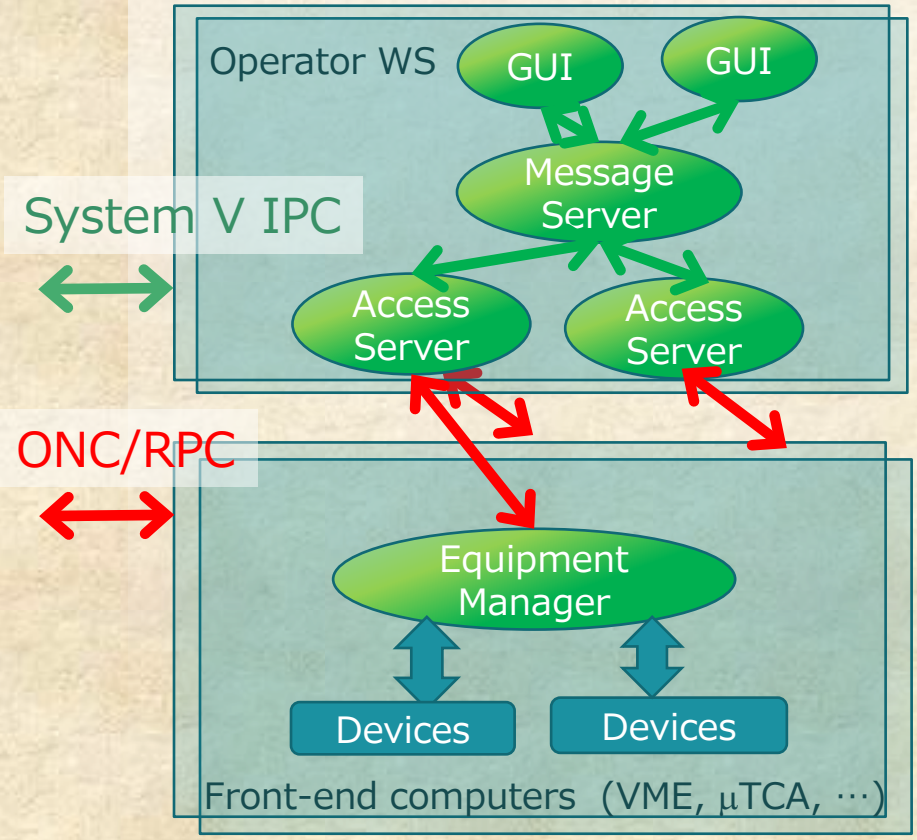
MADOCA



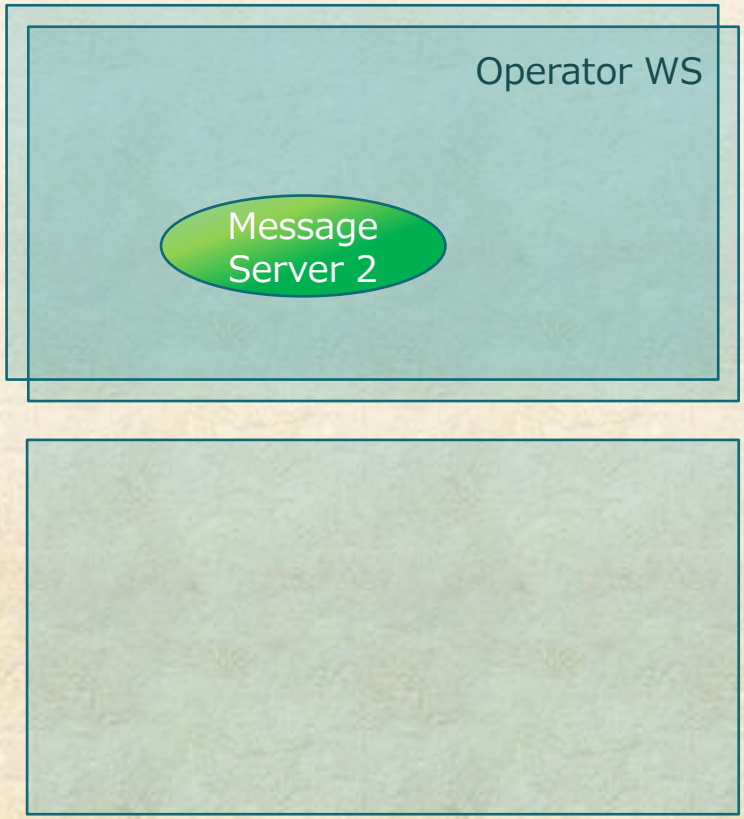
MADOCA II



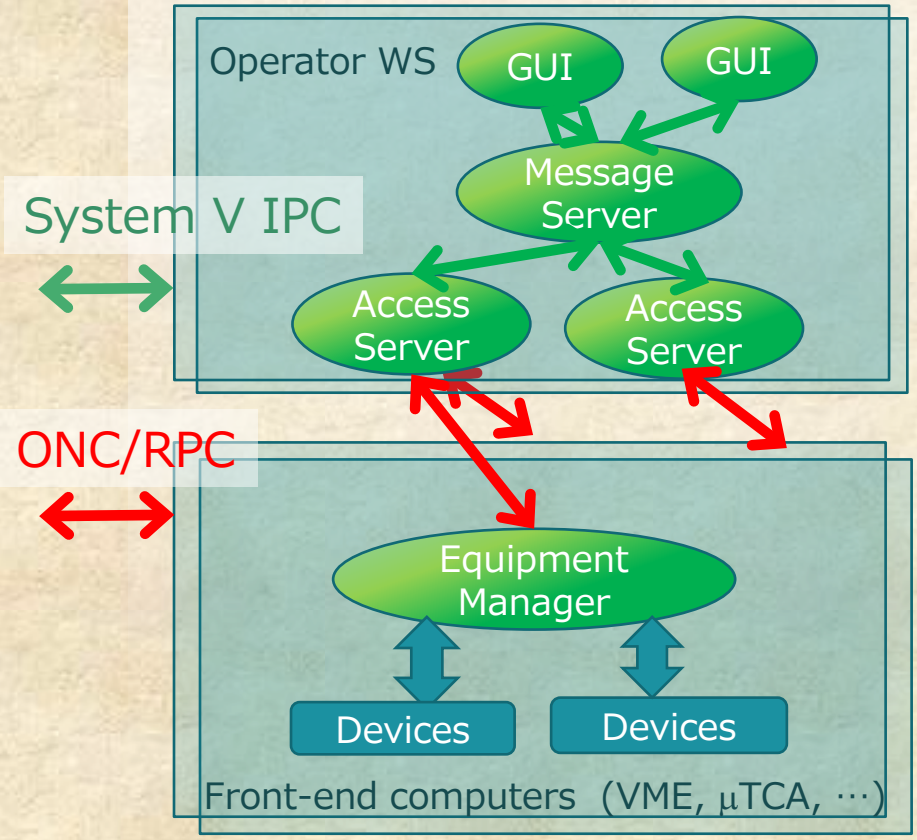
MADOCA



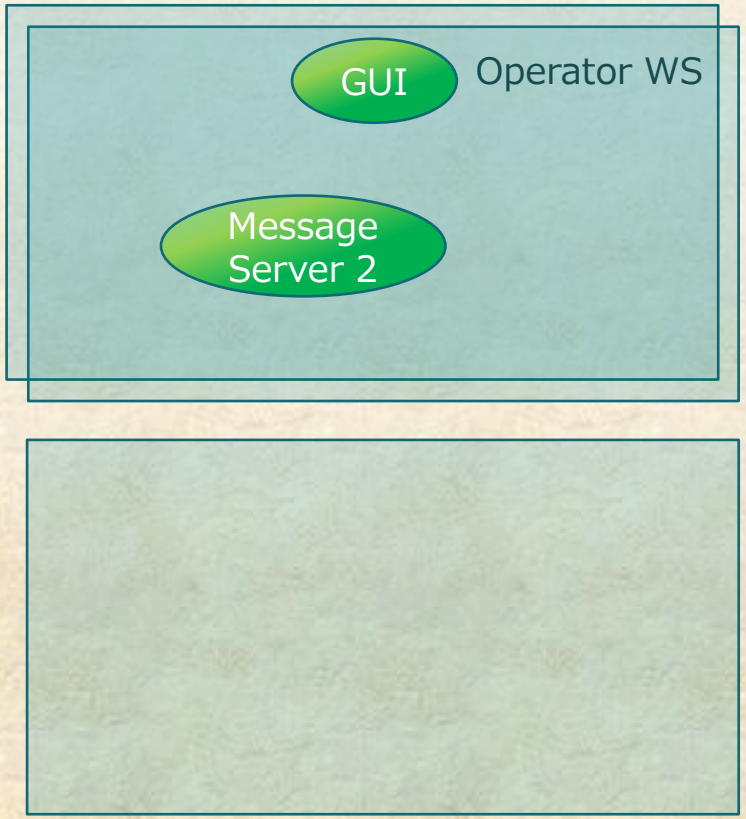
MADOCA II



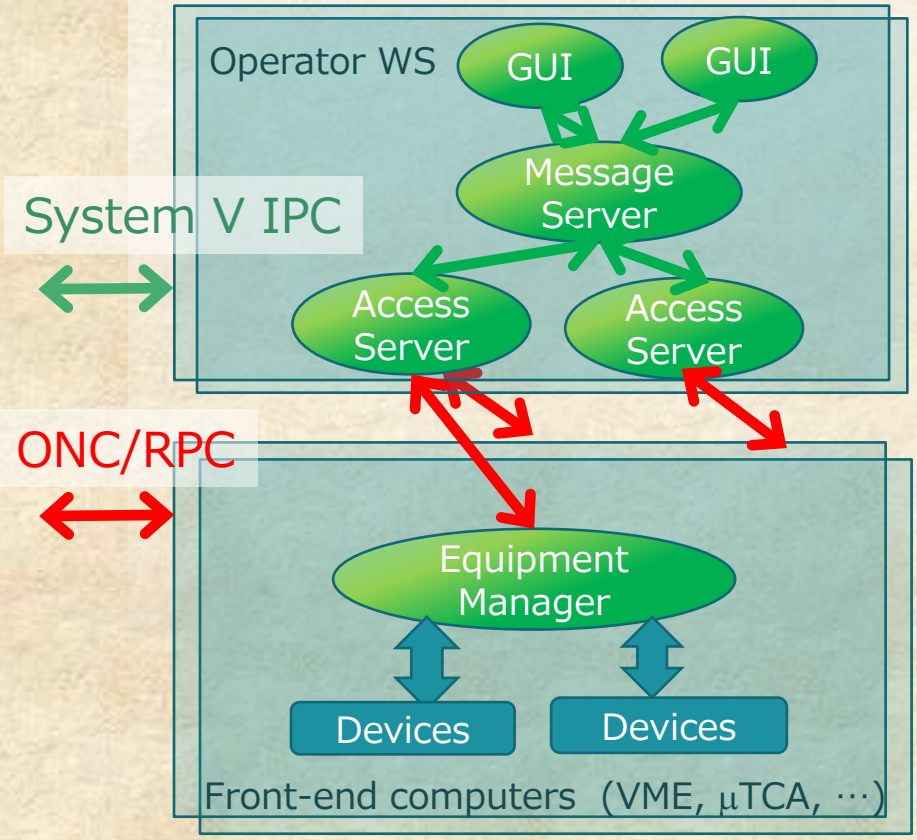
MADOCA



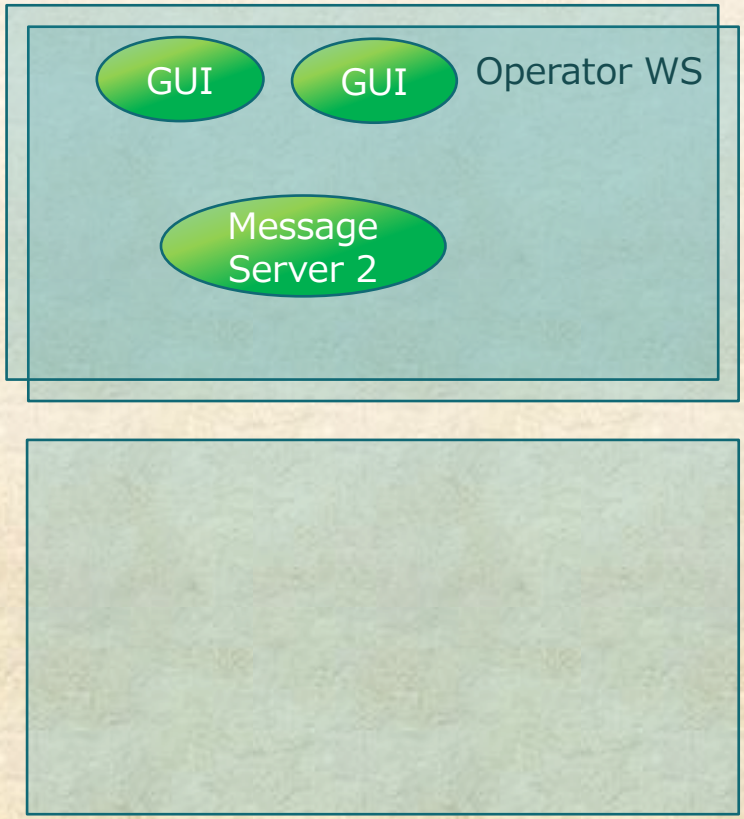
MADOCA II



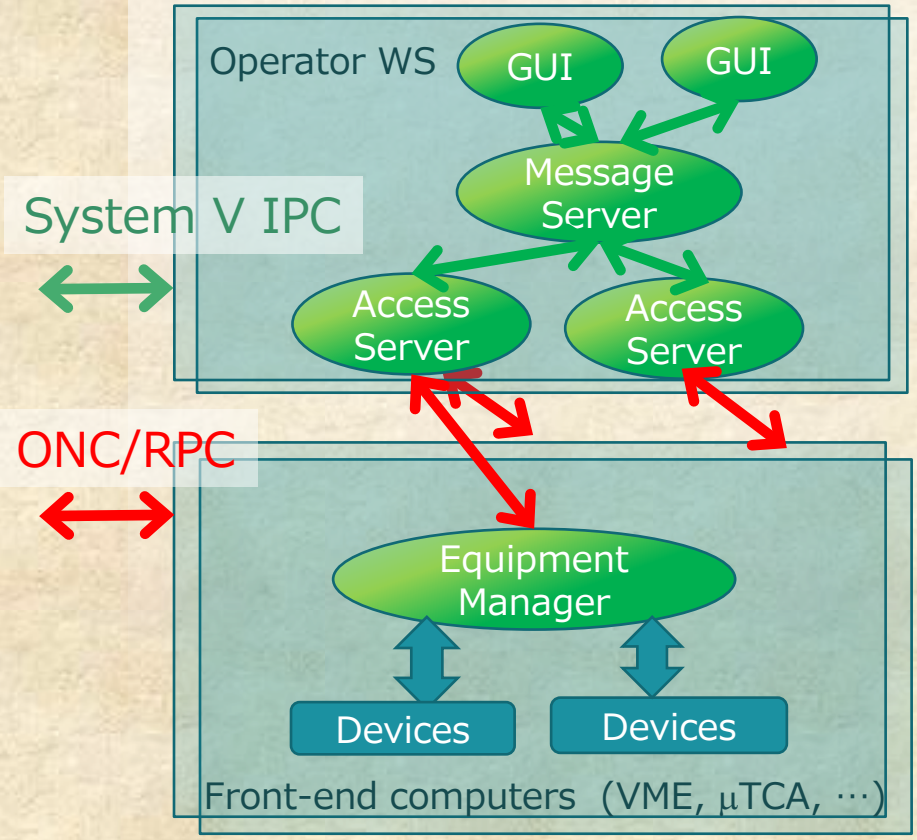
MADOCA



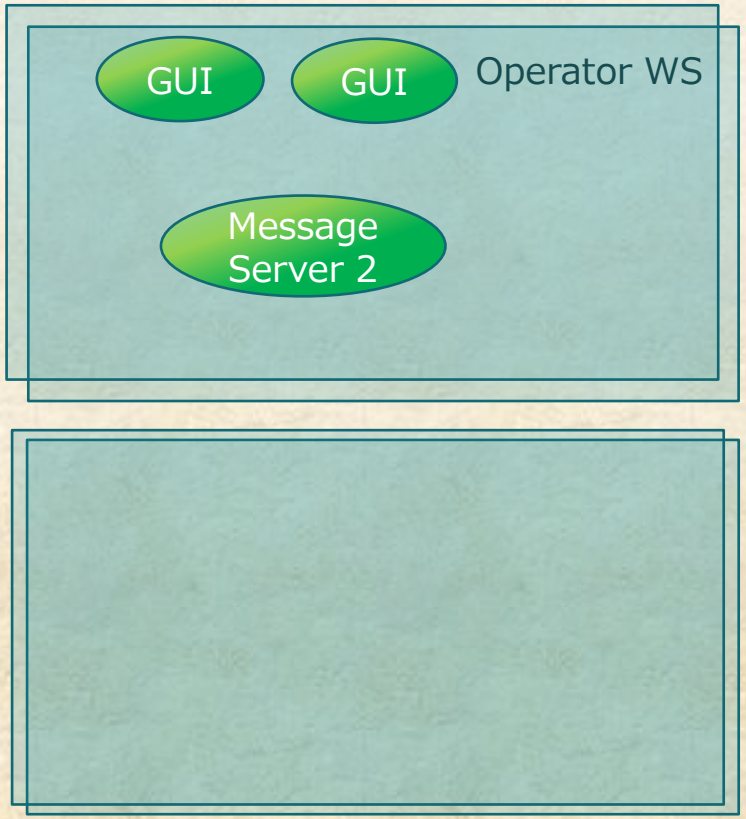
MADOCA II



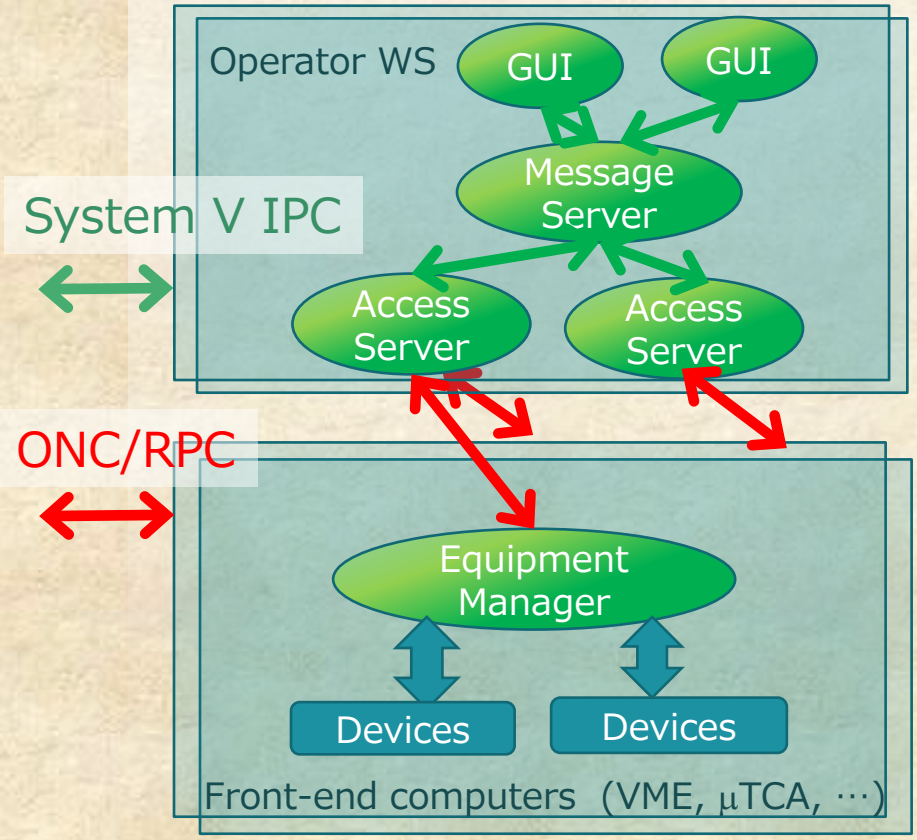
MADOCA



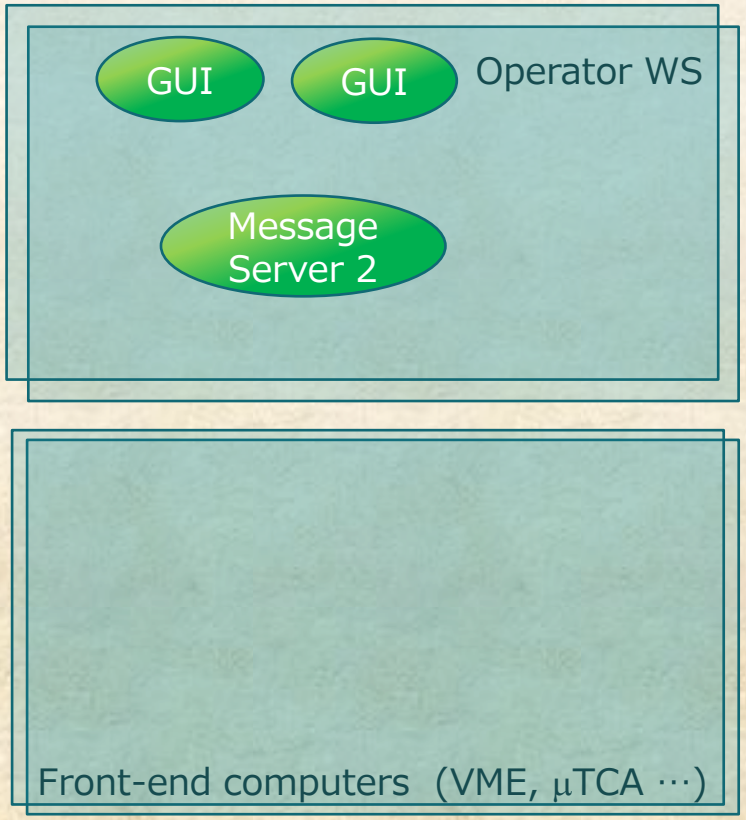
MADOCA II



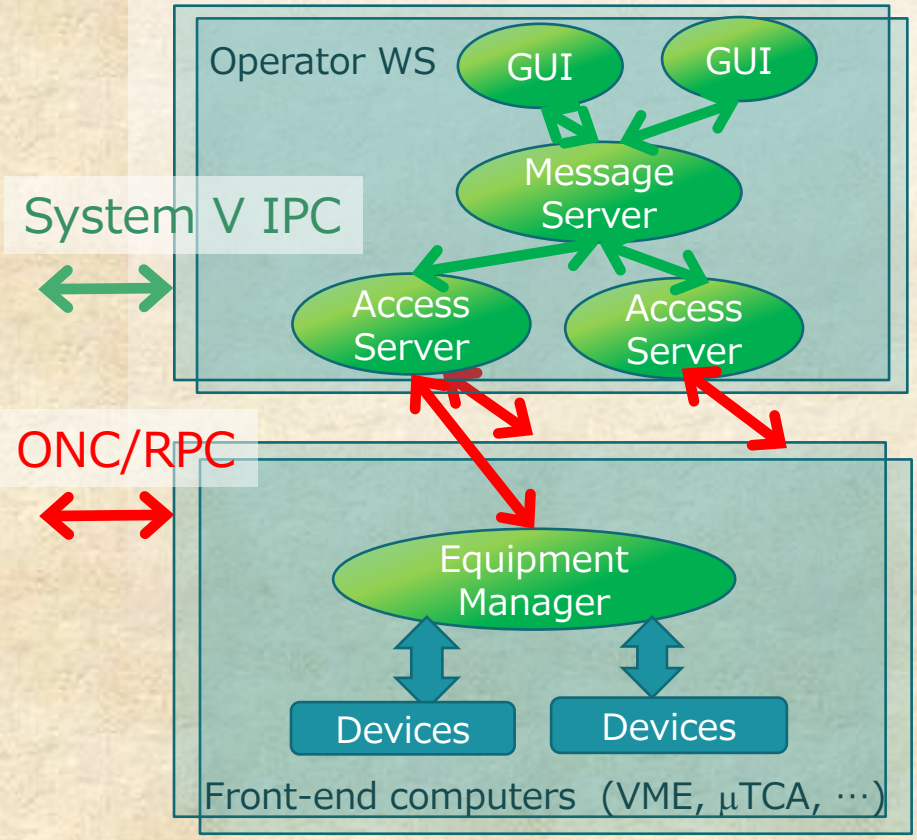
MADOCA



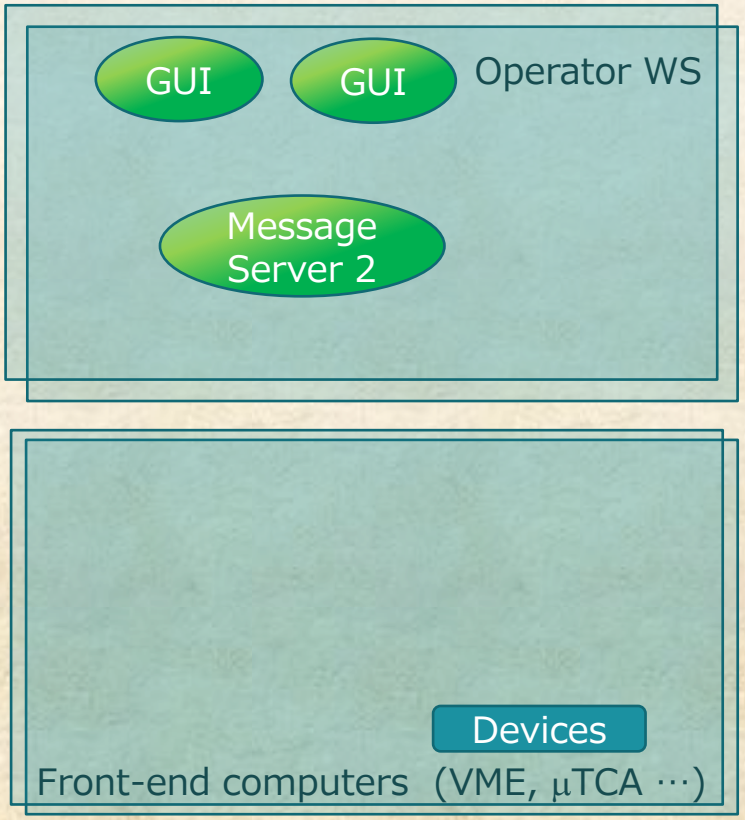
MADOCA II



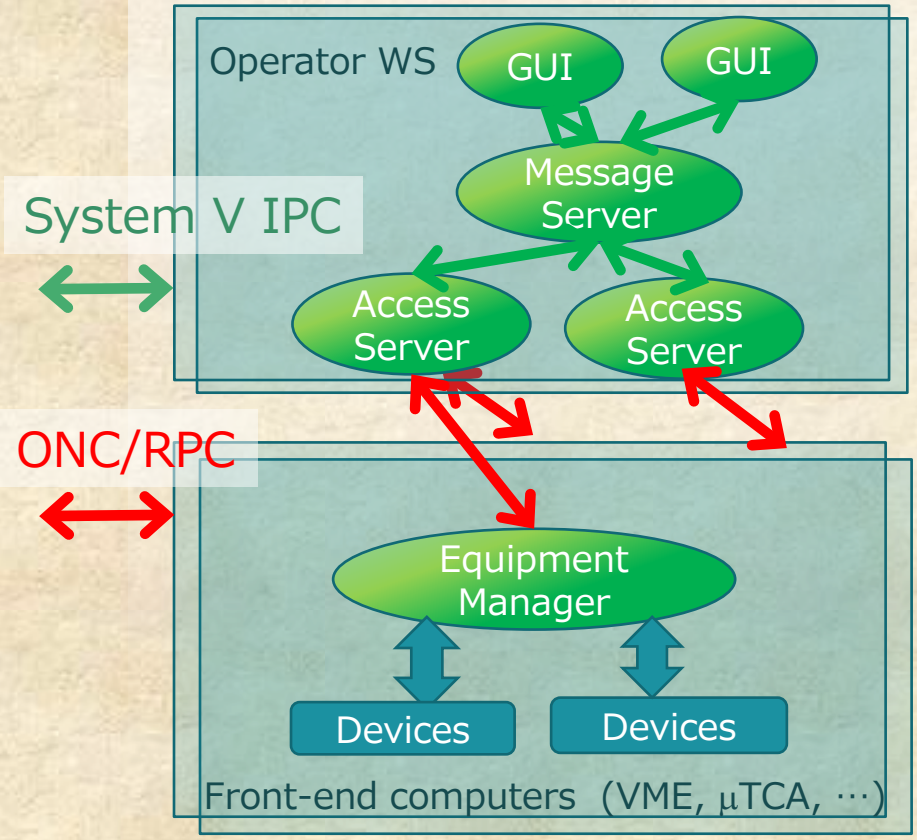
MADOCA



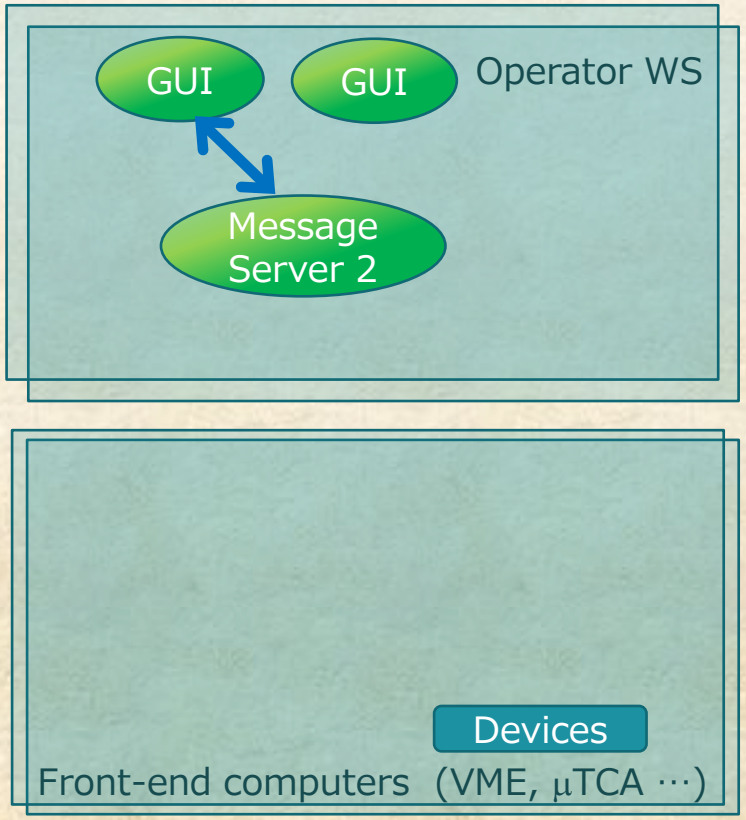
MADOCA II



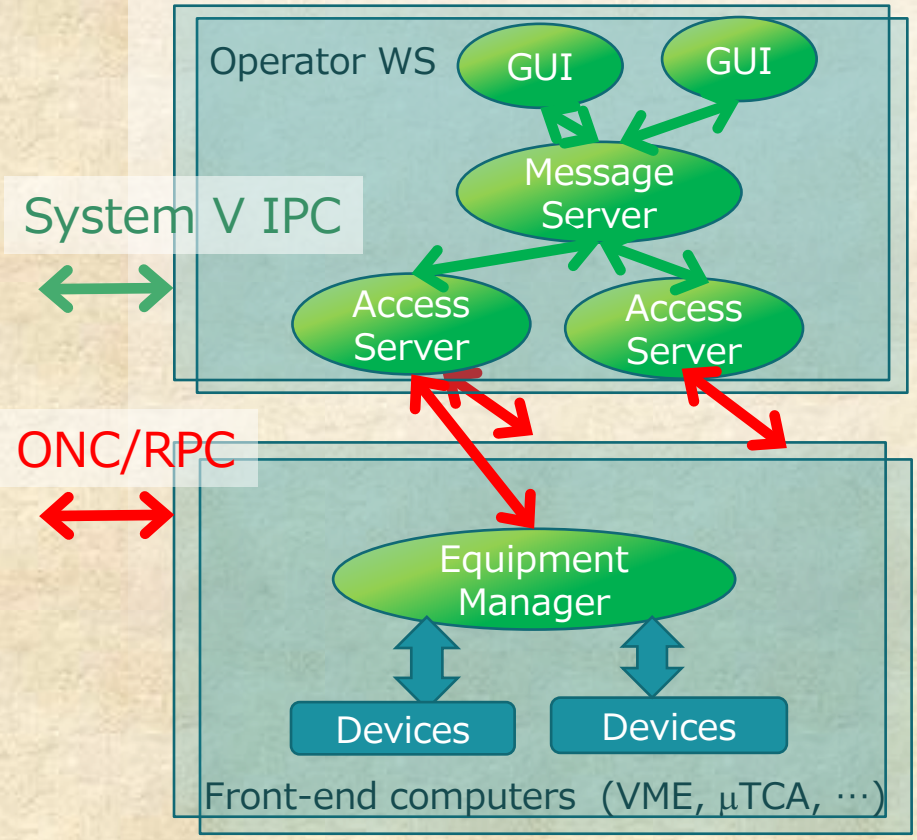
MADOCA



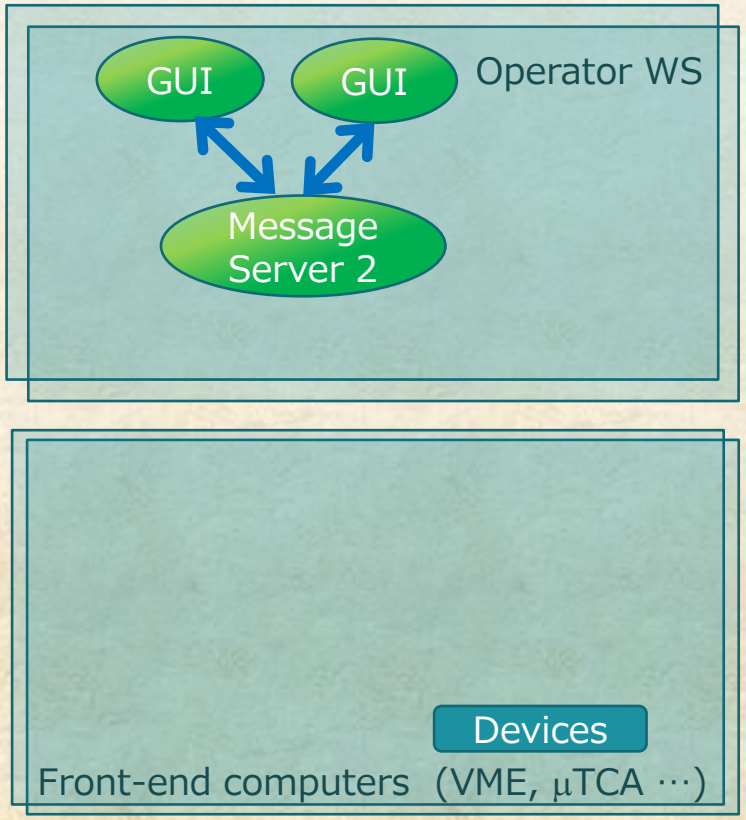
MADOCA II



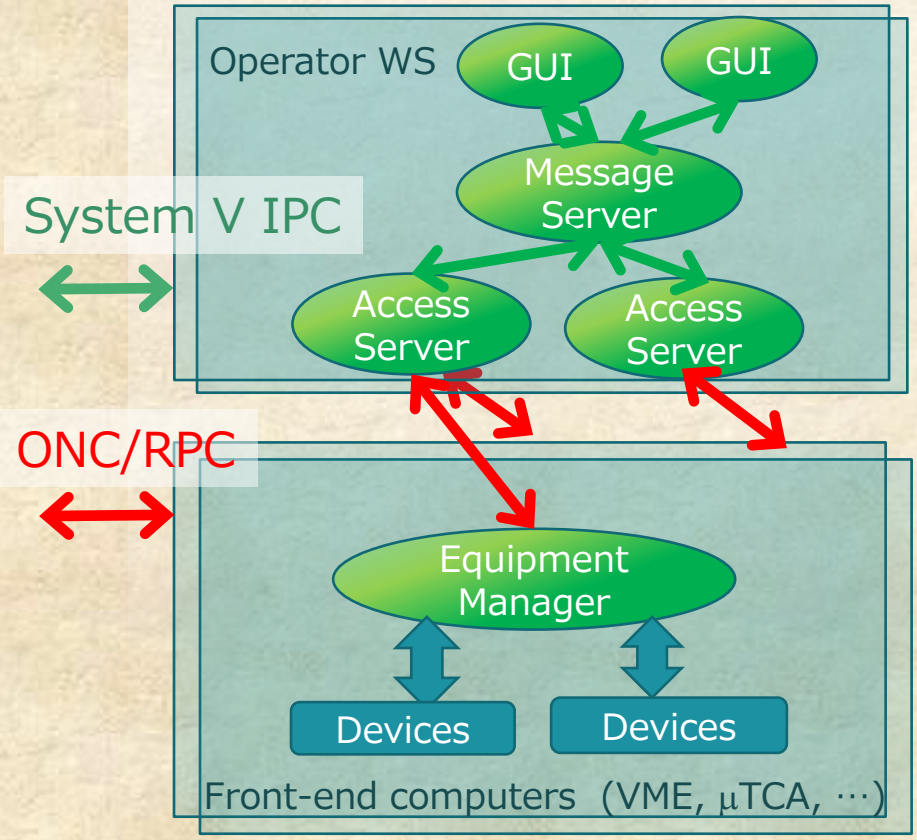
MADOCA



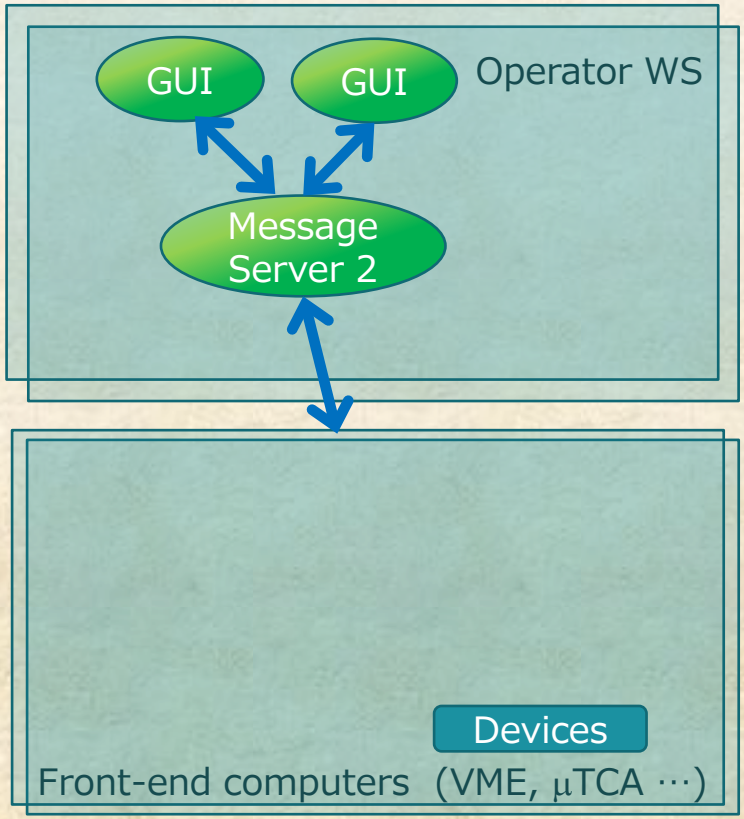
MADOCA II



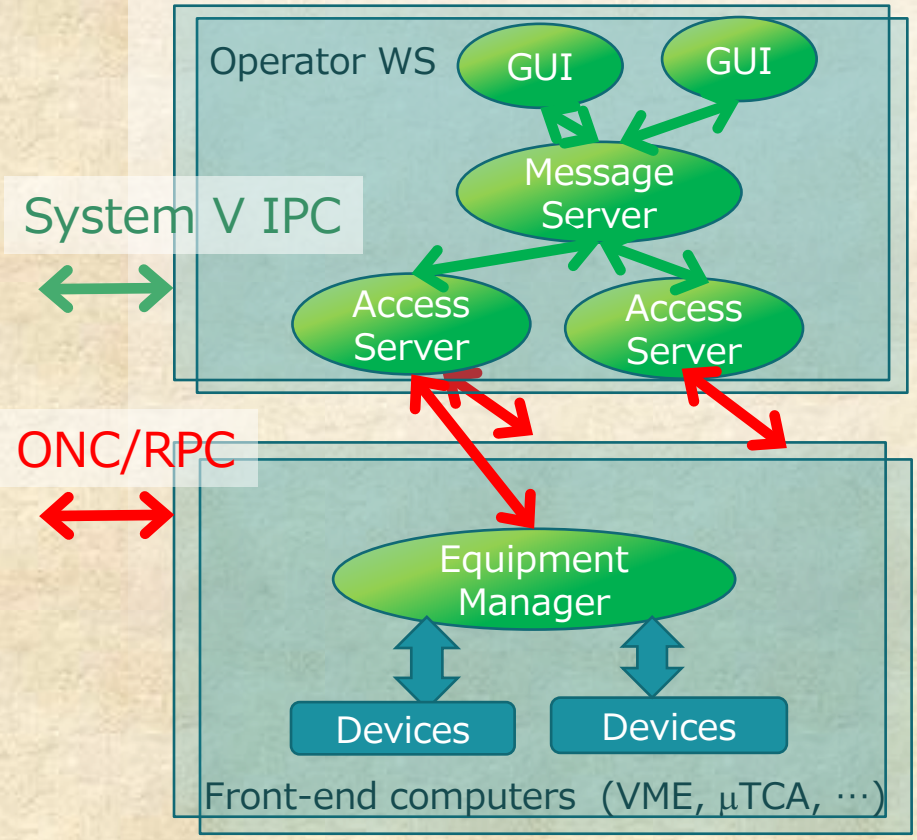
MADOCA



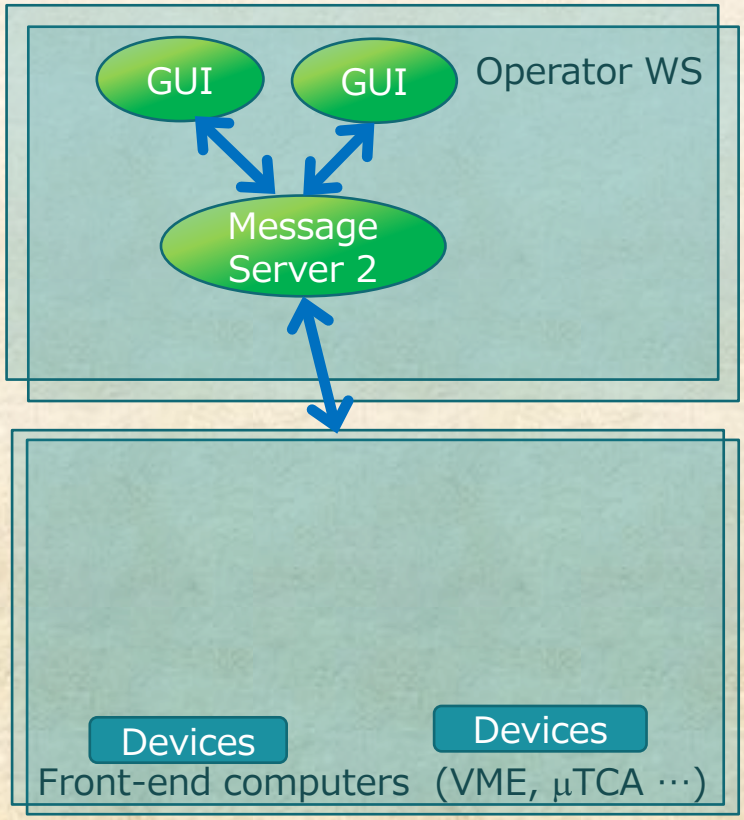
MADOCA II



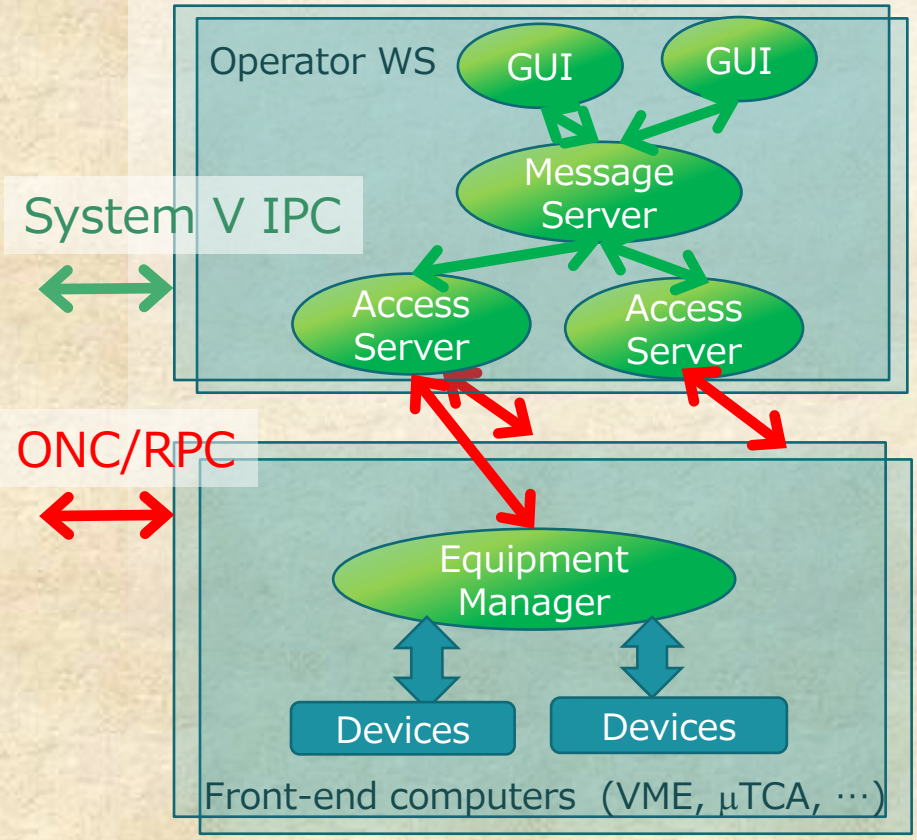
MADOCA



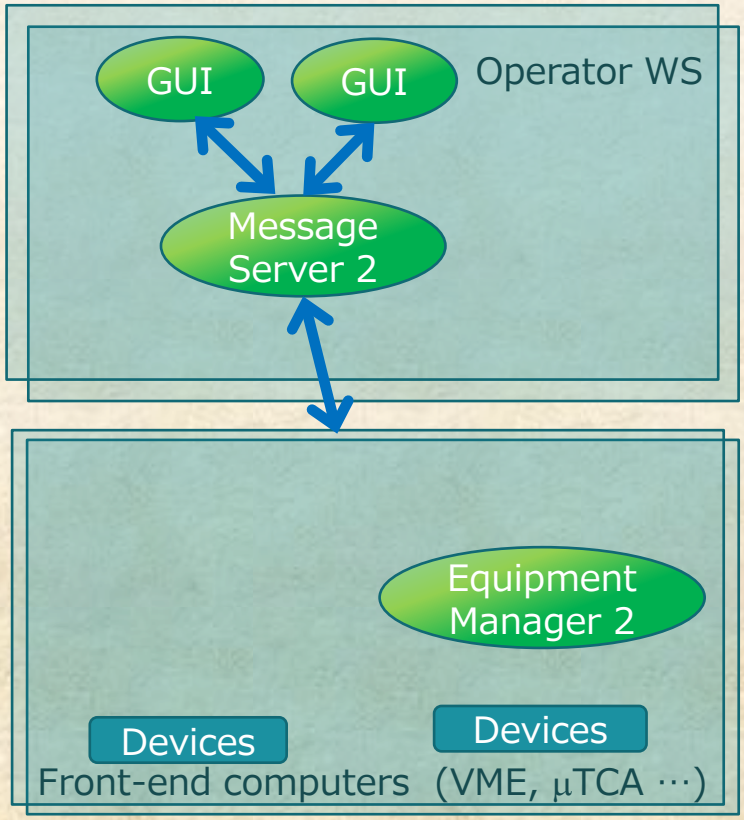
MADOCA II



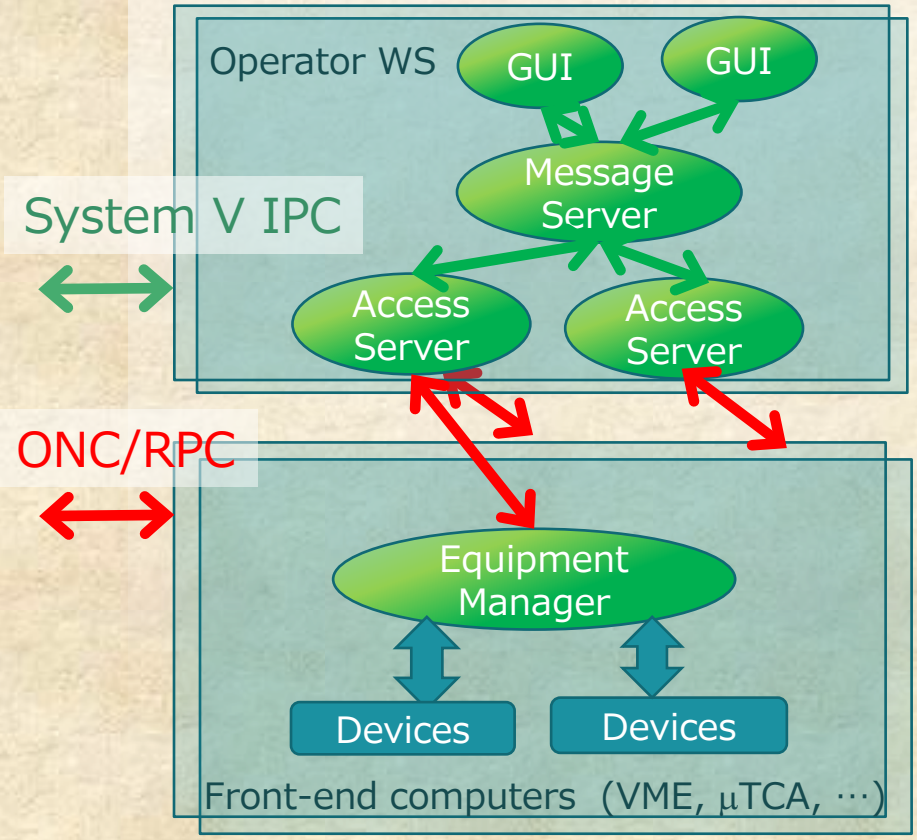
MADOCA



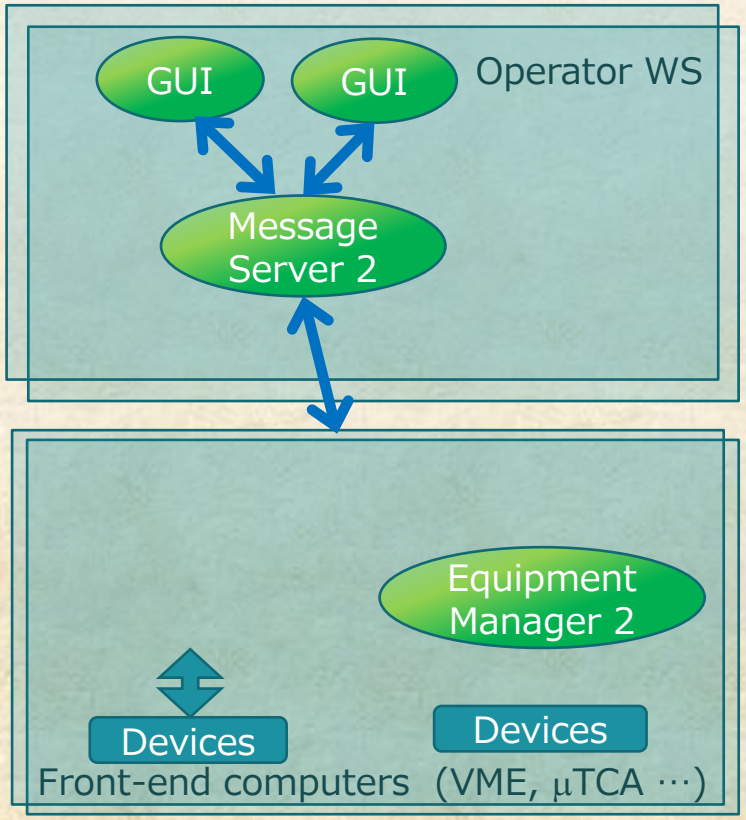
MADOCA II



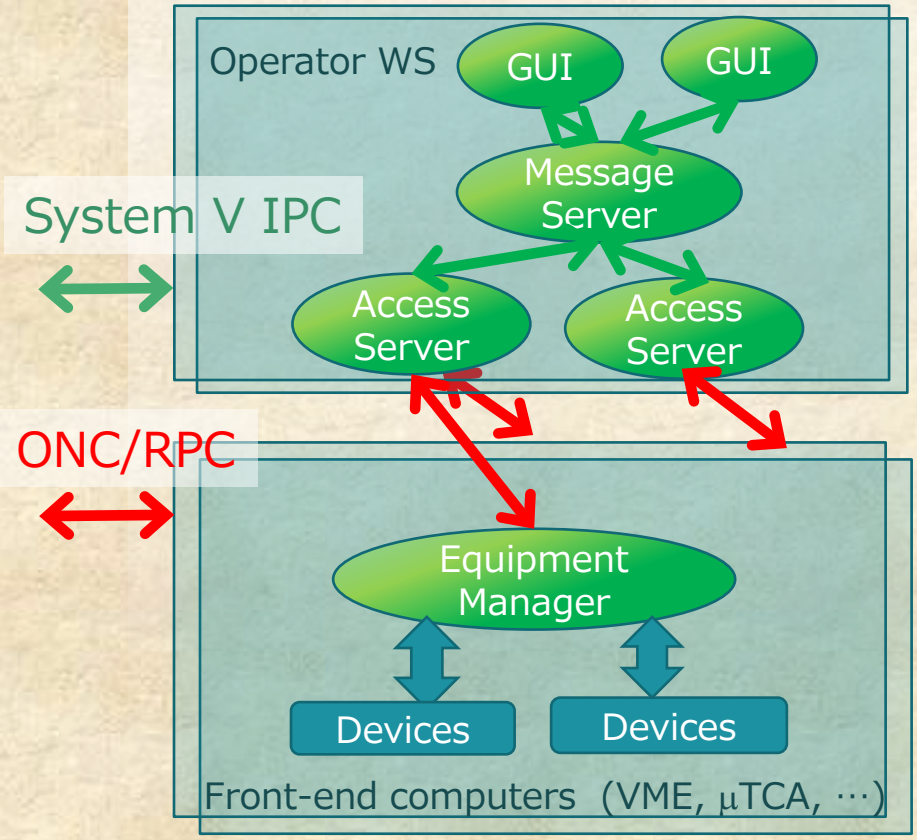
MADOCA



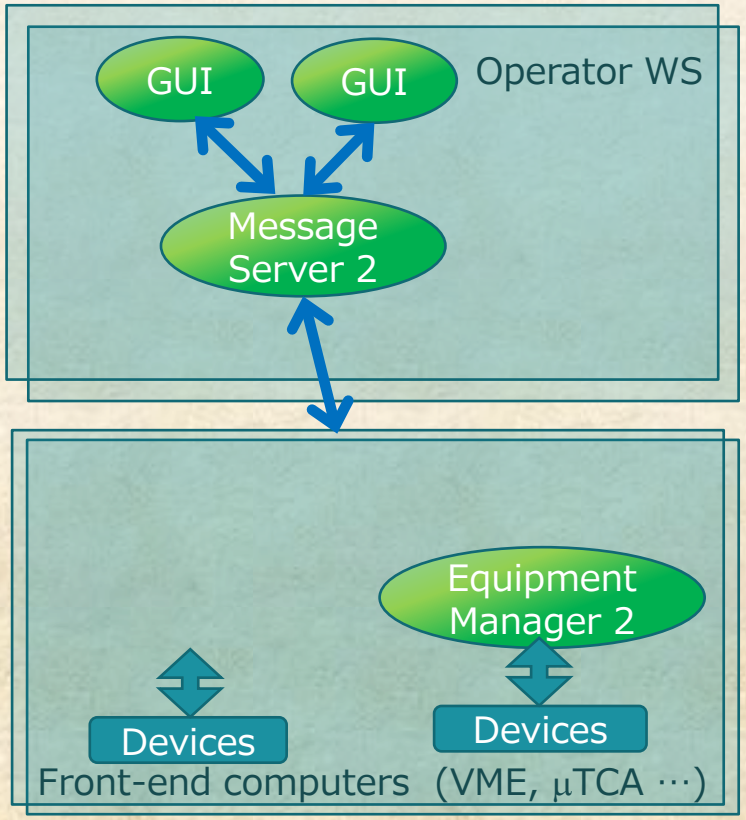
MADOCA II



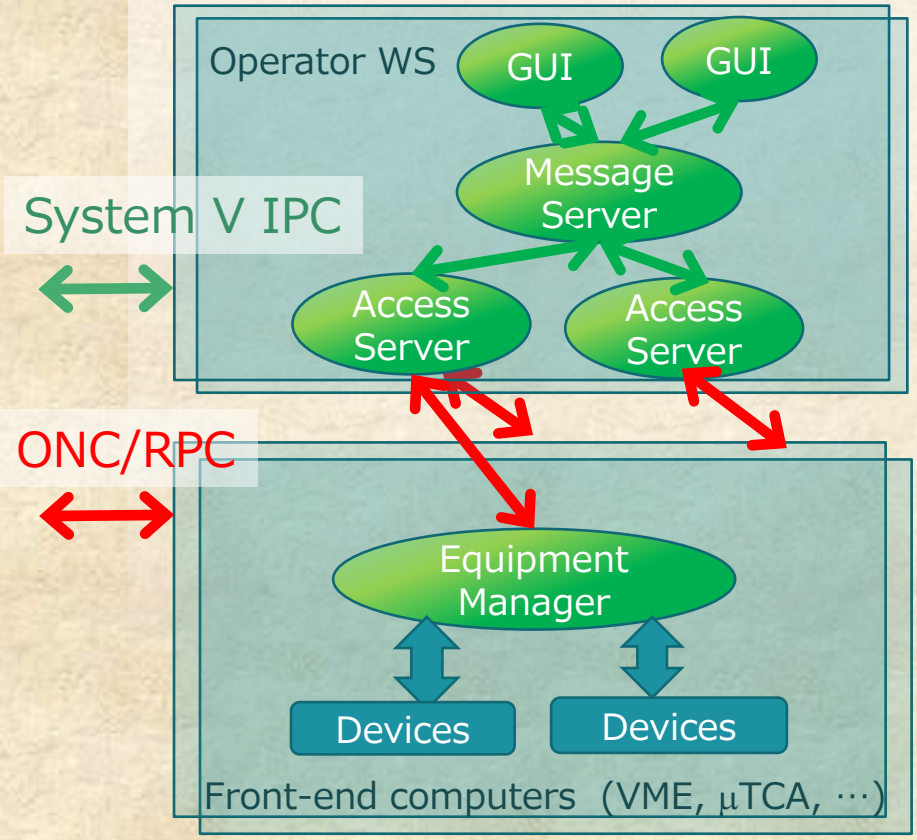
MADOCA



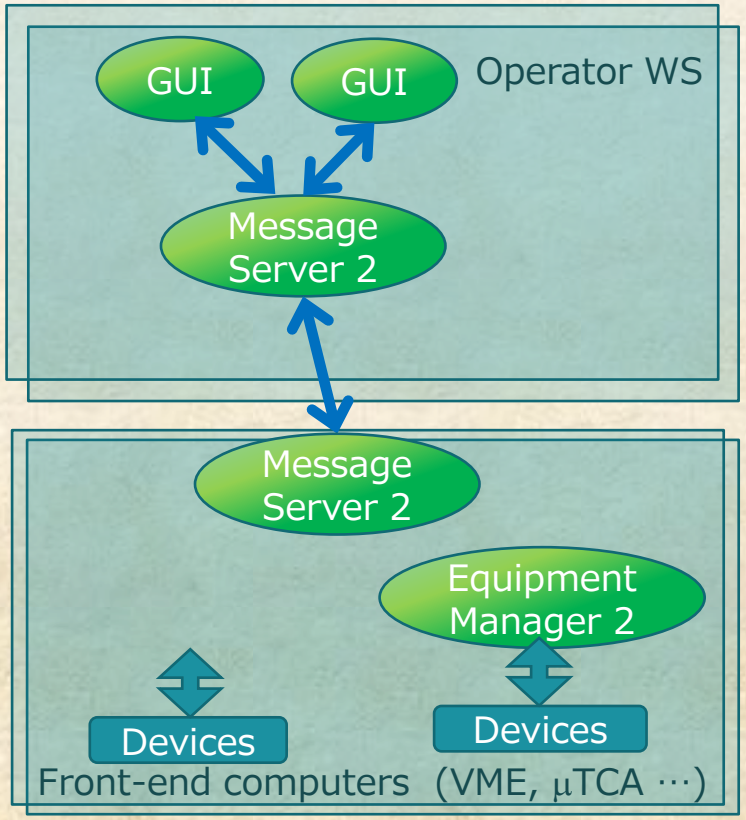
MADOCA II



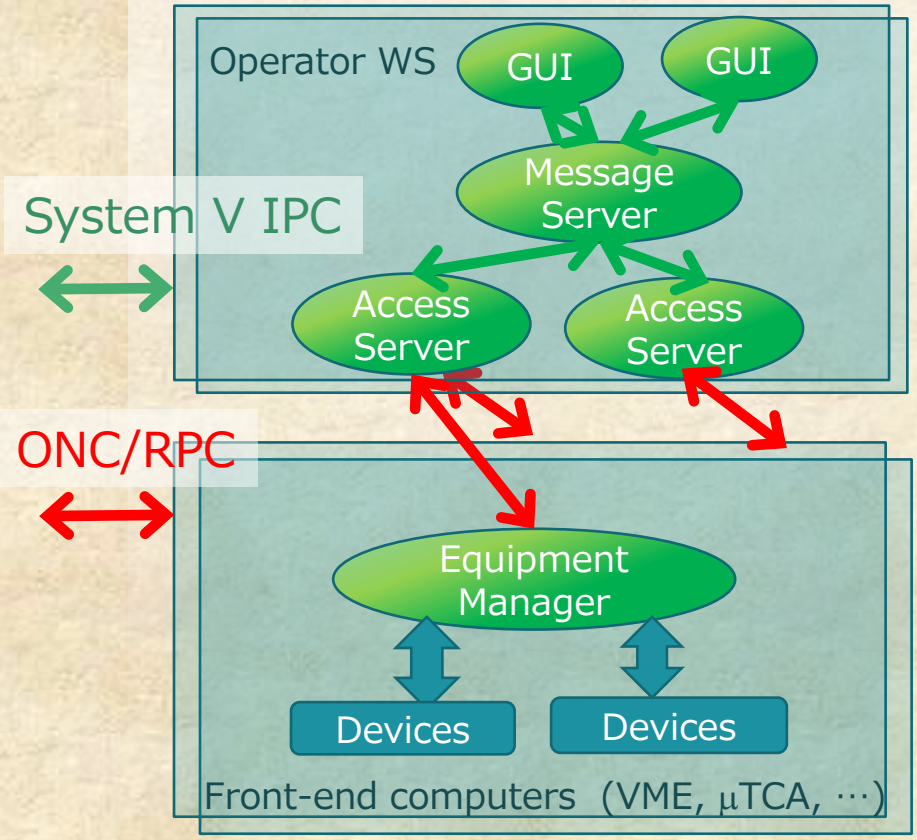
MADOCA



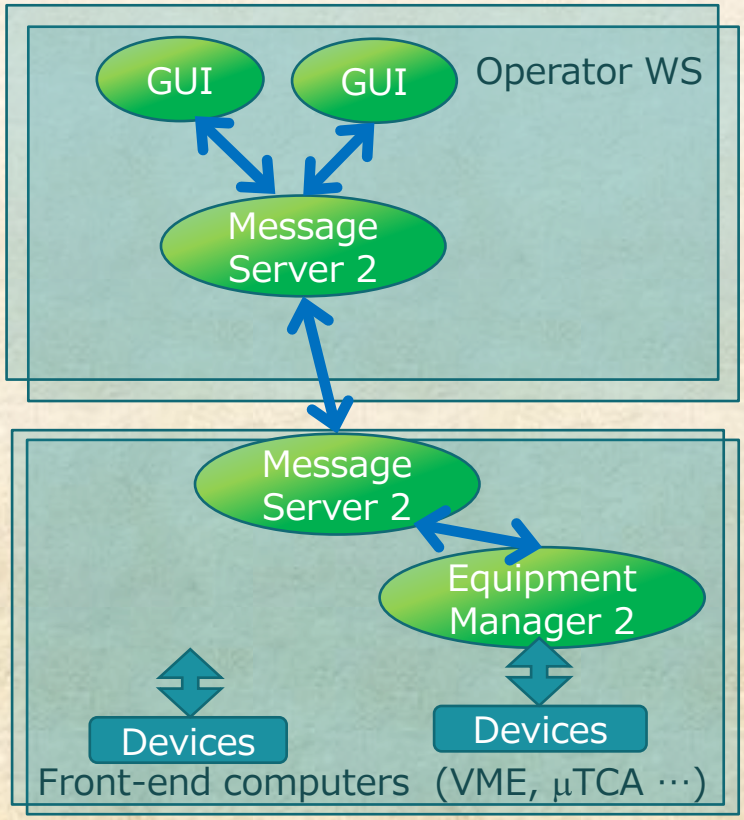
MADOCA II



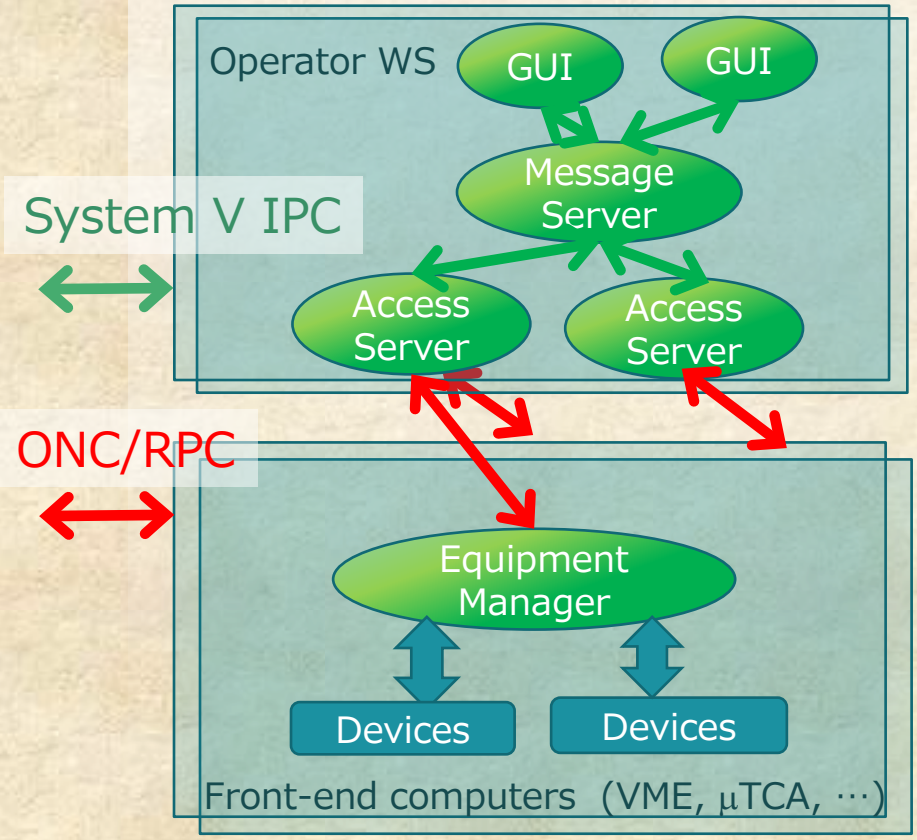
MADOCA



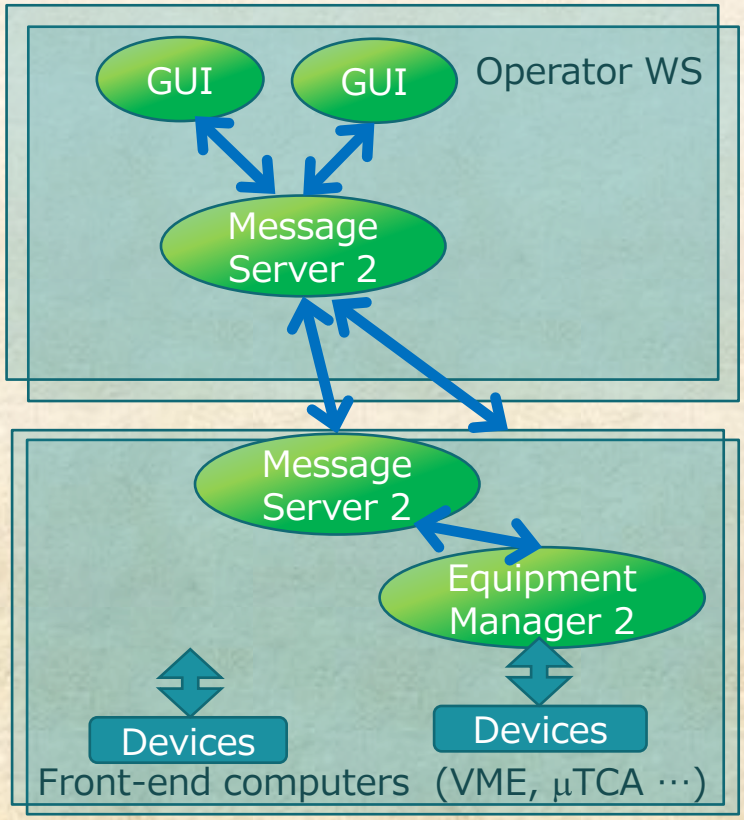
MADOCA II



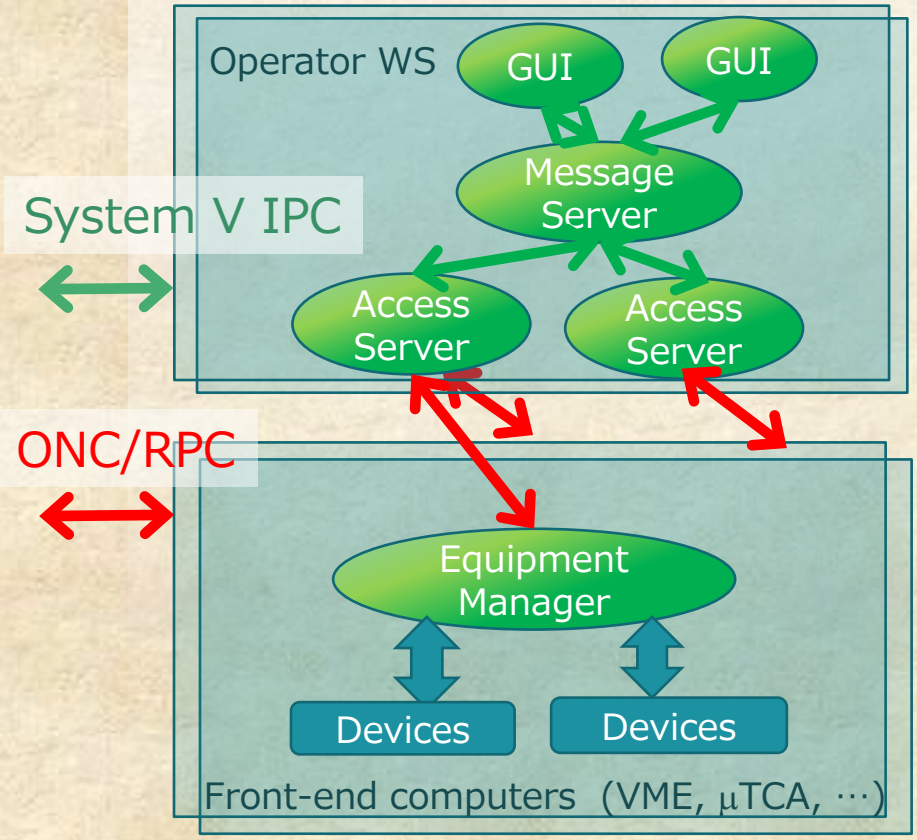
MADOCA



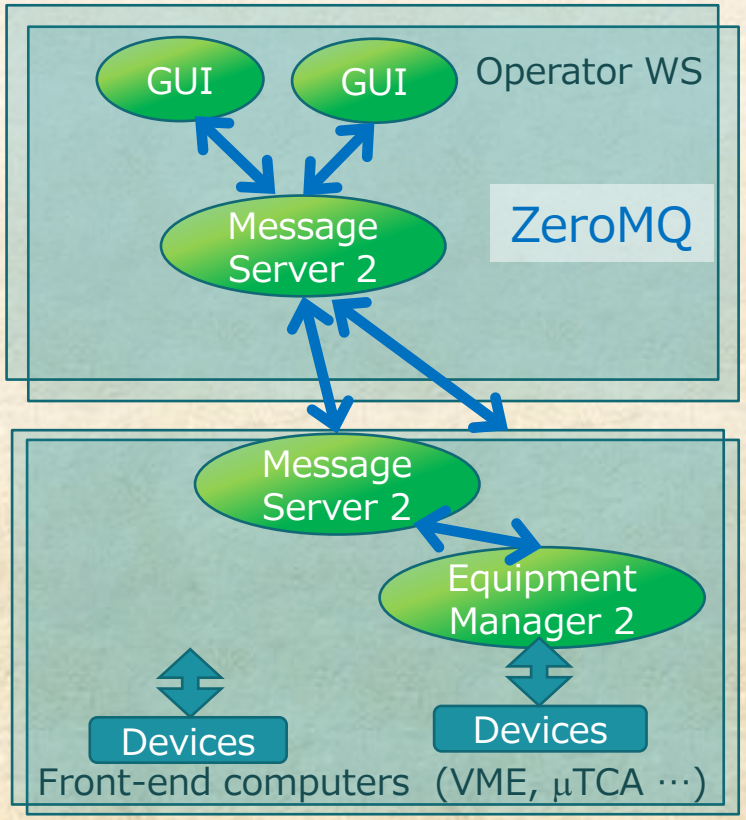
MADOCA II



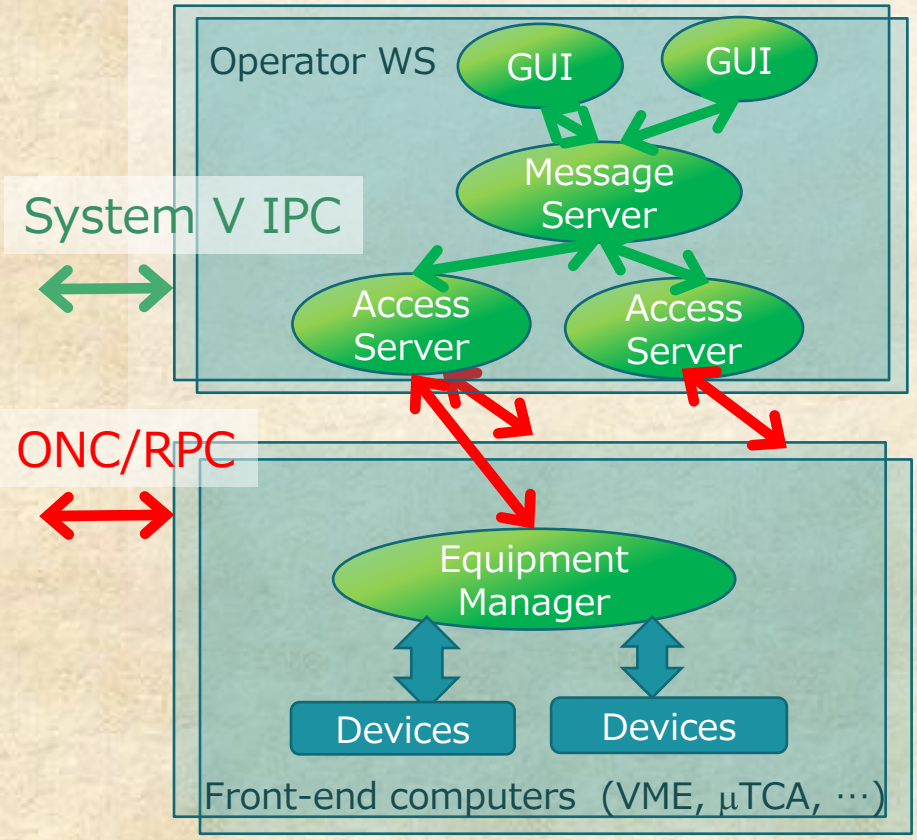
MADOCA



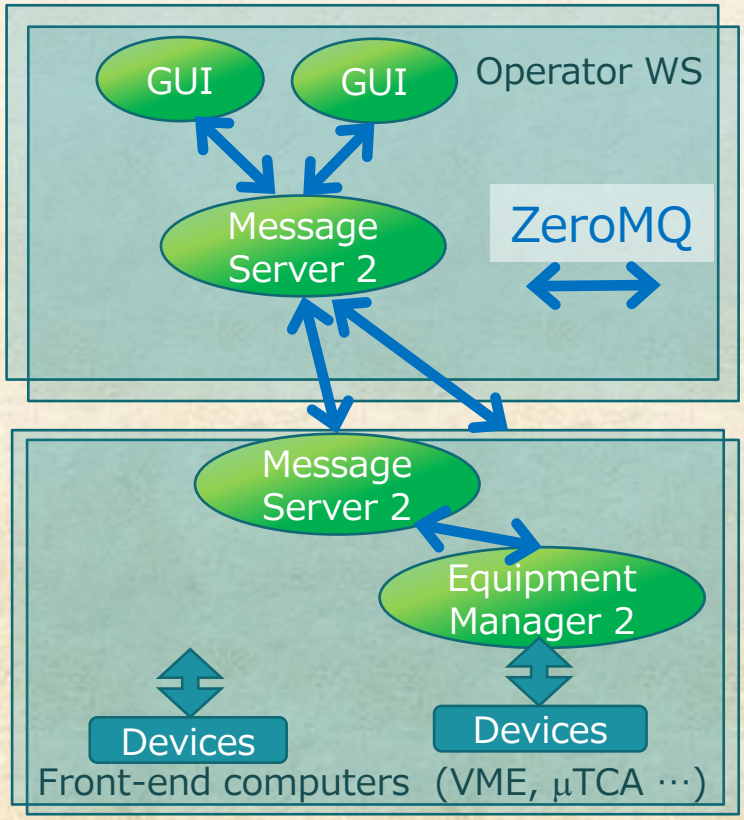
MADOCA II

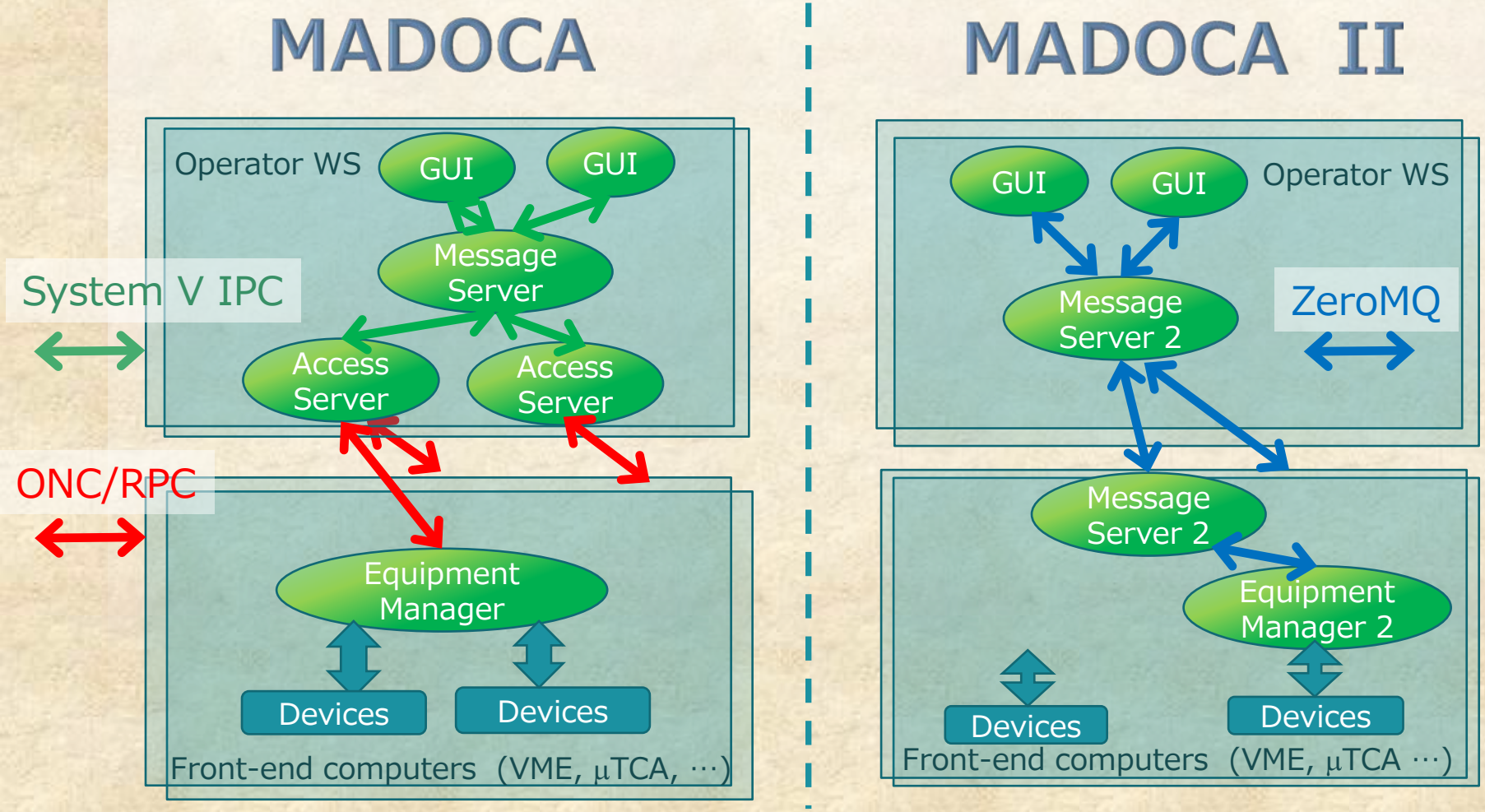


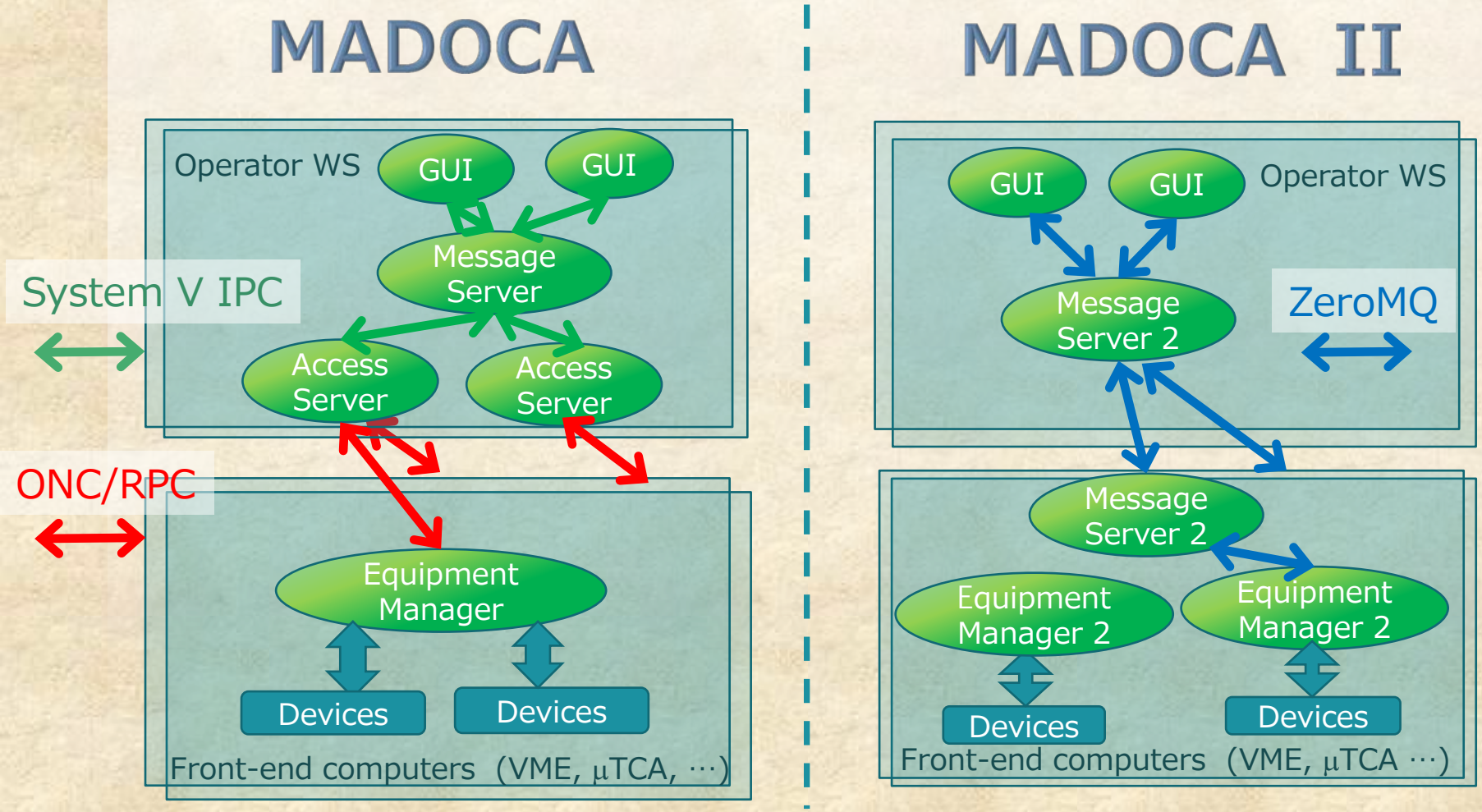
MADOCA

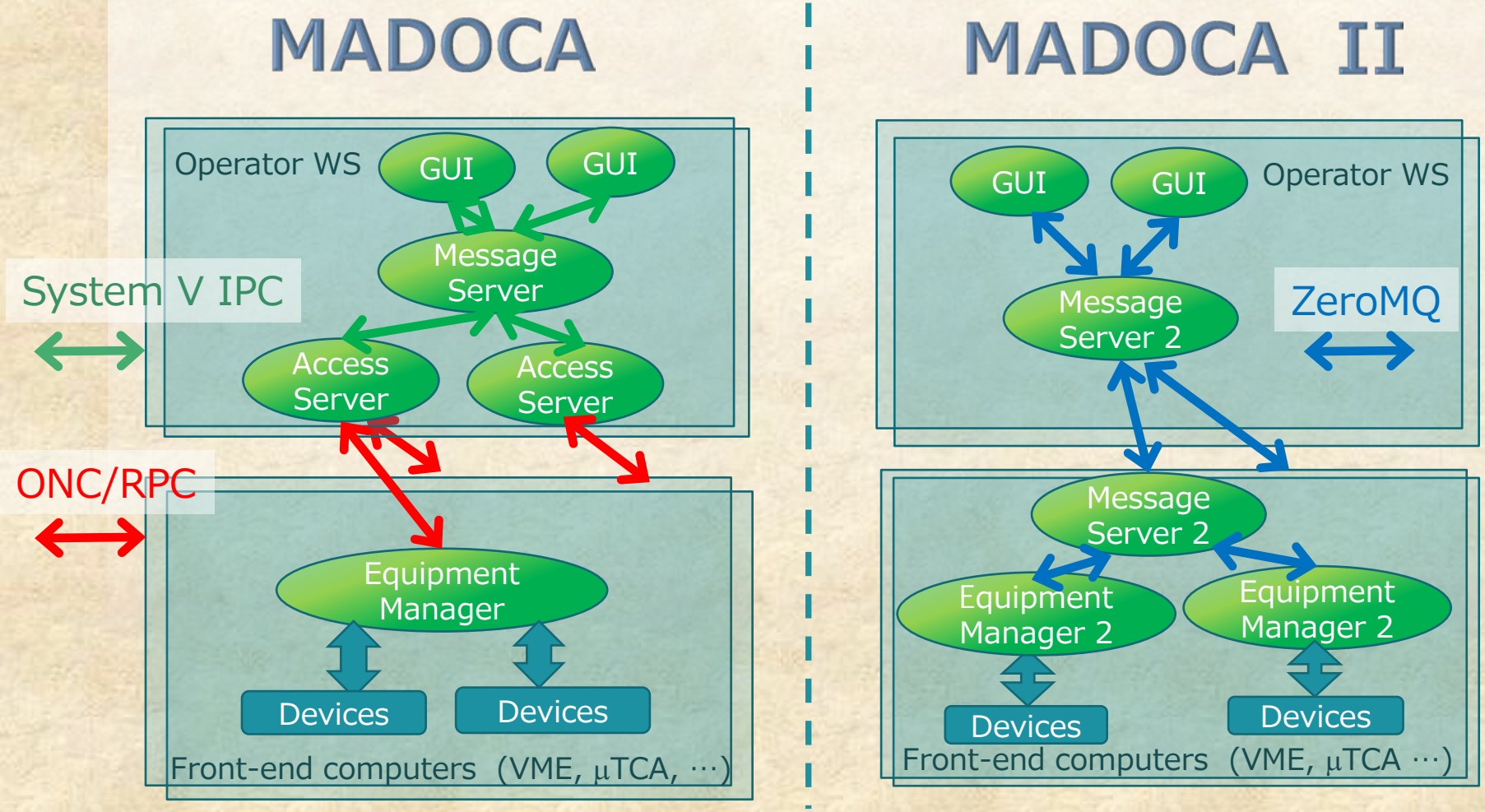


MADOCA II




MADOCA
Operator WS
GUI
GUI
Message Server
System V IPC
Access Server
Access Server
ONC/RPC
Equipment Manager
Devices
Devices
Front-end computers (VME, μTCA, ⋯)
MADOCA II
GUI
GUI
Operator WS
Message Server 2
ZeroMQ
Message Server 2
Equipment Manager 2
Devices
Devices
Front-end computers (VME, μTCA ⋯)


MADOCA
Operator WS
GUI
GUI
Message Server
System V IPC
Access Server
Access Server
ONC/RPC
Equipment Manager
Devices
Devices
Front-end computers (VME, μTCA, ⋯)
MADOCA II
GUI
GUI
Operator WS
Message Server 2
ZeroMQ
Message Server 2
Equipment Manager 2
Equipment Manager 2
Devices
Devices
Front-end computers (VME, μTCA ⋯)


MADOCA
Operator WS
GUI
GUI
Message Server
System V IPC
Access Server
Access Server
ONC/RPC
Equipment Manager
Devices
Devices
Front-end computers (VME, μTCA, ⋯)
MADOCA II
GUI
GUI
Operator WS
Message Server 2
ZeroMQ
Message Server 2
Equipment Manager 2
Equipment Manager 2
Devices
Devices
Front-end computers (VME, μTCA ⋯)

ZeroMQ @ MADOCA II (1)

✓Messaging with Variable-length data

S/V/O/C

✓Multi-OS

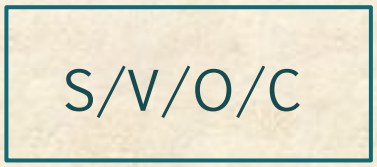
- Can be used for Linux, Solaris

✓Multi-languages

- MADOCA II has been written in C++

ZeroMQ @ MADOCA II (1)

✓Messaging with Variable-length data



✓Multi-OS

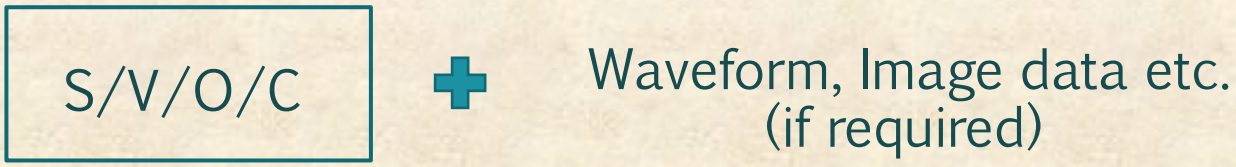
- Can be used for Linux, Solaris

✓Multi-languages

- MADOCA II has been written in C++

ZeroMQ @ MADOCA II (1)

✓Messaging with Variable-length data



✓Multi-OS

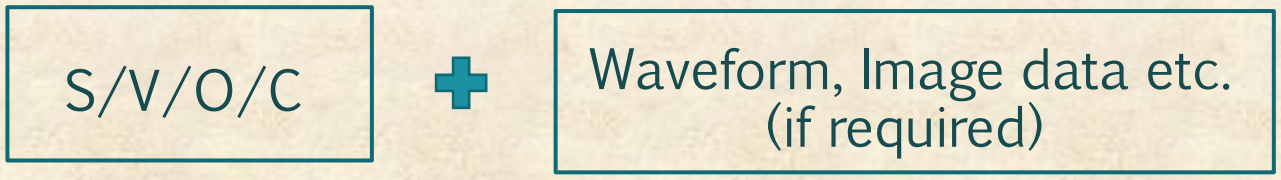
- Can be used for Linux, Solaris

✓Multi-languages

- MADOCA II has been written in C++

ZeroMQ @ MADOCA II (1)

✓Messaging with Variable-length data



✓Multi-OS

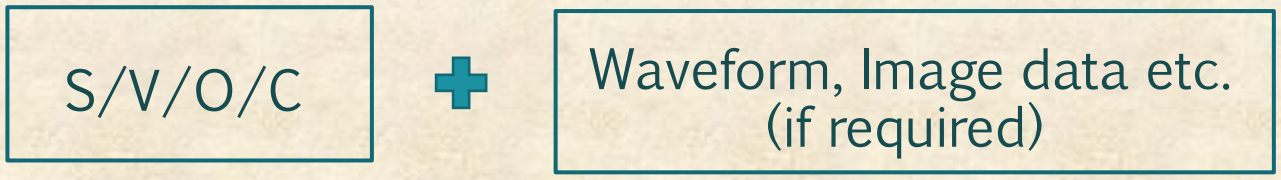
- Can be used for Linux, Solaris

✓Multi-languages

- MADOCA II has been written in C++

ZeroMQ @ MADOCA II (1)

✓Messaging with Variable-length data



✓Multi-OS

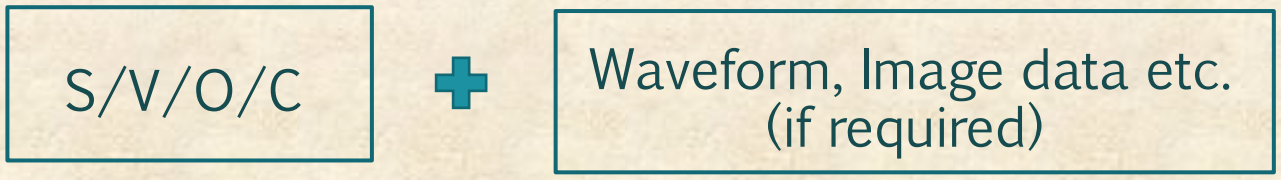
- Can be used for Linux, Solaris and Windows

✓Multi-languages

- MADOCA II has been written in C++

ZeroMQ @ MADOCA II (1)

✓Messaging with Variable-length data



✓Multi-OS

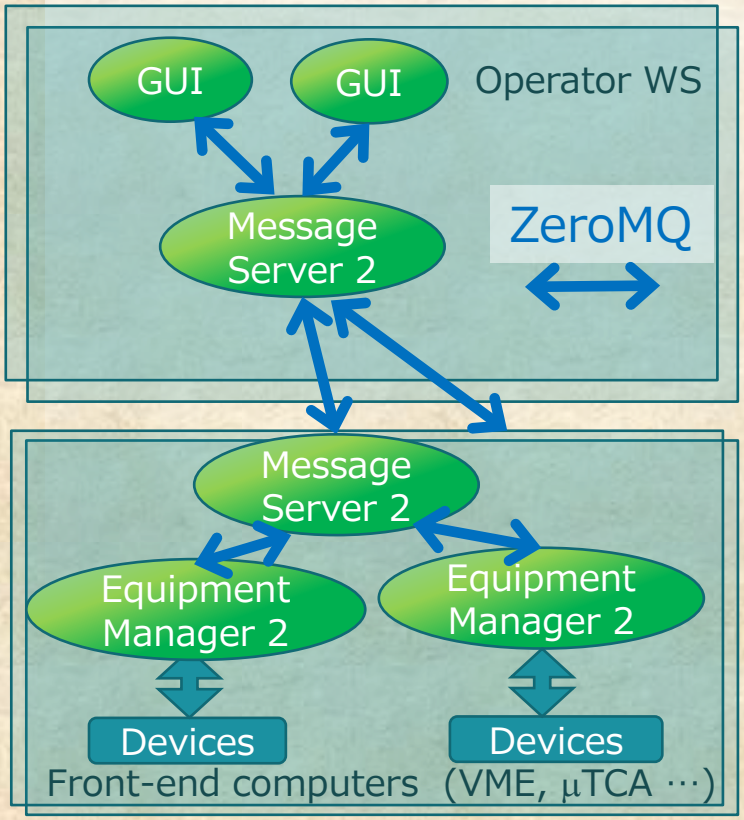
- Can be used for Linux, Solaris and Windows

✓Multi-languages

- MADOCA II has been written in C++

  Also applied to Python and LabVIEW so far

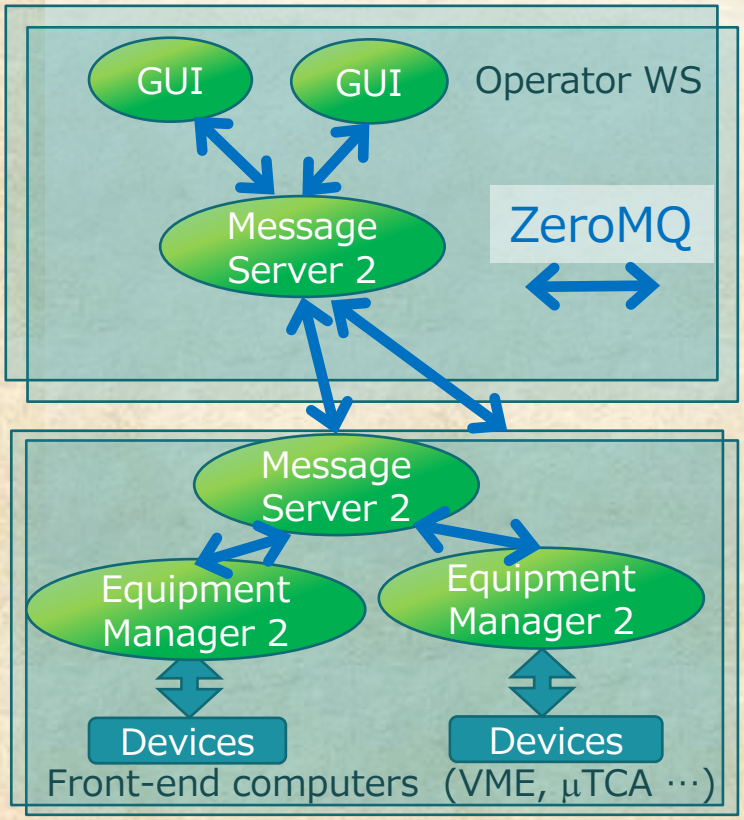
ZeroMQ @ MADOCA II (2)



✓Asynchronous communications

- No need to wait to finish processing of each message

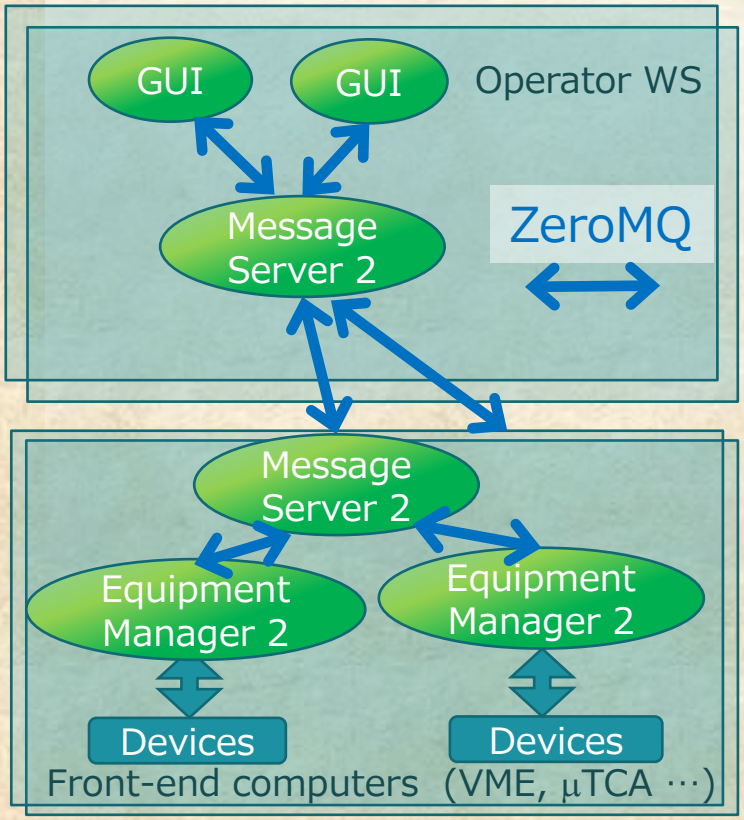
ZeroMQ @ MADOCA II (2)



✓Asynchronous communications

- No need to wait to finish processing of each message
  - Message ID is assigned to each message to handle the relation between sending and reception

ZeroMQ @ MADOCA II (2)



✓Asynchronous communications

- No need to wait to finish processing of each message
  - Message ID is assigned to each message to handle the relation between sending and reception
- Distributed processing of control messages
  - Multiple Equipment Managers can help

MessagePack @MADOCA II (1)

Used for Serialization of Messaging data

✓Message exchange among different computing environments

- byte-order procedures are managed

MessagePack @MADOCA II (1)

Used for Serialization of Messaging data

✓Message exchange among different computing environments

- byte-order procedures are managed

MessagePack @MADOCA II (1)

Used for Serialization of Messaging data

✓Message exchange among different computing environments

- byte-order procedures are managed

✓Flexible data formats

→ Applied to variable-length data, No size limit

MessagePack @MADOCA II (1)

Used for Serialization of Messaging data

✓Message exchange among different computing environments

- byte-order procedures are managed

✓Flexible data formats

→ Applied to variable-length data, No size limit

Array

MessagePack @MADOCA II (1)

Used for Serialization of Messaging data

✓Message exchange among different computing environments

- byte-order procedures are managed

✓Flexible data formats

→ Applied to variable-length data, No size limit

Array [1.23, 4.56,7.89, ⋯]

MessagePack @MADOCA II (1)

Used for Serialization of Messaging data

✓Message exchange among different computing environments

- byte-order procedures are managed

✓Flexible data formats

→ Applied to variable-length data, No size limit

Array [1.23, 4.56,7.89, ⋯]

Map

MessagePack @MADOCA II (1)

Used for Serialization of Messaging data

✓Message exchange among different computing environments

- byte-order procedures are managed

✓Flexible data formats

→ Applied to variable-length data, No size limit

Array [1.23, 4.56,7.89, …]

Map { “image_data_type” : “MONO”,
“image_data”: [1,3,5,…] }

MessagePack @MADOCA II (1)

Used for Serialization of Messaging data

✓Message exchange among different computing environments

- byte-order procedures are managed

✓Flexible data formats

→ Applied to variable-length data, No size limit

Array [1.23, 4.56,7.89, ⋯]

Map { “image_data_type” : “MONO”, “image_data”: [1,3,5,⋯] }

→ Waveform

MessagePack @MADOCA II (1)

Used for Serialization of Messaging data

✓Message exchange among different computing environments

- byte-order procedures are managed

✓Flexible data formats

→ Applied to variable-length data, No size limit

Array [1.23, 4.56,7.89, ⋯] → Waveform

Map { “image_data_type” : “MONO”, “image_data”: [1,3,5,⋯] } → Image

MessagePack @MADOCA II (2)

Case for Image:

Key	Data format	Value
image_data_type	string	“MONO”, ”RGB”, ”RGBA”
image_width	int32_t	
image_height	int32_t	
image_depth	int32_t	
image_num_type	string	“int32_t”,”uint16_t,” “uint32_t”,”uint64_t,” “int16_t”, “int32_t”,”int64_t”,”float”,”double”
image_data	defined by [image_num_type]	→Array
image_pixel_order	string	“lefttop”, “leftbottom”

MessagePack @MADOCA II (2)

Case for Image:

Key	Data format	Value
image_data_type	string	“MONO”, ”RGB”, ”RGBA”
image_width	int32_t	
image_height	int32_t	
image_depth	int32_t	
image_num_type	string	“int32_t”,”uint16_t,” “uint32_t”,”uint64_t,” “int16_t”, “int32_t”,”int64_t”,”float”,”double”
image_data	defined by [image_num_type]	→Array
image_pixel_order	string	“lefttop”, “leftbottom”

Key’s can be flexibly added by users

MADOCA II Performance (1)

Operator WS

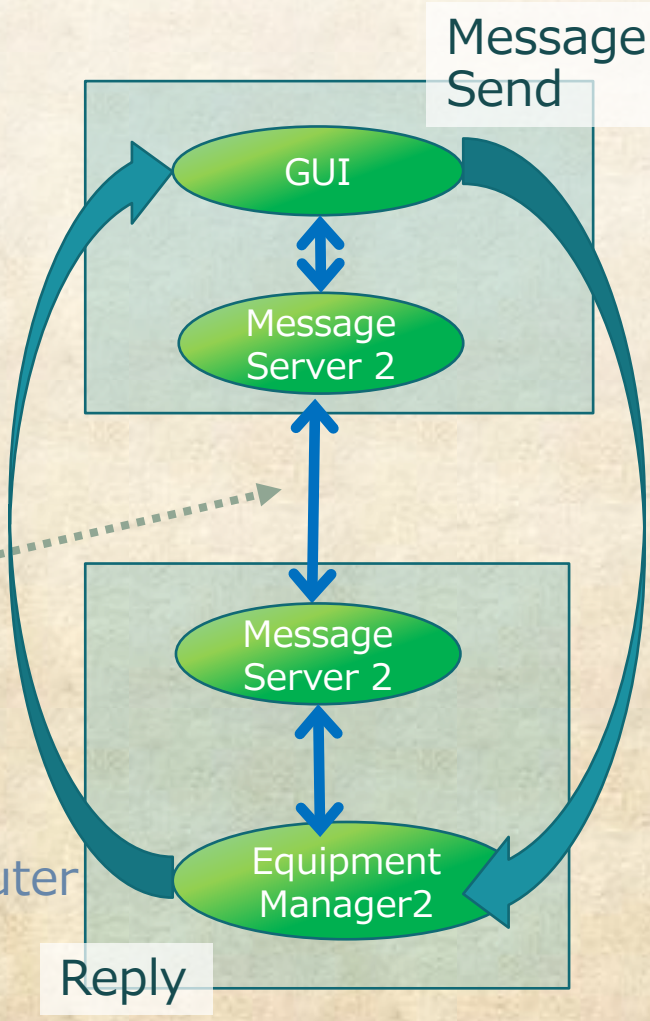
Core i7-2670QM
2.2GHz

Network

GbE cable
* Direct connection

Front-end computer

Core 2Duo P9500
2.53GHz

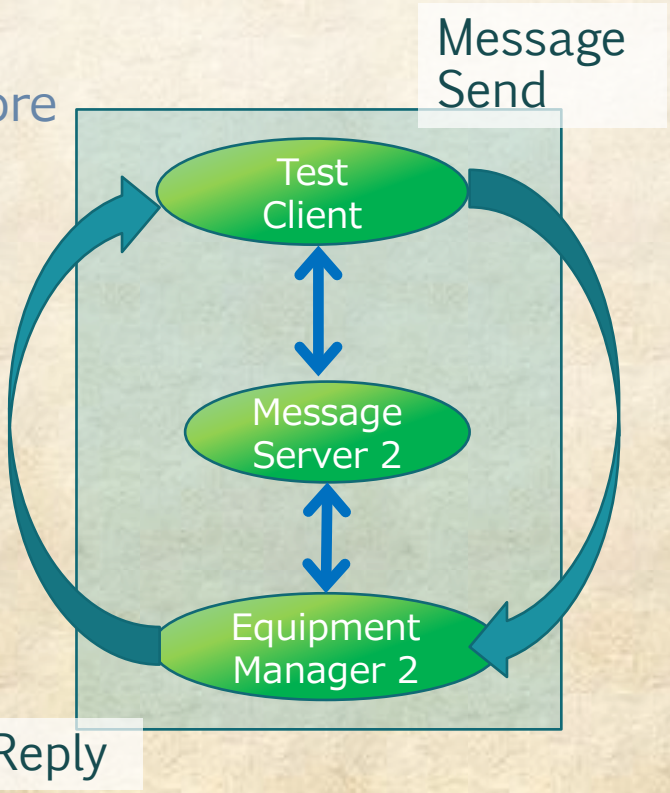


Message size
80 byte

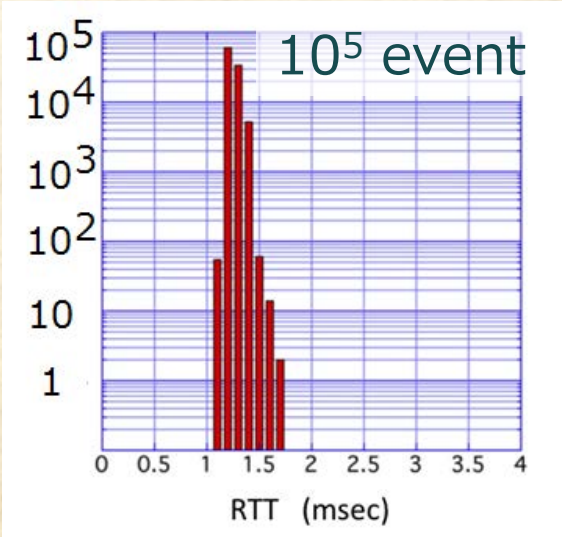
- Round Trip Time (RTT)
  - 1.2 msec / message
  - > 850 Hz

MADOCA II Performance (2)

Solaris
Multi-core
CPU



- "Real-time Process control on Multi-core Processors"
  - MOPPC128 M. Ishii et al.

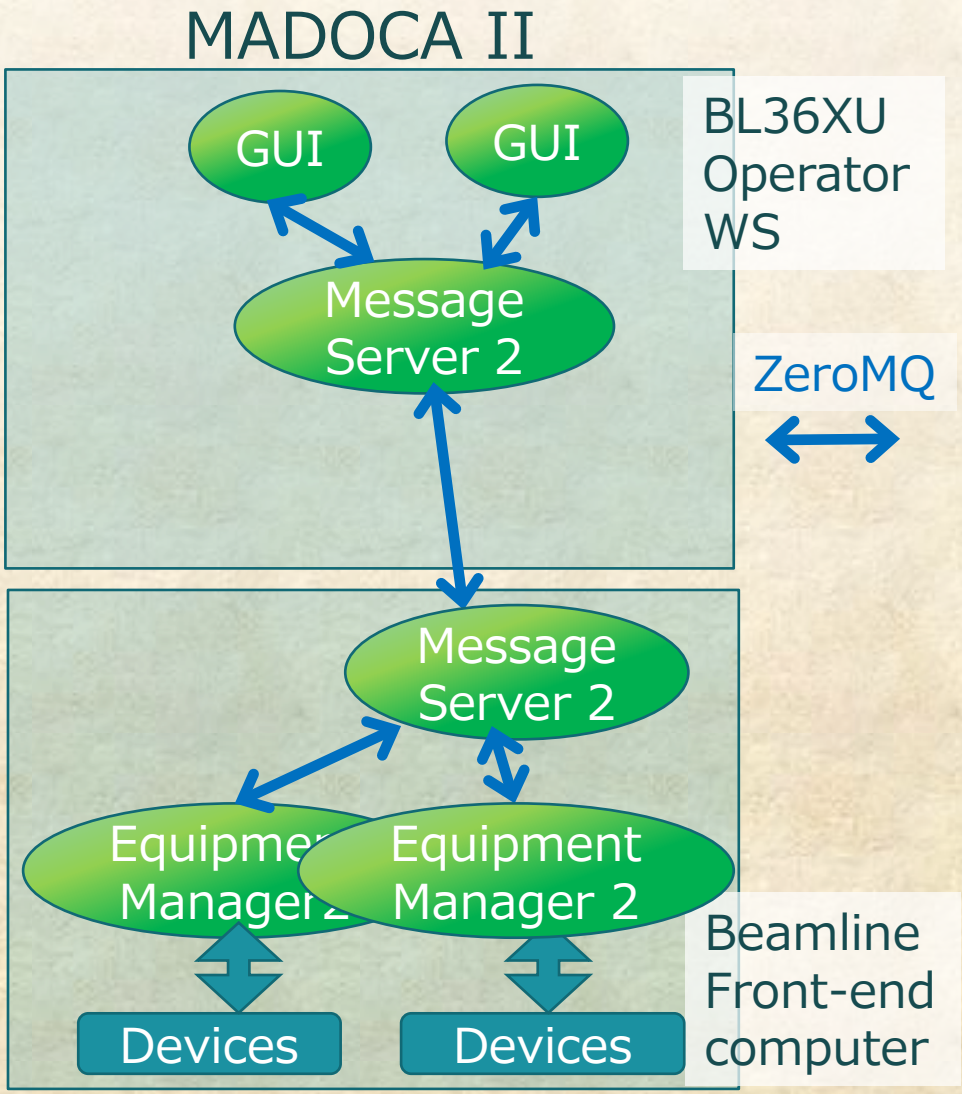


- RTT < 2ms
  - With binding processes into real-time class

→ Real-time control

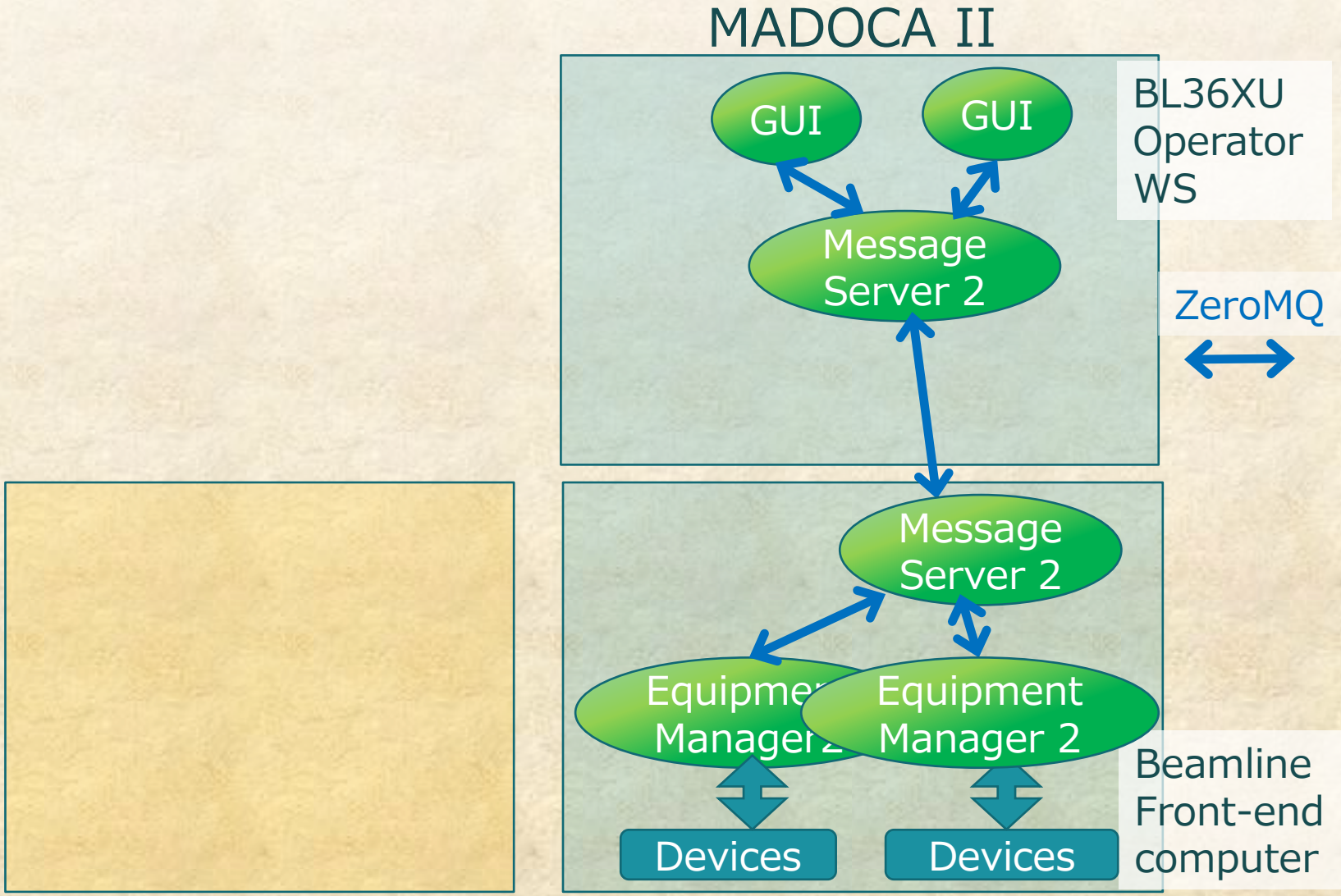
MADOCA II @ BL36XU beamline

Since Sep. 2012



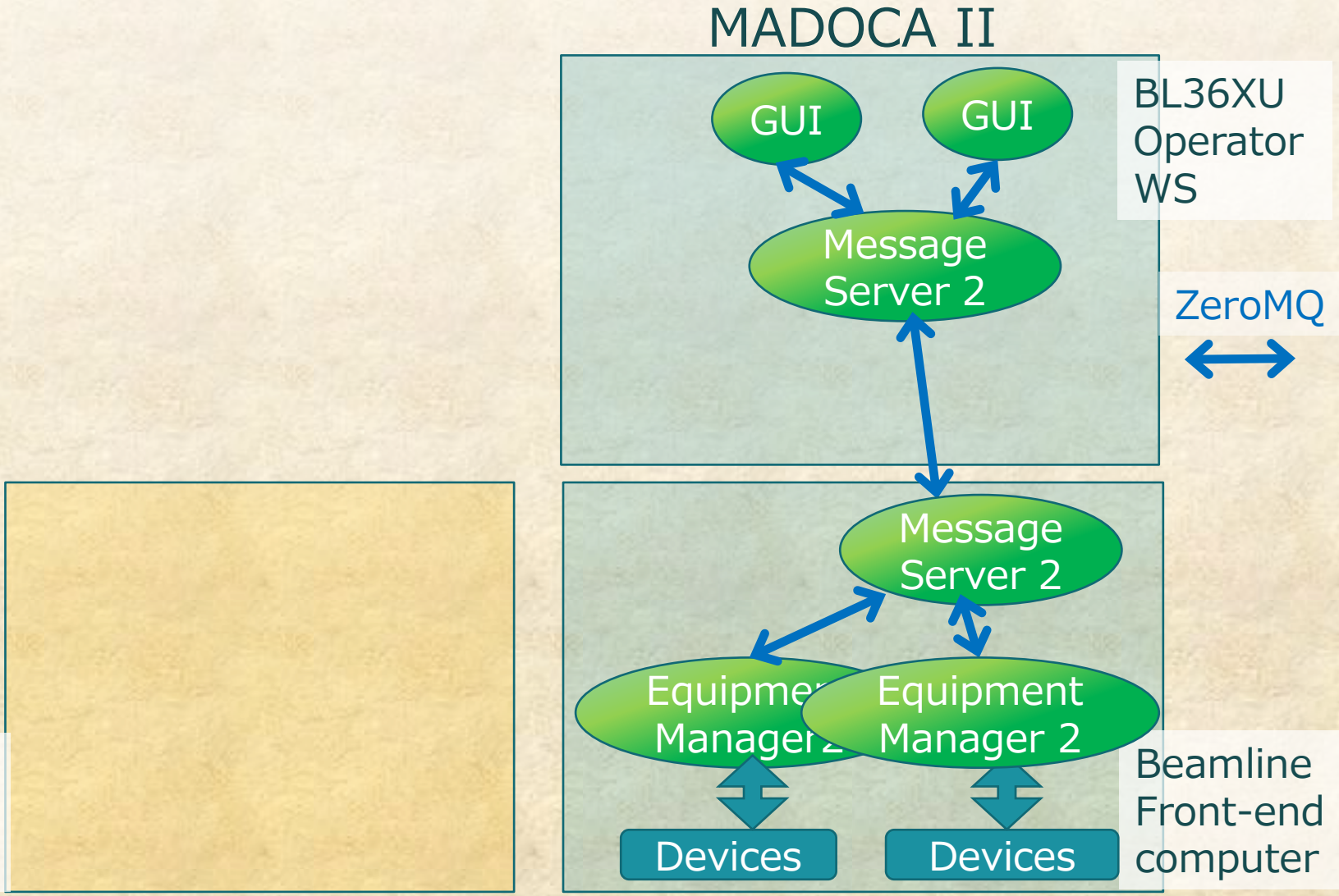
MADOCA II @ BL36XU beamline

Since Sep. 2012



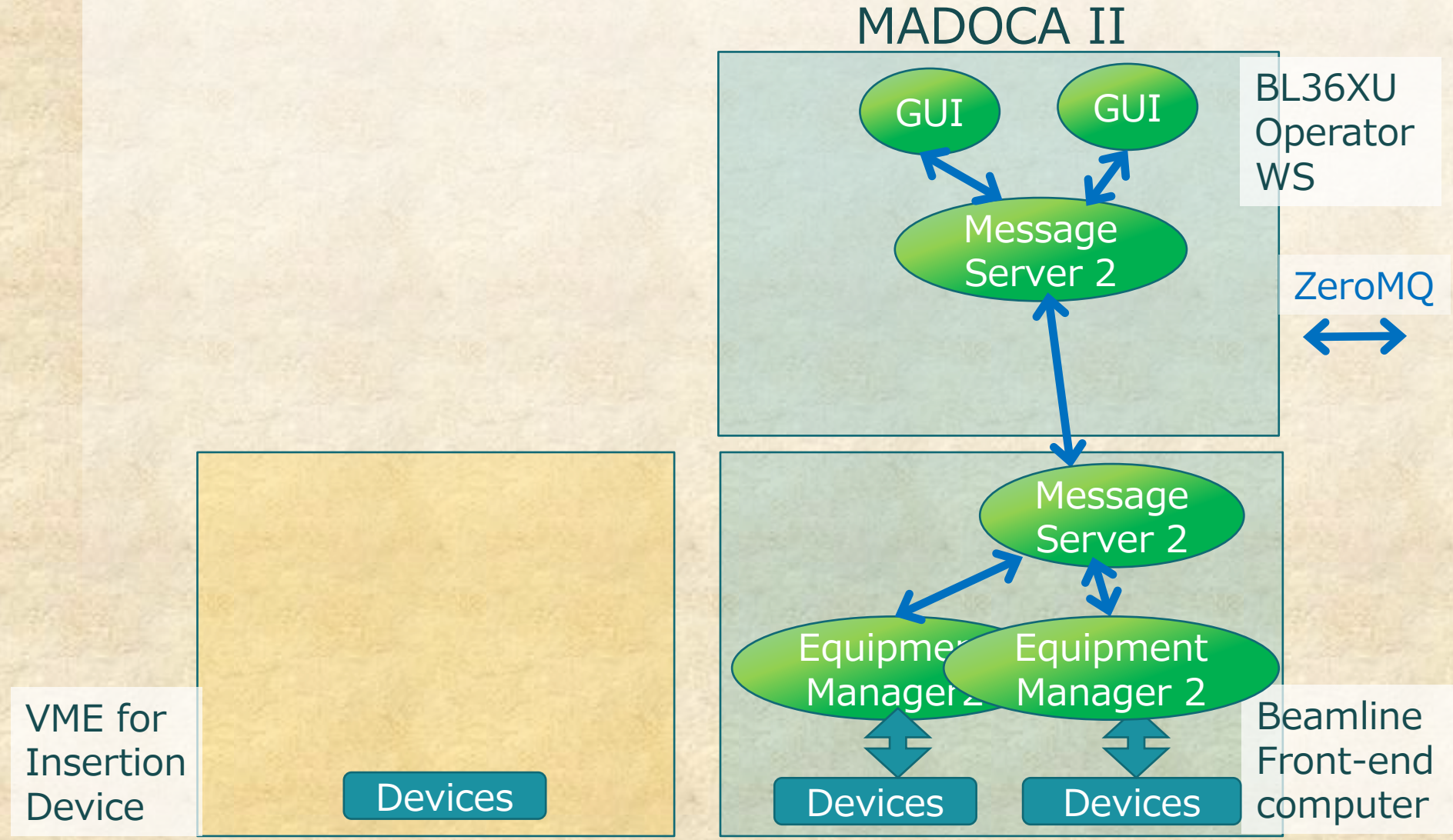
MADOCA II @ BL36XU beamline

Since Sep. 2012



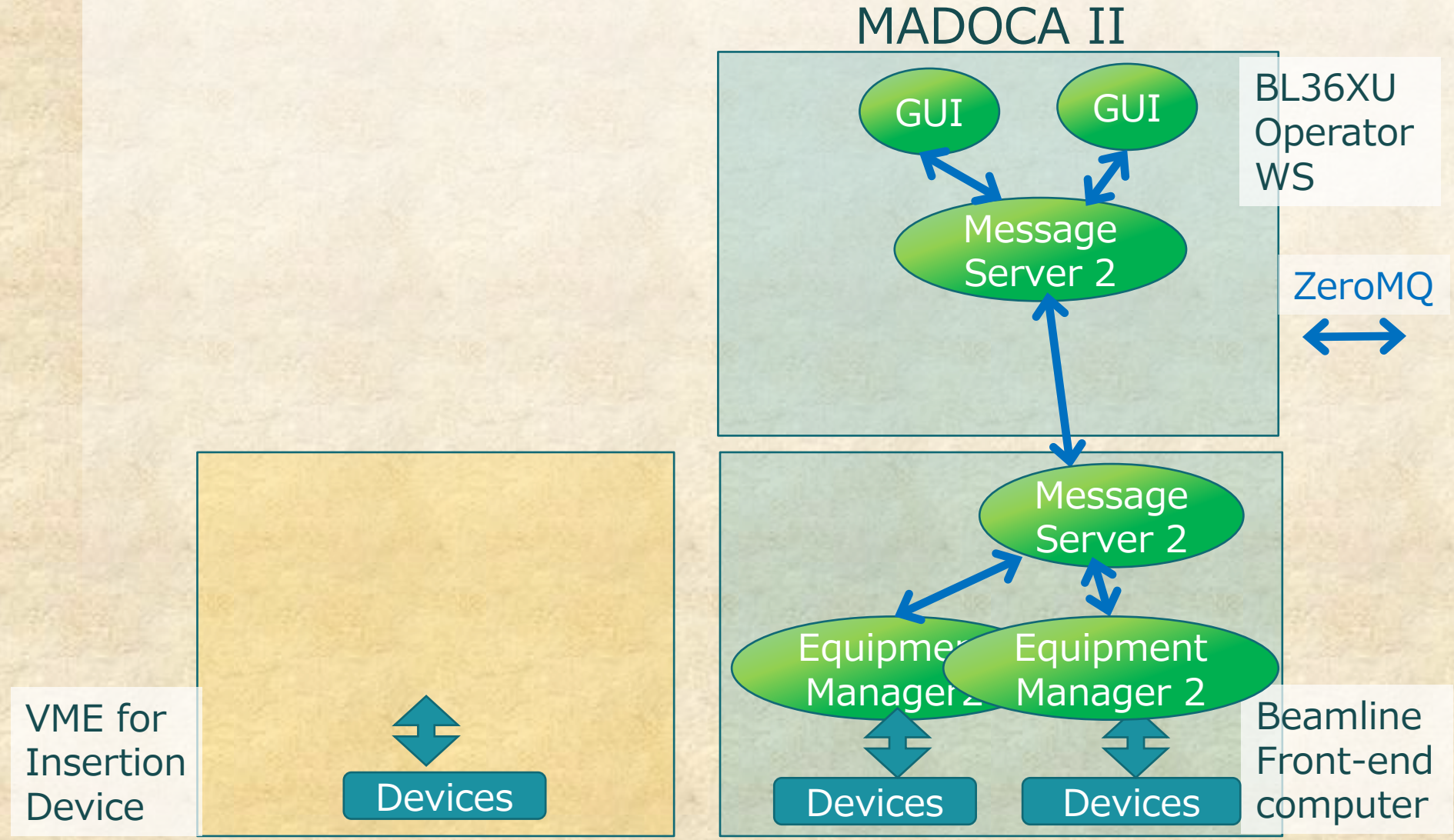
MADOCA II @ BL36XU beamline

Since Sep. 2012



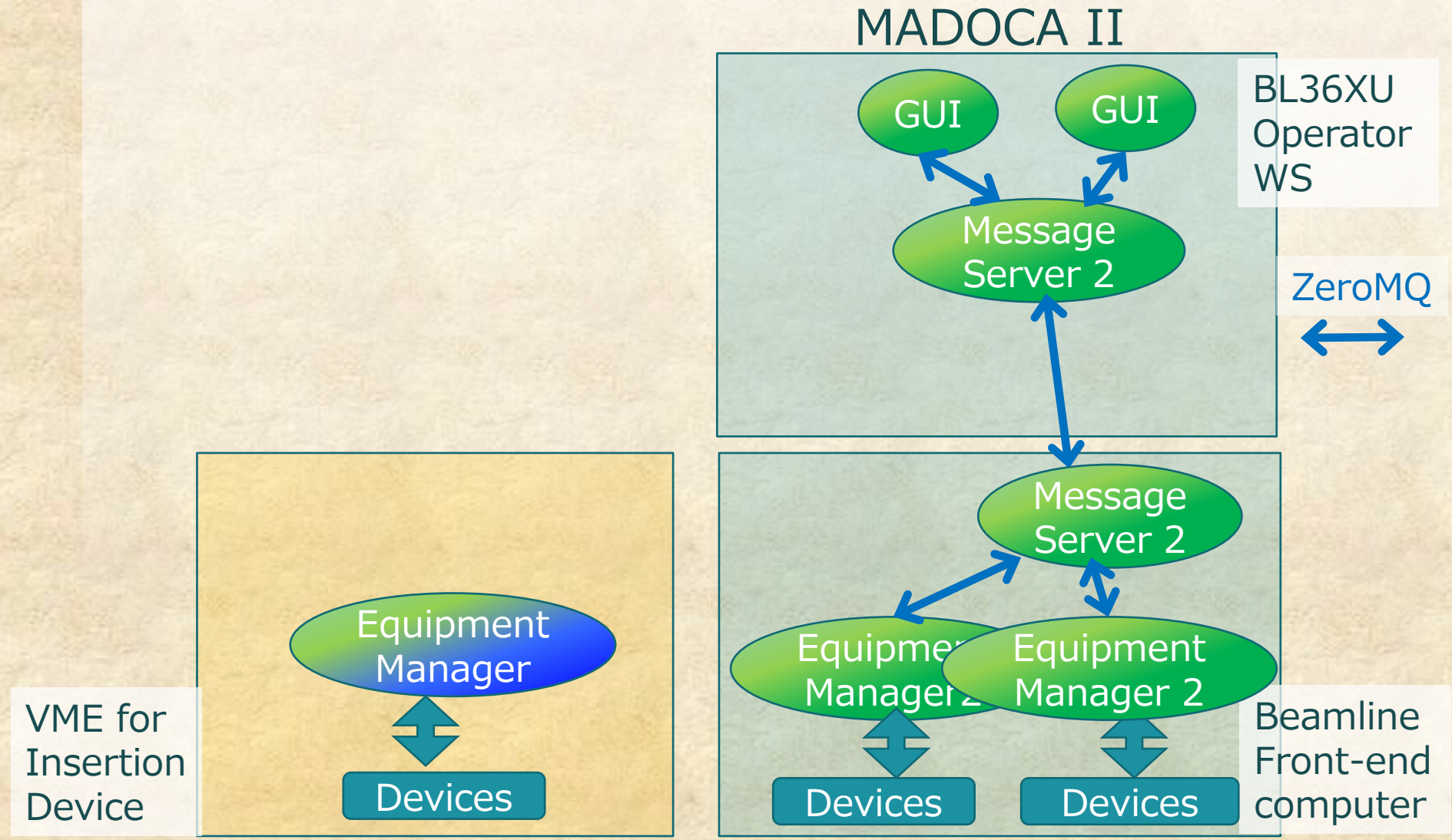
MADOCA II @ BL36XU beamline

Since Sep. 2012



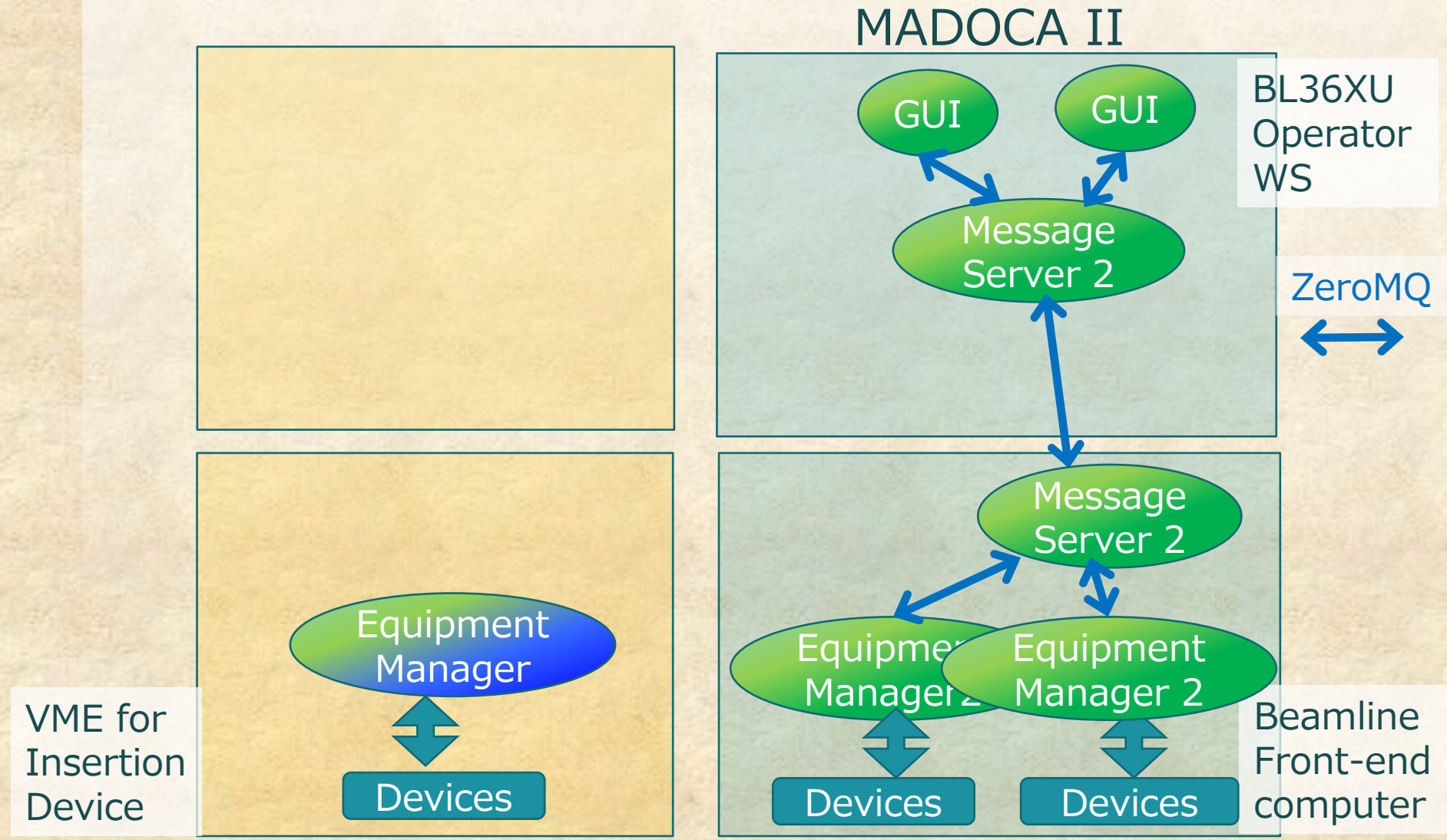
MADOCA II @ BL36XU beamline

Since Sep. 2012



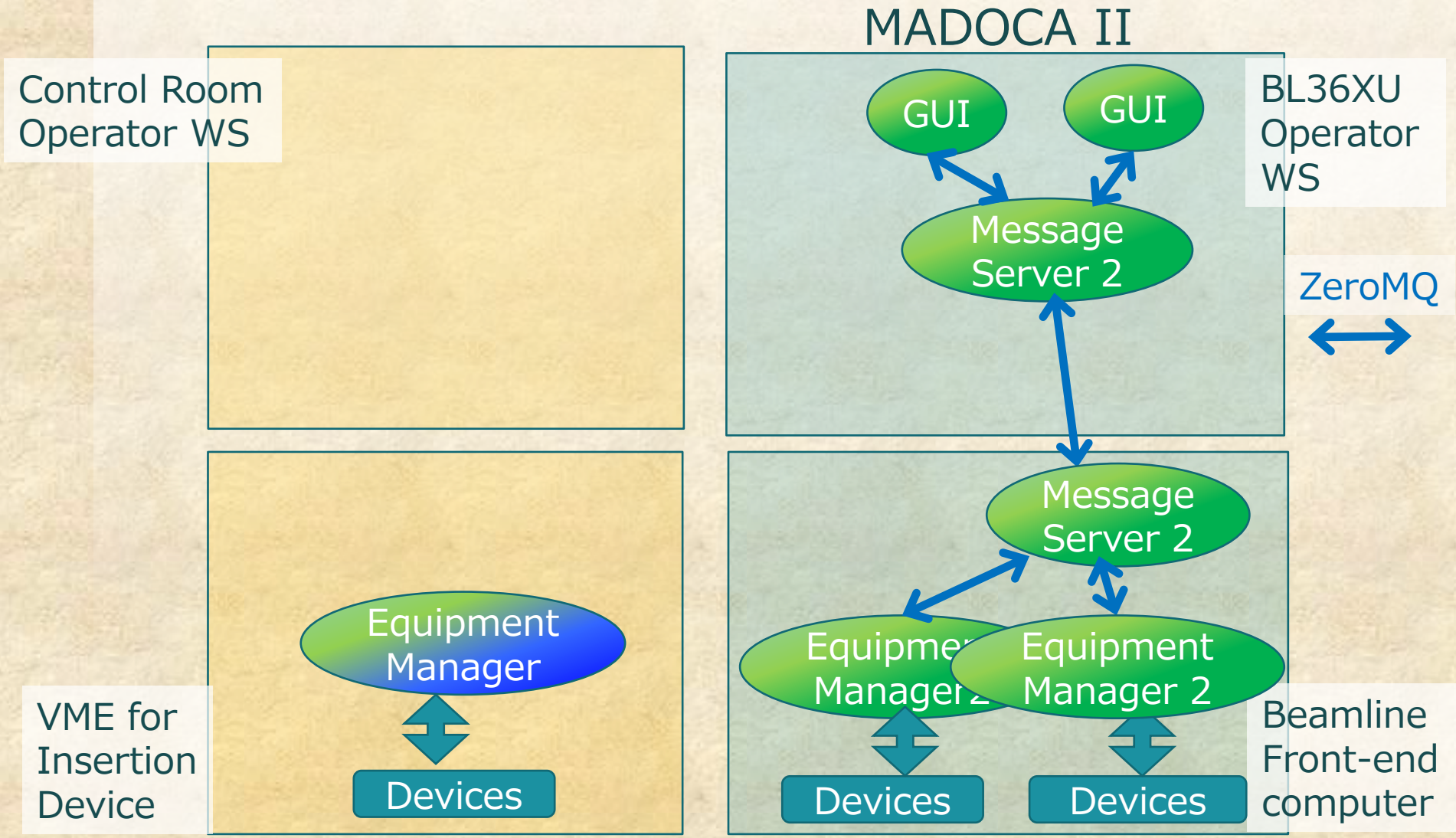
MADOCA II @ BL36XU beamline

Since Sep. 2012



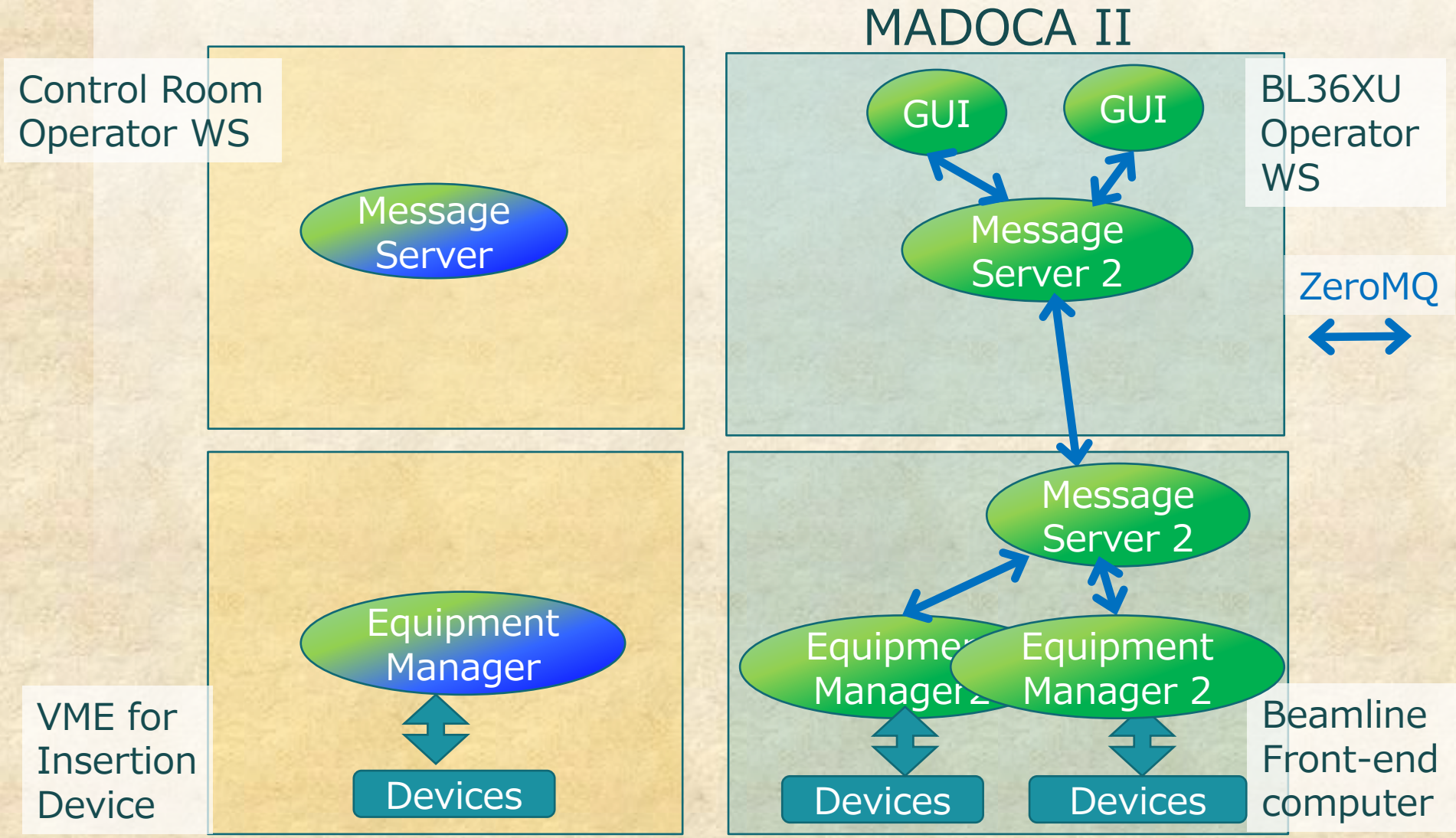
MADOCA II @ BL36XU beamline

Since Sep. 2012



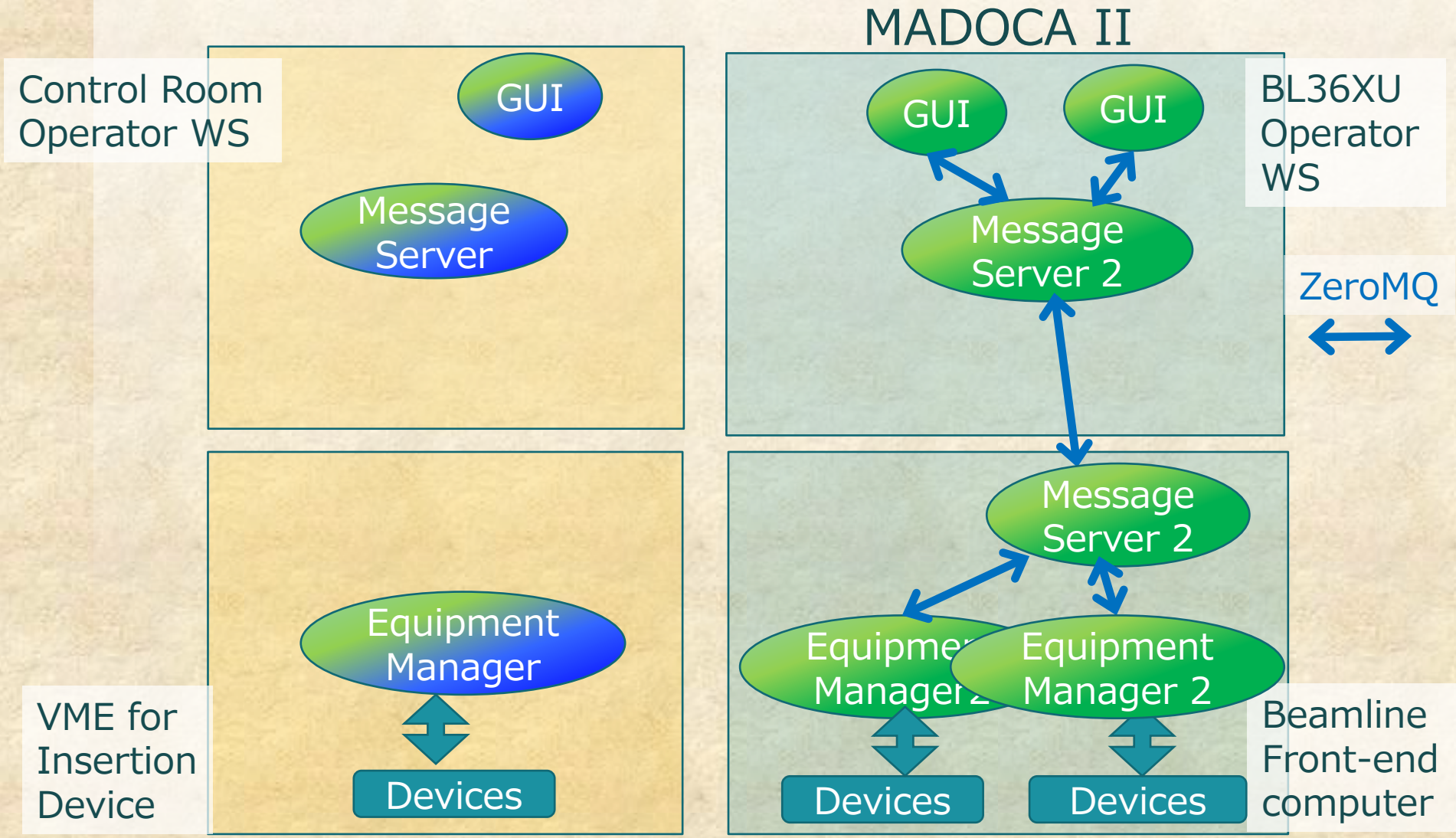
MADOCA II @ BL36XU beamline

Since Sep. 2012



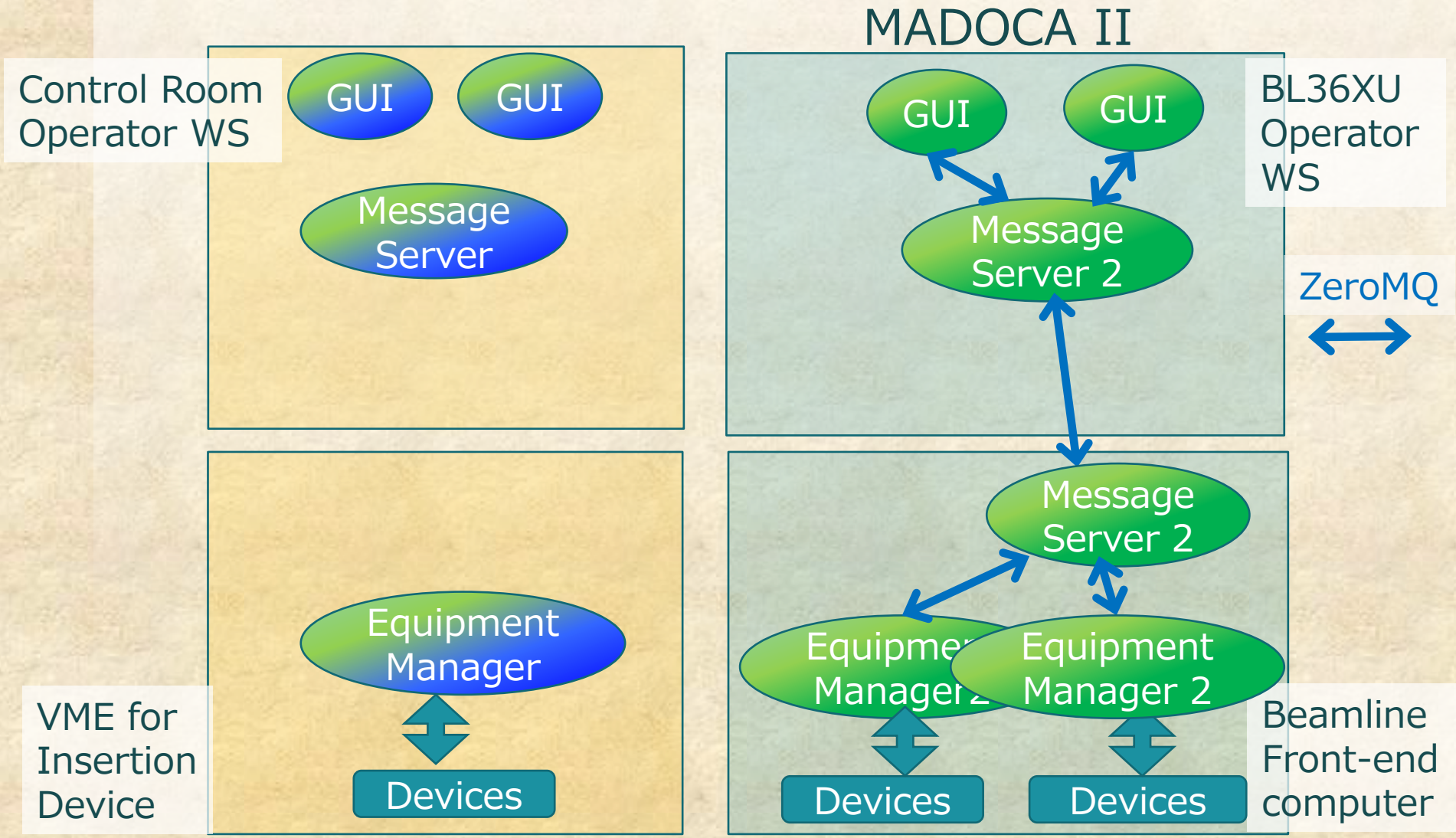
MADOCA II @ BL36XU beamline

Since Sep. 2012



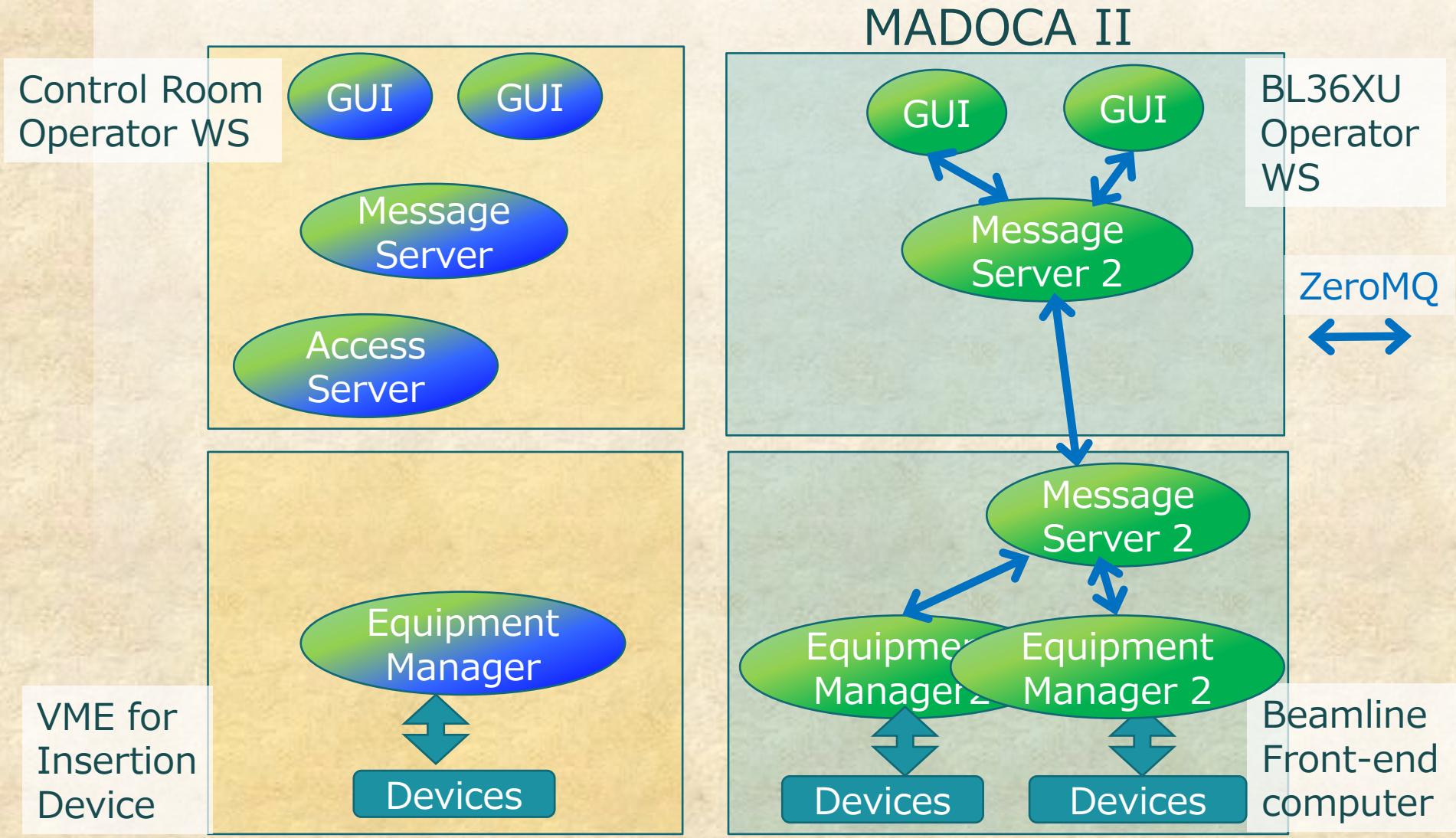
MADOCA II @ BL36XU beamline

Since Sep. 2012



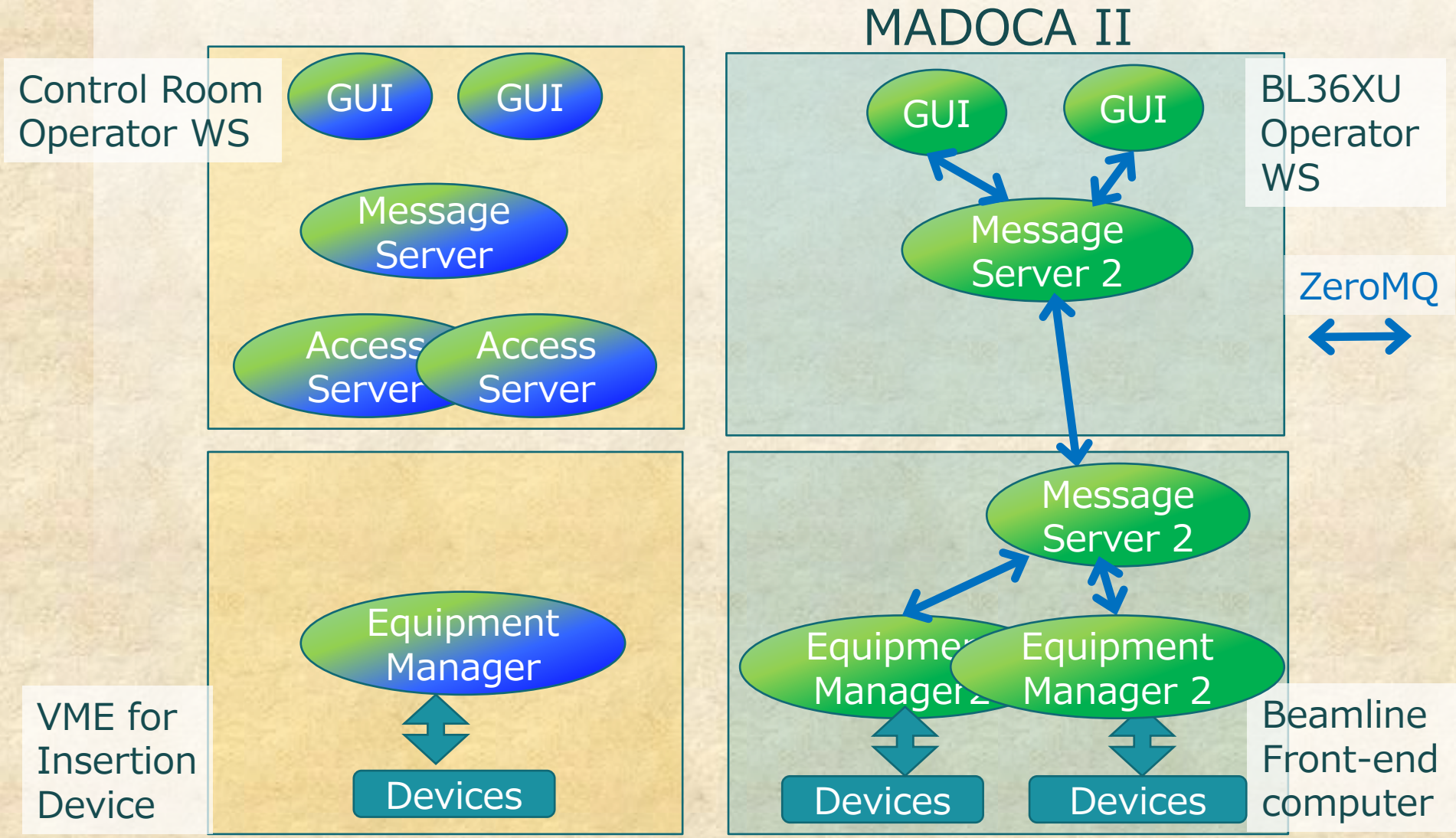
MADOCA II @ BL36XU beamline

Since Sep. 2012



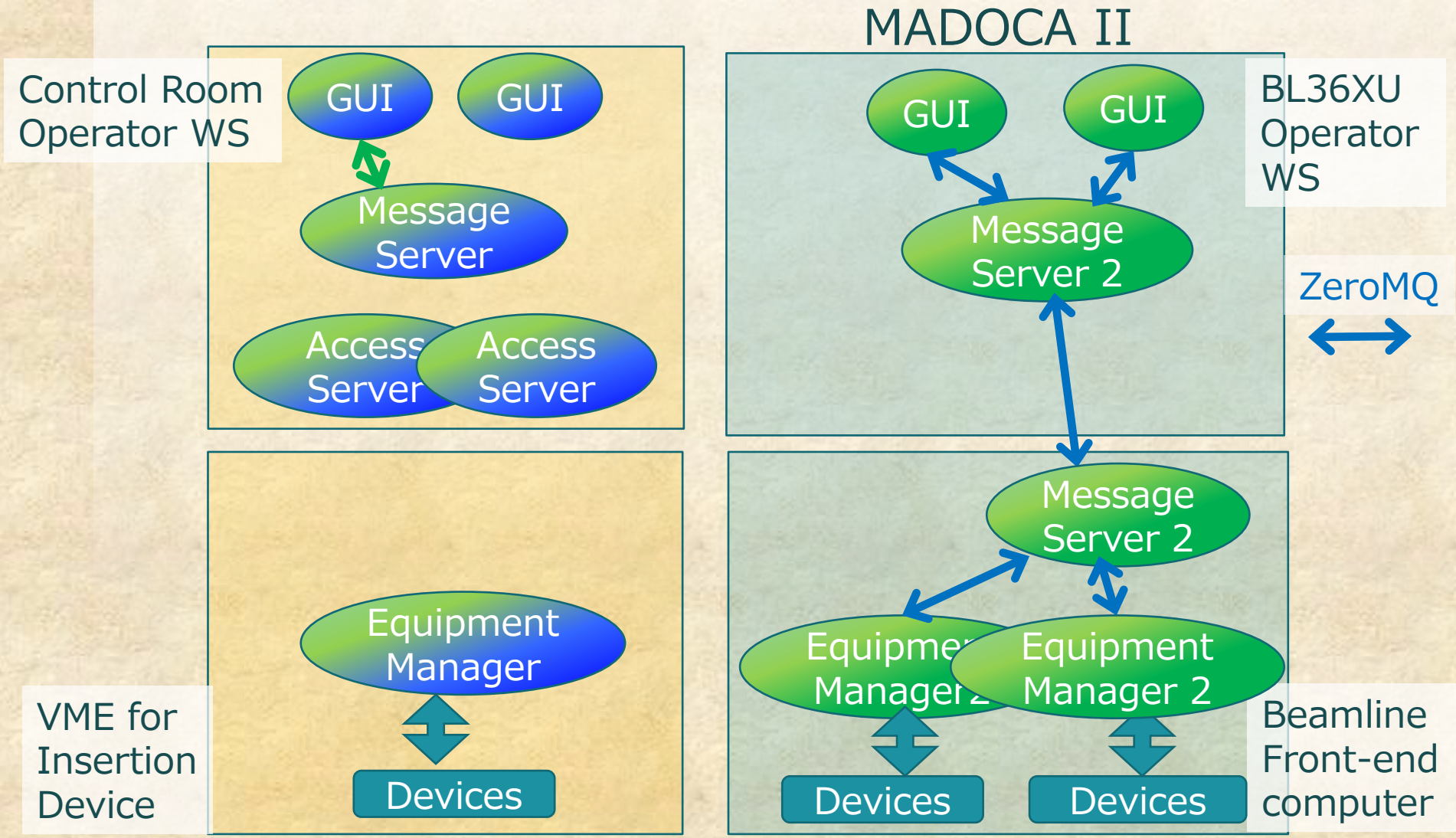
MADOCA II @ BL36XU beamline

Since Sep. 2012



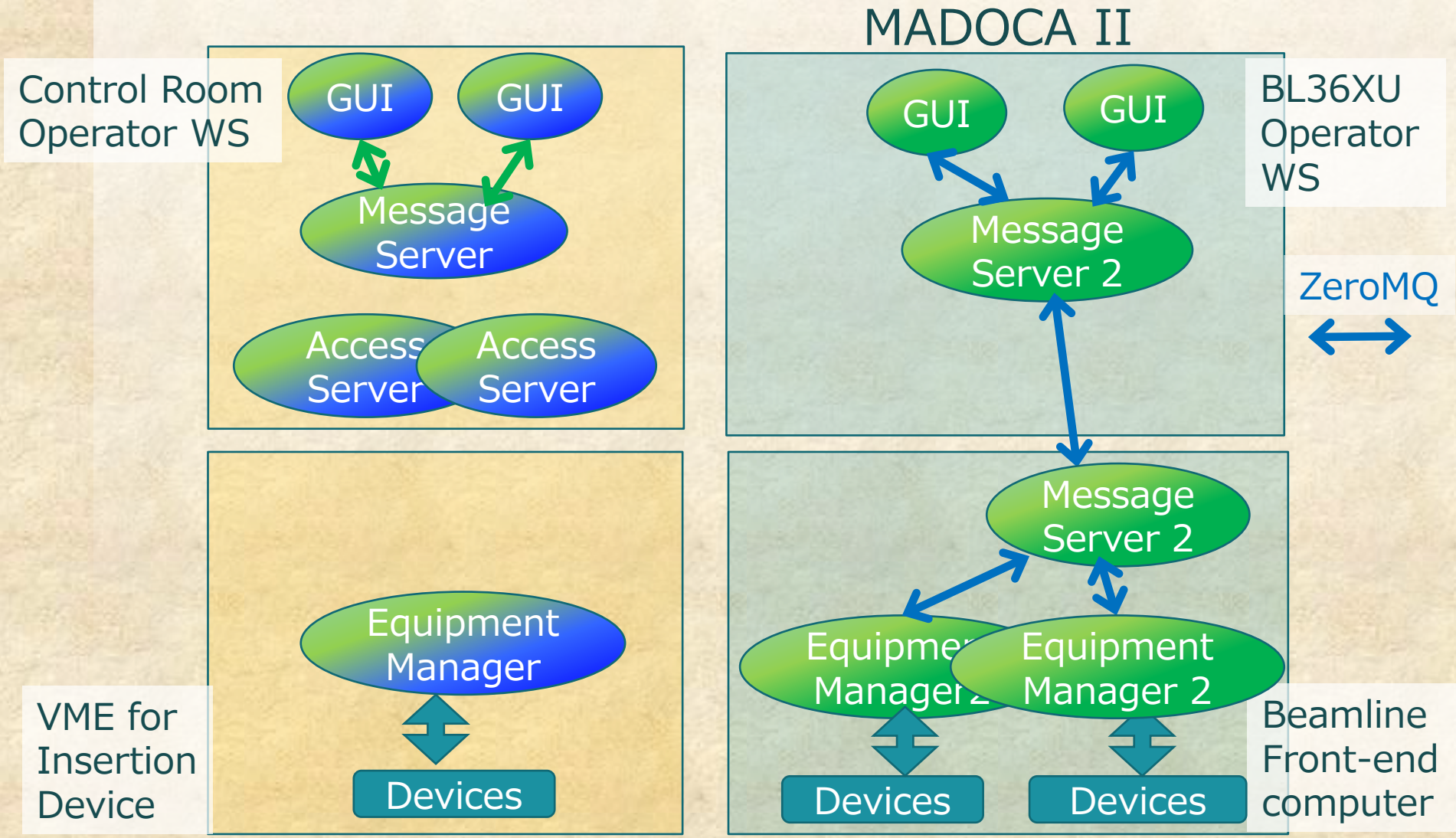
MADOCA II @ BL36XU beamline

Since Sep. 2012



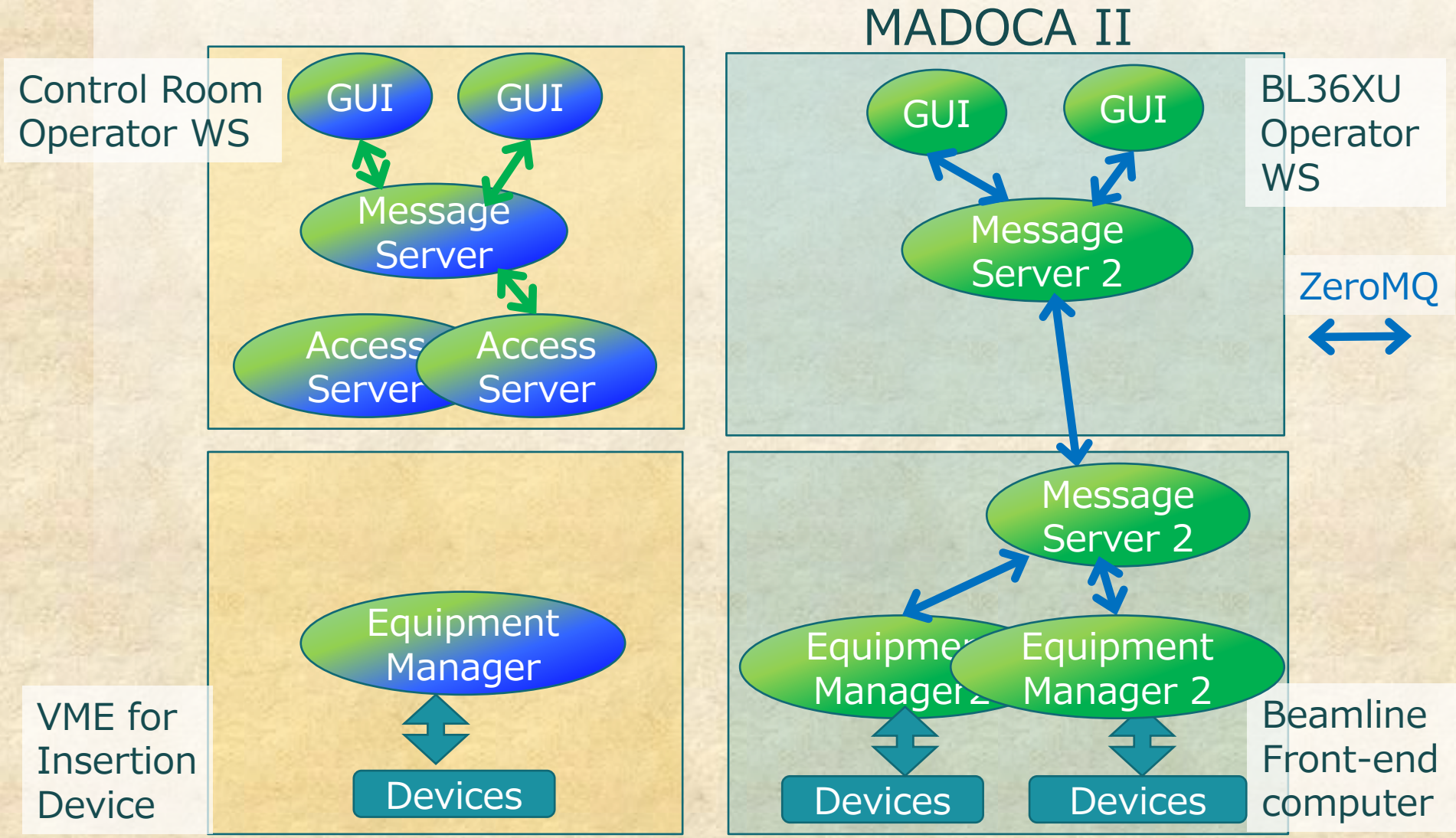
MADOCA II @ BL36XU beamline

Since Sep. 2012



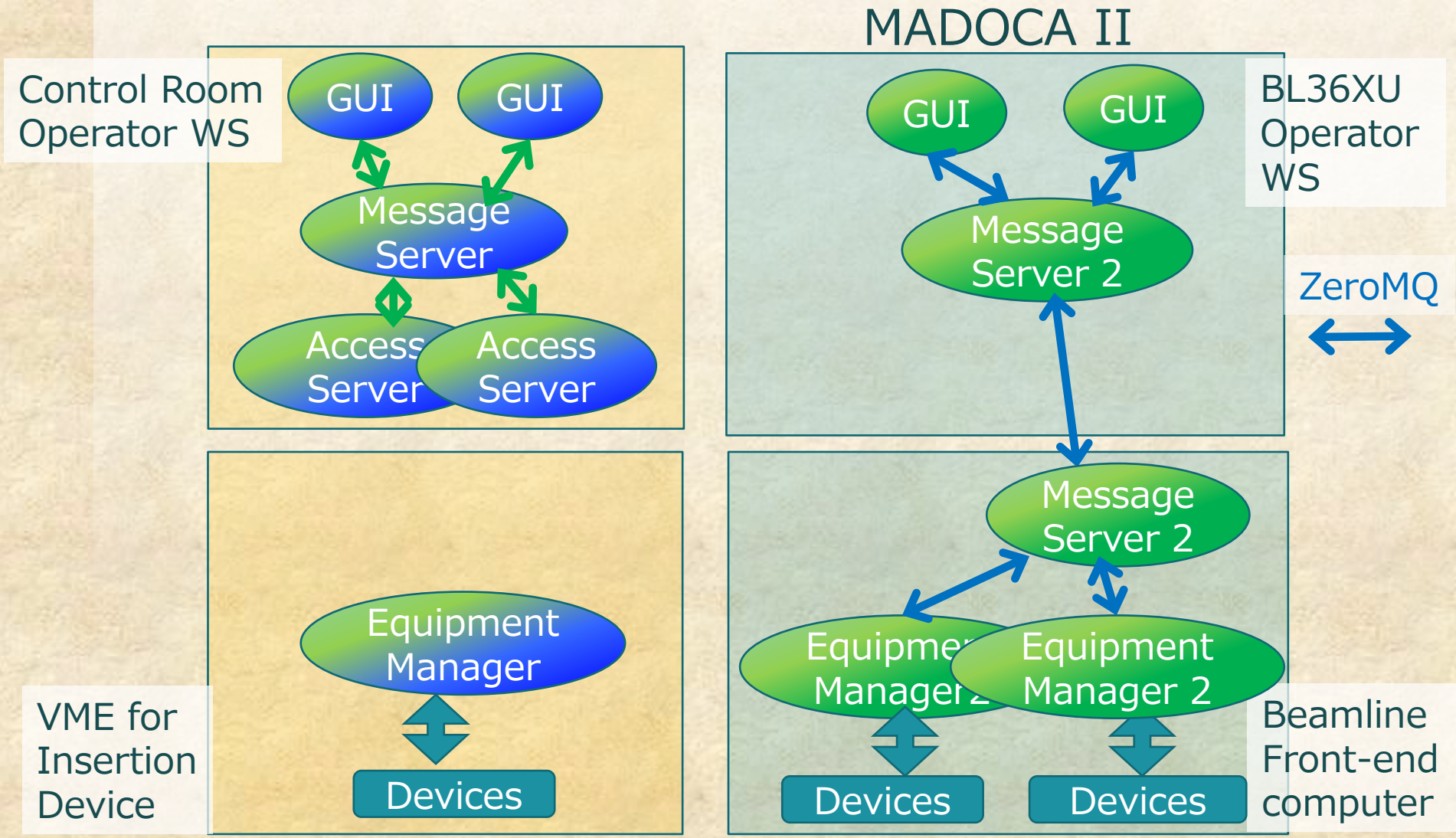
MADOCA II @ BL36XU beamline

Since Sep. 2012



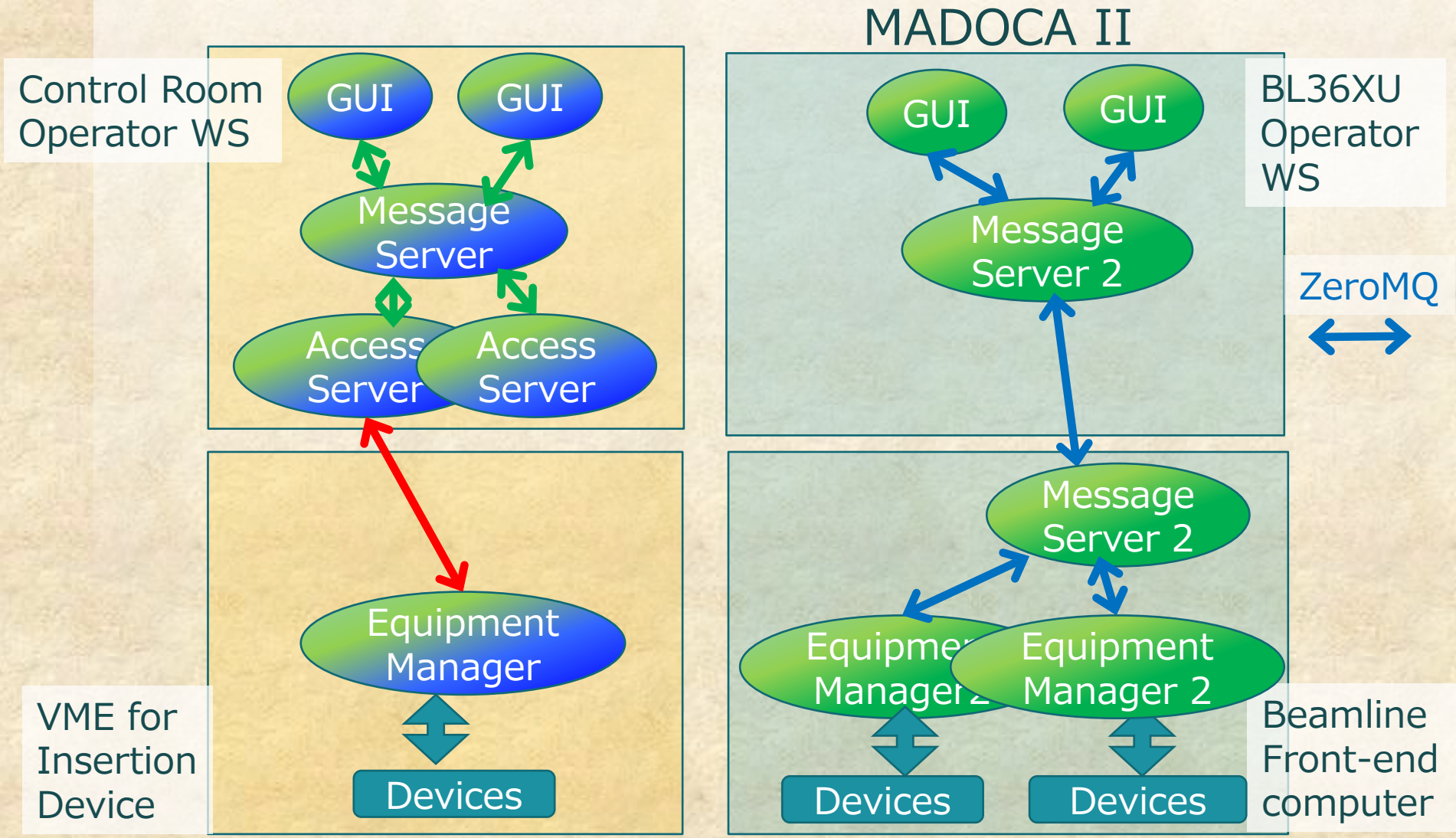
MADOCA II @ BL36XU beamline

Since Sep. 2012



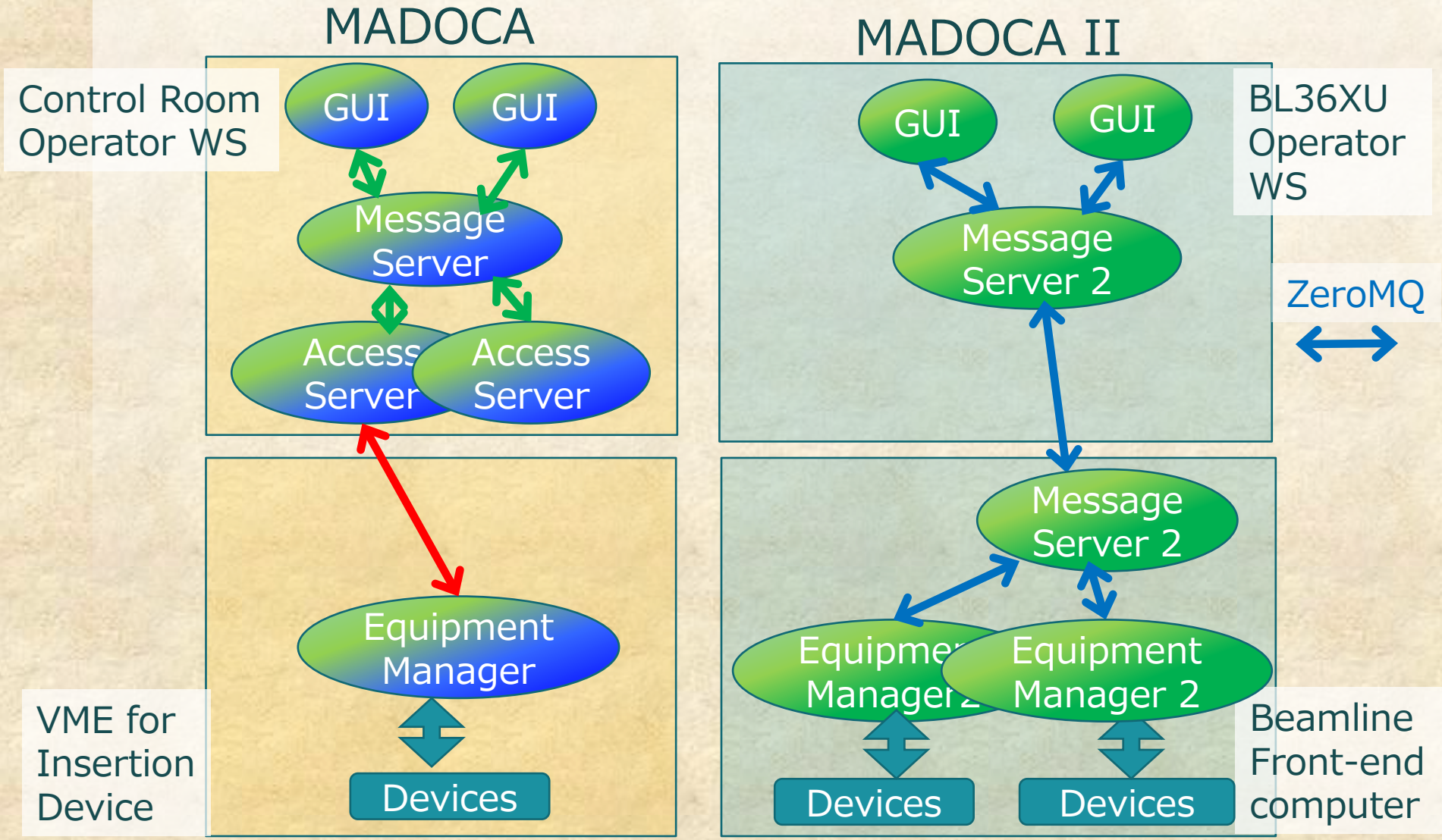
MADOCA II @ BL36XU beamline

Since Sep. 2012



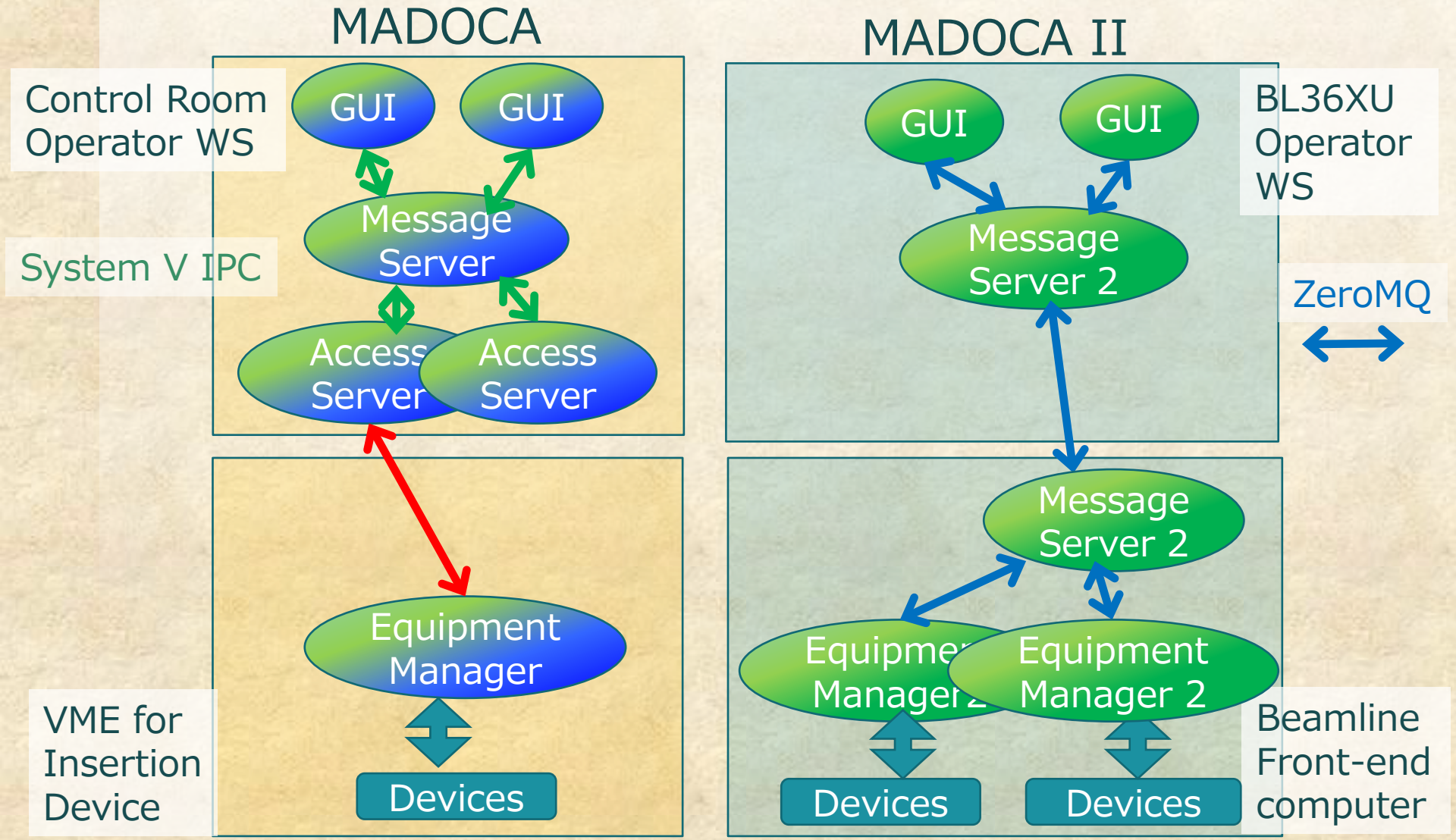
MADOCA II @ BL36XU beamline

Since Sep. 2012



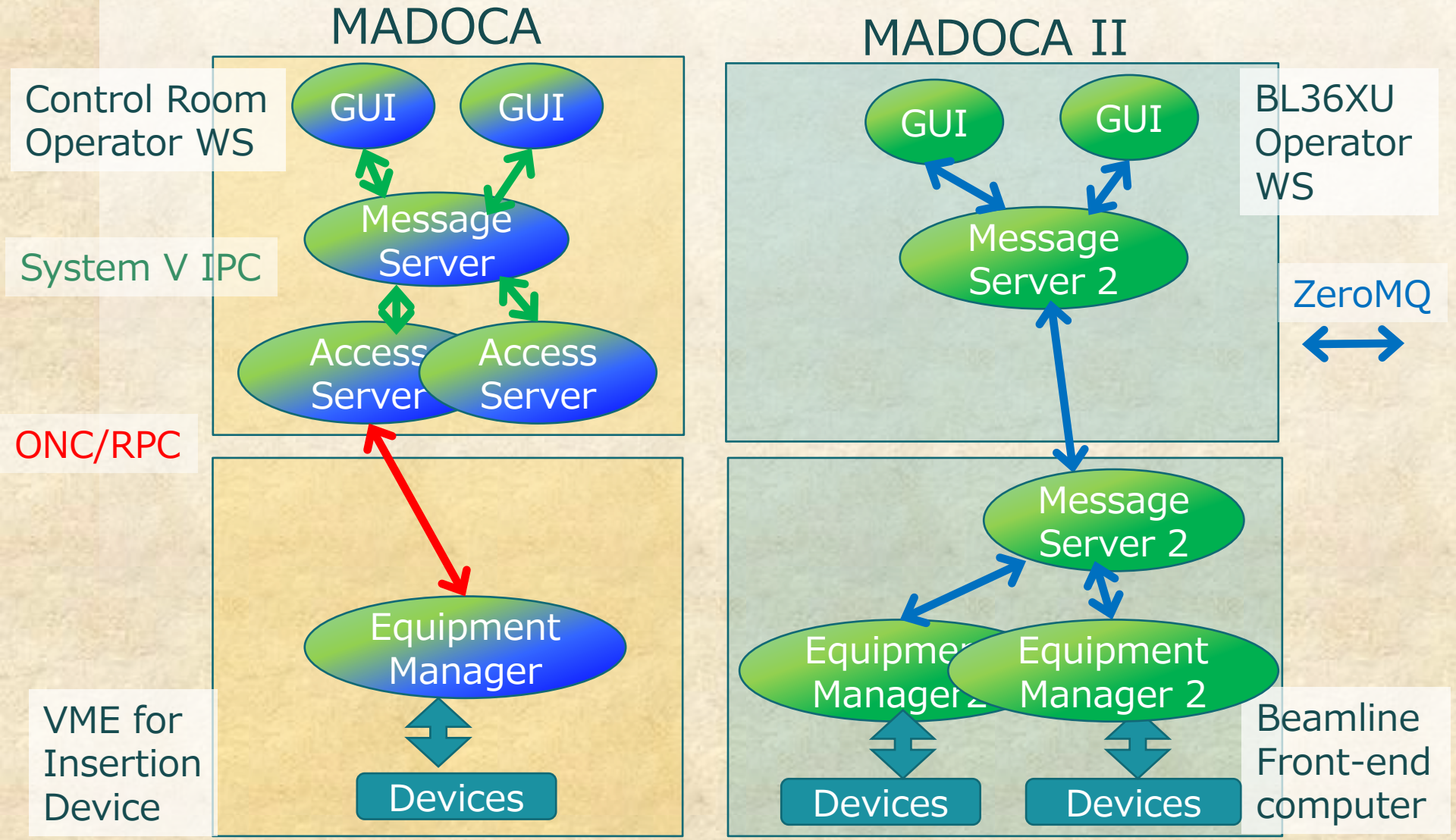
MADOCA II @ BL36XU beamline

Since Sep. 2012



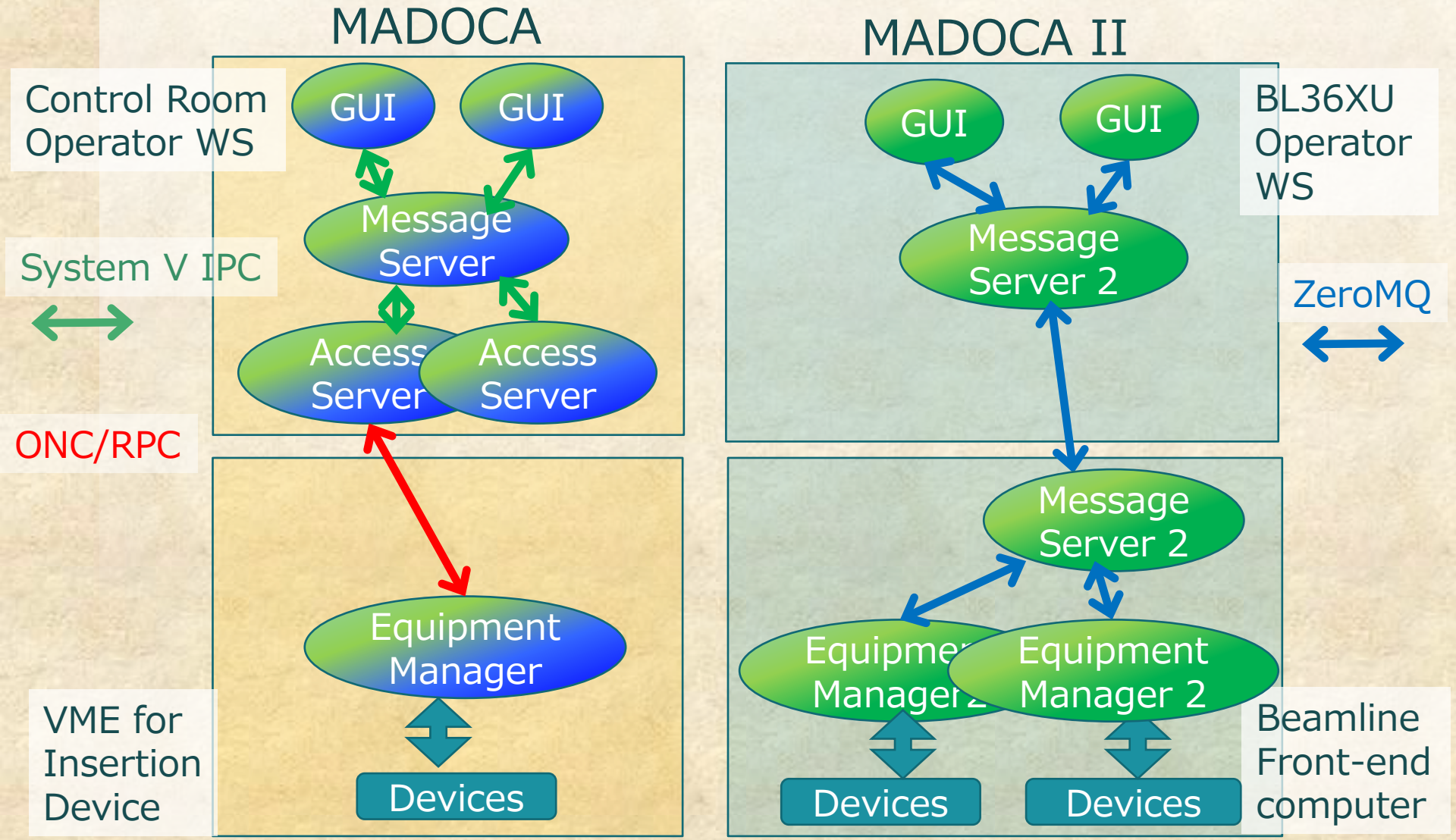
MADOCA II @ BL36XU beamline

Since Sep. 2012



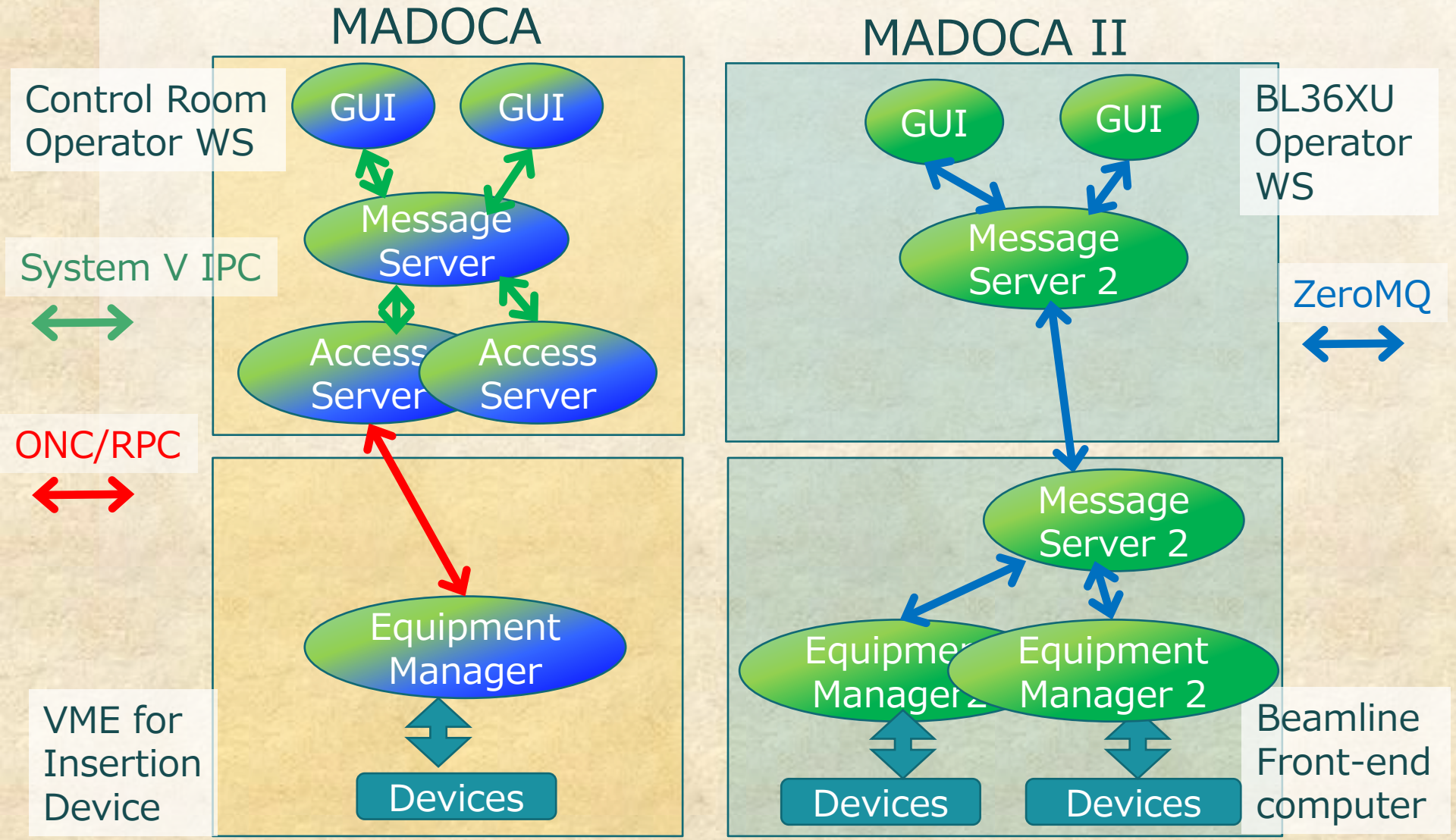
MADOCA II @ BL36XU beamline

Since Sep. 2012



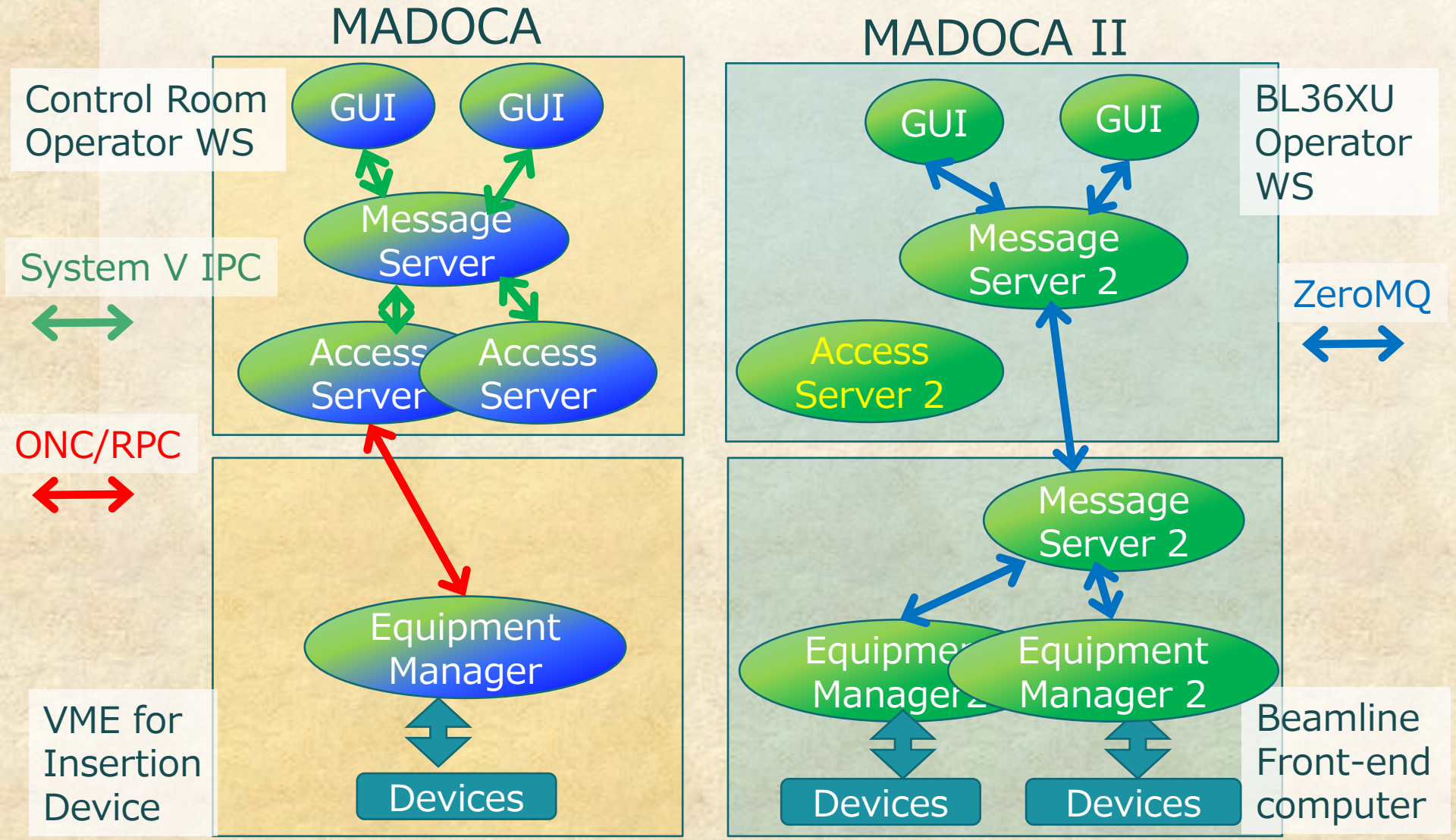
MADOCA II @ BL36XU beamline

Since Sep. 2012



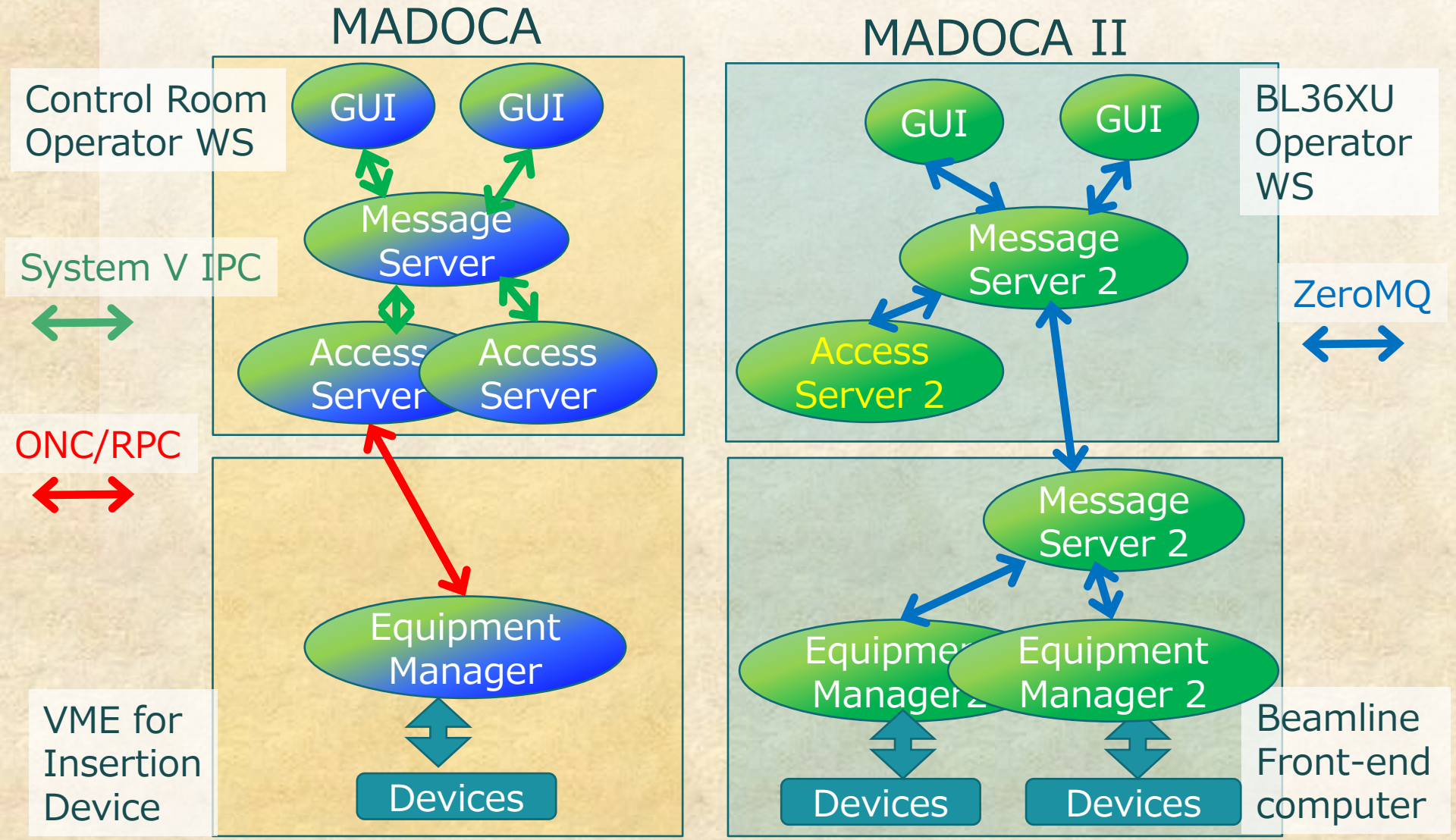
MADOCA II @ BL36XU beamline

Since Sep. 2012



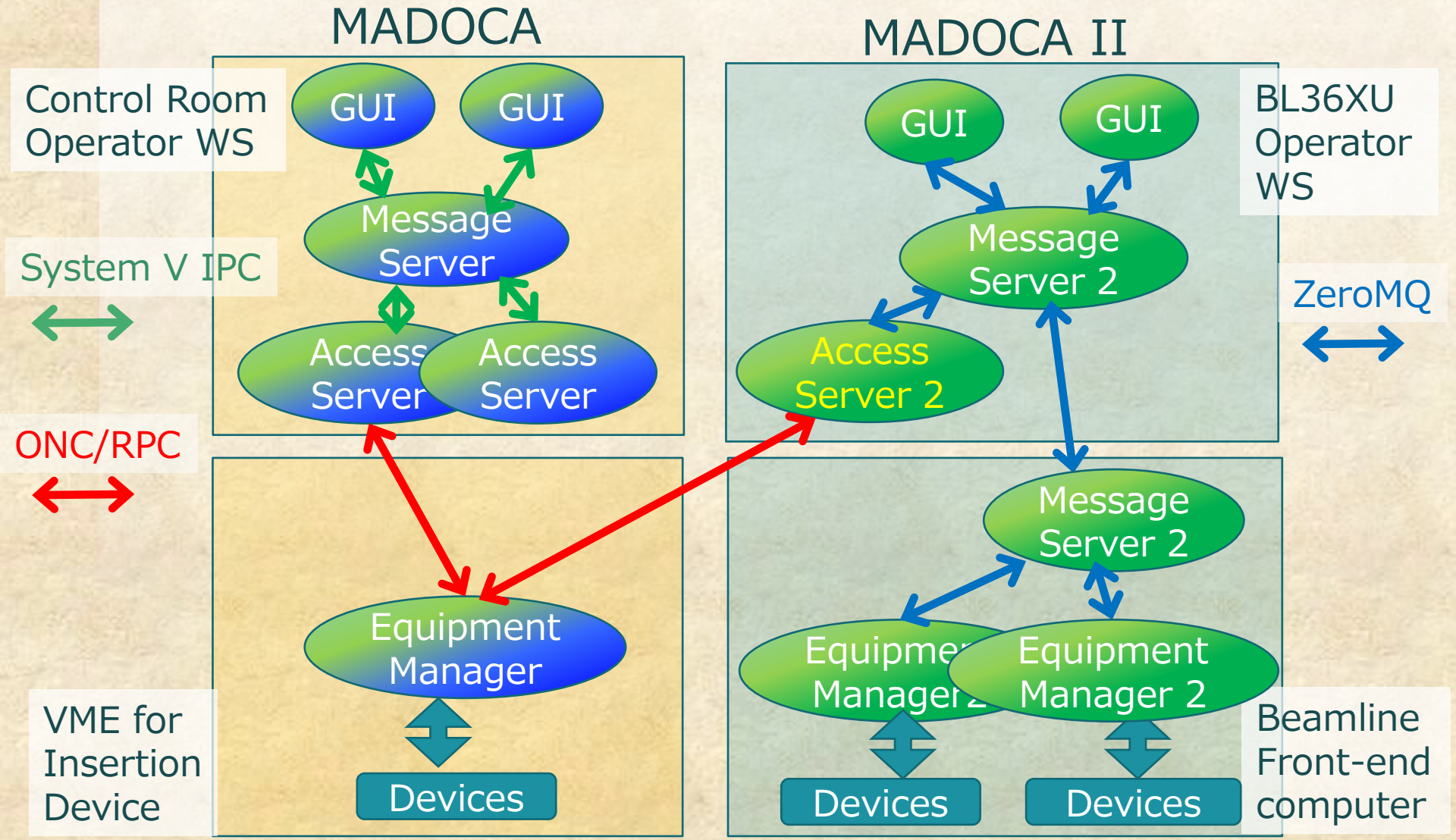
MADOCA II @ BL36XU beamline

Since Sep. 2012



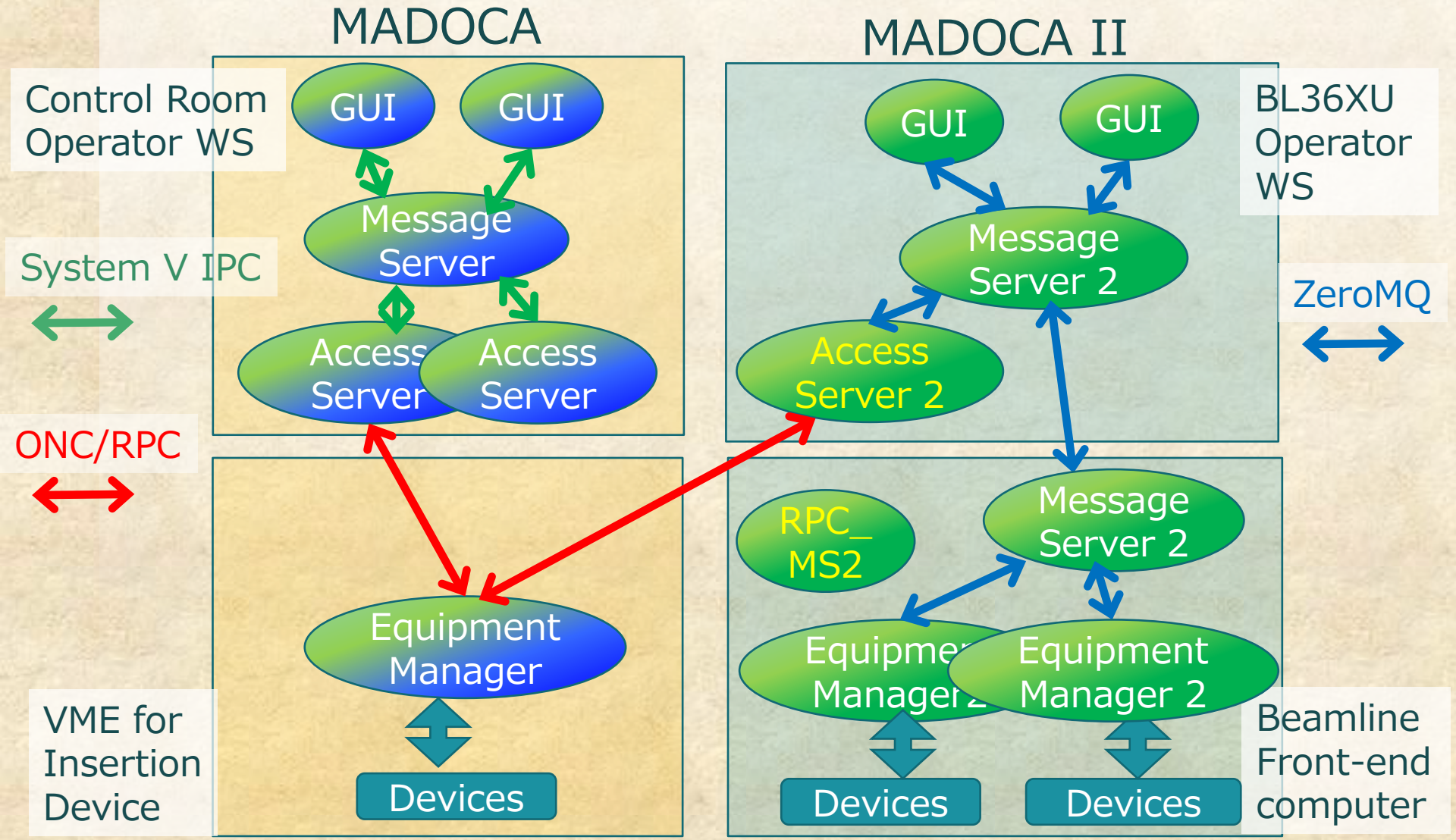
MADOCA II @ BL36XU beamline

Since Sep. 2012



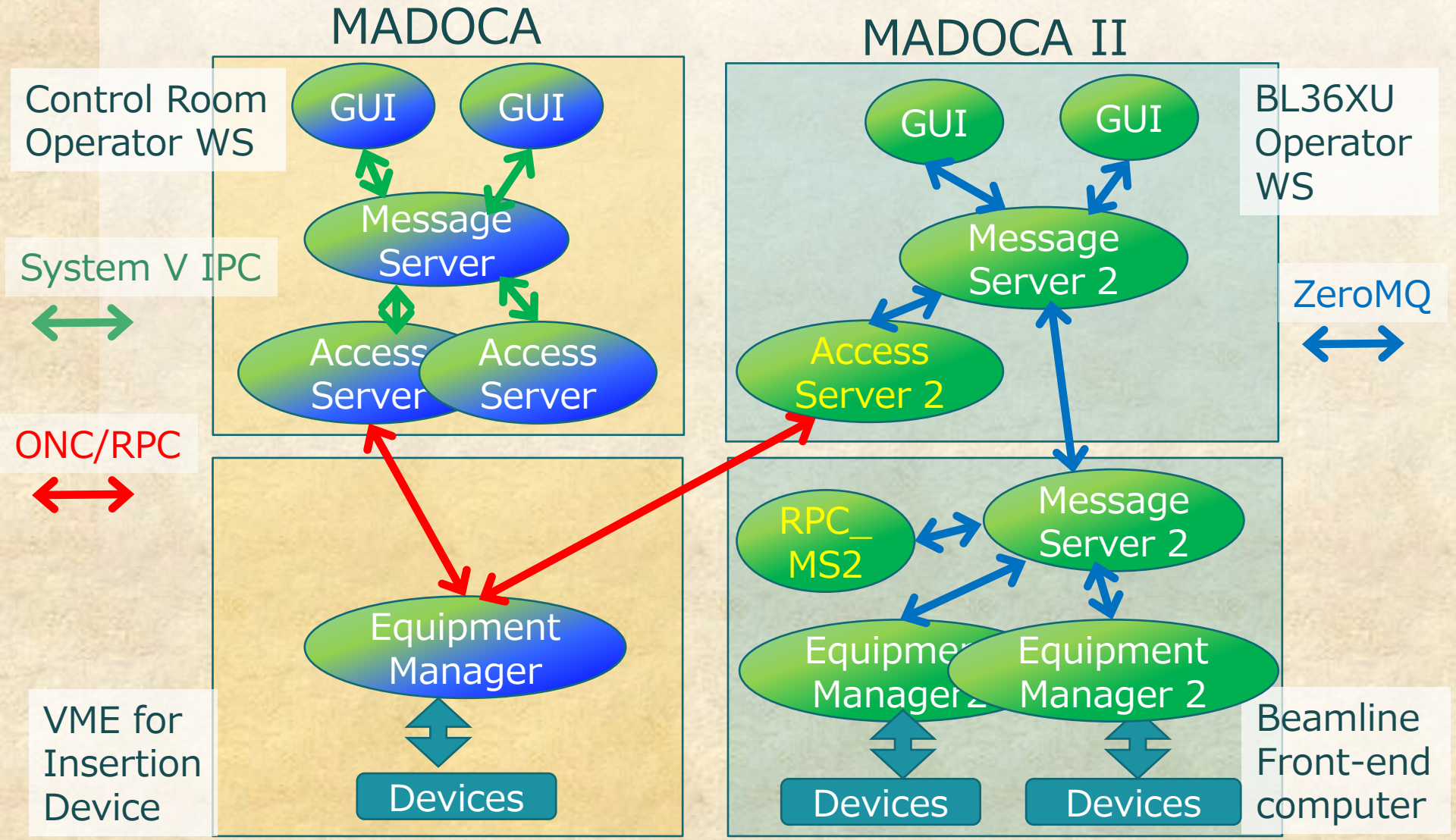
MADOCA II @ BL36XU beamline

Since Sep. 2012



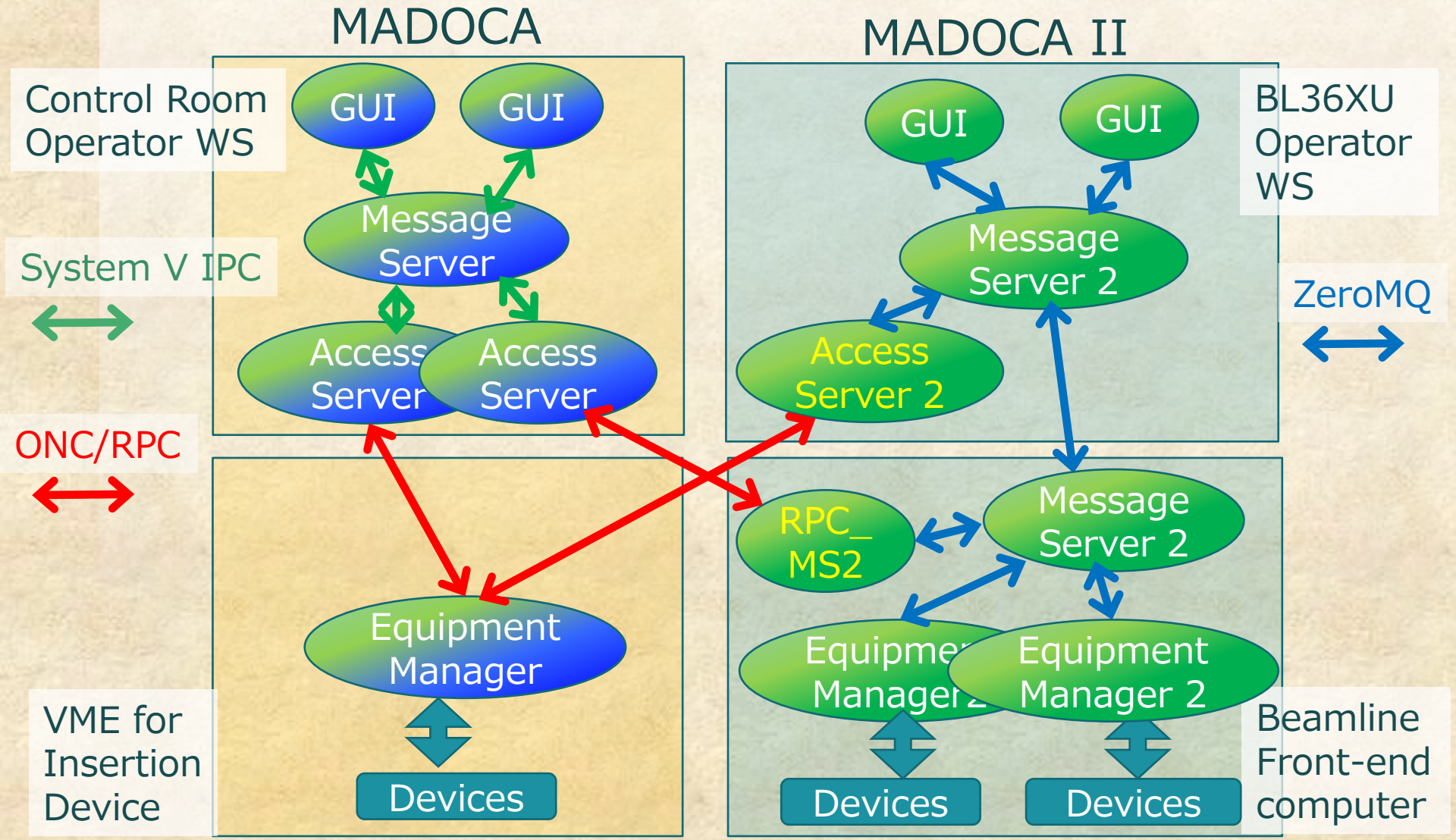
MADOCA II @ BL36XU beamline

Since Sep. 2012



MADOCA II @ BL36XU beamline

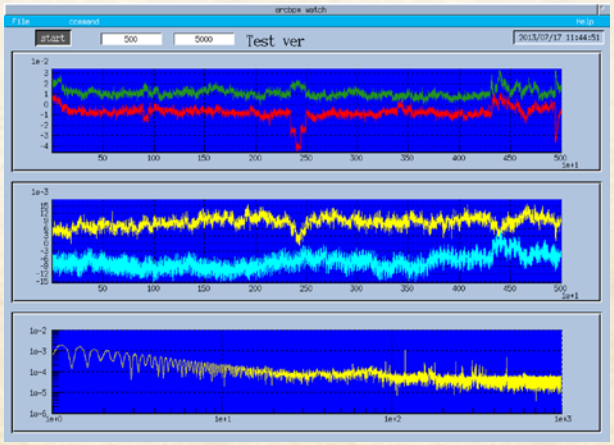
Since Sep. 2012



MADOCA II Applications

- "MADOCA II Interface with LabVIEW"
  - MOPPC129　Y. Furukawa et al.
  - Applied to Beam Position Monitor (BPM)
    - Waveform, 20k points/sec
    - Running on Windows
    - LabVIEW protocol for MADOCA II

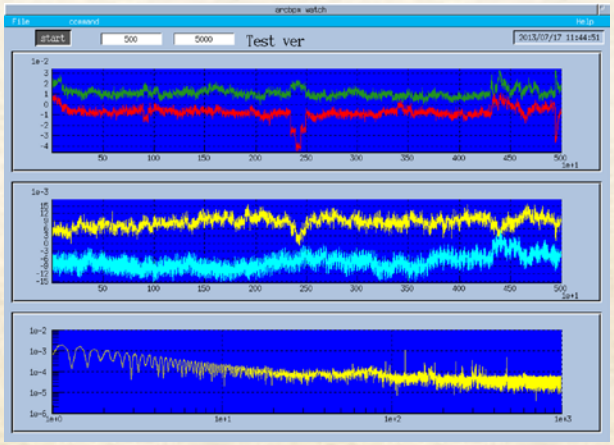
Applied since Sep. 2013



MADOCA II Applications

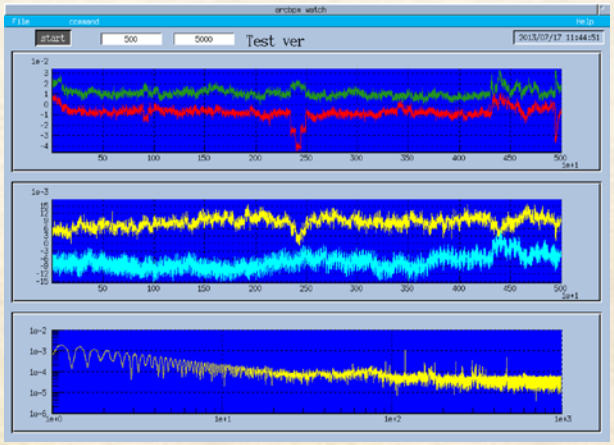
- "MADOCA II Interface with LabVIEW"
  - MOPPC129　Y. Furukawa et al.
  - Applied to Beam Position Monitor (BPM)
    - Waveform, 20k points/sec
    - Running on Windows
    - LabVIEW protocol for MADOCA II

Applied since Sep. 2013



MADOCA II Applications

Applied since Sep. 2013

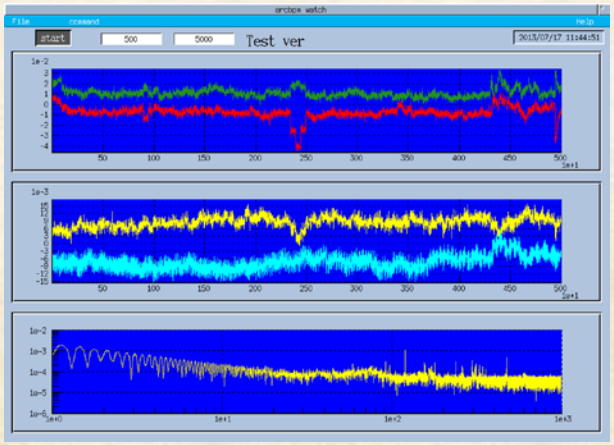


- "MADOCA II Interface with LabVIEW"
  - MOPPC129 Y. Furukawa et al.
  - Applied to Beam Position Monitor (BPM)
    - Waveform, 20k points/sec
    - Running on Windows
    - LabVIEW protocol for MADOCA II

- "Development of MicroTCA-based Image Processing System at SPring-8"
  - TUPPC088 A. Kiyomichi et al.

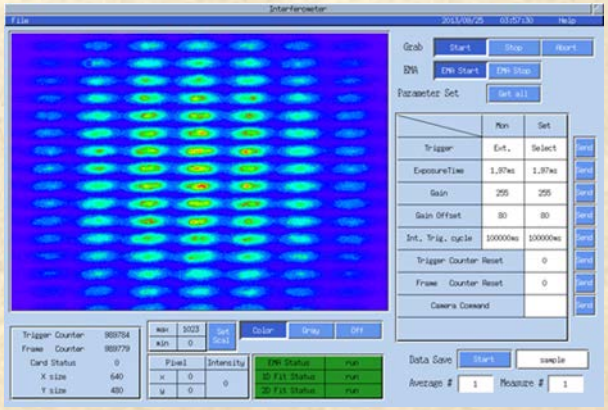
MADOCA II Applications

Applied since Sep. 2013



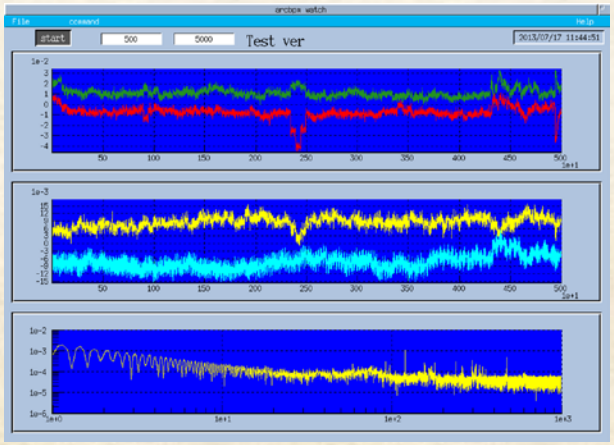
- "MADOCA II Interface with LabVIEW"
  - MOPPC129 Y. Furukawa et al.
  - Applied to Beam Position Monitor (BPM)
    - Waveform, 20k points/sec
    - Running on Windows
    - LabVIEW protocol for MADOCA II

- "Development of MicroTCA-based Image Processing System at SPring-8"
  - TUPPC088 A. Kiyomichi et al.



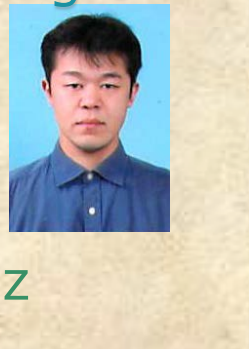
MADOCA II Applications

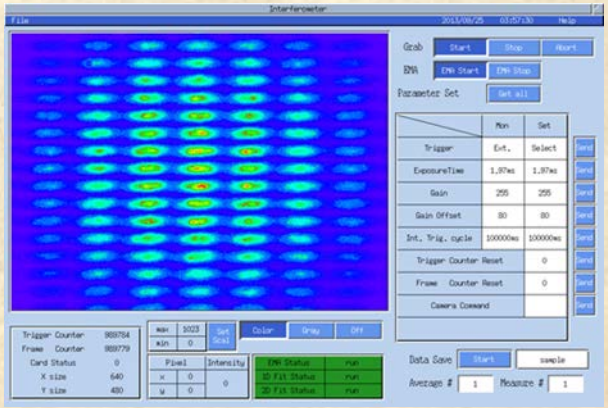
Applied since Sep. 2013



- "MADOCA II Interface with LabVIEW"
  - MOPPC129 Y. Furukawa et al.
  - Applied to Beam Position Monitor (BPM)
    - Waveform, 20k points/sec
    - Running on Windows
    - LabVIEW protocol for MADOCA II

- "Development of MicroTCA-based Image Processing System at SPring-8"
  - TUPPC088 A. Kiyomichi et al.
  - Applied to two-dimensional synchrotron interferometer
    - Image data (VGA), up to 10 Hz





Summary

- New control framework, MADOCA II has been developed
  - Shortcomings in MADOCA are fixed
  - Flexibilities in the Messaging using ZeroMQ/MessagePack
    - Messaging with variable-length data
    - Asynchronous communications
    - Controls on Linux, Solaris and Windows
- Control systems with MADOCA II are implemented
  - Stabilities → confirmed in BL36XU beamline since Sep. 2012
  - BPM and two-dimensional interferometers
  - SACLA experimental station controls

⇨ Since Sep. 2013

- Next step: utilize MADOCA II for various applications
  - Upgrade & Replacement of SPring-8 control systems
  - Apply MADOCA II into other facilities