Status of the Configuration Information System for BM@N online processing
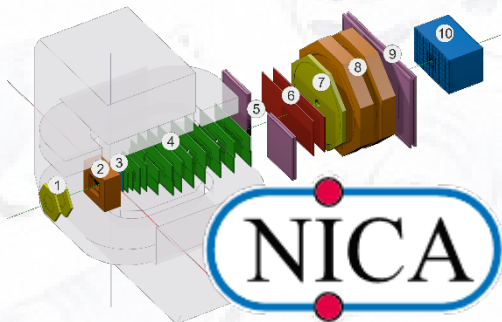
E. Alexandrov[1], *I. Alexandrov*[1], *I.Filozova*[1], *K.Gertsenberger*[1], *D.Pryahina*[1], *G.Shestakova*[1], *A. Yakovlev*[1]

[1]JINR, Dubna

9th Collaboration Meeting of the BM@N Experiment at the NICA Facility, 13-16 September 2022

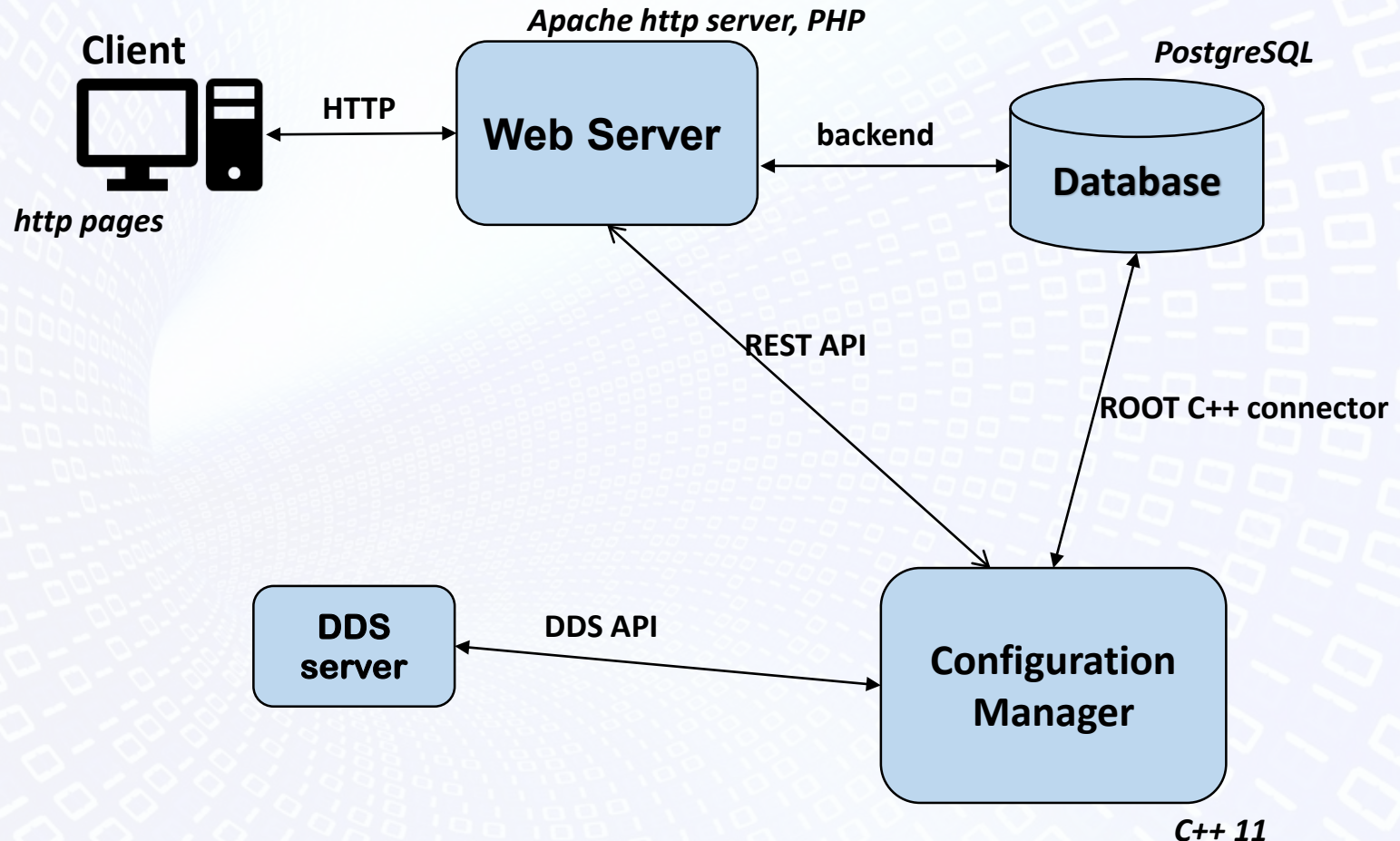**Joint Institute for Nuclear Research**

Outline

- **Goals of the system**
- **General Architecture of the BM@N Configuration System**
- **DB Object model**
- **WEB interface: Configuration Manager view, monitoring, add/tdit task views**
- **Configuration Manager: functionality**
- **Configuration Manager: implementation**
- **Configuration Manager: task states**
- **Configuration Manager: implementation structure**
- **Use of the Dynamic Deployment System (FAIR)**
- **Test environment**
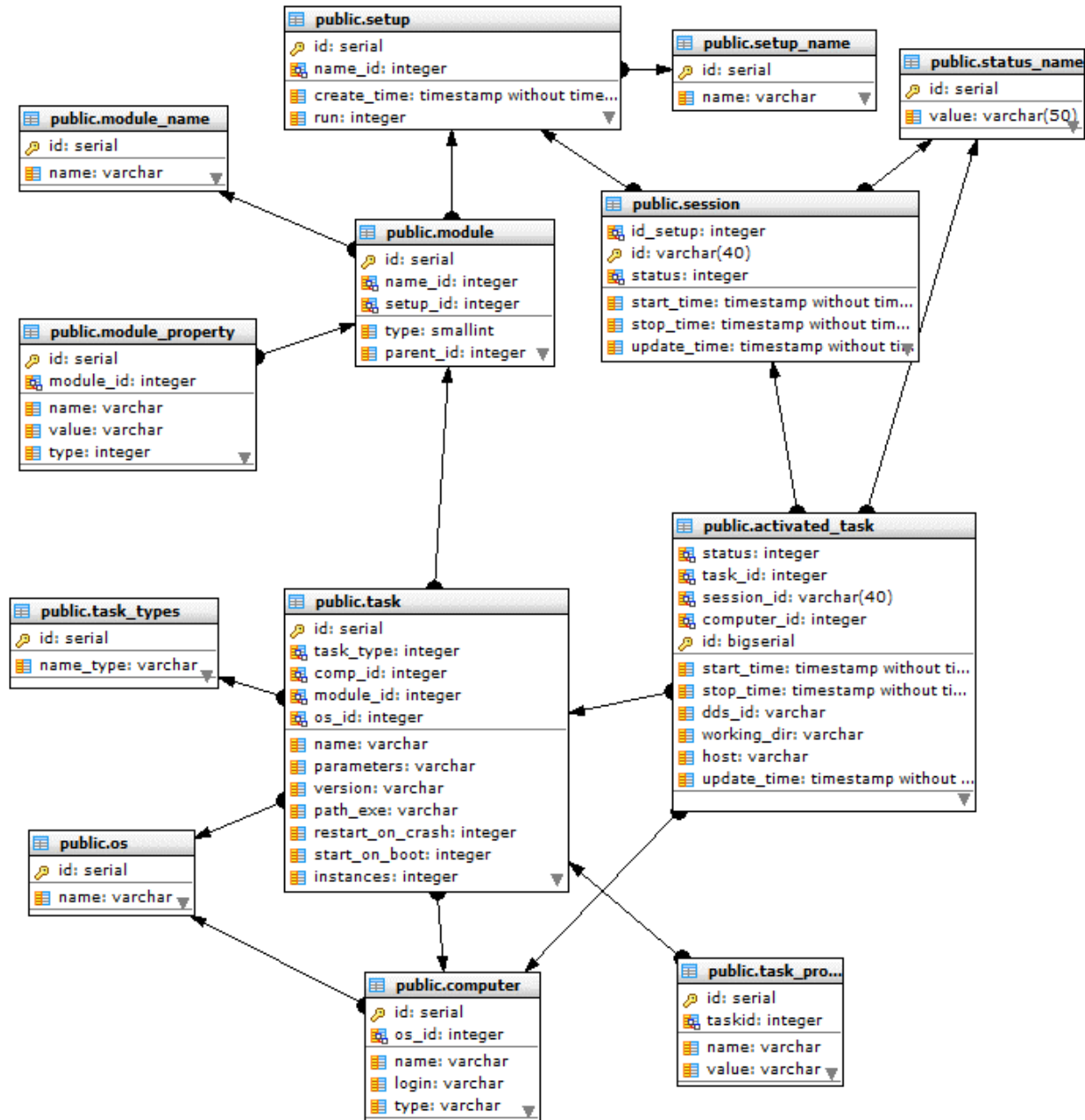- **Status**
- **Next steps**

Goals of system

- **store and provide configuration data for online processing :**
  - **set of various detectors configuration parameters (working voltage etc.)**
  - **sequences of software tasks with their dependencies**
    - **online raw data digitization**
    - **online histogramming**
    - **fast event reconstruction**
    - **event monitor**
  - **setup and tasks dependencies**
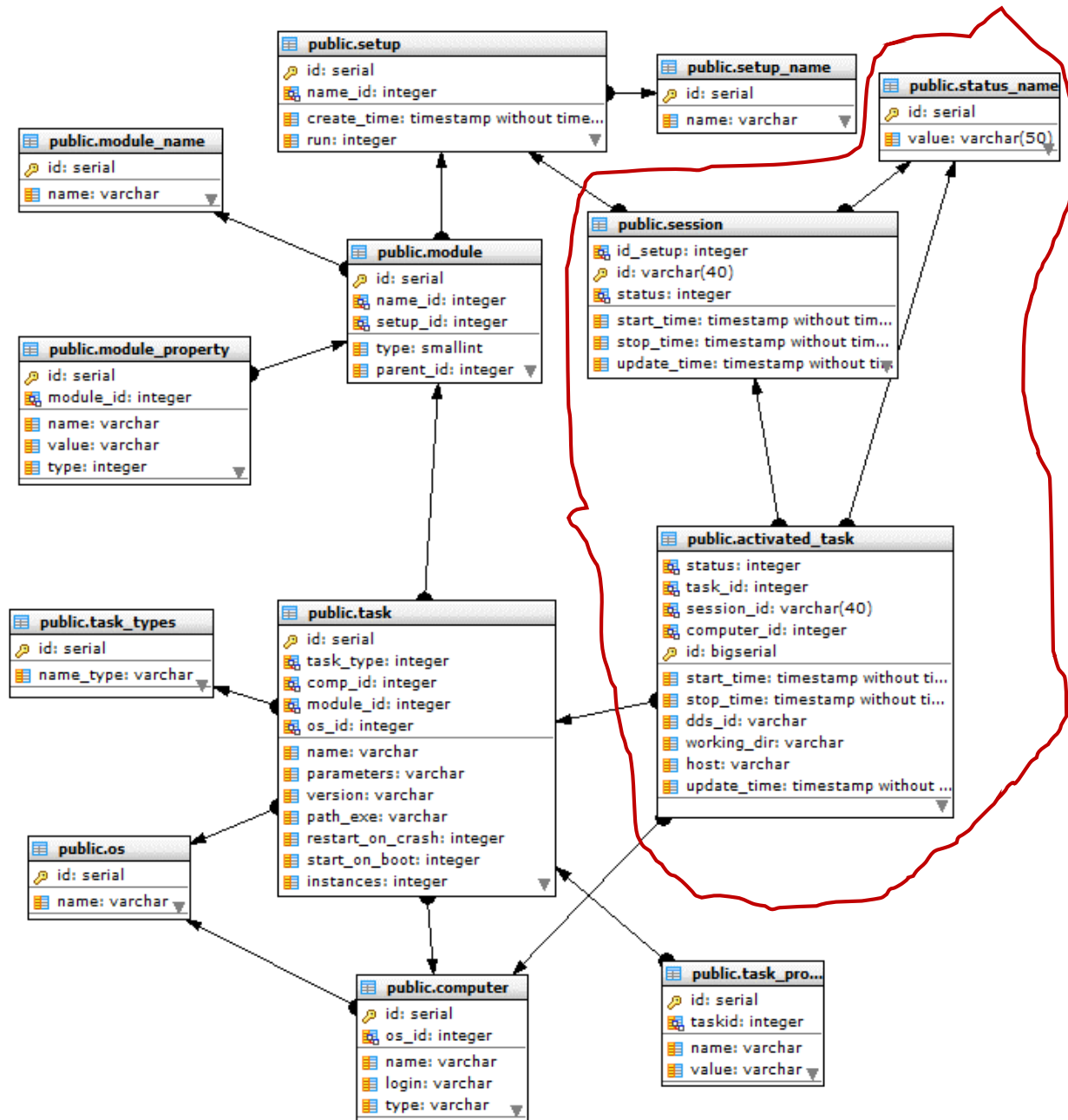- **should to be able to start, stop and monitor setup tasks during experiment sessions**

General Architecture of the BM@N Configuration System

Client

http pages

HTTP

Apache http server, PHP

Web Server

backend

PostgreSQL

Database

REST API

ROOT C++ connector

DDS server

DDS API

Configuration Manager

C++ 11

DB Object model

DB Object model

Web-interface. Configuration Manager view

Web-interface. Monitoring view

Web-interface. Monitoring view (2)

Web-interface. Add task view

Web-interface. Task edit mode

Configuration Manager functionality

- listens and performs commands (REST API in use)
    - Start all setup tasks
    - Stop all setup tasks
    - Update setup
    - Get log of the activated task
    - Check if configuration manager is alive
    - Stop configuration manager
- reads from DB setup data using setup ID
- prepares tasks for start using DDS system (convert data into DDS topology form)
- starts all tasks of topology using DDS
- gets from DDS server info about all started tasks
- stores in the DB the info about DDS session and tasks

Configuration Manager implementation

- **class ConfigSupport**
  - **performs operations that connected with database**
    - loads from DB the setup by setup id and keeps these data in memory during manager run
    - writes into DB all dynamic data concerning DDS sessions and activated tasks
    - converts DB setup data into DDS topology

- **class DDSCommunicator**
  - **performs all operations that connected with DDS facilities**
    - creats, starts, updates and stops DDS session.
    - gets info about DDS tasks using callbacks
    - gets task log stream

- **class RestService**
  - **Performs all operations that connected with REST API**
    - listens socket, gets and parses requests, performs requests using ConfigSupport and DDSComunicator

Configuration Manager: states

- **States of the setup and its tasks**
  - **Starting**
    - Configuration Manager (ConfigMan) got command to start setup, activated the command and have sent response to sender
  - **Started**
    - ConfigMan got response from DDS server that task was started and updated database with this info.
  - **Stopping**
    - ConfigMan got command to stop setup, activated this operation and have sent response to sender
  - **Stopped**
    - ConfigMan got response from DDS server that task is stopped and updated databse with this info

Test environment

Configuration manager uses test_topology.xml and starts:

1

root_digi_imit

Sends message

onlineHisto_imit

2

Sends message and get reply

fastEventReco_imit

3

Sends message

eventDisplay
_imit

4
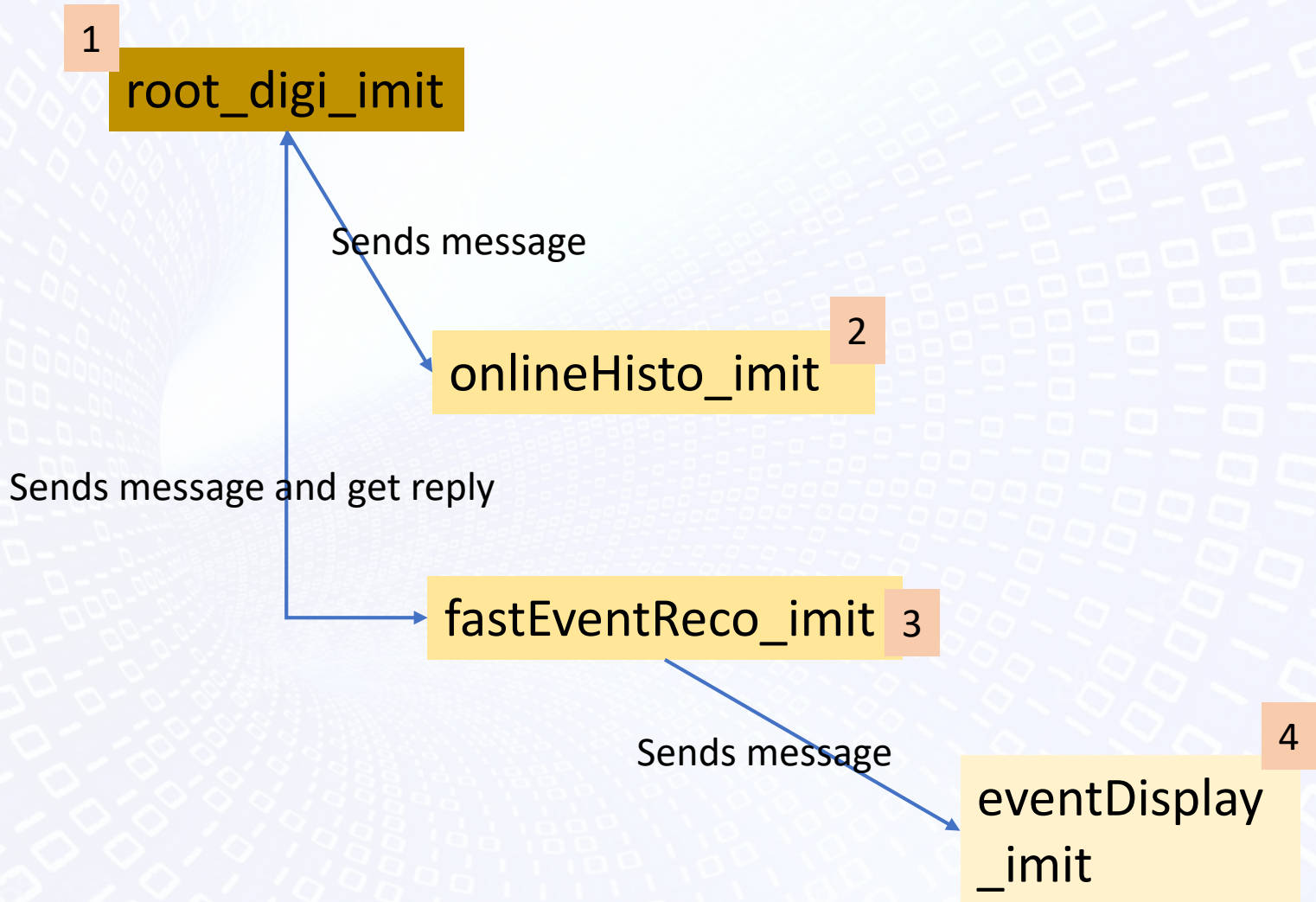
Use the Dynamic Deployment System for tasks management

- **DDS: tool-set that automates and significantly simplifies a deployment of user defined processes (tasks) and their dependencies**

- **DDS: deploys agents to execute user tasks**

- **DDS agent:**
  - **supports multiple tasks slots**
  - **is able to run and watchdog multiple tasks simultaneously**

- **Some of DDS features did not work as described**
  - **used less convenient methods**
    - **SSlotInfoRequest instead of SOnTaskDoneRequest**

16

Status

- **Design is developed**
- **Beta version of system with all requested functionality is implemented and put in Gitlab**
  - **Database in PostgreSQL: done**
  - **WEB interface: done**
  - **Configuration Manager (C++ library and config_manager executable): done**
- **Test environment is used for tests**

Next steps

- **Web interface improvement**
  - **Add tab for DDS sessions**
  - **Improve Control panel of Configuration Manager view on the base of test results (control panel states)**

- **Produce intensive tests of full system**

- **Make updates on the base of the test results for all three components (DB, Web interface and Configuration manager)**

- **Add real tasks of online processing and use it in tests**

- **Try to optimize interactions with DDS**