

# Архитектура Mathbrain.ru – облачного решения для анализа данных магнитной энцефалографии

Рыкунов С.Д., Оплачко Е.С., Устинин М.Н.

ИМПБ РАН – филиал ИПМ им. М.В. Келдыша РАН

## Задача:

- Создание простого и интуитивного инструмента, который позволит специалистам в области медицины и биологии обрабатывать и анализировать данные энцефалографии

## Требования к ресурсу:

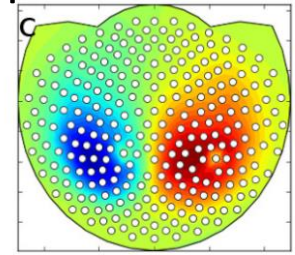
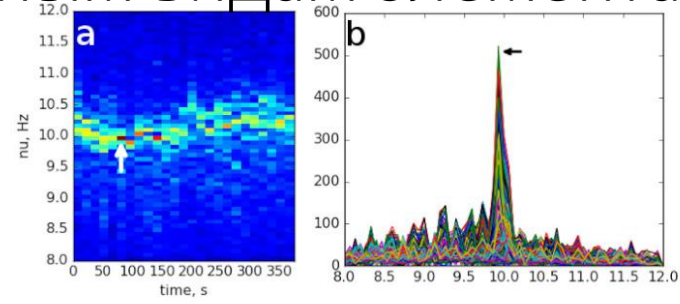
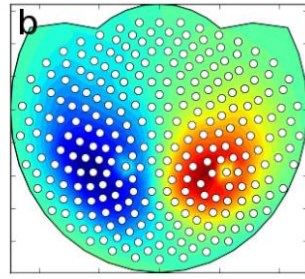
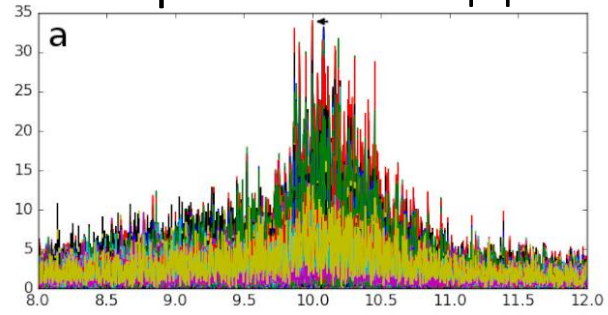
1. Инструмент должен работать без установки дополнительного ПО и не требовать в использовании навыков программирования;
2. Отсутствие дополнительного лицензирования для пользователя
3. Все используемые языки программирования и библиотеки должны находиться в открытом доступе и свободно распространяться
4. При реализации сервиса должна быть предусмотрена возможность обработки большого объема данных
5. Система должна быть многопользовательской
6. Удаленная обработка данных, без использования локальных ресурсов (данные снимаются и обрабатываются в территориально удаленных местах)
7. Ресурс должен предоставлять возможность добавления дополнительных методов обработки данных

# Предоставляемые методы обработки данных

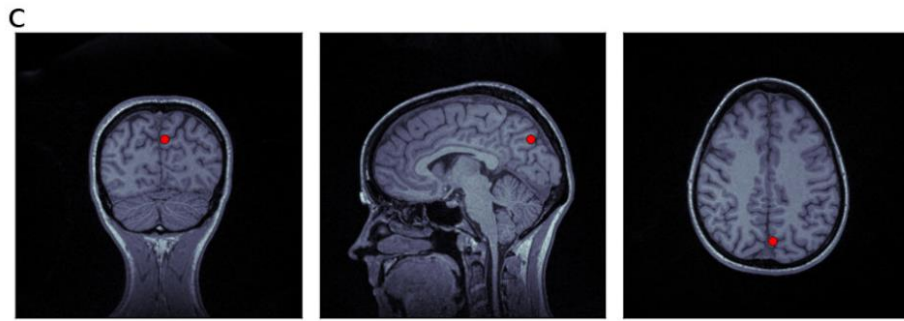
- Прямое и обратное преобразования Фурье временных рядов
- Анализ методом главных компонент (PCA, principal component analysis)
- Анализ независимых компонент (ICA, independent component analysis)
- Количественный анализ энцефалограмм – преобразование Фурье скользящим окном
- Решение обратной задачи магнитной энцефалографии в однодипольной модели

# Решение обратной задачи по различным видам элементарных компонент

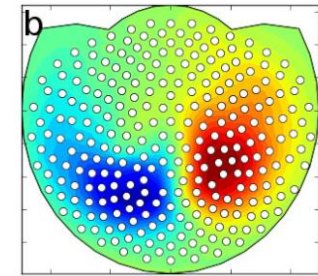
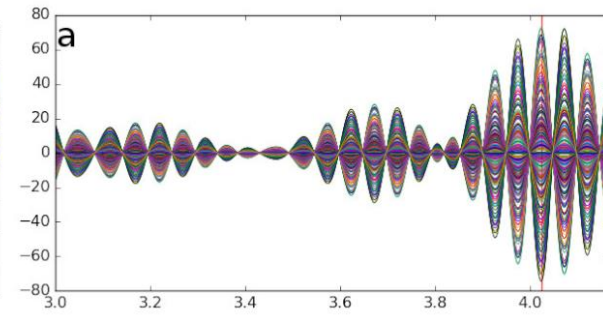
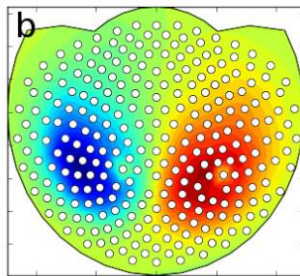
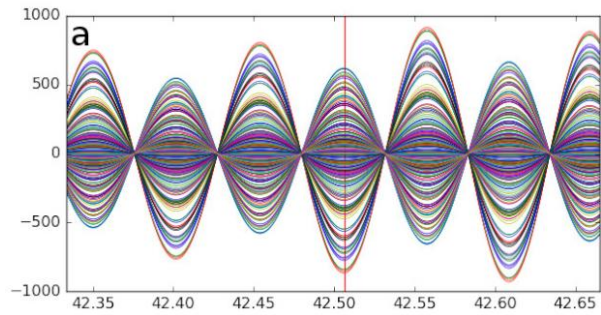
Преобразование Фурье



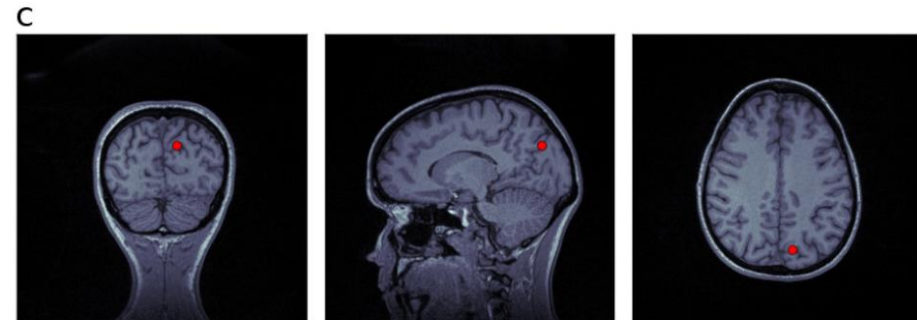
Количественный анализ



PCA



ICA



# Используемые технологии

- Docker Swarm
- MiniO
- Python, Numpy, Scipy, pyFFTW, matplotlib, mpld3
- MariaDB
- Nginx, PHP, JQuery

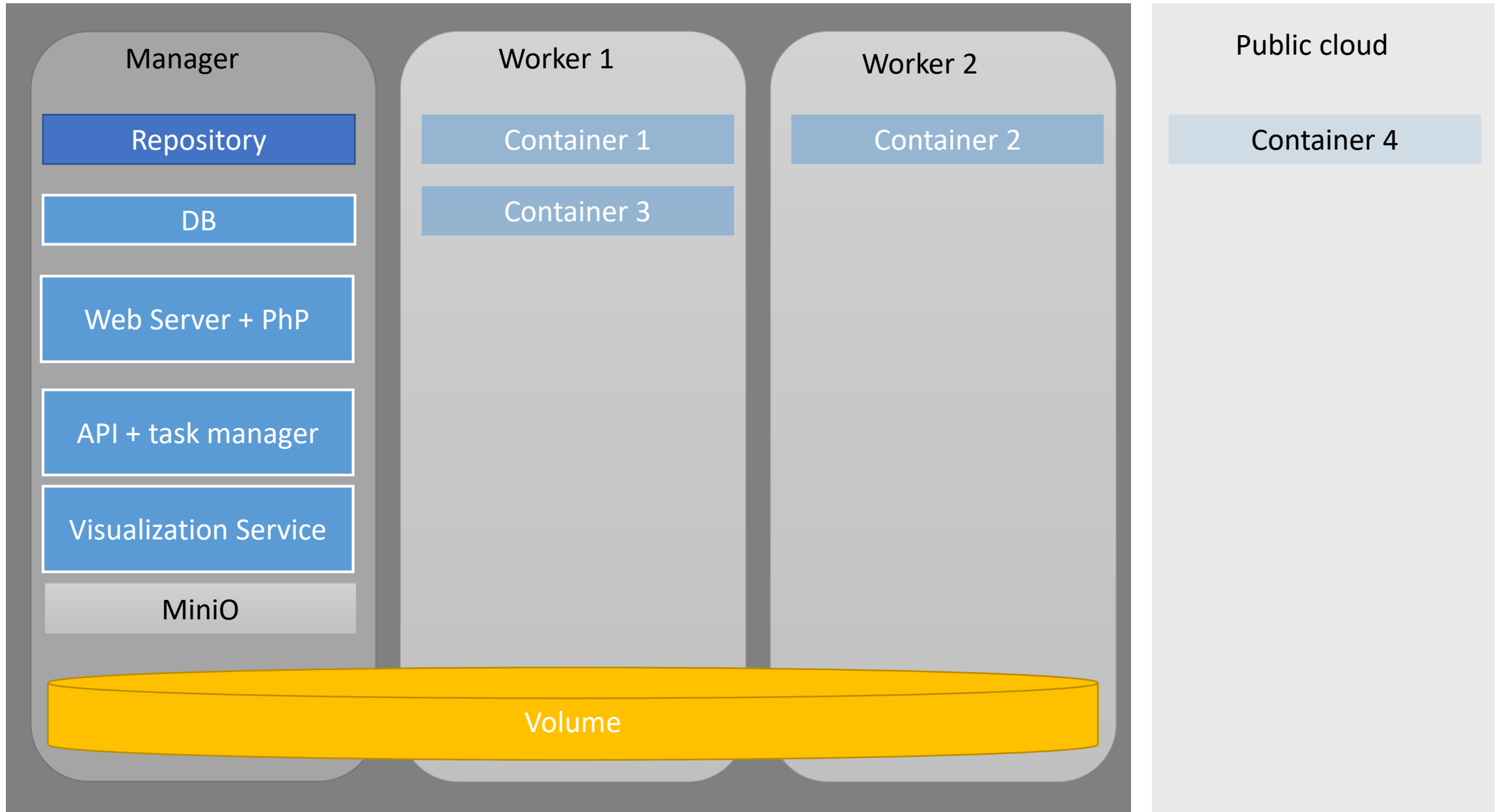
## Формат данных

- Matlab 7.3 mat ( HDF5 based)
- Описание измерительных приборов в формате FieldTrip

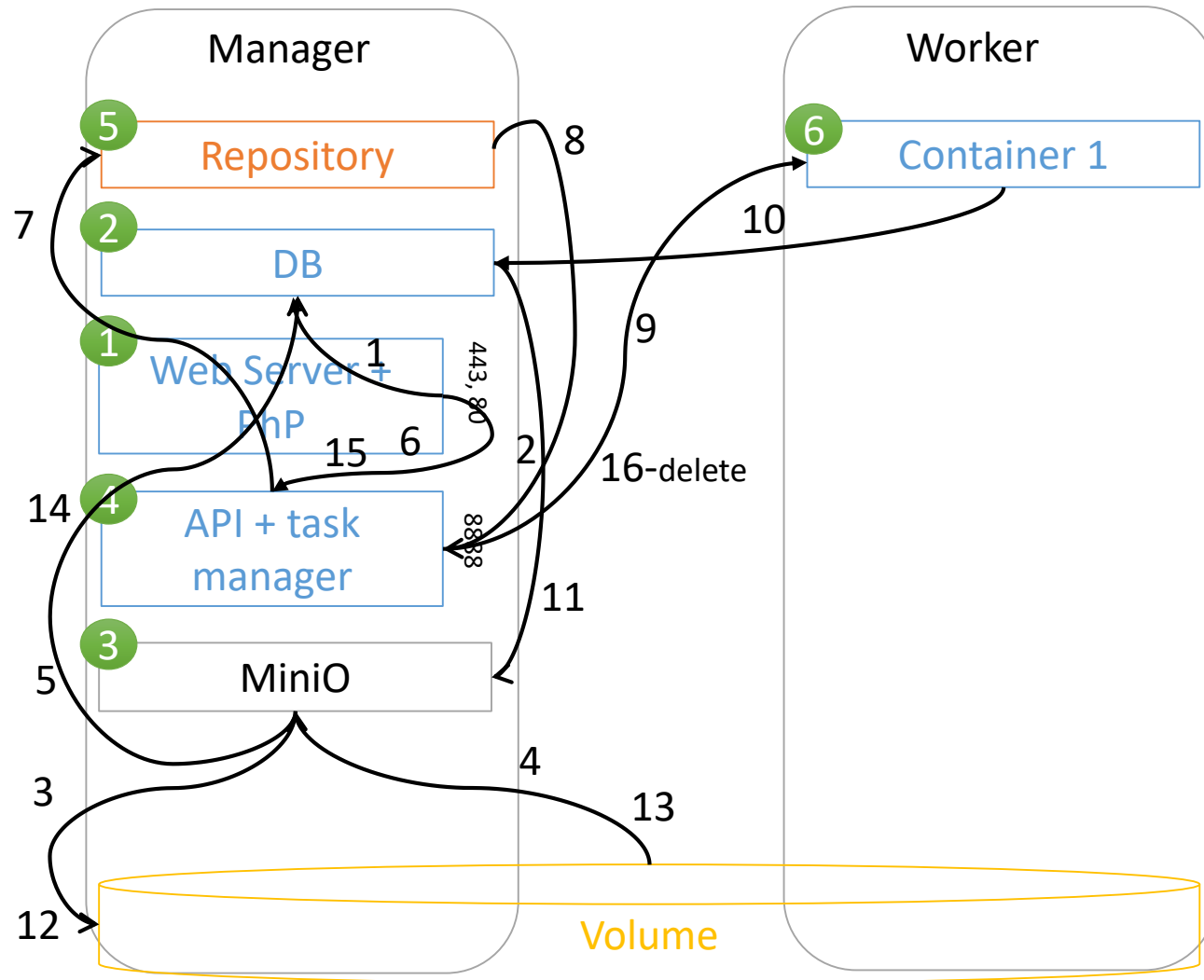
# Почему Docker Swarm?

- Лёгкое масштабирование
- Возможность масштабирования с привлечением внешних публичных облачных решений (Microsoft Azure, Amazon EC2, Google Compute Engine)
- Простота в управлении созданием/запуском/остановкой контейнеров
- Встроенные механизмы распределения вычислительных ресурсов
- Изоляция данных/вычислительных программ

# Архитектура ресурса MathBrain



# Схема обработки данных



- 1 Web сервер получает задачу на выполнение метода
- 2 Из БД получается информация об исходном файле для обработки
- 3 Т.к. БД хранит данные локально, любое обращение идет через MiniO
- 4 Данные передаются в контейнер с API, откуда инициируется создание контейнера
- 5 Из репозитория создается контейнер на свободной ноде (Worker)
- 6 Выполняется вычислительная задача, данные сохраняются на подключенном HDD, контейнер удаляется





Работа выполнена при поддержке Программы Президиума РАН №27 «Фундаментальные проблемы решения сложных практических задач с помощью суперкомпьютеров» и грантов РФФИ 17-29-02178, 18-29-02087, 16-07-00937, 16-07-00677, 17-07-00686.