

# Фреймверк для распределения задач с использованием сетей функциональных вычислений

Кристалев Данил

дипломник группы ПИ-21

руководитель: к.ф-м.н., доцент каф ПМ С.М.Старолетов

# Актуальность

- ОТЗЫВЧИВОСТЬ
- Отказоустойчивость
- Масштабируемость



# Акторная модель

- Актор
- Асинхронные передачи сообщений:
  - Слабая связность
  - Изоляция
  - Прозрачность местоположения
  - Делегирование ошибки в виде сообщений



## Цель

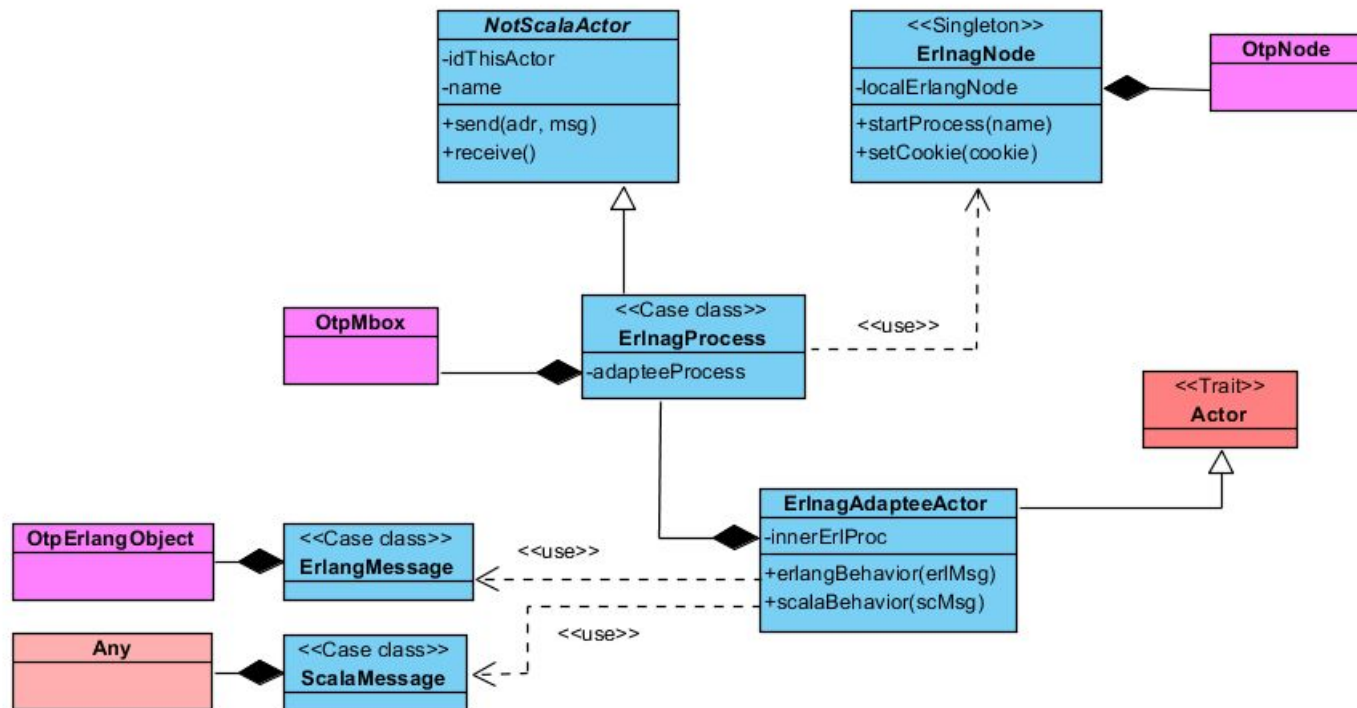
Проектирование фреймворка для написания систем распределения задач на основе акторной модели и реализация систем, демонстрирующей работу фреймворка.

# Выбранные языки

- Erlang:
  - + Леговесные потоки
  - Производительность
- Scala
  - + JVM
  - Тяжеловесные акторы



# Диаграмма классов фреймверка



## Сценарии использования

- Построение систем, основанных на взаимодействии большого количества акторов
- Обработка большого потока данных
- Проектирование многопользовательских игр
- И тд.

# Рассмотренные задачи

- Файловая система на процессах
- Поиск минимального значения в непрерывном потоке данных



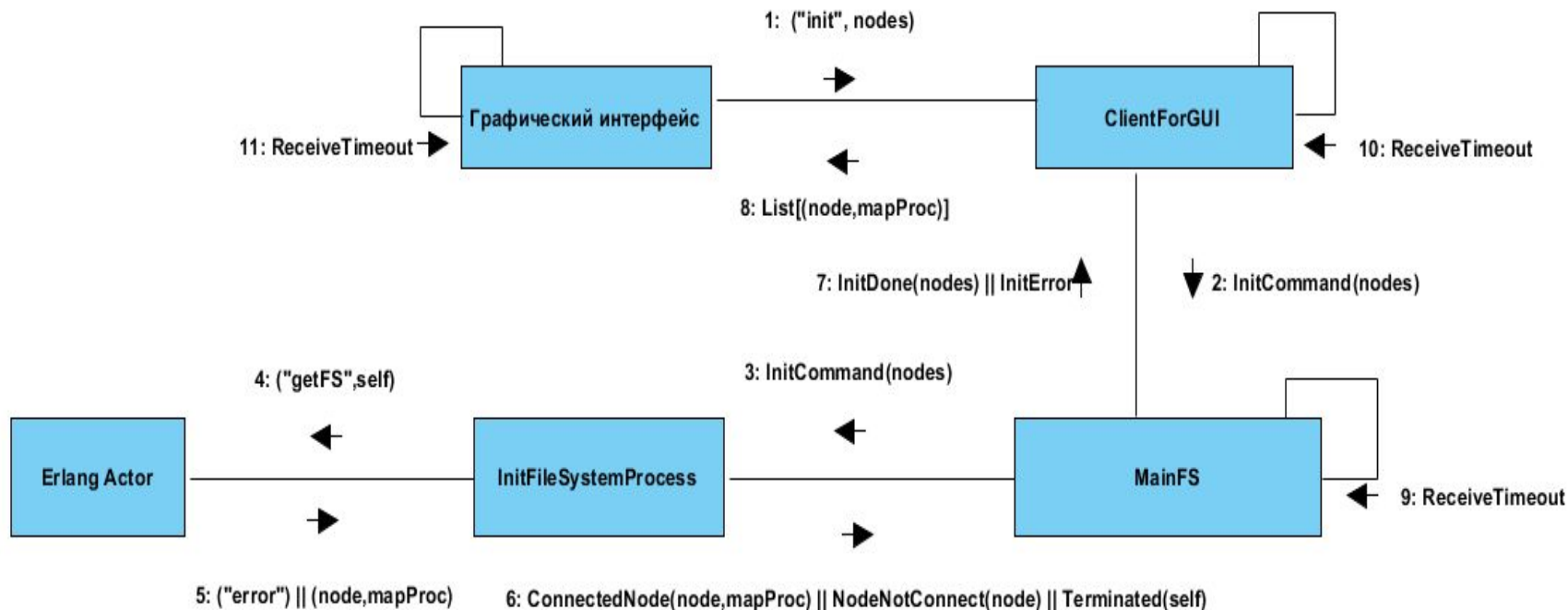
## Система основанная на взаимодействии (задача)

- Имеется множество файлов на диске и такое же количество процессов, запущенных параллельно
- Процессу с именем файла можно можно давать команды, например, delete, copy и rename.

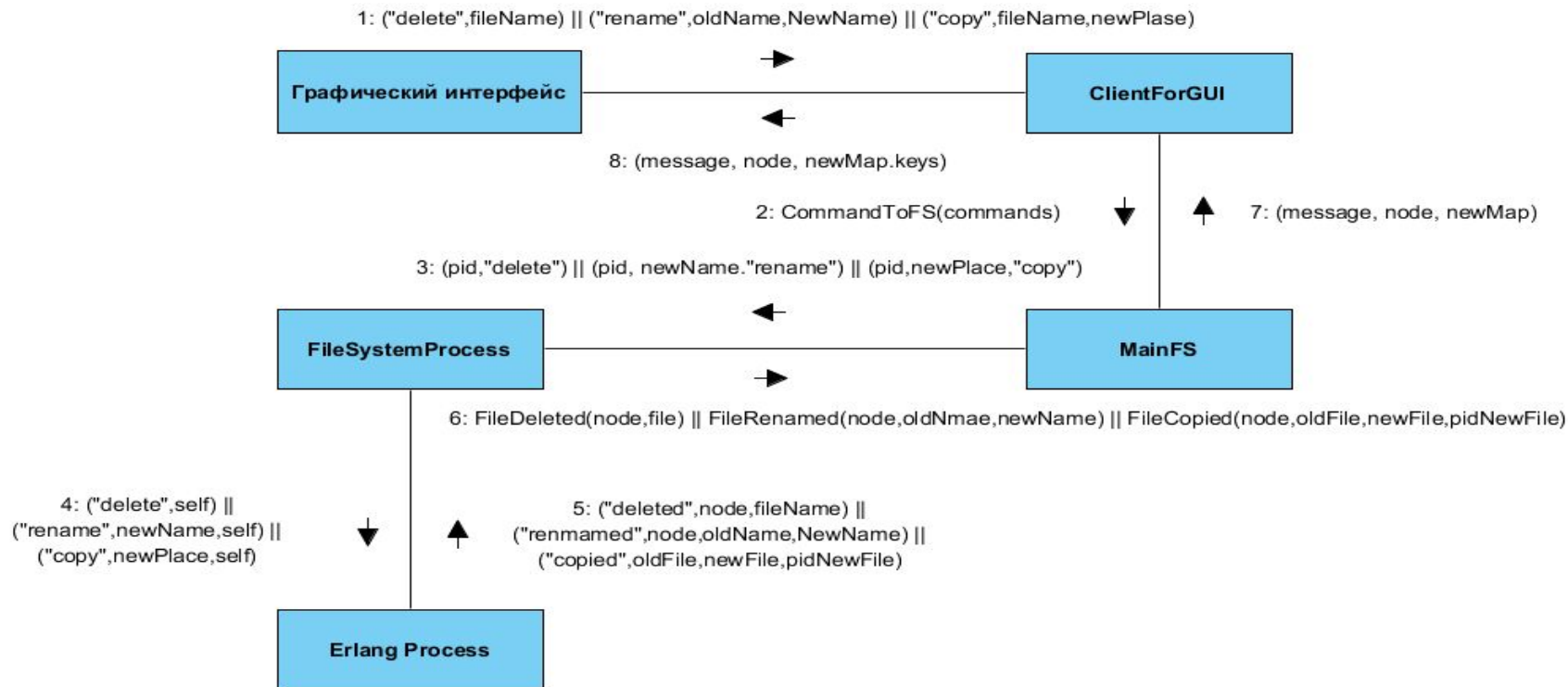
# Анализ задачи

- Файлов много -> много процессов -> процессы должны быть легковесны -> Erlang.
- К такой системе хочется иметь централизованный доступ, который бы мог управлять ФС на нескольких узлах сразу, с удобным интерфейсом -> Scala.

# Инициализация файловой системы



# Отправка команд к файловой системе



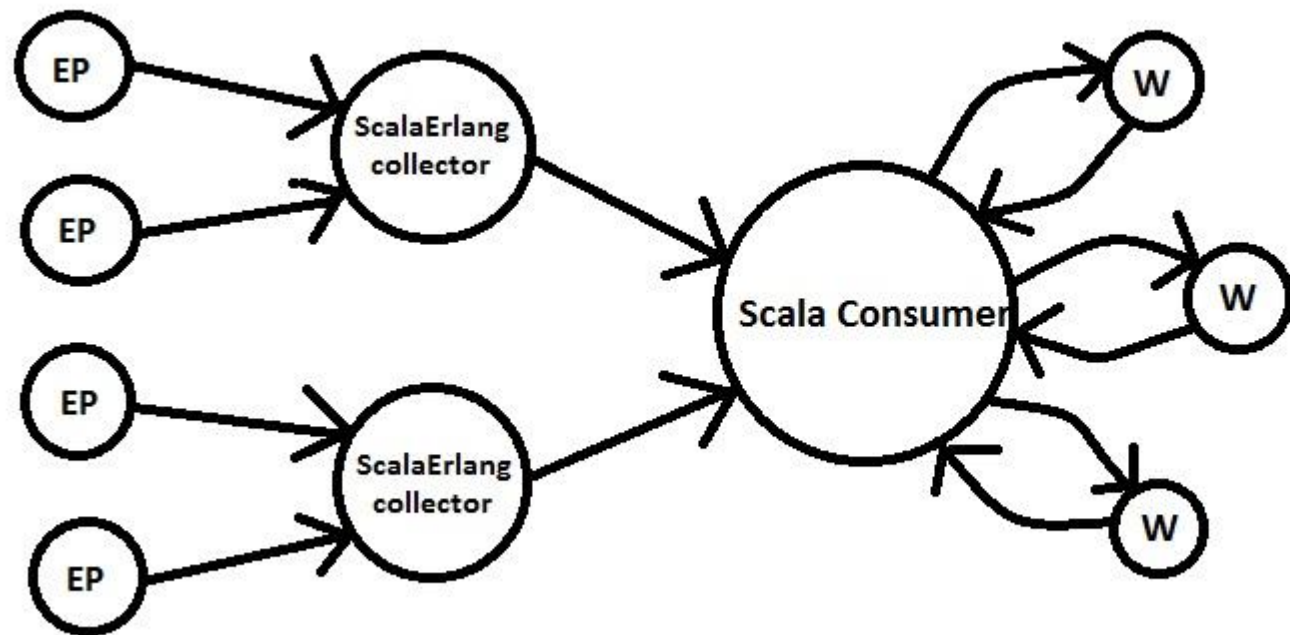
# Тестирование системы

|    |  |       |
|----|--|-------|
| OK | send back messages InitDone              | 374ms |
| OK | send back message FileDeleted            | 57ms  |
| OK | send back message FileCopied             | 11ms  |
| OK | send back message FileRenamed            | 8ms   |
| OK | send back message about complete actions | 32ms  |

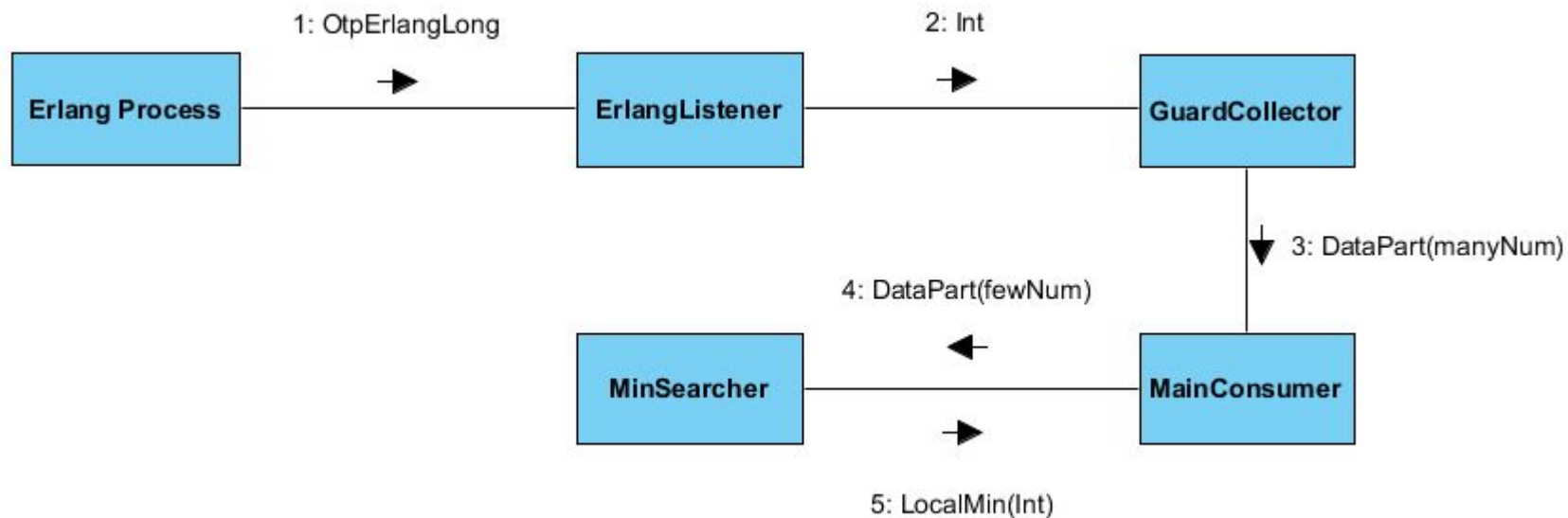
# Потоковая обработка данных (задача)

- Есть большое число источников данных ( $\gg 10000$ ), каждый из которых в некоторый промежуток времени, генерирует некоторое число
- Необходимо найти минимальное число за все время работы системы и минимальное число в последней порции полученных данных

# Модель работы системы



# Работа системы





## Итоги

- Спроектирован и реализован фреймверк, который позволяет совместно использовать акторные системы языков Scala и Erlang.
- Были рассмотрены сценарии использования фреймверка, на примере конкретно решенных задач.
- Работа была представлена на конференции “Наука и молодежь - 2016”

# Доклад окончен.

Ваши вопросы.