

Бронислав Житников



# **NIFI: Пишем код для codeless - системы**

# О себе

Бронислав Житников

- Positive Technologies
- Более 20 лет в ИТ
- От внедрения до разработки
- 8 лет в инженерии данных
- Сообщество Powerfull NiFi



# О чем расскажу

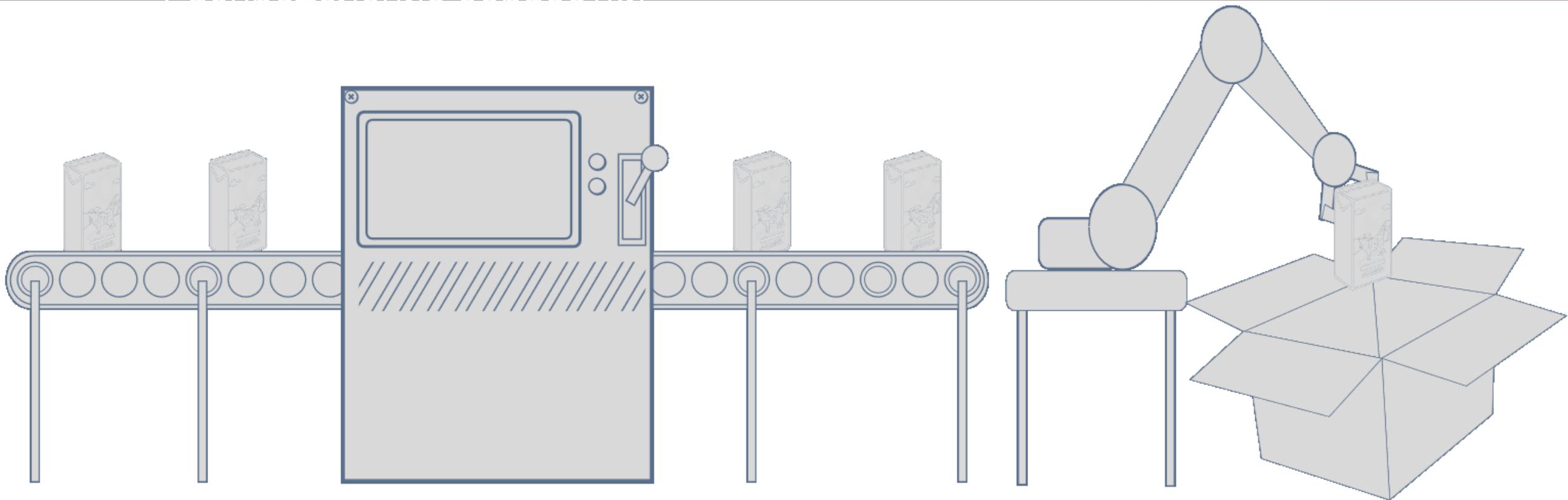
- Про NIFI
- Как мы приходим к коду
- Из чего NIFI состоит
- Как создать Processor
- Что не стоит делать



# Προ ΝΙΦΙ

# Что такое NIFI

- Завод по переработке данных
- Statefull конвейеры
- Распределение обработки



# Как выглядит со стороны

- Граф процесса
- Вершины – процессоры обработки
- Ребра – очереди конвейера
- Разработка рисованием
- И немного Ctrl-C + Ctrl-V



# 02

pt

# Зачем здесь Код

# Так зачем здесь код?

- Не хватает процессоров
- Оптимизация процесса
- Можем сделать правильнее
- Исправить ошибки



# Как мы приходим к Коду

## ExecuteStreamCode

Внешний скрипт или приложение

## ScriptedService

Меняем логики интеграций

## Code

Вы потеряны

1 шаг

2 шаг

3 Шаг

4 шаг

Конец

## ExecuteScript

Используем встроенные возможности для скриптов

## ScriptedProcessor

Используем все возможности API

# Что меня привело на этот Путь



# Что меня привело на этот Путь



# 03

pt

## Как устроен NIFI

# А ЧТО ПОД КАПОТОМ

Java

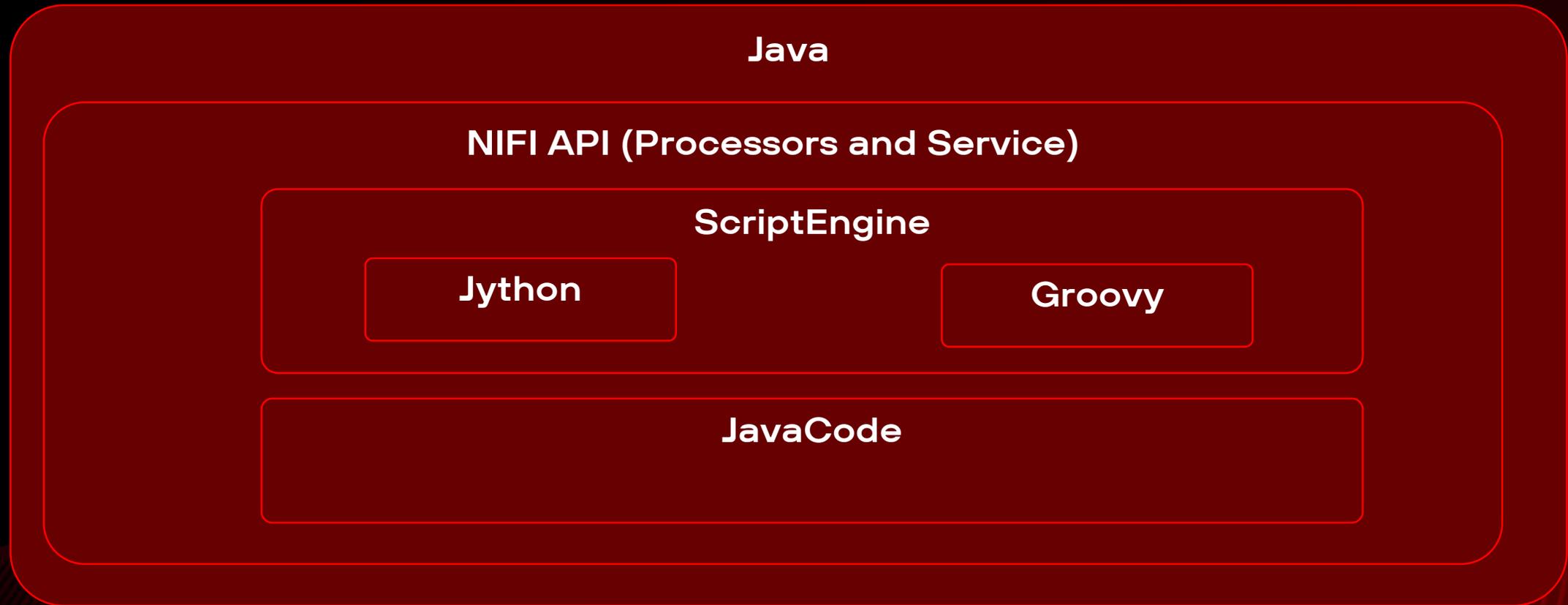
Maven

NIFI

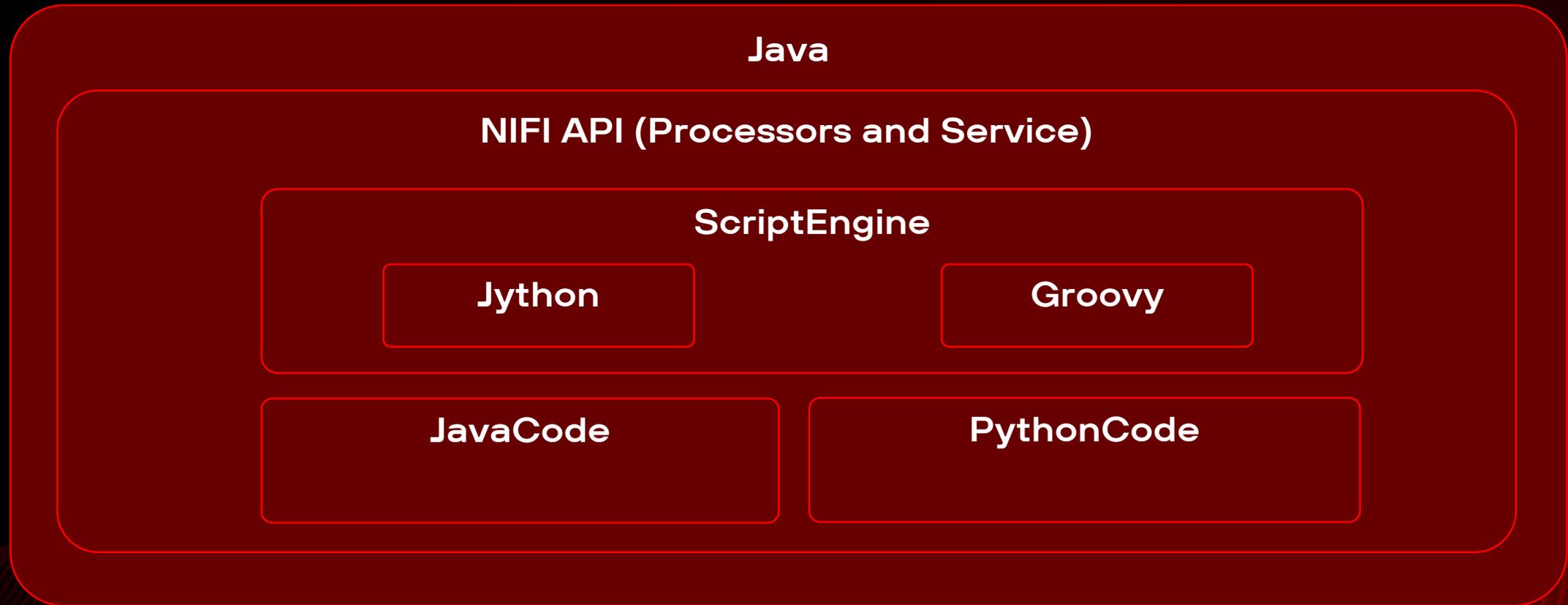
MiniFI

NIFI Stateless

# А ЧТО ПОД КАПОТОМ



# А ЧТО ПОД КАПОТОМ



# Plugins

Processors

Процессоры - обработка информации

# Plugins

Controller  
Services

Processors

Сервисы - общая функциональность и State между компонентами.

# Plugins

Reporting  
Tasks

Controller  
Services

Processors

Движок отчетов – механизм извлечения информации о работе NIFI и передаче ее во вне.

# Plugins

FlowFile  
Prioritizer

Reporting  
Task

Controller  
Services

Parameter  
Provider

Processors

Authority  
Provider

04

pt

# Создаем Processor

# Шаги процесса

- Читаем Developers Guide
- Используем Maven-Archetype
- Заполняем аннотации
- Заполняем OnSchedule
- Заполняем OnTrigger



# Developers Guide



# Maven Archetype

- Быстрый старт
- Базовые элементы
- Есть только для процессоров и сервисов



# Аннотация

- Определяют поведение
- Наполняют документацию
- Определяют ключевые методы процессоров

# Поведение

- @InputRequirement – Требуется ли входящей очереди
- @TriggerWhenEmpty – Требуется ли FlowFile
- @PrimaryNodeOnly – Запускать только на ведущей ноде
- @SupportBatching – Возможность запускать длительный Run.

# Документирование

- @CapabilityDescription - Описание
- @UseCase(s)/@MultyProcessorUseCase(s) - Сценарии
- @ReadAttribute/@WriteAttribute – Данные на вход  
выход
- @SeeAlso – что еще почитать

# Расширенное Документирование

- В каталоге `resources` создаем `docs`
- Создаем каталог с полным именем
- Создаем файл `additionalDetails.html`

# Важные элементы

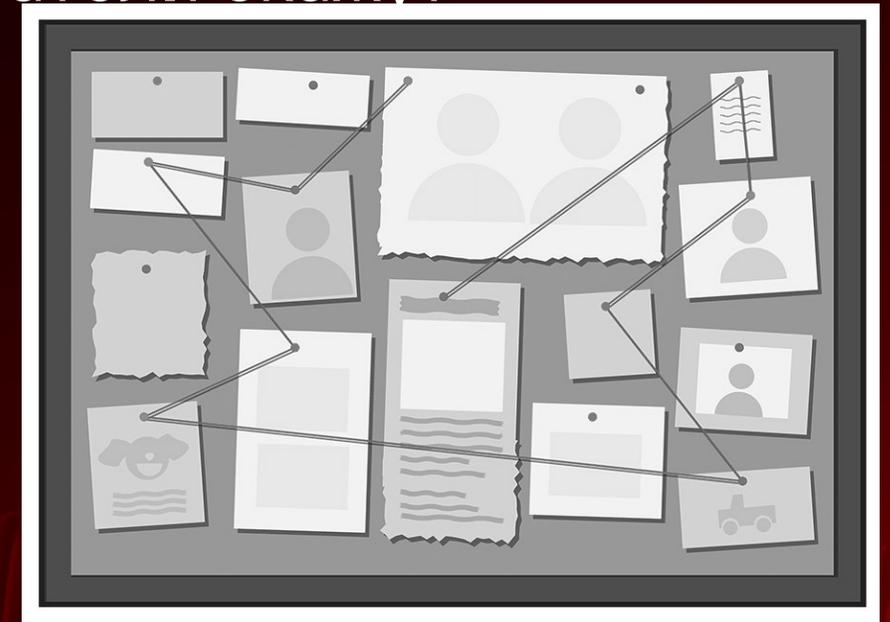
- Properties – параметры процессора
- Relationships – выходы процессора
- Lifecycle – старт, обработка, отключение

# Properties

- Как обрабатываются
- Работают ли с сервисами
- Dynamic properties
- Настраиваем валидаторы и обязательность
- Добавляем описание

# Отношения

- Имя
- Описание
- Выключен по умолчанию – Пользователи скажут спасибо



# Жизненный Цикл

- OnAdded – Создание экземпляра процессора
- OnScheduled – Запуск процессора
- OnUnscheduled – Выключение процессора
- OnStopped – Остановка процессора
- OnRemoved – Удаление экземпляра процессора
- OnShutdown – Выключение NIFI

# OnTrigger

- Проверка наличия файла
- Помним о экземплярах вызова
- Рассчитываем только из атрибутов
- Обновление контента и атрибутов не пересекаются
- Не забываем отправлять файлы или удалять
- `session.penalize`
- `context.yeld`

# OnTrigger

- ProcessSession – управляет обработкой файла
  - session.get, session.transferr
  - session.commit, session.rollback
- ProcessContext – доступ к информации процессора и среды
  - Получение информации о процессоре
  - Получение информации о среде
  - Управление процессором (yield)

# Опасности



---

## Оптимизация процесса

- Ответственность
- Прозрачность
- Контроль



---

## Можем лучше

- Механизмы NIFI
- Оптимизация хранения
- Опыт сообщества

# Про Python

- Ожидаемая фишка
- Упрощение разработки
- Но не все так радужно

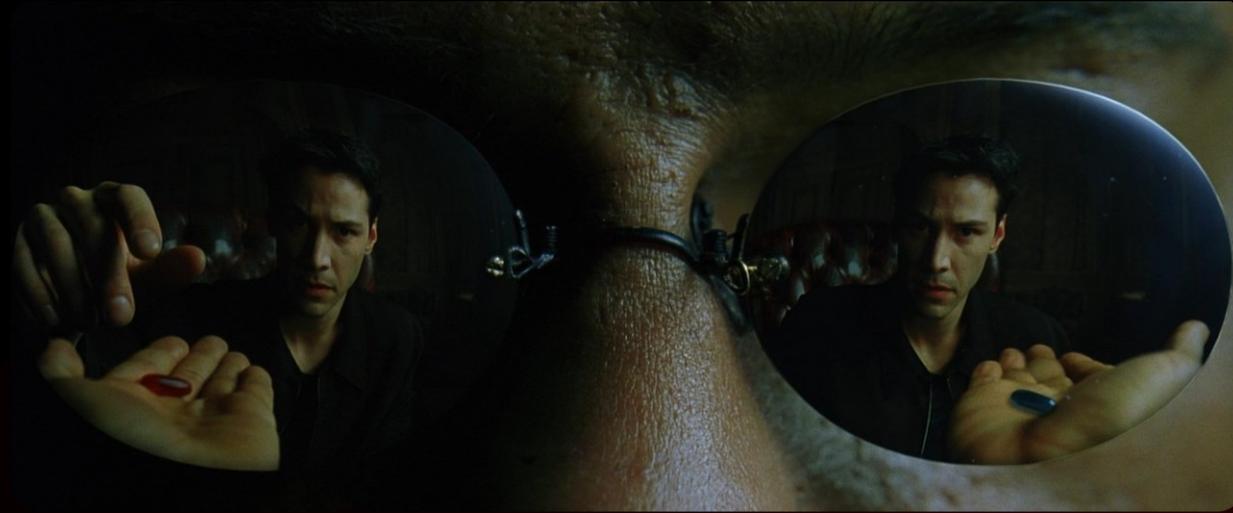


# Developers Guid



# В чем разница

- API все-таки отличается
- FlowFileTransform – для General
- RecordTransform – для Record
- Можно задать и Properties и Relations
- onScheduled и onStopped тоже есть
- Установка в python/extensions



Java



Python



Powerfull NIFI

# Вопросы?