# Разработка инструмента для добавления статической верификации в код

**Автор:** Маллабаев Азамат Нурмухамадович, 143 группа
**Научный руководитель:** ст.пр. Григорьев Семён Вячеславович

Санкт-Петербургский государственный университет

19 мая 2016г.

# Введение

- Тестирование — доказательство некорректности программы
- Верификация — доказательство корректности программы
  - Статическая — во время компиляции
  - Динамическая — во время выполнения

# Инструменты верификации

- AutoProof — верификатор языка Effel
- Coq — интерактивное средство для доказательства теорем
- F$*$ — язык с поддержкой верификации
  - ML-подобный, каким также является F#
  - Работает в DotNet
  - Транслируется в F# без учета ограничений
- Z3 — низкоуровневый инструмент верификации

# Атрибуты

**Атрибуты** — поля, применимые к элементам программы

**Атрибуты в F#**

```
1 [<Obsolete("Код -- баян")>] //устаревший метод
2 [<EntryPoint>] //атрибут начала программы
3 let helloSayer () = //метод, к которому применены атрибуты
4    printfn "Hello, world!!!"
```

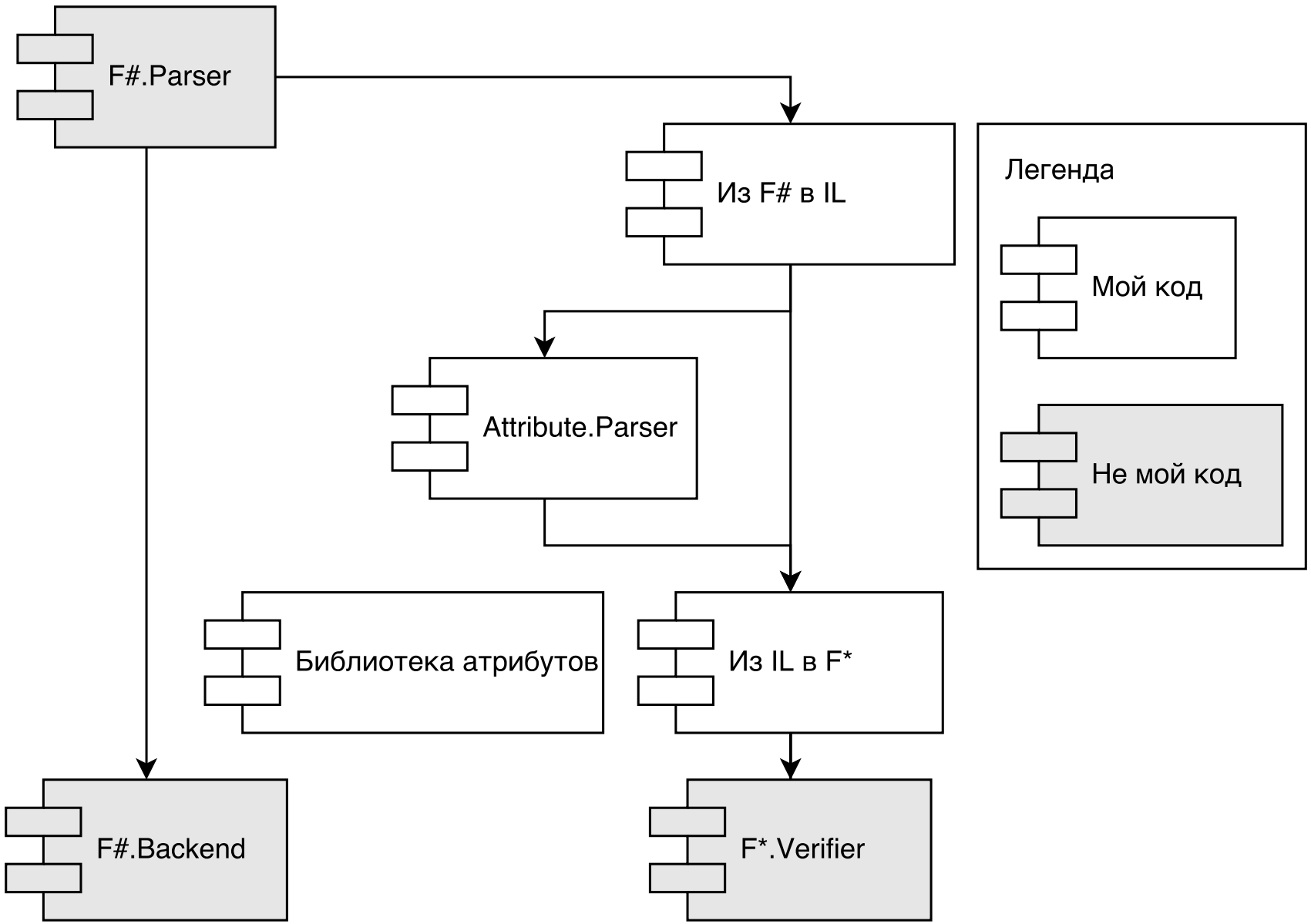
# Постановка задачи

**Цель:** разработка инструмента для добавления статическиой верификации в код на F# с использованием верификатора F*

**Задачи**

- Изучить компилятор F# и верификатор F*
- Разработана архитектура системы
- Разработать транслятор подмножества F# в F*

# Аритектура инструмента



# Поддерживаемые выражения

- Модули
- Типизированное опрделение функции
- Условный оператор if-then-else
- Применение функции
- Бинарные операции
- Целые числа
- Выражения верификации forall и exist

# Пример использования атрибута верификации

```
///Верифицируемая функция
1 [<Total("forall x y . x < y ==> i x + 2 <= i y")>]
2 let f (x: int): int =
    if x > 1
    then x * x * x
    else x - 2
```

```
//Неверифицируемая функция
1 [<Total("forall x y . x < y ==> i x < i y")>]
2 let g (x: int): int = x * x
```

# Результаты

- Изучены компилятор F# и верификатор F*
- Разработана архитектура системы
- Разработан транслятор из подмножества F# в F*