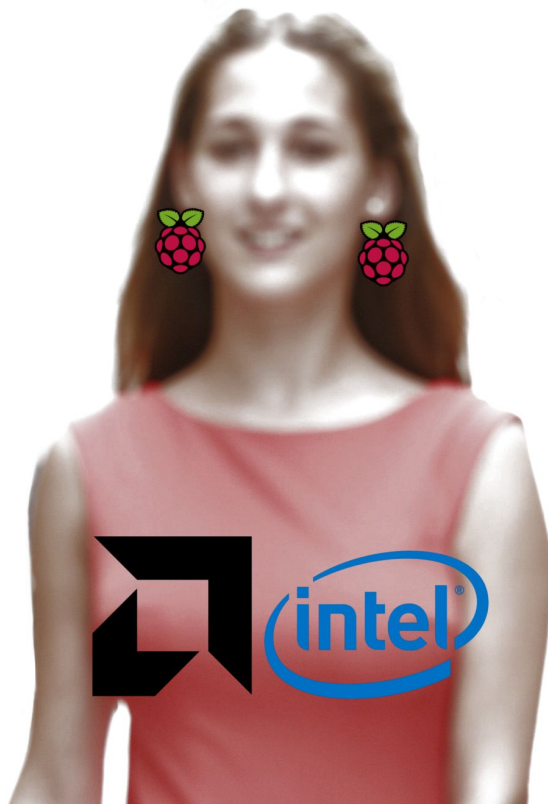


K/N S/P

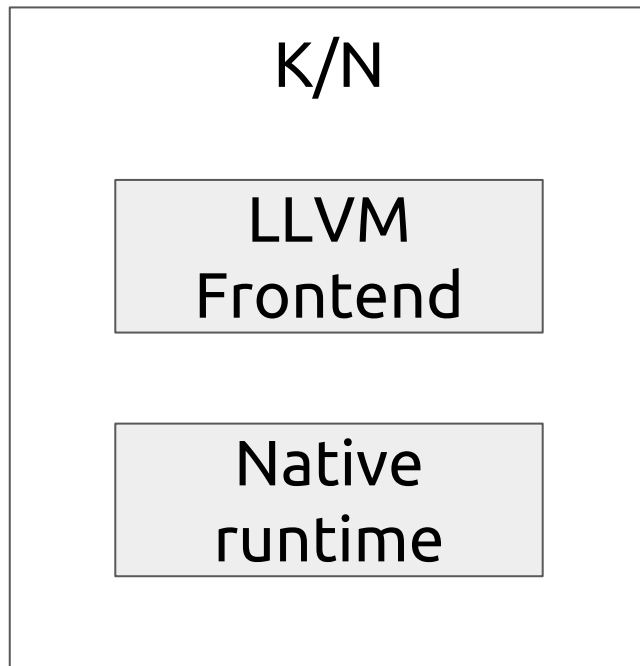
Ведутся работы!



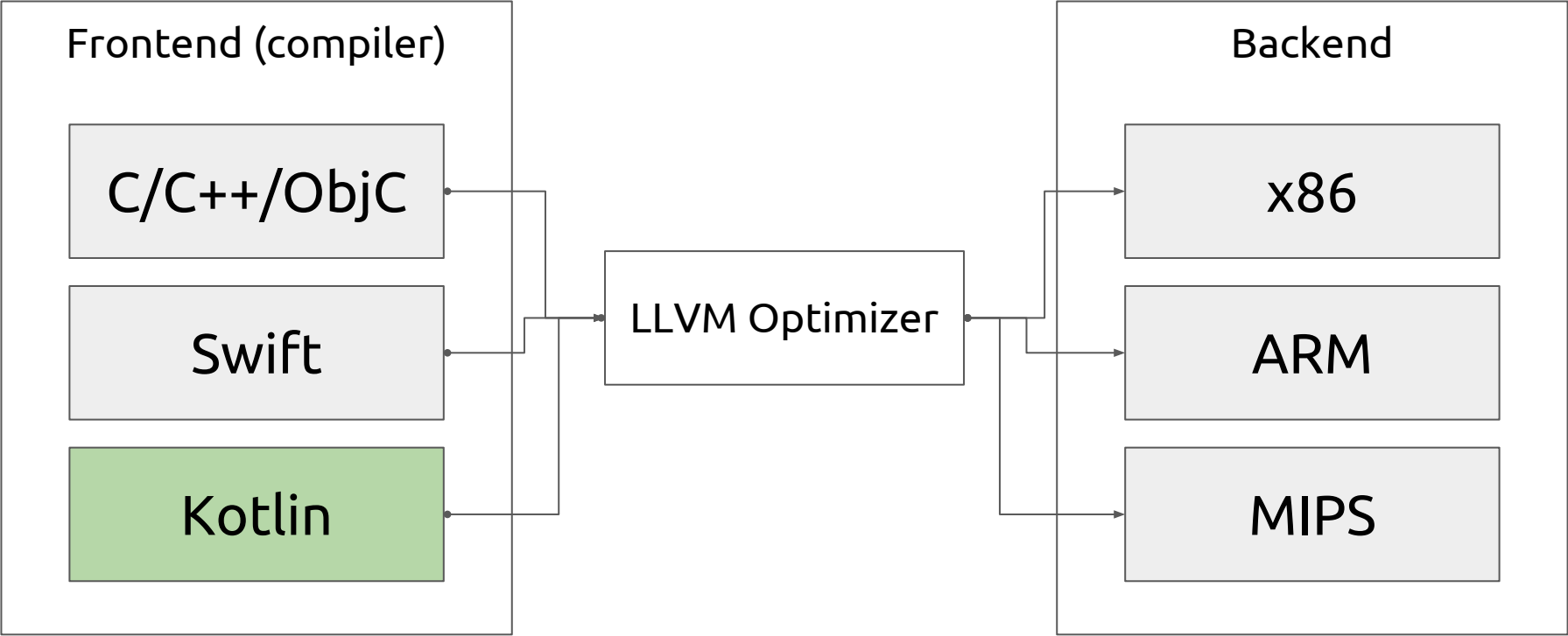
DistraKted boyfriend



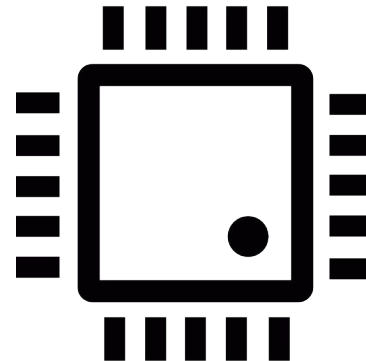
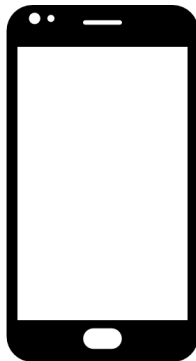
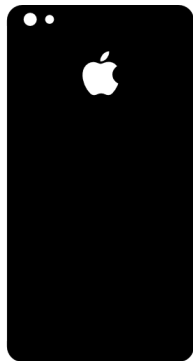
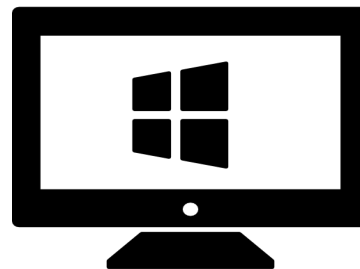
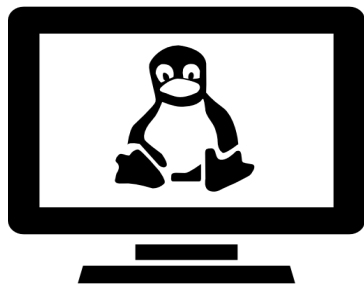
На пальцах



LLVM



Платформы



Kotlin stdlib

`kotlin`

`kotlin.collections`

`kotlin.io`

`kotlin.math`

`kotlin.coroutines`

`kotlin.reflect`

`kotlinx.cinterop`

K/N “stdlib”

platform.posix

platform.linux

platform.osx

platform.windows

platform.android

platform.ios

Сторонние библиотеки



Артефакты

Исполняемые файлы

Динамические библиотеки

KLibray

Интерпретатор-библиотеки

Objective-C-фреймворки

LLVM биткод

Входная точка

```
import kotlinx.cinterop.*
import platform.posix.*
import mqtt.*

fun main(args: Array<String>) {
    ...
}
```

Типы

```
u_byte    =>    kotlin.Int
int*      =>    CPointer<kotlin.Int>
enums     =>    [kotlin.Int, enum]
struct    =>    class
typedef   =>    typealias
```

Указатели

`CPointer<T>.pointed` \Rightarrow `T`
`T.ptr` \Rightarrow `CPointer<T>`

Управление памятью

```
nativeHeap.alloc<T>()
```

```
nativeHeap.allocArray<T>(size)
```

```
nativeHeap.free(ptr)
```

Лексические скоупы

```
memScoped {  
    alloc<T>()  
}
```

Строки

```
CPointer<ByteRef>.toKString() => String  
String.cstr    => CValuesRef<ByteRef>
```


cinterop .def

```
# Dynamic library
```

```
headers = libastral.h
```

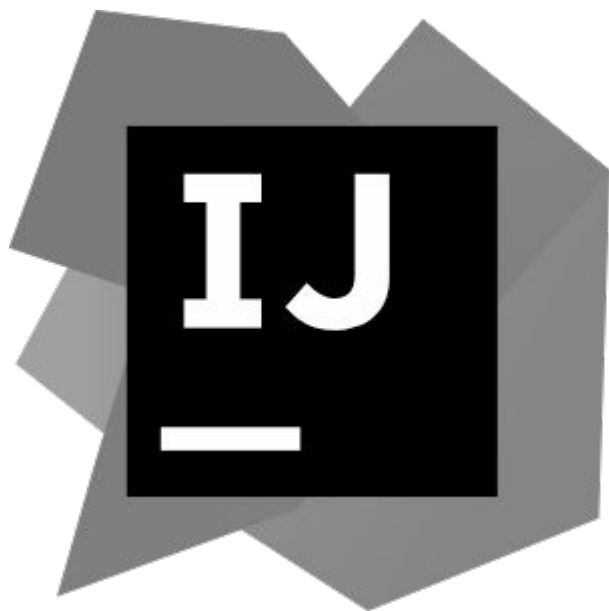
```
linkerOpts.linux = -L/usr/local/lib -llibastral
```

```
# Static library
```

```
staticLibraries = libastral.a
```

```
libraryPaths = /usr/local/lib64
```

Поддержка IDE



Демо



ИТОГИ

- + It works!
- + Развивается
- Недоделанный

Спасибо!