Actors as co-routines

```
exec::task<std::string> get_first_name() const
{
    co_return co_await stdexec::then (stdexec::schedule (get_scheduler()),
                                      [this] { return person.get_first_name(); });
}
```

```
exec::task<void> set_first_name (std::string new_first)
{
    co_return co_await stdexec::then (stdexec::schedule (get_scheduler()),
                                      [this, =]
                                      { return person.set_first_name (new_first); });
}
```

Actors as co-routines

```
exec::task<std::string> get_first_name() const
{

                                      [this] { return person.get_first_name(); });
}
```

```
exec::task<void> set_first_name (std::string new_first)
{

                                      [this, =]
                                      { return person.set_first_name (new_first); });
}
```

```swift
actor Person
{
    private var first_name: String = "";

    func set_first_name (n: String) {
       first_name = n;
    }

    func get_first_name() -> String {
        return first_name
    }
}
```

```cpp
struct person(actor)
{
    std::string get_first_name() const {
        return first_name;
    }

    void set_first_name (std::string_view n) {
        first_name = n;
    }

private:
    std::string first_name;
};
```