

Asynchronous Effects



Asynchronous Effects

"Lift" an external computation to ZIO

- use a callback object created by the ZIO runtime
- invoke the callback with a value/error to complete the ZIO
- block the ZIO (semantically) until the callback is invoked

```
def loginAsZIO(id: String, pw: String): ZIO[Any, AuthError, UserProfile] =  
  ZIO.async[Any, LoginService.AuthError, LoginService.UserProfile] { cb =>  
    LoginService.login(id, pw)(  
      profile => cb(ZIO.succeed(profile)),  
      error => cb(ZIO.fail(error))  
    )  
  }
```

ZIO native implementations

- ZIO.fromFuture
- ZIO.never

ZIO rocks

