```kotlin
class Chamber(private val client: HttpHandler) {
 fun echo(input: String) = client(
   Request(POST, “http://server.com/echo").body(input)).bodyString()
}

abstract class EchoChamberCdc {
    abstract val client: HttpHandler

    @Test
    fun `echoes message`() {
        assertThat(Chamber(client).echo("hello"), equalTo("hello"))
    }
}

class RealEchoChamberTest : EchoChamberCdc() {
    override val client: HttpHandler = OkHttp()
}

class FakeEchoChamberTest : EchoChamberCdc() {
    override val client: HttpHandler =
      { r: Request -> Response(Status.OK).body(r.body) }
```

CDC example

```
class Chamber(private val client: HttpHandler) {
 fun echo(input: String) = client(
   Request(POST, “http://server.com/echo").body(input)).bodyString()
}

abstract class EchoChamberCdc {
    abstract val client: HttpHandler

    @Test
    fun `echoes message`() {
        assertThat(Chamber(client).echo("hello"), equalTo("hello"))
    }
}

class RealEchoChamberTest : EchoChamberCdc() {
    override val client: HttpHandler = OkHttp()
}

class FakeEchoChamberTest : EchoChamberCdc() {
    override val client: HttpHandler =
      { r: Request -> Response(Status.OK).body(r.body) }
```

Application Testing

- All internal and external applications and clients are HttpHandlers