TESTING EMBEDDED SYSTEMS

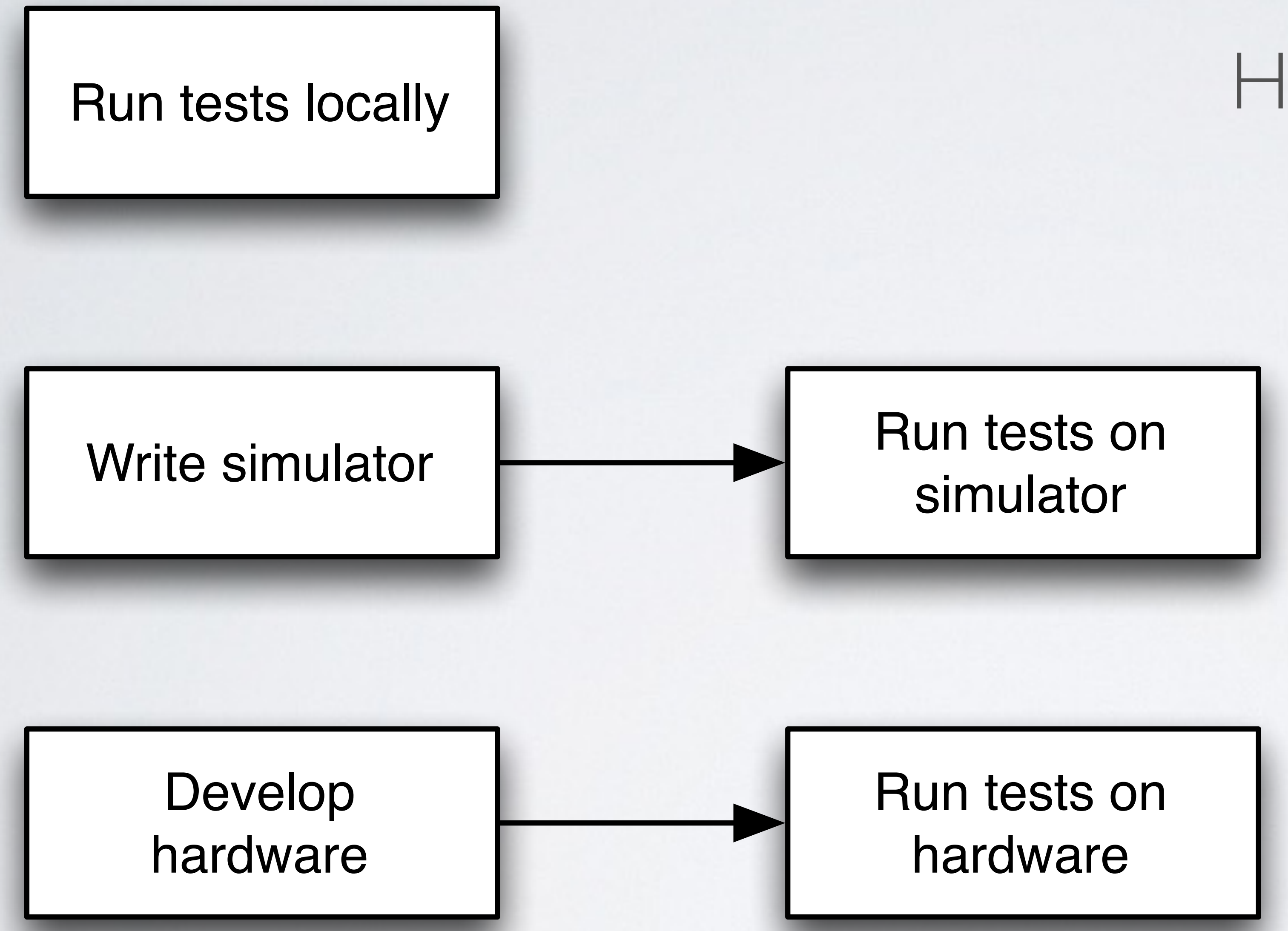
ThoughtWorks®

Itamar Hassin
Fosdem 2016
@itababy

SUBJECTS COVERED

- Unit testing (Unity)
- BDD (Cucumber) as a front-end for functional & acceptance tests
- Orchestrating tests across multiple targets

CHALLENGES TESTING EMBEDDED SOFTWARE



Hope that it cross-compiles

Hope that it runs on hardware

Optimal

SOLID TESTING

UNITY TEST CODE

```
void test_1(void)
{
   TEST_ASSERT_EQUAL(2+2, 4);
}

void test_2(void)
{
   TEST_ASSERT_EQUAL(1+1, 3);
}
```

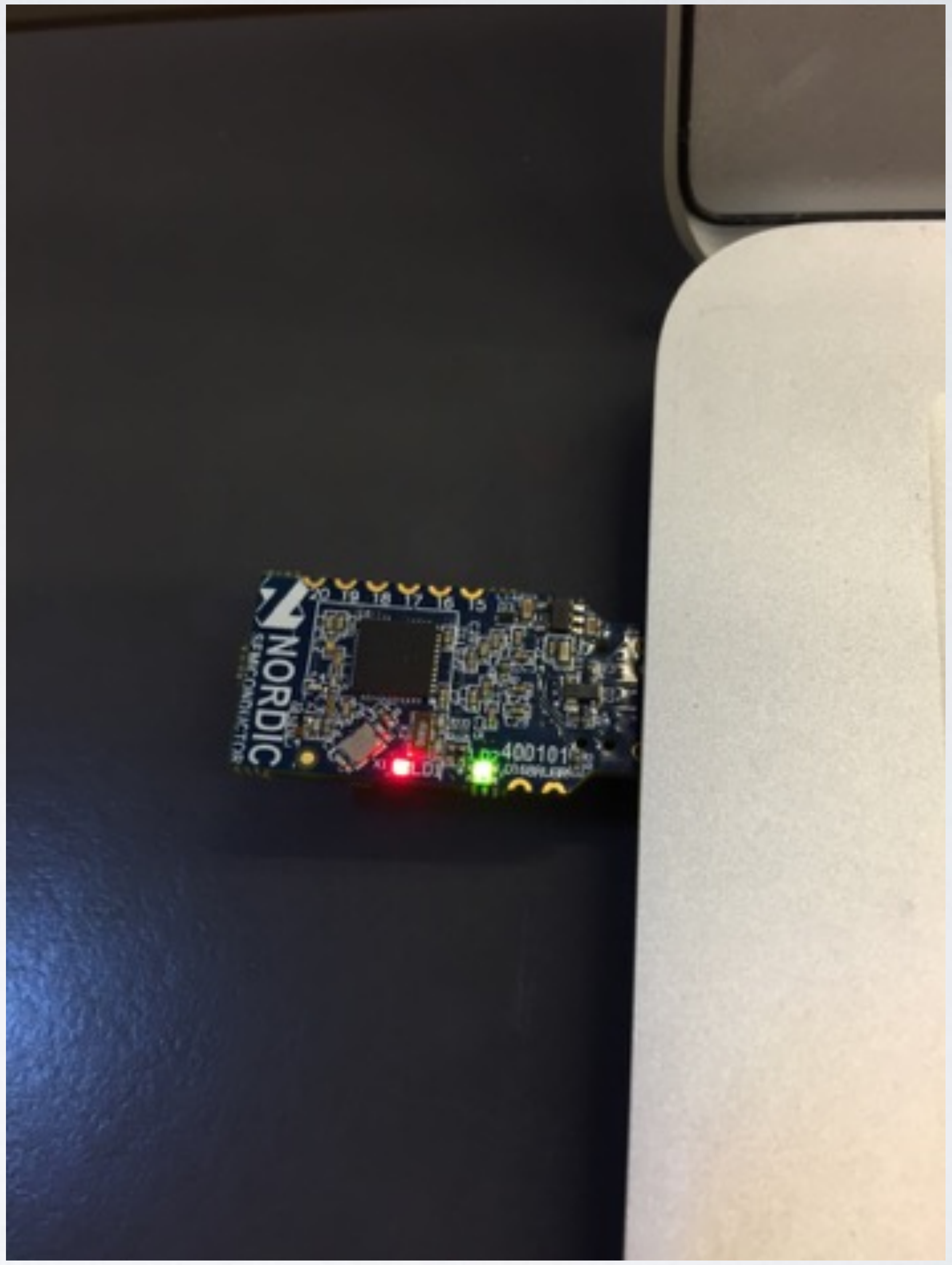
UNITY RUNNER CODE

```
int main(void)
{
    SetupTests();

    RUN_TEST(test_1);
    RUN_TEST(test_2);

   TeardownTests();
}
```

BUILD UNIT TESTS

FLASH UNIT TESTS

SEE IT RUN



BDD FOR EMBEDDED

THE CASE FOR BDD

- Describes the behaviour in simple English
- Promotes collaboration within the product team
- Highlights business value
- Direct mapping from user story acceptance criteria
- Living documentation, unified view of the product

COLLABORATION

```
Feature: Patient monitoring

Scenario: Alert nurse on disconnect
  Given patient is monitored
  When I disconnect the monitor
  Then I am alerted
```

IMPLEMENT A SIMULATOR

```
class Monitor
 def disconnect
  driver.led(RED, ON)
 end
end
```

IMPLEMENT FEATURE STEP

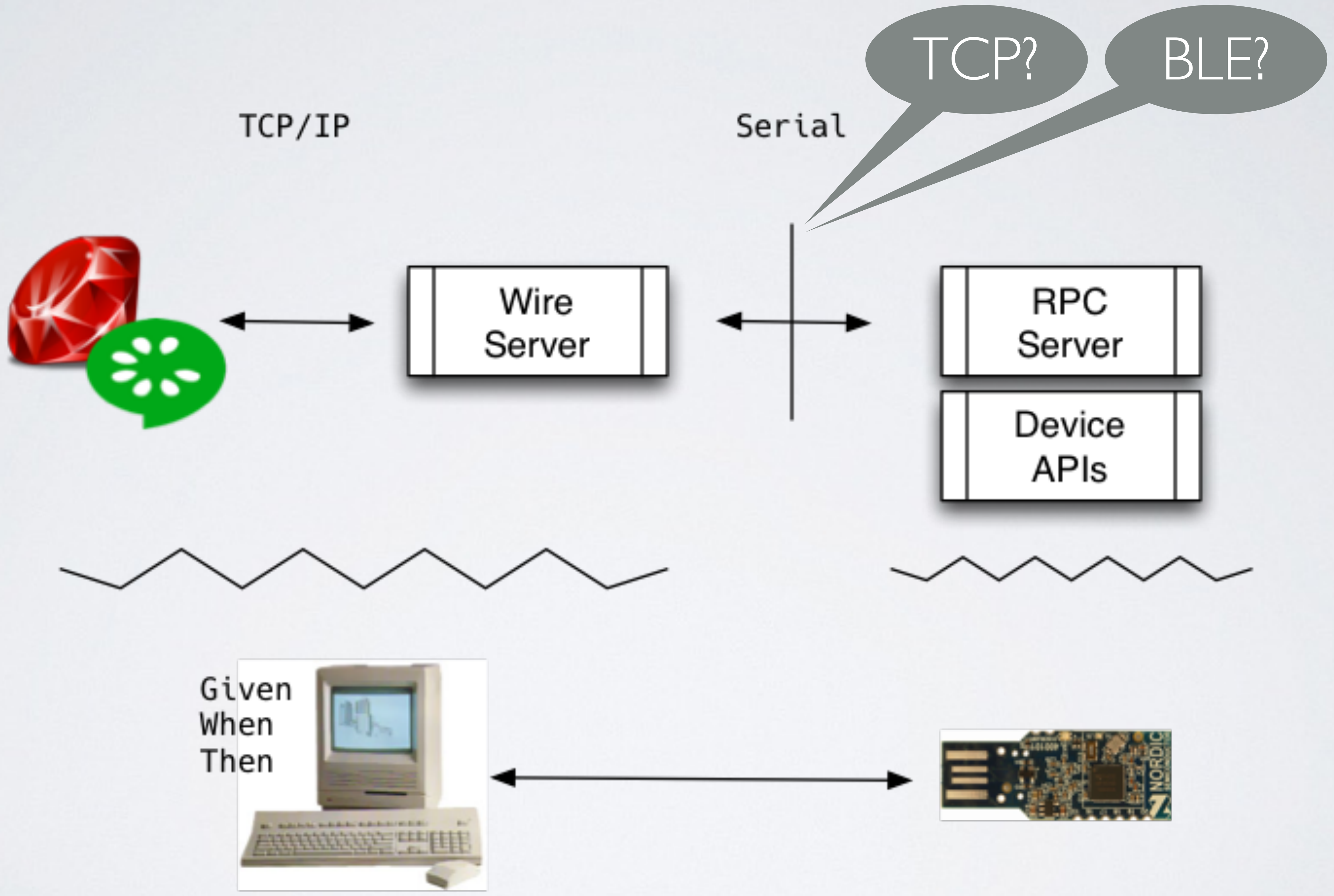
```
Given(/^patient is monitored$/) do
  pending
end

When(/^I disconnect the monitor
$/) do
  monitor.disconnect
end
```

VALIDATE UNDER SIMULATOR

```
Feature: Patient monitoring

Scenario: Alert nurse on disconnect
  Given patient is monitored
  When I disconnect the monitor
  Then I am alerted
…
1 scenarios (1 passed)
3 steps (3 passed)
0m0.0052s
```

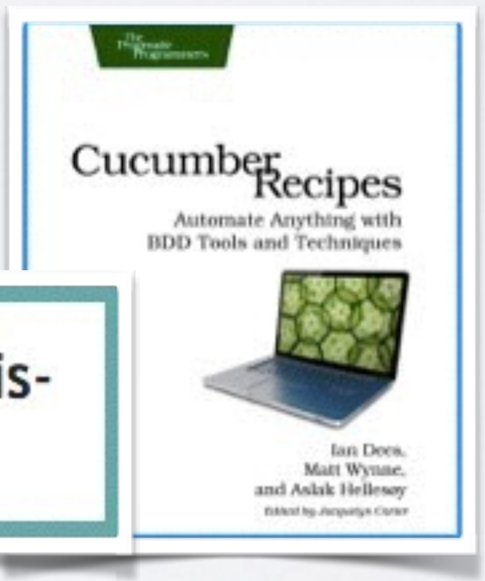
WHEN SIMULATION IS NOT ENOUGH

IN-SITU TESTING



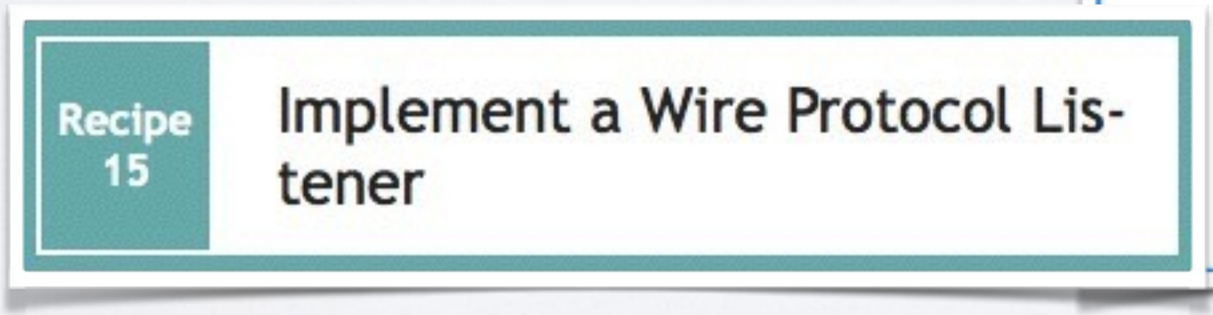
THE "WIRE"

- When your system does not have native support
- When you want a lean, portable implementation


Cucumber Recipes

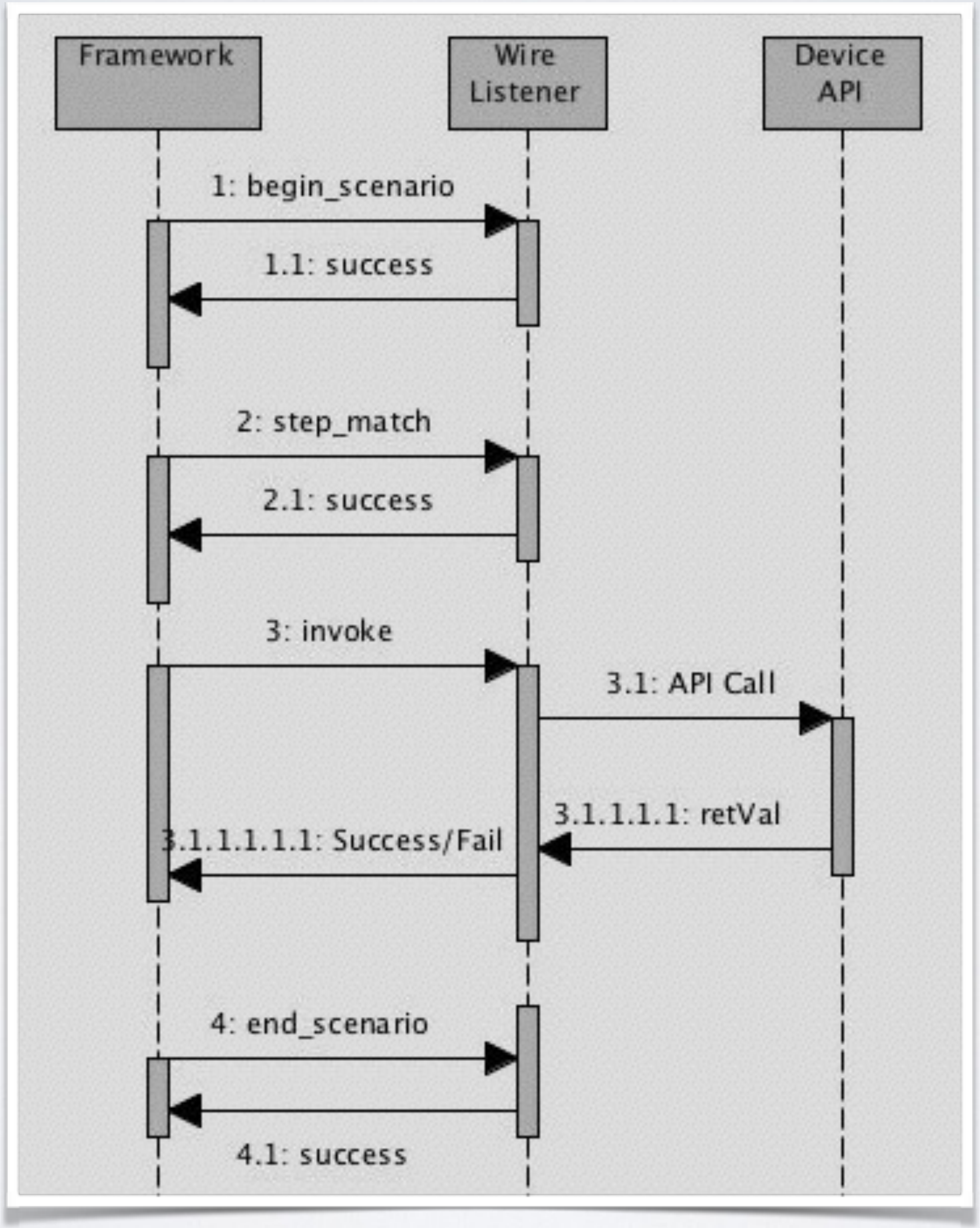
Automate Anything with BDD Tools and Techniques

Ian Dees, Matt Wynne, and Aslak Hellesøy


Recipe 15

Implement a Wire Protocol Listener

SIMPLIFIED WIRE PROTOCOL

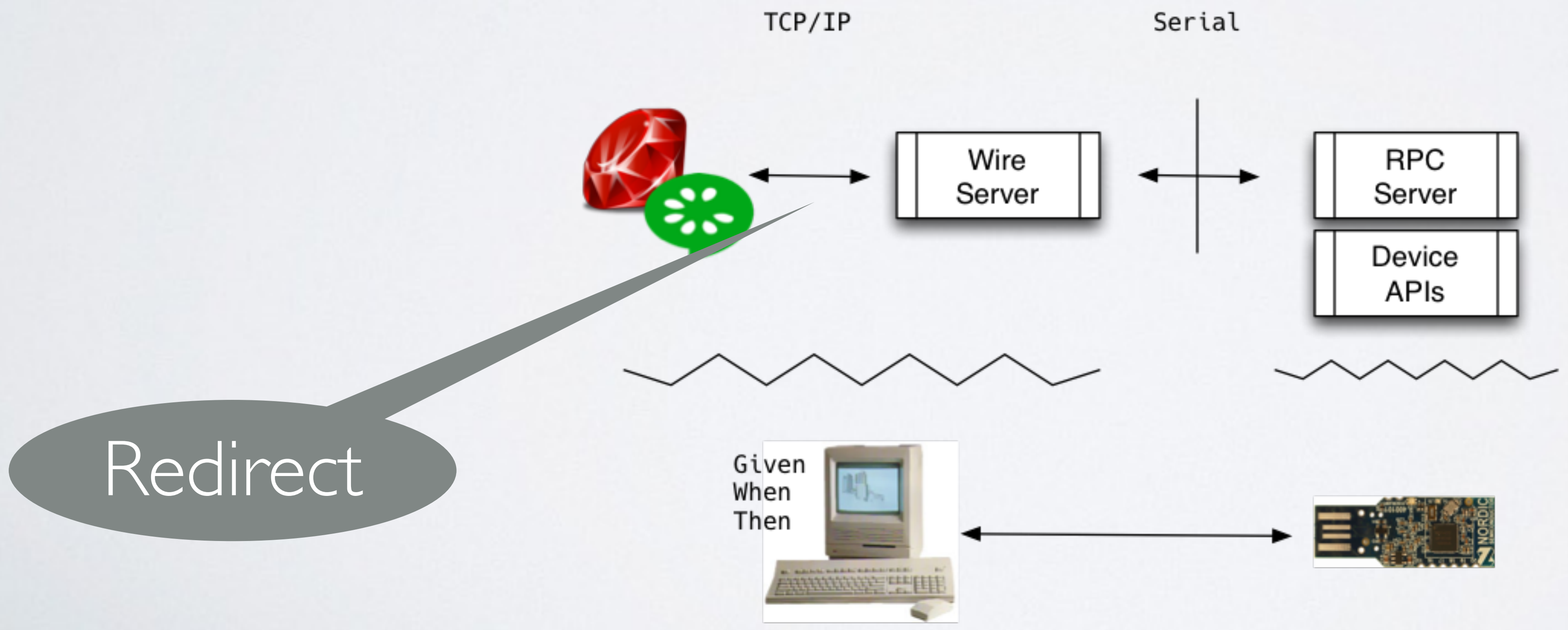


WIRE IMPLEMENTATION BLUEPRINT

- TCP/IP loop managing Cucumber protocol
- Function table for API invocation
- API proxy implementation returning status to Cucumber

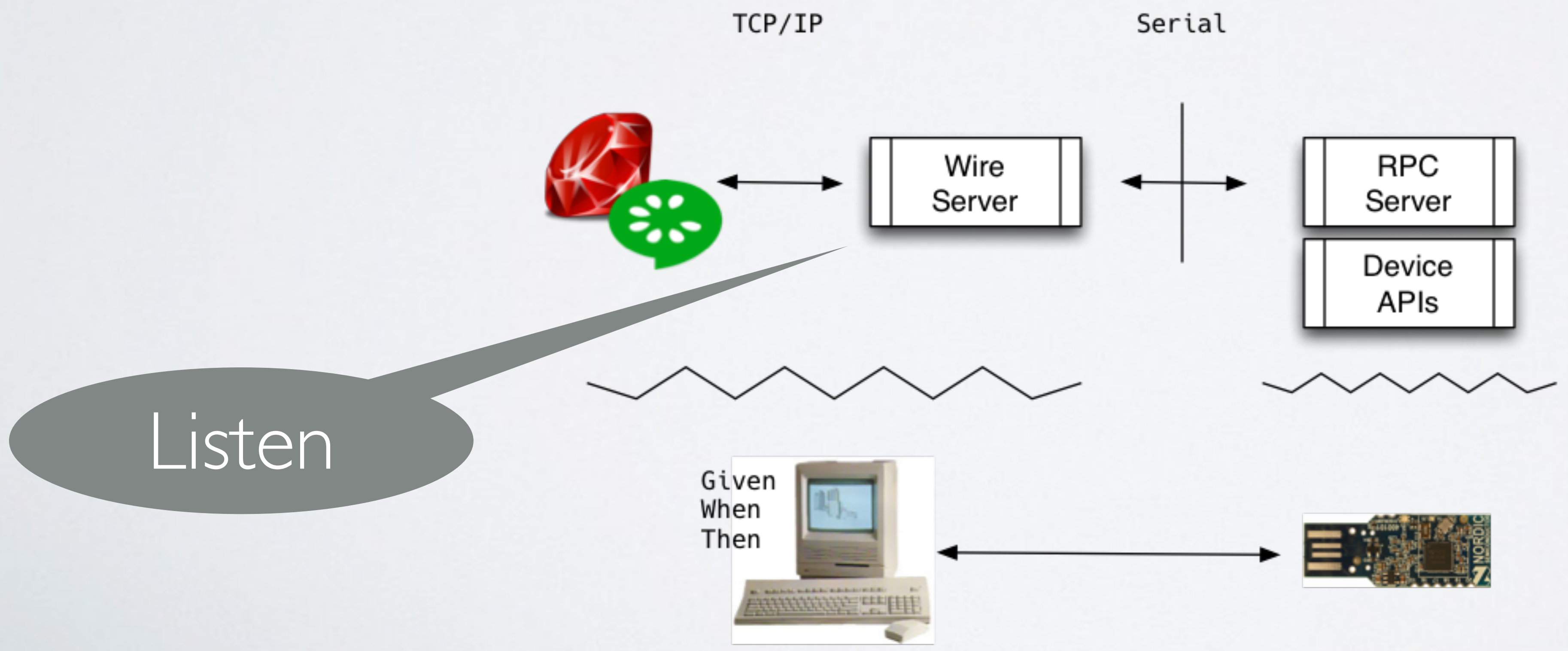
HOST HOOKING CUCUMBER TO WIRE SERVER

```
features/step_definitions/cucumber.wire
 host: host
 port: 3901
```



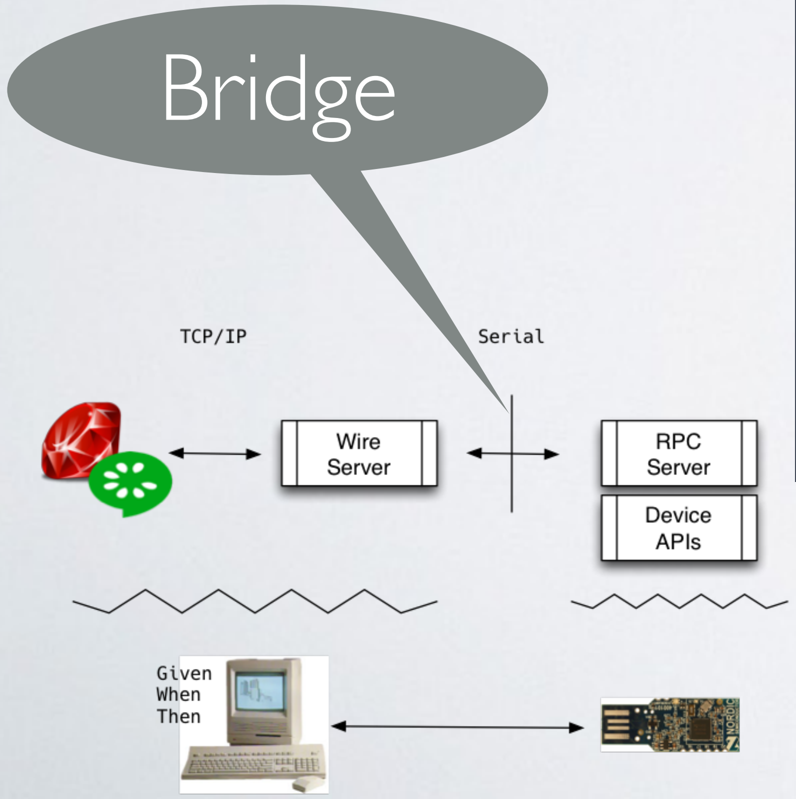
SERVER TCP/IP LOOP

```
while(1)
{
   getRequest(…);
   handleRequest(…);
}
```



BRIDGE TO DEVICE

```
int begin_callback(…)
{
    serial_open(…);
}

int end_callback(…)
{
    serial_close(…);
}
```



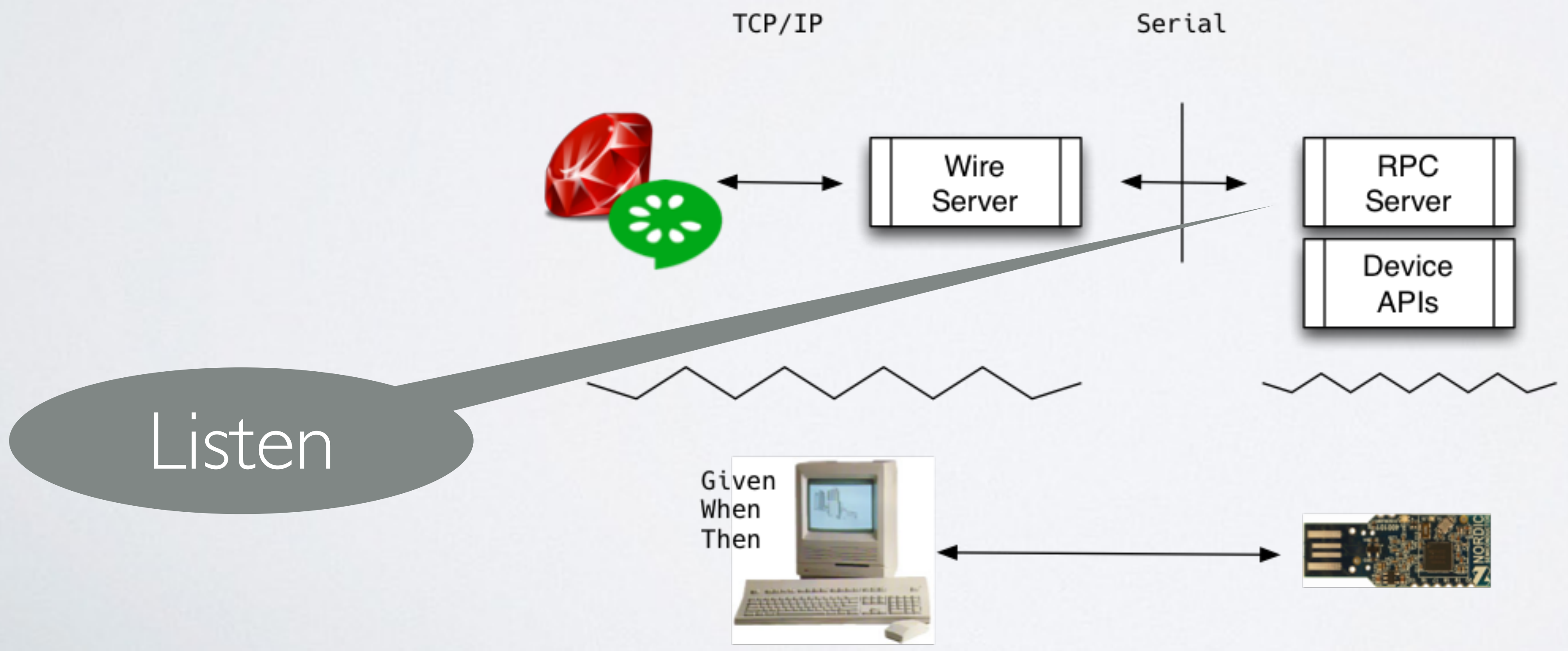
UP CLOSE AND PERSONAL

WIRE SERVER TO THE DEVICE

```
int patient_is_monitored(…)
{
 serial_write(…,”EXEC 0\r”);
 serial_read(…);
 return(retVal);
}
```

DEVICE RPC SERVER LOOP

```
while (true)
{
  chr = uart_read_byte();
  handle(command_buffer);
}
```



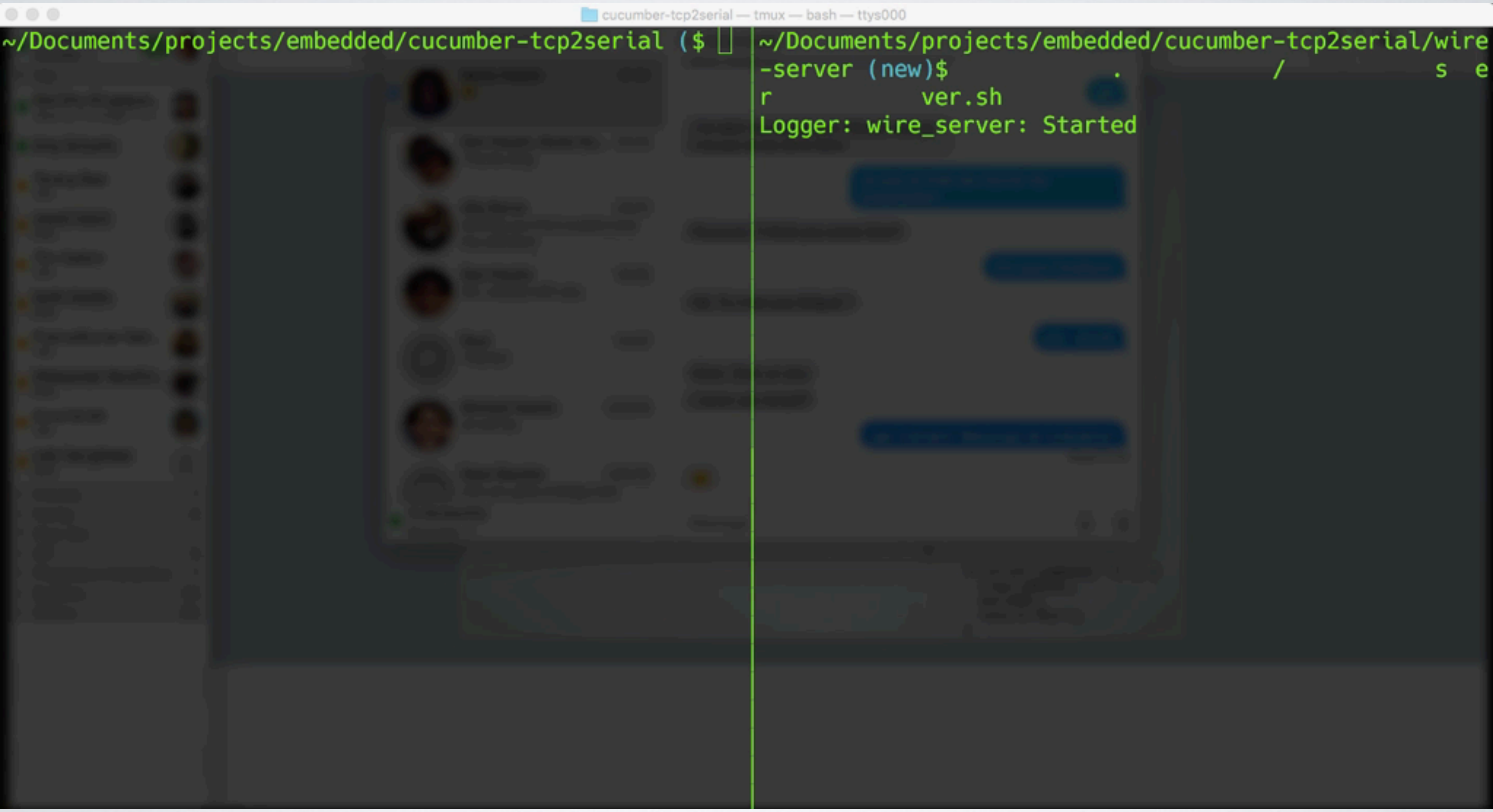
DEVICE API IMPLEMENTATION

```
if(strstr(command, “1”))
{
  nrf_gpio_pin_clear(GREEN);
  nrf_gpio_pin_set(RED);
  return(”0\n”);
}
```

WIRE SERVER BACK TO CUCUMBER

```
if(retVal == 0)
{
  strcpy(buffer, ”[\”success\”]\n”);
} else {
  sprintf(buffer, “[\”fail\”, …);
}
```

RUNNING THE TEST

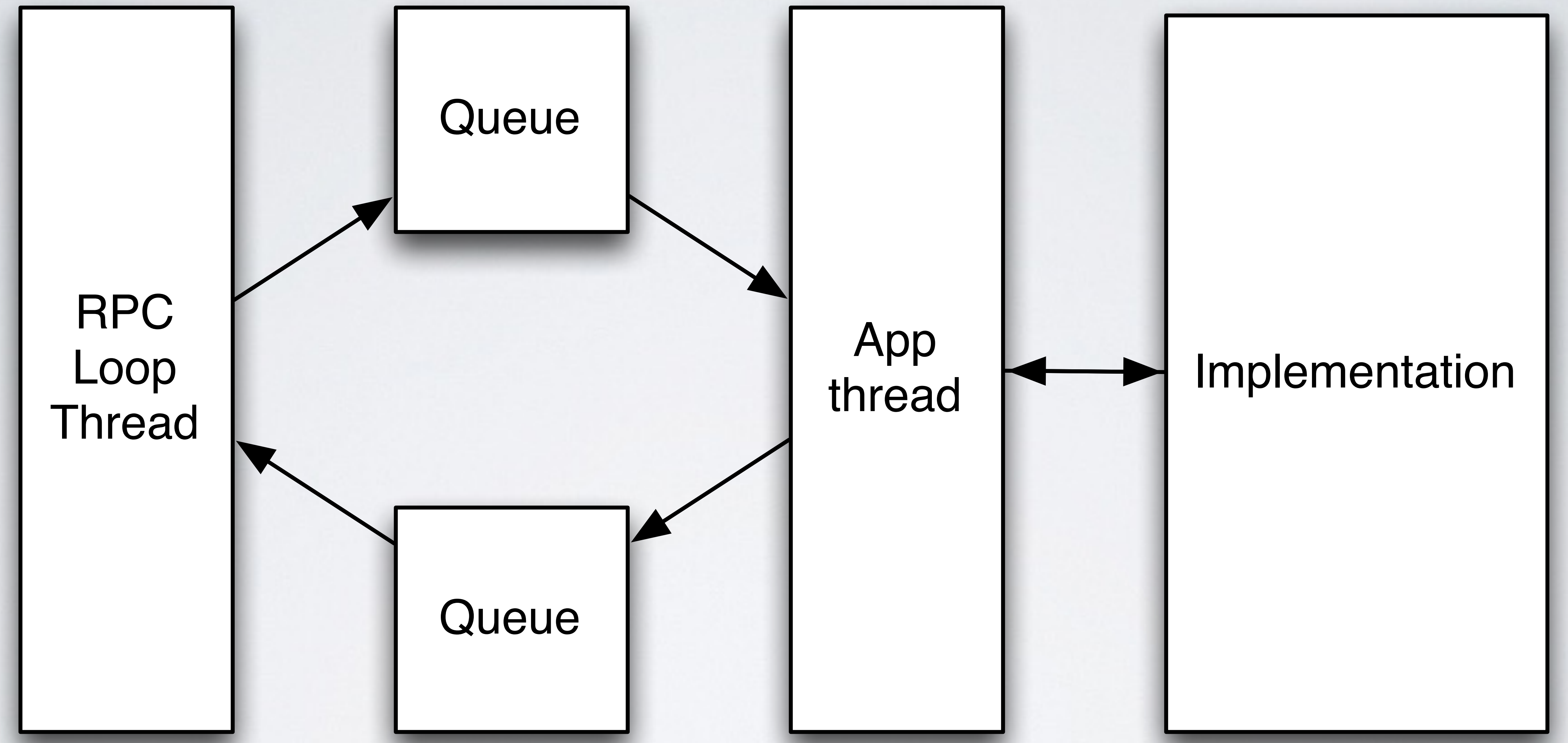


```
~/Documents/projects/embedded/cucumber-tcp2serial ($ 
```

```
~/Documents/projects/embedded/cucumber-tcp2serial/wire
-server (new)$          .          /           s  e
r           ver.sh
Logger: wire_server: Started
```

SEE IT RUN

REAL APPS NEED THREADS

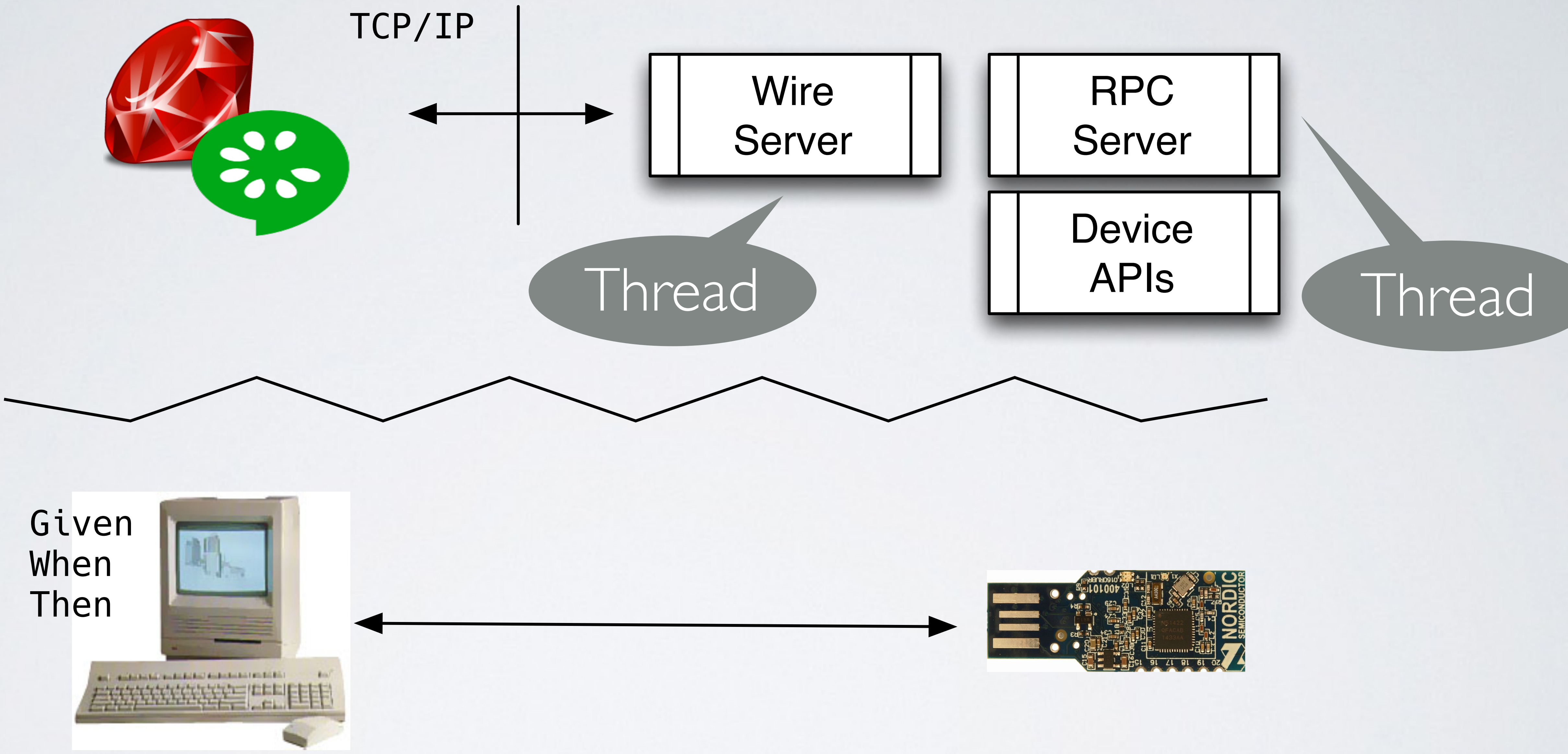


IMPLEMENTATION STACK

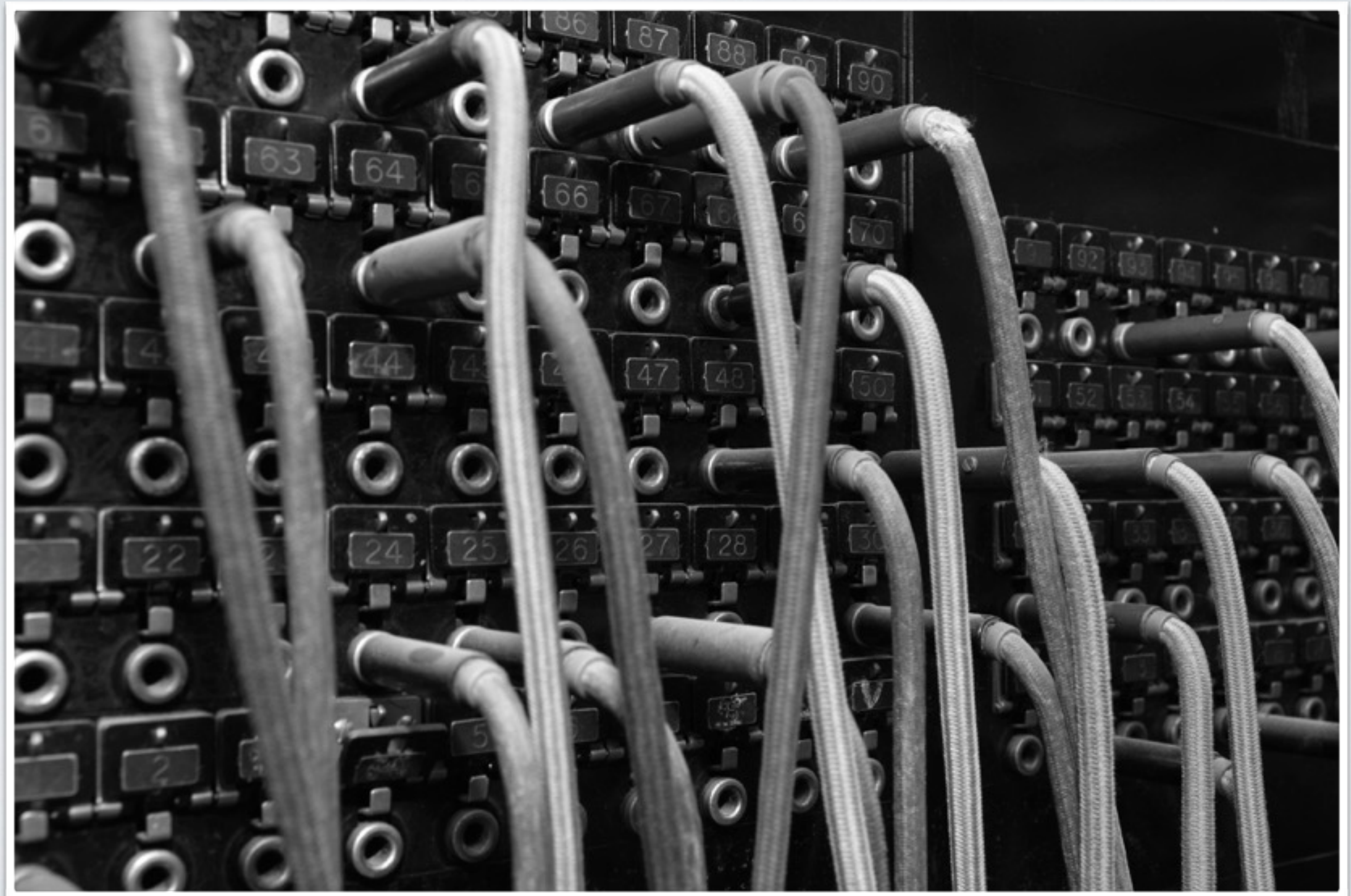
WORKING WITH CUCUMBER

- Decide on a strategy (off-board, on-board)
- Get appropriate toolchain (cross compiler, linker)
- Implement and port Wire to target
- Run the feature files
- **fail/implement/pass/refactor/repeat**

SCRIPTING THE DEVICE



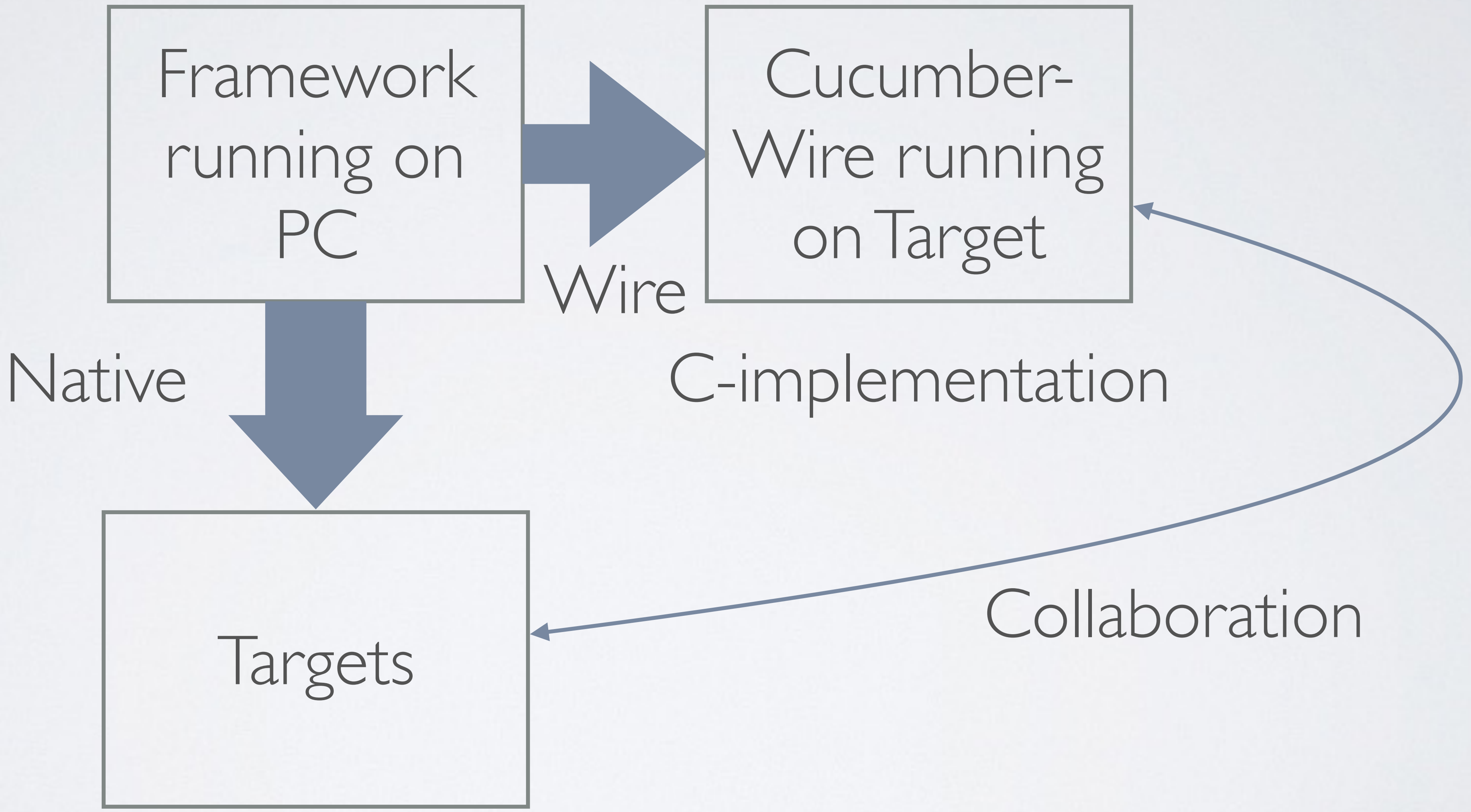
COMPLEX ENVIRONMENT



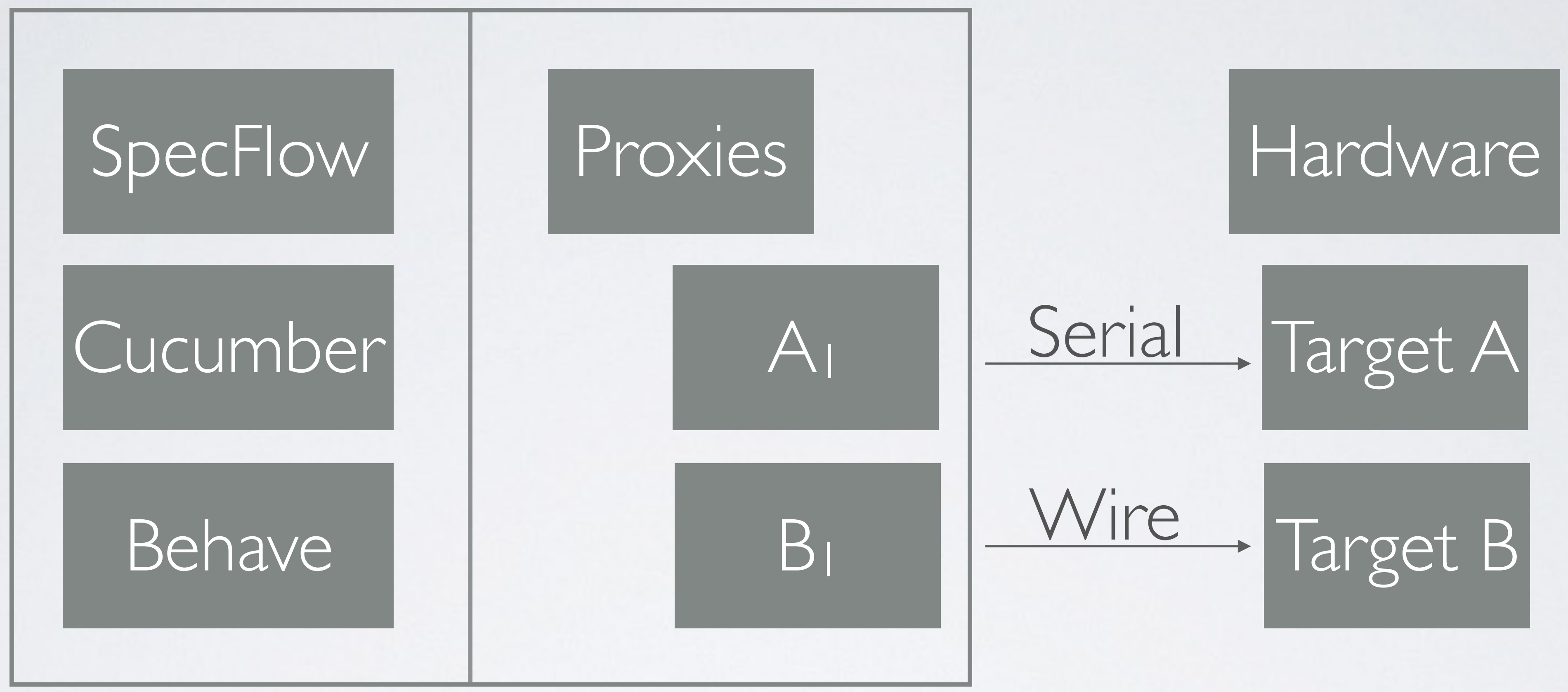
GATEWAY

- Acts as an end-to-end test orchestrator
- Switchboard events across heterogeneous devices

COLLABORATIVE END-TO-END TESTING



GATEWAY ARCHITECTURE



END-TO-END FEATURES

Feature: Alarm assured to appear in quiet mode

Scenario: Pressure alarm
    Given device is in quiet mode
    When pressure sensor is disconnected
    Then a silent alarm will appear

GATEWAY STEPS

```
public class QuietModeSteps
{
    SignalSimulator signalSimulator = new **SignalSimulator**();
    MedicalDevice medicalDevice = new **MedicalDevice**(“192.168.1.1”, 3901);

    [Given(@"device is quiet mode")]
    public void GivenDeviceIsQuietMode()
    {
        Assert.IsTrue(medicalDevice.SetQuietMode());
    }

    [When(@“pressure sensor is disconnected")]
    public void GivenPressureSensorIsDisconnected()
    {
        Assert.IsTrue(signalSimulator.SetPressure(off));
    }
}
```

Serial

Wire
on device

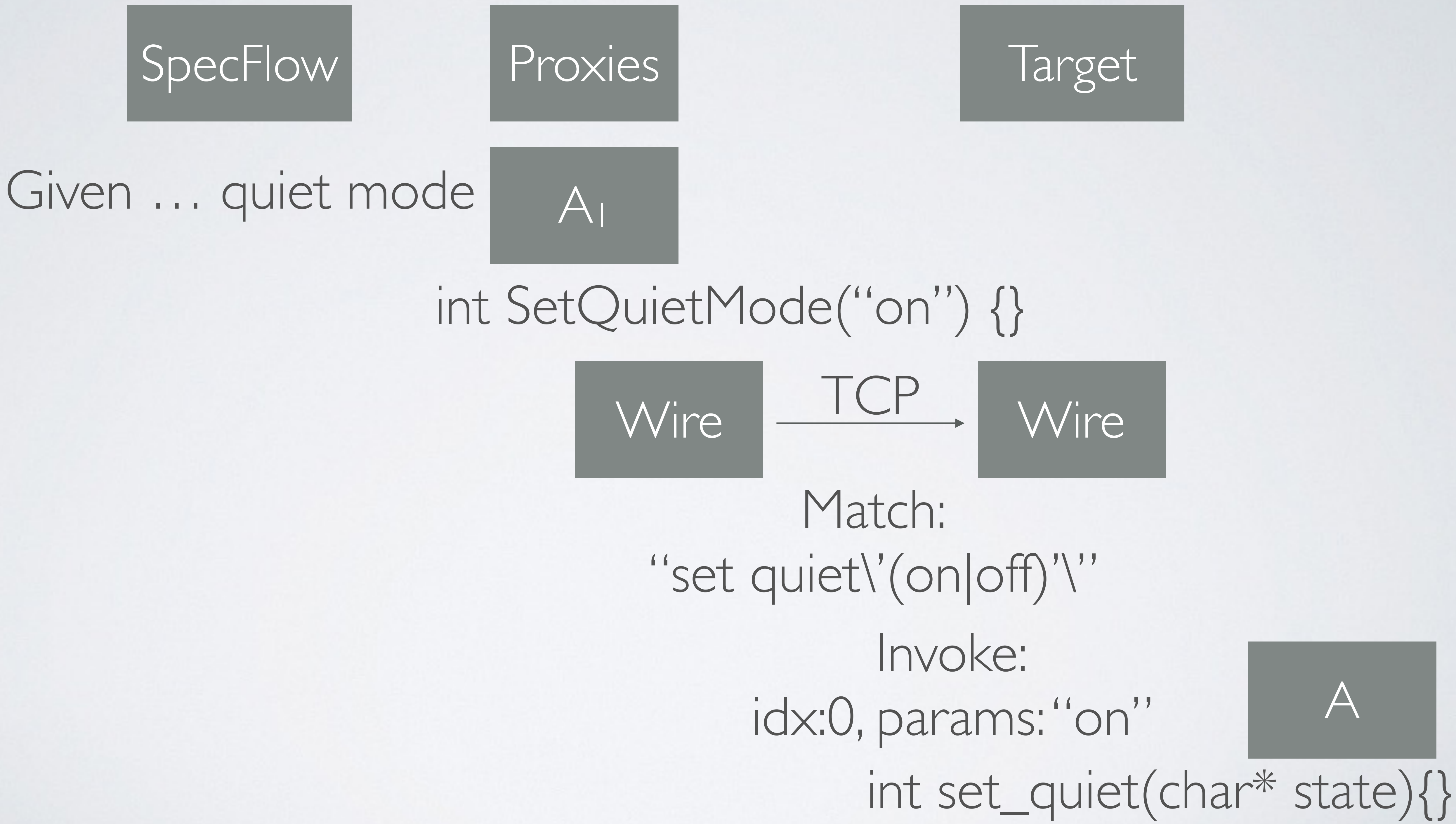
GATEWAY PROXIES

```
class MedicalDevice
{
 public MedicalDevice(string ipAddress, int port)
  {
    wire = new Wire(myAddress, port);
    wire.Open();
  }

  public bool SetQuietMode()
  {
    wire.Send(“[\"step_matches\",
      {\"name_to_match\":\"set quiet mode on\"}]\n");
    wire.Send("[\"invoke\",{\"id\":\"7\",\"args\":[\"on\"]}]\n");
    return(wire.Ack());
   }
}
```

EMULATING WIRE

```
public class Wire
{
  public int Open()
  {
    client = new TcpClient(myAddress, myPort);
    stream = client.GetStream();
    return(Send(“[\”begin_scenario\"]\n"));
  }

  public int Close()
  {
    stream = client.GetStream();
    Send("[\"end_scenario\"]\n");
    return(client.Close());
   }
}
```

SPECFLOW TO WIRE



Given … quiet mode

int SetQuietMode(“on”) {}

Match:
“set quiet\’(on|off)’\”

Invoke:
idx:0, params: “on”

int set_quiet(char* state){}

COMPLIANCE CONSIDERATIONS

- Security - Anyone can connect to Wire!
- Regulation may not allow non-application code on a production system

Shut down the wire thread in production

LESSONS LEARNED

Threads & Target Architecture

Vocabulary

OPEN SOURCE

- Unit testing example

  https://github.com/ihassin/nrf51-unity

- Cucumber/Listener/RPC example

  https://github.com/ihassin/cucumber-wire-tcp2serial

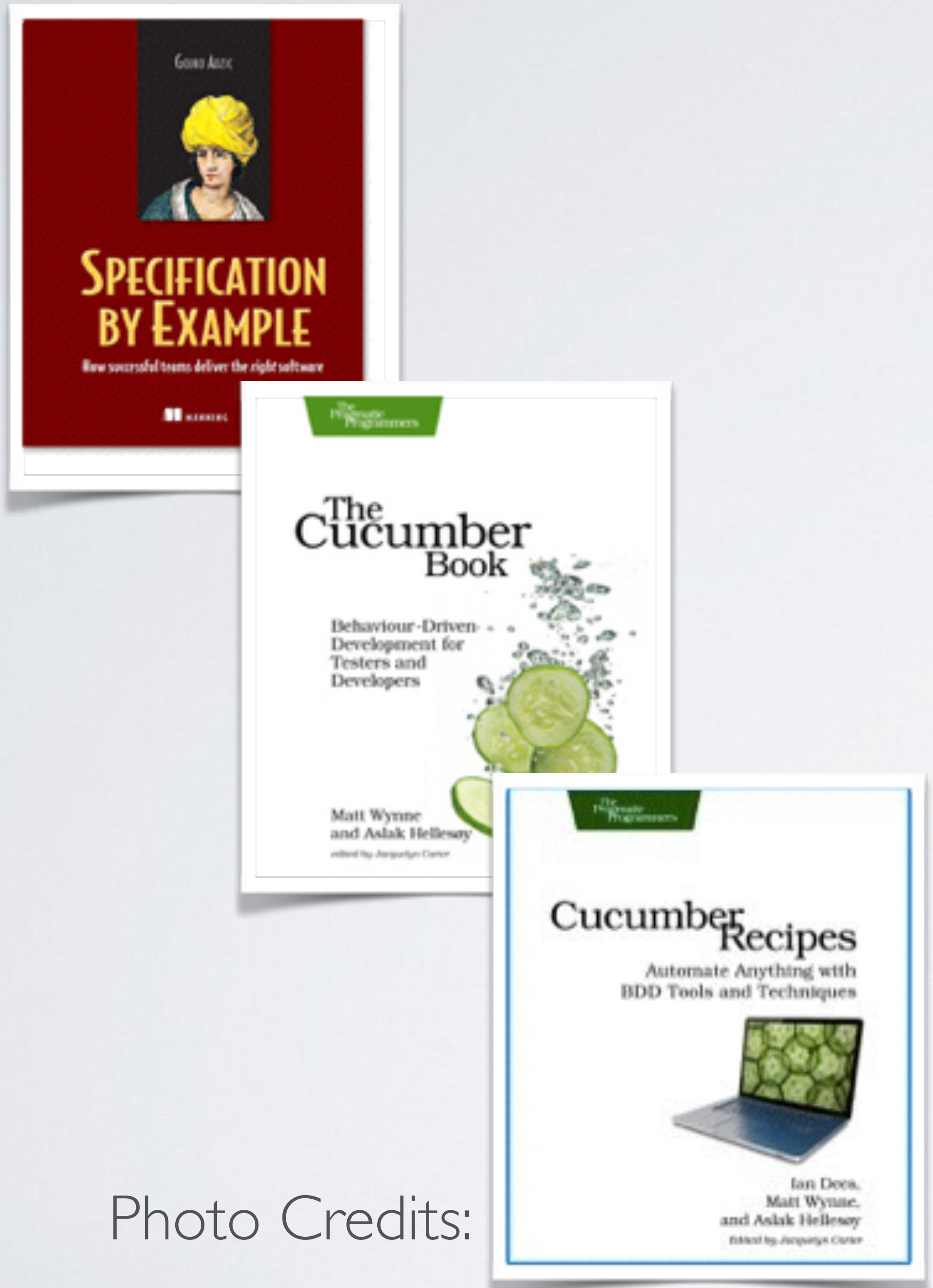
- Development environment provisioning (Linux)

  https://github.com/ihassin/fruitymesh-ubuntu-vm

- Development environment provisioning (OS-X)

  https://github.com/ihassin/fruitymesh-mac-osx

REFERENCES


Specification by Example
How successful teams deliver the right software
Manning
The Cucumber Book
Behaviour-Driven Development for Testers and Developers
Matt Wynne and Aslak Hellesøy
Cucumber Recipes
Automate Anything with BDD Tools and Techniques
Ian Dees, Matt Wynne, and Aslak Hellesøy

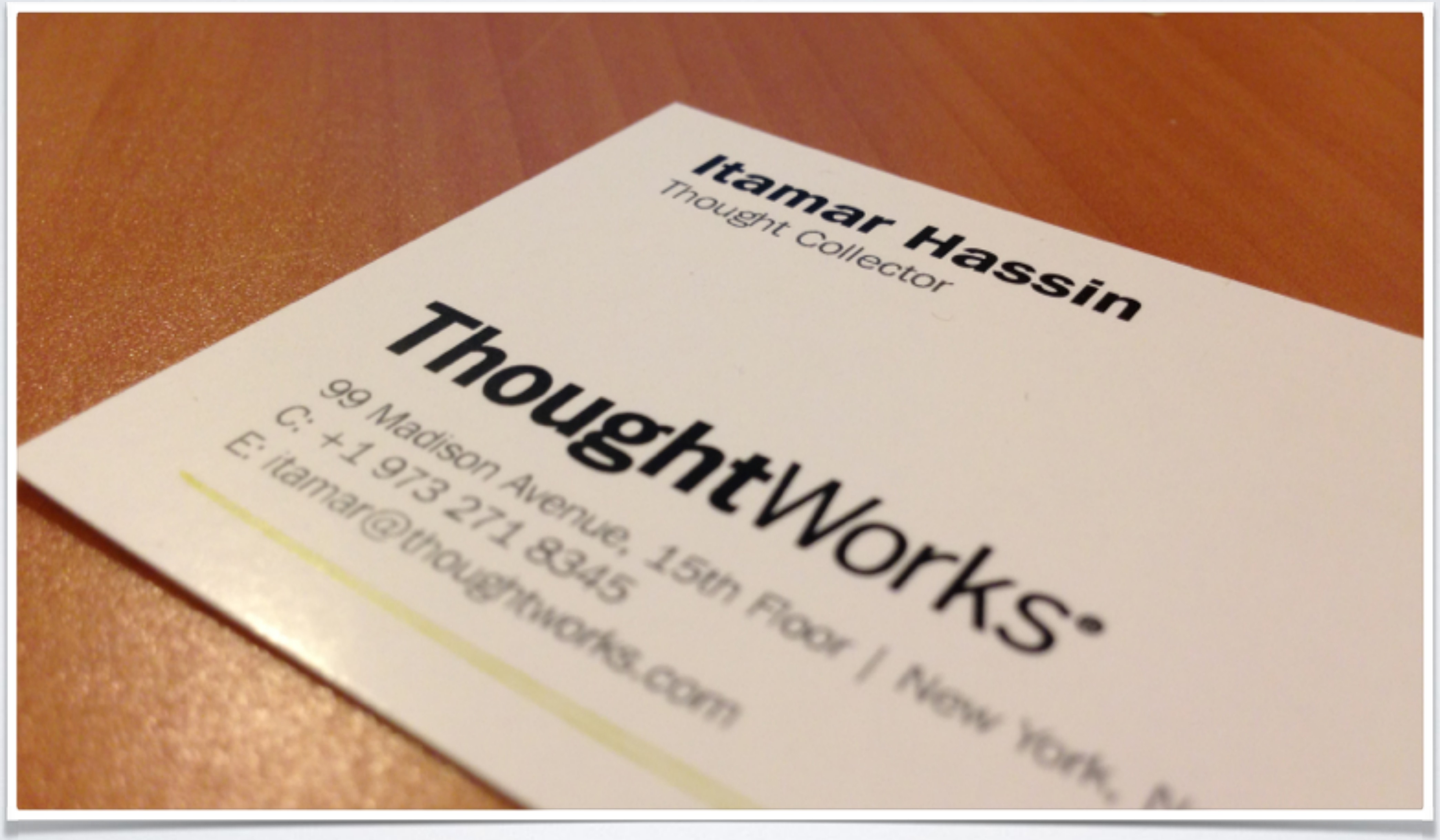
- Unity
- Cucumber
- Specification by example
- The Cucumber Book
- Cucumber Recipes
- SpecFlow
- Nordic Semiconductor

Photo Credits:
@history_pics/@historyinpics
Jim Reese#Wikipedia
National Library of Australia

THANK YOU!

@itababy

www.in-context.com



Itamar Hassin
Thought Collector

ThoughtWorks®

99 Madison Avenue, 15th Floor | New York, N
C: +1 973 271 8345
E: itamar@thoughtworks.com