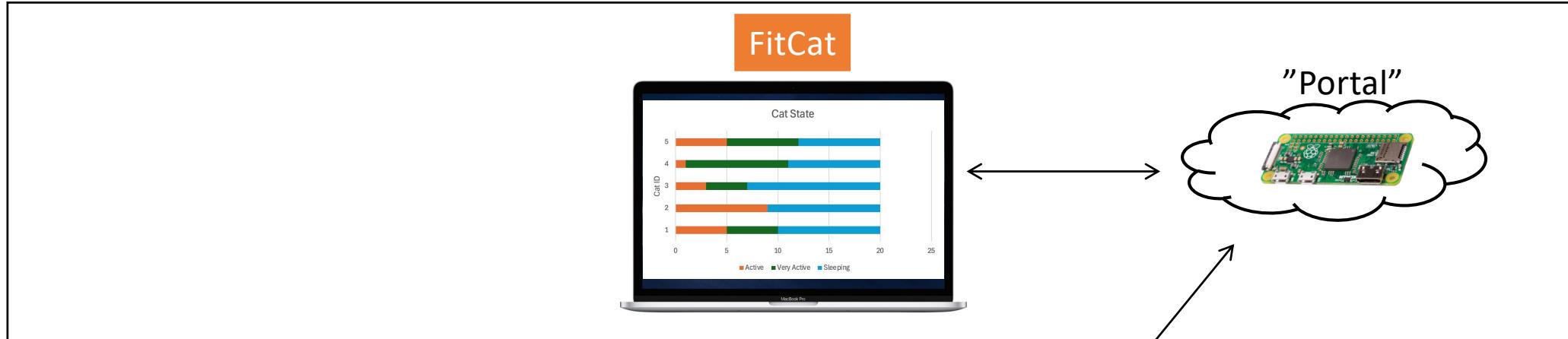


BU-EC444

Fall 2024
Prof. Little

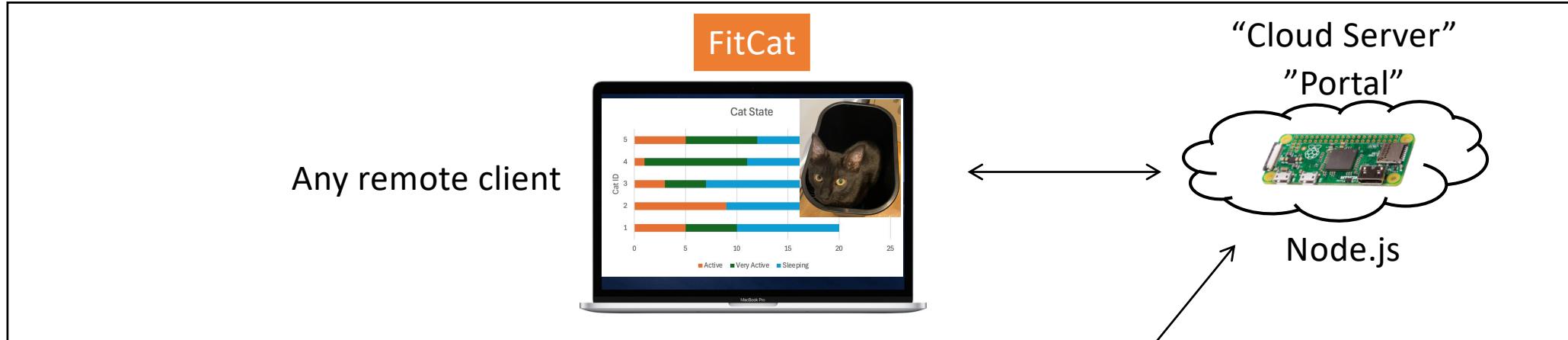
Review of Quest: FitCat – Activity Tracking for Cats

Quest 3: FitCat Concept

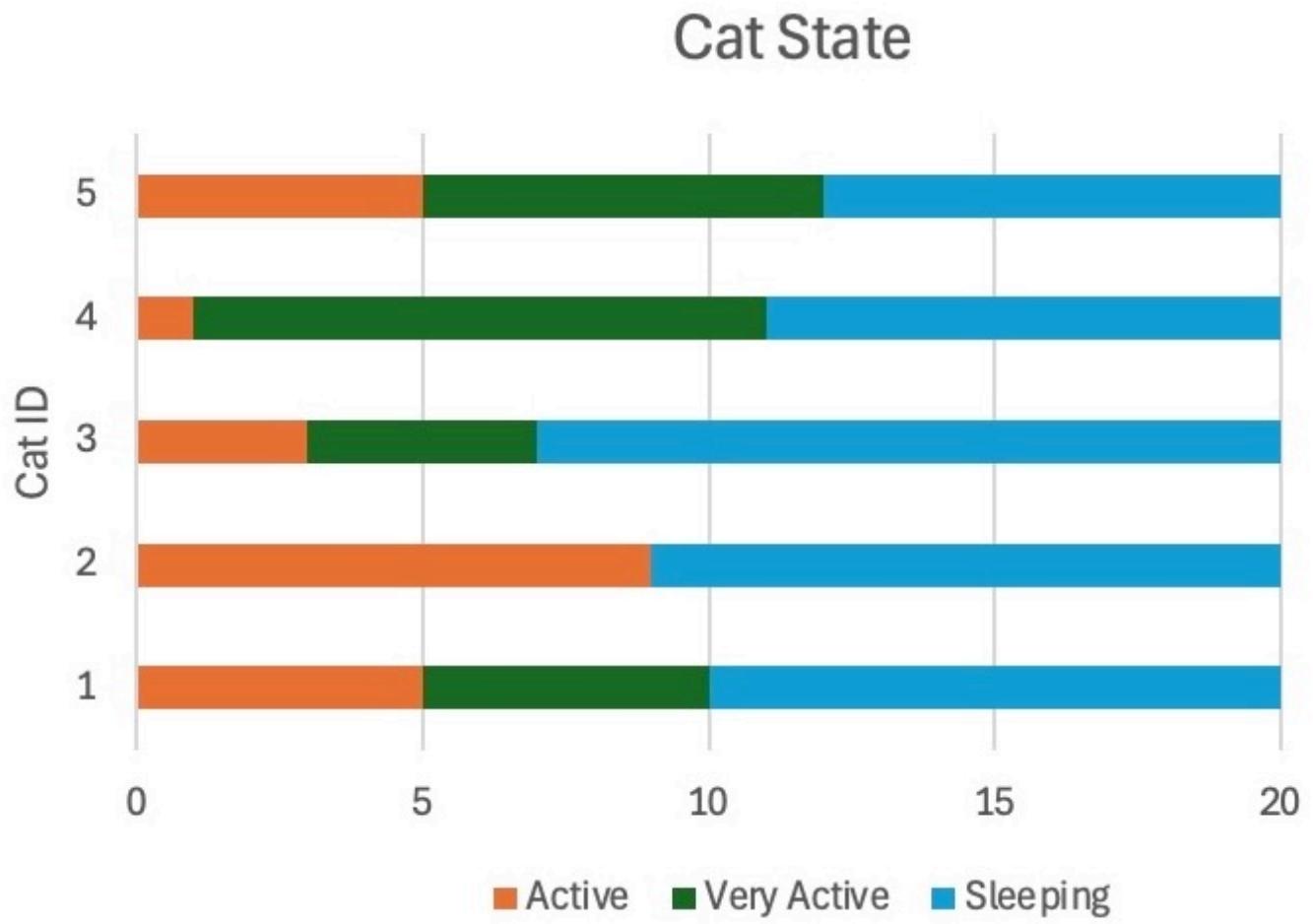
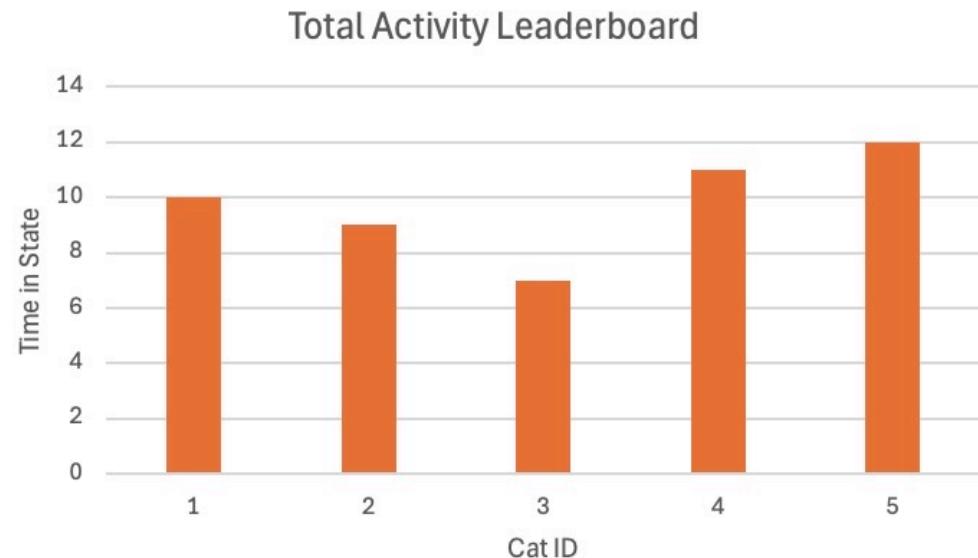
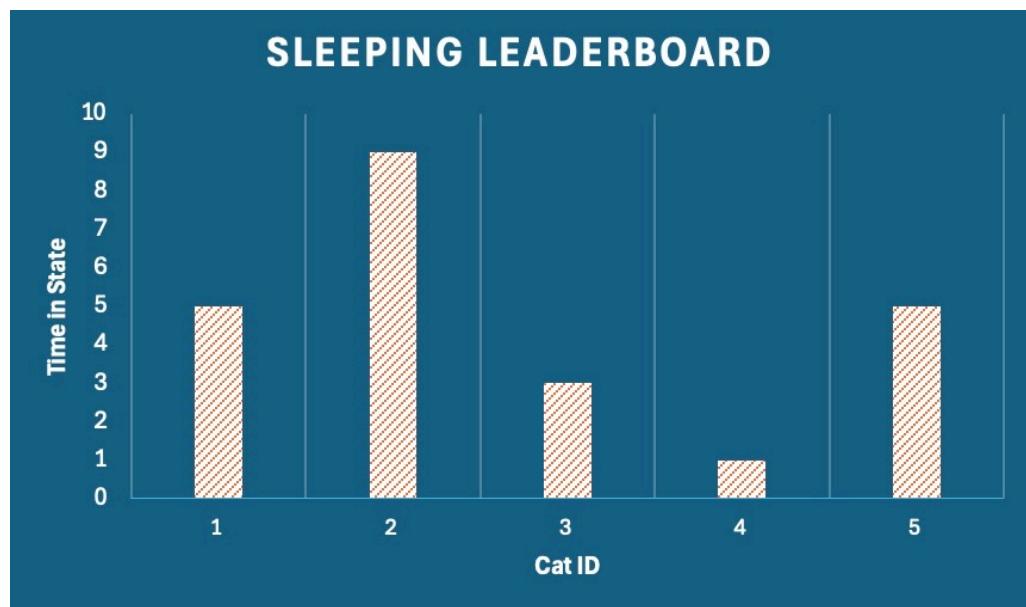


Engages multiple Cat Trackers (and Cats)

Quest 3: FitCat

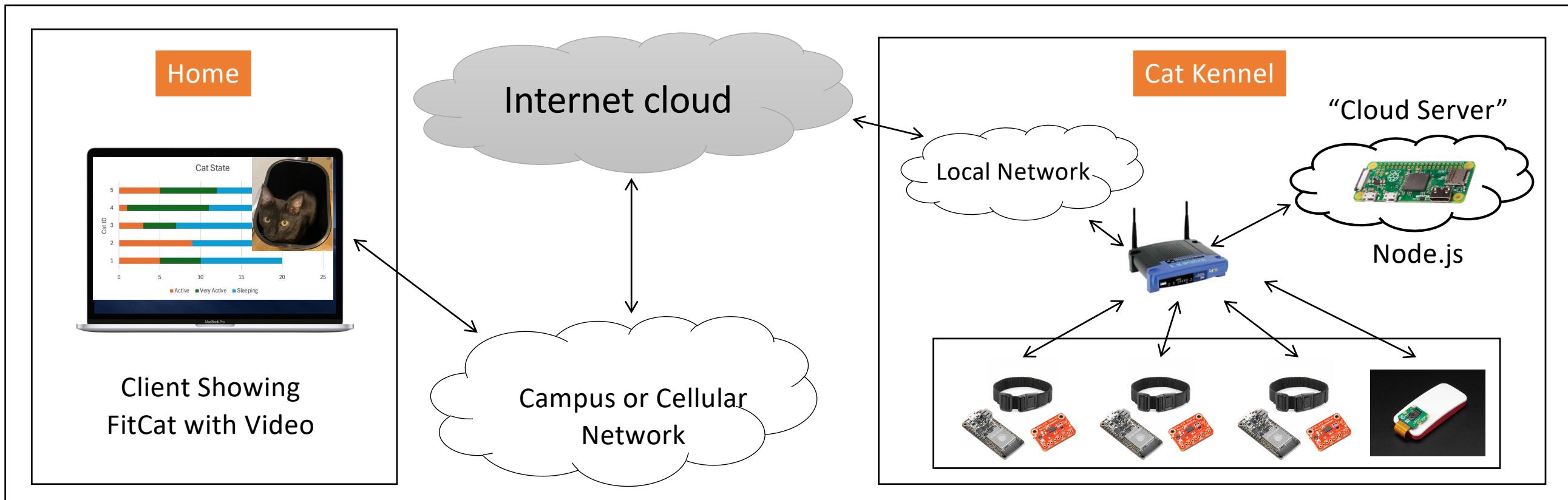


Quest 3: FitCat – some ideas for data display



Nice to have: leader floats to top

Quest 3: Hosting the FitCat Social Platform



Key takeaways:

- Data aggregation
- Remote IP control
- Wireless embedded system
- Webcam

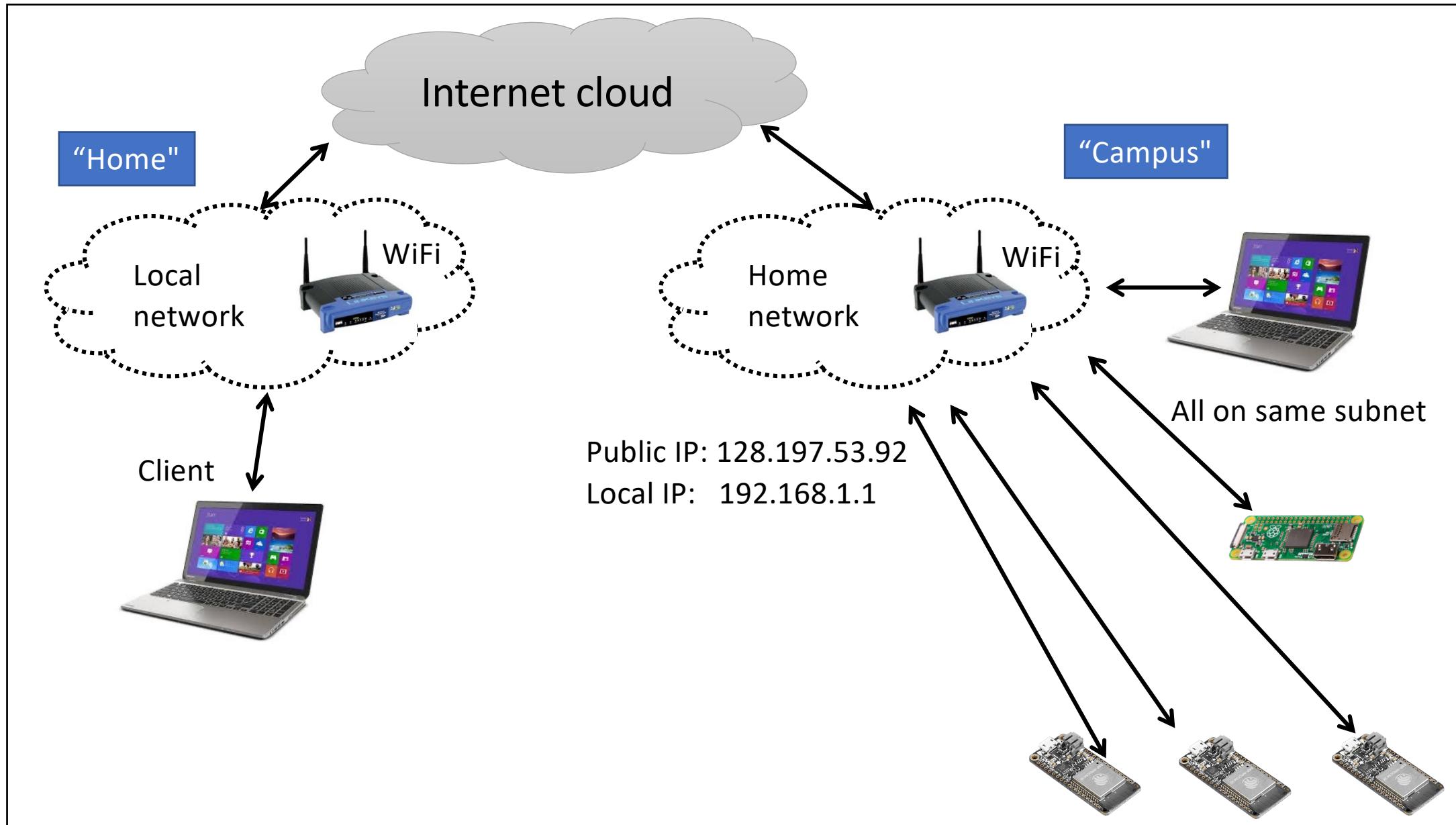
- Multiple ESPs
- Nodejs on pi
- Video from pi added
- WiFi
- Hook to future database

Quest 3 Cluster

- **Router Configuration (team)**
- **DDNS on Router (team)**
- **WiFi-ESP-Station**
- **Client and Server with UDP**
- **Raspberry Pi and Node.js**
- **Pi Web Cam**

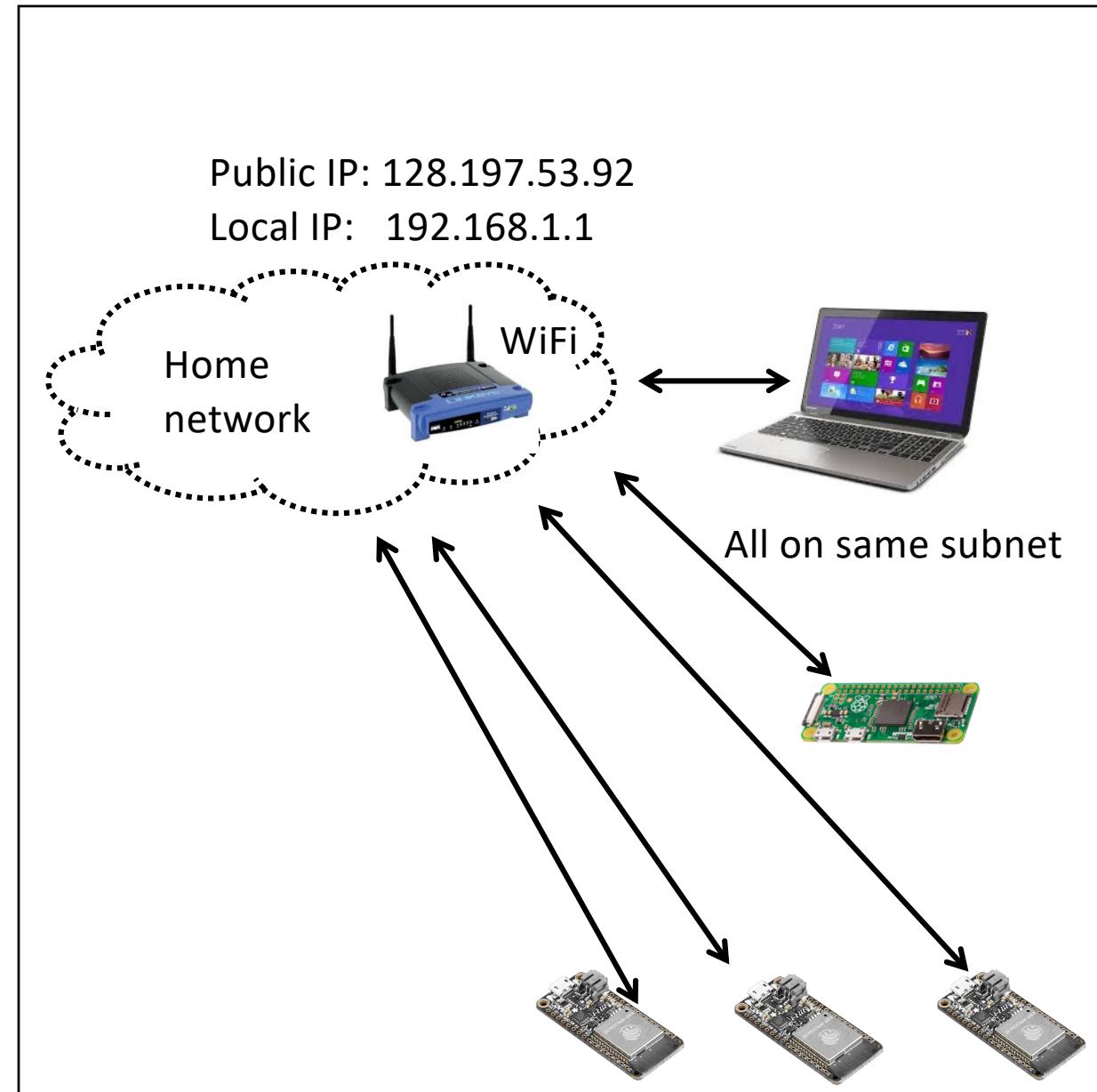
Router Configuration

Typical Wireless Setup



Typical Wireless Setup

- Router serves to create a **local subnet with its own IP address group**
- (Different from a **switch** which just **replicates ports** on same subnet)
- Can serve as firewall, limiting types of traffic
- Supports many different functions (VPNs, bit torrent etc.)
- Web access typically at :80 or :8080
- DHCP – **assigns local IP address** to local machines



Two router types – WRT54GS and E1200 v2



Router details

- WAN port
- Local ports
- Reset button
- Antennae
- Typical access address of 192.168.1.1
- UID: root, PWD: admin

Attach **your computer** to one of the **LAN ports**

Flash the router firmware using the existing admin interface of the router.

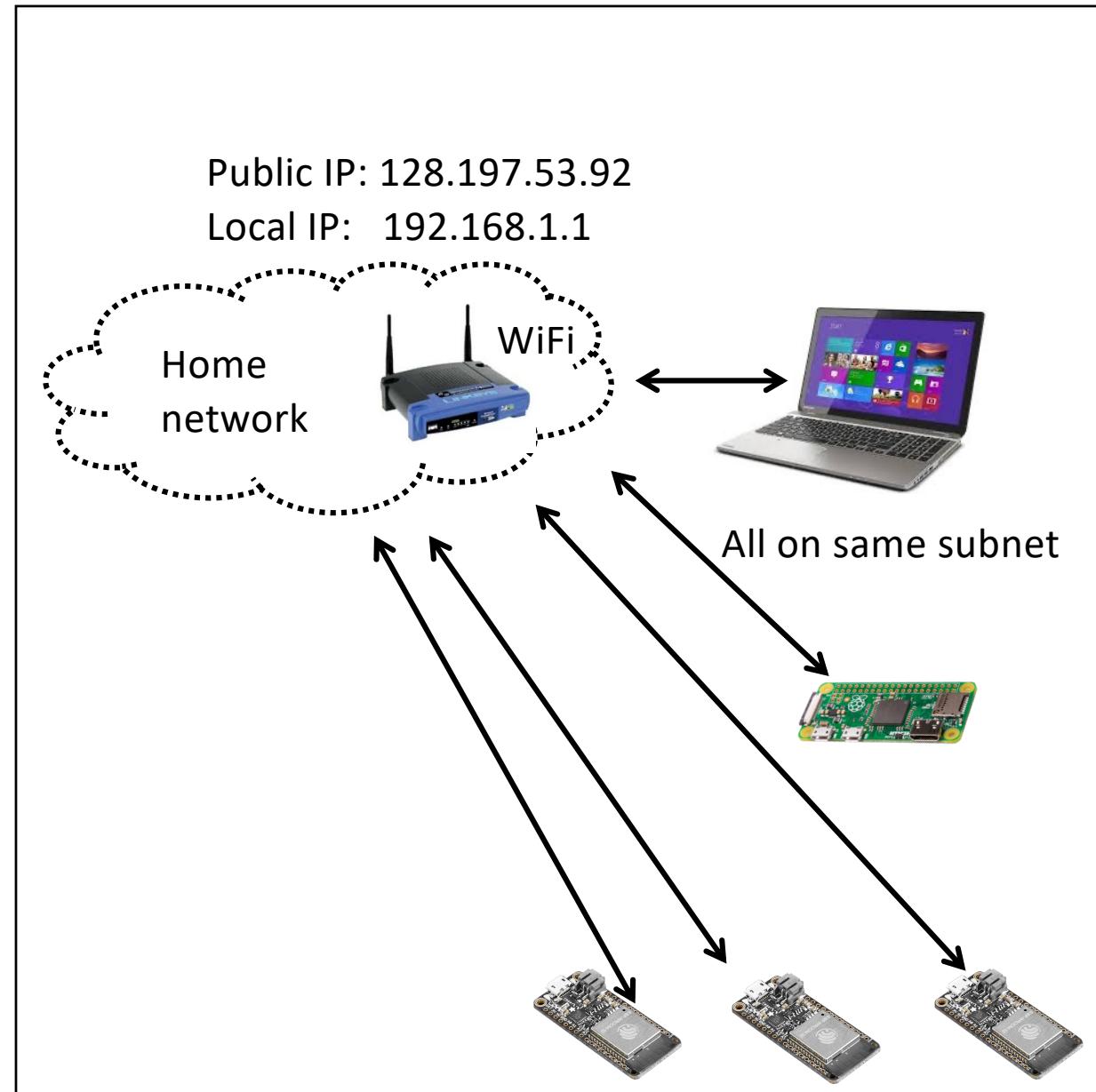
Attach **LAN port** to a **campus LAN port**

Power up. Press and hold reset button for 5s to reset password

Log in with default credentials

Typical Wireless Setup: Configuration Settings

- Local subnet of 192.168.1.1 (or similar)
- Enable local DHCP server (should be set up already)
- Set SSID with your group number (e.g., 'Group01')
- Set security to WPA/WPA2 Personal
- Set shared key to be 'smartsys'
- Enable NTP server (separate step)
- Enable Dynamic DNS as per (separate step)



Opening screen (<http://192.168.1.1> from LAN port or WiFi)

To identify devices on the local LAN

Most of the important configuration menus

Port forwarding

FreshTomato
Version 2023.1 on Linksys E1200 v2.0

!! Attention !!
You did not configure TomatoAnon project setting.
Please go to TomatoAnon configuration page and make a choice.

E1200 | Wiki

System (Hide)

Name	E1200
Model	Linksys E1200 v2.0
Bootloader (CFE)	--
Chipset	Broadcom BCM53572 chip rev 1 pkg 8
CPU Frequency	300 MHz
Flash Size	8 MB

Time Not Available
Uptime 0 days, 00:05:49
CPU Load (1 / 5 / 15 mins) 0.30 / 0.15 / 0.05
Used / Total RAM 9,948.00 KB / 28.14 MB (34.52%)
Used / Total NVRAM 35.34 KB / 64.00 KB (55.22%)

Wireless Temperature eth1: 2.4G - 43°C / 109°F

Ethernet Ports State

WAN0  100Mbps Full	LAN0  Unplugged	LAN1  Unplugged	LAN2  Unplugged	LAN3  Unplugged
--	---	---	---	---

WAN0 (Hide) » Configure

More opening screen (status)

WAN address from ISP

WAN0 (Hide)	
MAC Address	30:23:03:C7:9F:66
Connection Type	DHCP
IP Address	128.197.164.92
Subnet Mask	255.255.255.192
Gateway	128.197.164.65
DNS	128.197.253.183:53,128.197.253.120:53,128.197.253.254:53
MTU	1500
Status	
Status	Connected
Connection Uptime	0 days, 00:06:31
Remaining Lease Time	0 days, 00:53:29
Renew Release	

SSID

LAN (Hide)	
Router MAC Address	30:23:03:C7:9F:56
Router IP Addresses	br0 (LAN0) - 192.168.1.1/24
DHCP	br0 (LAN0) - 192.168.1.2 - 192.168.1.50

WiFi channel

Wireless eth1 (wl0) / 2.4 GHz (Hide)	
MAC Address	30:23:03:C7:9F:58
Wireless Mode	Access Point
Wireless Network Mode	Auto
Interface Status	Up (LAN0)
Radio	Enabled
SSID	Grouper
Broadcast	Enabled
Security	WPA2 Personal (PSK) + TKIP / AES
Channel	6 - 2.437 GHz
Channel Width	20 MHz
Interference Level	Severe

Device List

To identify devices on the local LAN

FreshTomato
Version 2023.1 on Linksys E1200 v2.0

E1200 | Wiki

Device List

Interface	Media	MAC Address	IP Address	Hostname	RSSI	Quality	TX/RX Rate	Lease
WAN0 (vlan2) DHCP	[DR] [BWL] [AR]	00:C1:64:C7:D5:40	128.197.164.65	mary008-0202net-gw				
LAN0 eth1 (wl0) Grouper	[DR] [BWL] [AR]	E0:B5:5F:F1:66:CE 2.4	192.168.1.34	Thomass-MBP	100	117 / 144	0 days, 23:51:52	

Noise eth1 (wl0) / 2.4 GHz : -99 dBm

Network Discovery mode

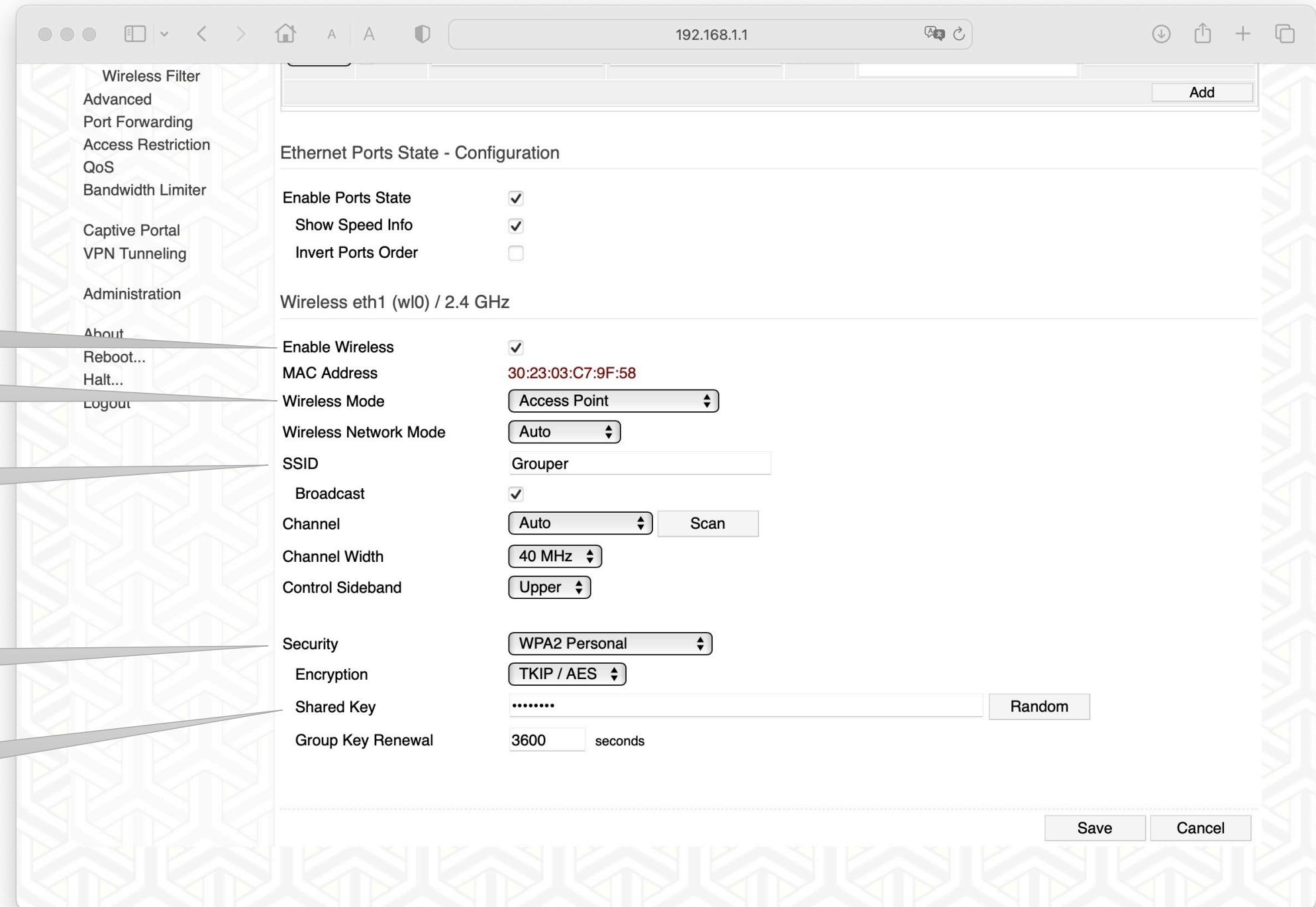
3 seconds

IP assigned from BU

IP assigned to laptop by router

The screenshot shows the FreshTomato web interface for a Linksys E1200 router running version 2023.1. The main page title is "FreshTomato Version 2023.1 on Linksys E1200 v2.0". On the left, there's a sidebar with various status and configuration links. The main area is titled "Device List" and displays a table of network interfaces. The table has columns for Interface, Media, MAC Address, IP Address, Hostname, RSSI, Quality, TX/RX Rate, and Lease. Two rows are visible: one for WAN0 (vlan2) with IP 128.197.164.65 and one for LAN0 eth1 (wl0) with IP 192.168.1.34. A callout box labeled "IP assigned from BU" points to the WAN0 row, and another labeled "IP assigned to laptop by router" points to the LAN0 row. A third callout box labeled "To identify devices on the local LAN" points to the sidebar.

Basic -- wireless



Login credentials

Logout

Authorized Keys

Telnet Daemon

Enable at Startup

Port

Admin Restrictions

Allowed Remote IP Address
(optional; ex: "1.1.1.1", "1.1.1.0/24", "1.1.1.1 - 2.2.2.2" or "me.example.com")

SSH / Telnet

every seconds

Username / Password

Username (empty field means "root")

Password
(re-enter to confirm)

Notes (Show)

Update router login:
UID: root PWD: smart

Enable router login access constraints

Enable wireless access
on port 80

Ways to make more
secure:

- HTTPS
- No wireless login
- No remote access
- Limit access by certain domains, IP addresses, or time windows

The screenshot shows the FreshTomato Web Admin interface for a Linksys E1200 v2.0 router. The left sidebar lists various management options like Status, Web Usage, Bandwidth, IP Traffic, Tools, and Administration. The main panel is titled 'Web Admin' and contains the following settings:

- Local Access:** Set to HTTP (dropdown), Port 80 (input), and Allow Wireless Access (checkbox checked).
- Remote Access:** Set to Disabled (dropdown).
- UI files path:** Default: /www (dropdown). A note says: "Please be sure of your decision before change this settings!"
- Theme UI:** Set to Default (dropdown).
- Open Menus:** Status (checkbox checked), Bandwidth (checkbox checked), IP Traffic (checkbox unselected), Tools (checkbox unselected), Basic (checkbox unselected), Advanced (checkbox unselected), Port Forwarding (checkbox unselected), QoS (checkbox unselected), VPN Tunneling (checkbox unselected), and Administration (checkbox unselected).

At the bottom of the main panel, there is a note: "Warning! In some peculiar cases the content of JFFS might not be preserved after a firmware upgrade" and "keep disabled for smaller memory footprint during upgrade".

DDNS setup on router (not on Pi or laptop)

FreshTomato
Version 2023.1 on Linksys E1200 v2.0

E1200 | Wiki

Set up DDNS

Status

- Overview
- Device List
- Web Usage
- Logs

Bandwidth

- Real-Time
- Last 24 Hours

Daily

Weekly

Monthly

IP Traffic

Tools

Basic

- Network
- IPv6
- Identification
- Time

DDNS

- DHCP Reservation
- Wireless Filter

Advanced

- Port Forwarding
- Access Restriction
- QoS
- Bandwidth Limiter

Captive Portal

VPN Tunneling

Administration

Dynamic DNS Client

IP address

Use WAN0 IP Address 128.197.164.92 (recommended)

Auto refresh every 28 days (0 = disable)

Dynamic DNS 1

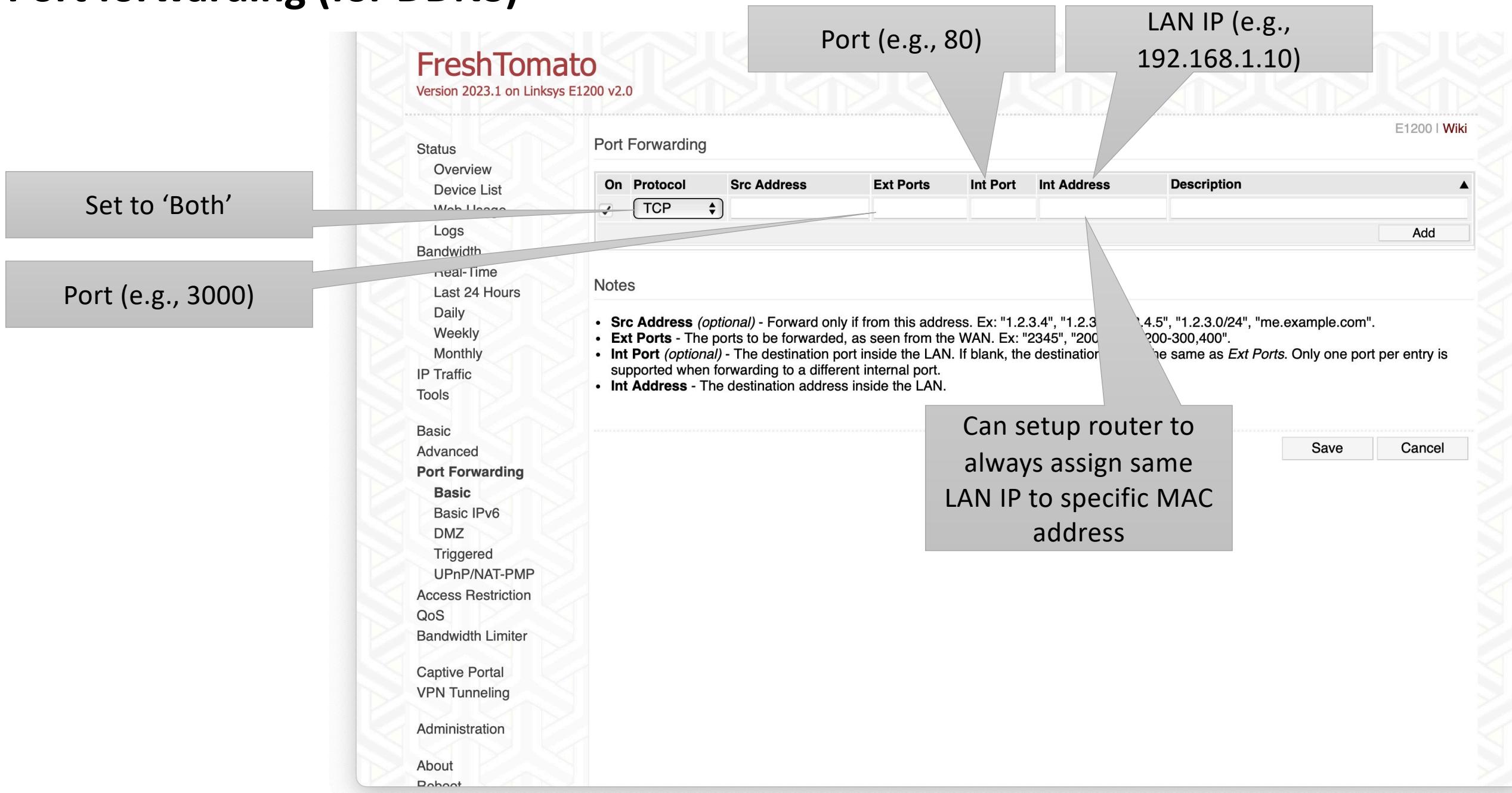
Service None

Dynamic DNS 2

Service None

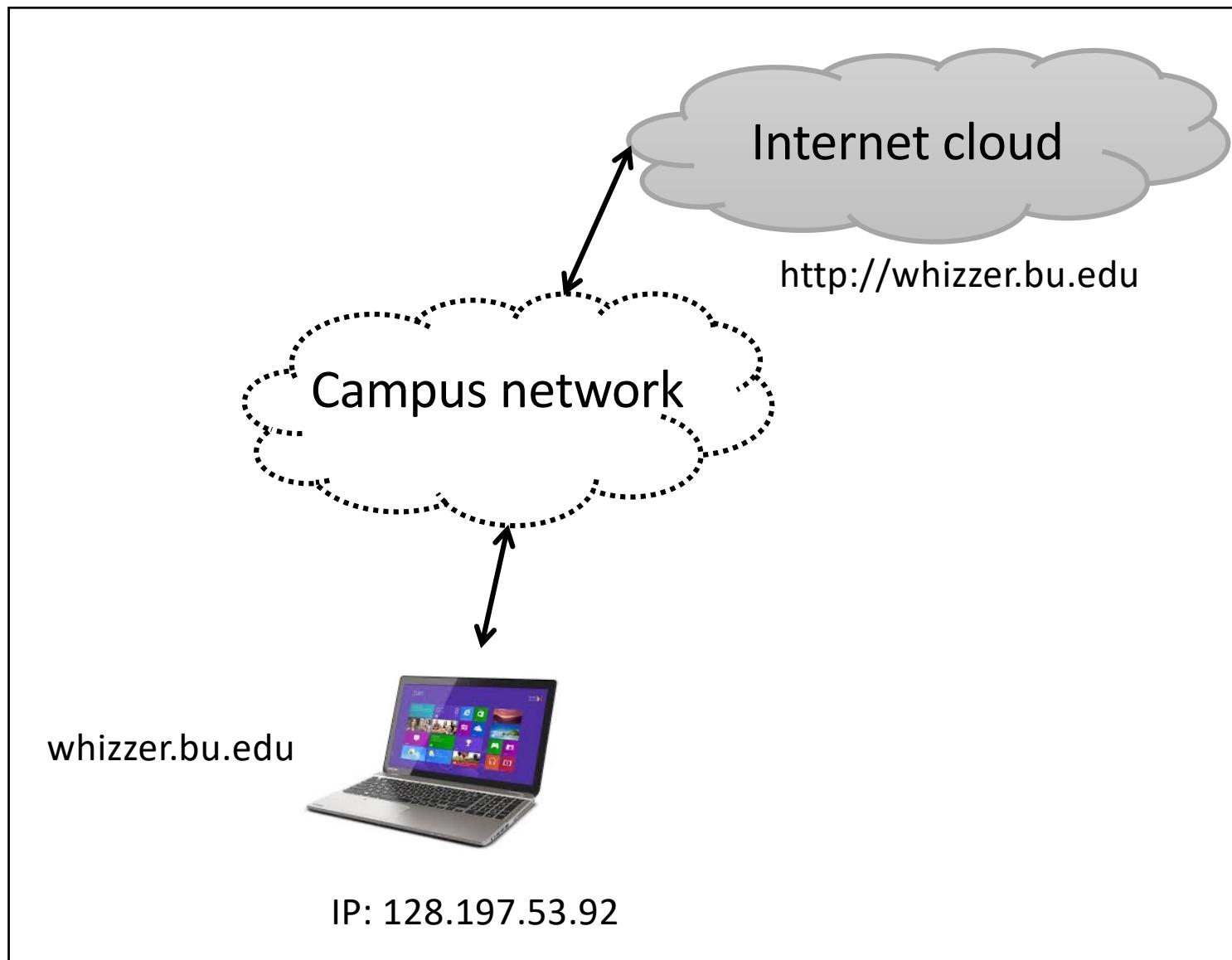
Save Cancel

Port forwarding (for DDNS)



Skills: Dynamic DNS

DNS and static address mapping – static IPs in this illustration



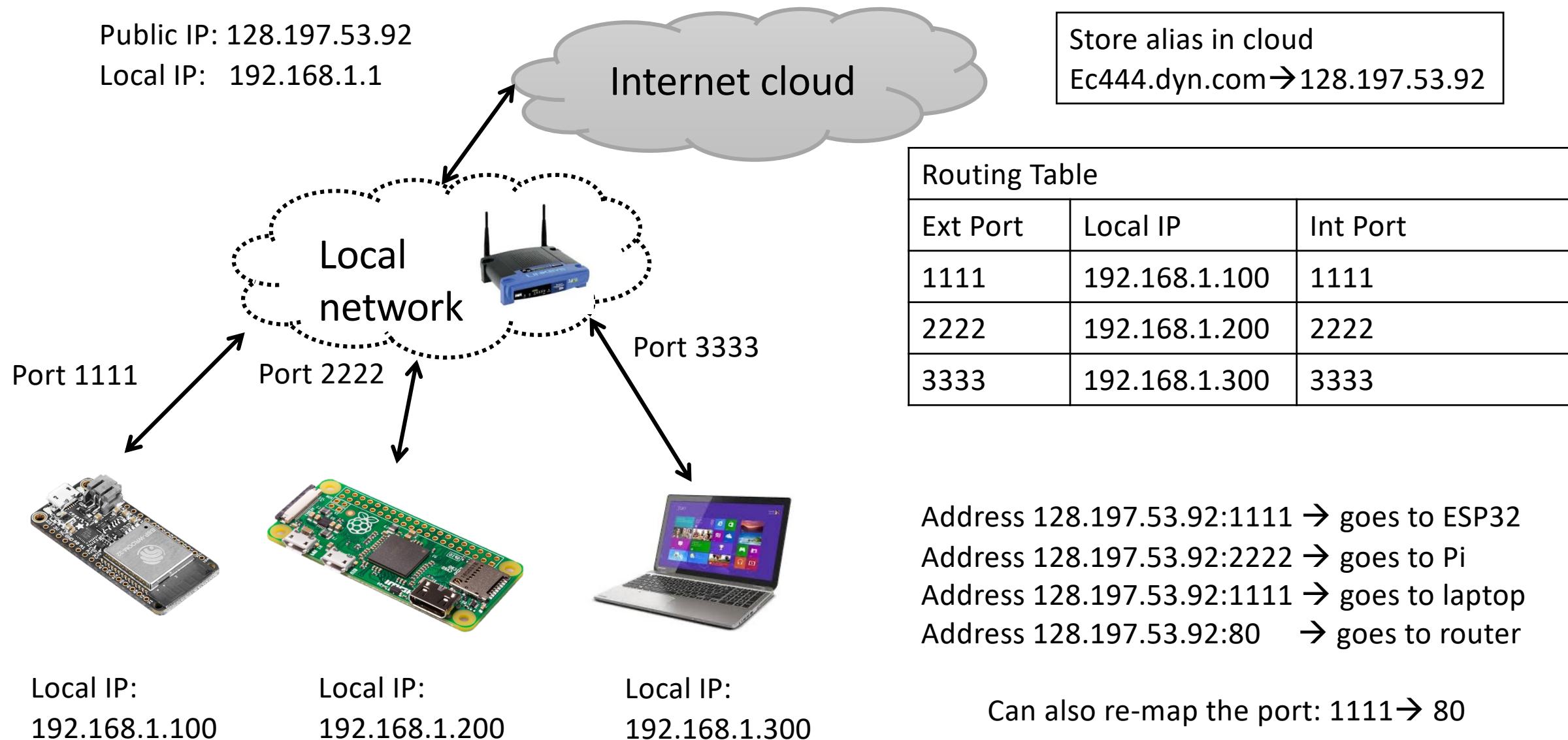
Typical static configuration

DNS Table (e.g., /etc/hosts)

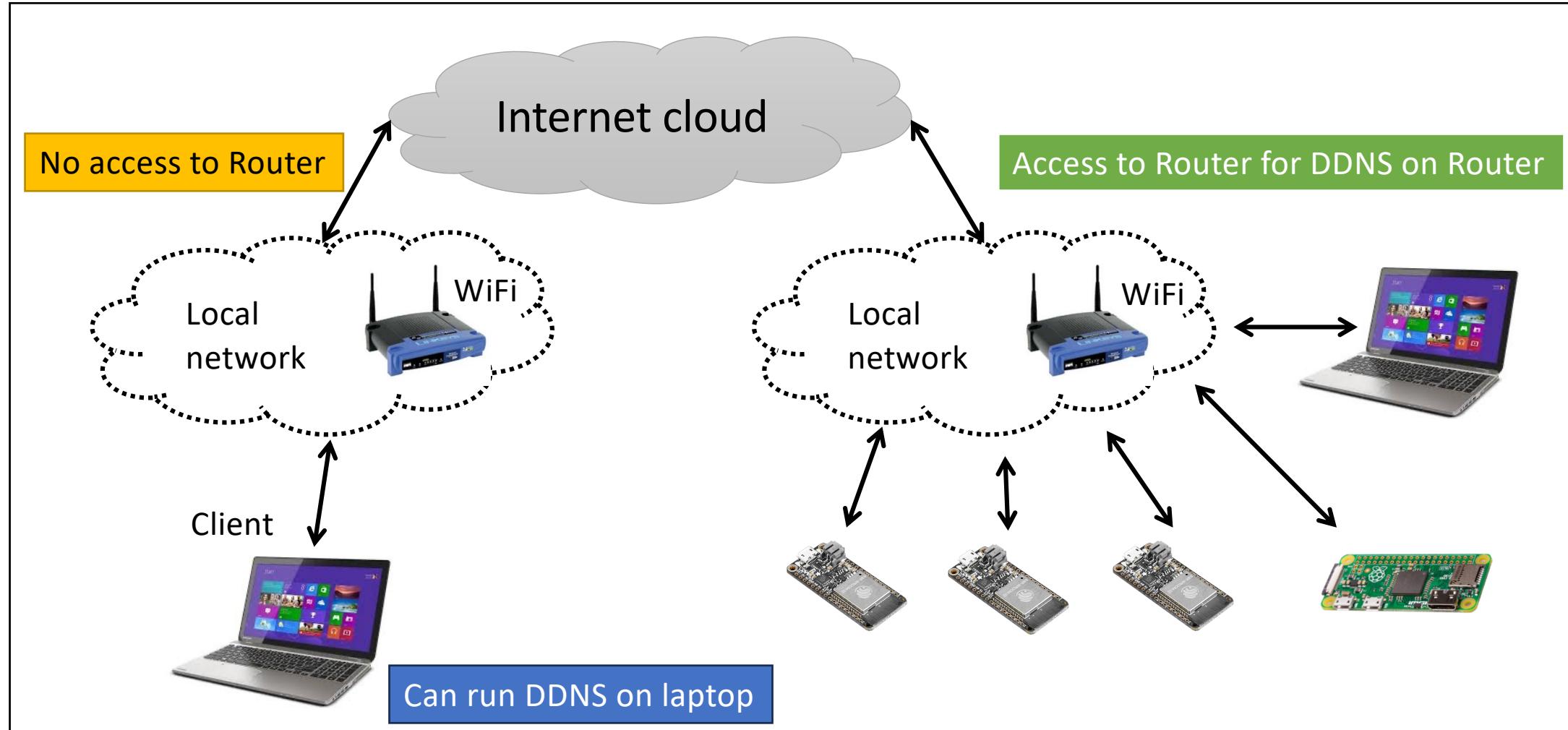
Name	IP Address
whizzer.bu.edu	128.197.53.92
hulk.bu.edu	128.197.53.89
spiderman	128.197.53.88
...	...

But if IP addresses are not static,
the table needs to be updated

Dynamic DNS – address remapping and port forwarding

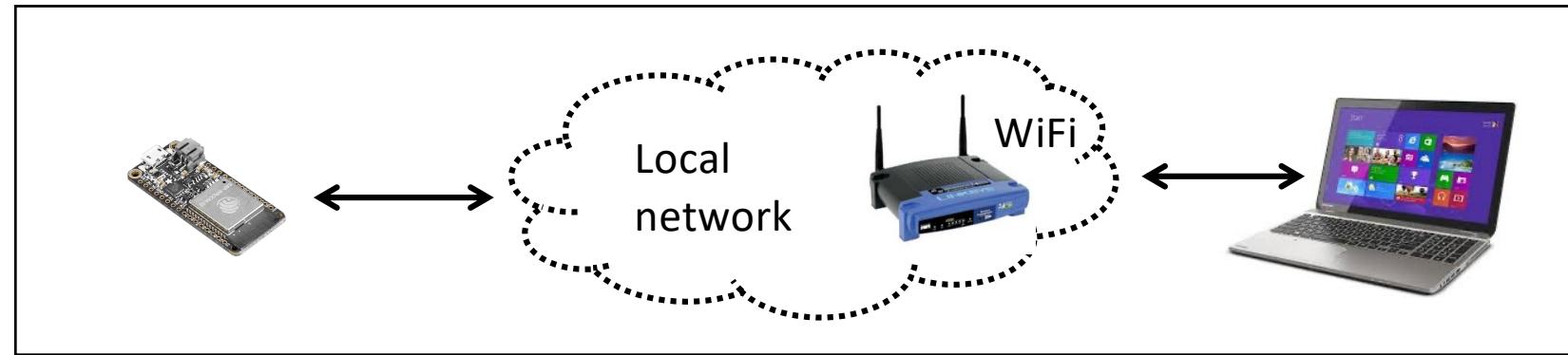


Sometimes a problem: one end can't do DDNS



Skills: WiFi-ESP-Station

Skill: WiFi for ESP -- WiFi “station” or “client”



ESP32 program

- Find SSID
- Authenticate → “connection establishment”

Task:

- Bring up ESP32 for connecting to your router
- Code: https://github.com/espressif/esp-idf/tree/master/examples/wifi/getting_started/station

Skills: Client and Server UDP with ESP32 and Node

Next: getting some data across the WiFi

- Some definitions
- **IP**: Internet Protocol (IPv4 / IPv6): rules and formats to send (route) packets in and between networks
- **Datagrams**: header and payload (packets); connectionless; order of arrival not guaranteed (no acknowledgements) not specifically reliable
- **UDP** (User Datagram Protocol): protocol for sending and receiving datagrams; connectionless, checksums, no handshaking
- **TCP** (Transmission Control Protocol): Connection-oriented and reliable (with retransmissions), flow control
- **Socket**: enables application multiplexing (IP address : port)
- **Port**: communication endpoint – binding point between service and network; a number

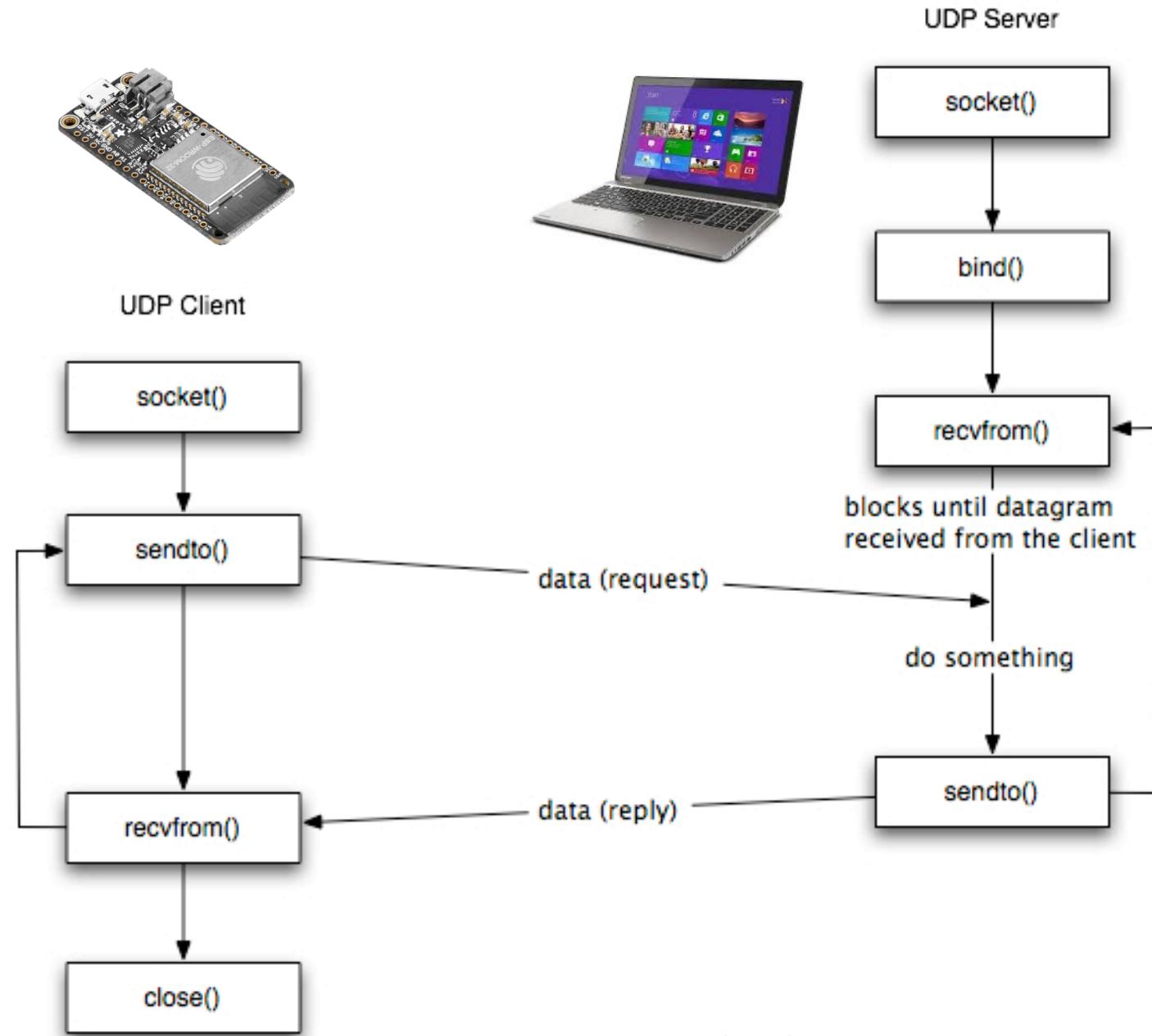
UDP enabled with a client and a server

Essentials

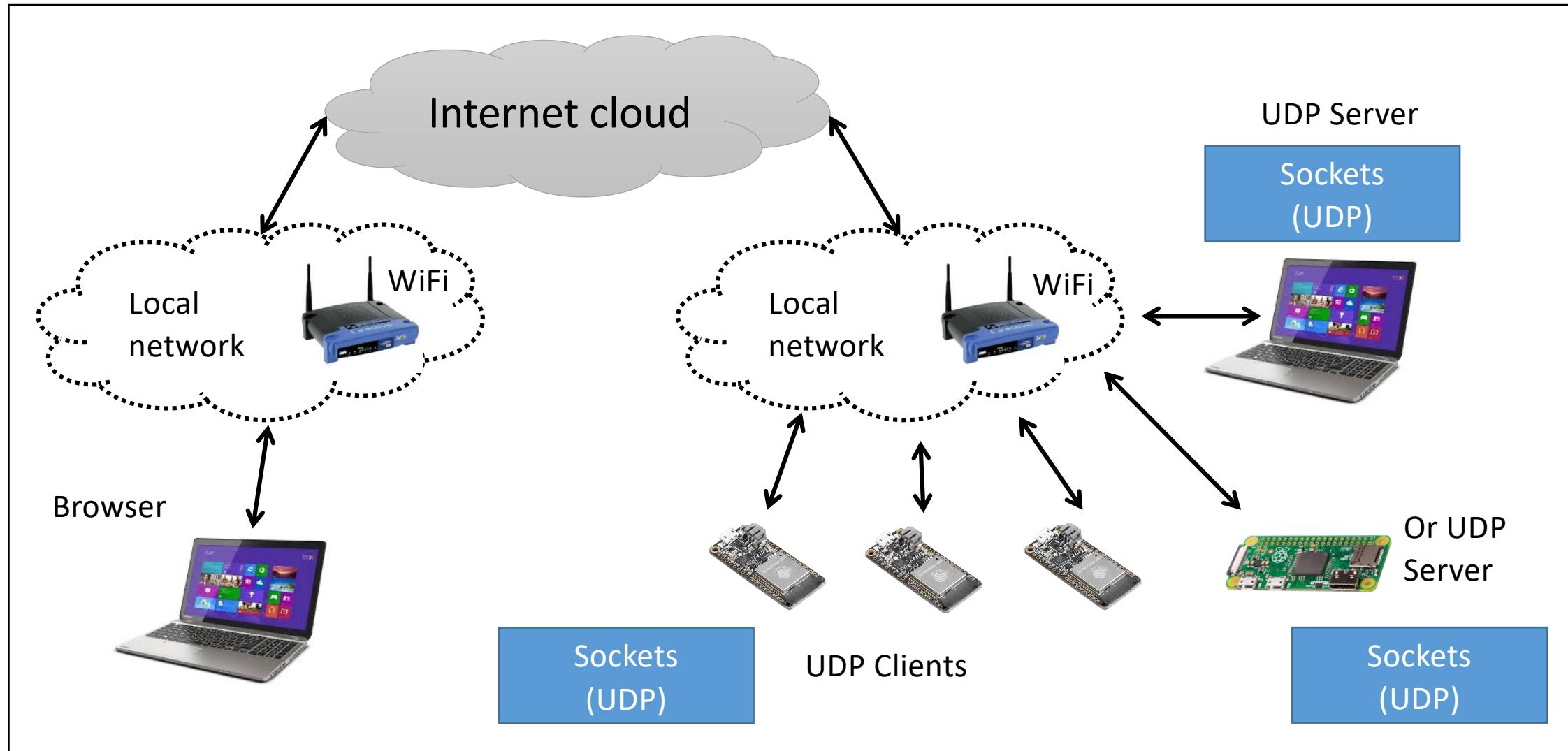
- Client and server aware of IP address and port numbers
- Server set up to receive and respond
- Client set up to request and receive

Client initiates in this model

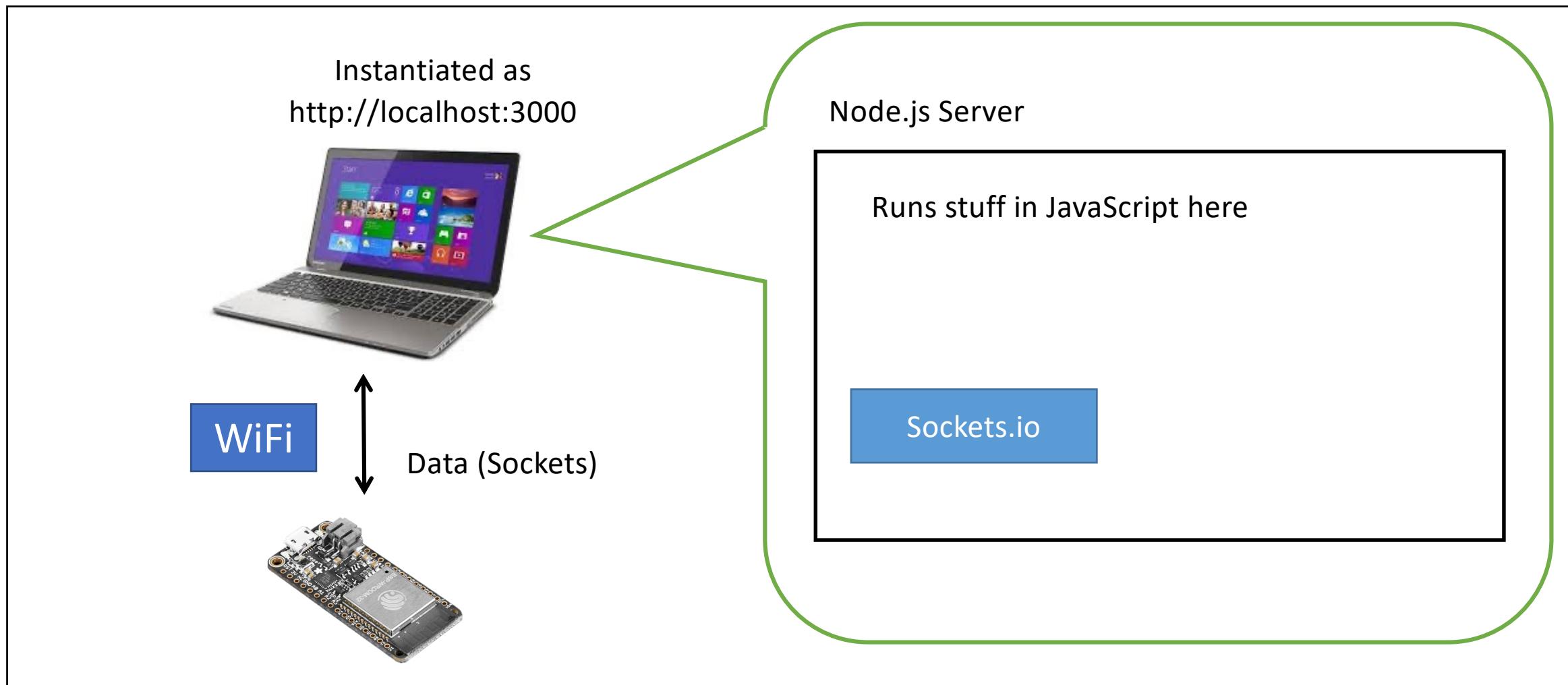
Also in node.js socket model



In our context



Data Flow – node.js and comms



***Not the only way to do it

UDP enabled with a client and a server

Skill Part 1:

- Read **design pattern** for UDP Sockets
- Bring up a UDP server in the node app
- Bring up a UDP client at the ESP32 including WiFi
- The ESP32 should send a message requesting the blink time from the laptop (node server)
- Then node server responds with a new blink time
- The ESP32 receives the blink time and flashes the onboard LED

Skill Part 2:

- Bring up a UDP client app on the node server
- Bring up a UDP server at the ESP32 including WiFi
- The node app should send a message to change the status LED on the ESP32
- The ESP32 changes the LED and then acknowledges the change back to the node server.

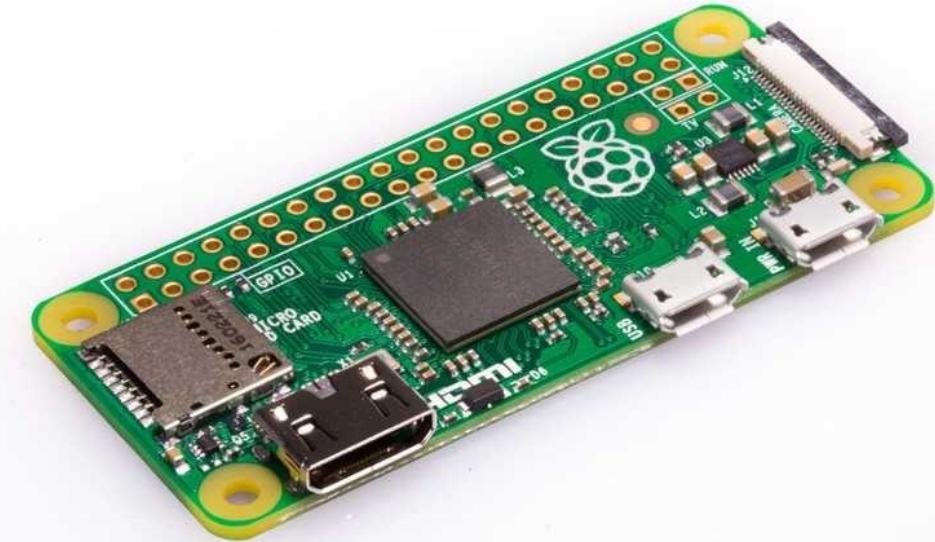
<https://github.com/BU-EC444/01-EBook/blob/main/docs/briefs/design-patterns/dp-sockets.md>

Skills: Raspberry Pi and Node.js

Raspberry Pi Zero

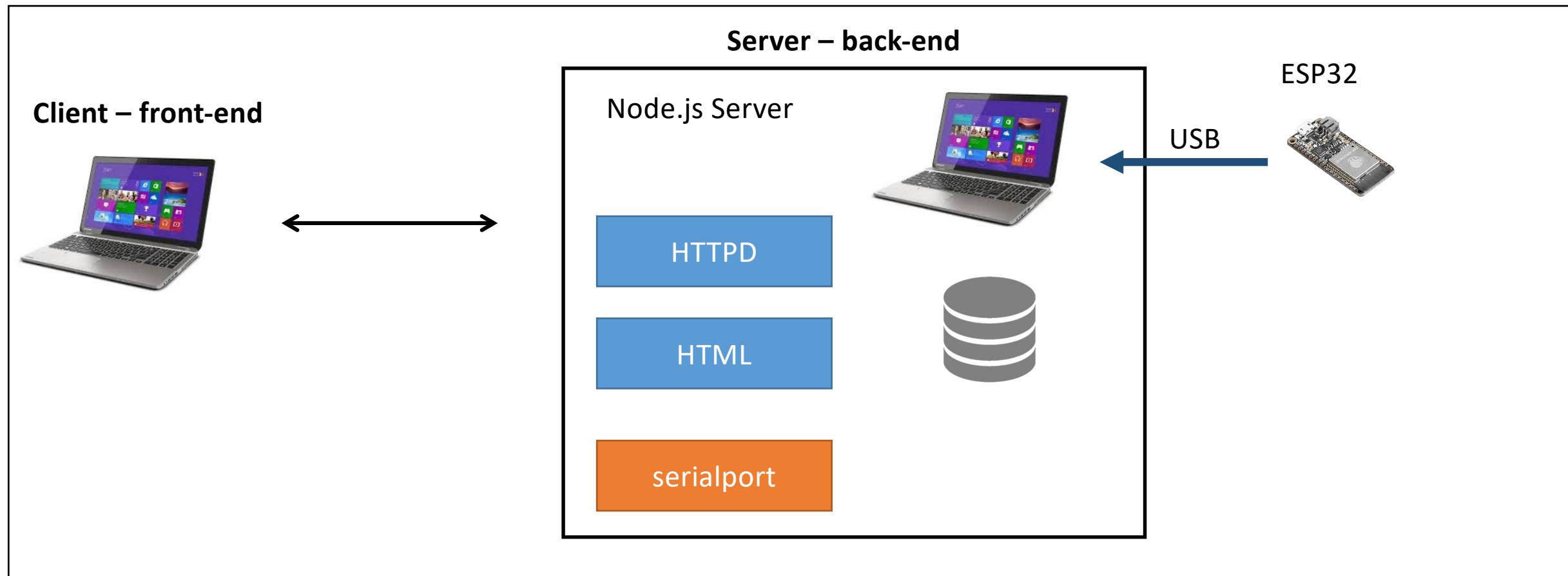
- 1GHz single-core CPU
- 512MB RAM
- Mini HDMI port
- Micro USB OTG port (on the go – for ext drives)
- Micro USB power
- HAT-compatible 40-pin header
- Composite video and reset headers
- Camera connector

From the Pi zero web site

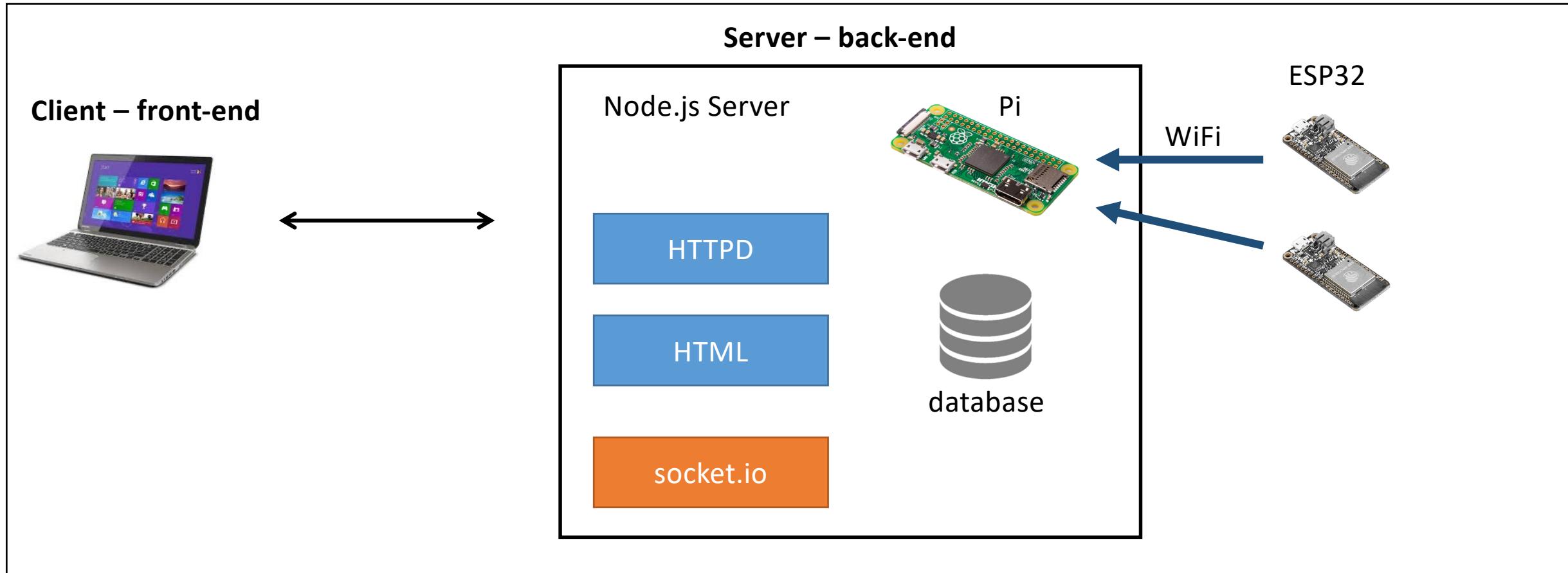


We use this primarily as
an embedded server,
not for the I/O

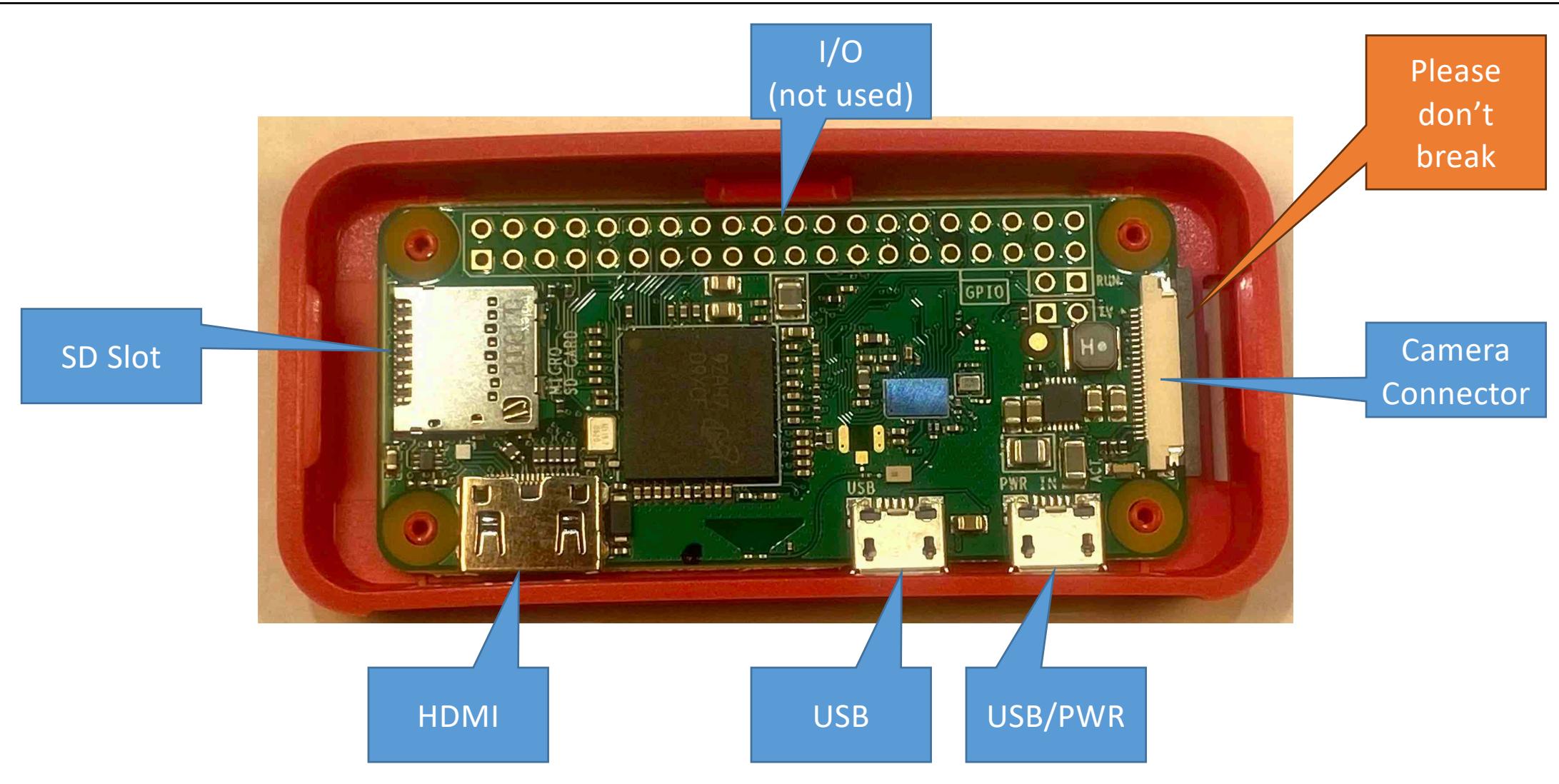
This is what we previously did for hosting data from the ESP32



Then move to WiFi and Pi

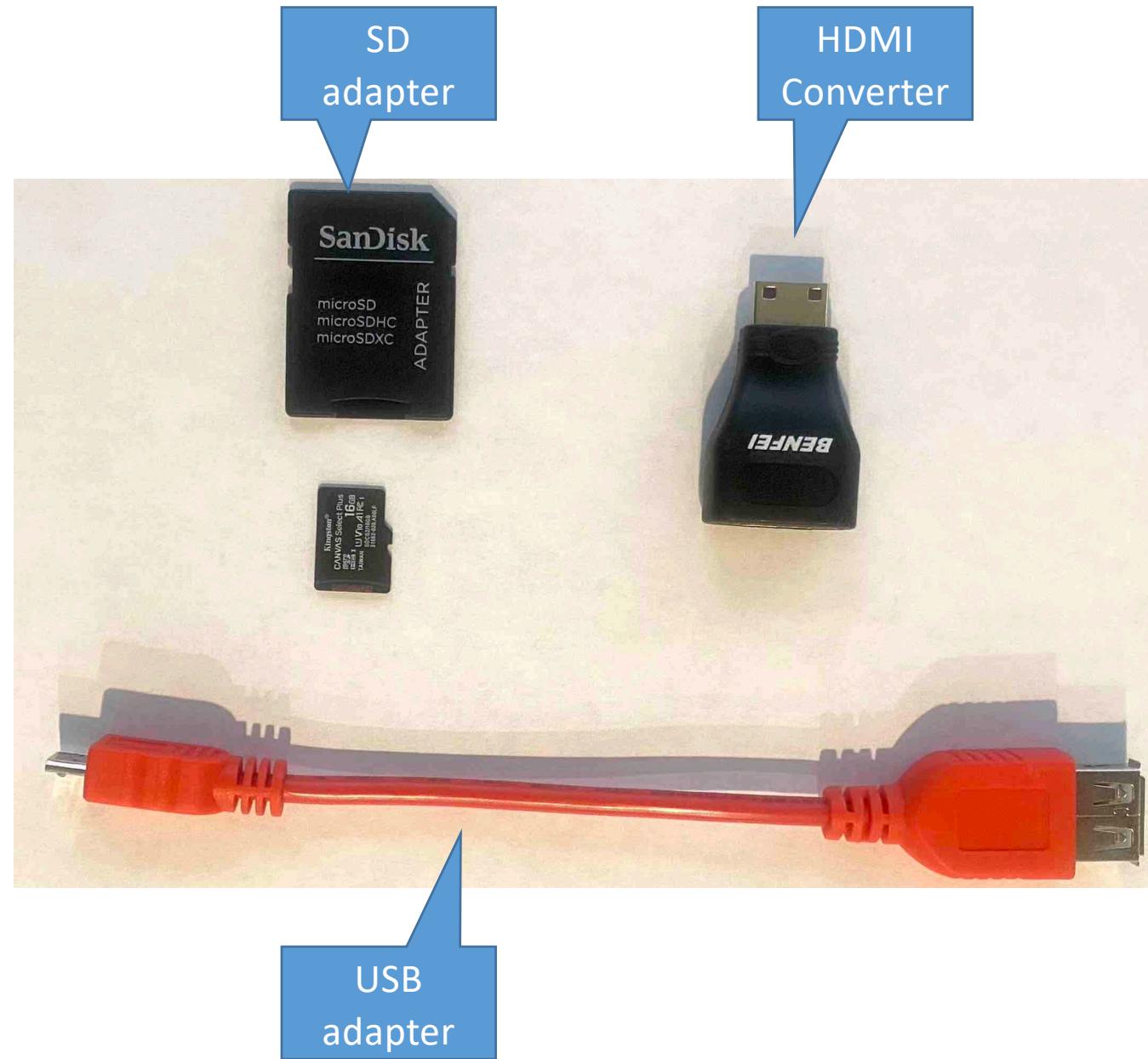


Notes – Navigating the Pi Zero board



Notes – Pi Zero kit

- Each kit has a 16GB SD card
- Micro SD to SD adapter
- HDMI Mini converter (or cable)
- Mini USB to USB adapter cable



Bringing up the RPi – skill part (1)

- Download the Raspberry Pi Imager.
- Image your SD card with the Raspberry Pi OS
- Use the menu under the gear for setting SSID/PWD and UID/PWD for your router (Group_X/smartsys) and pi (pi/smart)
- You are doing a “headless” install – without keyboard, mouse, or monitor
- Unplug your Pi, insert the SD card you just flashed, and plug the pi into power (do not unplug until it is done)
- After boot (could be 5 min), you should see the pi appear on your local network (log into the router and look at the **device list**)
- Now you can open a command prompt and `ssh pi@ip-address-of-pi`
- UID pi, PWD smart

* <https://www.raspberrypi.com/documentation/computers/configuration.html#configuring-networking>

Flashing the drive from laptop



Brings up menu
to configure SSID,
UID/PWD

Bringing up the pi – skill (part 2)

Install node.js and npm on pi – follow instructions on skill assignment

- This method gets the latest distribution for the ARMv6 used on our pi and does installation
- Bring up ‘hello world’ using node (from earlier skill, just now on pi)

Configure git on your pi to pull code from your repo

- Set up your pi to pull code from your EC444 git repo
- For example: (on the pi)

```
git clone https://github.com/BU-EC444/myrepo
```

- And to refresh with latest code:

```
git pull https://github.com/BU-EC444/myrepo
```

Using git pull for pi

- We recommend that you test node.js code on your laptop before deploying on the pi.
- Using git on the RPi will greatly simplify deploying code on the RPi as you will not need to write and debug on the RPi.
 - Write the code on your laptop
 - Push to your git repo
- Pull from the pi to transfer working files to the pi
- ***Remember: don't save your build folders on git.***
- Do an `idf.py fullclean` before you push from your laptop

Skills: Pi Web Cam

Pi web cam

- Pi zero with Camera Board v2
- We're doing this 'headless' – no keyboard or monitor
- Be careful with the camera connector
- Multiple options available for software we will use
rpicam-apps:
https://www.raspberrypi.com/documentation/computers/camera_software.html#rpicam-apps
- Demonstrate streaming on browser either from pi or from local network browser
 - Use rpicam-vid to source the video
 - Vlc to play the video from another machine
 - https://www.raspberrypi.com/documentation/computers/camera_software.html#stream-video-over-a-network-with-rpicam-apps

